College of Electrical and Mechanical Engineering

Department of Software Engineering

Undergraduate Project Document

Title: Cancer Community Support Social Media (Soul Fighting Cancer)

June, 2022

Group Members

No.	Name	ID
1	**ABDULAZIZ WENDESEN**	**ETS0011/11**
2	**DAWIT GADISA**	**ETS0296/11**
3	**ABDULHAFIZ JUHAR**	**ETS0012/11**
4	**ABDURAHIM MIFTAH**	**ETS0015/11**
5	**ABDUL AHMED**	**ETS0009/11**

Advisor Name: ADDIS ABEBE **Signature_____**

No.	Examiner Name	Signature
1		
2		
3		
4		
5		

Acknowledgement

First of all, we would like to thank God for letting us reach this moment and help us gain the wisdom we need to develop our proposed project.

Next, we would like to thank our advisor Addis Abebe who has been attending our works and mainly guiding us throughout the entire process of making the project. He has not stopped believing on us for a single second and has been motivating us to work more.

We also would like to appreciate the health centers, cancer associations and foundations, and people who work there for sharing us the vast information they have about the topic issue. Tikur Anbessa Oncology Department Head was helping us to get the information we needed sacrificing her time. Sister Mekdelawit, a nurse at Alemtsehay Breast Cancer Foundation let us in the foundations' compound and gave us the information we needed and we would like to thank both the foundation and Sister Mekdelawit. We also want to thank Sister Beza, who happened to be a nurse at Ethiopian Cancer Association, helped us a lot and told us they would expect a great work from us to help the patients.

Last but not least we would like to express our respect, love, and gratefulness for unstoppable, continues and believing support we get from our parents, family and friends.

Contents

List of Tables

List of Figures

Abstract

This document outlines a report for the design and development of a social media platform where cancer patients connect with each other to share their experience and connect with professionals to follow their treatment properly. The moving factor behind this idea is that we wanted to help cancer patients who have no social network with other patients and who are emotionally depressed. Awareness about cancer disease among the society, psychosocial impact of cancer diagnosis and treatment on the patient, Imbalance between cancer professionals and increasing number of cancer patients, and preferred social media platform are among the research problems the report attempts to address. Interviews were conducted with domain experts from which were gathered the target users for the large-scale requirements analysis, the details of which can be found on section 1.6 Methodology of this report. A survey was then conducted through interviews and the results were analyzed. The results show that there is a definite gap in the society for social media platform that works around cancer patients within the Ethiopian context, showing that the proposed system is essential for our country. According to the survey that was conducted 70% of the cancer patients are from rural areas, and these patients do not have enough knowledge about using technological devices, therefore mobile application is easier to introduce to those patients. Requirements were also gathered from this survey, in conjunction with the team's observations. A full list of requirements can be found on section 2.2 Requirements of this report. The preferred platform was decided to be mobile (android/iOS), in accordance with the conducted survey, most of the participants claim to predominantly use these platforms.

1 Introduction

1.1 Background of the Project

There is not much awareness of cancer in our country especially in the rural areas of the country. So when peoples get ill in our country having cancer doesn't even cross their minds because cancer is a disease that is only known by a very few percent of the population and the rest of the population do not know about cancer, it's symptoms, stages and other information. So people start to know about cancer after they have gone to many other clinics and get tested for many other diseases and only after they tested false for these diseases they get suggested that it is maybe a cancer by doctors of some clinics because there is only a handful facility that can diagnose cancer in our country. As a result, by the time they come to a cancer facility to be tested the cancer would have already advanced to next stage and this makes it much more difficult to treat and the chance of the treatment curing the cancer will also decrease.

After people are diagnosed with cancer the next step is to start the cancer treatment and while being treated only few patients finish the treatment and the rest of the patient quit midway of the treatment. The main reason for this is that the patients will also have depression problems after they know they have cancer and the treatment of cancer is also very long and hard, so only providing the treatment is not enough for this patients they also need to be provided with a consistent emotional support to continue the treatment and actually be optimist towards the treatments effectiveness.

So after seeing these problems in our research we came up with the idea to build a platform that creates awareness about cancer in our country and worldwide. We also believe that it provides a way for patients to get emotional support from professionals and also other patients who are going through the same condition as they are going through. It will be a well-designed platform and provide information about actual cancer communities that give both physical and emotional support to these patients.

1.2 Statement of the problem

1.2.1. **Existing Systems**

Some platforms were being developed internationally in the past times which perform almost the same as ours, but with some kind of limitations, gaps, and defects. We have tried to see different kinds of platforms and identified the gaps in those platforms. Some of the existing systems are viewed below.

LivingWith: Cancer Support: free on iTunes: A "**My Circle"** function gives patients a simple way to communicate with the family members and friends who matter most to them. It has a feature that allows patients to request someone manage the app on their behalf when needed. A "Requests" function allows patients to send requests for help with daily tasks like meals, rides to doctors' appointments or other support. A "Health Notes" function records key takeaways from doctors' visits, and tracks notes and questions in between appointments to improve communication with healthcare teams. It stores test results, medication details and insurance documents in one central folder. And a "Living With" function allows patients to track mood,

pain, sleep and counts steps. Users can sync data with other fitness apps and wearable. Patients choose which information to share in personalized graphs and reports regarding their wellness for better communication with doctors and support personnel. It's important to remember that these tools are not meant in any way to replace a doctor or specialist, but they can bring relief, organization and help connect to resources and people who wish to provide support to cancer patients.

Cancer.net: Free for iPhone, iPad and Android. The app includes features that allow patients to get up to date information on more than 120 types of cancer, log and track their treatments, receive advice on how to manage side effects, provide cost of care information, and connect to links for cancer-related podcasts, videos, and blogs. This app is available for Spanish speakers as well and allows patients to track symptoms, log medication and facilitate communication between doctor and patient.

CancerConnect: combines original content and news with expert doctor commentary from many leading health providers with editor curated content from the web to ensure users have access to the most relevant and through educational content available to support their decision making. By combining information with a social community CancerConnect provides cancer patients and their caregivers a unique destination to seek information, support, and inspiration.

CancerConnect is used by the top U.S. cancer centres which allow over 50,000 members to connect with other patients being treated by some of the best cancer doctors where they can learn about the most innovative treatments.

CancerConnect provides information and support from diagnosis through survivorship. There are currently over 60 CancerConnect "groups" including leukemia, lymphoma, breast, colon, lung, and prostate cancer and rare tumors.

CancerSupportSource app: is a free app that was created to empower people living with cancer to rate and track concerns over time and have the ability to accurately report progress to their health care team. In addition, the app assists users in locating a nearby Cancer Support Community affiliate and the option of joining the Online Community to network with others who are experiencing many of the same things that are challenging you. You also get links to valuable cancer and cancer caregiving articles.

Belong Beating Cancer Together: is the world's largest social and professional network for managing and navigating cancer treatment. Belong aims to improve quality of life and quality of care by connecting patients, caregivers, and leading healthcare professionals. The app is free and anonymous. The app provides people with cancer and their families a unique and innovative personalized solution to help manage and fight cancer more. The patients can join to the groups that are already created by the admins of the app and can post texts, photos, and videos that they believed can help the cancer community and can comment, like, and share those posts like other social media platforms. You can also store your information like blood type and others in files so that they can be used to detect cancer stages and other treatment issues.

These are the existing systems we have found which have been developed by different developers in order to support cancer patients support with each other and share their experiences across the world.

1.2.2. **Major Problems of Existing Systems**

The existing systems are reliable in delivering information about the cancer disease and connecting the patients with professionals, but we do not see these systems as a social media where cancer patients get together and share their experiences. The existing systems are more like a digital cancer care system where cancer patients track their treatments and focus more on the medication process. But we are working on a system that can help figure out the problem from the root, as it can help cancer patients emotionally to believe that they can survive from the disease by sharing experiences with other patients and professionals.

Some of the systems do not have enough features to help patients get together and share their experiences to improve their quality of life and survival. On some of the existing systems mentioned above, the patients or professionals do not get to create support groups to connect with. They just simply join to the groups created by the admins of the app or system which do not give the ability of being free in your connections with others. Such systems are more likely focused on providing information and treatment advices on different kinds of cancer types. Existing system like CancerConnect is not much of user intuitive and does not let the community participate a lot. The users can only join to groups that are already created by the centre and only chat in private. They are not able to host virtual conferences and do not give them that much chance of supporting with each other. This platform is only based on by a single cancer centre which is U.S. cancer centre.

We believe that enough feasibility study has been made developing these existing systems on cancer, as much of them do not concern the age of the patients, locations of the patients (where they come from), languages they talk, literacy of the patients and others that could have been helpful to make the system more user friendly and reach out to lot of patients.

1.2.1 Proposed System

Before directly diving into the description of the system we are proposing to build, it is better to describe some points about our topic of study which is concerning on **cancer** and mainly on **cancer support groups**.

Cancer is a disease in which some of the body's cells grow uncontrollably and spread into other parts of the body. Cancer starts almost anywhere in the human body made up of trillions of cells. [1] According to the information found from the National Cancer Institute, one out of every two men and one out of every three women suffer from Cancer. In Ethiopia based on the information of Ethiopian Cancer Care Institute more than 70000 people diagnose with different kinds of cancer types every year.

Cancer support groups are meetings for people with cancer and anyone who cares about the disease. They can have many benefits. Even though a lot of people receive support from family and friends, the number one reason they join a support group is to be with others who have similar cancer experiences and got out of huge stress and depressions. Some research shows that

joining a support group improves both quality of life and survival. Support group help the patients feel better, more hopeful and not so alone, give them a chance to talk about their feelings and work through them, help them deal with practical problems, such as problems at work or school, and help them cope with side effects of treatment. [2]

Cancer supports develop the bond of social networks among the patients. Social networks defined the network of social relationships that surround an individual and the characteristics of those bonds. [3]

In a meta-analysis of 87 papers, larger social networks were found to be meaningfully connected with lower cancer mortality. [4]

Concerning the basic use of social networks for cancer patients, we are proposing to build a platform that connect cancer patients from all over the world to share their experience and support with each other to improve the quality of life and survival. Our proposed platform will be a social media only for cancer patients, care givers, professionals and others who are related with the disease.

The platform will have different kinds of functionalities that can help the patients. The users can create support groups and meet up in video conferencing from different parts of the world.

The users also can chat with each other and the patients can ask professionals about their type of cancer and treatments need to take. The users can join in different groups that are created by other users of the platform. The users can also post texts, images, and videos that they believe can motivate the patients to fight the disease more. The users can like, share, and comment on the posts of other fellow users the platform. The users can also create groups and channels with common interests and same cancer experiences to share information and posts.

The platform can also help patients remember treatments or check-up dates summarize drug administration protocol, deliver medication and Doctor's visit reminders.

We believe that this system will help a lot of cancer patients to improve their quality of life and survival by increasing their mental strength and make them believe that they can battle the disease together as a team with other cancer patients and professionals.

1.2.2 Advantage of Proposed System

- It provides a clear information for users to find cancer support groups that are available in our country
- It allows the users to communicate remotely through online video conference
- It allows the users to easily join support groups and improve their quality of life
- It facilitates the communication between patient, care giver, professional and volunteers to work together
- It helps the patients to have a good social relationship

- It helps the patient to gain hope and emotional strength when they see others in the group who are further along their road to recovery and who have made great strides toward having happier and healthier lives
- It allow the patients to share their experience easily
- It facilitates mentorships between members with similar diagnosis that can lead to peer-support
- It allows the patients to get treatment advises and ask anything related with their disease from registered professional
- It allows the caregivers to get different kinds of emotional supports from the cancer community
- It allows the caregivers to get information or advice on how to take care of the patients
- It allows the patients or caregivers to schedule treatment and medication times and get notified

1.3 Motivation

We know that cancer is called as one of World's killing disease that is difficult to protect with medical treatments. According to the information found from the National Cancer Institute, one out of every two men and one out of every three women suffer from Cancer. In Ethiopia based on the information of Ethiopian Cancer Care Institute more than 70000 people diagnose with different kinds of cancer types every year.

As it is very easy to notice, nothing much is done in our country Ethiopia regarding this disease. We have seen the patients suffer from the disease in different parts of the country where they do not get much of a support from their community. We noticed there isn't enough awareness that has been distributed in the community which indirectly affects the patients not to get to health centers early. The disease has been taken as a curse from God for years and is still taken as a curse in most parts of the country.

We have seen The GLOBOCAN Cancer Observatory summary statistic 2020 information on number of new cancer cases and deaths in Ethiopia. The number of new cases goes high up to 77352 and number of deaths reaches about 51865 annually which is a big number for a single nation. So a citizen of this nation, we were motivated to do something helpful to the cancer community in order to create awareness to improve quality of life and survival of the patients.

We believe that disease is in subconscious mind which everyone can fight it by spreading positive energy to their mind and help with each other. We have made some researches in some association and health centres working on cancer, and they told us the luck of information among the community about cancer even make the disease worse. So we believe that this proposed system of ours helps a lot in distributing awareness about cancer for the community and increase the integrity of the patients with one another in order to survive from the disease.

1.4 Scope and Limitation

The scope of this study would be confined to Addis Ababa by taking three health centers and cancer associations namely Tekur Anbessa specialised hospital department of oncology, Alemtsehay breast cancer association (The pink house project) and Ethiopian cancer association. Generally, the study is limited to examining the nature of information communication between patients as well as patients with their health care professionals and also steps taken by cancer patients to follow up their cancer treatment there by identifying the root causes of achieving supportive care for cancer patients at current time 2014 EC. It focuses on studying the impacts of psycho-social support, information support, enhanced patient-physician relationship, informing upcoming cancer related clinical trials and cancer community support groups on cancer patients. Furthermore, this study does not intend to substitute in-person support or in-person cancer treatment rather create a portal to help patients communicate with their health care professionals and other patients like them, get published information about their disease by verified health centers, get the best health centers by their rating and also schedule their meals and day to day life activities with accordance to their treatment.

This study actually needs special attention as it is related to health. Such study by its nature requires long period of time, adequate budget and large health center coverage on country side. Due to these constraints the study is limited only to 3 health centers and cancer associations in Addis Ababa which are mentioned above. Since we had limited access to health centers and cancer associations' internal documentation and patients' lack of willingness for interview make it difficult to implicitly understand the patient's treatment hence limit the amount of gathered data to some extent.

The system we proposed is focused on developing mobile application that will be available for both android and ios users. Within the given time space to develop the system which is applied on multiple platform is difficult and make the project development more complex because the developed system for different platform need to be integrated to work properly, Therefore the system will not have a desktop and web version. The other and major limitation to this study is patients' literacy problem. As most of cancer patients are more than age 35 and lives in country side they don't know how to use smartphones and they don't have internet access mostly.

1.5 Objectives

1.5.1 General objective

The general objective of this project is to develop a social media application that connects cancer patients, caregivers, health center, and health professional all together and enable the patients to support each other, gain more awareness about cancer and its treatment, and improve their quality of life and survival while battling cancer.

1.5.2 Specific objective

To achieve the general objective of this project the following specific objectives are set:

- ✓ Identify the major problems in existing system.

- ✓ Understanding each stakeholder requirement.
- ✓ Gather all the functional and non-functional requirements.
- ✓ Choose the best design and development method that fits the requirements.
- ✓ Design the new system
- ✓ Implement the new system
- ✓ Test the system on different scenarios.

1.6 Methodology

From the beginning to the end of our project we will be using a lot of different methods and tools. The software process model that we are using to develop this project is Agile and waterfall software process model. We use waterfall for planning and design and the development is carried out in agile style. We use the combination of both because we have time constraint and we want the iterative nature of the agile method.

1.6.1 Data collection methodology

The method that we used to gather data and requirement is interview. We have gone to three places to gather data which are Tikur Anbessa hospital, Ethiopian cancer association and pink house association. We asked nurse mekdelawit from pink house association and nurse beza from Ethiopian cancer association and the head of oncology centre in Tekure Anbessa questions related to cancer patients and other topics and we use voice recording tools in our mobile to record the answers. We were able to gather information regarding many things and the overall requirements. In addition to this we used document analysis to get some numerical information about cancer.

1.6.2 System design and analysis tool

The tools that we are going to use are:

- ✓ Microsoft Office - to prepare documents like requirement specification and other documents including this one we have used Microsoft word 2019.
- ✓ Enterprise Architecture - to prepare a simplified unified modelling language (UML) design of the proposed system.
- ✓ Adobe XD and Adobe AfterEffect - to design the user interface.
- ✓ HIPO (Hierarchical Input Process Output) Diagram - to document the relationships between modules and the data flow within them. We will decompose each component to its constituent parts

1.6.3 System development tools

We are going to use different kinds of development tools, Programming languages, and styling frameworks in the system development process.

- ✓ For the frontend, we are going to use Flutter frame work for building user interface. We used Flutter because its single code base meaning it works on both IOS and android also it could be translated in to web. And the other reason we used flutter is with flutter the responsiveness will increase and it takes less storage space.

- ✓ For backend side, we are going to use Node.js and socket IO. We choose node because it is fast, scalable and it is employed easily in the server-side proxy and it has a lot of API's that help us.
- ✓ For the database, we are going to use MYSQL which is relational database. We choose MYSQL because it's has robust transactional engine that supports a lot of transaction and it offers unmatched scalability in addition to this it is simple and easy to use.
- ✓ For version control we are going to use Git/GitHub
- ✓ For software development toolchains we are going to use VsCode and Android Studio as our IDE
- ✓ For planning and executing tasks within a team we are going to use Trello

2 System Requirement Specification

2.1 Background Overview

2.1.1 The Area of Investigation

The primary area of investigation of this project is information exchange platform for cancer patient within the context of Ethiopia. It is an attempt to create a social network for cancer patient.

2.1.2 Context

The problem, as gathered by our team, is that there is no easily accessible platform for cancer patient to communicate, share experience and to motivate each other in Ethiopia, There is not much awareness of cancer in our country specially in the rural areas of the country, so when peoples get ill in our country having cancer doesn't even cross their minds because cancer is a disease that is only known by a very few percent of the population and the rest of the population doesn't know about cancer, it's symptoms, stages and other information so this people start to know about cancer after they have gone to many other clinics and get tested for many other diseases and only after they tested false for this diseases they get suggested that it is maybe a cancer by doctors of some clinics because there is only a handful facilities that can diagnose cancer in our country, so by the time they come to a cancer facility to be tested the cancer would have already advanced to next stage and this makes it much more difficult to treat and the chance of the treatment curing the cancer will also decrease.

2.2 Requirements

This is the part of the document that details the requirements of the project and the requirements elicitation process used to arrive at these requirements.

2.2.1 Approach

The approach used in the requirement elicitation was that of target users selection, information gathering and analysis. To this end we concluded on our research methodologies which were the basis for our overall project plan, including deployment platforms and key features.

2.2.2 Methodology

As listed in chapter one, we used some of research methodologies. We used these methodologies for requirement elicitation. These methods are Surveys, Observation and interviewing potential users.

2.2.3 The Stake Holders

Cancer Patients: These are the primary users of our platform. They will post, chat, and follow other users in the platform.

Health Professionals: These are the users of our platform that have been verified as professionals on the cancer field.

Caregivers: These are the users of our platform that are family member or paid helper who regularly looks after the patient.

Health Organizations: These are the users of our platform that provide cancer related information and create awareness amongst the users.

Cancer Community Associations: These are the users of our platform that help cancer patients go through the cancer treatment and provide service that are mandatory for the patients.

2.2.4 Requirement General Overview

The system to be built will have these general requirements

- The system will allow the cancer patients to join support groups
- The system will allow the health professionals to help the cancer patients
- The system will allow the caregivers to gain information on how to take care of patients
- The system will allow the health organizations to post cancer related informations
- The system will allow the cancer community associations to provide services to patients

2.2.5 Requirements Legend

ReqId: a unique requirement ID that we generated for each requirement

Priority: HIGH/ LOW, represents the urgent-ness of the requirement and the severity of its impact on the system

Requirement: is the actual requirement definition for the system

Referenced Requirements: are all the requirements that are referenced within the definition for the current requirement, or all requirements that could clarify the meaning of any word or term referenced within the current requirement

ReqType: FUNCTIONAL/NON-FUNCTIONAL is the type of the requirement that is currently being defined.

