# E-MEDIX

| | |
|---|---|
| **Submitted by** | **Asad Masood Chaudhry**<br>**Hafiz Abdul Sattar Butt** |
| **Roll. No** | **0224-BSCS-17**<br>**0249-BSCS-17** |
| **Session** | **2017 - 2021** |
| **Supervised by** | **Mr. Yahya Khuram**<br>**(Assistant Professor)** |


**BS(HONS)**
**IN**
**COMPUTER SCIENCE**


## DEPARTMENT OF COMPUTER SCIENCE
## GC UNIVERSITY LAHORE

# E-MEDIX

**Submitted to GC University Lahore in partial fulfillment of the requirements for the award of degree of**

## BS(HONS)
## IN
## COMPUTER SCIENCE


| | |
|---|---|
| **Submitted by** | **Asad Masood Chaudhry**<br>**Hafiz Abdul Sattar Butt** |
| **Roll. No** | **0224-BSCS-17**<br>**0249-BSCS-17** |
| **Session** | **2017 - 2021** |
| **Supervised by** | **Mr. Yahya Khuram**<br>**(Assistant Professor)** |



## DEPARTMENT OF COMPUTER SCIENCE

## GC UNIVERSITY LAHORE

# Declaration

We, Asad Masood Chaudhry and Hafiz Abdul Sattar Butt students of **BS(Hons)** in the subject of **Computer Science** session **2017-2021**, hereby declare that the matter printed in this Project titled, **E-Medix** is our own work and has not been printed, published and submitted as research work, thesis or publication in any form in any University, Research Institution etc in Pakistan or abroad.

**Date:** _____ **Signatures of Deponents**

# Research Completion Certificate

It is certified that the research work contained in this thesis titled **E-MEDIX** has been carried out by **Asad Masood Chaudhry and Hafiz Abdul Sattar Butt** Roll. Nos **0224-BSCS-17 and 0249-BSCS-17** under my supervision.

**Mr. Yahya Khurram**
**Assistant Professor**

**Date:** _____

**Submitted Through**

**Prof. Dr. Syed Asad Raza Kazmi**
**Chairperson**
**Department of Computer Science**
**GC University Lahore**

**Controller of Examination**
**GC University Lahore**

# Acknowledgements

We are grateful to the Almighty **Allah**, Who blessed us with health, wisdom, knowledge, thoughts and opportunity to make some contribution in the form of present effort. We offer our humblest thanks from the deepest core of our hearts to the **Holy Prophet Muhammad (Peace be upon him)**, the most perfect and excelled among and ever born on the surface of earth.

The research work embodied in this dissertation was accomplished under the able guidance and affectionate supervision of **Mr. Yahya Khurram**, Assistant Professor, GC University, Lahore. we will always remember his moral encouragement, skillful guidance, positive criticism and valuable advice throughout the course of our study.

# *Dedication*

*Our work is dedicated to our families and our honorable teacher Mr. Yahya Khurram who encouraged and helped us to complete our Project E-Medix.*

# Contents

**Declaration** i

**Research Completion Certificate** ii

**Acknowledgements** iii

**Dedication** iv

**Contents** v

**List of Tables** ix

**List of Figures** xi

**Abstract** 1

**1 Introduction** 2
- 1.1 Introduction . . . 2
  - 1.1.1 Problem Area . . . 3
- 1.2 Aims of the Project . . . 4
- 1.3 Project Scope . . . 4

**2 Requirement Specification** 6
- 2.1 Functional Requirements . . . 6
  - 2.1.1 Register . . . 6
  - 2.1.2 Login . . . 7
  - 2.1.3 Update Information . . . 7
  - 2.1.4 Add Doctor . . . 7
  - 2.1.5 Update Doctors . . . 7
  - 2.1.6 Delete Doctors . . . 8
  - 2.1.7 Manage Patients . . . 8
  - 2.1.8 Manage Appointments . . . 8

2.1.9 View Patient History . . . 8
2.1.10 Prescribe Patient . . . 8
2.1.11 Diagnose Patient Remotely . . . 9
2.1.12 Book Appointment . . . 9
2.1.13 View Appointment History . . . 9
2.1.14 Download Prescription . . . 9
2.1.15 Upload Medical Record . . . 9
2.1.16 Get Help . . . 9
2.2 Non-functional Requirements . . . 10
2.2.1 Security . . . 10
2.2.2 Performance . . . 10
2.2.3 Availability . . . 10
2.2.4 Usability . . . 11
2.2.5 Flexibility And Scalability . . . 11
2.2.6 Maintainability . . . 11
2.2.7 Modification . . . 11

**3 Project Design 13**
3.1 Methodology . . . 13
3.2 Architecture Overview . . . 14
3.2.1 Models . . . 15
3.2.2 Views . . . 15
3.2.3 Templates . . . 15
3.3 Design Description . . . 16
3.3.1 Use Case Diagram: . . . 17
3.3.1.1 Use Cases: . . . 18
3.3.2 Sequence Diagrams: . . . 32
3.3.3 Class Diagram: . . . 41
3.3.4 Data Flow Diagrams: . . . 42
3.3.5 Entity Relationship Diagram: . . . 43

**4 Implementation and Evaluation 44**
4.1 Development Stages . . . 44
4.1.1 Stage 1(Requirements Gathering and Analysis): . . . 44
4.1.2 Stage 2(Design): . . . 45
4.1.3 Stage 3(Development): . . . 45
4.1.4 Stage 4(Testing): . . . 46
4.1.5 Stage 5(Implementation): . . . 46
4.1.6 Stage 6(Maintenance): . . . 46
4.2 System Integration . . . 47
4.3 User Interface . . . 47
4.3.1 Main page . . . 48
4.3.2 Sign up . . . 49
4.3.3 Login . . . 49

4.3.4 Patient Details . . . 50
4.3.5 Doctor Details . . . 50
4.3.6 Clinic Details . . . 51
4.3.7 Two Factor Verification . . . 51
4.3.8 OTP Email . . . 52
4.3.9 Patient Dashboard . . . 52
4.3.10 Book Appointment . . . 53
4.3.11 Payment . . . 53
4.3.12 Payment Success . . . 54
4.3.13 Payment Error . . . 54
4.3.14 Confirmation Email Patient . . . 55
4.3.15 Confirmation Email Doctor . . . 55
4.3.16 Doctor Dashboard . . . 56
4.3.17 Prescribe Patient . . . 56
4.3.18 View Prescription . . . 57
4.3.19 Clinic Dashboard . . . 57
4.3.20 Update Patient Details . . . 58
4.3.21 Update Doctor Details . . . 58
4.3.22 Update Clinic Details . . . 59
4.3.23 Change Password . . . 59
4.3.24 Upload Medical Records . . . 60
4.3.25 View Medical Records . . . 60
4.3.26 Video Consultancy . . . 61
4.4 Evaluation . . . 62
4.5 Unit Testing . . . 63
4.6 Unit Tests . . . 64
4.6.1 TC-01 Register . . . 65
4.6.2 TC-02 Login . . . 66
4.6.3 TC-03 Two Factor Verification . . . 67
4.6.4 TC-04 Add Patient Details . . . 68
4.6.5 TC-05 Add Doctor Details . . . 69
4.6.6 TC-06 Add Clinic Details . . . 70
4.6.7 TC-07 Book Appointment . . . 71
4.6.8 TC-08 View Past Appointments . . . 72
4.6.9 TC-09 Order Medicines . . . 73
4.6.10 TC-10 Print Prescription . . . 74
4.6.11 TC-11 View Past Prescriptions . . . 75
4.6.12 TC-12 Consult Patient . . . 76
4.6.13 TC-13 Notify Patient . . . 77
4.6.14 TC-14 Prescribe Patient . . . 78
4.6.15 TC-15 Analyze Earnings . . . 79
4.6.16 TC-16 View Doctors . . . 80
4.6.17 TC-17 Change Password . . . 81
4.6.18 TC-18 Update Profile . . . 82

4.6.19 TC-19 Upload Medical Record . . . . . 83
4.6.20 TC-20 View Patient History . . . . . 84
4.7 Functional Testing . . . . . 85
4.7.1 Testing Requirements . . . . . 85

**5 Conclusion & Future Work** **86**
5.1 Conclusion . . . . . 86

**Bibliography** **88**

# List of Tables


3.1 UC-01 Register . . . 18
3.2 UC-02 Login . . . 19
3.3 UC-03 Update Information . . . 20
3.4 UC-4 Book Appointment . . . 21
3.5 UC-05 View Appointment history . . . 22
3.6 UC-06 Download Prescription . . . 23
3.7 UC-07 Update Medical Record . . . 24
3.8 UC-08 Get help from chatbot . . . 25
3.9 UC-09 View Patient history . . . 26
3.10 UC-10 Prescribe Patient . . . 27
3.11 UC-11 Online Diagnosis . . . 28
3.12 UC-12 Manage Appointments . . . 29
3.13 UC-13 Manage Doctor . . . 30
3.14 UC-14 Manage Patients . . . 31

4.1 Test Cases . . . 64
4.2 TC-01 Register . . . 65
4.3 TC-02 Login . . . 66
4.4 TC-03 Two Factor Verification . . . 67
4.5 TC-04 Add Patient Details . . . 68
4.6 TC-05 Add Doctor Details . . . 69
4.7 TC-06 Add Clinic Details . . . 70
4.8 TC-07 Book Appointment . . . 71
4.9 TC-08 View Past Appointments . . . 72
4.10 TC-09 Order Medicines . . . 73
4.11 TC-10 Print Prescription . . . 74
4.12 TC-11 View Past Prescriptions . . . 75
4.13 TC-12 Consult Patient . . . 76
4.14 TC-13 Notify Patient . . . 77
4.15 TC-14 Prescribe Patient . . . 78
4.16 TC-15 Analyze Earnings . . . 79
4.17 TC-16 View Doctors . . . 80
4.18 TC-17 Change Password . . . 81
4.19 TC-18 Update Profile . . . 82
4.20 TC-19 Upload Medical Record . . . 83

4.21 TC-20 View Patient History . . . . . 84

# List of Figures


3.1 The Waterfall Software Development Lift Cycle . . . . . 14
3.2 Django - Model View Template Architecture . . . . . 15
3.3 Use Case Diagram . . . . . 17
3.4 SD UC-01 Register . . . . . 32
3.5 SD UC-02 Login . . . . . 32
3.6 SD UC-03 Update Information . . . . . 33
3.7 SD UC-04 Book Appointment . . . . . 34
3.8 SD UC-05 View Appointment History . . . . . 34
3.9 SD UC-06 Download Prescription . . . . . 35
3.10 SD UC-07 Upload Medical Reports . . . . . 35
3.11 SD UC-08 Get help from chatbot . . . . . 36
3.12 SD UC-09 View Patient History . . . . . 36
3.13 SD UC-10 Prescribe Patient . . . . . 37
3.14 SD UC-11 Online Diagnosis . . . . . 38
3.15 SD UC-12 Manage Appointments . . . . . 39
3.16 SD UC-13 Manage Doctors . . . . . 40
3.17 SD UC-14 Manage Patients . . . . . 40
3.18 UML Class Diagram . . . . . 41
3.19 Level 0 - Data Flow Diagram . . . . . 42
3.20 Level 1 - Data Flow Diagram . . . . . 42
3.21 Entity Relationship Diagram . . . . . 43

4.1 Main Page . . . . . 48
4.2 Sign up . . . . . 49
4.3 Login . . . . . 49
4.4 Patient Dashboard . . . . . 50
4.5 Doctor Dashboard . . . . . 50
4.6 Clinic Dashboard . . . . . 51
4.7 Two Factor Verification . . . . . 51
4.8 OTP Email . . . . . 52
4.9 Patient Dashboard . . . . . 52
4.10 Book Appointment . . . . . 53
4.11 Payment . . . . . 53
4.12 Payment Success . . . . . 54
4.13 Payment Error . . . . . 54

4.14 Confirmation Email Patient . . . . . 55
4.15 Confirmation Email Doctor . . . . . 55
4.16 Doctor Dashboard . . . . . 56
4.17 Prescribe Patient . . . . . 56
4.18 View Prescription . . . . . 57
4.19 Clinic Dashboard . . . . . 57
4.20 Update Patient Details . . . . . 58
4.21 Update Doctor Details . . . . . 58
4.22 Update Clinic Details . . . . . 59
4.23 Change Password . . . . . 59
4.24 Upload Medical Records . . . . . 60
4.25 View Medical Records . . . . . 60
4.26 Video Consultancy . . . . . 61

# *Abstract*

Since the outbreak of COVID-19 on December 1, 2019, we have become aware of the significant gap between people and proper healthcare facilities. The unavailability of healthcare facilities is one of the most salient factors for the increased death rate in Pakistan. Our primary focus is to bridge this gap with the help of our Web Application. E-Medix is a platform designed and developed to be used by Patients, Doctors and Clinics.

Our web application allows a patient to upload his medical history for the doctor's ease. Before consulting the patient, the doctor can study the patient's medical history and plan their treatment accordingly. E-Medix also allows a patient to manage all their past and upcoming appointments and view the prescriptions of any past appointment at any time. Patients can avoid wasting their time using E-Medix.They no longer have to travel for long durations just to travel to the clinics because they can receive online consultancy through a video call from anywhere around the world.

Using our platform doctors can manage their appointments and keep track of their overall and monthly income. The doctors may work remotely at the hours and place of their own convenience through our platform. Our platform also helps clinics to keep track of the monthly and overall revenue generated by the doctors.

Our platform promotes the culture of working remotely and adaptability of technology in our daily lives. We aim to make healthcare easily accessible anywhere anytime.

# Chapter 1

# Introduction

## 1.1 Introduction

Since the outbreak of COVID-19 on December 1, 2019, we have become aware of the significant gap between people and proper healthcare facilities. According to the WHO(World Health Organisation) [11], 2,709,041 people have lost their lives due to this pandemic. Pakistan alone has 13,848 deaths caused due to COVID-19 numbers can be anticipated to go higher. The deaths caused could have been avoided with proper healthcare facilities in the country. As a developing country, Pakistan has numerous battles to fight. Due to which the healthcare system of Pakistan has suffered a lot. When it comes to the quality and accessibility of healthcare facilities, Pakistan does not do very well. As stated on Wikipedia[12], Pakistan is ranked 122nd out of 190 in the World Health Organization performance report. According to another study conducted by the Lancet, Pakistan is ranked 154th out of 195 countries. The unavailability of healthcare facilities is one of the most salient factors for the increased death rate in Pakistan [3]. Our primary focus is to bridge the existing gap with the help of our Web Application. E-Medix is a generic web-based system that will facilitate patients, doctors, and clinics by making healthcare accessible anytime, anywhere. Our platform will be a complete platform where patients can maintain their medical reports, book doctors, and

order medicine. Our platform will also serve the doctors by allowing them to manage their clinics online. The doctor can view the patient's medical history. The doctor can also provide consultation to the patient through a video call using E-Medix. We will also include a chat bot to help the patients connect with the best doctors and cater to their general queries. We hope our product will play a vital role in the betterment of the healthcare system of Pakistan. We aim to make healthcare easily accessible at all times.

### 1.1.1 Problem Area

When a patient comes to a new doctor to get diagnosed, they may not have all their medical files. It makes it difficult for the doctor to analyze their problem accurately and suggest treatment accordingly. In developing countries like Pakistan, specialist doctors are scarce.[5]. Most of the medical specialists of Pakistan are working in urban areas resulting in a shortage of specialists in rural areas. As a result, the people living in rural areas do not have quick access to quality healthcare services.[7]. It is quite problematic for them to get access to basic needs like quality healthcare services. They have to travel a distance to get to urban areas for treatments resulting in many deaths occurring on the way. We can avoid such mishaps if quality healthcare services are readily available.[4] Today we are surrounded by technology. Humans are using technology to bring comfort to their lives. In this fast era, we find ways to save our time and are dedicated to doing more in less time. It will be much more comfortable and time-saving to book appointments and consult your doctor with the click of a button rather than going to a clinic.[2] Another observation is that many people are very skeptical when it comes to healthcare. They feel comfortable getting the consultation from their regular doctor instead of someone new. Now let us suppose that they are traveling and fall sick? With a platform like ours, they will be able to consult their regular doctor from miles away instead of going to a new local doctor.

## 1.2 Aims of the Project

The aims of this project are to:

- Manage patient's medical record
- To facilitate video conferencing between health care experts and the patients for better treatment and care.
- Online issuing of prescriptions to facilitate both the doctor and patient.
- Provide healthcare to remote locations.
- Help Doctors and clinics manage appointments.
- Help Doctors and clinics keep track of their earnings.
- Making Pharmacy, Laboratory and Doctor accessible at one platform.

## 1.3 Project Scope

The Scope of the project to make healthcare accessible anytime, anywhere. The patient can book appointments with the selected doctor from my platform and also receive online video consultation from anywhere they are at any time. They can also order the medicines and book clinical tests prescribed by the doctor from the same platform. The patients can chat with a chat bot to connect to the best doctor as per their requirements. The chat bot will also cater to the general queries of the patients. The patients can upload their old prescriptions and lab test reports to the platform. The medical records generated will automatically be added to the patient record. This platform will manage the patient health record for the ease of both the patient and the doctor. This app will also have an emergency portal so the patient can consult the emergency cases with the on-duty doctors. The doctors can use this app to keep track of their appointments and reschedule according to their timetable. The doctors can keep track of his earnings through this platform

instead of hiring an assistant. Moreover, the doctor can also view the patient's medical history before the appointment to diagnose the problem accurately and provide better healthcare.

