Single Screen Mobile Application Lab Guide

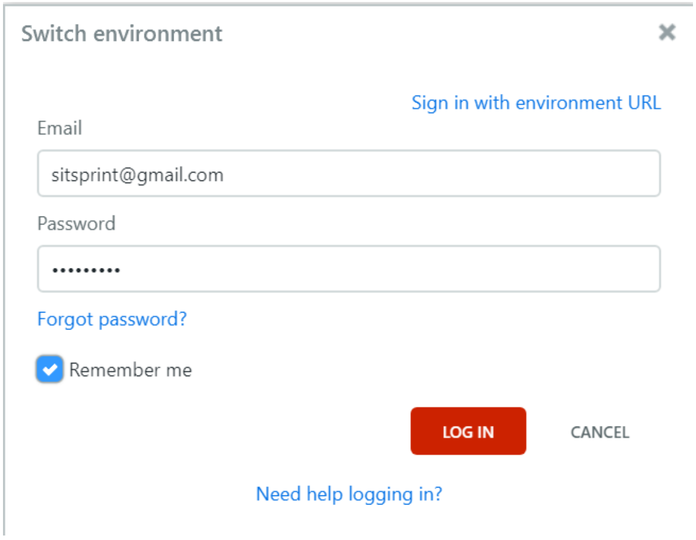
Exercise 1: Create a BMI Mobile Application

In this part of the exercise, you will create the **BMI** application. This application will have a User Interface (Mobile) module. The **BMI** application will also have an icon and a description.

To develop any OutSystems application, we need the OutSystems Development Environment, **Service Studio**, and an OutSystems server (or **environment**).

1. Open Service Studio and login in your personal environment.
    a. Open Service Studio from the Start Menu or by double clicking the icon.

    b. In the **Connect to Environment** or **Switch Environment** dialog, enter the username and password you will be using to carry out the exercises, and clickLOG IN.



Switch environment

Sign in with environment URL

Email

sitsprint@gmail.com

Password

•••••••••

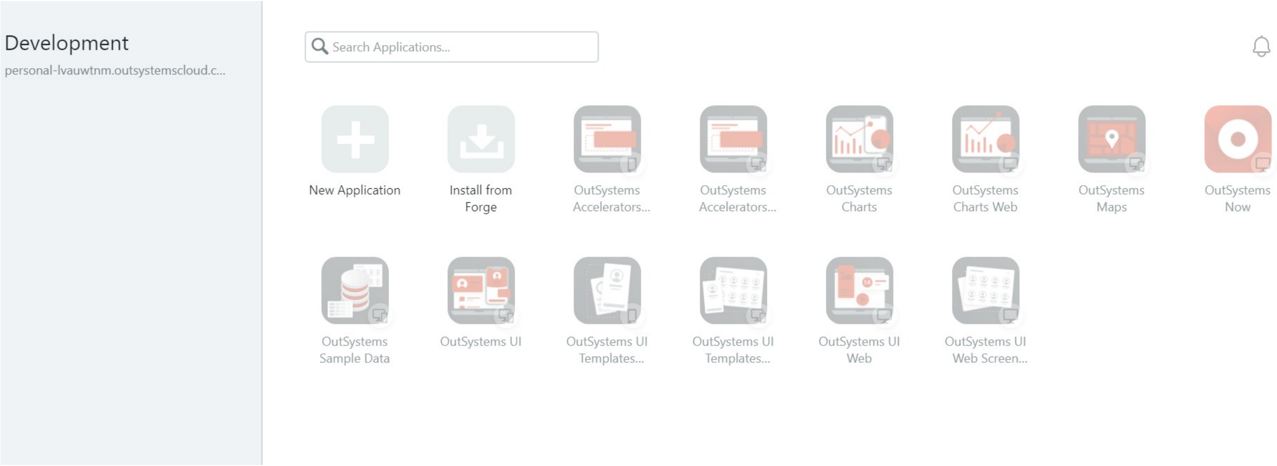
Forgot password?

☑ Remember me

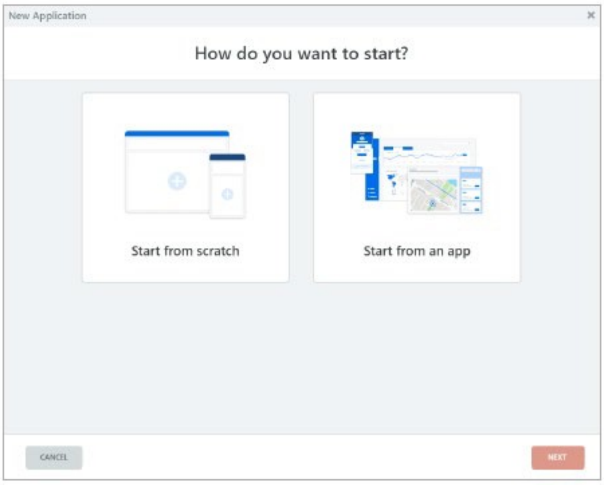
LOG IN CANCEL

Need help logging in?

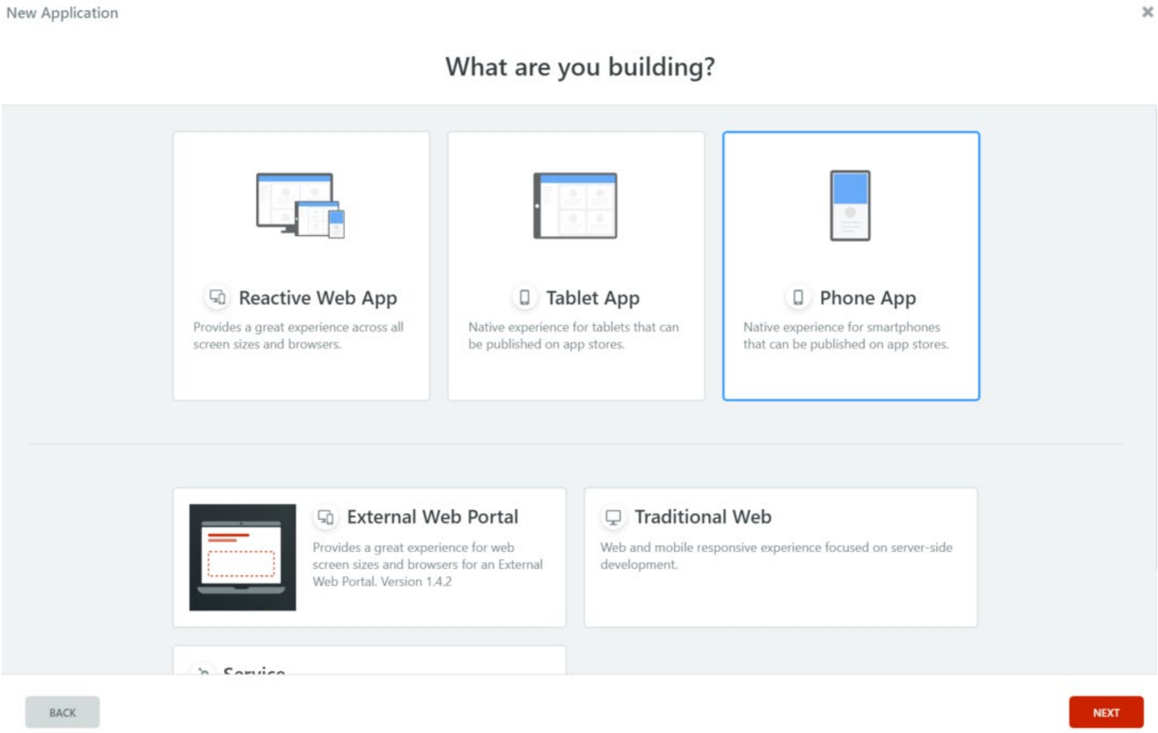
2. Create a new mobile application named **BMI App**, with a **Phone App** Module
    a. In the ‘Development’ area, click **New Application**.



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.



Provides a great experience across all screen sizes and browsers.

Native experience for tablets that can be published on app stores.

Native experience for smartphones that can be published on app stores.

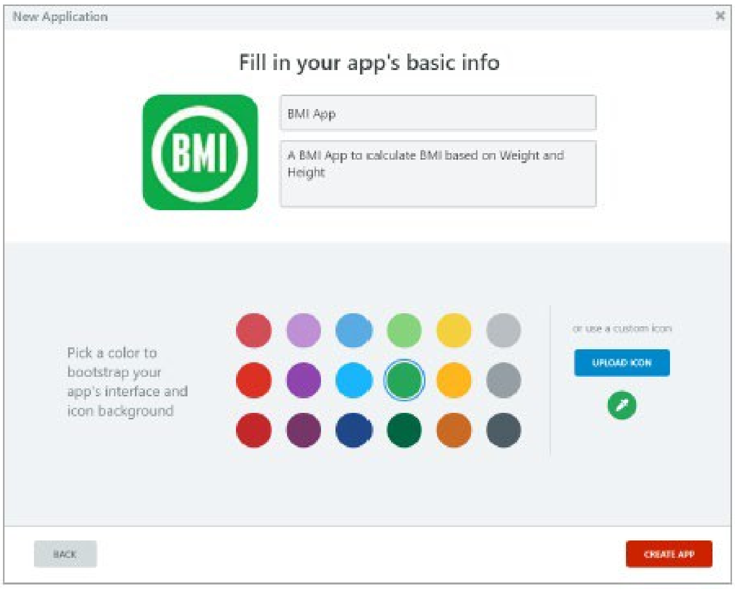
Provides a great experience for web screen sizes and browsers for an External Web Portal. Version 1.4.2

Web and mobile responsive experience focused on server-side development.

d. Set the Application Name to ***BMI App***.

e. Type in a simple description for the application.

f. Click the **Upload Icon** button, select the **bmi.png** provided in the course package. Select **Create App**.



New Application

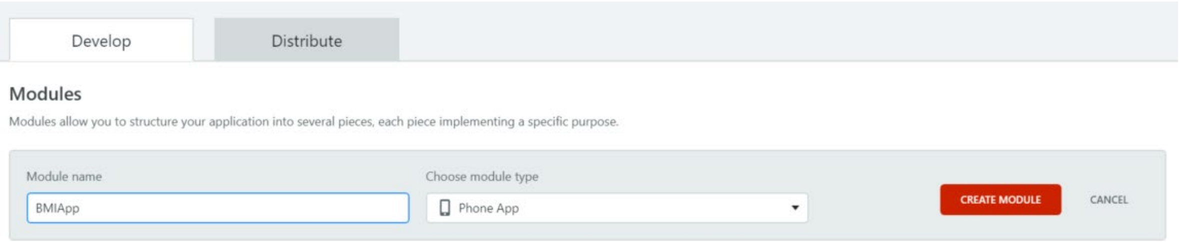
Fill in your app's basic info

BMI App

A BMI App to calculate BMI based on Weight and Height

Pick a color to bootstrap your app's interface and icon background

or use a custom icon

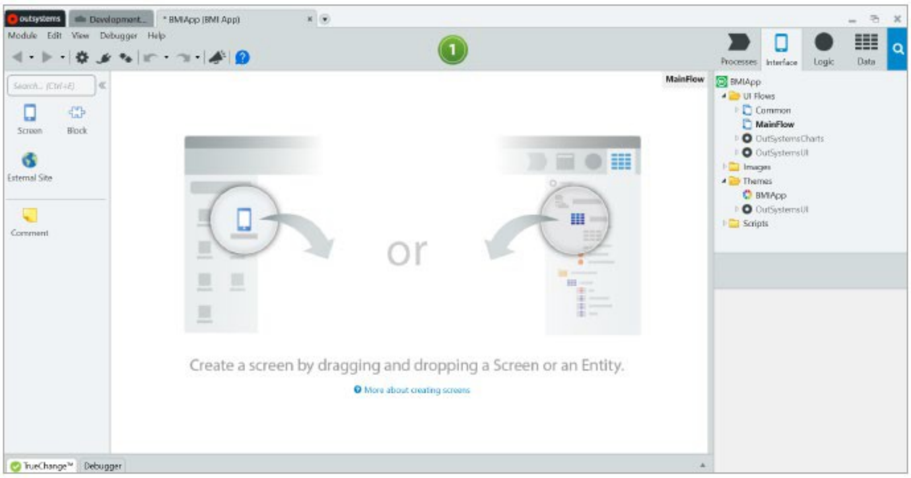


Modules

Modules allow you to structure your application into several pieces, each piece implementing a specific purpose.

