



**National University of Computer and Emerging Sciences**



## **SHIFA**

### **FYP Team**

Ummar Ikram.....19L-1049  
Muhammad Burhan.....19L-1105  
Umar Farooq.....19L-1045

**Supervised by**

**Mr. Saifullah Tanvir**

**FAST School of Computing**

**National University of Computer and Emerging Sciences**

**Lahore, Pakistan**

**October 2022**

## **Anti-Plagiarism Declaration**

This is to declare that the above publication produced under the:

### **Title: SHIFA (Secure Health and Intelligent Features Application)**

---

is the sole contribution of the author(s) and no part hereof has been reproduced on **as it is** basis (cut and paste) which can be considered as **Plagiarism**. All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is determined.

Date: Wednesday, 12 October

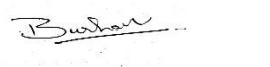
Student 1

Name: Ummar Ikram

Signature: 

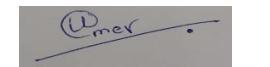
Student 2

Name: Muhammad Burhan

Signature: 

Student 3

Name: Umar Farooq

Signature: 

---

### **Authors' Declaration**

This states Authors' declaration that the work presented in the report is their own, and has not been submitted/presented previously to any other institution or organization.

## **Abstract**

SHIFA is a decentralized web application that will be built on blockchain technology. It will provide patients hassle-free access to their reports/prescriptions and allow them to communicate securely with doctors over the internet. Unfortunately, our local system still uses the traditional method of health record sharing and has not adopted digital infrastructure. Traditional paper prescription has a greater probability of error due to script writing and the fact that it is not occasionally preserved makes it difficult to maintain the patient history for better assessment [1]. In essence, SHIFA will be a platform that revolutionizes health technology and bring ease to patients and doctors all over the globe. Furthermore, SHIFA introduces an innovative offline record feature that enhances security even further. Each patient's offline record is fully secured with a system-generated password, providing an additional layer of protection. This allows patients to securely share their medical reports with any healthcare consultant of their choice, ensuring that the information is shared only once and remains confidential. By combining advanced encryption techniques, secure communication channels, and offline record security, SHIFA sets new standards for data protection in healthcare. This will enable efficient and accurate sharing of patient information, leading to faster diagnoses and better coordination of care. With its robust security measures, advanced functionalities, and commitment to improving healthcare delivery, SHIFA has the potential to revolutionize the way healthcare is accessed and delivered globally.

## **Executive Summary**

SHIFA is a decentralized web platform that aims to provide security, transparency, and control to its user through the use of blockchain technology and artificial intelligence. It will store sensitive health data as a non-fungible token and make use of cryptography to provide secure communication. Patients can greatly benefit from this product as it can help in early disease detection using disease classification models [2]. Moreover, it will provide a platform for small laboratories and lab doctors to check patients; by using our online platform, doctors can earn money. Related products are either centralized or do not have artificial intelligence incorporated, making SHIFA unique and appealing [3].

Big corporations such as Amazon, Apple, and Google are using natural language processing in their flagship products such as Alexa, Siri and LaMDA to listen, capture, analyze and utilize healthcare data [4]. Whereas, healthcare companies have started to develop their blockchains to provide security and integrity. The industry trend will most likely be to incentivize patients in exchange for sharing their data anonymously with research organizations [5].

The product will heavily follow a micro-services architecture to ensure reliability, usability, and modularity. Sensitive user data will be stored on blockchain whereas dynamic data will be encrypted and settled on cloud databases. Traditional interactions between doctor and patient do not provide a feasible way to maintain patient health history because they rely on paper reports and prescriptions. SHIFA aims to solve this grave issue by making use of electronic prescriptions and AI-assisted to remove the hassle of safekeeping reports and make the life of patients easier.

Data breaches cost an organization \$3.86 million every year, with the healthcare industry suffering the effects of the consequences. [6]. The exchange of medical information between facilities and specialists could lead to better treatment but also has the factor of a security breach. The solution for this is blockchain technology. An enormous amount of data is kept in a distributed ledger by blockchain technology built around a secure cryptography system. It not securely stores data but also eliminates the middleman. However, the acceptance of cryptocurrency is still in progress for some countries so that would be a challenge [7].

## Table of Contents

Table of Contents .....	i
List of Tables .....	v
List of Figures .....	vi
Chapter 1: Introduction .....	1
1.1 Purpose of this Document .....	1
1.2 Intended Audience .....	1
1.3 Definitions, Acronyms, and Abbreviations .....	1
1.4 Conclusion .....	3
Chapter 2: Project Vision.....	4
2.1 Problem Domain Overview .....	4
2.2 Problem Statement.....	4
2.3 Problem Elaboration .....	4
2.4 Goals and Objectives .....	4
2.5 Project Scope .....	5
2.6 Sustainable Development Goal (SDG) .....	5
2.7 Constraints .....	5
2.8 Business Opportunity.....	6
2.9 Stakeholders Description/ User Characteristics.....	6
2.9.2 Stakeholders Summary .....	6
2.9.3 Key High-Level Goals and Problems of Stakeholders .....	6
2.10 Conclusion .....	7
Chapter 3: Literature Review / Related Work .....	8
3.1 Methods/Algorithms .....	8
3.2 Related Products .....	8
3.3 Literature Review Summary Table .....	9
3.4 Conclusion .....	9
Chapter 4: Software Requirement Specifications .....	10
4.1 List of Features .....	10
4.1.1 Authentication.....	10
4.1.2 Secure communication.....	10
4.1.3 NFT Reports.....	10
4.1.4 Digital currency .....	10
4.1.5 E-prescriptions .....	11
4.1.6 Disease Classification .....	11
4.2 Functional Requirements .....	11
4.2.1 Account .....	11
4.2.2 Doctor .....	11
4.2.3 Patients .....	12
4.3 Quality Attributes.....	13
4.3.1 Reliability.....	13
4.3.2 Usability .....	13
4.3.3 Supportability.....	13
4.4 Non-Functional Requirements .....	13
4.4.1 Performance Requirements .....	13
4.4.2 Availability .....	14
4.4.3 Security .....	14
4.5 Assumptions.....	14
4.6 Hardware and Software Requirements .....	14
4.6.1 Hardware Requirements.....	14

4.6.2 Software Requirements .....	15
4.7 Use Cases .....	15
4.7.1 Login .....	15
4.7.2 Patient Sign Up .....	16
4.7.3 Doctor Sign Up .....	16
4.7.4 View Profile .....	18
4.7.5 Edit Profile .....	18
4.7.6 View Free Slots .....	19
4.7.7 Add Free Slot .....	19
4.7.8 Edit Free Slot .....	20
4.7.9 Delete Free Slots .....	20
4.7.10 View Doctors .....	21
4.7.11 View Doctor Profile .....	21
4.7.12 Get Appointment .....	21
4.7.13 View Appointments .....	22
4.7.14 Cancel Appointments .....	23
4.7.15 Filter Appointments .....	23
4.7.16 Add E-Prescription .....	24
4.7.17 Download Prescription .....	24
4.7.18 QR Code of Prescription .....	25
4.7.19 Preview Prescription .....	25
4.7.20 Filter Prescription: .....	26
4.7.21 User Chat List .....	26
4.7.22 Chat Room .....	27
4.7.23 Start New Chat .....	27
4.7.24 Send Message .....	28
4.7.25 Download Reports .....	28
4.7.26 View Reports .....	28
4.7.27 Add Reports .....	29
4.7.28 Filter Report .....	30
4.7.29 Logout .....	30
4.8 Graphical User Interface .....	31
4.9 Database Design .....	36
4.9.1 ER Diagram .....	37
4.9.2 Data Dictionary .....	37
4.10 Risk Analysis .....	41
4.10.1 Business Risk .....	41
4.10.2 Technical Risk .....	41
4.10.3 Technology Risks .....	42
4.11 Conclusion .....	42
<b>Chapter 5: High-Level and Low-Level Design .....</b>	<b>43</b>
5.1 System Overview .....	43
5.2 Design Considerations .....	43
5.2.1 Assumptions and Dependencies .....	43
5.2.2 General Constraints .....	44
5.2.3 Goals and Guidelines .....	45
5.2.4 Development Methods .....	46
5.3 System Architecture .....	46
5.3.1 Doctor Authentication .....	47
5.3.2 E-Prescription/Report Minting .....	48

5.3.3 Digital Currency.....	49
5.4 Architectural Strategies.....	49
5.4.1 Technology Stack.....	49
5.4.2 Reuse of existing software components.....	50
5.4.3 Future plans for extending or enhancing the software.....	50
5.4.4 User interface paradigms .....	51
5.4.5 Error detection and recovery.....	51
5.4.6 External databases and/or data storage management and persistence .....	51
5.4.7 Distributed data or control over a network .....	52
5.4.8 Concurrency and synchronization.....	52
5.4.9 Communication mechanisms .....	52
5.5 Class Diagram.....	53
5.6 Sequence Diagrams.....	54
5.7 Policies and Tactics.....	68
5.7.1 Clarinet Compiler for Smart Contract.....	68
5.7.2 Tools to be used .....	69
5.7.3 Coding guidelines and conventions .....	69
5.7.4 Testing.....	69
5.7.5 Maintenance .....	69
5.7.6 Interfaces for end-users.....	69
5.8 Conclusion .....	69
Chapter 6: Implementation and Test Cases .....	70
6.1 Implementation .....	70
6.1.1 Pakistan Medical Council (PMC) .....	70
6.1.2 Disease Classification Models .....	70
6.1.3 Non-Fungible Token Generation .....	70
6.1.4 Asymmetric/Symmetric Cryptography .....	71
6.1.5 Gaia Decentralized Storage.....	71
6.1.6 Wallet Authentication .....	71
6.1.7 Appointment Booking.....	71
6.1.8 Patient Records .....	71
6.1.9 Payment Gateway .....	72
6.2 Test case Design and description .....	72
6.2.1 User Login .....	74
6.2.2 Hiro Wallet Extension.....	75
6.2.3 User Login Alternative .....	76
6.2.4 Patient Sign Up .....	77
6.2.5 Patient Sign Up Alternative .....	78
6.2.6 Doctor Sign Up .....	79
6.2.7 Doctor Sign Up Alternative 1 .....	80
6.2.8 Doctor Sign Up Alternative 2 .....	81
6.2.9 View Profile .....	82
6.2.10 Update Profile .....	83
6.2.11 View Free Slots .....	84
6.2.12 Add Free Slots.....	85
6.2.13 Add Free Slots Alternative.....	86
6.2.14 Edit Free Slots.....	87
6.2.15 Edit Free Slots Alternative.....	88
6.2.16 Delete Free Slots .....	89
6.2.17 View Doctors .....	90

6.2.18 View Doctors Profile .....	91
6.2.19 Schedule Appointments .....	92
6.2.20 Schedule Appointments Alternative .....	93
6.2.21 View Appointments .....	94
6.2.22 Cancel Appointments.....	95
6.2.23 Filter Appointments .....	96
6.2.24 Add E-Prescription.....	97
6.2.25 Add E-Prescription Alternative.....	98
6.2.26 Download E-Prescription.....	99
6.2.27 QR Code E-Prescription .....	100
6.2.28 Preview E-Prescription .....	101
6.2.29 Doctor View Record .....	102
6.2.30 Patient View Record .....	103
6.2.31 Filter E-Prescription.....	104
6.2.32 Filter E-Prescription Alternative.....	105
6.2.33 User Chat List .....	106
6.2.34 Start New Chat.....	107
6.2.35 Start New Chat Alternative.....	108
6.2.36 Send Message.....	109
6.2.37 Send Message Alternative.....	110
6.2.38 Download Report.....	111
6.2.39 Preview Reports .....	112
6.2.40 Add Reports .....	113
6.2.41 Add Report Alternative.....	114
6.2.42 Filter Report .....	115
6.2.43 Filter Report Alternative.....	116
6.2.44 Log out .....	116
6.3 Test Metrics .....	117
6.4 Conclusion .....	117
<b>Chapter 7: User Manual .....</b>	<b>118</b>
7.1 Register .....	118
7.2 Login .....	118
7.3 Book Appointments (Only Patients) .....	118
7.4 View Appointments .....	118
7.5 Send Message.....	118
7.6 View Messages .....	119
7.7 Upload Reports (Lab Technician).....	119
7.8 View Reports .....	119
7.9 Upload Prescriptions (Doctor) .....	119
7.10 View Prescriptions .....	119
7.11 Troubleshoot .....	119
7.12 Conclusion .....	120
<b>Chapter 8: Conclusion and Work Completed .....</b>	<b>121</b>
8.1 Completed Work .....	121
8.2 Findings/Results.....	121
8.3 Challenges .....	121
8.4 Recommendations .....	121
8.5 Conclusion .....	121
<b>References .....</b>	<b>123</b>

**List of Tables**

Table 1: Definitions, Acronyms, and Abbreviations .....	1
Table 2: Problem Statement.....	4
Table 3: Related Products .....	8
Table 4: Features of Related Products .....	9
Table 5: Hardware Requirements .....	14
Table 6: Doctor .....	37
Table 7: Qualification .....	38
Table 8: Service .....	38
Table 9: Free Slot.....	38
Table 10: Patient .....	38
Table 11: Complication.....	39
Table 12: Allergy .....	39
Table 13: Chat Room .....	39
Table 14: Chat Messages .....	39
Table 15: Appointment .....	39
Table 16: Reports.....	40
Table 17: Additional Details.....	40
Table 18: Report Details .....	40
Table 19: Prescription .....	40
Table 20: Drugs.....	41
Table 21: Prescription Drugs .....	41
Table 22: Test Case ID and Name Mapping.....	72

## List of Figures

Figure 1: Signup Page .....	31
Figure 2: Doctor's Dashboard Page .....	31
Figure 3: Doctor's Report Page .....	32
Figure 4: Doctor's Appointment Page .....	32
Figure 5: Doctor's Prescription Page .....	33
Figure 6: User's Message Page .....	33
Figure 7: Doctor's Profile Page .....	34
Figure 8: Patient Dashboard Page .....	34
Figure 9: Patient Reports Page .....	35
Figure 10: Patient Appointment Page .....	35
Figure 11: Patient Prescription Page .....	36
Figure 12: Patient Profile Page .....	36
Figure 13: ER Diagram .....	37
Figure 14: System Architecture .....	46
Figure 15: Doctor Authentication Subsystem Architecture .....	47
Figure 16: E-Prescriptions Subsystem Architecture .....	48
Figure 17: Digital Currency Subsystem Architecture .....	49
Figure 18: Class Diagram .....	53
Figure 19: Signup Sequence .....	54
Figure 20: Login Sequence .....	55
Figure 21: Edit Profile Sequence .....	55
Figure 22: View Profile Sequence .....	56
Figure 23: Add Free Slot Sequence .....	56
Figure 24: Delete Free Slot Sequence .....	57
Figure 25: Edit Slot from Schedule Sequence .....	57
Figure 26: Delete slot from schedule Sequence .....	58
Figure 27: View Doctors Sequence .....	58
Figure 28: Doctors Profile Sequence .....	59
Figure 29: Cancel Appointment Sequence .....	59
Figure 30: Filter Appointment Sequence .....	60
Figure 31: Get Appointment Sequence .....	60
Figure 32: View Appointment Sequence .....	61
Figure 33: Add Prescription Sequence .....	61
Figure 34: Download E-Prescription Sequence .....	62
Figure 35: Filter E-Prescription Sequence .....	63
Figure 36: Preview Prescription Sequence .....	64
Figure 37: Logout Sequence .....	64
Figure 38: QR-Code Prescription Sequence .....	65
Figure 39: User Chat Sequence .....	65
Figure 40: Download Reports Sequence .....	66
Figure 41: Chat Room Sequence .....	66
Figure 42: Send Message Sequence .....	67
Figure 43: Start New Chat Sequence .....	68

## Chapter 1: Introduction

The demand for secure and convenient healthcare has increased drastically in recent times [14]. The traditional communication between doctors and patients is restricted to physical proximity and the use of paper prescriptions creates difficulty in maintaining patient medical history between different doctors. Digital ownership is not transparent in existing mechanisms and due to the sensitivity of health data, it is crucial to protect it from unauthorized access. Moreover, with the increase in population, it is only feasible that we take some load off doctors [13] and intelligently predict, analyze and assist doctors regarding their patients' health statistics.

SHIFA will be a platform that revolutionizes health technology and bring ease to patients and doctors all over the globe. Existing applications are centralized and do not offer the security and transparency that SHIFA will offer its users.

Project vision covers the complete domain overview, user characteristics and details about stakeholders, goals and problems. The literature review contains information of all work done in the past. It contains both a detailed description of the related work as well as the summary. SRS contains the requirements of the project, which we will need to implement the application and the detailed design.

### 1.1 Purpose of this Document

The objective of this document is to give an overview of the entire project, including the project's purpose, scope, terminology, references, and an overview of the SRS. By elaborating the issue statement in detail, this document intends to acquire, research, and provide an in-depth understanding of the entire Secure Health and Intelligent Features Application. However, it includes an overview and needs of the client while developing products with high features. This document will provide a detailed view of SHIFA.

### 1.2 Intended Audience

The intended audience for this document is

- FYP Advisors/Evaluators: These individuals assist developers and give them guidance.
- Developers: These are individuals working on developing the product for the end user.
- Testers: These are individuals working on testing the functionality of the software.

### 1.3 Definitions, Acronyms, and Abbreviations

**Table 1: Definitions, Acronyms, and Abbreviations**

*Table includes the list of acronyms and abbreviations used in this document*

<b>Blockchain</b>	Blockchain is a distributed, immutable ledger that may be used to keep track of transactions, keep an eye on assets, and establish trust.
<b>Artificial Intelligence (AI)</b>	The concept of artificial intelligence is the ability of technology, particularly computer systems, to mimic human intelligence processes.
<b>Decentralized</b>	To divide (a central authority's) administrative powers or tasks over an area less intense area.

<b>Smart Contracts</b>	Smart contracts are obviously programs that run when certain criteria were met and are stored on a blockchain.
<b>Non-Fungible Tokens (NFTs)</b>	A blockchain record that is connected to a specific digital or physical asset is known as a non-fungible token.
<b>Fungible- Token</b>	Fungible tokens or assets are non-unique and replaceable. For example, fiat currencies such as the US dollar are fungible.
<b>Cryptography</b>	Protocols that prohibit unauthorized third parties from reading private messages.
<b>Encryption</b>	To prevent unauthorized access, convert (information or data) into a cypher.
<b>Asymmetric Encryption</b>	For encryption and decryption, asymmetric encryption utilizes a simple mathematical associated pair of keys.
<b>E-Prescription</b>	E-Prescribing allows doctors to send a new prescription to pharmacists on a patient's behalf.
<b>Blockchain Wallet</b>	A Blockchain wallet is a type of digital wallet that facilitates individuals to store, monitor, and exchange cryptocurrencies.
<b>Minting/Mint</b>	To mint an NFT, digital assets is transformed into crypto coins or digital assets that are maintained on the blockchain.
<b>Disease Classification</b>	The act or process of classifying/predicting the chance of a particular disease to occur
<b>Static Site Generation (SSG)</b>	A software that builds a complete static HTML website from raw data and a collection of templates.
<b>Primary Key (PK)</b>	The primary Key is a unique identifier that is unique for each record.
<b>Foreign Key (FK)</b>	A field (or group of fields) in one table that connects to the PRIMARY in another table is known as a FOREIGN KEY.
<b>Composite key (CK)</b>	A composite key combines many fields or columns from a single table to represent a unique record.
<b>LaMDA</b>	Google's Language Model for Dialogue Applications is a series of conversational neural language systems.
<b>Fyp</b>	Final Year Project

## **1.4 Conclusion**

The chapter explicitly describes the objective of the document and states the anticipated audience being developers, testers and fyp evaluators. Furthermore, for the ease of the intended audience, the chapter illustrates the various definitions, abbreviations, and acronyms for all the technical terms used throughout the document.

## Chapter 2: Project Vision

### 2.1 Problem Domain Overview

The domain of our project focuses on health problems where patients can communicate with the doctor securely (all users will be identified based on their unique wallet address which ensures anonymity) and SHIFA will make use of its own digital currency that will work as a fungible- token to act as a payment method for appointments with doctors. Moreover, the patient will have complete control over the digital asset and can choose whom to share it with and his reports will be minted by the doctor as a non-fungible token (NFT) the blockchain.

Furthermore, doctors will be able to electronically prescribe medication to patients that will be password protected and dynamically hosted on the web application. Lastly, Disease classification models will output a discrete answer based on the disease parameters passed to it. It will help in timely diagnosis and will improve quality of life.

### 2.2 Problem Statement

**Table 2: Problem Statement**

*The problem statement with their solution is given below.*

<b>Problem</b>	Health data needs immense security over the internet because of its sensitivity and users should have complete control of their data. Furthermore, the increase in unemployment of doctors [15] is a serious issue that needs to be addressed.
<b>Effect</b>	There is limited work for doctors and it is hard to maintain patient history using traditional paper methods. Paper wastage has increased drastically [10].
<b>Impact</b>	It is difficult for patient to preserve their paper prescriptions and understand them .Data breaches have cost companies millions and the rising eco-friendly wave of reducing paper consumption requires going digital [6].
<b>Solution</b>	SHIFA will make use of Blockchain technology to secure patient data and give complete control to users for their digital assets. Moreover, it will provide a platform for doctors to earn by providing services to patients.

### 2.3 Problem Elaboration

Unfortunately, our local system still uses the traditional method of health record sharing and has not adopted digital means. Traditional paper prescription has a greater tendency of error due to script writing and the fact that it is not occasionally preserved which makes it difficult to maintain patient history for better assessment [1]. Moreover, it is troublesome for patients to safe-keep and carry their paper prescription every time they visit a pharmacy.

Moreover, there are very limited platforms that provide transparent and convenient doctor appointment system for patients. Furthermore, with the increase in population, it is only feasible that we take some load off doctors and intelligently predict, analyze and assist doctors regarding their patients' health statistics.

### 2.4 Goals and Objectives

The application will have the following key aspects:

- Easiness in communication between patient and doctor
- Provide patient a complete authority over the reports.
- Appointment time can be set between doctor and patient to discuss health issues.
- Allow patients to get in touch with different kind of doctors.
- Overcome the difficulty of reading and understanding doctor written prescriptions.
- Create easiness for doctor, pharmacist and patients.

## 2.5 Project Scope

SHIFA will consist of creating a web application based upon a solution provided by us. For the front-end, we will be using Next.JS, a robust react framework that allows server-side rendering (SSR) and the other is static site generation (SSG) that greatly accelerate and speeds up the web page loading for the client [18].

Moving on to the back-end, we are using a hybrid model of data storage. All the conversations between the doctor and patient will be end-to-end encrypted using asymmetric encryption, stored inside MongoDB off-chain storage for fast retrieval [19]. The report information and other sensitive data will be stored on the blockchain as on chain storage.

Furthermore, we are incorporating artificial intelligence as micro-services. For instance, disease classification models will be trained using python and the saved models will be deploying as REST API using Flask on Heroku Cloud [20]. They can then be called from the main application.

## 2.6 Sustainable Development Goal (SDG)

Our First Sustainable Goal is to provide *GOOD HEALTH AND WELL BEING* because our application is one of the applications, which will be providing patients convenient access to their medical reports and allow them to communicate with doctors over the internet. Moreover, Patients arrange appointment with doctor for reasonable prescription and medical history will be maintained conveniently.

The application assists the doctor based on the patient report for quick recovery of patient from disease. Second Sustainable Goal is *DECENT WORK AND ECONOMIC GROWTH* because SHIFA provide a platform to doctor for work and start earning money.

## 2.7 Constraints

The following are the six constraints that are recognized as determining factors in SHIFA project management:

- **Budget:** There is possibility that our deployment server, database and web hosting expenses start running out.
- **Time:** It is difficult for developers to manage development with studies and other obligations.
- **Resources:** Most of time it is difficult to find a health-related data set for a critical system with minimum errors.
- **Risk:** Government regulations regarding healthcare is very rigorous [16]. Moreover, the acceptance of cryptocurrency is still in progress for some countries [7].

## 2.8 Business Opportunity

SHIFA will be a platform for doctors and small-scale laboratories to earn digital currency in exchange for their valuable services.

## 2.9 Stakeholders Description/ User Characteristics

There will two kinds of users of our system i.e. Patient and Doctor.

1. Patient
2. Doctor
3. Lab Technician

### 2.9.1.1 Patient

Patient will be able to authenticate through blockchain wallets. All patients will be identified based on their unique wallet address, which ensures anonymity. Patients can communicate with the doctor securely (all users will be identified based on their unique wallet address which ensures anonymity). The patient will have complete control over the digital asset and can choose whom to share it with and his reports will be minted by the doctor as a non-fungible token (NFT) the blockchain. The patient can also choose appointment time from the free slots of doctor.

### 2.9.1.2 Doctor

Doctor will be able to authenticate through blockchain wallets. All Doctors will be identified based on their unique wallet address, which ensures anonymity. Doctors can communicate with the patient securely (all users will be identified based on their unique wallet address which ensures anonymity). Doctors can use disease classification models will output a discrete answer based on the disease parameters passed to it that will help in timely diagnosis and will improve quality of life. Doctors set the free slots to set the appointment with the patient.

## 2.9.2 Stakeholders Summary

All of the internal people and teams whom the SHIFA project will involve or affect are called its stakeholders. The different types of stakeholders:

- **Developers:** These are individuals working on developing the product for the end user.
- **User:** These individuals/customers such as doctors and patients will use the system functionality.
- **Organization:** This is an institution such as FAST NUACES, which oversees the project and approves it.
- **Supervisor:** These individual(s) assist developers and guide them.

## 2.9.3 Key High-Level Goals and Problems of Stakeholders

The following is a list of Stakeholders' Key-High Level Goals.

### 2.9.3.1 Key High-Level Goals:

- Developers of SHIFA want to ensure about the product is successful and serves the public interest.

- Patients want good quality and services at reasonable prices.
- Doctors want a platform where they can provide services and earn.
- Evaluator and organization want to ensure that the product developed follows good professional practices and standards.

### **2.9.3.2 Problems:**

Some of the important problems faced by stakes holder are as follow:

#### **2.9.3.2.1 Technological difficulty**

- The advancement in technologies creates difficulty for developers. They may take longer time than expected time to learn the tools and technologies required to do the project.
- Developers should understand the architecture of technology otherwise; he/she may move in the wrong direction.

#### **2.9.3.2.2 End User Training**

Users such as doctors and patients may face issues in using cryptocurrency wallets and digital currency. If the end user cannot be able to use a product, it will fail. End users need to test our application to provide feedback on requirements. It will help to determine in any additional training or consultants will be needed or not. Therefore, we should have to conduct a training session with our users to give an overview of the application.

#### **2.9.3.2.3 Stakeholders resistant to share information**

It is necessary for doctors to share general and professional information for authenticity. For instance, doctors are required to upload their degree, personal photo, and verify themselves through a live webcam. Moreover, patients have to trust the application with their previously known complications for a better, accurate experience.

#### **2.9.3.2.4 Resource constraints**

The doctor should have a functional web camera and microphone at the time of registration. Doctor workspaces consist of standalone tower systems that do not have embedded Microphones and web cameras so their organizations would have to arrange it for them.

#### **2.9.3.2.5 Government Regulation**

Health Regulations such as HIPAA and NHSRC have strict compliance regarding health care that needs to be approved for any organization to work. Moreover, the acceptance of cryptocurrency is still in progress for some countries such as Pakistan [13] whereas countries like Japan, Canada, and Germany have made cryptocurrency use legal.

## **2.10 Conclusion**

The chapter covers the problem statement and elaboration being health care security and ownership of patient data with their effect, impact, and the feasible solution being blockchain technology. SHIFA react-based web application that will ease the communication between doctor and patient, and provide earning opportunities to promote the economy, good health care and well-being. Finally, it describes the characteristics of stakeholders with their key high-level goals and potential problems.

## Chapter 3: Literature Review / Related Work

Artificial intelligence is playing an effective role in this regard. Machine learning is the main trend for artificial intelligence in healthcare, and it is used to analyse an enormous amount of patient data [2]. This means that doctors or patients will be provided recommended diagnoses, drugs, and treatment plans after reviewing the history of the patient along with current indications. This will help to reduce costs, improve outcomes, ease, and boost the work of staff.

According to Bain, a consultancy firm, by 2025, the market for medical services is anticipated to be worth \$70 billion. As the data on healthcare keeps increasing the treatment options also keep getting better so do the outcomes of the patients. However, transferring data from one healthcare organization to another is difficult.

### 3.1 Methods/Algorithms

Digital assistants like Alexa, Siri, and Google use Natural Language Processing and listening to capture, analyze and utilize healthcare data [4]. A new voice assistant that can listen to patient-doctor interactions has been released by AI startup Saykara without being prompted.

Personalized medicine uses an individualized genetic profile to prevent, diagnose and treat disease [17]. Knowing a patient's genetic background might help doctors choose the best drug for them. It also allows for the potential of selecting the appropriate dose for the appropriate person, as opposed to the "one size fits all" approach in medication therapy. As we understand more about individuals, there will be little medication that is unaffected by this. RNA-based treatments, for example, attempt to "interfere" with genetic information at the RNA level, preventing a genetic flaw from being translated into useful proteins.

### 3.2 Related Products

**Table 3: Related Products**  
*Following is a list of products related to SHIFA project.*

<b>Buoy Health</b>	Algorithms are used by an AI-based symptom and cure checker to identify and treat illnesses[22]. A Chabot examines to a patient's condition and then leads the patient to the appropriate care according on the diagnosis. Harvard Medical School, a hospital and healthcare provider, uses this to aid in the treatment and diagnosis of patients.
<b>BurstIQ</b>	It is a software platform that allows healthcare organizations to manage huge amounts of patient data in a secure and reliable manner [23]. Its blockchain platform allows the protection, selling, exchange, or licenses of data while following to HIPAA regulations.
<b>Medicalchain</b>	Medicalchain uses block chain technology to establish one source of truth while protecting the privacy of patient records [24]. Patient information that can be identified and protects the patient's private information from attack can be requested by medical specialists, healthcare facilities, and laboratories.
<b>Marham</b>	Marham is the Pakistani people's digital healthcare system. It is intended to address health-related and medical problems that both patients and doctors face [25]. It might assist you in locating Doctors, Clinicians, and Advisors in your area.

### 3.3 Literature Review Summary Table

**Table 4: Features of Related Products**  
*Features Comparison of related products with SHIFA.*

Feature	Marham	Medicalchain	BurstIQ	Buoy Health	SHIFA
Decentralized		✓	✓		✓
AI Assist				✓	✓
E-Prescription		✓	✓		✓
Appointments	✓				✓
E2E Messaging				✓	✓

### 3.4 Conclusion

Companies lose \$3.86 million a year due to data breaches, with the healthcare sector bearing the most losses [6]. The United States loses \$60 billion a year to Medicare scam. The most sensitive and valuable data about people are found in medical institutions, which are data gold mines. Sharing medical information among clinics and researchers might enhance therapy, but there is also the risk of a security breach. Blockchain technology offers a remedy for this. Blockchain technology maintains an enormous distributed ledger of data using a system of safe cryptography. Not only does it keep data securely, but it also eliminates the middleman. In this decentralized eco-system, research organizations would need to offer incentives to consumers to share their health data with them [11].

