

Mobile App Development
Assignment 8 (100 Points)

Basic Instructions:

1. In every file submitted you MUST place the following comments:
 - a. Assignment #.
 - b. File Name.
 - c. Full name of the student.
2. Each group is required to submit the assignment on Canvas.
3. **Submit Codes:**
 - a. Zip all the project folder to be submitted on canvas.
4. Submission details:
 - a. The file name is very important and should follow the following format:
Assignment#.zip
 - b. You should submit the assignment through Canvas: Submit the zip file.
5. **Failure to follow the above instructions will result in point deductions.**

In this assignment you will develop a contacts application that uses Firebase Authentication and Firestore. The main objectives are to provide you with an introduction to Firebase concepts. Your application should contain only **one Activity and multiple fragments** to implement the required screens.

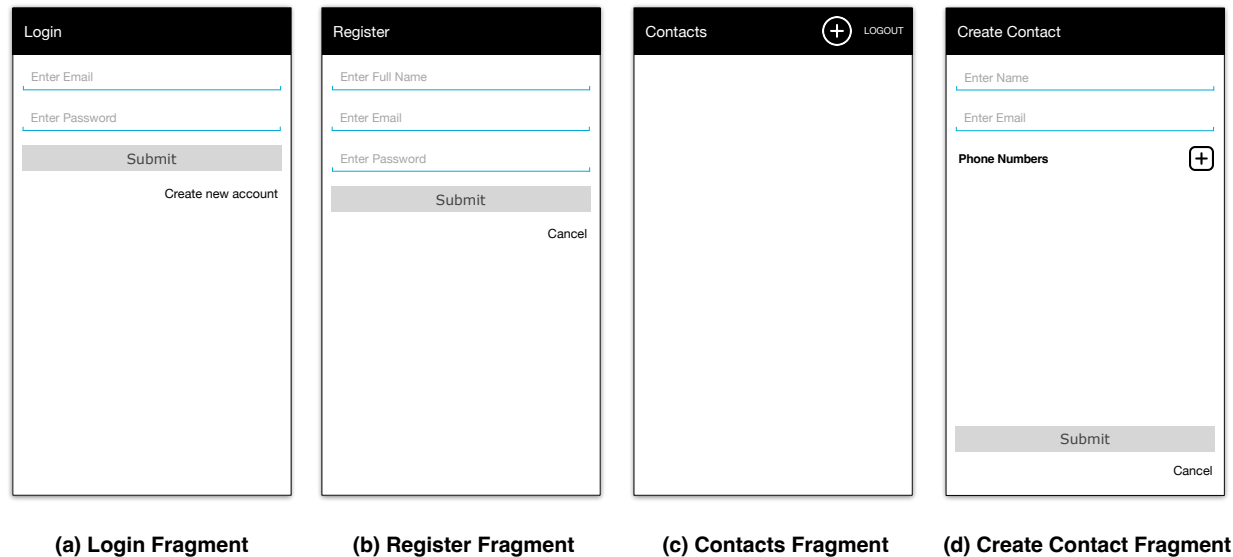


Figure 1, Application Wireframe

Part 1: Login Fragment (5 Points)

The interface should be created to match the UI presented in Figure 1(a). The requirements are as follows:

1. Clicking the "Create New Account" button should replace this fragment with the Register Fragment.
2. Clicking the "Login" button, the app should check if email and password fields are entered and display an alert dialog if any of the entries is missing indicating that the missing field is required.
 - a. Use the Firebase Auth to login the user. If the login is successful then replace the current fragment with the Contacts Fragment.
 - b. If the login is not successful then display an alert dialog indicating the reason returned by Firebase Auth.

Part 2: Register Fragment (5 Points)

The interface should be created to match the UI presented in Figure 1(b). The requirements are as follows:

1. This screens allows the user to register a new user account using email and password.
2. Clicking the "Cancel" button should dismiss this Screen and show the Login Fragment.
3. Clicking the "Submit" button, the app should check if email and password fields are entered and display an alert dialog if any of the entries is missing indicating that the missing field is required.
 - a. Use the Firebase Auth to register a new user. If the registration is successful then replace the current screen with the Contacts Fragment.

- b. If the registration is not successful then display an alert dialog indicating the reason returned by Firebase Auth.

Part 3 : Contacts Fragment (30 Points)

This screen enables the user to view the list of contacts for the currently logged in user. As shown in Figure 1(c) and 2(c), The requirements are as follows:

1. This screen should retrieve the list of contact from Firestore that belong to the currently logged in user.
 - a. You should create a Contact class to store the contact information.
 - b. Setup a snapshot listener to listen to realtime updates for the contacts collection and to update the contacts RecyclerView when the contacts are updated.
 - c. **The snapshot listener should use a query to receive only the contacts created by the current user from Firebase Firestore.**
2. The contacts list should be displayed using a RecyclerView as shown in Figs 1(c) and 2(c). Each row item should display the contact name, email, first phone number and the number of phone numbers as shown in Fig 2(c).
 - a. **The trash “Delete” icon should:**
 - i. Delete the selected contact from Firestore.
 - ii. You are required to initially setup realtime updates (snapshot listener) which should trigger when the contact is deleted and refresh the displayed list after deletion to reflect this change.
 - b. Clicking on a contact row item should:
 - i. Replace the current fragment with the Contact Details Fragment and Push the current fragment on the back stack.
 - ii. The selected contact should be sent to the Contact Details Fragment.
3. Clicking the “LOGOUT” menu button should logout the currently logged in user and replace this fragment with the Login Fragment.
4. Clicking the “+” menu button should:
 - a. Replace the current fragment with the Create Contact Fragment and Push the current fragment on the back stack.

