Abstract

In today's life medical records of a person are one the most important documents in his life as they contain really sensitive and important information about their health. So the management and safety of such documents should also be really important. Most of the doctors still examine a patient, take notes from that and note them down on a paper sheet and give them to the patient. And it is really difficult for the patient to manage them and keep them safe at one place. In most of the cases these crucial medical records get misplaced by the patient. But now the time has been changed. Electronic records are taking place of paper records in almost every field of life as they are easier to manage, store and are less likely to get misplaced in contrast of paper records thus making them far more efficient than paper records. This transformation of non-electronic and paper records to electronic records has made their management easy. Our project will be an electronic health record (EHR) system which will provide a central platform where all the medical records of the patient can be stored and managed easily. This will not only reduce the risk of losing important medical records by patients but also help the doctors in the diagnosis of the patient based on their previous records. In most of the cases patients cannot convey their issue correctly to their doctor due to shyness or lack of knowledge which can lead to the false diagnosis and can cause major issues. So here are system will be of real help by providing the previous record of the patient to the doctor for better diagnosis. Communication gap between the two different doctors of a common patient will be filled by the chat feature provided by our system where a doctor can communicate to the previous doctors of the patient in case of any confusion. This project will be a web application developed in React, Django and MongoDB where patients can access their records by signing into their account. This project will provide a great service in field of medical by providing the essential information to the doctors about the patient which they need for an effective diagnosis.

Key words: EHR, React, Django, MongoDB, Management, effective Diagnosis

Chapter 1: Introduction

Not long ago, when a patient visited to a doctor clinic or a hospital, the doctor used to examine patient, take notes from that and note them down on a paper sheet and give them to the patient. But now the time has been changed. Electronic records are taking place of paper records in almost every field of life as they are easier to manage, store and are less likely to get misplaced in contrast of paper records thus making them far more efficient than paper records. This transformation from paper records to an all-electronic office has made the management of the medical records easy now. Here is where Medi Log comes into action as it provides a platform for its users an easy access and management of all of their medical records at one place.

Our Project would be providing services to different hospitals, clinics and laboratories to upload and access the medical record of the patient. All the medical records can be uploaded and stored in form of pdf files and a doctor can access and examine the patient better on the basis of his previous medical records. Moreover, Patients can also access their medical records whenever they want by singing into their account. Medi Log will also be providing a chat feature by which a doctor can chat with the previous doctor of the patient in case of any confusion and a patient can also chat with its doctor for any guidance or confusion. It will also provide a feature of trusted contacts where one patient can add the other contact as its trusted contact who can access that patient’s records. Our project would be providing these services via a web application.

1.1 Goals and Objectives

The goals and objectives of our project are as follows:

- Design and develop a web application to access and manage medical records of a patient by his Doctor.
- Providing a common platform for the doctors and hospitals to store patient records.
- Providing the main information about patient to doctor for an efficient diagnosis.
- This application would be web based only.
- This application will be developed using React, Django and Mongo DB.
- Achieving a full working system in given interval of time.

The subsequent part of this report would be about the literature review and the summary of similar work, requirements and design, implementation and a conclusion

1.2 Scope of the Project

Main use of our Project is clearly in the medical field. This system will be deployed in different hospitals and laboratories so that all type of medical records of a patient can be stored at one place. This system will be really helpful for doctor in better diagnosis of the patient. Currently in Pakistan there is no such system for maintaining the complete medical records of a patient. Some hospitals do have their own systems to maintain their patient records but when that patient visits another doctor or hospital then they cannot access their previous medical records. Here this system will be really helpful as it will provide a central platform where different hospitals, laboratories and doctors which have this system deployed will be able to access all the previous medical records of their patient uploaded by other doctors or laboratories.

This document contains a complete overview of literature study, implementation strategies along with all the diagrams necessary to understand the initial requirements and working of the project. Chapter 2 contains a brief discussion about the research we have done online about our project and it contains some systems similar to our project which are currently working. Chapter 3 contains all the functional requirements and design of our project. It contains all the required diagrams like use cases, Sequence and database design diagrams which would be helpful to understand the working and flow of our project. Chapter 4 describes about the implementation and test cases which have been done so far by our team.

Chapter 5 contains the details about the experiments we have done on our so far implemented system and the results of these experiments.

Chapter 2: Literature Survey / Related Work

2.1 EMR vs EHR

An EMR is an electronic medical record and an EHR is an electronic health record. Both can be a piece of clinical records management. An EMR is typically a record inside a singular clinic or hospital. An EHR, nevertheless, is more thorough, and the patients can access it across almost all types of health organizations including clinics, labs, hospitals and pharmacies etc. Ordinarily, EHRs are moveable with patients but the EMRs are not. Taking everything into account an EHR is far away better and successful in dealing with the entire medical history of a patient.

2.2 Work flow of an EHR

Supervision of clinical records identifies the activity of a medical care practice. Medical records all over their lifecycle are a responsibility of management field. This field oversee these documents from their creation, receipt, maintenance, and use to disposal. Now and again alluded to as Health information management (HIM) or Health records information management (HRIM), clinical records administration can include everything without exception about a practice and a patient. This can include whole patient medical history, clinical discoveries, different test reports and their results, medical status of a patient and medications.

2.3 Information an EHR contains

An electronic health record (EHR) system is not just a computerized version of a prescription that a doctor gives to his patient. It is a digital medical record that provides the comprehensive information about the health of a patient.

Main propose of developing an EHR is to share all the information of a single patient among all the healthcare organizations which include Hospitals, Clinics, Laboratories, Pharmacies, Emergency facilities, medical imaging Facilities on a single platform to ensure an effective diagnosis of the patient. An ER can contain following information about the patient health conditions.

- Progress Notes
- Billing and administrative data
- Vital signs
- Patient demographics
- Allergies
- Medication
- Immunization dates
- Test results and Lab reports
- Radiology images
- Diagnoses
- Medical Histories

2.4 Specifications and Advantages

Both Patient and doctors can get benefit through this system. For the patient, the record can diagram one's set of experiences and treatment plan in an effectively available manner. For the doctor, it can offer help about the accuracy of that treatment plan. A decent medical record management system can be the contrast among life and passing in the event that somebody needs to check a test, a drug, or settle on a brisk clinical choice.

Following are some advantages of an EHR which will explain Why Electronic health records are so necessary in medical fields.

- An EHR provides complete updated and correct medical information about the patient at the point of care
- It can make a diagnosis more efficient and coordinated by providing the quick access to the patient’s medical records.
- It avoids the lack of communication between patient and doctor by providing a complete medical record of a patient.
- It enhances the privacy of the patient data by providing access to only patient and his doctor
- It reduces the chances of misplacement of important medical records.
- Reduces cost by decreasing paper work and improves efficiency by reducing the duplication of testing and improving security.

2.5 Related EHR Systems

We studied a few Medical/Health Record systems and what services they are providing they all differ in one way or another but the ones we found closest to our FYP Idea are as follows.

2.5.1 M-Files

M-Files is a software which provides the service to store and manage all type of documents on a single platform. It basically arranges the documents on the bases of what they relate to and keep them on a same place which is really helpful as it saves a lot of time spent on arranging the documents. User can install this software on their system and can access their documents anywhere and anytime as it can be deployed on cloud. Due to its simple, dynamic and flexible environment you can manage a large variety of documents easily. The medical record management service of the m-files provides a very easy method of organizing the medical documents by giving them a property such as by patient, physician or by date. It can be really helpful for managing and storing any kind of documents at a single platform.

M-files contains following components:

- **Setup:** This will be needed to install the software at user end system
- **Client:** This is the most common used component of the software which is tightly integrated into the windows and is used to display content in different views.
- **Client Settings:** Purpose of this component is to form a connection between the client computer and the vaults formed on the online servers of m-files. A feature of editing the local settings like changing the drive letters has also been provided.
- **Server:** this component manages the storage and sharing of the user’s documents.
- **Server Administrator:** This tool is used by the m-files company to change and modify the Structure of vaults, Server settings and manage the document vaults.
- **Show Status:** This component provides the monitoring of the status of file transfer. This feature can be helpful for the users using the software on a slow internet connection
- **Web Access:** A user can also access the m-files online through your browser instead of using the m-files client
- **Automatic Updates:** This feature of the software keeps your m-files up to date and let you use the updated features.

Overall m-files is a good software to manage any type of documents easily and access them anywhere anytime.

2.5.2 Epic EHR

Epic EHR is a cloud based electronic health record system providing some features to cater the medical record management problems. Several health organizations like independent clinics community hospitals and several multi hospital groups are currently using this software to manage the health records of their patients. Epic provides the medium range of basic EHR functionalities and flexibility of adding features on the basis of different specialty. Main focus of epic now is to engage its patients and provide mobile medical care to its patients.

Epic EHR have different portals for web, android and IOS to cover all type of users and allowing their patients to manage their healthcare requirements with more flexibility. Epic EHR is cloud-based, so accessible on any gadget with a web program introduced. Local applications are accessible for iOS and Android working frameworks.

Following are the features which are currently provided by the epic EHR:

- Appointment Management
- Management of billing documents
- Management of Clinical Workflow
- Management of medical documents
- Security and maintainability
- Integration with laboratories
- Medical figures and diagrams
- Stats of a patient
- Total medical history of a patient
- Separate portal for patients
- Benchmarks
- Reporting and Analytics
- Organizing the documents
- Recognizing voice of user
- Electronic receipts and prescriptions

More than 250 million patients as of now have an EHR with Epic. Patients who utilize Epic's MyChart Companion from home can plan arrangements, complete surveys, and become more learned about their own wellbeing information. Furthermore, patients utilizing MyChart Companion in a medical services office can keep in contact with their consideration group and survey their timetable and patient materials.

In the midst of the COVID pandemic, Epic and Cleveland Clinic appeared a far-off observing application to make it workable for clinicians to follow changes in COVID patients' side effects from a good way —cementing Epic as a top EHR arrangement.

2.5.3 Cerner EHR

Cerner EMR is a product coming from a bunch of e-health record frameworks and other clinical arrangements. The purpose behind the tool is to smooth out clinical work process for medical care staff and clinical specialists, improve patient safety, and to reduce down on expenses. This system offers a bunch of strong devices and functionalities for medical care organizations and clinical experts. Cerner EMR Reviews gives nearby/offsite facilitating for use in any medical services association paying little heed to measure. The product intends to improve quiet consideration coordination and documentation by giving clinical data admittance to medical services suppliers. This can help them concoct arrangements just as settle on the correct choices in advancing patient security. The product is utilized to so patients are engaged to stay proactive with their wellbeing as they reliably collaborate with

physicians and medical services suppliers. Moreover, Cerner EMR is completely consistent with HIPAA.
At the start of 2019, Cerner had control of more than one-fourth of the EHR market for US clinics — together, Cerner and Epic make up 54% of the EHR market for US medical clinics. Remarkably, in 2019 Business Insider Intelligence detailed that Cerner joined forces with Uber Health, permitting suppliers to effortlessly plan nonemergency transportation for patients.

Additionally, in 2019, Cerner made an arrangement with Amazon Web Services (AWS), where AWS gave Cerner a cloud-based stage to help customers open prescient experiences for upgrading the nature of care. What's more, in 2020 the two organizations collaborated again to help make COVID information accessible to analysts.

These were the few EHR systems currently working which we found related to our project.

Chapter 3: Requirements and Design

3.1 Functional Requirements

3.1.1 Functional Requirements for Administration

System will allow admin to:

- Create verified accounts for new users (Doctors, Labs and Patients).
- Verify existing accounts.
- Upload existing medical records for patients.
- Suspend accounts in case of any misbehave.
- Grant special access to doctors in case a patient can't verify.
- Edit user profile.
- Reset passwords.

3.1.2 Functional Requirements for Doctors

System will allow doctor to:

- Add new check-up records.
- View a patient's previous medical history.

Contact the previous doctors of a particular patients.

3.1.3 Functional Requirements for Laboratories

System will allow laboratories to:

Add medical test reports/scans of a patient.

3.1.4 Functional Requirements for Patients

System will allow patients to:

- Create their account.
- Edit user profile (for non-verified users).
- View all kinds of personal medical records.
- Add/update trusted contact (A person who can access your record, maybe Parent/Guardian/Partner etc).
- View trusted contact record.

Be a trusted contact of one or more users, i.e., father of 3 children can be trusted contact of all of them and his wife too.

3.2 Non-Functional Requirements

3.2.1 Performance

- Maximum allowed data limit for each patient will be 200 MB.
- System would be capable of at least 150 users at a time.

The application would load within 3-5 seconds due to optimization techniques available in react and node but this duration may vary due to internet speed.

3.2.2 Sustainability

- This system would be accessible from any browser with the given hardware specifications.

It will be a responsive web app and would behave as expected on all devices.

3.2.3 Usability

System will rely on HCI concepts and conventions to make it super easy to use.

3.2.4 Reliability

System would have an uptime of near to 100% due to the very reliable servers.

3.2.5 Reusability

We will be using modern modularization and design pattern techniques to make our code readable and reusable.

3.3 Hardware and Software Requirements

3.3.1 Hardware Requirements

For development, the hardware requirements are the following:

- Desktop with 8GB RAM and 64-bit operating system.

For usage, the hardware requirements are the following:

- Web Browser (PC or Smart Phone).
- Finger Print Scanner (For Administration only).

3.3.2 Software Requirements

For development, the software requirements are the following:

- Code editor: Visual Studio code, Sublime Text, Atom.
- MongoDB Atlas Database.
- Express: WEB application Framework.
- React: A JavaScript library for Frontend.
- Node: JavaScript runtime environment.
- Node Package Manager (For installing dependencies)

Browser: Google chrome, Mozilla Firefox, Microsoft Edge, Safari.

3.4 System Architecture

3.4.1 High Level System Architecture

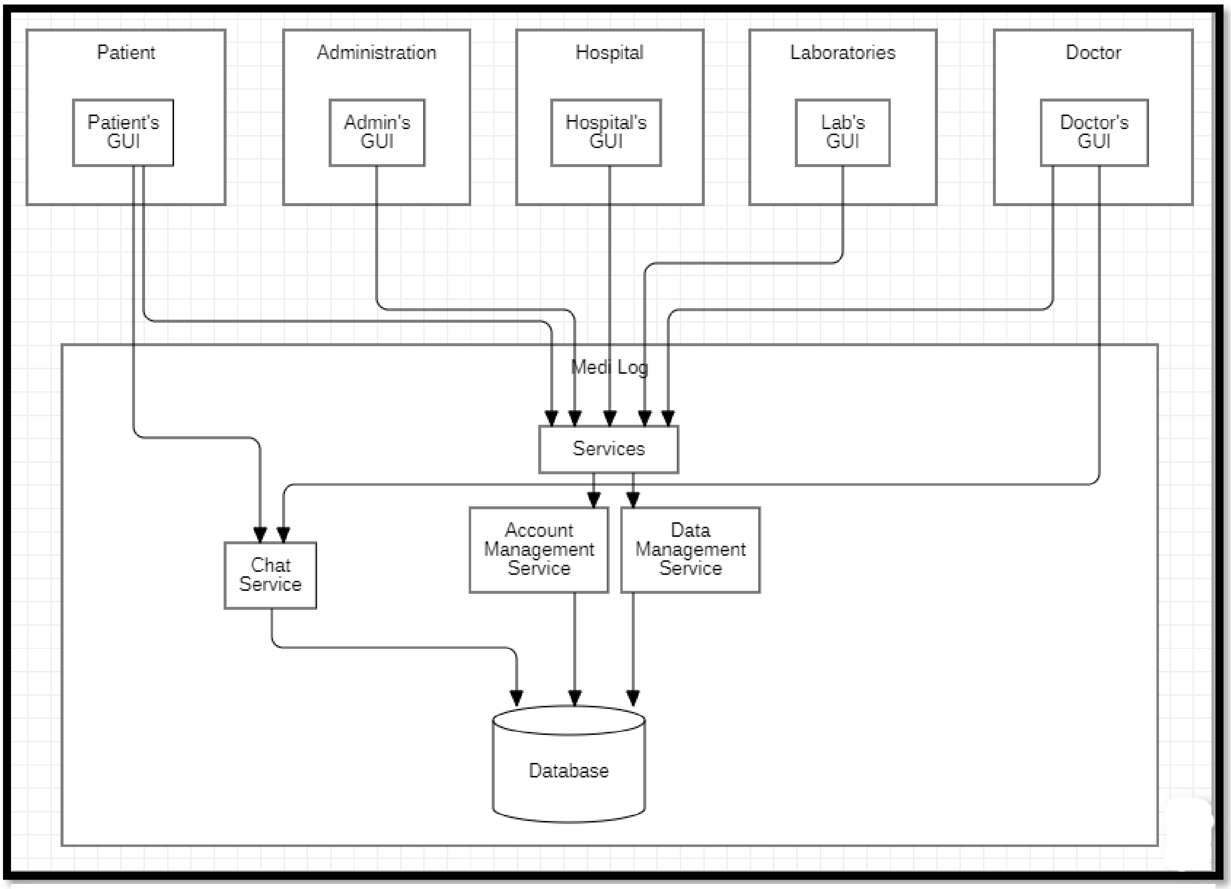


Figure 1 : High Level System Architecture
Diagram for high level system Architecture

As shown in the high-level architecture diagram this application has a client server structure where users send request to perform their functions. We have five categories of users namely Administration, Patient, Doctor, Hospital, Laboratory. Server consists of 3 modules Accounts management service, Data management service and chat module. Users interact with the server via User Interface and that request is forwarded to the server. Afterwards server responds to the request as necessary and returns the response to the user via User Interface. Server also communicates with the database if necessary.

