Guru Guide
Chapter 6. How to Use Tabs ........................................................................................................211
6.1 New Tab .............................................................................................................................211
6.2 ListTab ..............................................................................................................................214
6.3 Map ....................................................................................................................................214
6.4 Calculator Tab .....................................................................................................................270
6.5 ChartTab ............................................................................................................................271
6.6 Container Tab .....................................................................................................................299
6.7 Forum Tab ..........................................................................................................................301
6.8 Facebook ............................................................................................................................308
6.9 Report List Tab ...................................................................................................................310
6.10 Gallery ...............................................................................................................................312
6.11 JS Plugin Tab ......................................................................................................................359
6.12 LinkedIn Tab .....................................................................................................................361
6.13 Login Form .........................................................................................................................362
6.14 Logout ................................................................................................................................380
6.15 Nearby .................................................................................................................................382
6.16 Notifications .......................................................................................................................396
6.17 PDF Form Tab ...................................................................................................................407
6.18 PDF Viewer Tab ..................................................................................................................409
6.19 Catalog Tab .........................................................................................................................412
6.20 QR Code Tab ....................................................................................................................416
6.21 QR Scanner Tab ..................................................................................................................427
6.22 Reminder Tab ......................................................................................................................432
6.23 RSS ....................................................................................................................................434
6.24 Web Search Tab ..................................................................................................................437
6.25 Shopping Cart Tab ..............................................................................................................437
6.26 Sign/Draw Tab ....................................................................................................................439
6.27 Sign up Form Tab ...............................................................................................................443
6.28 Twitter Tab ........................................................................................................................450
6.29 Form Tab ............................................................................................................................453
6.30 Web Tab .............................................................................................................................454
6.31 YouTube Tab ......................................................................................................................456
Chapter 7. How to Use Elements and Buttons

7.1 Text Area .......................................................... 458
7.2 Container Button ..................................................... 479
7.3 Image Button .......................................................... 481
7.4 Call Button ............................................................ 491
7.5 Email Button .......................................................... 493
7.6 Web Button ............................................................. 494
7.7 Calendar ................................................................. 495
7.8 Form Button ............................................................ 500
7.9 Map Button ............................................................. 626
7.10 Gallery Button ......................................................... 672
7.11 Report List Button .................................................... 674
7.12 Add to Contacts Button ............................................. 685
7.13 Add to Favorites ....................................................... 687
7.14 Audio Player Button .................................................. 702
7.15 Calculator Button .................................................... 707
7.16 Catalog Button ....................................................... 713
7.17 Chart ................................................................. 714
7.18 Directions .............................................................. 714
7.19 Facebook Button ..................................................... 718
7.20 Forgot Password ..................................................... 719
7.21 In-App Purchase ...................................................... 722
7.22 JS Plugin Button ...................................................... 726
7.23 LinkedIn Button ...................................................... 726
7.24 Location Button ....................................................... 727
7.25 Login Button .......................................................... 731
7.26 Login Form Button ................................................... 733
7.27 Logout Button ......................................................... 734
7.28 Navigation Button ................................................... 734
7.29 Nearby Button ......................................................... 744
7.30 PayPal ................................................................. 746
7.31 PDF Form Button ..................................................... 748
7.32 PDF Viewer Button ................................................... 754
7.33 Photo Button .......................................................... 759
8.3 Boolean Button ................................................................................................................................. 761
8.2 Barcode Field ..................................................................................................................................... 763
8.1 Audio Field ......................................................................................................................................... 766
7.61 Payments in apps (Stripe) .................................................................................................................. 767
7.60 New drawing functionality ................................................................................................................ 768
7.59 Geo Tracking ..................................................................................................................................... 769
7.58 Ongoing Notifications ....................................................................................................................... 772
7.56 One-Time Notifications ..................................................................................................................... 777
7.55 YouTube Button ................................................................................................................................ 780
7.54 Web Search Button ........................................................................................................................... 782
7.53 Video Player Button ............................................................................................................................ 787
7.52 Twitter Button ..................................................................................................................................... 795
7.51 SMS Button ....................................................................................................................................... 797
7.49 Sign up Button .................................................................................................................................... 803
7.48 Sign/Draw Button ............................................................................................................................... 806
7.47 Shopping Cart Button ........................................................................................................................ 808
7.46 Share Button ...................................................................................................................................... 809
7.45 Email Button ...................................................................................................................................... 812
7.44 Select Contact Button ....................................................................................................................... 813
7.43 Select Button ....................................................................................................................................... 815
7.42 Section Break Button ......................................................................................................................... 822
7.41 RSS ..................................................................................................................................................... 827
7.40 Remove Account Button .................................................................................................................... 828
7.39 Reminder Button ................................................................................................................................. 829
7.38 QR Scanner Button ............................................................................................................................. 835
7.37 QR Image Button ............................................................................................................................... 841
7.36 QR Code Button ................................................................................................................................. 841
7.35 Post to Twitter Button ......................................................................................................................... 862
7.34 Post to Facebook Button .................................................................................................................... 862
7.33 Post to Pinterest Button ...................................................................................................................... 870
7.32 Select Button ...................................................................................................................................... 880
Preface
Snappii is an online mobile application development platform that allows individuals and companies of any size to easily and quickly build high quality custom apps. No programming skills are required.

The Snappii platform enables you to rapidly build apps without coding and having to build an app for each individual platform such as Android or iOS. With our WYSIWYG (What You See Is What You Get) drag and drop editor, you will see text and graphics closely to the form in which they will appear on mobile devices, while Snappii’s Preview app will allow you to test your app at any stage of the app development and as often as necessary. Our editor also enables you to submit your apps easily and conveniently to the app stores.

With the robust nature and flexibility of the Snappii platform, app creation is exponentially faster than when writing “code” is involved and usually takes from a few days to a few weeks. Updating your app in the apps stores is also easy as all you will need is just to click “publish” to distribute the changes to thousands of mobile devices at once. Not surprisingly, all these significantly bring the cost of app development and maintenance down, thereby enabling you to build and release apps faster and at a lower cost. Last but not least, while no coding skills are required to successfully build and submit apps to the app store, experienced programmers can further extend the platform capabilities through JS/HTML for additional features and functions.

Chapter 1. Options of Mobile Development on the Snappii Platform
Snappii offers two ways of app development: Do-It-Yourself and Custom Development.

1.1 Do-It-Yourself (DIY) Option

App development with the DIY option, as well as testing with the Snappii preview app, is free during the 30-day trial period. You select a paid subscription only after you are confident your app is ready to be released.
Snappii’s WYSIWYG (What You See Is What You Get) editor allows building apps from scratch or using Demo apps as templates. In the latter case, you will customize the app of your choice by adding new features and branding elements and changing the visual design.

1.2 Custom Development
Snappii’s full service mobile app development is ideal for companies that are short on time and development resources or just aren’t sure exactly how to go about creating a robust, user friendly app for their business. Our Custom Development options include the following:

1. **Full development from scratch**: We take your ideas and mock ups and create a fully functional app from the ground up.

2. **Template customization**: Select from one of our numerous ready-made apps, and we will add all your branding (logos, photos, etc.) elements to it. Additional functions and features can be added to the app on an hourly or a per project basis.

3. **App additions**: The Snappii team can augment your development team by providing custom development services to the apps you are building on our platform. We can take as much or as little of the development piece as you need, including custom development for the functionality that is not yet available.

Please specify your requirements for custom app development in this form so that we can give you an estimate on how long it will take to build and test your application and what the total cost will be.

Whether you select the DIY or Custom Development option, Snappii’s support team can assist you by offering you one-on-one online training sessions. During these sessions, paid on an hourly basis, a member of our support team will show you a few examples with different features applicable to your potential app, help you add the required functionality to it, and answer all your questions. You can take advantage of this service at any time and don’t have to be on a subscription plan.

**Chapter 2. What Mobile Platforms We Support**
Currently Snappii specializes in creating native iOS and Android apps, the native iPhone, iPad and Android (phone and tablet) platforms in particular.
Our WYSIWYG editor allows rapid and convenient submissions of Android apps to the Google Play Store. An app submitted to the Google Play Store usually appears in the store and becomes searchable within 24 hours of submission.

**Chapter 3. The Overall Platform Description**
Regardless of which of these two options you will go with, the following holds true.

In the editor, you will see an emulator, where you can drag and drop tabs and buttons to add text and graphics, change color and predefined app themes, and upload spreadsheets, to name but a few.
3.1 App Settings
App settings are located on the left-hand side. These settings are mostly related to how your app will appear in the app store and certain functionalities regarding PayPal settings and user management.

In the App Settings section, you can add and display the app icon, splash screen, choose colors and fonts, enter the app name, descriptions and tags for the Google Play, choose the app language, manage SMTP (Simple Mail Transfer Protocol), determine what users will be able to access the app, configure PayPal settings, add a cloud storage service, and manage ads and in-app purchases.

3.1.1 Icon
Presently your icon is set by default as Snappii. To change it, click on the icon button and select the icon you are going to use to represent your app in the app store. You can upload something from your computer; you can specify a web page to pull the information from; you can find it on the web by searching a particular item, or you can enter a specific URL from Dropbox, for example.

3.1.2 Splash Screens
The splash screen is what will first appear when someone opens your app. Here you might want to share some information about your company, such as its name, logo, contact information, etc.
You will need to upload splash screens specific to mobile devices your application is designed for, be it for the universal canvas (intended to be used by both iOS and Android phones and tablets), Android phones/tablets only, or iPhone 5/iPad only.

Please see the screenshot below for the splash screens we currently require and their resolutions.
3.1.3 Colors and Fonts
To choose colors and fonts for your mobile application, you can use any of the ready-made color themes as they are, adjust them (after you set them as default), or manually choose and combine them as you wish.

3.1.3.1 Color Theme – is a number of ready-made color themes.

Here are a few examples.
Color themes

- General
  - Background
  - Foreground
  - Navigation bar

- Lists
  - Search bar
  - Alternating row 1
  - Alternating row 2

- Font
  - Primary text
  - Secondary text

[Image of a mobile phone interface with a title bar and list items]
3.1.3.2 Global

This section lets you choose the background and foreground of the entire app.
Select a color for the background of your app.

For example, if I choose *light crimson*, the background of my entire app will be light-pink.

Select a color for the foreground of your app.

This determines what color the labels of the elements and buttons of the entire application will be.

For instance, if we choose to make the foreground dark purple (B02B61), then the color of the labels of any added element will be dark purple, as shown below.
3.1.3.3 Font Settings

This section allows you to set up the font used in your app.
The following options are available:

- Select font
- Change specific font

### 3.1.3.3.1 Select Font

This option enables you to specify the font for all the elements used in your app.

To illustrate how this option works, let us select *Courier New*
3.1.3.3.2 CHANGE SPECIFIC FONT

This option enables you to change any font from the first drop-down menu (From) to any font from the second drop-down menu (To) for all the elements using the selected font from the first drop-down menu. This change will keep those elements utilizing fonts different from our specified font intact.

To illustrate how this option works, we will change Courier New to Arial.

In the screenshot below, you can see that our Text element called Second utilizes the Courier New font.
After we have changed Courier New to Arial in the Font Settings section of the entire app, our Text element has automatically changed its font to Arial.
3.1.3.4 Navigation Bar

This section enables you to choose a color option for the navigation bar which appears at the top of the app.
3.1.3.5 Search Bar

This section lets you select a color option for the search bars in your app.

3.1.3.6 Lists

This section allows you to configure the background and font color for any lists, such as lists in Advanced Lists, Image Gallery, and Advanced Map, etc.
3.1.3.7 Buttons

This section lets you choose a font color, a button color, and a button color when a button is pressed. The screenshot below illustrates how these options work.
Our only recommendation is that, when selecting a button color and a font color, you use a contrast checker. Here is an example of such free online resources allowing you to do so.

3.1.3.8 Hamburger menu

Hamburger menu – is a new type of the user interface that allows you to move the usual tabs from the bottom of the screen to the right side of it and let it slide. Let’s consider some new visual settings in the Editor and create a new app with a few tabs:

Hamburger menu will not affect the content of the tabs, it will just impact the app general layout.
Switch to App settings ➔ Visual Style. You will see a new setting – App Layouts:

Click this option to see the following settings:

You can select the way to display the tabs for iOS, Android, HTML5 and the editor. Side navigation will move all the tabs to the right hand side, Bottom navigation is the standard tabs view with all the tabs placed at the bottom of the screen. You can set up the layout of your apps just the way you want. You can also upload a picture that will be displayed at the top of the side navigation. The maximum width of the side navigation image is 960px and the recommended aspect ratio is 16:9. This information is available on the question mark button click next to Upload:
The maximum width of the side navigation is 960px. The recommended aspect ratio of header menu is 16:9.

Size of image header menu is 960px x 540px.

If you want to show the user information on the side navigation header, click a checkmark to enable that feature. This way, you will have the user name and email show up. Let’s upload an image, enable “showing user info on the header of the Hamburger menu” setting and see how it looks on the device:

Side navigation is convenient in a way it shows the app content entirely. When the menu is hidden, you can see the tab content in full.
Along with uploading an image and displaying user info, you can customize the look and feel of the menu. This can be done under Visual settings → Color themes → Navigation menu

Menu background color – is responsible for the color of the sliding menu

Menu foreground (text) color – is a color of the tab names

Selected item background color – is a color of the tab background you are clicking at the moment

Selected item foreground (text) color – is a color of the tab you are clicking at the moment

Divider color – is a color of the divider between the tabs

Header foreground color (text) color – is a text used to display the user info at the top Let’s add colors and check how they look on the device:
Navigation menu

Menu background color
Menu foreground (text) color
Selected item background color
Selected item foreground (text) color
Divider color
Header foreground (text) color

Tab navigation bar twice to view Preview buttons.
In order for the dividers to show, you need to click a checkmark in each tab settings:
Ability to select a user interface and flexible settings will allow you to create a unique style in each of your apps.

3.1.4 App Name
Here you can name the app you are working on.

Please keep in mind that the name of an Android app should not exceed 30 characters.
3.1.5 App Information
This section allows you to configure the way your application will be displayed in the app stores. This will determine how searchable, appealing to its potential users, and, ultimately, successful it will be.

3.1.5.2 Google Play Description (up to 4,000 characters).

The information about your application that you enter here will appear in the Google Play Store.

Here’s an example:
3.1.5.3 Google Play Short Description (limited to 80 characters).

This description has a very limited number of characters but plays a more important role in the indexing of your app in the Google Play Store than the longer description (see above).
3.1.5.4 Category

Carefully select a category for your app to make sure that it improves the ability of your app to be found in the app store.

3.1.5.5 Google Play Feature Graphic

The feature graphic tool helps you attract more users in the Google Play Store and is displayed at the top of your store listing page in the Google Play Store app on your Android device. If you've added a promo video, a Play button will overlay your feature graphic so users can play your promo video.

Here’s an example:
3.1.5.6 Push Notification Icon for Android

If you enable push notifications in your app, you’ll need a push notification icon, which will indicate the origin of the app sending the notification.
3.1.5.7 Tags

Tags are key words describing the nature and features of your app and that help your app to get found in the app stores.

![App settings interface with tags field highlighted]

3.1.5.8 App web site

If you have a particular website where you are going to feature your app and you’d like to direct people to that website through the app store, you might want to enter this website here.
3.1.6 Miscellaneous

### 3.1.6.1 Language Used in the App

Our editor supports a number of languages, such as Arabic, French, German, Italian, Spanish, Japanese, etc. When you chose a different language from the dropdown menu and then save the app, the changes won’t be visible in the editor itself. You’ll need to use the Preview App to see them.
These changes only affect the default buttons. Any tab, button, or field, which users can name themselves, needs to be renamed manually. However, you can change the translation of the buttons yourself by clicking on Change translation from English and entering your variant in the column on the right.
3.1.6.2 Facebook App ID

Here you can add the ID of your app built using Facebook App development tools. This would enable the app you are making on the Snappii platform to pull some information from the Facebook app on the Nearby and Forum tabs. This field is optional.

To learn more about Facebook app building tools, please click here.

3.1.6.3 Error Logs Email Address

To enable our platform to send you error codes found in your Android app and allow our programmers to promptly fix them, enter your email address in this field.
3.1.6.4 SMTP

SMTP (Simple Mail Transfer Protocol) is a protocol for sending email messages between servers. Most email systems that send mail over the Internet use SMTP to send messages from one server to another.

Please see the screenshot below for more information.

3.1.6.4.1 HOST

SMTP Host is the server sending the email. Most server names are written in the form "smtp.domain.com" or "mail.domain.com": for instance, a Gmail account will refer to smtp.gmail.com. But that's not a unified rule, so you should pay attention and get the right SMTP parameters. You can find more information here.

3.1.6.4.2 PORT

Port 25 - this is the default SMTP non-encrypted port.

Port 465 and Port 587 are used to send messages using SMTP securely.

As the screenshot below indicates, if you use a Gmail account, enter 587. Otherwise, contact us at support@snappii.com
3.1.6.4.3 LOGIN
This is the login you use to sign into your email account. For instance, if your email address is mike1964@gmail.com, then your login is mike1964.

3.1.6.4.4 PASSWORD
This is the same password you use to sign into your email account.

3.1.6.5 Show logout button on first tab
3.1.6.6 Show settings button on first tab

This check box allows adding the Settings button.
3.1.6.7 At least one login tab required

We recommend checking off this box as a reminder not to delete the Login tab and so that you won’t be able to save the app without at least one Login tab.
All tabs except for the Login tab are usually made invisible to anonymous users in the User Permissions section of each tab. (Read more in Chapter 3: User Management). That is why, if the Login tab gets removed or hasn’t been created yet, no tabs will get displayed to an anonymous user in the Preview App, making it impossible for the user to log in and view the content of the app.

3.1.6.8 Back button function for Android

Opting for Close the app for the Back button, will display Press BACK again to exit every time the Back button is pressed and will allow you to exit the app.
Opting for *None* won’t let you close the app by pressing the Back button.

3.1.6.9 Use Redlaser as QR/Bar code scanning mechanism (for iOS only).

Checking off this box and uploading your license to use Redlaser will enable you to utilize Redblaser’s QR and Bar code scanning. (Read more on the Redlaser licensing [here](#).)

To purchase a RedLaser license you need to know the app bundle ID: com.store2phone.bu1257609a80948

OK
3.1.6.10 PDF Report Templates

This section allows configuring PDF reports by uploading headers, footers, selecting font, color themes, and the overall layout.
(The same options are available on some tabs and buttons in the Share Settings section.)

3.1.6.11 Pop-up Dialogs

Here you can set up pop-up dialogs asking your app users to rate and review your app, as well as register, if you allow anonymous users to view the content of your app. All you need to do is to specify how many times a user needs to open the app before the pop-up screen appears. You can also enter the URL of the app’s page in its app store.
3.1.6.12 Automated Emails

Here you can set up automatic emails: sign up emails and inactivity notifications.

3.1.7 User Management

Here you can define specific user access within your app. Within the Snappii platform, you have the ability to have specific items be seen by certain users and not seen by other users. Please see Chapter 3 for more detailed information on user management and levels of in-app access.
3.1.8 PayPal Settings

3.1.8.1 Live App ID (Android)

3.1.8.2 Sandbox Merchant PayPal ID

“The PayPal Sandbox is a self-contained, virtual testing environment that mimics the live PayPal production environment. It provides a shielded space where you can initiate and watch your application process the requests you make to the PayPal APIs without touching any live PayPal accounts.” (Read more here.)

“Note: You need a Buyer test account in order to complete transactions in the Sandbox. Completing a transaction typically entails going through a PayPal purchase flow as a buyer. Be sure to execute your test transactions on the test site (https://www.sandbox.paypal.com), and not on the live PayPal site (https://www.paypal.com).” (Read more here.)

To obtain your Sandbox Merchant PayPal ID, take the following steps:

1. Go to https://developer.paypal.com/
2. Click on **Dashboard**

3. Click on **Accounts** under **Sandbox**.

4. Click on **Log in** and log in with your existing PayPal credentials or sign up.

5. Click on **Create Account** and fill in the fields. Make sure to select the Business account type.

*(First Name, Last Name, and Payment Balance are optional fields. You can choose a type of the credit card associated with this account with a fake credit card number associated with this.)*

Click on **Create Account**.
6. Now you can see the merchant account you have just created.

7. Enter https://www.sandbox.paypal.com with the newly created credentials.

Under My Profile – My Business Info, you will see the 13-character Merchant Account ID – this is the ID you’ll need to enter.
3.1.8.3 Live Merchant PayPal ID

Once you've finished coding and debugging your application (including testing all your PayPal NVP/SOAP API calls in the Sandbox), go live by moving your application to PayPal's production environment.

To take your application live:

1. Ensure your application adheres to the PayPal Application Policies and Guidelines.
2. Update the Sandbox API credentials in your PayPal calls to those assigned to a live PayPal account.
3. Update your PayPal endpoints from the Sandbox to the PayPal production servers.
4. Register your application with PayPal.
5. If your application makes use of Adaptive operations, obtain a live AppID.
6. Go Live!

For more details, please click here.
3.1.8.5 Merchant Name

3.1.8.6 Shipping and Tax

<table>
<thead>
<tr>
<th>countryCode</th>
<th>shippingType</th>
<th>beginPriceRange</th>
<th>endPriceRange</th>
<th>shippingPrice</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td>standard</td>
<td>0</td>
<td>4.99</td>
<td>2</td>
</tr>
<tr>
<td>CA</td>
<td>standard</td>
<td>0</td>
<td>4.99</td>
<td>3</td>
</tr>
<tr>
<td>US</td>
<td>standard</td>
<td>5</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>CA</td>
<td>standard</td>
<td>5</td>
<td>100</td>
<td>1.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>countryCode</th>
<th>state</th>
<th>taxRate</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td>AL</td>
<td>1</td>
</tr>
<tr>
<td>US</td>
<td>AK</td>
<td>1</td>
</tr>
<tr>
<td>US</td>
<td>AZ</td>
<td>1</td>
</tr>
<tr>
<td>US</td>
<td>AR</td>
<td>1</td>
</tr>
<tr>
<td>US</td>
<td>CA</td>
<td>1</td>
</tr>
<tr>
<td>US</td>
<td>CO</td>
<td>1</td>
</tr>
<tr>
<td>US</td>
<td>CT</td>
<td>1</td>
</tr>
<tr>
<td>US</td>
<td>DE</td>
<td>1</td>
</tr>
<tr>
<td>US</td>
<td>FL</td>
<td>1</td>
</tr>
<tr>
<td>US</td>
<td>GA</td>
<td>1</td>
</tr>
<tr>
<td>US</td>
<td>HI</td>
<td>1</td>
</tr>
<tr>
<td>US</td>
<td>ID</td>
<td>1</td>
</tr>
</tbody>
</table>

3.1.8.7 IPN URL (optional)

3.1.8.8 Merchant Custom ID (optional)

3.1.9 Cloud Storage
To the left of the canvas, just under PayPal, you will find Cloud Storage settings.
Click on the plus sign to add a new cloud storage provider. You will see a window with the following options:

- Dropbox
- Box
- Google Drive
- MS One Drive
If you put a check mark next to *Provide a key and secret for my own app folder in my cloud storage choice*, you will be asked to enter your key and secret. This feature allows you to specify the exact folder on Dropbox where the data will be stored. (Only Dropbox offers this feature.) You can leave this option unchecked, and the data from your app will go to the root of the cloud storage service selected. *To learn more about getting a key and secret for your app, please click here.*

Next, click *Save and Create Token* to be redirected to the login page of the selected storage provider.

(In our case, we opted for Dropbox.)
Click *Allow*, if already logged into the cloud storage service, or simply log in with your credentials, and you will see the message “Successfully authorized!”

When you go back to the Snappii Editor, you will see two new buttons.

Now you can edit or delete this storage option or add another one.
These cloud storage options allow saving reports generated every time a form gets submitted.

For the cloud storage option we have set up to work, you need to configure it in the settings of the Form element under *Submit to Cloud Storage*.

Put a check mark next to *Cloud Storage Mode* and choose your storage option.
Click Ok.

**Use Case:** We have an app that allows its users to fill out a daily report and send it. Once the report is filled out, the app user clicks Submit, and the report gets sent to the user’s cloud storage service.

---

**How to save files to specific folders in Box?**

Open the settings window of the form that sends reports to the Box storage in the Snappii Editor.
Under "Upload Reports to Cloud Drives" section if you select the Box storage you can find the setting Storage Folder Id. By default it has 0 value and it means that your reports will go to the root of the storage. You need to change this value if you want your reports to go to a specific folder.

To figure out the Id of a folder in your Box storage, please open this folder in your browser and check the URL in the address bar. The folder Id should be extracted from the URL the following way: just copy and paste the folder Id to the settings window as per the screenshot above and save the form and the app.
3.1.9.1 Getting a Key and Secret for Your App

To specify an exact folder in Dropbox where you would like to store data from your app, first of all, you will need to get a special key and secret generated by Dropbox and then enter them into Snappii’s editor. Here are the steps to follow:

1. In your Dropbox account, find the Developers section and click on it.

2. In the new window, click on My apps.
Build your app on the Dropbox platform

A powerful API for apps that work with files.

Read our docs
Docs are organized by language, from .NET to Swift.

Create your app
Getting started is simple and quick from the App Console.

Test your ideas
It's easy to prototype and test examples with our API Explorer.

Learn from our examples
3. Select *Create app.*
4. Choose an API, the type of access you need, enter the name of your app, and click *Create app.*
5. A new window will open up, and there you will see your app key and secret. Copy them and enter them in the corresponding fields in the Snappii editor. Click on *Save and create token.*
If the following error pops up,

```
Error (400)
```

It seems the app you were using submitted a bad request. If you would like to report this error to the app's developer, include the information below.

**More details for developers**

*Invalid redirect_uri: "https://www.snappii.com/oauth.aspx": It must exactly match one of the redirect URIs you've pre-configured for your app (including the path).*

go back to the Dropbox page with the app key and secret and enter

| [https://www.snappii.com/oauth.aspx](https://www.snappii.com/oauth.aspx) under Redirect URIs. Click Add. Then refresh the page and make sure that the entered address ([https://www.snappii.com/oauth.aspx](https://www.snappii.com/oauth.aspx)) is still under Redirect URIs. |
Then enter the key and secret again and click on Save & Create Token.

This time you should see a message like this. After you click Allow,
you will finally get this:

![Image](https://www.snappii.com/oauth.aspx?state=255198&code=fc-9FgLQxEsAAAAAAAABA12u-e2pT0SEB89eU6adLY)

Successfully authorized!

Now hit *Check token*.

6. The following message will appear.
That is all you need to do to connect your app to a specific folder in Dropbox.
3.1.10 Ads and In-App Purchases

3.1.10.1 Banner Ads and Interstitial Ads

Applications built on our platform support banner* and interstitial** ads.

* **Banner ads** are shown at the top or at the bottom of the screen.

** **Interstitial ads** are full screen ads that can be closed by app users. “They’re typically displayed at natural transition points in the flow of an application, such as between activities or during the pause between levels in a game. When an app shows an interstitial, the user has the choice to either tap on the ad and continue to its destination, or close it and return to the application, “says Google at [Google Developers](https://developers.google.com).
In the Snappii Editor, banner ads can be added and configured in the settings of tabs and elements for which you would like to display the banner ads, while interstitial ads are set up in App Settings section under Ads and In App Purchase.

### Banner Ads

![Banner Ads](image)

### Interstitial Ads
3.1.10.1.1 Adding and configuring banner ads

Complete the following steps to create an ad unit for an existing app.

2. Click the Monetize tab.
3. Select your application from the All apps list on the left-hand side.
4. Click + New ad unit.
5. Click Banner
6. Provide the following details for the ad unit:

**Ad type:** Choose the ad type(s) allowed for this ad unit. You can choose any combination of text and image by checking the boxes. Checking both boxes is recommended since unchecking either of the boxes might limit the revenue generated by the ad unit. Unchecking all of the boxes will result in an error. You must choose at least one ad type for each ad unit.

**Automatic refresh** Determine how often a new ad impression is generated. You can choose not to refresh or to refresh ads every 30 to 120 seconds. We recommend setting a refresh rate between 45 and 60 seconds.
Text ad style: Select a text ad style that complements your app. You can use the standard style or customize your own style.

Ad unit name: Enter a unique name and description that will help you find this ad unit later (e.g., Top Banner on Home).

7. Click Save. You'll see the ad unit ID for this ad unit.

Make sure to copy it and paste it in the corresponding sections in the Snappii editor, which are located in the settings of each tab/element to which you’d like to apply the banner. Here you can also configure the position of the ad: at the top or bottom of the screen.

8. Click Done to return to the Monetize tab or click Send an email with these instructions to have instructions sent to you via email.

You've now created a banner ad unit for your app.

3.1.10.1.2 Adding and configuring interstitial ads
Complete the following steps to create an ad unit for an existing app.

2. Click the Monetize tab.
3. Select your application from the All apps list on the left-hand side.
4. Click + New ad unit.
5. Click Banner
6. Provide the following details for the ad unit:
   **Ad type:** Choose the ad type(s) allowed for this ad unit. You can choose any combination of text, image, and video by checking the boxes. Checking all three boxes is recommended since unchecking some of the boxes might limit the revenue generated by the ad unit. Unchecking all of the boxes will result in an error. You must choose at least one ad type for each ad unit.

   **Frequency capping:** Choose whether or not to set a cap on the number of impressions you want to allow to show to each user. Limits the number of times ads are shown to the same user per minute, per hour, or per day for a given ad unit.

   **Ad unit name:** Enter a unique name and description that will help you find this ad unit later (e.g., Interstitial on level complete).

7. Click Save. You'll see the ad unit ID for this ad unit. Make sure to copy it and paste in the corresponding section in the Snappii editor.
Here you can also track and control the number of clicks a user can make before the interstitial ad starts popping up. To do so, just enter this number and specify the places - the Custom Tab, the other tabs, or cell view – where the clicks will be counted.

8. Click *Done* to return to the *Monetize* tab or click *Send an email with these instructions* to have instructions sent to you via email.

You've now created an interstitial ad unit for your app.

**3.1.10.2 In -App Purchases: Subscription Products**

**3.1.10.2.1 Subscription products for ios devices**

To set up in-app purchases for iOS, you need to do the following:

1. Open your app in Snappii’s editor.
2. Go to *Ads and In App Purchases*.
3. Click on *Subscription Products* → Enable → Monthly/Yearly or both.
4. Save the app and create a web2phone copy. Select *iOS web2phone copy*, check *Add this app to iTunes connect* and click *OK*.

5. After your iOS web2phone has been generated, you can find the app in your iTunes account. Login to itunesconnect.apple.com → click on *My Apps* → select your app from the given list.
6. Select Features → In-App Purchases and click on the cross sign.

7. Select Auto-Renewable Subscriptions to set up monthly and annual subscriptions.
8. Click on *Select* and enter any Reference Name (*Subscription*, for example).
9. Set up the duration and pricing by hitting Add Duration, choosing the duration from the drop-down menu, entering the Product ID, and selecting the price tier.

10. Make additional configurations, if necessary, and click Save.
Important: If you want to create both annual and monthly subscriptions, do not create a new Auto-Renewable Subscription. Instead, please click Add Duration in the created subscription.

11. Set up the annual subscription if necessary. (Follow again the steps above to do so.)
12. Click Add Language and fill out the required fields.

13. Now click Choose File and upload a screenshot that specifies the in-app purchases your offer.
14. Save. A pop up window will ask you to enter the URL of your Privacy policy.
15. Click Save.
16. Go back to the Snappii editor and enter the generated unique IDs into iTunes Product ID for both monthly and annual subscriptions.

17. Scroll down to *Free form submission limits* and insert the number of free submissions you allow per ever/month/day.
For instance, if you allow users to make 2 free form submissions per day, the app will prompt them to purchase the monthly/annual subscription after they exceed these 2 submissions. Next day, they’ll be able to submit another 2 forms before being asked to pay for the subscription. Similarly, selecting month will give them 2 free submissions per month, whereas opting for ever, will only enable them to submit 2 forms in total.

18. Put a check next to Enforce registration if you want to collect users’ contact info and won’t allow unregistered users to go beyond the registration screen of your application.

19. Save the app.

### 3.1.10.2.2 Subscription Products for Android Devices

Adding in-app purchases for Android devices does not require creating web2phone copies. In fact, in-app purchases can be created right after the first app submission.

1. Open your app in Snappii’s editor.
2. Go to Ads and In App Purchases.
3. Click on *Subscription Products* → Enable → Monthly/Yearly or both.

4. Create and enter a unique Google Play Product ID in the Editor before you submit your app. Save the app.

5. Submit the app to the Google Play Store.

6. After the apk file has been uploaded to Google Play developer console, you can set up in-app purchases.

7. Log into your **Google Play developer account**. Click *All Applications*, find the app and select *In-app Products* in the menu on the left.
8. Click Add New Product.

9. Select Subscription.
10. Enter the unique ID you created in the Snappii editor and click Continue.

11. Fill out the fields shown in the screenshot below.
Important: Keep in mind that once the info has been saved, you can’t make changes to the pricing.

12. Save it. Click *Activate* on the top of the screen.

13. Configure the annual subscription, if necessary, in a similar manner, save it, and *Activate*.

14. Go back to the Snappii editor and click *Resubmit your app to Google Play* to enable the in-app purchases.

### 3.1.10.3 Ad Removal Products

#### 3.1.10.3.1 Ad removal for android devices

To set up ad removals for Android devices, you need the following information:
1. Google Play Product ID. You should have a valid Google Developer Account. Before setting up ad removal in your app, you should submit the app to Google Play without in-app purchases enabled.

2. Once you have submitted the app, you should log into Google Developer Portal http://www.google.com/accounts/Logout?continue=https://play.google.com/apps/publish

3. Click All Applications, find the app and select In-app Products in the menu on the left.

4. Click Add New Product.
5. Select *Managed Product* and enter your product ID, which should be a unique ID for your app. Example: For *Billable Hours Tracker*, we could use: `ad_removal_for_billable_hours_tracker`. Click *Continue*.

**ADD NEW PRODUCT**

*What type of product would you like to add?*

- Managed product
- Subscription

Managed items that can be purchased only once per user account on Google Play. Google play permanently stores the transaction information for each item on a per-user basis.

**Product ID**

`ad_removal_for_billable_hours_tracker`

37 of 132 characters

Please note that you can NOT change the product type and product ID later and that you cannot re-use the product ID again. Learn more

6. Add the title, some description for your app, and set the price. (You can auto-convert the price).
7. Click *Activate* at the top and publish your app changes.

8. Copy your Product ID and paste it in the corresponding section in the Snappii editor. Make sure to add a checkmark next to *Enable.*

### 3.1.10.4 How to Set up the In-App Purchase Button?

To enable your users to pay for their monthly/yearly subscription and ad removal, drag and drop the In App Purchase Button onto the canvas.
In the pop-up window, configure the label, icon, dimensions and font of the button. Next to *Google Play product ID*, choose which of these three options the buttons will be used for.
Click OK. We added three buttons for each of the three options. Here is how they look in the editor and on a mobile device.
3.1.10.5 License Key for Android Purchases

Licensing prevents unauthorized distribution of your app and verifies in-app billing purchases. For more information, please read here.

1. Before setting up ad removals and subscription in your app, you should submit it to Google Play without in-app purchases.


3. Next, open your app and go to Services & APIs.

4. Now copy your base 64 encoded PSA public key.
5. Finally, paste it in the corresponding section of the Snappii editor.

3.2 Canvases

The Canvases sections allows you to determine whether the app is meant for iPhones, iPads, Android phones, and Android Tablets specifically or across all these devices (Universal Canvas).

3.3 Alignment

The Alignment section consists of a set of options that allow you to align elements on the screen with ease.
3.3.1 Vertical Grid Line
When you select *Vertical Grid Line*, a vertical line that you can manually place anywhere on the screen appears. Just click on the line and drag it to where you want to place it.

The `→` and `←` buttons enable you to align your element to the left of the line or to the right of the line accordingly. All you need to do is to click on `→` and then on the element you need to align.
or on and then on the element you need to align.

at The grey cross button at the top of the screen allows hiding the line (see below), and the red cross completely removes it from the screen.

3.3.2 Horizontal Grid Line
This line works similarly. Just select Horizontal Grid line and drag it to where you need to place it.
Then click on ▲ and select the element you need to align.

or ▼ and the element you need to align.
The grey cross button at the top of the screen allows hiding the previously selected lines.

(Before)

(After)
button completely removes the selected line from the screen.

### 3.3.3 My Grid Lines
This option reveals all the previously hidden lines.

![Image](image.png)

The red cross  

### 3.3.4 Select All
This option allows selecting all the elements on the screen to align them at the same time.

(Before)
3.3.5 Align Center

This option enables you to align simultaneously one or a number of elements to the center.
3.3.6 Undo

*Undo* allows you to undo the last action.
3.3.7 Show Grid

This option reveals the grid to more easily align the existing elements.

To hide the grid again, simply click on *Hide Grid*.

3.3.8 Snap to Grid

*Snap to Grid* option brings the left side of the element to the nearest grid line. (Before)
Each cell is 10 by 10 pixels.

3.4 The Main Section

- Up at the top, there are such items as Save App,
- Save App As,
- Submit/Publish,
- Tutorials,
- New App,
- Data Sources,
- QR Codes,
• Preview, and
• My Apps.

On the right, you can see the main core of the Snappii platform. These are the drag-and-drop functions that you will be able to utilize to visually build your application. These are tabs, elements, fields, and buttons, which can be dragged over and configured to create your app.

3.4.1 Tabs

Tabs are placed on the lower bar of your app. Just select the tab you need from the Tab section and drop it on the lower bar. Then you can edit the tab, copy it, specify what types of users can see it, or simply delete it.

3.4.2 Elements

Unlike tabs, placed on the lower bar, elements are located on specific tabs.
Thus, for instance, we can place the Call and Send Text elements on the Home tab by dragging them and then copy, edit, or delete these elements as necessary.
To see the copy, edit, and delete options, just click on whatever element you need.
3.4.3 Fields and Buttons

Most of the times, you will need to use this section when you are working on a form within a specific tab, implementing custom search in advanced controls, such as Advanced List, Gallery Layout, and Advanced Map, configuring the visibility of elements and buttons, or performing actual calculations.

Thus, for instance, you can drag and drop the Text/Number element and then copy, or delete it as necessary.
Along with the manual, we offer free video tutorials and paid one-on-one online training sessions, meant to further assist you with using our platform. (Once you select a subscription plan, you will be able to utilize whatever training and support options are available with that subscription.)

Please note that our Live Chat team is available to address potential bug issues but is not meant for “how to” questions.

!!! App development with the DIY option, as well as testing with the Snappii preview app, is free during the 30-day trial period. You select a paid subscription only after you are confident your app is ready to be released.

3.5 Custom Amazon S3
You can use your Amazon S3 account to store all the data you get from your app, including images, audio, video and documents that are being uploaded via the forms in your app.

To add the info about your storage, go to Miscellaneous → Amazon S3 that is placed under App Settings to the left from the device:
Follow the steps to set up your Amazon S3 account:

Step 1. Amazon S3 bucket
   Choose US Standard as region for the bucket.
2. Create a folder inside the bucket
3. Enter the bucket name and the folder name into the Editor settings Bucket name and Bucket label.

Step 2. AWS user
1. Create a user on the IAM Users page (https://console.aws.amazon.com/iam/home?region=useast-1#users)
2. Copy Access Key ID and Secret Access Key and paste them to the Editor settings.
3. Open page of the user and go to Permissions tab
4. Under Inline Policies you need to have 3 policies:
Policy #1:
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [
"mobileanalytics:PutEvents",
"cognito-sync:*",
"cognito-identity:*"
],
"Resource": [
"*
]
}
]
}

Policy #2 (instead of "bucket-name" you need to use the bucket name from the step 1):
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": "s3:*",
"Resource": [
"arn:aws:s3:::bucket-name",
"arn:aws:s3:::bucket-name/*"
]
}
]}
}
Policy #3 (it should be added if app uses geo tracking)

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": "dynamodb:*",

"Resource": "*"

}

]

]

]

}

Step 3. Federated Identity
1. Create a federated identity on the Congito page
   (https://console.aws.amazon.com/cognito/federated?region=us-east-1)
2. While you create the identity please enter a developer provider name (you can use a string you
   want) under Authentication providers -> Custom. This value should be used in the Editor
   settings as Identity pool login.
3. After you click on Create button you will see the Roles page. Just click on Allow here.
4. Then you need to click on "Edit identity pool".
5. You need to copy Identity pool ID and paste it to the Editor setting.
6. Please remember the name of Authenticated role for the identity pool.
7. Open the Roles page on the IAM portal and find the role from the previous step.
8. Open the role page and find Role ARN. This value should be used in the Editor settings.
9. Go to the Permission tab for the role and add 2 new role policies with the same text as Policy #2 and Policy #3 above.

Chapter 4. Levels of In App Access and User Management
The user management option will help you specify the groups of users that will be able to use your app, e.g. thereby determining what specific tabs will be accessible to them. Find this button in the left part of the editor menu in “App Settings” under “User Management”.

4.1 User Types
By default, these are the following user types:

1) an anonymous user: someone who isn’t logged in and can only see the Log in tab, the Blog tab, the Related Apps tab, and the Customize tab;
2) a logged in user: someone who has logged into the app and the one who most of the times has more access to the app tabs and buttons than anonymous users do;

However, you can add additional user types and specify which specific tabs will be visible to which users. (See “Predefined users”).

4.2 Registration Modes
User Management offers 3 registration modes:

1. General mode allows anyone to register in the app and use the default user types.
2. Creating a list of *predefined users* and assigning tabs visible to them. (Predefined users don't have to register in the app. They get access to the app just by logging in.) To take advantage of this mode, please do the following:

1. Create a spreadsheet containing the following fields:

   1) Company Name
   2) Company ID
   3) userName
   4) emailAddress
   5) password
   6) memberId (optional)
7) userType (will default to LoggedInUser if not specified)

8) expiration date (required; shows when the user will lose the access to the app) A sample spreadsheet is available for your convenience. (Click on the question mark to the right of “Predefined Users” to download it).

Predefined user mode allows you to add specific known users to your app. You can specify user types for these predefined users so that they have special access to tabs that general users cannot see. For example, employees may need to be designated as "Administrator" so they can access tabs general users do not see.

Your Excel spreadsheet should have the following columns: userName, emailAddress, password, memberId (will default to empty if not specified), userType (will default to loggedInUser if not specified), expirationDate, company, companyId.

[Download sample]

Here’s an example:

Here’s an example:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>userName</td>
<td>emailAddress</td>
<td>password</td>
<td>memberId</td>
<td>userType</td>
<td>expirationDate</td>
<td>company</td>
</tr>
<tr>
<td>2</td>
<td>Jack Smith</td>
<td><a href="mailto:jack_smith@gmail.com">jack_smith@gmail.com</a></td>
<td>qwerty123</td>
<td></td>
<td>staff member</td>
<td>12/12/2030 0:0:0</td>
<td>ABC</td>
</tr>
<tr>
<td>3</td>
<td>admin</td>
<td><a href="mailto:admin@gmail.com">admin@gmail.com</a></td>
<td>admin</td>
<td></td>
<td>administrator</td>
<td>12/12/2030 0:0:0</td>
<td>ABC</td>
</tr>
<tr>
<td>4</td>
<td>Olivia Jones</td>
<td><a href="mailto:olivia_jones@gmail.com">olivia_jones@gmail.com</a></td>
<td>qwerty321</td>
<td></td>
<td>staff member</td>
<td>12/12/2030 0:0:0</td>
<td>ABC</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2. Upload the list by clicking on “Upload” to the right of “Predefined users”

3. Now you can see the list of the users under “Add new type”

and also in the permission settings of each tab:
4. Next, open the permission settings of each tab to determine what user types will be able to view it.

For example, we’ve decided that every user type will be able to see the New Tab by checking off each user type listed there.
However, we don’t want anonymous users to have access to the New Tab 1, so we remove the check mark next to it but make sure that the other user types are checked off.
5. Put a check mark next to “Predefined users”. You can also select General mode. For example, your staff members will access the app as is predefined users, but your clients will be able to register in it and log in as default users via the General mode.

4.3 In-App Members
If you have built an app and decided that you will sell user membership, you can do so. You will determine the membership fee and duration.

Here is how to use this functionality.

1. Create an Excel spreadsheet. A sample spreadsheet is available when you click on the question mark to the right of “Members” and then to “Download sample”
Members mode can be used to allow only users who purchase membership in an organization to have access to your app. Upload a list of available member ids, then when users purchase membership, you can give them their member id (or allow the app to automatically assign one), and when they register or login to your app they can supply the member id to get access.

Your Excel spreadsheet should have the following columns: memberId, itemType (will default to LoggedInUser if not specified), expirationDate (or membershipLength and membershipLengthUnits).

Download sample

Available values for membershipLengthUnits column are: days, weeks, months, years.

The app-builder specifies the IDs of the members by entering the spreadsheet with the following fields:

- memberId (required)
- userType (will default to LoggedInUser if not specified) • itemId (required) • itemId from PayPal element or product list
- Either: expirationDate (required for the admin; shows when the administrator will lose his/her admin access to the app) or membershipLength (required) • membershipLengthUnits (required) - days, weeks, months, years

Here is an example:
2. Upload the spreadsheet.

3. The new members/user types will appear under “Add New Type”

4. Now you will need to set permissions to these new user types in the same way you would do so for predefined users (see above). Also, make sure to check off “Members” and the other modes if necessary.

Please note that users can purchase their app membership and get assigned manually or automatically. Specify your choice by selecting one of the two options. Manually assigned members
After the membership has been paid for, the app builder sends users their member IDs that they can use to register in the app. If the user has already been registered as a general user, he/she will be able to update his/her profile and enter his/her member ID. **Auto assigned members**

Once the user pays for the membership and registers in the app, he/she gets assigned a member ID and an expiration date. When a user is assigned a member ID, the app creator will receive an email specifying the ID number, user’s information, and the PayPal transaction ID for the app builder’s records. Like manually assigned members, auto assigned members can also update their profile and enter member ID in case they were registered as general users.

