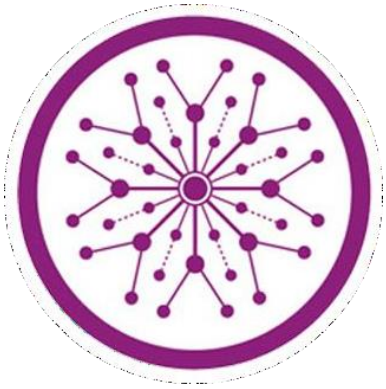
Doctor Appointment System with AI Medical Agent

Session 2020-2024

A project submitted in partial fulfillment of the degree of

BS in Computer Science



Department of Computer Science

Faculty of Computer Science & Information Technology

The Superior University, Lahore

Spring 2024

Type (Nature of project)	[✓] **D**evelopment [ ] **R**esearch [ ] **R&D**
Area of specialization	
FYP ID	FYP-BCSM-F23-060

Project Group Members

Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	BCSM-F20-135	M Awais	BCSM-F20-135@superior.edu.pk	Awais
(ii)	BCSM-F20-190	Muhammad Sohaib	BCSM-F20-190@superior.edu.pk	Sohaib
(iii)	BCSM-F20-152	Ali raza gul	BCSM-F20-152@superior.edu.pk	Ali

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I M Awais S/D of Muhammad Iqbal, group leader of FYP under registration no _____at Computer Science Department, The Superior University, Lahore. I declare that my FYP report is checked by my supervisor.

Date: Name of Group Leader: M Awais Signature: _____Awais_____

Name of Supervisor: Ms Hafiza Maria

Designation: Lecturer

Signature: _____

HoD: Irfan-ud-din

Signature: _____

Doctor Appointment System with AI Medical Agent

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
	1.0		<Original Draft>	
			<Changes Based on Feedback from Supervisor>	
			<Changes Based on Feedback From Faculty>	
			<Added Project Plan>	
			<Changes Based on Feedback from Supervisor>	

APPROVAL

PROJECT SUPERVISOR

Comments: _____

_____

Name: _____

Date: _____ Signature: _____

PROJECT MANAGER

Comments: _____

_____

Date: _____ Signature: _____

HEAD OF THE DEPARTMENT

Comments: _____

_____

Date: _____ Signature: _____

Dedication

Here, we come up with an idea of the project name which is dedicated to unrelenting quest for reforming healthcare. Every line of code, design choice and strategic decision made was fueled by our unwavering commitment to leveraging technology for improved patient care. This project is a statement of how devoted a team is that wants to reshape the future of medical appointments through AI integration.

This project's aim is for healthier, more accessible and patient-centric future in tribute to all health professionals working towards excellence, all patients looking for personalized attention and care as well as all team members who offered their expertise. Our dedication will help us find new creative ways of solving health issues improving lives globally.

Acknowledgement

My invaluable guidance and mentorship throughout the development of this project are so much appreciated by my supervisor Ms Maria Iqbal. Their constructive suggestions and insightful feedback has been the driving force behind the quality and direction of the work.

I would also like to extend my thanks to Ms Mehreen whose contributions and collaboration significantly enriched the project. Their dedication and expertise played a important role in challenges and achieving goles.

A collaborative and innovative environment was nurtured by my colleagues support as well as those of the entire project team.

Finally, heartfelt thanks to my friends and family for their understanding, encouragement, and motivation during the course of this endeavor.

This project is an outcome of the effort made by a group of amazing people who I feel privileged having had them on board.

Executive Summary

A groundbreaking solution to healthcare accessibility and appointment management is the new web-based doctor appointment system using MERN stack and an ai agent.this scheme brings state of the art technology face to face with medical know-how in order to facilitate chats between patients and doctors. This enhances user experience by making it easy for people to book their appointments hence saving them time which they use otherwise scheduling. Patients can equally avoid waiting lines through our system while at the same time getting basic knowledge on how to lead a healthy life

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 1: Introduction

The online appointment booking system for medical appointments allows patients to book appointments at specific clinics to meet their health needs. For instance, Dental, Trauma, Mental Health, Strain of Muscle, Obesity, Arthritis, Asthma and many others. There are many people with multiple diseases (about 95 percent of the world's population). Every week there are individuals who plan appointments with local doctors for needed medical care. Adding a doctor's appointment booking system on your practice or doctor's website can bring huge benefits. This involves a booking form that has several sections that patients must fill out and submit in order to make an appointment. Not only does this simplify the process for patients but also for healthcare providers An appointment-booking form that entails many fields to be completed by the patient after filling it up and submitting it is what it is about. The whole process consumes much of the time of both the patient as well as you.

AI Medical Agent: Our platform has an advanced AI medical agent, which helps patients in medical information and giving relevant advice. This means the AI agent acts as a human being while interacting with patients to ensure their needs are met during appointment booking process.

How It Works:

***Registration*:** Patients create their account first on our platform, providing basic information and past medical records

***AI Interaction*:** Patient chatting like when patients talk about their illness and other guidance

***Medical Information*:** Patients may also use the AI agent to pose questions on health matters, knowledge of drugs and general advice on health.

The Doctor Appointment System is an overview of the project, which is based on the application of a medical expert system that uses artificial intelligence. It summarises what this research aims at and explains why it's important; how it can be achieved and finally gives a briefintroduction to its functionalities. This chapter then goes ahead to give a synopsis of what the system wants to attain, its importance as well as features guiding users through subsequent sections in this paper.

1.1. Background

The healthcare industry is forever changing for the better with access to information, efficiency and patient care as major goals. this is because:

Complexity in Appointment Management: Unlike traditional systems, these systems are not flexible and lack real-time updates hence putting patients and practitioners at crossroads.

Gaps in Communication: Inefficiency due to lack of enough communication networks between doctors who attend to patients hindering service delivery timeline.

Accessibility and Convenience: Healthcare facilities have limited access to patients since thereare few appointment processes that hinder many people from accessing it immediately.

Simplify Appointment Management Process: An easy interface enabling patients to schedule appointments easily without needing much effort.

Helping in Communication: Such channels create smoother platforms where patients can inquire about an appointment, other questions they might be having or any health problems.

Increasing Accessibility: This means deploying a platform that ensures prompt reach of healthcare services thereby improving the patient experience.

1.2. Motivations and Challenges

Motivations:

1. ***Improved patient experience*:** This system is primarily developed to prioritize convenience ofpatients in order to make scheduling appointments seamless and reduce waiting time.

2. ***Integration of Technological Innovation*:**AI-Driven Solutions: AI technology use, providing intelligent support and enhancement of healthcare for the sake of both patients and healthcaregivers

Challenges:

1. **Appointment Conflicts:** Overlapping appointments are often scheduling conflicts that shouldbe resolved and ensuring ease management of doctor's availability

2. **Data Security and Privacy** The strict security measures have been put in place so as to protect the patients' data as well as comply with the established regulations in regards to the privacy ofhealth care data.

1.3. Goals and Objectives

1 **Better user experience:** Provide a platform where users can schedule, revise or cancelappointments with ease.

2 **Smart Assistance:** A.I. agent integrated into the digital interface assisting users virtually byoffering preliminary information about certain medical conditions

Objectives:

Implement AI for Medical Consultation: Develop an Artificial Intelligence (AI) powered system that can provide answers to basic inquiries about health issues.

Ensure Accuracy and Reliability: Means for ensuring accurate medical information should be put in place.

1.4. Literature Review/Existing Solutions

A survey of existing healthcare appointment systems revealed a need for streamlined appointment scheduling system and improved patient doctor communication. current healthcare solutions often lack robust integration into appointment systems presenting an opportunity for our system ai agent to facilitates patients intelligently

Gap Analysis

Identified Gaps:

1 ***The need for a simplified version*:** The systems that already exist are not user-friendly, and theydo not have Ai assistant service

2 ***Improved Interaction Platforms*:** Currently the existing ones have poor interaction interfacesbetween patients and doctors, hence the need for a better one.

3 ***Accessing Efficiency in Care*:** This is accomplished by closing the distance between patients' demands and timely healthcare providers through optimization of appointment availability and accessibility.

Addressing the Gaps:

1 ***User Friendly Design*:** To create an interface that will make it possible to easily accessappointments and modify them whenever necessary.

2 ***Uninterrupted Interaction Capabilities*:** In this regard an effective patient doctor communication. system will be built in into the platform which has been implemented tooperate properly.

3 ***Simplified Appointment Procedures*:** It is vital to streamline these processes so that people canget medical treatment without any delays.

1.5. Proposed Solution

we propose a online booking system that is both user friendly and efficient to take care of the difficulties of scheduling doctor appointments.This will be done through harnessing technology and making the process easy so that there are no challenges for patients and service providers.

1.6. Project Plan

Project Initiation:

1 Define Project Scope and Objectives.

2 Conduct Market Research and Analysis.

3 Develop Project Charter.

System Analysis and Design:

2.6.1 Requirements Gathering and Analysis

2.6.1 Architectural design & Prototyping

2.6.1 UI/UX Design & Wireframing

Development Phase:

1. Front-end development
2. Back-end Development and Database Integration.
3. AI Agent integration & Testing.

Testing and Quality Assurance:

1. Debugging & Unit testing
2. Integration Testing
3. UAT (User acceptance testing)

Deployment and Launch:

1. Deployment on Hosting Server (e.g Heroku)
2. Final System testing & bug fixing
3. Launch preparation, marketing collateral

1.6.1. Work Breakdown Structure

Final Documentation Introduction

This is a comprehensive document that will unveil the intricacies and technical designs of our innovative Doctor Appointment System. This system illustrates how recent advancement in technology can be combined with healthcare through the MERN stack MongoDB, Express.js React and Node.js and AI agent respectively that has changed the patient-doctor interaction.

Project Overview:

We were aiming to build a sophisticated but user friendly solution that changes how people interact with healthcare. this documentation covers all the painstaking steps followed by our team to develop an intuitive AI-enabled system which is a scheduling tool as well as an intelligent assistant for patients.

Key Components:

MERN Stack Foundation: The choice of MERN stack as a foundation guarantees a responsive strong system that can scale easily. MongodB serves like a supple datastorage, Express.js and Node.js drive back-end logic while React allows dynamic front- end design.

AI Agent Integration:It significantly enhances the interactivity of the system that makes it smarter and faster during user interaction. Invaluable advice in medicine is providedby this agent.

Purpose of Documentation:

The maze of our system's architecture, functions and deployment plans for all stakeholders is what this documentation seeks to guide you through. Each section uncovers a piece of our careful design, development and implementation journey.

Literature/Market Survey:

To make the appointment scheduling process in todays healthcare systems more streamlined and to enhance the relationship between doctors and patients there is need to review the existing healthcare appointment systems. our AI-based on an intelligent algorithm can assist in solving problems of many health care products that donot easily integrate into appointment management systems

Requirements Analysis:

Functional Requirements:

1. user login and password
2. AI supported chatbot for health related information.
3. Immediate appointment reminder service

Non-Functional Requirements:

1. Patient data privacy assurance through security measures.
2. Increase in the number of users and appointment loads.
3. Simple user interface that can be easily used by anyone

System Design:

Architecture:

1. The front end design uses react to enable dynamic user interfaces
2. The backend is built using Node.js and Express.js as a Rest Full API
3. MongoDB as the database provider for storing and retrieving data
4. Intelligent assistance by invoking api calls to an ai-agent

Database Design:

Patient profiles, doctor details, and appointment schedules stored in MongoDB collections

AI Agent Integration:

AI agent integrated via API endpoints to assist users in providing basic medical information

Implementation:

Development Process:

In order to manage versions, the agile development methodology was adopted Git.

The biggest challenge was how to merge AI Agent responses into the appointment booking flow within MERN stack development, as there is an organized file structure for this purpose

Testing & Performance Evaluation:

Testing Strategy:

Validation of functionality through unit tests for individual components, API integration tests and user acceptance testing

Performance Metrics:

Response times during appointments bookings and AI agent interactions are used in evaluation of system performance

Conclusion & Outlook:

1. **Project Summary:**

   Developed a doctor's appointment system that integrates AI assistance's for smartness

2. **Future Enhancements:**

   Exploring natural language processing to enrich ai agent responses; video consultations;wearable health devices integration for data input 2.6

End User Documentation:

User Guide:

How to register book appointments interact with the AI agent, and manage them.

Application Administration Documentation:

Admin's guide:

Managing doctor's details, appointment timetable, setting up and reviewing the systemsas well as retrieving data.

System Administrator Documentation:

1. ***Introduction:***

   The duties of a system administrator and how he/she maintains the efficiency of thesystem.

2. ***System Overview:***

   A brief summary of the design of this system along with constituent components thatrun it.

3. ***Maintenance & Security:***

   Obligations regarding routine support, backup together with protection means.

4. ***Upgrades & Changes:***

   Methods for controlling changes made in order to upgrade a system.

5. ***Support & Troubleshooting:***

   Handling breakdowns, providing technical assistance and solving problems.

