



**SZABIST**  
UNIVERSITY

## Final Year Project Report

---

**ILAAJ**

---

### **Project Team:**

Arslan Ali 2012110

Naveen Kingrani 2012127

### **Project Supervisor:**

Sheikh Usama Khalid

**Date:** June 10<sup>th</sup>, 2024

Submitted in the partial fulfillment of the requirements for the degree of  
Bachelor of Science in Computer Science  
in the  
Faculty of Computing and Engineering Sciences

Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology  
(SZABIST)  
Karachi Campus

---

## Plagiarism Free Certificate

This is to certify that, I am Arslan Ali S/o Soorih Suhag, group leader of FYP under the registration no. 2012110 at Computer Science Department, SZABIST, Karachi. I declare that my FYP report is checked by my supervisor.

Date: June 10<sup>th</sup> 2024 Name of Group Leader: Arslan Ali Signature: \_\_\_\_\_

Name of Supervisor: Sheikh Usama Khalid

Co-Supervisor: \_\_\_\_\_

Designation: Lecturer

Designation: \_\_\_\_\_

Signature: \_\_\_\_\_

Signature: \_\_\_\_\_

HOD: \_\_\_\_\_

Signature: \_\_\_\_\_

## **Declaration of Authorship**

We, Arslan Ali [2012110] & Naveen Kingrani [2012127] declare that this report titled, “ILAAJ” and the work presented in it are our own. We confirm that:

This work was done wholly or mainly while in candidature for a bachelor’s degree at this University.

Where any part of this report has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.

Where we have consulted the published work of others, this is always clearly attributed.

Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this report is entirely our own work.

We have acknowledged all main sources of help.

Where the report is based on work done by ourselves jointly with others, we have made clear exactly what was done by others and what we have contributed ourselves.

Signed:

---

Date: June 10<sup>th</sup>, 2024

## Project Description

ILAAJ is an health application which is developed to facilitate patients. And industry which we targeted is health and medical industry. The purpose of this app to create a platform which facilitates patients, doctors even normal users. This user interface of this app is very user friendly so everyone can easily use it. Through this app user can book doctor appointments, order medicines and lab products, chat with health professionals to get valuable guidance, add reminders (alarms), see health articles uploaded by professional doctors, access emergency SOS feature and user feedback / suggestions. Also this app serves as our business startup where we are selling pharmaceutical and lab products via admin panel. This app has 3 panels i.e user panel, doctor panel and admin panel.

User panel, through user panel user can register themselves on our app and login. They can also update their profile. In user panel home page there are multiple features provided to user such as 1) doctor appointment; user can see doctor info, their time slots, location, rating and also via this feature they can chat with doctors. 2) Medicines: by this feature user can order medicines. 3) lab: via this feature user can order lab products and devices. 4) Health Articles: in this feature user can see posts by doctors related health alerts. 5): SOS Feature: in this feature user can share their current locations with emergency contact via sms, call, whatsapp, also user can add multiple emergency numbers. 6) Feedback & Suggestions: In this feature user can send complaints and chat with Admin. Also in user panel user can see their appointment history, order details, remainders, favourite products. Alongside that they can cancel orders, appointments too.

On the other hand, In Doctor panel doctor first have to register themselves in app and add their important details, after approval from admin panel they will be able to login. Doctor can see patients who has booked appointments and their details. Also doctor can interact with patients via chat and can be also able to cancel or change appointments. Doctors can add health articles, update and delete. Also they will be able to update their profile and see appointments history in doctor panel.

In Admin panel, admin can add, update and delete products, see order details of users and can mark order as delivered after successfully delivering order. Also admin can perform CRUD operations in Admin panel like add and delete users. Admin can also chat with users and answer their complaints and suggestions.

In the end, this app act as bridge between patients and doctors. It serves as a platform that aims to revolutionize the way patients and doctors can interact with each other and benefit from it. We also received positive feedback from multiple doctors when we showed them our app and discussed its potential.

## **Acknowledgement**

In the name of ALLAH the most beneficent and merciful who gave us the knowledge and courage to work on this research area.

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project.

We would first like to thank our supervisor Mr. Sheikh Usama Khalid of the Computer Science faculty at Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology. The door to his office was always open whenever we ran into a trouble spot or had a question about our research or writing. He consistently helped, cooperated and motivated us throughout the research.

We would like to thank to our teachers who guided us in the light of their knowledge and experience. We would also like to express our gratitude to our loving parents and family members who helped and gave us encouragement. Furthermore, we would also like to acknowledge with much appreciation the crucial role of the staff of SZABIST, who gave the permission to use all required equipment and the necessary materials to complete the project.

At the end, we would like to thank Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology for providing us with such an inspiring environment. The quality education, the cooperative faculty members and the challenging environment have always motivated and boosted the confidence level of each and every student who has been a part of Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology.

# Table of Contents

Plagiarism Free Certificate.....	1
Declaration of Authorship.....	2
Project Description.....	3
Acknowledgement.....	4
Project Proposal.....	8
1. Introduction.....	8
2. Objective.....	8
3. Problem Description.....	8
4. Methodology.....	8
5. Project Scope.....	9
6. Feasibility Study.....	10
7. Solution Application Areas.....	10
8. Tools/Technology.....	10
9. Expertise of the Team Members.....	10
10. Milestones.....	10
11. Project Schedule.....	11
12. Work Breakdown Structure.....	11
13. References.....	12
Software Requirement Specification.....	13
1. Introduction.....	13
1.1 Purpose.....	13
1.2 Document Conventions.....	13
1.3 Intended Audience and Reading Suggestions.....	13
1.4 Product Scope.....	13
1.5 References.....	13
2. Overall Description.....	14
2.1 Product Perspective.....	14
2.2 Product Functions.....	14
2.3 User Classes and Characteristics.....	15
2.4 Operating Environment.....	15

2.5	Design and Implementation Constraints.....	15
2.6	User Documentation.....	15
2.7	Assumptions and Dependencies.....	15
3.	External Interface Requirements.....	16
3.1	User Interfaces.....	16
3.2	Hardware Interfaces.....	16
3.3	Software Interfaces.....	16
3.4	Communications Interfaces.....	16
4.	System Features.....	17
4.1	Login/Signup.....	17
4.1.1	login.....	17
4.1.2	(b) Sign up.....	18
4.2	Doctor Appointment.....	19
4.3	Chat box.....	20
4.4	Add Pharmaceutical Product.....	21
4.5	Search Bar.....	22
4.6	Admin (Block User / Doctor Account).....	23
4.7	Profile management.....	24
4.8	Health article.....	25
4.9	Appointment reminder.....	26
4.10	S.O.S feature.....	27
5.	Other Nonfunctional Requirements.....	27
5.1	Performance Requirements.....	27
5.2	Safety Requirements.....	28
5.3	Security Requirements.....	28
5.4	Software Quality Attributes.....	28
5.5	Business Rules.....	28
6.	Other Requirements.....	28
6.	Admin (Block User / Doctor Account).....	34
	Software Design Specification.....	37
1.	Introduction.....	37
1.1	Purpose of the Document.....	37
1.2	Scope of the Development Project.....	37
1.3	Definitions, Acronyms and Abbreviations.....	37

1.4	References.....	37
1.5	Overview of the Document.....	37
2.	System Architecture Description.....	38
2.1	Section Overview.....	38
2.2	General Constraints.....	38
2.3	Data Design.....	38
2.4	Program Structure.....	39
3.	Detailed description of components.....	39
3.1	Section overview.....	39
3.1.1	Login.....	40
3.1.2	Sign up.....	41
3.1.3	Dashboard.....	42
3.1.4	Doctor Profile:.....	43
3.1.5	Doctor appointment.....	44
3.1.6	Chat box.....	45
3.1.7	Health article:.....	46
3.1.8	Profile management.....	47
4.	User interface design.....	48
4.1	Section overview.....	48
4.2	Interface Design Rules.....	48
4.3	GUI Components.....	49
4.4	Detailed description.....	49
5.	Reuse and relationship to other products.....	49
6.	Design decisions and tradeoffs.....	49
7.	Pseudo code for Components.....	50
7.1	Login.....	50
7.2	Sign Up.....	50
7.3	Doctor appointment.....	50
7.4	Dashboard.....	50
7.5	Profile modification.....	51
7.6	Health article.....	51
7.7	Search box.....	51
7.8	Chat box.....	51
8.	Appendices.....	52



8.1	Class diagram.....	52
8.2	Object Diagram.....	53
8.3	State chart Diagram.....	54
8.4	Activity Diagram.....	55
8.5	Sequence Diagram.....	61
8.6	Collaboration Diagram.....	67
8.7	Use-Case Diagram.....	72
8.8	Component diagrams.....	75
8.9	Deployment diagram.....	76
8.10	System block diagram.....	77
8.11	ERD.....	78
	Software Testing Document.....	79
1.	Introduction.....	79
1.1	Purpose of this Document.....	79
1.2	Intended Audience.....	79
1.3	Document Convention.....	79
1.4	Project Overview.....	79
2.	Test Case Design and Description.....	80
2.1	Sign up.....	80
2.2	Login.....	81
2.3	User Profile Modification.....	82
2.4	Doctor Appointment.....	83
2.5	Search.....	84
2.6	Chat box.....	85
2.7	Add Pharmaceutical product.....	85
2.8	S.O.S Feature.....	86
	User Manual.....	87
1.	General Information.....	87
1.1	System Overview and Project Scope.....	87
1.2	Platforms.....	87
1.3	Acronyms and Abbreviations.....	87
1.4	System Functions and Configuration's.....	87
2.	System Summary.....	88
2.1	System Overview.....	88

2.2	User Level Access.....	88
2.3	Contingencies.....	88
3.	System Features.....	88
3.1	Sign up.....	89
3.2	Sign In.....	90
3.3	Profile Modification.....	91
3.4	User dashboard.....	92
3.5	Doctor appointment.....	93
3.6	Chat box.....	94
3.7	Order Medicines and Lab products.....	95
3.8	Feedback.....	97
3.9	Health Article.....	98
3.10	SOS Feature.....	99
3.11	Admin Panel.....	100
	Project Meeting Log Sheet.....	101

# Project Proposal

## 1. Introduction

Ilaaj (إلاج) is an Medical Health App as our Final Year Project (FYP). After the wake of the COVID-19 pandemic and the increasing importance of telehealth, a robust and user-friendly medical app is crucial to bridge the gap between patients and healthcare providers. Our app is going to be a platform through which user can book appointments with doctors, they can take consultation from health specialists, can also order medicines, and they can use it for emergency and also for health tips. More importantly they can also use our app in multiple languages.

## 2. Objective:

The primary objectives of this FYP are as follows:

1. Design: Create an intuitive, user-friendly interface app which user can easily use.
2. Features: Provide such features that user can manage his profile, find and search doctors, book appointments, order medicines, telehealth consultations, emergency service features, meditation reminders, health education and multiple languages option.
3. Security: Implement robust security measures to safeguard users' confidential health data.
4. Usability Testing: Conduct usability tests to ensure the app is accessible to users of all ages and backgrounds.
5. Scalability: Design the app architecture for scalability, accommodating a growing user base.

## 3. Problem Description:

As we all know since covid-19 the demand of online apps has increased nowadays people are preferring online mode for most of things like online shopping, food service and many more things. But in terms of medical health there are not many applications in pakistan that can facilitate patients at their door step like there are some apps by which you can order medicines and book appointments but for each function there's different app. But Question arises Why our App? Answer is through our app they can get all facilities on single platform instead of many apps for single features and we are offering way more features than other apps. Like you can book appointments, take consultation, order medicine, meditation reminder, health guidance ,emergency features and they can also give feedback. Another Question How you can get all this? All you need is to download our app and get started. And our UI is going to be user-friendly very easy to use.

## 4. Methodology:

The development of our Medical Health App will follow an agile methodology to ensure flexibility and responsiveness to changing requirements. The project will consist of the following phases:

1. Requirements Analysis: Gather detailed requirements through user interviews, surveys, and market research.

2. Design: Create wireframes, prototypes, and user interface designs for the app.
3. Development: Build the app's front-end and back-end, incorporating security features and API integrations.
4. Testing: Perform comprehensive testing, including functional, usability, security, and performance testing.
5. User Feedback: Collect user feedback and make iterative improvements based on usability testing.
6. Deployment
7. Documentation: Prepare detailed documentation for users.

## **5. Project Scope:**

ILAAJ Medical Health App aims to provide a medical health solution to users, incorporating features such as:

### **FEATURES:**

**User Registration and Profiles:** Allow users to create and edit their profiles.

**Search and Find Doctors:** Like user can see categories of doctors according to their specialization and can find appropriate doctor.

**Appointment Booking:** Allow users to schedule appointments with doctors, clinics, or hospitals.

**Medicine Service:** User can order medicines through our app.

**Telehealth Consultations:** Enable video or text-based consultations with healthcare professionals.

**Appointment Reminders:** Set up medication schedules and send reminders to users.

**Health Article:** Share articles, videos, and tips on various health topics.

**Emergency Services:** Include an SOS feature for immediate medical assistance.

## 6. Feasibility Study

With above defined scope, would you be able to meet your project schedule? Do mention following aspects:

- i. **Risks Involved:** The major risk involved in these kind of project is will it reach number of audience. Like we will try our best to make sure our app will be accessible to everyone.
- ii. **Resource Requirement:** We would most likely be needing domain hosting to run the following platform in it.

## 7. Solution Application Areas

Create a platform for patients where they can book appointments, take consultation, order medicines, meditation reminders, emergency services, and many more.

## 8. Tools/Technology

### Technologies used:

Technologies we are going to use in our project are Flutter, Dart, Mongo Db MySQL.

Whereas IDE / SW we are going to use are VS CODE, Android Studio.

## 9. Expertise of the Team Members

**Arslan Ali:** I have great experience in Web development, creating UI, and also in developing apps.

**Naveen Kingrani:** I have great experience in Designing, data structures, and also in web development.

Both of us have worked on many projects together throughout our journey in szabist. We have good understanding between each other and know our expertise.

## 10. Milestones

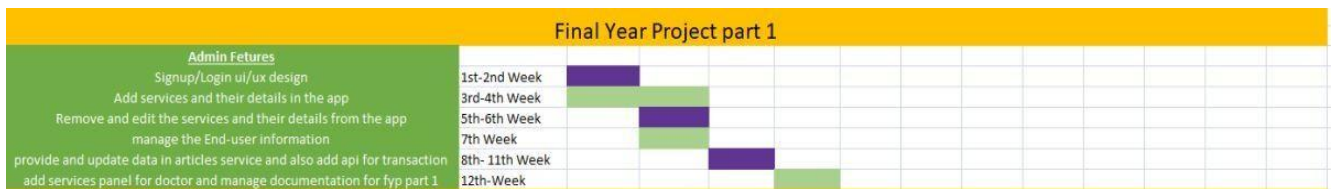
- A. Documentation Research
- B. Frontend Development
- C. User Registration
- D. Database Development
- E. User and Admin Panel
- F. Adding features
- G. Testing
- H. Deployment

## 11. Project Schedule

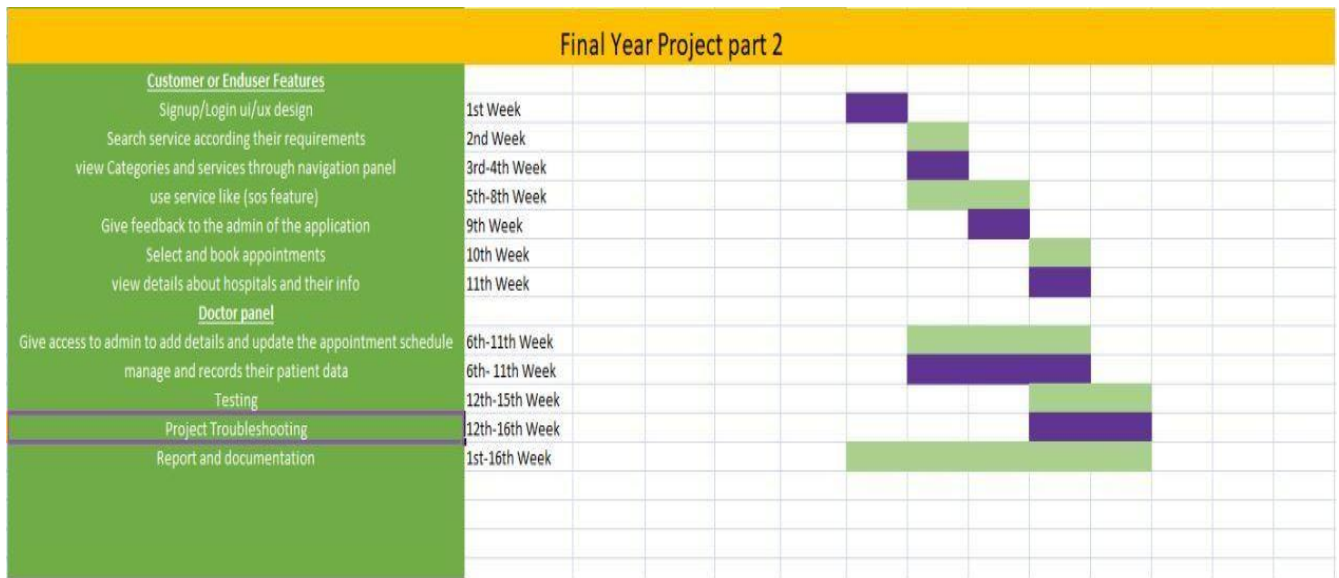


## 12. Work Breakdown Structure

FALL SEMESTER:



## SPRING SEMESTER:



**Deliverables:** 2 WBS, each representing 1-semester tasks, milestones along with the timeline details.

## 13. References

- A) <https://www.w3schools.com/> B)
- B) <https://flutter.dev/> C)
- C) <https://www.codecademy.com/>

# Software Requirement Specification

## 1. Introduction

### 1.1 Purpose

The aim of Mobile Health Care Application (illaj) is to make our lives easier and to improve the Quality of life. Mobile Health Care System is a system to monitor the user's health, Based on the data collected from the users and the doctors. Patient can manage their Own health and gain benefit from accessing the expertise of health care professionals at their fingertips when and where they need. Users are provided consultation and Guidance based on the collected information. The user can view information on the Available medicines, order them, and also book an appointment to the relevant doctor and also able to chat with particular doctor. The key feature is S.O.S that you can use as for emergency service once you click on this feature the first responder will be dispatched to your location this application will provide a best solution for digital health care system for the people.

### 1.2 Document Conventions

Format: IEEE Standard

- Font Family: Times
- Main Headings Times, 18sp
- Subheading times, 14sp
- Text Size: 11sp, 14sp, 18sp, 12sp
- Text Style: Bold
- Text Color: Black#000000
- Text Alignment: Left Align.

### 1.3 Intended Audience and Reading Suggestions

Our different audience is our supervisor, Faculty, Team Members and Users.

### 1.4 Product Scope

The project will be designed for a specific online health care platform known as “(ilaaj)”. This project will help to personalize the health care for a particular user. The project Targets the general public with smart devices and also provides them with the precise Information about their health. The data thus collected and analyzed can be used by the doctors to determine patient's health. Users can also use it to get general information about their condition. Overall, the scope of the project revolves around the users, Doctors and researchers. As the technology improves, it's scope and impact in health Sector will expand.

### 1.5 References

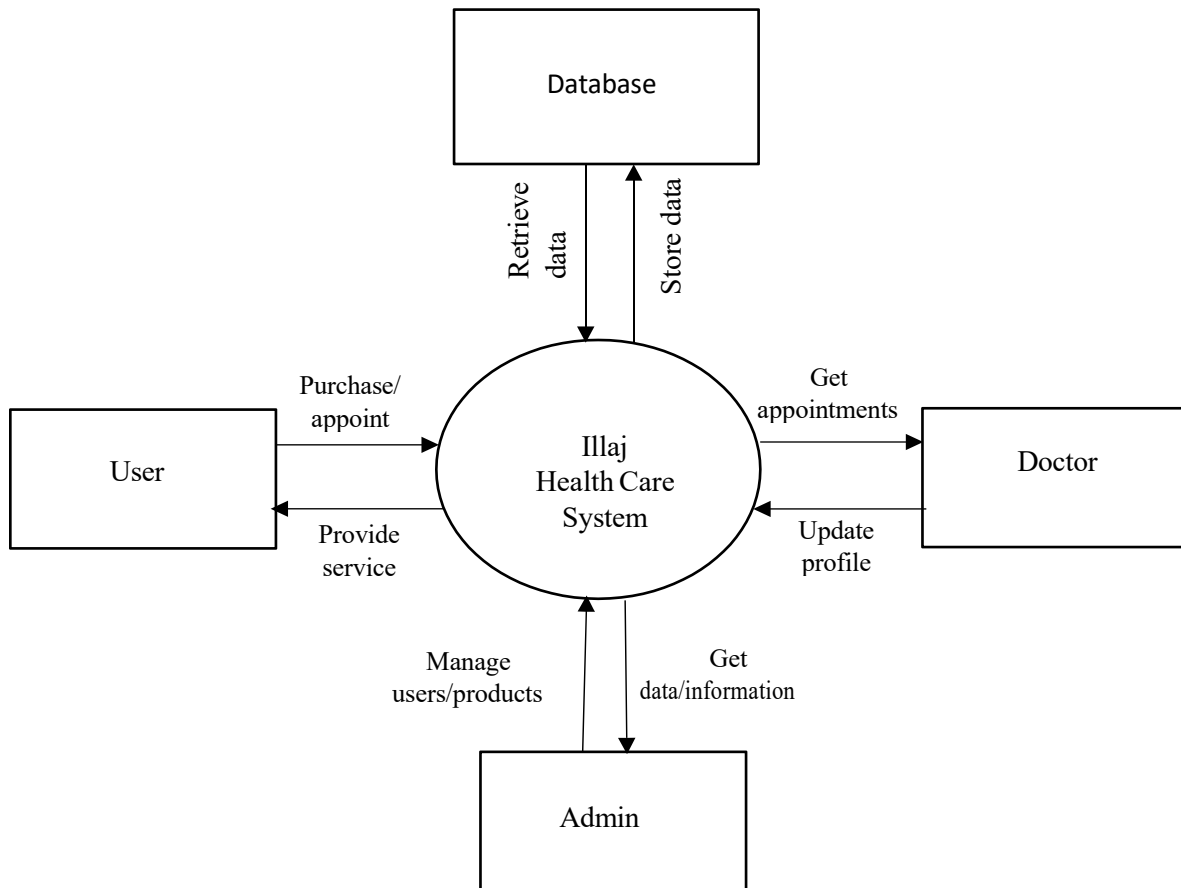
1. Stackoverflow.com



2. <https://www.coursera.org/courses>
3. <https://docs.flutter.dev>
4. <https://www.mongodb.com>
5. <https://www.github.com>

## 2. Overall Description

### 2.1 Product Perspective



### 2.2 Product Functions

The Salient features of our product are:

#### FYP 1:

- Signup & login for the Admin, User and Doctor account.
- Doctor appointment
- Search Bar to find Doctor
  - Filter Doctor to specify categories
- Add Doctor
- Profile Management and verification
- Chat box
- Health article

#### FYP 2:

- Admin panel
- Pharmaceuticals products
  - Add product/Medicine
- User purchase log
- SOS feature
- Appointment reminder

## 2.3 User Classes and Characteristics

The characteristics of our different type of users are that they will have a basic understanding of using smart phones and applications and will be able to use our chat box as well as other tools that we offer, while others with less knowledge will contact Illaj for any type of queries. The agent will be well-acquainted with our software and will be fully aware of the features and working of features that are listed on our app for the purpose of appointment and buying product. The owner or administrator will be fully aware of the system of the application.

## 2.4 Operating Environment

The Programming IDE that we are utilizing is Android Studio, and the operating environment that will be used is Android devices with a minimum SDK API 32: Android 11 (Red Velvet Cake or R) that will let our app to run.

## 2.5 Design and Implementation Constraints

The system is required to be complex in terms of features like bidding implementation on an ecommerce platform. There are no as such limitation in terms of hardware the application will run smoothly on any android version 11 or above and phone requirement to have space for application installment and minimum 2 GB ram needed to support the application process smoothly. The app will be using English language as a common language to communication these days. The admin will be responsible for the maintaining the application and up gradation.

## 2.6 User Documentation

- SRS (software requirement specification).
- SDS (software design specification).
- STD (software test-case document).

## 2.7 Assumptions and Dependencies

The application will be designed for android users but we assume to have cross platform functionality. Users i.e. doctors/end-user who will use this platform will be provided with enough knowledge and tutorials to use the application effectively. As a health care application will be dependent on internet connectivity all the time

## 3. External Interface Requirements

### 3.1 User Interfaces

The Flutter-based Android applications are crafted using a rich set of customizable widgets. These widgets follow Google's Material Design guidelines like widgets 'AppBar', 'BottomNavigationBar', 'Card', 'Button', offering a unified and visually appealing design language for android and if you want IOS like UI for that Cupertino Widgets that mimic the IOS design language. Developers benefit from hot reload for rapid UI iteration and can create responsive layouts, smooth animations, and interactive gestures. Flutter's flexibility and third-party libraries facilitate the development of modern, inclusive, and dynamic user interfaces for Android apps.

### 3.2 Hardware Interfaces

To use this application in laptop or any desktop a person will need to install an emulator to run this application. Hardware requirements will be mouse and the keyboard for interacting with the application window. For phone the application will run smoothly with its animations and gestures by the help of touch screen availability and sensor working appropriately as of the optimum condition in any phone these days. The important thing to note application will only run on android version 11 and above with minimum of 2 GB ram and space required to install the application with internet connectivity intact.

### 3.3 Software Interfaces

User Registration:

- User registration form with fields for personal information (name, age, gender).
- Creation of a secure username and password.
- Option to sign up using social media or email.
- screens to introduce the app's features and benefits.

Login Interface:

- Secure login with username and password authentication
- "Forgot Password" or password recovery option.

