Dashboard Exercise

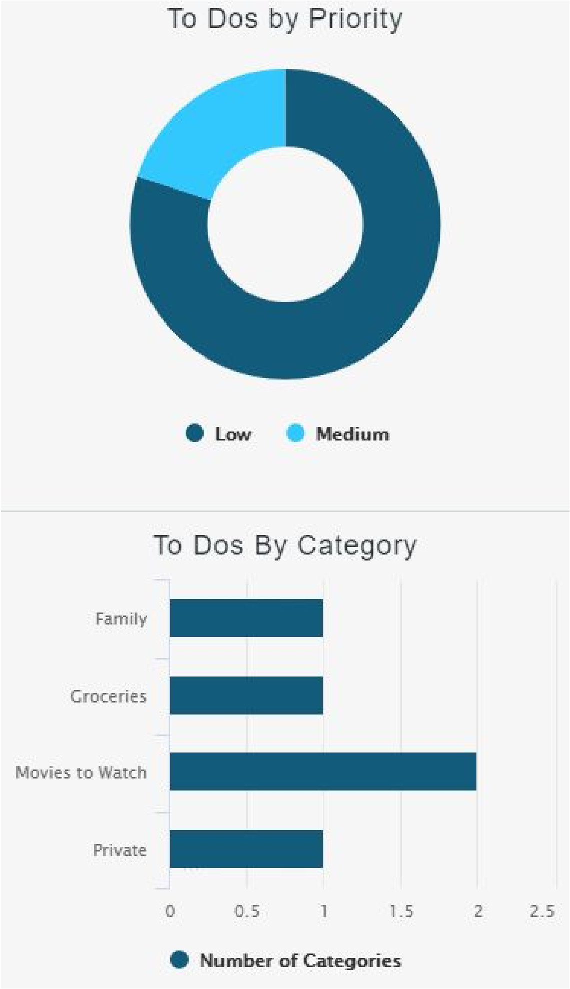


Table of Contents

Introduction

In this exercise lab, we will create the final Screen of our application, the Dashboard. This Screen has an objective of displaying information about the ToDos, namely how many ToDos do we have in our Local Storage per priority and per category. This information will obviously vary depending on the user.

We will start by creating the Screen and define links from the Menu and Bottom Bar to provide easy access to the Screen.

Then, we will create a Donut Chart that will display the number of To Dos, by priority, and a Bar Chart to display the number of To Dos, by category. This will require two Aggregates to fetch the information from local storage, meaning that will be different from user to user.

As a summary, in this specific exercise lab, we will:

- Create the Dashboard Screen and create the Links to access it
- Create a Donut Chart for displaying the To Dos by Priority
- Create a Bar Chart for displaying the To Dos by Priority