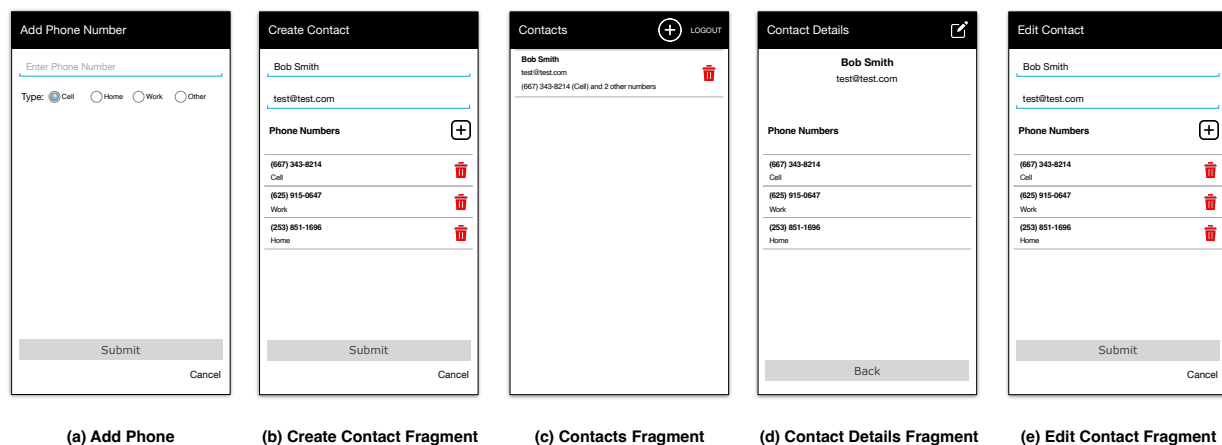


Figure 2, Application Wireframe

Part 4 : Create Contact Fragment (20 Points)

The interface should be created to match Figure 1(d) and 2(b), this fragment allows the user to create a new contact and to add phone numbers to the new contact. The requirements are as follows:

1. Clicking the “+” button should:
 - a. Replace the current fragment with the Add Phone fragment.
 - b. Push the current fragment on the back stack.
 - c. Upon returning from the Add Phone fragment the created PhoneNumber object should be received through the Main Activity, sent to the Create Contact fragment, added to the ArrayList of PhoneNumber in the fragment and should be displayed in a RecyclerView as shown in Fig 2(b).
2. Clicking on the trash can on a row item should:
 - a. Delete the selected Phone Number from the ArrayList of PhoneNumber maintained in the fragment.
 - b. Refresh the RecyclerView to display the updated list of PhoneNumbers.
3. Clicking the “Submit” button, should check if all the fields are entered and display an alert dialog if any of the entries is missing indicating that the missing field is required. It is required to have at least ONE phone number added in order to proceed with creating the new contact.
 - a. If all the fields are entered, and at least one phone number is added, the app should create a new Contact on Firestore, upon a successful update the app should pop the back stack to display the Contacts fragment.
 - b. Upon returning to the Contacts fragment, the list should be refreshed to show the newly added contact.
4. Clicking the “Cancel” button should:
 - a. Pop the back stack which should return back to the Contacts Fragment.

Part 5 : Add Phone Number Fragment (5 Points)

The interface should be created to match Figure 2(a), this fragment allows the user to enter a phone number and select it's type. The requirements are as follows:

1. Clicking the “Submit” button, should check if all the fields are entered and display an alert dialog if any of the entries is missing indicating that the missing field is required.
 - a. Find the previous fragment by tag and send it the created PhoneNumber object.
 - b. Pop the back stack which should display the previous fragment and display the newly added phone number.
2. Clicking the “Cancel” button should:
 - a. Pop the back stack.

Part 6 : Contact Details Fragment (5 Points)

The interface should be created to match Figure 2(d), this fragment displays details of the selected contact. The requirements are as follows:

1. Setup a SnapshotListener to listen to realtime updates for the received contact object. This should be used to display the current contact details and phone numbers.
2. Clicking the “Back” button should:
 - a. Pop the back stack.

3. Clicking the “Edit” menu button should:
 - a. Replace the current fragment with the Edit Contact Fragment and Push the current fragment on the back stack.
 - b. Send the Contact object to the Edit Contact Fragment.
 - c. Replace the current fragment with the Contact Details Fragment and Push the current fragment on the back stack.

Part 7 : Edit Contact Fragment (30 Points)

The interface should be created to match Figure 2(e), this fragment allows the user to edit a current contact, add or remove phone number to this contact. The requirements are as follows:

1. Setup a SnapshotListener to listen to realtime updates for the received contact object. This should be used to display the current contact details and phone numbers.
2. Clicking the “+” button should:
 - a. Replace the current fragment with the Add Phone fragment.
 - b. Push the current fragment on the back stack.
 - c. Upon returning from the Add Phone fragment the PhoneNumber object should be received through the Main Activity, sent to the Edit Contact fragment, which should update the contact document on Firebase Firestore to add the newly added PhoneNumber.
 - d. The fragment already setup a SnapshotListener for this contact which should be triggered and is expected to refresh the PhoneNumbers RecyclerView.
3. Clicking on the trash can on a row item should:
 - a. Delete the selected Phone Number from the contact document on Firebase Firestore.
 - b. The fragment already setup a SnapshotListener for this contact which should be triggered and is expected to refresh the PhoneNumbers RecyclerView.
4. Clicking the “Submit” button, should check if the name and email are entered correctly. This feature allows the user to update the name and email of the current contact.
 - a. If the name and email are entered, update the contact document with the updated name and email on Firebase Firestore. Upon a successful update the app should pop the back stack to display the Contact Details fragment.
5. Clicking the “Cancel” button should:
 - a. Pop the back stack.