Software Requirements Specification (SRS)

Table of Contents

List of Figures

- Chapter 1 :

- Chapter 2 :

1.0 Introduction

Healthcare is one of the most critical areas of human life, but many diseases remain difficult to diagnose at an early stage. Conditions such as **Heart Disease, Parkinson's Disease, and Diabetes** are life-threatening if not detected and managed early. Each of these diseases has distinct risk factors, symptoms, and medical challenges, making diagnosis a time-consuming and error-prone process if done manually.

This document describes a smart, integrated software system designed to predict the likelihood of these diseases using **machine learning models** trained on patient health data. The system allows patients and healthcare providers to input relevant health parameters, processes them using advanced algorithms, and generates prediction results. The goal is not just to classify risk, but to **assist doctors in decision-making, improve preventive care, and empower patients** to take timely action. This system transforms the challenge of multi-disease diagnosis into an opportunity to improve healthcare efficiency and outcomes.

1.1 Purpose

The primary purpose of this project is to develop a **robust, intelligent, and user-friendly disease prediction system** that can provide early diagnosis support for Heart Disease, Parkinson's Disease, and Diabetes.

The system is intended to:

- **Enhance Early Detection**: Identify high-risk individuals before symptoms worsen, minimizing long-term complications.
- **Support Medical Professionals**: Provide doctors with predictive insights that can complement clinical examinations and laboratory tests.
- **Empower Patients**: Allow individuals to self-assess their health status based on standard clinical inputs and seek medical advice sooner.
- **Improve Healthcare Efficiency**: Reduce diagnostic delays, minimize misdiagnosis, and allow hospitals to manage large patient datasets effectively.
- **Provide Actionable Insights**: Present prediction results with visual reports, confidence scores, and disease risk categories to aid decision-making.

1.2 Scope of Project

The scope of this project encompasses the **design, development, and deployment** of a web-based predictive system for multiple diseases. The system focuses on three major chronic conditions: Heart Disease, Parkinson's Disease, and Diabetes.

Key functionalities and boundaries:

- **Diseases Covered**: Heart Disease, Parkinson's Disease, Diabetes.
- **Data Sources**: Medical datasets from repositories such as UCI Machine Learning Repository, Kaggle medical datasets, and hospital-provided anonymized records.
- **Core Technologies**:

    - **Machine Learning Models**: Logistic Regression, Random Forest, Support Vector Machines, Gradient Boosting.
    - **Data Processing**: Standardization, normalization, and handling missing values.
    - **Visualization**: Interactive dashboards with prediction graphs and downloadable reports.
- **System Output**: Disease prediction results expressed as a **probability score (e.g., 72% likelihood of diabetes)** and a categorical risk level (Low, Medium, High).
- **Out of Scope**:
    - The system does not provide medical treatment or prescribe medication.
    - It does not replace clinical laboratory testing or physical examination.
    - It is designed as a **decision-support tool** only.

1.3 Glossary

Term	Definition
Accuracy	Ratio of correctly predicted outcomes to the total cases.
Precision	Proportion of true positives among all positive predictions.
Recall	Ability of the model to correctly identify true positives.
ML (Machine Learning)	Algorithms that enable systems to learn from data and make predictions.
Dataset	A collection of patient health records used for model training and testing.
SVM	Support Vector Machine – a classification algorithm.
Random Forest	An ensemble learning method using multiple decision trees for predictions.

1.4 References

1. UCI Machine Learning Repository – Heart Disease, Parkinson's, and Diabetes Datasets.
2. World Health Organization (WHO) – Global Reports on Non-Communicable Diseases.
3. Research Paper: "Machine Learning in Healthcare: Review and Applications" – IEEE, 2023.
4. Kaggle Dataset: Parkinson's Disease Voice Measurements.
5. Journal of Diabetes Research – "Predictive Models for Diabetes Using Patient Records", 2022.
6. Journal of Medical Systems – "Artificial Intelligence for Early Detection of Heart Disease", 2021.

1.5 Overview of Document

This document is structured to provide a **comprehensive and detailed specification** of the software requirements. It serves as a foundational guide for developers, testers, healthcare professionals, and project managers throughout the development lifecycle.

- **Section 2 (Overall Description)** provides a high-level perspective, outlining the system's environment, intended users, functional requirements, and non-functional requirements.
- **Section 3 (Requirements Specification)** details the specific functionalities, including user interfaces, external data input, machine learning model outputs, and quality benchmarks such as accuracy, security, and performance.

