**TABLE OF CONTENTS**

**Pages**

**Acknowledgements ……………………………………………………………….... i**

1. **Introduction**……………………………………………………………………… 1
   1. purpose…………………………………………………………………………… 2
   2. scope……………………………………………………………………………… 3
   3. goals………………………………………………………………………………..4
   4. objectives…………………………………………………………………………..5
2. **system features, user and their roles**……………………………………………..6

2.1 system features………………………………………………………………….6

2.2 user characters…………………………………………………………………..7

1. **requirements**……………………………………………………………………….8

3.1functional requirements………………………………………………………….8

* 1. nonfunctional requirements……………………………………………………..9
  2. **system models**……………………………………………………………………….10

3.5use cases………………………………………………………………………….10

* 1. Admin use case …………………………………………………..….…...10
  2. Patient use case ……………………………………………………..……11
  3. System use case …………………………………………………………..12
  4. Doctor …………………………………………………………………….13

1. Activity diagram ………………………………………………………………….14
   * 1. Doctor activity diagram …………………………………………………..14
     2. Patient activity diagram ……………………………………………..……15
     3. Admin activity diagram ………………………………………………......16
   1. class diagram………………………………………………………………………17
   2. sequence diagram ………………………………………………………………....18
      1. Admin sequence diagram …………………………………………………18
      2. Doctor sequence diagram …………………………………………………19
      3. Patient sequence diagram …………………………………………………20
   3. system architecture………………………………………………………………...21

4.6 deployment diagram……………………………………………………………….22

* 1. entity relationship diagram………………………………………………………...23

4.8 Languages and tools used for the system implementation…………………………….24

1. **interfaces**………………………………………………………………………….......25

5.1homepage…………………………………………………………….……….…...25

* 1. registration page………………………………………………………….……......26
  2. admin panel………………………………………………………………………..27
  3. add doctor………………………………………………………………..…...…....28

6.Doctor ……………………………………………………………………………..29

6.1 doctor dashboard …………………………………………………………….29

6..2 Doctor Schedule appointment ………………………………………………30

* 1. View list of patients………………………………………………………………..31

7Patient ……………………………………………………………………………..32

7.1Book appointment ………………………………………………………...32

7.2View medical records ……………………………………………………..33

7.3View doctors available…………………………………………………….34

7.4Talk to My Doctor ………………………………………………………...35

7.5hardware interfaces…………………………………………………………...........36

7.6software interfaces………………………………………………………………....37

**8.Conclusion ……………………………………………………………………………30**

**9.Glossary ………………………………………………………………………………31**

**ACKNOWLEDGEMENTS**

First and foremost, praises and thanks to the God Almighty, for the blessings of wisdom, knowledge intelligence and hard work throughout our research work and studies to complete our project successfully.

We would like to express our deep and sincere gratitude to our project supervisor Mr Sande Kachale for giving us the opportunity to do the project and providing invaluable guidance and for our ongoing support during the project, from initial advice, motivation and encouragement throughout this Project. We extending our heartfelt thanks to Mr. Tchili Alindiamao, Mr Finn Msiska and Mr. Lovemore Ngwira and Mr Mwase for the helpful information, practical advice and unceasing ideas that have always helped us tremendously in working on the lifeline project. Their immense knowledge and profound experience have enabled us to complete this Project successfully. Once again Special thanks to my project coordinators Tchili Alindiamao and Mr. Finn, who were very considerate, graceful and patient from day one till to the day our project was successfully completed. Not forgetting the entire ICT team of staff members who allowed us to go through with the project.

1. **INTRODUCTION**

Lifeline is a specialized patient portal that aims to revolutionize communication in the healthcare industry. It incorporates an expert system doctor substitute to address situations where personnel are currently unavailable. The expert system intervenes to provide prompt responses to patient inquiries, utilizing symptom checks and diagnosis. This augmentation of patient satisfaction is further reinforced by enabling patients to book appointments on available time slots. The development of Lifeline involves a thorough description of its architecture, functionality, and system components at this stage of the software development process.

The architecture of Lifeline refers to the overall design and structure of the system, including its hardware, software, and network infrastructure. This includes the selection of appropriate technologies, programming languages, and platforms to support the system’s functionality. The architecture must be designed to ensure scalability, reliability, and security, while also being user-friendly and accessible to both physicians and patients.

Functionality refers to the specific features and capabilities of Lifeline. These may include appointment scheduling, patient registration, medication management, lab results tracking, and secure messaging between patients and physicians. The functionality must be tailored to meet the unique needs of the hospital and its patients, while also complying with relevant regulations and industry standards.

System components refer to the individual parts or modules that make up Lifeline. These components must be designed to work together seamlessly, providing a cohesive and intuitive user experience. Examples of system components may include patient registration forms, medication management tools, and messaging systems.

To successfully deploy Lifeline, it is essential to thoroughly describe each of these elements. This allows developers to create a system that is tailored to the needs of the hospital and its patients, while also ensuring that it is secure, reliable, and user-friendly. By providing a clear and detailed description of the architecture, functionality, and system components, developers can ensure that Lifeline meets its objectives and provides value to both physicians and patients.

1.1 **PURPOSE**

The purpose of the work is to develop a specialized patient portal called Lifeline, which aims to transform communication in the healthcare industry. The system incorporates an expert system doctor substitute to provide prompt responses to patient inquiries and enable patients to book appointments on available time slots. The portal is designed to meet the unique needs and challenges of a specific hospital, providing physicians and patients with enhanced access to essential medical services. In additional, it also improves collaboration, and coordination between patients and healthcare providers. By providing a secure and accessible platform for patients and doctors to interact, a Patient Doctor Portal can help to streamline the healthcare process, reduce administrative burdens, and enhance the overall quality of care.

**1.2 SCOPE**

1. Functionality:

* Develop specific features and capabilities of Lifeline, such as appointment scheduling, patient registration, medication management, lab results tracking, and secure messaging between patients and physicians.
* Integrate an expert system doctor substitute to provide prompt responses to patient inquiries, utilizing symptom checks and diagnosis.
* Tailor functionality to meet the unique needs of the hospital and its patients while complying with relevant regulations and industry standards.

1. System Components:

* Develop individual parts or modules that make up Lifeline, such as patient registration forms, medication management tools, messaging systems, and the expert system doctor substitute.
* Ensure that system components work together seamlessly, providing a cohesive and intuitive user experience.

1. Testing and Quality Assurance:

* Conduct thorough testing and quality assurance to ensure that Lifeline is functioning properly and meets the needs of the hospital and its patients.
* Identify and address any bugs or issues that arise during testing.

1. Deployment and Maintenance:

* Deploy Lifeline in a secure and reliable manner.
* Provide ongoing maintenance and support to ensure that Lifeline remains secure, reliable, and up-to-date.
* Monitor system performance and make updates as needed to improve functionality and user experience.

1. System Architecture:

* Design the overall architecture of Lifeline, including hardware, software, and network infrastructure.
* Select appropriate technologies, programming languages, and platforms to support the system's functionality.
* Ensure scalability, reliability, security, and user-friendliness while being accessible to both physicians and patients.
  1. **GOALS**

1. Enhanced Patient-Doctor Interaction: The primary goal of the PDP is to facilitate seamless communication and interaction between patients and healthcare providers. This includes providing a user-friendly interface for patients to submit their medical concerns and receive timely responses.
2. Efficient Symptom Checking: To achieve this goal, the PDP will allow patients to input their symptoms into the system. The system will then provide an initial assessment, ensuring that patients can seek preliminary guidance conveniently.
3. Expert Diagnosis: The PDP aims to provide accurate diagnoses and medical advice through the Expert Doctor Substitute System (EDSS) when necessary. This goal is crucial for ensuring that patients receive reliable medical information.
4. Improved Patient Satisfaction: Enhancing patient satisfaction is a fundamental objective. The PDP will achieve this by offering prompt responses to patient queries, convenient appointment scheduling, and a user-friendly experience.
5. Optimized Resource Utilization: To achieve efficient resource allocation, the PDP will utilize the EDSS during peak or unavailable hours, ensuring that medical personnel are used effectively **1.4 OBJECTIVES**
6. User-Friendly Interface: Develop a user-friendly patient portal interface that allows patients to easily report symptoms and book appointments.
7. Data Security and Compliance: Maintain robust data security measures to protect patient information and ensure compliance with healthcare regulations and standards.
8. Rapid Response Time: Achieve a response time of less than [specified time] for patient queries through the EDSS to provide timely medical advice and assistance.
9. User Account Management: Implement user registration and authentication functionalities for patients, doctors, and administrators to manage their accounts securely.
10. Medical Record Access: Enable authorized medical personnel to access patient medical records securely and efficiently to provide informed care.
11. Product Perspective: The Patient-Doctor Portal is part of the healthcare ecosystem; it connects to current healthcare systems and has features such as appointment scheduling software and Electronic Health Records (EHR) systems. It complements these systems by providing improved appointment booking and communication features.

**2.1 USERS AND THEIR ROLES**

**SYSTEM FEATURES**

* User Registration and Authentication
* Appointment Booking
* Patient records management
* Reviews and feedback
* Security and privacy
* Reports and Analytics (Optional)

**2.2 USER CHARACTERS**

1. **Patients**

Individuals who are seeking healthcare services. These individuals may be sick or injured and in need of medical attention, or they may be healthy and seeking preventive care or routine check-ups. Patients can use the Patient Doctor Portal to access their medical records, review their healthy history to ensure that all information is accurate and up-to-date. They can also use portal to communicate with their healthcare providers, and schedules appointments.

1. **Doctors**

Medical professionals who are responsible for providing diagnoses and medical advice to patients. They can use portal to provide patients with educational resources and personalized health advice, to manage patient appointments.

1. **Administrators**

System administrators who are responsible for managing user accounts, setting up access permission, monitoring user activity to prevent unauthorized access or data breaches, ensuring that the system is functioning properly, and overseeing data security and compliance. These administrators may include IT professionals and system managers.

**3. REQUIREMENTS**

**3.1 FUNCTIONAL REQUIRMENTS**

1.User Registration: Patients should be able to create accounts with their personal information securely.

2.User Authentication: Patients must log in securely using credentials (e.g., username and password).

3.Expert Doctor Substitute System (EDSS): The EDSS should be capable of diagnosing common medical conditions based on patient-reported symptoms.

4.Appointment Booking and Management: Patients should be able to view available slots and book appointments conveniently and Patients must have the ability to reschedule or cancel appointments.

5.View Medical Records: Patients can securely access their medical records and history.