!!!It is possible to enable the general, predefined, and membership modes at the same time. However, it is not recommended that a predefined user/member creates a new account through the general mode to avert confusion.

### 4.4 Registration Confirmation
If you enable registration confirmation,
then the following message will pop up,

![Image of the phone screen showing a message to verify email]

and, when the user clicks “OK, he/she will receive an email with a link to click on to confirm the registration.
5.1 Why to Use Data Sources and Data Connectors
Understanding how to create and use data sources and data connectors is essential to successfully building mobile apps on our platform. Not every tab, button, element, or field requires data sources or data connectors. However, building advanced lists, advanced maps, image gallery, and the universal form, which are the critical components of our platform, does require them.
You might not fully understand data sources and data connectors immediately, but as you start working with the above-mentioned components, you will. That is why we recommend that you refer to this chapter as needed.

5.2 How to Add Data to a Data Source
Data to your Data Sources can be added in three different ways.

1. Using the Data Source Dialog
2. Using the Advanced List tab/button
3. Using the Universal Form tab/button Let us discuss each of these options.

5.2.1 Using the Data Source Dialog. Building a New Data Source
To view all the currently available data sources and data connectors or add a new one, click on Data Sources, located to the right of the canvas and above the Tab Bar.

This is the pop up window, the so-called “Data Source Dialog, which you will use every time you need to view, build, add, edit, or delete data sources and data connectors.
On the left hand side, under **App data sources**, you will see all the data sources and data connectors you will have built and saved.

Under **Shared data sources**, you will see all the data sources and data connectors you will have built, saved, and shared among all of your apps.

Under **Data source templates**, you will see those data sources that we have pre-built for you.

Below we are going to illustrate how to add a new data source and how to copy an existing data source into your app.

The process of building a data source consists of three parts: creating an Excel spreadsheet, uploading and configuring its settings in the Data Source Dialog, and selecting how it is going to be displayed in the app.
5.2.1.1 Creating an Excel Spreadsheet

If we were to build a mobile app that would display a list of cupcake bakeries with their addresses, contact information, logos, menus, and hours of operation, we would create the following columns*:

- Name
- Menu
- Address
- PhoneNumber**
- Logo
- Email
- HoursOfOperation
- Location

*The cells of the first row in the spreadsheet will be used as the column names for the data table in the Data Source Dialog.

**The column names can only contain letters and cannot have any spaces.

5.2.1.2 Uploading the Excel Spreadsheet and Configuring Its Settings

A. Under Development Environment, click Upload and select the Excel file you have just created and saved on your computer.
B. On the pop up window, click OK.

C. Next to Name, enter whatever name you need to give to your app.

D. Next to Data Storage, select Global, Local, or Runtime, depending on where you’d like your information/data to be stored and who will be able to access it. Please see the explanations below.
E. Next to Type, select Data Table, Web Service, External DB, or Favorites depending on where the app is going to pull its data from. Please see the explanations below.

Data Table - choose this option to create a data table on Snappii cloud server. Load the table using Excel spreadsheet only. CSVs is not supported at this time.
Web Service - data is accessed from a remote web service. Choose web service to access or change data on internal enterprise servers or cloud based servers.
External DB - access to external databases.
Favorites - use this data source type to store bookmarks or copies of items from other data sources.

In our case, we opted for Data Table since we want our application to pull the information from the Excel spreadsheet we have just created.

F. Next to Time zone, select a time zone you need to use in your app. This feature is optional.

G. Check off the box next to Shared if you would like to make this data available in your other apps.
H. Match the type of each column with its name and ID. The column type determines how the information from each column is going to be displayed in the app. When you upload your spreadsheet, the Snappii editor tries to match each column type and upload its content automatically. However, it’s recommended that you always double check each column type. Otherwise, your data source might not work correctly.

5.2.1.3 Column Types:

- **Text**: any textual information is displayed under this type. However, please keep in mind that YouTube videos, Facebook, Twitter, LinkedIn and other social networks, as well as URLs also fall under this category.
- **Numeric**: any numbers, such as prices, ID numbers, order numbers, belong to this type.
- **Datetime**: this type is used to store and display dates and times
- **Image**: the URLs of the images entered in your Excel spreadsheet belong to this type.
- **Location**: utilize this column type when you use the longitude and latitude of a specific place.
Use the following format: “xx.xxxxx,yy.yyyyy” (where x – is latitude, y – longitude. Both values are in angles from 90.00000 to -90.0000.) For example, if a place has 42.989562 as its latitude and -71.463407 as its longitude, in the location field you should enter: 42.98956,-71.46341

You can use a free online service like [http://www.latlong.net/](http://www.latlong.net/) to find the latitude and longitude of a specific place.

- **Address**: textual information about a specific location is stored and displayed under this type.
- **Date**: this type is used to store and display dates
- **Time**: this type is used to store and display times
- **Boolean**: this type is used to store and display Boolean points.
- **Audio**: audio files fall under this type.
- **Video**: video files fall under this type.

I. If you would like any of the uploaded information to be sorted and/or searched for within the app you are building, check off the corresponding boxes.

### Fields:

<table>
<thead>
<tr>
<th>Column ID</th>
<th>Column Name</th>
<th>Type</th>
<th>Sort</th>
<th>Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name</td>
<td>text</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Menu</td>
<td>Menu</td>
<td>text</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Address</td>
<td>Address</td>
<td>text</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>PhoneNumber</td>
<td>PhoneNumber</td>
<td>text</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Logo</td>
<td>Logo</td>
<td>image</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Email</td>
<td>Email</td>
<td>text</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>HoursOfOperation</td>
<td>HoursOfOperation</td>
<td>text</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

Once you have saved the data source and configured the way the information from it will be displayed, in the Preview app you will see an Advanced Search and Sort buttons you will be able use to search/sort this information.
5.2.1.4 Sorting

In the list view, you can sort its items by clicking on (sort in ascending order) or (sort in descending order).
In our case, you can clearly see how the cupcake bakeries were sorted based on their name.

To enable sorting, after you have specified the way the information from the data source will be displayed, in the settings of the tab using this data source, in the Sorting Filter section, check off the Enable Filter box and specify what fields (name, address, hours of operations, etc.) should be sorted and how: a) always; b) be overridden when a user selects sort options; whether they should be sorted in c) ascending or d) descending order. Click OK. Please keep in mind that if you opt for always, the sort button won’t appear in the app, so its users won’t be able to sort any information from the list themselves.
5.2.1.5 Searching

This option allows finding specific information. For example, if we only needed to find cupcake bakeries in Manchester, NH, we could simply enter “Manchester” to find them on the map. In our case, only one cupcake bakery was found since we entered one cupcake bakery per town.
To enable searching, after you have specified the way the information from the data source will be displayed, in the settings of the tab using this data source, in the Search Settings section under Simple search settings, check off the box next to Enable simple search (search bar) in app. Then select the items specified in the data source (name, menu, address, hours of operation, etc.) that can be searchable within the app. Click OK.
J. Click *Save* and then, on the pop up window, click *OK.*
K. After you have saved the data source, you will see it either under App Data Sources or Shared Data Sources, depending on whether you have shared it or not. In our case, it is located under Shared Data Sources.
L. Now you can click Close.

Please note that you will be able to change the settings of this data source, edit, or even delete it at any time later on.

5.2.2 Using the Advanced List Tab/Button
Once you upload your spreadsheet to the Data Source Dialog, you can add an Advanced List tab/button and display data from the uploaded data sources in the app. The Advanced List feature also allows adding/editing/deleting, and emailing information from the data source.

In the screenshots below, you can see the whole process of adding new items into the Advanced List tab/button. The same data will go to the existing data source. To see these changes, you can download its development data using the Download Data button.
By using the User Management options and the Security section of the Advanced List tab/button, access to add/edit/delete operations can be limited. (See Chapter 4 for more information.)

5.2.3 Using the Universal Form Tab/Button
The Universal Form tab/button is also connected to data sources and stores entered data in an Excel spreadsheet, which users can always download from the Data Source Dialog.

To illustrate how this works, let’s take our Inspect Anything app and fill out a few fields, including the Notes field.
After all necessary fields have been filled out, hit Save and Send. Now you can see the filled out fields in both emailed PDF and Excel reports and the data sources. (It goes without saying that the report has also been saved on your mobile device and now can be accessed and edited/deleted at any time.)

Here is how this report looks in the emailed Excel report.
As for the data sources, the filled out information has been exported into several of them: notes have been sent to the *Notes* data source, photos – to the *Photos* data source, and the rest – to the *Inspect Anything* data source. Please see the screen shot below.

To view the exported information, select the intended data source and hit *Download*. In our case, we’ve chosen the *Notes* data source.
Here is how the *Notes* field looks in the downloaded Excel spread sheet.

![Excel sheet screenshot]

### 5.2.3.1 Adding Data from Universal Forms into Advanced Lists

In order to display the information that was previously filled out and saved in a Universal Form and enable your users to access, edit, and delete this data, you will need to bind your Universal Form and Advanced List by using the same data source. We’ll illustrate how this functionality works by creating an inspection app, where filled out reports can be viewed by clicking on the *Saved Reports* button, displayed in the List View, and then edited or deleted.

![App screen capture]
To implement the functionality above, please follow these steps:

1. Upload, configure, and save your data source.
2. Add a Universal Form tab/button.
3. Bind the Universal Form tab/button with the uploaded data source and check off the box next to Add to datasource. Configure the view mode and size of your Universal Form if it is a button. Click OK.
4. Drag and drop enough Table fields to enter data into them and bind each field with a corresponding value from the uploaded data source.
5. Drag and drop enough Text elements to name the Table fields and configure them as you wish.
Please make sure to leave the box next to *Use Data Source* unchecked in the text element settings. Only then will the required text show up. If you choose *Data Source* as the source, the data from your data source will appear.

6. Add one Advanced List tab/button and bind it with the same Data Source as you did in the Universal Form tab/button. (In our case, we opted for an Advanced List button and selected *Framed View* as its viewing mode.)
Watch video tutorial

Use this element to display data from your Data Sources to users. When an item is selected by the user from the list a detailed view is shown. Select the exact layout of the list view and detailed view you would like from the options (templates) below or customize them according to your needs. You can also set user permissions to edit and add data. Study the User Manual carefully before using DataSource/Data Connector. Read more in FAQ.
7. Select *Custom List View* for your Advanced List tab/button and hit *OK*.

8. Open your Custom List view and drag and drop as many text elements as required to show the field names and display information received via the Universal Form. Bind each element with a corresponding field from the data source. We have two elements: *Name* and *Date*. 
We will make this view clickable, so we have also added an Image element, changed its icon to an arrow, and made it constant to show that our users can click on it.
Here is how our Custom List View looks in the editor.

To make this Advanced List clickable so that another Universal Form opens once a user selects a specific saved report, drag a Universal Form button onto the screen and bind it to the same data source (*Inspect Anything*). This Universal Form should be identical to the Universal Form described above since it will display the previously filled out and saved information.
5.2.3.2 Making Universal Forms Inside Universal Forms

To be able to create Universal Forms inside of other Universal Forms, you need to use the Table element available under *Fields and Buttons*.

Once added onto the canvas, it will look as a usual field but with a green plus sign next to it. A double click on the “Plus” will open another blank page, where you can place more fields and create a whole new form inside.
When this Table field has been filled out, users can save it and again open another form by clicking on the plus sign. They can also delete and edit these fields at any time.
5.3 Copying a Data Source

If you have previously created a shared data source and would like to copy it into your current app, you can do so by selecting a data source you need under *Shared Data Sources* and clicking on *Copy Data Source.*
Then click *Ok* on the pop up window.

Data Source copied successfully

Now you can see the copied data source under *App Data Sources*. 
Please not that all the settings of the original data source have been preserved.

5.4 Editing a Data Source in an Unpublished App
You can always update an existing data source of an unpublished app by editing the original spread sheet, choosing the required data source in the list of the previously saved data sources, clicking on *Upload* or *Append*, and saving it in the Data Source Dialog.

5.5 Editing a Data Source in an Published App
Once you publish your app, all the uploaded data table changes status to “published”. You will not be able to change or add any new data fields of the published app this point. Snappii stores data for published apps in the production environment and unpublished apps in the development environment so that further app development can proceed without affecting data in published apps in production.
(However, Snappii’s developer can continue to test apps using our Preview app and perform Web2Phone copy builds. It will not damage the data in the published application in any way. )

You can still download the data from the published app by simply clicking on Download and exporting the data into a spreadsheet. You can also replace production data table with development data by hitting Transfer Development Data.

(If you don’t want the production app data to be lost, you need to download it using Download Data and add it to the development data (Append Data or Upload Data)). Only after this do you transfer it from the development environment to the production environment.

5.6 Copying a Data Connector
A data connector can be built using any web service’s restful API by simply creating an HML document or utilizing an existing document or feed that you might already have. For example, there is no code required to create a Sales Force data connector, which will enable you to pull information from your Sales Force database into your app and use it a number of different ways, such as displaying or updating the information.

Follow these steps to copy and configure a data connector. We are going to use Sales Force Leads under Data source templates as an example.

A. Click on Sales Force Leads. All the fields in the Data Source Dialog will instantly be populated with the fields and settings of this pre-built connector.
B. Click on *Copy Data Source* to add it to your app and then click *OK* on the pop up window.

Now a copy of this data connector has appeared under *App data sources*. 
C. Under **App data sources**, click on **Copy of Salesforce Leads**. Now you can rename this data connector if you wish.

D. **Web Service** will be set by default since this is a connector to a web service.

E. Next to **Source of Credentials for Consumer**, choose what type of credentials will be utilized in your app in this data connector. You can select **Not used** if the connector you are using does not require any sort of authorization. Opt for **App Creator’s Account** when you are building a data connector because it allows testing the connector. Finally, choose User’s account when your app is ready to go live with the data connector in your app. This option will cause a screen to pop up in the app and will ask your user to log in with his/her credentials. Please see the screenshot below for more information on the OAuth types.
Select when credentials for OAuth will be entered and by whom.

**App Creator’s Account** - App users do not need to enter their login and password to access the data. The app uses App Creator’s credentials. Select this option when testing apps in development environment. Also use this option when you want to use a shared set of credentials and when you don’t want the app to prompt users for credentials.

**User’s Account** - Users will be prompted by the app to enter username and password when accessing data from a web service. This option provides better security because each user action can be recorded and therefore should be the default option for apps in production environment.

If your web service doesn’t use OAuth, select "Not used".

(oAuth stands for an *Open Statement for Authorization* and determines what type of credentials will be utilized in the given data connector. You can get more info about OAuth [here](#).)

To learn more about OAuth and consumers, click on *Quick Study Guide* and download the guide.

F. To test your connector, you can click on *Test Select operation*.

G. Click on *Save* and *OK* on the two pop up windows.
Click on Close. Your data connector is now ready.

5.7 Creating a Data Connector

A. Web service is a universal mechanism that connects apps with on premise data sources inside organizations as well as with websites on the Internet. There are literally hundreds of thousands of really useful data sources available for anything from weather, file storage, videos, pictures, marketing services, etc. Our apps can retrieve data, show it and, if needed, update it.

To build a data connector, you need to create an XML file. To see an example of an XML file, click here or choose any of the data source templates and then click on Download Connector.

B. Once the XML file with your connector is ready, select Upload XML Connector.

C. Now name the connector, choose Web Service as its type, select the source or credentials, and match the column names with their types. Enable sorting and searching if necessary. See Copying a Data Connector and Building a New Data Source for more information on these operations.

D. Test your connector by clicking on Test Select operation and then hit the Save button.
E. Click on OK on the two pop up windows.

Your data connector is now ready.

5.8 Three Types of Relationships

The Manage Relationships section allows Snappii’s users to create multilevel lists using advanced lists and advanced form tabs and buttons.

These are the types of currently available relationships: One-to-One, One-to-Many and Many-toMany.
5.8.1 One-to-One

*The One-to-One Relationship* is used when one item offers only one option.

**Use Case:** listing authors and their books, in which each author corresponds to a single book option.

1. Create two spreadsheets: one with the authors and the other with their books. The ID field is key here as it indicates which book belongs to which author. For example, the book titled “Last Fall” belongs to the 1st author, who was born in 1900.

<table>
<thead>
<tr>
<th>name</th>
<th>bio</th>
<th>photo</th>
<th>id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author1</td>
<td>Born 1900</td>
<td><a href="http://sm">http://sm</a></td>
<td>1</td>
</tr>
<tr>
<td>Author2</td>
<td>Born 1888</td>
<td><a href="http://yok">http://yok</a></td>
<td>2</td>
</tr>
<tr>
<td>Author3</td>
<td>Born 1777</td>
<td><a href="http://ww">http://ww</a></td>
<td>3</td>
</tr>
<tr>
<td>Author4</td>
<td>Born 1900</td>
<td><a href="http://sm">http://sm</a></td>
<td>4</td>
</tr>
<tr>
<td>Author5</td>
<td>Born 1888</td>
<td><a href="http://yok">http://yok</a></td>
<td>5</td>
</tr>
<tr>
<td>Author6</td>
<td>Born 1777</td>
<td><a href="http://ww">http://ww</a></td>
<td>6</td>
</tr>
<tr>
<td>Author7</td>
<td>Born 1900</td>
<td><a href="http://sm">http://sm</a></td>
<td>7</td>
</tr>
</tbody>
</table>

2. When you have uploaded the spreadsheets, set the ID field type as “text” instead of “numeric”.
3. Name the first spreadsheet *Authors* and the other – *Books*. Save the data sources.

4. Click on the data source called *Authors* and then on *Manage relationships*. Choose “One to One” as a relationship type → choose *Books* for the related data source → select ID in both ID fields. (The ID fields must match the ID columns from the spreadsheets). Click on *Add*, and you’ll see the relationship. Hit *Close*.

4. Drag and drop the Advanced List tab and choose the data source called *Authors*.

In the settings of the Advanced List tab, you can:
- Specify the title of the Advanced List tab;
- Specify its icon;
- Select the view mode;
- Configure its security settings;
- Select the data source;
- Specify the title of the List and Detail Views;
- Configure the List and Detail Views (select a template or Custom view to display the data from the spreadsheet);
- Configure the sorting and searching options;
- Specify how the app users can share some data from the app; - Opt for ads if necessary.

5. After you have saved the Advanced List tab, drag and drop the elements you would like to be displayed in the Cell View and then in the Detail View. In our case, in the Cell View, we added two text documents, one for the authors’ names and the other – for their biographies, and one image element for the photo of each author.
6. We also made sure to connect the elements we dragged onto the canvas with corresponding columns of the specified data source. (In our case, it is *Authors.*) For example, to show the authors’ names, in the settings for the text element for the authors’ names, we chose *Data Source* next to *Source* and *Name* next to *Select Text Field.*
Similarly, to show the authors’ biographies, we chose *Data Source* next to *Source* and *Bio* next to *Select Text Field*; to display the authors’ photos, we chose *Data Source* next to *Source* and *Photo* next to *Select Text Field*.

Here is how our Cell View looks both in the editor and on a mobile device.
7. In the *Detail View*, we dragged and dropped 2 text documents (one for the name of each book and one for its price) and one image document (for the image of each book).

8. Here is when the Manage Relationships section comes into play. For each text element, next to *Source*, we chose *Data Source*, and next to *Select Text Field*, we chose *Book* and *Price* accordingly.
As for the image element, we kept it the same for each book, so next to Source, we chose *Constant*, and next to *Select Image Field – Photo*. 
Here is how our Detail View looks both in the editor and on a mobile device.
The screens below clearly illustrate that once a particular author is selected, the user can view the author’s book and all the information specified by the app builder.
5.8.2 One-to-Many

The One-to-Many Relationship is used when one item offers several options.

Use case. Let’s say we need to list several publishers, authors and books with their descriptions so that our app users can choose any of the listed publishers, see the list of authors, click on any of the listed authors, and see the books they offer.

To do so, we need to create 3 spreadsheets with the information on each publisher, author, and their books, and bind these spreadsheets inside the Snappii Editor.

1. Create 3 spreadsheets (Publishers, Authors, and Books) with the fields specified in each screen below:
2. Upload the spreadsheets into the DataSource dialog. Keep in mind that you need to choose the “text” type instead of “number” for ID and publisherID. Save each data source (spreadsheet).
3. Select the Publisher data source and then Manage Relations.

Add new relationship:

<table>
<thead>
<tr>
<th>Relationship Type</th>
<th>Related Data Source</th>
<th>Related ID Field</th>
<th>Related Label Field</th>
<th>Current ID Field</th>
<th>Current Label Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>One to Many</td>
<td>Authors list</td>
<td>publisherID</td>
<td>name</td>
<td>ID</td>
<td>Publisher</td>
</tr>
</tbody>
</table>

4. Choose One to Many as a relationship type → Authors List for the Related Data Source → publisherID for the Related ID field → Name for the Related Label Field → ID for the Current ID Field, and Publisher for the Current Label Field.

5. Click on the Add button, and you will see the newly added relationship.
Hit Close.

6. Select the Authors data source and then Manage Relations.

7. Choose One to Many as a relationship → Books for the Related Data Source → publisherID for the Related ID Field and Current ID Field, Book for the Related Label Field, and Name for the Current Label Field. Click on the Add button, and you will see the newly added relationship.
8. Close the Data Source dialog.

9. Drag and drop the Advanced List tab. Choose *Publishers* as the data source for this tab and specify the *List View* and *Detail View* by either using any of the templates or customizing the views. In our case, we will need the *Custom Detail View* to be able to add another Advanced List button on it.

10. Close this window. Now you’ll see the following screen.
11. Double click on the Edit Cell View to drag and configure the elements that will be displayed here from the Publisher spread sheet. First, we added one text document to display the publishers’ names, configured it in the way shown in the screenshot below, and chose Data Source next to Source and Publisher next to Select Text Field.
Next, we added one image document, configured it in the way shown in the screenshot below, and chose Data Source next to Source and Photo next to Select Text Field.
Here is how our Cell View looks both in the editor and on a mobile device.
12. Double click on the *Edit Detail View* to drag and configure the elements that will be displayed here from the *Authors* spread sheet. In our case, we added one Advanced List element.

13. In the pop-up window, select *Framed View* next to *Select View Mode*. 
14. Since this screen will display the list of authors from the Authors spreadsheet, choose List of Authors next to Select Data Source. Also, specify the List View and Detail View by either using any of the templates or customizing the views. In our case, we will need the Custom Detail View to be able to add another Advanced List button on it. Close the window, click twice on the List element, and you’ll see the following screen:

![List View Options](image)

15. Double click on the Edit Cell View to drag and configure the elements that will be displayed here from the List of Authors spread sheet. First, we added two text documents to display the authors’ names and their biographies, configured them in the way shown in the screenshots below, and chose Data Source next to Source and Name and Bio next to Select Text Field accordingly.
Next, we added one photo element to display the authors’ photos, configured them in the way shown in the screenshot below, and chose \textit{Data Source} next to \textit{Source} and \textit{Photo} next to \textit{Select Text Field}.

Here is how our Cell View looks both in the editor and on a mobile device.
16. Double click on the *Edit Detail View* to drag and configure the elements that will be displayed here from the *Books* spread sheet. In our case, we added one Advanced List element.
17. In the pop-up window, select *Framed View* next to *Select View Mode*.

18. Since this screen will display the list of books from the *Books* spreadsheet, choose *List of Books* next to *Select Data Source*. Also, specify the *List View* and *Detail View* by either using any of the templates or customizing the views. Close the window, click twice on the *List* element, and you’ll see the following screen:

19. Double click on *Edit Cell View* and place there any elements you wish do display any data from the *Books* spreadsheet. In our case, we added two text elements and one image
element and configured them accordingly (*Book* next to *Select Text Field* for the element displaying the book names, *Price* – for the element displaying the book prices, and *Photo* – for the book photos.)

Here is how our Cell View looks both in the editor and on a mobile device.

![Cell View](image)

20. Double click on *Edit Detail View* and place there any elements you wish to display any data from the *Books* spreadsheet. We added three text elements (for the books’ names, prices, and ISBN numbers) and one image element (for the books’ photos).

Here is how our Detail View looks both in the editor and on a mobile device.
Your Advanced List tab with one-to-many relationships is now ready.

The screenshots below illustrate in what sequence this tab works.
5.8.3 Many-To-Many

*The Many-to-Many Relationship* is used when one item offers several options, while some of these options overlap.

**Use case.** Let’s say we need to list several albums and individual musicians to illustrate that some of these albums were recorded by several of these musicians.
To create such an app, we need to create 2 spreadsheets with the information on each album and artist and then one binding spreadsheet to bind these spreadsheets inside the Snappii Editor.

1. Create 3 spreadsheets (Albums, Artists, and Binding document) with the fields specified in each screen below:

### Albums

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Albums</td>
<td>Images</td>
<td>Description</td>
<td>currentid</td>
<td></td>
</tr>
<tr>
<td>Circumstances</td>
<td><a href="https://s.1">https://s.1</a></td>
<td>The album was released</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Colored Life</td>
<td><a href="https://m.1">https://m.1</a></td>
<td>The album was released</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Rocking Around the Clock</td>
<td><a href="https://s.1">https://s.1</a></td>
<td>The album was released</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Oh, Marilyn</td>
<td><a href="http://w.1">http://w.1</a></td>
<td>The album was released</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Drum Till You Drop</td>
<td><a href="http://w.1">http://w.1</a></td>
<td>The album was released</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>It's Surreal</td>
<td><a href="https://s.1">https://s.1</a></td>
<td>The album was released</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Butterflies</td>
<td><a href="http://w.1">http://w.1</a></td>
<td>The album was released</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Merry Christmas</td>
<td><a href="https://s.1">https://s.1</a></td>
<td>The album was released</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Oh, Dali</td>
<td><a href="http://u.1">http://u.1</a></td>
<td>The album was released</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>Inspiration</td>
<td><a href="https://s.1">https://s.1</a></td>
<td>The album was released</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

### Artists

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artist</td>
<td>Bio</td>
<td>Photo</td>
<td>relatedid</td>
<td></td>
</tr>
<tr>
<td>Aaron Davis</td>
<td>Aaron Davis was born in Jamaica in 1974.</td>
<td><a href="http://images.clipartpanda.com/free-mus">http://images.clipartpanda.com/free-mus</a></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Ariel Belanger</td>
<td>Ariel Belanger was born in France in 1991.</td>
<td><a href="http://cliparts.co/cliparts/pkK/pkKXX">http://cliparts.co/cliparts/pkK/pkKXX</a></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Jose Lopez</td>
<td>Jose Lopez was born in Mexico in 1985.</td>
<td><a href="http://images.clipartpanda.com/musical-c">http://images.clipartpanda.com/musical-c</a></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Braden MacAuliffe</td>
<td>Braden MacAuliffe was born in Ireland in 1990.</td>
<td><a href="http://images.clipartpanda.com/handwrit">http://images.clipartpanda.com/handwrit</a></td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Christine DelDeo</td>
<td>Christine DelDeo was born in the USA in 1988.</td>
<td><a href="http://www.computerclipart.com/comput">http://www.computerclipart.com/comput</a></td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Jane Lumpu</td>
<td>Jane Lumpu was born in Canada in 1993.</td>
<td><a href="http://clipartsign.com/upload/2015/12/05">http://clipartsign.com/upload/2015/12/05</a></td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

10
(In the spreadsheet above, you can see how several artists participated in the recording of a single album and any given artist released several albums. Thus, for instance, the 9th and 10th artists recorded album number 5, while the 5th, 6th, and 7th artists recorded album number 3. At the same time, the 10th artist recorded album number 6 together with the 4th, 8th, and 7th artists, and the 6th artist recorded albums number 10, 9, and 3)
2. Upload the spreadsheets into the DataSource dialog. Keep in mind that you need to choose the “text” type instead of “number” for currentid and relatedid. Save each data source (spreadsheet).

3. Select the Albums data source and then hit Manage Relations.

5. Click on *Upload bindings*.

### Junction Table

This Excel file contains bindings between two related Data Sources. It must contain only two rows: "currentId" and "relatedId". In those rows you need to write all combinations of ids from two Data Sources.

Example:

<table>
<thead>
<tr>
<th>currentId</th>
<th>relatedId</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

[Upload Junction] [Close]

6. In the pop-up window, hit *Upload Junction* and upload the spreadsheet *Binding document*. Hit *OK* and then *Add*. Now you should see the newly created relationship.

7. Hit *Close*, *Save*, and close the data source dialog. (Please keep in mind that you can always rewrite the relationships by clicking on *Rewrite Table* and uploading the spreadsheet with updated relationships.)
8. Drag and drop the Advanced List tab. Choose Albums as the data source for this tab and specify the List View and Detail View by either using any of the templates or customizing the views. In our case, we will need the Custom Detail View to be able to add another Advanced List button on it.

9. Close this window. Now you’ll see the following screen. (If you have opted for the Cell View with a template, than the Cell View will be inactive.)
10. Since we have opted for custom Cell View, we can now customize it. Double click on the Edit Cell View to drag and configure the elements that will be displayed here from the Albums spread sheet. First, we added one text document to display the albums’ names, configured it in the way shown in the screenshot below, and chose Data Source next to Source and Album next to Select Text Field.
11. Next, we added one image document, configured it in the way shown in the screenshot below, and chose **Data Source** next to **Source** and **Photo** next to **Select Text Field**.
12. Then we added one more text document for the albums’ description, configured it in the way shown in the screenshot below, and chose Data Source next to Source and Description next to Select Text Field.
Here is how our Cell View looks both in the editor and on a mobile device.
13. Click on the *Edit Detail View* to drag and configure the elements that will be displayed here from the *Artists* spreadsheet. In our case, we added one Advanced List element.

14. In the pop-up window, select *Framed View* next to *Select View Mode*. 
15. Since this screen will display the list of artists from the Artists spreadsheet, choose Artists next to Select Data Source. Also, specify the List View and Detail View by either using any of the templates or customizing the views. Close the window, click twice on the List element, and you’ll see the following screen:

(In our case, Edit Detail View is inactive because we have opted out of the Detail View.)

16. Click on the Edit Cell View to drag and configure the elements that will be displayed here from the Artists spread sheet. First, we added two text documents to display the artists’ names and their biographies, configured them in the way shown in the screenshots below, and chose Data Source next to Source and Artist and Bio next to Select Text Field accordingly.
17. Next, we added one photo element to display the artists’ photos, configured them in the way shown in the screenshot below, and chose Data Source next to Source and Photo next to Select Text Field.
Here is how our Cell View looks both in the editor and on a mobile device.

Our Advanced List tab with the Many to Many relationship is now ready.
The screenshots below illustrate in what sequence this tab works.

5.9 Shared Data Sources

Shared Datasources enable Snappii’s users to create native mobile role-based apps that use share datasources and, hence, provide the same data to a series of apps.

Let’s review a possible use case.

You would like to create a work management app but prefer to split it into two separate apps, one for employees, and the other for their manager. Thanks to the shared datasource option, you can do so. Both apps will use the same data source to enable the supervisor to assign tasks and check job statuses, while the employees will be able to review the assigned tasks and edit them when the tasks are completed.

Similar scenarios can be utilized for a majority of apps that have assignment features such as apps for logistics, dispatchers, drivers, etc.
To enable this feature, open the Datasource Dialog, select the data source you want to share and put a check next to Share. Then save the settings. Now this data source will appear in the Shared Data Sources dialog.

5.10 Runtime Datasources
Using runtime data sources can be a great option for big companies. This feature allows sharing data sources among specific users but not all of them. When registering, users can be given a special registration form, where they will need to enter the company identifier. If users work in the same company, they should enter the same value for this identifier. It will allow them to work with the same set of data.

Use Case: Users from a given company register in the app and share the same companyId.
When a user logs in, he gets to the main tab. He has the Share button, by clicking on which he can send the companyId to the other users:

A user with another companyId will have another version of data:
5.11 Datasource triggers
Datasource triggers allow you to send push notifications to the app users based on the changes made in the datasources. To set up the trigger, please go to your app dashboard.

Now open the Data source triggers tab
As you can see there are no triggers yet. To create a new one, click Create button. You will see the dialog:

**Data source notification**

Let’s see the available settings. On the upper right corner you need to select the name of the datasource you will work with, then you can insert the notification name, and finally the notification text. It will look the following way on the device:
To the left you will see the app icon (the default one), the notification name and its text.

You can send a notification based on one of four actions:

Data selection

Add

Update

Remove

If these actions are not enough, you can create a custom action. For example, you want to receive a notification each time a user with the name Mike creates an account. To have it work click on “Add” option – since this is an adding of the new entry – and under the additional sending conditions specify Name equals Mike. This way, you will receive a notification each time a user with the name Mike is created.
Then you need to select notification recipients. You can group your app users via the Dashboard and send a notification to that group or you can filter the users by email, name, ID, device ID or a custom filter.

Now select the search type:

You will choose if the data in the field Equals to or doesn’t equal to a value, is an empty value, not empty value, contains or doesn’t contain a value. We will consider some examples below.

And finally, you will choose a value to compare to:

Here you can select any value from the datasource or create a custom value.
Let’s consider a few examples. For example, you have a small company with all the staff having their company position in the email address Mike-worker@myCompany.com, Joshmanager@myCompany.com, etc. So you want to send a weekly job plan to the management. You will need to set up the following parameters:

- User’s email he is logged in under in the app must contain the word “manager”. The “contains” parameter will do a search in the line and even inside one word, while “=” parameter will search for the word “manager” alone. If you save these settings, then the employees who have a word “manager” in their email addresses will receive the data meant for them only.

You can have several conditions:

Let’s consider the parameters we have just entered and put them all here:

User.email = Profile.Email

User.name Is empty

User.Device Id not contains 132456789
User email he uses to login to the app is present in the Profile datasource and equals to its value. In other words, if after registration the user has created a profile, he will receive a notification unless there are other, more detailed conditions.

User name is not set up.

deviceID value doesn’t contain “132456789”.

This way, user filter has to address these 3 parameters. This option gives you a great flexibility in setting up different parameters. You can send notifications to user groups or filter users by custom parameters. These parameters will be checked each time a new notification is scheduled.

We have 2 last parameters to review:

Notification URL (optionally specify a web link for additional details)

Image URL (optionally specify or upload an image to be used in the notification)

Here’s how these 2 fields will look on the device:

The image is at the top, then the notification text and a link, if it is needed.
5.12 Preloading of data sources
This option will allow you to preload data for one or more data sources as long as you have stable Internet connection and work with the actual data even if the Internet connection is lost. This option is useful in case your work is associated with transportation or warehousing where the Internet connection may not be stable. To load predefined data, add a new element and in the settings choose the data sources you wish to load

![Image of data preload settings]

- Label: Data Preload
- Icon: Change icon
- Height: 45
- Width: 45
- Use rounded corner
- Font: Default
- Button style: Icon
- Select Data Source: weather, users
- Viewing Condition: users
- Data Sources: weather, users, chat, location
Chapter 6. How to Use Tabs
6.1 New Tab
The New tab, formerly known as the Custom tab, provides you with the most flexibility. Its functionality is not as limited as the functionality of the Map, List, or Form tabs, designed to perform specific functions. On the New tab, you can create a totally custom page by adding and configuring any buttons and elements of our editor.

Click here to watch a video about the New tab.

Here are a few examples of how the New tab was utilized by our app builders.
6.2 ListTab

The List tab works very similarly to the List button. For more information, please click here.

6.3 Map

The Map tab, formerly known as the Advanced Map tab, allows app users to view a number of organizations on a map. For instance, you would like to display all the locations of your car dealership business together with the contact information, available merchandise, price lists, and directions. Map enables you to easily do so. Let us look at another example. The Chamber of Commerce in your state needs to build an app that would allow the residents and guests of the state to easily access local restaurants, stores, entertainment centers, etc. Our Map functionality is a great solution in this case as well.
How is the Map tab different from the Nearby tab? First of all, it utilizes data sources and requires that one enter the latitude and longitude of each location specified in the data source.

The Map tab also allows customizing the annotation, detail, and cell views, thereby giving you freedom to make these screens look the way you want. You can also add the Call, E-mail, Send Text, Directions, and many other buttons, thus enabling your app users to contact you right from the app. Moreover, the Map functionality lets the app users add new organizations/locations to the list or edit the existing ones right within the app. However, it is up to the app owner to enable or disable these features in the editor. The Map tab also lets the app users sort the listed organizations or even search for specific ones. Last but not least, our Map functionality allows pulling information from external servers via the so-called data connectors.

Before we discuss in more detail how to work with the Map tab, please see the annotation, detail, and cell views both in their standard and customized forms.

**Locations Shown as Pins**

**Locations Shown as Logos**
Standard Annotation View

Customized Annotation View

Standard Detail View

Customized Detail View
As the screens above illustrate, the annotation view is what a user sees when he/she taps on a pin. When the user clicks on the pop up window of the annotation view, more detailed information
appears in the so-called “detail view”. The cell view is the display of all the available organizations/locations as a list.

Here is a step-by-step description on how to utilize the Map tab.

1. Create the data source/data connector from which the app will pull its information, upload, configure, and save it in the data source dialog. (Please see Data Sources and Data Connectors for more details.)

In our case, to build an app displaying a list of cupcake bakeries, we have created and uploaded in the data source dialog section an Excel spreadsheet with the following columns.

- Name
- Menu
- Address
- PhoneNumber
- Logo
- Email
- HoursOfOperation
- Location

(We have split the screenshot of our Excel file in two parts so that you can see it better.)
219

(In the data source dialog, we saved this file as *Cupcakes of NH.*)

2. Once your data source/data connector has been built, configured, and saved, close the data source dialog, select the Map tab and drag it onto the canvas.
View

3. In the pop up window, change the title, the name of the tab, and the font and icon, if necessary.
In the Security section, check off the checkbox if you would like to enable the app users to add, delete, or update the information accessible in the app. Specify what user types can perform these operations within the app itself by clicking on *Add Operation Permission*, *Delete Operation Permission*, and *Update Operation Permission* one at a time and unchecking the corresponding checkboxes.
Map View

5. In the Map View section, select the data source/data connector the tab will use. In our case, it is *Cupcakes of NH*. 
6. Next to Select Location Field, in the drop-down menu, select the column that contains the latitudes and longitudes of the organizations/locations specified in the data source/data connector.

7. Rename the annotation, detail, and cell views if necessary.
8. Now determine if the organizations/locations will be displayed as pins or differently. If you choose the pin option, just leave the default settings.

If not, then check off the box next to Use different map pins icon, select the data source option in the drop-down menu next to Map pin image source, and select the image column from your data source/data connector. In our case, it is the logo column.

9. If you wish, change the message that would be displayed when a search returns no results. In our case, it is “No Cupcake bakeries have been found”.

10. Click change next to Annotation View template and select any of the three templates or opt for Custom View.
Here is how each of these options works.

A. The template with an image, header, and details

When you select this template, three drop-down menus appear on the right. Under Header and Details, you can select any of the columns that you categorized as text in the data source dialog, whereas under Image, you can choose any of the columns categorized as image.
In our case, we chose *Name* for the header, *Logo* for the image field, and *Address* for the details.

Here’s what our annotation view with this template looks like on the map and by itself.

**On the Map**

![Map View](image)

**By Itself**

![By Itself View](image)

**B. The template with a header only**
When you select this template, one drop-down menu appears on the right. Here again you can choose any of the columns categorized as text in the data source dialog. In our case, we'll simply opt for Name (the names of the bakeries). Here's what our annotation view with this template looks like on the map and by itself.

**On the Map**

![Map view of Cupcake Bakeries](image)

**By Itself**

Lakes Region Cupcakes

C. The template with a header and an image.

When you select this template, two drop-down menus appear on the right.

In our case, we'll select Name for the header and Logo for the image field.
Here’s what our annotation view with this template looks like on the map and by itself.

**On the Map**

![Map View Image]

**By Itself**

![Annotation View Image]

**D. Custom View**

a) When you select *Custom View*, two fields pop up on the right. These are the dimensions for the annotation view window. Specify the height and width and click *OK.*
You will need to further configure this view but only after the other two views have been selected because the editor won’t let you save the settings and close the pop up window of the tab until all the views have been specified.

b) After all the three views have been specified, click on the Map tab and then on *Annotation Cell View*.

c) In this inner square, you can add any buttons or elements. In our case, we’ll add 2 Text elements with the name and address of each cupcake bakery and one Image element with the logo of each bakery. Here are the steps to add and configure these three elements:
Select the Text element under *Elements and buttons* and drag it onto the canvas.

In the pop up window, enter the name of the element. This is important when the source of the element is set as *constant*. However, when you pull the information from the data source/data connector, you can leave the original message.

Select the font size, its alignment, color, and font. Specify the width of this text element.

Next to *Source*, choose between *Data Source* and *Constant*. In our case, we selected *Data Source*.

Next to *Text Field*, select the column from the data source from which the information will be taken for this text element. In our case, it is *Name* (the name of each cupcake bakery).
Click OK. The first element is now ready.
➢ Now select the Image element under *Elements and buttons* and drag it onto the canvas.

➢ In the pop up window, change the image if you are planning to use the same image for all the organizations/locations. (In this case, the source of your image needs to be set as *constant*.)

➢ Configure the dimensions of the image.

➢ Next to *Source*, select *Constant* or *Data Source*. In our case, it is *Data Source* because our app will pull the information from the data source.

➢ Next to *Select Image field*, choose the image column from your data source/data connector that will provide the images for this image element.
➢ Click OK. The image element is now ready.
Drag one more Text element onto the canvas and configure it in the same way described above except for Select Text Field. There choose a different column to pull the information from. In our case, we selected Address (the address of each cupcake bakery). Click OK. Align the elements if necessary.

Here is what your annotation view looks like in the app when you click on any of the pins.

11. Click change next to Detail View template and select any of the four templates, opt for Custom View, or simply opt out of the detail view.
Here is how these options work.

A. The template with an image, header, and details
When you select this template, three drop-down menus appear on the right. Under *Header* and *Details*, you can select any of the columns that you categorized as *text* in the data source dialog, whereas under *Image*, you can choose any of the columns categorized as *image*.

In our case, we chose *Name* for the header, *Logo* for the image field, and *HoursOfOperation* for the details.

Here’s what our detail view with this template looks like in the app.
B. The template with a header, image, and two fields for details

When you select this template, four drop-down menus appear on the right. Under **Header** and **Details**, you can select any of the columns that you categorized as *text* in the data source dialog, whereas under **Image**, you can choose any of the columns categorized as *image*.

In our case, we chose **Name** for the header, **Logo** for the image field, **HoursOfOperation** for the first field for details, and **PhoneNumber** for the second field for details.

Here’s what our detail view with this template looks like in the app.
C. The template with a header and one field for details

When you select this template, two drop-down menus appear on the right. Under \textit{Header} and \textit{Details}, you can select any of the columns that you categorized as \textit{text} in the data source dialog.

In our case, we chose \textit{Name} for the header and \textit{HoursOfOperation} for the field with details.

Here’s what our detail view with this template looks like in the app.
D. The template with a header, image, and one field for details

When you select this template, three drop-down menus appear on the right. Under *Header* and *Details*, you can select any of the columns that you categorized as *text* in the data source dialog, whereas under *Image*, you can choose any of the columns categorized as *image*.

In our case, we chose *Name* for the header, *Logo* for the image field, and *HoursOfOperation* for the details.

Here’s what our detail view with this template looks like in the app.
E. Custom View

a) When you select *Custom View*, you will be able to further configure this view only after the other two views have been selected.

b) After all the three views have been specified, click on the Map tab and then on Edit *Detail View*. 

![Custom View Image](image-url)
Here you can add any buttons or elements. In our case, we’ll add 3 Text elements with the name, address, and hours of operation of each cupcake bakery, one Web element with the menu of each bakery, and the Call, Email, and Directions buttons. Here are the steps to add and configure all these elements.

- Select the Text element under *Elements and buttons* and drag it onto the canvas.
- In the pop up window, enter the name of the element. This is important when the source of the element is set as *constant*. However, when you pull the information from the data source/data connector, you can leave the original message.
- Select the font size, its alignment, color, and font. Specify the width of this text element.
- Next to *Source*, choose between *Data Source* and *Constant*. In our case, we selected *Data Source*.
- Next to *Text Field*, select the column from the data source from which the information will be taken for this text element. In our case, it is *Name* (the name of each cupcake bakery).
 Click OK and align this text element if necessary. The first element is now ready.
➢ Now select the Web element under *Elements and buttons* and drag it onto the canvas.

➢ Configure its icon, dimensions, choose its source (in our case, it is *Data Source*) and the column with URLs. We selected the Menu column since it contains the URLs to the cupcake bakeries’ menus.
➢ Click OK and align the Web element if necessary. Now the element is ready, and the required URL opens once you click on the image in the app.
To add and configure the text field with the hours of operation, select and drag the Text element onto the canvas and then configure it similarly to the Name text field. However, choose \textit{HoursOfOperation} next to Select Text Field.

- Click OK and align this element if necessary.
To add and configure the text field with the address of each cupcake bakery, select and drag the Text element onto the canvas and then configure it similarly to the Name and Hours of Operation text fields. However, choose Address next to Select Text Field.
➢ Click OK and align this element if necessary.
To enable your app users to call the featured organizations right from the detail view, you can add the Call button by selecting it under *Elements and buttons* and dragging it onto the canvas.

Configure its icon, dimensions, choose its source (in our case, it is *Data Source*) and the column *PhoneNumber*.
Click OK and align this element if necessary.

To allow your app users to email the featured organizations right from the detail view, you can add the Email button by selecting it under Elements and buttons and dragging it onto the canvas.
Configure the icon, dimensions, choose its source (in our case, it is *Data Source*) and *Email* for the Email field.
Click OK and align this element if necessary.
➢ To enable your app users to get the directions to the featured organizations right from the detail view, you can add the Directions button by selecting it under Elements and buttons and dragging it onto the canvas.

➢ Configure the icon, dimensions, and select the Location column for the Location field. Click Ok. Align the button if necessary.
The customized detail view is now ready.

12. Click *change* next to *Cell View template* and select any of the three templates or opt for *Custom View.*
Here is how these options work.

