ITIS/ITCS 4180/5180 Mobile Application Development
In Class Assignment 10

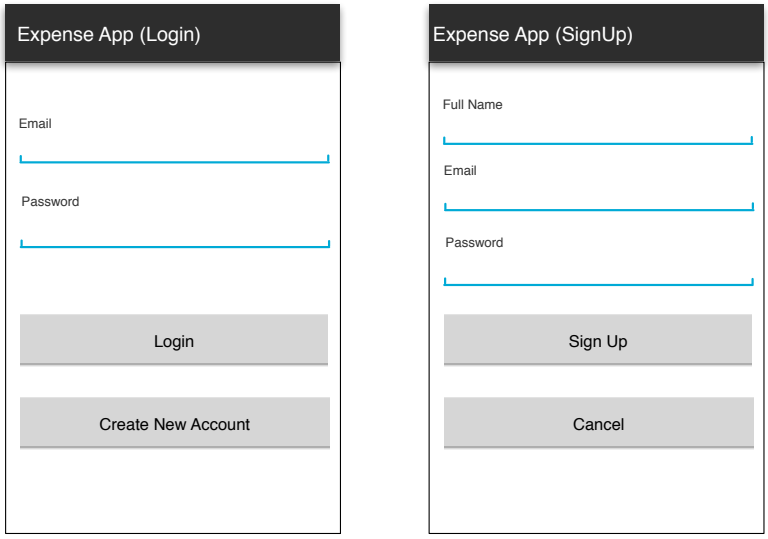
Basic Instructions:

---

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#_InClass10.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 10 (100 points)

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



(a) Login Activity (b) SignUp Activity

Figure 1 : Login and SignUp Activities

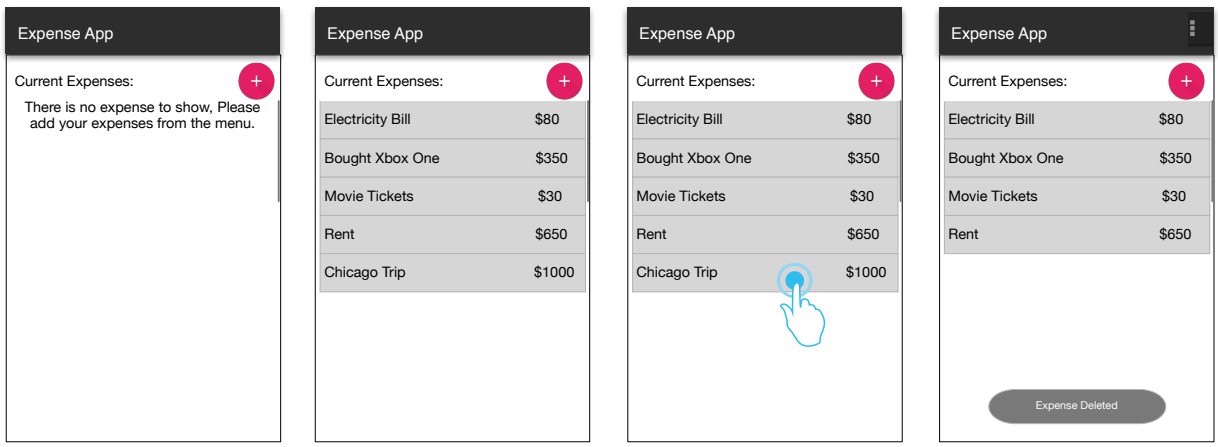
Part A: User SignUp and Login (25 points)

You should implement both login and signup functionalities. You should use Firebase to register a user with an email and password, and also store the user's full name, email address, and password in the User object. The requirements are as follows:

1. In Firebase make sure to enable Email & Password Authentication.
2. The launcher activity should be set to the Login activity. When the app first starts, the Login activity should check if there is a current user session, by using the Firebase provided methods to check if there is a valid current user.
   a. If there is a current valid user, then start the ExpensesList activity, and finish the Login activity.
   b. If there is no current valid user, then the Login activity should be used to provide user login.
3. Create a Login Activity (Figure 1a):
   a. The user should provide their email and password. The provided credentials should be used to authenticate the user using Firebase. Clicking the "Login" button should submit the login information to Firebase to verify the user's credentials.
      i. If the user is successfully logged in then start the ExpensesList activity, and finish the Login activity.
      ii. If the user is not successfully logged in, then show a toast message indicating that the login was not successful.
   b. Clicking the "Create New Account" button should start the Signup activity and finish the login activity.

