Mobile App Development
Final Exam

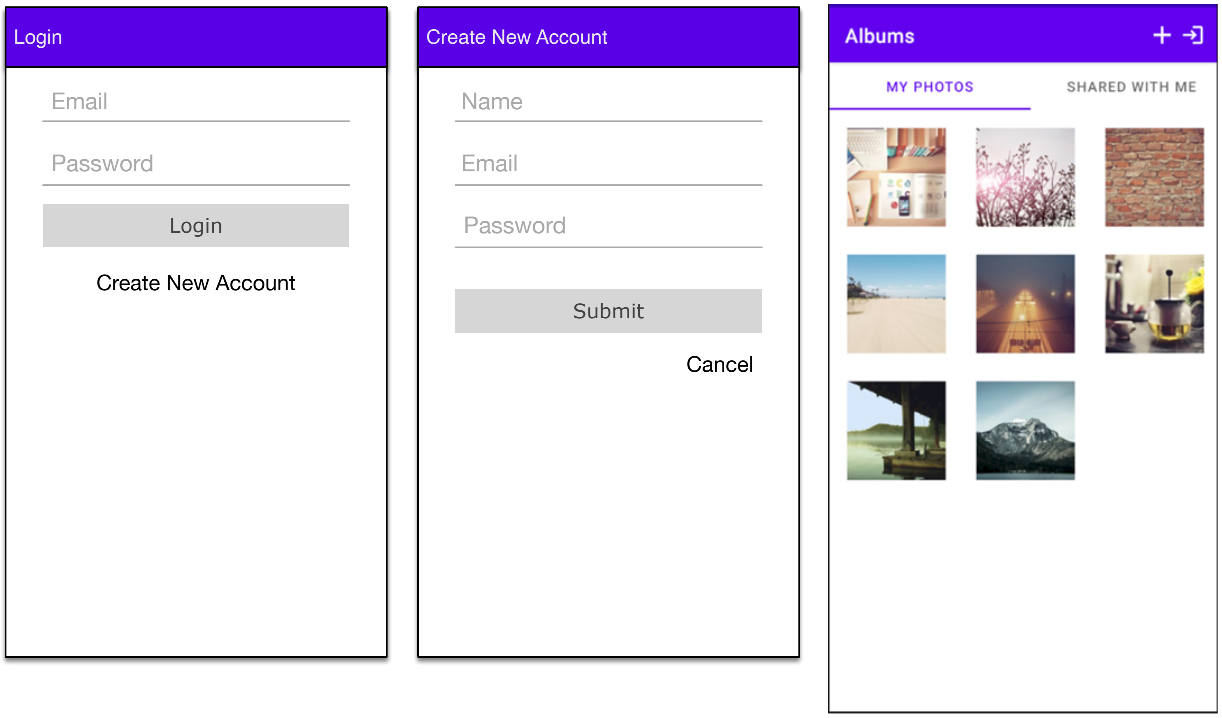
Basic Instructions:

1. This Final is an individual effort. Each student is responsible for her/his own Final and its submission.
2. Once you have picked up the exam, you may not discuss it in any way with anyone until the exam period is over.
3. During the exam, you are allowed to use the course videos, slides, and your code from previous home works and in class assignments. You can use the internet to search for answers. You are NOT allowed to use code provided by other students or solicit help from other people online.
4. Answer all the exam parts, all the parts are required.
5. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will loose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
6. Create a zip file which includes all the project folder and any required files. Submit the exported file using the provided canvas submission link.
7. **Do not try to use any Social Messenger apps, Email, or Cloud File Storage services in this exam.**
8. **Failure to follow the above instructions will result in point deductions.**
9. **Any violation of the rules regarding consultation with others will not be tolerated and will result disciplinary action and failing the course.**

Final Exam (100 Points)

This app will provide an online photo gallery that uses Firebase Authentication to manage users, Firestore to manage data storage and Firebase Storage to save the pictures from the users. This application should be implemented using activities/ fragments as indicated below:

- **AuthActivity:** this is the launcher activity and hosts the following fragments:
  - LogIn Fragment
  - SignUp Fragment
- **MainActivity:** which hosts the ViewPager and TabLayout, that will serve the following fragments:
  - MyPhotos Fragment
  - SharedWithMe Fragment
- **AddPhotoActivity:** which displays the add new photo form
- **PhotoActivity:** which hosts the following:
  - Photo Fragment
  - Share Fragment



(a) LogIn Fragment (b) SignUp Fragment (c) Main Activity (My Photos)

Figure 1, Application Wireframe

Part 1: (Auth Activity) Authentication and SignUp (5 Points)

The Auth Activity is the launcher activity and below are the requirements:

1. This activity should be configured to be the launcher of the application
2. If the user is already logged-in then the Main Activity should be started and the Auth Activity should be finished.
3. If the user is not logged-in then the Login Fragment should be displayed in the Auth Activity.

