

Data-Driven Mobile Application Lab Guide

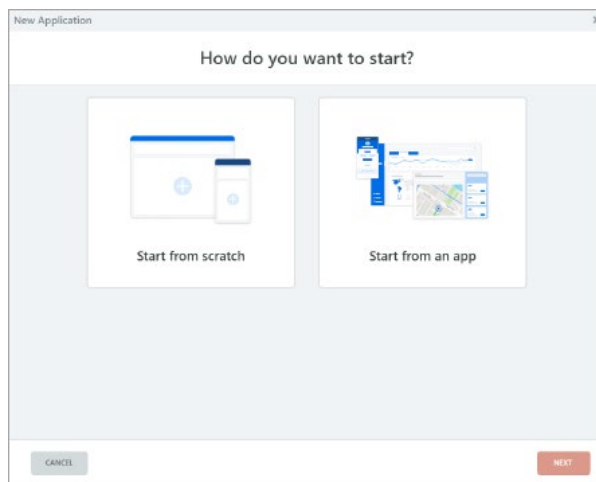
Exercise 1: Importing Data

In this exercise, you will create a new application that imports data from an Excel file.

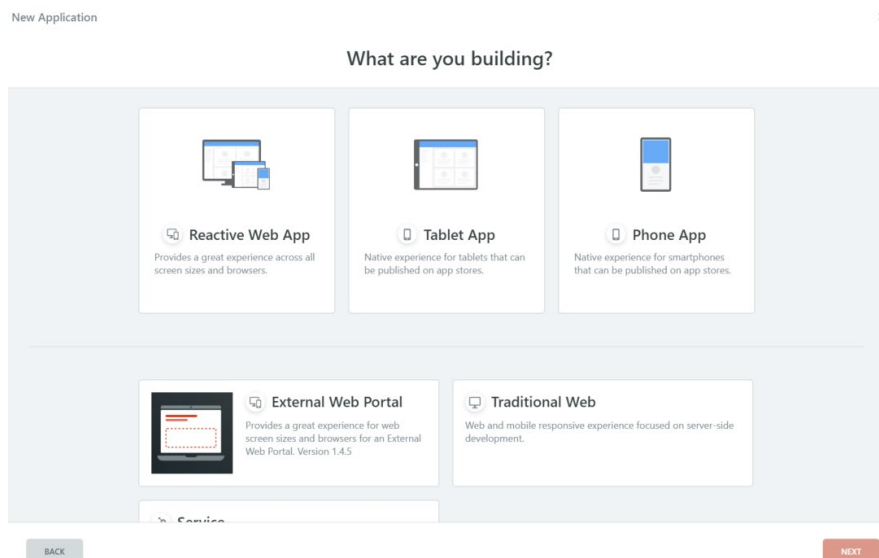
1. Create a new mobile application named **Data-driven App**, with a **Phone App** Module
 - a. In the 'Applications in Development' area, click **New Application**.



- b. In the **New Application** dialog, select **Start from scratch**, and then click **Next**.



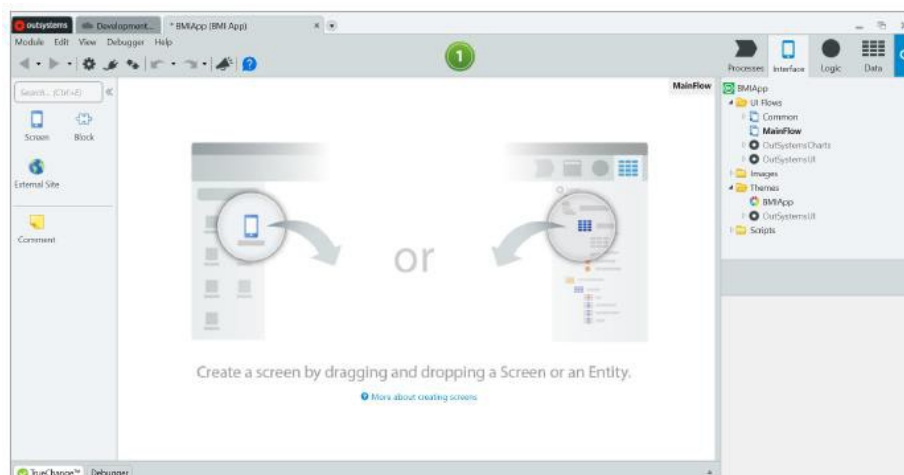
- c. Select the **Phone App** template, and then click Next. The templates provide a starting point for the application, containing the layout structure for the mobile app.



- d. Set the Application Name to **Data Driven App**.
- e. Type in a simple description for the application.
- f. Select **Create App**.

- g. In the **Modules** area, the list of modules of the application can be found. Specify the **Module Name** as **DataDrivenApp** and select the **Phone App** module type. Click **Create Module** to create the module.

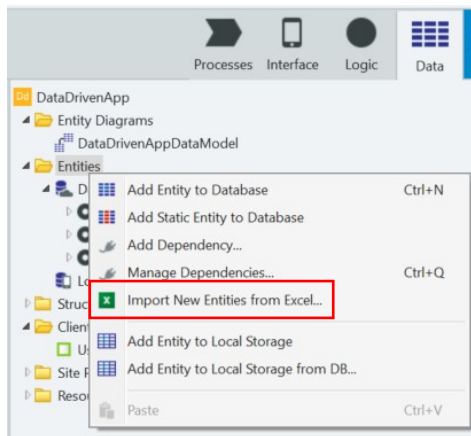
- h. Notice that the application is created and the module is opened. You should see the workspace as shown below.



