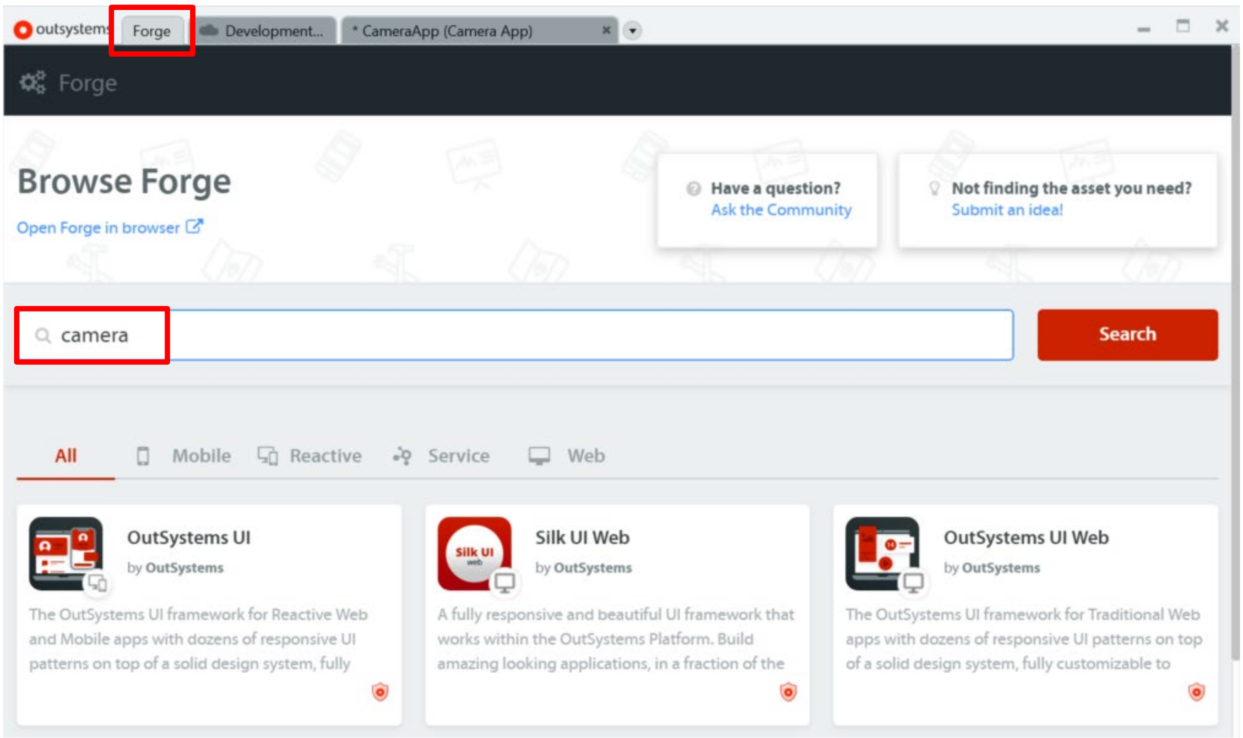
Outsystems Camera Plugins Lab Guide

Exercise 1: Camera App

In this exercise, you will make use of the Camera Plugin to allow us to use the phone camera to capture image.

1. Create a new mobile application named **Camera App**, with a **Phone App** Module
    a. In the 'Applications in Development' area, click **New Application**.
    b. In the **New Application** dialog, select **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 ***Camera 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 **CameraApp** and select the **Phone App** module type. Click **Create Module** to create the module.

2. Click on the **Forge** tab as shown. Type Camera in the "Search in Forge..." input box, and click on Search button.

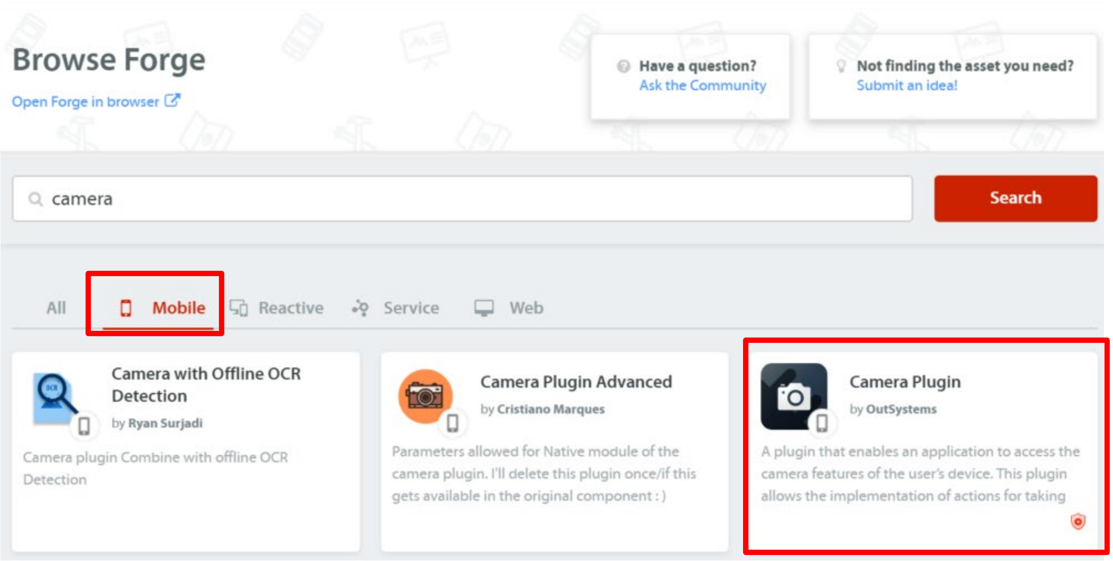


OutSystems UI
by OutSystems
The OutSystems UI framework for Reactive Web and Mobile apps with dozens of responsive UI patterns on top of a solid design system, fully

Silk UI Web
by OutSystems
A fully responsive and beautiful UI framework that works within the OutSystems Platform. Build amazing looking applications, in a fraction of the

OutSystems UI Web
by OutSystems
The OutSystems UI framework for Traditional Web apps with dozens of responsive UI patterns on top of a solid design system, fully customizable to

3. Click on the Mobile tab to select only plugins that are applicable for mobile apps. Click on Camera Plugin to select the plugin.

