Software Requirements Specification

for

Healthcare Diagnosis System

Version 1.0

Prepared by Batirniyaz Muratbaev

Project Group Number : #2 eng

student 1	**Batirniyaz**	**#2 eng**

Instructor: Saidobid Khusanov
Course: ML Course
Lab Section: Software
Teaching Assistant: Saidobid Khusanov
Date: 26.02.2025

NOTE: Made with help of gpt.

1 Introduction.....1
1.1 Document Purpose..... 1
1.2 Product Scope..... 1
1.3 Intended Audience and Document Overview..... 2
1.4 Definitions, Acronyms and Abbreviations..... 2
1.5 Document Conventions..... 3
2 Overall Description.....4
2.1 Product Overview..... 4
2.2 Product Functionality..... 4
2.3 Design and Implementation Constraints.....4
2.4 Assumptions and Dependencies..... 4
3 Specific Requirements.....5
3.1 External Interface Requirements..... 5
3.2 Functional Requirements.....6
3.3 Use Case Model..... 6
3.4 Class Model..... 7
3.5 Behavioural Model..... 7
4 Other Non-functional Requirements.....7
4.1 Performance Requirements.....7
4.2 Safety and Security Requirements..... 8
4.3 Software Quality Attributes..... 8
5 Other Requirements..... 8

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00

1. Introduction

This document outlines the **Software Requirements Specification (SRS)** for the **AI-Powered Healthcare Diagnosis System** developed by **MediAI**. The purpose of this document is to define the functional and non-functional requirements of the system, ensuring clarity for all stakeholders, including developers, project managers, healthcare professionals, and system testers.

The document provides a structured breakdown of the software's purpose, scope, intended audience, and terminology. It serves as a reference for all project contributors, ensuring alignment with the system's objectives.

1.1 Document Purpose

The purpose of this SRS document is to define the software requirements for the **AI-Powered Healthcare Diagnosis System**. The system is designed to assist doctors in diagnosing diseases by analyzing patient symptoms, lab results, and medical history using AI-driven insights. It will generate possible diagnoses and recommend further tests when necessary.

This document specifies the system's functional, performance, and security requirements while ensuring compliance with medical data privacy regulations. Additionally, it establishes a framework for integrating the AI system with **Electronic Health Records (EHRs)** and deploying it in a **cloud-based infrastructure** for scalability and accessibility.

This SRS will serve as a foundation for development, testing, and future enhancements of the system.

1.2 Product Scope

The **AI-Powered Healthcare Diagnosis System** is a **web-based** application that leverages **machine learning (ML) and artificial intelligence (AI)** to provide real-time diagnostic assistance to medical professionals. By analyzing **historical patient records, medical imaging data, and expert-validated diagnoses**, the system continuously improves its accuracy over time.

Key benefits of the system include:

- **Enhanced Diagnostic Accuracy**: Supports doctors by providing AI-generated suggestions based on patient data.
- **Faster Diagnosis**: Reduces the time required to diagnose diseases, improving patient outcomes.
- **Remote Accessibility**: Enables both doctors and patients to access the system remotely via a web application.
- **Integration with EHR Systems**: Ensures seamless access to past medical records for better decision-making.
- **Scalability & High Availability**: Designed for deployment on cloud infrastructure, making it accessible across multiple hospital branches.
- **Explainability & Transparency**: Provides confidence scores and highlights key medical factors influencing AI-generated diagnoses.

The system is intended to serve **doctors, medical staff, and patients**, improving efficiency in clinical decision-making and patient engagement.

1.3 Intended Audience and Document Overview

This document is intended for the following stakeholders:

- **Software Developers & Engineers**: To understand system functionalities and implementation requirements.
- **Project Managers**: To ensure alignment with project goals and deliverables.
- **Healthcare Professionals**: To review AI-driven diagnostic features and ensure clinical feasibility.
- **System Testers & Quality Assurance (QA) Teams**: To validate system performance and compliance.
- **Regulatory and Compliance Officers**: To ensure adherence to medical data privacy laws.
- **Professors & Academic Reviewers**: To evaluate the project's adherence to software engineering principles.

