ITIS/ITCS 4180/5180 Mobile Application Development
In Class Assignment 11

Basic Instructions:

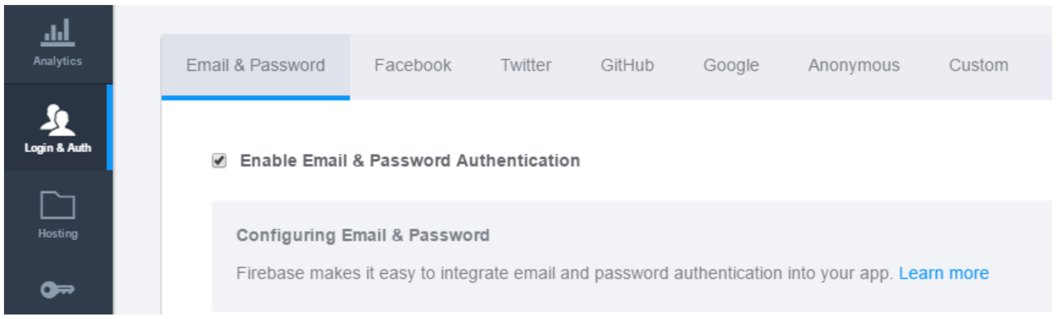
1. In every file submitted you MUST place the following comments:
   a. Assignment #.
   b. File Name.
   c. Full name of all students in your group.
2. Each group should submit only one assignment. Only the group leader is supposed to submit the assignment on behalf of all the other group members.
3. Please download the support files provided with this assignment and use them when implementing your project.
4. Export your project as follows:
   a. From eclipse, choose *“Export…”* from the File menu.
   b. From the Export window, choose *General* then *File System*. Click *Next*.
   c. Make sure that your project for this assignment is selected. Make sure that all of its subfolders are also selected.
   d. Choose the location you want to save the exported project directory to. For example, your *Desktop* or *Documents* folder.
   e. When exporting make sure you select *Create directory structure for files*.
   f. Click Finish, and then go to the directory you exported the project to. Make sure the exported directory contains all necessary files, such as the .java and resource files.
5. Submission détails:
   a. Only a single group member is required to submit on moodle for each group.
   b. The file name is very important and should follow the following format: **Group#_InClass11.zip**
   c. You should submit the assignment through Moodle: Submit the zip file.
6. **Failure to follow the above instructions will result in point deductions.**

In Class Assignment 11 (100 Points)

In this assignment you will implement an app to add and display expenses. You will use Firebase to store and retrieve user expenses.

Part A: User Signup and Login (25 Points)

Your app 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:

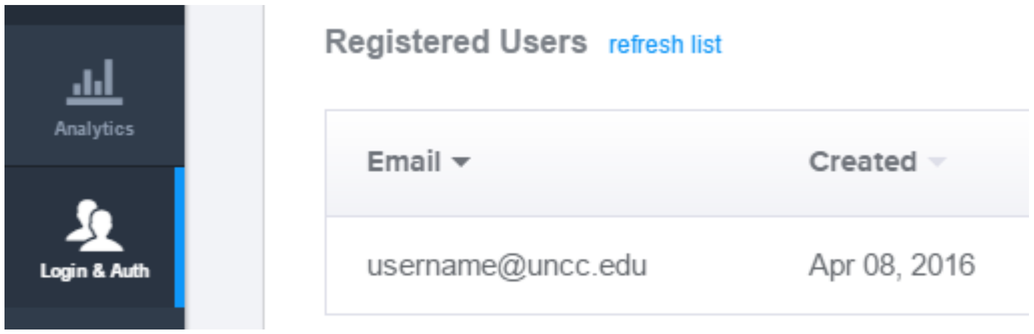


☑ Enable Email & Password Authentication

Configuring Email & Password

Firebase makes it easy to integrate email and password authentication into your app.