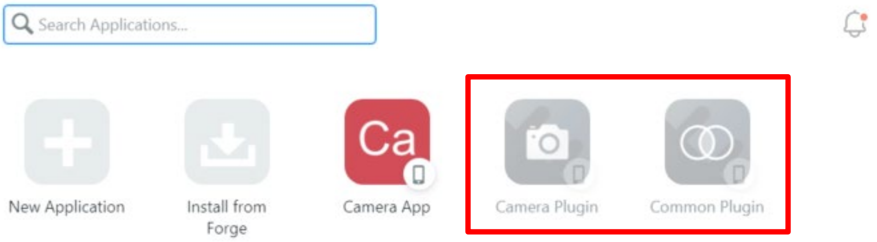


Camera with Offline OCR Detection
by Ryan Surjadi
Camera plugin Combine with offline OCR Detection

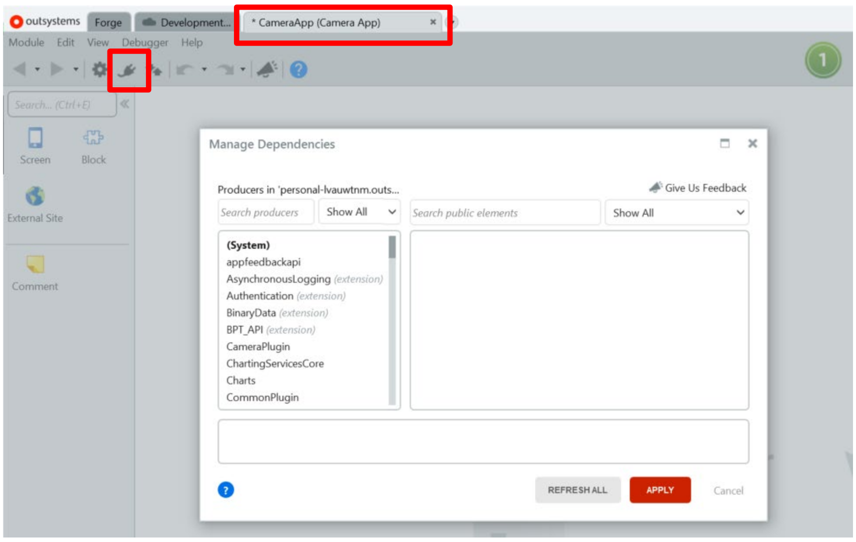
Camera Plugin Advanced
by Cristiano Marques
Parameters allowed for Native module of the camera plugin. I'll delete this plugin once/if this gets available in the original component :)

Camera Plugin
by OutSystems
A plugin that enables an application to access the camera features of the user's device. This plugin allows the implementation of actions for taking

4. Click on "**Install…**" to install the Camera plugin. Note: You may need to click on Install twice. You will notice that both **Camera Pugin** and **Common Plugin** will be added to the development workspace.



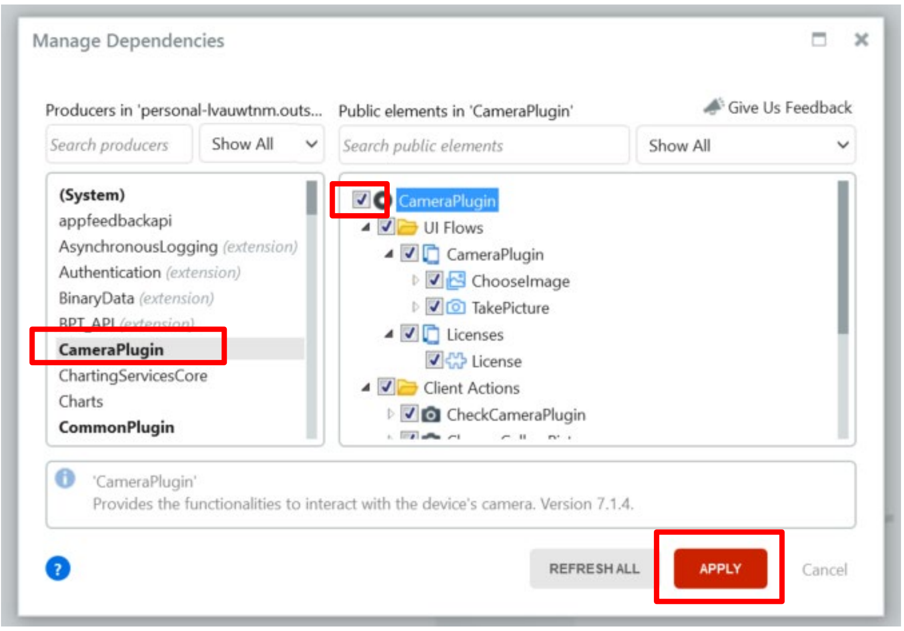
5. Switch back to the CameraApp, and click on the Manage Dependency icon.



Producers in 'personal-lvauwtnm.outs...

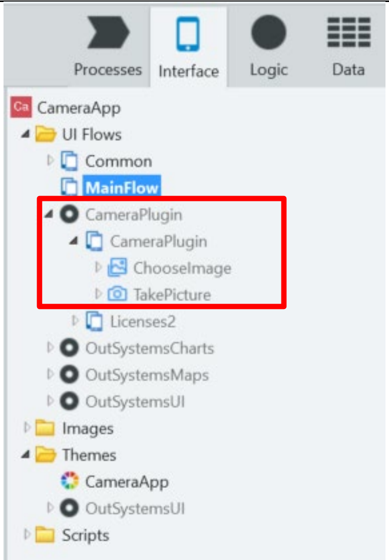
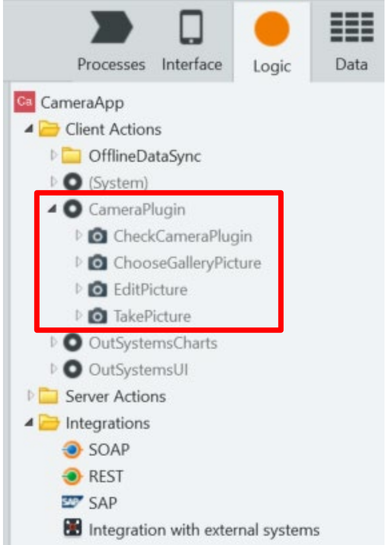
(System)
appfeedbackapi
AsynchronousLogging (extension)
Authentication (extension)
BinaryData (extension)
BPT_API (extension)
CameraPlugin
ChartingServicesCore
Charts
CommonPlugin

6. Select **Camera Plugin** from the left panel, click on the checkbox to select all items for the camera plugin. Click on **Apply** button.

