

Software Design Specification Document (CS360)

Back Space



Group Number: 05

Aswad Tariq
Zeerak Babar
Haseeb-ur-Rehman Faheem
Ibrahim Sanaullah
Nashrah Shaukat

Course: Software Engineering CS360

Instructor: Suleman Shahid

University: Lahore University of Management Sciences (LUMS)

Version: 1.0

Date: (08/03/2022)

Number of hours spent on this document: 110

Contents

1 Change Log	4
1.1 Project Scope	4
1.2 Change log	4
2 Introduction	6
2.1 Document Purpose	6
2.2 Product Scope	6
2.3 Intended Audience and Document Overview	6
2.4 Definitions, Acronyms and Abbreviations	8
2.5 References and Acknowledgments	9
3 Overall Description	10
3.1 System overview	10
3.2 System constraints	11
3.3 Architectural strategies	12
4 System Architecture	14
4.1 System Architecture	14
4.2 Subsystem Architecture	18
4.3 Data Structure	23
4.4 Database Model	23
4.4.1 Database scheme and detailed description	24
4.4.2 Database	30
4.5 External Interface Requirements	30
4.5.1 User Interfaces	30
4.5.2 Hardware Interfaces	31
5 User Interface Design	32
5.1 Description of the user interface	32
5.2 Information architecture	33
5.3 Screens	34
5.4 User interface design rules	63
6 Other Non-functional Requirements	67
6.1 Performance Requirements	67
6.2 Safety and Security Requirements	67
6.3 Software Quality Attributes	67

Appendix A	68
Appendix B	70

1 Change Log

1.1 Project Scope

The project scope is the same as described previously in the SRS document. More details about project scope can be found in section 2.2.

1.2 Change log

After re-evaluating our project and going over certain use cases and discussion over product flows we discovered that certain essential use cases were missing in the SRS document.

Changes:

1. Users should be able to Add work to find a work tab related to RQ6 in SRS. (NEW USE CASE)
2. Users should be able to join any number of subspaces created by the admin related to RQ31 in SRS. (NEW USE CASE)
3. For Lums exclusive authentication accounts may only be created using a Lums email ID that will be verified using email verification related to RQ 2 in SRS.
4. Modify RQ 14 Add Post the user should be able to add posts to all subspaces not only newsfeed as previously mentioned. (MODIFIED OLD USE CASE)
5. Remove RQ 25 View Terms and Conditions. We felt that the app was intended only for the LUMS community therefore Terms and Conditions were not required for the app.

NEW USE CASE:

RQ 1. Add Find work

- **Description**

The system will allow the user to add a post in the find work tab locally.

- **Input**

The user will click on the “add post” widget in the find work tab. He/She will then fill the form and click the submit button.

- **Processing**

System will process the form and the changes will be stored in the database.

- **Output**

The post will be visible to everyone in the find work tab.

RQ2. Join Subspace

- **Description**

The system will allow the user to join any subspace created by the admin.

- **Input**

The user will click on the “subspace” button on the bottom navigation bar. He/She will be able to view all the subspaces and click the specific subspace and click the join button to join.

- **Processing**

System will process the request and store the changes in the database.

- **Output**

The user will be successfully added in the subspace

MODIFIED USE CASE:

RQ 3. Add Post

- **Description**

The system will allow the user to upload a picture or post a status to the newsfeed.

- **Input**

The user will click the “Post” button. A new screen will open where the user can either write a text or upload any picture before clicking the “post” widget. .

- **Processing**

These changes will be stored in the database after the post has been submitted

- **Output**

The post will be visible to everyone in the respective subspace or Newsfeed.

2 Introduction

2.1 Document Purpose

Back Space is an android application created with the purpose of providing students of LUMS with a platform that would allow them to connect with their respective Luminites. It will provide Luminites a platform to easily navigate course and instructor information by making it easily accessible using in-app search functionality.

The SDS document encompasses the overall system architecture and the user interface design of our product. The document also goes into the detail about the different subsystem architecture, database models, database structure as well as the different hi fidelity screens of our user interface. The types of diagrams used, context, component, activity, sequence and class also help understanding the process of the whole system and subsystems. The document is also extremely important for the smooth implementation of the development phase of the software as it highlights both the system workings and the interface development of our product.

2.2 Product Scope

Back Space is a much needed application with a goal to provide Luminites with an exclusive platform to connect with each other. The application will allow users to access information regarding courses and instructors, chat with their friends in groups, receive real-time notifications and much more. This will save the time of the users by offering better access to up-to-date information using multiple channels and tabs, all on one platform. Setting up the app will also be very convenient. The app will also be freely available and accessible to everyone with an android device. It will provide Luminites a platform to easily navigate course and instructor information by making it easily accessible using in-app search functionality. Backspace will also provide subspace for people where people with similar interests can come together and interact. Overall, it will serve the purpose of being an all in one app for Lums students with regards to being a hub of communication as well as a source of information.

2.3 Intended Audience and Document Overview

Intended Audience:

1-Developers:

Names: Nashrah Shaukat, Aswad Tariq, Zeerak Babar, Ibrahim Sanaullah, Haseeb-Ur-Rehman Faheem

The developers are the team members involved in the development process and documentation of the process of the product. The document serves the purpose of a guideline for the developers making it easier for them to highlight the key features to be included in the software as well as the

functional and nonfunctional requirements. The document also provides the developers with a clear mind set about what the functionality of the product will be like. The entire document is essential for them since the developers also constitute the documentation writers. The reading order recommended below will be useful for the team:

- Section 2 - all subsections
- Section 3 - all subsections
- Section 6 - all subsections
- Section 5 - all subsections
- Section 4 - all subsections

2-Teaching Assistants:

Names: Irzum Bin Mansoor Jafri, Maha Sajid, Suleman Khan, Muhammad Mustansar Saeed, Syed Ibrahim Mustafa Shah Bukhari

The teaching assistants for the project are responsible for overlooking the project's progress and the process of the development phase. The document hence provides a framework for the teaching assistant to assess the functionality of the application and the progress of the development team. The document will also help them to provide feedback. The document is also useful for the teaching assistants to assist the developers in the further phases of software development. The contribution statements and the group log will also help the teaching assistants to assess each individual developers' contribution to the project. The teaching assistant can also go through the entire document in any specified order.

- The entire document is of importance for the teaching assistants.

2-Instructor:

Names: Dr. Suleman Shahid

The instructor is the supervisor for the course. He will be overlooking the project during the semester and will be overseeing the progress and guiding the team through each phase of the project.

- The entire document is of importance for the instructor.

Document Overview:

- **Section 2: Introduction**

Section 2 provides a brief introduction to our project as well as the document and a brief overview of what the reader will find in this section.

- **Section 3: Overall Description**

Section 3 provides a general description of our software system including its functionality and the constraints present in the system

- **Section 4: System Architecture**

Section 4 provides a high-level overview of how the functionality and responsibilities of the system were partitioned and then assigned to subsystems or components. It includes a UML component diagram as well as 3 activity diagrams.

- **Section 5: User Interface**

A description of the user interface design of the software is presented in Section 5. A brief description of the front-end and back-end is presented, followed by a description of each screen present in our application and how every element on the screen maps to a function.

- **Section 6:Non-Functional Requirements**

Section 6 provides the performance, safety and security requirements of the software as well as software quality attributes.

- **Appendices:**

Appendix A: In-group meetings and TA meetings

Appendix B: Contribution statement of the team

2.4 Definitions, Acronyms and Abbreviations

<i>Terms</i>	<i>Definitions</i>
Back-end	It refers to the program's code that handles database and server-side rendering and cannot be accessed by a user.
Deployment	One of the important steps of creating a website or an app. It means setting up the app or website live so that it could be used by others in their system.
Front-end	It is part of a website or web application through which a user can interact.
LDF	Lums discussion forum a platform where lums students communicate over facebook
LUMS	A university located in Lahore,Pakistan(Lahore University of Management Sciences).
Malicious	It refers to the intention of causing any harm.
Bottom Navigation	A navigation bar on the bottom of the screen which provides quick navigation between top-level views of an app.

Side Navigation	vertical navigation to place at the side of a page.
Subspace	Tabs that contain sub forums related to a specific category or genre of posts can only be created or removed by admins and users must join these subspaces to view their posts.
Web Application	An application that runs on a web server through a web browser.
Widget	Building blocks of Flutter app's user interface, it is a framework that displays information to the user.
Figma	web-based graphics editing and user interface design app.
SRS	Specification Requirement Specification
SDS	Software Design Specification
Database	organized collection of structured information that is stored in computer system
TA	Teaching Assistants

2.5 References and Acknowledgments

References:

- UML Component Diagram:
<https://creately.com/blog/diagrams/component-diagram-tutorial/>
- Design Heuristics:
<https://www.nngroup.com/articles/ten-usability-heuristics/>
- Sequence Diagram Guide:
<https://www.youtube.com/watch?v=pCK6prSq8aw>
- Firebase:
<https://firebase.google.com/>

Acknowledgements:

- Draw.io for Database Schema
- LucidChart for System Architecture diagrams
- Figma for screens

3 Overall Description

3.1 System overview

Overview:

Back Space is an android application created with the purpose of providing students of LUMS with a platform that would allow them to connect with their respective Luminites. Although a similar platform already exists (LUMS Discussion Forum) on facebook, the existing platform has countless limitations including non categorized data, tedious induction process and hard to find data. BackSpace will be a new self-contained product, not dependent on any previous versions. It will provide Luminites a platform to easily navigate course and instructor information by making it easily accessible using in-app search functionality. Backspace will also provide subspace for people where people with similar interests can come together and interact. Overall, it will serve the purpose of being an all in one app for Lums students with regards to being a hub of communication as well as a source of information. The application will support 2 actors: **users** and **admin**.

Back Space aims to connect its users through various communication platforms. This means that the application should provide with enough functionality so the users, students can navigate the application easily and are able to connect and interact with their peers and also the administrators who will be moderating the application can also do so efficiently.

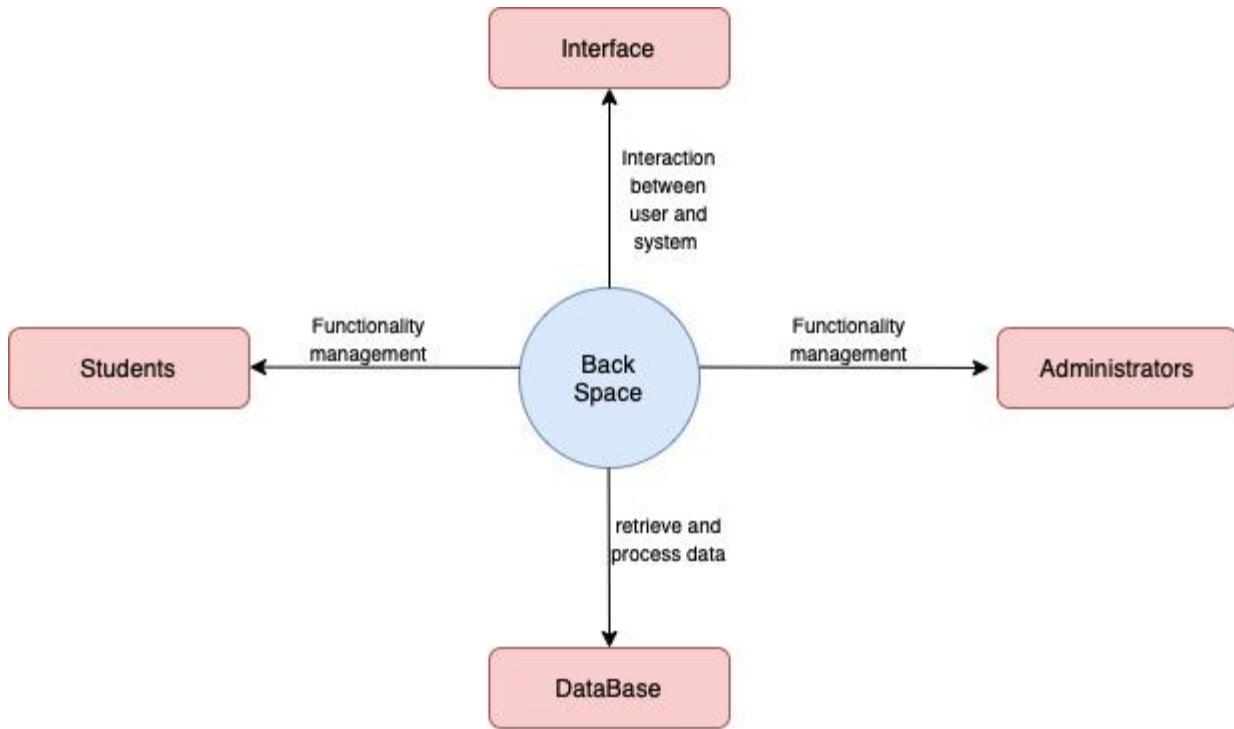
Main Functionality:

The major functionality of the application would include:

- **Module management of the Users:**
This would enable the users to carry out multiple functionalities in the application including, Add Post, Delete a Post, Like a Post, Search Users, Search joined subspace, Search Find work Tabs, Search Newsfeed among other things.
- **Module management of the Administrators:**
The administrators would be able to carry different functionalities such as approving posts, denying posts, adding events, creating subspaces and adding events.
- **Database management:**
The backend database would be accessed to process queries and access stored data.
- **Interface:**
The user interface would allow for a smooth interaction between the user and the system software.

High Level Context Diagram

The following high level context diagram gives a brief overview of the entire system.



3.2 System constraints

- The users must have access to a smartphone since our product is a mobile only application.
- The users must have a stable internet connection for the application to work smoothly.
- The application is entirely in the English language so it is assumed that the users are familiar with english.
- Since the developers would be using Firebase basic free plan, only a limited amount of storage would be available, 1 GB specifically. Hence the application can not hold too much data and the developers might have to upgrade for more storage.
- Our product is a social media application, hence it is assumed that the user has an email to sign up with and has the basic knowledge to navigate the application.
- Since only the five developers of the development team would be working, it would be hard to identify and fix all of the bugs in the product.
- Users might face delays due to low signal strength.
- Development team is not responsible if the application is breached by a hacker.
- The application will make use of multiple icons and widgets, and it is assumed that the user has some basic knowledge about the different icons present in the system
 - If the user clicks the notification widget, they will be taken to the notification screen.
 - Similarly, if the user clicks the chat widget, they will be taken to the chat screen.

3.3 Architectural strategies



For our project we will be using flutter for the front-end and firebase for backend.

Front-End:

We will be using Flutter which is a free and open-source mobile UI framework created by Google for mobile app development. We will be using this for our front-end for the following reasons:

- While choosing a framework it is necessary to take into consideration the performance and efficiency of the mobile. With flutter, we can easily change our code and see results in real-time. The Hot-Reload property allows users to save and update applications instantly.
- Since we have limited time, learning flutter would be easy as it is one of the fastest growing languages for mobile applications and is easy to learn.
- Flutter gives us widgets which makes apps more user-friendly and easy to use. Since our application is going to be used by many students, the application should be visually appealing.
- It also provides a user with smooth app experience and quick loading since flutter consists of a Skia graphics library which is fast as compared to other libraries.

Back-End:

We will be using firebase for back-end which is a google-backed application development software. Following are some of the reasons that were taken into consideration before selecting firebase as a back-end

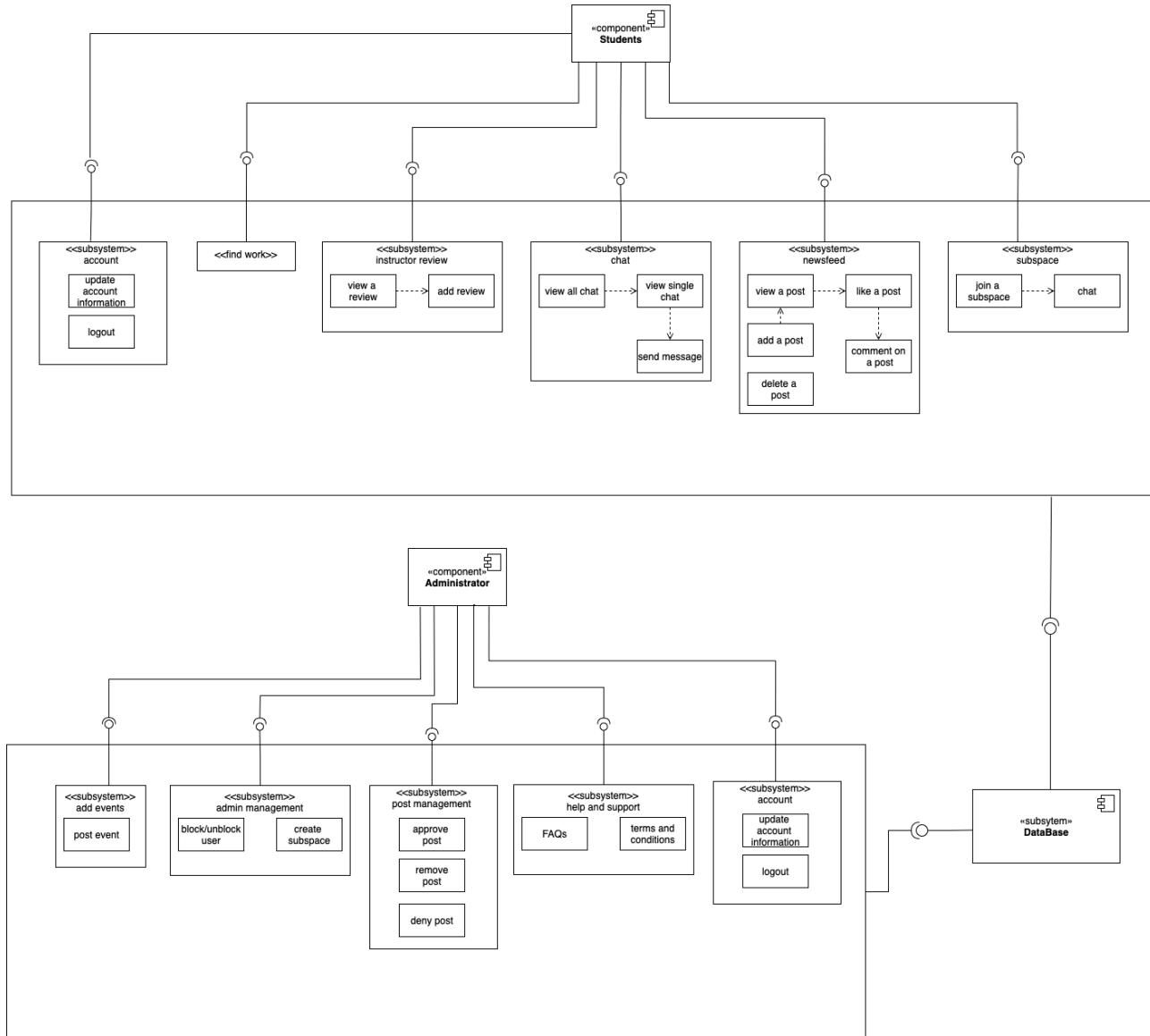
- Although the free version of firebase only allows 1GB of storage, we can easily upgrade to one of the provided plans by firebase instead of changing the entire backend in case we need more storage.
- Firebase allows free multi platform authentication. This is particularly important for our social media app to authenticate users before they sign-in to our app.