Dashboard:

- Overview of the user's
- Quick access to important features, such as tracking, appointments, and notifications etc.

### 3.4 Communications Interfaces

Https communication protocol will be used with the internet.

## 4. System Features

This section includes use cases of our project.

### 4.1 Login/Signup

#### 4.1.1 login

<b>Use Case Name</b>	<b>Login</b>
<b>Version</b>	<b>1.0</b>
<b>Goal/Summary</b>	This use case allows the actors to login
<b>Actors</b>	Admin/owner, User ,Doctor
<b>Preconditions</b>	The actors should be registered once to login into the system.

#### Basic course of events/Happy Path (true condition)

<u>Actor Action</u>	<u>System Response</u>
1. Actors enter email and password	2. Verifies for the correct credentials from the database. 3. Checks for the account type. 4. Successful login shows home page of the app.
5. Enters the System.	

#### Alternative path (else path)

1. The actors enter wrong email.	2. Alert message will show “invalid Email or password”.
3. The actor enter wrong password	4. Alert message will show “invalid Email or password”.

<b>Post condition</b>	The actors can use the application.
<b>Author</b>	Naveen kingrani & Arslan Ali

#### 4.1.2 (b) Sign up

<b>Use Case Name</b>	<b>Signup</b>
<b>Version</b>	<b>1.0</b>
<b>Goal/Summary</b>	This use case allows the actors to register to the system.
<b>Actors</b>	Admin/owner, User ,Doctor
<b>Preconditions</b>	The actors should enter all the required details Correctly.

#### Basic course of events/Happy Path (true condition)

<b><u>Actor Action</u></b>	<b><u>System Response</u></b>
1. The actors will enter all the details.	2. System checks for the correct input of details and in correct required form.
3. The actors click on signup button.	4. The system will save the details in the database.

#### Alternative path (else path)

1. The actors leave any detail to enter in the signup form.	2. The system will not save the actors signup details into the database.
---	--

<b>Post condition</b>	The actors have registered into the application.
<b>Author</b>	Naveen kingrani & Arslan Ali

## 4.2 Doctor Appointment.

<b>Use Case Name</b>	<b>Doctor Appointment</b>
<b>Version</b>	<b>1.0</b>
<b>Goal/Summary</b>	This use case allows User to Make the appointment
<b>Actors</b>	User, Doctor.
<b>Preconditions</b>	The User must have been verified and logged into the application.

### Basic course of events/Happy Path (true condition)

<b><u>Actor Action</u></b>	<b><u>System Response</u></b>
1. User will enter the information about the Doctor specification and appoint the doctor as per there Requirements.	
2. The Actor will click the add button to appoint the Doctor Consultation	3. The system will send the information to the admin and Doctor for appointment verification.
4. The Doctor will approve the Appointment Request.	5. The system will display that Appointment on the application.

### Alternative path (else path)

1. The Actor didn't provided with the correct information of their requirement and Request is not Valid	2. Doctor will not approve the Request for the Appointment.
	3. The application will not display the Appointment on the application until it is approved.

<b>Post condition</b>	The actors can confirm appointment.
<b>Author</b>	Naveen kingrani & Arslan Ali

### 4.3 Chat box

<b>Use Case Name</b>	<b>Chat box</b>
<b>Version</b>	<b>1.0</b>
<b>Goal/Summary</b>	This use case allows the actors to communicate with each other using in app chat feature for Ease communication.
<b>Actors</b>	User, Admin/Owner, Doctor
<b>Preconditions</b>	The actors must have logged into the Application.

#### Basic course of events/Happy Path (true condition)

<u><b>Actor Action</b></u>	<u><b>System Response</b></u>
1. Actors will have to click on chat icon to communicate with others.	2. The system allows check for the profile logged in or not.
	3. If the actor is logged in then can successfully use this feature to communicate.
4. The actors communicate using chat.	

#### Alternative path (else path)

1. Actor has to log in first and verify profile.	2. Else Actor will not be allowed to use chat feature for communication.
--	--

<b>Post condition</b>	The actors can use the application.
<b>Author</b>	Naveen kingrani & Arslan Ali

#### 4.4 Add Pharmaceutical Product

<b>Use Case Name</b>	<b>Add Pharmaceutical Product</b>
<b>Version</b>	<b>1.0</b>
<b>Goal/Summary</b>	This use case allows Admin to add the product for sale.
<b>Actors</b>	Admin, User
<b>Preconditions</b>	The Admin must have been logged into the application.

##### Basic course of events/Happy Path (true condition)

<u><b>Actor Action</b></u>	<u><b>System Response</b></u>
1. Actor will buy the product according to product specification and details like images, Specification and price	
2. The Actor will click the add button to purchase the product	3. The system will send the information to the admin for product verification.
4. The Admin/owner will approve the product.	5. The system will display that product on the application.

##### Alternative path (else path)

1. The Actor doesn't clicked the add button	2. Admin will not confirm request of product for the sale.
---	--

<b>Post condition</b>	The actors can add product for purchase
<b>Author</b>	Naveen kingrani & Arslan Ali



## 4.5 Search Bar

<b>Use Case Name</b>	<b>Search Bar</b>
<b>Version</b>	<b>1.0</b>
<b>Goal/Summary</b>	This use case allows the actors to search for the product. Or Doctor
<b>Actors</b>	User, Doctor ,Admin/Owner,
<b>Preconditions</b>	Actors need to login to use this feature.

### Basic course of events/Happy Path (true condition)

<b><u>Actor Action</u></b>	<b><u>System Response</u></b>
1. Actors will search for the product or doctor by entering its name.	2. System will show the list of products and doctors with matching string / product name.

### Alternative path (else path)

1. System will enter the model no of product.	2. System will fail to respond.
---	---------------------------------

<b>Post condition</b>	The actors can find the products easily.
<b>Author</b>	Naveen kingrani & Arslan Ali
	4. Once the complaint is checked and verified the admin will block the user forever from the database.

#### 4.6 Admin (Block User / Doctor Account)

<b>Use Case Name</b>	<b>Admin (Block User / Doctor Account</b>
<b>Version</b>	<b>1.0</b>
<b>Goal/Summary</b>	This use case allows the actors to launch complaint by contacting admin for in appropriate action of user or Doctor.
<b>Actors</b>	User, Doctor , Admin/Owner,
<b>Preconditions</b>	The actors must have logged in and verified by admin.

#### Basic course of events/Happy Path (true condition)

<u><b>Actor Action</b></u>	<u><b>System Response</b></u>
1. Actors will have to report to admin by contact us page to launch complaint.	2. Admin will be notified for the complaint against the user.
	3. Admin will verify and check for the wrong activity of user.

#### Alternative path (else path)

1. Actor launches the false complaint against the user without any evidence.	2. The admin will act against the actor for launching false complaint and block the actor.
--	--

<b>Post condition</b>	The actors can use the application.
<b>Author</b>	Naveen kingrani & Arslan Ali

## 4.7 Profile management

<b>Use Case Name</b>	<b>Profile management and verification</b>
<b>Version</b>	<b>1.0</b>
<b>Goal/Summary</b>	This use case allows the actors to manage and verify their profile by application admin.
<b>Actors</b>	User, Doctor , Admin/Owner,
<b>Preconditions</b>	The actors should be registered with the correct information in order to profile verification request.

### Basic course of events/Happy Path (true condition)

<u><b>Actor Action</b></u>	<u><b>System Response</b></u>
1. Actors will have to click verify profile button in profile management section.	2. Admin will be notified for the profile verification.
	3. Admin will verify and check for the profile details which actor entered during registration.
	4. The admin will approve once all the details checked and mark the profile verified.

### Alternative path (else path)

1. Actor doesn't clicked profile verification button.	2. The actor is not allowed to fully use the application features by admin.
---	---

<b>Post condition</b>	The actors can use the application.
<b>Author</b>	Naveen kingrani & Arslan Ali

## 4.8 Health article

<b>Use Case Name</b>	<b>Health Article</b>
<b>Version</b>	<b>1.0</b>
<b>Goal/Summary</b>	This use case allows the actors to view the health articles newest which modified by admin
<b>Actors</b>	Admin, user, Doctor
<b>Preconditions</b>	The actors should login with verified information to use this feature.

### Basic course of events/Happy Path (true condition)

<u><b>Actor Action</b></u>	<u><b>System Response</b></u>
1. Actors will have to click on health article field to select and proceed to view the articles	2. The system will check for actors to be logged in and verified to use this feature.
	3. Then, system proceed the function and show the health article.

### Alternative path (else path)

3. Actor clicked on article field button.	4. The system application showed a message pop "Please Login with verified credentials".
---	--

<b>Post condition</b>	The actors can use the application.
<b>Author</b>	Naveen kingrani & Arslan Ali

## 4.9 Appointment reminder

<b>Use Case Name</b>	<b>Appointment Reminder</b>
<b>Version</b>	<b>1.0</b>
<b>Goal/Summary</b>	This use case allows the actors to view their history of Appointment
<b>Actors</b>	User, Doctor, Admin/Owner.
<b>Preconditions</b>	The actors must have logged in and verified to Use this feature.

### Basic course of events/Happy Path (true condition)

<u><b>Actor Action</b></u>	<u><b>System Response</b></u>
1. Actors can check their appointment log by tapping on appointment section in the features menu.	2. The system checks actor to be logged in and verified.
	3. Once account is logged in the system show all the appointment which happened or happening currently. And also notify about upcoming appointments
4. The actor can successfully view appointment log.	

### Alternative path (else path)

1. Actor has not taken appointment before.	2. The system will show empty appointment log and prompt message will show “all appointments will show here”.
--	---

<b>Post condition</b>	The actors can use the application.
<b>Author</b>	Naveen kingrani & Arslan Ali

## 4.10 S.O.S feature

<b>Use Case Name</b>	<b>SOS</b>
<b>Version</b>	<b>1.0</b>
<b>Goal/Summary</b>	This use case allows the actors to request for emergency service with one tap
<b>Actors</b>	User, Doctor
<b>Preconditions</b>	The actors should login first to use the feature.

### Basic course of events/Happy Path (true condition)

<b><u>Actor Action</u></b>	<b><u>System Response</u></b>
1. Actors will have to click on S.O.S feature button	2. The system will check actor logged in or not.
	3. If the actors are logged in successfully then the feature work as the automatically dispatch the first responders to the actor location by notifying the emergency service providers.

### Alternative path (else path)

1. Actor clicked to S.O.S feature	2. The system application shows alert message to “log in first to use this feature”.
-----------------------------------	--

<b>Post condition</b>	The actors can use the application.
<b>Author</b>	Naveen kingrani & Arslan Ali

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

The system will be designed for smoother user experience as per the user needs. For that, we are using latest versions of android studio IDE with updated android versions supported version 11 and above.

## 5.2 Safety Requirements

The application will be designed in a way that no such harm is done at the user experience. One of the major concerns is that the products should be verified for that we have taken multiple steps of verification under admin entity to ensure safety and good usage of This app. However, in the application we have included features like block user, verify user, As long as these precautions are followed, the system should not cause significant harm.

## 5.3 Security Requirements

To maintain security, whatever data is filled out by the user or Doctor will be sent to the admin for verification, whether the information is reliable or not, and then further action will be performed. For example, the available Doctor will be checked by the admin before the appointment to user.

## 5.4 Software Quality Attributes

Some important qualities of our system that are beneficial for user, Doctors and administrators include:

- **Adaptability:** anyone can easily understand and utilize this application.
- **Correctness:** the application will consistently perform calculations accurately, regardless of the number of instructions involved.
- **Maintainability:** The application can be easily maintained by admin.
- **Portability:** The application will be available as android application which is easy to download and install on any android phone.

## 5.5 Business Rules

The addition of new functions and any changes made to the system will have to be done through admin rights.

## 6. Other Requirements

### Appendix A: Glossary

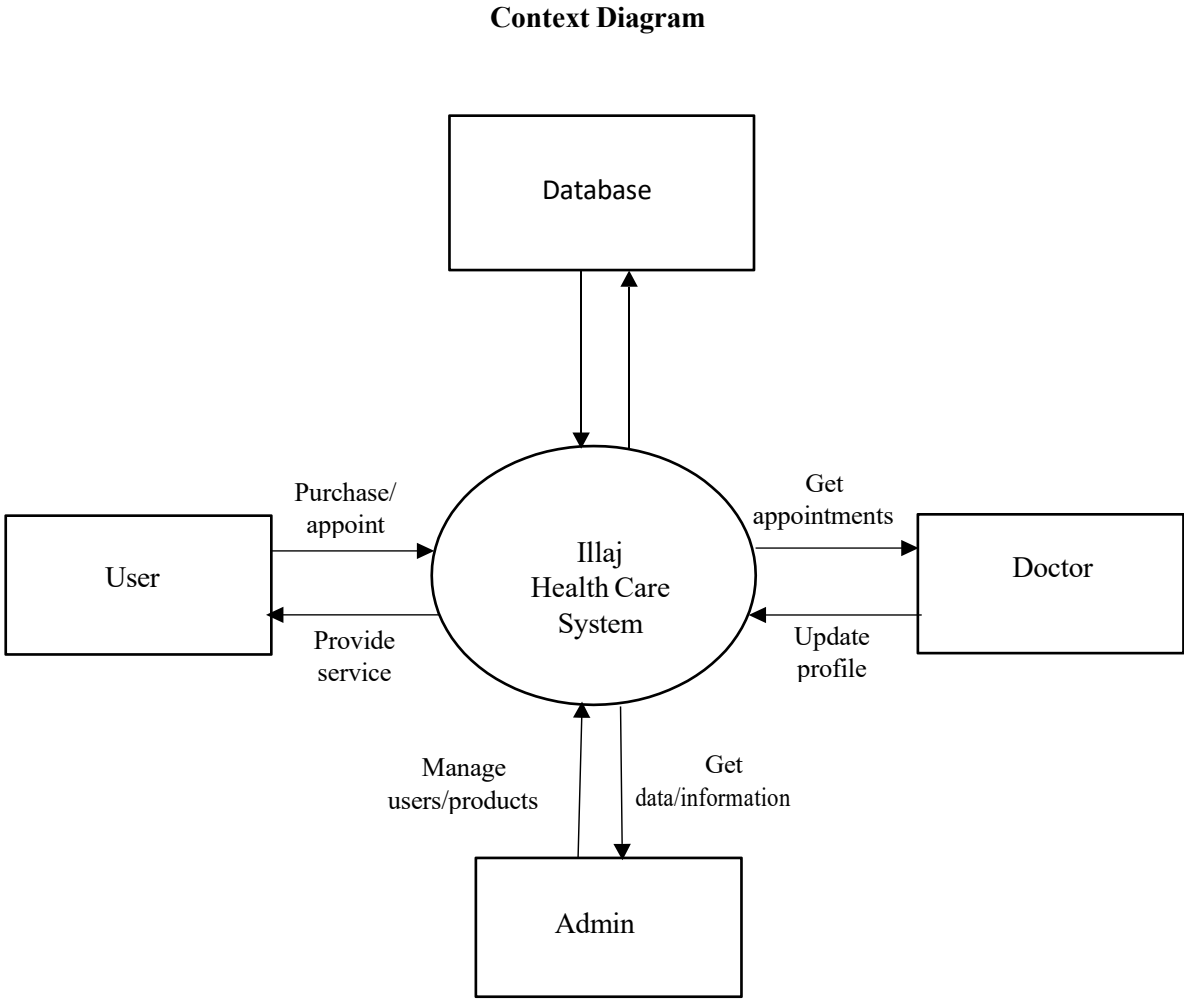
- SRS = Software Requirement Specification.
- SDS = Software Design Specification.
- STD = Software Testing Document.
- SDK = Software Development Kit.
- GUI = Graphical User interface.
- API = Application programming interface.
- HTTPS = Hypertext Transfer Protocol Secure.
- DB = Database.
- Illaj = the name of the application system.

# Appendix B: Analysis Models

N/A

# Appendix C: To Be Determined List

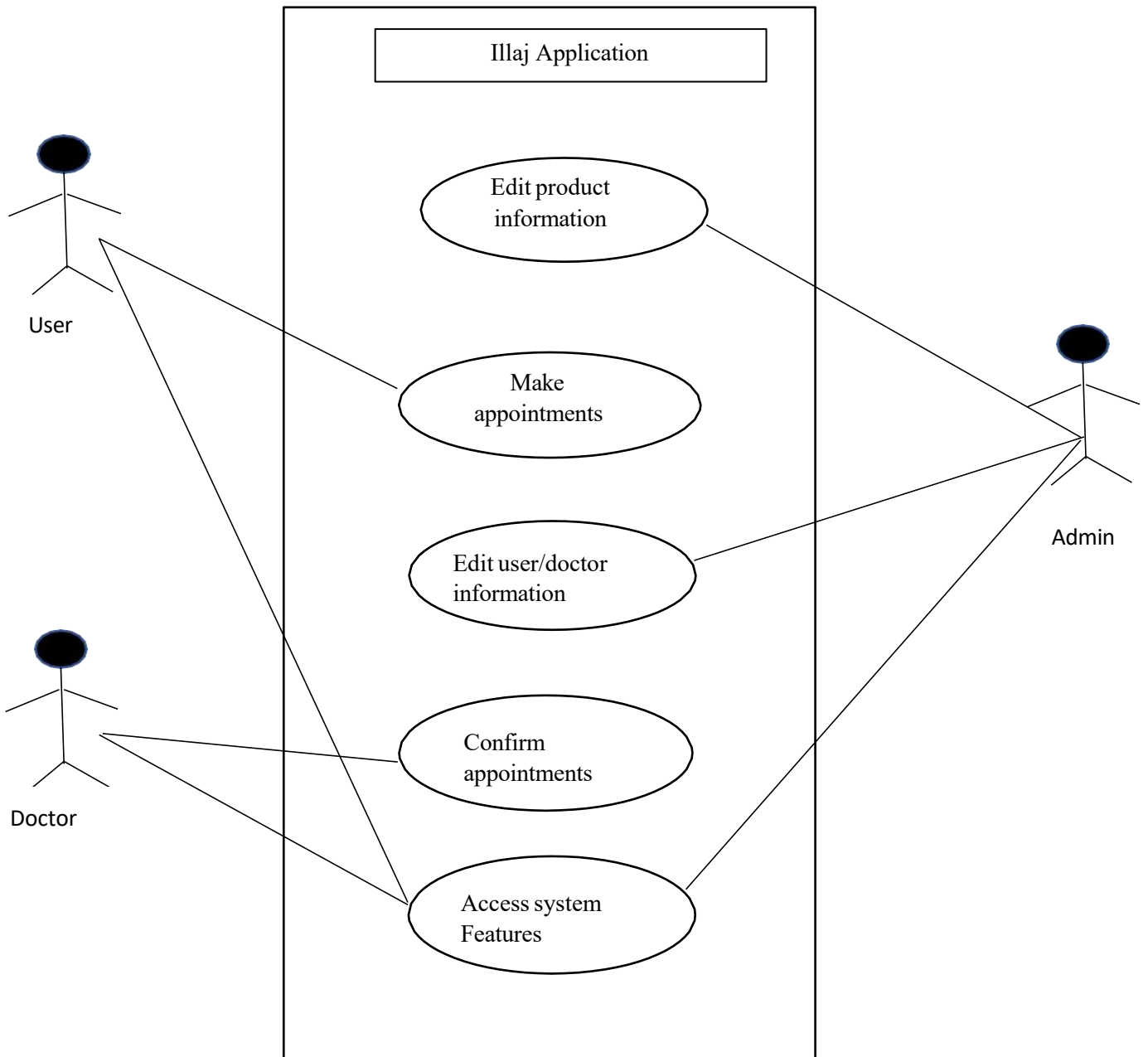
N/A



*Figure Context Diagram*



## Use Case Diagram



*Figure Use case Diagram*

## Entity Relationship Diagram (ERD)

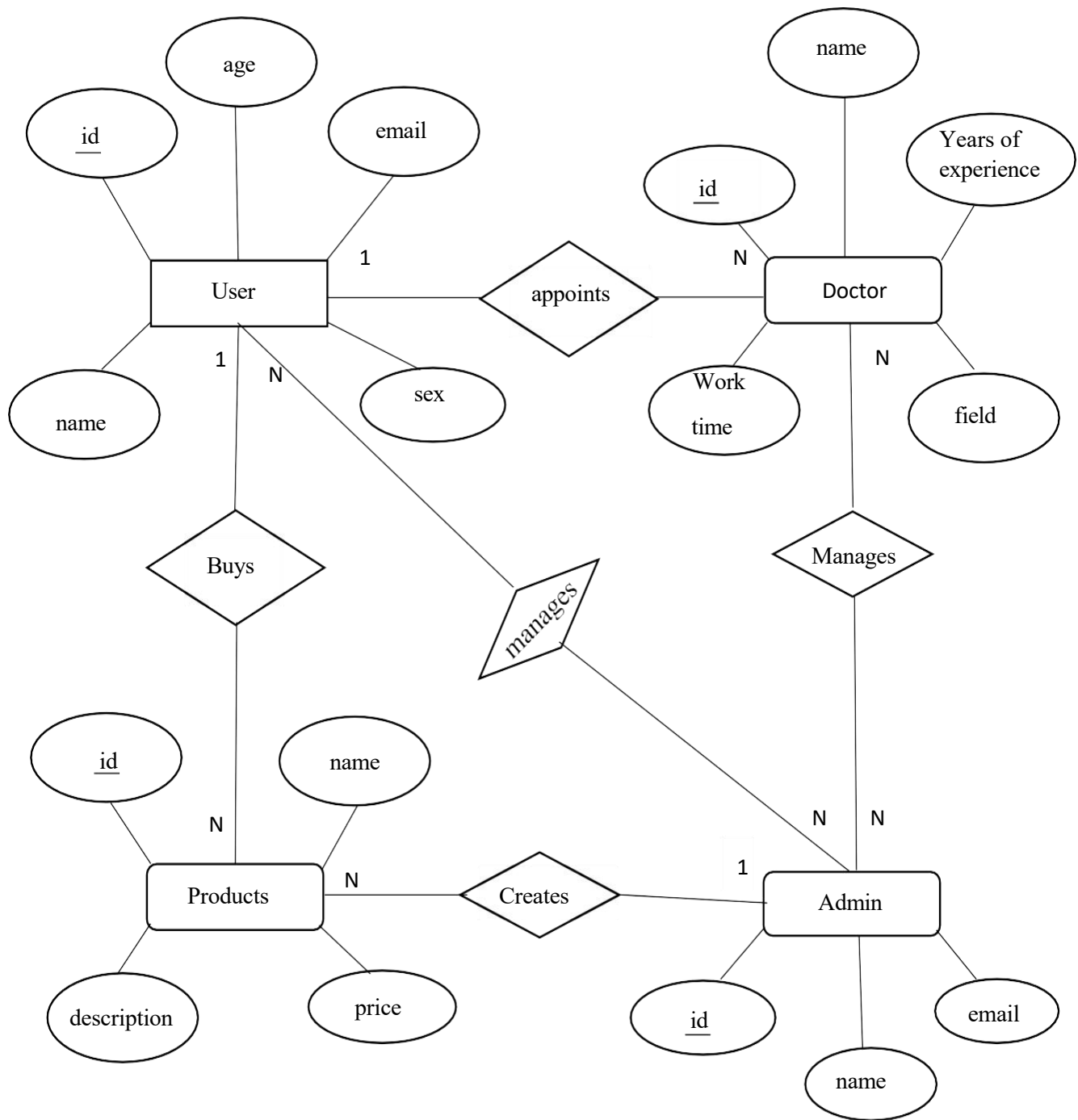
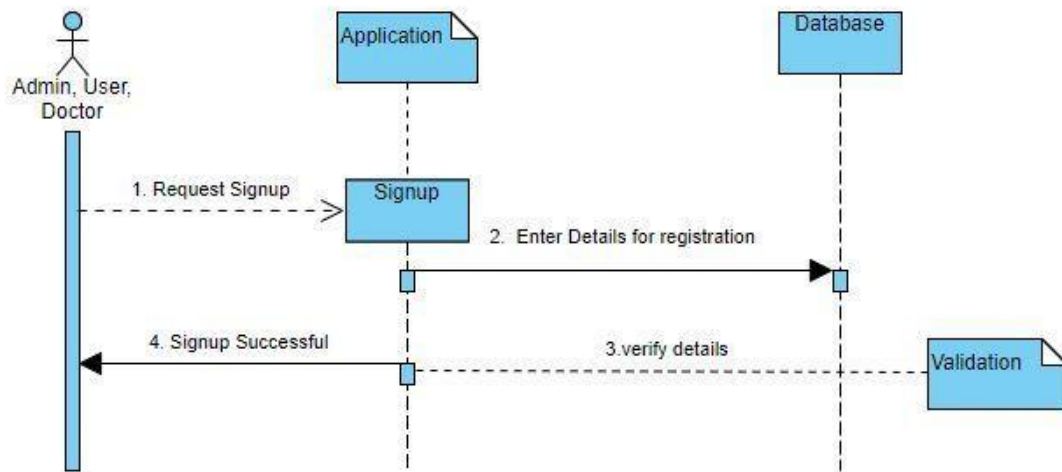