Module name: BMIApp

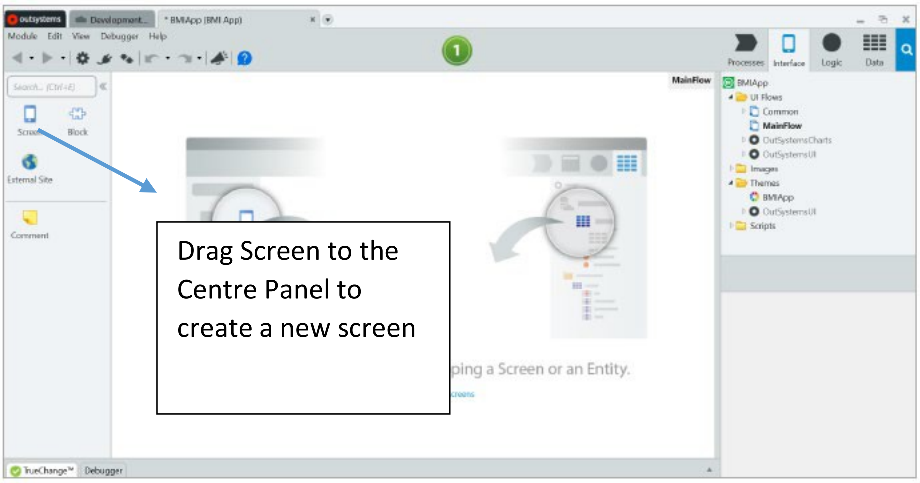
Choose module type: Phone App



Create a screen by dragging and dropping a Screen or an Entity.

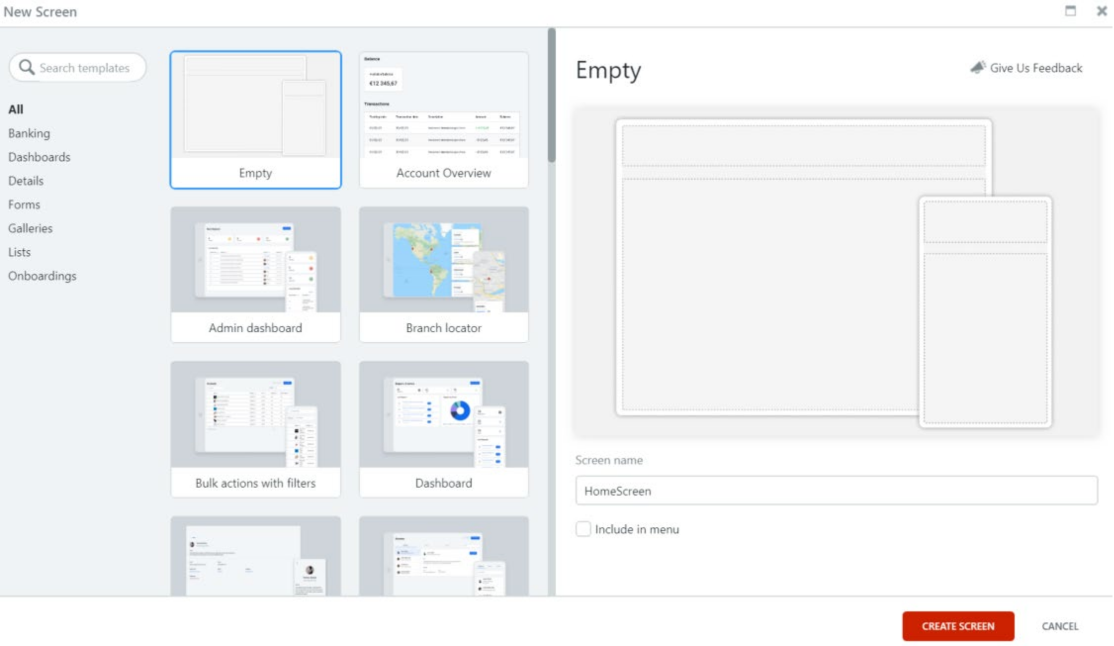
More about creating screens

i. In the **Interface** Tab, drag the Screen to the Center of the Panel to create a new screen.



Drag Screen to the Centre Panel to create a new screen

j. Select **Empty** in the **New Screen** Dialog, and enter **HomeScreen** for the Screen name. Click on **CREATE SCREEN** button.



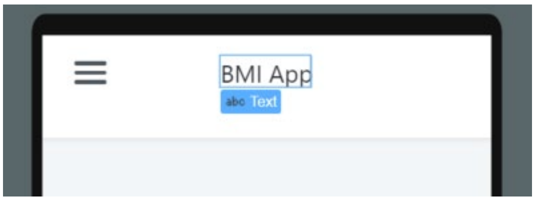
New Screen

Empty

Screen name: HomeScreen

□ Include in menu

k. Click on the highlighted box as shown below and enter the title: **BMI App**.

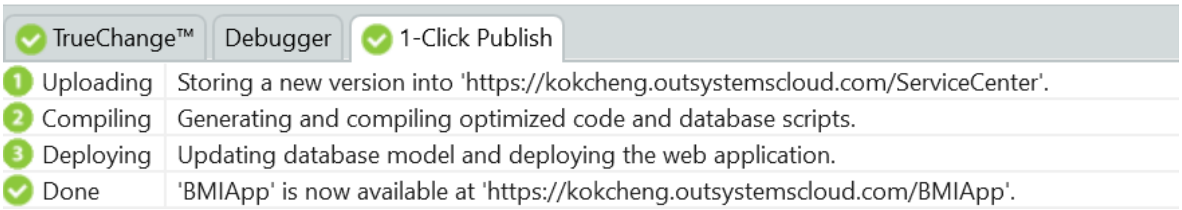


BMI App

Exercise 2: Publish the application modules and testing the App

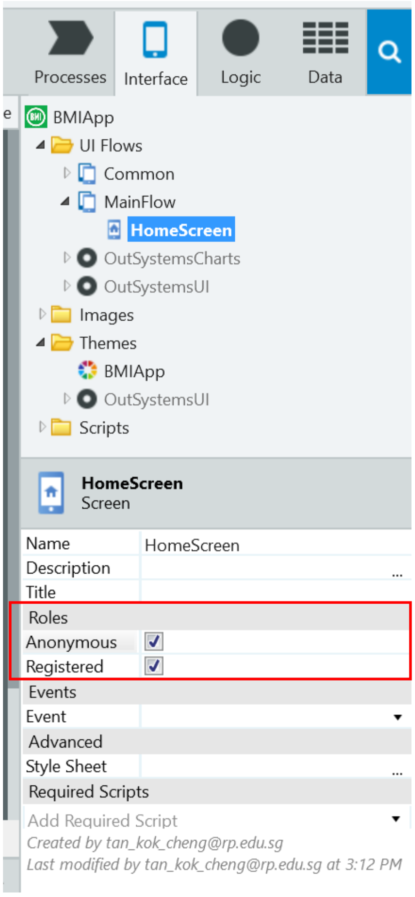
In this section, we will publish the module to the server for the first time. This will create the first version of the module in the application server. From now on, every new publish creates a new version of the module. The publishing process uploads the module's information to the server. Then, it proceeds to generate and compile the necessary code and then deploys the application to the server.

1. Click the ① 1-Click Publish button to publish the BMIApp module to the server.
2. Notice the **1-Click Publish** tab that appears near the bottom. This tab provides progress updates on the publishing process.



① Uploading	Storing a new version into 'https://kokcheng.outsystemscloud.com/ServiceCenter'.
② Compiling	Generating and compiling optimized code and database scripts.
③ Deploying	Updating database model and deploying the web application.
✓ Done	'BMIApp' is now available at 'https://kokcheng.outsystemscloud.com/BMIApp'.

3. In the **Interface** tab, click on **HomeScreen**, and under **Role**, make sure the checkbox for **Anonymous** is checked.

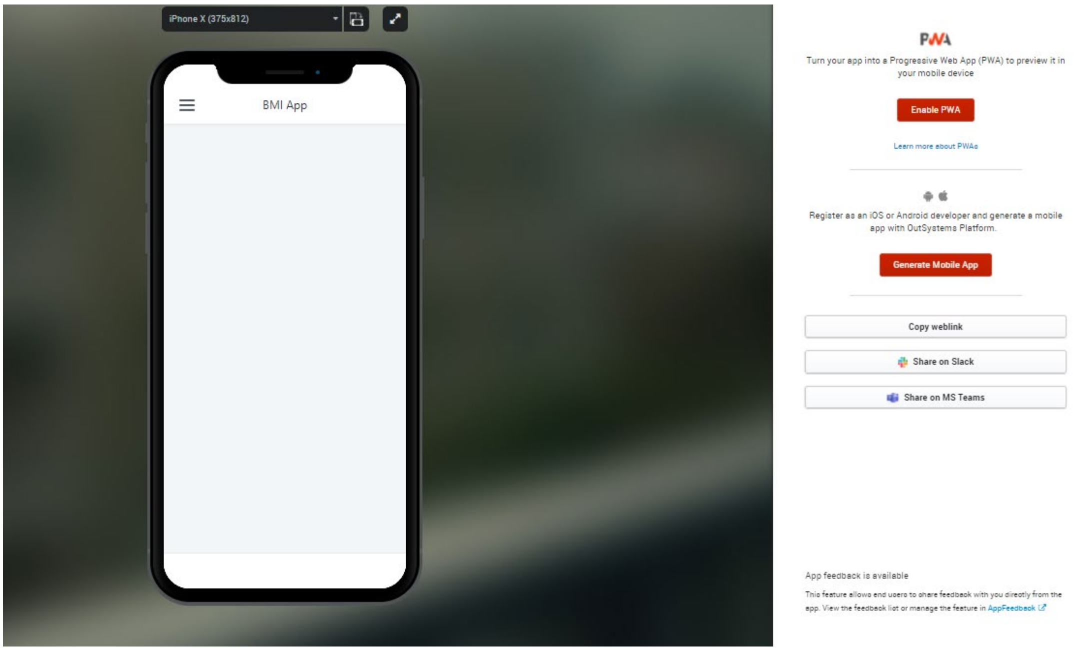


HomeScreen
Screen

Name	HomeScreen
Description	...
Title	
Roles	
Anonymous	☑
Registered	☑
Events	
Event	▾
Advanced	
Style Sheet	...
Required Scripts	
Add Required Script	▾

Created by tan_kok_cheng@rp.edu.sg
Last modified by tan_kok_cheng@rp.edu.sg at 3:12 PM

4. Click on **Open in Browser** button to launch and test the app in the browser. Note: Use Chrome Browser, the app don't work in Internet Explorer. You should be able to see the App with the title "BMI App" as shown as shown below.