The SRS document is structured as follows:

1. **Introduction** – Provides an overview of the project, including its purpose, scope, and intended audience.
2. **Overall Description** – Describes the general factors affecting the product and its requirements.
3. **Specific Requirements** – Details functional and non-functional requirements, interface specifications, and system constraints.
4. **External Interface Requirements** – Covers interactions with users, software, and hardware components.
5. **System Features** – Specifies the core functionalities of the system.
6. **Other Requirements** – Includes regulatory, security, and compliance considerations.

1.4 Definitions, Acronyms and Abbreviations

Term	Definition
AI	Artificial Intelligence
EHR	Electronic Health Records
HIPAA	Health Insurance Portability and Accountability Act (Regulation for patient data privacy)
ML	Machine Learning
MediAI	The healthcare provider developing the AI-powered diagnosis system

QA	Quality Assurance
SRS	Software Requirements Specification

1.5 Document Conventions

This document follows the **IEEE SRS formatting standards** and maintains consistency in the following areas:

- **Font & Formatting**: Uses **Arial, size 11 or 12**, with **single-spacing** and **1-inch margins**.
- **Headings & Subheadings**: Numbered according to standard IEEE SRS guidelines (e.g., 1.1, 1.2, etc.).
- **Terminology & Notations**: All technical terms are clearly defined in the **Definitions, Acronyms, and Abbreviations** section.
- **Italicized Text**: Used for **comments** or **placeholders** in the template.