A. The template with an image, header, and details
When you select this template, three drop-down menus appear on the right. Under Header and Details, you can select any of the columns that you categorized as text in the data source dialog, whereas under Image, you can choose any of the columns categorized as image.

In our case, we chose Name for the header, Logo for the image field, and Address for the details.

Here’s what our cell view with this template looks like in the app.

B. The template with an image and a header
When you select this template, two drop-down menus appear on the right. Under Header, you can select any of the columns that you categorized as text in the data source dialog, whereas under Image, you can choose any of the columns categorized as image.

In our case, we chose Name for the header and Logo for the image field.

Here’s what our cell view with this template looks like in the app.
C. The template with a header only

When you select this template, only one drop-down menu appears on the right. Under *Header*, you can select any of the columns that you categorized as *text* in the data source dialog. In our case, we chose *Name* for the header.

Here’s what our cell view with this template looks like in the app.
D. Custom View

a) When you select *Custom View*, one field pops up on the right. This is the height for the cell view window. Specify it and click *OK*. When you select *Custom View*, you will be able to further configure this view only after the other two views have been selected.

b) After all the three views have been specified, click on the Advance Map tab and then on *Edit Detail View*. 

![Custom View Image]
c) In this area above the horizontal line, you can add any buttons or elements. In our case, we’ll add 2 Text elements with the name and address of each cupcake bakery and one call button. Here are the steps to add and configure these three elements:

- Select the Text element under *Elements and buttons* and drag it onto the canvas.
- Configure the element and make sure to select *Data Source* as the source and *Name* next to *Select Text Field*. We recommend that you make the text element wide enough for the names to fit in in the app itself.
- Add one more Text element onto the canvas and configure it. Make sure to select *Data Source* as the source and *Address* next to *Select Text Field*. We recommend that you make the text element wide enough for the addresses to fit in in the app itself.
- Add the Call button and configure it. Make sure to select *Data Source* as the source and *PhoneNumber* next to *Select Text Field*.

Our custom cell view is now ready.

Here’s what our cell view looks like in the editor and in the app.

**Sorting Settings/Filter**

14. In the Sorting Filter section, if you would like to enable your users to sort query results, such as names, addresses, phone numbers, etc., in the cell view either in ascending or descending order, check off the box next to *Enable Filter.*
Select *Apply Always* to display query results in a certain order by default. For instance, on the first screenshot below, the names of the cupcake bakeries are listed in the alphabetical order ascendingly.
On the second screenshot, the cupcake bakeries are listed based on their location (from the closest to the user to the farthest.).
To enable your users to specify sorting preferences themselves, select *Be overridden when user selects search or sort options from the app.*
On the first screen, you can see that the names of the cupcake bakeries are listed alphabetically in ascending order. However, any user can change it by clicking on the Sort button and specifying how any of the given options (in our case, these are *Name, Address, and Hours of Operation*) should be displayed.

Just a reminder that you can enable the sorting of any number of the elements/fields from the data source/data connector in the data source dialog by checking off the corresponding boxes under *Sort.*
Search Settings

15. To enable your users to search for specific information from the data source, check off the box next to *Enable simple search (search bar) in app*. Choose between

- =
- not =
- contains
- doesn’t contain

For example, to enable our users to search among the listed bakeries based on their city/street number/ or street name – any information from the data source/data connector in the Address column-, we need to select *Contains* and then *Address* from the drop-down menu.

In the app itself, we can enter any town and see if there is a cupcake bakery in it.
Share Settings

16. If you would like to enable your users to email or print out some specific information from the app or the server, in the Share Settings, do the following:
a) put a checkmark next to *Show share button*

b) if you want all emails to be sent to a specific email address, enter it here; otherwise, your users will need to specify their email recipients every time they share information within this app;

c) enter the subject of the email;

d) enter a message, if you wish;

e) change the name of the Share button, if necessary:

f) specify the message that you users see upon successful email submission;

g) select whether to email data in the PDF, Excel (spreadsheet), or Word format;

h) enter the name of the shared file;

i) specify whether your users will be able to share data from the app or server;

j) click on *Data in Report* to uncheck those boxes that you don’t want to be shared in the report;
k) if you have selected *app* in the previous question, you can enable either emailing or printing of the data;

l) put a checkmark if you would like to include the user’s information in the report

m) choose the default template or click on *Show PDF templates dialog* to create your own template of the report. In the pop up window, click *New*, name the template, upload a header and a footer, if necessary, choose the title size, color, the layout of the fields, etc. Click *Save* and then *Close*. Now next to *PDF report template*, from the drop-down menu, choose the template you need. In our case, it is *New Hampshire*. 
Default Template

Cupcakes of NH Report

User name: Anonymous
User email: Anonymous

Name: Cupcakes 101
Address: 132 Bedford Center Road, Bedford, NH 3110
Phone Number: 603-488-5962
Logo:

Email: cupcakes101@gmail.com
Hours of Operation: Tuesdays-Fridays: 11:00 AM – 6:00 PM
Saturdays: 11:00 AM – 3:00 PM
Closed on Mondays and Sundays

Custom Template
That is what the report looks like as an attachment.

Ads

17. If you would like to add ads in this tab of your app, check off the box next to *Show fixed banner ad*, enter the ad unit ID for the iOS and/or Android version of your app, and choose the position of the ad from the drop-down menu.

Please click [here](#) to learn about how to get unit IDs.
6.4 Calculator Tab

The Calculator tab works very similarly to the Calculator button. For more details, please click here.
6.5 ChartTab

The Chart tab/button allows presenting information graphically in a variety of chart forms: pies, column charts, line charts, bar charts, and scatter charts.
In our editor, 5 data types are supported by the Chart tab/button: textual, numeric, date, time, and datetime.

To illustrate how to work with the Chart tab/button, let us create an Excel spreadsheet with 5 fields representing each of these 5 types of data:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>text</td>
<td>num</td>
<td>date</td>
<td>time</td>
<td>datetime</td>
<td></td>
</tr>
</tbody>
</table>

Then upload this spreadsheet in the Data Source Dialog, under Type, choose the right data type for each field, name the data source, and save it.
Now add the Form button onto the canvas, connect it to our data source, and add a field for each of the fields in the data source: *Text, Number, Date, Time, and Datetime*. In Select Field, select the corresponding field from the data source for each of these 5 fields in the digital form.

Here are the settings of each of them.

**The Text Field**
The Number Field

The Date Field
The Time Field

Name: Time
Format: Time
Height: 35
Width: 270
Web Parameter name: DateTimeField141
Select Field: Time
Order: 4
Next order: 4
Font: Default
Include Field in Report
Sharing enabled
Placeholder: Time
Save input history
Column width (for Excel report): Auto fit
The Dateime Field

Here is the digital form we have created.
Now let us open the app through the Preview app and add a few entries. Next, download this data source in the Data Source Dialog to see the format in which they had been saved.

These are our sample entries.

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>text</td>
<td>num</td>
<td>date</td>
<td>time</td>
<td>datetime</td>
</tr>
<tr>
<td>2</td>
<td>School</td>
<td>140</td>
<td>2016-05-08T00:00:00.000</td>
<td>00:00-01-01T10:28:00.000</td>
<td>2016-05-03T13:28:00.000</td>
</tr>
<tr>
<td>3</td>
<td>Business</td>
<td>31</td>
<td>2016-05-12T00:00:00.000</td>
<td>00:01-01T12:35:00.000</td>
<td>2016-05-15T12:35:00.000</td>
</tr>
<tr>
<td>4</td>
<td>Mike</td>
<td>311</td>
<td>2016-05-23T00:00:00.000</td>
<td>00:01-01T13:30:00.000</td>
<td>2016-05-16T13:30:00.000</td>
</tr>
<tr>
<td>5</td>
<td>Tree</td>
<td>200</td>
<td>2016-05-23T00:00:00.000</td>
<td>00:01-01T03:16:00.000</td>
<td>2016-05-16T13:16:00.000</td>
</tr>
</tbody>
</table>

All these manipulations were necessary to simplify the process of entering data in the data source. As you can see, the data, time, and datetime formats are specific and might be a bit tedious to enter. That is the reason why we first created an empty spreadsheet, added the Form button, and then added all this information into the data source. When you are working with the Chart tab/button, you can start by simply creating an Excel spreadsheet and entering all the data you need manually, not through the app.

Now it is time to add the Chart tab or Chart button wherever you would like and open its settings.

### 6.5.1 View

- **Title**: Chart
- **Tab Name**: Chart
- **Icon**: ![Change icon](change_icon.png)
- **Font**: Default
- **Chart Title**: Chart
- **Empty dataset message**: Dataset contains no items to show
- **Sharing enabled**: [ ]

The quick brown fox jumps over the lazy dog.
As usual, the View section allows you to configure the way the given tab/button looks. Here you can specify its title, name, icon, font, chart title, and the message your app users will receive every time no saved information has been found. You can also allow your users to share this tab/button by putting a check mark next to Sharing enabled.

6.5.2 Chart View

The Chart View section allows you to configure the way your chart will look. On the left, you can choose any of the supported chart types and select colors for the axes, grid, and legend. On the right, you can see a sample of the chart type you have selected.

Here is how our chart looks with the chart type and colors we have selected.
6.5.3 Data Sources
In the Data Source section, you will also be able to select your point type (none, circle, square, and rhombus), line type (line, dotted, dashed), color for the chart line, and legend title. Naturally, here you will also need to select the data source your Chart tab/button will be connected to and the corresponding fields for the X-axis and the Y-axis from the data source.
The Searching Filter allows you to determine what information will be displayed and what will be hidden.

Please keep in mind that every time you change your chart type, you will have to select your data source and fields again.

### 6.5.4 Axes

This section allows you to configure the x and y axes by specifying data type for each axis and giving each one a title.

As you can see, the y-axis is always numeric, while the x-axis supports all the 5 types of data: text, number, date, time, and datetime.

The axis shifting option cuts off the y-axis in the specified range, thus showing only a portion of the chart. Thus, for instance, if we set the start value at 10 and the end value at 80, we will see the following:
The Rates content allows you to further configure what data will be displayed by the chart.
You can use the following chart parameters in rates:

- `<chart.legend>` - title in legend
- `<chart.X>` - value on X-axis
- `<chart.Y>` - value on Y-axis
- `<chart.percent>` - percentage parity of Pie chart segments to each other.
- `<chart.count>` - number of identical elements in the field; for Pie and Column charts only.

For example, you can use the following formula:

```
IN_TEXT( <chart.X> ) + " : " + IN_TEXT( <chart.Y> ) + " : " + <chart.legend>
```

Data will be displayed in the following way on your device: 04.16.2015: 53 Visitors

For instance, the `<chart.Y>` formula will show our numeric data, linked to the y-axis, at the chart points.
By the same token, the `<chart.X>` formula will display our textual data, linked to the x-axis, at the chart points.

If we opt for the `<chart.legend>` formula, each chart point will show the name of our legend.
As for the `<chart.percent>` formula, it works only with pie charts. Therefore, we will need to choose this chart type to see how this formula works.
First, in the Colors section, let us select a color for each text field.

![Colors section]

Then we should configure the Axes settings in the following way:

![Axes section]

Next, in the Data Source section, we need to select our data source and the corresponding fields for the axes.

![Data Sources section]

Finally, we can see the following screen and that the <chart.percent> shows the ratio of the given elements.
Just as a reminder, these are the values of the text and number fields from our data source:

School  140
Business 31
Mike     311
Tree     200

Last but not least, to show the value of both x-axis and y-axis, we can utilize the following formula:

\[ \text{IN\_TEXT}(\text{chart.X}) + \text{" ": } \text{IN\_TEXT}(\text{chart.Y}) \]
Here you can see that now both the textual and numeric information gets displayed.

6.5.5 Update Chart

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Refresh Chart</strong></td>
<td>Manual</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td><strong>Refresh interval</strong></td>
<td>Manual</td>
<td>Auto</td>
<td>Auto</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Chart Image]
Here you can determine how and when your data will be updated. If you opt for manual updates, the Refresh button will appear and allow you to update your data on demand. If you choose auto updates, you will also need to specify how often the system will update your data. Alternatively, you can opt out of data updates by selecting None.

6.5.6 User Permissions

This section allows you to specify what user categories will be able to see this tab.

6.5.7 Types of Charts and Their Uses

6.5.7.1 Pie Charts

Data arranged in columns of a single data source can be plotted in a pie chart. Pie charts show the size of items in one data series, proportional to the sum of the items. The data points in a pie chart are displayed as a percentage of the whole pie.
6.5.7.2 Column Charts

Data arranged in a single column of a single data source can be plotted in a column chart. Column charts are useful for illustrating comparisons between items. In column charts, categories are typically organized along the horizontal axis and values along the vertical axis.
6.5.7.3 Line Charts

Line charts can display continuous data over time, set against a common scale, and are, therefore, ideal for showing trends in data at equal intervals. In a line chart, category data is distributed evenly along the horizontal axis, and all value data is distributed evenly along the vertical axis.
Or in case of multiple data sources

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>field_X</td>
<td>field_Y</td>
<td>field_X</td>
<td>field_Y</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>90</td>
<td>5</td>
<td>90</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>30</td>
<td>15</td>
<td>70</td>
</tr>
<tr>
<td>3</td>
<td>23</td>
<td>40</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>80</td>
<td>40</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>70</td>
<td>70</td>
<td>50</td>
<td>80</td>
</tr>
<tr>
<td>6</td>
<td>80</td>
<td>10</td>
<td>71</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>90</td>
<td>50</td>
<td>80</td>
<td>66</td>
</tr>
<tr>
<td>8</td>
<td>90</td>
<td>50</td>
<td>95</td>
<td>40</td>
</tr>
</tbody>
</table>

6.5.7.4 Bar Charts

Bar charts illustrate comparisons between individual items.
6.5.7.5 Clustered Bar Charts

Clustered bar charts compare values across categories.
6.5.7.6 Scatter Charts

Scatter charts show the relationships among the numeric values in several data series, or plots two groups of numbers as one series of xy coordinates. A scatter chart has two value axes, showing one set of numeric data along the horizontal axis (x-axis) and another along the vertical axis (y-axis). It combines these values into single data points and displays them at irregular intervals, or clusters.
**DataSource 1.**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>field X</strong></td>
<td><strong>field Y</strong></td>
</tr>
<tr>
<td>2</td>
<td>2.00</td>
<td>2.00</td>
</tr>
<tr>
<td>3</td>
<td>2.10</td>
<td>3.00</td>
</tr>
<tr>
<td>4</td>
<td>5.70</td>
<td>6.00</td>
</tr>
<tr>
<td>5</td>
<td>6.80</td>
<td>7.00</td>
</tr>
<tr>
<td>6</td>
<td>9.90</td>
<td>9.00</td>
</tr>
<tr>
<td>7</td>
<td>20.00</td>
<td>11.00</td>
</tr>
<tr>
<td>8</td>
<td>14.10</td>
<td>13.00</td>
</tr>
<tr>
<td>9</td>
<td>16.50</td>
<td>15.00</td>
</tr>
<tr>
<td>10</td>
<td>12.00</td>
<td>17.00</td>
</tr>
</tbody>
</table>

**DataSource 2.**
Use Case. You want to collect feedback about your website logo. You are considering changing the color of your logo. You have the following color choices: blue, green, pink, yellow, none. First, you create a data source with these colors, select data type “boolean” next to each choice in the datasource dialog (we will have them as checkboxes) and the field to send colors to (name it “choices”). Data for the chart will be taken from that text field (“choices”).

You can view the results on the chart right below the question, or add it on a separate tab.
6.5.8 Use Case
Let us say we want to track our sales of pens on any given day. To do so, we will use a scatter chart and see the correlation between our dates and sales. After we have configured our chart in the View and Chart View sections, we open the Axes section and choose the date time for the x-axis. (By default, the y-axis supports only numbers.)

Next to Rates Content, we enter IN_TEXT(<chart.Y>) + " Pens were sold".

Then we select our data source, the corresponding fields, point type, line color, and title of our legend.
In the Update Chart section, we opt for manual updates.

In the end, we get the following chart:
6.6 Container Tab
The Container tab is identical to the New tab. It simply better conveys the functionality of this tab, the tab, on which you can place virtually any button.
To learn more about working with the Container tab, please click here.

6.7 Forum Tab

Adding the Forum tab allows you to open a dialog with your clients as well as to get them to post and respond to each other’s questions and requests. In addition, it’s a convenient way to receive feedback from them directly or via their comments to each other.

To participate in the discussion forum, the user needs to log in, which makes it another benefit of signing up.
To enable this functionality, simply drag and drop the Forum tab onto the canvas. In the popped up window, specify the title, tab name, icon, welcome title, welcome message, and font. Click OK.
Here is how this tab looks on a mobile device.
By putting a checkmark next to *Disable adding discussions*, you will remove the Ask button so that your users can only view the previously added comments and responses without being able to participate in the discussion.
You can also change the colors of the rows and the primary and secondary fonts in the App Settings- Colors and Fonts Section, to the left of the canvas. Before

After
Here is how this tab looks on a mobile device now.
Your app users will also receive email notifications when other users respond to their questions, comments, or requests.

Dear ...,

This is a Insurance Adjusters John notification:

You asked: *I'm using coolors.co for great color combinations and loving it!*

John responded: *Sounds like a great tool! Thanks for sharing.*

Best regards,

App Team

If you wish to unsubscribe, please click on the link.
This tab allows integrating specific Facebook pages into your app. For instance, you might want to share your Facebook page here so that your employees, customers, and clients can follow your page and promptly receive the most recent news and updates. All you need to do is to drag the Facebook tab onto the canvas and, in the popped up window, enter the URL of your page, where www is replaced by an m.

For example, Snappii’s URL is https://www.facebook.com/Snappii/, so we need to enter https://m.facebook.com/Snappii/
Here is how this tab looks on a mobile device.
If you want to simply enable your app users to access their Facebook accounts within your app, you will need to enter https://m.facebook.com/. Once they log in, they will be directed to their Facebook home page.

There is also a video tutorial available for this tab. To watch it, please click here.

6.9 Report List Tab

The Report List tab works very similarly to the Report List button, which we describe in detail here.

The settings are almost identical with the only exception that, in the User Permissions section, you can determine what types of users will be able to view this tab.
6.10 Gallery
The Gallery tab allows creating image galleries, such as catalogs with product images and descriptions, instructional materials with visuals, staff contact information with photos, etc. This tab/button requires creating a data source/data connector. The Gallery tab offers the following layouts:

Grid          Horizontal View          Split Screen
Here is a step-by-step description on how to utilize the Gallery tab.

1. Create the data source/data connector from which the app will pull its information, upload, configure, and save it in the data source dialog. (Please see *Data Sources and Data Connectors* for more details).

In our case, to build an app displaying a list of rooms at a bed and breakfast, we have created and uploaded in the data source dialog section an Excel spreadsheet with the following columns.

- RoomNumber
- Video
- Photo
- Description
- Availability

(We have split the screenshot of our Excel file in two parts so that you can see it better.)
(In the data source dialog, we saved this file as *King Size Rooms*.)

2. Once your data source/data connector has been built, configured, and saved, close the data source dialog, select the Gallery tab and drag it onto the canvas.
3. In the pop up window, change the title, the name of the tab, and the font and icon, if necessary.

Check off the box next to *Show Refresh button* if you wish.
4. In the Security section, check off the checkbox if you would like to enable the app users to add, delete, or update the information accessible in the app. Specify what user types can perform these operations within the app itself by clicking on *Add Operation Permission*, *Delete Operation Permission*, and *Update Operation Permission* one at a time and unchecking the corresponding checkboxes.

![Security section screenshot]

**Gallery View**

5. In the Gallery View section, select the data source/data connector the tab will use. In our case, it is *King Size Rooms*.
6. Rename the gallery view and detail view if necessary. (The gallery view is the information displayed once you select the Gallery tab. The detail view is what you see when you click on any image in the gallery view.)

7. Select any of the three gallery types (Grid/Horizontal Row/Split Screen). See their differences above.
8. If you have opted for the Grid View, select the number of columns in each information will be displayed in the drop-down menu next to Column count.

Here is a couple of examples of the Grid View in 2 and then 3 columns.
9. Check off the box next to *Show details on scroll automatically* if you wish. This option is available for the Split Screen Gallery only. If you check off this box, then the specified information from the data source in the Detail View will change together with the information from the Gallery View. In other words, the bottom part will change together with the top part.
If you leave this box unchecked, then the information from the Detail View will stay the same and only the top part will change.
10. Check off the box next to *Show header panel* if you want to enable the Sort and Search buttons at the top of the screen. (Make sure to check off the corresponding boxes under *Sort* and *Search* in the data source dialog for the Sort and Search buttons to work correctly.)

11. You may change the message that would be displayed when a search returns no results. In our case, it is “No rooms have been found”.
12. Click *change* next to *Gallery View template* and specify the height and the width of the cell that will display the information from the data source/data connector in the Gallery View. For instance, we entered 110 by 160 for the Grid view (with 2 columns), 365 by 320 for the Horizontal Row view, and 180 by 320 for the Split Screen view.

13. Click *change* next to *Detail View template* and, in the pop up window, specify whether you will want to use a detail view (the top option) or not (the bottom option.) Click *OK*. Please keep in mind that you have to select *Custom View* for the Split Screen gallery layout.
Here is how each of these three gallery types works.

A. Grid

   a) After you have completed all the steps above and clicked OK to close the tab settings, you will get the following screen.

Now click on *Edit Gallery View*. You will see the screen below.
b) The inner rectangular is the area to place any visual information from the data source you would like to be displayed in the Grid Gallery View. To do so, drag the Image element under Elements and Buttons.

c) Specify the dimensions of the element and whether you would like it to be shared.

d) Next to Source, select Data Source if you would like this information to be pulled from the data source/data connector or opt for Constant if you would like the element to stay the same and the way you see it in the editor.

e) Next to Select Image Field, choose any of the fields of the image type. In our case, it is Photo. Click OK.
Here is what our Grid Gallery looks like in the app itself.
f) Hit the Gallery Layout tab again and select *Edit Detail View*. On this screen, you can place any information you would like to be displayed when a user taps on any of the images from the Gallery View.

In our case, we’d like to display each room number, the picture of each room, its description, its video, and availability. Here are the steps to add and configure these elements:

- Select the Text element under *Elements and buttons* and drag it onto the canvas.
- In the pop up window, enter the name of the element. This is important when the source of the element is set as *constant*. However, when you pull the information from the data source/data connector, you can leave the original message.
- Select the font size, its alignment, color, and font. Specify the width of this text element.
- Next to *Source*, choose between *Data Source* and *Constant*. In our case, we selected *Data Source*.
- Next to *Select Text Field*, select the column from the data source from which the information will be taken for this text element. In our case, it is *RoomNumber*. 
➢ Click OK. The first element is now ready.
Now select the Image element under *Elements and buttons* and drag it onto the canvas.
- In the pop up window, change the image if you are planning to use the same image for all the organizations/locations. (In this case, the source of your image needs to be set as constant.)
- Configure the dimensions of the image.
- Next to Source, select Constant or Data Source. In our case, it is Data Source because our app will pull the information from the data source.
- Next to Select Image field, choose the image column from your data source/data connector that will provide the images for this image element. In our case, it is Photo.

- Click OK. Align the element if necessary. The second element is now ready.
Drag one more Text element onto the canvas and configure it in the same way described above except for Select Text Field. There choose a different column to pull the information from. In our case, we selected Description (the description of each room). Click OK. Align the elements if necessary. The third element is now ready.
➢ Select the Web element under *Elements and Buttons* and drag it onto the canvas.
➢ In the pop up window, enter the label for the element, if necessary, its dimensions, and configure its icon.

➢ Next to Source, select Constant or Data Source. In our case, it is Data Source because our app will pull the information from the data source.

➢ Next to Select URL field, choose the column from your data source/data connector that will provide the URLs for this element. In our case, it is Video.
Click OK. Align the element. The fourth element is now ready.
Select another Web element under *Elements and Buttons*, drag it onto the canvas, and configure it in the same way described above except for the *Label* and *Select URL Field*. There choose a different column to pull the information from. In our case, we selected *Availability*. Click OK. Align the elements if necessary. The fifth element is now ready.

Here is what our Detail View of the Grid Layout looks like in the app itself.
B. Horizontal Row

a) After you have completed all the steps above and clicked OK to close the tab settings, you will get the following screen.
b) Now click on *Edit Gallery View*. You will see the blank screen below. Here you can add any information from the data source/data connector you’d like to be displayed.

In our case, we’ll simply add one Text element, one Image element, and one Web element and configure them in the same way described above.
Here is what our Gallery View now looks like in the editor and the app.

The Detail View can be set up in the same way as the one for the Grid Gallery.

C. Split Screen

a) After you have completed all the steps above and clicked OK to close the tab settings, you will get the following screen.
Now click on *Edit Gallery View*. You will see the screen below.

b) The inner rectangular is the area to place any visual information from the data source you would like to be displayed in the Split Screen Gallery View. In our case, we’ll just add one Image element and configure it in the same way described above.
Here is what our Detail View of the Split Screen Gallery looks like in the editor and in the app.
c) Hit the Gallery tab again and select *Edit Detail View*. On this screen, you can place any information you would like to be displayed under the information in the Gallery View. In our case, we’ll add one Text element with the description of our Bed and Breakfast and one Web element for the Availability button.

- Add one Text element and configure its font, font size, alignment, color width, and specify if it can be shared. Next to *Source*, select *Constant*. Enter the text you would like to be displayed in the app. Click OK. Align the element if necessary.
The first element is ready.
➢ Now add one Web element. Configure it in the same way described above to add the Availability button. Click OK.

The second element is now ready.

Here is what the Detail View of our Split Screen Gallery looks like in the editor and in the app (together with the Gallery View).
Sorting Filter
In the Sorting Filter section, if you would like to enable your users to sort query results, such as room types, room numbers, etc., in the cell view either in ascending or descending order, check off the box next to Enable Filter.
Select *Apply Always* to display query results in a certain order by default. For instance, on the first screenshot below, the room numbers are listed from the lowest to the highest (ascendingly). But to reverse the room numbers, all a user needs to do is to click on *Sort*, then *DESC*, and *Sort*.

To enable your users to specify sorting preferences themselves, select *Be overridden when user selects search of sort options from the app.*
Just a reminder that you can enable the sorting of any number of the elements/fields from the data source/data connector in the data source dialog by checking off the corresponding boxes under Sort.

Search Settings

Simple Search Settings

15. To enable your users to search for specific information from the data source, check off the box next to *Enable simple search (search bar) in app*. Choose between

- =
- not =
- contains
- doesn’t contain

Here is how each of these options works:

A. =
For example, we would like to enable our users to search for specific rooms by their number. Here we entered 208 in the search bar above and clicked on Search to see Room 208.

In the example above, we used the Grid Gallery. Here is how this function works in the Horizontal Row Gallery.
This option allows excluding specific information from search results. For example, if we enter 202 in the search bar and click on Search, all the room numbers will be displayed except for Room 202.
C. Contains

This option allows displaying only those results that contain the word/number entered in the search bar. For instance, we would like to view only the rooms with a fireplace. So Room 203 disappears from the list of the rooms displayed in the Grid View Gallery as only Room 203 doesn’t contain the word “fireplace” in its description.
D. Doesn’t Contain

This option allows displaying only those results that do not contain the word/number entered in the search bar. For instance, we would like to view only the rooms without a TV. In our case, only Room 202 and Room 204 don’t contain the word “television” in their description.
Searching Filter

Advanced Search

When you check off the box next to *Enable advanced search button in app*, the button called *Advanced Search* appears at the top of the screen.
This button allows setting multiple criteria at the same time within the app.

For example, we would like to find all the rooms that have a fireplace except for Room 201. Then next to RoomNumber, from the drop-down menu, we should choose NOT = and next to Description, contains
As we know, all our rooms except for Room 203 contain a fireplace. However, we also specified that Room 201 should be excluded from the output results. That is why we only see rooms 202, 204, 205, 206, 207, and 208.

**Share Settings**

16. If you would like to enable your users to email or print out some specific information from the app or the server, in the *Share Settings*, do the following:
a) put a checkmark next to *Show share button*;

b) if you want all emails to be sent to a specific email address, enter it here; otherwise, your users will need to specify their email recipients every time they share information within this app;

c) enter the subject of the email;

d) enter a message, if you wish;

e) change the name of the Share button, if necessary:

f) specify the message that you users see upon successful email submission;

g) select whether to email data in the PDF, Excel (spreadsheet), or Word format;

h) enter the name of the shared file;

i) specify whether your users will be able to share data from the app or server;

j) click on *Data in Report* to uncheck those boxes that you don’t want to be shared in the report;
k) if you have selected *app* in the previous question, you can enable either emailing or printing of the data;

l) put a checkmark if you would like to include the user’s information in the report

m) choose the default template or click on *Show PDF templates dialog* to create your own template of the report. In the pop up window, click *New*, name the template, upload a header and a footer, if necessary, choose the title size, color, the layout of the fields, etc. Click *Save* and then *Close*. Now next to *PDF report template*, from the drop-down menu, choose the template you need. In our case, it is *Bed and Breakfast*. 
To email or print specified information from the app, click on *Share* and then select what needs to be shared.
Default Template

King Size Rooms Report

**RoomNumber:** 202  
**Photo:**

![Image of King Size Room](image-url)  

**Description:** This room features much original art, a French beveled mirror, antique writing desk, private bath, king size British Period four poster, canopied bed, original marbled working fireplace, a twin pull-out sofa bed, an over-sized bath with a corner Jacuzzi for two, and a private balcony overlooking Chippewa square.

Custom Template
That is what the report looks like as an attachment.

17. If you would like to add ads in this tab of your app, check off the box next to Show fixed banner ad, enter the ad unit ID for the iOS and/or Android version of your app, and choose the position of the ad from the drop-down menu.
Please click [here](#) to learn about how to get unit IDs.

### 6.11 JS Plugin Tab

The JS Plugin tab allows adding your own HTML, CSS, and JavaScript code to apps built on our platform. For example, you might want to use in-app HTML to create custom database query, registrations, or integration with other systems. You can also reuse existing code from your website. However, please keep in mind that presently you cannot call back into native code from the JS/HTML code.

**JavaScript plug-in capabilities for in depth app customization include:**

- **Code Reuse:** Snappii’s app builders can reuse existing code from their websites and other apps and drop it in the JS PLUGIN control, bringing the same features to their mobile apps.
- **JavaScript library use:** JavaScript is the most popular programming language in the world. It has become common in both game development and the creation of desktop applications, and it is now possible to use it in mobile app development as well.
• Extensibility: Further extend the Snappii platform by enabling JavaScript plugins to create fully customized, enterprise mobile apps. It is possible to add numerous features, such as image galleries, graphics, and animations, with different existing JS/jQuery plugins. Even simple games can be added to Snappii’s mobile apps.
• Enable Javascript extensions to work when offline.

For instance, you can use “Photo Swipe” plugin (http://www.photoswipe.com/) for a gallery. On its site, you can see that this plugin is mobile-oriented. That is why it works and looks fine on all devices.

A Zip file with the HTML code has been uploaded. It contains only one html file (index.html), js (JavaScript) and css (styles) files and also image files. The js and css files are connected to the .html file, and all links are relative (not absolute):

<link href="photoswipe.css" type="text/css" rel="stylesheet" />

<script type="text/javascript" src="klass.min.js"></script>

The links for images are relative too (e.g., <img src="images/01.jpg"…).

When using the JS/HTML control, all paths must be relative for the offline mode to work properly.

You can read more about integrating the JS Plugin here.
This tab allows implementing specific LinkedIn pages within your app. You have two options: you can either leave the home LinkedIn address http://www.linkedin.com/ as is so that your users can enter their credentials to access their page, or you can enter the URL of a certain profile on LinkedIn you would like your customers to look at. For instance, we could direct our app users to Snappii’s page by entering https://www.linkedin.com/company/snappii here.
All you need to do is to drag and drop the LinkedIn tab down, specify the title, tab name, URL, icon, and font.

Here is how this tab works on a mobile device for the home LinkedIn page and Snappii’s page.

There is also a video tutorial available for this tab. To watch it, please click here.

### 6.13 Login Form

This tab is used to enable app users to register and then log in. Below is a step-by-step description on how to successfully create the registration and login screens and take advantage of the Login tab.

1. Choose the Login Form under *Tabs* and drag it onto your canvas.
2. In the pop up screen, click **OK**. You can also change its title, name, icon, and font if necessary.
3. To create a field for entering email addresses, select the Text/Number field under *Fields and Buttons* and drag it onto the canvas.

4. In the pop up window, next to *Name*, enter *E-mail address*; next to *Data type*, select *email address*; next to *Placeholder*, enter *E-mail address*; next to *Login Field*, select *email*. Specify the width and click *OK*. You might also put a check mark next to *Save input history* to enable the auto-filling of previously entered e-mail addresses.
5. Align the e-mail field if necessary.
6. Now we will need to create a field for entering passwords. To do so, select the Text/Number field under Fields and Buttons and drag it onto the canvas.

7. On the pop up window, next to Name, enter password; next to Data type, select password; next to Placeholder, enter password; next to Login Field, select password. Specify the width you need and click OK.
8. Align the password field if necessary.

9. Under *Fields and Buttons*, select the Submit button and drag it onto the canvas.
10. In the pop up window, specify the height and width of the Submit button and whether you’ll want to view it as text only or change its icon by uploading it from the Internet or your computer.
11. Click OK. Align the button.

In our case, I just opted for “Global style” instead of changing the icon.
12. Under *Elements and Buttons*, select and drag onto the canvas the Forgot Password button. This button enables a user to receive an email with his/her password if the user has forgotten it.
13. On the pop up window, adjust the width and the height of the Forgot Password button. Change its icon, if necessary, or select *Global style*. Click *OK*.

14. Align the button. The login screen is ready.
15. Now we need to create the registration screen. To do so, select the Signup Form under *Elements and Buttons* and drag it onto the canvas.
16. Label this form as *Create Account*, change its title to *Create Account*, specify the height and width, change the icon or opt for *View only text*.

17. Click *OK*. Align the Create Account button.
18. Tap twice on *Create Account*. Now we will need to create an email field and a password field for this registration screen. We can either do so by following the steps described above for the login screen or by simply copying these two fields from the login screen. Below we will discuss the second option.
19. Click back on the Login tab and copy the email address field by
  a. selecting it and clicking on *Copy Element*,

  b. choosing *Copy to clipboard*,

  c. tapping twice on the Create Account button, and then hitting *Paste*. 
20. Repeat the same steps to copy the password field. Align the elements as necessary.
21. Select the Submit button under *Fields and Buttons* and drag it onto the canvas.

22. On the pop up window, specify the height and width of the Submit button and decide whether to keep the current icon, change it, or view the button as text only.
23. Hit OK and align the elements if necessary.

24. Make sure to make the Login Tab invisible to logged-in users by clicking on the Edit symbol
and then unchecking the box next to \textit{Logged in User}. 
Please keep in mind that you can add as many fields to your Sign up and Log in forms as you need. Thus, for instance, you might require your users to enter their first and last names, company’s name, phone number, or add any other field you find appropriate. Moreover, you can connect these forms to a data source or submit newly received data by email.

6.14 Logout
The Logout tab is another essential tab. In fact, until you add it, the Create Account and Login buttons will be inactive.

To add the tab, simply select it, drag it onto the canvas, and then click OK. You can also change its name, title, icon, and font if necessary.
Make sure to make this tab invisible to anonymous users by clicking on the Edit symbol and then unchecking the box next to Anonymous.
6.15 Nearby
(A video tutorial for this functionality is available [here](#).)

The nearby feature is a great option for vendors with multiple locations or for such businesses as hotels, which need to show to their guests restaurants, movie theatres, local attractions, etc. available in the area. All the search results will be conveniently displayed based on the user’s location (from the nearest to the farthest).

Here you can:

1) Upload a list of businesses you need to be displayed in the application (all cupcake bakeries in your state, for example).
2) Choose from the list of the offered business categories, such as banks, restaurants, movie theatres, etc.
3) Create your own list of business categories (car dealerships and auctions, for instance).
4) Limit the search results based on predefined criteria, such as 3D movie theatres, for example.

1st Option: Uploading a List of Business Locations
1. Add the Nearby Tab

2. Create an Excel spreadsheet with the following columns for any business location you’d like to add:

- the name of the needed location (for example, *Queen City Cupcakes*);
- its URL (for example, http://www.qccupcakes.com/);
- address (for example, 774 *Shem Valley Road*);
- city (for example, *Manchester*);
- region (for example, *NH*);
- country (if it’s not located in the USA);
- postal code (for example, *3110*);
- phone number (for example, *603-000-0000*);
- email address (for example, *qcc@gmail.com*);
- logo link (for example, http://goo.gl/fxURMQ);
- notes (if you want to show more details) Here is an example for you:

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>City</td>
<td>State</td>
<td>Phone number</td>
<td>Logo</td>
<td>Email</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Manchester</td>
<td>NH</td>
<td>603-624-4099</td>
<td><a href="http://www.qccupcakes.com/">Link</a></td>
<td><a href="mailto:queencitycupcakes@gmail.com">queencitycupcakes@gmail.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Bedford</td>
<td>NH</td>
<td>603-285-1967</td>
<td><a href="http://www.qccupcakes.com/">Link</a></td>
<td><a href="mailto:cupcake101@gmail.com">cupcake101@gmail.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Nashua</td>
<td>NH</td>
<td>603-386-5796</td>
<td><a href="http://www.qccupcakes.com/">Link</a></td>
<td><a href="mailto:cupcake10101@gmail.com">cupcake10101@gmail.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Dover</td>
<td>NH</td>
<td>603-302-6006</td>
<td><a href="http://www.qccupcakes.com/">Link</a></td>
<td><a href="mailto:dovercupcakes@gmail.com">dovercupcakes@gmail.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Concord</td>
<td>NH</td>
<td>603-728-6464</td>
<td><a href="http://www.qccupcakes.com/">Link</a></td>
<td><a href="mailto:concordcupcakes@gmail.com">concordcupcakes@gmail.com</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Note: You can download a sample spreadsheet from the Nearby tab and fill it out with your own information.

3. Click on the Edit symbol of the Nearby tab.

This window will pop up.
4. Select “I will upload a list of business locations”

5. Upload the spreadsheet

6. Make sure that the names of the columns of your spreadsheet correspond to the names of the categories in the editor.
7. Click *Ok* and then *Finish.*
Now you can open this tab in the Preview app, click on any of the listed business locations, and see all the information from the spreadsheet.

Here’s an example.
2nd Option: Using the Default Categories

1. Add the Nearby Tab

2. Click on the Edit symbol of the Nearby tab and then in the popped up window choose “I will use default categories”
3. Click Next to get to this screen

4. Here you can select or delete any of the given categories and change their images. When you are done, click Finish.

5. Here is an example of what you could see as a result.
For each business category, the user will see a list of business locations and will be able to call them, browse their websites, add to Contacts, get directions to them, etc.
3rd Option: Specifying Your Categories

1. Add the Nearby Tab

2. Click on the Edit symbol of the Nearby tab and then in the popped up window choose “I will specify my categories”

Use Google Search. Please, specify list of categories

- I will use default categories
- I will specify my categories
- Show search results for the following

3. Click on Add and type in any category/categories you need. You can also change the image of each category by clicking on Change image. Click Finish.
Here is our example based on the category “car” and “search string” car dealership."
4th Option: Showing Search Results for a Specific Word

1. Add the Nearby Tab

2. Click on the Edit symbol of the Nearby tab and then in the popped up window choose “Show search results for the following”.

Type in any word/phrase for which you need to display search results and then any filter if only those search results that do not contain the word/phrase specified in the filter should be displayed.

Here is an example with unfiltered results for “public schools” as the key phrase.
Watch video tutorial
This button will list your business locations that are closest to user location. Your apps users will be able to select one location and get directions. You can provide us with a list of locations that we will display to a user or you can specify a search phrase and we will automatically find all locations that match the phrase. For example “starbucks” will find all starbucks locations. You can also provide your own categories. For example “luxury hotel” and “economy hotel”. Read more in FAQ.

NearBy Icon: http://storage.store2phone.com/IconLibrar

Nearby Title: NearBy

Nearby Tab Name: NearBy

Font: Default

Use business list:

I will upload a list of business locations

Use Google Search. Please, specify list of categories

I will use default categories

I will specify my categories

Show search results for the following: public school

Filter results not containing any of the following phrases:

< Back  Finish

NearBy

Penacook Elementary School (703 m)

Boscawen Elementary School (4 km)

Broken Ground School (7 km)

Concord School District (8 km)

Concord High School (8 km)

Christa McAuliffe School (8 km)

Loudon Elementary School (10 km)
Here is an example for “public schools” where we have excluded elementary schools.
All these options utilize Google Maps, but, unlike the Advanced Maps, the output format of the Nearby tab cannot be customized.

6.16 Notifications
Notifications are a great way to attract more customers and clients. Thus, for instance, when you have an announcement, you can send it as a push notification, and all your users will see it on their Notifications tabs.

There are two ways to create a notification: on the platform itself and within one’s app.
To send a notification through the platform, follow the steps below:

1. Add the Notifications tab and save the app.

2. Click on Dashboard*.

3. Choose the application you wish to send a notification from and to which you have added the Notifications tab.
4. Go to the Notifications tab and click on *Create New Notification*.

5. Name the notification and specify what users the notification will be sent out to.

6. Write your message, insert images or/and URL, and schedule the date the notification will be sent out. You can select either one notification and send it immediately or schedule a specific date, or you can select ongoing notifications and indicate how often
and when your users will be receiving your notifications. The expiration date shows when your notification will disappear from the users’ Notifications tab.
*As a registered (non-subscribed user), you are able to access the Snappii dashboard features 5 times to test its functionality. Once you become a subscribed customer, you get unlimited access to the dashboard.

To send a notification from your app, please do as follows:

1. Add the One Time Notifications/Ongoing Notifications/ or Scheduled Notifications tab and save the app.
2. In the User Permissions settings of the tab, make sure that only you (the app owner) or some predefined users can see this tab and, hence, create notifications within the app.

3. You can also change the name, title, font, icon, and data source to which newly added notifications will be sent. Click OK.
You don’t have to change anything for these tabs to work correctly. However, since they implement the full functionality of the List tab, they can be easily customized to your needs.

4. Click on the Detail View of your tab and change the layout and design of this screen if you wish. Save the app.
Here is how we have customized each of these tabs.

The app owner’s/your device

Your user’s device
The app owner’s/your device

Your user’s device

The app owner’s/your device

Your user’s device
The PDF Form tab works very similarly to the PDF Form button, described in detail here.

However, unlike the PDF Form button, which allows users to edit and share PDF templates uploaded on their mobile devices via the Document Upload button, the PDF Form tab can only edit and share the template you upload in the editor in the settings of this tab.
Here is how this tab works on a mobile device.
6.18 PDF Viewer Tab
The PDF Viewer tab is used for displaying PDF files from a specified URL. Thus, for instance, you might want to enable access to an employee handbook or license agreement in this way.
All you need to do is to drag and drop the PDF Viewer tab onto the canvas and, in the popped up window, enter the URL from which the desired PDF file will be taken. You might also want to configure the title, tab name, icon, and font of the PDF tab.
Here is how this tab looks on a mobile device.
6.19 Catalog Tab

The Catalog tab allows creating product catalogs within your app. This tab doesn’t require building a data source/data connector and uploading it in the Data Source Dialog. Instead, all you need to do is to add the Catalog tab onto the canvas and create a simple spreadsheet. You may want to click *download sample spreadsheet* and enter your information into this spreadsheet.

Sample spreadsheet
As you can see, this tab requires the Name, Image1, Details1, Price, Taxed, Id, Image2, and Details2 columns. You cannot remove or add new columns.

To illustrate how this tab works, we created a spreadsheet for an organic farm. Please see the screenshot below.
We also specified our selection of the List view and Detail view templates by clicking *change*.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Name</td>
<td>Image1</td>
<td>Details</td>
<td>Price</td>
<td>Taxed</td>
<td>Id</td>
<td>Image2</td>
<td>Details2</td>
</tr>
<tr>
<td>2</td>
<td>Brie</td>
<td><a href="https://dl.dropbox.com">https://dl.dropbox.com</a></td>
<td>per lb</td>
<td>18.00</td>
<td>No</td>
<td>1</td>
<td><a href="http://cdn">http://cdn</a></td>
<td>Mild and creamy with a soft bloomy rind, our</td>
</tr>
<tr>
<td>3</td>
<td>Camembert</td>
<td><a href="https://dl.dropbox.com">https://dl.dropbox.com</a></td>
<td>per lb</td>
<td>18.00</td>
<td>No</td>
<td>2</td>
<td><a href="https://www">https://www</a></td>
<td>Much like the Brie, only made in smaller rounds.</td>
</tr>
<tr>
<td>4</td>
<td>Cheddar</td>
<td><a href="https://dl.dropbox.com">https://dl.dropbox.com</a></td>
<td>per lb</td>
<td>16.00</td>
<td>No</td>
<td>3</td>
<td><a href="http://food">http://food</a></td>
<td>A classic hard cheese made from our raw milk.</td>
</tr>
<tr>
<td>5</td>
<td>Blue</td>
<td><a href="http://static1.squarespace">http://static1.squarespace</a></td>
<td>per lb</td>
<td>20.00</td>
<td>No</td>
<td>4</td>
<td><a href="http://blog">http://blog</a></td>
<td>Our blue cheese is a creamy delicacy. Some co</td>
</tr>
<tr>
<td>6</td>
<td>Gouda</td>
<td><a href="http://static1.squarespace">http://static1.squarespace</a></td>
<td>per lb</td>
<td>18.00</td>
<td>No</td>
<td>5</td>
<td><a href="https://www">https://www</a></td>
<td>One of the oldest cheeses in modern producti</td>
</tr>
<tr>
<td>7</td>
<td>Buttermilk</td>
<td><a href="http://static.wixstatic">http://static.wixstatic</a></td>
<td>quart</td>
<td>5.00</td>
<td>No</td>
<td>6</td>
<td><a href="http://www">http://www</a></td>
<td>Our cultured buttermilk is made using the liqu</td>
</tr>
<tr>
<td>8</td>
<td>Raw Milk</td>
<td><a href="http://static1.squarespace">http://static1.squarespace</a></td>
<td>half gal</td>
<td>5.00</td>
<td>No</td>
<td>7</td>
<td><a href="http://www">http://www</a></td>
<td>Our milk has gained recognition across the sta</td>
</tr>
<tr>
<td>9</td>
<td>Greek Yogurt</td>
<td><a href="http://www.brookford">http://www.brookford</a></td>
<td>quart</td>
<td>5.00</td>
<td>No</td>
<td>8</td>
<td><a href="http://www">http://www</a></td>
<td>An extra strained version of our low-fat yogur</td>
</tr>
<tr>
<td>10</td>
<td>Feta</td>
<td><a href="http://static1.squarespace">http://static1.squarespace</a></td>
<td>per lb</td>
<td>15.00</td>
<td>No</td>
<td>9</td>
<td><a href="http://www">http://www</a></td>
<td>Traditionally, feta is a sheep's milk cheese, bu</td>
</tr>
<tr>
<td>11</td>
<td>Kefir</td>
<td><a href="http://www.brookford">http://www.brookford</a></td>
<td>quart</td>
<td>5.00</td>
<td>No</td>
<td>10</td>
<td><a href="http://www">http://www</a></td>
<td>Kefir is made from a symbiotic culture of yeast</td>
</tr>
</tbody>
</table>
You can also change the title, tab name, font, and the icons, but we left them as they were.