6.Symptom Reporting: Patients should have the capability to input their symptoms through a user-friendly interface.

7.Symptom Assessment: The system should analyze patient-entered symptoms to provide an initial assessment.

**DOCTOR AND ADMINSTRATORS PORTAL**

1.User Authentication: Doctors and administrators must log in securely using credentials.

2.View Appointments: Doctors and administrators should have access to appointment schedules.

3.Access Patient Records: Authorized medical personnel can view and update patient medical records.

4.User Management: Administrators can manage user accounts, including adding, modifying, or deactivating accounts

**3.2 NON-FUNCTIONAL REQUIREMENTS**

* 1. Scalability: The PDP should handle an increasing number of users and data without compromising performance.

1.2 Availability: The system should be available 24/7 to handle emergency queries.

1.3 Data Encryption: All sensitive patient data should be encrypted to ensure privacy and security.

1.4 Access Control: Access to patient medical records and user accounts must be strictly controlled and limited to authorized personnel.

1.5 User Interface Design: The user interface should be intuitive, user-friendly, and accessible to users with varying technical proficiency.

1.6. External Systems: The PDP should integrate with external systems, such as medical records databases and scheduling systems, for seamless data exchange.

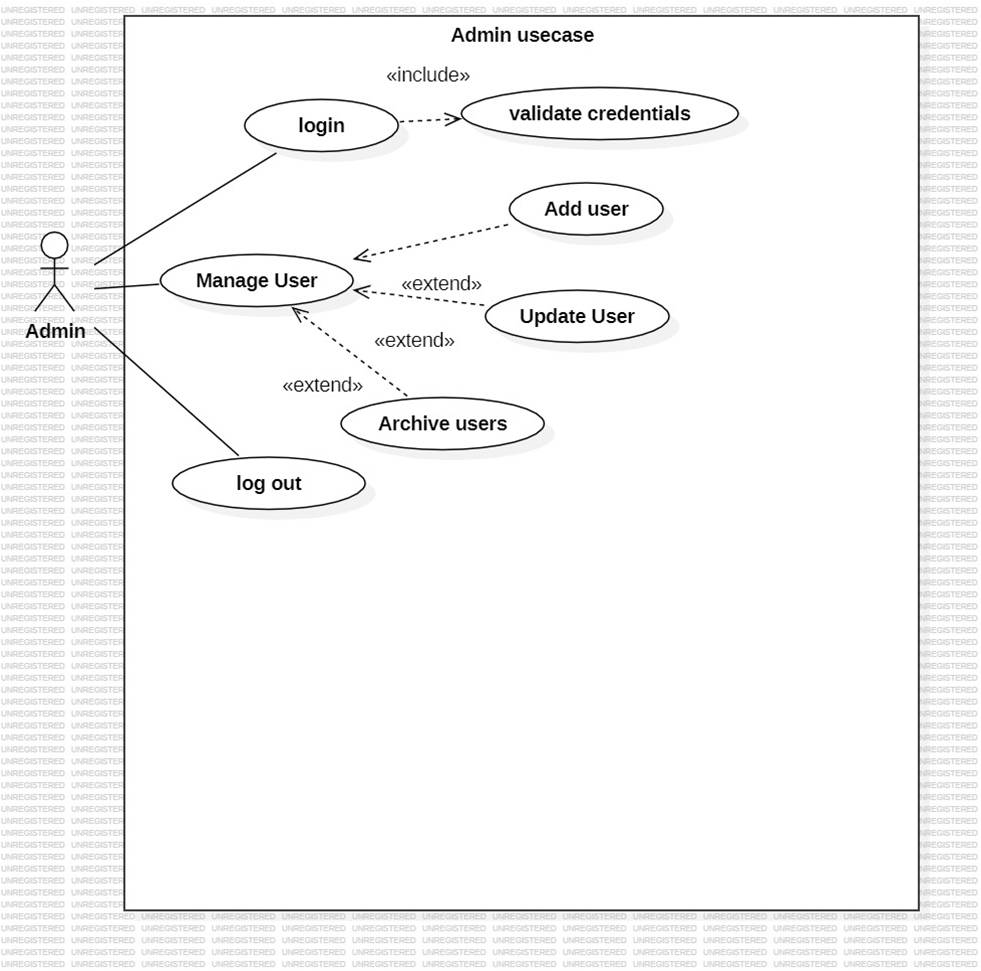
**USER RELATED NON-FUNCTIONAL REQUIREMENTS**

1. Usability Testing: The PDP must undergo usability testing with representative users to ensure that the interface is intuitive, and users can easily navigate through the portal.
2. Data Privacy: Ensure that patient data is encrypted in transit and at rest to maintain patient privacy and compliance with healthcare regulations.
3. Skill levels: The users of the system should have some computer skills. They should know how to navigate through different browsers and have some competency in English language.
4. Training: The users will be oriented on how to use the system through different ways like advertisements of the system.

**4. SYSTEM MODELS**

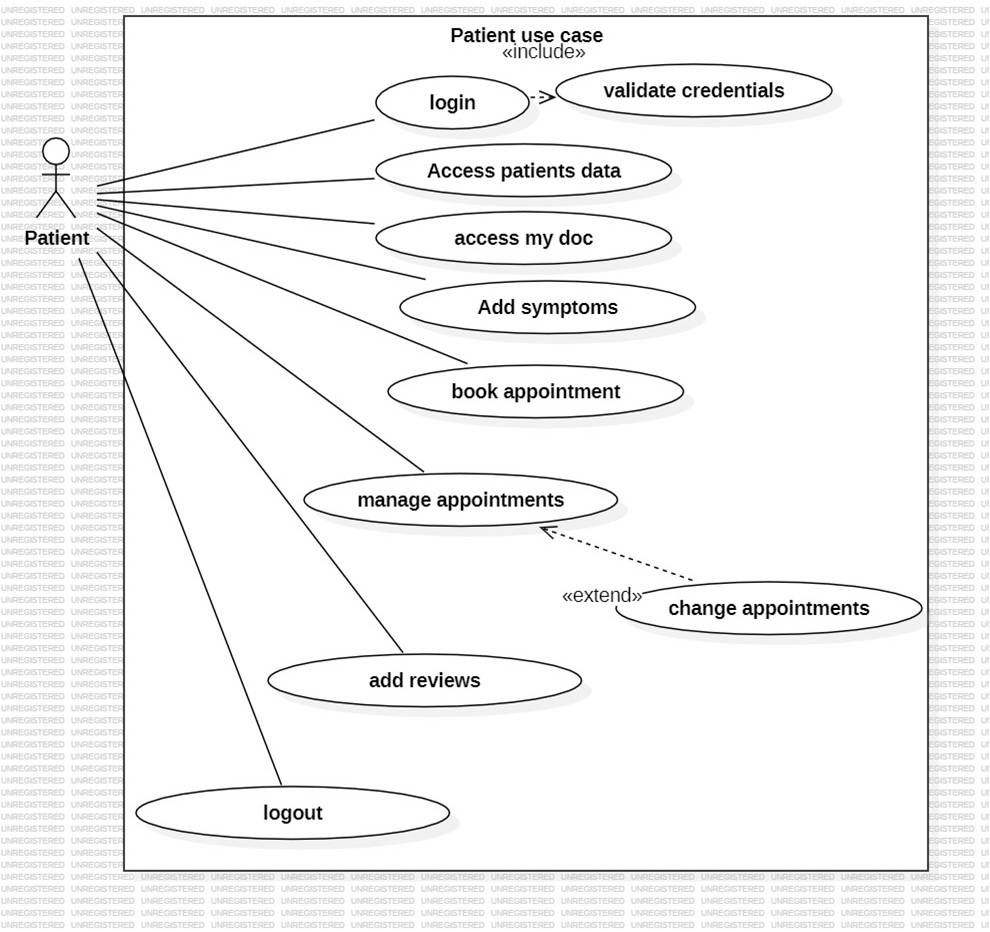
**4.1 USE CASES**

The use cases diagram below illustrates system segments, users and their interaction with the system.

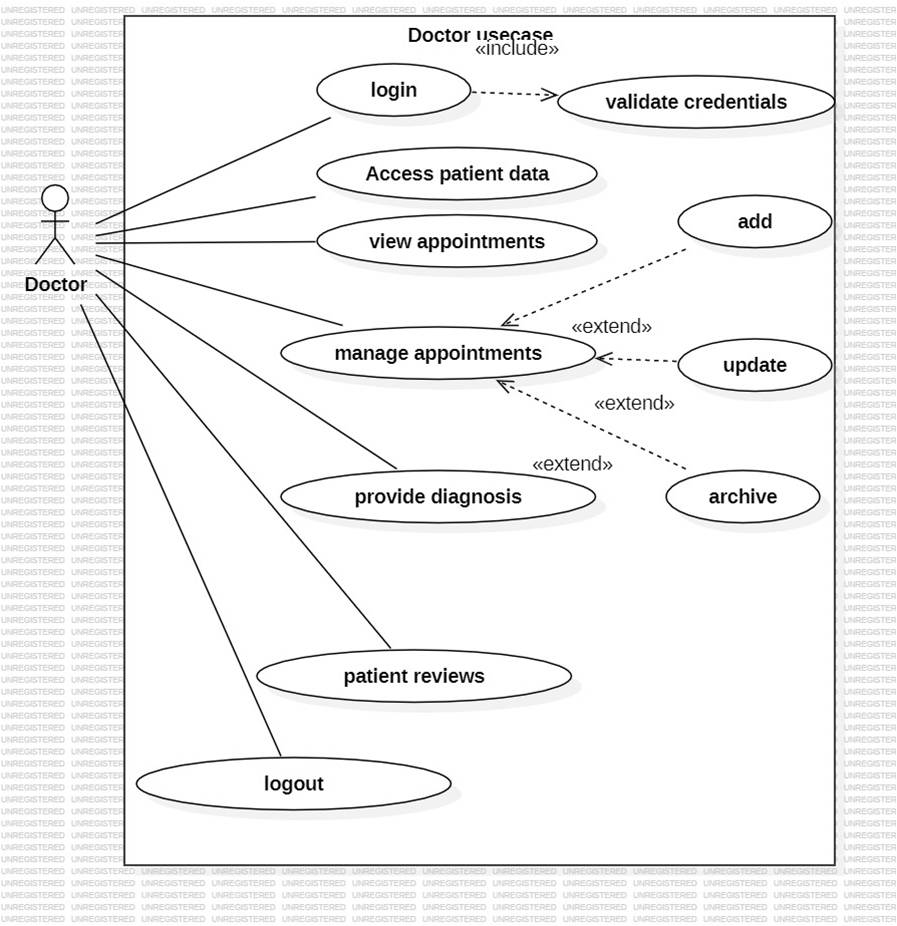
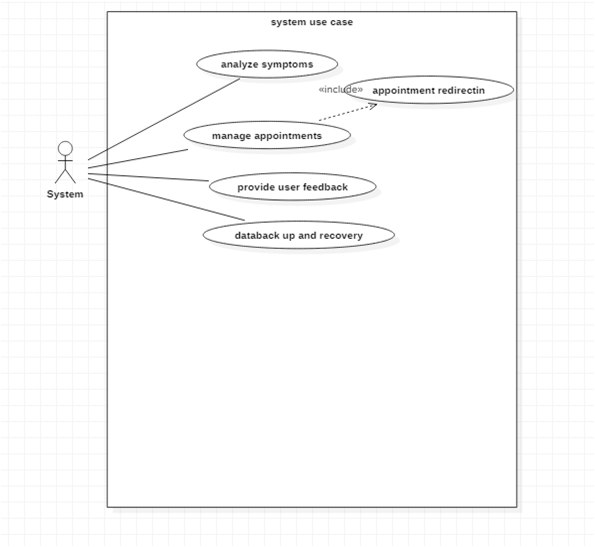
****

