KINGDOM OF SAUDI ARABIA
Ministry of Higher Education
Taibah University
College of Computer Science and Engineering

المملكة العربية السعودية
وزارة التعليم العالي
جامعة طيبة
كلية علوم و هندسة الحاسبات

AI Diagnosing System

A Project Report Submitted to Fulfill the Requirements of CS289

By Group (4)

Layan Hosni Esmat (SID: 4450283)

Manahel Alahmadi (SID: 4450015)

Yasmeen Alrashidi (SID: 4456801)

Munifah Aljabri (SID: 4455350)

Mawaddah Almowallad (SID: 4456191)

Supervised By

Dr.ohood Alrohili

30 November, 2025

Abstract

In an era where technology is rapidly transforming the healthcare sector, this project aims to enhance the Sehhaty application by integrating artificial intelligence (AI) to streamline the diagnostic and appointment booking process. The core objective of the project is to reduce the time and effort patients spend navigating the healthcare system by automating the preliminary diagnosis and referral stages.

The proposed system allows users to input their symptoms directly into the application. Using AI algorithms trained on a broad medical dataset, the app performs a preliminary diagnosis and automatically schedules an appointment with the most suitable medical specialist. This method not only expedites the patient's access to care but also reduces the load on healthcare facilities by directing patients to the right provider from the outset.

This project holds significant potential for improving the efficiency of medical services in Saudi Arabia, aligning with Vision 2030 goals of digital transformation in healthcare. By leveraging AI, the enhanced Sehhaty application represents a step forward in personalized, accessible, and smart healthcare delivery

Table of Content

Chapter 1: Introduction

1.1 Overview

Healthcare in Saudi Arabia is now more accessible with the Sehaty app, which allows patients to book appointments and access a wide range of services. Patients face challenges in assigning patients to the appropriate clinic. The usual approach involves patients making an appointment with a general practitioner and then being referred to a specialist. The System finds a solution to assign patients directly to specialists. It helps reduce waiting time and assists patients by developing an artificially intelligent system that provides a preliminary diagnosis based on symptoms then automatically refers patients to the appropriate specialist.

We will use the UP model since the project is large and requires gradual development with continuous reviews.

In conjunction with the AI models, we intend to use the natural language processing model (NLP) to understand and analyze texts and identify diseases ,the deep learning model to gather and classify information as well as the reinforcement learning model to continuously improve system performance over time.

1.2 Client Context

A Governmental Health Institution desires to improve a healthcare application using Artificial Intelligence.

1.3 Similar Projects

-Skin Image Search
-Buoy Health
-k Health
https://www.technologyally.com/blog/ai-medical-diagnosis-apps/

1.4 Problem Statement

With the vision of 2030 and the accelerated evolution of science, the healthcare system in Saudi Arabia is in continuous improvement, especially when it comes to the primary

healthcare centers' systems, as there was a positive transformation, yet there still lies an issue.

The issue is, when a patient uses one of the governmental health websites to book a doctor's appointment, multiple referrals can occur due to the lack of initial diagnosis on those websites, which leads to constant delays.

By developing an AI system that can give the patient an initial diagnosis, then books an appointment with the proper specialized department to skip the Triage with general practitioners in order to save both the doctor's and the patient's time and energy.

The integration of this AI powered software into government healthcare websites will result in an improvement in patient care, which is a key priority to any healthcare department.

Chapter 2: Requirements Elicitation

2.1 Methodology

Interview Question

- What AI features would you like to see added to the app?

I would like to add a feature that provides a preliminary diagnosis of symptoms and considers past illnesses along with current conditions to achieve a more accurate diagnosis.

- Do you prefer entering symptoms manually, or using voice or images?

I prefer entering symptoms manually because it's more accurate and easier, .especially in cases that require detailed descriptions

- What is your opinion on including patients' medical records in the app?

Including the patient's medical records is very important as it helps provide an .accurate diagnosis, but there must be strong data protection

- Would you prefer to directly communicate with a doctor after receiving the initial diagnosis?

Yes, I'd like to speak directly with a doctor after receiving the initial diagnosis .so I can ask questions and get advice and instructions

- What concerns or challenges do you expect from using an AI-powered app like this?

I'm concerned about receiving an incorrect diagnosis or relying too much on AI .without human involvement. Privacy is also very important to me

- Would you prefer an automatic appointment booking after diagnosis, or to choose the doctor yourself?

I prefer to choose the doctor myself to make sure I'm comfortable with them or to .select a doctor I already know

- Would you like to see an explanation of the diagnosis or recommendations before booking?