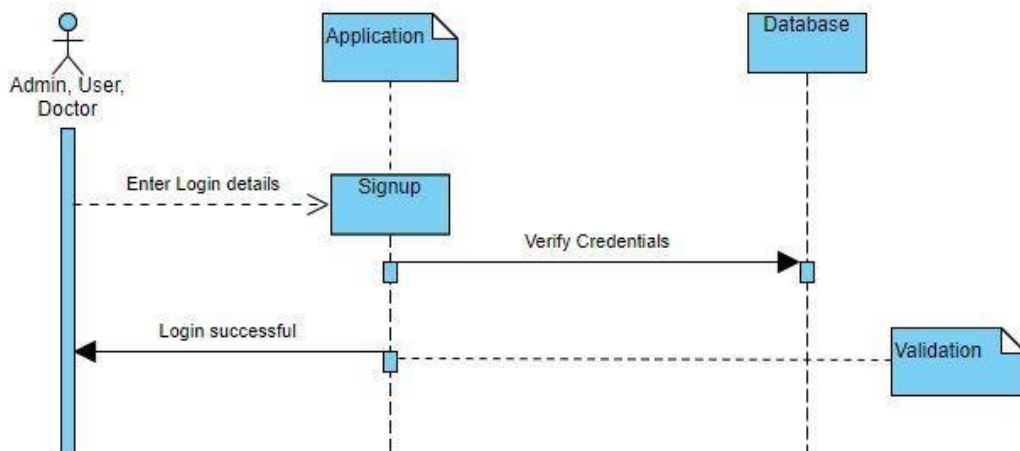
Figure ER Diagram

## Sequence diagram

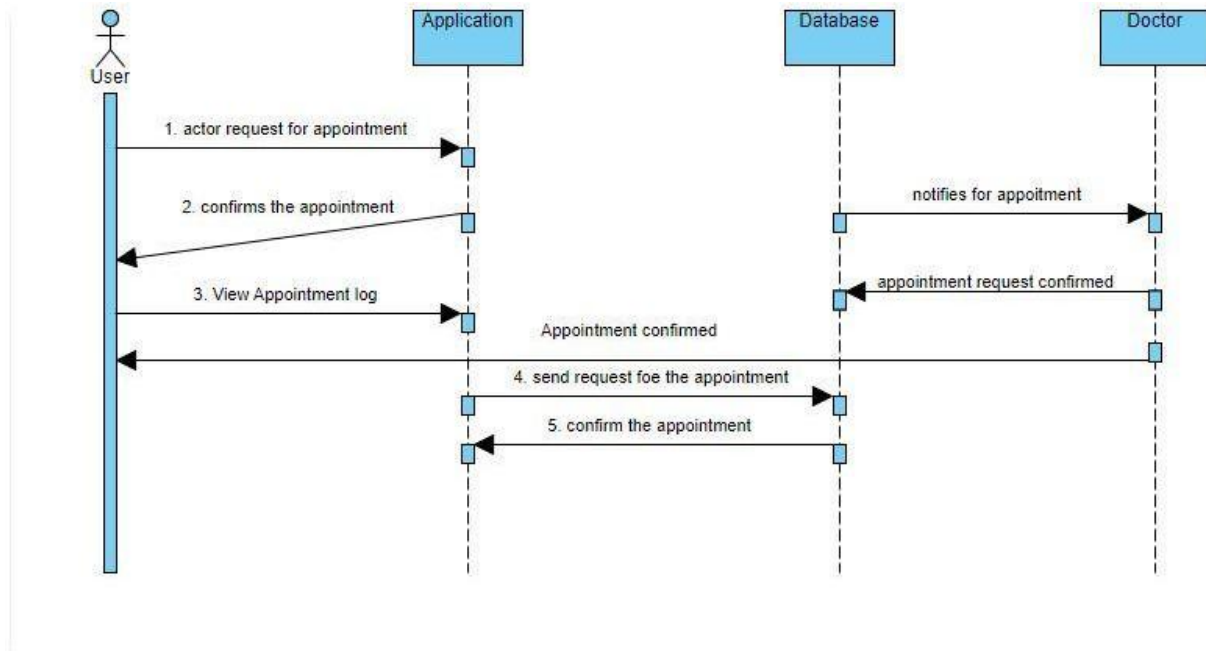
### 1. Signup:



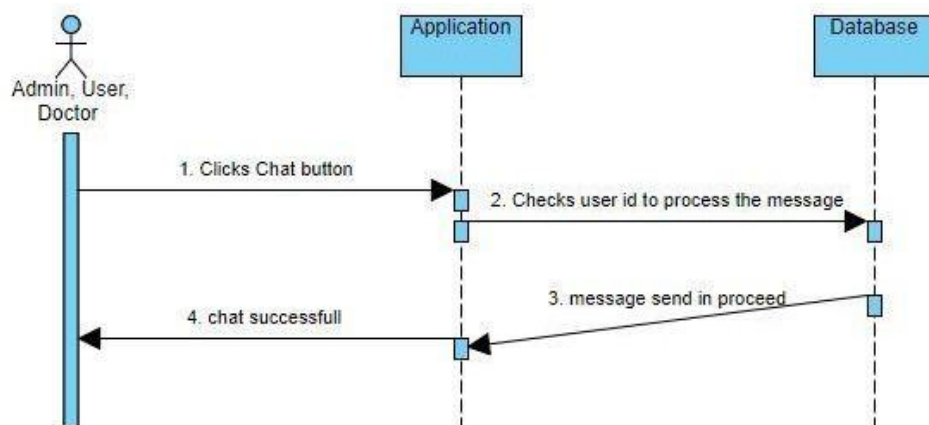
### 2. Login:



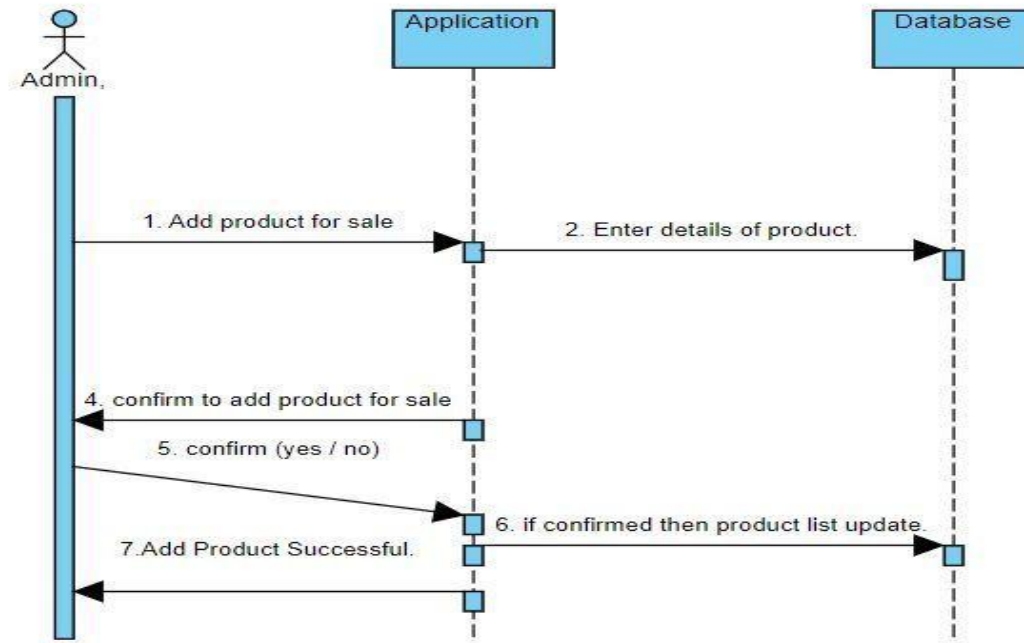
### 3. Doctor Appointment:



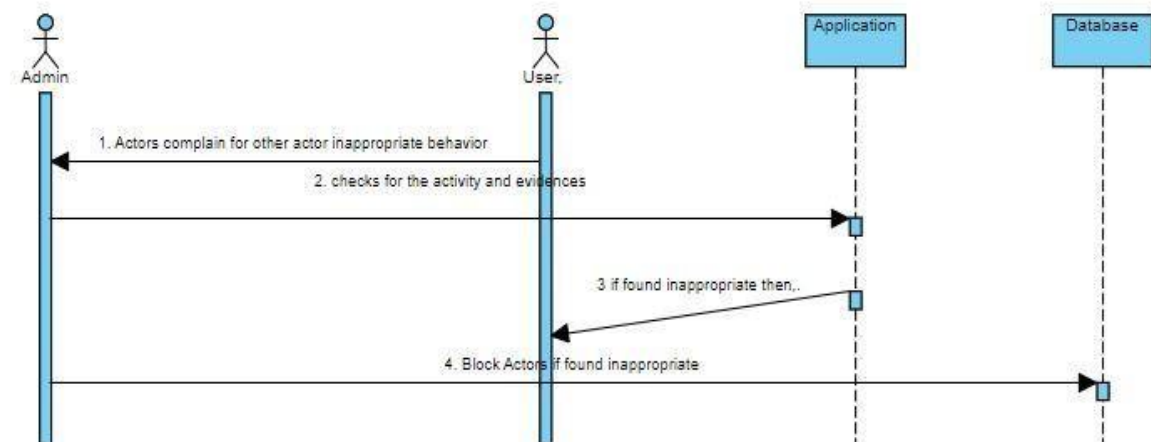
### 4. Chat box



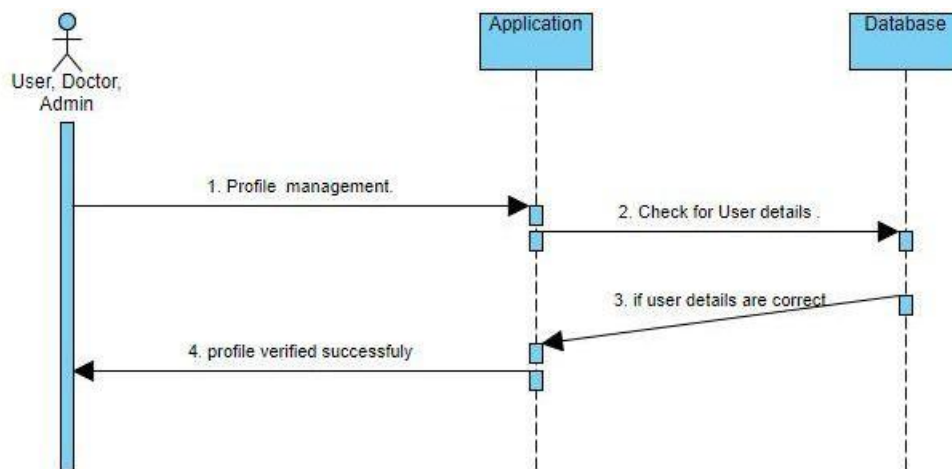
#### 4. Add Pharmaceutical Product :



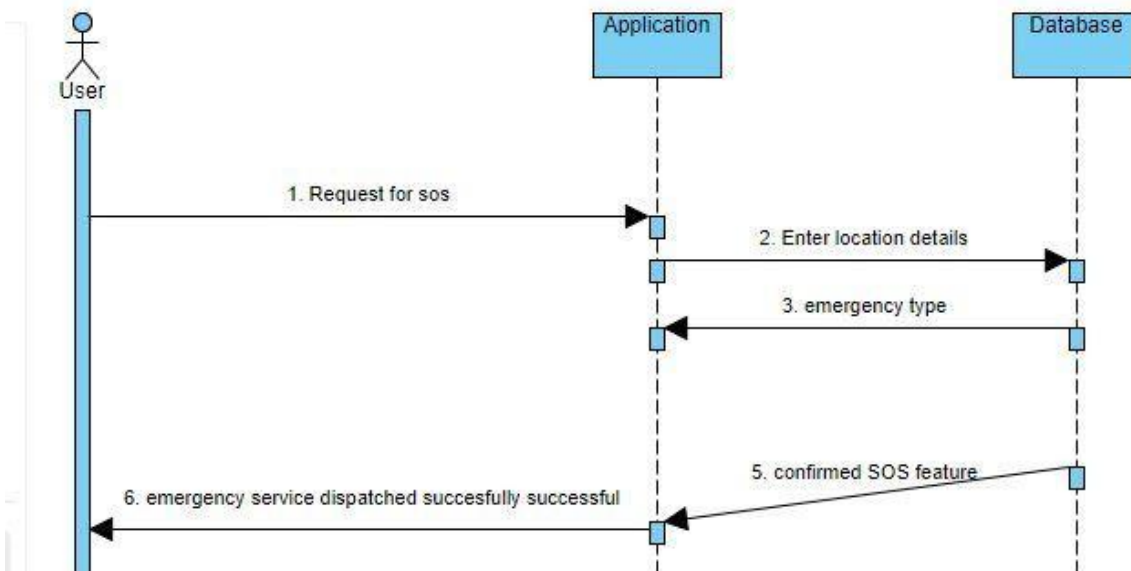
#### 6. Admin (Block User / Doctor Account)



## 6. Profile management :



## 7. S.O.S feature :





# Software Design Specification

## 1. Introduction

### 1.1 Purpose of the Document

The purpose of this document is to serve for the development of a smartphone app aimed at addressing the challenge of managing mobile health care system in today's busy and stressful world. It emphasizes the need for ongoing updates and improvements to ensure the platform remains competitive and aligned with evolving industry trends. Overall, the document serves as a blueprint for the development team, guiding the creation of a user-friendly and effective app to promote E-health well-being.

### 1.2 Scope of the Development Project

The project scope involves the development of a two-phase Smartphone app for mobile health care system. FYP 1 focuses on features like Doctor Appointment and patient consultation, while FYP 2 introduces Pharmacy service, notifications, SOS feature, and admin panel. The app aims to cater to users, healthcare professionals, and administrators. The primary goal is to provide a comprehensive, user-friendly tool for managing health application well-being in a convenient and accessible manner.

### 1.3 Definitions, Acronyms and Abbreviations

- Terms and Description
- ERD            Entity Relationship
- SDS            Software Design Specification
- SRS            Software Requirements Specification

### 1.4 References

- ✓ <https://developers.google.com/learn/pathways/intro-to-flutter>
- ✓ <https://blog.logrocket.com/introduction-to-using-dart-in-flutter/>
- ✓ <https://www.pinterest.org/newsroom/articles/mental-health-awareness-blog/>
- ✓ <https://www.healthline.com/health/mental-health/mental-health-apps>

### 1.5 Overview of the Document

The next sections will describe the architectural design of the project, the high level components and their interactions. Some issues will be discussed which might be faced in user interface. The system architecture description section describes the overview of the components, structure and relationships among the components along the way it also identifies the user interface issues. On the other hand, section 3 gives in depth description of components, section 4 gives overall description of the interface of the system, section 5 identifies the relations with other products, section 6 describes the design trade off, section 7 describes the pseudocodes for components.



## 2. System Architecture Description

This section provides an overview and rationale for the program's data and architectural design decisions.

### 2.1 Section Overview

This section delves into crucial factors shaping the design of the mental health app, encompassing overarching constraints, data architecture, program structure, and the chosen architectural model. It outlines constraints affecting app development, elucidates database structures and internal data organization, and introduces the selected architectural model, along with considerations for alternative approaches.

### 2.2 General Constraints

- ✓ **Time constraints:** The project must adhere to a set timeline, necessitating efficient development and testing.
- ✓ **Resource constraints:** Availability of limited resources, including manpower and technology, may impact project execution.
- ✓ **Compatibility constraints:** Ensuring the app works seamlessly across various Android devices poses a challenge.
- ✓ **Security constraints:** Robust measures are essential to safeguard user data and maintain the app's integrity.

Tools:

- a. Visual Studio Code
- b. MongoDB
- c. Android studio

### 2.3 Data Design

Data design involves structuring databases for the health app, specifying how information is organized and accessed. It encompasses defining data entities, relationships, and attributes crucial for functionalities like doctor appointment and appointment scheduling. The design aims to ensure efficient data storage, retrieval, and management within the app's framework. Considerations include data security, integrity, and scalability to support evolving user needs.

## **2.4 Program Structure**

The program structure defines the organization and arrangement of components within the health app. It outlines how modules and functions interact to achieve the app's objectives. This structure ensures efficient code organization, enhancing maintainability and scalability. The program structure is pivotal for a well-coordinated and effective implementation of mobile application health features.

## **2.5 Alternatives Considered:**

Various alternatives were evaluated during the app design, considering factors like user interface frameworks, database systems, and architectural models. The selection process involved weighing pros and cons to determine the most suitable choices. Alternatives were explored to ensure optimal performance, scalability, and compatibility with project requirements. The chosen alternatives align with the goal of creating a robust and user-friendly mental health app

# **3. Detailed description of components**

## **3.1 Section overview**

In this section we describe the details of components in the system, it will give also give the overall view of all the functional components. It will describe each component's identification, type, purpose, function subordinates, Dependencies, interfaces, resources, processing and data.

### 3.1.1 Login

<b>Identification</b>	Login
<b>Type</b>	Form, method
<b>Purpose</b>	To authenticate and authorize users, providing access to personalized mental health features and ensuring secure interaction within the app.
<b>Function</b>	Validate user credentials (username and password).
<b>Dependencies</b>	Relies on a secure database for storing and retrieving user credentials. Requires network connectivity for user authentication.
<b>Interfaces</b>	User interface for inputting login credentials. Integration with a secure backend system for authentication.
<b>Resources</b>	Software required: Flutter, Dart, mongodb. Operating system: iOS and android
<b>Processing</b>	Verify entered credentials against stored records. Generate and manage authentication tokens.
<b>Data</b>	User credentials (username, password). Authentication tokens for active user sessions.

### 3.1.2 Sign up

<b>Identification</b>	Signup
<b>Type</b>	Form, method
<b>Purpose</b>	To enable users to create accounts and access the app's features, fostering personalized mental health management.
<b>Function</b>	Facilitates user registration by collecting and verifying personal information, creating unique user accounts, and ensuring secure access to the app.
<b>Dependencies</b>	Relies on a secure and stable internet connection.
<b>Interfaces</b>	User through the app's registration forms and interacts with the database for user profile creation.
<b>Resources</b>	Software required: Flutter, Dart, mongodb. Operating system: iOS and android
<b>Processing</b>	Executes user authentication, data validation, and account creation processes during the registration workflow.
<b>Data</b>	captures and stores user-provided data, including personal information required for account creation and app access.

### 3.1.3 Dashboard

<b>Identification</b>	Dashboard
<b>Type</b>	Class/Page/Method
<b>Purpose</b>	to serve as the central hub for users, offering a consolidated view of key app features and information.
<b>Function</b>	Aggregates and displays relevant data such as Doctor appointment, health article, for user engagement and drawer for all specified features of the application.
<b>Dependencies</b>	Relies on data from appointment, and input from user
<b>Interfaces</b>	The user through the app's main dashboard and interacts with the app's notification system for reminders.
<b>Resources</b>	Software required: Flutter, Dart, MongoDB. Operating system: iOS and android
<b>Processing</b>	Manages the arrangement and presentation of data, ensuring a user-friendly and responsive dashboard interface.
<b>Data</b>	Retrieves and displays user mood data, meditation session history, and wellness content dynamically.

### 3.1.4 Doctor Profile:

<b>Identification</b>	Doctor Profile
<b>Type</b>	Form, method
<b>Purpose</b>	The purpose of the Doctor Profile module is to enable healthcare professionals to create and manage their profiles within the health app, facilitating seamless interaction with patients.
<b>Function</b>	This module functions to allow doctors to input and update their professional information, including specialization, contact details, and availability. It ensures accurate representation of healthcare providers within the app.
<b>Dependencies</b>	The Doctor Profile module relies on user authentication and database connectivity to store and retrieve doctor-specific information. It may also depend on scheduling and appointment modules for coordination.
<b>Interfaces</b>	The module interfaces with the app's user interface for doctors, providing a platform for profile creation and updates.
<b>Resources</b>	Software required: Flutter, Dart, mongodb. Operating system: iOS and android
<b>Processing</b>	The processing aspect involves handling requests from doctors for profile creation, updates, and ensuring the secure storage of their professional information. It also includes validation processes to maintain data accuracy.
<b>Data</b>	The data associated with the Doctor Profile module includes doctor-specific information such as name, specialization, contact details, and availability.

### 3.1.5 Doctor appointment:

<b>Identification</b>	Appointments
<b>Type</b>	Form, method
<b>Purpose</b>	Facilitate the scheduling and management of appointments between users and healthcare professionals within the app.
<b>Function</b>	Allow users to request, view, and cancel appointments; enable healthcare professionals to accept or reject appointments.
<b>Dependencies</b>	Relies on user profiles, availability data, and notification services to ensure effective appointment coordination.
<b>Interfaces</b>	Interfaces with the user profile module for patient information, the calendar module for scheduling, and the notification system for appointment reminders.
<b>Resources</b>	Software required: Flutter, Dart, mongodb. Operating system: iOS and android
<b>Processing</b>	Involves processing user input for appointment requests, updating calendar schedules, and sending notifications for appointment-related events.
<b>Data</b>	Stores data such as appointment date, time, user and professional details, and status of appointments.

### 3.1.6 Chat box

<b>Identification</b>	Real-time communication
<b>Type</b>	Form, method
<b>Purpose</b>	The chat box serves as a real-time communication tool within the health care platform, facilitating direct interaction between users and Doctors to coordinate efficiently.
<b>Function</b>	Its primary function is to enable users and Doctors to exchange messages, discuss details, and coordinate case, enhancing the overall communication experience.
<b>Dependencies</b>	The chat box depends on stable internet connectivity for real-time message delivery and relies on the backend server infrastructure to handle message processing and storage.
<b>Interfaces</b>	The chat box interfaces with the user interface of the illaj app, allowing users and doctors to send and receive messages seamlessly. It also interfaces with the backend server for message processing.
<b>Resources</b>	Software required: Flutter, Dart, Mongodb. Operating system: iOS and android
<b>Processing</b>	The chat box processes incoming and outgoing messages, ensuring timely and accurate communication between users and Doctors. It also handles notifications for new messages.
<b>Data</b>	The chat box manages user-generated data, including message content, timestamps, and user identifiers. It relies on the data storage infrastructure to maintain a history of messages for reference.



### 3.1.7 Health article:

<b>Identification</b>	Health Article
<b>Type</b>	Form, method
<b>Purpose</b>	The purpose of the health article module is to allow users and doctors to share latest health updates through health articles.
<b>Function</b>	The function of the health article module is to provide a user-friendly platform for and saving articles entries. It enables users and doctors to input and organize their thoughts chronologically.
<b>Dependencies</b>	The health article module may depend on user authentication services to ensure secure access. It could also rely on storage services for saving and retrieving Articles entries.
<b>Interfaces</b>	The user through the app's main dashboard and interacts with the app's notification system for reminders.
<b>Resources</b>	Software required: Flutter, Dart, Mongodb. Operating system: iOS and android
<b>Processing</b>	The module processes user inputs, such as text entries and optional media attachments, ensuring seamless integration with the overall app functionality. It handles data storage and retrieval processes efficiently.
<b>Data</b>	Data managed by the article module includes user-generated text entries, timestamps, and optional multimedia content.

### 3.1.8 Profile management:

<b>Identification</b>	Profile Management
<b>Type</b>	Form, method
<b>Purpose</b>	The purpose of the profile management module is to allow users and doctors to update their profile.
<b>Function</b>	The function of the module profile management is to provide a user-friendly platform for and saving profile entries. It enables users and doctors to input and organize their profile chronologically.
<b>Dependencies</b>	The profile management module may depend on user authentication services to ensure secure access. It could also rely on storage services for saving and retrieving entries.
<b>Interfaces</b>	The user through the app's main dashboard and interacts with the app's notification system.
<b>Resources</b>	Software required: Flutter, Dart, mongodb. Operating system: iOS and android
<b>Processing</b>	The module processes user inputs, such as text entries, ensuring seamless integration with the overall app functionality. It handles data storage and retrieval processes efficiently.
<b>Data</b>	Data managed by the profile management module includes user-generated text entries.

## 4. User interface design

### 4.1 Section overview

This section delineates the development standards employed in crafting a business growth application with a user-friendly interface. These standards directed the design process to achieve a harmonious blend of visual appeal, simplicity, and user comprehension. The goal was to create a balanced design that seamlessly integrates aesthetic elements with intuitive navigation, fostering an enhanced and user-friendly experience.

### 4.2 Interface Design Rules

The design rules used in our interface are as follows:

- The interface is flexible
- Relays on recognition rather than recall.
- Contains buttons that are easy to use.
- Labels are of reasonable size.
- The interface is user friendly, it allows convenient usability even for non-traditional mobile users.
- Displays descriptive texts and messages where necessary to help user identify any requirements and or errors.
- It allows user to directly manipulate interface objects.
- It provides visual cues.
- Use of real world metaphor.

### **4.3 GUI Components:**

All the above GUI components collectively contribute to a user-friendly and engaging interface for the "Mental health & Meditation" platform, enhancing the overall stress relieving and meditation experience.

### **4.4 Detailed description**

The project's GUI is designed for a seamless user experience, featuring intuitive navigation through components like appointment feature, health Article, and pharmacy. A user-friendly interface ensures easy access to health article feature, Reminder, and healthcare professional appointments. The incorporation of profiles for users, doctors, and admin enhances the platform's versatility and usability.

### **5. Reuse and relationship to other products**

The app leverages reusable code modules for features like user authentication and database interactions, optimizing development efficiency. The platform integrates seamlessly with external databases and APIs, fostering connections with healthcare services. Utilizing Flutter and Dart ensures cross-platform compatibility, allowing for a consistent user experience across various devices. Ongoing updates and improvements align the platform with evolving industry trends, ensuring relevance and competitiveness in the health and wellness sector.

### **6. Design decisions and tradeoffs**

Design decisions for this project involve selecting Flutter and Dart for cross-platform development, ensuring wider accessibility. Tradeoffs include addressing potential challenges in thorough app testing and ensuring compatibility across diverse Android devices, balancing comprehensive testing with resource constraints. The chosen architecture prioritizes user-friendly interfaces, yet meticulous consideration is required to balance simplicity with robust functionality. Alternative considerations are explored, aiming to optimize performance, security, and user experience in tandem.

## 7. Pseudo code for Components

### 7.1 Login

Begin

Enter valid email = True Enter valid password: = True

Print (“user log in successful:”) If email already exist

Then

Print (“user already exit”) While (logged\_in:= false) do Begin

Print (“Please enter your valid email”) Read(inputted email)

Write (“Please enter your password”) Read (inputted password)

If inputted email is in log in dictionary, then

//match inputted login details with each existing valid set of log in details from log in dictionary

Valid input log in email:=True And password are both matched Logged\_in:= true

Valid input password: =false email is not matched

Write(“Enter your valid email and password”) If logged\_in: =true, then

Write(“Log in successful”) End if

End

### 7.2 Sign Up

Begin Enter name Enter email

Enter password Enter

If all details are correct print “Sign up successful” Else if details incorrect

Print “enter the details again” Account created

### 7.3 Doctor appointment

function scheduleAppointment(doctor, date, time):

if appointmentAvailable(doctor, date, time):

bookAppointment(doctor, date, time)

else:

displayError("Appointment not available.")