System:

Development Environment

1. ***IDE:*** IDE: Visual Studio Code for frontend (React) and backend (Node.js)
2. ***Version Control****:* Version Control: Git for version control, repository hosted on GitHub
3. ***Server:*** Server: Node.js server hosted on AWS EC2 instance
4. ***Database****:* MongoDB Atlas for cloud-based database hosting

Presentation Layer

1. ***Deliverable 1:*** React components for user authentication and profile management
2. ***Deliverable 2:*** Appointment scheduling interface with interactive calendar view
3. ***Additional :*** User interfaces dealing with AI agent interaction and appointment notifications

Business Logic Layer

1. ***Deliverable 1*:**RESTful API endpoints for user authentication and profile management
2. ***Deliverable 2:*** AI agent integration logic, appointment scheduling handling logic, userqueries handling logic

Data Management Layer

1. ***Deliverable 1****:* MongoDB schema design for storing user profiles and appointmentdata
2. ***Deliverable 2:*** Implementation of data models and CRUD operations interacting withthe database

Physical Layer

1. **Deliverable 1:** Deployment of frontend React app on AWS S3 bucket

1.6.2. Roles & Responsibility Matrix

Table 1 Roles & Responsibility Matrix

WBS #	WBS Deliverable	Activity #	Activity to Complete the Deliverable	Duration (# of Days)	Responsible Team Member(s) & Role(s)
1	Frontend Development	1.1	UI Design and Wireframing	7	Ali RazaGul (Frontend Developer)
1	Frontend Development	1.2	Implement React Components	10	Ali Raza Gul (Frontend Developer)
2	Frontend Development	2.1	Set up Node.js Server	3	M Awais (Backend Developer)
2	Frontend Development	2.2	Develop RESTful APIs	10	M Awais (Backend Developer)
3	Database Setup	3.1	Configure MongoDB Atlas	2	M Awais (Database Administrator)
3	Database Setup	3.2	Design MongoDB Schema	7	M Awais (Database Administrator)
4	AI Integration	4.1	Medical AI Agent Integration	10	Sohaib (Frontend Developer)

4	AI Integration	4.2	Implement API Calls for AI Integration	10	Sohaib (Backend Developer)
5	Testing & Optimization	5.1	Unit Testing of Frontend Components	15	Ali Raza Gul (Frontend Developer)
5	Testing & Optimization	5.2	API Testing and Integration Testing	15	Ali Raza Gul (Backend Developer)

1.6.3. Gantt Chart

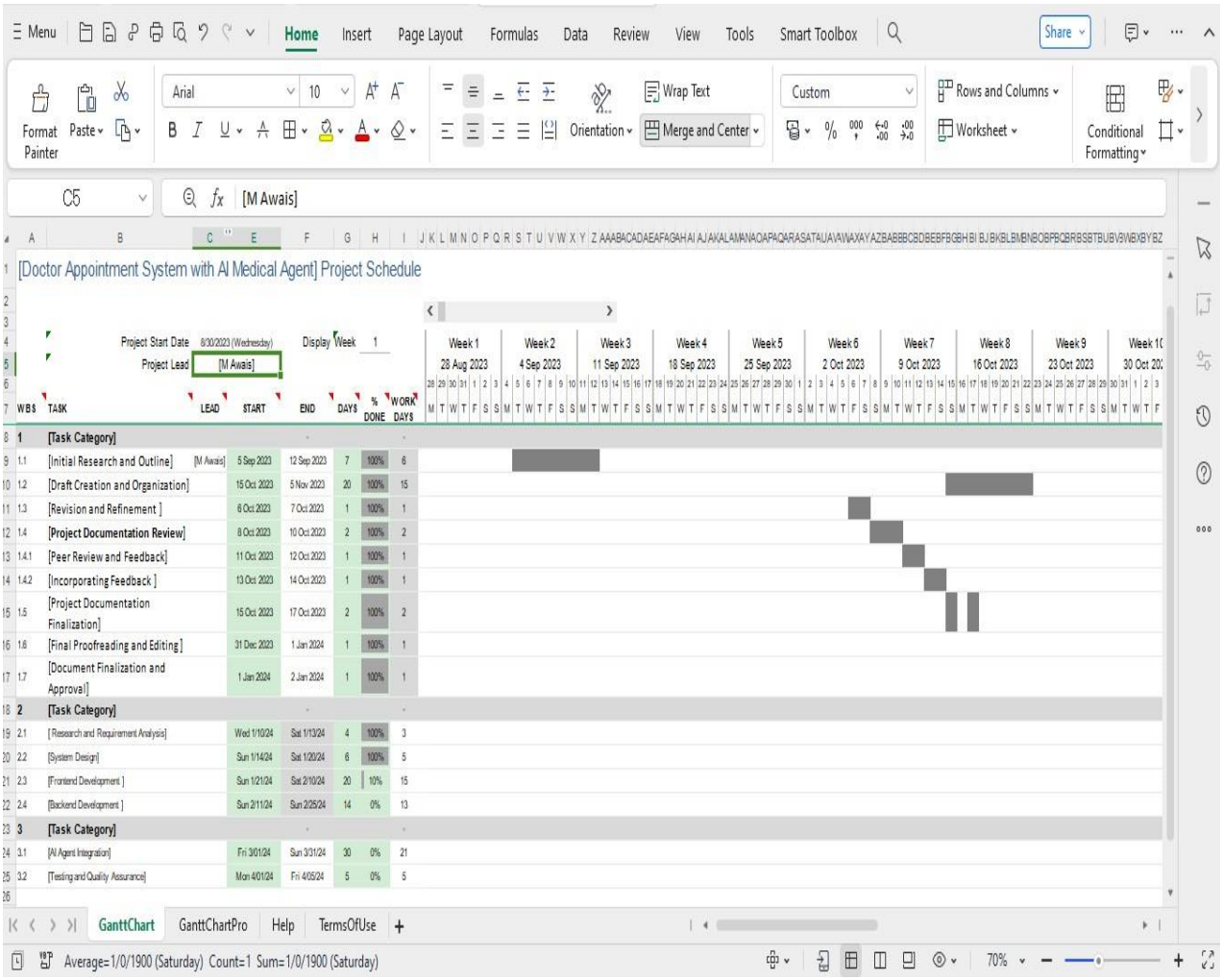


Figure 1.1 Gantt Chart

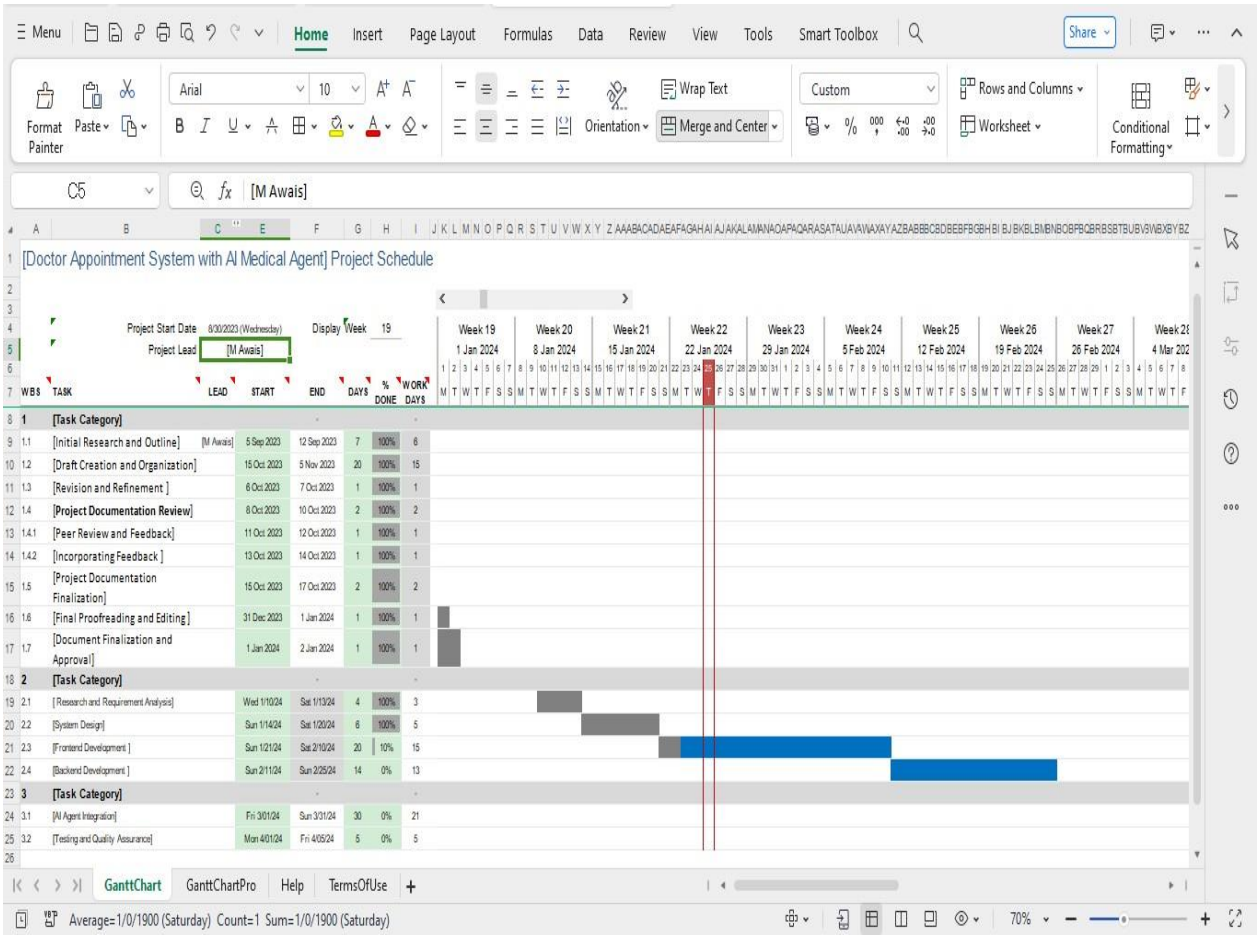


Figure 1.2 Gantt Chart

1.7. Report Outline

Introduction:

A digital appointment scheduling system for medical appointments allows patients to book appointments at specialized clinics that are specific to their health concerns; e.g., dentistry, injury, mental health, muscle strain, obesity, arthritis, asthma among other types of disease. Most human beings have diversified illnesses with a proportion of around 95 percent worldwide population. This is because countless people make weekly appointments with the closest doctors around them so that they can receive appropriate care. You can gain significantbenefits from joining the doctor appointment booking system into your clinic or doctor's site. Itinvolves just one form filled out by the patient through different fields and sent back for confirmation that an appointment has been scheduled. It simply means there is a booking form

available, which had filled up by patients and submitted so as to be considered as an appointment. This whole process saves a lot of time for the patient as well as you.

AI Medical Agent: We have integrated our platform with state-of-the-art AI medical agent platform which assists patients in getting medical information about various diseases and guiding them accordingly on what to do next regarding their illness .This AI agent mimics human-like interactions ensuring that patients feel understood and cared throughout the time they are doing online booking for appointments.

How It Works:

2 **Registration:** First, patients fill in their medical information online and create profiles

3 **AI Interaction:** The chatbot offers a conversation space where patients can ask questions abouttheir condition and other useful information.

4 **Medical Information:** Patients can also use the same chatbot to ask questions regarding medication or general health advice.

The Doctor Appointment System with an AI medical agent is basically the first chapter of the whole work. It is meant as a summary of what the project entails such as its objectives,motivation, scope and a short overview of how the system works. In this way, we introduce readers to what the system does at a high scale level what its purpose in reference to making subsequent sections easy for them.

Implementation

5 MongoDB:

A database management system to keep records for appointment, user data and systemconfigurations.

6 Express.js:

The framework in the backend side to handle server logic routing and API creation.

7 React:

This library used at the front end is principally meant for creating interactive user interfaces that facilitate smooth customer experience while scheduling an appointment or interacting with AI agent.

8 Node.js:

It is a back-end environment that helps run JavaScript code outside of a web browser therebyfacilitating server-side scripting.

9 AI Libraries:

Python (version 3.8 or higher) integrated with appropriate AI libraries such as TensorFlow,Scikit-learn, spaCy among others in order to support AI agent functionalities.

1.8. Empathy Map

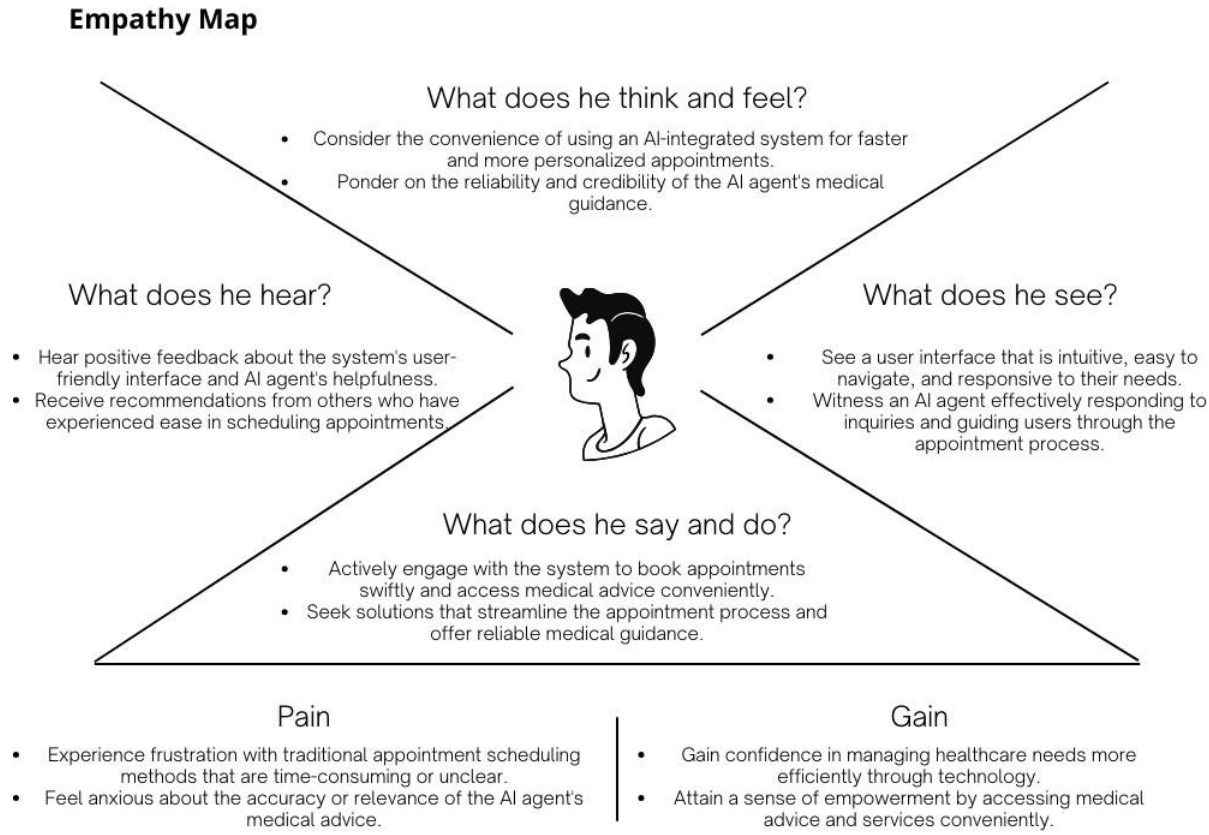


Figure 1.3 Empathy Map

Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1 Introduction:

Purpose:

The software requirements given in this great document are applicable to the Doctor Appointment System, version 2.0. This SRS lays out the scope without leaving any thing that represents the entire Doctor Appointment System, its extensive features and its interactions with other systems. It covers all sub systems and components of which a system is made to enable one understand what is required by such system from a global point of view. This SRS provides a definitive roadmap for system development by highlighting the vital software preconditions necessary for a successful implementation and improvement of the Doctor Appointment System.

2.1.1. Document Conventions

Font Usage: The SRS shall be majorly done in Calibri font, which may make some parts or titles appear bolded for better visibility.

Priority Assignment: Each requirement statement has its own priority level indicated by symbols:

High Priority: An exclamation mark (!) appears red.

Medium Priority: An amber triangle (Δ) serves as an identifier.
Low Priority: A green circle (●) symbolizes it.
Requirement Hierarchy: Higher-level requirements dominate detailed ones in terms of priorities towards maintaining consistency thus emphasizing on greater importance within wider goal framework.

2.1.2. Intended Audience and Reading Suggestions

Software Engineers: Detailed technical specifications, system architecture and implementation.Project Managers: Overview sections using project scope, milestones, and timelines.

2 **Testers:** Functional requirements, validation and verification of procedures.

3 **Documentation Writers:** System overview, feature descriptions as well as user manualreferences.

4 **Users/Clients– Refer to high-level system functionalities;** user-centric features that help usunderstand what this system can do.

5 Document Structure and Reading Sequence:

6 **Overview Sections:** Start with introduction, purpose, scope sections to get acquainted with theobjectives of this project and its coverage areas

7 **Developers:** Go deeper into the nitty gritties such as technical details,system architecture,functional requirements for implementation.

8 **Project Managers:** Look at project scope to be in line with project planning and management.

9 **Testers:** Review functional requirements as well as validation procedures and testing methodologies.