Create the Dashboard Screen

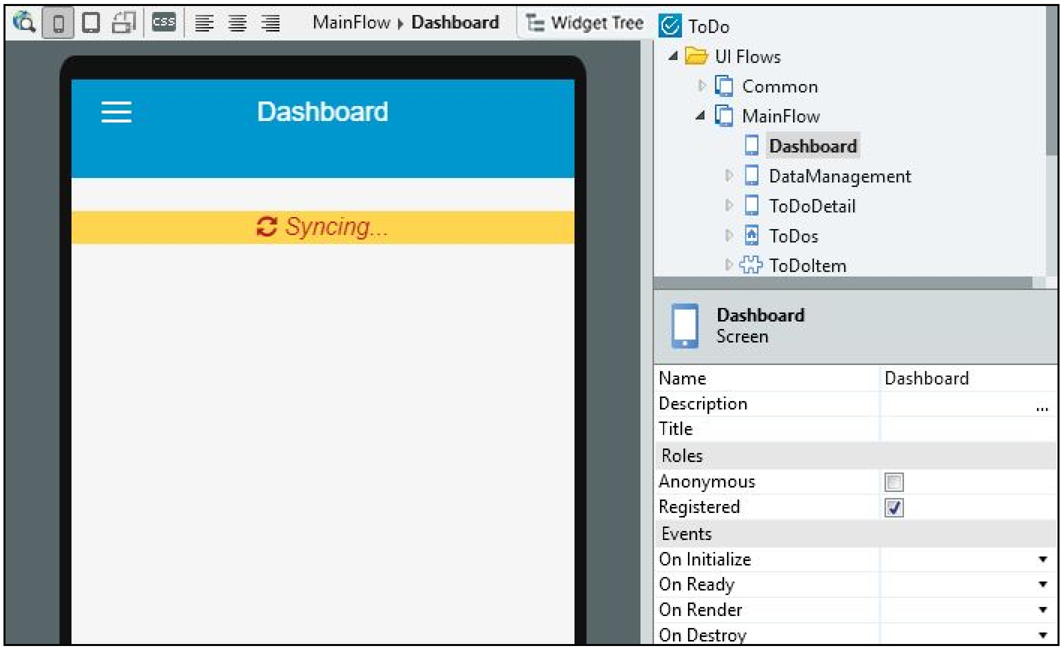
In this section of the exercise, we will create the final Screen of our application, the *Dashboard* Screen. This Screen will display two different charts: one to display the ToDos by priority and one to display the ToDos by category. This Screen will also be part of the menu and bottom bar, with links to access it directly.

1) Create a new Screen to the application using the **Empty** Template. Set its **Name** to *Dashboard* and its access to every **Registered** user. On the Screen itself, the **Title** should display *Dashboard*.

    a) Under the Interface tab, add a new Screen to the MainFlow.

    b) In the New Screen dialog, select the Empty Template and set the Name of the Screen to Dashboard.

    c) On the Screen **Title** Placeholder, type *Dashboard*. At this point, we should have something similar to the following screenshot

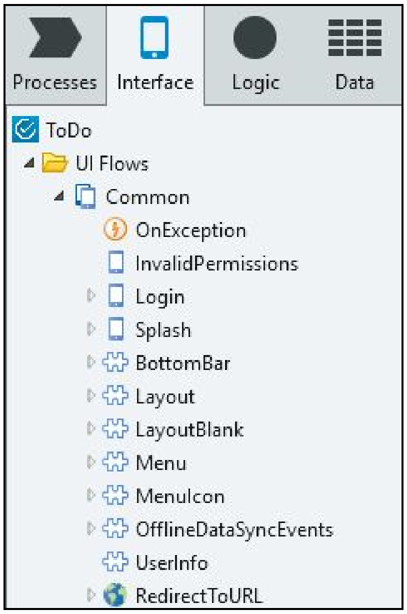


Dashboard
Screen

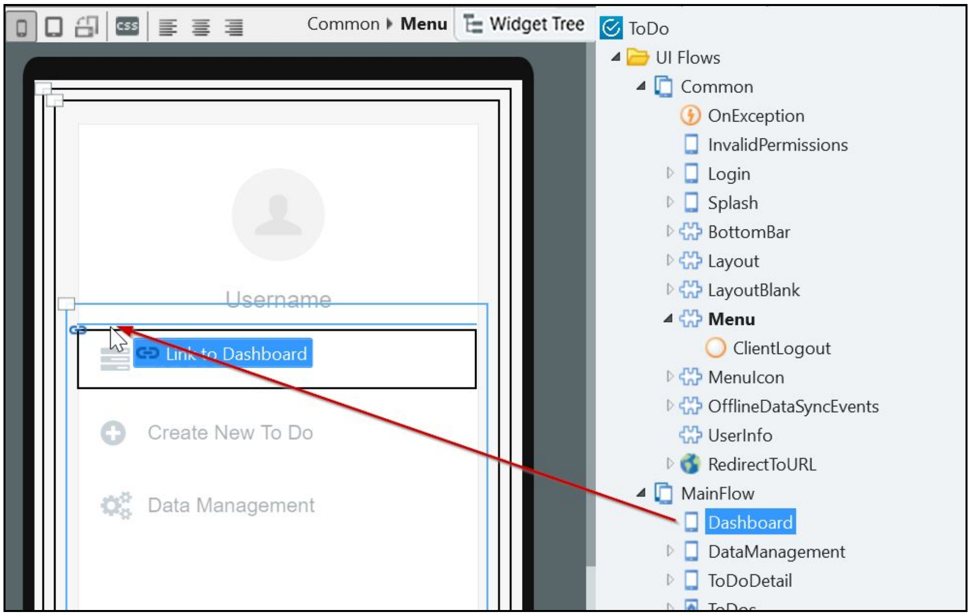
Name	Dashboard
Description	...
Title	
Roles	
Anonymous	□
Registered	☑
Events	
On Initialize	▾
On Ready	▾
On Render	▾
On Destroy	▾