Definitely, I would like to see an explanation of the diagnosis or recommendations before booking an appointment. It gives me more confidence in the result and helps .me understand my condition better

- How would you prefer to receive appointment confirmation or notifications? (Push notifications, SMS, in-app)?

.I prefer in-app notifications, with an SMS backup in case I miss the notification

2.2 Requirements Listing

2.2.1 Functional Requirements

- **The system shall allow users to submit symptoms via a designated interface.**
- **The “preliminary diagnosis” section shall include a multiple-choice questionnaire that consists of chronic diseases, previously diagnosed diseases and occurring symptoms, along with a text box that gives the user an option of entering more specific symptoms.**
- **Using NLP, the system must be able to recognize and analyze text in order to identify the possible causes of listed symptoms.**
- **By employing deep learning, the program is expected to recognize the relationships between symptoms to provide a preliminary diagnosis and generate a patient report with the gathered information.**
- **Based on the identified diagnosis, the system should be able to access nearby hospitals and clinic systems with the intention of booking an appointment with the appropriate medical consultant.**
- **The system should allow the user to choose one of the available appointments with the appropriate doctor.**
- **The system should send a message containing the appointment information to the user’s number.**

2.2.2 Non-Functional Requirements

A. Performance

- **The system must display initial symptom triage results in less than one minute after the user enters symptoms. The system must support 10,000 active users without performance degradation.**
- **The accuracy of artificial intelligence in selecting specialists based on symptoms must be no less than 85% in the initial stages, with gradual improvement thereafter.**

B. Security

- **User data must be encrypted and protected.**

C. Quality

- **The system must be easy to use for all ages and be available in both English and Arabic.**
- **The system must be available 24/7.**
- **Periodic system testing must be conducted to ensure quality**.

D. General System Constraints

- **The system must support the addition of new information and updating of patient data.**

Chapter 3: System Models

3.1.1 Use Case Descriptions

Use Case 1: Entering Symptoms

Actors: patient

Base Case:

- **The user logs in to the application.**
- **The user navigates to "Enter Symptoms."**
- **The user types the symptoms.**
- **The user submits the symptoms successfully.**
- **The system stores the symptoms for analysis.**
- **Inclusion Case:**
- **Validates the user's input (ensures that the user entered the symptoms correctly).**
- **Extension scenarios:**
- **If the user submits empty symptoms, or numbers instead of text, the system displays an error message: "Please enter at least one symptom."**
- **In case of a lost connection, the data is saved locally and sent once the internet is**
- **restored.**

Pre-condition:

The patient must be logged in to the application.

Use Case 2: Analyze Symptoms and Diagnose

Base Case:

- **System receives the entered symptoms.**
- **AI model analyzes the symptoms.**
- **System suggests a probable diagnosis to the user.**
- **Diagnosis is saved to the user's medical record.**
- **Inclusion Case:**

Access Symptoms Database (The system enters the symptom database to match symptoms.).

- **Extension:**
- **If AI model fails or cannot find a diagnosis, the system suggests seeing a general doctor.**

Pre-condition: Symptoms must have been entered and saved successfully by the patient.

Use Case 3: Generate Patient Report

Actors:

- **Doctor**
- **Patient**

Goal:

To view and generate a detailed medical report for a specific patient.

Base Case :

1.The doctor logs into the system.

2.The doctor searches for the patient.

3.The system displays the patient's medical history.

4.The doctor selects the "Generate Report" option.

5.The system compiles the report including symptoms, diagnosis, and appointment history.

6.The doctor adds notes or recommendations.

7.The report is saved to the patient's profile.

Preconditions:

- **The doctor must be logged in.**
- **The patient must already exist in the system.**

Extensions Case

- **If the patient is not found → show "Patient not found" message.**
- **m. If data is incomplete → allow partial report generation with a warning.**

Included Use Case:

- **View Patient Medical History → This is automatically included during step 3.**

Use Case 4: Book Appointment

Actors: patient

Base Case:

- **Based on diagnosis, the system finds the specialized doctor.**
- **The system checks available time slots.**
- **The system presents the available time slots to the patient.**
- **The patient selects the preferred time slot.**
- **The system confirms the booking.**
- **Appointment details are saved.**

Inclusion Case:

Verify Doctor Availability (Make sure your doctor's appointments are available).

