Software Design Specification Document

(CS360)

LUMSafar

Group Number: 19

Ahsan Sarwar

Saood Ahmad

Omer Kamran

Abdullah Ahmad

Course: Software Engineering CS360

Instructor: Suleman Shahid

University: Lahore University of Management Sciences (LUMS)

Version: 1.0

Date: (09/03/2020)

Number of hours spent on their document: <104hrs>

Contents

1 Change Log

1.1 Project Scope

Lumsafar 1.0 aims to be the online social hub for LUMS students, by facilitating their day-to-day socializations. The product, as the name suggests, is targeted for the LUMS community, however, if it gains enough traction, we can expand to other institutions as well.

Socialization is something many at LUMS struggle with. Whether it's owing to busy schedules and study routines leaving no time for partaking in social activities, or social anxiety preventing people from putting themselves out there. With public forums like LDF not being efficient enough for their purpose due to poor filtration and no categorization, a specialized platform was needed.

Once the application is populated, users will be able to meet new people and try out new things happening on and off campus. The apps categorization and tags system will further improve their process by connecting with users with the type of people that they actually want to meet. All this, at the touch of a button.

Societies of all scales will be able to increase their reach, and will be able target people with goals that align with the society's mission and expertise. It will provide them a platform to promote their events in order to maximize visibility and minimize the need to spam social media groups. The app can keep a record of attendance which will aid them in estimating the expected attendance of the event, which will in turn help in managing accommodation and logistics.

Holistically, the LUMSafar intends to promote a sense of community and increase networking by bringing people together to create meaningful experiences.

1.2 Change log

Authentication:

- Added the option for user to reset password

Socialization:

- Removed RQ11,12
    - Removed the feature for users to comment on event posts
    - Added the feature for people to chat with their friends.
- Removed RQ8
    - Removed the feature for users to link their other social media apps. They can chat directly with other people using the application.
- Removed RQ16
    - Removed the feature that suggests other people to users based on similar interests.
- Removed RQ18
    - Removed the functionality where
- Users can search people and follow them. This will allow them to receive notification for their events
- Users can send friend requests to each other. This will allow them to see their location on the lums map

2 Introduction

2.1 Document Purpose

This document describes how the requirements of Lumsafar 1.0 will be implemented. It is intended to aid the programmers of the software. It will cover the software in its entirety, including the system architecture and the user interface. This is a standalone mobile based application. thisdocument represents the very first build of the product - Version 1.0.

Furthermore this document also lists the methods we will use to implement an interactive user interface and a backend database management. This document is supposed to guide the development team throughout the development phase and save time as it will help identify the possible limitations in the final phase, ahead of time.

2.2 Product Scope

Lumsafar 1.0 aims to be the online social hub for LUMS students, by facilitating their day-to-day socializations. The product, as the name suggests, is targeted for the LUMS community, however, if it gains enough traction, we can expand to other institutions as well.

Socialization is something many at LUMS struggle with. Whether it's owing to busy schedules and study routines leaving no time for partaking in social activities, or social anxiety preventing people from putting themselves out there. With public forums like LDF not being efficient enough for this purpose due to poor filtration and no categorization, a specialized platform is required

Once the application is populated, users will be able to meet new people and try out new things happening on and off campus. The apps categorization and tags system will further improve this process by connecting with users with the type of people that they actually want to meet. All this, at the touch of a button.

Societies of all scales will be able to increase their reach, and will be able target people with goals that align with the society's mission and expertise. It will provide them a platform to promote their events in order to maximize visibility and minimize the need to spam social media groups. The app can keep a record of attendance which will aid them in estimating the expected attendance of the event, which will in turn help in managing accommodation and logistics.
LUMSafar intends to promote a sense of community and increase networking by bringing people together to create meaningful experiences.

In a nutshell, LUMSafar:

- Intends to promote a sense of community and increase networking by bringing people together to create meaningful experiences.
- Provide a common platform to societies and students which is directed towards social activities hosted by students and societies.
- Provide students an opportunity to socialize at the push of a button without having to make strenuous plans beforehand.
- Allows students to match up with other people at LUMS with similar interests as their own.

2.3 Intended Audience and Document Overview

Intended Audience:

This document is intended for the development team to refer to from time to time in order to avoid any discrepancies, if we face any at all. Moreover this document can also be used by the client, in our case, the OSA to review how different components work individually and fall in place together as well as provide a visual representation of how the final product will look once the requirements are mapped to it.

Students of LUMS / Societies of LUMS: These are the main users of the application. All of the requirements and features described in this document are subject to customer approval and may be changed at any time. Furthermore, all product specifications specified here that have been

authorized by the client will be binding and irreversible. As a result, the following is the recommended reading format:

Instructor / Teaching Assistant:
Dr. Suleman Shahid is the instructor for the course and supervisor for this project. He will be overlooking our progress and guide us through the project.
Hareem Raza is the assigned TA to our group to overlook the project. She reviews our work and provides crucial inputs and updates. She is also very nice.

Document Overview:
The overall description section gives a brief overview of the system stating how the different components of the system are organized and how they interact with one another as the application undergoes its main purpose. It also contains architectural strategies which explain why we are using a certain tool that we are using.
The system architecture section contains information about how we decompose our system into components and how the overall view of the system looks with respect to its technical details. This section also includes details about how the system backend will function along with the front end application. Illustrations are used to convey the exact functioning of some important components of the system.

The user interface section provides details on the visual and frontend of the application. We provide a visual description of what each screen will look like and how all the frontend components are linked and how they can be navigated. Each screen of the application is described in detail.

2.4 Definitions, Acronyms and Abbreviations

Agile (development team)	A development team that creates software while following the principles of cross-functionality and self-organization

Android	A mobile operating system.
API gateway	an API management tool that sits between a client and a collection of backend services.
Connect with Stranger	A feature of the app that will allow users to converse with other users of similar interests
Driver	a computer program that operates or controls a particular type of device that is attached to a computer
Event	Refers to a social activity taking place on LUMS. Can be a study session or a social hangout activity.
Expo	"a framework and a platform for universal React applications"
Following	Society accounts the user is following
Friends	Users whose QR codes the user has scanned so as to be able to share location on LUMS Map with them
GPS	Global Positioning System
Home page	The introductory page of the application
Interactive LUMS Map	The idea is pretty straightforward as the name suggests. The app will feature an interactive map that allows the students to view avatars of other students and their locations and also points of interests around LUMS.
Interest tag	A word or phrase that classifies one interest of the user
iOS	"an operating system used for mobile devices manufactured by Apple Inc."
Javascript	A programming language, usually used for web

	development
Jest	“a JavaScript testing framework designed to ensure correctness of any JavaScript codebase”
Latency	“The delay before a transfer of data begins following an instruction for its transfer.”
LDF	LUMS Discussion Forum
LUMS	Lahore University of Management Sciences
mongoDB	A document-oriented NoSQL database program
QR Code	a machine-readable matrix barcode that contains information about the item it is attached to
React	an open-source front-end JavaScript library for building user interfaces
React-Native	A UI Software framework used to develop applications.
Schema approach	an approach that involves representing the plan in form of a model or outline
Society account	a user account that will represent a unique society at LUMS.
Student account	a user account that will represent a student at LUMS
Talking space	an open online podcast where everyone can talk with each other
Toolbars	a strip of icons that can be clicked to perform certain functions.”
Touch-screen	a display screen with which the user interacts by touching it
Widget	“an application, or a component of an interface, that enables a user to perform a function or

	access a service.”

2.5 References and Acknowledgments

Lucid Chart Online Diagram Software & Visual Solution | Lucidchart - Web based platform used to create the activity diagrams in section 4.

Draw.io draw.io – Diagrams for Confluence and Jira - draw.io (drawio-app.com) - Online diagramming website used to create the component diagram in section 4 and the architectural layer diagram in section 3. Also the information architecture diagram in section 5.

Miro Miro | Online Whiteboard for Visual Collaboration - Online diagramming tool used for creating all other diagrams.

Material.io - Design guideline by Google

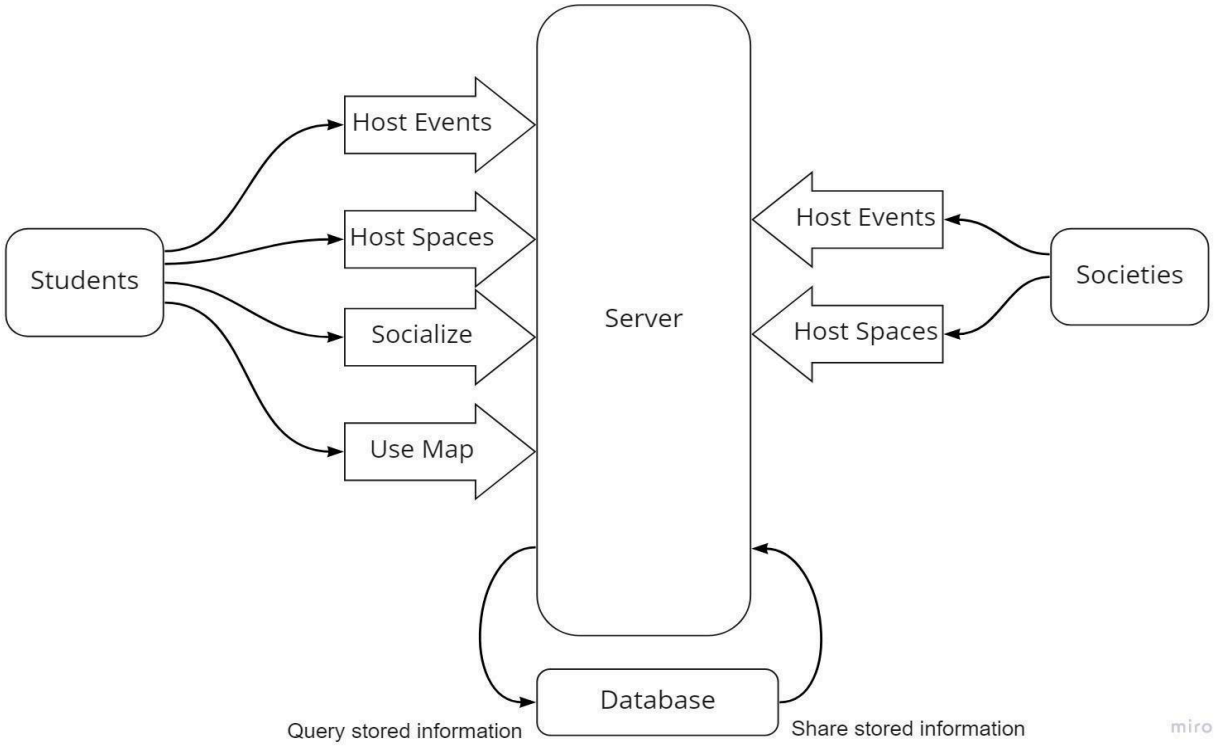
5 Benefits of Developing JavaScript Sites & Apps (telerik.com) - Why we are using JS

https://talkjs.com/dashboard/signup/premium/ - Chat APIs

3 Overall Description

3.1 System overview

The main purpose of the application is to provide students of LUMS a social app specifically tailored to make the user experience LUMS social life to the fullest. The users interact with the system by using a user-friendly interface. Since the application is aimed at facilitating socialization among LUMS students, we will be keeping track of users and their interactions with all other users, events and talking spaces. A MongoDB server will be used to store the above data.



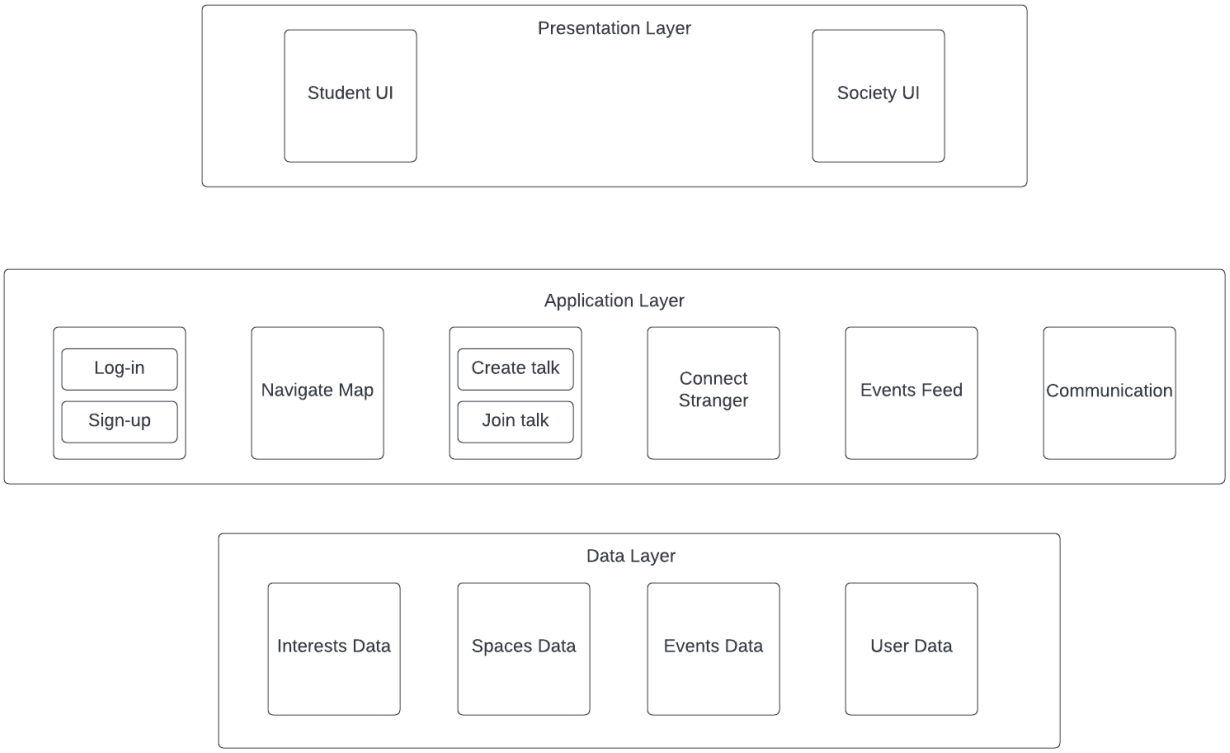
Context Diagram

3.2 System constraints

- Security of stored passwords must be maintained. (Security)
- The application requires a stable internet connection otherwise not all subsystems will function properly .
- The application will need to take following permissions from the Android device:
    - Permission to access, edit and delete data
    - Permission to use camera
    - Permission to use microphone/ speakers
    - Permission to use GPS
- The application will only be installable on Android cell phones (environment)
- The application will require the device to have GPS capabilities to operate.