Turn your app into a Progressive Web App (PWA) to preview it in your mobile device

Register as an iOS or Android developer and generate a mobile app with OutSystems Platform.

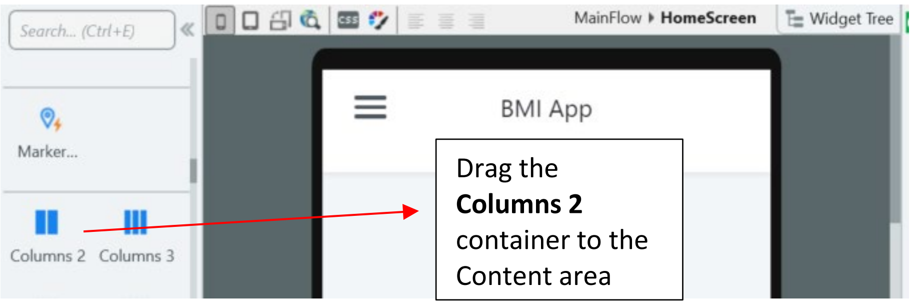
App feedback is available

This feature allows end users to share feedback with you directly from the app. View the feedback list or manage the feature in AppFeedback

---

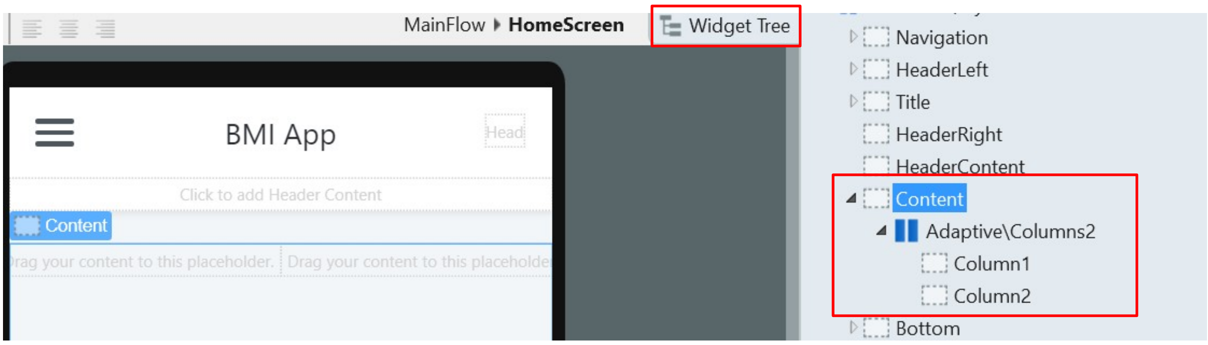
In this part of the exercise, you are going to create some user interface components for the BMI application. The user interface will be as shown below:

1. In the Interface Tab, click on HomeScreen. In the left-hand panel, scroll and search for the **Columns 2** container. Drag the widget into the HomeScreen Content area.

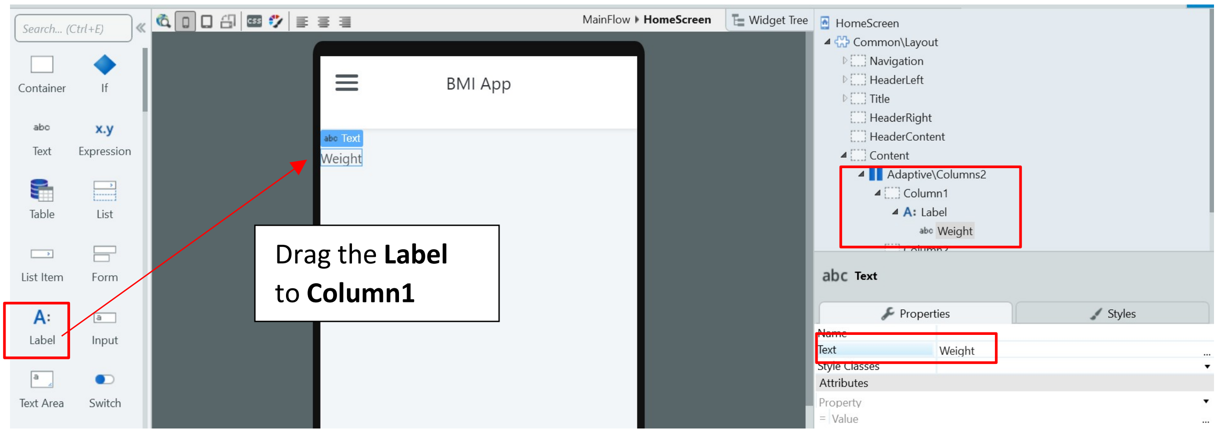


Drag the **Columns 2** container to the Content area

2. Click on **Widget Tree** as shown below, you will be able to see how the UI components or widgets are organized in relation to one another.



3. Add a **Label** component to **Column1**, and change the Text property to Weight.

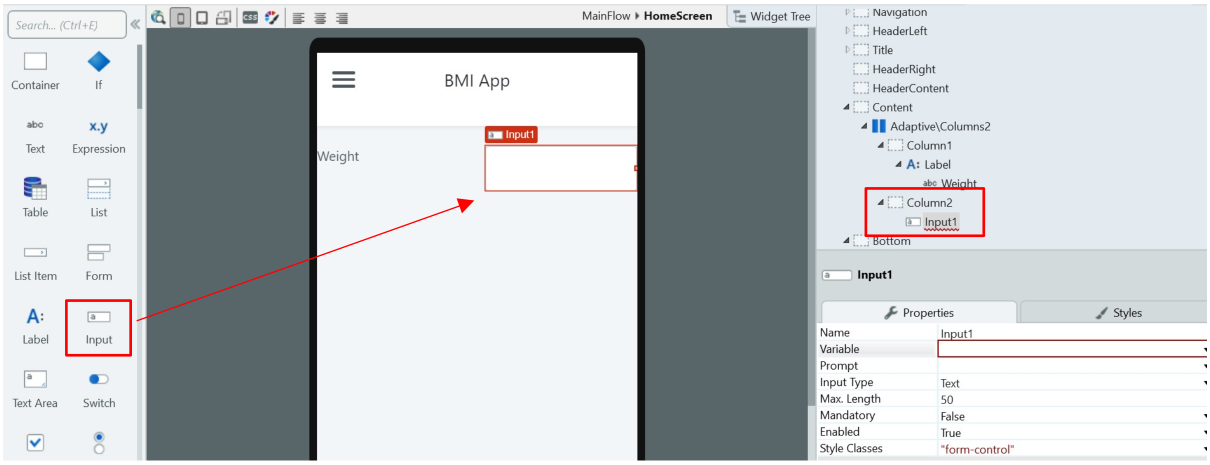


Drag the **Label** to **Column1**

abc **Text**

Name	
Text	Weight
Style Classes	

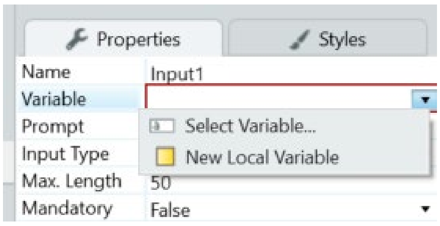
4. Add an **Input** component to Column2. You will see a red box around the Input component, indicating that there is some error.



Input1

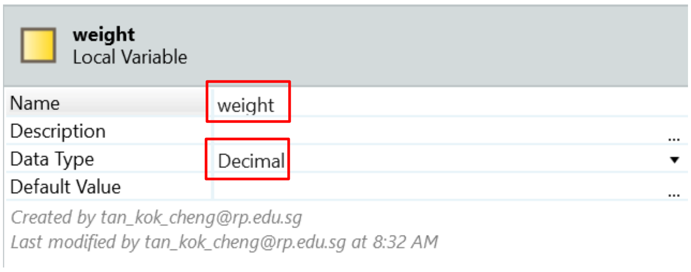
Name	Input1
Variable	
Prompt	
Input Type	Text
Max. Length	50
Mandatory	False
Enabled	True
Style Classes	"form-control"

5. Click on the **Variable** property, and select **New Local Variable**.



Name	Input1
Variable	
Prompt	
Input Type	
Max. Length	50
Mandatory	False

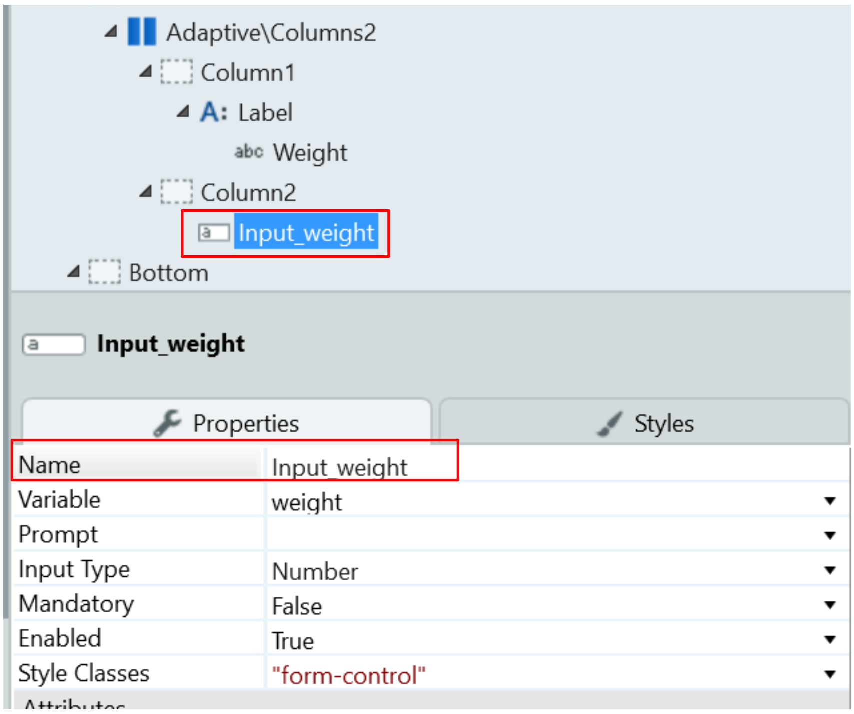
6. Enter **weight** for the Name, and change the **Data Type** to **Decimal**.



weight
Local Variable

Name	weight
Description	...
Data Type	Decimal
Default Value	...