Here is how our Catalog tab looks and works on a mobile device.
In order to fully utilize the Catalog tab, you need to add one Shopping Cart Tab. Please read here how to work with the Shopping Cart Tab.

6.20 QR Code Tab

QR codes are effective marketing tools as they allow displaying and sharing information with others with the click of a button. They save space, time, and paper. Thus, for instance, you can share contact information, offer discounts and vouchers, share event details, enable your users to access and share with others audio and video files, to name but a few options. Here you will find several additional ideas for using QR codes.

Once you create a QR code you need (see QR codes in the upper right corner), add the QR Code tab, and select this QR code in the tab settings, your users will be able to scan it and instantly see your message.

The QR Code tab allows adding and displaying QR Codes and offers a variety of options.
Here are the steps on how to create QR codes in our editor and integrate them into the QR Code tab.

1. Click on "QR codes" in the right top corner.

2. Fill out the form with data depending on the type of the QR code you want to create. (The required fields are marked in red.)

Here are the QR code types we currently support:

- Contact info (MECARD or vCARD*)
- Email address
- Geo location
- Phone number
- Text
- URL
- YouTube
- Calendar Event
A vCard or Versitcard is the standardized file format for electronic business cards. This is a file format developed by the Versit group in 1995, which at that time consisted of members from Apple, IBM and Siemens. This is a universal format understood by most devices and software that handle things like contact information and email. MECARDs are used for the same purpose but have been developed to be slightly more portable. This makes them better suited for being distributed as QR codes. So the difference is slight and you can choose either of the options.

- Next to Name, name the QR code. After you fill out all the required fields and save the newly-created QR code, its name will appear on the left-hand side.
- Next to Tag, enter a short description of the main function of the QR code. This could be anything from “Product Description”, “Upcoming Trade Show”, to “Our Contact Info” or “Find Us on YouTube.”
- Select the type of the QR code you are creating. Below you’ll see that a different number of fields appear depending on the type you choose. Fill them out.
- Specify the size of the QR code on a mobile device.

Here are a few examples of QR codes we’ve created to illustrate these options.

**Contact Info**

*In the Settings*
On a Mobile Device before Being Scanned

On a Mobile Device after Being Scanned

YouTube URL

In the Settings
On a Mobile Device before Being Scanned

On a Mobile Device after Being Scanned

Calendar/Event

In the Settings
3. You can use QR codes as coupons. Your app will track the number of times a user scans the coupon and unlock after a specified number of scans to give the user the reward.

You can also require users to register and log in to be able to use the coupon.
• Put a checkmark next to *Use as a coupon*.
• Put a checkmark next to *Active* to activate the coupon.
• Specify the number of scans needed to redeem the coupon.
• Put a checkmark next to Show QR code image if necessary.

If you have set up a coupon with "number of scans to unlock" \( \geq 1 \), then you will probably not want to display the QR code image in the app. You will want to keep the QR code image where you can give users access only when they have fulfilled some requirement like visiting your store or making a purchase.

• Indicate that only logged in users will be able to use the coupon.
• Click *Save*. If you need more coupons, press *Add* to proceed.

Here is how this option works on a mobile device.
4. Now you can select this ready-made QR code in the settings of the QR Code tab/button by clicking on Select QR Code.
Here you can also specify the title, tab name, QR image, icon, and font. Click OK.

Here is how the QR Code tab looks and works on a mobile device.
EVENT
We are Hiring!

Location: 36 Industrial Way, Rochester, NH 03867 USA

Starts: Wed Jun 01 09:00:00 EDT 2016
Ends: Wed Jun 01 14:00:00 EDT 2016

Show QR Code Image
Share by Email
Add Event

Add to Dropbox  Bluetooth
ChatON  Copy to clipboard
Email  Flipboard
Gmail  Google+
5. To make another QR code, go to QR codes again and press Add in the upper left corner. All the created QR codes will be listed in the menu to the left.

6.21 QR Scanner Tab
The QR Scanner tab allows your app users to scan QR codes, save them, and share them with others.
To enable this functionality, please follow the steps below:

1. Drag and drop the QR Scanner tab onto the canvas.

2. In the popped up window, specify the title, tab name, tab icon, font, and whether back or front camera will be used by default.
3. Put a checkmark next to *Show scan button to user.*

*Show scan button to user enabled*  
*Show scan button to user disabled*
4. Put a checkmark next to **Load QR Code list** to enable your app users to see all the QR Codes made public by the app builder as well as see all the QR codes they have previously scanned.

This option also allows deleting previously scanned QR codes.
5. Put a checkmark next to *Show action buttons* to enable your app users to view scanned QR codes in browser, show QR Code images, and share them with others.
6.22 Reminder Tab

The Reminder tab allows your users to add reminders within the app. To add a reminder, drag and drop the Reminder tab onto the canvas.

In the popped up window, specify its title, tab name, icon, and font.
Now open the app in the Snappii’s Preview App on your device and click Add to add a reminder. You can specify its title, date, time, type (once, repeatable, alarm), snooze duration (1/10/30 minutes), vibration (off/on), and ringtone.

Once the pop-up notification arrives, you can choose to stop it or snooze.
6.23 RSS
RSS looks like a news feed when added to an app. Users can read news or blogs right within your app. Thus, for instance, you can share your recent blogs or feeds on something related to your industry.

To enable this functionality, simply drag and drop the RSS tab onto the canvas. In the popped-up window, enter the link to the RSS feed you plan on sharing with your users.
For example, we entered “New York Times RSS” in a search engine to get directed to the page below.

Then we chose a feed we are interested in and clicked on it to get to the screen below.

After that, we simply copied the URL link of this feed into the settings of our RSS tab.
This is how our RSS tab looks on a mobile device now.
6.24 Web Search Tab

The Web Search tab works very similarly to the Web Search button. Please click here to read about working with the Web Search tab/button.

6.25 Shopping Cart Tab

The Shopping Cart tab enables your users to add products/services from the Catalog tab/button into their shopping carts.
This tab is extremely easy to configure. All you need to do is to drag and drop the Shopping Cart tab onto the canvas and, in the popped up window, specify the icon, tab name, title, and what categories of users will be able to view this tab. Click OK.

This tab works in conjunction with the Catalog tab/button, so you will need to add one Catalog tab/button in order to fully utilize the functionality of this tab.
In the first screenshot below, you can see the shopping cart icon. When a user clicks on it, a window with *This item will be added to your shopping cart. Yes/No* pops up. When the user clicks *Yes*, this item goes into the shopping cart. When the user opens the Shopping Cart tab/button, he/she can see all the items that have been added in this way.

![Shopping Cart Screenshot](image)

6.26 Sign/Draw Tab

With the help of the Sign/Draw tab, you can create and share digital signatures or draw pictures, as well as send them to your co-workers or clients.

To enable this functionality, simply drag and drop the Sign/Draw tab onto the canvas.
In the popped up window, specify the title, tab name, icon, and font of the tab.

Watch video tutorial

Read more in FAQ

Title: Draw a Sketch
Tab Name: Draw
Icon: http://storage.store2phone.com/IconLibrary/Device
Font: Arial

Here is how this tab looks on a mobile device.
Now your app users can choose font colors and sizes by clicking on the rectangle and circle accordingly.

**Changing Colors**

**Changing Font Sizes**
Sharing Signatures/Sketches

There is also a video tutorial available for the Sign/Draw tab. Click here to watch it.
6.27 Sign up Form Tab

The Sign up Form tab allows creating registration screens and customizing them the way you like.

To enable this capability, follow the steps below.

1. Select the Signup Form tab and drag it onto the canvas.

2. In the popped up window, specify its title, tab name, icon, and font.
3. Under *Submit to Data Source*, indicate if your users’ registration information will be sent and stored in a specific data source. To do so, the corresponding Excel file needs to be uploaded and the data source needs to be configured in the Data Source dialog. Then the highlighted fields on the screenshot below also need to be configured.

4. Under *Send reports via e-mail*, specify if you would like to receive an email with a PDF/Excel/Word report containing the registration information of every new user.
5. Under Messages and labels, specify the messages your users will see after they have entered their registration information and clicked Submit/Sign up/Create Account.

Click OK.

6. Now we will need to create an email field and a password field for this registration screen.

To create a field for entering email addresses, select the Text/Number field under Fields and Buttons and drag it onto the Sign up Form tab.
7. In the popped up window, next to Name, enter E-mail address; next to Data type, select email address; next to Placeholder, enter E-mail address; next to Sign up Field, select email. Specify the width and click OK. You might also put a check mark next to Save input history to enable the auto-filling of previously entered e-mail addresses. Make the field required so that your users can’t register without entering their email address. Put a check mark next to Include field in report to receive the email address of each newly registered user in a PDF/Excel/Word report. Click OK.
8. Now we will need to create a field for entering passwords. To do so, select the Text/Number field under Fields and Buttons and drag it onto the Sing Up Form tab. In the popped up window, next to Name, enter password; next to Data type, select password; next to Placeholder, enter password; next to Sign up Field, select password. Specify the width you need. Make the field required so that your users can’t register without entering their password. Click OK.
9. Select the Submit button under *Fields and Buttons* and drag it onto the Sign up Form tab. In the popped up window, specify the label, dimensions, font, and style of the Submit button. Click OK.
Here is how our Sign up Form tab now looks on a mobile device.

10. We recommend that you make the Sign up Form tab invisible to logged-in users. To do so, in the settings of the tab, under User permissions uncheck the box next to Logged in User.
This tab allows integrating Twitter timelines within your app and, thus, enabling your users to follow the tweets of predefined users.

All you need to do is to drag and drop the Twitter tab down, specify the title, tab name, username, icon, and font. Thus, for instance, we entered *nytimes* to allow our users to follow the tweets by the New York Times.
Alternatively, you can enable your users to follow your tweets.

Thus, we entered *snappii_app* to direct our users to our page on Twitter.
Use this tab to show your tweets to the users.

<table>
<thead>
<tr>
<th>Title</th>
<th>Twitter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tab Name</td>
<td>Twitter</td>
</tr>
<tr>
<td>Username</td>
<td>snappi_app</td>
</tr>
</tbody>
</table>

The quick brown fox jumps over the la...

User permissions

- Change icon

- OK
- Cancel
The Form tab works very similarly to the Form button. For more details, please click here.
The Web tab enables your users to access a predefined website right within your app. For example, you might want this to be the website of your company or any website they might need to frequently use.

All you need to do is to drag and drop the Web tab onto the canvas and, in the pop-up window, specify the title, tab name, URL, icon, and font. You might also want to check off the box so that the tab can be opened in a new browser.
To illustrate how this tab works, we entered the URL of one Bed and Breakfast Inn. Here is how the tab works on a mobile device.

There is also a video tutorial available for this tab. To watch it, please click here.
The YouTube tab enables your users to access a predefined page with a video right within your app.

All you need to do is to drag and drop the YouTube tab onto the canvas and, in the pop-up window, specify the title, tab name, icon, and font.

Next to *URL to YouTube video*, enter the URL from which the app will pull the video.
This is how the tab works on a mobile device.

There is also a video tutorial available for this tab. To watch it, please click here.
Chapter 7. How to Use Elements and Buttons

7.1 Text Area

The Text button is one of the most frequently used buttons in our editor. Its name speaks for itself: it allows adding text inputs and configuring them in a variety of ways.

To enable this capability, please follow the steps below.

1. Drag and drop the Text button onto the canvas.
2. In the popped up window, next to Text, enter any text that needs to be displayed.

3. Specify the font size, alignment (left, center, right), font color, width (auto or custom), and font.

4. Put a checkmark next to Sharing enabled if you would like to allow your users to share this textual element with others when the Share button is located on the same tab/form.
Here is how the Sharing feature works on a mobile device.
5. Put a checkmark next to *Use Data Source* if you would like the information for the Text button to be pulled from a data source. For instance, we used the Gallery tab for the app below and needed textual information (a description) to change for each room.

Here is how the Text button looks in the editor in our case.
In the settings of the Text button below, you can see that the text we have entered in the text field is different from the one that gets displayed on a mobile device because we have chosen to pull the information for the Text button from a data source, not to keep it constant (the way specified in the text field.)

Next to Select Text Field, we have selected Description since this is the field in the connected data source that contains the room descriptions we need.

On the screenshot below, you can see that we use the data source called Bed and Breakfast Deluxe for this Gallery tab.
Here is how this data source looks in the data source dialog.

Here is the Excel spreadsheet of our data source with the description field.

Alternatively, you might want to keep the information for your Text button constant, like in the app below.
Here you can see that the information from the text field (New Report) and the app is displayed in exactly the same way.
There is also a video tutorial available for the Text button. Click [here](#) to watch it.

### 7.1.1 New Text area

New settings for the text field make them more flexible and useful. Let’s consider these changes on examples:

![View Settings](image)

**Name** – this parameter is not reflected anyhow when you work with the app, its purpose is to send the ID to the server. It’s best to be left untouched, because by default all these names are numbered correctly by order. However, if you decide to change that name, please use unique names for each text area.

**Text** – this is the field you insert the text into, it will be displayed in the app.
Allow HTML – once enabled, you can use html tags to edit your text. You can use:

1. `<a>` (supports attribute "href")
2. `<b>`
3. `<big>`
4. <blockquote>
5. <br>
6. <cite>
7. <div>
8. <em>
9. <font> (supports attribute "color")
10. <i>
11. <p>
12. <small>
13. <strong>
14. <sub>
15. <sup>
16. <u>

Some color names are supported for the font:

1. aqua
2. black
3. blue
4. fuchsia
5. green
6. grey
7. lime
8. maroon
9. navy
10. olive
11. purple
12. red
13. silver
14. teal
15. white
16. yellow

Let’s use purple Arial bold italics font:
If we disable html tag view, we will get exactly what we wrote:

Height, width – stands for height and width of the text area. By default the size is 45x100 pixels which looks plain:

If you apply “auto” for Height, the text area will be adjusted to include text:
If you want to escape a line break, set up the width manually, let’s use 200 pixels:

Now the text is displayed in one line.
Auto Shrink – if this option is enabled, the font size inside this text area will automatically change to adjust to the area width. Let’s enter some text and set up a wider size of the text area than needed:

And now with the auto shrink option enabled:
The font size has enlarged to fill in the text area we have set.

Auto Link – if enabled, it will have all the URLs clickable. The system will search for the [www.website_address](http://www.website_address) and highlight it in blue. You will be able to click that link and open it in browser. Let’s try to add it and check on device:
Alignment – this setting allows you to select three ways to align the text: to the left, center and to the right.

To the left:

Centered:
To the right:

Alpha Channel – determines the font transparency. The available values vary from 0 to 1 and are inserted with a dot, for example, 0.1, 0.2, 0.3, etc. Let’s set up 30% transparency inserting the value as 0.3:
Sharing enabled – once enabled, it allows you to share the entered information via sms, email or social channels.

Text style and background style – are the settings taken from the Visual app style. If the text area setting states “Global Style”, it means that the settings are taken from the Visual app style, all new text areas that are added to the app will have Global style by default. Let’s set up the visual settings:
Font – select the font.

Font size – stands for the size in pixels.

Color – font color.

Shadow – enable/disable the shade.

  Color – shade color.

  X, Y – vertical and horizontal shade drift

Background color – stands for the color of the background of the text area.

Border – enable/disable displaying the border.

  Color – border color.
Width – border width in pixels.

Radius – corner rounding radius in pixels.

Padding – enable/disable padding in the inner borders of the text area.

Top, Left, Bottom, Right – indicates the padding from a certain border.

Here’s what we will get:

![New Widget](image)

We have set up Global style, which means that any new text entry will have this look. However, each text area has the same set of features and can be changed however you want, just choose Custom Style:

<table>
<thead>
<tr>
<th>Text style</th>
<th>Custom style</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom style</td>
<td>Customize</td>
</tr>
<tr>
<td>Background style</td>
<td>Custom style</td>
</tr>
<tr>
<td>Custom style</td>
<td>Customize</td>
</tr>
</tbody>
</table>

Once you click Customize, you will see the same settings we have in Visual Style. The only difference is they will be applied to this particular text area.
Now any text area can have a unique style. This will help you create amazing design for your apps.
The Container button, formerly known as the Custom button, provides you with the most flexibility. Its functionality is not as limited as the functionality of the Map, List, or Form tabs, designed to perform specific functions. On the Container button, you can create a totally custom page by adding and configuring any buttons and elements of our editor.

Click [here](#) to watch a video about the Container button.

Here are a couple of examples of how the Container button was utilized by our app builders.
To enable this capability, simply drag and drop the Container button onto the canvas and, in the popped up window, specify its label, icon, dimensions, icon style, and font.
7.3 Image Button

The image button is widely used within our apps. Start by dragging and dropping the Image button onto the canvas.
1. Now you need to specify what image will be used in this button. Images can be pulled from data sources, and then the same image button can display different images. Alternatively, you can use an image that doesn’t change by uploading it from your computer, specifying the web page where it can be found, or specifying the URL of the image.

**Uploading Image from Computer**

**Specifying Web Page**
2. Once the image has been chosen, enter its dimensions. Not to distort the image aspect ratio, we recommend that you check off the box next to *Save original aspect ratio*. To
preserve the size of the original image file, you might also want to check off the box next to Don’t reduce the size after saving.

3. Check off the box next to Sharing enabled if you would like to allow your users to share the image by email, on Facebook, Twitter, etc.

For instance, we checked off the box for the image below.
Then we added the Share button, specified its label, e-mail settings, share settings, dimensions, font, and button style.

Use this element when you want to share information into social networks, send by SMS or into email, print. Data for sharing will be taken from controls where option 'Sharing enabled' is checked.

Now our app users can share this image by clicking on Share, selecting the image to share, and the sharing option.
4. Next to Source, select Data Source or Constant, depending on where the app will need to pull the image/images from.
(The image above is Constant, so the app pulls the image from the URL we specified next to Change Image.)

The image button on the welcome screen of this app uses a data source because the images change as we scroll them down. (The image button was placed on the Gallery tab.)
Watch video tutorial

Use this element to add pictures in your application.
5. Next to **Select Image Field**, choose the name of the image field in the data source dialog. (In our case, it is `image`.)
You can also watch a video tutorial available for this button by clicking here.
7.4 Call Button

The Call button enables users to reach predefined numbers easily and conveniently. All you need to do is to drag and drop the Call button onto the canvas and, in the popped up window, enter the phone number you would like your users to reach with the click of a button. In our case, we entered Snappii’s phone numbers to enable our users to contact us for assistance.
You might also want to change the label, icon, dimensions, style, and font of the Call button.

Here is how our version of the Call button looks and works on a mobile device.

There is also a video tutorial available for the Call button. To watch it, please click here.
7.5 Email Button

The Email button enables users to contact predefined companies and individuals easily and conveniently. All you need to do is to drop the Email button onto the canvas wherever you would like to place it and, in the popped up window, specify the label, icon, dimensions, style, and font of the Email button. Most importantly, enter the default email address you would like your users to send their emails to. In our case, we provided Snappii’s email address to enable our users to reach our customer support with the click of a button.
Here is how our version of the Email button looks and works on a mobile device.

There is also a video tutorial available for the Call button. To watch it, please click here.

7.6 Web Button
The Web button works very similarly to the Web tab.
7.7 Calendar

The Calendar will help you track different events using a convenient user interface. The datasource that will be used to display the data must contain two fields with “datetime” type: name and event description and two “datetime” type fields with Start and End time:

You can tie this data source to a form and add data. To display the data, add the Calendar element. Let us consider its settings:

We will review the List View block as we have reviewed the other blocks earlier.

Along with the standard settings you will need to add 4 parameters for the calendar: Title field, Description field, Event start field and Event end field.
These values will be displayed in your event description:
As you can see we have added Event 1 on April 12, that will last from 11am till 2pm. Such a layout is convenient if you provide services like barber’s shop or car wash. The calendar clearly outlines the date and time to schedule an appointment. You can switch to hourly format to see the time frame taken for this event:
Since the Calendar serves to display data, you can set up the detail view to show more details about your event. Just click on the event to view more details:
Event 1

The event starts: 04/12/2017 11:00 AM
End of the event: 04/12/2017 2:00 PM

New technical event

You can certainly add any information to the event description: video, photo, etc.
Along with the List tab/button, the Form button is a key element of our platform and is used in the majority of our apps. The Form button allows creating digital forms and filling them out with textual and numeric information, audio and video files, images, etc. In fact, the whole section Fields and Buttons was designed to be used with the Form button inside of our digital forms.

After the form is filled out and the user clicks Submit/Send/Email, all the filled out information gets saved in a data source connected to the form, gets stored on the device, gets sent to the specified email address, and becomes available for accessing, editing, sharing, or deleting at any time later both when the user is online or offline.

To give you an example, here are a few screenshots of the digital form called Newly Daily Report Form from one of our apps.
It is possible to collect various data types from your app users. You can collect the following information:

- Names
- Contact information
- Photos and pictures
- Email addresses
- Dates
- Times
- Locations
- Other details

To enable this functionality, please follow the steps below.

1. Drag and drop the Form button onto the canvas. In the popped up window, you will see the following settings.
7.8.1 View

Configure the View settings of your Form button.

The View section allows configuring the way your Form button will look.

Here you can specify the label, title, icon, dimensions, font, and view mode of your Font button.
Here is how the View section of our *Newly Daily Report Form* button is set up.

### 7.8.1.1 Viewing Conditions

The Viewing Conditions subsection allows making elements active and inactive, as well as hiding or revealing them, - all based on a given formula.
When we enable an element, we make it active and, therefore, can fill it out. When we disable an element, its field turns grey and can’t be filled out or edited in any way until we satisfy the condition required by the formula we have specified in its settings.

*Use formula to show/hide element* shows or hides the given element based on the specified conditions.

To illustrate how the Use formula to enable/disable element option works, we have added one Form element onto the canvas and placed a few different fields onto it.
Let us say the City field needs to be disabled when the Name field is empty. To enable this capability, follow the steps below.

1. Open the settings of the City element and click on *Viewing Conditions*.

2. Put a check mark next to *Use formula to enable/disable element* and select *Use formula to disable element*.

3. Next to *Formula*, enter `IS_EMPTY(<name>)`

As you can see, here we utilize the formula `IS_EMPTY()`, where in the parentheses we indicate the name of the element that needs to be filled out to make our element (the City field in our case) active.

Let us check how this formula works on a mobile device.
When the Name Field Is Empty

When the Name Field Is Filed out

In the screenshot to the left, the Name field is empty, and the City field is grayed out, which indicates that the latter field is inactive. In the screenshot to the right, the Name field is not empty, and, therefore, the City field is active. Naturally, removing the text from the Name field will disable the City field again.

Now let us look at using formulas to show and hide elements. Here we use the same digital form we created above. However, this time we are going to work with the Photo field and the radio buttons.
Here is how we have configured the radio buttons:

**The Radio Button to the Left**

<table>
<thead>
<tr>
<th>Value</th>
<th>yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group name</td>
<td>rad1</td>
</tr>
<tr>
<td>Checked</td>
<td></td>
</tr>
<tr>
<td>Read only</td>
<td></td>
</tr>
</tbody>
</table>

**The Radio Button to the Right**
We would like the Photo field to appear when our users select Yes (the radio button on the lefthand-side) and to disappear when our users select No (the radio button on the right-hand-side).

To enable this capability, follow the steps below.

1. Open the settings of the Photo field and click on Viewing Conditions.

2. Put a check mark next to Use formula to show/hide element and select Use formula to show element.

3. Next to Formula, enter `<rad1> = = “yes”`

As you can see, here we utilize the formula `()- = “yes”`, where `rad1` means the group name of the radio buttons we have created above, and “yes” indicates the button with the positive meaning.
Let us see how this formula works on a mobile device.

When we have just opened the form and, therefore, haven’t chosen the positive radio button, we don’t see the Photo field since the condition specified in the formula hasn’t been satisfied.

When we click Yes, the Photo field appears.
When we select *No*, the Photo field disappears again.
These are some of the easiest and most frequently used formulas in connection to the viewing options. However, the possibilities are nearly limitless. Let us look at a couple of other examples.

For instance, we might need to display a specified photo only after our users have entered the correct password.

To enable this capability, follow the steps below.

1. In the settings of the Photo element, next to Viewing Conditions, select Use formula to show/hide element.

2. Select Use formula to show element.

3. Next to Formula, enter <pass> = = “123”
As you can see, here we utilized the following formula: () = = “()”, where in the first parentheses we entered the name of the field (in our case, the Password field) that needs to be filled out correctly for the photo to be displayed. In our case, the password is 123, so we entered “123” in the second parentheses.

In this example, we have also put a check mark next to *Automatically move other controls up to fill space when this control is hidden* to move all the elements below the Password field up when the Photo element is hidden. Here you can see that the blue rectangle moves up when the Photo element disappears. However, once we have entered the correct password, the photo appears, and the blue rectangle moves down the Photo element.

If we uncheck the box next to *Automatically move other controls up to fill space when this control is hidden*, we see that the blue rectangle doesn’t change its position regardless of the visibility of the Photo element.
As you can see, opting for moving the other controls up when our target element is hidden prevents empty spaces between visible elements when some others are hidden.

Let us look at one more example. This time, we need to make sure that every new field appears only after the previous ones have been filled out. To illustrate this option, we built one digital form with three Text fields: Password, Gender, and Phone number.
Since we need to keep the first element always visible, we are not going to make any changes to it. However, for the other two fields and the text elements, we had to specify the viewing conditions.

For the Gender field and its caption (a text element), we:

1. Put a check mark next to *Use formula to show/hide element,*
2. Selected *Use formula to show element,*
3. Next to *Formula,* entered `IS_NOT_EMPTY(<pass>)`

As the screenshot above illustrates, here we employed the following formula: `IS_NOT_EMPTY(),` where `<pass>` stands for the name of the very first field of our form (the Password field).

For the Phone field and its caption (a text element), we:

1. Put a check mark next to *Use formula to show/hide element,*
2. Selected *Use formula to show element,*
3. Next to *Formula,* entered `IS_NOT_EMPTY(<pass>) && IS_NOT_EMPTY(<gender>)`
As you can see, here we entered both the Password field and the Gender field, which need to be filled out for the Phone field to appear. The && means “and”. If we wanted only one of the fields to be filled out for the Phone field to show up, instead of && we would have entered ||, which means or.

Here is how these formulas work on a mobile device.

Here are some additional formulas you can use in the Viewing Conditions section.

1. `<FieldName>.length > 0` – will enable/disable one field if the other field (standing for ‘FieldName’) is filled out;

2. `DISTANCE(44.123456, 54.123456, <user.location>) > 200` - will enable/disable the field if the user is located further than in 200 miles/km from a certain point (the coordinates of which are specified in the formula);
3. **DISTANCE(<Location1>, <user.location>) < DISTANCE(<Location1>, <Location2>)** - will enable/disable a field if the first distance (from a certain location to the user) is smaller than the second distance (from one location to the other);

4. **(<date1> - <date2>) < 3** - will enable/disable a field if the difference between the two dates is less than 3;

5. **<user.userName>.length > 100** - will enable/disable a field if the user’s name has more than 100 characters in it;

6. **SUM(<TableName.FieldName>) <= <NumberField1> + <NumberField2>** - will enable/disable a field if the sum of all the data from the table is less or equal to the sum of the two number fields (‘NumberField1’ and ‘NumberField2’);

7. **<date1> > <date2>** - will enable/disable a field if the date in one field (called ‘date1’) is greater than the date in the other field (called ‘date2’);

8. **<date1> == TODAY** - will enable/disable a field if the date entered in one field (called ‘date1’) is today’s date;

9. **<user.userType> = “LoggedIn”** - will enable/disable a field if the user who sees it is logged in.

### 7.8.2 Report Common Settings

3. Configure the way filled out information will be emailed once the user finishes completing the form and clicks *Submit/Send/Email*.

The Report Common Settings section allows you to specify the name of the report, whether your app users will be able to choose its type (PDF, Excel or both), whether the user’s information and time of submission will be included in the report (for Default reports only), and configure the email, PDF, and Excel report formats.
Please keep in mind that you can configure the names of your reports in a variety of ways. Thus, for instance, you can add a report number, timestamp, or username to each of your report.

Here you can specify a file name for PDF and/or Excel reports. You can insert NUMBER, TIMESTAMP and USERNAME variables to add a sequence number and a timestamp to the file name. Examples of what you can use as a report file name:

- **Report NUMBER**
  Your reports will look like: Report 1, Report 2, Report 3, ...

- **Report TIMESTAMP**
  Your reports will look like: Report 04.05.14, 02:43:55 AM, Report 04.05.14, 02:43:53 AM ...

- **USERNAME Report NUMBER**
  Your reports will look like: John Smith Report 1, John Smith Report 2, John Smith Report 3

If, for example, next to *Report file name*, you enter `<Project name> Report Number`, your report name will look like this:
In this way, every time you submit your form, the number of the report will go up: Report 1, Report 2, Report3, Report 4, Report 5, etc.

If, for instance, next to Report file name, you enter <Project name> Construction Daily Log NUMBER, you will receive reports under the following name with Construction Daily Log 1, 2, 3, 4, etc.:
If next to *Report file name*, you enter `<Project name> Inspection TIMESTAMP`, your report name will look in the following way:

The screenshots below illustrate how the other above-mentioned options work.

**Report File Name**
Allow Users to Choose the Format of the Report

Include User Information
## Construction Daily Log

**DateTime (UTC 0):** 6/25/2016 7:55:33 PM  
**User name:** John Smith  
**User email:** johnsmith@gmail.com

**Date:** 06/25/2016  
**Project name:**  
**Contractor:**  
**Day:**  
**latitude:** 43.265167  
**longitude:** -71.59011  
**address:** Concord, NH 03303, 2 A Way

**State:**  
**Temperature:**  
**Client Manager On-Duty:**  
**Address:** Way, 2  
**Arrival time:**  
**Depart time:**  
**Work Performed Today:**  
**Subcontractor's Progress:**  
**Issues - Delays:**  
**Extra Work Requests (include by whom):**  
**Material Purchased / Received:**  
**Labor - Total Work Hours:**  
**Labor - Total Travel Hours:**  
**Subcontract - Total Travel Hours:**  
**Subcontract - Total Work Hours:**  
**Rentals - Total Hours:**  
**Rentals - Total Cost:**  
**Weather:** Sunny  
**Measure:**

---

**Include Submission Timestamp**

---
Construction Daily Log

DateTime (UTC 0): 6/25/2016 7:53:33 PM
User name: John Smith
User email: johnsmith@gmail.com
Date: 06/25/2016
Project name:
Contractor:
Day:
latitude: 43.252187
longitude: -71.59011
address: Concord, NH 03303, 2 A Way
map/image:

State:
Temperature:
Client Manager On-Duty:
Address: A Way, 2
Arrival time:
Depart time:
Work Performed Today:
Subcontractor's Progress:
Issues - Delays:
Extra Work Requests (include by whom):
Material Purchased / Received:
Labor - Total Work Hours:
Labor - Total Travel Hours:
Subcontract - Total Work Hours:
Subcontract - Total Travel Hours:
Rentals - Total Hours:
Rentals - Total Cost:
Weather: Sunny
Measure:

Default Email Body Text
support@snappii.com

To

App name: Construction Daily Log
Form name: New Daily Report Form
Attached you will find the data collected by Form “New Daily Report Form” from your users.

2 Attachments | View all | Download all

from 06.25.xlsx from 06.25.pdf

Default PDF Report Template

Custom PDF Template
## Daily Report

**Report Date:** 09/01/2016

**Contractor:** [Name]

### Project Information
- **Project Name:** [Name]
- **Client Manager On-Duty:** [Name]
- **Address:** [Address]
- **Weather:**
  - **Temp:** [Temp]
  - **Arrival Time:** [Time]
  - **Departure Time:** [Time]

### Work Performed Today

### Subcontractor Progress

### Table

<table>
<thead>
<tr>
<th></th>
<th>User 1</th>
<th>User 2</th>
<th>User 3</th>
<th>User 4</th>
<th>User 5</th>
<th>User 6</th>
<th>User 7</th>
<th>User 8</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Total:** [Total]
7.8.2.1 How to Change the Default PDF Template

To change the default PDF template, simply click on Show PDF report templates dialog and specify:

- Name of the template,
- First page header,
- First page footer,
- Other pages’ header,
- Other pages’ footer (For headers and footers, you need to upload images. The recommended image size is 1305*150 and higher),
- Page number format and its font size and color,
- Font size and color of the PDF report title,
- Color for the separator (line after the PDF report title and between submissions in one report).
Fill out the information you want to be displayed and click **Save**. Once saved, the new template will appear to the left in the Templates list. If you want to change it, select it from the list, make the changes, and click **Save**.

To select this template for your form, open the form settings → Report common settings → PDF report template → choose the template you need.
To make any changes to the template, go back to the dialog, select it from the list, make the changes and click Save. To add a new template, click Add.

Here is an example of a PDF template we have created and its comparison with the default PDF template.
PDF templates:
1. **Bed and Breakfast**

   - Name: **Bed and Breakfast**
   - **Show header on first page:**
   - **Show footer on first page:**
   - **Show header on other pages:**
   - **Show footer on other pages:**
   - **Show page number in footer:**
     - **Show page number for first page:**
     - **Page number format:**
     - **Page number size:**
     - **Page number color:**
   - **Title size:**
   - **Title color:**
   - **Separator color:**
   - **Layout for first level fields:**
   - **Layout for sub fields:**

   [Upload]  [Download]
   [Upload]  [Download]
   [Upload]
   [Upload]
   [Upload]
   [Upload]
   [Upload]
   [Upload]
   [Upload]

   [New]  [Save]  [Copy]  [Delete]
King Size Rooms Report

RoomNumber: 202
Photo:

Description: This room features much original art, a French beveled mirror, antique writing desk, private bath, king size British Period four poster, canopied bed, original marbled working fireplace, a twin pull-out sofa bed, an over-sized bath with a corner Jacuzzi for two, and a private balcony overlooking Chippewa square.
7.8.2.2 How to Create a Custom PDF Report/Template

You can create custom PDF templates by using tools like PDFescape or Adobe Acrobat Pro. Take any PDF form you need and turn it into a fillable one. Here is a great video tutorial on how to do so.

Make sure to name each field by clicking on the field with the right click of the mouse and selecting Object Properties.
The names of the fields should correspond to the field names inside the digital form. After you have named each fillable field, downloaded this PDF file onto your computer, and then uploaded it into your form by clicking on Upload, you need to click on Change to match the fields from the PDF template you have created to the field names inside your form. (You can slightly change the names of the fillable fields: add spaces, change the capitalization, etc. However, make sure that their names overall match the names of the fields inside your form so that you have no hard time recognizing and matching them in Template Fields Mapping.)

This is the fillable PDF template we use in our Construction Daily Log app and the way we have configured it.
This is the data source.

This is an example of one field’s name inside the form.
These are the matching fields.
7.8.2.2.1 ADDING A TABLE IN A CUSTOM PDF REPORT/TEMPLATE

There are several rules you need to follow while naming field elements connected to a table.

Let us look at the Construction Daily Log App, which utilizes the following custom PDF template:
As you can see, this table contains a number of fields, and we need to connect each of them to the corresponding fields in our app so that the filled out information in the app gets displayed in our custom PDF report correctly.

The table consists of three major columns: LABOR, SUBCONTRACT и RENTALS. In the first column, there are 4 fields: NAME, LABOR, WORK HRS, TRAVEL HRS.
In the second column, we have: NAME, SUBCONTRACT, WORK HRS, TRAVEL HRS.

And in the third column, we see: EQUIPMENT, HRS, COST/HOUR, TOTAL COST.

Here is how these elements look in our app.
**Labor Column/Table**

![Labor Column/Table](image1)

**Subcontract Column/Table**

![Subcontract Column/Table](image2)

**Rentals Column/Table**

![Rentals Column/Table](image3)
To make sure you can see the names of the fields better, we have retyped them in the following form:

<table>
<thead>
<tr>
<th>#</th>
<th>NAME</th>
<th>LABOR</th>
<th>WORK HOURS</th>
<th>TRAVEL HOURS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>table_namelabor_1</td>
<td>table_LABOR_1</td>
<td>table_WORK</td>
<td>table_TRAVEL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HRS_Labor_1</td>
<td>HRS_Labor_1</td>
</tr>
</tbody>
</table>

**Labor Column/Table**
<table>
<thead>
<tr>
<th></th>
<th>table_namelabor_2</th>
<th>table_LABOR_2</th>
<th>table_WORK HRS_Labor_2</th>
<th>table_TRAVEL HRS_Labor_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>table_namelabor_3</td>
<td>table_LABOR_3</td>
<td>table_WORK HRS_Labor_3</td>
<td>table_TRAVEL HRS_Labor_3</td>
</tr>
<tr>
<td>4</td>
<td>table_namelabor_4</td>
<td>table_LABOR_4</td>
<td>table_WORK HRS_Labor_4</td>
<td>table_TRAVEL HRS_Labor_4</td>
</tr>
<tr>
<td>5</td>
<td>table_namelabor_5</td>
<td>table_LABOR_5</td>
<td>table_WORK HRS_Labor_5</td>
<td>table_TRAVEL HRS_Labor_5</td>
</tr>
<tr>
<td>6</td>
<td>table_namelabor_6</td>
<td>table_LABOR_6</td>
<td>table_WORK HRS_Labor_6</td>
<td>table_TRAVEL HRS_Labor_6</td>
</tr>
<tr>
<td>7</td>
<td>table_namelabor_7</td>
<td>table_LABOR_7</td>
<td>table_WORK HRS_Labor_7</td>
<td>table_TRAVEL HRS_Labor_7</td>
</tr>
<tr>
<td>8</td>
<td>table_namelabor_8</td>
<td>table_LABOR_8</td>
<td>table_WORK HRS_Labor_8</td>
<td>table_TRAVEL HRS_Labor_8</td>
</tr>
<tr>
<td>9</td>
<td>table_namelabor_9</td>
<td>table_LABOR_9</td>
<td>table_WORK HRS_Labor_9</td>
<td>table_TRAVEL HRS_Labor_9</td>
</tr>
<tr>
<td>10</td>
<td>table_namelabor_10</td>
<td>table_LABOR_10</td>
<td>table_WORK HRS_Labor_10</td>
<td>table_TRAVEL HRS_Labor_10</td>
</tr>
<tr>
<td>11</td>
<td>table_namelabor_11</td>
<td>table_LABOR_11</td>
<td>table_WORK HRS_Labor_11</td>
<td>table_TRAVEL HRS_Labor_11</td>
</tr>
<tr>
<td>12</td>
<td>table_namelabor_12</td>
<td>table_LABOR_12</td>
<td>table_WORK HRS_Labor_12</td>
<td>table_TRAVEL HRS_Labor_12</td>
</tr>
<tr>
<td>13</td>
<td>table_namelabor_13</td>
<td>table_LABOR_13</td>
<td>table_WORK HRS_Labor_13</td>
<td>table_TRAVEL HRS_Labor_13</td>
</tr>
<tr>
<td>14</td>
<td>table_namelabor_14</td>
<td>table_LABOR_14</td>
<td>table_WORK HRS_Labor_14</td>
<td>table_TRAVEL HRS_Labor_14</td>
</tr>
<tr>
<td>15</td>
<td>table_namelabor_15</td>
<td>table_LABOR_15</td>
<td>table_WORK HRS_Labor_15</td>
<td>table_TRAVEL HRS_Labor_15</td>
</tr>
</tbody>
</table>

Subcontract Column/Table
<table>
<thead>
<tr>
<th>#</th>
<th>NAME</th>
<th>SUBCONTRACT</th>
<th>WORK HOURS</th>
<th>TRAVEL HOURS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>table_NAME_subcontract_1</td>
<td>table_subcontract_1</td>
<td>table_WORK HRS_subcontract_1</td>
<td>table_TRAVEL HRS_1</td>
</tr>
<tr>
<td>2</td>
<td>table_NAME_subcontract_2</td>
<td>table_subcontract_2</td>
<td>table_WORK HRS_subcontract_2</td>
<td>table_TRAVEL HRS_2</td>
</tr>
<tr>
<td>3</td>
<td>table_NAME_subcontract_3</td>
<td>table_subcontract_3</td>
<td>table_WORK HRS_subcontract_3</td>
<td>table_TRAVEL HRS_3</td>
</tr>
<tr>
<td>4</td>
<td>table_NAME_subcontract_4</td>
<td>table_subcontract_4</td>
<td>table_WORK HRS_subcontract_4</td>
<td>table_TRAVEL HRS_4</td>
</tr>
<tr>
<td>5</td>
<td>table_NAME_subcontract_5</td>
<td>table_subcontract_5</td>
<td>table_WORK HRS_subcontract_5</td>
<td>table_TRAVEL HRS_5</td>
</tr>
<tr>
<td>6</td>
<td>table_NAME_subcontract_6</td>
<td>table_subcontract_6</td>
<td>table_WORK HRS_subcontract_6</td>
<td>table_TRAVEL HRS_6</td>
</tr>
<tr>
<td>7</td>
<td>table_NAME_subcontract_7</td>
<td>table_subcontract_7</td>
<td>table_WORK HRS_subcontract_7</td>
<td>table_TRAVEL HRS_7</td>
</tr>
<tr>
<td>8</td>
<td>table_NAME_subcontract_8</td>
<td>table_subcontract_8</td>
<td>table_WORK HRS_subcontract_8</td>
<td>table_TRAVEL HRS_8</td>
</tr>
<tr>
<td>9</td>
<td>table_NAME_subcontract_9</td>
<td>table_subcontract_9</td>
<td>table_WORK HRS_subcontract_9</td>
<td>table_TRAVEL HRS_9</td>
</tr>
<tr>
<td>#</td>
<td>NAME</td>
<td>HOURS</td>
<td>COST</td>
<td>TOTAL COST</td>
</tr>
<tr>
<td>----</td>
<td>-----------------------</td>
<td>-------------------</td>
<td>--------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>1</td>
<td>table_equipment_1</td>
<td>table_HRS_rentals_1</td>
<td>table_cost_hour_1</td>
<td>table_total_cost_1</td>
</tr>
<tr>
<td>2</td>
<td>table_equipment_2</td>
<td>table_HRS_rentals_2</td>
<td>table_cost_hour_2</td>
<td>table_total_cost_2</td>
</tr>
<tr>
<td>3</td>
<td>table_equipment_3</td>
<td>table_HRS_rentals_3</td>
<td>table_cost_hour_3</td>
<td>table_total_cost_3</td>
</tr>
<tr>
<td>4</td>
<td>table_equipment_4</td>
<td>table_HRS_rentals_4</td>
<td>table_cost_hour_4</td>
<td>table_total_cost_4</td>
</tr>
<tr>
<td>5</td>
<td>table_equipment_5</td>
<td>table_HRS_rentals_5</td>
<td>table_cost_hour_5</td>
<td>table_total_cost_5</td>
</tr>
<tr>
<td>6</td>
<td>table_equipment_6</td>
<td>table_HRS_rentals_6</td>
<td>table_cost_hour_6</td>
<td>table_total_cost_6</td>
</tr>
<tr>
<td>7</td>
<td>table_equipment_7</td>
<td>table_HRS_rentals_7</td>
<td>table_cost_hour_7</td>
<td>table_total_cost_7</td>
</tr>
<tr>
<td>8</td>
<td>table_equipment_8</td>
<td>table_HRS_rentals_8</td>
<td>table_cost_hour_8</td>
<td>table_total_cost_8</td>
</tr>
<tr>
<td>9</td>
<td>table_equipment_9</td>
<td>table_HRS_rentals_9</td>
<td>table_cost_hour_9</td>
<td>table_total_cost_9</td>
</tr>
<tr>
<td></td>
<td>table_equipment</td>
<td>table_HRS_rentals</td>
<td>table_cost_hour</td>
<td>table_total_cost</td>
</tr>
<tr>
<td>---</td>
<td>-----------------</td>
<td>-------------------</td>
<td>-----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Here are the names of the fields with the fields inside our digital form they correspond to.

![Back](image1)

![Back](image2)
As you can see, while naming field elements connected to a table, you have to add the word *table* at the beginning of each name and its corresponding row number at the end. 

Thus, for instance, for the Labor column, the Name field, we have the name *table_namelabor_1*, where 1 is the number of the first row of the table. What this means is that the first entry will fall into this field; the second – the field named *table_namelabor_2*; the third – the field under the name *table_namelabor_3*, etc. We number our fields in this way till our last row. (In our case, it is *table_namelabor_20*.)

