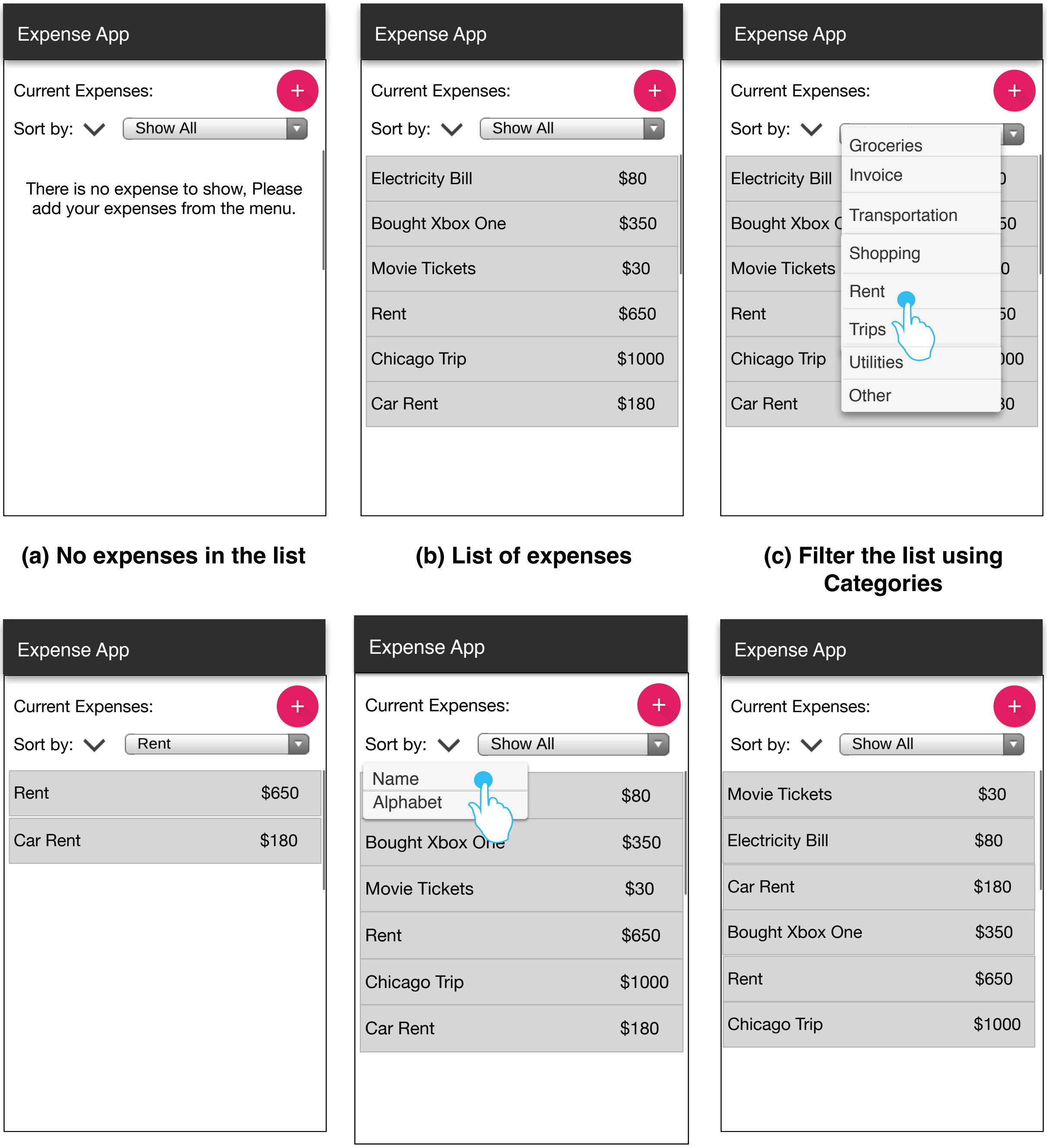
ITIS/ITCS 4180/5180 Mobile Application Development
In Class Assignment 13

Basic Instructions:

1. In every file submitted you MUST place the following comments:
   1. Assignment #.
   2. File Name
2. This is an individual assignment. Everyone must submit the assignment individually.
3. Please download the support files provided with this assignment and use them while implementing the project.
4. Export your Android project and create a zip file which includes all the project folder and any required libraries.
5. Submission details:
   1. Compress the contents of your project folder. The file name is very important and should follow the following format: **GroupID#_InClass13.zip**
   2. 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 13 (100 points)

In this assignment you will be implementing the Expense App using Realm to store and retrieve user expenses.