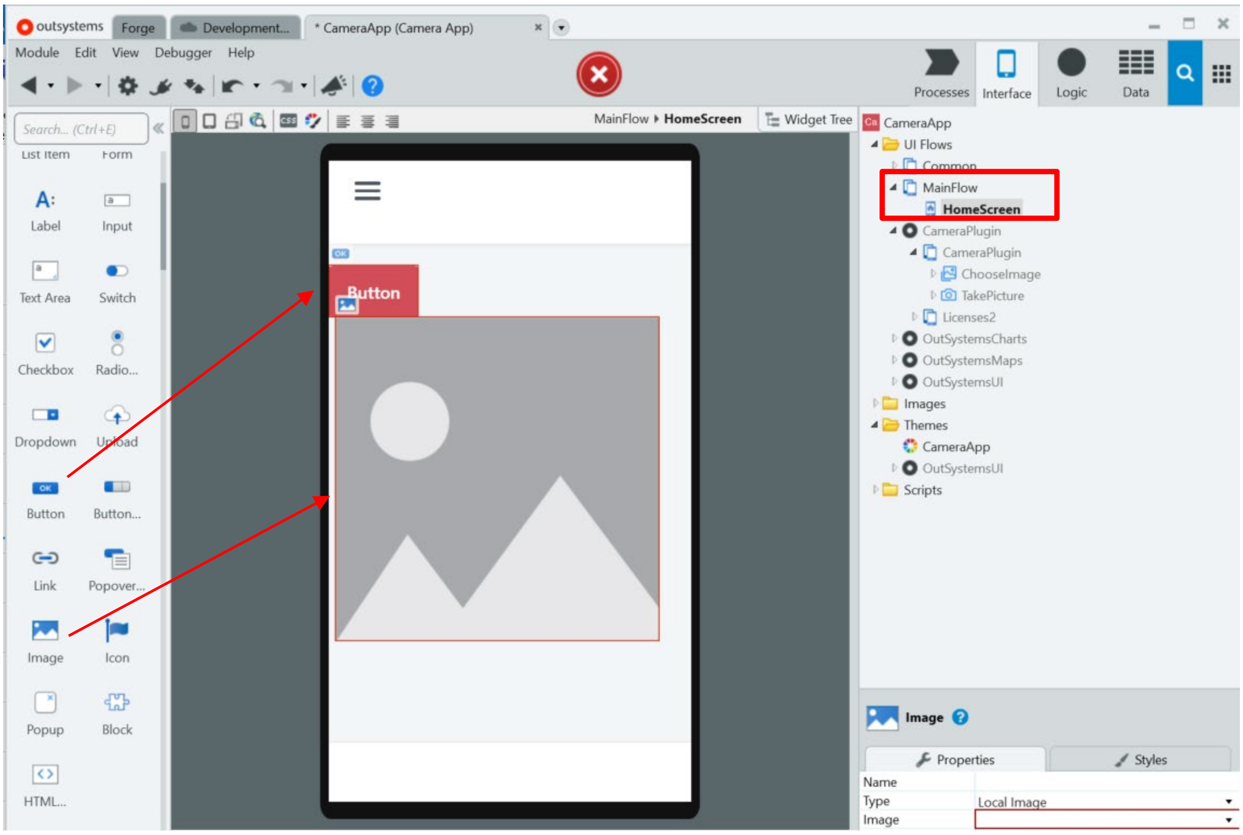


'CameraPlugin'
Provides the functionalities to interact with the device's camera. Version 7.1.4.

7. You will notice some additional items created in both the **Interface** and the **Logic** tab after you have added the Camera Plugin dependency to your project. You will see how these items are used later on to enable the use of camera in your app.

	
Interface Tab	Logic Tab

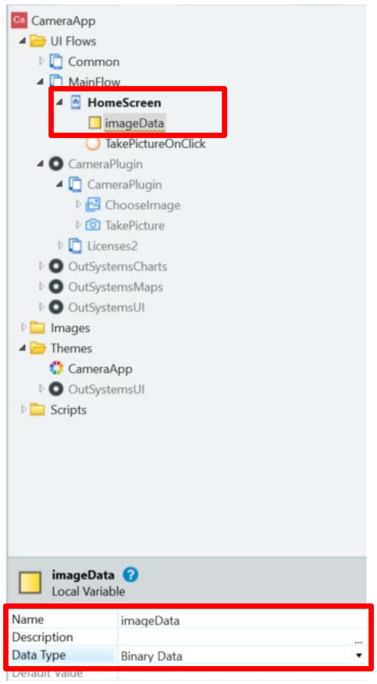
8. Click on the Interface tab, add a new Screen to the MainFlow, name the new screen as **HomeScreen**. Add a Button and an Image to the HomeScreen as shown. Ignore the Errors for now.



Name	
Type	Local Image
Image	

9. Select the Button:
    a. Change its Text to “TAKE PICTURE”.
    b. Add a new **Client Action** for the On Click event. The new Client Action will be named as **TakePictureOnClick** by default.