## Chapter 4: Software Requirement Specifications

A brief overview of the whole Software Requirements Specification (SRS) is provided, including an analysis at the Hardware and Software requirements as well as Functional and Non-Functional requirements. By describing the Use cases and ERD in detail, this paper attempts to discover, examine, and provide a comprehensive understanding into the whole Secure Health and Intelligent Features Application. While describing high-level product characteristics, it also focuses on the capabilities needed by the customer and their demands. In this document, overview of SHIFA will be covered in detail.

### 4.1 List of Features

The goal is to deliver basic healthcare care services online digitally to anyone, regardless individuals reside in a major city or an in far region. Users have access to these facilities via their internet service or by visiting any local cafe. The following are the reasons for developing this system:

- There are few or no doctors in distant areas.
- A deficiency of modern medical equipment
- There is no medical history or test information management.
- Patient get E-prescription

SHIFA will be providing patients convenient access to their reports through the web application and allow them to communicate securely with doctors over the internet. The application will have the following key aspects:

#### 4.1.1 Authentication

SHIFA will make use of blockchain wallets to authenticate users to the application. All users will be identified based on their unique wallet address, which ensures anonymity.

#### 4.1.2 Secure communication

The messaging system between the users will make use of the cryptographic keys to ensure End-to-End (E2E) Encryption. An associated public-private key pair will be generated for every user that will allow asymmetric encryption for all their communication through the application.

#### 4.1.3 NFT Reports

The patient will have complete control over the digital asset and can choose whom to share it with others. The patient reports will be minted by the doctor as a non-fungible token non-fungible token (NFT) the blockchain and the associated metadata will be encrypted and stored off-chain on a decentralized database.

#### 4.1.4 Digital currency

SHIFA will make use of its own digital currency that will work as a fungible- token to act as a payment method for appointments with doctors. The payments will be processed through smart contracts to ensure a trustless system.

#### **4.1.5 E-prescriptions**

Doctors will be able to electronically prescribe medication to patients that will be password protected and dynamically hosted on the web application. The patient will then be able to share that with any person by providing them the link with password that he/she can change later.

#### **4.1.6 Disease Classification**

Classification models will output a discrete answer based on the disease parameters passed to it. It will help in timely diagnosis and will improve quality of life.

### **4.2 Functional Requirements**

The objective of this section is to give a brief explanation of the major services provided by the product.

#### **4.2.1 Account**

1. Patient shall be able to create a Patient Account with basic signup credentials.
2. The doctor shall be able to create Doctor Account.
3. Doctors must have to upload their pictures use in Authentication.
4. System shall be to Authenticate Doctor using degree authentication and face-script recognition.
5. User shall be able to login their Account.
6. User shall be able to sign out their Account.
7. Users shall be able to edit their Accounts.
8. Users shall be able to delete their Accounts.

#### **4.2.2 Doctor**

Functionality of doctor entity that it should be able to perform are as given below:

##### **4.2.2.1 Medical Report**

1. Doctor shall be able to upload medical report of patient.
2. Doctor shall be able to view medical reports of their all patients.
3. Doctor shall be able to get assistance from system to diagnose disease and classify patient condition based on reports parameters.

##### **4.2.2.2 Appointment**

1. Doctor shall be able to view their appointments.
2. Doctor shall be able to confirm the appointment with patient.
3. Doctor shall be able to reject the appointment with patient.
4. The doctor shall be able to add his free slots to the schedule.
5. The doctor shall be able to update free slots.
6. Doctor shall be able to delete his free schedule slots.

7. Doctor shall be able to receive payment in the form of cryptocurrency

#### **4.2.2.3 E-prescriptions**

1. Doctor shall be able to write drugs detail in prescription.
2. Doctor shall be able to set quantity of drugs.
3. Doctor shall be able to set the start and end date of drug doses.
4. Doctor shall be able to upload prescription.

#### **4.2.2.4 Communication**

1. Doctor shall be able to respond patient query through chat messenger.
2. Doctor shall be able to delete chat.
3. Doctor shall be able to delete message.

### **4.2.3 Patients**

Functionality of Patient entity that it should be able to perform are as given below:

#### **4.2.3.1 Medical Report**

1. The patient shall be able to view all their reports uploaded by doctors.
2. The patient shall be able to download reports.
3. Patient shall be able to view the status of report.
4. Status include disease classification, update by doctors with system assistance.

#### **4.2.3.2 Communication**

1. Patient shall be able to view Doctors in chat room with their appointments.
2. The patient shall be able to communicate with all Doctors through chat messenger.
3. Patient shall be able to delete chats.
4. Patient shall be able to delete Message.

#### **4.2.3.3 E-prescriptions**

1. The patient shall be able to get E-prescriptions of doctor's with their appointment.
2. Patient shall be able to get QR code of E-prescriptions.
3. Each QR code must be secure with some password.
4. Patient shall be able to scan QR code to view E-prescription.
5. The patient shall be able to download E-prescription.

#### **4.2.3.4 Appointment**

1. The patient shall be able to view available doctors for appointment.
2. The patient shall be able to view the profile of doctor.
3. The patient shall be able to reserve an appointment online with the doctor.

4. The patient shall be able to view his appointment.
5. The system will be allowing patient to pay appointment fees using Shifa-Coin (Digital currency)
6. A patient shall be able to cancel his appointment.
7. Appointment should be cancel within limited amount of time.
8. Cancel appointment get back their appointment payment.

### **4.3 Quality Attributes**

The objective of this section is to give a brief explanation of the quality attributes provided by the product.

#### **4.3.1 Reliability**

1. If a server breaks, a backup server will be operational in less than an hour.
2. The system's mean time to repair after failure shall not exceed 90 seconds.
3. Data on our system is critical and should be available all the time therefore it will be ensured that the system is accessible 100% of the time

#### **4.3.2 Usability**

1. Our system's UI is simple to use
2. It is easy to navigate, and the headers, buttons, and error warnings are clear.
3. Color scheme of the user interface shall be easy on the eyes
4. There is no specific training required for the user. The user will be able to learn all the functionalities and features of the site within 3-5 attempts.

#### **4.3.3 Supportability**

1. The website will be compatible with IE10 and further versions.
2. The system will be compatible with all operating systems.
3. The system will run on the latest Chrome, IE-09, Firefox, Safari, and Opera Mini.

### **4.4 Non-Functional Requirements**

The objective of this section is to give a brief explanation of the non-functional requirements provided by the product.

#### **4.4.1 Performance Requirements**

1. The maximum number of people who can access the system concurrently depends on the limit set by our hosting/server provider, horruku in this case, and our server plan, "Premium."
2. After a user submits information to the system, the system must show them a confirmation message within 4 seconds.
3. System efficiency and performance will be improved. There will be as little wait time as possible.

4. System login/logout shall take less than 5 seconds, and the system shall support a number of simultaneous users

#### **4.4.2 Availability**

1. The system will be unavailable in case of maintenance and if the system is down it will be up and running in one-day
2. The system will be available 24x7

#### **4.4.3 Security**

1. The system will prevent any kind of breach of data.
2. System will be secured under an SSL license.
3. System will be compliance all state cyber security laws.
4. All users are authenticated by user name and a password.
5. Permission is assigned to the user according to their roles.
6. Only IT team members can access and maintain database servers locally.

### **4.5 Assumptions**

1. Users should be computer literate.
2. Intended users are educated enough to operate and understand how the system works.
3. Users should have a little knowledge of the English language.
4. Users should know basic usage of the internet and web browsers

### **4.6 Hardware and Software Requirements**

A brief description of both software and hardware prerequisites needed to create and implement the project is given below.

#### **4.6.1 Hardware Requirements**

1. Desktop PC/Laptop
2. Internet is required.

The minimum and recommended hardware specifications for the web application are shown in the following table.

**Table 5: Hardware Requirements**  
*Minimum and recommended hardware specifications for the application*

<b>Component</b>	<b>Minimum</b>	<b>Recommended</b>
<b>Processor</b>	1.9 GHz dual-core x86 or x64 CPU with the SSE2 instruction set	64-bit architecture dual-core CPU running at 3.3 GHz or greater and using the SSE2 instruction set
<b>Memory(GB)</b>	2	4 or more

## 4.6.2 Software Requirements

1. Visual Studio Code
2. Python
3. MongoDB Compass
4. Node
5. Hiro Wallet
6. Chrome

## 4.7 Use Cases

This section lists use cases description or scenarios below.

### 4.7.1 Login

<b>Name</b>	Login		
<b>Actors</b>	Doctor, Patient, Lab Technician		
<b>Summary</b>	The user must establish a connection to the wallet before being sent to the dashboard page following successful verification.		
<b>Pre-Conditions</b>	The user must already be listed in the database records, either manually or by one of the authorized users. The user must not be logged in.		
<b>Post-Conditions</b>	After successfully establishing the user's session, the user will be sent to the dashboard page.		
<b>Special Requirements</b>	Hiro wallet extension		
<b>Basic Flow</b>			
Actor Action	System Response		
1 The user opens the login page and the Hiro wallet extension is enabled.	2	The login page is displayed a popup with a "Connect Wallet" button.	
3 The user press on the "Connect Wallet" button.		4 The system gets the wallet address and verifies it, establishes a session for the user, and redirects the user to the dashboard page.	
<b>Alternative Flow</b>			
1 The user opens the login page and the Hiro extension is missing.	2-A	The system responds with an error message: <i>Hiro wallet not found.</i>	
3 The user presses the "Connect Wallet" button.		3-A The system responds with an error message: <i>No account associated with this wallet address.</i>	

### 4.7.2 Patient Sign Up

<b>Name</b>	Sign Up		
<b>Actors</b>	Patient		
<b>Summary</b>	The patient shall connect to the wallet. The patient shall provide a first name, last name, blood group, email, and the type of account (Patient or Doctor) on the signup form. Patients must select the patient account type and after the successful creation of the account, a new session is established and the system must redirect the patient to the “ <i>Dashboard</i> ” page.		
<b>Pre-Conditions</b>	The same wallet address must not be already in the database records.		
<b>Post-Conditions</b>	The patient’s session is successfully established and shall be redirected to the “ <i>Dashboard</i> ” page.		
<b>Special Requirements</b>	Hiro wallet extension		
<b>Basic Flow</b>			
<b>Actor Action</b>		<b>System Response</b>	
1	The patient opens the Signup page and the Hiro wallet extension is enabled.	2	The login page is displayed a popup with a “ <i>Connect Wallet</i> ” button.
3	The patient click on the “ <i>Connect Wallet</i> ” button.	4	The “ <i>Sign Up</i> ” page with the address of the associated cryptocurrency wallet is displayed asking for first name, last name, blood group, email, and the type of account (Patient or Doctor).
5	The patient enters first name, last name blood group, email, and the type of account (Patient)	6	The system checks if the wallet address and email are unique, creates an account for the patient, establishes a session, and redirects the patient to the “ <i>Dashboard</i> ” page

#### Alternative Flow

1	The patient opens the signup page and the Hiro extension is missing.	2-A	The system responds with an error message: <i>Hiro wallet not found</i> .
5	The patient enters an already used email.	6-A	The system responds with an error message: “ <i>Email already used</i> ”.
5	The patient enters an already used wallet address.	6-A	The system responds with an error message: “ <i>wallet ID already used</i> ”.
5	The patient enters the type of account (Doctor)	6-A	The system displays the Doctor verification page and asks for an image and degree to verify the doctor.

### 4.7.3 Doctor Sign Up

<b>Name</b>	Sign Up
<b>Actors</b>	Doctor
<b>Summary</b>	The doctor shall connect to the wallet. Doctor shall provide first name, last name, blood group, email, degree, image and the type of account

	(Patient or Doctor) on the signup form and after successful creation of the account, a new session is established and the system must redirect the doctor to the “ <i>Dashboard</i> ” page.
<b>Pre-Conditions</b>	The same wallet address must not be already in the database records.
<b>Post-Conditions</b>	The patient’s session is successfully established and shall be redirected to the “ <i>Dashboard</i> ” page.
<b>Special Requirements</b>	Hiro wallet extension

**Basic Flow**

<b>Actor Action</b>		<b>System Response</b>	
1	The doctor opens the Signup page and Hiro wallet extension is enabled.	2	The login page is displayed a popup a “ <i>Connect Wallet</i> ” button.
3	The doctor click on the “ <i>Connect Wallet</i> ” button.	4	The “ <i>Sign Up</i> ” page with the address of the associated cryptocurrency wallet is displayed asking for first name, last name, blood group, email and the type of account (Patient or Doctor).
5	The doctor enters first name, last name blood group, email and the type of account (Doctor)	6	The system checks if the wallet address and email are unique, creates an account for the patient, and redirects the doctor to the “ <i>doctor verification</i> ” page. System ask doctor for image and degree.
7	The doctor select doctor type and upload image and degree.	8	System verify degree, generate English random script and open camera for doctor verification.
9	The doctor read script in front of camera.	10	The system verify doctor with face and scrip recognition, establishes a session and redirects the doctor to the “ <i>Dashboard</i> ” page

**Alternative Flow**

1	The doctor opens the sign up page and Hiro extension is missing.	2-A	The system responds with an error message: <i>Hiro wallet not found</i> .
5	The doctor enters already used email.	6-A	The system responds with an error message: “ <i>Email already used</i> ”.
5	The doctor enters already used wallet address.	6-B	The system responds with an error message: “ <i>wallet ID already used</i> ”.
5	The doctor enters the type of account (Patient)	6-C	The system checks if the wallet address and email are unique, creates an account for the patient, establishes a session and redirects the patient to the “ <i>Dashboard</i> ” page
7	The doctor upload image only.	8-A	The system responds with an error message: “ <i>Degree is missing</i> ”.
7	The doctor upload degree only.	8-B	The system responds with an error message: “ <i>Image is missing</i> ”.
7	The doctor upload nothing.	8-C	The system responds with an error message: “ <i>Degree and image is missing</i> ”.

9	The doctor read script in front of camera.	10-A	The system verify doctor, and face does not match with uploaded image. The system responds with an error message: " <i>Face recognition Fail</i> ".
9	The doctor read script in front of camera.	10-B	The system compare the script with user voice input for verification, and script does not match. The system responds with an error message: " <i>Voice not match</i> ".

#### 4.7.4 View Profile

<b>Name</b>	View Profile		
<b>Actors</b>	Doctor, Patient		
<b>Summary</b>	User presses the "Profile" tab to view this profile.		
<b>Pre-Conditions</b>	The user must be login		
<b>Post-Conditions</b>	The "Profile" page is successfully displayed.		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
<b>Actor Action</b>		<b>System Response</b>	
1	User presses on the "Profile" tab	2	The profile is displayed.

#### 4.7.5 Edit Profile

<b>Name</b>	Edit Profile		
<b>Actors</b>	Doctor, Patient		
<b>Summary</b>	Click on the "Profile" tab. The user update the information and press the save button confirmation box is displayed and user presses. System update the profile and success message displayed		
<b>Pre-Conditions</b>	The user must be login		
<b>Post-Conditions</b>	The User profile is successfully updated.		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
<b>Actor Action</b>		<b>System Response</b>	
1	User presses on the "Profile" tab	2	The profile is displayed.
3	User updates the information and presses the "Save" button.	4	The system updates the information and displays a success message " <i>Update successfully</i> "

#### 4.7.6 View Free Slots

<b>Name</b>	View Free Slots		
<b>Actors</b>	Doctor, Patient, Lab Technician		
<b>Summary</b>	The user press on the “My Schedule” tab. The free slots schedule by user is displayed.		
<b>Pre-Conditions</b>	The user must be login.		
<b>Post-Conditions</b>	“Appointment” page is open and list of free slots scheduled by user is displayed		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
Actor Action	System Response		
1	The user presses the “Appointment” tab.	2	The “Appointment” page is opened and display the list of free slots schedule by user.

#### 4.7.7 Add Free Slot

<b>Name</b>	Add Free Slot		
<b>Actors</b>	Doctor, Lab Technician		
<b>Summary</b>	The user Click the “Add Free Slot” button. The user shall provide the time and date on the Free Slot form and after successfully adding free slot, system displays a success message.		
<b>Pre-Conditions</b>	The user must be login The user are on the “Appointment” page		
<b>Post-Conditions</b>	The Free slot is successfully added and user shall be redirected to the “Appointment” page.		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
Actor Action	System Response		
1	Click on an “Add Free Slot” button.	2	The Free slot form is displayed asking for time and date.
3	The user enter time and date and presses “Add Slot”.	4	The system validate the information, add the free slot, redirect to the “Appointment” page and displays a success message
<b>Alternative Flow</b>			
3	The user enter time and date that already exists in free slots and presses “Add Slot”.	4-A	The system responds with an error message: <i>Free slot already added.</i>
3	The user close the Free slot form.	4-A	The form is disappeared.

#### 4.7.8 Edit Free Slot

<b>Name</b>	Edit Free Slot		
<b>Actors</b>	Doctor, Lab Technician		
<b>Summary</b>	The user Click the “Edit” button place in front of Free Slots. The user shall provide the time and date on the Free Slot form and after successfully editing free slot, system displays a success message.		
<b>Pre-Conditions</b>	The user must be login The user are on the “Appointment” page		
<b>Post-Conditions</b>	The Free slot is successfully edited and user shall be redirected to the “Appointment” page.		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
<b>Actor Action</b>		<b>System Response</b>	
1	Click on an “Edit” button.	2	The Edit Free slot form is displayed asking for time and date.
3	The user enter time and date and presses “Edit Slot”.	4	The system validate the information, edit the free slot, redirect to the “Appointment” page and displays a success message
<b>Alternative Flow</b>			
3	The user enter time and date that already exists in free slots and presses “Edit Slot”.	4-A	The system responds with an error message: <i>Free slot already exists.</i>
3	The user close the Free slot form.	4-A	The form is disappeared.

#### 4.7.9 Delete Free Slots

<b>Name</b>	Edit Free Slot		
<b>Actors</b>	Doctor, Lab Technician		
<b>Summary</b>	The user Click the “Delete” button place in front of Free Slots. The confirmation box is displayed and user presses “Yes” to delete free slot and after successfully deleting free slot, system displays a success message.		
<b>Pre-Conditions</b>	The user must be login The user are on the “Appointment” page		
<b>Post-Conditions</b>	The Free slot is successfully deleted and user shall be redirected to the “My Schedule” page.		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
<b>Actor Action</b>		<b>System Response</b>	
1	Click on an “Delete” button.	2	The confirmation box is appeared.
3	The user presses “Yes”.	4	The system remove the free slot, redirect

			user to the “Appointment” page and displays a success message
<b>Alternative Flow</b>			
3	The user presses “No”.	4-A	The confirmation box is disappeared.

#### 4.7.10 View Doctors

<b>Name</b>	View Doctors		
<b>Actors</b>	Patient		
<b>Summary</b>	The user presses the “Dashboard” tab. The list of doctors with their appointment details is displayed.		
<b>Pre-Conditions</b>	The user must be login		
<b>Post-Conditions</b>	The list of doctors with their appointment details is displayed.		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
<b>Actor Action</b>		<b>System Response</b>	
1	Click on an “Dashboard” tab.	2	The list of doctors with their appointment details is displayed.

#### 4.7.11 View Doctor Profile

<b>Name</b>	View Doctors		
<b>Actors</b>	Patient		
<b>Summary</b>	The user presses the “Dashboard” tab. The list of doctors with their appointment details is displayed. The users then presses the “Profile” button against the desired doctor. The doctor profile page is opened.		
<b>Pre-Conditions</b>	The user must be login		
<b>Post-Conditions</b>	The desired doctor profile page is displayed.		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
<b>Actor Action</b>		<b>System Response</b>	
1	Click on an “Dashboard” tab.	2	The list of doctors with their appointment details is displayed.
3	The user presses the “Profile” button.	4	The profile page of doctor is appeared

#### 4.7.12 Get Appointment

<b>Name</b>	Get Appointment		
<b>Actors</b>	Patient		

<b>Summary</b>	The user presses the “ <i>Dashboard</i> ” tab to view the available doctors. The users then presses the “ <i>Appointment</i> ” button against the desired doctor. Free time slots form is open which display the available time slots of that doctor and user select one time slot among them. The system verifies that the user has required balance in the wallet and after confirmation, the appointment process is completed.		
<b>Pre-Conditions</b>	The user must be login		
<b>Post-Conditions</b>	The appointment must be added in the patient appointment list.		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
Actor Action	System Response		
1 Click on an “ <i>Dashboard</i> ” tab.	2 The list of doctors is displayed.		
3 The user presses the “ <i>Appointment</i> ” button.	4 The free slots form of doctor is appeared		
5 User click on any available slot.	6 The free slot is highlighted.		
7 User presses the “ <i>Book Appointment</i> ” button.	8 After verification that the user has required balance in the wallet, a confirmation box appears.		
9 User presses the “ <i>Yes</i> ” button.	10 Confirmation box is disappeared. The Appointment is booked after the doctor acceptance and added to the user’s appointment list.		
<b>Alternative Flow</b>			
5 The user close the free slot form.	6-A The form is disappeared.		
7 User presses the “ <i>Book Appointment</i> ” button.	8-A The system responds with an error message: “ <i>Not enough balance in the wallet.</i> ”		
9 User presses the “ <i>No</i> ” button.	9-A The confirmation box disappear.		

#### 4.7.13 View Appointments

<b>Name</b>	View Appointments
<b>Actors</b>	Doctor, Patient, Lab Technician
<b>Summary</b>	The user press on the “ <i>Appointment</i> ” tab. The Appointment schedule by user is displayed.
<b>Pre-Conditions</b>	The user must be login.
<b>Post-Conditions</b>	“My Schedule” page is open and list of free slots scheduled by user is displayed
<b>Special Requirements</b>	None
<b>Basic Flow</b>	

<b>Actor Action</b>		<b>System Response</b>	
1	The user presses the “Appointment” tab.	2	The “Appointment” page is opened and display the list of Appointment schedule by user.

#### 4.7.14 Cancel Appointments

<b>Name</b>	Cancel Appointments		
<b>Actors</b>	Patient		
<b>Summary</b>	The user press on the “Appointment” tab. The Appointment schedule by user is displayed. System change the status of appointment when user click on “Cancel” Button		
<b>Pre-Conditions</b>	The user must be login.		
<b>Post-Conditions</b>	“Appointment” page is open and status of appointment is cancel.		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
<b>Actor Action</b>		<b>System Response</b>	
1	The user presses the “Appointment” tab.	2	The “Appointment” page is opened and display the list of Appointment schedule by user.
3	The user presses the “cancel button”.	4	System update the status of appointment to cancel.

#### 4.7.15 Filter Appointments

<b>Name</b>	Filter Appointments		
<b>Actors</b>	Patient, Doctor, Lab Technician		
<b>Summary</b>	The user press on the “Appointment” tab. The Appointment schedule by user is displayed. User can search the required appointment already scheduled by user in their appointment list.		
<b>Pre-Conditions</b>	The user must be login.		
<b>Post-Conditions</b>	The appointment has been searched		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
<b>Actor Action</b>		<b>System Response</b>	
1	The user presses the “Appointment” tab.	2	The “Appointment” page is opened and display the list of Appointment schedule by user.
3	User fill the search form by entering required appointment details (date, patient email, or time slot )	4	The system read all the information and displayed it in an input box.

5	Click the “ <i>Filter</i> ” button	6	Displays list of Appointment related to search
<b>Alternative Flow</b>			
5	Click the “ <i>Filter</i> ” button	6-A	The system responds with an error message: “ <i>Does not find required appointment.</i> ”

#### 4.7.16 Add E-Prescription

<b>Name</b>	Add E-Prescription		
<b>Actors</b>	Doctor, Lab Technician		
<b>Summary</b>	The doctor shall provide patient email, purpose, and list of drugs their quantity and duration on the new E-Prescription form and after successfully adding an E-Prescription, the system displays a success message.		
<b>Pre-Conditions</b>	The doctor must be login.		
<b>Post-Conditions</b>	The new E-Prescription is added.		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
<b>Actor Action</b>		<b>System Response</b>	
1	The user presses the “ <i>Prescription</i> ” tab.	2	The “ <i>Prescription</i> ” page is opened and display the list of prescription added by doctor.
3	User fill the add prescription form by entering required prescription details (patient email, purpose and drugs details).	4	The system read all the information and displayed it in an input field.
5	Click the “ <i>Mint</i> ” button.	6	The system validates the information, adds the prescription and displays a success message.
<b>Alternative Flow</b>			
5	Click the “ <i>Mint</i> ” button without entering required prescription details.	6-A	The system responds with an error message: “ <i>Prescription details is missing.</i> ”

#### 4.7.17 Download Prescription

<b>Name</b>	Download Prescription		
<b>Actors</b>	Doctor, Patient, Lab Technician		
<b>Summary</b>	The user presses the Prescription tab. The list of prescription owned by the user is displayed. User click “ <i>Download</i> ” button to download prescription.		
<b>Pre-Conditions</b>	The user must be login.		

<b>Post-Conditions</b>	The Prescription is downloaded.		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
<b>Actor Action</b>		<b>System Response</b>	
1	The user presses the “ <i>Prescription</i> ” tab.	2	The “ <i>Prescription</i> ” page is opened and display the list of prescription added by doctor.
3	User hover over desired prescription and presses “ <i>download</i> ” button.	4	The system validates the information, and start downloading the prescription.

#### 4.7.18 QR Code of Prescription

<b>Name</b>	QR Code of Prescription		
<b>Actors</b>	Patient		
<b>Summary</b>	The user presses the Prescription tab. The list of prescription owned by the user is displayed. User click “QR Code” button. System download the QR code of prescription.		
<b>Pre-Conditions</b>	The user must be login.		
<b>Post-Conditions</b>	The Prescription QR code is downloaded.		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
<b>Actor Action</b>		<b>System Response</b>	
1	The user presses the “ <i>Prescription</i> ” tab.	2	The “ <i>Prescription</i> ” page is opened and display the list of prescription added by doctor.
3	User hover over desired prescription and presses “ <i>QR Code</i> ” button.	4	The system validates the information, and start downloading the prescription QR code.

#### 4.7.19 Preview Prescription

<b>Name</b>	Preview Prescription		
<b>Actors</b>	Patient		
<b>Summary</b>	The doctor can view the prescription before adding a new prescription for a patient.		
<b>Pre-Conditions</b>	The user must be login.		
<b>Post-Conditions</b>	The new tab is opened with the Prescription view.		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			

<b>Actor Action</b>		<b>System Response</b>	
1	The user presses the “ <i>Prescription</i> ” tab.	2	The “ <i>Prescription</i> ” page is opened and display the list of prescription added by doctor.
3	User hover over desired prescription and presses “ <i>Preview</i> ” button.	4	The system validates the information and starts displaying the prescription view.

#### 4.7.20 Filter Prescription:

<b>Name</b>	Filter Prescription		
<b>Actors</b>	Patient, Doctor, Lab Technician		
<b>Summary</b>	The user press on the “ <i>Prescription</i> ” tab. The prescriptions added by doctor is displayed. User can search the required prescription in their prescription list.		
<b>Pre-Conditions</b>	The user must be login.		
<b>Post-Conditions</b>	The prescription has been searched.		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
<b>Actor Action</b>		<b>System Response</b>	
1	The user presses the “ <i>Prescription</i> ” tab.	2	The “ <i>Prescription</i> ” page is opened and display the list of Appointment schedule by user.
3	User fill the search form by entering required <i>Prescription</i> details (date, patient email, or time slot).	4	The system read all the information and displayed it in an input box.
5	Click the “ <i>Filter</i> ” button.	6	Displays list of <i>Prescription</i> related to search.
<b>Alternative Flow</b>			
5	Click the “ <i>Filter</i> ” button	6-A	The system responds with an error message: “ <i>Does not find required prescription.</i> ”

#### 4.7.21 User Chat List

<b>Name</b>	User Chat List		
<b>Actors</b>	Patient, Doctor, Lab Technician		
<b>Summary</b>	The user shall be able to view all Chat List started with other user.		
<b>Pre-Conditions</b>	The user must be login.		
<b>Post-Conditions</b>	The list of chat threads is display.		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			

<b>Actor Action</b>		<b>System Response</b>	
1	The user presses the “Chat” tab.	2	The “Chat” tab is opened and display the list of user started with other user

#### 4.7.22 Chat Room

<b>Name</b>	Chat Room		
<b>Actors</b>	Patient, Doctor, Lab Technician		
<b>Summary</b>	User shall be able to view a previous conversation with other user in a separate chat room.		
<b>Pre-Conditions</b>	The user must be login.		
<b>Post-Conditions</b>	Chat room with user is displayed.		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
<b>Actor Action</b>		<b>System Response</b>	
1	The user presses the “Chat” tab.	2	The “Chat” tab is opened and display the list of user started with other user
3	User click on the desired user name displayed in chat list.	4	The system displays a chat box of that desired user with previous conversation

#### 4.7.23 Start New Chat

<b>Name</b>	Start Chat		
<b>Actors</b>	Patient, Doctor, Lab Technician		
<b>Summary</b>	Users will be able to initiate new chats with other users by entering their email address.		
<b>Pre-Conditions</b>	The user must be login.		
<b>Post-Conditions</b>	Chat room with user is displayed.		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
<b>Actor Action</b>		<b>System Response</b>	
1	The user presses the “Chat” tab.	2	The “Chat” tab is opened and display the list of user started with other user
3	User click on the “New” button.	4	The system displays a field and asking for user email,
5	User enter the email of user to start chat.	6	The system create a new chat room with user.
<b>Alternative Flow</b>			
5	User enter the invalid email of user to start chat.	6-A	The system responds with an error message: “User not found.”

