

ITIS/ITCS 4180/5180 Mobile Application Development
Homework 2

Basic Instructions:

1. In every file submitted you **MUST** place the following comments:
 - a. Assignment #.
 - b. File Name.
 - c. Full name of all students in your group.
2. Each group should submit only one assignment on behalf of all the other group members.
3. 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.
4. Export your Android project and create a zip file which includes all the project folder and any required libraries.
5. Submission details:
 - a. Compress the contents of your project folder. The file name is very important and should follow the following format: **HW2.zip**
 - b. Only one group member is required to submit on behalf of the whole group.
 - c. You should submit the assignment through Canvas: Submit the zip file.
- 6. Failure to follow the above instructions will result in point deductions.**

Homework 2 (100 Points)

In this assignment you will get familiar with Android Intents (Explicit/ Implicit), Data Passing between Intents (Parcelable/ Simple Data Passing), Starting activity for result and Alert Dialogs. This is a Expense Management App, where the user gets to enter expense information, the name, category and receipt. User should be able edit, display and delete Expenses. Main screen is displayed in Figure 1.

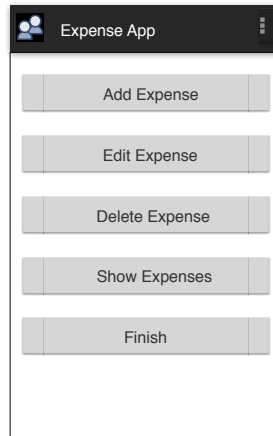


Figure 1, Main Activity

Main Activity (10 Points)

The Main Activity contains the 5 buttons arranged as displayed in Figure 1. The Main Activity should maintain a list of the current Expense Objects, the list should be shared with the other activities through intents when needed. The activity requirements are as follows:

- The expense list should be stored in the Main Activity, you should use an ArrayList to store the list of Expense Objects. The list should be passed using intent extras to other activities when needed, **you should not declare the list as static**.
- The Add Expense, Edit Expense, Delete Expense, Show Expenses, and Finish buttons, should start the Add Expense Activity, Edit Expense Activity, Delete Expense Activity, Show Expense Activity and Finish the activity respectively.

Add Expense Activity (20 Points)

The Add Expense activity is displayed in Fig 2. Implement the following requirements:

- When the user selects “Add Expense” button in the Main Activity, the Add Expense Activity should be started as shown in Figure 2.
- The Expense Name should not exceed 50 characters. The categories should be in a selection pane as in the figure. The amount should be in an appropriate format and the date should be selected using a DatePickerDialog. Any missing or incorrectly entered information the user should be alerted through Alert messages.
- Receipt should initially display a default browsing icon. The user has a choice to associate a receipt with the expense. Tapping browsing icon should enable user to get an image from the system Image Gallery App. This should start New Activity for result. The selected image should be returned back to the previous activity. Hint: for intent information check <https://developer.android.com/guide/components/intents-common.html>

- The image retrieved from the gallery should be displayed instead of the default browsing icon as shown in Figure 2.
- Clicking the Add Expense button should save the fields as a Expense object, send it back as a result to the Main Activity so that it can be stored in the Expense list and finish the Add Expense Activity.
- NOTE: To save the image selected, you should save the **Uri** object that points to the selected image, **do not store the full image bitmap**.