2.0 Overall Description

This section provides a high-level overview of the **Multiple Disease Prediction System**, outlining its operational context, key functionalities, intended users, and constraints. The system is designed as a specialized, data-driven healthcare solution for predicting the likelihood of **Heart Disease, Parkinson's Disease, and Diabetes** based on patient medical data. It combines machine learning models, a secure database, and an interactive web interface to support both patients and healthcare professionals in early detection and decision-making.

2.1 System Environment

The prediction system is designed to operate within a modern **web-based and healthcare IT ecosystem**. It functions as a cloud-hosted or locally deployed diagnostic support tool for hospitals, clinics, and individual users.

- **Operating System**: Platform-independent, deployable on Windows, Linux, or cloud-based environments.
- **Core Technologies**: Backend logic developed in Python, using machine learning libraries (e.g., Scikit-learn, TensorFlow, PyTorch). Web framework such as Flask or Django ensures smooth communication between frontend and backend.
- **Data Environment**: The system ingests structured patient datasets (e.g., UCI ML Repository for training, hospital databases for real-time predictions). Outputs include disease prediction results with probability scores and reports.
- **User Interface**: A web-based application (React/HTML/CSS frontend) provides forms for inputting patient details and dashboards for displaying results. Accessible from any standard browser across devices (desktop, tablet, mobile).
- **Hardware**: Runs on standard servers, laptops, or cloud VMs. Hardware such as clinical diagnostic devices (e.g., glucose monitors, ECG) may provide patient input data, but the system itself does not include medical hardware.

2.2 Functional Requirements Specification

This section details the primary functions of the system, broken into specific use cases.

2.2.1 Use Case: Heart Disease Prediction

- **Description**: Predicts the risk of heart disease based on inputs such as age, cholesterol level, chest pain type, resting blood pressure, ECG results, and exercise-induced angina.
- **Process**:
    1. User enters patient medical parameters via the input form.
    2. Backend ML model processes the inputs.
    3. The system outputs a **risk probability score** and classification (Low/Medium/High).

2.2.2 Use Case: Parkinson's Disease Prediction

- **Description**: Detects early Parkinson's disease likelihood using biomedical voice features (jitter, shimmer, frequency, etc.) or related motor-symptom data.
- **Process**:
    1. User provides patient features (uploaded dataset or entered manually).
    2. The ML model analyzes the data to detect Parkinson's indicators.
    3. The system displays a prediction with a percentage likelihood.

2.2.3 Use Case: Diabetes Prediction

- **Description**: Predicts diabetes risk based on glucose levels, BMI, insulin concentration, blood pressure, and family medical history.
- **Process**:
    1. User inputs test results into the system.
    2. The model processes inputs using trained diabetes datasets.
    3. The output is a categorical classification (Positive/Negative) along with a confidence score.

2.2.4 Use Case: Multi-Disease Dashboard Visualization

- **Description**: Provides a unified interface to view predictions for all three diseases.
- **Process**:
    1. User submits patient data (covering all required parameters).
    2. The system runs each prediction model in parallel.
    3. The dashboard displays results for Heart, Parkinson's, and Diabetes side-by-side with charts for risk levels.

2.3 User Characteristics

The system is designed for a diverse range of users with different technical and medical expertise:

- **Patients**: Non-technical users seeking self-assessment of their health status. Interface must be simple and guide them clearly through input steps.
- **Doctors & Healthcare Professionals**: Use the system as a **decision-support tool**. They require accurate predictions, probability scores, and downloadable reports for clinical use.
- **System Administrators**: Manage datasets, update models, and ensure secure handling of patient data. They need access to technical dashboards and system logs.

2.4 Non-Functional Requirements

- **Performance**:
  - Predictions must be generated within **5 seconds** of data submission.
  - Dashboard visualization should load instantly for smooth interaction.
- **Accuracy**:
  - Each disease model should achieve a minimum **80% accuracy** on test datasets.
  - Models should maintain **balanced precision and recall** to reduce false positives/negatives.
- **Reliability**:
  - The system must produce **consistent, repeatable results** for the same input data.
  - Should handle large patient datasets without crashing.
- **Usability**:
  - Web UI must be **intuitive and user-friendly**, requiring minimal training.
  - Support for multiple devices (desktop/mobile).
- **Security**:
  - Patient data must be encrypted both at rest and in transit (HIPAA compliance).
  - User authentication required for doctor and admin access.