- The application will require the device to have a touch-screen
- The application will require a microphone and speakers/headphones to make use of the voice chat in talking spaces.
- The application will require a camera to capture QR codes to add friends.
- The application is meant to run on android phones running version 8.0 or higher.
- The users should have devices that support JavaScript (environment)
- There is only one lingual language that will be supported, the English Language, which will make it difficult for users who are not fluent in English to have trouble navigating the application
- The application must follow standard compliance protocols such as Data Retention, Encryption, Low Latency, Clarity, etc.
- Rapid expansion and further integration of features in the application would be difficult down the line due to the limitation of React JS and the MERN Stack as a whole, as compared to MEAN Stack, which uses Angular.

3.3 Architectural strategies



Architectural Layer Diagram

We used layered architecture for the application because this allows a clean separation in types of components used and also helps in gathering all similar programming code together in one location. These layers become independent by isolating them, so it becomes easy to use.
Since our audience is digitally literate therefore we employed a narrow and deep architecture model.

Programming Language:

“We are developing our backend using JavaScript due to the following reasons:

- **Fast & Responsive:** One of the key bottlenecks in modern web development is network latency. The time it takes to make a request to the server and get the results can account for a huge portion of page load time. With JavaScript, this is achieved by sending JSON data to the client instead of a mix of other data types.

- **Universal Front-End Platform:** No matter which technology is used on the server, JS can be used to provide a rich front-end. Not only does this make your front-end development more reusable and insulated against a changing server landscape, but it also lets you optimize server-side code for backend tasks.
- **Industry Momentum:** Finally, while not the best benefit of JavaScript development, the industry momentum behind JavaScript/HTML5 should be considered. JavaScript is the world's most popular programming language. Google, Apple and Microsoft have all thrown significant muscle and money behind this universal approach to development. And new tools, frameworks, and money making opportunities are emerging at a rapid pace for JavaScript developers."

Libraries:

- **React-Native Frontend** - It is a component based library that acts as an abstraction over the native platform. it allows us to quickly make UI across multiple platforms without worrying about the native intricacies. It also has a vast package library and built in elements
  since our team has prior experience with React, migrating to react native will be a smooth process
- **ExpressJS Backend**
- **NodeJS** to run code for server application

APIs:

Additionally we will be using established APIs for network tasks such as VOIP for the talking spaces and the chat feature

Will be leveraging the functionality of google maps to implement our lums map function

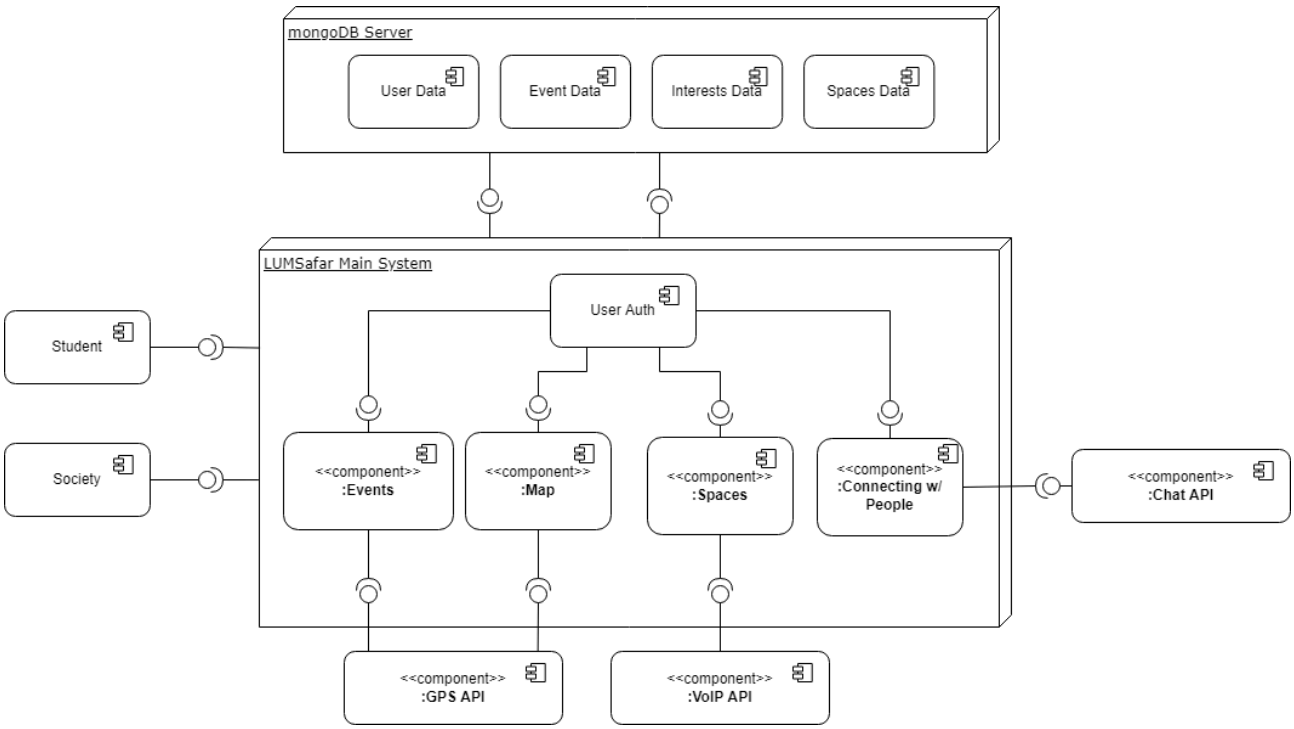
For ensuring quick logins and avoidance of redundant functions, selective data, such as login credentials

MongoDB is a source-available, cross-platform, database program and is classified as a NoSQL database program. We will be doing backend development of the application on MongoDB. Reasons for using the database of choice are following:

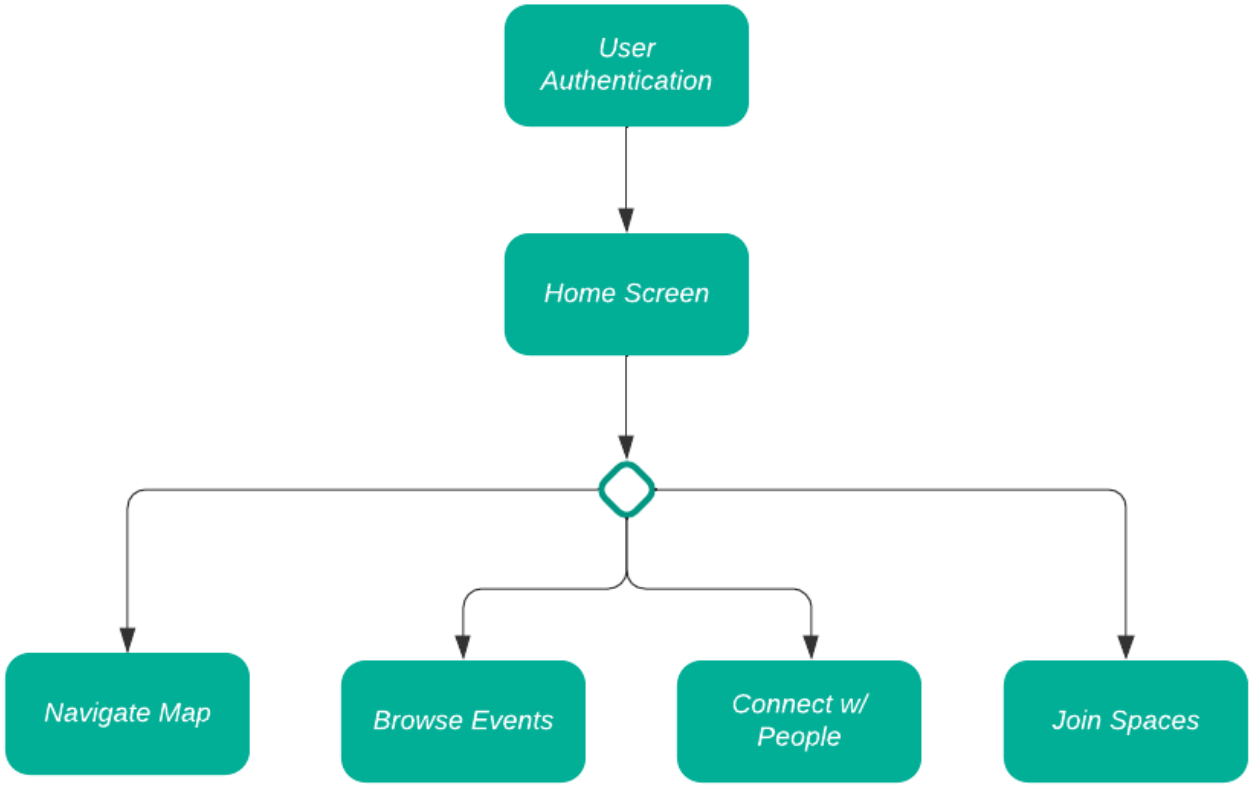
- MongoDB is an easy to use document database used to create scalable and highly accessible mobile apps. It's useful among agile development teams because of its flexible schema approach. MongoDB provides drivers for all major programming languages, allowing you to start developing your project right away without having to worry about setting up a database.
- MongoDB is efficient because of its ability to handle large amounts of unstructured data. Most people experience mongoDB in real life performance because users are allowed to query in a sensitive-to-workload manner.
- In previous projects we gave MySQL a go, however, we faced difficulties in managing sql queries, therefore, we wished to avoid that in a noSql database

4 System Architecture

4.1 System Architecture

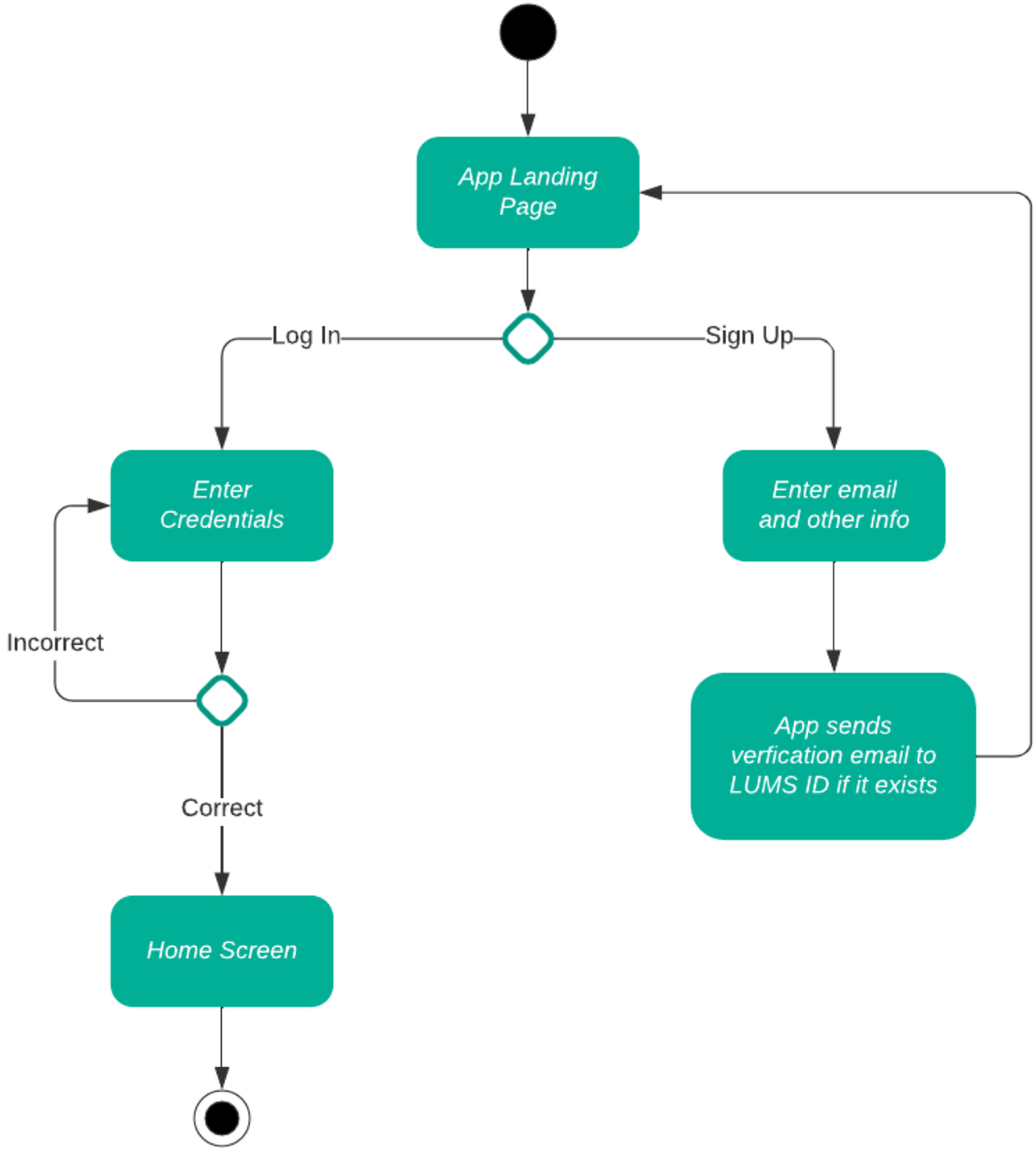


The above diagram illustrates how each component in the application interacts with the other components in the same system and the components that exist outside the system.