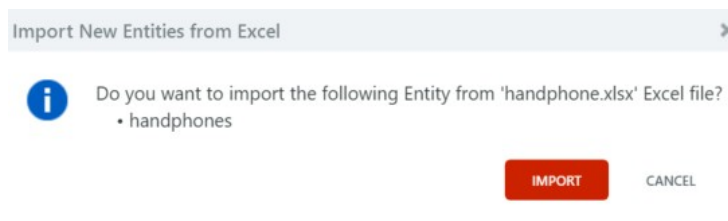
- i. Open the **handphone.xlsx** file given to you to look at the content of the file. We are going to import the data in the Excel file into the app later.

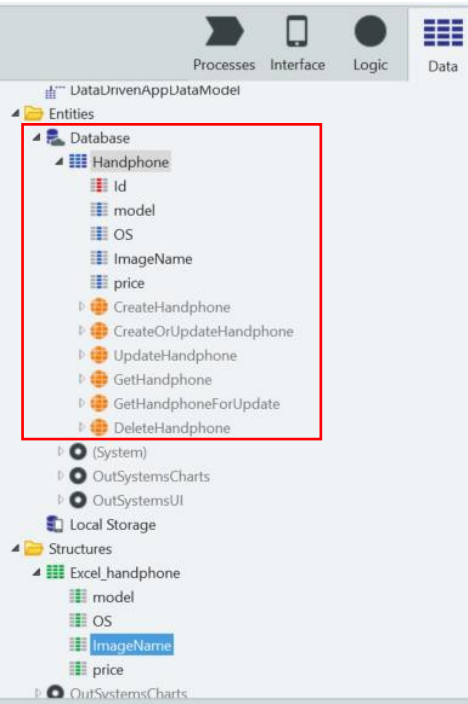
	A	B	C	D
1	model	OS	ImageName	price
2	xiao mi	android	android.png	500
3	iphone	ios	apple.png	1400
4	ipad	ios	apple.png	700
5	note 5	android	android.png	800
6	huawei pro 30	android	android.png	1200

- j. In the **Data** Tab, right-click on entities, and select **Import New Entities from Excel...**

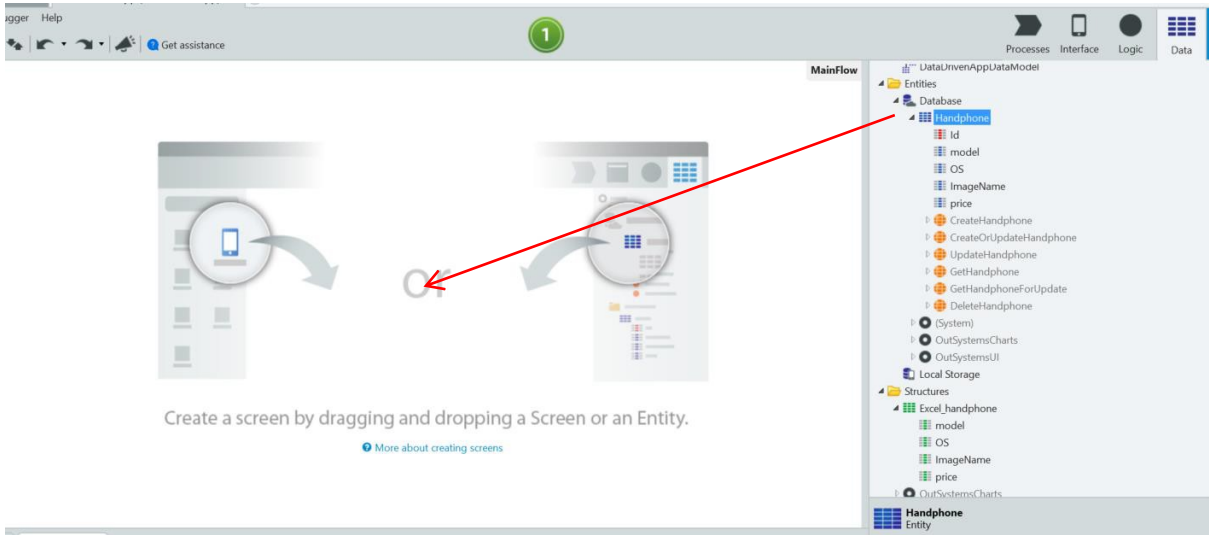


- k. Select **handphone.xlsx** file given to you, and click on the **Open** button. Click on the **IMPORT** button to import the Excel file into the project.





2. Drag the **Handphone** entity into the centre region of the workspace.



- 3. Two screens will be created using the data in the **Handphone** entity.
 - a. Handphones
 - b. HandphoneDetail