2. Overall Description

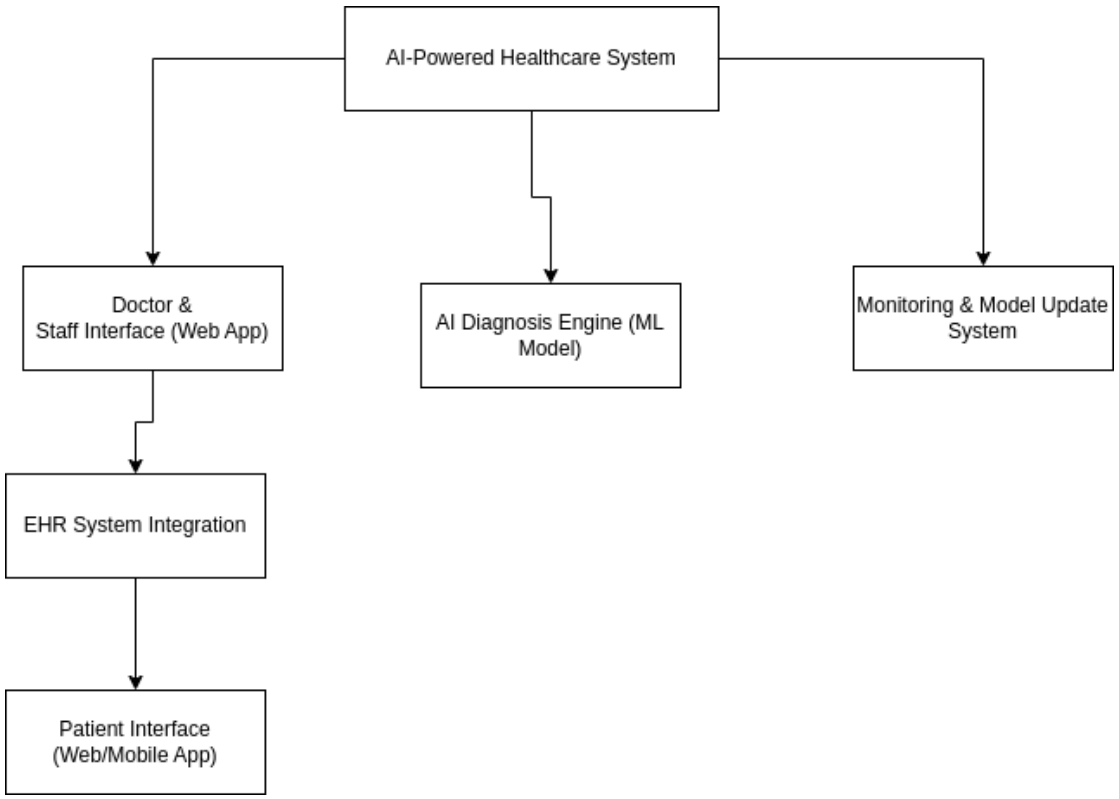
2.1 Product Overview

The **AI-Powered Healthcare Diagnosis System** is a **new, self-contained software product** designed to assist **medical professionals** in diagnosing diseases by analyzing **patient symptoms, lab test results, and medical history** using **artificial intelligence (AI) and machine learning (ML)** techniques.

This system is intended to be a **web-based** application that integrates with a hospital's **Electronic Health Records (EHR)** system to access past medical data. It will be deployed on a **cloud-based infrastructure**, allowing access across multiple hospital branches.

The system consists of the following main components:

- **AI Diagnosis Engine**: The core machine-learning model that processes patient data and suggests possible diagnoses.
- **Web Application**: A user-friendly interface for doctors and patients to interact with the system.
- **EHR Integration Module**: A system that retrieves historical medical data to enhance diagnosis accuracy.
- **Monitoring & Model Improvement System**: Tracks AI performance over time and updates the model to ensure accuracy.



This diagram illustrates how **doctors and patients interact** with the system, with AI acting as an intermediary to provide diagnosis suggestions and recommendations. The **monitoring system** ensures that the AI model remains accurate over time.

2.2 Product Functionality

The **AI-Powered Healthcare Diagnosis System** provides the following core functionalities:

- **Patient Symptom Input**: Patients can enter their symptoms via a web or mobile interface.
- **AI-Driven Diagnosis Suggestions**: The system processes patient symptoms, medical history, and lab results to generate **probable diagnoses**.
- **Recommendation of Further Tests**: If necessary, the AI suggests additional medical tests to refine its predictions.
- **Doctor Review & Modification**: Medical professionals can **accept, modify, or reject** AI-generated diagnoses before finalizing them.
- **EHR System Integration**: The system pulls past patient records from the hospital's **Electronic Health Records (EHR)** system.
- **Real-Time Performance Monitoring**: The AI model's accuracy is tracked, and improvements are made based on real-world data.
- **Confidence Scores & Explainability**: The system provides **confidence levels** for diagnoses and highlights key factors influencing AI recommendations.
- **Security & Compliance**: Ensures compliance with **HIPAA** and other medical data regulations, protecting patient information.

2.3 Design and Implementation Constraints

The system must adhere to the following constraints:

- **AI Model Training Data**: The AI model requires large datasets of **historical patient records, medical imaging, and expert-validated diagnoses** for training.
- **Cloud-Based Deployment**: The system will be hosted on a **secure cloud infrastructure**, ensuring high availability and accessibility across multiple hospital branches.
- **Real-Time Processing**: The AI must analyze **symptoms and medical history** in real-time to provide **instant diagnostic support** to doctors.
- **Security & Compliance**: The system must **encrypt** sensitive patient data and comply with **HIPAA** and **GDPR** regulations.
- **COMET Method for Software Design**: The system will follow the **COMET (Collaborative Object Modeling and Enterprise Transformation)** method, which emphasizes component-based architecture. [Reference: COMET Design Methodology]
- **UML Modeling Language**: The **Unified Modeling Language (UML)** will be used for designing **use case diagrams, class diagrams, and system workflows**. [Reference: UML Specification]
- **Limited Medical Decision Autonomy**: The system acts as a **decision-support tool** rather than an autonomous diagnosis system—final decisions are **made by doctors**.