3.2 Functional Requirements

These requirements define the specific behaviors and functions the **Multiple Disease Prediction System** must perform.

3.2.1 Collect Patient Health Inputs

- **Description**: The user must be able to provide patient data such as age, blood pressure, glucose level, BMI, cholesterol, ECG results, or biomedical voice features for Parkinson's.
- **Process**:
    1. The user logs into the system and selects the disease(s) to test.
    2. The user fills in health parameters in structured input forms.
    3. The system validates the entered data (range checks, missing values).
    4. The inputs are sent to the backend prediction engine.

3.2.2 Process Medical Dataset

- **Description**: The system must preprocess the input data to make it machine-readable for the ML models.
- **Process**:
    1. Normalize/standardize values such as glucose, cholesterol, and voice features.
    2. Handle missing values using imputation or default ranges.
    3. Convert categorical values (e.g., chest pain type, family history) into numerical codes.
    4. Store processed inputs temporarily in memory or database before prediction.

3.2.3 Generate Disease Predictions

- **Description**: The system must compute the likelihood of each targeted disease using trained ML models.
- **Process**:
    1. The backend invokes the respective ML model (Heart, Parkinson's, Diabetes).
    2. The model computes a probability score (e.g., 0.76 → 76% risk).
    3. Based on thresholds, the system classifies the result into Low/Medium/High Risk (or Positive/Negative for Diabetes).
    4. Predictions are formatted and sent back to the frontend.

3.2.4 Optimize Results for Interpretability

- **Description**: The prediction outputs must be human-friendly and medically interpretable.
- **Process**:
    1. Raw probability outputs are converted into categorized risk levels.

2. Confidence scores and supporting metrics (precision, recall) are displayed.
3. The dashboard highlights key factors influencing the prediction (e.g., high cholesterol strongly contributed to Heart Disease risk).
4. This ensures doctors and patients can **understand the "why" behind the result**.

3.2.5 Visualize Prediction Dashboard

- **Description**: The system must provide clear and interactive visualization of predictions.
- **Process**:
    1. The web UI displays a consolidated dashboard with disease results.
    2. Each disease is shown with probability, risk level, and charts (bar/pie graphs).
    3. The system highlights "High Risk" results in red for immediate attention.
    4. Users can interact with the dashboard to expand results, download reports, or compare past predictions.

3.2.6 Configure Model Parameters

- **Description**: The system should allow administrators to fine-tune model parameters and thresholds.
- **Process**:
    1. A configuration panel (JSON/YAML file or admin UI) is available.
    2. Configurable parameters include:
        - **Risk Thresholds**: Adjust cutoffs for Low/Medium/High risk.
        - **Model Selection**: Choose between Logistic Regression, Random Forest, SVM, etc.
        - **Data Update Frequency**: Define how often models are retrained with new data.
    3. Updated configurations are applied without affecting stored patient records.

3.3 Detailed Non-Functional Requirements

These requirements define the quality attributes and operational constraints of the disease prediction system, ensuring it is not only functional but also efficient, reliable, and secure.

3.3.1 Performance

- **Description**: The system must be highly responsive to support real-time healthcare needs.

- **Requirements**:
    - **Prediction Latency**: Each disease prediction must be completed in under **5 seconds** on standard hardware.
    - **Simultaneous Users**: The system must support at least **100 concurrent users** without performance degradation.
    - **UI Responsiveness**: The dashboard must load within **2 seconds**, and interactions (charts/filters) should remain smooth.

3.3.2 Accuracy and Precision

- **Description**: The system's effectiveness is directly dependent on prediction accuracy.
- **Requirements**:
    - Heart Disease Model: Accuracy ≥ **82%**, Precision ≥ **0.80**, Recall ≥ **0.78**.
    - Parkinson's Model: Accuracy ≥ **85%**, Precision ≥ **0.82**, Recall ≥ **0.80**.
    - Diabetes Model: Accuracy ≥ **80%**, Precision ≥ **0.78**, Recall ≥ **0.77**.
    - Predictions must align with **clinical significance** and avoid bias.

3.3.3 Robustness and Reliability

- **Description**: The system must remain stable, consistent, and fault-tolerant.
- **Requirements**:
    - **Error Handling**: If inputs are incomplete or invalid, the system must display an error message without crashing.
    - **Data Integrity**: Patient inputs and predictions must be stored securely and consistently across sessions.
    - **System Recovery**: In case of failure, the system must restore the last stable state without losing records.