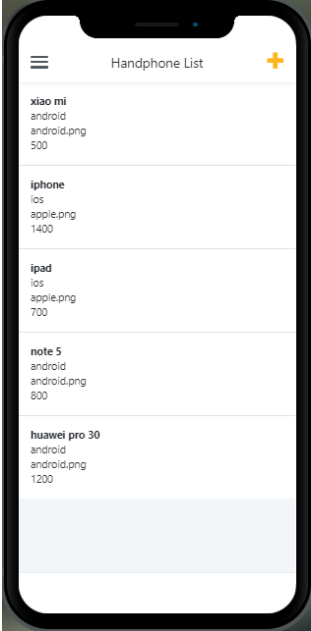
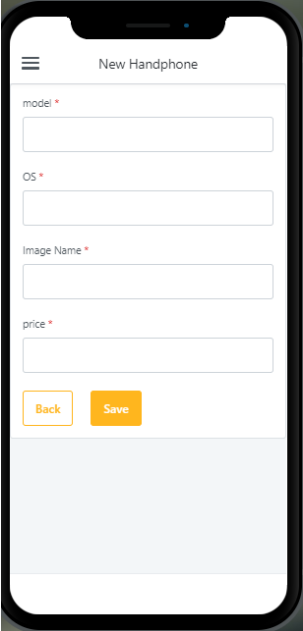
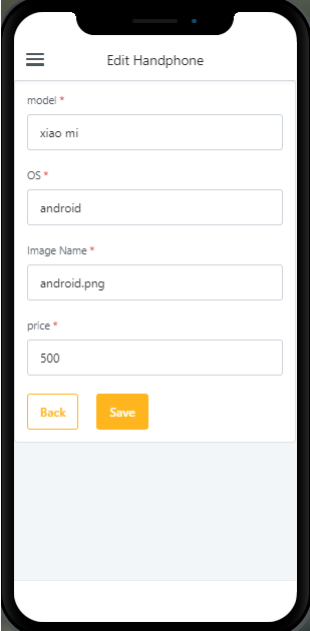
- 4. Double-click on each of the screen to see the user-interface that have been generated.

HandphoneDetail Screen	Handphones Screen
<p>The HandphoneDetail screen features a top navigation bar with a hamburger menu icon on the left and two buttons, "Edit Handphone" and "New Handphone", on the right. Below the bar is a form with four input fields labeled "model", "OS", "Image Name", and "price". At the bottom of the form are two orange buttons: "Back" and "Save".</p>	<p>The Handphones screen features a top navigation bar with a hamburger menu icon on the left, the text "Handphone List" in the center, and a plus icon on the right. The main content area displays a list of three items, each consisting of the text "xiao mi", "android", "android.png", and "500". At the bottom of the screen is a circular home indicator.</p>

5. Go to both the **Handphones** and the **HandphoneDetail** screen. Click on Anonymous under Roles for both screens.

Handphones Screen	HandphoneDetail Screen
Name: Handphones	Name: HandphoneDetail
Description: ...	Description: ...
Title: ...	Title: ...
Roles:	Roles:
Anonymous: <input checked="" type="checkbox"/>	Anonymous: <input checked="" type="checkbox"/>
Registered: <input checked="" type="checkbox"/>	Registered: <input checked="" type="checkbox"/>

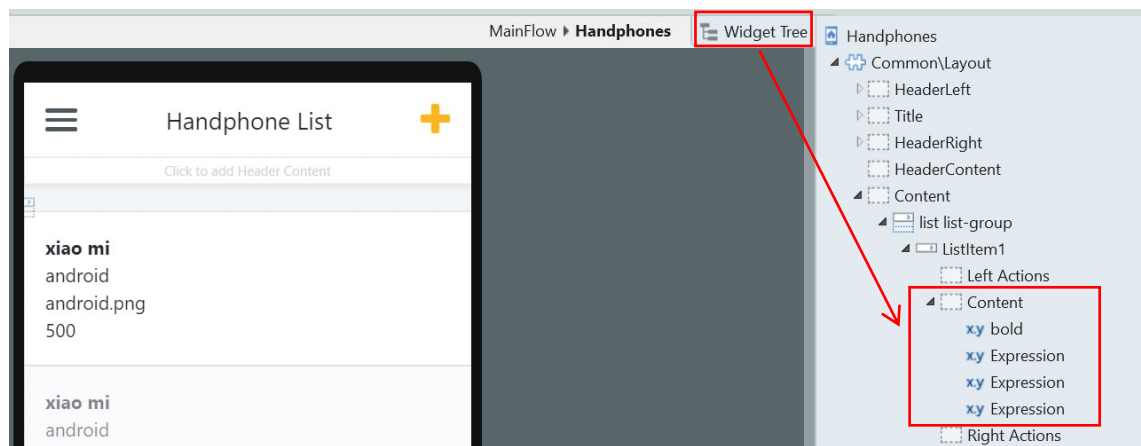
6. You may click on the  1-Click Publish button to update and test the app. Click on the + icon or the item in the listview to go to the Detail screen.

ListView with the content taken from the Handphone Entity	New Handphone view upon clicking the '+' button	Edit Handphone view upon clicking on the list item. The form is pre-filled with existing data
		

Exercise 2: Modifying the Screen generated from the Data

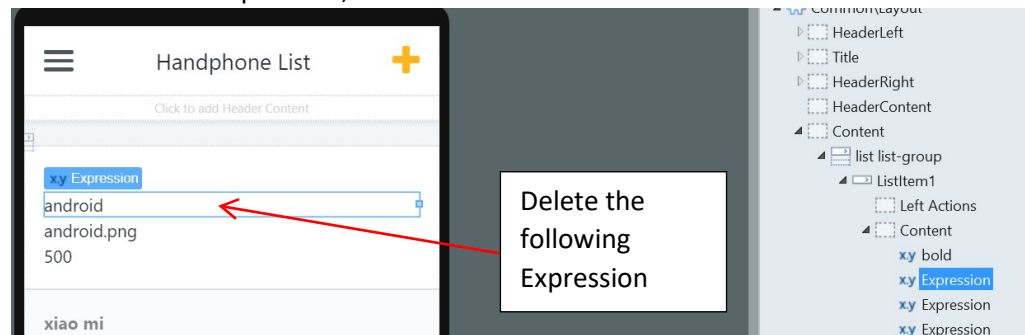
After creating the screens from the data, it is common that we will want to make some changes on the UI components generated. For example, we will want to show the image instead of the file name: apple.png and android.png. We will also want to re-arrange how the information is displayed on the screen. In this exercise, we will learn how to make some change to the generated screen.