- Firebase boasts a real-time database. The change in the database, either by the user or the admin, will be instantly visible on the app in realtime.
- Firebase is integrated with google analytics, which reports how users are behaving towards our mobile application. This will provide us with analytics of the usage of our application.
- Firebase provides fast, easy and most importantly, a secure web hosting platform. This is especially important since our app should only be accessed using a LUMS account.

4 System Architecture

4.1 System Architecture

UML Component Diagram

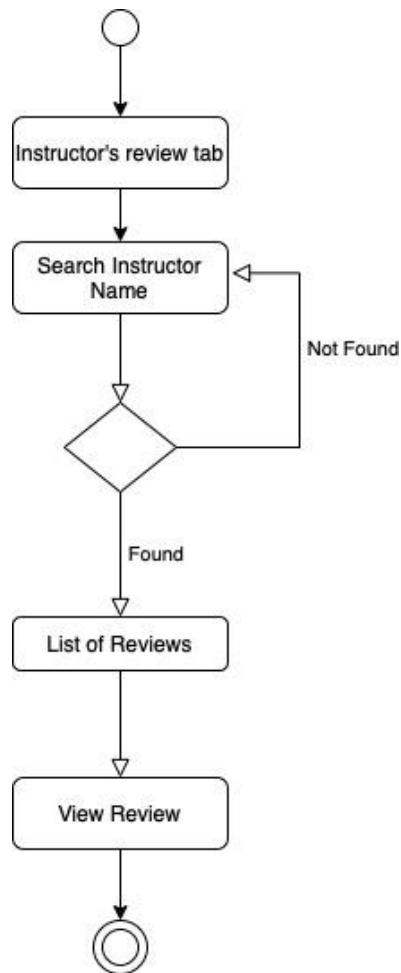


The component diagram shows two major components of our system design, the administrators and the students, the users of our product. Since both types of users would have different functionalities, they are shown as different components. Students would have functionalities such as account, where they can manage their account, find work, accessing instructor review which would include viewing reviews and then adding review, chat system which would include viewing chat and sending messages, newsfeed which would include viewing posts, liking, commenting on posts and joining subspaces. All these functionalities are shown as different subsystems. The subsystems further show the objects which are dependent on others. Similarly the administrators have other major subsystems such as adding events, admin management which includes blocking/unblocking users and creating subspaces, managing posts and help and support subsystems. The two components are further connected to the database. The diagram is a high level presentation of how the system would work hence provides very little detail to the flow of the individual objects.

Activity Diagram

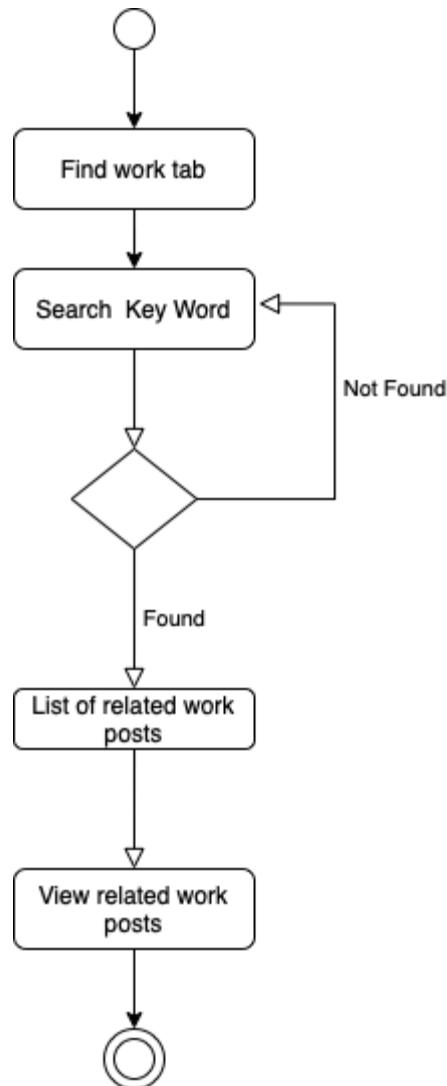
1- Search Instructors Review

The activity diagram shows how the user can search a particular instructor's reviews. Users would have to navigate to the instructors review tab and from there search an instructor's name. If the instructor's name is not found in the database then the user would have to try again. Once the instructor's name is found all the relevant entries are displayed. The user can select a particular review and view that.



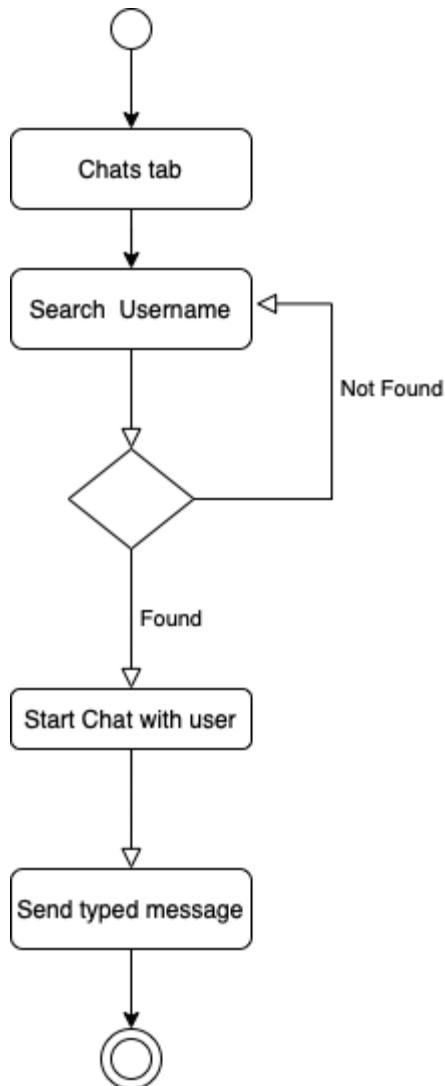
2- Search Find Work tab

The activity diagram displays how the user can navigate to the find work tab. Users would have to go to the find work tab on the navigation tab. The user can then search keywords relevant to the work they are looking for. If search results are available they would be displayed. The user can then view a certain post.



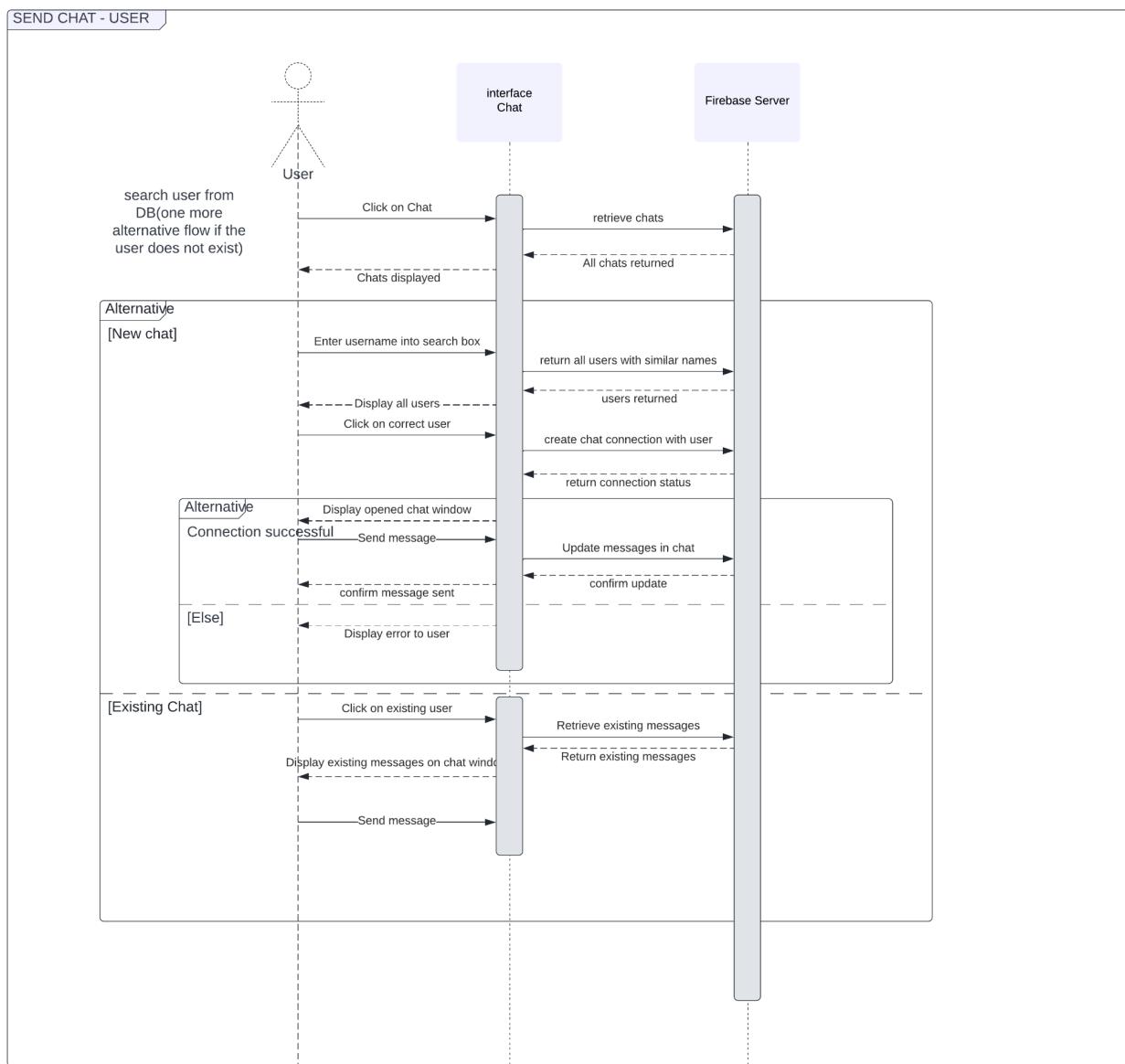
3- Start Chat

To chat with other users, the user would have to navigate to the messaging tab on the navigation bar. The user would then search for a particular user. If the username is found the user would click on their profile and start chatting with that user.

