Unit

University Social Media

Product Design Specifications Document | Version 1.3

February 25, 2020

Team: Adeel Asghar, Tyler Gross, Palak Patel, and Hala Ali

Clients: Thomas Anter and Jahnu Best

GTA: Son Dang

VERSION HISTORY

Version	Author	Date	Description
1.0	Hala Ali	2/10/20	Initial Design Definition draft
1.1	Adeel Asghar	2/21/20	Formatting
1.2	Adeel, Hala, Tyler, Palak	2/24/20	Added diagrams and finished document. Completion of Design Document Final Draft.
1.3	Adeel, Hala, Tyler, Palak	4/12/20	Added additional test cases, changed app name, and added logo. Updated diagrams

TABLE OF CONTENTS

1. INTRODUCTION

The introduction section of the Product Design Specification Document will cover the purpose of developing such a document and the significance the document holds for all parties involved. This includes the development team, the client, and other stakeholders.

1.1 Purpose of the Product Design Specification Document

The Product Design Specification document documents and tracks the necessary information required to effectively define architecture and system design. It provides the development team with guidance on the architecture of the system to be developed. The Product Design Specification document is created during the Planning Phase of the project. Its intended audience is the project manager, project team, and development team. Some portions of this document such as the user interface (UI) may on occasion be shared with the client/user, and other stakeholder whose input/approval into the UI is needed.

2. GENERAL OVERVIEW & DESIGN GUIDELINES/APPROACH

This section describes the principles and strategies to be used as guidelines when designing and implementing the system.

2.1 Assumptions / Constraints / Standards

2.1.1 Assumptions

1. The users have an android mobile device which runs an operating system of at least Android 5.0 or an emulator with an API level of 21. In the case of the user not having access to either of the mentioned technologies, our product will not run.
2. The user must be in ownership of a google email address and have knowledge of their username and password. The system currently supports users resetting their passwords by sending users a link via email. If the user does not have access to their email, they will not be able to recover their account.

2.1.2 Technical Constraints

1. The app is targeted for Android devices.
2. MVVM is a technical constraint.
3. Firebase limits nonpaying users to 1Gb of stored data, 50,000 document reads, 20,000 document writes, and 20,000 document deletes.

2.1.3 Business Constraints

1. The Firebase Realtime Database server must always be available.
2. Due to the self-funded nature of the project, there will be no funds allocated to buying the Firebase Realtime database spark plan. In recognition of this the application must have under 100 simultaneous connections, less than 1GB of data stored, and under 10 GBs downloaded per month. The application must also only have one Firebase database
3. The requirements for this project must be achieved by 4/14/2020.

2.1.4 Design Constraints

1. The application design should adhere to the MVVM architectural design pattern. Data from the database will flow through models, repositories, view models, and into the views.
2. In the case of dependent classes, the application should adhere to the dependency inversion principle. All class dependencies should be passed from the outside of functions using dependency injection. To implement dependency injection the team will be using the Dagger dependency injection framework

3. Data from the database that is displayed to the user as well as data taken in by the user interface will be bound between the view models and the XML layout files. The activities in the application will not contain any data related functions.

2.1.3 Standards

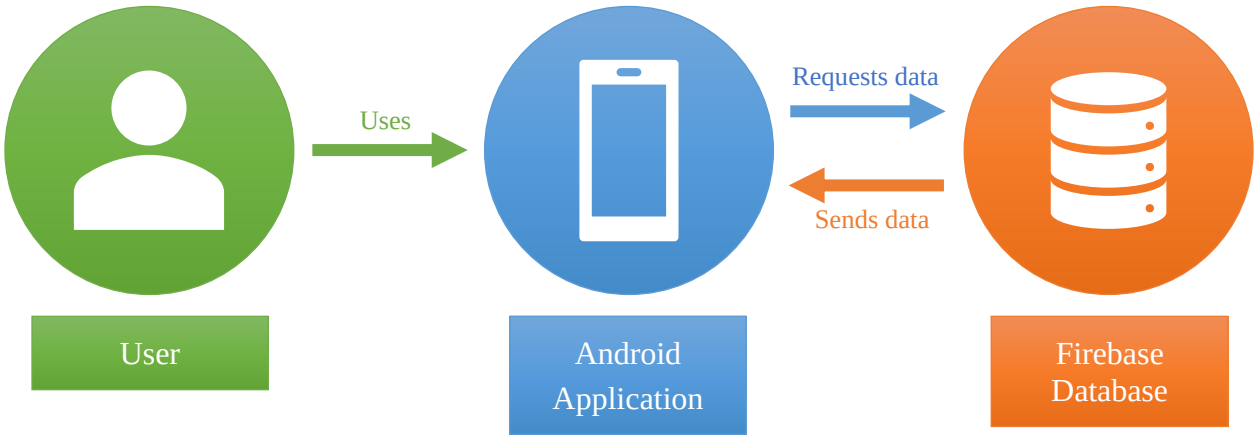
1. All features of the application must be thoroughly tested before release.
2. The UI of the application must be easily usable by the products target demographic.

3. ARCHITECTURE DESIGN

This section outlines the University Collaborative Forum system and hardware architecture design of the system that is being built.

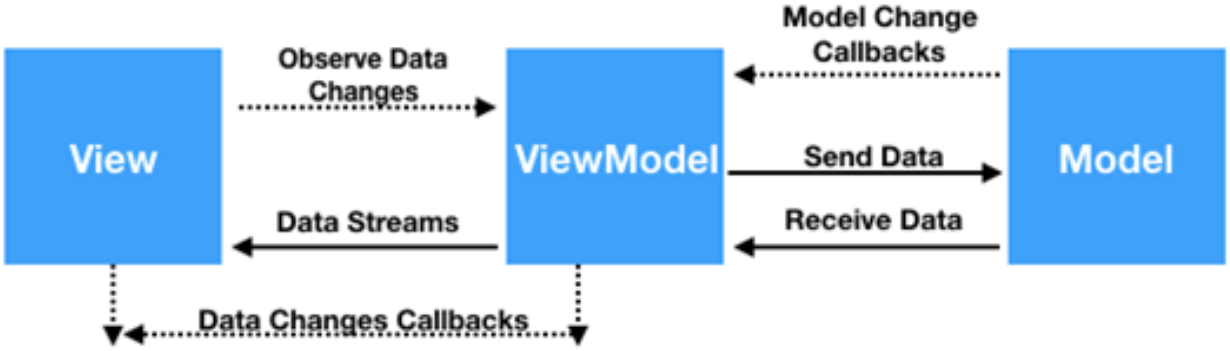
3.1 Hardware Architecture

The application will have 3 main components: the user, the Android device, and Firebase Realtime Database. The user will use the Android Device to open the Collaborative University Application. The application will request authentication or data from the Firebase Realtime Database which will send back that authentication or data to display in the application. The following diagram displays the University Collaborative Application interactions within a larger system.



3.2 Software Architecture

The following diagram displays the architectural design of the University Collaborative Application. The software architectural pattern used is MVVM (Model-View-View Model). The three components of the MVVM are models, views and view models. The View is displayed to the user. The view model passes data to and from the view to the model. The model contains data containers that are populated with data from the database.

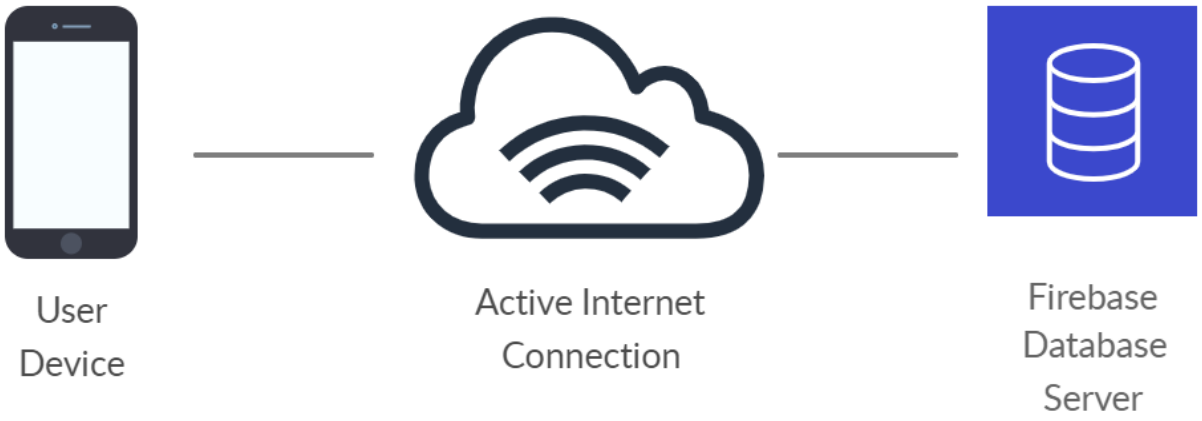


3.3 Security Architecture

The security architecture for this application is done through Firebase Authentication which supports registration and login for valid users. Registration is done using an email, username, and password. These passwords are hashed through a Firebase script. Login can then be done using a username and password which are passed to the Firebase Authentication SDK. The SDK verifies those credentials using its backend services and returns a response to our client. Users can then read and write to the database but only delete posts that they have made themselves.

3.4 Communication Architecture

The communication architecture of this application is dependent on the connection to the internet of the cellar device being used to run the application. The application must be connected to the internet to access the Firebase Realtime Database. The application depends on Firebase for both user authentication and storage.



3.5 Performance

The time it takes for the app to open once a user has clicked on it will be 200 milliseconds. After a user registers their information correctly and hits register button, it will take anywhere between 200 to 500 milliseconds to redirect to the login page. Once the user has typed in their email and password and hits login button, it should take anywhere between 500 milliseconds to 1 second to direct user to main forum page.

4. SYSTEM DESIGN

The following section goes over the use cases, sequence diagrams, dataflow diagrams, database design, class diagram, and application program interface.

4.1 Use-Cases

4.1.1 Use Case 1: User Registration	
Created by	Palak Patel
Date Created	2/18/2020
Last Updated by	Palak Patel
Date of last revision	4/12/2020
Actors	Application User
Description	This use case will describe a user registering for an account.
Trigger	The user wants to register for an account, so they click the register button.
Preconditions	The user does not already have an account.
Post conditions	The User's credentials are added to Firebase Authentication and Database.
Normal Flow	1. The user selects the I do not have an account button. 2. The user enters their username, email and password. 3. The user selects the register button. 4. the user is redirected to the login page.
Alternative Flows	Alt Flow 1. Username formatting parameters are not met. 1. The user enters invalid information. 2. The user is prompted with an error message, returns to UC-1. Alt Flow 2. Email formatting parameters are not met. 1. The user enters invalid information. 2. The user is prompted with an error message, returns to UC-1.

	Alt Flow 3. Password formatting parameters are not met. 1. The user enters invalid information. 2. The user is prompted with an error message, returns to UC-1. Alt Flow 4. User already has an account with that email address. 1. The user enters an email that is already in use. 2. The user is prompted with an error message, returns to UC-
Frequency of Use	Once per account.
Assumptions	There is an active internet connection and the Firebase Realtime Database is available.

4.1.2 Use Case 2: User Login	
Created by	Palak Patel
Date Created	2/18/2020
Last Updated by	Palak Patel
Date of last revision	4/12/2020
Actors	Application User
Description	This use case will describe a user logging into the application.
Trigger	The user wants to login to their account, so they click the login button.
Preconditions	The user has not already logged in.
Post conditions	The user is now logged and directed to the main forum screen.
Normal Flow	1. The user enters their username and password. 2. The user selects the login button. 3. The user is redirected to the main forum page.
Alternative Flows	Alt Flow 1. The user's email cannot be found by Firebase Authentication. 1. The user enters invalid Email. 2. They are prompted with an error message, returns to UC-2. Alt Flow 2. The user inputs an incorrect password 1. The user enters invalid password. 2. They are prompted with an error message, returns to UC-2
Frequency of Use	Every time a user logs out of the application.
Assumptions	There is an active internet connection and Firebase Authentication is Available

4.1.3 Use Case 3: Subscribe to a Class Forum	
Created by	Palak Patel
Date Created	2/18/2020
Last Updated by	Palak Patel
Date of last revision	2/24/2020
Actors	Application User
Description	This use case will describe a user subscribing to a class forum, so they select the subscribe button.
Trigger	The user wants to subscribe to a class forum.
Preconditions	The user is logged in and is not already subscribed to the forum they wish to unsubscribe from.
Post conditions	The posts from the forum that the user subscribed to will now show on their main forum page. The subscription is also added in Firebase subscription list for user.
Normal Flow	1. The user navigates to the all classes screen. 2. The user selects the subscribe button for their chosen class that they are not already subscribed to.
Alternative Flows	N/A
Frequency of Use	This will be used whenever a user would like to subscribe to a forum.
Assumptions	There is an active internet connection and the Firebase Realtime Database is available.

4.1.4 Use Case 4: Unsubscribe from Class	
Created by	Palak Patel
Date Created	2/18/2020
Last Updated by	Palak Patel
Date of last revision	2/24/2020
Actors	Application User
Description	This use case will describe a user unsubscribing for a class.
Trigger	This user wants to unsubscribe from a class forum, so they click the unsubscribe button.
Preconditions	The user is logged in and subscribed to the class that they wish to unsubscribe from.
Post conditions	The posts from the unsubscribed forum will disappear from that user's main forum page. Subscription removed from Firebase subscription list for user.
Normal Flow	1. The user navigates to the all classes screen. 2. The user selects the un-subscribe button for their chosen class that they are subscribed to.
Alternative Flows	N/A
Frequency of Use	Every time the user wishes to no longer view posts from a specific forum.
Assumptions	There is an active internet connection and the Firebase Realtime Database is available.

4.1.5 Use Case 5: Create Post	
Created by	Palak Patel
Date Created	2/18/2020
Last Updated by	Palak Patel
Date of last revision	4/12/2020
Actors	Application User
Description	This use case will describe a user creating a post.
Trigger	The user wants to create a post, so they click the create post button.
Preconditions	The user is logged in and has navigated to the main forum page.
Post conditions	The post will show up on the user's main forum page, their profile page as well as in the forum that they posted to. The post is added in Firebase in specified forum.
Normal Flow	1. The user navigates to the create post screen. 2. The user chooses the forum that the wish to create their post in. 3. The user chooses the class to make the post to 4. The user enters the title of their post. 5. The user enters the content of their post. 6. The user selects the add post button.
Alternative Flows	Alt Flow 1. The user leaves the title field blank 1. The user navigates to the create post screen. 2. The user chooses the forum that they wish to create their post. in. 3. The user chooses a class to post to. 4. The user enters a post body. 5. The user clicks the add post button. 6. The user is prompted with an error toast message clarifying their mistake and returns to UC 5. Alt Flow 2. The user leaves the text field blank 1. The user navigates to the create post screen.

	2. The user chooses the forum that they wish to create their post in. 3. The user chooses a class to post to. 4. The user enters a post title. 5. The user clicks the add post button. 6. The user is prompted with an error toast message clarifying their mistake and returns to UC 5. Alt Flow 2. The user tries to post to a class that they are not subscribed to 1. The user navigates to the create post screen. 2. The user chooses the forum that they wish to create their post in. 3. The user chooses a class they are not subscribed to, to post to. 4. The user enters a post body and title. 5. The user clicks the add post button. 6. The user is prompted with an error toast message clarifying their mistake and returns to UC 5.
Frequency of Use	Every time the user wishes to make a post.
Assumptions	There is an active internet connection and the Firebase Realtime Database is available.

4.1.6 Use Case 6: Delete Post	
Created by	Palak Patel
Date Created	2/18/2020
Last Updated by	Palak Patel
Date of last revision	2/24/2020
Actors	Application User
Description	This use case will describe a user deleting a post.
Trigger	The user wants to delete a post, so they select the delete post button.
Preconditions	The user must be logged as well as successfully created the post that they wished to delete.
Post conditions	The post will no longer appear in the forum it was posted to, the users main forum screen or their user profile screen. Post deleted from Firebase in specified forum.
Normal Flow	1. The user navigates to their profile screen. 2. The user located the post they wish to delete by scrolling through their posts on their profile. 3. The user selected the delete post button. 4. The post is no longer visible because it has been deleted.
Alternative Flows	N/A
Frequency of Use	Every time the user decides to delete a post.
Assumptions	There is an active internet connection and the Firebase Realtime Database is available.

4.1.7 Use Case 7: Create Comment	
Created by	Palak Patel
Date Created	2/19/2020
Last Updated by	Palak Patel
Date of last revision	4/12/2020
Actors	Application User
Description	This use case will describe a user adding a comment to a post.
Trigger	The user wants to add a comment, so they select the add comment button.
Preconditions	The user must be logged in.
Post conditions	A new comment will appear under the post that the user decided to comment on. Comment added in Firebase in specified post.
Normal Flow	1. The user navigates to their main forum screen or to a specific forum. 2. The user locates the post by scroll through the list on any of the previously described screens. 3. The user clicks on the post. 4. The user selects add comment. 5. The user is prompted with a text field and they enter their comment. 6. The user selects post comment.
Alternative Flows	Alt Flow 1. User tries to post an empty comment 1. The user navigates to their main forum screen or to a specific forum. 2. The user locates the post by scroll through the list on any of the previously described screens. 3. The user selects a post. 4. The user selects add comment. 5. The user is prompted with a text field and they don't enter a comment. 6. The user selects post comment.

	7. User is prompted with an error message and returns to UC-7 Alt Flow 2. User tries to post a comment to a forum that they are not subscribed to. 1. The user navigates to a specific forum that they are not subscribed to. 2. The user locates the post by scroll through the list on any of the previously described screens. 3. The user selects a post. 4. The user selects add comment. 5. The user is prompted with a text field and they don't enter a comment. 6. The user selects post comment. 7. User is prompted with an error message and returns to UC-7
Frequency of Use	Every time the user wishes to comment on their own post or another student's post.
Assumptions	There is an active internet connection and the Firebase Realtime Database is available.

4.1.8 Use Case 8: Delete Comment	
Created by	Palak Patel
Date Created	2/19/2020
Last Updated by	Palak Patel
Date of last revision	4/12/2020
Actors	Application User
Description	This use case will describe a user deleting a post comment.
Trigger	The user wants to delete a comment, so they click the delete comment button.
Preconditions	The user must be logged in and have successfully made the post that they wish to delete.
Post conditions	A comment that the user intended to delete will be removed from all screens displaying that comment. Comment removed from Firebase in specified post.
Normal Flow	1. The user navigates to their profile screen, main forum screen or to a specific forum. 2. The user locates a comment that they made wish to delete by scrolling through the list. 3. The user selects delete comment. 4. The comment is no longer visible anywhere in the application
Alternative Flows	N/A
Frequency of Use	This will be used every time the user wishes to delete their own comment on a post.
Assumptions	There is an active internet connection and the Firebase Realtime Database is available.

4.1.9 Use Case 9: Searching for a Forum	
Created by	Palak Patel
Date Created	2/19/2020
Last Updated by	Palak Patel
Date of last revision	4/12/2020
Actors	Application User
Description	This use case will describe a user searching for a forum.
Trigger	The user wants to search for a forum, so they click the search button after they enter the subject and title names.
Preconditions	The user must be logged in.
Post conditions	The forum that the user searched for will open on the screen
Normal Flow	1. The user navigates to the all classes search screen. 2. The user starts to enter the subject name. 3. As the user types, classes that correlate to what they typed show up in the search list. 4. The user finishes typing. 5. Only the class they were searching for remains
Alternative Flows	Alt Flow 1. User enters the subject or class name incorrectly 1. The user navigates to the all classes page. 2. The begins the search by entering the subject name. 3. The user enters incorrect class title name. 4. If the user entered the class title incorrectly, nothing shows up in the search list, return to UC-9 Alt Flow 2. The user only partially types in the class or subject name 1. The user navigates to the all classes search screen. 2. The user starts to enter the subject name. 3. As the user types, classes that correlate to what they typed show up in the search list. 4. The user finishes typing without typing the full class name.

	5. All classes that correlate to what the user typed show up in the search list.
Frequency of Use	This will be used every time the user wishes to search for a class.
Assumptions	There is an active internet connection and the Firebase Realtime Database is available.

4.1.10 Use Case 10: Reset Password	
Created by	Palak Patel
Date Created	2/19/2020
Last Updated by	Palak Patel
Date of last revision	4/12/2020
Actors	Application User
Description	This use case will describe a user resetting their password.
Trigger	The user wants to reset their password, so they select the reset password button.
Preconditions	The user must already have an account registered with an email address that they can log on to.
Post conditions	The user will be able to log in with their new password. Password reset in Firebase Authentication.
Normal Flow	1. The user navigates to reset password screen. 2. The use enters their email address. 3. The user logs into the email address they just entered and selects the email from Firebase. 4. The user clicks on the link in the email and is navigated to a reset password page. 5. The user enters their new password.
Alternative Flows	Alt Flow 1. The user enters an invalid email 1. The user navigates to the reset password screen. 2. The user enters an invalid email. 3. The user is prompted with an error message to re-enter their email. 4. Return to UC-1.
Frequency of Use	This will be used every time the user forgets their password.
Assumptions	There is an active internet connection and the Firebase Realtime Database is available.