1. Open the **Handphones** screen, click on **Widget Tree** to show the hierarchical structure of the UI components in the in the screen. You can see that there is a **List** widget, which contains a **List Item** with 4 **Expression** components.

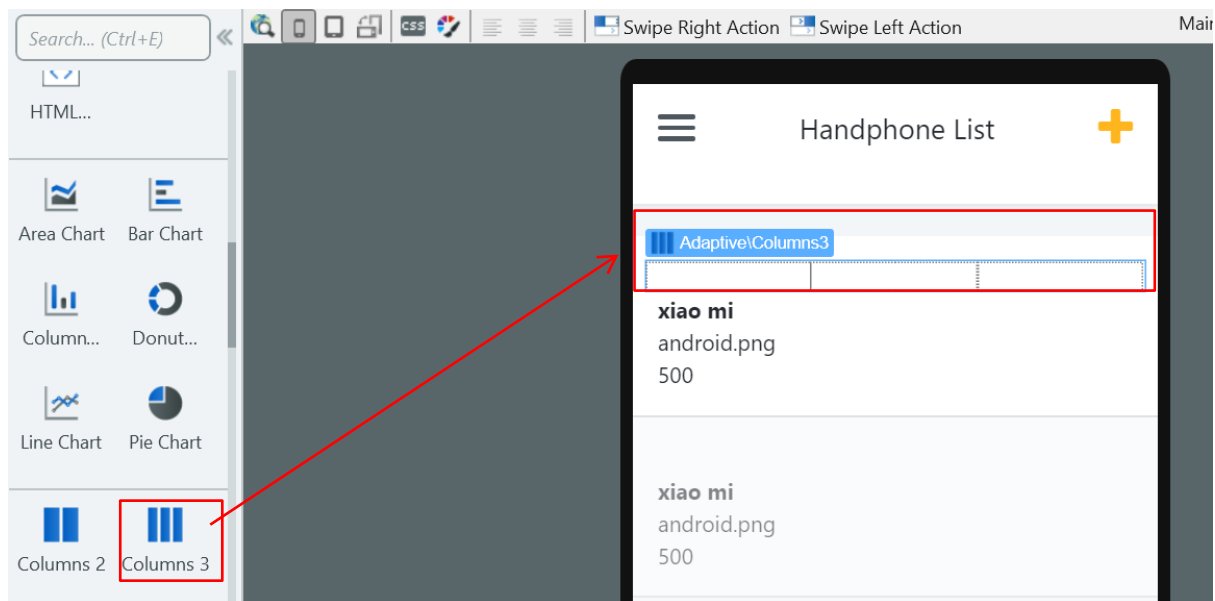


2. We will re-arrange the **Expression** components so that they will be lined out horizontally next to each other.

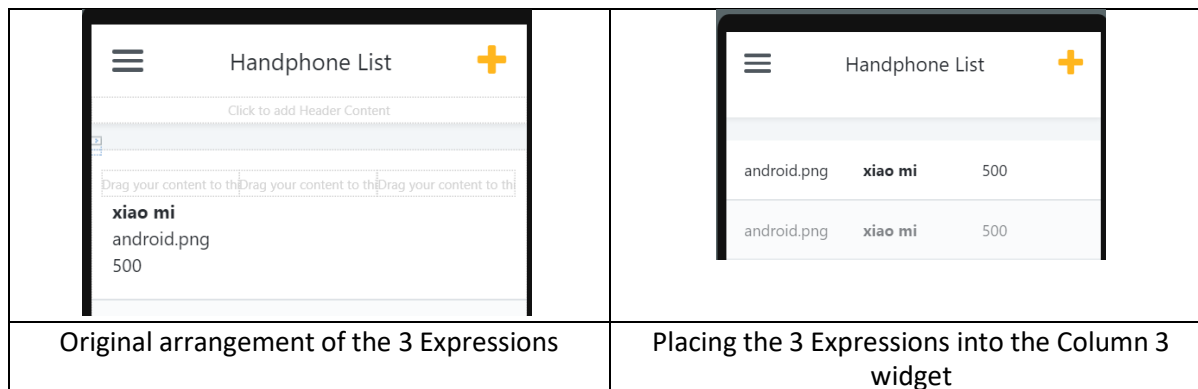
- a. Delete the 2nd Expression, the one that shows Android.




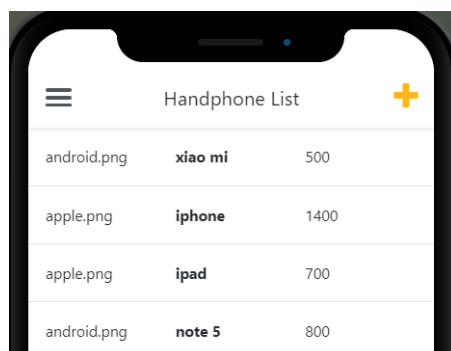
- b. Drag the **Columns 3** widget into the space just above the **xiao mi** Expression.



- c. Drag the 3 Expressions into the **3 columns** as shown below. Take note that **android.png** is placed on the left-hand side.