#### 4.7.24 Send Message

<b>Name</b>	Send Message		
<b>Actors</b>	Patient, Doctor, Lab Technician		
<b>Summary</b>	In a chat room, users will be able to send messages to one another.		
<b>Pre-Conditions</b>	The user must be login.		
<b>Post-Conditions</b>	New chat room with user is created.		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
<b>Actor Action</b>		<b>System Response</b>	
1	The user presses the “Chat” tab.	2	The “Chat” tab is opened and display the list of user started with other user
3	Select a specific user from a list of users in a chat room.	4	Display the history of messages of that specific user.
5	Click on the message text box and write text	6	System display the message written in chat box.
7	Click the “send” button.	8	Displays message in chat history
<b>Alternative Flow</b>			
5	Click on “Back” button.	6-A	Closes the chat box

#### 4.7.25 Download Reports

<b>Name</b>	Download Report		
<b>Actors</b>	Patient, Lab Technician		
<b>Summary</b>	Unfortunately, if patient is suffering from disease or fever, its treatment take place as soon as possible, The patient should want to download his reports.		
<b>Pre-Conditions</b>	The patient want to see his previous and current reports.		
<b>Post-Conditions</b>	The patient see his reports successfully		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
<b>Actor Action</b>		<b>System Response</b>	
1	Click on the “Report” button.	2	Displays an all reports with date.
3	Select a specific report from a list of reports in a report section.	4	Download that specific Report

#### 4.7.26 View Reports

<b>Name</b>	View Report
-------------	-------------

<b>Actors</b>	Patient, Lab Technician		
<b>Summary</b>	Unfortunately, if patient is suffering from disease or fever, its treatment take place as soon as possible, The patient should want to view his reports.		
<b>Pre-Conditions</b>	The patient want to see his previous and current reports.		
<b>Post-Conditions</b>	The patient see his reports successfully		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
Actor Action	System Response		
1 Click on the “Report” button.	2 Displays an all reports with date.		
3 Select a specific report from a list of reports in a report section and click on preview button.	4 Display that specific Report		

#### 4.7.27 Add Reports

<b>Name</b>	Add Reports		
<b>Actors</b>	Lab Technician		
<b>Summary</b>	The doctor wants to mint the report on block chain by entering email, test name, feedback (can get from disease classification model), additional details such as property (Ca, Zn), value, and normal range and can delete these details, and the doctor will add the report. After that, the patient should follow that prescription.		
<b>Pre-Conditions</b>	The user must be login.		
<b>Post-Conditions</b>	The doctor add report successfully.		
<b>Special Requirements</b>	None		
<b>Basic Flow</b>			
Actor Action	System Response		
1 Click on the Report” button.	2 Displays an all reports with date, a form where we add report details and asking for report details.		
3 User enter email, test name, feedback (can get from Disease classification model by click on “AI Assist”), Additional details such as property(Ca ,Zn), Value, and Normal and click on ‘mint’ Button	4 Display that specific Report		
5 Range and can delete these details.	6 System display a success message and report details below with date		
<b>Alternative Flow</b>			

5	Click the “Filter” button	6-A	The system responds with an error message: “Does not find required appointment.”
---	---------------------------	-----	--

#### 4.7.28 Filter Report

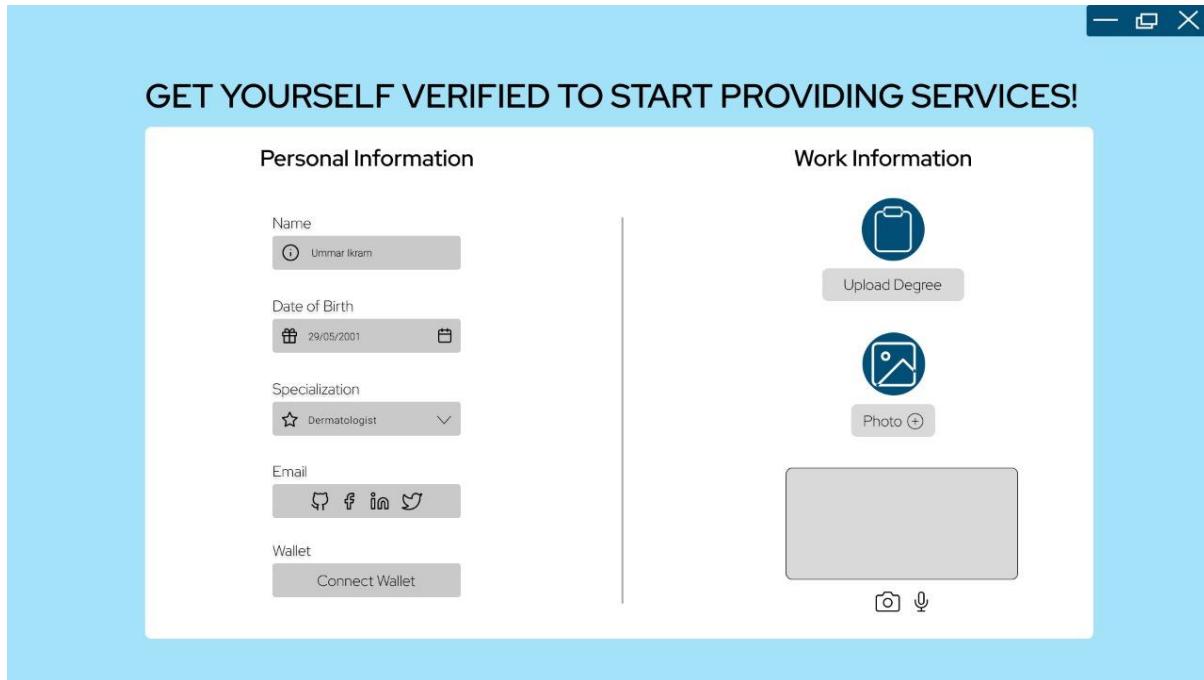
<b>Name</b>		Filter Report	
<b>Actors</b>		Patient, Lab Technician	
<b>Summary</b>		The user press on the “Report” tab. The report added by doctor is displayed. User can search the required report in their report list.	
<b>Pre-Conditions</b>		The user must be login.	
<b>Post-Conditions</b>		The report has been searched.	
<b>Special Requirements</b>		None	
<b>Basic Flow</b>			
<b>Actor Action</b>		<b>System Response</b>	
1	The user presses the “Report” tab.	2	The “Report” page is opened and display the list of Appointment schedule by user.
3	User fill the search form by entering required Report details (date, patient email, or time slot).	4	The system read all the information and displayed it in an input box.
5	Click the “Filter” button.	6	Displays list of Report related to search.
<b>Alternative Flow</b>			
5	Click the “Filter” button	6-A	The system responds with an error message: “Does not find required Report.”

#### 4.7.29 Logout

<b>Name</b>		Logout	
<b>Actors</b>		Patient, Doctor, Lab Technician	
<b>Summary</b>		This use case describes the steps for logout of a user.	
<b>Pre-Conditions</b>		The user must be login.	
<b>Post-Conditions</b>		User is logged out and login page is displayed	
<b>Special Requirements</b>		None	
<b>Basic Flow</b>			
<b>Actor Action</b>		<b>System Response</b>	
1	Click on the “logout” button.	2	System end the session of the current user and displays the login page and hides private pages.

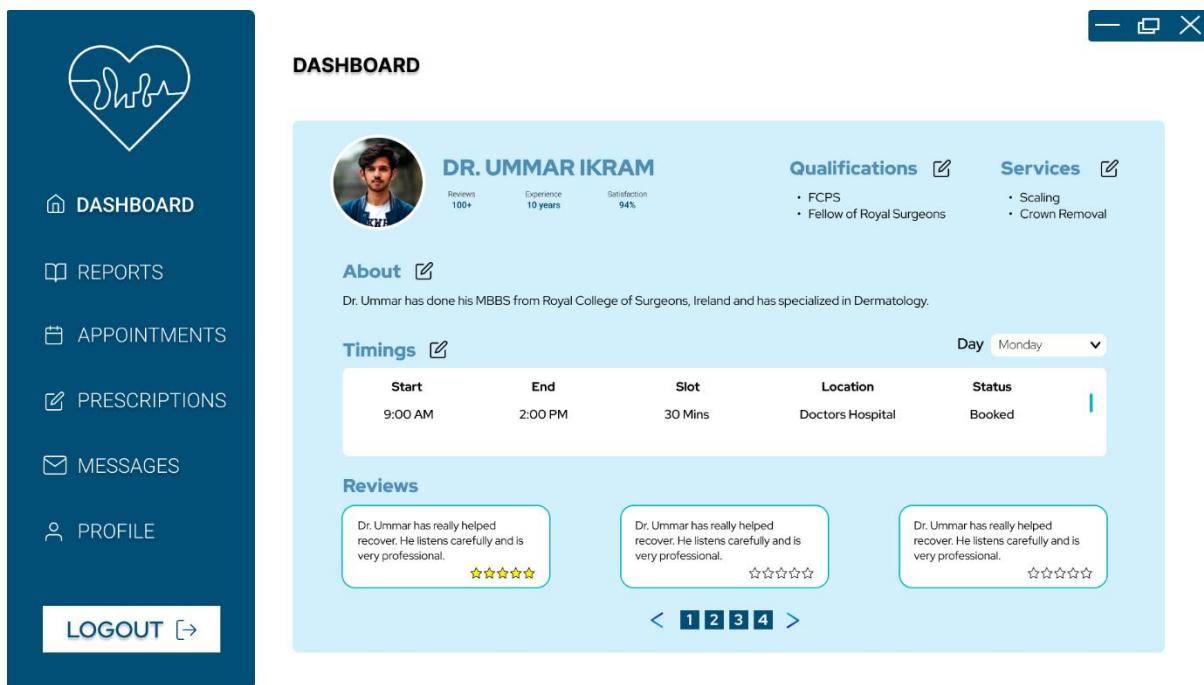
## 4.8 Graphical User Interface

The figures below represent the GUI of core functionalities of the SHIFA website.



**Figure 1: Signup Page**

The above figure represent the GUI of Signup Page.



**Figure 2: Doctor's Dashboard Page**

The above figure represent the GUI of Doctor's Dashboard Page.

**REPORTS**

**Additional Details**

Property	Value	Normal-Range	Actions
Potassium	50 mg	30-80 mg	<span style="color:red;">✖</span> <span style="color:green;">✚</span>
Calcium	80 mg	30-80 mg	<span style="color:red;">✖</span> <span style="color:green;">✚</span>

Patient Wallet/Email Address

Test Name: \_\_\_\_\_  
e.g. CBC, Covid

Feedback: \_\_\_\_\_

**Start Date :** 01/07/2022    **Test Name :** \_\_\_\_\_  
e.g. CBC, Covid

**End Date :** 01/07/2023    **Patient Wallet/Email :** \_\_\_\_\_  
e.g. ummar@gmail.com

**RESET** **AI Assist** **Download** **Preview** **Mint**

**Date**    **Test Name**    **Patient**    **Feedback**

21/07/2022	Malaria	ummar@gmail.com	Positive
------------	---------	-----------------	----------

**Download**

< 1 2 3 4 >

**Figure 3: Doctor's Report Page**  
The above figure represent the GUI of Doctor's Report Page.

**APPOINTMENTS**

**MY SCHEDULE**

Day: Monday

Start	End	Slot Size	Location	Actions
9:00 AM	2:00 PM	30 Mins	Doctors Hospital	<span style="color:red;">✖</span> <span style="color:green;">✚</span>
4:00 PM	7:00 PM	30 Mins	Lateef Hospital	<span style="color:red;">✖</span>

**Download** **SAVE**

**Start Date :** 01/07/2022    **Patient Wallet/Email :** \_\_\_\_\_  
e.g. DR. Burhan

**End Date :** 01/07/2022    **Time Slot :** \_\_\_\_\_  
e.g. 1:00 PM - 2:00 PM

**RESET** **Filter**

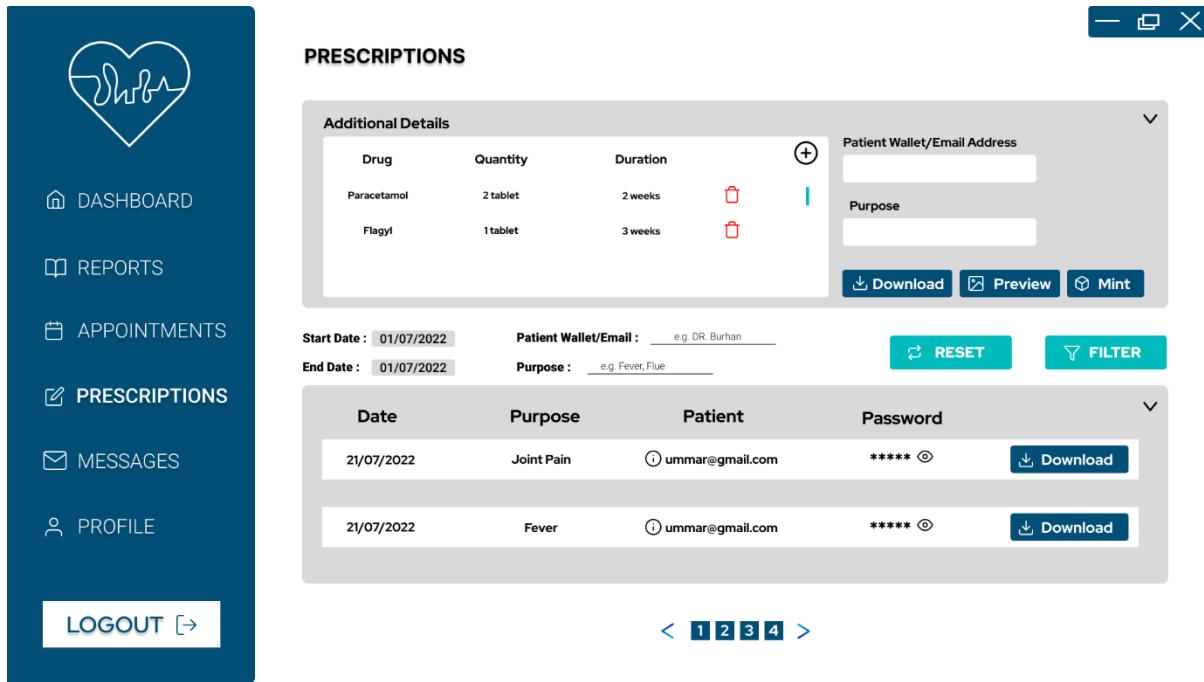
**Date**    **Time Slot**    **Patient**    **Rating**

21/12/2022	1:00 PM-2:00PM	ummar@gmail.com	★★★★★
------------	----------------	-----------------	-------

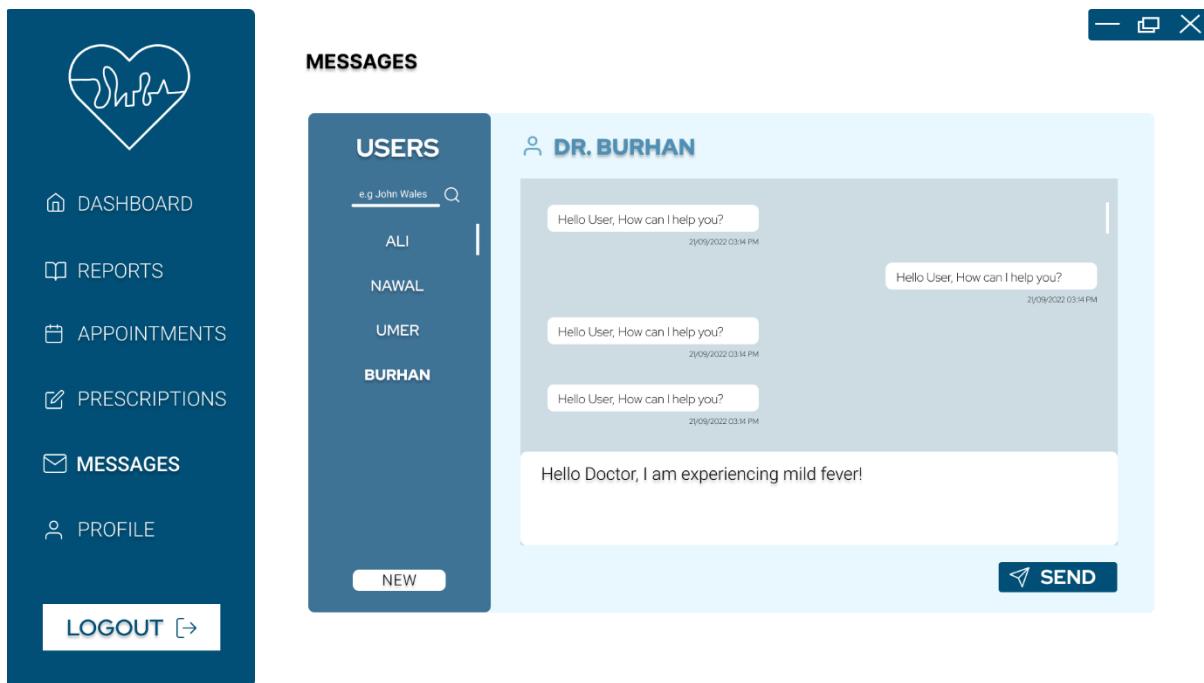
**Cancel** **Download**

< 1 2 3 4 >

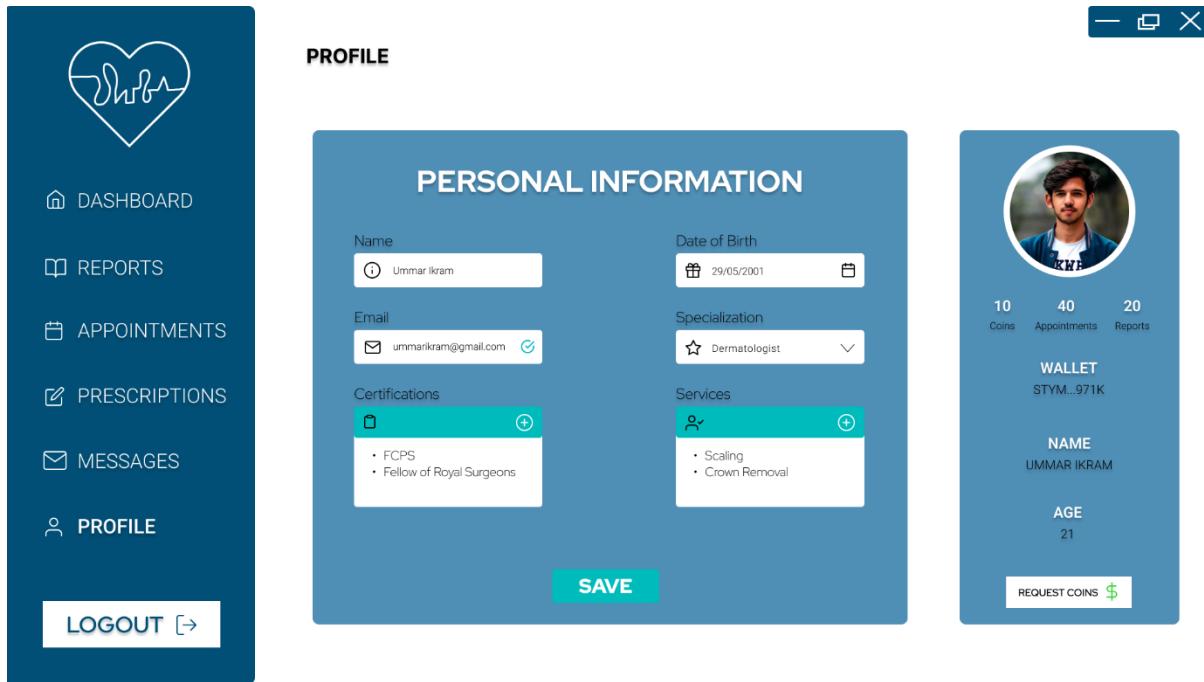
**Figure 4: Doctor's Appointment Page**  
The above figure represent the GUI of Doctor's Appointment Page.



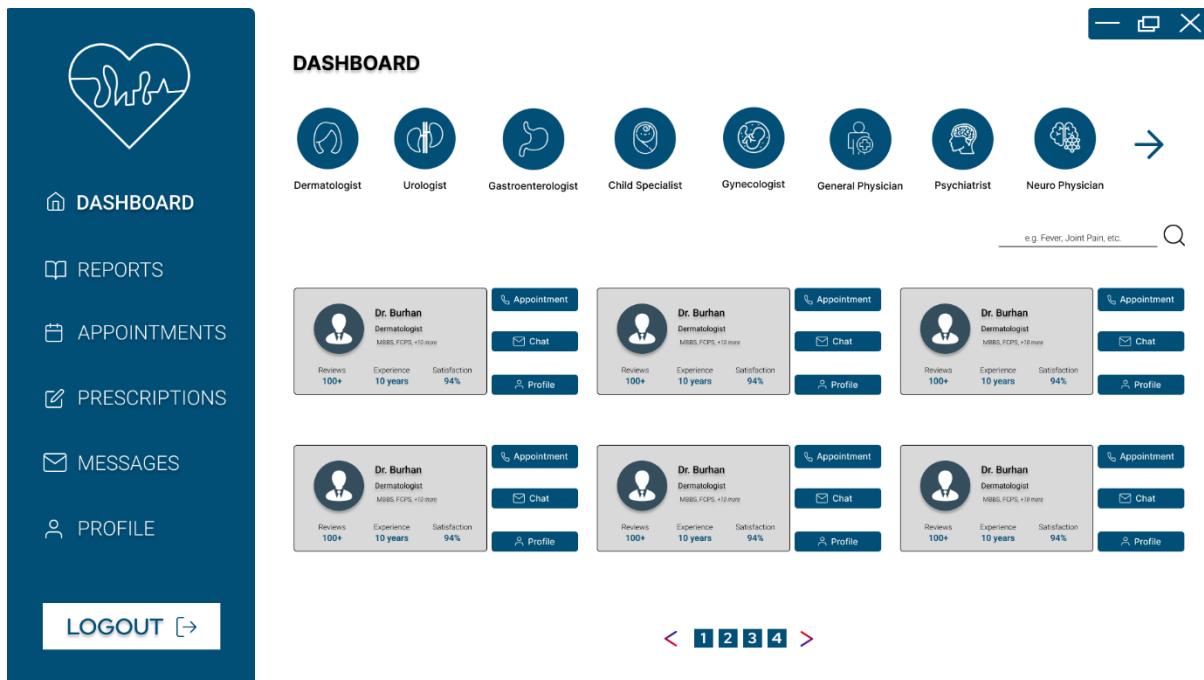
**Figure 5: Doctor's Prescription Page**  
*The above figure represent the GUI of Doctor's Prescription Page.*



**Figure 6: User's Message Page**  
*The above figure represent the GUI of User's Message Page.*



**Figure 7: Doctor's Profile Page**  
*The above figure represent the GUI of Doctor's Profile Page.*



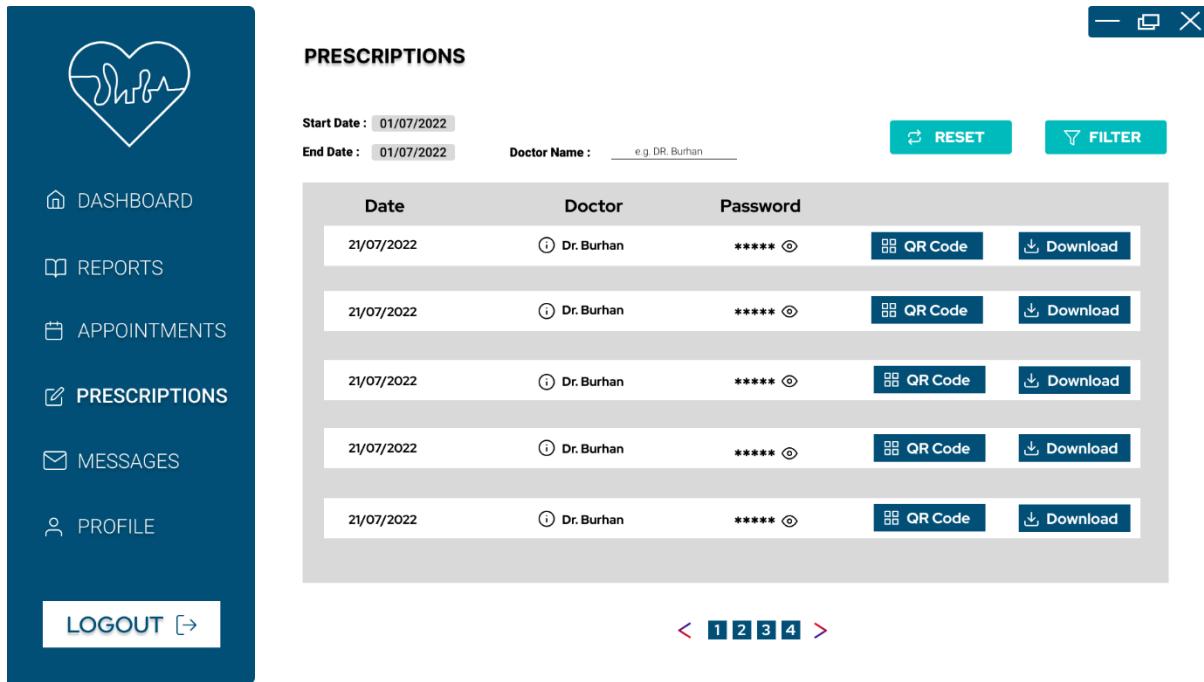
**Figure 8: Patient Dashboard Page**  
*The above figure represent the GUI of Patient Dashboard Page.*

Date	Test Name	Doctor	Feedback
21/07/2022	CBC	Dr. Burhan	Normal
21/07/2022	Vitamin-D	Dr. Burhan	High Cholesterol
21/07/2022	Zinc	Dr. Burhan	High Blood Pressure
21/07/2022	Malaria	Dr. Burhan	Normal
21/07/2022	Covid-19	Dr. Burhan	Normal

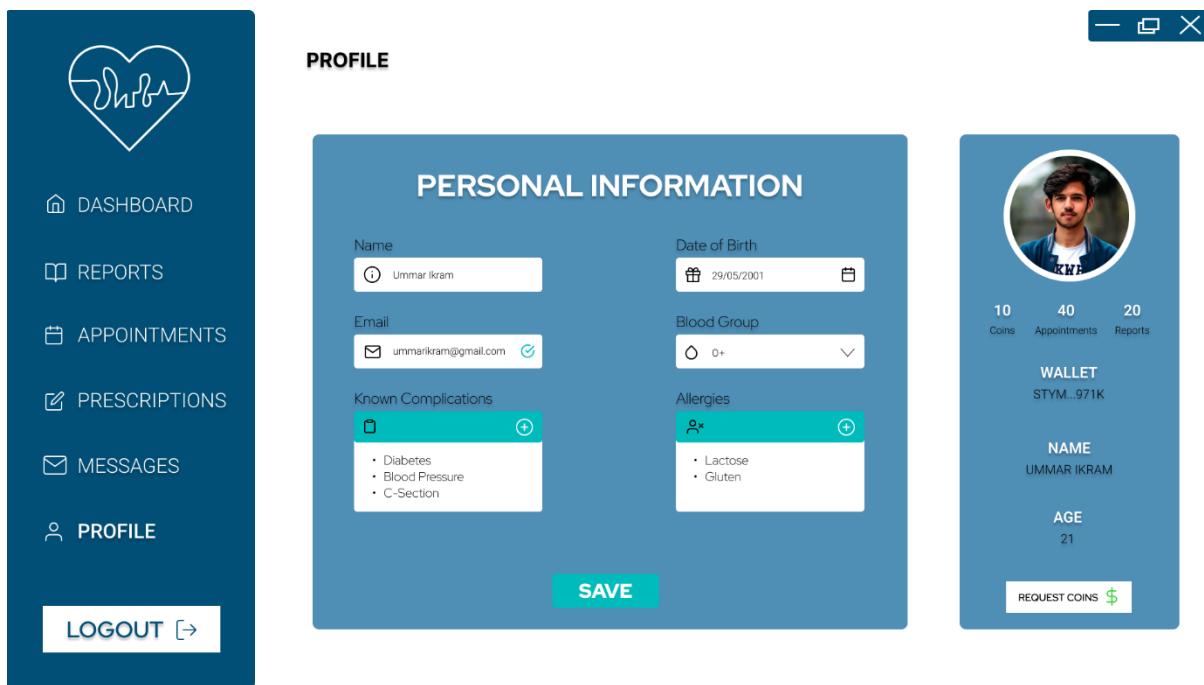
**Figure 9: Patient Reports Page**  
The above figure represent the GUI of Patient Reports Page.

Date	Time Slot	Doctor	Rating
21/07/2020	1:00 PM-2:00PM	Dr. Burhan	★★★★★
21/07/2021	1:00 PM-2:00PM	Dr. Burhan	★★★★★
21/07/2022	1:00 PM-2:00PM	Dr. Burhan	★★★★★
21/12/2022	1:00 PM-2:00PM	Dr. Burhan	★★★★★
24/12/2022	1:00 PM-2:00PM	Dr. Burhan	★★★★★

**Figure 10: Patient Appointment Page**  
The above figure represent the GUI of Patient Appointment Page.



**Figure 11: Patient Prescription Page**  
The above figure represent the GUI of Patient Prescription Page.

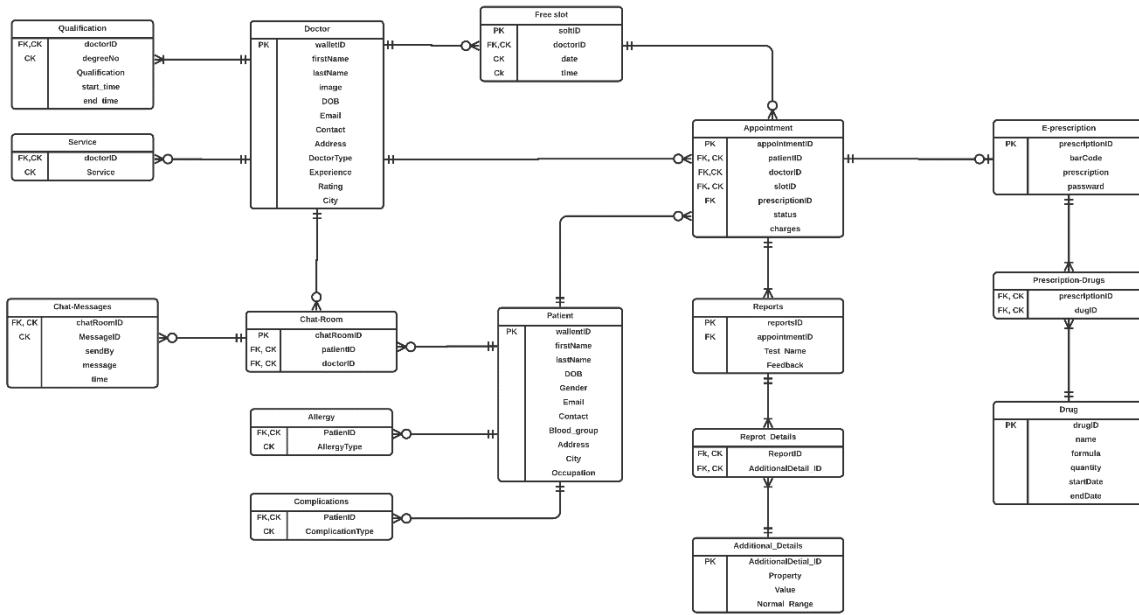


**Figure 12: Patient Profile Page**  
The above figure represent the GUI of Patient Profile Page.

## 4.9 Database Design

The following section contain the entity relation diagram with data dictionary.

### 4.9.1 ER Diagram



**Figure 13: ER Diagram**  
The figure above shows entities and relations between them.

### 4.9.2 Data Dictionary

**Table 6: Doctor**  
The data dictionary of Doctor Table along with their description is given below.

Column Name	Data Type	Constraints	Description
Wallet_ID	Varchar(20)	PK	Wallet Address
FirstName	Varchar(20)	Not null	First Name of Doctor
LastName	Varchar(20)	Not null	Last Name of Doctor
Image	Varchar(100)	Not null	Doctor photo
DOB	date	Not null	Date of birth
Email	Varchar(50)	Candidate key	Any valid email
DoctorType	Varchar(20)	Not null	Field of doctor
Experience	Int	Nullable	Year of experience
Rating	Float	Nullable	Rating of doctor
City	Varchar(20)	Nullable	City of doctor
Address	Varchar(100)	Nullable	Address of doctor
Contact	Varchar(20)	Not null	Doctor valid contact number

**Table 7: Qualification**  
*The data dictionary of Qualification Table along with their description is given below.*

Column Name	Data Type	Constraints	Description
Doctor_ID	Varchar(20)	FK, CK	Doctor Wallet ID
DegreeNo	Varchar(20)	CK	Degree No of doctor
Qualification	Varchar(20)	Not null	Qualification of doctor
Start_time	Date	Not null	Starting time of degree
End_time	Date	Nullable	Ending date of degree

**Table 8: Service**  
*The data dictionary of Service Table along with their description is given below.*

Column Name	Data Type	Constraints	Description
Doctor_ID	Varchar(20)	FK, CK	Doctor Wallet ID
Service	Varchar(20)	CK	Service provide by doctor

**Table 9: Free Slot**  
*The data dictionary of Free Slot Table along with their description is given below.*

Column Name	Data Type	Constraints	Description
Doctor_ID	Varchar(20)	FK	Doctor Wallet ID
Time	date	Not null	Free slot time

**Table 10: Patient**  
*The data dictionary of Patient Table along with their description is given below.*

Column Name	Data Type	Constraints	Description
Wallet_ID	Varchar(20)	PK	Wallet Address
FirstName	Varchar(20)	Not null	First Name of Patient
LastName	Varchar(20)	Not null	Last Name of Patient
BloodGroup	Varchar(5)	Not null	Blood group
DOB	date	Not null	Date of birth
Email	Varchar(50)	Candidate key	Any valid email
Gender	Varchar(20)	Not null	Gender of Patient
Occupation	Varchar(20)	Nullable	Occupation of Patient
City	Varchar(20)	Nullable	City of Patient

Address	Varchar(100)	Nullable	Address of Patient
Contact	Varchar(20)	Nullable	Patient valid contact number

**Table 11: Complication**

The data dictionary of Complication Table along with their description is given below.

