Mobile App Development
In-Class Assessment 3

Basic Instructions:

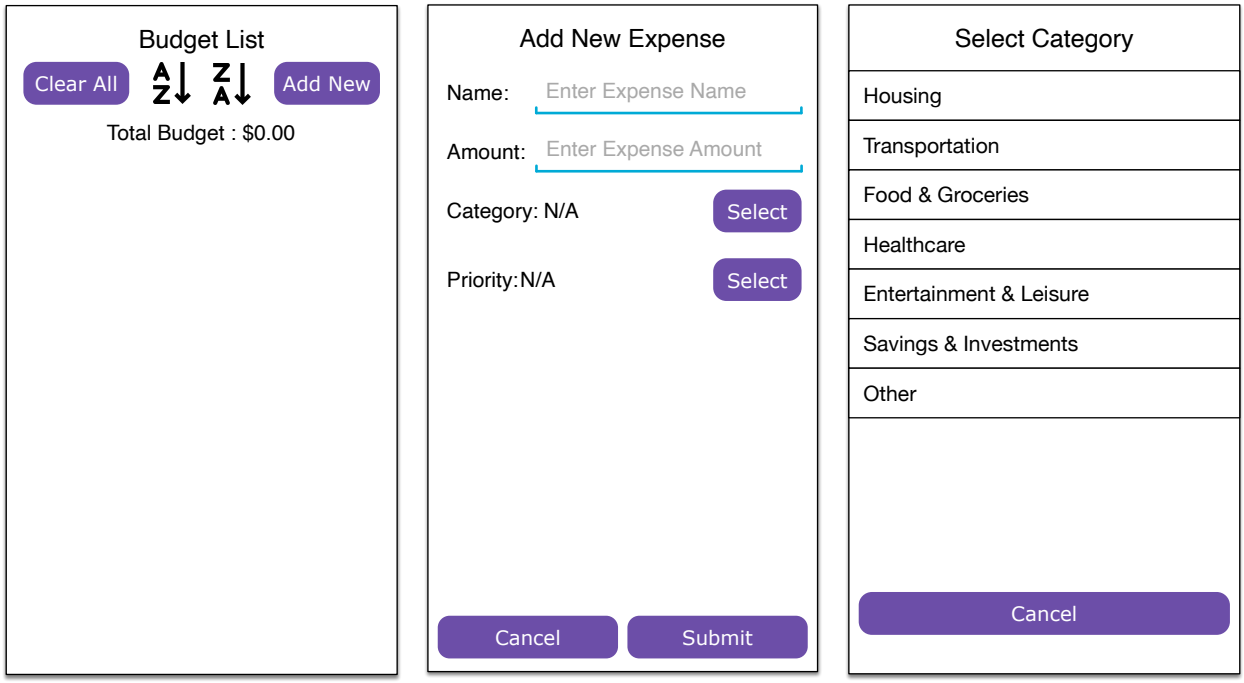
1. This is and In Class Assessment, which will count for 8% of the total course grade.
2. This assessment is an individual effort. Each student is responsible for her/his own assessment and its submission.
3. Once you have picked up the assessment, you may not discuss it in any way with anyone until 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 loose 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 3 (100 Points)

In this assignment you will build as simple budgeting app. You are provided with the skeleton app that includes the basic app flow and layouts.

Setup:

1. Unzip the provided zip file and open the provided project in Android Studio.
   a. Create a new project named “Evaluation03”
   b. The package name is “edu.uncc.evaluation03”
   c. Unzip the provided source zip folder, and move the unzipped folder “src” folder to replace the “src” in your project.
2. All communication between fragments and all fragment transitions should be performed through their hosting activities through interfaces.
3. **The ArrayList of Expense is stored and managed in the Main Activity.**



(a) Budget List Fragment (b) Add Expense Fragment (c) Select Category Fragment

Figure 1, App Wireframe

Part 1 (40 Points): Budget List Fragment

This fragment is shown in Figures 1(a) and 2(b). The requirements are listed below:

1. Use an interface to receive the ArrayList of Expenses from the Main Activity.
2. Display the Total Budget, which its the sum of all expenses as shown in Figure 2(b).
3. Display the list of Expenses using a RecyclerView as shown in Fig 2(b).
   a. Clicking on the trash icon should:
      - Delete the selected expense from the list of expenses in the Fragment and use and interface to delete the expense item from the list in the Main Activity.
      - Reload the RecyclerView to display the latest expenses.
      - Update the Total Budget displayed at the top.