**Fig 4.1.1**

This is an administrator’s use case diagram that shows the operations of the admin in the system. Firstly the admin log is then manages users of the system and log out of the system

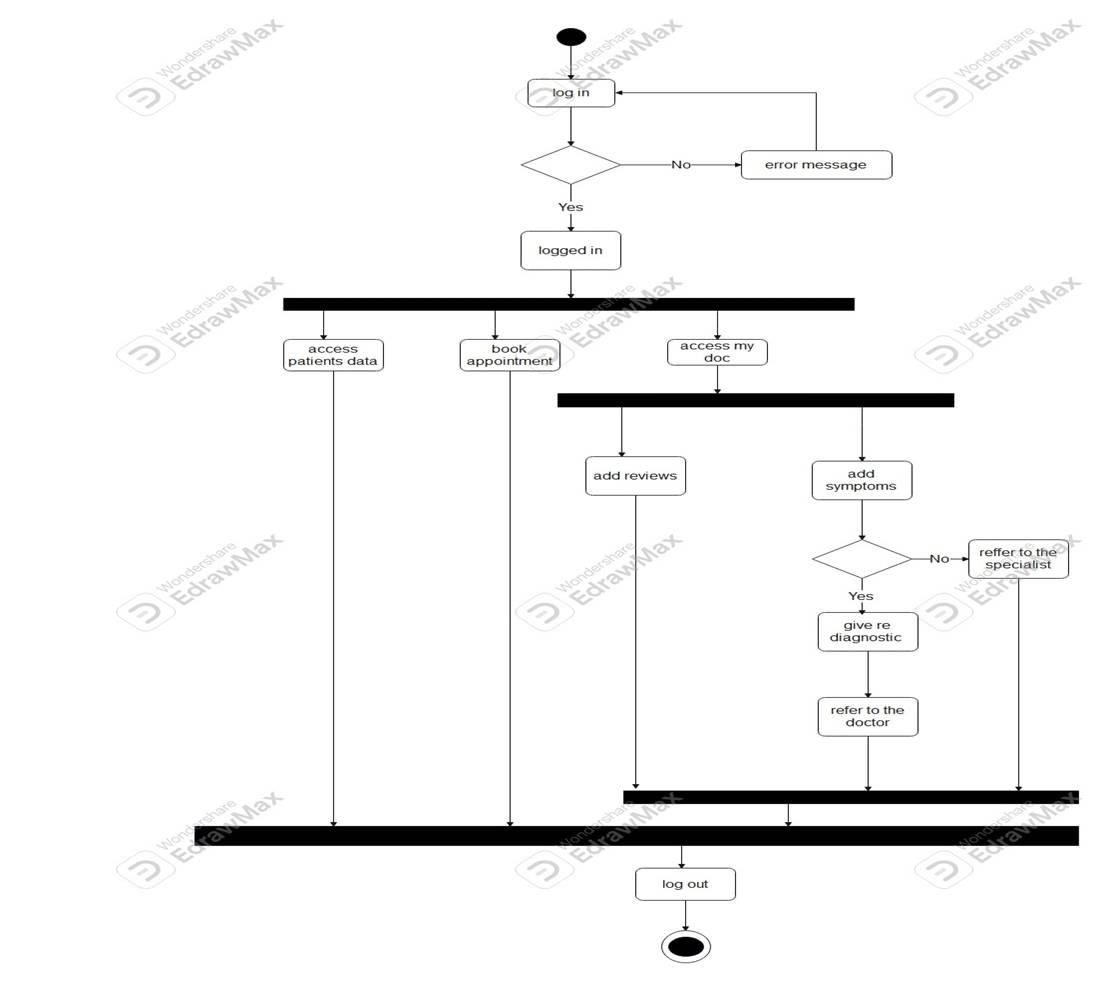
****

**Fig 4.1.2.Patients Use case.** These are the activities done by the doctor in details

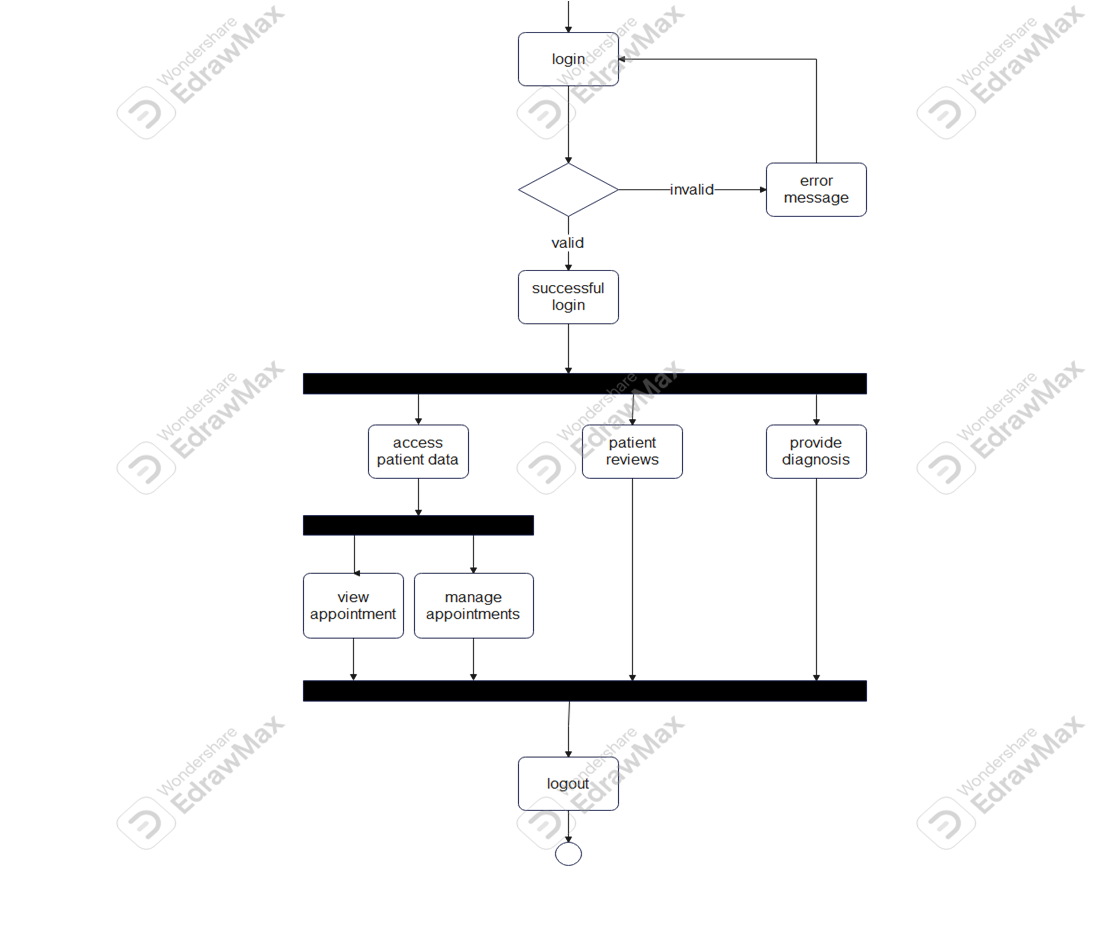
****

**Fig 4.1.4.Doctors Usecase.**These are the activities done by the doctor in details

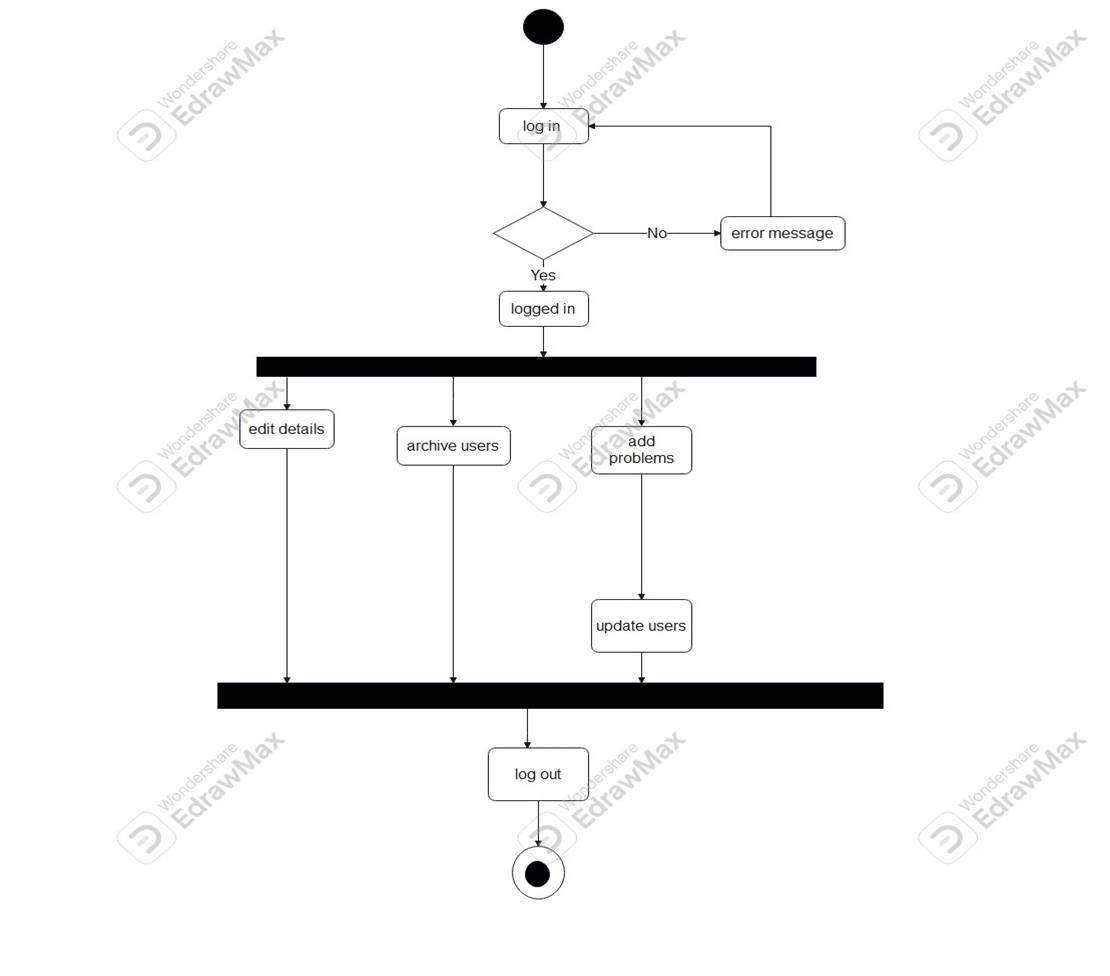
**4.2 ACTIVITY DIAGRAM**

****

**Fig5.Doctors Activity Digram.**Activities done by the doctor

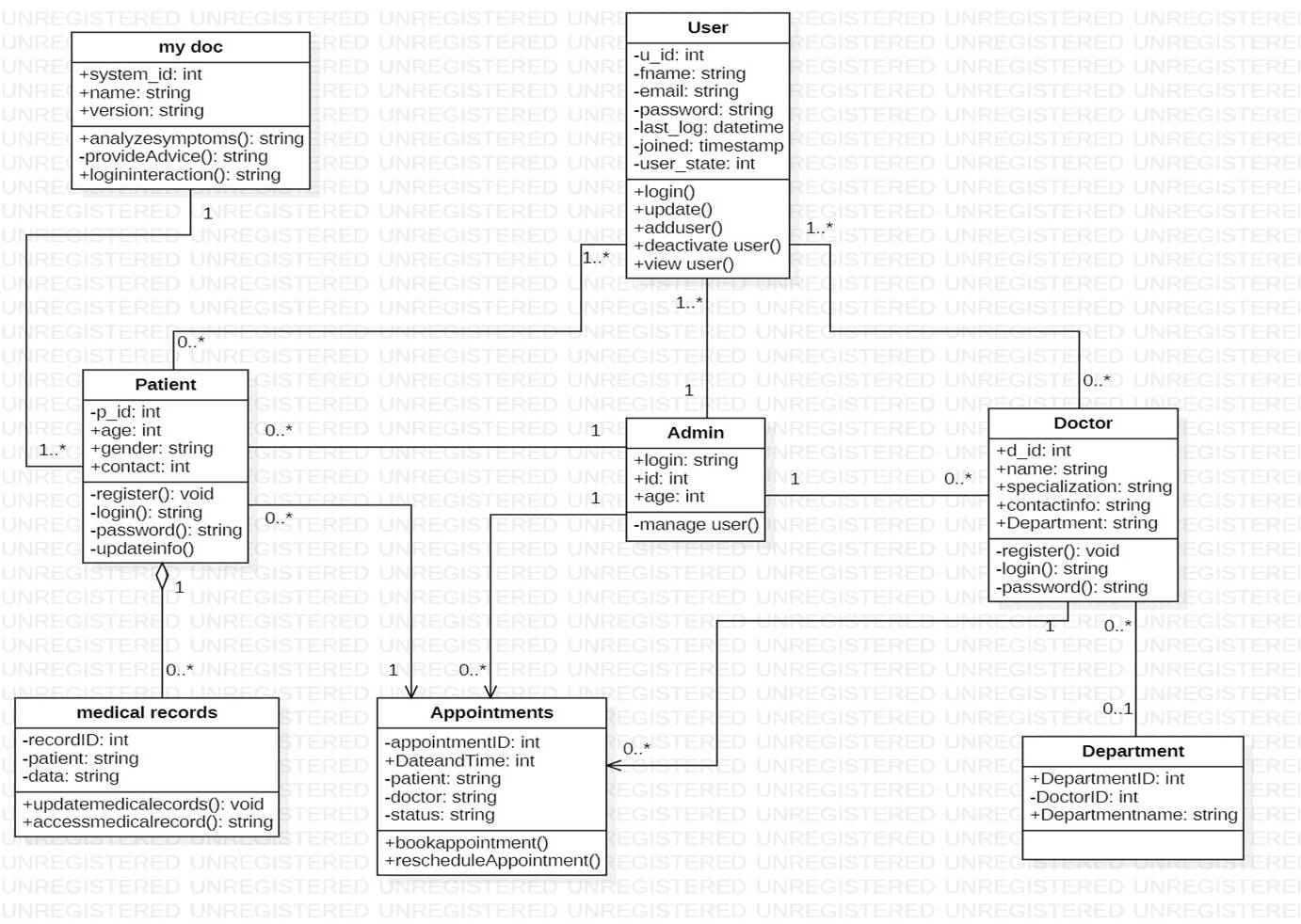