### 7.4 Dashboard

function addWidget(widget):

if isValidWidget(widget):

widgets.add(widget)

function removeWidget(widget):

if widgetExists(widget):

widgets.remove(widget)

function rearrangeWidgets(newOrder):

if isValidOrder(newOrder):

widgets = newOrder

### **7.5 Profile modification**

```
modifyProfile(newUsername, newEmail, newPassword):  
if isValidUsername(newUsername):  
    setUsername(newUsername)  
if isValidEmail(newEmail):  
    setEmail(newEmail)  
if isValidPassword(newPassword):  
    setPassword(newPassword)
```

### **7.6 Health article**

```
Function display health articles():  
Show Health article()
```

### **7.7 Search box**

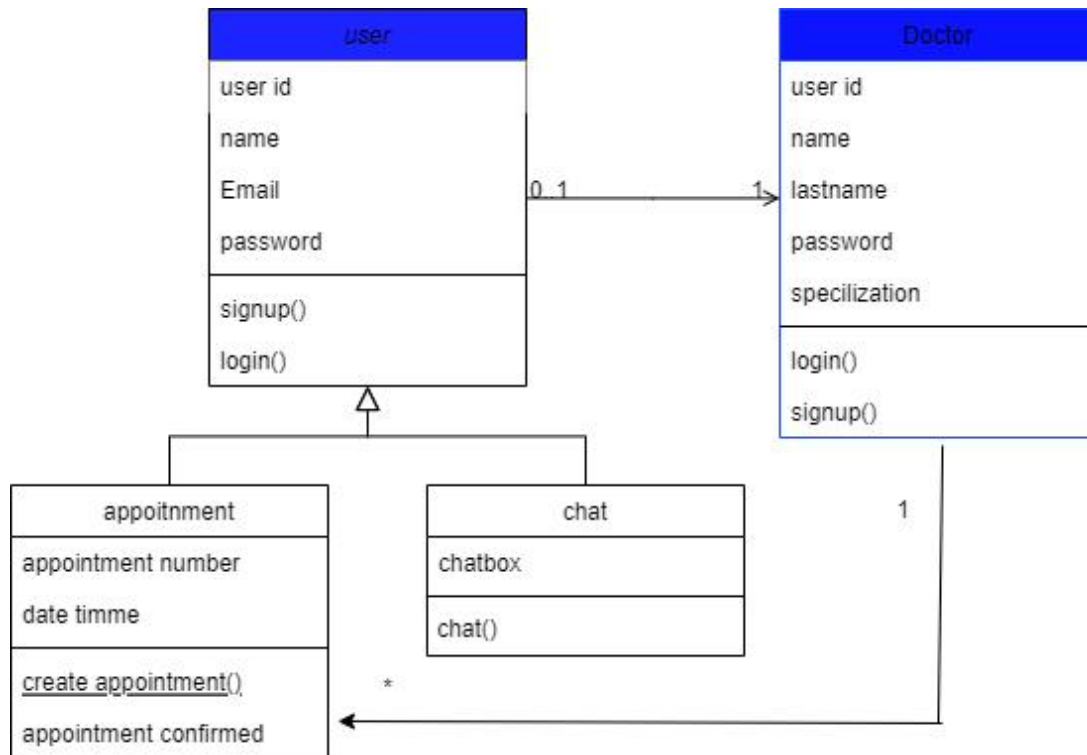
```
Function display Search Bar():  
Show Searched item()
```

### **7.8 Chat box**

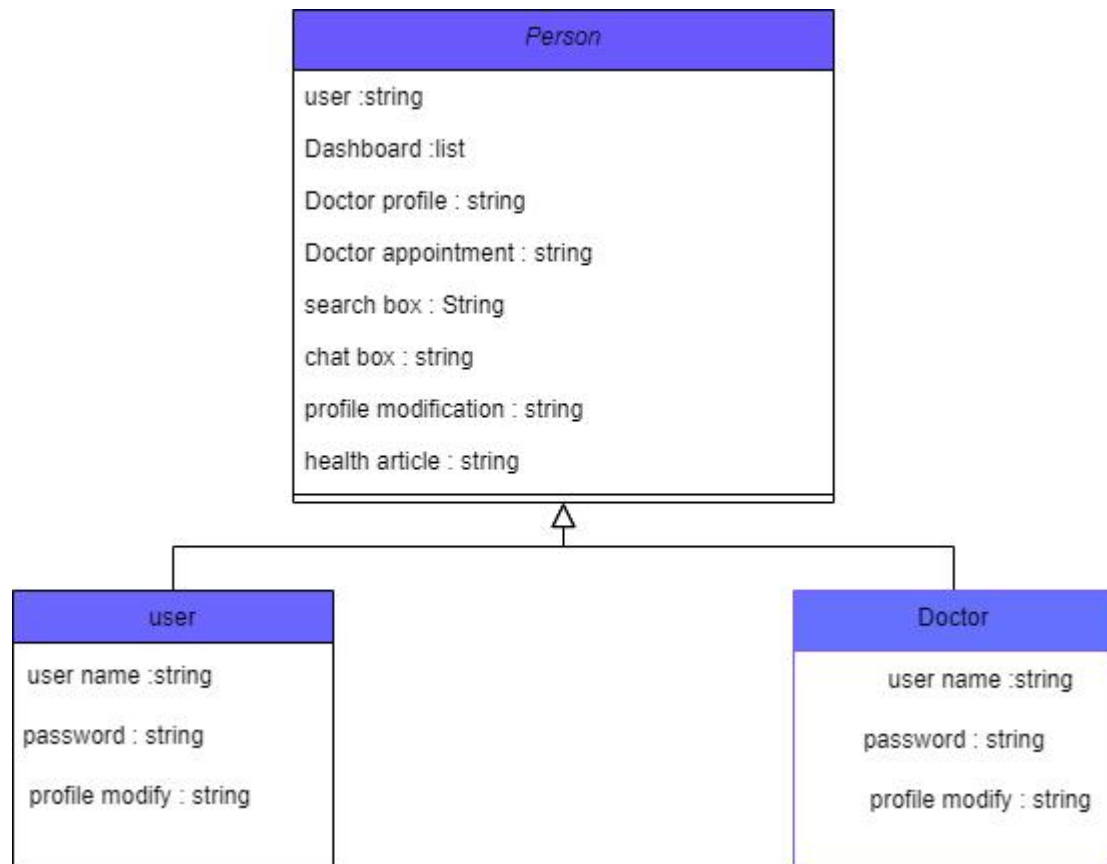
```
Initialize an empty list to store chat messages.  
Function to add a message to the chat.  
Append the new message to the chat_messages list.  
Function to display all messages in the chat.  
Iterate through each message in chat_messages.  
Print the sender and content of each message.
```