3.3.4 Security & Privacy

- **Description**: As the system handles sensitive patient health data, strong privacy and security measures are required.
- **Requirements**:
    - **Data Transmission**: All communication must be encrypted using **HTTPS/TLS**.
    - **Data Storage**: Patient data must be encrypted at rest (AES-256 or equivalent).
    - **Authentication & Access Control**: Patients, doctors, and admins must have separate access roles.

- **Regulatory Compliance**: System must comply with **HIPAA / GDPR** guidelines for healthcare data privacy.

List of Figures

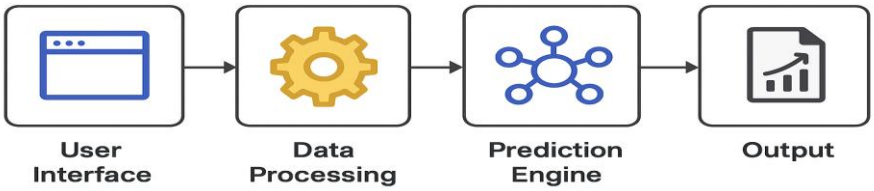
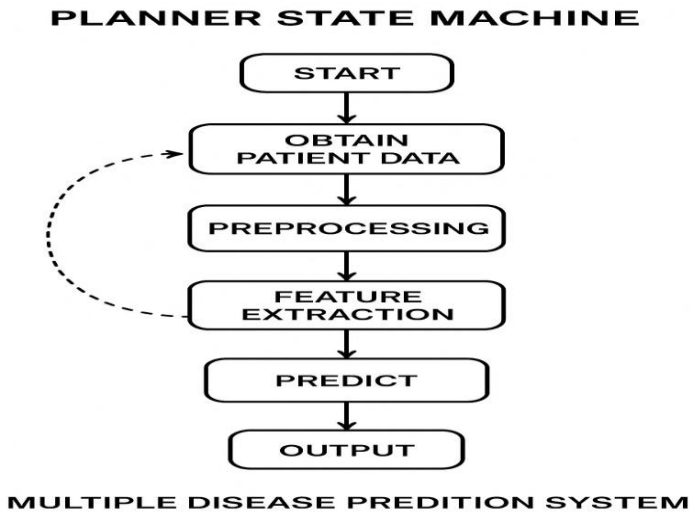