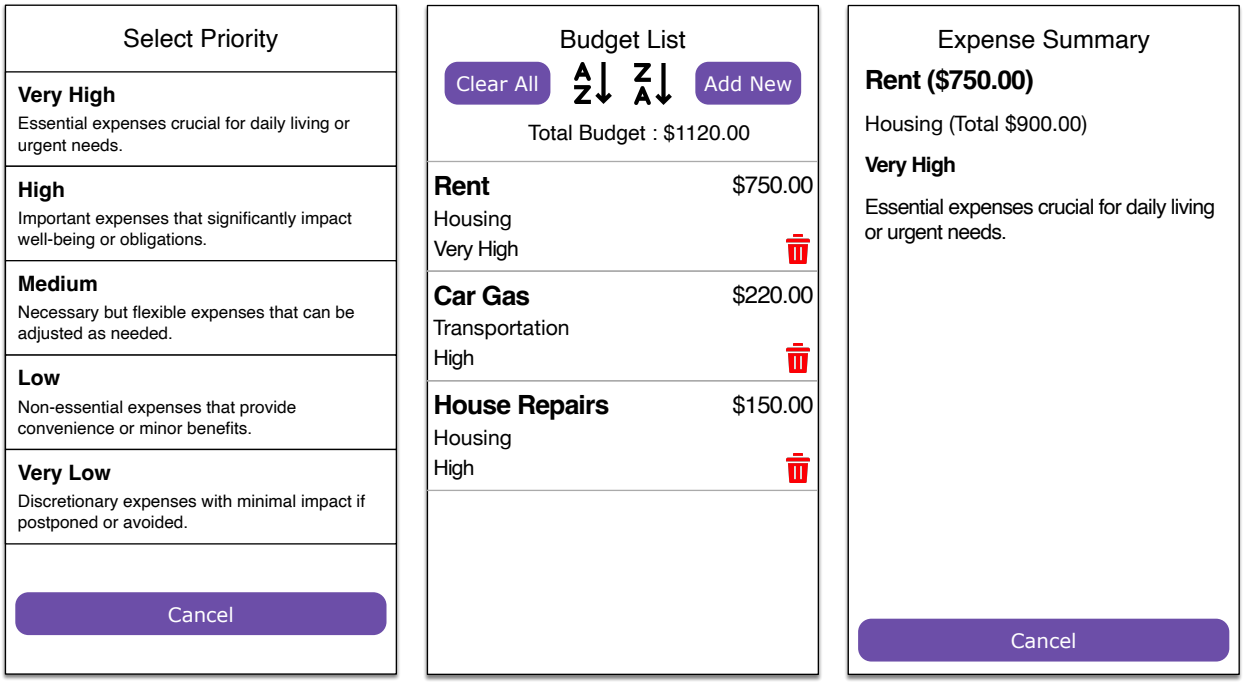
b. Clicking on a row item should:
   - Using an interface, send the selected Expense item to the Main activity.
   - Replace this fragment with the Expense Summary Fragment.
   - Push the current fragment on the back stack.

4. Clicking “Add New” button should:
   a. Using an interface, communicate with the Main Activity to replace this fragment with the Add Expense Fragment.
   b. Push the current fragment on the back stack.
   c. Upon returning from the Add Expense Fragment, the displayed list and total should updated based on the latest expenses ArrayList that is received from the Main Activity.
5. Clicking “Clear All” button should:
   a. Using an interface, communicate with the Main Activity to clear the ArrayList of Expenses.
   b. Reload the RecyclerView to display the latest expenses.
   c. Update the Total Budget displayed at the top.
6. Clicking Sort ASC (DESC) button should:
   a. Sort the ArrayList of Expenses in ascending (depending) ordered by amount.
   b. Reload the RecyclerView to display the sorted expenses.