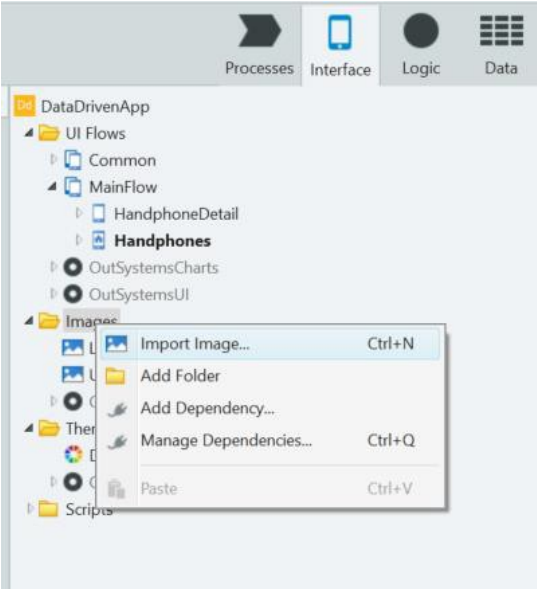


3. You may click on the  1-Click Publish button to update and test the app. You will see that the information is arranged horizontally now.

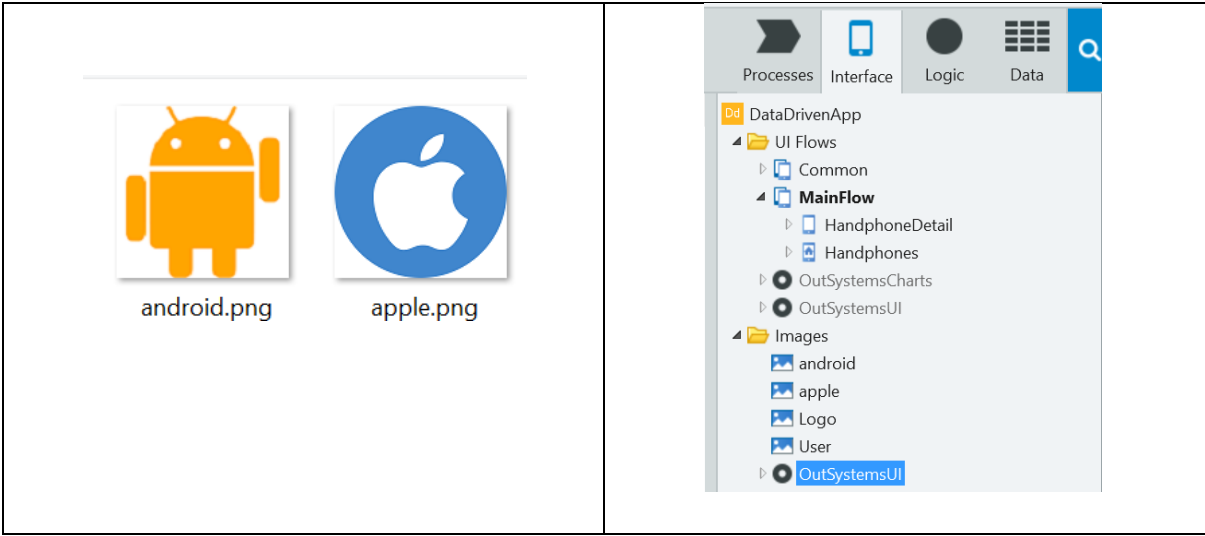


In the next section, instead of displaying the image name, we will display the image itself. Follow the next few steps to complete this task.

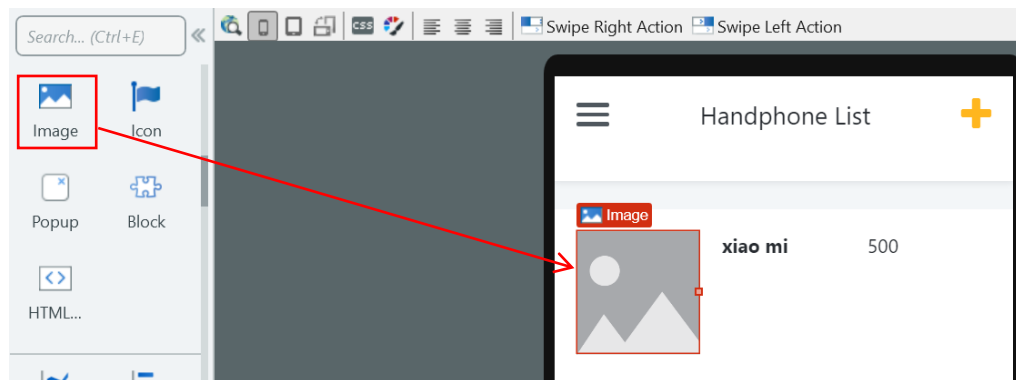
4. Select the Interface tab, right-click on Images, and select **Import Image...** .





5. Upload the following 2 images given to you. You will see 2 additional items in the Images folder if you have uploaded the images successfully.