10 **Documentation Writers:** Discussing about system overview; feature descriptions; userdocumentation references while creating content

Users/Clients –High-level system functionalities including user-centric features

2.1.3. Product Scope

The software at hand is the Doctor Appointment System which was designed to relieve and promote scheduling of medical appointments for patients and healthcare providers, respectively. It will be a user-friendly interface making appointment management efficient as well as accessible and convenient in health care industry.

Key Objectives and Benefits:

2 **Enhanced User Experience:** It will help the patients to book an appointment easily or quicklythus reducing their waiting time.

3 **Efficient Healthcare Delivery:** Hospitals should be rid of operational bottlenecks that result inineefficiency of resources, without affecting efficiency of health care delivery.

4 **AI Integration:** Users are given primary advice on basic knowledge about health issues throughan integrated AI agent.

5 Aligning with Corporate Goals:

6 Several reasons why this Doctor appointment system is compatible with corporate goalsinclude enhancing patient experience in healthcare services by improving quality service.

7 **Improving Service Quality:** This shows how dedicated the organization is when it comes tousing technological tools to make better access to healthier options possible.

8 **Innovation and Technological Integration:** Showcasing the organization's commitment toleveraging technology for improved healthcare accessibility.

9 **Customer-Centric Approach:** Demonstrating a commitment to user-centric solutions byprioritizing convenience and efficiency in healthcare services.

9.1.1. References

https://staff.emu.edu.tr/alexanderchefranov/Documents/CMPE412/SRS.pdf

2.2 Overall Description

The Doctor Appointment System with AI integration is a modern healthcare solution designedto streamline and enhance the medical appointment booking process by exploiting MERN (MongoDB, Express.js, React, Node.js) stack and AI capabilities which change the interaction between users and healthcare services.

System Overview:

This system has a user-friendly web interface that can be used by both patients and healthcare providers. It allows people to schedule appointments in an easy way and get some advice concerning their health issues as well as general management of their healthcare needs.

Similarly, this platform enables doctors schedule patients' visits, check their information or histories besides offering personalized care.

Key Features:

1. **Appointment Booking:** Users can book, reschedule or cancel preferred doctor's appointments depending on their availability.
2. **AI Medical Assistance:** Incorporation of artificial intelligence technologies enables users receive individualized medical guidelines, prognosis of symptoms as well as preliminary consultations.
3. **User Authentication:** To guarantee data confidentiality and privacy mechanisms for secure user authentication and authorization are established.
4. **Real-time Communication:** To allow for follow-ups necessary answers to questions from medical practitioners they have consulted
5. **Data Analytics:** Comprehensive data analytics tools assist healthcare providers in understanding patient trends and improving services.

System Benefits:

1. **Accessibility:** Patients can reach out to it whenever they want and make appointments as well as get some basic medical advice at any time when they are free.
2. **Efficiency:** The system minimizes waiting times in hospitals because appointment scheduling is automated with AI based services improving efficiency.
3. **Personalized Care:** Healthcare AI helps patients to tailor their wellness programs according to their own specific needs.
4. **Data-Driven Insights:** Analytical tools offer important data for better decision making and improved care among health care personnel.

Target Audience:

This system will be of great use to the general public in terms of its relevance for persons looking for medical advices; doctors, clinics and other health professionals; administrators responsible for health institutions

2.2.1. Product Perspective

The Doctor Appointment System (DAS) with Artificial Intelligence integration is a standalone product that was designed to modernize how primary hospital positions are scheduled together with the provision of initial aid advices through it, therefore, it has been conceptualized as an independent entity complementing conventional appointment reservations by integrating advanced AI features.

Context and Origin:

Self-Sufficient Product: The DAS is an amazing stand-alone system that not only replaces, but also reinvents, the conventional dispositions at a medical clinic. It serves two main groups of people, i.e. patients and medical practitioners.

AI Integration: These developments started right from the inception of this AI-empowered smart health assistant. On other hand, this AI feature complements regular appointment service by offering specific healthcare recommendations as well as first-hand consultations to anyone in need.

Diagram:

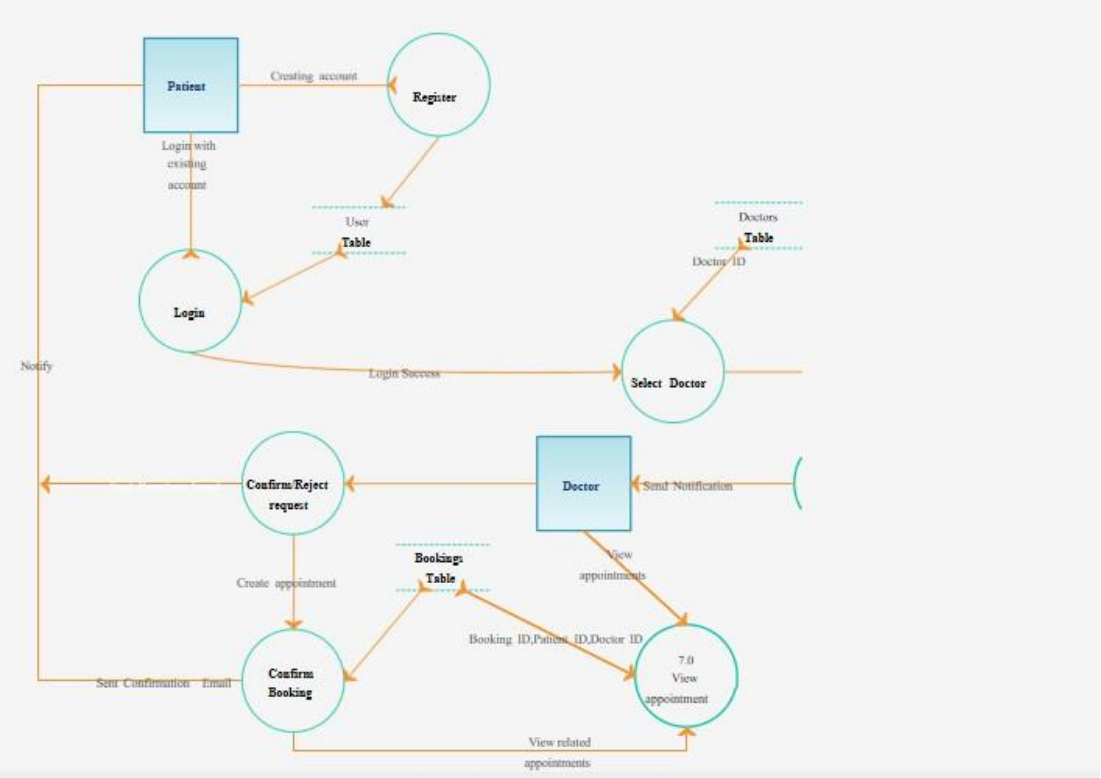


Figure 2.1 ProtoType

2.2.2. User Classes and Characteristics

Patients:

1 ***Frequency of Use:*** Regular or occasional use based on medical needs.

2 ***Product Functions:*** Appointment booking, accessing medical advice, managing personal healthrecords.

3 ***Technical Expertise*****:** Varies from basic to moderate computer literacy.

4 ***Security/Privilege Levels*****:** Access to personal health information and scheduling privileges.

5 ***Educational Level/Experience*****:** Diverse educational backgrounds and experiences in usinghealthcare services.

Healthcare Providers (Doctors, Clinics):

1 ***Frequency of Use*****:** It is always being used in interacting while making appointments and seeingpatients.

2 ***Product Functions*****:** This will allow quick appointment booking in case of an emergency; updateselectronic medical records and issues drugs.

3 ***Technical Expertise*****:** They can efficiently operate more sophisticated health software applications.

4 ***Security/Privilege Levels*****:** These may organize visits, tell patients about their previousconsultations with a doctor or treatments received.

5 ***Educational Level/Experience*****:** It has a PhD in digital health solutions for practitioners as well

System Administrators:

1 ***Frequency of Use*****:** At times people hire them to do system maintenance.

2 ***Product Functions*****:** Sets up accounts for users, software protection systems and other configurations.

3 ***Technical Expertise*****:** It has knowledge exceeding that in basic systems administration skills.

4 ***Security/Privilege Levels*****:** At installation during system setup it may have any highest administrative privileges possible

5 **Educational Level/Experience*:*** A technical background having been a former systems administrator.

Importance of User Classes:

Primary Users: Patient and Health Care Provider Users are the prime User Classes, who are expected to manage appointments and provide or obtain health services smoothly.

Secondary Users: Other users such as system administrators also play a role in the Doctor Appointment System, but not directly interacting with the core functionalities used by patients and healthcare providers.

Operating Environment

Hardware Platform:

The Doctor Appointment System operates within the following hardware environment:

1 ***Server Infrastructure*:** Utilizes Heroku cloud-based servers for hosting and deployment.

2 ***Minimum Server Configuration*:** 4GB RAM, 2 vCPUs, 50GB Storage (adjustable based on scalingrequirements).

3 ***Client Devices*:** Accessible through standard web browsers on desktops, laptops, tablets, andmobile phones.

2.2.3. Operating System and Versions:

4 ***Server OS*:** Linux-based OS (Ubuntu 20.04 LTS or later) for optimal system stability and security.

5 ***Client OS Compatibility*:** Supports major operating systems - Windows (7 or later), macOS,Linux,

Software Components and Dependencies:

1 ***Backend Framework:*** Node.js runtime environment (version 14 or higher) with Express.js forserver-side scripting.

2 ***Frontend Framework*:** React.js (latest stable version) for the user interface.

Database: MongoDB (version 4.4 or higher) as the primary database management system.

AI Integration:

JavaScript Libraries:

Used for frontend AI interactions and user-facing AI agent functionality.

Integration with JavaScript-based AI libraries such as Langchain browser-compatible machine learning tasks.

Integration-Friendly: Designed to seamlessly integrate with third-party APIs for additional functionalities (e.g., healthcare provider databases, payment gateways).

***Web Services Compatibility*:** Ensures interoperability with various web services and protocols (RESTful APIs, HTTPS, etc.) for data exchange and communication.

2.2.4. Design and Implementation ConstraintsRegulatory Compliance:

1 **Healthcare Regulations:** Adhering to HIPAA or other like legislations on data protection in healthcare and secure storage of protected health information during transmission.

2 **AI Ethical Guidelines:** Observance of moral principles regarding AI's use in healthcare to ensure justice concerning the medical suggestions provided as well as their violation of human rights.

2. Technological Limitations:

1. ***Technology Stack*:** Use MERN stack that is a must-have across all systems in order to make it uniform and interconnected
2. ***AI Libraries/Frameworks*:** Some popular AI library frameworks such as TensorFlow or PyTorch are used for building intelligent agents

3. Interoperability and Integration:

1. ***API Integration*:** Compatible with legacy systems of providers, so they can sendmessages, appointment scheduling
2. ***Database Constraints*:** Should be compatible with MongoDB or may have problems when migrating to any other databases because of system requirements.

4. Performance and Scalability:

- ***Real-time Interaction*:** To allow online user-AI agent interactions

- ***Scalability*:** ability to accommodate increasing numbers of users and more workloads for AI computations

- **5. Security Considerations:**

  - ***Data Security*:** If sensitive patient data is encrypted and safety stored, it can never get into the hands of unauthorized persons.
  - ***User Authentication*:** In logging, only those people who are authorized to do so should strictly adhere to the necessary procedures required for logging in.

- **6. Usability and Accessibility:**

- ***User Interface Guidelines*:** A user friendly design that is usable by different groups end-users with varying skill levels need to be put in place.
- ***Accessibility Standards*:** The platform is WCAG compliant hence can be accessed by physically challenged persons.

- **7. Maintenance and Support:**

- ***Ongoing Maintenance*:** This includes after delivery customer managed operationssupport as well as system maintenance handbook which has everything.
- ***Scalable Infrastructure*:** It means that other enhancements, variations or even modifications can be made without affecting service availability.

2.2.5. Assumptions and Dependencies

Assumptions:

1 **Third-Party AI Libraries:** Assuming availability and compatibility of AI libraries (e.g., TensorFlow, Scikit-learn) for AI agent implementation, subject to their consistent maintenance and updates.

2 **Regulatory Compliance:** Assuming continued compliance with healthcare regulations (such as HIPAA) and ethical guidelines for AI usage without significant regulatory changes impacting system operations.

3 **Stable Technology Stack:** Assuming stability and reliability of the MERN stack components (MongoDB, Express.js, React, Node.js) and their support for future upgrades without major disruptions.

4 **API Integration:** Assuming seamless integration capabilities with external healthcare provider systems for data exchange and appointment scheduling, contingent on consistent API availability.

5 **User Adoption:** Assuming a certain level of user acceptance and adoption of AI- driven medical assistance, considering it's a relatively new feature in healthcare services.

Dependencies:

- **Third-Party Services:** Dependency on external AI service providers or libraries for maintaining and updating AI models used for medical advice and symptom analysis.

- **API Compatibility:** Dependency on the stability and compatibility of APIs providedby healthcare institutions for seamless data exchange and appointment scheduling.

- **Hardware and Hosting Services:** Dependency on reliable hosting services (such as Heroku) and hardware stability to ensure system availability and performance.

- **Development Environment:** Dependency on development tools, frameworks, and libraries' consistent support for the MERN stack.

- **Regulatory Changes:** Dependency on healthcare regulatory changes, as any alterations could impact data handling, security protocols, or AI usage guidelines.

2.2.6. External Interface Requirements

User Interfaces

1. ***Patient Interface:***
    1. ***Dashboard*:** Displaying upcoming appointments, AI-assisted medical advice, and quick access to medical history.
    2. ***Appointment Booking*:** Intuitive interface for scheduling appointments, specifying doctors, and selecting preferred timings.
    3. ***Profile Management*:** Editing personal information, managing health records, and accessing past appointments.
2. ***Healthcare Provider Interface:***
    1. ***Appointment Schedule*:** Clear display of daily/weekly appointments, patient details, and medical histories.
    2. ***Patient Records*:** Access to patient profiles, medical histories, and the AI-generated initial medical advice.
    3. ***Calendar View*:** Providing an overview of daily schedules, upcoming appointments, and availability.
3. ***Administrator*** Interface:
    1. ***User Management*:** Adding/removing users, assigning roles, and managing system configurations.
    2. ***System Dashboard*:** Monitoring system performance, handling system maintenance, and access to logs and reports.

UI Characteristics:

1. ***Consistent Design Language*:** Uniform color schemes, font types, and layout for a cohesive user experience.
2. ***Responsive Design*:** Ensuring usability across various devices (desktops, tablets, mobiles).
3. ***AI Integration Interface*:** Seamless integration of AI-driven advice in aconversational and user-friendly manner.

2.2.7. Hardware Interfaces

Supported Device Types:

The Doctor Appointment System interacts with various devices, including:

Desktops: Windows, macOS, Linux-based systems.

Laptops: Compatible with various brands and operating systems.

Data and Control Interactions:

Input/Output Devices: The system interacts with standard input devices (keyboard, mouse, touchscreens) and output devices (displays, printers).

***Communication Protocols*:** Utilizes HTTP/HTTPS protocols for data exchange between the software and hardware components.

Nature of Interactions:

***Data Transmission*:** Real-time data transmission between the software and hardware for appointment scheduling, user queries, and system responses.