4.1.11 Use Case 11: Change Profile Photo	
Created by	Palak Patel
Date Created	2/19/2020
Last Updated by	Palak Patel
Date of last revision	4/12/2020
Actors	Application User
Description	This use case will describe a user changing their profile picture.
Trigger	The user wants to change their profile photo so the select the change username button in the edit profile activity.
Preconditions	The user must log in as well as have the photo they want to use saved to the camera roll of their phone.
Post conditions	The new chosen profile photo will show up in the navigation drawer.
Normal Flow	1. The user navigates to profile screen using the navigation drawer. 2. The user selects the edit profile button. 3. The user selects the profile image button. 4. The user selects a photo and is navigated back to the edit screen. 5. The user selects the done button. 6. The user is navigated back to the profile screen and the new profile image is displayed.
Alternative Flows	N/A
Frequency of Use	This will be used every time the user wants to change their profile picture.
Assumptions	There is an active internet connection and the Firebase Realtime Database is available.

4.1.12 Use Case 12: Change Username	
Created by	Palak Patel
Date Created	4/12/2020
Last Updated by	Palak Patel
Date of last revision	4/12/2020
Actors	Application User
Description	This use case will describe a user changing their username.
Trigger	The user wants to change their username, so they type in a new username and click the done button located in the edit activity screen.
Preconditions	The user must log in.
Post conditions	The new chosen username will show up on every post and comment the user previously made as well as everywhere else in the application that displays usernames.
Normal Flow	1. The user navigates to profile screen using the navigation drawer. 2. The user selects the username text box. 3. The user types in their new username. 4. The user selects the done button. 5. The user is navigated back to the profile screen and the new profile username is displayed.
Alternative Flows	N/A
Frequency of Use	This will be used every time the user wants to change their profile picture.
Assumptions	There is an active internet connection and the Firebase Realtime Database is available.

4.1.13 Use Case 13: Change User Bio	
Created by	Palak Patel
Date Created	4/12/2020
Last Updated by	Palak Patel
Date of last revision	4/12/2020
Actors	Application User
Description	This use case will describe a user changing their bio.
Trigger	The user wants to change their bio, so they type in a new username and click the done button located in the edit activity screen.
Preconditions	The user must log in.
Post conditions	The new chosen bio is displayed on the user's user profile and the old bio will not.
Normal Flow	1. The user navigates to profile screen using the navigation drawer. 2. The user selects the bio text box. 3. The user types in their new username. 4. The user selects the done button. 5. The user is navigated back to the profile screen and the new profile username is displayed.
Alternative Flows	N/A
Frequency of Use	This will be used every time the user wants to change their profile picture.
Assumptions	There is an active internet connection and the Firebase Realtime Database is available.

4.1.14 Use Case 14: Edit Post	
Created by	Adeel Asghar
Date Created	4/12/2020
Last Updated by	Adeel Asghar
Date of last revision	4/12/2020
Actors	Application User
Description	This use case will describe a user editing one of their posts.
Trigger	The user wants to edit their post, so they type in a new title and text and click the save button located in the Edit Post activity screen.
Preconditions	The user must log in and must be the author of the post.
Post conditions	The edited post will show on the user profile, community page, and home fragment.
Normal Flow	1. The user navigates to profile screen using the navigation drawer. 2. The user swipes left on post they want to edit. 3. The user edits the title and/or text. 4. The user selects the save button. 5. The user is navigated back to the profile screen and the edited post is displayed.
Alternative Flows	Alt Flow 1. The user edits a post to have an empty title 1. The user navigates to profile screen using the navigation drawer. 2. The user swipes left on post they want to edit. 3. The user edits the title to be blank 4. The user selects the save button. 5. The user prompted with an error message that lets them know that the title should not be blank, return to UC-14. Alt Flow 2. The user edits a post to have an empty post body 1. The user navigates to profile screen using the navigation drawer.

	2. The user swipes left on post they want to edit. 3. The user edits the text to be blank 4. The user selects the save button. 5. The user prompted with an error message letting them know that they cannot leave the text empty, return to UC-14.
Frequency of Use	This will be used every time the user wants to edit one of their posts.
Assumptions	There is an active internet connection and the Firebase Realtime Database is available.

4.1.15 Use Case 15: Edit Comment	
Created by	Adeel Asghar
Date Created	4/12/2020
Last Updated by	Adeel Asghar
Date of last revision	4/12/2020
Actors	Application User
Description	This use case will describe a user editing one of their comments.
Trigger	The user wants to edit their comment, so they type in a edited comment and click the save button located in the Edit Comment activity screen.
Preconditions	The user must log in and must be the author of the comment.
Post conditions	The edited comment will show on the user profile and in the comments of whatever post it's in.
Normal Flow	1. The user navigates to profile screen using the navigation drawer. 2. The user clicks the comments fragment. 3. The user swipes left on post they want to edit. 4. The user edits the title and/or text. 5. The user selects the save button. 6. The user is navigated back to the profile screen and the edited post is displayed.
Alternative Flows	Alt Flow 1. The user edits a comment text to be empty 1. The user navigates to profile screen using the navigation drawer. 2. The user clicks the comments fragment. 3. The user swipes left on post they want to edit. 4. The user edits the text to be blank. 5. The user selects the save button. 6. The user is prompted with an error message let them know the text is blank and returns to UC-15.

Frequency of Use	This will be used every time the user wants to edit one of their posts.
Assumptions	There is an active internet connection and the Firebase Realtime Database is available.

4.1.16 Use Case 16: Hamburger Menu	
Created by	Palak Patel
Date Created	4/12/2020
Last Updated by	Palak Patel
Date of last revision	4/12/2020
Actors	Application User
Description	This use case describes the user opening the hamburger menu.
Trigger	The user selects the button in the left corner.
Preconditions	The user must log in.
Post conditions	The hamburger menu will be open.
Normal Flow	1. The user navigates to a screen where the hamburger menu button is visible. 2. The user selects the button. 3. The menu opens.
Alternative Flows	N/A
Frequency of Use	This will be used every time the user accesses their user profile or settings.
Assumptions	N/A

4.1.17 Use Case 17: Switch Themes (Night Mode)	
Created by	Adeel Asghar
Date Created	4/12/2020
Last Updated by	Adeel Asghar
Date of last revision	4/12/2020
Actors	Application User
Description	This use case will describe a user changing the theme of the application.
Trigger	The user wants to change the theme of their application to either light mode, night mode, or system theme which follows the theme of phone.
Preconditions	The user must log in.
Post conditions	The chosen theme will display app wide and will remain selected even if the app is exited.
Normal Flow	1. The user clicks the hamburger menu. 2. The user clicks Settings. 3. The user clicks whatever mode they wish to specify: Light Theme, Night Theme, or System Theme. 4. The user then selects the Set Theme button.
Alternative Flows	N/A
Frequency of Use	This will be used anytime the user wants to change the app theme.
Assumptions	N/A

4.1.18 Use Case 18: Messaging – New Conversation	
Created by	Adeel Asghar
Date Created	4/12/2020
Last Updated by	Adeel Asghar
Date of last revision	4/12/2020
Actors	Application User
Description	This use case will describe a user starting a new conversation with a user.
Trigger	The user wants to send a message, so they navigate to the Messaging fragment and send a message.
Preconditions	The user must log in.
Post conditions	The chat log will update with the latest messages and the Messaging fragment will display the most recent message underneath the new user's messaging row.
Normal Flow	1. The user navigates to Messaging fragment. 2. The user clicks the New Message button on the title bar. 3. The user selects the specified user. 4. The user types a message. 5. The user clicks send.
Alternative Flows	N/A
Frequency of Use	This will be used every time the user wants to start a new conversation with a user.
Assumptions	There is an active internet connection and the Firebase Realtime Database is available.

4.1.18 Use Case 19: Messaging – Existing Conversation	
Created by	Adeel Asghar
Date Created	4/12/2020
Last Updated by	Adeel Asghar
Date of last revision	4/12/2020
Actors	Application User
Description	This use case will describe a user sending a message to an existing conversation.
Trigger	The user wants to send a message, so they navigate to the Messaging fragment and send a message.
Preconditions	The user must log in and must have already messaged the specified user.
Post conditions	The chat log will update with the latest messages and the Messaging fragment will display the most recent message underneath the specific user's messaging row.
Normal Flow	1. The user navigates to Messaging fragment. 2. The user clicks the row of the specified user with whom there is already previous conversation. 3. The user types a message. 1. The user clicks send.
Alternative Flows	1. The user navigates to Messaging fragment. 2. The user clicks the New Message button on the title bar. 3. The user selects the specified user. 4. The user types a message. The user clicks send.
Frequency of Use	This will be used every time the user wants to send a message.
Assumptions	There is an active internet connection and the Firebase Realtime Database is available.

4.1.20 Use Case 20: Report Post	
Created by	Adeel Asghar
Date Created	4/12/2020
Last Updated by	Adeel Asghar
Date of last revision	4/12/2020
Actors	Application User
Description	This use case will describe a user reporting a post.
Trigger	The user wants to report a post, so they select the report confirmation button.
Preconditions	The user must log in.
Post conditions	An email will be sent to unit.school.team.help@gmail.com with information about the post.
Normal Flow	1. The user navigates to a forum of their choice. 2. The user swipes left on the post that they wish to report. 3. The user selects the report post button. 4. The user is prompted to select a reason for reporting the post. 5. The user selects the confirm button. 6. The user is informed that report has been sent.
Alternative Flows	Alt Flow 1. A user tries to report themselves. 1. The user navigates to a forum of their choice. 2. The user cannot swipe on their own posts.
Frequency of Use	This will be used every time the user wants to report a comment or post, the user can report a post multiple times.
Assumptions	The user has an active internet connection and the Firebase real time database is available.

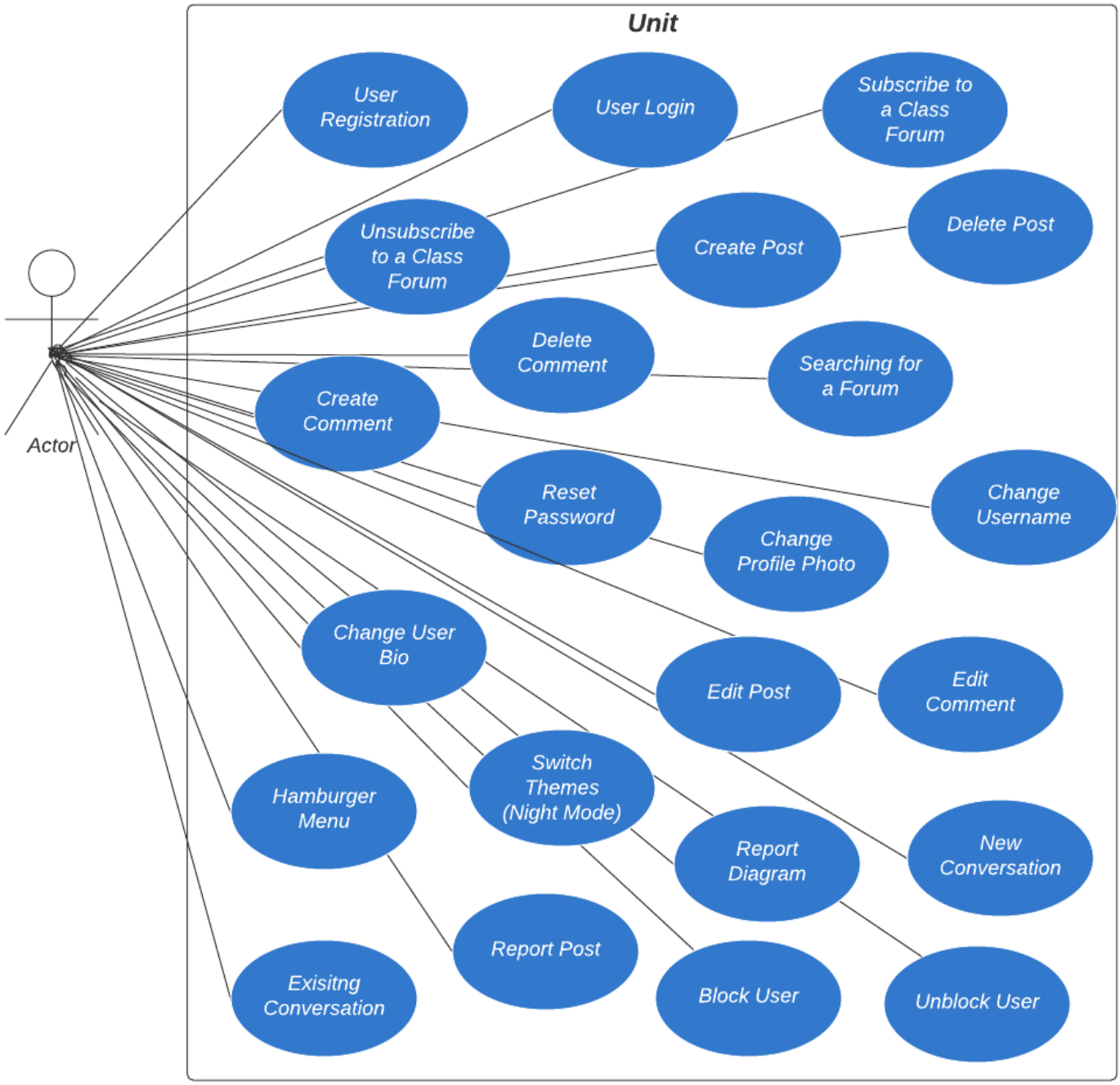
4.1.21 Use Case 21: Report Comment	
Created by	Palak Patel
Date Created	4/12/2020
Last Updated by	Palak Patel
Date of last revision	4/12/2020
Actors	Application User
Description	This use case will describe a user reporting a comment
Trigger	The user wants to report a comment, so they select the report confirm button.
Preconditions	The user must log in.
Post conditions	An email will be sent to unit.school.team.help@gmail.com with information about the comment.
Normal Flow	1. The user navigates to a forum of their choice. 2. The user swipes left on the post that they wish to report. 3. The user selects the report post button. 4. The user is prompted to select a reason for reporting the post. 5. The user selects the confirm button. 6. The user is informed that report was sent.
Alternative Flows	Alt Flow 1. The User tries to report a comment that they make 1. The user navigates to a forum of their choice. 2. The user clicks on a comment. 3. The user cannot swipe on a comment they made.
Frequency of Use	This will be used every time the user wants to report a comment. users can report the same comment multiple times.
Assumptions	There is an active internet connection and the Firebase Realtime Database is available.

4.1.21 Use Case 22: Block User	
Created by	Adeel Asghar
Date Created	4/12/2020
Last Updated by	Palak Patel
Date of last revision	4/12/2020
Actors	Application User
Description	This use case will describe a user blocking another user.
Trigger	The user wants to block a user so the select the confirm block button.
Preconditions	The user must log in.
Post conditions	The blocked user's posts are no longer are visible from anywhere in the application.
Normal Flow	1. The user navigates to a forum of their choice. 2. The user swipes left on the post whose user they wish to block. 3. The user selects the block.
Alternative Flows	Alt Flow 1. The user tries to block themselves 4. The user navigates to a forum of their choice. 5. The user clicks on a comment. 6. The user cannot swipe on a comment they made. Alt Flow 2. User blocks another user from a comment 1. The user navigates to a forum of their choice. 2. The user selects a post. 3. The user swipes left on a comment created by a user they wish to block. 4. The user selects the block user button. 5. The user confirms that they wish to block the user. 6. The user is navigated by the community forum.
Frequency of Use	This will be used every time the user wants to block another user.

Assumptions	There is an active internet connection and the Firebase Realtime Database is available.

4.1.21 Use Case 23: Unblock User	
Created by	Palak Patel
Date Created	4/12/2020
Last Updated by	Palak Patel
Date of last revision	4/13/2020
Actors	Application User
Description	This use case will describe a user blocking another user.
Trigger	The user wants to unblock a user so the select the confirm unblock button.
Preconditions	The user must log in and block at least one user.
Post conditions	The blocked user's posts are again visible in the application
Normal Flow	1. The user navigates to the settings screen. 2. The user swipes left on who they wish to unblock. 3. The user selects the unblock button. 4. The user confirms unblocking in the dialog box.
Alternative Flows	N/A
Frequency of Use	This will be used every time the user wants to unblock another user.
Assumptions	There is an active internet connection and the Firebase Realtime Database is available.

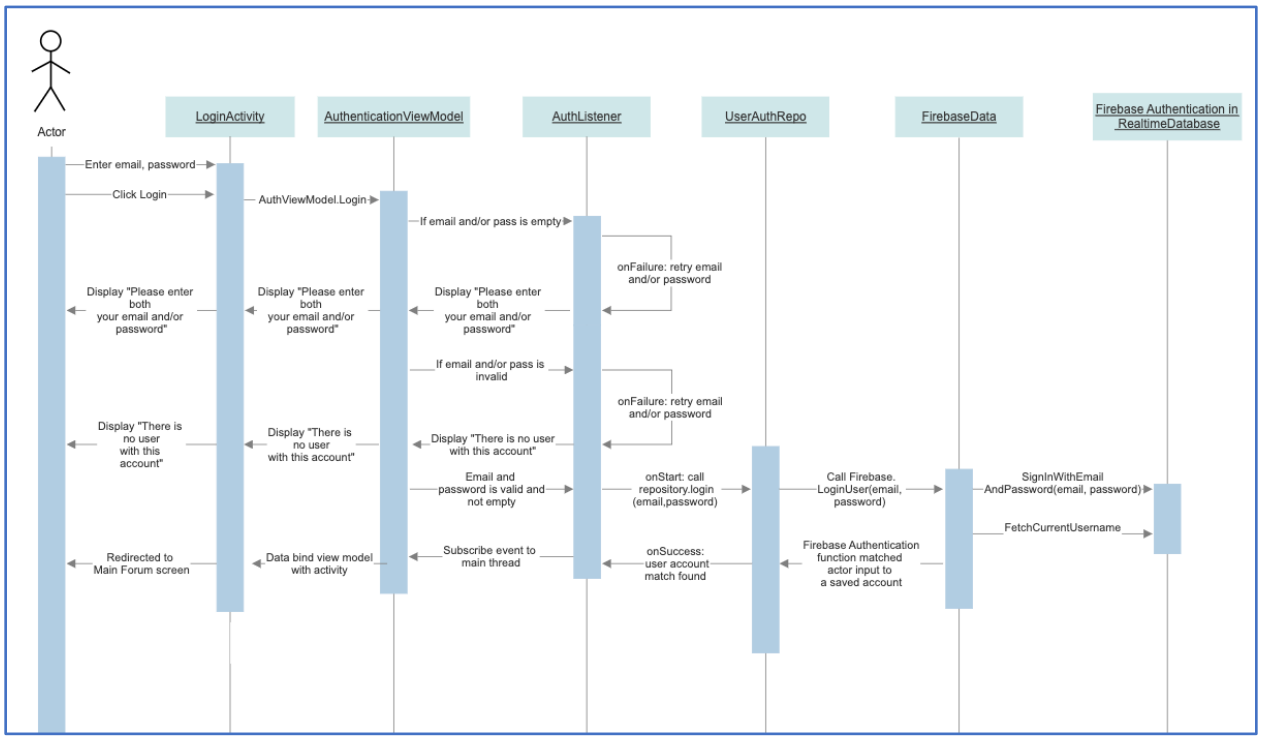
4.2 Use Case Diagram



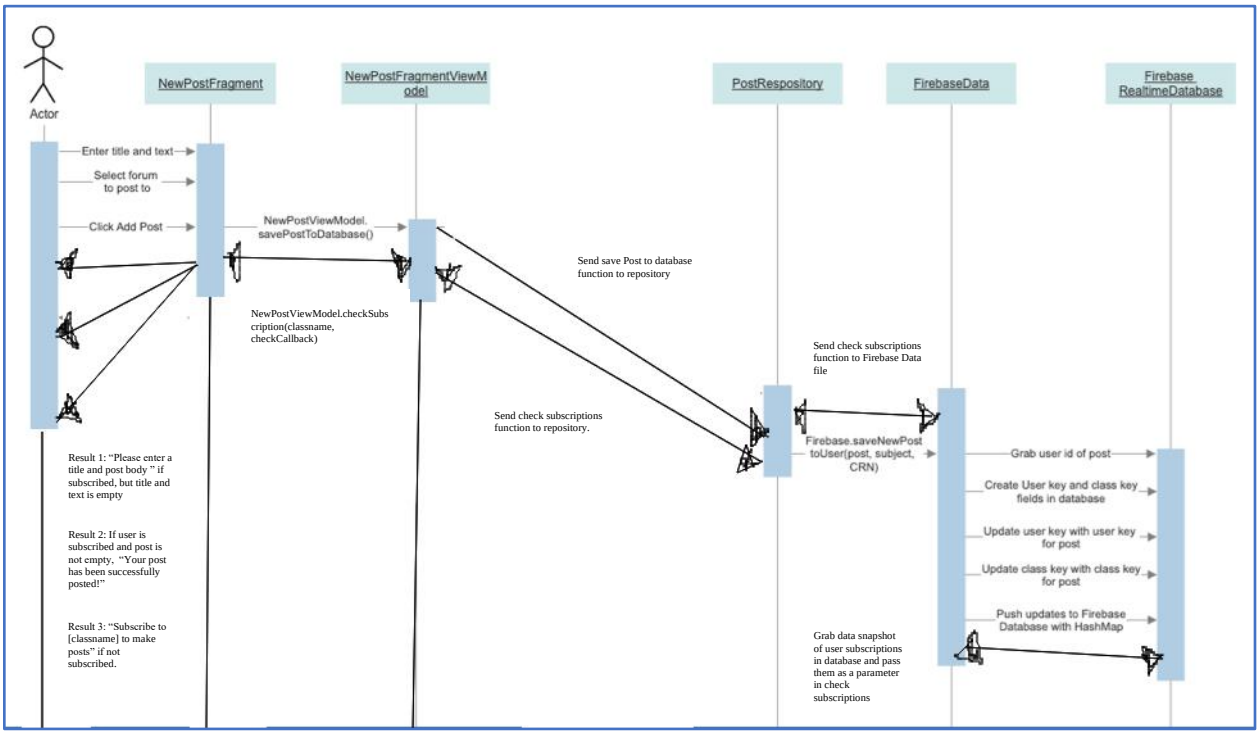
4.3 Sequence Diagram

The Sequence Diagram section of the Product Design Specification document will consist of three sequence diagrams that model significant functionalities found in the software product that are of high-level priority. These diagrams will model how the software product interacts with the user and updates itself accordingly from the following UI changes made. The three diagrams that are in this section include the “Login” sequence diagram, the “Add Post” sequence diagram, and the “Subscribe to Class Forum” sequence diagram. The second diagram involves callbacks which explains the double arrow on checking subscriptions.