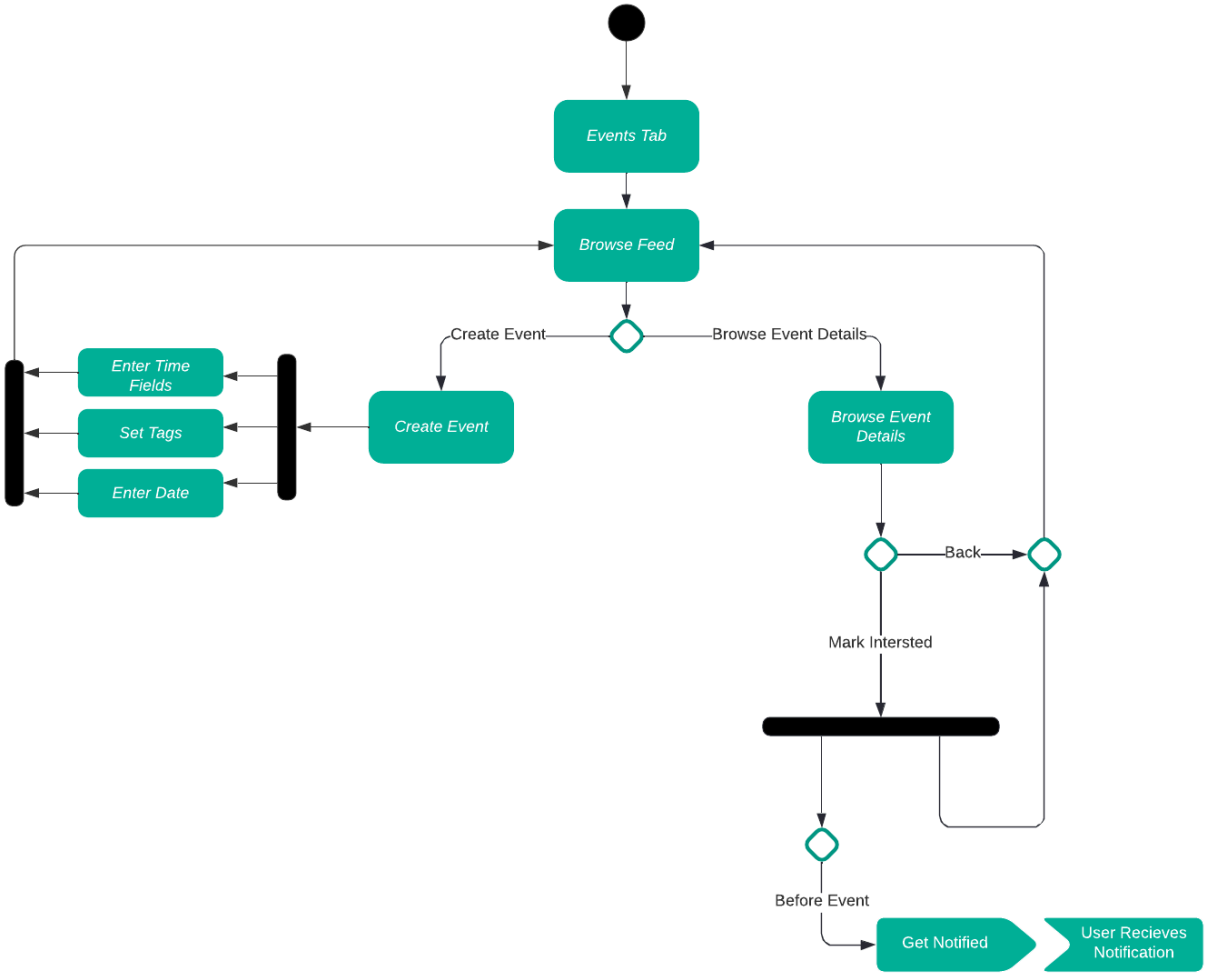
High Level Activity Diagram

The diagram above gives an abstract high level representation of the application. It starts with the user authentication and then lands the user on the home screen. From where the user gets the choice to navigate to different tabs in the application.



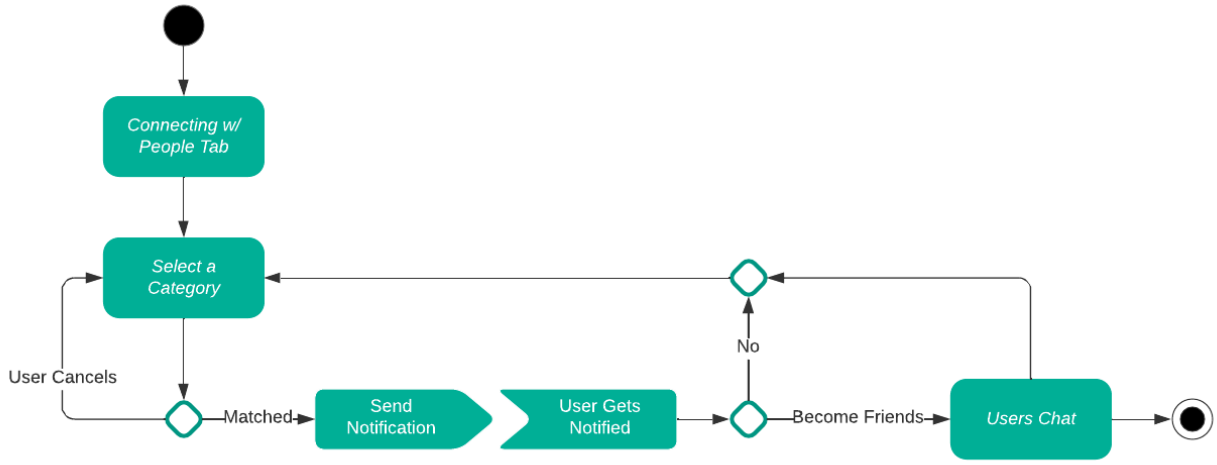
User Authentication Activity Diagram

The diagram above illustrates the user authentication process. The user will be required to either sign up or log in. Signing up will require the user to authenticate their LUMS email via a verification code that is sent to their email. Logging in will simply land the user or the home screen.



Events Activity Diagram

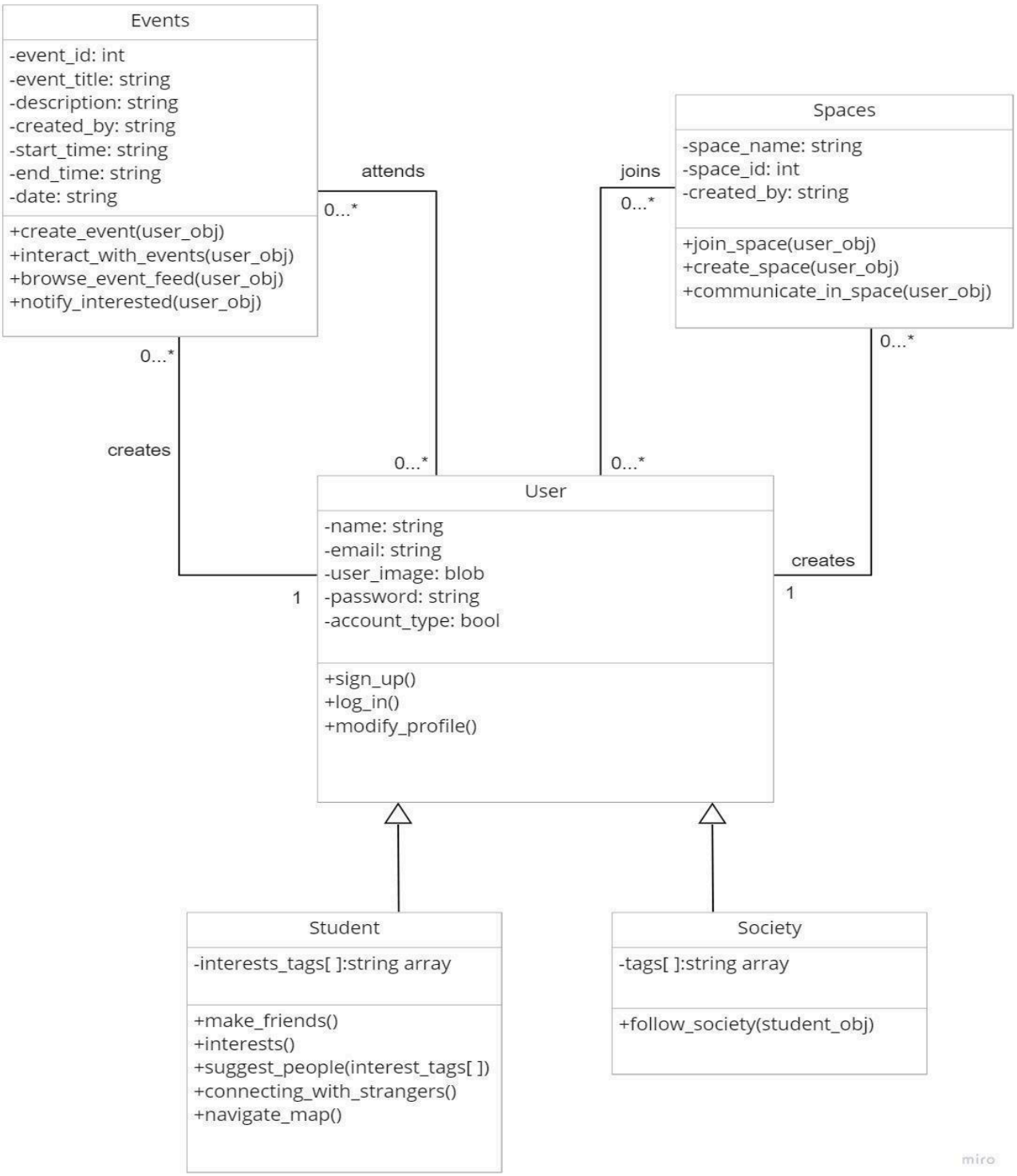
The above diagram illustrates the flow of the application when the user has to deal with events. They can create or mark themselves interested in events and get notified a specified time before the event. Creating an event will take them to the event creation page where they can fill in the event details and have the event published on the event feed.