****

**Fig6.Patient Activity Diagram.**Activities done by the patient

****

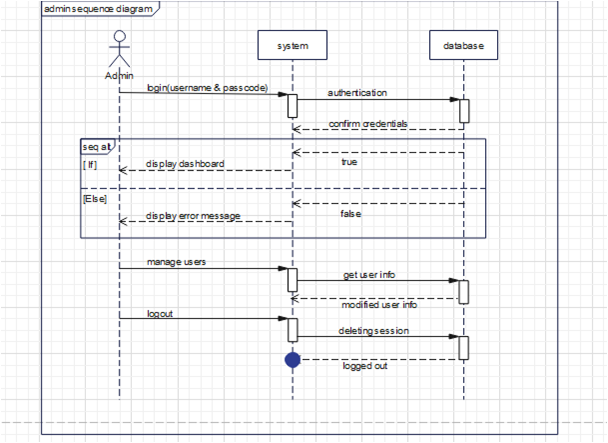
**Fig7.Admin Activity Digram.**Activities done by the adminstrator

**4.3 CLASS DIAGRAM**

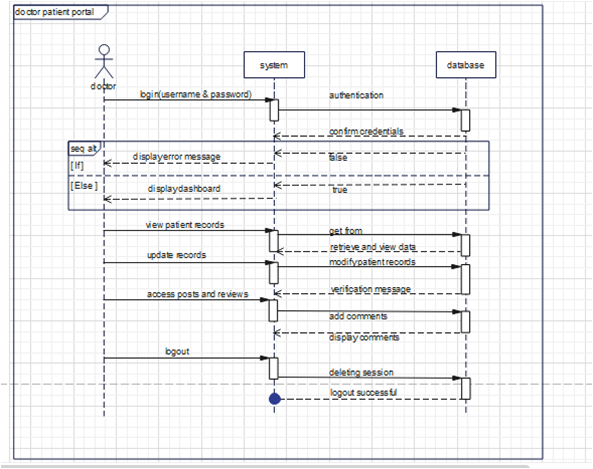
****

**Fig8.**

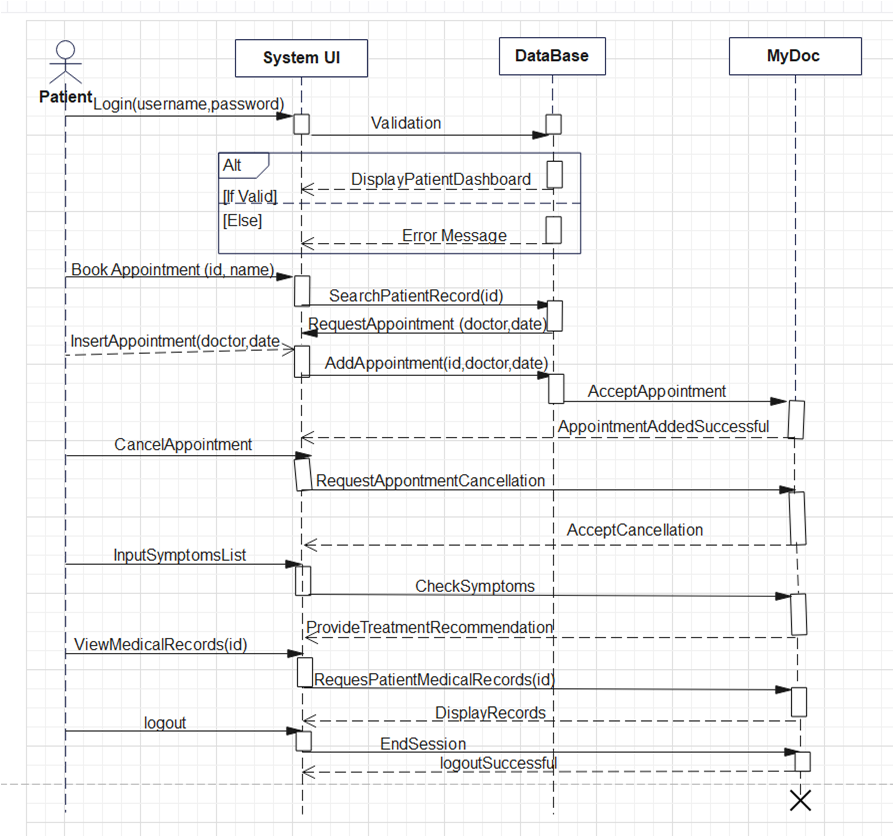
**4.4 SEQUENCE DIAGRAM**

****

**Fig4.4.1.** This sequence applies only to the admin of the system.the password is already defined in the code.

****

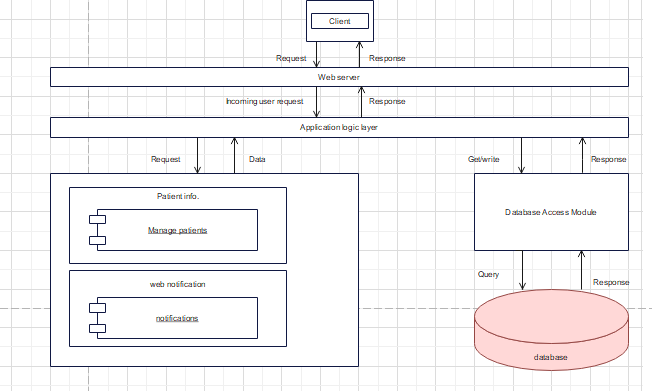
**Fig 4.4.2.** this sequence applies to all doctors using this system