User Interface Interactions: Intuitive and responsive user interfaces optimized for various devices to ensure seamless interactions.

Control Mechanisms:

***Device Access Permissions*:** The system adheres to device access permissions for security anddata privacy.

***Event Handling*:** Manages hardware events (e.g., touch events, clicks) for user interactions and system responses.

2.2.8. Software Interfaces

Connected Software Components:

1 ***Database Management System:***

2 ***Database*:** MongoDB (version 4.4 or higher).

3 ***Purpose*:** Store and retrieve user information, appointment schedules, and system configurations.

4 ***Data Sharing*:** User profiles, appointment records, and configuration settings are shared withthe database.

Operating Systems:

1 ***Supported OS*:** Windows, macOS, Linux-based systems.

2 ***Purpose:*** Ensure cross-platform compatibility and user accessibility.

3 ***Data Sharing*:** Handles system calls, file operations, and resource management.

Frontend Framework:

1 ***Framework***: React.js (latest stable version).

2 ***Purpose*:** Facilitate a user-friendly interface for appointment scheduling and interaction.

3 ***Data Sharing*:** Manages user interactions, retrieves and displays information from the backend.

Backend Framework:

1 ***Framework*:** Node.js with Express.js.

2 ***Purpose****:* Handle server-side operations, manage data retrieval and processing.

3 ***Data Sharing:*** Communicates with the database, manages business logic, and serves clientrequests.

Data Exchange and Communication Nature:

1 ***API Communication:*** Utilizes RESTful APIs for communication between frontend and backendcomponents.

2 ***Data Items:*** Incoming data includes user authentication details, appointment requests, and user queries. Outgoing data involves appointment confirmations, user information, and systemresponses.

Implementation Constraints:

1 ***Data Security:*** Enforces encryption and secure communication protocols (HTTPS) for datatransmission to ensure user privacy.

2 ***API Protocols:*** Adheres to documented API protocols for seamless integration and standardizedcommunication between software components.

2.2.9. Communications Interfaces

Communication Functions:

Web Browser Interaction:

1 ***Protocol:*** Utilizes HTTP/HTTPS for web-based interactions and data transmission.

2 ***Message Formatting:*** JSON (JavaScript Object Notation) for structured data exchange betweenclient-server interactions.

3 ***Data Transfer Rates:*** Optimized for efficient data transfer to accommodate various usernetwork speeds.

Network Server Communication:

Protocol: HTTP/HTTPS requests and responses between client and server components.

Security and Encryption**:** Implements TLS/SSL encryption protocols for secure data transmission over the network.

Data Synchronization**:** Utilizes asynchronous data transfer mechanisms for real-time updatesand synchronization.

Message Formatting*:* HTML or plain text formatting for email notifications and communication.
Security: Ensures encryption of sensitive email content and secure authentication mechanisms.

Security Measures:

Encryption**:** Adopts TLS (Transport Layer Security) or SSL (Secure Sockets Layer) encryption for secure data transmission.

Authentication**:** Implements robust user authentication mechanisms to validate user access and prevent unauthorized access.

2.2.10. System Features

2.2.11. System Feature 1 Feature 1:

Appointment SchedulingDescription:

Purpose: Enables users to schedule appointments with healthcare providers efficiently.

Functionality:

2 Users can select preferred dates, times, and specific healthcare providers.

3 The system displays available slots based on the provider's schedule.

4 Confirmation notifications are sent upon successful appointment booking.

Integration:

- Communicates with the backend system to update appointment slots in real-time.
- Utilizes the user interface for seamless interaction and booking confirmation.

Description and Priority Feature:

Appointment SchedulingDescription:

Functionality: Facilitates user-friendly appointment scheduling with healthcare providers.

User Interaction: Allows selection of preferred dates, times, and specific providers.

Confirmation: Sends confirmation notifications upon successful appointment booking.

Priority:

***Overall Priority*:** Medium

Component Ratings:

- Benefit: 7/9 - Enhances user experience and accessibility for scheduling appointments.
- Penalty: 3/9 - Minor penalties for potential system latency during peak usage times.
- Cost: 5/9 - Moderate development and maintenance costs due to real-time updates and notifications.
- Risk: 4/9 - Moderate risk associated with data accuracy and timely notificationdelivery.

Stimulus/Response Sequences Stimulus/Response

Sequences: Appointment SchedulingUser Actions:

Selection of Preferred Date and Time:

- ***Stimulus***: User selects a preferred date and time slot for the appointment.
- ***Response*:** System updates and displays available healthcare providers for the selected slot.

Selection of Healthcare Provider:

- ***Stimulus*:** User selects a specific healthcare provider.
- ***Response*:** System confirms the availability of the chosen provider for the selected slot.

Confirmation of Appointment Booking:

- ***Stimulus*:** User confirms the appointment booking.
- ***Response*:** System generates a confirmation notification and updates the appointment schedule.

Availability Display:

- ***Stimulus:*** User selects a date and time slot.
- ***Response*:** System presents a list of available healthcare providers for the selected slot.

Provider Availability Confirmation:

- ***Stimulus:*** User selects a healthcare provider.
- ***Response:*** System confirms the availability of the chosen provider or suggests alternate available times if necessary.

Confirmation Notification:

- ***Stimulus*:** User confirms the appointment booking.
- ***Response:*** System generates a confirmation message and updates the appointment schedule accordingly.

2.2.12. Functional Requirement

REQ-APT-SCH-1: Date and Time Selection

- ***Description:*** Users must be able to select preferred dates and time slots for appointments.
- ***Response to Invalid Input:*** Display an error message for non-select able dates or times.

REQ-APT-SCH-2: Healthcare Provider Selection

- ***Description:*** Users should have the option to choose specific healthcare providers.
- ***Response to Invalid Input:*** If provider unavailability, suggest alternate times or providers.

REQ-APT-SCH-3: Appointment Confirmation

- ***Description:*** The system should confirm the appointment booking upon user confirmation.
- ***Response to Invalid Input:*** Alert users if booking is not confirmed due to system error.

REQ-APT-SCH-4: Real-time Availability Update

- ***Description:*** Ensure real-time updating of provider availability based on selected slots.
- ***Response to Invalid Input:*** Display alternative available slots if chosen slot isunavailable.

REQ-APT-SCH-5: Notification Delivery

- ***Description:*** Send confirmation notifications via email or SMS upon successful booking.
- ***Response to Error Conditions:*** Retry notification delivery in case of a transmissionfailure.

4.4.2. System Feature 2

System Feature 2: User Profile Management

Description and Priority:

Description: Enables users to create, view, and manage their profiles within the system.

Priority:

Overall Priority: *High*

Component Ratings:

- Benefit: 8/9 - Enhances user experience and personalization.
- Penalty: 2/9 - Minor delays if profile management requires additional serverprocessing.
- Cost: 6/9 - Moderate development and maintenance costs due to database interactions.
- Risk: 5/9 - Moderate risk related to data security and user privacy.

Stimulus/Response Sequences:

User Actions:

Profile Creation:

- ***Stimulus:*** User initiates profile creation by providing necessary information.
- ***Response:*** System validates input and generates a user profile upon successfulsubmission.

Profile Viewing/Editing:

- ***Stimulus*:** User accesses the profile section to view or edit personal details
- ***Response*:** System displays the user's profile data or allows modifications.

System Responses:

Profile Creation Confirmation:

- ***Stimulus:*** User submits profile information.
- ***Response:*** System confirms profile creation and provides an acknowledgmentmessage.

Profile Modification Confirmation:

- ***Stimulus*:** User saves edited profile details.
- ***Response*:** System updates the profile information and confirms successfulmodification.

Functional Requirements:

REQ-USER-PROFILE-1: Profile Creation

- ***Description:*** Users should be able to create profiles by entering required personal details.
- ***Response to Invalid Input:*** Display error messages for missing or invalidinformation.

REQ-USER-PROFILE-2: Profile Viewing

- ***Description:*** Users must access their profile information for viewing purposes.
- ***Response to Invalid Input:*** Display user profile data without allowingmodifications.

REQ-USER-PROFILE-3: Profile Editing

- ***Description:*** Users should edit existing profile information as needed.
- **Response to Invalid Input:** Validate and update user information upon successfulchanges.

REQ-USER-PROFILE-4: Profile Security

- ***Description:*** Implement security measures to protect user profile data from unauthorized access.

- ***Response to Error Conditions*****:** Restrict access and provide error messages forunauthorized attempts.

4.4.3. System Feature 3 (and so on)

User Authentication and Authorization

Description: Allow users (patients, doctors, administrators) to register, log in securely, and manage access rights.
Priority: High

2. Appointment Management

Description: Enable users to schedule, reschedule, or cancel appointments with doctors.

Priority: High

3. AI-Assisted Medical Advice

Description: Provide initial medical advice using AI algorithms based on symptom inputs.

Priority: High

4. Patient Profile Management

Description: Allow patients to manage their profiles, including medical history updates and record access.
Priority: Medium

5. Doctor Dashboard

Description: Present doctors with an overview of their appointments, patient details, and medical histories.
Priority: Medium

6. Administrator Tools

Description: Admin dashboard for managing users, system configurations, and accessing logs/reports.
Priority: Medium

7. Appointment Notifications

Description: Automated reminders or notifications for upcoming appointments to both patients and doctors.

8. System Monitoring and Reporting

Description: Tools to monitor system performance and generate reports on appointmenttrends or system usage.

Priority: Low

4.1. Nonfunctional Requirements

4.5.1. Performance Requirements

Performance Expectations:

- **Response Time:** System should respond to user actions within 2 seconds for appointment scheduling and profile management.
- **Concurrent Users:** Support a minimum of 100 simultaneous users without significant performance degradation.
- **Database Query Time:** Database queries should execute within 300 milliseconds to maintain responsiveness.
- **System Uptime:** Aim for 99.9% uptime for user accessibility.

4.5.2. Safety Requirements

Safety Measures:

- **Data Encryption:** All user data should be encrypted using industry-standard encryption protocols to prevent unauthorized access.
- **Access Control:** Implement role-based access control to ensure users access only authorized functionalities.
- **Regular Backups:** Schedule regular backups to prevent data loss in case of systemfailure.

4.5.3. Security Requirements

Security Measures:

- **User Authentication:** Implement secure user authentication methods (e.g., multi-factor authentication) to verify user identities.
- **Secure API Communication:** Use HTTPS and token-based authentication for secure communication between system components.
- **Regular Security Audits:** Conduct periodic security audits to identify and address vulnerabilities.

4.5.4. Usability Requirements

Usability Expectations:

- **Intuitive User Interface:** The system interface should be user-friendly and intuitive, requiring minimal guidance for navigation and actions.
- **Accessibility Features:** Support accessibility standards (e.g., WCAG) to ensure usabilityfor users with disabilities.
- **Clear Feedback:** Provide clear and concise feedback for user actions, ensuring users understand system responses.

4.5.5. Reliability Requirements

Reliability Expectations:

- **Error Handling:** Proper error handling mechanisms should be in place to prevent system crashes and provide informative error messages.
- **System Redundancy:** Implement fail-over mechanisms to ensure system continuity in case of server failures.
- **Regular Maintenance:** Schedule routine maintenance to address system vulnerabilities and ensure reliability.

4.5.6. Maintainability/Supportability Requirements

Maintainability/Sustainment Expectations:

- **Modularity:** Maintain code modularization to facilitate easier updates and enhancements.
- **Documentation:** Provide comprehensive system documentation for easy troubleshooting and future development.
- **Support Channels:** Establish support channels for user queries and technical assistance.

4.5.7. Portability Requirements

Portability Expectations:

- **Cross-Browser Compatibility:** Ensure system functionality across major web browsers (Chrome, Firefox, Safari, Edge).
- **Mobile Responsiveness:** Optimize system layout and functionality for mobile devices to ensure usability on various screen sizes.

4.5.8. Efficiency Requirements

Efficiency Expectations:

- **Resource Optimization:** Optimize system resources (CPU, memory) to ensure efficient performance during peak usage.
- **Bandwidth Management:** Implement strategies to minimize bandwidth usage, enhancing system performance.

4.2. Domain Requirements

Database Requirements:

- **Data Schema Design:** Define a structured database schema for efficient data storage and retrieval.
- **Backup and Recovery:** Implement regular backups and a recovery plan to preventdata loss.
- Internationalization Requirements:
- **Multilingual Support:** Enable system localization to support multiple languages fordiverse user accessibility.

Date and Time Formats: Accommodate various date and time formats based on user preferences.

Legal Requirements:

- **Compliance:** Ensure compliance with healthcare data privacy laws (e.g., HIPAA) toprotect user medical information.
- **User Consent:** Implement features for obtaining user consent for data usage and storage.
- Reuse Objectives:
- **Modular Code-base:** Design code modules for potential reuse in future system expansions or similar projects.
- **Documentation Standardization:** Maintain standardized documentation for easier future system updates or handovers.

Chapter 3

Use Case Analysis

Chapter 3: Use Case Analysis

Use case analysis also entails recognizing describing and distinguishing the different use cases that a system should support through interaction of users (actors) and the system. This is a use case analysis for a doctor appointment system with an AI medical agent.

Actors:

1.Patient:

- Start by requesting for an appointment
- Gets information on confirmed appointments, changes, and diagnosis.

2.Doctor:

- Get appointment requests
- Confirm or reject appointments
- Has information on diagnosis results.

3.AI Medical Agent:

- Helps in initiation and completion of diagnosis

Use Cases:

1.**Request Appointment:**

- Primary Actor: Patient

- Description: patient initiates request for doctors appointment.

a) Flow:

- Patient Provides Details (Preferred Date, Time etc)

- System Validates & Records The Request.

2. Confirm Appointment:

- ***Primary Actor***: Doctor

- ***Description*:** The doctor confirms or rejects the appointment.

- **Flow:**

  - Doctor receives the appointment request.

  - Doctor decides to confirm or reject.

  - System notifies the patient of the decision.

3. Initiate Diagnosis:

- ***Primary Actor*:** AI Medical Agent

- ***Description*:** The AI medical agent initiates the diagnosis process.

- **Flow:**

  - AI Medical Agent receives information from the system.

  - AI Medical Agent performs initial diagnosis tasks.

4. Complete Diagnosis:

- ***Primary Actor*:** AI Medical Agent

-***Description*:** The AI medical agent completes the diagnosis process.

- **Flow:**

  - AI Medical Agent completes the required diagnostic tasks.

  - AI Medical Agent updates the system with diagnosis results.

5. Inform Changes:

- ***Primary Actor*:** Doctor

- ***Description*:** The doctor informs the patient about changes in the appointment schedule.

- **Flow:**

  - Doctor initiates changes in the system.

  - System notifies the patient about the changes.

6. Get Appointment History:

- ***Primary Actor*:** Patient

- ***Description*:** The patient retrieves the appointment history.

- **Flow:**

  - Patient requests the appointment history.

  - System provides the patient with the relevant information.

Relationships Between Use Cases:

- Request Appointment and Confirm Appointment:

- These use cases are related in the sense that they form a sequence. The patient initiates the request, and the doctor follows up with confirmation or rejection.