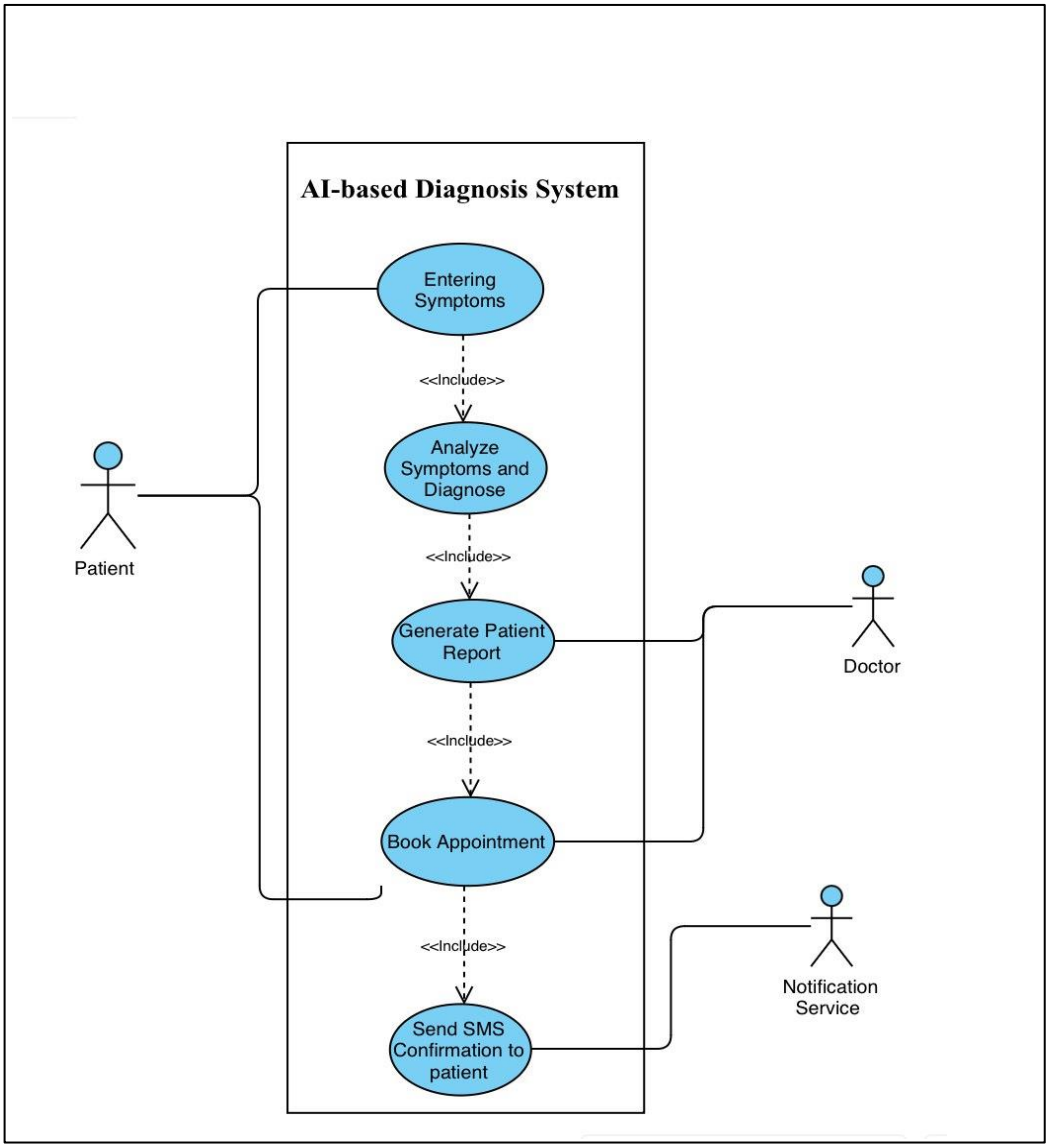
- **Extension:**
- **If no slots are available, the system suggests the next available date.**
- **If booking fails, the system notifies the patient.**
- **Pre-condition: A diagnosis must have been completed, and the patient must be registered in the system.**

Use Case 5: Send SMS Confirmation to Patient

Base Case:

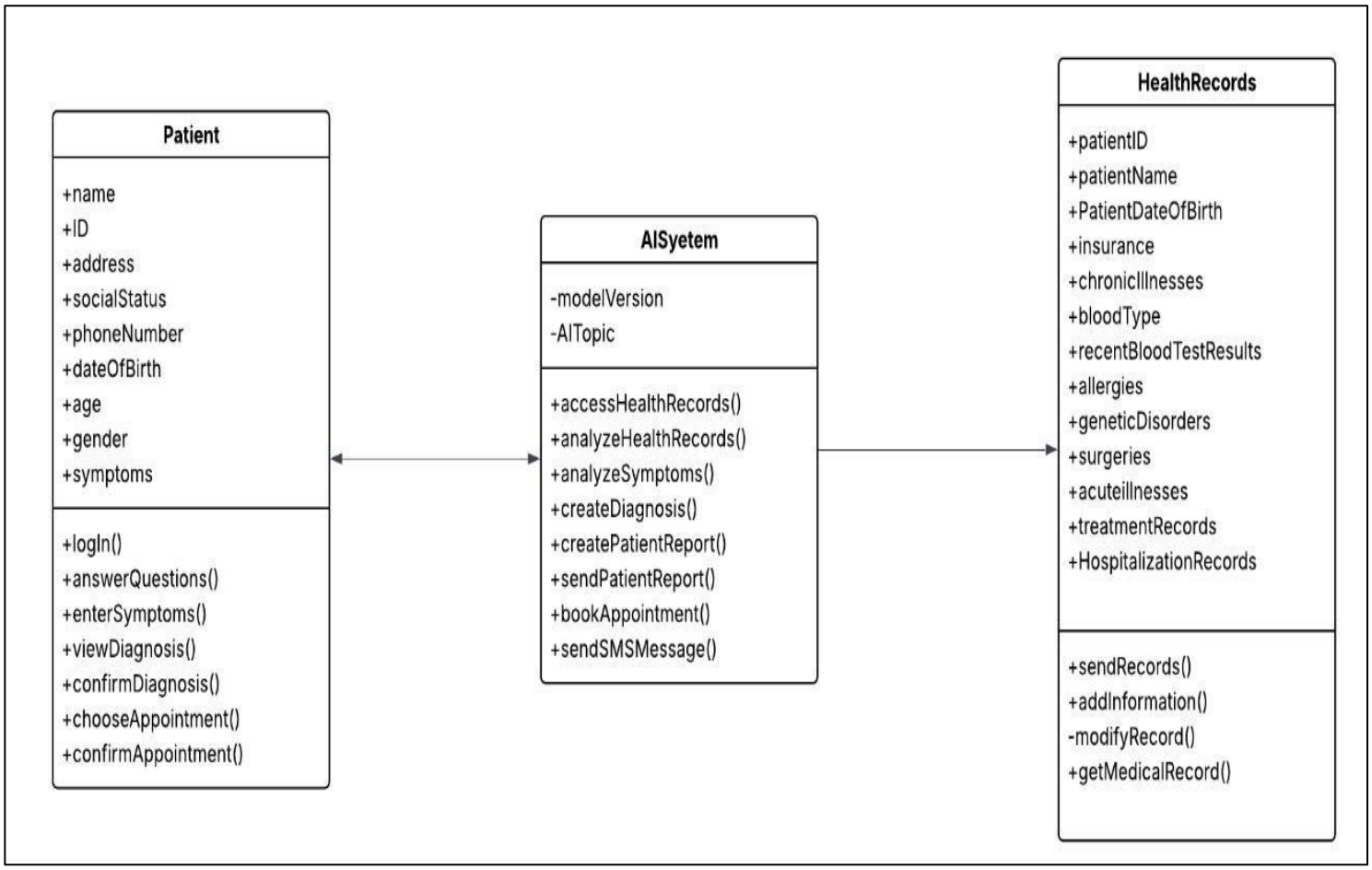
- **After booking the appointment, the system generates an SMS message.**
- **System sends the SMS with appointment details (time, date, doctor name, clinic location) to the patient's registered number.**
- **Inclusion Case:**
- **Format SMS Message**
- **Extension:**
- **If SMS gateway fails, system sends an in-app notification instead.**
- **If the user's phone number is invalid, system prompts user to update contact information**