2.2.6 Requirements Specification

The Requirements of the system are specified as follows

Table 1: The requirements specification for both Functional and Non-Functional

Req ID	Priority	Requirement	Referred Requirement	Requirement Type	Actor
U001	HIGH	A user must sign up to use the platform		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U002	HIGH	A user must insert their user name, password and email to sign up		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U003	HIGH	A user must sign in to access their account		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U004	HIGH	A user must enter their email and password to sign in		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U005	HIGH	A user can edit their profile picture, username, User type, email, password		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U006	LOW	A user can recover their password if they lost it		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community

					Associations
U007	HIGH	A username must be unique		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U008	LOW	A user can upload a profile picture		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U009	LOW	Profile pictures shouldn't contain explicit content		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U010	LOW	Users can report suspicious / harmful / inappropriate posts		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U011	LOW	Users can report other users that post inappropriate contents		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U012	HIGH	A user can edit their profile information at any time		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U013	LOW	A user can block another user from interacting with them		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U014	HIGH	A user can share posts with other users both on the platform and other platforms as a link		NON - FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations

U015	HIGH	A user can see their chat history		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U016	HIGH	A user can check their inbox		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U017	HIGH	A user can choose their preferred language		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U018	HIGH	A user can search for another users and groups		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U019	HIGH	A user can create, read, update and delete any private message and private message session that belongs to them		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U020	HIGH	A user can follow other users		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U021	HIGH	A user can join, leave and host private or public video conference		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U022	HIGH	A user can unfollow followed users		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U023	HIGH	A user can post anonymously		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers,

					Health Organizations, Cancer Community Associations
U024	HIGH	A user can post of picture type, text type, rich text type, video type,		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U025	HIGH	A user can give a comment on a post		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U026	HIGH	A user can edit a post		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U027	HIGH	A user can delete a post		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U028	LOW	A user can like a post		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U029	HIGH	users can chat by text message and video both In private and in public		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
U030	HIGH	Users can attach files during chat		FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
A001	HIGH	The admin can block a user from accessing the system temporarily or indefinitely		FUNCTIONAL	Administrator

A002	HIGH	The admin can view reports made on the users and posts		FUNCTIONAL	Administrator
A003	HIGH	Admin can do everything user can do	U001-U039	FUNCTIONAL	Administrator
S001	HIGH	The system must be protected from security threats		NON - FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations
S002	HIGH	The system must be reliable		NON - FUNCTIONAL	Cancer Patient, Health Professional, Caregivers, Health Organizations, Cancer Community Associations

2.3 Feasibility Study

2.3.1 Operational Feasibility

Systems that will be developed should be easy to use and responsive to the users. In order to make that happen, the User Interface design should be user friendly and simple. Our proposed system will be having a simplistic UI design which will be easy for users to use and will not need literacy level of the users. The system will have a video tutorial how to use the system for those users who face difficulties in using it.

2.3.2 Economic Feasibility

Our proposed system is economically feasible as it is a social media platform like other social media platforms that people are already using. We know that most of smart phone and device users use different kinds of social media platforms working with internet which is not expensive to pay for the internet services they get. The users of our proposed system only need an internet connection to use it.

Instead our system brings an income for different institutions and companies that will be using it. The users, mainly cancer patients can also get connected with people who experience the same issues and learn from each other with no much money to spend to get such a great service.

2.3.3 Technical Feasibility

Technically, the newly proposed system will be simple to be use by low skilled users and easily accessible by the people who can use the system. As we have mentioned above there will video and textual tutorials will be provided to help the users interact with the system well.

2.3.4 Legal Feasibility

The proposed system will not have any conflict with any government rule and regulation, and with any cultural aspects or norms. This will be simply a place where cancer patients get strengthened emotionally and where they help out each other. Instead the proposed system will

work fine with and help the Health Institute of the country's government. So the proposed system will be legally feasible.

3 System Analysis and Modeling

3.1 Overview

This chapter will describe and explain how the requirements mentioned above are interlinked and modeled in the system.

3.2 Scenario Based Modeling

3.2.1 Use-case Identification

Table 2: Use-case Id, name, description

Use-case ID	Use-case Name	Brief Description
UC001	Sign up	Allows users to register using their phone number, email, or other social media accounts
UC002	Sign in	Allows users to be authenticated in order to access resources on the platform
UC003	Choose user type	Allows users to choose user type like patient or health professional when registering on the system
UC004	CRUD Profile	Allows users to create, Read, Update, and Delete their profile
UC005	Forgot password	Allows users to recover their account if they forgot their password
UC006	Verify account	Allows users to verify their account on the email they registered with
UC007	Invite friend	Allows users to invite their friends from their contact list and other social media platforms
UC008	View new feed	Users are allowed to view new posts on the platform
UC009	Post information	Allows users to post text, image, and video information on the platform
UC010	Like post	Allows users to like the posts of other users
UC011	Comment on post	Allows users to comment on posts of other users

UC012	Follow user	Allows users to follow other users
UC013	Un-follow user	Allows users to un-follow other users they have been following
UC014	Send message	Users are allowed to chat with other users privately and publicly in a group
UC0015	Video chat	Allows users to chat with live video call
UC0016	Attach file	Allows users to send text, image, video and other files to other users
UC017	Report user	Users are allowed to report a user who posts inappropriate contents
UC018	Report post	Allows Users to report posts to be deleted if they have inappropriate contents
UC019	Block user	Allows users to block other users from being their friends
UC020	Ban user	Admins are allowed to ban users who have been reported many times from using the system
UC021	Host live video conference	Allows users to host live video conferences where everyone can join in or can also be private
UC022	Join video conference	Allows users to join online video conference hosted by other users
UC023	Leave video conference	Users can leave the live video chat they are attending if they do not like the content of the conference
UC024	View report	Allows admins to view the reports made by users of the system
UC025	View chat history	Allows users to view their chat history
UC026	Check notification	Users can check for new information that they have been notified
UC027	Check inbox	Users can check their inbox if new private messages are sent for them

UC028	Choose language	Users are allowed to choose the language they want to display the platform
UC029	Search for user	Users can search for another users using their user names of the platform in order to make the searching process easy
UC030	CRUD private message	Users can create, read, update and delete any private message and private message session that belongs to them

3.2.2 Actor Identification

Table 3: Actors that could interact with the system

Actor ID	Actor Name	Description
A001	Admin	A named entity who controls the whole platform and response requests of the users
A002	Moderator	A named entity who is employed to monitor and resolve issues within the platform
A003	Cancer Patients	A named entity who are users of the platform that requests for different services and connect with other patient users
A004	Health Professionals	A named entity who are users of the system and volunteer to help the patients in answering health related questions and give advice about Cancer
A005	Care givers	A named entity that are also users of the system and interact with the system in order to get information about Cancer to treat their relatives or friends with cancer
A006	Health centers and Cancer associations	A named entity that interact with the system in order to post useful information about cancer
A007	Others	A named entity that interact with the system in order to

	give services and help cancer patients voluntarily

3.2.3 Use-case Mapping

Table 4: Use cases and the actors that own them

Use-case ID	Use-case Name	Use-case Actors
UC001	Sign up	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers, Admin, Moderator
UC002	Sign in	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers, Admin, Moderator
UC003	Choose user type	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers
UC004	CRUD Profile	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers
UC005	Forgot password	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers
UC006	Verify account	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers
UC007	Invite friends	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers
UC008	View new feed	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers
UC009	Post information	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers
UC010	Like post	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers
UC011	Comment on post	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers
UC012	Follow user	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers
UC013	Un-follow user	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers
UC014	Send message	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers
UC0015	Video chat	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers
UC0016	Attach file	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers
UC017	Report user	Cancer patients, health professionals, care givers,

		health centers and cancer associations, volunteers
UC018	Report post	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers
UC019	Block user	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers friends
UC020	Ban user	Moderator, Admin
UC021	Host live video conference	Cancer patients, health professionals, care givers, health centers and cancer association
UC022	Join video conference	Cancer patients, health professionals, care givers, health centers and cancer associations
UC023	Leave video conference	Cancer patients, health professionals, care givers, health centers and cancer associations
UC024	View report	Admins, moderator
UC025	View chat history	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers
UC026	Check notification	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers
UC027	Check inbox	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers
UC028	Choose language	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers, Admin, Moderator
UC029	Search for user	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers
UC030	CRUD private message	Cancer patients, health professionals, care givers, health centers and cancer associations, volunteers

3.2.4 Use-case Diagram

3.2.4.1 Admin/ Moderator Diagram

This diagram shows the actions that can be taken by a moderator/admin within the system. Moderators are appointed by the system owners and can be demoted at any given time without interaction to the system.

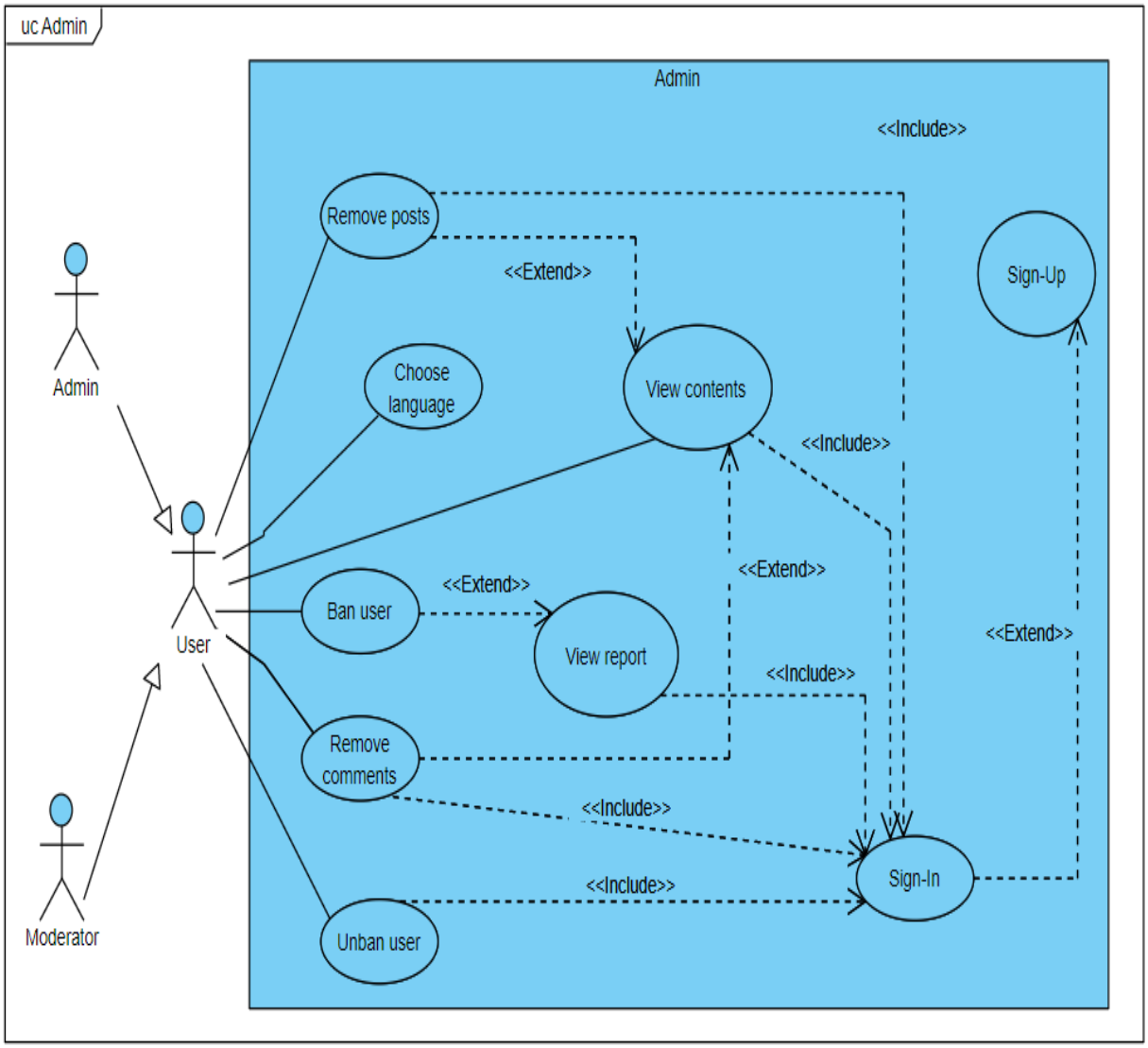

Figure 1: Admin Use case Diagram

3.2.4.2 Platform Diagram

This diagram shows the common interactions a user has with the system. The user can be of type patients, health professionals, care givers, Health organizations, and others.

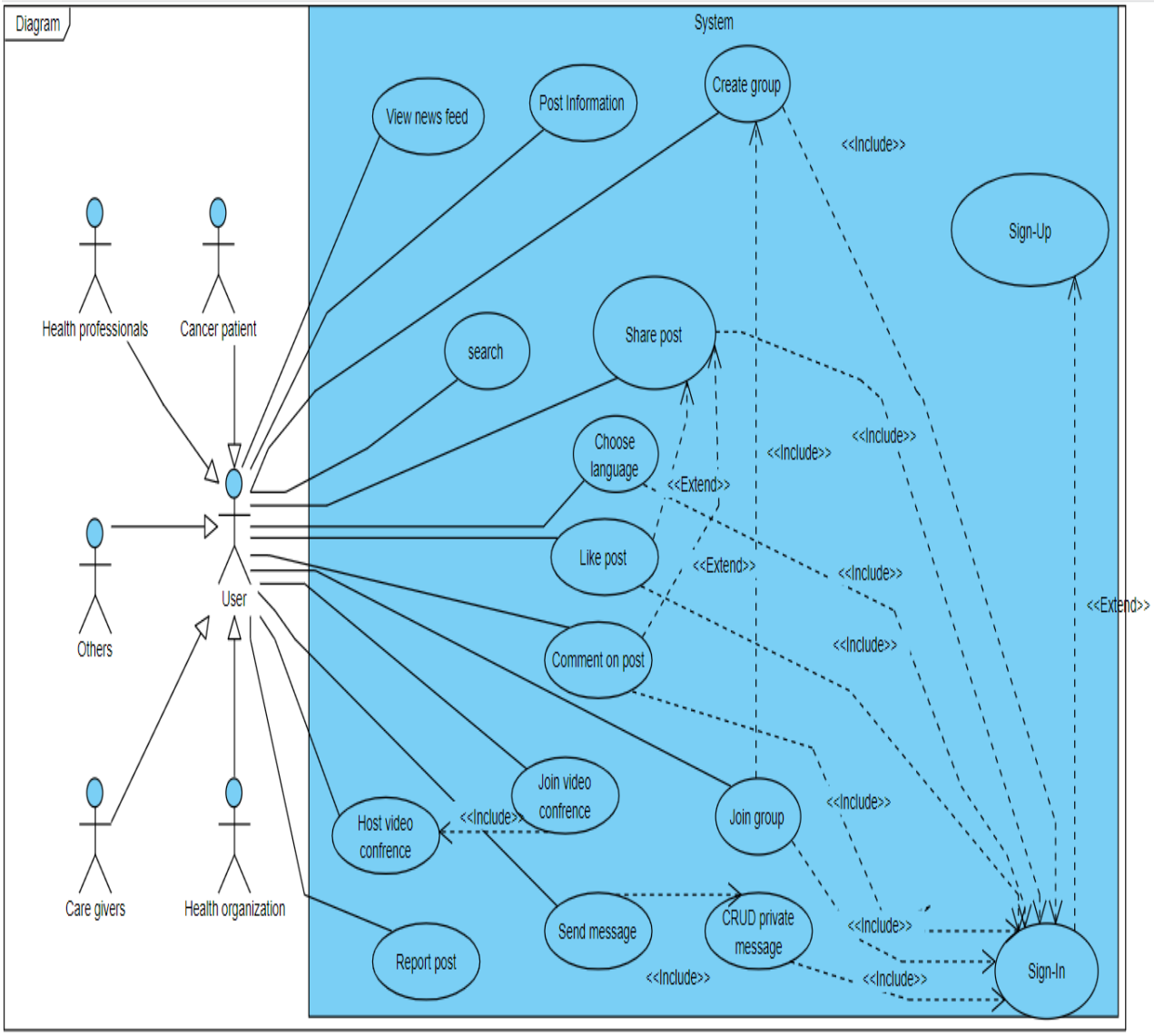

Figure 2: Platform Use case Diagram

3.2.5 Use-case Description

Table 5: UC001 description

Use-case ID	**UC001**
Use-case Name	Sign up
Description	Allows users to register using their phone number, email, or other social media accounts.

Trigger	Users request to register on the system.
Actors	user
Preconditions	User must reside in Ethiopia; User must have a valid phone number and/or social accounts
Main flow	1. user enters necessary information 2. user submits necessary information 3. system requires users email account 4. user submits email account 5. system sends a verification message in email account 6. user verifies the account 7. system creates the account for the user
Alternative flow	None
Exception Conditions	Email verification failed
Post conditions	User registered

Table 6: UC002 description

Use-case ID	**UC002**
Use-case Name	Sign in
Description	Allows users to be authenticated in order to access resources on the platform
Trigger	Users request to sign in
Actors	user
Preconditions	User must have an account on the platform
Main flow	1. user enters username and password 2. user submits username and password 3. system validates username and password 4. system verifies username and password 5. system displays the user's homepage 6. the use case ends
Alternative flow	3a. Missing username and password 1. the system requests email address 2. system sends a code on email to the user to reset password 3. use case resumes at main flow step 1 3b. Maximum 3 attempts exceeded 1. system displays "Maximum attempts exceeded, try to contact administrator" message 2. the system locks user account 3. the use case ends
Exception Conditions	Email verification failed
Post conditions	User signed in

Table 7: UC003 description

Use-case ID	**UC003**
Use-case Name	Choose user type
Description	Allows users to choose user type like patient or health professional when registering on the system
Trigger	Users request to choose the type to be registered on
Actors	user
Preconditions	User must reside in Ethiopia; User must have a valid phone number and/or social accounts
Main flow	1. user submits type of user to be registered on 2. system asks questions related with the user type chosen 3. user submits the needed information 4. system registers the user on the chosen user type 5. system displays suggestions based on the user type 6. the use case ends
Alternative flow	3a. Missing listed user types 1. the system suggests "others" preference for the user 2. user chooses "other" option if listed users is not the type the user wants to be registered on 3. use case ends
Exception Conditions	User type doesn't match with the displayed content
Post conditions	User registered on the chosen type

Table 8: UC004 description

Use-case ID	**UC004**
Use-case Name	CRUD Profile
Description	Allows users to create, Read, Update, and Delete their profile
Trigger	Users request to create, read, update, and delete their profile
Actors	user
Preconditions	User must have an account
Main flow	1. user requests to create, read, update, and delete their profile 2. system displays the information in user profile 3. user perform the CRUD on specific topic on the profile 4. system submits the users request 5. system displays the profile after CRUD is performed 6. the use case ends
Alternative flow	None
Exception Conditions	Account is not created
Post conditions	CRUD profile is performed

Table 9: UC005 description

Use-case ID	**UC005**
Use-case Name	Forgot password
Description	Allows users to recover their account if they forgot their password
Trigger	Users request to recover their account
Actors	user
Preconditions	User must have an account
Main flow	1. user requests to recover account 2. system submits user request 3. system request user email address 4. user submits email address 5. system sends code on user email address 6. user enters the code 7. user submits code 8. system requests user to reset password 9. user submits new password 10. system displays user's homepage 11. use case ends
Alternative flow	None
Exception Conditions	Account is not created
Post conditions	Account is recovered

Table 10: UC006 description

Use-case ID	**UC006**
Use-case Name	Verify account
Description	Allows users to verify their account on the email they registered on with
Trigger	Users request to verify their account
Actors	user
Preconditions	User must have a valid email address
Main flow	1. user submits email address 2. system validates email address 3. system sends verification message on user's email 4. user submits verification 5. system verifies user 6. use case ends
Alternative flow	None
Exception Conditions	Email address is not valid, account verification failed
Post conditions	Account is verified