3.4.2 Low Level System Architecture

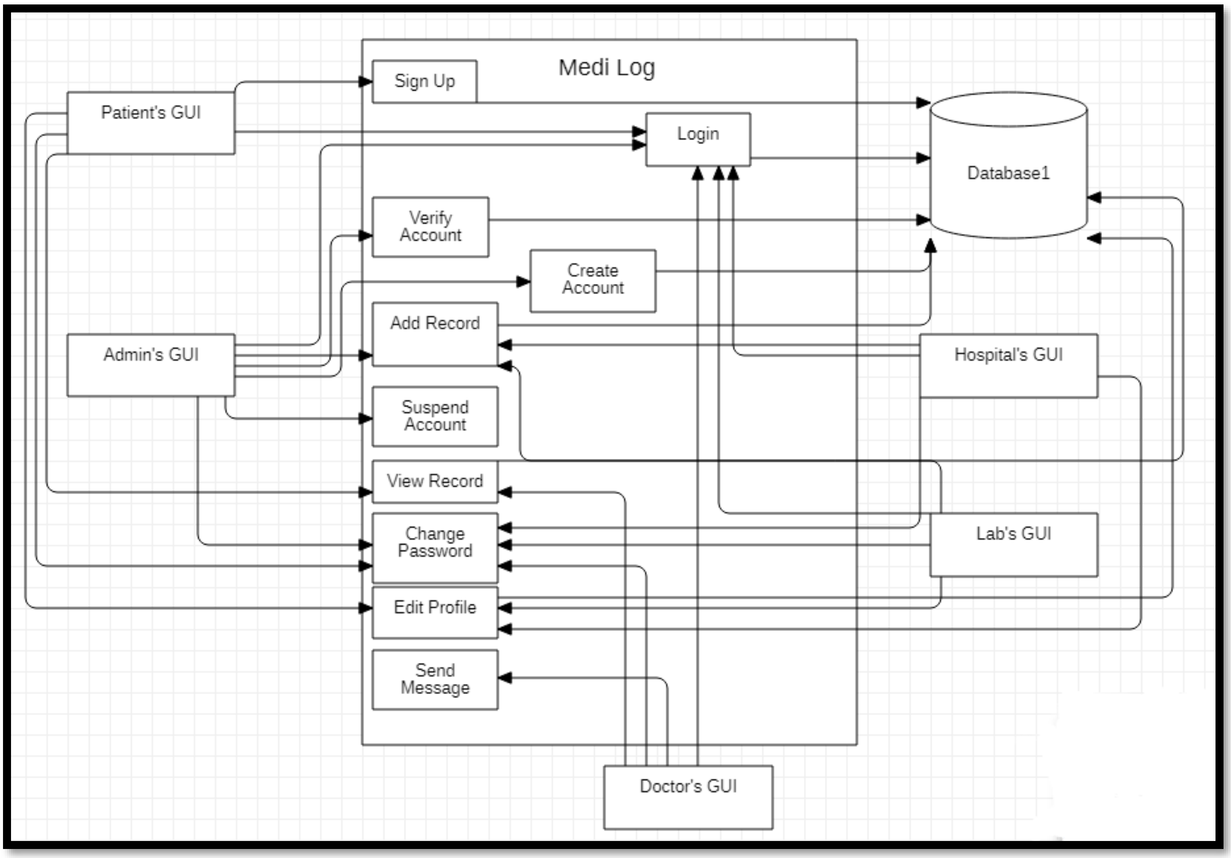


Figure 2 : Low Level Architecture
Low Level Architecture diagram with all modules

3.5 Architectural Strategies

3.5.1 React, Django and Mongo

This goes without any doubt that React is most famous and is being widely used in front-end development. This is not just a co-incidence this is because react has so much to offer that nobody wants to ignore. The main benefits for using react in front-end development include but are not limited to

- Easy to learn
- SEO Friendly
- Speed
- Flexibility
- Performance
- Usability
- Strong community support.

Now moving to our second decision of using Django as our back-end programming language we had multiple options mainly Django and Node Js we chose Django over Node Js for various reasons which include

- Security
- Performance
- Loose coupling

- Efficiency
- Reputation
- Easy to learn

The third decision is to use MongoDB instead of SQL. It is an open-source platform NoSQL database developed in C++. This also a well thought decision and there are also many reasons behind this which are stated below

- High Performance
- High Scalability
- High Availability
- High Flexibility

3.5.2 Future Plans for Enhancing Software

Currently this project is only going to target the already selected features and actions. However, in future we can implement new options and may roll out an update but currently we don't plan to do so yet it is possible.

3.5.3 User Interface Paradigms

We have been studying different principles and guidelines on building effective and easy to understand and easy to use user interfaces and we aim to follow the Nielsen and Molich's User Interface Design Guidelines and Joshua Porter's principles of User Interface Design.

These include

- User control and freedom.
- Consistency and standards
- Error prevention.
- Recognition rather than recall.
- Aesthetic and minimalist design.
- Flexibility and efficiency of use.
- Visibility of system status
- One primary action per screen
- Provide a natural next step

3.6 Use Cases

3.6.1 Signup

Name	UC-1		
Actors	Patient		
Summary	**User** should enter the information of new patient like CNIC and Passwords and a new patient account should be created and redirect to home page of patient account.		
Pre-Conditions	User must have an active internet connection and be on home page of the web application.		
Post-Conditions	New patient account should be created and his information should be added to database.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	User clicks signup or new patient button on the screen.	2	A form is displayed asking for patient information like CNIC, credentials and other personal info.
3	The user provides the required info and click create account button.	4	The system processes the provided info, add it to data base, establishes a session for patient and redirect him to home page of patient account.
Alternative Flow			
3	The user enters info of patient.	4-A	The system responds with an error message: *Database not connected.*
		4-B	The system responds with an error message: *Invalid CNIC.*
		4-C	The system responds with an error message: *Passwords don’t match.*
		4-D	The system responds with an error message: *An information field is missing.*
		4-E	The system responds with an error message: *Invalid contact number.*
		4-F	The system responds with an error message: *Patient already exists.*

3.6.2 Verify account:

<table>
<tr><td><b>Name</b></td><td colspan="3">UC-2</td></tr>
<tr><td><b>Actors</b></td><td colspan="3">Admin</td></tr>
<tr><td><b>Summary</b></td><td colspan="3">Admin search a patient account by entering username or CNIC and verify it.</td></tr>
<tr><td><b>Pre-Conditions</b></td><td colspan="3">Actor must be logged in with an admin account and connected to internet.</td></tr>
<tr><td><b>Post-Conditions</b></td><td colspan="3">Patient account should be marked as verified in database and a notification of success pops up on the screen.</td></tr>
<tr><td><b>Special Requirements</b></td><td colspan="3">None</td></tr>
<tr><td colspan="4"><b>Basic Flow</b></td></tr>
<tr><td colspan="2"><b>Actor Action</b></td><td colspan="2"><b>System Response</b></td></tr>
<tr><td>1</td><td>User searches the patient account by entering his CNIC in search bar and entering the OTP sent to his mobile or by scanning his fingerprint.</td><td>2</td><td>Patient account with that username or CNIC shows up on the screen.</td></tr>
<tr><td>3</td><td>Admin clicks on verify account.</td><td>4</td><td>System marks the patient account as verified and shows a notification saying account verified successfully.</td></tr>
<tr><td colspan="4"><b>Alternative Flow</b></td></tr>
<tr><td>1</td><td>User searches the patient account by entering his CNIC in search bar and entering the OTP sent to his mobile or by scanning his fingerprint.</td><td>2-A</td><td>The system responds with an error message: <i>there is no account with such username or CNIC.</i></td></tr>
<tr><td></td><td></td><td>2-B</td><td>The system responds with an error message: <i>Invalid OTP or fingerprint</i></td></tr>
<tr><td>3</td><td>Admin searches patient by entering its username or CNIC.</td><td>4-A</td><td>The system responds with an error message: <i>Database not connected</i></td></tr>
</table>

3.6.3 Create Accounts:

Name	UC-3
Actors	Admin
Summary	Admin clicks on add new user and select doctor, clinic/hospital or lab and then enters info and click create account. Account info is stored in database and a notification of success is showed up.
Pre-Conditions	Actor must be logged in with an admin account and have an active internet connection.
Post-Conditions	Account info must be stored in data base and a success notification showed up on screen.
Special Requirements	None

Basic Flow			
Actor Action		**System Response**	
1	Admin clicks on add a new user account button.	2	Three to four options show up i.e. hospital/clinic, patient, doctor, laboratory
3	Admin selects an option from the four options.	4	An entry form shows up requiring info of the selected user.
5	Admin fills the form and clicks on create account.	6	System pops up a notification saying that account created successfully.
Alternative Flow			
5	Admin fills the form and clicks on create account.	6-A	The system responds with an error message: *Database not connected.*
		6-B	The system responds with an error message: *Account already exists.*
		6-D	The system responds with an error message: *Passwords don’t match*
		6-E	The system responds with an error message: *An information field is missing*

3.6.4 Login:

<table>
<tr><td><b>Name</b></td><td colspan="3">UC-4</td></tr>
<tr><td><b>Actors</b></td><td colspan="3">Admin, Doctor, Clinic/Hospital, laboratories, Patient</td></tr>
<tr><td><b>Summary</b></td><td colspan="3">Actor enters the credentials; System verifies the credentials and redirects the user to their home screen.</td></tr>
<tr><td><b>Pre-Conditions</b></td><td colspan="3">User must have an active internet connection.</td></tr>
<tr><td><b>Post-Conditions</b></td><td colspan="3">A session must be established for respective user and redirect him to his home screen.</td></tr>
<tr><td><b>Special Requirements</b></td><td colspan="3">None</td></tr>
<tr><td colspan="4"><b>Basic Flow</b></td></tr>
<tr><td colspan="2"><b>Actor Action</b></td><td colspan="2"><b>System Response</b></td></tr>
<tr><td>1</td><td>User enters the credentials on login page and clicks login.</td><td>2</td><td>System verifies the credentials, establishes an active session and redirect user to its homepage.</td></tr>
<tr><td colspan="4"><b>Alternative Flow</b></td></tr>
<tr><td>1</td><td>User enters the credentials on login page and clicks login.</td><td>1-A</td><td>The system responds with an error message: <i>Invalid username or password.</i></td></tr>
</table>

3.6.5 View other’s profile

<table>
<tr><td><b>Name</b></td><td colspan="3">UC-15</td></tr>
<tr><td><b>Actors</b></td><td colspan="3">Admin</td></tr>
<tr><td><b>Summary</b></td><td colspan="3">Actor can view personal info of any user.</td></tr>
<tr><td><b>Pre-Conditions</b></td><td colspan="3">Actor must be logged in with an admin account.</td></tr>
<tr><td><b>Post-Conditions</b></td><td colspan="3">Personal info of respective user will be displayed.</td></tr>
<tr><td><b>Special Requirements</b></td><td colspan="3">None</td></tr>
<tr><td colspan="4"><b>Basic Flow</b></td></tr>
<tr><td colspan="2"><b>Actor Action</b></td><td colspan="2"><b>System Response</b></td></tr>
<tr><td>1</td><td>Admin will search the account of user by entering his CNIC or user name.</td><td>2</td><td>Account of user with its details will be shown on the screen.</td></tr>
<tr><td colspan="4"><b>Alternative Flow</b></td></tr>
<tr><td>1</td><td>Admin will search the account of user by entering his CNIC or user name.</td><td>1-A</td><td>The system responds with an error message: <i>Incorrect CNIC or username.</i></td></tr>
</table>

3.6.6 Add medical record:

<table>
<tr><td colspan="2"><b>Name</b></td><td colspan="2">UC-5</td></tr>
<tr><td colspan="2"><b>Actors</b></td><td colspan="2">Hospital/Clinic, Laboratory, Admin</td></tr>
<tr><td colspan="2"><b>Summary</b></td><td colspan="2">User Uploads the checkup or laboratory report in form of pdf in account of the respective patient. System store that file in database and shows the success notification.</td></tr>
<tr><td colspan="2"><b>Pre-Conditions</b></td><td colspan="2">User must be logged in with a Hospital/Clinic or laboratory account and patient’s account must be verified.</td></tr>
<tr><td colspan="2"><b>Post-Conditions</b></td><td colspan="2">Selected report should be added to database and should show up in account of that patient.</td></tr>
<tr><td colspan="2"><b>Special Requirements</b></td><td colspan="2">None</td></tr>
<tr><td colspan="4"><b>Basic Flow</b></td></tr>
<tr><td colspan="2"><b>Actor Action</b></td><td colspan="2"><b>System Response</b></td></tr>
<tr><td>1</td><td>User searches the patient account by entering his CNIC in search bar and entering the OTP sent to his mobile or by scanning his fingerprint.</td><td>2</td><td>System shows the details of account of that patient.</td></tr>
<tr><td>3</td><td>User goes to the medical records and click on upload records.</td><td>4</td><td>System shows option to upload pdf file from the current PC on the screen.</td></tr>
<tr><td>5</td><td>User Selects the files from his system and clicks on the upload button.</td><td>6</td><td>System stores the selected files in database and shows the success notification.</td></tr>
<tr><td colspan="4"><b>Alternative Flow</b></td></tr>
<tr><td>1</td><td>User searches the patient account by entering his CNIC in search bar and entering the OTP sent to his mobile or by scanning his fingerprint.</td><td>2-A</td><td>The system responds with an error message: <i>Incorrect Credentials</i></td></tr>
<tr><td></td><td></td><td>2-B</td><td>The system responds with an error message: <i>Invalid OTP or fingerprint</i></td></tr>
<tr><td>3</td><td>User goes to the medical records and click on upload records.</td><td>3-A</td><td>The system responds with an error message: <i>Incorrect file format please select a .pdf file.</i></td></tr>
<tr><td>5</td><td>User Selects the files from his system and clicks on the upload button.</td><td>5-A</td><td>The system responds with an error: <i>Database not connected.</i></td></tr>
</table>

3.6.7 View medical record

Name	UC-6		
Actors	Admin, Patient, Doctor		
Summary	User can view the medical records by going to the medical records tab on the respective patient's account.		
Pre-Conditions	User must have an access to that patient's account and patient's account must be verified.		
Post-Conditions	User can view all the medical history of the patients which includes all the checkup reports and medical reports added by different clinics/hospitals or laboratories.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	User searches the patient account by entering his CNIC in search bar and entering the OTP sent to his mobile or by scanning his fingerprint. If the user is logged in with a patient account the he can simply click on medical records tab on home screen of his account.	2	System shows the details of account of that patient.
3	User clicks on the medical records tab.	4	System shows all the medical records of that patient uploaded by different doctors or laboratories.
Alternative Flow			
3	User searches the patient account by entering his CNIC in search bar and entering the OTP sent to his mobile or by scanning his fingerprint. If the user is logged in with a patient account the he can simply click on medical records tab on home screen of his account.	2-A	The system responds with an error message: *Incorrect Credentials*
		2-B	The system responds with an error message: *Invalid OTP or fingerprint*