4. Login Fragment (Shown in Fig 1(a)):
    a. Clicking “Login” button, if all the inputs are entered correctly, you should attempt to login the user using Firebase. If the login is successful then trigger the Auth Activity to start the Main Activity and then finish the Auth Activity. If login is not successful show an alert dialog highlighting the error.
        i. If there is missing input, show an alert dialog indicating missing input.
    b. Clicking the “Create New Account” should replace this fragment with the SignUp Fragment.
5. SignUp Fragment (Shown in Fig 1(b)):
    a. Clicking the “Submit” button, if all the inputs are entered correctly, you should attempt to signup the user using Firebase. If the signup is successful then trigger the Auth Activity to start the Main Activity and then finish the Auth Activity. If signup is not successful show an alert dialog highlighting the error.
        i. If there is missing input show an alert dialog indicating the error.
    b. Clicking “Cancel” should replace this fragment with the Login Fragment.

Part 2: Main Activity (10 Points)

This activity should include a ViewPager and TabLayout as shown in Fig 1(c), this should allow the user to pick a tab item or swipe between the tab items. The requirements are listed below:

1. The TabLayout should include and serve the following options:
    a. My albums: displays the MyPhotos Fragment.
    b. Shared with me: displays the SharedWithMe Fragment
2. The activity should display “Logout” and “Add” menu items as icons as in Fig 1(c).
3. Clicking the “Add” menu button should:
    a. Take the user to the AddPhotoActivity
4. Clicking the “Logout” menu button should:
    a. Logout the currently logged in user.
    b. Then start the Auth Activity and then finish the Main Activity.

Part 3: (Main Activity) MyPhotos Fragment (15 Points)

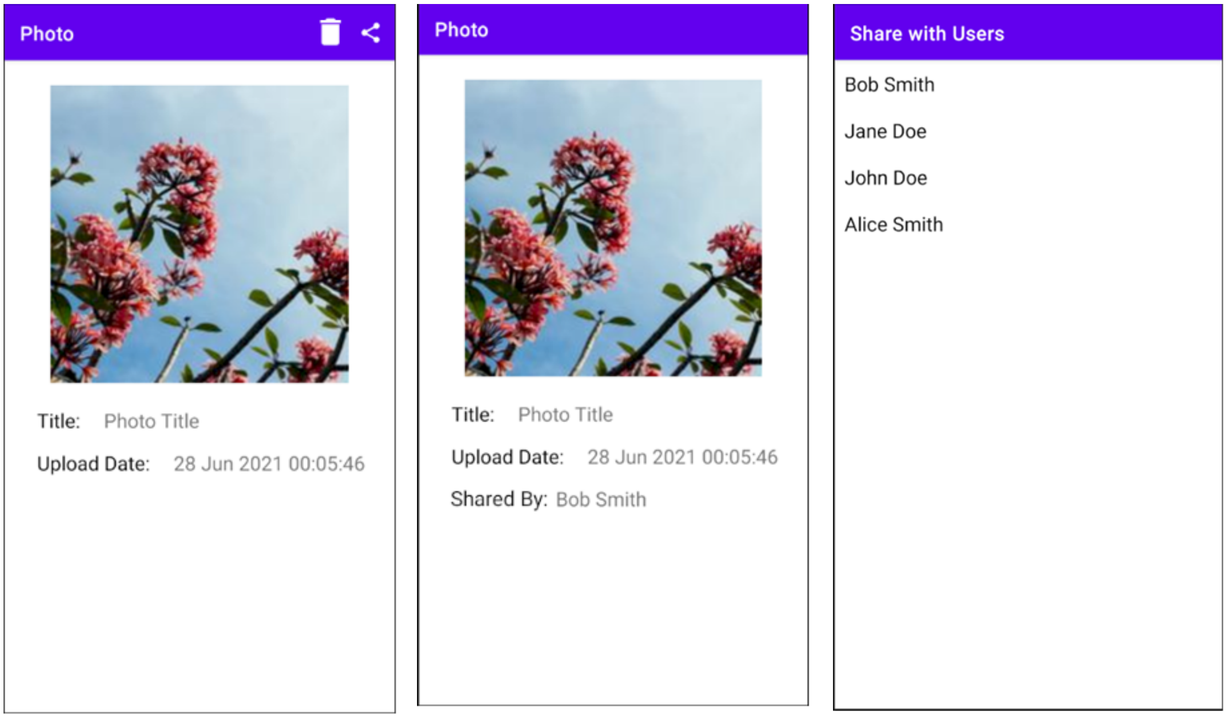
This fragment allows the user to see all the uploaded pictures. The requirements are listed below:

1. The MyPhotos Fragment is shown in Fig 1(c). The fragment provides a grid (hint: recycler views support grid structure instead of vertical/horizontal) to show all the pictures that were previously uploaded by the currently logged in user.
2. The images should be loading using a library for that (Picasso, Glide, etc.)
3. Upon clicking any of the shown images, it should take you to the PhotoFragment in the PhotoActivity.