Table 11: UC007 description

Use-case ID	**UC007**
Use-case Name	Invite friend
Description	Allows users to invite their friends from their contact list and other social media platforms
Trigger	Users requests to invite a friend to the platform
Actors	user
Preconditions	User must have an account and sign in
Main flow	1. user requests to invite a friend to the platform 2. system displays user's contact and social network options 3. user submits friends account from contact or social network 4. system sends friend invitation to the other user selected 5. system displays other user's response 6. use case ends
Alternative flow	None
Exception Conditions	Friend invitation not sent
Post conditions	Friend invitation received

Table 12: UC008 description

Use-case ID	**UC008**
Use-case Name	View new feed
Description	Users are allowed to view new posts on the platform
Trigger	Users requests to view new news feeds and posts
Actors	user
Preconditions	User must sign in to the system
Main flow	1. user requests to view new news feeds 2. system refreshes the page 3. system displays new news feeds and posts 4. use case ends
Alternative flow	None
Exception Conditions	No new feeds
Post conditions	New feeds displayed

Table 13: UC009 description

Use-case ID	**UC009**
Use-case Name	Post information

Description	Allows users to post text, image, and video information on the platform
Trigger	Users request to post new information on the platform
Actors	user
Preconditions	User must sign in to the system
Main flow	1. user enters the data to be posted 2. user submits the data 3. system checks for type of data 4. system displays the entered data 5. use case ends
Alternative flow	3a. check for type of data 1. "Enter only text, image, and video files" message displayed 2. system requests the user to re-submit the file if the file is not the type listed above 3. system displays the file 4. use case ends
Exception Conditions	Unable to post this type of file
Post conditions	New post is displayed

Table 14: UC010 description

Use-case ID	**UC010**
Use-case Name	Like post
Description	Allows users to like the posts of other users
Trigger	Users request to like the posts on the platform
Actors	user
Preconditions	User must sign in to the system
Main flow	1. user opens the post 2. user requests to like the post 3. system submits the like request 4. system displays the liked post 5. use case ends
Alternative flow	None
Exception Conditions	Unable to like this post
Post conditions	Liked post displayed

Table 15: UC011 description

Use-case ID	**UC011**
Use-case Name	Comment on post
Description	Allows users to comment on posts of other users

Trigger	Users request to comment on the posts of the platform
Actors	user
Preconditions	User must sign in to the system
Main flow	1. user opens the post 2. user requests to comment on the post 3. system provides a field for users to write the comments 4. user enters the text comment 5. user submits the comments 6. system checks for the type of file of the comment 7. system validates the comment 8. system displays the comments on the posts 9. use case ends
Alternative flow	None
Exception Conditions	Unable to send this type of file on the comment
Post conditions	Comments displayed on the posts

Table 16: UC012 description

Use-case ID	**UC012**
Use-case Name	Follow user
Description	Allows users to follow other users using the platform
Trigger	User requests to follow other users
Actors	user
Preconditions	User must sign in to the system
Main flow	1. user searches for other users by their user names 2. system displays the search results 3. user requests to view the searched user's profile 4. system displays profile of searched user 5. user requests to follow the searched user 6. system sends follow request to the searched user 7. system retrieves the request of the searched user requested to be followed by the other user 8. system displays the accepted and denied follow requests 9. use case ends
Alternative flow	1. system displays the posts of users 2. user requests to view the profile of the user who posted the information 3. use case resumes at main flow step 4
Exception Conditions	Unable to follow this user, the user blocked you
Post conditions	Followed user is displayed

Table 17: UC013 description

Use-case ID	**UC013**
Use-case Name	Un-follow user
Description	Allows users to un-follow other users using the platform that the user has been following before
Trigger	Users request to un-follow they have been following before
Actors	user
Preconditions	User must sign in to the system, user must follow the other user
Main flow	1. user searches for other users by their user names 2. system displays the search results 3. user requests to view the searched user's profile 4. system displays profile of searched user 5. user requests to un-follow the searched user 6. system displays the un-followed user 9. use case ends
Alternative flow	1. system displays the posts of users 2. user requests to view the profile of the user who posted the information 3. use case resumes at main flow step 4
Exception Conditions	The user blocked you
Post conditions	Un-followed user is displayed

Table 18: UC014 description

Use-case ID	**UC014**
Use-case Name	Send message
Description	Users are allowed to chat with other users privately and publicly in a group
Trigger	User requests to send message to other users
Actors	user
Preconditions	User must sign in to the system, users should be followed each other
Main flow	1. user searches for user to send message 2. system displays the search results 3. user requests to send message to the searched user 4. system checks for type of file to be sent 5. system sends the message to the other user 6. system displays the message request 7. use case ends
Alternative flow	None
Exception Conditions	Unable to send this type of file
Post conditions	Sent messages are displayed

Table 19: UC015 description

Use-case ID	**UC015**
Use-case Name	Video chat
Description	Allows users to chat with live video call
Trigger	User requests to chat with live video call
Actors	user
Preconditions	User must sign in to the system, users should be followed each other
Main flow	1. user searches for user to send message 2. system displays the search results 3. user requests to chat video call 4. system retrieves the other user's response 6. system displays live video stream 7. use case ends
Alternative flow	None
Exception Conditions	Unable to connect with the other user due to low internet
Post conditions	Live video stream displayed

Table 20: UC016 description

Use-case ID	**UC016**
Use-case Name	Attach file
Description	Allows users to send text, image, video and other files to other users or posts those files
Trigger	User requests to attach file to send to other person or post it
Actors	user
Preconditions	User must sign in to the system
Main flow	1. user requests to attach file 2. system provides choices from the local user file or other platforms 3. user submits the chosen file 4. system displays the file 6. user sends these attached file 7. system displays the sent file 8. use case ends
Alternative flow	None
Exception Conditions	Unable to attach this type of file
Post conditions	The attached file is sent

Table 21: UC017 description

Use-case ID	**UC017**

Use-case Name	Report user
Description	Users are allowed to report a user who posts inappropriate contents
Trigger	User requests to report other user
Actors	user
Preconditions	User must sign in to the system
Main flow	1. user requests to view other user's profile 2. system displays other user's profile 3. user requests to report the other user 4. System submits user's request 6. Admin and moderator checks for the report and decides what to do 7. system displays the reported users 8. use case ends
Alternative flow	None
Exception Conditions	User is already banned
Post conditions	The reported user is displayed

Table 22: UC018 description

Use-case ID	**UC018**
Use-case Name	Report post
Description	Allows Users to report posts to be deleted if they have inappropriate contents
Trigger	User requests to report posts
Actors	user
Preconditions	User must sign in to the system
Main flow	1. user requests to view other user's post 2. system displays other user's post 3. user requests to report the post 4. System submits user's request 6. Admin and moderator checks for the report and decides what to do 7. system deletes the post if it is not appropriate 8. use case ends
Alternative flow	None
Exception Conditions	Post is already deleted
Post conditions	The reported post is deleted from the platform

Table 23: UC019 description

Use-case ID	**UC019**
Use-case Name	Block user
Description	Allows users to block other users from being their friends

Trigger	User requests to block other user
Actors	user
Preconditions	User must sign in to the system
Main flow	1. user requests to view other user's profile 2. system displays other user's profile 3. user requests to block the user 4. System submits user's request 6. system blocks the selected user 7. use case ends
Alternative flow	None
Exception Conditions	User is banned
Post conditions	The blocked user is deleted from other user account

Table 24: UC020 description

Use-case ID	**UC020**
Use-case Name	Ban users
Description	Admins are allowed to ban users who have been reported many times from using the system
Trigger	Admin and moderator bans reported users
Actors	Admin/ Moderator
Preconditions	Admin/ Moderator should sign in with valid accounts to the dashboard
Main flow	1. user requests to report a user needs to be ban 2. system submits the user request 3. Admin/ Moderator checks the user if needs to be ban 4. Admin/ Moderator approve the ban 6. system bans the user from using it 7. use case ends
Alternative flow	None
Exception Conditions	User deletes the account
Post conditions	The banned is deleted from the system

Table 25: UC021 description

Use-case ID	**UC021**
Use-case Name	Host live video conference
Description	Allows users to host live video conferences where everyone can join in or can also be private
Trigger	User hosts live video conferences
Actors	User
Preconditions	User needs to sign in

Main flow	1. user requests to host live video conference 2. system submits user request 3. system generates links for other users to join the live conference 4. system displays the live video conference 5. use case ends
Alternative flow	None
Exception Conditions	Unable to host live video conference due network error
Post conditions	Live video conference is hosted

Table 26: UC022 description

Use-case ID	**UC022**
Use-case Name	Join video conference
Description	Allows users to join online video conference hosted by other users
Trigger	User requests to join live video conferences
Actors	User
Preconditions	User needs to sign in
Main flow	1. user requests to join live video conference 2. system submits user request 3. system allows user to join the conference 4. use case ends
Alternative flow	None
Exception Conditions	Unable to join the video conference due to the privacy issues
Post conditions	Live video conference is joined

Table 27: UC023 description

Use-case ID	**UC023**
Use-case Name	Leave video conference
Description	Users can leave the live video chat they are attending if they do not like the content of the conference
Trigger	User requests to leave live video conferences
Actors	User
Preconditions	User needs to sign in, user needs to be joined to the live video conference
Main flow	1. user requests to leave live video conference 2. system submits user request 3. system allows user to leave the conference 4. use case ends
Alternative flow	None
Exception Conditions	None

Post conditions	Erased from live video conference members

Table 28: UC027 description

Use-case ID	**UC024**
Use-case Name	View report
Description	Allows admins to view the reports made by users of the system
Trigger	Admin/ Moderator views reports from users
Actors	Admin/ Moderator
Preconditions	Admin/ Moderator need to be signed in with valid authentication
Main flow	1. Admin/ Moderator requests to view reports from user 2. system displays the reports 3. use case ends
Alternative flow	None
Exception Conditions	No reports sent
Post conditions	Reports displayed

Table 29: UC030 description

Use-case ID	**UC025**
Use-case Name	View chat history
Description	Allows users to view their chat history
Trigger	User requests to view chat history
Actors	Users
Preconditions	User needs to sign in, user needs to have a chat history
Main flow	1. user requests to view chat history 2. system submits user request 3. system displays user history 4. use case ends
Alternative flow	None
Exception Conditions	No chat history available
Post conditions	Chat history displayed

Table 30: UC034 description

Use-case ID	**UC026**
Use-case Name	Check notification
Description	Users can check for new information that they have been notified
Trigger	User requests to check their notifications
Actors	Users
Preconditions	User must have an account, user must sign in to the system

Main flow	1. user requests to check their notifications 2. system displays the notifications 3. use case ends
Alternative flow	None
Exception Conditions	No notification in the account
Post conditions	Notifications checked

Table 31: UC035 description

Use-case ID	**UC027**
Use-case Name	Check inbox
Description	Users can check their inbox if new private messages are sent for them
Trigger	User requests to check their inbox
Actors	Users
Preconditions	User must have an account, user must sign in to the system
Main flow	1. user requests to check their inbox 2. system displays the inbox 3. use case ends
Alternative flow	None
Exception Conditions	No inboxes in the account
Post conditions	Inbox checked

Table 32: UC036 description

Use-case ID	**UC028**
Use-case Name	Choose language
Description	Users are allowed to choose the language they want to display the platform
Trigger	User requests to choose language
Actors	Users
Preconditions	User must have an account, user must sign in to the system
Main flow	1. user requests to choose language 2. system displays languages to be chosen 3. user submit language chosen 4. system displays the content of the platform in the specified language 4. use case ends
Alternative flow	None
Exception Conditions	This language is not available on this platform yet
Post conditions	Language chosen

Table 33: UC037 description

Use-case ID	**UC029**
Use-case Name	Search for user
Description	Users can search for another users using their user names of the platform in order to make the searching process easy
Trigger	User requests to search another user
Actors	Users
Preconditions	User must have an account, user must sign in to the system
Main flow	1. user requests to search for user 2. user enters username to be searched 3. user submits username 4. system searches based on username 5. system displays the results of the search 5. use case ends
Alternative flow	None
Exception Conditions	The searched user is not available, username invalid
Post conditions	Searched results displayed

Table 28: UC040 description

Use-case ID	**UC030**
Use-case Name	CRUD private message
Description	Users can create, read, update and delete any private message and private message session that belongs to them
Trigger	User requests to CRUD private message
Actors	Users
Preconditions	User must have an account, user must sign in to the system
Main flow	1. user requests to CRUD private message 2. system submits user request 3. system displays the CRUD result 4. use case ends
Alternative flow	None
Exception Conditions	User is not available
Post conditions	CRUD performed on private message

3.2.6 Activity Diagram

3.2.6.1 Sign up

This diagram will show the activities involved when a new user is signing up and getting registered to the system.

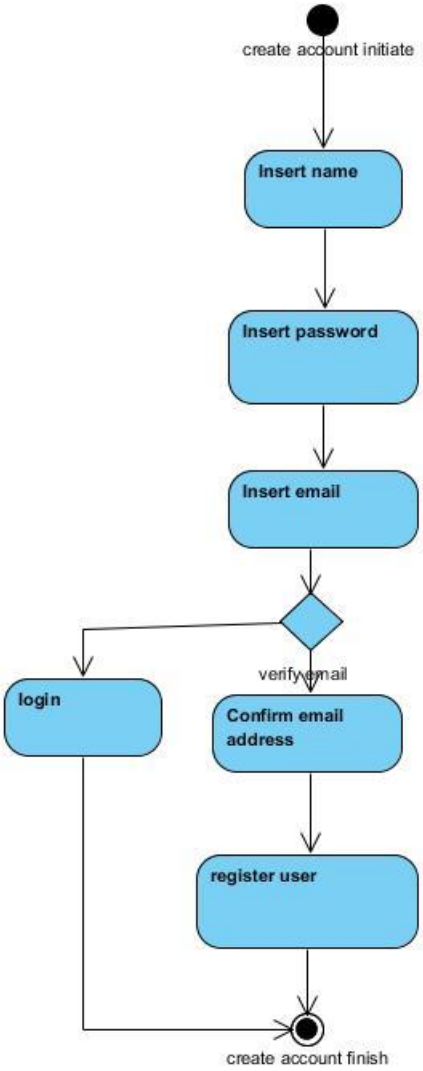

Figure 3: Sign up Activity Diagram

3.2.6.2 CRUD profile

This diagram will show the activities involved when creating, reading, updating, and deleting user profiles created by the user.

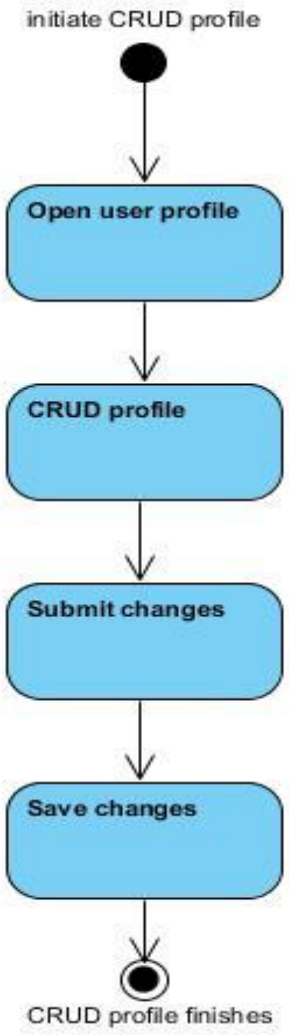

Figure 4: CRUD profile Activity Diagram

3.2.6.3 Post information

This diagram will show the activities involved when user gets to post information like text, image, and video files.

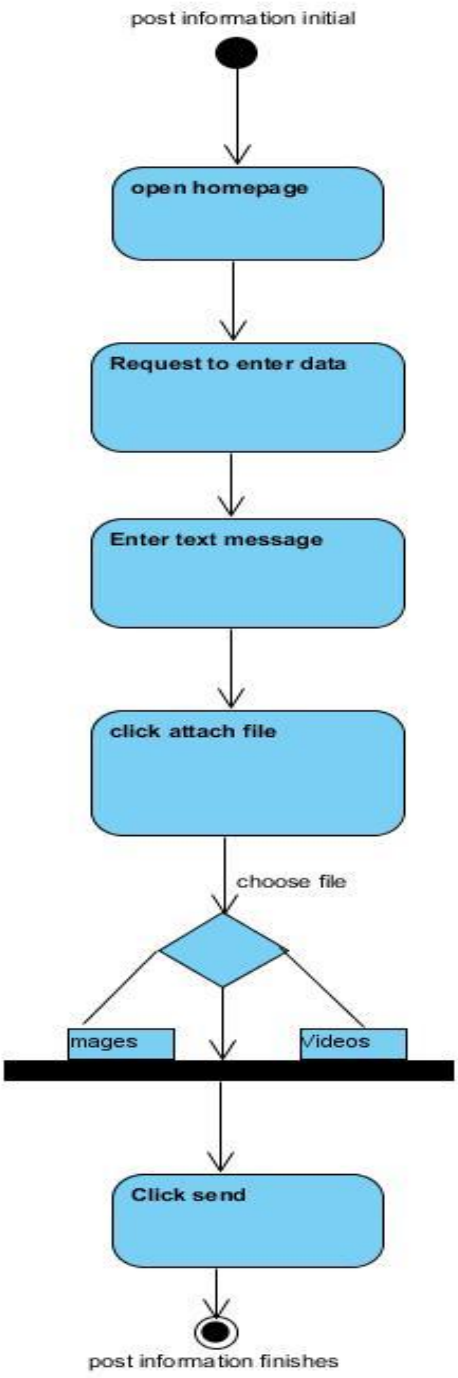

Figure 5: Post information Activity Diagram

3.2.6.4 Host live video conference

This diagram will show the activities involved when user hosts live video conferences in the group and as a private person.

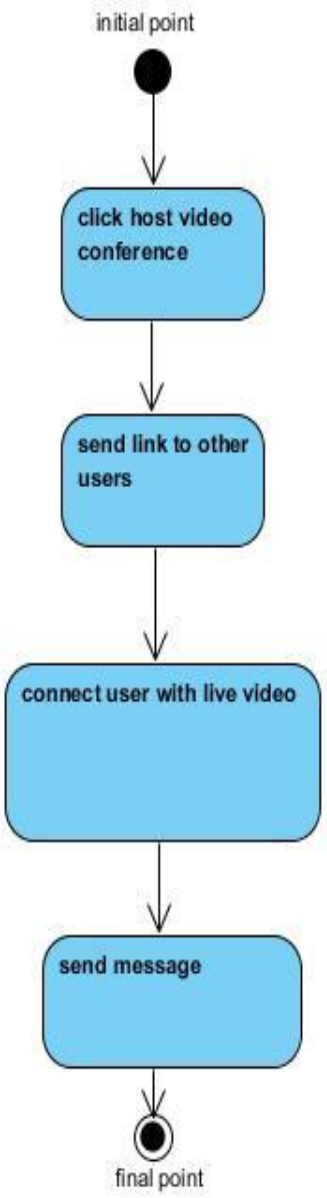

Figure 6: Host live video conference Activity Diagram

3.2.6.5 Create group

This diagram will show the activities involved when user wants to create a group in order to share information among users.