3.1.2 Use Case Diagram



3.2 Object Model

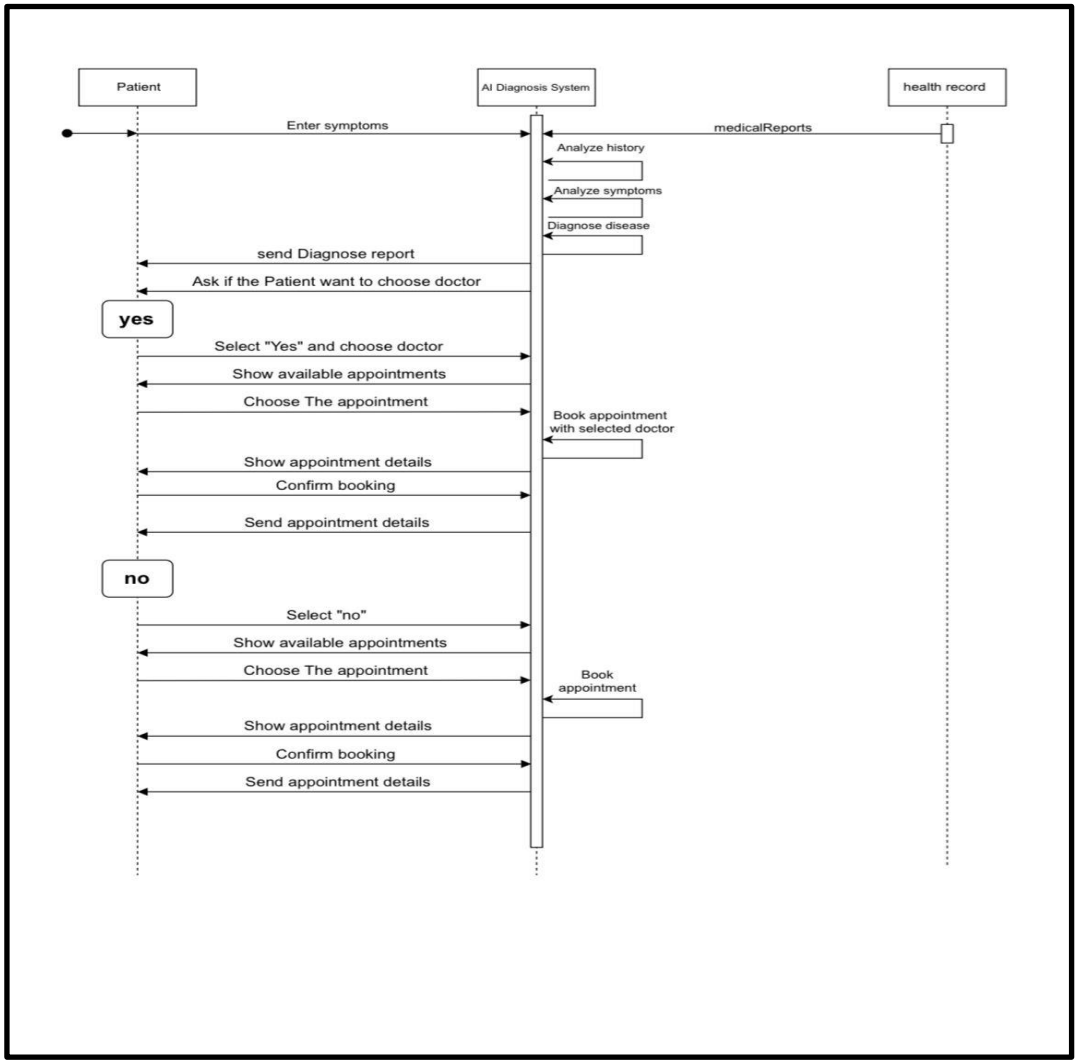
3.2.1 Class Diagram



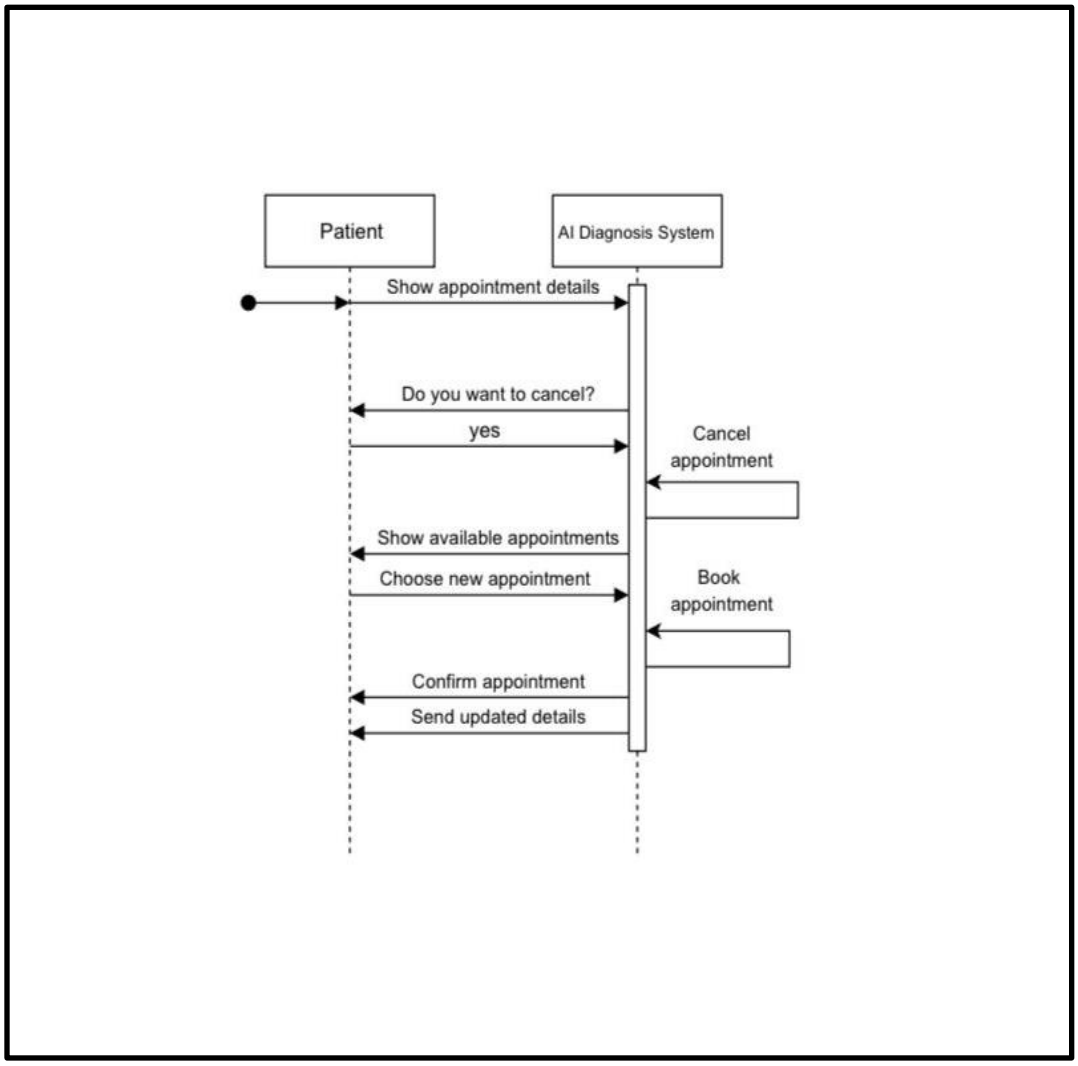
3.3 Dynamic Model

3.3.1 Sequence Diagrams

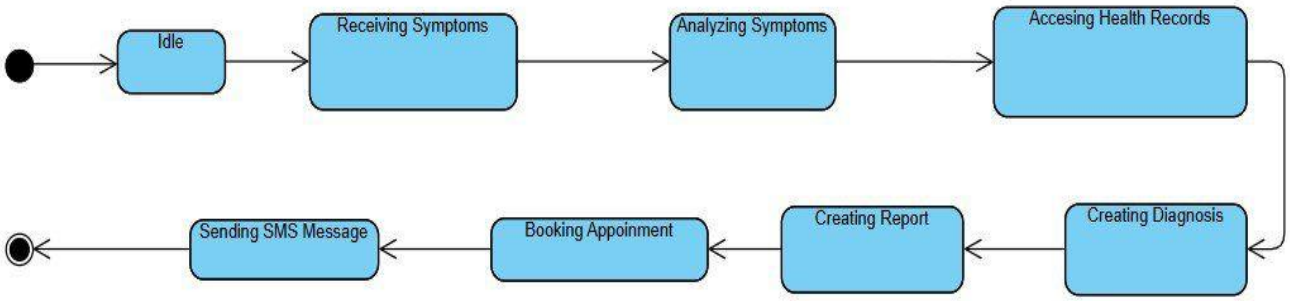
1-



2-


Patient
AI Diagnosis System
Show appointment details
Do you want to cancel?
yes
Cancel appointment
Show available appointments
Choose new appointment
Book appointment
Confirm appointment
Send updated details

3.3.2 State Diagram



Chapter 4: Architecture Models & Design Patterns

(Approx. 300–500 words)

4.1 Architecture Models

- 4.1.1 Overview:

The system’s architecture is designed to diagnose patients and

Refer them to a medical specialist at the nearest hospital or

a medical center with a department that suits the patient's needs.

The architecture is divided into four major components:

1- User Interface (UI)

The user interface is the front-end layer where the user interacts with the system as it

allows the patient to enter their symptoms, and the system displays

a preliminary diagnosis. Eventually, it displays available appointments.

2- AI model

The AI model analyzes the patient’s medical records and symptoms,

then it assigns the patient to a department compatible with their

initial diagnosis, additionally, the system will provide data to the AI algorithm in this project we used three AI models :

A- deep learning model

In this project we have data, symptoms, and various diseases, the system based on them will choose the diagnosis.