3.6.8 Suspend account

<table>
<tr><td><b>Name</b></td><td colspan="3">UC-7</td></tr>
<tr><td><b>Actors</b></td><td colspan="3">Admin</td></tr>
<tr><td><b>Summary</b></td><td colspan="3">Actor can suspend the account of any type of user in case of any misbehavior.</td></tr>
<tr><td><b>Pre-Conditions</b></td><td colspan="3">Actor must be logged in with an admin account.</td></tr>
<tr><td><b>Post-Conditions</b></td><td colspan="3">Account of that user will be restricted and he cannot use his account further but only admin can view his account.</td></tr>
<tr><td><b>Special Requirements</b></td><td colspan="3">None</td></tr>
<tr><td colspan="4"><b>Basic Flow</b></td></tr>
<tr><td colspan="2"><b>Actor Action</b></td><td colspan="2"><b>System Response</b></td></tr>
<tr><td>1</td><td>Admin will search the account of respective user by entering his CNIC or username.</td><td>2</td><td>Al the details of that user will be shown on the screen.</td></tr>
<tr><td>3</td><td>Admin will click on the suspend account button.</td><td>4</td><td>Account of that user will be suspended and in database that account will be added to suspended account table.</td></tr>
<tr><td colspan="4"><b>Alternative Flow</b></td></tr>
<tr><td>3</td><td>Admin will search the account of respective user by entering his CNIC or username.</td><td>4-A</td><td>The system responds with an error message: <i>Incorrect credentials.</i></td></tr>
</table>

3.6.9 Change Password

Name	UC-8		
Actors	Admin, Patient, Doctor, Clinic/Hospital, Laboratories.		
Summary	Password of the respective account will be changed.		
Pre-Conditions	User must be logged in and connected to internet.		
Post-Conditions	The password of his account must be changed in database		
Special Requirements	Old password must be provided		
Basic Flow			
Actor Action		**System Response**	
1	The user clicks Change password.	2	The System prompts for old password, new password and confirm password.
3	The user provides old password and new mandatory password and confirm password field.	4	Changes in Database are made and a new password is stored. A success popup appears.
Alternative Flow			
3	The user provides old password and new mandatory password and confirm password field.	4-A	The system responds with an error message: *Incorrect old password*

3.6.10 Reset password

Name	UC-9		
Actors	Admin		
Summary	Admin will reset password of any account in case they have forgotten their password.		
Pre-Conditions	Actor must be logged in with an admin account.		
Post-Conditions	Password of that user will be changed and modified in database.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	Admin will search the account of user by entering his CNIC or user name.	2	Account of user with its details will be shown on the screen.
3	Actor will click on the reset password tab.	4	The System prompts for new password and confirm password.
5	The user provides new mandatory password and confirm password field.	6	Changes in Database are made and a new password is stored. A success popup appears.
Alternative Flow			
3	The user provides new mandatory password and confirm password field.	4-A	The system responds with an error message: *Passwords does not match.*

3.6.11 Edit own profile

Name	UC-10		
Actors	Admin, Patients, Doctors, Clinic/Hospitals, Laboratories		
Summary	Profile info of the user’s account will be changed which he wants.		
Pre-Conditions	User must be logged into his account.		
Post-Conditions	Modifications in database will be made according to change in information.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	User clicks on edit profile button.	2	User profile with all his information appears.
3	User changes the personal information which he wants and click save.	4	Changes will be made in database and a success notification popup appears.
Alternative Flow			
3	User changes the personal information which he wants and click save.	4-A	The system responds with an error message: *Incorrect information.*

3.6.12 View own profile

Name	UC-14		
Actors	Admin, Patient, Clinic/Hospital, Laboratory		
Summary	User can view his personal info in his profile.		
Pre-Conditions	User must be logged in into his account.		
Post-Conditions	Personal info user will be displayed on screen.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	User will click on the profile section on his screen.	2	Personal info of the user will be displayed on the screen.

3.6.13 Edit other’s profile

Name	UC-11		
Actors	Admin		
Summary	Profile info of any user’s account will be changed which he wants.		
Pre-Conditions	Actor must be logged in with an admin account.		
Post-Conditions	Modifications in database will be made according to change in information.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	Admin will search the account of user by entering his CNIC or user name.	2	Account of user with its details will be shown on the screen.
3	User changes the personal information which he wants and click save.	4	Changes will be made in database and a success notification popup appears.
Alternative Flow			
1	Admin will search the account of user by entering his CNIC or user name.	1-A	The system responds with an error message: *Incorrect CNIC or username.*
3	User changes the personal information which he wants and click save.	4-A	The system responds with an error message: *Incorrect information.*

3.6.14 View trusted contacts

Name	UC-13		
Actors	Patient, Admin		
Summary	User can view his trusted contact.		
Pre-Conditions	User must be logged in.		
Post-Conditions	Info of the trusted contact of patient will be displayed on screen,		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	User clicks on trusted contact tab on his screen.	2	Trusted contact profile card will show up on his screen
3	User clicks on the profile card.	4	Trusted contact information will show up on the screen.

3.6.15 Edit trusted Contact

Name	UC-12		
Actors	Patient, Admin		
Summary	Actor can add another patient as his trusted contact who can view his medical		
Pre-Conditions	Actor must be logged in and the other patient must have a verified account.		
Post-Conditions	Changes will be made to data base and trusted contact can view medical record of the patient who trusts him.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	User clicks on trusted contact tab on his account and then click add/Update.	2	System prompts for CNIC of the trusted contact.
3	User will enter the CNIC of the person whom he wants to be his trusted contact.	4	System will notify the respective person giving him option either to accept or decline.
5	Other patient will either accept or decline the request.	6	System will modify the database according to the action taken.
Alternative Flow			
3	User will enter the CNIC of the person whom he wants to be his trusted contact.	4-A	The system responds with an error message: *Incorrect CNIC*

3.6.16 Search Medical Record

Name	UC-16		
Actors	Admin, Doctor, Patient		
Summary	User can search a specific medical record in his total medical history by its name.		
Pre-Conditions	User must be logged in.		
Post-Conditions	Specific medical record which user searched will be displayed on the screen.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	User will click on medical records tab on his screen.	2	A list of all his medical records will be displayed.
3	User will enter the name of record which he wants to search in the search bar.	4	The Searched document will be displayed on the screen.
Alternative Flow			
3	User will enter the name of record which he wants to search in the search bar.	4-A	The system responds with a message: *There are no medical records with this name.*

3.6.17 Send message

Name	UC-17		
Actors	Patient, Doctor		
Summary	Patient can chat with doctor if he needs any help. Or Doctor can chat with the previous doctors of a patient.		
Pre-Conditions	User who is sending message must be logged in.		
Post-Conditions	Message should be delivered to the respective user.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	User will click the chat option.	2	A list of chats and the doctors he visited/ Patients he attended will appear on screen.
3	User will select a previous chat and send message or start a new chat with another user.	4	Message will be delivered to the respective user.

3.6.18 View Connections

Name	UC-18		
Actors	Patient, Administration		
Summary	Patient can view the people's profile who have marked them as trusted contact.		
Pre-Conditions	Patient must be trusted contact of the other patient.		
Post-Conditions	Profile of the connections will be displayed on the screen.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	User will click the view connections tab	2	A list of all his connections will be displayed.
3	User will select a profile from the list.	4	Profile info selected connection will be displayed.

3.7 GUI

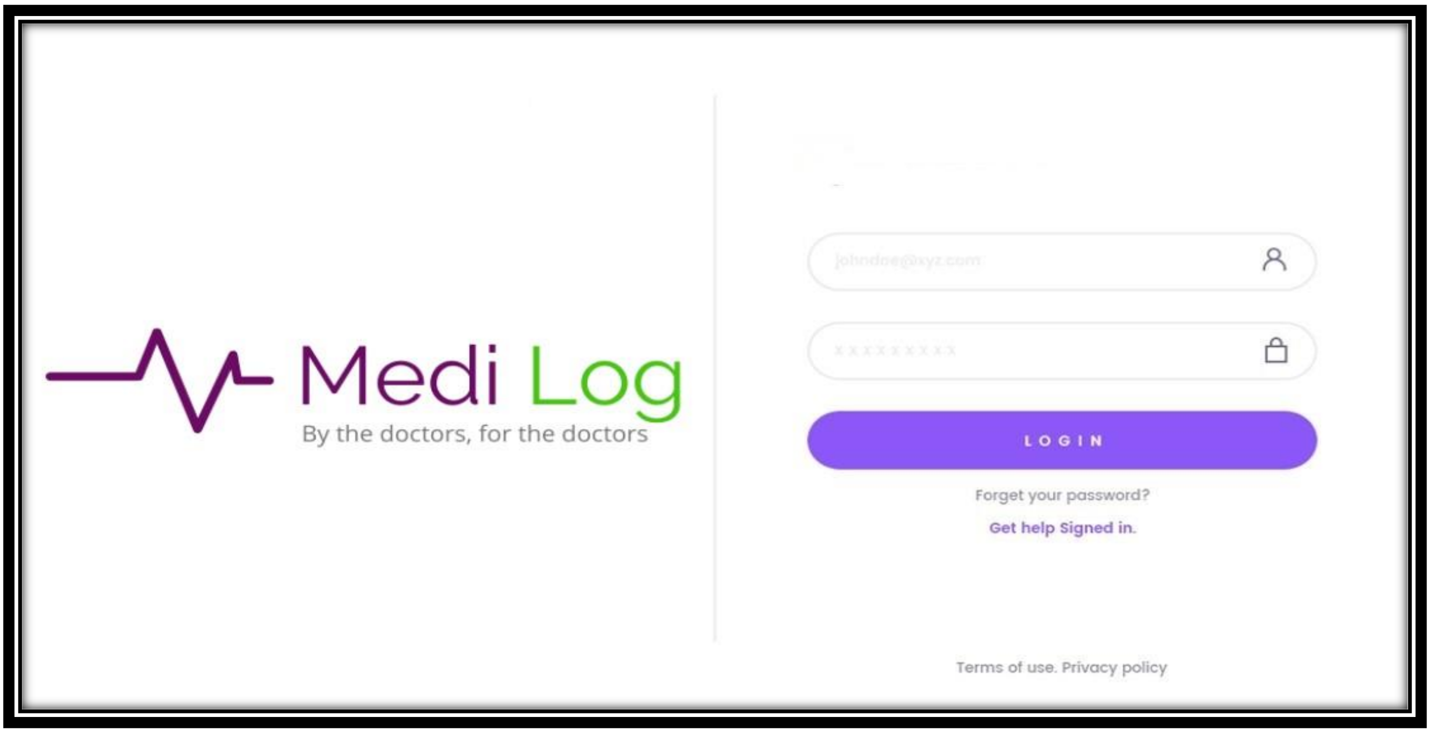
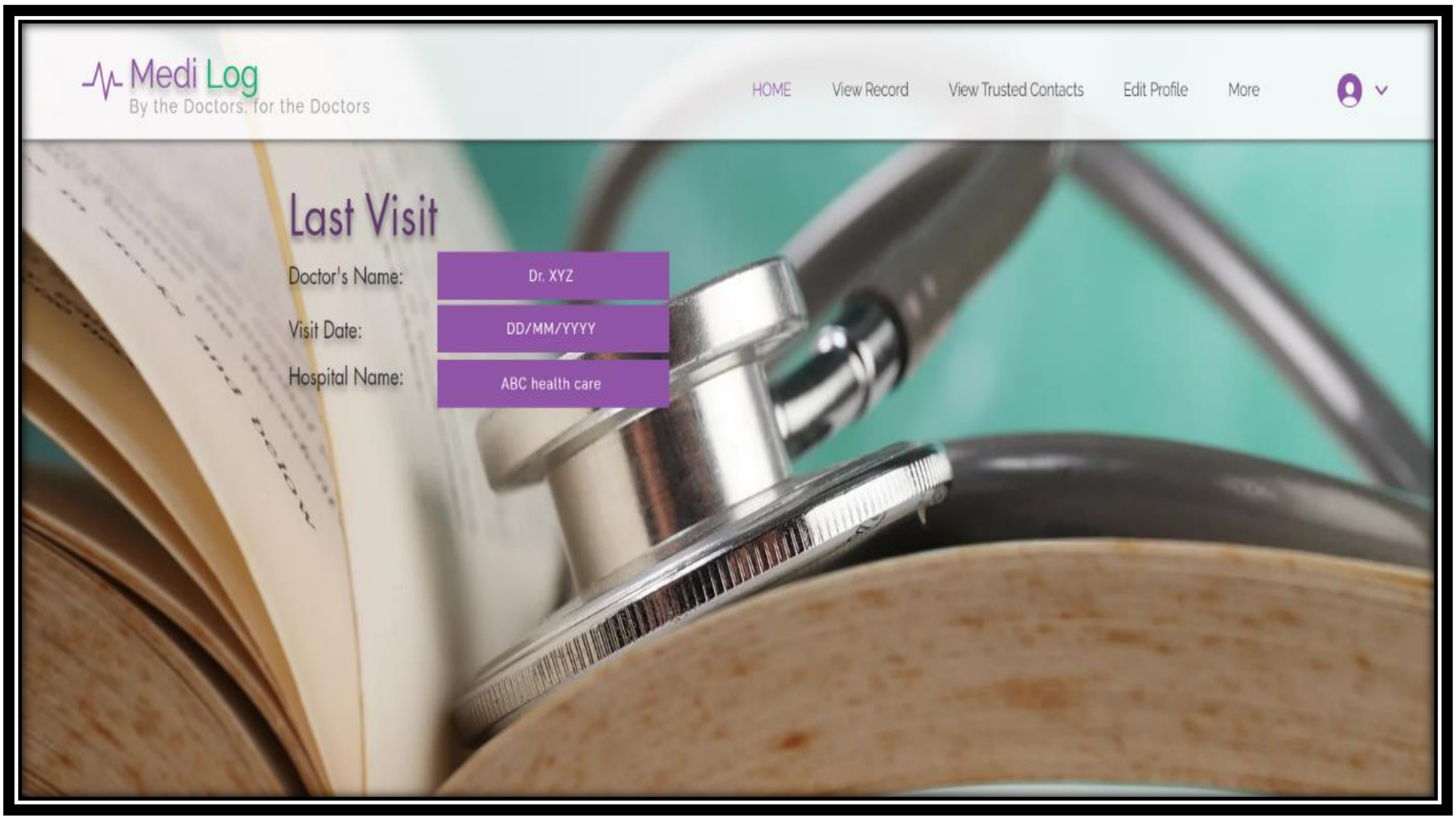