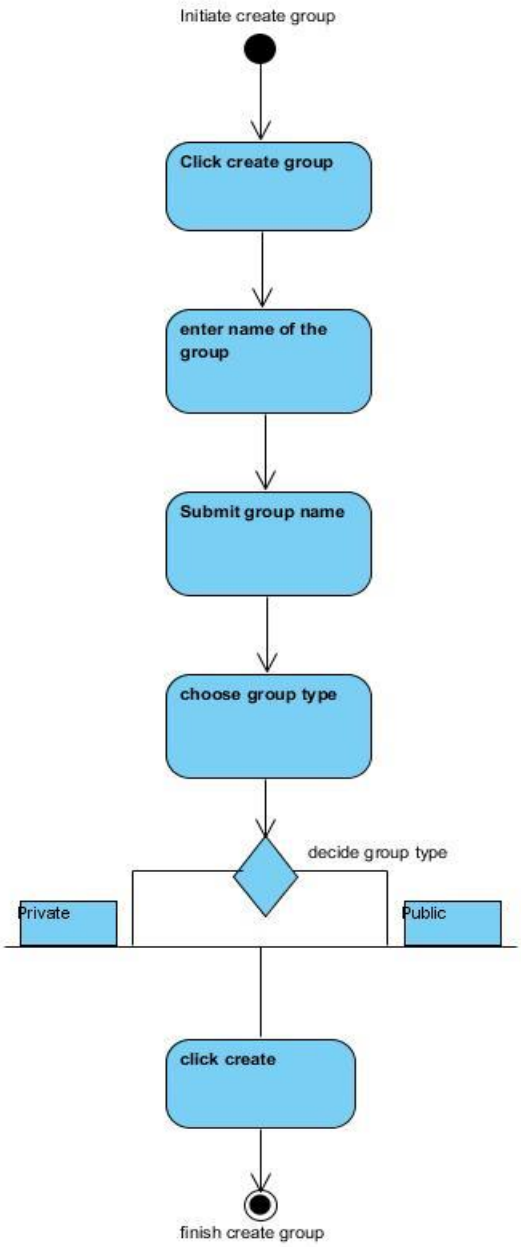

Figure 7: Create group Activity Diagram

3.3 Behavioral/ Dynamic Modeling

This section of the document will illustrate the behavioral functionality of the system using sequence diagrams and state diagrams.

3.3.1 Sequence Diagram

3.3.1.1 Sign up sequence diagram

This diagram shows the sequences taken to sign up on the system.

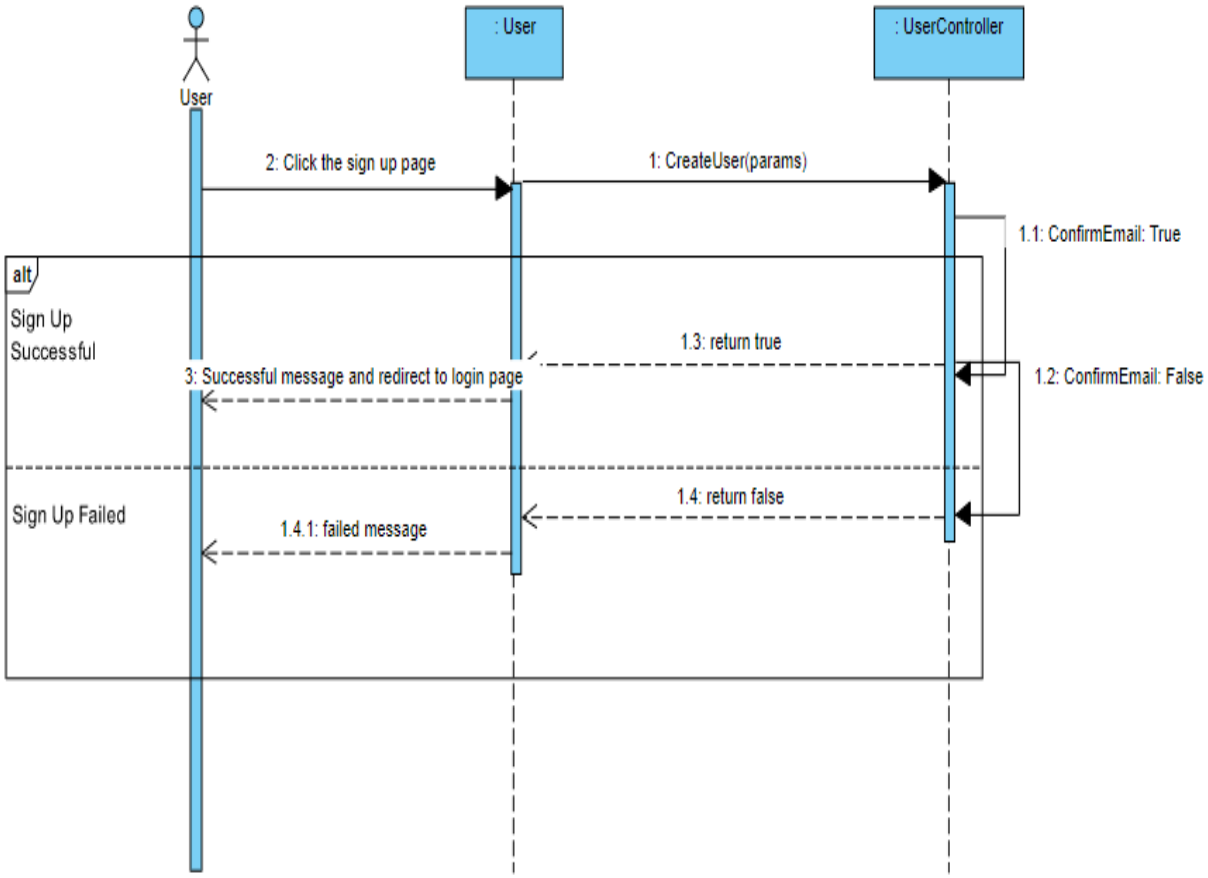

Figure 8: Sign up Sequence Diagram

3.3.1.2 Message sequence diagram

This diagram shows the sequences taken to chat with other users on the system.

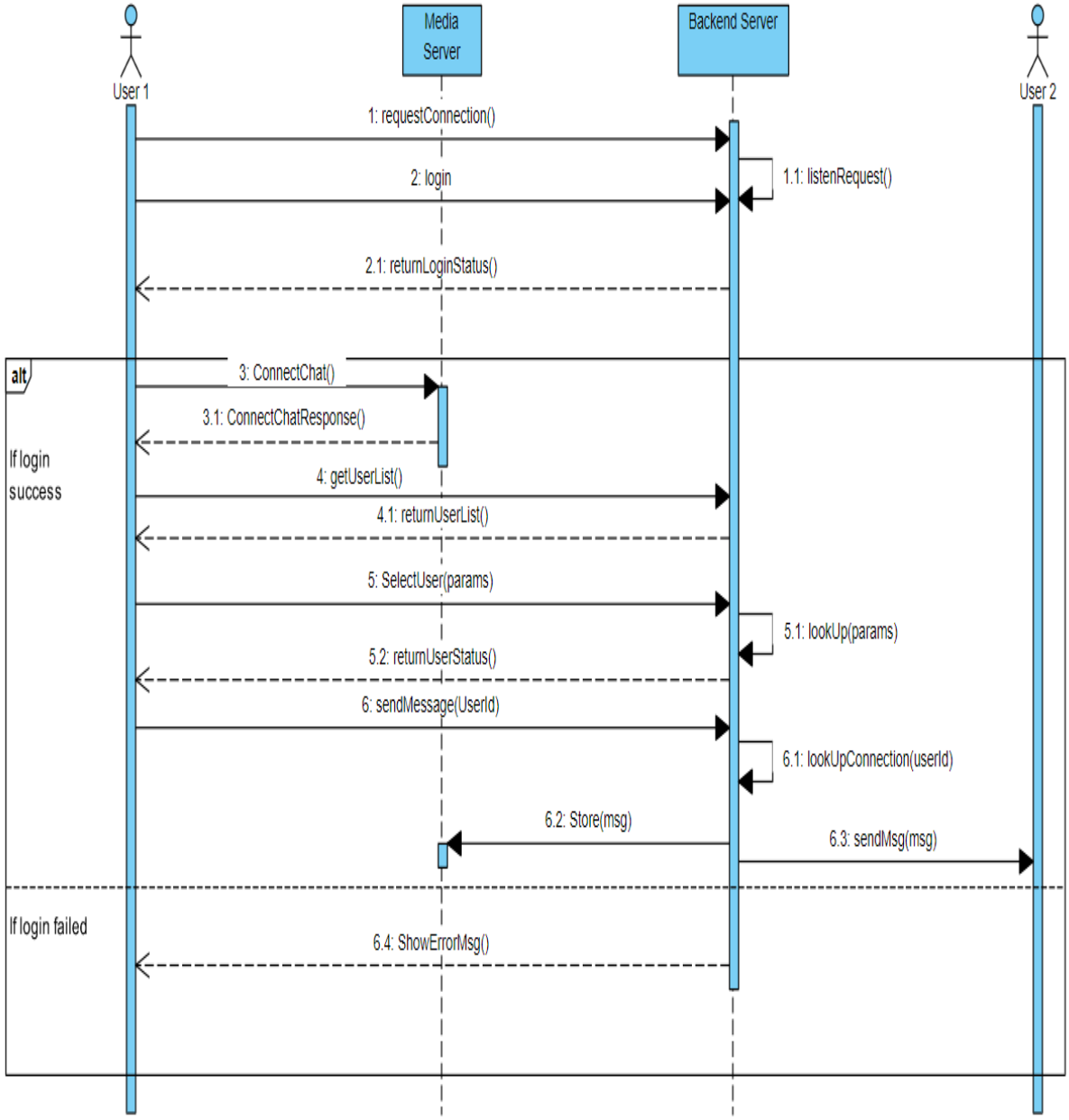

Figure 9: Message Sequence Diagram

3.3.1.3 Post information sequence diagram

This diagram shows the sequences of posting information on the system.

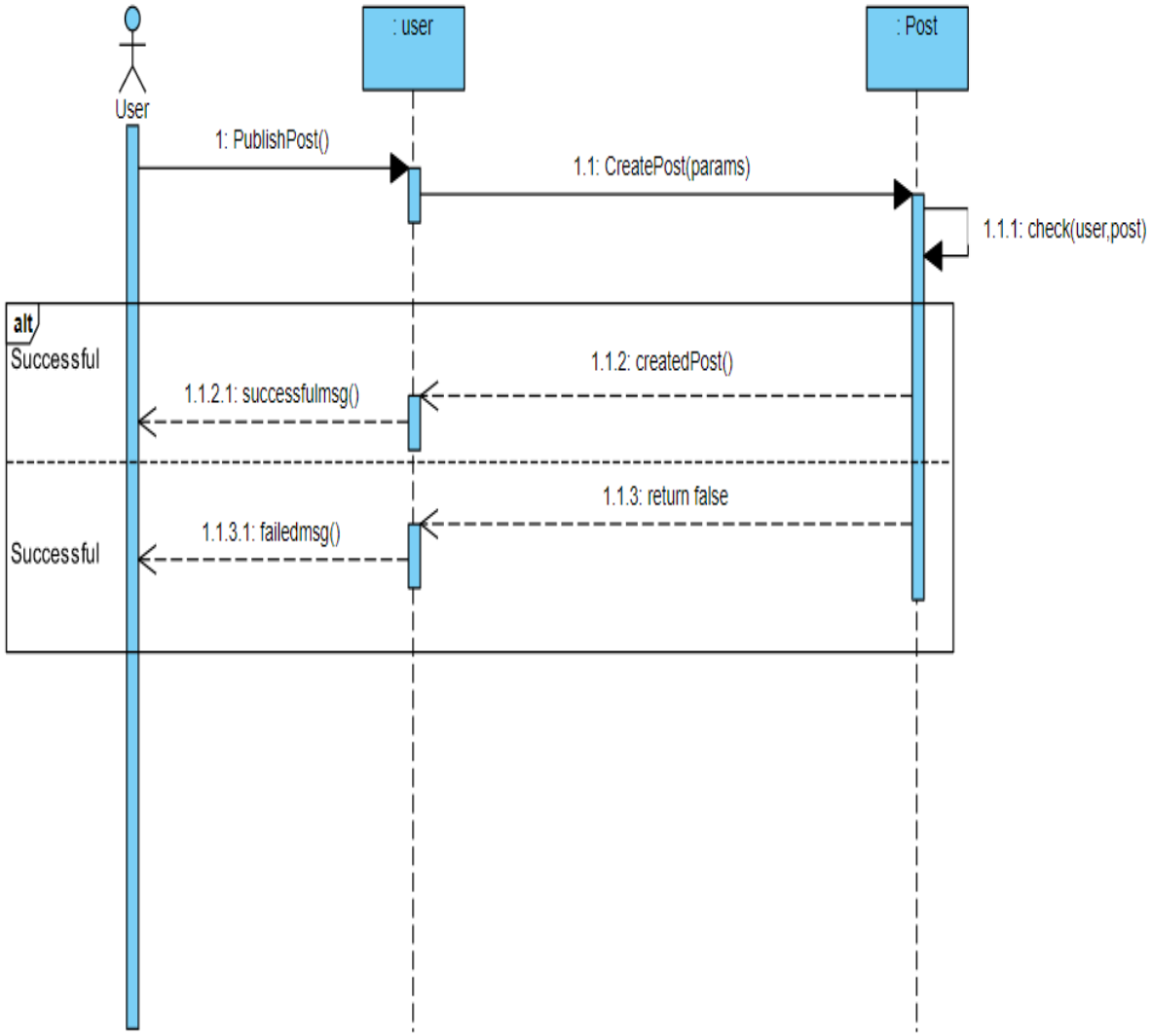

Figure 10: Post information Sequence Diagram

3.3.1.4 Create group sequence diagram

This diagram shows the sequences used to create a group in the system.

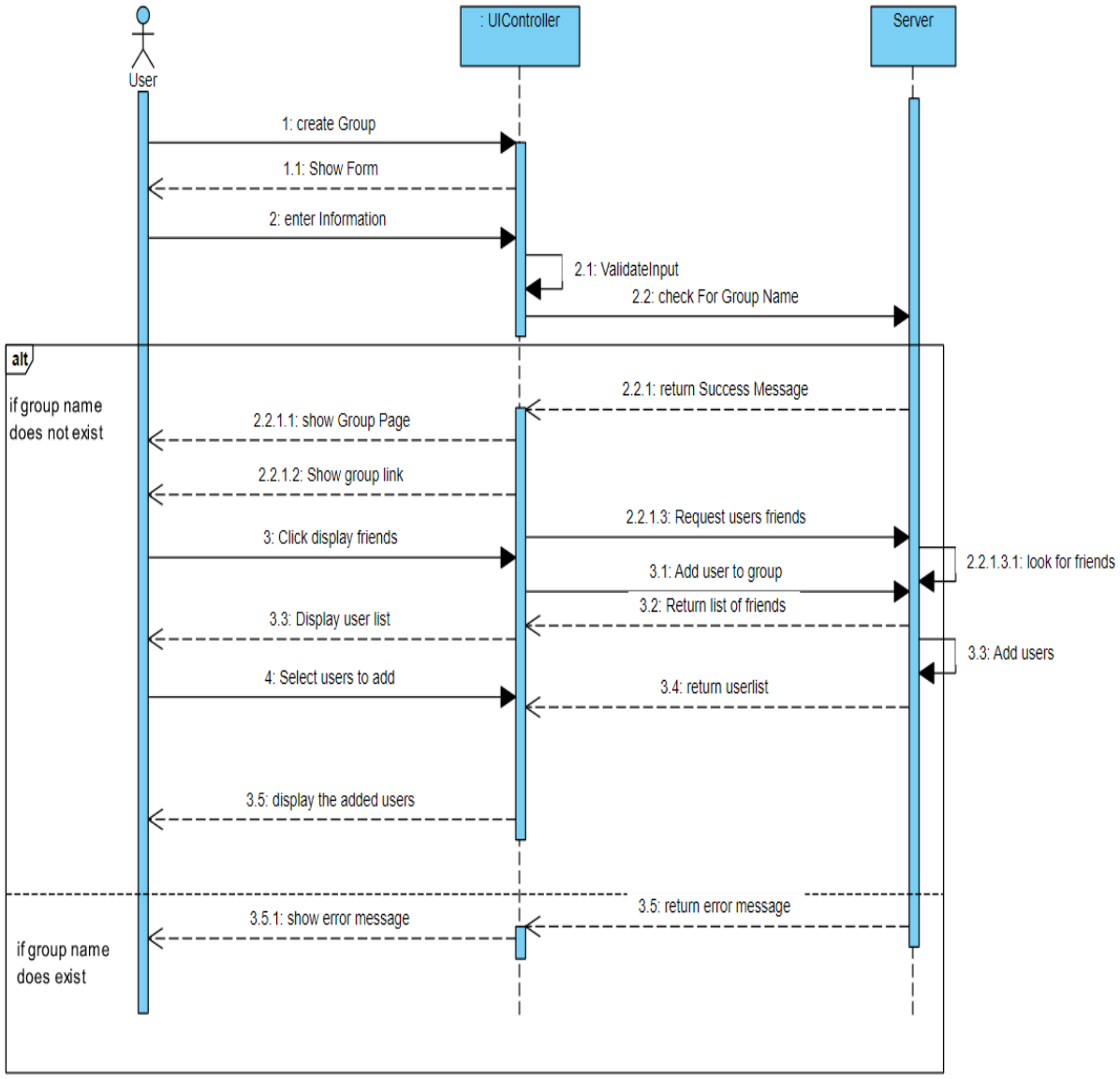

Figure 11: Create group Sequence Diagram

3.3.2 State Diagram

This section will show the navigational state diagram of the service when implemented on different platforms and navigation among different pages on the platform.

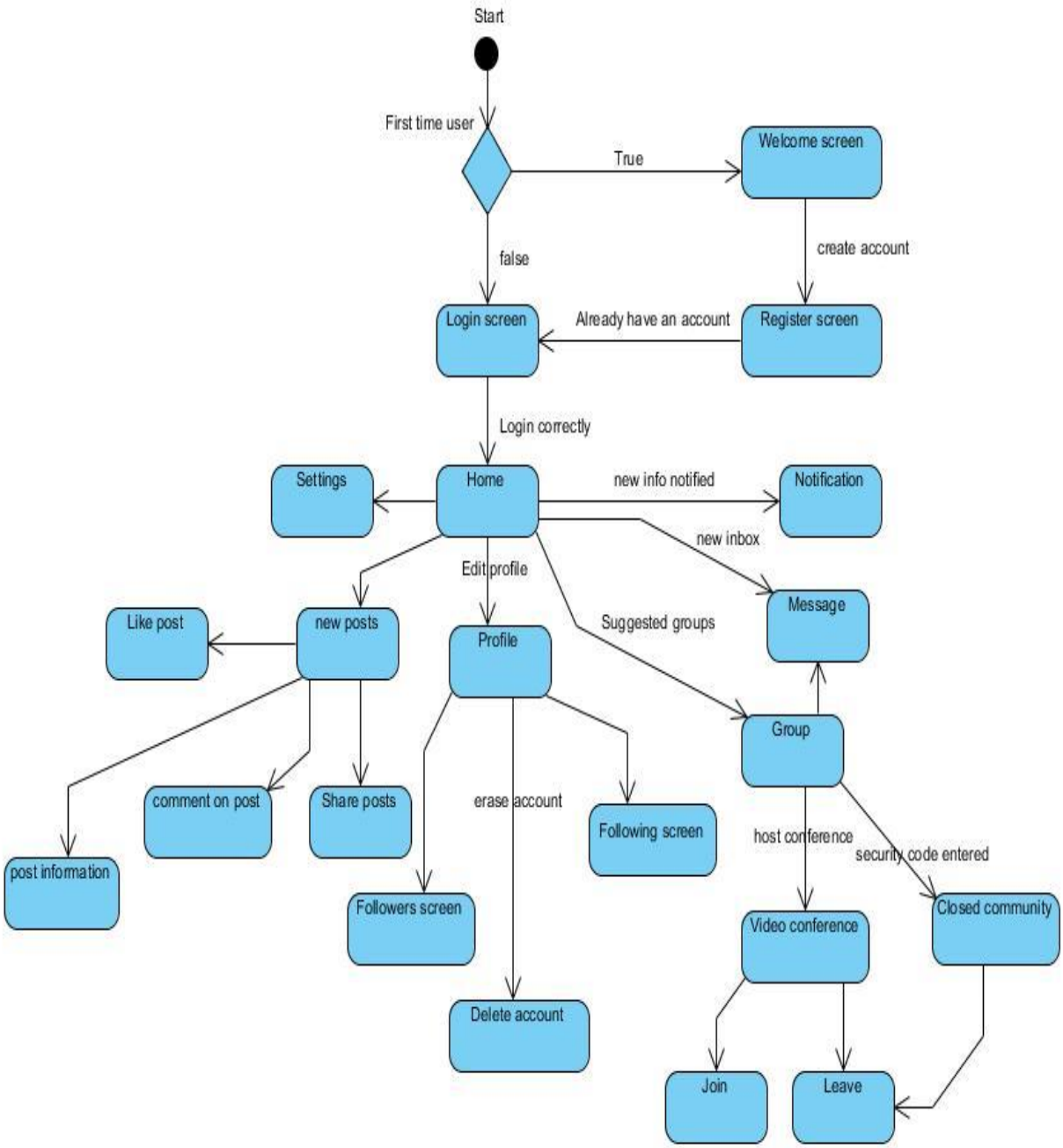

Figure 12: Platform navigation State Diagram

3.4 Class-Based Modeling

3.4.1 Identifying Classes

Table 35: Identification of class

Class name	Class description

Profile	This class will hold information related to the profile of the user who has created an account on the system.
Post	This class will hold information to be posted on the platform.
Group	This class will hold the groups created by users of the system and holds information need to be included in the group.
Like	This class will hold all the posts liked.
Comment	This class will hold all comments on different posts of users.
Activity	This class will hold all the information about the liked, commented, and shared posts.
Message	This class will hold all the messages sent and received users and also contains type of messages to be transferred.
Follow	This class will hold all the users followed with each other.
User	This class will hold all the information about a user and their status
Search	This class will hold all the searched information by users of the system.