## 8. Appendices

### 8.1 Class diagram

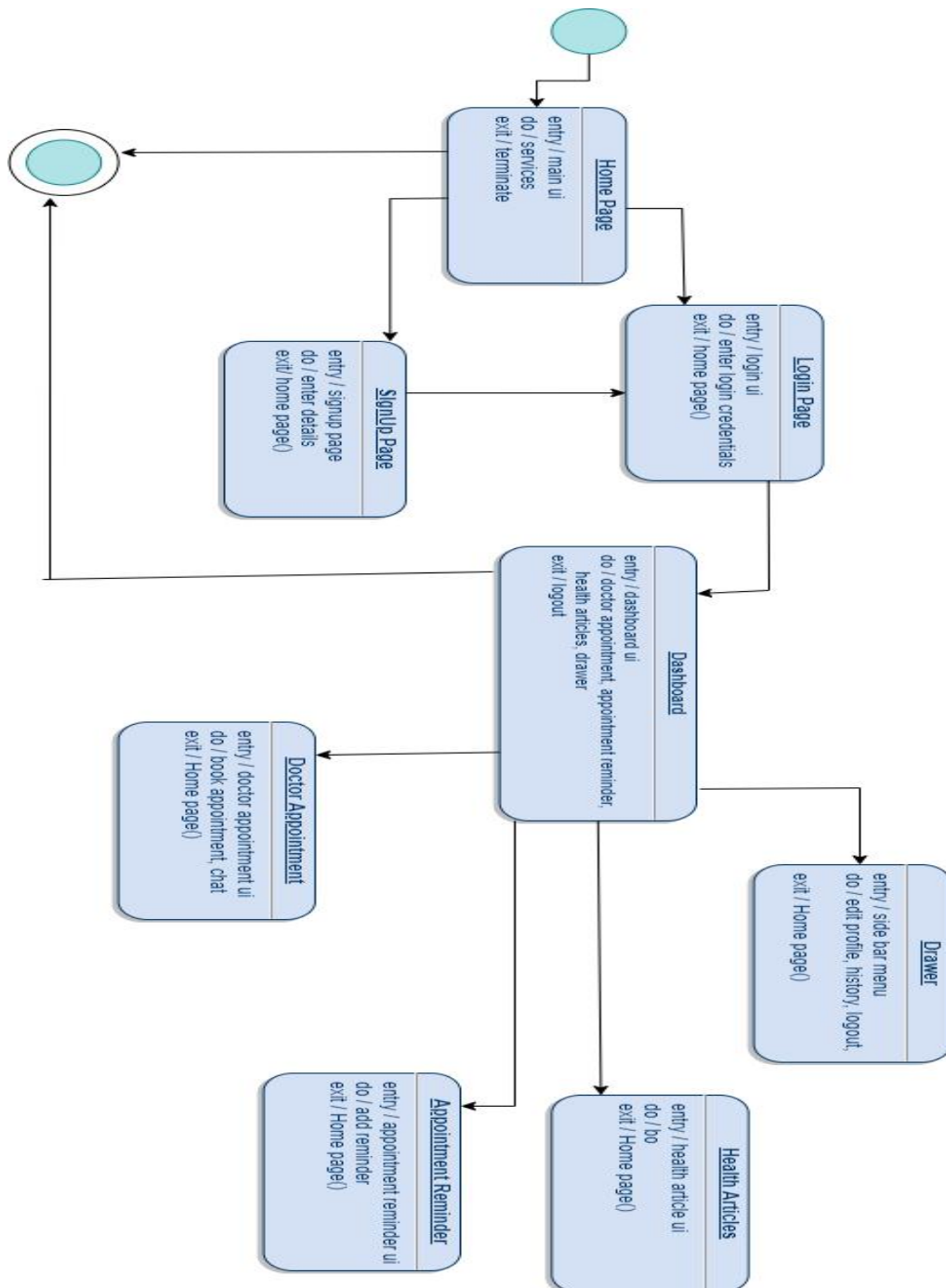


## 8.2 Object Diagram



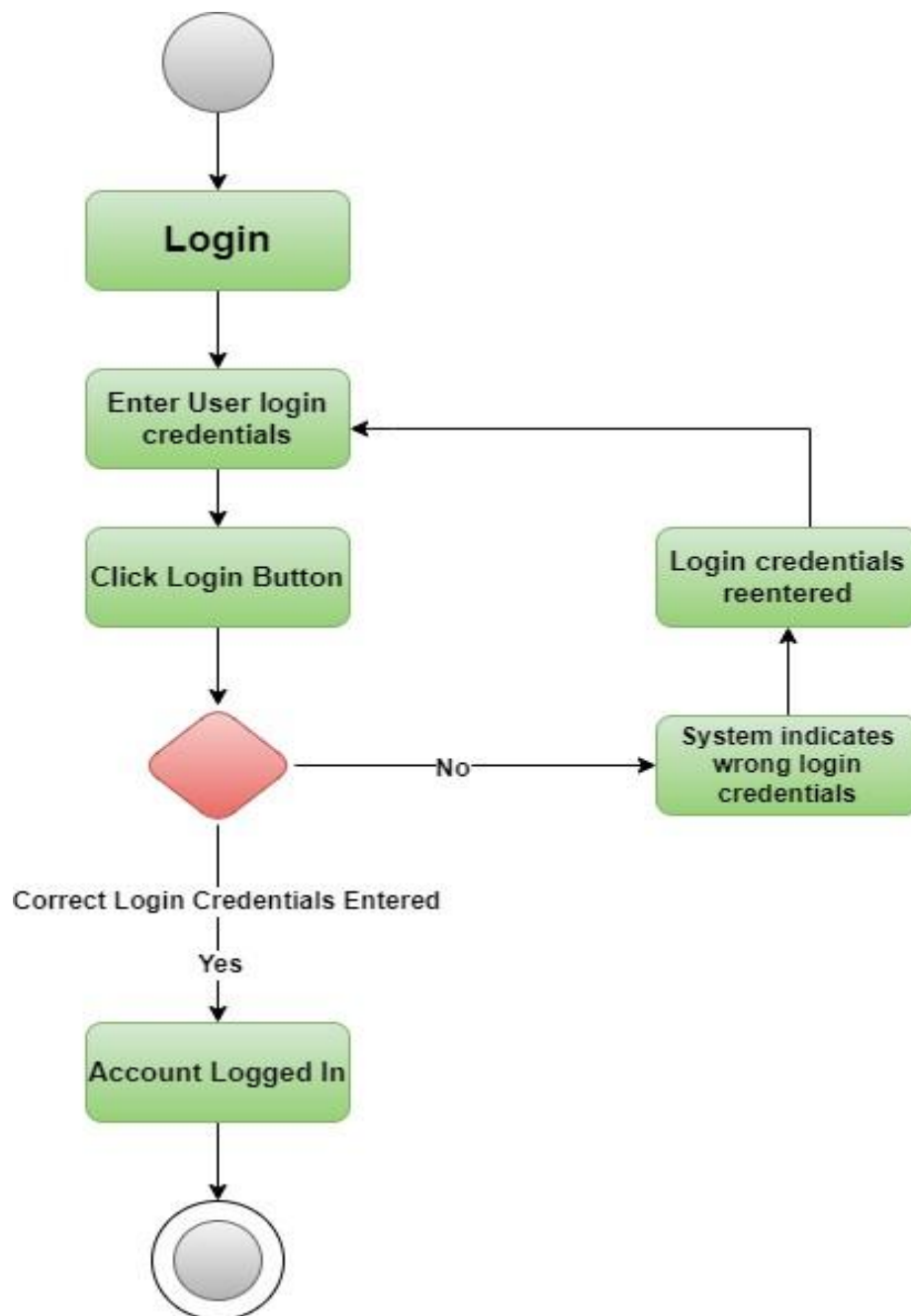


### 8.3 State chart Diagram

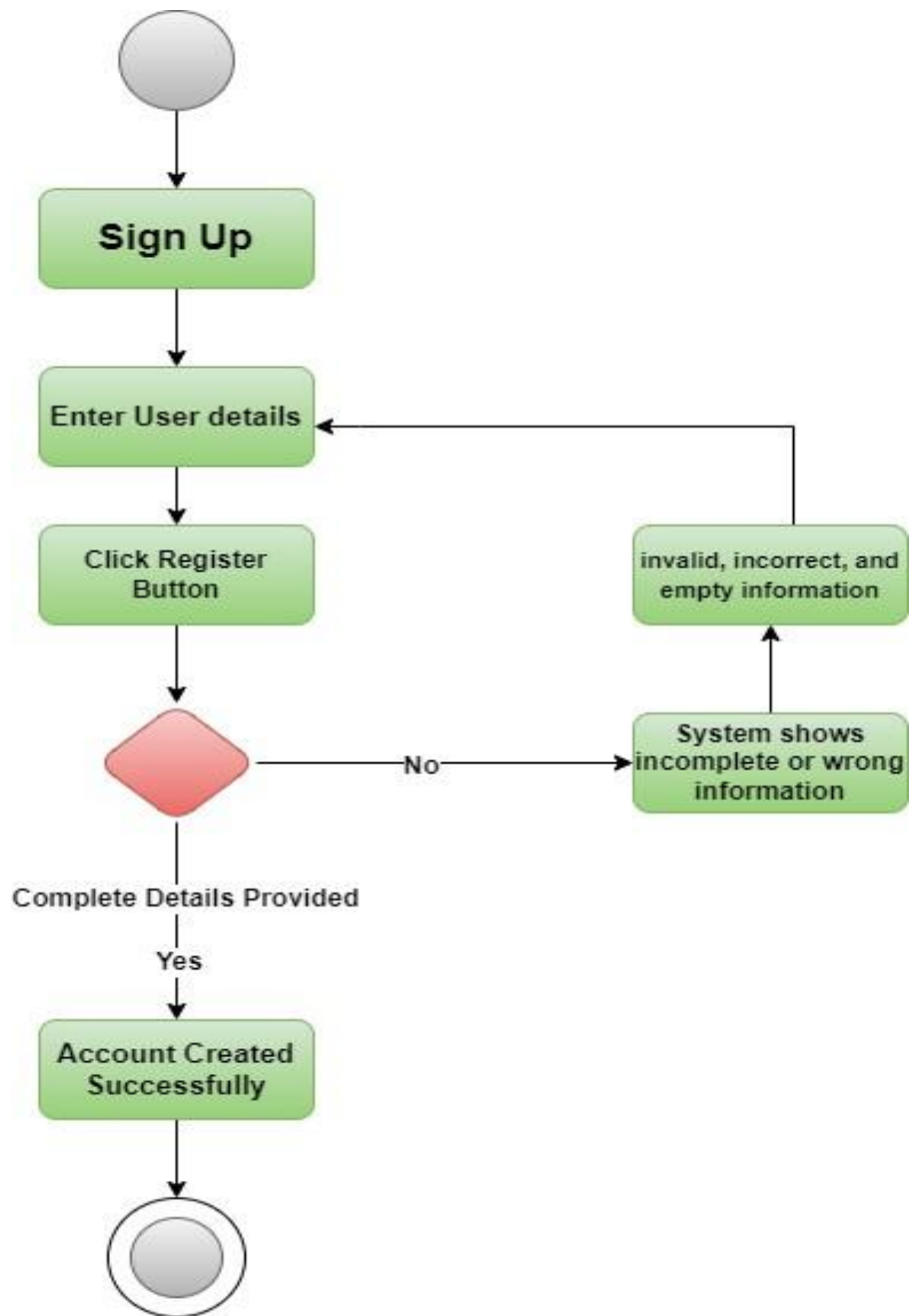


## 8.4 Activity Diagram

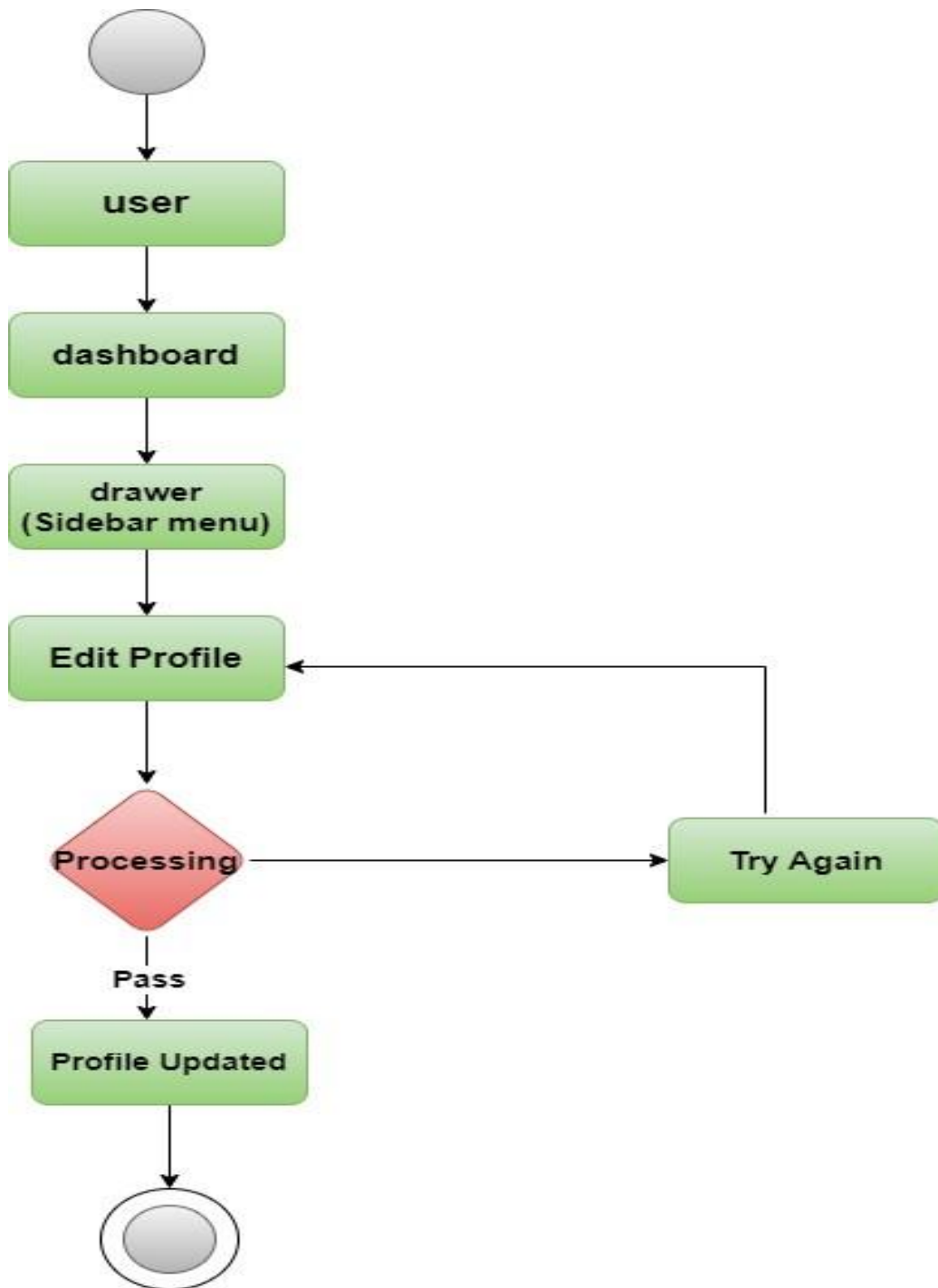
### 8.4.1 Login



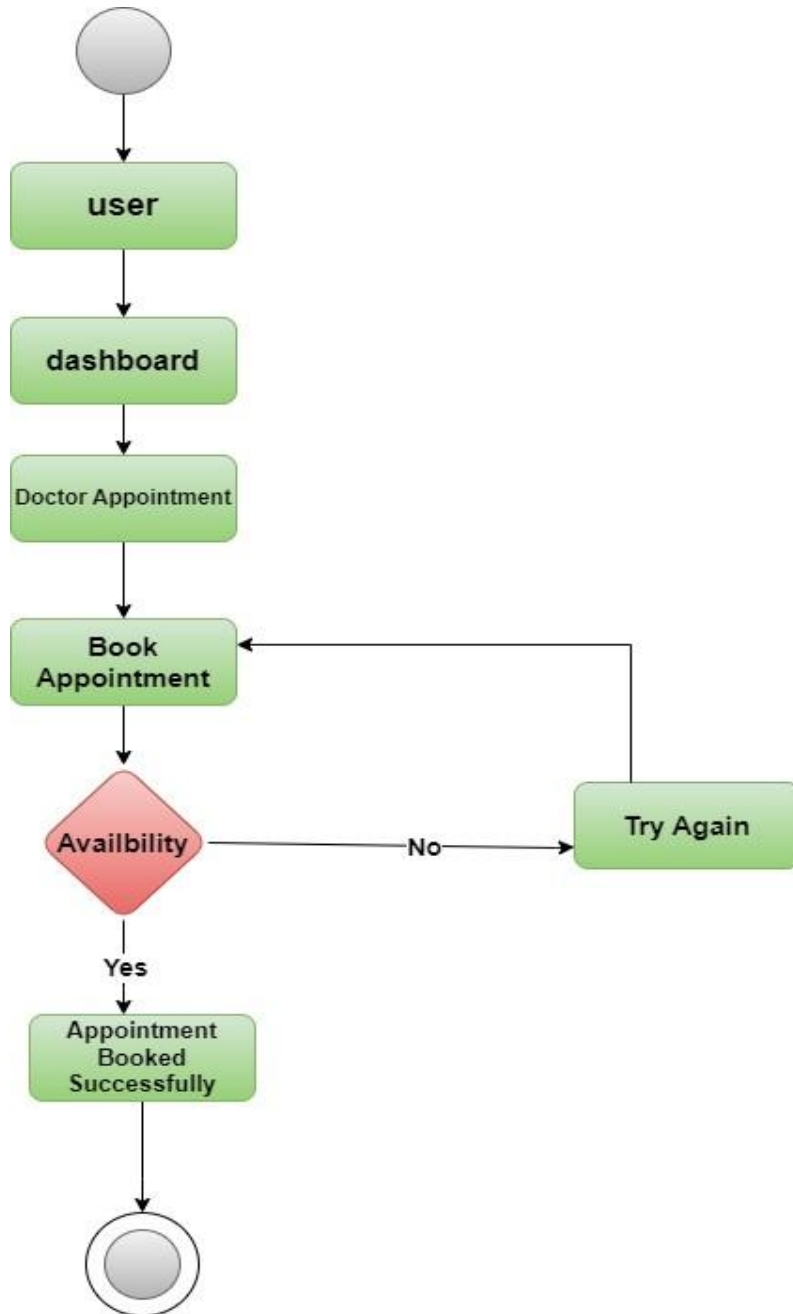
#### 8.4.2 Sign Up



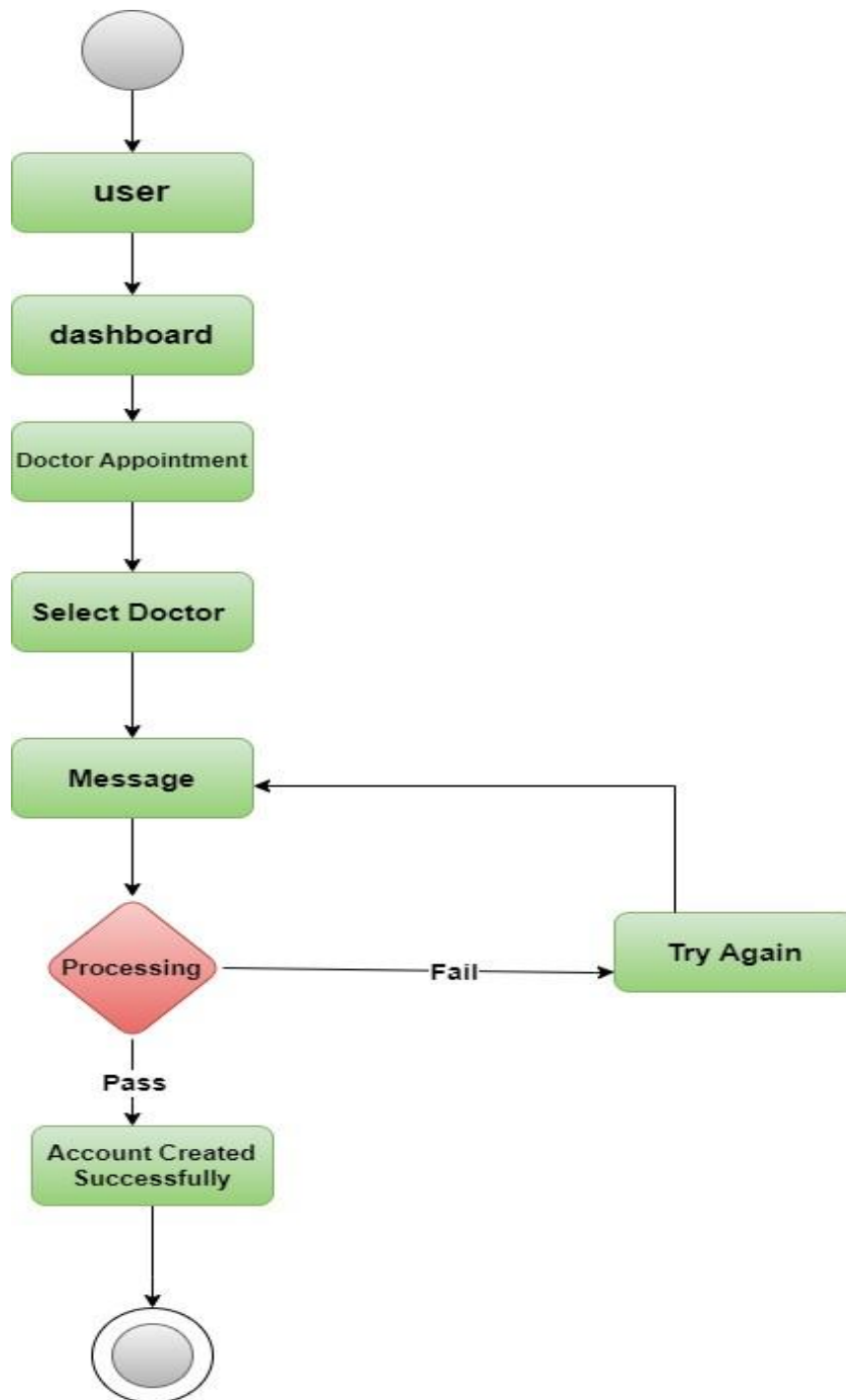
#### 8.4.3 Profile modification



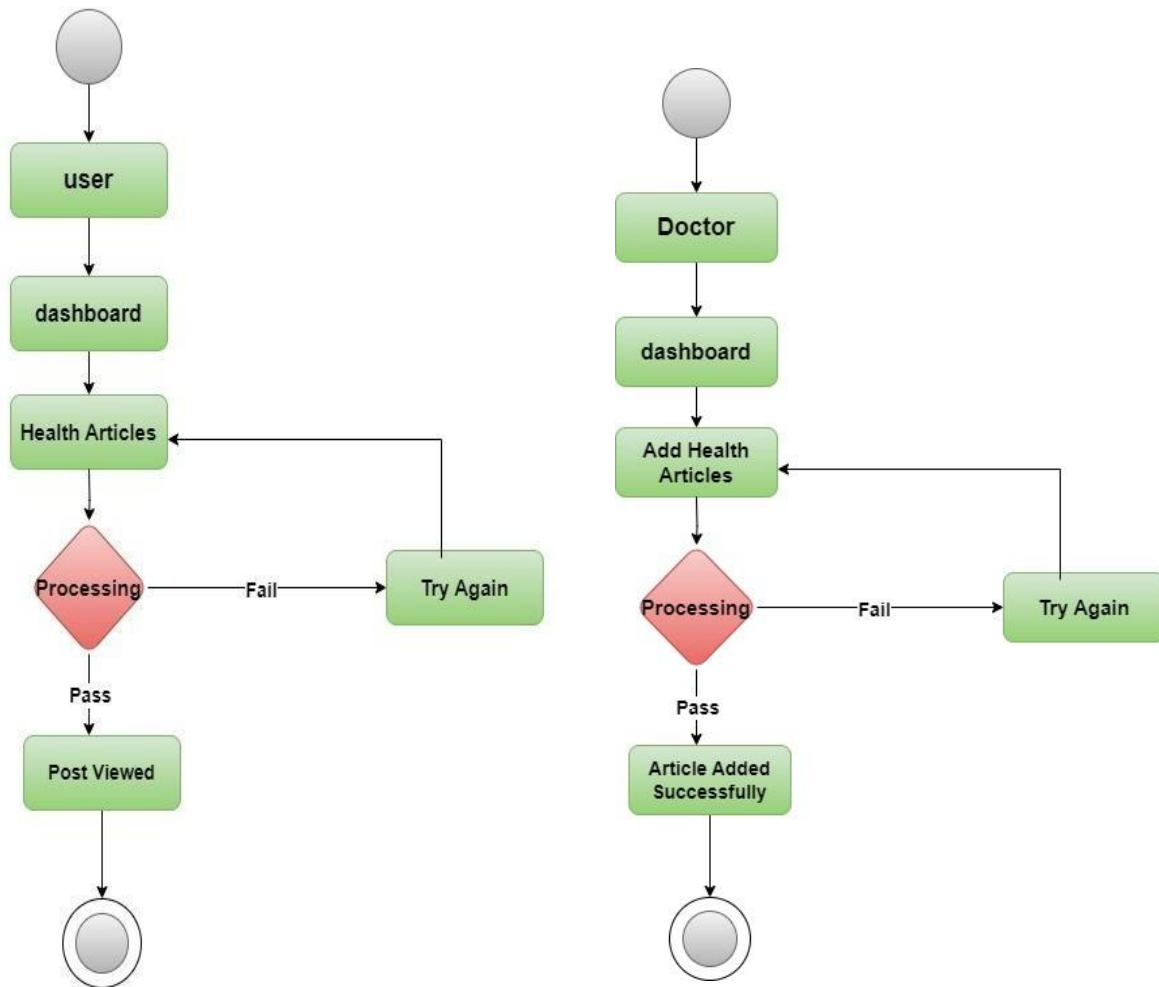
#### 8.4.4 Doctor Appointment



#### 8.4.5 Chat box

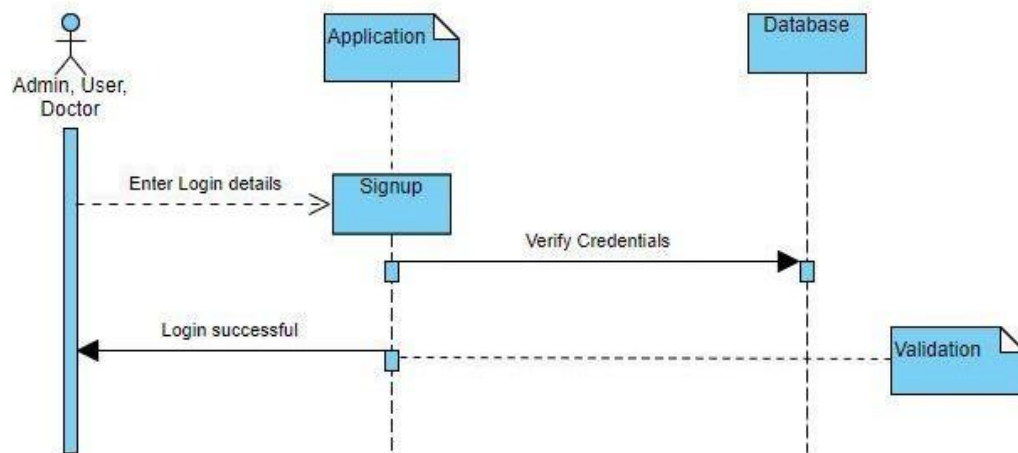


#### 8.4.6 Health Article:



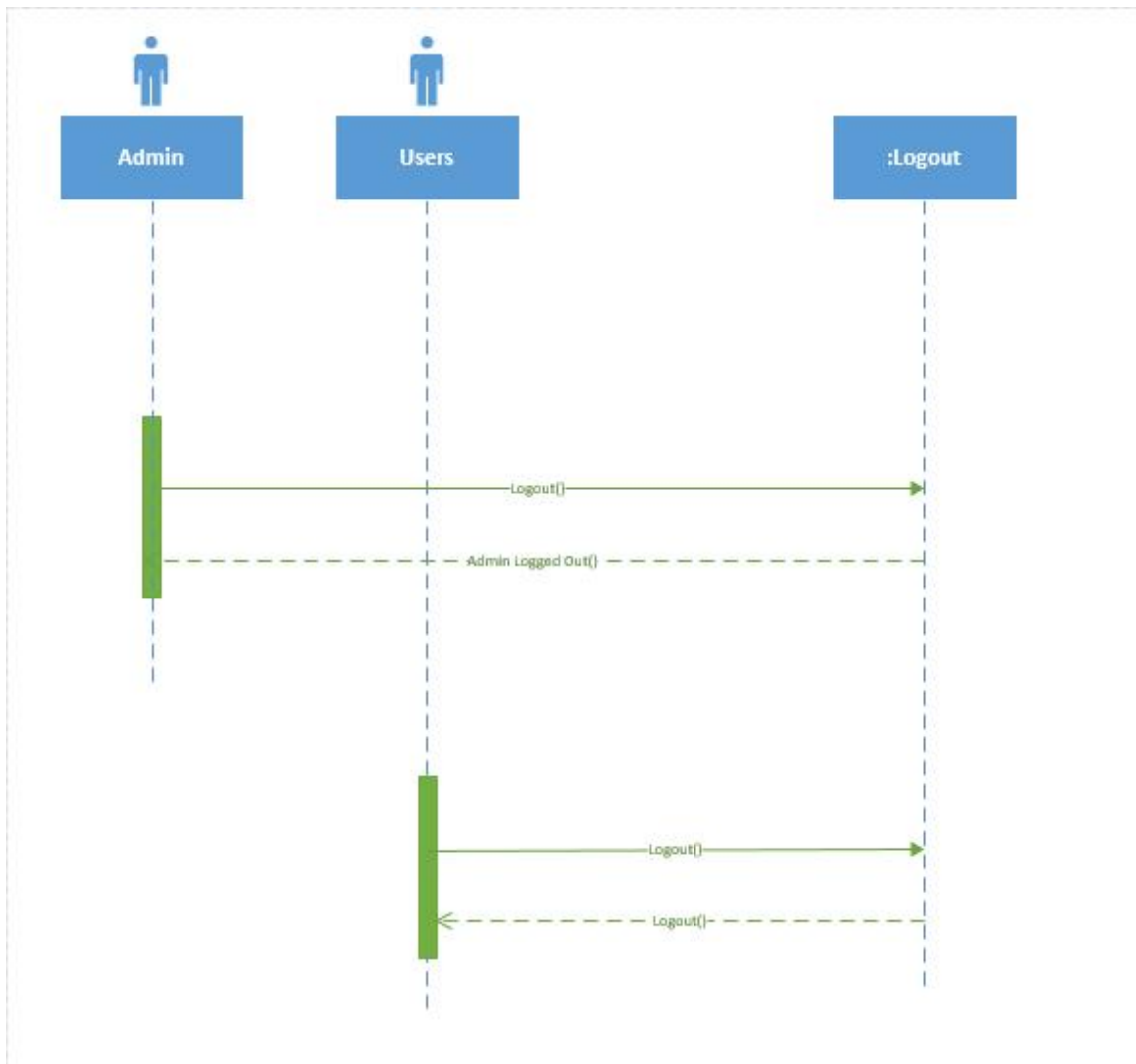
## 8.5 Sequence Diagram

### 8.5.1 Login

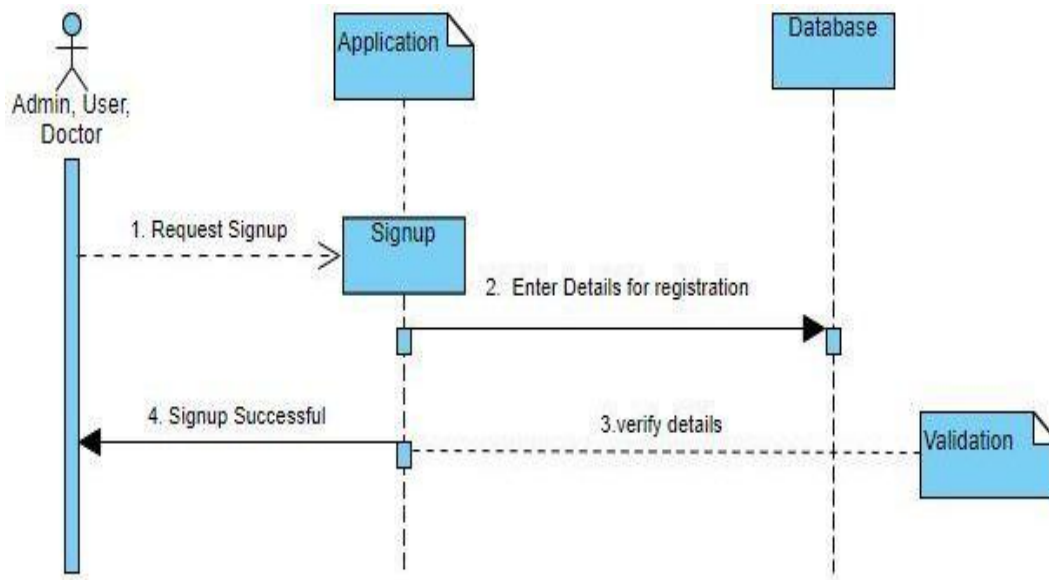




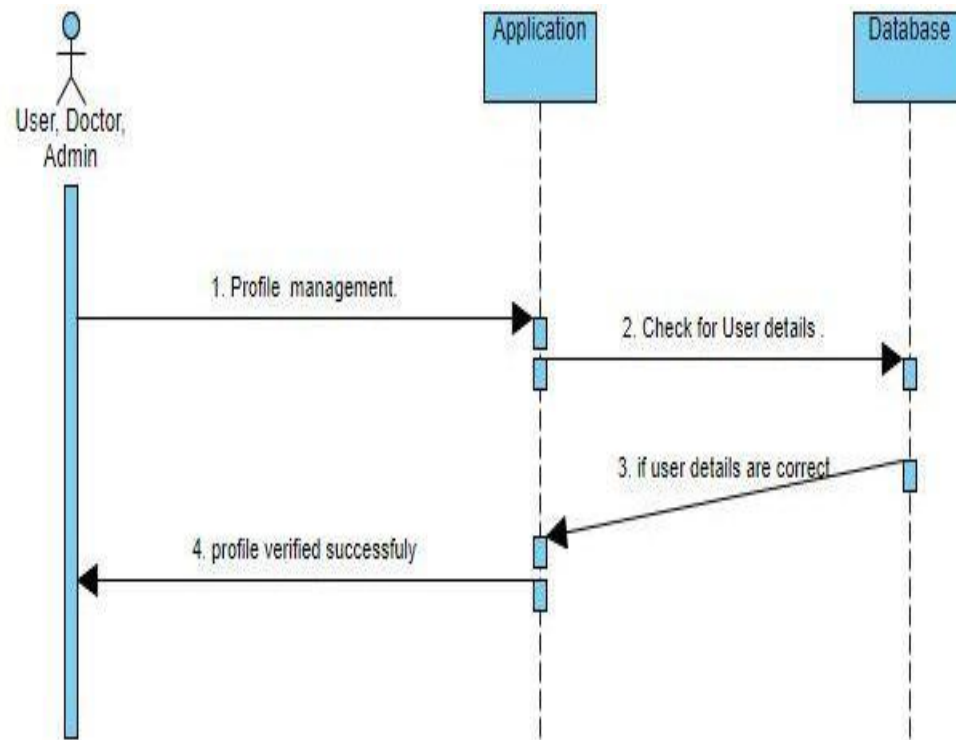
## 8.5.2 Logout



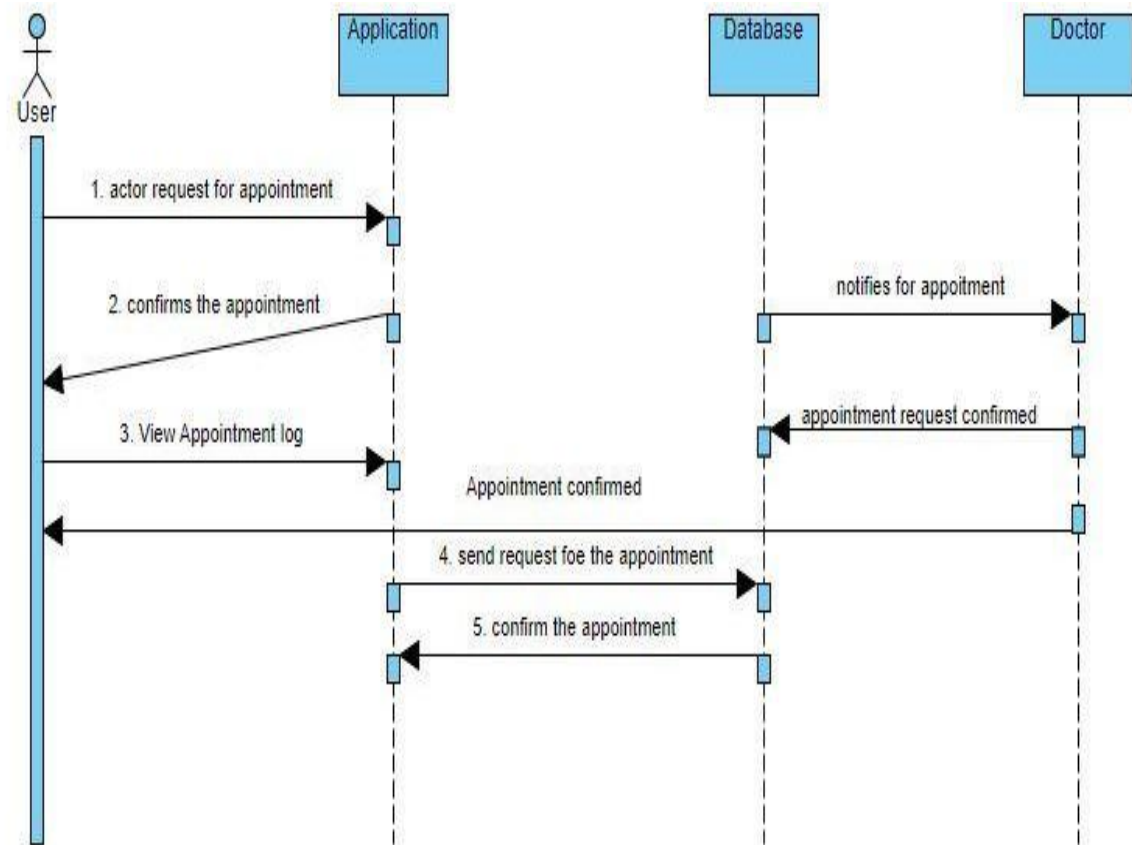
### 8.5.3 Sign Up



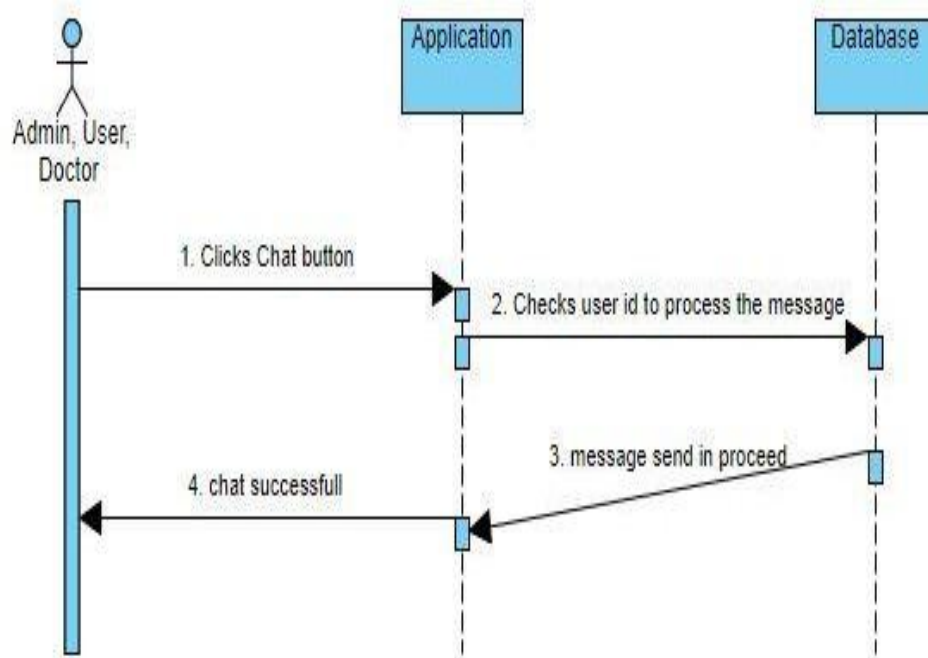
### 8.5.3 Profile modification



### 8.5.4 Doctor Appointment

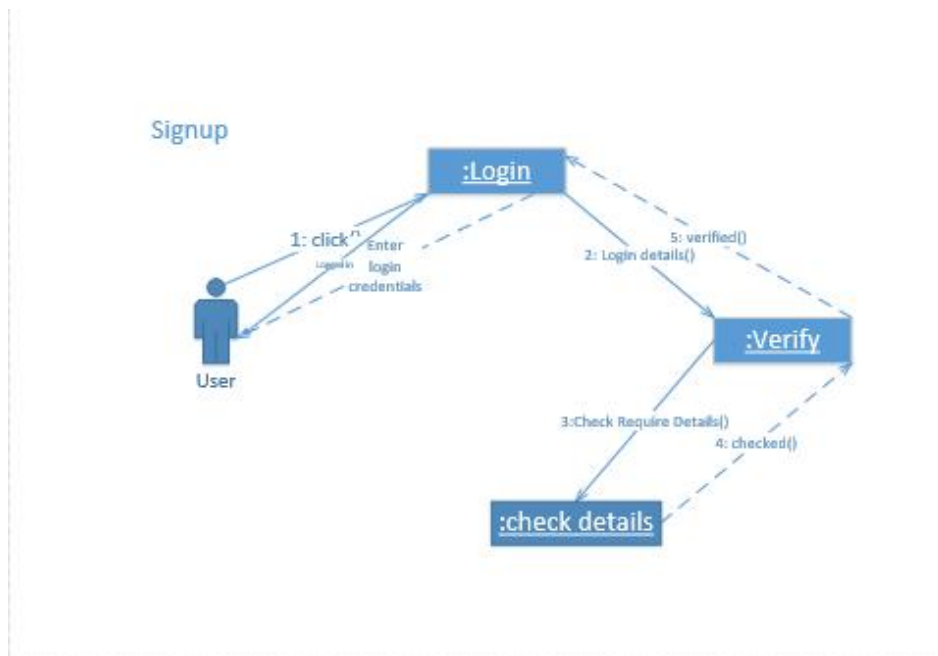


### 8.5.5 Chat box



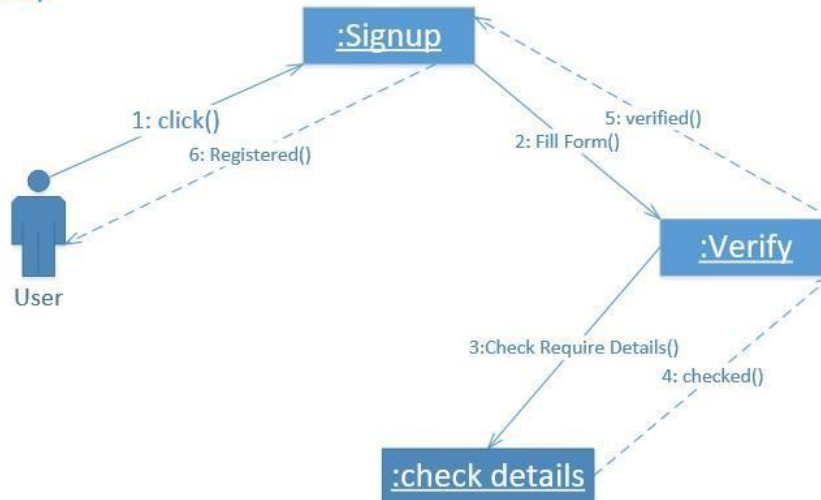
## 8.6 Collaboration Diagram

### 8.6.1 Login

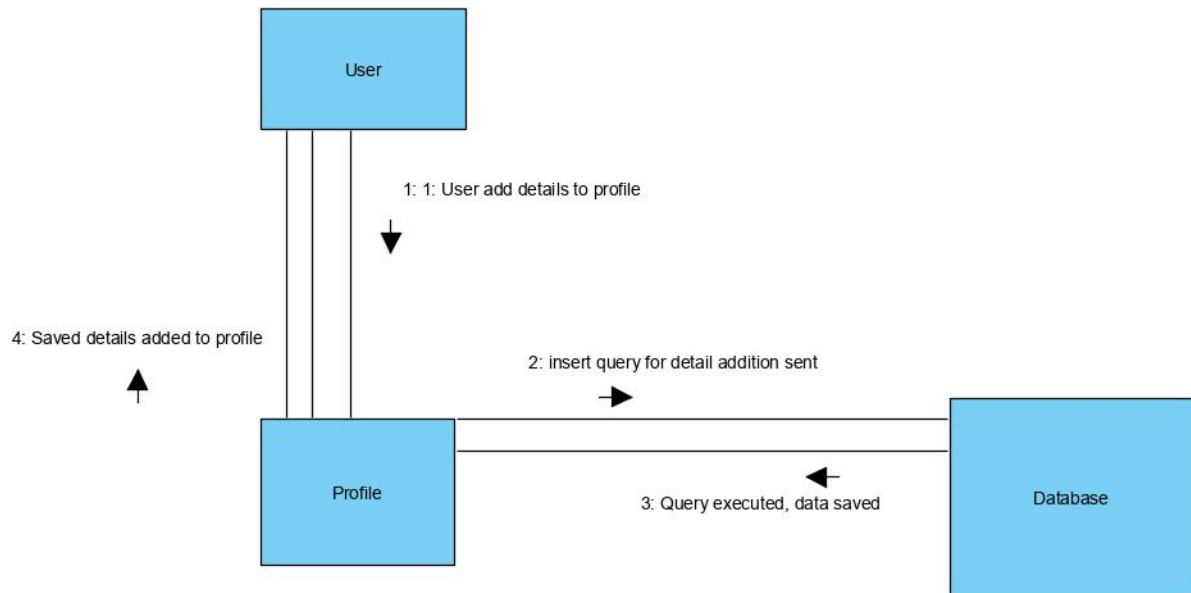


## 8.6.2 Sign up

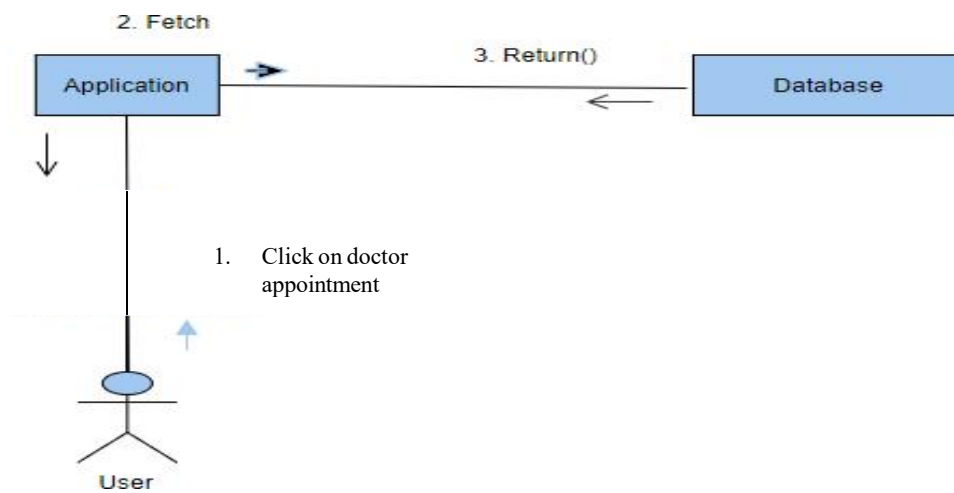
Signup



### 8.6.3 Profile modification

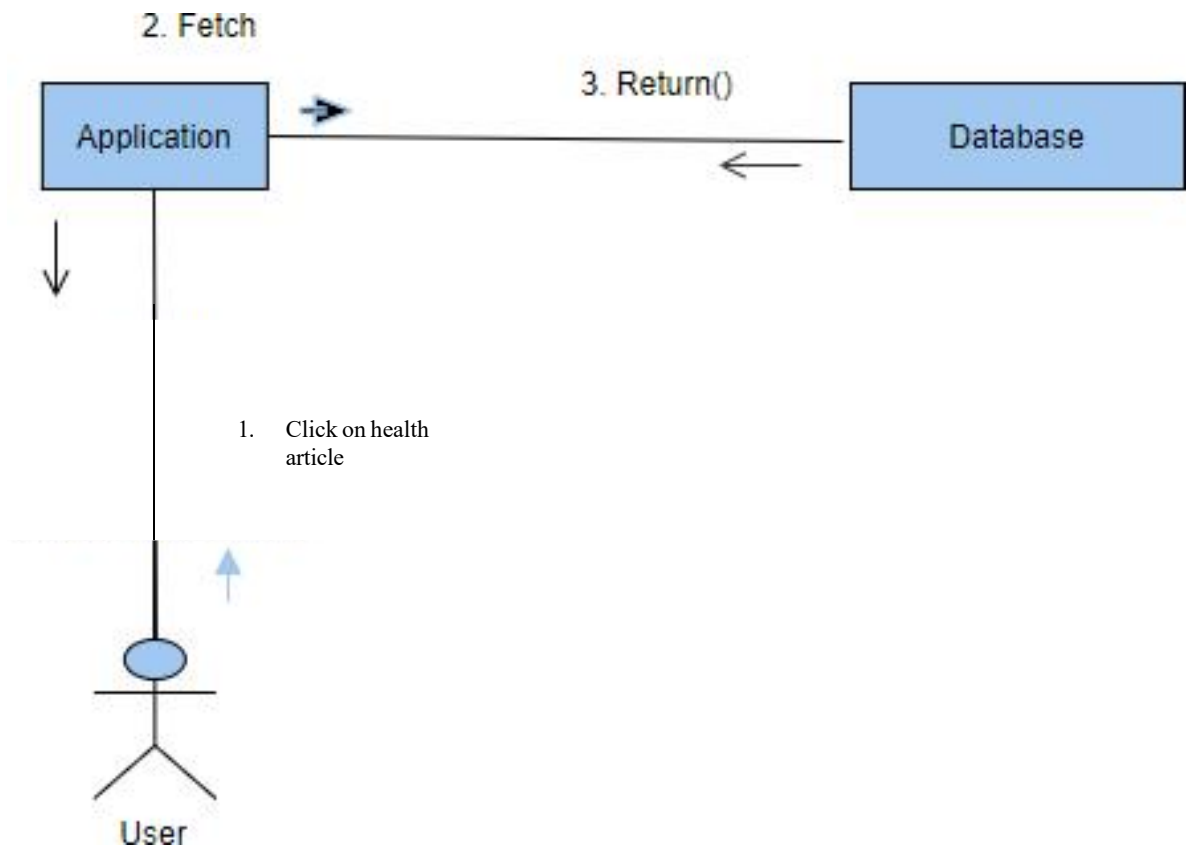


### 8.6.4 Doctor Appointment

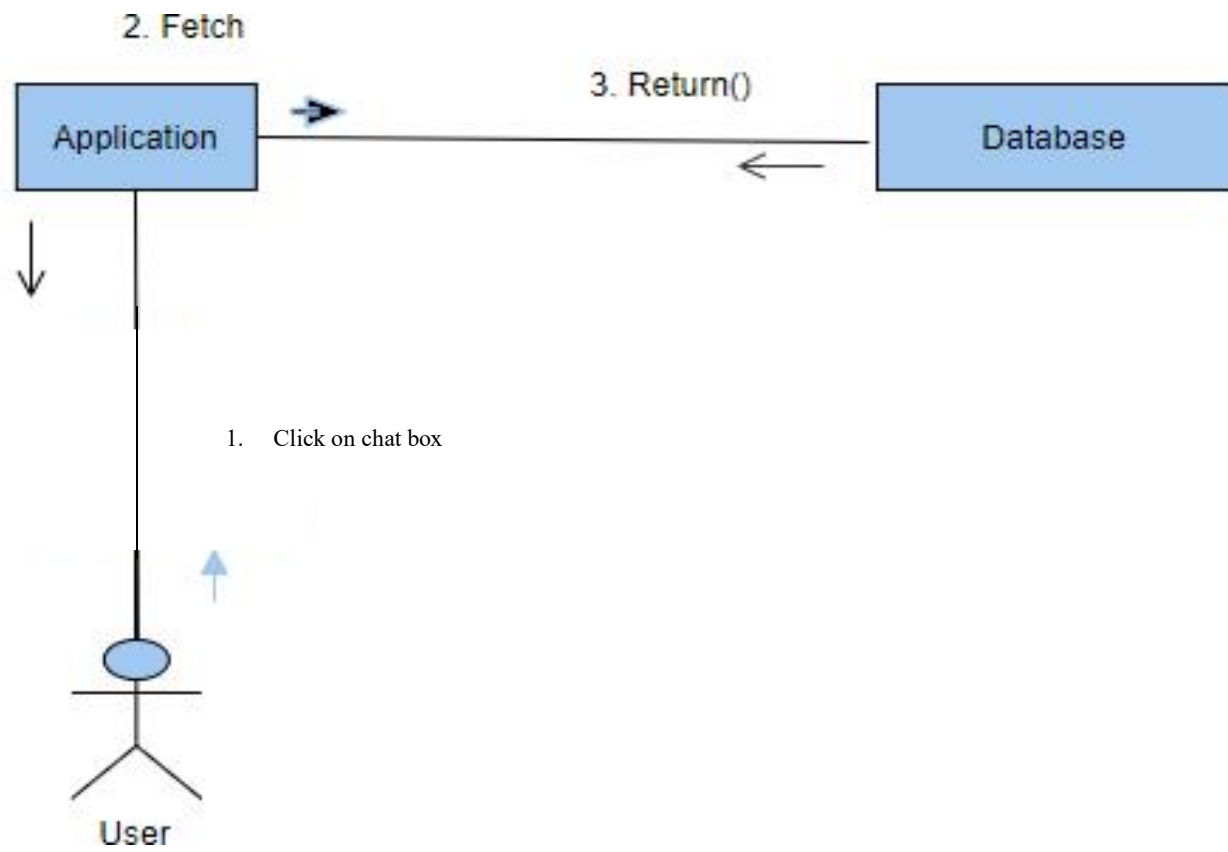




### 8.6.5 Health Article

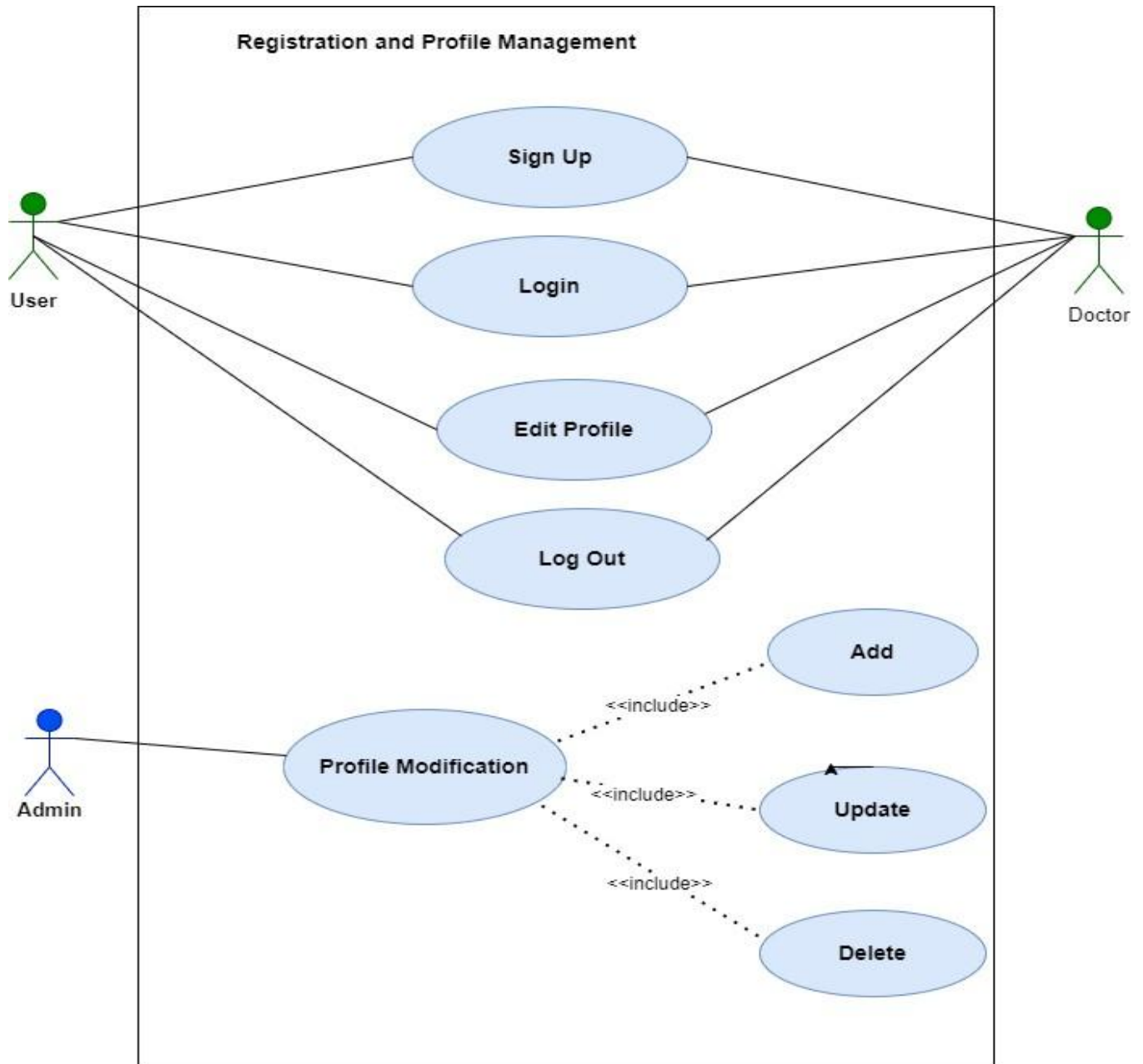


### 8.6.6 Chat box

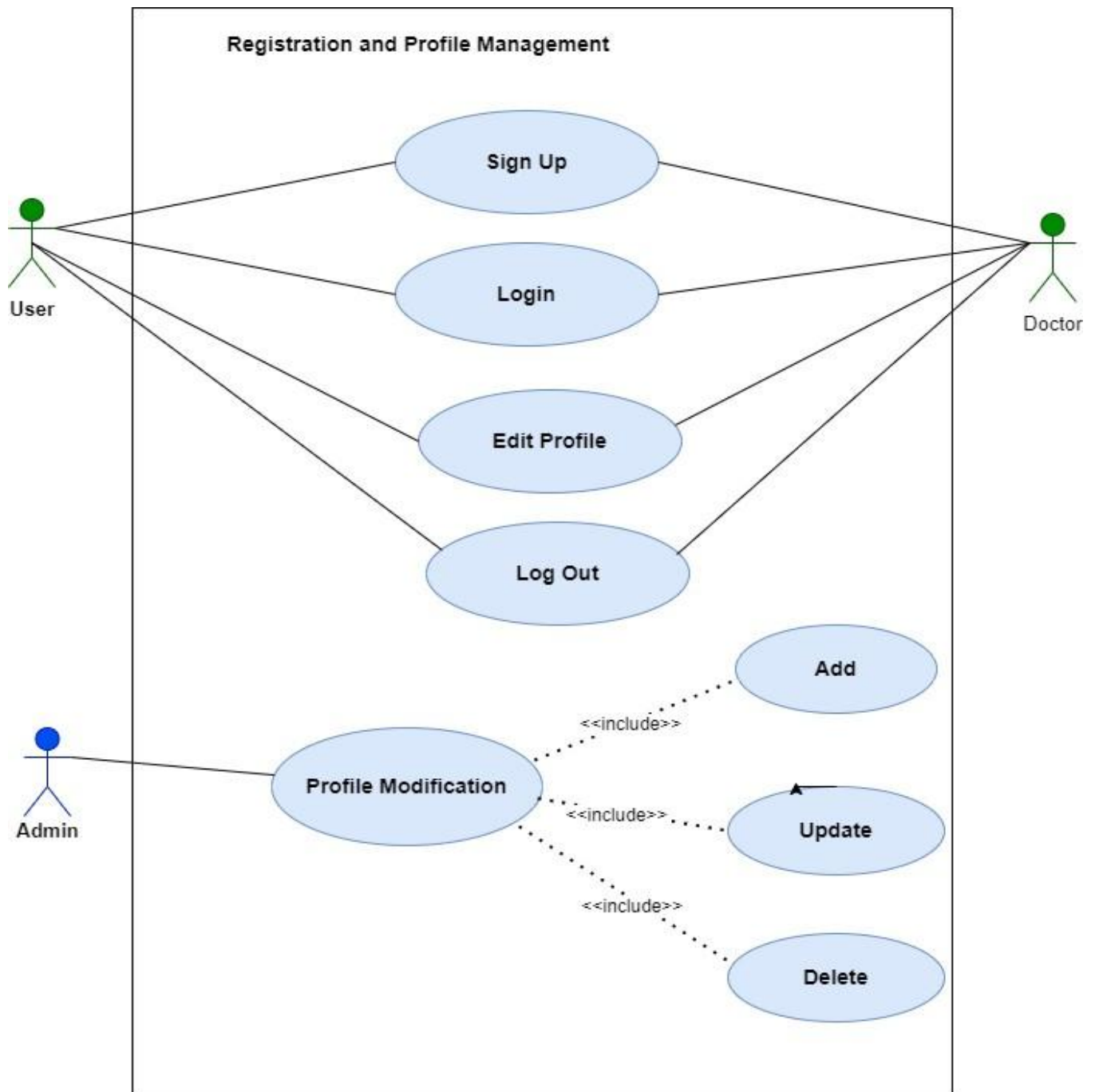


## 8.7 Use-Case Diagram

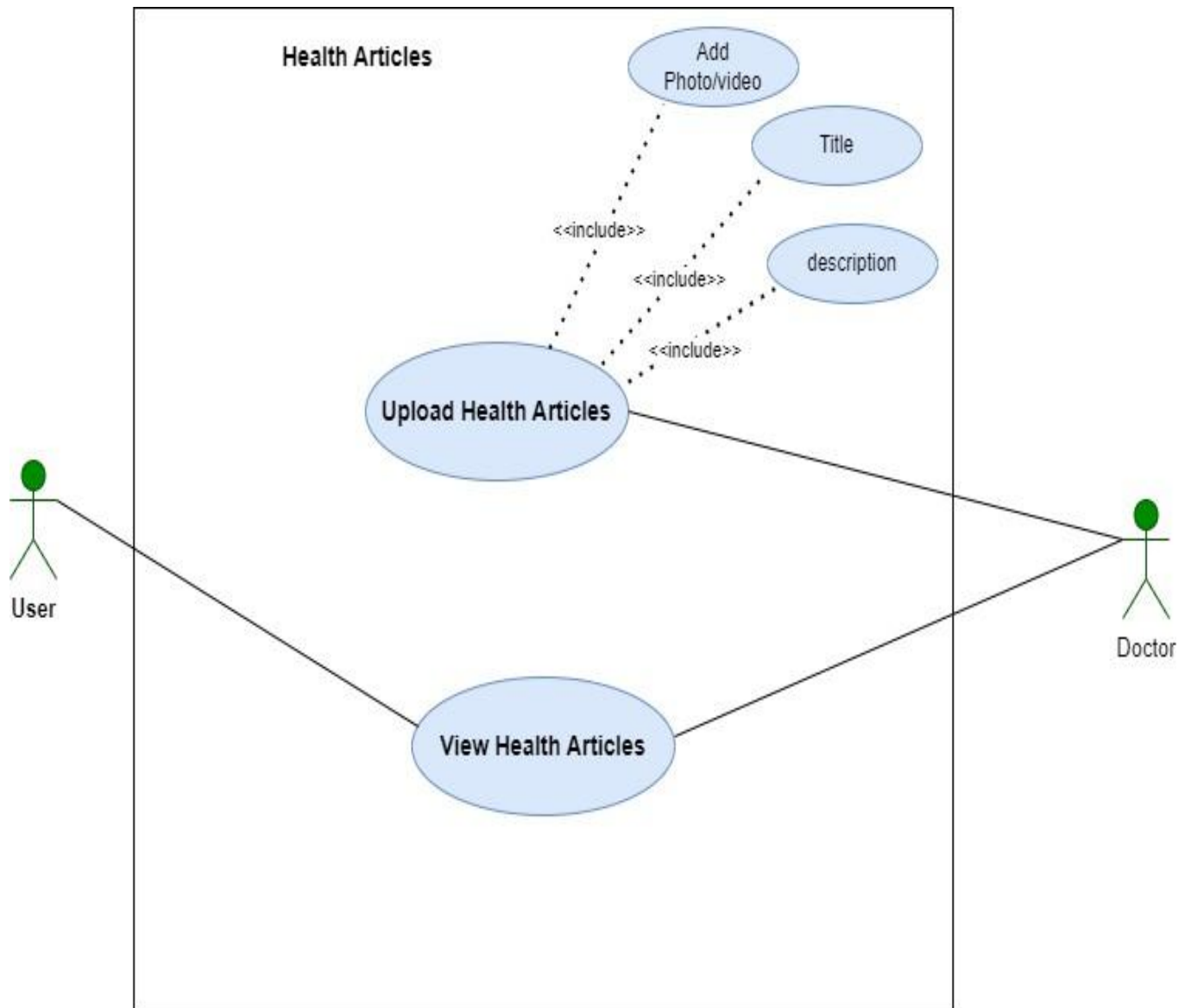
### 8.7.1 Use-Case Diagram 1



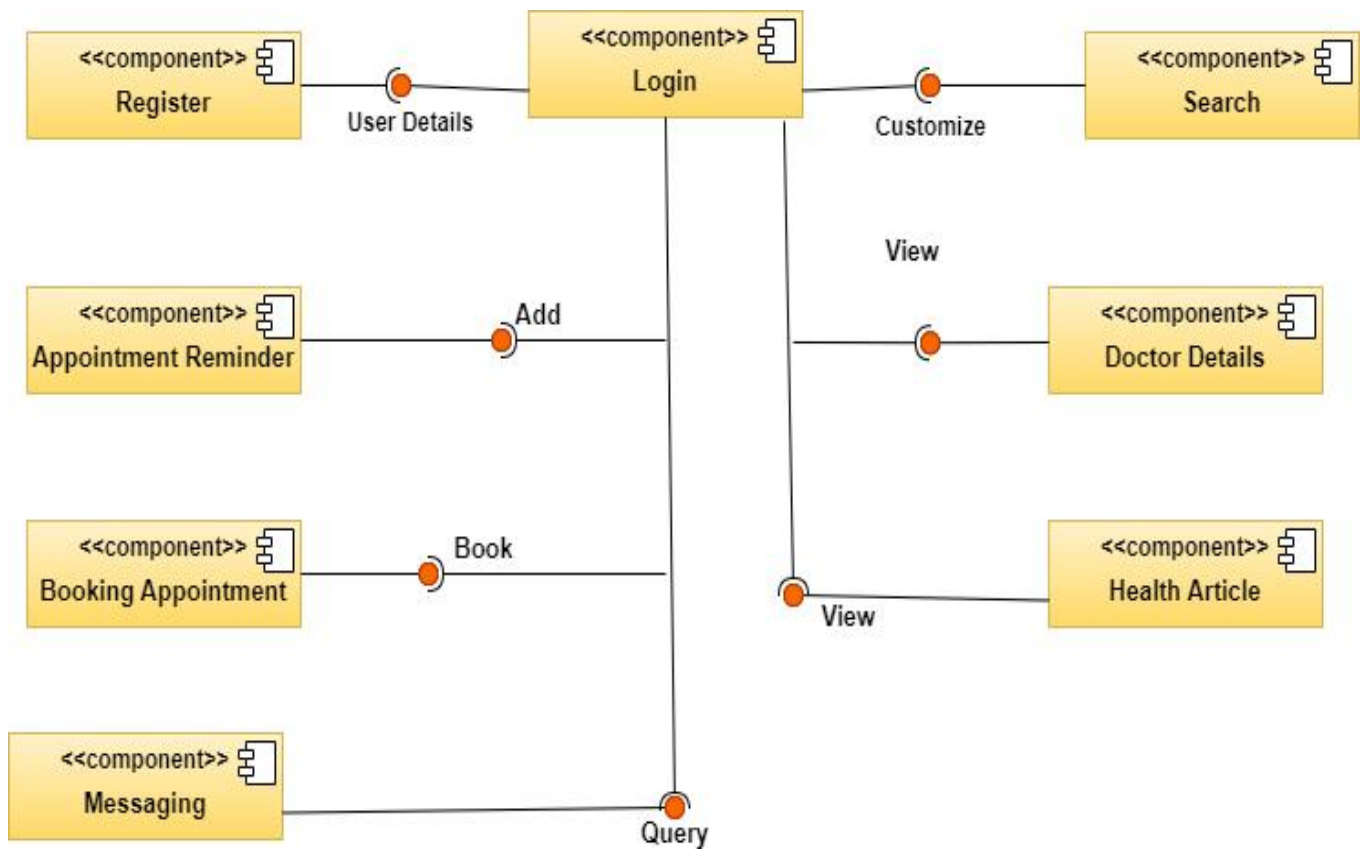
### 8.7.2 Use-Case Diagram 2



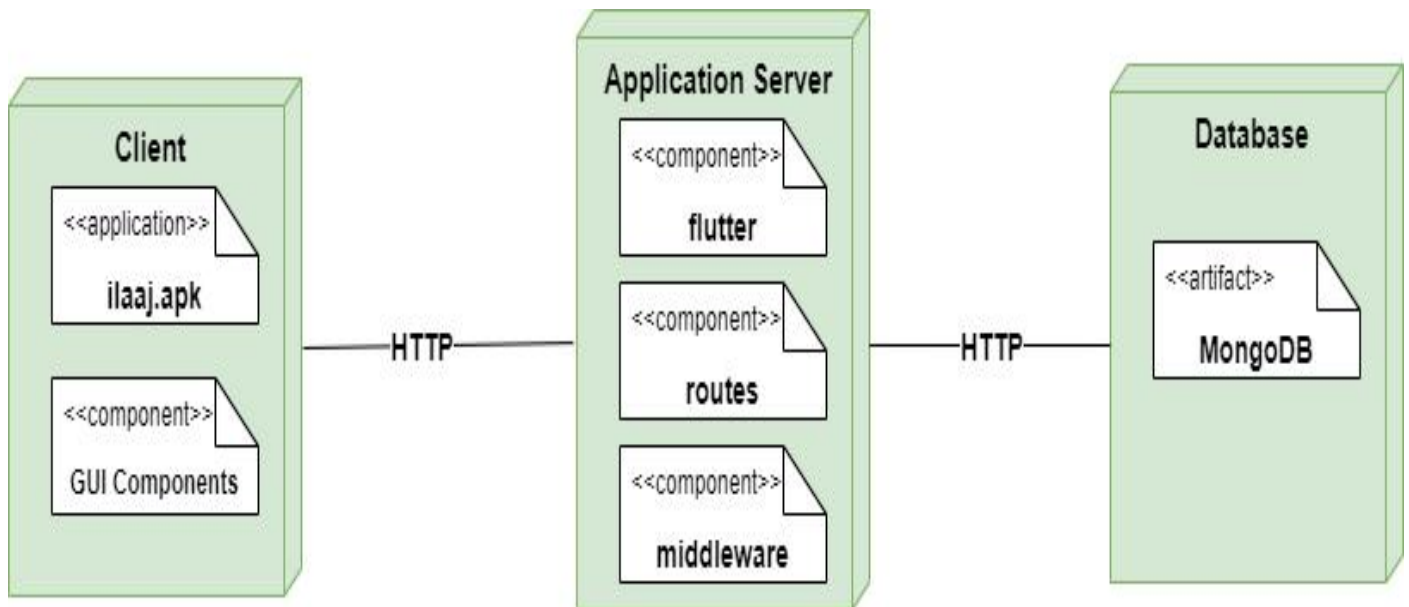
### 8.7.3 Use-Case Diagram 3



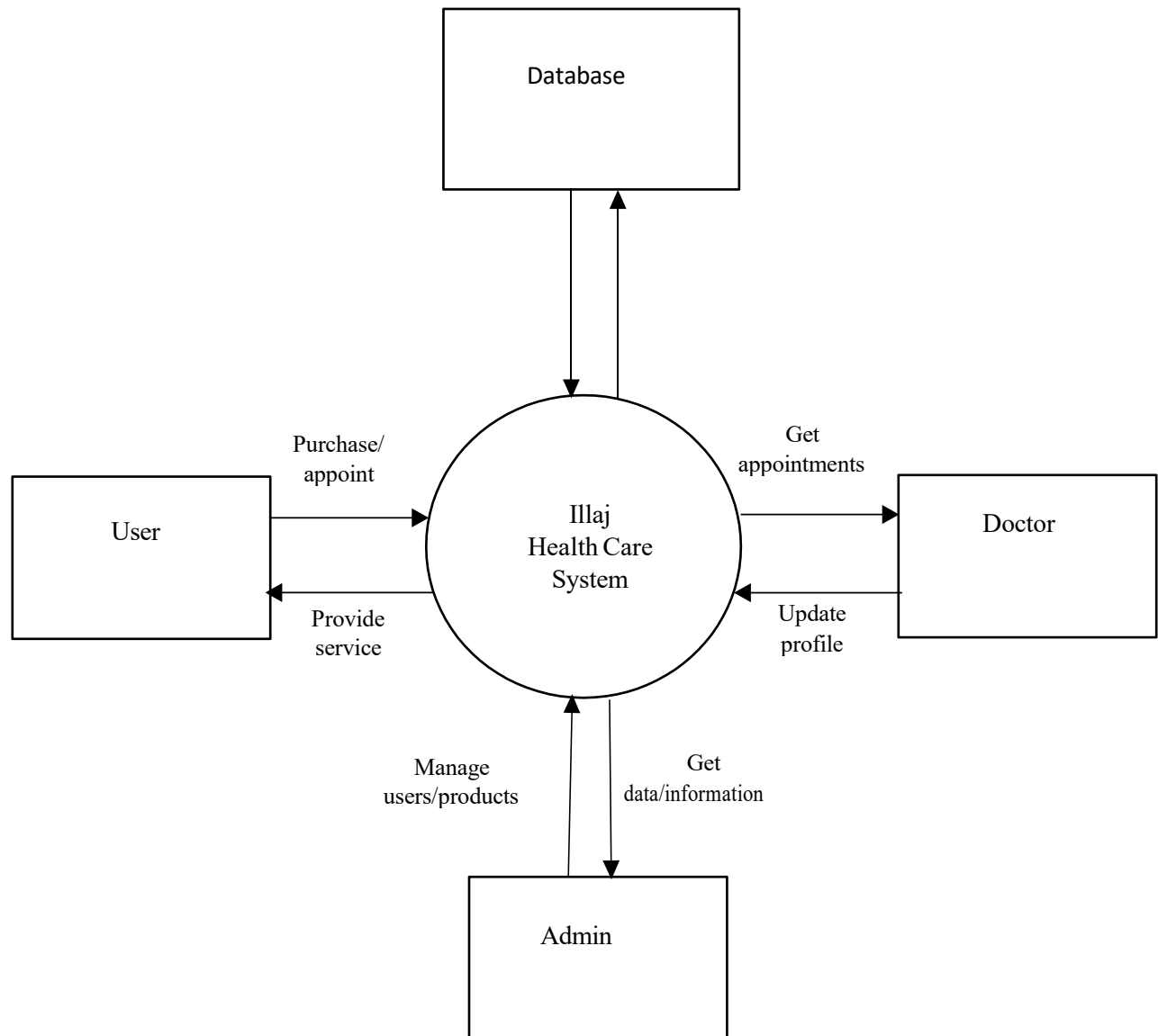
## 8.8 Component diagrams



## 8.9 Deployment diagram

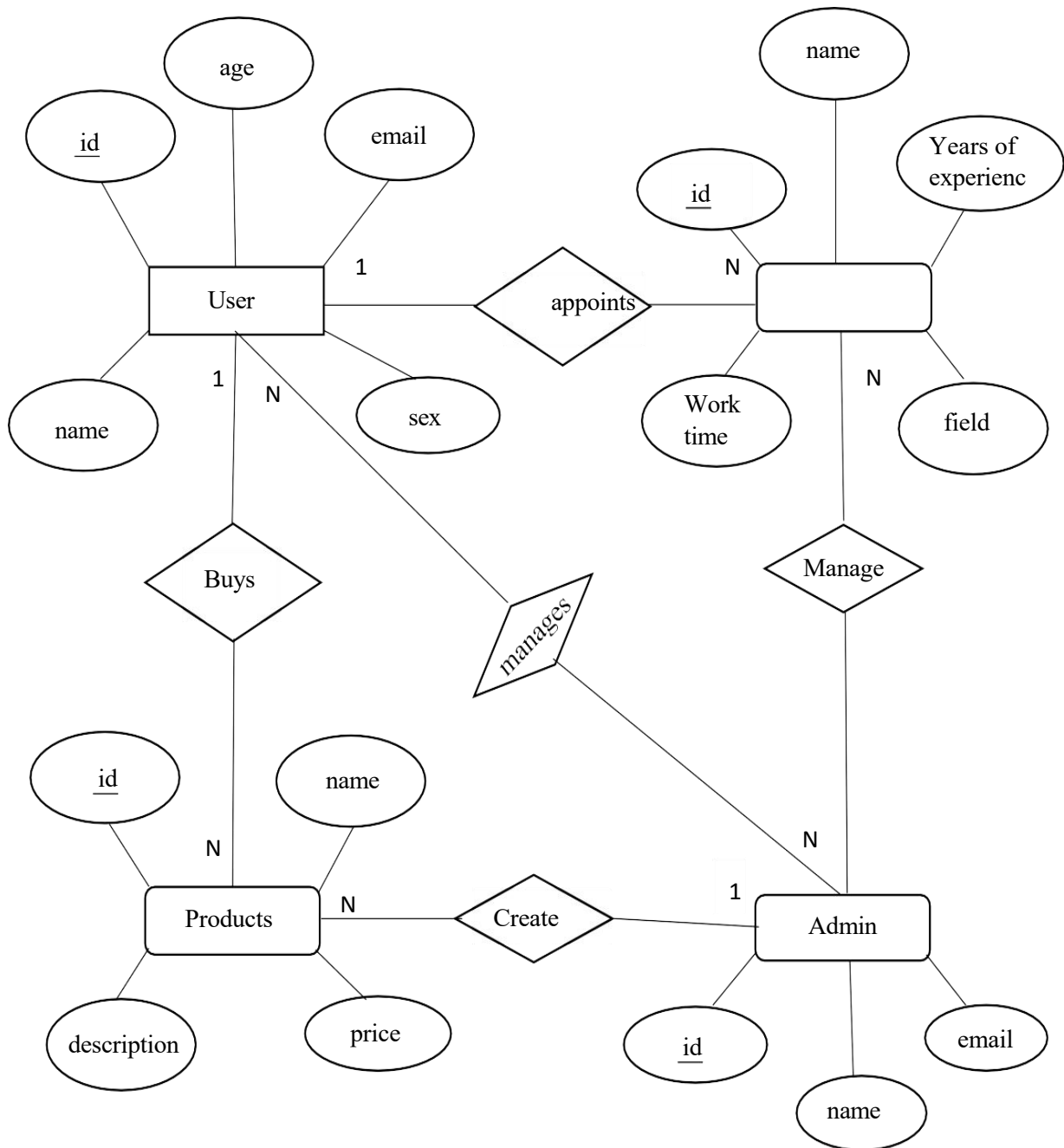


### 8.10 System block diagram





## 8.11 ERD



# Software Testing Document

## 1. Introduction

### 1.1 Purpose of this Document

The primary purpose of this software testing document is to define the testing strategy and procedures for the health mobile application. The testing process aims to ensure the reliability, functionality, and security of the application in accordance with the specified requirements. This document will serve as a guide for the testing team, providing a comprehensive overview of the testing approach and specific objectives.

### 1.2 Intended Audience

Our different audience is our supervisor, Faculty, Team Members and Users

### 1.3 Document Convention

Format: IEEE Standard

Font Family: Times New Roman Text Size: 11sp, 14sp, 18sp, 12sp Text Style: Bold

Text Color: Black#000000 Text Alignment: Left Align

### 1.4 Project Overview

The project will be designed for a specific online health care platform known as “(ilaaj)”. This project will help to personalize the health care for a particular user. The project Targets the general public with smart devices and also provides them with the precise Information about their health. The data thus collected and analyzed can be used by the doctors to determine patient’s health. Users can also use it to get general information about their condition. Overall, the scope of the project revolves around the users, Doctors and researchers. As the technology improves, it’s scope and impact in health Sector will expand.

## 2. Test Case Design and Description

### 2.1 Sign up

<b>Test Case ID:</b>	1			
<b>QA Test Engineer</b>	Naveen kingrani			
<b>Test case Version:</b>	1.0			
<b>Test Execution Date:</b>				
<b>Use Case Reference(s)</b>	1 (Signup)			
<b>GUI Reference(s)</b>	User registration page			
<b>Objective</b>	To register a new user and to check if the user already exists, the user details are registered and that there is no error in that process.			
<b>Pre-Requisite:</b>	The database must exist, in which users table must exist where the registered users will be inserted			
<b>Test Case Description</b>	This test case is going to check the Registration page submit button functionality, if we click it must insert record in the database and if there exists, it must prompt error.			
<b>Input Parameters</b>	<b>Expected Output</b>	<b>Actual Output</b>	<b>Test Conformance Status</b>	<b>Possible Reason(s) in case of failure</b>
Enter Name	User must be prompted to enter name in case of empty field	Name is accepted if valid input is entered	Passed	Database connectivity
Enter Password	This field must not be empty otherwise, it prompts error to Re-enter password.	Password field is checked for being non-empty	Passed	
Enter Email	User is prompted if an invalid email address is used.	The email address is accepted	Passed	