You can name your fields in any way as long as you remember which field in your custom PDF template matches which field in the digital form. Here is the formula to follow while naming your fields:

```
table_Your Field Name_n, *
```

where *n* is the row number.

*alternatively, you can omit the underscore before the number: *table_Your Field Name n*

However, we recommend that you keep it since it makes it easier to visually separate the number from the rest of the name.

After you have finished naming the fields and saved this fillable PDF file, open the settings of your
Form button and, under *Report Common Settings*, put a check mark next to *Use custom PDF template* and then click *Upload* to upload the fillable PDF file you have just created. After the file has been uploaded, click *Change*.

Here you need to match the fields from the fillable file to the fields in the digital form.
Let us look only at the fields connected to the table. On the left-hand-side, you see the fields from your fillable PDF file; on the right-hand-side, you can see the fields from the app (digital form). Let us connect the Labor: Name field to `table_namelabor_1`. (Here you can see only the first row, but this is enough for the app to recognize the rest.) Connect the rest of the fields accordingly.

The infographic below illustrates how entered information will fall into our custom PDF report.
Let us see how this works on a mobile device.
In the screenshot of our custom PDF report, you can see that our entered information has correctly been placed into the corresponding fields in the reports.
A word of caution: the table size limits our ability to enter long, multiline descriptions since it cannot match the size of the entries and always stays the same. Also, if there are more entries than, let us say 20, like in our case, the entries numbered 21 and higher won’t be reflected in our custom PDF report. That is why, while deciding whether to use the default or custom PDF report options, take into consideration this limitation. In some cases, you might be better off choosing the default PDF report, which doesn’t look as nice as the custom PDF report, but allows more flexibility.

Here are a few additional tips:

- Instead of inserting an entire picture in your PDF report, you can substitute it for the link to this image. Once a user clicks on the link inside the PDF report, the full picture will open. This can significantly save you space in the PDF report.

To enable this option, in the settings of the image element of your form, simply put a check mark next to Show URL instead of image (for PDF report).

- PDF fields connected to the Table Selection button should start with "tableSelection_" or simply “table_” and end with the row number (>= 1), like the ones connected to a table.
(see the description above). This option can be used only when the table selection is configured as Frame View.

**7.8.2.3 How to Create a Custom Excel Report/Templa**

If the default Excel format does not work for you, you can always create a custom one. The custom Excel report can be laid out and formatted in any way you want. You can also use different fonts and colors, add logos and leverage other tools to make it look even more professional.

When you have configured your Excel report the way you need, upload it into the app by putting a check mark next to *Custom Excel template* and clicking *Upload*.

As soon as the report has been uploaded, you will need to match input fields in your form to the correct cells in your custom Excel report. There are two ways to do so:

1. **Using Cell Address**
   
   In Excel, each cell has its address, which is a combination of a letter and a number that specifies the column and row in which the cell is located on a spreadsheet (A1, B2, C3 etc.). You can use cell addresses to connect your Excel report to your mobile form. In order to do that,

   1. Upload your custom Excel report
   2. As soon as it has been uploaded, you will see the following window:

```
<table>
<thead>
<tr>
<th>Excel Report</th>
<th>Universal Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter cell address</td>
<td>Select an input field</td>
</tr>
</tbody>
</table>
```

   - Enter the address of the cell in the left-hand column;
   - Select the corresponding input field from the right-hand column
- Click the "+" (plus) button to add a new line
- Use the "-" (minus) button to remove the incorrect line / the lines you don't need anymore;

Notice! You can use different tabs (sheets) in your Excel Report for data collection. If you use tabs, the cell address should look as follows: "Sheet name"! "Cell address", e.g. Sheet1!A1

2. Using Cell Names
As soon as you have defined the cell the information from your form will go to, you can name this field and use this name to connect your Excel sheet to your form. When naming cells, please stick to the following format: <cellName>

Take a look at the image below. It illustrates some examples of cell names (column B) and their labels (column A).

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Name</td>
<td>&lt;1st name&gt;</td>
</tr>
<tr>
<td>Last Name</td>
<td>&lt;2nd name&gt;</td>
</tr>
<tr>
<td>Address</td>
<td>&lt;address&gt;</td>
</tr>
<tr>
<td>Phone #</td>
<td>&lt;phone&gt;</td>
</tr>
<tr>
<td>Email Address</td>
<td>&lt;email&gt;</td>
</tr>
</tbody>
</table>

Notice! A spreadsheet cannot contain duplicate cell names.

When you are done with the spreadsheet and the cell names, save this Excel file and upload it in the Report Common Settings of your form. As soon as it has been uploaded, you will see the following window:
- You will see all the cell names located in your Excel form in the left-hand column; -
  You will only need to select the corresponding input fields from the right-hand column.
- You can still utilize cell address for connecting some "extra-fields";

As a result, you will receive a report with the output in the Excel format similar to this:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>1</td>
<td><strong>First Name</strong></td>
</tr>
<tr>
<td>2</td>
<td><strong>Last Name</strong></td>
</tr>
<tr>
<td>3</td>
<td><strong>Address</strong></td>
</tr>
<tr>
<td>4</td>
<td><strong>Phone #</strong></td>
</tr>
<tr>
<td>5</td>
<td><strong>Email Address</strong></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

7.8.3 Auto-Fill Options

In the Auto-Fill Options section, specify which, if any, fields of your digital form need to be automatically filled out and what data source this information will be taken from. If you opt for enabling fields mapping, click on `Fields Mapping` and match the fields from the form to the fields from the corresponding data source by hitting the Add button and choosing the matching fields from the drop-down menu.

For instance, for my construction daily log form, I need the following fields to be autopopulated.
Under **Sorting Filter**, you can specify in what order the information from the chosen fields will auto-populate the selected fields. For example, I would like to make sure that the most recent information auto fills my fields. Then I choose *desc* (descending) since in our data source the most recent entries occupy the lowest rows.

Under **Search Settings**, you can specify some criteria that the previously entered information needs to satisfy to auto populate your fields. For instance, I only need the information from those reports which I sent to my email address and only when I was logged in under the same email address.
In my digital form, the email field is used for sending each completed report to (see the first screenshot below), and the formula `<user.emailAddress>` indicates the email address under which the user is logged in.

The screenshot above (to the right) shows that every time I open the New Daily Report Form, the Project Name, Contractor, State, and Address fields get auto-populated with the corresponding information from the most recent reports. Therefore, I don’t need to fill out the same fields again and again while working on the same project. However, such fields as Day Number, Weather, and Temperature do not need to be auto-populated as this information will be different each time.
Here is another example of using the Auto-Fill options. You have a list of Companies in one data source called *Companies* and a list of documents in another data source called *Documents*. You add the Form button and connect it to the *Companies* data source in the Submit to Data Source section of your Form button. Then, in the Auto-Fill Options section of your form, you enable fields-mapping and choose the *Documents* data source. In this way, once your users select a company of interest, the fields will be auto-populated with the data from the *Documents* data source. As usual, make sure to accurately match the fields from the *Documents* data source to the fields inside the form.

### 7.8.4 Submit to Data Source

To ensure that the information our app users enter into the digital form gets saved and can be shared, edited, accessed, deleted later on, we need to connect the Form button to the previously created data source with all necessary fields and specify whether this information will get added every time or updated. All these need to be done in the Submit to Data Source section. (Alternatively, you might want to simply generate Excel, PDF, and Word reports out of filled out information and not save and store it anywhere. Then you need to leave the box next to *Add to datasource* unchecked.)

![Submit to Data Source](image)

If you opt for updating previously entered information, you will also need to specify which line in the connected data source will be updated: first or last.
First Line Updated

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>notesid</td>
<td>photosid</td>
<td>locationsid</td>
<td>signature</td>
<td>grid</td>
<td>name</td>
<td>email</td>
<td>date</td>
<td>email2</td>
</tr>
<tr>
<td>2</td>
<td>330F6A8A FC320B86-1027CE69-C938B266-22009211-Elm St Project</td>
<td>yulia gar</td>
<td>2016-06-2</td>
<td>yulia gar</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>F95BC74C:994EC5C1-B55AEAD3D2C88788886A53EFDB</td>
<td>Report 5</td>
<td>2015-11-0</td>
<td>prototype</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Last Line Updated

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>72</td>
<td>bef9fd9-bea9fd9-be9fd9-be9fd9-be9fd9-Emily</td>
<td>yulia gar</td>
<td>2016-06-2</td>
<td>yulia gar</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>0bcd442-0bcd442-0bcd442-0bcd442-0bcd442-Elm St Proy</td>
<td>yulia gar</td>
<td>2016-06-2</td>
<td>yulia gar</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

You might also need to put a checkmark next to Enable Multiline Table Input to send several element from the Table element into the connected data source if the Table element has been added to the digital form.
The Command Mode is used only in those apps that use web connectors and only for custommade apps.

Here is how the Submit to Data Source section of our Newly Daily Report Form button is set up.

7.8.4.1 How to Use Relationships in Digital Forms and Make Forms inside Forms

As described in the Chapter about Data Sources and Data Connectors, there are three types of relationships between data sources: one to one, one to many, and many to many. Most of the times, we use the one-to-many relationship. This relationship means that one entry from the main data source (Parent) corresponds to one or several entries in a secondary data source (child). We connect our Form button to the parent data source, while the secondary data sources (child/children) are linked to Table and Multiline Entry elements, which allow creating digital forms within the main digital form. Almost every our app utilizes these options.

Below we are going to create a simple hotel inspection app to illustrate how data relationships work with the Form button and how to make forms inside forms. These are the steps to follow:

1. Create three Excel spreadsheets:
   - The first will be called Hotel Inspection and will serve as the parent data source. It will contain the following fields: Name, Date, Hotel_Name, Address, Phone, ID, and ID1.
• The second one will be called Rooms and will serve as a child data source. It will allow inspectors to check and record several rooms. (Hence, for each single hotel any given inspector will be able to take notes on any number of rooms; this is why here we need to use the one-to-many relationship.) It will contain the following fields: ID, Number, Room_Cleanliness, and Bedclothes_Cleanliness. (Here you can see that both of these Excel spreadsheets contain one identical field – ID. This field is essential and will be used to connect the parent data source to the 1st child one.) Naturally, many more fields could be added here. This is just an example.

• The third one will be called Rooms and will serve as a child data source. It will allow inspectors to take as many room photos as necessary. (Hence, for each single hotel any given inspector will be able to take photos of as many rooms as he/she needs; this is why here we need to use the one-to-many relationship.) It will contain the following fields: ID1 and Photo. (As you can see, the parent data source and the 2nd child data source have one identical field – ID1, which will be used to bind them together.)

2. Upload these three Excel spreadsheets in the Data Source Dialog, name them, and configure the data types. Choose “Date” for the Date field, “Numeric” for the (hotel) Number field, and “Image” for the Photo field. Make sure that the ID and ID1 fields are marked as “Text”. Save each of these data sources.
3. Now we need to set up the relationships between these three data sources. First, select the parent data source (Hotel Inspection) and click on Manage Relationships. Choose One to Many as a relationship type, Rooms as the related child data source, and ID for each of the following fields.
Click Add to see that we have created our first one-to-many relationship, which is displayed in the Manage Relationships section of both our parent and child data sources.

Select the parent data source again and now connect it to the 2nd child data source accordingly.
This time the ID1 field is used to bind both of these data sources. Click Add and save the changes.
4. Now that we have configured and saved the data sources and the relationships between them, we need to create a digital form. Start by adding the Form button onto any tab you want and then, under the Submit to Data Source section, select Hotel Inspection as the parent data source. Save the settings by clicking OK and then open the Form button.
Now we need to add 4 Text fields and one Datetime field and connect them to our parent data source via Select Field.

Here is how we configured each of these elements.
Inspector’s Name Caption

The quick brown fox jumps over the lazy dog.

Inspector’s Name Text Field

The quick brown fox jumps over the lazy dog.
The quick brown fox jumps over the lazy dog.
Hotel’s Name Caption

The quick brown fox jumps over the l...
Address Caption

Address Text Field
Phone Number Caption

Phone Number Text Field
Now let us add one Table element and connect it to our first child data source (Rooms) to enable our users to add records of as many rooms as they need into digital inspection reports.

In the Settings of the Table element, enter Room Inspection as its name and choose Rooms as the data source. Click OK and then open the Table element to add the fields reflecting the content of the Rooms data source.
Make sure to connect each of these fields to the corresponding fields in this data source.

<table>
<thead>
<tr>
<th>Name</th>
<th>Room Inspection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Room Inspection</td>
</tr>
<tr>
<td>Web Parameter name</td>
<td>TableFields</td>
</tr>
<tr>
<td>Height</td>
<td>50</td>
</tr>
<tr>
<td>Width</td>
<td>225</td>
</tr>
<tr>
<td>Order</td>
<td></td>
</tr>
<tr>
<td>Next order</td>
<td></td>
</tr>
<tr>
<td>List View template</td>
<td>Custom</td>
</tr>
</tbody>
</table>

Select Data Source: Rooms

View in PDF report: Default

Viewing Conditions

Make sure to connect each of these fields to the corresponding fields in this data source.

Room Number
- Room Number

Room Cleanliness
- Room Cleanliness

Bedclothes Cleanliness
- Bedclothes Cleanliness

Here is how we configured each of these elements.
Room Cleanliness Text Field

The quick brown fox jumps over the lazy dog.

Bedclothes Cleanliness Caption

The quick brown fox jumps over the lazy dog.
Here is how the ready Detail View of our Table element looks.
You might also want to customize the cell view of the Table element. To do so, in the settings of the Table element, click on Change, select Custom View, and enter the size you need. Click OK.

Choose Cell View options
Or customize it in Editor

- Custom View
- Standard view

Cell Height 100

Then select Cell View, and in the space provided, enter any number of the Text elements to pull the information from our child data source (Rooms). Here are the ones we entered and how we configured each of them.
Room Number Text Element

Room Cleanliness Text Element
Now that we have fully configured the Table element, let us add and set up one Multiline Entry element to add the photos of inspected rooms. To do so, drag and drop the Multiline Entry element onto our digital form below the Table element and then, in the popped up window, name it and select *Photos* as the data source. Click *OK*.
Here is how our digital form looks with the Multiline Entry element.

Now open the Multiline Entry element to add one Text element and one Photo field to enable our users to take and upload photos to integrate them into inspection reports. Make sure to connect the Photo field with the Photos data source.
Here is how we configured each of these elements.
Last but not least, add one Submit button to enable the submission of new data.

Our main form with two forms within it is now ready. Let us fill it out and submit it.
Here is how the submitted data looks in the default PDF report.

Inspector's Name: Max Weber
Date: 07/03/2016
Hotel's Name: Days Inn
Address: 21 Harrison St, Orlando, Florida
Phone Number: 1010000000

Room Inspection 1:
Room Number: 101
Room Cleanliness: Excellent
Bedding Cleanliness: Good
Room Photos 1:
Photo:

Room Photos 2:
Photo:
If we download the generated Excel report as well, we can see that the same information got sent into it. However, our platform has also generated a number of letters and digits for the ID and ID1 fields, and these unique combinations are the same for all these three data sources.

Moreover, for each new report, a new unique letter and number combination gets assigned and sent into the ID and ID1 fields of these three data sources.

5. You might also want to add one List button and connect it to our parent data source to store, access, and edit all the previously submitted inspection reports.

Here you might also want to configure the Cell View and Detail View of your List button by clicking on Change next to each view type and selecting an optimal variant.
Then you will need to add any number of fields connected to our parent data source and configure them in any way you need.

Here is how our Cell View and Detail View look in the editor.
Here is how they look on a mobile device.
7.7.4.2 Multiple Table Input

In most of our apps, we have one parent data source and several secondary (child) data sources for which we add the Table element most of the times. However, with the Multiple Table Input, we can add the Table element for the parent data source within the Form button itself (also connected to the parent data source).

Let us take the same hotel inspection app we described above. However, this time, we are going to bind the Table element located within the main form with the parent data source (Hotel Inspection), not the 1st child data source (Rooms).

And then we are going to add all the fields located in the Hotel Inspection data source. Another Table element linked to the 1st child data source (Rooms) will be located within the first Table element, and the Multiple Entry element connected to the 2nd child data source (Photos) will also be located within the first Table element. (The settings of all the fields inside the first Table element, of the 2nd Table element, and of the Multiple Entry element with their fields are the same
as the ones described in How to Use Relationships in Digital Forms and Make Forms inside Forms.

The screenshot below illustrates how the Multiple Table Input allows recording and submitting multiple lines of entry at once. In our case, we can enter the records on several hotels together with detailed description of each room and room photos and submit these all at once into a single PDF/Excel report.
Hotel Inspection 11:
Inspector’s Name: Max Weber
Date: 07/03/2016
Hotel’s Name: Days Inn
Address: 21 Harrison St, Orlando, Florida
Phone Number: 101-000-0000

Room Inspection 1:
Room Number: 101
Room Cleanliness: Excellent
Bedclothes Cleanliness: Good

Room Inspection 2:
Room Number: 205
Room Cleanliness: Good
Bedclothes Cleanliness: Excellent

Room Inspection 3:
Room Number: 202
Room Cleanliness: Satisfactory
Bedclothes Cleanliness: Good

Room Photo 1:
Photo:

Room Photo 2:
7.7.5 Send Report via Email

After we have configured the report settings in the Report Common Settings section, we need to specify the addressees, the subject line, report format, after how many submissions each report will be sent, and whether different report formats should be sent in one message or separately. All these can be done in the Send Reports via E-mail section.
Here is how we configured this section of our *Newly Daily Report Form* button:

<table>
<thead>
<tr>
<th>Email Mode</th>
<th>Email addresses (one per line)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><a href="mailto:sample@sample.com">sample@sample.com</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subject</th>
<th>Data collected for your app</th>
</tr>
</thead>
</table>

| Send data after how many submissions | 1 |

<table>
<thead>
<tr>
<th>Report format</th>
<th>PDF, Spreadsheet, Word, Email body</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Send reports in one message</td>
</tr>
</tbody>
</table>

and here is how our reports look as a result.
7.7.5.1 Ordering the Sequence of Form Elements/Fields in Reports

Binding the elements inside a specific form with the fields of the connected data source is an essential step of ensuring that submitted information gets saved and displayed correctly. However, making sure that the existing in your digital form elements and fields get displayed in the order intended, please enter the corresponding numbers next to the order field.
The quick brown fox jumps over the l...
From the screenshots above and below, you can see how we use the order field in the settings of the form elements to order the sequence of the elements inside a digital form for reports.
7.7.5.2 Scheduled Reporting

You can further manage how and when you send and receive reports. To do so, open the Data Source Dialog, select the data source from which the app sends reports, and click Manage AutoReporting.
Click *Add new report*.

You will see the following dialog:
Here you can specify your report name, schedule recurrence (daily/weekly/monthly), time, email address the reports will be sent to, subject line, message, report file name, and data you want to be submitted in the report by clicking *Change* and selecting the corresponding fields.
You can also set up the searching filter to send the reports based on a specified selection. For example, you can specify the dates you want the reports to be received from.

*Use Case:* A construction manager receives many reports on a daily basis from his crew members. Instead of receiving these reports on a daily basis, he can schedule them to be sent on a weekly basis in one file.

### 7.7.6 Submit Data to REST API (Web post)

The Submit Data to REST API (Web Post) section is used for retrieving information from websites based on some specified criteria. Thus, for instance, your app users can get weather forecasts for their areas by entering their zip codes.

![Submit data to REST API (Web post)](image)

Here is what you would need to do to enable this functionality.

A. Put a checkmark next to *Web Post Mode*.
B. Next to *Method*, from the drop-down menu, select *Get*.
C. Next to *Response Type*, from the drop-down menu, select *html*.
D. Next to *Web Server Url*, enter the URL of the website you are planning to get information from and put a question mark at the end of the link. Here’s what we entered, for example: [http://www.wtsp.com/weather/local-weather.aspx](http://www.wtsp.com/weather/local-weather.aspx)? E. Click *OK*. 
F. Under *Fields and Buttons*, select and drag onto the canvas the Text/Number field.
G. In the popped up window, next to Web Parameter name, enter the word Location.

H. Next to Data type, from the drop-down menu, select number.

I. Next to Placeholder, enter the word ZIP Code.

J. Adjust the width of the text field if necessary. Click OK.
K. Under *Fields and Buttons*, select the Submit button and drag it onto the canvas.

L. Adjust the size and change the icon of the Submit button if necessary. Click *OK*. Save the app.

M. Now you can open this application in Snappii’s Preview app, enter any zip code, and get the weather forecast for that area.
7.7.7 Upload Reports to Cloud Drives

In the Upload Reports to Cloud Drives, you can specify where you would like to send and store the copies of your filled out information. If you have linked your cloud storage account(s) to the app you are building, you can select this account in the drop-down menu next to Storage. Here you can also specify the format you would like this information to be saved and stored in your cloud storage account.

7.7.8 Share Form Report

Make sure to put a checkmark next to Allow users to share reports in the Share Form Report section so that Excel and PDF reports generated upon the completion and submission of each form get added to the Report List tab/button and can be viewed and shared later on.
Here you can decide what will happen every time your app users submit a new form: 1) they will be able to view the generated reports and then share them, if necessary; 2) the generated reports will simply get saved; 3) the users will be able to choose whether to view and save, save, or do neither; or 4) they will be asked to choose from the options offered by the Boolean element inside the digital form.
For the third option, you can also change the text that your users will see when they are offered the three possible options. For the forth option, you will need to configure the value of the Boolean choices.

View and Save
By User’s Choice
7.7.9 Configure Your Element
In the Configure Your Element section, you can specify what time zone will appear in the generated reports and choose what will happen after each form gets submitted: 1) it will be closed, 2) all the filled out fields will be cleared, or 3) it will be reset. Here you can also put checkmarks next to Show Submit button, Show Reset button, and Show Clear button to see them appear at the top of the screen.

7.7.9.1 Invisible Form Mode
The Invisible Form Mode option allows your app users to submit a digital form without opening it.
Let us look at an example. We are going to create an app where our users will be able to check off books from the given list as soon as they have finished reading them and where they will see a specified image next to each book mark as read. Here are the steps to follow to enable this capability.

1. Create an Excel spreadsheet with the following fields: Name for the name of each book, Author for the name of each author, and Already_Read for the status of each book. The default status will be a no.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>name</td>
<td>author</td>
<td>already_read</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>book1</td>
<td>author1</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>book2</td>
<td>author2</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>book3</td>
<td>author3</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>book4</td>
<td>author4</td>
<td>no</td>
<td></td>
</tr>
</tbody>
</table>

2. Upload this spreadsheet in the Data Source Dialog, name, and save this data source.

3. Now add the List button onto any tab you like and, in the settings of the List button, connect it to the data source we have just created. Click on Change to change the List View template to the custom one and enter your size of the List View. Click OK and then again OK.
List View

Select Data Source: or choose Books

List View Title: ListView
Detail View Title: DetailView
Empty dataset message: Dataset contains no items to show

List View template: Custom
Detail View template: Custom

Choose template and assign mapping for View:

Or customize it in Editor: Custom View

Cell Height: 100
4. Click on your List button and then, in a new window, on Cell View.

5. Add 5 Text buttons: one for the Book caption, one for the name of each book taken from our data source, one for the Author caption, one for the name of each author taken from our data source, and one for the message “Click if you have already read”.

Here are the settings of each of these elements.

The Book Caption
The Name of Each Book from Our Data Source

(Connected to the Name Field from Our Excel Spreadsheet)
The Author Caption

The quick brown fox jumps over the.

The Author of Each Book from Our Data Source

(Connected to the Author Field from Our Excel Spreadsheet)
6. Add one Image element and change its image to any picture symbolizing the end of a task by clicking on Change image.
We want this image to appear only after we specify that we have read one of the books from the list by clicking on the Click button (see below). To do so, we put a check mark next to Use formula to show/hide element and selected Use formula to show element. Next to Formula, we entered \(<\text{read}> = = \text{"yes"}\) (We will explain the meaning of this formula below).

7. Add one Form button and, in the popped up window, name it, specify its dimensions, and select your button style. Here is how we have configured the View Section of our Form button.
In the Submit to Data Source section, put a check mark next to *Use Data Source*, select our data source, and then, next to *Submission type*, choose *Update*.

![Submit to Data Source section]

Most importantly, in the Configure Your Element section, put a check mark next to *Use invisible form mode*. Click OK.

![Configure your element section]

8. Click on the Form button we have just created and drag one Text Formula field onto it.
In the settings of the Text Formula field, next to Formula, enter “yes” as the constant value of this text field and, next to Select Field, choose already_read as the field in our data source where the status of our books from the list will get updated from a “no” to a “yes” every time a user clicks on our Form button labeled as Click.
9. Go back to the List View of our List button and add another Text Formula field.
In the settings of the field, name the field (in our case it is *read*), enter `<current.already_read>` next to *Formula* to show us the current status of each book, and put a check mark next to *Visible* to make this formula field visible.
10. Now, if you open the Viewing Conditions subsection of the Image element we created above, you will understand that the visibility of this Image element will depend on the value of our Text Formula field on the List View screen. If \(<\text{read}> = \text{“no”}\), the Image element will be hidden, but after we click on our Form button, thus automatically updating the status of the given book in the already_read field from “no” to “yes”, the value of the Text Formula field will change from a “no” to a “yes” and will satisfy our condition specified in the Viewing Options of the Image element.
Now let us see how all the changes we have made work on a mobile device. When we open our app for the first time, we see that the status of each book is set as a “no” since we haven’t marked any of the books as Read yet. We also don’t see the image next to any of the books.
Now let us press on Click next to the first book to see the status change from a “no” to a “yes” and the image appear.

Now let us mark the second book as read as well.
We can see that everything works as we intended.

Now let us open the settings of the Text Formula field on the List View screen and remove the check mark next to Visible to hide the formula field.
We will see that our Text Formula field has disappeared. However, if you mark another book as read, our settings will still work regardless of the visibility of the formula field.

The quick brown fox jumps over the la..
This is all you need to do to take advantage of the Invisible Form Mode. Another common way to use it is to make a list of purchases and mark each item as purchased whenever needed by clicking on the Form button labeled as “Purchased”. In fact, the Invisible Form Mode offers a limitless number of options and can be quite useful in a variety of situations.

7.7.10 Messages and Labels

11. In the Messages and Labels section, you can specify whether to show system messages upon the successful submission of each form or not and what these messages will be. Here you can also specify the Submit button label.
7.7.11 Paging
In the Paging section, if your form is too long to fit into your users’ screen sizes, you can opt for splitting your digital form into several pages by clicking on the Manage button. The Manage button allows you to add new pages, edit, reorder, and delete them.
“+” - use it to add a new page

“-“ – use it to delete a page

“Edit” – use it to change a page name. (Please keep in mind the title of the Form button will be used as the name of the first page.)

“Up” “Down” - use them to change the order of pages

“Clear” – use it to delete all pages at once

In the editor, you will see all the pages after a click on the Form button. You can tap on any of the pages to drag and drop any elements you will need to add to them. Please see the screenshot below (the one to the left).

If you enable pagination in your app, we recommend that, in addition to the default scroll bar at the bottom of each screen, you indicate this in some way so that your users can clearly see that they need to scroll to the next screen/page. Please see the screenshot to the right for an example on how you can do this.
Please keep in mind that Pagination will not divide already existing forms into several parts. It will just place all your form elements onto the first page of your form. You will have to copy or make those inputs from the very beginning for the other pages.

Here you can also enable your app users to see what percentage of fields have been completed. (This option applies to required fields only.)
Last but not least, you can get the Submit button displayed at the top of each page. You might want to do so if a number of fields are optional (not marked as required) and your users don’t need to open each page to submit the form.
ZIP Code

Submit

Project Photos

PDFs

Work Force - Labor

Work Force - Subcontract

Rentals

Labor - Travel Hrs

Subcontract - Work Hrs

Subcontract - Travel Hrs

Rentals Hrs

Rentals Cost

Send a copy of this report to:

snappii@snappii.com

(separate email addresses with commas to send it out to multiple recipients)

Save and Send
7.7.12 Ads
When you have submitted your app to the app store, you can generate Android ad unit IDs (banner ad links), and, in the Ads section, enter them and specify the location of the ads on the screen. Please click here to learn more about using ads in apps built on our platform and obtaining ad IDs.

All in all, when you start working with the Form button, you will see that some sections are used almost all the time: View, Report Common Settings, Auto-Fill Options, Submit to Data Source, Send Reports via Email, Configure Your Element, and Share Form Report. Others, such as Paging and Messages and Labels, are used less frequently. The rest – Submit Data to REST API, Upload Reports to Cloud Drives, and Ads- are very rarely used.

7.8 List
Along with the **Form button**, the List tab/button (also known as the *Advanced List*) is a key element of our platform and is used in the majority of our apps. The List tab/button is used for displaying saved data from the connected data source.

Let us look at the setting options of the List button using our *Inspect Anything* app as an example.

### 7.8.1 View
The View section allows configuring the way your List button will look as well as activating, deactivating, hiding, and showing it.

Two View Modes are available: Button and Framed View.
For the Button View Mode, you can change the label of the List button. In the screenshot above to the left, you can see that we have labeled it *List*. For the Framed View Mode, we use the default label – List, and changing the label in the View section does not change the label of the List button in the Framed View.

Here you can put a check mark next to *Show Refresh button* for the Refresh button, enabling data updates, to appear.

Under *Viewing Conditions*, you can make your List button active/inactive and hide or show it depending on some predefined conditions. To read more about Viewing Conditions, please click [here].
7.8.2 Security

In the Security section, you can enable certain user categories to add, delete, and update information displayed by the List button. (By default, we have four user categories: anonymous (anyone who is not logged in), logged in, app reviewer, and app owner (the person who is building this app).

Thus, for instance, if we want to enable only logged in users to add new information to our data source through the List button, we need to open Add Operation Permission and put a check mark next to Logged in. In the screenshot below, you see that the Add button has appeared.
If we want to enable only logged in users to delete any saved information from our data source through the List button, we need to open *Delete Operation Permission* and put a check mark next
to *Logged in*. In the screenshot below, you see that the Edit button has also appeared. When our users click on it, they’ll be able to delete any of the saved reports by clicking on the cross sign.

Similarly, if we want to enable only logged in users to edit any saved information from our data source through the List button, we need to open *Update Operation Permission* and put a check mark next to *Logged in*. In the screenshot below, you see that once we click on the Edit button, we can also see the pencil symbol. Now our app users can also edit any of the saved forms by clicking on this pencil symbol.
A word of caution: to be able to test this functionality, you might also put a check mark next to *App Owner* in each of these options or log into the app under a different account to be considered a logged in user, not the app owner.

If we put a check mark next to *Allow data creators to update and delete their own data*, our app users will only be able to edit or delete the information that they have previously added into the app, leaving the information added by others intact.
Last but not least, if we enable adding and editing saved data, we can also choose the format of the form to be added and/or edited by clicking on the drop-down menu next to Adding Form and Update Form accordingly. Opting for the standard form will open the form generated from our data source, but the overall design and layout might not be preserved. The editor will simply generate the fields existing in the data source. That is why we recommend selecting the digital form connected to the same data source and from which our List button pulls information.

Here are the screenshots for you to see the difference.
All these setting options are easy to configure but are extremely useful. For instance, most app builders don’t allow anonymous users to add, delete, or edit any information from the List button. They, however, enable different user types to perform different operations. Thus, for instance, they might allow regular staff members to only view saved information, while letting managers view and add new information, and the administrator – view, add, edit, and delete any saved information.

7.8.3 List View
The List View section is the most important one.
Here we choose one data source from the list of previously uploaded data sources in the Data Source Dialog, name the List and Detail Views, configure them, and add the message that will get displayed every time no previously saved information will have been found.

The data source is used to enable the List button to display any information previously entered and saved in this data source. In our case, it is *New Report*. Our Form button, through which our app users fill out inspection reports, is also connected to the same data source (*New Report*). That is how any information entered and submitted in our digital form (via the Form button) can be accessed, edited, and shared at any time later on by clicking on our List button.
If we named the List View *Report List* and the Detail View *Detailed Report*, here is what we would see.
To configure the List View, click on Change next to List View template, and, in the popped up window, select any of the three standard templates and select what fields from the data source connected to our List button will be displayed in the List View as the heading, image, and details.

Alternatively, you can opt for Custom View, which gives you much more flexibility, and specify your size of the List View cell.

To configure the Detail View, click on Change next to Detail View template, and, in the popped up window, select any of the four standard templates and select what fields from the data source
connected to our List button will be displayed in the Detail View as the heading, image, and details.

Alternatively, you can opt for **Custom View**, which gives you much more flexibility, or choose not to have any Custom View by selecting **No view details**.
After you have finished configuring the List View and Detail View, you can either continue with the rest of the settings of the List button or click OK to see how these two views now look. If you have opted for any of the standard templates, you won’t be able to customize the corresponding views, and their buttons will look grayed out, like in the screen on the left hand side with the Detail View button inactive. If you have chosen custom views, now you can open each view and add any number of elements to it.
Let us open the Cell View and add several fields connected to the data source (*New Report*).

We are going to add 4 elements: one Text element to display the name of each saved report, one Text element to display the date of each report, one Image element to show that our users can see more on each report and edit it, and one Form button to enable our users to edit and resubmit each saved report.

Here is how our List/Cell View looks once we have added all these elements (You can’t see the Form button, which is located above the other three elements, because we made its transparent so that our users can get redirected to the digital form every time they click on the Image element or anywhere else inside the cell).
Here is how we configured them.

**Text Element for Report Names**
Text Element for Report Dates

Image Element

Form Button
The quick brown fox jumps over the lazy dog.
Here is how we get from the List/Cell view into the digital form inside the Form button.
Now that the List/Cell View is ready, we could open the Detail View to add several fields connected to our data source (*New Report*). However, since we added the Form button for viewing, editing, and sharing saved reports onto the List/Cell View, there is no need for the Detail View. Just to illustrate how to work with the Detail View, we will delete the Form button and add a few fields onto the Detail View.

Here is how our Detail View looks after we have added one Image element and 6 Text elements. (This is just a sample. You can add as many elements as you need and configure them in a variety of ways.)
Here is how we configured each of these elements.

**The Image Element**

![Image Element Configuration](image)

**The Report Name Caption**
The Report Name Element Connected to the Data Source

The Date Caption
### The Report Date Element Connected to the Data Source

<table>
<thead>
<tr>
<th>Name</th>
<th>TextArea 2961</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>ReportDate</td>
</tr>
<tr>
<td>Font size</td>
<td>12px</td>
</tr>
<tr>
<td>Alignment</td>
<td>Left</td>
</tr>
<tr>
<td>Font color</td>
<td>#3482AF</td>
</tr>
<tr>
<td>Source</td>
<td>Data Source</td>
</tr>
<tr>
<td>Select Text Field</td>
<td>date</td>
</tr>
</tbody>
</table>

The quick brown fox jumps over the l...
The Email Address Element Connected to the Data Source

Here is how we get from the List/Cell view to the Detail View.

The quick brown fox jumps over the la...
7.8.4 Sorting Filter

The Sorting Filter section allows you to set up data sorting either in the ascending order or the descending order. You can also add several sorting filters by clicking on the plus sign and selecting any field from the data source you would like to use as the sorting criterion.
Thus, for instance, you can sort saved data based on the report date so that the most recent reports come first and based on the price so that the least expensive orders show first.

In our case, we used the date and name fields as the sorting criteria. Here is what we got.
You can see that the names of the reports have been sorted in the ascending order (From A to Y). However, the dates or the reports have been sorted in the descending order (We have two reports called Alex, but the most recent one is shown first.) If we change the sorting filter for the date field from the descending to the ascending order, here is what we will get:

![Sorted Reports](image)

Now you can see that out of the two reports called Alex, the most recent one comes second.

The Sorting Filter section also allows you to specify whether your sorting choices will be always applied or your app users will be able to override them by manually changing the sorting order.

![Sorting Filter](image)

Please keep in mind that to enable your users to sort saved reports, in the Data Source Dialog, you need to select the data source and then put a check mark next to each field you would like to be sorted.
After you have done so and selected *Be overridden when a user selects search or sort options from the app*, in your app you and your app users will see the Sort button. Now your app users will be able to manually sort saved reports based on the order they prefer, thus overriding the criteria specified by you.
7.8.5 Search Settings

7.8.5.1 Simple Search Settings

Opting for Simple Search Settings adds a search bar at the top of the screen and allows your app users to look for previously saved reports based on some specified condition.

Four options are available: =, not =, contains, and doesn’t contain.

You can select All fields to enable this search filter for all the fields in the data source or select one of the fields.

Let us look at each of these four options.

If we indicate that our app users can look among the saved reports based on some pay rate and that they need to be able to find all the reports containing the pay rate equal to the one they enter into the search bar, here is what they will get.
As the screenshots above illustrate, when we enter 25, we get 3 reports with the same hourly rate - $25.

If we indicate that our app users can look among the saved reports based on some pay rate and that they need to be able to find all the reports not containing the pay rate equal to the one they enter into the search bar, here is what they will get.
As the screenshots above illustrate, when we enter 25, we get a number of reports with different hourly rates; but none of them has the rate equal to $25.

If we indicate that our app users can look among the saved reports based on report names and that they need to be able to find all the reports containing the name they enter into the search bar, here is what they will get.
The screenshots above clearly illustrate that out of a number of the saved reports only three were found based on the first search requirement (Alex) and only one was found based on the second search requirement (Dunn).

If we indicate that our app users can look among the saved reports based on report names and that they need to be able to find all the reports not containing the name they enter into the search bar, here is what they will get.
This time you can see that only 5 reports were found because the other three contain Alex in their names.

### 7.8.5.2 Searching Filter

The Searching Filter is used in the majority of our apps. Most of our digital forms have an invisible field capturing the email address of the logged in user. In addition, in the Searching Filter of the List button connected to the same data source the digital form is linked to, we have the following filter: emailper = <user.emailAddress>.
The quick brown fox jumps over the lazy dog.
We make all these changes because every time our users click on the List button, this filter gets automatically utilized, and our users only see the reports they have submitted under their email addresses.

Naturally, more than one searching filter can be applied and linked to the fields inside the app. Thus, for instance, in our sample app, two more filters are employed: by report name and by date. Let us look at them separately.

The first filter retrieves only those reports that contain the word entered in the search bar.
The second filter retrieves only those reports whose date is equal to the one entered in the search bar.
Last but not least, under the Searching Filter subsection, you can specify whether these filters will be used at all times or your app users will be able to override your searching criteria. (See Simple Search Settings for more information.)

Moreover, you will be able to specify whether all of these filters are required, whether only some of them are mandatory, whether only one must be applied, or none is required at all.

### 7.8.5.3 Advanced Search Button

Enabling the *advanced search button* allows your app users to add their own search filters.
All you have to do is to ensure that all the fields that they could search by are checked off in the Data Source Dialog.

Right now, only two fields (name and date are selected). That is why our users can only utilize them.
If, for instance, we put a check mark next to *email2* (the email address under which they are logged in), they will be able to search by it as well.
To conclude, these are the most typically used search options and scenarios. You don’t have to use every one of these search variants (Simple Search Settings, Searching Filter, or Advanced Search Button). We recommend that you choose the most optimal variants for you and your app users.
The Share Settings section allows sharing data from the List button. This is not a very frequently used section because most of the times we email filled out data right from the digital form upon the submission. However, this option also exists and, to be able to take advantage of it, you need to put a check mark next to Show share button, enter the addressee(s), email subject, body of the email, title of the Share button, message appearing after the email has been successfully sent, report format, and report name.

Next, you need to specify if saved data will be shared from the app or server. (If you choose the app option, after clicking on the Send button, your app users will be able to edit the email message. If you opt for the server option, once they click on Send, the message will be instantly sent.)

If you click on Data in report, you will be able to manually put a check mark next to each field you would like to be sent in each report.

Data in Report
Then you need to keep the share options as they are, put a check mark next to *Include user information* if you want to include the information on each user in the report, and specify what PDF report template should be used: the default one or configure and select a custom-made one.

Here is how we have configured the settings of our sample app.
Here is how these settings work.
Attached please find a new inspection report.

Sent from my U.S. Cellular® Smartphone.
All in all, the biggest advantage of sharing data from the List button is that here you can share several reports in one PDF/Excel/ or Word file, while when sharing from the Form button, you can only send one report at a time.

### 7.8.7 Configure Your Element

<table>
<thead>
<tr>
<th>Configure your element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height: 45</td>
</tr>
<tr>
<td>Width: 185</td>
</tr>
<tr>
<td>Use rounded corner</td>
</tr>
<tr>
<td>Font: Default</td>
</tr>
<tr>
<td>Button style: Icon</td>
</tr>
</tbody>
</table>

This section allows configuring the dimensions, font, and button style (global, icon, or custom) of the List button. However, this section is only available for the button style, so if in the View section you opted for the Framed View style of your List button, this section will be grayed out, like in our case.
When you have submitted your app to the app store, you can generate Android ad unit IDs (banner ad links), and, in the Ads section, enter them and specify the location of the ads on the screen. Please click here to learn more about using ads in apps built on our platform and obtaining ad IDs.

### 7.9 Map Button

The Map button works very similarly to the Map tab. The main difference is that you can locate the Map button on any tab.

1. Open the tab on which you’ll need to place the Map button.
2. Place the Map button on it.
3. Create an Excel spreadsheet with the business locations you need to display in the application (see the section on the Nearby tab).

4. Click on “Edit Element” of the Map button.
5. Change the icon of the map, if necessary. Adjust the button size. Click “Upload XLSX file”

6. All the other steps are identical to the ones discussed in the Map tab section.

The results are displayed in exactly the same way as when using the Map tab (see above).

7.10 Gallery Button
The Gallery Layout button and tab work very similarly. For more information, please read Gallery Layout.
The Report List button is very frequently used in our apps. In fact, the majority of our apps have the Report tab with the following structure:

1. A form that our users fill out (the Form button)
2. A list that stores filled out forms (the List button)

Let us see how to utilize the functionality of the Report List button.
1. Drag and drop the Report List button on any tab you need. (We usually select the New tab for such purposes.)

2. Specify the icon, dimensions, font, button style, and view mode.

3. Specify the list view title, which will be displayed at the top of the screen. In our case, it is *Daily Report PDF*. (See the screenshot below.)
4. Enter the message that will be displayed if no saved PDF reports are found. We opted for “No saved PDF reports have been found.”

5. Select one of the available templates for your Report List or opt for Custom View.
6. From the drop-down menu, choose the form from which generated reports will be sent to our Report List. In our case, it is *New Daily Report Form*. 
7. Put a check mark next to Group by Data if you wish.
8. Open the settings of the Form button/tab from which reports will be generated and sent to our Report List.

9. Put a check mark next to Allow users to share reports.

10. Next to Report Options after Form Submission, from the drop-down menu, choose any of the available options. (We opted for View and Save.)
11. Choose the format(s) of the report: PDF, Excel, or both.

That is all you need to do to configure the Report List button. The screenshots below show how it works.
The screenshots clearly illustrate that after a completed form gets submitted (Save and Send), the report gets saved into the List button/tab and a PDF/Excel (depending on your settings) gets sent to the Report List button/tab. Your users can view generated PDF reports instantly or view them later on by clicking on the Report List button/tab. There they can also delete previously saved reports.
7.12 Add to Contacts Button

The Add to Contacts button enables your app users to create contacts without exiting your app.

To enable this functionality, simply drag and drop the Add to Contacts button onto the canvas. In the popped up window, specify the label, icon, dimensions, button style, and font of your Add to Contacts button.
Here is how this button works on a mobile device.
7.13 Add to Favorites

The Add to Favorites button allows adding any items to a list of favorites. Let us say we have a list of doctors and would like to make a list of the doctors we most frequently go to. This could be used for easier access as well as to auto-populate fields with items from our favorites.

To enable this capability, follow the steps below:

1. Create your data source from which you will select items to add to the favorites.

Here is our example:
2. Upload this Excel spreadsheet in the Data Source Dialog, name it, and save it.

3. Add the List tab/button onto the canvas and connect it to this data source. Make sure to choose the Custom view for the Cell/List View.

4. Go back to the Data Source dialog and, next to Type, select Favorites.
5. Next to Fields, click on the plus sign and add the same fields located in the previously added data source. (In our case, it is the data source called Doctors.) Here are the fields we added.

```
favoritesRowName is the default field that is always created by our editor when we are working with the Add to Favorites button. This field is necessary for the names created by the option Allow the app user to enter a name for the added item to be displayed in the Cell/List View of the Add to Favorites button.
```

6. Name the data source and save it. Now we have two data sources: one for the main list (in our case, a list of doctors) and one for the items added to the list of favorites.
7. Open the Cell/List View of the List tab/button we have previously created and drag the Add to Favorites button onto it. (Here you can also add any other elements to display the fields from the first data source. Just make sure to select Data Source next to Source and the corresponding fields for each Select Text Field.)

8. In the popped-up window, configure the way this button will look: its add and remove labels, icons, dimensions, font, and button style.
9. Next to Select Data Source, select the data source you have created for the list of favorites.

10. Click on Change and match the fields from the two data sources you have created. (In our case, the fields from the data source for adding favorites is on the left-hand-side, and the fields from the original list – on the right-hand-side.)
11. Next to *Favorites Mode*, specify if you would like selected items to be added to this data source (*Copy*) or simply added to the list of favorites (*Reference*).

Here is how both of these options work.

**Reference**
No saved doctors have been found.
As you can see in the screenshots above, you can easily add an item to and remove it from the favorites at any time by clicking on this item. (See the heart with the minus sign.)
In this case, you can easily add an item by clicking on the plus sign, but once you do so, the item gets automatically saved in the linked data source. That is why you won’t be able to delete the item afterwards.

Here is a screenshot from our *Favorites* data source.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>favorites</td>
<td>Photo</td>
<td>Specialty</td>
<td>StreetAdd</td>
<td>City</td>
<td>State</td>
<td>PhoneNumber</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><a href="http://car">http://car</a> pediatric</td>
<td>100 Please</td>
<td>Boston</td>
<td>MA</td>
<td>617-000-0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

12. Now that your *Add to Favorites* button has been fully configured, add one more List tab/button to display any items added to the list of favorites and connect it to the second data source. (In our case, it is *Favorites.*) Configure the Cell/List and Detail views for this List tab/button and connect the added elements to the specified data source. Here are our sample Cell View and Detail View.

