

Mobile App Development

In-Class Assessment 2

Basic Instructions:

1. This is an In Class Assessment, which will count for 8% of the total course grade.
2. This assessment is an individual effort. Each student is responsible for her/his own assessment and its submission.
3. Once you have picked up the assessment, you may not discuss it in any way with anyone until the assessment period is over.
4. During the assessment, you are allowed to use the course videos, slides, and your code from previous home works and in class assignments. You can use the internet to search for answers. You are NOT allowed to use code provided by other students or solicit help from other online persons.
5. Answer all the assessment parts, all the parts are required.
6. During the assessment the teaching assistants and Instructors will pass by each student and ask them to demonstrate their application. Your interaction with the teaching assistants and instructors will be taken into consideration when grading your assessment submission.
7. Please download the support files provided with the assessment and use them when implementing your project.
8. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will lose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
9. Create a zip file which includes all the project folder, any required libraries, and your presentation material. Submit the exported file using the provided canvas submission link.
10. **Do not try to use any Social Messenger apps, Emails, Or Cloud File Storage services in this exam.**
11. **Failure to follow the above instructions will result in point deductions.**
12. **Any violation of the rules regarding consultation with others will not be tolerated and will result disciplinary action and failing the course.**

In-Class Assessment 2 (100 Points)

In this assignment you will build a simple bill management app. You are provided with the skeleton app that includes the basic activities and fragments that need to be implemented in this project.

Setup:

1. Unzip the provided zip file and open the provided project in Android Studio.
2. All communication between fragments and all fragment transitions should be performed through their hosting activities through interfaces that you should define and implement.

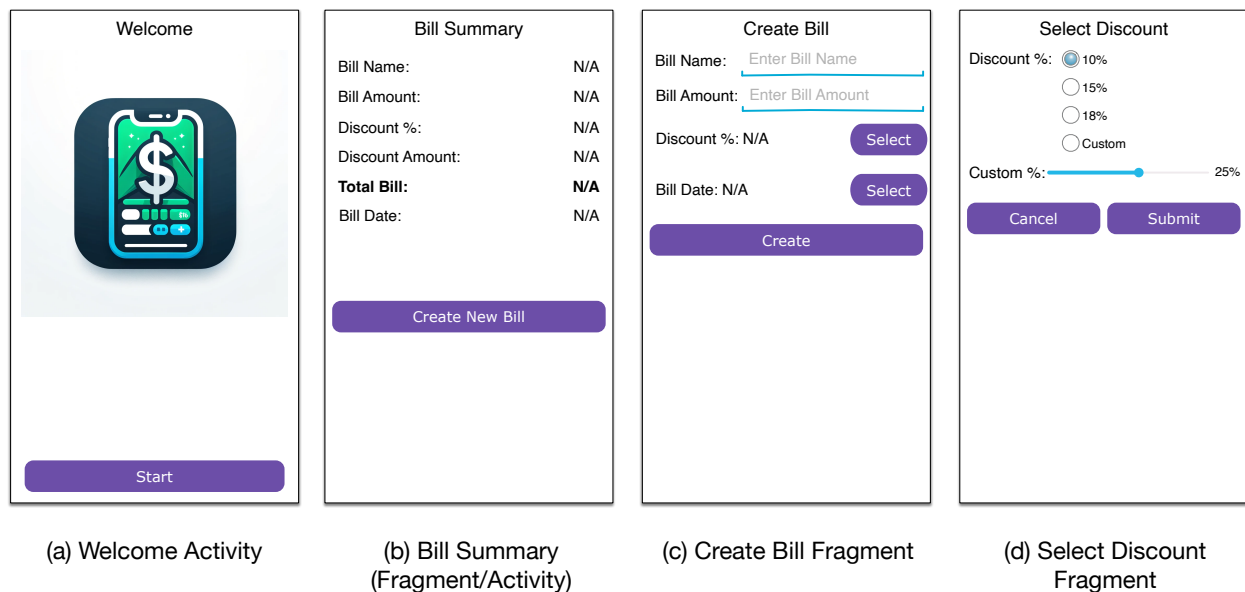


Figure 1, App Wireframe

Part 1 (5 Points): Welcome Activity:

This activity displays the welcome screen as shown in Figure 1(a).

1. This is the launcher activity, it displays the banner photo as shown in Figure 1(a).
2. Clicking the "Start" button should
 - a. Start the "Bill Summary" Activity. Then finish this activity.

Part 2 (30 Points): Bill Summary Activity/Fragment

The Bill Summary Activity should host the Bill Summary Fragment as shown in Figure 1(b). The requirements are listed below:

1. This activity should load the "Bill Summary Fragment" as shown in Figure 1(b).
2. **In the fragment,** Clicking "Create New Bill" Button should:
 - a. **Launch the Create Bill Activity for Result.** This can be done through the activity though an interface or directly from the fragment. You are free to pick one of these approaches.
 - b. Upon returning from "Create Bill Activity" and receiving the result as the newly created Bill object.
 - The newly received Bill object, should be sent to the Bill Summary Fragment and should be displayed as shown in Figure 2(c).

Note that: The Create Bill Activity hosts the Create Bill Fragment, Select Discount Fragment, and the Select Bill Date Fragment.