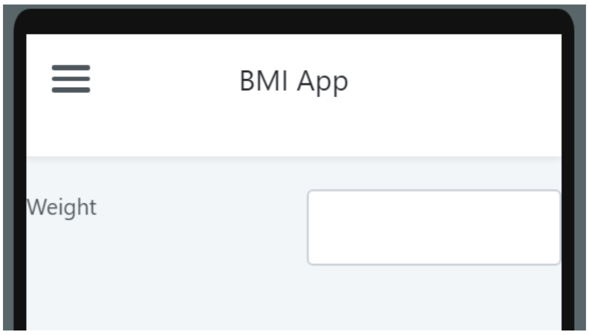
Created by tan_kok_cheng@rp.edu.sg
Last modified by tan_kok_cheng@rp.edu.sg at 8:32 AM



Input_weight

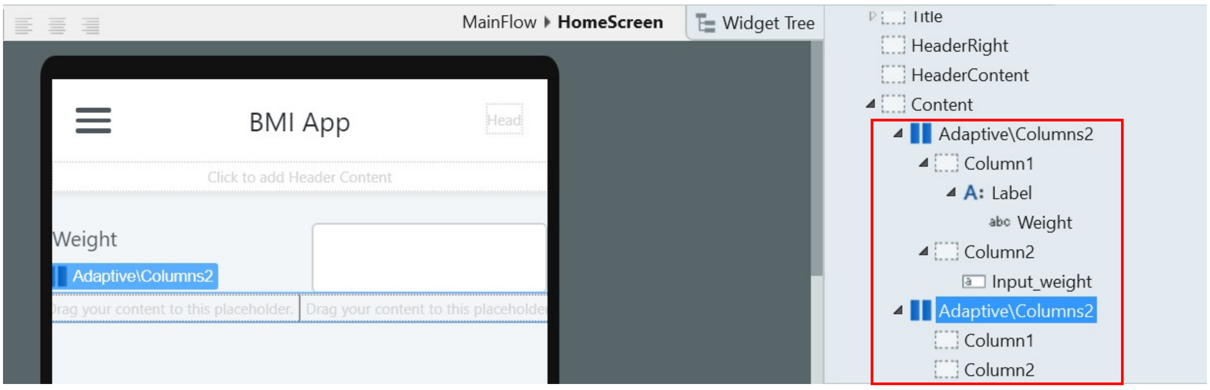
Name	Input_weight
Variable	weight
Prompt	
Input Type	Number
Mandatory	False
Enabled	True
Style Classes	"form-control"

8. At this point, the User Interface will be as shown below. You may click on the 1-Click Publish button to save the app in the server.

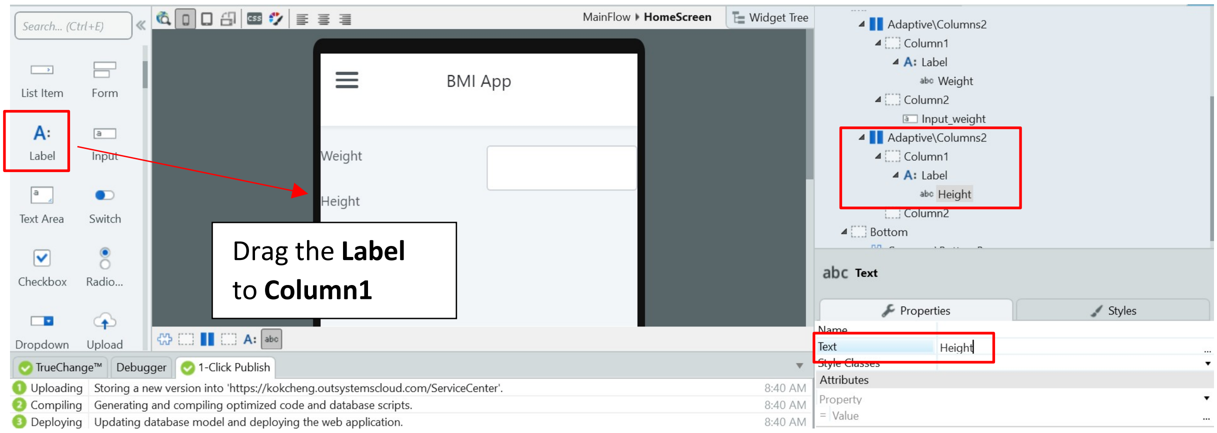


9. We will be repeating Step 1 to Step 8 to create the **Label** and **Input** component for the **Height** related information.

10. Add a **Columns 2** component below the one created at step 1.



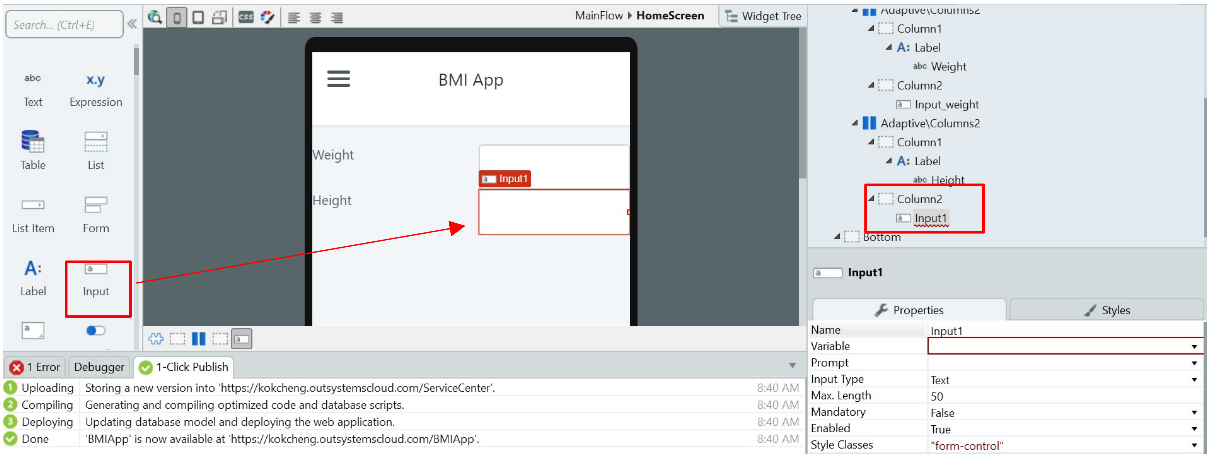
11. Add a **Label** component to **Column1**, and change the **Text** property to **Height**.



Drag the **Label** to **Column1**

1 Uploading Storing a new version into 'https://kokcheng.outsystemscloud.com/ServiceCenter'. 8:40 AM
2 Compiling Generating and compiling optimized code and database scripts. 8:40 AM
3 Deploying Updating database model and deploying the web application. 8:40 AM

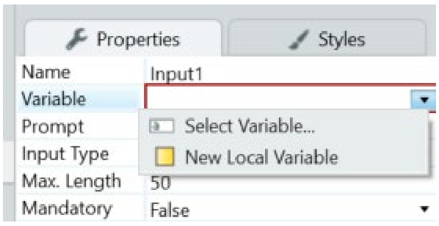
12. Add an **Input** component to Column2. You will see a red box around the Input component, indicating that there is some error.



```
Uploading   Storing a new version into 'https://kokcheng.outsystemscloud.com/ServiceCenter'.   8:40 AM
Compiling   Generating and compiling optimized code and database scripts.                       8:40 AM
Deploying   Updating database model and deploying the web application.                          8:40 AM
Done        'BMIApp' is now available at 'https://kokcheng.outsystemscloud.com/BMIApp'.         8:40 AM
```

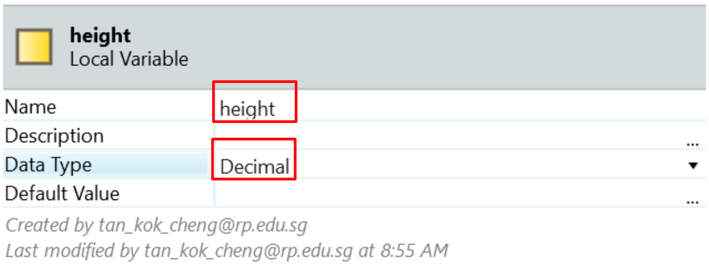
Name	Input1
Variable	
Prompt	
Input Type	Text
Max. Length	50
Mandatory	False
Enabled	True
Style Classes	"form-control"

13. Click on the **Variable** property, and select **New Local Variable**.



Name	Input1
Variable	
Prompt	
Input Type	
Max. Length	50
Mandatory	False

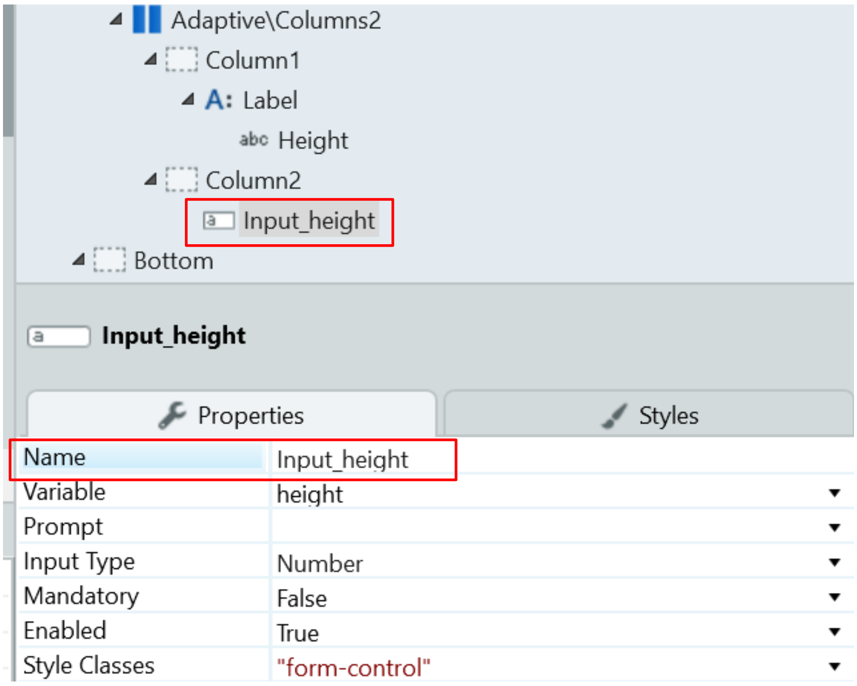
14. Enter **height** for the Name, and change the **Data Type** to **Decimal**.


height
Local Variable

Name	height
Description	
Data Type	Decimal
Default Value	

Created by tan_kok_cheng@rp.edu.sg
Last modified by tan_kok_cheng@rp.edu.sg at 8:55 AM

15. Select the **Input** component, change the **Name** property to **Input_height**.
    Note: it is easier to select component using the Widget Tree.

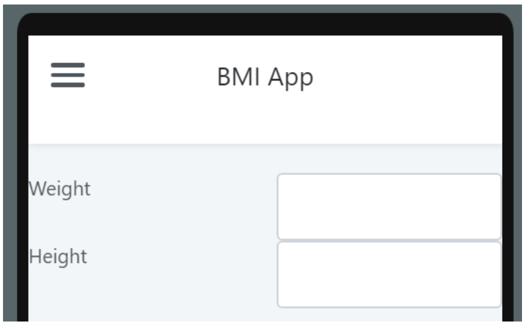


Adaptive\Columns2
Column1
A: Label
abc Height
Column2
Input_height
Bottom

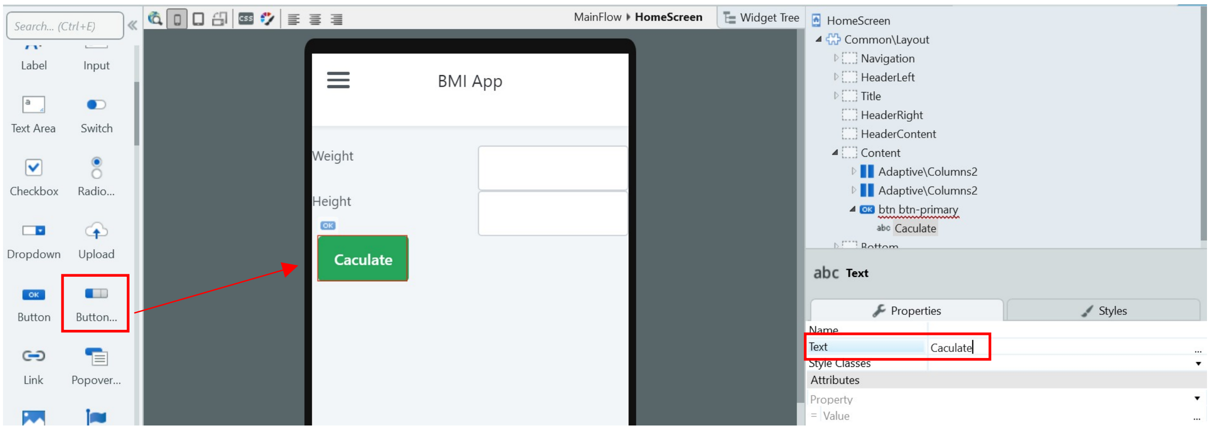
Input_height

Name	Input_height
Variable	height
Prompt	
Input Type	Number
Mandatory	False
Enabled	True
Style Classes	"form-control"

16. At this point, the User Interface will be as shown below. You may click on the 1-Click Publish button to save the app in the server.



17. Add a Button to the screen and change the **Text** property to **Calculate**. You will see that there is an error message.

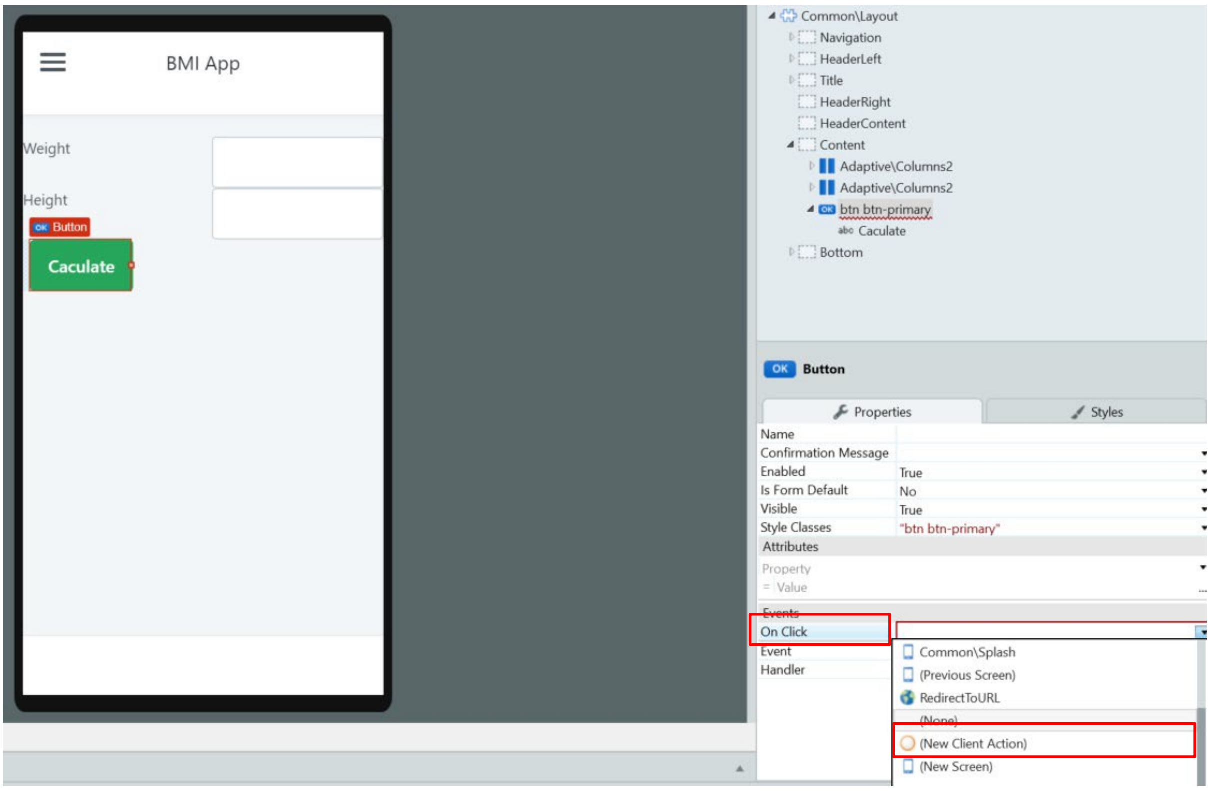


HomeScreen
Common\Layout
Navigation
HeaderLeft
Title
HeaderRight
HeaderContent
Content
Adaptive\Columns2
Adaptive\Columns2
btn btn-primary
abc Caculate
Bottom

abc Text

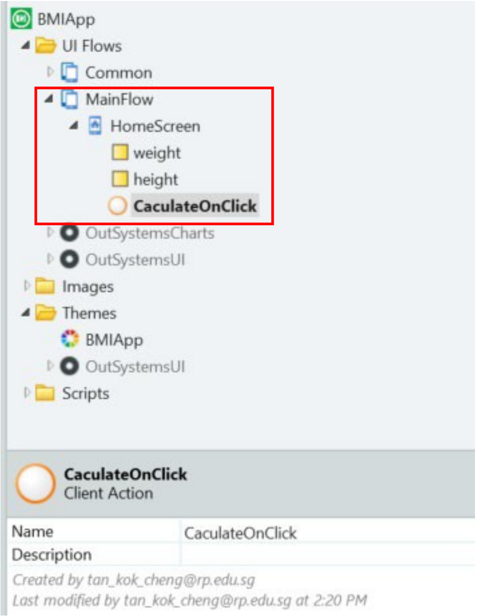
Name	
Text	Caculate
Style Classes	
Attributes	
Property	
= Value	

18. Add a **New Client Action** to the **On Click** event for the **Calculate** button.



Name	
Confirmation Message	
Enabled	True
Is Form Default	No
Visible	True
Style Classes	"btn btn-primary"
Attributes	
Property	
= Value	
Events	
On Click	
Event Handler	

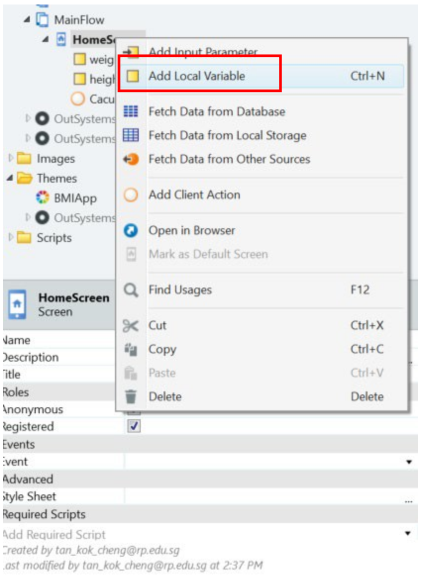
19. The New Client Action will be automatically given the name **CalculateOnClick**. You will find that the action can be found under the HomeScreen as shown below.



Name	CaculateOnClick
Description	

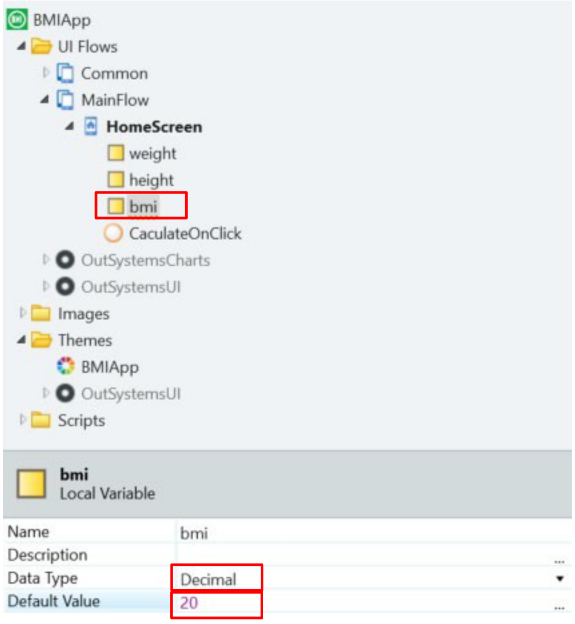
Created by tan_kok_cheng@rp.edu.sg
Last modified by tan_kok_cheng@rp.edu.sg at 2:20 PM

20. Right click on HomeScreen, add select **Add Local Variable**.



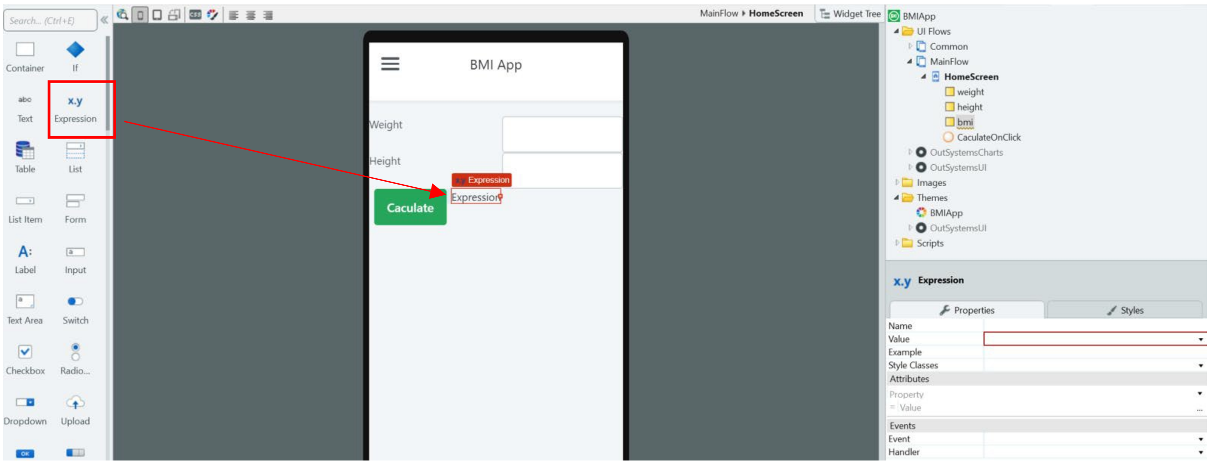
21. Name the variable as **bmi**, and select **Decimal** as the Data Type. Set the default value to **20** or any other numeric value of your choice.

22.22.

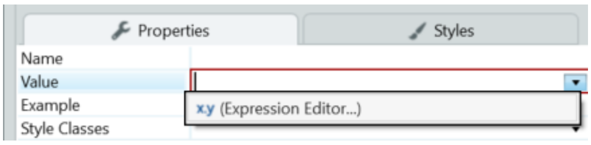


Name bmi
Description
Data Type Decimal
Default Value 20

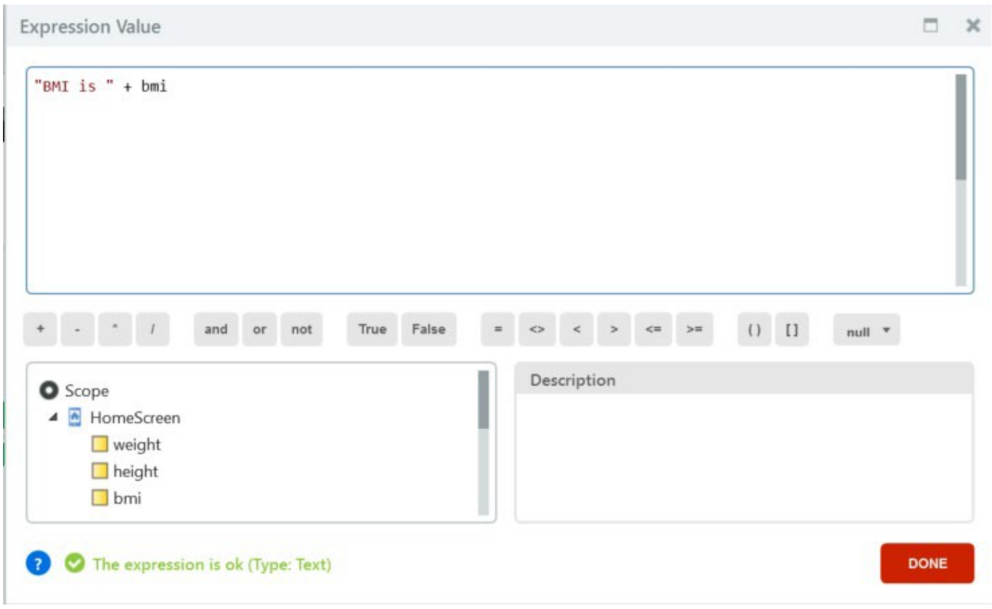
23. Double-click on HomeScreen to show the user interface of the BMI app again. Drag the **Expression** component, and place it next to the Calculate Button.



24. With the **expression** component selected, click on the **Value** property, and click on x.y (Expression Editor…) as show below.



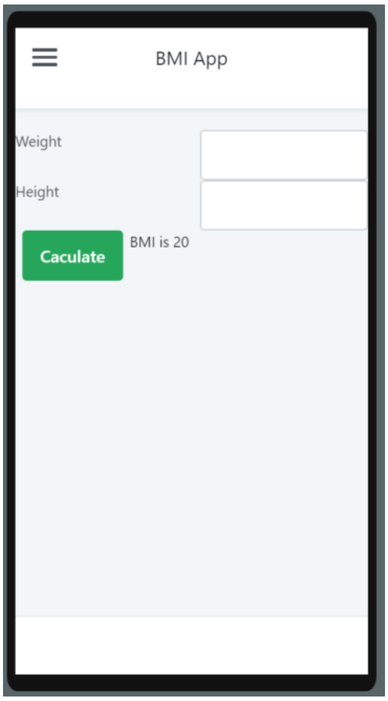
25. In the Expression Value dialog, enter the string as shown:
“BMI is “ + bmi
and click on the **Done** button.



```
"BMI is " + bmi
```

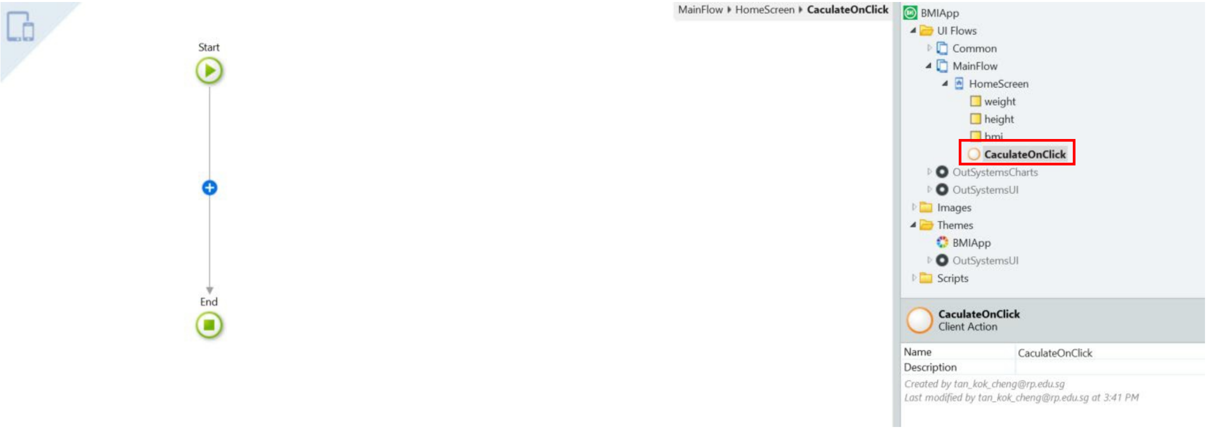
The expression is ok (Type: Text)

26. We have completed the creation of all the UI components. At this point, you should have a screen as shown below.

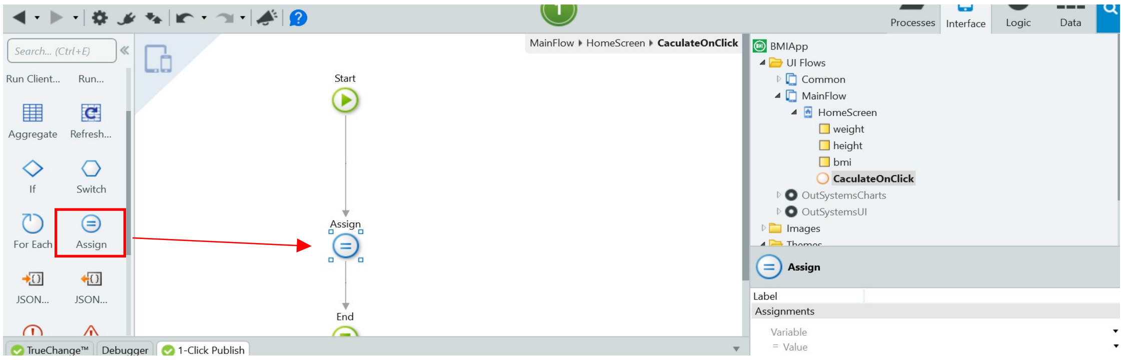


Exercise 4: Adding Application Logic

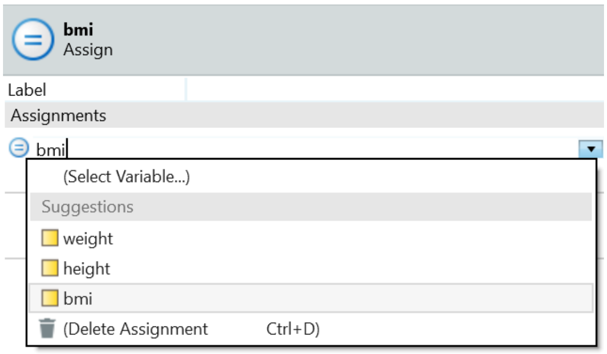
1. Click on **CalculateOnClick** client action, and you will be able to see the workflow area as shown below.



2. Drag the **Assign** widget, and place it between the Start and End node



3. Under Assignments, set the **Variable** and **Value** property.
   a. Variable: bmi

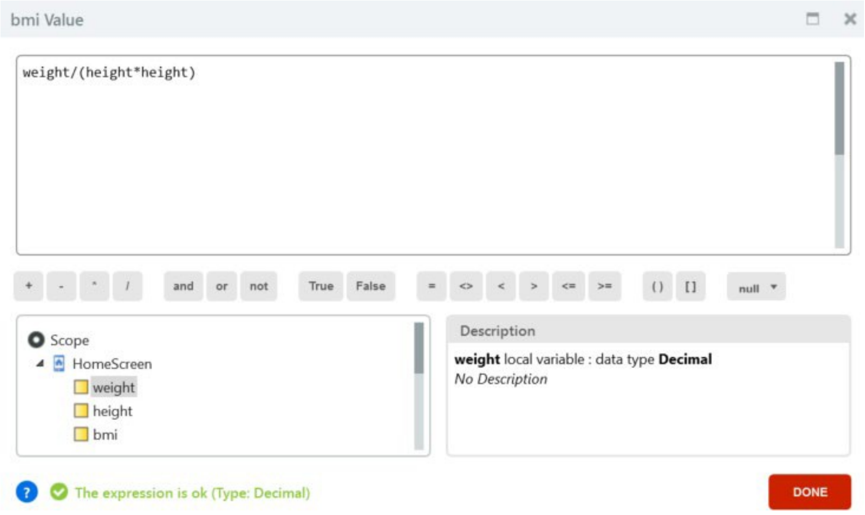


bmi
Assign

Label
Assignments

bmi

   b. Value: weight/(height * height)



bmi Value

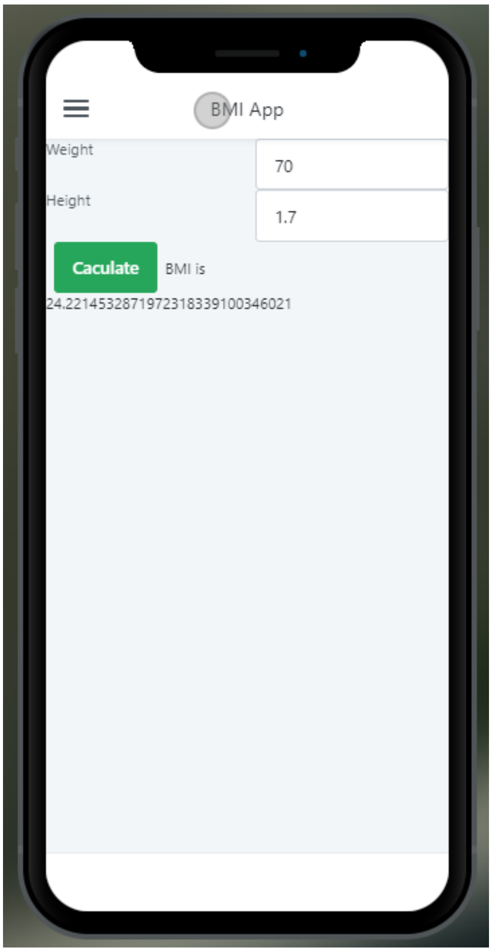
```
weight/(height*height)
```

weight local variable : data type **Decimal**
No Description

The expression is ok (Type: Decimal)

4. Click on the **1-Click Publish** button to upload, compile and deploy the app. Click on the **Open in Browser** button to test the app.
   a. Enter Weight: **70**
   b. Enter Height: **1.7**
   c. Click on the Calculate Button

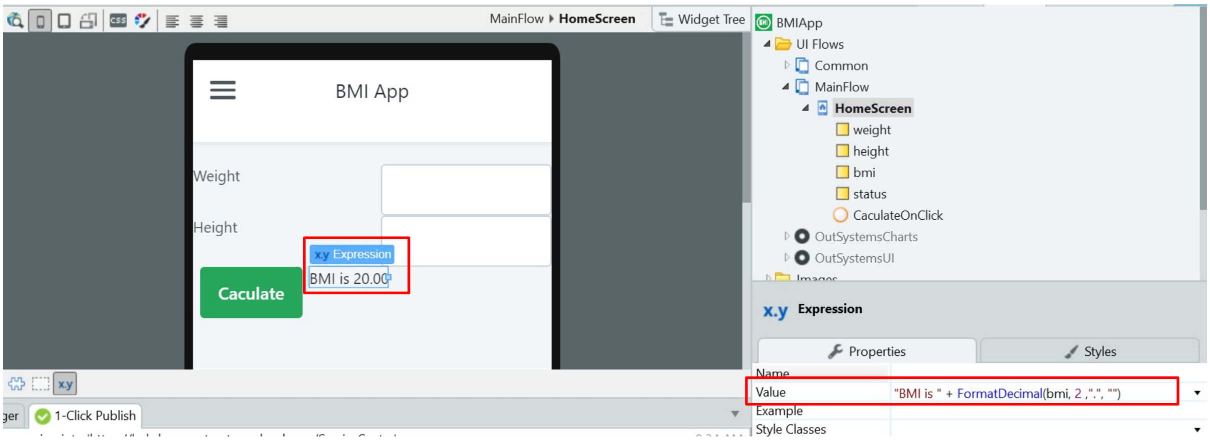
You should see the result as shown below.