Video Conference	This class will hold all the attributes needed to host a video conference.
Member	This class will hold information of all members of groups created on the platform.
Notification	This class will hold all new information notified on the system.
Share	This class will hold all the necessary information to share some post and also holds all the shared posts.
Report	This class will hold the necessary information needed to report a user who has been posting inappropriate contents on the platform.
Channel	This class will hold all the necessities to create a channel.
Schedule	This class will hold all the attributes users use to schedule their tasks on the platform.

3.4.2 Class Diagram

The following is the class diagram that explains the layout of the class structure inside the system. More in-depth clarifications will be shown in the database design on the next chapter.

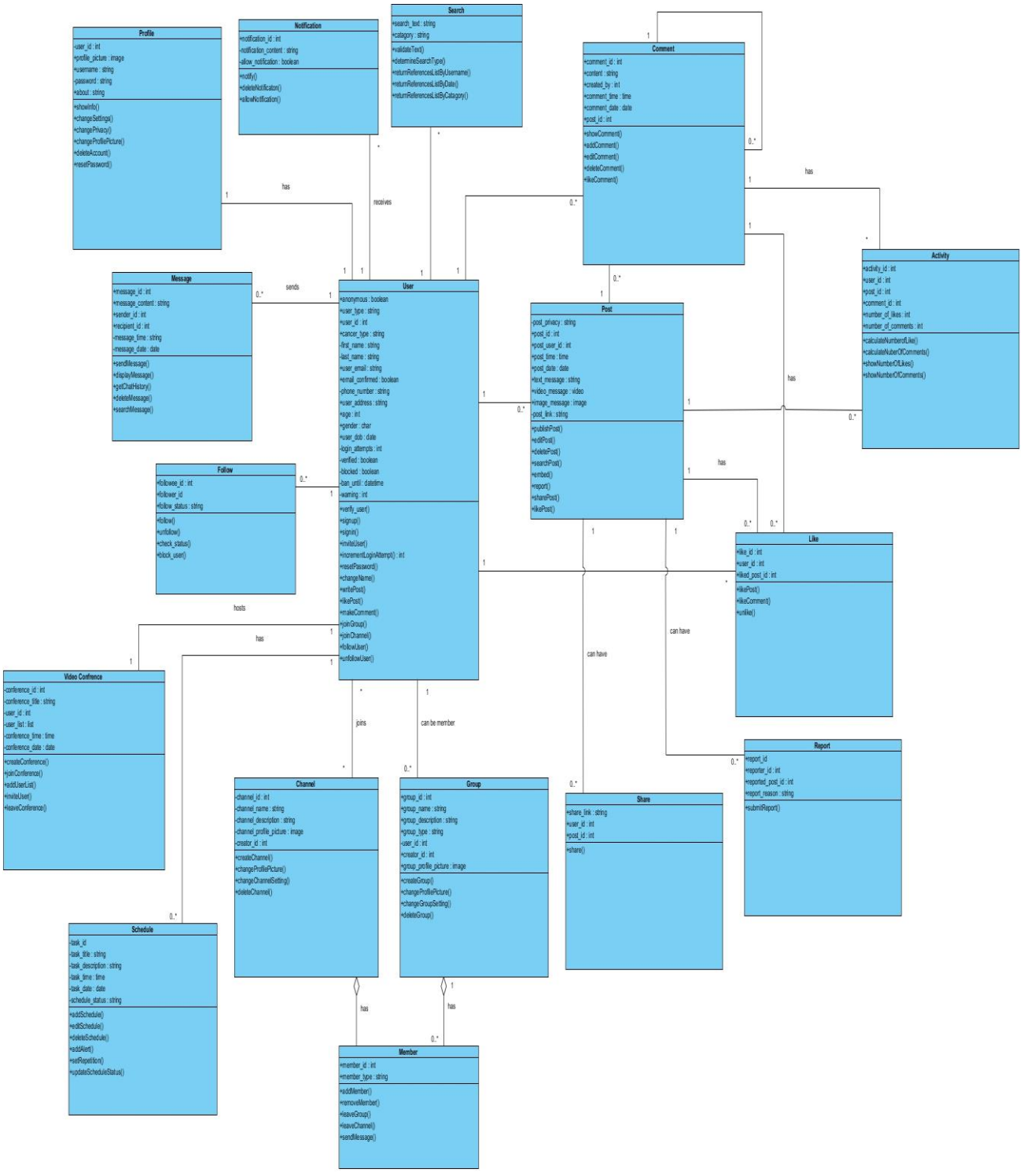

Figure 13: Class Diagram

4 System Design

4.1 Overview

This section mainly focuses on design part of our system. It describes our system using sets of symbols to represent various components and relationships within the system diagrammatically. The diagrams in this section will form the foundation for the system design and are important part of the document to make the implementation phase easy. It will also show some of the user interfaces of the mobile application to be built during implementation phase.

4.2 System design

4.2.1 Components

Internal Auth: This component does not authenticate the users. This is the component that authenticates Request Origins and Permissions. Handled internally.

Requests Controllers: The Components that listen and respond to UI changes and send API requests.

RESTful API: The component that has the end-points to handle database requests from the frontend.

Firebase Auth: External framework used to handle user authentication.

PostgreSQL Datastore: datastore component.

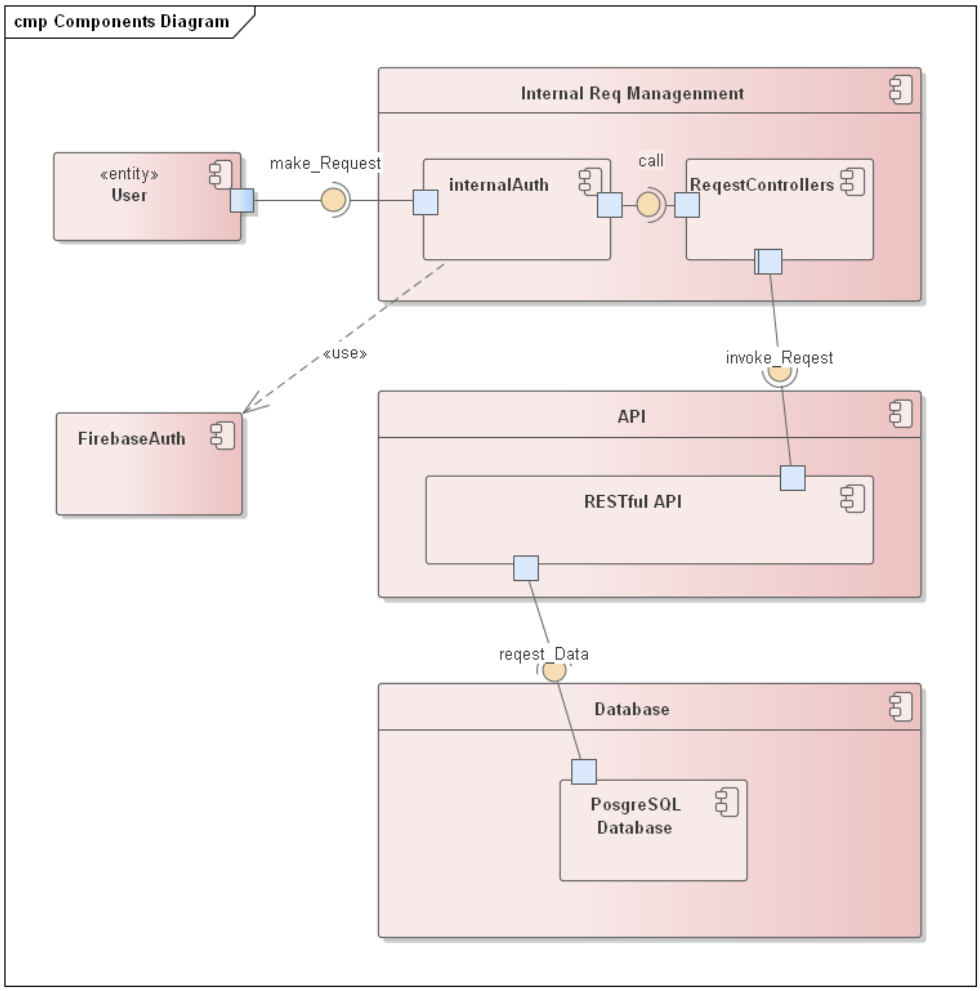

Figure 14: Components Diagram

4.2.2 Packages

Controllers: Internal to the client side. Authenticate and invoke API requests from the frontend.

PostgreSQL Datastore: Datastore component.

API: Interface between the PostgreSQL datastore and the internal request controller. Contains the components that deal with API implementation

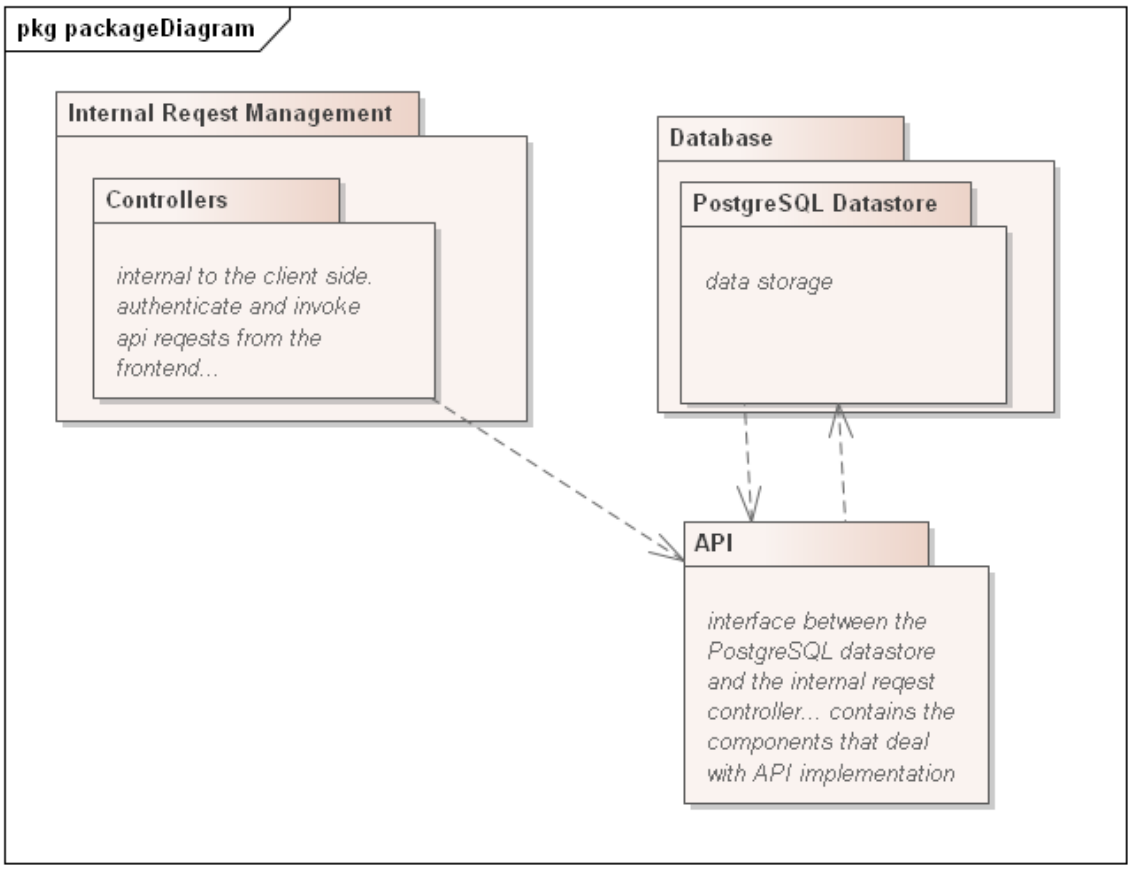

Figure 15: Package Diagram

4.3 Architecture of the System

4.3.1 Generalized system architecture

From a development point of view, the system can best be described as a social media platform. With that in mind we have made decisions on the architecture based on the specific needs and requirements of the system.

Our chosen design pattern is client-server architecture. This is necessary as continuous server requests are the best way to keep up to date the ever-changing data and environment that the system will have. It is also necessary since the data is well protected due to its centralized architecture.

The client side, by virtue of being implemented in flutter will have an MVC (Model View Controller) design pattern in addition to the Client Server architecture that will be used for the system. This ensures that business logic stays separate from layouts and views and makes the system maintainable, at scale, by allowing the business logic to be tweaked at any point in time.

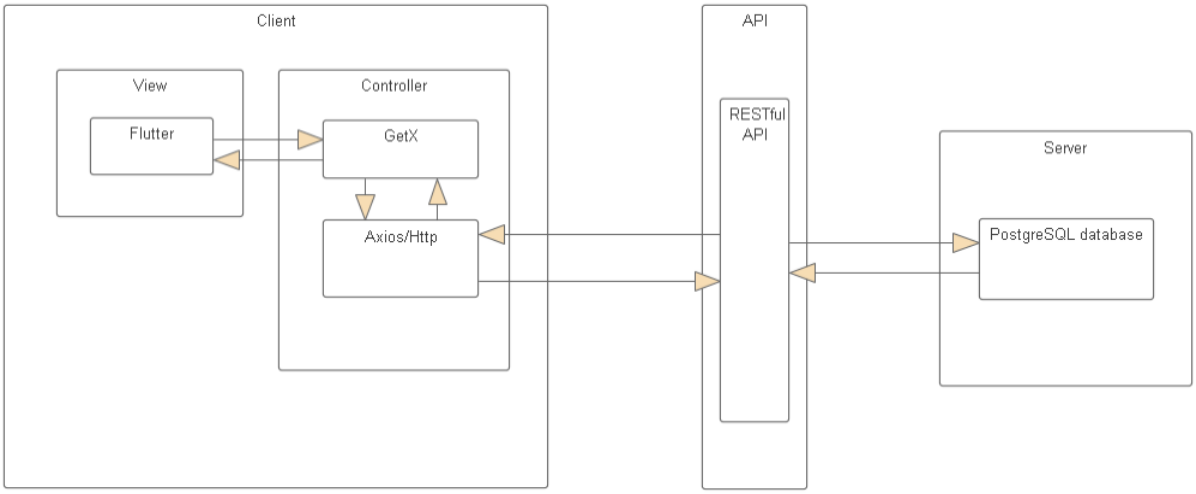

Figure 16: Generalized System Architecture

This means that we will essentially be using a fat client as the business logic will indubitably have to remain client side. This, we recon, will be ideal, since we estimate the system bottleneck will not be business logic but instead copious amounts of data which will be stored server side.

Implementing in flutter also has the advantage of simplifying the client-side architecture, since routing is handled internally by means of context switching. We do however intend to use flutters Axios and/or http packages in order to make http requests for the client server integration.

On the server side there is a database. By nature, a social media platform tends to swell database sizes at scale. It is also hard to say what the potential data types will be. To this end, we have chosen to use PostgreSQL, for its expandability and ability to process complex data types.

The API layer will consist of the server and client side communicating through a RESTful API type that we will have designed.

4.3.2 The Backend

As the system, in its core, can be described as a social media platform, the majority of the heavy lifting lies in the backend, i.e., the database. We estimated that the primary requirements of the database, we settled on **high query speeds**, since the speed at which data is sent off to the view has a direct impact on usability and low-down time is essential for social media platforms, **high concurrency** since the number of users and requests at scale is expected to be high, and finally

the **object relational** nature of the database since that will allow us to make potentially redundant implementations minimal.

To this end we chose PostgreSQL for the database implementation over simple MySQL, since although t closely resembles the familiar SQL, it allows us the flexibility of using table inheritance. We chose an SQL based database system since NoSQL databases aren't designed for the one-to-one or one-to-many entity relationships that are essential in our app, despite being faster.

On the API side, we chose to use the popular RESTful API design. We believe that using more complicated GraphQL model would make the API implementation overly complicated, and decided to use RESTful design which suits our purposes well instead of overengineering.

4.3.3 The Frontend

By virtue of being developed in flutter, the frontend follows flutters guidelines for an MVC system. The controller, which is, in itself a part of the application, is separated from the view and hands business logic.

That said, flutter has its own internal mechanisms for routing, and state management. However, they can be too semantic, and getting reactive state management is overly complicated, since it requires the passing of context from one view to another through constructors, which is an inelegant solution. We therefore chose to use the GetX flutter sub-framework, which adds to flutters context-based routing and state management a seamless integration of named routing and cleaner state management, which will allow our application to have a complex frontend, without any compromises.

4.3.4 State Management

GetX has two choices for state management, *Simple State Management*, which allows for UI surface updates to follow manual updating of the underlying data. However, we chose to use the more advanced *Reactive State Management* in order to make the underlying data observable, and let GetX perform surface updates automatically during data changes.

