Mobile App Development

Assignment 09

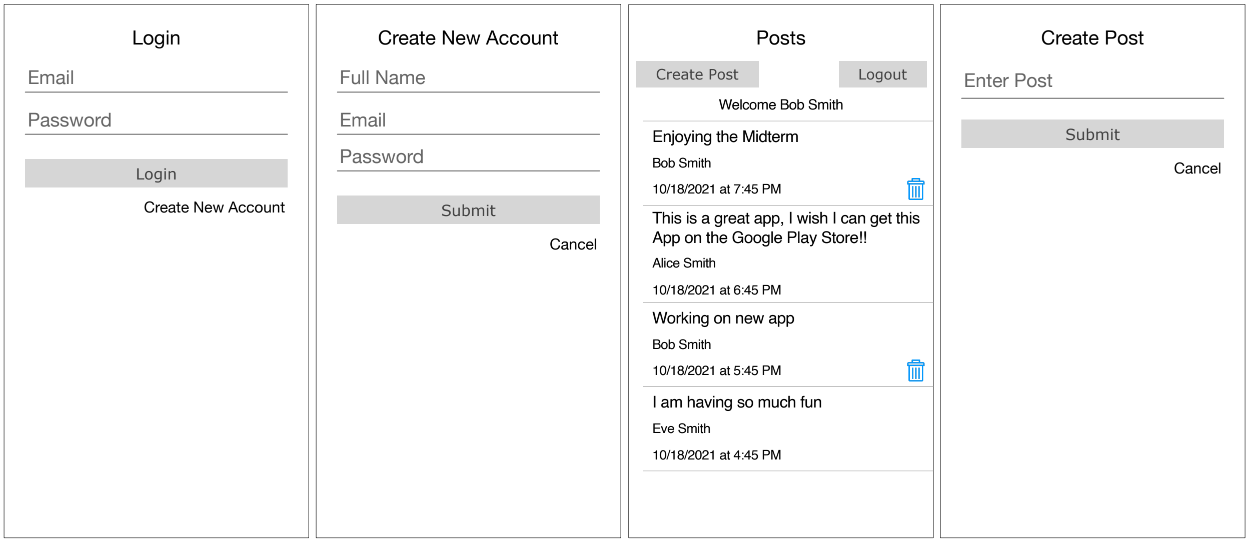
Basic Instructions:

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.**

Assignment 9 (100 Points)

In this assignment you will develop a simple posts application, which uses Firebase for Authentication and Data storage.

1. Create a new Firebase project.
2. Import Firebase into the provided skeleton app make sure to register the app in your Firebase project.
3. Import Firebase Authentication and Firestore libraries into your project.
4. All the flow and UI parts of the app have been included in the skeleton project, you should focus on making the required Firebase Auth and Firestore calls to implement the required functions.



Enjoying the Midterm
Bob Smith
10/18/2021 at 7:45 PM

This is a great app, I wish I can get this App on the Google Play Store!!
Alice Smith
10/18/2021 at 6:45 PM

Working on new app
Bob Smith
10/18/2021 at 5:45 PM

I am having so much fun
Eve Smith
10/18/2021 at 4:45 PM

(a) Login **(b) Register** **(c) Posts** **(d) Create Post**

Figure 1, Application User Interface

Part 1: Checking User Authentication (10 Points)

The requirements are as follows:

1. When the app starts, you should check if the user has successfully logged in or registered, then go to the Posts screen, else show the Login Screen.
2. Upon user logout, show the Login Screen.

Part 1: Login Screen (10 Points)

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

1. Upon entering the email and password:
   a. Clicking “Login” button, if all the inputs are not empty, you should attempt to login the user by Firebase Authentication.
   b. If login is successful, then replace the current screen with the Posts Screen.
   c. If login is not successful, show an alert dialog showing the error message returned by the Firebase Auth.
   d. If there is missing input, show an alert dialog indicating missing input.
2. Clicking the “Create New Account” should goto the Create New Account Screen.

Part 2: Create New Account Screen (10 Points)

This screen allows a user to create a new account. The interface should be created to match Figure 1(b). The requirements are as follows:

1. Upon entering the full name, email and password, clicking the Submit button should:
   a. If all the inputs are not empty, you should attempt to signup a new user using Firebase Authentication and Profile update feature.
   b. If the registration is successful, then goto the Posts Screen.
   c. If the registration is not successful, show an alert dialog showing the error message returned by the api.
   d. If there is missing input, show an alert dialog indicating missing input.
2. Clicking “Cancel” should show the Login Screen.

Part 3 : Posts Screen (50 Points)

This screen enables the user to view the posts list. As shown in Figure 1(c), The requirements are as follows:

1. Clicking the “Logout” button should logout the current user using the Firebase Authentication library and then show the Login Screen.
2. Clicking the “Create Post” button should show the Create Post Screen.
3. Greeting TextView shows “Hello XX” where XX is the name of the logged in user.
4. The list of posts should be retrieved be getting all the documents stored under the “posts” collection.
   a. The returned post documents should be parsed into an ArrayList of Post objects. Each post document should included the docId, post text, post creation date/time, post’s creator userId and name.
   b. Use the parsed list of Post objects to display the posts list in RecyclerView.
5. For the posts list, each post row item should display the post text, creators name, creation date as shown in Figure 1(c). The trash icon should only be visible for post items that were created by the currently logged in user, you should use the user id to perform this comparison. For example, in Figure 1(c) the currently logged in user is “Bob Smith” who has created the first and third posts displayed.
   a. Clicking on the trash icon, the app should present an alert dialog asking the user if the selected post should be deleted. If the user picks “OK” the selected post should be deleted by using the Firestore delete document. Upon successfully deleting a post item, the “posts” collection should be reloaded from Firestore to retrieve the latest list of posts and the posts list should be refreshed with the retrieved posts.

Part 4 : Create Post Screen (20 Points)

This screen enables the user to create a new post. The requirements are as follows:

1. Upon entering the post text, clicking the Submit button should:
   a. If all the inputs are not empty, a new post should be created as a new document under the “posts” collection on Firestore.
   b. Upon successfully creating a new post document, go back to the Posts screen which should reload the “posts” collection from Firestore to retrieve the latest list of posts and the posts list should be refreshed with the retrieved posts.