BMI is 24.2214532871972318339100346021

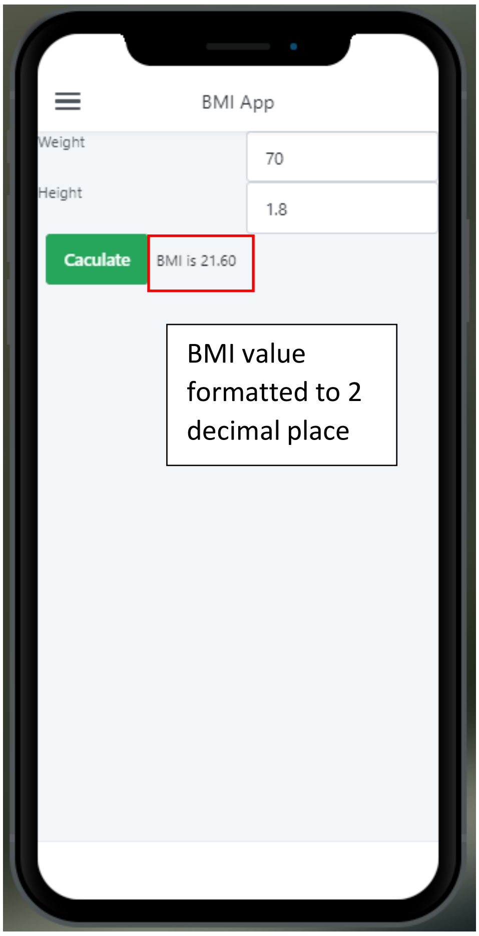
5. You will have noticed that the BMI value have many decimal places. We will use the **FormatDecimal()** function to set the BMI value to 2 decimal place. Go to the **HomeScreen**, select the **Expression** component next to the **Calculate** button.
   a. Click on the Value property, and update the value to

```
"BMI is " + FormatDecimal(bmi, 2 ,".", "")
```



Value "BMI is " + FormatDecimal(bmi, 2 ,".", "")

6. Click on the ① **1-Click Publish** button to deploy and test the app again. You should see that the BMI value is formatted to 2 decimal place.



Weight 70

Height 1.8

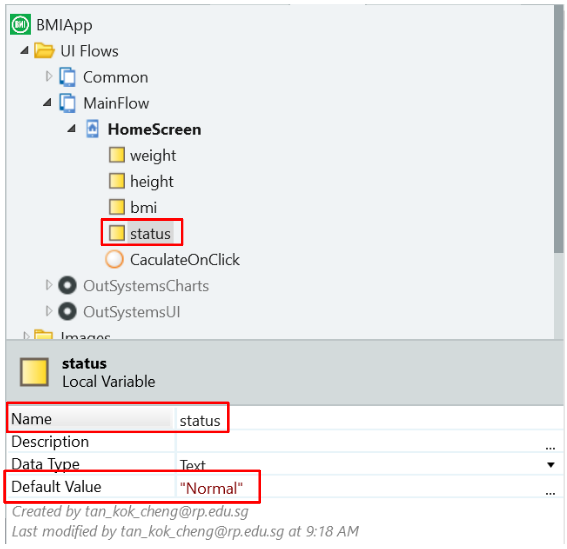
BMI is 21.60

BMI value formatted to 2 decimal place

7. In the next section, we will add application logic to display the BMI status of an individual based on the BMI value. For simplicity, we will be using the following mapping of BMI value to status.

BMI Value	BMI Status
>30	Obese
>25	Overweight
>20	Normal
<= 20	Underweight

8. Declare another Local Variable, name it as **status**, and give it a default value of **Normal**.

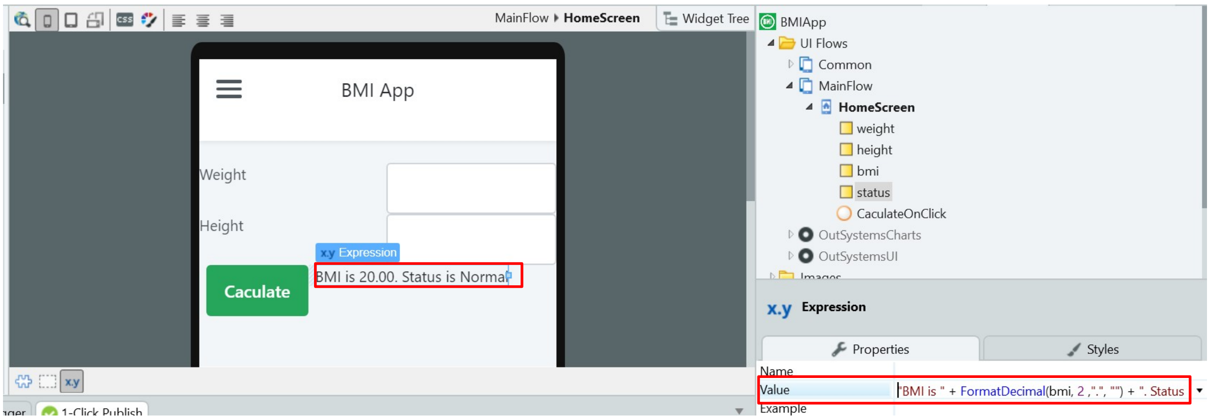


Name	status
Description	
Data Type	Text
Default Value	"Normal"

Created by tan_kok_cheng@rp.edu.sg
Last modified by tan_kok_cheng@rp.edu.sg at 9:18 AM

9. Update the Expression component next to the Calculate button with the following value:

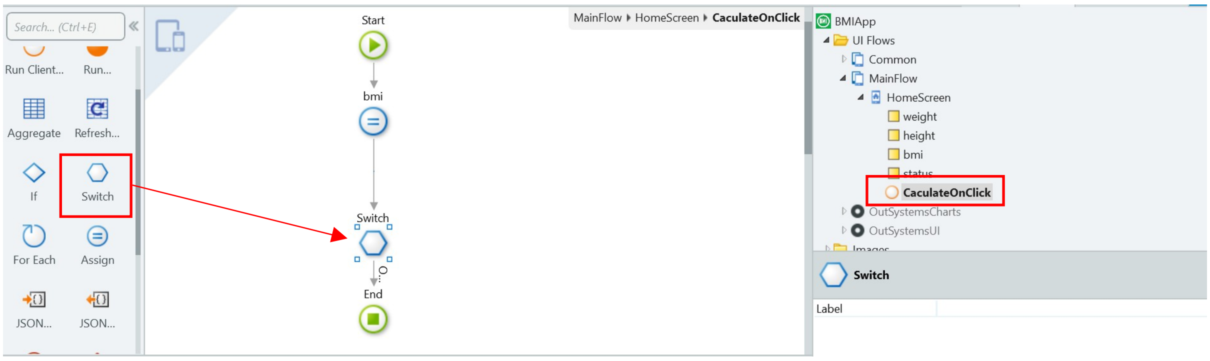
```
"BMI is " + FormatDecimal(bmi, 2 ,".", "") + ". Status is " + status
```



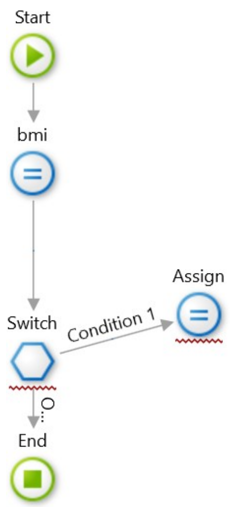
BMI is 20.00. Status is Normal

Name	
Value	"BMI is " + FormatDecimal(bmi, 2 ,".", "") + ". Status
Example	

10. In the **HomeScreen**, click on the **CalculateOnClick** client action to display the workflow view for the client action. Add a **Switch** widget to the workflow as shown below.

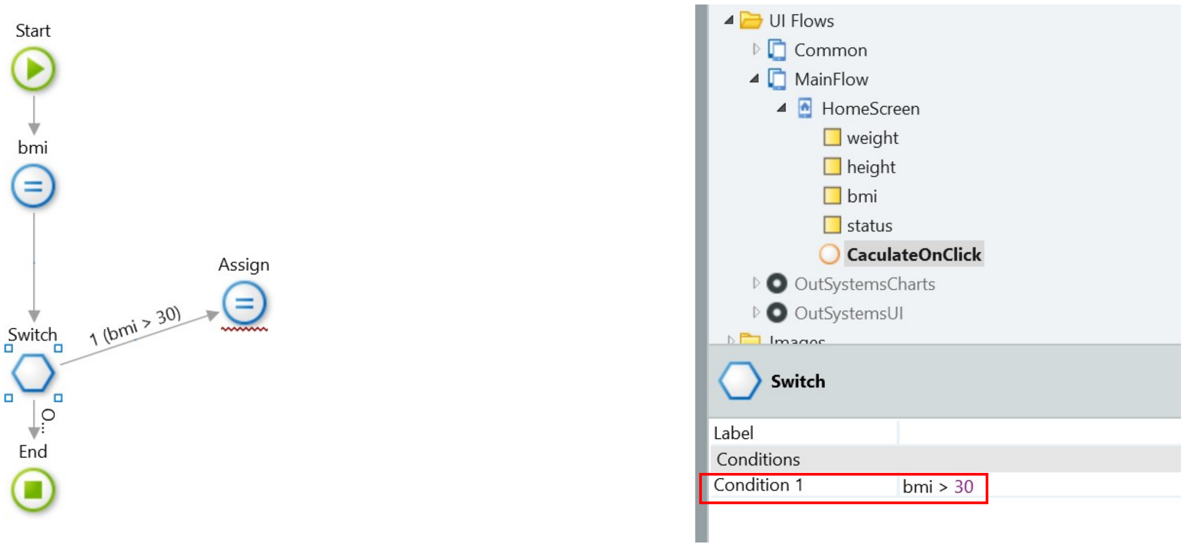


11. Add an Assign widget on the right side of the Switch widget. Connect the Switch widget to the Assign widget. You will see the word Condition 1 as shown below.



12. Select the **Switch** widget, set the Condition 1 property to the following:

```
bmi > 30
```

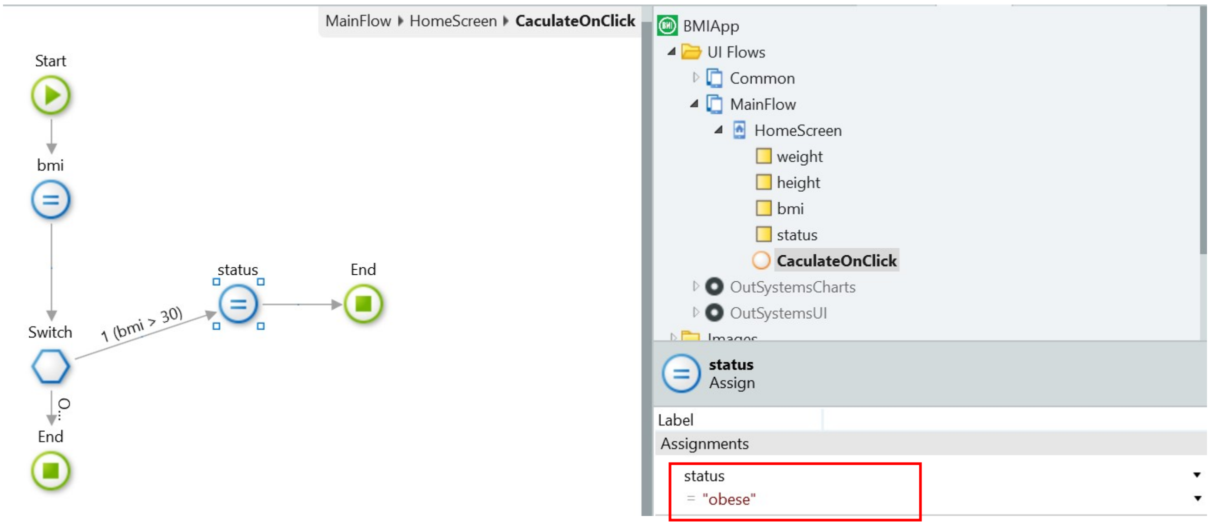


UI Flows
Common
MainFlow
HomeScreen
weight
height
bmi
status
CaculateOnClick
OutSystemsCharts
OutSystemsUI
Images

Switch

Label	
Conditions	
Condition 1	bmi > 30

13. Select the Assign widget, set the Local Variable: **status** value to “**obese**”.



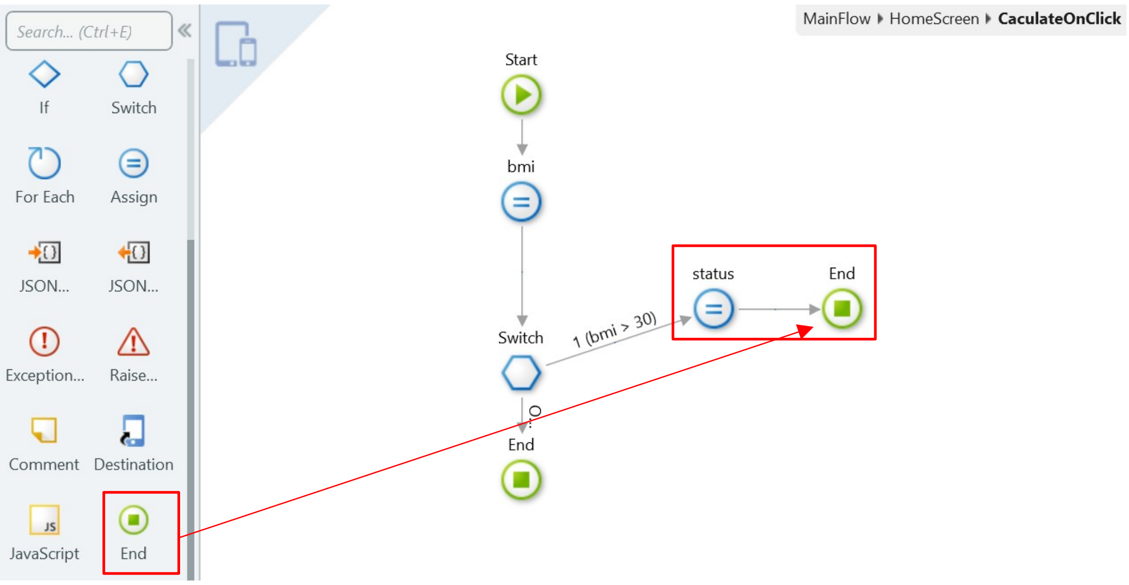
BMIApp
UI Flows
Common
MainFlow
HomeScreen
weight
height
bmi
status
CaculateOnClick
OutSystemsCharts
OutSystemsUI
Images

status
Assign

Label	
Assignments	

status
= "obese"

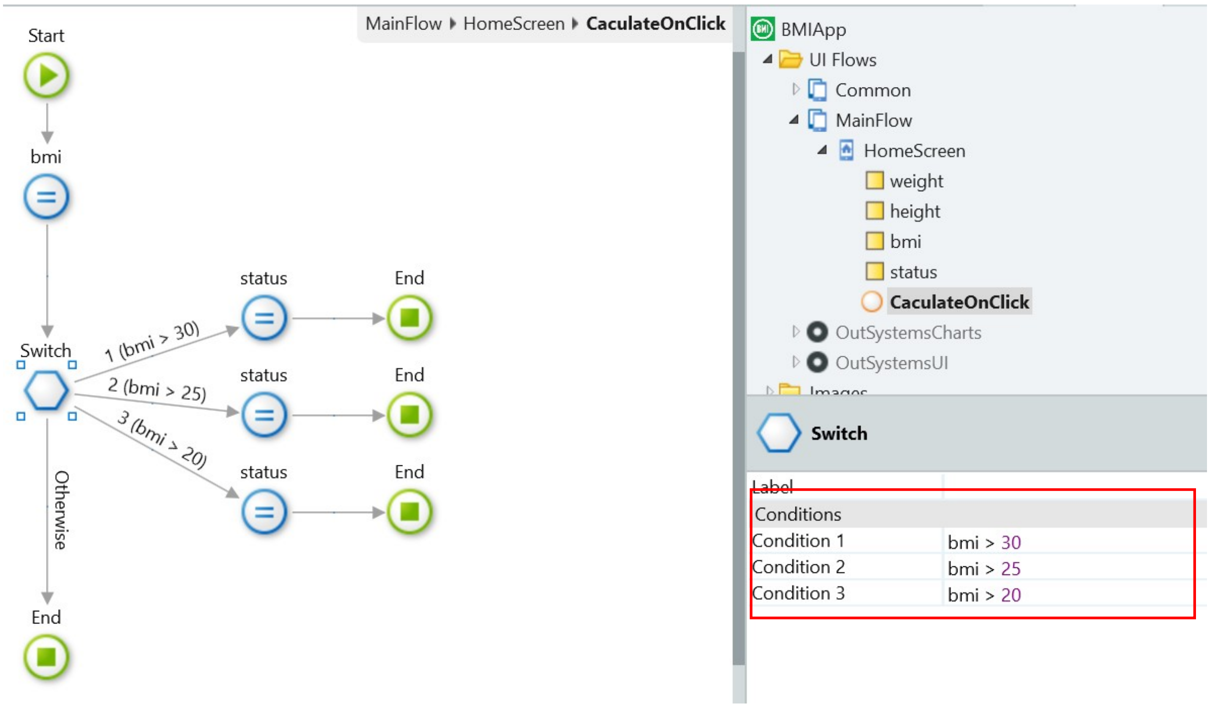
14. Add an **End** widget, and connect the **status** assignment widget to the End widget.



15. Repeat Step 11 to Step 14 for the following set of BMI values:

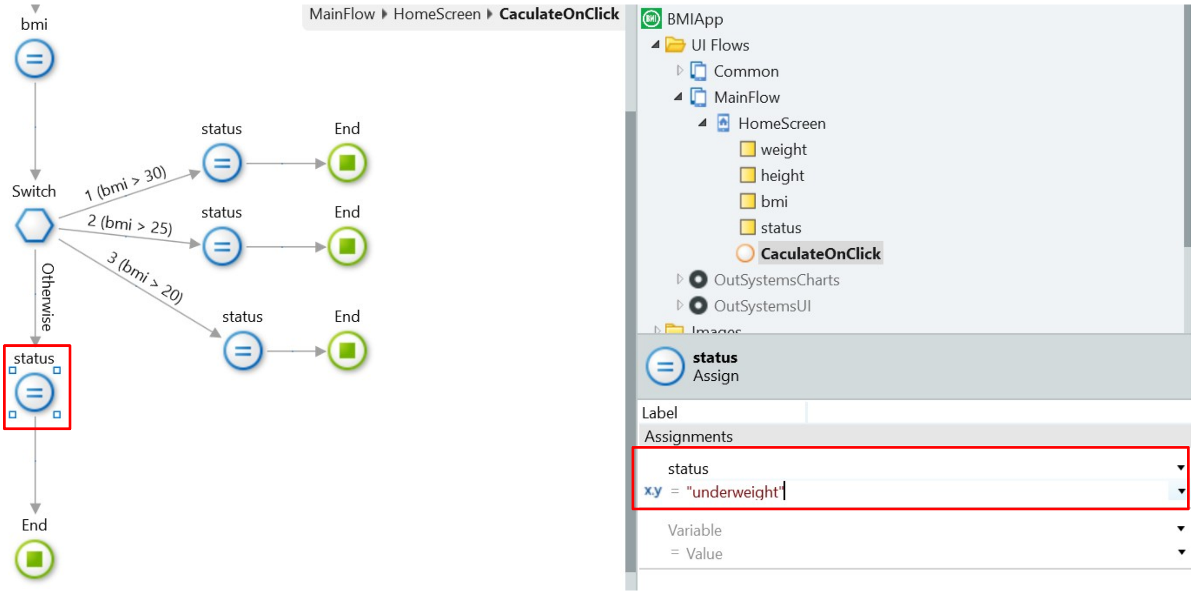
BMI Value	BMI Status
>25	Overweight
>20	Normal

Upon completing all the steps, we will produce a similar flowchart as shown below.



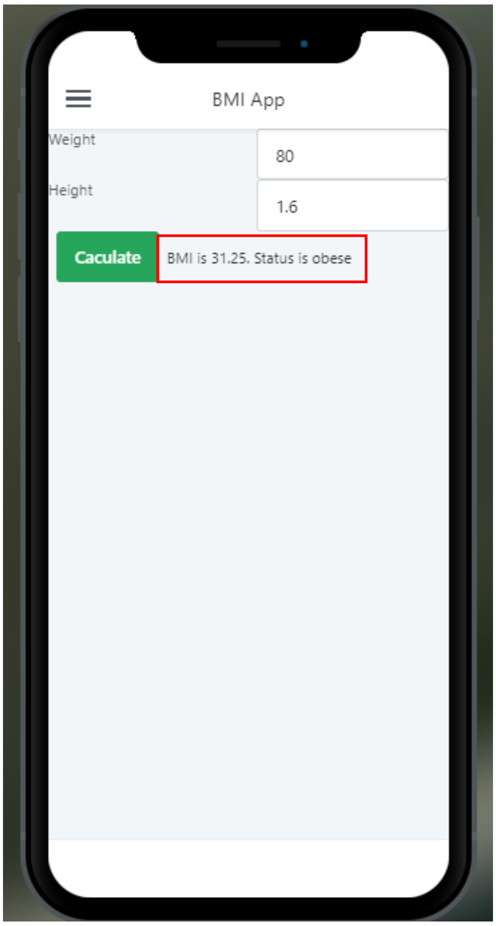
Label	
Conditions	
Condition 1	bmi > 30
Condition 2	bmi > 25
Condition 3	bmi > 20

16. Finally, add an Assign widget for the Otherwise branch of the Switch widget. Set the **status** variable to **underweight**.



status
= "underweight"

17. Click on the 1 **1-Click Publish** button to deploy and test the app again. You should see the BMI status displayed changes according to the BMI value.



Weight: 80
Height: 1.6

BMI is 31.25. Status is obese

End of Lab 1