# Outsystems Plugins Lab Guide

Exercise 1: Barcode App

In this exercise, you will make use of the Barcode Plugin to allow us to scan barcode to retrieve the url of the barcode.

1. Create a new mobile application named **Barcode 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 *Barcode 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 **BarcodeApp** and select the **Phone App** module type. Click **Create Module** to create the module.

2. Click on the **Forge** tab as shown. Type barcode in the "Search in Forge…" input box, and click on Search button.
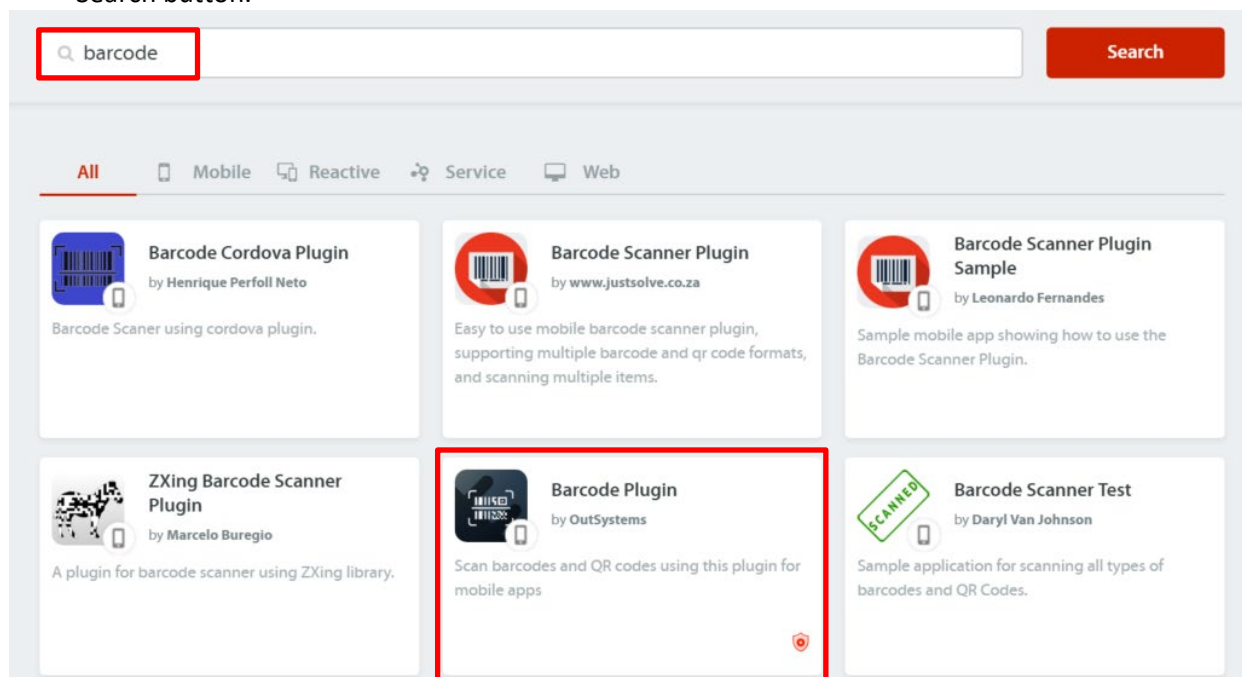
3. Click on "**Install…**" to install the Barcode plugin. Note: You may need to click on Install twice. You will notice that both **Barcode Pugin** will be added to the development workspace.
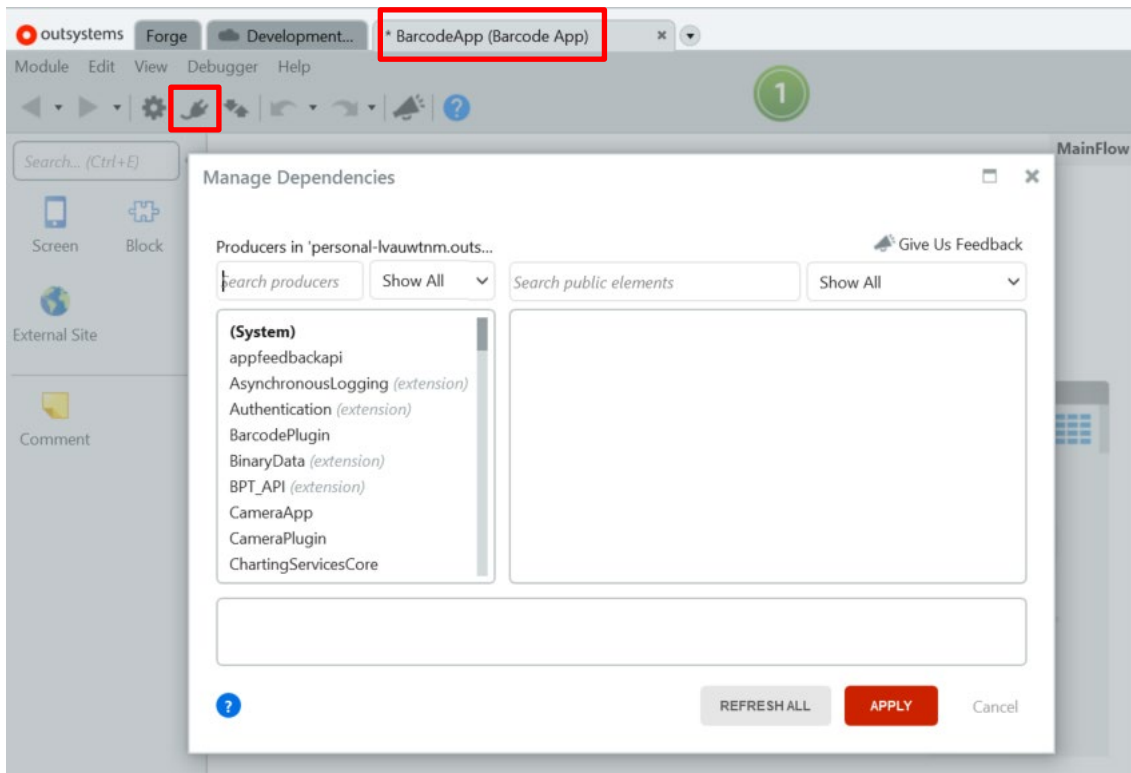Note: Common Plugin was already installed as part of Camera Plugin.
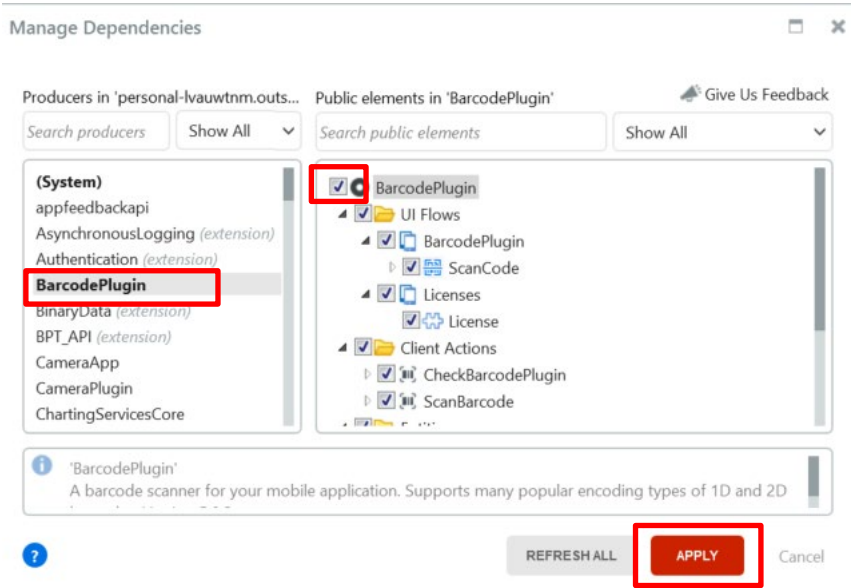

4. Switch back to the BarcodeApp, and click on the Manage Dependency icon.

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


6. 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
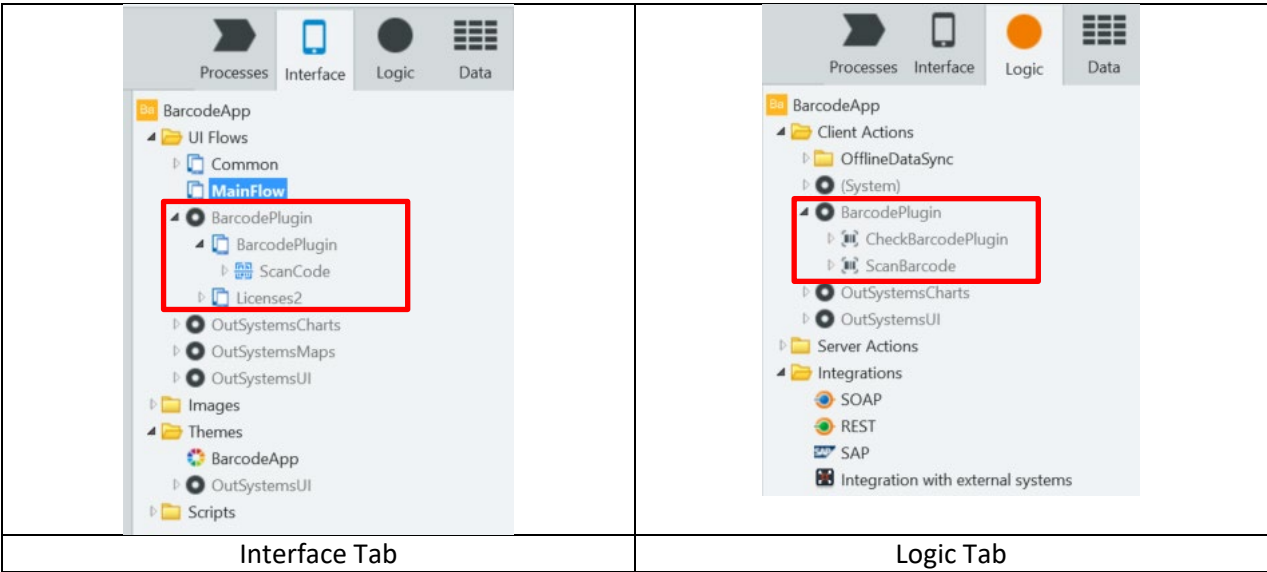
7. Click on the Interface tab, add a new Screen to the MainFlow, name the new screen as
    HomeScreen. Add a Button and an Expression to the HomeScreen as shown. Ignore the Errors for
   now.
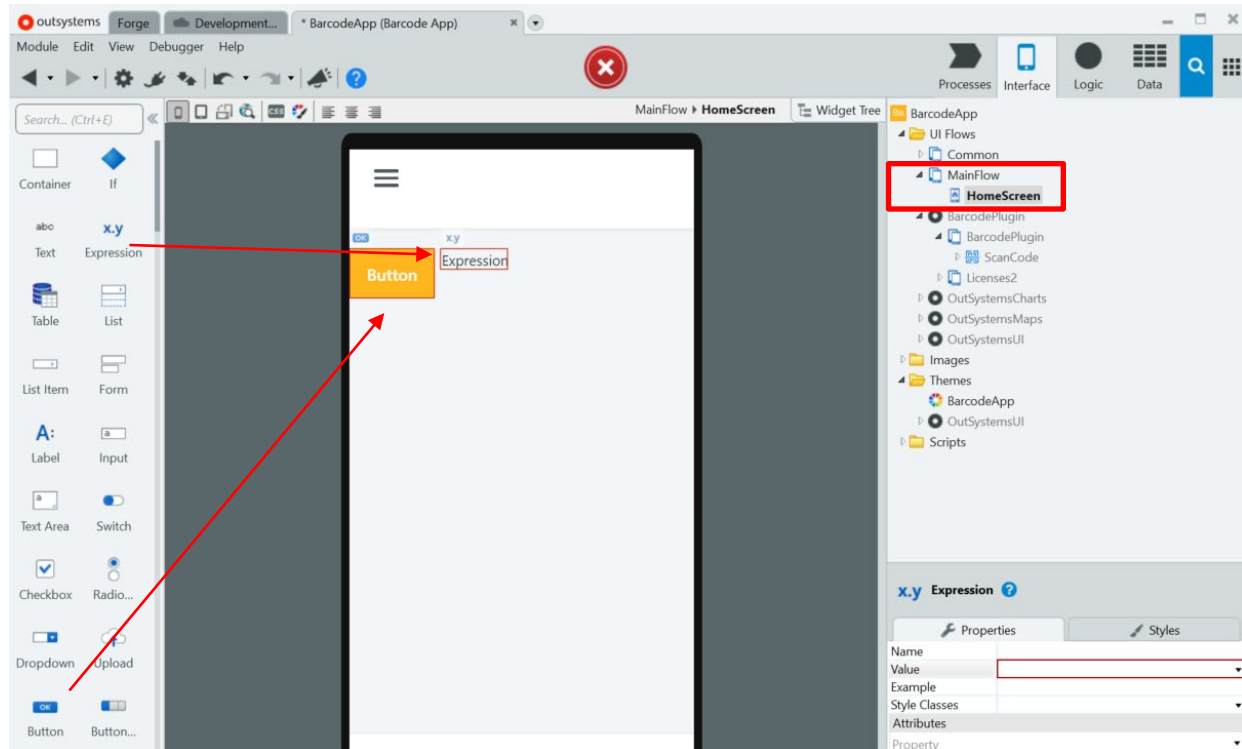
8. Select the Button:
    a. Change its Text to "SCAN QR Code".
    b. Add a new **Client Action** for the On Click event. The new Client Action will be named as
        ScanQrCodeOnClick by default.

9. Create a new Local Variable with the following properties.
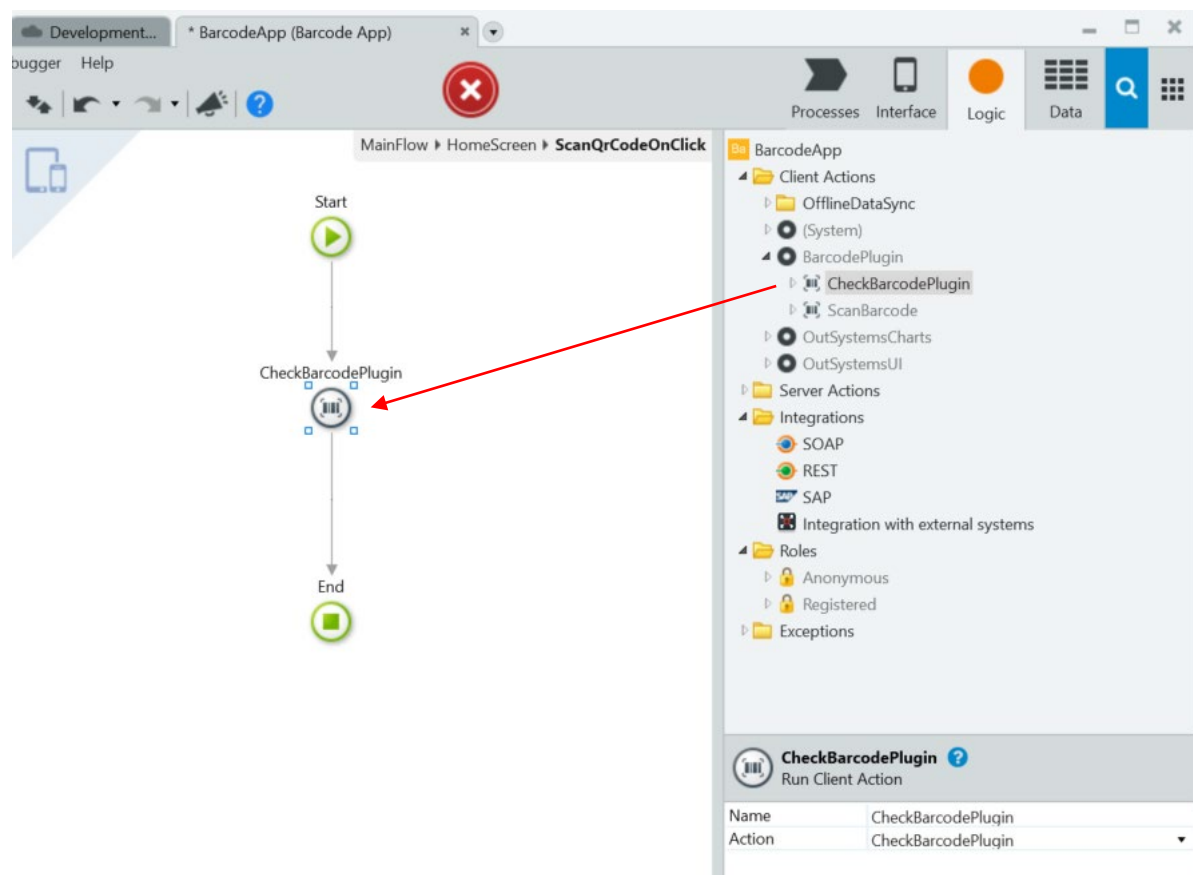   a. Name: scanText
    b. Data Type: Text
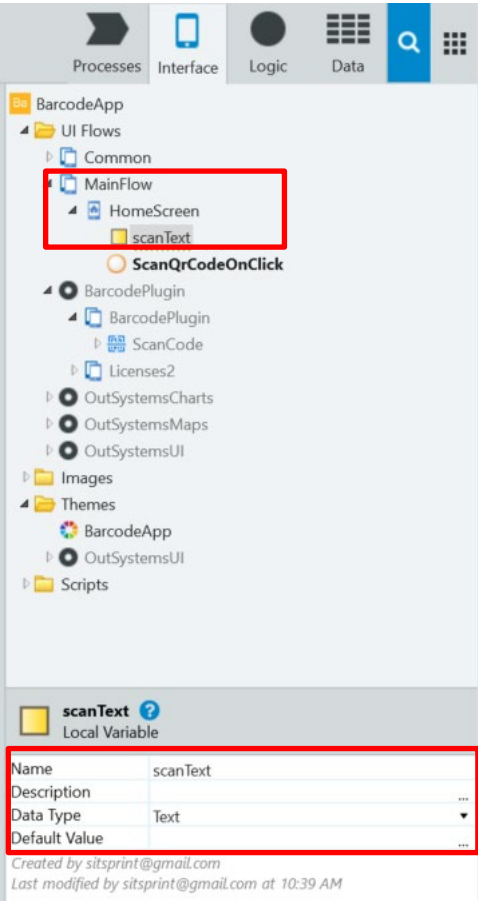
10. Click on the **Expression** in the HomeScreen, set its value to the **scanText** Local Variable.
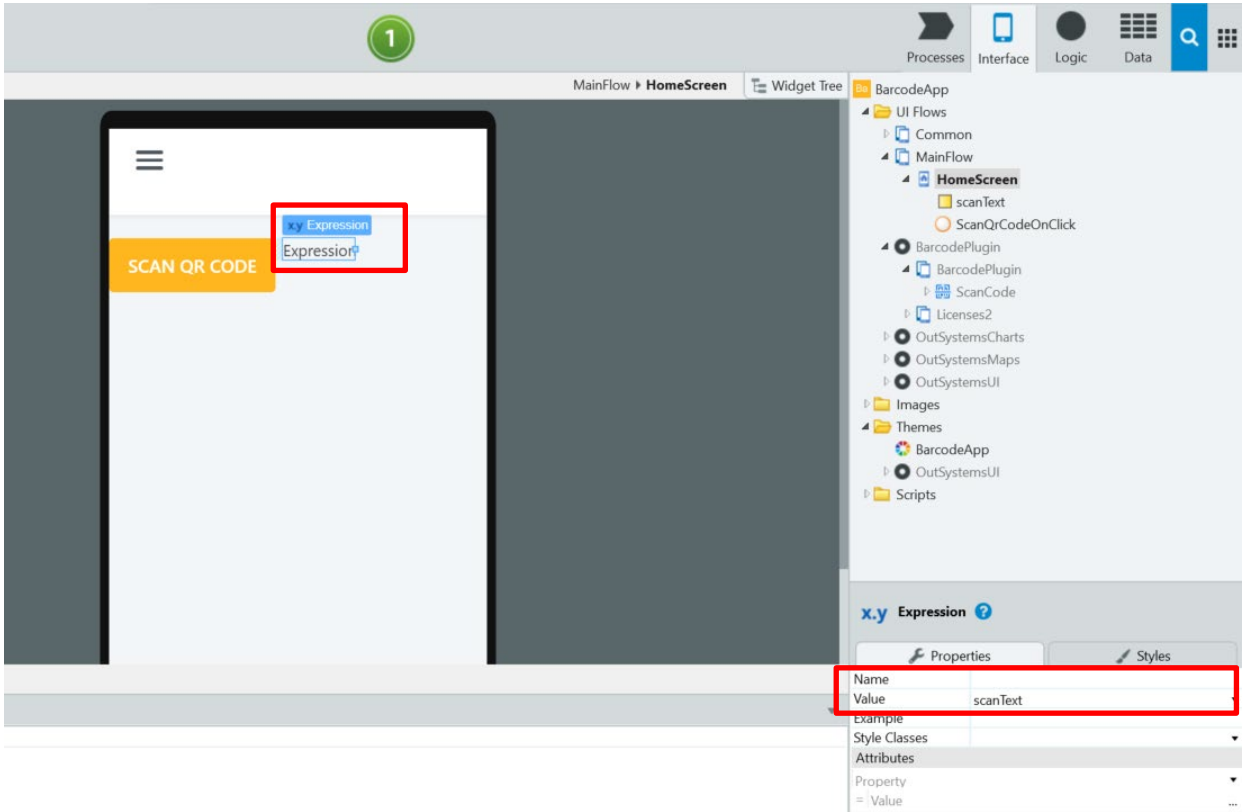
11. Click on **ScanQrCodeOnClick** action to open up the workflow view.
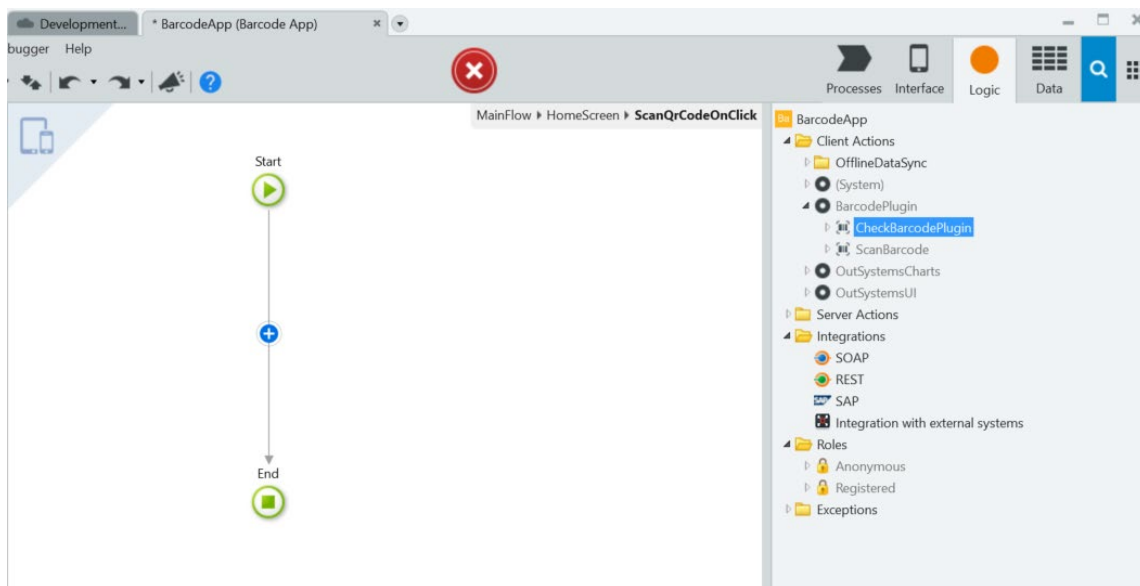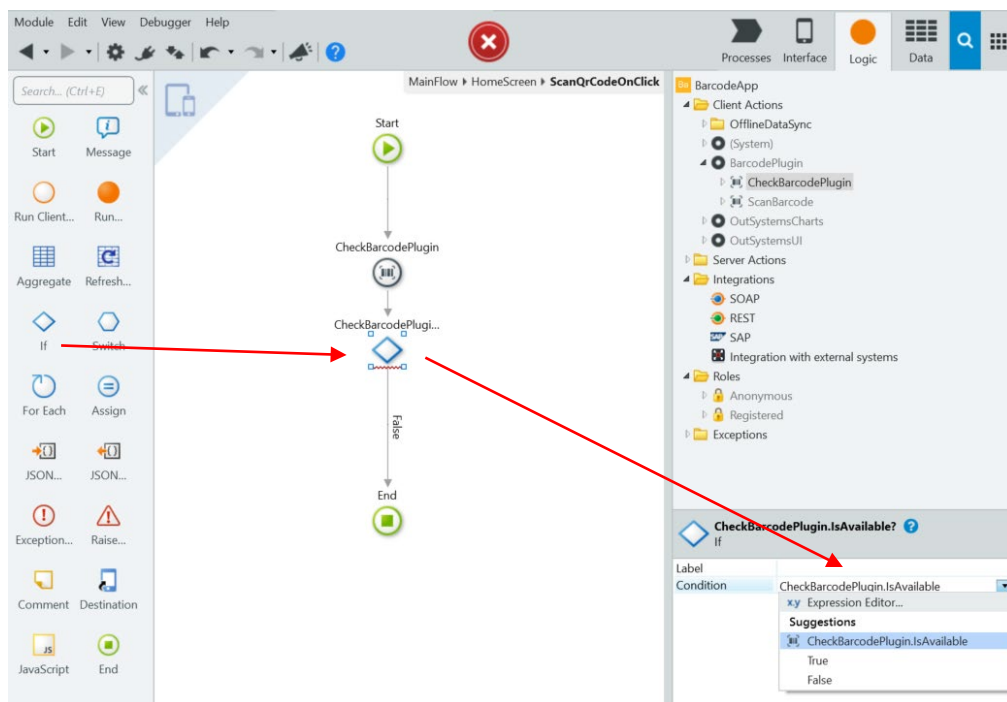
12. Click on the Logic tab, drag the **CheckBarcodePlugin** action to the workflow as shown.
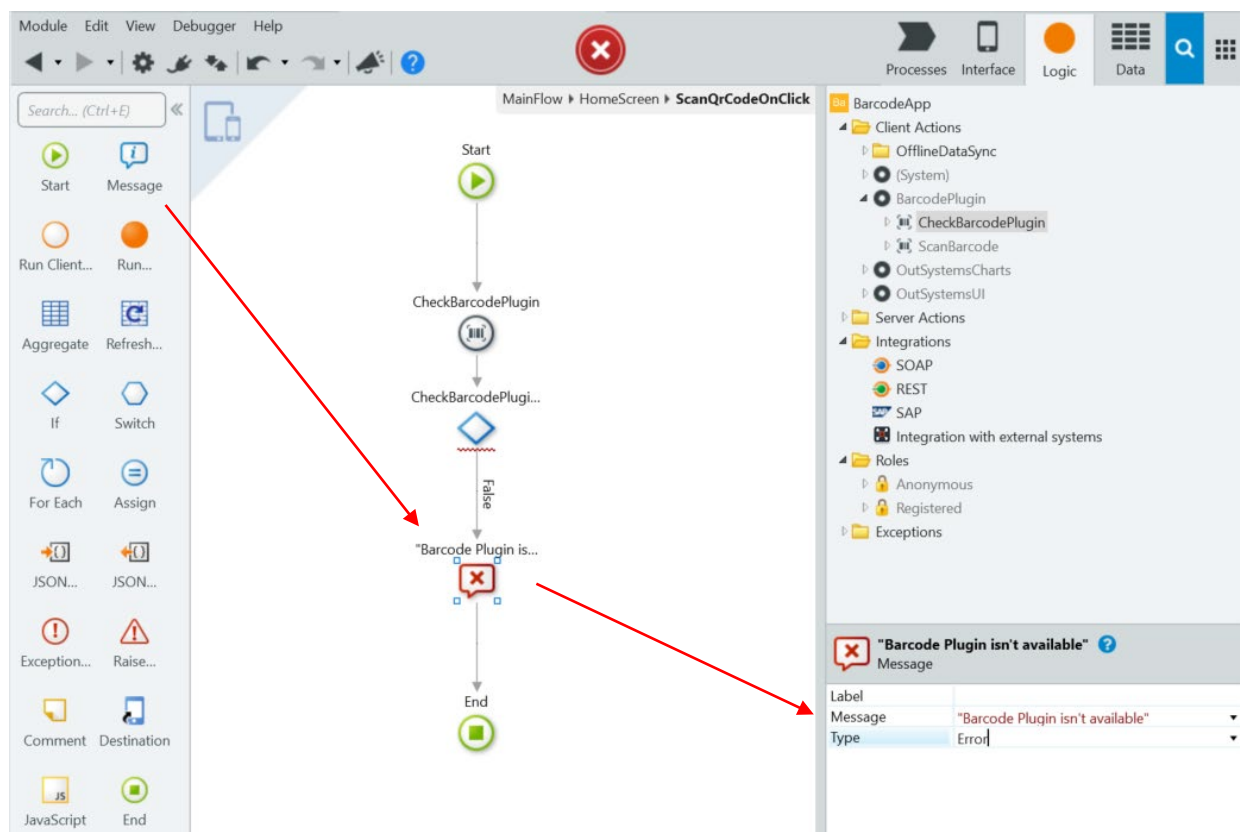

13. Drag the if action to the workflow, and define the condition as shown below:
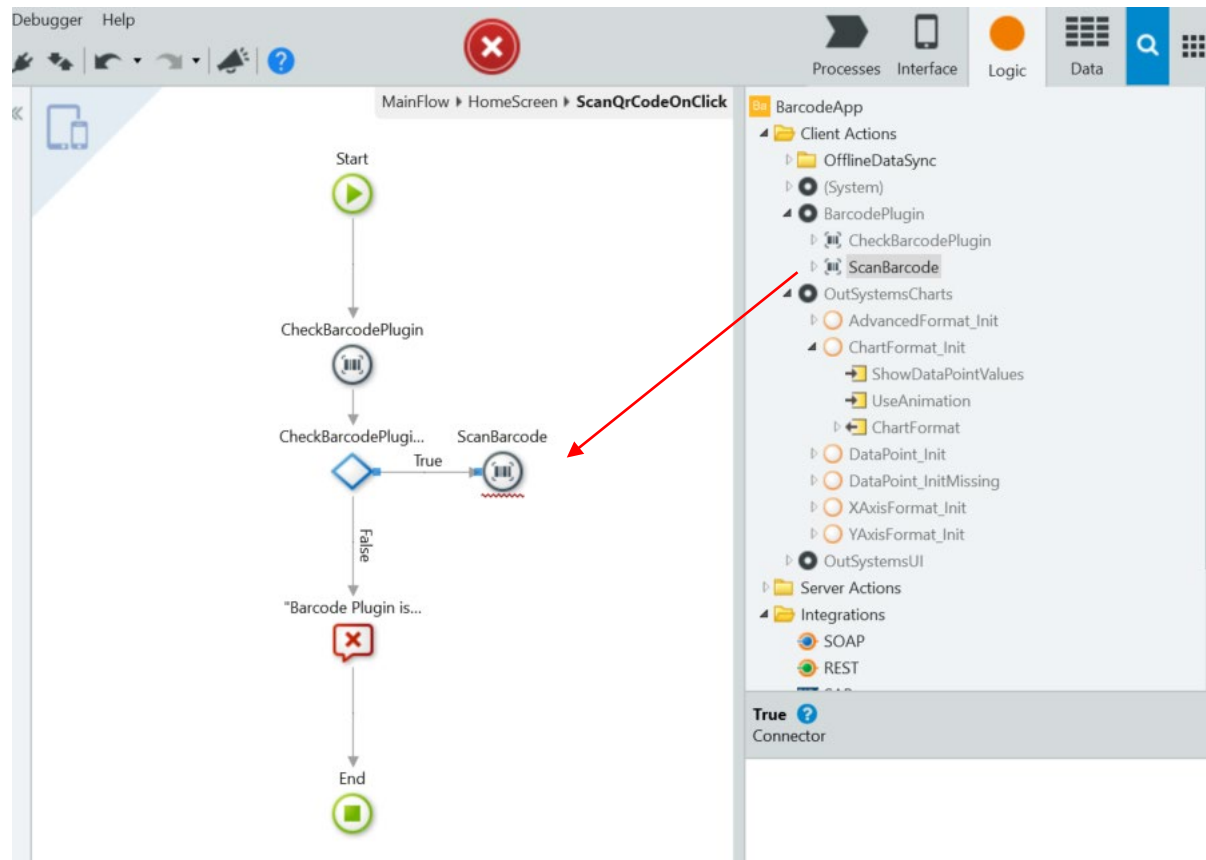
Condition: CheckBarcodePlugin.IsAvailable

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

15. Drag the **ScanBarcode** action and place it next to the **CheckBarcodePlugin if** condition. Join the **ScanBarcode** action with the **CheckBarcodePlugin if** condition.

16. Drag the Assign action and place it next to the ScanBarcode action. Set the following properties for the Assign action.
    a. Variable: scanText
    b. Value: ScanBarcode.ScanResult
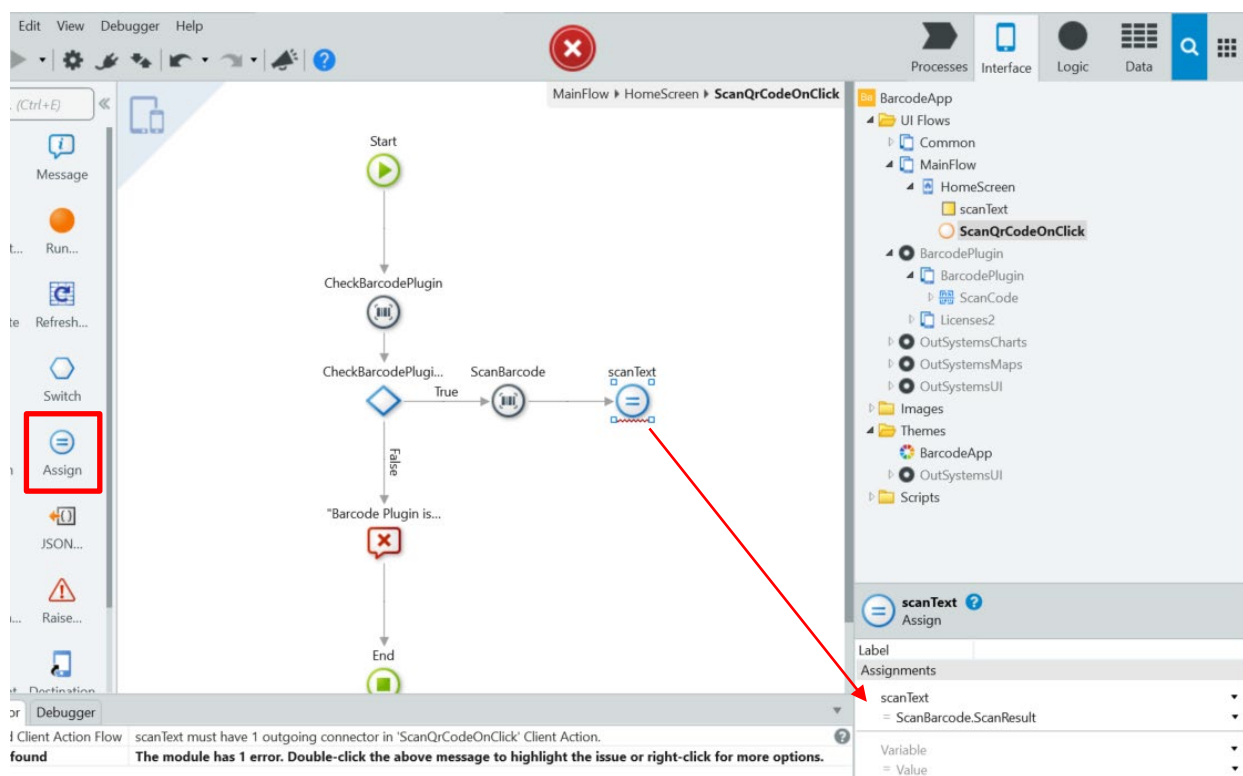
17. Drag the End action, and place it next to the ScanBarcode action as shown.
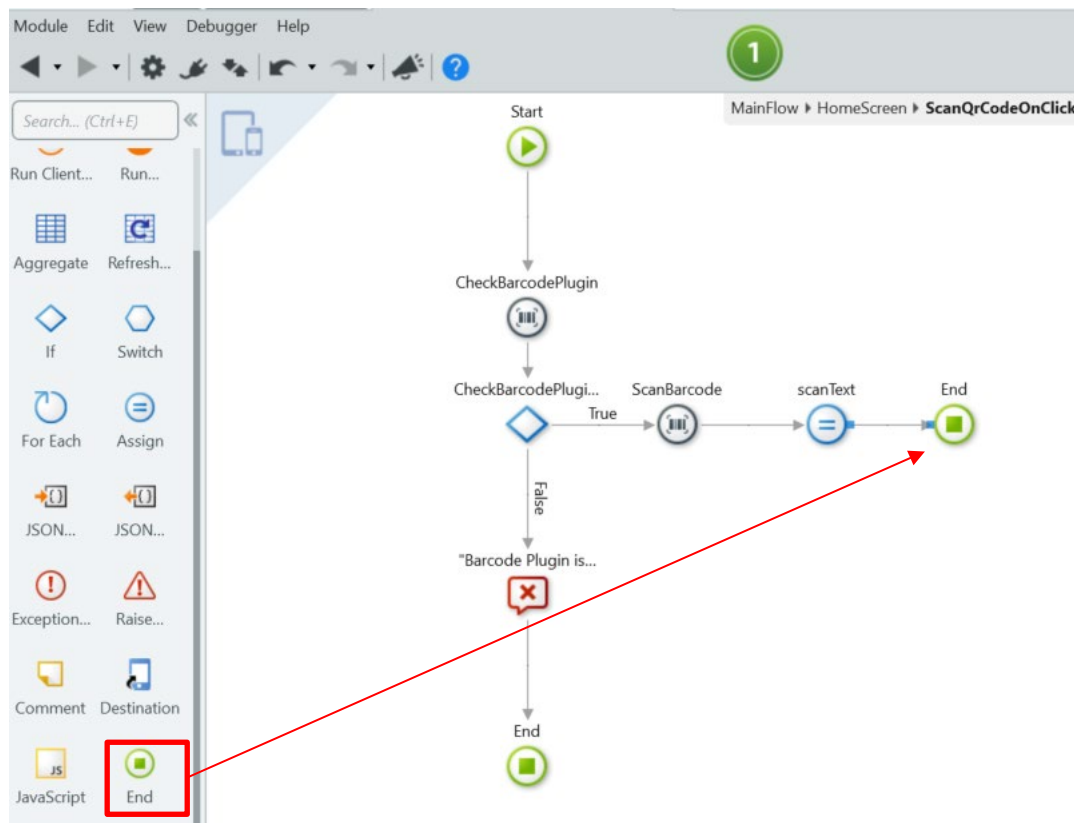
18. You may click on the ① 1-Click Publish button to update and publish the app. For the Barcode Scanner, we will need to package the app as an android application in order to test the app. If you do not have an android phone, refer to the demonstration by the trainer.

You may scan the following QR code for your testing, which is the link to RP website.

End of Lab 7