2.4 Assumptions and Dependencies

*The following a*ssumptions are made for the system's development and operation:

1. **Accurate Medical Data**: The AI system assumes that **historical patient data** used for training is **accurate and validated** by medical experts.
2. **Internet Connectivity**: The system depends on **continuous internet access** for cloud-based operations and remote accessibility.
3. **EHR System Compatibility**: The hospital's **EHR system** will have a standardized **API for seamless integration**.
4. **Regulatory Compliance**: The system assumes that all hospitals using the software will comply with medical **data privacy laws (HIPAA, GDPR, etc.)**.
5. **User Training for Medical Staff**: Doctors and healthcare professionals will undergo **training** to interpret AI-generated suggestions correctly.
6. **Scalability**: The system must be capable of scaling **without performance degradation** as patient data increases.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The **AI-Powered Healthcare Diagnosis System** will be accessible through:

- **Doctor & Medical Staff Interface** (Web-based application)
- **Patient Interface** (Web-based and mobile application)

Doctor & Medical Staff Interface:

- Access AI-generated **diagnostic suggestions**.
- Accept, modify, or reject AI recommendations.
- View **patient medical history** (retrieved from EHR).
- View AI **confidence scores and explanations** for predictions.
- Recommend **further medical tests** based on AI outputs.

Patient Interface:

- Enter symptoms via an **interactive questionnaire**.
- View AI-generated **preliminary diagnostic insights**.
- Book appointments or request further medical tests.

The interfaces will support **touchscreen interactions**, dropdown menus, and data input fields for structured symptom entry.

3.1.2 Hardware Interfaces

The system will interact with the following **hardware components**:

- **Hospital Servers & Cloud Storage**: Store AI model, patient records, and diagnostic logs.
- **Electronic Health Records (EHR) System**: Retrieve patient history and test results.
- **Medical Imaging Devices (e.g., X-ray, MRI, CT scanners)**: Provide image data for AI diagnosis.
- **Wearable Health Devices (optional future expansion)**: Collect patient vitals (e.g., heart rate, temperature, BP).

Each device will provide **read interfaces** to supply medical data to the AI system in **standardized formats (DICOM, HL7, FHIR, etc.)**.

3.1.3 Software Interfaces

The system will integrate with the following **external software components**:

- **Electronic Health Records (EHR) API**: Enables retrieval of patient history.
- **Cloud-Based AI Model Deployment**: Manages AI diagnosis computations.
- **Authentication & Access Control System**: Implements **role-based access** (doctors, patients, admin).

- **Security & Encryption Services**: Ensures **HIPAA-compliant data encryption**.

3.1.4 F1: Patient Data Entry & Processing

F1.1: The system shall allow patients to input symptoms using a structured form.
F1.2: The system shall retrieve the patient's past medical records from the **EHR system** (if available).

3.1.5 F2: AI Diagnosis & Recommendation

F2.1: The system shall generate probable diagnoses based on symptoms, medical history, and lab results.

F2.2: The system shall provide a confidence score for each AI-generated diagnosis.

F2.3: The system shall explain key medical factors influencing the AI's diagnosis.

F2.4: The system shall recommend additional tests (e.g., blood work, imaging) if necessary.

3.1.6 F3: Doctor Review & Decision Support

F3.1: The system shall allow doctors to accept, modify, or reject AI-generated diagnoses**.**
F3.2: The system shall provide an interface for doctors to input additional notes**.**