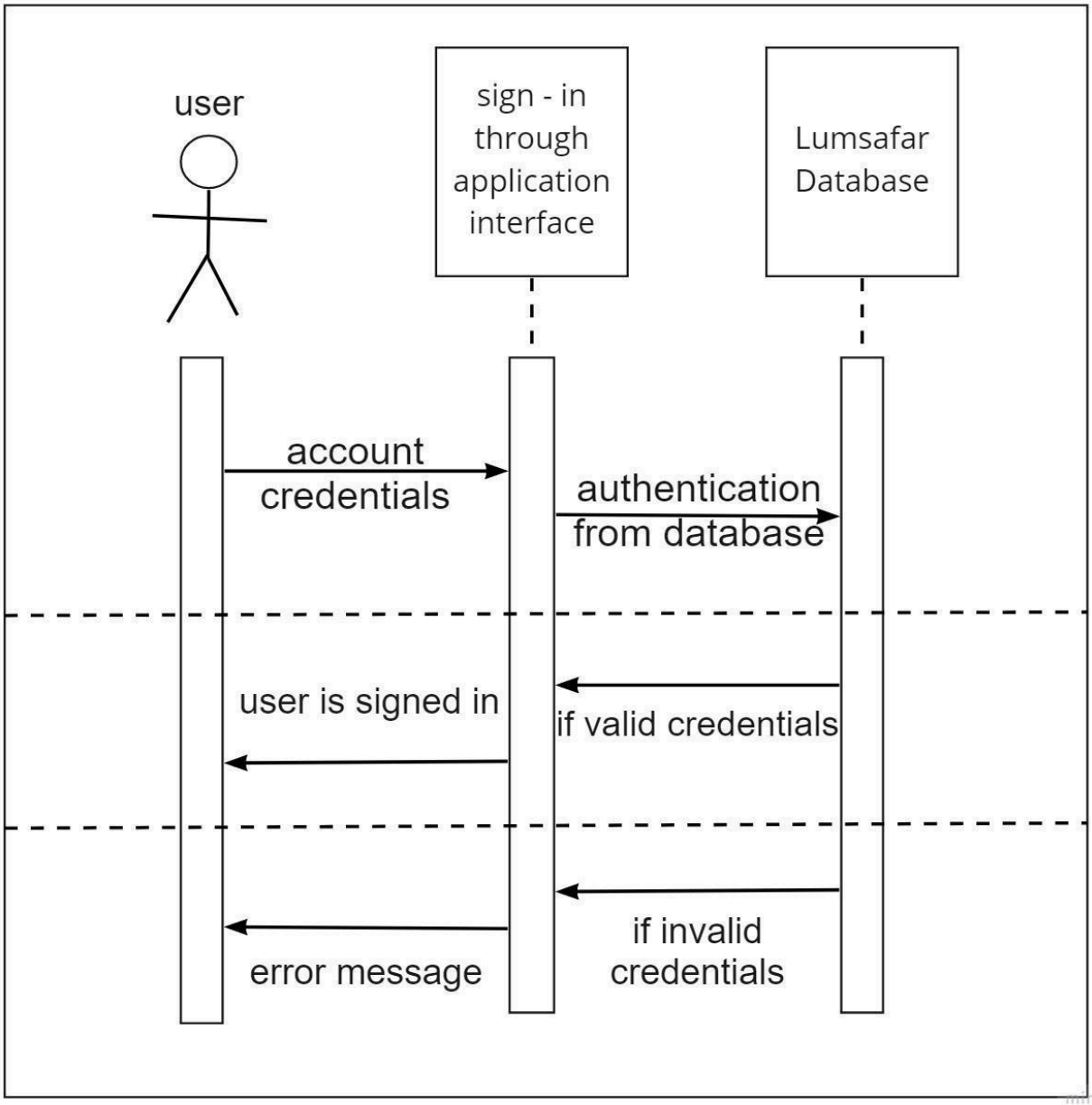
Connecting With People Activity Diagram

The diagram above is for the “connecting with people” feature. The user will be required to select a particular category for which they want to be matched up with another person. Being successfully matched up will notify the user about the match. The matched pair will then decide to continue by chatting or by declining the match.

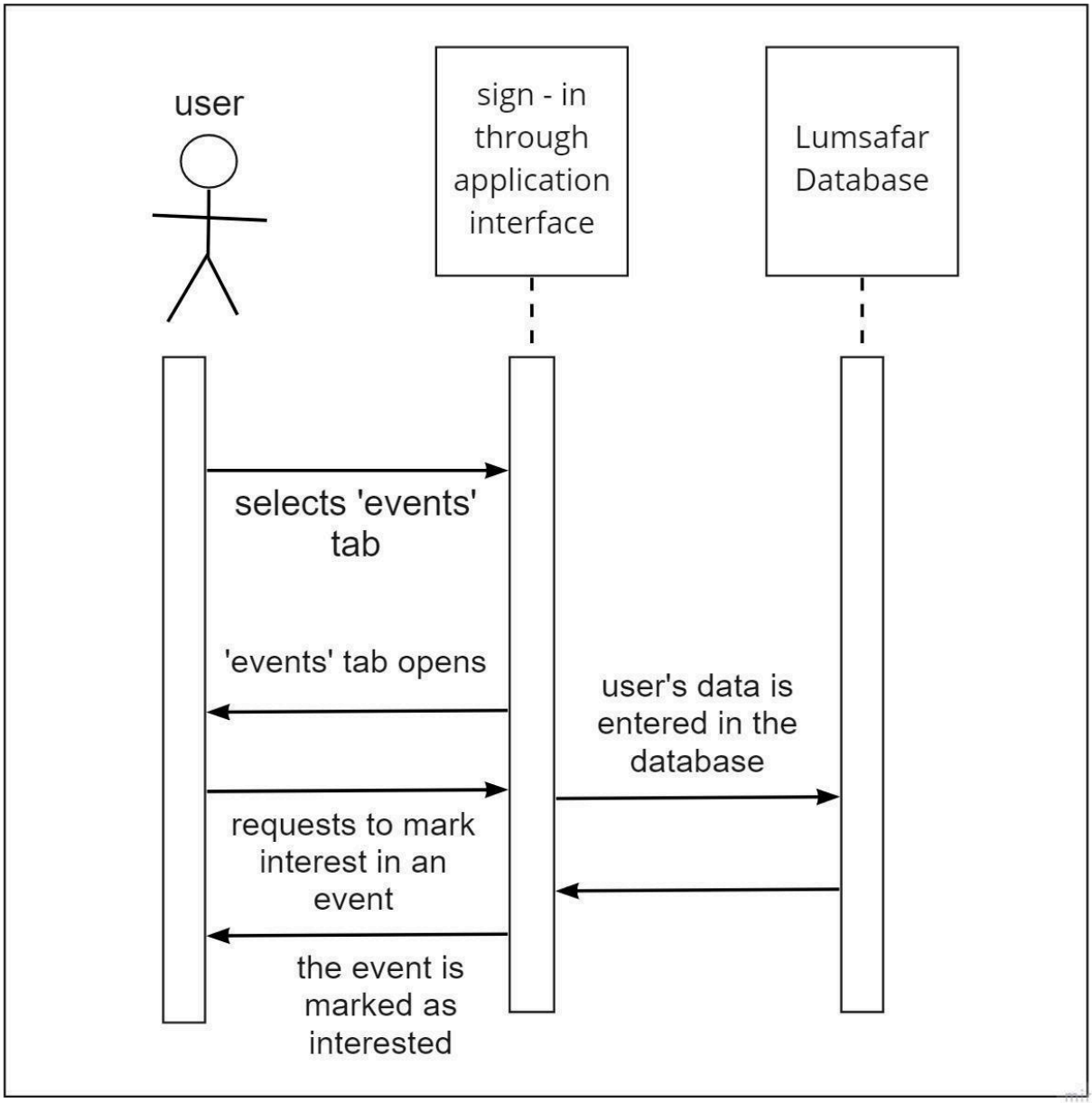
4.2 Subsystem Architecture



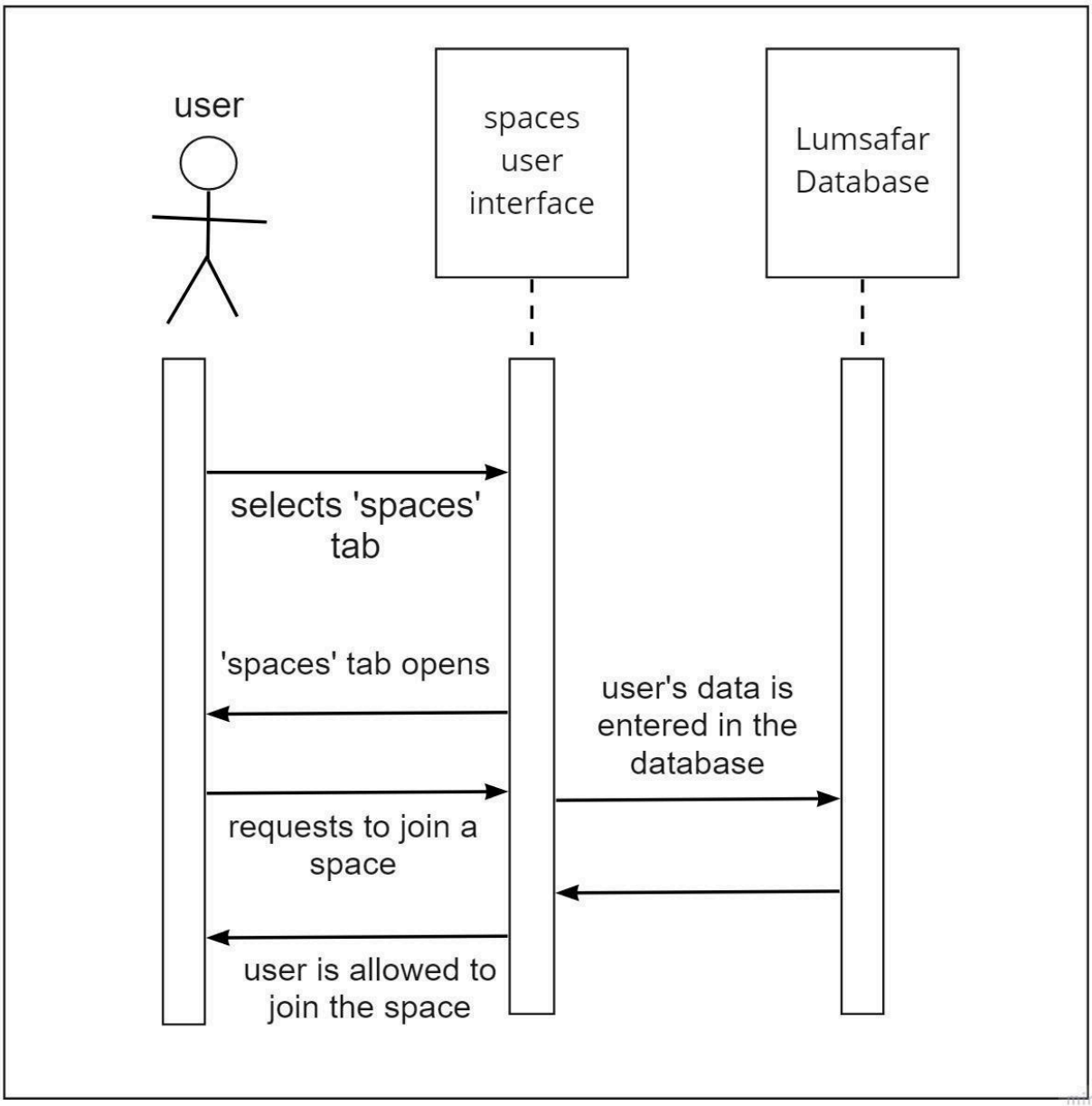
Class Diagram



User Sign-In Sequence Diagram



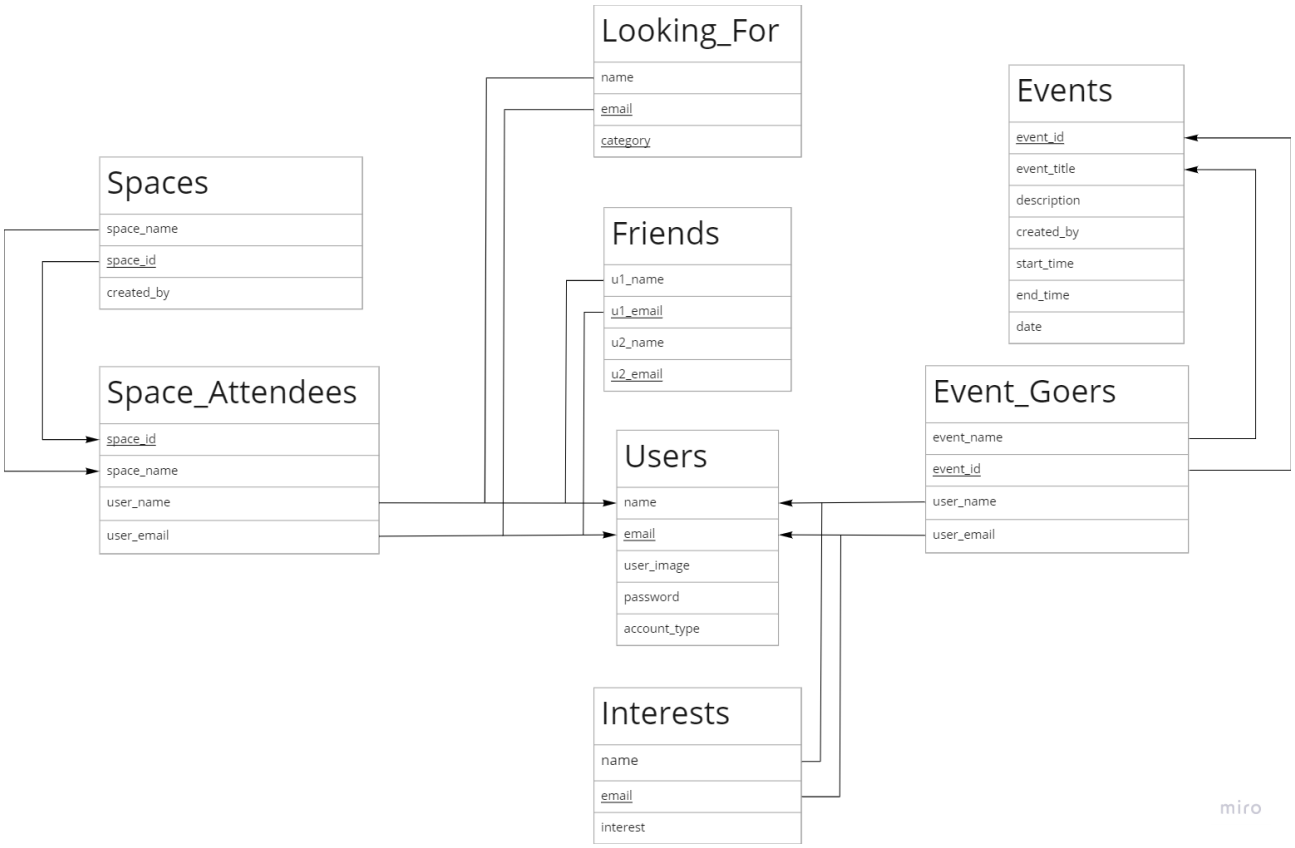
Mark Interested in an Event Sequence Diagram