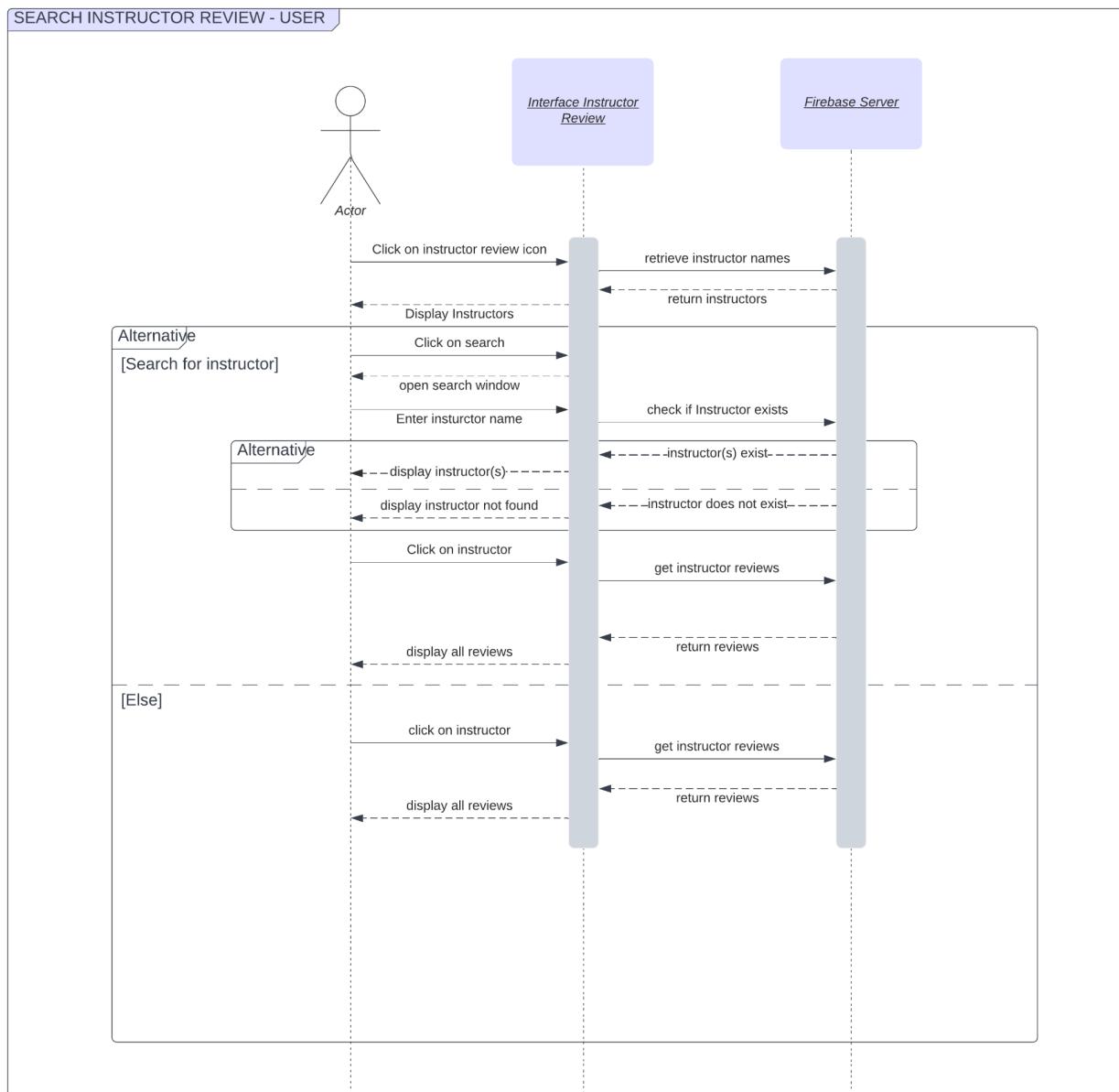


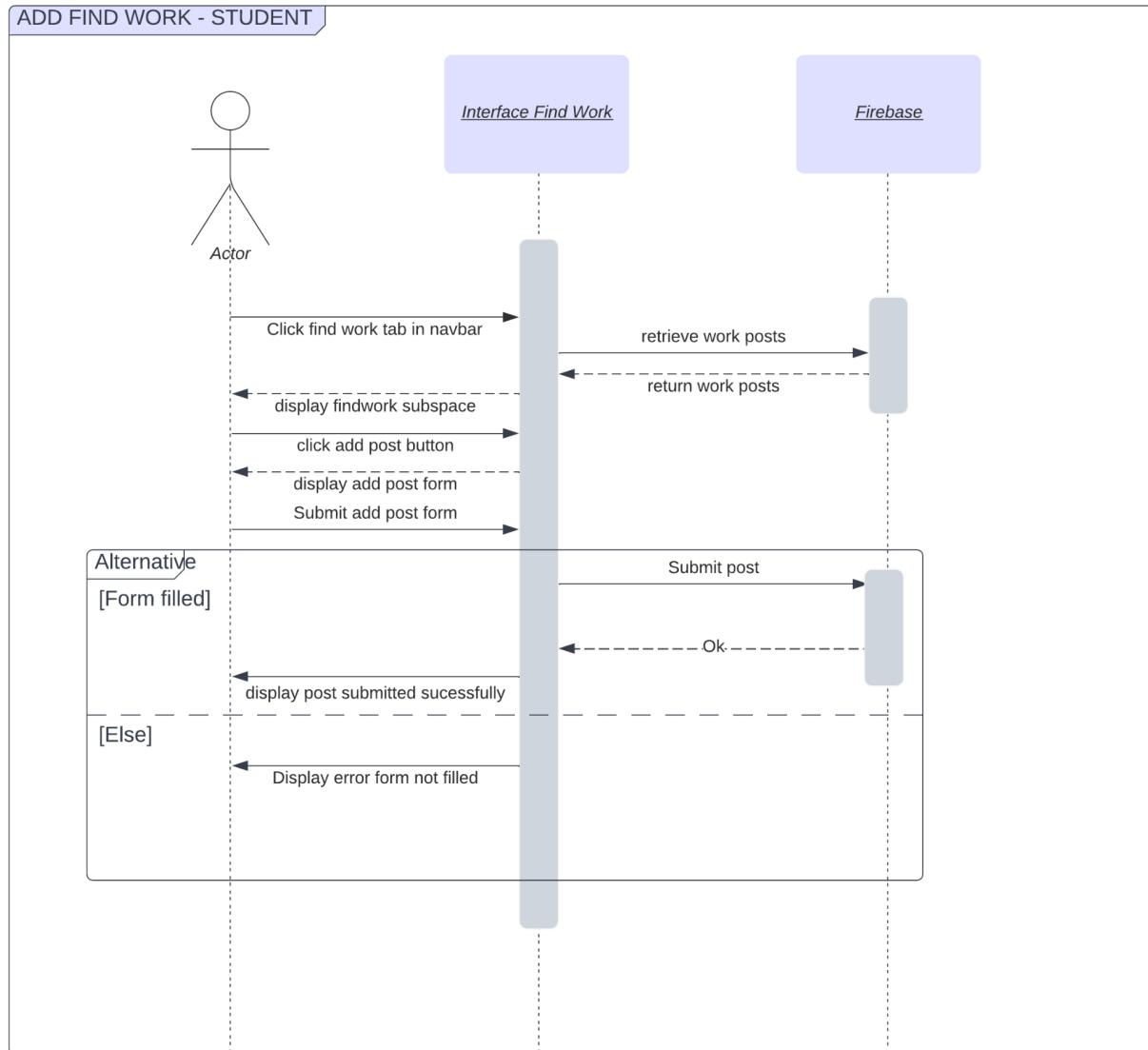
4.2 Subsystem Architecture

Sequence diagram: Send Chat

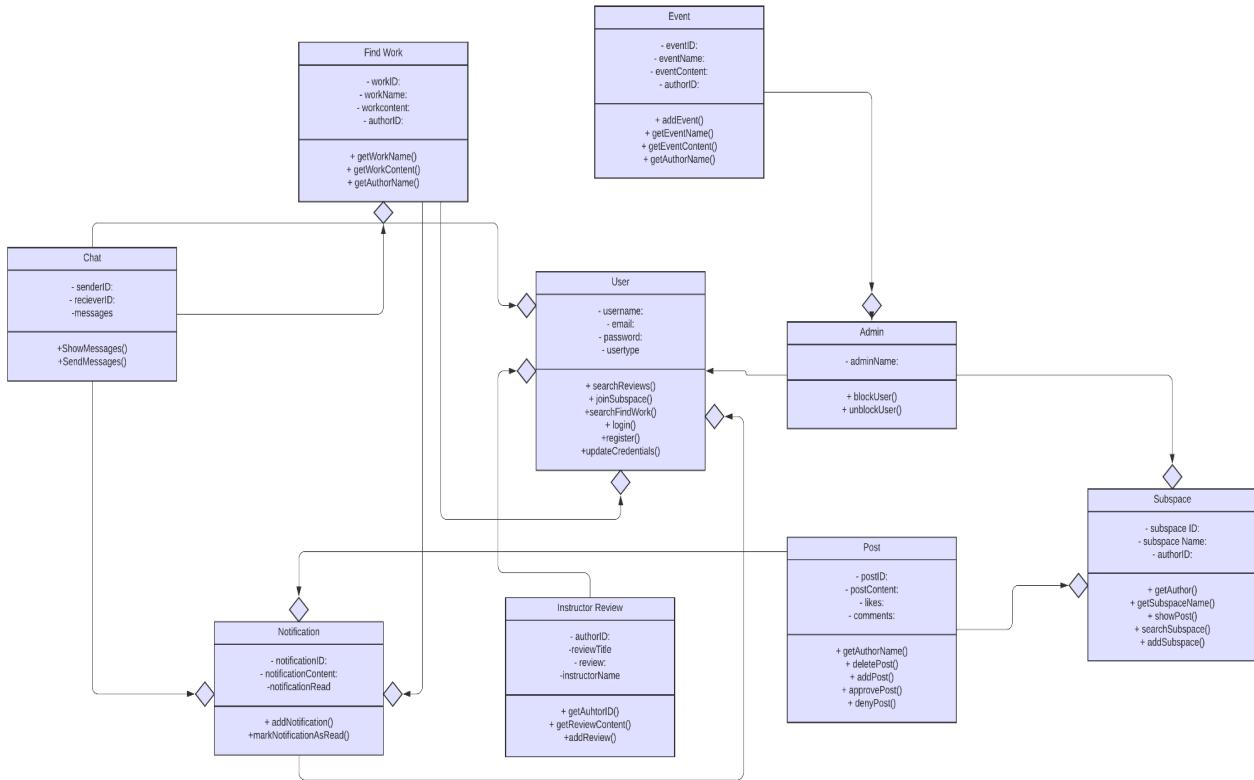


Sequence diagram: Search Instructor Review

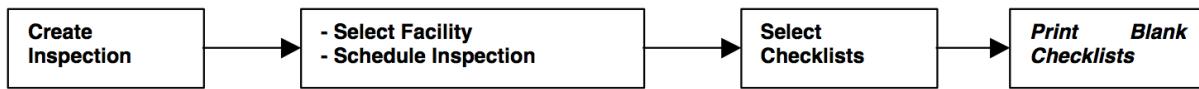


Sequence diagram: Add Find-Work

Class Diagram:



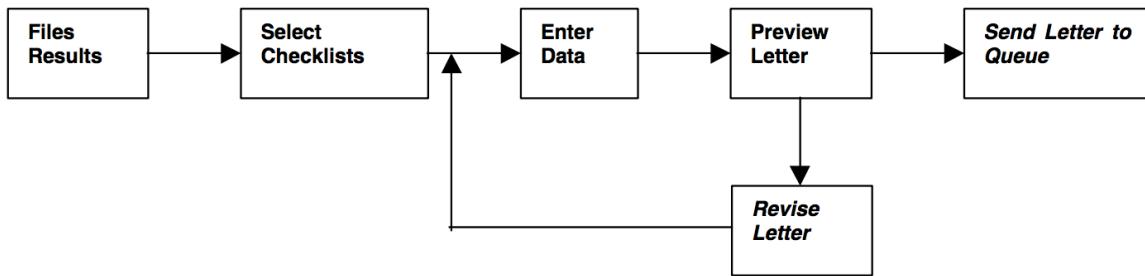
3.1.2 Create Inspection



3.1.3 During Inspection



3.1.4 Post-Inspection



Example: Flow of each component

4.3 Data Structure

<A description of all data structures including internal, global, and temporary data structures.

4.3.1 Internal software data structure

Data structures that are passed among components of the software are described.

4.3.2 Global data structure

Data structures that are available to major portions of the architecture are described.

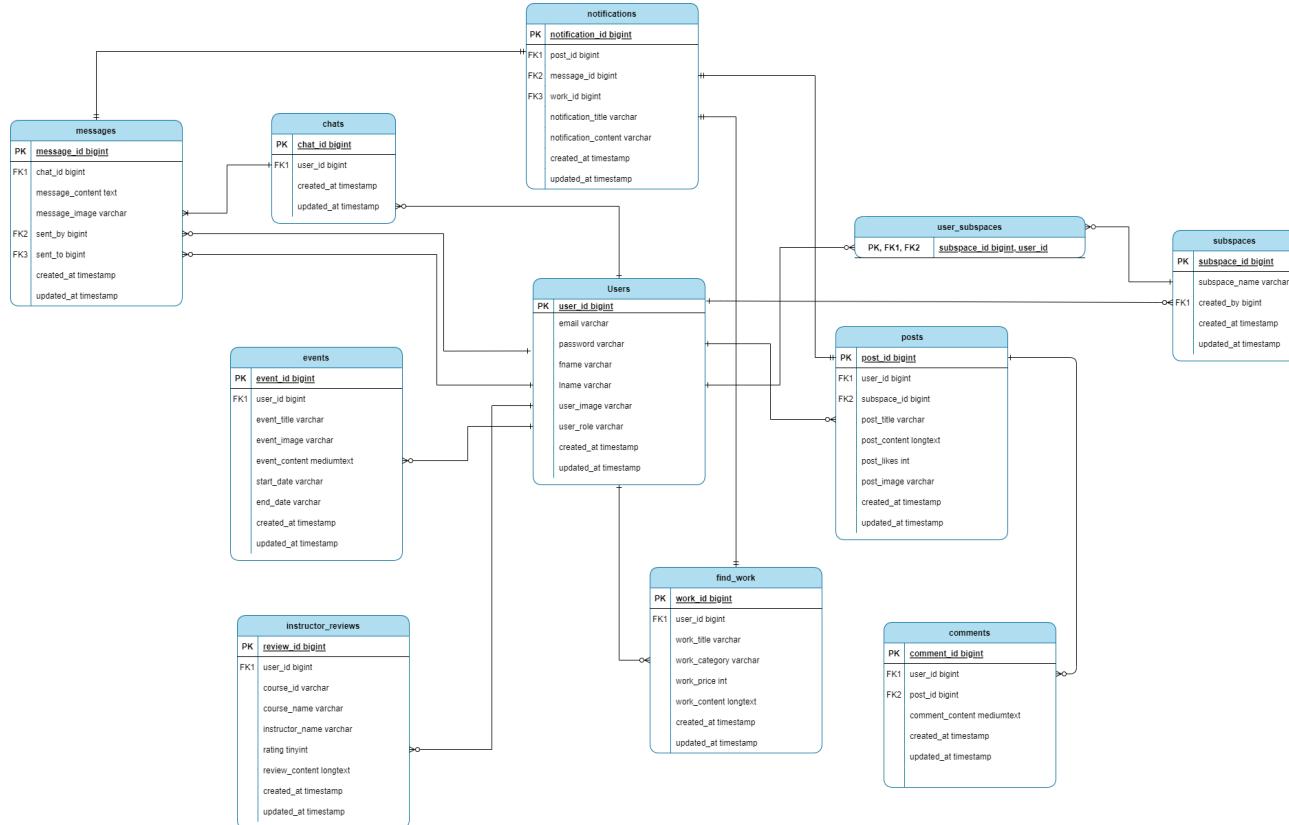
4.3.3 Temporary data structure

Files created for interim use are described.

>

4.4 Database Model

4.4.1 Database scheme and detailed description



User Table

The User table will store the information of both types of users(admin and user). It will differentiate between the types of user using user_role. The timestamp at which the account was created and last updated will also be stored.

Title	Data Type	Description
user_id	bigInt	user id will be generated automatically and will act as a primary key of the table.
email	varchar(225)	email of the user
password	varchar(225)	password of the user
fname	varchar(225)	first name of the user

lname	varchar(225)	last name of the user
user_image	varchar(225)	profile picture of the user account
user_role	varchar(225)	whether the user is admin or user
created_at	timestamp	time at which the account was created
updated_at	timestamp	time at which the account was last updated

Post Table

The post table will store all the information related to a post. It will save the title and content of the post, as well as the number of likes on the post. The `post_id` of each post will be unique and act as the primary key of the table.

Title	Data Type	Description
post_id	bigInt	post id will be generated automatically and will act as a primary key of the table.
user_id	bigInt	user id will act as a foreign key and will be used to distinguish that the post belongs to which user.
subspace_id	bigInt	subspace id will act as a foreign key and will be used to distinguish that the post belongs to which subspace.
created_at	timestamp	exact time at which the post was uploaded
post_title	varchar(255)	title of the post
post_content	longtext	content of the post
post_likes	int	the number of likes on the post
post_image	varchar(255)	image attached with the post

Comment Table

The comment table will contain all the information regarding the comments on the post. Both the `user_id` and `post_id` will act as the foreign key in this table used to distinguish the user who posted the comment and the post on which the comment was posted. Timestamp at which the comment was posted will also be stored.

Title	Data Type	Description

comment_id	bigint	comment id will be generated automatically and will act as a primary key of the table.
user_id	bigint	user id will act as a foreign key and will be used to distinguish that the comment belongs to which user.
post_id	bigint	post id will act as a foreign key and will be used to distinguish that the comment belongs to which post.
created_at	timestamp	store the exact time at which the comment was posted
comment_content	mediumtext	content of the comment

Notification Table

The notification table will contain all the information regarding the notification. Post_id, chat_id and work_id will act as a foreign key in this table. The title and the content of the notification will also be stored. The timestamp at which the notification was sent will be saved in the notification table as well.

Title	Data Type	Description
notification_id	bigint	notification id will be generated automatically and will act as a primary key of the table.
post_id	bigint	post id will act as a foreign key and will be used to distinguish that the notification was triggered by which post.
chat_id	bigint	chat id will act as a foreign key and will be used to distinguish that the notification was triggered by which post.
work_id	bigint	work id will act as a foreign key and will be used to distinguish that the notification was triggered by which work post.
created_at	timestamp	store the exact time at which the notification was sent
notification_title	varchar	title of the notification
notification_content	text	content of the notification

Message Table

The message table will contain all the information regarding the messages. chat_id,sent_by and sent_to will act as the foreign key in this table. The message content and the picture (if attached) will be stored. The timestamp at which the message was sent will be saved in the message table as well.

Title	Data Type	Description
message_id	bigint	message id will be generated automatically and will act as a primary key of the table.
chat_id	bigint	chat id will act as a foreign key and will be used to distinguish that the message belongs to which chat.
message_content	text	content of the message
picture	varchar(255)	picture attached with the message
sent_by	bigint	sent_by will act as a foreign key from the user table and will be used to distinguish that the message was sent by which user.
sent_to	bigint	sent_to will act as a foreign key from the user table and will be used to distinguish that the message was sent to which user.

Chat Table

The chat table will contain all the information regarding the chat. user_id will act as the foreign key in this table. chat_id will be unique and act as the primary key for each user. The timestamp at which the chat was created will be saved in the chat table as well.

Title	Data Type	Description
chat_id	bigint	message id will be generated automatically and will act as a primary key of the table.
user_id	bigint	user id will act as a foreign key and will be used to distinguish that the chat belongs to which user.
created_at	timestamp	store the exact timestamp at which the chat was created

Event Table

The event table will store the information regarding the events posted by the admin. Event title, content, image, and dates will be stored in the table. The user_id will act as a foreign key in this table.

Title	Data Type	Description
event_id	bigint	event id will be generated automatically and will act as a primary key of the table.
user_id	bigint	event id will act as a foreign key and will be used to distinguish that the event was created by which user.
start_date	varchar(255)	store the date at which the event will start
end_date	varchar(255)	store the date at which the event will end
event_title	varchar(255)	title of the event
event_image	varchar(255)	image attached with the event post
event_content	mediumtext	content of the event

Instructor Review Table

The Instructor review table will store the information of reviews posted regarding the instructor on the app. Courseid will store the exact id of the course for which the review is posted. Rating will store a number between 1 and 5.

Title	Data Type	Description
review_id	bigint	review id will be generated automatically and will act as a primary key of the table.
user_id	bigint	user id will act as a foreign key and will be used to distinguish that the review was added by which user.
instructor_name	varchar(255)	the name of the instructor
course_id	varchar(255)	the id of the course
course_name	varchar(255)	the name of the course
rating	tinyint	contain the rating that the user came. Rating will be between 1 and 5.

review_content	longtext	content of the review
----------------	----------	-----------------------

Find Work Table

The Find work table will store the information regarding the find work posts on the app. Userid will act as the foreign key in this table. Work category, price, content and title will also be saved in this table.

Title	Data Type	Description
work_id	bigint	work id will be generated automatically and will act as a primary key of the table.
user_id	bigint	user id will act as a foreign key and will be used to distinguish that the find work post was added by which user.
work_title	varchar(255)	title of the post
work_category	varchar(255)	store the category to which the post belongs
work_price	int	compensation provided for completing the work
work_content	longtext	content of the work

Subspace Table

The Subspace table will store the information regarding all the subspaces present in the app. Subspaceid will be unique and act as a primary key of the table. User_id will act as a foreign key to distinguish who created the subspace.

Title	Data Type	Description
subspace_id	bigint	subspace id will be generated automatically and will act as a primary key of the table.
subspace_name	varchar(255)	name of the subspace
created_by	bigint	created_by will act as a foreign key from the user table and will be used to distinguish that the subspace was created by which user.

Joined Subspace Table

The joined subspace table will store the information regarding all the subspaces joined by a particular user. Both, subspace_id and user_id will act as a foreign key to distinguish which user has joined what subspace.

Title	Data Type	Description
subspace_id	bigint	subspace id will act as a foreign key and will be used to distinguish the particular subspace.
user_id	bigint	created_by will act as a foreign key from the user table and will be used to distinguish that the subspace is joined by which user.

4.4.2 Database

We will be using Firebase for our backend and database. Firebase is a google backend application development software. It will be a real-time database which does not require setting up a server. We chose Firebase for the following reasons:

- **Security:** Firebase provides access control, allows developers to write server-side rules for security. Security rules along with Firebases' integrated authentication make applications more secure.
- **Scalability:** Firebase allows developers to quickly setup highly scalable and flexible real time databases. It eliminates the middle synchronization between the layer application and backend database and provides direct access to Firebase SDK.
- **Time and Structure:** While building for a backend of an app, a server, hosting, database and other backend supporting services are required. This often leads to increased integration time and development efforts. With Firebase, all prerequisites of backend development are integrated with it, thus less effort will be required.
- **Performance:** Firebase is cloud service and works better than other servers and backend services providers. Fetching and displaying the data from Firebase is much faster than other database management systems.

4.5 External Interface Requirements

4.5.1 User Interfaces

Back Space will be an interactive mobile application which can only be used in android smartphones with an active internet connection. The interface will be easy to use, and all the text will be displayed in English. There will be two main screens:

- 1) Newsfeed Page
- 2) Admin Page

All of these screens will consist of proper icons with texts where necessary so that any user could easily navigate. Newsfeed page will consist of high-definition images with appropriate captions added by other users. The color scheme will be mainly black, gray and white, which will be used in all screens. Moreover, the interface will be designed using common UI elements such as buttons, pagination and text fields etc.

4.5.2 Hardware Interfaces

The application is a mobile based application hence the user should have access to a mobile phone to run a Back Space. Some of the basic requirements are listed below:

- The supported device should also have a touch screen so the user is able to enter their input to the system.
- The supported device should be able to connect to a stable network.
- The device must be an android compatible device, and must support android version 5 or higher, API level greater than 21 and version code Lollipop.

The mobile application is to be hosted on a server. Our app will be communicating with firebase to fetch data and store details depending on the user operation.

5 User Interface Design

5.1 Description of the user interface



Flutter is a free and open-source mobile UI framework created by Google for mobile app development. It uses Dart as the programming language which is language for client development such as web or mobile apps.