4. Create a Sign Up activity (Figure 1b):
   a. Clicking the “Cancel” button should finish the Signup activity and start the Login activity.
   b. The user should provide their Full name, email and password. The provided credentials should be added as a registered user on Firebase. Clicking the “Sign Up” button should submit the user’s information to Firebase to verify the user’s credentials.
      i. If an account with the same email already exists, display an error message indicating that the account was not created and the user should select a different email.
      ii. If an account with the provided credentials does not already exist, then store the new account information and display a Toast indicating that the user has been created. Then start the Login activity and finish the Signup activity.
      iii. Note that, the user data should be stored in Firebase Users object because the user authentication for registered users will not take Full name in Firebase.

Part B: ExpenseApp activity (35 points)



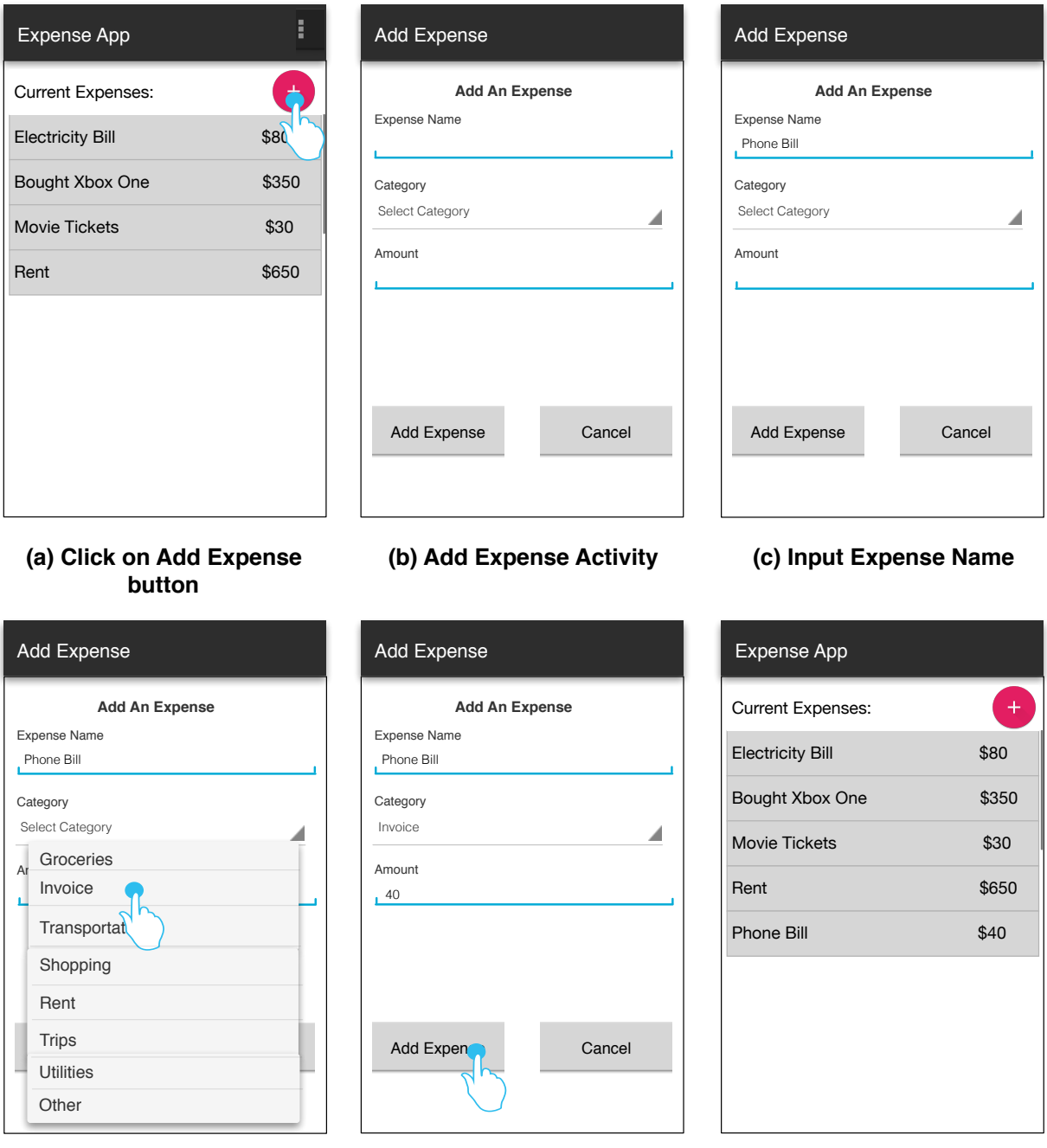
(a) No expenses in the list **(b) ListView with expenses** **(c) Long Click to delete expense** **(d) Expense is deleted**