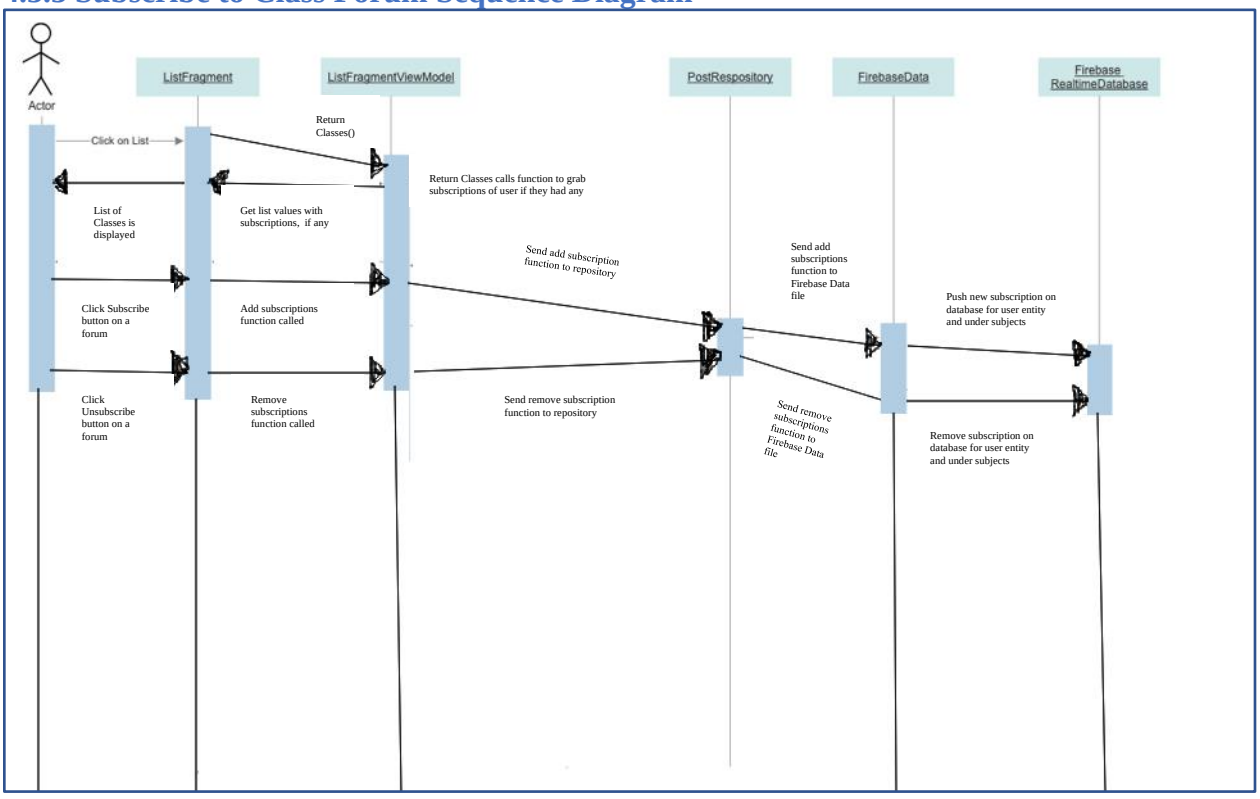
4.3.1 Login Sequence Diagram



4.3.2 Add Post Sequence Diagram



4.3.3 Subscribe to Class Forum Sequence Diagram

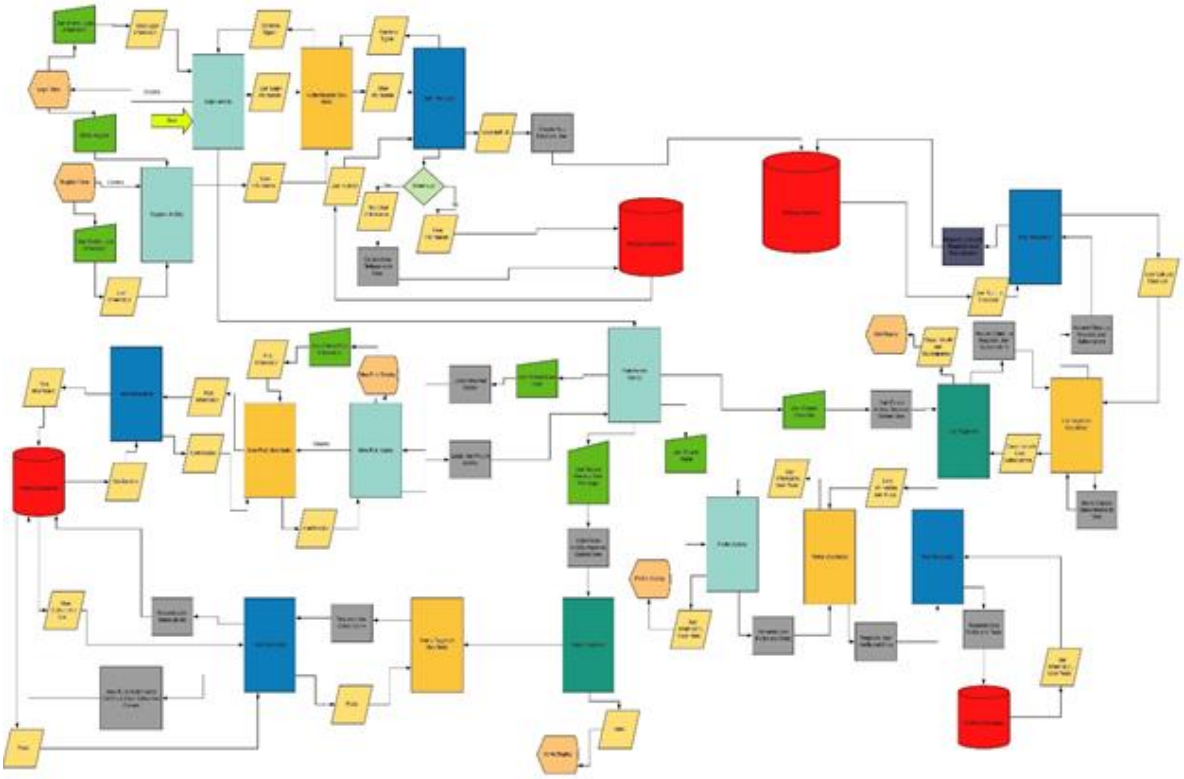


4.4 Data Flow Diagram

The Data Flow Diagram section will consist of 5 data flow diagrams that will explain each of the complex requirements of the software product: overarching application, user registration, user login, loading posts to the class forum, and creating new posts.

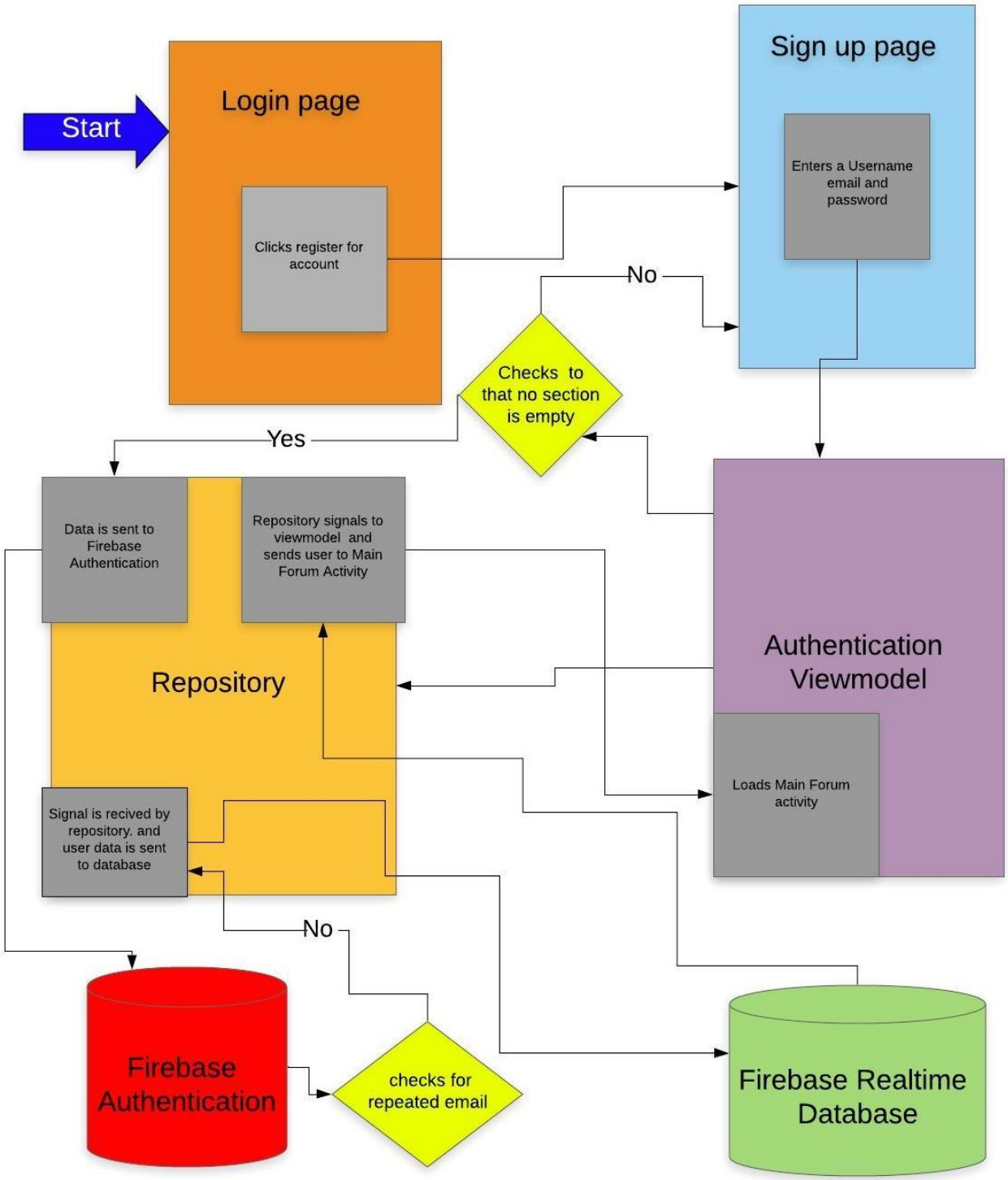
4.4.1 Application Data Flow Diagram (DFD)

The DFD shows how the system processes the user registration. The registration process is shown interacting with the registration activity, the authentication view model, the repository class, the Firebase Authentication, and the Firebase Realtime Database. OBJ start of the data flow begins with the login activity screen and shows how the data is updated into the Firebase Realtime Database.



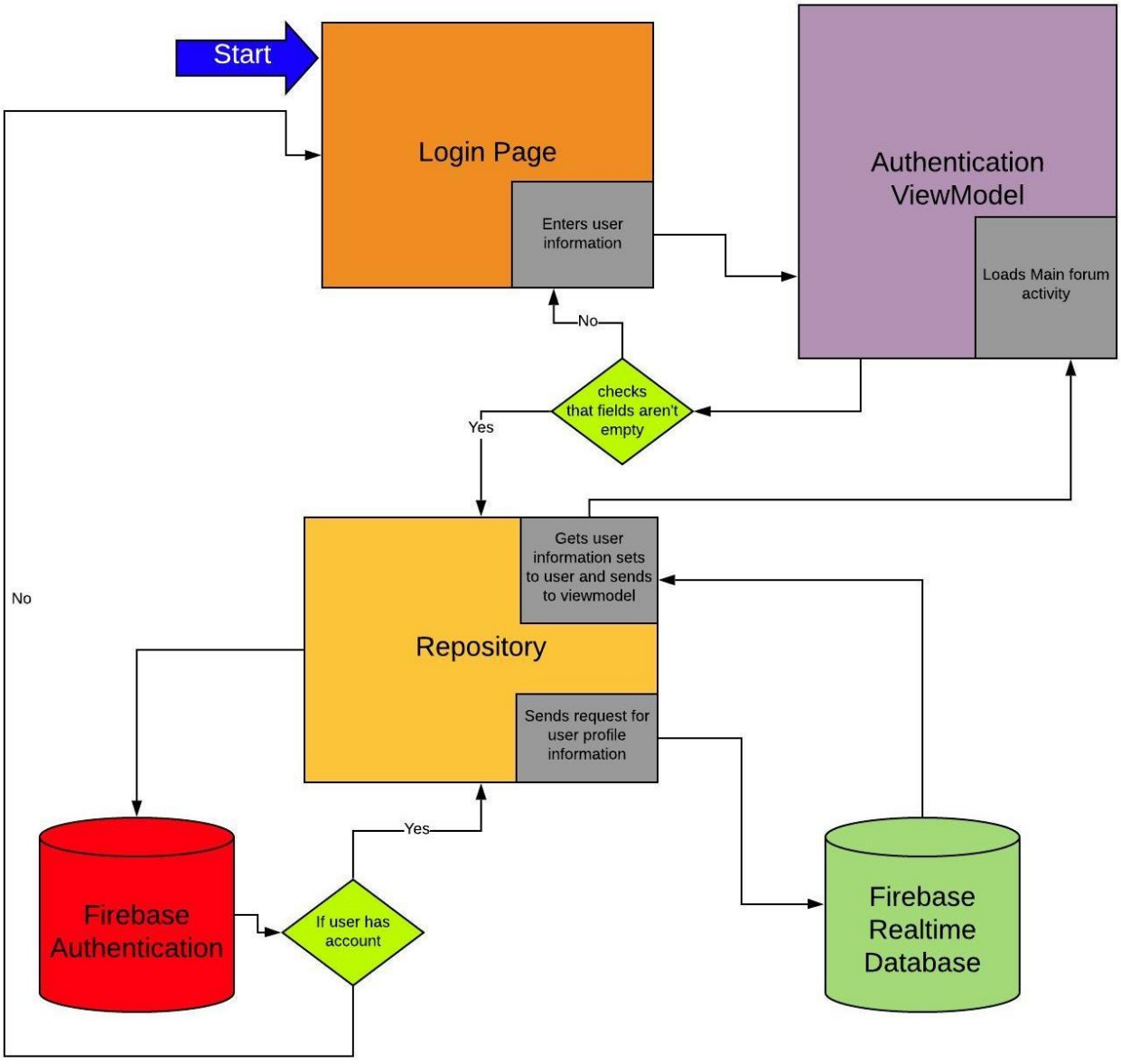
4.4.2 Data Flow Diagram (DFD)

The DFD shows how the system processes the user registration. The registration process is shown interacting with the registration activity, the authentication view model, the repository class, the Firebase Authentication, and the Firebase Realtime Database.



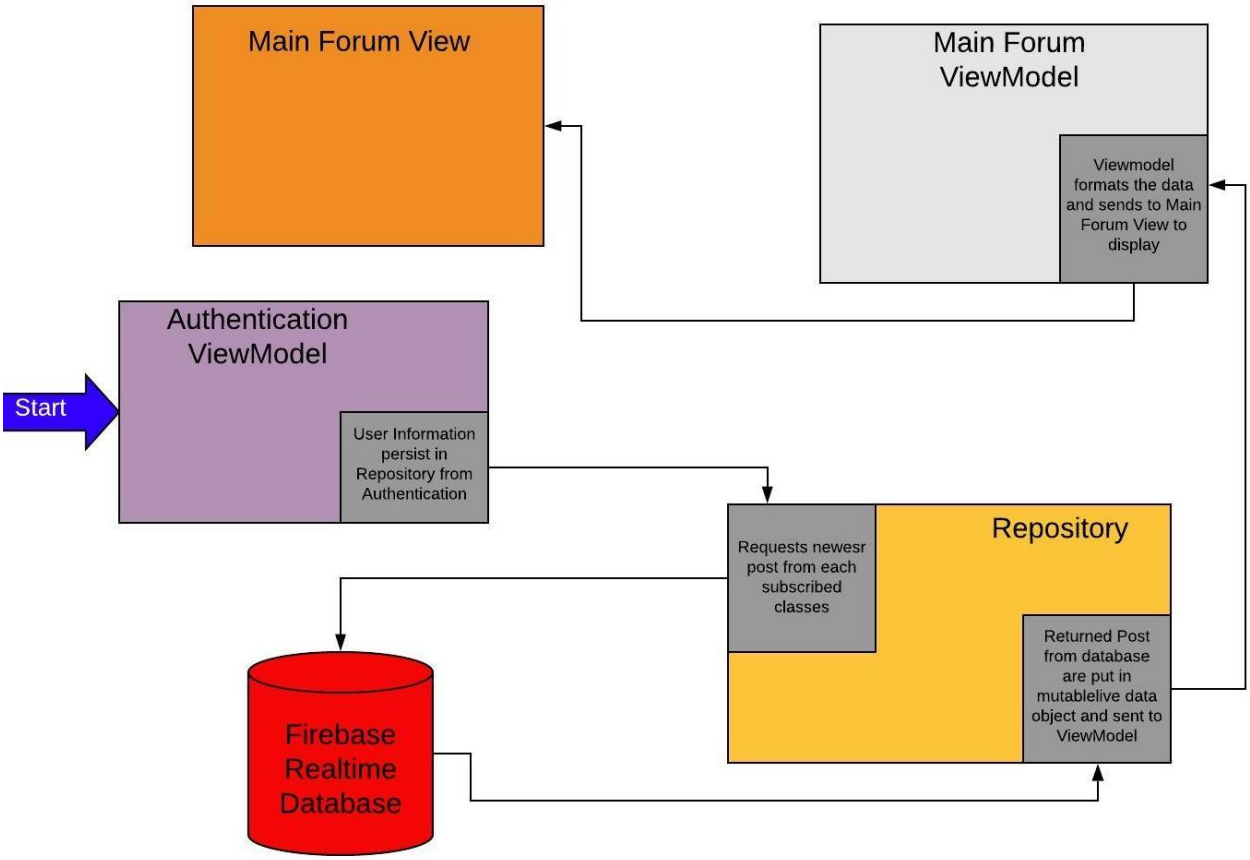
4.4.3 Log in Data Flow Diagram (DFD)

The DFD shows how the system processes the log in. The login process is shown interacting with the log in activity, the authentication view model, the repository class, the Firebase Authentication, and the Firebase Realtime Database.