We are using flutter for our front end technology for the following reasons:

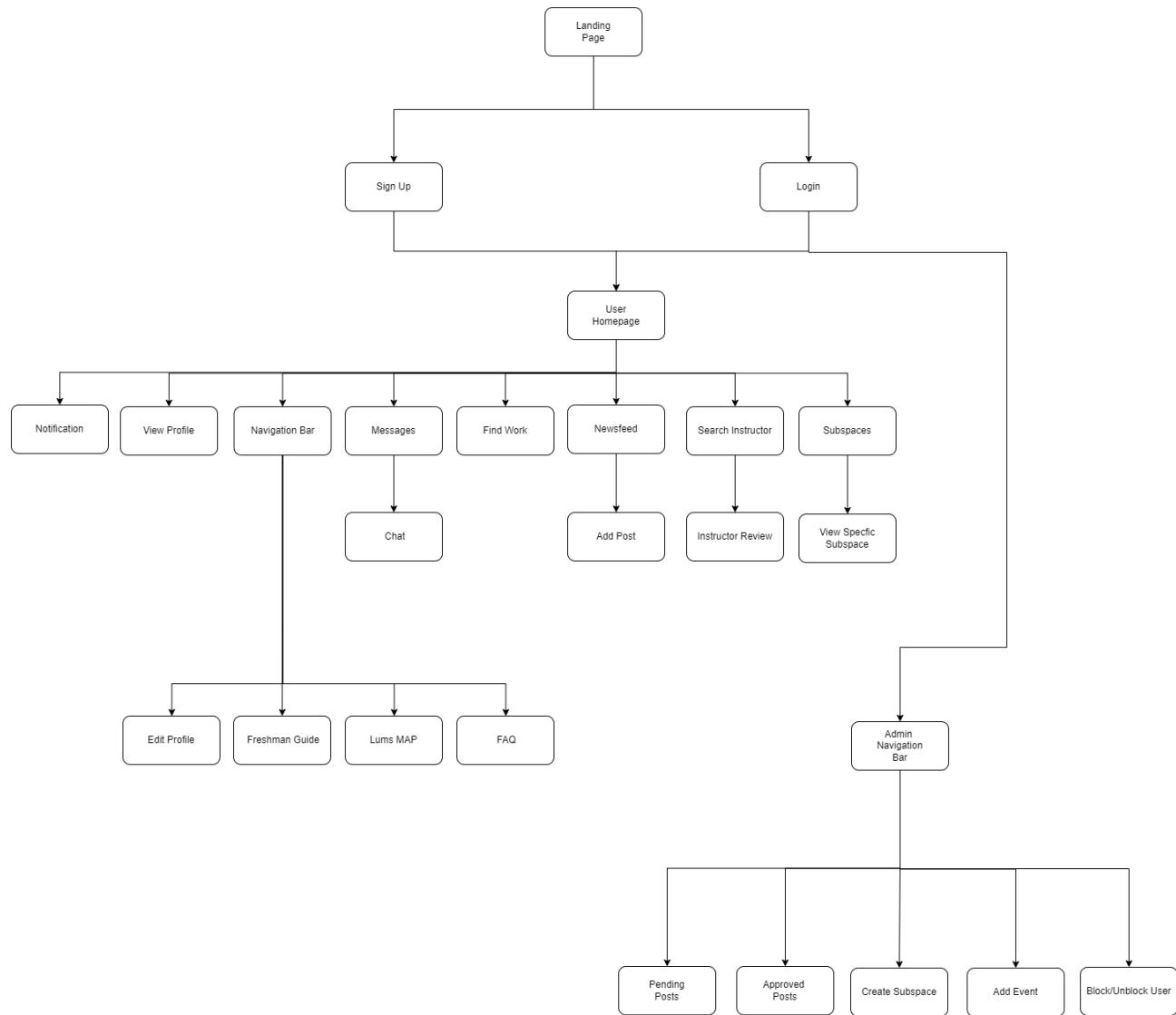
1. It consists of widgets and comes with built-in styles thus we don't have to use another language for our front-end.
2. It allows fast testing of mobile applications with hot-reload and all changes are immediately displayed in real time.
3. It also provides a user with smooth app experience and quick loading since flutter consists of a Skia graphics library which is fast as compared to other libraries.

Visual Studio Code:

We will be using visual studio code as our editor. The reason for choosing vs code is that it supports multiple languages and provides other features such as syntax highlighting, auto-indentation, and bracket matching. It also allows us to customize features and install any third party extension required for development.

5.2 Information architecture

The attached figure shows the information architecture presented visually. The screens and their navigation to other screens is shown. The architecture is broad and shallow, allowing users to navigate easily and quickly between all the different screens. The use of components such as bottom navigation and navigation drawer allow the architecture to be broad. Further description of the functionality and navigation routes from each screen are described in section 5.3



5.3 Screens

We have divided our screens according to the functionalities of the users, including students and administrators and the screens with common functionalities to both.

1- Common Interface

- 1) Landing
- 2) Login
- 10) Search

2- Students Side Interface

- 3) Sign up
- 4) Verification Code
- 5) Verification Success
- 6) Bottom Navigation
- 7) Top Bar
- 8) Navigation Drawer For Students
- 9) NewsFeed
- 11) Comment
- 12) View Profile
- 13) Instructor Profile
- 14) My Subspaces
- 15) Subspace Chat
- 16) Messages
- 17) Messages
- 18) Notifications
- 19) Find Work
- 20) Edit Profile
- 21) FAQs
- 22) Lums Map
- 23) Freshman Guide

3- Administrators Side Interface

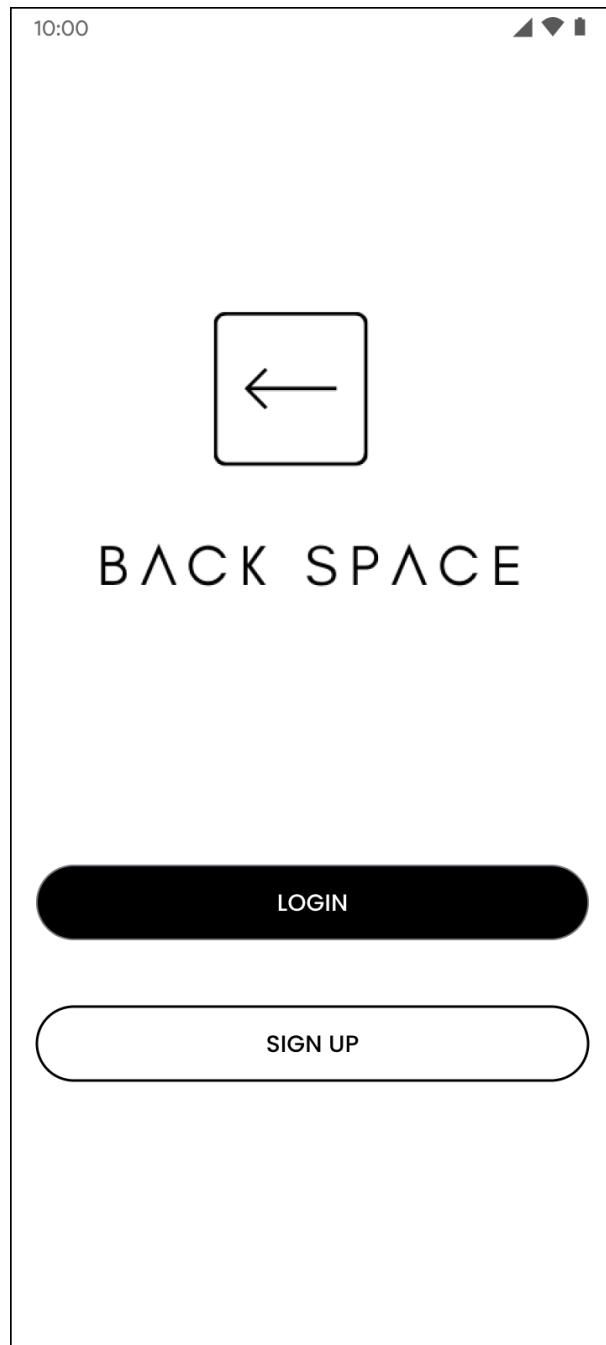
- 24) Pending Posts
- 25) Approve Posts
- 26) Add Events
- 27) Add Subspace
- 28) Block Users
- 29) Navigation Drawer

Note: We used Figma to design our screens.

1) Landing

Use Case: None

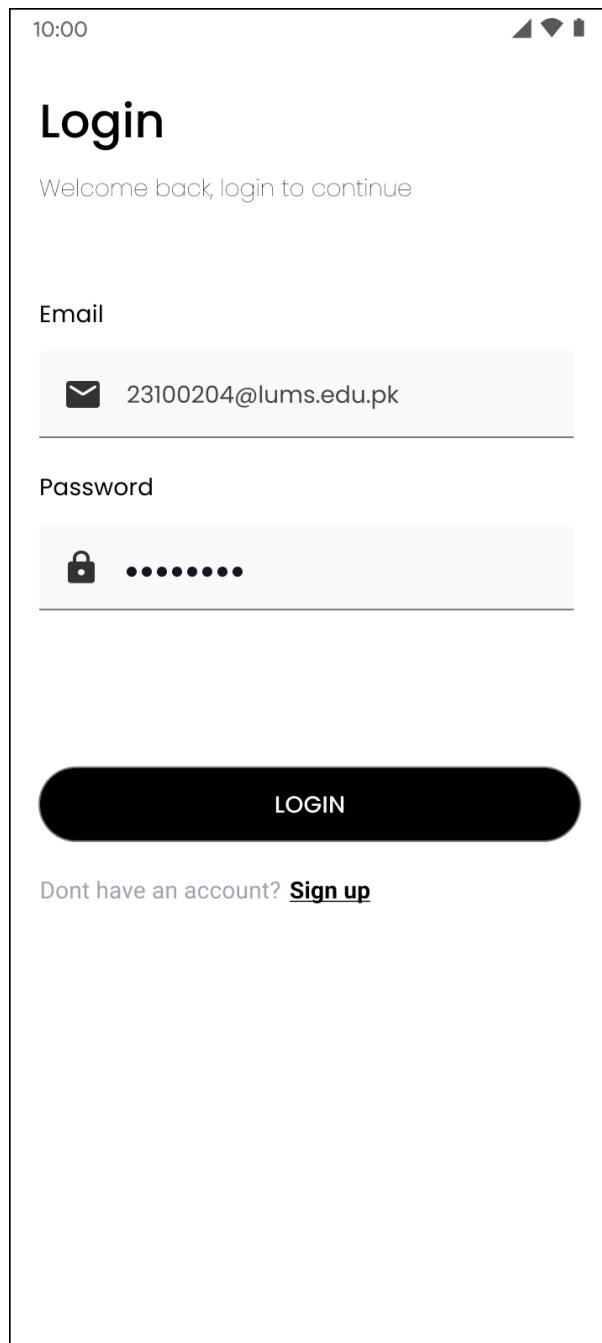
Description: The default page lands on when the app is first opened. It allows the user to either sign up if they do not have an account by clicking on the Sign-Up button or Login to the app if they have already created an account by pressing the Login Button.



2) Login

Use Case: Login

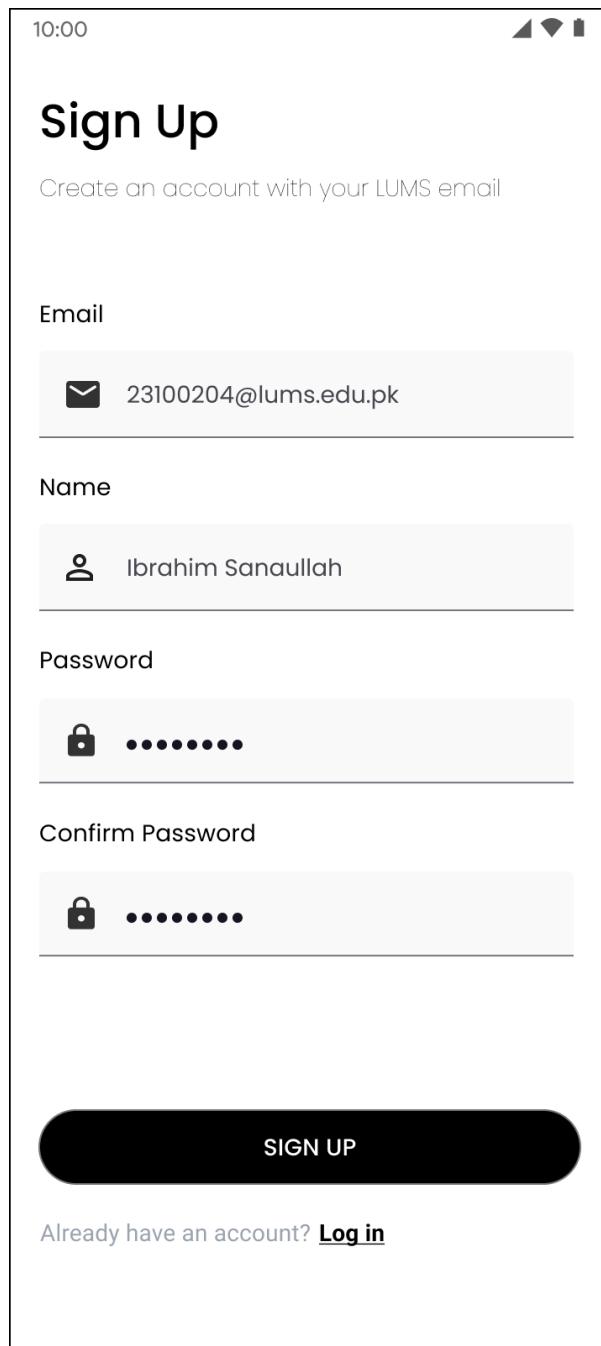
Description: The user needs to enter their email from which they have already created their account and enter their respective password. If the password of the corresponding email is correct the user will be logged in, otherwise an error will display. If the user has not created an account, they can click on the Sign up button, so they are directed to the sign-up page



3) Sign up

Use Case: Sign Up

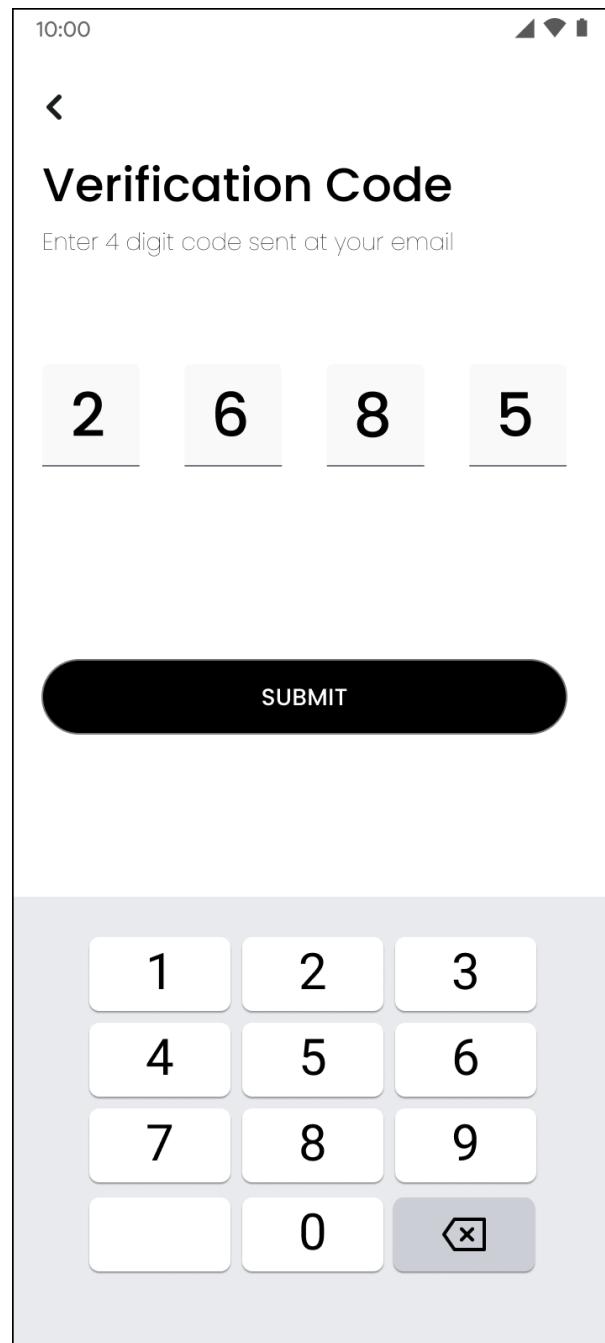
Description: The user needs to enter their credentials into the given boxes then click on the Sign-Up button. For email they need to enter their LUMS email otherwise an error will display. They also need to re-enter their typed password and an error will be shown if the entries typed under “Password” and “repeat Password” are different. If no errors are shown they will be redirected to the verification code page. If the user already has an account, they can click on the Login button to go to the Login page.



4) Verification Code

Case: None

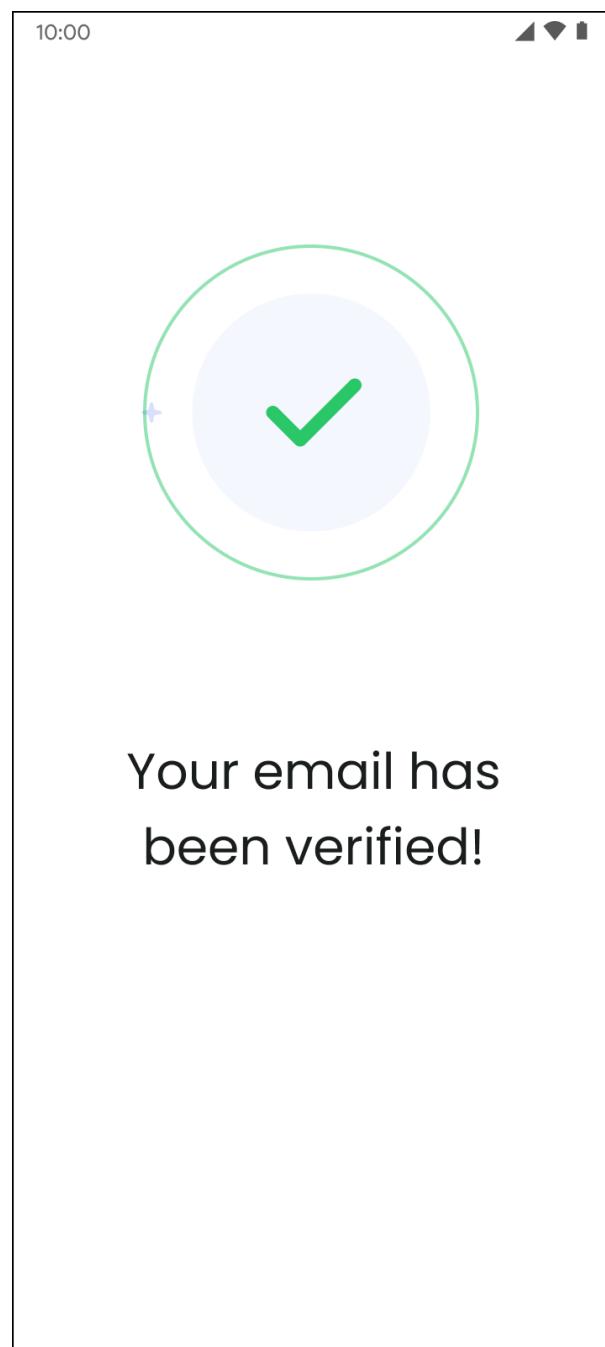
Description: After users enter their email to create an account and click on Sign up, they are sent a 4 digit verification code to ensure they are the owner of the email entered. Users need to look at the code sent at their email and enter it into the respective fields. If the code entered is identical to the one sent at their email they are shown a verification success page otherwise they are shown an error. The user can also click on the back button to go back to the Sign up page in case they might have entered an incorrect email.