- Initiate Diagnosis and Complete Diagnosis:

- These use cases represent the two phases of the diagnostic process performed by the AI Medical Agent.

- Confirm Appointment and Inform Changes:

- These use cases are related as changes may occur after confirmation. The doctor informs the patient about any changes to the appointment schedule.

- Request Appointment and Get Appointment History:

- The patient, after making an appointment, may want to retrieve their appointment history.

This use case analysis provides a foundation for understanding the interactions andfunctionalities of a doctor appointment system with an AI medical agent. The relationships between use cases help to define the flow of activities and dependencies in the system.

3.1. Use Case Model

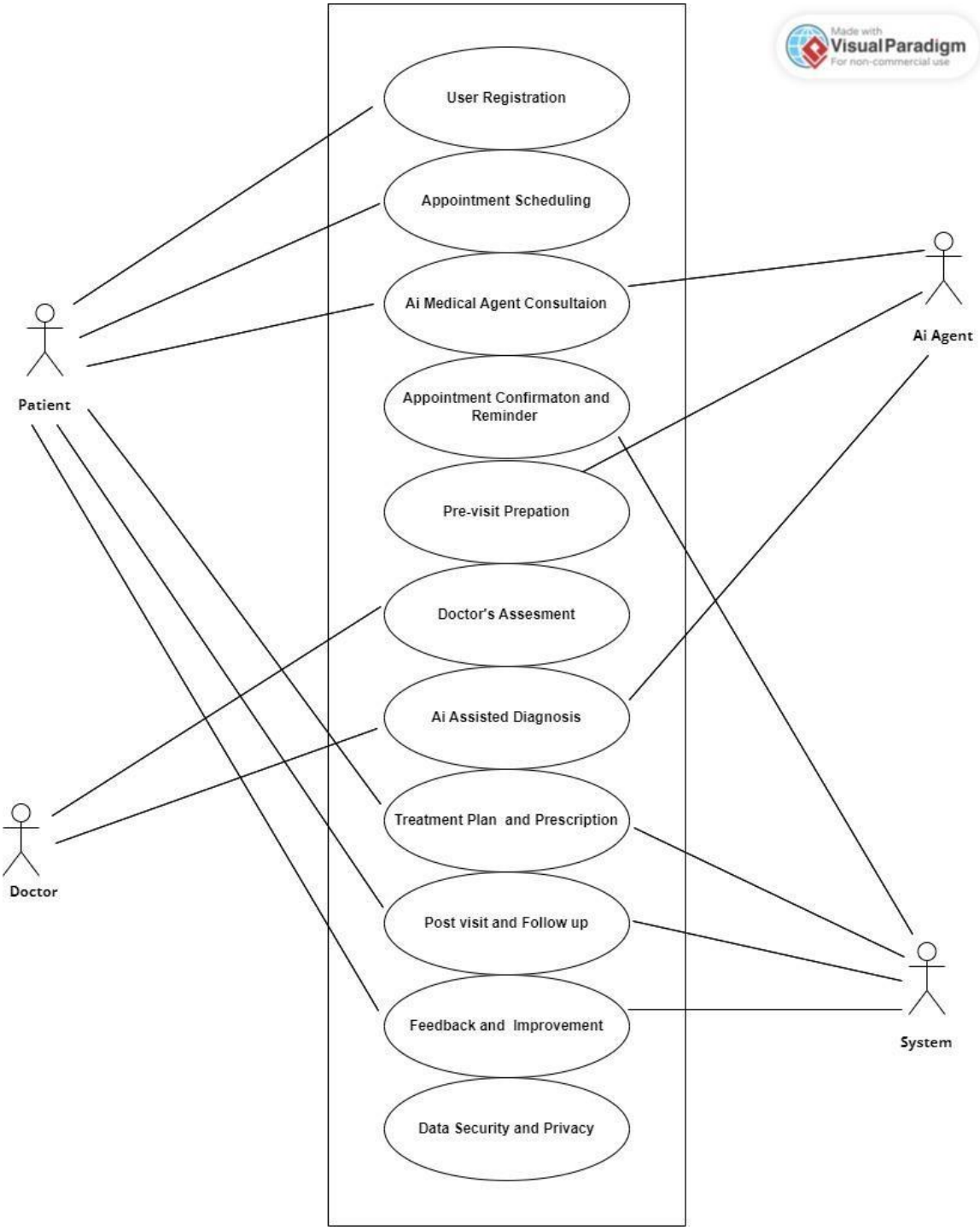


Figure 3.1 Use Case Diagram

3.2. Use Cases Description

Efficiency and accessibility of health care service can be enhanced by a doctor appointment system facilitated with an ai medical agent. here is a use case analysis of one such system. **Use Case:** Doctor Appointment System with AI Medical Agent

User Registration:

- *Actor*: Patient

- *Description:* Patients are able to register themselves on the platform where they providetheir personal information, medical history and contact details.

Appointment scheduling:

- *Actor: Patient*

- *Description:* Patients can book appointments with doctors based on their own timetable. Urgency, past appointments as well as other patient preferences are put into consideration by the system.

AI Medical Agent Consultation:

- ***Actor***: Patient, AI Medical Agent

- *Description:* The AI medical agent contacts the patient before their scheduled visit in order to find out basic information about symptoms prior, medical background and present health condition; thus it helps collect relevant data for doctor's evaluation.

Appointment Confirmation and Reminders:

- *Actor:* System

-*Description:* The patients' confirmation notifications are sent when the appointment is scheduled via email. It also sends timely reminders through email, SMS or app notifications.**Pre-Visit Preparation:**

- **Actor:** AI Medical Agent

6. Doctor's evaluation:

- Actor*:* Physician

- Description*:* During the visit AI medical agent produces a comprehensive report that includes patient's history, complains, and initial analysis; these are handed over to the doctor, to help him have an informed diagnosis.

7. AI-guided diagnosis

- ***Actor*:** Doctor, AI Medical Agent

- ***Description:*** In the process of diagnosing a patient the doctors are helped by ai medical agents who go through patients data looking at different information sources and providingprobable diagnoses as well as treatment methods.

8. Treatment plan and prescription.

- ***Actor*:**Doctor

- ***Description*:**A treatment plan is prepared based just on ai supported diagnosis and physician's examination. A physician might give prescriptions or recommend medication in form of lifestyle changes or may be advise for more tests.

9. Post Visit Follow up

-Actor: System,Patient

-***Description*:**The system sends some post-visit instructions medication prompts and follow-up appointment ideas. The process does not stop even after discharge from hospital since AI medical agent acts as companion for a while making sure that you recover well by providing helpful health tips.

10 Feedback And Improvement:

-***Actor*:** Patient,System

-***Description*:**Patinets give feedback regarding their consultations thereby helping the system and the AI medical agent get better each day.

11. ***Data Security and Privacy*****:**

- ***Actor*****:** System

- ***Description*****:** The system ensures the security and privacy of patient data. It complies with healthcare regulations and standards to safeguard sensitive medical information.

This doctor appointment system with an AI medical agent streamlines the healthcare process, improves patient engagement, and assists healthcare professionals in making more informed decisions. It combines the convenience of online appointment scheduling with the power of AI for more personalized and efficient healthcare delivery.

Chapter 4
System Design

Chapter 4: System Design

In brief this chapter explains the architectural framework and structural blueprints for the Doctor Appointment System that has been integrated with ai through the mern stack. This section offers a detailed look at the systems design components using several illustrative diagrams to convey its nuance:

4.1. Architecture Diagram

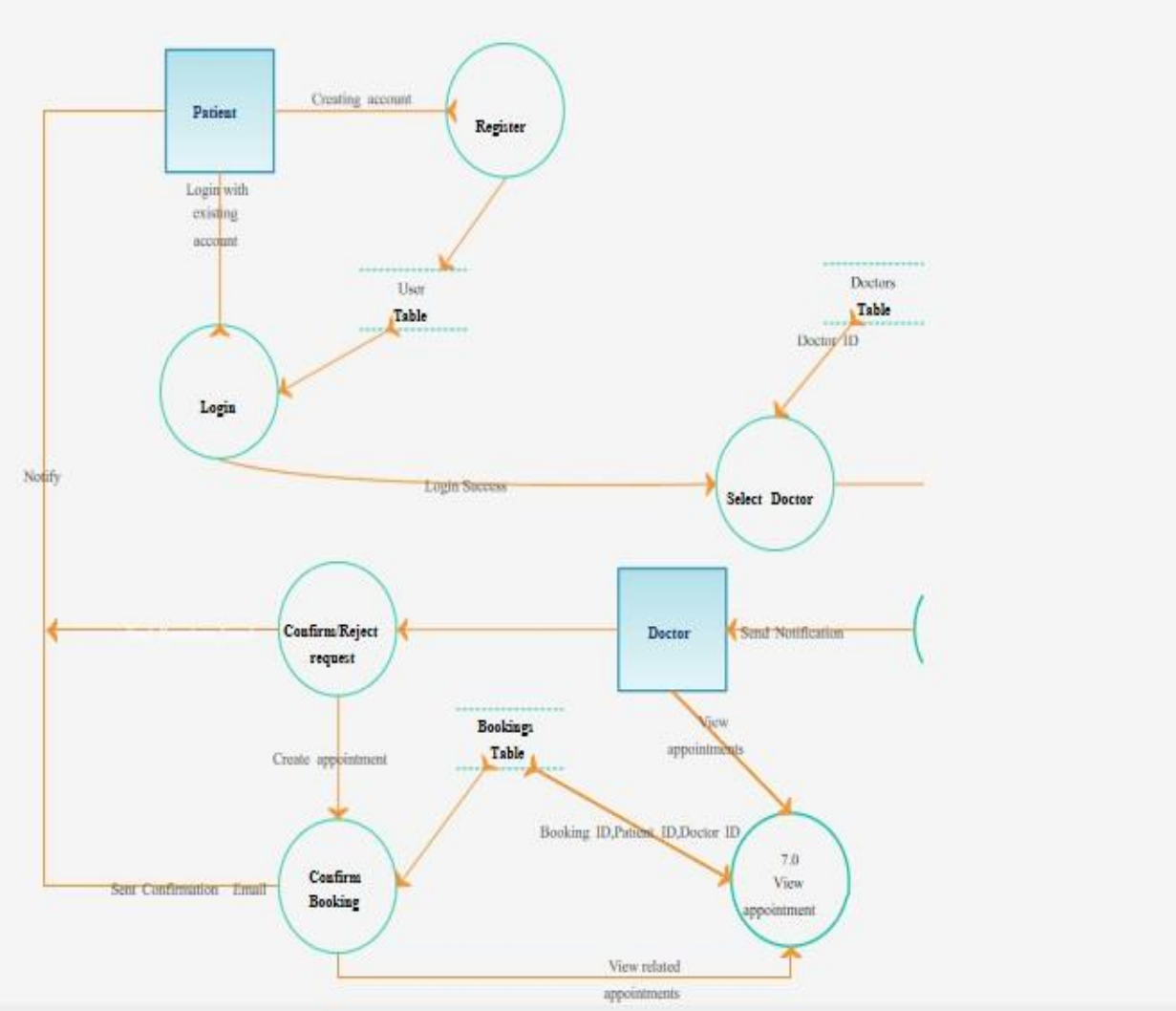


Figure 4.1 Architecture Diagram

4.2. Domain Model

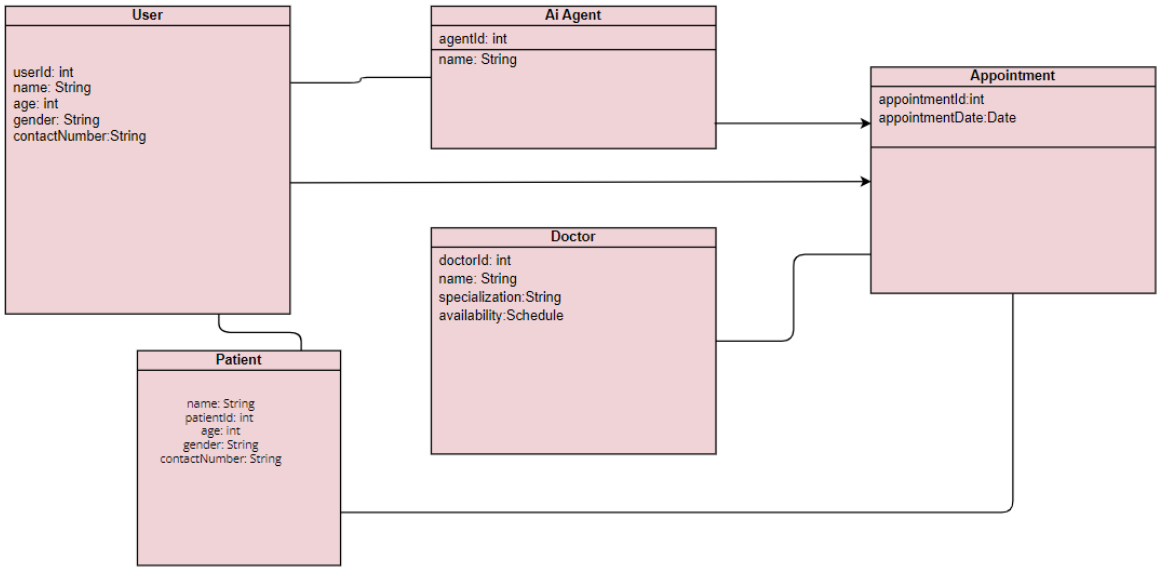


Figure 4.2 Domain Model

4.3. Entity Relationship Diagram with data dictionary

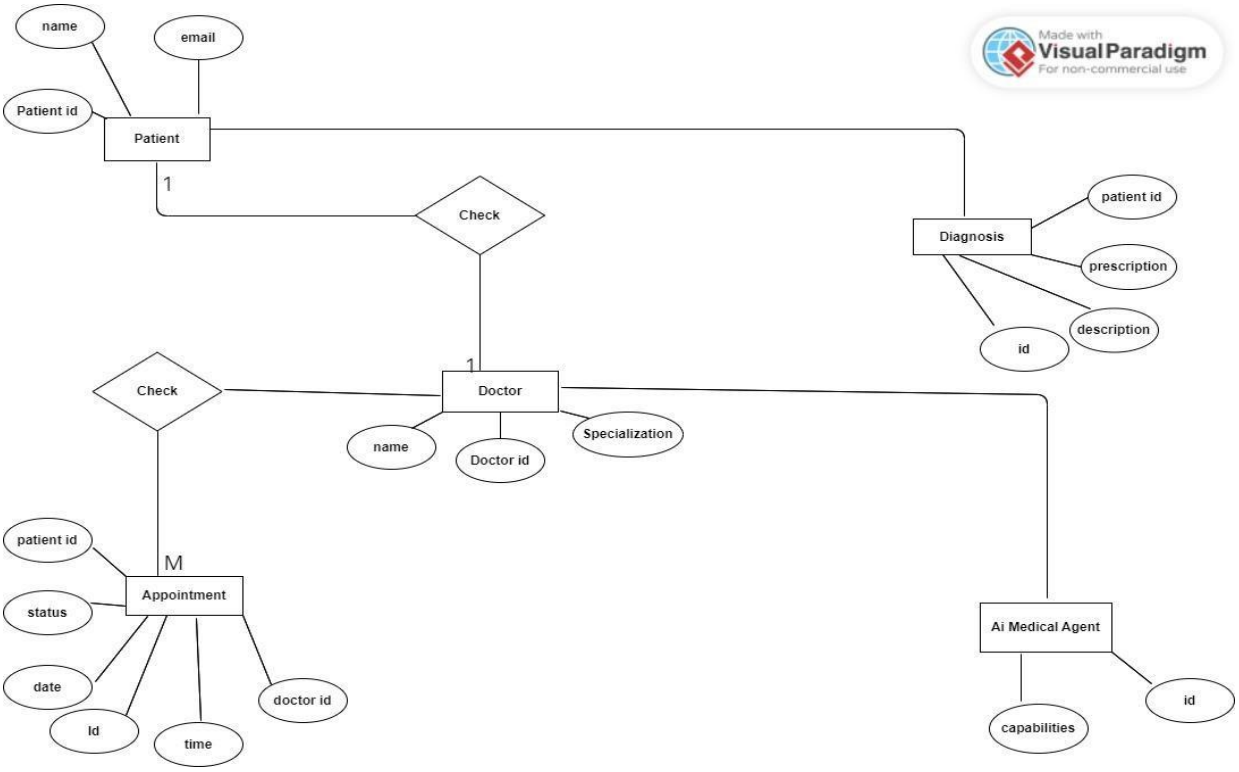


Figure 4.3 Entity Relationship Diagram with data dictionary

Key Points:

1. The entity, Patient Doctor and ai Medical Agent can be represented by us.

2. Every symbolizes their distinctiveness by the attributes of the entities (e.g patientID,doctorID, agentID).

3. Appointments between patients and doctors are represented in this Appointment entity

- It has a foreign key relationship with both Patient and Doctor entities.