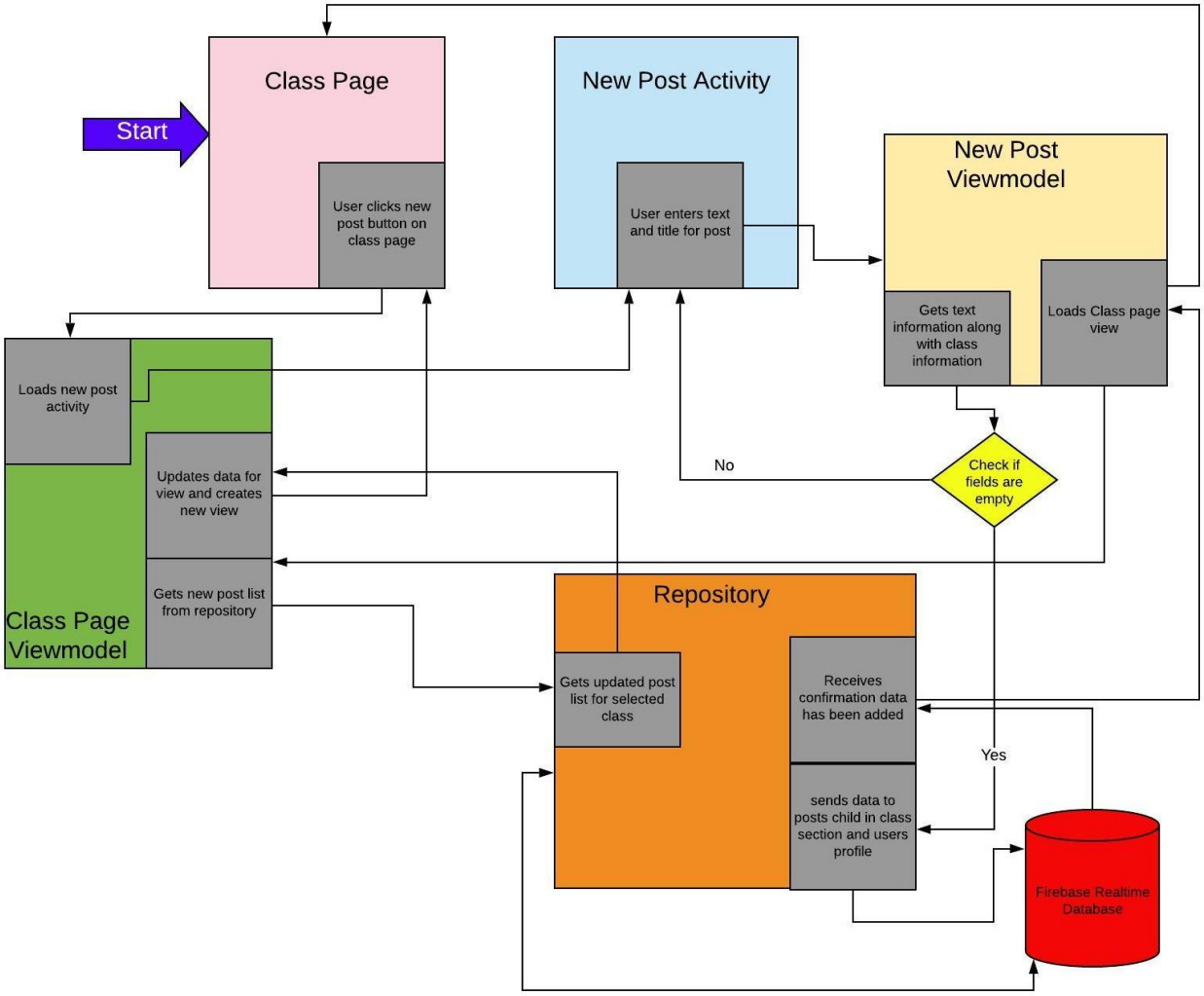
4.4.4 Load Posts Flow Diagram (DFD)

The DFD shows how the system processes loading posts for the user to see on the main forum. The load posts process is shown interacting with the new posts activity, the view model, the repository class, and the Firebase Realtime Database.

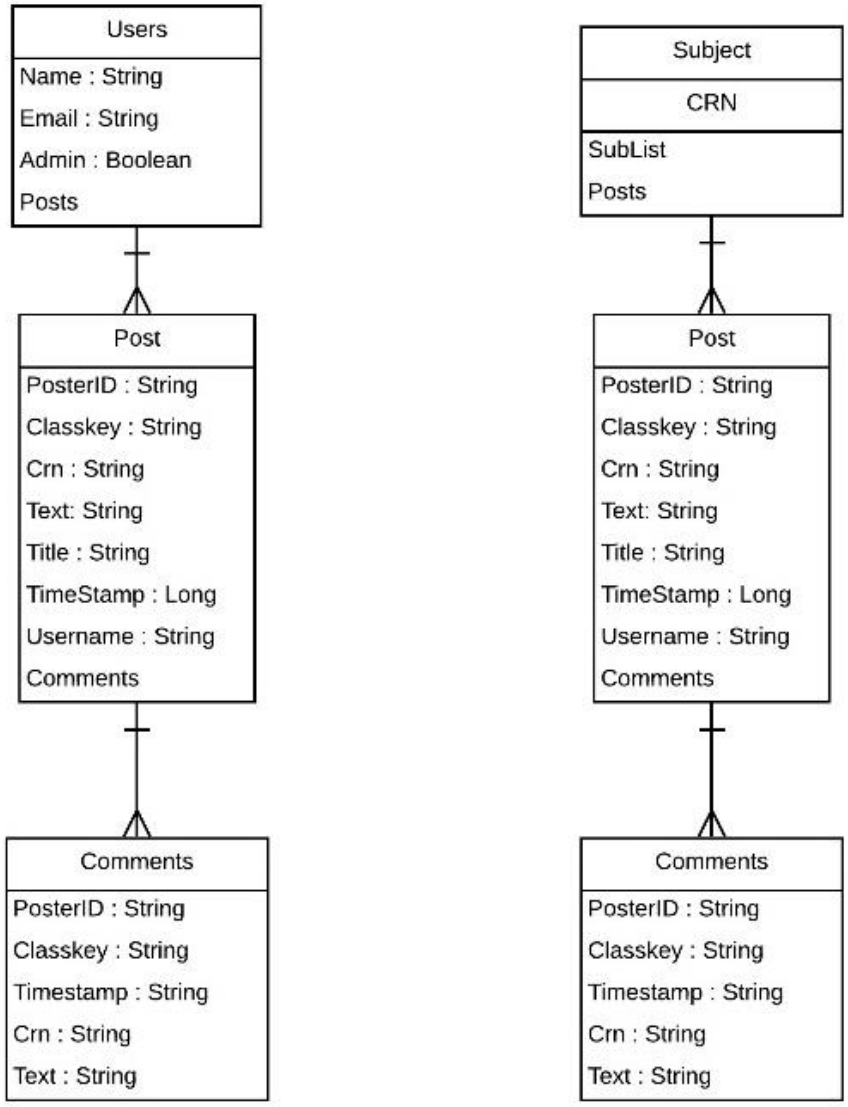


4.4.5 New Post Data Flow Diagram (DFD)

The DFD shows how the system processes the user creating a new post. The load posts process is shown interacting with the new posts activity, the view model, the repository class, and the Firebase Realtime Database.



4.5 Database Design

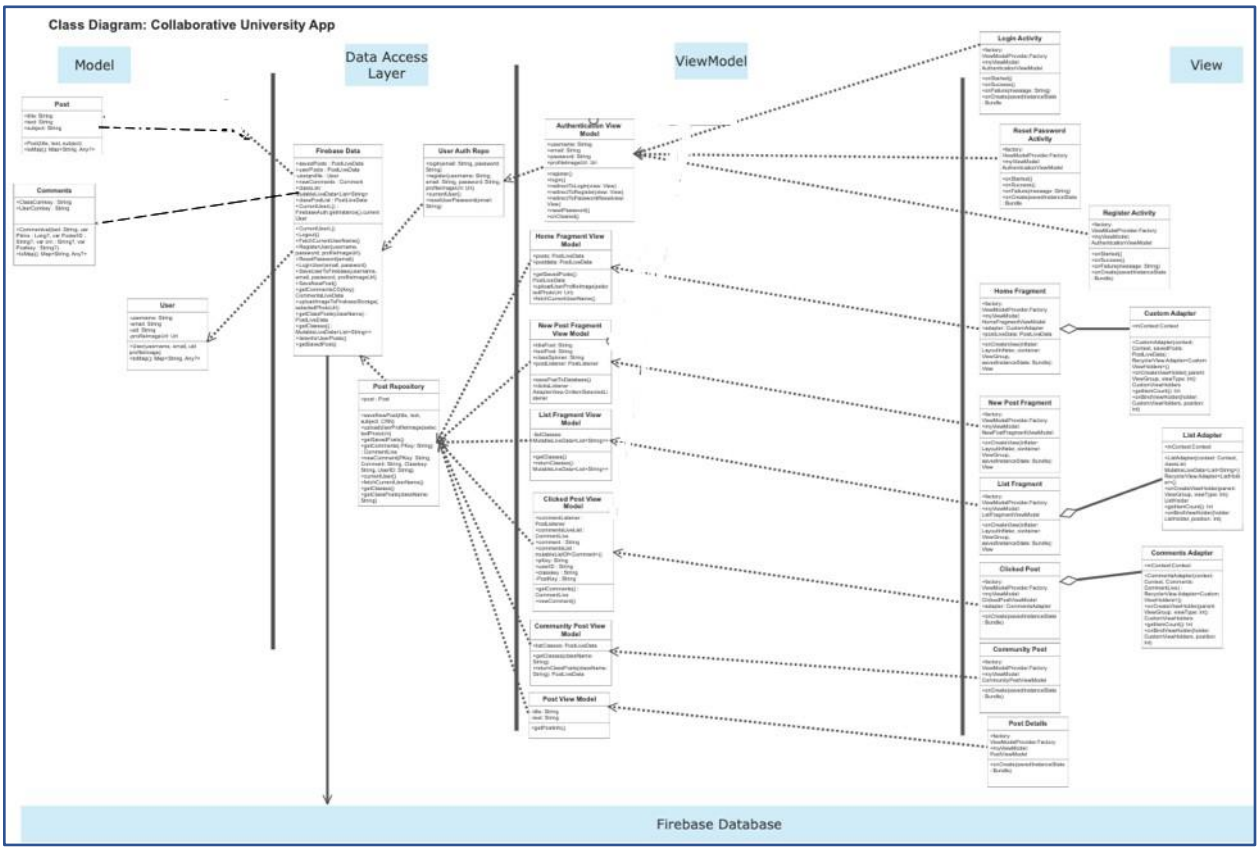


4.6 Class Diagram

The Class diagram includes two diagrams. The application class diagram covers the general structure of the entire system. Another one is a smaller class diagram that explains how the binding done with Dagger dependency injection framework works and how the view models interact with view model factory etc.

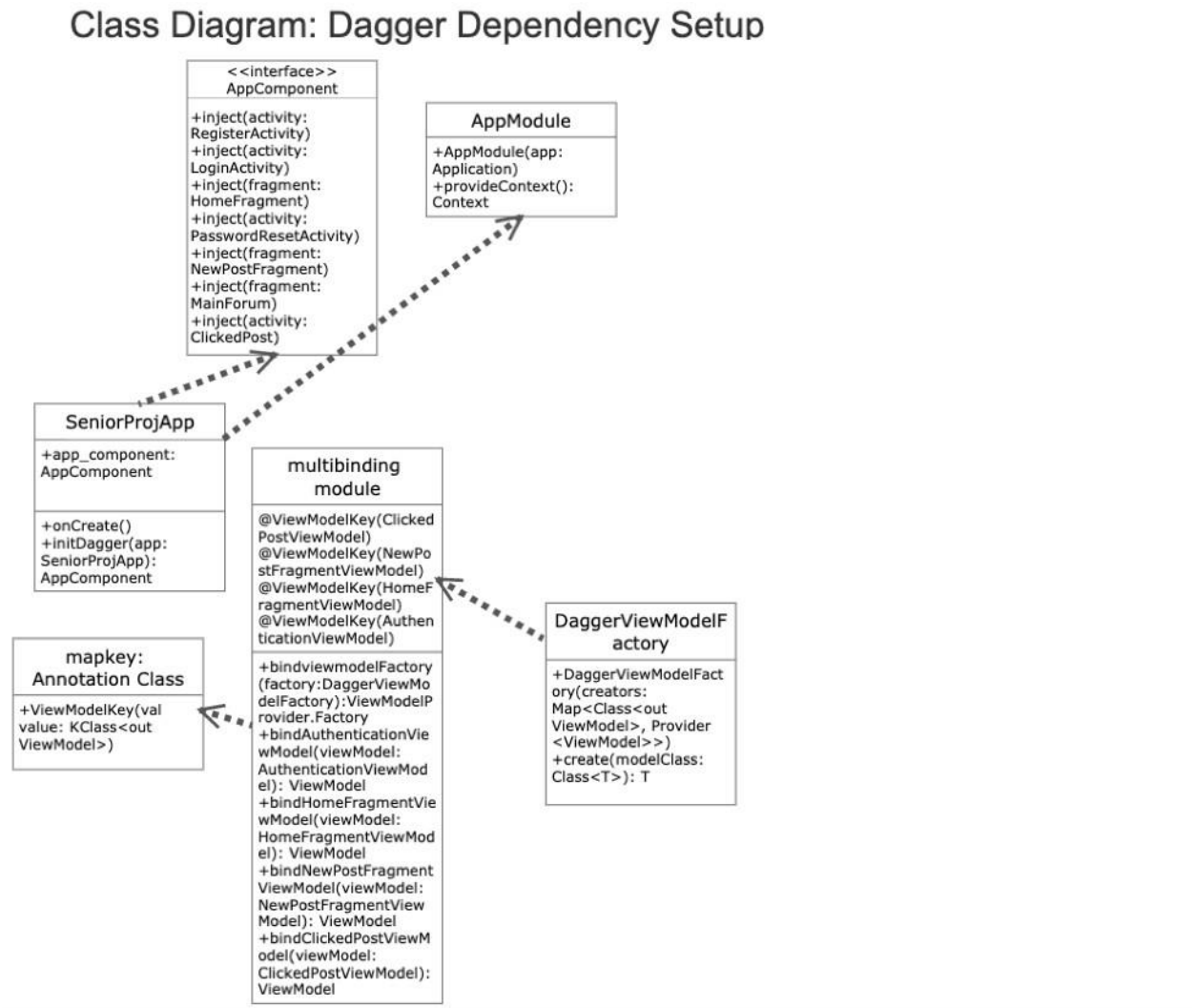
4.6.1 Application Class Diagram

The over-arching class diagram shows the entire system design of the software product using the MVVM design pattern. The major changes since then have been the creation of new models for chat messaging such as ChatMessage and LatestMessage as well as new models called Classes and Reports for a new reporting feature. New views/activities include a setting screen, viewing blocked users list, a user profile screen, and enhancements on existing functionality. Posts now can include images and a fragment specifically for image posts have been added. The view models for these activities have been created and connect with post repository. Data listeners are no longer needed and neither is live data since the shift of the application to coroutines using Kotlin Flow API for real-time updates.



4.6.2 Dagger Dependency Class Diagram

The dagger dependency class diagram shows how the dagger dependency framework is implemented in the app. All view models are now bound to a generic dagger view model factory with the help of the multi-binding module. Dagger is initialized in the Senior Project Application class with the help of Application Module and Application Component.



4.7 Application Program Interfaces

We are using the Firebase API which provides us with Authentication and Realtime Database. The Firebase Authentication includes email verification, password reset functionality, and email and password formatting. The Authentication Realtime Database allows us to store users' information, new posts, communities, and comments. These are then displayed in the application using text views and recycler views.