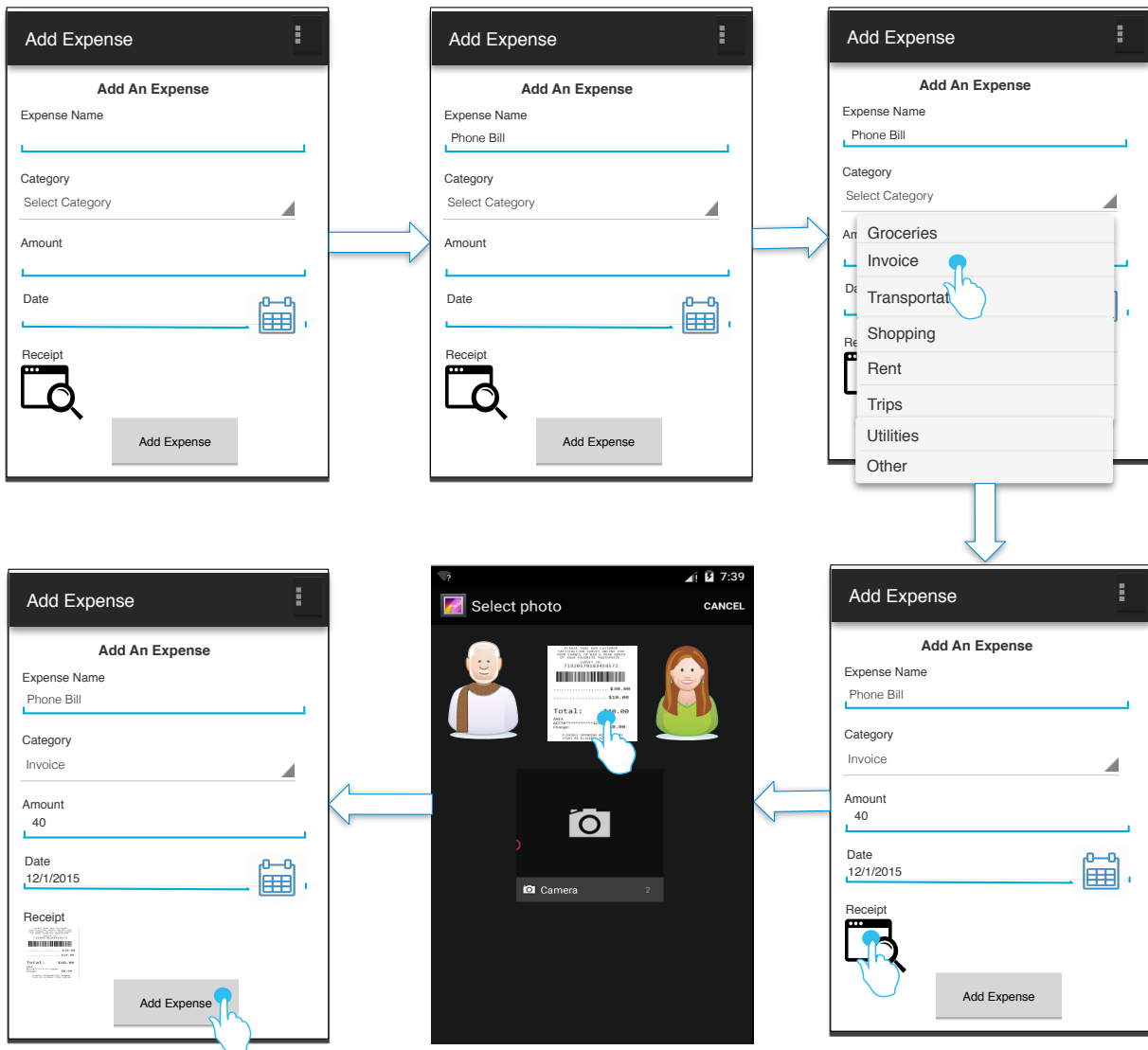


Figure 2, Add Expense Activity

Edit Expense Activity (25 Points)

The Edit Expense activity is displayed in Figure 3. Implement the following requirements:

- When the user selects “Edit Expense” button in the Main Activity, the Edit Expense Activity should be started as shown in Figure 3.
- Clicking the Select Expense button should show an alert dialog displaying the list of Expense names sorted in ascending order. The list enables the user to pick which

Expense to edit. Upon selecting an Expense from the list, the Expense information should be populated in the Edit Expense activity as shown in Figure 3.