**Cell View**

Here is how we configured the elements for the List/Cell View.
The quick brown fox jumps over the fence.
(In our case, the Detail View is identical for both of our List buttons.)
(For each of these elements we also chose \textit{Data Source} for each source and the corresponding fields next to \textit{Select Text Field}.)

13. Instead of adding several fields to the Cell View of the second List tab/button, you might want to put a check mark next to \textit{Allow the app user to enter a name of the added item} and add only one field to the Cell View – the field for this name to go into.

Here is how we configured this new field.

Here is how this option works on a mobile device.
Next to Favorites Mode, select Parent or Custom mode to specify how the Detail View of the second List tab/button will look. If you select Parent, it will be identical to
the Detail View of the first List tab/button, and you don’t have to add any fields to the Detail View of the second List tab/button.

As you can see, the Detail View of our second List button is empty. However, on a mobile device, all the information gets displayed.
Please keep in mind that these two options are available only for the Reference mode. The Copy mode only supports the Custom Detail View.
The Audio Player button enables you to add M4A files within your app so that your users can access and play them on demand.

To enable this capability, follow the steps below.

1. Drag and drop the Audio Player button onto the canvas.

2. In the popped up window, click on Specify your audio to upload your M4A file from your computer or from the URL of your audio file. Click OK.
3. Specify the name and dimensions of your Audio Player button.

4. Put a checkmark next to Show stop button if you wish.
5. Put a checkmark next to *Show seek bar* if necessary.
6. Put a checkmark next to *Show playing time* if you wish.

*Show playing time enabled*  
*Show playing time disabled*

7. Put a checkmark next to *Transparent background* to make the background transparent.

*Transparent background enabled*  
*Transparent background disabled*
Here you can also change the background and foreground colors.

8. Put a checkmark next to *Sharing enabled* if you wish. This option will allow your users to share your audio files with others if there is the *Share button* on the same tab.
7.15 Calculator Button

The Calculator tab/button is similar to the New and Container tabs/buttons. However, its main purpose is to contain fields and formulas to perform calculations.
Here is an example of an app employing the Calculator button.

The button consists of:

1. 5 Image elements:
2. 5 Text elements: Loan amount, Interest rate, First installment, Loan term (years), and Monthly payment.

3. 3 Text/Number fields: A, B, and D.

4. 1 Multi-Choice element: M

5. 5 Number-Calculated fields: (the rest of the fields you see in the screenshots below). In the app, only one (the Number-Calculated field for monthly payments) is visible because the other four have been configured as invisible.

This is how the first Number Calculated Field (the one for monthly payments) is configured:
Here is how Field F (the second Number Calculated Field) is configured:

```
ROUND((((<A> * <D>) * <f> * (POWER((1 + <f>), 2) - 1), <b>)) / (POWER((1 + <f>), 2) - 1), 2)
```

Here is how the third (second invisible) Number Calculated Field is configured:

```
(<B> / 12) / 100
```
Here is how Field G (the fourth Number Calculated Field) is configured:

Here is how Field L (the fifth Number Calculated Field) is configured:
Here is the formula we used to calculate monthly mortgage payments, where

\[ M = P \frac{r(1 + r)^n}{(1 + r)^n - 1} \]

The variables are as follows:

- M = monthly mortgage payment (First Number Calculated field)
- P = the principal, or the initial amount you borrowed (Field A - Field D)
- r = your monthly interest rate. Your lender likely lists interest rates as an annual figure, so you’ll need to divide by 12, for each month of the year. So, if your rate is 5%, then the monthly rate will look like this: 0.05/12 = 0.004167 (Field B divided by 12 = Field F)
- n = the number of payments, or the payment period in months. If you take out a 30-year fixed rate mortgage, this means: n = 30 years x 12 months per year, or 360 payments. (Field L)
Now the formula for the first Number Calculated field becomes clear:

```
ROUND(((<A> * <D>) * <F> * (POWER((1 + <F>), <B>)) / (POWER((1 + <F>), <B>) * 1.2))
```

This field is the only visible one because the other 4 are used to simply perform the calculations with the visible Text/Number and Multi-Choice fields and are not to be seen by the user.

Thus, the invisible Field F calculates our monthly interest rate based on the value from Field B. Field G converts the textual Field M to a number with the formula TO_NUMBER (<M>). Field L calculates the number of payments based on the value from Field M.

This is just an example of how the Calculator button can be used. In fact, the possibilities of using formulas in our apps are limitless.

### 7.16 Catalog Button

The Catalog button works very similarly to the Catalog tab. The only difference is that it can be added on a number of tabs together with other buttons and that its dimensions and style need to be configured. To read about working with the Catalog tab, please click [here](#).
7.17 Chart
The Chart button works very similarly to the Chart tab.

7.18 Directions
This button can be added to all advanced controls, including Advanced Map, Gallery Layout, and Advanced List. The button utilizes the location information from the selected data source/data connector of the tab it is used on and Google Maps.
When the button is tapped, the data from a specified location field gets used as the destination. The starting location in most cases will be the user’s current location. However, on Advanced Map, if the user has changed the starting point shown on the map to some manually entered location, this is used as the starting point.
Here are the steps to set up this button:

- Click on the tab you need the Directions button for and select the Directions button under *Elements and buttons*. Drag it onto the canvas.
Specify the label or remove it, configure the icon, dimensions, and select the Location column for the Location field. Click Ok. Align the button if necessary.
The directions button is now ready!

7.19 Facebook Button
The Facebook button works very similarly to the Facebook tab.

**7.20 Forgot Password**

The Forgot Password button allows sending your registered users an email reminding them of their password. All they will need to do is to enter their email address linked to their account at Snappii and click *Forgot password*?
An email with their password will be instantly sent to them.
To enable this capability, drag and drop the Forgot password button on the Log In form and, in the popped up window, specify its label, icon, dimensions, icon style, and font.
7.21 In-App Purchase

The In-App Purchase button enables you to charge your users for ad removal and monthly/annual subscription. Here you can read how to configure subscription products and ad removal in the App Settings section of your app (to the left of the canvas). Here you will also learn how to obtain Google Play Store product IDs. Once you have configured all in-app products you would like to offer in your app, you need to add one In App Purchase button for each type of subscription and one for ad removal.

Simply drag and drop the In App Purchase button onto the canvas and, in the popped up window, specify its label, icon, dimensions, icon style, and font. Next to Google Play Product ID, choose the type of the subscription/ad removal option this button will enable. Click OK.
Thus, for instance, for the Heavy Equipment Inspection app featured below, next to *Apple App Store product ID*, from the drop-down menu, we selected *Ad Removal* for the first button, *Monthly Subscription* for the second button, and *Yearly Subscription* for the third button. On the screenshot below, you can see that we only have product IDs for iOS devices, so these three in-app purchase products will only be available on iOS devices. That is why it doesn’t matter which in-app product type we select next to *Google Play product ID*. However, if, in the App Settings section, you have configured your in-app products for the Google Play Store as well, you need to specify these product types correctly in the settings of the In App button.
7.22 JS Plugin Button

JS Plugin button works very similarly to the JS Plugin tab, described in more detail here.

7.23 LinkedIn Button
The LinkedIn button works very similarly to the LinkedIn tab.

7.24 Location Button
The Location button enables you to enter a specific address and have your users instantly get directions to it right within the app.

To enable this functionality, follow the steps below:

1. Drop the Location button onto the canvas.

2. In the popped up window, specify the label, icon, dimensions, style and font of the Location button.

3. Most importantly, enter the latitude and longitude of the address you would like your users to get directions to. You might find this website helpful when looking for latitude and longitude coordinates.
4. To display the address of the destination, you can enter its address as well.

**With the Address Provided**

**Without the Address Provided**
From the screenshots above, you can see that the current location of the user is marked by a blue pin, whereas the final destination (the business where we’d like to direct our users) is marked by a red pin. The screenshots also illustrate that when we only provide the latitude and longitude, no address gets displayed.

To give you an example of how the Location button works, we entered Snappii’s address. The screenshots below illustrate how one can calculate the distance between his/her current address or any other address to the final destination specified by the app builder.
7.25 Login Button
The Login button enables your customers to log in. Unlike the Login Form button, where you can configure the fields the way you like, here your users will see a standard form.

Here is how the Login button looks on an Android phone and iPad.

To enable this capability, simply drag and drop the Login button onto the canvas and, in the popped up window, specify its label, dimensions, font, mode, and style.
7.26 Login Form Button
The Login Form button works very similarly to the Login Form tab and allows more flexibility than the Login button.
Click [here](#) to read about configuring the [Login Form tab/button](#).

### 7.27 Logout Button

The Logout button works very similarly to the [Logout tab](#). Please click [here](#) to learn to configure the [Logout tab/button](#).

### 7.28 Navigation Button

With the Navigation button, you can redirect users to a specific tab in your app, be the tab visible or invisible to them (in case of Android apps). This option eliminates the number of tabs visible to a user and improves the overall user experience.

For the button to work properly, all you need to do is to correctly indicate the tab you want to redirect your users to by choosing it in the drag-and-drop menu next to [Navigation path](#).
For example, we utilized this navigation option for the Customize This App button.
Here are the steps we followed:

1. When we dragged the Navigation button onto the canvas, the following screen popped up:
2. In our case, we left the field next to *Label* blank and uploaded our own icon for this Navigation element.

![Navigation button settings](image)

3. We also changed the dimensions and font of the Navigation button.

4. Most importantly, we chose the Customize tab next to *Navigation path* and made sure the chosen path was correct by clicking on *Show control*.

5. We chose *Icon* as the button style and clicked *OK*.

Here is the result we got.
If we were to make the Customize tab invisible, we would see the following:

Instead of the *Icon* style for the Navigation button, we could have chosen *Global Style* and slightly changed its dimensions like in the screen shot below.
Then the Navigation button (Customize This App) would have looked like this:

Alternatively, we could have opted for the Custom style,
clicked **Customize**, and changed the button style, color, text, color, and border.

Then the Navigation button (**Customize This App**) would have looked this way:
There is one more option, but it is only valid for iOS apps: *Open control screen over current screen.*

(In Android apps, you always return to the initial tab be this box checked off or not.)
When we check off this box, we get onto the destination screen and then can go back to the initial tab by simply clicking Help. (You can see that we stay on the Help tab throughout this whole process.)

**The box checked**

![Image 1](image1)

![Image 2](image2)

**The box unchecked**

![Image 3](image3)
When we leave this box unchecked, we completely go to the destination tab (the Customize tab in our case) without being able to return to the initial tab (the Help tab).

Please keep in mind that in iOS apps, you can’t navigate to invisible tabs. For instance, if we make the Customize tab invisible to logged in users, the navigation button won’t work, be the box next to Open control screen over current screen checked off or not.
7.29 Nearby Button

The Nearby button works very similarly to the Nearby tab with only a few minor differences:

1) The name, size, and the icon of the button can be changed.
2) The Nearby button can be placed on any tab;
7.30 PayPal
PayPal is an international online service that allows users to pay for goods and services, send money, and receive money either on their computers or mobile devices. By integrating PayPal into your app, you enable your users to make payments inside the app.

To enable PayPal transactions in your app, choose the PayPal button under *Elements and Buttons*, configure it, and place it onto the canvas.
Please keep in mind you can only add one PayPal button per app.

Now you’ll need to obtain and enter several essential credentials, such as Sandbox merchant PayPal ID and Live App ID, in the PayPal Settings section to the left of the canvas to be able to test and submit the app with this functionality.
The PDF Form button allows editing and sharing PDF documents. It is commonly used in our apps. In fact, the majority of our apps have the Upload tab with the following structure:

1. A form allowing our users to upload their own PDF documents (the Form button with the Document Upload button.)
2. A form that enables our users to edit and share uploaded PDF documents (the List button with the PDF Form button (*Use*) and the Form button (*Edit*), whose *Submit to Data Source* submission type is set as *Update* )
Daily Observation Report

Location and Date of Work: [Location and Date of Work]

Weather: [Weather]

Site Conditions: [Site Conditions]

Work Completed:
- Completed installing sediment fence along [Location]

Discrepancies Noted:
- Nonconformance Report [#]

Discussion with Contractor:
- Include order given and received.

Possible Issue:
- With [Description of Issue] the contractor may need assistance with [Specific Assistance Needed].

Work in Progress (Include report of operators):
- [Operator Report]

Possible issue with Kinder Morgan’s rail line:
- [Possible Issue Details]
3. A list that stores saved PDF documents and allows deleting and sharing them (the Report List button)

Here is how the PDF button works on a mobile device.
1. To begin with, let us upload some PDF file by opening the Form button and clicking on the Document Upload button. Next, enter your name of this PDF template and click on the symbol of the PDF file, which is now your PDF Form button. (Alternatively, you can hit \textit{Save Template}, which is the Submit button, and which will send the PDF file you have just uploaded into the linked data source and will allow viewing it via the List button, which is also connected to the same data source.)

2. When you click on \textit{Edit} from the PDF file symbol or \textit{Use} on the List button, your uploaded PDF form will open up, and, if there are any fillable fields, they will get highlighted automatically.
Alternatively, you can click on any area and some information there with the help of the tools on the popped-up black-and-white screen.

3. When you are ready, simply click **Save** to store this form on your mobile device or share it with others.

This can be a very useful button, in particular for registration and booking forms.

**7.32 PDF Viewer Button**

The PDF Viewer button works very similarly to the PDF Viewer tab and is also used for displaying PDF files from a specified URL.
All you need to do is to drag and drop the PDF Viewer button onto the canvas, configure its label, icon, dimensions, font, type, and style. Most importantly, enter the URL from which the desired PDF file will be taken.
Here is how the button works on a mobile device.
Alternatively, you can add the PDF Viewer button onto the List tab/button and specify the field in the linked data source to store the links to PDF files that were previously entered through the connected mobile form or manually into the data source.
Here is how the latter option works on a mobile device.
7.33 Photo Button
The photo button allows taking, uploading, and sharing photos right within the app.

To utilize this capability, follow the steps below:

1. Drag and drop the Photo button wherever you would like.

2. In the popped up window, specify the label if any, the email to send photos to (if you enable this type of sharing below), the subject of the message, and change the icon if necessary.
3. Specify the dimensions, font, and button style.

4. As for the sharing options, here you can select any number of options available. (You’ll need to use the Shift/Ctrl key to select several options.) For instance, sharing via email looks like this.
There is also a video tutorial available for the Photo field. Click [here](#) to watch it.

### 7.34 Post to Facebook Button
Here you can redefine the information you want your users to share on Facebook.
All you need to do is to add the Post to Facebook button onto the canvas, simply type this message next to Text, and then click OK. When you users click Post to Facebook, it will bring them to their Facebook page and this predefined message will automatically pop up, enabling them to post this message on their Facebook page.

Here is how your message will look on Facebook.
As you can see, this is a terrific way to help your app users spread the word about your app.

### 7.35 Post to Twitter Button

Here you can predefine the information you want your users to share on Twitter.

All you need to do is to add the Post to Twitter button onto the canvas, simply type this message next to Text, and then click OK. When you users click Post to Twitter, it will bring them to their Twitter page and this predefined message will automatically pop up, enabling them to post this message on their Twitter page.
Here is how this button works on a mobile device. When you click Post to Twitter, the Complete action using Direct Message/Tweet window pops up. Select Tween on then again Tweet to post your message.
Here is how our tweet now looks on Twitter.

There is a video tutorial for the Post to Twitter button. Click here to watch it.
The QR Code button works very similarly to the QR Code tab.

[Image of QR Code button and interface elements]
For more details, click here.

7.37 QR Image Button

The QR Image button allows displaying QR codes created in the QR codes section located at the top of the screen to be scanned and shared afterwards. Please click here to learn about how to create QR codes in the QR codes section.

Once you have created a QR code, drag and drop the QR Image button onto the canvas. In the popped up window, click Select QR Code and choose the QR code you’ve created.
Here you can also rename and resize the QR Code Image button.

**7.38 QR Scanner Button**

The QR Scanner button works very similarly to the QR Scanner tab.
For more details, please click here.

7.39 Reminder Button
The Reminder button allows your users to add reminders within the app. To add a reminder, drag and drop the Reminder button onto the canvas.

In the popped up window, specify its title, label, icon, dimensions, font, and button style. Then click OK.

Now open the app in the Snappii’s Preview App on your device and click Add to add a reminder. You can specify its title, date, time, type (once, repeatable, alarm), snooze duration (1/10/30 minutes), vibration (off/on), and ringtone.

Once the pop-up notification arrives, you can choose to stop it or snooze.
The Remove Account button allows removing the information of a user who has previously registered in the app through our Create Account form.
To enable this functionality, drag the Remove Account button onto the tab you wish and configure it the way you need. Then click OK.

This is all you need for the Remove Account button to work properly. The only word of caution is that you, as the app builder, won’t be able to delete the account under which you build apps on our platform. To test the button, you will need to create a new account in the app itself and log in under those new credentials.

For instance, we created a new account under Ann Smith and verified her email. Please see the screenshots below.
Then we logged in under these new credentials and clicked on *Remove Account.*
We clicked *Yes* under *Do you really want to remove your account from this app?* in the popped up window and then were automatically logged out. When we tried to log in under the same credentials, we weren't able to do so because our account had been removed.
7.41 RSS
RSS looks like a news feed when added to an app. Users can read news or blogs right within your app. Thus, for instance, you can share your recent blogs or feeds on something related to your industry.

To enable this functionality, simply drag and drop the RSS button onto the tab you need it for. (In our case, it is the New tab.) In the popped-up window, enter the link to the RSS feed you plan on sharing with your users.
For example, we entered “New York Times RSS” in a search engine to get directed to the page below.

![RSS Feed XML](http://bits.blogs.nytimes.com/feed/)

For example, we entered “New York Times RSS” in a search engine to get directed to the page below.
Then we chose a feed we are interested in and clicked on it to get to the screen below.

After that, we simply copied the URL link of this feed into the settings of our RSS tab.

This is how our RSS feed on a mobile device now.
7.42 Section Break Button
This button allows adding section break lines and configuring their dimensions and colors.

All you need to do is to add the Section Break button and, in the pop up window, specify its height, weight, and color. Make sure to enter valid dimensions.

The sizes are out of range. Correct sizes are between 1x15 px and 10x320 px.
Here is how section breaks look on a mobile device.
By clicking on the Select button, you can initiate some processes, such as a search. For instance, we have a list of saved reports and would like to search for specific ones based on their names and dates. In our example, we have named our Select button Search.
Let us enter one of the names of the reports that have been previously saved on our mobile device. As you can see in the screenshot below, by simply entering the name, we did not get any search results. However, as soon as we clicked on the Select button, we got several reports containing this name.

Similarly, when we enter any date under *Search by date*, we don’t get any search results. However, as soon as we click on *Search*, we get the reports having been generated on that day.
To enable this capability, follow the steps below:

1. Drag and drop the Select button onto the canvas. In the popped-up window, specify its label, dimensions, font, button style, and, most importantly, the control you would like to enable the Select button for.

2. Add the fields you will enter your search criteria into. In our case, we added one Text field and one Datetime field. Here are there settings:
3. In the Search Settings section of that control you enabled the Select button for, set up the searching filter and click OK.
Your Select button is now ready.

Overall, the Select button is used infrequently. However, it comes in handy in situations like the one illustrated in the screenshot below, when one needs to perform a search based on multiple criteria.
7.44 Select Contact Button

The Select Contact button allows you to auto-populate some fields inside of your digital form with contact information from the address book on your mobile device.

The following information can be retrieved from your address book:

- Company name
- Full name
- Website
- Home and mobile phone numbers
- Fax number
- Address, including the street name, country, and city.

To take advantage of this capability, follow the steps below:

1. Add the Form button and the fields you would like to be auto-populated inside of it. Here is an example:
Here are the settings of each of these fields we have added:
<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Last Name</td>
</tr>
<tr>
<td>Web Parameter name</td>
<td>TextField39</td>
</tr>
<tr>
<td>Data type</td>
<td>Text</td>
</tr>
<tr>
<td>Width</td>
<td>270</td>
</tr>
<tr>
<td>Height</td>
<td>35</td>
</tr>
<tr>
<td>Order</td>
<td>1</td>
</tr>
<tr>
<td>Next order</td>
<td>1</td>
</tr>
<tr>
<td>Font</td>
<td>Default</td>
</tr>
<tr>
<td>Place holder</td>
<td>Last Name</td>
</tr>
<tr>
<td>Column width (for Excel report)</td>
<td>Auto fit</td>
</tr>
</tbody>
</table>

**The quick brown fox jumps over the lazy dog.**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Street Address</td>
</tr>
<tr>
<td>Web Parameter name</td>
<td>TextField40</td>
</tr>
<tr>
<td>Data type</td>
<td>Text</td>
</tr>
<tr>
<td>Width</td>
<td>270</td>
</tr>
<tr>
<td>Height</td>
<td>35</td>
</tr>
<tr>
<td>Order</td>
<td>1</td>
</tr>
<tr>
<td>Next order</td>
<td>2</td>
</tr>
<tr>
<td>Font</td>
<td>Default</td>
</tr>
<tr>
<td>Place holder</td>
<td>Street Address</td>
</tr>
<tr>
<td>Column width (for Excel report)</td>
<td>Auto fit</td>
</tr>
</tbody>
</table>

**The quick brown fox jumps over the lazy dog.**
<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Parameter name</td>
<td>TextField141</td>
</tr>
<tr>
<td>Data type</td>
<td>Text</td>
</tr>
<tr>
<td>Width</td>
<td>270</td>
</tr>
<tr>
<td>Height</td>
<td>55</td>
</tr>
<tr>
<td>Order</td>
<td>3</td>
</tr>
<tr>
<td>Next order</td>
<td>0</td>
</tr>
<tr>
<td>Font</td>
<td>Default</td>
</tr>
<tr>
<td>Placeholder</td>
<td>City</td>
</tr>
<tr>
<td>Column width (for Excel report)</td>
<td>Auto fit</td>
</tr>
</tbody>
</table>

The quick brown fox jumps over the la..

<table>
<thead>
<tr>
<th>Name</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Parameter name</td>
<td>TextField142</td>
</tr>
<tr>
<td>Data type</td>
<td>Text</td>
</tr>
<tr>
<td>Width</td>
<td>270</td>
</tr>
<tr>
<td>Height</td>
<td>55</td>
</tr>
<tr>
<td>Order</td>
<td>4</td>
</tr>
<tr>
<td>Next order</td>
<td>4</td>
</tr>
<tr>
<td>Font</td>
<td>Default</td>
</tr>
<tr>
<td>Placeholder</td>
<td>Country</td>
</tr>
<tr>
<td>Column width (for Excel report)</td>
<td>Auto fit</td>
</tr>
</tbody>
</table>

The quick brown fox jumps over the la..

Max length (or max number value)
2. Next, add the Select Contact button inside the digital form.
3. In the popped-up window, specify the label of your Select Contact button, its dimensions, font, and button style.

4. In the Inputs Mapping section, match the fields you have created in the digital form (the left column) to the fields in the Select Contact button (the right column). To add a new field, press on the plus sign. To remove a field, press on the minus sign. Here is our variant.
For the Select Contact button to work properly, please follow the following format for the right
column.

\(<\text{personOrganization}\>\ <\text{personFirstName}\>

\(<\text{personMiddleName}\>\)

\(<\text{personLastName}\>\)

\(<\text{personHomePage}\>\)

\(<\text{personPhoneNumber}\>\)

\(<\text{personMobilePhoneNumber}\>\)

\(<\text{personFaxNumber}\>\)

\(<\text{personEmail}\>\)

\(<\text{personStreetAddress}\>\)

\(<\text{personCity}\>\)

\(<\text{personCountry}\>\)

Here is our variant.

5. Once all the fields have been matched, click OK and save the app.

Here is how the Select Contact button works on a mobile device:
As the screenshots illustrate, by clicking on the Select Contact button, we are opening our address book. Then we select any contact we need, and the corresponding fields inside of our digital form get automatically populated with this contact information.

Our Select Contact button has been fully configured. Now you can submit this auto-populated information as a report, share it with others, store it on your mobile device, or continue filling out this digital form.
7.45 Email Button

The Email button allows you to predefine an email address and have your users email the specified addressee with a click of a button. This is a great way to get your app users to easily and conveniently get in touch with you by email.

To add this capability, please follow the steps below.

1. Drag and drop the Email button onto the canvas.

2. In the popped up window, specify its label, dimensions, font, and button style. You might also want to change the icon.

3. Next to Email to, enter an email address your users will send their emails to by default. Click OK.
Here is how the Email button works on a mobile device.

There is also a video tutorial available for the Email button. Please click here to watch it.
The Share button allows sharing data by posting it via Facebook, Twitter, Weibo, sending via Email, SMS, assigning to Contacts, copying to Board, printing out and saving to Camera Roll. In this way, you can easily share audio and video files, photos, signatures, bar codes, QR codes, map locations, to name but a few. The Share button can be added to the Custom tab/button as well as to the cell/detail views of the List tab/button.

To share an element, follow the steps below.

1. First of all, enable sharing in its settings, as shown in the screenshot below for the Photo field.
Next, drag and drop the Share button onto the canvas and, in the popped up window, in the Share Settings section, specify the sharing options available by holding the CTRL button and selecting the required options with the left key of the mouse. If all the options are necessary, use the CTRL+A to select all the items in the list.
3. In the message field, you can enter a message that will be displayed in your email together with the shared element/elements.

4. To attach the screenshot of the form on which the shared element is located, check off the box next to *Always attach the screen shot*.

Here is how this option looks on a mobile device.
5. To share only the message entered in the message field without sharing any elements, select the box next to *Share predefined message only*. (See the screenshot below.)

6. You might also want to change the label and icon of the Share button.

Use this element when you want to share information into social networks, send by SMS or into email, print. Data for sharing will be taken from controls where option ‘Sharing enabled’ is checked.

7. You can enter the default email of the recipient and the subject of the email.
8. Last but not least, you might also want to change the dimensions of the Share button, its style, and font.

Here is how the Share button works on a mobile device. (We’ve also enabled sharing the Location field.)
To: sales@snappii.com

Subject: New Photos

Attached please find the photos of...
7.47 Shopping Cart Button
The Shopping Cart button works very similarly to the Shopping Cart tab. The only difference is that it can be added on a number of tabs together with other buttons and that its dimensions and style need to be configured. To read about working with the Shopping Cart tab, please click here.

7.48 Sign/Draw Button
Like the Sign/Draw tab, the Sign/Draw button enables users to share digital signatures or draw pictures, as well as send them to their co-workers or clients.
To enable this functionality, simply drag and drop the Sign/Draw button onto the canvas.

In the popped up window, specify its label, icon, dimensions, title, font, and button style. Click OK.
Here is how this button looks and works on a mobile device.

There is also a video tutorial available for the Sign/Draw button. Click here to watch it.
The Sign up button enables your customers to register in your app. Unlike the Sign up Form button, where you can configure the fields the way you like, here your users will see a standard form.

Here is how the Sign up button looks on an Android phone and iPad.
To enable this capability, simply drag and drop the Sign up button onto the canvas and, in the popped up window, specify its label, dimensions, font, mode, and style.
The Sign up Form button works very similarly to the Sign up Form tab and allows configuring the registration screen the way you like.
To learn more about working with the Sign up Form tab/button, please click here.

7.51 SMS Button

The SMS button allows you to predefine a phone number and have your users send a text message to the specified business or individual with a click of a button. This is a great way to get your app users to easily and conveniently get in touch with you by email.
To add this capability, please follow the steps below.

1. Drag and drop the SMS button onto the canvas.

2. In the popped up window, specify its label, dimensions, font, and button style. You might also want to change the icon.

3. Next to Phone number, enter a phone number your users will send their text messages to by default.

4. Next to Message, you might want to enter a predefined message as well to save your users time. Click OK.

Here is how the SMS button works on a mobile device.
There is also a video tutorial available for the SMS button. Please click here to watch it.
The Twitter button works very similarly to the Twitter tab.
The Video Player button enables you to add MP4 files within your app so that your users can access and play them on demand.
To enable this capability, follow the steps below.

1. Drag and drop the Video Player button onto the canvas.

2. In the popped up window, click on *Specify your video* to upload your MP4 file from your computer or from the URL of your video file. Click OK.
3. Specify the dimensions and name of your Video Player button.

4. Put a checkmark next to *Sharing enabled* if you wish. This option will allow your users to share your video files with others if there is the *Share button* on the same tab.

**7.54 Web Search Button**

This button enables your users to search for external information and businesses right within your app.
There are three options available:

1. **Business search**: look for businesses near you by simply entering a keyword.
For instance, we entered “organic milk” to get a list of nearby farms offering organic milk and their geographic distance from us.
2. **Only search for businesses related to:** specify in the editor your criteria for the search results.

For instance, in the editor, we entered the word *farm* so that our users can always get a list of farms in the close proximity to them.
3. **Search website for information**: unlike the first option, here the search results will be limited to the ones found on a website specified by you.

To illustrate this option, we entered [www.fda.gov](http://www.fda.gov) in the editor and then the word *dairy* in the search bar of the app on our mobile device.
Maytag Dairy Farms Voluntarily Recalls Blue Cheese Products Due to Possible Health Risk

Maytag Dairy Farms Expands Voluntary Recall Of Blue Cheese

Maytag Dairy Farms Blue Cheese Sold at 17 Select Schnuck ...

Dairy Groups Exporting to China Face New Requirements

List of U.S. Dairy Product
To implement this button within our app, simply drag and drop the Web Search button onto the tab of your choice and then configure its label, dimensions, font, title, button style, and search option.

**7.55 YouTube Button**

The YouTube button enables your users to access a predefined page with a video right within your app. Like any button, it can be placed on a number of different tabs, including the List, Map, and Gallery ones, together with other buttons. However, unlike the YouTube tab, the button allows pulling information not only from a constant website (Source – Constant) but also from a data source (Source – Data Source). Let’s look at each of these options. **Source – Constant**

1. Specify the label, title, font, dimensions, and button style.

2. Next to URL, enter the URL from which the app will pull information. In our case, it is [http://www.bed&breakfast.com](http://www.bed&breakfast.com), the website for the rooms featured in the app.

3. Next to Source, choose Constant.
4. Check off the box if you want the button to support file sharing and downloading. Click OK.

Here is how the button works on a mobile device.

Source – Data Source
1. Create a data source with the column Video and upload it in the data source dialog. (In our case, we uploaded the URL of the YouTube videos for each room featured in the app.)

<table>
<thead>
<tr>
<th>Name</th>
<th>Video</th>
<th>Photo</th>
<th>Description</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>208</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>209</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>210</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. Chose Text as the type of the video column. Save the data source.

3. Add the tab the YouTube button will be placed on and, next to Select Data Source, choose the data source with the video column we created above.
We used the Gallery tab and the data source called *Bed and Breakfast Deluxe.*

4. Drag and drop the YouTube button onto the canvas.

5. In the pop-up window, specify the label, title, font, dimensions, and button style.

6. Next to Source, choose *Data Source.*

7. Next to *Select Url Field*, select *Video.* Click OK.
Here is how the button works on a mobile device.

The room features a private bath, original art, an antique white-mirrored working fireplace, coordinated oriental carpeting, and a sitting area with French carved armchairs.
This button allows you to create in your app and send push notifications right away to all or predefined groups of users. Your users will be able to see all available notifications on their Notifications tab.

The One-Time Notifications button works very similarly to the One-Time Notifications tab. Please click here for more details.
This button allows you to create in your app and send ongoing push notifications to all or predefined groups of users. You can set up your notifications so that your users receive them on a daily, weekly, or monthly basis and specify when exactly these notifications will be sent out to them. Your users will be able to see all available notifications on their Notifications tab.

The Ongoing Notifications button works very similarly to the Ongoing Notifications tab. Please click here for more details.
7.58 Scheduled Notifications

This button allows you to create, schedule and send push notifications from your app to all or predefined groups of users. Your users will be able to see all available notifications on their Notifications tab.

The Scheduled Notifications button works very similarly to the Scheduled Notifications tab. Please click here for more details.

7.59 Geo Tracking

Geo Tracking allows you to calculate distance, create route in real time and view it on the map.

First, you need to create datasource with the “Geotracking” type and save it.
Now let’s add a Geotracking button and check out its settings. The first block is responsible for the look and feel of this button:
The second block contains the settings for geotracking:

Mode is responsible for Start/Stop or Start options. If we select “Start”, we need to add another button. This won’t influence the app behavior but it can help if you want to either use one button to start and stop tracking, or split these two options into two buttons. If you have selected to use two buttons, you can disable or hide the second button, i.e.: 
- if you clicked Start button then the Stop button will be disabled or hidden

OR

- when you clicked Start button, then this button will be unavailable for a new click

Enable notification option is responsible for sending Push Notifications when the tracking status was changed.

Data collection period is responsible for the frequency of data being saved on the device. The less the data collection period, the more often user’s current location has to be saved on device, which will result in more resources used on device.

Data synchronization period is responsible for the frequency of sending the data from the device to the server and has to be equal to or larger than “Data Collection Period” setting.

Binding of coordinates is responsible for setting up a route based on the selected option: road means that the route will be planned by roads; footpath means that the route will be planned through the places where you can walk on foot, i.e. courtyards, alleys, parks, etc.

And the last one is selecting the datasource. Here you can only select the datasource that has “Geotracking” type.

The last block is Data Mapping.
It is needed to add extra fields to the datasource (these extra fields need to be added upon datasource creation first). This is optional, the required set of fields is added by default and cannot be removed.

Let’s consider an example and set up the following options:

![Tracking options](image)

On your device click ON and start to move, then click OFF when you wish to stop. To view the route you can use Advanced Map element. Open its settings and select the datasource with Geotracking type and you will see a new block “Tracking”:
Here you set up permissions for the device to:

1. analyze the geotracking data taken from the database
2. plan a route on a map
3. set up different color schemes for different routes
4. use predefined refresh period for the map to display current location

Here’s what you will see on the map:
After the route is completed, there will be one entry in the datasource. If you want to see more detailed data, you can add Advanced List element to view:

- start time
- end time
- whether the route is active at the moment
- current location
- covered distance

This way you can track the data for all active and/or completed routes. Read how to display data in Advanced List in this chapter.

7.60 New drawing functionality
We have extended drawing functionality so now you can use different figures and signs during
editing. The new features are available for the following elements:

The Sign/Draw field already has the new features by default. To enable these features for the Photo element, you need to turn on a setting:

Let’s consider the changes made on the device. All the new features are presented at the bottom of the screen:
Clicking the element to the left from the palette will open a menu with tools to choose from. It’s convenient to see what you are selecting.

The Hand tool is used to select and change the position of the existing element. You can select free painting, straight line, an arrow, a two-side arrow, circle, square and the ability to add text.
If you click on the palette, you will see the option to select color and transparency:
You can set up the font size which will influence the thickness of all the elements:
To the right you see the 2 arrows that allow you to redo and undo the actions made.

Let’s add a few elements, change their color, width and layout:
You can load a photo from the gallery or take a photo from the camera. If you perform an inspection or inventory, you can quickly add a note or point out a defect right on the picture using the tools.

7.61 Payments in apps (Stripe)
You can make payments using the Stripe payment system in our apps. With this functionality we can now create product catalogs, add items to a shopping cart, purchase goods or services via the app.
First, you need to register at the Stripe website [https://stripe.com/](https://stripe.com/). We will need keys to guarantee security of the payments made to ensure the payments will be sent to you directly. These keys can be found under API in your Account:

Test secret key and test publishable key are needed to test payment in the app.
Live secret key and live publishable key are needed to make real payments when the app is live.

In the App settings that are placed to the left in the Snappii editor, you will see Stripe Payment Settings where you need to insert the keys:

You can insert all the keys but if you only want to test Stripe payments for now, you can leave the Live mode field unchecked. When you use test mode, you will be paying with the virtual money and track the transactions in your Stripe account.

To make virtual payments, you can use this list of test credit cards:
You can learn more about settings and payments here: https://stripe.com/docs/testing#cards.

Let us consider an example and purchase an item with the test card.

Add the keys and enable Stripe:
Let’s create a data source with several items:

<table>
<thead>
<tr>
<th></th>
<th>name</th>
<th>price</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ball</td>
<td>4,5</td>
<td>soccer ball</td>
</tr>
<tr>
<td>2</td>
<td>Pants</td>
<td>6</td>
<td>black jeans</td>
</tr>
<tr>
<td>3</td>
<td>Phone</td>
<td>117,8</td>
<td>new modern phone</td>
</tr>
</tbody>
</table>

and display them on the list:
Now we need to add a Payment Form:

<table>
<thead>
<tr>
<th>Product</th>
<th>Quantity</th>
<th>Price ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ball</td>
<td></td>
<td>4.5</td>
</tr>
<tr>
<td>Soccer ball</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Black jeans</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>Modern phone</td>
<td></td>
<td>117.8</td>
</tr>
<tr>
<td>New modern</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rubber ball</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Payment Form:
This form has Payment Settings block that we need to set up.

To enable purchases we need to add 3 main parameters: item name, price and description:

If we try to save the app at this stage, we will receive this error message:
We didn’t provide the fields to enter the info about the card that a user will use to make payments from. As you can see from this message, you need to have several fields inside the Payment Form:

- Card number
- Code
- Expiration: year and month

This data is also required to make standard transactions. To add these fields, click the Payment Form and drag and drop text/number inputs with a special type:
Now we will need to add the fields and select the corresponding type for each. For the month and year we can use multichoice with options. It will be more convenient for the user, for example:
Of course, you can set up the layout the way you want. Here’s an example:
Now we will save the app without any error messages and see how it looks on the device:
I want to buy black jeans and click Buy It button, but I see a warning message that I’m not logged in in the app:

**Warning**

Log in to your account to complete this action

This is done to improve security. Anonymous users cannot make purchases in the app, so we will need to register or log in and will be redirected to the Payment form:
Let’s insert the test card, select month, year and any CSC. Make sure you enter the card number listed among the test cards, otherwise, you will see the error message.
and click Submit. The payment should be successfully proceeded and you will see a message about it at the bottom of the screen:
In the Stripe account we can check that the payment was received:

If the payment went through, you can turn on Live Mode and submit your app.

Now let’s consider the options to add items to the Shopping Cart. We can leave our items but instead of the payment form we will drag and drop Add to Cart button. We will need to add a few extra fields ID and Image:
We will add the following changes to our spreadsheet:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>id</td>
<td>name</td>
<td>price</td>
<td>description</td>
</tr>
<tr>
<td>2</td>
<td>Ball</td>
<td>soccer ball</td>
<td>4.5</td>
<td><a href="https://dl.dropbox.com/s/35bthhs8b8d2z7fz/ball.jpg">https://dl.dropbox.com/s/35bthhs8b8d2z7fz/ball.jpg</a></td>
</tr>
<tr>
<td>3</td>
<td>Pants</td>
<td>black jeans</td>
<td>6</td>
<td><a href="https://dl.dropbox.com/s/r8d1ooyqdklmoo9u/pants.jpg">https://dl.dropbox.com/s/r8d1ooyqdklmoo9u/pants.jpg</a></td>
</tr>
<tr>
<td>4</td>
<td>Phone</td>
<td>new modern phone</td>
<td>117.8</td>
<td><a href="https://dl.dropbox.com/s/dc4hted4y7qrl/Phone.png">https://dl.dropbox.com/s/dc4hted4y7qrl/Phone.png</a></td>
</tr>
</tbody>
</table>

Now let’s add Shopping cart . This element is available only in a framed view and cannot be added as a button, that’s why we will add it just below the product list.
The Shopping Cart has the following settings:
Besides the shopping cart look and feel you will see a setting to display Pay button. The Shopping cart reminds of the Advanced list where you can set up the cell and detail view to see the items. If this checkmark is enabled, then the Pay button will be available at the top of the list as a common button. You can select the payment method below. If you don’t want to create a new form, you can use our standard form for the payment, it already contains all the required fields. If you want to create your own payment form, you should add Payment form and enable “use shopping cart” option.
Now this form is available for selection in your Shopping Cart. Let’s have the standard form and leave the checkmark next to Shop payment button. You will see that you can set up the cell view either yourself or selecting a predefined template, just like you do it with the Advanced List element. For example, here’s how we have placed the elements on the cell view and:

![Cell view with elements](image)

- **Product list**
- **Subtotal:** $0.00
- **Your shopping cart is empty**
Let’s move to the catalog and add the soccer ball and jeans to the shopping cart:

![Image of shopping cart with soccer ball and jeans added]

You can add this element that will help add the required amount of items. However, if you only sell one item at a time, you are free not to add it. As soon as the amount of items is increased, the Subtotal cost is automatically updated:
Now let’s click Checkout button to see the standard payment form:
You can add a payment form to your tab and lay it out the way you prefer, then enable “Use my shopping cart” option and select your form in the list of Shopping cart settings. On our form you can use the same values for the payments you had used before: card number selected from the list of test cards, expiration month and year, and cvc. When in test mode, it is important to enter valid data for the card number, expiration month and year, as well as the cvc if you are using common fields.

We have reviewed the options to create a onetime payment and use a Shopping cart; learned to create and add Stripe payment keys; tested a standard payment form and created our own form. Using these instructions you can set up payment in your App via Stripe.

Chapter 8. How to Use Fields and Buttons

8.1 Audio Field

The Audio field allows users to record, play back, integrate into PDF and Excel reports, and share with others audio files in the M4A format.

To enable these capabilities, follow the steps below:

1. Drag and drop the Audio field onto the canvas.
2. In the popped up window, specify its name and the web parameter name.

3. Check off the box next to Required if you need to make this element mandatory.

Here’s how the Audio field looks when it’s required and when it’s optional.

<table>
<thead>
<tr>
<th>Required</th>
<th>Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Web Parameter name</td>
<td></td>
</tr>
<tr>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>Width</td>
<td></td>
</tr>
<tr>
<td>Height</td>
<td></td>
</tr>
<tr>
<td>Order</td>
<td></td>
</tr>
<tr>
<td>Next order</td>
<td></td>
</tr>
<tr>
<td>Audio Quality</td>
<td>high</td>
</tr>
<tr>
<td>Maximum duration (seconds)</td>
<td>60</td>
</tr>
<tr>
<td>Column width (for Excel report)</td>
<td>Auto fit</td>
</tr>
<tr>
<td>Select Audio Field</td>
<td>audio</td>
</tr>
</tbody>
</table>

Viewing Conditions
4. Check off the box next to Read only to disable any accidental copying or deletion of this field in the Express Mode.

*Read only enabled*  
*Read only disabled*
On the screenshots above, you can see that when Read only is enabled, the Audio field cannot be copied or deleted, whereas when it’s disabled, you see the copy and delete symbols.

5. Check off the box next to Include Field in Report if the Audio field needs to be added in the PDF/Excel reports upon the completion and submission of each form. (In reports, it will be displayed as the link to the recorded M4A audio file.)

6. Specify its dimensions, order, and next order.

7. Check off the box next to Sharing enabled if you would like to allow your app users to share their audio files separately with the help of the Share button, not only within PDF/Excel reports.
8. Select the audio quality: from high to low.

   Audio Quality
   high
   medium
   low

9. Specify the maximum audio duration.

10. Select one of the three column width options for Excel reports.

    Column width (for Excel report)
    Auto Fit
    Wrap Text
    Fixed

   Select Audio Field
   audio

Auto Fit

Wrap Text
11. Next to Select Audio Field, from the drop down menu, select the Column ID of this field in the data source where the newly added links to recorded audio files will be sent.

In our case, the Audio field is located on the Table field, and next to Select Data Source, we have indicated that the information from the fields located on the Table field will be added to the data source called Audio.
In the Excel file called *Audio*, used as our data source, we use the column *audio* to store generated links to our recorded audio files.

Here is how our data source looks in the data source dialog.

Here is how the Audio field works on a mobile device.
8.2 Barcode Field

The Barcode field allows scanning QR codes, sharing them with others, and importing them into PDF and Excel reports.

To enable this functionality, follow the steps below.

1. Drop the Barcode field onto the canvas wherever you would like.

2. In the popped up window, specify the name of the field and whether this field should be required. (When an element is made required, a user can’t submit the form without filling out that element first.)

<table>
<thead>
<tr>
<th>Name</th>
<th>Bar Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Parameter</td>
<td>BarCodeInput30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Required</th>
<th>Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Here’s how the Barcode field looks when it’s required and when it’s optional.</td>
<td></td>
</tr>
</tbody>
</table>
3. Check off the box next to *Read only* to disable any accidental copying or deletion of this field in the Express Mode.
On the screenshots above, you can see that when Read only is enabled, the Barcode field cannot be copied or deleted, whereas when it’s disabled, you see the copy and delete symbols.

4. Enter Image Parameter and Data Parameter names the way you would like them to appear in the PDF/Excel reports.
5. Specify its dimensions and order.

6. Select whether to use back camera or front camera by default.

7. Check off the box next to **Include Field in Report** if the Barcode field needs to be added in the PDF/Excel reports upon the completion and submission of each form.

8. Check off the box next to **Sharing enabled** if you would like to allow your app users to share their scanned barcodes separately, not only within PDF/Excel reports.
When the **Share button** is added on the same Table element, this option allows sharing barcodes uploaded through the Barcode field with others.

9. Next to **Select Code Image Field**, from the drop down menu, select the Column ID of this field in the data source where the newly added images of scanned barcodes will be sent.

In our case, the Barcode field is located on the Table field, and next to **Select Data Source**, we have indicated that the information from the fields located on the Table field will be added to the data source called **QR**.
In the Excel file called *QR*, used as our data source, we use the column *image1* to store uploaded barcode images from the barcode field.

Here is how our data source looks in the data source dialog.