# Chapter 2

# Requirement Specification

## 2.1 Functional Requirements

A Functional Requirement describes what the system has to offer,it explains what a system consists of and what does it's components do.

### 2.1.1 Register

Registering to the platform is the first step.

- This function will register to the platform.
- The user will have to choose one from Patient, Doctor or Clinic.
- The user will then enter the required information in order to proceed with the registration process. The information required will be slightly different for the categories Patient, Doctor and Clinic.
- The user will then click the register button that will redirect them to the home page if their registration is successful and display an error message if their is any error.

### 2.1.2 Login

- This function will login the user if they are already registered.
- The user will specify if they are attempting to login as a patient, doctor or clinic.
- The user will then enter their specified credentials.
- The credentials will be compared to the credentials stored in the database.
- If a match is found the user will be redirected to their concerned dashboard.
- If no match is found an error message is displayed to the user.

### 2.1.3 Update Information

- This function will allow the user update their basic information as per their requirements.
- The user will go to their profile where they can change their information except for a few fixed fields.

### 2.1.4 Add Doctor

- This Function will allow a clinic to add a doctor to their clinic.
- This function will only be available to clinic owners.

### 2.1.5 Update Doctors

- This Function will allow a clinic to update the information of a doctor who is registered under their clinic.
- This function will only be available to clinic owners.

### 2.1.6 Delete Doctors

- This Function will allow a clinic to delete doctors from their existing list of doctors at any time.
- This function will only be available to clinic owners.

### 2.1.7 Manage Patients

- This Function will allow a clinic to add,update or delete patients in their list of patients.
- This function will only be available to clinic owners.

### 2.1.8 Manage Appointments

- This Function will allow a doctor to manage their appointment.
- The doctor can accept the appointment.
- The doctor can reschedule the appointment.
- The doctor can send reminder to the patient.

### 2.1.9 View Patient History

- This Function will allow doctor to preview the medical history of the patient who has scheduled an appointment with them.

### 2.1.10 Prescribe Patient

- This Function will allow a doctor to write a prescription and send it to the patient's dashboard.

### 2.1.11 Diagnose Patient Remotely

- This Function will allow a doctor to generate a link for video call and send it to the patient for online diagnosis.
- This function will only operate if the patient has booked an online appointment.

### 2.1.12 Book Appointment

- This Function will allow a patient to book an appointment with the doctor of their choice.

### 2.1.13 View Appointment History

- This Function will allow a patient to view their past appointments.

### 2.1.14 Download Prescription

- This Function will allow a patient to download the prescription given by the doctor.

### 2.1.15 Upload Medical Record

- This Function will allow a patient to upload their past medical records

### 2.1.16 Get Help

- This Function will allow a patient to get help from a chat bot.
- The chat bot will cater to the general queries of a patient.

## 2.2 Non-functional Requirements

A Non Functional Requirement focused on how the system should perform it's functionalities rather than what functionalities it should perform.

### 2.2.1 Security

- Users should register into the system to perform any operation.
- Only login users are allowed to interact with the system.
- All important operations are handled by a server (administrator) only.
- The app must ensure that no one can access or bypass the secured information or data.
- The personal information of any person or organization must be kept safe.

### 2.2.2 Performance

- The application should work well with other interactive systems.
- All the features of the application must ensure that work is done on time and the app performs well on different devices.
- Data integrity must be maintained for good response of time.
- System must give access guidelines.
- System must response in seconds.

### 2.2.3 Availability

- When an action is called for at an unknown, i.e. random, time, the degree to which a device, subsystem, or piece of equipment is in a given operable and commit able state at the start of the mission must be available.

- If any system failure or error occurs, system broadcast message to all user.

### 2.2.4 Usability

- The device should be simple to use, or user friendly.
- In general, usability is a central tenant in social interaction, and it is concerned with making programs simple to understand and use, as well as reducing the frequency and severity of errors.

### 2.2.5 Flexibility And Scalability

- Flexibility is a personality trait that defines a person's ability to adapt to changing situations and think of problems and tasks and imaginative ways. System must provide scalability in the future for increased number of users.
- Software scalability is a feature of a tool or device that allows it to expand its capacity and functionality in response to user demand. Scalable software can adjust to updates, improvements, overhauls, and resource reductions while remaining stable.

### 2.2.6 Maintainability

- The system should be easy to maintain.
- Maintainability is a measure of an item's ability to be kept in or returned to a given condition when maintained by staff with a specific skill set.
- The System shall provide ease for updating system with new features.

### 2.2.7 Modification

- Must ensure its modifying anytime without affecting other modules(functions).

- A software product requires maintenance in order to accommodate new functionality requested by customers or to modify various types of device functionalities in response to customer demands.
- Modifying keeps all features from previous update.

# Chapter 3

# Project Design

## 3.1 Methodology

Software development life cycle is a process used during the development process of any software. SDLC provides us with a detailed explanation of how we can develop, maintain, replace or enhance the software we are developing. SDLC plays a vital role in ensuring the quality of the software. It also makes sure of a smooth flow of the overall development process.
Some of the most famous SDLC models used are as under:

- Waterfall Model
- Iterative Model
- Spiral Model
- V-Model
- Big Bang Model

We will be using the Waterfall Software Development life cycle for this project. The waterfall model is the most simple and easy-to-use SDLC model. The waterfall model is a step-wise software development process. In the waterfall model,

each phase must be completely executed before proceeding to the next phase. The waterfall model is most suitable for our project because our project has no uncertain requirements. All the requirements of the project are clear, fixed, well defined, and documented. The technology to be used is understood and static and there are no ambiguous requirements.[10]

The phases of the Waterfall Software Development Lift Cycle are as follows[6]:

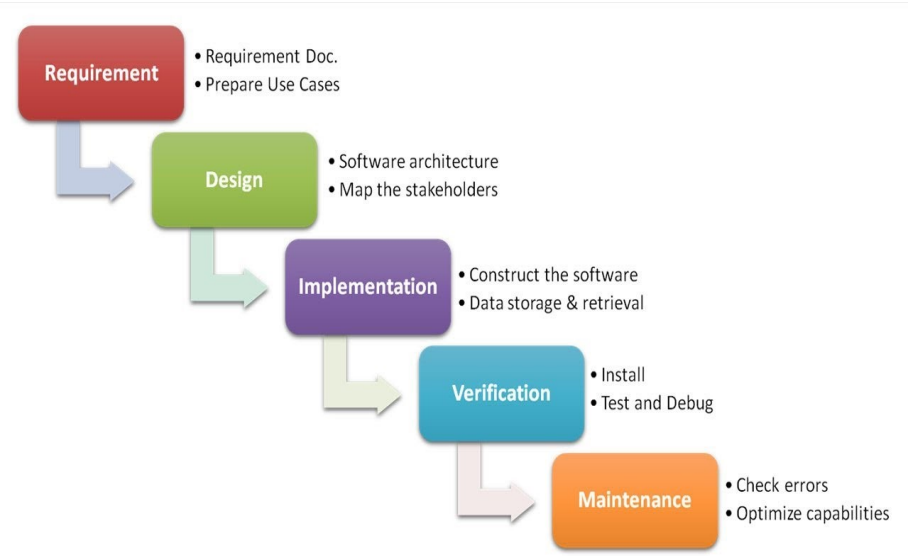


FIGURE 3.1: The Waterfall Software Development Lift Cycle

## 3.2 Architecture Overview

We will be using the Django framework during the development process of this project. Django is a high-level open-source Python web framework. Many famous web applications like Instagram got developed using the Django Framework.[1] The Django framework implies an MVT architecture for the development of web applications. The MVT architecture consists of three major components.

1. Models
2. Views
3. Templates

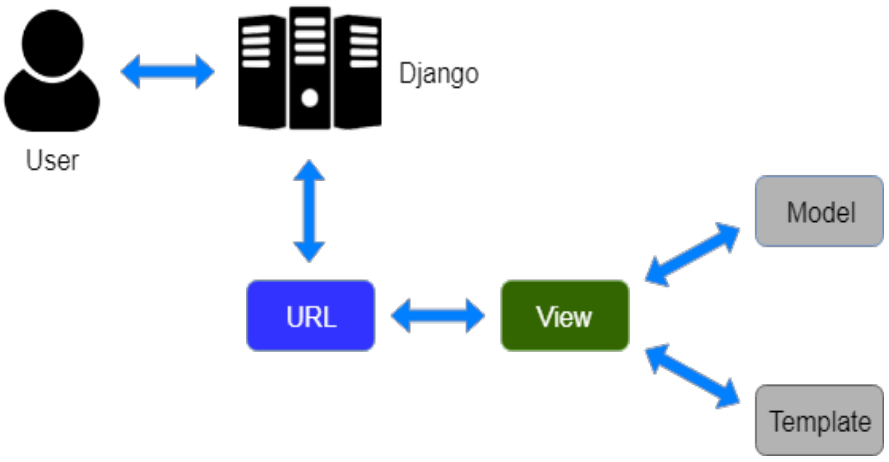


FIGURE 3.2: Django - Model View Template Architecture

### 3.2.1 Models

The model works as the interface of the data on the web application. The model is the component where we implement all our logic for manipulating the data in the database. Model is used to maintain the data used in our application. It is responsible for retrieving, inserting, updating, and deleting data from the database.

### 3.2.2 Views

The Views in Django form the user interface of the web applications. The views folder stores all the Front end files of the web application (HTML, CSS, Javascript, etc.).

### 3.2.3 Templates

Templates are an important part of the MVT architecture. Django Templates are used to provide a layout to our web application. A template comprises the HTML of static parts of the user interface and a special syntax explaining how the dynamic parts will be included.

## 3.3 Design Description

The diagrams included in our documentation are:

- Use Case Diagrams
- Sequence Diagrams
- Class Diagram
- Data Flow Diagrams
- Entity Relationship Diagram

### 3.3.1 Use Case Diagram:

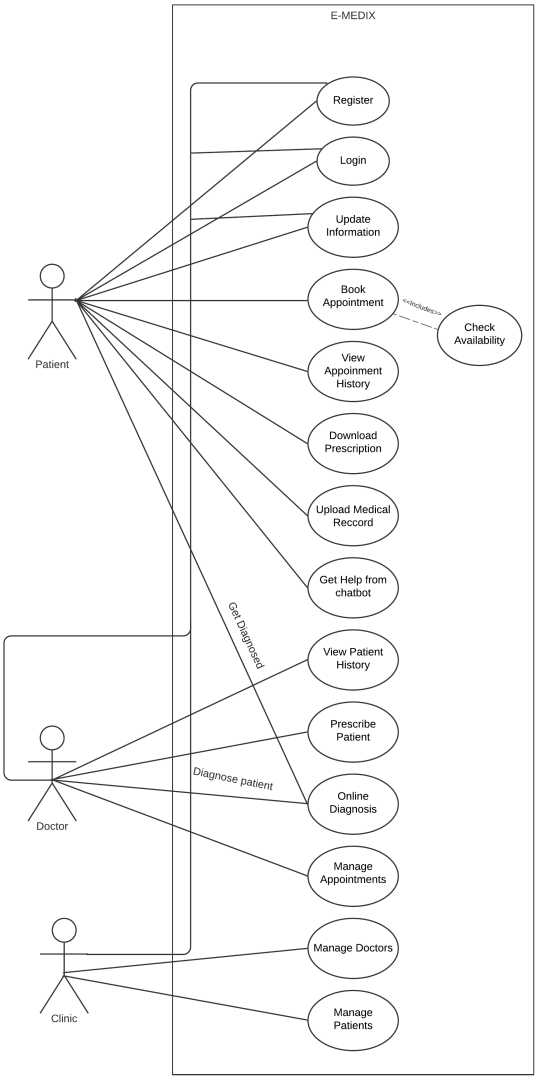


FIGURE 3.3: Use Case Diagram

#### 3.3.1.1 Use Cases:

Table 3.1: UC-01 Register

| | |
|---|---|
| **ID** | UC-01 |
| **Name** | Register |
| **Description** | This use case will describe how a user registers on our web application. |
| **Actors** | Patient, Doctor or Clinic |
| **Precondition** | User must be connected to internet |
| **Scenario** | • User will select the type of registration i.e Patient, Doctor or Clinic.<br>• User will enter their information.<br>• User will press the register button. |
| **Expectations** | User might get registered to the web application. |
| **Post Conditions** | User will get registered to the web application. |

Table 3.2: UC-02 Login

| | |
|---|---|
| **ID** | UC-02 |
| **Name** | Login |
| **Description** | This use case will describe how a user logs in our web application. |
| **Actors** | Patient, Doctor or Clinic |
| **Precondition** | User must be already registered |
| **Scenario** | • User will select the type of login i.e.,Patient,Doctor or Clinic.<br>• User will enter their credential<br>• User will press the login button. |
| **Expectations** | User might get logged into the web application. |
| **Post Conditions** | User will get logged into the web application. |

Table 3.3: UC-03 Update Information

| | |
|---|---|
| **ID** | UC-03 |
| **Name** | Update Information |
| **Description** | This use case will describe how a user changes their general information. |
| **Actors** | Patient, Doctor or Clinic |
| **Precondition** | User must be logged in |
| **Scenario** | • User log in as a doctor, patient or clinic.<br>• User will then update the fields available.<br>• The available fields will vary for each type of login. |
| **Expectations** | User might update their information. |
| **Post Conditions** | User information is updated. |

Table 3.4: UC-4 Book Appointment

| | |
|---|---|
| **ID** | UC-04 |
| **Name** | Book Appointment |
| **Description** | This use case will describe how a user books an appointment |
| **Actors** | Patient |
| **Precondition** | User must be logged in as patient |
| **Scenario** | • User will select the book appointment option.<br>• User will then select their desired doctor.<br>• The appointment will be booked if the doctor is available. |
| **Expectations** | User might book an appointment with a doctor |
| **Post Conditions** | • If the doctor is available the appointment will be booked.<br>• If the doctor is not available the user will be asked to select a different doctor |

Table 3.5: UC-05 View Appointment history

| | |
|---|---|
| **ID** | UC-05 |
| **Name** | View Appointment history |
| **Description** | This use case will describe how a user views the details of their past appointments. |
| **Actors** | Patient |
| **Precondition** | • User must be logged in as a patient.<br>• User must have booked doctor appointments using the web application in the past. |
| **Scenario** | • User will select the view appointment history option.<br>• All the past appointment history of the patient will view in front of them |
| **Expectations** | User might view their appointment history |
| **Post Conditions** | The user's appointment history will be loaded.<br>. |

Table 3.6: UC-06 Download Prescription

| | |
|---|---|
| **ID** | UC-06 |
| **Name** | Download Prescription |
| **Description** | This use case will describe how a user downloads the prescription given by a doctor. |
| **Actors** | Patient |
| **Precondition** | • User must be logged in as a patient.<br>• Any doctor must have given the patient a prescription |
| **Scenario** | • User will select the download prescription option.<br>• User search for the prescription they want to download from the list of the prescriptions prescribed to them.<br>• The required prescription will be downloaded |
| **Expectations** | User might download a prescription. |
| **Post Conditions** | The prescription required is downloaded. |

Table 3.7: UC-07 Update Medical Record

| | |
|---|---|
| **ID** | UC-07 |
| **Name** | Update Medical Record |
| **Description** | This use case will describe how a user will upload their past medical records. |
| **Actors** | Patient |
| **Precondition** | User must be logged in as patient |
| **Scenario** | • User will select the upload medical records option.<br>• The user will then upload the desired filed by browsing through their device. |
| **Expectations** | User's files might be uploaded to the website. |
| **Post Conditions** | The user successfully uploaded their past medical records. |

TABLE 3.8: UC-08 Get help from chatbot

| | |
|---|---|
| **ID** | UC-08 |
| **Name** | Get help from chatbot |
| **Description** | This use case will describe how a user can use the chatbot to solve their general queries and make the best out of the web application. |
| **Actors** | Patient |
| **Precondition** | User must be logged in as patient |
| **Scenario** | • User will click on the chatbot icon.<br>• The user will then type in their query..<br>• The chatbot will get back to them regarding their query |
| **Expectations** | User might get the answer they required for their query. |
| **Post Conditions** | The user will get the required guidance. |

Table 3.9: UC-09 View Patient history

| | |
|---|---|
| **ID** | UC-09 |
| **Name** | View Patient history |
| **Description** | This use case will describe how a user will view the medical history of their patient. |
| **Actors** | Doctor |
| **Precondition** | • User must be logged in as a doctor.<br>• A patient must have booked an appointment with the doctor.<br>• The medical history of the patient does exist. |
| **Scenario** | • User will select patient that has booked appointment with them<br>• The user will then select the view medical history option.<br>• The medical history of the patient will be loaded in front of the doctor. |
| **Expectations** | User might get the history of their patient. |
| **Post Conditions** | The doctor gets the medical history of their patient. |

Table 3.10: UC-10 Prescribe Patient

| | |
|---|---|
| **ID** | UC-10 |
| **Name** | Prescribe Patient |
| **Description** | This use case will describe how a doctor will write a prescription to their patient. |
| **Actors** | Doctor |
| **Precondition** | • User must be logged in as a doctor.<br>• The doctor must have had an appointment with the patient they are willing to write a prescription to. |
| **Scenario** | • User will select the write prescription option.<br>• The user will then browse through the past appointments with various patients.<br>• The user will select the desired patient.<br>• The user will then fill in a short form.<br>• Finally, the user will press the issue prescription button. |
| **Expectations** | The doctor may issue a prescription for their patient. |
| **Post Conditions** | The prescription is issued to the desired patient. |

