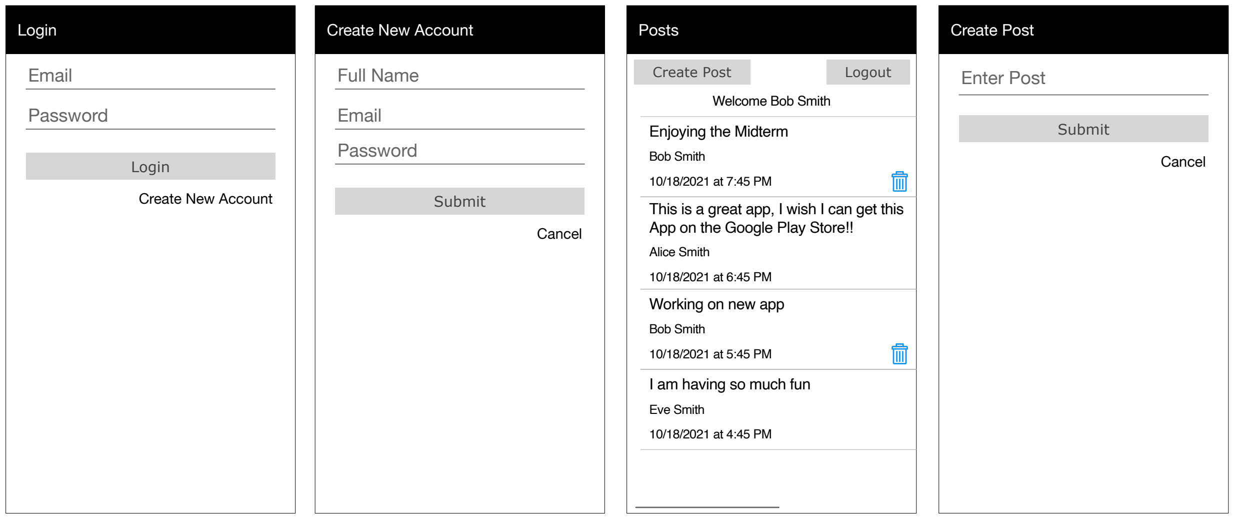
<u>In Class 07 (100 Points)</u>

In this assignment you will develop a simple posts application, in which users can make short 140 character posts visible to other users on the app. In this assignment you will get familiar with RecyclerViews and using Firebase Authentication and Firestore. You will develop a posts application which enable you to list, create, and delete posts. The app should use Firebase Firestore to store and manage the posts and Firebase Authentication to manage user authentication. Your application should contain only **one Activity and multiple fragments** to implement the required screens.



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 Fragment **(b) Register Fragment** **(c) Post List Fragment** **(d) Create Post Fragment**

Figure 1, Application Wireframe

Part 1: Login Fragment (20 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 Create New Account 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 Posts Fragment.
    b. If the login is not successful then display an alert dialog indicating the reason returned by Firebase Auth.

Part 2: Create New Account Fragment (20 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 Posts Fragment.
   b. If the registration is not successful then display an alert dialog indicating the reason returned by Firebase Auth.

Part 3 : Posts Fragment (40 Points)
This screen enables the user to view the list of posts created by all the users. As shown in Figure 1(c), The requirements are as follows:

1. This screen should retrieve the list of posts from Firestore.
   a. You should create a Post class to store the post information.
   b. Setup a snapshot listener to listen to realtime updates for the posts collection and to update the posts RecyclerView when the posts are updated.
2. The posts list should be displayed using a RecyclerView as shown in Fig 1(c). Each row item should display the post title, created by, creation date/time.
   a. **The trash “Delete”** icon should **only** be displayed for post items that were created by the currently logged in user.
      i. If the user clicks the “Delete” icon the post should be deleted on Firestore.
      ii. You are required to initially setup realtime updates which should trigger when the post is deleted and refresh the displayed list after deletion to reflect this change.
3. Clicking the “Logout” menu button should logout the currently logged in user and replace this fragment with the Login Fragment.
4. Clicking the “Create Post” button should:
   a. Replace the current fragment with the Create Post Fragment and Push the current fragment on the back stack.

Part 4 : Create Post Fragment (20 Points)
The interface should be created to match Figure 1(d). The requirements are as follows:

1. Clicking the “Cancel” button should:
   a. Pop the back stack which should return back to the Posts Fragment.
2. Clicking the “Submit” button, the app 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. If all the fields are entered, the app should store a new post on Firestore, upon a successful update the app should pop the back stack to display the Posts fragment.
   b. Upon returning to the Posts fragment, the list should be refreshed to show the newly added post.