Join Space Sequence Diagram

4.3 Database Model

4.3.1 Database scheme and detailed description



Description of Table:

Users: Holds the essential details and credentials of the application users

Attribute	Data type	Description
name	string	name of the app user will be stored here

email	string	LUMS email ID will be stored here. This will be used for signing in the user account
user_image	blob	the profile image of the user will be stored here
password	string	password of the user account which will have a combination of alphanumeric and symbols. Will be used for signing in the user account along with the email ID
account_type	boolean*	specifies if the user is a society account holder or a normal user

Friends: holds the relationship between those users who are each other's friends on the application

Attribute	Data Type	Description
u1_name	string	name of a user
u1_email	string	email of a user
u2_name	string	name of the friend of the user1
u2_email	string	email of the friend of the user1

Interests: stores the users and the corresponding categories in which the user is interested in (upto 5)
(categories TBA*)

Attribute	Data Type	Description

name	string	name of the user
email	string	email of the user
interests	string	category the user is interested in. These are predefined

Events: Essential details about the events created by the users. when a new event is created, its is added to the database. Once its end time is reached, it is removed

Attribute	Data Type	Description
event_id	Int	unique ID for the event, generated by the system
event_title	string	title of the event for easy identification
description	string	short description of the event provided by the creator
created_by	string	email of the user who created event
start_time	time*	the time at which the event is scheduled to start
end_time	time*	the time at which the event is scheduled to end
date	date	the exact date of the event

Event_Goers: lists the people interested in events. These records exist as long as the event exists or the user is no longer interested in the event.

Attribute	Data Type	Description
event_name	string	name of the event
event_id	int	unique ID of the event
user_name	string	name of the user interested in an event
user_email	string	email of the above user

Looking_For:

Attribute	Data Type	Description
name	string	username
email	string	user email
category	string	name of the category that the user is currently interested in, and wants to find a buddy for

Spaces: List the essential details of currently active spaces

Attribute	Data Type	Description

space_name	string	provided by the creator
space_id	string	system generated
created_by	string	email of the creator

Space_Attendees: Lists the attendees of spaces. The records remain as long as the space is active or if the user leaves the space

Attribute	Data Type	Description
space_name	string	name of the space
space_ID	string	unique system generated ID
username	string	names of the users who have joined the space
user_email	string	emails of the users who have joined the space

4.3.2 Database

MongoDB is a source-available, cross-platform, database program and is classified as a NoSQL database program. We will be doing backend development of the application on MongoDB. Reasons for using the database of choice are following:

- MongoDB is an easy to use document database used to create scalable and highly accessible mobile apps. It's useful among agile development teams because of its flexible schema approach. MongoDB provides drivers for all major programming languages, allowing you to start developing your project right away without having to worry about setting up a database.
- MongoDB is efficient because of its ability to handle large amounts of unstructured data. Most people experience mongoDB in real life performance because users are allowed to query in a sensitive-to-workload manner.
- In previous projects we gave MySQL a go, however, we faced difficulties in managing sql queries, therefore, we wished to avoid that in a noSql database

4.4 External Interface Requirements

4.4.1 User Interfaces

Our software will be a touch based android application. The entire user interface will be written in English. We'll use a graphical user interface that will be simple to use.

- Application will be used by LUMS students and societies which have their student ids.
- There will be normal google logos to guide through the app.
- Every button/function will have its own description for the starters
- Buttons will be large and every button will have legible text.
- Notification will be sent for all the updates
- The app is more of a socializing app, so its features will be mostly about socialization.
- Users will be taken to a new screen for every single button according to their use.

4.4.2 Hardware Interfaces

Hardware requirement for the application will be as simple as for other mobile applications which are following:

- This will require a stable internet connection because the data needs to be sent or received from the server.
- The app will take from 70-100 MB of the storage which will be mostly for the data stored on the local devices.
- HTTPS will be used to send and receive messages from the server.

- In order to run the application, Android phones should be updated to atleast Android Jellybean 4.1 (API 16). All hardware interfaces will be dealt with by the android OS. Recommended mobile specifications include:
    - 1GB RAM
    - Touch input
    - Minimum 360p display resolution

5 User Interface Design

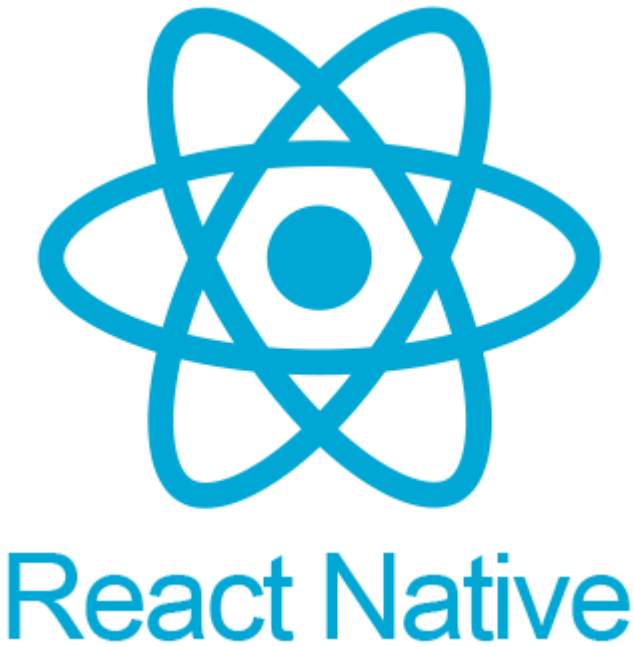
<A description of the user interface design of the software is presented.>

note : ***The User interface need to be complete in all functionality***

5.1 Description of the user interface

We will be using the following frameworks for our application development.

5.1.1 React Native



We will be using react native for the front end development of our application because of the following reasons:

- **Easy to Use:** It is a JavaScript framework that allows you to create real-time, natively rendered mobile apps for iOS and Android. It's built on React, Facebook's JavaScript library

for creating user interfaces. As most of the code you create can be shared across platforms, React Native makes it simple to develop for both Android and iOS at the same time.

- **Familiary:** It is easy to use and all the members of the group are familiar with it. Furthermore, developers can use React Native to construct mobile apps using website technologies. As a result, a developer with web development experience may quickly create a mobile app using React Native.
- **Efficiency**It is an efficient framework to use and it has a fast, efficient, and responsive user interface that reduces load time considerably. It's also much quicker and less costly to build React Native apps than to build native apps, without reducing the quality or functionality.

5.1.2 Expo

We will be using Expo as a layer on top of react native due to the following reasons:

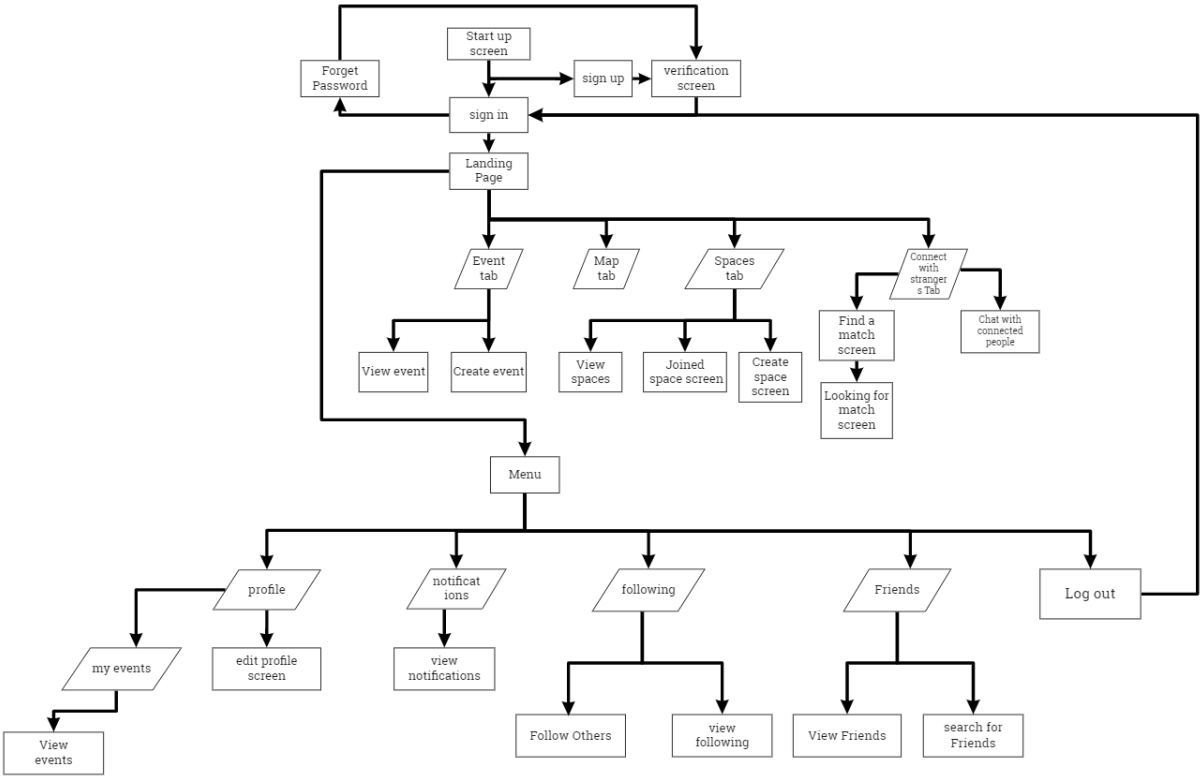
- It allows us to develop cross platform apps without touching any platform-native code. As our team is relatively inexperienced in Android development, this will allow us to focus on the functionality of our own app rather than the intricacies of the platforms we are developing on.It allows us to test our app on real physical devices at the scan of a QR code. This makes iterating over designs and fixing bugs extremely fast.

5.2 Information architecture

<To show flow between different screens. Google it for more details. I will also discuss their in class.>

TO DO: Design the complete Information architecture and present it visually

- Discuss the relationship between screens



5.3 Screens

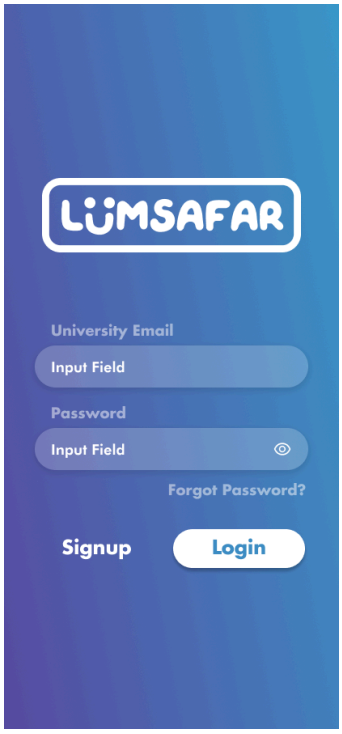
To do: Screen images with description – explain each item on the screen e.g. button, text field, etc. Explain how the screen is mapped on one of the user requirements/use cases

1. **Landing Screen**

Goal: Provide the user an attractive screen to look at while they wait for the application to start.

UI: The UI consists of a stationary image of the application's logo.

2. Log in process



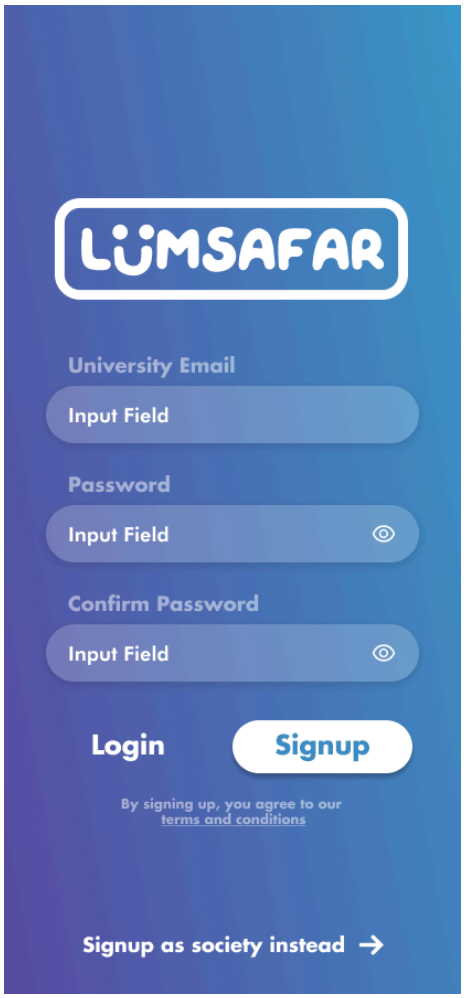
a. Login

Goal: Verify that the user indeed is a LUMS student.