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>value1</th>
<th>value2</th>
<th>grid</th>
<th>image1</th>
<th>image2</th>
<th>EmailPer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>058400536</td>
<td>9BDD3F58</td>
<td><a href="http://lh5">http://lh5</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>22047880C</td>
<td>993EB0D9</td>
<td><a href="https://s3">https://s3</a></td>
<td><a href="https://s3">https://s3</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
10. Next to *Select Code Data Field*, from the drop down menu, select the Column ID of this field in the data source where the details about newly scanned barcodes will be sent. In our case, it is *value1*.

11. Select one of the three column width options for the Excel reports.

**Auto Fit**
12. If necessary, check off the box next to *Submission right after a code is taken*. For instance, you might want to add the Barcode field at the end of a form and enable instant
submission after all the required fields have been filled out and a barcode has been scanned.

For this option to work properly, make sure you have specified the correct email address for the submission of PDF/Excel reports in the form settings.

Please keep in mind that this option is currently available only for iOS devices.

13. When the Submission right after a code is taken box is selected, a new checkbox that says Allow multiple scanning appears. (This option is also supported by iOS devices only at this point.)

This feature allows you to scan several barcodes before the automatic submission of the form.

Using Text Formula with the QR Code Field

To see the details about the scanned barcodes, add the Text Formula field next to the Barcode field.
In the popped up window, enter the name of the Barcode field from the Barcode field settings in the following format <name>.
8.3 Boolean Button

The Boolean button allows working with simple “Yes” or “No” questions or any other questions when a user needs to select one of the two options. You can configure this button as a Boolean on or check box.

To enable this functionality, follow the steps below.

1. Drag and drop the Boolean button onto the canvas.

2. In the popped up window, specify its name, the web parameter name, and dimensions.
3. If you prefer to have this button configured as a check box, next to Positive image and Negative image, upload the icons of a check box with a checkmark and one without.

4. Enter “Yes” next to Positive value, and “No” next to Negative value.

5. Next to Select Bool Field, from the drop down menu, select the Column ID of this field in the data source where the newly added yes/no (true/false) values will be sent.

In our case, the Boolean button is located on the Form button, and next to Select Data Source in the settings of the Form button, we have indicated that the information from the fields located on the Form will be added to the data source called New Report.

In the Excel file called New report, used as our data source, we use the column Chemicals to store uploaded yes/no (true/false) values from the Boolean button.
Here is how our data source looks in the data source dialog.

6. Enter the order, next order, and select any of the two types: default (boolean) or checkbox.

7. Check off the box next to *Include Field in Report* if data from the Boolean button needs to be added in the PDF/Excel/Word reports upon the completion and submission of each form.

8. Check off the box next to *Read only* to disable any accidental copying or deletion of this button in the Express Mode.
**Read only disabled**

```plaintext
Select devices used

- Did you use chemicals?

Select pests
```

**Read only enabled**

```plaintext
Select devices used

- Did you use chemicals?

Select pests
```
On the screenshots above, you can see that when *Read only* is enabled, the Boolean button cannot be copied or deleted, whereas when it’s disabled, you see the copy and delete symbols.

9. Select one of the three column width options for the Excel reports.
8.4 Calculate Button

The Calculate button is used together with the Form button and formulas. If there is no Calculate button, all formula fields give out the results automatically. However, if you would like certain calculations to be performed upon clicking on the Calculate button, you need to add it to your mobile form to which all the necessary formulas have been added and configured and, in the popped up window, select the formula(s) for which you would like to enable the Calculate button.

For instance, to calculate monthly mortgage payments, we have added a number of fields and formulas. Now we want the Number Calculated Field 243 to display the calculated monthly payment only once the Calculate button is pressed.

Here is how we have configured the button and how it works on a mobile device.
The quick brown fox jumps over the lazy dog.

<table>
<thead>
<tr>
<th>Calculation fields</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number Calculated Field 240</td>
<td>ROUND(((&lt;A&gt; - &lt;D&gt;) * &lt;F&gt; * (POWER((1 + &lt;F&gt;), &lt;C&gt;))) / 100, 2)</td>
</tr>
<tr>
<td>f</td>
<td>(&lt;D&gt; / 12) / 10</td>
</tr>
</tbody>
</table>

Loan amount: $280000
Interest rate: 5%
First installment: $20000
Loan term: (years) 10
Monthly payment: $2757.75
The Clear button enables your users to make all input fields blank. The same functionality is available in the Configure your element section of the Form button.
However, the Clear button under *Fields and Buttons* gives you more flexibility since you can place it wherever you would like and configure it the way you need: you can use the global style, use the default icon, upload a new icon, or opt for a custom style.
The Datasource Formula field is used to work with data sources. Allows app creator to select a scalar value from the data source. For example:

1) first(fieldName) - returns the first record in a specified column;
2) last(fieldName) - returns the last record in a specified column;
3) count() - returns the total number of items in a specified data source;
4) sum(fieldName) - returns the sum of records in specified columns. Note: The column must be numeric;
5) avg(fieldName) - returns the average value of records in specified columns. Note: The column must be numeric;
6) max(fieldName) - returns the maximum element in a specified column. Note: The column must be numeric;
7) min(fieldName) - returns the minimum element in a specified column. Note: The column must be numeric.

To illustrate the above functions, let us create a data source with three fields: name, price, and datetime and add 5 random entries to these fields. Next, we need to upload, configure, and save the data source, add the List button, and then connect the List button to this data source.

1) first(fieldName)

To illustrate the use of the first(fieldName) formula, let us add one DS Calculated field and in its settings, next to Select Data Source, choose the data source we have just uploaded. Then, next to Aggregate function, we need to write first(price).
Here is what we will see on a mobile device:

The first entry (250) appears first, as it should.

2) last(Firstname)

To illustrate this formula, let us change the formula in the DS Calculated field we have created above from first(price) to last(price).
On a mobile device, we will see that our last entry (1250) comes up first, as it should.

3) Count()
This time, in the DS Calculated field we have added above, let us change the formula \texttt{last(price)} to \texttt{count()}. On a mobile device, we see that so far we have made 5 entries, which is the correct number.
The search filters can help us find the number of saved entries satisfying a certain requirement. Let us say, for example, that we need to find the number of entries whose price is lower than 1000.

To do so, in the Search Settings of our DS Calculated field, we should enable the filter, and enter: \textit{price(numeric)} < 1000.

As a result, we will see that we have 3 saved entries satisfying this condition. This is true because the other 3 have higher prices: 1000 and 1250.
4) \texttt{Avg(fieldname)}

This time, in our DS Calculated field we will enter the following formula: \texttt{avg(price)} and limit the number of entries to those whose prices are not equal to 1250, which will leave us with 4 entries. (To illuminate the 5\textsuperscript{th} entry, in the Search Settings section, we will enter: \texttt{price(numeric)!=1250}.)

\begin{center}
\begin{tabular}{ll}
Test1 & 250  \\
12/28/2015 11:17  \\
\hline
Test2 & 500  \\
12/29/2015 11:17  \\
\hline
Test3 & 750  \\
12/31/2015 11:23  \\
\hline
Test4 & 1000  \\
12/30/2015 12:24  \\
\hline
Test5 & 1250  \\
01/08/2016 12:24  \\
\end{tabular}
\end{center}
Here is what we will see on a mobile device.
Let us check: $250+500+750+1000=2500$; $2500/4=625$. So the formula works correctly.

5) `sum(fieldName)`.

To illustrate how this formula works, let us sum up three columns: Test2, Test3, and Test4. To eliminate the other 2 with the lowest and the highest prices, in the Search Settings section, we will enter: `price(numeric)!=250` and `price(numeric)!=1250`.

Let's show a total for Test2, Test3 and Test4.

Here is what we will get on a mobile device.
Let us check: 500+750+100=2250. So the formula works correctly.

6) \text{max(FieldValue)}

Now, to illustrate this function, let us find the maximum value of the first three entries. To do so, we will enter \text{max(price)} next to Aggregate function and, in the Search Settings section, \(\text{price(numeric)} \leq 750\), since the prices of the 4\textsuperscript{th} and 5\textsuperscript{th} entries are higher than 750.
Here is what we will see on a mobile device:
This is true because out of the first 3 entries (250, 500, and 750), 750 has the highest value.

7) \text{min(Fieldname)}

This time, to show how this function works, let us get the minimum value out of Test2, Test3, Test4 and Test5. To eliminate the first entry, in the Search Settings section, we will enter \text{price(numeric)!}=250 since the price of Test1 = 250.

Here is what we will see on a mobile device:
This is true because out of the last 4 entries (500, 750, 1000, and 1250), has the lowest value.

Now let us look at an example of the use of the DS Calculated field in one of our published apps.

*My Income Calculator*
The My Income Calculator app allows our users to calculate their weekly income by entering the days they work, the hours, and their pay rate. The app utilizes the Ds Calculated field with the formula \textbf{sum(TotalN)} to calculate the total amount of money the user made/will make within a particular period of time.

where TotalN is the invisible Number Calculated field located under \textit{New Weekly Plan} and calculating the total user’s income per week. So \textit{sum(TotalN)} calculates the total amount of money the user has made/is planning to make within a given period of time. In our case, we are looking at 4 weeks during the month of July.
Week 1

<table>
<thead>
<tr>
<th>Day</th>
<th>Date</th>
<th>Work hours</th>
<th>Pay per hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mo</td>
<td>07/04/21</td>
<td></td>
<td>35</td>
</tr>
<tr>
<td>Tu</td>
<td>07/05/21</td>
<td>9:00 AM</td>
<td></td>
</tr>
<tr>
<td>We</td>
<td>07/06/21</td>
<td>9:00 AM</td>
<td></td>
</tr>
<tr>
<td>Th</td>
<td>07/07/21</td>
<td>9:00 AM</td>
<td></td>
</tr>
<tr>
<td>Fr</td>
<td>07/08/21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sa</td>
<td>07/09/21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Su</td>
<td>07/10/21</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total: 542.50 $

Week 2

<table>
<thead>
<tr>
<th>Day</th>
<th>Date</th>
<th>Work hours</th>
<th>Pay per hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mo</td>
<td>07/11/21</td>
<td>3:00 PM</td>
<td>35</td>
</tr>
<tr>
<td>Tu</td>
<td>07/12/21</td>
<td>3:00 PM</td>
<td></td>
</tr>
<tr>
<td>We</td>
<td>07/13/21</td>
<td>3:00 PM</td>
<td></td>
</tr>
<tr>
<td>Th</td>
<td>07/14/21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fr</td>
<td>07/15/21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sa</td>
<td>07/16/21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Su</td>
<td>07/17/21</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total: 367.50 $
So, based on the Sum field, by the end of July, we will have made $2,467.50.

If we add our income for each of these 4 weeks manually, we will get the same number:

$542.50 +$367.50 +$735 +$822.50 = $2,467.50

So our formula works correctly.
8.7 Datetime Field

The Datetime field allows users to enter dates and times and then share them together with the rest of the form.

Here is how the Datetime field looks and works on a mobile device.

**Entering a Date into the Datetime Field**
This Datetime Field in the Data Source It Is Linked to

As you can see, to enable entering only a date into the Datetime field, its type should be *Date*.

**Entering a Time into the Datetime Field**
This Datetime Field in the Data Source It Is Linked to

As you can see, to enable entering only a time into the Datetime field, its type should be *Time.*

**Entering a Date and a Time into the Datetime Field**

This Datetime field in the Data Source It Is Linked to
As you can see, to enable entering a date and a time into the Datetime field, its type should be Datetime.

To enable this functionality, please follow the steps below:

1. Drag and drop the Datetime field onto any form you need.

2. In the popped up window, specify the name of the field, its dimensions, and whether it will be required. If you choose to make it required, your app users won’t be able to submit the form on which this field is located without filling it out.

Here’s how the Datetime field looks when it’s required and when it’s optional. (The Date Datetime field is used to illustrate these options.)
3. Next to *Select Field*, from the drop down menu, select the Column ID of this field in the data source where the newly added dates/times/dates and times will be added.

Here is why we selected this field:

A. Our Datetime field is located on the Form button connected to the data source called *Estate Sales*.

B. Here is how our data source looks in the data source dialog.
4. Enter the order in which this field will appear in the report. (The *next order* can be left as is.)

5. Specify the font used in this field or leave the default one.

6. Check off the box next to *Include Field in Report* if the Datetime field needs to be added in the PDF/Excel reports upon the completion and submission of each form.

7. Check off the box next to *Sharing enabled* if you would like to allow your app users to share this field separately with the help of the *Share button*, not only within PDF/Excel reports.

8. Specify its placeholder if you wish.
9. Put a check mark next to *Save input history* to be able to auto-populate this field with any of the previously entered dates/times/dates and times.

10. Select one of the three column width options for the Excel reports.
8.8 Datetime Formula Field

The Datetime formula field works with dates and times. That is why it supports only three data types: date, time, and datetime.
**Name** (show description)
- Weekly Report

**Data Storage**
- Global

**Type**
- Data Table

**Time zone**
- UTC0

**Fields:**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Type</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tu_date</td>
<td>Tuesday date</td>
<td>date</td>
<td></td>
</tr>
<tr>
<td>Tu_status</td>
<td>Tuesday status</td>
<td>text</td>
<td></td>
</tr>
<tr>
<td>We_date</td>
<td>Wednesday date</td>
<td>date</td>
<td></td>
</tr>
<tr>
<td>We_status</td>
<td>Wednesday status</td>
<td>text</td>
<td></td>
</tr>
<tr>
<td>Th_date</td>
<td>Thursday date</td>
<td>date</td>
<td></td>
</tr>
<tr>
<td>Th_status</td>
<td>Thursday status</td>
<td>text</td>
<td></td>
</tr>
<tr>
<td>Fr_date</td>
<td>Friday date</td>
<td>date</td>
<td></td>
</tr>
<tr>
<td>Fr_status</td>
<td>Friday status</td>
<td>text</td>
<td></td>
</tr>
<tr>
<td>Sa_date</td>
<td>Saturday date</td>
<td>date</td>
<td></td>
</tr>
<tr>
<td>Sa_status</td>
<td>Saturday status</td>
<td>text</td>
<td></td>
</tr>
<tr>
<td>Su_date</td>
<td>Sunday date</td>
<td>date</td>
<td></td>
</tr>
<tr>
<td>Su_status</td>
<td>Sunday status</td>
<td>text</td>
<td></td>
</tr>
</tbody>
</table>

**Development Environment**
- [Upload]
- [Append]
- [Clear]
- [Download]
- [SQL]

**Production Environment**
- [Download]
- [Transfer Development Data]

**Info:**
- [Submit logo]
- [Submit video]
- [Submit Number of videos]
As we know, the date type has the following format: MM/DD/YYYY; the time type - HH:MM AM/PM, and the datetime - MM/DD/YYYY HH:MM AM/PM.

This is how these formats look on a mobile device:

**Date**

**Time**
Date and a Time

The Datetime formula supports the following operations:

1) Arithmetic operations with dates and time
YEARS, MONTHS, WEEKS, DAYS, HOURS, and MINUTES are used.

For example, we can utilize the following formula:

\[
\text{<Date>} + 10 \text{ DAYS}
\]

To give you an example, let us add 2 Datetime fields: one of the date type and the other of the time type. In the first Datetime Formula field, let us use the following formula:

\[
\text{<date>} + 10 \text{ DAYS},
\]

where \textit{date} is the name of the first Datetime field.
Next, let us add one more Datetime Formula field and enter the following formula:

\[
<\text{time}> - 2 \text{ HOURS},
\]

where \textit{time} is the name of the second Datetime field.
<table>
<thead>
<tr>
<th>Name</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>time</td>
</tr>
<tr>
<td>Height</td>
<td>35</td>
</tr>
<tr>
<td>Width</td>
<td>270</td>
</tr>
<tr>
<td>Web Parameter name</td>
<td>DateTimeField140</td>
</tr>
<tr>
<td>Order</td>
<td>2</td>
</tr>
<tr>
<td>Next order</td>
<td>2</td>
</tr>
<tr>
<td>Font</td>
<td>Default</td>
</tr>
<tr>
<td>Placeholder</td>
<td>Time</td>
</tr>
<tr>
<td>Column width (for Excel report)</td>
<td>Auto fit</td>
</tr>
</tbody>
</table>

| Name                  | Time minus 2 hours                |
| Web Parameter name    | DateTimeCalculatedField139        |
| Formula               | `<time> - 2 HOURS`                 |
| Height                | 35                                |
| Width                 | 270                               |
| Order                 | 4                                 |
| Font                  | Default                           |
| Placeholder           | Time minus 2 hours                |
| Font size             | 14px                              |
| Alignment             | Left                              |
| Font color            | `#333333`                         |
| Format of result      | time                              |
| Column width (for Excel report) | Auto fit                        |
Please keep in mind that the field names should be entered in the formulas exactly the way they are written. For instance, if your field is called *Date*, in the formula, it should say `<Date>`. If it is *date*, it should say `<date>`, etc.

Here is how these two formulas work on a mobile device:

Please keep in mind that, while working with the Datetime field, you need to correctly indicate the format of the result. In our case, for the first Datetime Formula field, we selected *date*, and for the second - *time*.
2) Constant operations
TODAY brings today’s date, while NOW brings the current time or date and time.

Here is how we configured each of the fields above.
Here again it is essential to select the correct format of your result. Thus, our 2\textsuperscript{nd} and 3\textsuperscript{rd} fields utilize the same formula (NOW) but have a different format: the first – time, and the second – datetime.

3) \textbf{Manual setting dates and times}

The following formulas are used here:

\begin{align*}
\text{TO\_DATETIME("mm/dd/yyyy hh:mm am"), TO\_DATE("mm/dd/yyyy"), TO\_TIME("hh:mm am")}. \\
\end{align*}

For example,

- \text{TO\_DATETIME("09/01/2016 09:05 AM")}

- \text{TO\_DATE("09/01/2016") \_ TO\_TIME("09:05 AM").}

- \text{TO\_TIME("09:05 PM").}

Here is how our formulas work on a mobile device:

\begin{itemize}
\item \text{09/01/2016}
\item \text{9:05 AM}
\item \text{9:05 PM}
\item \text{09/01/2016 9:05 AM}
\end{itemize}

Here is how we configured each of these 4 fields:
### View

<table>
<thead>
<tr>
<th>Name</th>
<th>To Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Parameter name</td>
<td>DateTimeCalculatedField148</td>
</tr>
<tr>
<td>Formula</td>
<td><code>TO_DATE('09/01/2016')</code></td>
</tr>
<tr>
<td>Height</td>
<td>15</td>
</tr>
<tr>
<td>Width</td>
<td>270</td>
</tr>
<tr>
<td>Order</td>
<td>1</td>
</tr>
<tr>
<td>Font</td>
<td>Default</td>
</tr>
<tr>
<td>Placeholder</td>
<td>To Date</td>
</tr>
<tr>
<td>Font size</td>
<td>14px</td>
</tr>
<tr>
<td>Alignment</td>
<td>Left</td>
</tr>
<tr>
<td>Font color</td>
<td></td>
</tr>
<tr>
<td>Format of result</td>
<td>date</td>
</tr>
<tr>
<td>Column width (for Excel report)</td>
<td>Auto fit</td>
</tr>
</tbody>
</table>

### View

<table>
<thead>
<tr>
<th>Name</th>
<th>To Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Parameter name</td>
<td>DateTimeCalculatedField148</td>
</tr>
<tr>
<td>Formula</td>
<td><code>TO_TIME('09:05 am')</code></td>
</tr>
<tr>
<td>Height</td>
<td>15</td>
</tr>
<tr>
<td>Width</td>
<td>270</td>
</tr>
<tr>
<td>Order</td>
<td>1</td>
</tr>
<tr>
<td>Font</td>
<td>Default</td>
</tr>
<tr>
<td>Placeholder</td>
<td>To Time</td>
</tr>
<tr>
<td>Font size</td>
<td>14px</td>
</tr>
<tr>
<td>Alignment</td>
<td>Left</td>
</tr>
<tr>
<td>Font color</td>
<td></td>
</tr>
<tr>
<td>Format of result</td>
<td>time</td>
</tr>
<tr>
<td>Column width (for Excel report)</td>
<td>Auto fit</td>
</tr>
</tbody>
</table>
Here again selecting the correct format of our results was critically important.

Here is how some of the above formulas are used in our published apps.
Construction Daily Log
As you can see from the screenshots above, the Datetime Formula field with the NOW formula auto-populates the Datetime field called *Date* the moment our users open this mobile form. The Datetime Formula field is also made invisible so that our users can only see the Datetime field and not the formula field.

Naturally, this formula is used to save our users a little bit of time. However, they can always change the date by simply clicking on the Datetime field and selecting their own date.
The My Income Calculator app allows our users to calculate their weekly income by entering the days they work, the hours, and their pay rate. This app utilizes a variety of formulas, which we discuss in other sections of this manual. However, it uses only one datetime formula: `<date> +1 DAYS`

In this app, the user has to manually select the date for Monday, and then, the consecutive dates for the rest of the week appear.
To enable this capability, each Datetime field from Tuesday through Sunday has to have the same formula <Date> + 1 DAYS, where <Date> is the name of the previous Datetime field.

For instance, the Tuesday Datetime field utilizes the following formula:
because Monday has the following Datetime field:

For the Wednesday Datetime field, we enter:

because Tuesday’s Datetime field is called *Tuesday date*.

The same sequence holds true for the rest of the days.
8.9 Document Upload Button

The Document Upload button allows you to upload files of any format to store, edit/fill out (currently available only for PDF files), and share them with others.

To enable this functionality, please follow the steps below:

1. Drag and drop the Document Upload button onto any form you need.

2. In the popped up window, specify the name of the button and whether it will be required. If you choose to make it required, your app users won’t be able to submit the form on which this button is located without uploading any file.

3. Check off the box next to Read only to disable any accidental copying or deletion of this field in the Express Mode.
On the screenshots above, you can see that when Read only is enabled, the Document Upload button cannot be copied or deleted, whereas when it’s disabled, you see the copy and delete symbols.
4. Put a checkmark next to *Include Field in Report* to include the URLs of uploaded through this button documents into PDF/Excel reports.

The report settings are configured in the settings of the Form button on which the Document Upload button is located.

Under *Report Common Settings*, you can specify the report name, what information will be included into it (user information and submission timestamp), the body text, and whether default or custom PDF and Excel report templates will be used.
Here is an Excel report with the Document Upload button included and without.

**Included**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime (UTC 0)</td>
<td>User name</td>
<td>User email</td>
<td>Template Name</td>
</tr>
<tr>
<td>6/11/2016 7:49:26 PM</td>
<td>yulia smith</td>
<td><a href="mailto:yulia@snappii.com">yulia@snappii.com</a></td>
<td>Inspection</td>
</tr>
<tr>
<td>6/11/2016 7:49:26 PM</td>
<td>yulia smith</td>
<td><a href="mailto:yulia@snappii.com">yulia@snappii.com</a></td>
<td>Inspection</td>
</tr>
</tbody>
</table>

**Without**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime (UTC 0)</td>
<td>User name</td>
<td>User email</td>
<td>Template Name</td>
</tr>
<tr>
<td>6/11/2016 7:49:26 PM</td>
<td>yulia smith</td>
<td><a href="mailto:yulia@snappii.com">yulia@snappii.com</a></td>
<td>Inspection</td>
</tr>
<tr>
<td>6/11/2016 7:49:26 PM</td>
<td>yulia smith</td>
<td><a href="mailto:yulia@snappii.com">yulia@snappii.com</a></td>
<td>Inspection</td>
</tr>
</tbody>
</table>

5. Specify the dimensions of the button, its order and next order.

- **Width**: 180
- **Height**: 200
- **Order**: 2324
- **Next order**: 2324

6. Specify the image field name and the document field name.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image field name</td>
<td>Image</td>
</tr>
<tr>
<td>Document field name</td>
<td>Document</td>
</tr>
<tr>
<td>Select Field to store PDF URL</td>
<td>DocUpload</td>
</tr>
<tr>
<td>Select Field to store PDF Image</td>
<td>DocImage</td>
</tr>
</tbody>
</table>

7. Next to **Select Field to store PDF URL** and **Select Field to store PDF Image**, from the dropdown menus, select the Column IDs of these fields in the data source where the newly added PDF and image URLs will be sent.
In our case, the Upload Document button is located on the Form button, and in the settings of the Form button, under *Submit to Data Source*, next to *Select Data Source*, we have indicated that the information from the fields located on this form will be added to the data source called *Upload Template*.

In the Excel file called *Upload Template*, used as our data source, we use the columns *DocUpload* and *DocImage* to store generated links to our uploaded documents and their images.

Here is how our data source looks in the data source dialog.

Here is how these fields look in the Fields Mapping.
8. Next to Select Storage, specify the cloud storage location to send and store all files uploaded and saved via the Document Upload button. By default, they will be stored at the Snappii Cloud. You can add and configure your cloud storage preferences in the settings under Cloud Storage, to the left of the canvas.

Please click here for more details.
Once you have added and saved your cloud storage options, they will appear in the settings of the Upload Document button in the drop-down menu next to Select Storage.

9. Next to Format of document to upload, specify whether your users will be able to upload only PDF files (select PDF) or any other file formats (select any).

10. Select the column width for this field in Excel reports.

11. If you would like to enable your users to only upload files stored on their mobile devices (local storage), leave the box next to Use this input to import document from another app unchecked. To allow them to upload files from their cloud storage accounts (cloud storage), please put a check mark next to this option.

With this box checked

With this box left unchecked
12. Put a check mark next to *Show action after a new element is added* if you would like the action buttons specified in the settings of the form on which the Document Upload button is located to appear every time a new file gets uploaded. Configure your element

For instance, under *Configure your element*, we selected only the Clear button. When we choose to show this action after a new element is added, we see the following screen.

When we leave this box unchecked, we don’t see the Clear button on the top of the screen.
13. Next to Help message mode,

specify whether the help message will be available only for first time users,

whether the help button will be displayed on the navigation bar,
whether both the help message and help button will be available,
Here is what your users will see once they click on the Help button or See more of the help message.

14. You can either use our default help message (see above) or upload your own one.
In order to do so, you will need to create and upload an HTML document with the information you would like to be displayed. This HTML document needs to be archivated. You can download an example by clicking on Download.

For the Document Upload button to work correctly, together with the Form button on which the Document Upload button is located, you need to add one List button to display uploaded and saved documents and allow your users to edit them and one Report List button to display and share completed documents/reports.
Inside the Report List Button
8.10 Location Field

The Location field allows users to integrate locations into their reports. Thus, for instance, a building inspector will find it useful to be able to include the locations of inspected sites by either manually entering them or having the app automatically detect them.

To enable this functionality, follow the steps below.

1. Drop the Location field onto the canvas wherever you would like.

2. In the popped up window, specify the name of the field and whether this field should be required. (When an element is made required, a user can’t submit the form without filling out that element first.)

Here’s how the Location field looks when it’s required and when it’s optional.
3. Check off the box next to Read only to disable any accidental copying or deletion of this field in the Express Mode.
On the screenshots above, you can see that when *Read only* is enabled, the Location field cannot be copied or deleted, whereas when it’s disabled, you see the copy and delete symbols.

4. Check off the box next to *Include Field in Report* if it needs to be added in the PDF/Excel reports upon the completion and submission of each form.

5. Specify its dimensions, order, and next order.

6. Enter latitude, longitude, and address field names the way you would like them to appear in PDF/Excel reports.
Check off the box next to *Include map image* for the image to appear in the reports.

7. Include Map Image enabled

8. Include Map Image disabled
The screenshots above illustrate how by selecting the Include Map Image option, we add the image of the identified location on the map.

8. Specify your name of the Map image field.
9. Check off the box next to *Sharing enabled* if you would like to allow your app users to share their map locations separately, not only within PDF/Excel reports.

When the [Share button](#) is added on the same Table element, this option allows sharing locations uploaded through the Location field with others.
10. Next to *Select Location Field*, from the drop down menu, select the Column ID of this field in the data source where newly added latitude and longitude coordinates of identified/specifyied locations will be sent.

![Image of a data selection interface]

In our case, the Location field is located on the Table field, and next to *Select Data Source*, we have indicated that the information from the fields located on the Table field will be added to the data source called *Location*.

![Image of a data source selection interface]

In the Excel file called *Location*, used as our data source, we use the column *Location* to store uploaded latitude and longitude coordinates from the Location field.
Here is how our data source looks in the data source dialog.

11. Next to Select Address Field, from the drop down menu, select the Column ID of this field in the data source where newly added address of identified/specified locations will be sent. Since we don’t have such a field, we selected “Don’t use.”

12. Next to Select Image Field, from the drop down menu, select the Column ID of this field in the data source where newly added location images will be sent. In our case, it is LocationImage.
13. Select one of the three column width options for the Excel reports.
Wrap Text

Fixed

14. Specify if you would like to enable the app to use your user’s current location by default.
As you can see from the screenshots above, when we enable this function, the user’s current location gets automatically detected once he/she opens the location of the field.
8.11 Multi-Choice

The Multi-Choice field allows you to create a list of options to choose from.

To enable this capability, follow the steps below:

1. Drag and drop the Multi-Choice field onto the canvas.

2. In the popped up window, in the View Section, specify its name and the web parameter name.
3. Enter the placeholder if you wish.

With the Placeholder

Without the Placeholder

4. Check off the box next to Read only to disable any accidental copying or deletion of this field in the Express Mode.
Read only enabled

Read only disabled

On the screenshots above, you can see that when Read only is enabled, the Multi-Choice field cannot be copied or deleted, whereas when it’s disabled, you see the copy and delete symbols.

5. Specify its dimensions, order, next order, and font.

6. Select one of the three column width options for Excel reports.
7. In the Choices section, next to *Selection type*, indicate whether your app users will be able to choose one option or several.

A Single Choice

Multiple Choices

8. Next to *Submit Field*, choose the column from your data source connected to the MultiChoice field where your users’ answers will be added and stored. In our case, it is *Weather*. 
Our Multi-Choice field is located on the Form, and next to Select Data Source, we have indicated that the information from the fields located on the Form will be added to the data source called *New Daily Report Form*.

In the Excel file called *New Daily Report Form*, used as our data source, we use the column *weather* to store our users’ answers.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Contract</td>
<td>Project Name</td>
<td>Address</td>
<td>State</td>
<td>WorkPerformed</td>
<td>SubContractor</td>
<td>IssuesAndDelays</td>
<td>ExtraWorkRequests</td>
<td>MaterialPurchased</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Here is how our data source looks in the data source dialog.
9. Now indicate whether you will provide your list of choices here or have the app pull them from a specified data source.

10. In the Configure Your Element Section, put a checkmark next to allow Adding Users’ Options to enable them to add their own choices to supplement the list of options provided by you.
Allow Adding Users’ Options Enabled

Allow Adding Users’ Options Disabled
11. If in the Choices section you have chosen the Single selection type, a new option becomes available to you: multiline entry output.

12. Check off the box next to Required if you need to make this element mandatory.

Here's how the Audio field looks when it's required and when it's optional.

**Required**

**Optional**

13. Check off the box next to Include Field in Report if the Multi-Choice field needs to be added in the PDF/Excel reports upon the completion and submission of each form.
14. Check off the box next to *Sharing enabled* if you would like to allow your app users to share their answers separately with the help of the *Share button*, not only within PDF/Excel reports.

15. In the Instructions/Validation section
The Number Calculated field allows making calculations with numbers.

This is how the settings of the Number Calculated field look.

Some of the unique settings for this field, along with the other formula fields, are:

1) The formula field, where you enter the formula this field will utilize;
2) Visible – some Number Calculated fields are visible to the user; the others are made invisible since their purpose is to perform internal calculations and to give data to the visible Number Calculated fields. Click here to see an example of such use.

3) Frame shown:

(Please keep in mind that for this option to work, first the field needs to be made visible.)

In addition to such simple arithmetic operations as addition, subtraction, and multiplication, the field supports more sophisticated formulas, like the ability to calculate the distance based on two GPS locations.

Let us take these formulas, for example:

This formula calculates the distance between two locations based on their latitude and longitude coordinates.
This formula calculates the distance between the user’s current location and another specified location. This is an example of the use of these two formulas in a live app.

As we have already said, the Number Calculated field works with data of the *numeric* type.

However, it can still utilize data from textual fields if we use the following formula:

TO_NUMBER(<fieldname>), which converts textual data into numeric data.

For instance, this app allows users to calculate monthly mortgage payments and uses the following formula:
The variables are as follows:

- M = monthly mortgage payment  (First Number Calculated field)
- P = the principal, or the initial amount you borrowed  (Field A - Field D)
- r = your monthly interest rate. Your lender likely lists interest rates as an annual figure, so you’ll need to divide by 12, for each month of the year. So, if your rate is 5%, then the monthly rate will look like this: 0.05/12 = 0.004167 (Field B divided by 12 = Field F)
- n = the number of payments, or the payment period in months. If you take out a 30-year fixed rate mortgage, this means: n = 30 years x 12 months per year, or 360 payments. (Field L)

Here is the way the formula above is written in the app.

\[ M = P \frac{r(1+r)^n}{(1+r)^n - 1} \]
Here Field G is used solely to convert the textual Field M to a number with the formula
TO_NUMBER (<M>).

The quick brown fox jumps over the la.
Here are additional most frequently used formula operations:

1) Arithmetic operations:
   - Addition +
   - Subtraction -
   - Multiplication *
   - Division /
   - Finding percentages %
   - Finding the modulus (modulo)
   - Using the ampersand &
   - Using the vertical line |
   - Finding exponents ^
   - Using the (exclusive or)
   - Using the << and >> symbols. Let us look at addition.

We use Field 1 and Field 2 to enter some information. To add the values of these fields, in the formula of the Number Calculated field, we enter

<field1> + <field2>
Here is how this formula works on a mobile device.

That was a very simple formula. However, if you follow the basic rules of the order of operations
(Rule 1: First perform any calculations inside parentheses. Rule 2: Next perform all multiplications and divisions, working from left to right. Rule 3: Lastly, perform all additions and subtractions, working from left to right), you can work with more complex formulas, such as:

\[(f1) + (f2) \div f3 \times f4\]

2) Finding the sum of the whole table column

The Number Calculated field allows finding the sum of the whole table column.

For instance, if we need to calculate the total cost of all ordered menu items, we will use this formula:

`SUM(<TableName.FieldName>).`

Here is how this formula works on a mobile device:
Here is how one of our published apps (*Construction Manager App*) utilizes this formula:

A. Total Labor
where WorkDescriptionCosts is the name of the table, and Labor Total is the name of the field whose values are summed up.
B. Total Parts

where \textit{PartsCosts} is the name of the table, and \textit{Total Part} is the name of the field whose values are summed up.
C. Total Equipment

where \textit{EquipmentCosts} is the name of the table, and \textit{Total Rent} is the name of the field whose values are summed up.
3) Calculating the distance between two locations

As we have described above, the formula DISTANCE(<Location1>, <Location2>) is used to calculate the distance between two locations based on their latitude and longitude coordinates.
4) Calculating the number of entries in a specified data source

Here is how the TOTAL formula woks on a mobile device.

The Total Number Calculated field calculates the number of previously saved entries. In this case, we have made three entries, which is proved by our List button:
5) Converting dates into the number of days

IN_DAYS(<date2> - <date1>) turns dates into the number of days.

Here is how Snappii’s My Income Calculator app utilizes this formula:
The quick brown fox jumps over the la...
The quick brown fox jumps over the lazy dog.
6) Rounding to the nearest tenth, hundredth, and thousandth

ROUND(<Fieldname>,n) rounds a given value to:

- Tenth when n=1
- Hundredth when n=2
- Thousandth when n=3, etc.

Let us look at the app calculating the distance between two specified locations and indicate that the distance should be rounded to the nearest tenth. In the Number Calculated field called Distance, we use the following formula:

Here is how this formula works on a mobile device.

Let us look at how our published app Billable Hours utilizes this formula.
Rounding to the Nearest Tenth

<table>
<thead>
<tr>
<th>Name</th>
<th>Total Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Parameter name</td>
<td>NumberCalculateField78</td>
</tr>
<tr>
<td>Formula</td>
<td><code>ROUND(&lt;Hours&gt;-1)</code></td>
</tr>
<tr>
<td>Height</td>
<td>55</td>
</tr>
<tr>
<td>Width</td>
<td>180</td>
</tr>
<tr>
<td>Order</td>
<td>179</td>
</tr>
<tr>
<td>Font</td>
<td>Arial</td>
</tr>
<tr>
<td>Select field</td>
<td>totalHours</td>
</tr>
<tr>
<td>Placeholder</td>
<td></td>
</tr>
<tr>
<td>Font size</td>
<td>10px</td>
</tr>
<tr>
<td>Alignment</td>
<td>Left</td>
</tr>
<tr>
<td>Font color</td>
<td></td>
</tr>
</tbody>
</table>

The quick brown fox jumps over the lazy dog.

---

**Create an invoice**

**Employee**
- **Yulia**

**Service category**
- **Design**

**Select currency**
- **$**

**Hourly rate ($)**
- **35**

**Start time**
- **06/16/2016 9:44 PM**

**End time**
- **06/21/2016 9:52 PM**

**Total hours**
- **120.1**

**Non-billable hours**
- ****

**Billable hours**
- ****
Rounding to the Nearest Hundredth

<table>
<thead>
<tr>
<th>Name</th>
<th>Total Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Parameter name</td>
<td>NumberCalculateField(79)</td>
</tr>
<tr>
<td>Formula</td>
<td>ROUND(&lt;Hours&gt;, 2)</td>
</tr>
</tbody>
</table>

The quick brown fox jumps over the lazy dog.
Rounding to the Nearest Thousandth

The quick brown fox jumps over the la...

Create an invoice

Select currency: $

Hourly rate ($): 35

Start time: 06/16/2016 9:44 PM
End time: 06/21/2016 9:52 PM

Total hours: 120.133
Non-billable hours

Billable hours

Total cost ($):

Description:

Report date
Signature
Now let us look at the way some of our published apps use a number of the described above formulas at once:

**Billable Hours – Subtraction**

The quick brown fox jumps over the lazy dog.
**Billable Hours**

**Total Cost – Multiplication**

<table>
<thead>
<tr>
<th>Name</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Parameter name</td>
<td>NumberCalculatedField21</td>
</tr>
<tr>
<td>Formula</td>
<td>988</td>
</tr>
</tbody>
</table>

The quick brown fox jumps over the lazy dog.

![Invoice creation interface](image)

Select currency: $ 35

| Total hours | 120.133 |
| Non-billable hours | 10 |
| Billable hours       | 110.133 |
| Total cost ($)       | 3854.655 |

Description

![Invoice creation interface](image)
Multiplication plus Rounding

The quick brown fox jumps over the la...
**Warehouse Lease and Capacity Calculator**

**Max Height – Subtraction**

![Formula Image]

- Name: Max Height
- Web Parameter name: NumberCalculatedField208
- Formula: `<Height> - 3`
- Height: 35
- Width: 140
- Order: 208
- Font: Arial
- Select Field: `max`
- Font size: 14px
- Alignment: Left

*The quick brown fox jumps over the la...*

![Calculator Image]

**Warehouse Lease and Capacity Calculator**
Warehouse Capacity: Multiplication, Subtraction, and Division

The quick brown fox jumps over the la...
Grand Total: Addition, division, multiplication, and subtraction

My Income Calculator

Hours – Rounding, Subtraction, and Division
(Since F1 and F2 use seconds, not minutes or hours, here we divide the total sum by 3600 to convert the total sum into hours)
The quick brown fox jumps over the lazy dog.
My Income Calculator

Total – Multiplication and Rounding
A. First we calculate the number of seconds.
B. Next, we turn the seconds into hours by dividing them by 3600.

C. Then we multiply the hours by the cost per hour.

D. Finally, we round the total cost to the hundredth. In the screenshot above, you can see that this is the only visible field. The others perform the calculations necessary for this field but cannot be seen by the user.

Construction Calculators for Building Professionals App

![Room Size Calculator Image]

**Formula**

```
(<!t:=="US" ? ROUND(((<Width feet>*12)+<Width inches>), 2) : ROUND(((<Width meter>*100)+<Width cm>), 2))
```
<table>
<thead>
<tr>
<th><strong>View</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>Web Parameter name</strong></td>
</tr>
<tr>
<td><strong>Data type</strong></td>
</tr>
<tr>
<td><strong>Required</strong></td>
</tr>
<tr>
<td><strong>Width</strong></td>
</tr>
<tr>
<td><strong>Height</strong></td>
</tr>
<tr>
<td><strong>Order</strong></td>
</tr>
<tr>
<td><strong>Next order</strong></td>
</tr>
<tr>
<td><strong>Font</strong></td>
</tr>
<tr>
<td><strong>Placeholder</strong></td>
</tr>
<tr>
<td><strong>Column width (for Excel report)</strong></td>
</tr>
<tr>
<td><strong>Save input history</strong></td>
</tr>
<tr>
<td><strong>Include Field in Report</strong></td>
</tr>
<tr>
<td><strong>Sharing enabled</strong></td>
</tr>
<tr>
<td><strong>Max length (or max number value)</strong></td>
</tr>
</tbody>
</table>

**Auto-Fill Options**

**Viewing Conditions**

**Formula:**

- ✔ **Use formula to show/hide element**
- Use formula to show element

**Formula:**

<st> == "US"
The quick brown fox jumps over the lazy dog.

**View**

- **Name**: TextArea 7808
- **Text**: Total Square Cm
- **Font size**: 16px
- **Alignment**: Left
- **Font color**: #2D2F30

**Viewing Conditions**

- Use formula to show/hide element
- Use formula to show element: `<st> == "metric"`
  - Automatically move other controls up to fill space when this control is hidden
Total Square Inches

The quick brown fox jumps over the lazy dog.
To enable this functionality, follow the steps below:

1. Drag and drop the Photo field onto the desired tab/element.
2. In the popped up window, specify its name and the web parameter name.
3. Check off the box next to Required if you need to make this element mandatory.
Here’s how the element looks when it’s required.
4. Check off the box next to *Read only* to disable the copying or deletion of this field in the Express Mode by accident.

On the screenshots above, you can see that when *Read only* is enabled, the Photo field cannot be copied or deleted, whereas when it’s disabled, you see the copy and delete symbols.
5. Check off the box next to *Include Field in Report* if it needs to be added in the PDF/Excel reports upon the completion and submission of each form.

6. Specify its dimensions, order, and next order.

7. Choose the photo sources: select *Camera* if you would like to enable your users to take photos within the app and imbed them into reports; select *Library* to allow your users to choose previously taken/downloaded photos.

![Photo selection window](image)

We have chosen both options, so when users click on the Photo field, the following window pops up:
8. Check off the box next to *Allow memos on a photo* to enable the sign/draw capability above uploaded photos.
Once a user has uploaded a photo, he/she needs to click on the photo once for the following menu to appear:

To use the sign/draw feature, the user needs to click on *Draw* and then choose a color and brush size to sign/write something on the photo.
9. Next to *Select Image Field*, select the Column ID of this field in the data source where newly added photos will be sent. In our case, the Photo field is located on the Form button, and in the Submit to Data Source settings, we have indicated that we need to use Data Sources and the information from the fields located on the Form button will be added to the data source called *Farmer's data present*.

In the Excel file called *Farmer's data present*, used as our data source, we use the column *EmblemOrPhoto* to store uploaded photos from the Photo field.
Here is how our data source looks in the data source dialog.

<table>
<thead>
<tr>
<th>Name (show description)</th>
<th>Farmer's data parent</th>
<th>Type</th>
<th>Data Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date Storage</td>
<td>Global</td>
<td></td>
<td>UTC 0</td>
</tr>
</tbody>
</table>

**Fields:**
- WorkHoursFrom
- WorkHoursTo
- EmailToContact
- Facebook
- Twitter
- EmailLogin
- Password
- EmblemOrPhoto
- Location
- EmailForShare

10. Check off the box next to *Sharing enabled* if you would like to allow your users to share their photos separately, not only within PDF/Excel reports.
When the **Share button** is added on the same Table element, this option allows sharing photos uploaded through the Photo field with others.

11. Check off the boxes next to *Use URL instead of image* if necessary.
12. Specify the scale of the image and column width.

13. If necessary, check off the box next to Submission right after a photo is taken. For instance, you might want to add a photo field at the end of a form and enable instant submission after all the required fields have been filled out and a photo has been taken/uploaded. For this option to work properly, make sure you have specified the correct email address for the submission of PDF/Excel reports in the form settings.

Please keep in mind that this option is currently available only for iOS devices.

14. When the Submission right after a photo is taken box is selected, a new checkbox that says Allow multiple photos appears.
This option allows you to take/upload several photos before the automatic submission of the form.

Like the previous option, this one is also only supported by iOS devices at this moment.

There is also a video tutorial available for the Photo field. Click here to watch it.
8.14 QR Code Field

The QR Code field allows scanning QR codes, sharing them with others, and importing them into PDF and Excel reports.

To enable this functionality, follow the steps below.

1. Drop the QR Code field onto the canvas wherever you would like.

2. In the popped up window, specify the name of the field and whether this field should be required. (When an element is made required, a user can’t submit the form without filling out that element first.)

Here’s how the QR Code field looks when it’s required and when it’s optional.
3. Check off the box next to *Read only* to disable any accidental copying or deletion of this field in the Express Mode.
On the screenshots above, you can see that when Read only is enabled, the QR Code field cannot be copied or deleted, whereas when it’s disabled, you see the copy and delete symbols.

4. Enter Image Parameter and Data Parameter names the way you would like them to appear in the PDF/Excel reports.
5. Specify its dimensions, and order.

6. Select whether to use back camera or front camera by default.

7. Check off the box next to Include Field in Report if the QR Code field needs to be added in the PDF/Excel reports upon the completion and submission of each form.

8. Check off the box next to Sharing enabled if you would like to allow your app users to share their scanned QR codes separately, not only within PDF/Excel reports.
When the **Share button** is added on the same Table element, this option allows sharing QR codes uploaded through the QR Code field with others.

9. Next to **Select Code Image Field**, from the drop down menu, select the Column ID of this field in the data source where the newly added images of scanned QR codes will be sent.

In our case, the QR Code field is located on the Table field, and next to **Select Data Source**, we have indicated that the information from the fields located on the Table field will be added to the data source called **QR**.
In the Excel file called *QR*, used as our data source, we use the column *image2* to store uploaded QR Code images from the QR Code field.