Column Name	Data Type	Constraints	Description
Patient_ID	Varchar(20)	FK, CK	Patient Wallet Address
ComplicationType	Varchar(20)	CK	Type of complication

**Table 12: Allergy**

The data dictionary of Allergy Table along with their description is given below.

Column Name	Data Type	Constraints	Description
Patient_ID	Varchar(20)	FK, CK	Patient Wallet Address
AllergyType	Varchar(20)	CK	Allergy Name

**Table 13: Chat Room**

The data dictionary of Chat Room Table along with their description is given below.

Column Name	Data Type	Constraints	Description
ChatRoom_ID	Int	PK	Chat Room ID
Patient_ID	Varchar(20)	FK	Patient Wallet Address
Doctor_ID	Varchar(20)	FK	Doctor Wallet ID

**Table 14: Chat Messages**

The data dictionary of Chat Messages Table along with their description is given below.

Column Name	Data Type	Constraints	Description
ChatRoom_ID	Int	FK, CK	Chat Room ID
Message_ID	Int	CK	Message Unique ID
SendBy	Varchar(20)	Not null	Sender
Message	Varchar(1000)	Not null	Message
Time	Date	Not null	Send time of message

**Table 15: Appointment**

The data dictionary of Appointment Table along with their description is given below.

Column Name	Data Type	Constraints	Description

Appointment_ID	Int	PK	Appointment Unique ID
Doctor_ID	Varchar(20)	FK	Doctor Wallet Address
Patient_ID	Varchar(20)	FK	Patient Wallet ID
Slot-ID	Int	FK	Slot unique ID
Prescription_ID	Int	FK	Prescription Unique ID
Status	Varchar(20)	Nullable	Status of appointment
Charges	Varchar(20)	Not null	Charges of appointment

**Table 16: Reports**  
*The data dictionary of Reports Table along with their description is given below.*

Column Name	Data Type	Constraints	Description
Report_ID	Int	PK	Report ID
Appointment_ID	Int	FK	Appointment unique ID
Test_Name	Varchar(20)	Not null	Name of report test
Feedback	Varchar(20)	Not null	Feedback of report

**Table 17: Additional Details**  
*The data dictionary of Additional Details Table along with their description is given below.*

Column Name	Data Type	Constraints	Description
Additional_ID	Int	PK	Report additional detail ID
Property	Varchar(20)	Not null	Name of property
Value	Varchar(20)	Not null	Amount of property
Normal_Range	Varchar(20)	Not null	Normal range of property

**Table 18: Report Details**  
*The data dictionary of Report Details Table along with their description is given below.*

Column Name	Data Type	Constraints	Description
Report_ID	Int	FK, CK	Report ID
Additional_ID	Int	FK, CK	Additional detail ID

**Table 19: Prescription**  
*The data dictionary of Prescription Table along with their description is given below.*

Column Name	Data Type	Constraints	Description

Prescription_ID	Int	Primary Key	Prescription ID
Barcode	Int	Not Null	Barcode of prescription
Password	Varchar(20)	Not Null	OTP of Barcode

**Table 20: Drugs**  
*The data dictionary of Drugs Table along with their description is given below.*

Column Name	Data Type	Constraints	Description
Drug_ID	Int	Primary Key	Drug ID
Name	Varchar(20)	Not Null	Name of drug
Formula	Varchar(20)	Not Null	Formula of drug
Quantity	Int	Not Null	Drug Quantity
Duration	Varchar(20)	Not Null	Time duration

**Table 21: Prescription Drugs**  
*The data dictionary of Prescription Drugs Table along with their description is given below.*

Column Name	Data Type	Constraints	Description
Prescription_ID	Int	FK, CK	Prescription unique ID
Drug_ID	Int	FK, CK	Drug Unique ID
Prescription_ID	Int	FK, CK	Prescription unique ID

## 4.10 Risk Analysis

Following are the risks that we may encounter during our project development life cycle.

1. Business Risk
2. Technical Risk
3. Technology Risks

### 4.10.1 Business Risk

Anything that threatens our system's ability to achieve its goal and will lower its profits or lead it to fail. The core functionality of our system, which has a great impact on our system, should be working fine. Our system will be at high risk, if any of these functions fails, resulting in system failure. Our purpose to implement all the major functionality without any bugs and failures, which caused our business to flop.

### 4.10.2 Technical Risk

1. **Integration:** Although it is our goal to make our system generic, it is possible that our module will have trouble merging with some systems.
2. **Maintenance:** Our first priority is to make our code work properly, which can create complexity in code and become difficult to maintain.

3. **Modular:** Due to time constraints, there might be a possibility that there will be more coupling between modules.

### 4.10.3 Technology Risks

1. A sudden fault could cause the system to crash.
2. Having trouble interacting with third parties.
3. The server's response time to a request is unreasonable.

## 4.11 Conclusion

This chapter provide the detail description of SRS, covering the list of feature e.g. authentication, secure communication, NFT reports, digital currency, E-Prescription and disease classifications. Moreover, it show cases the functional and non-functional requirement with quality attributes e.g. usability, reliability and supportability. Furthermore, give detail information about the recommended and minimum hardware requirement. Lastly, it cover all the potential use cases, illustrate the GUI and database design with data dictionary.

## Chapter 5: High-Level and Low-Level Design

The following chapter will provide a detailed high and low-level design of the system to be developed. These designs are a significant part of the project as they lay down the foundation for the project and make things transparent for the developers.

### 5.1 System Overview

SHIFA is a decentralized web-application built using Model-View Controller (MVC) design pattern. It will make use of a hybrid data storage model and strong encryption standards at each layer.

### 5.2 Design Considerations

Design considerations are issues of a system's specifications, design, or operating strategy, which must be resolved before making an attempt to develop a comprehensive design solution. Following are the different design consideration of our project given below.

#### 5.2.1 Assumptions and Dependencies

It is difficult to have complete knowledge about a project, to predict the future and determine what will change. Following are the key assumptions along with the project dependencies.

##### 5.2.1.1 Stacks Blockchain:

The project will be constructed on the Stacks Blockchain utilizing Clarity, their smart contract language. This results in a direct dependent on the Stacks infrastructure e.g. APIs, Libraries, and storage. However, if the need arises, we can easily switch to another Blockchain or create our own.

##### 5.2.1.2 Blockchain and Cryptocurrency awareness

We expect that at the organizational level, employees are familiar with blockchain technology and its benefits. This is necessary so that they understand the need that our project fills and how it does so. Government regulations (HIPAA) regarding healthcare is very rigorous. Moreover, the acceptance of cryptocurrency is still in progress for some countries. People will be unwilling to invest in shifting unless they are well informed about the flaws in the present system and how our approach addresses them.

##### 5.2.1.3 Hiro Wallet Extension

It works as a crypto wallet which authorizes transactions for the user and maintains their ledger records. Since it is a web-based application, a constant internet connection is required all the time to communicate with the wallet. User's web browser must have an extension "Hiro Wallet" to connect his/her crypto-wallet with our application.

##### 5.2.1.4 End-user characteristics

Users should be computer literate. Users should know basic usage of internet and web browsers. People will come back because they like the platform.

##### 5.2.1.5 MongoDB

We will be utilizing MongoDB for storing/manipulating data that requires frequent updates or low latency e.g. user profile data, messages. We rely on their clusters/servers for data availability.

### 5.2.1.6 Vercel

Our content-delivery network (CDN) would be Vercel that will be responsible for providing the application with server-side content and features e.g. static HTMLs, incremental site generation, optimized images and more.

## 5.2.2 General Constraints

The overall restrictions or limits that significantly affect the software design of the system are described in this section. The general restrictions on our project are shown below.

### 5.2.2.1 Hardware or software environment

Since it is a web-based application, browsers must be installed on the computers. Up-to-date browsers that enable JavaScript, such as Google Chrome, are recommended. The Hiro wallet extension must be installed in the browser. User's device must have a minimum hardware and software requirement as describes in hardware and software section.

### 5.2.2.2 End-user environment

The end-user will access the Application via the internet in a computer browser.

### 5.2.2.3 Availability or Volatility of resources

The Stacks blockchain makes use of bitcoin blockchain underneath, so it is subject to volatile changes with respect to cryptocurrency market around the globe. The number of miners and transaction fee is subject to change depending on external conditions. This can result in slow transactions when there are limited miners on the network.

### 5.2.2.4 Standard Compliance

The application will be following non-fungible and fungible token standards (SIP009, SIP010) to ensure scalability and interoperability among tokens on the blockchain network. Furthermore, we will make use of advanced-encryption standard (AES) to ensure compliance with HIPAA and other health regulatory bodies.

### 5.2.2.5 Blockchain resources

Blockchain programming has just recently gained popularity, learning resources are limited. Platforms also lack courses in this field. As a result, gathering necessary information regarding the system's progress is a time-consuming task.

### 5.2.2.6 Interface/protocol requirements

The HTTPS protocol will be utilized for communication between internet users. Appointment payments in the form of SHIFA Coin are made in cryptocurrency through Hiro Wallet.

### 5.2.2.7 Data repository and distribution requirements

We are using a hybrid model of data storage. All the conversations between the doctor and patient will be end-to-end encrypted using asymmetric encryption, stored inside MongoDB off-chain storage for fast retrieval. The report information and other sensitive data will be stored on the blockchain as on-chain storage. We will be using clusters (Asia-region) to communicate with MongoDB servers and Clarity Smart contracts to connect with the Blockchain.

### **5.2.2.8 Security requirements (or other such regulations)**

Authentication and authorization will be handled by the system. Because we are employing Blockchain, non-editable records will be kept for all transactions and NFTs exchanges that occur. Furthermore, we will be using Advanced Encryption Standard (AES-256) and SHA-256 to encrypt and hash data respectively.

### **5.2.2.9 Verification and validation requirements (testing)**

Users with valid login credentials will be able to access the application, according to the system. The crypto-wallet will handle the initial verification/validation process using mnemonics/secret phrases to generate public-private key pairs for the user. The system will then identify the user type (doctor, patient, new) and act accordingly. Users need to login to the website in order to use functionalities.

### **5.2.2.10 Language Constraints**

The web application will be available in English language only.

## **5.2.3 Goals and Guidelines**

The technology architecture will have the capability to automatically scale according to the incoming traffic so for this we must follow design principles

### **5.2.3.1 Modularity and Maintainability**

The technology architecture will allow for building new features in a modular way without the need to drastically refactor old services. We should ensure there is low coupling and a high cohesion between different modules. Less coupling between module and component increase the modularity that help in maintenance.

### **5.2.3.2 Quality Standards**

- Follow DRY - Don't repeat yourself.
- Make use of re-usable components in the UI
- KISS principle, software is best when it is kept simple.
- YAGNI, your software should contain what you are going to need.
- Code should be limited according to the style guide.

### **5.2.3.3 Monitoring**

Basic monitoring of the infrastructure will be provided. This includes areas such as hardware utilization and logs. Detailed service level monitoring such as CPU spikes, memory leakages are monitored and alerts are configured. We will use Linters to ensure there is no memory leakage and code is bug free.

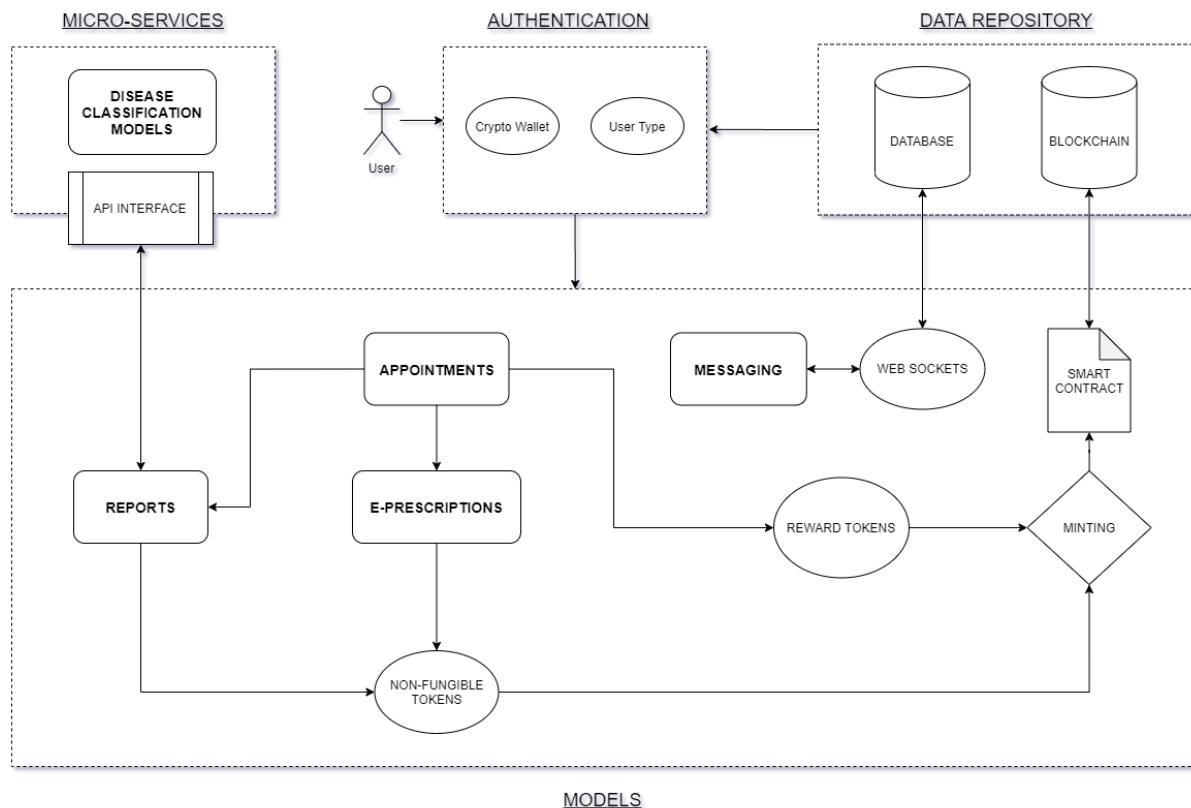
### **5.2.3.4 Compatibility**

Our web users will be able to run our web system on all famous browsers. Web users should have a Web version 4.4 or above.

### 5.2.4 Development Methods

We should divide our application into different modules in order to develop different modules in parallel as a result the wastage of time and unnecessary usage of resources will be minimized. After a module is developed and tested locally, it will be deployed on production. We are using a hybrid of agile and prototyping development methods. The server will first write using the agile method as agile is a fast software development lifecycle (SDLC) and then we would focus on the other modules using the prototyping development method. After the prototyping is complete, we will be adopting the incremental model. We will be using evolutionary prototyping. Development using modules is also a key feature for testing an application part by part to assure quality, which ensures a good quality, and bug free application in a short period. During the development, the developer will write the unit tests as well. Unit Tests created by the developer are created automatically in Visual Studio using IntelliSense Test. After a module is completed, it will be tested using black box testing. After all modules are completed then we will be integrating them using top down integration testing.

## 5.3 System Architecture



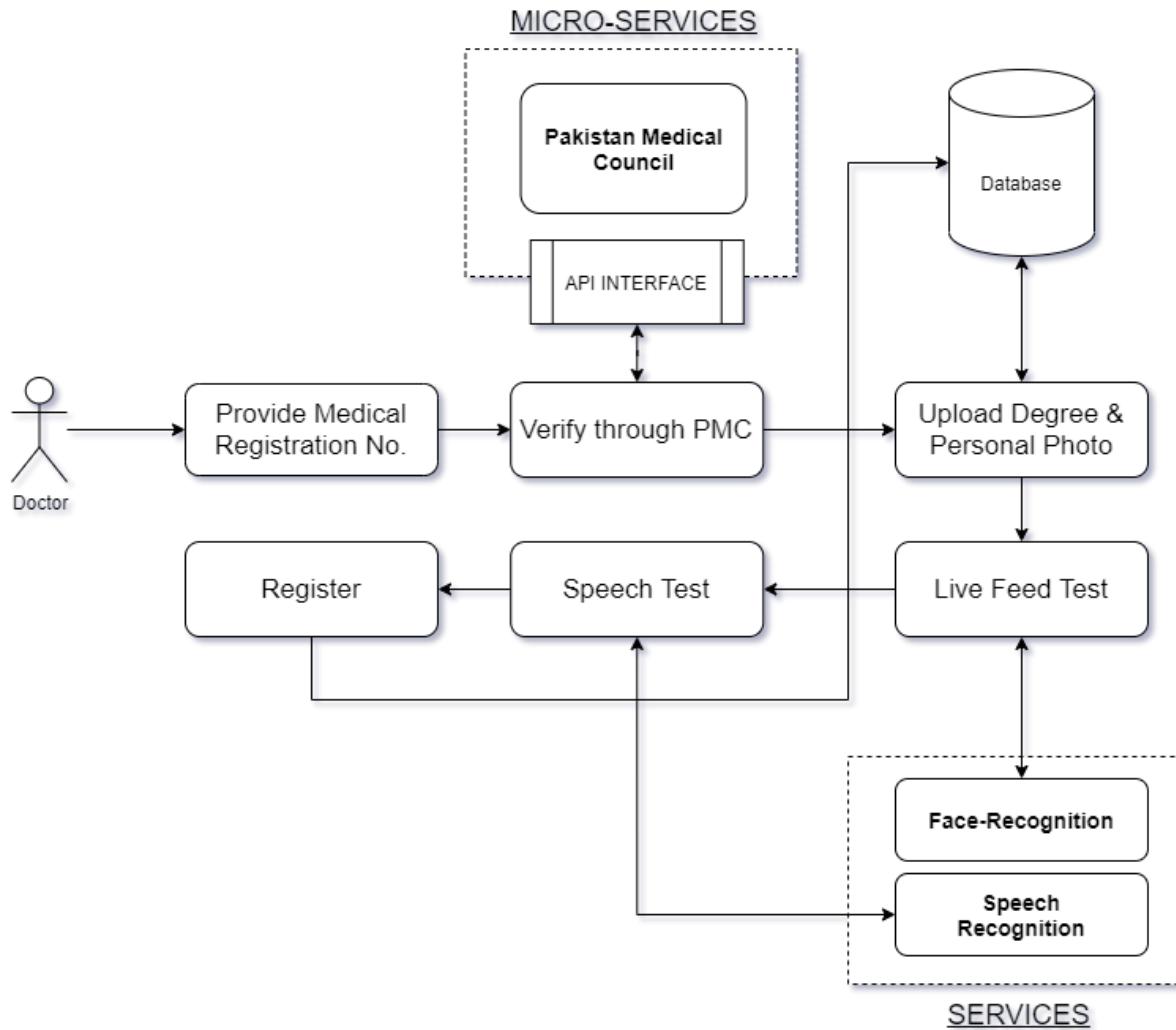
**Figure 14: System Architecture**  
*Figure shows system architecture of application and communication between modules*

The system consists of five high-level components/modules that complete the architecture.

- **Disease Classification Models:** It will be a micro-service hosted independently on a separate server. The communication will be performed via REST APIs.
- **Messaging:** It will make use of web sockets to achieve real-time communication between two users.

- **E-Prescription:** They will be encrypted using asymmetric encryption and minted as a non-fungible token on the blockchain network
- **Reports:** They will work similar to prescriptions but will include an extra integration with disease classification micro-service through an API.
- **Appointments:** Each appointment will generate reward tokens for the patient and it will be the basis for reports/prescription minting and payments for the doctor.

### 5.3.1 Doctor Authentication



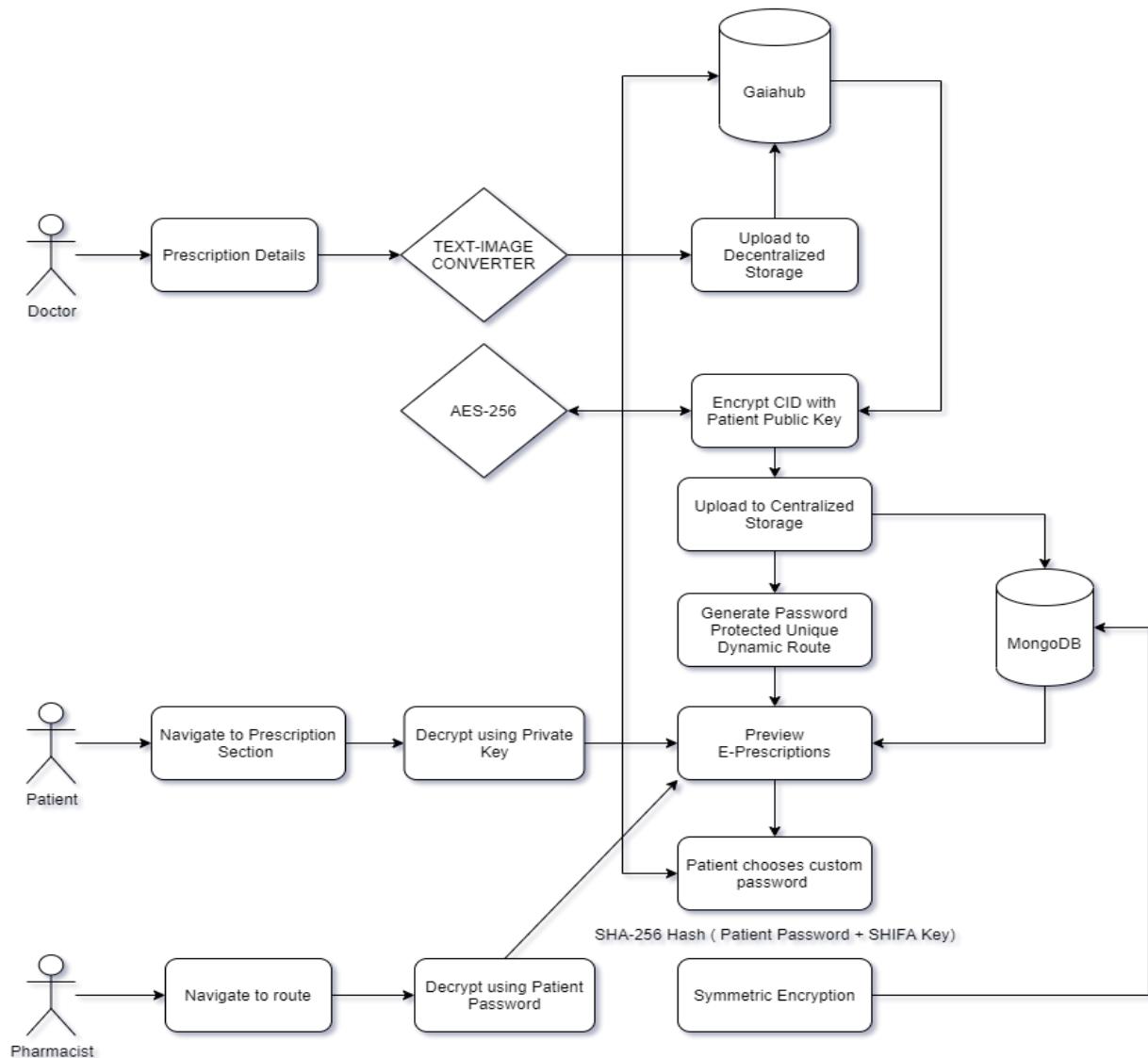
**Figure 15: Doctor Authentication Subsystem Architecture**  
*Figure shows subsystem architecture of authentication component*

The application needs to ensure that only authorized doctors are able to register to fulfil compliance and regulations. To achieve that, it follows a rigorous registration process for doctors where they have to provide their medical registration allotted by the medical council e.g. PMC. The system then fetches doctor data from the PMC database and compares it with provided information.

Additionally, the doctor is required to upload his/her head photo along with relevant educational documents. The head photo is compared with live web-cam feed and then a random script of words is generated which the doctor has to speak using the device microphone. After

successfully passing all the checks, the doctor is allowed to register on the platform and he/she can build their profile further to be featured.

### 5.3.2 E-Prescription/Report Minting



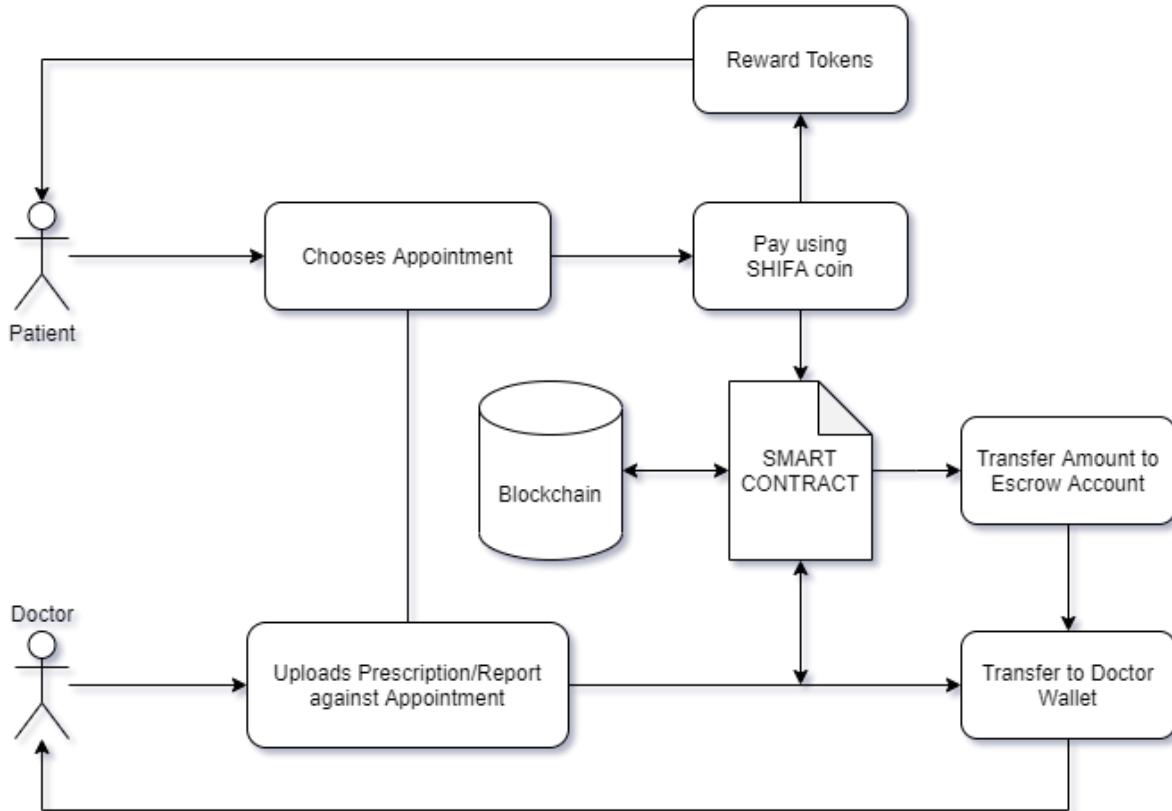
**Figure 16: E-Prescriptions Subsystem Architecture**  
*Figure show subsystem architecture of E-Prescriptions component*

One of the primary purposes of SHIFA is to ensure confidentiality of reports/prescriptions. The design is built in such a way that the information is end-to-end encrypted. The application will make use of asymmetric encryption along with other security controls like hash functions to achieve the goal.

Doctors input the necessary report information via user interface. The textual information is converted at run-time to an image and stored on the Doctor's decentralized storage. The content-identifier/URL of the report image is then encrypted using the patient's public key and stored on the centralized storage. The patient can then decrypt the URL using their private key. Additionally, a password-protected dynamic route is generated for that particular report for any user to view without logging-in.

Furthermore, the patient can choose a custom password that will be used to access that report from the dynamic route e.g. by another doctor/pharmacist.

### 5.3.3 Digital Currency



**Figure 17: Digital Currency Subsystem Architecture**  
*Figure show subsystem architecture of Digital Currency component*

Since integrating payment gateways in Pakistan is not an easy task due to various restrictions, we are implementing our own payment system using smart contracts that makes use of our custom digital currency called SHIFA coin. It will be used to charge patients for setting up appointments and as a result they can get reward coins as a cashback incentive.

Once the appointment is placed, the fees gets transferred to an escrow account and remains there until the doctor uploads a prescription/report against that appointment. Once that happens, the amount gets transferred from the escrow account to the doctor's wallet. This process is automatic and trustless.

## 5.4 Architectural Strategies

Following are the strategies we will use for designing of the Architecture.

### 5.4.1 Technology Stack

We are incorporating following technology stack for our web application.

#### 5.4.1.1 NextJS

We will be using NextJS for our application. It is a robust react production framework, which supports server-side rendering (SSR), and static site generation (SSG) that tremendously

improves web application performance. Moreover, it provides built-in CSS support and file-based API Routes, which optionally create dynamic endpoints to provide backend functionality.

#### **5.4.1.2 MongoDB**

Our project is centered around the semi-centralized storage solution. For the off-chain storage, we utilize MongoDB which is a popular NoSQL database. All the conversations and appointment details between the doctor and patient will be end-to-end encrypted using asymmetric encryption, stored inside MongoDB off-chain storage for fast retrieval.

#### **5.4.1.3 Python**

For disease prediction, we are integrating artificial intelligence. Python will be used to train disease classification models, and the saved models will be deployed. They are then accessible from the main application.

#### **5.4.1.4 Stacks**

Blockstack.js is a JavaScript library that used for smart contracts running on the Stacks blockchain. It also includes built-in authentication via Hiro Wallet, a wallet that manages Stacks STX tokens, which are used for blockchain transactions.

### **5.4.2 Reuse of existing software components**

The following existing software components, we will utilize in our project.

#### **5.4.2.1 React Components**

Since we are using NextJS a robust React framework, it provides different built in components, which we will utilize in our project instead of making similar new components that save development time e.g. cards, headers, buttons.

#### **5.4.2.2 Encryption Modules**

As security will be our major concern in application. All the data stored in the off-chain storage must be encrypted with different encryption model. We are going to utilize Advanced Encryption Standard (AES) algorithm to encrypt or decrypt user critical information.

#### **5.4.2.3 Storage Modules**

As discuss above, we are using hybrid storage, MongoDB (off chain) and Gaia (on chain) storage. MongoDB provides numerous amounts of built-in function for CRUD operation, which we are going to utilize instead of writing it again. Moreover, Gaia storage provides a generic function for adding and retrieving files that work with all file type extensions. Hence, we did not require any type of file conversion mechanism.

### **5.4.3 Future plans for extending or enhancing the software**

Following are our plans for extending or enhancing the software.

#### **5.4.3.1 Multi-Chain**

Our web application will be built on the Stacks infrastructure e.g. APIs, Libraries, and storage. However, in the future, if the need arises, we can easily expand or switch to another Blockchain or create our own.

### **5.4.3.2 Mobile App**

Our major concern is to develop web application so that user access it at any platform using browser. However, for the easiness of our user, in future our goal is to develop Mobile Application where user can access SHIFA Application through their mobile phones.