Figure 1 — System Architecture for Disease Prediction



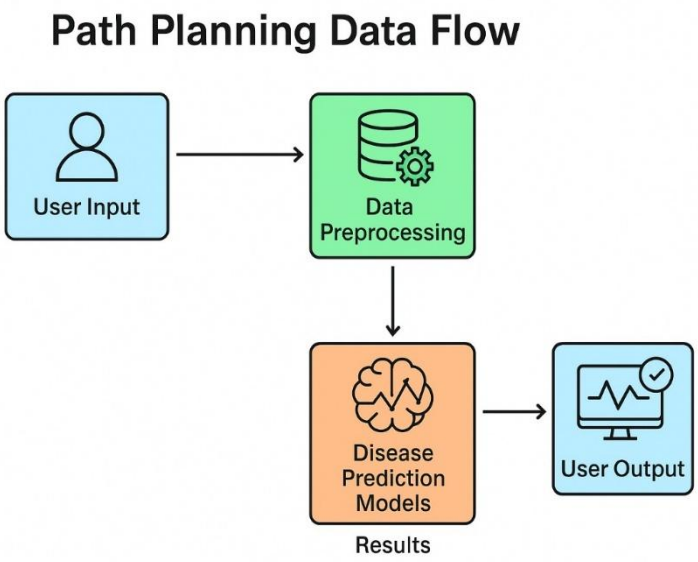


Figure 1.3 : Planner State Machine

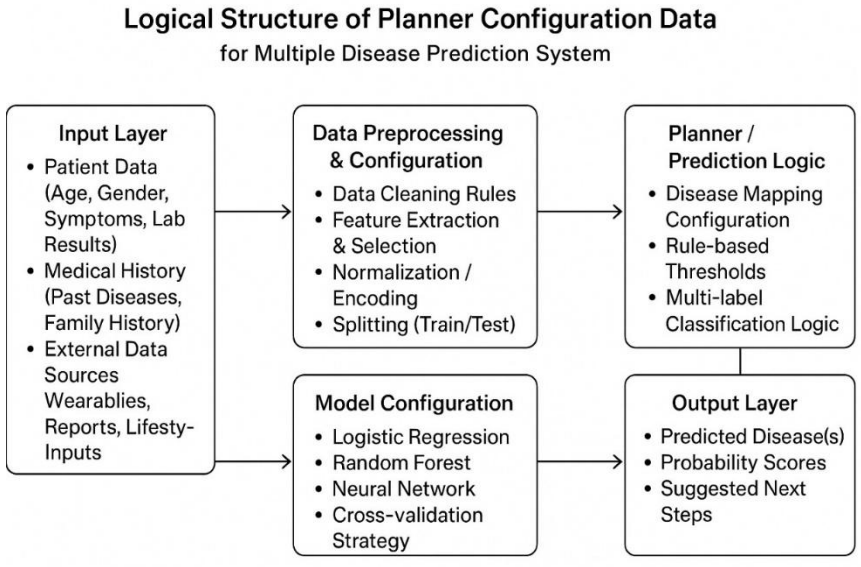


Figure 1.4 : Logical Structure of Planner Configuration Data