****

**Fig 4.4.3.** this sequence applies to all patients using this system

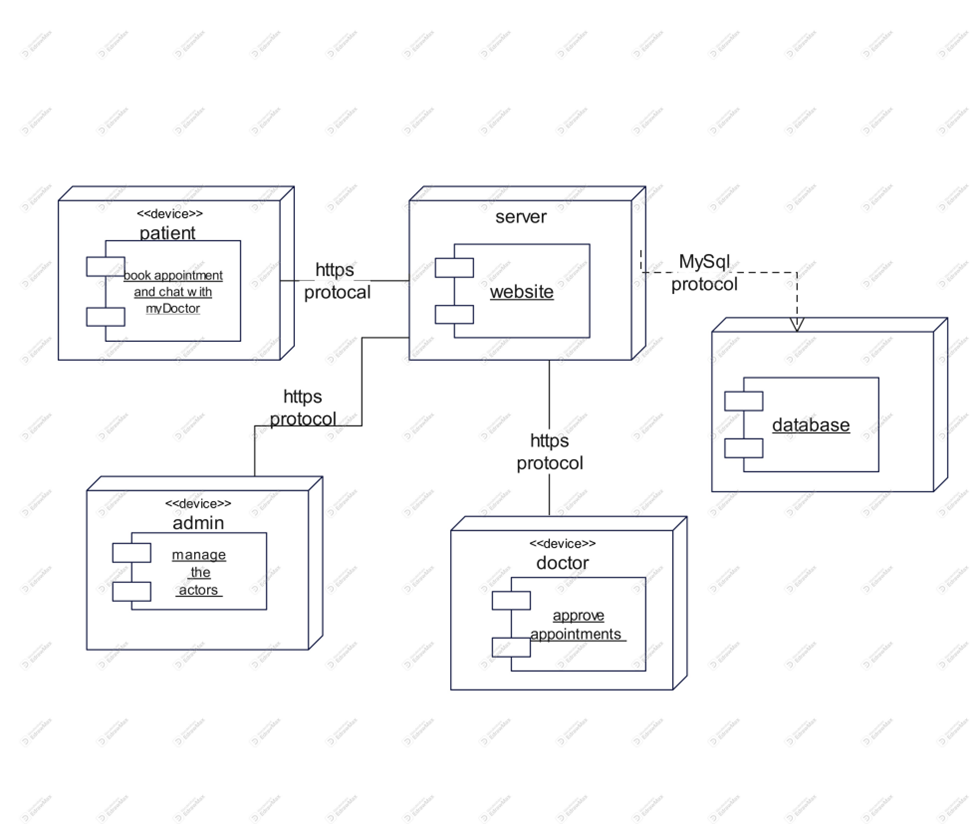
20

**4.5 SYSTEM ARCHITECTURE**

****

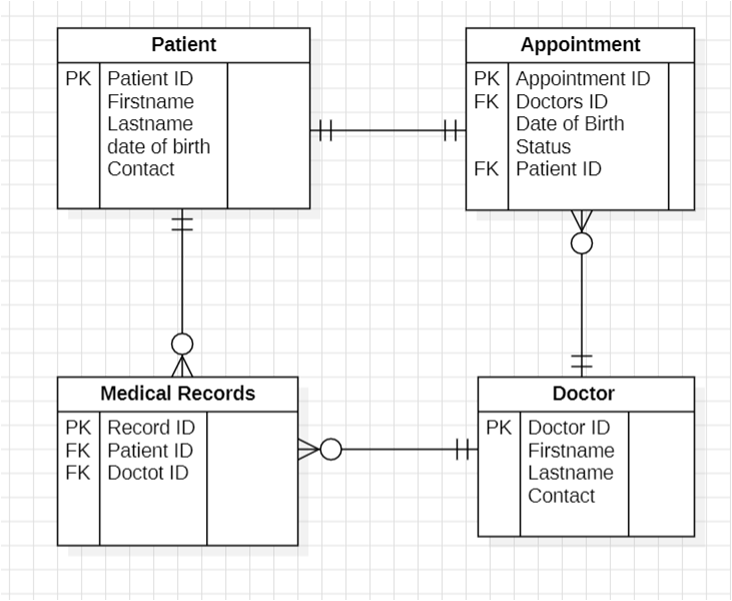
**Fig 4.5.** Shows a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviours of the system. The components designed are functional units and will be implemented as classes.

**4.6 DEPLOYMENT DIAGRAM**

****

**Fig 4.6.**  shows the execution architecture of system, include nodes such as hardware (pc) execution or software (MySQL) execution and middleware (protocols) connecting them

**4.7 ERD**

****

**Fig14.**

**23**

## **5 LANGUAGES AND TOOLS USED FOR THE SYSTEM IMPLEMENTATION**

* To develop Doctor Patient Portal System, we used the following tools and languages:
* PHP, java script and HTML
* Web browser
* MySQL
* Windows 10 OS