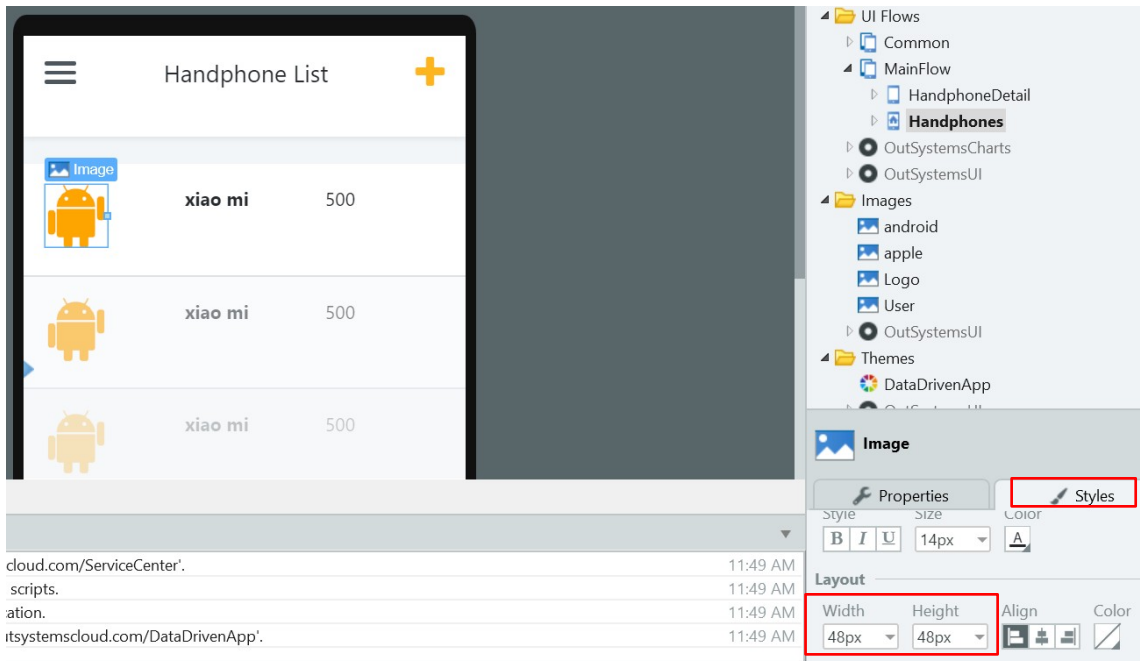
6. Delete the **Expression** showing the image file name, and replace it with an **Image** widget. There will be a red boundary shown around the Image widget because the file name has not been specified.



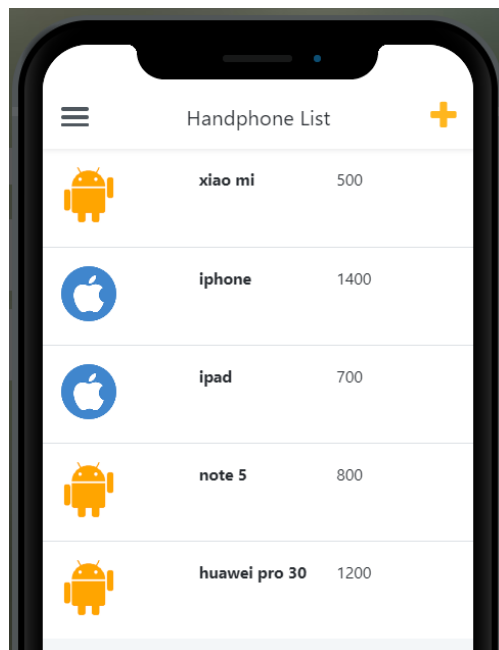
7. Click on the Image widget, and update the properties to the following:
- Type: Local Image
 - URL: `"/DataDrivenApp/img/DataDrivenApp." + GetHandphones.List.Current.Handphone.ImageName`
- [Note: You will need to change the word: **DataDrivenApp** if you have named your app differently].

Default Type: Local Image	Change Image Type to: External URL
 Image	 Image
<div> <div>⚙️ Properties</div> <div>🔨 Styles</div> </div>	<div> <div>⚙️ Properties</div> <div>🔨 Styles</div> </div>
Name	Name
Type	Type
Image	URL
Style Classes	Style Classes
Local Image	External URL
	<code>"/DataDrivenApp/img/DataDrivenApp." + GetHandphones.List.Current.Handphone.ImageName</code>

8. The Image should now show in the screen. Click on the **Styles** tab, and change the height and weight of the Image to **48 px by 48 px** to reduce the size of the image.

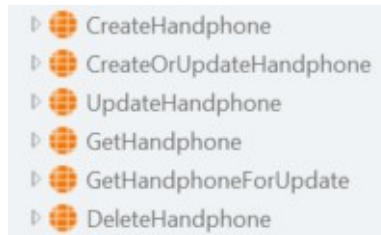


9. You may click on the  1-Click Publish button to update and test the app.



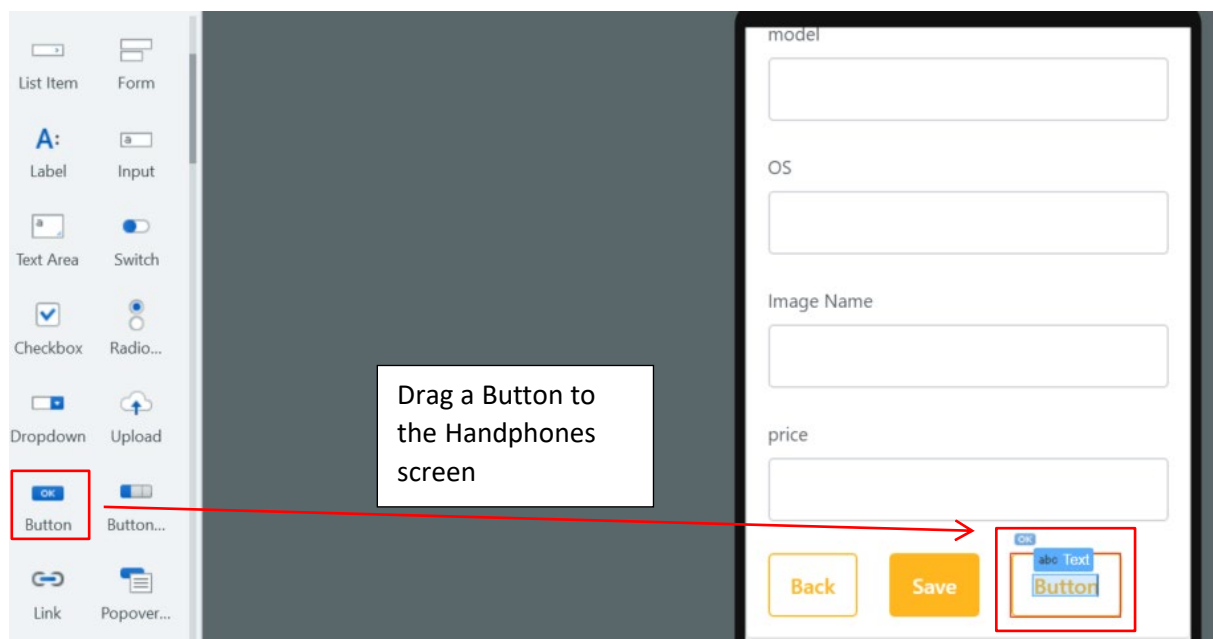
Exercise 3: Adding the Delete Function

When we created the Handphone Database table by uploading the Excel files, a few server methods are created in the process.

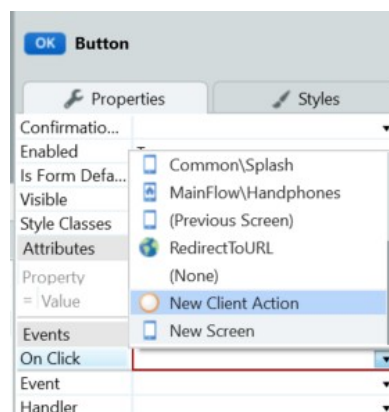


We have made use of the **CreateOrUpdateHandphone** method when we update or create a new Handphone entry. In this exercise, we will use the **DeleteHandphone** method to delete a Handphone entry.

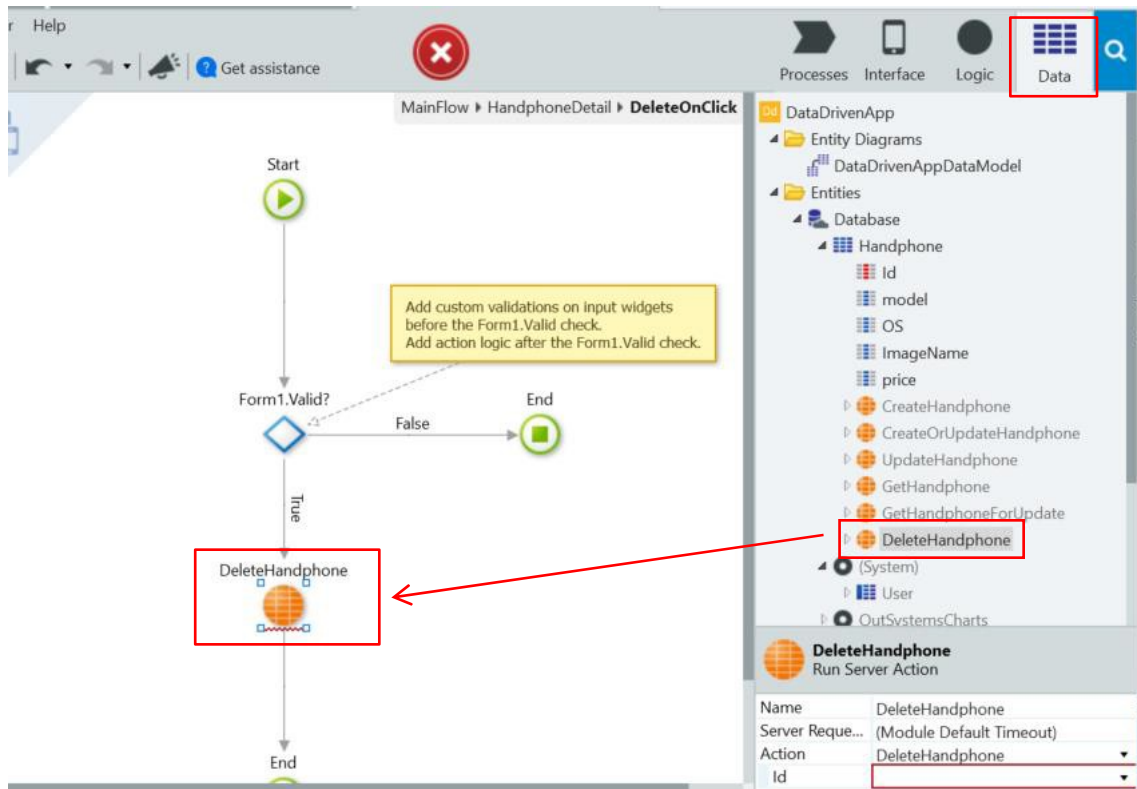
1. Click on the **HandphoneDetail** screen. Add a new **Button** next to the Save Button. Name the Button as Delete.



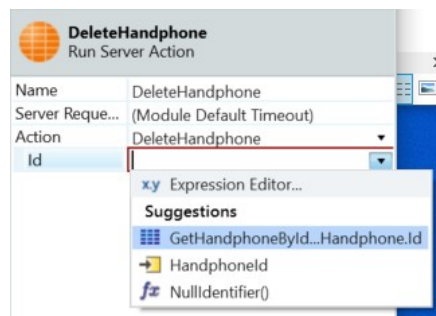
2. An error message will appear. Click on the **Delete** Button, select **New Client Action** for the On Click Event for the button.



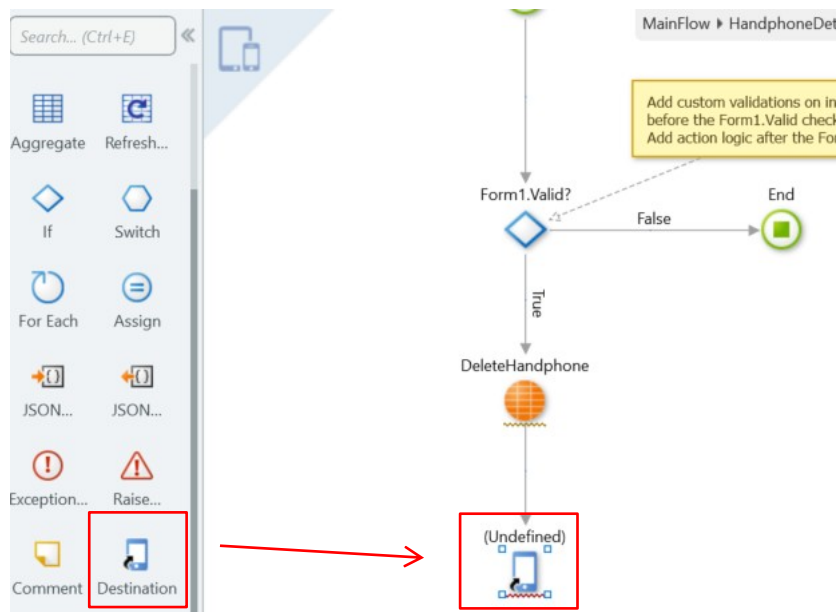
- Click on the **Data** Tab, and Drag the **DeleteHandphone** method to the workflow for the Client Action created.



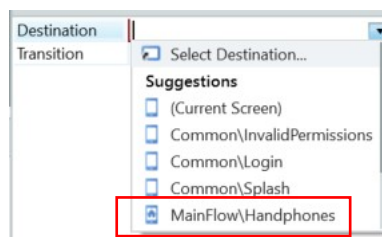
- Click on the **DeleteHandphone** method, and accept the Suggestion **GetHandphoneById...Handphone.Id** to update the **Id** property of the method.



- Drag the **Destination** widget to the workflow, placing it just below the **DeleteHandphone** method.




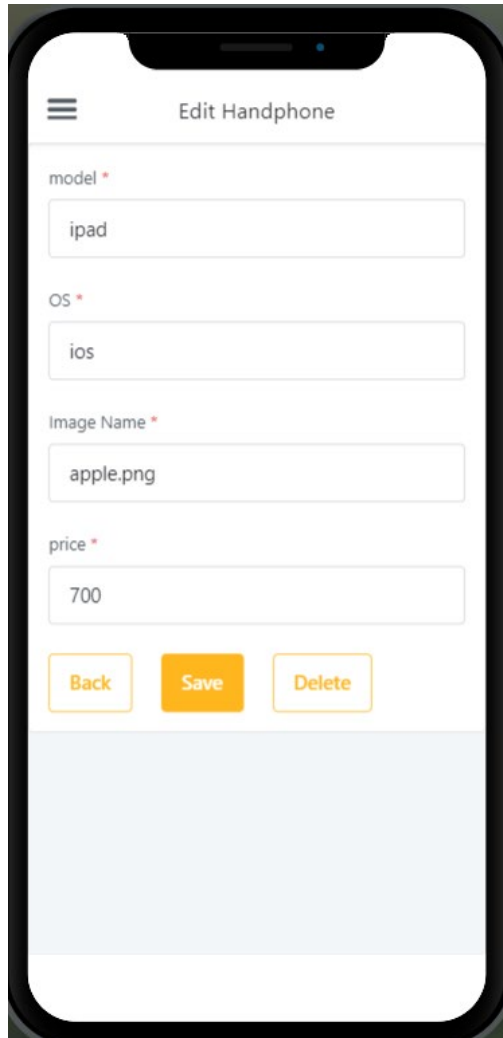
- Click on the **Destination** Widget, change its Destination property to **MainFlow\Handphones**. The purpose of adding the Destination widget is to redirect the app to the Handphones screen after the delete action.



- You will notice an **End** widget somewhere at the bottom of the workflow area. Delete away the End widget.



10. You may click on the  1-Click Publish button to update and test the app. Select a Handphone item and click on the Delete Button to delete the entry. You will be redirected back to the main screen after the delete action.

A screenshot of a mobile application interface titled "Edit Handphone". The screen features a white background with a black header bar at the top. On the left side of the header is a hamburger menu icon, and on the right is the title "Edit Handphone". Below the header, there are four text input fields, each with a red asterisk indicating a required field. The first field is labeled "model" and contains the text "ipad". The second field is labeled "OS" and contains the text "ios". The third field is labeled "Image Name" and contains the text "apple.png". The fourth field is labeled "price" and contains the text "700". At the bottom of the form, there are three orange buttons with black text: "Back", "Save", and "Delete". The "Save" button is slightly larger and more prominent than the others. Below the buttons is a large, empty light blue rectangular area.

End of Lab 3