4.8 User Interface

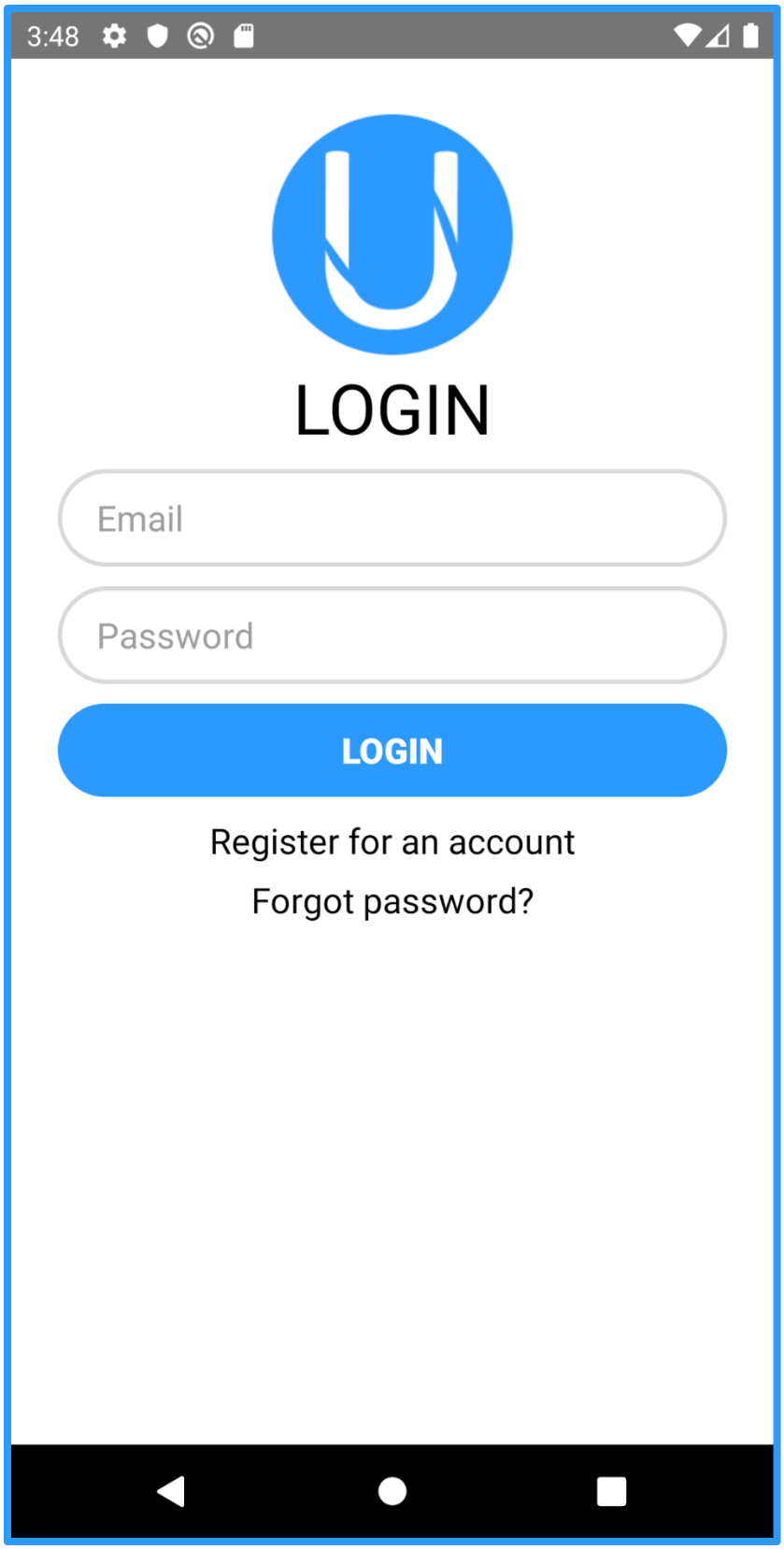


Figure 1: Login screen

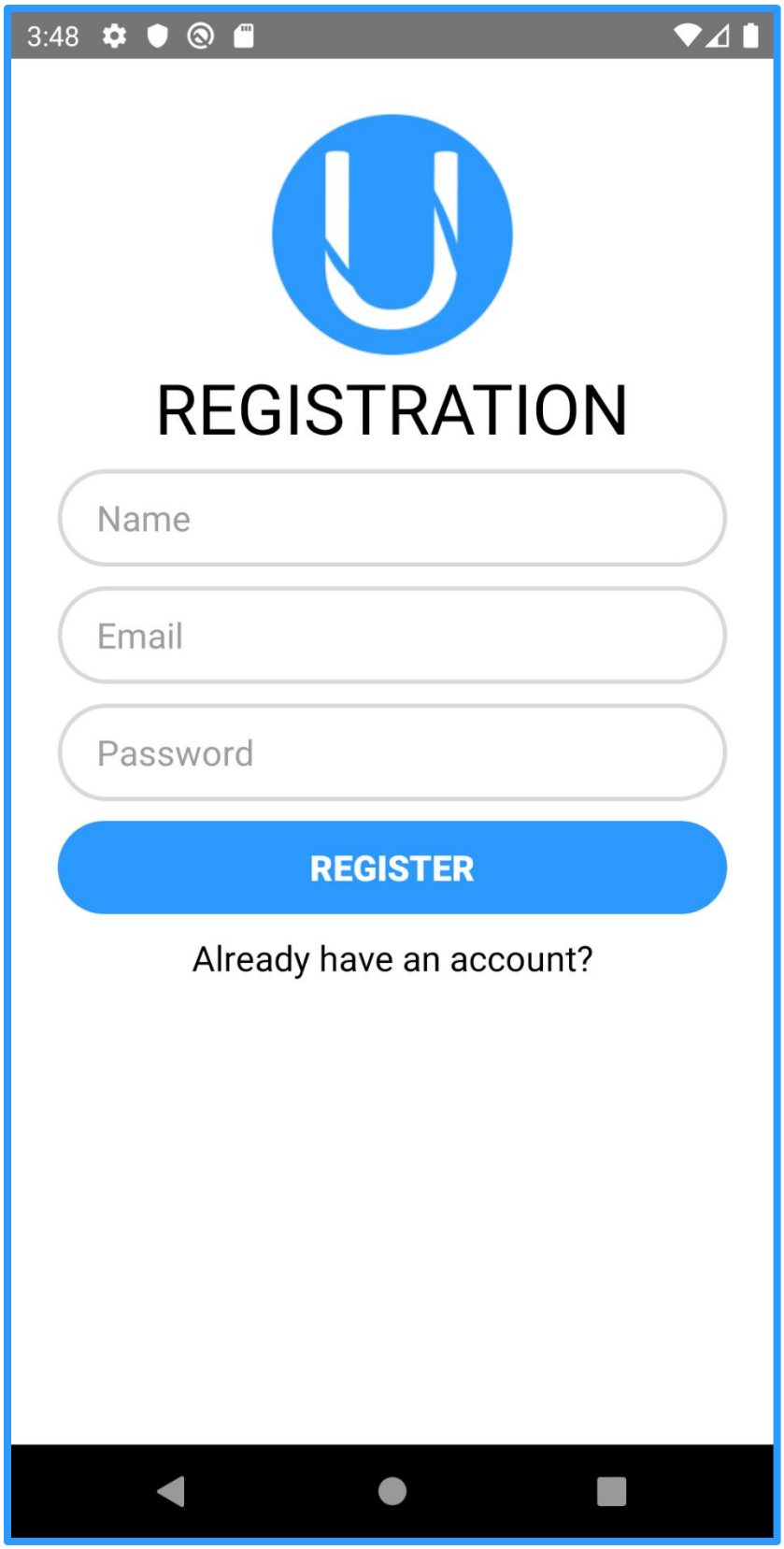


Figure 2: Registration screen

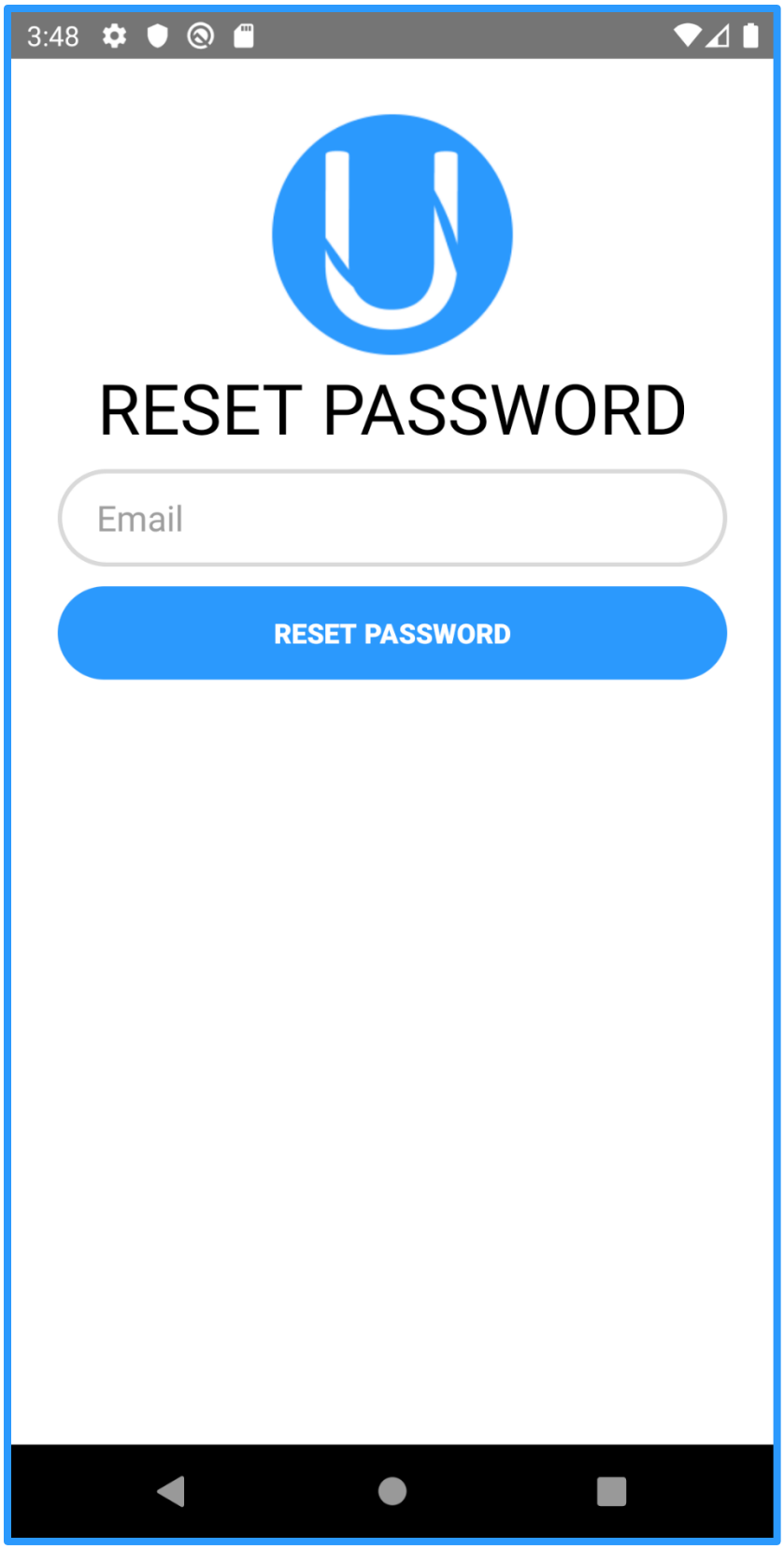


Figure 3: Password Reset Screen

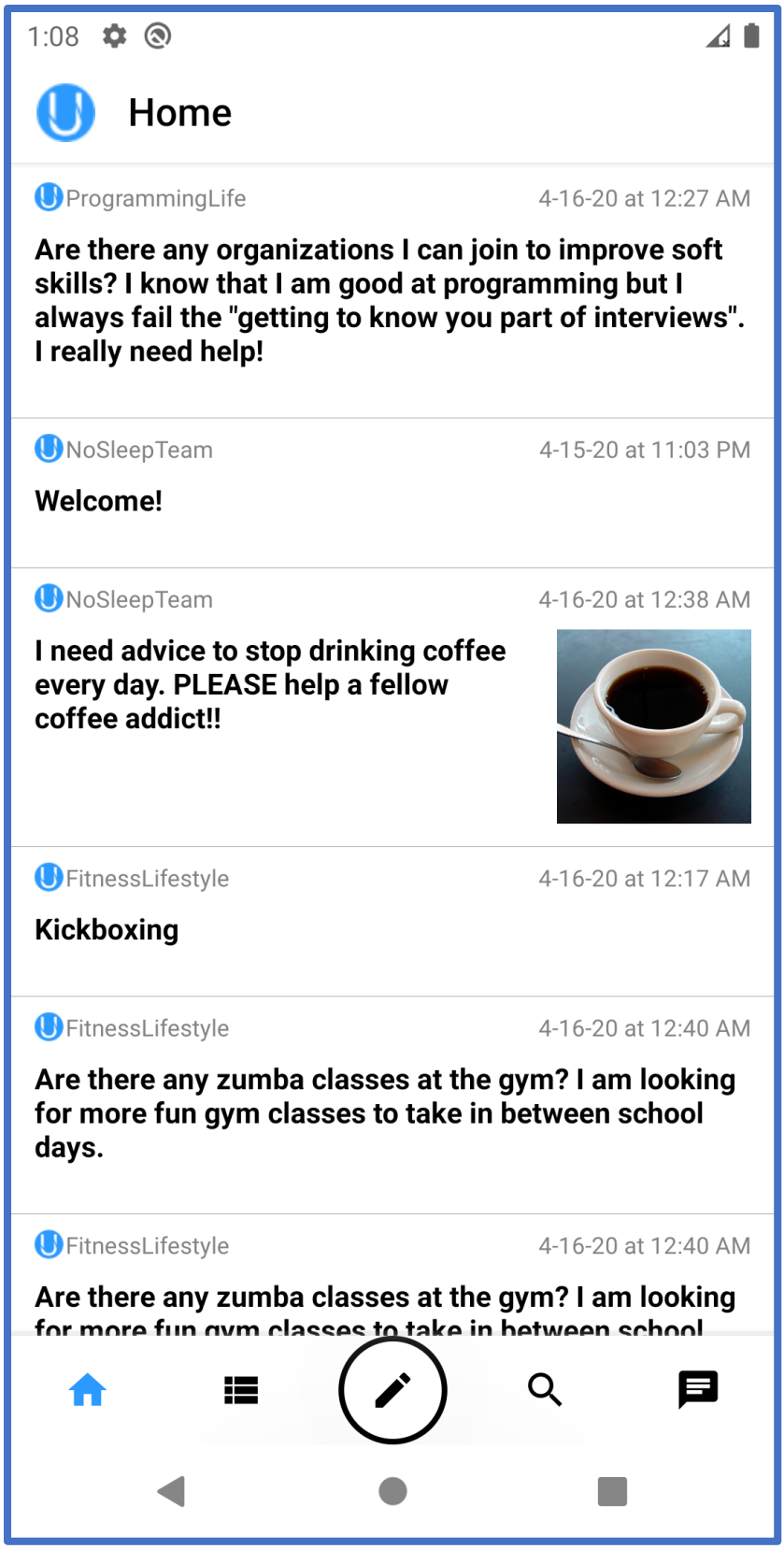


Figure 4: Home Page

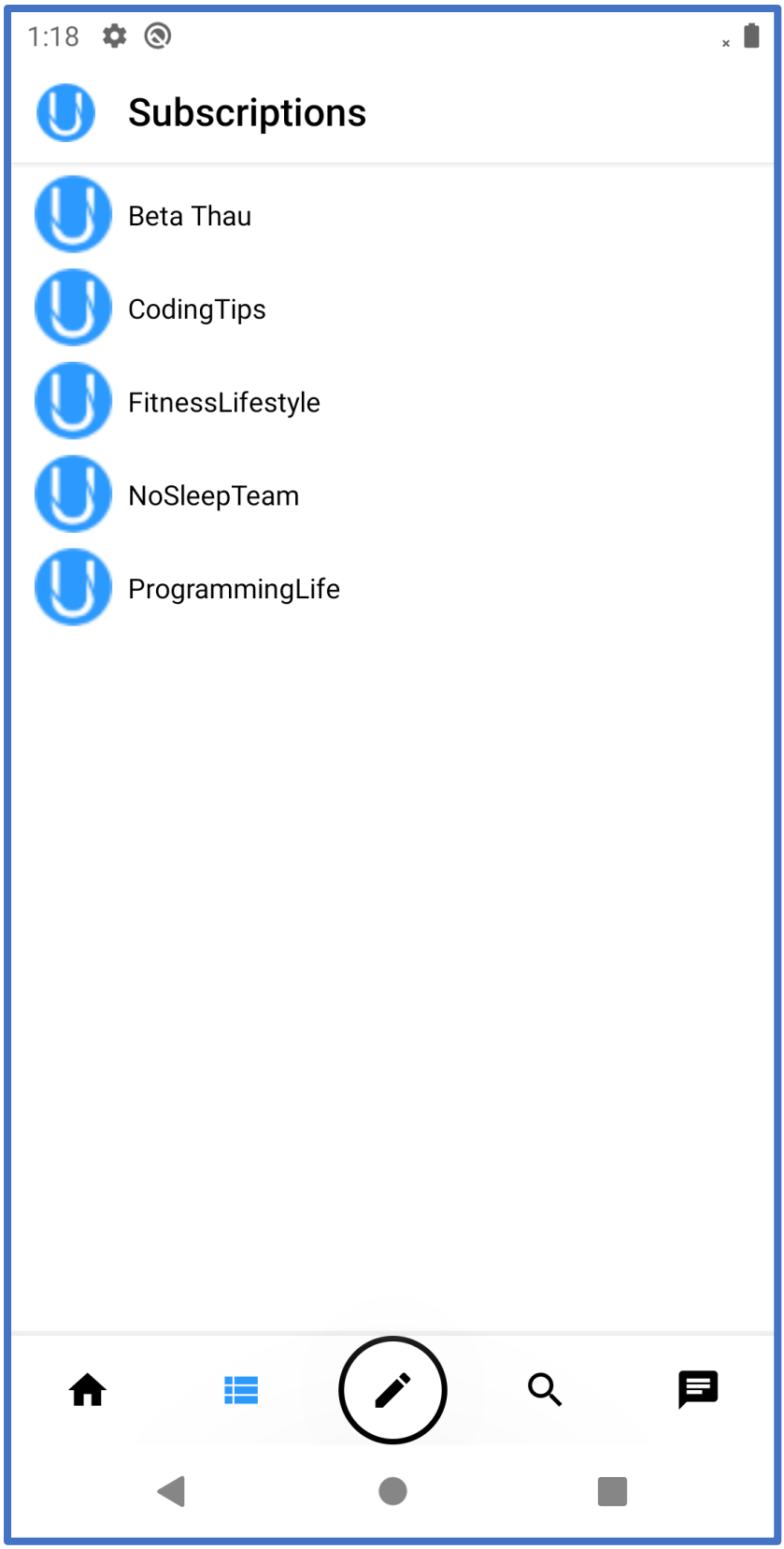


Figure 5: User's Subscriptions

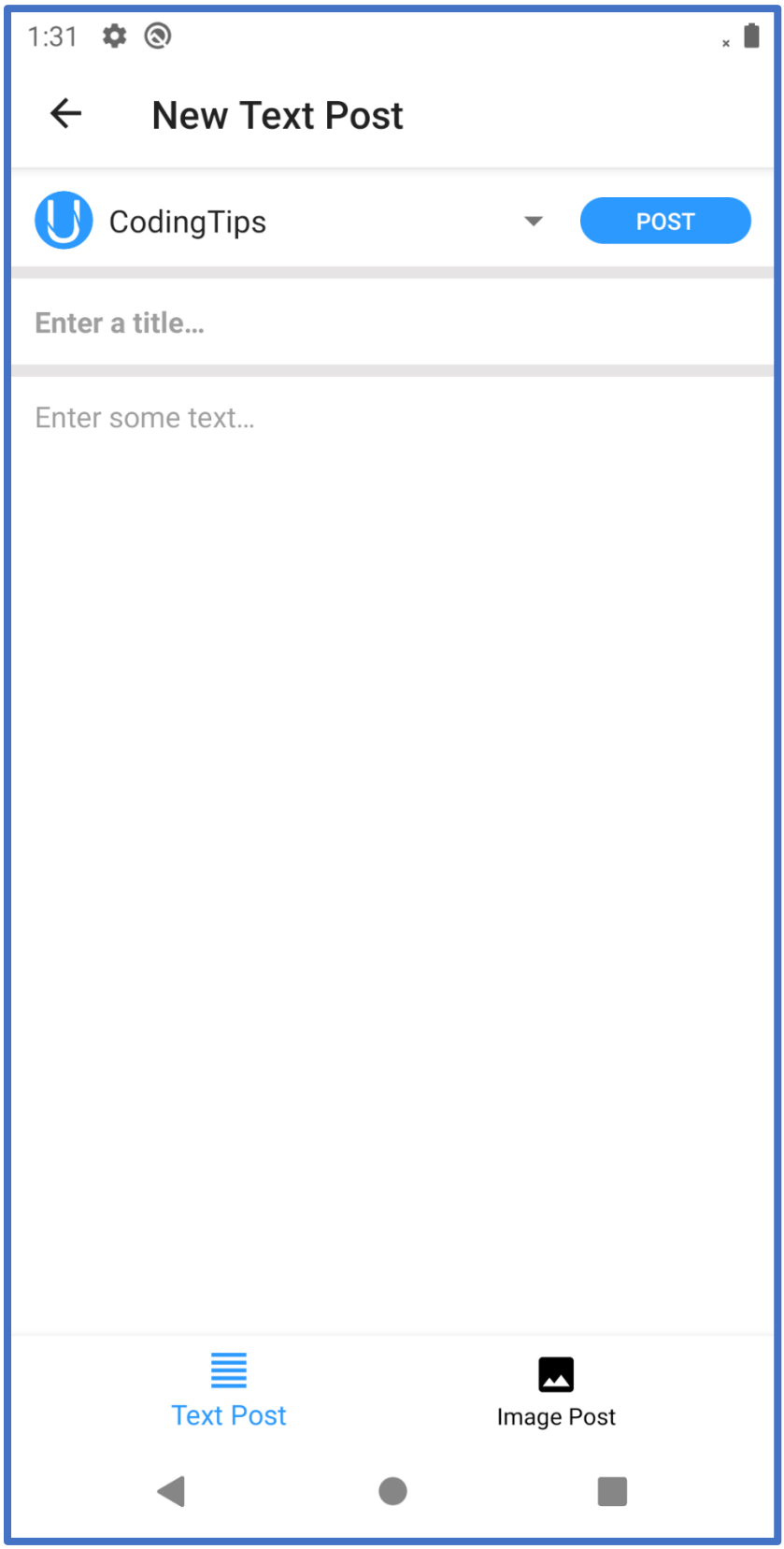


Figure 6: New Text Post

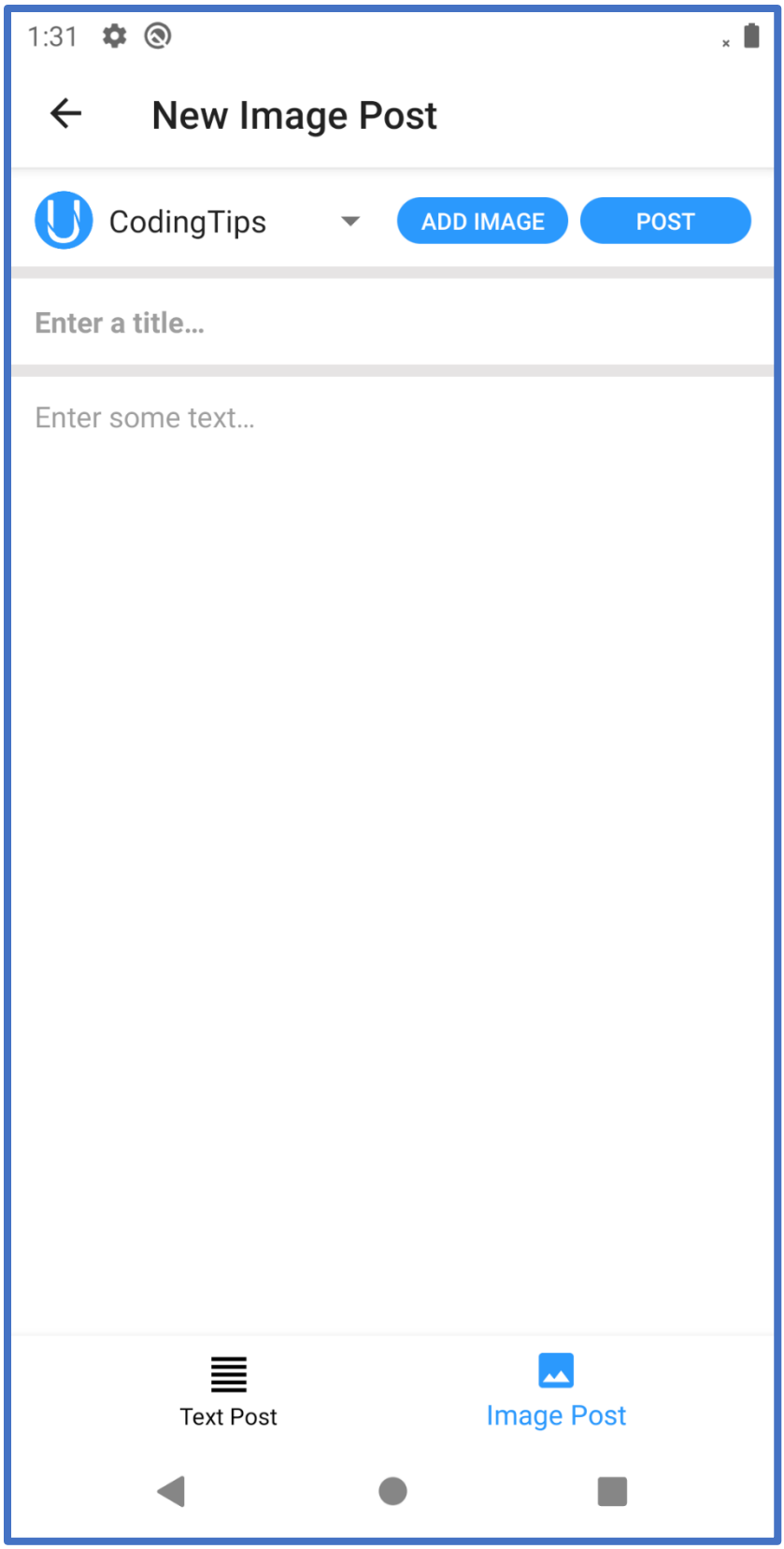


Figure 7: New Image Post

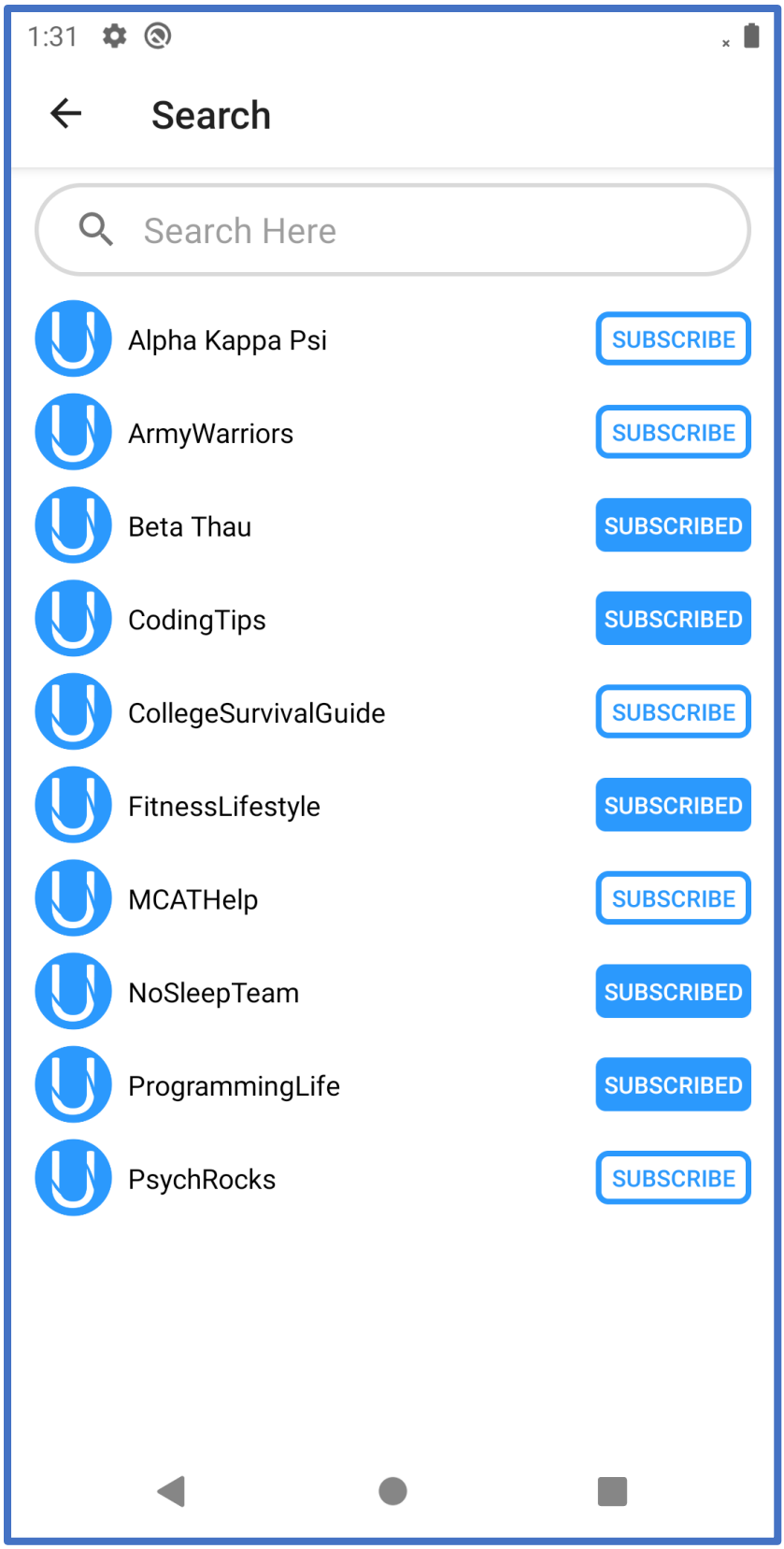


Figure 8: Class List

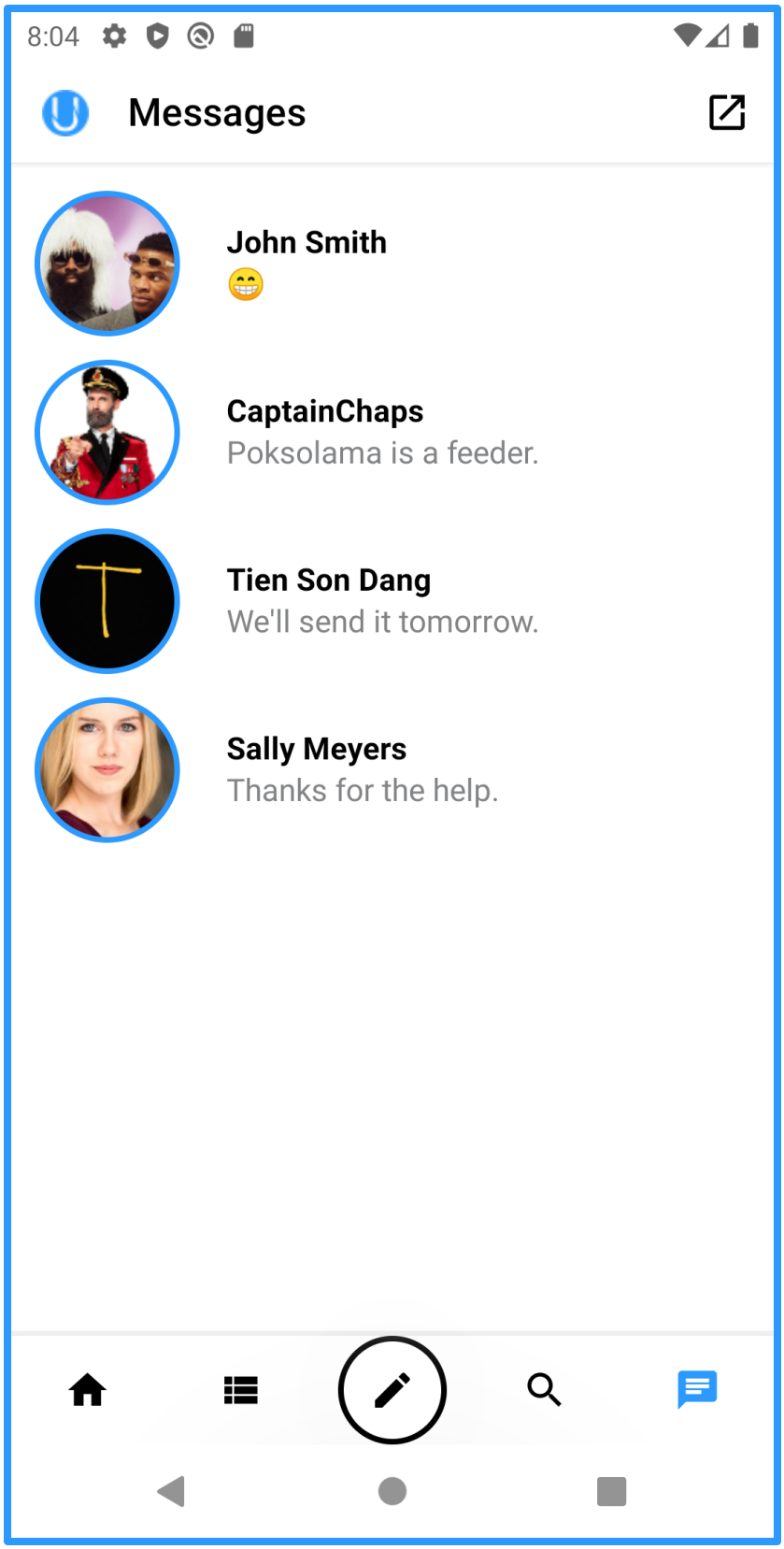


Figure 9: Recent Messages

