Mobile App Development
Midterm Exam

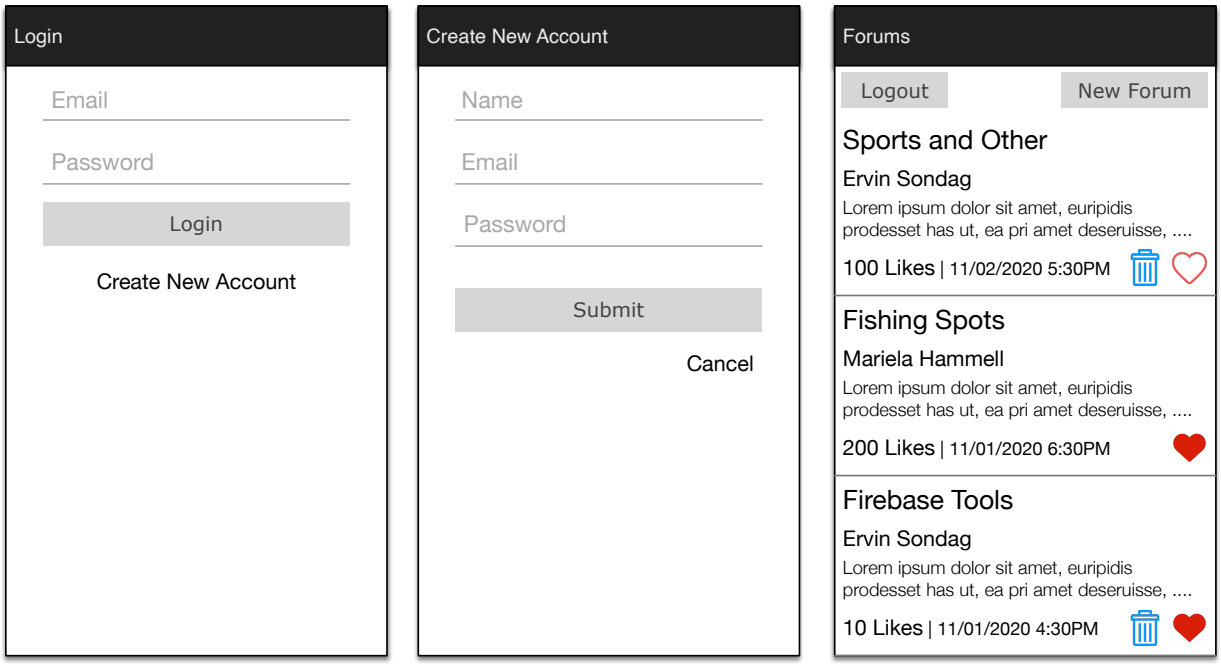
Basic Instructions:

1. This is the Midterm Exam, which will count for 20% of the total course grade.
2. This Midterm is an individual effort. Each student is responsible for her/his own Midterm and its submission.
3. Once you have picked up the exam, you may not discuss it in any way with anyone until the exam period is over.
4. 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 online persons.
5. Answer all the exam parts, all the parts are required.
6. Please download the support files provided with the Midterm and use them when implementing your project.
7. 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.
8. Create a zip file which includes all the project folder, any required libraries, and your presentation material. Submit the exported file using the provided canvas submission link.
9. **Do not try to use any Social Messenger apps, Emails, Or Cloud File Storage services in this exam.**
10. **Failure to follow the above instructions will result in point deductions.**
11. **Any violation of the rules regarding consultation with others will not be tolerated and will result disciplinary action and failing the course.**

Midterm Exam (100 Points)

In this assignment you will develop a Forums application, which enables users to view forums, create forums and comment on forums. You are provided with support files the include data and classes to use for this project.

1. You are provided with a DataServices class which should be imported in your project. The DataServices class should emulate the account and data management functions.
2. You **should not edit the DataServices class** and should use the methods provided.
3. **Note that all the DataServices methods should be called in the background and NOT on the main thread. You should use AsyncTasks for this assignment.**
4. All UI changes, updates and edits should be performed on the main thread only.
5. In this app you will have only one Activity and 5 fragments, all communication between fragments should be managed by the activity.



(a) Login Fragment (b) Register Fragment (c) App Categories Fragment

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. 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 calling the ***DataServices.login()*** method. ***Note that this method should be called on a child thread using AsyncTask.***
   b. If there is missing input, show a Toast indicating missing input.
   c. If the login is successful, then communicate the returned AuthResponse object to the activity and replace the current fragment with the Forums Fragment. Note that

that the AuthResponse contains both the token and the logged in Account object.
   d. If login is not successful, show a Toast message indicating that the login was not successful.
2. Clicking the “Create New Account” should replace this fragment with the Create New Account Fragment.