1. **INTERFACE REQUIREMENTS**

**5.1Homepage**

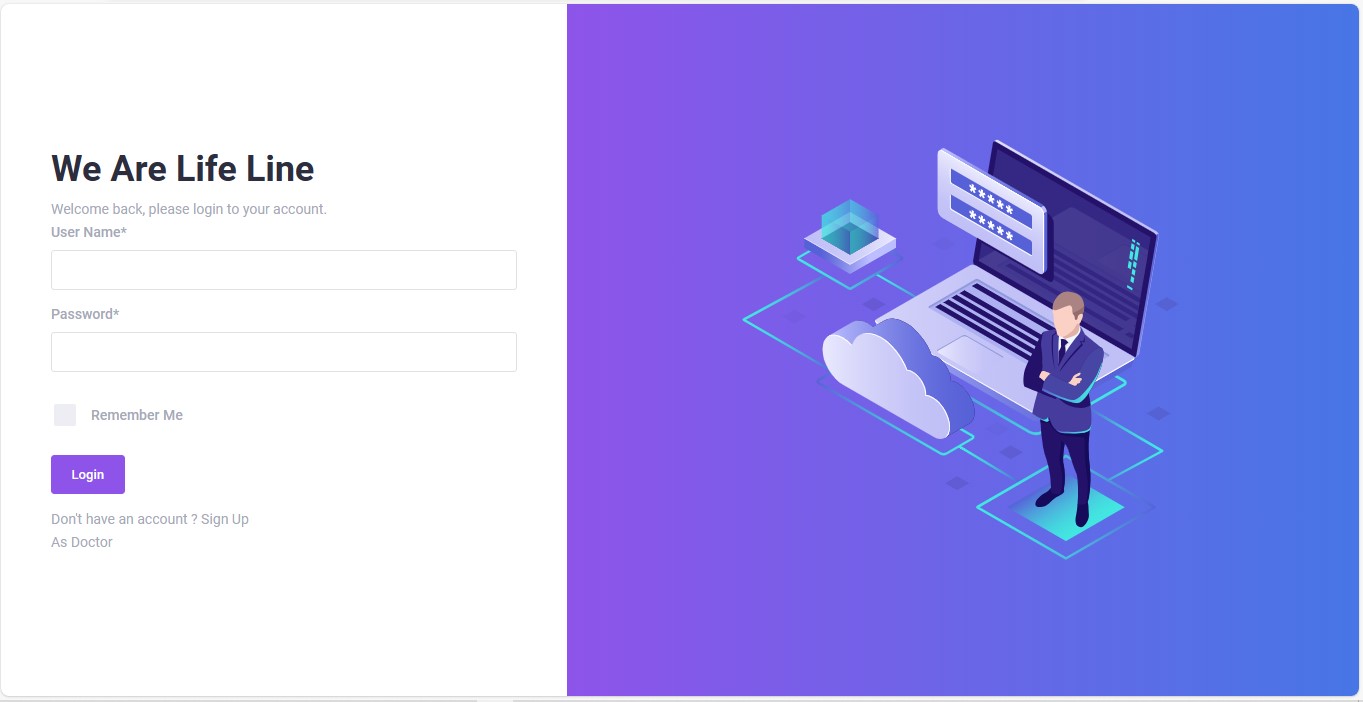
****

Fig 5.1

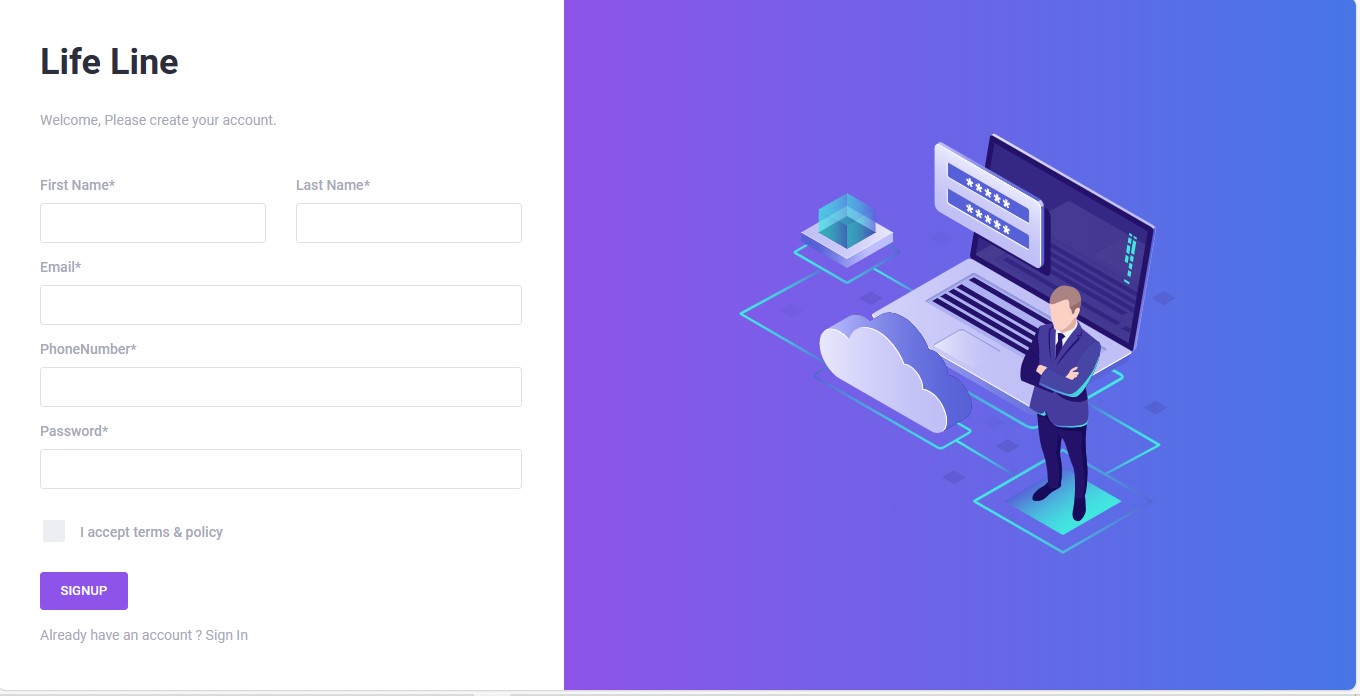
* 1. **Registration page**

Fig 5.2

**5.3Admin Panel**

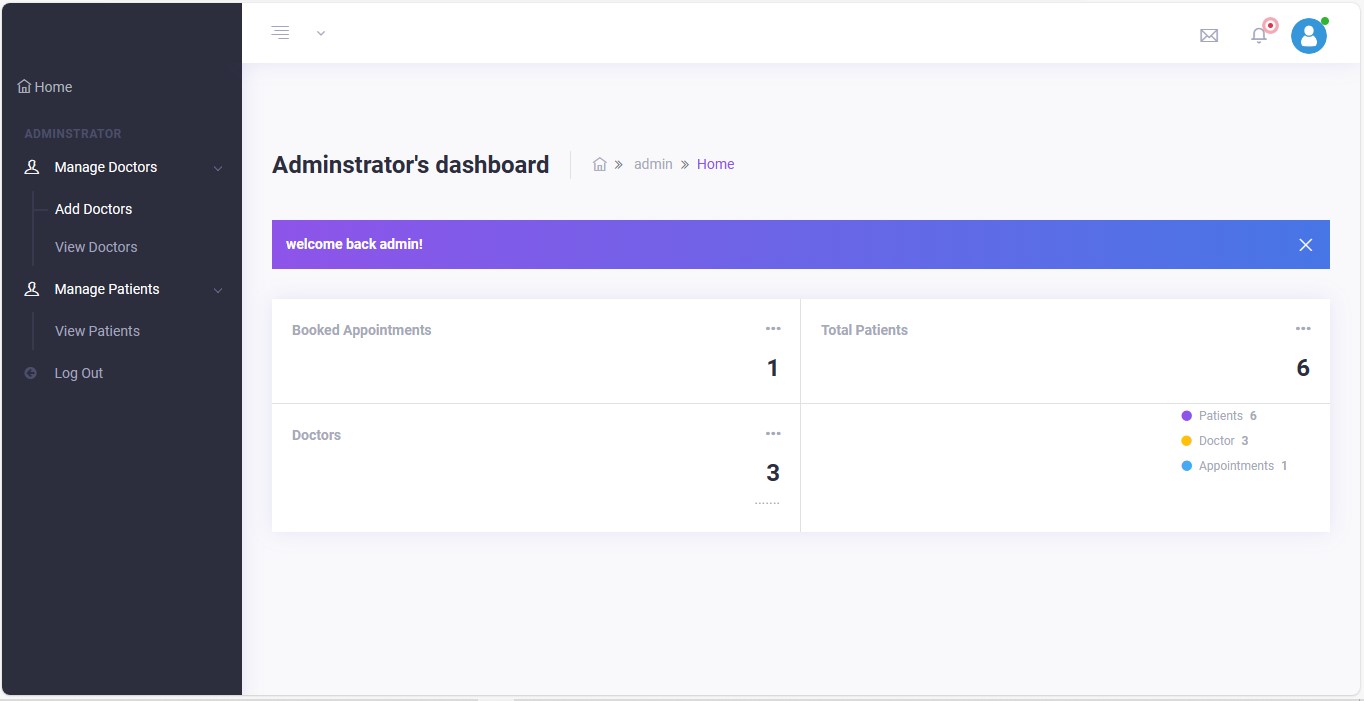


Fig 5.3

* 1. **Add doctor**

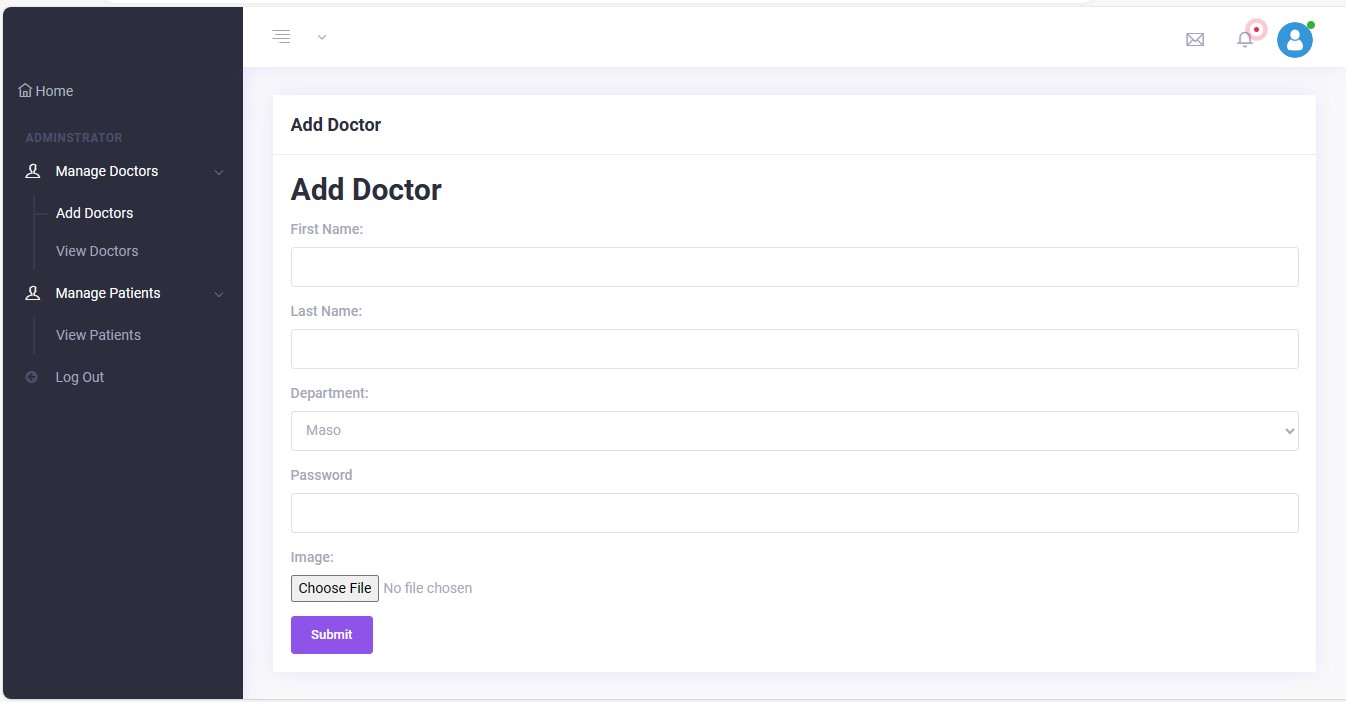
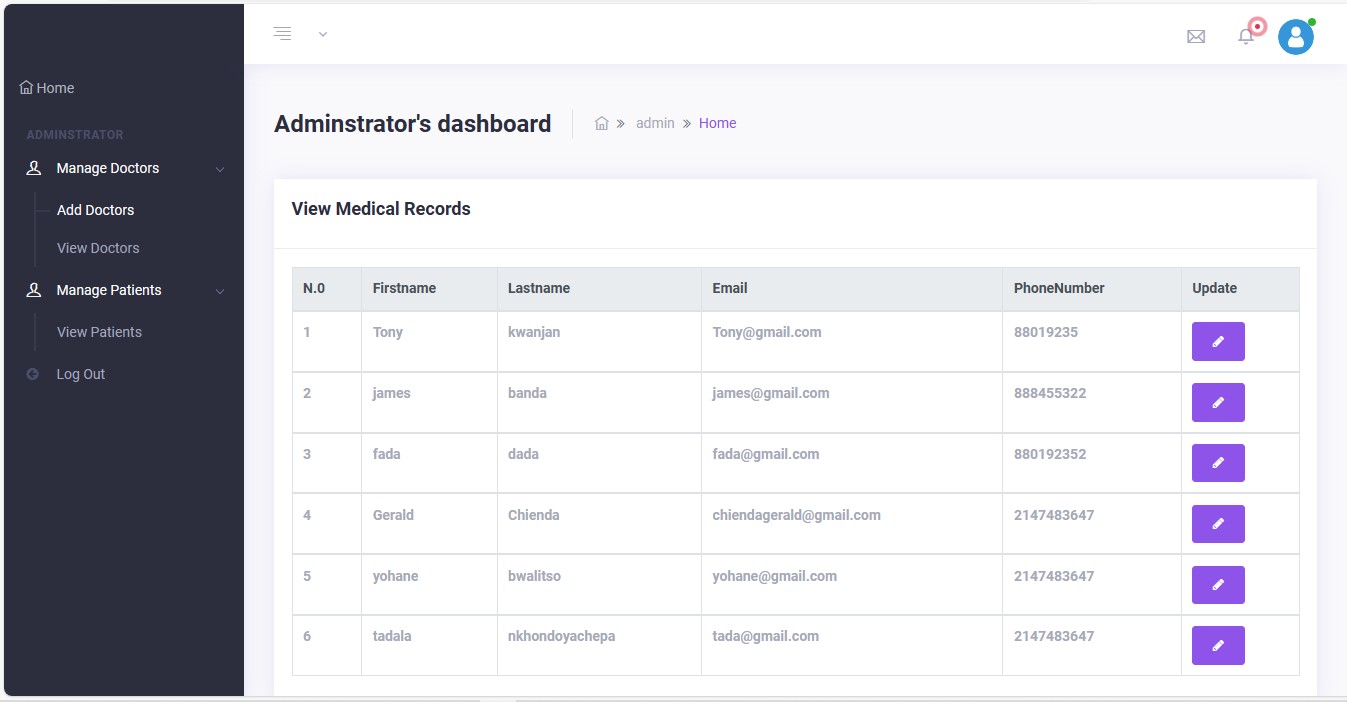