Table 3.11: UC-11 Online Diagnosis

| | |
|---|---|
| **ID** | UC-11 |
| **Name** | Online Diagnosis |
| **Description** | This use case will describe how a patient and doctor will interact through a video call for an online consultation. |
| **Actors** | Doctor,Patient |
| **Precondition** | • The doctor must be logged in to their account.<br>• The patient must be logged in to their account.<br>• The patient must have booked an appointment with the doctor for online consultation. |
| **Scenario** | • The doctor will initiate the video call when it's time for the appointment.<br>• The user will receive a link on their phone and email.<br>• When the user clicks on that link, they will join the doctor on a video call. |
| **Expectations** | Doctor may diagnose the patient online through a video call. |
| **Post Conditions** | The doctor successfully diagnoses the patient on a video call. |

Table 3.12: UC-12 Manage Appointments

| | |
|---|---|
| **ID** | UC-12 |
| **Name** | Manage Appointments |
| **Description** | This use case will describe how a doctor can manage their appointments using this platform. |
| **Actors** | Doctor |
| **Precondition** | User must be logged in as a doctor. |
| **Scenario** | • User will select the appointments option.<br>• It will show all the past and upcoming appointments to the doctor separately..<br>• The doctor can also see the appointment requests.<br>• The doctor can change their availability status. |
| **Expectations** | The doctor might be able to manage their appointments. |
| **Post Conditions** | The doctor is successful in managing their appointments. |

Table 3.13: UC-13 Manage Doctor

| | |
|---|---|
| **ID** | UC-13 |
| **Name** | Manage Doctor |
| **Description** | This use case will describe how a clinic can manage the doctors working for them. |
| **Actors** | Clinic |
| **Precondition** | User must be logged in as a Clinic. |
| **Scenario** | • The user selects the manage doctors option.<br>• The user can view the complete list of the doctors working under them.<br>• The user can add or delete doctors to and from the list.<br>• The user can also update the information of a doctor. |
| **Expectations** | User might be able to manage the doctors working under their clinic. |
| **Post Conditions** | The user is successful in managing the doctors working under their clinic |

Table 3.14: UC-14 Manage Patients

| | |
|---|---|
| **ID** | UC-14 |
| **Name** | Manage Patients |
| **Description** | This use case will describe how a clinic can manage their patients. |
| **Actors** | Clinic |
| **Precondition** | User must be logged in as a Clinic. |
| **Scenario** | • The user selects the manage patients option.<br>• The user can view the complete list of the patients being treated at their clinic.<br>• The user can add or delete patients to and from the list.<br>• The user can also update the information of a patient. |
| **Expectations** | User might be able to manage the patients being treated at their clinic |
| **Post Conditions** | The user is successful in managing the patients being treated at their clinic. |