Figure 3 : Login Page
GUI for user login



Last Visit

Doctor's Name: Dr. XYZ

Visit Date: DD/MM/YYYY

Hospital Name: ABC health care

Figure 4 : Patient Home Page View
GUI for patient home screen view

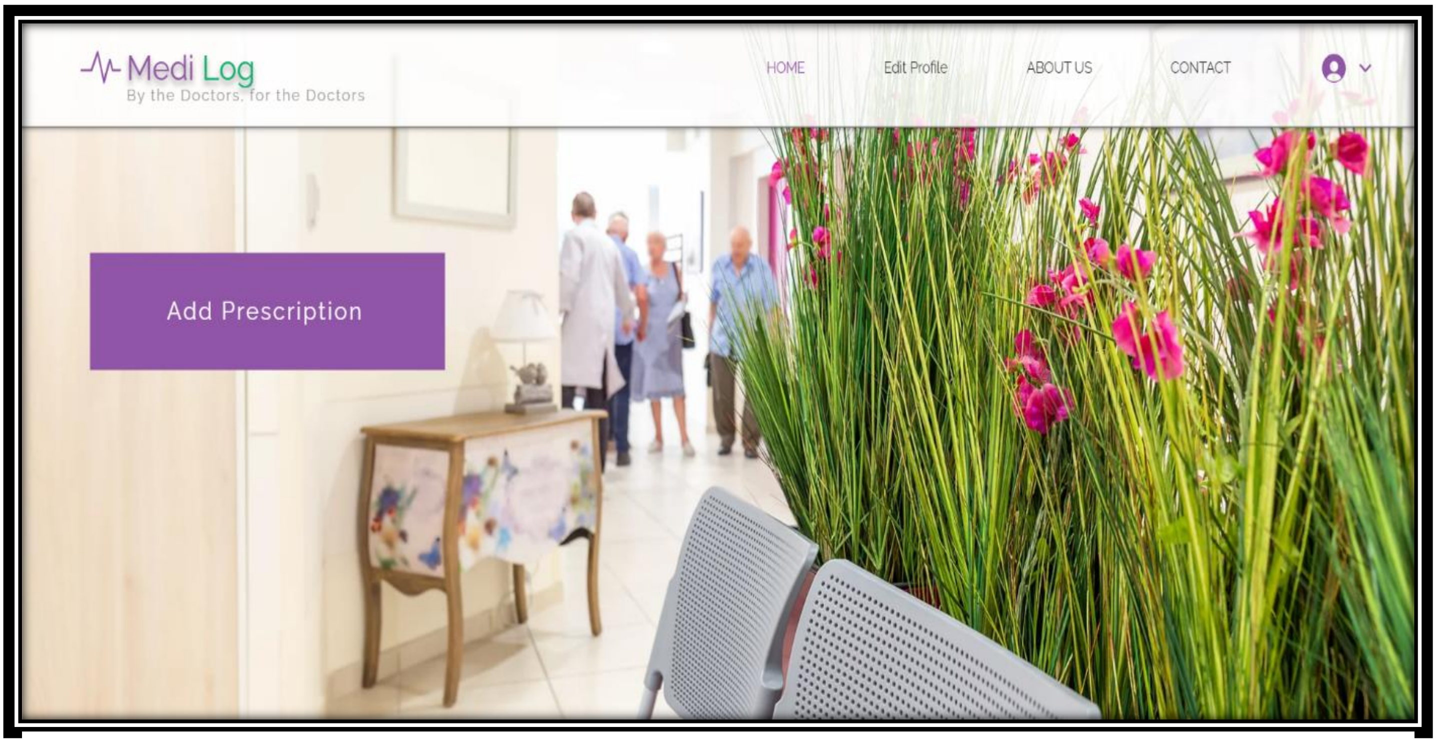


Figure 5 : Hospital Home Page View
GUI for hospital home screen

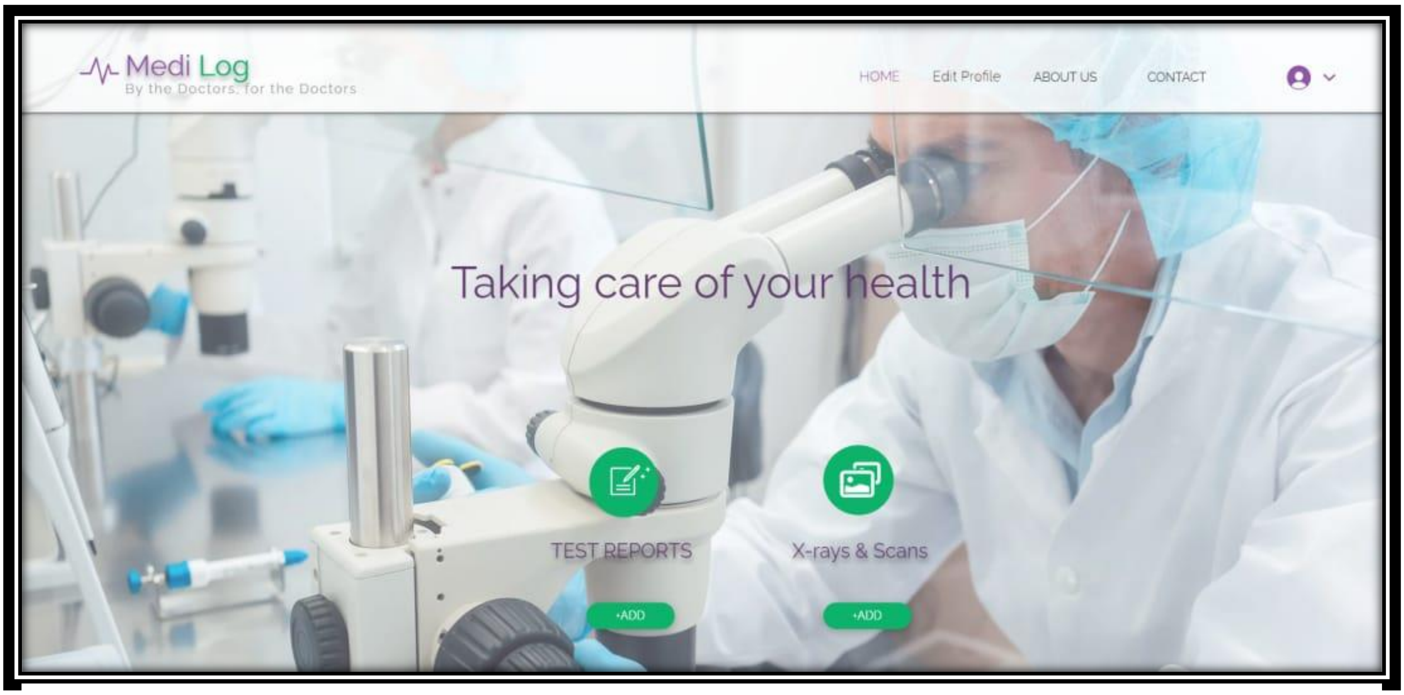
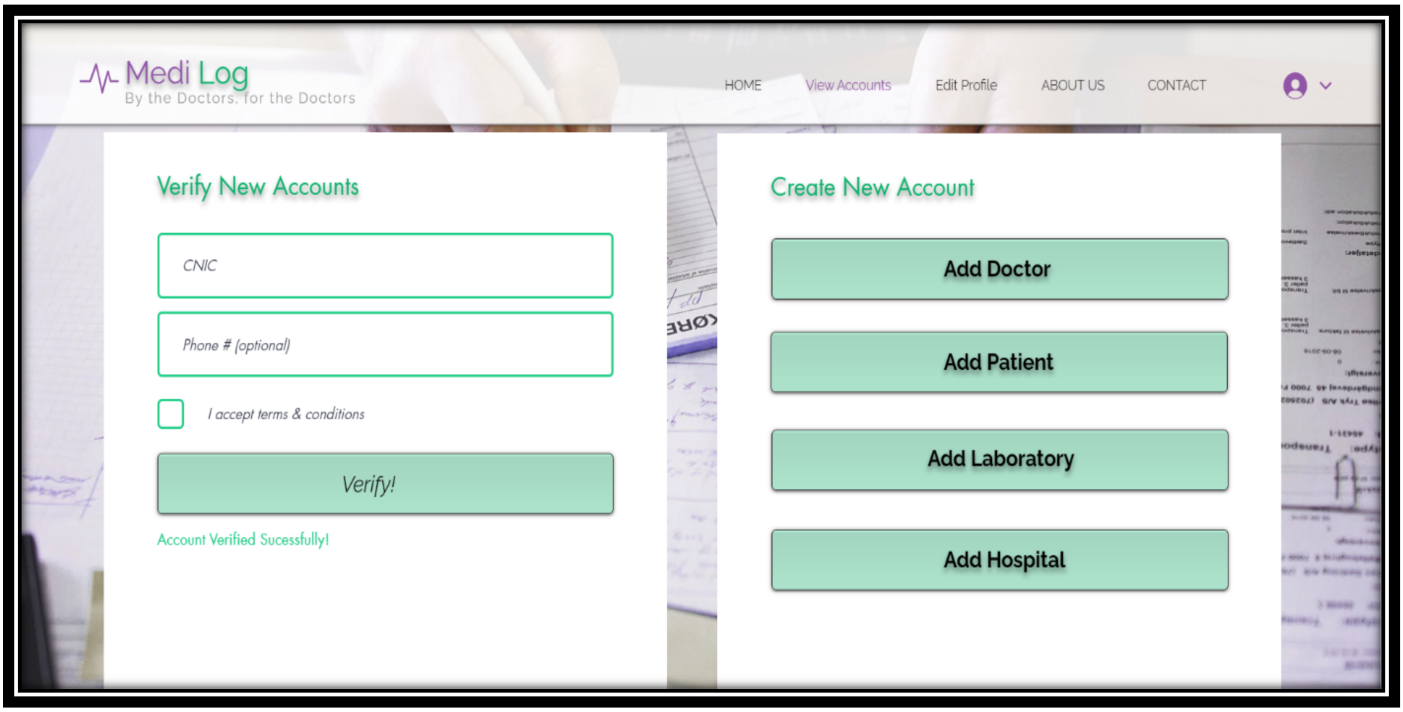


Figure 6 : Laboratory Home Page View
GUI for Laboratory home screen

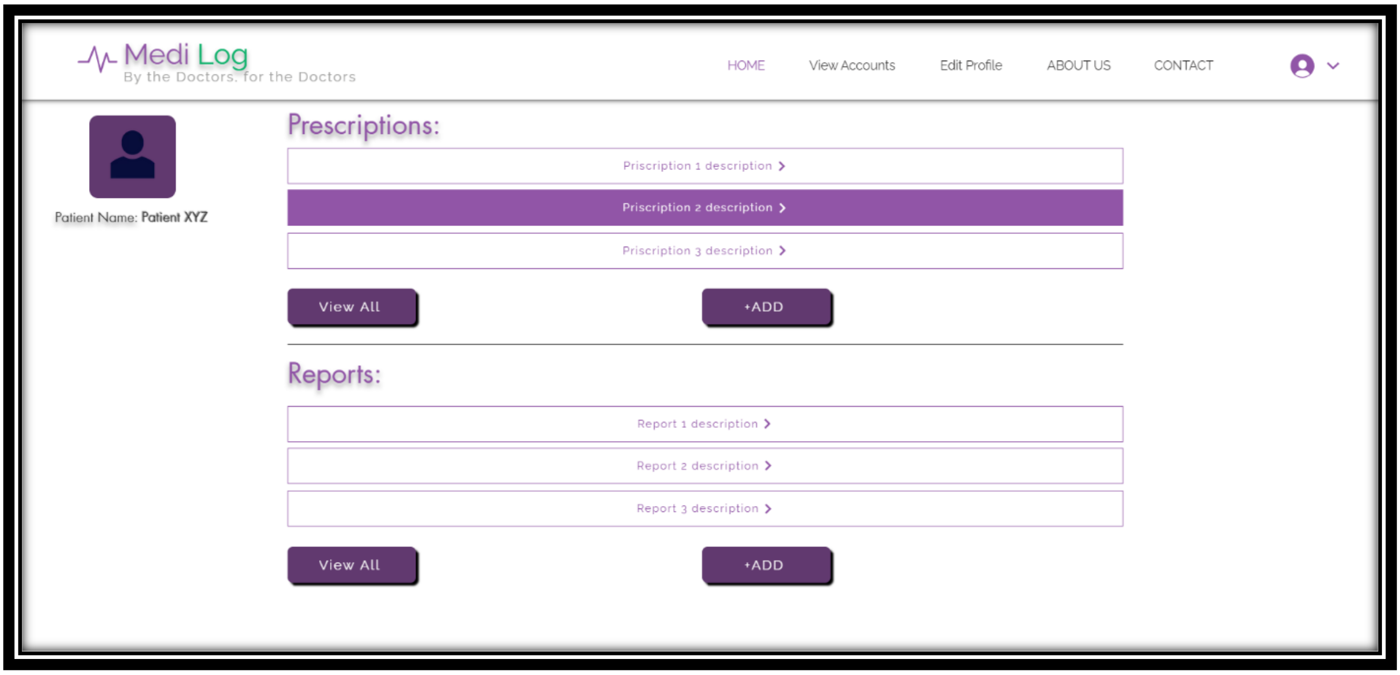


Verify New Accounts

Account Verified Sucessfully!

Create New Account

Figure 7 : Admin Home Page View
GUI for administration home screen



Patient Name: Patient XYZ

Prescriptions:

Priscription 1 description ›

Priscription 2 description ›

Priscription 3 description ›

Reports:

Report 1 description ›

Report 2 description ›

Report 3 description ›

Figure 8 : Administration Upload Data View
GUI for Administration to upload data

3.8 Database Design

3.8.1 ER Diagram

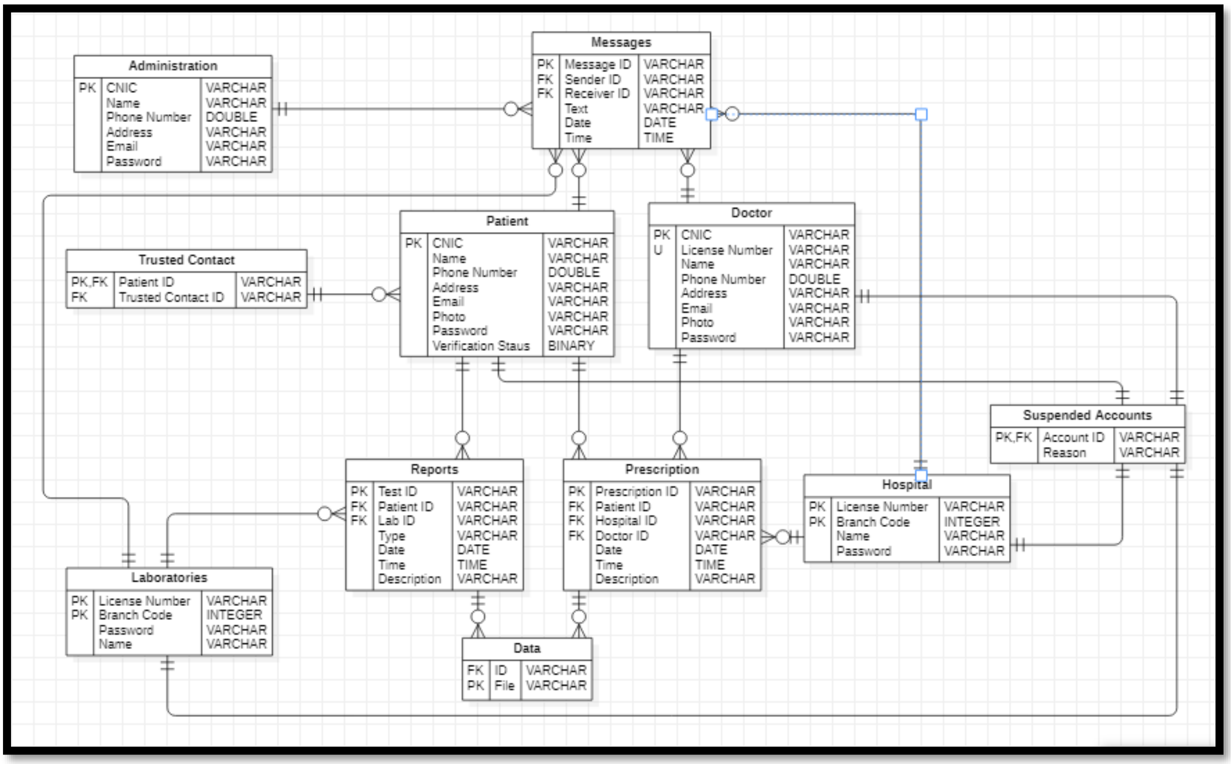


Figure 9 : ER Diagram
Entity Relation Diagram of our system

3.8.2 Data Dictionary

3.8.2.1 Administration

Table 1 : Administration Table
People controlling the smooth flow and working of system

Fields	Datatype	Example
CNIC (PK)	Varchar	36402-1111111-1
Name	Varchar	Ahmad Ali
Phone Number	Double	03211122333
Address	Varchar	852-B Faisal Town Lahore
Email	Varchar	L213344@lhr.nu.edu.pk
Password	Varchar	11111112