2) Add a Link to the application **Menu** and to the **Bottom Bar** to access the **Dashboard** Screen.

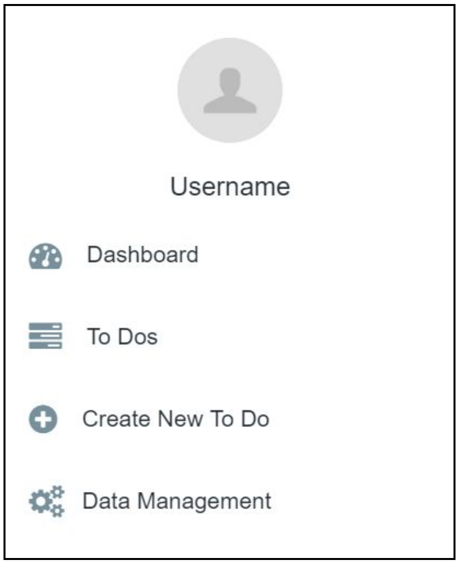
    a) Under the Interface tab, expand the **Common** flow.



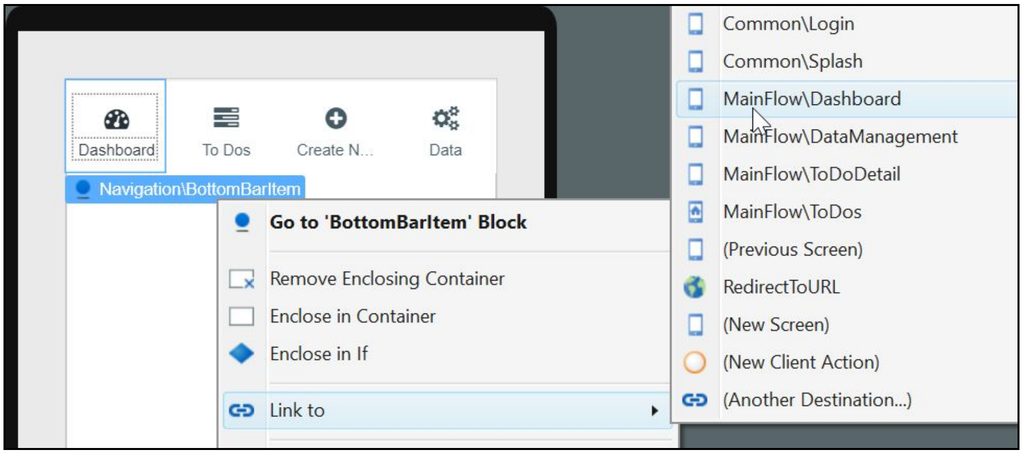
b) Open the Menu Block, drag the Dashboard Screen and drop it just above the ToDos Menu option.



c) Drag an icon from the widget toolbar and drop it right before the Dashboard text. Select the **dashboard** icon.



d) Open the **BottomBar** Block, select the **BottomBarItem** for the Dashboard. Right-click on the item and select *Link to* > *MainFlow\Dashboard*.