## **5.4.4 User interface paradigms**

Our interface will be built following Schneider man's Eight Golden Rules, which form the foundation of almost any UI design.

### **5.4.4.1 Consistency**

Consistent UI is characterized as having comparable design patterns, the same vocabulary in prompts, the same menus and screens, and the same instructions across the interface. Use of numerous colors, unrelated typefaces, and styles should be avoided.

### **5.4.4.2 Informative Feedback**

It is best to display feedback promptly after your users have performed or are doing actions on your website. It will notify the user of their activity.

### **5.4.4.3 Error Handling**

Make the user interface as simple to use as feasible by avoiding severe user errors. If a user makes a mistake, provide them with a proper definition of the situation and a simple solution to fix it.

### **5.4.4.4 Reverse actions easily**

When users make a mistake, we should provide them with simple and obvious ways to reverse their actions.

### **5.4.4.5 Design dialogue to yield closure.**

Informative discourse provides consumers with a sense of success. Inform them of the consequences of their actions.

## **5.4.5 Error detection and recovery**

To detect errors, we will make use of hashes that ensure data integrity. Without discovering the original message, Hashes will be used to check whether original data has been changed or not. Moreover, smart-contracts will be developed using a test-driven approach, where we will write test cases first in the Clarinet CLI and then develop contracts accordingly to minimize errors. Furthermore, libraries like Jest will be used for API testing.

Additionally, it is necessary to make backups of our centralized storage MongoDB. We will not use any third party services for that instead devise our own structure. We will make use of a scheduling package called "node-cron" to trigger NodeJS child process to run scripts for us. The backups will be saved on Google drive until a dedicated storage is purchased.

## **5.4.6 External databases and/or data storage management and persistence**

Health records will be stored on the Stacks Blockchain, whereas other personal data e.g. email, passwords will be stored on Gaia decentralized storage. All other frequently changeable data will be stored on MongoDB.

#### **5.4.7 Distributed data or control over a network**

We are utilizing both distributed and centralize data Storage infrastructure. For messaging and appointment detail we use MongoDB as a centralize storage in which central authority have a control over it and for critical records we utilize decentralized storage allows data to be dispersed over numerous physical servers(peer to peer model) in which the user have a complete control over their data.

#### **5.4.8 Concurrency and synchronization**

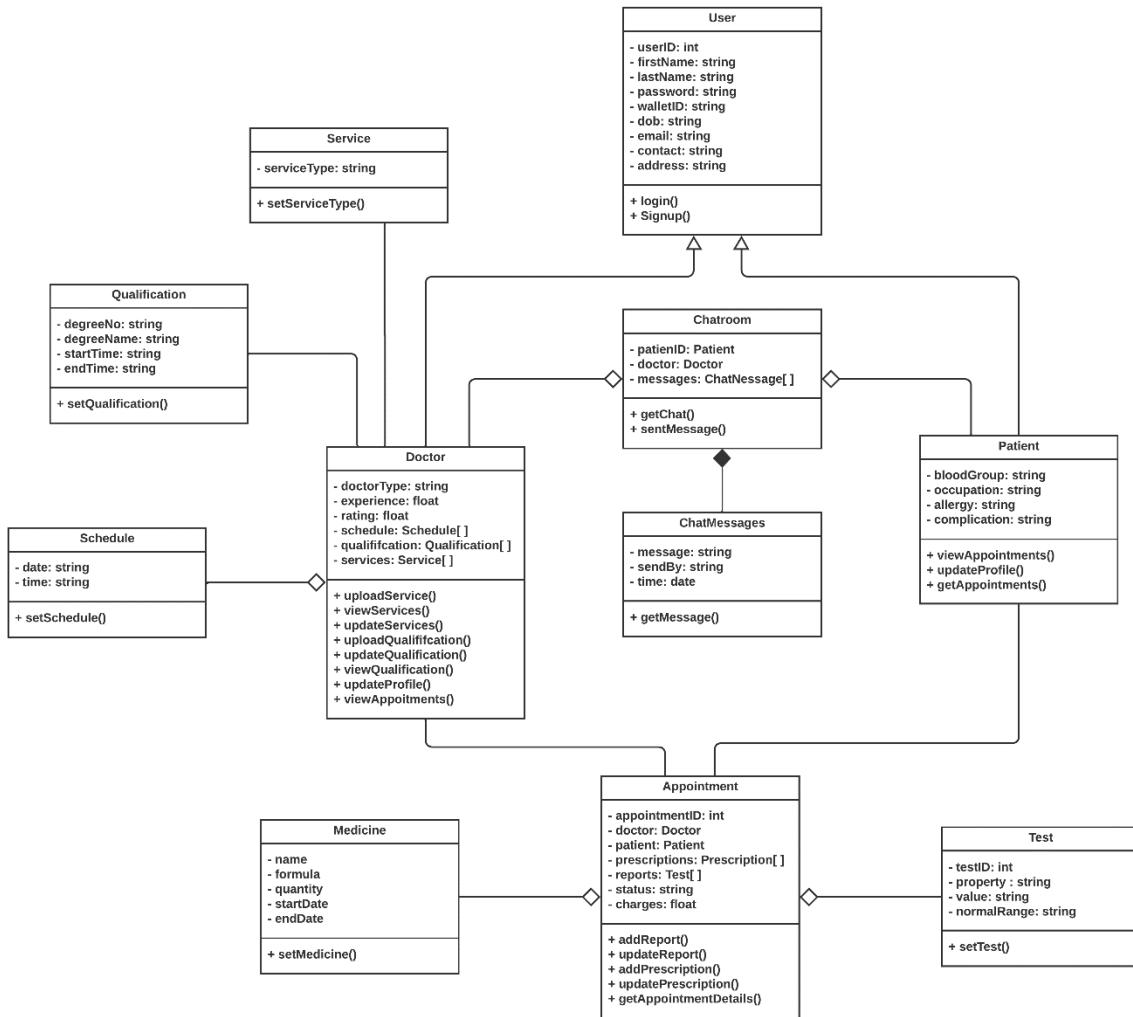
JavaScript is inherently single-threaded but we will make use of asynchronous programming and promises to effectively utilize callbacks and pools.

#### **5.4.9 Communication mechanisms**

For the communication between modules in our project, we are using two communication mechanisms are as given below.

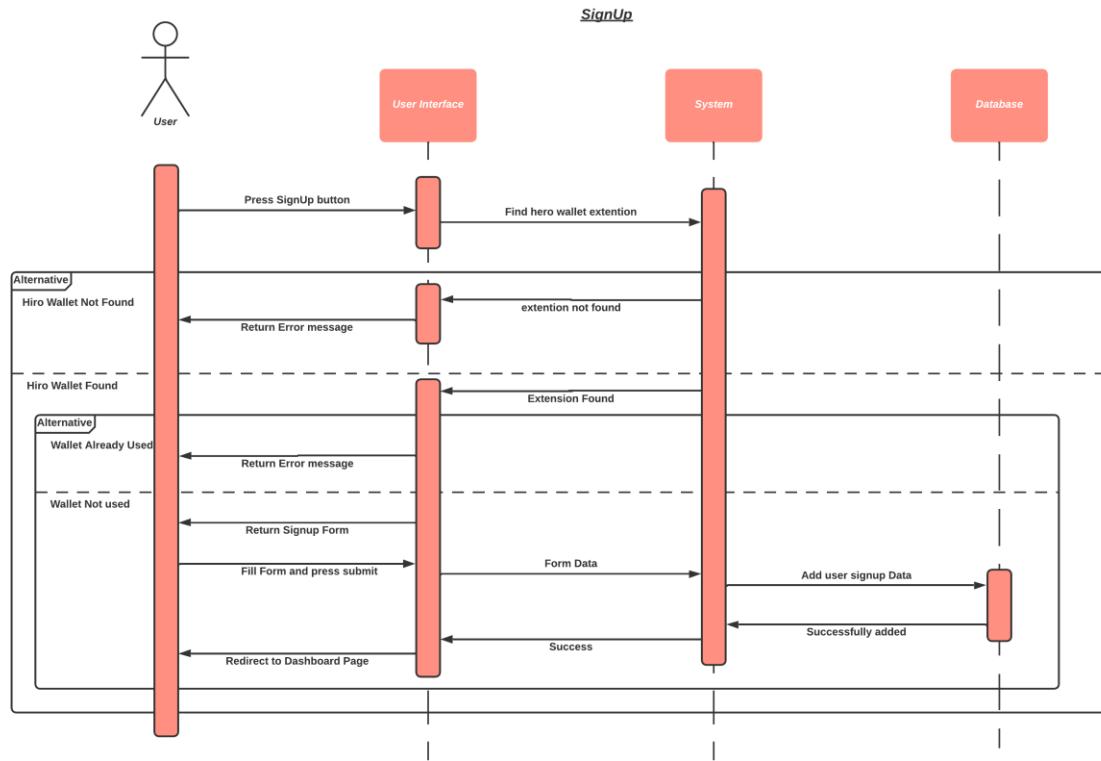
- REST APIs
- HTTPs

## 5.5 Class Diagram

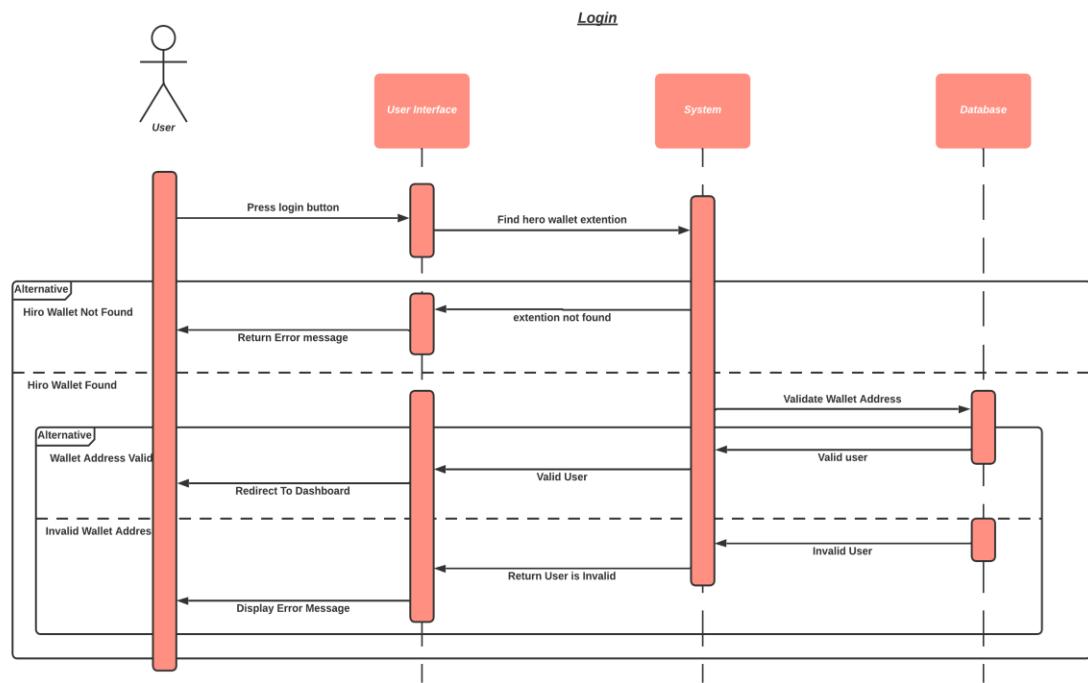


**Figure 18: Class Diagram**  
*Figure show object classes and their relationship*

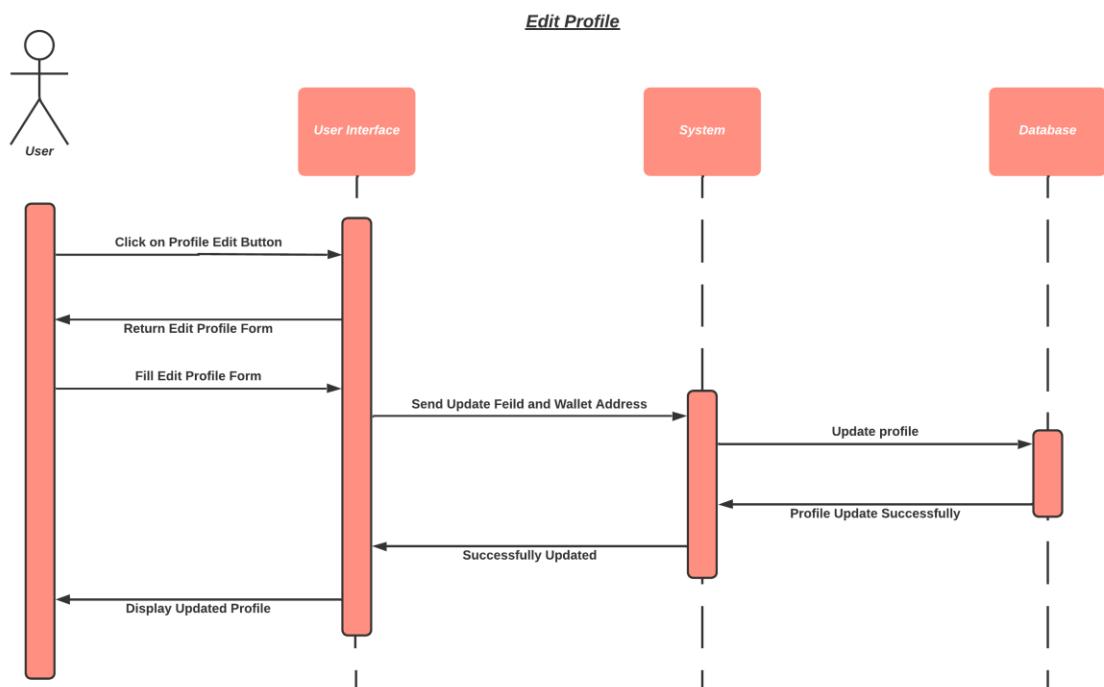
## 5.6 Sequence Diagrams



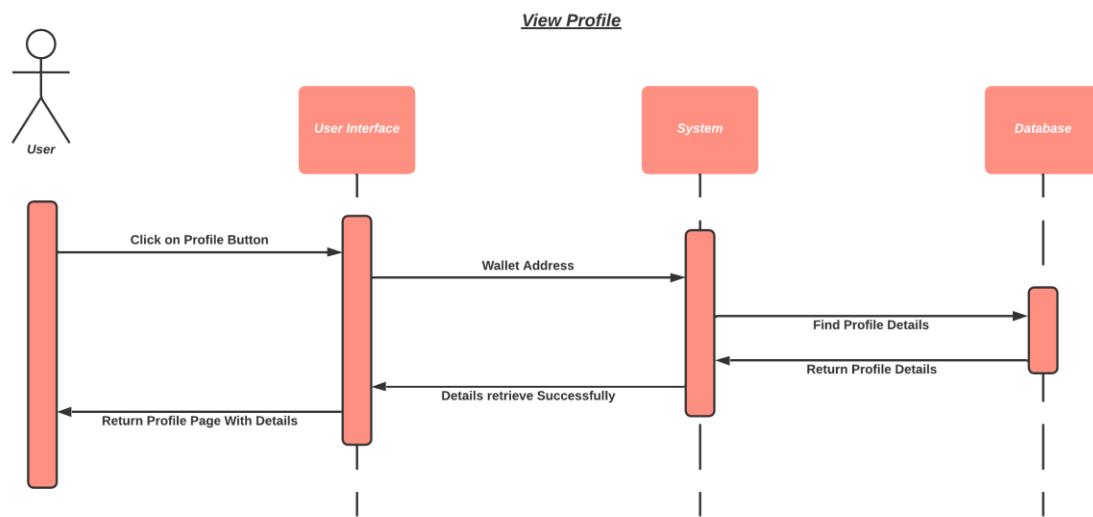
**Figure 19: Signup Sequence**  
Shows sequence of operations in Signup Process



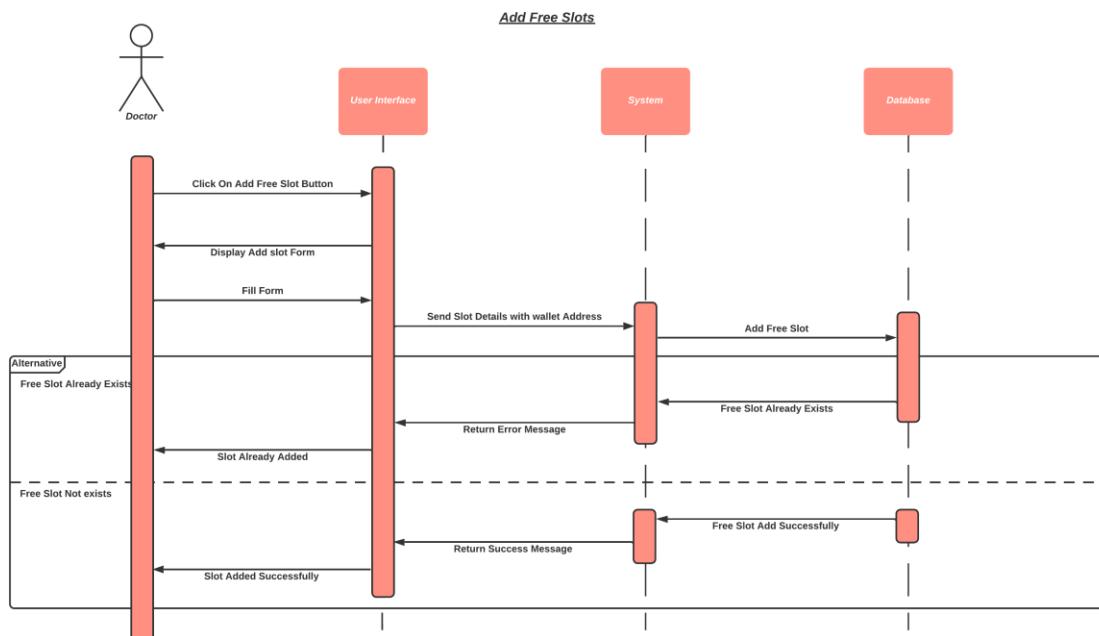
**Figure 20: Login Sequence**  
*Figure Shows sequence of operations in login process*



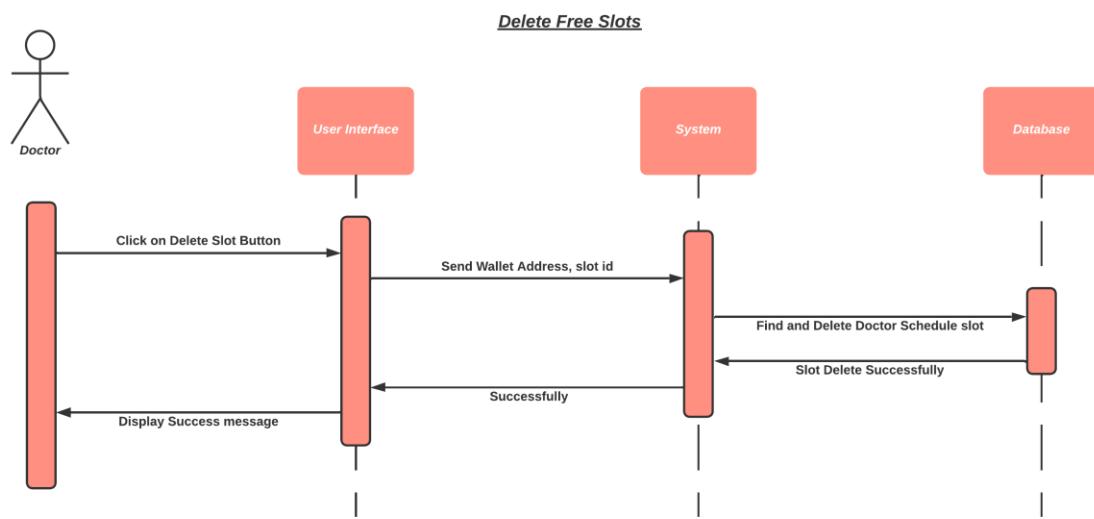
**Figure 21: Edit Profile Sequence**  
*Figure Shows sequence of profile editing process*



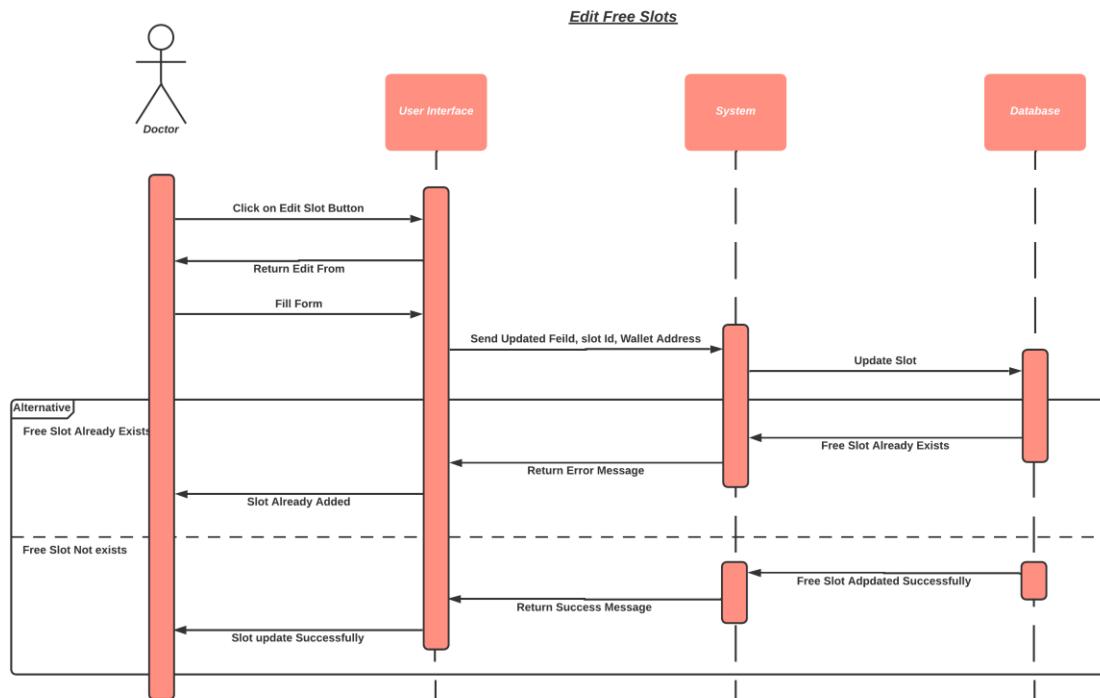
**Figure 22: View Profile Sequence**  
*Figure Shows sequence of operations in displaying profile details*



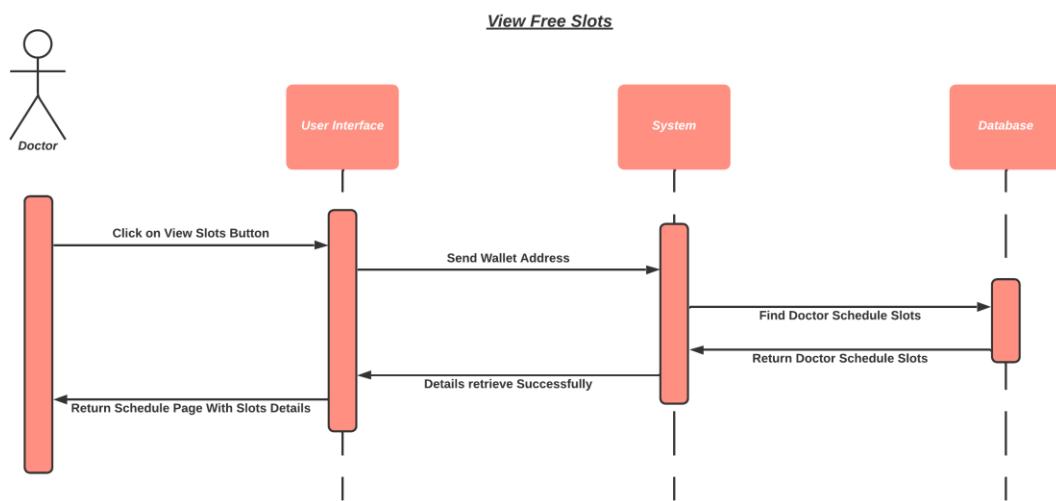
**Figure 23: Add Free Slot Sequence**  
*Figure Shows sequence of Adding new slot in schedule*



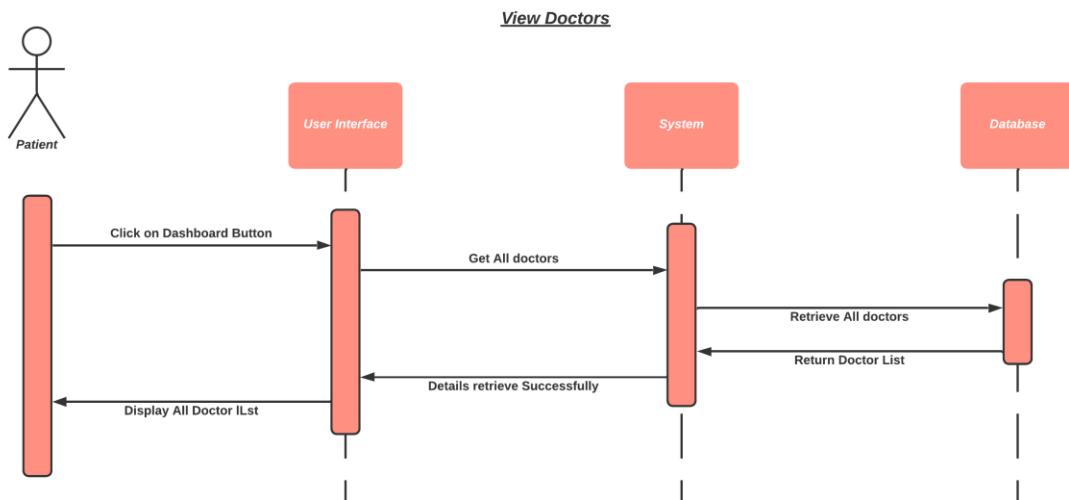
**Figure 24: Delete Free Slot Sequence**  
*Figure Shows sequence of operations in order to remove slot from schedule*



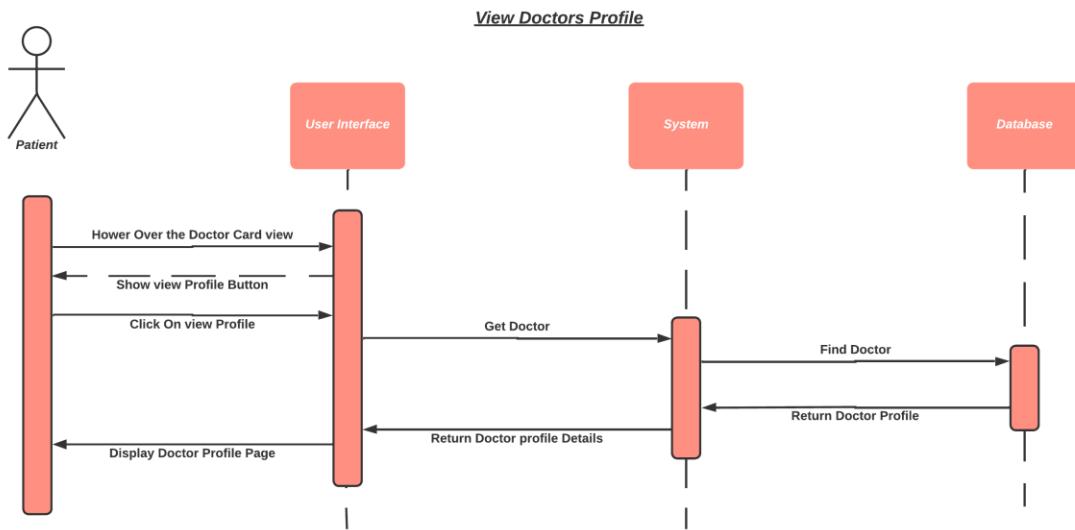
**Figure 25: Edit Slot from Schedule Sequence**  
*Figure Shows sequence to edit slot from doctor schedule*



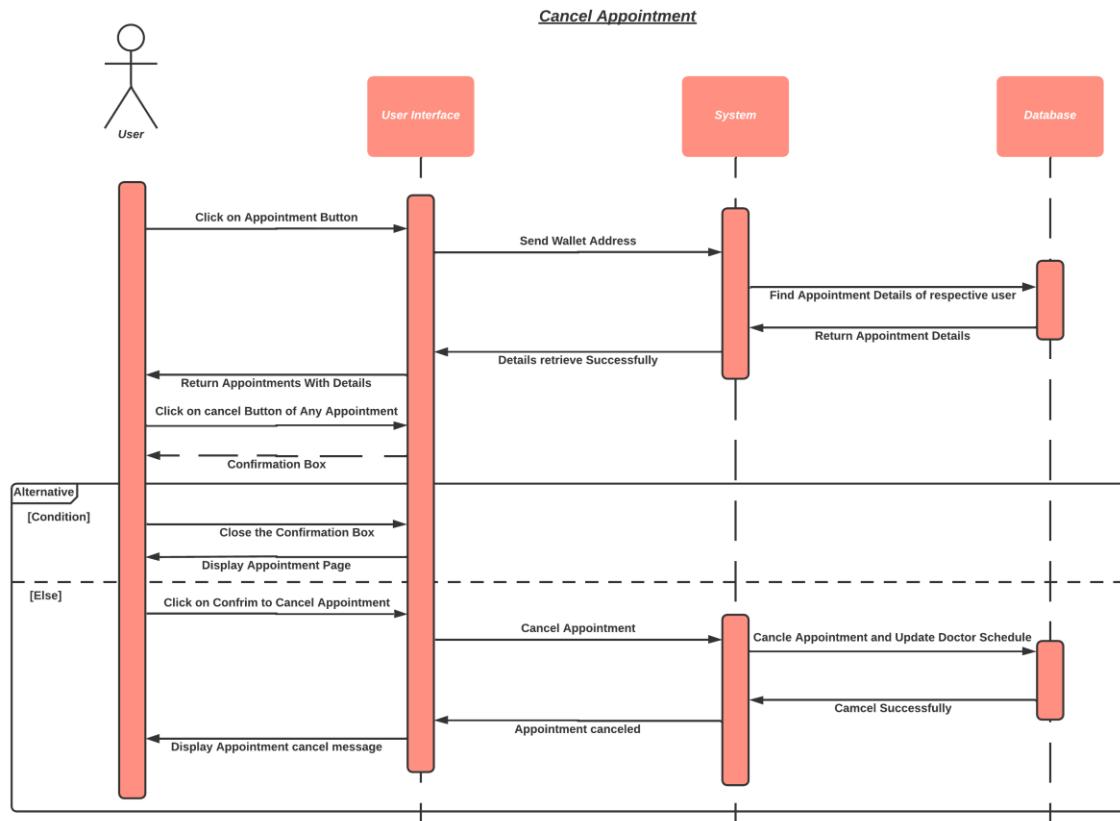
**Figure 26: Delete slot from schedule Sequence**  
*Show sequence of operations to delete time slot from doctor schedule*