Part 2 (10 Points): Add Expense Fragment

This fragment is shown in Figure 1(b). The requirements are listed below:

1. The fragment requests the expense name, amount, category and priority. You are provided with a Expense class to hold these attributes, the class implement Serializable interface.
2. Clicking the “Select” Category (Priority) button should:
   a. Using an interface, communicate with the Main Activity to replace this fragment with the Select Category (Priority) Fragment.
   b. Push the current fragment on the back stack.
   c. Upon returning from the Select Category (Priority) Fragment, the received value should be displayed.
3. Clicking the “Submit” 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:
      - Create an Expense object initialized with the provided input.
      - Using an interface, communicate with the Main Activity to send back the newly created Expense object, which should be added to the Array List of Expenses hosted in the Main Activity.
      - Pop the back stack.



(a) Select Priority Fragment (b) Budget ListFragment (c) Expense Summary Fragment

Figure 2, App Wireframe

Part 3 (10 Points): Select Category Fragment

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

1. Setup the required adapter and ListView to display the categories.
2. Upon selecting a category from the ListView should:
   a. Communicate with the Main Activity:
      - Send the selected category value to the Main Activity using the interface.
      - Find the Add Expense fragment by tag, send it the received category value.
      - Pop the back stack, which should display the Add Expense fragment which should display the received category value.
3. Clicking the "Cancel" button should:
   a. Using an interface, communicate with the Main Activity to pop the back stack, which should display the Add Expense fragment.

Part 4 (20 Points): Select Priority Fragment

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

1. Setup the required adapter and ListView to display the priorities. ***You are required to create a custom adapter***.
2. Upon selecting a category from the ListView should:
   a. Communicate with the Main Activity:
      - Send the selected priority value to the Main Activity using the interface.
      - Find the Add Expense fragment by tag, send it the received category value.
      - Pop the back stack, which should display the Add Expense fragment which should display the received priority value.
3. Clicking the "Cancel" button should:
   a. Using an interface, communicate with the Main Activity to pop the back stack, which should display the Add Expense fragment.

Part 5 (20 Points): Expense Summary Fragment

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

1. This fragment receives the required information from the Budget List Fragment and the Main Activity.
2. The fragment displays the name, amount, priority, and priority description of the selected Expense item.
3. In addition, the fragment displays the total amount of expenses that match the selected item's category. For example, in Figure 2(c) the total "Housing" category expenses are $900.00.
4. Clicking the "Cancel" button should:
    a. Using an interface, communicate with the Main Activity to pop the back stack, which should display the Budget List fragment.

Student Name:	
Student ID:	

#	Features	Total	Grade
1	**(Part 3) Select Category Fragment:** - ListView and adapter setup. - Sends selected item using interface to the New Expense Fragment.	10	
2	**(Part 4) Select Priority Fragment:** - ListView and custom adapter setup. - Sends selected item using interface to the New Expense Fragment.	20	
3	**(Part 2) Add Expense Fragment:** - Input validation, and sends new Expense object to the Main Activity. - Adds new Expense object to the ArrayList hosted in the Main Activity	10	
4	**(Part 1) Budget List Fragment:** - Retrieves the Array List of Expenses from the Main Activity using the interface. - RecyclerView and Custom Adapter. - Total Budget displayed.	20	
5	**(Part 1) Budget List Fragment:** - Clicking trash icon, deletes the expense from the local list and the list stored in Main Activity. - Reload the RecyclerView to display the update expenses. - Total Budget updated.	5	
6	**(Part 1) Budget List Fragment:** - Clicking 'Clear All', deletes all the expenses from the local list and the list stored in Main Activity. - Reload the RecyclerView to display the update expenses. - Total Budget updated.	5	
7	**(Part 1) Budget List Fragment:** - Sort ASC (DESC), sorts the local list of expenses. - Reload the RecyclerView to display the update expenses.	10	
8	**(Part 5) Expense Summary Fragment:** - Displays the expense item. - Calculate and display the sum of expenses for item's category	20	
	Total	**100**	
	Table 1: Grading Key		