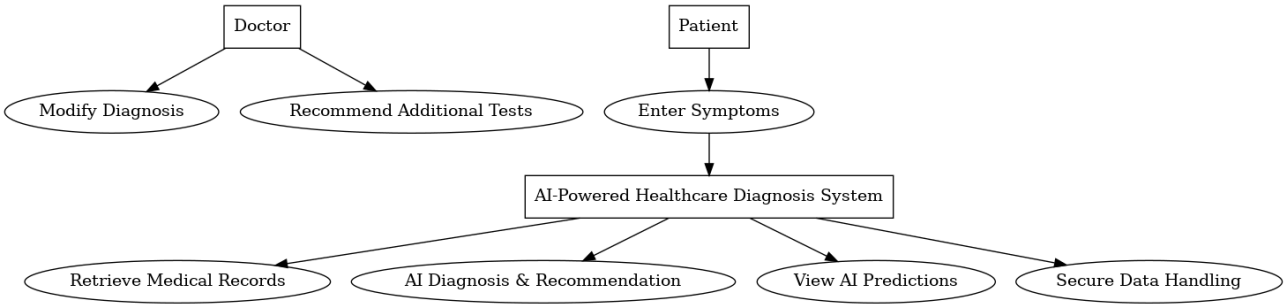
3.1.7 F4: Security & Compliance

F4.1: The system shall enforce role-based access control (RBAC) for different users.
F4.2: The system shall encrypt all stored and transmitted patient data using AES-256 encryption**.**

3.2 Use Case Model

Use Case Diagram

Below is a **high-level use case diagram** illustrating **all major interactions** within the system



3.2.1 Use Case: AI Diagnosis & Recommendation - U1

Author – Batirniyaz Muratbaev

Purpose - The AI system analyzes patient symptoms and medical history to generate probable diagnoses and recommend additional tests if needed.

Requirements Traceability – F2.1, F2.2, F2.3, F2.4

Priority - High

Preconditions -

- Patient symptoms must be entered.
- AI must have access to **EHR data** (if available).

Post conditions -

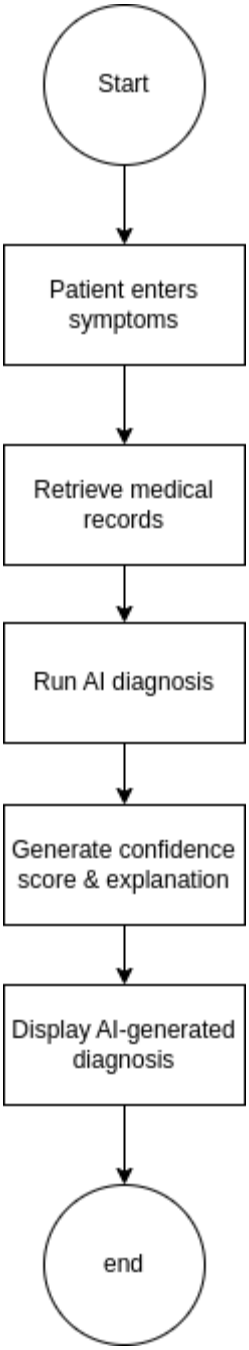
- AI generates a **diagnostic suggestion**.
- A **confidence score** is displayed.
- AI provides **recommendations for further tests**.

Actors – Patient, Doctor, AI System

Flow of Events

1. Basic Flow -
    - The patient enters symptoms via the web/mobile app.
    - The AI retrieves **past medical records (if available)**.
    - The AI runs an analysis and generates **probable diagnoses**.
    - The system displays the **AI-generated diagnosis** with a **confidence score**.
    - If necessary, the AI suggests **additional medical tests**.
2. Alternative Flow - If **no past medical records are available**, the AI relies only on **symptom input**.
3. Exceptions - If the AI model is unavailable, the system **alerts the doctor** and **falls back on manual diagnosis**.

Includes U2 (Retrieve Medical Records), U3 (Modify Diagnosis)



3.2.2 Use Case: Retrieve Medical Records (U2)

Author – Batirniyaz Muratbaev

Purpose - To retrieve a patient's past medical records from the **Electronic Health Records (EHR) system** to assist in AI diagnosis.

Requirements Traceability – F1.2

Priority - High

Preconditions -

- The patient has an existing record in the **EHR system**.
- The system has valid **authentication credentials** to access EHR data.