Click Sign Up Button	Checks all the fields are filled and email address does not already exists. Inserts the record in the database.	Checks if all the fields are filled and email is successfully registered.	Passed	Database connectivity

## 2.2 Login

<b>Test Case ID:</b>	2			
<b>QA Test Engineer</b>	Naveeen Kingrani			
<b>Test case Version:</b>	1.0			
<b>Test Execution Date:</b>				
<b>Use Case Reference(s)</b>	1 (Login)			
<b>GUI Reference(s)</b>	User login page			
<b>Objective</b>	To sign in as an existing user of the system			
<b>Pre-Requisite:</b>	The database must be connected so that the user credentials can be verified. The user must be registered.			
<b>Test Case Description</b>	This test case is going to check the if a user can successfully login into the system.			
<b>Input Parameters</b>	<b>Expected Output</b>	<b>Actual Output</b>	<b>Test Conformance Status</b>	<b>Possible Reason(s) in case of failure</b>
Correct email and password	Successful login	Successfully logged in to system	Passed	
Incorrect email and password	Fail to log in	Failure logging in	Passed (login failed)	Invalid email
Correct email incorrect password	Fail to login	Failure logging in	Passed (login failed)	Incorrect password
Incorrect username and incorrect password	Fail to login	Failure logging in	Passed (login failed)	Incorrect username and password

## 2.3 User Profile Modification

<b>Test Case ID:</b>	3			
<b>QA Test Engineer</b>	Arslan Ali			
<b>Test case Version:</b>	1.0			
<b>Test Execution Date:</b>				
<b>Use Case Reference(s)</b>	User profile modification			
<b>GUI Reference(s)</b>	User profile page			
<b>Objective</b>	Modifying user profile			
<b>Pre-Requisite:</b>	user is logged into the system			
<b>Test Case Description</b>	This test case is going to check if a user can successfully modify his/her profile information.			
<b>Input Parameters</b>	<b>Expected Output</b>	<b>Actual Output</b>	<b>Test Conformance Status</b>	<b>Possible Reason(s) in case of failure</b>
Change name	Name updated	Name successfully updated	Passed	<ul style="list-style-type: none"> <li>• Session expired</li> <li>• Disconnected from server</li> </ul>
Change password	Password updated	Password successfully updated	Passed	<ul style="list-style-type: none"> <li>• Session expired</li> <li>• Disconnected from server</li> </ul>

## 2.4 Doctor Appointment

<b>Test Case ID:</b>	4			
<b>QA Test Engineer</b>	Naveen Kingrani			
<b>Test case Version:</b>	1.0			
<b>Test Execution Date:</b>				
<b>Use Case Reference(s)</b>	Doctor Appoitment			
<b>GUI Reference(s)</b>	User homepage			
<b>Objective</b>	Doctor Appointment			
<b>Pre-Requisite:</b>	User is logged into the system			
<b>Test Case Description</b>	This test case is going to check if a user can successfully appoint the Doctor will se the functionality is working or not.			
<b>Input Parameters</b>	<b>Expected Output</b>	<b>Actual Output</b>	<b>Test Conformance Status</b>	<b>Possible Reason(s) in case of failure</b>
Doctor appointment	1. Choose a Doctor 2. Appointment confirmed	1. Doctor Choice specified. 2. Doctor appointed successfully	Passed	<ul style="list-style-type: none"> <li>• Session expired</li> <li>• Disconnected from server</li> </ul>

## 2.5 Search

<b>Test Case ID:</b>	5			
<b>QA Test Engineer</b>	Arslan Ali			
<b>Test case Version:</b>	1.0			
<b>Test Execution Date:</b>				
<b>Use Case Reference(s)</b>	Search Bar			
<b>GUI Reference(s)</b>	Search bar in dashboard			
<b>Objective</b>	Searching			
<b>Pre-Requisite:</b>	User is logged into the system			
<b>Test Case Description</b>	This test case is going to check if a user can successfully search the specified Doctor			
<b>Input Parameters</b>	<b>Expected Output</b>	<b>Actual Output</b>	<b>Test Conformance Status</b>	<b>Possible Reason(s) in case of failure</b>
Enter string in the search box	<ol style="list-style-type: none"> <li>String should be entered in the box</li> <li>Option and range should be selected</li> </ol>	<ol style="list-style-type: none"> <li>String executed</li> <li>Output Searched successful</li> </ol>	Passed	<ol style="list-style-type: none"> <li>Code has errors.</li> <li>Data base connection failed.</li> </ol>

## 2.6 Chat box

<b>Test Case ID:</b>	6			
<b>QA Test Engineer</b>	Naveen Kingrani			
<b>Test case Version:</b>	1.0			
<b>Test Execution Date:</b>				
<b>Use Case Reference(s)</b>	Chat box			
<b>GUI Reference(s)</b>	IDE			
<b>Objective</b>	Asynchronous communication			
<b>Pre-Requisite:</b>	User is logged into the system			
<b>Test Case Description</b>	This test case is going to check if a user can successfully communicate with doctor through chat			
<b>Input Parameters</b>	<b>Expected Output</b>	<b>Actual Output</b>	<b>Test Conformance Status</b>	<b>Possible Reason(s) in case of failure</b>
chat	1. User can chat with doctor	1. Message delivered in the chat module	Passed	1. User not logged in 2. Server connectivity lost