Fig 5.4

* 1. View doctors list



1. **Doctor**

**6.1 Doctor dashboard**

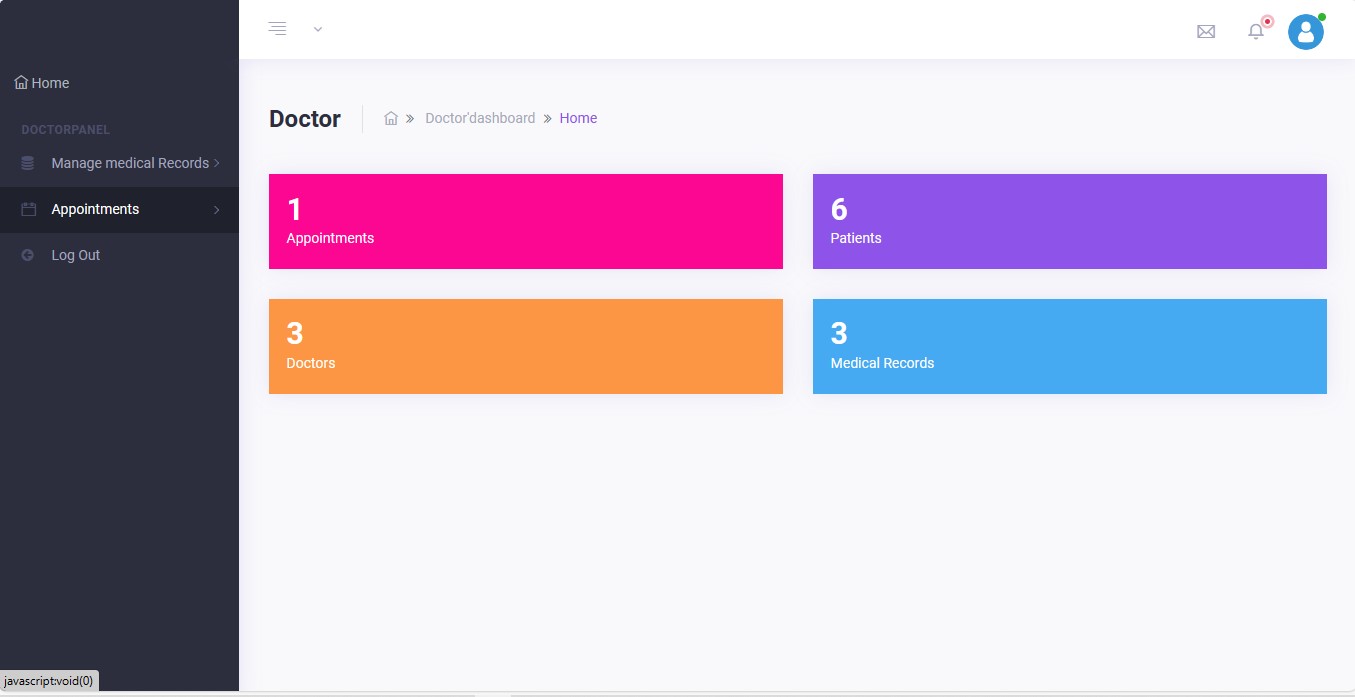


Fig 6.1

* 1. **Doctor Schedule appointments**

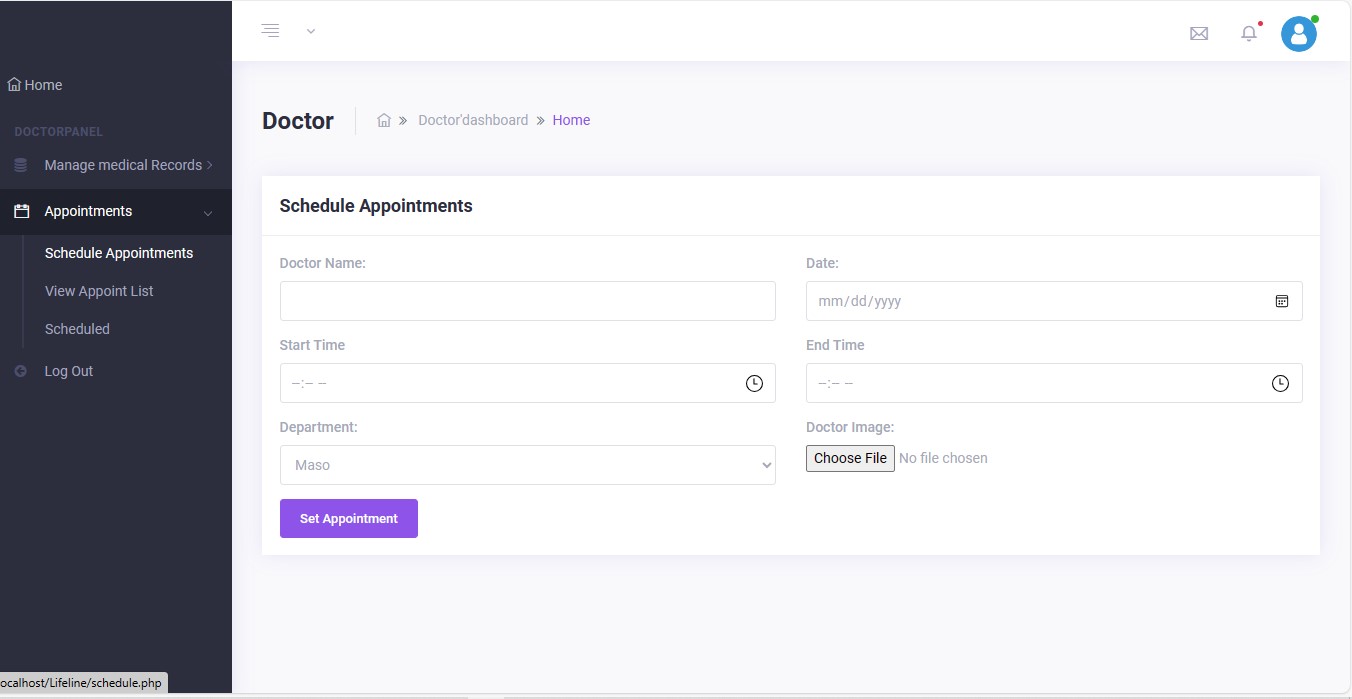


Fig 6.2

**7.Patient activities**

* 1. **book appointment**

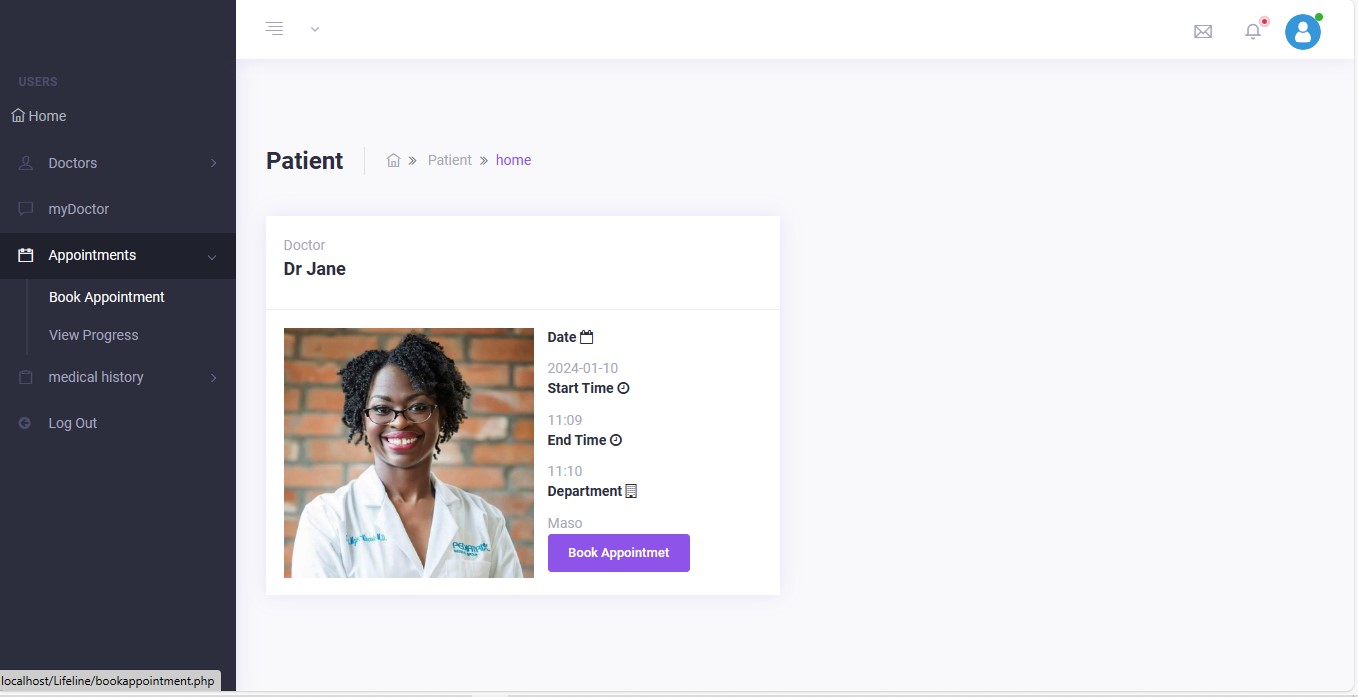


Fig 7.1

7.2 **view** **medical history**

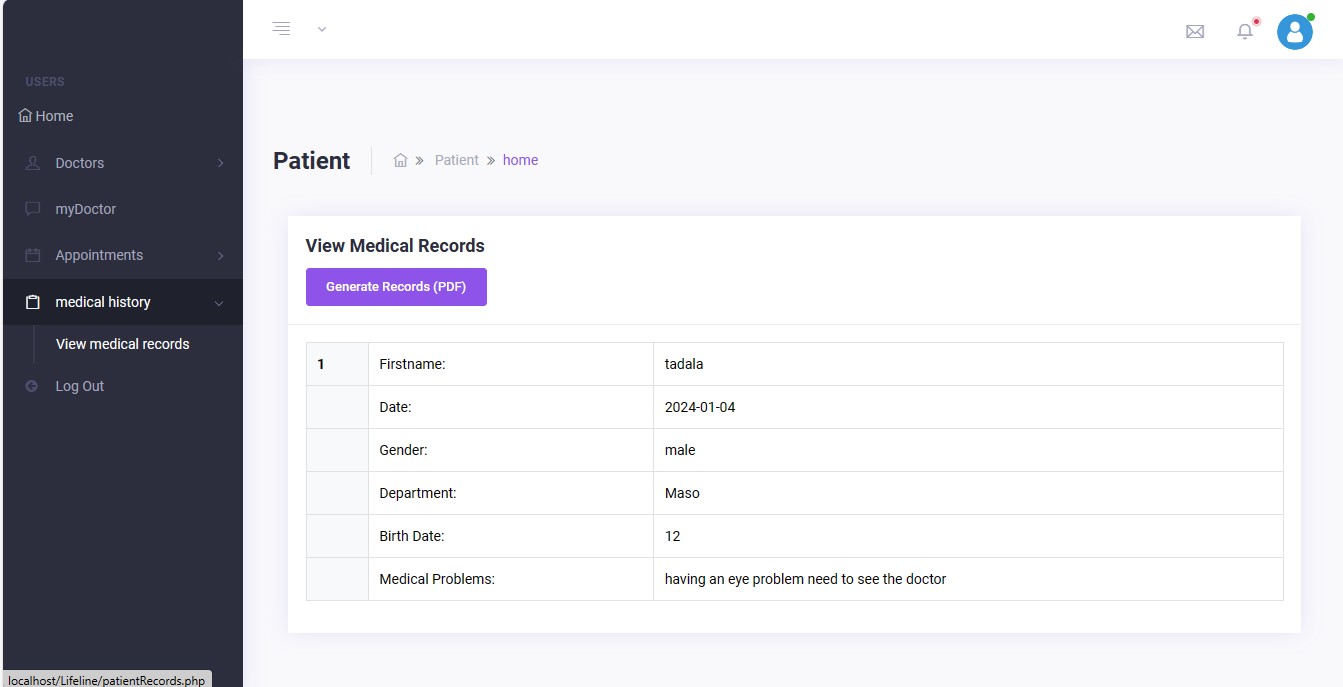


Fig 7.2

* 1. **view doctors available**

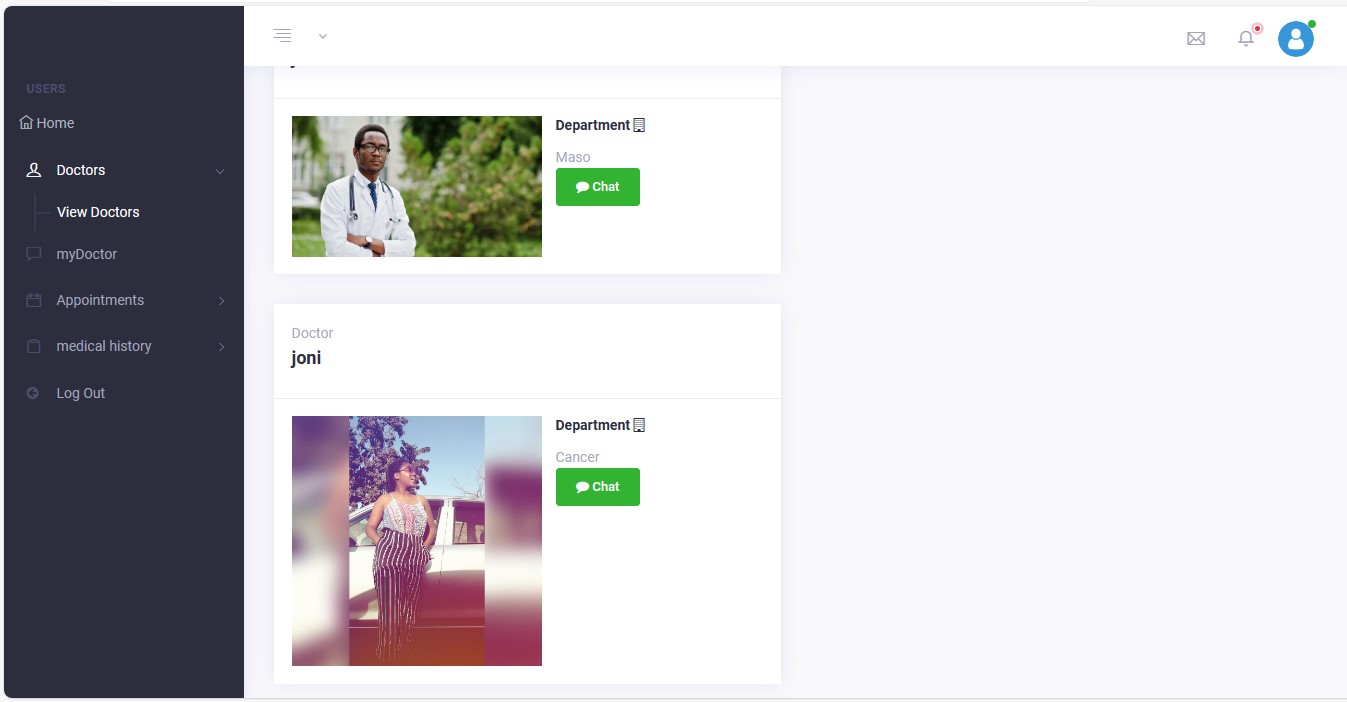


Fig 7.3

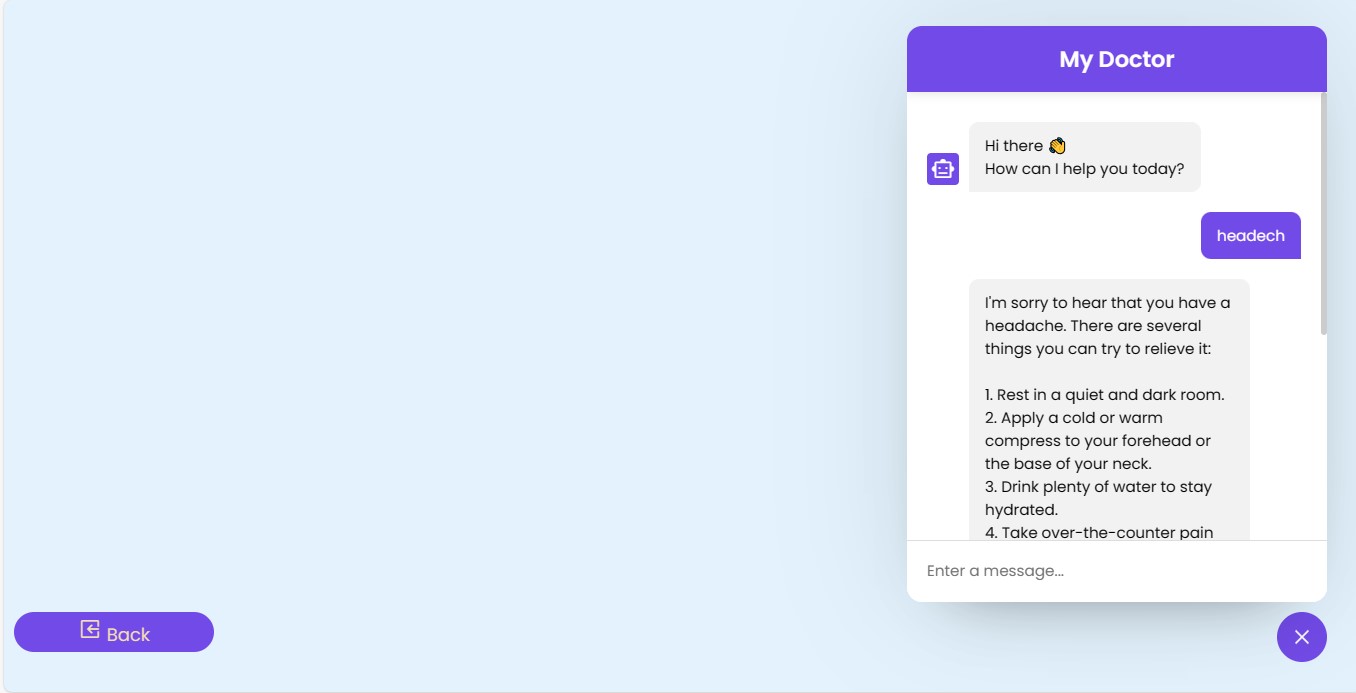


Fig 7.4

**HARDWARE INTERFRACE REQUIREMENTS**

1. RAM

A minimum of 1 GB of RAM is required to ensure that the system has enough memory to run the necessary applications and services smoothly. However, the recommended amount of RAM would depend on the specific hardware interface requirements of the software being used. For example, some software may require a minimum of 2 GB or 4 GB of RAM for optimal performance. Therefore, it's always best to check the system requirements of the specific software being used to ensure that the system has enough memory.

1. Storage

A minimum of 1 GB of storage space may be sufficient for very basic hardware interfaces, but it is generally recommended to have more storage available. This is because software applications, drivers, and other system files can quickly take up space on a hard drive or solid-state drive. Additionally, if the hardware interface requires storage-intensive features such as caching or logging, then more storage space may be necessary. As a general rule, it's recommended to have at least 32 GB of storage for a basic system, but more may be necessary depending on the specific hardware interface requirements.