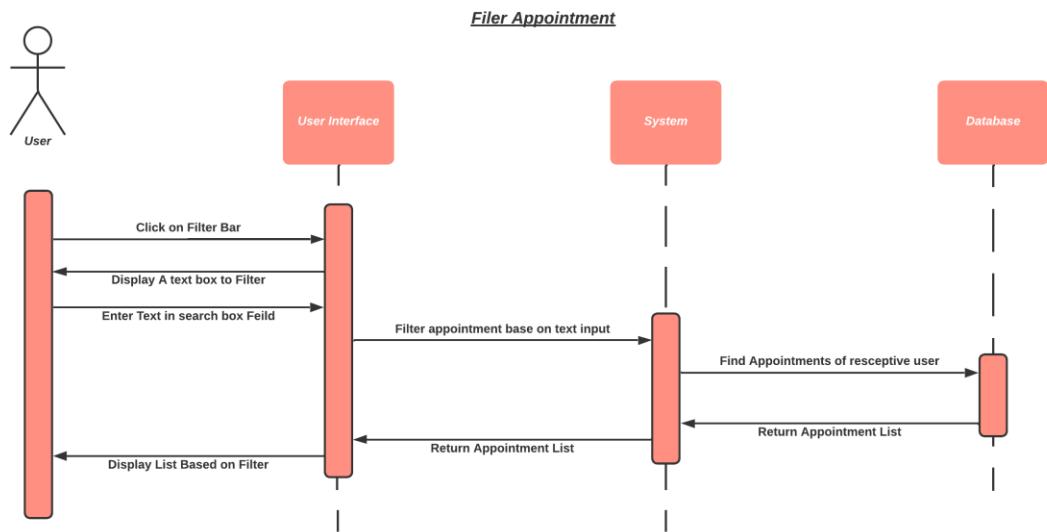
**Figure 27: View Doctors Sequence**  
*Figure Shows sequence of operations to view available doctors fro appointments*



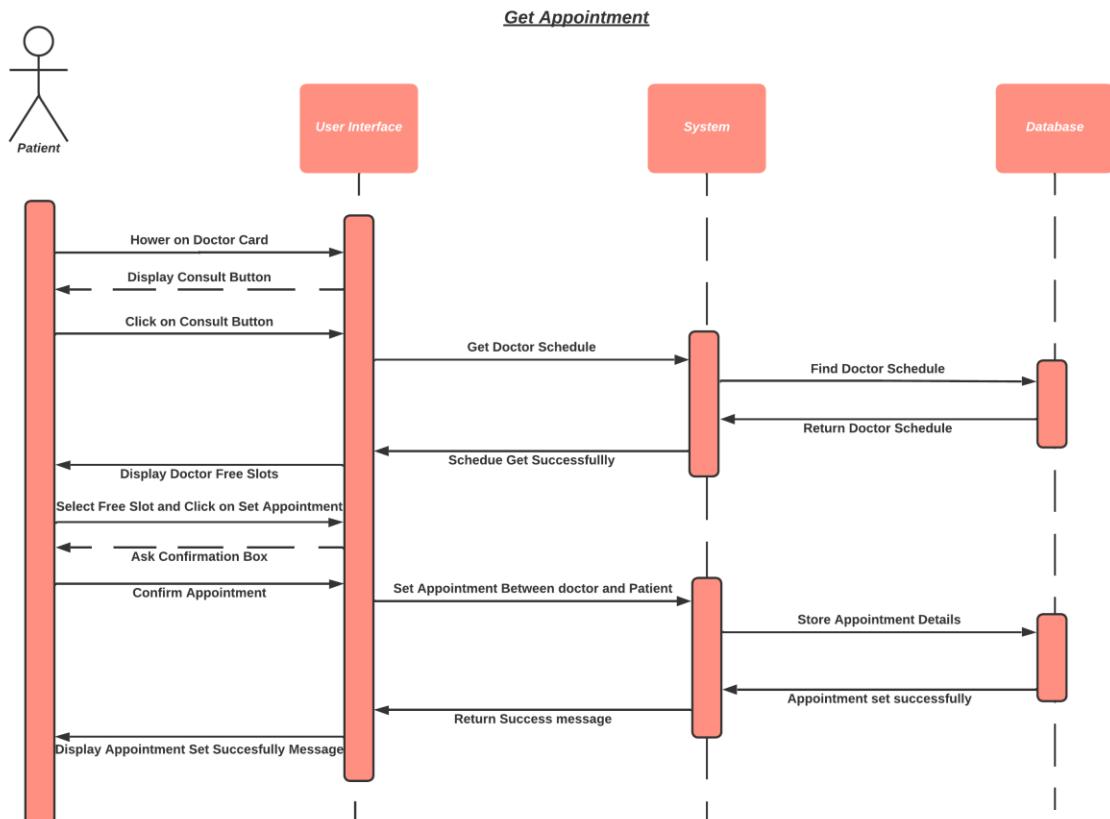
**Figure 28: Doctors Profile Sequence**  
*Show sequence of operations for patient to view any doctors profile details*



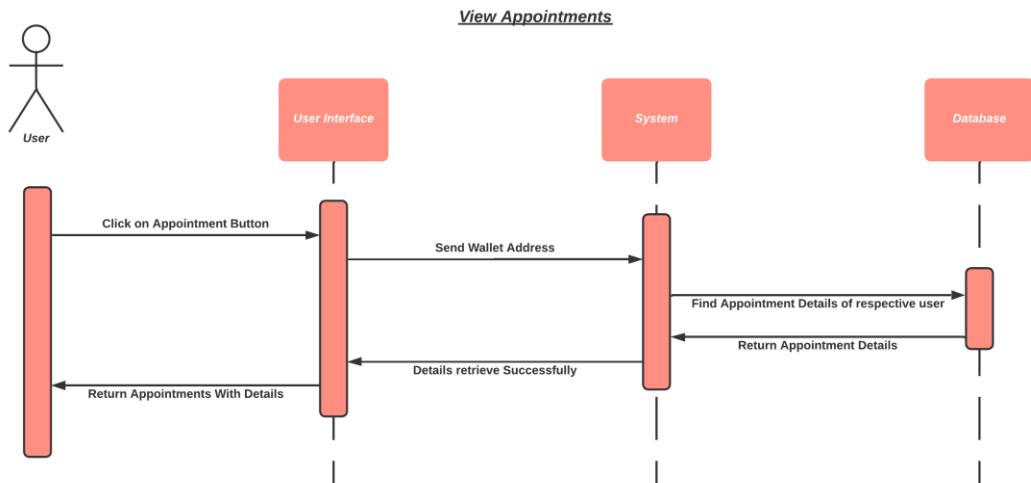
**Figure 29: Cancel Appointment Sequence**  
*Figure Shows sequence of operations in order to cancel appointment*



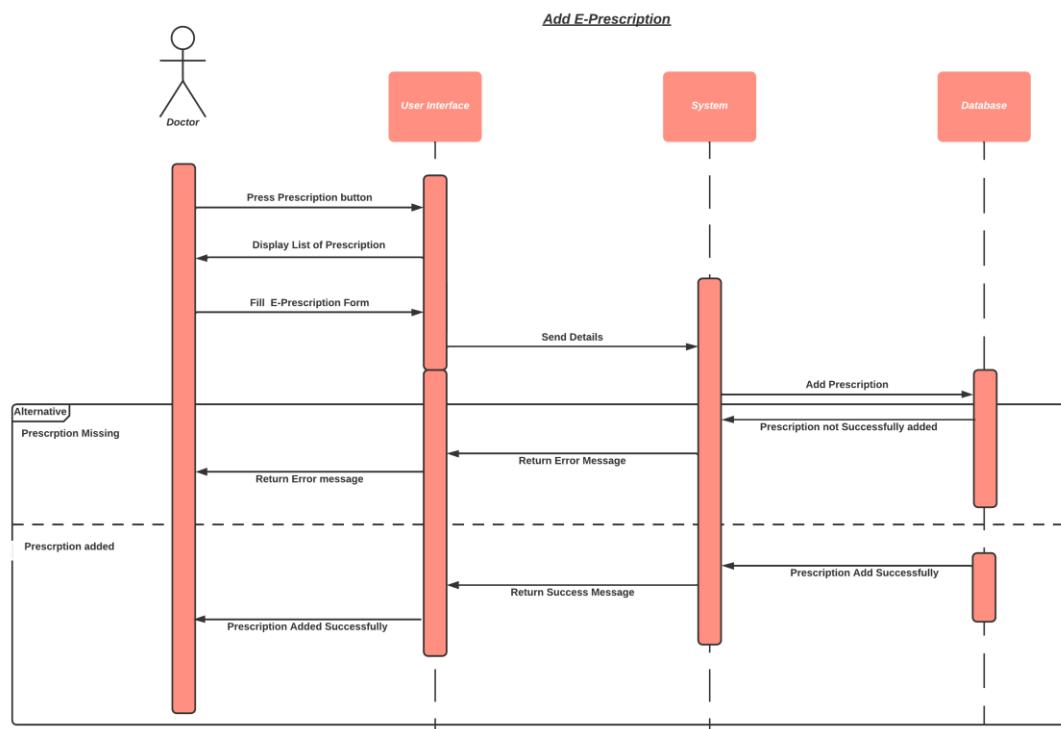
**Figure 30: Filter Appointment Sequence**  
Shows sequence to search specific appointment



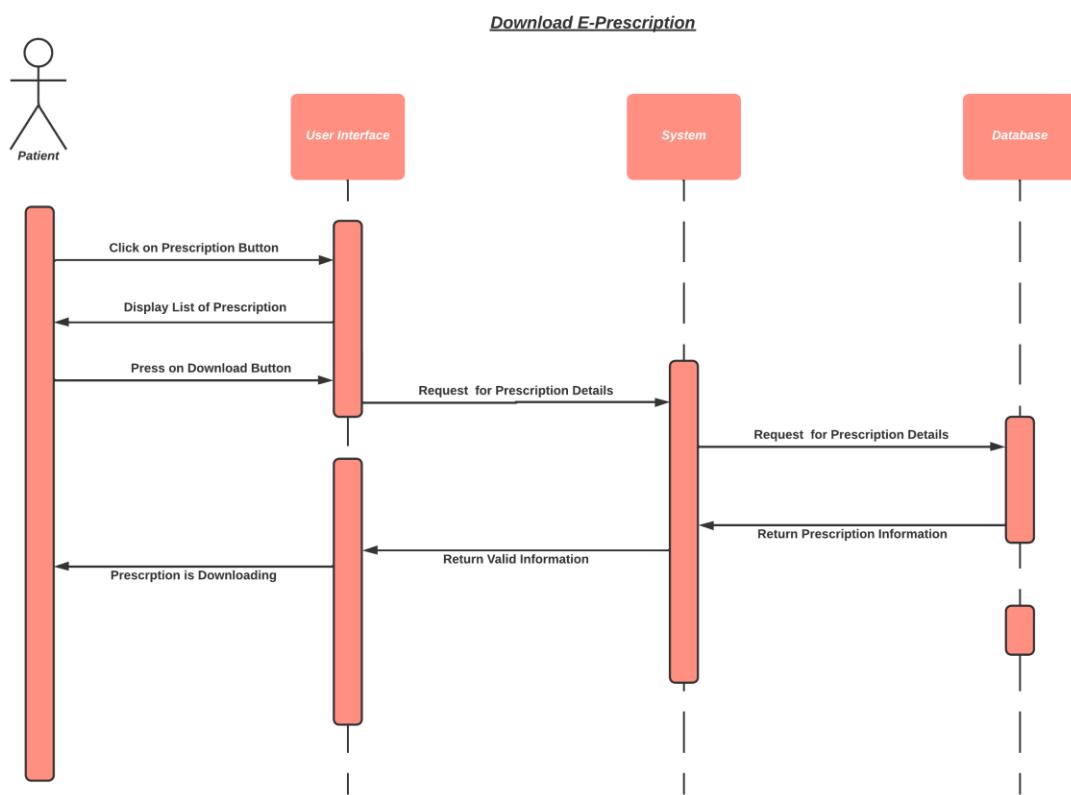
**Figure 31: Get Appointment Sequence**  
Shows sequence to get specific appointment from doctors



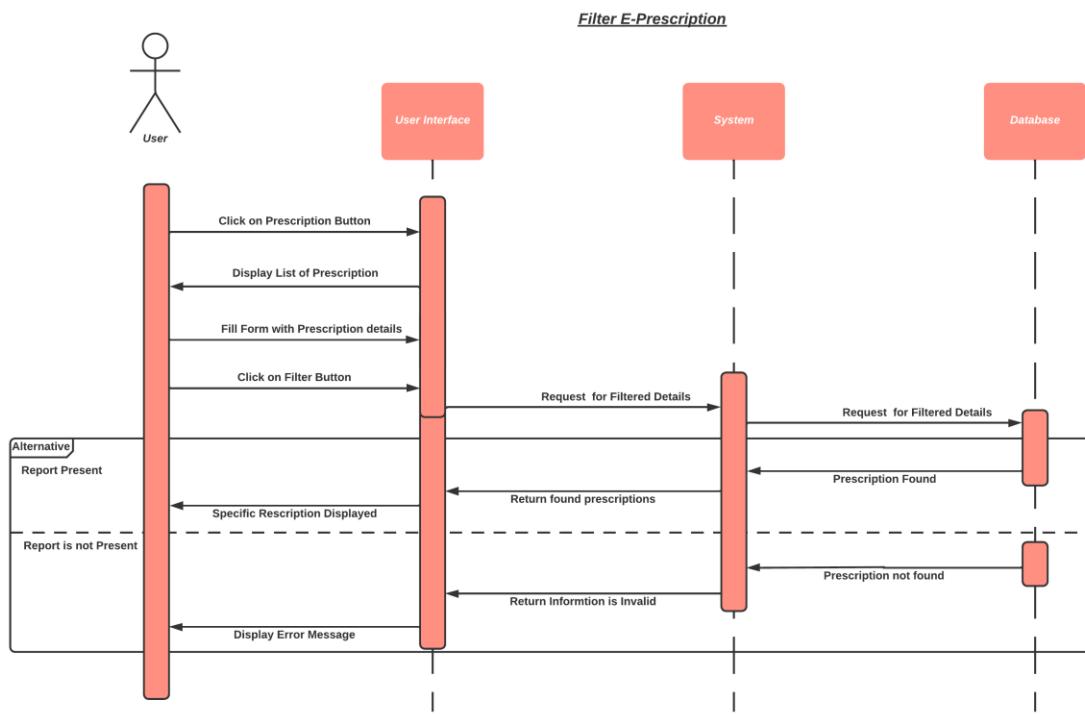
**Figure 32: View Appointment Sequence**  
*Shows sequence to view all pending appointment*



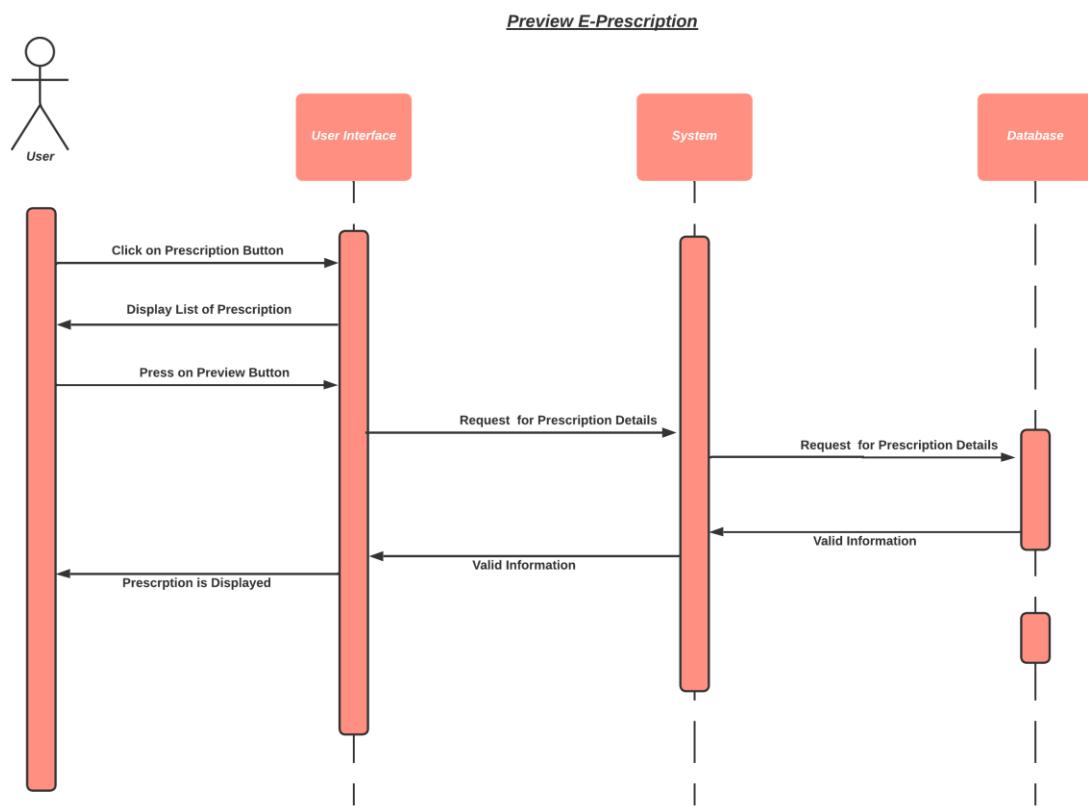
**Figure 33: Add Prescription Sequence**  
*Shows sequence to Add Prescription of specific appointment*



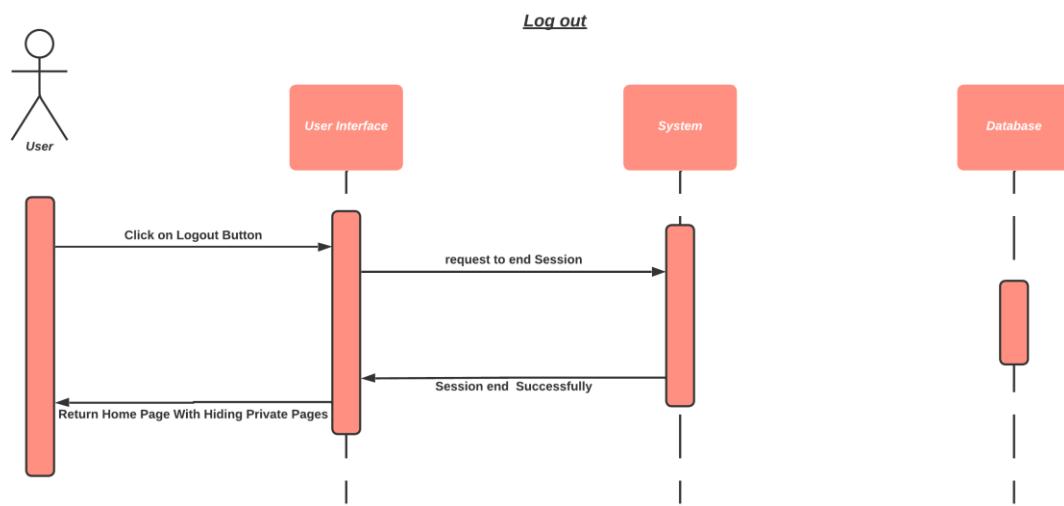
**Figure 34: Download E-Prescription Sequence**  
Figure shows sequence to download specific appointment  
prescription



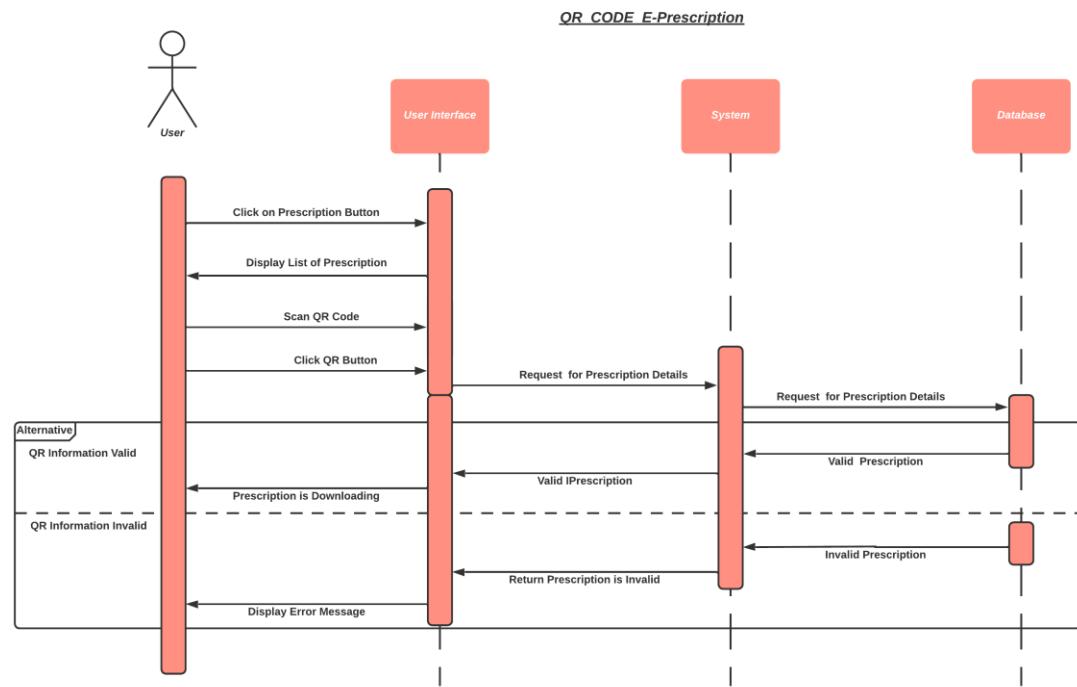
**Figure 35: Filter E-Prescription Sequence**  
*Shows sequence to search specific prescription*



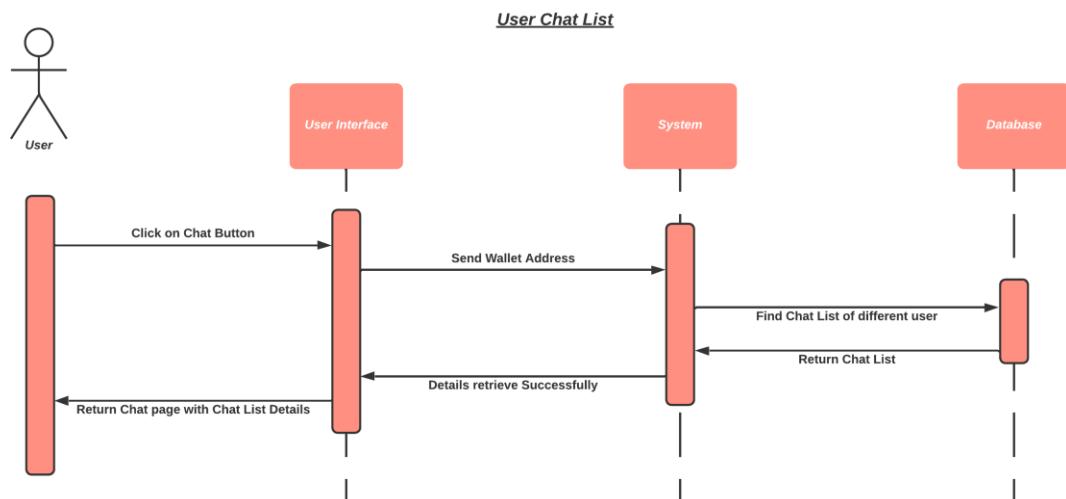
**Figure 36: Preview Prescription Sequence**  
*Figure shows sequence to view specific appointment*



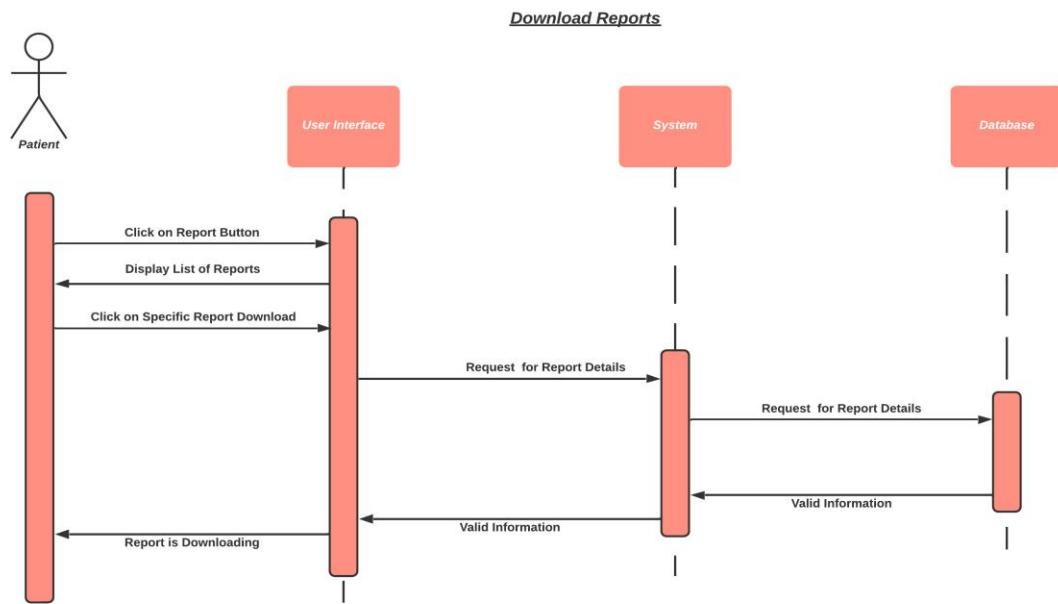
**Figure 37: Logout Sequence**  
*Shows sequence to Log out from application*



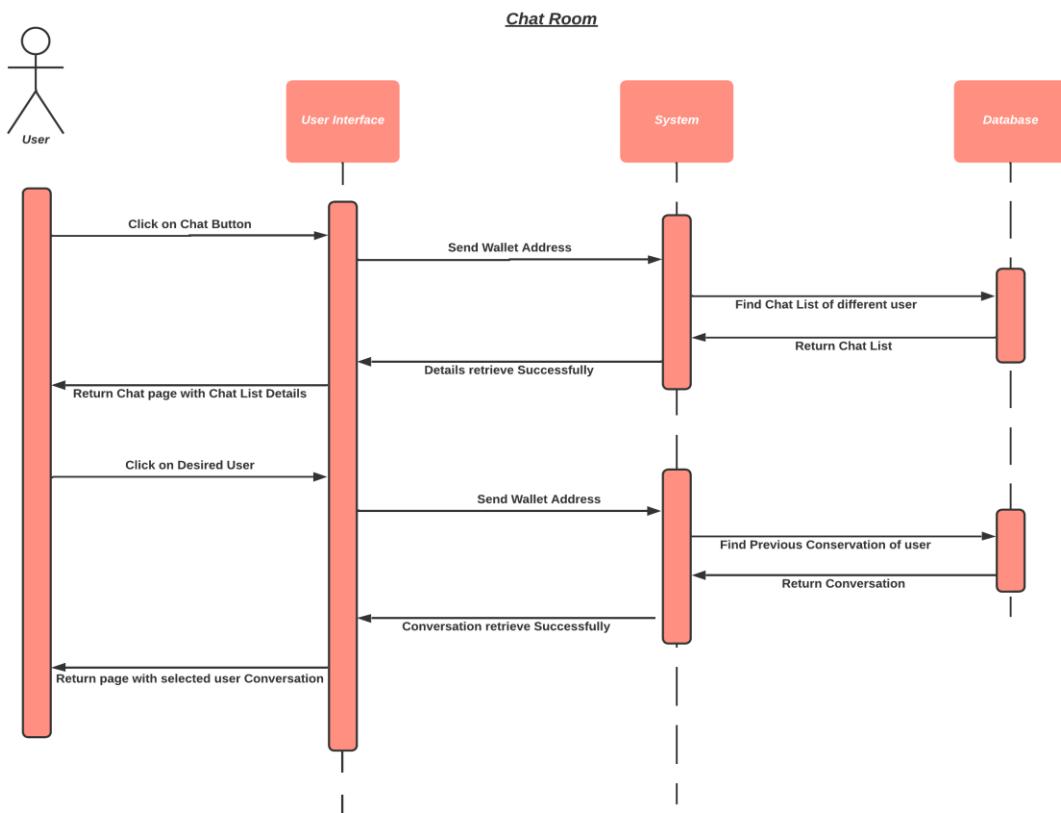
**Figure 38: QR-Code Prescription Sequence**  
Shows sequence to Generate QR Code of prescription



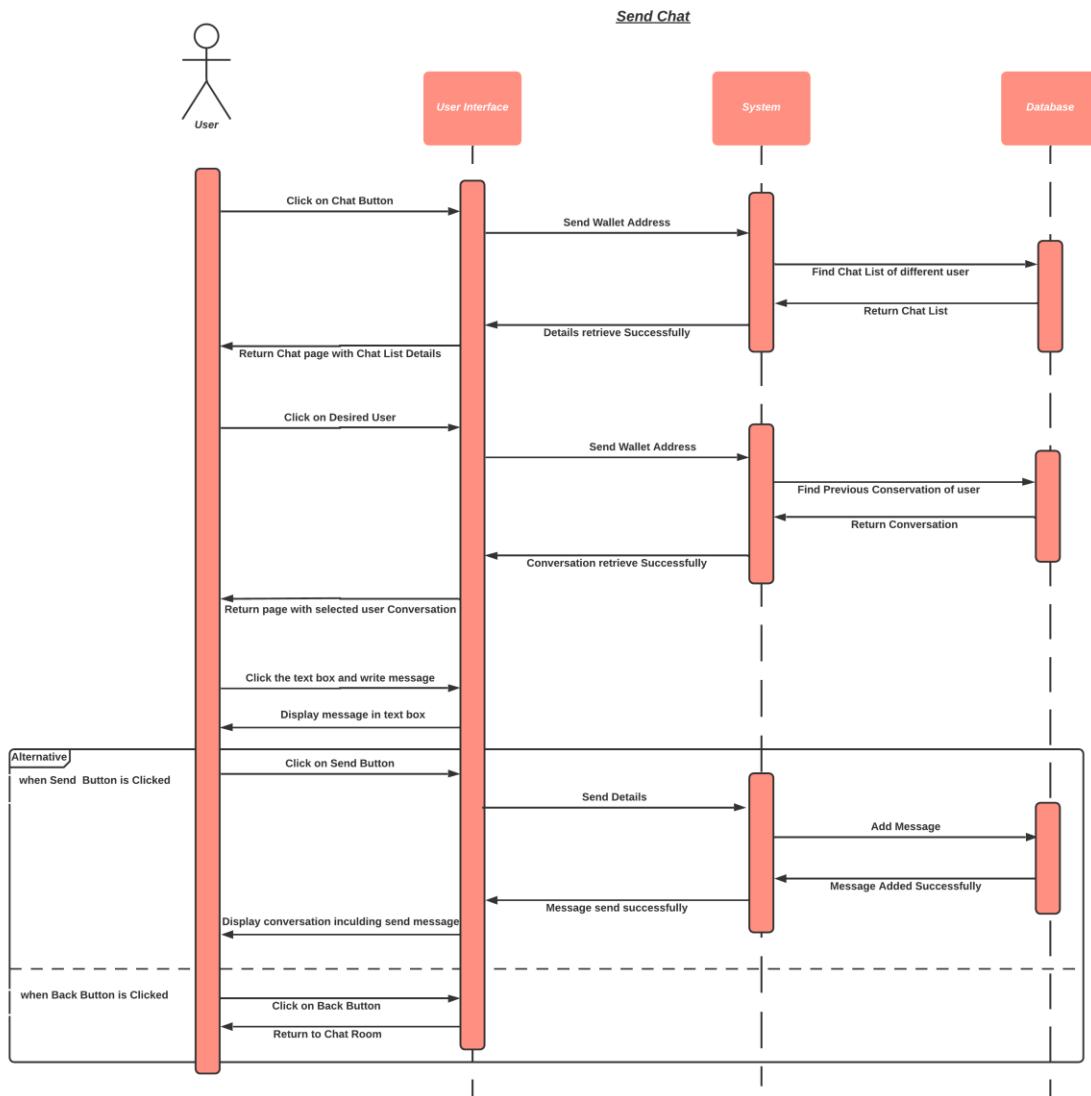
**Figure 39: User Chat Sequence**  
Shows sequence to Display the list of user with chats