4. The medical tests findings found out by the ai Medical Agent is what Diagnosis entity represents.

- In addition to that it has foreign key relationships with both Patient and Doctor tables.

4.4. Class Diagram

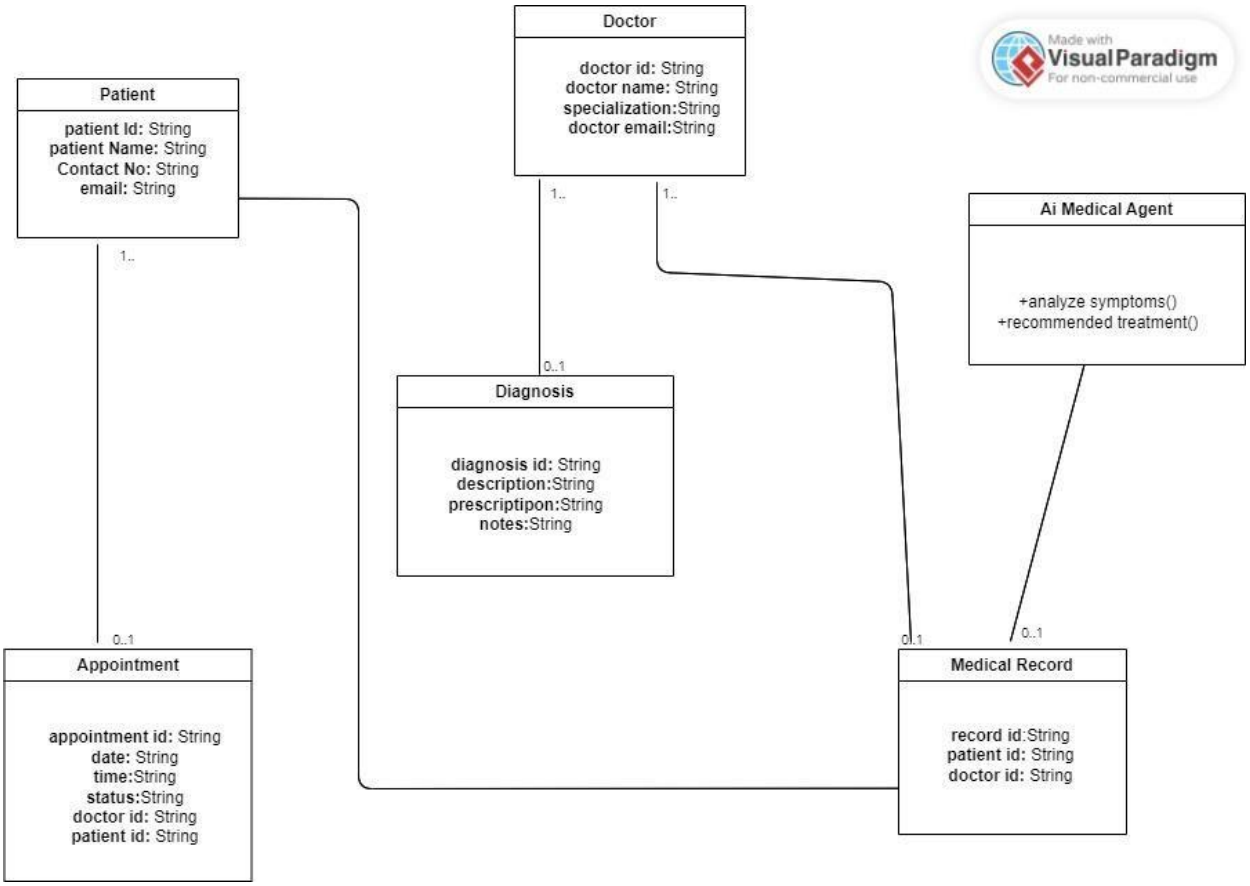


Figure 4.4 Class Diagram

Key Classes and Associations:

1. A class for patients:

***Elements*:** patientNo, handle, phone number

***Features*:** setAppointment(), getUptodateInfo(), viewPastAppointments() etc.

2. A class for doctors:

Properties: docID, name, specializatio

3. This sketch does not depict any specific attributes or operations.

4. A class for appointments:

***Details*:** appointmentNo; date; time; status; doctorNo; patientNoIt
connects the Doctor to the Patient.

5. A group of diagnoses:

– Traits: diagnosisNo diagnosis description prescription remarks

– It shows results of tests that are conducted.

6. A group of medical records;

***Description*:** recordID patientID doctorID and history of diagnosis

By showing a continuous record of diagnostic events as part of a persons health records this phrase can be used in describing what is seen in this figure.
According to this diagram depicting simplified representation with an ai medical agent as part of the static part of a doctor appointment system its simple though but real life there may be more details and relationships required depending on what systems requires about conditions that exist.

4.5. Sequence / Collaboration Diagram

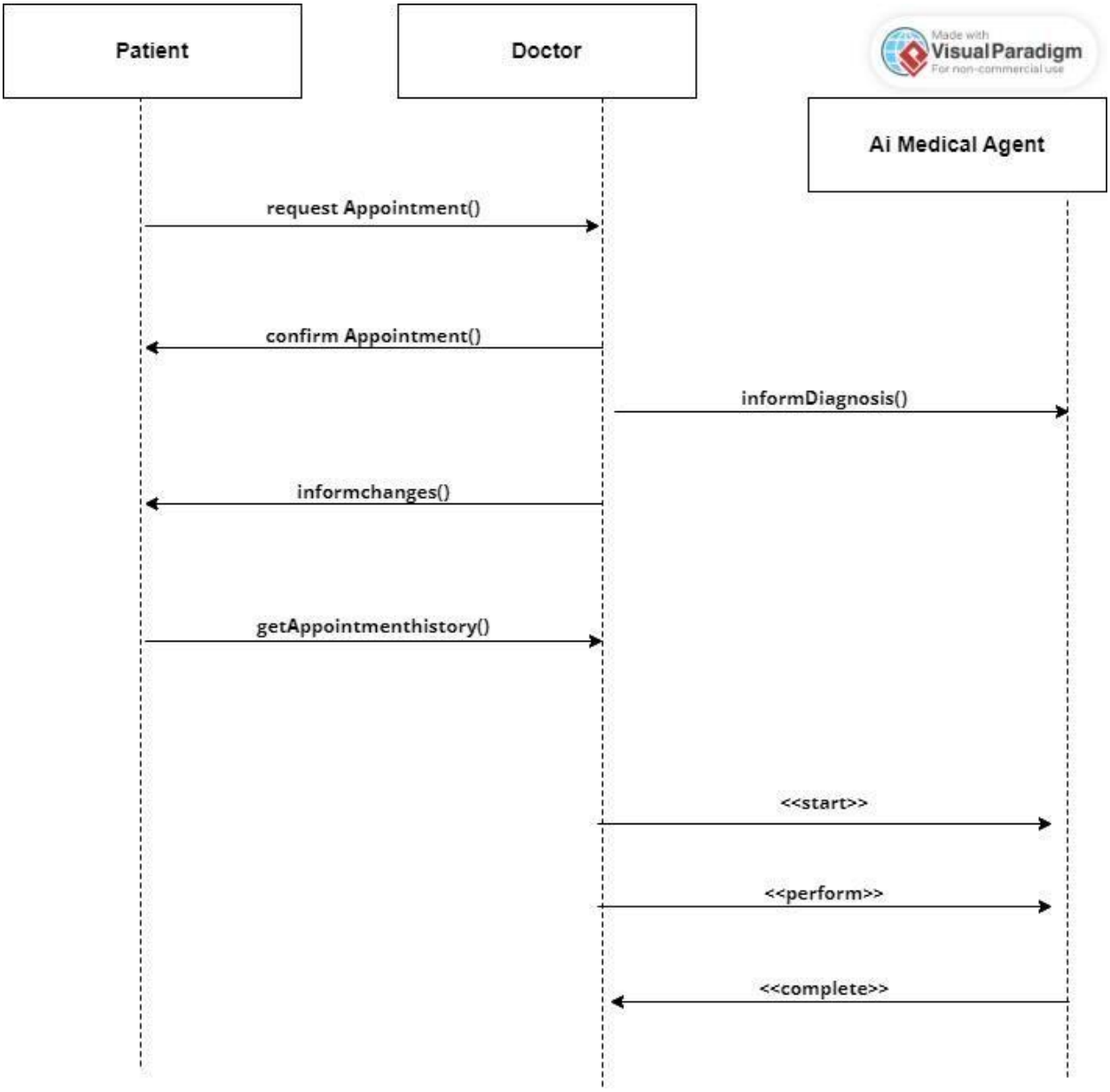


Figure 4.5 Sequence / Collaboration Diagram

Key and Transitions:

The initial state of the patient is idle**.**

TRANSITION: This requestAppointment() operation will be triggered whenever a patient requests for an appointment.

2. Doctor :

- Transition: Upon receipt of a request from the patient, the doctor changes his/her confirmation state to appointing with the use of confirmAppointment().

3. AI Medical Agent:

- Transition: First, this starts off as soon as AI Medical Agent enters its process mode where it can either perform <<perform>> or complete <<complete>> its activities.

4. Transitions Between :

- Requesting an appointment change into being told about what has changed or obtaining appointment history by patients.
- Confirming appointments turn into diagnosing by doctors.
- Starting their process represents start doing any tasks means perform andfinishing all activities stand for completein ai Medical Agent transitions

This diagram provides a broad overview of how Patients, Doctors and AI Healthcare Agents interact during doctors appointments. keep in mind that this representation is simplified and in real world cases there may be more states and transitions depending on complexity and requirements in place.

4.6. Operation contracts

1. User Authentication

Preconditions:

- ***User Credentials:*** The authentication process is where the users give their correct login credentials (username/password).

Postconditions:

- ***Access Granted*****:** Once they have been authenticated, successful users are immediately directed to their dashboards
- ***Access Denied*****:** People who fail in authentication cannot enter and must redo the sign-in process
- .

2. Appointment Scheduling

Preconditions:

- ***User Authentication*****:** To reserve appointments, patients need verification.
- ***Doctor Availability*****:** The doctor selected should have time slots on him/her.

Postconditions:

- ***Appointment Confirmation*****:** This will occur when scheduling completes successfullywith an appointment scheduled for a preferred date and time.
- ***Conflict Resolution*****:** Whenever a selected slot is unavailable, this system prompts usersto choose another period of time.

3. AI-Assisted Medical Advice

Preconditions:

- ***Patient Input*****:** In other words, patients provide true symptom details to AI algorithm.

Postconditions:

- ***AI Output*****:** The symptoms provided help the model generate the first medical advice.
- ***Patient Guidance*****:** Based on the AI-generated advice, guidance is given or next steps are suggested to users.

4. Profile Management

Preconditions:

***Authenticated User*:** It is mandatory for a person to be logged in so as s/he can access and update his/her profile(s).

Postconditions:

- ***Profile Updates***: Personal and medical information are updated successfully by those people concerned in this work.
- ***Data Integrity*:** Updated information accuracy and consistency are ensured through this system

5. Administrator

***FunctionsPreconditions*:**

- **Admin Credentials:** Access by administrators requires verification first.

Postconditions:

- ***System Configurations*:** Administrators manage people or things like user's settings aswell as configurations regarding systems' behavior.
- ***Monitoring and Reporting***: Admins explore logs, create reports and oversee system performance.