3.8.2.2 Patient

Table 2 : Patient Table

Patient is the user whose data we are going to save

Fields	Datatype	Example
CNIC	Varchar	33001-2222222-3
Name	Varchar	Ali Ahmed
Phone	Double	03211111111
Address	Varchar	852-B Faisal Town Lahore
Email	Varchar	aliahm@gmail.com
Password	Varchar	12341234
Verification-status	Binary	0
Photo	Varchar	Xxxxx

3.8.2.3 Doctor

Table 3 : Doctor Table

Our mainstream user who will be able to view patient data

Field	Datatype	Example
CNIC	Varchar	33221-8742989-0
License-no	Varchar	46764gfuef
Name	Varchar	Usama Khan
Phone	Double	03401234123
Address	Varchar	764 A1 Johar Town Lahore
Email	Varchar	usamaK@gmail.com
Password	Varchar	12345678
Photo	Varchar	Xxxxx

3.8.2.4 Hospital

Table 4 : Hospital Table

Actor of our system who will upload prescriptions for patients

Field	Datatype	Example
License-No	Varchar	562574aa
Branch-code	Int	12
Name	Varchar	FAST Hospital
Password	Varchar	12312312

3.8.2.5 Laboratories

Table 5 : Laboratories Table

Our actor of the system who will upload test reports

Field	Datatype	Example
License-No	Varchar	1234asdf
Branch-code	Int	23
Name	Varchar	FAST Lab
Password	Varchar	12345612

3.8.2.6 Reports

Table 6 : Reports Table

Test results that labs will be conducting and later uploading

Field	Datatype	Example
Test-id	Varchar	1
Patient-id	Varchar	33100-1234567-1
Lab-id	Varchar	1234wert
Type	Varchar	XRAY
Date	Date	23-07-2000
Time	Time	12:35:01
Description	Varchar	Hand XRAY

3.8.2.7 Prescription

Table 7 : Prescription Table

Scanned copy or Image of the doctor's findings

Field	Datatype	Example
Prescription-id	Varchar	123as
Patient-id	Varchar	33313-2156689-3
Hospital-id	Varchar	12as
Doctor-id	Varchar	33232-5434255-9
Type	Varchar	Cough
Date	Date	28-10-2020
Time	Time	11:29:12
Description	Varchar	Severe Cough

3.8.2.8 Data

Table 8 : Data Table

Table in which actual files will be stored

Field	Datatype	Example
Id	Varchar	123as
File		Xxxxx

3.8.2.9 Trusted Contacts

Table 9 : Trusted Contacts

The contacts who can access a patient's data

Field	Datatype	Example
Patient ID	Varchar	33245-3543535-6
Trusted Contact ID	Varchar	33456=2342345

3.8.2.10 Messages

Table 10 : Messages Table

Messages that will be conveyed to doctors etc

Field	Datatype	Example
Message Id	Varchar	123
Sender Id	Varchar	33446-5653453-5
Receiver Id	Varchar	33245-3423562-7
Text	Varchar	Hello
Date	Time	23-09-2013
Time	Date	12:34:09

3.9 System Requirements

Following are the software and hardware requirements for end users and developers.

3.9.1 Hardware Requirements

For development, the hardware requirements are the following:

- Desktop with 8GB RAM and 64-bit operating system.

For usage, the hardware requirements are the following:

- Web Browser (PC or Smart Phone).
- Finger Print Scanner (For Administration only).

3.9.2 Software Requirements

For development, the software requirements are the following:

- Code editor: Visual Studio code, Sublime Text, Atom.
- MongoDB Atlas Database.
- Express: WEB application Framework.
- React: A JavaScript library for Frontend.
- Node: JavaScript runtime environment.
- Node Package Manager (For installing dependencies)

Browser: Google chrome, Mozilla Firefox, Microsoft Edge, Safari.

3.10 Design Considerations

This section describes many of the issues which need to be addressed or resolved before attempting to devise a complete design solution.

3.10.1 Assumptions and Dependencies

We assume that:

- Users have a device to access this platform.
- Users have a knowhow of technology.
- Users are Pakistani citizens.
- Users have access to internet connection.

3.10.2 General Constraints

3.10.2.1 Hardware or software environment

- Our software is web application, so end device must have a browser which supports JavaScript.
- In case of hardware, web application requires 4 GB RAM, 64-bit Operating System, Core i5.
- Due to the fact that it is an online web application and we are going to have loads of data coming and going to our server's users must have a stable high-speed internet.

3.10.2.2 Availability or volatility of resources

- High speed internet and a device capable enough to access this system are key attributes for required performance.

3.10.2.3 Standards compliance

Our system follows MVT architecture because we are using Django as our back-end [8].

- Its full form is Model, View, Template.
- Model is the data access layer it handles the data.
- Template is presentation layer which handles User Interface part as a whole.
- View is used for business logic and it also interacts with model to carry data and update the UI.

3.10.2.4 Interface/protocol requirements

- HTTP protocol will be used for communication between server and client via internet.
- We are going to build the interface using React and it will work perfectly fine if all the requirements specified here are met.

3.10.2.5 Data repository and distribution requirements

- We will be using MongoDB to store this application's data.

3.10.2.6 Security requirements

- System will be performing actions after authentication, verification and authorization.

3.10.2.7 Memory and other capacity limitations

- System requires CPU, Memory, I/O capacity, Bandwidth and cache space for better performance.

3.10.2.8 Performance requirements

- High speed internet is required for sending/retrieving data from database.
- Unoptimized code, if the code is properly optimized its execution takes more time and it also consumes more memory.

3.10.2.9 Network communications

- HTTP protocol will be used for network communications.

3.10.2.10 Language Constraints

This system can only be used by people who are familiar with English.

3.11 Development Methods

Thinking about development methods and the nature of our project we have to go with the waterfall model [2]. There are certain reasons for this choosing this model. Starting from the fact that it is the most basic, traditional choice. Second that in our case the whole applications have several phases. Every new phase is dependent on the previous one so we cannot start the next part before actually completing the previous one. The development of our application consists of

- Designing phase
- Implementation phase
- Testing phase
- Deployment phase

Another approach that we are following is called Feature driven Development [2]. Here we have sprints or delivery cycles which help in developing bigger applications step by step and it helps in keeping track of the results and errors.

3.12 Class diagram

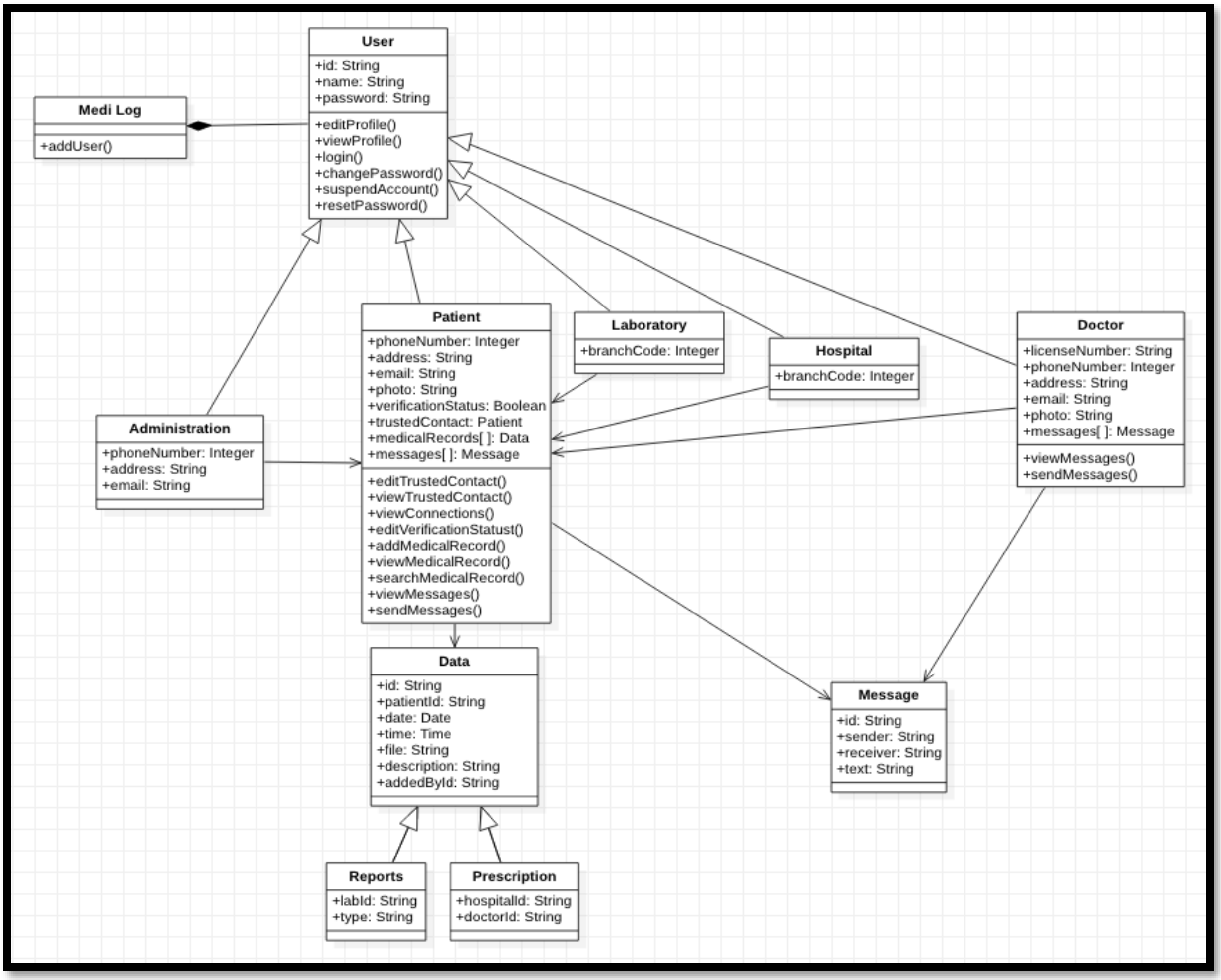


Figure 10 : Class Diagram
Class Diagram of our system

3.13 Sequence diagrams

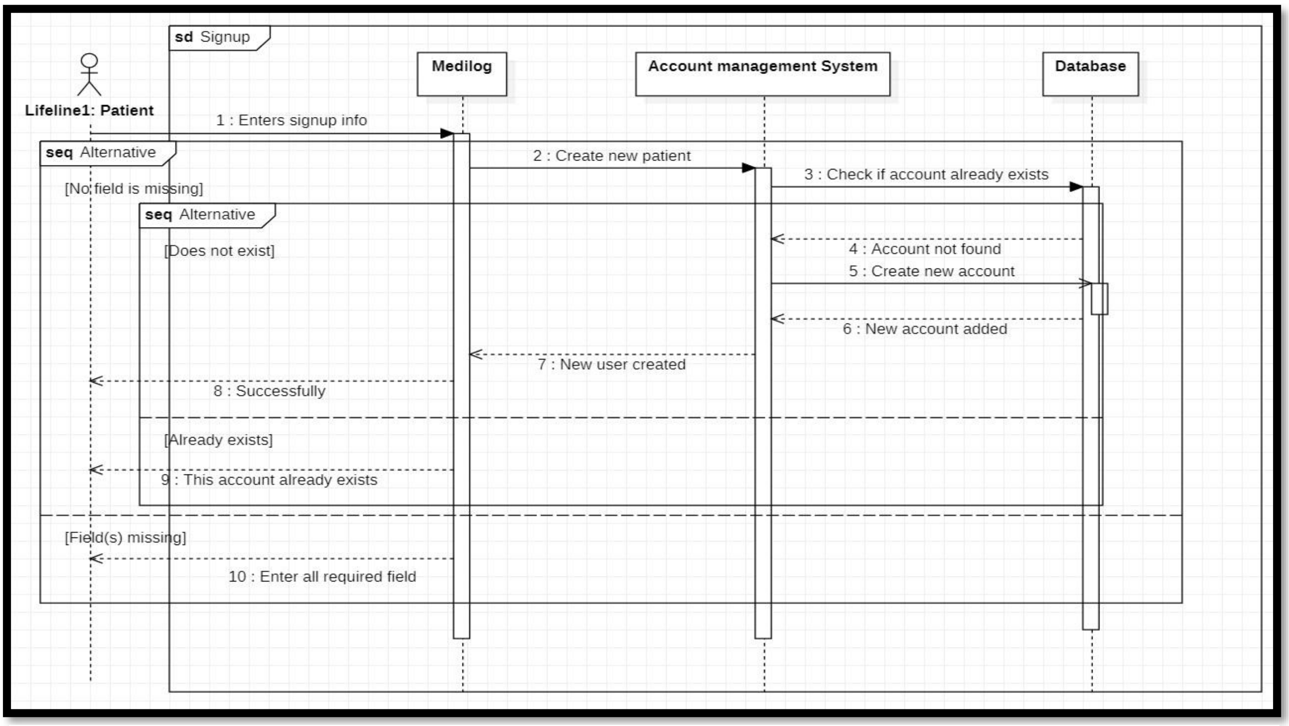


Figure 11 : Sign Up sequence diagram
Sequence Diagram for our sign-up use case

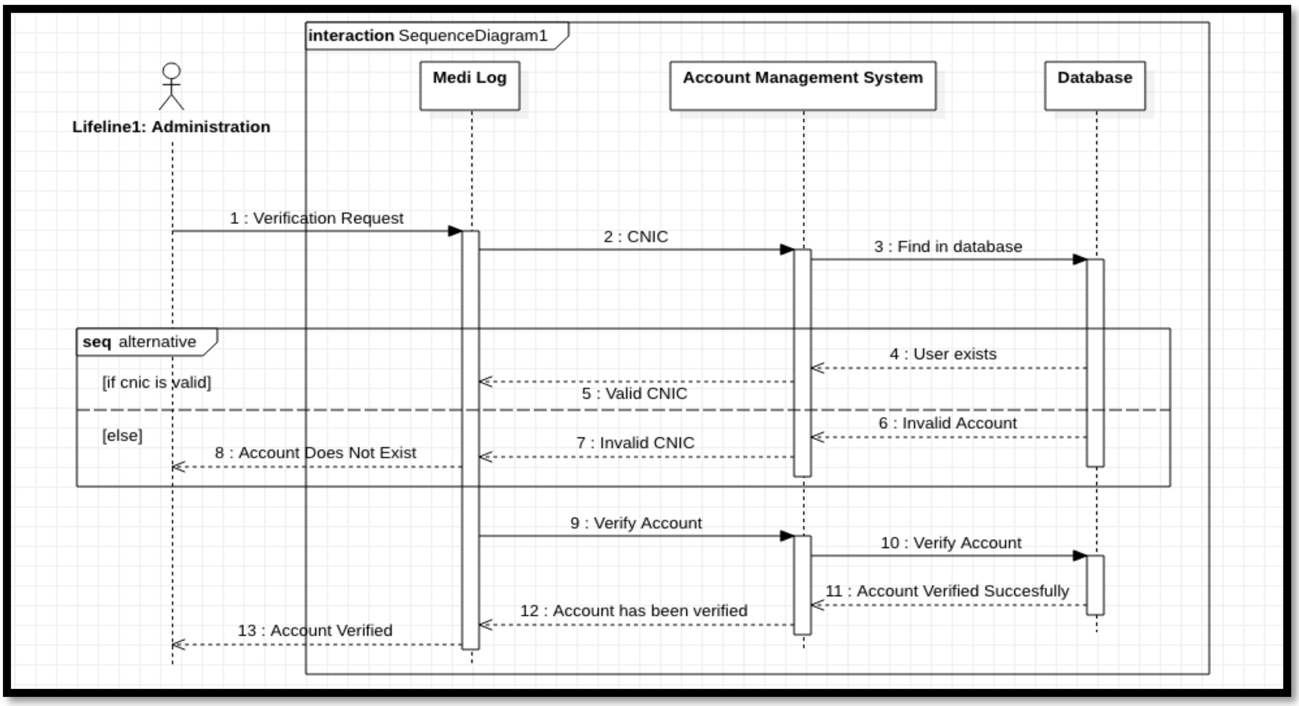


Figure 12: Verify Account Sequence Diagram
Sequence Diagram for our Account Verification Use Case