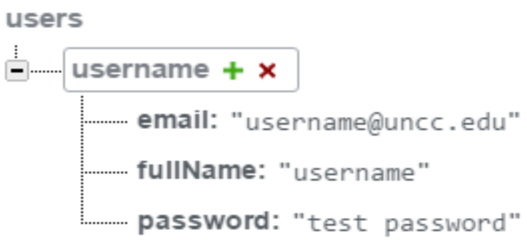
(a) Enable Email & Password Authentication in Login & Auth section in Firebase



Registered Users

Email	Created
username@uncc.edu	Apr 08, 2016

(b) User should appear after sign up in Registered Users section in Firebase



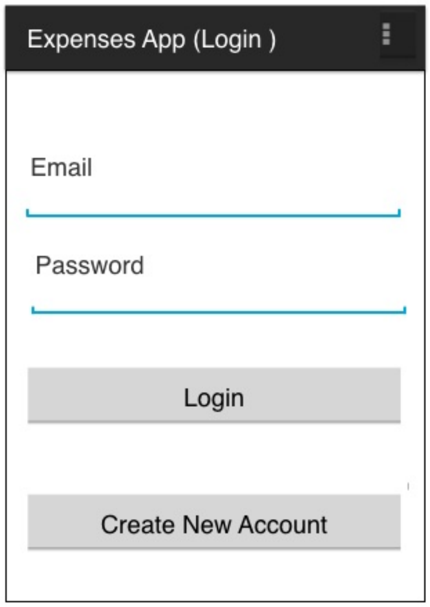
```
users
username
    email: "username@uncc.edu"
    fullName: "username"
    password: "test password"
```

(c) User should also be added to the Users object

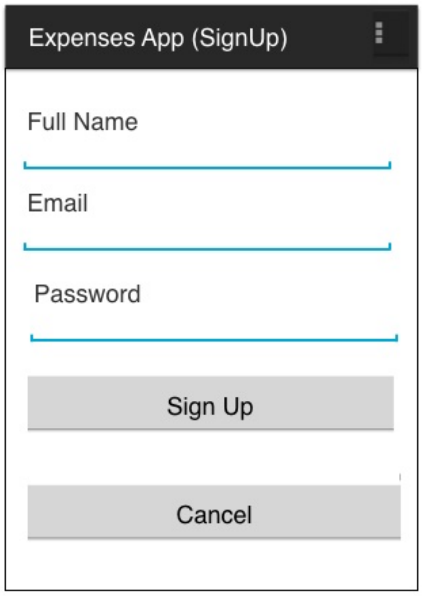
1. Please make sure to enable Email & Password Authentication as shown in the figure (a)
2. The launcher activity should be set to the Login activity. When the app first starts, the Login activity should check if there is a current user session, by using the Firebase provided methods to check if there 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 1(a)):
   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.
      - If the user is successfully logged in then start the ExpensesList activity, and finish the Login activity.
      - 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 Signup activity (Figure 1(b)):
   a) Clicking the "Cancel" button should finish the Signup activity and start the Login activity.
   b) The user should provide their username, email and password. The provided credentials should be added as a registered user as shown in figure (b). Clicking the "Sign Up" button should submit the user's information to Firebase to verify the user's credentials.
      - 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.
      - 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.
      - Note that, the user data should be stored in Firebase Users object as data as shown in the figure (c) because the user authentication for registered users will not take Username in Firebase.



(a) Login Activity



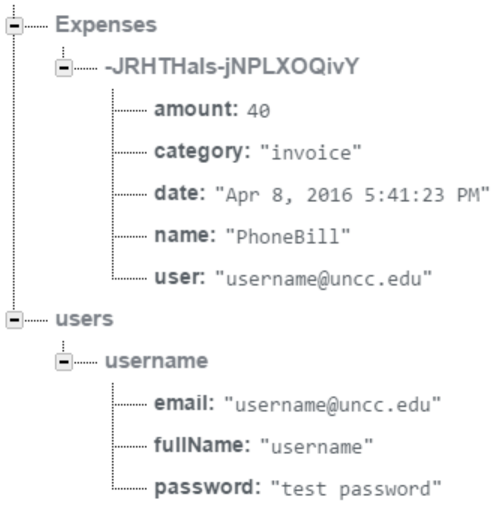
(b) SignUp Activity

Figure 1, Wireframe for Login and SignUp Activities

Part B: Expenses List Activity (35 Points)

The Expenses List activity should retrieve the expenses for the currently logged in user that are stored in Firebase. A ListView should display the expenses names and amount of the expense of the retrieved expenses, see Figure 2a. The requirements are as follows:

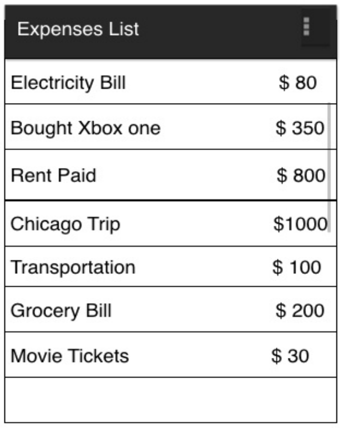
1. You need to retrieve list expenses from the stored Expenses object in Firebase as shown in the below figure.



```
Expenses
  -JRHTHals-jNPLXOQivY
    amount: 40
    category: "invoice"
    date: "Apr 8, 2016 5:41:23 PM"
    name: "PhoneBill"
    user: "username@uncc.edu"
users
  username
    email: "username@uncc.edu"
    fullName: "username"
    password: "test password"
```

2. The Expenses should be retrieved belonging to the currently logged in user. Check the documentation provided at https://www.firebase.com/docs/web/guide/retrieving-data.html

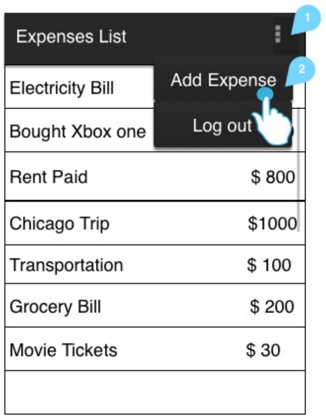
3. Clicking an expense in the ListView should start the Expense Detail activity.
4. Clicking the “Add Expense” menu item should start the Add Expense activity. As shown in Figure 2(b), and should result in an added expense, as in 2(c).
5. Clicking on logout, see 2(h), should end the current user session and display the login activity.



Expenses List

Electricity Bill	$ 80
Bought Xbox one	$ 350
Rent Paid	$ 800
Chicago Trip	$1000
Transportation	$ 100
Grocery Bill	$ 200
Movie Tickets	$ 30

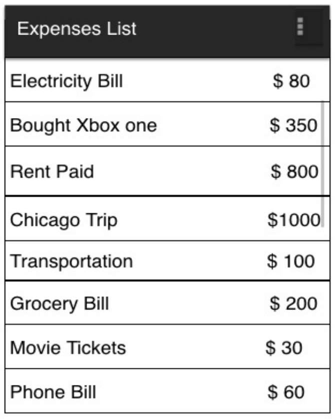
Figure 2a, Expenses List Activity



Expenses List

Electricity Bill	
Bought Xbox one	
Rent Paid	$ 800
Chicago Trip	$1000
Transportation	$ 100
Grocery Bill	$ 200
Movie Tickets	$ 30

Figure 2b, On click Add Expenses opens Add Expenses Activity



Expenses List

Electricity Bill	$ 80
Bought Xbox one	$ 350
Rent Paid	$ 800
Chicago Trip	$1000
Transportation	$ 100
Grocery Bill	$ 200
Movie Tickets	$ 30
Phone Bill	$ 60

Figure 2c, Expenses gets added into the list

Part C: Add Expense Activity (25 Points)

The Add Expense activity should enable the currently logged-in user to create a new expense. The user should enter the expense name, Category, amount and date, as show in Figure 3. The requirements are as follows:

1. Clicking the “Add Expense” button:
   a) 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, then save the new expense in the Expenses object in Firebase. If the Expenses object is not been created already please add a child to the root object as expenses and then add the created expense.
      - For every expense object please save Expense name, category, amount, date and user email id who is creating the expense.
      - After successful saving in to the Firebase the Add Expenses activity should be finished. Upon returning to the Expenses List activity the list should be updated to display the newly added expense.