5) Verification Success

Use Case: None

Description: After the user enters the correct verification code they are sent to this page to provide feedback of their action (rephrase this). The user does not click on anything on this page, they are automatically directed into the main part of the application after a few seconds of displaying this screen.



6) Bottom Navigation

Use Case: None

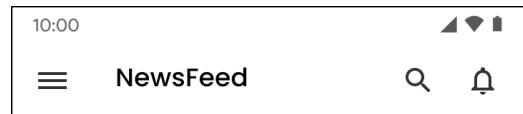
Description: Bottom Navigation bar is used in the user side to navigate through 5 major parts of the Application and will be visible from all pages in the user side. Clicking on the homepage will take you to the Newsfeed, Clicking on the instructors button will take you to the Instructor review section. The Work button will navigate you to the Find work section. The Subspace section will take you to your joined Subspaces. The messages button will take you to your chats with other users. The section that is currently active is kept in normal colors whereas the opacity of other sections have been reduced to show the current selection. The following image just shows the bottom navigation bar which is a part and incorporated in our other screen and is written about separately to avoid redundancy when describing other screens.



7) Top Bar

Use Case: None

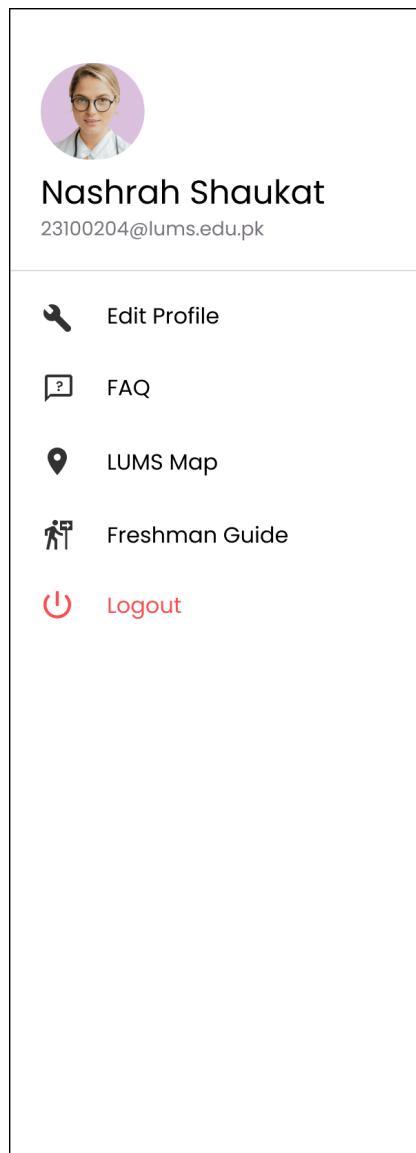
Description: Present on all the main screens in the client Side. Contains the Navigation drawer icon which will allow the user to access it. The text in the middle provides the title of the screen you are currently at. Clicking on the bell icon will take you to the Notification screen. Some pages have an option of a search icon which will allow the user to search locally within the section. The following image just shows the top navigation bar which is a part and incorporated in our other screen and is written about separately to avoid redundancy when describing other screens.



8) Navigation Drawer For Students

Use Case: None

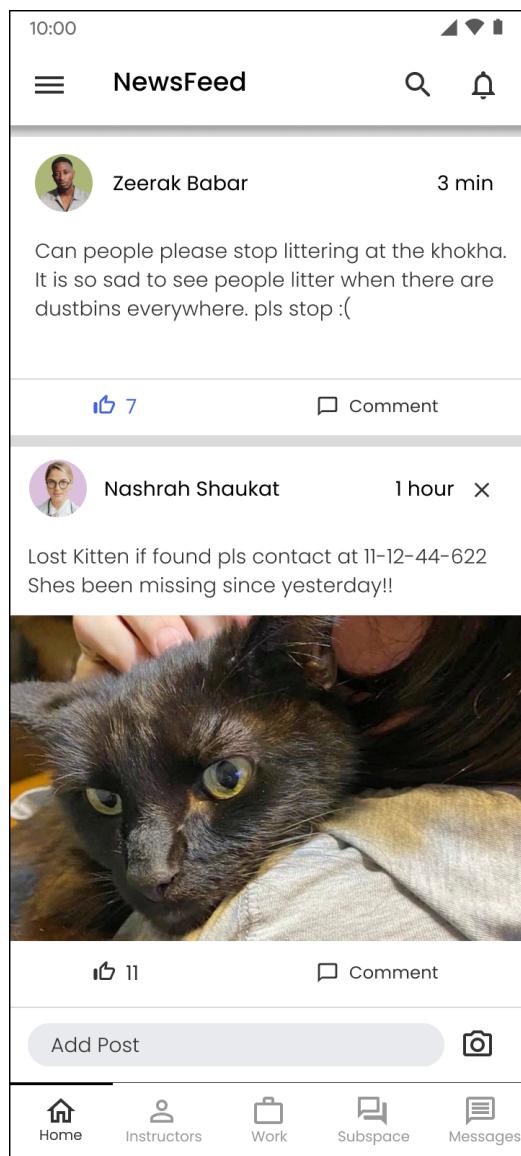
Description: The side Navigation drawer can be used to navigate to different sections of the program. Namely “Edit Profile”, “Lums Map”, “View FAQ” and “Freshman Guide” by clicking each of their respective buttons. The Nav bar can also be used to Logout after which the user will be redirected to the Login Page.



9) Newsfeed

Use Case: View NewsFeed, Like Post, Add to NewsFeed, Search NewsFeed, Delete Post

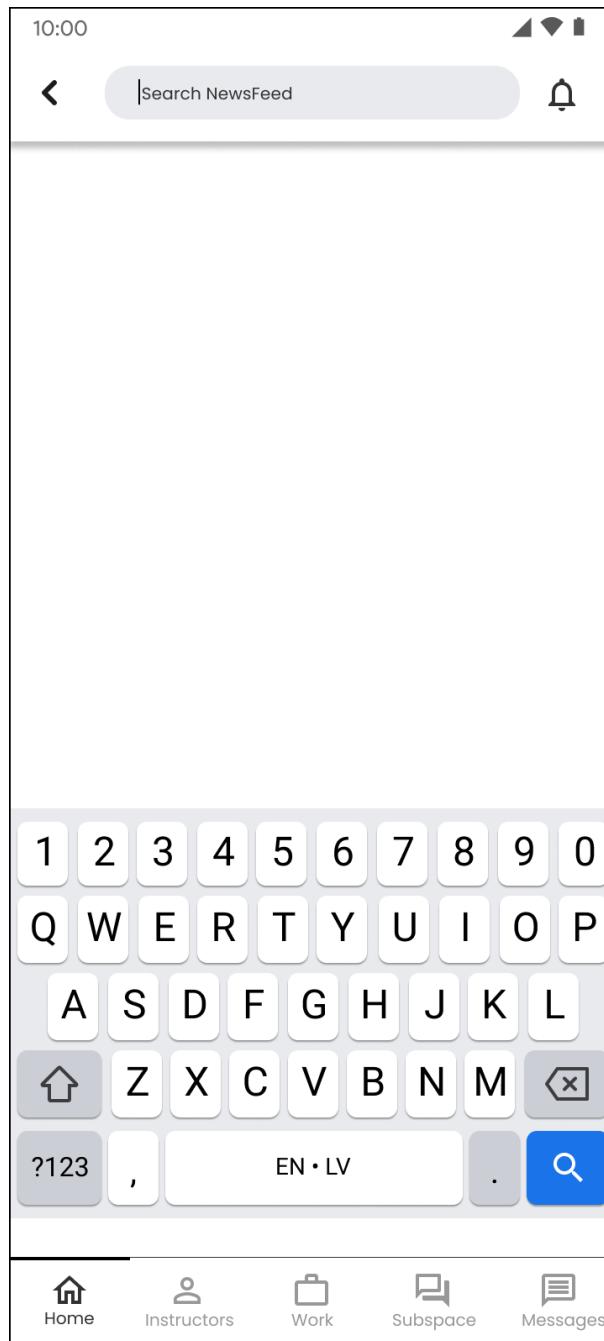
Description: Users will be able to scroll through the newsfeed and view posts from other users. They can also add their post by typing it in the text field label "Add Post" after which it is sent to the Admin for approval after which is shown in the Newsfeed. The name and picture of the user who wrote the post is shown, clicking on either of them will open the page containing their profile. The user may also choose to like the Post by clicking on the thumbs up icon after which it will turn blue to give feedback to the user and the count of likes on the post will increase. The users will also be able to click the comment button which will take them to a screen which displays the comments under that post.



10) Search

Use Case: None

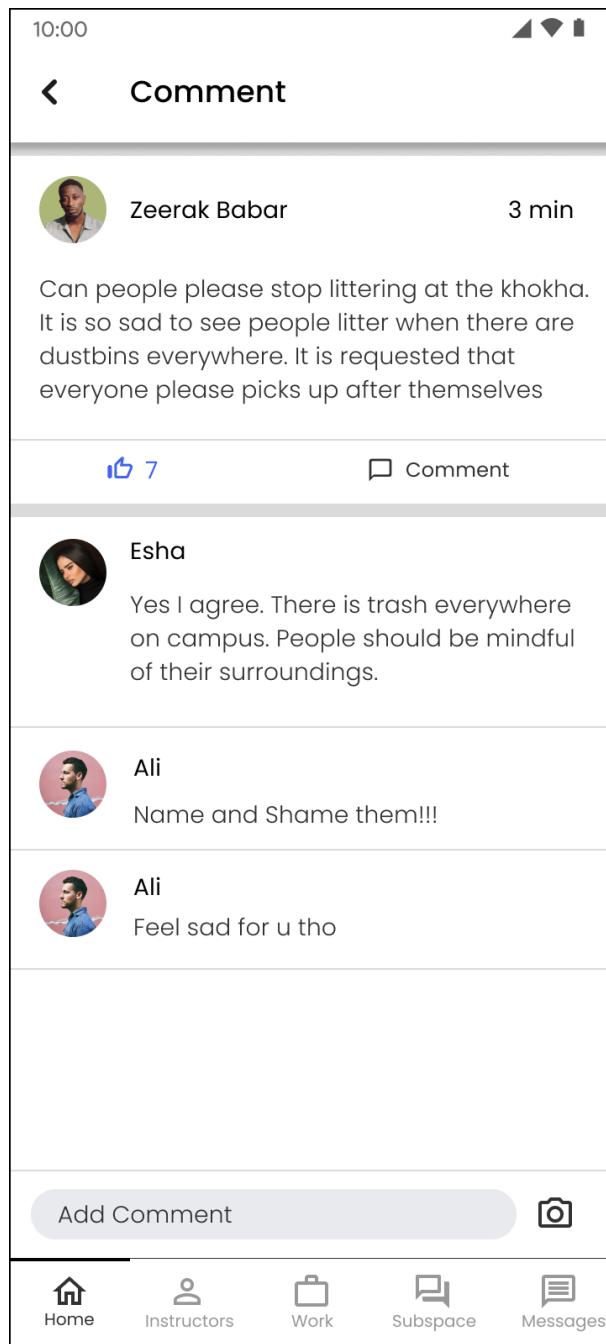
Description: Screens containing the search icon in the top bar will be able to click on the icon and open this screen. This screen will allow users to type in keywords and search for items that contain those keywords. All items containing those keywords will be displayed, if no such item exists “No results found” text will be displayed.



11) Comment

Use Case: Comment Post

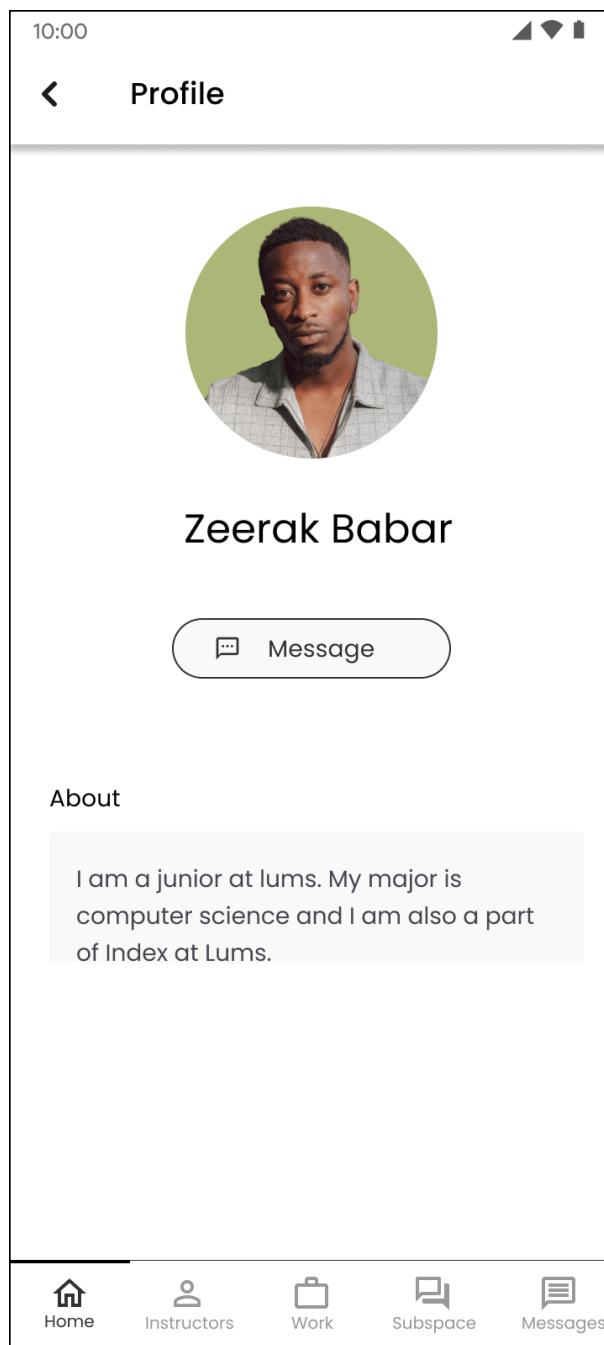
Description: Users will be able to view the comments under that post and scroll through them. Users can press the back button in the top bar to go back to the NewsFeed. Users can also comment under the post by typing their comment in the text field labeled “Add comment”.



12) View Profile

Use Case: View Profile

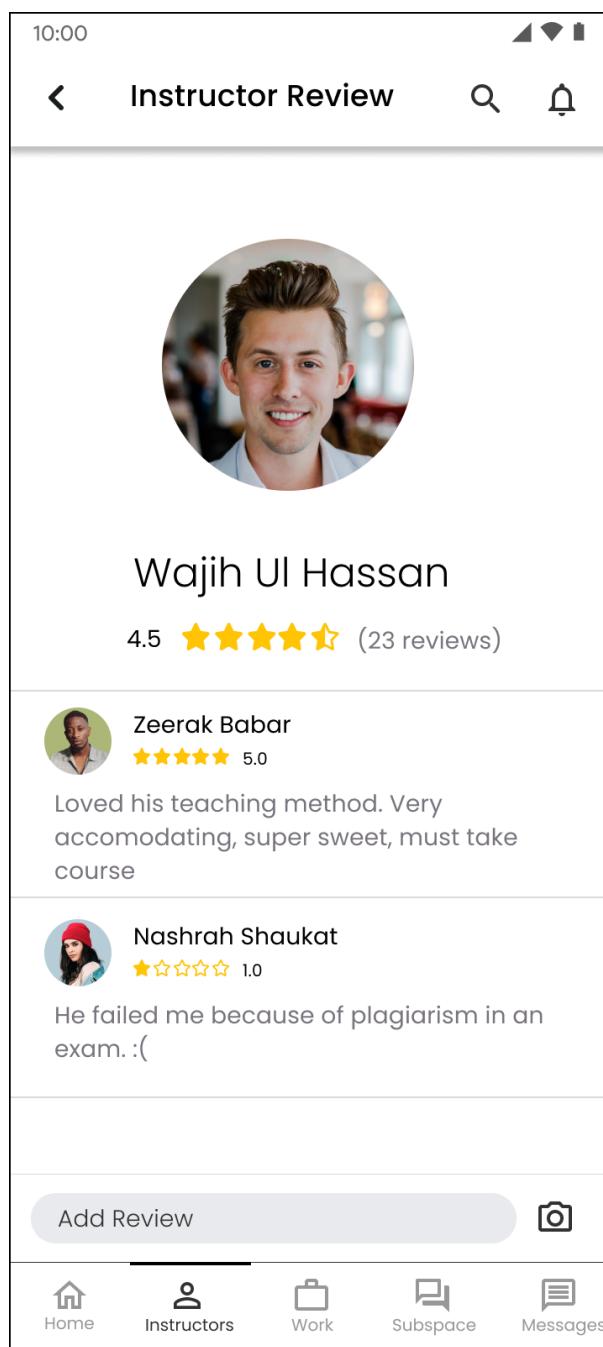
Description: Users will be to view the name and picture of the respective person, read the about section they have written about themselves. Users can click on the message button to msg the specific user. Clicking on the back icon will take the user back to the previous screen they were present at.