### 3.3.2 Sequence Diagrams:

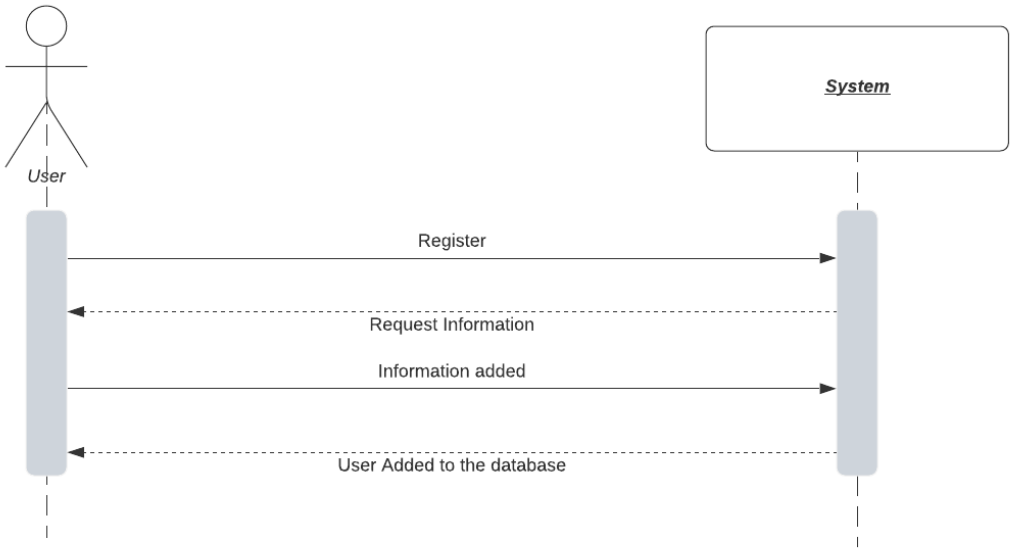


FIGURE 3.4: SD UC-01 Register

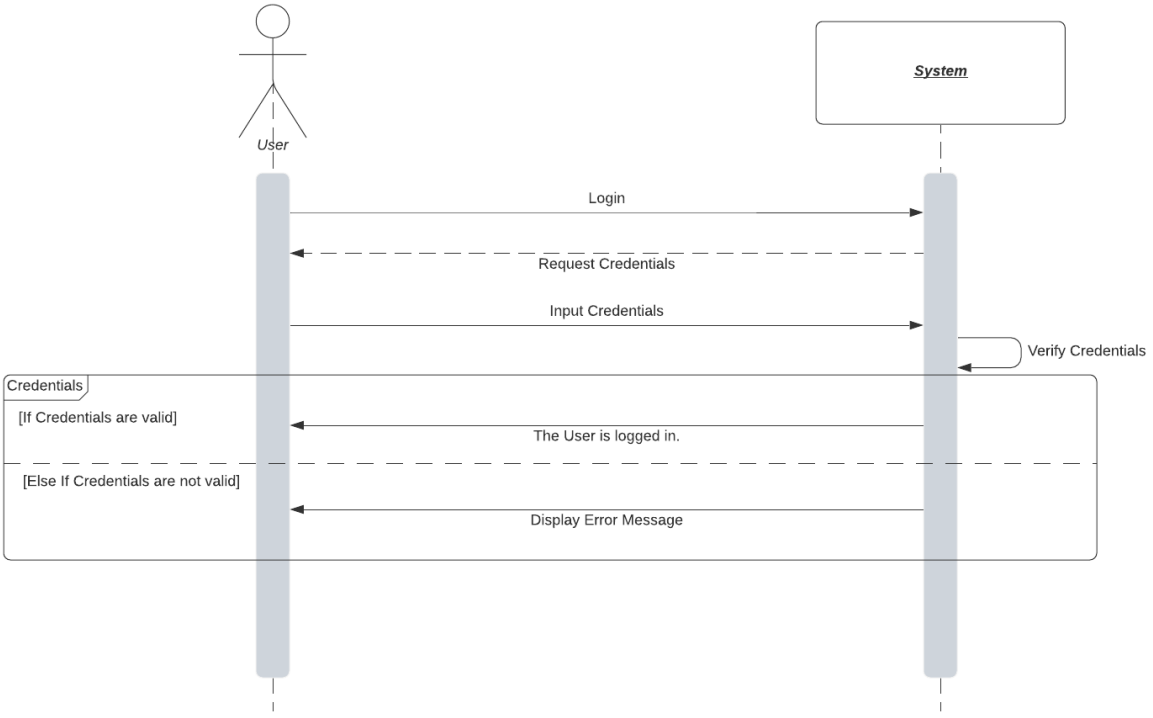


FIGURE 3.5: SD UC-02 Login

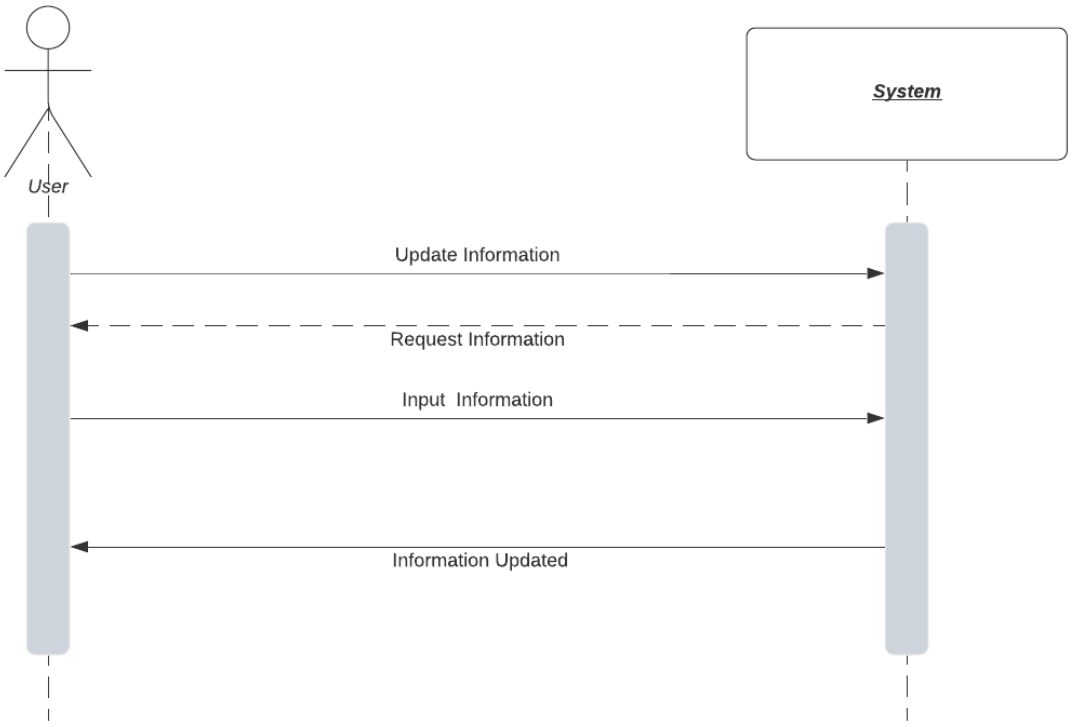


FIGURE 3.6: SD UC-03 Update Information

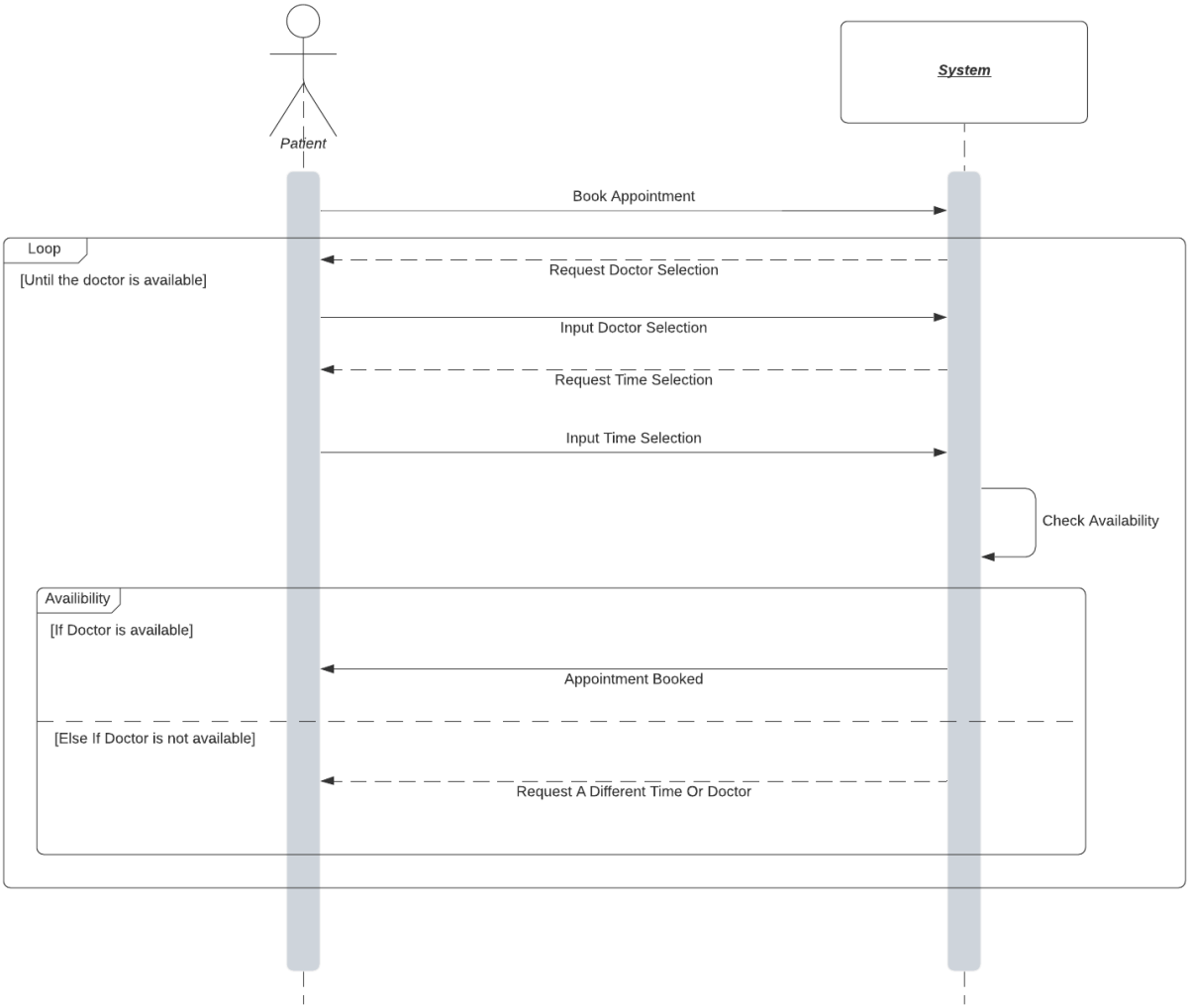


FIGURE 3.7: SD UC-04 Book Appointment

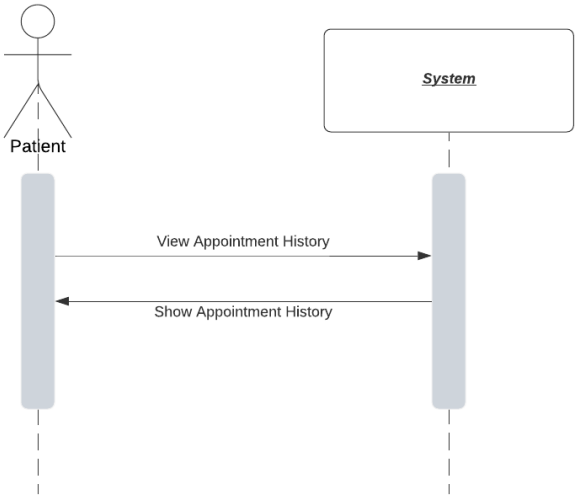


FIGURE 3.8: SD UC-05 View Appointment History

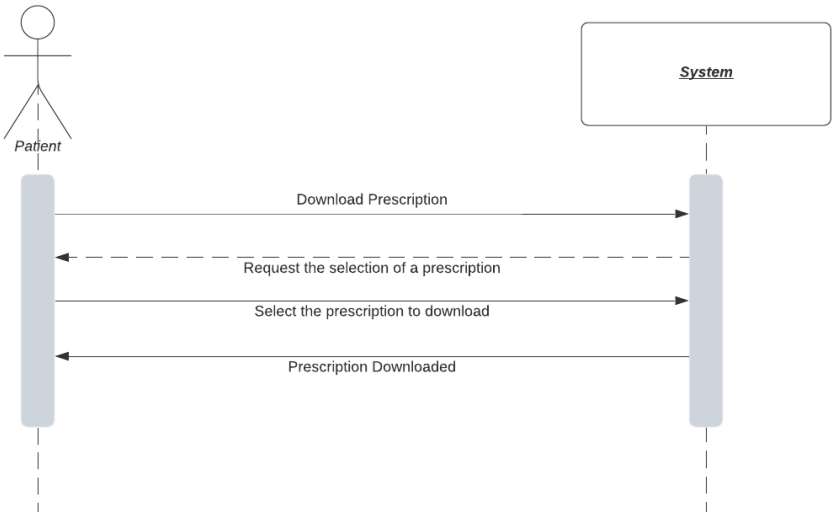


FIGURE 3.9: SD UC-06 Download Prescription

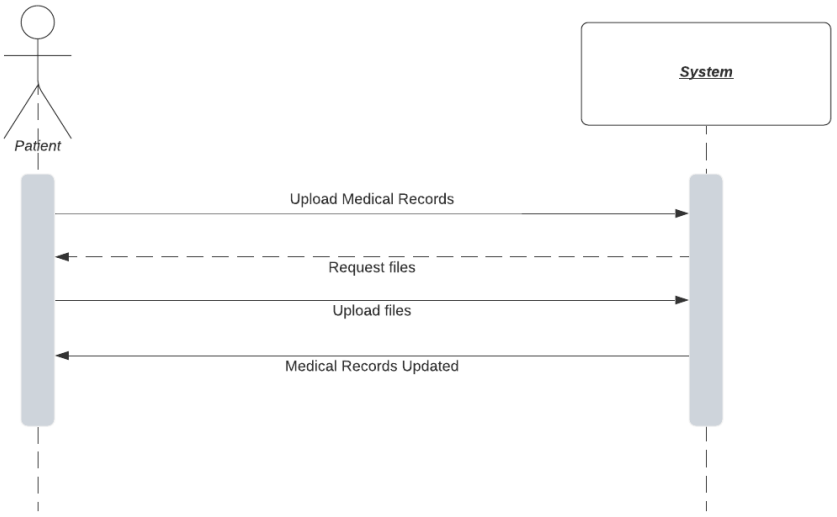


FIGURE 3.10: SD UC-07 Upload Medical Reports

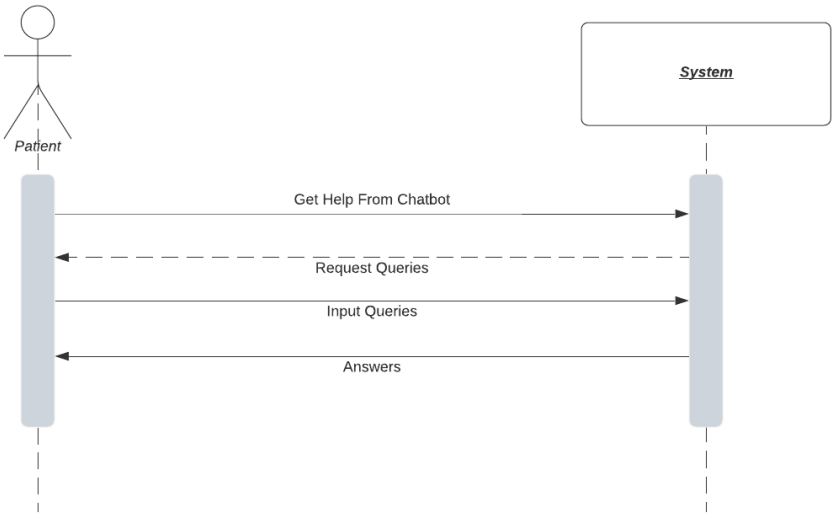


FIGURE 3.11: SD UC-08 Get help from chatbot

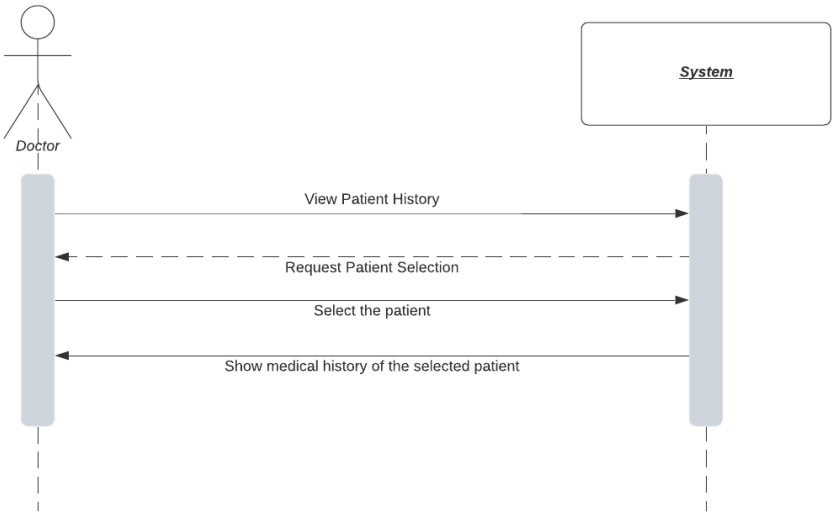


FIGURE 3.12: SD UC-09 View Patient History

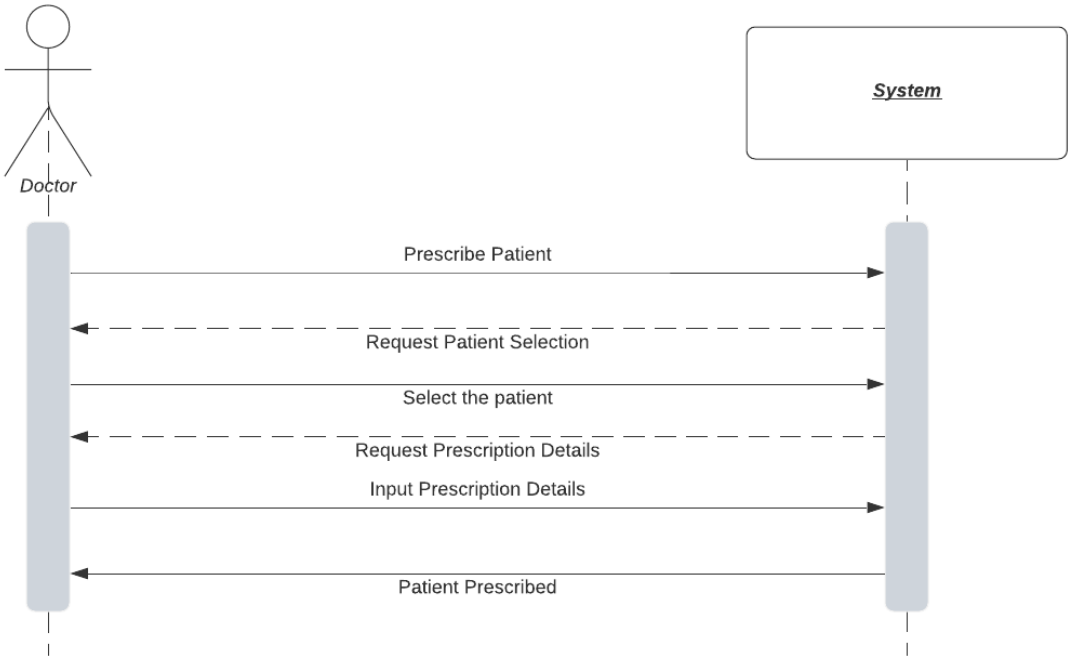


FIGURE 3.13: SD UC-10 Prescribe Patient

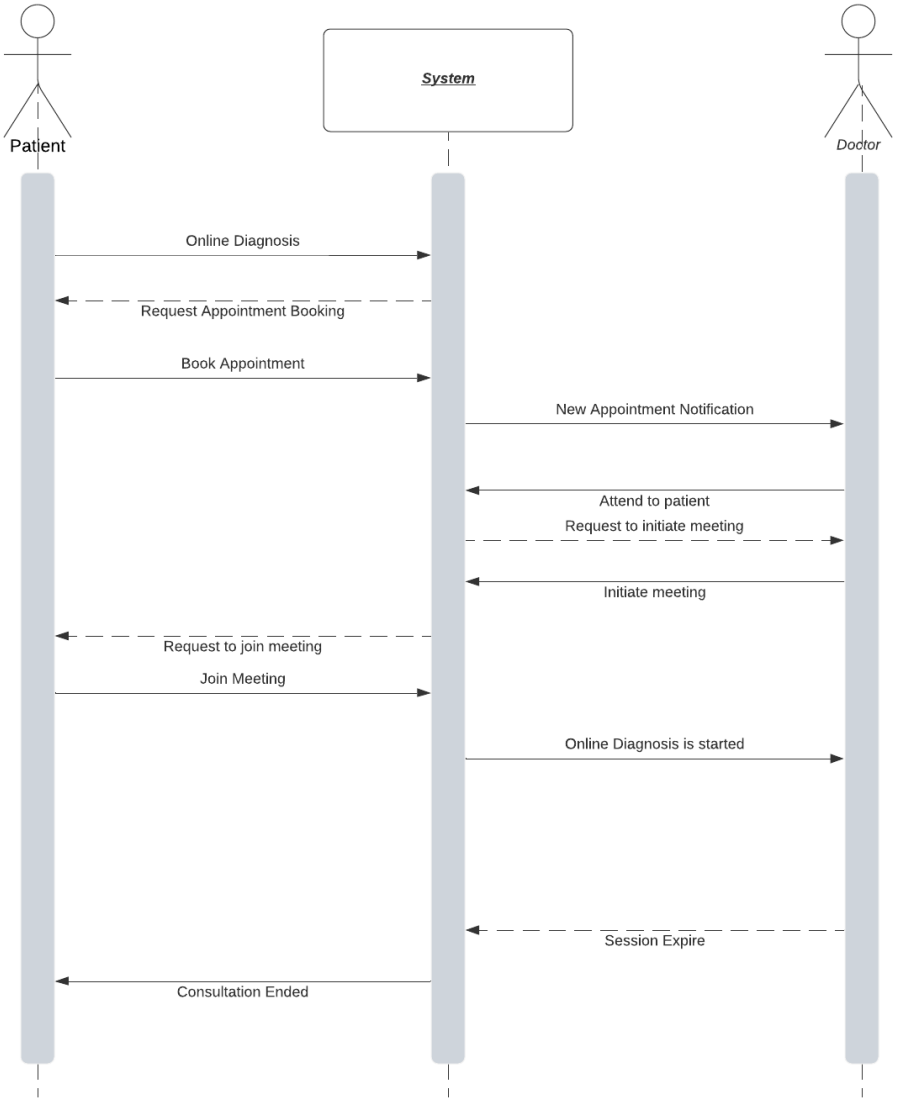


FIGURE 3.14: SD UC-11 Online Diagnosis

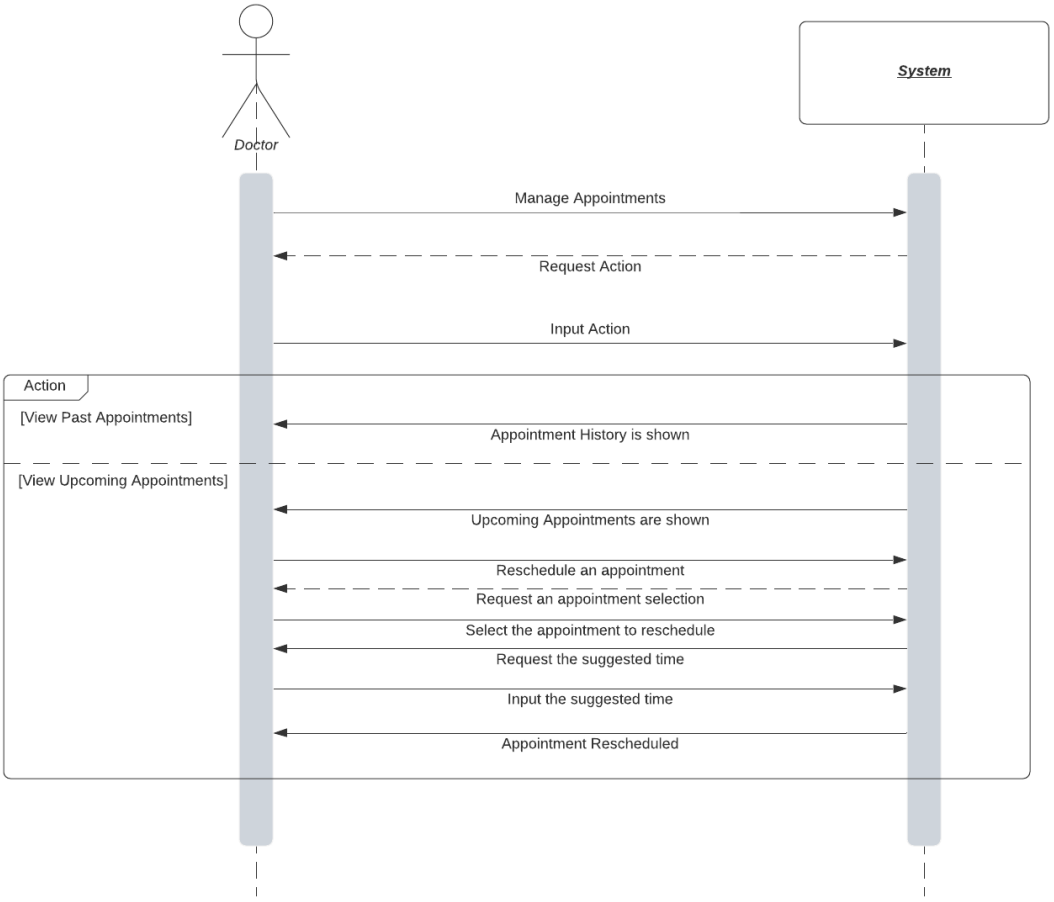


FIGURE 3.15: SD UC-12 Manage Appointments

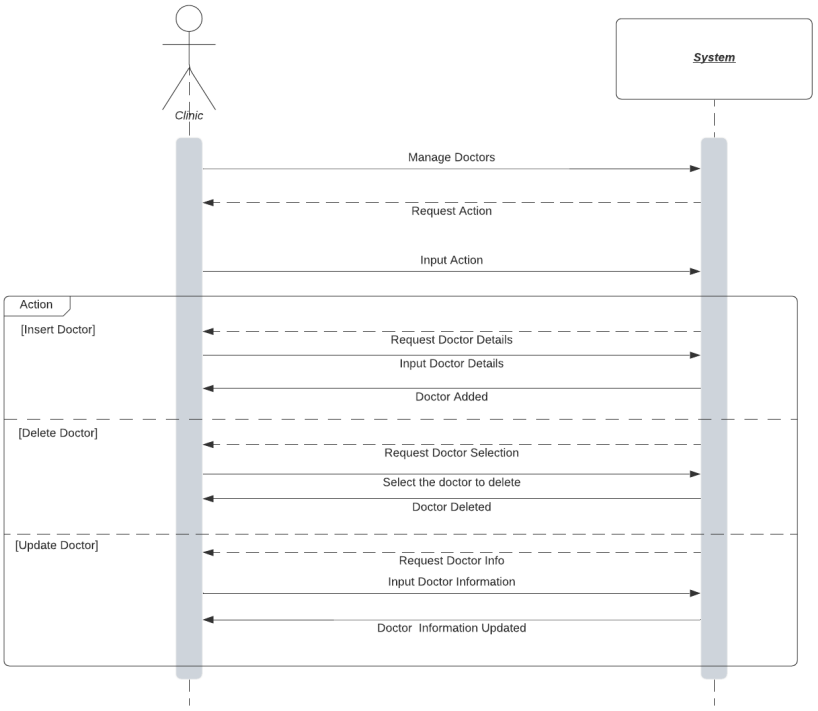


FIGURE 3.16: SD UC-13 Manage Doctors

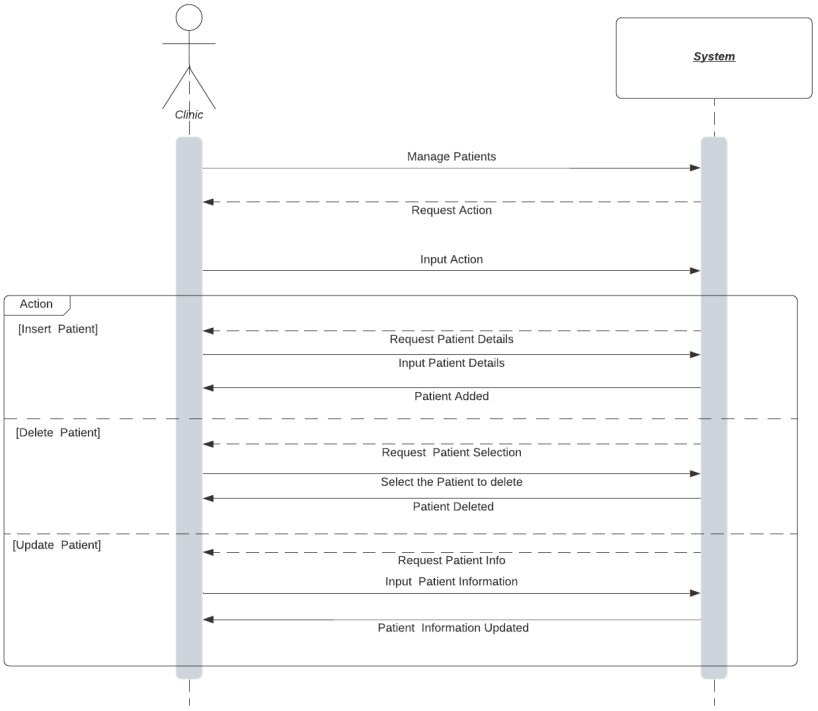


FIGURE 3.17: SD UC-14 Manage Patients

### 3.3.3 Class Diagram:

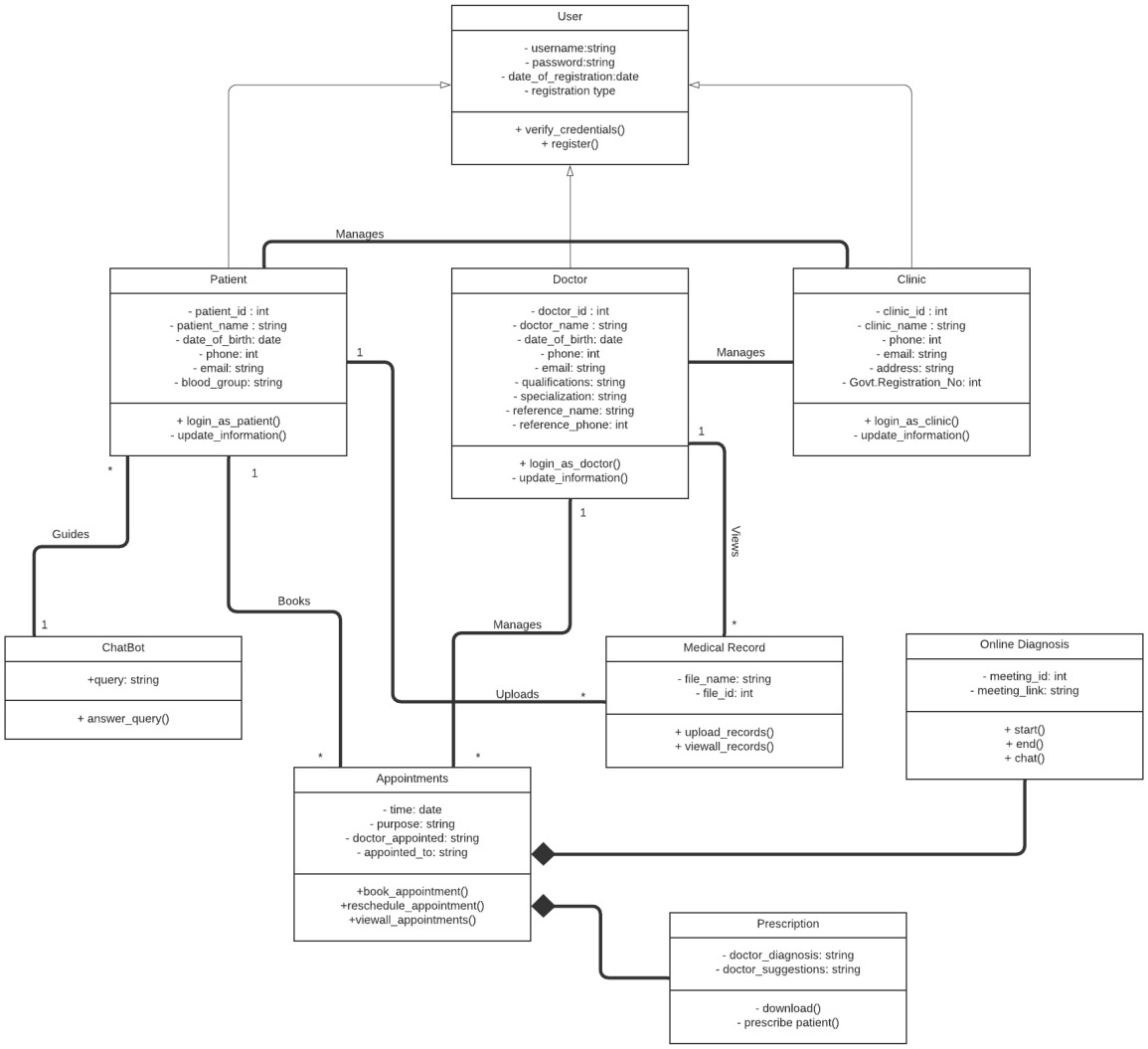


FIGURE 3.18: UML Class Diagram

### 3.3.4 Data Flow Diagrams:

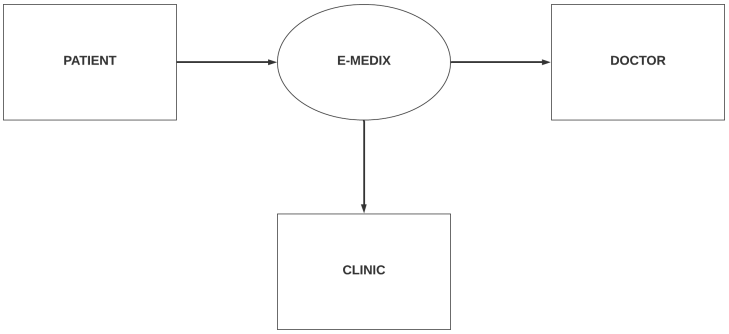


FIGURE 3.19: Level 0 - Data Flow Diagram

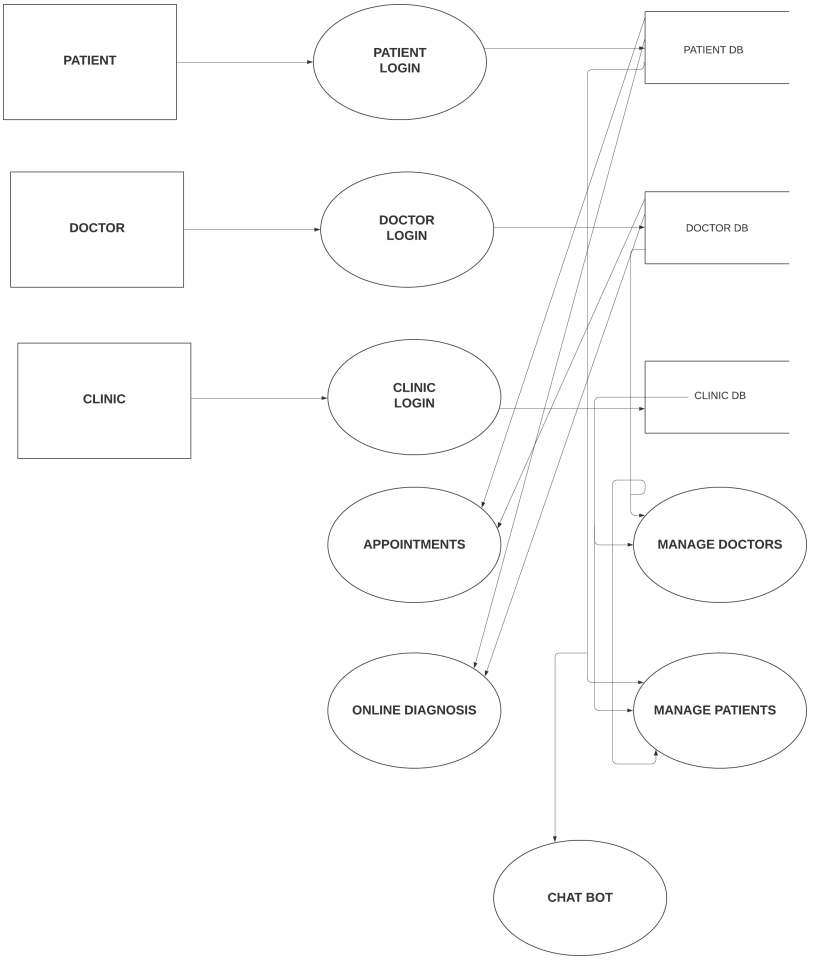


FIGURE 3.20: Level 1 - Data Flow Diagram

### 3.3.5 Entity Relationship Diagram:

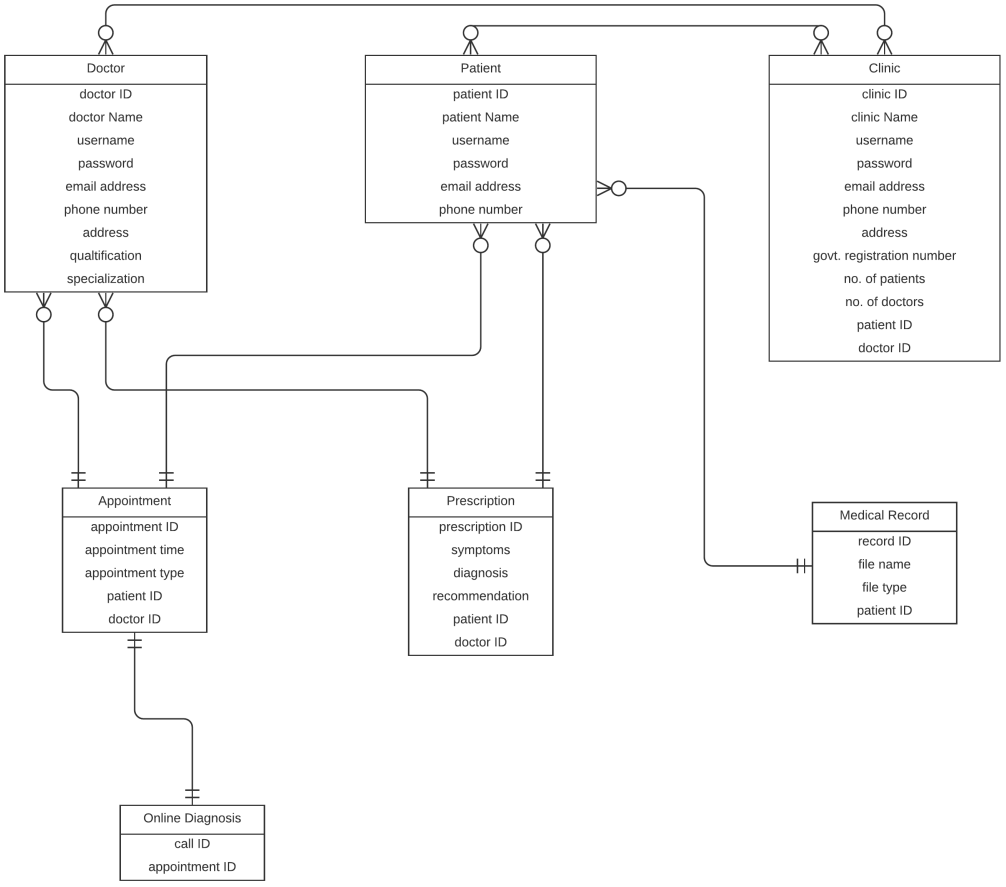


FIGURE 3.21: Entity Relationship Diagram

# Chapter 4

# Implementation and Evaluation

## 4.1 Development Stages

The software development life cycle(**SDLC**) stages are Requirement Analysis, Design, Development, Testing, Implementation and Maintenance.[9]The major aim of SDLC is to improve the development quality of the application to meet or exceed the set requirements. It also ensures that the timeline of the project is following and the project is completed within the cost estimated. Depending on the these stages our application also exhibits the key features of these stages. Every Stage of the SDLC plays a vital role in chalking out the "ready to hit market" web application. Let's look into how we carried forward our project through each stage of the SDLC[8].

### 4.1.1 Stage 1(Requirements Gathering and Analysis):

The first stage of the SDLC is Requirements Gathering and Analysis. This stage of the SDLC plays a crucial role in the development of our web application. To gather the requirements for the project we conducted one on one surveys with health care workers and explained them the idea of our platform and asked them about how will our system affect their practice. 95 Percent of the total conducted

surveys were positive Hence reflecting a need of a platform like ours in the society. Moreover we also conducted interviews with the owners of the existing similar platforms and after analysing their work flow we tried to target the areas they were collectively lacking in.

### 4.1.2 Stage 2(Design):

The second stage of our project was the design phase. During this phase we decided the basic flow and working of our project and worked on how we want our project to look like. We as a team did brainstorming and noted down the ideas of the flow of the web application and created a list of ways a user will interact with the web application. Once the flow was finalized we went on to decide the look and feel of the platform. We analysed the top trending websites to get an idea of the UI trends. Then we created Mocks with the help of tools like Adobe Photoshop and Adobe XD to finalize how our web application should look like.