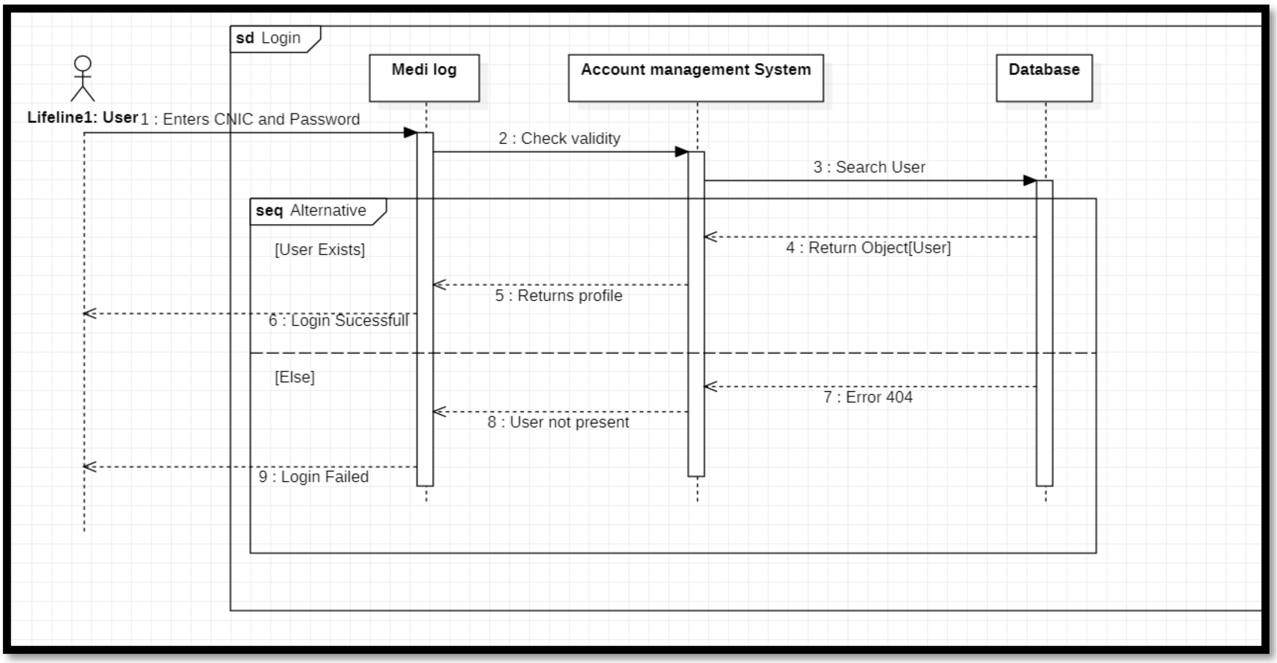


Figure 13 : Login Sequence Diagram
Sequence diagram for login use case

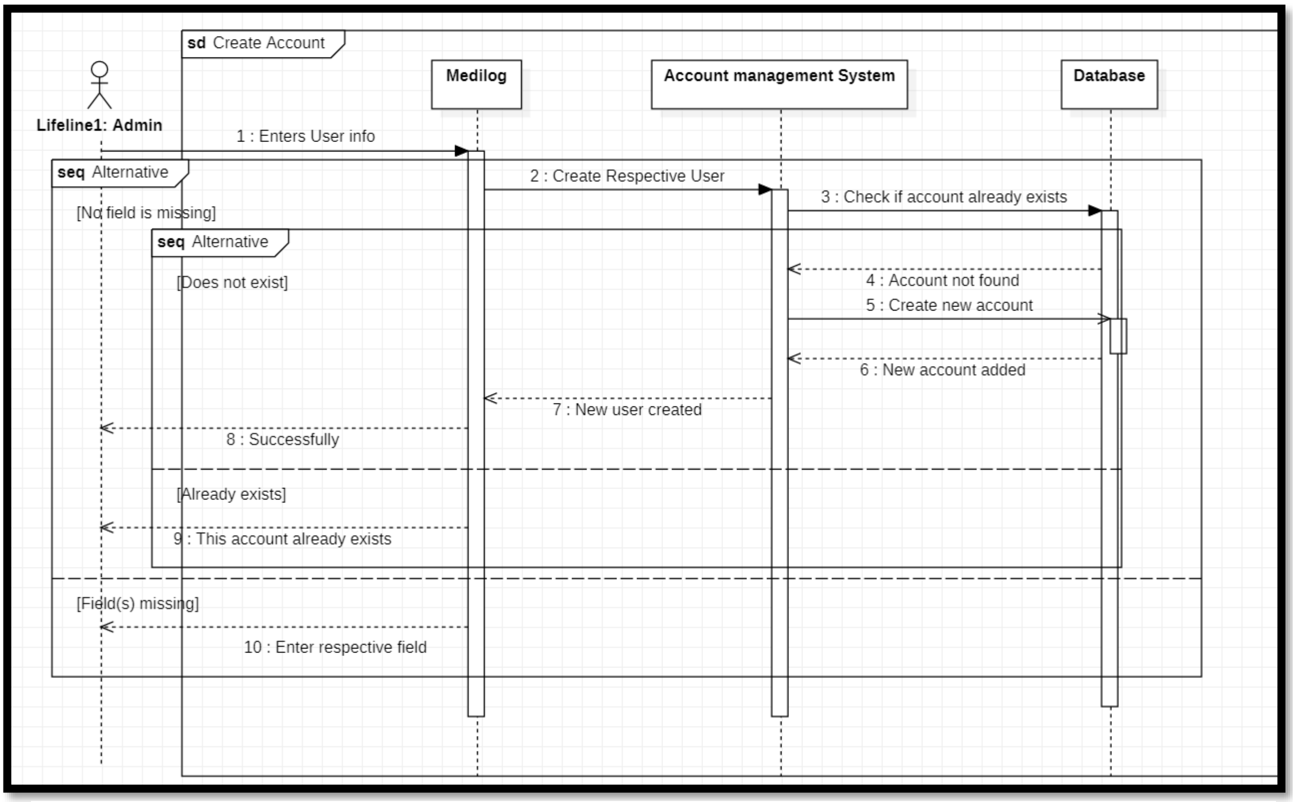


Figure 14 : Create Account Sequence Diagram
Sequence Diagram for account creation by admin

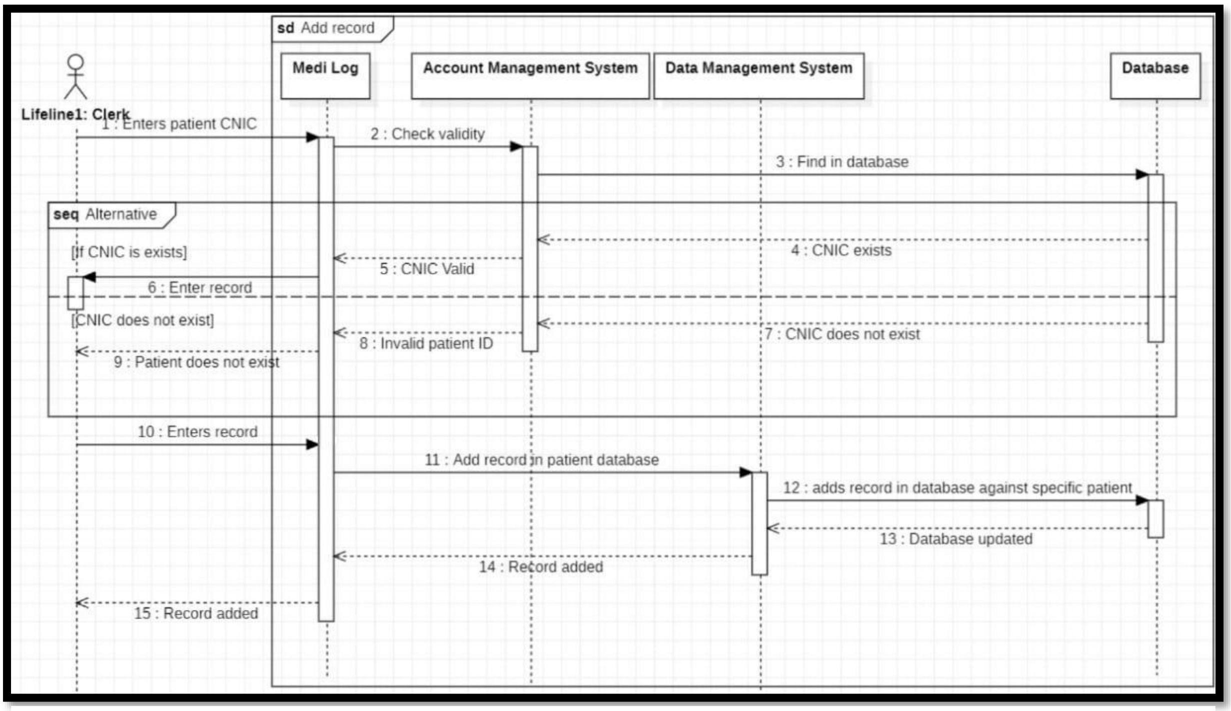


Figure 15 : Add Medical Record Sequence Diagram
Sequence Diagram for adding new medical record

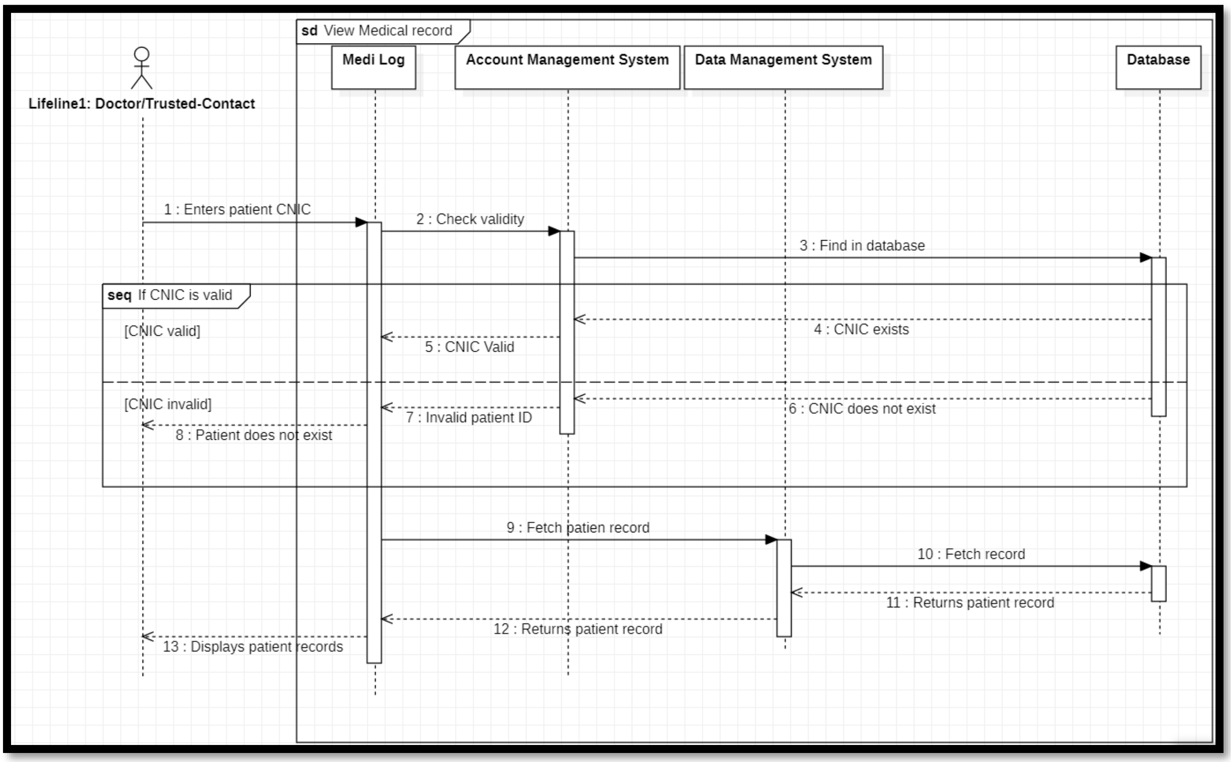


Figure 16 : View Medical Record Sequence Diagram
Sequence Diagram for viewing medical record

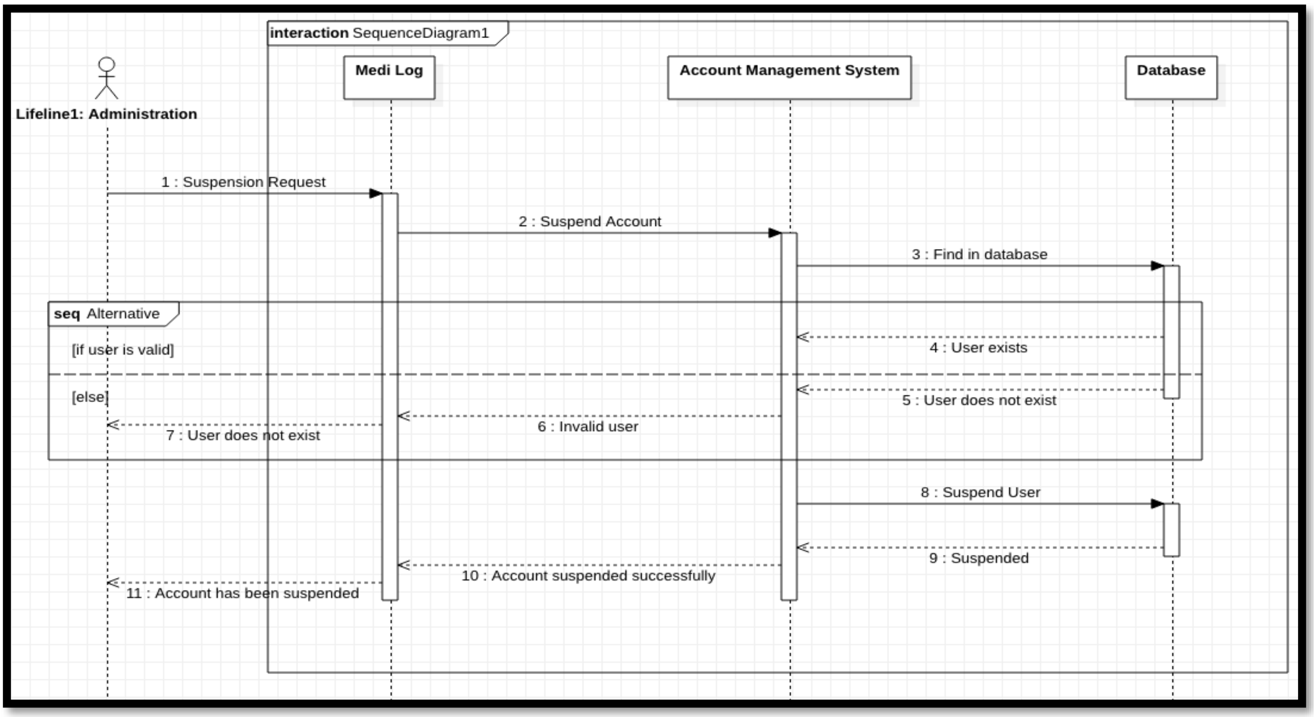


Figure 17 : Suspend Account Sequence Diagram
Sequence Diagram for suspension of account