Post conditions -

- Patient **medical history** and **lab test results** are retrieved.
- Data is processed by the AI engine for diagnosis.

Actors – System, EHR System, Doctor

Flow of Events

4. Basic Flow -
    - The doctor or patient initiates a request to retrieve **medical records**.
    - The system sends a request to the **EHR system**.
    - The **EHR system** validates the request and retrieves **patient history**.
    - The system receives the data and stores it for AI processing.
5. Alternative Flow - If **no medical history** exists, the system **only relies on patient symptom input**.
6. Exceptions - If the **EHR system is offline**, the system **alerts the doctor** and logs the failure.

Includes None

3.2.3 Use Case: Modify Diagnosis (U5)

Author – Batirniyaz Muratbaev

Purpose - To allow doctors to **accept, modify, or reject** the AI-generated diagnosis.

Requirements Traceability – F3.1, F3.2

Priority - High

Preconditions -

- The AI system has provided a **diagnostic suggestion**.
- The doctor is logged into the system.

Post conditions -

- The doctor's decision is **recorded and stored**.
- If modified, the final diagnosis is updated in the system.

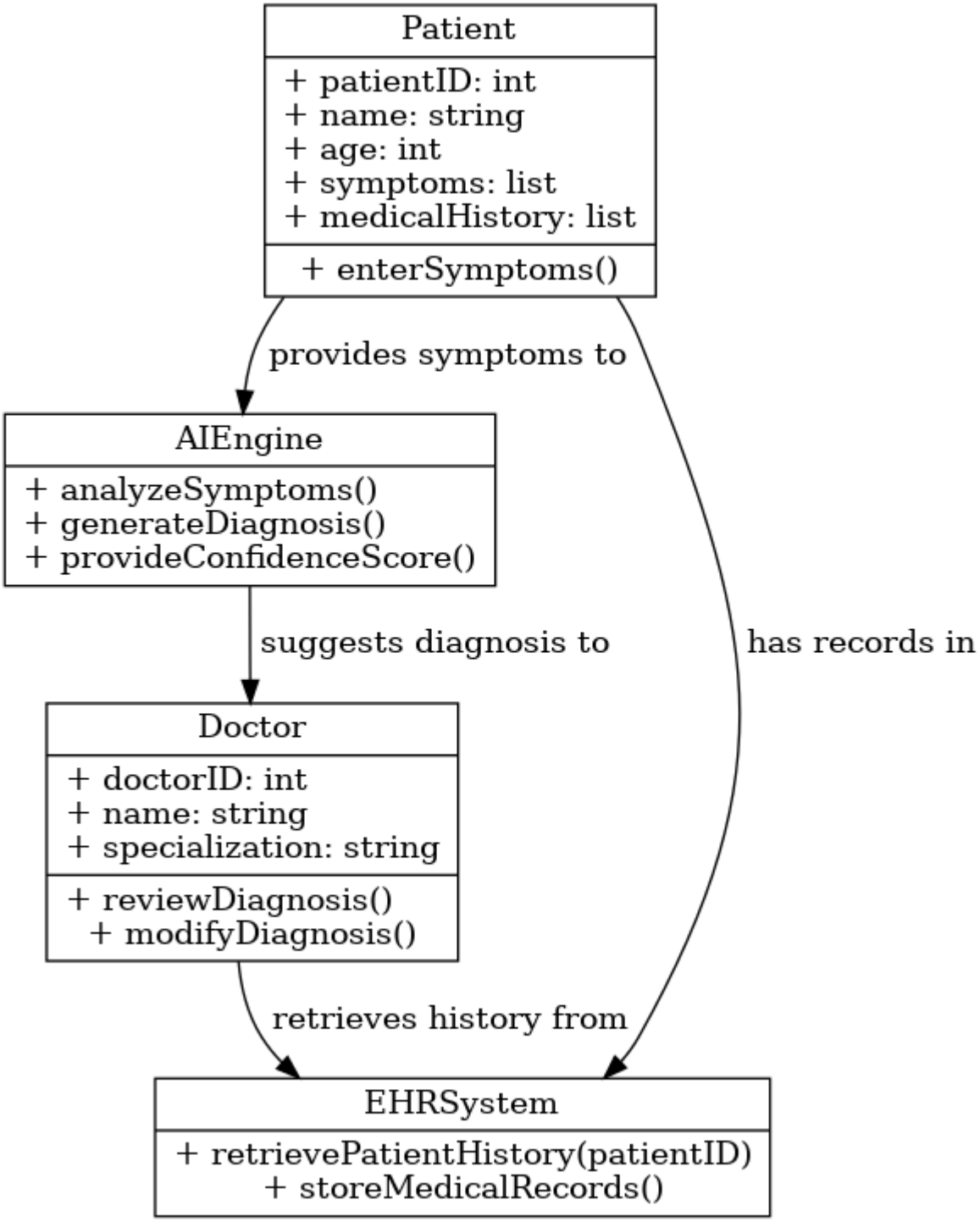
Actors – Doctor, System

Flow of Events

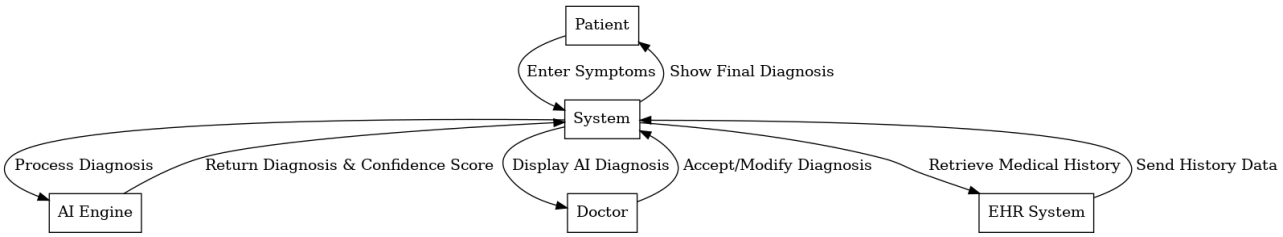
7. Basic Flow -
    - The doctor logs in and views the **AI-generated diagnosis**.
    - The doctor can **accept, modify, or reject** the suggestion.
    - If modified, the doctor updates the **final diagnosis**.
    - The system stores the final decision and updates the patient's **medical records**.
8. Alternative Flow - If the doctor **needs additional tests**, they can **request further evaluations** before finalizing the diagnosis.
9. Exceptions - If the AI system fails, doctors rely on **manual diagnosis methods**.

Includes None

3.3 Class Model



3.4 Behavioural Model



4. Other Non-functional Requirements

4.1 Performance Requirements

P1. Real-Time Response:
The system shall generate a diagnostic suggestion, along with a confidence score and key explanatory factors, within **5 seconds** after the patient submits their symptoms. This ensures doctors receive timely assistance, which is critical in emergency situations.
P2. Scalability and Throughput:
The system must support concurrent access by multiple hospital branches. It shall be designed to handle up to **100 simultaneous user requests** (including both patients and doctors) without significant degradation in performance. In a cloud-based environment, the system should auto-scale based on demand.
P3. Data Retrieval Speed:
The integration with the Electronic Health Records (EHR) system shall ensure that patient medical histories are retrieved and made available for AI processing within **3 seconds** under normal operating conditions.