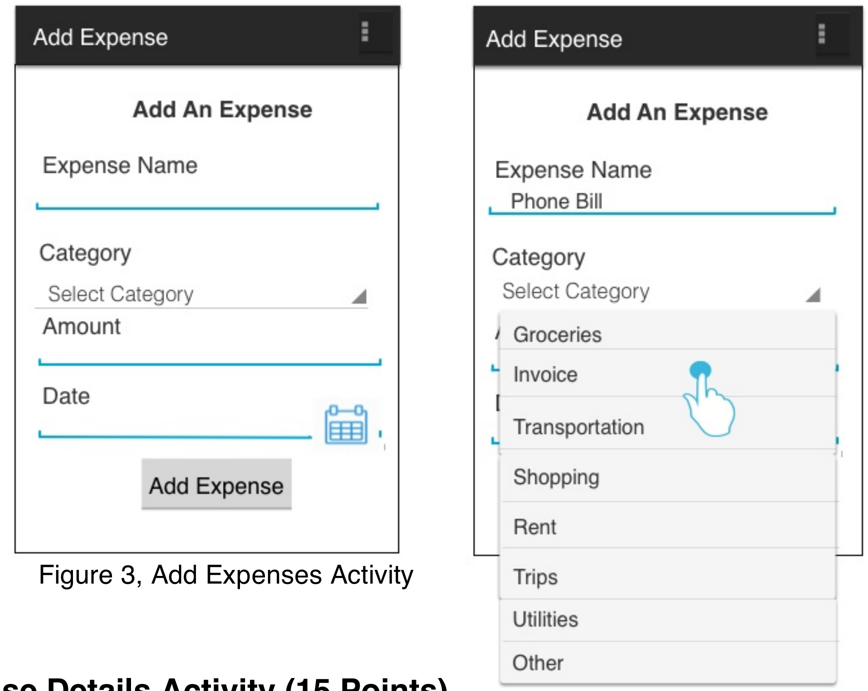
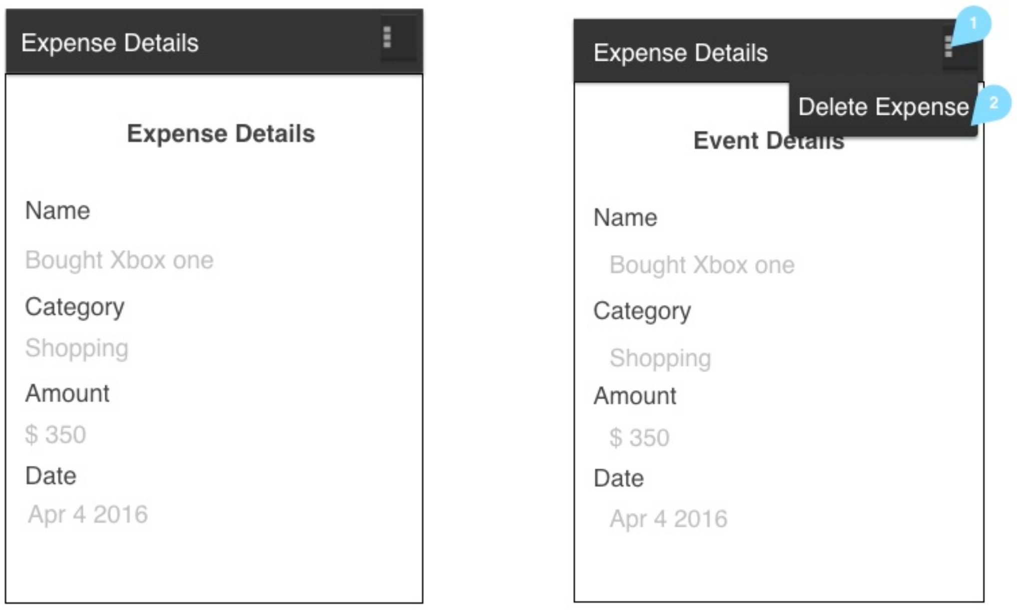


Figure 3, Add Expenses Activity

Part D: Expense Details Activity (15 Points)

The Expense Details activity shows the Expense details, which include the Expense name, category, amount and date, see Figure 4.

1. Clicking the back button should finish the Expense Details activity and go back to the Expense List activity.
2. Clicking the "Delete Expense" action bar menu item should delete the currently displayed Expense in the Firebase, as shown in Figure 4(b).
   a) If the event is successfully deleted then display a toast message indicating the successful deletion of the event, and finish the Expense detail activity.
   b) If the deletion is not successful then display a toast message indicating that there was an error while deleting the event.



(a) Expense Details Activity (b) Deleting Expense

Figure 4, Wireframe for Expense Details Activity