Part 3, Create Bill Fragment (30 Points):

The Create Bill Activity should load this fragment when it loads. This fragment is shown in Figure 1(c). The requirements are listed below:

1. The fragment requests the bill amount, discount and bill date. You are provided with a Bill class to hold these attributes, the class implement Serializable interface.
2. Clicking the “Select” Discount button should:
 - a. Using an interface, communicate with the Create Bill Activity to replace this fragment with the Select Discount Fragment.
 - b. Push the current fragment on the back stack.
 - c. Upon returning from the Select Discount Fragment, the received discount value should be displayed as shown in Figure 2(b).
3. Clicking the “Select” Bill Date button should:
 - a. Using an interface, communicate with the Create Bill Activity to replace this fragment with the Select Bill Date Fragment.
 - b. Push the current fragment on the back stack.
 - c. Upon returning from the Select Bill Date Fragment, the received bill date value should be displayed as shown in Figure 2(b).
4. Clicking the “Create” button should:
 - a. If any of the inputs or selections are missing, then show a Toast message indicating that the corresponding input is required.
 - b. If all the inputs and selections are correctly provided, then perform the following:
 - i. Create a Bill object initialized with the provided input.
 - ii. **Using an interface, communicate with the Create Bill Activity to send back the newly created Bill object as a result to the Bill Summary Activity/Fragment. Then finish this Activity.**

Part 4 (20 Points): Select Discount Fragment

This fragment is shown in Fig 1(d). The requirements are listed below:

1. Use a RadioGroup containing RadioButtons to enable the user to pick from the discount options 10%, 15%, 18% and Custom. When the user picks “Custom” from the RadioGroup, then the SeekBar progress is used as the custom discount value.
2. The SeekBar should be setup as follows:
 - a. Use the SeekBar to enable the user to pick a custom discount value, the maximum value should be set to 50%.
 - b. When the fragment starts the custom discount value should be set to 25%.
 - c. On the right of the SeekBar use a TextView to display the current custom discount progress of the SeekBar, which represents the current custom discount value.
 - d. Setup the SeekBar event handler such that if it is changed the current custom discount progress should be updated to show the SeekBar progress.
3. Clicking the “Submit” button should:
 - a. If a selection is not made, then show a Toast message indicating that a selection is required.

- b. If the selection is correctly provided then using an interface, communicate with the Create Bill Activity:
 - 1) Send the selected discount to the Create Bill Activity using the interface.
 - 2) Find the Create Bill fragment by tag, send it the received discount value.
 - 3) Pop the back stack, which should display the Create Bill fragment which should display the received discount value.
4. Clicking the “Cancel” button should:
 - a. Using an interface, communicate with the Create Bill Activity to pop the back stack, which should display the Create Bill fragment.

The figure displays three mobile app wireframes. Fragment (a) is a date picker titled 'Bill Date' with a header 'Select the bill date:'. It shows the year '2024' and the selected date 'Sat, Jun 1'. Below is a calendar grid for June 2024, with the 1st highlighted. At the bottom are 'Cancel' and 'Submit' buttons. Fragment (b) is titled 'Create Bill' and contains form fields for 'Bill Name' (Apt Rent), 'Bill Amount' (200), 'Discount %' (10%), and 'Bill Date' (06/09/2024). Each field has a corresponding 'Select' button. A 'Create' button is at the bottom. Fragment (c) is titled 'Bill Summary' and displays the entered values: 'Bill Name: Apt Rent', 'Bill Amount: \$200.00', 'Discount %: 10%', 'Discount Amount: \$20.00', 'Total Bill: \$180.00', and 'Bill Date: 06/09/2024'. A 'Create New Bill' button is at the bottom.

(a) Select Bill Date Fragment

(b) Create Bill Fragment

(c) Bill Summary Fragment

Figure 2, App Wireframe

Part 5 (15 Points): Select Bill Date Fragment

This fragment is shown in Fig 2(a). The requirements are listed below:

1. The user is able to select the bill date using the DatePicker, **make sure to setup the maximum date to today.**
2. Clicking the “Submit” button should:
 - a. If no selection, then show a Toast message indicating that a selection is required.
 - b. If the selection is correctly provided then using an interface, communicate with the Create Bill Activity:
 - 1) Send the selected date value to the Create Bill Activity using the interface.
 - 2) Find the Create Bill fragment by tag, send it the received received date value.
 - 3) Pop the back stack, which should display the Create Bill fragment which should display the received date value.
3. Clicking the “Cancel” button should:
 - a. Using an interface, communicate with the Create Bill Activity to pop the back stack, which should display the Create Bill fragment.

Section:	
Student Name:	
Student ID:	

Step #	Features	Total	Grade
1	(Part 1) Welcome Activity: Button click starts the Bill Summary Activity and finishes this activity.	5	
2	(Part 2) Bill Summary Fragment: UI and event handlers. Clicking the “Create New Bill” button should <i>launch the “Create Bill Activity” for Result.</i>	10	
3	(Part 3) Create Bill Fragment: UI and event handlers. Used interface to transition to the required two Select fragments.	10	
4	(Part 4) Select Discount Fragment: UI, event handlers and interface implemented as listed in the requirements. Uses an interface to send back the selected value. Selected value visible in the Create Bill Fragment.	20	
5	(Part 5) Select Bill Date Fragment: UI, event handlers and interface implemented as listed in the requirements. Uses an interface to send back the selected value. Selected value visible in the Create Bill Fragment.	15	
6	(Part 3) Create Bill Fragment: Display the selected values correctly, and validation implemented. <i>Uses interface to communicate with the activity and send the Result back to “Bill Summary Activity”. Then finishes the current activity.</i>	20	
7	(Part 2) Bill Summary Fragment: Receives the new Bill object as a launcher result, then displays the received Bill object correctly.	20	
	Total	100	
	Table 1: Grading Key		