Mobile Application Development
Final Exam

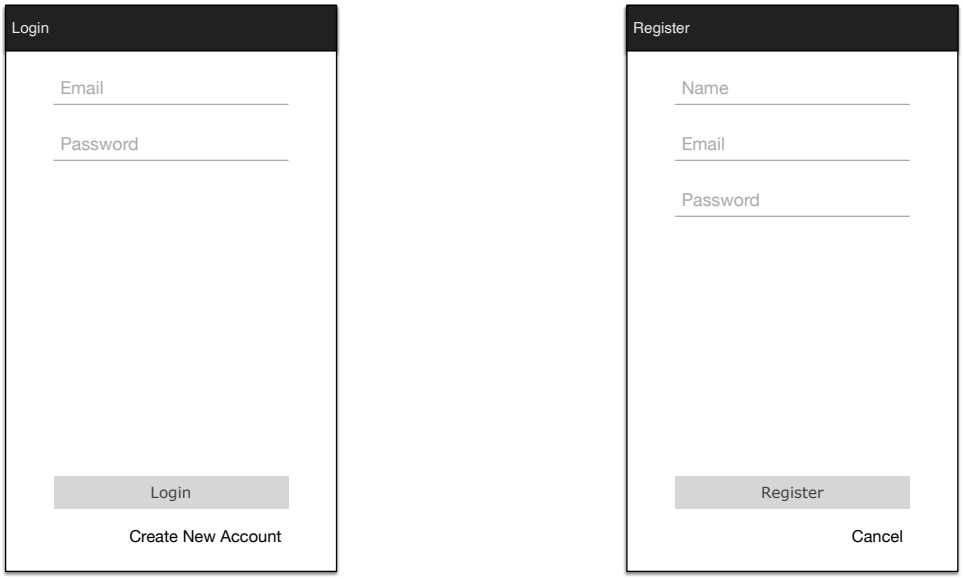
Basic Instructions:

1. This is the Final Exam, which will count for 15% of the total course grade.
2. This Final Exam is an individual effort. Each student is responsible for her/his own Final Exam 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. During the Final Exam the teaching assistants and Instructors will pass by each student and ask them to demonstrate their application. Your interaction with the teaching assistants and instructors will be taken into consideration when grading your exam submission.
7. Please download the provided support files provided with the Final Exam and use them when implementing your project.
8. 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.
9. It is required to demo each of the features listed in the rubric doing the exam in order to get the corresponding grade for each feature.
10. **Do not try to use any Social Messenger apps, Emails, Or Cloud File Storage services in this exam.**
11. **Failure to follow the above instructions will result in point deductions.**
12. **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)

In this assignment you will develop News app the allows the user to view the technology news, create news lists, add news to lists, and view news details and lists.

1. You are provided with the zip file which includes a starter Android project that implements Login, Registration and has all the required layout files.
   a) Create a Firebase project, add an Android app with package name **"edu.uncc.finalexam"**
   b) Replace the **"google-services.json"** to include your file instead.
   c) All the required libraries are already included in the provided project.
2. Use the OkHttp library in this app in order to make all the http connections and API calls. All the data returned by the APIs is in JSON format.
3. All the network calls should be done in a background thread. All UI changes, updates and edits should be performed on the main thread.
4. Use Firebase for authentication and for storing data on Firestore to manage the storage of the user's news lists and to retrieve these lists.
5. This app will have one Activity and 7 Fragments, **all communication between fragments should be managed by the activity.**