- The UI for this activity is similar to the Add Expense Activity and should follow the same text size and implementation and restrictions.
- Clicking the Save button should save the fields as a Expense object and send it back as a result to the Main Activity so that it can be updated and stored in the Expense list.
- Clicking the cancel button should finish the edit expense activity and does not update the edited expense.

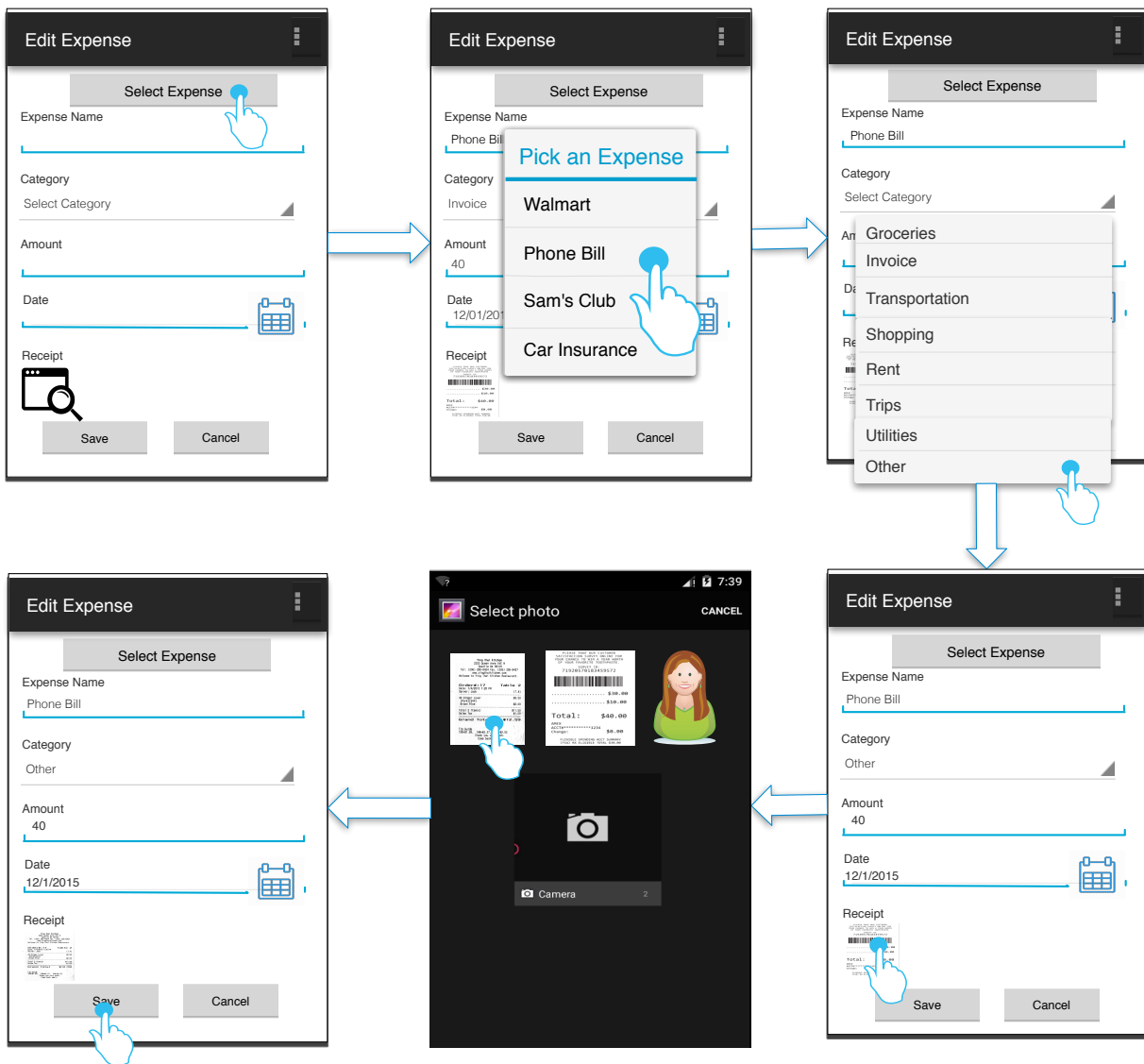


Figure 3, Edit Expense

Delete Expense Activity (25 Points)