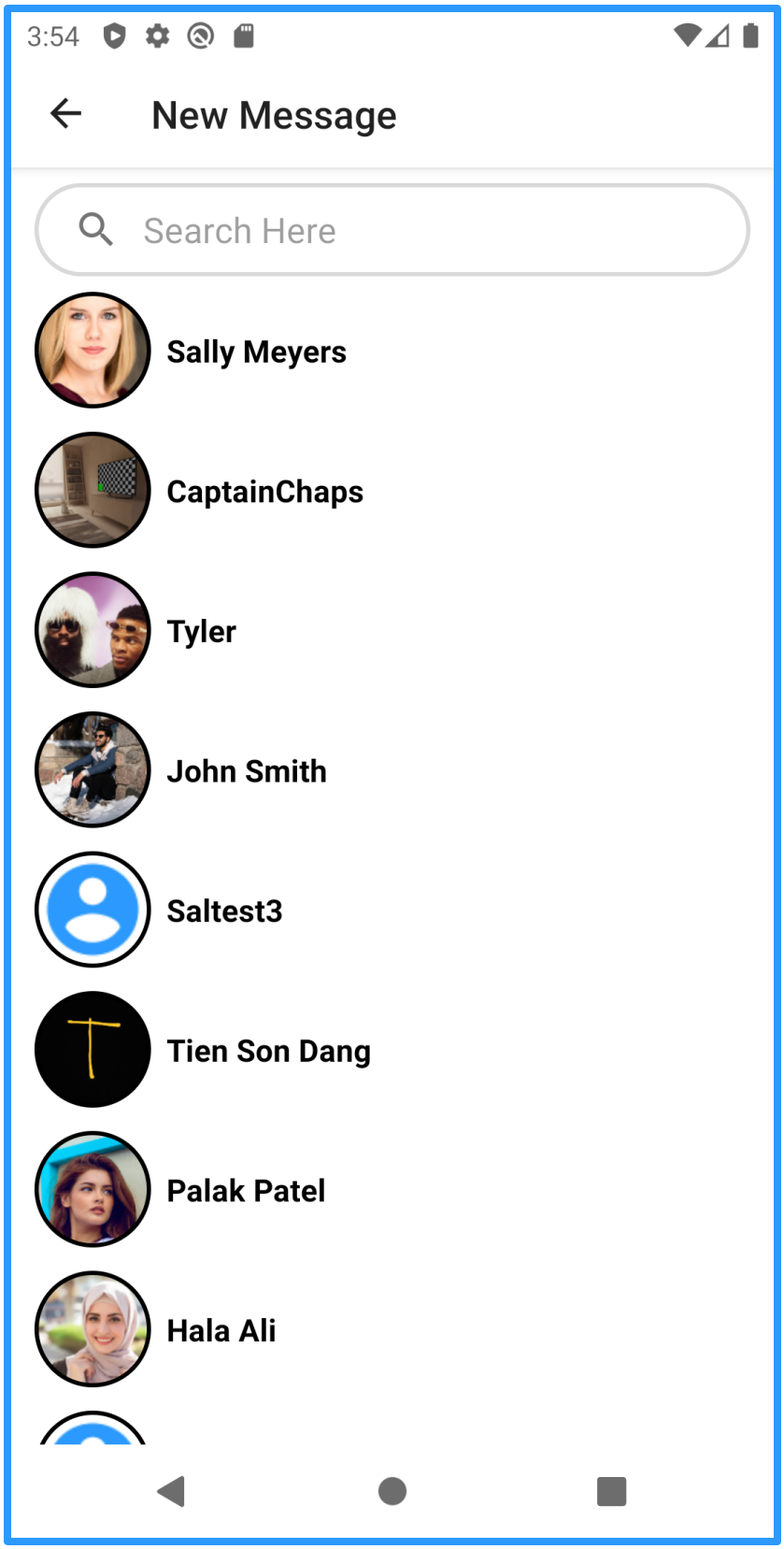


Figure 10: Send a New Message

Figure 11: Chatlog

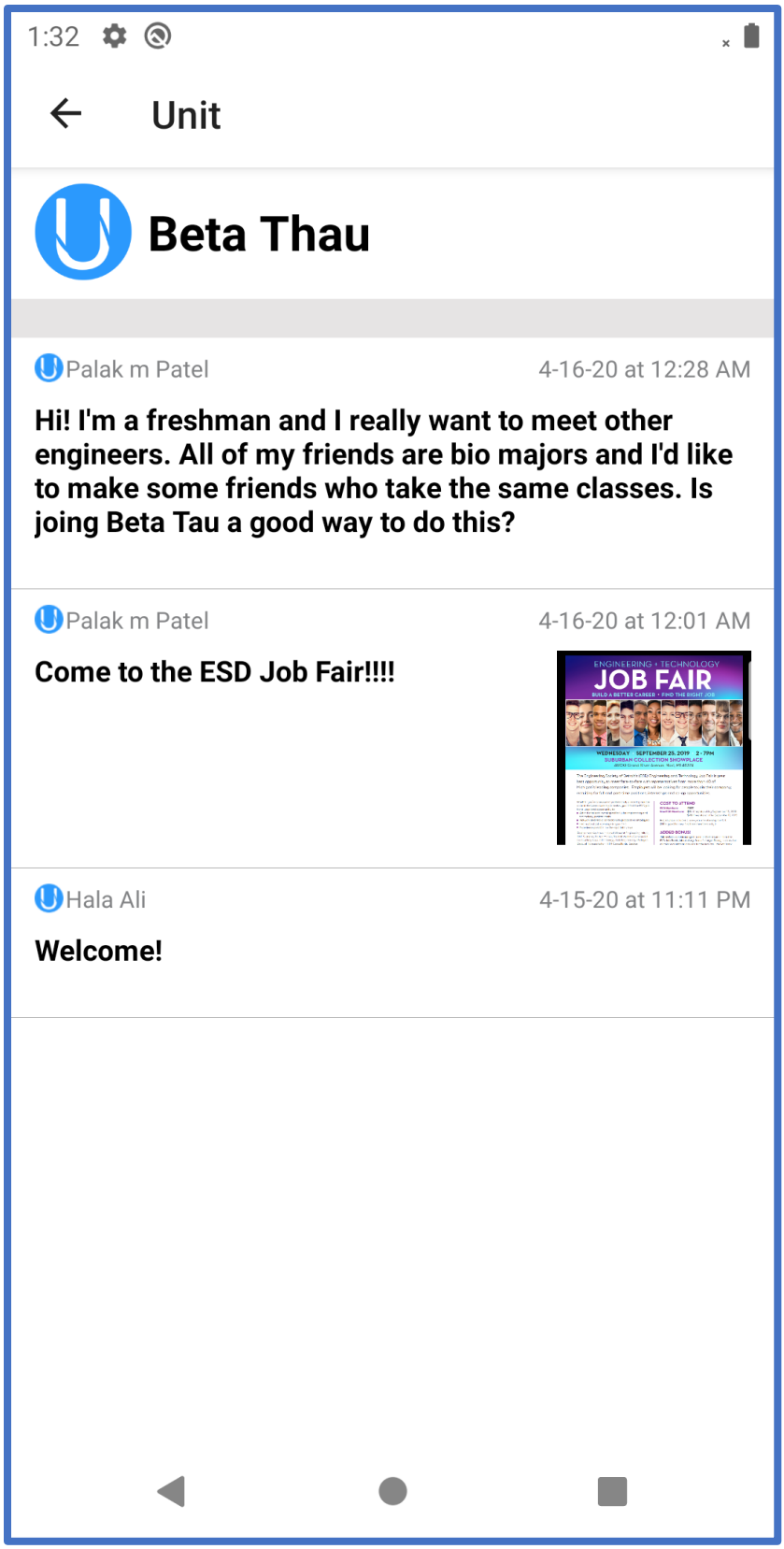


Figure 12: All posts for a class screen

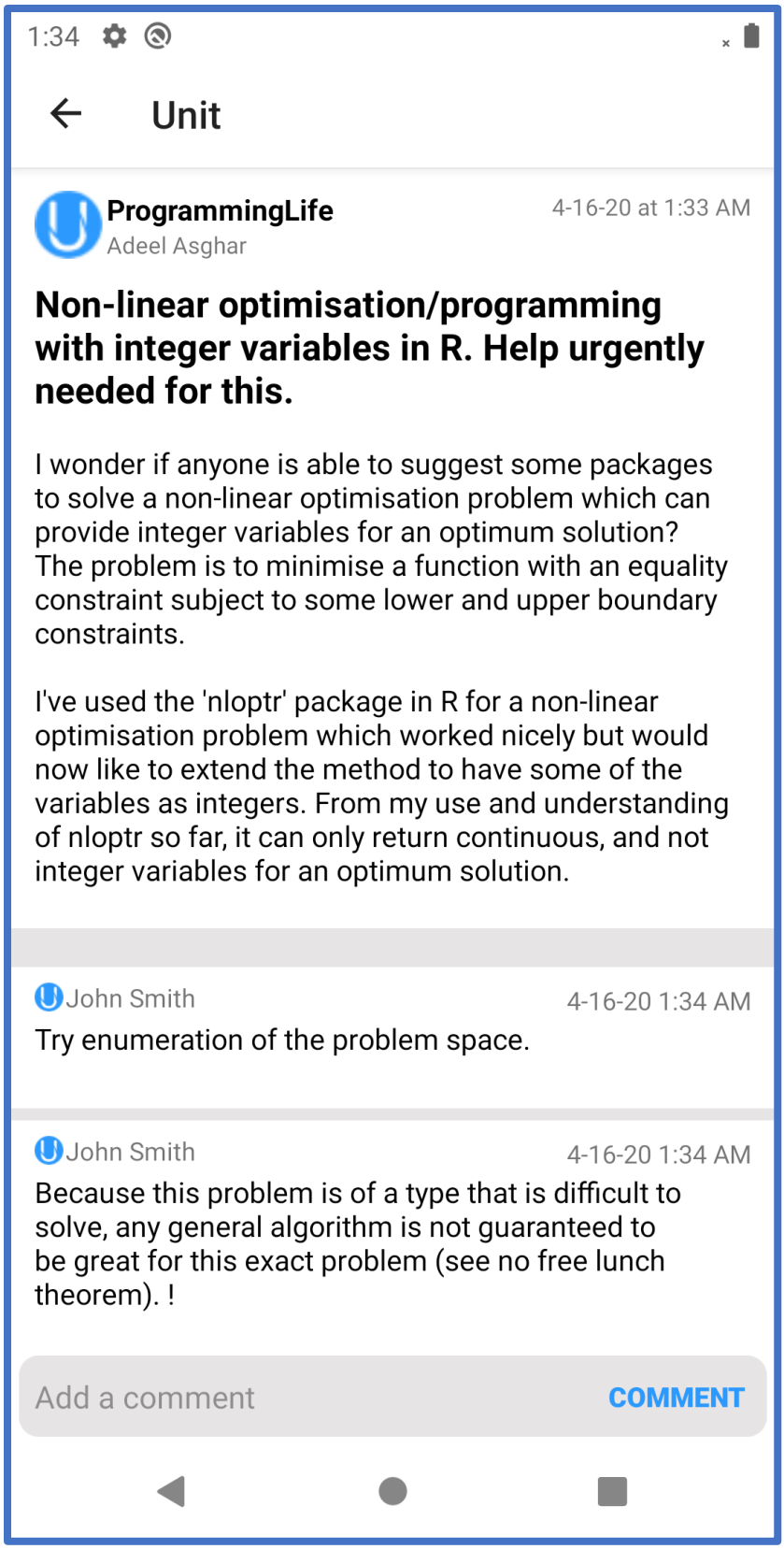


Figure 13: Clicked Post with Comments

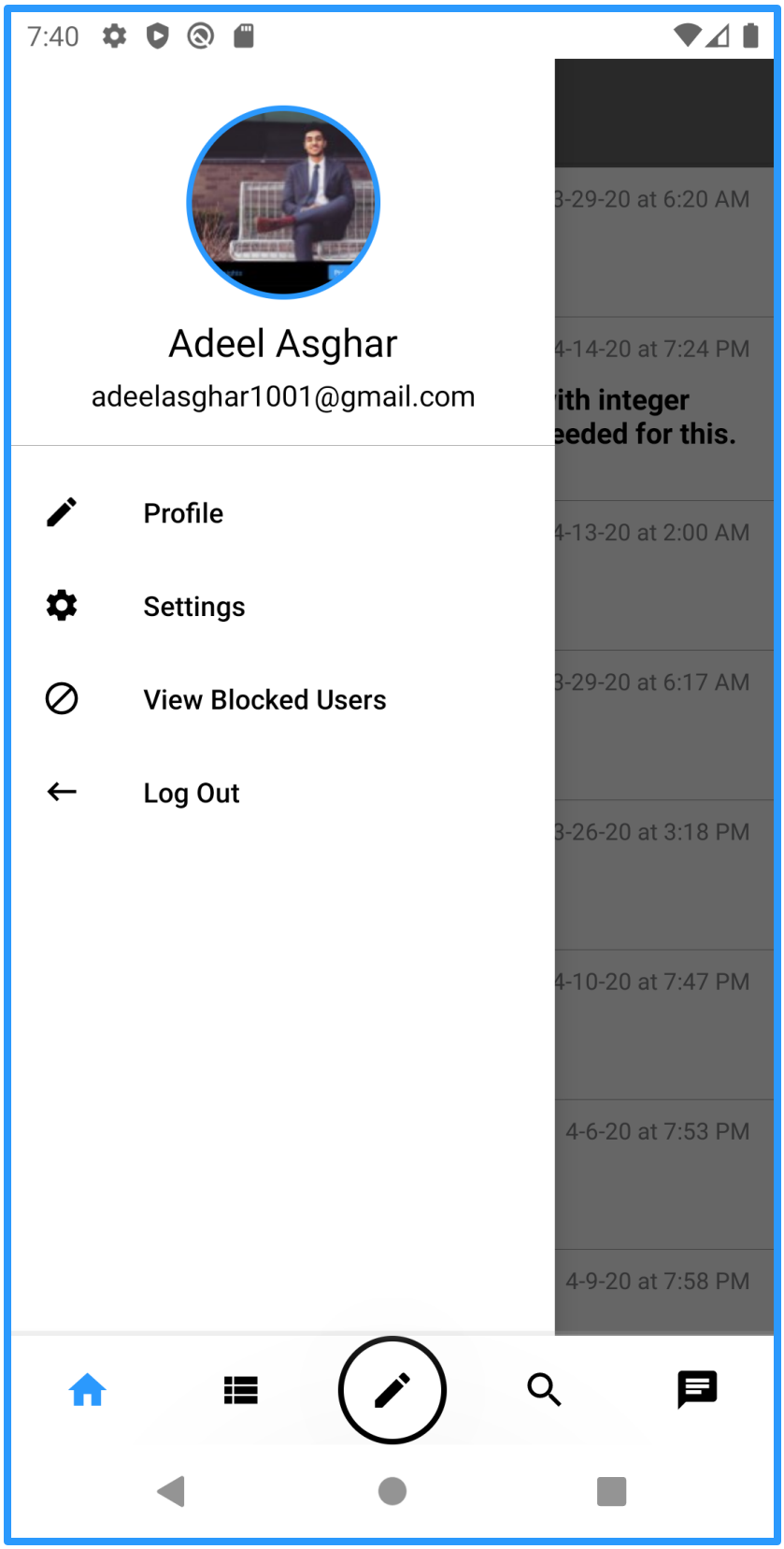


Figure 14: Navigation Drawer

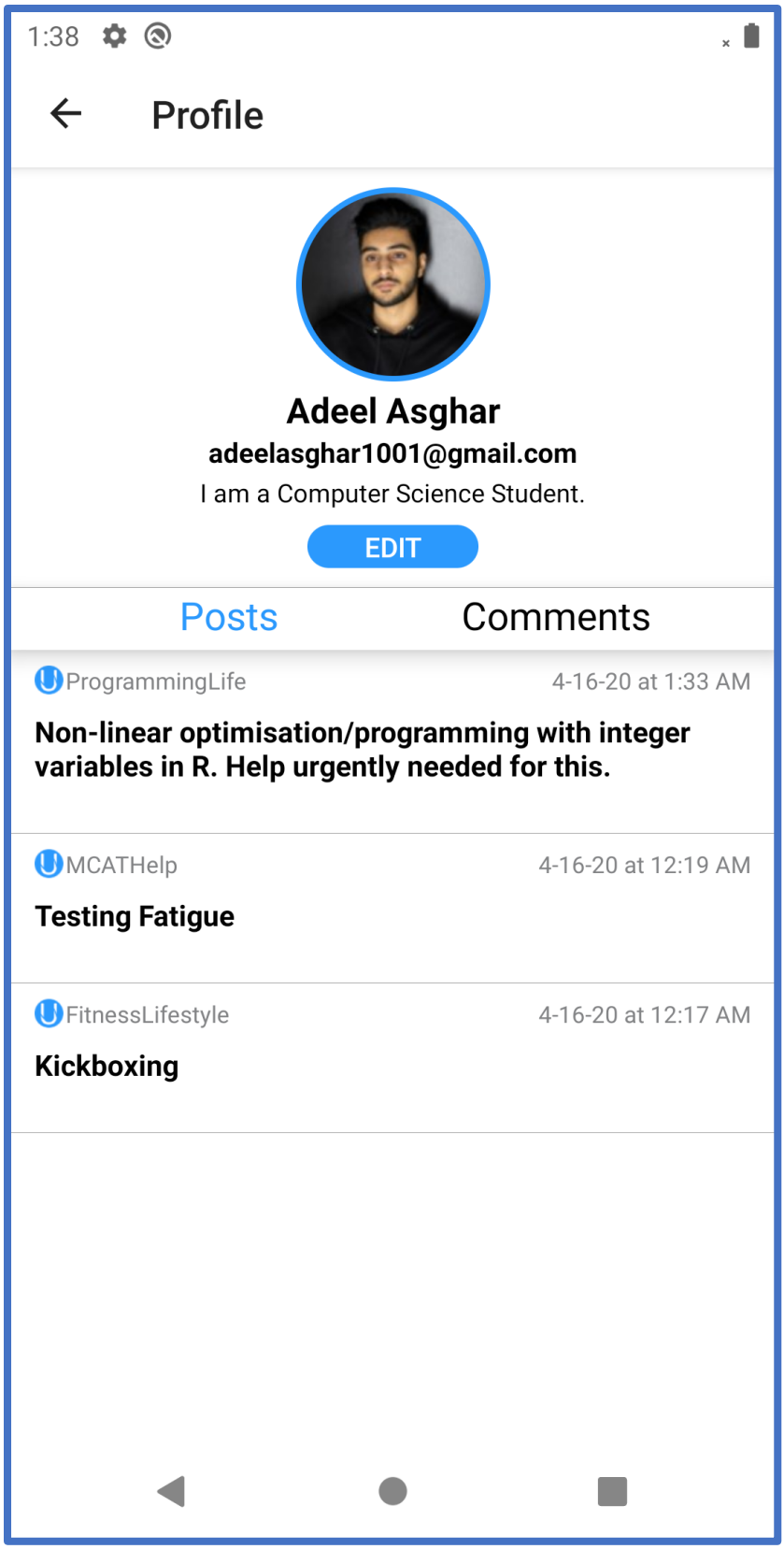


Figure 15: User Profile with User's Posts

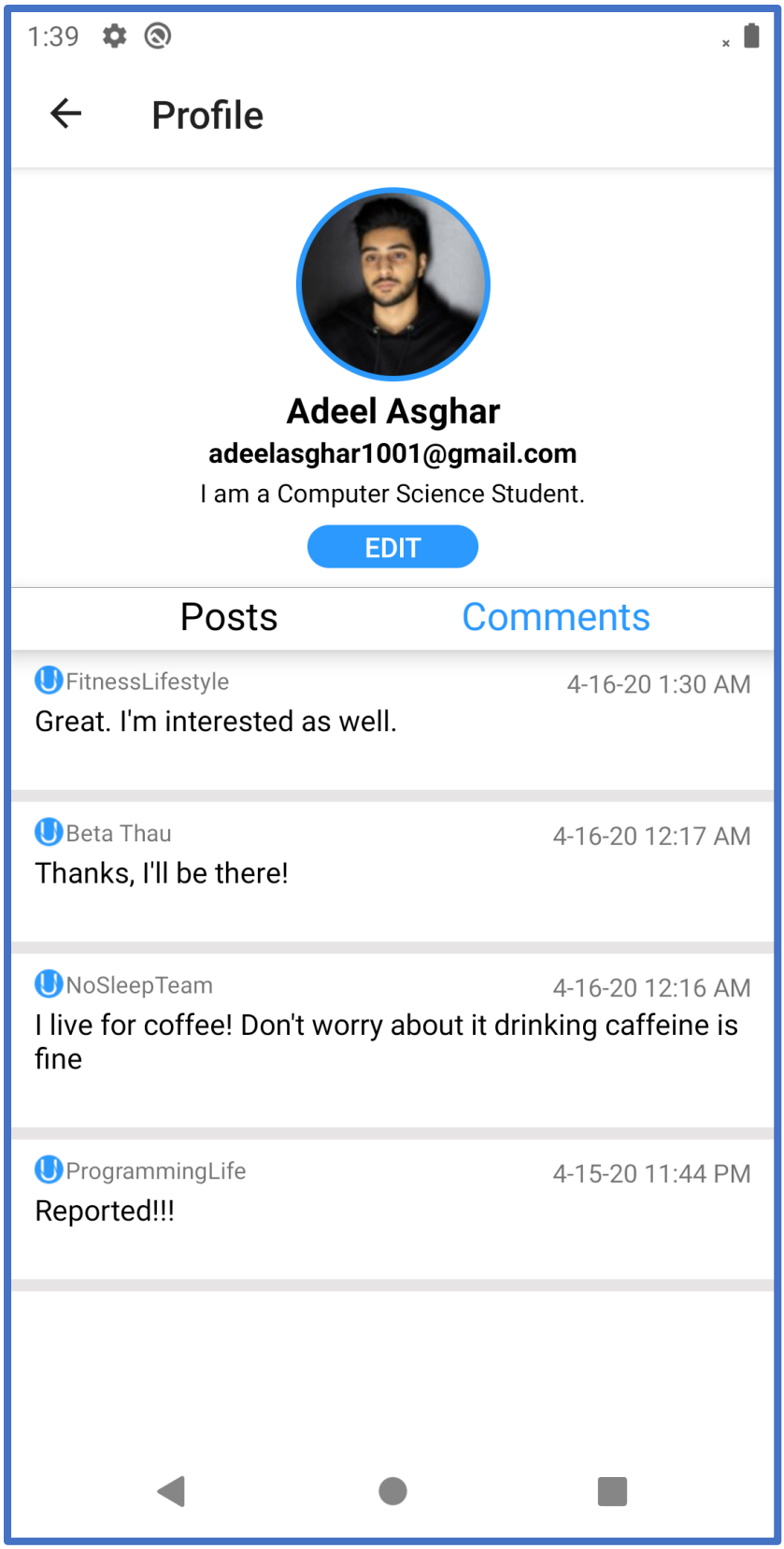


Figure 16: User Profile with User's Comments

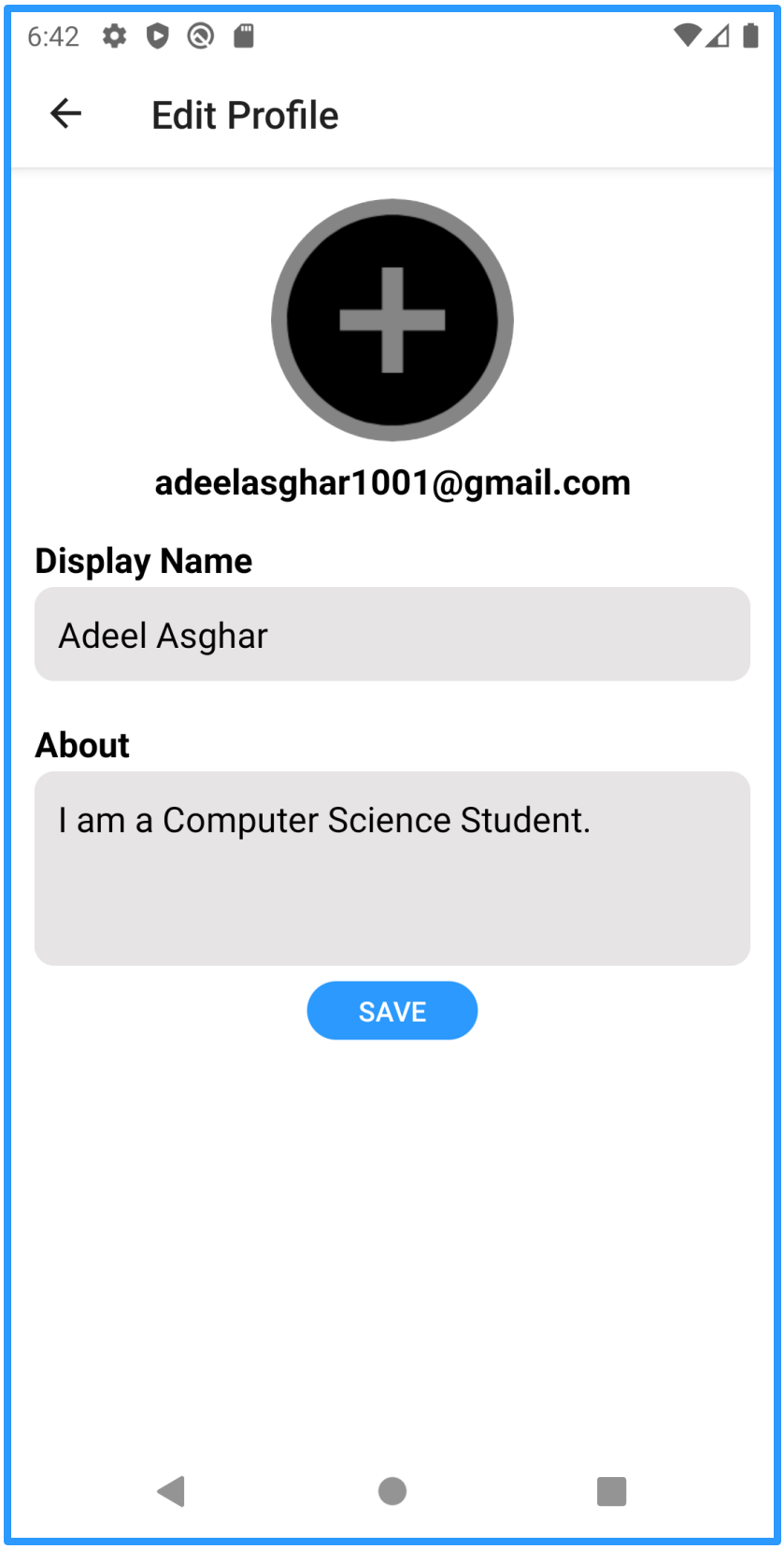


Figure 17: Edit User Profile

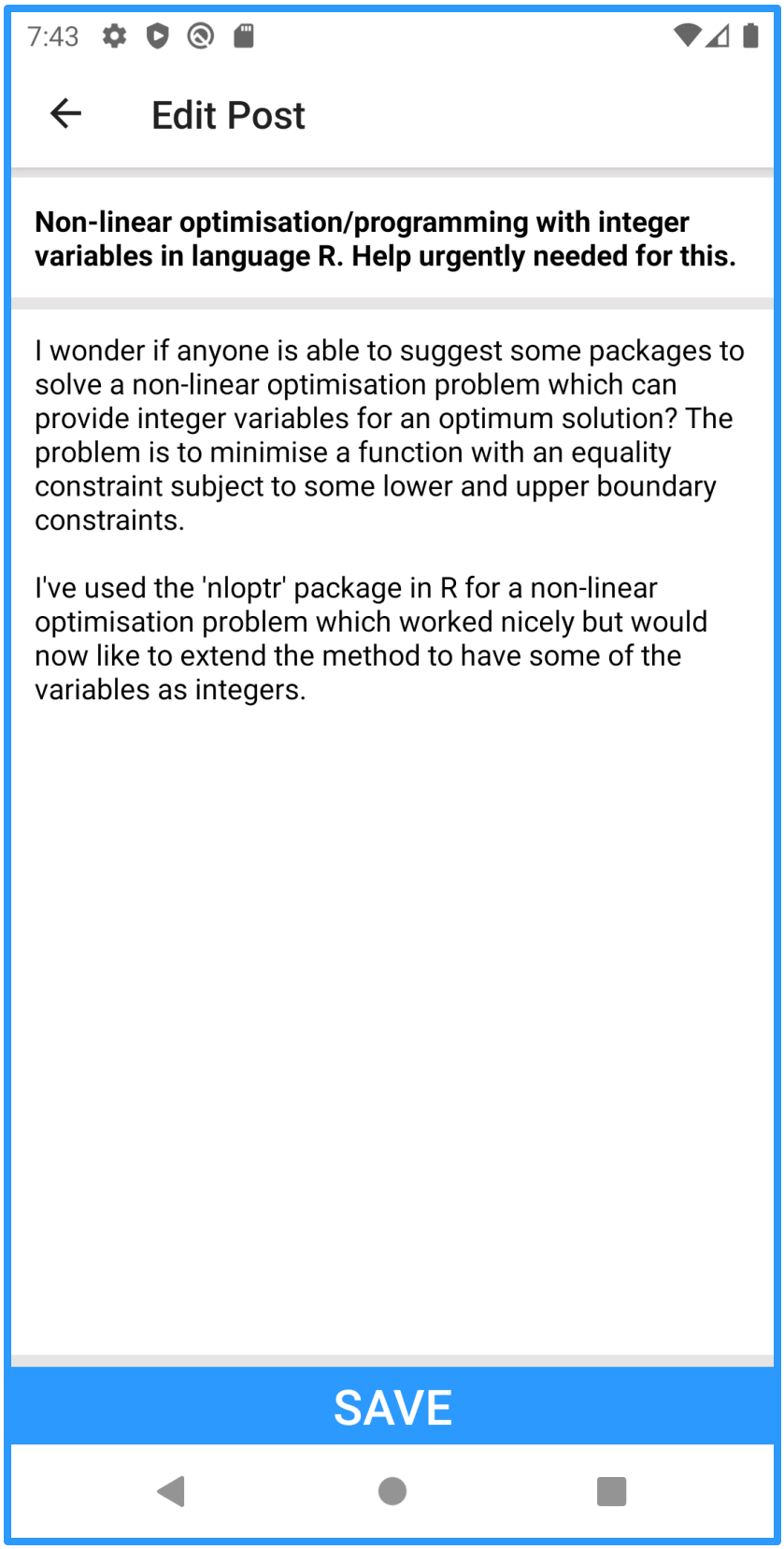


Figure 18: Editing a Post

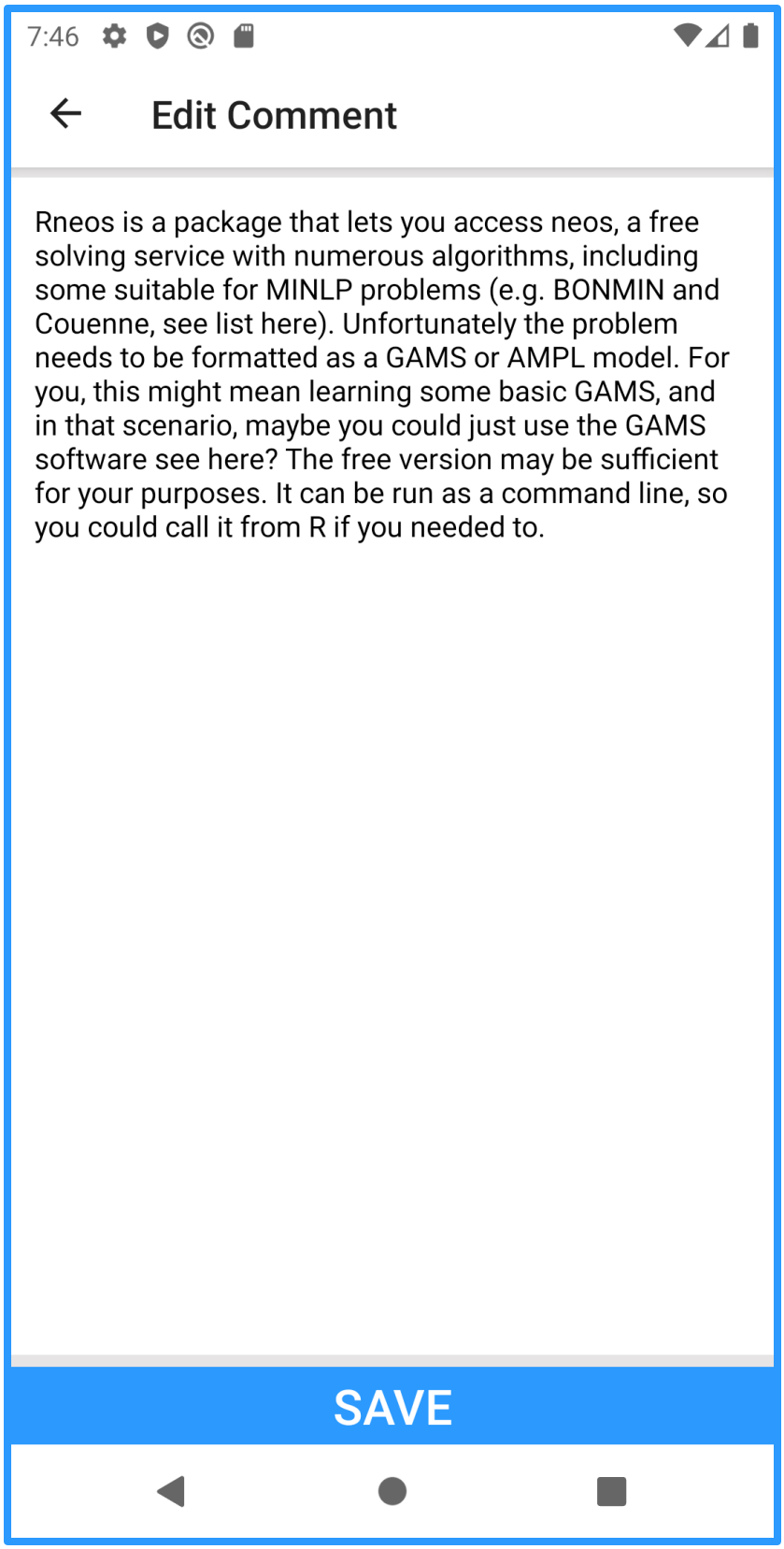