UI: The user is prompted to enter their university email address and password to log into their account. Two labeled text fields clearly cue the user to input the above details. A login button processes the entered information, while a login button can be used to navigate to the login screen.

In case the user has logged in on the application, they will not need to do so again until they log out.

Relevant requirement: RQ3

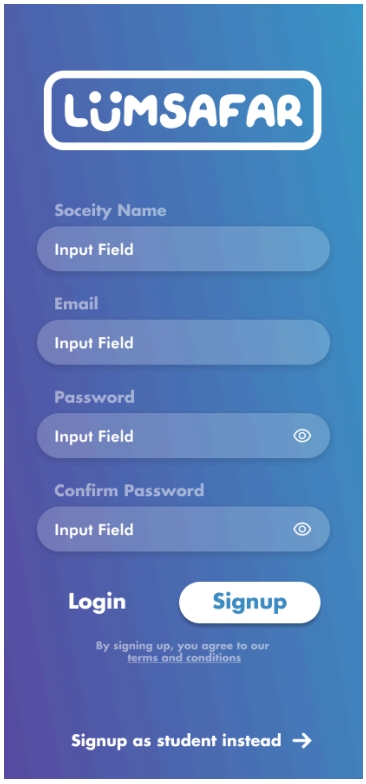


b. Sign Up as Student

Goal: Allow the user to create a student account on the application.

UI: The user is prompted to enter their username, email address, password and confirm password. Four labeled text fields clearly cue the user to input the above details. A signup button processes the entered information, while a login button can be used to navigate to the login screen. The user may press the arrow at the bottom to sign up as society instead.

Relevant requirement: RQ1



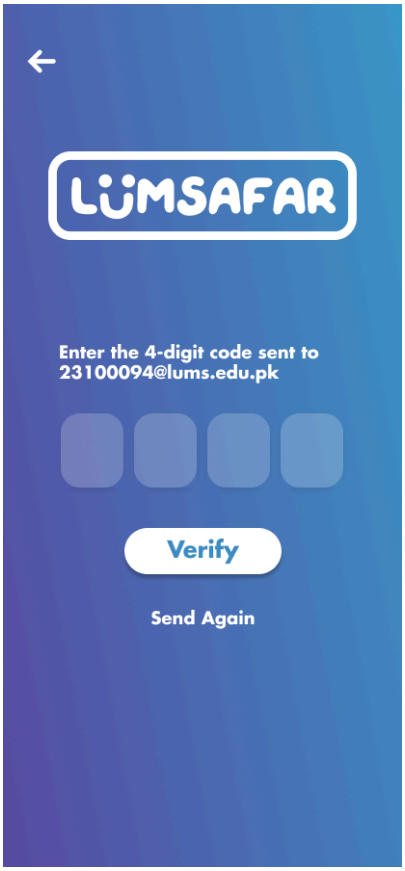
c. Sign Up as Society

Goal: Allow the user to create a society account on the application.

UI: The user is prompted to enter the society's name, its email address, password and confirm password. Four labeled text fields clearly cue the user to input the above

details. A signup button processes the entered information, while a login button can be used to navigate to the login screen. The user may press the arrow at the bottom to sign up as a student instead.

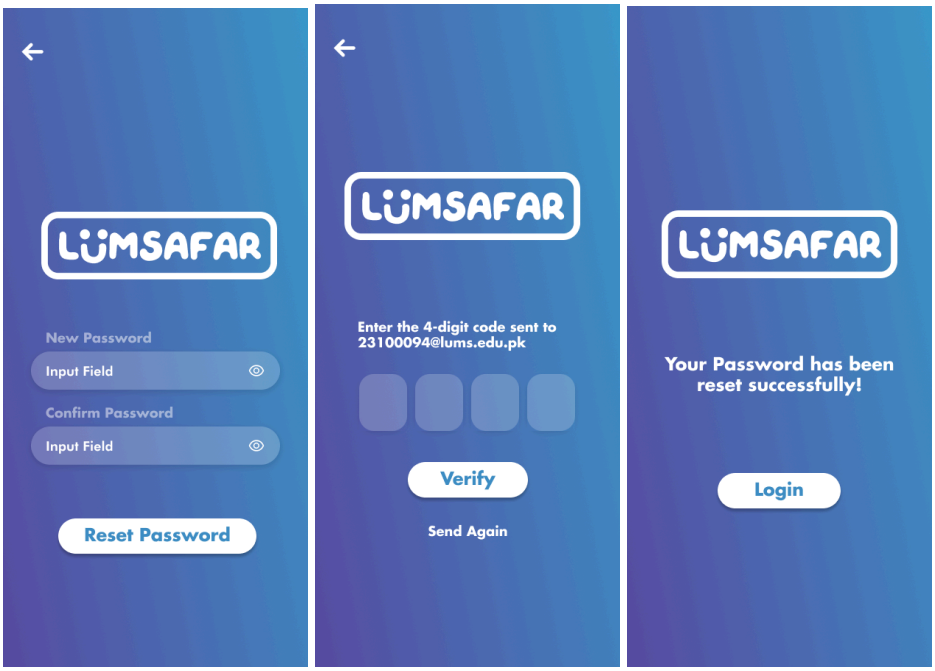
Relevant requirement: RQ1



d. Verification

Goal: Allow the user to authenticate their identity as a student/society of LUMS.
UI: The user is prompted to enter the 4-digit verification code sent to their LUMS email address. One text field will clearly cue the user to input the above detail. A verify button will process the entered code while a 'Send Again' button will send the code again in case the user did not receive the email.

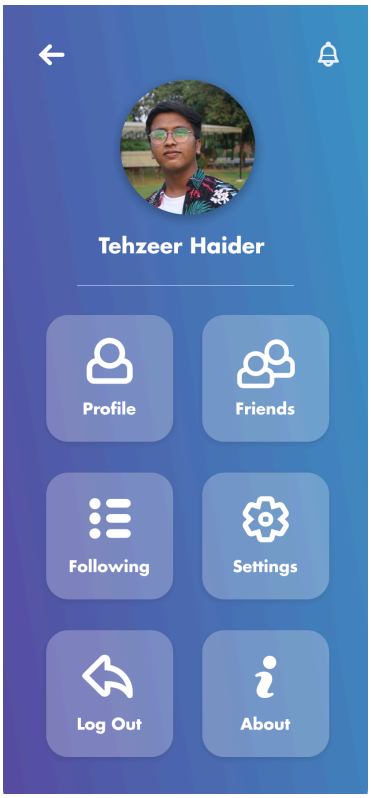
Relevant requirement: RQ2



e. Reset Password

Goal: To allow the user to reset their password in case he has forgotten it.

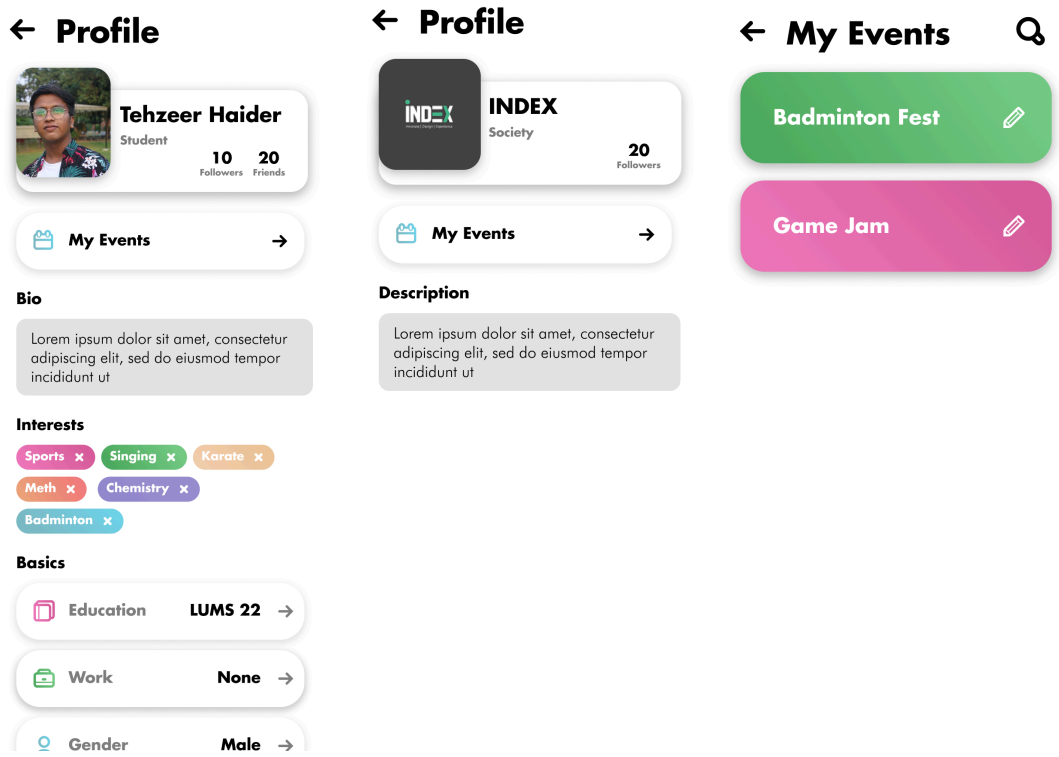
UI: The user will be cued to input their choice of new password in two text fields twice, after which he will press the Reset Password button. In case the passwords match, the password will be reset. If the passwords do not match, the same screen with an error message will be displayed to the user.



3. Menu

Goal: Two allow the user to access secondary features

UI: The menu button is denoted by a button of two horizontal lines (the higher one being the longer one), pressing which will open a screen with six options, as well as basic info of the logged in user, and a button to access the notifications.



a. Profile

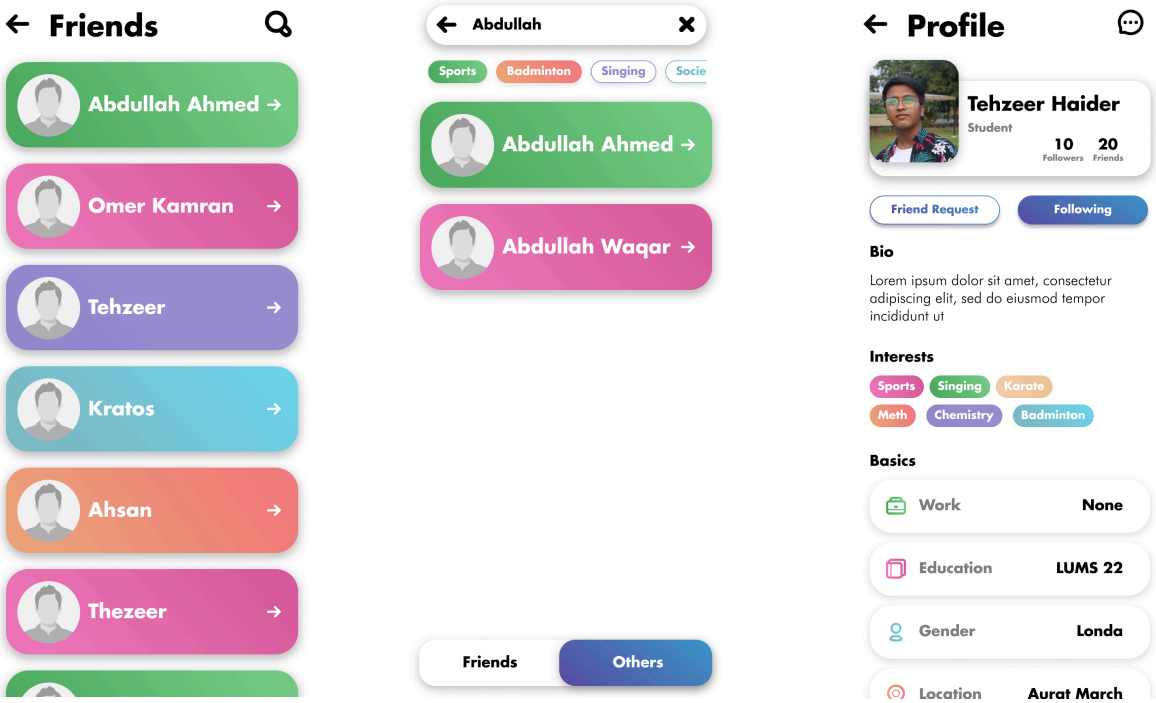
Goal: To allow the user to view their profile page and change their personal profile details such as username, biography, interest tags, basic personal details. The user may also view their number of followers, number of friends and all the events created by them.

UI: There will be clearly-labeled rounded rectangles containing text fields to cue the user to input the above details, if he wants to change any of them. The interest tags will be of different colors for better differentiation. The interface is user-friendly so as to not cause users any difficulty navigating through the icons.

The UI for society accounts will be different, with information such as Basics and Interests absent.

The My Events button will allow the user to browse and search events that they created and edit them

Relevant requirement: RQ5,6,7

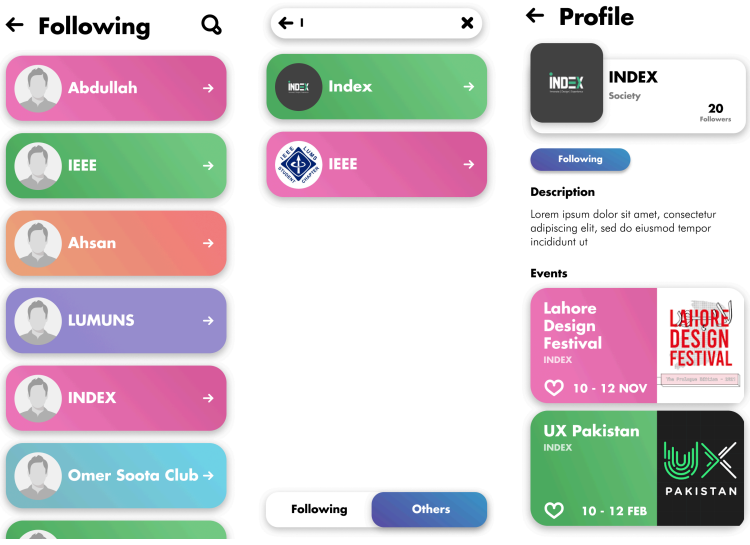


b. Friends

Goal: To allow the user to view and edit their friends' list.

UI: The user will be displayed a list of users that they have added as friends. When search mode is activated by clicking the search button, a toggle at the bottom will allow them to search for friends or for other users. A bar below the search bar will contain categories to filter the result.

A chat button (only visible if the user is a friend with this person) in the top bar will allow the user to chat with this friend.



c. Following

Goal: To allow the user to view the accounts they are following. The user should be able to unfollow the accounts they are currently following. The user should also be allowed to search all the accounts they are following.

UI: A list of followed accounts will be displayed in a scrollable view. The top bar will include a search button, the pressing of which will generate a text field for the user to input their search query, as well as a back button to go back to the menu.
In search mode, a bottom toggle will allow the users to search accounts that they are not following.
If the other user is a society, users will also be able to view events created by the society.

Relevant requirement: RQ20

d. Log out

Goal: To allow the user to log out of their account.
UI: The user will be cued to confirm whether he wants to log out of their account. On pressing 'Yes' he will be logged out and on pressing 'No' he will be displayed the previous Profile Page again.

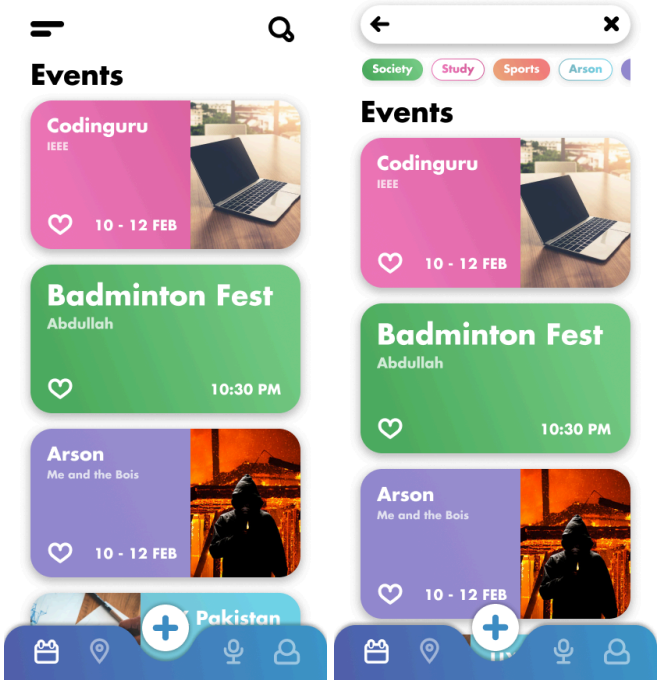
Relevant requirement: RQ4


LUMSAFAR
LUMSafar
Created By
Ahsan Sarwar
Omer Kamran
Abdullah Ahmed
Saood Bilal
Syed Tehzeer

e. About

Goal: To show the user details about the application, such as application' name, logo, and the creators' names.

UI: The above information will be shown in text form to the user, in an attractive form.



4. Events' Feed

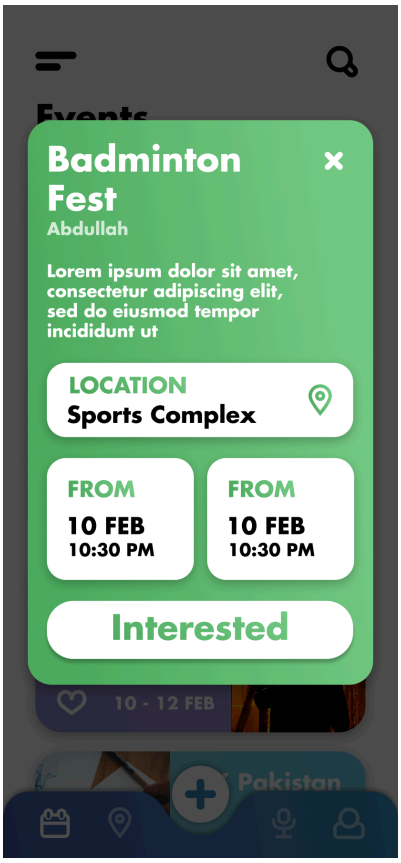
a. Default Tab

Goal: To show the list of events shared by societies.

UI: The information will be shown as a list of events along with their pictures, event creator names, dates/timings and a heart button to show your interest in the event.

The user can get further details by clicking on the event which will take the user to a new screen with further information which are location, time and interest or not.

Relevant requirement: RQ13

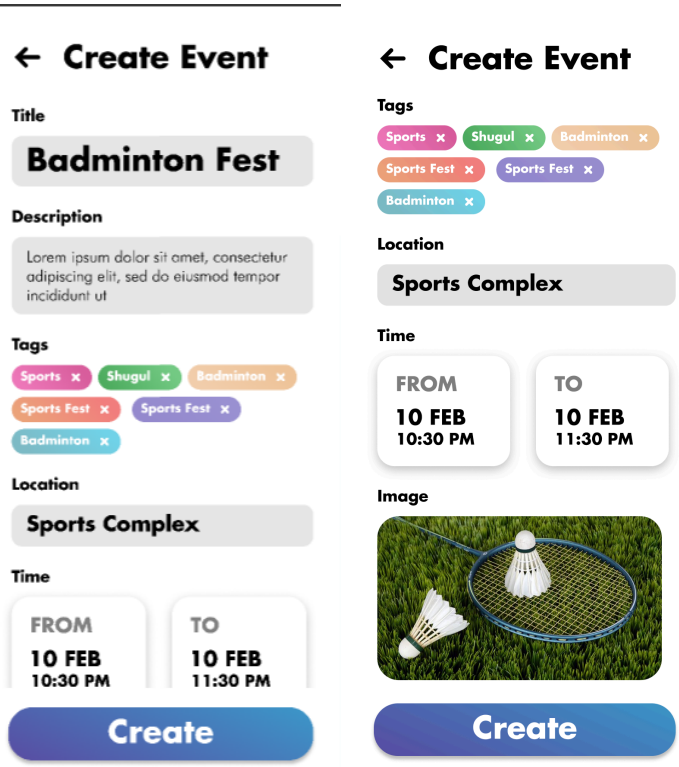


b. View an Event

Goal: To view the expanded details of an event

UI: The user will be displayed all the information, enclosed in curved rectangles.

Relevant requirement: RQ13



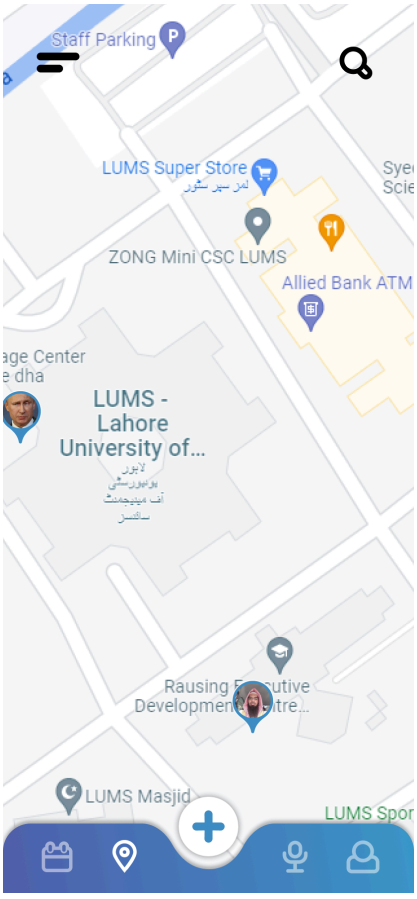
c. Create an Event

Goal: To allow the user to create an event.

UI: With a user-friendly interface, the user will be cued by labeled text fields to input information such as event's title, description, interest tags, location, time/date

and image. (The above image shows a scrollable view, with the second image shown scrolled down)

Relevant requirement: RQ9



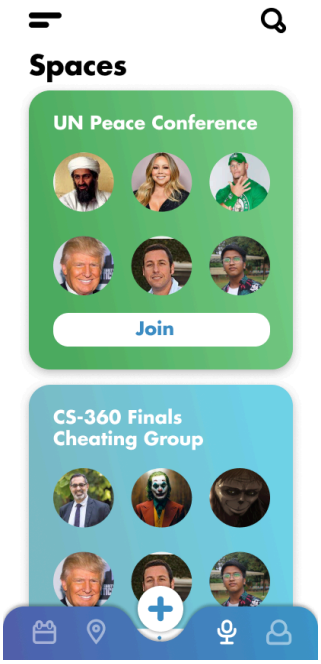
5. LUMS Map

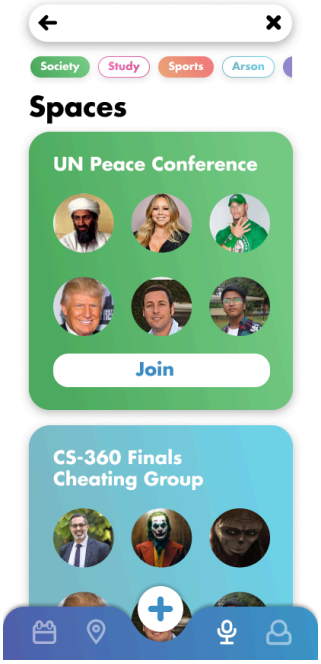
Goal: To show a google map with his and his friends' location on it.

UI: The user will be displayed his location and the locations of his friends in the form of small circles containing profile pictures of his friends' accounts. The user can zoom in or out and scroll through the whole map to see friends and the campus

The alternate map on the right is a concept of a possible collaboration between our team and The INDEX Design society.

Relevant requirement: RQ27,28





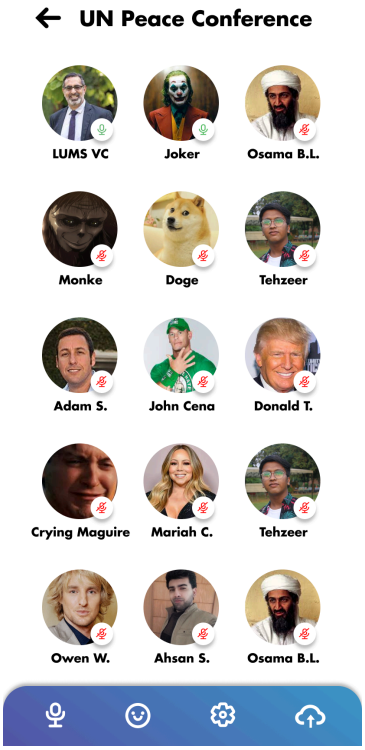
6. Talking Spaces

a. Talking Spaces' Feed

Goal: To show the the list of talking spaces made for discussions

UI: The information will be shown as a list of talking spaces along with a logo for each space and the user can scroll through this. The user can select a genre for the talk and can join it. Further, users can see a summary of who is available in the specific chat.

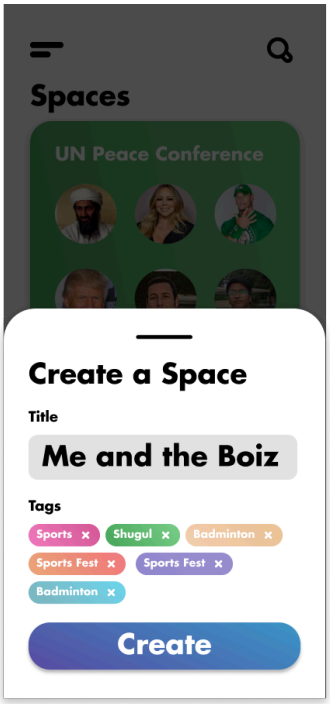
Relevant requirement: RQ22



b. Join a Space

Goal: To allow the user to join a Talking Space
UI: The user will be displayed the name of the space and the names and profile pictures of participants. If the user wants to join he may press the back arrow button, and join from the previous screen. There are buttons to allow the user to mute themselves, send emojis etc.

Relevant requirement: RQ22

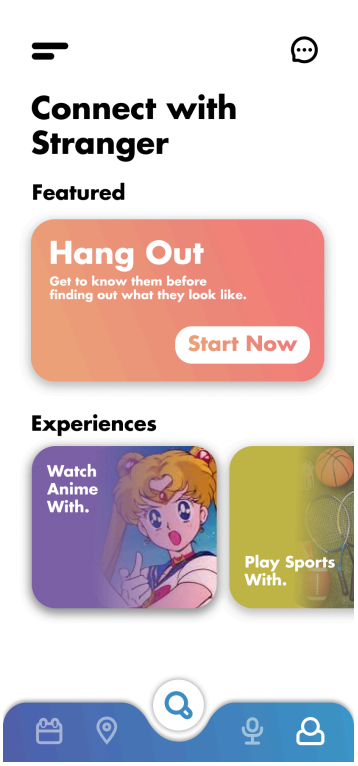


c. Create a Space

Goal: To allow the user to create a talking space.

UI: The user is cued to enter the space title, and the relevant interest tags. The interface is user-friendly.

Relevant requirement: RQ21



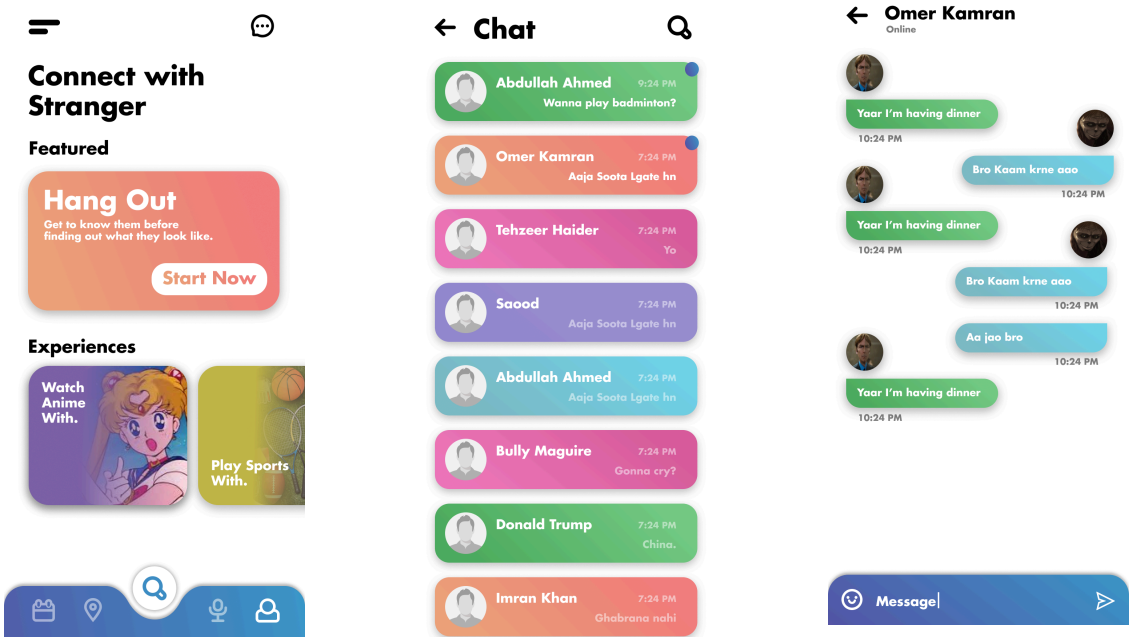
7. **Connect with Stranger**

   Goal: To match two users based on their interests

   UI: The information will be shown as it will ask the user what they are looking for and they can select Featured and Experiences. Once users select an option, it will take them to a new screen which shows the progress of searching. If another user who is also searching for someone with the same option is found, the user is taken to the 'Found Match' screen, with the matched user's profile picture and name. The user can then accept or reject this suggestion. Accepted suggestions will allow people to chat with each other.

   The user can also use the search button to match with users with similar interest tags.

Relevant requirement: RQ15,16



8. **Chat with your connections:**

Goal: Allow the user to have a conversation with their friends.

UI: On the Connect with Stranger default page, there is a clearly distinguishable Chat icon. Pressing that will open all the names of people the user has been communicating with. Pressing those will open individual conversations where in text fields, user will be cued to type his new message and send it.

5.4 User interface design rules

<Conventions and standards used for designing/implementing the user interface are stated. >

5.4.1 Overview

The user interface design follows the guidelines presented in Google's Material Design and Norman's 6 Principles of Design. In addition, the layout follows the general trends in modern mobile applications. For example, the back button and the search button are located where the users generally expect them to be. This serves the purpose of making the app easy to get into and avoid erroneous touches caused by muscle memory. All screens must display the minimum amount of information required by the user at the given moment. Information that is not required must be displayed only if the user wishes to see it.

5.4.2 Action Buttons

We use a consistent set of action-buttons across all our screens. Buttons that provide similar functionality are placed at the same location on each screen to work well with the user's muscle memory. In addition, we also make the distinction between the importance of grouped buttons. Each button group can have a maximum of one important button, which is displayed differently from the others (with a filled color in our case).

Any button which has the job of canceling, closing or going back must be shown clearly and separately.

5.4.3 Input Fields

Fields that can be edited by the users are highlighted in a color different from the background color. Each field must provide its contextual title so as to leave no space for confusion of what it means. A field with invalid input should be outlined in red, with a corresponding message explaining the error

5.4.4 Text

No text should be smaller than 12 pts. It should also follow a clear hierarchy, both in terms of size, positioning as well as color, to efficiently guide the user's eye.

We follow the following color guides for the text:

- White for all important text on non-white background
- Translucent-white for secondary text on non-white background
- Black for all important text on white background
- Translucent-black for secondary text on white background

5.4.5 Colors and Themes

No screen should display more than 5-6 colors at the same time (apart from images).

We provide the user with the ability to change the theme to their liking. As such, we have the following themes:

- Dark Theme
- Blue Theme (Default)
- Green Theme
- Others…

5.4.6 Icons

The icons must clearly communicate their intent. In case the corresponding label is not shown with the icon (for aesthetic reasons and to remove clutter), the icon itself must be self-sufficient to guide the user. Such non-text icons should only be used in places that the user becomes familiar with quickly, like the main navigation bar. Other screens which are frequented less often (settings, profile) must have an accompanying label with each icon.

The icons follow the following theming guidelines for the icons:

- White for all enabled icons on non-white background
- Translucent-white for all disabled icons on non-white background
- Black for all enabled icons on white background
- Translucent-black for all diabled icons on white background

6 Other Non-functional Requirements

6.1 Performance Requirements

1. Delivering messages between students and societies will not take more than 5 seconds.
2. Search results will not take more than 3-6 seconds to display.
3. Logging in to the app will not take more than 5 seconds.
4. Users in talking spaces will be able to hear each other with a maximum latency of 5 seconds. This is to provide a coherent communication experience.
5. Created events will be shown to the rest of the users in no more than 10 seconds.
6. The user location on the map will update at least once per minute.
7. The system won't take more than 10 seconds to find users with similar interest tags (if any).
8. Uploading changes to the profile will take no more than 5 seconds.

9. The developers will start action on any glitch or bug reports within 1 working day.
10. The LUMS map and the feed will be synchronous. When the time for an event has passed. It will be removed from the database hence resulting in it being removed from the map and the event feed.

All these requirements assume that the users have a stable internet connection. The requirements have been derived from the performance of similar apps like Snapchat, Tinder etc. We have given ourselves some leg room on top of that, since we are not that experienced.

Safety and Security Requirements

- In order to ensure the privacy of the students, they will have to turn their location off if they do not wish to be discovered. Furthermore, no one will be able to access each other's profile until they are friends. This can be improved further by receiving feedback from the students.
- In order to make the product more secure or protected, it will be made sure that the user joins only after the correct authentications. It will send an authentication token with the request, and if the user does not validate, it will be canceled. Important data like password and user details shared by the users will be encrypted to protect 3rd party interference and data leakage.
- In order to make the app more exclusive or safe(only for LUMS users), users have to log in with their campus id. This will make the users and user data more safe.
- In order to maintain the security of the app, testing will be done once or twice in a month to avoid any kind of security issue.
- In order to make the app secure,it will be locked down with API gateways to ensure that no unknown user will be able to access it.
- In order to ensure that the digital space serves as a safe platform for students to engage in healthy, constructive discussions, the developers reserve the right to ban users known to engage in misbehavior, to ensure there is no abuse of the Community Terms and Agreement by warning.

6.2 Software Quality Attributes

6.2.1 Portability

We are developing a mobile phone application. So, it can be accessed by the users from anywhere if they have an active internet connection.

6.2.2 Availability

- It will be available to all the students who have their personal campus ids as mentioned, if they have a proper internet connection.
- The personal data of the users will be stored in their local devices, so they can access it whenever they want. In order to ensure the integrity and confidentiality of user data, the system must also safeguard all data and maintain security. Filtering input and encoding data will be used to accomplish this. This ensures that user data is always accessible.

6.2.3 Usability

- The application should have an easy-to-use and understand interface that does not require any training for students to use. This will be accomplished by including toolbars, a home page, and other widgets to assist the user.

6.2.4 Robustness

- The system will run checks to ensure that all data entered into the system is correct. this information includes user phone numbers, email addresses, and a backup personal details. If something happens, users will be able to backup their data from their devices.

6.2.5 Adaptability

- The application will go through regular scheduled maintenance checks for bugs which will be fixed accordingly. Similarly, feedback from users will be taken into account, and positive changes in line with the feedback will be added to the application.

6.2.6 Testing

- Because the majority of our use cases are highly testable, we will use component testing before launching. Jest will be used to test the application. It will test all functionalities with acceptable, unacceptable, and other cases automatically. In addition, as necessary, we will use our own test cases.

6.2.7 Maintainability

- In order to maintain the application, we will be working with our specific team to work on the application. We will be working to update the website once in a while.
- Using React Native will allow us to write highly reusable and modular components, which will aid us in fixing bugs quickly as well as easily extending the functionality of the app.

6.2.8 Scalability

- The database used for the application will make it more scalable as the data in mongodb is coupled relationally, so the server will not be overburdened due to the large amount of data.

6.2.9 Security

- Mongodb is a more robust database known for best in class security and higher performance. This will make the system more secure and users will be able to use the application with complete trust and safety.

6.2.10 Compatibility

- The application will be compatible for almost all the updated mobiles unless there is someone with a very old version of OS.

Appendix A - Group Log

26 February 2022, night :

Omer, Ahsan – Made some screens on Figma

28 February 2022, 10 pm:

Saood, Ahsan – Made screens on Figma

1 March 2022, 340 pm :

Meeting with TA (Abdullah, Omer, Ahsan, Saood)

- Feedback on screens taken
- General guidance on future course of action

3 March 2022:

Abdullah, Omer, Saood, Ahsan – Worked on filling SDS document

4 March 2022:

Saood – worked on screens

5 March 2022:

Omer Abdullah – working on SDS.

Ahsan Saood – screens on Figma

6 March 2022:

Saood -Sequence diagrams

Ahsan – Figma screens

Abdullah – SDS 5.3.

Tehzeer – SDS 5.3 , section 6

7 March 2022:

Ahsan – screens on Figma

Saood – Figma

Abdullah – filling SDS, 3.1, 5.3

Tehzeer – filling SDS 5.3, 5.1, 3.3

Omer – filling SDS, diagrams on Miro, component diagram

8 March 2022:

Tehzeer – 5.2, 3.2, 4.5

Abdullah – 2.4 glossary

Omer – diagrams on Miro, component diagram

Ahsan – figma screens

Saood – diagrams

Meeting with TA (Omer Ahsan Saood)

- feedback taken on SDS document
- schema approved
- sequence diagram approved
- class diagram approved
- screens approved

9 March 2022:

Tehzeer – filling SDS

Saood – filling SDS

Omer – diagrams on Miro, SDS

Ahsan – screens on Figma, SDS

Abdullah – SDS

9 March 2022:

Meeting with TA (Abdullah Ahsan Omer Saood Tehzeer)

- Work on SDS document reviewed

Appendix B – Contribution Statement

Name	*Contributions in* this *phase*	*Approx. Number of hours*	*Remarks*
Ahsan Sarwar	*Screens, 5.1, 5.3, 5.4*	*24*	*– figma goddess* *- bro*
Saood Ahmad	*Screens,* *section 5.2* *section 5.3* *section 1* *section 2* *section 4.2* *section 4.3* *section 3.3*	*24*	*– amazing leadership skills dont know why he isnt team lead* *– i like his work ethic* *- bro*
Omer Kamran	*Section 2, Section 3, Section 4*	*24*	*Amazing at diagrams dedicated miro boy*
Abdullah Ahmad	*1.2,* *2.1, 2.2, 2.3, 2.4* *5.3* *Appendix A*	*24*	*Managed everything single handedly. Excellent work bro* *too many gym breaks* *- bro*
Syed Tehzeer Haider	*Section 3:* *3.2,3.3* *Section 4:* *4.5* *Section 5:* *5.1,5.2,5.3*	*8*	*baaz ajao bro* *too many dinner breaks* *- baaz ajao bro(2)*