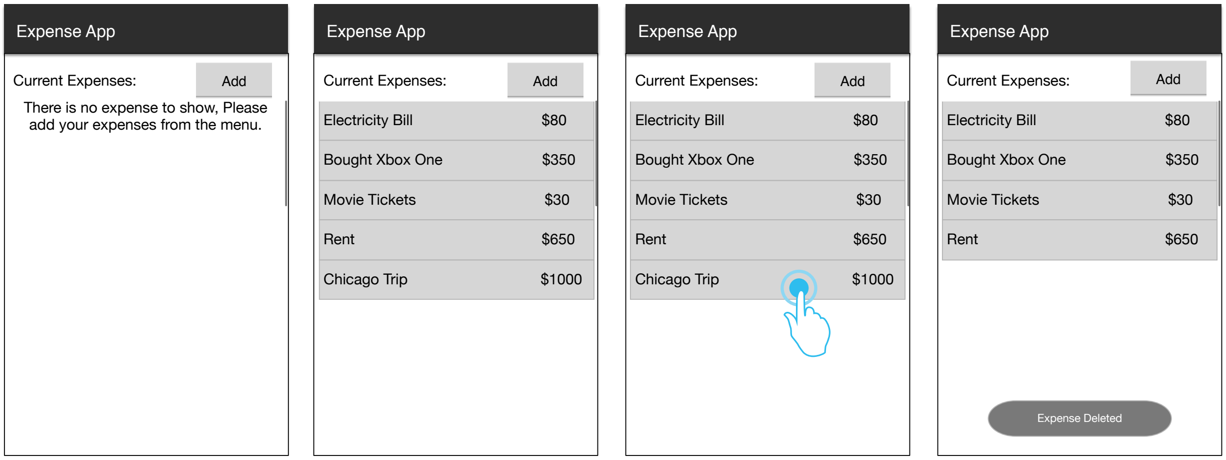
ITIS/ITCS 5180 Mobile Application Development
In Class 07

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 loose 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: **Group#_InClass07.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.**

In Class Assignment 07 (100 points)

In this assignment you will get familiar with fragments and implement an app to add and display expenses. You will practice how to pass data between Fragments. The whole application is a single activity, which hosts multiple fragments.



There is no expense to show, Please add your expenses from the menu.

Electricity Bill	$80
Bought Xbox One	$350
Movie Tickets	$30
Rent	$650
Chicago Trip	$1000