(a) No expenses in the list

(b) List of expenses

Electricity Bill	$80
Bought Xbox One	$350
Movie Tickets	$30
Rent	$650
Chicago Trip	$1000
Car Rent	$180

(c) Filter the list using Categories

(d) Filtered list

Sort by: Rent

Rent	$650
Car Rent	$180

(e) Sort dropdown options

(f) Sorted by price

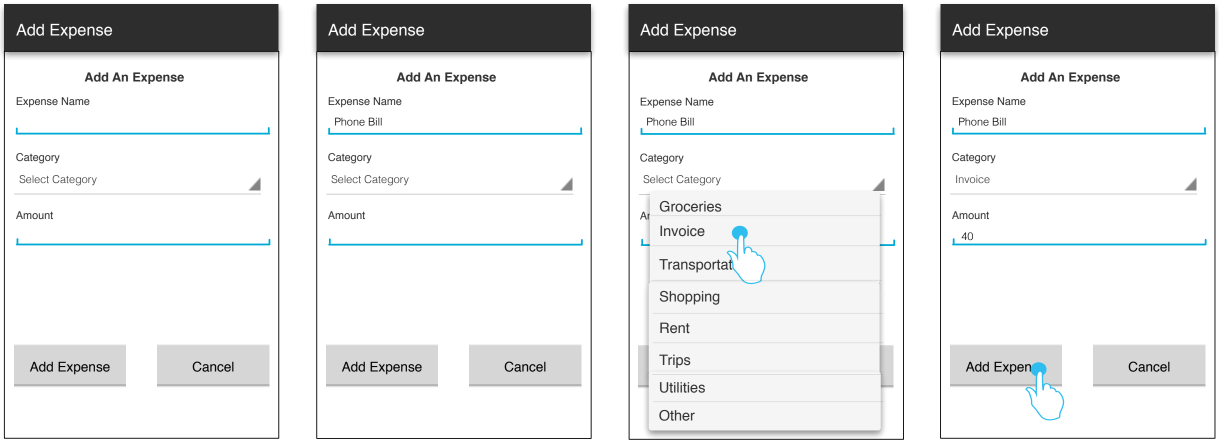
Movie Tickets	$30
Electricity Bill	$80
Car Rent	$180
Bought Xbox One	$350
Rent	$650
Chicago Trip	$1000

Figure 1: ExpenseApp activity

Part A: ExpenseApp activity (70 Points):

The ExpensesApp activity should retrieve the expenses from the Objects that are stored in Realm. A ListView/RecyclerView should display the expenses names and amount of the expense of the retrieved expenses (Figure 1). The requirements are as follows:

1. You need to retrieve list of the expenses from Realm.
2. Clicking an expense in the ListView should start the ShowExpense activity.
3. Clicking on the Add icon (+) should start the AddExpense activity.
4. Long click on any item of the list should delete the item from the list and the Realm. A Toast should be displayed having the message, “Expense Deleted”.
5. Clicking on Sorting dropdown icon (Figure 1, e) would let you select between two options to sort the list in ascending order by: Name and according to the Price. The sorting should be done through Realm.
6. You should be able to filter the list according to the category you select from another drop down (Figure 1, c). The filtering should be done through Realm.



(a) Add Expense activity.

(b) Input Expense Name

(c) Select Category

(d) Input amount and Add Expense

Figure 2: AddExpense activity

Part B: AddExpense activity (10 points)

This activity 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 Realm.
2. The categories should be in a selection pane as in the Figure 2(c). 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

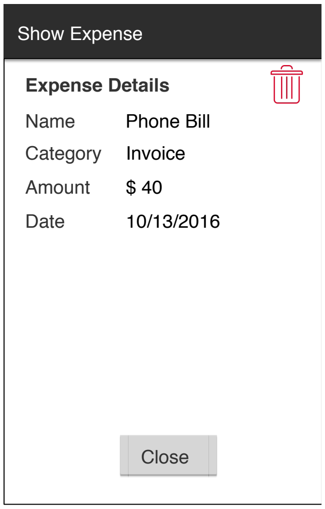
missing field. If all the fields are provided correctly, save the fields in Realm. Then the ExpenseApp activity should display the updated list of expense items.

Part C: ShowExpenses and EditExpenses Activity (20 Points):

The ShowExpenses activity should display the Expense details, which include the name, category, amount and date of the Expense, see Figure 3.

Implement the following requirements:

4. Clicking on Close button should finish the ShowExpenses activity and go back to the ExpenseApp activity.
5. Clicking on the Trash icon should delete the current expense from realm, and finish the current activity.



Expense Details

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

Figure 3: ShowExpense activity