4.2 Safety and Security Requirements

S1. Data Encryption and Privacy:
All patient data, both in transit and at rest, must be encrypted using industry-standard methods (e.g., **AES-256**). This is to comply with **HIPAA**, **GDPR**, and other relevant privacy regulations. The system should implement secure socket layer (SSL) encryption for all mobile and web connections.
S2. Access Control and Authentication:
The system shall enforce **role-based access control (RBAC)**, ensuring that only authorized users (e.g., doctors, nurses, and patients) can access specific parts of the application. Multi-factor authentication (MFA) must be required for doctor and admin accounts to further strengthen security.
S3. Audit and Monitoring:
All access and data modifications shall be logged for auditing purposes. The system should have mechanisms to detect, report, and mitigate any unauthorized access attempts. Alerts must be generated in case of security breaches.

S4. Mobile Connection Security:
Since patients may use mobile devices to interact with the system, the mobile app shall incorporate secure communication protocols, regular security updates, and a strict session management policy to prevent unauthorized access.

4.3 Software Quality Attributes

4.3.1 Reliability

- **Requirement**: The system shall be available 99.9% of the time during operational hours.
- **Implementation**:
  - Use a **redundant, cloud-based infrastructure** with failover mechanisms.
  - Regularly conduct **stress and load testing** to identify potential points of failure.
  - Implement real-time monitoring and automated recovery processes for system faults.
- **Verification**: Monitor uptime using system logs and third-party services, ensuring that outages are documented and addressed promptly.

4.3.2 Maintainability and Adaptability

- **Requirement**: The software must be designed for easy maintenance and future enhancements, including the ability to integrate additional sensors or data sources (e.g., new types of medical imaging devices).
- **Implementation**:
  - Follow the **COMET method** and use **UML** diagrams to maintain clear, modular architecture.
  - Adopt a **service-oriented architecture (SOA)** to ensure components are loosely coupled and can be updated or replaced without impacting the overall system.
  - Document code and system design thoroughly, and enforce coding standards and best practices.
- **Verification**:
  - Regular code reviews and technical audits will be conducted to ensure compliance with design guidelines.
  - Maintenance tasks and updates will be tracked via a version control system and a dedicated issue tracking system.

5. Other Requirements

Database Requirements:
The system shall utilize a robust, relational database management system (e.g., PostgreSQL or MySQL) to store patient data, diagnostic logs, and system configurations. The database must support secure data storage, backup, and recovery mechanisms. It should be designed for scalability and high availability to accommodate growing data volumes and concurrent access by multiple hospital branches.

Internationalization and Localization:
The system shall support multiple languages. The initial release will be in English, but the design must allow for future expansion to support additional languages (e.g., Spanish, French). Date, time, and measurement units shall be adaptable to regional settings, ensuring a consistent user experience for international users.

Legal and Regulatory Requirements:
The system must comply with all relevant legal and regulatory standards including HIPAA, GDPR, and other local healthcare regulations. This includes requirements for data privacy, patient consent, and secure data transmission. Regular audits and compliance checks must be incorporated into the system's maintenance procedures.

Reusability and Modular Design:
Critical components of the system, such as the **EHR integration module** and the **AI Diagnosis Engine**, shall be designed with reusability in mind. These modules should be documented and built using modular principles to facilitate reuse in future projects or related systems with minimal modifications.

Logging and Audit Requirements:
The system shall maintain comprehensive logs of all user interactions and system events. Audit trails must be maintained to support compliance with legal and regulatory standards, enabling the tracking of data access and modifications for security reviews.

Backup, Recovery, and Disaster Recovery:
Regular data backups must be performed to prevent data loss. The system shall include a disaster recovery plan ensuring that critical services can be restored within a predefined Recovery Time Objective (RTO) in the event of system failure or data breach.

Training and Documentation:
Comprehensive training materials and user documentation must be provided for both medical staff and system administrators. This will ensure proper usage of the system and facilitate quick troubleshooting and maintenance.

Interoperability and Future Expansion:
The system should be designed to easily integrate with future healthcare applications, medical devices, and external data sources. This includes supporting standard communication protocols and data formats (e.g., HL7, FHIR) to ensure seamless interoperability within the healthcare ecosystem.

Appendix A – Data Dictionary

<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to produce this document>