Figure 2: ExpenseApp Activity

The ExpensesApp activity should retrieve the expenses from the Objects that are stored in Firebase. A ListView/RecyclerView should display the expenses names and amount of the expense of the retrieved expenses (Figure 2). The requirements are as follows:

1. You need to retrieve list of the expenses from the stored Expenses object in Firebase. Please note that the list retrieved should belong to the currently logged in user.
2. Check the documentation provided at https://www.firebase.com/docs/web/guide/retrieving-data.html for retrieving the objects from Firebase.
3. Clicking an expense in the ListView should start the ShowExpense activity.
4. Clicking on the Add icon (+) should start the AddExpense activity.

5. Long click on any item of the list should delete the item from the list and the Firebase. A Toast should be displayed having the message, “Expense Deleted” (Figure 2, d).



(a) Click on Add Expense button **(b) Add Expense Activity** **(c) Input Expense Name**

(d) Select Category **(e) Input amount** **(f) Added to Expense List**

Figure 3: AddExpense Activity

Part C: AddExpense activity (25 points)

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

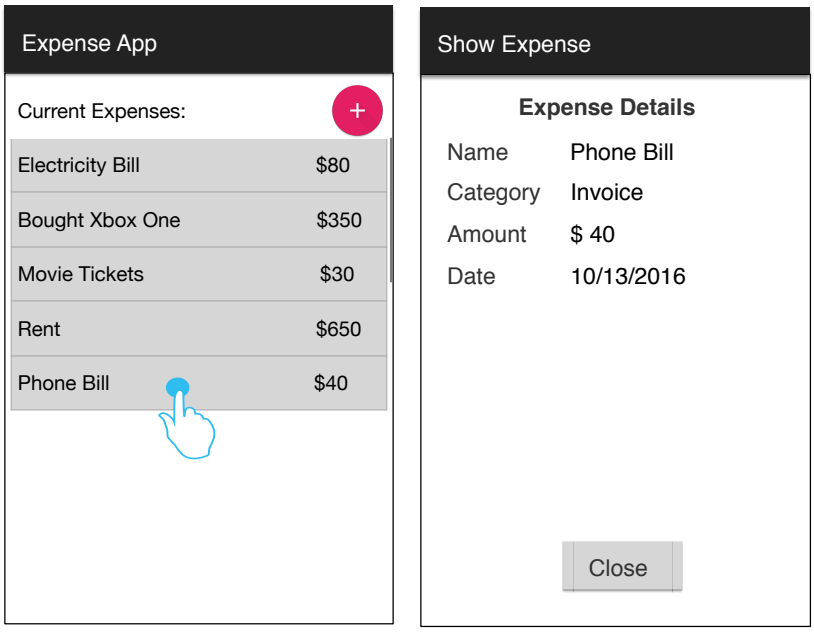
1. The user should be able to enter the expense name, category and amount. The app should take use the current date as the expense date. This information should be stored in an Expense Object in Firebase.
2. The categories should be in a selection pane as in the Figure 3(d). The categories you should 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 Firebase object. Then the ExpenseApp activity should display the updated list of expense items as shown in Figure 3(f).
4. The documentation for Saving Expenses in Firebase are given here: https://firebase.google.com/docs/database/android/read-and-write

Part D: ShowExpenses Activity (15 Points):

The ShowExpenses activity should display the Expense details, which include the name, category, amount and date of the Expense, see Figure 4.

Implement the following requirements:

1. Display the details as shown in Figure 4, b.
2. Clicking on Close button should finish the ShowExpenses activity and go back to the ExpenseApp activity.



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

Figure 4: ShowExpenses activity