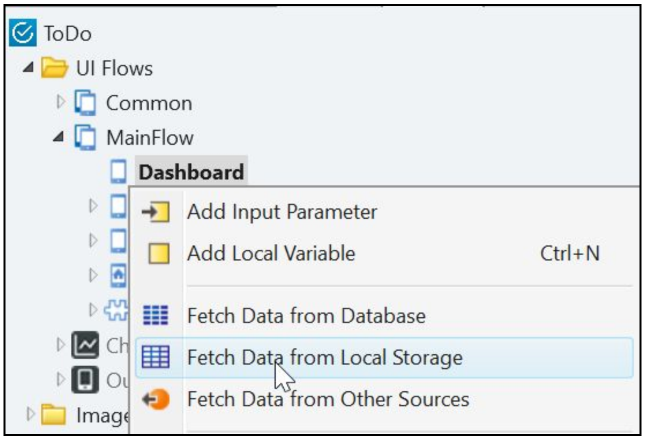


Add a Donut Chart to the Dashboard Screen

In this part of the exercise, we will add a first **Donut Chart** to the **Dashboard** Screen. This Chart will display the number of ToDos per priority. To get that information, we need a new Aggregate on the Screen.

1) Add a new **Aggregate** to the Screen that fetches the number of LocalToDos, grouped by their priority. Set the **Name** of the Aggregate to *GetToDosWithPriorities*.

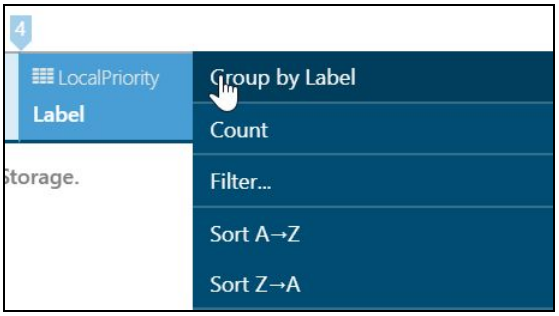
   a) Right-click the **Dashboard** Screen and choose *Fetch Data from Local Storage* to add a new Aggregate.



   b) From the Data tab, drag the **LocalToDo** Entity and drop it inside the Aggregate editor, then repeat the same for the **LocalPriority** Entity.

   c) Change the **Name** of the Aggregate to *GetToDosWithPriorities*

   d) Place the mouse over the LocalPriority **Label** attribute, then click the arrow to expand the context menu and choose *Group by Label* in the context menu.



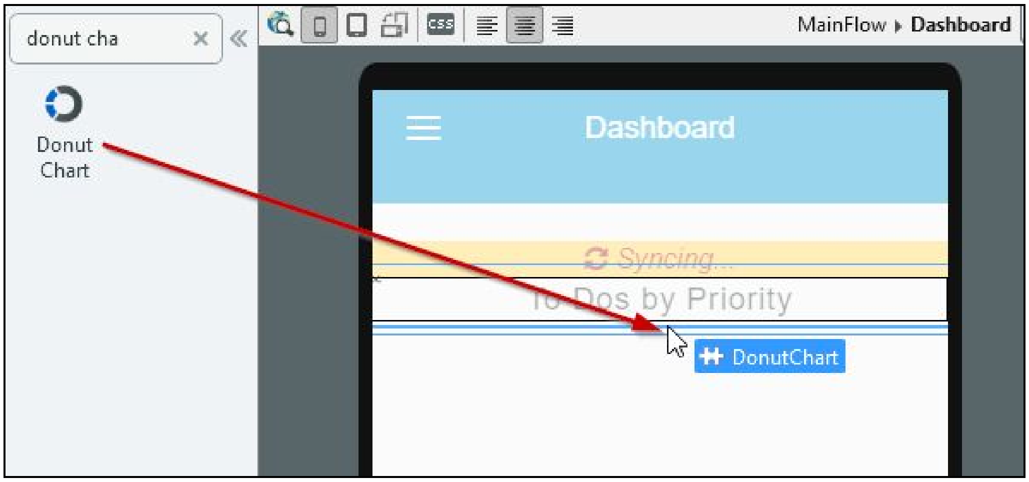
   e) Open the context menu of the LocalToDo **Title** attribute, and choose the *Count*. This will count the number of LocalToDos, by priority, since it will apply the Count after the Group By priority was made.