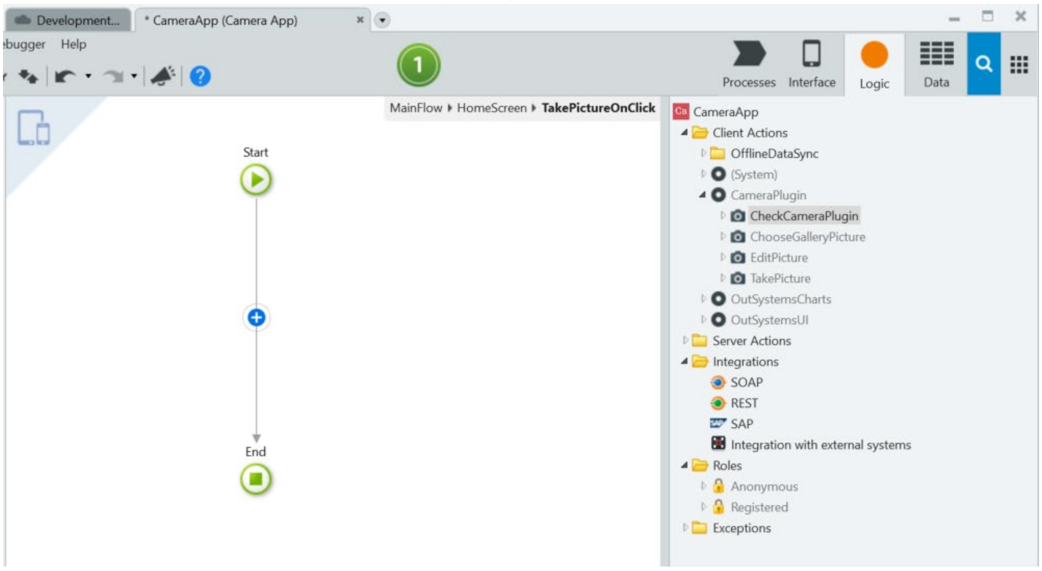
10. Create a new Local Variable with the following properties.
    a. Name: imageData
    b. Data Type: Binary Data



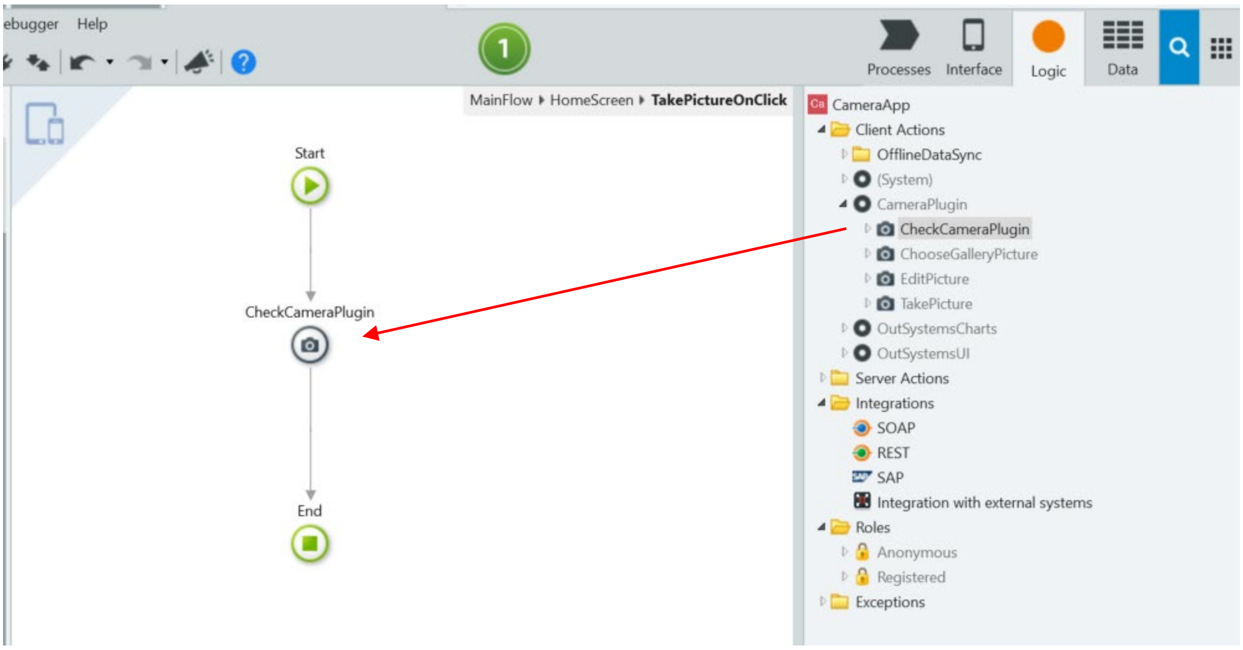
Name	imageData
Description	
Data Type	Binary Data
Default Value	

11. Click on the Image in the HomeScreen. Change the following properties for the Image.
    a. Type: Change from Local Image to **Binary Data** from the Dropdown list**.**
    b. Image: Select **User** from the Dropdown List.

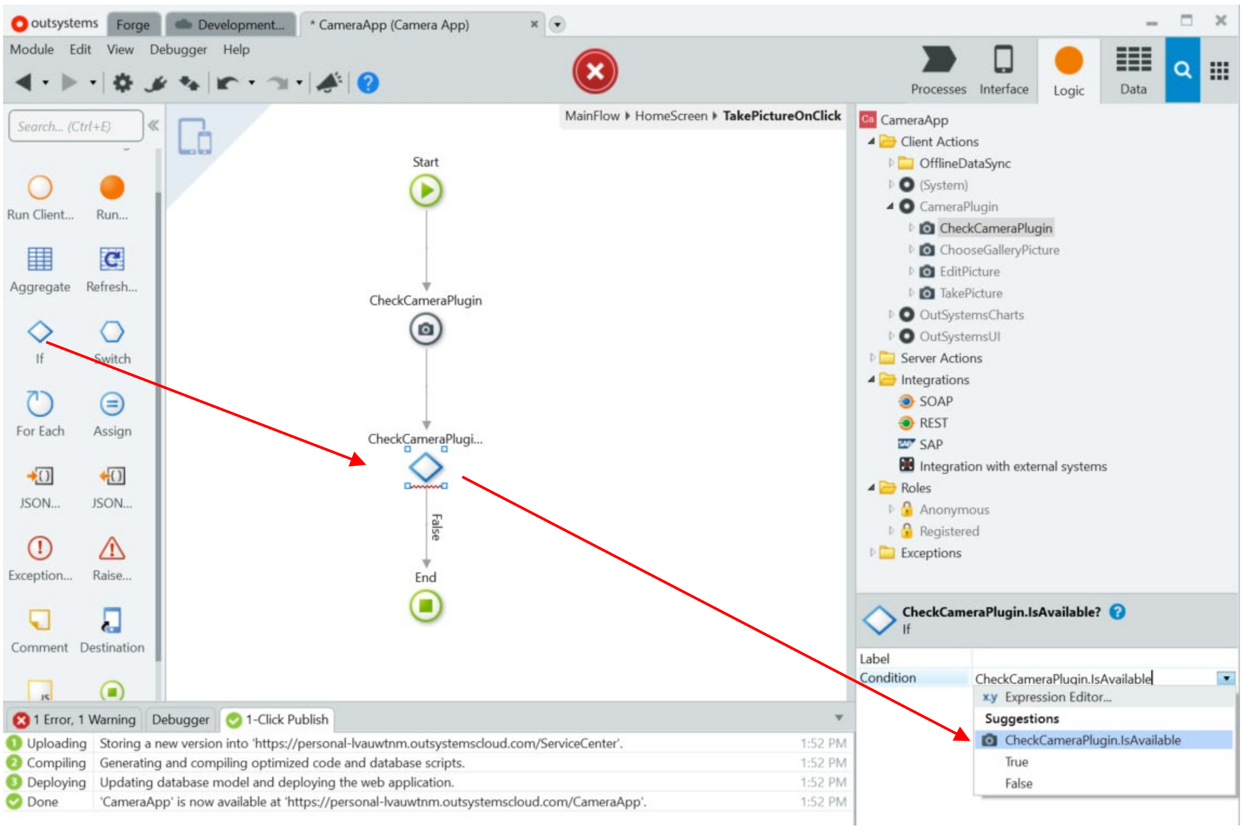
12. Click on **TakePictureOnClick** action to open up the workflow view.