B- Reinforcement model

The system will need to develop its information continuously.

C- ML(machine learning):

Since we have entered the patient's history, there will be analyses for his blood tests, which are considered structured data, and the ML will work with them.

D- NLP(natural language processing):

The patient provides a detailed description of their symptoms, the system semantically analyzes text to get an understanding beyond syntax by interpreting the meaning of each listed symptom. Therefore, getting to the right diagnosis.

3- Data Storage

This layer deals with data storage and retrieval. It includes.

user symptoms, health records, available appointments and AI

learning data, ensuring that all system interactions are persistent

and secure.

4-Processing Unit

This unit checks if the patient input matches the symptoms with possible conditions using the AI module, it also checks the available appointments.

These components work together to support the system in order to deliver both the diagnosis and the appointment management. The separation of layers guarantees that the system maintained and upgraded each part independently.

- ### 4.1.2 Architectural style:

The chosen architectural style for our AI-enhanced healthcare application is the Data Flow Architecture. In this style, data flows sequentially through a series of loosely-coupled stages, with each stage performing a specific processing task before passing the output to the next.

This approach is particularly suitable for AI-based solutions because it mirrors the natural data flow in intelligent systems. It also offers high modularity, scalability, and ease of maintenance—making it ideal for systems that may require regular updates to individual AI models or modules. If a specific component, such as the diagnosis model, needs improvement, it can be replaced independently without disrupting the entire system.

System Components (as Pipeline Stages):

1.Input & Preprocessing Module:

Users input symptoms in natural language. This module performs text cleaning, tokenization, and keyword extraction to prepare data for analysis.

2.AI Diagnosis Engine:

Processes the preprocessed symptoms using an AI model (e.g., NLP-based classifier) to generate a preliminary diagnosis.

3.Decision & Recommendation Module:

Interprets the AI's output and decides whether to recommend medical attention. It then suggests suitable nearby clinics or doctors.

4.Appointment Booking Module:

Displays available time slots and lets the user select a preferred appointment. Integrates with doctor schedules.

5.Confirmation & Notification Module:

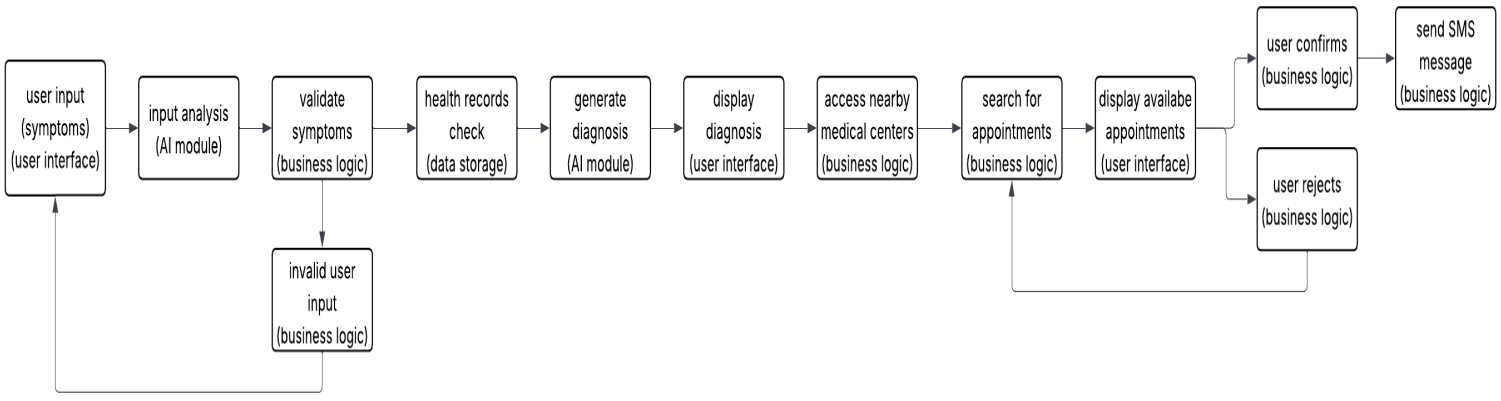
After appointment selection, the system requests user confirmation and sends an SMS to confirm the booking.

6.Feedback & Learning Module (optional):

Gathers user feedback to retrain or fine-tune AI models, improving accuracy over time

This module is optional but can be added in future iterations to enable continuous learning from user feedback, enhancing diagnostic accuracy.