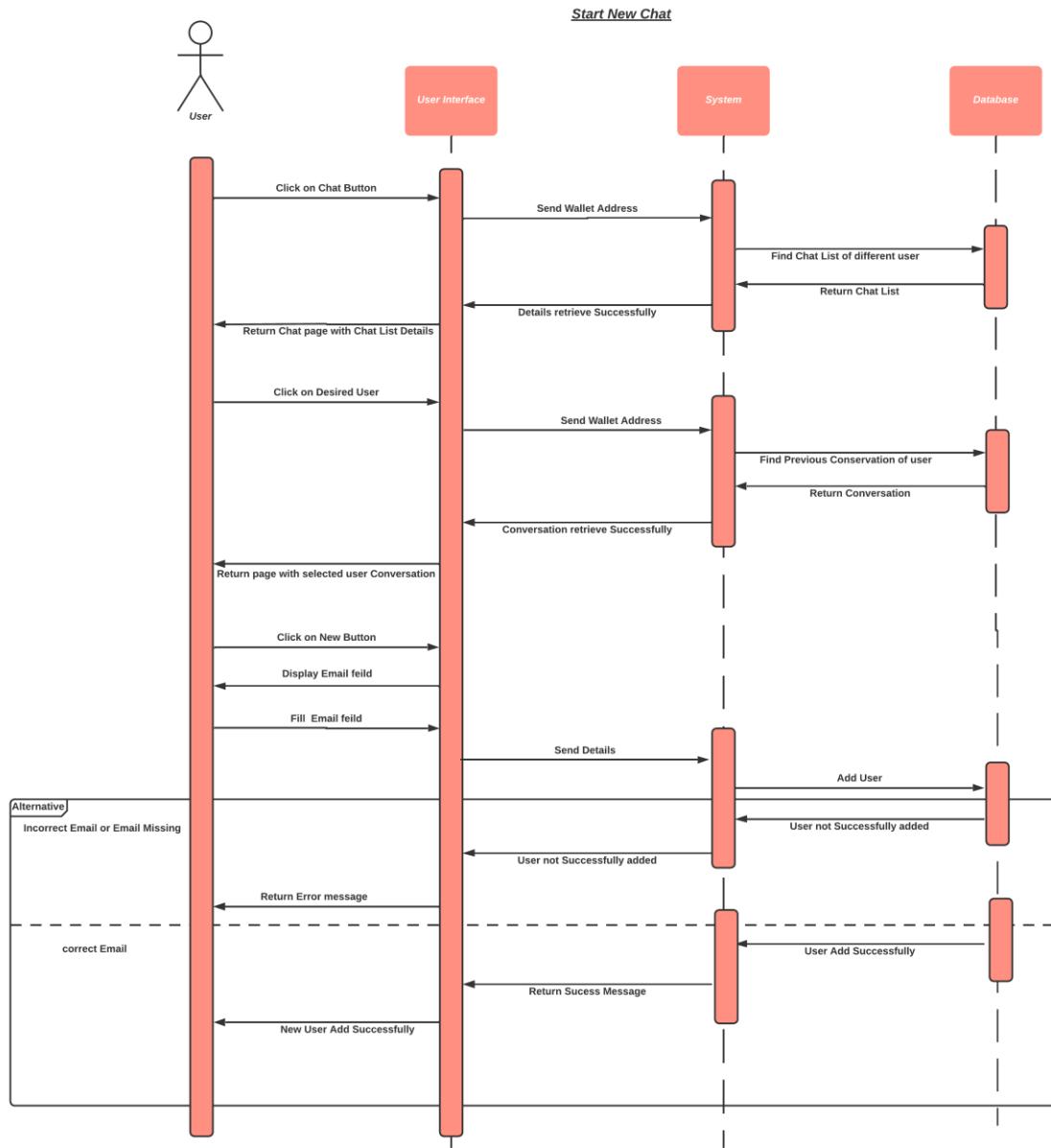
**Figure 40: Download Reports Sequence**  
*Shows sequence to download report of specific appointment*



**Figure 41: Chat Room Sequence**  
*Shows sequence of to display the chat of specific user*



**Figure 42: Send Message Sequence**  
*Figure Shows sequence of operation to send message to specific user*



**Figure 43: Start New Chat Sequence**  
*Figure shows sequence to start new chat room with specific user*

## 5.7 Policies and Tactics

The following are the policies and techniques that will be used throughout the project.

### 5.7.1 Clarinet Compiler for Smart Contract

We are going to use clarinet as a compiler for our smart application. As Clarinet is the only Clarity language compiler that we can use to test our applications smart contracts before deploying them.

### 5.7.2 Tools to be used

Our staff members will use a Prettier for Linting and styling, therefore the styling should be similar across the board. Visual Studio Code will be our IDE of choice because it is lightweight, includes a terminal, and includes out-of-the-box support for JavaScript. Because of its extensive feature set, Visual Studio Code stands out among competing IDEs.

Most crucially, the Smart Contract language we will be using, Clarity, Visual Studio Code provides a plugin that supports Linting and IntelliSense for Clarity smart contract. This is a significant benefit because, as a new language, no other IDE supports Clarity. This aspect alone makes selecting an IDE simple. Moreover, Visual Studio Code provide the GitHub integration and live share feature, which help our team members to work together.

### 5.7.3 Coding guidelines and conventions

Communication between our components must be as secure and transparent as feasible. Furthermore, we want to keep the door open for growth and expansion. A modular design is the best way to accomplish this because it separates each component using interfaces and makes adding more modules simple. Less coupling and cohesion between module and component increase the modularity. Making decentralized software is also regarded excellent strategy, thus this was the natural decision for us.

### 5.7.4 Testing

We shall use both white box but our focus will be on Risk-Based testing. Create a checklist of major risk for our project than priorities all risk and perform testing based on priority risk. Managing and identifying risk is the part of all phases of development life cycle. Stacks platform has created a developer test net. Because real currency in the form of digital tokens (SHIFA Coins) is required to deploy decentralized applications on the main net of the Blockchain, this test net will allow us to evaluate our smart contracts and overall system features without spending real coins.

### 5.7.5 Maintenance

As we might be need to add more and more features, the system should be easily extendable. During the programming phase of the project, rules and standards set for web development would be followed, which increase the modularity of system and help in maintenance.

### 5.7.6 Interfaces for end-users

The eight golden guidelines of interface design would be used as a standard for the user interface. The application will feature a very user-friendly interface that will make it simple to use and understand.

## 5.8 Conclusion

The chapter illustrated the complete system architecture in detail along with the sub-modules. Additionally, it highlights all assumptions, dependencies, constraints and design consideration of the project. Moreover, it includes the possible architectural strategies, policies and additional low-level diagrams e.g. sequence, class/model to better depict the system.

## Chapter 6: Implementation and Test Cases

The implementation details are significant for any system and the primary purpose of documenting it is for better understanding and concept. The chapter will explicitly describe all primary software modules that are being developed or used by the system.

### 6.1 Implementation

The implementation consists of developing different micro-services along with some reusable software components. The front-end layout is defined along with working API endpoints. These are the following components that have been implemented or used so far.

#### 6.1.1 Pakistan Medical Council (PMC)

We are using the API Endpoint provided by Pakistan Medical Council to verify the Doctor's medical registration no against other details. It is a POST method that requires the doctor registration no in JSON format for the body tag. The response returns an object with the following details about the doctor:

- Full Name
- Father Name
- Registration Type
- Registration Date
- Valid Until
- Status
- Is Faculty
- Qualifications List

Endpoint: [POST] <https://www.pmc.gov.pk/api/DRC/GetQualifications>

Requested Schema: {RegistrationNo: "DDDDDD-N"} e.g. 10001-N, 30519-N

#### 6.1.2 Disease Classification Models

Our disease classification models gateway runs on a separate local Flask server. It exposes a GET method that will require different query parameters based on the disease. For instance, the diabetes prediction will need parameters like age, cholesterol, Electrocardiogram (ecg) and etc. The following python libraries are used for model training:

**Sklearn:** For machine learning techniques like the Standard Scaler, the Logistic Regression, and the Random Forest Classifier.

**Pandas:** For reading and manipulating dataset

**Joblib:** For storing and retrieving the trained model

#### 6.1.3 Non-Fungible Token Generation

We have written non-fungible token smart contracts in Clarity and tested them using TypeScript. Furthermore, we have coded an interface in JavaScript to connect them from our back-end server running on NodeJS. For development purposes, we initialized a local instance

of the Stacks Blockchain using Docker and made use of dummy wallets to perform functionality.

### **6.1.4 Asymmetric/Symmetric Cryptography**

We have developed our custom symmetric encryption approach using Advanced Encryption Standard (AES-256) and Secure Hash Algorithm (SHA-256). We add a salt of 16 bytes to further secure our process. For asymmetric encryption, we are utilizing built-in methods of Stacks encryption modules that make use of AES-256 and Secp256k1; used by Bitcoin to implement its public key cryptography.

### **6.1.5 Gaia Decentralized Storage**

We have connected GaiaHub decentralized storage to our application that provides us 10 Gigabytes of free storage per account. Currently, the data is encrypted by a private key generated with the combination of user's wallet key and the app domain for persistency and segmentation. To scale it, we need to configure our own storage servers. Our user profile picture is currently stored on their GaiaHub and served directly from it when they authenticate using their wallet.

### **6.1.6 Wallet Authentication**

The primary authentication for our decentralized application is a crypto wallet. We make use of Hiro Wallet that allows us to quickly connect to our front-end. It returns a session object containing key-pairs as a payload and makes use of JSON Web Token under the hood to maintain user sessions.

### **6.1.7 Appointment Booking**

In this implementation, the patient will begin by searching for relevant doctors using a search bar or a filter system. Once the user has selected a doctor, the system will display the doctor's available time slots for appointments.

The patient will then select a preferred time slot, and the system will prompt the user to confirm the booking. If the user confirms the booking, the system will save the appointment and send a confirmation message to the users.

### **6.1.8 Patient Records**

In the implementation of record module, the doctor or lab technician will be able to add additional information about the patient's medical condition or test results.

For doctors, the report may include details about the patient's diagnosis, prescribed medication, recommended treatments, and any feedback for the patient. The doctor may also add notes about the patient's medical history or any other relevant information.

For labs, the report will include the patient's test results, including any abnormal or concerning values. The lab technician may also include a brief analysis of the results or any additional information that may be useful for the doctor or patient.

Once the report is uploaded, the patient will be notified and able to view the report through their account.

### 6.1.9 Payment Gateway

In the payment gateway module, when a doctor or lab technician uploads a report or prescription for the patient, the system will deduct the corresponding fee from the patient's wallet and transfer it to an escrow account.

The escrow account will hold the funds until the doctor or lab technician confirms that the report or prescription has been uploaded successfully. Once the upload is confirmed, the funds will be transferred from the escrow account to the doctor's or lab technician's wallet.

This process ensures that the patient's payment is protected until they receive the expected service. If there is any issue with the report or prescription, the patient may be able to request a refund or dispute the payment.

## 6.2 Test case Design and description

All the test cases are designed for every use case that has been specified and also all the non-functional requirements that were described. Moreover, an internet connection is required for all test cases.

**Table 22: Test Case ID and Name Mapping**  
*Table shows names of different test cases id and their respective name.*

Test Case ID	Test Case Name
1	User Login
2	Hiro Wallet Extension
3	User Login Alternative
4	Patient Sign Up
5	Patient Sign Up Alternative
6	Doctor Sign Up
7	Doctor Sign Up Alternative 1
8	Doctor Sign Up Alternative 2
9	View Profile
10	Update Profile
11	View Free Slots
12	Add Free Slots
13	Add Free Slots Alternative
14	Edit Free Slots
15	Edit Free Slots Alternative
16	Delete Free Slots
17	View Doctors
18	View Doctors Profile

19	Schedule Appointments
20	Schedule Appointments Alternative
21	View Appointments
22	Cancel Appointments
23	Filter Appointments
24	Add E-Prescriptions
25	Add E-Prescriptions Alternative
26	Download E-Prescription
27	QR Code E-Prescription
28	Preview E-Prescription
29	Doctor View Record
30	Patient View Record
31	Filter E-Prescription
32	Filter E-Prescription Alternative
33	User Chat List
34	Start New Chat
35	Start New Chat Alternative
36	Send Message
37	Send Message Alternative
38	Download Report
39	Preview Report
40	Add Report
41	Add Report Alternative
42	Filter Report
43	Filter Report Alternative
44	Log Out

## 6.2.1 User Login

<b>User Login</b>					
<b>Test Case ID:</b>	1	<b>QA Test Engineer:</b>	Muhammad Burhan		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	Ummar Ikram		
<b>Test Date:</b>	26-03-2023	<b>Use Case Reference(s):</b>	Login		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The user should be able to sign into his/her account.</i>				
<b>Product/Ver/Module:</b>	<i>Sign In Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet</i>				
<b>Pre-Requisite:</b>	<i>The user must be in the database records.</i>				
<b>Step No.</b>	<b>Execution description</b>	<b>Procedure result</b>			
<b>1</b>	<i>The user presses connect wallet button</i>	<i>Dialog box appears asking using to enter password.</i>			
<b>2</b>	<i>User enter the password.</i>	<i>Dialog box appears with different stacks account.</i>			
<b>3</b>	<i>User select and account for login</i>	<i>User redirect to user dashboard page</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

## 6.2.2 Hiro Wallet Extension

Hiro Wallet Extension					
Test Case ID:	2	QA Test Engineer:	Muhammad Burhan		
Test case Version:	1	Reviewed By:	Ummar Ikram		
Test Date:	26-03-2023	Use Case Reference(s):	Login		
Revision History:	<i>Refer to previous test case identity (if any)</i>				
Objective	<i>The user should be able to sign into his/her account.</i>				
Product/Ver/Module:	<i>Sign In Module</i>				
Environment:	<i>Web Browser Working Internet Connection</i>				
Assumptions:	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet</i>				
Pre-Requisite:	<i>The user must be in the database records.</i>				
Step No.	Execution description	Procedure result			
1	<i>The user presses connect wallet button</i>	<i>The system shows an error message: Please Connect The Hiro wallet</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.3 User Login Alternative

User Login Alternative					
Test Case ID:	3	QA Test Engineer:	Muhammad Burhan		
Test case Version:	1	Reviewed By:	Ummar Ikram		
Test Date:	26-03-2023	Use Case Reference(s):	Login		
Revision History:	<i>Refer to previous test case identity (if any)</i>				
Objective	<i>The user should be able to sign into his/her account.</i>				
Product/Ver/Module:	<i>Sign In Module</i>				
Environment:	<i>Web Browser Working Internet Connection</i>				
Assumptions:	<i>1- The system is connected to the Internet</i>				
Pre-Requisite:	<i>The user must be in the database records.</i>				
Step No.	Execution description	Procedure result			
1	<i>The user presses connect wallet button</i>	<i>Dialog box appears asking using to enter password.</i>			
2	<i>User enter the password.</i>	<i>Incorrect Password error is shown.</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

## 6.2.4 Patient Sign Up

<b>Patient Sign Up</b>					
<b>Test Case ID:</b>	4	<b>QA Test Engineer:</b>	Muhammad Burhan		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	Ummar Ikram		
<b>Test Date:</b>	26-03-2023	<b>Use Case Reference(s):</b>	Patient Sign Up		
<b>Revision History:</b>	Refer to previous test case identity (if any)				
<b>Objective</b>	The user should be able to sign up his/her account.				
<b>Product/Ver/Module:</b>	Sign Up Module				
<b>Environment:</b>	Web Browser Working Internet Connection				
<b>Assumptions:</b>	1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet				
<b>Pre-Requisite:</b>	None				
Step No.	Execution description	<b>Procedure result</b>			
1	The user presses connect wallet button	Dialog box appears asking using to enter password.			
2	User enter the password.	Dialog box appears with different stacks account.			
3	User select an account for sign up	User redirect to user register page			
4	User select the patient type and click on verify your email button	Dialog box appear to sign up with Google or GitHub			
5	Patient click on Google or GitHub Button and Select an Account	Dialog box disappears and select account display on register page			
6	Patient press on register button	Patient register successfully and redirect to dashboard			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.5 Patient Sign Up Alternative

<b>Patient Sign Up Alternative</b>					
<b>Test Case ID:</b>	5	<b>QA Test Engineer:</b>	Muhammad Burhan		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	Ummar Ikram		
<b>Test Date:</b>	26-03-2023	<b>Use Case Reference(s):</b>	Patient Sign Up		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The user should be able to sign up his/her account.</i>				
<b>Product/Ver/Module:</b>	<i>Sign Up Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
<b>Step No.</b>	<b>Execution description</b>	<b>Procedure result</b>			
<b>1</b>	<i>The user presses connect wallet button</i>	<i>Dialog box appears asking using to enter password.</i>			
<b>2</b>	<i>User enter the password.</i>	<i>Dialog box appears with different stacks account.</i>			
<b>3</b>	<i>User select an account for sign up</i>	<i>User redirect to user register page</i>			
<b>4</b>	<i>User select the patient type and click on verify your email button</i>	<i>Dialog box appear to sign up with Google or GitHub</i>			
<b>5</b>	<i>Patient click on Google or GitHub Button and Select an Account</i>	<i>Dialog box disappears and select account display on register page</i>			
<b>6</b>	<i>Patient press on register button</i>	<i>Error message appears "Email Already Registered"</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

## 6.2.6 Doctor Sign Up

<b>Doctor Sign Up</b>					
<b>Test Case ID:</b>	6	<b>QA Test Engineer:</b>	Muhammad Burhan		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	Ummar Ikram		
<b>Test Date:</b>	26-03-2023	<b>Use Case Reference(s):</b>	Doctor Sign Up		
<b>Revision History:</b>	Refer to previous test case identity (if any)				
<b>Objective</b>	The user should be able to sign up his/her account.				
<b>Product/Ver/Module:</b>	Sign Up Module				
<b>Environment:</b>	Web Browser Working Internet Connection				
<b>Assumptions:</b>	1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet				
<b>Pre-Requisite:</b>	None				
Step No.	Execution description	<b>Procedure result</b>			
1	The user presses connect wallet button	Dialog box appears asking using to enter password.			
2	User enter the password.	Dialog box appears with different stacks account.			
3	User select an account for sign up	User redirect to user register page			
4	User select the doctor type and click on verify your email button	Dialog box appear to sign up with Google or GitHub			
5	Doctor click on Google or GitHub Button and Select an Account	Dialog box disappears and select account display on register page			
6	Doctor press on next button	Dialog box appears asking using to enter Medical Registration Number.			
7	Doctor enter the registration number and click on register button	Doctor register successfully and redirect to dashboard			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.7 Doctor Sign Up Alternative 1

Doctor Sign Up Alternative 1					
Test Case ID:	7	QA Test Engineer:	Muhammad Burhan		
Test case Version:	1	Reviewed By:	Ummar Ikram		
Test Date:	26-03-2023	Use Case Reference(s):	Doctor Sign Up		
Revision History:	Refer to previous test case identity (if any)				
Objective	The user should be able to sign up his/her account.				
Product/Ver/Module:	Sign Up Module				
Environment:	Web Browser Working Internet Connection				
Assumptions:	1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet				
Pre-Requisite:	None				
Step No.	Execution description	Procedure result			
1	The user presses connect wallet button	Dialog box appears asking using to enter password.			
2	User enter the password.	Dialog box appears with different stacks account.			
3	User select an account for sign up	User redirect to user register page			
4	User select the doctor type and click on verify your email button	Dialog box appear to sign up with Google or GitHub			
5	Doctor click on Google or GitHub Button and Select an Already register Account	Dialog box disappears and select account display on register page			
6	Doctor press on next button	Dialog box appears asking using to enter Medical Registration Number.			
7	Doctor enter the registration number and click on register button	Error message appears "Email Already Registered"			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.8 Doctor Sign Up Alternative 2

Doctor Sign Up Alternative 2					
<b>Test Case ID:</b>	8	<b>QA Test Engineer:</b>	<i>Muhammad Burhan</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	26-03-2023	<b>Use Case Reference(s):</b>	<i>Doctor Sign Up</i>		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The doctor should be able to sign up his/her account.</i>				
<b>Product/Ver/Module:</b>	<i>Sign Up Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	Execution description	Procedure result			
1	<i>The user presses connect wallet button</i>	<i>Dialog box appears asking using to enter password.</i>			
2	<i>User enter the password.</i>	<i>Dialog box appears with different stacks account.</i>			
3	<i>User select an account for sign up</i>	<i>User redirect to user register page</i>			
4	<i>User select the doctor type and click on verify your email button</i>	<i>Dialog box appear to sign up with Google or GitHub</i>			
5	<i>Doctor click on Google or GitHub Button and Select an Already register Account</i>	<i>Dialog box disappears and select account display on register page</i>			
6	<i>Doctor press on next button</i>	<i>Dialog box appears asking using to enter Medical Registration Number.</i>			
7	<i>Doctor enter the wrong registration number and click on register button</i>	<i>Error message appears "Record not found"</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.9 View Profile

<b>View Profile</b>					
<b>Test Case ID:</b>	9	<b>QA Test Engineer:</b>	<i>Muhammad Burhan</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	26-03-2023	<b>Use Case Reference(s):</b>	<i>View Profile</i>		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The user should be able to view his/her profile.</i>				
<b>Product/Ver/Module:</b>	<i>Profile Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The user must be login 3- The system is connected to the Internet</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	<b>Execution description</b>		<b>Procedure result</b>		
1	<i>The user presses profile button</i>		<i>Profile page is displayed</i>		
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

## 6.2.10 Update Profile

<b>Update Profile</b>					
<b>Test Case ID:</b>	10	<b>QA Test Engineer:</b>	Muhammad Burhan		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	Ummar Ikram		
<b>Test Date:</b>	26-03-2023	<b>Use Case Reference(s):</b>	Update Profile		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The user should be able to update his/her profile.</i>				
<b>Product/Ver/Module:</b>	<i>Profile Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The user must be login 3- The system is connected to the Internet</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	<b>Execution description</b>		<b>Procedure result</b>		
1	<i>The user presses profile button</i>		<i>Profile page is displayed</i>		
2	<i>User updates the information and presses the “Save” button.</i>		<i>System updates the information and displays a success message “Update successfully”</i>		
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.11 View Free Slots

View Free Slots					
<b>Test Case ID:</b>	11	<b>QA Test Engineer:</b>	Muhammad Burhan		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	Ummar Ikram		
<b>Test Date:</b>	26-03-2023	<b>Use Case Reference(s):</b>	View Free Slots		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The user should be able to view his/her schedule.</i>				
<b>Product/Ver/Module:</b>	<i>Doctor Dashboard Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The user must be login 3- The system is connected to the Internet</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	Execution description	Procedure result			
1	<i>The doctor presses dashboard button</i>	<i>Dashboard page is displayed with list of free slots table.</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.12 Add Free Slots

<b>Add Free Slots</b>					
<b>Test Case ID:</b>	12	<b>QA Test Engineer:</b>	Muhammad Burhan		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	Ummar Ikram		
<b>Test Date:</b>	26-03-2023	<b>Use Case Reference(s):</b>	Add Free Slots		
<b>Revision History:</b>	Refer to previous test case identity (if any)				
<b>Objective</b>	The Doctor should be able to schedule his/her free slot.				
<b>Product/Ver/Module:</b>	Doctor Dashboard Module				
<b>Environment:</b>	Web Browser Working Internet Connection				
<b>Assumptions:</b>	1- Hiro Wallet extension is added to the browser. 2- The user must be login 3- The system is connected to the Internet				
<b>Pre-Requisite:</b>	None				
Step No.	Execution description	<b>Procedure result</b>			
<b>1</b>	The doctor presses dashboard button	Dashboard page is displayed with list of free slots table.			
<b>2</b>	The doctor presses the plus button in time slot table.	Dialog box appears asking for the user enter time and date.			
<b>3</b>	The doctor enter time, date, location and presses "Add Slot"	The system validate the information, add the free slot and displays a success message.			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.13 Add Free Slots Alternative

Add Free Slots Alternative					
Test Case ID:	13	QA Test Engineer:	Muhammad Burhan		
Test case Version:	1	Reviewed By:	Ummar Ikram		
Test Date:	26-03-2023	Use Case Reference(s):	Add Free Slots		
Revision History:	Refer to previous test case identity (if any)				
Objective	The Doctor should be able to add his/her schedule.				
Product/Ver/Module:	Doctor Dashboard Module				
Environment:	Web Browser Working Internet Connection				
Assumptions:	1- Hiro Wallet extension is added to the browser. 2- The user must be login 3- The system is connected to the Internet				
Pre-Requisite:	None				
Step No.	Execution description	Procedure result			
1	The doctor presses dashboard button	Dashboard page is displayed with list of free slots table.			
2	The doctor presses the plus button in time slot table.	Dialog box appears asking for the user enter time and date.			
3	The doctor enter time, date, location and presses "Add Slot"	The system responds with an error message: Free slot already added.			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.14 Edit Free Slots

Edit Free Slots					
<b>Test Case ID:</b>	14	<b>QA Test Engineer:</b>	Muhammad Burhan		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	Ummar Ikram		
<b>Test Date:</b>	26-03-2023	<b>Use Case Reference(s):</b>	Add Free Slots		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The Doctor should be able to schedule his/her free slot.</i>				
<b>Product/Ver/Module:</b>	<i>Doctor Dashboard Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The user must be login 3- The system is connected to the Internet</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	Execution description	Procedure result			
<b>1</b>	<i>The doctor presses dashboard button</i>	<i>Dashboard page is displayed with list of free slots table.</i>			
<b>2</b>	<i>The doctor presses the Edit button in time slot table.</i>	<i>Dialog box appears asking for the user to enter the updated time and date.</i>			
<b>3</b>	<i>The doctor enter time, date, location and presses "Add Slot"</i>	<i>The system validate the information, add the free slot and displays a success message.</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.15 Edit Free Slots Alternative

Edit Free Slots Alternative					
Test Case ID:	15	QA Test Engineer:	Muhammad Burhan		
Test case Version:	1	Reviewed By:	Ummar Ikram		
Test Date:	26-03-2023	Use Case Reference(s):	Edit Free Slots		
Revision History:	Refer to previous test case identity (if any)				
Objective	The Doctor should be able to add his/her schedule.				
Product/Ver/Module:	Doctor Dashboard Module				
Environment:	Web Browser Working Internet Connection				
Assumptions:	1- Hiro Wallet extension is added to the browser. 2- The user must be login 3- The system is connected to the Internet				
Pre-Requisite:	None				
Step No.	Execution description	Procedure result			
1	The doctor presses dashboard button	Dashboard page is displayed with list of free slots table.			
2	The doctor presses the Edit button in time slot table.	Dialog box appears asking for the user to enter the updated time and date.			
3	The doctor enter time, date, location and presses "Add Slot"	The system responds with an error message: Free slot already added.			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.16 Delete Free Slots

<b>Delete Free Slots</b>					
<b>Test Case ID:</b>	16	<b>QA Test Engineer:</b>	<i>Muhammad Burhan</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	26-03-2023	<b>Use Case Reference(s):</b>	<i>Delete Free Slots</i>		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The Doctor should be able to delete his/her free slots.</i>				
<b>Product/Ver/Module:</b>	<i>Doctor Dashboard Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The user must be login 3- The system is connected to the Internet</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	<b>Execution description</b>		<b>Procedure result</b>		
<b>1</b>	<i>The doctor presses dashboard button</i>		<i>Dashboard page is displayed with list of free slots table.</i>		
<b>2</b>	<i>The doctor presses the delete button in time slot table.</i>		<i>Dialog box appears asking for confirmation</i>		
<b>3</b>	<i>The doctor click on confirm</i>		<i>The system remove the free slot</i>		
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.17 View Doctors

<b>View Doctors</b>					
<b>Test Case ID:</b>	17	<b>QA Test Engineer:</b>	Muhammad Burhan		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	Ummar Ikram		
<b>Test Date:</b>	26-03-2023	<b>Use Case Reference(s):</b>	View Doctors		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The Patient should be able to view list of doctors.</i>				
<b>Product/Ver/Module:</b>	<i>Patient Dashboard Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The user must be login 3- The system is connected to the Internet</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	Execution description	<b>Procedure result</b>			
1	<i>The patient presses dashboard button</i>	<i>Dashboard page is displayed with list of Doctors</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.18 View Doctors Profile

<b>View Doctors Profile</b>					
<b>Test Case ID:</b>	18	<b>QA Test Engineer:</b>	Muhammad Burhan		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	Ummar Ikram		
<b>Test Date:</b>	26-03-2023	<b>Use Case Reference(s):</b>	View Doctors profile		
<b>Revision History:</b>	None				
<b>Objective</b>	The Patient should be able to view Doctor's profile.				
<b>Product/Ver/Module:</b>	Patient Dashboard Module				
<b>Environment:</b>	Web Browser Working Internet Connection				
<b>Assumptions:</b>	1- Hiro Wallet extension is added to the browser. 2- The user must be login 3- The system is connected to the Internet				
<b>Pre-Requisite:</b>	None				
Step No.	Execution description	<b>Procedure result</b>			
1	The patient presses dashboard button	Dashboard page is displayed with list of Doctors			
2	The user presses the "Profile" button	The profile page of doctor is appeared			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.19 Schedule Appointments

Schedule Appointments					
Test Case ID:	19	QA Test Engineer:	Muhammad Burhan		
Test case Version:	1	Reviewed By:	Ummar Ikram		
Test Date:	26-03-2023	Use Case Reference(s):	Get Appointment		
Revision History:	None				
Objective	<i>The Patient should be able to schedule Appointment with Doctor.</i>				
Product/Ver/Module:	<i>Appointment Module</i>				
Environment:	<i>Web Browser Working Internet Connection</i>				
Assumptions:	<i>1- Hiro Wallet extension is added to the browser. 2- The user must be login 3- The system is connected to the Internet</i>				
Pre-Requisite:	None				
Step No.	Execution description	Procedure result			
<b>1</b>	<i>The patient presses dashboard button</i>	<i>Dashboard page is displayed with list of Doctors</i>			
<b>2</b>	<i>The user presses the “Profile” button</i>	<i>The profile page of doctor is appeared</i>			
<b>3</b>	<i>User click on Book button of available slot</i>	<i>The free slot is highlighted and button color changed .</i>			
<b>4</b>	<i>User presses the “Yes” button.</i>	<i>Appointment booked successfully</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.20 Schedule Appointments Alternative

<b>Schedule Appointments Alternative</b>					
<b>Test Case ID:</b>	20	<b>QA Test Engineer:</b>	Muhammad Burhan		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	Ummar Ikram		
<b>Test Date:</b>	26-03-2023	<b>Use Case Reference(s):</b>	Get Appointment		
<b>Revision History:</b>	None				
<b>Objective</b>	<i>The Patient should be able to schedule Appointment with Doctor.</i>				
<b>Product/Ver/Module:</b>	<i>Appointment Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The user must be login 3- The system is connected to the Internet</i>				
<b>Pre-Requisite:</b>	None				
Step No.	Execution description	<b>Procedure result</b>			
<b>1</b>	<i>The patient presses dashboard button</i>	<i>Dashboard page is displayed with list of Doctors</i>			
<b>2</b>	<i>The user presses the “Profile” button</i>	<i>The profile page of doctor is appeared</i>			
<b>3</b>	<i>User click on Book button of available slot</i>	<i>The free slot is highlighted and button color changed.</i>			
<b>4</b>	<i>User presses the “No” button.</i>	<i>The confirmation box disappear and Appointment is not booked.</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.21 View Appointments