### 4.1.3 Stage 3(Development):

The third stage of the SDLC is the development stage. During this phase we converted all the planning from the stage 2 into code. The development stage was subdivided into two sub phases. Front End Development and Back End Development. During the front end phase we converts the selected User interface from pictorial form into markup language. We used HTML5, CSS 3, Bootstrap4 and Javascript to code our front end. Once all the pages were coded we went on to the back end development to add the functionalities to our user interface. Our Back End was majorly integrated using the programming language Python through the Django web framework. However, during the integration of video consultation several modules of Node.js were also used.

### 4.1.4 Stage 4(Testing):

This fourth stage of the SDLC is the Testing stage. This is a very critical stage of the Software development process. During the testing phase of the SDLC multiple tests were conducted on the developed web application to ensure the quality of the platform. The testing stage involved the testing of each functionality. Every button was clicked individually to analyse it's working and different scenarios were repeatedly conducted and the behaviour of the system was observed to filter out the abnormalities. Apart from the manual testing automated testing was also incorporated during the testing stage. Scripts were written and executed and run to perform different tasks on the platform. The script analysed how many times did the desired task failed or passed. A tool named Jmeter was also used for an overall vulnerability test. All the bugs and errors in the system were detected, analysed and fixed until no error or bug was left.

### 4.1.5 Stage 5(Implementation):

The fifth stage of the SDLC is the Implementation stage. Once all the errors are omitted in the testing stage the web application was deployed and made available to be accessed through a URL and the URL was sent to a group of individuals to proceed to the next stage.

### 4.1.6 Stage 6(Maintenance):

The last stage of the SDLC is the Maintenance stage. This stage helps you to know the opinion of the end user. During this stage the feedback of the group of individuals with the URL for our web application was looked into. This phase helped us to reanalyze some of the functionalities of our system and resulted in the updating on the functionalities of specific modules.

## 4.2 System Integration

The different components of the E-Medix web application were bought together in the views.py file. Most of the components are co related and act as the foundation of the functionalities of this platform. Successful execution of each component results in the proper functioning of E-Medix.

## 4.3 User Interface

A good interface is almost unnoticeable by the user. It does not have any unnecessary elements and uses clear language to portray a clear message. A good UI is also consistent and it is created such that it enhances the user's experience on the web application. For our web application we analyzed the general UI trends in the industry and then chalked out a user interface accordingly. Our UI was designed as per the W3C standards and it was created such that it makes the experience of a user coming to our web application better. Our UI is also self explanatory and is targeted to help the users to navigate through the whole website without any external help.