13. Click on the Logic tab, drag the **CheckCameraPlugin** action to the workflow as shown.



Condition: CheckCameraPlugin.IsAvailable



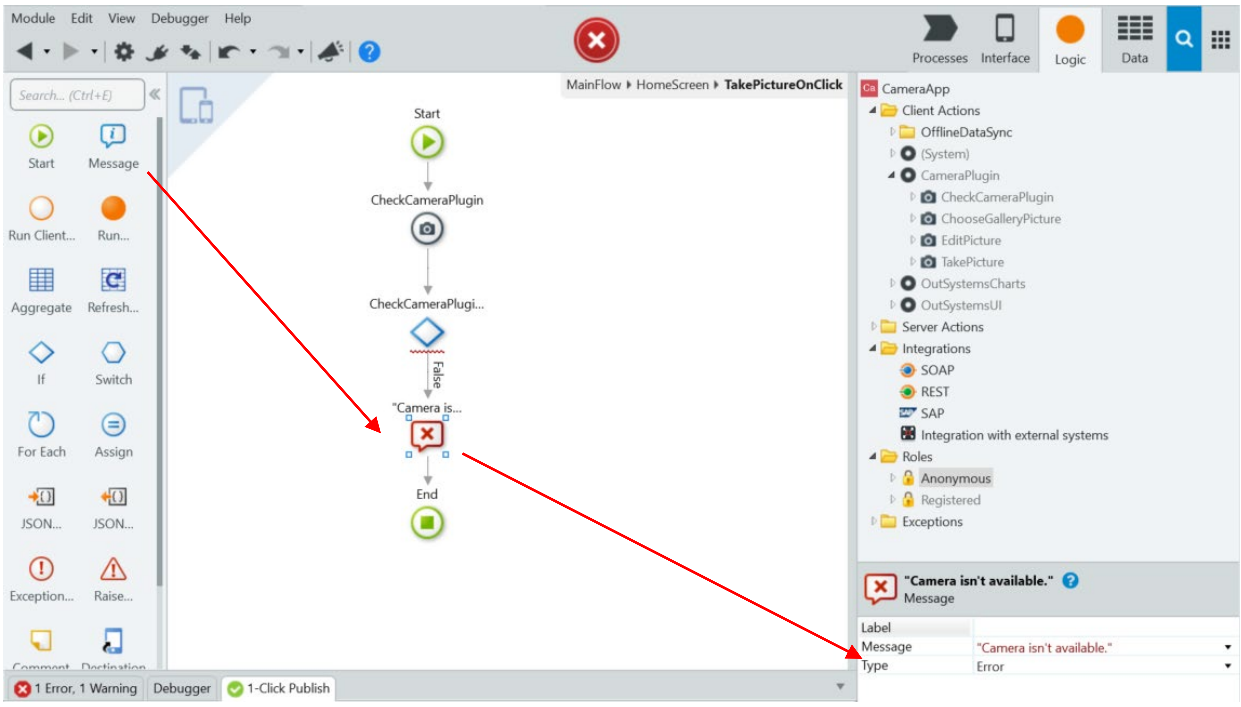
CheckCameraPlugin.IsAvailable? If

Label
Condition CheckCameraPlugin.IsAvailable

Expression Editor...
Suggestions
CheckCameraPlugin.IsAvailable
True
False

```
Uploading   Storing a new version into 'https://personal-lvauwtnm.outsystemscloud.com/ServiceCenter'.   1:52 PM
Compiling   Generating and compiling optimized code and database scripts.                              1:52 PM
Deploying   Updating database model and deploying the web application.                                 1:52 PM
Done        'CameraApp' is now available at 'https://personal-lvauwtnm.outsystemscloud.com/CameraApp'.   1:52 PM
```