13) Instructor Profile

Use Case: View Instructor Review, Add Instructor Review, Search Instructor Reviews

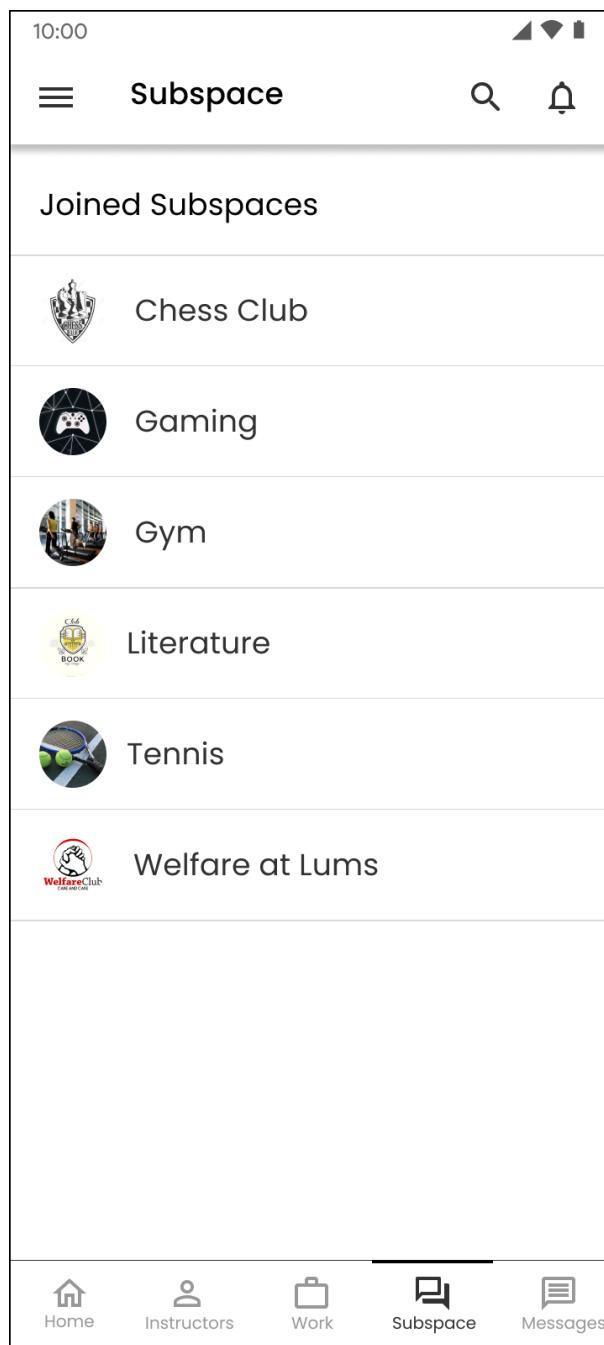
Description: Users will be able to view a specific instructor's review. The screen would display the reviews posted by other users along with the rating of the instructor. The user can scroll through the screen to read all the views. Users would also be able to post his/her own review of the instructor by typing in the 'add review'. Once the user does that, the keyboard would be available. Once the user has written the review they would also be able to rate the instructor and then post the review. The screen provides a comprehensive design so all the reviews about a specific instructor are in one place for easy viewing. The user can also add reviews on the same screen with easy navigation which would then be available under the same instructors page.



14) My Subspaces

Use case: Search Joined Subspace

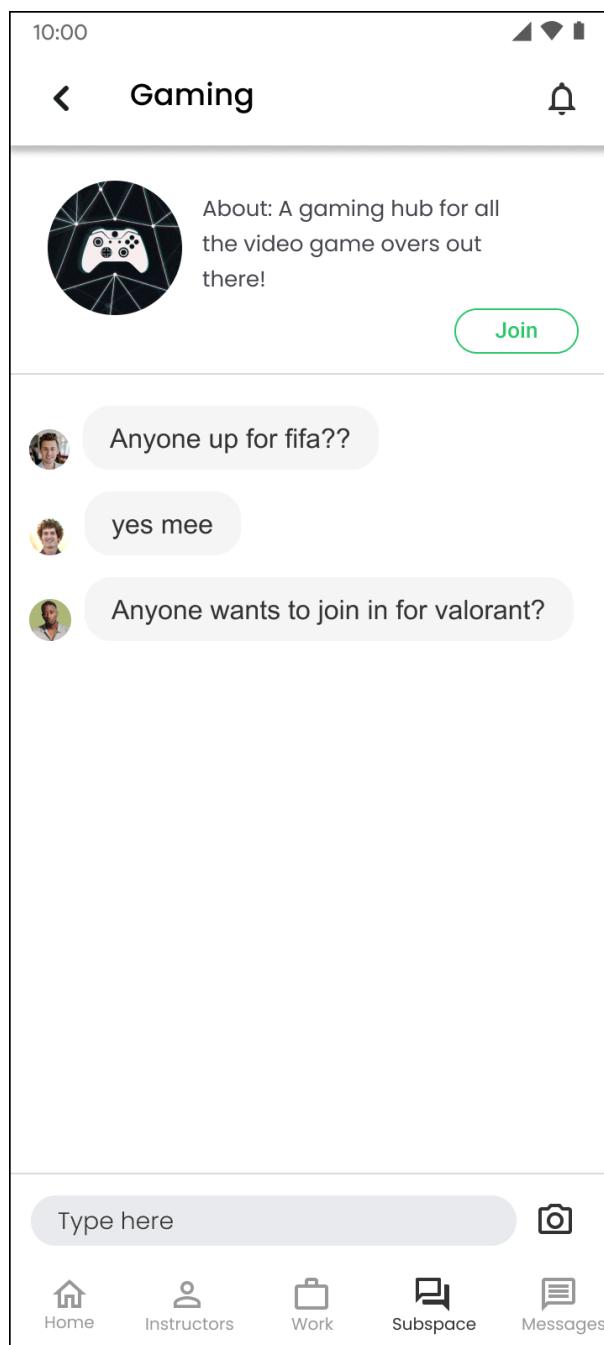
The screen would display all the subspaces joined by the user. The user can click on a specific subspace to access the group chat for that subspace. The user can access the screen from the bottom navigation bar for easier access. It would also be efficient to have all the subspaces the user has joined in one place.



15) Subspace Chat

Use Case: Send chat, Join Subspace

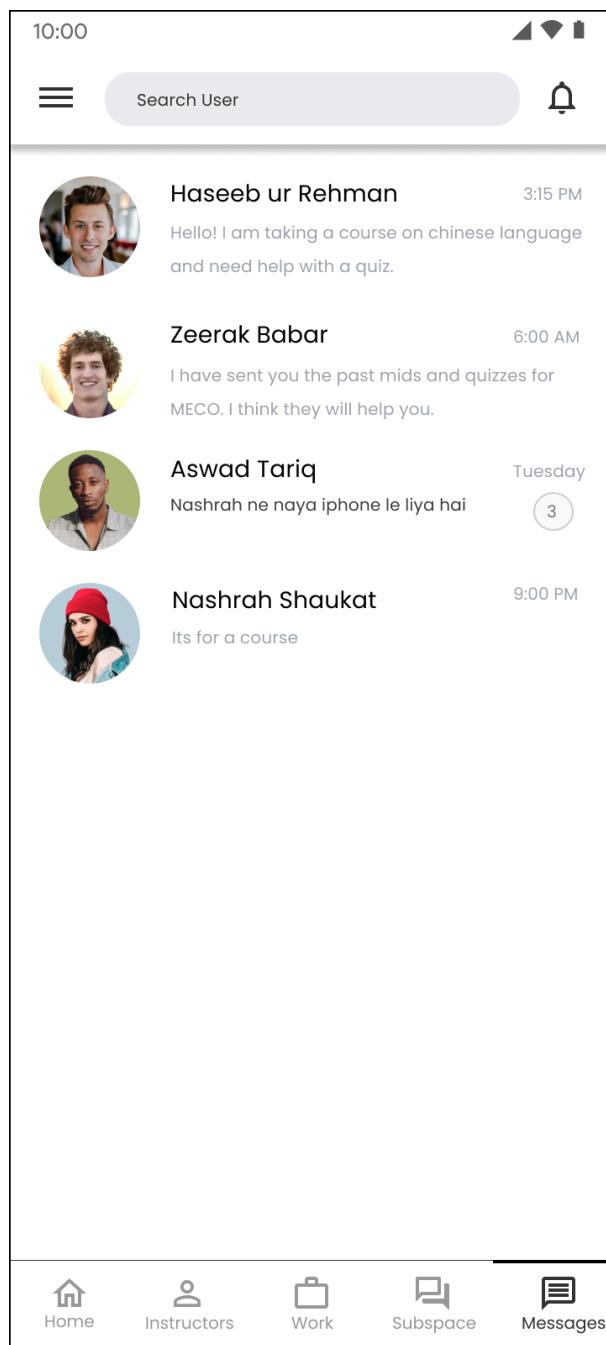
The screen displays the group chat for the chosen subspace. The display shows the messages exchanged among different users. The screen also shows a brief subspace screen added by the moderator. The user can also send a message in the group chat by typing in the chat box. Moreover the user can also attach pictures to send in the group chat. Since the subspace would be a community space where everyone can interact, group chat is an efficient means of communication.



16) Messages

Use case: View all chats, Search Users

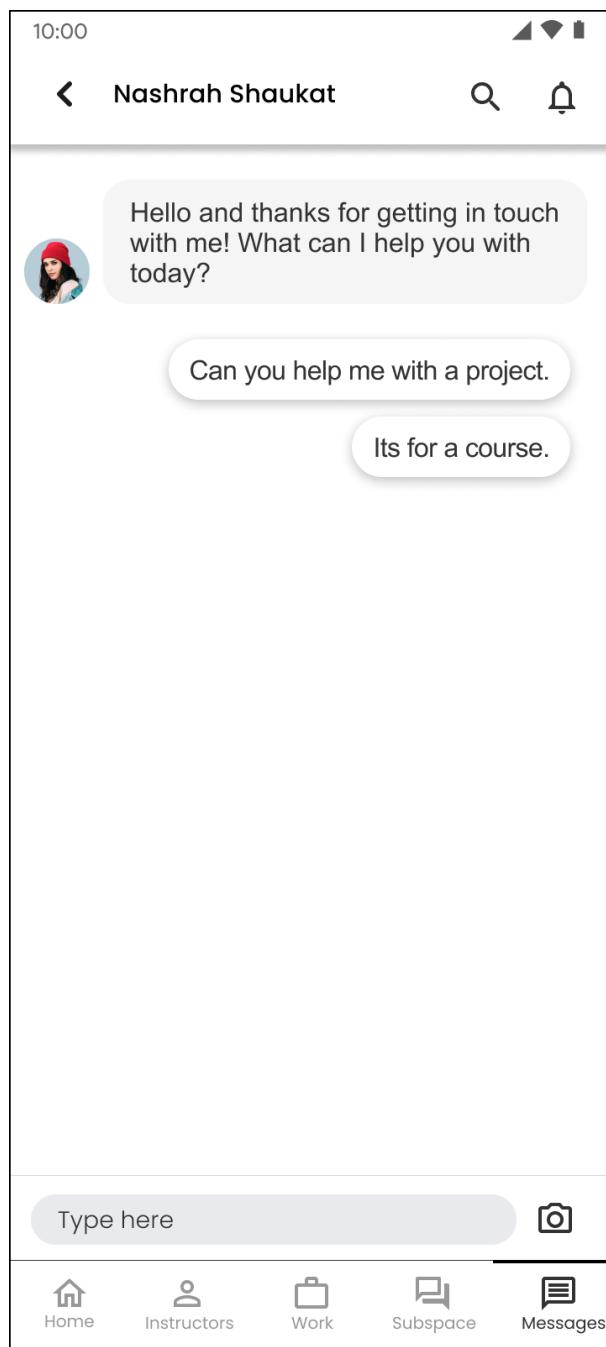
With the messages tab on the bottom navigation user can access the messages screen. The screen would display the chat between the user and other users. The chats would only be between two users, unlike a subspace group chat. Users can click on any specific chat to open that chat. The relevant details about the messages exchanged is also available.



17) Messages

Use Case: View a specific chat, Send chat

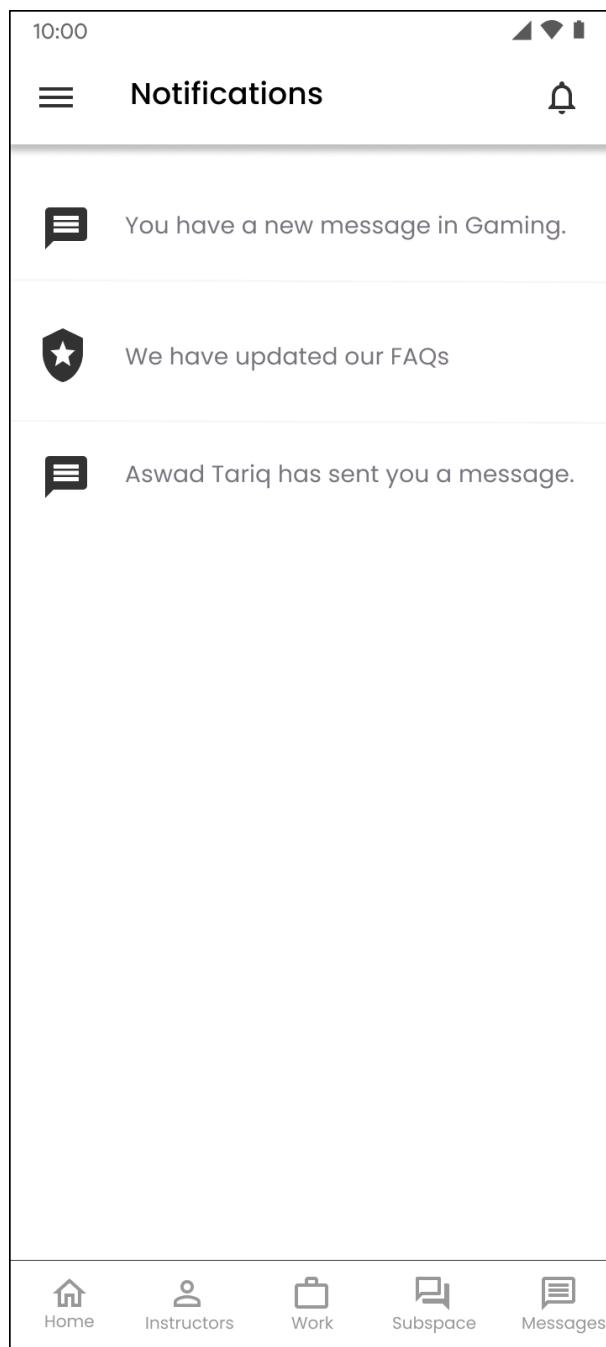
After the user has opened a specific chat, the next screen would display all the messages exchanged between the user. The user can send a message by typing in the chat and also attach a picture to send as a message. The screen would also display the name and display picture of the user, with whom the messages are being exchanged.



18) Notifications

Use Case: View all notifications, View a specific notification

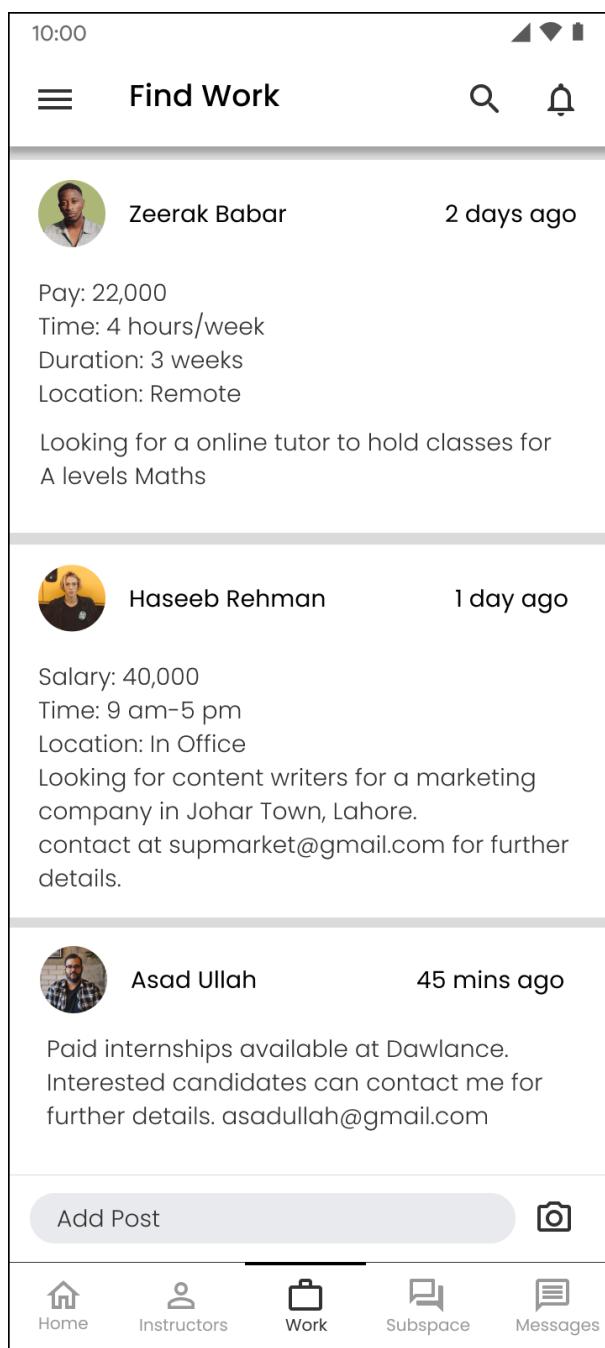
The user can click on the bell icon on the top right of all the screens to view the notifications screen. The screen would display all the current notifications, including new messages from a user or in a group chat, when someone likes or comments on the users post etc. The user can click on a notification to go to the specific message or post.