- ## 4.1.3 Architecture diagram:



4.2 Design Patterns

Design Patterns:

1. Creational Patterns:

Abstract Factory:

It is used to create linked objects such as symptom analyzers, for example, when adding a new diagnosis or analysis, you do not have to rewrite the code again, which provides the system with scalability without changing the basic structure.

Builder:

To build a complex object such as a diagnosis, it is built in steps and each step separately, allowing for easy maintenance, modification, and an organized form of code.

2. Structural Patterns:

Adapter:

It is used to link the system to external systems such as medical records and patient data so that it contributes to the independence of the system because each object is independent and therefore easy to reuse or replace.

Container:

It manages the creation of objects and ties them together, which makes it easier to maintain because it minimizes redundancy and simplifies the code, allowing me to easily change or test any part.

3. Behavioral Patterns:

Chain of Responsibility:

The request goes through several processes, such as symptom processing, starting from validating the symptoms to analyzing them as a chain, which makes it easy to remove a stage or add a stage without affecting the rest of the chain.

Command:

It is used to represent requests such as a reservation, representing it as an independent object and enabling many functions such as canceling a reservation or changing an appointment.

Chapter 6: Conclusion

(Approx. 100–200 words)

6.1 Conclusion

This project offered a significant and practical experience related to the theoretical concepts, and principles discussed in this curriculum. Each chapter provided us with a new level of learning, from requirement analysis to system design and implementation, which helped to fill the gap between the Practical and theoretical content.

In the Introduction phase:

Our project focused on cutting short the long steps we have in our healthcare system, The UP model was the first choice since the project is large and requires gradual development with continuous reviews and Based on the project idea, we decided to use the NLP model and the deep learning model, added to the reinforcement learning model.

In the Requirements Elicitation phase:

We focused on the methodology of collecting data from stakeholders, we chose the interview questions so we can gather the accurate requirements as it also provides flexibility in interactions, additionally it helps in building a good connection with the stakeholders and in discovering unmentioned requirements. Finally, in this stage, we focused on extracting the functional and nonfunctional requirements.

In the System Models Design phase:

In this phase we examined various use cases starting with entering the symptoms by the patient to sending the SMS Confirmation to the patient as the final step, and by adding three diagrams to clarify the project starting with use case diagram that shows how patients interact with the AI diagnosis system then, we used a UML class diagram that exhibits the internal structure of the system. In addition, we used the sequence diagram to express two scenarios, the first scenario describes two cases, the first one being: if the patient wants to choose a specific doctor, and the second case being: if the patient does not mind being set with any doctor. The second diagram shows if a patient wants to cancel their current appointment and change it. Finally, we have the state diagram which shows how the AI system class flows.

6.2 Lessons Learnt

The importance of clear requirements

We realized that defining system and user requirements precisely from the beginning is essential for building a solid foundation for future implementation.

User-centered design must be prioritized.

Even in the planning stage, we learned that designing a healthcare app requires a strong focus on usability, simplicity, and clarity.

Modular architecture supports flexibility.

Choosing the Data flow architecture helped us understand how separating system functions into stages can ease future development and updates.

Security and privacy should be considered early.

Even without implementation, we acknowledged the importance of planning for data protection, especially when dealing with sensitive medical information.

References

[1]Technology Ally, “28 Best AI Medical Diagnosis Apps in 2024,” *Technology Ally Blog*, Feb. 6, 2024. [Online]. Available: https://www.technologyally.com/blog/ai-medical-diagnosis-apps/

[2] E. Bisht, A. Rathi, and M. Chaudhary, *Hospital Management System: Software Engineering Project Report*, Mata Sundri College for Women, University of Delhi. [Online]. Available: https://mscw.ac.in/NAAC/Criteria1/Samples-of-ProjectWork_Fieldwork/Computer_Science/software_Engineering/Software%20Engineering/Hospital%20Managment%20Project%20SE-converted.pdf

[3] I. Sommerville, *Software Engineering*, 11th ed., Addison-Wesley, 2015.

[4] ohood Alrohili, Software *Engineering* Spring 2025, YouTube Playlist. [Online]. Available: https://youtube.com/playlist?list=PLTDREejaNLGRY0_S57t9R-1eJH0EXh_4z&si=KjXx1crlRk6I4c1n

[5] Wondershare Edraw. (2021). *UML State Diagram Tutorial* [video]. YouTube. https://youtu.be/iaX11vYFhZ4

[6] Grischa L. (2021). *Module* 05: *Architecture, Part 08: Pipe-and-Filter Architecture* [video]. YouTube. https://youtu.be/LXy2Y_lJqFE