4.4 Database Design

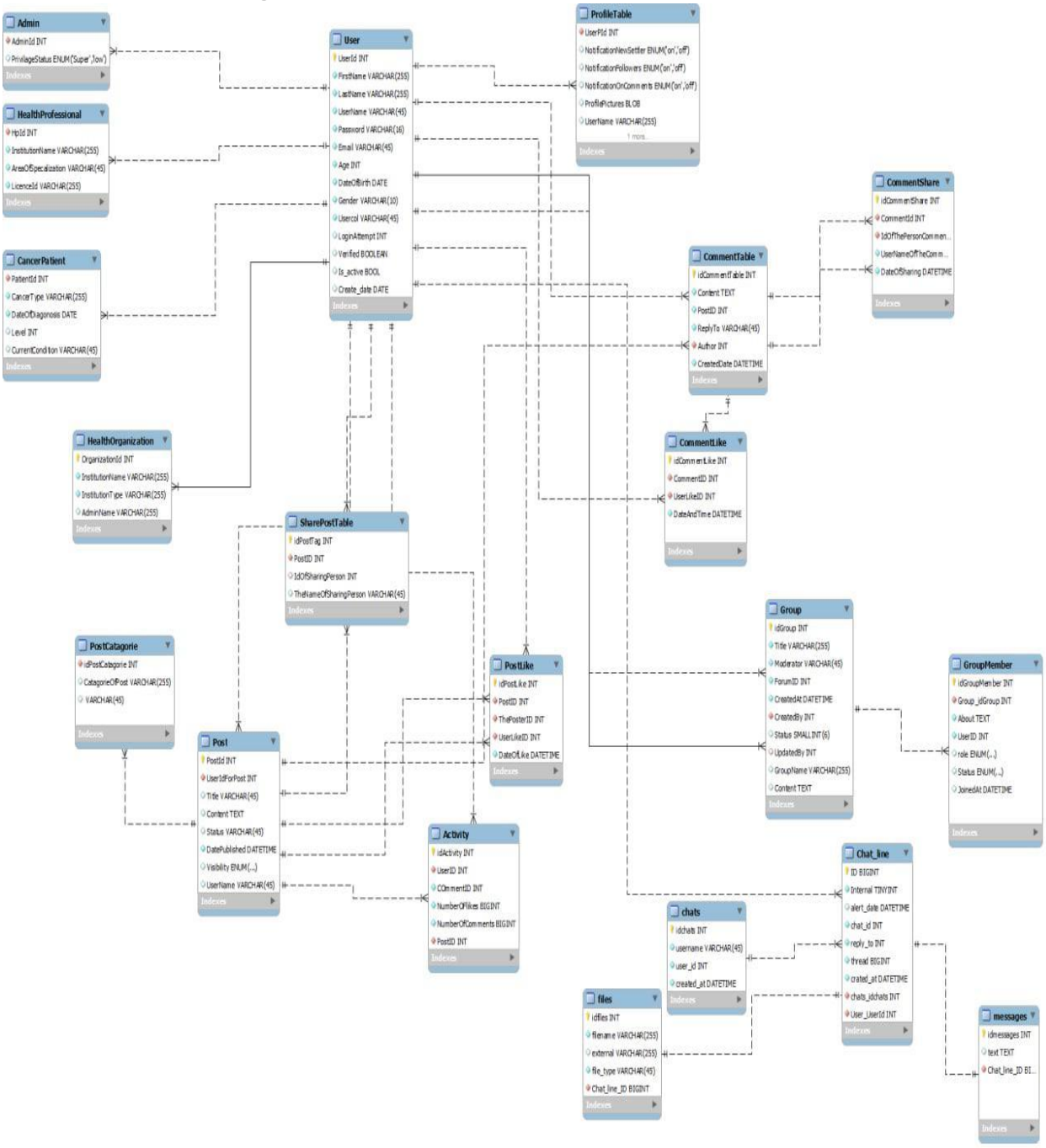

Figure 17: Database Design

4.5 User Interface Design

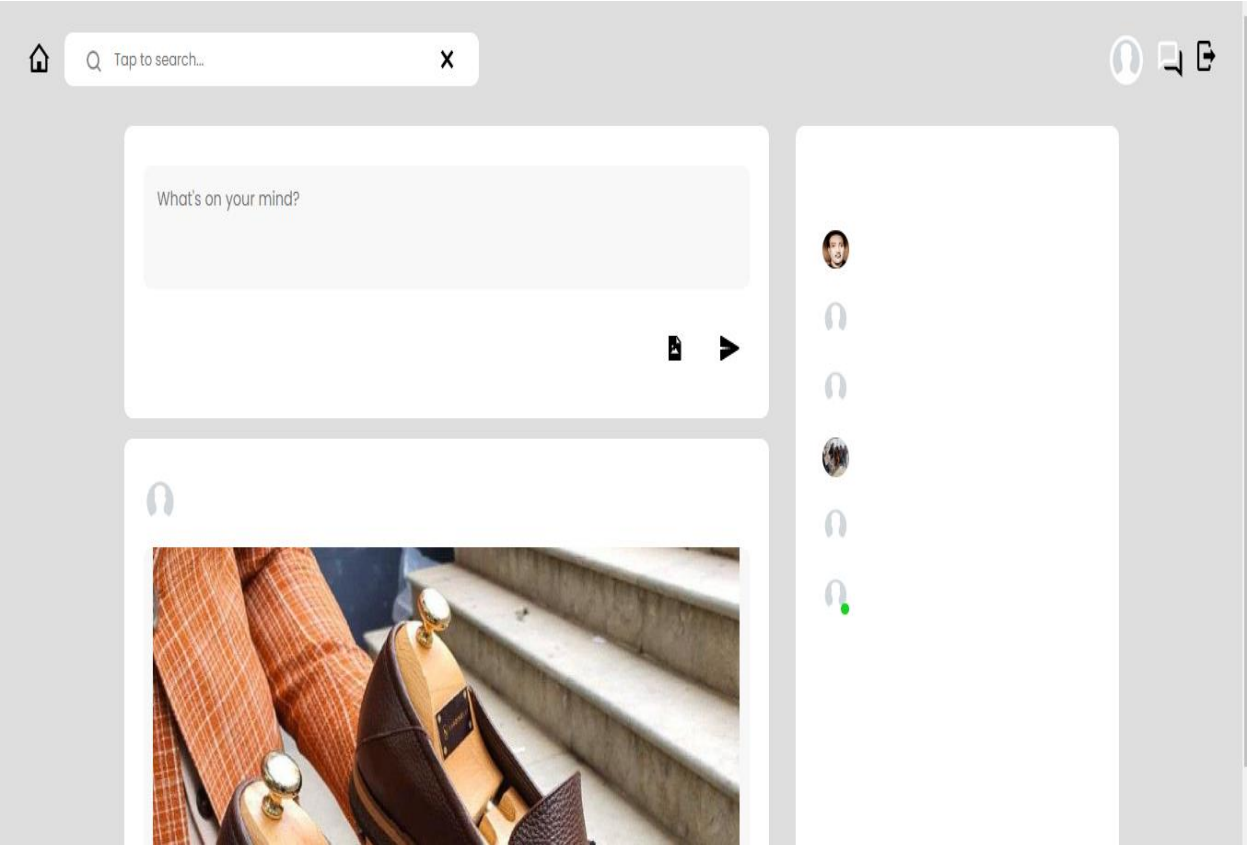

Figure 18: UI design for home Screen

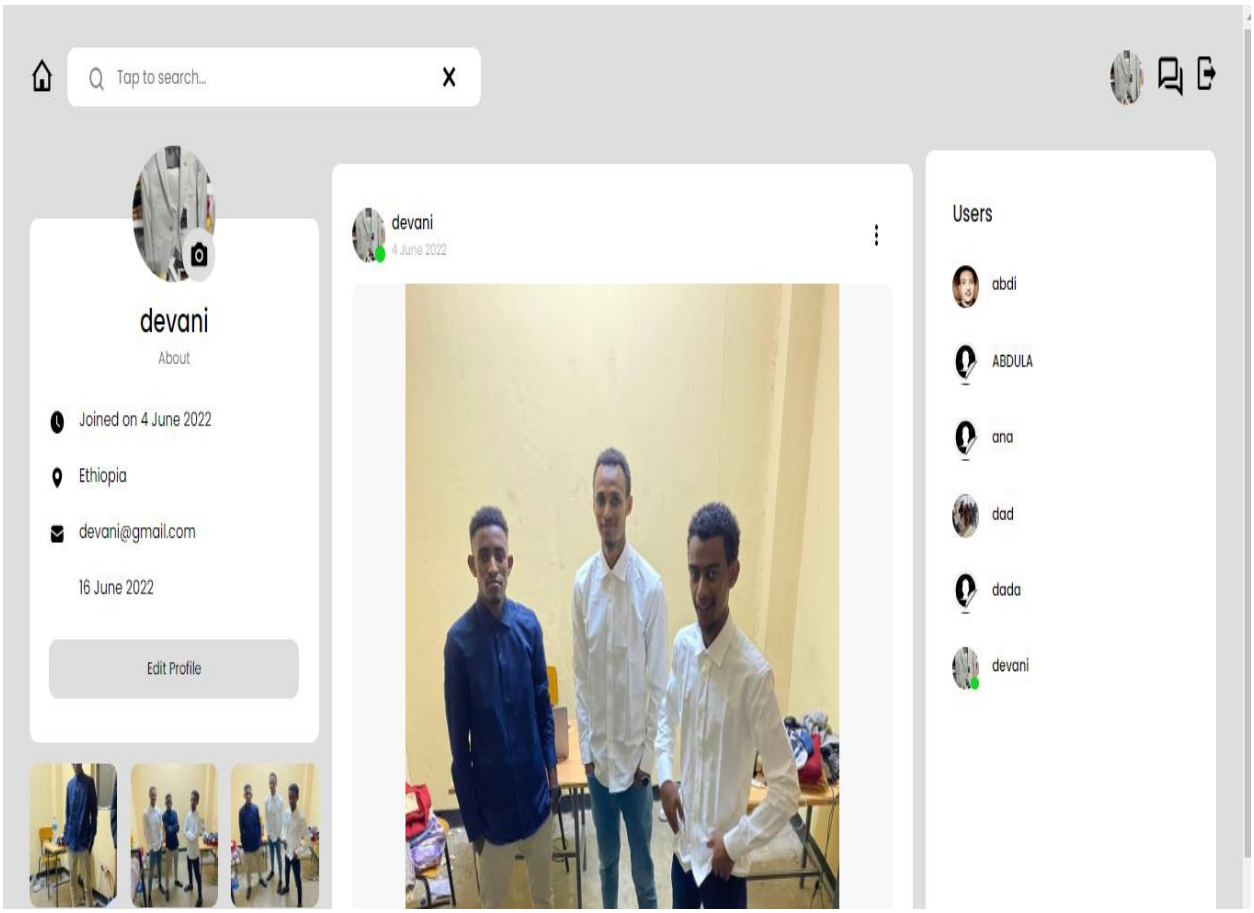

Figure 19: UI design for profile screen

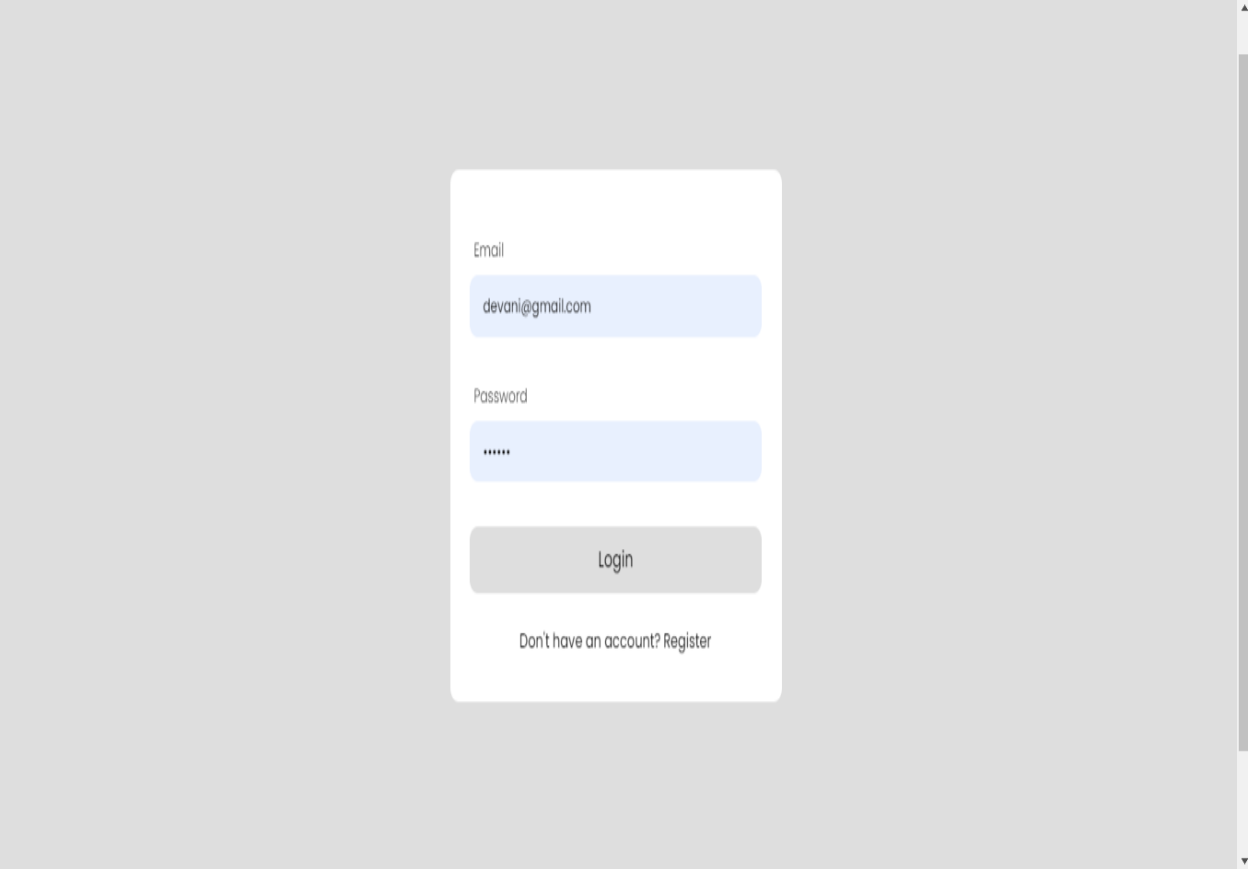

Figure 20: UI design for login screen

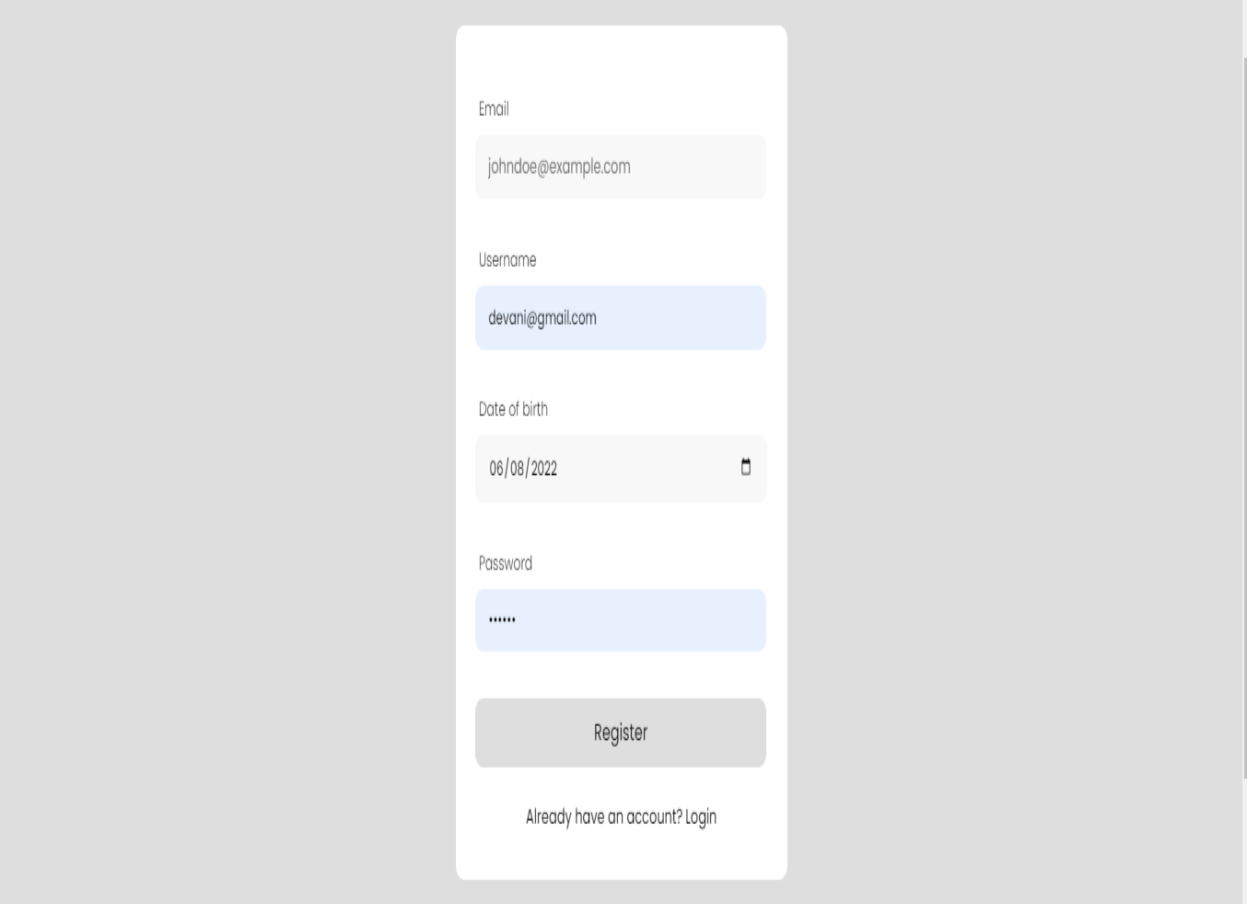

Figure 21: UI design sign up screen

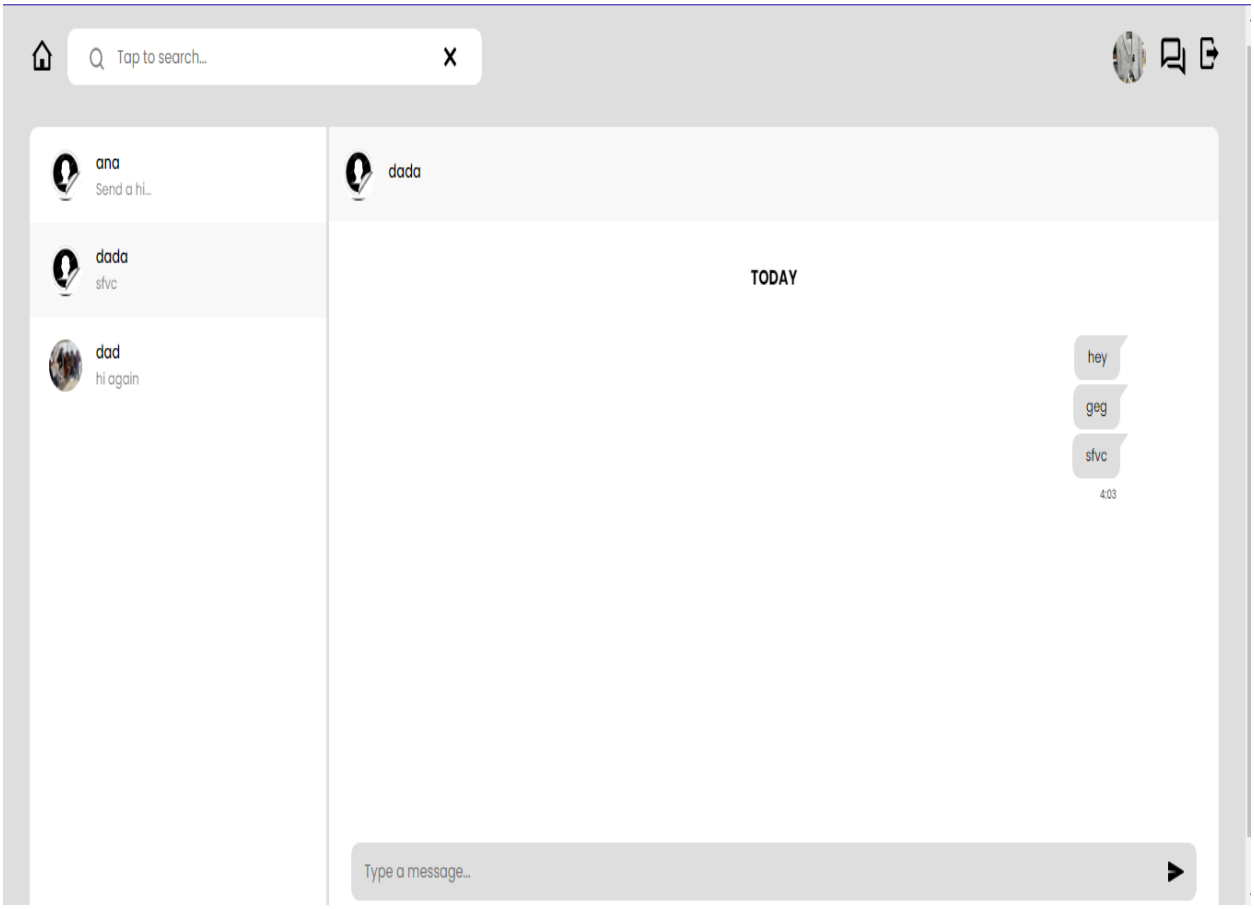

Figure 22: UI design for chatting screen

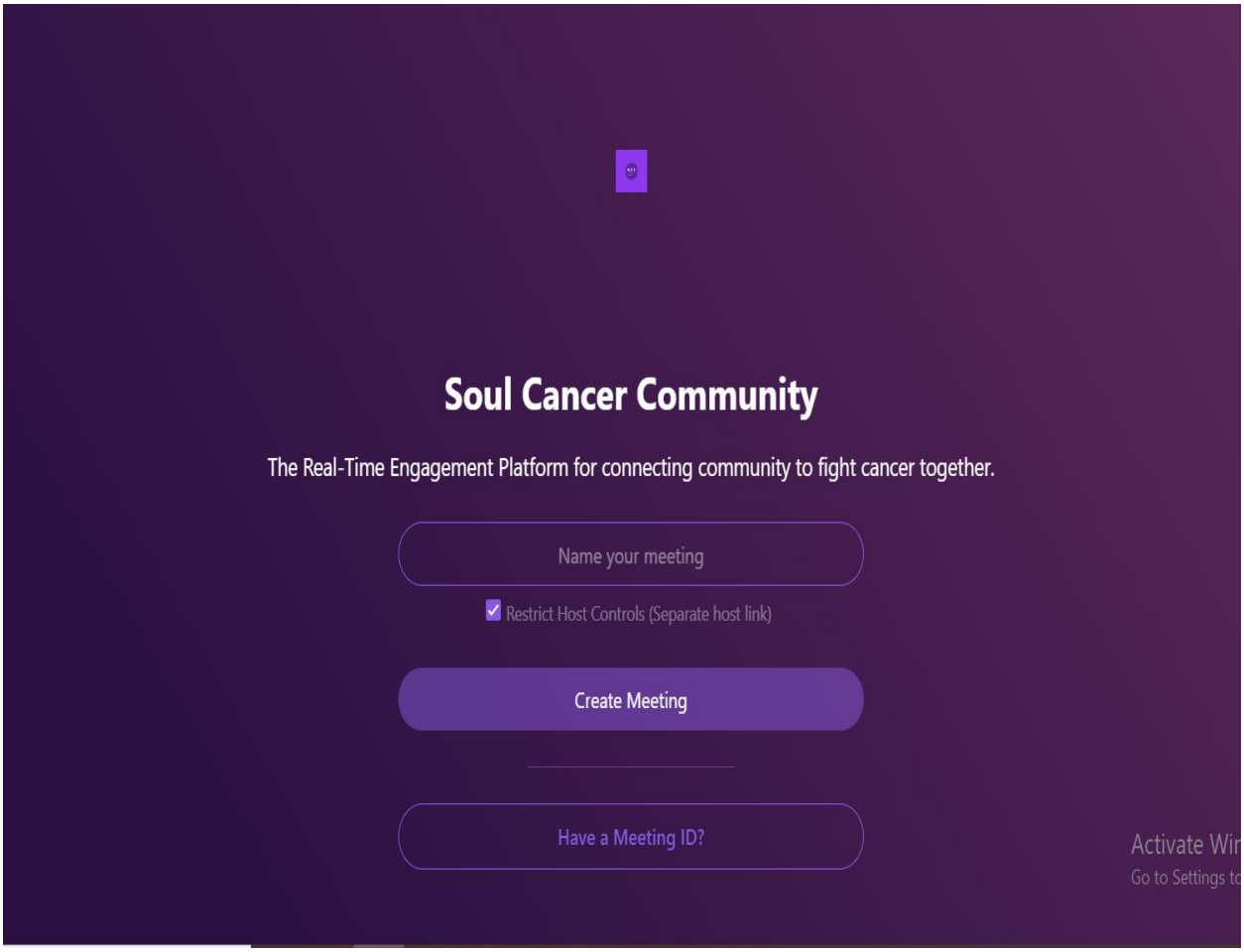

Figure 23: UI design for video conferencing home screen

Figure 24: UI design for live video conferencing screen

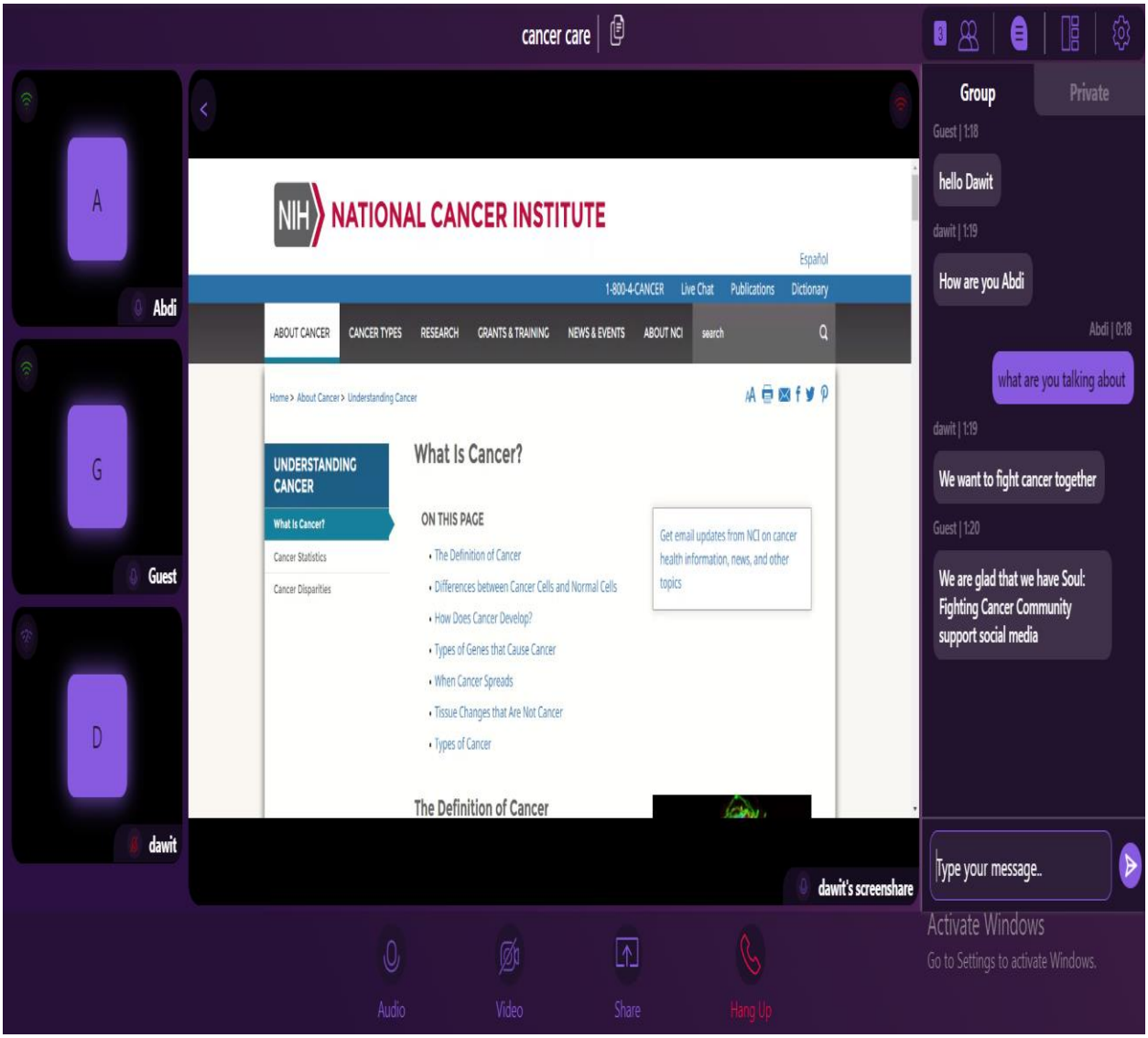

Figure 25: UI design for online group and private chat on video conferencing and screen sharing

5 Implementation and Testing

5.1 Implementation

Implementation is one part of the software development cycle which means the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy. Let us describe some of the schema designs for our project.

Back End implementation

Backend is server side of the website. It stores and arranges data, and also makes sure everything on the client-side of the website works fine. It is the part of the website that you cannot see and interact with. It is the portion of software that does not come in direct contact with the users. The parts and characteristics developed by backend designers are indirectly accessed by users through a front-end application. Activities, like writing APIs, creating libraries, and working with system components without user interfaces or even systems of scientific programming, are also included in the backend.

- **User schema**

This schema is the representation of users that participate in our system. It includes the necessary information about the user.

```
const mongoose = require("mongoose");

const bcrypt = require("bcryptjs");

const jwt = require("jsonwebtoken");


const UserSchema = new mongoose.Schema(
  {
    name: {
      type: String,
      required: [true, "Please provide name"],
      minlength: 2,
      maxlength: 20,
    },
    email: {
      type: String,
      required: [true, "Please provide email"],
      match: [
```

```
        /^(([^<>()[\]\\.,;:\s@"]+(\.[^<>()[\]\\.,;:\s@"]+)*)|(".+"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/,
          "Please provide a valid email",
      ],
      unique: true,
    },
    profileImage: {
      type: String,
      default: "",
    },
    password: {
      type: String,
      required: [true, "Please provide password"],
      minlength: 6,
    },
    dob: {
      type: Date,
      required: [true, "Please provide date of birth"],
    },
    location: {
      type: String,
      maxlength: 20,
    },
    about: {
      type: String,
      maxlength: 20,
      default: "About",
    },
  },
```

```
  { timestamps: true }
);

UserSchema.pre("save", async function () {
  const salt = await bcrypt.genSalt();
  this.password = await bcrypt.hash(this.password, salt);
});

UserSchema.methods.createJWT = function () {
  return jwt.sign(
    { id: this._id, name: this.name, profileImage: this.profileImage },
    process.env.JWT_SECRET,
    {
      expiresIn: process.env.JWT_LIFETIME,
    }
  );
};

UserSchema.methods.comparePassword = async function (pw) {
  const isCorrect = await bcrypt.compare(pw, this.password);
  return isCorrect;
};
module.exports = mongoose.model("User", UserSchema);
```

- **post schema**

This schema represents the data model of the posting item that is posted by different users of the website in order to receive reactions from their followers and other users.

```
const mongoose = require("mongoose");

const PostSchema = new mongoose.Schema(
```

```
{
  caption: {
    type: String,
  },
  image: {
    src: {
      type: String,
    },
    publicID: {
      type: String,
    },
  },
  createdBy: {
    type: mongoose.Types.ObjectId,
    ref: "User",
    required: true,
  },
  userDetails: {
    name: {
      type: String,
      required: true,
    },
    image: {
      type: String,
    },
  },
  likes: {
    type: [String],
```

```
    },
    comments: [
      {
        commentedBy: {
          type: mongoose.Types.ObjectId,
          ref: "User",
          required: true,
        },
        comment: {
          type: String,
          required: true,
        },
        commentedAt: {
          type: Date,
          default: new Date(),
          required: true,
        },
      },
    ],
  },
  { timestamps: true }
);

module.exports = mongoose.model("Post", PostSchema);
```

- **profile schema**

This schema represents the data model of the profile of the registered users on our website, what the profile page holds will be represented down below.

```
const mongoose = require("mongoose");

const bcrypt = require("bcryptjs");

const jwt = require("jsonwebtoken");

    profileImage: {
      type: String,
      default: "",
    },
    password: {
      type: String,
      required: [true, "Please provide password"],
      minlength: 6,
    },
    dob: {
      type: Date,
      required: [true, "Please provide date of birth"],
    },
    location: {
      type: String,
      maxlength: 20,
    },
    about: {
      type: String,
      maxlength: 20,
      default: "About",
    },
  },
```

```
  { timestamps: true }
);

UserSchema.pre("save", async function () {
  const salt = await bcrypt.genSalt();
  this.password = await bcrypt.hash(this.password, salt);
});
```

Front End implementation

The part of a website that user interacts with directly is termed as front end. It is also referred to as the 'client side' of the application. It includes everything that users experience directly: text colors and styles, images, graphs and tables, buttons, colors, and navigation menu. HTML, CSS, Javascript, and ReactJS Framework are the languages used for Front End development. The structure, design, behavior, and content of everything seen on browser screen when websites, web applications, or mobile apps are opened up, is implemented by front End developers. Responsiveness and performance are two main objectives of the front End. The developer must ensure that the site is responsive i.e. it appears correctly on devices of all sizes no part of the website should behave abnormally irrespective of the size of the screen.

- **Login.jsx: this implementation represents login page from front end**

```
import React, { useState } from "react";
import { useDispatch } from "react-redux";
import { login } from "../../features/userSlice";
import { loginUser } from "../../API";
import useFetch from "../../hooks/useFetch";
import { setIsLoading } from "../../features/modalSlice";

const Login = ({ setIsRegistering }) => {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const dispatch = useDispatch();
  const customFetch = useFetch();
```

```
  const loginHandler = async (e) => {
    e.preventDefault();
    dispatch(setIsLoading(true));
    const data = await customFetch(loginUser, email, password);
    if (data) dispatch(login(data));
    dispatch(setIsLoading(false));
  };

  return (
    <form onSubmit={loginHandler} className="login">
      <label htmlFor="login-email">Email</label>
      <input
        type="email"
        id="login-email"
        placeholder="johndoe@example.com"
        value={email}
        onChange={(e) => {
          setEmail(e.target.value);
        }}
      />
      <label htmlFor="login-password">Password</label>
      <input
        type="password"
        id="login-password"
        placeholder="Top secret"
        value={password}
        onChange={(e) => {
```

```
            setPassword(e.target.value);
          }}
        />
        <button type="submit">Login</button>