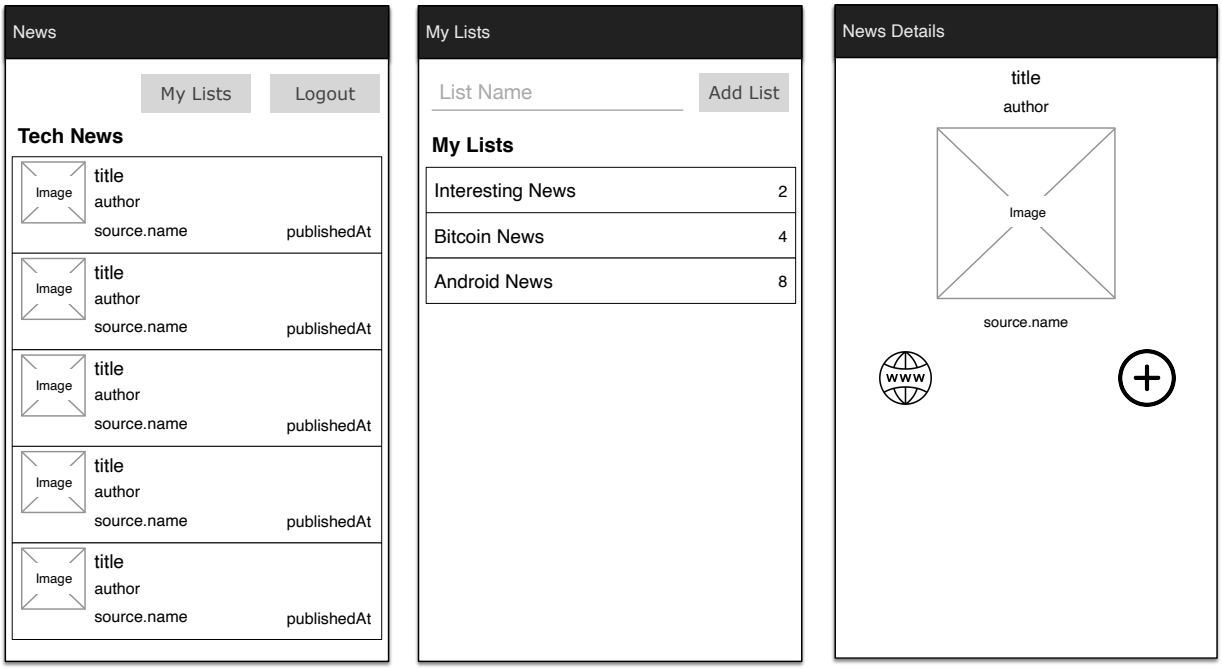
(a) Login Fragment (b) Register Fragment

Figure 1, App Wireframe

Part 0 : Auth Screens (Login and Register) (0 Points)

The login and register fragments are already implemented in the provided starter project and are shown in Figure 1.

1. Upon successful login/registration the fragment will communicate with the Main Activity to replace the current fragment with the NewsFragment.



(a) News Fragment (b) My Lists Fragment (c) News Details Fragment

Figure 2, App Wireframe

Part 1 : News Fragment (25 Points)

This fragment is shown in Figure 2(a). The requirements are as follows:

1. **The starter app already includes the implementation of the "Logout" feature**, where clicking the "Logout" button should logout the user and replace the current fragment with the Login Fragment.
2. Clicking the "My Lists" button should:
   a. Replace the current fragment with the "My Lists" Fragment.
   b. Push the current fragment on to the back stack.
3. Use the OkHttp library to retrieve the top headlines for the technology news items by making a GET request to the below URL: https://www.theappsdr.com/news_api.json
   a. Create a News class and parse the returned list of news into an ArrayList containing the parsed News objects. Display the returned news items in a ListView as shown in Figure 2(a).
   b. Each row item should display the new title, author, source name, published at date formatted as ("dd/MM/yyyyy hh:mm a") and news image. Use the Picasso Library to display the news image.
4. Clicking on a list row item should:
   a. Send the selected news item to the Main Activity which should Replace the current fragment with the News Details Fragment shown in Figure 2(c).
   b. Push the current fragment on to the back stack.

Part 2 : My Lists Fragment (20 Points)

This fragment is shown in Figure 2(b). The requirements are as follows:

1. Register a snapshot listener on Firestore to retrieve the currently logged in user's

news lists. News lists are lists created by a user to store news items. It is like news favorite lists or reading lists.
   a. Create a NewsList class that should be used to store the news list item. Display the news list objects retrieved from Firestore in ListView as shown in Figure 2(b).
   b. Each row item should display the news list name, and the number of news items included in this list.
2. Clicking on a list row item should:
   a. Send the selected news list item to the Main Activity which should Replace the current fragment with the List Details Fragment.
   b. Push the current fragment on to the back stack.
3. This fragment also allows the user to create a new news list.
   a. The user should enter a news list name.
   b. Clicking “Add List” button should check if the list name is not empty, and should display an error if it is empty.
      i. If list name is not empty, then the new list should be stored on Firestore.
      ii. The ListView of new list items should automatically refresh to show the newly added news list time, which is automated by the snapshot listener.
4. **Note that each user should have their own new list items, so the displayed news list items belong to the currently logged in user.**

Part 3 : News Details Fragment (15 Points)

This fragment is shown in Figure 2(c). The implemented requirements are as follows:

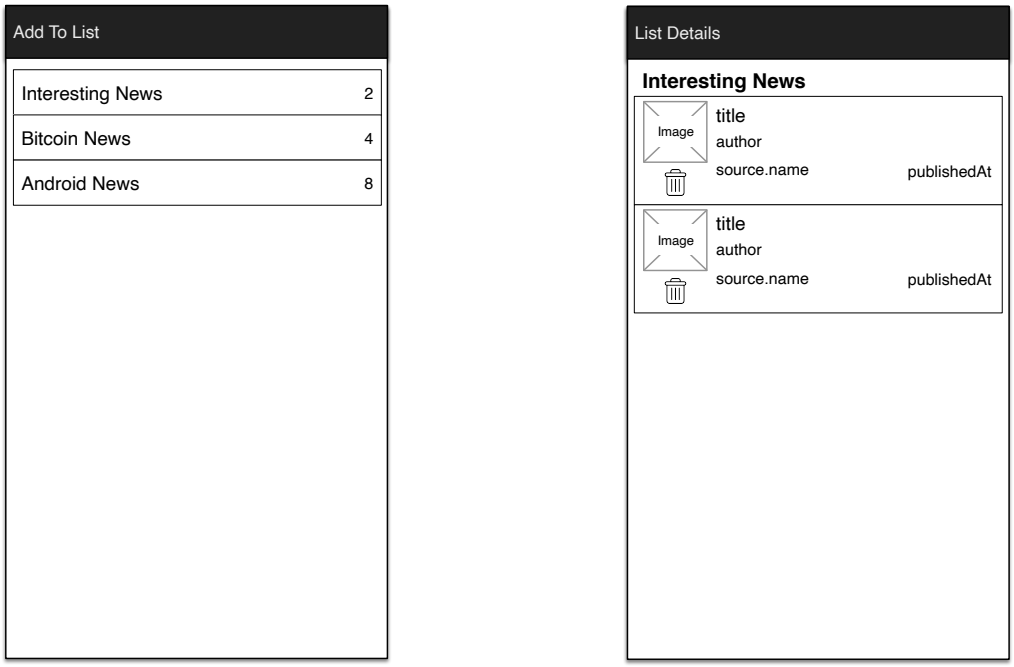
1. Display the new title, author name, news image, and source name. Use the Picasso Library to display the news image.
2. Clicking the “Globe” icon should open the news url by using the **“openNewsUrl”** method that is already provided in the NewsDetailsFragment class. This should open the news url, which is one of the parameters returned in the API JSON.
3. Clicking the “Plus” icon should:
   a. Send the news item to the Main Activity which should Replace the current fragment with the Add To List Fragment.
   b. Push the current fragment on to the back stack.

Part 4 : Add To List Fragment (20 Points)

This fragment is shown in Figure 3(a). The requirements are as follows:

1. The fragment should receive the news item that should be added to one of the currently logged user’s news lists.
2. Retrieve from Firestore the currently logged in user’s news lists.
   a. Display the news list objects retrieved from Firestore in ListView as shown in Figure 3(a).
   b. Each row item should display the news list name, and the number of news items included in this list.
3. Clicking on a list row item should:
   a. Store (add) the news item to the selected news item on Firestore.
   b. Should communicate with the Main Activity which should pop the back stack, which should result in going back to the News Details Fragment.
4. Note that by adding the news item to the selected news item this means that this

news item is now stored under that list. If the user visits the My Lists fragment, this news list item count is now incremented by 1. Also if you visit the List Details fragment it should show this news item as one of the news items under this news list.



(a) Add To List Fragment (b) List Details Fragment

Figure 3, App Wireframe

Part 5 : List Details Fragment (20 Points)

This fragment is shown in Figure 3(b). The requirements are as follows:

1. The Main Activity should send this fragment the news list item that will be displayed.
   a. The name of the news list item should be displayed at the top of the fragment.
2. Register a snapshot listener on Firestore to retrieve the news items for this news list:
   a. Parse the returned list of news items into an ArrayList containing the parsed News objects. Display the returned news items in a **RecyclerView** as shown in Fig 3(b).
   b. Each row item should display the new title, author, source name, published at date formatted as ("dd/MM/yyyyy hh:mm a") and news image. Use the Picasso Library to display the news image.
3. Clicking on a list row item should:
   a. Send the selected news item to the Main Activity which should Replace the current fragment with the News Details Fragment.
   b. Push the current fragment on to the back stack.
4. Clicking the row item delete button should:
   a. Delete the selected news items from the news list on Firestore.
   b. The RecyclerView of news items should automatically refresh to show reflect the removed news item, which is automated by the snapshot listener.

Section:	
Student Name:	
Student ID:	

Part #	Item	Total	Grade	Comments
P1	API call and JSON Parsing to get the News Items	10		
P1	Display the ListView and row items correctly	5		
P1	Handle on news row item click and correct navigation	5		
P1	Handle button click “My Lists” and correct navigation	5		
P2	Validation and store new list on Firebase	10		
P2	Firebase snapshot listener and display ListView and row items with correct count	10		
P3	Show news information and image correctly	5		
P3	Handle globe click correctly	5		
P3	Handle add click and correct navigation	5		
P4	Firebase get items and display ListView and row items	10		
P4	Handle on row item click, add to firebase and correct navigation	10		
P5	Firebase get items and display RecyclerView and row items	10		
P5	Handle on news row item click and correct navigation	5		
P5	Handle delete of news item, Firebase delete and refresh the RecyclerView	5		
	Total	**100**		