The delete Expense activity is displayed in Figure 4. Implement the following requirements:

- When the user selects “Delete Expense” button in the Main Activity, the Delete Expense Activity should be started as shown in Figure 4.

- Clicking the Select Expense button should show an alert dialog displaying the list of expense names sorted in ascending order. The list enables the user to pick which expense to delete. Upon selecting an expense, the Expense information should be populated in the Delete Expense activity as shown in Figure 4.
- All the EditText components and ImageViews should **not** be clickable, and should only be used to present the current Expense's information.
- Clicking the Delete button should delete the current Expense and send the Main activity the required information in order to delete this Expense from the Expense list. Clicking the Cancel button should finish the Delete Expense activity and should not delete the Expense.

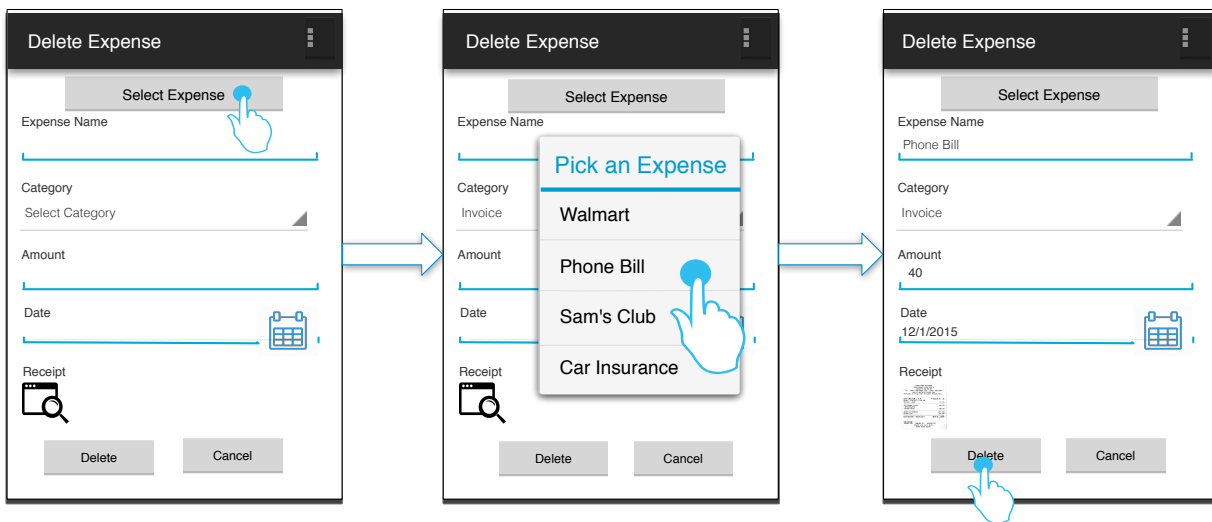


Figure 4, Delete Expense

Show Expenses Activity (20 Points)

The Show Expenses activity is displayed in Figure 5. Implement the following requirements:

- When the user selects “Show Expenses” button in the Main Activity, the Show Expenses Activity should be started as shown in Figure 5.
- This activity enables the user to view all the currently stored Expenses.
- The actions required from the each of the GUI images are listed in Table 1. Upon clicking finish the activity should be finished.



Figure 5, Show Expenses

Button	Action
⏮	Show First Expense
⏪	Show Previous Expense
⏩	Show Next Expense
⏭	Show Last Expense

Table 1, GUI Actions