## 2.7 Add Pharmaceutical product

<b>Test Case ID:</b>	7			
<b>QA Test Engineer</b>	Naveen Kingrani			
<b>Test case Version:</b>	1.0			
<b>Test Execution Date:</b>				
<b>Use Case Reference(s)</b>	Pharmaceutical Product			
<b>GUI Reference(s)</b>	IDE			
<b>Objective</b>	Asynchronous communication			
<b>Pre-Requisite:</b>	User is logged into the system			
<b>Test Case Description</b>	This test case is going to check if a user can successfully order or add product in the pharma product section			
<b>Input Parameters</b>	<b>Expected Output</b>	<b>Actual Output</b>	<b>Test Conformance Status</b>	<b>Possible Reason(s) in case of failure</b>
Add	2. User can add the specific product	1. product added successfully	Passed	3. User not logged in 4. Server connectivity lost



## 2.8 S.O.S Feature

<b>Test Case ID:</b>	8			
<b>QA Test Engineer</b>	Arslan Ali			
<b>Test case Version:</b>	1.0			
<b>Test Execution Date:</b>				
<b>Use Case Reference(s)</b>	S.O.S feature			
<b>GUI Reference(s)</b>	IDE			
<b>Objective</b>	Asynchronous communication			
<b>Pre-Requisite:</b>	User is logged into the system			
<b>Test Case Description</b>	This test case is going to check if a user can successfully use sos feature and request the first responders			
<b>Input Parameters</b>	<b>Expected Output</b>	<b>Actual Output</b>	<b>Test Conformance Status</b>	<b>Possible Reason(s) in case of failure</b>
S.O.S	3. User can request the first responders	1. Message and request delivered to the first responders	Passed	5. User not logged in 6. Server connectivity lost

# User Manual

## 1. General Information

### 1.1 System Overview and Project Scope

The idea for our final year project is a mobile health application. The project scope involves the development of a two-phase Smartphone app for mobile health care system. FYP 1 focuses on features like Doctor Appointment and patient consultation, while FYP 2 introduces Pharmacy service, notifications, SOS feature, and admin panel. The app aims to cater to users, healthcare professionals, and administrators. The primary goal is to provide a comprehensive, user-friendly tool for managing health application well-being in a convenient and accessible manner.

### 1.2 Platforms

The application is a mobile application that be accessed through the local host address or DNS on web browsers or emulator.

### 1.3 Acronyms and Abbreviations

### 1.4 System Functions and Configuration's

- 1.4.1 User registration and login
- 1.4.2 Dashboard visualize
- 1.4.3 Doctor appointment
- 1.4.4 User can search doctor
- 1.4.5 User can chat with doctor
- 1.4.6 User can access health articles
- 1.4.7 User and doctor can update their profile

The application is accessible by the users through the defined web address. Users can sign up, sign in and use the facilities provided.

## 2. System Summary

### 2.1 System Overview

The system will build on industry standard, three tier web application, the three tiers are: Data tier, user tier and application tier.

- **Data Tier:** Mongodb server acts as data tier, the relational database schema provides a standard definition and storage of data required by the system,
- **User Tier:** User will access the system through application, both applications present the user with multiple pages and views of data stored in database server.
- **Application Tier:** App User will access the system through android application, the android application presents the user with activity forms and view of data stored in database server.

### 2.2 User Level Access

After a successful sign up the users will be able to use the application through login.

### 2.3 Contingencies

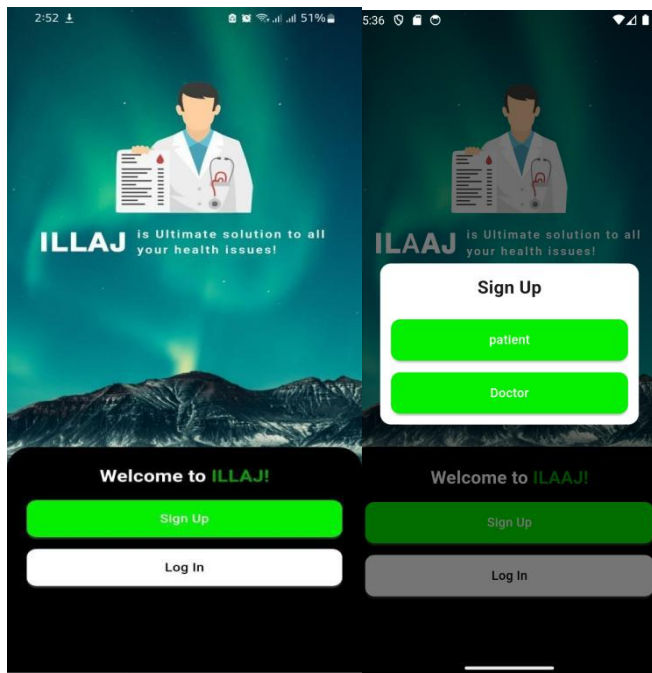
Regular backup should be performed, due to the fact that in the event of an emergency and/or accidents, stored data could become lost.

## 3. System Features

Following are the user screens for ilaaj

### 3.1 Sign up

This screen enables the user to sign up. User have two option to sign up as either as patient or as doctor. Then user have to add sign up details and click sign up button.



**Sign Up**

Already have an account? [Sign In!](#)

Name\*

Email\*

Phone\*

Password\*

Confirm Password\*

**Sign Up**

**Sign Up**

Already have an account? [Sign In!](#)

Name\*

Email\*

Experience\*

Specialization\*

Phone\*

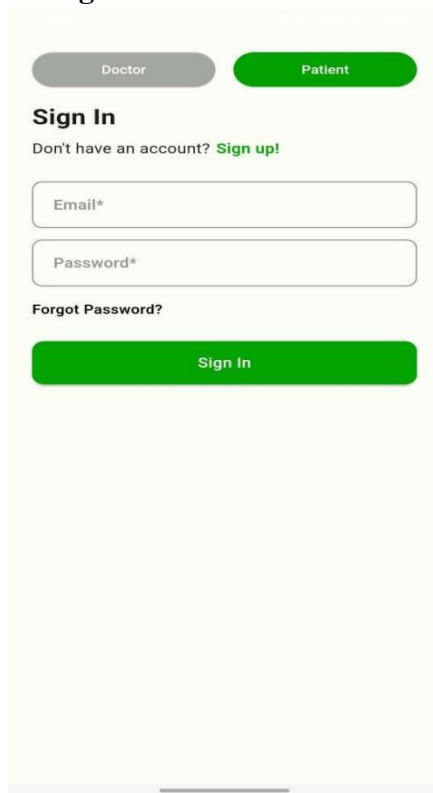
Password\*

Confirm Password\*

Description\*

**Sign Up**

### 3.2 Sign In



Doctor Patient

**Sign In**

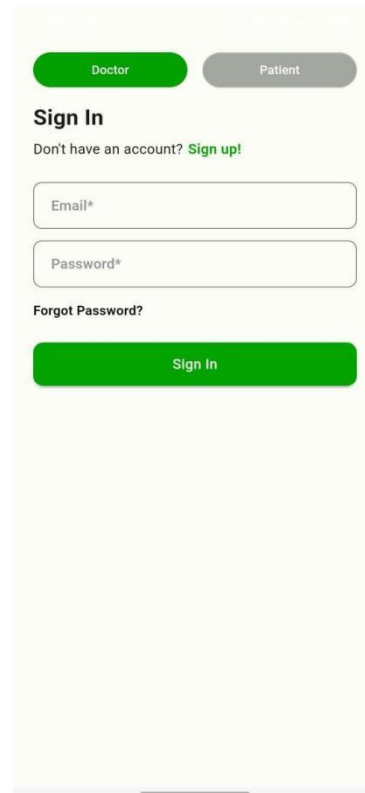
Don't have an account? [Sign up!](#)

Email\*

Password\*

Forgot Password?

Sign In



Doctor Patient

**Sign In**

Don't have an account? [Sign up!](#)

Email\*

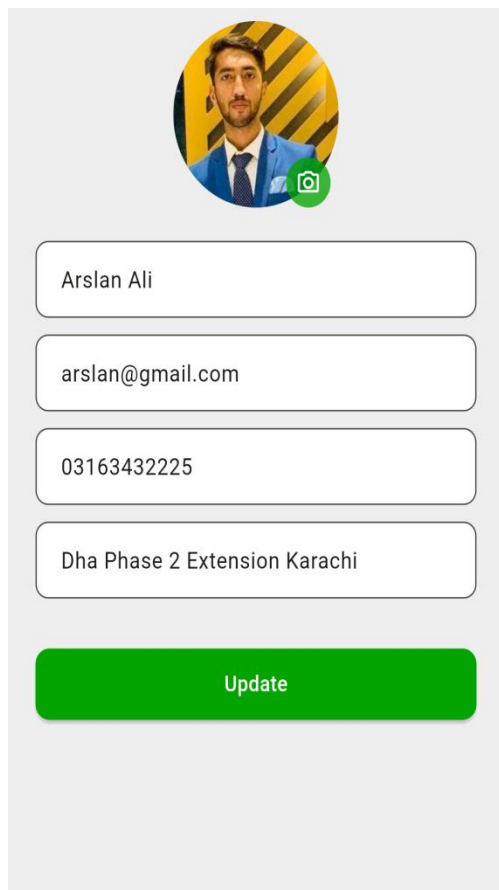
Password\*

Forgot Password?