**6.17 SOFTWARE INTERFACE REQUIREMENTS**

1. Web Browsers: The PDP should be compatible with popular web browsers, including Chrome, Firefox, Safari, and Edge.
2. Mobile Devices: Ensure that the portal is responsive and usable on various mobile devices, including smart phones and tablets.
3. External Systems: Specify the integration points with external systems, such as medical records databases and scheduling software, and define the protocols or APIs used for data exchange.

**CONCLUSION**

In summary, Lifeline is a game-changer in healthcare communication for a specific hospital. The Patient-Doctor Portal introduces “My Doc,” letting patients easily share minor health issues and get quick advice. It ensures timely appointments for persistent problems, making health a top priority.

The platform focuses solely on one hospital, giving patients and doctors easy access to medical records. A review system lets patients share feedback, improving healthcare services continually. With clear goals like efficient symptom checking, accurate expert diagnosis, enhanced patient satisfaction, and optimized resource use, Lifeline aims to transform patient-doctor interactions.

It promises a user-friendly interface, strong data security, and compliance with healthcare rules. Lifeline links with existing healthcare systems, adding better appointment booking and communication features. As Lifeline evolves, its solid foundation promises a patient-centered, efficient, and secure healthcare experience for all

**GLOSSARY**

EDSS - Expert Doctor Substitute System

EHRs - Electronic Health Records

ERD - Entity Relationship Diagram

HTML - Hyper Text Markup Language

GB - Gigabyte

FK - Foreign Key

IT - Information Technology

OS - Operating System

PK - Primary Key

PDP - Patient Doctor Portal

RAM - Random Access Memory