Here is how our data source looks in the data source dialog:

```
<table>
<thead>
<tr>
<th></th>
<th>value1</th>
<th>value2</th>
<th>grid</th>
<th>image1</th>
<th>image2</th>
<th>EmailPer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>068400536741</td>
<td>9BDD3F5E</td>
<td><a href="http://lh5">http://lh5</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2204788001565</td>
<td>items://itunes.app 993EB0D9</td>
<td><a href="https://s3">https://s3</a></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
10. Next to *Select Code Data Field*, from the drop down menu, select the Column ID of this field in the data source where the details about newly scanned QR codes will be sent. In our case, it is *value2*.

11. Select one of the three column width options for the Excel reports.

*Auto Fit*
Wrap Text

<table>
<thead>
<tr>
<th>My Bar / QR Codes : Bar Code Data</th>
<th>My Bar / QR Codes : QR Code Image</th>
<th>My Bar / QR Codes : QR Code Data</th>
<th>My Bar / C</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://schnappii-producti">http://schnappii-producti</a>...</td>
<td><a href="http://schnappii-producti">http://schnappii-producti</a>...</td>
<td><a href="http://schnappii-pr">http://schnappii-pr</a>...</td>
<td><a href="http://sn">http://sn</a>...</td>
</tr>
<tr>
<td>on.s3.amazonaws.com/APP...</td>
<td>on.s3.amazonaws.com/APP...</td>
<td>on.s3.amazonaws.com/APP...</td>
<td>on.s3.amazonaws.com/APP...</td>
</tr>
<tr>
<td>_STORAG...</td>
<td>_STORAG...</td>
<td>_STORAG...</td>
<td>_STORAG...</td>
</tr>
<tr>
<td>E/19681/B4478AC0-475A-4DEB-9E1B-FA8655D33A78/Us...</td>
<td>E/19681/B4478AC0-475A-4DEB-9E1B-FA8655D33A78/Us...</td>
<td>E/19681/B4478AC0-475A-4DEB-9E1B-FA8655D33A78/Us...</td>
<td>E/19681/B4478AC0-475A-4DEB-9E1B-FA8655D33A78/Us...</td>
</tr>
<tr>
<td>80592fe5-5559-4211-8eee-00b8e22f7b52.png</td>
<td>80592fe5-5559-4211-8eee-00b8e22f7b52.png</td>
<td>80592fe5-5559-4211-8eee-00b8e22f7b52.png</td>
<td>80592fe5-5559-4211-8eee-00b8e22f7b52.png</td>
</tr>
</tbody>
</table>

Fixed

12. If necessary, check off the box next to *Submission right after a code is taken*. For instance, you might want to add the QR Code field at the end of a form and enable
instant submission after all the required fields have been filled out and a QR code has been scanned.

For this option to work properly, make sure you have specified the correct email address for the submission of PDF/Excel reports in the form settings.

Please keep in mind that this option is currently available only for iOS devices.

13. When the *Submission right after a code is taken* box is selected, a new checkbox that says *Allow multiple scanning* appears. (This option is also supported by iOS devices only at this point.)

This feature allows you to scan several QR codes before the automatic submission of the form.

**Using Text Formula with the QR Code Field**

To see the details about the scanned QR Codes, add the Text Formula field next to the QR Code field.
In the popped up window, enter the name of the QR Code field from the QR Code field settings in the following format <name>. 
8.15 Radio Button

The Radio button enables your app users to select a single option from the list of options and integrate it into PDF/Excel/Word reports to share it with others or store it for personal use.

To enable this functionality, follow the steps below:

1. Drag and drop the Radio button onto the canvas.

2. In the popped up window, specify its value and group name.
Value is what this specific radio button means.

Group name is the group this specific radio button belongs to.

For instance, in this app, we have created two groups:

1. How am I feeling?

2. How was my day?

The first group contains five radio buttons, whereas the second – three.

The value of the 1st radio button from the 1st group is Great, the 2nd – Good, etc.
In order to add a group name, click on *Manage* and then *Add*. 
• Enter your group name. (Group names can have any name.)
• Choose if the data goes as text or as number. In our case, it is text.
• Next to Field Id, from the drop down menu, select the Column ID of this field in the data source where the newly added answers will be sent. In our case, it is feel. Our Radio buttons are located on the Form button, and next to Select Data Source in the settings of the Form button, we have indicated that the information from the fields located on the Form will be added to the data source called circle.

In the Excel file called circle, used as our data source, we use the column feel to store uploaded answers from the Radio button.

Here is how our data source looks in the data source dialog.

• Check off the box next to Required if you need to make this element mandatory.
• Check off the box next to Include Field in Report if the Radio button needs to be added in the PDF/Excel/Word reports upon the completion and submission of each form.
• Click *Add New Group* and then *OK*.

3. By default, the negative (unchecked) image is displayed. Select *Checked* if you need the positive image to be displayed. (In our case, it doesn’t make sense to select *Checked*. However, if you go with the Yes/No answers only, for example, you might want the Yes answer to be always displayed unless your users change the answer.)

   **Positive Image Selected by Default**  **Negative Image Selected by Default**

   ![Image](image-url)

   In our case, it doesn’t really make sense to select *Checked*. However, if you go with the Yes/No answers only, for example, you might want the Yes answer to be always displayed unless your users change the answer.

   4. Specify the dimensions of the Radio button.

   5. Upload your own positive and negative images for the Radio button if you wish.

   In another app, we have uploaded a green icon for the positive image (if the answer is *Yes*), and a grey icon for the negative/default/unchecked image.
For the second radio button that stands for the *No* option, we have a red icon for the positive image, and a grey icon for the negative/default/unchecked image. In other words. Here is how this variant works on a mobile device.
Let us go back to our original app now. After we added the first radio button with the value = Great, we added 4 more with identical settings with the exception of the value: Good, OK, Bad, and Very bad accordingly. (Please keep in mind that all these options belong to the same group – How am I feeling?.)

Next, we configured the other 3 radio buttons (Great, OK, Bad) for the group called How was My Day? and configured the buttons and the group similarly.
Here is how these buttons work on a mobile device.

8.16 Reset Button
The Reset button enables your users to revert to the default screen. The same functionality is available in the Configure your element section of the Form button.

On the screenshots above, you can see that by default we have the answers next to Great selected. After we have filled out any number of the fields, we can revert to the default settings by simply clicking on Reset. Keep in mind that clicking on Clear would clear all the fields, including the radio buttons next to Great.

However, the Reset button under Fields and Buttons gives you more flexibility since you can place it wherever you would like and configure it the way you need: you can use the global style, use the default icon, upload a new icon, or opt for a custom style.
Another example of utilizing the Reset button would be reverting to the default options when the date, time, and location gets automatically identified the moment one opens the Form.

To configure the Reset button, all you need to do is to drag and drop it onto the canvas and, in the popped up window, specify the label, if any, icon, dimensions, font, and style of the Reset button.
8.17 Sign/Draw Field
Like the Sign/Draw tab and button, this field allows users to create and share digital signatures or draw pictures, as well as send them to their co-workers or clients. It also enables users to choose brush colors and sizes. This field can be added to a number of tabs and elements, including the Form tab/button and Table field. Moreover, unlike the Sign/Draw tab and button, signatures and drawings completed in this field can be included together with the rest of a report and then stored on the device or shared with others.
To enable this functionality, follow the steps below:

1. Drag and drop the Sign/Draw field onto the desired tab/element.
2. In the popped up window, specify its name and the web parameter name.
3. Check off the box next to *Required* if you need to make this element mandatory.
Here's how the element looks when it's required.
4. Check off the box next to Read only to disable any accidental copying or deletion of this field in the Express Mode.

On the screenshots above, you can see that when Read only is enabled, the Sign/Draw field cannot be copied or deleted, whereas when it’s disabled, you see the copy and delete symbols.
5. Check off the box next to *Show full screen view* to enable the full screen opening of the Sign/Draw field.

*Show full screen view enabled*  
*Show full screen view disabled*

![Sign/Draw field with full screen view enabled](image1.png)  
![Sign/Draw field with full screen view disabled](image2.png)

Please keep in mind that the *Full screen view* option is not available on some mobile devices.

6. Check off the box next to *Include Field in Report* if it needs to be added in the PDF/Excel reports upon the completion and submission of each form.

7. Specify its dimensions, order, and next order of the Sign/Draw field.

8. Next to Select Image Field, select the Column ID of this field in the data source where newly added sketches and signatures will be sent.
9. Check off the box next to *Sharing enabled* if you would like to allow your users to share their signatures and sketches separately, not only within PDF/Excel reports.
This option adds the Share button on the top and allows emailing, printing, adding sketches and signatures to Dropbox, etc.
It also allows sharing the Sign/Draw field via the Share button added at a specified location.

10. Check off the boxes next to Use URL instead of image if necessary.

11. Specify the scale of the image and column width.

There is also a video tutorial available for the Sign/Draw field. Click here to watch it.
The Submit button enables your users to submit a filled out form to a specified email, web post, or data source. These submission options are available in the settings of the Form button on which the Submit button is placed.

The same functionality is available in the *Configure your element section* of the Form button.
However, the Submit button under *Fields and Buttons* gives you more flexibility since you can place it wherever you would like and configure it the way you need: you can use the global style, use the default icon, upload a new icon, or opt for a custom style.
8.19 Video Field

The Video field allows users to record, play back, integrate into PDF and Excel reports, and share with others video files in the MP4 format.

To enable these capabilities, follow the steps below:

1. Drag and drop the Video field onto the canvas.

2. In the popped up window, specify its name and the web parameter name.

3. Enter the URL and thumbnail field names the way you would like them to appear in PDF/Excel reports.
4. Check off the box next to *Required* if you need to make this element mandatory.

Here’s how the Video field looks when it’s required and when it’s optional.

<table>
<thead>
<tr>
<th>Required</th>
<th>Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video Url: <a href="http://snappi-production.s3.amazonaws.com/APP_STORAGE/1257609/2E9309BB-078A-4D5A-B4F8-05E391E7BAF4/Video/1e8ddcd-3a3a-e30a-5bcf-578c5f8f6c9c/982f67e6-d17c-4b4f-980c-f3c35d7d6acb.mp4">http://snappi-production.s3.amazonaws.com/APP_STORAGE/1257609/2E9309BB-078A-4D5A-B4F8-05E391E7BAF4/Video/1e8ddcd-3a3a-e30a-5bcf-578c5f8f6c9c/982f67e6-d17c-4b4f-980c-f3c35d7d6acb.mp4</a></td>
<td>Video Thumbnail:</td>
</tr>
</tbody>
</table>

Name: Painting 1
5. Check off the box next to *Read only* to disable any accidental copying or deletion of this field in the Express Mode.

*Read only enabled*  
*Read only disabled*
On the screenshots above, you can see that when *Read only* is enabled, the Video field cannot be copied or deleted, whereas when it’s disabled, you see the copy and delete symbols.

6. Check off the box next to *Include Field in Report* if the Video field needs to be added in the PDF/Excel reports upon the completion and submission of each form. (In reports, it will be displayed as the link to the recorded MP4 video file.)

7. Specify its dimensions, order, and next order.

8. Check off the box next to *Sharing enabled* if you would like to allow your app users to share their video files separately with the help of the *Share button*, not only within PDF/Excel reports.
9. Select the audio quality: from high to low.
10. Specify the maximum video duration.

11. Next to Select Video Field, from the drop down menu, select the Column ID of this field in the data source where the newly added links to recorded video files will be sent.

In our case, the Video field is located on the Table field, and next to Select Data Source, we have indicated that the information from the fields located on the Table field will be added to the data source called Video.

In the Excel file called Video, used as our data source, we use the column video to store generated links to our recorded video files.

Here is how our data source looks in the data source dialog.
Here is how the Audio field works on a mobile device.

12. Select one of the three column width options for Excel reports.
### Auto Fit

*My video:* [video link]  
[video link]

### Wrap Text

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>My videos:</td>
<td>Video Thumbnail</td>
<td>My videos:</td>
<td>Name</td>
</tr>
<tr>
<td>s1e.sm</td>
<td>s1e.sm</td>
<td>89AF1</td>
<td>89AF1</td>
</tr>
<tr>
<td>89AF1</td>
<td>89AF1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Video Thumbnail</td>
<td>Video Thumbnail</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Fixed

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>My Bar</td>
<td>My video:</td>
<td>My video:</td>
<td>My videos: Name</td>
</tr>
<tr>
<td><a href="http://sn">http://sn</a></td>
<td><a href="http://sn">http://sn</a></td>
<td>Painting 1</td>
<td></td>
</tr>
</tbody>
</table>
The Table button allows adding multiple entries, such as notes, photos, locations, sketches, QR and Barcodes, audios, and videos, and storing and sharing uploaded information with others in PDF/Excel/Word formats.
To enable these capabilities, follow the steps below.

1. Drag and drop the Table button onto the canvas.

2. In the popped up window, specify its name, title, and the web parameter name.

3. Check off the box next to **Required** if you need to make this element mandatory.
In the screenshot above, you can see that we have made *My Notes* table required, while *My Photos, My Locations,* and *My Drawings* tables are optional. (They do not have to be filled out to complete the form.)

4. Specify the dimensions of the Table button, as well as its order and next order.

5. Next to List View Template, click on *Change* and select one of the two options: Custom View or Standard View. For the Custom View, specify the cell height.

![Standard List Cell View](image)
As you can see from the screenshots below, choosing the Standard List Cell View does not allow editing it in the editor as the Cell View button is inactive. Here is how the Standard List Cell View looks on a mobile device.
As you can see from the screenshots above, choosing the Custom Cell View does allow editing it in the editor. Thus, for instance, you can specify the size of the cell and add the Image button and a few Text elements anywhere you would like within the cell.

Let’s look at the Table button titled *My Photos*, whose screenshots you can see above.

We added two Text elements: *Description* and *Note*. 
For the *Description* Text element, we entered *Description* next to *Text*, specified its font, alignment, font, size, and font color. Most importantly, next to *Select Input*, we chose *Don’t use* since this Text element will always display the same information - the word *Description*.

For the *Note* Text element, we entered *Note* next to *Text*, specified its font, alignment, font, and font color. We also made this element long enough to make sure that entered text won’t get cut...
off. Most importantly, next to Select Input, we chose Note since this Text element will send its information to the data source connected to this Table button (the data source called Photos) and, more specifically, to the column called Notes.

In the settings Submit to Data Source of the Form on which this Table button is located, we can see that the Notes column corresponds to the Note field.

In the Excel file called Photos, used as our data source, you can see that we utilize the column notes to store our users’ notes.

Here is how our data source looks in the data source dialog.
For the Image button, we specified its dimensions and, next to Select Input, we chose Photo since this Image button will send its information to the data source connected to this Table button (the data source called Photos) and, more specifically, to the column called Image.
In the settings *Submit to Data Source* of the Form on which this Table button is located, we can see that the Image column corresponds to the Photo field.

Here is how our data source looks in the data source dialog.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>notes</td>
<td>image</td>
<td>photosid</td>
<td>EmailPer</td>
</tr>
<tr>
<td>2</td>
<td>Castle 1</td>
<td><a href="http://sn">http://sn</a></td>
<td>8bd598e3:</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Castle 2</td>
<td><a href="http://sn">http://sn</a></td>
<td>8bd598e3:</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Castle 3</td>
<td><a href="http://sn">http://sn</a></td>
<td>8bd598e3:</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Now that your List Cell View, be it a standard or custom one, is ready, open the Detail View and add to it whatever fields and buttons you need.
We added two Text elements, one Photo field, and one Text/Number field.
For both of the Text elements, we specified the text that needs to be displayed by them, their font, font color and size, alignment, and dimensions.
We configured the Photo field accordingly. Read more on working with the Photo field here.
Similarly, you can read about configuring the Text/Number field [here](#).

6. Connect this Table button to the data source you have previously uploaded in the data source dialog so that the information uploaded through the fields of this Table button gets saved in this data source.

   In our case, it is *Photos*.

7. Check off the boxes next to *Show Reset button* and *Show Clear button* if necessary.
8. Check off the box next to *Read only* to disable any accidental copying or deletion of this field in the Express Mode.

**8.21 Multiline Entry Button**

Like the Table button, the Multiline Entry button allows adding multiple entries, such as notes, photos, locations, sketches, QR and Barcodes, audios, and videos, and storing and sharing uploaded information with others in PDF/Excel/Word formats. However, unlike the Table button, you cannot change the dimensions of the Multiline Entry button; neither can you customize the cell view of the Multiline Entry button.
Nevertheless, the Multiple Entry button asks users if they are sure they want to delete an item, while the Table button doesn’t.

To enable the capabilities of the Multiple Entry button, follow the steps below.
1. Drag and drop the Multiline Entry button onto the canvas.

2. In the popped up window, specify its name, title, and the web parameter name.

3. Check off the box next to Required if you need to make this element mandatory.

4. Specify its order and next order.

5. Under Select Data Source, specify the data source where uploaded information (in our case, photos and their descriptions) will be added.

6. Specify whether the Reset and Clear buttons will be displayed on the navigation bar.

7. Enter your placeholder. Click OK.

8. Now that you have configured the settings of the Multiple Entry button, open its Detail View and add to it whatever fields and buttons you need.
We added two Text elements, one Photo field, and one Text/Number field.
For both of the Text elements, we specified the text that needs to be displayed by them, their font, font color and size, alignment, and dimensions.
We configured the Photo field accordingly. Read more on working with the Photo field [here](#).
Similarly, you can read about configuring the Text/Number field [here](#).

### 8.22 Table Selection Button

The Table Selection button allows selecting information from a number of previously saved in the data source options (the Single and Multiple options) or saving previously entered information and reusing it by auto-filling the fields in a new form with it (the Fill Inputs option).

#### Single and Multiple Options

For example, a restaurant owner might want to create an app enabling its waiters to take orders on their mobile devices by choosing dishes from the list of menu options. In this case, he/she will need to add the Table Selection button, enter all the menu options in the data source connected to the button, and then choose the Multiple option to enable the waiters to choose several dishes for each order.
In the screenshots above, you can see two Form buttons: one for adding items and their prices to the menu and one (with the Table Selection button in it) for taking orders.

After an item has been added, it goes to the data source this Form button is linked to.
When a waiter is ready to take an order, he/she opens the Order button, enters the table number, his/her name, and selects any number of menu items by clicking on the Table Selection button. (In our app, it is called *Select Dishes*.)

Here is how the Table Selection button was configured in this case.
In the settings of the Table Selection button, you can also see *Table Field 246*, which is the Table button we have added in order to display the items chosen through the Table Selection button.

The Table Selection button with the Single Option is configured very similarly. However, in this case, the user can choose only one item at a time:

---

**Fill Inputs Option**
For example, often, while filling out reports, we have to repeatedly enter the same information, such as names, addresses, email addresses, etc. With the Table Selection button, your app users won’t have to do so anymore. Once they have entered this information and saved the form, this information gets saved in a specified data source. Next time, while filling out a new form within your app, they can simply click on the Table Selection button and choose the name of the previously saved report to instantly auto-populate the required fields with this information.

To enable this functionality, please follow the steps below:

1. Drag and drop the Table Selection button onto any form you need.

2. In the popped up window, specify its label (if any), title, and dimensions.

3. Configure the style of your button: Global Style/Icon/Global Style.
4. Specify, if the information will be displayed by clicking on the button or in the framed view.

<table>
<thead>
<tr>
<th>Type</th>
<th>Button</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checkbox position</td>
<td>Frame</td>
</tr>
</tbody>
</table>

**Button**

**Frame**

5. Select the checkbox position: right/left.
6. Next to *List View template*, click on *change* to configure how the saved information will be displayed. If you opt for the Custom view, you will also need to specify the height of the cell.

Click OK.
7. Specify the message that will appear if no saved information has been found.

   Empty dataset message: No saved projects have been found.
   Show Refresh button

8. Put a check mark next to *Show Refresh button* if you wish.

9. Specify the font used for the button.

   Font: Arial
   The quick brown fox jumps over the la...

10. The *Show all items* option works only for the Table Selection button configured as Frame and allows displaying all saved reports at once, while by default, only 25 cells with saved information get displayed and you need to click on *Show more* to see the rest. (This option is currently available for iOS devices only; Android devices always show the whole list whether it is selected or not).

   - Show all items
   - Select all items by default

11. Put a check mark next to *Select all items by default* if necessary. (This option is currently available for Android devices only).
12. Under **Selection Options**, choose the data source where filled out information will be chosen from and stored.

Under **Input Mode**, select **Fill Inputs** if you would like previously saved information to auto-populate the fields you need.
13. Under Data Mapping, match the fields in the data source to the ones in the Form. You don’t have to match all of them – add the matching pairs only for the fields that you would like to be auto-populated.
14. Click OK and then *Edit View* to configure the way saved reports will be displayed by the Table Selection button.

15. Here you need to enter the fields you want your users to be able to see in the list view. We added one text field to display project names and one to display dates.
The quick brown fox jumps over the la..
We chose *Data Source* next to *Source* for both elements so that the information for this list can be taken from the data source (*New Daily Report Form*) and also chose the corresponding fields from which project names and dates will be taken.

When you have added and configured the necessary elements, click OK.

Here you can see how these two fields look in the editor and on a mobile device.
8.23 Text Formula/Text Calculated Field

The Text Calculated field works with data of the text type.

Here are some of the most common examples of the Text Formula usage.
1) Combining information from several Text/Number fields into one with the formula 
\(<\text{field1}> + <\text{field2}>\)

To illustrate the use of this formula, let us make a mobile form with 4 fields: 2 Text/Number fields (for the name of the product and price), one Datetime field (for the date), and one Text Calculated field (for the combination of the product, price, and date). Here is how we configured each of the fields.

In the Text Calculated field, next to formula, we entered: \(<\text{Product Name}> + " + \text{"} + \text{<Price>} + " + \text{"} + \text{<Date>}\)
IN_TEXT(<Price>)+ "  " + GET_DATE(<Date>)

Here is how this formula works on a mobile device:

As you can see from the screenshots above, the first three fields are of different types: text, number, and date. As we know, the Text Formula works only with the text type. So to transfer numerical data into the text format, we use the formula: IN_TEXT(<number field>). In our example, we converted the numeric price field into text by the formula IN_TEXT(<Price>). For the datetime fields, we use GET_DATE(<datetime field>) for dates, and GET_TIME(<datetime field>) for times. In our case, we converted the Datetime field with the data into text by the formula GET_DATE(<Date>)

Please keep in mind that whatever is given in the quotation marks" ", will be shown exactly as it is written in the quotation marks, including spaces. For instance, if we enter <name>+IN_TEXT(<price>), we will see both the name and the price together without any space. Last but not least, all field names should be entered into formulas exactly the way they are. If it is date, in the formula it should say <date>. If it is Date, in the formula it should be <Date>. Finally, if it is Date and Time, it should say <Date and Time>.

2) Converting data of the numeric type into the currency format with the help of
TO_CURRENCY(<Field>)

To illustrate how this formula can be used, let us add two fields to a mobile form: 1 Text/Number field and one Text Calculated field.

Here is how we configured them:
As you can see, in the Text Calculated Field, next to formula, we entered:

TO_CURRENCY(<Price>) + " $".

3) Retrieving a year, month, a date, an hour, and minute from the Datetime field with the help of the following formulas:

<FieldName>.year is used to retrieve a year.

<FieldName>.month is used to retrieve a month.

<FieldName>.day is used to retrieve a day.

<FieldName>.hour is used to retrieve an hour.

<FieldName>.minute is used to retrieve a minute.

<FieldName>.dayOfWeek is used to retrieve a day of the week (1 – Monday, 2 – Tuesday, 3 – Wednesday, 4 – Thursday, 5 – Friday, 6 – Saturday, and 7 – Sunday).

To illustrate these formulas, let us add one Datetime field and six Text Calculated fields to a mobile form.
Here is how we have configured each of these fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Current Date and Time</td>
</tr>
<tr>
<td>Format</td>
<td>Subtitle</td>
</tr>
<tr>
<td>Width</td>
<td>270</td>
</tr>
<tr>
<td>Web Parameter name</td>
<td>DateTimeField546</td>
</tr>
<tr>
<td>Order</td>
<td>1</td>
</tr>
<tr>
<td>Next order</td>
<td>1</td>
</tr>
<tr>
<td>Font</td>
<td>Default</td>
</tr>
<tr>
<td>Placeholder</td>
<td>Current Date and Time</td>
</tr>
<tr>
<td>Column width (for Excel report)</td>
<td>Auto fit</td>
</tr>
<tr>
<td>Name</td>
<td>Year</td>
</tr>
<tr>
<td>Web Parameter name</td>
<td>TextCalculateFields47</td>
</tr>
<tr>
<td>Formula</td>
<td>Current Date and Time-year</td>
</tr>
<tr>
<td>Width</td>
<td>170</td>
</tr>
<tr>
<td>Height</td>
<td>35</td>
</tr>
<tr>
<td>Order</td>
<td>2</td>
</tr>
<tr>
<td>Placeholder</td>
<td>Year</td>
</tr>
<tr>
<td>Font size</td>
<td>12pt</td>
</tr>
<tr>
<td>Alignment</td>
<td>Left</td>
</tr>
<tr>
<td>Font color</td>
<td>Bind with global theme</td>
</tr>
<tr>
<td>Visible</td>
<td>Yes</td>
</tr>
<tr>
<td>Frame shown</td>
<td>Yes</td>
</tr>
</tbody>
</table>

1091
<table>
<thead>
<tr>
<th>View</th>
<th>View</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>Test Calculated Field147</td>
<td>Test Calculated Field147</td>
</tr>
<tr>
<td><strong>Formula</strong></td>
<td><strong>Formula</strong></td>
</tr>
<tr>
<td><code>&lt;Current Date and Time&gt;.month</code></td>
<td><code>&lt;Current Date and Time&gt;.day</code></td>
</tr>
<tr>
<td><strong>Width</strong></td>
<td><strong>Width</strong></td>
</tr>
<tr>
<td>270</td>
<td>121</td>
</tr>
<tr>
<td><strong>Height</strong></td>
<td><strong>Height</strong></td>
</tr>
<tr>
<td>75</td>
<td>135</td>
</tr>
<tr>
<td><strong>Order</strong></td>
<td><strong>Order</strong></td>
</tr>
<tr>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td><strong>Placeholder</strong></td>
<td><strong>Placeholder</strong></td>
</tr>
<tr>
<td>Month</td>
<td>Day</td>
</tr>
<tr>
<td><strong>Font Size</strong></td>
<td><strong>Font Size</strong></td>
</tr>
<tr>
<td>12px</td>
<td>12px</td>
</tr>
<tr>
<td><strong>Alignment</strong></td>
<td><strong>Alignment</strong></td>
</tr>
<tr>
<td>Left</td>
<td>Left</td>
</tr>
<tr>
<td><strong>Font Color</strong></td>
<td><strong>Font Color</strong></td>
</tr>
<tr>
<td>Red</td>
<td>Red</td>
</tr>
<tr>
<td><strong>Font</strong></td>
<td><strong>Font</strong></td>
</tr>
<tr>
<td>Default</td>
<td>Default</td>
</tr>
<tr>
<td><strong>Sharing Enabled</strong></td>
<td><strong>Sharing Enabled</strong></td>
</tr>
<tr>
<td>Visible</td>
<td>Visible</td>
</tr>
<tr>
<td><strong>Frame Shown</strong></td>
<td><strong>Frame Shown</strong></td>
</tr>
<tr>
<td>Visible</td>
<td>Visible</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>View</th>
<th>View</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>Test Calculated Field147</td>
<td>Test Calculated Field147</td>
</tr>
<tr>
<td><strong>Formula</strong></td>
<td><strong>Formula</strong></td>
</tr>
<tr>
<td><code>&lt;Current Date and Time&gt;.hour</code></td>
<td><code>&lt;Current Date and Time&gt;.minute</code></td>
</tr>
<tr>
<td><strong>Width</strong></td>
<td><strong>Width</strong></td>
</tr>
<tr>
<td>270</td>
<td>121</td>
</tr>
<tr>
<td><strong>Height</strong></td>
<td><strong>Height</strong></td>
</tr>
<tr>
<td>75</td>
<td>135</td>
</tr>
<tr>
<td><strong>Order</strong></td>
<td><strong>Order</strong></td>
</tr>
<tr>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td><strong>Placeholder</strong></td>
<td><strong>Placeholder</strong></td>
</tr>
<tr>
<td>Hour</td>
<td>Minute</td>
</tr>
<tr>
<td><strong>Font Size</strong></td>
<td><strong>Font Size</strong></td>
</tr>
<tr>
<td>12px</td>
<td>12px</td>
</tr>
<tr>
<td><strong>Alignment</strong></td>
<td><strong>Alignment</strong></td>
</tr>
<tr>
<td>Left</td>
<td>Left</td>
</tr>
<tr>
<td><strong>Font Color</strong></td>
<td><strong>Font Color</strong></td>
</tr>
<tr>
<td>Red</td>
<td>Red</td>
</tr>
<tr>
<td><strong>Font</strong></td>
<td><strong>Font</strong></td>
</tr>
<tr>
<td>Default</td>
<td>Default</td>
</tr>
<tr>
<td><strong>Sharing Enabled</strong></td>
<td><strong>Sharing Enabled</strong></td>
</tr>
<tr>
<td>Visible</td>
<td>Visible</td>
</tr>
<tr>
<td><strong>Frame Shown</strong></td>
<td><strong>Frame Shown</strong></td>
</tr>
<tr>
<td>Visible</td>
<td>Visible</td>
</tr>
</tbody>
</table>
Here is how these formulas work on a mobile device.
4) Replacing some letters with new ones with the help of the formula

`(<fieldname>).replace("searchvalue","newvalue")`

This is a very useful formula since it replaces all old values with new ones at once. Thus, for instance, if you have a long text and need to replace the old name of your company with a new one, this formula can be very convenient.

Let us add one Text/Number field and one Text Calculated field to show how this formula works on an example.

Here is how we configured both of the fields.

So we used the following formula: `(<Previous>).replace("Hillary Clinton","John Kerry")` to replace *Hillary Clinton* with *John Kerry* every time her name is mentioned.

Here is how this formula works on a mobile device.
5) Turning lower case letters to upper case letters and vice versa with

(<fieldName>).toUpperCase() и (<fieldName>).toLowerCase().

This time, we also added one Text/Number field and one Text Calculated field but changed the Text Formula to (<Previous>).toUpperCase() Here is what we got as a result:

6) Extracting symbols from a line with the help of the formula
(<fieldname>.substring (from, to) from
shows where to start the extraction.

to shows where to stop the extraction. If omitted, it extracts the rest of the string
To illustrate this formula, we will keep the two previously added fields but change the formula to
(<Previous>.substring (23, 54) to extract the Declaration of Independence from the following
sentence: “Thomas Jefferson wrote the Declaration of Independence in 1776.”
Here is how we have configured these two fields:

Here is how this formula works on a mobile device.
As we can see, the Declaration of Independence falls in the range between 23 and 54.

7) Returning the position of the first repetition of a specified value in a string with the help of the formula `(fieldName).indexOf(searchvalue,start)`

`Searchvalue` indicates the string to search for; `start` indicates at what point to start the search. The default position is 0.

To illustrate this formula, we will keep the two previously added fields but change the formula to `(Previous).indexOf("e",5)` for the following sentence: “Hello, World. Welcome to the Universe.”

Here we can see that the first repetition of “e” happens under the 15th character.
8) Retrieving specific information with the following formulas:

`<user.userName>` retrieves the user’s name entered upon the registration and shows the name of the user when he/she is logged in.

`<user.emailAddress>` retrieves the email address under which the user is logged in.

`<user.memberId>` retrieves the member ID if any.

`<user.userType>` retrieves the user’s type.

`<user.location>` retrieves the latitude and longitude of the user.

`<user.location.latitude>` retrieves the latitude of the user.

`<user.location.longitude>` retrieves the longitude of the user.

`<user.location.textAddress>` retrieves the full address of the user in the text format.

`<user.location.textAddress.city>` retrieves the city of the user.

`<user.location.textAddress.region>` retrieves the region of the user.

`<user.location.textAddress.country>` retrieves the country of the user.

`<user.location.textAddress.zip>` retrieves the zip code of the user.
<user.company> retrieves the user’s company.

<user.companyId> retrieves the user’s company ID.

<device.version> retrieves the version of the user's mobile device.

<device.model> retrieves the model of the user’s mobile device.

<device.language> retrieves the language chosen on the user’s mobile device.

<device.id> retrieves the ID of the user’s mobile device.

<device.platform> retrieves the platform of the user’s device.

Here is how these formulas work on a mobile device.
These formulas are very widely used all together or separately. For example, you might want to show to or hide from some users some information based on the users’ email addresses, locations, devices, etc.

Here are a few examples of how these formulas are unitized in our apps.

**Inspect Anything**

In this app, two Text Calculated fields allow displaying the value of captured barcodes and QR Codes, saving, and sharing them with others, if necessary.
Here is how these two fields are configured.
In the same form, another Text Calculated field is used.
However, unlike the other two, it is invisible and serves a different purpose. This field captures the email address of the user filling out this form and saves it so that, when retrieving the previously saved reports through the List button, our user can see only his/her reports (the reports submitted under his/her email address). Here is the formula we use to enable this function in the settings of the List button.

(In this app, the email addresses our users are logged in are saved under email2.)

In fact, we use the Text Calculated field with <user.emailAddress> on every single form, and, in the Search Settings section of every List button, we specify that the report must contain our user’s emailAddress for this report to be visible to the given user.
My Income Calculator

The My Income Calculator app allows our users to calculate their weekly income by entering the days they work, the hours, and their pay rate. This app utilizes a variety of formulas, some of which we discuss in other sections of this manual. Here we will look at one Text Calculated field.

This field takes advantage of the formula `TO_CURRENCY(<DS>) + " $"` to convert the number calculated by the `DS` field (the DS Calculated field) to currency and add the $ sign.
Here is how the final result looks on a mobile device. (See the value next to Sum.)
8.24 Text/Number Field

The Text/Number field allows users to enter textual information, numbers, email addresses, passwords, currency, web sites, and phone numbers.

For Text/Number fields to work properly, the fields connected to them in the Data Source Dialog should be configured as Text.

To fully configure your Text/Number field, please follow the steps below:
1. Drag and drop the Text/Number field onto any form you need.

2. In the popped up window, specify the name of the field and its type, depending on what type of information it will be used for.

   ![Diagram showing field properties](image)

   3. Put a check mark next to **Multiline** to enter text into multiple lines.

   ![Multiline Enabled](image)  
   ![Multiline Disabled](image)
These options are a matter of convenience. Regardless of which option you choose, in PDF reports, entered information will look the same: in one line.

4. Indicate whether this field will be required. If you choose to make it required, your app users won’t be able to submit the form on which this field is located without filling it out.
Here’s how the Text/Number field looks when it’s required and when it’s optional. (The Street Address Text/Number field is used to illustrate these options.)

<table>
<thead>
<tr>
<th>Required</th>
<th>Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Required Image]</td>
<td>![Optional Image]</td>
</tr>
</tbody>
</table>

5. Specify the dimensions of the field and enter the order in which this field will appear in the report. (The *next order* can be left as is.)

6. Select a new font for this field or leave the default one.

7. Next to *Select Field*, from the drop down menu, select the Column ID of this field in the data source where the newly textual information will be added.

Here is why we selected this field:

A. Our Text/Number field is located on the Form button connected to the data source called *Estate Sales.*
B. Here is how our data source looks in the data source dialog.

8. Specify its placeholder if you wish.

The Datetime Field with and without a Placeholder
9. Select one of the three column width options for the Excel reports.

10. Put a check mark next to Save input history to be able to auto-populate this field with any of the previously entered information. If you enable saving input history, after your app users have submitted their information for the first time, they will see a triangle in the right corner of the field and will be able to auto-populate this field with any previously saved information by simply clicking on this field and selecting the entry they need.
11. Check off the box next to *Include Field in Report* if the Text/Number field needs to be added in the PDF/Excel reports upon the completion and submission of each form.

12. Check off the box next to *Sharing enabled* if you would like to allow your app users to share this field separately with the help of the Share button, not only within PDF/Excel reports.
Chapter 9. Widgets
9.1 Creating and Adding Widgets

To demonstrate how to create a widget, let us create a widget for the Login Tab.

1. Please see *Chapter 5 Tabs and Buttons/the Login Tab* to build Login and Create Account screens.

2. When your Login Tab is ready, select this tab and click *Copy*.

3. On the pop up screen, select one of the five options available. Click *OK*.
4. On the next pop up screen, enter the name of your widget and its description. Click OK.

Name for the widget:
Login

Description for the widget:
Login tab and Sign up Form inside

Data Source:
Do not copy
Tags:

The widget is ready! You will be able to use it by simply dragging it onto the tab bar below the canvas, delete it by clicking on the cross sign, or read its description by clicking on the question mark.
Chapter 10. App Testing and Submission

10.1 Snappii’s Preview App

Snappii’s Preview app allows viewing both demo and live apps built on our platform in the App Store and Google Play. You can download Snappii’s Preview app from the AppStore or Google Play for free at any time. Regardless of where you are in the app development process, to view your apps, you can log in under your Snappii’s account credentials, which you will get upon your free registration when you start using the editor.
When you have modified your app,

A. On your iOS device - tap twice at the top of the screen and press "Refresh" to see the changes to the app open in the Preview app or tap “Back to Snappii” and then "View" in the app list if you have modified an app different from the one currently open.
B. On your Android device – press on the Menu button and then “Refresh” to see the changes to the app open in the Preview app or press on the Back button twice and select the app you need if you have modified an app different from the one currently open.
10.2 Web2phone Copy
A web2phone copy of your unsubmitted or submitted application allows viewing it without the Preview app. You will see exactly what your app will look like once available in the Google Play Store. It will show the app icon, the splash screen and remove Snappii’s “Powered By”, if you have white labeled the application.

Please note that you need to choose a subscription plan prior to requesting a web2phone copy.

Since an application submitted into the Google Play Store becomes available within 24 hours, there is rarely a need to create a web2phone copy for an Android application. However, make sure that you create a web2phone copy before submitting the app because after the submission this option won’t be available.

10.2.1 Building a Web2Phone Copy for the Android Platform
1. Go to snappii.com and hit “My apps
2. Choose the application for which you need a web2phone copy and click “Web-2phone”. You will get an email with the installation link in about 10-15 minutes. (Sometimes it may take more time depending on the number of test builds in line)
3. Once you have received the installation link, click on it on the device you are planning to use this web2phone copy.

10.2.2 Building a Web2Phone Copy for the iOS Platform

1. Join the Apple Developer program [here](#).
2. Complete the Apple Developer profile [here](#) and pay the membership fee.
3. Add at least one device ID to your Apple Developer profile [here](#). If you need to be able to install the web2phone copy on any number of additional mobile devices, make sure to add all of them to your profile before requesting this web2phone copy.

You will need the UDID (Unique Device Identifier) of your mobile device(s). In order to find it, attach your iPhone or iPad to your computer with the USB cable. The following screen will pop up.

Click on “For You” tab. Then hit “Serial Number”
The UDID of your device will be displayed. Now you’ll need to copy it with the right click.

To learn about an alternative way of finding the UDID of your device, please click here.
4. Go back to snappii.com and click on “My Apps”

5. Choose the application for which you need a web2phone copy and click “web2phone copy”

You will be emailed the installation link to view the app on your device in about 10-20 minutes. Once you receive the link, make sure to open it on the device for which you need the web2phone copy.
!!! Please keep in mind that if you decide to add another device after you have made this web2phone copy, you won’t be able to install the web2phone copy on the new mobile device. You would need to build a new web2phone copy. Also, after saving any changes to your app within the “app settings” portion of the Snappii platform, you will need to request a new web2phone copy. When the new link is received, delete the old Web2phone copy from your device, install the new copy, and your changes will appear.

### 10.3 HTML5 Emulator

HTML5 Emulator allows you to work with your app in the browser. App user can submit forms in the browser. Working with the app in the Snappii Editor, the user can instantly see the changes via the browser publishing the HTML5 web app.

HTML5 preview can be turned on for both: Express and Guru modes. Just select the viewing mode:

Let’s consider an example and create a simple app to add different products and view them via the list. Here’s how the app looks in the Editor:
Now let’s switch to HTML5.
The app will be loading for a few seconds, plus you will see that the Editor background has darkened, which is done on purpose in order to keep you focused on the app. Now you see that the app appearance inside the device has changed so you can start working with it. You see 2 buttons: add new product and Product list. Let’s add a new product:
After submitting the data you will see the following message:
The new product has been successfully added, you can check this entry in the product list:

As you can see, the product has been added which means that it was also added to the datasource. In the list settings (check Advanced List settings here) → Security you can set up permissions to Add/Delete/Update data based on user types. In our case we have allowed users to delete and update the data. If you click Edit button you will be able to change the information about the item:
Using the browser you can add, delete and update the items while other app users will see the changes on their devices. And vice versa: app users can make changes to the items on their devices, and you will see them in the browser.

To publish an HTML5 app click Submit/Publish button:

Now see there’s an option to publish your app as an HTML5 app:
Upon publishing, you will be prompted to enter the app name:

Enter a name of your HTML5 Web app. This name will be a part of the app link. 
Caution: don’t use these characters: / : * ? " <> | 

Web Name: HTML5 Emulator app

When you click OK, you will see the link to your app:
You need to share this link with those users you would like to have access to your app. If your app was published as a Private app, only invited users can use the HTML5 app version.

10.4 App Submission
Your ready application can be distributed to its end users through the Google Play Store or internally, therefore bypassing the public app stores. Please follow this link, where you will specify all the necessary information we need to successfully submit your application. The rest of this section applies to those who have chosen the Do-It-Yourself option and, therefore, are going to submit their applications themselves.

When your application is ready and has been thoroughly tested, you can submit it to the Google Play Store. However, to proceed, you need to join the Google Play developer programs to have the Google Play developer accounts respectively.

Please keep in mind that after signing up with Android Developer programs, it can take up to a couple days for your account to get approved. During this time, you won’t be able to submit your applications.

10.4.1 Creating developer accounts
Before you can upload your you we will need to create developer account. Google has a onetime fee of $25. You can set up your Google developer account here.
After you set up your developer accounts, you can publish your app to Store.

Some extra steps are required to activate your account in Google. You need to add a user to the developer account so that Snappii can submit your apps to Google Play through our editor.

Note: We will not have access to your other Google information (Gmail, drive, etc…). This is solely to integrate with our app editor to make future app updates hassle-free.

1. Login to your developer account https://play.google.com/apps/publish/
2. Go to Settings
3. Then User Accounts and Rights
4. Click Invite New User Button
5. Input support_android@snappii.com in the Email Address field
6. Select Release Manager Role and then Send Invitation
7. Snappii will accept your invitation
8. Check your email for Snappii’s acceptance letter and complete the steps to grant access.

Don’t worry, you only have to do this process once. If the process looks scary, feel free to contact support at any time via support@snappii.com or a Live Chat. We’re always more than willing to help!

10.4.2 Publishing your app
Once the developer accounts have been created, we can go back to the editor. We’ve streamlined the publishing process to an easy 7 step flow.

1. Click on

2. Select between a Private or Public app.
a. Private is used for apps that are restricted to in-house business use.

b. Public is used for apps that you would like to appear in the stores.

3. Toggle the store you would like your app to appear in.

4. Fill in all credentials.
5. Select the Icon your app will be associated with in the end-users phone, and the splash screen you would like to appear when the user opens the app.
6. Finally, select your preview screenshots you would like the end-user to see when they find your app in the store.

7. Click Submit button and you’re done!
10.4.3 App Information Breakdown

App Name – App name as it will appear in the app stores.
App Category – Specifies what industry app pertains to.
Description – Full description that will appear when user checks application information.
Tags/Key words – Words used to discover the app when typed in search bar.
Short Description – Shorthand description that appears when browsing for app (Only available in Google Play).
App package name – Helps end user discover multiple apps within one business. They are usually formatted as “com.domainname.appname”.
Feature Graphic (1024x500px) – First graphic user will see when seeing description of app inside the Google Play store.
Email Address – App admin’s email address.
Phone Number – App admin’s phone number.
Chapter 11. App Updates and Distribution of Changes
There are two ways to update the content of your application after the app has gone live in the Google Play Store.

11.1 App Resubmission
It is required that you resubmit your application if you need to change one or more of the following:
• Splash screen,
• Icon,
• Name of the app,
• Description,
• Keywords,
• Category,
• Website of the app, or
• Language.

If you have added new elements such as tabs or buttons, which are not supported by the currently submitted version of your app, you will get a message telling you to resubmit your application. When you resubmit your application, the review process usually takes a few days.

11.2 App Publishing
It is enough to use the Publish option when you need to change:
• App color theme,
• Content of the tabs and elements, such as a phone number, email, text, Url, or their order in the app;
• The name or icons of tabs and elements.

The changes should be visible next time the users of your application open it.
Whenever you are not sure whether you should publish or resubmit your app, try to publish it first and, if no changes have been made in the live app, resubmit it.

Chapter 12. White Label and the Dashboard

• White Label
Our white label solutions enable you to add your own company’s brand, such as your own splash screen and icon, and get an app free from Snappii’s ads and *Powered by Snappii* button.

If you want to enable this option, you can set it up in your Account Settings after you log in to the Snappii website.

![Profile data](image)

If you put a checkmark here, all the invited users will receive a link to My Preview App that doesn’t have Snappii branding.

- **Dashboard**

Our dashboard helps you track your app usage. Simply go to snappii.com, log in, click on the Dashboard, and choose an app you need to analyze.
As a registered (non-subscribed user), you are able to access the Snappii dashboard features 5 times to test your app functionality.
Once you become a subscribed customer, you will have unlimited access to the Dashboard. You can see there are numerous reports. You can choose to see you app users over a specific period of time.

**The First Time Users Option** shows users who use your app for the 1str time

**The Usage Frequency Option** shows the number of times your app has been run by users

**The Active Users Option** shows users' activity over time

**The Registered Users Option** shows when and how many users have registered.