Expense Deleted

(a) No expenses in the list

(b) ListView with expenses

(c) Long Click to delete expense

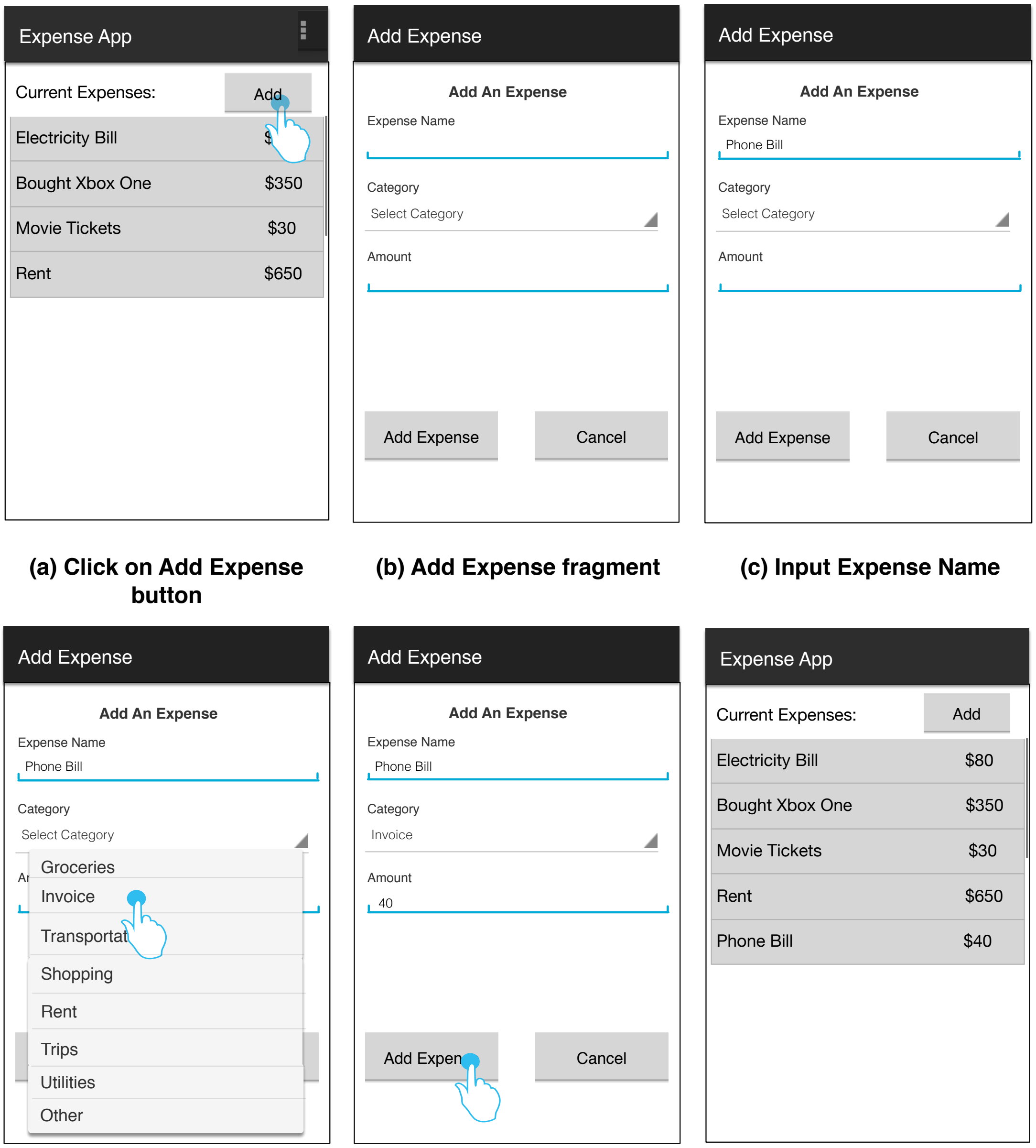
(d) Expense is deleted

Figure 1: ExpenseApp fragment

Part A: ExpenseApp fragment (40 points)

The main activity should start by displaying the ExpenseApp fragment, with the following requirements:

1. When the app first starts, there should be no expenses added in the list. So it should display the ExpenseApp fragment with a message, “There is no expense to show, Please add your expenses from the menu.” (Figure 1(a)).
2. The list of expenses (ArrayList) should be stored in the main activity, and the fragments are able to access the list of expenses through interfaces implemented in the main activity.
3. The ExpenseApp fragment should use a ListView to display the list of expenses as shown in (Figure 1(b)).
   a) Long press on an item should delete the expense from the list and refresh the ListView to indicate this change. A Toast should be displayed having the message, “Expense Deleted” (Figure 1(c, d)).
   b) Clicking on an expense item should display the ShowExpense fragment, you should push the ExpenseApp fragment on the fragment stack.
4. Clicking on the add (+) icon should start the AddExpense fragment and replace the current fragment, you should push the ExpenseApp fragment on the fragment stack.



Current Expenses:

Electricity Bill	$
Bought Xbox One	$350
Movie Tickets	$30
Rent	$650

Add An Expense

Expense Name: Phone Bill

Category: Select Category

Category: Invoice

Amount: 40

Current Expenses:

Electricity Bill	$80
Bought Xbox One	$350
Movie Tickets	$30
Rent	$650
Phone Bill	$40

(a) Click on Add Expense button

(b) Add Expense fragment

(c) Input Expense Name

(d) Select Category

(e) Input amount

(f) Added to Expense List

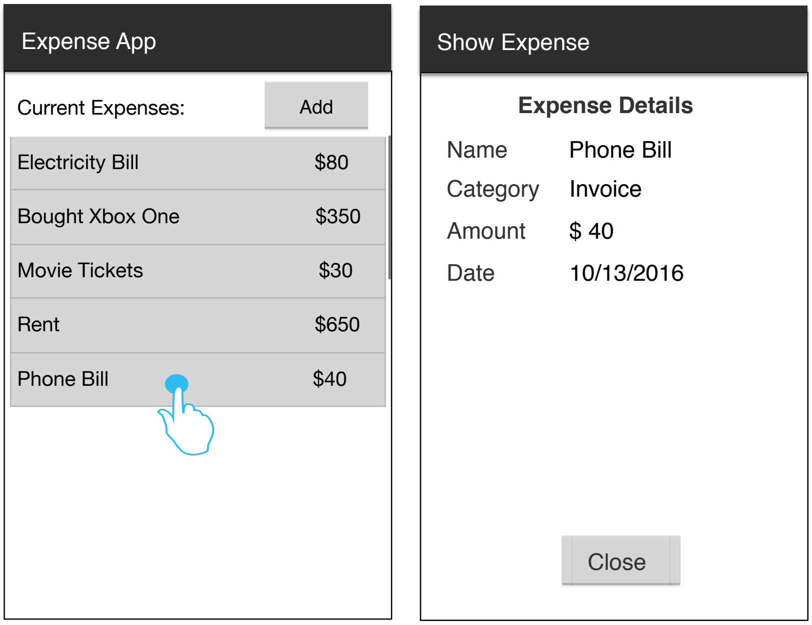
Figure 2: AddExpense fragment

Part B: AddExpense fragment (40 points)

This fragment should enable user to add a new expense. You should complete the following tasks:

1. The user should be able to enter the expense name, category and amount. The app should take use the current date as the expense date. This information should be stored in an Expense Object.
2. The categories should be in a selection pane as in the Figure 2(d). The categories you should include are: Groceries, Invoice, Transportation, Shopping, Rent, Trips, Utilities and Other.

3. Clicking on “Add Expense” button should validate the user’s input and ensure that all the fields are provided. If any field is missing, display a Toast to indicate the missing field. If all the fields are provided correctly, save the fields as an Expense object, send the new expense to the main activity to be stored in the expense list. Then the ExpenseApp fragment should be popped off the stack and should be displayed with the newly updated list of expense items as shown in Figure 2(f).



Electricity Bill	$80
Bought Xbox One	$350
Movie Tickets	$30
Rent	$650
Phone Bill	$40

Name	Phone Bill
Category	Invoice
Amount	$ 40
Date	10/13/2016

(a) Click on an Expense item **(b) Expense Details**

Figure 3: ShowExpenses fragment

<u>Part C: ShowExpenses fragment (20 Points):</u>

Implement the following requirements:

1. When the user clicks on an expense item in the ExpenseApp fragment, the ShowExpenses fragment should be started to show the details of selected expense item. as shown in Figure 3(b).
2. Upon clicking Close button, the fragment should be closed and should navigate back to the ExpenseApp fragment.