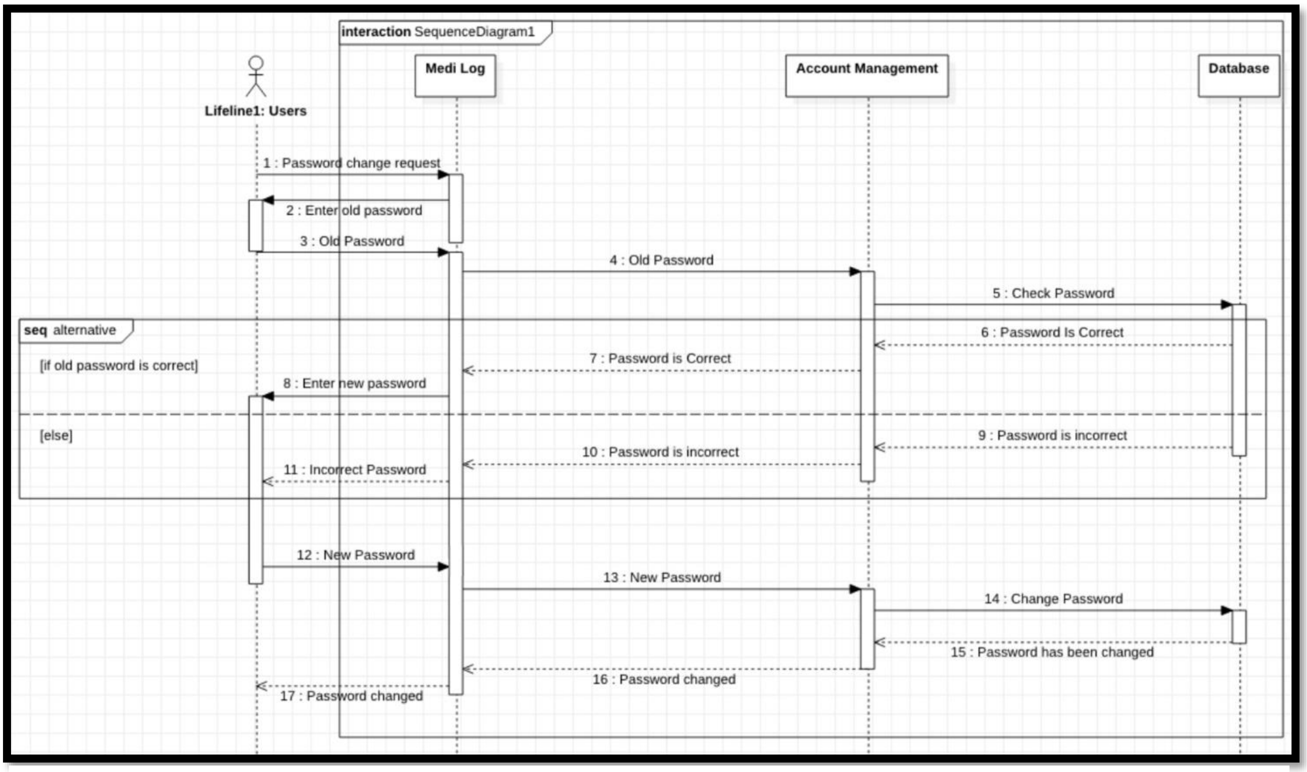


Figure 18 : Change Password Sequence Diagram
Sequence Diagram for changing the password

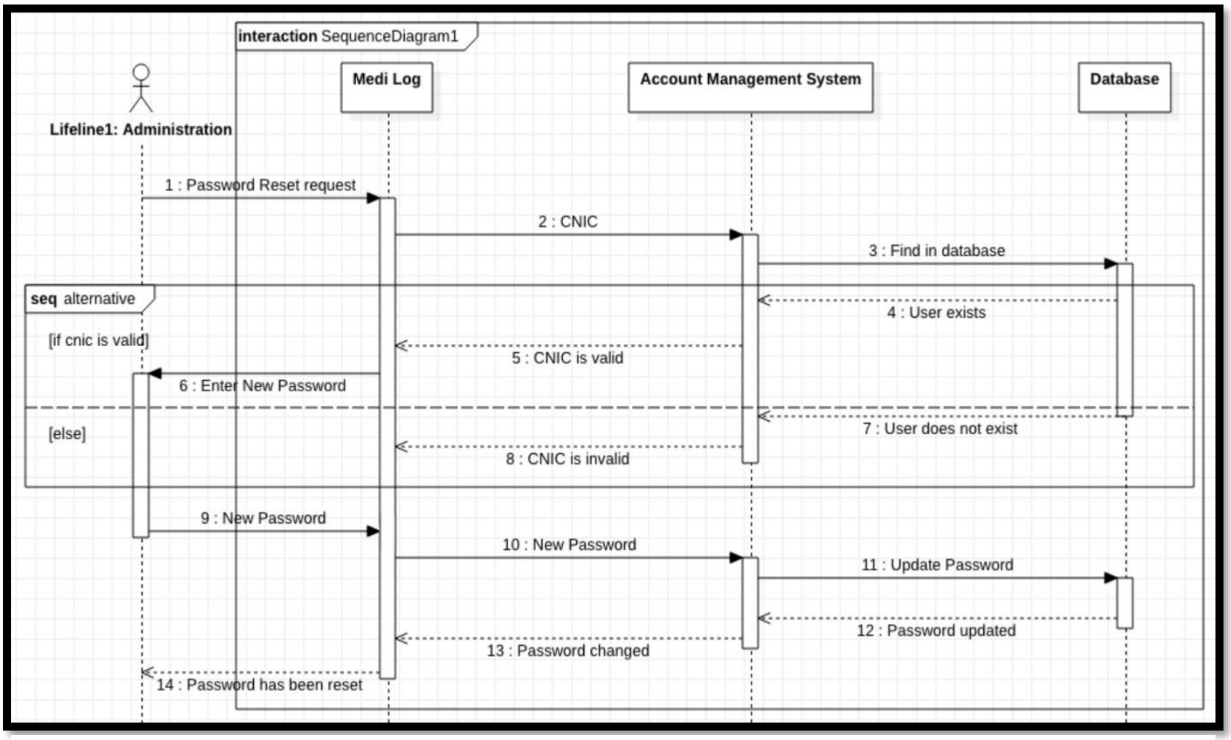


Figure 19 : Reset Password Sequence Diagram
Sequence Diagram for resetting the password

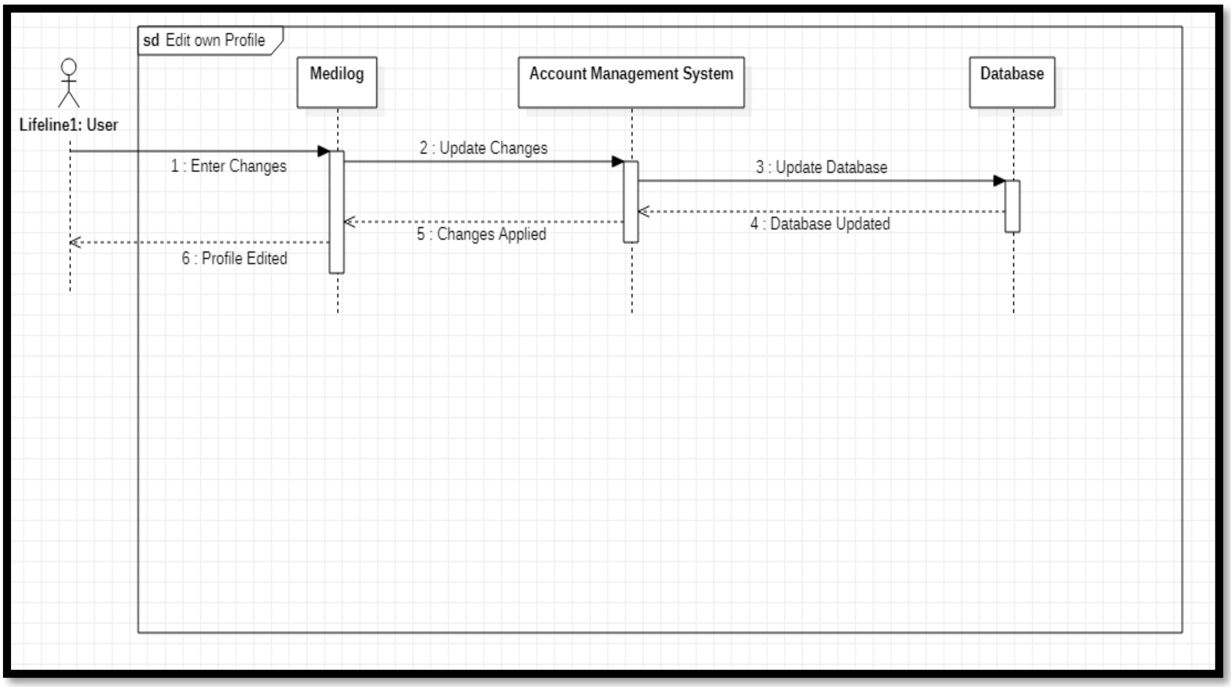


Figure 20 : Edit Own Profile Sequence Diagram
Sequence Diagram for editing one's own profile

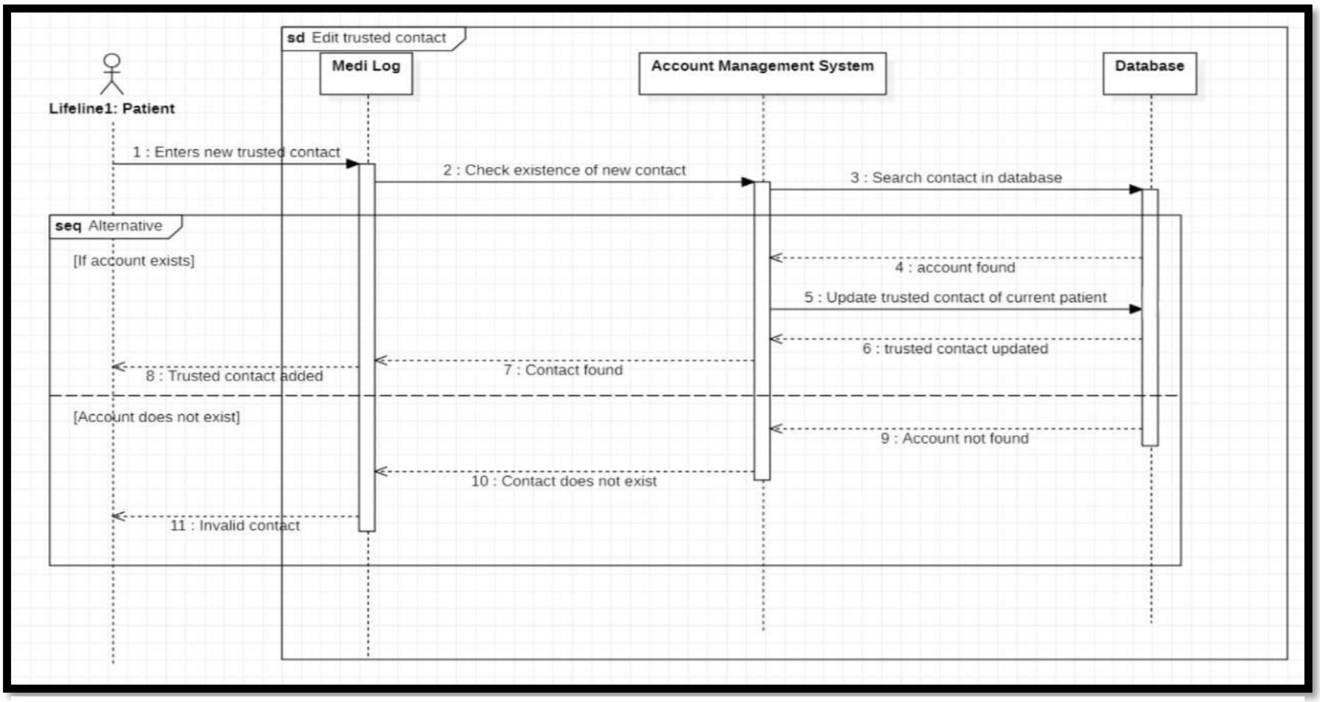

sd Edit trusted contact
Lifeline1: Patient
Medi Log
Account Management System
Database
1 : Enters new trusted contact
2 : Check existence of new contact
3 : Search contact in database
seq Alternative
[If account exists]
4 : account found
5 : Update trusted contact of current patient
6 : trusted contact updated
7 : Contact found
8 : Trusted contact added
[Account does not exist]
9 : Account not found
10 : Contact does not exist
11 : Invalid contact

Figure 21 : Edit Trusted Contact Sequence Diagram
Sequence Diagram for editing one's trusted contacts

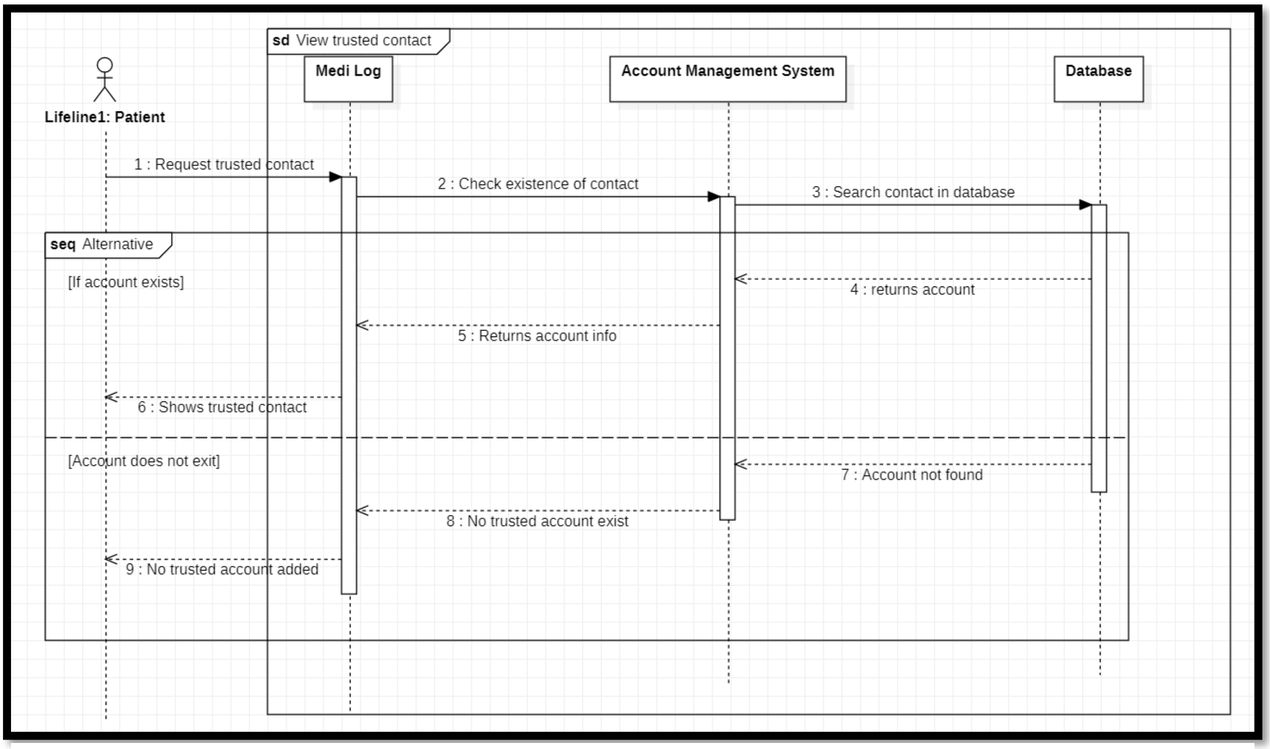

sd View trusted contact
Lifeline1: Patient
Medi Log
Account Management System
Database
1 : Request trusted contact
2 : Check existence of contact
3 : Search contact in database
seq Alternative
[If account exists]
4 : returns account
5 : Returns account info
6 : Shows trusted contact
[Account does not exit]
7 : Account not found
8 : No trusted account exist
9 : No trusted account added

Figure 22 : View Trusted Contact Sequence Diagram
Sequence Diagram for viewing one's trusted contacts