View Appointments					
<b>Test Case ID:</b>	21	<b>QA Test Engineer:</b>	Muhammad Burhan		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	Ummar Ikram		
<b>Test Date:</b>	26-03-2023	<b>Use Case Reference(s):</b>	View Appointment		
<b>Revision History:</b>	<i>None</i>				
<b>Objective</b>	<i>The Patient should be able to view Appointment with Doctor.</i>				
<b>Product/Ver/Module:</b>	<i>Appointment Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The user must be login 3- The system is connected to the Internet</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	Execution description	Procedure result			
1	<i>The user presses the "Appointment" tab.</i>	<i>The "Appointment" page is opened and display the list of Appointment schedule by user.</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.22 Cancel Appointments

Cancel Appointments					
<b>Test Case ID:</b>	22	<b>QA Test Engineer:</b>	<i>Muhammad Burhan</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	26-03-2023	<b>Use Case Reference(s):</b>	<i>Cancel Appointment</i>		
<b>Revision History:</b>	<i>None</i>				
<b>Objective</b>	<i>The user should be able to cancel Appointment with Doctor.</i>				
<b>Product/Ver/Module:</b>	<i>Appointment Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The user must be login 3- The system is connected to the Internet</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	Execution description	Procedure result			
<b>1</b>	<i>The user presses the “Appointment” tab.</i>	<i>The “Appointment” page is opened and display the list of Appointment schedule by user.</i>			
<b>2</b>	<i>The user presses the “cancel button”.</i>	<i>System update the status of appointment to cancel.</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.23 Filter Appointments

Filter Appointments			
Step No.	Execution description	Procedure result	
<b>1</b>	<i>The user presses the “Appointment” tab.</i>	<i>The “Appointment” page is opened and display the list of Appointment schedule by user.</i>	
<b>2</b>	<i>User fill the search form by entering required appointment details (date, patient email, or time slot )</i>	<i>The system read all the information and displayed it in an input box</i>	
<b>3</b>	<i>Click the “Filter” button</i>	<i>Displays list of Appointment related to search</i>	
<b>Comments:</b> The test case is passed. Our system is working properly.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

### 6.2.24 Add E-Prescription

Add E-Prescription					
<b>Test Case ID:</b>	24	<b>QA Test Engineer:</b>	<i>Umar Farooq</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	15-04-2023	<b>Use Case Reference(s):</b>	<i>Doctor Add E-Prescription</i>		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The doctor should be able to add E-Prescription for patient.</i>				
<b>Product/Ver/Module:</b>	<i>E-Prescription Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet 3-The user must be login</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	Execution description	Procedure result			
1	<i>The user presses the Prescription tab.</i>	<i>Redirect to prescription page which contain a form page.</i>			
2	<i>Doctor fill the add prescription form by entering required prescription details (patient email, purpose and drugs details).</i>	<i>The system validate the information, add the prescription with a success message.</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.25 Add E-Prescription Alternative

Add E-Prescription Alternative					
<b>Test Case ID:</b>	25	<b>QA Test Engineer:</b>	<i>Umar Farooq</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	15-04-2023	<b>Use Case Reference(s):</b>	<i>Doctor Add E-Prescription</i>		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The doctor should be able to add E-Prescription for patient.</i>				
<b>Product/Ver/Module:</b>	<i>E-Prescription Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet 3-The user must be login</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	Execution description	Procedure result			
1	<i>The user presses the Prescription tab.</i>	<i>Redirect to prescription page which contain list of prescriptions and a form page.</i>			
2	<i>User fill the add prescription form by entering required prescription details (patient email, purpose and drugs details).</i>	<i>The system responds with an error message.</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.26 Download E-Prescription

Download E-Prescription					
<b>Test Case ID:</b>	26	<b>QA Test Engineer:</b>	<i>Umar Farooq</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	15-04-2023	<b>Use Case Reference(s):</b>	<i>User Download E-Prescription</i>		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The doctor should able to download E-Prescription for patient.</i>				
<b>Product/Ver/Module:</b>	<i>E-Prescription Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet 3-The user must be login</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	Execution description	Procedure result			
1	<i>The user presses the Prescription tab.</i>	<i>Redirect to prescription page which contain list of prescriptions and a form page.</i>			
2	<i>User click on download button</i>	<i>The system validate the information, download the prescription and displays a success message.</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.27 QR Code E-Prescription

QR Code E-Prescription					
<b>Test Case ID:</b>	27	<b>QA Test Engineer:</b>	<i>Umar Farooq</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	15-04-2023	<b>Use Case Reference(s):</b>	<i>QR Code E-Prescription</i>		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The doctor should able to download QR Code E-Prescription for patient.</i>				
<b>Product/Ver/Module:</b>	<i>E-Prescription Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet 3-The user must be login</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	Execution description	Procedure result			
1	<i>The user presses the Prescription tab.</i>	<i>Redirect to prescription page which contain list of prescriptions and a form page.</i>			
2	<i>User click on desired QR Code of prescription.</i>	<i>The system validate the information, download the QR Code of prescription and displays a success message.</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.28 Preview E-Prescription

Preview E-Prescription					
<b>Test Case ID:</b>	28	<b>QA Test Engineer:</b>	<i>Umar Farooq</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	15-04-2023	<b>Use Case Reference(s):</b>	<i>User view E-Prescription</i>		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The doctor should able to view E-Prescription for patient.</i>				
<b>Product/Ver/Module:</b>	<i>E-Prescription Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet 3-The user must be login</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	Execution description	Procedure result			
1	<i>The user presses the Prescription tab.</i>	<i>Redirect to prescription page which contain list of prescriptions and a form page.</i>			
2	<i>User click on desired "Preview" of prescription.</i>	<i>The system validate the information, start display the prescription and displays a success message.</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.29 Doctor View Record

Doctor/Lab technician View Record					
<b>Test Case ID:</b>	29	<b>QA Test Engineer:</b>	<i>Umar Farooq</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	15-04-2023	<b>Use Case Reference(s):</b>	<i>Doctor/Lab Technician view Record</i>		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The doctor/lab technician should able to view E-Prescription for patient.</i>				
<b>Product/Ver/Module:</b>	<i>E-Prescription Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet 3-The user must be login</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	Execution description	Procedure result			
1	<i>The doctor/lab technician presses the record(prescription, reports) tab.</i>	<i>Redirect to record page which contain a form page (prescription, reports).</i>			
2	<i>User click on view record (prescription, reports) toggle button.</i>	<i>Start display the record with a success message.</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.30 Patient View Record

<b>Patient View Record</b>					
<b>Test Case ID:</b>	30	<b>QA Test Engineer:</b>	<i>Umar Farooq</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	15-04-2023	<b>Use Case Reference(s):</b>	<i>Patient view Record</i>		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The doctor/lab technician should able to view E-Prescription for patient.</i>				
<b>Product/Ver/Module:</b>	<i>E-Prescription Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet 3-The user must be login</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
<b>Step No.</b>	<b>Execution description</b>	<b>Procedure result</b>			
1	<i>The patient presses the record(prescription, reports) tab.</i>	<i>Start display the record with a success message.</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.31 Filter E-Prescription

<b>Filter E-Prescription</b>					
<b>Test Case ID:</b>	31	<b>QA Test Engineer:</b>	<i>Umar Farooq</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	15-04-2023	<b>Use Case Reference(s):</b>	<i>User search E-Prescription</i>		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The doctor should able to search E-Prescription for patient.</i>				
<b>Product/Ver/Module:</b>	<i>E-Prescription Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet 3-The user must be login 4-User must be on at prescription page.</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	Execution description	Procedure result			
1	<i>The user presses the Prescription tab.</i>	<i>Redirect to prescription page which contain list of prescriptions and a form page.</i>			
2	<i>User fill the search form by entering required Prescription details (date, patient email, or time slot).</i>	<i>The system validate the information, start display the prescription and a success message.</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.32 Filter E-Prescription Alternative

<b>Filter E-Prescription Alternative</b>					
<b>Test Case ID:</b>	32	<b>QA Test Engineer:</b>	<i>Umar Farooq</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	15-04-2023	<b>Use Case Reference(s):</b>	<i>User search E-Prescription</i>		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The doctor should not be able to search E-Prescription for patient.</i>				
<b>Product/Ver/Module:</b>	<i>E-Prescription Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet 3-The user must be login 4-The user must be on view prescription page.</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
<b>Step No.</b>	<b>Execution description</b>	<b>Procedure result</b>			
1	<i>The user presses the Prescription tab.</i>	<i>Redirect to prescription page which contain list of prescriptions and a form page.</i>			
2	<i>User fill the search form by entering required Prescription details (date, patient email, or time slot).</i>	<i>The system validate the information, start display with an error message “you do not have any record”</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.33 User Chat List

<b>User Chat List</b>					
<b>Test Case ID:</b>	33	<b>QA Test Engineer:</b>	<i>Umar Farooq</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	15-04-2023	<b>Use Case Reference(s):</b>	<i>User Chat List</i>		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The doctor should able to view Chat List.</i>				
<b>Product/Ver/Module:</b>	<i>Chat Module</i>				
<b>Environment:</b>	<i>Web Brower Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet 3-The user must be login</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	<b>Execution description</b>	<b>Procedure result</b>			
1	<i>The user presses the Chat tab.</i>	<i>Redirect to Chat page which contain list of user.</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.34 Start New Chat

Start New Chat					
<b>Test Case ID:</b>	34	<b>QA Test Engineer:</b>	<i>Umar Farooq</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	15-04-2023	<b>Use Case Reference(s):</b>	<i>Start New Chat</i>		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The doctor should able to start new Chat.</i>				
<b>Product/Ver/Module:</b>	<i>Chat Module</i>				
<b>Environment:</b>	<i>Web Brower Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet 3-The user must be login</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	Execution description	Procedure result			
1	<i>The user presses the Chat tab.</i>	<i>Redirect to Chat page which contain list of user.</i>			
2	<i>User click on the “New” button.</i>	<i>The system displays a pop up and asking for user email</i>			
3	<i>User enter the email of user to start chat.</i>	<i>The system create a new chat room with user with a success message.</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.35 Start New Chat Alternative

Start New Chat Alternative					
Test Case ID:	35	QA Test Engineer:	Umar Farooq		
Test case Version:	1	Reviewed By:	Ummar Ikram		
Test Date:	15-04-2023	Use Case Reference(s):	Start New Chat		
Revision History:	<i>Refer to previous test case identity (if any)</i>				
Objective	<i>The doctor should not be able to start new Chat.</i>				
Product/Ver/Module:	<i>Chat Module</i>				
Environment:	<i>Web Browser Working Internet Connection</i>				
Assumptions:	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet 3-The user must be login</i>				
Pre-Requisite:	<i>None</i>				
Step No.	Execution description		Procedure result		
1	<i>The user presses the Chat tab.</i>		<i>Redirect to Chat page which contain list of user.</i>		
2	<i>User click on the “New” button.</i>		<i>The system displays a pop up and asking for user email</i>		
3	<i>User enter the invalid email of user to start chat.</i>		<i>The system validates and does not create a new chat room with user with an error message.</i>		
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.36 Send Message

Send Message					
<b>Test Case ID:</b>	36	<b>QA Test Engineer:</b>	<i>Umar Farooq</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	15-04-2023	<b>Use Case Reference(s):</b>	<i>Send Message</i>		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The doctor should able to start new Chat.</i>				
<b>Product/Ver/Module:</b>	<i>Chat Module</i>				
<b>Environment:</b>	<i>Web Brower Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet 3-The user must be login</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	<b>Execution description</b>		<b>Procedure result</b>		
1	<i>The user presses the Chat tab.</i>		<i>Redirect to Chat page which contain list of user.</i>		
2	<i>User click on a specific user from a list of users in a chat room.</i>		<i>The system displays the history of messages.</i>		
3	<i>User click on message text box and write text.</i>		<i>The system displays the message written in chat box.</i>		
4	<i>User click on send button.</i>		<i>Display message in chat history</i>		
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.37 Send Message Alternative

Send Message Alternative					
Test Case ID:	37	QA Test Engineer:	Umar Farooq		
Test case Version:	1	Reviewed By:	Ummar Ikram		
Test Date:	15-04-2023	Use Case Reference(s):	Send Message		
Revision History:	<i>Refer to previous test case identity (if any)</i>				
Objective	<i>The doctor should able to start new Chat.</i>				
Product/Ver/Module:	<i>Chat Module</i>				
Environment:	<i>Web Browser Working Internet Connection</i>				
Assumptions:	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet 3-The user must be login</i>				
Pre-Requisite:	<i>None</i>				
Step No.	Execution description		Procedure result		
1	<i>The user presses the Chat tab.</i>		<i>Redirect to Chat page which contain list of user.</i>		
2	<i>User click on a specific user from a list of users in a chat room.</i>		<i>The system displays the history of messages.</i>		
3	<i>User click on back button</i>		<i>Closes the chat box</i>		
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.38 Download Report

<b>Download Report</b>					
<b>Test Case ID:</b>	38	<b>QA Test Engineer:</b>	<i>Umar Farooq</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	15-04-2023	<b>Use Case Reference(s):</b>	<i>Download Report</i>		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The user should able to download Reports for patient.</i>				
<b>Product/Ver/Module:</b>	<i>Report Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet 3-The user must be login</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	<b>Execution description</b>		<b>Procedure result</b>		
1	<i>The user presses the Report tab.</i>		<i>Redirect to Report page which contain list of Reports.</i>		
2	<i>User click on specific report download button</i>		<i>The system validate the information, download the report and displays a success message.</i>		
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.39 Preview Reports

<b>Preview Reports</b>					
<b>Test Case ID:</b>	39	<b>QA Test Engineer:</b>	<i>Umar Farooq</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	15-04-2023	<b>Use Case Reference(s):</b>	<i>User view Reportd</i>		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The user should able to view Reports for patient.</i>				
<b>Product/Ver/Module:</b>	<i>Reports Module</i>				
<b>Environment:</b>	<i>Web Brower Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet 3-The user must be login</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	<b>Execution description</b>		<b>Procedure result</b>		
1	<i>The user presses the Report tab.</i>		<i>Redirect to report page which contain list of reports.</i>		
2	<i>User click on desired "Preview" of report.</i>		<i>The system validate the information, start display the report with a success message.</i>		
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.40 Add Reports

<b>Add Reports</b>					
<b>Test Case ID:</b>	40	<b>QA Test Engineer:</b>	<i>Umar Farooq</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	15-04-2023	<b>Use Case Reference(s):</b>	<i>User Add Reports</i>		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The user should be able to add Reports for patient.</i>				
<b>Product/Ver/Module:</b>	<i>Report Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet 3-The user must be login</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	<b>Execution description</b>		<b>Procedure result</b>		
1	<i>The user presses the Report tab.</i>		<i>Redirect to Report page which contain list of report.</i>		
2	<i>User fill email, test name, feedback (can get from Disease classification model by click on “AI Assist”), Additional details such as property(Ca ,Zn), Value, and Normal and click on ‘mint’ Button.</i>		<i>The system validate the information, add the report and displays a success message.</i>		
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.41 Add Report Alternative

<b>Add Report Alternative</b>					
<b>Test Case ID:</b>	41	<b>QA Test Engineer:</b>	<i>Umar Farooq</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	15-04-2023	<b>Use Case Reference(s):</b>	<i>User Add Report</i>		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The user should not be able to add Report for patient.</i>				
<b>Product/Ver/Module:</b>	<i>Report Module</i>				
<b>Environment:</b>	<i>Web Browser Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet 3-The user must be login</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	<b>Execution description</b>		<b>Procedure result</b>		
1	<i>The user presses the Report tab.</i>		<i>Redirect to Report page which contain list of report.</i>		
2	<i>User fill invalid email, test name, feedback (can get from Disease classification model by click on “AI Assist”), Additional details such as property(Ca ,Zn), Value, and Normal and click on ‘mint’ Button.</i>		<i>The system validate the information, does not add the report and displays an error message.</i>		
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.42 Filter Report

<b>Filter Report</b>					
<b>Test Case ID:</b>	42	<b>QA Test Engineer:</b>	<i>Umar Farooq</i>		
<b>Test case Version:</b>	1	<b>Reviewed By:</b>	<i>Ummar Ikram</i>		
<b>Test Date:</b>	15-04-2023	<b>Use Case Reference(s):</b>	<i>User search Report</i>		
<b>Revision History:</b>	<i>Refer to previous test case identity (if any)</i>				
<b>Objective</b>	<i>The user should able to search report of patient.</i>				
<b>Product/Ver/Module:</b>	<i>Report Module</i>				
<b>Environment:</b>	<i>Web Brower Working Internet Connection</i>				
<b>Assumptions:</b>	<i>1- Hiro Wallet extension is added to the browser. 2- The system is connected to the Internet 3-The user must be login</i>				
<b>Pre-Requisite:</b>	<i>None</i>				
Step No.	<b>Execution description</b>	<b>Procedure result</b>			
1	<i>The user presses the Report tab.</i>	<i>Redirect to report page which contain list of reports.</i>			
2	<i>User fill the search form by entering required report details such as property(Ca ,Zn), Value, and Normal.</i>	<i>The system validate the information, start display the report and a success message.</i>			
<b>Comments:</b> The test case is passed. Our system is working properly.					
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed					

### 6.2.43 Filter Report Alternative

Filter Report Alternative			
Step No.	Execution description	Procedure result	
1	The user presses the Report tab.	Redirect to report page, which contain list of reports.	
2	User fill the search form by entering invalid Prescription details (date, patient email, or time slot).	The system validate the information, start display with an error message "you do not have any record"	
<b>Comments:</b> The test case is passed. Our system is working properly.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

### 6.2.44 Log out

Log out			
Step No.	Execution description	Procedure result	
1	The user presses the Log out tab.	Redirect to log in page and hides private pages.	
<b>Comments:</b> The test case is passed. Our system is working properly.			
<input checked="" type="checkbox"/> Passed <input type="checkbox"/> Failed <input type="checkbox"/> Not Executed			

### 6.3 Test Metrics

The relevant test metrics are presented in the table below:

Metric	Value
<b>Number of Test Cases</b>	44
<b>Number of Test Cases Passed</b>	44
<b>Number of Test Cases Failed</b>	0
<b>Test Case Defect Density</b>	0%
<b>Test Case Effectiveness</b>	0%
<b>Traceability Matrix</b>	Included in separate spreadsheet

### 6.4 Conclusion

The prototype of our system was designed to illustrate the basic features of our system. This chapter highlighted all the implementation details of the prototype. It is a web-based application, which has been implemented in MERN stack. Moreover, it provided all the depth in details of most major functionalities of user wallet authentication with integrating Blockchain and doctor's verification using PMC Api, their communications security using asymmetric cryptography along with the hybrid data storage (Mongo and Gaia) and for NFT's we utilizing dummy wallets to perform functionality. Lastly, the implementation of disease classification model run as a micro service for the assistance of doctors.

Moreover, conducting thorough testing and executing test cases is critical in ensuring that our system is free from bugs and risks that could potentially harm our main business route. By designing and executing comprehensive test cases, we have been able to identify and address any potential issues before they can cause any major disruptions to our business. This approach has not only helped us to protect our main business route but has also improved the overall quality and reliability of our system.

## Chapter 7: User Manual

The following chapter will demonstrate the essential workflows of all the users on the SHIFA platform. It will act as a guide in case of confusion/ambiguity regarding the interaction with the system.

### 7.1 Register

1. Navigate to the home page
2. Click on the “Connect Wallet” button
3. Enter your wallet credentials and allow access.
4. Fill out the necessary information such as email, name, type, etc.
5. Click on the “Register” button

### 7.2 Login

1. Navigate to the home page
2. Click on the “Connect Wallet” button
3. Enter your wallet credentials and allow access.

### 7.3 Book Appointments (Only Patients)

1. Perform Login
2. Navigate to the “Dashboard” page using the sidebar navigation panel.
3. Click on “Appointments” button against the relevant doctor profile card.
4. Choose your desired time slot against the available ones.
5. Click on the “Book” button.
6. Make payment through your wallet.

### 7.4 View Appointments

- Perform Login
- Navigate to the “Appointments” page using the sidebar navigation panel
- If you are a doctor, you can preview your appointments by clicking on the “View Appointments” toggle button.
- If you are a patient, you will see your appointments list by default

### 7.5 Send Message

- Perform Login
- Navigate to the “Messages” page using the sidebar navigation panel
- Select a recipient from the list show on the left or add a new one by clicking on the “New” button
- Type the message in the box below and click on the “Send” button

## 7.6 View Messages

- Perform Login
- Navigate to the “Messages” page using the sidebar navigation panel
- Select a recipient from the list show on the left to preview conversation

## 7.7 Upload Reports (Lab Technician)

- Perform Login
- Navigate to the “Reports” page using the sidebar navigation panel
- Add the patient details along with the report information e.g. property, value
- Click on the “Mint” button
- Make payment through your wallet

## 7.8 View Reports

- Perform Login
- Navigate to the “Reports” page using the sidebar navigation panel
- If you are a doctor, you can preview your reports by clicking on the “View Reports” toggle button.
- If you are a patient, you will see your reports list by default

## 7.9 Upload Prescriptions (Doctor)

- Perform Login
- Navigate to the “Prescriptions” page using the sidebar navigation panel
- Add the patient details along with the prescription information e.g., salt, frequency
- Click on the “Mint” button
- Make payment through your wallet

## 7.10 View Prescriptions

- Perform Login
- Navigate to the “Prescriptions” page using the sidebar navigation panel
- If you are a doctor, you can preview your prescriptions by clicking on the “View Prescriptions” toggle button.
- If you are a patient, you will see your prescriptions list by default

## 7.11 Troubleshoot

If any issue occurs using the SHIFA web application, some troubleshooting tips are listed below

- **Check your internet connection:** Make ensure it is steady and that you are not having any connectivity problems.

- **Clear your cache and cookies:** Frequently, problems with website speed may be fixed by clearing the cache and cookies in your browser.
- **Try a different browser:** If you're experiencing issues with one browser, try using a different one to see if that resolves the issue.
- **Disable browser extensions:** Some browser extensions can interfere with website functionality. Try disabling any extensions you have installed to see if that resolves the issue.
- **Check for updates:** Make sure your browser and operating system are up to date. Outdated software can sometimes cause issues with website functionality.
- **Contact support:** If you've tried these troubleshooting steps and are still experiencing issues, contact support@shifa.com for further assistance. Our team will be happy to help you resolve the issue.

## 7.12 Conclusion

In conclusion, the SHIFA platform is a user-friendly and efficient system that helps connect patients, doctors, lab technicians, and pharmacists in a seamless manner. This user manual chapter serves as a comprehensive guide for all users to navigate through the different workflows such as registration, appointment booking, messaging, report and prescription upload and viewing. Additionally, the troubleshooting section provides valuable tips for resolving any issues that may arise while using the SHIFA web application. By following this manual, users can make the most of the platform's features and benefit from its integrated healthcare services.

## Chapter 8: Conclusion and Work Completed

The chapter lists down the findings and challenges faced during the implementation of the modules followed by recommendations for improvement in the architecture. At last, it summarizes the overall document by briefly discussing the work done so far.

### 8.1 Completed Work

All the implementation of system services like face/speech recognition which involves the doctor authentication. Moreover, core components like messaging via sockets, and appointment system integrated with escrow-based payments development is completed. Furthermore, report/prescription creation with secure dynamic routes will be incorporated into the application. Lastly, after the implementation of the core modules of the system, we define test cases based on use case scenarios and test all the functionality of the working system.

### 8.2 Findings/Results

The disease classification model training produced feasible results based on the data set taken from Kaggle. We are able to get an accuracy of 70-75% for the three models; diabetes, liver, and heart. Upon inspecting the Pakistan Medical Council (PMC) Endpoint payload, one of a crucial parameter (Gender) about the doctor was not properly stored on their database; resolves to null for every doctor. Furthermore, our analysis showed that because Stacks is its own layer-1 blockchain, whose blocks are anchored to Bitcoin's, the time to mine a Stacks block matches the expected block time of Bitcoin, which is on average 10 minutes.

### 8.3 Challenges

The disease classification model data set was hard to find and had to be cleaned, normalized before training. Additionally, the Pakistan Medical Council (PMC) API had to be retrieved by static code analysis of their website. Moreover, for instantiating a local blockchain node, we had to configure a Docker container that would run locally and simulate an actual blockchain. Furthermore, key-sharing in symmetric cryptography is a huge challenge, and there is an extra storage and computation overhead when using asymmetric encryption so it requires further architectural improvements.

### 8.4 Recommendations

Use of a One-time Password (OTP) instead of symmetric encryption when displaying prescriptions/reports via dynamic route seems more promising since it adds additional security and eliminates password setting hassle for the patient. The OTP can be sent to the patient's verified email address and can get automatically expired after a certain time/event. Moreover, additional fungible tokens can be taken on-board as a method of payment for appointments. It can be achieved using traits/interfaces that allows different smart contracts to work together. Lastly, patients can opt for sharing their reports anonymously in return for reward tokens. This will allow us to further refine our disease classification models.

### 8.5 Conclusion

We are developing a decentralized web application for patients and doctors that allows secure communication through a convenient interface. There are separate dashboards for patient and doctor where each user can preview reports/prescriptions, chat and manage appointments. Initially the abstract, introduction, scope and goals provide the overall details about the functionality a user can perform along with the main purpose/aim of developing the application.

Additionally, the sequence, class/model, and architectural diagrams make the workflow explicit and simple so that the project's functionality and methodology can be understood. These diagrams provide a complete representation of the system architecture and the core components that make the system. They also play a crucial role in the development of a prototype that illustrates how the user interacts with the application. Furthermore, a hybrid data-storage infrastructure will be used where sensitive data is stored on the Stacks blockchain, personal data on a private decentralized storage and all other data will be securely stored on MongoDB centralized storage. Moreover, all system constraints along with any hardware/software requirements is also mentioned in detail. Existing products and other cited research material are also documented. The Graphical User Interface (GUI) of the application was designed on Figma and will be developed using NextJS with Vercel as our Content Delivery Network (CDN). In next development phase, we will develop a working prototype representing a subset of our application and its functionality.

## References

- [1] "Electronic prescribing" [Online]. Available: [https://en.wikipedia.org/wiki/Electronic\\_prescribing](https://en.wikipedia.org/wiki/Electronic_prescribing). [accessed 8 October , 2022].
- [2] F. Jiang, Y. Jiang, H. Zhi, Y. Dong "Artificial intelligence in healthcare: past, present and future" *BMJ Journals*, Vol. 2, Issue 4, [Online]. Available: <https://svn.bmjjournals.org/content/2/4/230> [accessed Dec. 2017].
- [3] S. Daley, "Blockchain in Healthcare: 17 Examples to Know." [Online]. Available: <https://builtin.com/blockchain/blockchain-healthcare-applications-companies> [accessed 8 October 2022].
- [4] A. Jiang, "Revolutionizing Health with AI Voice Assistants" [Online]. Available: <https://towardsdatascience.com/how-ai-voice-assistants-can-revolutionize-health-b08d925bd018> [accessed 4 October 2022].
- [5] L. Lovett, "10 digital health companies using cryptocurrency as incentives" [Online]. Available: <https://www.mobihealthnews.com/content/10-digital-health-companies-using-cryptocurrency-incentives> [accessed 4 October 2022].
- [6] S. Muchmore, "Healthcare remains costliest industry for data breaches" [Online]. Available: <https://www.healthcaredive.com/news/healthcare-breach-costs/628344/> [accessed October 4, 2022].
- [7] "Legality of cryptocurrency by country or territory" [Online]. Available: [https://en.wikipedia.org/wiki/Legality\\_of\\_cryptocurrency\\_by\\_country\\_or\\_territory](https://en.wikipedia.org/wiki/Legality_of_cryptocurrency_by_country_or_territory) [accessed 1 October 2022].
- [8] A. Haleem, M. Javaid, R. Pratap, R. Suman, S. Rab, "Blockchain technology applications in healthcare: An overview" *International Journal of Intelligent Networks*, Vol. 2, pp. 130-139, [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S266660302100021X> [Accessed Sept 2021].
- [9] "What is digital health technology and what can it do for me?", *National Institute of Health and Care Research*, [Online]. Available: <https://www.techtarget.com/searchhealthit/definition/digital-health-digital-healthcare#:~:text=According%20to%20the%20FDA%2C%20digital,medicine%20more%20personalized%20for%20patients>. [Accessed 27 Sept. 2022]
- [10] "Paper Waste: Why Does It Matter?" [Online]. Available: <https://baleforce.com/paper-waste-why-does-it-matter-and-how-to-manage-it> [accessed 1 October 2022].
- [11] V. Jaiman, V. Urovi, "User incentives for blockchain-based data sharing platforms" *National Library of Medicine*, [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9009633/> [Accessed April 2022].
- [12] E. Scheuer, "Blockchain solves healthcare data obstacles" *Unblock the Chain*, Vol. 19 -Issue [Online]. Available: <https://healthmanagement.org/c/healthmanagement/issuearticle/blockchain-solves-healthcare-data-obstacles> [Accessed 1, 2019]
- [13] M. Abdullah, F. Mukhtar, S. Wazir, I. Gilani, Z. Gorar and B. Shaikh, "The Health Workforce Crisis in Pakistan: A Critical Review and the Way Forward" *World Health & Population*, [Online]. Available: <https://europepmc.org/article/med/25576749> [Accessed June 2014]
- [14] A. Ghorbani, "Demand for Health and Healthcare" *Healthcare Access*, [Online]. Available: <https://www.intechopen.com/chapters/77622> [Accessed July 2021].
- [15] "Unemployed doctors", *The Express Tribune*, [Online]. Available: <https://tribune.com.pk/letter/2253189/unemployed-doctors> [Accessed 10 November 2021].

- [16] I. Field, "Why Is Health Care Regulation So Complex?" *National Library of Medicine*, Vol. 33 pp. 607-608, [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2730786/> [Accessed October 2008].
- [17] F. Randy, C. Isaacson, M. Pursel, "Personalized Medicine" *National Library of Medicine*, Vol. 35 pp. 560-562, [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/21037908/> [Accessed October 2010].
- [18] G. Rauch, "NextJS", Web Software, [Online]. Available: <https://nextjs.org>, [Accessed October 2016].
- [19] P. Ryan, Eliot Horowitz, Dwight Merriman, "MongoDB", Web Software, [Online]. Available: <https://www.mongodb.com>, [Accessed October 2007].
- [20] J. Lindenbaum, A. Wiggins, O. Henry, "Heroku", Web Software, [Online]. Available: <https://www.heroku.com>, [Accessed June 2007].
- [21] A. Le, "buoyhealth", Web Software, [Online]. Available: <https://www.buoyhealth.com>, [Accessed Feb 2014].
- [22] F. Ricotta, "BurstIQ", Web Software, [Online]. Available: <https://burstiq.com>, [Accessed April 2015].
- [23] A. Albeyatti, "MedicalChain," Web Software, [Online]. Available: <https://medicalchain.com>, [Accessed Oct, 2016].
- [24] E. Imam, "Marham," Mobile/Web Software, [Online]. Available: <https://www.marham.pk> [Accessed Jan, 2015].