2) Add a **Donut Chart** to the Dashboard Screen and set the **Data Points** to the values returned by the **GetToDosWithPriorities** (Label and Count). Above the chart the Screen should display the text *To Dos by Priority*, with *heading3* style.

   a) Drag a **Container** widget to the main area of the Screen and align it to center.

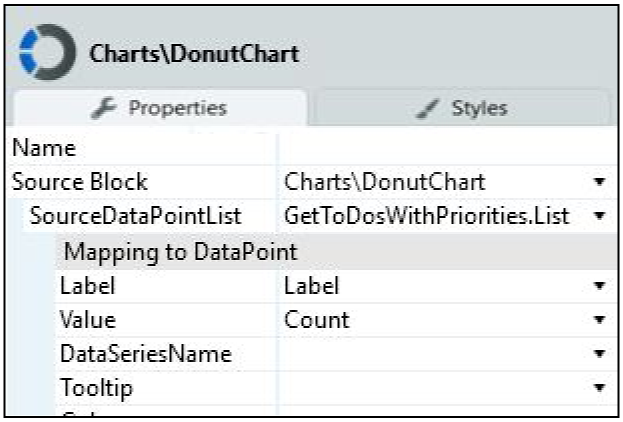
   b) Inside the **Container** type *To Dos by Priority*, then apply the *"heading3"* **Style Class** to the text.

   c) Drag a **Donut Chart** and drop it below the text, but still inside the surrounding Container.



   d) In the properties, set the **SourceDataPointList** parameter to *GetToDosWithPriorities.List*. This will set the source of data of the Chart to be the Aggregate defined in the previous step.

   e) In the **Mapping to Data Point**, set the **Label** to *Label* and the **Value** to *Count*. These are the two attributes returned from the Aggregate. The first one will be used for the Label of the Chart, which will be the priority (High, Medium or Low), while the second one will be how many ToDos exist for each category.


Charts\DonutChart

Name	
Source Block	Charts\DonutChart
SourceDataPointList	GetToDosWithPriorities.List
Mapping to DataPoint	
Label	Label
Value	Count
DataSeriesName	
Tooltip	

3) Publish the module and test the **Donut Chart** on the **Dashboard** Screen.

   a) Click the **1-Click Publish** Button to publish the module to the server.

   b) Open the application in the browser or device and navigate to the Dashboard Screen, using the Menu or the Bottom Bar.

   c) Make sure that the Donut Chart appears and that its values match the ToDos that exist in Local Storage.

Add a Bar Chart to the Dashboard Screen

To finish the Dashboard Screen, we will add an additional chart to display the ToDos by Category. In this case, it will be a Bar Chart. To get the information needed, we will need a new Aggregate on the Screen.

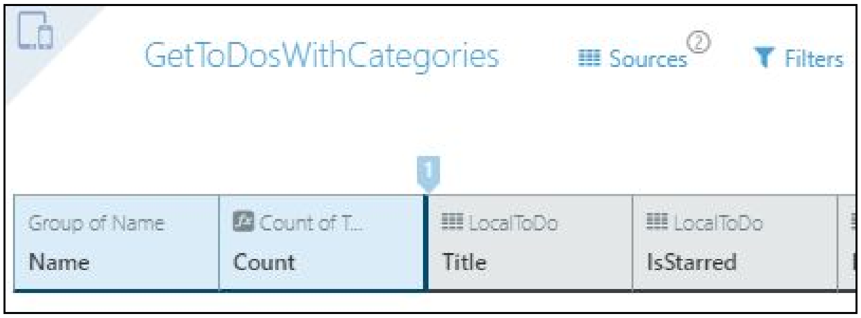
1) Add a new **Aggregate** to the Screen that fetches the number of LocalToDos, grouped by their category. Set the **Name** of the Aggregate to *GetToDosWithCategories*.

    a) Right-click the **Dashboard** Screen and choose *Fetch Data from Local Storage* to create a new Aggregate.

    b) Drag the **LocalToDo** and **LocalCategory** Entities into the Aggregate editor.