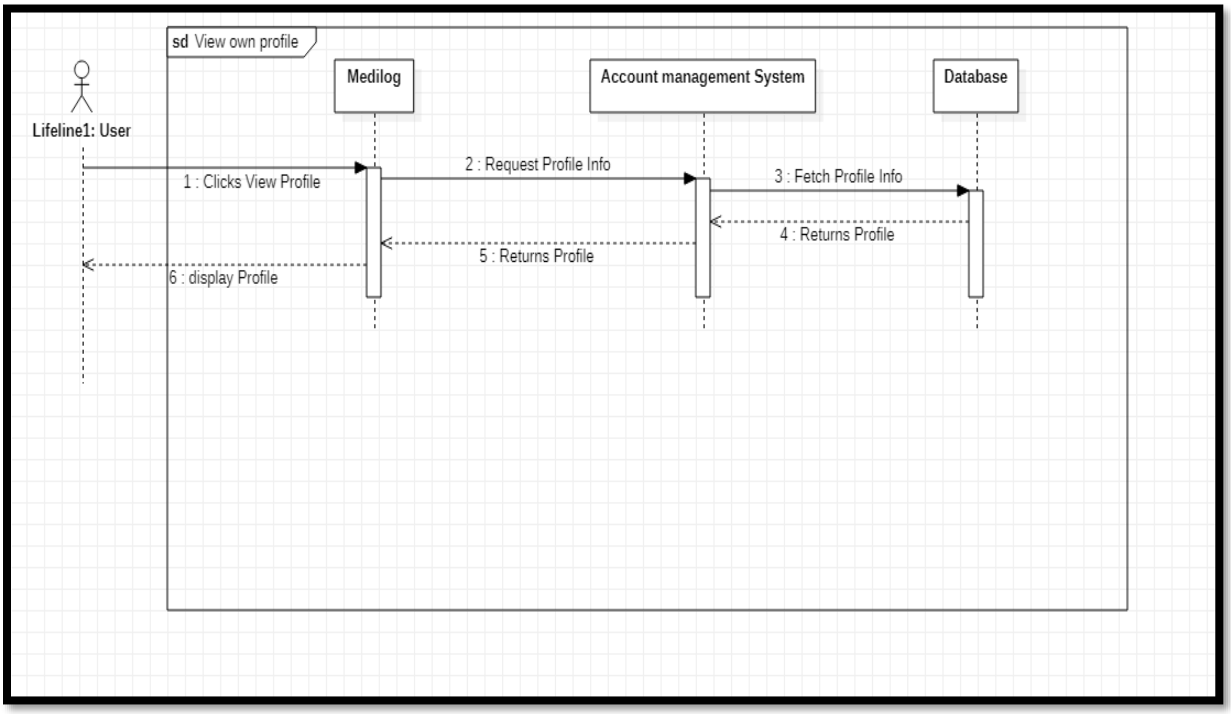


Figure 23 : View Own Profile Sequence Diagram
Sequence Diagram for viewing profile

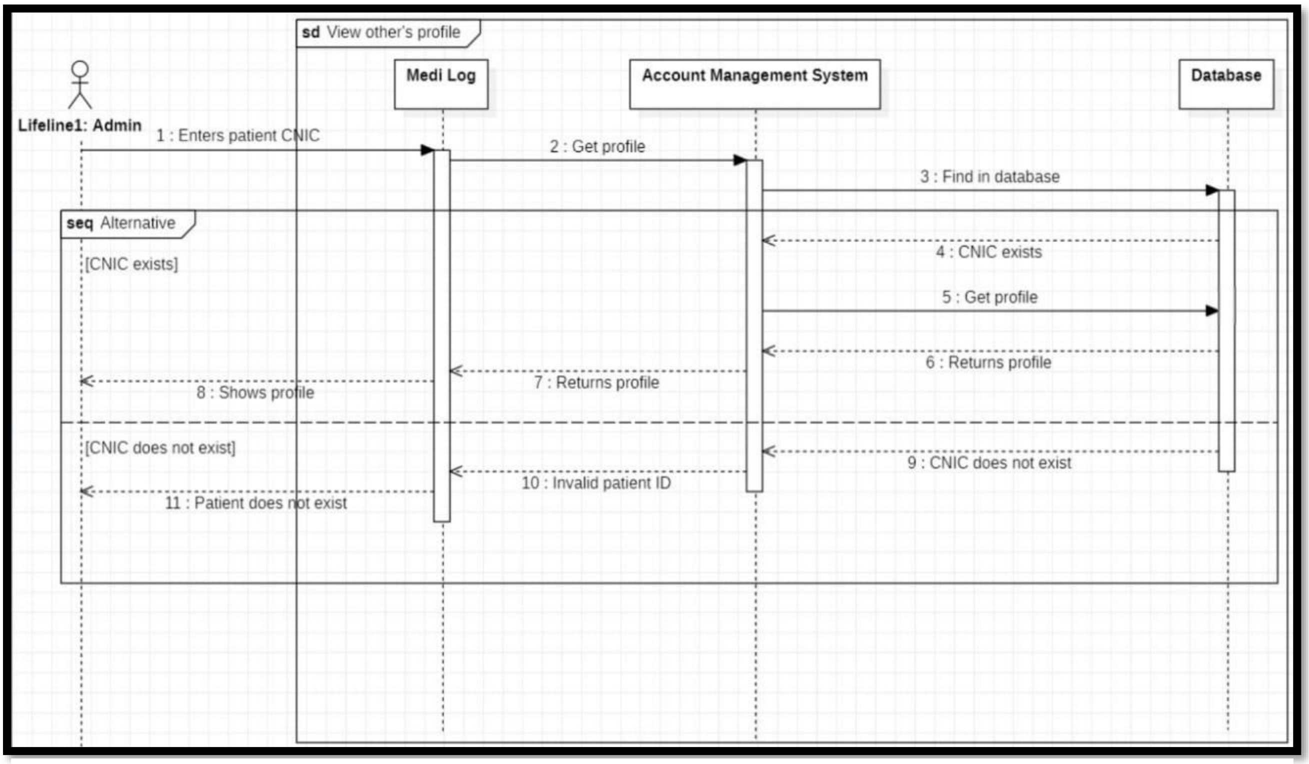


Figure 24 : View Other's Profile Sequence Diagram
Sequence Diagram for viewing other’s profiles

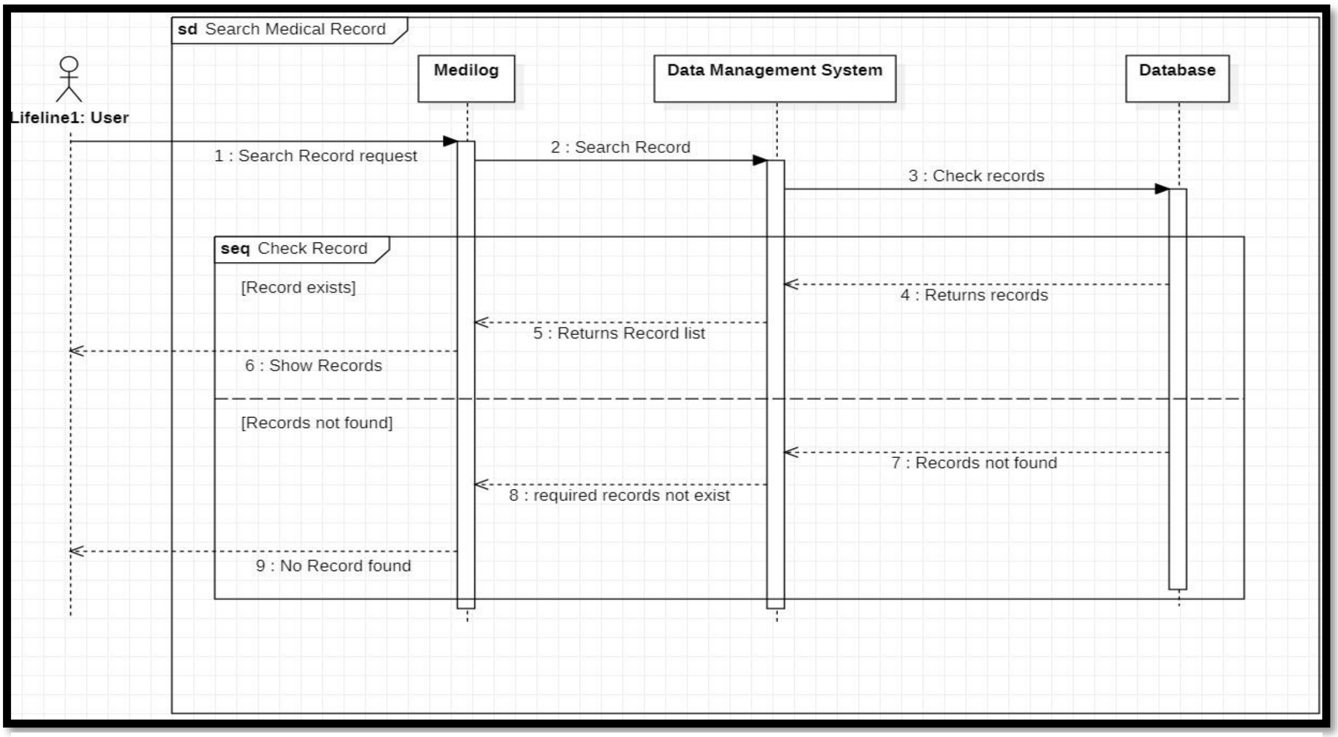


Figure 25 : Search Medical Record Sequence Diagram
Sequence Diagram for searching medical records

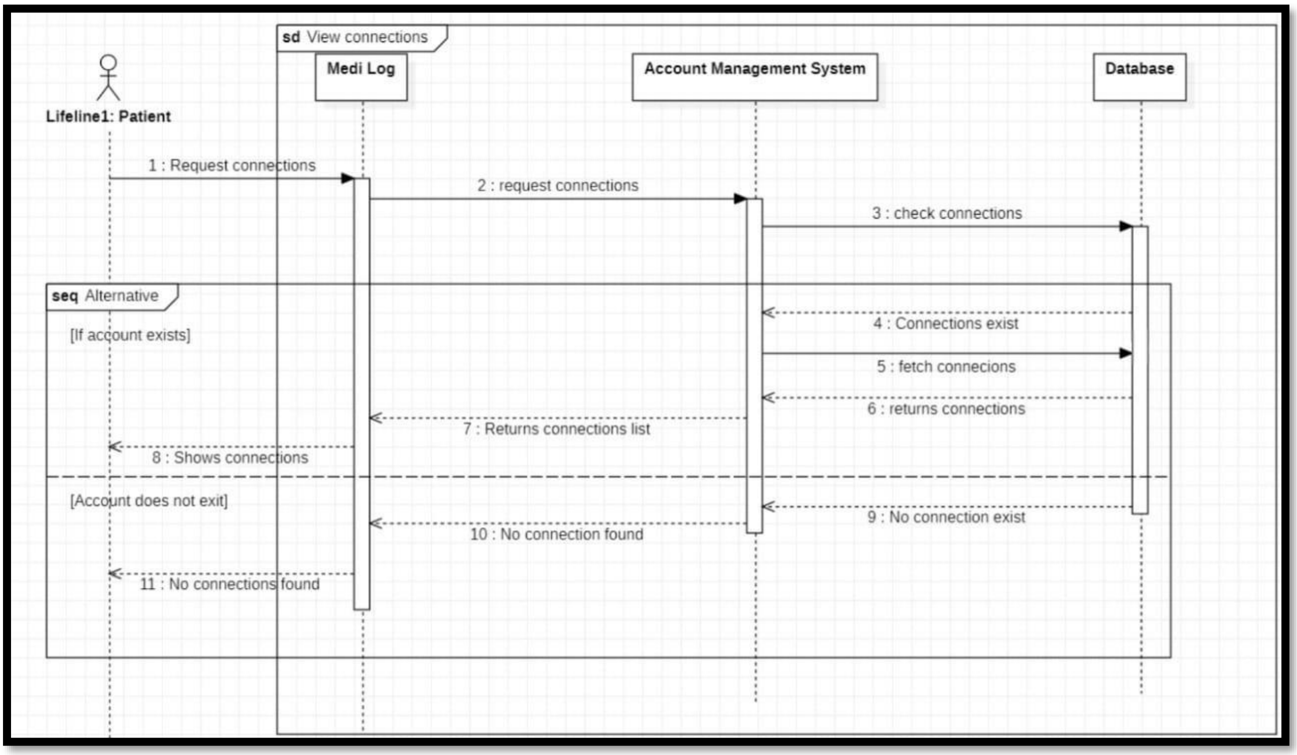


Figure 26 : View Connections Sequence Diagram
Sequence Diagram for viewing connections

3.14 Policies and Tactics

The following are the considerations for the policies that would be followed in the project.

3.14.1 Coding guidelines

For the user interface the eight golden rules of interface design would be utilized as a standard. During the programming phase of the project, rules and standards set in python and node/ react would be followed.

3.14.2 Testing

The accuracy of the different models would be tested with testing data after training the model on the training data provided by the user.

3.14.3 Maintenance/Versions

More advanced versions of the software can be introduced after the completion of this project if we are able to convince the industry to use our product.

3.14.4 Accessing the Application

The web application would be hosted on cloud accessible with a URL link. The application would be developed and accessible on the internet via any computer having minimum requirements.

Chapter 4: Implementation and Test Cases

This chapter gives an overview of the development we have made so far in the implementation of our system. What is built and how it is built.

4.1 Implementation

In this phase of implementation, we have developed a prototype in which we have provided some basic functionalities along with the front-end or graphical user interfaces of the system developed so far.

4.2 Prototype Description

The general overview of the system we have developed so far is given below. We have used Django a python framework for developing this system. For the work done till now we have 5 types of users using this system namely Patient, Doctor, Laboratory, Hospital and Administration. We have made Django models for registration of all users and we have made API's for transferring data between our front-end and back-end. After successfully implementing user sign-up and login we have been working on updating the profile. All users share the same login page and get redirected to their respective home pages afterwards. We have used MongoDB for storing the data and for connecting the Django app to our database we have used djongo. Then moving on to one of our main use cases of storing a patient's medical data we have implemented the feature of storing a doctor's prescription from the hospital account. Now we are working on further modules of storing lab reports and scans and then also how to make all this data and information actually beneficial in diagnosis and helping the doctor in making an accurate decision.

Chapter 5: Conclusion

Our project is in continuous progress state as we have been through several phases now and many are yet to come. We have been working on how to make this project more and more helpful in accurate diagnosis by not just relying on storing and displaying data but to make it actually useful we are working on how to make the displaying of data part as much interesting as we can. We all know that medical data is very sensitive so we have to be very careful in this regard. We are going to add bio-metric verification for an extra layer of security. For this purpose, we have a few ideas that are mapping a human body model in our front-end to make it visually appealing and more interesting for a doctor to extract data of a person the map will be responsive to the actual user and will portray important information for each user dynamically also, we are going to add notification alerts to users via email if a new report or a prescription is added to user’s account.

References

[1] DotNetTricks. 2020. *Different Types of Software Design Principles.* [Online]. Available at: <https://www.dotnettricks.com/learn/designpatterns/different-types-of-software-design-principles> [Accessed 1 December 2020].

[2] TutorialsPoint. 2020. *Project Management Methodologies*. [Online]. Available at: <https://www.tutorialspoint.com/management_concepts/project_management_methodologies.htm> [Accessed 1 December 2020].

[3] PeerBits. 2020. *The benefits of ReactJS and reasons to choose it for your project*. [Online]. Available at: <https://www.peerbits.com/blog/reasons-to-choose-reactjs-for-your-web-development-project.html> [Accessed 1 December 2020].

[4] Upgrad Blog. 2020. *Django vs NodeJS: Difference Between Django and NodeJS*. [Online]. Available at: <https://www.upgrad.com/blog/django-vs-nodejs-difference-between-django-and-nodejs/ > [Accessed 1 December 2020].

[5] EDUCBA. 2020. *MongoDB vs SQL server*. [Online]. Available at: <https://www.educba.com/mongodb-vs-sql-server/> [Accessed 1 December 2020].

[6] Interaction Design Foundation. 2020. *User Interface Design Guidelines: 10 Rules of Thumb*. [Online]. Available at: <https://www.interaction-design.org/literature/article/user-interface-design-guidelines-10-rules-of-thumb> [Accessed 1 December 2020].

[7] Bokardo. 2020. *Principles of User Interface Design*. [Online]. Available at: <http://bokardo.com/principles-of-user-interface-design/> [Accessed 1 December 2020].

[8] Rinu Gour. 2020. *3 Major Architectural pattern Django follows*. [Online]. Available at: <https://medium.com/@rinu.gour123/3-major-architectural-pattern-django-follows-8c7aff0a5af6> [Accessed 1 December 2020].