        <p>
          Don't have an account?{" "}
          <span
            onClick={() => {
              setIsRegistering(true);
            }}
          >
            Register
          </span>
        </p>
      </form>
    );
};
export default Login;
```

- **Profile.jsx: this implementation represents profile page from front end**

```
import React, { useEffect } from 'react';
import Gallery from '../../components/Gallery/Gallery';
import Online from '../../components/Online/Online';
import ProfileCard from '../../components/ProfileCard/ProfileCard';
import { useParams } from 'react-router';
import CreatePost from '../../components/CreatePost/CreatePost';
import { useDispatch, useSelector } from 'react-redux';
import { fetchPosts } from '../../API';
import Cookies from 'js-cookie';
```

```
import { setUserPosts } from '../../features/postSlice';
import './profile.css';
import useFetch from '../../hooks/useFetch';
import Posts from '../../components/Post/Posts';

const Profile = () => {
	const { id } = useParams();
	const { token } = JSON.parse(Cookies.get('user'));
	const { userPosts } = useSelector(state => state.post);
	const isOwnProfile = id === useSelector(state => state.user.id);
	const mainRef = React.useRef(null);
	const nextPageLoaderRef = React.useRef(null);

	const dispatch = useDispatch();
	const customFetch = useFetch();

	useEffect(() => {
		(async () => {
			const data = await customFetch(fetchPosts, token, id);
			if (data) dispatch(setUserPosts(data.posts));
		})();
	}, [token, dispatch, id, customFetch]);

	const getNextPage = () => {
		nextPageLoaderRef.current?.getNextPage?.();
	};

	return (
```

```
            <section className='profile'>
                <article className='profile__left'>
                    <ProfileCard id={id} isOwnProfile={isOwnProfile} />
                    <Gallery />
                </article>
                <article className='profile__center'>
                    <div ref={mainRef} onScroll={getNextPage}>
                        {isOwnProfile && <CreatePost />}
                        {userPosts.length < 1 && <h2>No Posts</h2>}
                        <Posts
                            posts={userPosts}
                            containerRef={mainRef}
                            user={{ id }}
                            nextPageLoaderRef={nextPageLoaderRef}
                        />
                    </div>
                </article>
                <article className='profile__right'>
                    <Online />
                </article>
            </section>
        );
};
export default Profile;
```

5.2 Testing

This test plan describes the testing approach and overall framework that will drive the testing of our system specifically developed for Cancer Community Support Social Media. The objective of the test is to verify that the functionality of our system's specifications. The test will execute and verify the test scripts, identify, fix, and retest all high and medium severity defects per the entrance criteria, prioritize lower severity defects for future fixing.

5.2.1 Scope

This document will describe the approach, resources, and schedule of the testing activities possible at this time. This document will also identify the particulars of testing (what to test, what not to test), the tasks that need to be performed, and the method of testing. At this time, we are mainly focused on testing the functional requirements of the system.

5.2.2 Test Item

This section of the Test Plan deals with the testing of all those items that constitutes our system. The following is the list to be tested: Functional Testing: Functional testing will be performed to check the functions of the application. The functional testing is carried out by feeding the input and validates the output from the application.

The function of our system to be tested includes:

- User registration process
- User log in process
- Updating profile
- Posting information
- Report inappropriate posts
- Like and comment on posts
- Follow and Un-follow users
- Chatting process
- Video conferencing
- Admin tasks

5.2.3 Test Cases

Table 36: Sign up with valid data

Test Case ID	TC-001
Test Case Description	verify sign up using the system with valid data
Pre-requisites	The User must not use already registered email
Test Steps	1)The user navigates to the signup page 2) User Enters his name, email and password 3) User Clicks Signup
Test Data	First Name=”Dawit”

	Middle Name=”Gadisa” Last Name=”Waktola” Email=dawit@gmail.com Password=123456 Confirm Password=123456
Expected Result	The user directly sent to the home page of the website
Test Designed By	Dawit Gadisa
Test Executed By	Dawit Gadisa

Table 37: Sign up with invalid data

Test Case ID	**TC-002**
Test Case Description	verify sign up using the system with invalid data
Pre-requisites	The User registers invalid email address
Test Steps	1)The user navigates to the signup page 2) User Enters invalid email address 3) User Clicks Signup
Test Data	Email=abdulgmail.com Password=123456 Confirm Password=123456
Expected Result	The user will get an error message to enter a valid email address
Test Designed By	Abdul Ahmed
Test Executed By	Abdul Ahmed

Table 38: clicking signup button

Test Case ID	**TC-003**
Test Case Description	Verify that clicking signup button after entering all the required fields, submits the data to the server
Pre-requisites	The database should be setup
Test Steps	1)The user navigates to the signup page 2) User Enters all required fields 3) User Clicks Signup 4) Go to database and check for newly registered user
Test Data	First Name=”Abdul” Middle Name=”Ahmed” Last Name=”Mohammed” Email=abdul@gmail.com Password=123456 Confirm Password=123456
Expected Result	The newly registered user should be found in the database
Test Designed By	Abdul Ahmed
Test Executed By	Abdul Ahmed

Table 39: clicking signup button with empty field

Test Case ID	TC-004
Test Case Description	Check that not filling the mandatory fields and clicking the submit button will lead to a validation error
Pre-requisites	The required fields on the registration page are available
Test Steps	1)The user navigates to the signup page 2) User leaves the required fields empty 3) User Clicks Signup
Test Data	First Name=" " Middle Name=" " Last Name=" " Email= Password= Confirm Password=
Expected Result	Validation error message should be shown and asks the user to fill the required fields
Test Designed By	Abdulhafiz Juhar
Test Executed By	Abdulhafiz Juhar

Table 40: password less than six digits

Test Case ID	TC-005
Test Case Description	Check that entering password value less than six digits or alphabets
Pre-requisites	The password should not be short
Test Steps	1)The user navigates to the signup page 2) User enters the required fields 3) User enters password value less than six digits 4) user clicks signup button
Test Data	Password=1234 Confirm Password=1234
Expected Result	Error message that informs the user, the password is too short and reminds to be equal or greater than six digits
Test Designed By	Abdulhafiz Juhar
Test Executed By	Abdulhafiz Juhar

Table 41: confirm password

Test Case ID	TC-006
Test Case Description	Check that entering different values password and confirm password field
Pre-requisites	The confirm password should be same as password value entered first

Test Steps	1)The user navigates to the signup page 2) User enters the required fields 3) User enters password value 4) user enters a different value on confirm password field 5) user clicks signup button
Test Data	Password=123456 Confirm Password=345678
Expected Result	Error message that informs the user to enter similar value on the confirm password field
Test Designed By	Abdulhafiz Juhar
Test Executed By	Abdulhafiz Juhar

