SMART HEALTH CONSULTING APP

Group Members

Hudabia Shafique (01-131182-014)

Mahnoor Abdul Razzaq (01-131182-041)

Supervisor: Ma'am Rafia Hassan

A Final Year Project submitted to the Department of Software Engineering, Faculty of Engineering Sciences, Bahria University, Islamabad in the partial fulfillment for the award of degree in Bachelor of Software Engineering.

July 2022

THESIS COMPLETION CERTIFICATE

Student Name: _____Hudabia Shafique____ Enrolment No: __01-131182-014__

Student Name: _____Mahnoor Abdul Razzaq_____ Enrolment No: __01-131182-041_

Program of Study: Bachelor of Software Engineering

Project Title: _____Smart Health Consulting App_____

It is to certify that the above students project has been completed to my satisfaction and my belief, it's that standard is appropriate for submission for evaluation. I have also conducted a plagiarism test of this thesis using HEC prescribed software and found a similarity index at _____ that is within the permissible limit set by the HEC. I have also found the thesis in a format recognized by the department.

Supervisor's Signature:

Date: _____ Name: _____Rafia Hassan_____

CERTIFICATE OF ORIGINALITY

This is certify that the intellectual contents of the project _____Smart Health Consulting App_____

_____

are the product of my/our own work except, as cited properly and accurately in the acknowledgements and references, the material taken from such sources as research journals, books, internet, etc. solely to support, elaborate, compare, extend and/or implement the earlier work. Further, this work has not been submitted by me/us previously for any degree, nor it shall be submitted by me/us in the future for obtaining any degree from this University, or any other university or institution. The incorrectness of this information, if proved at any stage, shall authorities the University to cancel my/our degree.

Name of the Student: _____Hudabia Shafique_____

Signature: _____ Date: _____

Name of the Student: _____Mahnoor Abdul Razzaq___

Signature: _____ Date: _____

Abstract

Smart Health Consulting App is an online cross-platform android and iOS application. Consulting a doctor is an obvious thing in our day-to-day life, but the availability of the doctor during the time of our requirement is unpredictable. The main motivation for creating this project was the Covid-19 pandemic. Due to the pandemic situation, the lockdown was imposed. Hospital opds were closed for general check-ups. People were unable to get guidance about their health. There was a huge risk of going to the hospitals because the death toll was increasing day by day. Another purpose of this system is to reduce the problems that are faced during the manual health consultation system. In the Smart Health Consulting app, every user can register in the system as a doctor or a patient. In our app, there are two options that are free consultation and paid consultation. If the patient has selected paid consultation, he must pay before the consultation. If the patient has selected free consultation, he can post his problem and any available doctor can answer their problems willingly. To have better communication between the doctor and patient, we have provided the facility of chat, audio, and video calls.

Chapter 1 Introduction of "Smart Health Consulting App" mention the project brief introduction, tools, methodologies, and literature review of the project. Chapter 2 Background Study/Literature Review explains the project problem statement, current system, similar work, deliverables, and Development requirements. Chapter 3 System Requirements demarcate detailed use case diagrams with use case descriptions of all modules, functional and Non-functional requirements of the system. Chapter 4 System Design illustrates the whole system through the help of a System Architecture Diagram, Data Representation Diagram, Process/Flow Representation Diagram, and Design Model. Chapter 5 System Implementation describes the working of the system with the user interface. Chapter 6 System Testing and Evaluation 2 covers a detailed test case with test case methodology. Chapter 7 Conclusion summarizes the whole system.

Dedication

First, we dedicate this project to Almighty God, our Creator, our strong pillar, our source of inspiration, wisdom, knowledge and understanding. He has been the source of our strength throughout this FYP project, and it is only on his wings that we have soared. Second, we would like to dedicate our project work to our parents and teachers. A special feeling of gratitude goes to our loving parents and teachers, whose words of encouragement and striving for perseverance ring in our ears. We also dedicate this great project to our friends who supported us throughout the process. We faced obstacles on the way towards the completion of a project, but our parent's and teachers' encouragement and prayer never let us lose hope we kept working hard day and night and eventually we achieved the goal.

To our parents for their love and support

Acknowledgments

All praise goes to Almighty Allah, who bestowed upon us a minute portion of His infinite knowledge, allowing us to complete this complicated task.

We are extremely grateful to our project supervisor, "Ma'am Rafia Hassan." This project would not have been completed without their supervision, advice, and invaluable guidance. We are grateful to them for their support and encouragement throughout this project.

We are also grateful to our parents and family for always being a source of encouragement and instilling in us the values of honesty and hard work.

Table of Contents

List of Figures

LIST OF TABLES

Chapter # 1

Introduction

Chapter 1

Introduction

This chapter provides the introduction of the Smart Health Consulting App. This document will highlight all the functionalities and requirements of the project. SHC is an android and iOS mobile application that is created to get and provide medical consultations. Our product is the Smart Health Consulting app, which provides users with real-time health advice via an intelligent health care application. Our app is useful for people to get a consultation from home without leaving their comfort zone.

1.1. Motivation

The main motivation for creating this project was Covid 19 pandemic. Due to the pandemic situation, a lockdown was imposed. Hospital opds were closed for general check-ups. People were unable to get guidance about their health. There was a huge risk of going to the hospitals because the death toll was increasing day by day.

This was the main reason we created a Smart Health Consulting app so that the patient can consult with the doctor and get guidance regarding his/her health concerns.

1.2. Problem statement

During the Covid situation, the death toll was rising every day, there was a huge risk to go to hospitals for check-ups or some consultation. When covid cases were at their peak most of the hospitals had closed their opds. It may have happened numerous times that you or your relatives required quick medical assistance, but doctors are unavailable due to a variety of reasons. Our project is a Smart Health Consulting application that has the potential to change the way people access healthcare services, providing them with the opportunity to get the guidance and services they need without having to go through the traditional healthcare system.

1.3. Objectives

The primary goals in developing this system were:

- To provide a platform through which patients who require doctor assistance at home can consult doctors, send reports, chat with doctors, and discuss their issues and get appropriate remedies.

- That provides direct access to the specialists of your own choice.
- That is cost-effective and time-saving.
- That ensures the safety of the patients (w.r.t COVID-19).

1.4. Main Contributions

The uniqueness of the smart health consulting app is that we are providing satisfaction to our users by giving the options of both free and paid consultations. In a free consultation, a patient can post his problem related to health, and any available doctor can then answer his/her problem. The advantage of this feature is that if a person wants a consultation about medication or anything else related to health and he couldn't afford to pay he can consult a doctor without a fee. But in paid consultation, the patient will first select the specialist doctors. By looking at the doctor's profile patients can select a specialist of their own choice and they can book an appointment with the doctor after payment with the online transaction. If the doctor has confirmed the appointment, the patient will get a consultation. Otherwise, if the doctor has rejected the consultation due to any reason the fees will be refunded in the patient's wallet.

Flutter is a new technology that was released four years later December 4, 2018. Before flutter developers had to create separate codes for android and iOS applications, but we are creating our app using the latest technology flutter that is platform-independent. The major perspective of using flutter is learning new technology.

1.5. Report organization

The brief structure of our report is as follows:

Chapter 1

In chapter 1 we give a brief introduction of our project. We discussed in detail the objectives of the Smart Health Consulting app, the motivation behind the work, the main points of the project, the problem statement, and our contribution.

Chapter 2

In chapter 2 we provide the detailed knowledge of the literature review. It explores the preceding work we did in this field. It also gives an overview of the previous and the present work together,

provides a comparison between them, and also discusses the shortcomings and constraints of the previous work and the new contributions that we bring to the project.

Chapter 3

In this chapter, we concentrate on the system's functional as well as non-functional requirements. Different use case diagrams and use cases are presented to display the functional and non-functional requirements of the application.

Chapter 4

In this chapter, we included all the design patterns and the layouts that were used to build the android and iOS applications. Different kinds of diagrams from class diagrams to activity diagrams are there to present the design of the application.

Chapter 5

In this chapter, we provide a detailed review of the languages and algorithms on which the Smart Health Consulting android and iOS applications have been implemented.

Chapter 6

In this chapter, we provide all the detailed information about the phases of testing that this project has been through. There was a total of three phases of testing i.e., Unit testing, integration testing, and system and acceptance testing.

Chapter 7

In this chapter 7, we give a brief conclusion of the whole document that explains the work that has been done, the shortcomings, and the room for improvement in the application for the future.

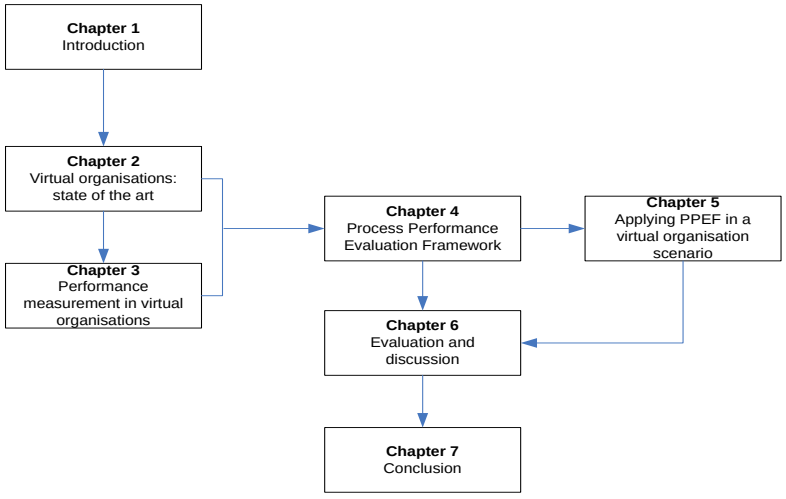

Figure 1: Thesis Organization

Chapter # 2

Background study/Literature Review

Chapter 2

Background Study/Literature Review

This project is related to consulting a doctor. With the advancement of mobile technology, the world is shrinking. As the number of users grows on a daily basis, so do the facilities. Beginning with simple regular telephones that were once used only for making phone calls, mobile has changed and become a part of our lives. As technology is changing extremely fast so is the requirement of users.

Consulting a doctor is an obvious thing in our daily lives, but the availability of the doctor at the time of our need is unpredictable. Under the manual Health Consultation System, you must first wait in line to make an appointment with the doctors and then wait for your time to meet with them and discuss your health problems.

There was another scenario in which covid cases were at a peak most of the hospitals had closed their opds It may have happened numerous times that you or your relatives required quick medical assistance, but doctors are unavailable due to a variety of reasons. So we have created, a Smart Health Consulting application that has the potential to change the way people access healthcare services, providing them with the opportunity to get the guidance and services they need without going through the traditional healthcare system.

2.1 Analysis of the existing work- Where is it strong and where is it weak?

The main limits will be on software as we are using API. On the strong side, it can be used on any smartphone (both android and iOS) and can be used by any type of user because in our application we have provided a user-friendly interface and facility to select the language (Urdu or English) in which the user wants to continue.

2.2 Other similar work:

By reviewing different available medical-related apps on play store like MediaQ and SmartHealth, we explored that there was no proper free consultation module and the only available option for patients was Paid Consultation. So, the patient who is currently signed in has only the option to do

paid consultation which mainly depends on the doctor's fee. We discovered it as a huge gap for patients who cannot afford to get paid consultations. Our application will give a separate free consultation module so there is an open choice for a user to select a consultation type willingly.
The interface of the previous application is quite confusing and will be replaced by a much more user-friendly mobile application that will be easier to use by the users.
One of the major drawbacks that the previous applications was that they did not provided the user with a dark theme option which is a key feature in enhancing user interactivity such as in Dawaai, MediaQ, Marham, and SmartHealth apps. As Dark mode apps can prolong the battery life of your smartphone and is better for reducing eye strain in low light conditions. Many previous applications such as MediaQ, Dawaai, and SmartHealth apps did not support the local language(Urdu).

2.3 Literature review:

- Ravi Aavula et al. made an effortless online application that allowed users to get immediate assistance with their health issues via a smart health care application online. This application contained various symptoms and diseases linked with them. The users could get a consultation with their desired doctor through the app. Enabled the users' to get instant reports on their related health concerns through a health care application online. The system was designed to provide Quality Health Care to everyone. This system was a comprehensive platform for mobile healthcare [1].

- Ibrahim Al-Mahdi et al. made an Online Medical Consultations (OMC) that were internet-based consultations between patients and doctors. This consultation was aimed at diagnosis or guidance about health issues. We studied that this research paper distinguished online medical consultations into three categories that were telephone only or radio-based consultations, shop around for medical consultations, and the third was direct patient-doctor consultations that was not included doctor-doctor consultations or consultations for health education and other reasons [2].

- Kshitij Kuber et al. was aimed to develop a mobile application to maintain health by knowing the related symptoms. The application was aimed to allow the user to log in and

signup then the user went to the search module where he/she was able to search for some issues if those issues were contained in the database, then it shows the prescription otherwise user had to consult a doctor by clicking call option. Mainly this application contained a doctor as an admin and, maintains the server. If there was no affiliated result, then the user was able to send a query to the doctor, and the doctor was able to view all the queries of the patients [3].

- Asst.Prof N.Vaswani et al. created a Health Care Application that was an Android application that helped the user to facilitates patients. Check the symptoms of the patient through a symptom analyzer which processed the entered patient's symptoms and gave a recommendation of medicines or doctors. In the case of an emergency, the user had the option to use the SOS facility which should be further directed to the ambulance authority in charge. The main objective was to take care of the patient [4].

- Aleksandar.K et al. describe that the process of healthcare was changed from a visit to the doctor's office to an online consultation. Digital technology had made it possible for digital devices to monitor the health of patients in their homes. The use of digital devices for health monitoring was not a new concept. Several health systems used digital technology to monitor patients in hospitals.

  The concept of remote health monitoring had evolved through the years and had been expanded to home health care technologies. Remote health monitoring was referred to as the use of home health care technologies to assist healthcare givers, medical personnel, and physicians to reach out to patients without physical contact or the presence of patients at clinics or hospitals. The purpose of remote health monitoring was to keep patients healthy and provide them with the care they needed without having to visit a clinic or hospital. This way, patients were able to get the care they needed without leaving the comfort of their homes [5].

- Ajeet Pal Singh et al. described that the goal was to present an extensive survey on the current state of the art in online medical consultations. This was aimed to indicate the web-

based remote patient-specialist medicinal discussion. According to this survey, with an increase in technology, many people were moved to online web portals to get an online consultation. The Utilization of this technology was able to give many advantages to doctors as well as patients. The main aim of this article was to study the possible reasons behind the starting of a new era of consultation, which was online medical consultation. In literature, it examined features and themes evident and the scope of currently operating platforms providing online medical consultation [6].

- Minh P et al. described that Smart home environments had became a central feature of our lives, from turning on the lights to controlling our heating to being able to monitor our children when they were away from home. Technology had enabled these spaces to become more than just a place to sleep and stay warm, they had become a means of improving our lives and improving the way we live. One of the most advanced smart home environments was a Cloud-Based Smart Home Healthcare Environment or CoSHE for short.

  CoSHE had a smart home, a wearable unit, a private cloud, and a robot assistant. The smart home were able to collect physiological, motion, and audio signals from residents through non-invasive wearable sensors. The wearable unit was connected to the smart home and provided residents with remote access to their recorded data and the ability to control in-home devices and appliances [7].

Chapter # 3

System Requirements

Chapter 3

System Requirements

3.1 System Level Use Case Diagram

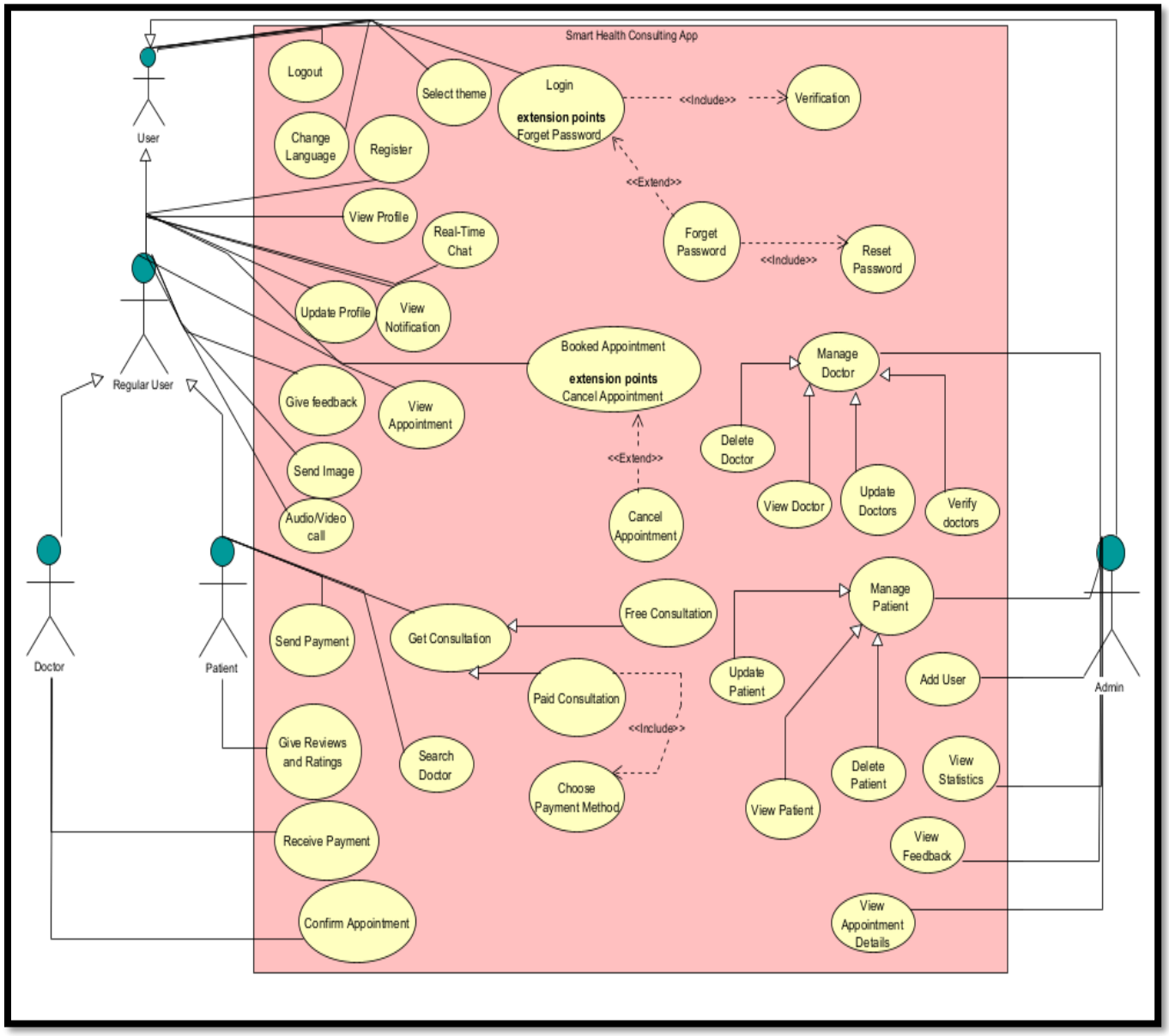

Figure 2: System-Level Use case Diagram

Explanation:

The above use-case diagram depicts the interaction of the Users with the Mobile application. In this, we have users who are generalized by admin and regular users. Regular users are further generalized by the patient and doctor. Users can change language (Urdu or English) and select themes depending on their feasibility. Regular users first must register (as a patient or doctor) to proceed further. If the user forgets their password, he/she can reset it. If the regular user is patient, then they can get both free and paid consultation depending on their choice. But before getting paid consultation patient has to give payment. Once the appointment is confirmed by the doctor, the patient and doctor can chat with each other or send images in the form of reports and can-do audio/video calls for better interaction. After getting an appointment patient can give a review and rating to the doctor. Regular users can view their profile and can update it if they want to update as well as they can view their appointments and notification. They can also give their feedback to the admin. On the other side, the admin can manage patients and doctors as well as verify doctors by looking into their profiles. Admin can add users, view appointments, statistics, and feedback as well.

3.2 Detailed Use cases:

3.2.1 User Management

3.2.1.1 Registration

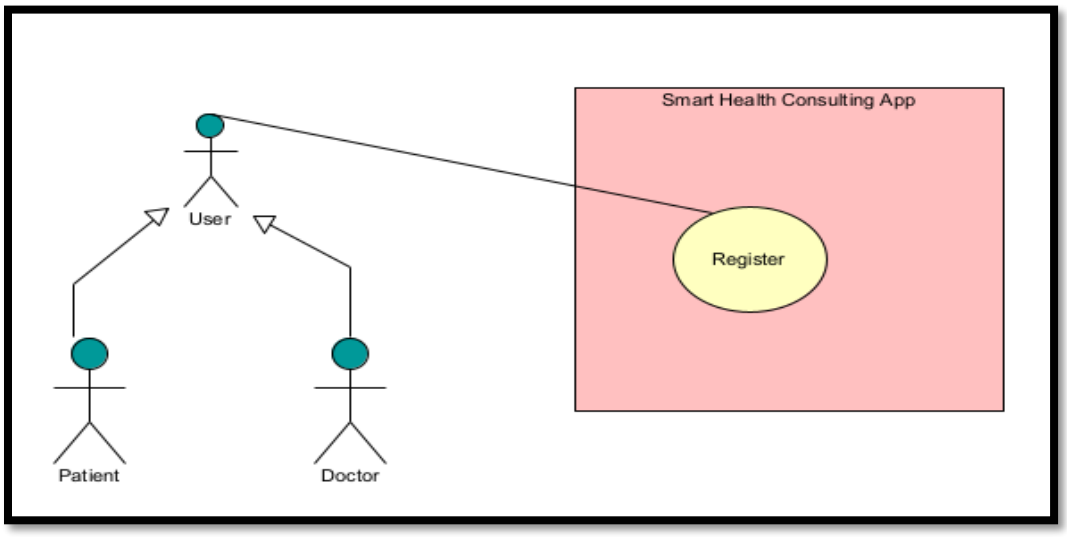

Figure 3: Use case Diagram of Registration

Table 1: Use Case Description of Registration

Use Case ID:	**UC1**	
Use Case Name:	**Registration**	
Actor(s):	User	
Pre-Conditions:	The user adds information to register.	
Priority:	High	
Basic Flow:	1. User Open Application. 2. Choose the language or skip that part (Default Language will be chosen). 3. Users can choose to register as a patient or doctor by sliding the toggle button. 4. The user enters their Email and Password. 5. The user checks the Terms and Conditions dialog box. 6. The user clicks on the register button. 7. The system verifies entered email and registers the user. System display message that "Your account has been registered successfully".	
Actor Actions		**System Response**
1	The user enters personal details in the relevant fields. The user clicks on the "Register" button.	System check user's provided credentials. If the user's credentials are valid, the user is authenticated, and the home screen is displayed.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	The user forgot to fill a field.	An error message will be prompted to fill all fields.
2	The user enters the email.	Email already exists. An error message will be prompted.
3	The user inputs a password of fewer than 7 characters.	An error message will be displayed password should be more than 7 characters.

4	The User forgot to enter the symbol.	An error message will be displayed that the password should have at least one letter, one symbol, and one character in the upper and lower case.

3.2.1.2 User Login

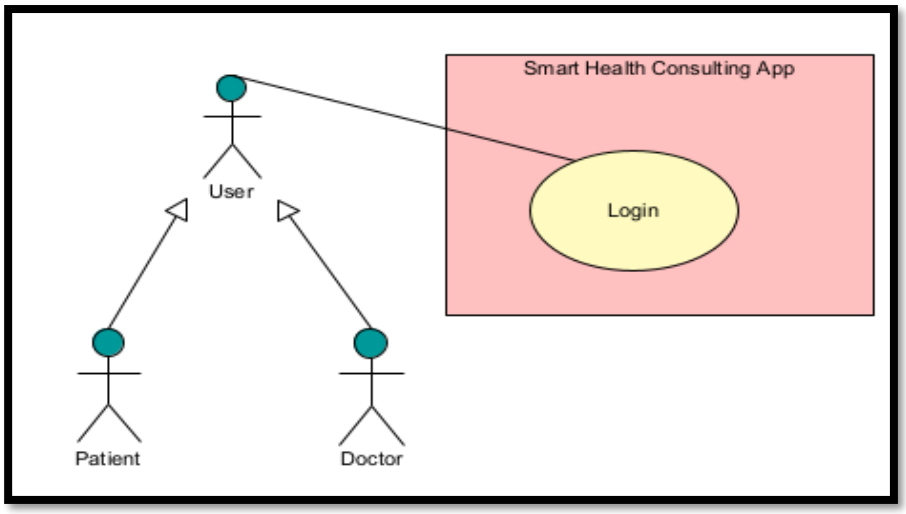

Figure 4: Use case Diagram of User Login

Table 2: Use Case Description of User Login

Use Case ID:	**UC2**
Use Case Name:	**Login**
Actor(s):	User
Pre-Conditions:	The User adds information to the login.
Priority:	High

Basic Flow:	1. Enter the email and password. 2. Click on the login button. System Display Message That “you have login Successfully”.	
Actor Actions		**System Response**
1	The user inputs an email and password.	System check user’s provided credentials. If the user’s credentials are valid, the user is authenticated, and the home screen is displayed.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	The user forgot to fill a field.	An error message will be displayed to fill all fields.
2	The user enters a password of fewer than 7 characters.	An error message will be displayed that the password should be more than 7 characters.
3	Users enter email.	An error message will be displayed Sign in failed.

3.2.1.3 Forgot Password

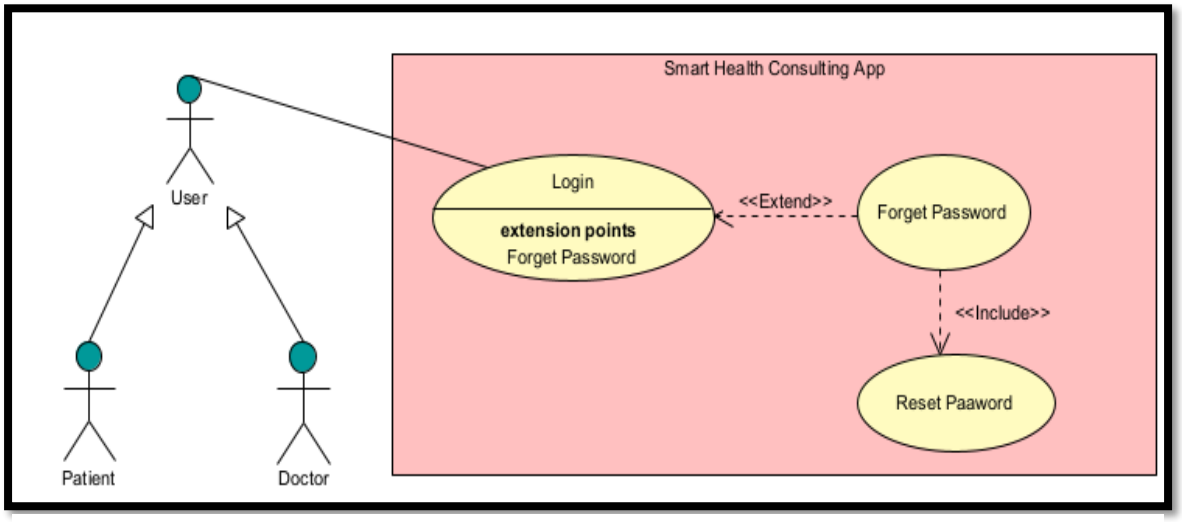

Figure 5: Use case Diagram of Forgot Password

Table 3: Use Case Description of Forgot Password

Use Case ID:	**UC3**
Use Case Name:	**Forgot Password**
Actor(s):	User, Google client, Google Admin
Pre-Conditions:	The mobile app is open, and the user is on the login page.
Priority:	High
Basic Flow:	1. Enters email and incorrect password. 2. Click Login. 3. The system shows an invalid username or password error. 4. The user clicks on forgot password button. 5. Forget the password page open in which users have to enter their email address and then click on the button. 6. User opens their email account in which he/she found a mail through which they can reset their password.

	7. The user opens that mail and changes the password and clicks on save. Password is Reset.

Actor Actions		**System Response**
1	The user enters the correct email and incorrect password.	Displays invalid username or password error message.
2	The user resets the password.	The Password is changed.
3	The user enters the correct email and correct password.	The Home screen is displayed.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	Users enter email.	An error message will be displayed that this account doesn't exist.

3.2.1.4 User Logout

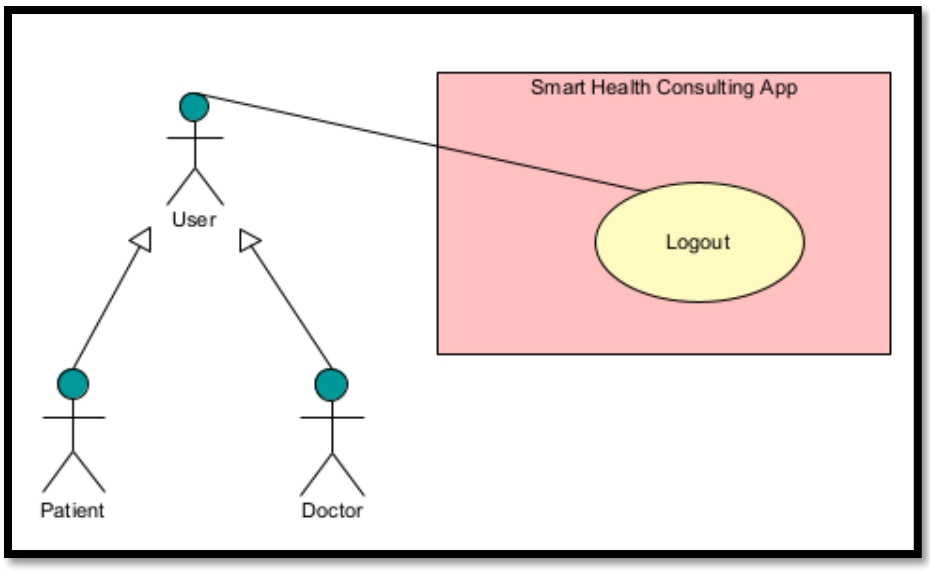

Figure 6: Use case Diagram of Logout

Table 4: Use Case Description of User Logout

Use Case ID:	**UC4**	
Use Case Name:	**Logout**	
Actor(s):	User	
Pre-Conditions:	1. The user is logged in. 2. The user no longer wants to be logged in.	
Priority:	Medium	
Basic Flow:	1. The user is done using the mobile application. 2. The user clicks on the logout button. 3. The system logs the user out.	
Actor Actions		**System Response**
1	Users click on the logout button.	The login screen is displayed.
Alternative Course of Action (if any)		
Actor Action		**System Response**
	NA	NA

3.2.1.5 Update Profile

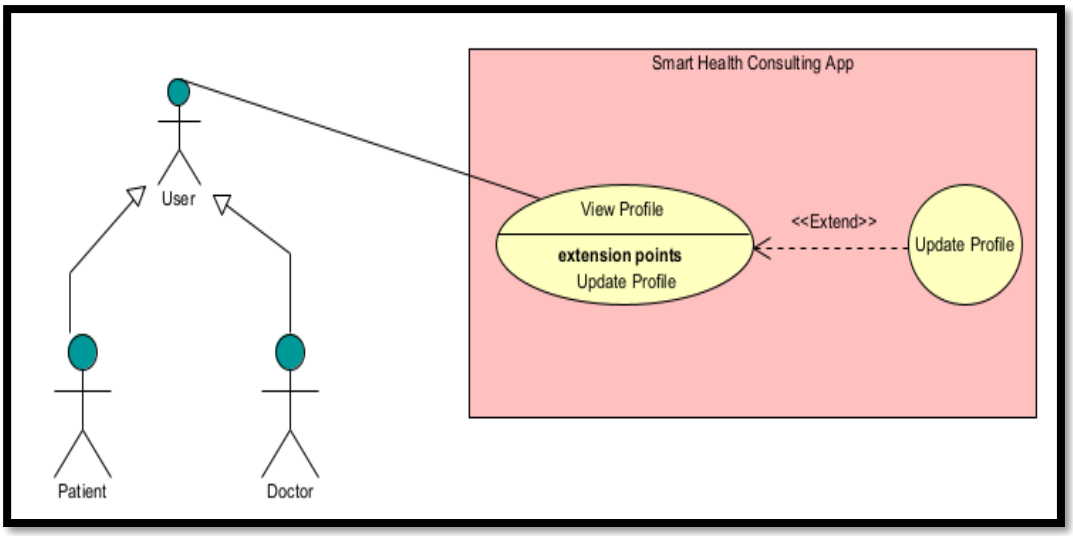

Figure 7: Use case Diagram of Update Profile

Table 5: Use Case Description of Update Profile

Use Case ID:	**UC5**
Use Case Name:	**Update Profile**
Actor(s):	User
Pre-Conditions:	The mobile app is open, and the user is on the Profile page.
Priority:	Medium
Basic Flow:	1. User selects update profile. 2. The app displays profile information. (Name, age, gender, date of birth, Image, etc.). 3. User updates profile, by clicking on the update button. 4. System display message that "Profile updated successfully".

Actor Actions		**System Response**
1	User updates Profile and clicks Ok.	The user profile is updated.

Alternative Course of Action (if any)		
Actor Action		**System Response**
	NA	NA

3.2.1.6 View Profile

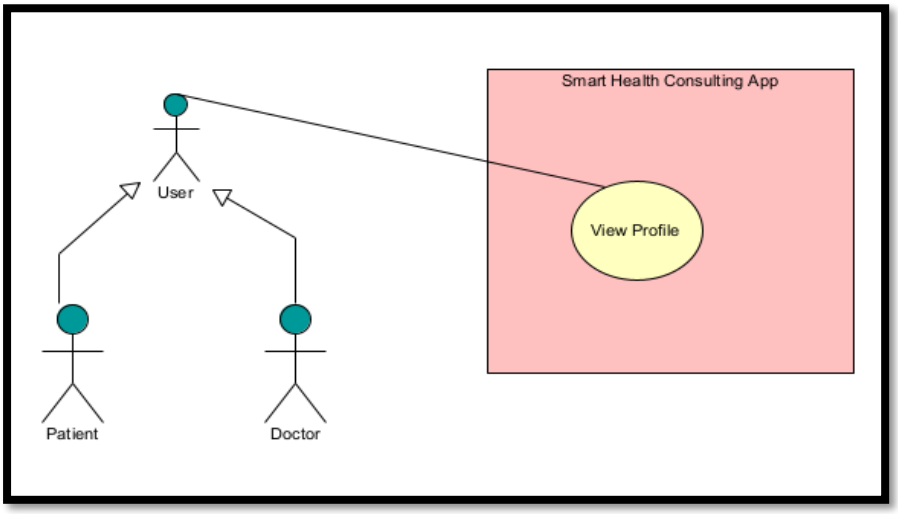

Figure 8: Use case Diagram of View Profile

Table 6: Use Case Description of View Profile

Use Case ID:	**UC6**
Use Case Name:	**View Profile**
Actor(s):	User
Pre-Conditions:	The mobile app is open, and the user is on the Home page.
Priority:	Medium
Basic Flow:	1. User selects Profile. 2. The app displays profile information. (Name, Age, Image. etc.) 3. User view profile information. 4. Press the Back button.

	5. The user is on the home page.	
Actor Actions		**System Response**
1	User selects profile and views information.	User Profile is displayed.
Alternative Course of Action (if any)		
Actor Action		**System Response**
	NA	NA

3.2.1.7 Give Feedback

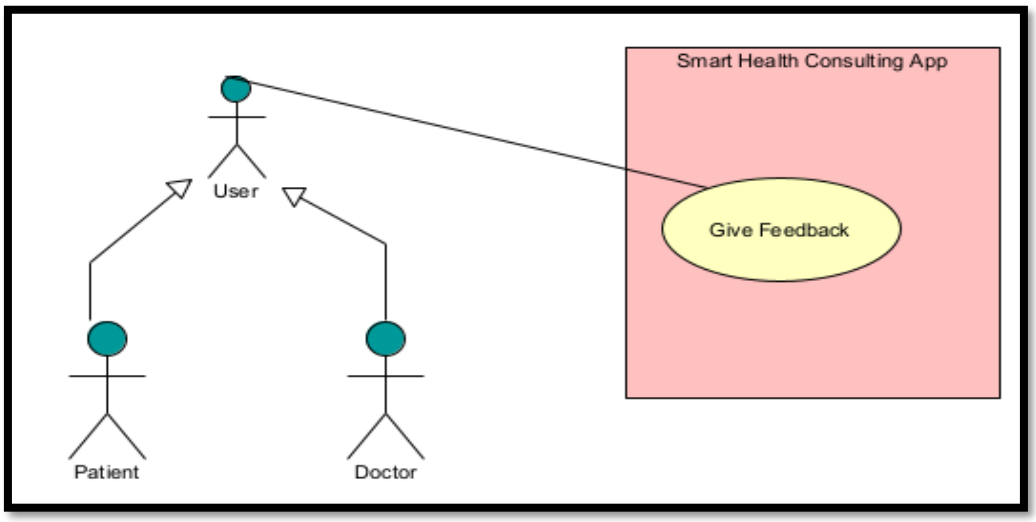

Figure 9: Use case Diagram of Give Feedback

Table 7: Use Case Description of Give Feedback

Use Case ID:	**UC7**
Use Case Name:	**Give Feedback**
Actor(s):	User

Pre-Conditions:	The mobile app is open, and the user is on the Home page.	
Priority:	High	
Basic Flow:	1. The user selects give feedback. 2. Users select a rating and enter feedback. 3. The user clicks on send review button. 4. The System display's the message that "Feedback sent successfully".	
Actor Actions		**System Response**
1	The user selects Feedback and enters feedback.	A message will be displayed "feedback is sent."
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	N\A	N\A

3.2.1.8 Cancel Appointment

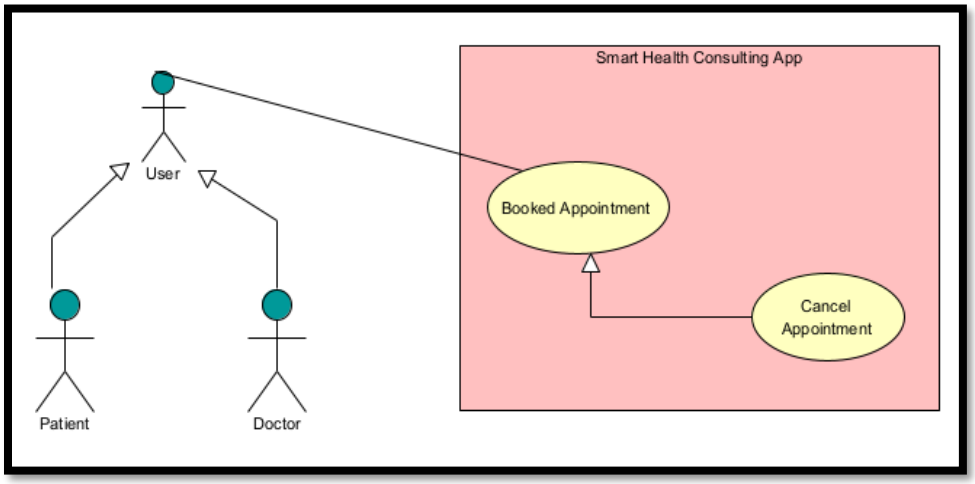

Figure 10: Use case Diagram of Cancel Appointment

Table 8: Use Case Description of Cancel Appointment

Use Case ID:	**UC8**
Use Case Name:	**Cancel Appointment**
Actor(s):	Doctor
Pre-Conditions:	1. The Doctor must log in to the app. 2. The Patient sends a consultation request to the doctor.
Priority:	High
Basic Flow:	1. The doctor receives the notification of a "New consultation request". 2. The doctor is on the home page. 3. The doctor selects the My appointment module. 4. The doctor selects the appointment request. 5. The doctor can view the patient's profile. 6. The doctor clicks on the Reject button and the appointment is canceled. 7. The system cancels the booked appointment and updates the system. The system displays the refunded payment in the patient's wallet.

Actor Actions		**System Response**
1	The doctor cancels the booked appointment.	A message will be displayed "Appointment is canceled successfully".
Alternative Course of Action (if any)		
Actor Action		**System Response**
	NA	NA

3.2.1.9. View MyAppointment

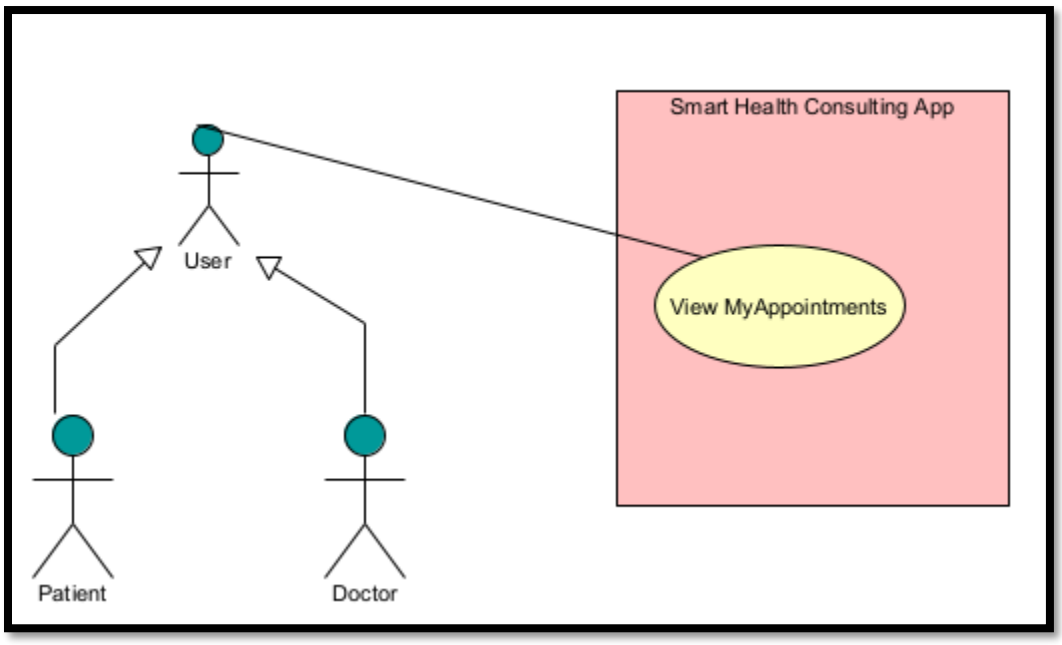

Figure 11: Use case Diagram of View MyAppointment

Table 9: Use Case Description of View MyAppointment

Use Case ID:	**UC9**	
Use Case Name:	**View My Appointment**	
Actor(s):	User	
Pre-Conditions:	User must log in to the app.	
Priority:	Medium	
Basic Flow:	1. User selects My Appointments. 2. The system shows the lists of appointments. (System displays the list of all appointments pending appointments, canceled appointments, and executed appointments)	
Actor Actions		**System Response**
1	User choose View 'My Appointment.'	The System displays an appointment list.

Alternative Course of Action (if any)	
Actor Action	**System Response**
NA	**NA**

3.2.1.10. View Notification

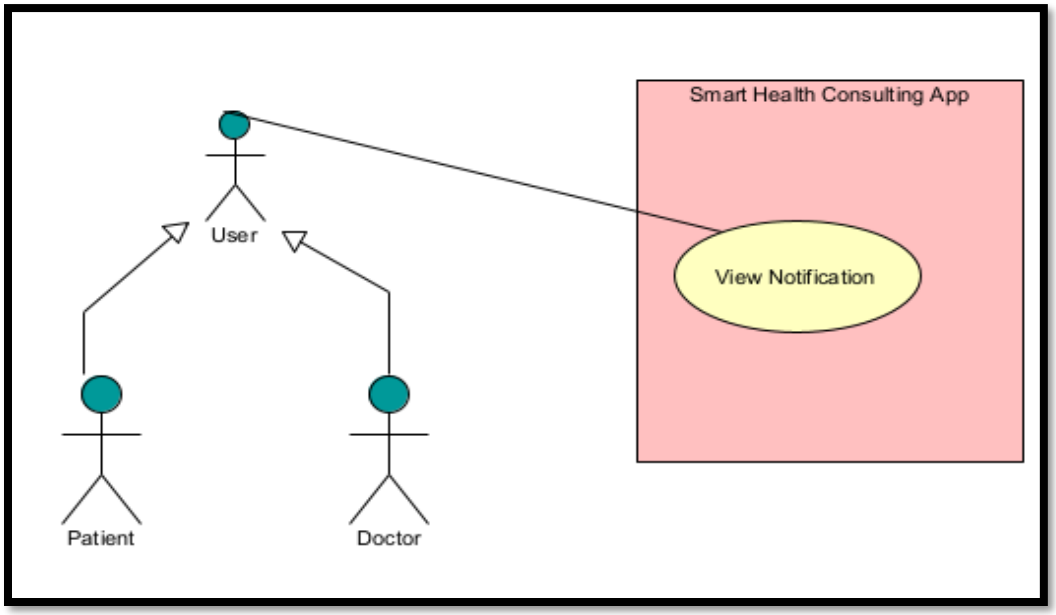

Figure 12: Use case Diagram of View Notification

Table 10: Use Case Description of View Notification

Use Case ID:	**UC10**
Use Case Name:	**View Notification**
Actor(s):	User
Pre-Conditions:	Users must log in to the app.
Priority:	Medium
Basic Flow:	1. The User is on the Homepage. 2. The user selects the notification. 3. The System shows a list of notifications.

Actor Actions		System Response
1	User chooses notification.	The System displays the lists of notifications.
Alternative Course of Action (if any)		
Actor Action		**System Response**
	NA	NA

3.2.1.11 Change Language

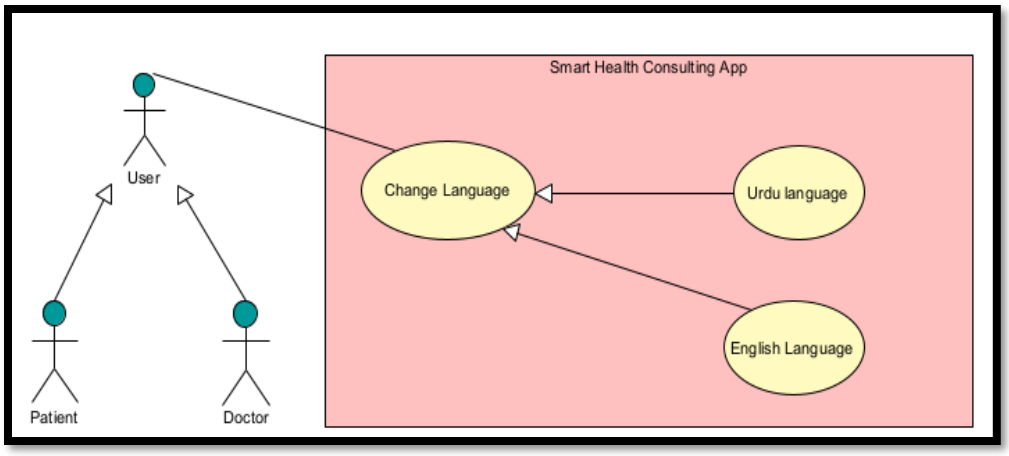

Figure 13: Use case Diagram of Change Language

Table 11: Use Case Description of Change Language

Use Case ID:	**UC11**
Use Case Name:	**Change Language**
Actor(s):	User
Pre-Conditions:	The App should be open.
Priority:	High
Basic Flow:	1. Users can click on the skip button to choose the default (English) Language or choose the Urdu language by selecting from the drop-down menu.

	2. Language is changed.	
Actor Actions		**System Response**
1	The user changes the language.	Language changed successfully.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	The user skips the language screen by clicking on the skip button.	English is selected as a default language.

3.2.1.12 Real-time chat

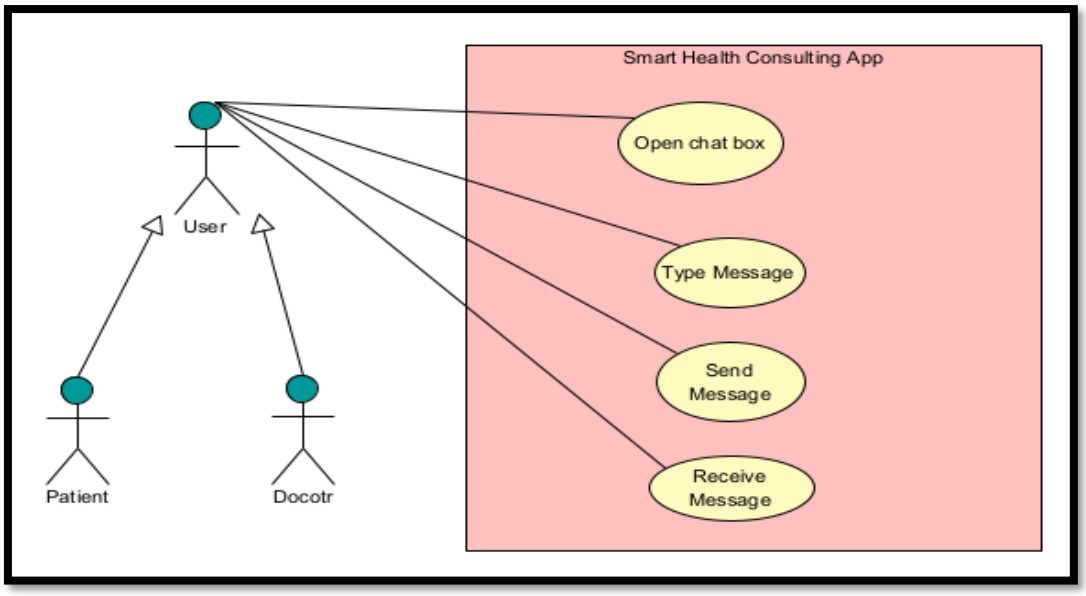

Figure 14: Use case Diagram of Real-time Chat

Table 12: Use Case Description of Send Message

Use Case ID:	**UC12**
Use Case Name:	Send message
Actor(s):	User
Pre-Conditions:	The user must log in to the app and the appointment is confirmed by the doctor.

Priority:	High	
Basic Flow:	1. User clicks on the Chat tab. 2. The user automatically shows in chat to which the appointment has been confirmed. 3. Click on that user to chat with them. Once the consultation time is reached, the patient and doctor can send messages to each other.	
Actor Actions		**System Response**
1	The user sends a message.	The message is delivered successfully.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	If consultation time is remaining.	The system displays" Your consultation is in a few minutes".

Table 13: Use Case Description of Receive Message

Use Case ID:	**UC13**	
Use Case Name:	Receive message	
Actor(s):	User	
Pre-Conditions:	Someone sends the message.	
Priority:	High	
Basic Flow:	1. User clicks on the Chat tab. 2. The user automatically shows in chat to which the appointment has been confirmed. 3. Click on that user to chat with them. Once the consultation time is reached, the patient and doctor can receive a message from each other.	
Actor Actions		**System Response**

1	The user receives a message.	The message is seen by the user.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	If consultation time is remaining.	The System displays" Your consultation is in a few minutes".

3.2.1.13 Send Image

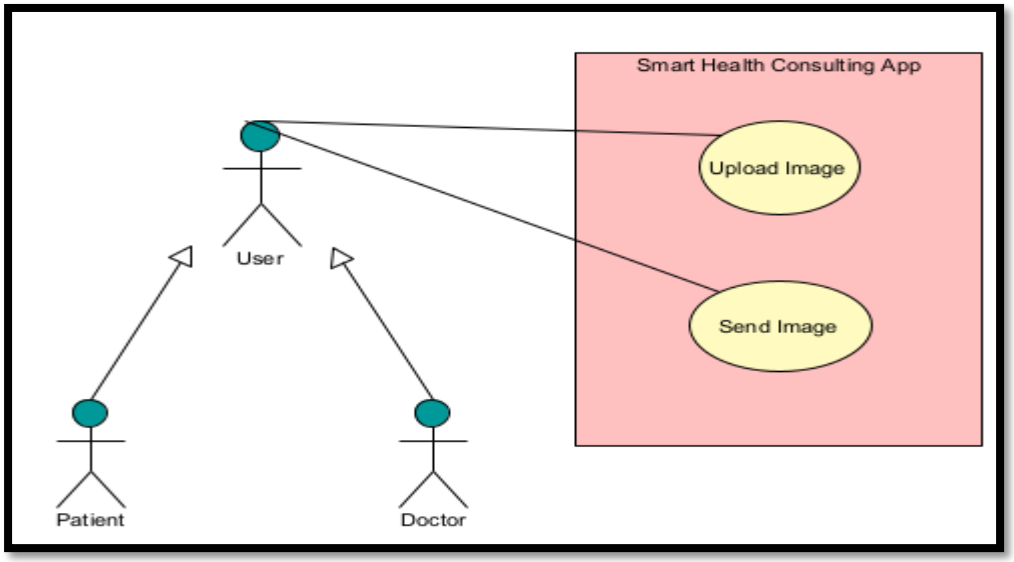

Figure 15: Use case Diagram of Send Image

Table 14: Use Case Description of Send Image

Use Case ID:	**UC14**
Use Case Name:	**Send Image**
Actor(s):	User
Pre-Conditions:	The user is logged in and consultation is confirmed by the doctor.
Priority:	High
Basic Flow:	1. User clicks on the Chat tab.

	2. The user automatically shows in chat to which the appointment has been confirmed. 3. Click on that user to chat with them. 4. Once the consultation time is reached, the patient and doctor can chat with each other. 5. The patient can send an image (reports, tests, etc.) to the doctor by clicking on the image icon.

Actor Actions		**System Response**
1	The user sends images.	Images sent successfully.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	If consultation time is remaining.	The System displays" Your consultation is in a few minutes".

3.2.1.14 Select theme

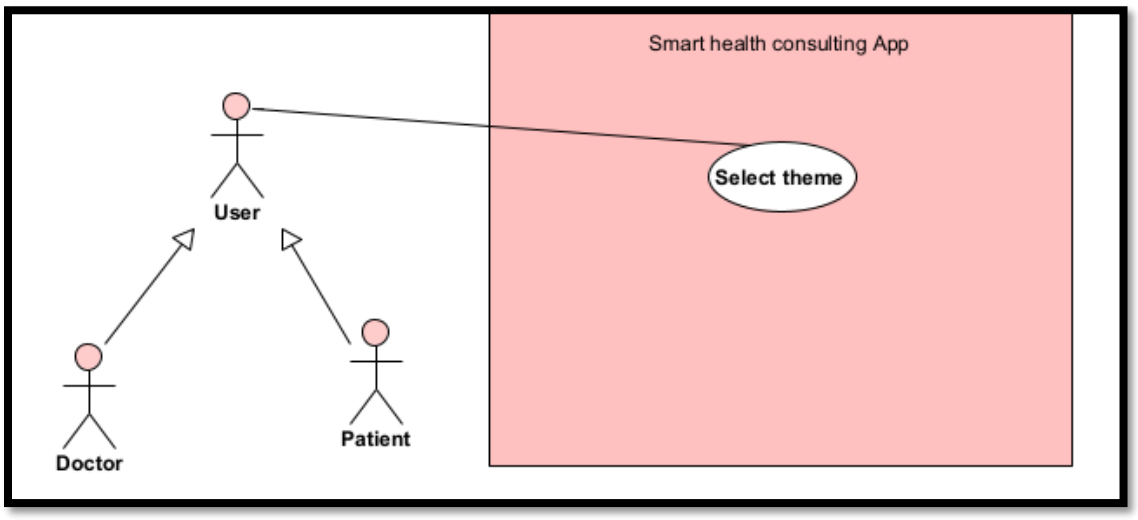

Figure 16: Use case Diagram of Select Theme

Table 15: Use Case Description of Select Theme

Use Case ID:	**UC15**	
Use Case Name:	**Select theme**	
Actor(s):	User	
Pre-Conditions:	The user is logged in.	
Priority:	High	
Basic Flow:	1. The user goes to the setting. 2. The user clicks on a theme. 3. User select theme (light or dark).	
Actor Actions		**System Response**
1	User selects theme (light or dark theme).	The theme is successfully changed.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	N/A	N/A

3.2.1.15 Select Audio/video call

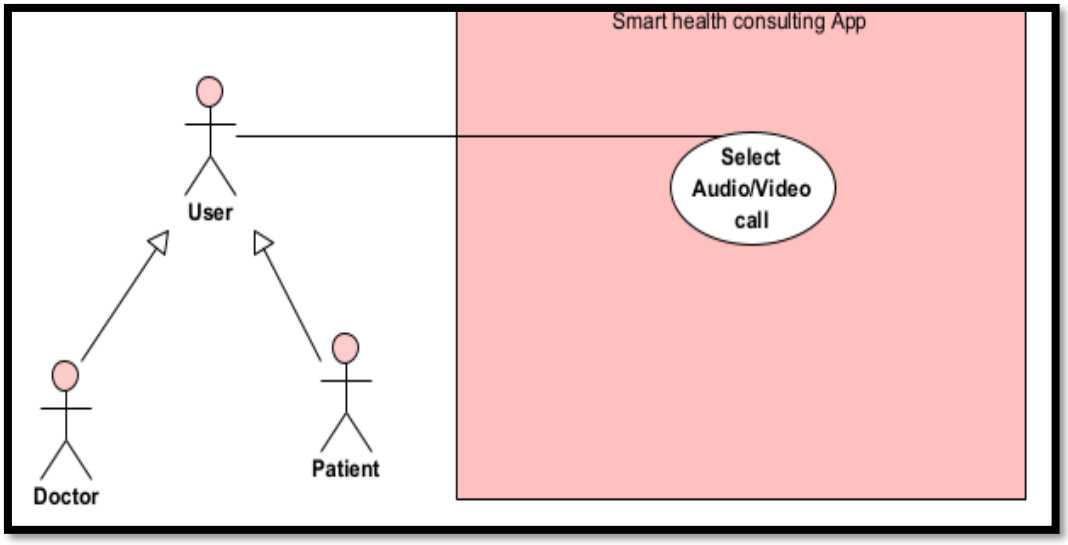

Figure 17: Use case Diagram of Select Audio/Video call

Table 16: Use Case Description of Select Audio/Video Call

Use Case ID:	**UC16**	
Use Case Name:	**Select Audio/video call**	
Actor(s):	User	
Pre-Conditions:	The user is logged in and the appointment is confirmed by a doctor.	
Priority:	High	
Basic Flow:	1. The user first has accepted an appointment. 2. Then the user clicks on the chat. 3. User selects audio/video call.	
Actor Actions		**System Response**
1	User presses the audio/video call button.	Audio/video call successfully done, patient and doctor can interact with each other on the live call.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	N/A	N/A

3.2.1.16 Get Consultation

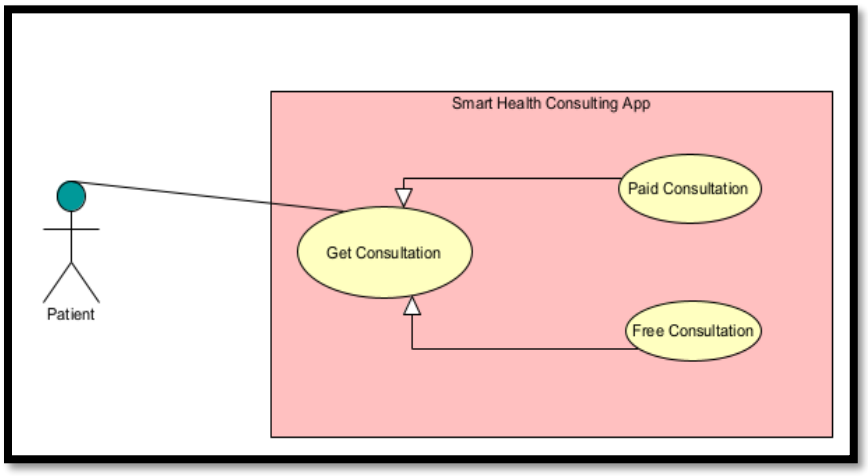

Figure 18: Use case Diagram of Get Consultation

3.2.1.16.1 Paid consultation

Table 17: Use Case Description of Paid Consultation

Use Case ID:	**UC17**
Use Case Name:	**Paid Consultation**
Actor(s):	Patient
Pre-Conditions:	Patient logged in.
Priority:	High
Basic Flow:	1. The patient is on the home page. 2. Patient clicks on paid consultation module. 3. The patient selects the Specialization field. 4. The patient selects a specialist from the list of specialists and can view his profile and review & ratings. 5. The patient enters the date and time and describes their issue by entering the title of that problem. 6. Then click on Book Appointment. 7. Before an appointment is booked, the patient enters a mobile no to send payment by clicking on the payment button. 8. The system displays the message that "your transaction is successful". 9. After payment, a consultation request is sent to the doctor for confirmation. 10. The system will update the wallet.

Actor Actions		**System Response**
1	The patient selects Paid Consultation.	The system shows a list of available specialist doctors.
2	The patient pays the consultation fee.	A message appears Payment paid successfully.

Alternative Course of Action (if any)		
Actor Action		**System Response**
1	The patient failed to pay a consultation fee.	The patient will not be able to get paid consultation.

3.2.1.16.2 Free consultation

Table 18: Use Case Description of Free Consultation

Use Case ID:	**UC18**
Use Case Name:	**Free Consultation**
Actor(s):	Patient
Pre-Conditions:	The Patient is logged in.
Priority:	High
Basic Flow:	1. The patient is on the home page. 2. The patient clicks on the free consultation module. 3. The patient can see posts of other patients and comment on that posts as well. 4. Patients click on it to describe the disease and by clicking on that patients can post their problem and attach a picture with that. 5. The system successfully posts the patient's problem.

Actor Actions		**System Response**
1	The patient selects Free Consultation.	The system shows a list of available doctors.
Alternative Course of Action (if any)		
Actor Action		**System Response**

1	If a patient failed to fill all fields.	System display message" Fill all fields".

3.2.1.17 Payment Integration

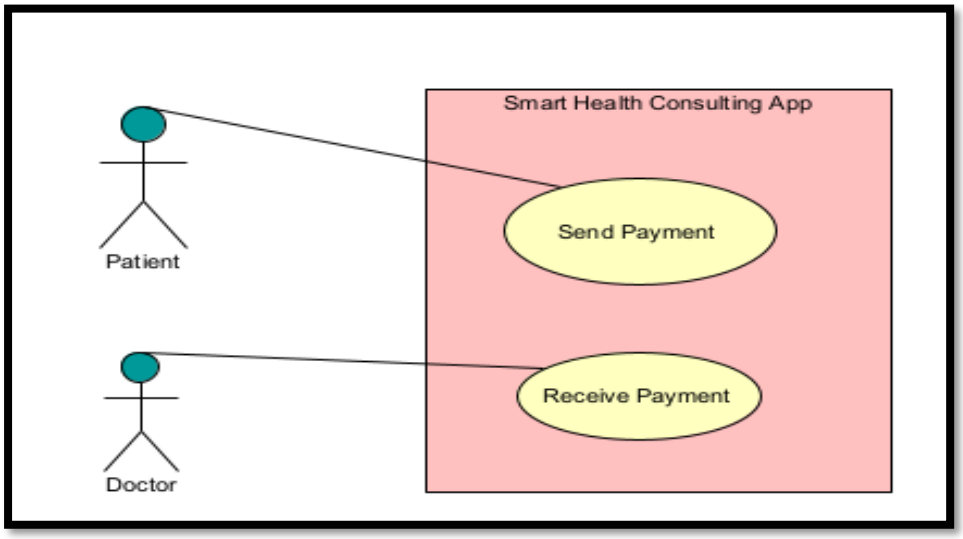

Figure 19: Use case Diagram of Payment Integration

3.2.1.17.1 Send Payment

Table 19: Use Case Description of Send Payment

Use Case ID:	**UC19**
Use Case Name:	**Send Payment**
Actor(s):	Patient
Pre-Conditions:	The patient is logged in.
Priority:	High
Basic Flow:	1. The patient is on the home page. 2. Patient clicks on paid consultation module. 3. The patient selects the Specialization field.

	4. The patient selects a specialist from the list of specialists and can view his profile and review & ratings. 5. The patient enters the date and time and describes their issue by entering the title of that problem. 6. Then click on Book Appointment. 7. Before an appointment is booked, the patient enters a mobile no to send payment by clicking on the payment button. 8. System displays the message that "your transaction is successful".

Actor Actions		**System Response**
1	The patient selects the available doctor and pays the consultation fee.	The consultation fee is paid successfully, and the consultation is booked.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	The patient failed to pay a consultation fee.	The consultation fee is not paid so a consultation is not booked.
2	The patient failed to enter mobile no.	The system displays an error message "Enter mobile No".

3.2.1.17.2. Receive Payment

Table 20: Use Case Description of Receive Payment

Use Case ID:	**UC20**
Use Case Name:	**Receive Payment**
Actor(s):	Doctor
Pre-Conditions:	The Doctor is logged in.
Priority:	High

Basic Flow:	1. The doctor is on the home page. 2. The doctor clicks on the MyAppointment screen. 3. The doctor can see the patient's profile. 4. The doctor confirms the Appointment. 5. The doctor receives payment, and the System displays the payment in Doctor's wallet.

Actor Actions		**System Response**
1	The doctor receives payment.	The system confirms receiving payment.
Alternative Course of Action (if any)		
Actor Action		**System Response**
	NA	NA

3.2.1.18 Search Doctor

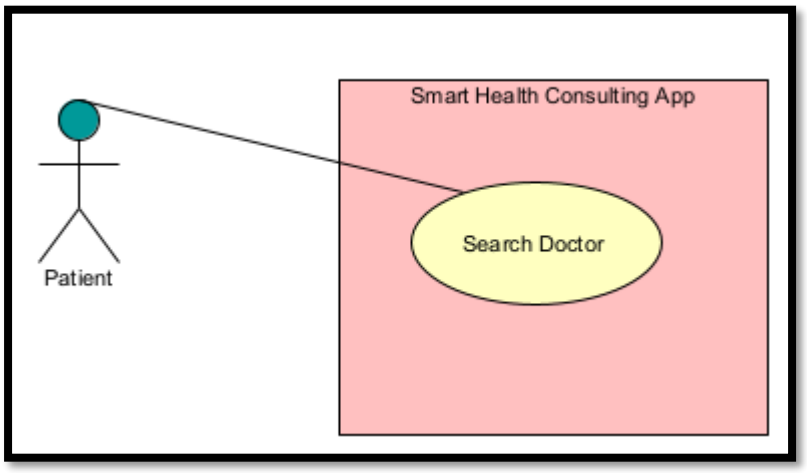

Figure 20: Use case Diagram of Search Doctor

Table 21: Use Case Description of Search Doctor

Use Case ID:	**UC21**
Use Case Name:	**Search Doctor**
Actor(s):	Patient
Pre-Conditions:	The patient must log in to the app and the appointment is confirmed.
Priority:	High

Basic Flow:	1. The patient clicks on the chat screen. 2. The system display's chat screen to the patient. 3. The patient can search for doctors by typing their full names into the search bar. 4. System display's the doctor successfully.	
Actor Actions		**System Response**
1	Patient search for the doctor from the search bar.	The searched doctor will appear on the screen.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	Patient search for the doctor from the search bar.	The doctor is not available so the message "No results found" will appear.

3.2.1.19 Give Reviews and Ratings

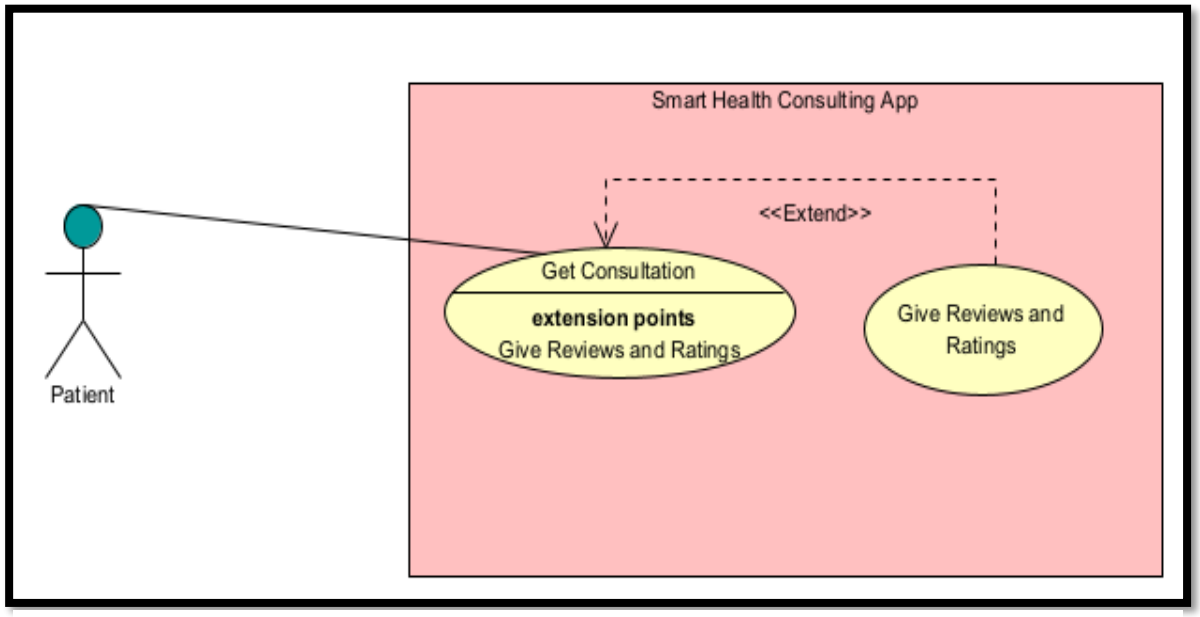

Figure 21: Use case Diagram of Give Reviews and Ratings

Table 22: Use Case Description of Give Reviews and Ratings

Use Case ID:	**UC22**	
Use Case Name:	**Give Reviews and Ratings**	
Actor(s):	Patient	
Pre-Conditions:	The patient is logged in.	
Priority:	High	
Basic Flow:	1. The patient is on the home page. 2. Patient clicks on paid consultation module. 3. The patient sends a Consultation request to the doctor. 4. The doctor accepts consultation. 5. After getting a consultation patient can give reviews and ratings to the doctor.	
Actor Actions		**System Response**
1	The patient gives Reviews and Ratings.	A message will be displayed "Thank you for your response."
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	NA	NA

3.2.1.20 Confirm Appointment

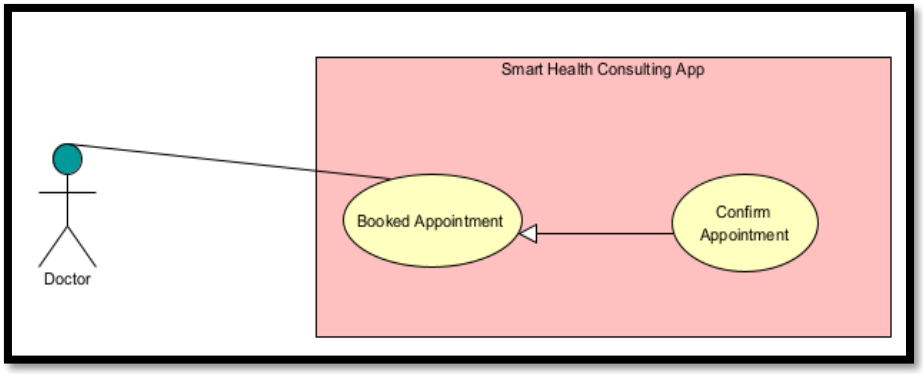

Figure 22: Use case Diagram of Confirm Appointment

Table 23: Use Case Description of Confirm Appointment

Use Case ID:	**UC23**
Use Case Name:	**Confirm Appointment**
Actor(s):	Doctor
Pre-Conditions:	The doctor is logged in.
Priority:	High
Basic Flow:	1. The doctor receives notification that a "New consultation request". 2. The doctor is on the home page. 3. The doctor clicks on the MyAppointment screen. 4. The doctor can see the patient's profile. 5. The doctor confirms the Appointment. 6. The system displays the payment in Doctor's wallet. 7. Once the appointment is confirmed patient receives notification that "Appointment is accepted".

Actor Actions		**System Response**
1	The doctor confirms appointments.	A message appears "appointments are confirmed".
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	The doctor is not available, so he cancels the appointment.	A message appears "Appointment canceled".

3.2.2 Admin Management

3.2.2.1. Login

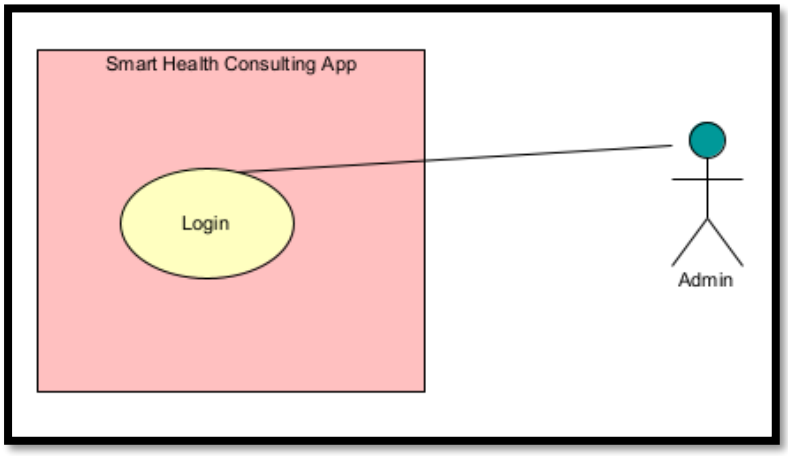

Figure 23: Use case Description of Admin Login

Table 24: Use Case Description of Admin Login

Use Case ID:	**UC24**	
Use Case Name:	**Login**	
Actor(s):	Admin	
Pre-Conditions:	Admin adds information to login.	
Priority:	High	
Basic Flow:	1. Opens the mobile app. 2. Enters email and password. 3. Click Login. Logged in to the mobile app.	
Actor Actions		**System Response**
1	Admin inputs an email and password.	System check admin's provided credentials. If the admin's credentials are valid, the admin is authenticated, and the home screen is displayed.
Alternative Course of Action (if any)		

Actor Action		System Response
1	Admin forgot to fill a field.	An error message will be displayed to fill all fields.
2	Admin enters password less than 7 characters.	An error message will be displayed password should be more than 7 characters.
3	Admin enters email.	An error message will be displayed that this account doesn't exist.

3.2.2.2. Add User

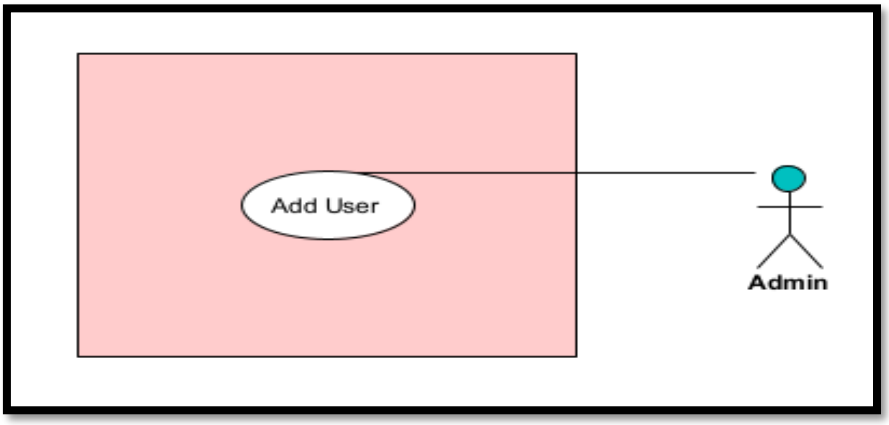

Figure 24: Use case Diagram of Add User

Table 25: Use Case Description of Add user

Use Case ID:	**UC25**
Use Case Name:	**Add User**
Actor(s):	Admin
Pre-Conditions:	Admin is logged in as admin.
Priority:	High

Basic Flow:	1. Admin is at Dashboard. 2. Admin clicks on the drawer. 3. Admin select Manage patient. 4. By clicking on the Add user icon, the admin can add users by entering mail and password.	
Actor Actions		**System Response**
1	Admin adds users.	The user added successfully.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	Admin forgot to fill a field.	An error message will be displayed to fill all fields.

3.2.2.3. Manage Patients:

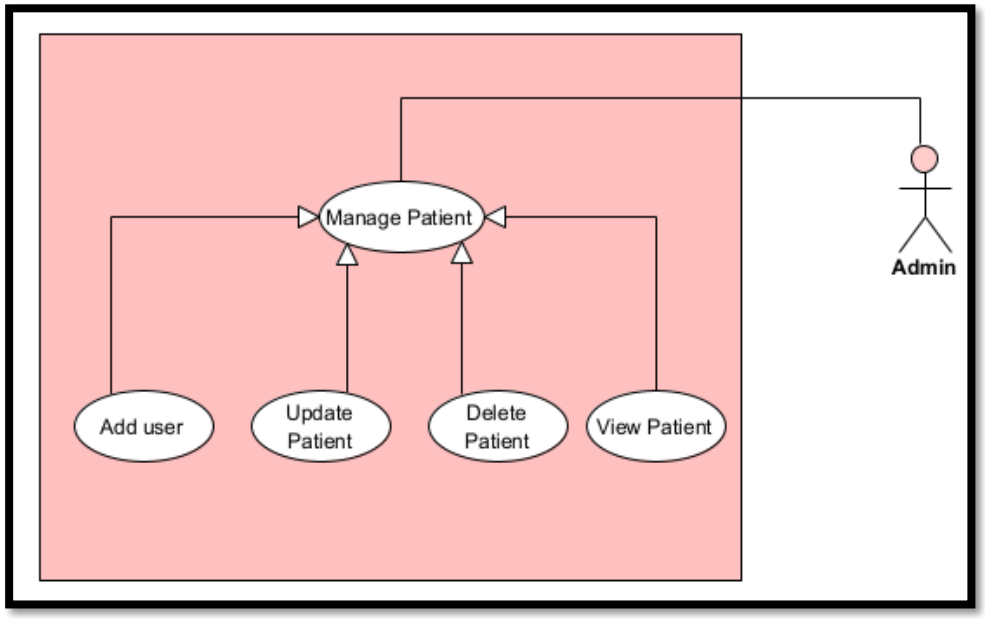

Figure 25: Use case Diagram of Manage Patient

3.2.2.3.1. Update Patient

Table 26: Use Case Description of Update Patient

Use Case ID:	**UC26**	
Use Case Name:	**Update Patient**	
Actor(s):	Admin	
Pre-Conditions:	Admin is logged in as admin.	
Priority:	High	
Basic Flow:	1. Admin is at Dashboard. 2. Admin clicks on the drawer. 3. Admin select Manage Patients. 4. The system displays a list of all Patients. 5. By clicking on The Edit icon, the admin can update the patient record.	
Actor Actions		**System Response**
1	Admin updates the information.	Patient information was updated successfully.
Alternative Course of Action (if any)		
Actor Action		**System Response**
	NA	NA

3.2.2.3.2. View Patient

Table 27: Use Case Description of View Patient

Use Case ID:	**UC27**
Use Case Name:	**View Patient**
Actor(s):	Admin

Pre-Conditions:	Admin is logged in as admin.	
Priority:	Medium	
Basic Flow:	1. Admin is at Dashboard. 2. Admin clicks on the drawer. 3. Admin select Manage patient. 4. The system displays a list of all Patients. 5. By clicking on the view icon, the admin can view the Patient record. System display record successfully.	
Actor Actions		**System Response**
1	Admin selects profile and views information.	The profile is displayed.
Alternative Course of Action (if any)		
Actor Action		**System Response**
	NA	NA

3.2.2.3.3. Delete Patient

Table 28: Use Case Description of Delete Patient

Use Case ID:	**UC28**
Use Case Name:	**Delete Patient**
Actor(s):	Admin
Pre-Conditions:	Admin is logged in as admin.
Priority:	Medium

Basic Flow:	1. Admin is at Dashboard. 2. Admin clicks on the drawer. 3. Admin select Manage Patients. 4. The System displays a list of all Patients. 5. By clicking on the delete icon, the admin can then delete the patient.	
Actor Actions		**System Response**
1	Admin deletes the patient.	System display message "Patient deleted successfully".
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	NA	NA

3.2.2.4. Manage Doctor:

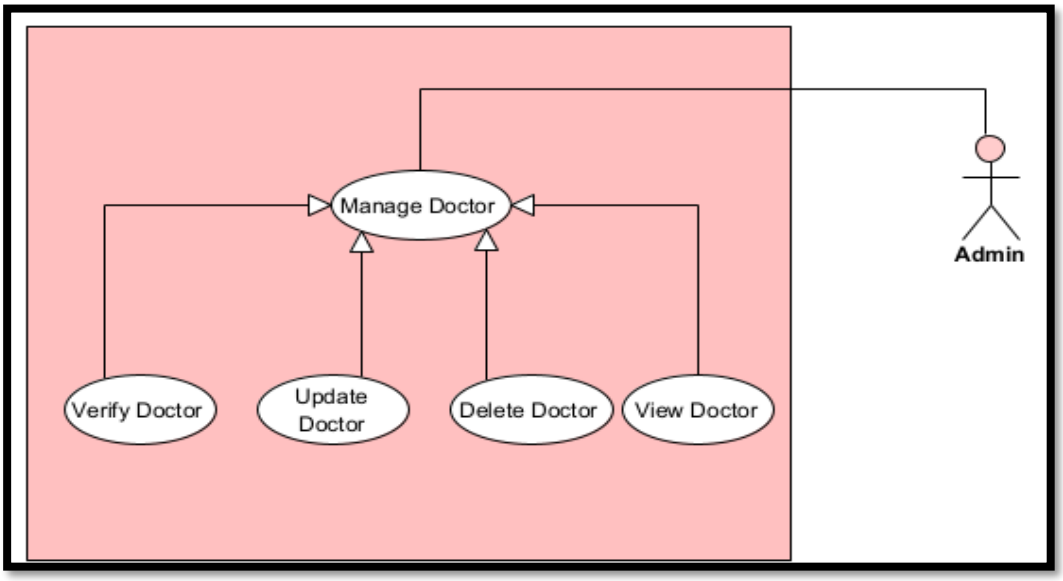

Figure 26: Use case Diagram of Manage Doctor

3.2.2.4.1. Update Doctors

Table 29: Use Case Description of Update Doctor

Use Case ID:	**UC29**	
Use Case Name:	**Update Doctor**	
Actor(s):	Admin	
Pre-Conditions:	Admin is logged in as admin.	
Priority:	High	
Basic Flow:	1. Admin is at Dashboard. 2. Admin clicks on the drawer. 3. Admin select Doctor verification. 4. The System displays a list of all Doctor. 5. By clicking on the Edit icon, the admin can update the Doctor's record.	
Actor Actions		**System Response**
1	Admin updates the information.	System display the message "Doctor Updated successfully".
Alternative Course of Action (if any)		
Actor Action		**System Response**
	NA	NA

3.2.2.4.2. View Doctors

Table 30: Use Case Description of View Doctor

Use Case ID:	**UC30**
Use Case Name:	**View Doctors**
Actor(s):	Admin
Pre-Conditions:	Admin is logged in as admin.

Priority:	Medium	
Basic Flow:	1. Admin is at Dashboard. 2. Admin clicks on the drawer. 3. Admin select Doctor verification. 4. The System displays a list of all Doctors. 5. By clicking on the view icon, the admin can view Doctor's record. System display record successfully.	
Actor Actions		**System Response**
1	Admin selects profile and views information.	The profile is displayed.
Alternative Course of Action (if any)		
Actor Action		**System Response**
	NA	NA

3.2.2.4.3. Delete Doctor

Table 31: Use Case Description of Delete Doctor

Use Case ID:	**UC31**
Use Case Name:	**Delete Doctor**
Actor(s):	Admin
Pre-Conditions:	Admin is logged in as admin.
Priority:	Medium

Basic Flow:	1. Admin is at Dashboard. 2. Admin clicks on the drawer. 3. Admin select Doctor Verification. 4. The System displays a list of all doctors. 5. By clicking on the delete icon, the admin can delete the doctors.	
Actor Actions		**System Response**
1	Admin deletes the doctor.	System display message "Doctor deleted successfully".
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	NA	NA

3.2.2.4.4. Verify doctor

Table 32: Use Case Description of Verify Doctor

Use Case ID:	**UC32**
Use Case Name:	**Verify doctor**
Actor(s):	Admin
Pre-Conditions:	Admin is logged in as admin.
Priority:	Medium
Basic Flow:	1. Admin is at Dashboard. 2. Admin clicks on the drawer. 3. Admin select Doctor verification. 4. The System displays a list of all Doctors.

	5. By clicking on the view icon, the admin can view approve doctors by looking at their authentic data.	
Actor Actions		**System Response**
1	Admin verifies doctor.	The System successfully approves the doctor.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1	NA	NA

3.2.2.5. View Statistics

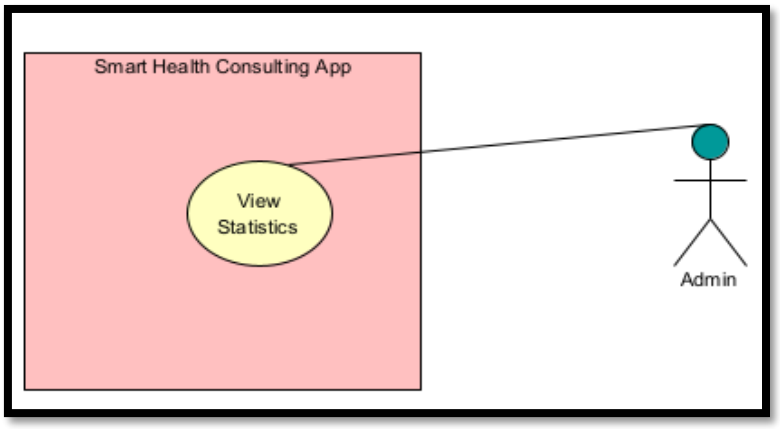

Figure 27: Use case Diagram of View Statistics

Table 33: Use Case Description of View Statistics

Use Case ID:	**UC33**
Use Case Name:	**View Statistics**
Actor(s):	Admin

Pre-Conditions:	Admin is logged in as admin.	
Priority:	High	
Basic Flow:	1. Admin is at Dashboard. 2. Admin can view statistics yearly, monthly, or weekly by clicking on that button.	
Actor Actions		**System Response**
1	Admin selects statistics.	The System displays all the Statistics of users successfully.
Alternative Course of Action (if any)		
Actor Action		**System Response**
	NA	NA

3.2.2.6. View Feedback

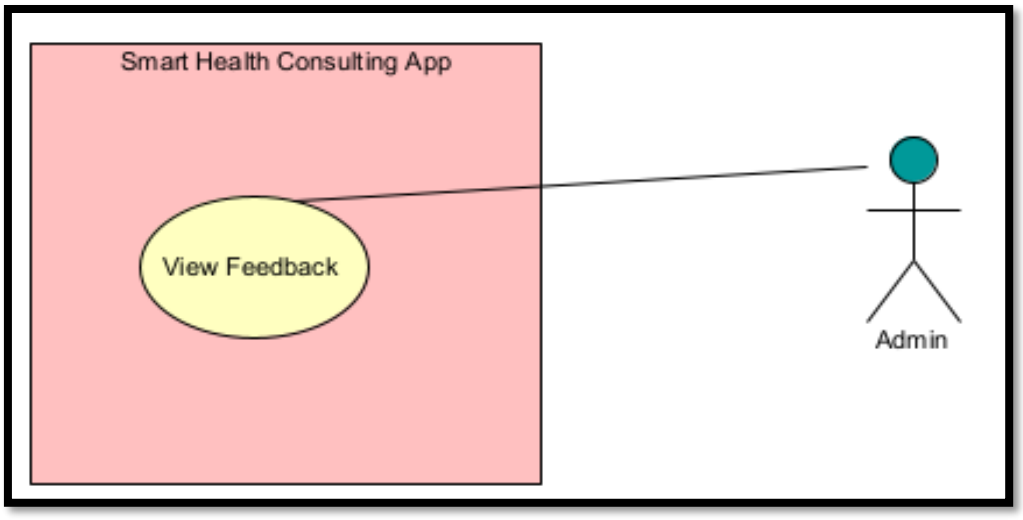

Figure 28: Use case Diagram of View Feedback

Table 34: Use Case Description of View Feedback

Use Case ID:	**UC34**	
Use Case Name:	**View Feedback**	
Actor(s):	Admin	
Pre-Conditions:	Admin is logged in as admin.	
Priority:	High	
Basic Flow:	1. Admin is at Dashboard. 2. Admin click on the drawer. 3. Admin select view feedback screen.	
Actor Actions		**System Response**
1	Admin selects view Feedback.	The System displays all the Feedback of users successfully.
Alternative Course of Action (if any)		
Actor Action		**System Response**
	NA	NA

3.2.2.7. View Appointment Details

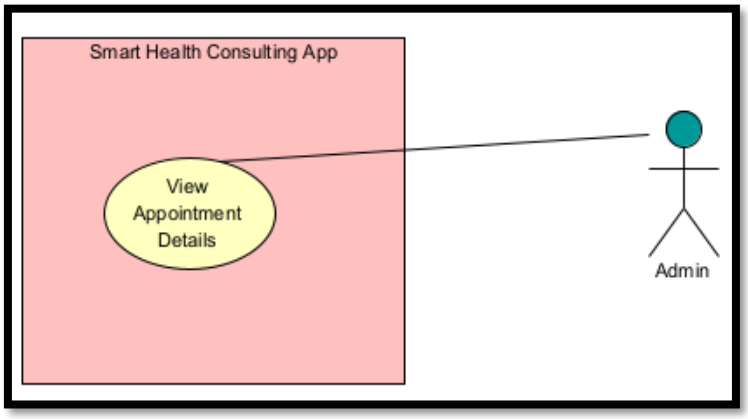

Figure 29: Use case Diagram of View Appointment Details

Table 35: Use Case Description of View Appointment Details

Use Case ID:	**UC35**	
Use Case Name:	**View Appointment Details**	
Actor(s):	Admin	
Pre-Conditions:	Admin is logged in as admin.	
Priority:	High	
Basic Flow:	1. Admin is at Dashboard. 2. Admin clicks on the drawer. 3. Admin select view Appointment screen.	
Actor Actions		**System Response**
1	Admin selects View Appointment.	The System displays all the appointments successfully.
Alternative Course of Action (if any)		
Actor Action		**System Response**
	NA	NA

3.3. Functional Requirements:

3.3.1 User Management

3.3.1.1. Registration:

Table 36: Functional Requirement of Registration

Identifier	FR1
Title	Registration
Requirement	The system should allow the user to register an account by selecting a role (as patient or doctor) and by providing an email and password, or through mobile and google authentication.
Source	Mobile Application
Rationale	Registration will help in providing security.
Restrictions and Risk	All the information provided by the user should be correct and no required field should be missing.
Dependencies	N/A
Priority	High

3.3.1.2. Login:

Table 37: Functional Requirement of Login

Identifier	FR2
Title	Log in
Requirement	The system should allow the user to log in to the system with their Email and Password.
Source	Mobile Application

Rationale	To give access to authenticate users only.
Restrictions and Risk	The user should enter the correct credentials.
Dependencies	FR1
Priority	High

3.3.1.3. Logout:

Table 38: Functional Requirement of Logout

Identifier	FR3
Title	Log out
Requirement	The system should allow the user to log out from the system by clicking the “Logout” button.
Source	Mobile Application
Rationale	The user wants to close the system and its functionalities.
Restrictions and Risk	N/A
Dependencies	FR2
Priority	High

3.3.1.4. Forget Password:

Table 39: Functional Requirement of Forgot Password

Identifier	FR4
Title	Forget password
Requirement	The system should allow the user to access his account, if he forgets his password.

Source	Mobile Application
Rationale	The user wants to access his account.
Restrictions and Risk	N/A
Dependencies	FR1
Priority	High

3.3.1.5. Update Profile:

Table 40: Functional Requirement of Update Profile

Identifier	FR5
Title	Update profile
Requirement	The system should allow the user to update any information in his profile.
Source	Mobile Application
Rationale	To maintain the most recent records.
Restrictions and Risk	N/A
Dependencies	FR2, FR6
Priority	High

3.3.1.6. View Profile:

Table 41: Functional Requirement of View Profile

Identifier	FR6
Title	View profile
Requirement	The system should allow the user to view their profiles and can view doctor profile as well.
Source	Mobile Application

Rationale	To provide information about the doctor and patient.
Restrictions and Risk	N/A
Dependencies	N/A
Priority	High

3.3.1.7. Give Feedback:

Table 42: Functional Requirement of Give Feedback

Identifier	FR7
Title	Give feedback
Requirement	The system should allow the user to give their feedback to the admin.
Source	Mobile Application.
Rationale	To provide feedback about App.
Restrictions and Risk	N/A
Dependencies	N/A
Priority	High

3.3.1.8. View MyAppointment:

Table 43: Functional Requirement of View MyAppointment

Identifier	FR8
Title	View MyAppointment
Requirement	The system should allow the user to view his/her Appointment that is booked.
Source	Mobile Application

Rationale	To make it easier for the doctor or patient to view their appointments.
Restrictions and Risk	N/A
Dependencies	FR15
Priority	High

3.3.1.9. View Notification:

Table 44: Functional Requirement of View Notification

Identifier	FR9
Title	View Notification
Requirement	The system should allow the user to view the Notification.
Source	Mobile Application
Rationale	To view notification.
Restrictions and Risk	N/A
Dependencies	N/A
Priority	High

3.3.1.10. Change Language:

Table 45: Functional Requirement of Change Language

Identifier	FR10
Title	Change Language
Requirement	The system should allow the user to change language (Urdu or English) according to their comfort.
Source	Mobile Application
Rationale	To provide facility to our users to change the language.
Restrictions and Risk	N/A

Dependencies	N/A
Priority	High

3.3.1.11. Real-time chat:

Table 46: Functional Requirement of Real-Time Chat

Identifier	FR11
Title	Real-time Chat
Requirement	The system should provide interactions between doctor and patient by providing a chat.
Source	Mobile Application
Rationale	To provide information about the doctor and patient.
Restrictions and Risk	Appointment must be Confirmed by a doctor to do chat with the doctor.
Dependencies	FR15
Priority	High

3.3.1.12. Book Appointment:

Table 47: Functional Requirement of Book Appointment

Identifier	FR12
Title	Book Appointment
Requirement	The system should allow the patient to book an appointment by doing payment first.
Source	Mobile Application
Rationale	To get a consultation from a doctor.
Restrictions and Risk	To book an appointment payment must be done.
Dependencies	FR14
Priority	High

3.3.1.13. Cancel Appointment:

Table 48: Functional Requirement of Cancel Appointment

Identifier	FR13
Title	Cancel Appointment
Requirement	The system should allow the doctor to cancel the appointment according to his/her feasibility.
Source	Mobile Application
Rationale	To cancel the appointment according to the doctor's availability.
Restrictions and Risk	N/A
Dependencies	FR12
Priority	High

3.3.1.14. Payment:

Table 49: Functional Requirement of Payment

Identifier	FR14
Title	Payment
Requirement	The system should allow the patient to make payment before booking any consultation with a doctor.
Source	Mobile Application
Rationale	To make payments online.
Restrictions and Risk	N/A
Dependencies	FR2
Priority	High

3.3.1.15. Confirm Appointment:

Table 50: Functional Requirement of Confirm Appointment

Identifier	FR15
Title	Confirm Appointment
Requirement	The system should allow the doctor to confirm the appointment according to their feasibility.
Source	Mobile Application
Rationale	To confirm an appointment with the patient to give a consultation.
Restrictions and Risk	N/A
Dependencies	FR12
Priority	High

3.3.1.16. Send Image:

Table 51: Functional Requirement of Send Image

Identifier	FR16
Title	Send Image
Requirement	The system should allow the user to send images of their reports or any skin disease they have.
Source	Mobile Application
Rationale	To send an image to have some proof.
Restrictions and Risk	N/A
Dependencies	FR11
Priority	High

3.3.1.17. Select Theme:

Table 52: Functional Requirement of Select Theme

Identifier	FR17
Title	Select Theme
Requirement	The system should allow the user to select a theme (dark or light) according to their feasibility.
Source	Mobile Application
Rationale	To select a theme of their own choice.
Restrictions and Risk	N/A
Dependencies	FR2
Priority	High

3.3.1.18. Select Audio/Video call:

Table 53: Functional Requirement of Select Audio/Video Call

Identifier	FR18
Title	Select Audio/Video call
Requirement	The system should allow the user to make audio/video call.
Source	Mobile Application
Rationale	To interact with another user (patient or doctor)
Restrictions and Risk	N/A
Dependencies	FR11
Priority	High

3.3.1.19. Give ratings and reviews to doctor

Table 54: Functional Requirement of Give Ratings and Reviews to Doctor

Identifier	FR19
Title	Give ratings and reviews to doctors
Requirement	The system should allow the patient to give ratings and reviews to the doctor.
Source	Mobile Application
Rationale	To be able to mention their experience with the specific doctor.
Restrictions and Risk	N/A
Dependencies	N/A
Priority	High

3.3.1.20. View ratings and reviews of doctor

Table 55: Functional Requirement of View Rating and Reviews to Doctor

Identifier	FR20
Title	View ratings and reviews of doctors
Requirement	The system should allow the patient to view ratings and reviews of doctor.
Source	Mobile Application
Rationale	To choose the best doctor by viewing ratings and reviews of doctors.
Restrictions and Risk	N/A
Dependencies	N/A
Priority	High

3.3.1.21. Search Doctor

Table 56: Functional Requirement of Search Doctor

Identifier	FR21
Title	Search Doctor
Requirement	The system should allow the patient to search doctors.
Source	Mobile Application
Rationale	To be able to search for a doctor.
Restrictions and Risk	N/A
Dependencies	N/A
Priority	High

3.3.1.22. Free Consultation

Table 57: Functional Requirement of Free Consultation

Identifier	FR22
Title	Free Consultation
Requirement	The system should allow the patient to post their problem in a free consultation post and the doctor can answer them according to their willingness.
Source	Mobile Application
Rationale	To be able to get free consultation.
Restrictions and Risk	N/A
Dependencies	N/A
Priority	High

3.3.1.23. Paid Consultation:

Table 58: Functional Requirement of Paid Consultation

Identifier	FR23
Title	Paid Consultation
Requirement	The system should allow the patient to get a consultation by booking an appointment.
Source	Mobile Application
Rationale	To be able to get paid consultation.
Restrictions and Risk	Payment must be done before getting paid consultation.
Dependencies	N/A
Priority	High

3.3.2. Admin Management:

3.3.2.1. Add User:

Table 59: Functional Requirement of Add User

Identifier	FR24
Title	Add User
Requirement	The system should allow the admin to add a user.
Source	Mobile Application
Rationale	To be able to add a user from the admin side.
Restrictions and Risk	N/A
Dependencies	N/A
Priority	High

3.3.2.2. Delete User:

Table 60: Functional Requirement of Delete User

Identifier	FR25
Title	Delete User
Requirement	The system should allow the admin to delete user.
Source	Mobile Application
Rationale	To be able to delete users from the admin side.
Restrictions and Risk	N/A
Dependencies	N/A
Priority	High

3.3.2.3. View User:

Identifier	FR26
Title	View User
Requirement	The system should allow the admin to view user.
Source	Mobile Application
Rationale	To be able to view users from the admin side.
Restrictions and Risk	N/A
Dependencies	N/A
Priority	High

Table 61: Functional Requirement of View User

3.3.2.4. Update User:

Table 62: Functional Requirement of Update User

Identifier	FR27
Title	Update User
Requirement	The system should allow the admin to Update user.
Source	Mobile Application
Rationale	To be able to Update users from the admin side.
Restrictions and Risk	N/A
Dependencies	N/A
Priority	High

3.3.2.5. View Appointment Details:

Table 63: Functional Requirement of View Appointment Details

Identifier	FR28
Title	View Appointment Details
Requirement	The system should allow the admin to view all the appointment details.
Source	Mobile Application
Rationale	To keep a track of all the appointments.
Restrictions and Risk	N/A
Dependencies	N/A
Priority	High

3.3.2.6. View Feedback:

Table 64: Functional Requirement of View Feedback

Identifier	FR29
Title	View Feedback
Requirement	The system should allow the admin to view all the Feedback given by the patient and doctor.
Source	Mobile Application
Rationale	To get reviews about the application from users.
Restrictions and Risk	N/A
Dependencies	N/A
Priority	High

3.3.2.7. View Statistics:

Table 65: Functional Requirement of View Statistics

Identifier	FR30
Title	View Statistics
Requirement	The system should allow the admin to view Statistics.
Source	Mobile Application
Rationale	To be able to view statistics.
Restrictions and Risk	N/A
Dependencies	N/A
Priority	High

3.3.2.8. Verify Doctors:

Table 66: Functional Requirement of Verify Doctor

Identifier	FR31
Title	Verify Doctors
Requirement	The system should allow the admin to verify doctors by looking at their degrees or PMC No.
Source	Mobile Application
Rationale	To be able to verify doctors.
Restrictions and Risk	N/A
Dependencies	N/A
Priority	High

3.4. Interface Requirements

3.4.1. User Interfaces

The Front-End and Back-End of the application will be developed using flutter by using the latest version of flutter. For user interface design, basic GUI standards will be followed along with the presence of error message display standards, etc., and standard buttons and functions (i.e. back button) will appear on every screen. Details of the user interface design are intended to be documented in a different user interface in detail.

UI-1: The Smart health consulting application screen will follow HCI rules and the concept of "Easy to learn, Easy to use".

UI-2: The system will provide a search doctor bar.

UI-3: The mobile pages shall permit complete navigation and selection of different services, and other information available on the application.

3.4.2. Hardware Interfaces

Any device that can connect to the internet can be used for the application to run.

3.4.3. Software Interfaces

Software tools and technology	Version	Rationale
Visual Studio Code	1.66	IDE
Flutter	2.5.0	Android and iOS App
Dart	2.14.1	Language
Firebase	19.0.1	Database (The system shall use firebase, which is open source and free)

Table 67: Software Interfaces

3.4.4. Communications Interfaces

The system will use APIs for the payment module and the database will communicate with the client and service provider for the sake of interaction between each other. Our system will provide an online chat support system for the patient and doctor to communicate with each other easily.

- HTTP Communication standard

3.5. Database Requirements

The Firebase Real-time Database is a cloud-hosted service. Data is stored as JSON and synchronized in real time to every connected client. When you create cross-platform apps with iOS, Android, and SDKs, all your clients share a single Real-time Database instance and are automatically updated with the most recent data.

3.6. Non-Functional Requirements

3.6.1. Performance Requirements

The Smart Health Consultation App should be loaded within 10 sec. The app should allow more than 1000 users to access it. The app will be updated consistently to maintain a strategic distance from issues. In addition to the above, the mobile app should have the following abilities and capabilities.

NFR1: The responsiveness of the app should be high, and the app should respond according to the user's action.

NFR2: The user must be acknowledged in the form of visual changes or feedback on the site to enhance the interaction.

NFR3: The response time on the app shall be minimal.

NFR4: Consistency on the mobile app shall be maintained across all the pages/screens

NFR5: The layout of the mobile app shall be kept simple and must be self-explanatory.

3.6.2. Safety Requirements

Our app also considers safety requirements as a crucial part of fulfilling non-functional requirements. We will provide all the authentic and verified doctors to maintain patients' safety by getting appropriate and authentic guidance regarding their health.

3.6.3. Security Requirements

NFR5. The Mobile App must provide a secure login option to the users to prevent unauthorized access to the system and the information.

NFR6: Security Requirements should be fulfilled to avoid misuse of the data sets. Technical controls, such as anti-malware, anti-denial, and intrusion detection tools must be integrated with the app.

3.6.4. Software Quality Attributes

Our app also specifies some of the software quality attributes that are:

3.6.4.1. Usability:

The user will be given the convenience of use by making the interface easy to use and clearer. Users will be notified of every wrong step that they perform, and the system will give clues to utilize the system in a better way.

3.6.4.2. Availability:

The system must be available 24 hours a day, 7 days a week.

3.6.4.3. Correctness:

Our mobile app should have the ability to perform the exact tasks that are defined by their specification.

3.6.4.4. Flexibility:

Our mobile app will be flexible in such a way that it can be modified to different environments (android or iOS), configurations, and user exceptions. To achieve flexibility in our app we are using the latest technologies like flutter which is a cross-platform mobile app development technology.

3.6.4.5. Maintainability:

Our Smart Health Consultation app would have the ability to go through changes with a fair degree of effortlessness. Our app will be easy to maintain in all aspects like enhancement, modification, fixing issues, reconstruction, or even adding new functionality.

3.6.4.6. Reliability:

Our app will perform the specified functions consistently without failure. Our team would eliminate all defects that could affect code safety and issues with system components in order to achieve high reliability.

3.6.4.7. Reusability:

Reusability is the extent to which a portion of the software system can be reused for the development of another system. Our app will have the property of reusability in which parts of the information system, or the design, can be reused for the development of different applications.

3.7. Project Feasibility

This project is an online platform where a patient can get a consultation from a doctor.
Our app will run on the internet because of the dynamic nature of the internet, anybody can access it from a smartphone. Patient and doctor chat will be confidential between them, ensuring the Patient's confidentiality We have classified feasibility into sub-categories; Technical, and ethical feasibility.

3.7.1. Technical Feasibility

This system is an application. We developed the system so that it can be scalable, separate implementations of logic, and easy to maintain and upgrade. This system was developed by keeping in mind the requirements, risks, and security issues related to the system.

3.7.2. Legal and Ethical Feasibility

Our system fulfills its legal and ethical aspects. It cannot be used without proper authorization. The system provides security to the data The doctors that we are providing are verified by PMC (Pakistan Medical Commission).

3.8. Conclusion

In the above chapter, we introduced how the different types of requirements that can be organized and analyzed. Categorizing the requirements such as functional and non-functional requirements along with use case description makes it easier to develop the question list as well as to identify gaps in knowledge. The objective of the above chapter is to identify the requirements of smart health consulting app and make sure that the project to be developed is fully understood from all perspectives.
In this chapter, we also analyzed the feasibility of smart health consulting apps from different perspectives i.e. technically, legally, and ethically.

Chapter # 4

System Design

Chapter no. 4

System Design

4.1. Design Approach

The Design Approach is a conceptual model that defines the structure, behaviour, and view of the system. It is a formal representation of the system and provides infrastructure for the system.

Our software is following the Observer pattern. We are using the observer pattern because it is used when there is a one-to-many relationship between objects such as if one object is modified, its dependent objects are to be notified automatically.

4.2. Design Constraints

Reliability	There should exist a reliable connection between mobile applications and databases.
Criticality of the Application	The application should respond and communicate the data in real-time immediately.
Safety and security considerations	The connection between the database and application should be a secured one.
Internet Connection	As smart health consulting app is about providing services online, there should be a good internet connection.

Table 68: Design Constraints

4.3. System Architecture

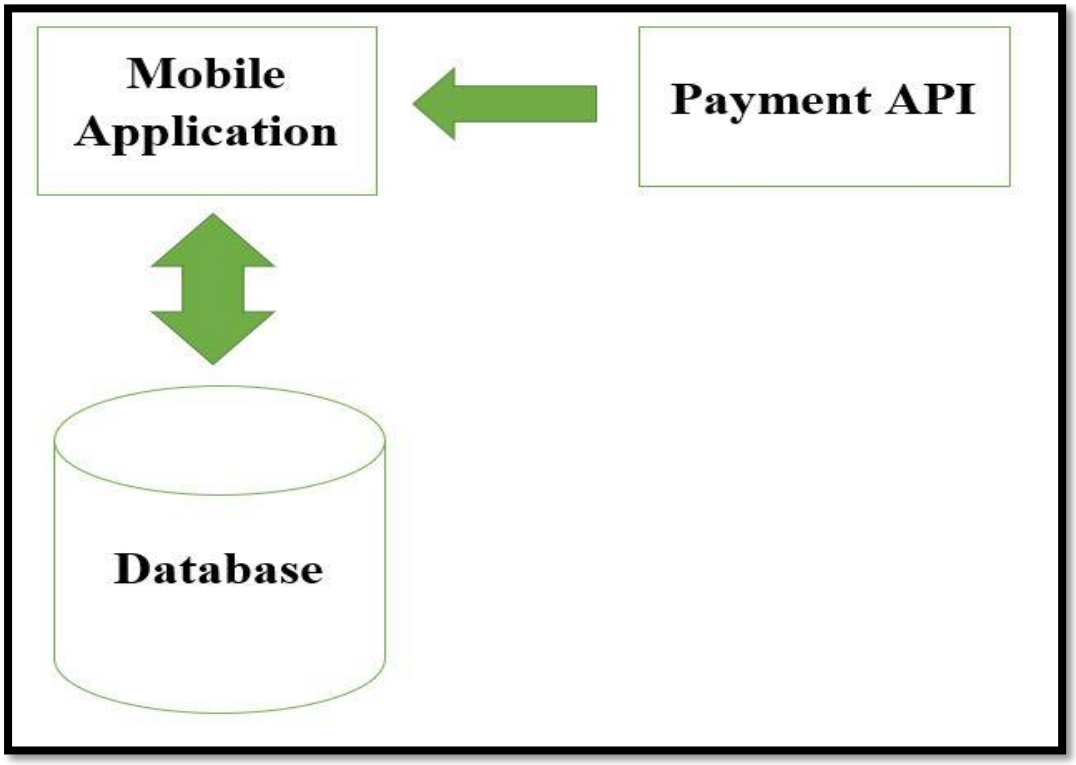

Figure 30: System Architecture

Our system is based on 2-tier architecture. The front end of the application was made by using flutter and firebase as the backend to store the data of the user. We are using the APIs of payment modules that are being hosted on their servers.

4.4. Logical Design

4.4.1 Class Diagram:

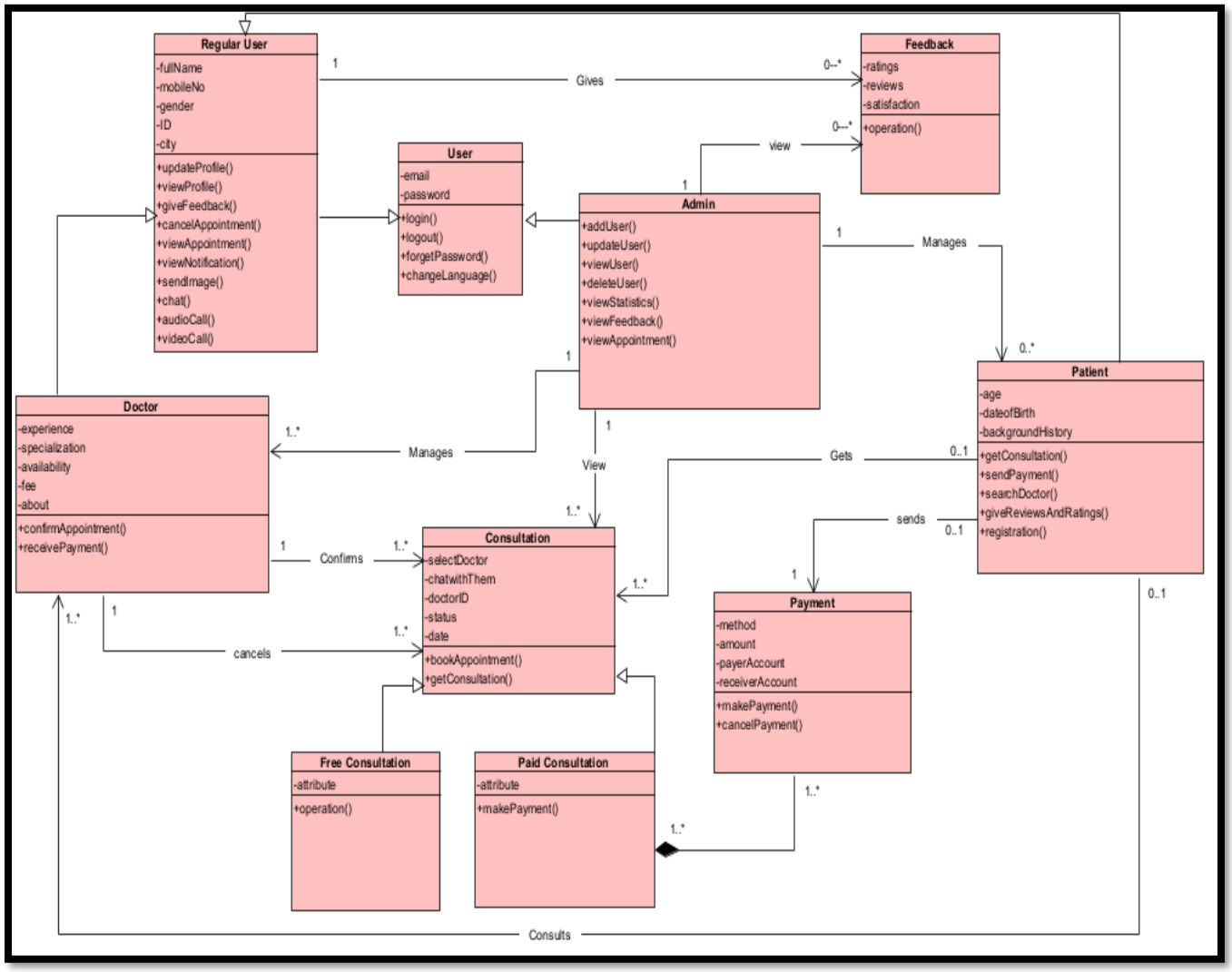

Figure 31: Class Diagram

Description:

1. Users Class:

The class of Users is generalized as Regular Users and Admin Classes. The attributes and operations of regular users and admin class are shown in figure 4.

2. Regular Users Class:

The class of Users is generalized by Regular Users. Regular Users class is further generalized by doctor class and Patient class. The attributes and operations of regular users are shown in figure 4.

3. Doctor Class:

Regular users are generalized doctor class. The attributes and operations of the doctor class are shown in figure 4. The doctor class is responsible for confirming or rejecting the appointment request. The doctor class can also receive payments after confirming appointments that are sent by the patients.

4. Patient Class:

Regular users are generalized by Patient class. The attributes and operations of patient class are shown in figure 4. The patient class can get consultations and send payments in case of paid consultation.

5. Admin Class:

Users are generalized by the Admin class. The attributes and operations of the admin class are shown in figure 4. The admin class is responsible for managing patients and doctors. The admin class can also view consultations.

6. Consultation Class:

The attributes and operations of the consultation class are shown in figure 4. The Consultation class is responsible for providing consultations to the patients. The doctor can confirm or reject consultation requests.

7. Payment Class:

The attributes and operations of the payment class are shown in figure 4. The payment class is responsible for receiving and sending payments in case of paid consultations.

8. Free Consultation:

Consultation class is generalized by free consultation. The patient has the option of getting free consultations.

9. Paid Consultation:

Consultation class is generalized by Paid consultation. The attributes and operations of paid consultation class are shown in figure 4. The patient has the option of getting paid consultation, but it is mandatory for a patient to first send payment before getting paid consultation.

10. Feedback class:

Patient and doctor can give their feedback to the admin and the admin can view their feedback.

4.5. Dynamic View

4.5.1. Sequence Diagram:

4.5.1.1. Admin Module:

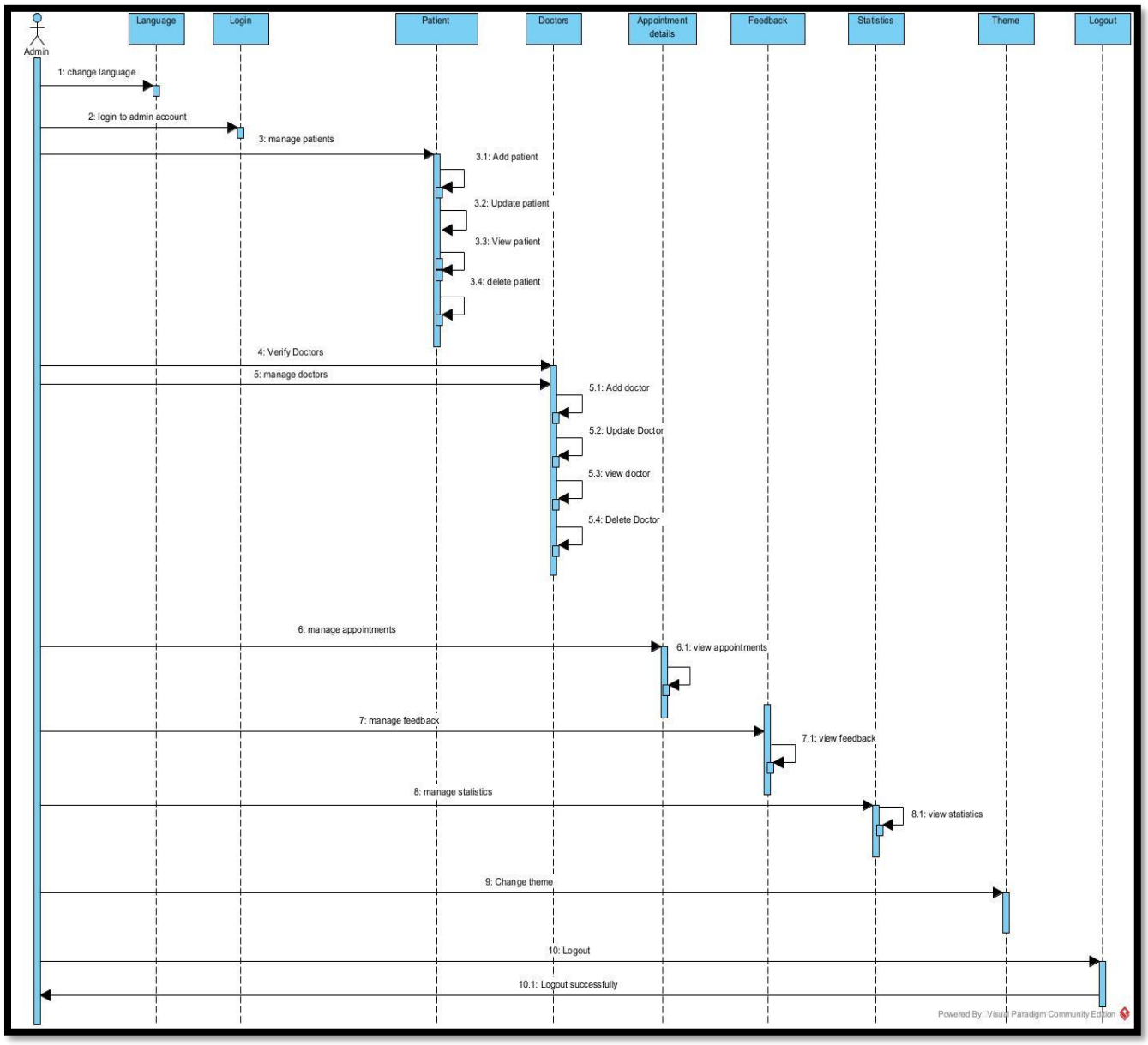

Figure 32: Sequence Diagram of Admin Module

Description:

The sequence diagram of the admin module represents the interaction among a set of objects that participated in a collaboration. The admin module has 9 lifelines that are language, patient, doctor, login, appointment details, feedback, statistics, theme, and logout. The admin can log in to his account, manage users, manage appointments, change themes, change language, manage feedback, manage statistics, and log out from his account.

4.5.1.2. Patient Module

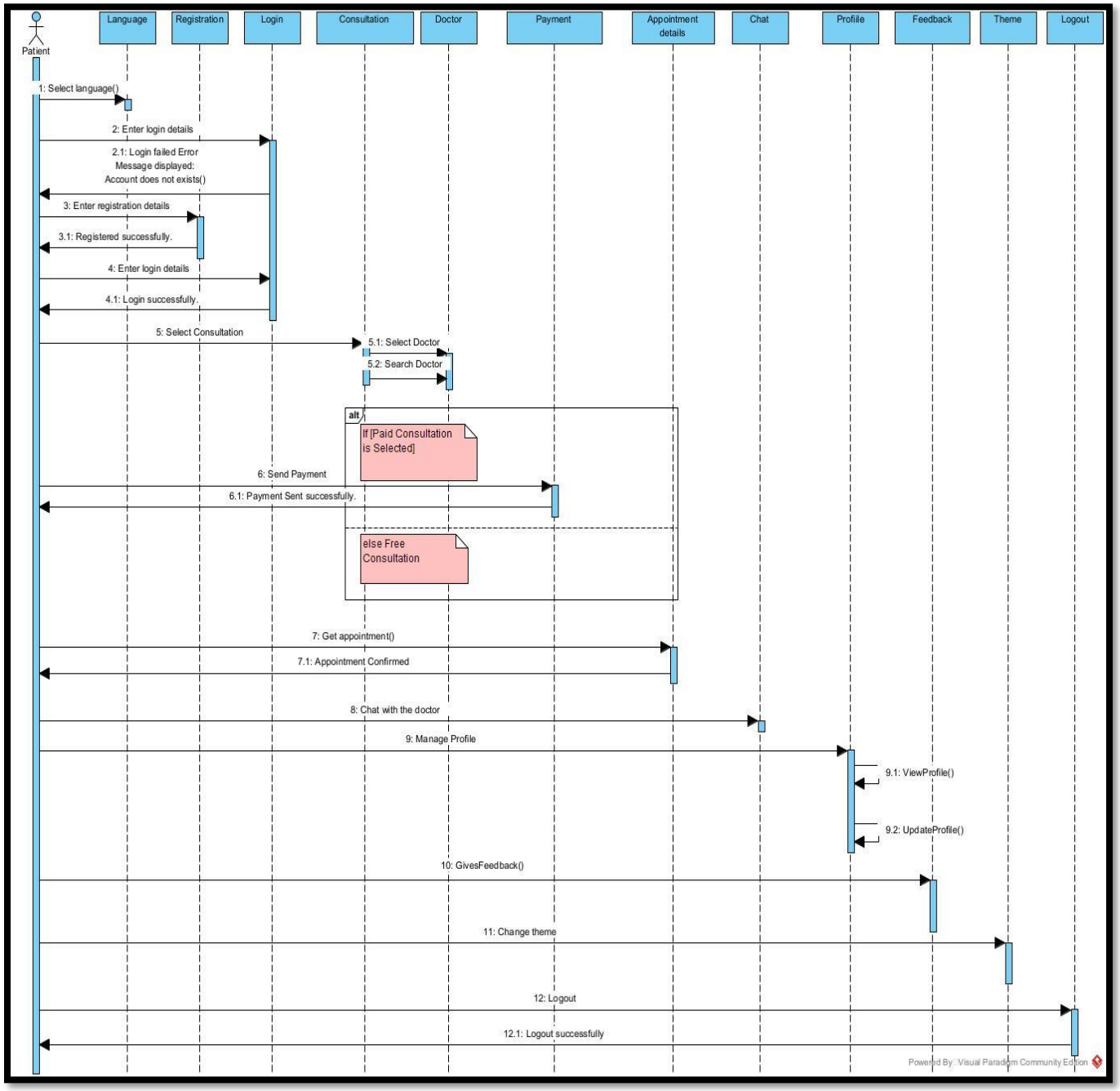

Figure 33: Sequence Diagram of Patient Module

Description:

The sequence diagram of the patient module represents the interaction among a set of objects that participated in a collaboration. The patient module has 12 lifelines. The patient can log in to his account, perform certain operations, and then log out from his account.

4.5.1.3. Doctor Module

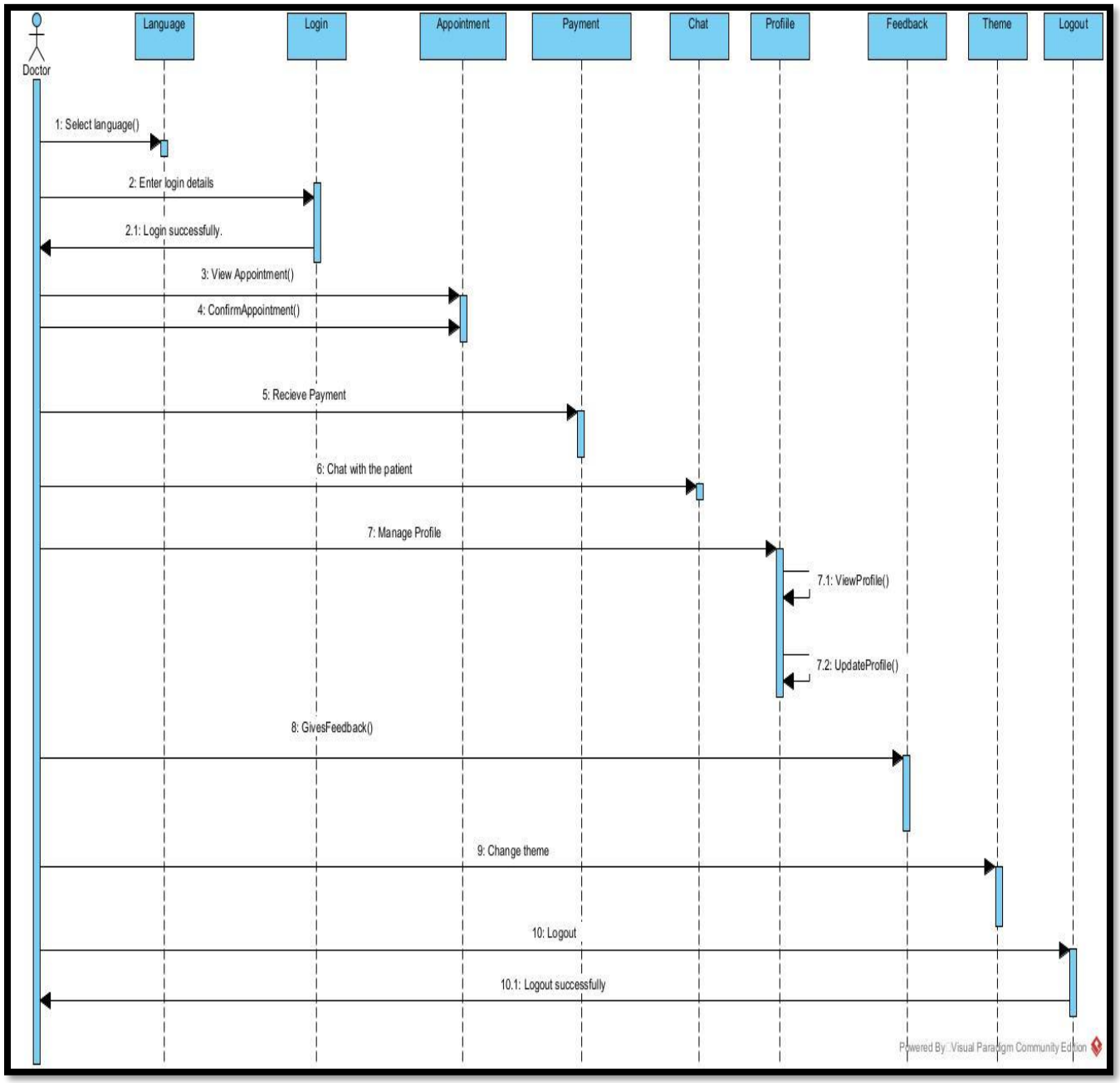

Figure 34: Sequence Diagram of Doctor Module

Description:

The sequence diagram of the doctor module represents the interaction among a set of objects that participated in a collaboration. The doctor module has 9 lifelines. The doctor can log in to his account, perform certain operations, and then log out from his account.

4.5.1.4. Patient Login

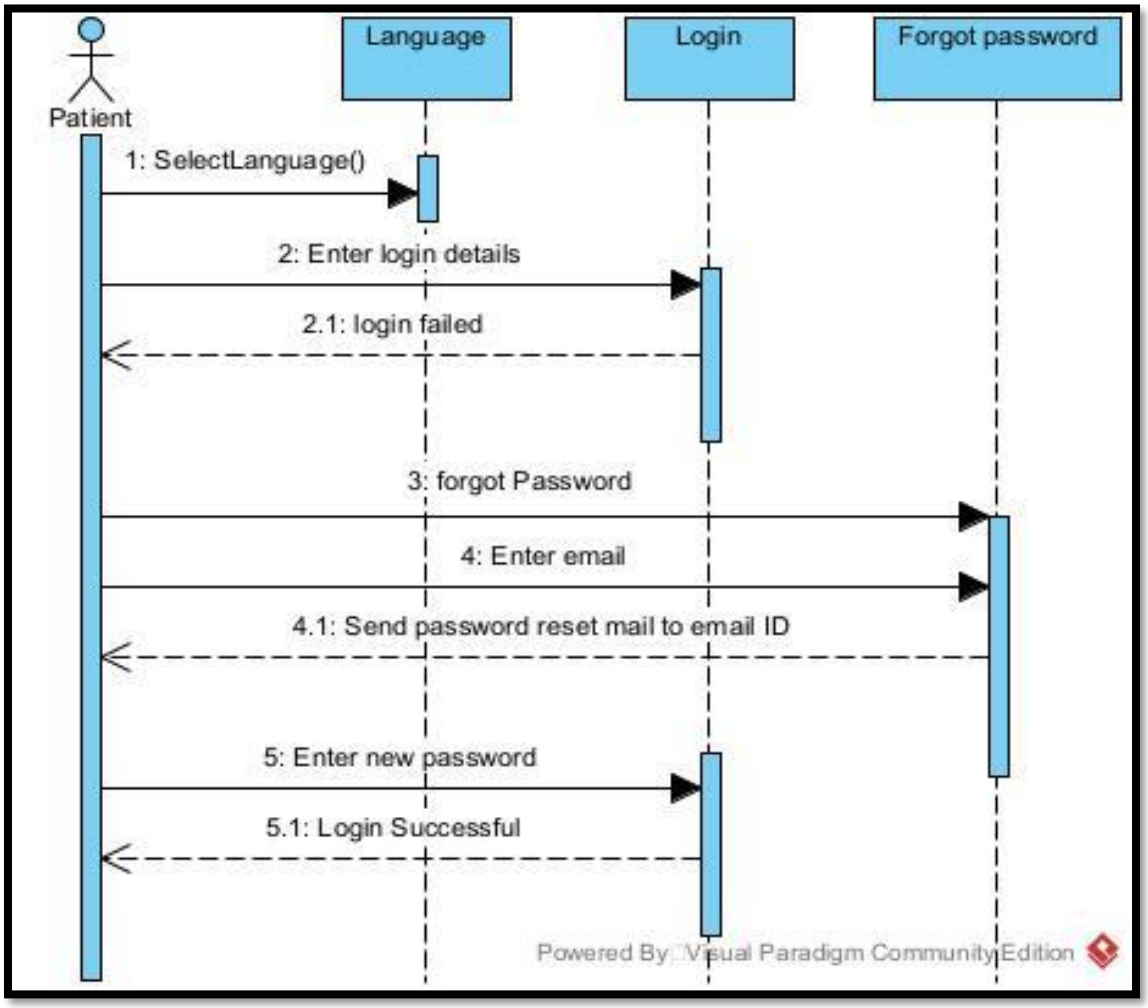

Figure 35: Sequence Diagram of Patient Login

Description:

The sequence diagram of the patient login represents the interaction among a set of objects that participated in a collaboration. The patient login has 3 lifelines. The patient can select a language and add login details if the login fails and the patient forgets his password, he can reset the password by entering an email and resetting the password. The patient will be able to successfully log in to his account with a new password.

4.5.2. Activity Diagram:

4.5.2.1. Admin Login

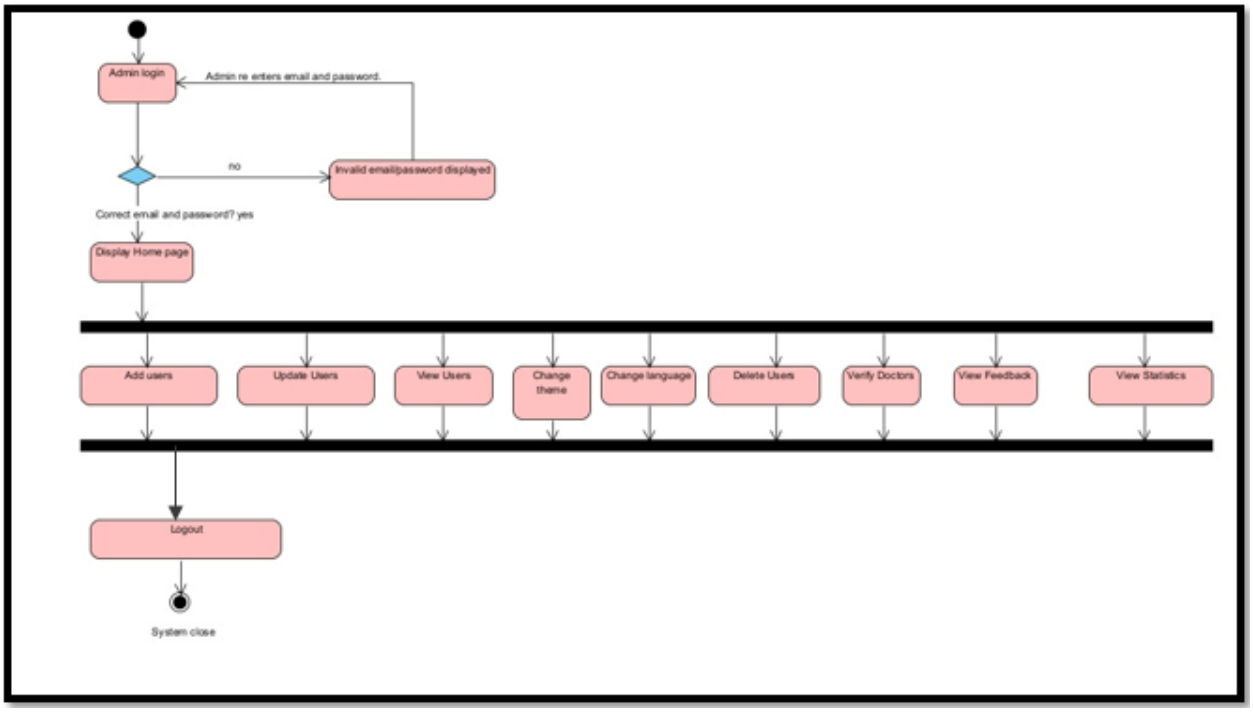

Figure 36: Activity Diagram of Admin Login

Description:

This is the admin activity in which the admin will login to his account, in case of invalid login details admin, will re-enter the correct login details. If the email and password are correct the home page will be displayed in front of the admin through which he can see the statistics, manage users, view feedback, and view appointment details.

4.5.2.2. Delete User by Admin

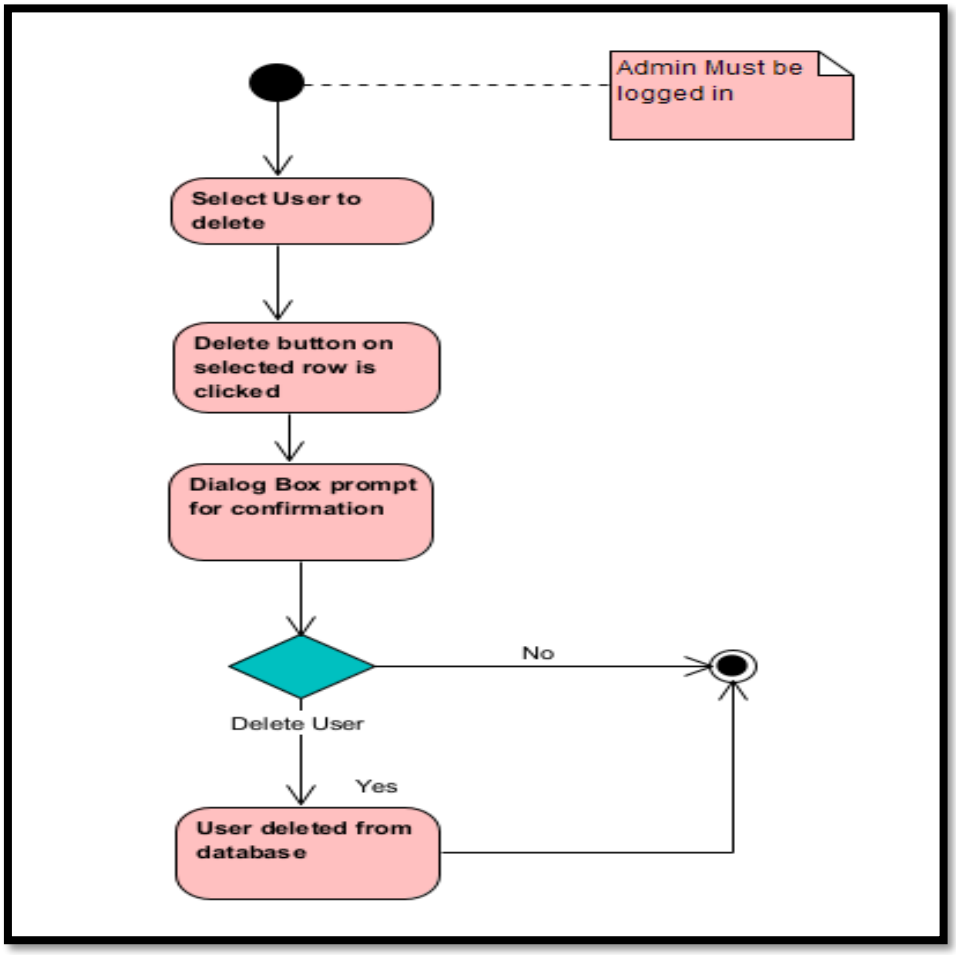

Figure 37: Activity Diagram of Delete User by Admin

Description:

To delete user activity the admin must be logged in with a valid email and password. The admin can then select the user to delete, the confirmation dialog box will appear in front of the admin. If the admin will click on the delete button in the dialog box, the user will be deleted from the database.

4.5.2.3. Doctor Module

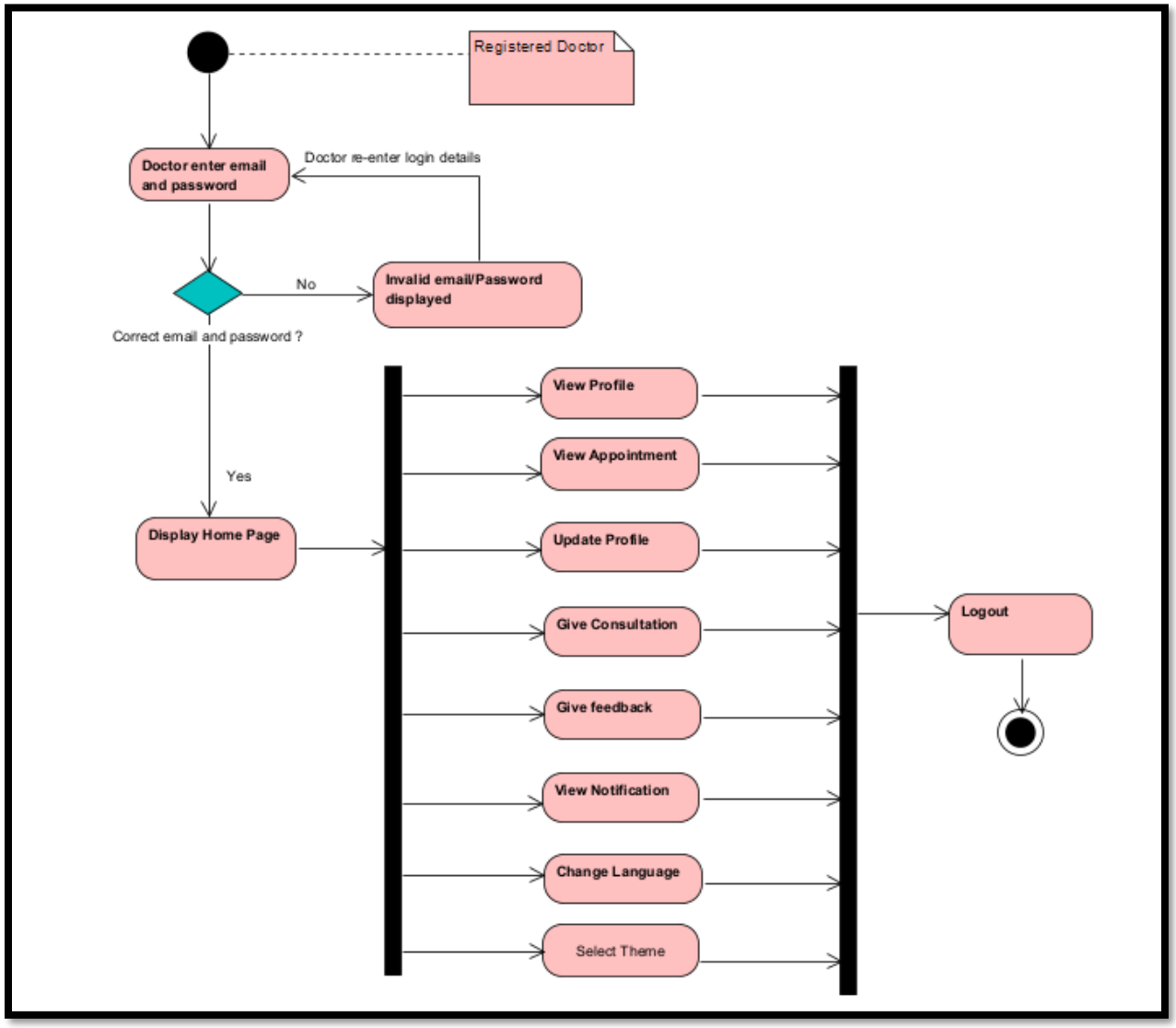

Figure 38: Activity Diagram of Doctor Module

Description:

In doctor activity, the doctor must log in using a valid email and password. The home page will be displayed in front of the doctor after successful login. The doctor can perform certain activities such as view profile, view appointment, update profile, give consultations, give feedback, view notification, and change language.

4.5.2.4. Confirm Appointment by Doctor

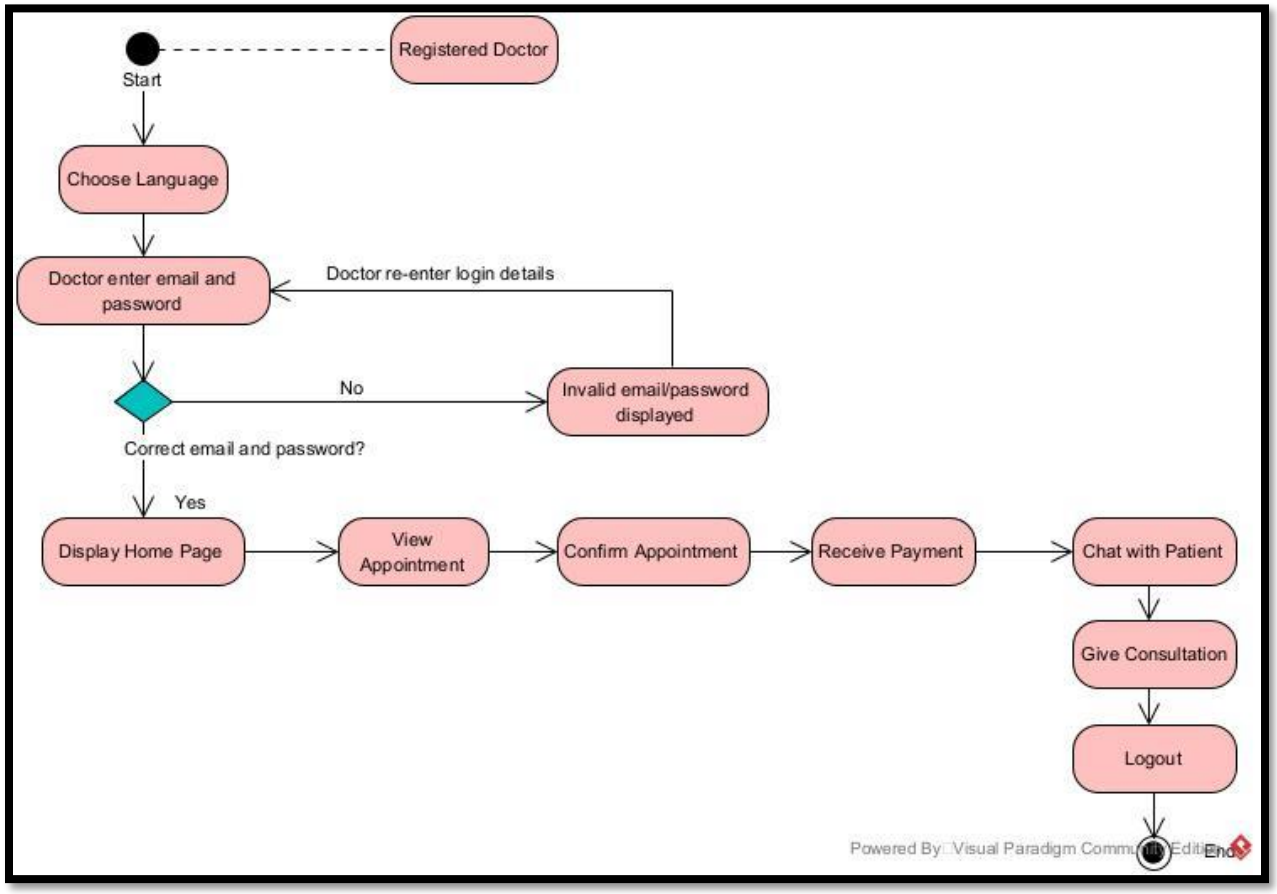

Figure 39: Activity Diagram of Confirm Appointment by Doctor

Description:

To confirm appointment activity, the registered doctor must sign in with a valid email and password. Before signing in the doctor will have the option of change language. The home page will be displayed in front of a doctor after successful login. Then the doctor can select view appointment to view the appointment request then the doctor can confirm an appointment. Once the doctor confirms an appointment, Payment will be received by the doctor. And Chat with the patient to give consultation to the patient. The doctor can then log out and finish this activity.

4.5.2.5. Patient Module

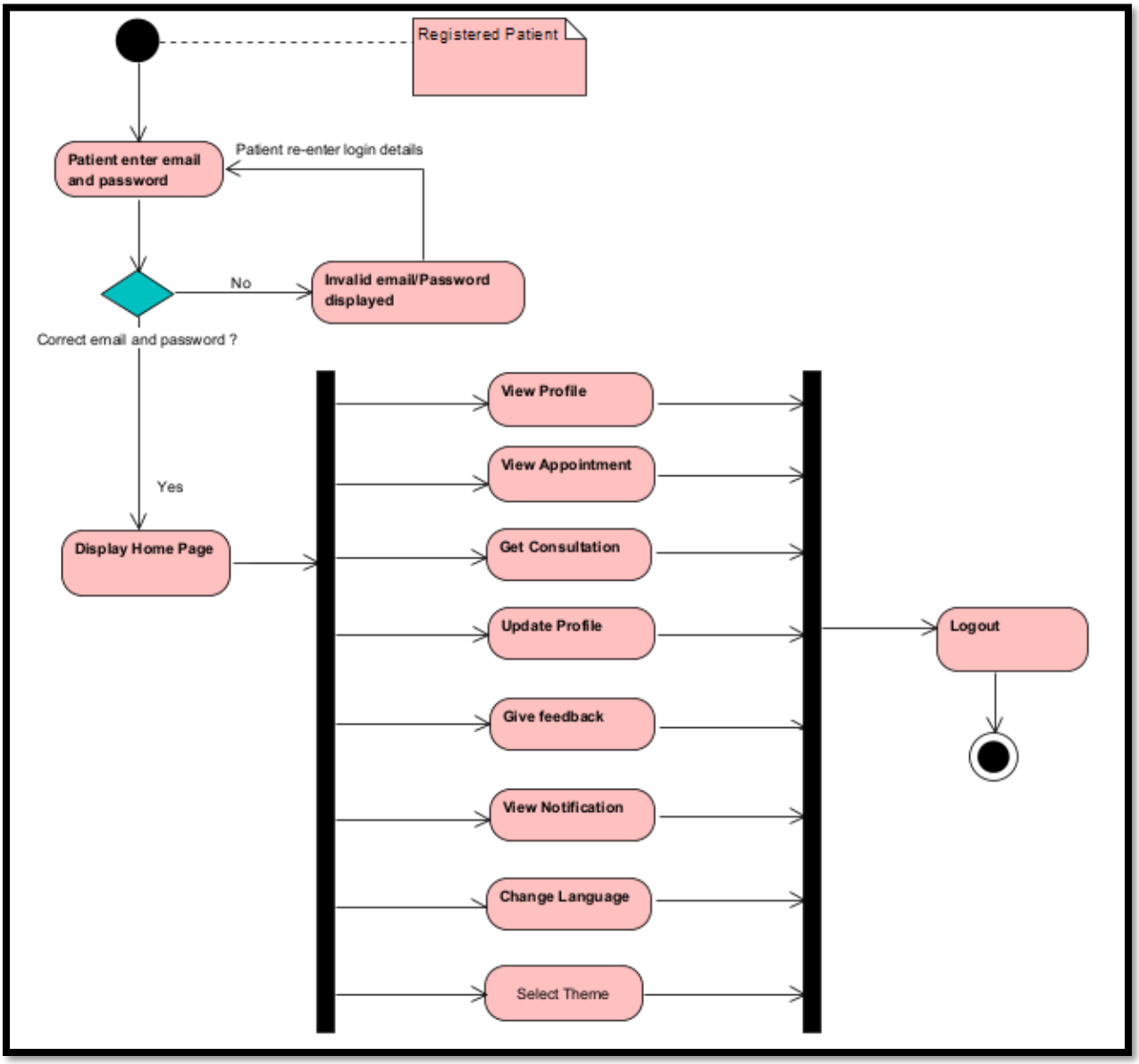

Figure 40: Activity Diagram of Patient Module

Description:

The patient can log in to his registered account using a valid email and password. After successful login, the patient can perform certain activities such as view profile, view appointments, get a consultation, update profile, give feedback, view notification, and change language.

4.5.2.6. Book Consultation by patient

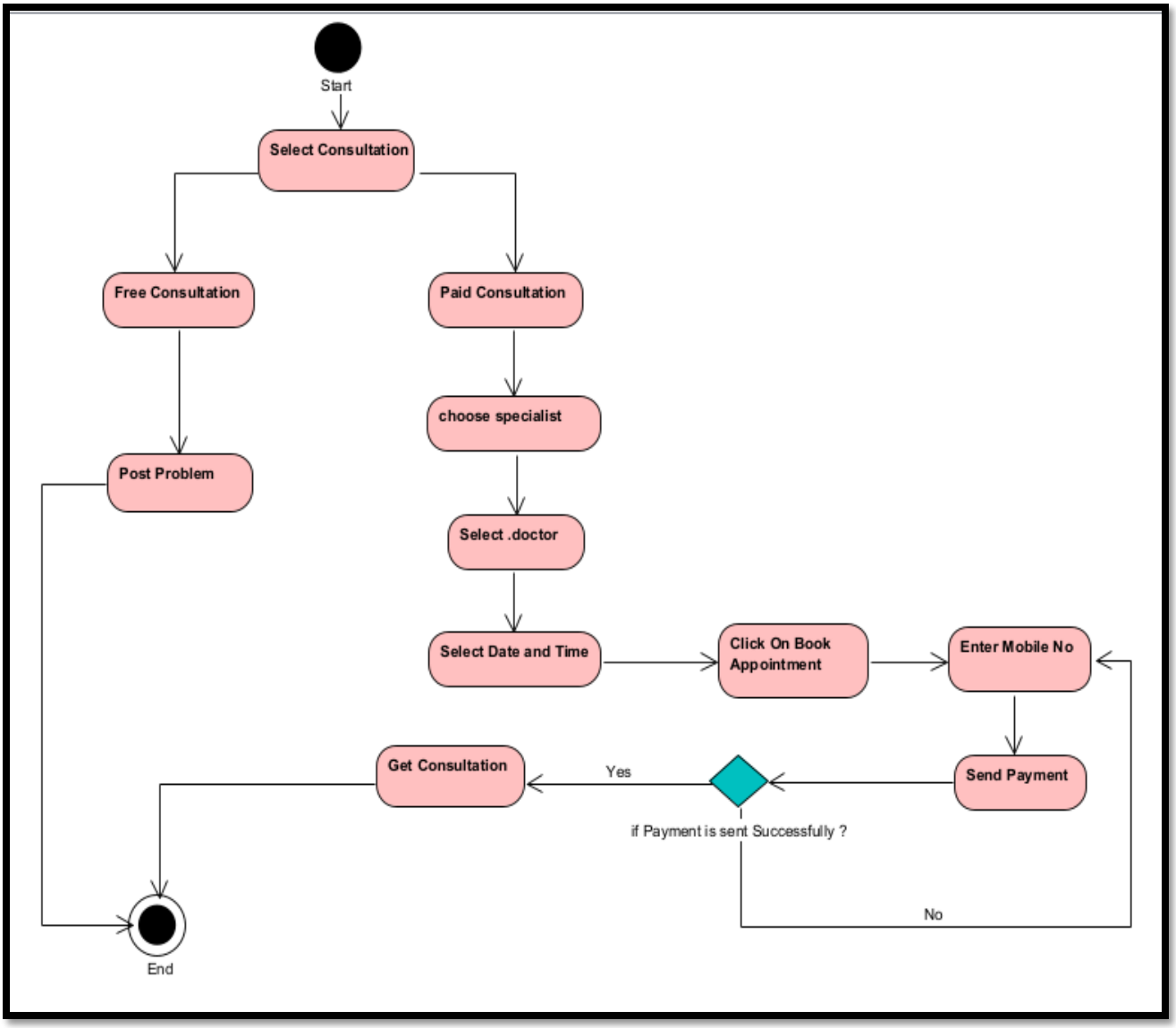

Figure 41: Activity Diagram of Book Consultation by Patient

Description:

In the book consultation activity, the patient has the option of selecting free and paid consultation. If the patient has selected the free consultation, the patient will post their problem and any doctor willingly can answer that post. If the patient has selected paid consultation the patient will first select a specialist, then select a doctor, and book an appointment after paying the fees of the doctor. And then he can get a consultation and finish this activity.

4.6. Component Design

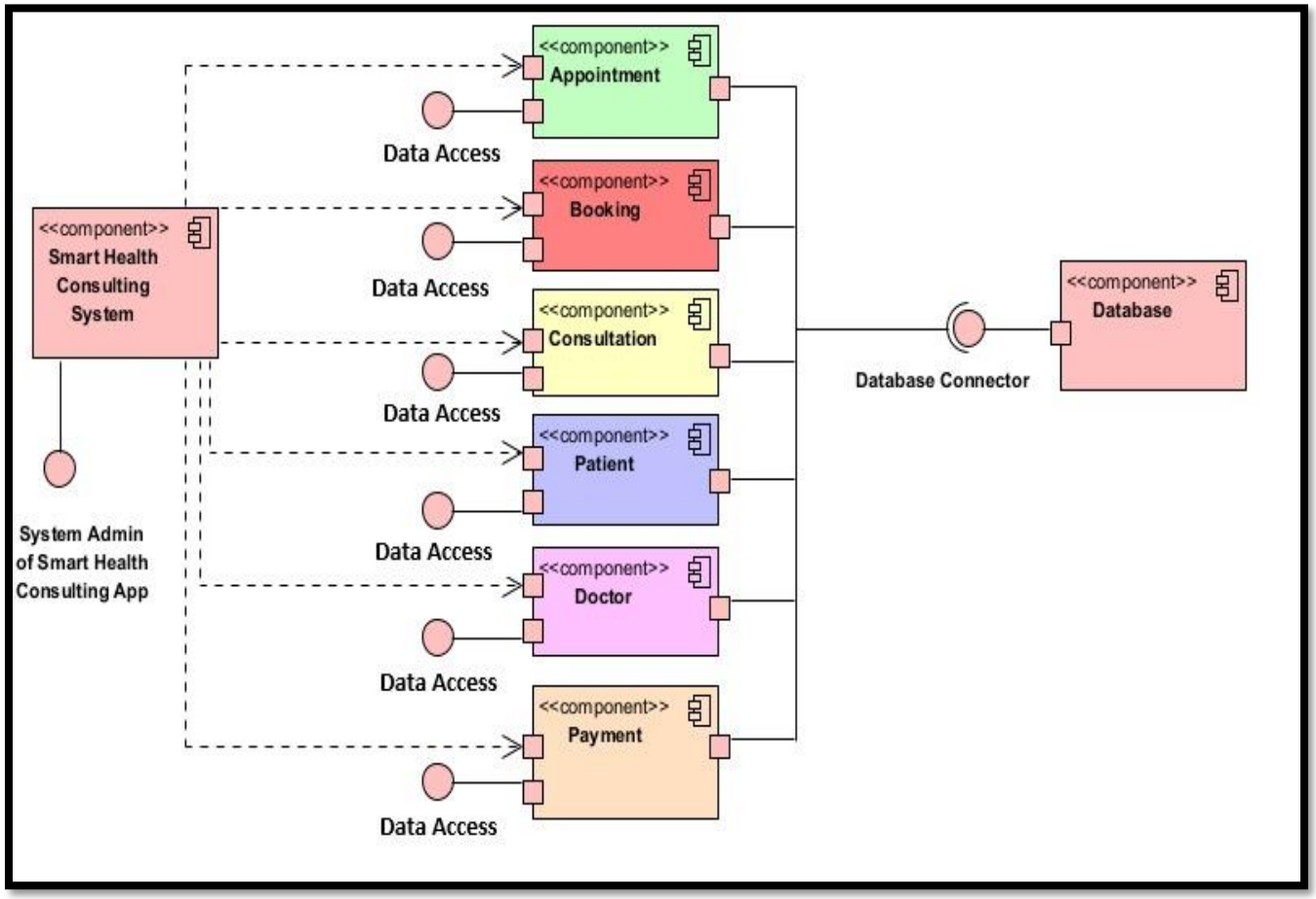

Figure 42: Component Diagram

Description:

The component diagram of our system is representing the organization and wiring of physical components. In this diagram, we have the components of appointment, booking, consultation, patient, doctor, and payment. These components are responsible for one clear aim within the entire system and only interact with other essential elements on a need-to-know basis.

4.7. Data Models

4.7.1 ER Diagram:

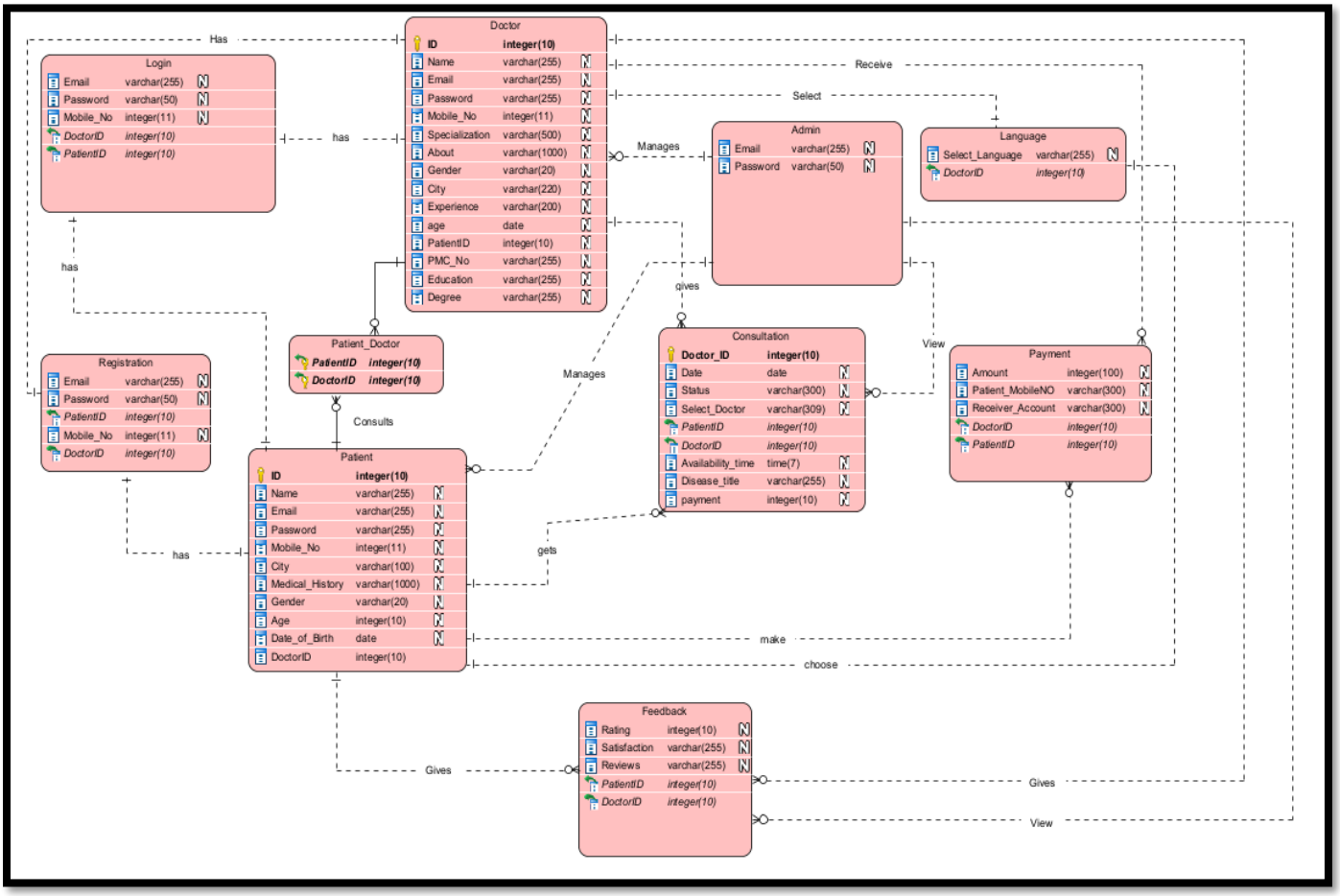

Figure 43: Entity Relationship Diagram

Description:

The above diagram depicts the relationship between various entities with various attributes. We have the entities of doctor, admin, language, login, patient, registration, feedback, Payment consultation, paid consultation, and free consultation. All these entities have different relationships like one to one, one to many, and many to many with each other.

4.8. User Interface Design

4.8.1. Screenshot of Splash Screen

Here is the first screen that will appear in front of user for a few seconds.

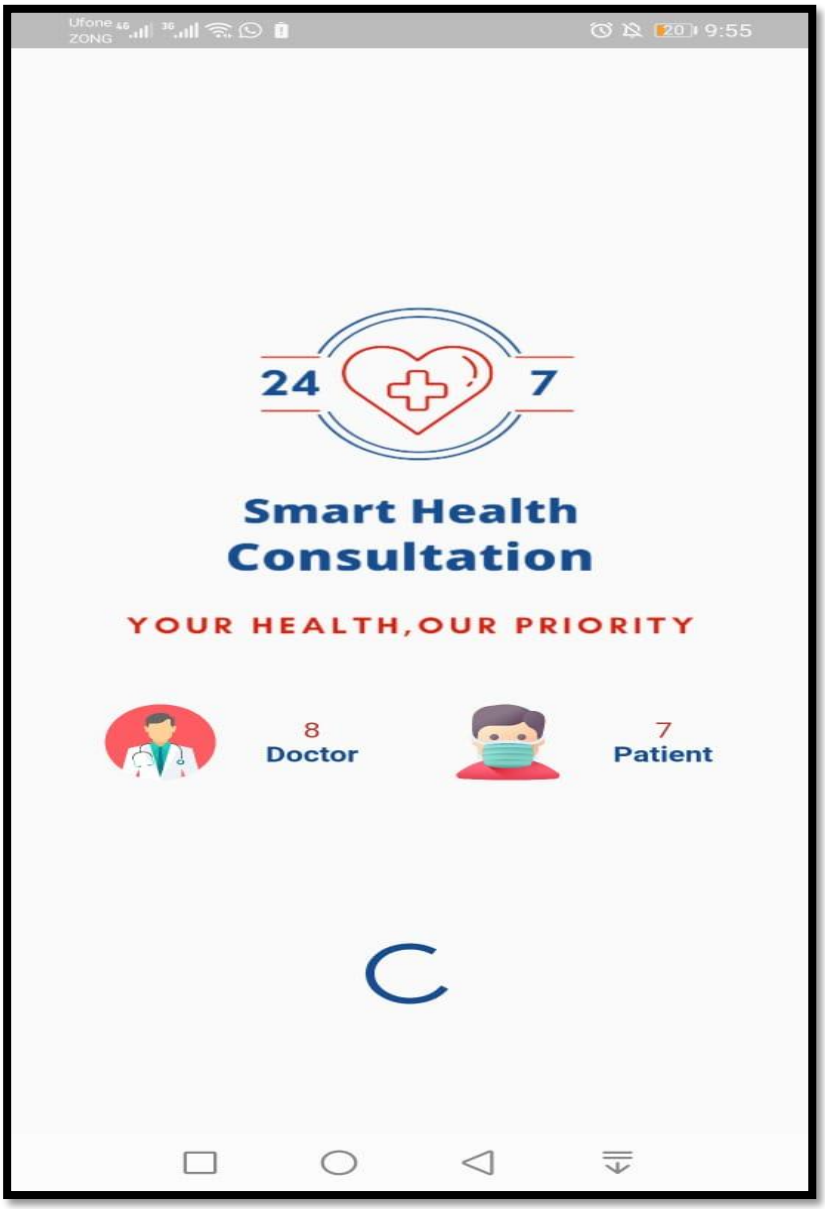

Figure 44: Screenshot of Splash screen

4.8.2. Screenshot of Change Language

After splash screen change language screen will be displayed.

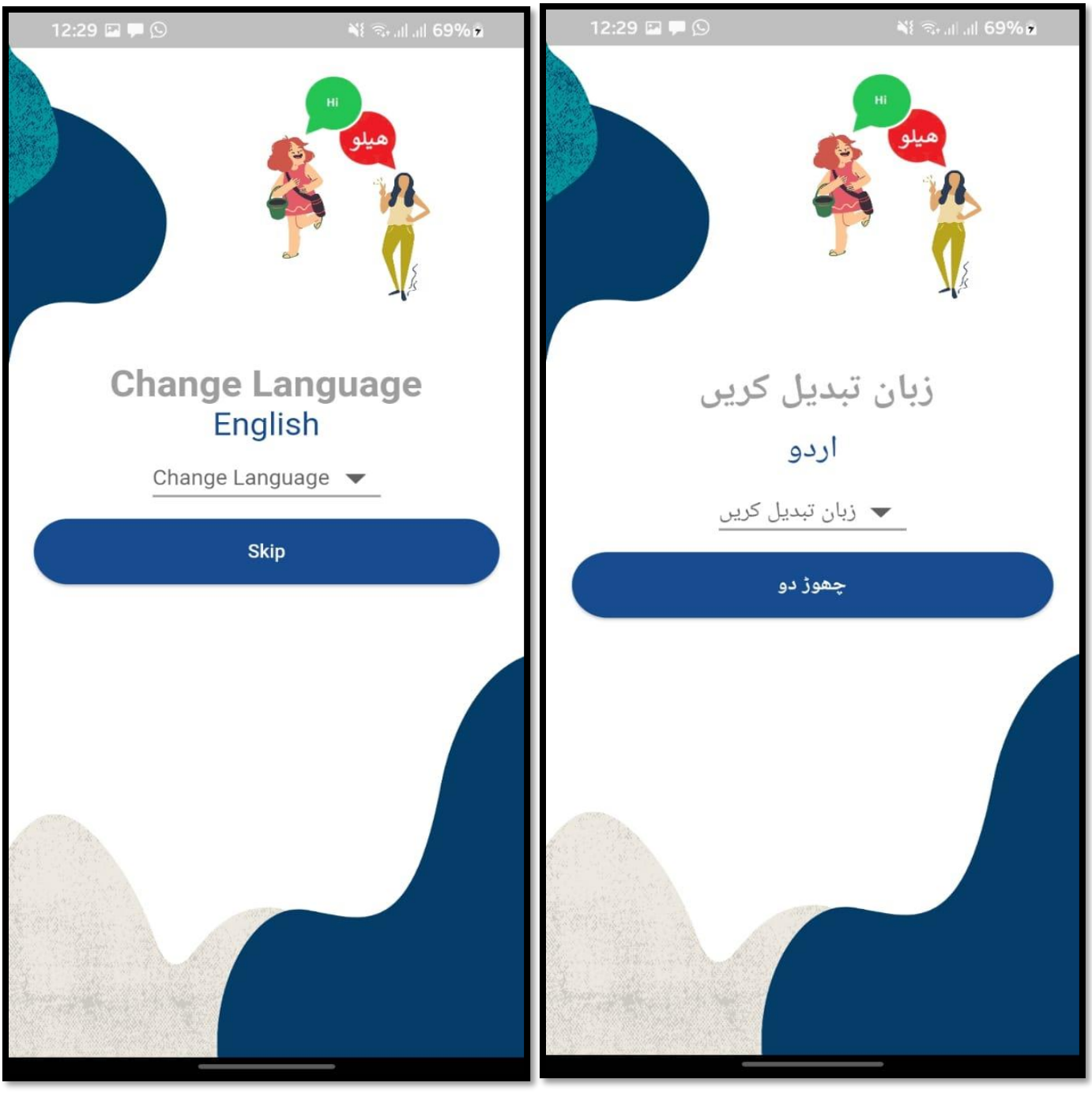

Figure 45: Screenshot of Change Language

4.8.3. Registration

The user has the option of registration by email and password, or by using a phone number and name. Users can select to register as a patient or doctor.

The user has to check the terms and conditions before clicking on the registration button otherwise error message will display.

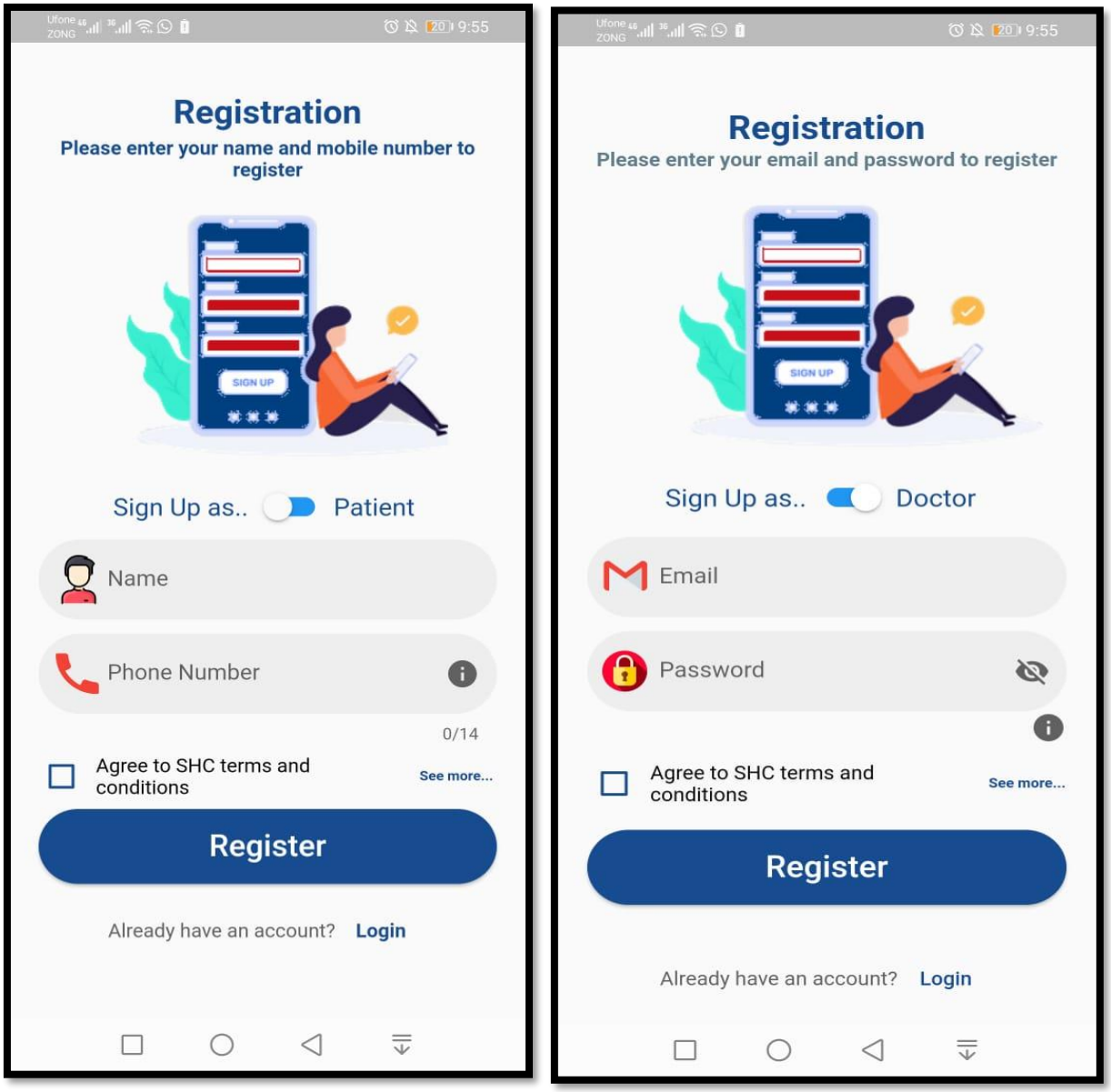

Figure 46: Screenshot of User Registration

Users receive a verification code if registered through mobile number. User can register their account through google authentication.

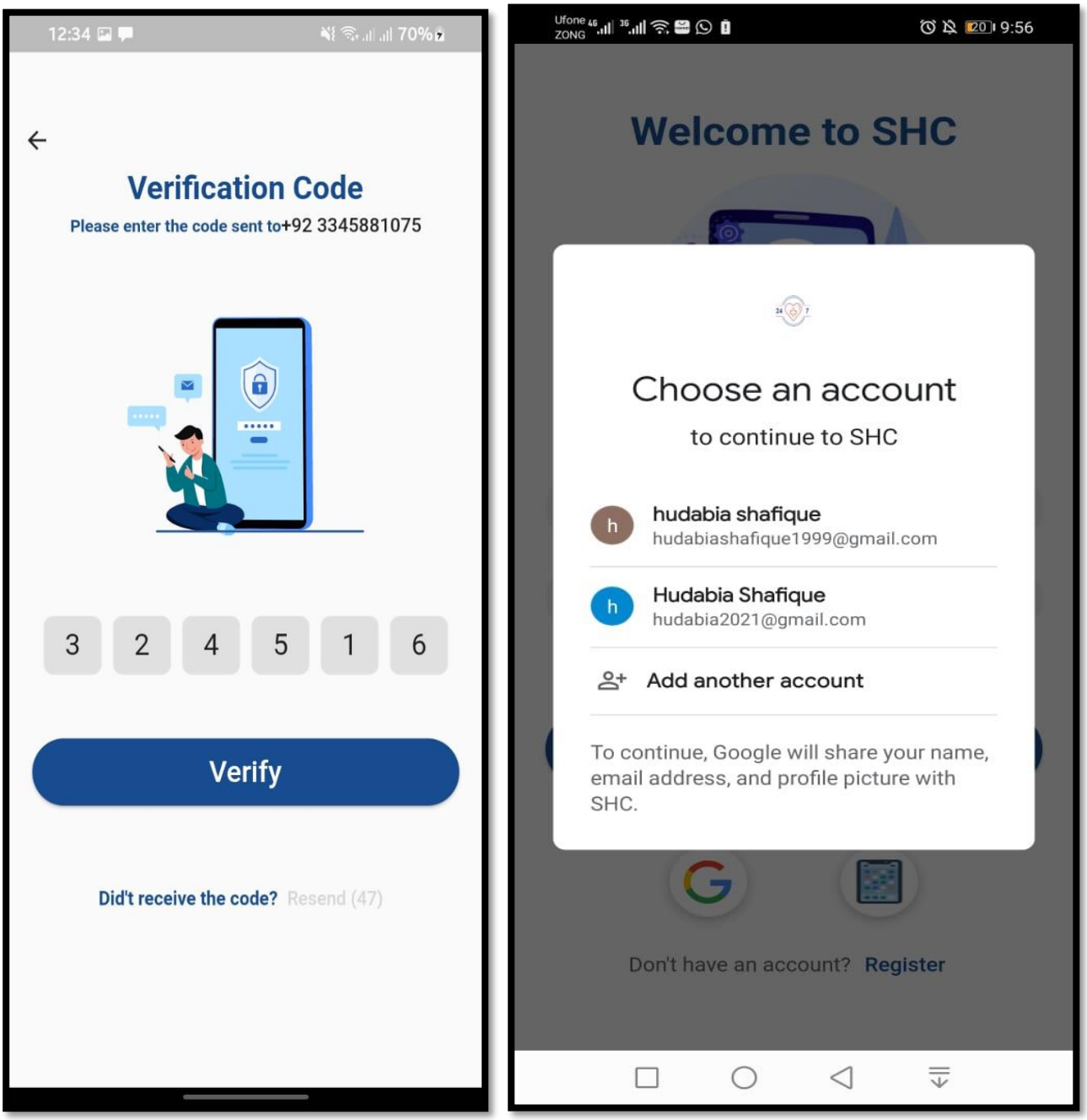

Figure 47: Screenshot of Verification Code and Google Authentication

4.8.4. Login and Reset Password

The user can login to the system, if the user forgets his password, he will have the option to select forgot password and then reset a password.

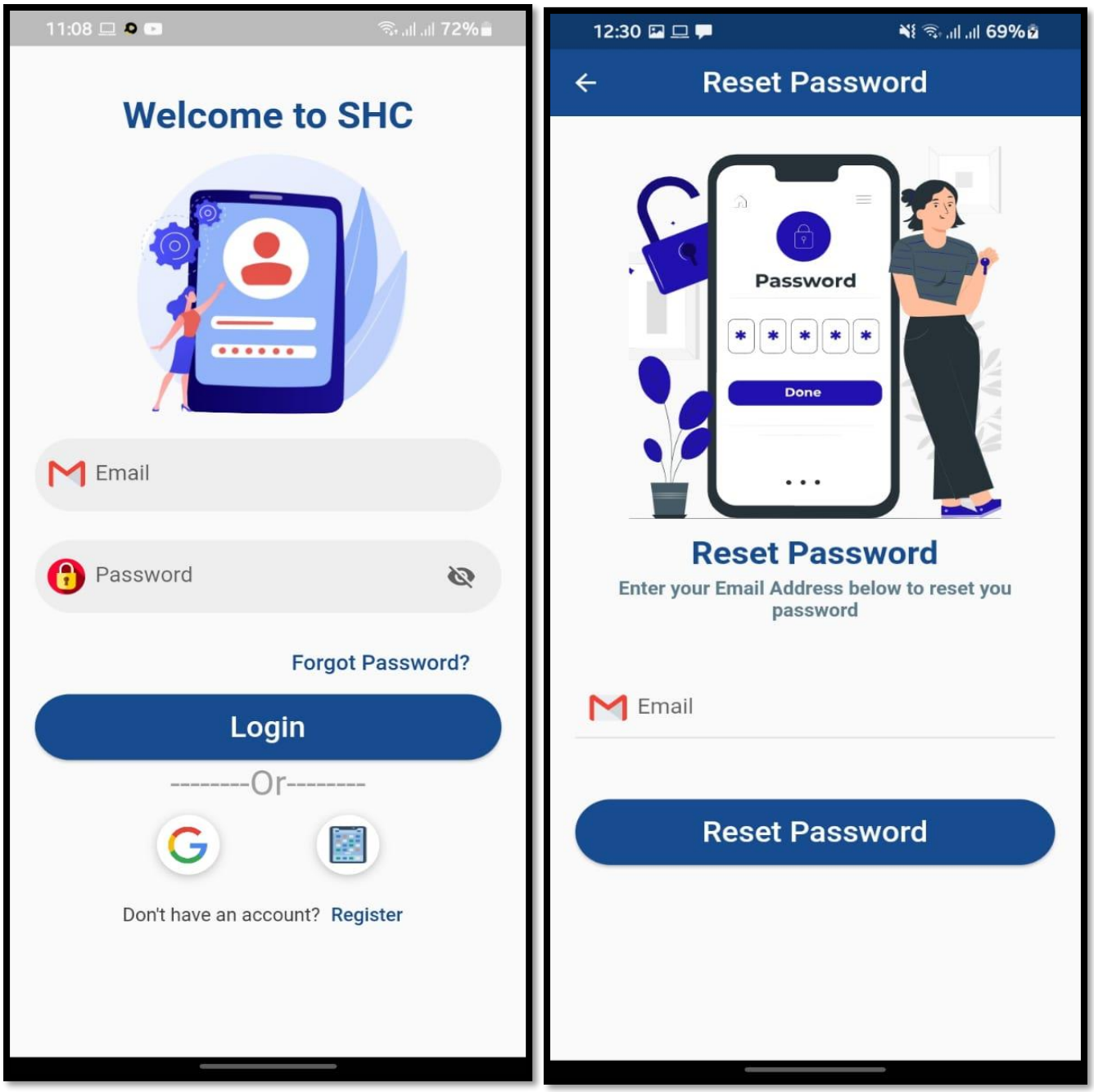

Figure 48: Screenshot of Login Screen and Reset Password

4.8.5. Patient Side

The update profile page will appear in front of the patient when he login first time in our application or if his profile is not updated so that he can update his profile.

.

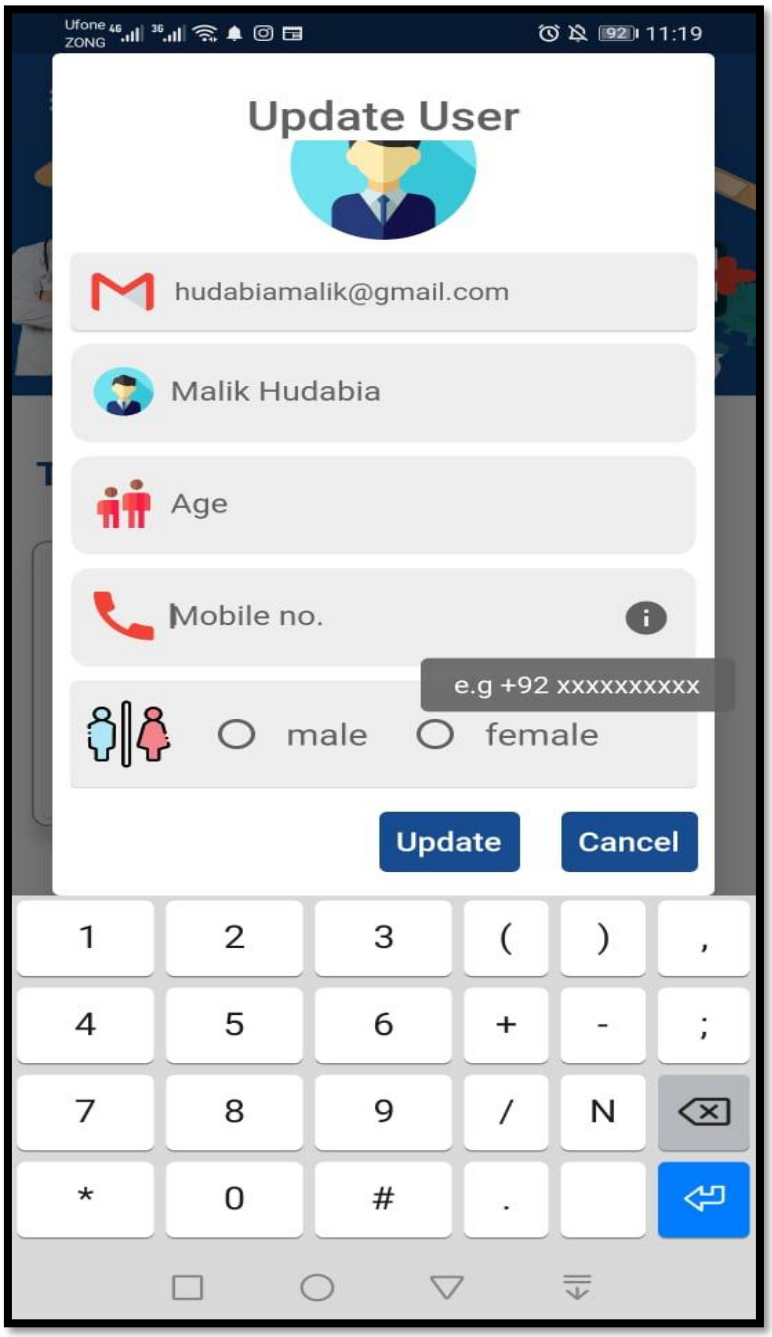

Figure 49: Screenshot of Update Profile

4.8.6. Home Page

After logging in home screen will appear in front of patient.

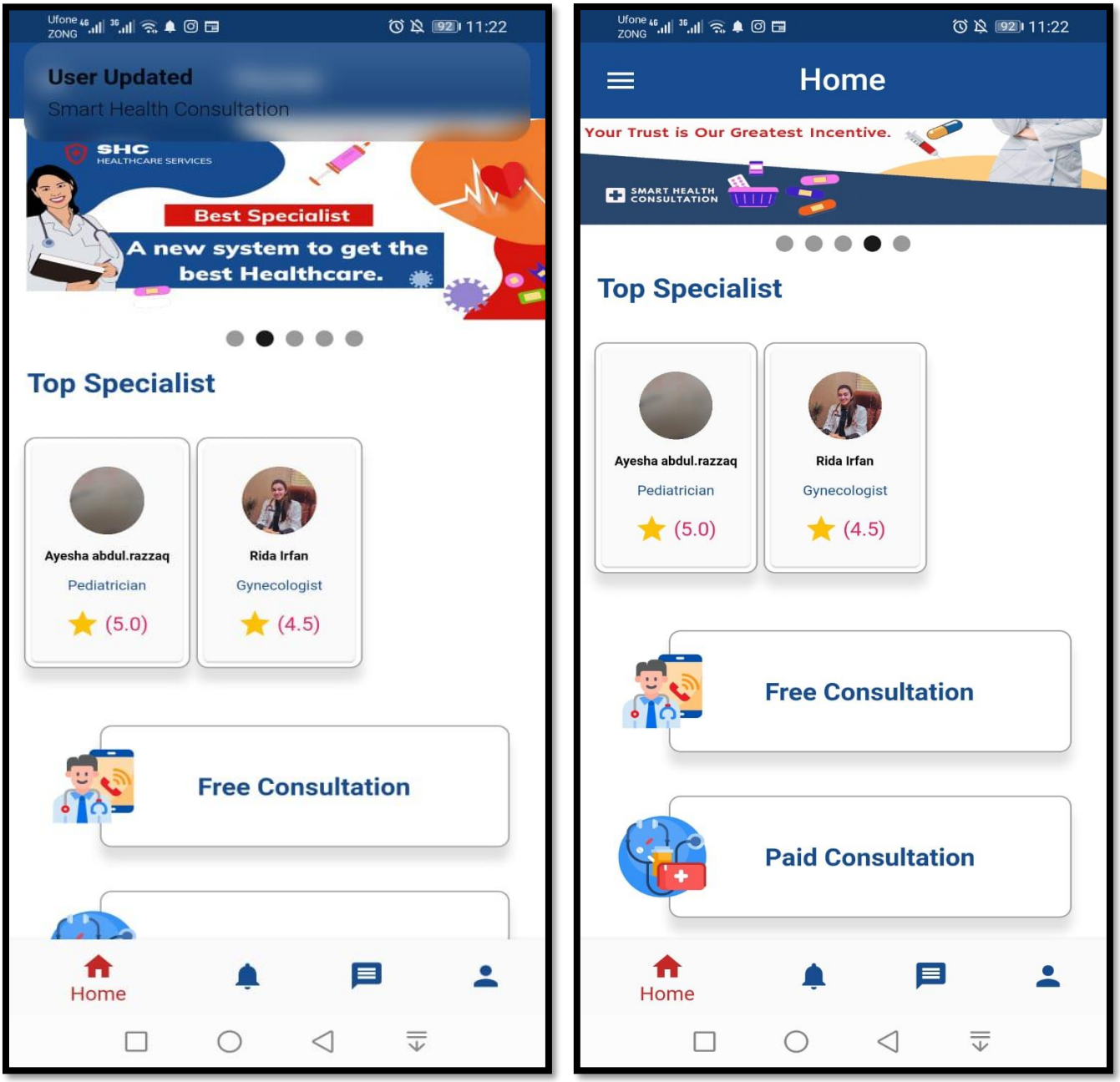

Figure 50: Screenshot of Patient Home page

4.8.7. Drawer and setting screen

Here is the interface of drawer and settings.

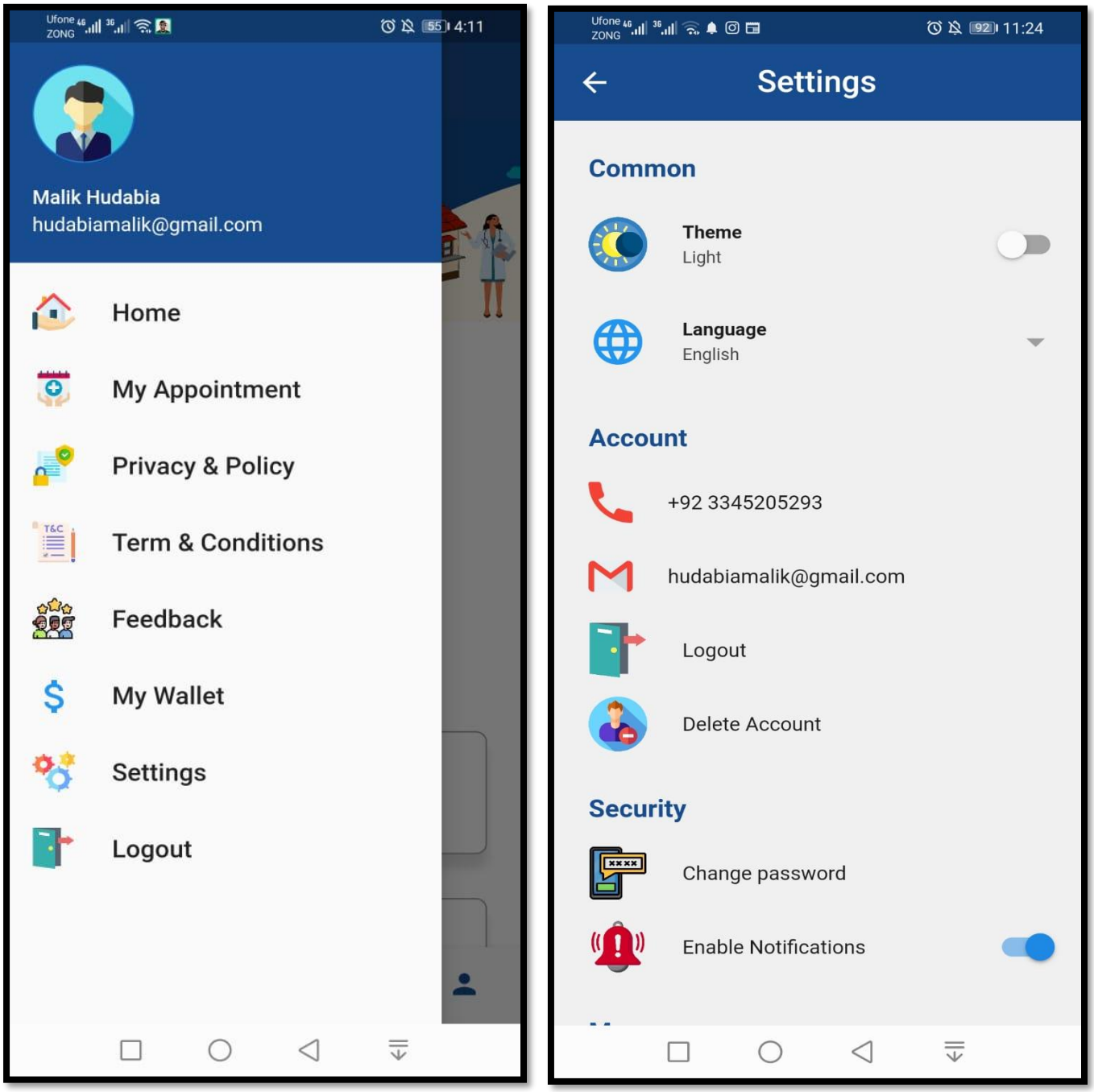

Figure 51: Screenshot of Drawer and Setting

4.8.9. Dark theme and about us

The user can also have the option to change theme in settings. The user can also select about, in which he can read about ourselves.

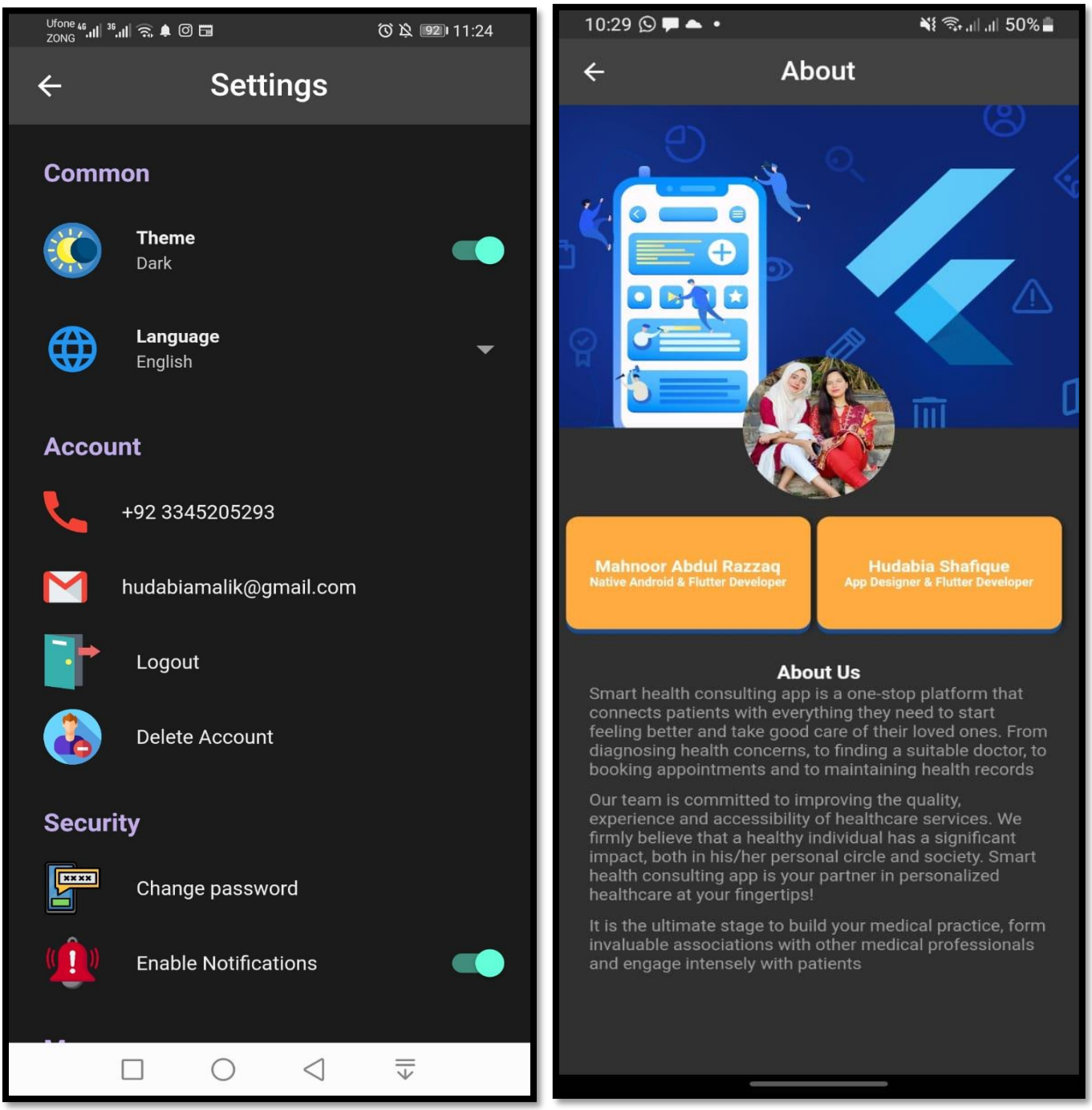

Figure 52: Screenshot of Dark Theme and About

4.8.10. Change Password and Delete Account:

In settings the user has the option to change password. The user can also delete his account permanently.

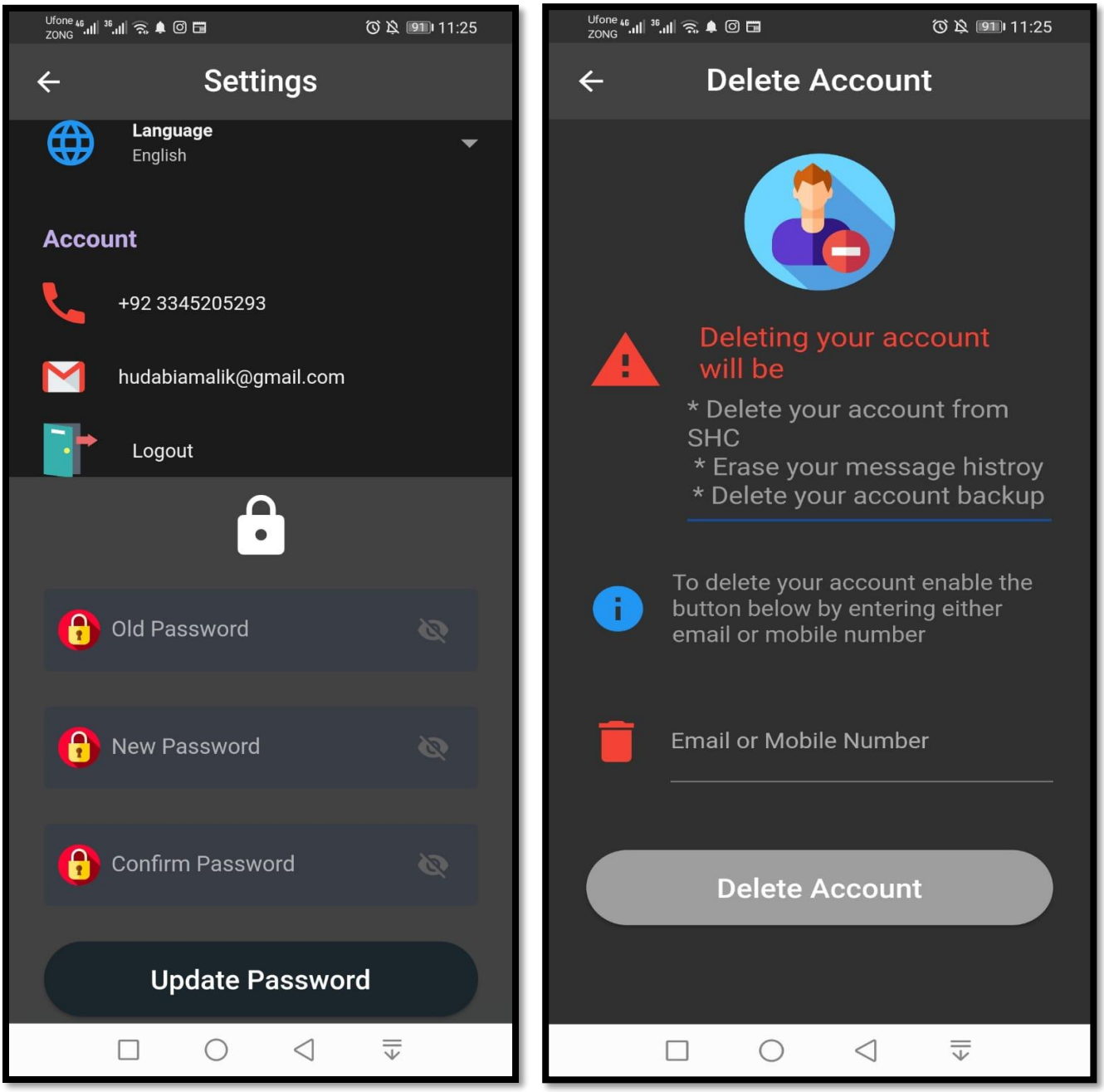

Figure 53: Screenshot of Change Password and Delete Account

4.8.11. Reviews and terms and conditions in the Urdu language

If the user has selected Urdu language and dark theme, Reviews and Terms and Conditions screens will appear.

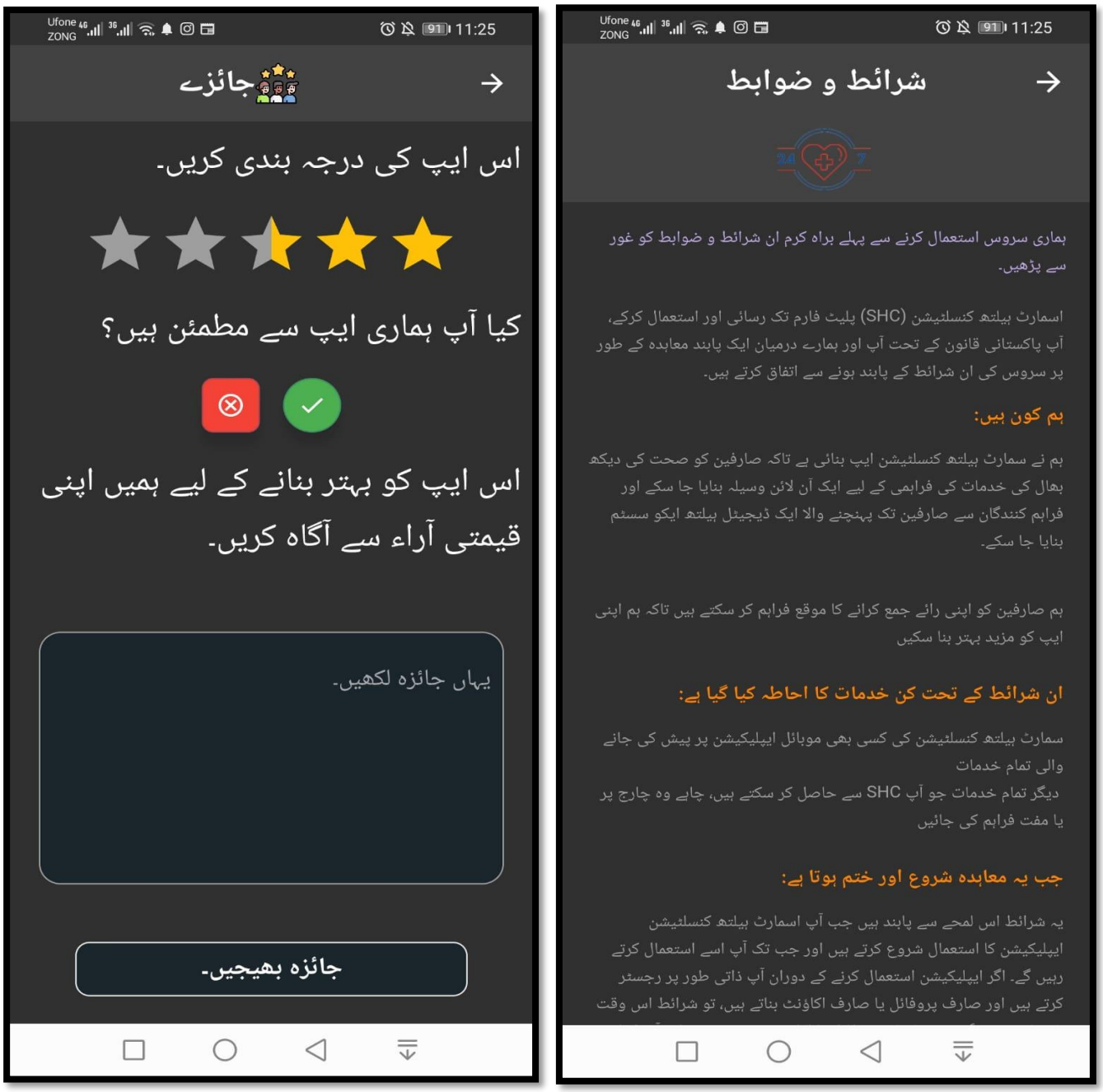

Figure 54: Screenshot of Reviews and Terms and Condition

4.8.12. Free Consultation

If the patient has selected free consultation option, he can post his problems.

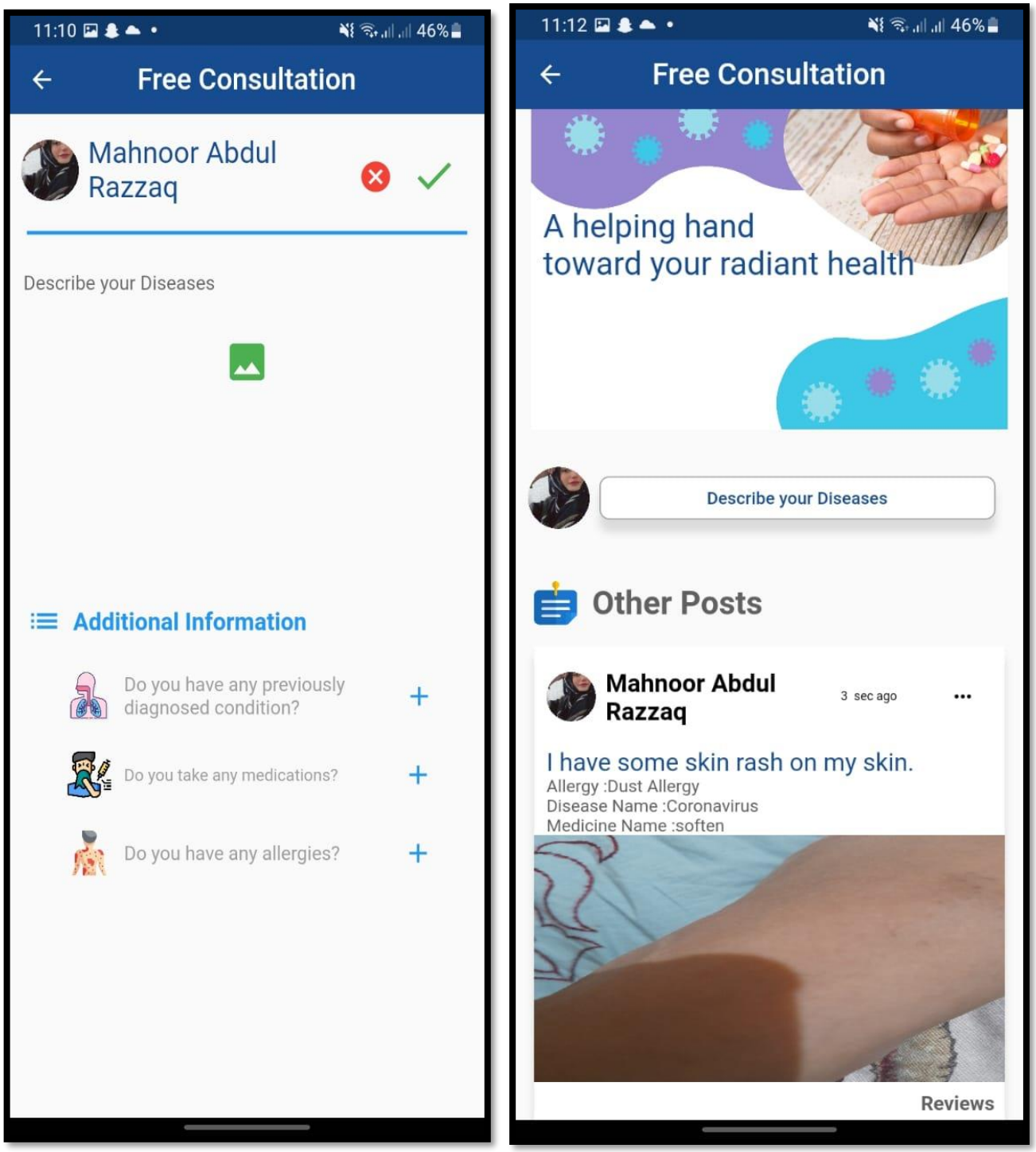

Figure 55: Screenshot of Free Consultation

4.8.13. Profile Page

Here is the profile page which can also be edited by the patient.

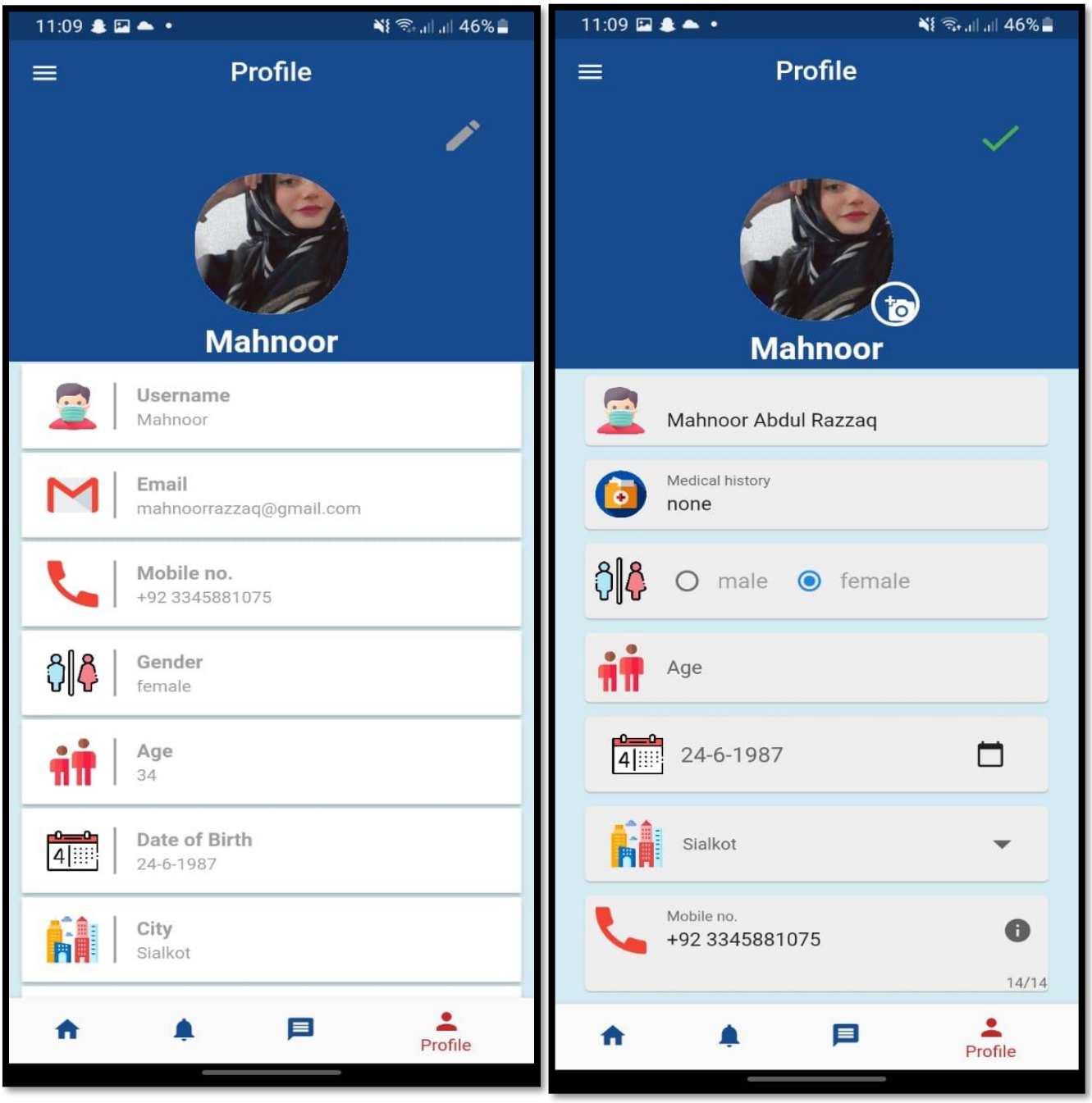

Figure 56: Screenshot of Profile Page

4.8.14. Paid Consultation

The patient will select the category of specialists.

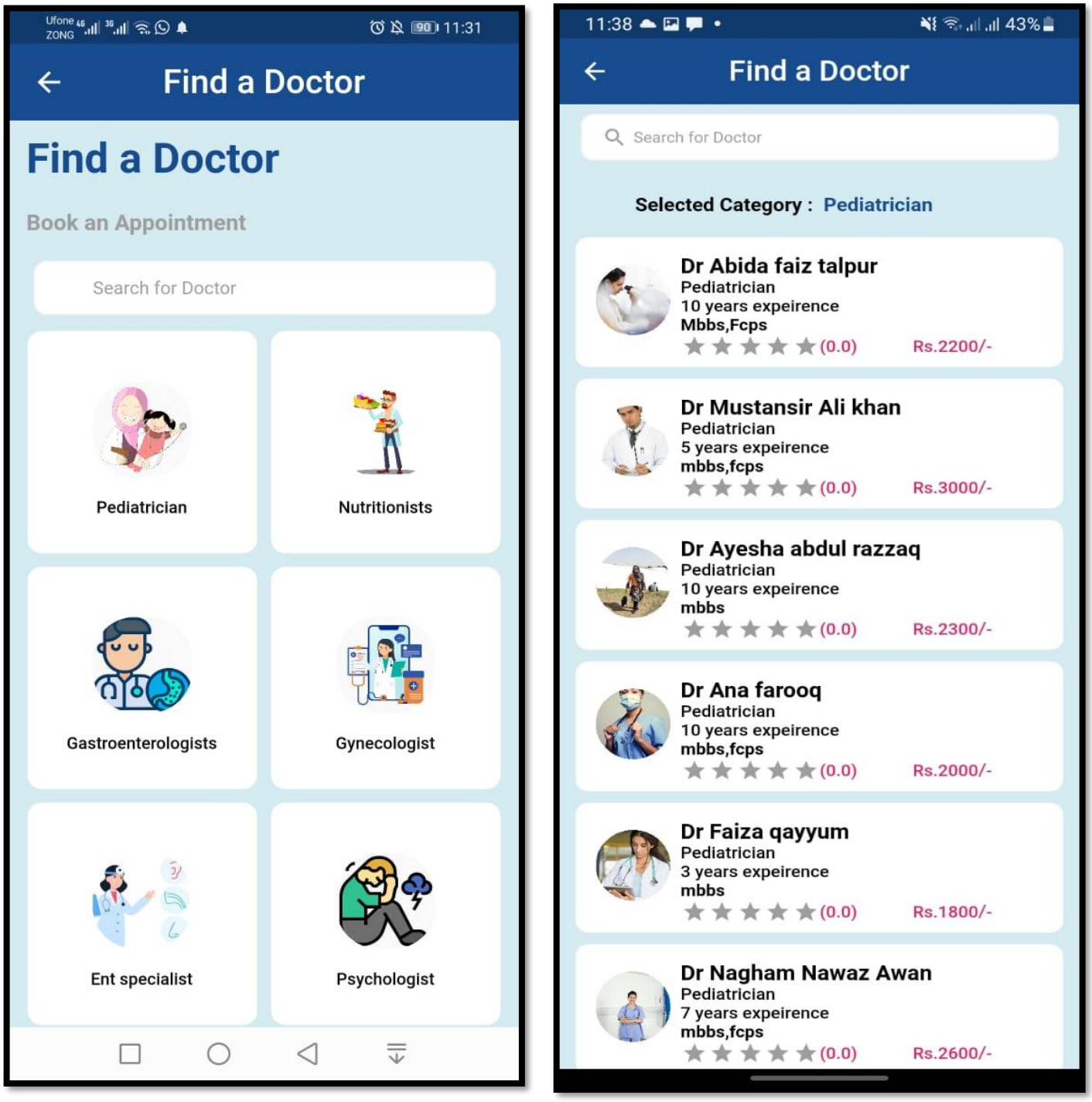

Figure 57: Screenshots of Selecting Category and Specialists

4.8.15. The patient can check doctor profile

The patient can view doctors profile and can also give reviews after consultation.

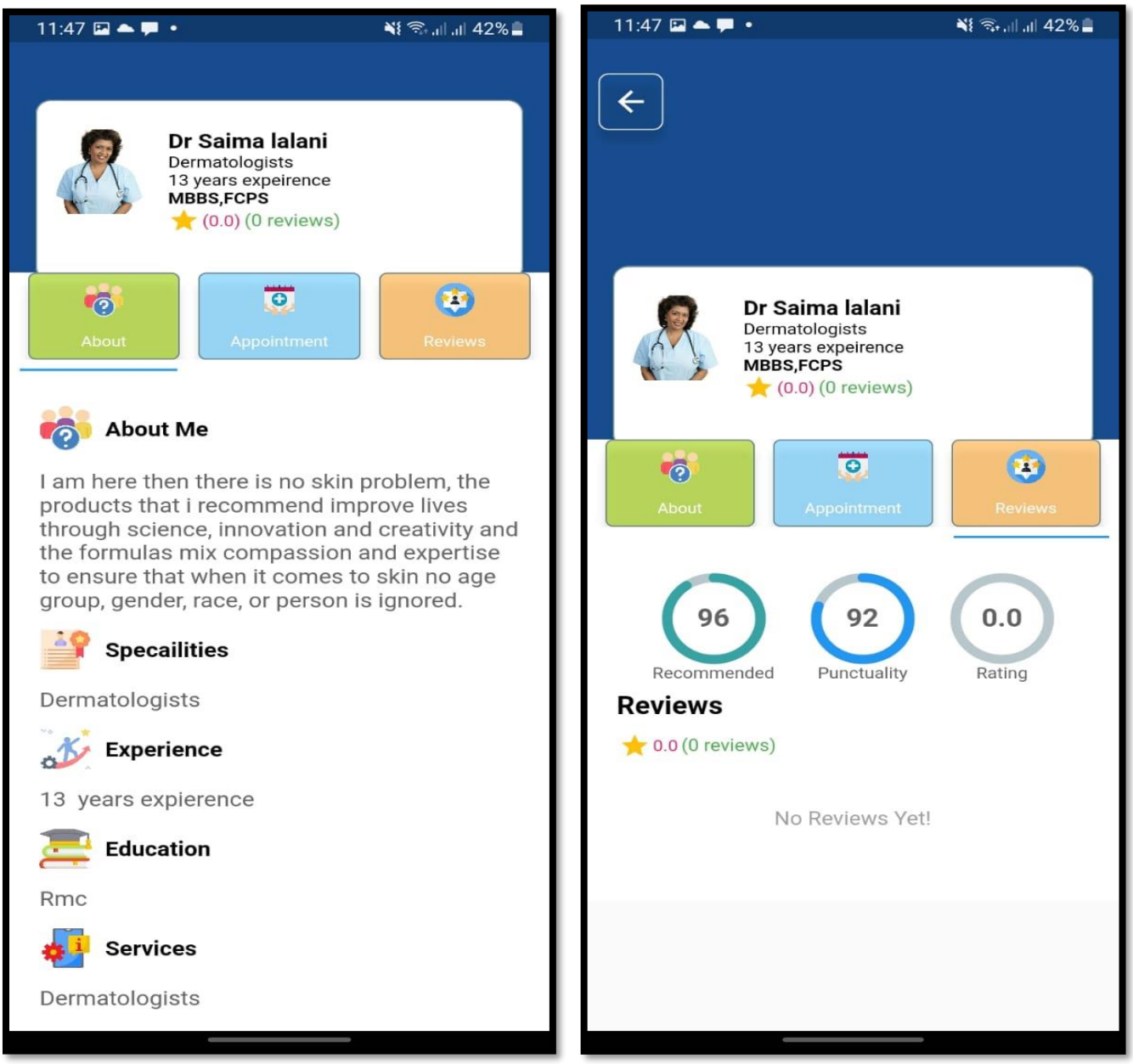

Figure 58: Screenshots of View Doctor Profile

4.8.16. Book an Appointment with the doctor

The patient can book the appointment with the doctor of his own choice.

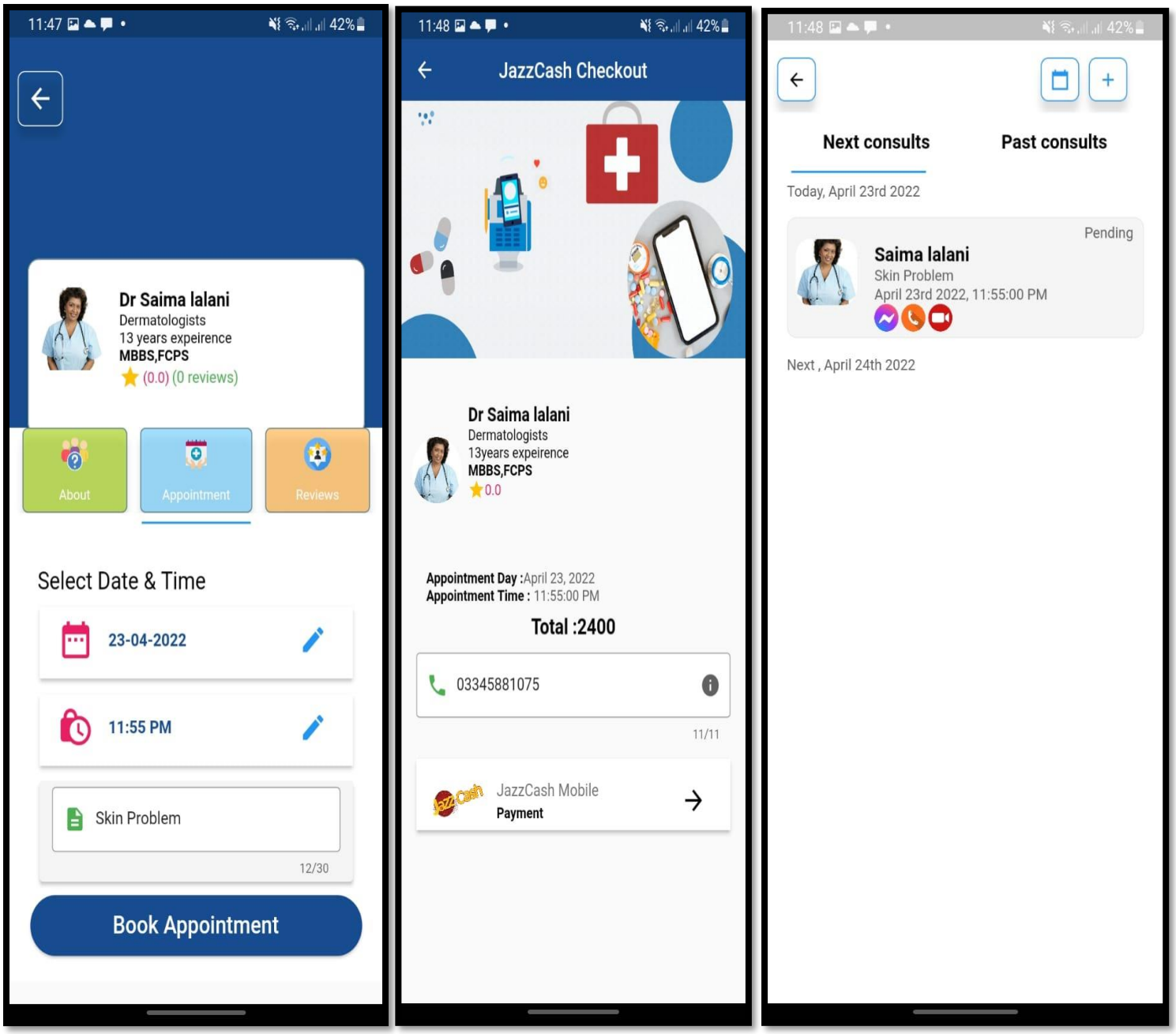

Figure 59: Screenshots of Booking an

4.8.17. Patients' wallet:

The patient can view his wallet. If the doctor has rejected an appointment due to some reason the patient will get back his money in his wallet.

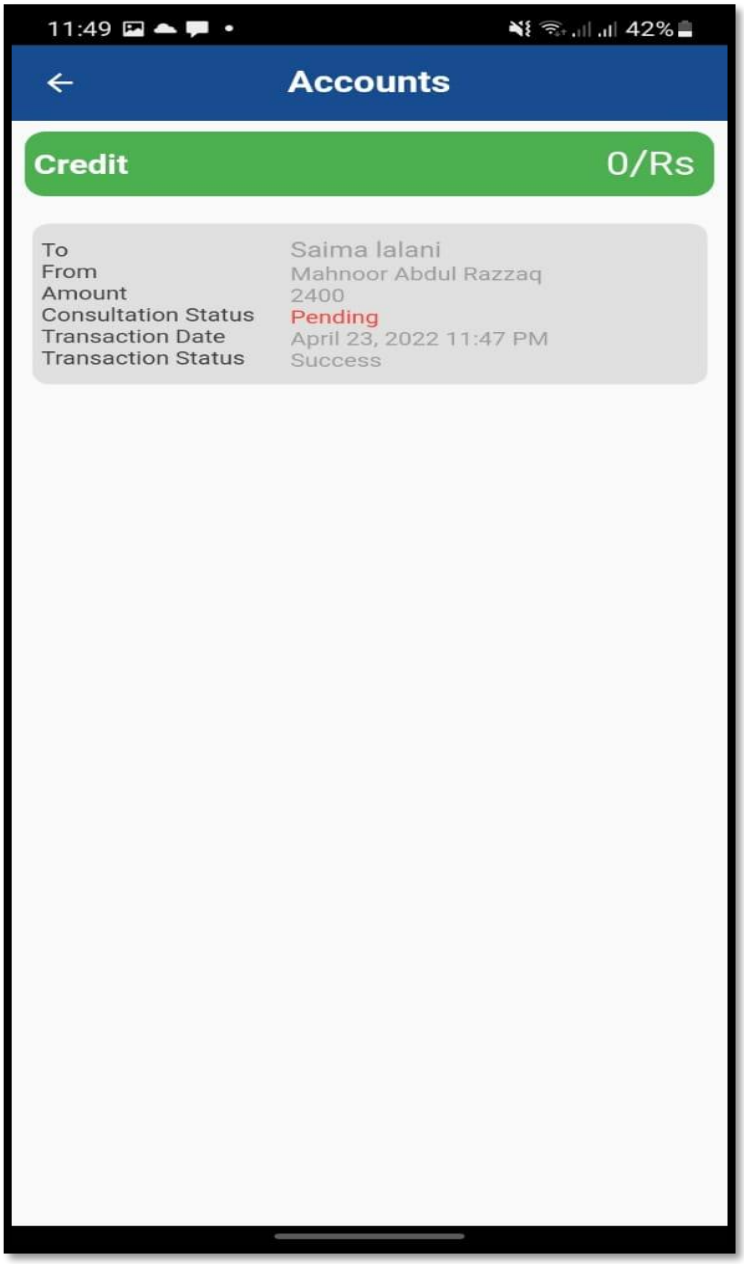

Figure 60: Screenshot of Patient Wallet

4.8.18. Appointment confirmed by doctor and doctor schedule updated

The doctor can view appointment requests and if the appointment is being confirmed, it will be listed in doctor's schedule.

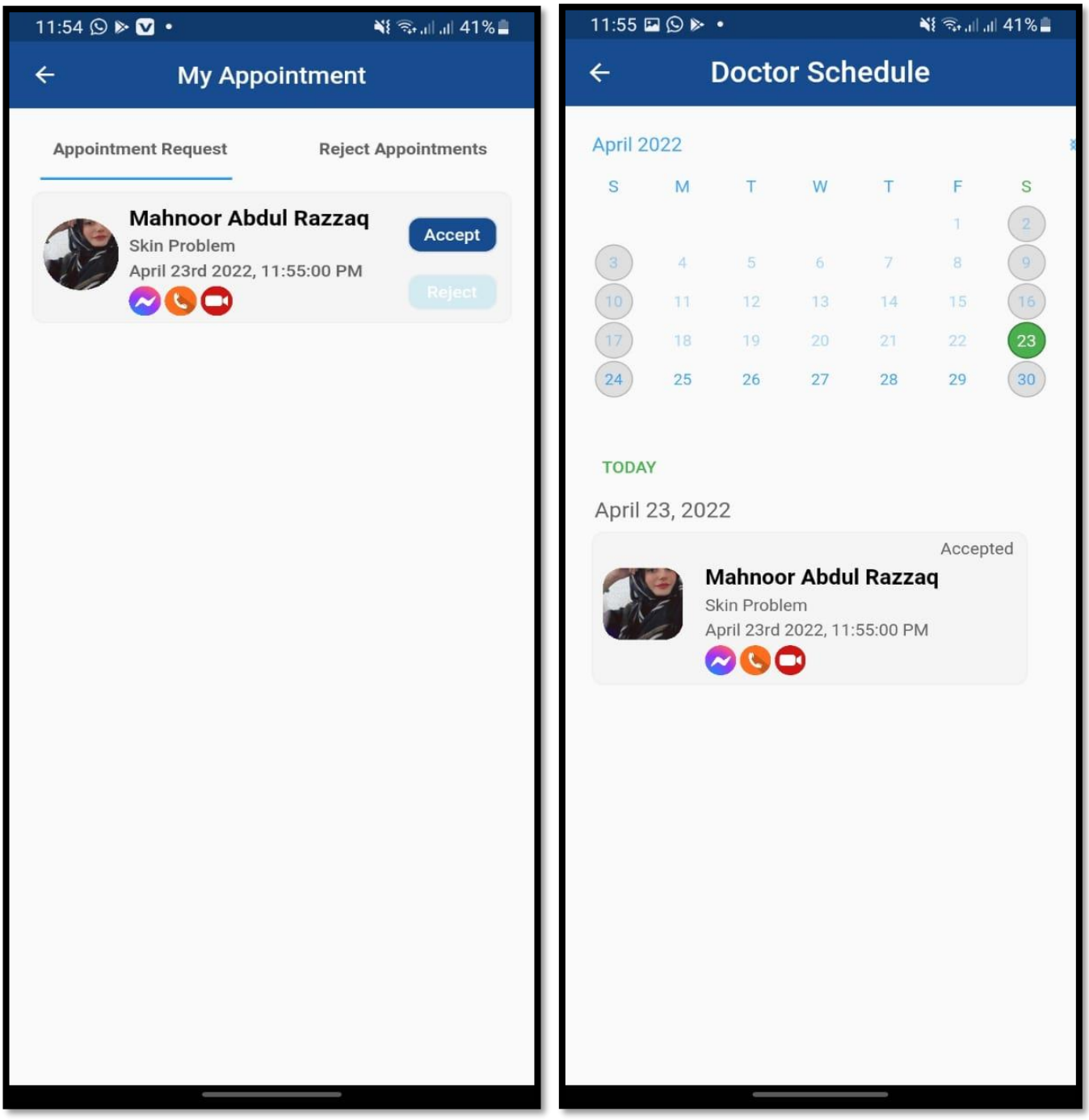

Figure 61: Screenshot of Appointment Requests and Schedule

4.8.19. Doctor's consult history page and patient are visible in the chat.

The doctor can view consult history and can start chatting with the patient at the time that is fixed for consultation.

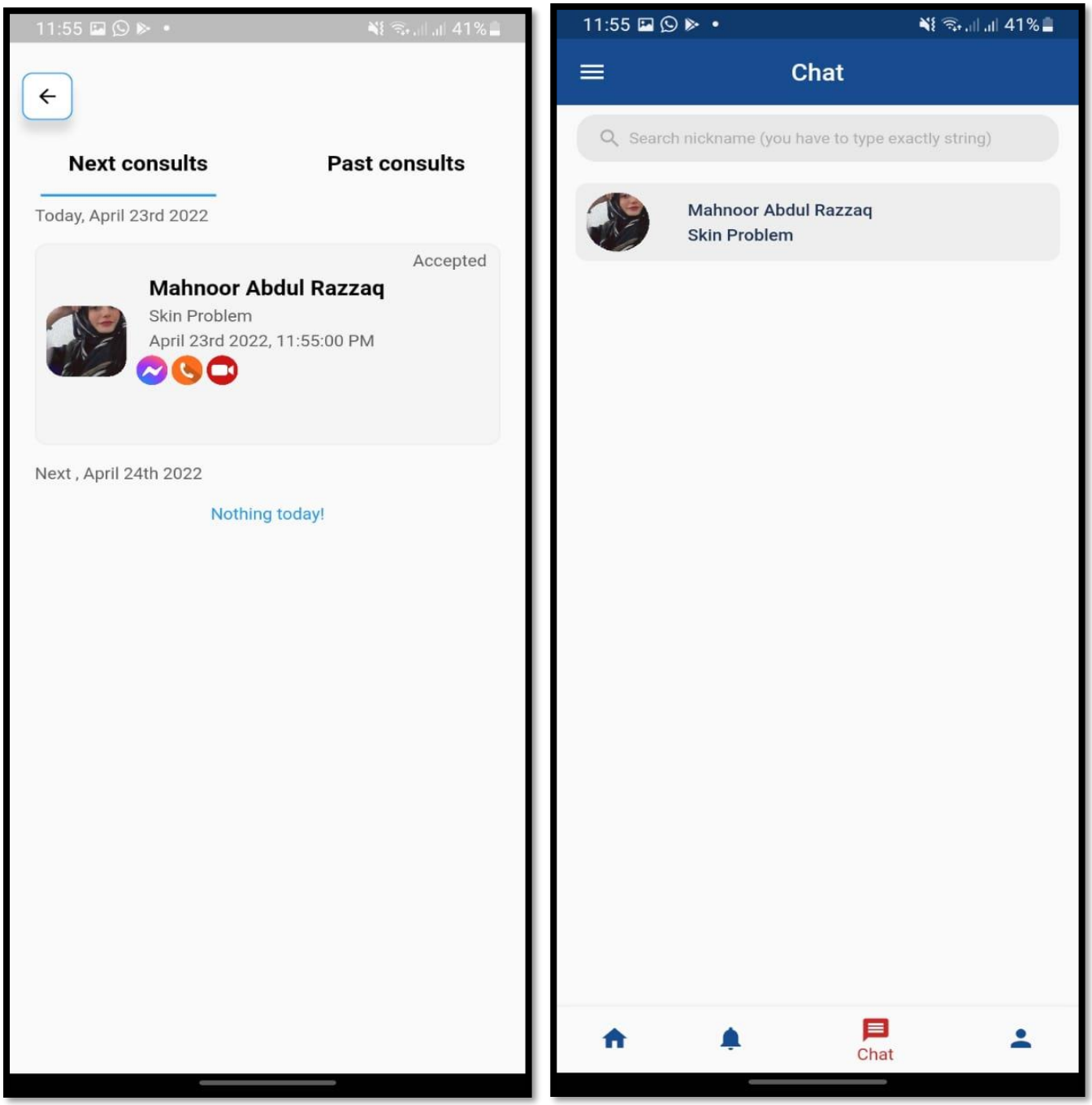

Figure 62: Screenshot of Consult History and Chat

4.8.20. Doctors' wallet page and can chat with the patient

The payment is visible in doctor's wallet and the doctor can chat or audio/video call with patient as well as send images or stickers.

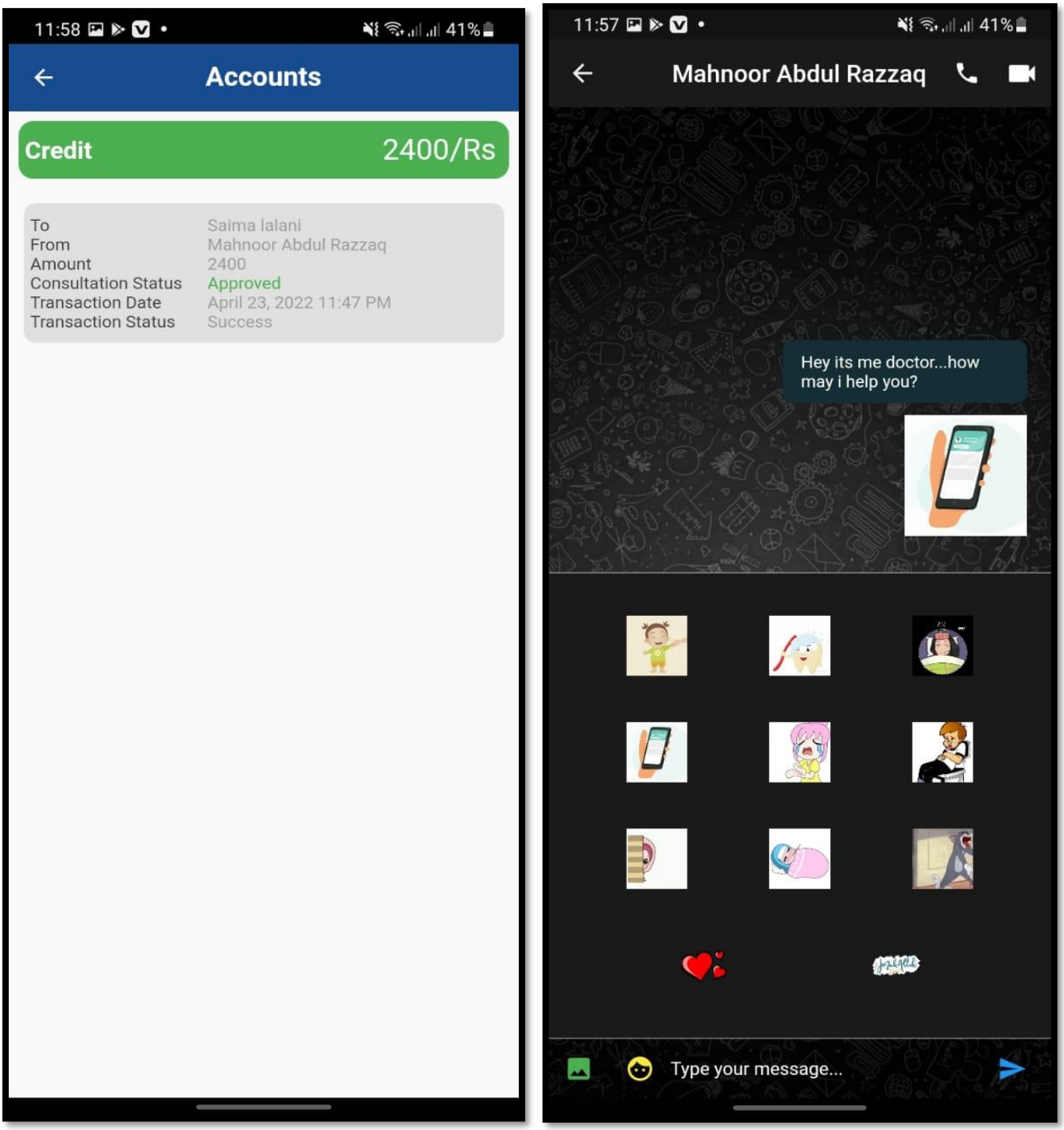

Figure 63: Screenshot of Doctors Wallet and Chat with the Patient

4.8.21. Doctor's Notification Page and Can reject Appointment

The doctor can view notifications and rejected appointments.

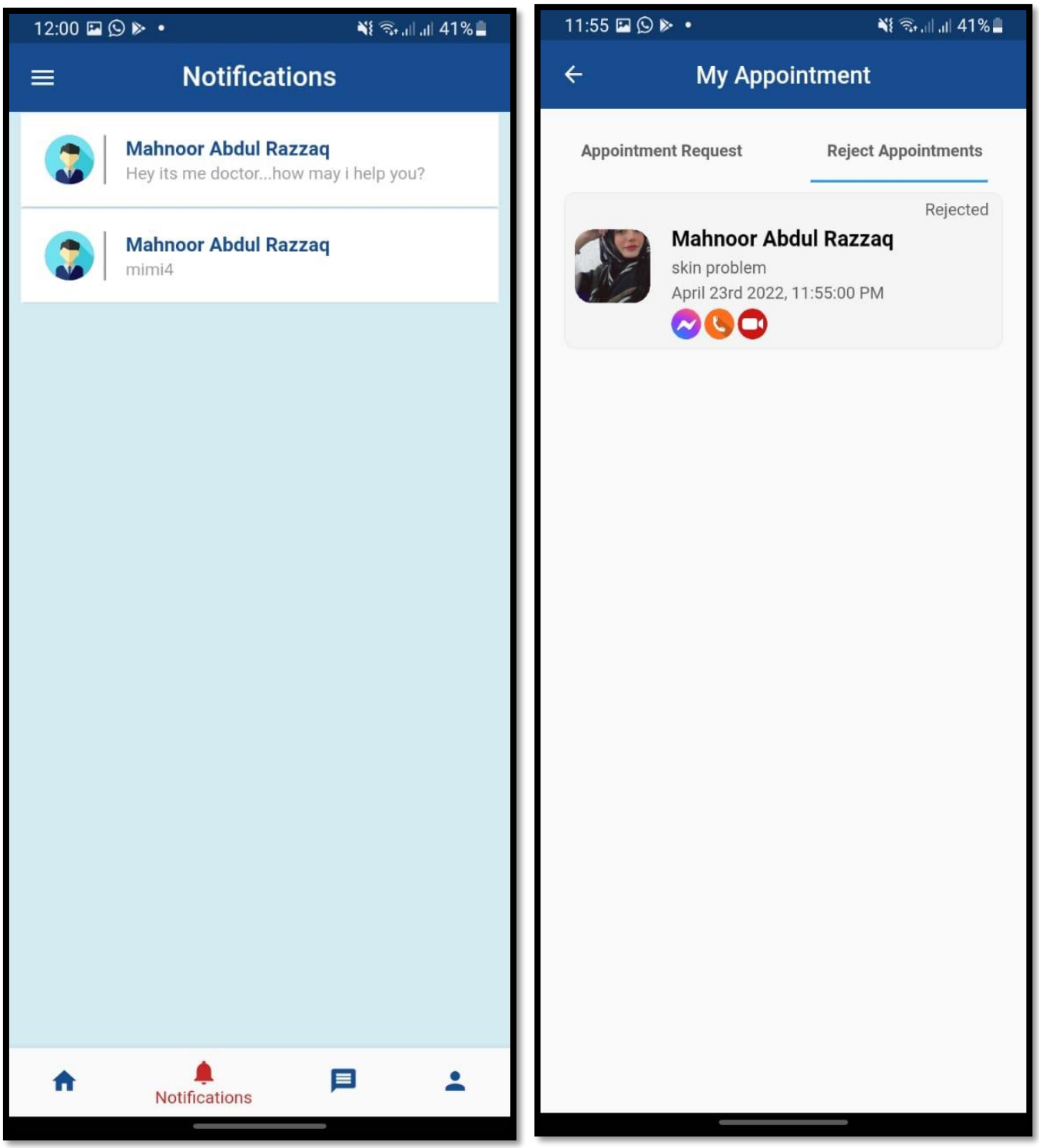

Figure 64:Screenshot of Notification and Rejected Appointment

4.8.22. Doctor's Activate Profile

After signing up the doctor will activate his profile which will be approved/rejected by the admin. And the doctor will be notified about admins response in a short while.

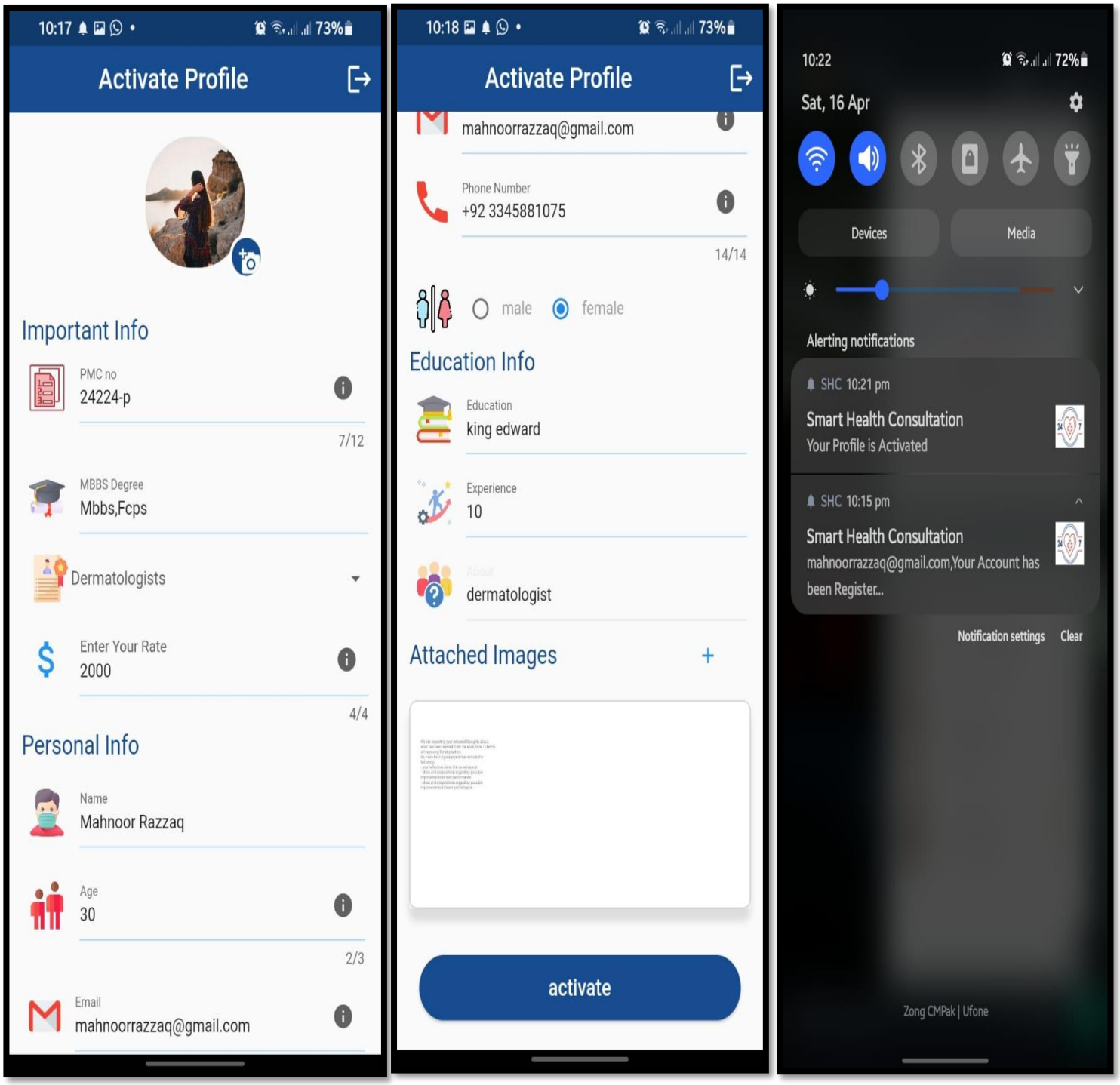

Figure 65: Screenshot of Activate Profile

4.8.23. Doctor's Home Page

Here is the home page of doctor's module.

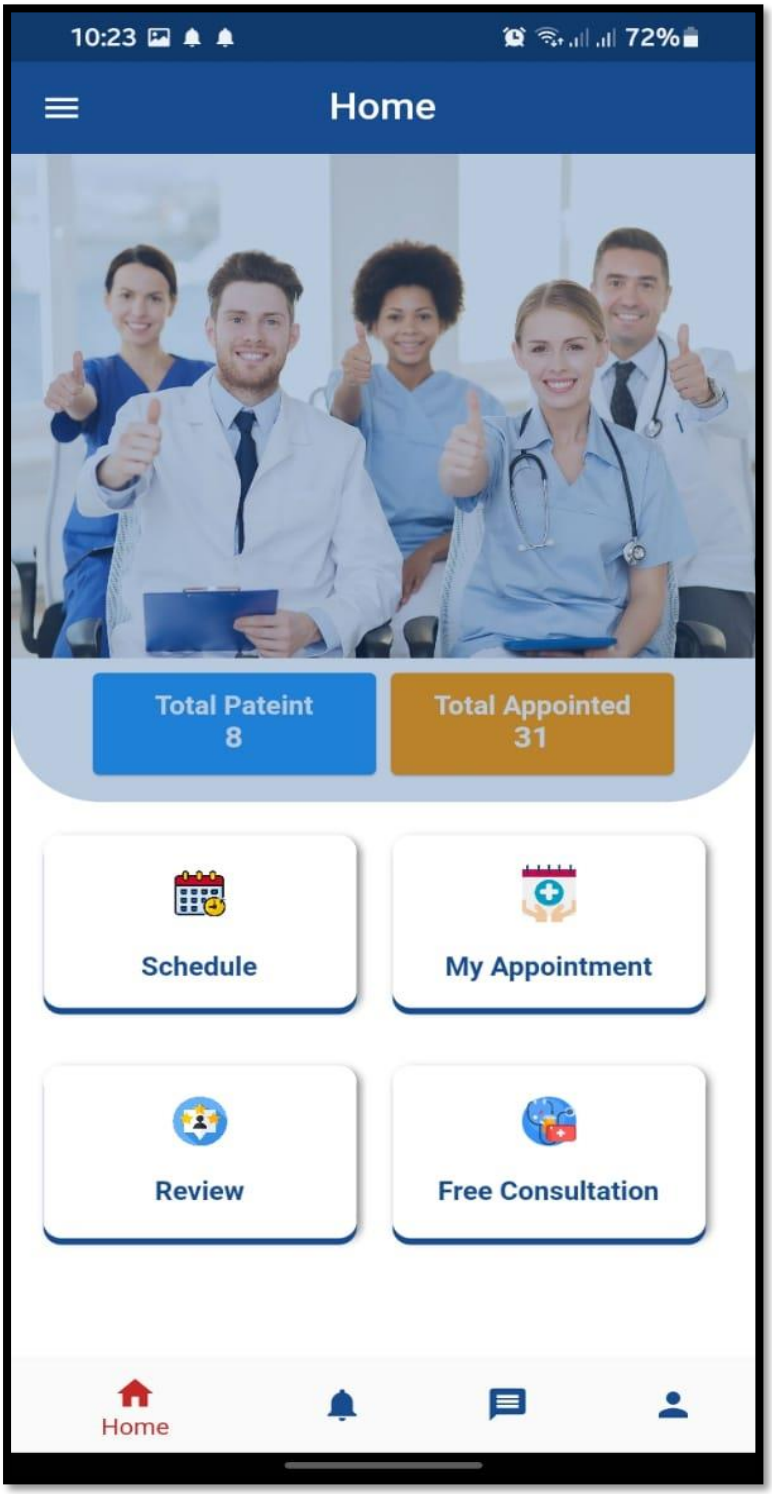

Figure 66: Screenshot of Doctors Home Page

4.8.24. Doctor can answer free consults

The doctor can answer free consultations willingly.

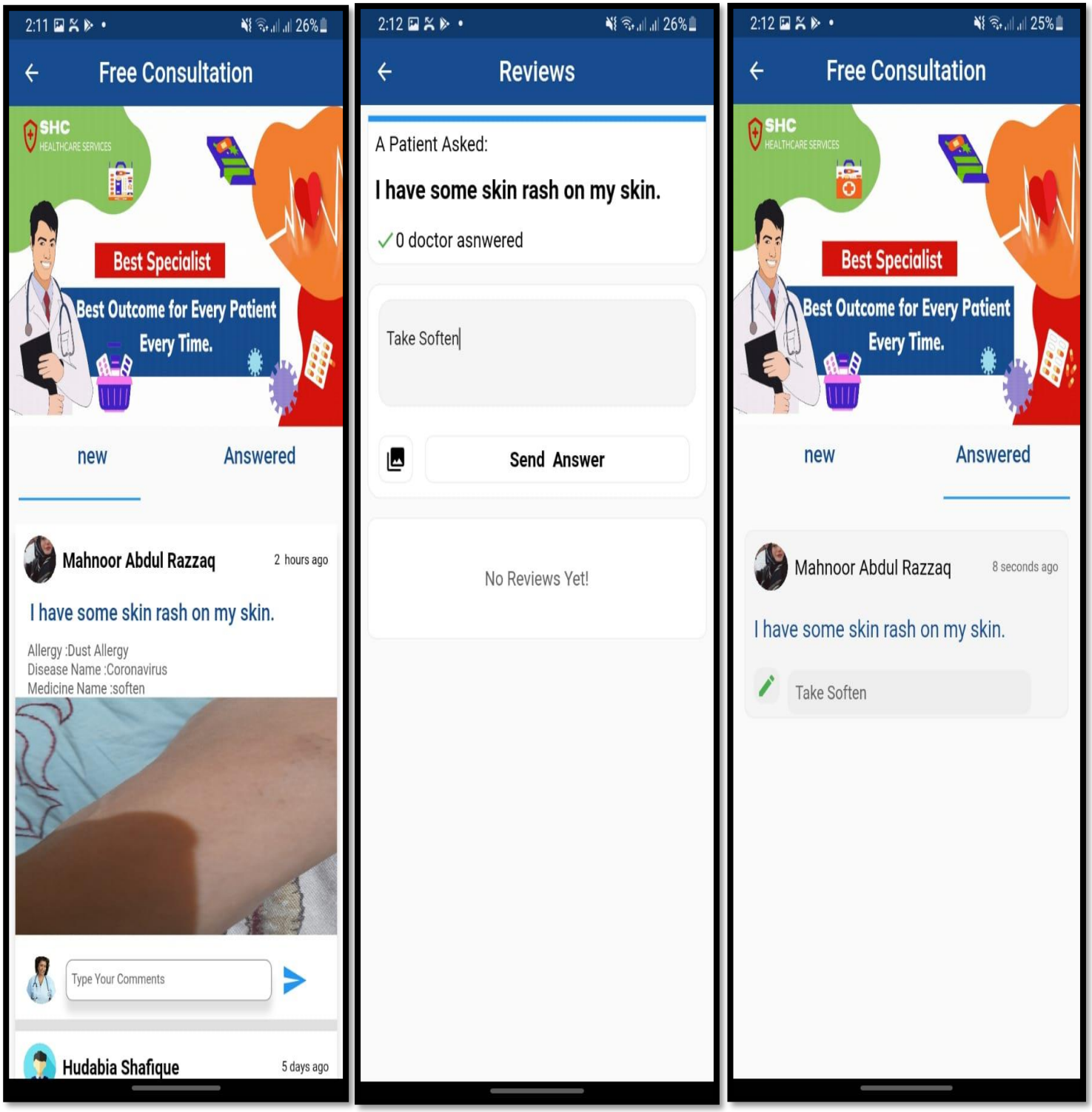

Figure 67: Screenshots of Answering Free Consultation

4.8.25. Doctor's Profile and Update Profile page:

The doctor can view and update his profile.

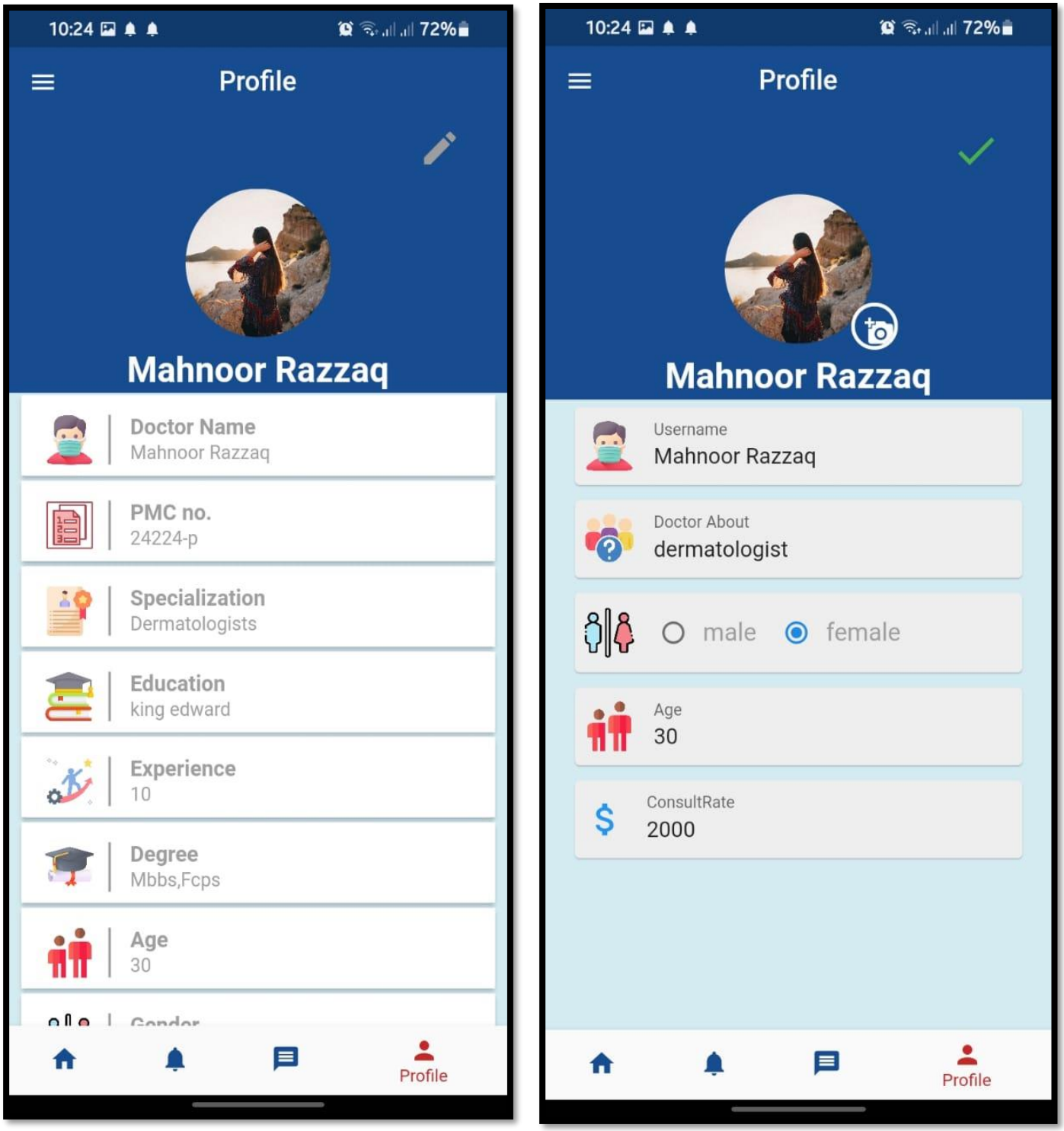

Figure 68: Screenshots of Doctors Profile

4.8.26. Drawer and review page:

Here is the screenshot of drawer and reviews that a doctor can give to the patient.

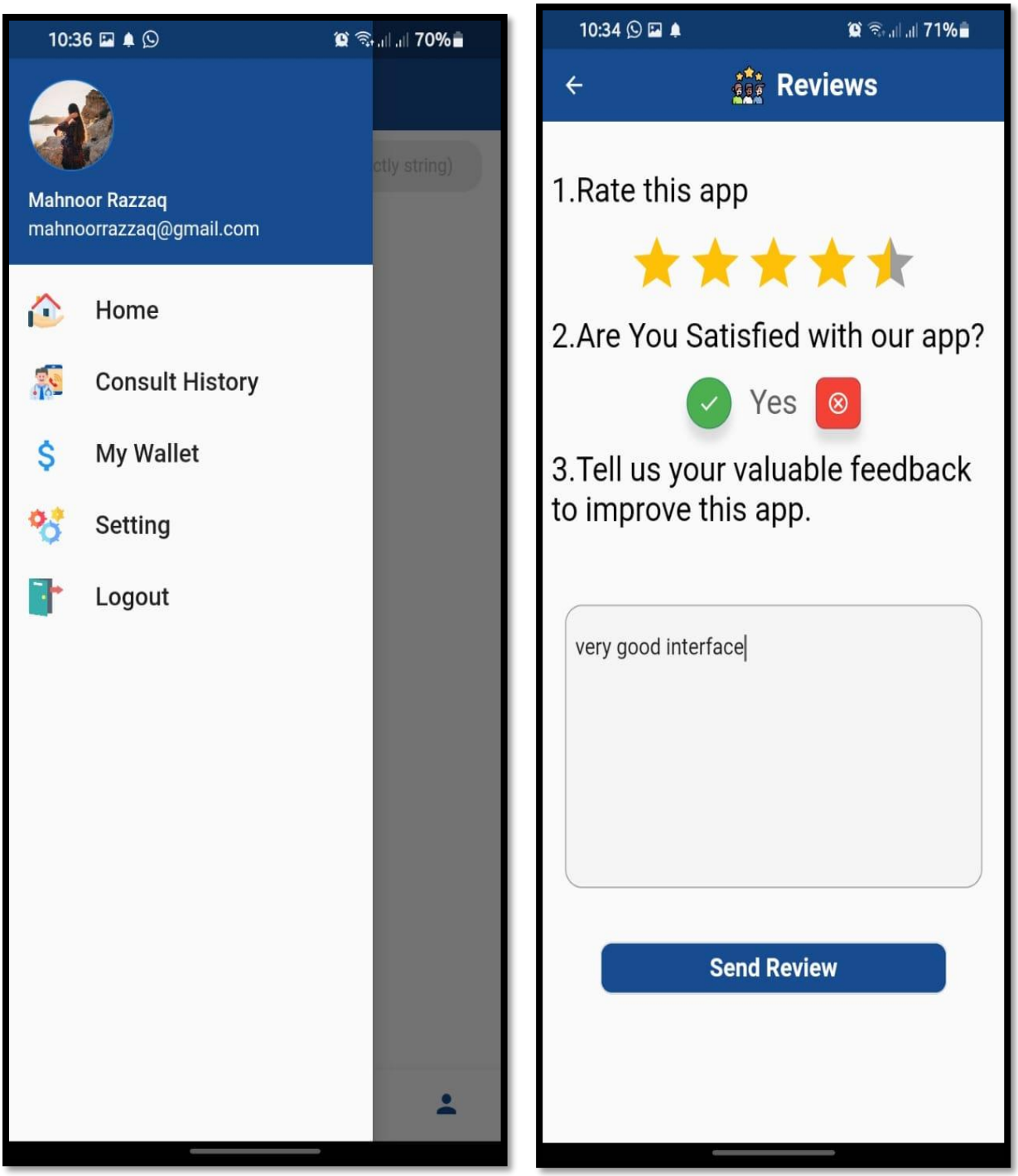

Figure 69: Screenshot of Drawer and Reviews

4.8.27. Admin Module

Here is the login screen for admin.

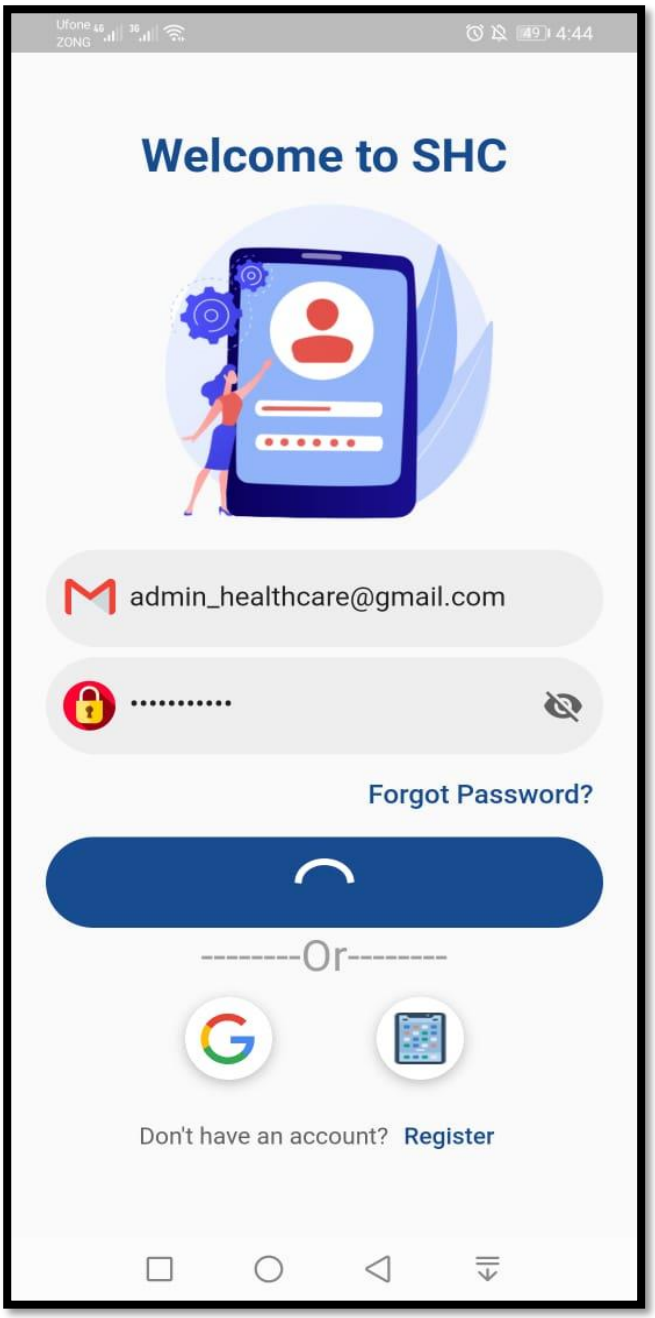

Figure 70: Screenshot of Admin Login

4.8.28. Yearly, Monthly and Weekly Statistics

The admin can view weekly, yearly, and monthly statistics.

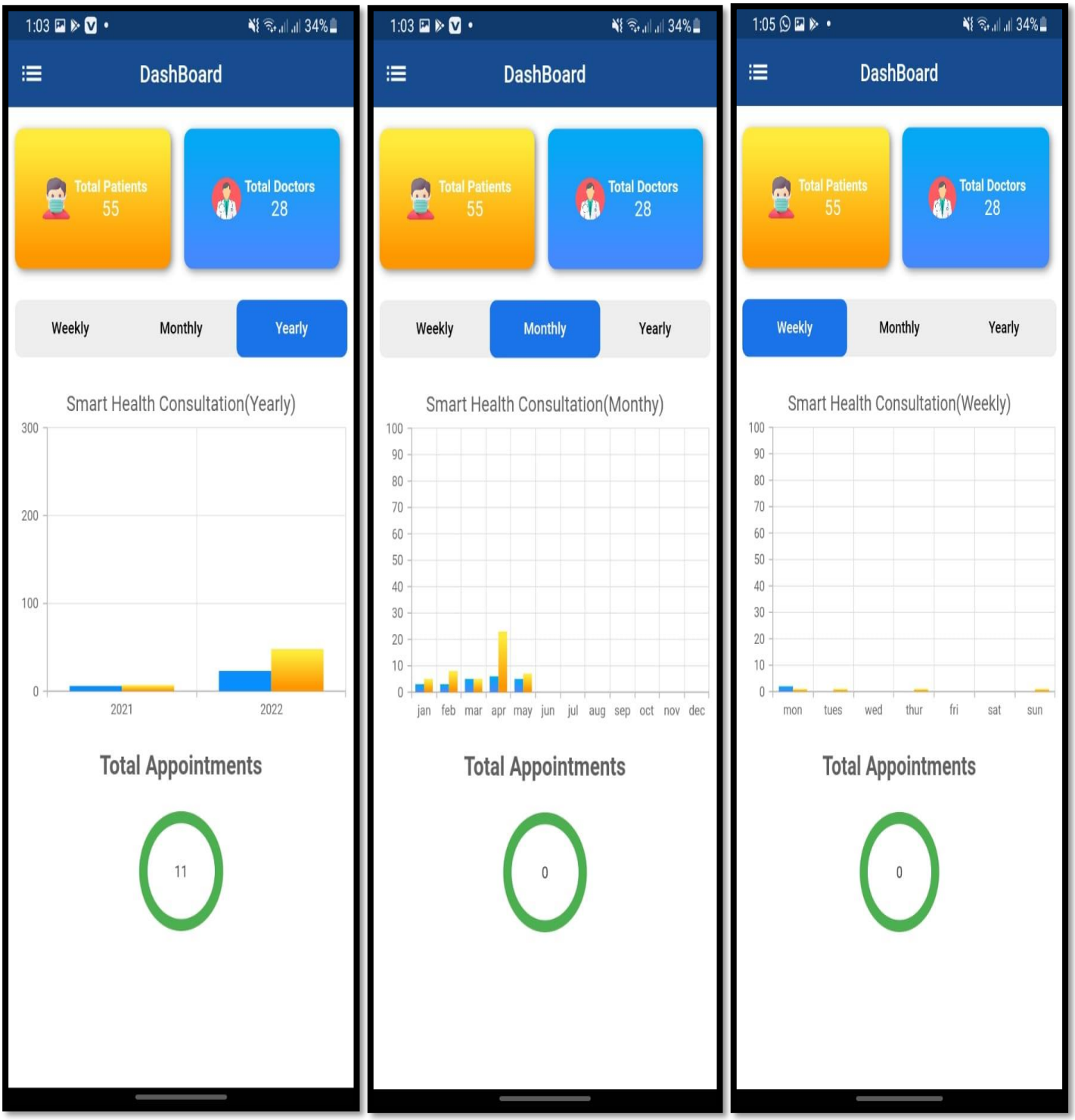

Figure 71: Screenshots of Statistics

4.8.29. Drawer of Admin Account and View Appointment

The admin can select different options by clicking on the drawer. The admin can also view the list of appointments.

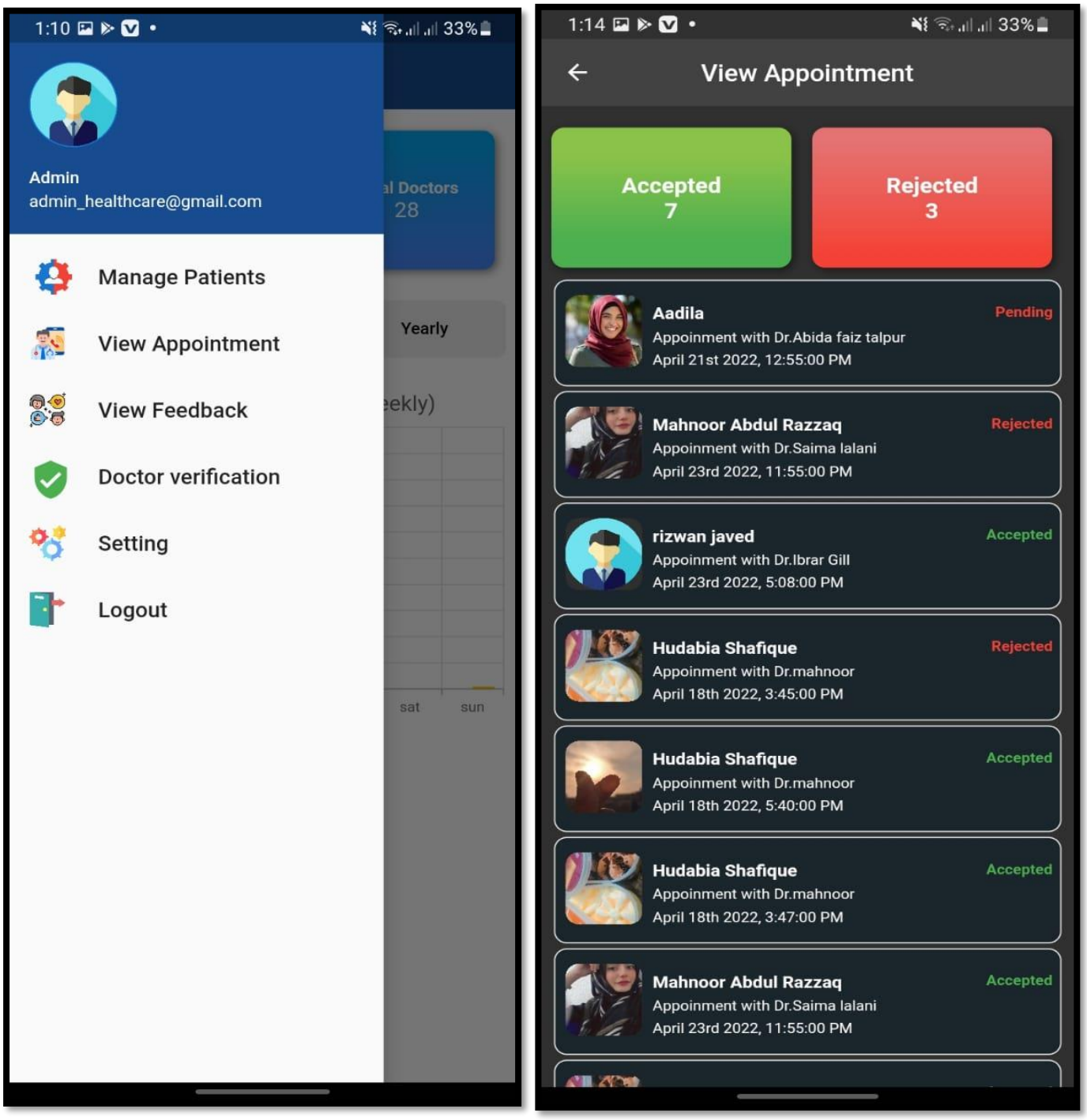

Figure 72: Screenshots of Admin Drawer and View Appointments

4.8.30. View Feedback of patient and doctor

The admin can view feedback of patients and doctors.

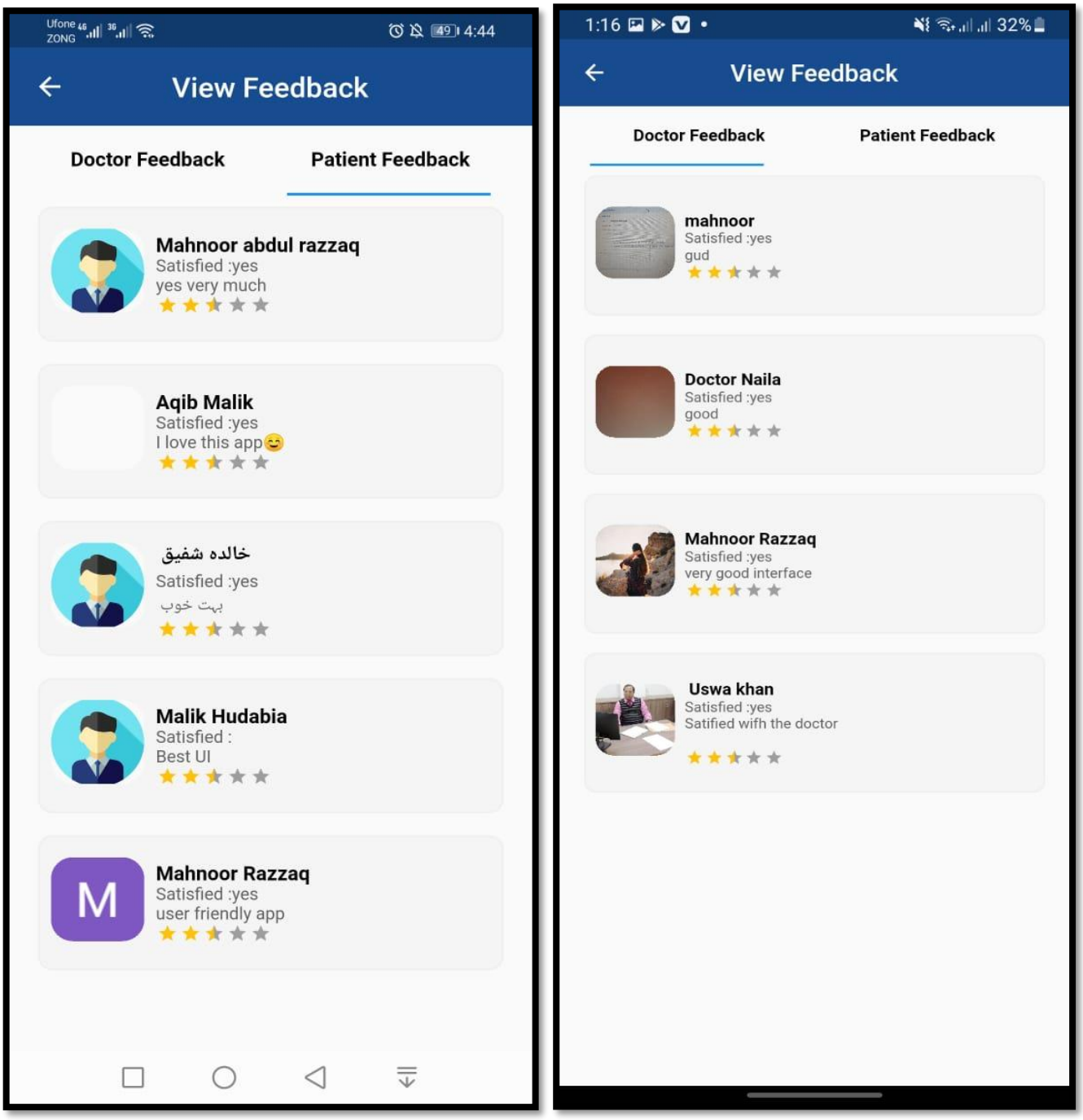

Figure 73: Screenshots of View Feedback

4.8.31. Manage Doctors

The admin can add, view, update, delete, accept/reject doctor's account.

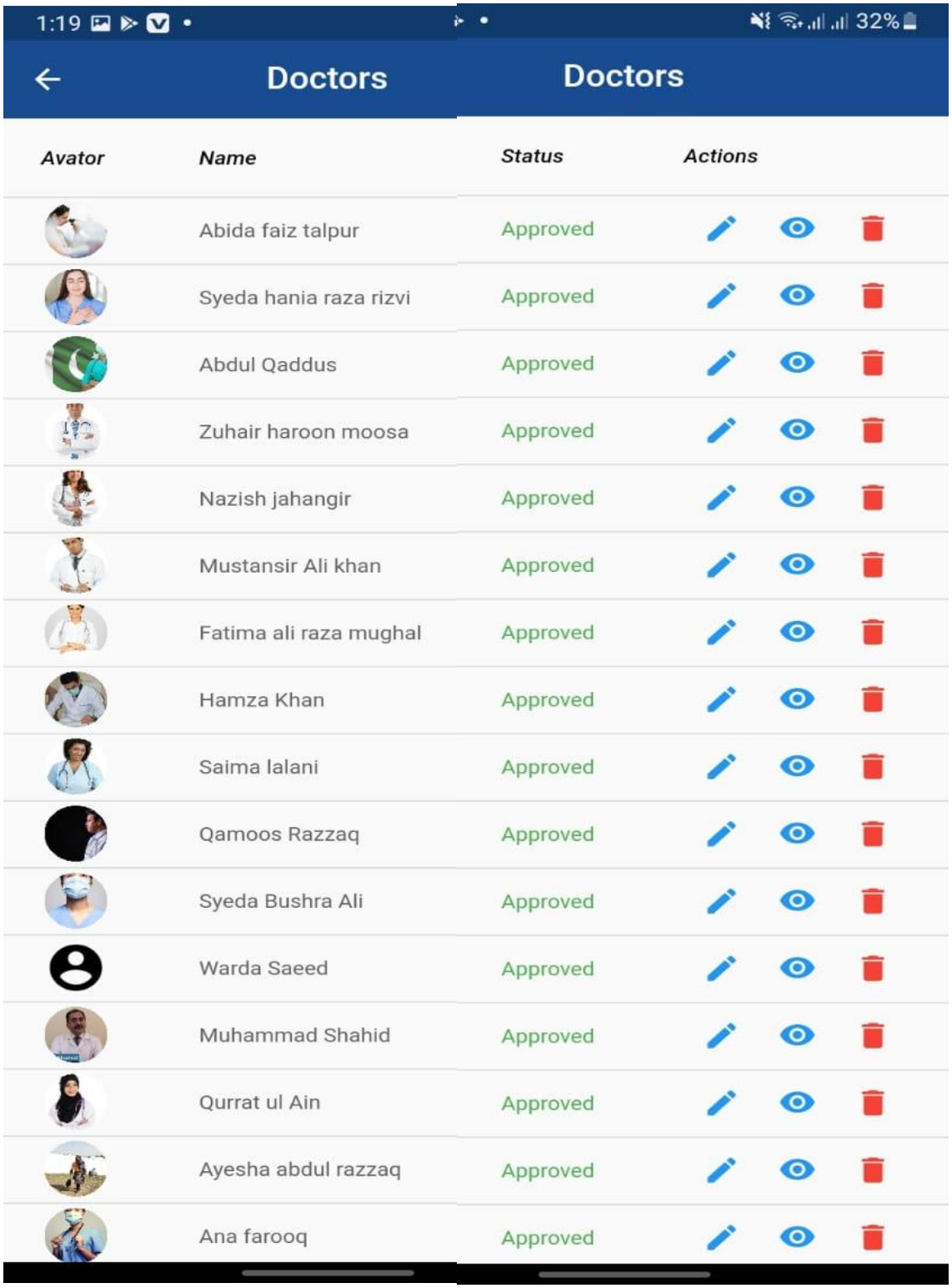

Figure 74: Screenshots of Manage Doctors

4.8.32. Add User

The admin can have the option to add patients or doctors.

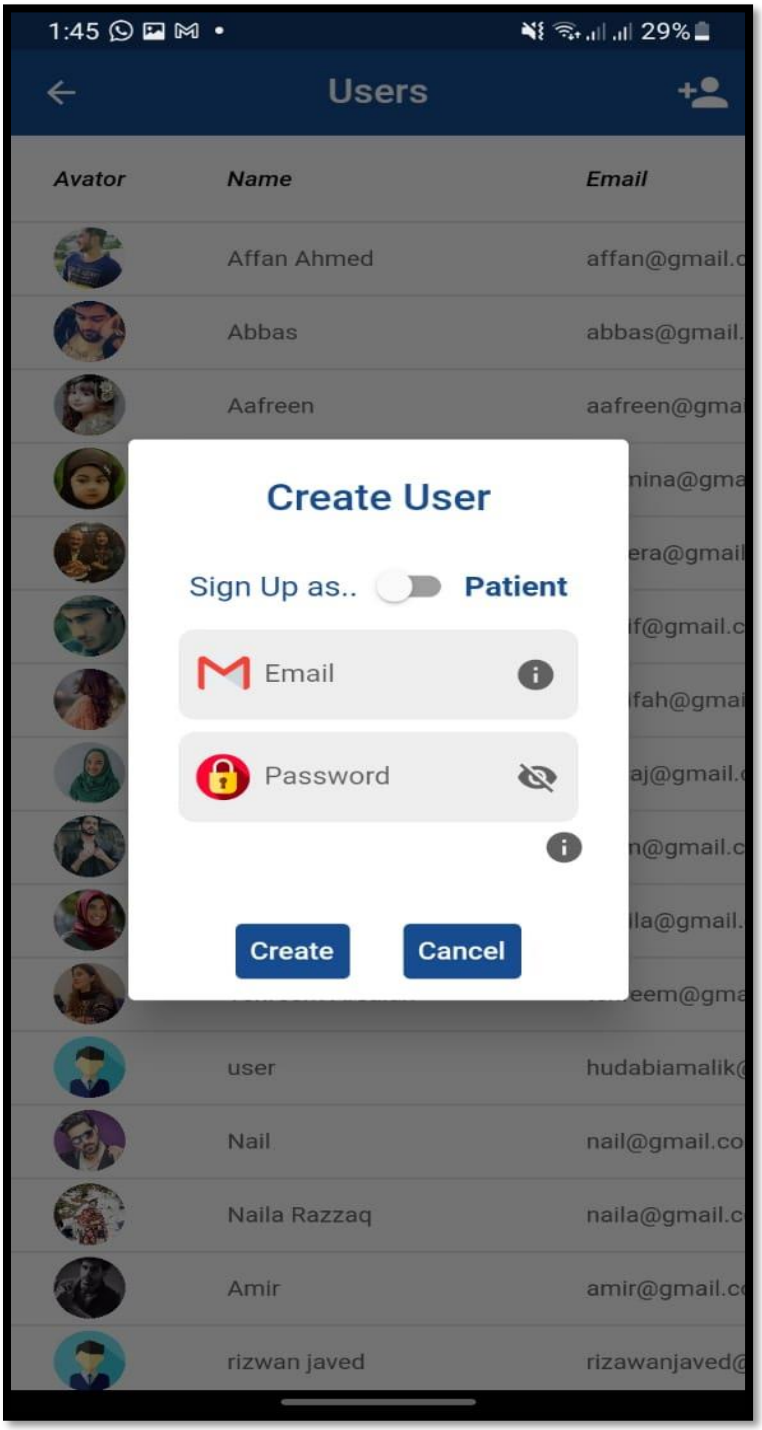

Figure 75: Screenshot of Add Users

4.8.33. Manage Patient

The admin can add, view, update, and delete patients.

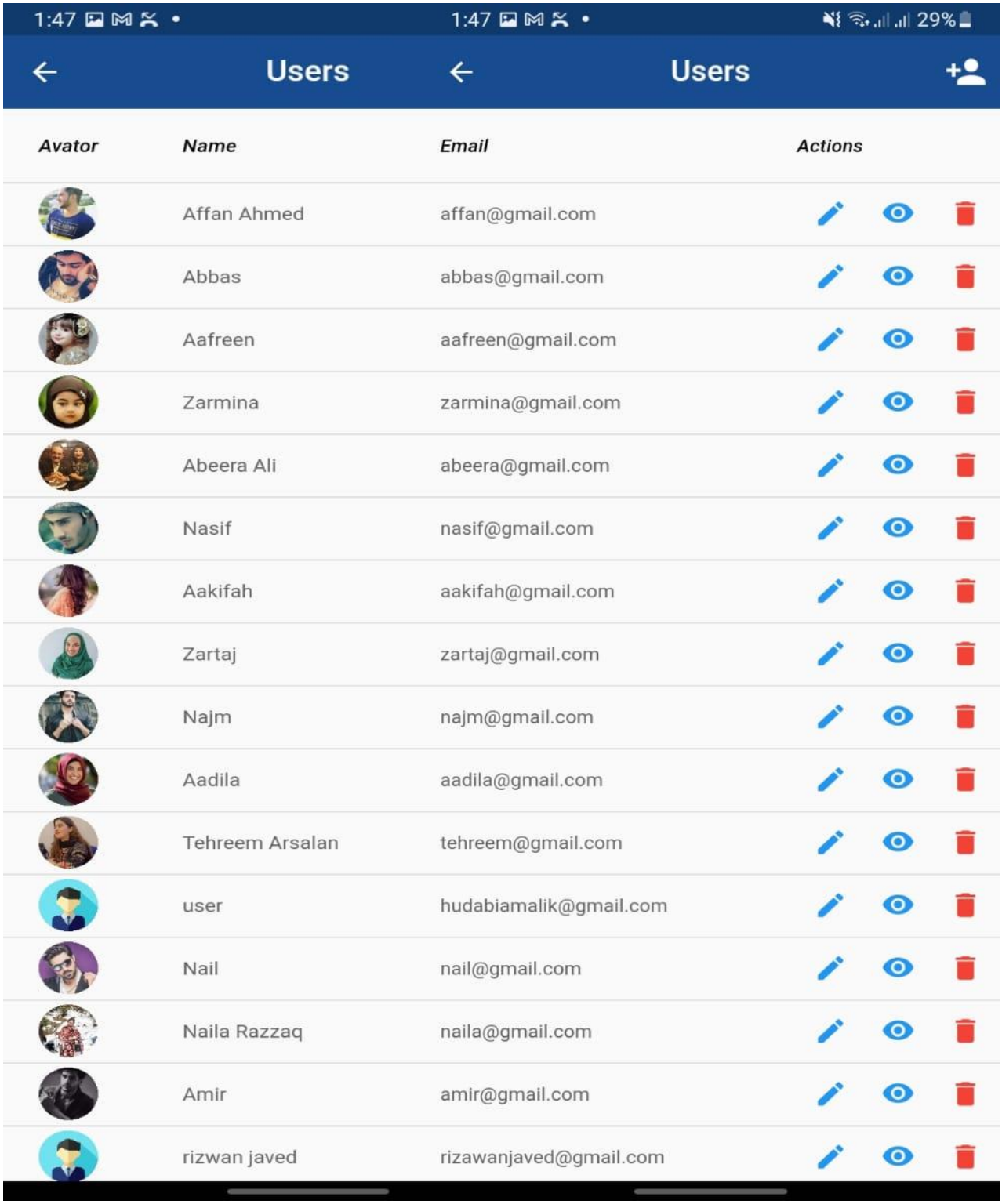

Figure 76: Screenshots of Manage Patients

4.9. Conclusion

In this chapter, we have shown the design approach, system architecture, and user interfaces of our application. We have shown the complete user interfaces of registration, admin module, patient module, and doctor module.

Chapter # 5

System Implementation

Chapter no. 5

System Implementation

This chapter describes the implementation phase of the Smart Health Consulting App. Tools and techniques used in the development and completion of our project are explained here. After the complete implementation, our system can perform all the functionality for a free consultation, paid consultation, chat, selecting specialists, booking appointments, and others that are already explained in previous chapters.

5.1 Strategy:

The Smart Health Consulting App was implemented using a hybrid of Code and Fix and Incremental Model. The basic features that were agreed upon during the requirements phase were implemented and then improvements upon those features were made with every iteration as more and more code was generated. Our aim was to create two main modules that were free consultation and paid consultation. First, we worked on free consultation module from both doctor and patient end. After the successful completion of the free consultation module, we worked on paid consultation module on both ends which are the doctor and patient end. After working on the main modules, we worked on all the other remaining features. Here is the pictorial representation of our strategy:

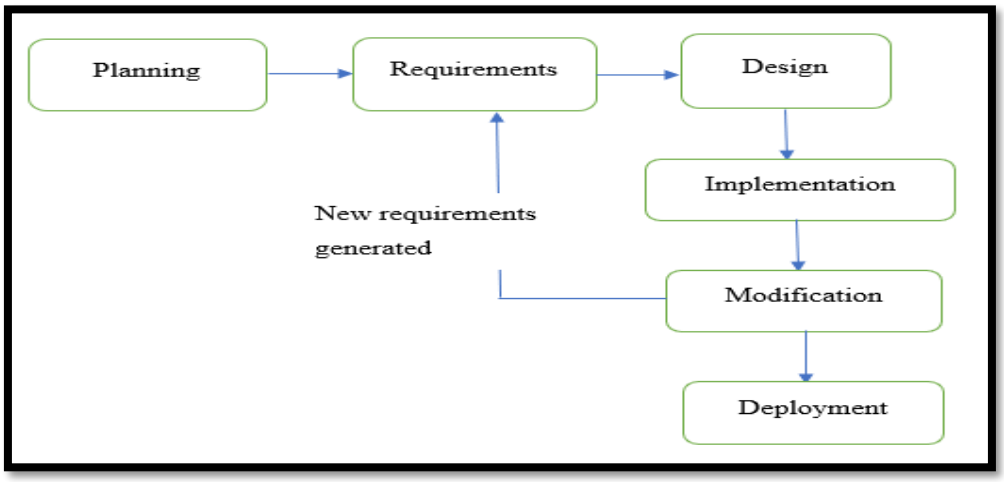

Figure 77: Strategy

5.2 Tools and Technologies:

Tools	Version	Rationale
Ms Word	2016	Documentation
Ms PowerPoint	2016	Presentation
Visual Studio Code	1.66	Ide
Visual Paradigm	16.3	UML Diagrams
Technology	**Version**	**Rationale**
Flutter	2.5.0	Android And IOS App
Dart	2.14.1	Language
Firebase	19.0.1	Database

Table 69: Tools and Technologies

5.2.1 Explanation:

1. **MS Word:**

   We have used Microsoft word for documentation purposes. We have used the latest version 2016 of Microsoft word.

2. **MS PowerPoint:**

   We have used Microsoft PowerPoint for creating presentations. We have used the latest version of 2016 of Microsoft PowerPoint.

3. **Visual Studio Code:**

   We have used Visual Studio Code as an integrated development environment for the creation of our application.

4. **Visual Paradigm:**

   We have used a visual paradigm for creating UML (Unified Modelling Language) diagrams.

5. **Flutter:**

   We have used the latest technology named flutter for our application because it is a cross-platform technology. The first stable version of flutter was released in 2018. Flutter provides custom, animated UIs of any complexity available.

6. **Dart Language:**

   We have used dart language for the development of our project.

7. **Firebase:**

   We have used firebase as a database because it is a real-time database. Firebase provides swift and secured hosting and optimized app performances.

5.3 Conclusion:

The system requires the study of Mobile Application Development in Android and Flutter. The mobile application interface and main features require a grip in Flutter and Dart language because all the implementation of interfaces are coded. The implementation of android and iOS applications requires ample knowledge of flutter and dart language because flutter is a platform-independent tool.

Chapter # 6

System Testing and Evaluation

Chapter 6

System Testing & Evaluation

This chapter will evaluate the system and its component per the expected outcomes. We will test each scenario against its expected outcome and results. Testing is the process that requires completing the software life cycle to ensure that the software is error-free and acquires high quality.

6.1 Test Strategy

Every software needs to be tested before deployment to ensure whether the software platform is as intended. It also ensures the reliability and usability of the system by unearthing the bugs. Testing ensures that maximum bugs are diagnosed in the system.

To test our system, we followed the code, implement and test strategy, initially we code a module and perform a test on it to ensure that the module work as per intention. We then implement the module in the overall system and perform more strenuous tests so that its functionality is assured to be optional when used with the overall system. To carry out testing, we performed different types of tests depending on the timing and nature of the module. These tests are as follows:

6.2 Component Testing

Component testing is a phase of testing in which the individual components or modules of a program or system are tested independently. This type of testing is also referred to as module testing. Component testing occurs before integration testing. Component testing occurs before integration testing. Here all the component is tested before their interaction with the rest of the module tasks is placed.

6.3 Unit Testing

Unit testing is the initial phase of testing in software. In unit testing, the tiniest components of the system usually referred to as units, are scrutinized for appropriate behaviour and performance individually. The process of unit testing can be carried out by automatic or manual means. To perform unit testing, we implemented a testing strategy known as use case testing.

6.4 Integration Testing

This is a form of testing where the correctness of the module's functionality of the system is tested by integrating different modules of the system. The different modules of the system are tested with different inputs and their actual results are compared with the expected result. After the complete integration testing, the system testing is performed.

6.5 System Testing

System testing is performed when the whole system is built. The system testing falls within the black box testing in which we only tested its overall features against the expected features of the system. Our application must completely fulfill the overall specification that we have mentioned in our requirement.

6.6 Use case Testing

6.6.1 User Management:

6.6.1.1 Test Case: 01

Table 70: Test Case of Registration by Mobile No

Test case	01
Objective	Registration By Mobile No
Pre-Condition	1. The App Should be open on mobile. 2. The User adds information to register.
Flow	1. User Open Application. 2. Choose Language or Skip That Part (Default Language Will Be Chosen).

	3. Users can choose to register as a patient or doctor by sliding the toggle button. 4. User Enter Name and Mobile No. 5. The user checks the Terms and Condition dialog box. 6. The user clicks on the Register button. 7. System Sends a Code On Entered Phone No. 8. User Enters the Code Received by SMS. 9. User click on submit button. 10. System Verify Entered Code and Register the User. 11. System Display Message That "Your Account Has Been. Registered Successfully".
Expected Output	Registration Successful
Actual Output	Registration Successful
Status	The Test Was Successfully Performed.

6.6.1.2 Test Case: 02

Table 71: Test Case of Registration by Email

Test case	02
Objective	Registration By Email
Pre-Condition	1. App Should Be Installed on the Mobile.

	2. The User adds information to register.
Flow	1. User Open Application. 2. Choose Language or Skip That Part (Default Language Will Be Chosen). 3. Users can choose to register as a patient or doctor by sliding the toggle button. 4. The User enters their Email and Password. 5. The User checks the Terms and Condition dialog box. 6. The user clicks on the register button. 7. System Verify Entered email and Register the User. 8. System Display Message that "Your Account Has been Registered Successfully".
Expected Output	Registration Successful
Actual Output	Registration Successful
Status	The Test Was Successfully Performed.

6.6.1.3 Test Case: 03

Table 72: Test Case of Login Screen

Test case	03
Objective	Login Screen
Pre-Condition	The User adds information to the login.
Flow	1. Enter the email and password. 2. Click on the login button. 3. System Display Message That "Your you have login Successfully".
Expected Output	Login Successfully and the home page is displayed.
Actual Output	Login Successfully and the home page is displayed.
Status	The Test Was Successfully Performed.

6.6.1.4 Test Case: 04

Table 73: Test Case of Change Language

Test case	04
Objective	Change language
Pre-Condition	The App should be open.
Flow	1. Users can click on the skip button to choose the default (English) Language or choose the Urdu language by selecting from the drop-down menu.

	2. Language is changed.
Expected Output	Language changed successfully.
Actual Output	Language changed successfully.
Status	The Test Was Successfully Performed.

6.6.1.5 Test Case: 05

Table 74: Test Case of Select Theme

Test case	05
Objective	Select Theme
Pre-Condition	The user should be Login.
Flow	1. The user clicks on a drawer. 2. Then the user can view the settings page. 3. Users can select a theme (dark or light) depending on their feasibility. 4. The theme is changed.
Expected Output	The theme changed successfully.
Actual Output	The theme changed successfully.
Status	The Test Was Successfully Performed.

6.6.1.6 Test Case: 06

Table 75: Test Case of Forgot Password

Test case	06
Objective	Forgot Password
Pre-Condition	The mobile app is open, and the user is on the login page.
Flow	1. Enters email and incorrect password. 2. Click Login. 3. The system display's invalid username or password error. 4. The user clicks on forgot password button. 5. Forget the password page open in which users have to enter their email address and then click on the Reset button. 6. User opens their email account in which user found a mail through which they can reset their password. 7. The user opens that mail and changes the password and clicks on save. 8. System display message that "Password is Reset successfully".
Expected Output	Password changed successfully.
Actual Output	Password changed successfully.
Status	The Test Was Successfully Performed.

6.6.1.7 Test Case: 07

Table 76:Test Case of Logout

Test case	07
Objective	Logout
Pre-Condition	1. The user is logged in. 2. The user no longer wants to be logged in.
Flow	1. The user is done using the mobile application. 2. The user clicks on the logout button. 3. The system logs the user out.
Expected Output	Logout successfully.
Actual Output	Logout successfully.
Status	The Test Was Successfully Performed.

6.6.1.8 Test Case: 08

Table 77: Test Case of Update Profile

Test case	08
Objective	Update Profile.
Pre-Condition	The mobile app is open, and the user is on the Profile page.
Flow	1. User selects Update Profile.

	2. The app displays profile information. (Name, age, gender, date of birth, Image, etc.) 3. User updates profile, by clicking on the update button. 4. System display message that "Profile updated successfully".
Expected Output	Update profile successfully.
Actual Output	Update profile successfully.
Status	The Test Was Successfully Performed.

6.6.1.9 Test Case: 09

Table 78: Test Case of View Profile

Test case	09
Objective	View Profile
Pre-Condition	The mobile app is open, and the user is at the Home page.
Flow	1. User selects Profile. 2. The app displays profile information. (Name, Age, Image, etc.) 3. User view profile information.
Expected Output	View profile successfully.
Actual Output	View profile successfully.
Status	The Test Was Successfully Performed.

6.6.1.10 Test Case: 10

Table 79: Test Case of Give Feedback

Test case	10
Objective	Give feedback
Pre-Condition	The mobile app is open, and the user is on the Home page.
Flow	1. User selects Give Feedback. 2. Users select a rating and enter feedback. 3. The user clicks on Send Review button. 4. The system display's the message that "feedback sent successfully".
Expected Output	Feedback was sent successfully.
Actual Output	Feedback was sent successfully.
Status	The Test Was Successfully Performed.

6.6.1.11 Test Case: 11

Table 80: Test Case of Cancel Appointment

Test case	11
Objective	Cancel Appointment
Pre-Condition	1. The doctor must log in to the app. 2. The patient sends a consultation request to the doctor.

Flow	1. The doctor receives notification that “New consultation request”. 2. The doctor is on the home page. 3. The doctor selects the My appointment module. 4. The doctor selects the Appointment request. 5. The doctor can view the patient's profile. 6. The doctor clicks on the Reject button and the appointment is canceled. 7. The system cancels the reserved appointment and records in the schedule then updates the system. 8. System displays the refunded payment in the patient’s wallet.
Expected Output	Appointment cancels successfully.
Actual Output	Appointment cancels successfully.
Status	The Test Was Successfully Performed.

6.6.1.12 Test Case: 12

Table 81: Test Case of View MyAppointment

Test case	12
Objective	View MyAppointment
Pre-Condition	User must log in to the app.
Flow	1. User selects My Appointments. 2. The system shows the lists of appointments.

	3. System displays the list of all appointments pending appointments, canceled appointments, and executed appointments.
Expected Output	Appointment viewed successfully.
Actual Output	Appointment viewed successfully.
Status	The Test Was Successfully Performed.

6.6.1.13 Test Case: 13

Table 82: Test Case of View Notification

Test case	13
Objective	View Notification
Pre-Condition	User must log in to the app.
Flow	1. User selects Notification tab. 2. System displays the list of all notification.
Expected Output	Notification viewed successfully.
Actual Output	Notification viewed successfully.
Status	The Test Was Successfully Performed.

6.6.1.14 Test Case: 14

Table 83: Test Case of Real-Time Chat

Test case	14
Objective	Real-time chat
Pre-Condition	The User must log in to the app and the appointment is confirmed by the doctor.

Flow	1. User clicks on the Chat tab. 2. User automatically shows in chat to which the appointment has been confirmed. 3. Click on that user to chat with them. 4. If consultation time is remaining, then the system displays" Your consultation is in a few minutes". 5. Once the consultation time is reached, the patient and doctor can chat with each other.
Expected Output	Patient and doctor chat with each other successfully.
Actual Output	Patient and doctor chat with each other successfully.
Status	The Test Was Successfully Performed.

6.6.1.15 Test Case: 15

Table 84: Test Case of Send Image

Test case	15
Objective	Send Image
Pre-Condition	The User must log in to the app and the appointment is confirmed by the doctor.

Flow	1. The user clicks on Chat tab. 2. User automatically show in chat to which the appointment has been confirmed. 3. Click on that user to chat with them. 4. If consultation time is remaining, then the system displays" Your consultation is in a few minutes". 5. Once the consultation time is reach, patient and doctor can chat with each other. 6. Patient can send an image (reports, tests, etc.) to the doctor by clicking on the image icon.
Expected Output	Send Image successfully.
Actual Output	Send Image successfully.
Status	The Test Was Successfully Performed.

6.6.1.16 Test Case: 16

Table 85: Test Case of Confirm Appointment

Test case	16
Objective	Confirm Appointment
Pre-Condition	The Doctor must log in to the app.

Flow	1. Doctor receives notification that "New consultation request". 2. Doctor is on the home page. 3. Doctor click on the MyAppointment screen. 4. Doctor can see the patient's profile. 5. Doctor confirms the Appointment. 6. System displays the payment in Doctor's wallet. 7. Once the appointment is confirmed patient receives notification that "Appointment is accepted".
Expected Output	Appointment confirm successfully.
Actual Output	Appointment confirm successfully.
Status	The Test Was Successfully Performed.

6.6.1.17 Test Case: 17

Table 86: Test Case of Give Reviews and Ratings

Test case	17
Objective	Give Reviews and Ratings
Pre-Condition	The Patient must log in to the app.
Flow	1. Patient is on the home page. 2. Patient click on paid consultation module.

	3. Patient send Consultation request to doctor. 4. Doctor accepts consultation. 5. After getting a consultation patient can give reviews and ratings to the doctor.
Expected Output	Reviews and ratings sent successfully.
Actual Output	Reviews and ratings sent successfully.
Status	The Test Was Successfully Performed.

6.6.1.18 Test Case: 18

Table 87: Test Case of Search Doctor

Test case	18
Objective	Search Doctor
Pre-Condition	Patient must log in to the app and appointment is confirmed.
Flow	1. Patient click on the chat screen. 2. System displays chat screen to patient. 3. Patients can search for doctors by typing their full names into the search bar. 4. System display's the doctor successfully.
Expected Output	Search doctor successfully.
Actual Output	Search doctor successfully.
Status	The Test Was Successfully Performed.

6.6.1.19 Test Case: 19

Table 88: Test Case of Send Payment

Test case	19
Objective	Send Payment
Pre-Condition	Patient must log in to the app.
Flow	1. Patient is at home page. 2. Patient click on paid consultation module. 3. Patient selects the Specialization field. 4. Patient selects a specialist from the list of specialists and can view his profile and review & ratings. 5. Patient enters date and time, and describe issue by title. 6. Then click on Book Appointment. 7. Before an appointment is booked, the patient enters a mobile no to send payment by clicking on the payment button. 8. System displays the message that "your transaction is successful".
Expected Output	Payment sends successfully.
Actual Output	Payment sends successfully.
Status	The Test Was Successfully Performed.

6.6.1.20 Test Case: 20

Table 89: Test Case of Receive Payment

Test case	20
Objective	Receive Payment
Pre-Condition	Doctor must log in to the app.
Flow	1. Doctor is at home page. 2. Doctor click on MyAppointment screen. 3. Doctor can see the patient's profile. 4. Doctor confirm the Appointment. 5. Doctor receives payment and the System display the payment in Doctor's wallet.
Expected Output	Payment receive successfully.
Actual Output	Payment receive successfully.
Status	The Test Was Successfully Performed.

6.6.1.21 Test Case: 21

Table 90: Test Case of Free Consultation

Test case	21
Objective	Free Consultation
Pre-Condition	Patient must log in to the app.
Flow	1. Patient is at the home page.

	2. Patient click on the free consultation module. 3. Patient can see posts of other patients and comment on that posts as well. 4. Patient click on describe the disease and by clicking on that patient can post their problem and attach a picture with that. 5. System successfully post patient's problem. 6. Doctors click on the free consultation module to see all posts and can answer them willingly. 7. System successfully post the answer of the doctor and the patient can view them.
Expected Output	Free consultation done successfully.
Actual Output	Free consultation done successfully.
Status	The Test Was Successfully Performed.

6.6.1.22 Test Case: 22

Table 91: Test Case of Paid Consultation

Test case	22
Objective	Paid Consultation
Pre-Condition	Patient must log in to the app.
Flow	1. Patient is at home page.

	2. Patient click on paid consultation module. 3. Patient select the Specialization field. 4. Patient selects specialist from the list of specialist and can view his profile and review & ratings. 5. Patient enter date and time, and describe issue by title. 6. Then click on Book Appointment. 7. Before appointment is booked, the patient enters mobile no in order to send payment by clicking on payment button. 8. System display message that "your transaction is successful". 9. After payment, a consultation request is sent to the doctor for confirmation. 10. Doctor receive notification that "New consultation request". 11. Doctors can accept or reject it depending on their feasibility. 12. System will update the wallet.
Expected Output	Paid consultation done successfully.
Actual Output	Paid consultation done successfully.
Status	The Test Was Successfully Performed.

6.6.1.23 Test Case: 23

Table 92: Test Case of View Wallet

Test case	23
Objective	View Wallet
Pre-Condition	User must log in to the app.
Flow	1. User is at home page. 2. User click on drawer. 3. User selects wallet. 4. User can view their transaction history. 5. System displays transaction history successfully.
Expected Output	The Wallet is shown successfully.
Actual Output	The Wallet is shown successfully.
Status	The Test Was Successfully Performed.

6.6.2 Admin Management:

6.6.2.1 Test Case: 24

Table 93: Test Case of View Appointment

Test case	24
Objective	View Appointment
Pre-Condition	The admin should be login.
Flow	1. Admin is at Dashboard. 2. Admin click on drawer. 3. Admin select view Appointment screen.

	4. System display all the appointment successfully.
Expected Output	View appointment Successfully.
Actual Output	View appointment Successfully.
Status	The Test Was Successfully Performed.

6.6.2.2 Test Case: 25

Table 94: Test Case of View Feedback

Test case	25
Objective	View Feedback
Pre-Condition	Admin should be login.
Flow	1. Admin is at Dashboard. 2. Admin click on drawer. 3. Admin select view feedback screen. 4. System display all the Feedback of users successfully.
Expected Output	View Feedback Successfully.
Actual Output	View Feedback Successfully.
Status	The Test Was Successfully Performed.

6.6.2.3 Test Case: 26

Table 95: Test Case of View Statistics

Test case	26
Objective	View Statistics
Pre-Condition	Admin should be login.

Flow	1. Admin is at Dashboard. 2. Admin can view statistics yearly, monthly or weekly by clicking on that button. 3. System display all the Statistics of users successfully.
Expected Output	View Statistics Successfully.
Actual Output	View Statistics Successfully.
Status	The Test Was Successfully Performed.

6.6.2.4 Test Case: 27

Table 96: Test Case of Delete Doctor

Test case	27
Objective	Delete Doctor
Pre-Condition	The admin should be login.
Flow	1. Admin is at Dashboard. 2. Admin click on drawer. 3. Admin select Doctor Verification. 4. System displays a list of all doctors. 5. By clicking on the delete icon, the admin can delete doctors. 6. System display message “Doctor deleted successfully”.
Expected Output	Doctor deleted Successfully.
Actual Output	Doctor deleted Successfully.
Status	The Test Was Successfully Performed.

6.6.2.5 Test Case: 28

Table 97: Test Case of Delete Patient

Test case	28
Objective	Delete Patient
Pre-Condition	The admin should be login.
Flow	1. Admin is at Dashboard. 2. Admin click on drawer. 3. Admin select Manage Patients. 4. System displays a list of all Patients. 5. By clicking on delete icon, admin can delete patient. 6. System display message “Patient deleted successfully”.
Expected Output	Patient deleted Successfully.
Actual Output	Patient deleted Successfully.
Status	The Test Was Successfully Performed.

6.6.2.6 Test Case: 29

Table 98: Test Case of Update Patient

Test case	29
Objective	Update Patient
Pre-Condition	The admin should be login.

Flow	1. Admin is at Dashboard. 2. Admin click on drawer. 3. Admin select Manage Patients. 4. System displays a list of all Patients. 5. By clicking on the Edit icon, the admin can Update patient record. 6. System display message “Patient Updated successfully”.
Expected Output	Patient Updated Successfully.
Actual Output	Patient Updated Successfully.
Status	The Test Was Successfully Performed.

6.6.2.7 Test Case: 30

Table 99: Test Case of Update Doctor

Test case	30
Objective	Update Doctor
Pre-Condition	Admin should be login.
Flow	1. Admin is at Dashboard. 2. Admin click on drawer. 3. Admin select Doctor verification. 4. System displays list of all Doctor. 5. By clicking on Edit icon, the admin can Update Doctor’s record. 6. System display message “Doctor Updated successfully”.

Expected Output	Doctor Updated Successfully.
Actual Output	Doctor Updated Successfully.
Status	The Test Was Successfully Performed.

6.6.2.8 Test Case: 31

Table 100: Test Case of View Doctor

Test case	31
Objective	View Doctor
Pre-Condition	Admin should be login.
Flow	1. Admin is at Dashboard. 2. Admin click on drawer. 3. Admin select Doctor verification. 4. System displays list of all Doctor. 5. By clicking on the view icon, the admin can view Doctor record. 6. System display record successfully.
Expected Output	Doctor viewed Successfully.
Actual Output	Doctor viewed Successfully.
Status	The Test Was Successfully Performed.

6.6.2.9 Test Case: 32

Table 101: Test Case of View Patient

Test case	32
Objective	View Patient
Pre-Condition	The admin should be login.

Flow	1. Admin is at Dashboard. 2. Admin click on drawer. 3. Admin select Manage patient. 4. System displays list of all Patient. 5. By clicking on view icon, admin can view the Patient record. 6. System display record successfully.
Expected Output	Patient viewed Successfully.
Actual Output	Patient viewed Successfully.
Status	The Test Was Successfully Performed.

6.6.2.10 Test Case: 33

Table 102: Test Case of Add User

Test case	33
Objective	Add user
Pre-Condition	The admin should be login.
Flow	1. Admin is at Dashboard. 2. Admin click on the drawer. 3. Admin select Manage patient. 4. By clicking on Add user icon, the admin can add users by entering mail and password. 5. System display Message that "User added successfully".
Expected Output	User added Successfully.
Actual Output	User added Successfully.
Status	The Test Was Successfully Performed.

6.6.2.11 Test Case: 34

Table 103: Test Case of Verify Doctor

Test case	34
Objective	Verify Doctor
Pre-Condition	The admin should be login.
Flow	1. Admin is at Dashboard. 2. Admin click on the drawer. 3. Admin select Doctor verification. 4. The system displays the list of all Doctor. 5. By clicking on view icon, admin can view approve the doctors by looking at their authentic data. 6. System successfully approves doctor.
Expected Output	Verify the doctor Successfully.
Actual Output	Verify the doctor Successfully.
Status	The Test Was Successfully Performed.

6.7 Conclusion

In this chapter, we provide all the detailed information about the phases of testing that this project has been through. There was a total of three phases of testing i.e. Unit testing, component testing, and integration testing.

Chapter # 7

Conclusion

Chapter 7

Conclusion

Smart Health Consulting app has the potential to change the way people access health care services, providing them with the opportunity to get the guidance and services they need without having to go through the traditional health care system. Our app is useful for people to get a consultation from home without leaving their comfort zone. Users are given the ability to ask questions and get instant feedback on their health concerns

The report started by giving a brief introduction of the project and explaining the rationale behind the project. The report then dived into the literature that readers needed to be acquainted with to understand the project and its rationale well. Then we follow the development process of the Smart health consulting app right from its requirements to its design, implementation, and finally, it's testing, in accordance with the software development lifecycle. This document serves as a technical document that was used by designers and developers during the design and implementation phases.

In this project, we will be creating a mobile application in which we use different technologies. The “Smart health consulting app” is helpful for the patient to select the list of specialized doctors. Then the doctor can confirm an appointment according to their feasibility. The project was successfully made in the given time. The goal of this project is to optimize the work of patients and doctors. It facilitates the interaction between doctor and patient. It is an effortless, efficient, and influential mobile application for society. Smart health consultation services will be at a rapid pace in the future based on providing more convenient and efficient medical services.

7.1. Contributions

All the requirements related to technical and non-technical constraints are covered in this project report. This mobile application tries to solve the problems related to online consultations with the doctor.

7.2. Reflections

7.2.1. Strengths:

1. Simple yet captivating and meaningful interface which is easy to navigate and use.
2. Our application supports different languages so that users can use any language according to their feasibility.
3. Users can use our application in dark themes as well, as Dark mode apps can prolong the battery life of your smartphone and are better for reducing eye strain in low light conditions.
4. Most of the work is automatically done by the application, with one to two clicks required to perform most tasks.
5. Little or no learning is required for use.

7.2.2. Weaknesses:

Currently only available on android and iOS platforms.

7.2.3. Disciplined Project Management:

The key to the successful completion of a project and most importantly making the correct product is the conduction of proper and disciplined project management, and through the course of the project, we learned the importance of managing the time and the resources simultaneously to meet the schedule and the plan of the respective project and what effect these factors can have if they are not dealt with the cautious effort.

7.2.4. Importance of Team Communication

Another aspect of the importance we got familiar with is the role the communication between the team members which is the primary goal of all projects. The communication should be open, honest, and fair so that everyone in the team can contribute his/her best and never feel like their ideas and contributions are minimized or marginalized. To get the best result, it is important to maintain a good relationship with all the team members.

7.3. Future work

7.3.1. Other platforms

Smart health consulting application has been developed for android and iOS-based mobile devices due to the large market of consumers present in our country, however, our product can be developed for web platforms too, in the future.

7.3.2. Additional features

One of the attractive features that can be included in this product is that users will be able to order medicines online and book lab tests. Another feature is that users will be able to view all hospitals in the current location.

REFERENCES

- [1] Aavula, Ravi, et al. “Smart Health Consulting Android System.” *International Journal of Innovative Research in Science, Engineering and Technology (an ISO*, vol. 6, no. 3, Mar. 2017, www.ijirset.com/upload/2017/march/89_SOLUTION.pdf. Accessed 6 June 2022.

- [2] Al-Mahdi, Ibrahim, et al. “Online Medical Consultation.*” A Review of Literature and Practice*. 30 Jan. 2015, crpit.scem.westernsydney.edu.au/confpapers/CRPITV164Al-Mahdi.pdf. Accessed 6 June 2022.

- [3] Kuber, Kshitij, et al. “Smart Health Consulting Android System.” 2020, ijesc.org/upload/1ee10d2bb873c45c8b6833dbb85cb8c0.Smart%20Health%20Consulting%20Android%20System.pdf. Accessed 6 June 2022.

- [4] Vaswani, Naveen, et al. “Modified and Advanced System for Health Care Application.” 2018, www.irjet.net/archives/V5/i3/IRJET-V5I3680.pdf. Accessed 6 June 2022.

- [5] Kotevski, Aleksandar, et al. “(PDF) E-Health Monitoring System.” *ResearchGate,* June 2016, www.researchgate.net/publication/304919050_E-health_monitoring_system. Accessed 6 June 2022.

- [6] Singh, Ajeet PAL, et al. “(PDF) Online Medical Consultation: A Review.” *ResearchGate,*Mar.2018,www.researchgate.net/publication/323973550_Online_medical_consultation_a_review. Accessed 6 June 2022.

- [7] Pham, Minh, et al. “Delivering Home Healthcare through a Cloud-Based Smart Home Environment (CoSHE).” *Future Generation Computer Systems*, vol. 81, Apr. 2018, pp. 129–140, www.sciencedirect.com/science/article/pii/S0167739X17302194. Accessed 6 June 2022.

- [8] Taiwo, Olutosin, and Absalom E. Ezugwu. “Smart Healthcare Support for Remote Patient Monitoring during Covid-19 Quarantine.” *Informatics in Medicine Unlocked*, vol. 20, 2020, www.ncbi.nlm.nih.gov/pmc/articles/PMC7490242/. Accessed 6 June 2022.

- [9] Habib, Anam, et al. “(PDF) Thesis-ANDROID-BASED HEALTH-CARE MANAGEMENT SYSTEM.” *ResearchGate*, July 2016, www.researchgate.net/publication/305787548_Thesis-ANDROID-BASED_HEALTH-CARE_MANAGEMENT_SYSTEM. Accessed 6 June 2022.

- [10] Tiwari, Prashant, et al. “SMART HEALTH CARE (an ANDROID APP to PREDICT DISEASE on the BASIS of SYMPTOMS.” *Www.academia.edu*, vol. 4, no. 4, Apr. 2017, www.academia.edu/33518986/SMART_HEALTH_CARE_AN_ANDROID_APP_TO_PREDICT_DISEASE_ON_THE_BASIS_OF_SYMPTOM. Accessed 6 June 2022.

- [10] Tiwari, P., Jaiswal, A., Vishwakarma, N., & Patel, P. (2017, June 17). Smart health care (an Android app to predict disease on the basis of symptoms. Academia.edu. Retrieved June 6, 2022, from https://www.academia.edu/33518986/SMART_HEALTH_CARE_AN_ANDROID_APP_TO_PREDICT_DISEASE_ON_THE_BASIS_OF_SYMPTOM.

APPENDIX A: GLOSSARY

User: The patient and doctor who are the beneficiary of the system.

Doctor: The user with a doctor role.

Patient: The user with the patient role.

Rating: The average score given by the patient to the doctor.

Review: The comment made by a patient on a doctor's profile.

IOS: iPhone Operating system.

API: Application Programming Interface.

SHC: Smart Health Consulting App.

Feedback: The review made by users on the app.

SDLC: Software development life cycle.

ORIGINALITY REPORT

14%	9%	0%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	3%
2	irigs.iiu.edu.pk:64447 Internet Source	1%
3	Submitted to University of Greenwich Student Paper	1%
4	ukdiss.com Internet Source	1%
5	www.coursehero.com Internet Source	1%
6	Submitted to University College London Student Paper	<1%
7	Submitted to Curtin University of Technology Student Paper	<1%
8	Submitted to City University Student Paper	<1%
9	Submitted to Harrisburg University of Science and Technology	<1%