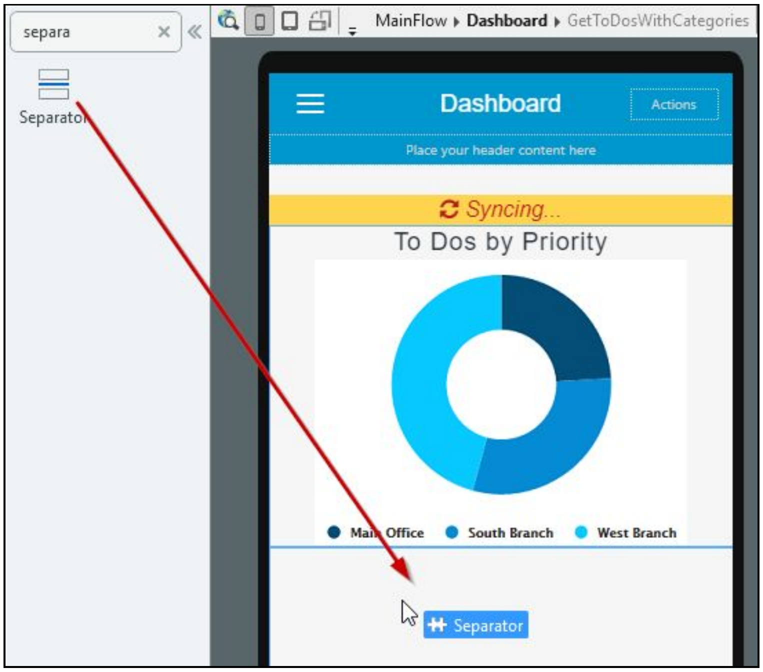
    c) Create a *Group By* of the LocalCategory **Name** attribute, and a *Count* by the LocalToDo **Title** attribute.

    d) The Aggregate should look like this at the end



2) Add a **Bar Chart** to the Dashboard Screen and set the **Data Points** to the values returned by the **GetToDosWithCategories** (*Name* and *Count*). Above the chart, the Screen should display the text *To Dos by Category*, with *heading3* style. Add a Separator between the two Charts to create some distance between them.

    a) Drag a **Separator** widget and drop it below the Container that has the Donut Chart.

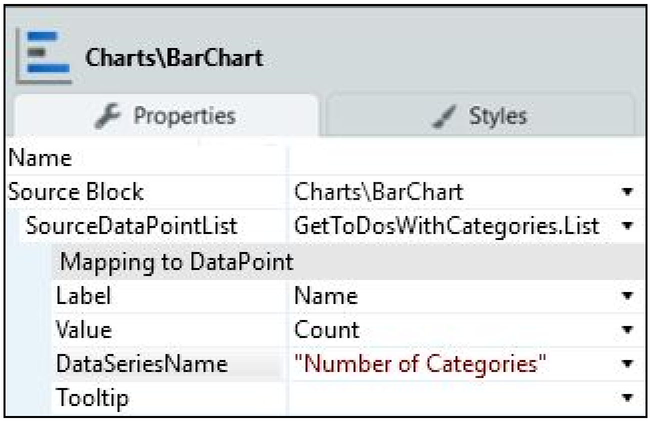


b) Drag a **Container** widget and drop it below the **Separator** and align it to center.

c) Inside the Container, type *To Dos by Category* and then apply the *"heading3"* **Style Class** to the text.

d) Drag a **Bar Chart** widget and drop it below the text written above, but still inside the Container.

e) Set the **SourceDataPointList** to use the list from the Aggregate created above, then define the mapping as below. Set the **DataSeriesName** to *"Number of Categories"*. This information will appear below the Chart.