Part 2: Create New Account Fragment (5 Points)
The interface should be created to match the UI presented in Figure 1(b). The requirements are as follows:
1. This fragment should allow a user to create a new account. Upon entering the name, email and password, clicking the Submit button should:
   a. If all the inputs are not empty, you should attempt to signup the user by calling the ***DataServices.register()*** method. ***Note that this method should be called on a child thread using AsyncTask.***
   b. If the registration is successful, then send the returned AuthResponse object to the activity and replace the current fragment with the Forums Fragment.
   c. If the registration is not successful, show a Toast message indicating that the registration was not successful.
   d. If there is missing input, show a Toast indicating missing input.
2. Clicking “Cancel” should replace this fragment with the Login Fragment.

Part 3 : Forums Fragment (35 Points)
This screen enables the user to view the list of forums created by all the users and to interact with these forums. As shown in Figure 1(c), The requirements are as follows:
1. Clicking the “Logout” menu button should logout the currently logged in user and replace this fragment with the Login Fragment.
2. Clicking the “New Forum” button should:
   a. Replace the current fragment with the New Forum Fragment and Push the current fragment on the back stack.
3. The list of forums should be retrieved by calling the ***DataServices.getAllForums()*** method. ***Note that this method should be called on a child thread using AsyncTask.***
4. The forums list should be displayed using a RecyclerView as shown in Fig 1(c). Each row item should display a forum:
   a. Forum title, name of forum creator, only the first 200 characters of the forum description, the number of forum likes, the date/time the forum was created.
   b. **The trash “Delete”** icon should **only** be displayed for forum items that were created by the currently logged in user.
      i. If the user clicks the “Delete” icon the forum should be deleted by calling the ***DataServices.deleteForum()*** method, which should be called on a child thread using AsyncTask.
      ii. Upon returning from a successful delete the list of forums should be refreshed by calling the ***DataServices.getAllForums()*** method in the background and should refresh the RecyclerView rows to reflect the deletion.
   c. The like or unlike indicator which are represented by un-filled heart icon and filled heart respectively. Each Forum object contains a set of user accounts that have

liked the forum. If the currently logged in user has liked a forum then the filled heart should be displayed, and otherwise the un-filled heart icon should be displayed for that forum list item.

   i. The user can like the forum by clicking the un-filled heart icon, which should call the ***DataServices.likeForum()*** method, which should be called on a child thread using AsyncTask. Upon returning from a successful like the list of forums should be refreshed by calling the ***DataServices.getAllForums()*** method in the background and should refresh the RecyclerView rows to reflect this change.
   ii. The user can un-like the forum by clicking the filled heart icon, which should call the ***DataServices.unLikeForum()*** method, which should be called on a child thread using AsyncTask. Upon returning from a successful un-like the list of forums should be refreshed by calling the ***DataServices.getAllForums()*** method in the background and should refresh the RecyclerView rows to reflect this change.

5. Clicking on a row item should:
   a. Replace the current fragment with the Forum Fragment and Push the current fragment on the back stack.

Part 4 : New Forum Fragment (10 Points)

The interface should be created to match Figure 2. The requirements are as follows:

1. Clicking the “Cancel” button should:
   a. Pop the back stack which should return back to the Forums Fragment.
2. Clicking the “Submit” button, the app should check if all the fields are entered and display Toast message if any of the entries is missing indicating that the missing field is required.
   a. If all the fields are entered, the app should call the ***DataServices.createForum()*** method, which should be called on a child thread using AsyncTask. Upon returning from a successful forum creation pop the back stack to return back to the Forums Fragment which should be refreshed to show the newly added forum in the forums list.

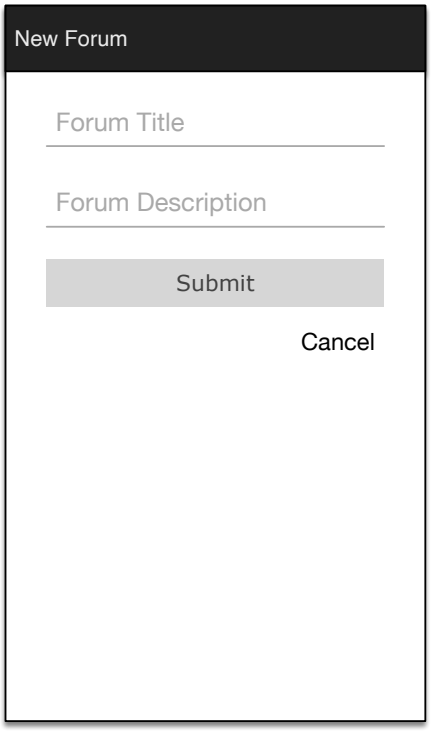


Figure 2, Create Forum

Part 5 : Forum Fragment (45 Points)