**Following are the main UI components of our Web Application:**

### 4.3.1 Main page

FIGURE 4.1: Main Page

### 4.3.2 Sign up

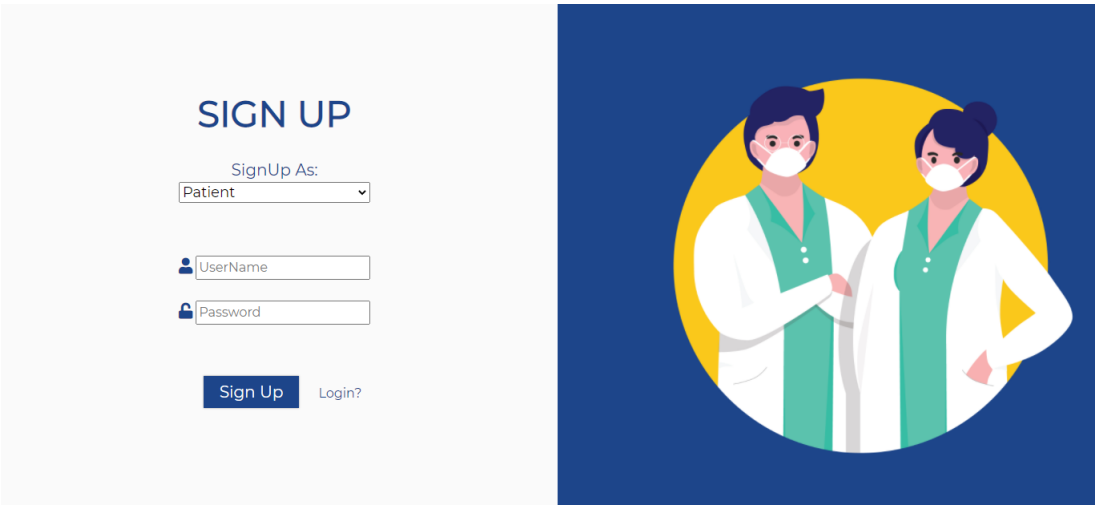


FIGURE 4.2: Sign up

### 4.3.3 Login

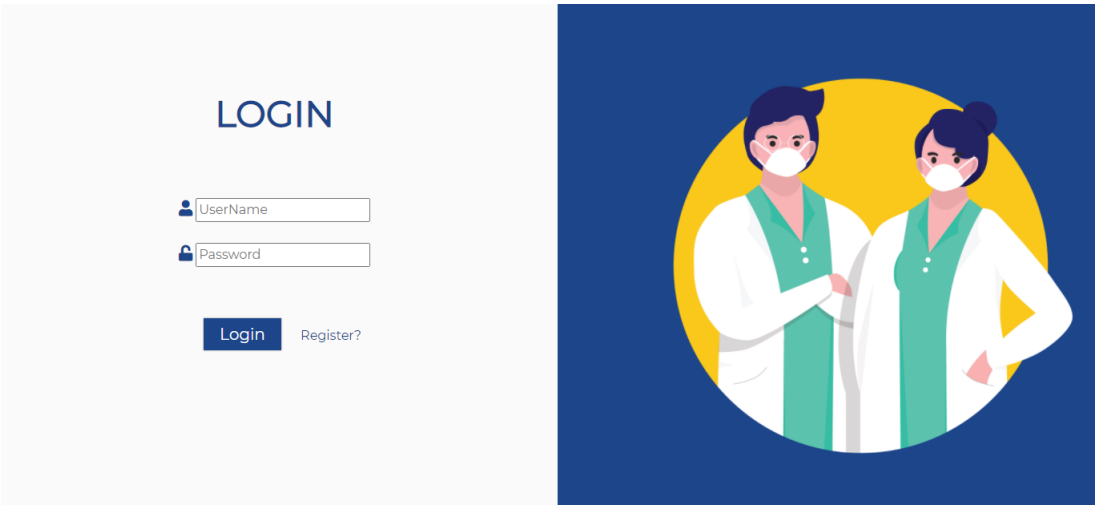


FIGURE 4.3: Login

### 4.3.4 Patient Details

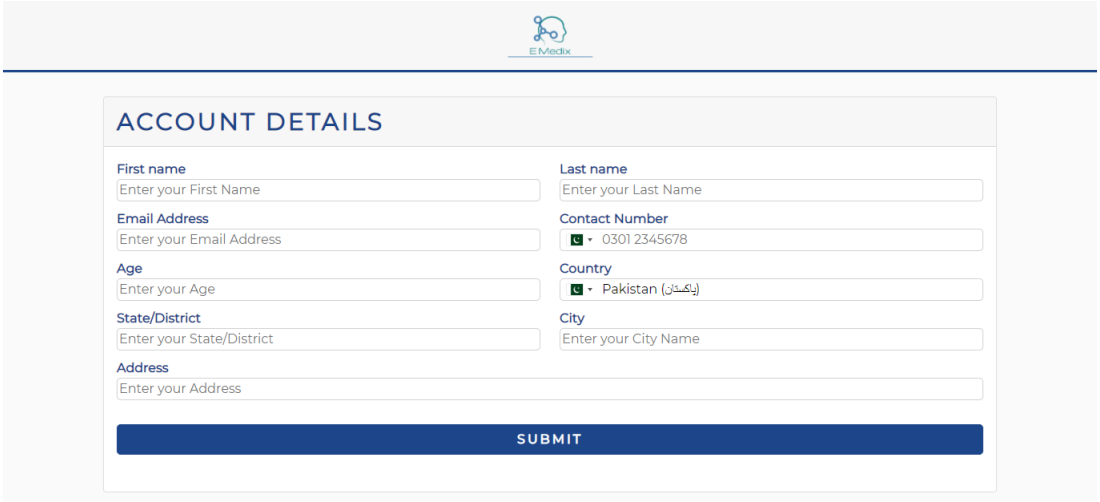


FIGURE 4.4: Patient Dashboard

### 4.3.5 Doctor Details

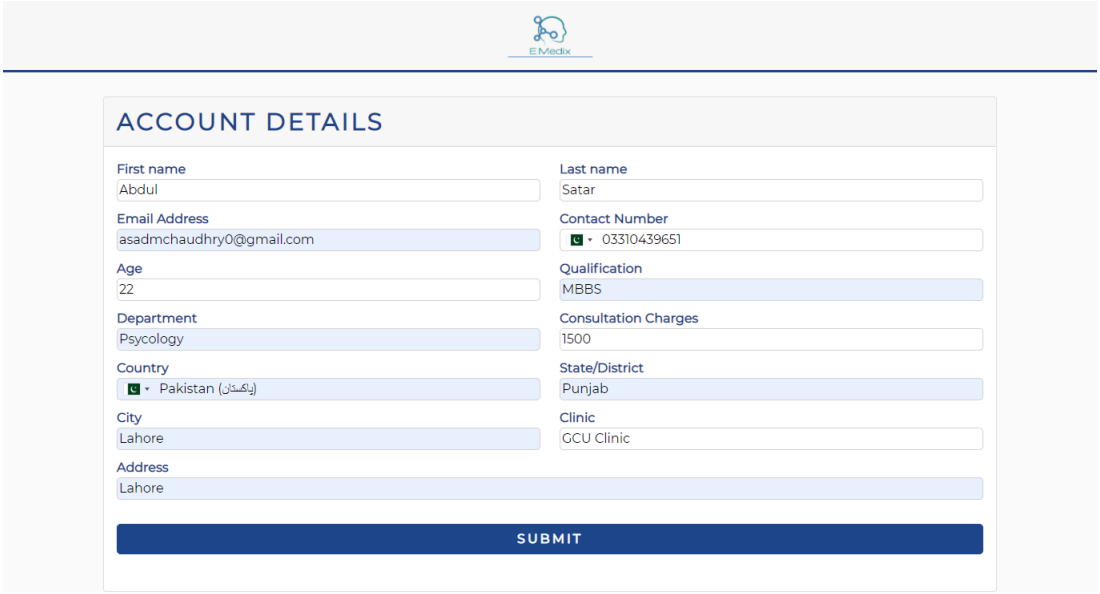


FIGURE 4.5: Doctor Dashboard

### 4.3.6 Clinic Details

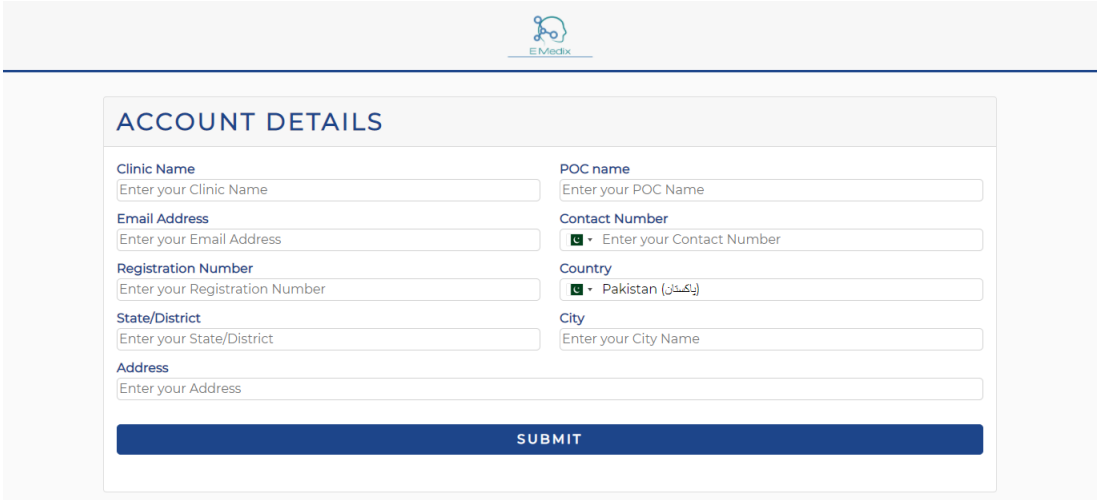


FIGURE 4.6: Clinic Dashboard

### 4.3.7 Two Factor Verification

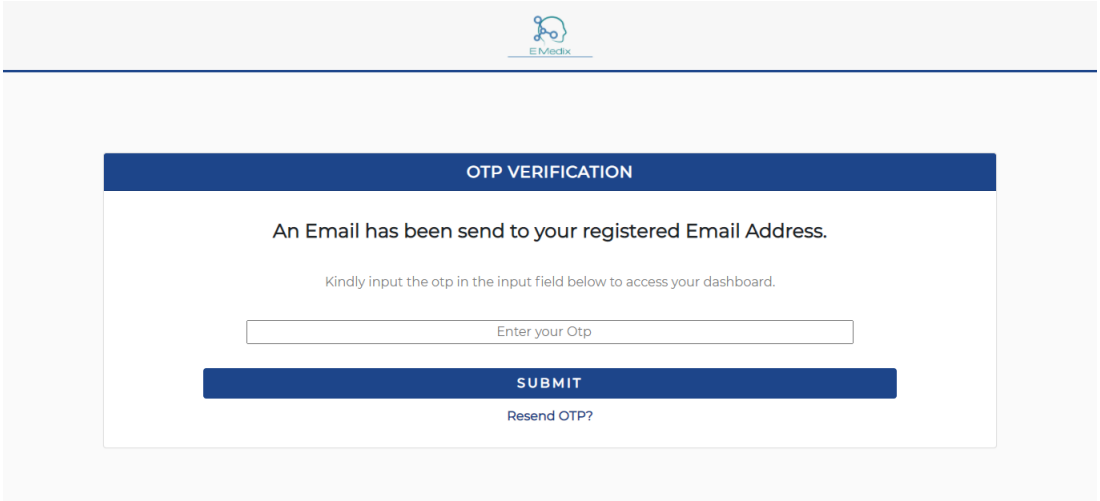


FIGURE 4.7: Two Factor Verification

### 4.3.8 OTP Email

FIGURE 4.8: OTP Email

### 4.3.9 Patient Dashboard

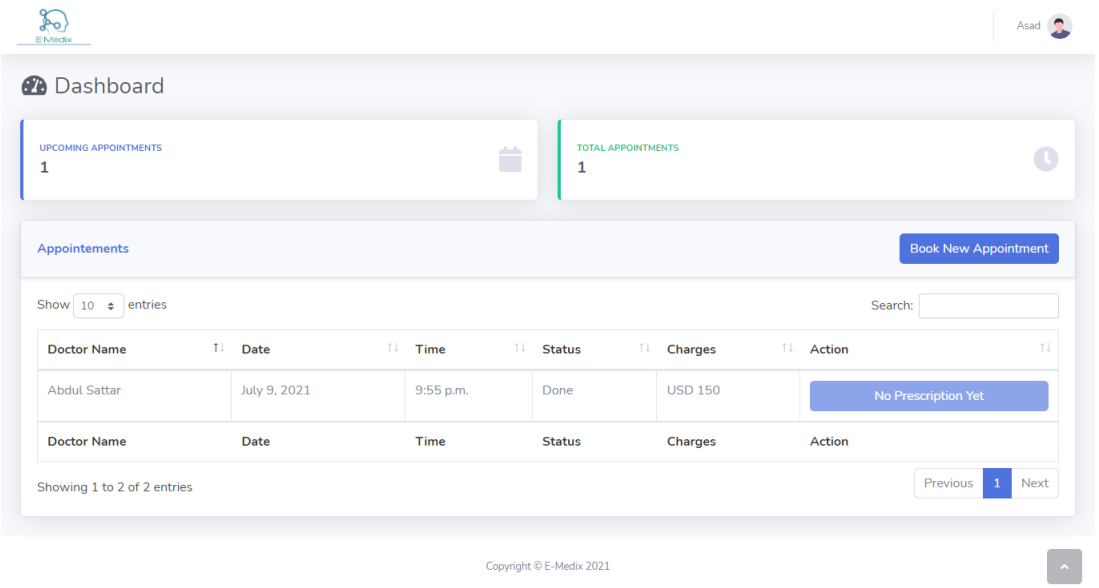


FIGURE 4.9: Patient Dashboard

### 4.3.10 Book Appointment

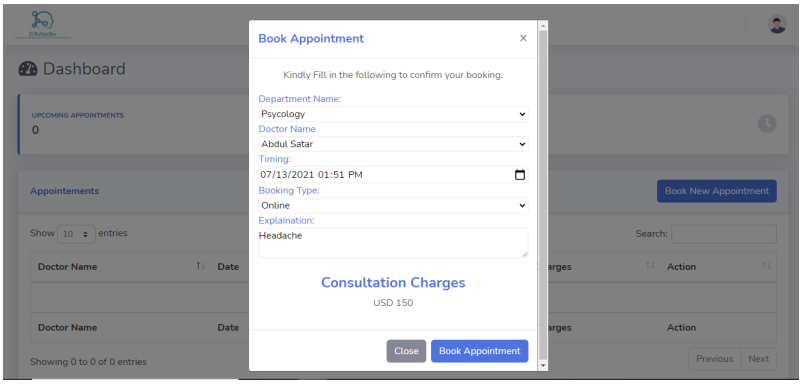


FIGURE 4.10: Book Appointment

### 4.3.11 Payment

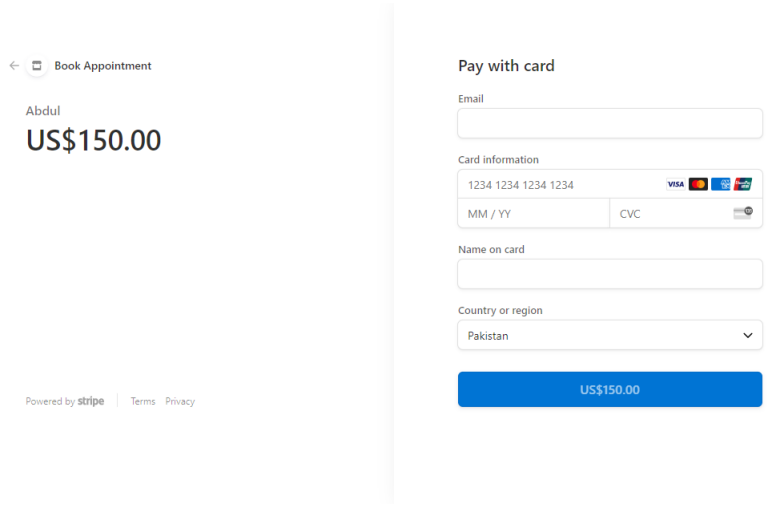


FIGURE 4.11: Payment

### 4.3.12 Payment Success

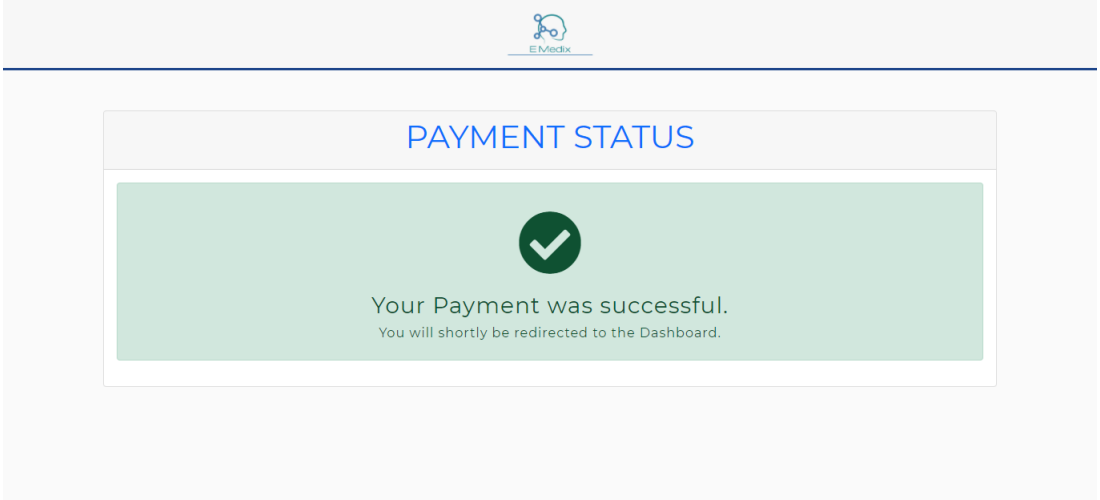


FIGURE 4.12: Payment Success

### 4.3.13 Payment Error

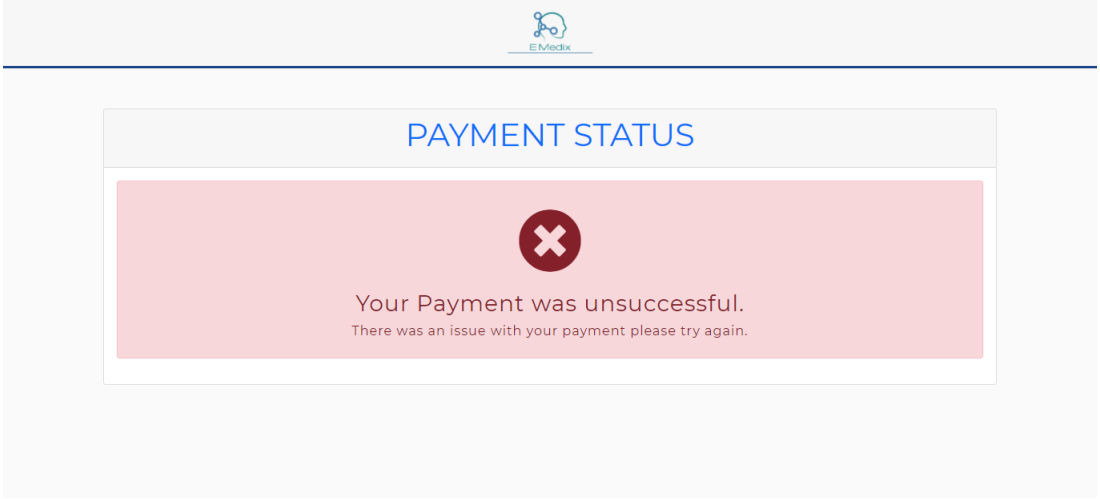


FIGURE 4.13: Payment Error

### 4.3.14 Confirmation Email Patient

FIGURE 4.14: Confirmation Email Patient

### 4.3.15 Confirmation Email Doctor

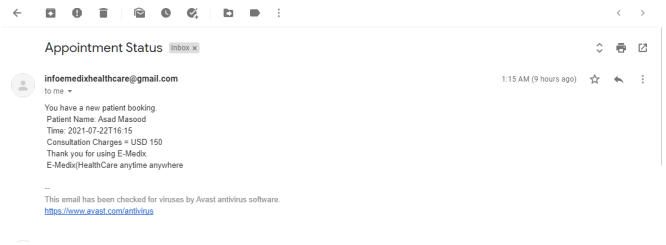


FIGURE 4.15: Confirmation Email Doctor

### 4.3.16 Doctor Dashboard

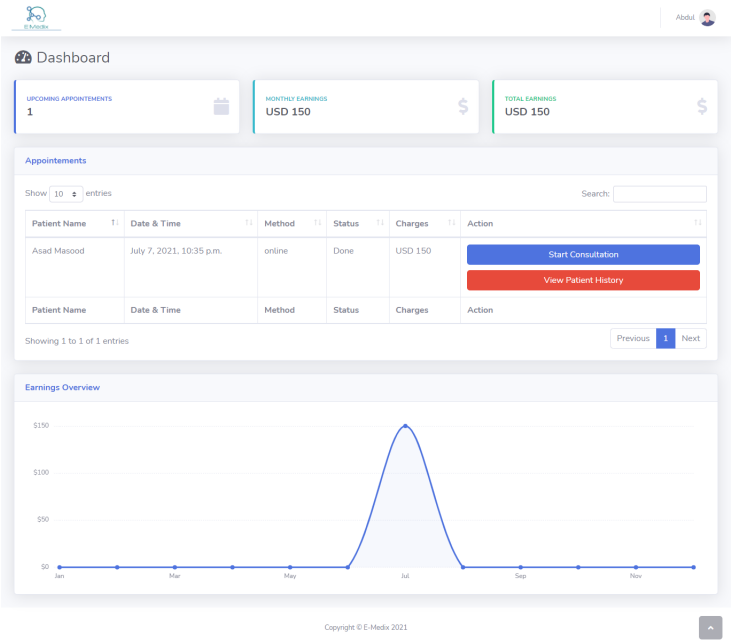


FIGURE 4.16: Doctor Dashboard

### 4.3.17 Prescribe Patient

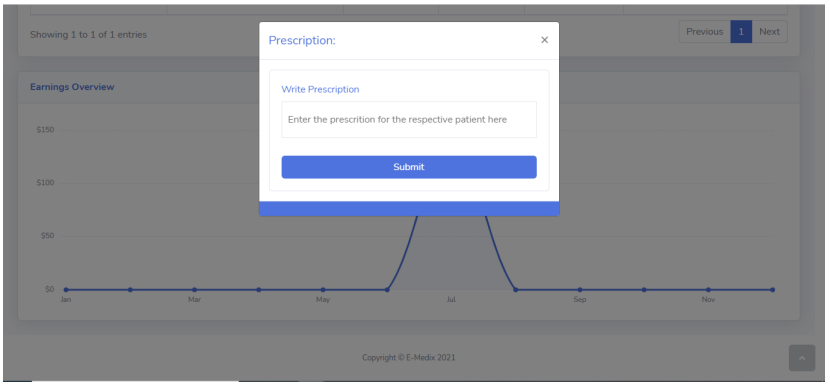


FIGURE 4.17: Prescribe Patient

### 4.3.18 View Prescription

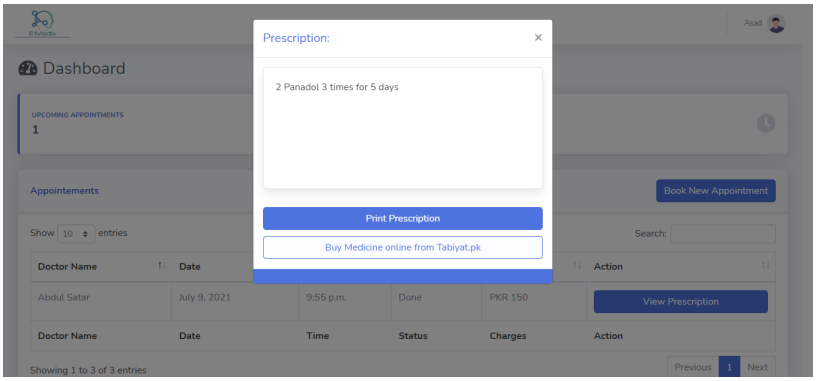


FIGURE 4.18: View Prescription

### 4.3.19 Clinic Dashboard

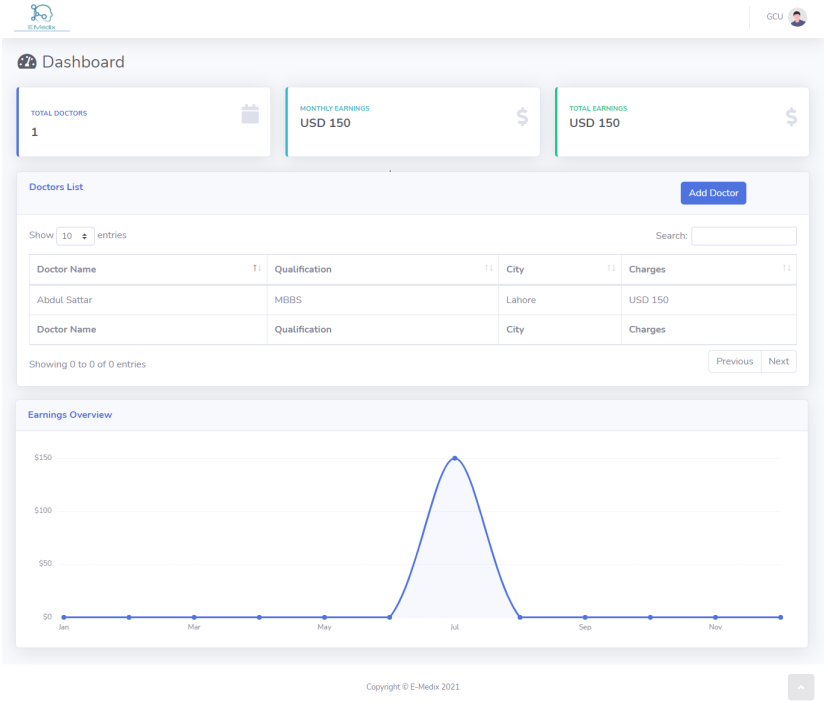


FIGURE 4.19: Clinic Dashboard

### 4.3.20 Update Patient Details

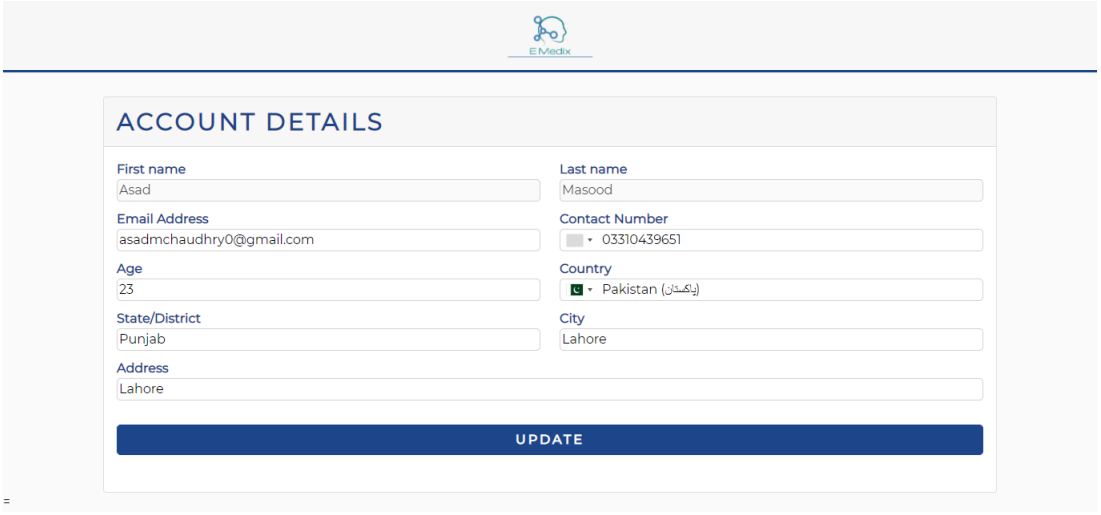


FIGURE 4.20: Update Patient Details

### 4.3.21 Update Doctor Details

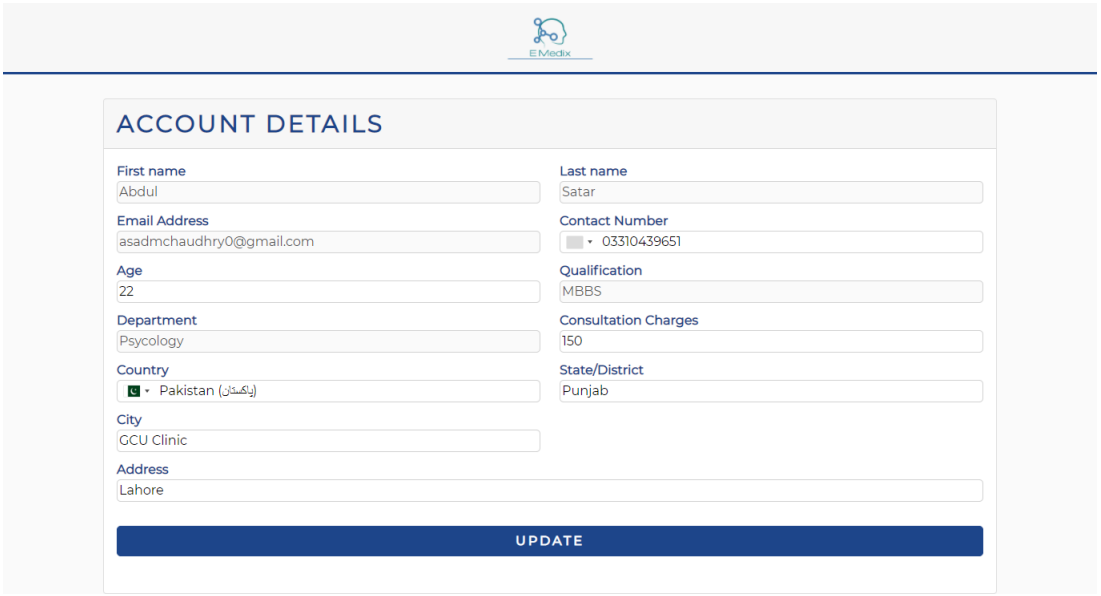


