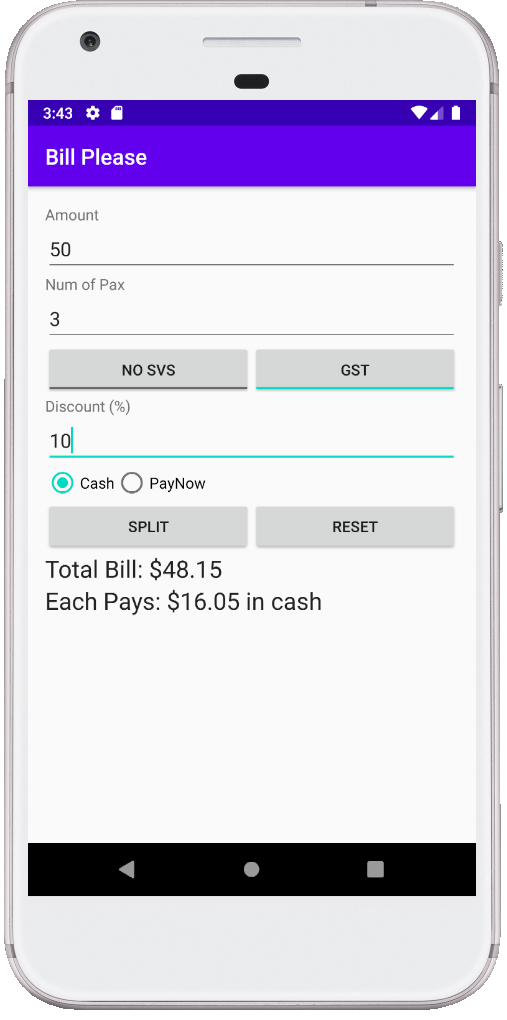
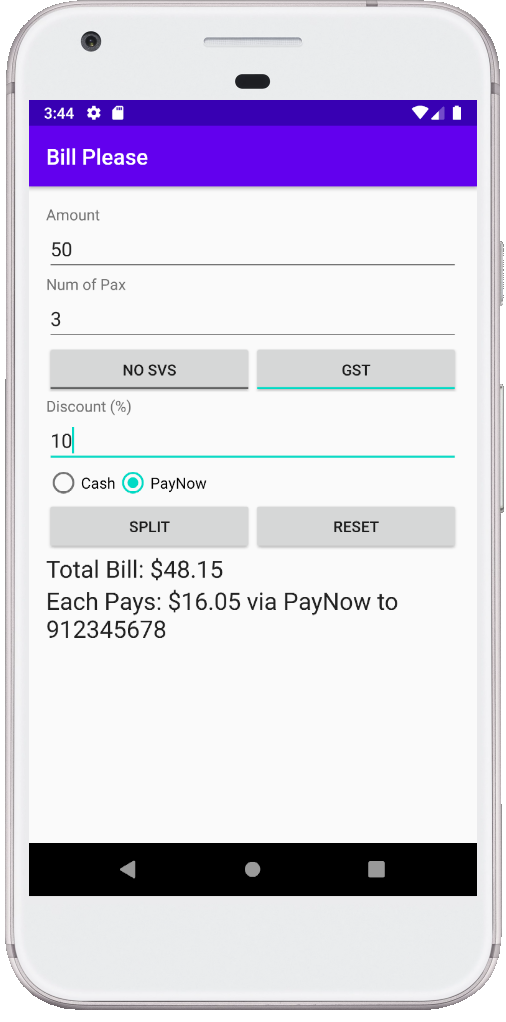
|  |
| --- |
| **Session 2 Worksheet** |

**Exercise – The Bill Calculator**

Design and create a bill calculator that helps to split the bill among a group of people. A typical bill will add on service charge (10%) and GST (7%).

**Note:** Your design can be different from the one given, but should include the required information and be user friendly.

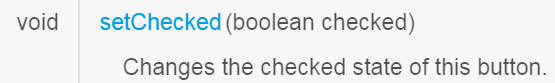
Your Bill Calculator should have the following features:

* Enter the original amount of the bill and number of people in the group
* Choose whether to add on service charge(10%) and GST(7%)
* Calculate and display
  + the total bill with or without the service charge and/or GST depending on the user option
  + how much each person has to pay
* Display the PayNow phone number with the amount to pay if the PayNow RadioButton is checked
* Reset the app by clearing the text fields (original amount and number of people) and reset the TextViews (total bill and how much each person has to pay) to the original text

**Enhancements**

Enhance user experience by improving on the behaviour of your app.

* Basic validation should be handled.
  + If any input is empty, display an error message
  + Calculation should only be performed for valid amount and number of people. Display an appropriate message if any input is invalid
  + Display the message in red
* Cater for discount (%)
* Display numerical keyboard instead of full keyboard for Amount and Number of People
* When the reset button is clicked, the two ToggleButtons for Service Charge and GST should be reset to default unchecked stated. See setChecked()



The Cash RadioButton should be checked. Use RadioGroup method check(int radiobuttonid)

**Getting started**

In Android Studio, create a new project with the following information:-

|  |  |
| --- | --- |
| **Project Template** | Empty Activity |
| **Application Name** | Bill Please |
| **Package Name** | sg.edu.rp.c346.id<your student ID>.billplease |
| **Project Location** | D:\C346\Workspace\P03\BillPlease |
| **Language** | Java |
| **Minimum API Level** | API 16 |

**Design the layout**

Based on the requirements given, do a quick sketch on paper on how you would like the User Interface (UI) of the application to look like. Fill in the table below with the UI elements and the number needed:

|  |  |  |
| --- | --- | --- |
| **Purpose** | **UI Element** | **Number of UI elements needed** |
| For labelling amount and number of pax respectively | TextView | 2 |
| For entering amount and number of pax respectively | TextView | 2 |
| For indicating whether to apply service charge and GST respectively | Toggle Button | 2 |
| For displaying total bill and the amount each pay respectively | TextView | 1 |
| For selecting the payment mode like cash or PayNow | Radio Button | 2 |
| For splitting the bill and resetting the text fields respectively | Button | 2 |
| You may add more here… |  |  |

Before you move on to the Java coding, remember to

* assign proper IDs to the UI elements
* use string resources to display all the texts

Tip 1: To clear the text of an EditText, think about how to change the text of a TextView. The text of the EditText can be changed similarly.

Tip 2: As you try out the application, the split amount of the final bill may come out with a long decimal point, e.g. $16.677777777777. You may use the Java String.format() method to format the sums to 2 decimal points.

E.g.

* 1. String val = String.format("%.2f", (1.0/3.0));
  2. val 🡪 "0.33"

Tip 3: Remember to convert data from EditText to numerical format.

E.g. Use Integer.parseInt()

**Your screenshot:**

|  |
| --- |
|  |

**activity\_main.xml**

|  |
| --- |
|  |

**strings.xml**

|  |
| --- |
|  |

**MainActivity.java**

|  |
| --- |
|  |

**GitHub URL**

|  |
| --- |
|  |