The Forum fragment enables the user to view the forum details and comments made by other users on this forum. In addition, this fragment enables the user to post comments on the forum. The requirements are as follows:

1. The Forum screen should display details about the selected forum, number of comments, and show the list of comments as shown in Fig 3.
2. At the top of the screen, the forum title, forum creator name, and forum description should be displayed.
3. The list of comments should be retrieved by calling the ***DataServices.getForumComments()*** method. ***Note that this method should be called on a child thread using AsyncTask.***
4. The list of comments should be displayed using a RecyclerView as shown in Fig 3. Each row item should display a comment:
   a. Name of comment creator, comment text, the date/time the comment was created.
   b. **The trash "Delete" icon should only be displayed for comment items that were created by the currently logged in user.**
5. If the user clicks the "Delete" icon the forum should be deleted by calling the ***DataServices.deleteComment()*** **method, which should be called on a child thread using AsyncTask.**
   a. Upon returning from a successful delete the list of comments should be refreshed by calling the ***DataServices.getForumComments()*** **method in the background and should refresh the RecyclerView rows to reflect the deletion.**

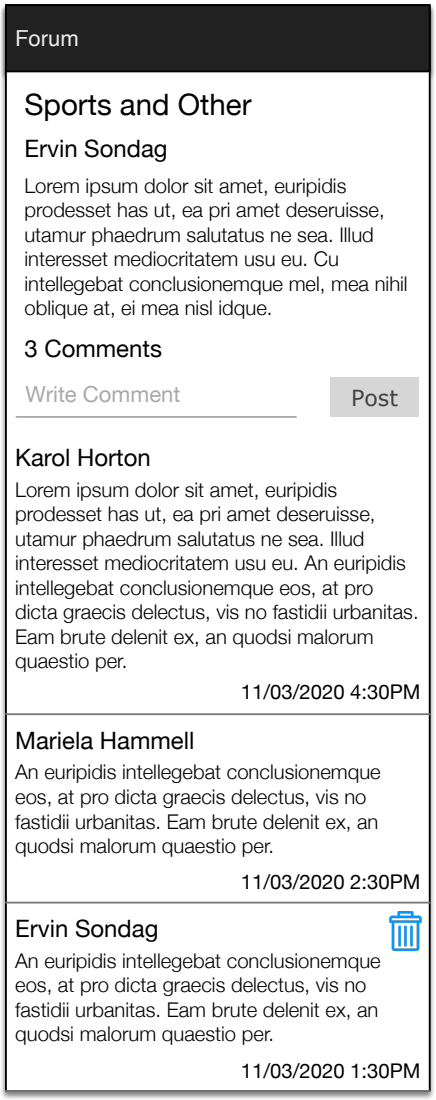


Figure 3, Forum Details

6. The number of comments should be shown under the description as shown in Fig 3.
7. Creating a comment:
   a. The write comment EditText should allow the user to enter a text comment. Clicking the "Post" button should check if there is a comment entered.
   b. If there is a comment entered then save the comment by calling the ***DataServices.createComment()*** **method, which should be called on a child thread using AsyncTask.**
   c. Upon returning from a successful comment creation the list of comments should be refreshed by calling the ***DataServices.getForumComments()*** **method in the background and should refresh the RecyclerView rows to reflect the comment addition.**
8. Pressing the back button should pop the back stack to show the Forums Fragment.

Item	Grade
Part 1: Login performed in a background using AsyncTask and transitioning correctly to other fragments through the Main Activity.	5
Part 2: Registration performed in a background using AsyncTask and transitioning correctly to other fragments through the Main Activity.	5
Part 3: Forums list retrieved in background thread and RecyclerView displayed as shown in Fig 1(c).	15
Part 3: Forum delete displayed correctly and delete done in background thread and forums retrieved in the background and RecyclerView refreshed.	10
Part 3: Forum like/unlike displayed correctly and like/unlike done in background thread and forums retrieved in the background and RecyclerView refreshed.	10
Part 4: Create new form performed in a background using AsyncTask and transitioning back to pop the stack upon successful creation.	10
Part 5: Forum details displayed correctly. The comments list retrieved in background thread and RecyclerView displayed as shown in Fig 3.	15
Part 5: Comment delete displayed correctly and delete done in background thread and comments retrieved in the background and RecyclerView refreshed.	15
Part 5: Create new comment done in background thread and comments retrieved in the background and RecyclerView refreshed.	15
Total	**100**