4.7. Activity Diagram

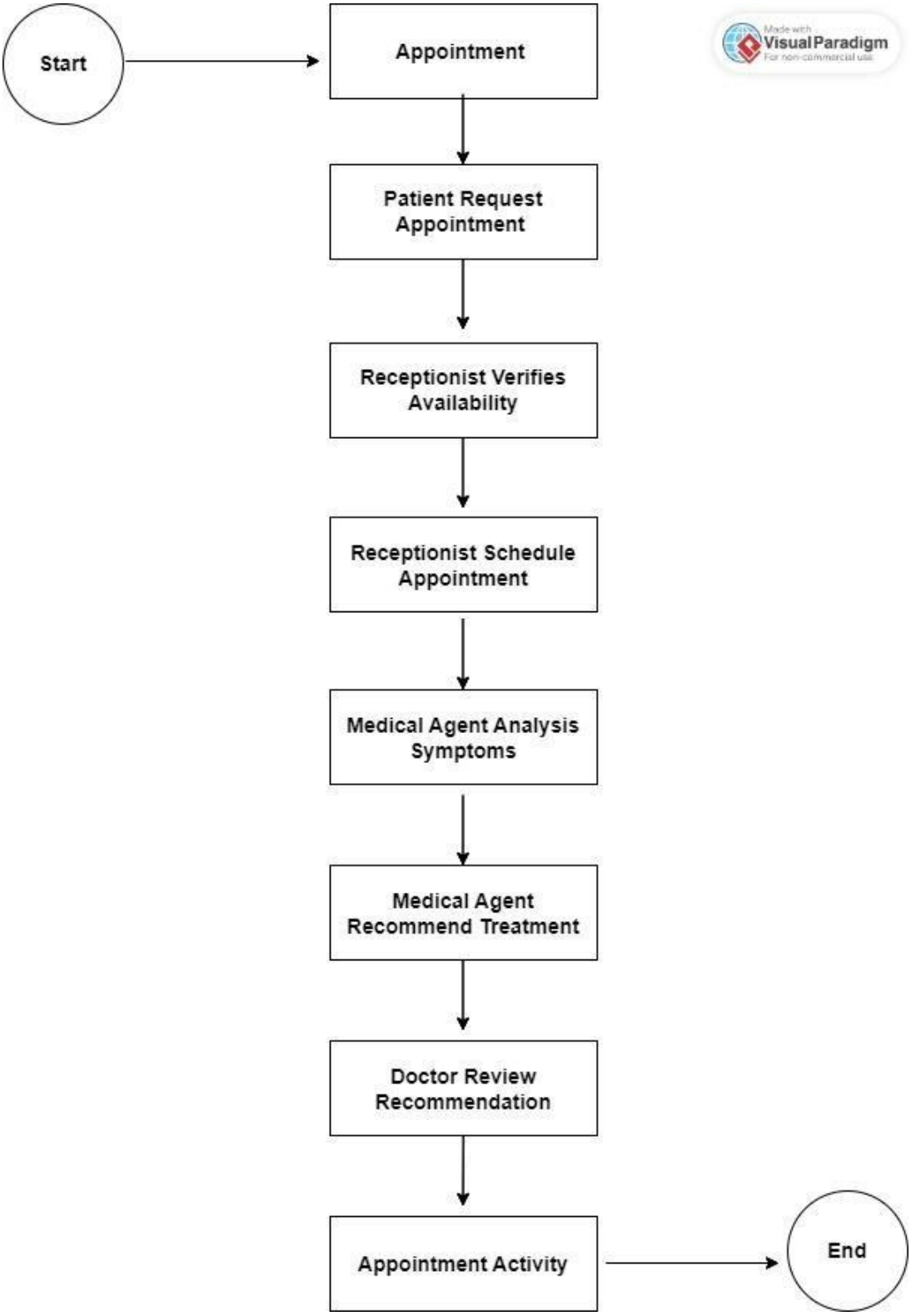


Figure 4.6 Activity Diagram

In this activity diagram:

- The first move is Start Appointment.
- This is what the patient does in Patient Requests Appointment activity that triggers this event.
- The clerk checks with a doctor if he is available in Receptionist Verifies Availability.
- If there is, a clerk makes arrangements for an appointment in ReceptionistSchedules Appointment
- Afterward, AI medical agent inspects symptoms in Medical Agent Analyzes Symptoms.
- Based on this analysis, the ai medic recommends treatment to patients in “Medical Agent Recommends Treatment”.
- Throughout an activity called Doctor Reviews Recommendation, this recommendation is assessed by a physician
- Lastly, process ends at End Appointment.

4.8. State Transition Diagram

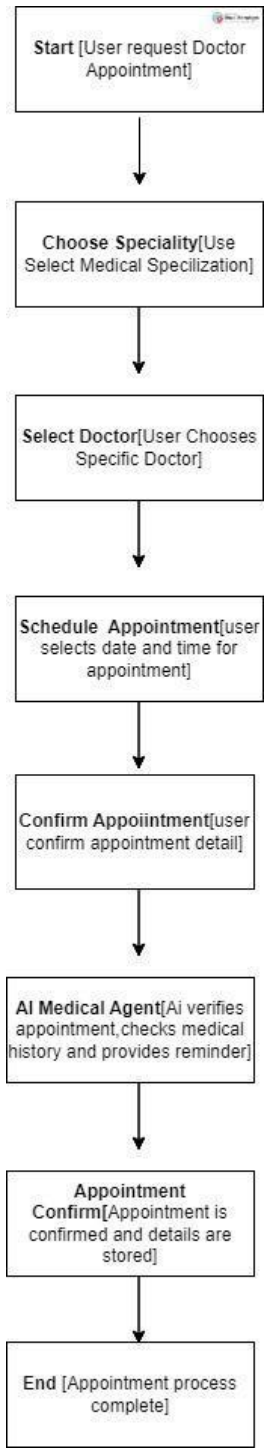


Figure 4.7 State Transition Diagram

Explanation of key states:

Explanations of the Key States:

1. ***Start*:** The beginning phase that a user initiates to make an appointment.
2. ***Choose Speciality*:** Users declare their preferred medical field during this stage.
3. ***Select Doctor*:** User picks a specialist after deciding on a treatment modality in medicine.
4. ***Schedule Appointment*:** A date and time available for consultation is indicated by the User.
5. ***Confirm Appointment*:** At this point users acknowledge information about appointments they made while verifying if it is correct;
6. ***AI Medical Agent:*** After the appointment has been confirmed, the artificial intelligence ai agent verifies it and simultaneously checks for any helpful reminders or specialized messages out of patients medical records.
7. ***Appointment Confirmed*:** When all details about an appointment are well documented and saved too, we say that appointment is confirmed.
8. ***End*:** The end may be defined as a situation whereby both parties have concluded whatever was expected from each of them at this time

4.9. Component Diagram

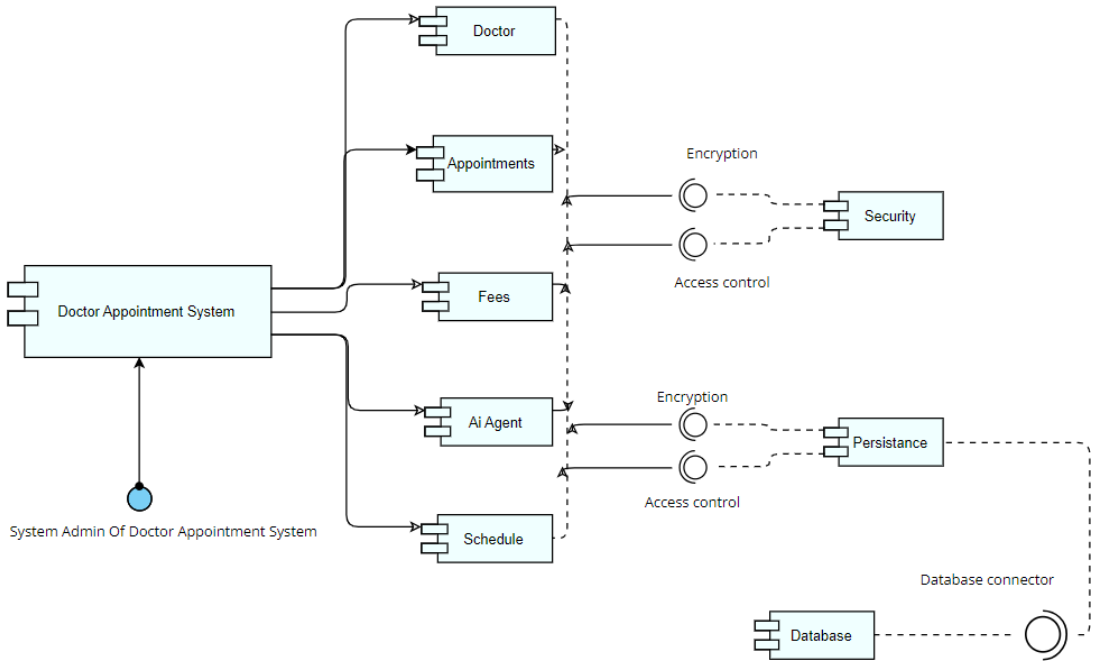


Figure 4.8 Component Diagram

4.10. Deployment Diagram

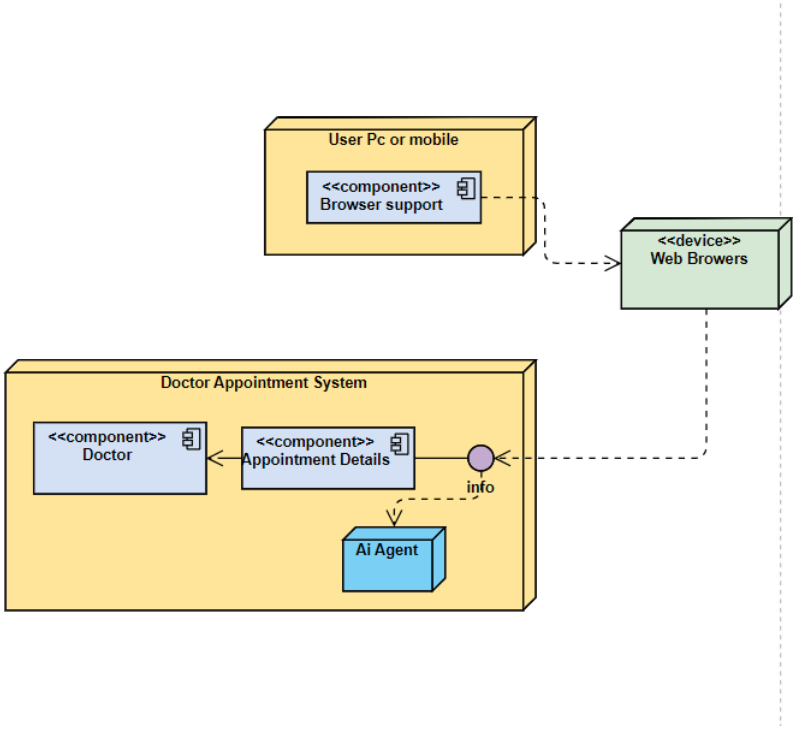


Figure 4.9 Deployment Diagram

4.11. Data Flow diagram [*only if structured approach is used - Level 0 and 1*]

Data Flow Diagram (Level 0)

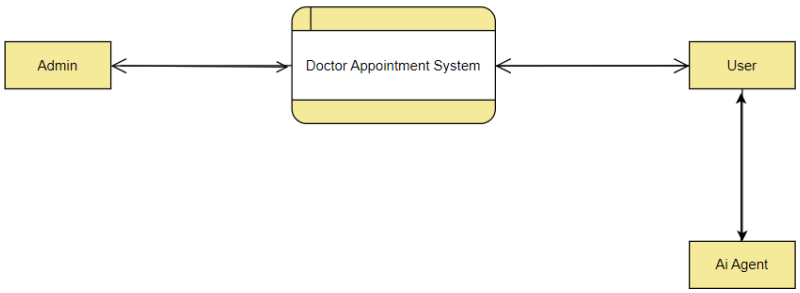
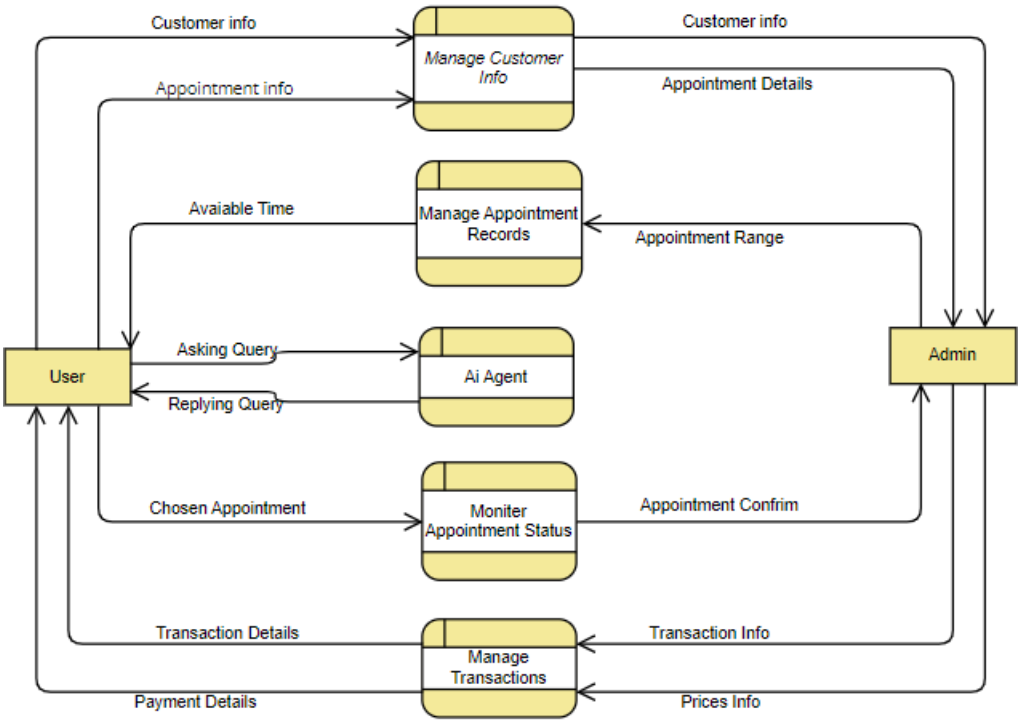


Figure 4.10 Data Flow Diagram

Doctor Appointment System



Data Flow Diagram Level 1

Figure 4.11 Data Flow Diagram Level 1

Chapter 5
Implementation

Chapter 5: Implementation

This chapter, being the fifth phase of system development, is where the project's design becomes an operational system. It explains how the actual coding takes place, development, component integration and deployment strategies. The section will describe all technical procedures, tools and methodologies that were used to bring out this imagined system. This part focuses on converting conceptual designs into physical software programs while emphasizing technological complexities through coding standards as well as stages of development. Furthermore, it highlights points of challenges met during implementation while putting forth strategies for successful execution in order to comprehend practically the practical application of this knowledge regarding system development techniques.

5.1. Important Flow Control/Pseudo codes

```
PROCEDURE ScheduleAppointment(patientID, doctorID,date,time):
  IF appointmentSlotAvailable(doctorID, date, time) THEN
    CREATE_APPOINTMENT(patientID, doctorID, date, time)
    RETURN Appointment booked successfully;
  ELSE
    RETURN Selected slot not available. Please choose another.
  END IF
END PROCEDURE
```

Doctor Details Retrieval:

```
FUNCTION getDoctorDetails(doctorID):
  doctor = QUERY_DB("SELECT * FROM doctors WHERE doctorID = ?",
  doctorID)RETURN doctor
END FUNCTION
```

Patient Login:

```
PROCEDURE patientLogin(username, password):
```

```
IF isValidPatient(username,password)
  THENSET_SESSION(patientID)
  RETURN Login successful.
ELSE
  RETURN Invalid credentials. Please try
again;END IF
END PROCEDURE
```

Patient Signup:

```
PROCEDURE patientSignup(name,email, username,password);
IF isUniqueUsername(username) THEN
CREATE_PATIENT(name,email,username,password)
RETURN Signup successful. Proceed to login;
ELSE
RETURN Username already exists. Please choose another.
ENDIF
END PROCEDURE
```

Cancel Appointment:

```
PROCEDURE cancelAppointment(appointmentID):
  appointment =
  GET_APPOINTMENT_DETAILS(appointmentID) IF
  appointmentExists(appointment) THEN
    IF appointmentNotPastCancellationTime(appointment) THEN
      DELETE_APPOINTMENT(appointmentID)
      RETURN "Appointment canceled
    successfully."ELSE
      RETURN "Cancellation time window has
    passed."END IF
  ELSE
    RETURN "Appointment not
  found."END IF
```

END PROCEDURE

5.2. Components, Libraries, Web Services and stubs

Components:

1 ***Frontend Framework (React)*:** Those are graphical elements meant to improve user interfaces.

2 ***Backend (Node.js, Express):*** It manages server-side actions and controls the API endpoints.***Database (MongoDB):*** This stores appointment data in a manner that is fast and efficient. Libraries:

3 ***Langchain:*** Enables easy interpretation of what users say by the AI agent. ***Date-time Libraries (moment.js):*** Aids in handling time-based functionalities.Web Services;

4 ***AI Medical Agent API (Custom or third-party):*** Gives advice on medical issues and helps inbooking appointments.