Figure 19: Editing a Comment

Figure 20: Blocked Users

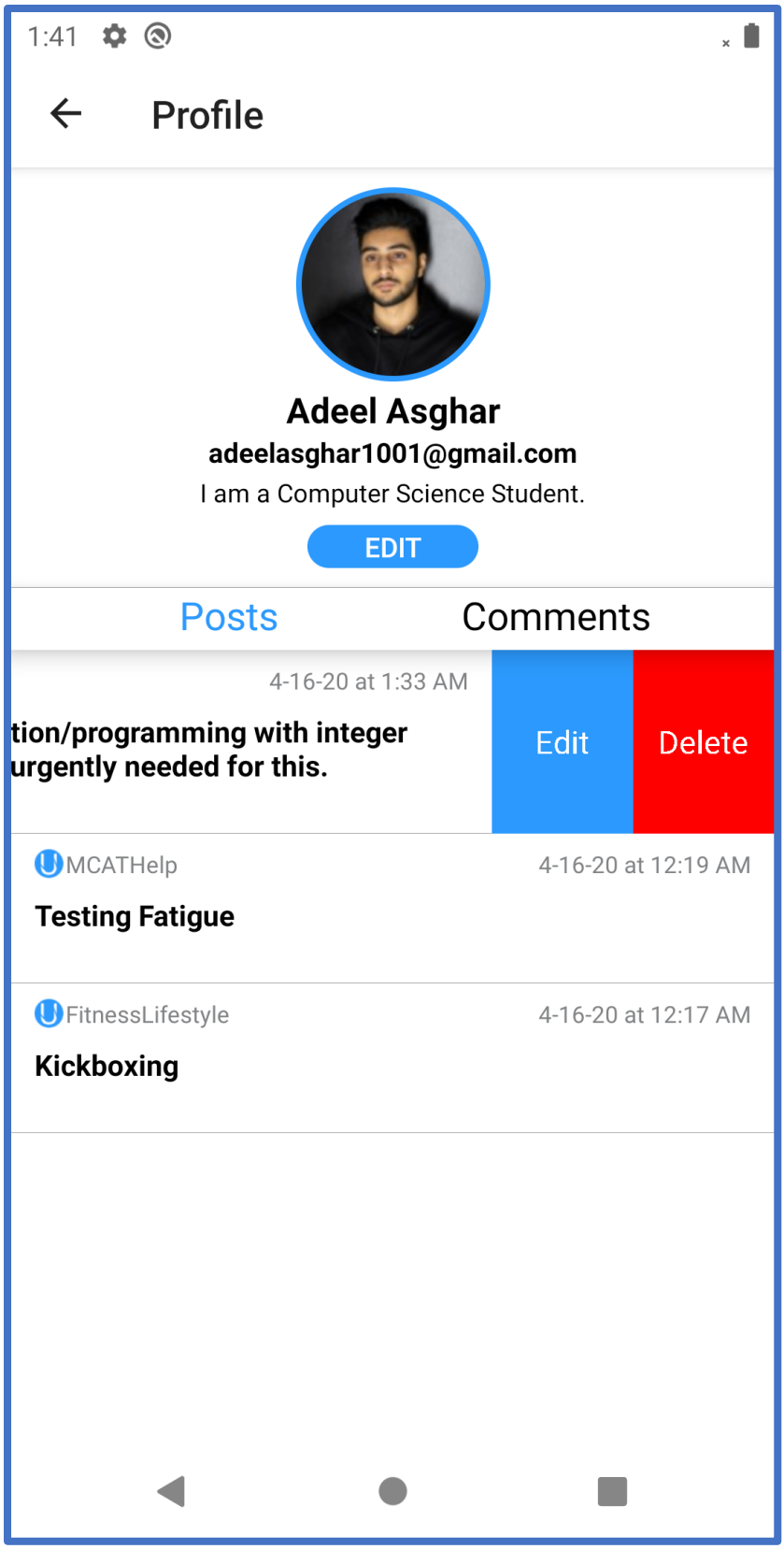


Figure 21: Editing and Deleting Posts

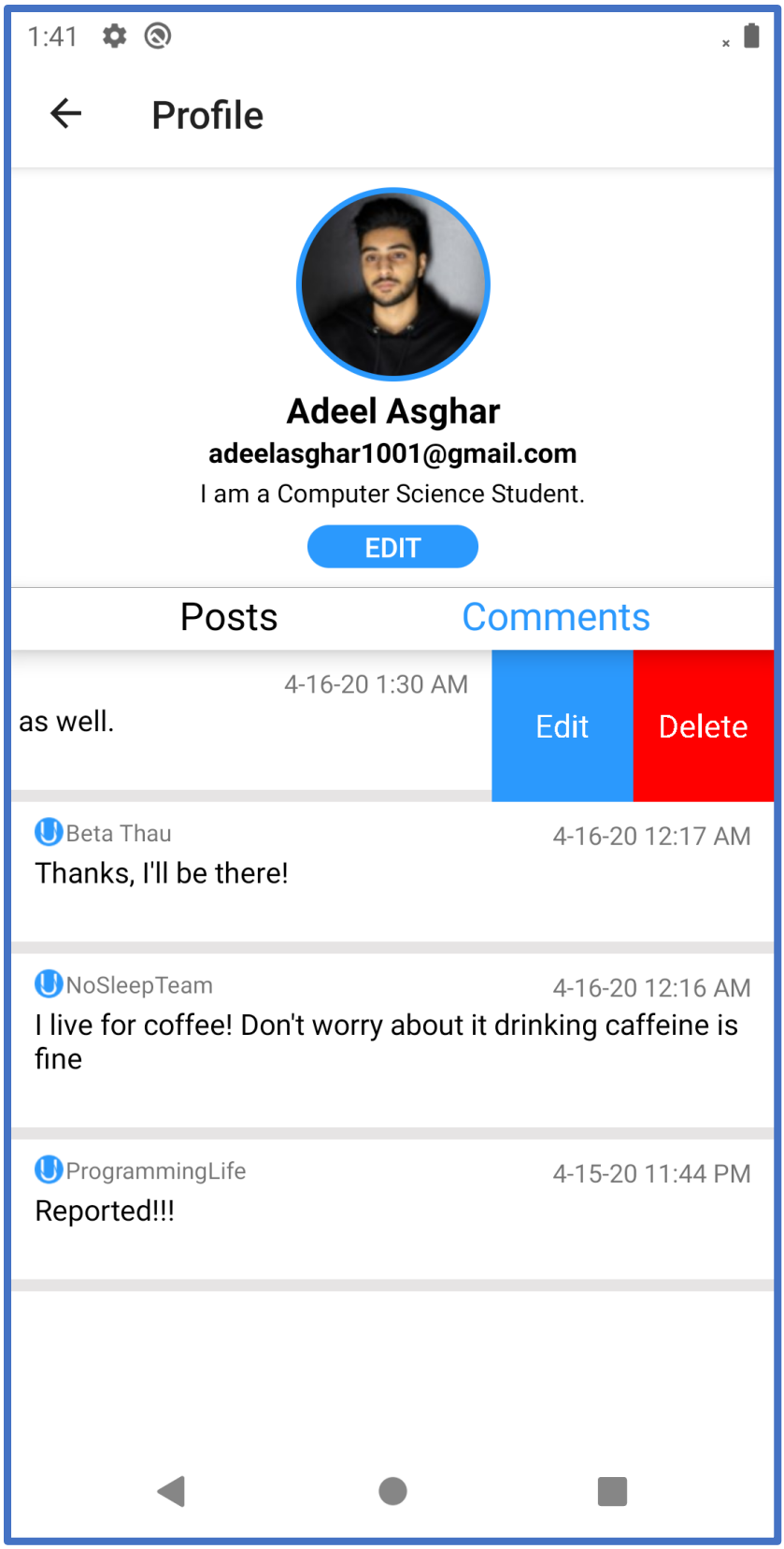


Figure 22: Editing and Deleting Comments

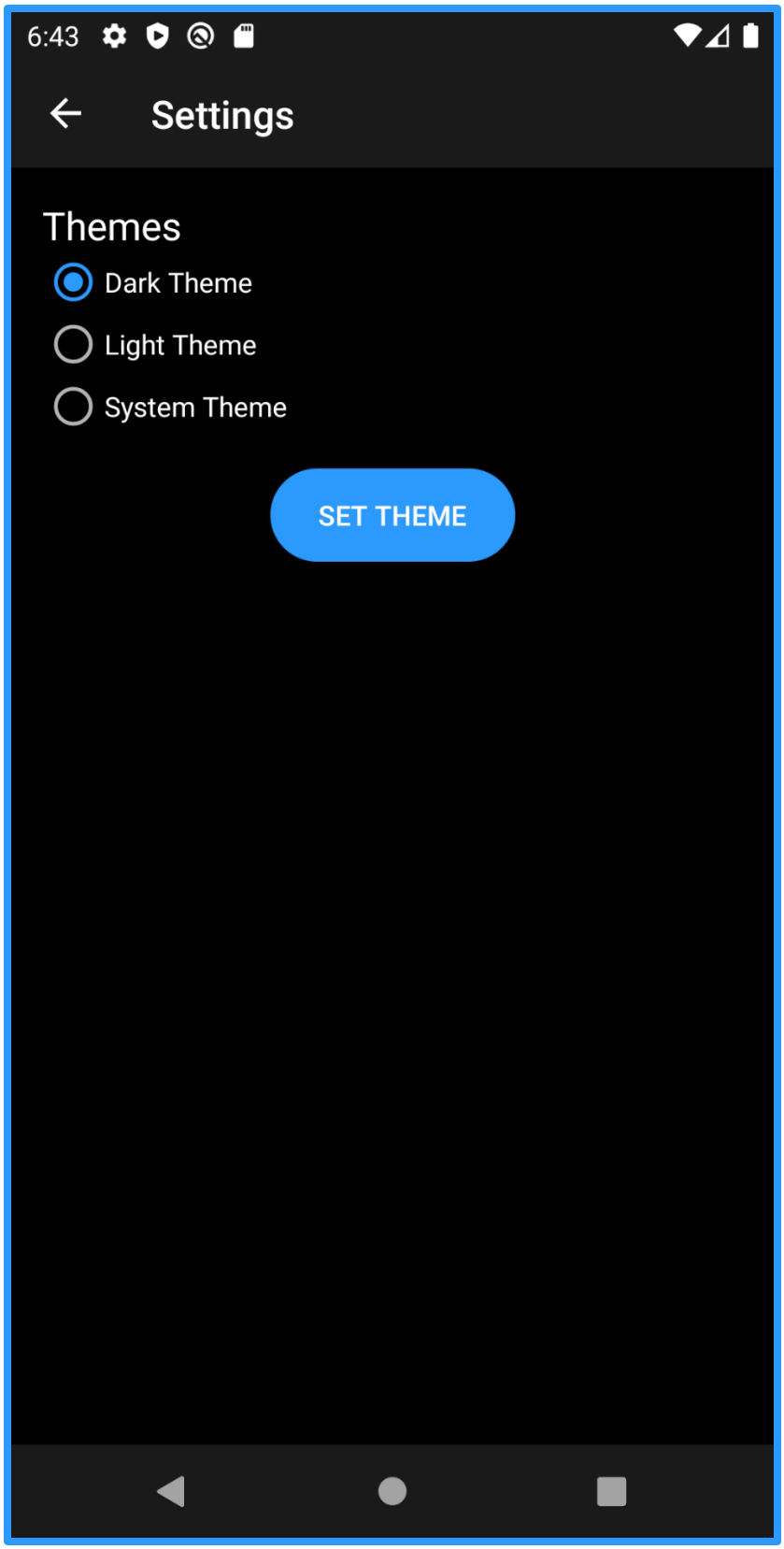


Figure 23: Enabling Night Mode

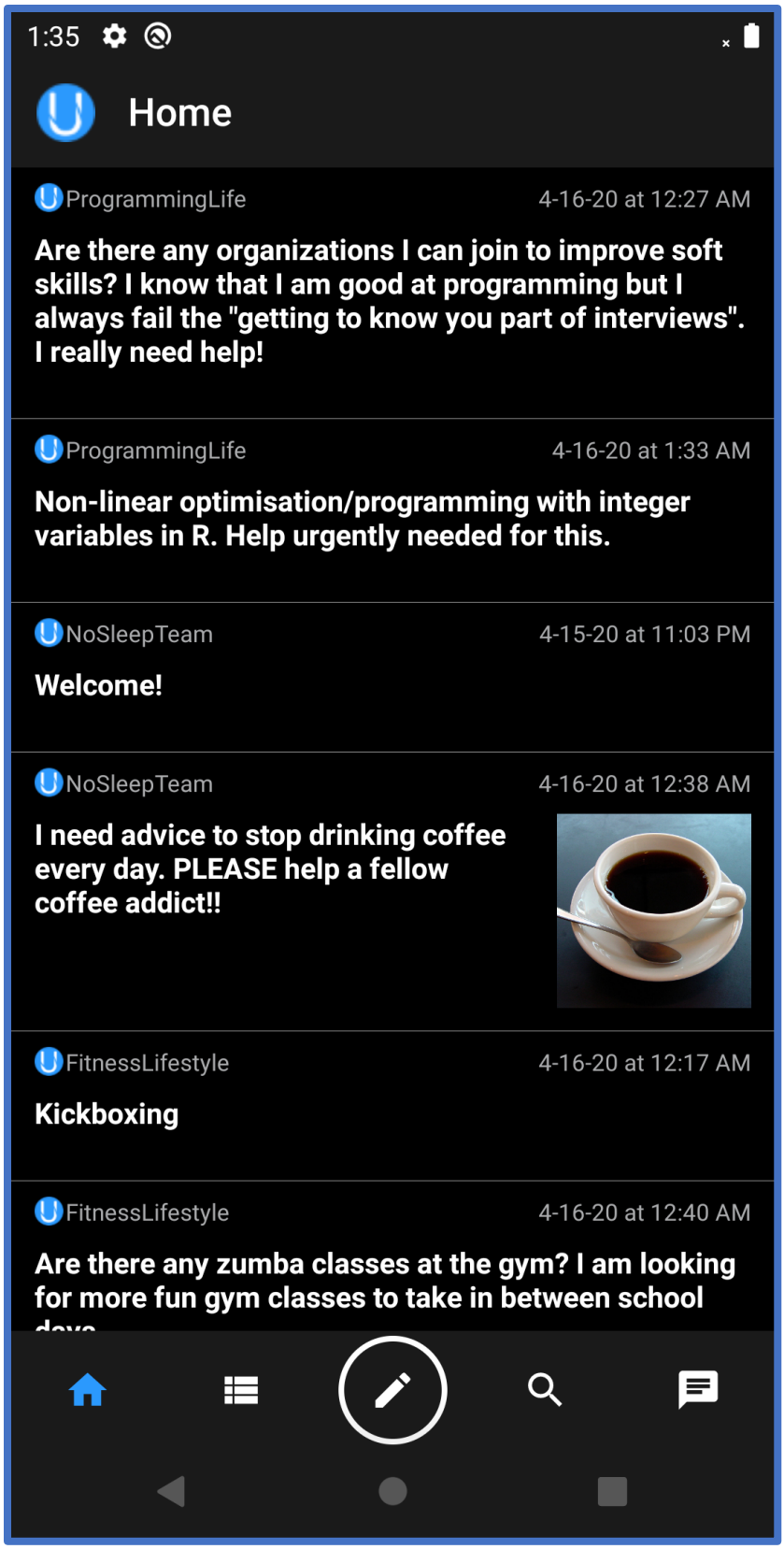


Figure 24: Home Page in Night Mode

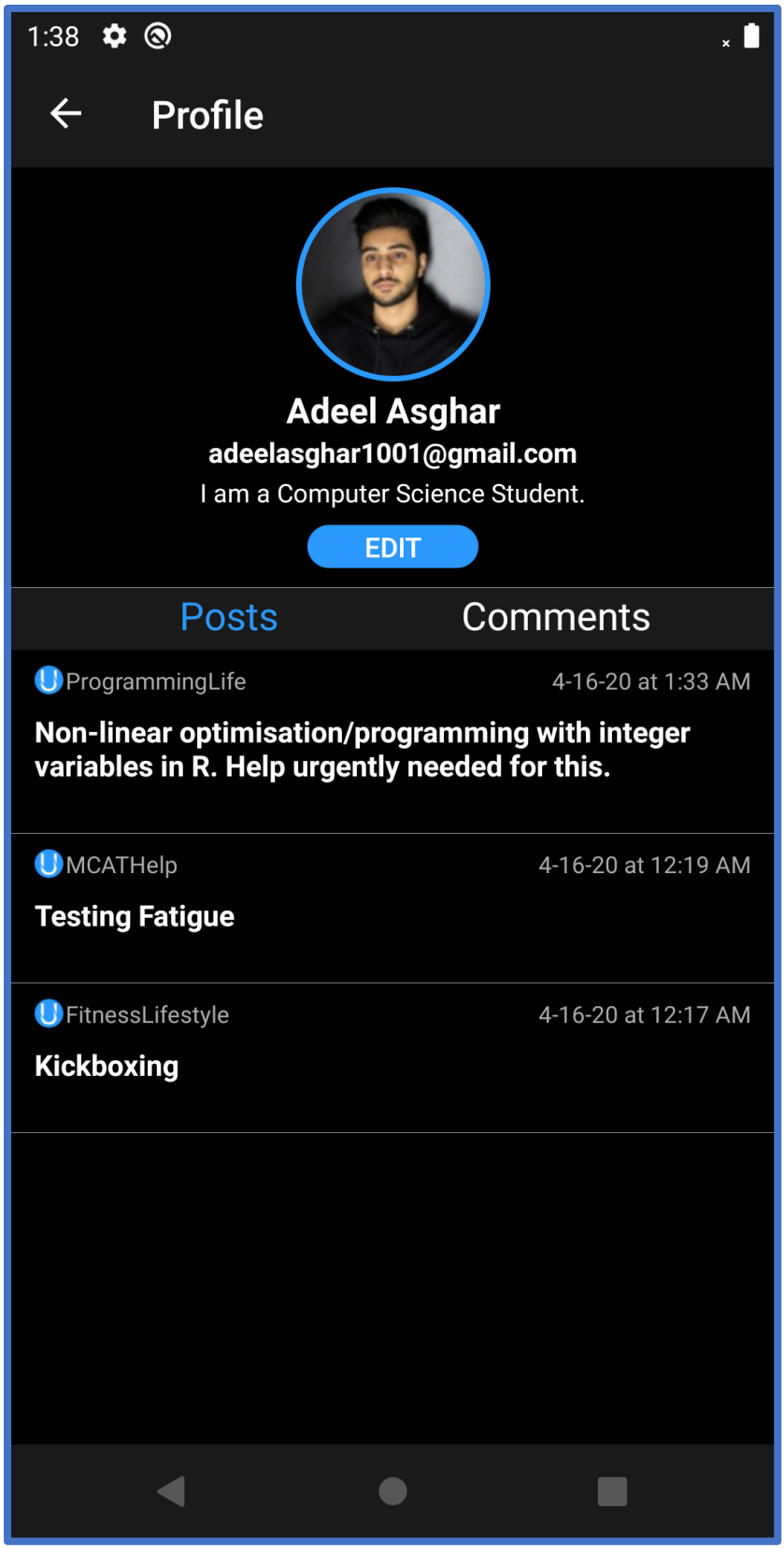


Figure 25: User Profile in Night Mode

5. PRODUCT DESIGN SPECIFICATION APPROVAL

The undersigned acknowledge they have reviewed the Collaborative University Application **Product Design Specification** document and agree with the approach it presents. Any changes to this Requirements Definition will be coordinated with and approved by the undersigned or their designated representatives.

Signature:	Thomas Anter
Print Name:	Thomas Anter
Title:	Client
Role:	Client
Date:	2/24/20

Signature:	Jahnu Best
Print Name:	Jahnu Best
Title:	Client
Role:	Client
Date:	2/24/20

Signature:	
Print Name:	
Title:	
Role:	
Date:	

APPENDIX A: KEY TERMS

The following table provides definitions for terms relevant to this document.

Term	Definition
MVVM	Model-View-View Model is an architectural software design pattern for designing a software product.
API	Application Program Interface
UI	User Interface