19) Find Work

Use Case: Search Find Work

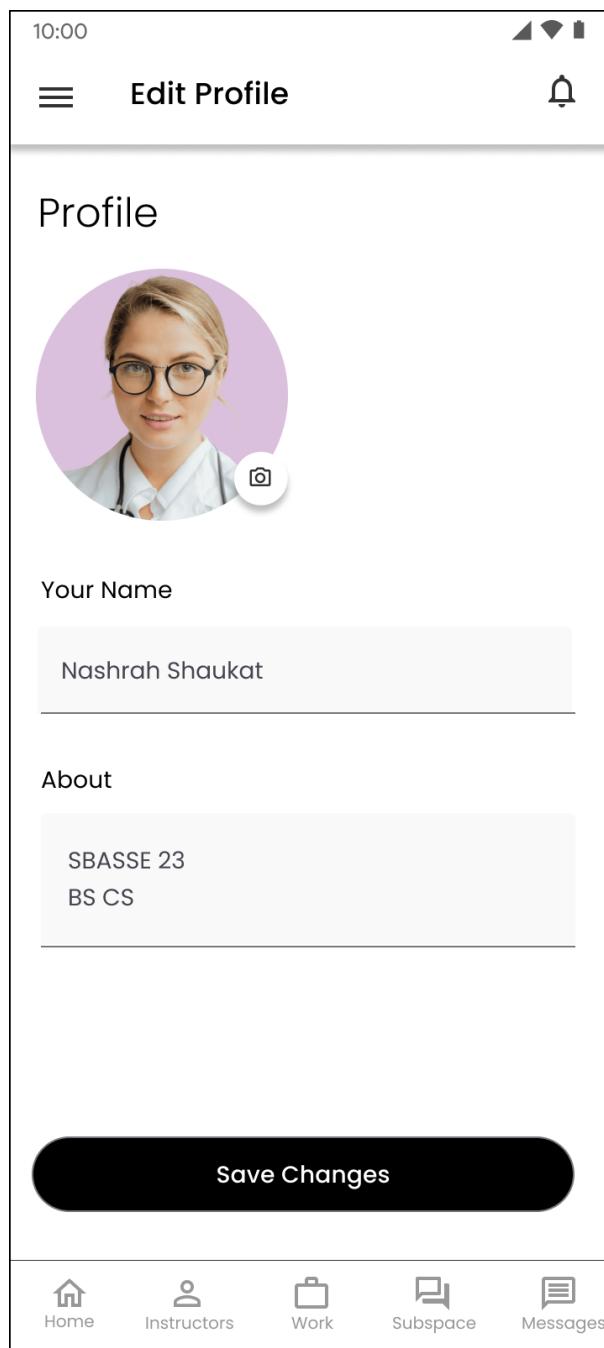
The user can click on the button ‘work’ on the bottom navigation to access the find work screen. The user can search for any work by typing in the search bar and typing relevant keywords. All the posts containing relevant keywords will be displayed. The user can scroll through the posts. The user can also enter a new post by writing in the text bar on the bottom of the screen.



20) Edit Profile

Use Case: Update Credentials

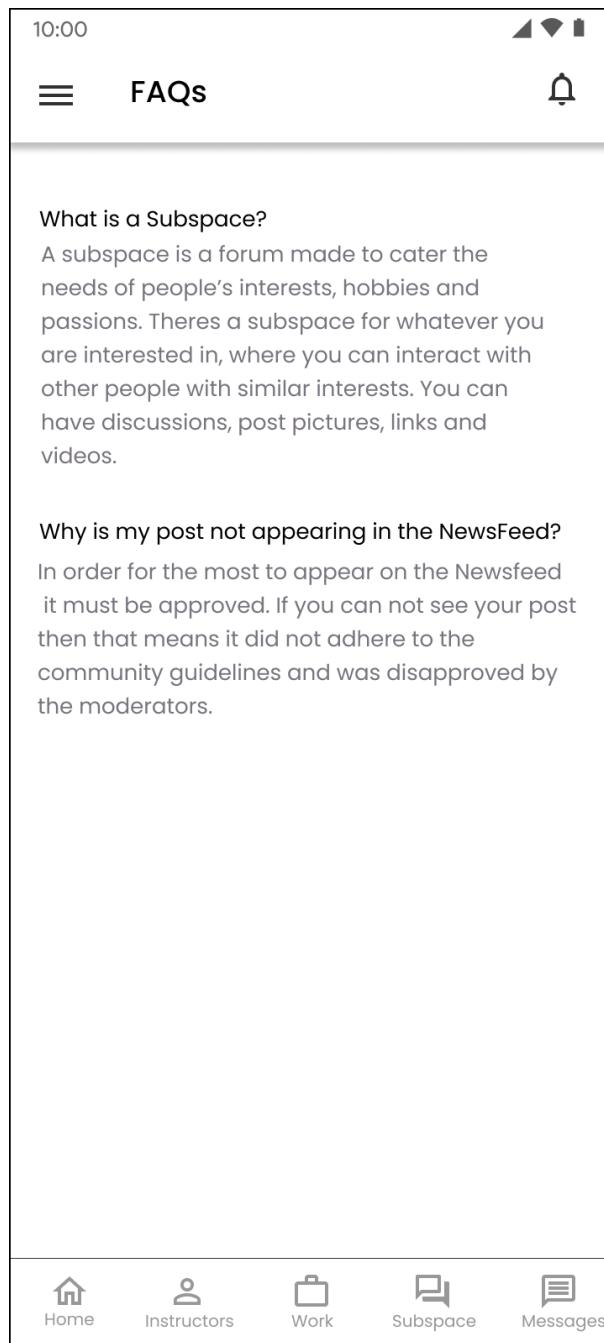
The user can edit the profile by accessing the ‘edit profile’ option from the side drawer. The user would be able to upload a picture. The user can also change their name. The user can also enter details about themselves in the ‘about’ section. The user would then click on the ‘Save changes’ button at the bottom of the screen to save and update the changes.



21) FAQs

Use Case: View FAQs

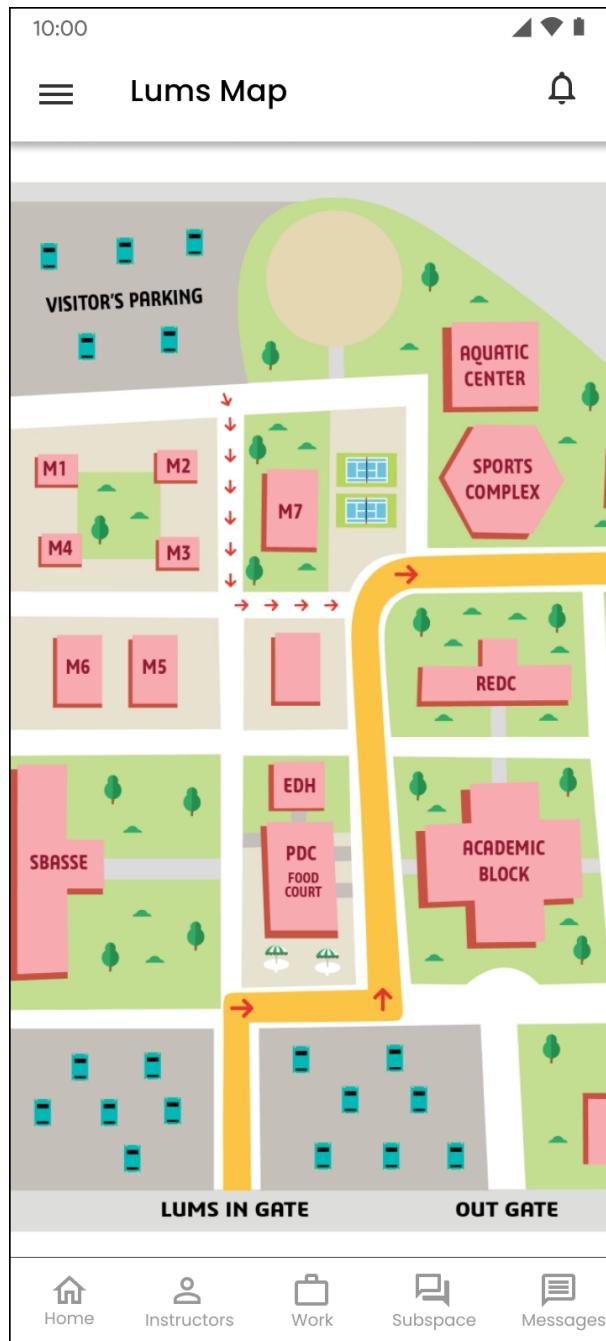
The user can access the FAQ screen from the side drawer. The FAQ screen would display the frequently asked questions about the application and its workings and their answers. The users can click the home button on bottom navigation to go back to the news feed or use other icons to switch to other screens.



22) Lums Map

Use Case: View Lums Map

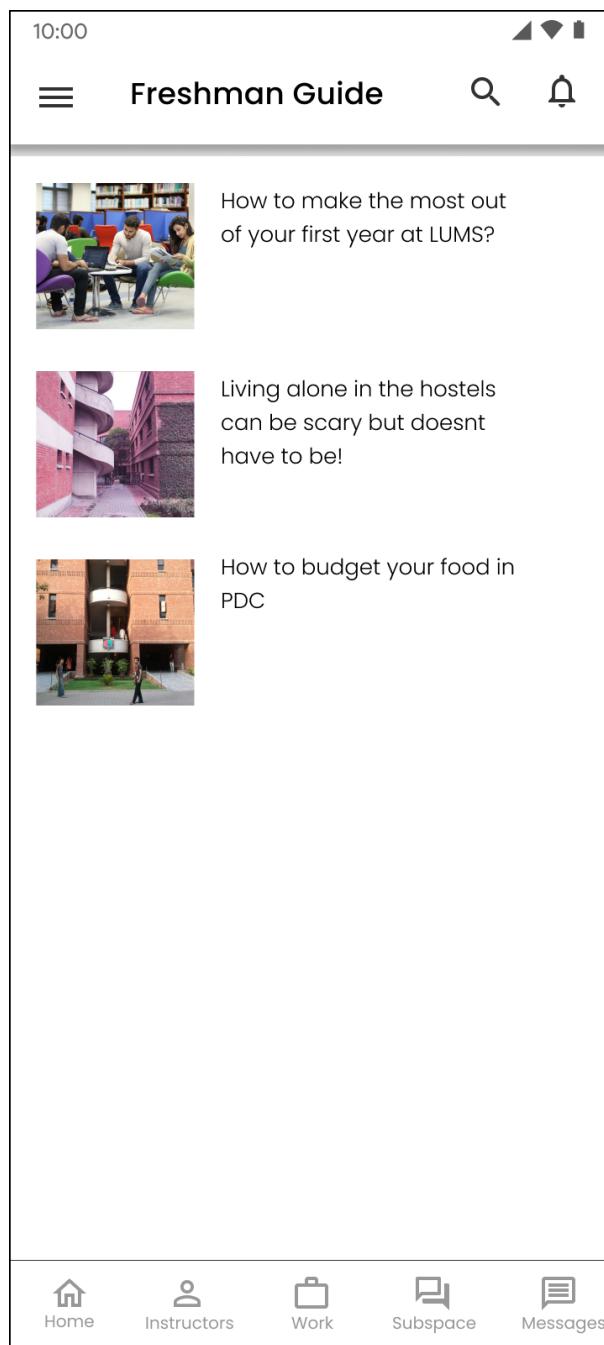
The user can access the Lums map from the side drawer. The screen would display the map of the lums campus. The user would be able to zoom in and zoom out by pinching in or pinching out their device's screen. The user can also move the lums map once zoomed in.



23) Freshman Guide

Use Case: View Freshman Guide

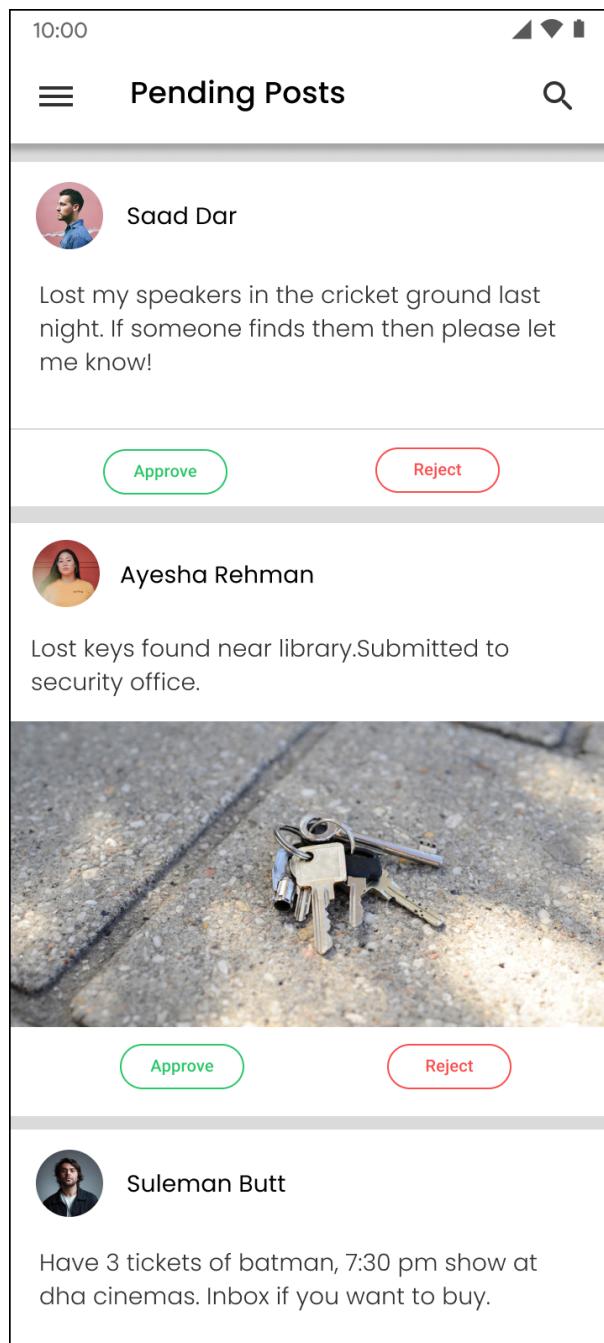
The user can access the freshman guide from the side drawer. The screen would display different articles to guide the freshman at LUMS. The user can click on a specific article to read the whole article.



24) Pending Posts

Use Case: Approve posts, Deny posts

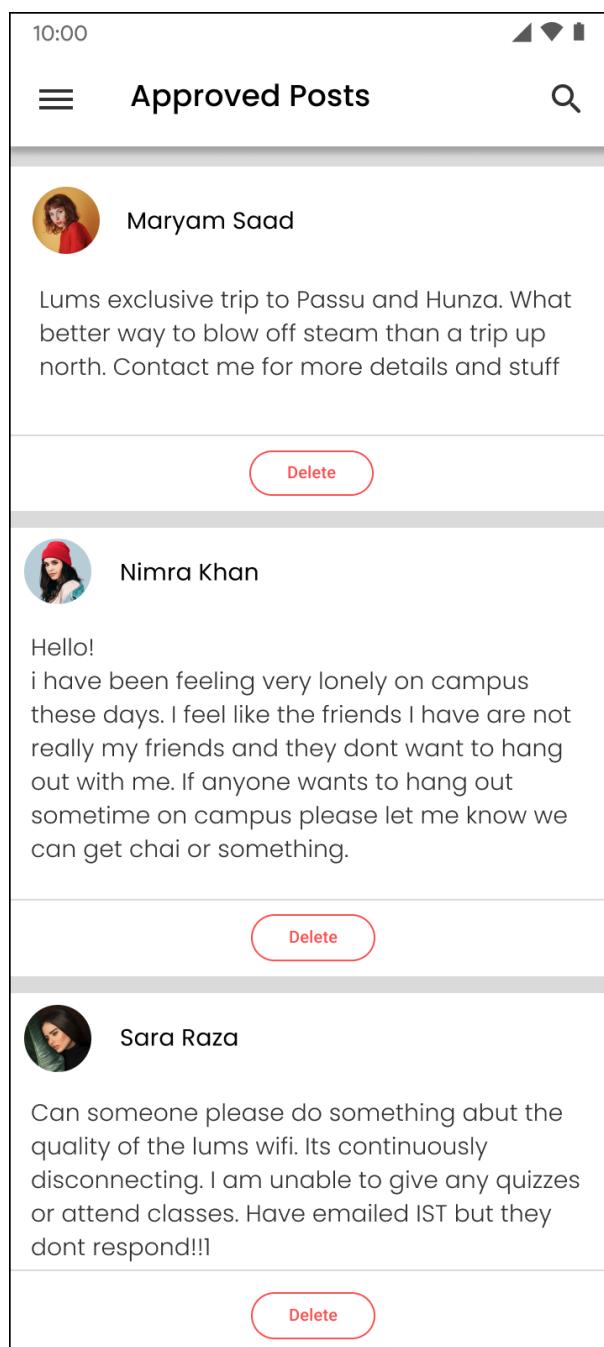
The pending posts screen would be the landing screen when an administrator signs in or log in. The screen would display all the current pending posts awaiting approval. The administrator would be able to either approve or reject the post using the ‘approve’ or ‘reject’ button.



25) Approved Posts

Use Case: Remove posts, Search Approved Posts

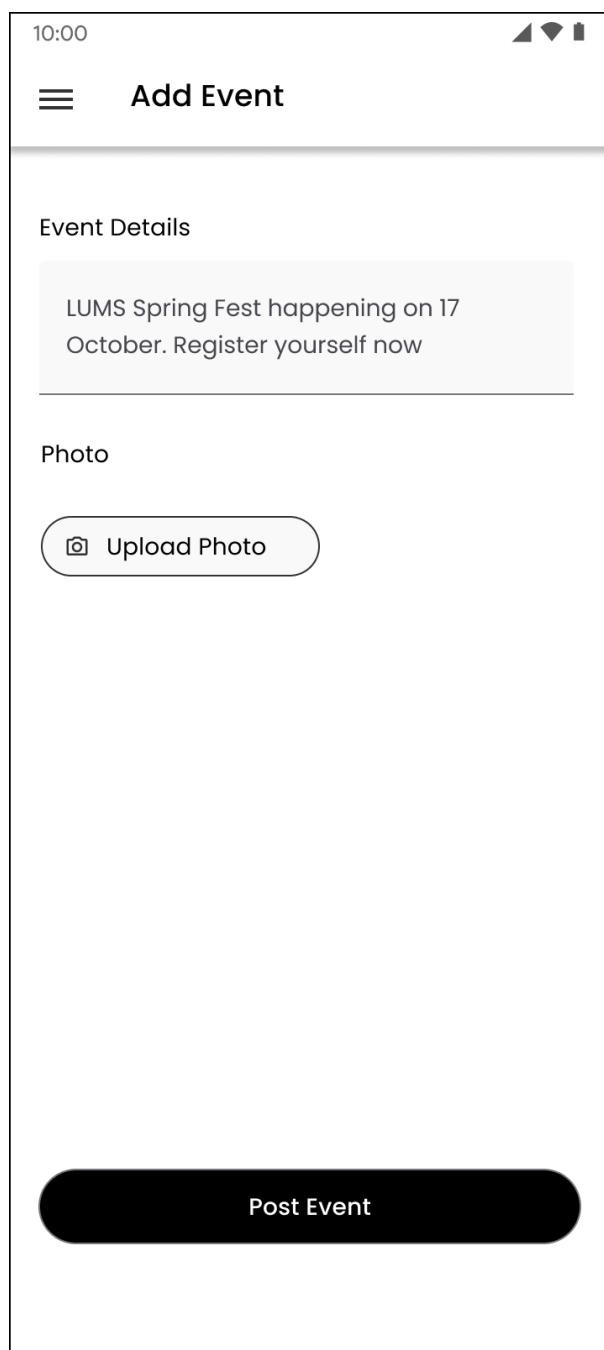
The approved post screen would display all the posts approved by any of the administrators. This is one way for the administrators to keep track of the posts which the students can view on their newsfeed. The admin can also delete already approved posts by tapping on the ‘delete’ button.