5 ***Email Service (SendGrid):*** The platform confirms appointments as well as send reminders tousers via email:

Stubs:

Mock Appointment Service: This one pretends to schedule appointments and retrieve dataduring development or testing phases.

Fake AI Response Module: To test integration and functionality it mimics responses from the ai agent.

5.3. Deployment Environment

Server Infrastructure:

***Web hosting*:** Heroku

***Application Deployment*:** It will be necessary to use heroku for the purpose of applicationhosting and deploying.

***Libraries*:** Langchain

Database Environment:

MongoDB Atlas

***Version*:** Latest

***Configuration* :** Managed cloud-based MongoDB cluster with automated backup and expandable storage.

Network Configuration:

Port Configuration:

***HTTP* :** Port 80,3000,5000

5.4. Tools and Techniques

Programming Languages: JavaScript can be used for frontend as well as backend while Python can be employed for Artificial Intelligence (AI) integration.
Database Management: For appointment and user data storage, MongoDB is used.

Development Environment: Visual Studio Code

Version Control: Using Git to track changes and work with others.

AI Libraries: There are various AI libraries like TensorFlow, Sci kit-learn or any other python library for AI agent implementation or js libraries for AI Agent.

Techniques:

MERN Stack Development: Using MongoDB, Express.js, React, Node.js to create a solid web application
API Integration: Incorporating APIs that will allow the booking system to include AI functions.
Responsive Design: Utilizing CSS frameworks such as Bootstrap to make sure the website works on mobile devices too.
Testing Frameworks: This particular project should use Jest on the front end and Mocha/Chaion the back end because they are ideal for testing in this case respectively.

Deployment Services: Examples include Heroku, AWS where hosting and deployment can take place.

5.5. Best Practices / Coding Standards

Best Practices:

***Uniformity in Naming Conventions is Essential*:** Ensure that the codes use the same naming convention throughout the whole package such as variables written in camelCase format for easy reading purposes.

***Always Comment Your Code*:** Go ahead and comment out your code through commentingtools like JSDoc for backend, or react style comments on the front end so that you can extensively explain everything about it.

***Express.js Error Handling Middleware*:** Established Express.js error handling middleware should exist.

***Helmet Middlewares among other Security Measures*:** Construct preventive measures ofhttp headers such as helmet middlewares that guards against security issues.

***Git Version Control*:** Use Git as a version control system create feature/fix branches and do frequent code reviews. Thus, having feature /fix branches is essential.

Backend Development Tends to Follow Test-Driven Development (TDD): The backend uses TDD frameworks such as jest while for fronted React Testing Library suffices. Also, go ahead with using testing frameworks such as jest for the backend part which focuses on robustness and reliability but React Testing Library best practices are recommended by industry standards for developing frontend applications.

Performance Optimization: API optimization at the back end together with lazy loading techniques for frontend components improves performance.

Coding Standards:

Consistent Indentation and Formatting: Ensure continuity in the indentation and coding format(ESLint/Prettier configurations) used throughout the project.

Descriptive Variable Names: In front-end code, as well as backend code, use meaningful and descriptivename for variables and functions.

Avoid Hardcoding: To minimize hardcoding, use environment variables or configuration files, or constants where necessary.

Code Reusability: To avoid repetition of codes, encapsulate reusable functionalities in functions, components or hooks.

Follow Design Patterns: For state management in React use Redux/MobX among others and MVC (Model-View-Controller) for backend APIs are some examples of common design patterns to implement.

Consistent Architecture: For easier navigation and maintenance always maintain a consistent folder structure such as grouping features/componentsby functionality).

Code Reviews and Pair Programming: It is good to perform regular code reviews that may also involve pair programming so as to ensure adherence to standards while identifying areas that need improvement.

5.6. Version Control

Version Control with Git:

5.6. Version Control

Git the version control system.

***Repository construction design*:** Develop separate repositories for frontend React, backend Node.js/Express.js, and ai-related codebase (Python), on such platforms as GitHub or GitLab.

***Branching strategy*:** Use a branching model such as GitFlow to manage feature development, releases and hotfixes. Devise branches for development, staging, andproduction.

***Commit algorithms*:** Ensure that commit messages have apt descriptions for every commitmade; adopt atomic commits (a smaller focused purpose commit) thus making it easier reviewing and understanding the code.

Collaboration Workflow: The team members should be encouraged to work on feature branches; merging changes made via pull requests. Review codes, discuss changes and conduct tests before merging them into the main branches.

Integrate issue tracking tools like Jira Trello or Github Issues to your version management system so that every change an individual makes can be traced back to specific tasks or issues. CI/CD pipelines setup (using tools like GitHub Actions, Jenkins or CircleCI) that will automate testing and deployment processes for frontend, backend, AI components.

Appendices

Appendix A: Information / Promotional Material

Supplementary Materials:

This appendix includes other supportive materials which are meant to help the users and stakeholders have an understanding of how Doctor Appointment System with AI Integration works.

User Guides & Manuals:

Full explanations about how to use the system, functionalities of AI and management of appointments.

Promotional Material:

Brochures, flyers, and marketable items that describe what the system does.

A.1. Broacher

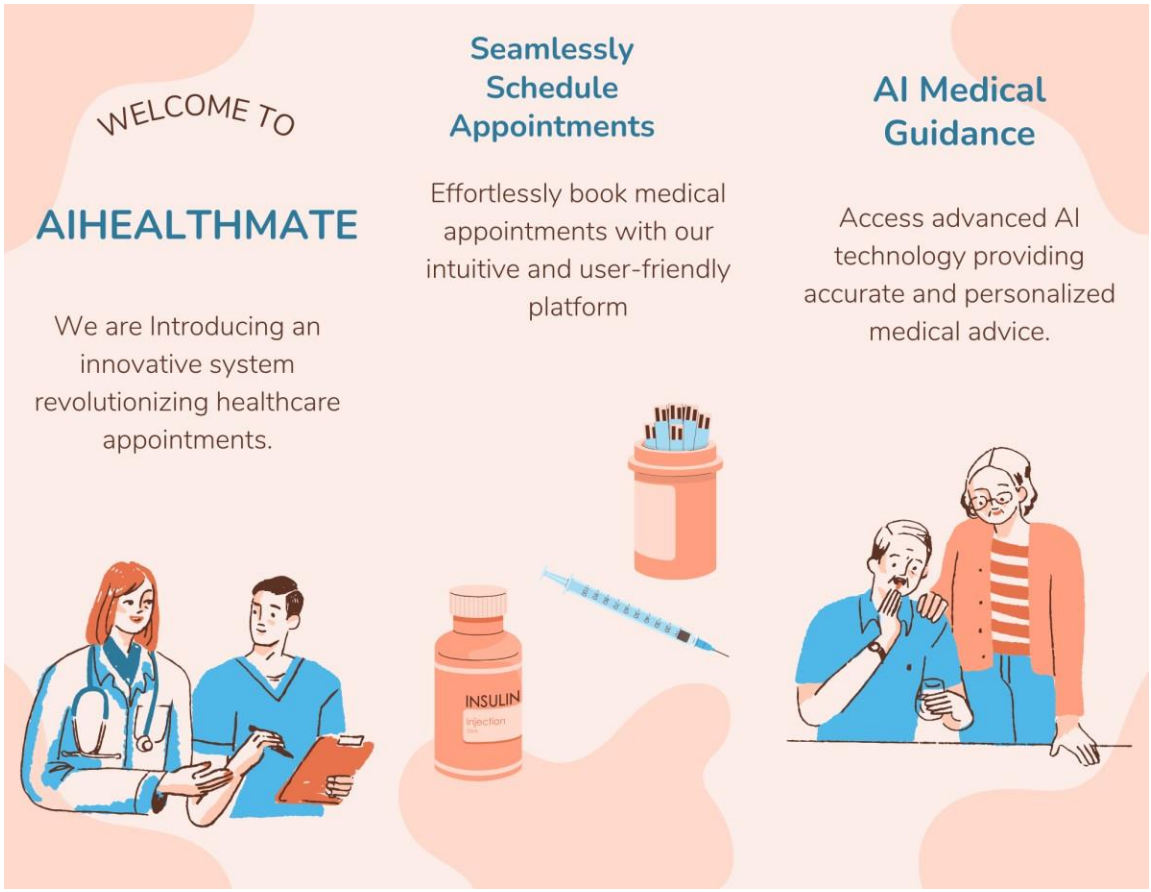


Figure 5.1 Broacher

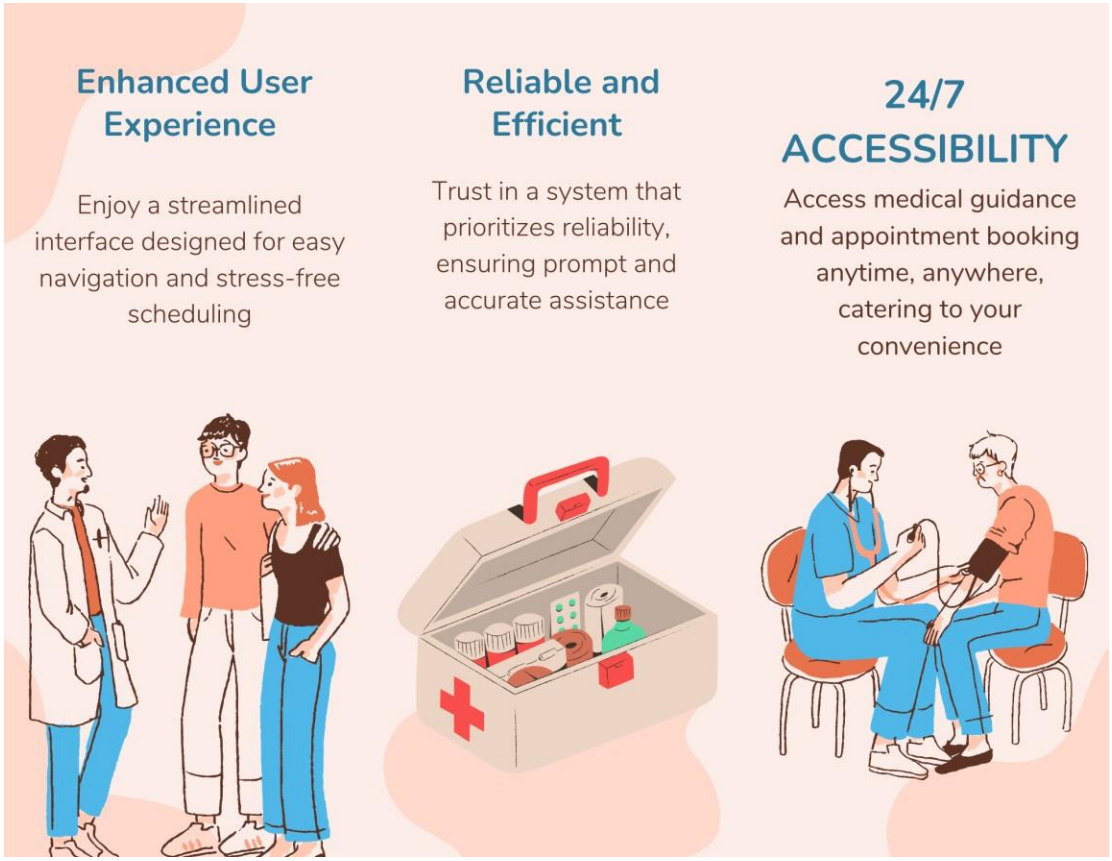


Figure 5.2 Broacher

A.2. Flyer:

Figure 5.3 Flyer

A.3. Banner:

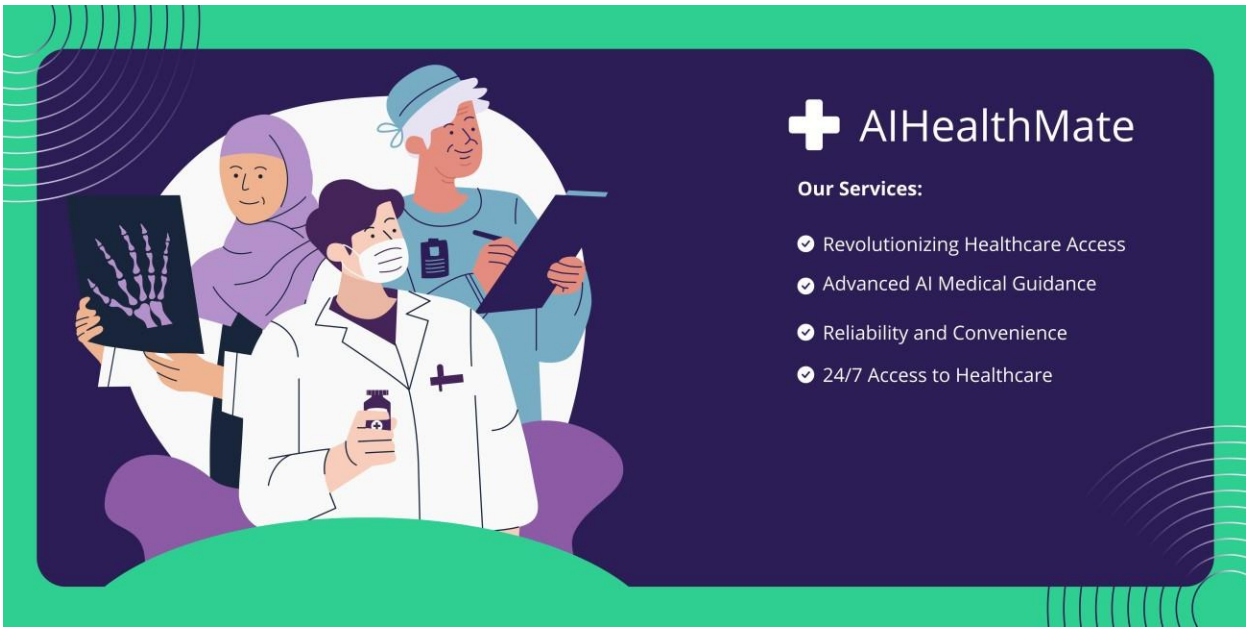


Figure 5.4 Banner

Reference and Bibliography

Reference and Bibliography

Hylton, A., & Sankaranarayanan, S. (2012). Application of Intelligent Agents inHospital Appointment Scheduling System. International Journal of Computer Theory and Engineering, 625-630.

5

Hylton, A., & Sankaranarayanan, S. (2009). Application of Intelligent Agents inHospital Appointment Scheduling System. In Applications of Intelligent Agents (pp. 12-29). Springer, Berlin, Heidelberg.

5

Hylton, A., & Sankaranarayanan, S. (2008). Applications of Intelligent Agents inHospital Appointment Scheduling System. In Applications of Intelligent Agents (pp. 12-29). Springer, Berlin, Heidelberg.

5

Hylton, A., & Sankaranarayanan, S. (1998). Applications of Intelligent Agents inHospital Appointment Scheduling System. In Applications of Intelligent Agents (pp. 12-29). Springer, Berlin, Heidelberg

Index

Index