Table 42: login with valid data

Test Case ID	**TC-007**
Test Case Description	Verify login to the system with Valid Data
Pre-requisites	The user must be registered in the database
Test Steps	1) The User navigates to login page 2) Enters Email and Password 3) User Clicks Login Button
Test Data	Email=ubda@gmail.com Password=123456
Expected Result	The User Logins to the System according to his/her privilege
Test Designed By	Abdurahim Miftah
Test Executed By	Abdurahim Miftah

Table 43: login with invalid data

Test Case ID	**TC-008**
Test Case Description	Verify login to the system with invalid Data
Pre-requisites	The User is not registered or does not use valid email
Test Steps	1) The User navigates to login page 2) Enters Email and Password 3) User Clicks Login Button
Test Data	Email=ubdagmail.com Password=123456
Expected Result	The User will see an error message telling him to register or to correct his input
Test Designed By	Abdurahim Miftah
Test Executed By	Abdurahim Miftah

Table 44: login with blank fields

Test Case ID	**TC-009**

Test Case Description	Verify the login page for both, when the field is blank and sign in button is clicked
Pre-requisites	The fields are empty
Test Steps	1) The User navigates to login page 2) leaves the fields empty 3) User Clicks Login Button
Test Data	Email= Password=
Expected Result	The User will see an error message telling him to fill the blank fields before clicking the sign in button
Test Designed By	Abdurahim Miftah
Test Executed By	Abdurahim Miftah

Table 45: login with new updated password

Test Case ID	**TC-010**
Test Case Description	Verify if user is able to login with a new password only after he/she has changed the password
Pre-requisites	The older password is changed
Test Steps	1) The User navigates to login page 2) User enter the required fields 3) User Clicks Login Button 4) User goes to profile page 5) user clicks edit profile 6) User updates the older password with a new one 7) User logs out 8) User again sign in with the new password
Test Data	Email= ubdagmail.com Old Password= 123456 New Password= 12345678
Expected Result	User should be signed in with the updated new password
Test Designed By	Abdurahim Miftah
Test Executed By	Abdurahim Miftah

Table 46: login with password less than six digits

Test Case ID	**TC-011**
Test Case Description	Check that entering password value less than six digits or alphabets
Pre-requisites	The minimum size of the password is six digits
Test Steps	1)The user navigates to the sign in page 2) User enters the required fields 3) User enters password value less than six digits 4) user clicks sign in button
Test Data	Email=afiz@gmail.com Password=1234

Expected Result	Error message that informs the user, the password is too short and reminds to be equal or greater than six digits
Test Designed By	Abdulhafiz Juhar
Test Executed By	Abdulhafiz Juhar

Table 47: upload profile picture

Test Case ID	TC-012
Test Case Description	Verify that user can set profile pic uploaded from his or her computer
Pre-requisites	User has option to set profile picture
Test Steps	1)User sign in to the website 2) User navigates to profile page 3) User clicks set profile picture 4) User chooses picture from computer file 5) User clicks post button to set the chosen picture as profile picture
Test Data	Example= photo.png
Expected Result	The selected picture from the computer is set to be profile picture
Test Designed By	Abdul Ahmed
Test Executed By	Abdul Ahmed

Table 48: upload unsupported image file

Test Case ID	TC-013
Test Case Description	Verify that uploading image of unsupported type should lead to error message
Pre-requisites	Only supported files are uploaded on the site
Test Steps	1)User sign in to the website 2) User clicks create post 3) User chooses the supported file format from computer 4) User clicks the upload button
Test Data	Example= video.gif
Expected Result	Error message telling the user the selected file format is not supported on the website
Test Designed By	Abdulaziz Wendesen
Test Executed By	Abdulaziz Wendesen

Table 49: profile picture update notification

Test Case ID	TC-014
Test Case Description	Verify that change in profile picture should get reflected in each post/comment of the user's timeline

Pre-requisites	Users are allowed to update their profile account
Test Steps	1)User sign in to the website 2) User navigates to profile page 3) User clicks edit profile button 4) User clicks the upload button 5) The profile picture change on the user's account is displayed on timeline
Test Data	Example= photo1.png
Expected Result	The updated picture is displayed on the front page of the user's profile account
Test Designed By	Abdulaziz Wendesen
Test Executed By	Abdulaziz Wendesen

Table 50: edit profile information

Test Case ID	TC-015
Test Case Description	Verify that user can add/edit their account information displayed to other users
Pre-requisites	Users are allowed to upload and create new posts
Test Steps	1)User sign in to the website 2) User clicks create post 3) Users write or choose file to be posted 4) User clicks the post button 5) The new posted information is displayed on front page of the website
Test Data	Example= photo.png
Expected Result	The updated picture is displayed on the top of the homepage news feed
Test Designed By	Dawit Gadisa
Test Executed By	Dawit Gadisa

Table 51: see new posts

Test Case ID	TC-016
Test Case Description	Verify that users can see the all the post in their timeline
Pre-requisites	Users are allowed to see every post by other users who follow them
Test Steps	1)User sign in to the website 2) User is navigated to the home page 3) User sees new posts on top of the homepage
Test Data	Example= "Hello Everyone"
Expected Result	The uploaded picture or text is displayed on the top of the homepage news feed
Test Designed By	Dawit Gadisa
Test Executed By	Dawit Gadisa

Table 52: view likes and comments

Test Case ID	TC-017
Test Case Description	Verify that users can see comments, likes and reactions in the posts present in the news feed
Pre-requisites	Posts need to have likes, comments or reactions
Test Steps	1)User sign in to the website 2) User is navigated to the home page 3) User sees new posts on top of the homepage 4) User sees the likes and comments underneath the posts
Test Data	Example= 1000 likes, 50 comments
Expected Result	The number of likes and comments displayed underneath each post
Test Designed By	Abdulhafiz Juhar
Test Executed By	Abdulhafiz Juhar

Table 53: comment and like posts

Test Case ID	TC-018
Test Case Description	Verify that users can post comments, like and react to the posts present in their timeline
Pre-requisites	Posts should allow user to like and comment
Test Steps	1)User sign in to the website 2) User is navigated to the home page 3) User sees new posts on top of the homepage 4) User clicks like and comment buttons 5) write some reaction comments on the comment section 6) user clicks post comment button
Test Data	Example= “Be strong” (comment)
Expected Result	The users’ like and comment is displayed on the post
Test Designed By	Abdulhafiz Juhar
Test Executed By	Abdulhafiz Juhar

Table 54: search users

Test Case ID	TC-019
Test Case Description	Verify that the user can search for friends
Pre-requisites	The user to be searched should be registered user
Test Steps	1)User sign in to the website 2) User is navigated to the home page 3) User writes the username of the friend to be searched on the search friends field 4) User clicks search button 5) Users with the searched user names are displayed vertically 6) User chooses the friend he/she is looking for

Test Data	Search a friend= “Dave”
Expected Result	Users with searched user names are displayed on the screen
Test Designed By	Dawit Gadisa
Test Executed By	Dawit Gadisa

Table 55: follow users

Test Case ID	**TC-020**
Test Case Description	Verify that users can follow any user by visiting their page
Pre-requisites	The user to be followed should be registered user
Test Steps	1)User sign in to the website 2) User is navigated to the home page 3) User writes the username of the friend to be searched on the search friends field 4) User clicks search button 5) Users with the searched user names are displayed vertically 6) User chooses the friend he/she is looking for 7) User is navigated to the chosen user page 8) User clicks follow button
Test Data	Example= “You are following Dave now”
Expected Result	Users are allowed to follow any registered user on the website
Test Designed By	Dawit Gadisa
Test Executed By	Dawit Gadisa

Table 56: un-follow users

Test Case ID	**TC-021**
Test Case Description	Verify that the user can unfollow any existing friend
Pre-requisites	The user to be unfollowed should be a followed user
Test Steps	1)User sign in to the website 2) User clicks profile page 3) User then clicks the following button 4) User searches for users he/she is following 5) Users choose the searched user 6) User clicks unfollow button
Test Data	Example= “You unfollowed Dave”
Expected Result	Message that informs the user that he/she unfollowed the following user
Test Designed By	Dawit Gadisa
Test Executed By	Dawit Gadisa

Table 57: notification

Test Case ID	TC-022
Test Case Description	Verify that users receive notification when new private message is sent to their accounts
Pre-requisites	The users who exchange should be registered users
Test Steps	1)User sign in to the website 2) User is navigated to the homepage 3) User receives a rectangle new message notification at the bottom of the homepage
Test Data	Example= “you have new message from Dawit”
Expected Result	Pop up new message notification at the bottom right of the homepage
Test Designed By	Dawit Gadisa
Test Executed By	Dawit Gadisa

Table 58: chatting process

Test Case ID	TC-023
Test Case Description	Verify there is a minimum of two Users who should be available for the chat
Pre-requisites	The minimum two users to chat should be registered users
Test Steps	1)User sign in to the website 2) User clicks chat icon 3) User chooses the user to send the message 4) users writes the message that he/she wants to send to the other user
Test Data	Example= “Hello Abdi”
Expected Result	Message is sent with no error response
Test Designed By	Abdul Ahmed
Test Executed By	Abdul Ahmed

Table 59: chatting with internet access

Test Case ID	TC-024
Test Case Description	Verify that any Medium like Internet, Wifi, Bluetooth, Public Switched Telephone Network, W-Lan Network, Lan Network, etc. should be available for the Chat functionality
Pre-requisites	There should be an internet access for the sender to send the message and for the receiver to receive the message
Test Steps	1)User sign in to the website 2) User clicks chat icon 3) User chooses the user to send the message

	4) users writes the message that he/she wants to send to the other user
Test Data	Example= “Hello Abdi”
Expected Result	Message is sent with no error response
Test Designed By	Abdul Ahmed
Test Executed By	Abdul Ahmed

Table 60: chatting without internet access

Test Case ID	**TC-025**
Test Case Description	Verify that any Medium like Internet, Wifi, Bluetooth, Public Switched Telephone Network, W-Lan Network, Lan Network, etc. is not available for the Chat functionality
Pre-requisites	There should be an internet access for the sender to send the message and for the receiver to receive the message
Test Steps	1)User sign in to the website 2) User clicks chat icon 3) User chooses the user to send the message 4) users writes the message that he/she wants to send to the other user
Test Data	Example= “Hello Abdi”
Expected Result	There is no Internet connection to send this message
Test Designed By	Abdul Ahmed
Test Executed By	Abdul Ahmed

Table 61: user status

Test Case ID	**TC-026**
Test Case Description	Verify that the Status (Active, Inactive) of the User is changing or not
Pre-requisites	User should be using the website at specific moment to be Active
Test Steps	1)User sign in to the website 2) User is navigated to the homepage 3) User is able to see active users with green circle icons on the right side of the homepage
Test Data	Example= “Abdurahim is online”
Expected Result	Green circles for active users and grey color circles for inactive users
Test Designed By	Abdulhafiz Juhar
Test Executed By	Abdulhafiz Juhar

Table 62: send message to offline user

Test Case ID	TC-027
Test Case Description	Verify that the User is able to send messages to other offline Users
Pre-requisites	Users able to send message for offline registered users
Test Steps	1)User sign in to the website 2) User clicks the message icon 3) User picks the offline user and send a specific message 4) the offline user sees the new message when he/she becomes active
Test Data	Example= "Dave talk to me when you see this later"
Expected Result	Message sent to offline user with no error message
Test Designed By	Abdulhafiz Juhar
Test Executed By	Abdulhafiz Juhar

Table 63: host video conferencing

Test Case ID	TC-028
Test Case Description	Verify that the User is able to host video conferencing
Pre-requisites	User should allow video and audio options
Test Steps	1)User sign in to the website 2) User clicks the video conference icon 3) User clicks host meeting button 4) User copies the attendees link 5) User creates the name for the meeting 6) User joins the meeting and start the video conference
Test Data	Meeting Title: Fighting Cancer
Expected Result	Video conference is hosted online
Test Designed By	Dawit Gadisa
Test Executed By	Dawit Gadisa

Table 64: block users on video conference

Test Case ID	TC-029
Test Case Description	Verify that the video conference host can block sound and video of the video conference attendees
Pre-requisites	User should be the host of the video conference to block video and sound of other users on the video conference
Test Steps	1)User sign in to the website 2) User clicks the video conference icon 3) User clicks host meeting button 4) User copies the attendees link 5) User creates the name for the meeting 6) User joins the meeting and start the video conference

	7) Host user clicks the uses icon 8) Host can choose the user he/she wants to block the sound and video 9) Host clicks the sound and video block icons
Test Data	Meeting Title: Fighting Cancer
Expected Result	The blocked users are unable to speak or shown on the video conference
Test Designed By	Abdurahim Miftah
Test Executed By	Abdurahim Miftah

Table 65: send video conference link to attendee

Test Case ID	**TC-030**
Test Case Description	Verify that users can attend video conference if the host send the link for the video conference
Pre-requisites	User should have the link to join the video conference
Test Steps	1)User sign in to the website 2) User clicks the video conference icon 3) User clicks host meeting button 4) User copies the attendees link 5) Host sends the copied link to attendee users 6) the attendees clicks that specific link to join the video conference
Test Data	Meeting Title: Fighting Cancer
Expected Result	Users can join the video conference by clicking the link generated by the host of the video conference
Test Designed By	Dawit Gadisa
Test Executed By	Dawit Gadisa

Table 66: group and private chat on video conference

Test Case ID	**TC-031**
Test Case Description	Verify that users can chat on the video conference
Pre-requisites	User should join to video conference to chat on the conference
Test Steps	1)User sign in to the website 2) User joins the meeting as a host or attendee 3) User clicks chat icon 4) User can choose a public or private message 5) Users writes the message he/she wants to deliver 6) User clicks send button
Test Data	Meeting Title: Fighting Cancer
Expected Result	Users are able to chat in public or private on video conferencing
Test Designed By	Abdul Ahmed
Test Executed By	Abdul Ahmed

Table 67: admin control

Test Case ID	TC-032
Test Case Description	Verify that admin can see posts and analysis
Pre-requisites	Admin should have a unique access to the system
Test Steps	1)Admin signs in to the page 2) Admin is able to see new posts displayed in numbers
Test Data	Example: 40 total posts
Expected Result	Updated number of total number of posts displayed
Test Designed By	Abdurahim Miftah
Test Executed By	Abdurahim Miftah

Table 68: admin delete posts

Test Case ID	TC-033
Test Case Description	Verify that admin is able to delete reported post
Pre-requisites	The post should be reported to be deleted
Test Steps	1)Admin signs in to the page 2) Admin views the reported post 3) Admin clicks delete post button
Test Data	Example: “this post is deleted successfully”
Expected Result	The post is deleted from the database
Test Designed By	Abdurahim Miftah
Test Executed By	Abdurahim Miftah

Table 69: admin ban users

Test Case ID	TC-034
Test Case Description	Verify that admin is able to ban reported user
Pre-requisites	The user should be reported to be banned
Test Steps	1)Admin signs in to the page 2) Admin views the reported user 3) Admin clicks ban user button
Test Data	Example: “this user is banned successfully”
Expected Result	The user is banned from using the system
Test Designed By	Abdulaziz Wendesen
Test Executed By	Abdulaziz Wendesen

Table 70: report inappropriate posts

Test Case ID	TC-035
Test Case Description	Verify that user is able to report post
Pre-requisites	The reported user should be registered user
Test Steps	1)User signs in to the website 2) User is navigated to the homepage 3) Admin clicks the three dot lines on the posts 4) User clicks report this post button if the post is inappropriate
Test Data	Example: “This post is not appropriate for this environment”
Expected Result	The reported post is sent to the admin page
Test Designed By	Abdulaziz Wendesen
Test Executed By	Abdulaziz Wendesen

5.2.4 Testing approaches

This application is tested with the test case input and its correct output. For example, to test the login functionality of the web application, the tester will enter the user login name and its password. The expected output is the user is logged into the system. The approaches to testing include:

- Unit testing
- Integration testing
- System testing

5.2.5 Item Pass/ Fail Criteria

Test cases executed on the application will pass if they meet the specific requirements as mentioned in the Requirement Document. The item is said to fail the system if the results that the test case produces does not match the desired result or the expected outcomes. Test cases are evaluated independently of one another, not affect the evaluation of the other test cases.

6 Conclusion and Future Recommendations

6.1 Conclusion

Soul Fighting Cancer is a multi-platform web and mobile application that connects Cancer patients with each other and health professionals in order to increase survival and quality of life. As the creators of this idea, we believe that every disease is stuck in every people who suffer from different kinds of diseases. It is very simple to win every disease for every people, if they work on their mentality. The only thing people need to do just believe that they can win any disease. As we all know the awareness about cancer is very less and people believe it can never be cured. The fact we see in different parts of the world is completely different. The only thing cancer patients need from everyone is simply moral and emotional support. In order to help Cancer patients, moral and emotional support is the best and preferable way. So our system

makes that process more effective and efficient by providing simple ways to connect Cancer patients with other patients and cured people from the disease in order to share their experience. The users of the system also get the privilege to connect in online video conference meetings with health professionals and other users. Some questions might come up to people's mind the idea of using other social media platforms: like Facebook, Instagram, twitter, and others. But how much of inappropriate post do we see in those social media platforms per day? It is very high. The main aim of our system is to build only a peaceful and positive environment for the patients. So we handle any inappropriate posts and information on our system.

6.2 Future Work

There are some things that we have not done in the present but that we will do in the future. The first of these is that we have worked in only one language. As you know, the people of Ethiopia are multilingual, so working in one language is not appropriate. In addition, since the user is still a beginner in technology, there are many things that are useful but difficult to use. We will include them. Including these features will bring more awareness about Cancer in our society and help cancer patients to be mentally brave, positive, and believers of surviving off of the diseases.

References

[1] National Cancer Institute, “What is Cancer?”

[2] National Cancer Institute, “Cancer Support Groups and their Impacts.”

[3] [4] Rahel Abraraw, Abdisa Boka, Roza Teshome, Addis Yeshiambel, Social networks & quality of life among female breast cancer patients at Tikur Anbessa Specialized Hospital, Addis Ababa, Ethiopia 2019

[5] Cancer and Mental Health, Mental Health America

[6] https://www.abcfonline.org/

[7] GetX and flutter documentation

Appendix

Some of the questions asked during the interview while gathering requirements from different health centers and cancer associations. As our elicitation technique was an interview, the questions were not like a simple Yes/ No. We tried to interview different health organizations and merge the information we get in order to document them.

1. Tell us about your cancer organization or association?
2. How do you provide awareness for the patients about cancer?
3. How much is the awareness about cancer among the community?
4. Is the availability of information is enough depending on cancer?
5. Are there any cancer support groups in Ethiopia? How do people with cancer join those support groups?
6. If there are no cancer support groups in Ethiopia, what makes it difficult to create such groups?
7. What is the struggle of cancer patients to find and join different support groups?
8. Do cancer patients fear to join any support groups? Why?
9. What is the use of social network and interaction among cancer patients?
10. Which type of cancer is difficult to find information on?
11. What is the rate of cancer patients who come to get treatments?
12. How many cancer patients are there in Ethiopia? (Statistical data)
13. How many people get diagnosed with different types of cancer annually in Ethiopia?
14. What do you find difficult in treating patients with cancer?
15. What do you think is the biggest weakness on cancer treatment in Ethiopia?
16. How do you manage cancer patient's time in order to make them take their medicines on time?
17. What makes your organization different from other cancer organizations in creating awareness about cancer for the community?
18. What does the patient's social interaction look like before and after joining this cancer association?
19. How do you explain patient's mentality strength after joining this cancer association?
20. How do the patients hear and arrive to this cancer association?
21. What kind of patient types arrives at this association? (Based on age, living area, literacy, linguistics)
22. Do the patients share their experience?
23. Is there any platform provided in this cancer association for the patients to share their experiences with each other?
24. What social life impacts do the patients face?
25. Do you remember any cancer survivor? Can you share us a little bit of their story?
26. How do you think our proposed system will help the cancer patients?