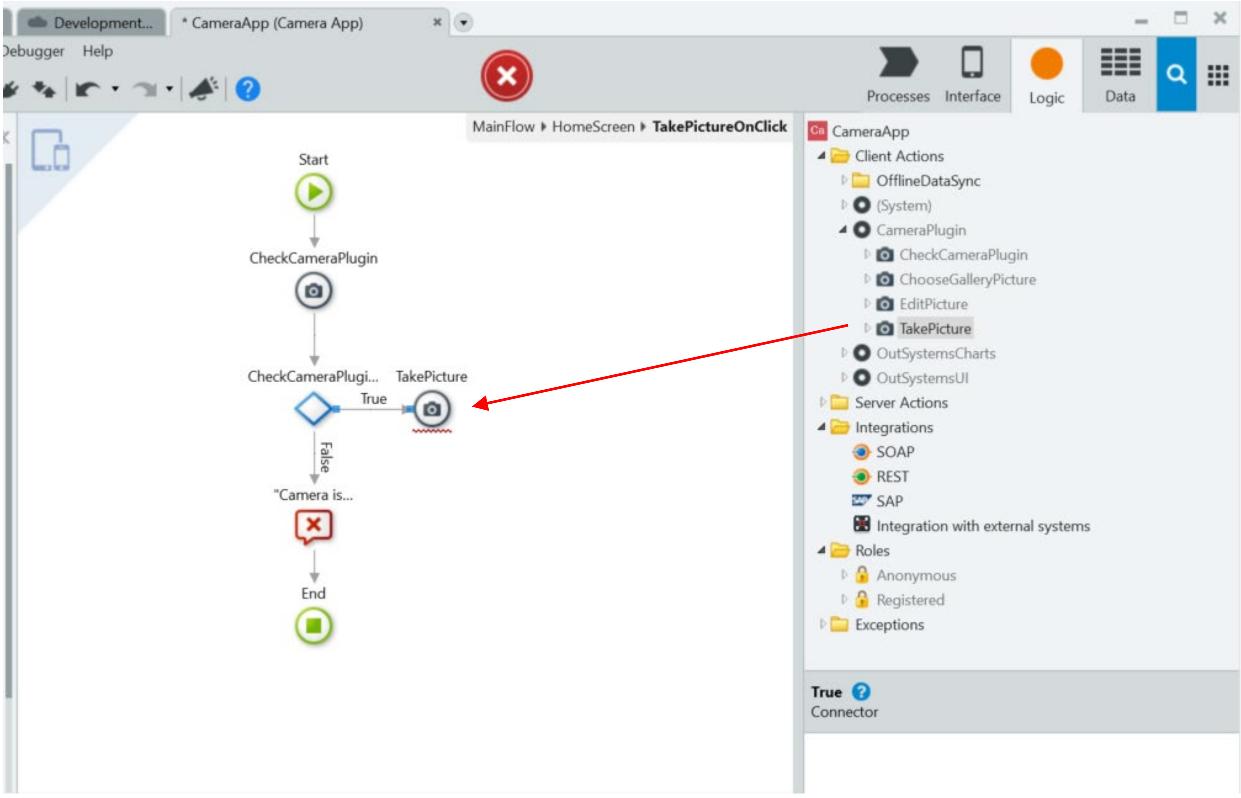
15. Drag a message action to the workflow, and change the properties to the following:
    a. Message: **"Camera isn't available."**.
    b. Type: **Error**



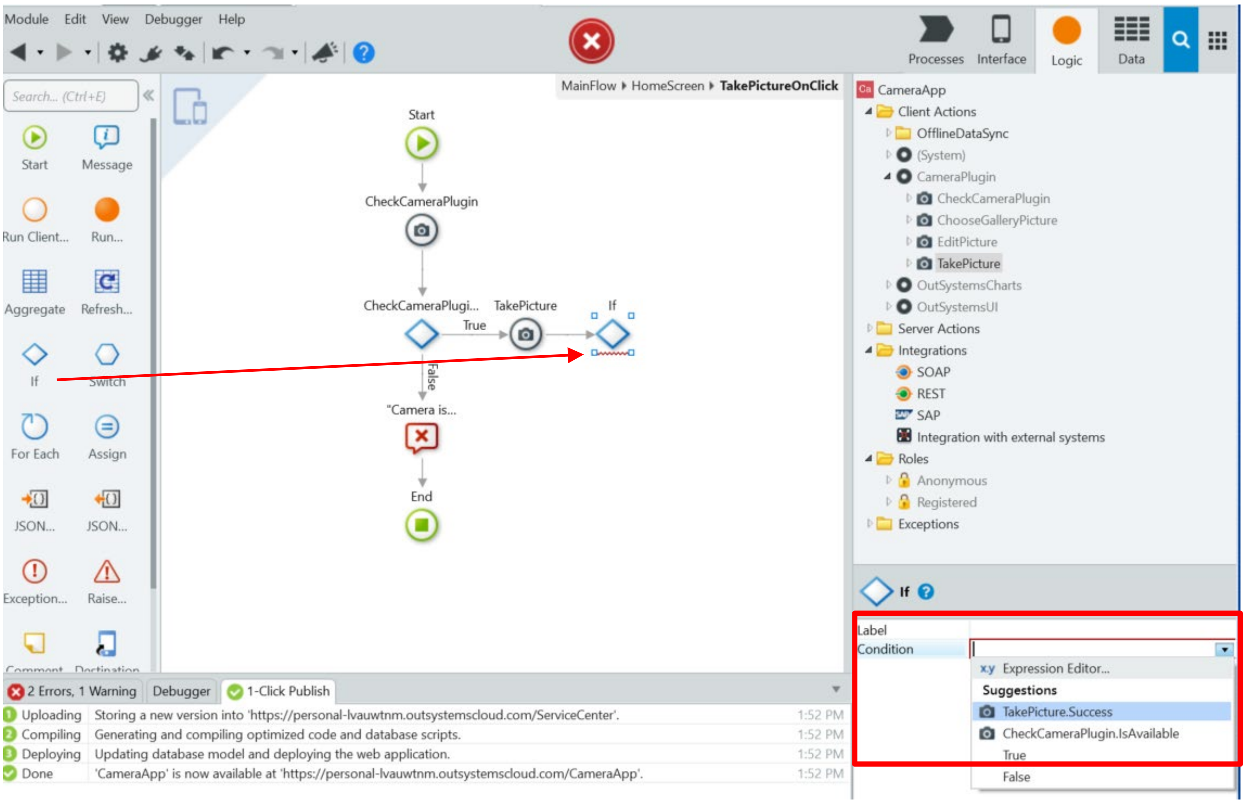
"Camera isn't available." Message

Label
Message "Camera isn't available."
Type Error

16. Drag the **TakePicture** action and place it next to the **CheckCameraPlugin if** condition. Join the **TakePicture** action with the **CheckCameraPlugin if** condition.

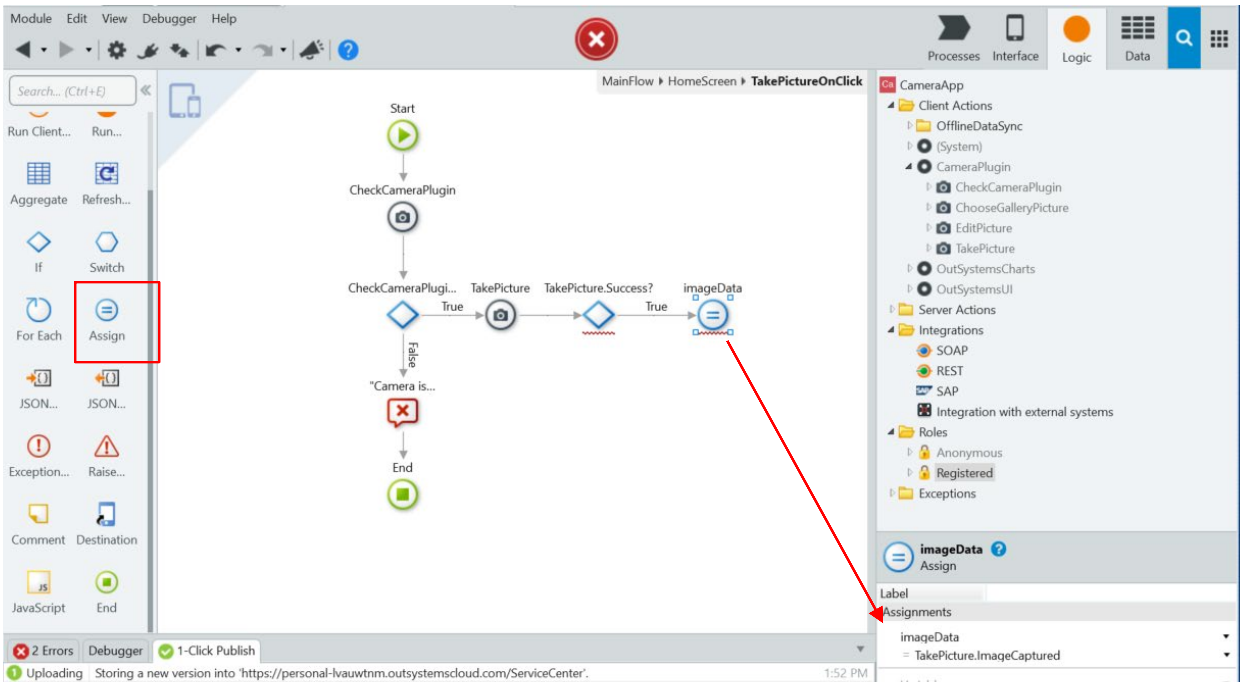


17. Drag the if condition and place it next to the TakePicture action. Set the condition to **TakePicture.Success**.



```
Uploading   Storing a new version into 'https://personal-lvauwtnm.outsystemscloud.com/ServiceCenter'.   1:52 PM
Compiling   Generating and compiling optimized code and database scripts.                               1:52 PM
Deploying   Updating database model and deploying the web application.                                  1:52 PM
Done        'CameraApp' is now available at 'https://personal-lvauwtnm.outsystemscloud.com/CameraApp'.  1:52 PM
```

18. Drag the Assign action and place it next to the TakePicture.Success if condition. Set the following properties for the Assign action.
    a. Variable: imageData
    b. Value: TakePicture.ImageCaptured



Label

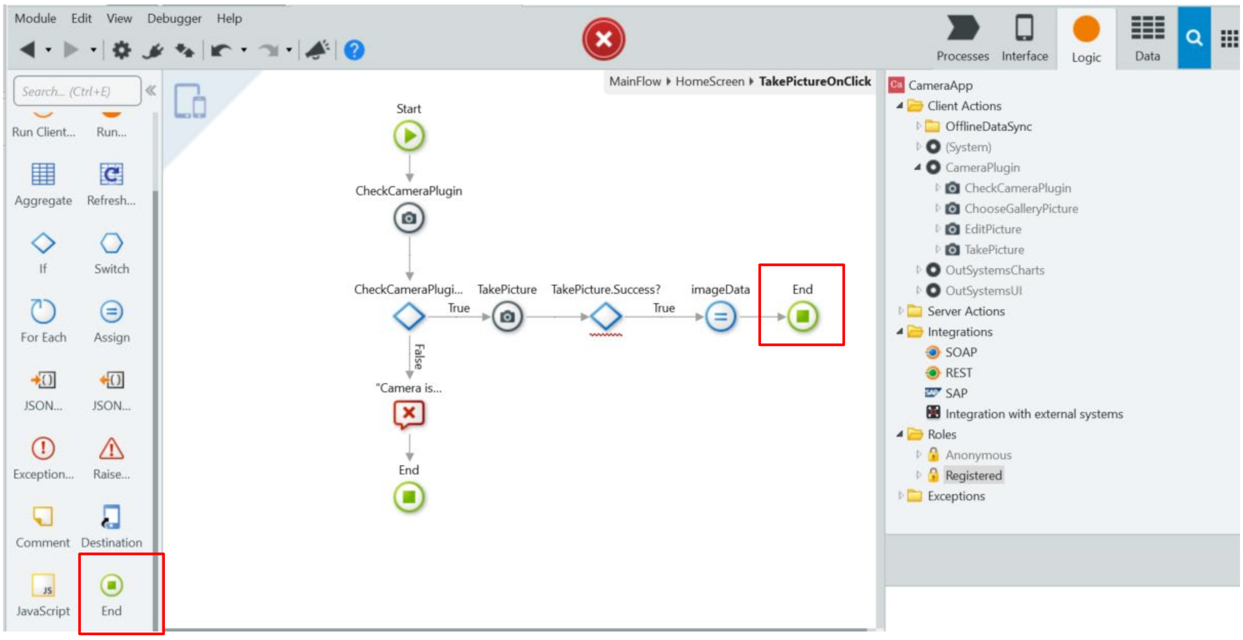
Assignments

imageData

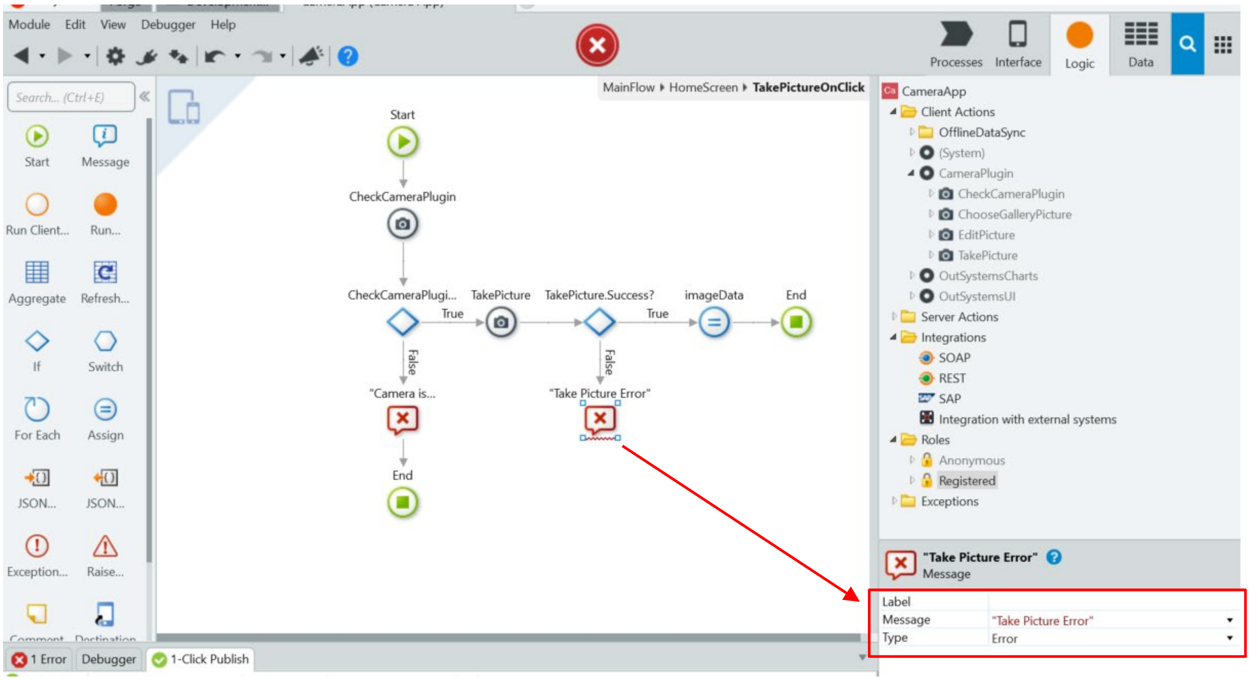
= TakePicture.ImageCaptured

Uploading Storing a new version into 'https://personal-lvauwtnm.outsystemscloud.com/ServiceCenter'. 1:52 PM

19. Drag the End action, and place it next to the Assign action as shown.



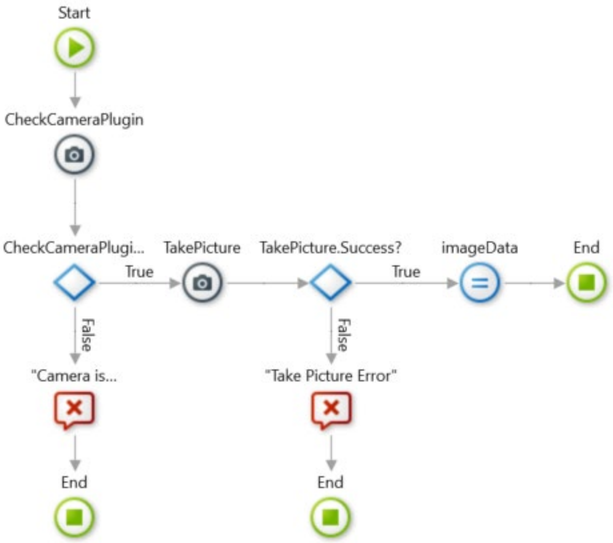
20. Drag a message action to the workflow and place it below the **TakePicture.Success if** condition, and change the properties to the following:
    a. Message: **"Take Picture Error!"**.
    b. Type: **Error**



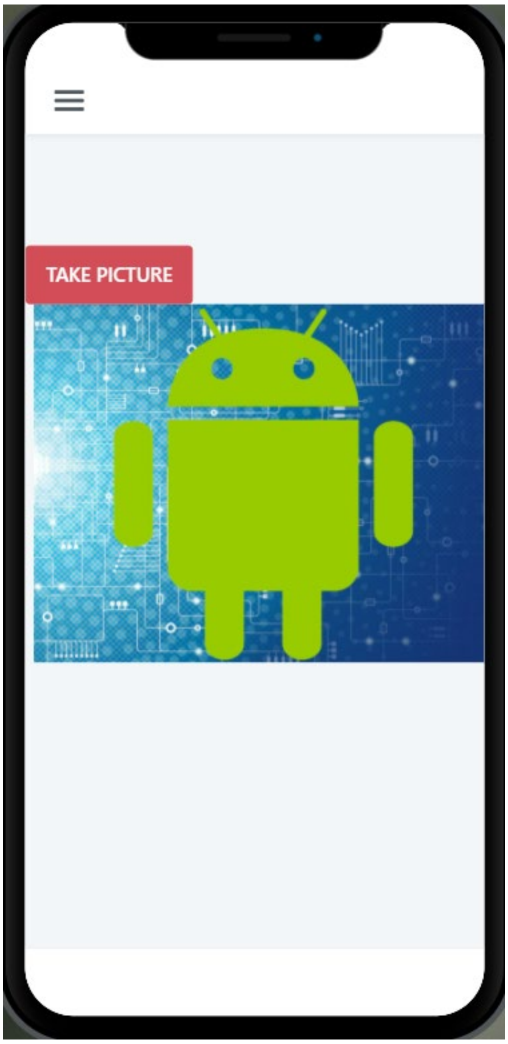
"Take Picture Error"
Message

Label	
Message	"Take Picture Error"
Type	Error

21. Add another End action just below the Message action. This will fix all error and complete the workflow. The final workflow is as shown below.



22. You may click on the 1-Click Publish button to update and test the app. You will be able to choose an image from your laptop, and see the chosen image as shown below. Alternatively, you may also test the app as a PWA , and use your phone to do the testing. You will be able to use your phone camera to take a picture in this case.



End of Lab 6