FIGURE 4.21: Update Doctor Details

### 4.3.22 Update Clinic Details

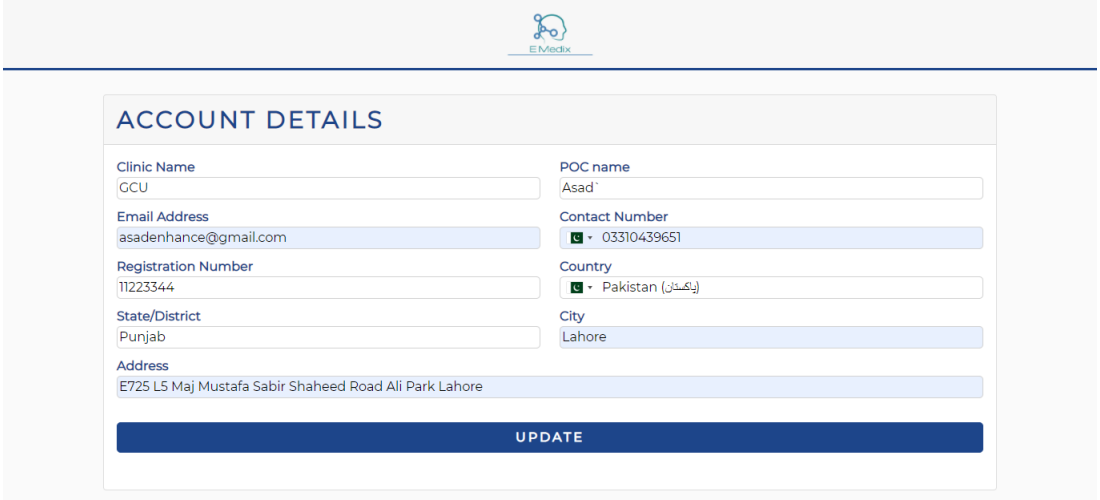


FIGURE 4.22: Update Clinic Details

### 4.3.23 Change Password

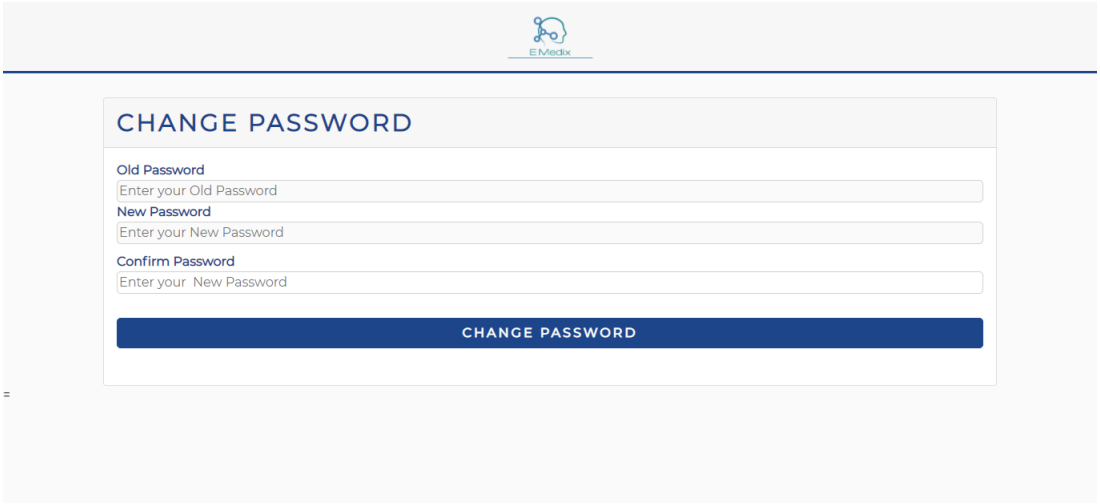


FIGURE 4.23: Change Password

### 4.3.24 Upload Medical Records

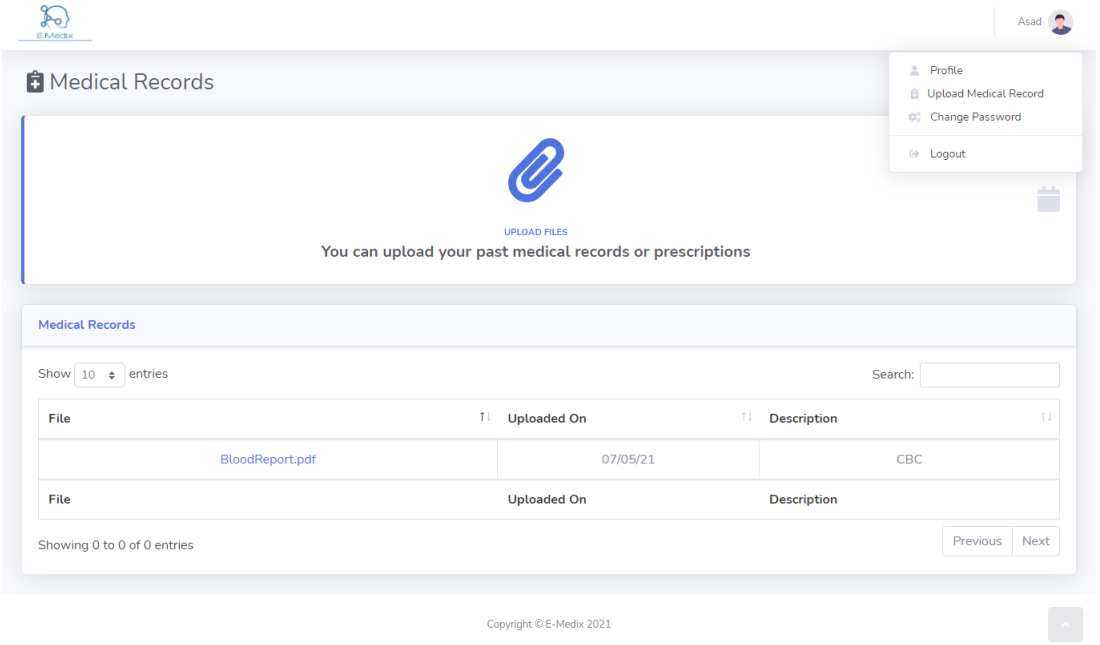


FIGURE 4.24: Upload Medical Records

### 4.3.25 View Medical Records

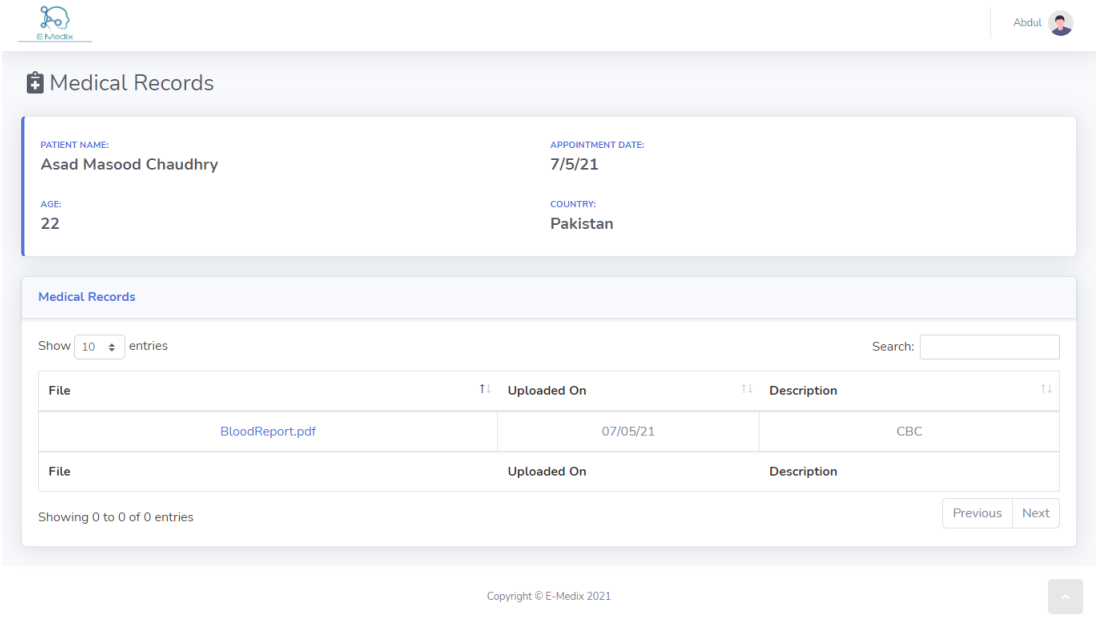


FIGURE 4.25: View Medical Records

### 4.3.26 Video Consultancy

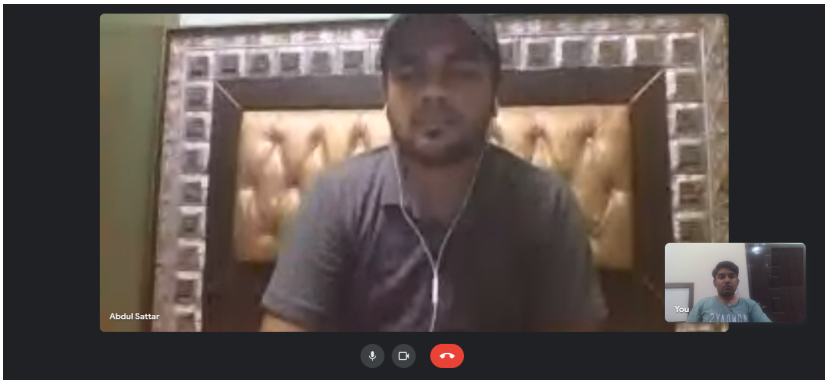


FIGURE 4.26: Video Consultancy

## 4.4 Evaluation

After the complete implementation of the code of each module they were integrated together as one unit to perform the complete functionalities. The resultant web application was then ready to be tested. Multiple tests were conducted and the detected errors were omitted.Different types of tests resulted in the detection of different types of errors. We conducted tests for back end code, tests for front end code, tests for the database connectivity, general feature testing, etc. It is certain that the web application will undergo changes once it is presented in front of the end user. These changes can occur due to several possible reasons. The system could be requesting some unexpected input from the user. Once the web application is being used by the end user the developer can analyze the requirement of some additional component or may find it necessary to modify the existing component. Changes in code have a great impact on the overall working of the web application. Testing processes were executed to ensure that each module is working as per the requirements after the changes in code have been made and any abnormality detected is immediately reported and removed. Near the completion of the project, we tested the application thoroughly, testing each individual form and button. We made sure that each module of the web application is working as expected in the form of a single unit.