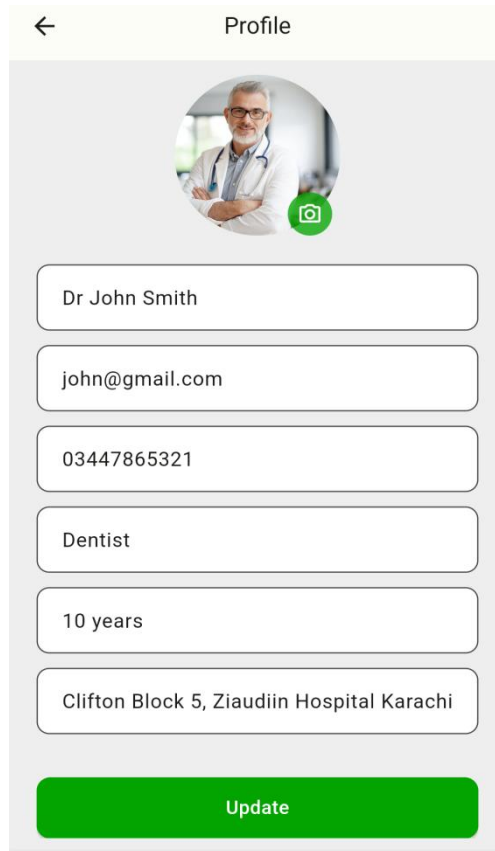
Sign In

The user uses their email and password to successfully log in to the system. After the email and password are verified the user is granted access into the system. But in-case of doctor admin approval required after approval he can login to app. Once logged into the system the user can access all available features.

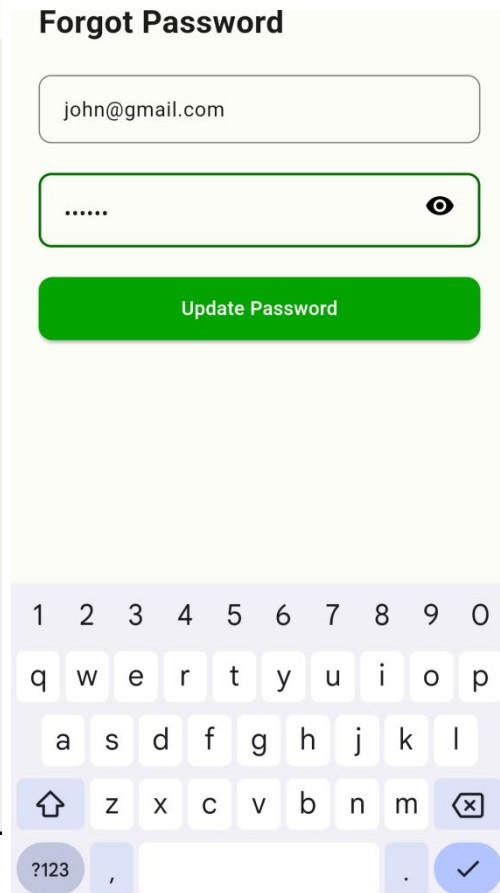
### 3.3 Profile Modification



A user profile modification form for a user named Arslan Ali. At the top is a circular profile picture of a man in a blue suit with a green camera icon. Below the picture are four text input fields: 'Arslan Ali', 'arslan@gmail.com', '03163432225', and 'Dha Phase 2 Extension Karachi'. At the bottom is a green 'Update' button.



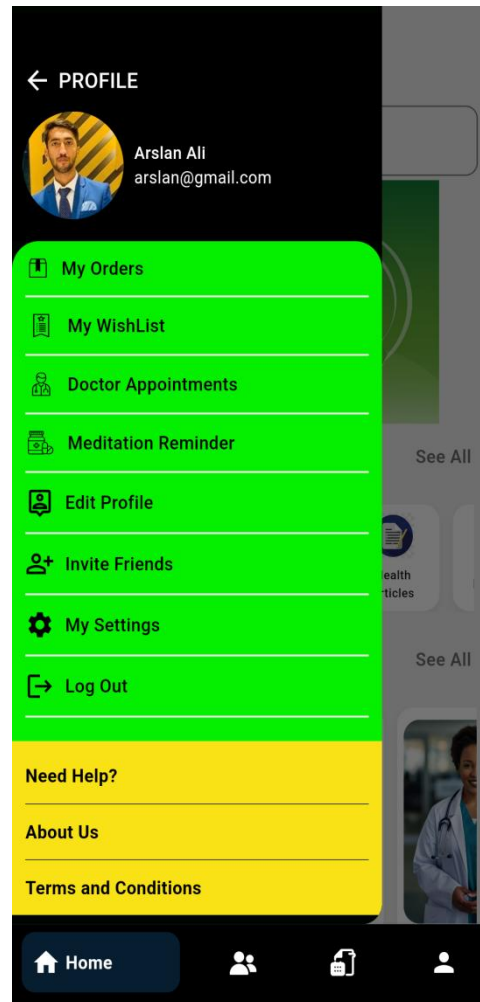
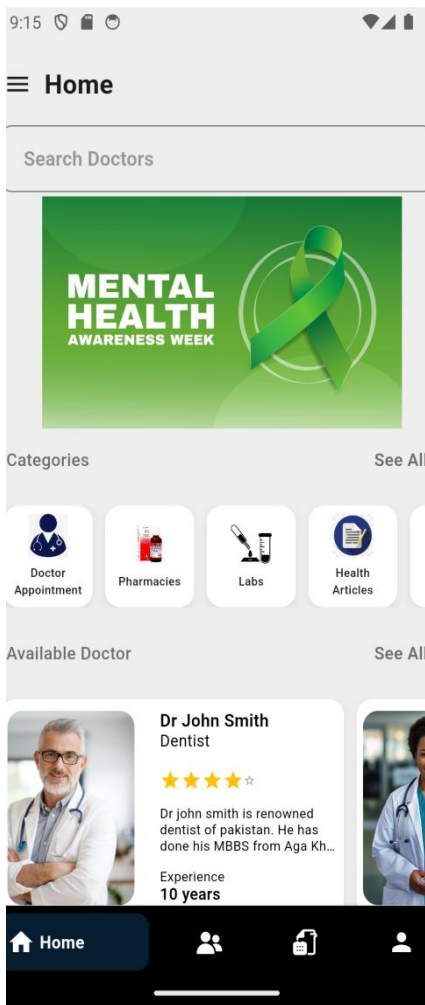
A user profile modification form for a user named Dr John Smith. At the top is a circular profile picture of a doctor with a stethoscope and a green camera icon. Below the picture are six text input fields: 'Dr John Smith', 'john@gmail.com', '03447865321', 'Dentist', '10 years', and 'Clifton Block 5, Ziauddin Hospital Karachi'. At the bottom is a green 'Update' button.



A 'Forgot Password' form. It has a title 'Forgot Password' at the top. Below it are two text input fields: the first contains 'john@gmail.com' and the second contains '.....' with an eye icon for password visibility. Below the fields is a green 'Update Password' button. At the bottom is a virtual keyboard with letters, numbers, and symbols.

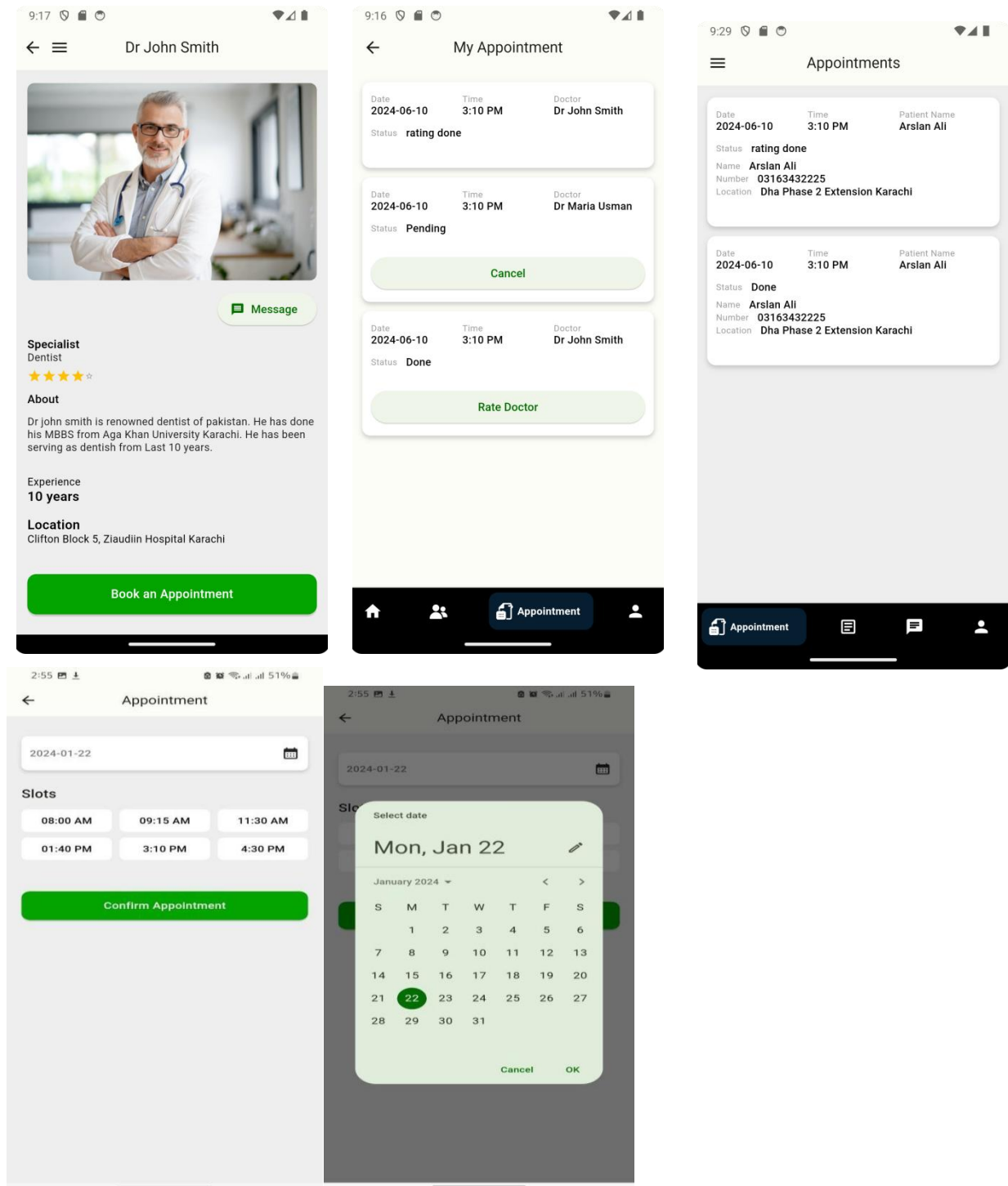
User can update their profile details by redirecting to the profile screen available on the top right side of the screen. All the user details are available on this page and can be edited. In case the user wants to change their name, password or mobile number. Clicking on the update profile will make changes to the database.

### 3.4 User dashboard



After successfully login the users are redirected to their personalized dashboard. This page contains a list of the all the features accessible to each panel. Also on side there is drawer which also contains history of orders, appointments, reminders, alongside other features such as logout, settings. Also user can navigate to different screens via bottom navigation bar, also they can search doctors.

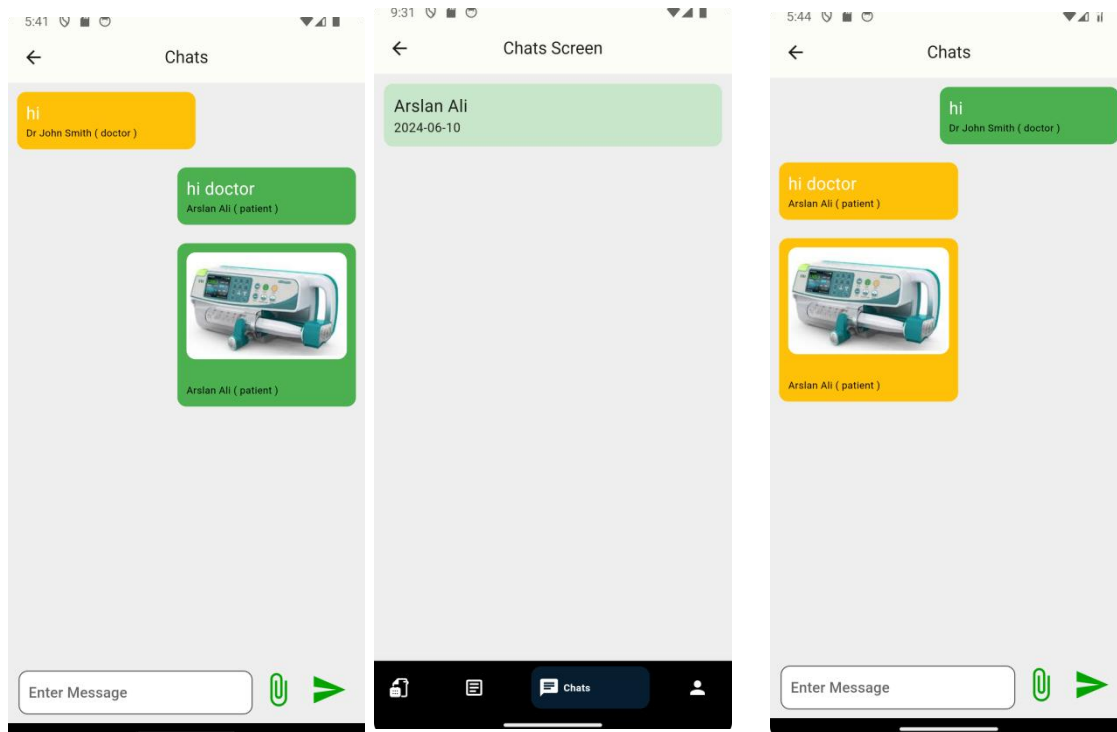
### 3.5 Doctor appointment



User can book doctor appointments by selecting given time slots, date. After booking appointment user can see appointment history and can also cancel appointment. In doctor panel doctor can see user details also can cancel and change appointment time. After appointment completion user can rate doctor.

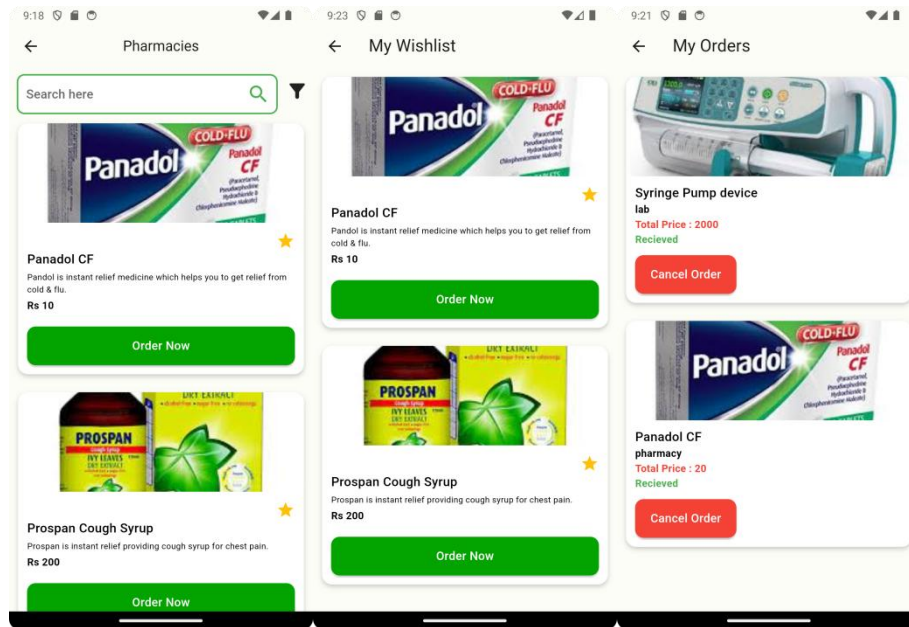


### 3.6 Chat box



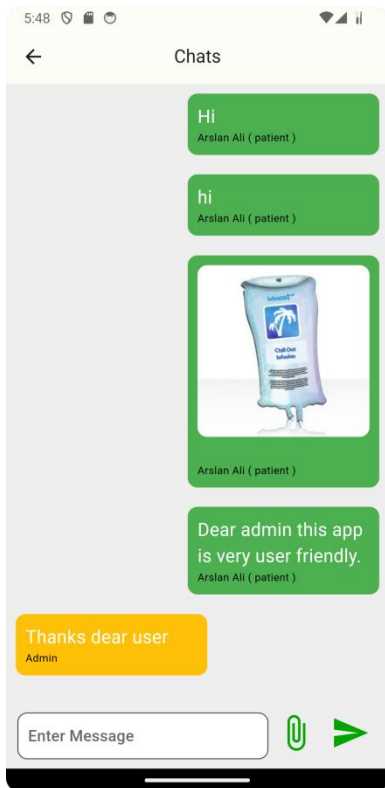
Both user and doctor can communicate with each other via chat box. The can exchange messages, pics and documents.

### 3.7 Order Medicines and Lab Products



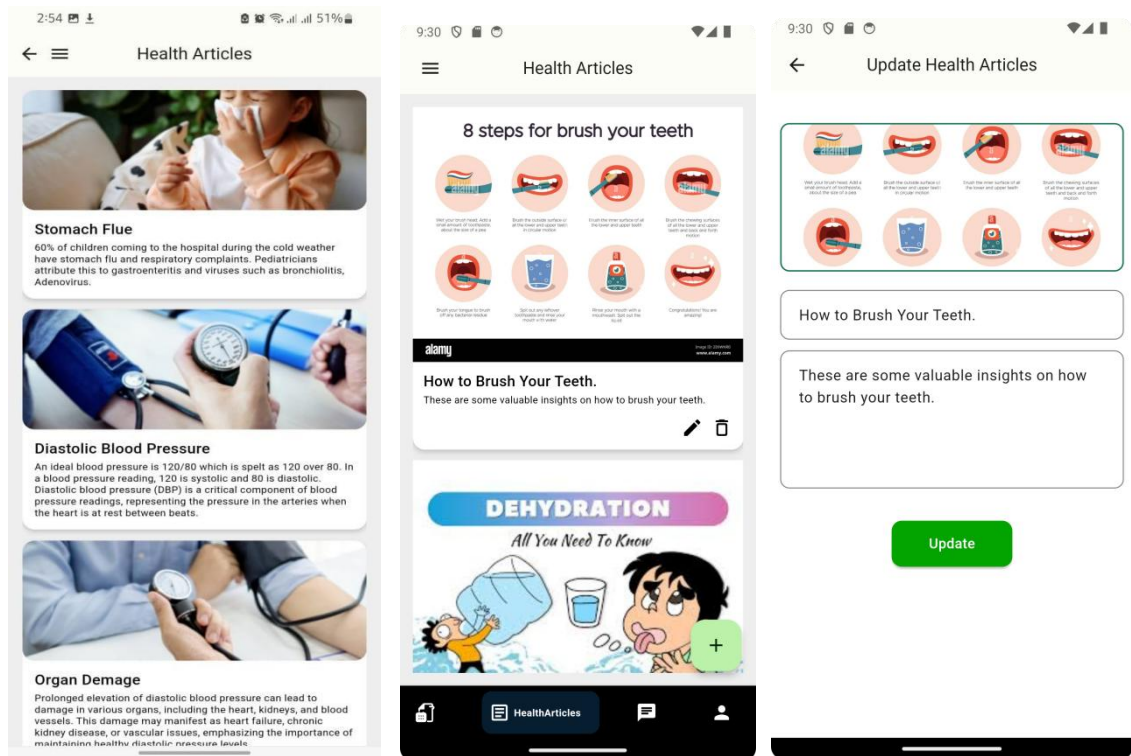
User can order medicines and lab products, Also can add their favourite products in wish list. They will be also able to cancel their order.

### 3.8 Feedback



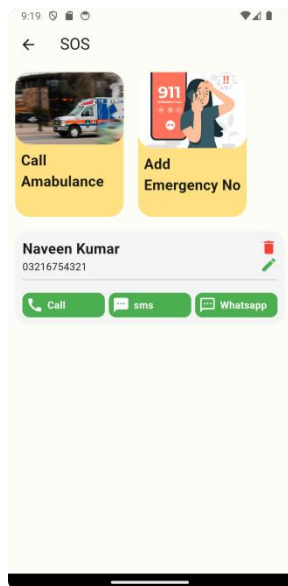
Via feedback feature user can give feedback such as suggestion, complaints or ask any query by directly chatting with Admin. And both admin and user can interact with each other.

### 3.9 Health Articles



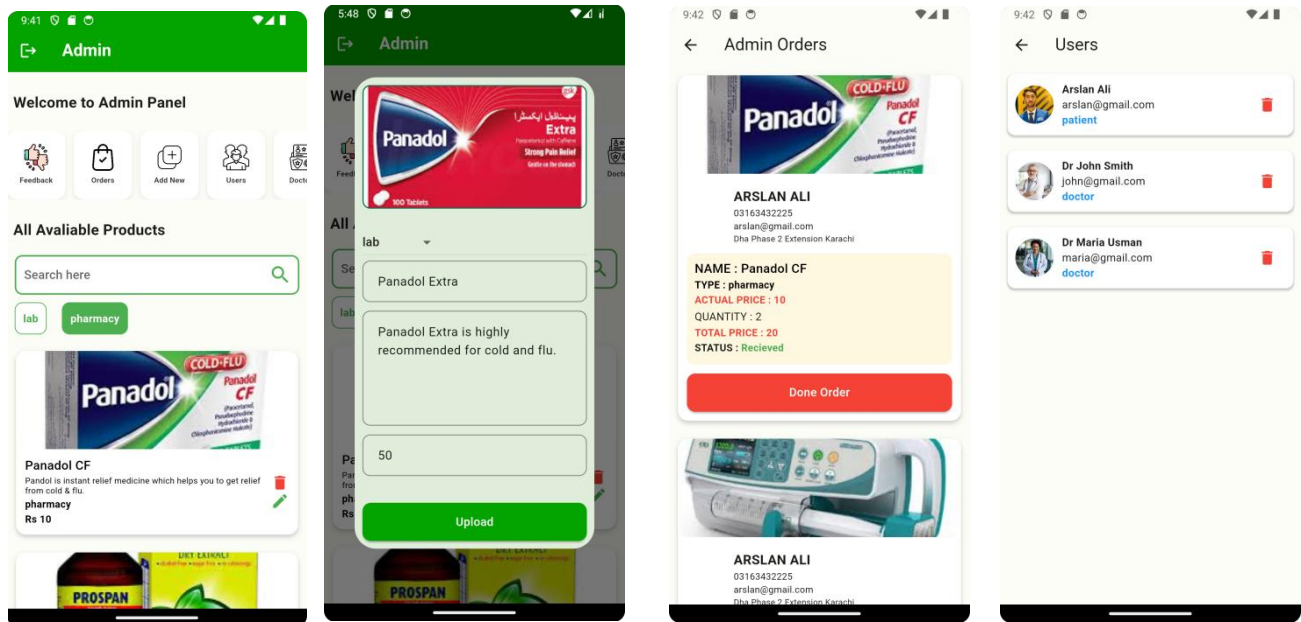
User (patient) can see health articles only and view who posted them. But doctor can add, update and delete health articles.

### 3.10 SOS Feature



User can call ambulance, add multiple emergency contacts also can call them and share their current location with emergency contact via sms or whatsapp.

### 3.11 Admin Panel



To sign in Admin Panel, Admin will add his credentials such as email and password. Through Admin panel admin can Add products, update and delete them, view order details, view users of App can remove them from app, also admin can chat with users via feedback feature alongside that admin can approve or disapprove doctor request to register on app through admin panel

# Project Log Meeting Sheet



SHAHEED ZULFIKAR ALI BHUTTO INSTITUTE OF SCIENCE & TECHNOLOGY KARACHI CAMPUS

## Form IV: Student Log Form

Title: ILaaS (Health Application)



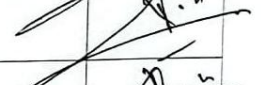






Supervisor: Sheikh Usama Khalid Batch/Sec: 2020/8-A Group #: \_\_\_\_\_

Reg. # (Group members): Naveen Iqbal (2012127), Arslan Ali (2012110)

Sr.	Task Assigned	Due	Task Completed (S)	Date (S)/Sign.
1	Admin Panel Integration	16-2-2024	Done	
2	Pharma Product U.I	23-2-2024	Complete	
3	Pharma Product Backend implementation	1-3-2024	Done	
4	Backend Cluster update	8-3-2024	Done	
5	Documentation (improved)	17-3-2024	Complete	
6	Software Testing	22-3-2024	Done	



SHAHEED ZULFIKAR ALI BHUTTO INSTITUTE OF SCIENCE &  
TECHNOLOGY KARACHI CAMPUS

7	user purchase log implementation	29-3-2024	Done	
8	chat box integration	5-4-2024	Done	
9	Appointment Reminder working	12-4-2024	Done	
10	Backend updation	19-4-2024	Done	
11	SOS feature Integration	26-4-2024	Done	
12	User-Interface update	4-5-2024	Done	
13	Software Testing	14-5-2024	complete	
14	documentation updation	20-5-2024	complete	
15	Project Report & Documentation compitation	26-5-2024	complete	

Supervisor's Authentication (Completed report):

FYP Coordinator Authentication:

Dated: 10/6/24

Dated: