Fault Reporting Application Lab Guide

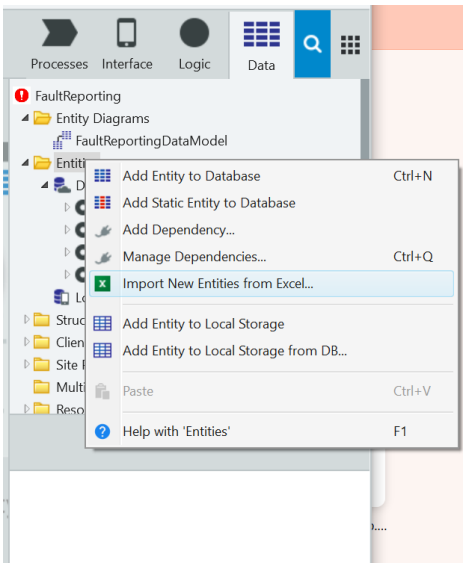
Exercise 1: Import Data and Create Default Screens

1. Create a new mobile application named **Fault Reporting 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 ***Fault Reporting 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 **FaultReporting** and select the **Phone App** module type. Click **CreateModule** to create the module.

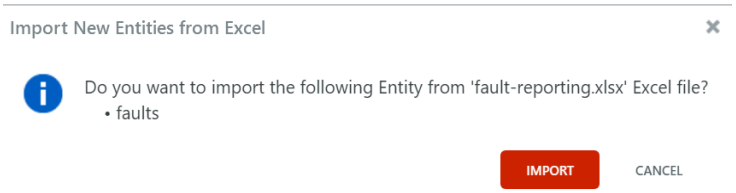
2. Open the **fault-reporting.xlsx** file given to you to look at the content of the file. We are going to import the data in the Excel file into the app later.

	A	B	C	D
1	title	description		
2	broken chair	chair in classroom has broken wheel		
3	lighting problem	light is flickering in E6-level 7		

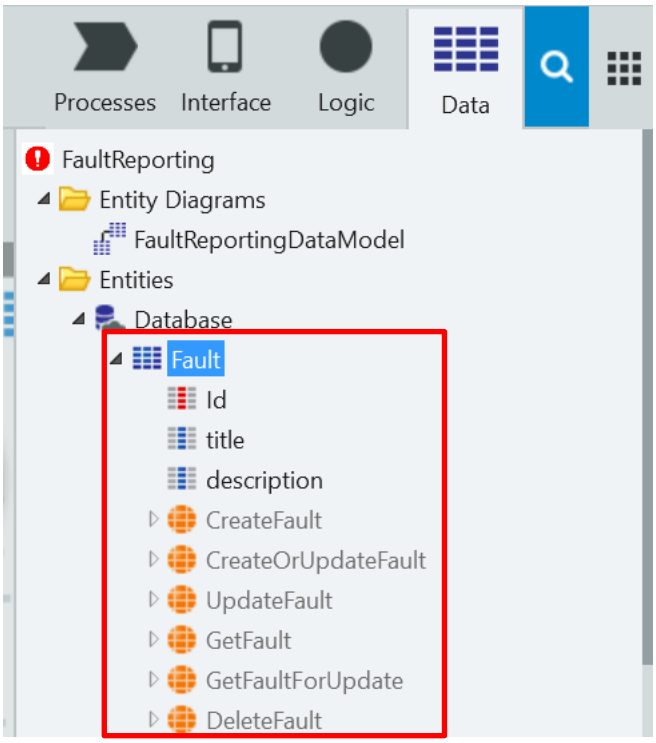
3. In the **Data** Tab, right-click on entities, and select **Import New Entities from Excel…**



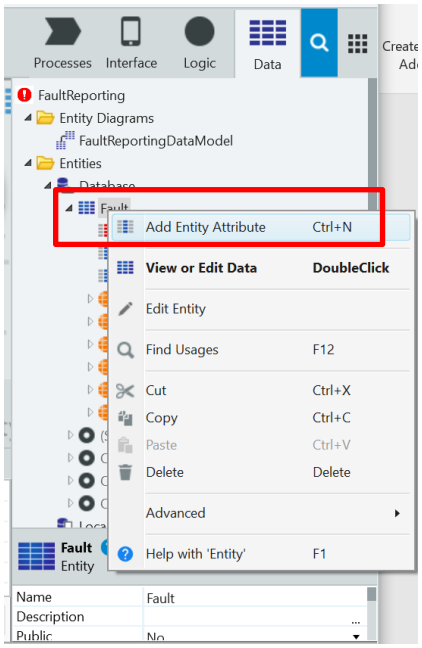
4. Select **fault-reporting.xlsx** file given to you, and click on the **Open** button. Click on the **IMPORT** button to import the Excel file into the project.



5. Under the Database, you will see a **Fault** entity with its attributes and somehelper database methods.

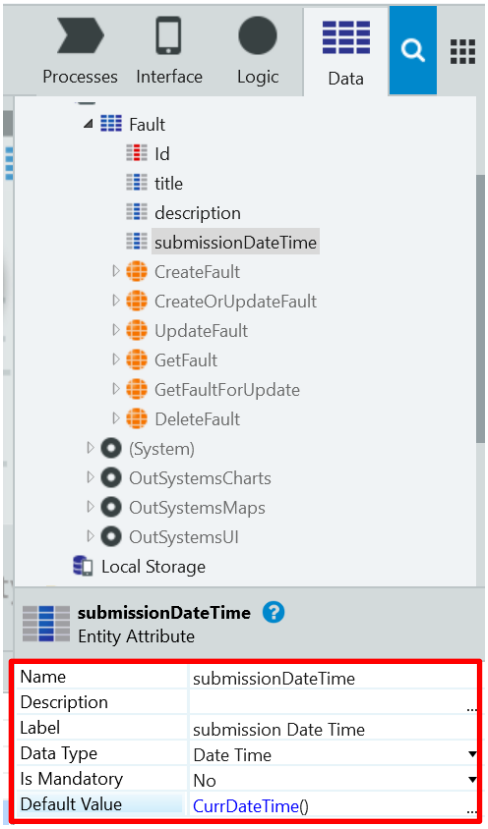


6. Right click on the **Fault** entity, select **Add Entity Attribute**.



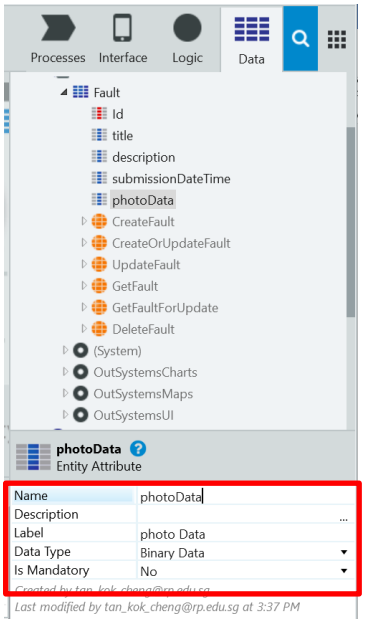
7. Change the Attribute name to **submissionDateTime**. Set the following properties for the attribute:

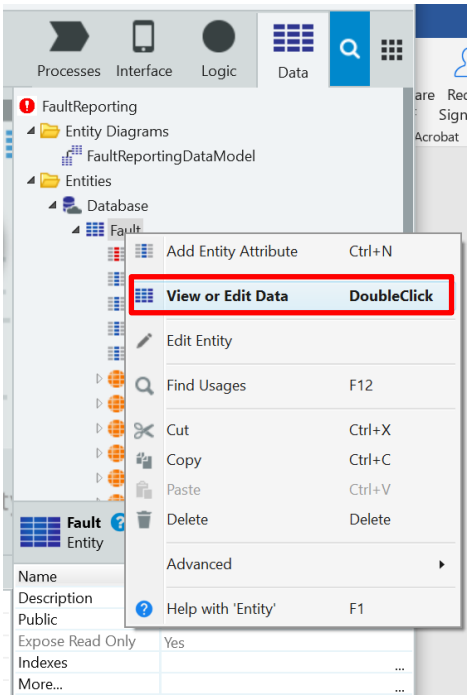
Name	submissionDateTime
Data Type	Date Time
Default Value	CurrDateTime()



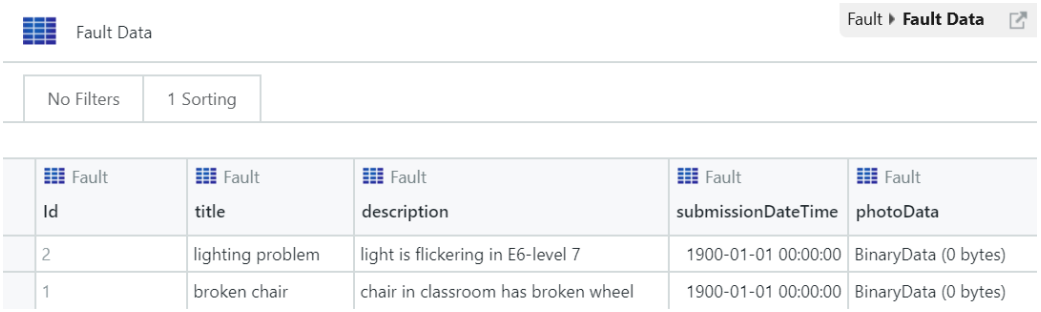
8. Follow the same process in step 7 to add another attribute. Set the following properties for the new attribute:

Name	photoData
Data Type	Binary Data



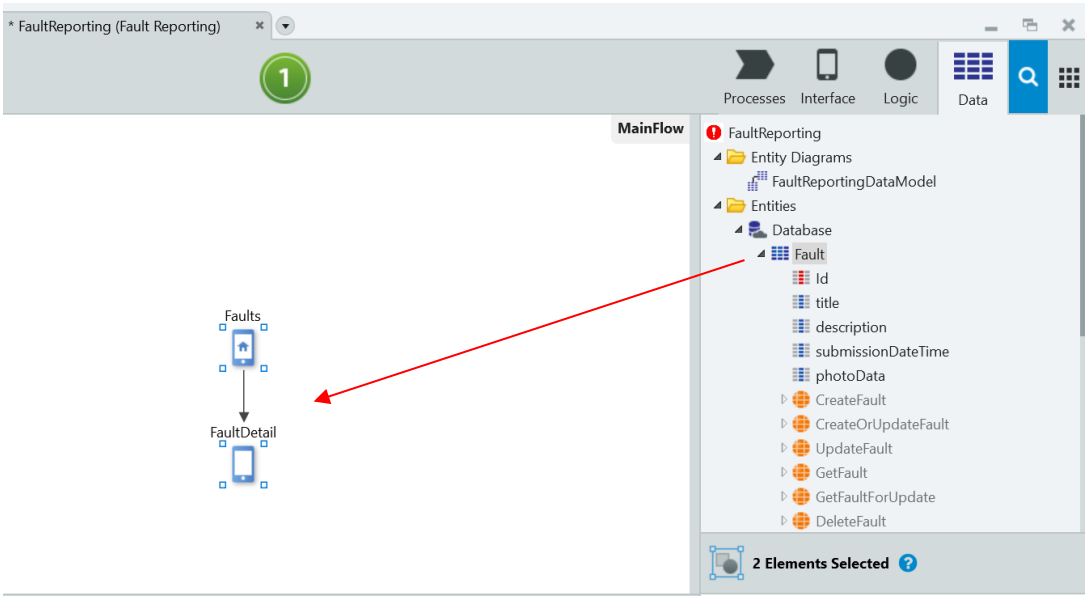


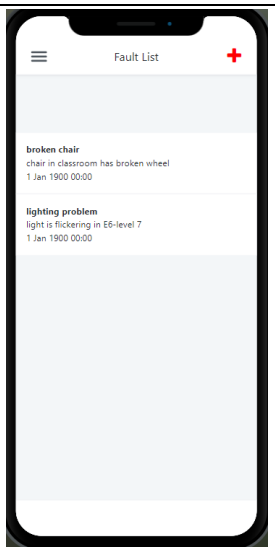
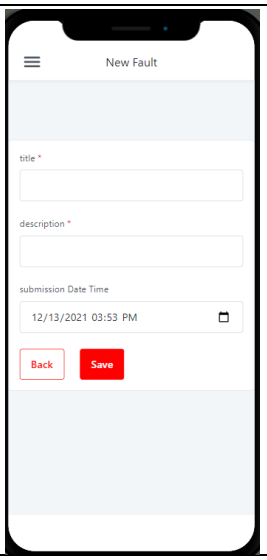
10. You will see 2 rows of data that is imported from the Excel file. **submissionDateTime** and **photoData** will be filled with some default or empty data for now.



Fault Id	Fault title	Fault description	Fault submissionDateTime	Fault photoData
2	lighting problem	light is flickering in E6-level 7	1900-01-01 00:00:00	BinaryData (0 bytes)
1	broken chair	chair in classroom has broken wheel	1900-01-01 00:00:00	BinaryData (0 bytes)

11. Click on the Interface tab, and double click on Main Flow. Next, click on the Data tab, and drag the Fault entity to the center of the screen as shown.

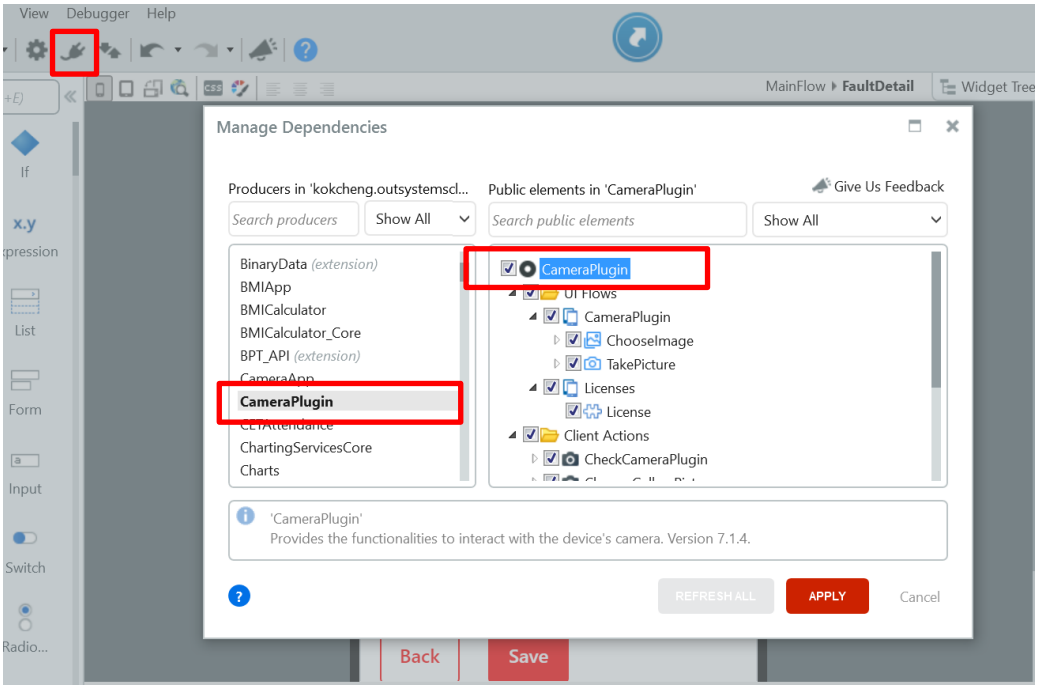


	
Faults Screen	FaultDetail Screen

Exercise 2: Using the Camera Plugin

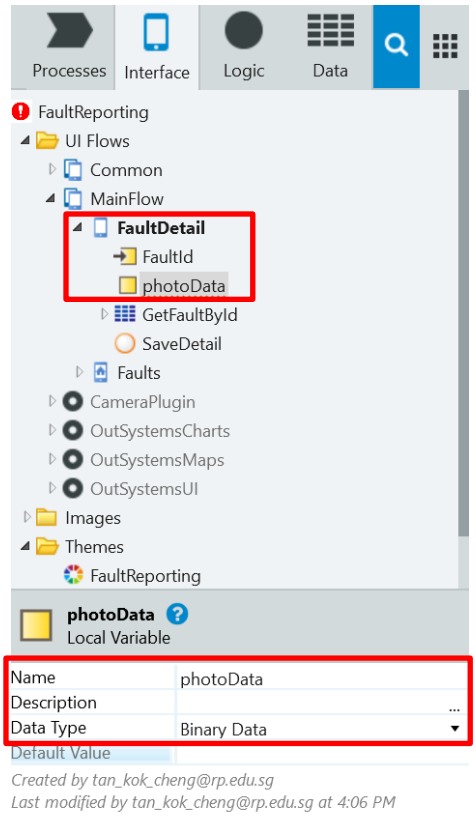
---

1. Click on the "Manage Dependencies…" icon, and select the CameraPlugin. Click on the Checkbox to select all the items for CameraPlugin.

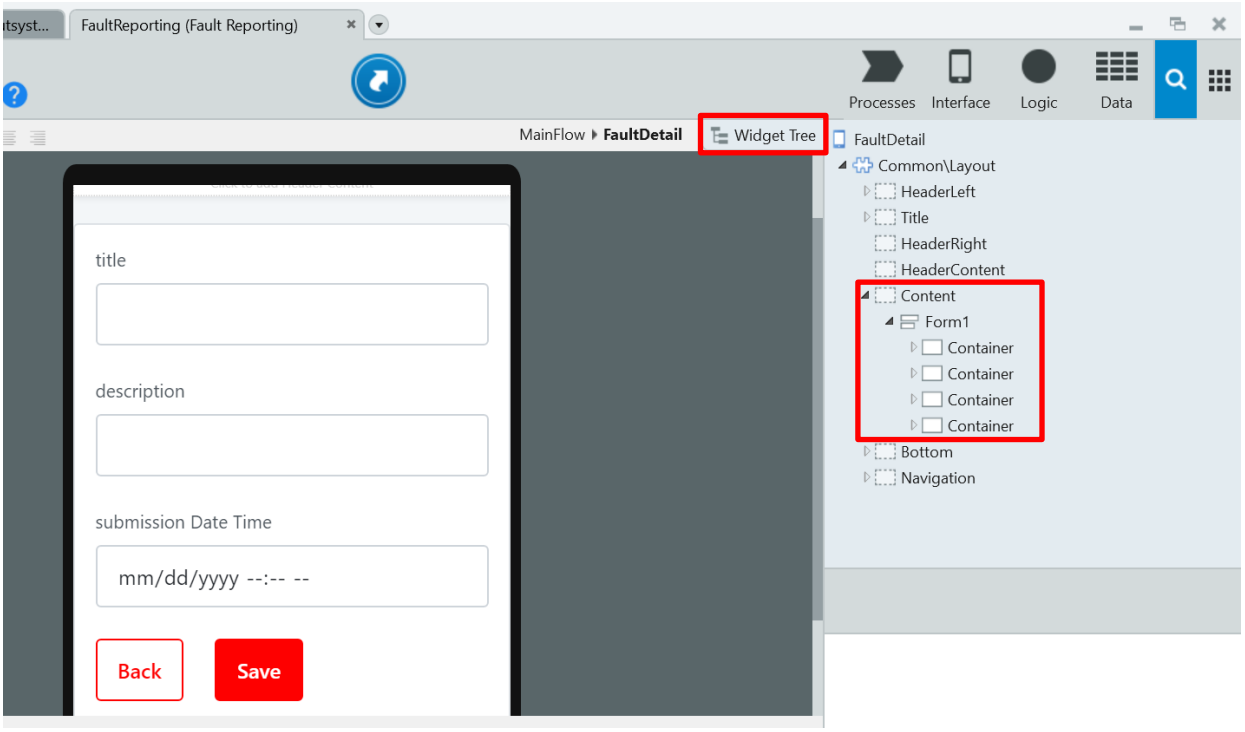


2. Add a new Local Variable to the **FaultDetail** screen. Set the following properties for the Local Variable:

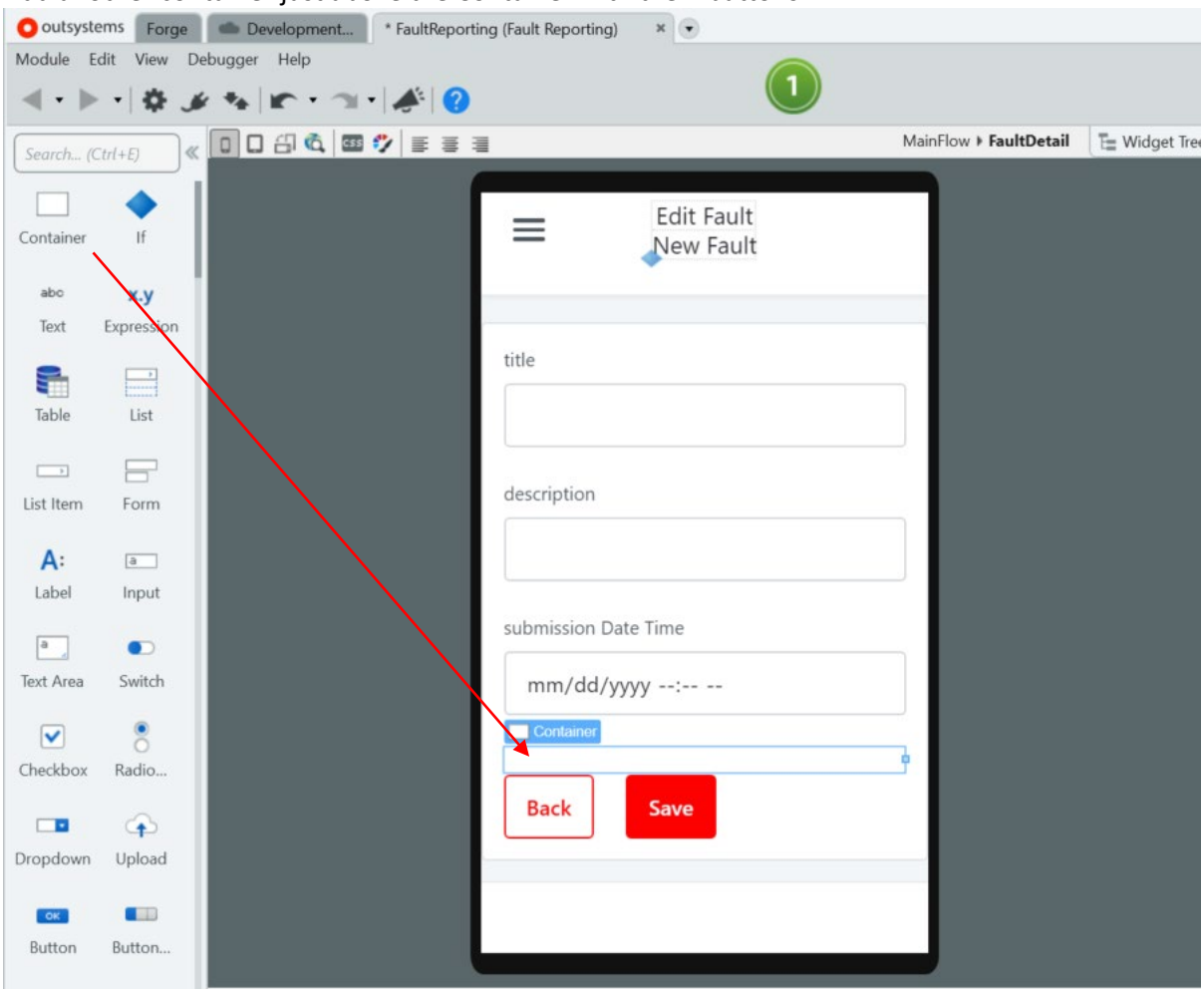
Name	photoData
Data Type	Binary Data



3. Open the **FaultDetail** screen, add click on Widget Tree. You will be able to see that there is a Form with 4 containers for the 4 set of data in the **FaultDetail** screen.

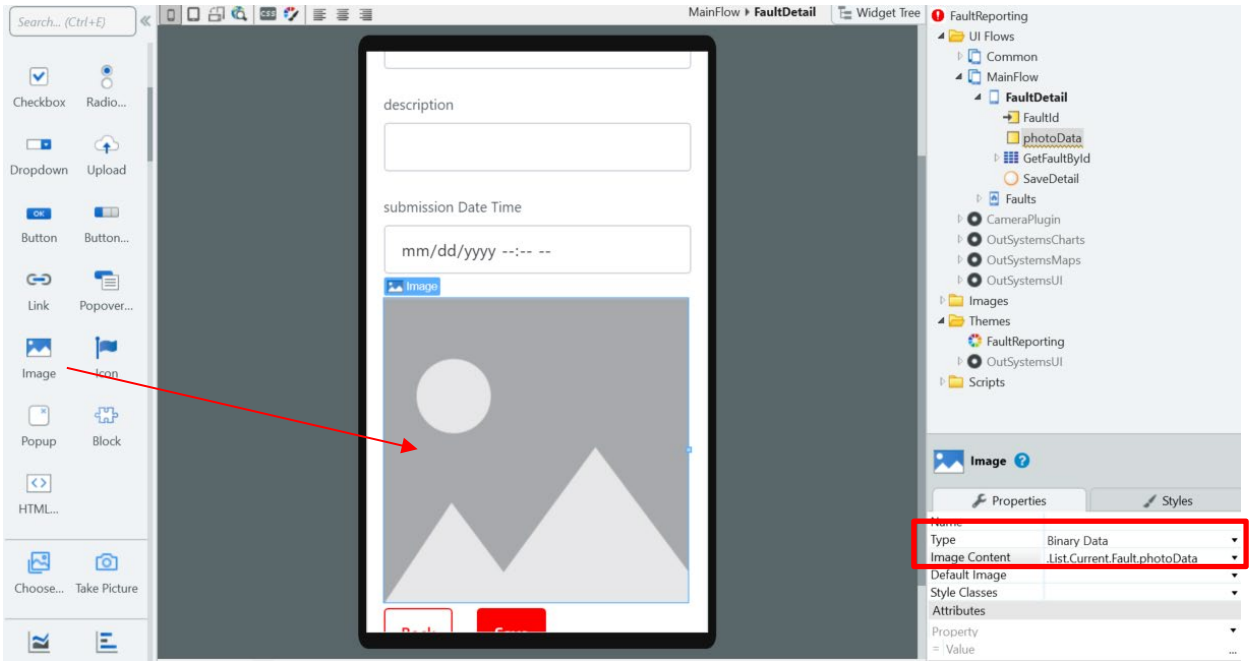


4. Add another container just above the Container with the 2 buttons.

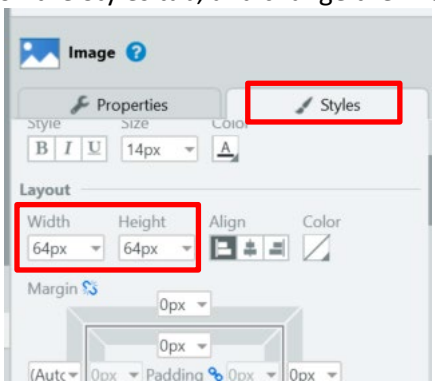


5. Add an Image and place it inside the container. Change the following properties for the image.

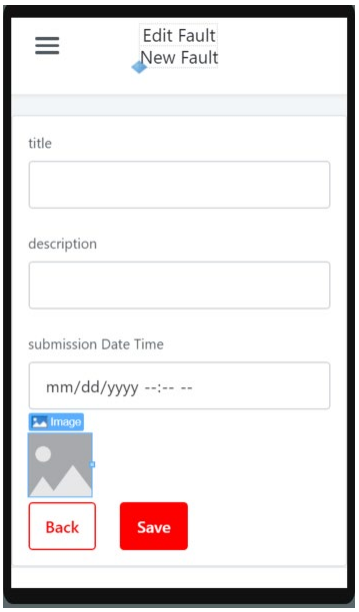
Type	Binary Data
Image Content	GetFaultById.List.Current.Fault.photoData



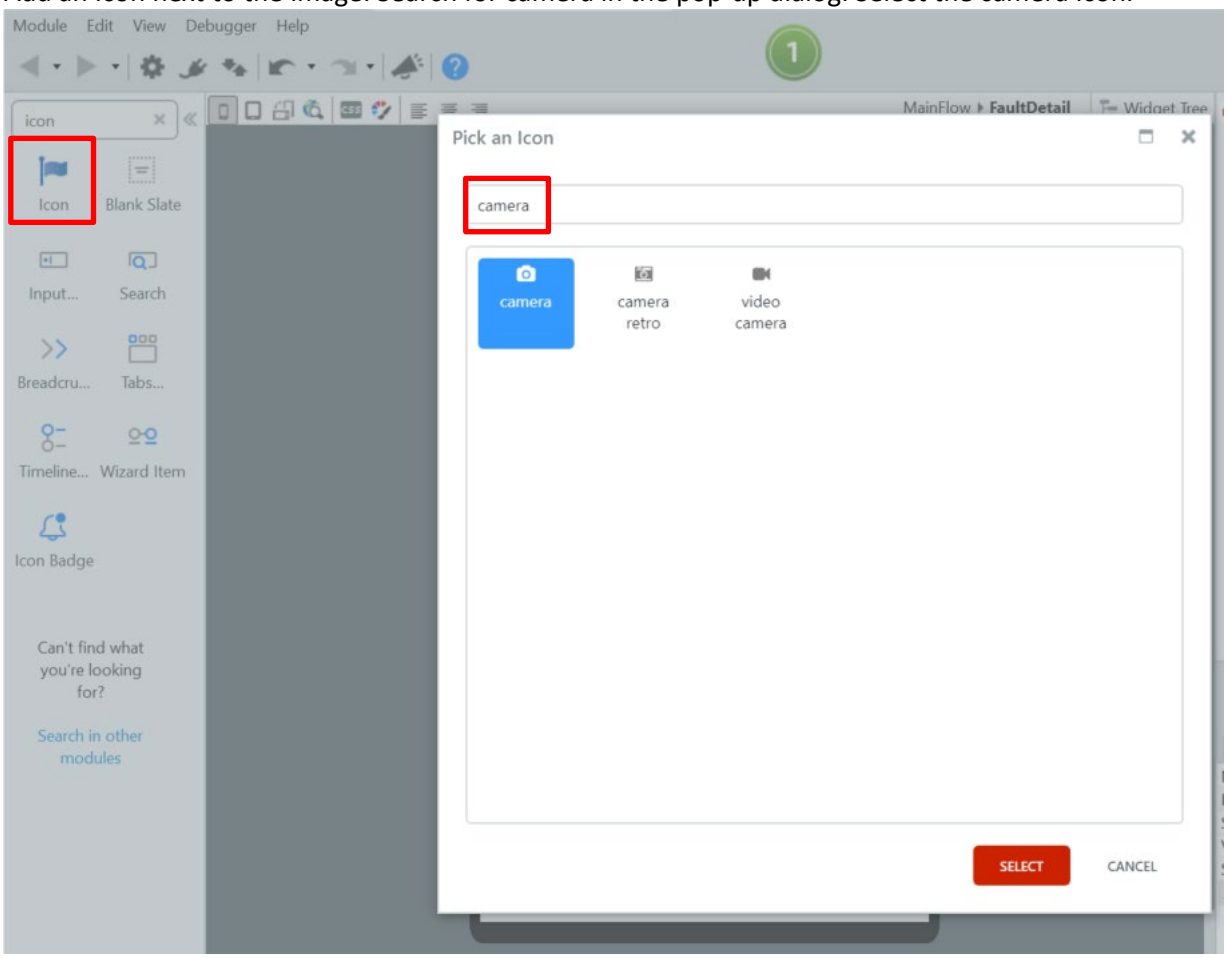
6. With the Image selected, click on the Styles tab, and change the width and height to 64px.



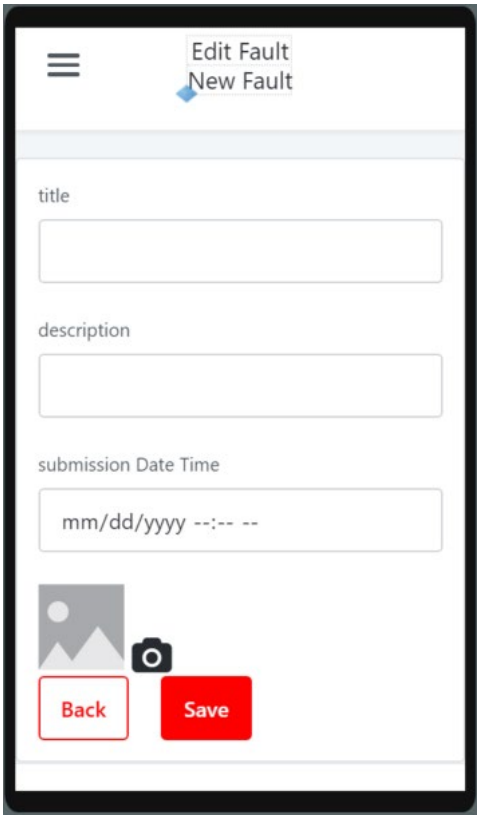
7. The FaultDetail Screen will need be as shown in the screen shot below:

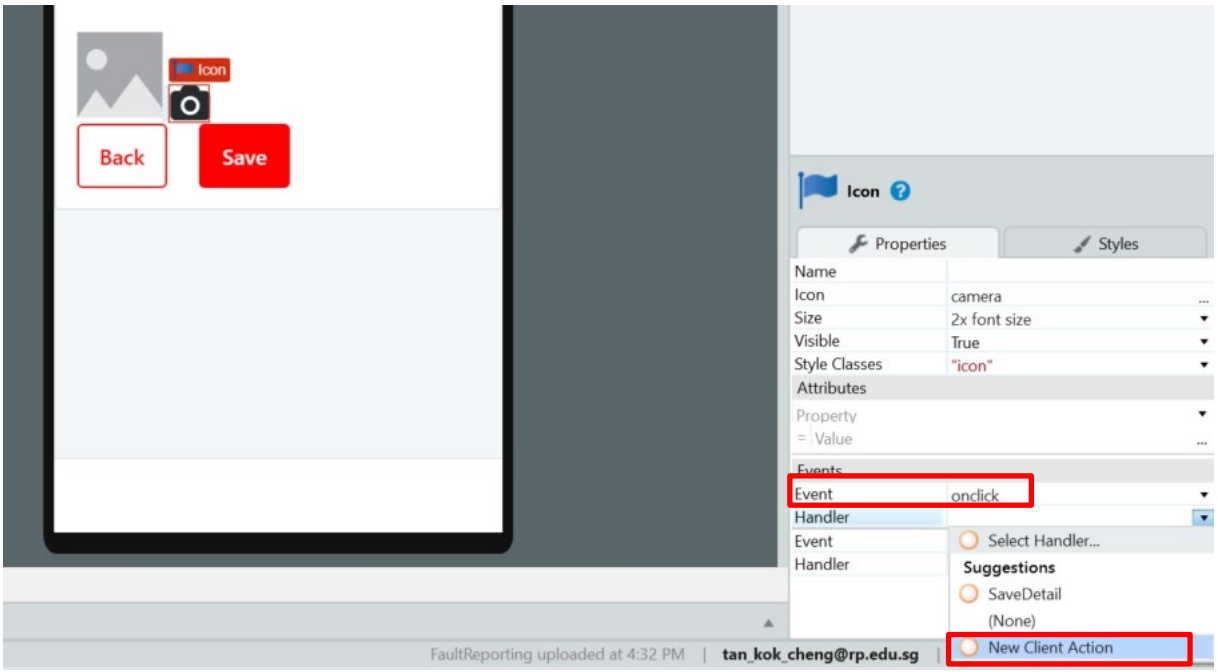


8. Add an Icon next to the Image. Search for camera in the pop-up dialog. Select the camera Icon.



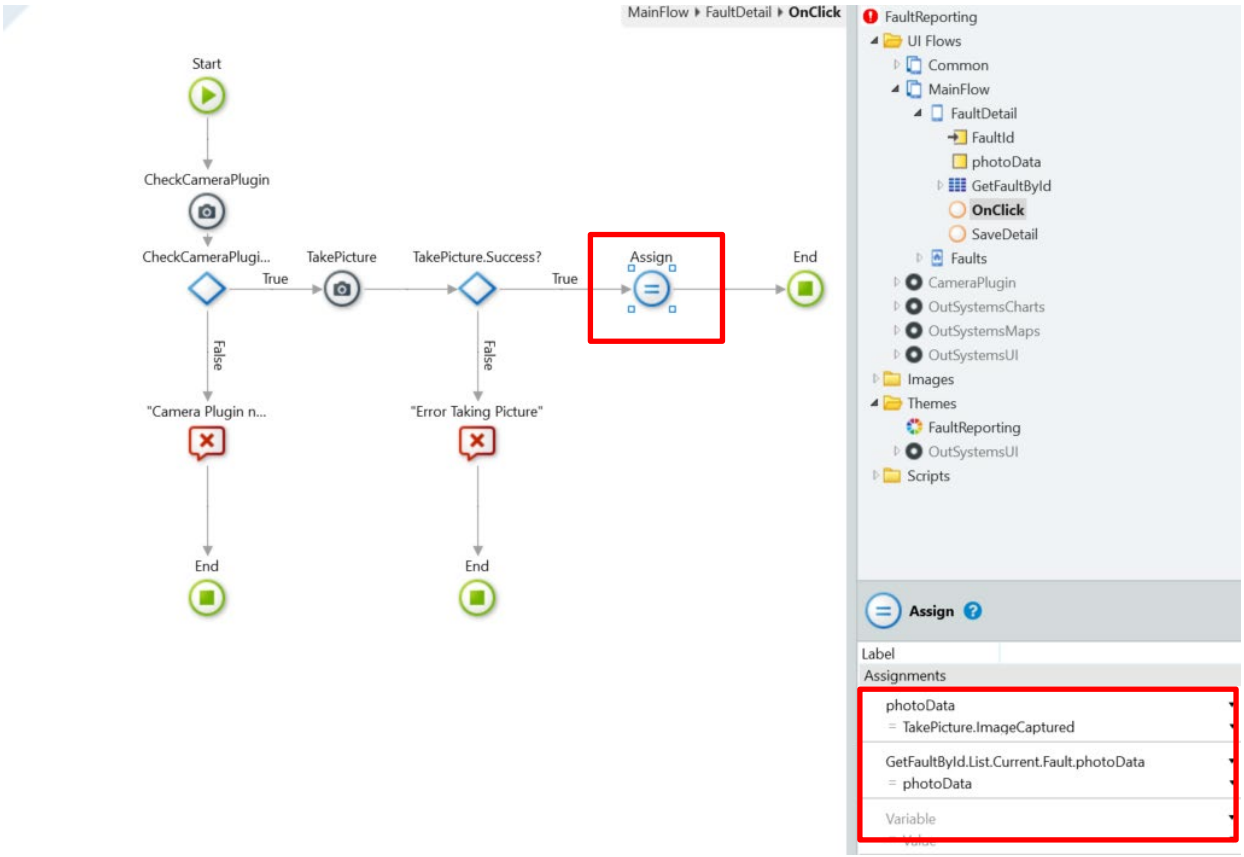
9. The FaultDetail Screen will need be as shown in the screen shot below:



10. Keep the Icon selected. Add an **onclick** event, and select **New Client Action**.
11. 

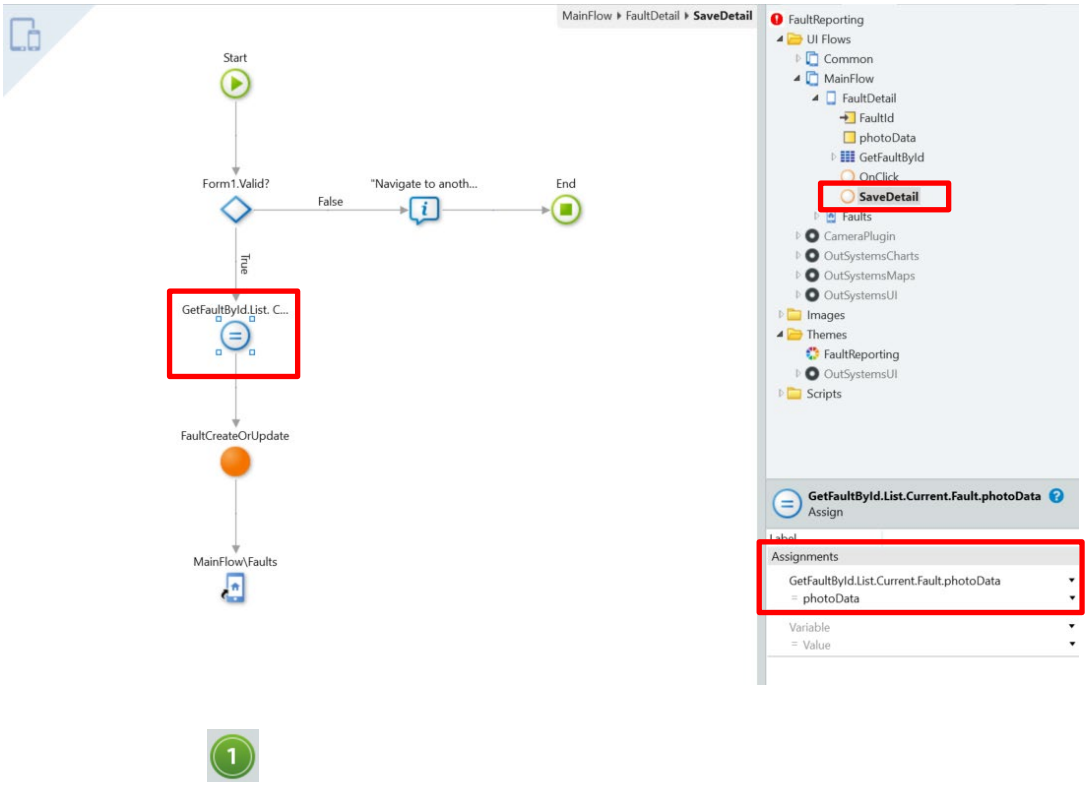
12. An **OnClick** client action will be created. Follow the steps that you have done in Lab 6 for the Camera Plugin to complete the client action as shown below. Add the following properties for the Assign widget:

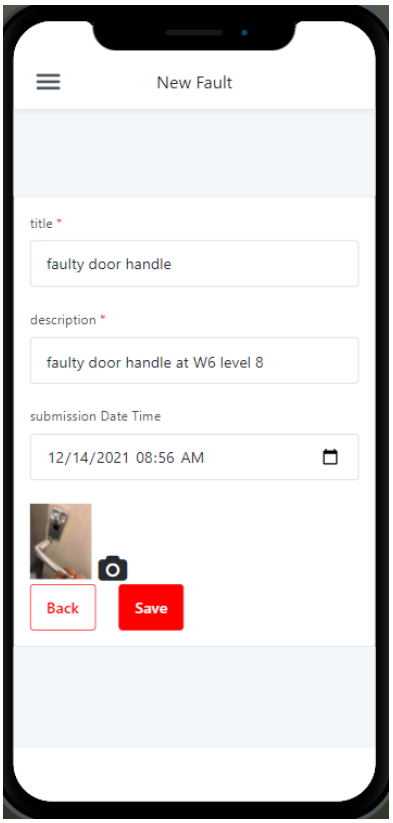
First Assignment	Variable	photoData
	Value	TakePicture.ImageCaptured
Second Assignment	Variable	GetFaultById.List.Current.Fault.photoData
	Value	photoData



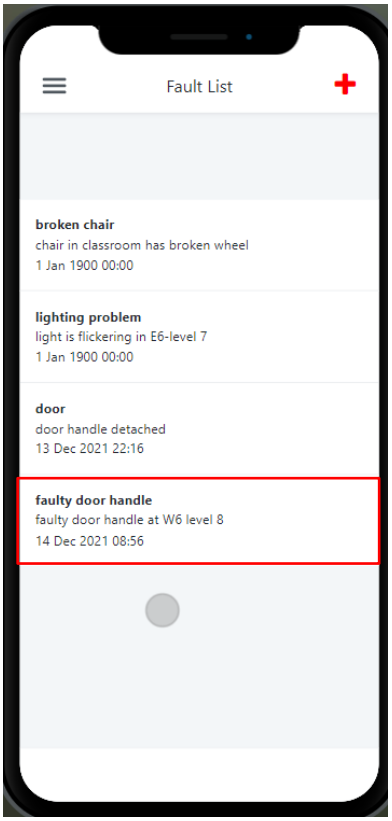
13. Click on the **SaveDetail** client action to show the workflow. Add an Assign widget to the workflow as shown. Add the following property for the assign widget.

Variable	GetFaultById.List.Current.Fault.photoData
Value	photoData





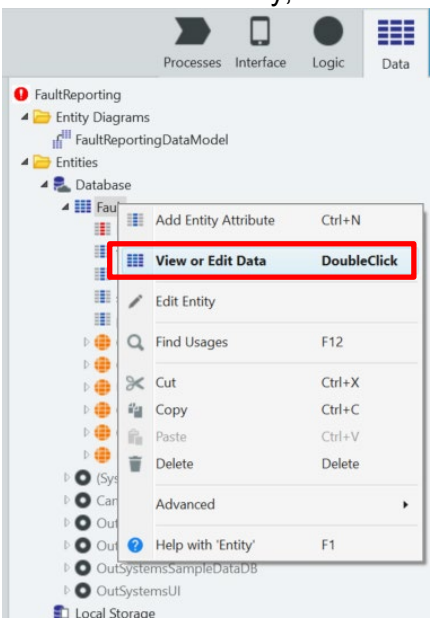
15. Click on the Save button. You will be able to see that an entry is created in the Faults screen. However, you will notice that the image is not shown in the Faults screen.



Exercise 3: Updating the Faults screen to show the Image

---

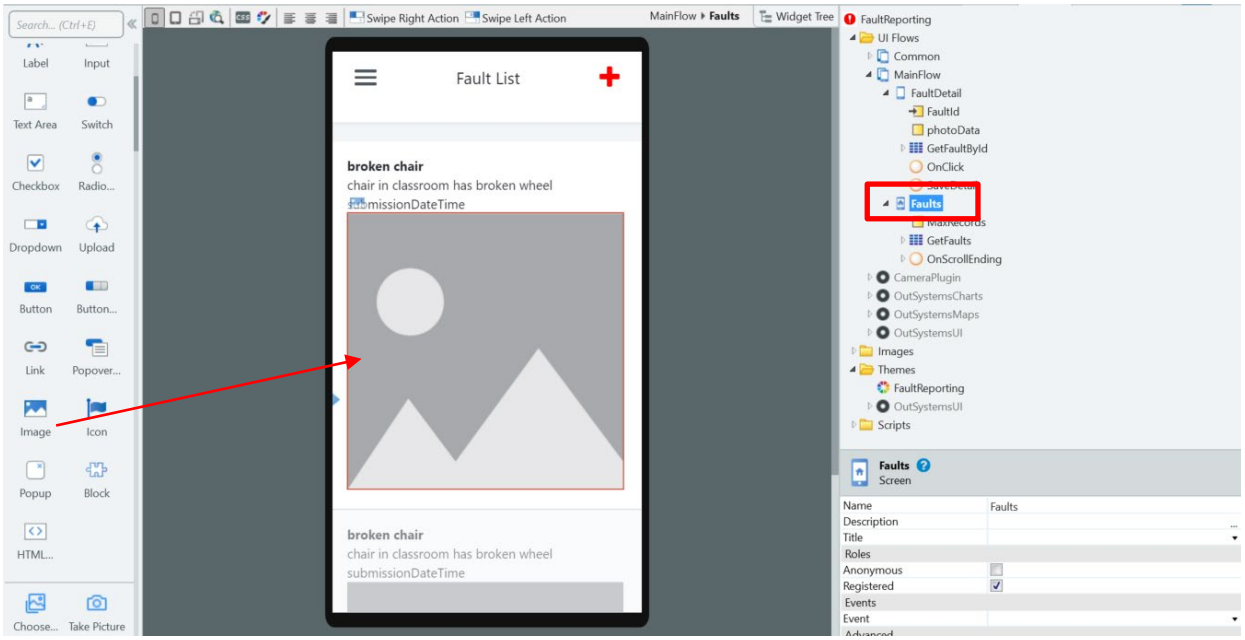
1. Click on the Data tab, right click on the Fault entity, select View or Edit Data.



2. You will be able to see that there is an additional entry with the information that you have submitted in exercise 2. The submissionDateTime and the photoData are filled up with the data submitted.

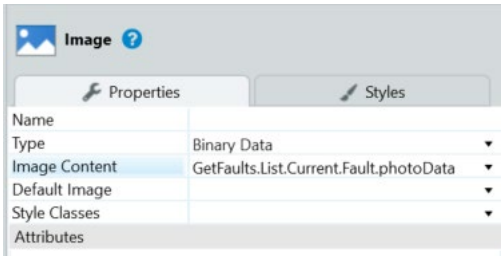
Fault Id	Fault title	Fault description	Fault submissionDateTime	Fault photoData
4	faulty door handle	faulty door handle at W6 level 8	2021-12-14 00:56:20	BinaryData (26282 bytes)
3	door	door handle detached	2021-12-13 14:16:35	BinaryData (0 bytes)
2	lighting problem	light is flickering in E6-level 7	1900-01-01 00:00:00	BinaryData (0 bytes)
1	broken chair	chair in classroom has broken wheel	1900-01-01 00:00:00	BinaryData (0 bytes)

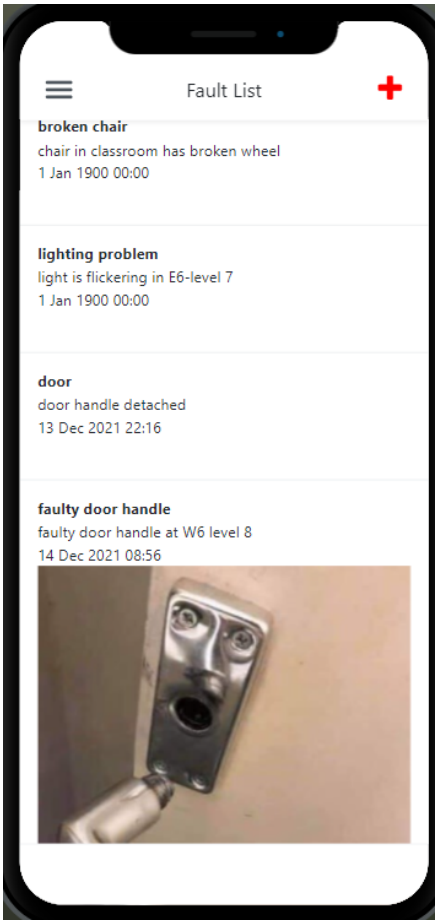
3. Click on the Interface tab, double-click on the **Faults** screen. Add an **Image** widget just below the submissionDateTime expression.



4. Change the following properties for the selected Image.

Type	Binary Data
Image Content	GetFaults.List.Current.Fault.photoData

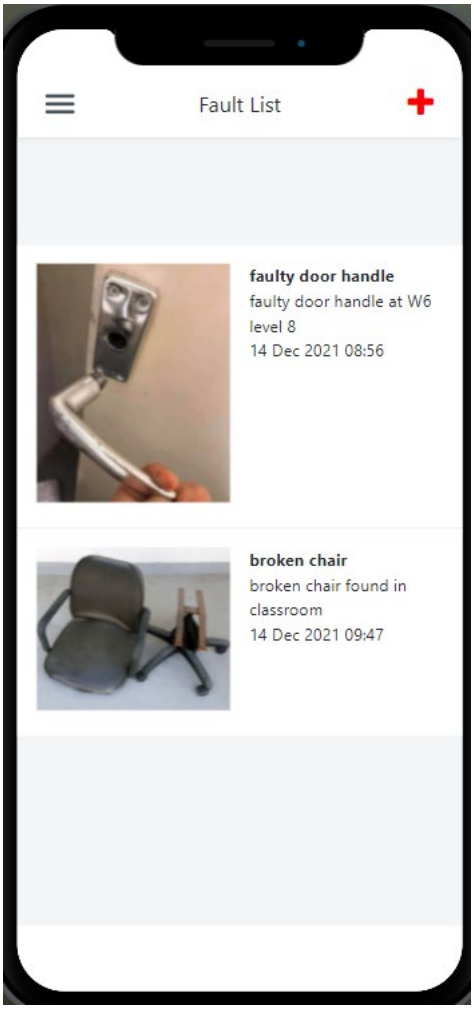




Exercise 4: Additional Challenge

The following tasks will be left as an exercise for you.

1. Label the values and re-arrange the information in the **Faults** screen. The screenshot below shows you one example of how you may want to layout the information.



2. Currently, there is no login feature added to the app. The app will show the submission of every user. Add a UserId attribute to the Fault entity. Make the necessary change so that the user will only be able to see their own submission.

End of Lab 9