- All tests were conducted to aid the user experience and requirements.
- Tests were organized and planned before the testing process begun.
- Third party services were availed to make the testing accurate.

## 4.5 Unit Testing

Testing is performed once the cohesion of framework is finalized. The purpose of the testing is to guarantee the rightness of the web application as far as the working of the modules is concerned. The criteria and steps of testing are laid out before the testing processes are initiated. The unit testing is centered around the rationals of interior preparing and all information structures introduced within the limits of the module. During unit testing we test every module independently. The testing engineer has the adaptability of changing the module undergoing the test until the required output is extracted from the module. We gave tested every module of our web application individually during the unit testing phase. During the unit testing if any error rises we know exactly which module to cater to remove the error.

**Chapter 4.6 discusses the test cases executed during the unit testing phase:**

## 4.6 Unit Tests

Table 4.1: Test Cases

| **Test Case No.** | Test Case Short Description |
|---|---|
| **T**C-01 | Register |
| **T**C-02 | Login |
| **T**C-03 | Two Factor Verification |
| **T**C-04 | Add Patient Details |
| **T**C-05 | Add Doctor Details |
| **T**C-06 | Add Clinic Details |
| **T**C-07 | Book Appointment |
| **T**C-08 | View Past Appointments |
| **T**C-09 | Order Medicines |
| **T**C-10 | Print Prescription |
| **T**C-11 | View Past Prescriptions |
| **T**C-12 | Consult Patient |
| **T**C-13 | Notify Patient |
| **T**C-14 | Prescribe Patient |
| **T**C-15 | Analyze Earnings |
| **T**C-16 | View Doctors |
| **T**C-17 | Change Password |
| **T**C-18 | Update Profile |
| **T**C-19 | Upload Medical Records |
| **T**C-20 | View Patient History |

### 4.6.1 TC-01 Register

Table 4.2: TC-01 Register

| | |
|---|---|
| **Test Case ID** | 01 |
| **Title** | Register |
| **Description** | User must register/sign up in order to get access to the platform. |
| **Input** | User must have the access to the home page of the website. |
| **Steps to Perform** | • User Visit the Home Page then click on Sign Up Button.<br>• He or she must select the user type i.e Patient, Doctor or Clinic.<br>• He or she must set Username and Password.<br>• Then he must press the sign up button. |
| **Output** | User is added to the E-Medix Platform. |

### 4.6.2 TC-02 Login

Table 4.3: TC-02 Login

| | |
|---|---|
| **Test Case ID** | 02 |
| **Title** | Login |
| **Description** | User Logs in to use the respective functionalities of specified Dashboard. |
| **Input** | User must already be registered to the platform and enter valid username and password. |
| **Steps to Perform** | • User Visit the Home Page then click on Log in Button.<br>• He or she must enter valid Username and Password.<br>• Then he must press the login button. |
| **Output** | User is logged in to E-Medix Platform. |

### 4.6.3 TC-03 Two Factor Verification

Table 4.4: TC-03 Two Factor Verification

| | |
|---|---|
| **Test Case ID** | 03 |
| **Title** | Two Factor Verification |
| **Description** | A four digit One time password should be entered to complete the login process. |
| **Input** | User must already have entered their valid username and password on the login screen. |
| **Steps to Perform** | • User enters the 4 digit one time password.<br>• He or she must click on the submit button. |
| **Output** | User is redirected to their respective Dashboard. |

### 4.6.4 TC-04 Add Patient Details

Table 4.5: TC-04 Add Patient Details

| | |
|---|---|
| **Test Case ID** | 04 |
| **Title** | Add Patient Details |
| **Description** | The Patient user adds the required information. |
| **Input** | The user must have selected the user type "Patient" when signing up. |
| **Steps to Perform** | • User signs up for the first time.<br>• User then entered all the required information the patient users needs to enter.<br>• User then presses the submit button |
| **Output** | User is redirected to the Patient Dashboard. |

### 4.6.5 TC-05 Add Doctor Details

Table 4.6: TC-05 Add Doctor Details

| | |
|---|---|
| **Test Case ID** | 05 |
| **Title** | Add Doctor Details |
| **Description** | The Doctor user adds the required information. |
| **Input** | The user must have selected the user type "Doctor" when signing up. |
| **Steps to Perform** | • User signs up for the first time.<br>• User then entered all the required information the doctor users needs to enter.<br>• User then presses the submit button |
| **Output** | User is redirected to the Doctor Dashboard. |

### 4.6.6 TC-06 Add Clinic Details

Table 4.7: TC-06 Add Clinic Details

| | |
|---|---|
| **Test Case ID** | 06 |
| **Title** | Add Clinic Details |
| **Description** | The Clinic user adds the required information. |
| **Input** | The user must have selected the user type "Clinic" when signing up. |
| **Steps to Perform** | • User signs up for the first time.<br>• User then entered all the required information the Clinic users needs to enter.<br>• User then presses the submit button |
| **Output** | User is redirected to the Clinic Dashboard. |

### 4.6.7 TC-07 Book Appointment

Table 4.8: TC-07 Book Appointment

| | |
|---|---|
| **Test Case ID** | 07 |
| **Title** | Book Appointment |
| **Description** | The user books an appointment to consult their medical issues with a doctor. |
| **Input** | The user books an appointment as per their requirements and pays for it. |
| **Steps to Perform** | • User clicks on the book appointment button.<br>• User then selects the department and doctor and adds the required information.<br>• User then goes on to the payment screen.<br>• User pays for booking fees.<br>• The user is redirected to a payment status screen. |
| **Output** | The appointment is successfully booked. |

### 4.6.8 TC-08 View Past Appointments

Table 4.9: TC-08 View Past Appointments

| | |
|---|---|
| **Test Case ID** | 08 |
| **Title** | View Past Appointments |
| **Description** | The User views all the past appointments they have ever had. |
| **Input** | The user opens the dashboard. |
| **Steps to Perform** | • User logs in as a patient.<br>• User completes the two factor verification.<br>• User scrolls down to the bookings table. |
| **Output** | The User can view the details of all their past appointments. |

### 4.6.9 TC-09 Order Medicines

Table 4.10: TC-09 Order Medicines

| | |
|---|---|
| **Test Case ID** | 09 |
| **Title** | Order Medicines |
| **Description** | The User orders medicines from a third party website Tabiyat.pk. |
| **Input** | The user opens a specific prescription. |
| **Steps to Perform** | • User clicks on the view prescription button.<br>• The user then clicks on the buy medicines from Tabiyat.pk button. |
| **Output** | The User is redirected to the Tabiyat.pk website. |

### 4.6.10 TC-10 Print Prescription

Table 4.11: TC-10 Print Prescription

| | |
|---|---|
| **Test Case ID** | 10 |
| **Title** | Print Prescription |
| **Description** | The User gets a printed form of prescription. |
| **Input** | The user opens a specific prescription. |
| **Steps to Perform** | • User clicks on the view prescription button.<br>• The user then clicks on the print prescription button. |
| **Output** | The User receives a printing options popup. |

### 4.6.11 TC-11 View Past Prescriptions

Table 4.12: TC-11 View Past Prescriptions

| | |
|---|---|
| **Test Case ID** | 11 |
| **Title** | View Past Prescriptions |
| **Description** | The user views the prescriptions of any past appointment. |
| **Input** | The user logs in and scrolls down to the appointments table. |
| **Steps to Perform** | • The user logs in as a patient.<br>• User clicks on the view prescription button. |
| **Output** | The User receives a prescription modal. |

### 4.6.12 TC-12 Consult Patient

Table 4.13: TC-12 Consult Patient

| | |
|---|---|
| **Test Case ID** | 12 |
| **Title** | Consult Patient |
| **Description** | The User consults their patients over a video call. |
| **Input** | The User is logged in as a doctor and a patient has booked an online appointment with them. |
| **Steps to Perform** | • The user logs in as a Doctor.<br>• User clicks on the consult patient button. |
| **Output** | The User is redirected to a video call screen and the respective patient is sent a link to the video call |

### 4.6.13 TC-13 Notify Patient

Table 4.14: TC-13 Notify Patient

| | |
|---|---|
| **Test Case ID** | 13 |
| **Title** | Notify Patient |
| **Description** | The User notifies the patient that they have an appointment with them. |
| **Input** | The user logs in and scrolls down to the bookings table. |
| **Steps to Perform** | • The user logs in as a Doctor.<br>• User clicks on the Notify Patient button. |
| **Output** | The respective patient is sent a reminder email |

### 4.6.14 TC-14 Prescribe Patient

Table 4.15: TC-14 Prescribe Patient

| | |
|---|---|
| **Test Case ID** | 14 |
| **Title** | Prescribe Patient |
| **Description** | The User Prescribes a patient that they have consulted recently. |
| **Input** | The user logs in and scrolls down to the bookings table. |
| **Steps to Perform** | • The user logs in as a Doctor.<br>• User clicks on the Prescribe Patient button.<br>• User inputs the prescription for the respective patient. |
| **Output** | The prescription is sent to the prescribed patient. |

### 4.6.15 TC-15 Analyze Earnings

Table 4.16: TC-15 Analyze Earnings

| | |
|---|---|
| **Test Case ID** | 15 |
| **Title** | Analyze Earnings |
| **Description** | The User keeps track of their earnings. |
| **Input** | The user logs in to the system. |
| **Steps to Perform** | • The user logs in.<br>• User sees tabs of total and monthly earnings on the dashboard.<br>• The user scrolls down to the graph.<br>• The line graph shows the comparison of the earning in each month. |
| **Output** | The User can analyze and keep track of their earnings. |

### 4.6.16 TC-16 View Doctors

Table 4.17: TC-16 View Doctors

| | |
|---|---|
| **Test Case ID** | 16 |
| **Title** | View Doctors |
| **Description** | The User views all the doctors registered under their name. |
| **Input** | The user logs in to the system and scrolls down to the doctors table. |
| **Steps to Perform** | • The user logs in.<br>• User scrolls down to the doctors table. |
| **Output** | The Clinic User can view all the doctors registered under their name. |

### 4.6.17 TC-17 Change Password

Table 4.18: TC-17 Change Password

| | |
|---|---|
| **Test Case ID** | 17 |
| **Title** | Change Password |
| **Description** | The User changes their existing password. |
| **Input** | The user logs in to the system and scrolls down to the doctors table. |
| **Steps to Perform** | • The user logs in.<br>• User clicks on the drop down on the dashboard<br>• User clicks on the change password button<br>• User enters the old and new password<br>• the use then presses the change password button |
| **Output** | The User's current password is changed. |

### 4.6.18 TC-18 Update Profile

Table 4.19: TC-18 Update Profile

| | |
|---|---|
| **Test Case ID** | 18 |
| **Title** | Update Profile |
| **Description** | The User updates the information stored inside the system. |
| **Input** | The user is logged in and has already provided the details. |
| **Steps to Perform** | • The user logs in.<br>• User clicks on the drop down on the dashboard<br>• User clicks on the update profile button<br>• User then adds the information that they need to update.<br>• User then clicks on the update profile button. |
| **Output** | The User's information is updated. |

### 4.6.19 TC-19 Upload Medical Record

Table 4.20: TC-19 Upload Medical Record

| | |
|---|---|
| **Test Case ID** | 19 |
| **Title** | Upload Medical Record |
| **Description** | The patient user uploads their medical records to enhance their experience of online consultancy and maintain records. |
| **Input** | The user is logged in as a patient. |
| **Steps to Perform** | • The user logs in.<br>• User clicks on the drop down on the dashboard<br>• User clicks on the Upload Medical Records button<br>• User then clicks on upload files<br>• User then selects the desired file to be uploaded. |
| **Output** | The User's medical record is uploaded. |

### 4.6.20 TC-20 View Patient History

Table 4.21: TC-20 View Patient History

| | |
|---|---|
| **Test Case ID** | 20 |
| **Title** | View Patient History |
| **Description** | The doctor user Views the medical history of a patient before consulting them. |
| **Input** | The user is logged in as a doctor. |
| **Steps to Perform** | • The user logs in.<br>• User clicks on View Patient History Button. |
| **Output** | The User can now view the medical records of the respective patient. |

## 4.7 Functional Testing

Functional Testing phase is performed once all the individuals modules are linked together to perform the complete functioning of the web application as one unit. We also perform integration testing of the system during the functional testing phase. One of the functional testing test cases is logging in and completing two factor verification to view the dashboard. If the user tries to enter the dashboard URL without logging in or completing the two factor verification they are redirected to the login page. Our system first recognises if the user is logged in or not. Once the credentials are matched it goes on to the two factor verification. A four digit One time password is sent to the user's email. The user then enters the One time password, the system compares the entered one time password with the one generated by the system and upon successful matching of the both the user is given the access to the respective Dashboard i.e Patient Dashboard, Doctor Dashboard or Clinic Dashboard.

### 4.7.1 Testing Requirements

The main requirement for the testing procedure is the requirement for the dummy data. Dummy data needs to be entered into the system to ensure the proper working of each component of the web application.Large amount of data must be flooded into the system to ensure the load optimization of the system. The most convenient way of doing it is by running an automated script for the testing purposes that automates several hundred users to access the web application simultaneously.

# Chapter 5

# Conclusion & Future Work

## 5.1 Conclusion

E-Medix is a major turnover in the health care industry.The COVID-19 pandemic has swiftly shifted the paradigm of medical consultancy from in clinic appointments towards online appointments. With this increasing need of online healthcare systems there are only a few service providers. This difference in the demand and supply makes our system a very essential addition to the health care industry. During the pandemic, going to a doctor is the most dangerous thing to do. A doctor's clinic is the place at which person is the most vulnerable to the virus. With our system patients and doctors can stay at their homes and perform their part. A patient can book an online appointment from the comfort of their house and the doctor can consult the patient from the comfort of their house. Once the pandemic is over the in clinic appointments may again get popular amongst the population. For that situation our platform also allows the patient to book in clinic appointments. In future we are looking forward to also implementing an intelligent chat bot into the system that may analyze your symptoms and suggest the best possible doctor for you. It is already a work in progress but due to the detailed and sensitive nature of the work it requires a very large data set for sufficient training to take such decisions. However, in the near future we will

implement the chat bot into the system and it will be available to end user for testing the accuracy. Moreover we are also planning to sign a contract with multiple pharmacies and medical laboratories like Clinix and Chughtai Lab to integrate the online delivery of medicines and booking of home sample collection services within our portal. Once the integration is completed our system will be capable of fulfilling all the health care needs of users at one place. The user will be able to order any prescribed medicine or book home sample collection for any prescribed lab test with just the click of a button.

# Bibliography


[1] Django mvt structure. URL https://www.javatpoint.com/django-mvt.

[2] Impacts of technology on our lives. URL https://thriveglobal.com/stories/how-technology-has-made-an-impact-in-making-our-lives-better/.

[3] Reasons of health conditions in pakistan. . URL https://www.jstor.org/stable/41258785.

[4] Health care facilites in rural areas. . URL https://saylordotorg.github.io/text_social-problems-continuity-and-change/s17-04-problems-of-rural-life.html.

[5] Lack of medical staff in pakistan. URL https://www.crux.pk/2020/04/22/pakistans-lack-of-medical-staff-could-the-healthcare-system-collapse-during

[6] Phases of the waterfall model. URL https://www.geeksforgeeks.org/software-engineering-classical-waterfall-model/.

[7] Doctor's prefer working in urban areas. URL https://ayubmed.edu.pk/JAMC/PAST/16-2/umar.htm.

[8] Software development life cycle. URL https://stackify.com/what-is-sdlc/.

[9] Stages of sdlc. URL https://www.researchgate.net/profile/Adetokunbo_Adenowo/publication/344194737_Software_Engineering_Methodologies_A_Review_of_the_Waterfall_Model_and_Object-_Oriented_Approach/links/5f5a803292851c07895d2ce8/

Software-Engineering-Methodologies-A-Review-of-the-Waterfall-Model-and-Obje pdf.

[10] What is the waterfall model. URL https://searchsoftwarequality.techtarget.com/definition/waterfall-model.

[11] World health organization coronavirus (covid-19) dashboard. URL https://covid19.who.int/.

[12] Health in pakistan. URL https://en.wikipedia.org/wiki/Health_in_Pakistan.