Name	
Source Block	Charts\BarChart
SourceDataPointList	GetToDosWithCategories.List
Mapping to DataPoint	
Label	Name
Value	Count
DataSeriesName	"Number of Categories"
Tooltip	

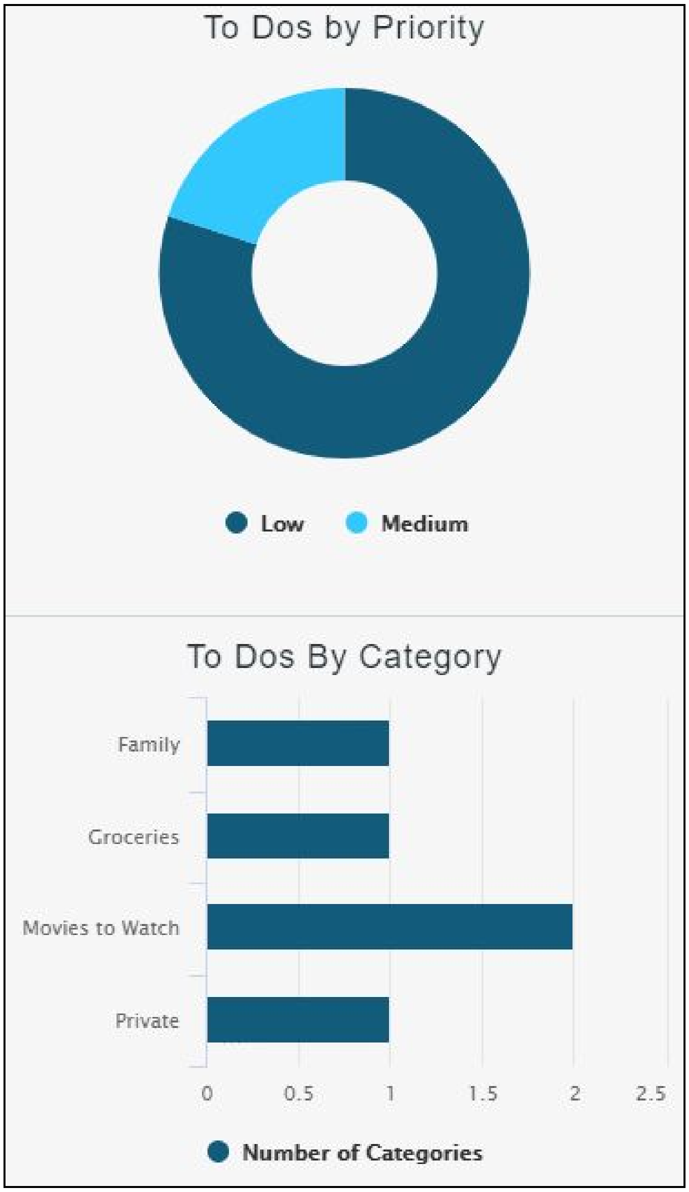
3) Publish the module and test the **Bar Chart** on the **Dashboard** Screen.

a) Click the **1-Click Publish** Button to publish the module to the server.

b) Open the application in the browser or device and navigate to the Dashboard Screen, using the Menu or the Bottom Bar.

c) Make sure that the Bar Chart appears and that its values match the ToDos that exist in Local Storage.

d) At the end of this exercise lab, the Screen should look like the following screenshot



End of Lab

In this exercise lab, we created a Dashboard Screen with two Charts.

The first Chart displays the number of To Dos, by their priority, while the second displays the number of To Dos, by their category. To accomplish this we used a Donut Chart and a Bar Chart. The Charts are part of the Charts Mobile component, and are available in the Widget Toolbar.

The Menu and Bottom Bar were also updated to include links to the Dashboard Screen.