26) Add Events

Use Case: Add events

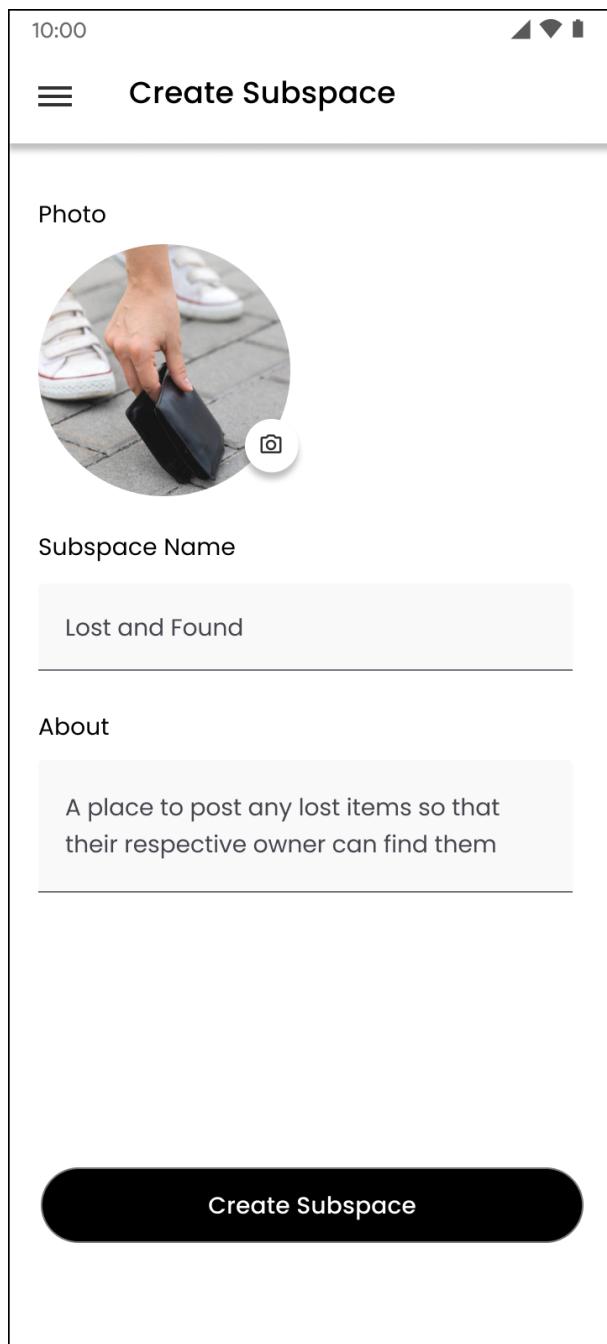
The add events page enables the administrators to make new events. Administrators can enter all the relevant event details in the text box under ‘event details’ and then upload a picture as well if required. The user can then tap on the ‘post event’ button to post it on the events tab.



27) Create Subspace

Use Case: Create subspace

Admin can navigate to the create subspace tab to make a new subspace. The screen would have the option to insert a picture for the subspace, enter the subspace name in the text field and to write a

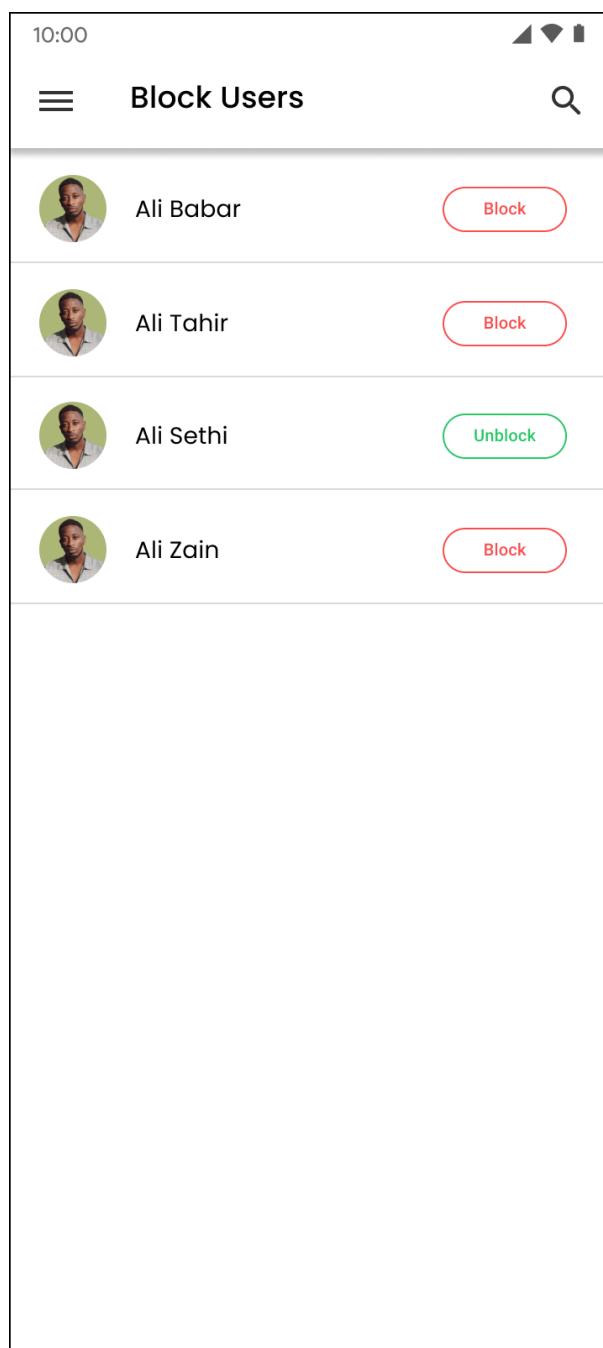


brief description about the subspace in the text field. The user can then click on the ‘create subspace’ button to post a new subspace.

28) Block Users

Use Case: Block user, Unblock user

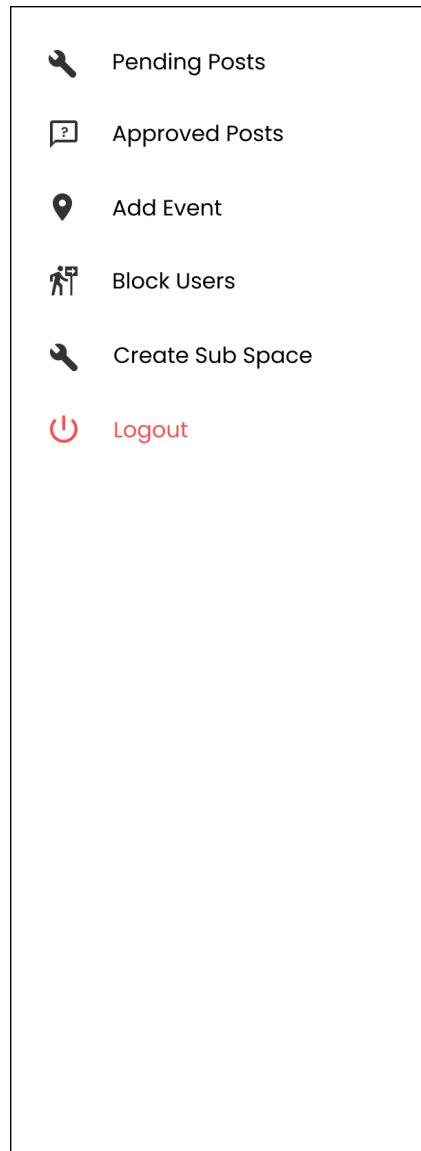
The screen allows the admin to block users by tapping on the ‘block’ button or unblock users by tapping on the ‘unblock’ button. The admin can also search the users from the search bar on the top to block or unblock.



29) Navigation Drawer for Administrators

Use Case: None

The side navigation drawer can be used by the administrators to access other screens. The ‘pending posts’ screen would navigate to the pending posts screen or homepage for the administrator. The ‘approved posts’ button would take the user to the approved posts screen, ‘Add events’ would take to the add events screen, ‘block users’ would take to the blocked users screen, ‘create subspace’ to the create subspace screen. The user can log out by clicking on the ‘log out’ button.



5.4 User interface design rules

The interface of the Back Space has been designed keeping both the android application conventions and design heuristics in mind. We have used the design heuristic of Jakob Nierlseb to further refine our application.

5.4.1 Website Conventions:

Layout

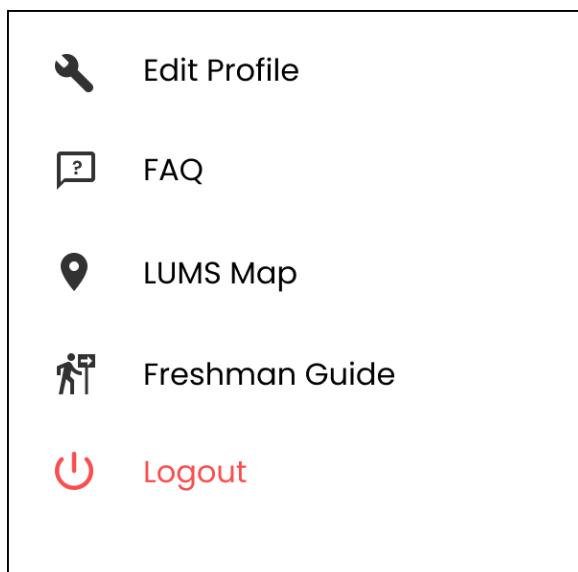
Placements of elements have been done according to their respective conventions. App bar is at the top of the screen which contains the navigation drawer. The bottom navigation has been placed at the bottom of the screen

Colors

3 main colors have been used in this design namely Black White and Gray to keep it coherent and consistent

Icons

We have used icons of material design in order to keep it consistent and also recognizable by people. Where needed icons are accompanied by text if the meaning of the icon alone might not be clear



Fonts

The same font has been used across the entire design to keep consistency. Different sizes of font have been used for titles, subtitles and text to highlight the hierarchy. Text has been in both black and gray for contrast purposes

5.4.2 Design Heuristics

Visibility of system status: Users are shown what is going on in the program by giving visual feedback. Feedback may be provided in many forms such as a verification success screen, color change or opacity change. A few examples of feedback would be the color of the like icon changing when a post is liked:



Zeerak Babar

3 min



Zeerak Babar

3 min

Can people please stop littering at the khokha. It is so sad to see people litter when there are dustbins everywhere. It is requested that everyone please picks up after themselves

14 6

 Comment

14 7

 Comment

Can people please stop littering at the khokha. It is so sad to see people litter when there are dustbins everywhere. It is requested that everyone please picks up after themselves

The opacity of the bottom navigation changing based on the section you are currently at:



Home



Instructors



Work



Subspace



Messages



Home



Instructors



Work



Subspace



Messages

Match between system and real word: Familiar and easy terminology have been used so that users may get easily familiar with the app as it improves usability. Metaphors are also used to show parallels between the system and real world such as “Like” to show if a user likes a post or “Notification” to notify people of activities going on within the app. Visual elements such as different buttons have also been used as users associate different colors with different meanings. Such as the use of red for a warning and green for a positive action



Saad Dar

Lost my speakers in the cricket ground last night. If someone finds them then please let me know!

 Approve Reject

User Control and freedom: Users are given freedom and control over the app by allowing them to post on multiple platforms such as the Newsfeed and Instructor reviews. Users are also allowed to navigate from one screen to any other screens via the use of components such as bottom navigation and navigation drawer. Users are also allowed to delete any post they might have posted, and like or comment on other users posts.

Consistency and standards: All the screens in the user side have the same layout as described in the website conventions. The same font has been used throughout the website to keep consistency. The icons used in the application are standard icons from material design. All of this is done so the users can get familiar with the application

Error Prevention: Actions which have negative consequences have their buttons in red as to warn the users.



Nimra Khan

Hello!
I have been feeling very lonely on campus
these days. I feel like the friends I have are not
really my friends and they don't want to hang
out with me. If anyone wants to hang out
sometime on campus please let me know we
can get chai or something.

Delete

Buttons have also been labeled with their respective actions so the user knows what action they are performing.

Aesthetic and Minimalist Design: The design of the screens is kept minimalistic in order to not overload the user's cognition. Color contrasts are used in order to keep the application aesthetically pleasing. Every screen serves a purpose and a screen only contains the functionality specific to that screen

Help users recognise, diagnose and recover from errors: If the users enter a non lums email the system will let them know that the email is non eligible. While signing up users will be required to enter their password twice and if the passwords are not similar an error will be displayed, this is done so the user may not accidentally enter another password. Similarly if the Verification code

entered by the user is incorrect the system will let the user know and also go back and recheck their entered email.

Recognition rather than recall: Every screen has a title on top of it to let the user know which screen they are at so that the user does not need to remember it.



The bottom navigation also helps the user identify which section of the application they are currently at. Text fields are also accompanied by their title so the user knows what to type inside them. All of this is done in order to reduce the information the user needs to remember so that they may not get frustrated while navigating the application.

Help and documentation: A view FAQ section has been included to help the users understand the application better and answer most of the questions that may arise from the use of the application



What is a Subspace?

A subspace is a forum made to cater the needs of people's interests, hobbies and passions. There's a subspace for whatever you are interested in, where you can interact with other people with similar interests. You can have discussions, post pictures, links and videos.

Why is my post not appearing in the NewsFeed?

In order for the most to appear on the Newsfeed it must be approved. If you can not see your post then that means it did not adhere to the community guidelines and was disapproved by the moderators.

6 Other Non-functional Requirements

- <Refine this section based on the final system design. Highlight the changes or additions>

6.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.

TODO: Provide at least 5 different performance requirements based on the information you collected from the client. For example you can say “1. Any transaction will not take more than 10 seconds, etc...>

6.2 Safety and Security Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied. Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements.

TODO:

- Provide at least 3 different safety requirements based on your interview with the client or, on your related research, and again you need to be creative here.
- Describe briefly what level of security is expected from this product by your client and provide a bulleted (or numbered) list of the major security requirements.>

6.3 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

TODO: Use subsections (e.g., 4.3.1 Reliability, 4.3.2 Portability, etc...) provide requirements related to the different software quality attributes. Base the information you include in these subsections on the material you have learned in the class. Make sure, that you do not just write “This software shall be maintainable...” Indicate how you plan to achieve it, & etc...Do not forget to include such attributes as the design for change. Please note that you need to include at least 2 quality attributes, but it is the mere minimum and it will not receive the full marks.>

Appendix A - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist **ME and the Teaching Assistants** to determine the effort put forth to produce this document>

Group meeting 01:

Date: 28/02/2022

Time: 10:00- 12:00pm

Discussion: Initial discussion over document and work division as well as start of making points for some parts of section 3.

Group meeting 02:

Date: 01/03/2022

Time: 5:00 - 7:00 pm

Discussion: Confusions over section 4 of document and all diagrams members discussed diagram sequence and structure diagrams divided between group members. Readings and research over respective diagrams initiated.

Group meeting 03:

Date: 02/03/2022

Time: 6:00 - 9:00 pm

Discussion: Work started on activity diagrams and further discussions/confusions over sequence diagrams identified. Meanwhile other group members initiate work on screens choosing color themes and setting up figma projects.

Group meeting 04:

Date: 03/03/2022

Time: 9:00 - 11:30 pm

Discussion: All activity diagrams ready in collaboration 2/3 sequence diagrams also completed work started on class diagram (initial discussion). Screens are made at a steady pace following alignment and consistency guidelines studied from HCI course.

Group meeting 05:

Date: 04/03/2022

Time: 6:00 - 9:00 pm

Discussion: All sequence diagrams completed class diagram entities identified from functional requirements of SRS confusion over their relationships more study started to understand relationships. Login signup and initial dash screens completed as well as newsfeed screen layout which contains major paths to other major use case screens.

Group meeting 06:

Date: 05/03/2022

Time: 10:00 - 12:00 pm

Discussion: Class diagram completed and entities discussed and written down for database schema relationships not yet completed confusions of UML component diagram still there but started nonetheless to wind up section 4. Further smooth progress on screens Sidebar and minor use case screens prepared.

Group meeting 07:

Date: 06/03/2022

Time: 2:00 - 5:00 pm

Discussion: Database Schema completed UML component diagram query asked on slack and cleared and component diagram progresses further. Major use case screens are being made at a steady pace. Nearing completion.

Meeting with TA:

Date: 07/03/2022

Time: 1:30-2:00pm

Discussion: Confusions about document Component diagram example shown to help progress queries about minor details about document as well as checking major diagrams. And showing screens to see if there are any major mistakes.

Group meeting 08:

Date: 08/03/2022

Time: 10:00 - 12:00 pm

Discussion: Screens completed and screen descriptions also near completion document almost ready with exception of minor parts such as group log change log contribution statements etc.

Group meeting 09:

Date: 09/03/2022

Time: 4:00 - 6:00 pm

Discussion: Completion of screen descriptions minor touches to screens all diagrams added to document, document compilation and formatting. Minor sections left completed and appended phase completed.

Appendix B – Contribution Statement

Name	Contributions in this phase	Approx. Number of hours	Remarks
Aswad Tariq	1 Change log 1.1 Project Scope 2.1 Document Purpose 2.2 Project Scope 2.3 Intended Audience and document overview 3.1 System Overview 4.1 System Architecture	22	
Haseeb-Ur-Rehman Faheem	1 Change log 4.4 Database model 4.5 External interface requirements 5.1 Description of the user interface 5.2 Information architecture	22	
Zeerak Babar	4.2 Subsystem Architecture 3.2 System Constraints 3.3 Architectural Strategies 5.3 Description of the screens	22	
Nashrah Shaukat	2.4 Definitions,Acronyms and Abbreviations 2.5 References and Acknowledgements 5.3 Screens and their documentation 5.4 User Interface Design rules Formatting of final document	22	
Ibrahim Sanaullah	4.5 External Interface Requirements 5.3 Designed all client-side screens as well as their description 5.4.3 User interface design rules Appendix A Appendix B	22	