Part 4: (Main Activity) SharedWithMe Fragment (15 Points)

This fragment displays the images the same way that the MyPhotos Fragment, with the difference that all the shown images should be only the ones that other users shared with the currently logged in user. The requirements are listed below:

1. This fragment should be displayed when the second tab is selected.
2. It will implement the same layout as the other fragment: a grid style.



Title: Photo Title

Upload Date: 28 Jun 2021 00:05:46

Title: Photo Title

Upload Date: 28 Jun 2021 00:05:46

Shared By: Bob Smith

Bob Smith

Jane Doe

John Doe

Alice Smith

(a) Photo Fragment, own image (b) Photo Fragment, shared image (c) Share Fragment

Figure 2, Application Wireframe

3. Clicking in an image should take you to the Photo Fragment in the Photo Activity for proper handling.

Part 5: AddPhoto Activity (25 Points)

This fragment enables the user to either select a picture from the gallery or take one with the camera and upload it to the user's own photos. The requirements are:

1. The AddPhoto Activity is shown in Fig 3(a). The fragment shows an empty image view with a placeholder that works as a button. When it's clicked, it goes to the respective action to select or take a picture.
2. When an image is selected from the camera or gallery, a preview of it should be shown in the image view, as shown in Fig 3(b).
3. Fig 4(a) shows the form for the user to enter the new photo name to be created.
    a. Clicking "Submit" button, if name input is not empty, the new photo should be created on Firestore and the corresponding image should be uploaded to Firebase Storage. Upon successful completion, should finish the current activity, which should show the previously displayed MainActivity.
        - Upon returning to the "MyPhotos" Fragment, the list should be refreshed to show the updated list of photos.
        - If there is missing input or any errors, show an alert dialog indicating the error or missing input.

Part 6: (Photo Activity) Photo Fragment (15 Points)

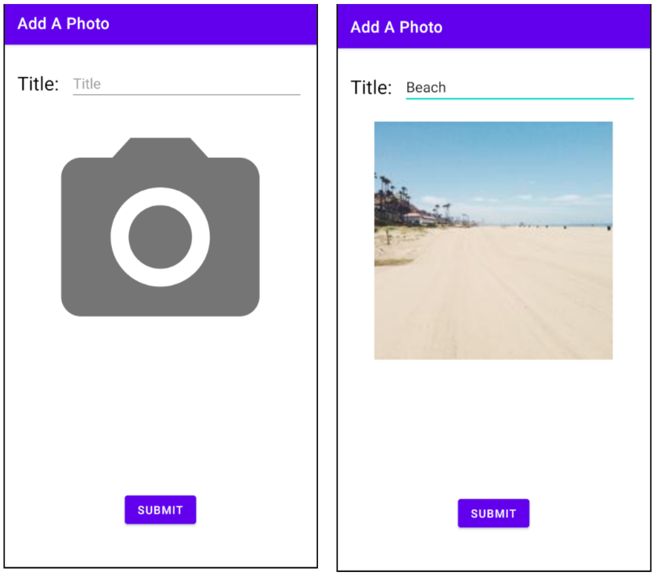
This fragment enables the currently logged in user to view the photo details. The requirements are listed below:

1. This fragment should be displayed when the user selects any of the images from either My Albums or Shared with me.
2. If the user owns the selected image, then:
   a. The fragment should show the image, title and upload date
   b. The action bar should have two buttons:
      i. Delete: which displays an alert dialog confirming the user choice to delete the selected image, and finishes the PhotoActivity to go back to the MainActivity.
      ii. Share: which changes the screen to the Share Fragment
3. If the user does not own the selected image, meaning that it was shared with the user, then:
   a. The fragment should show the image, title, upload date and the name of the owner of the image
   b. The action bar should have no options
4. Pressing the back button should finish the current activity which should show the previously displayed Main Activity.

Part 7: (Photo Activity) Share Fragment (15 Points)

This fragment displays the names of users to enable the selection of the user to share the photo with. The requirements are listed below:

1. The Share Fragment is shown in Fig 2(c). The fragment shows the names of the list of users on Firebase. You should consider creating a collection of users on Firestore to facilitate the listing of users.
2. When the user row item is clicked then the photo should be shared with the selected user through Firestore. Then the back stack should be popped which should show the Photo Fragment again.



Title: Beach

(a) Photo Fragment (no image selected)

(b) Photo Fragment (image selected)

Figure 3, Application Wireframe