CHAPTER 2

➢ Architectural diagram & high level design

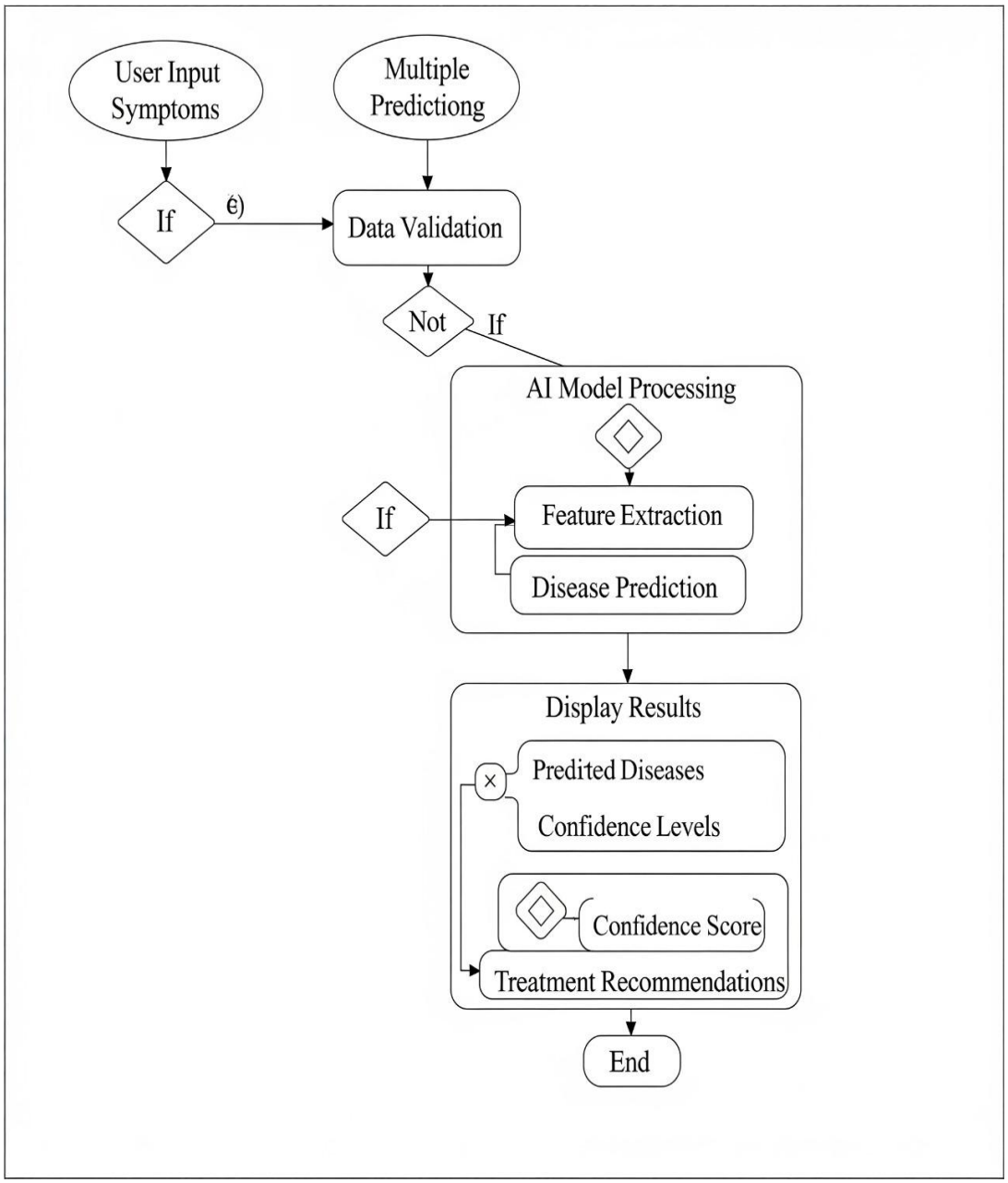


Fig 2.1 : Activity Diagram

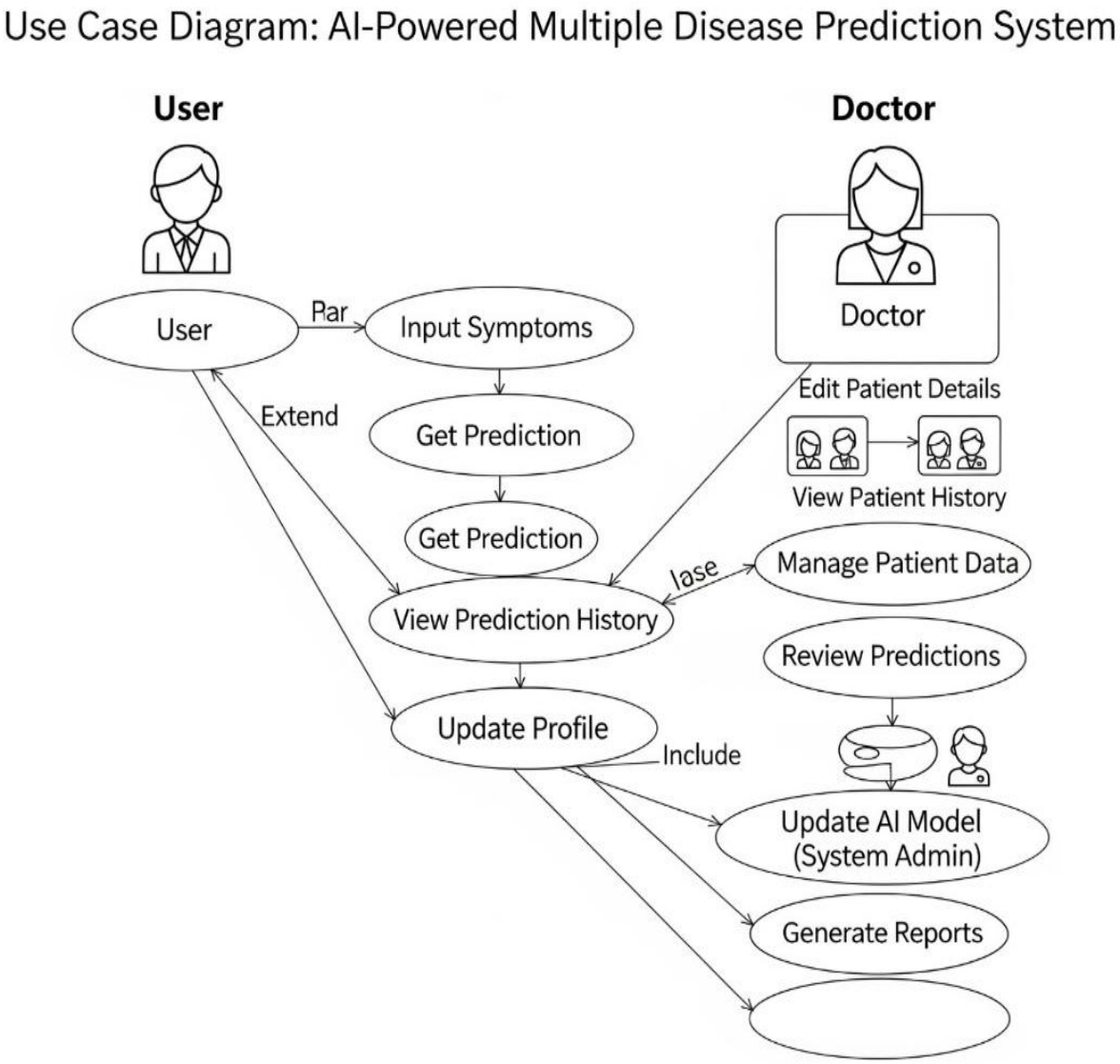


Fig 2.2 : Use Case Diagram

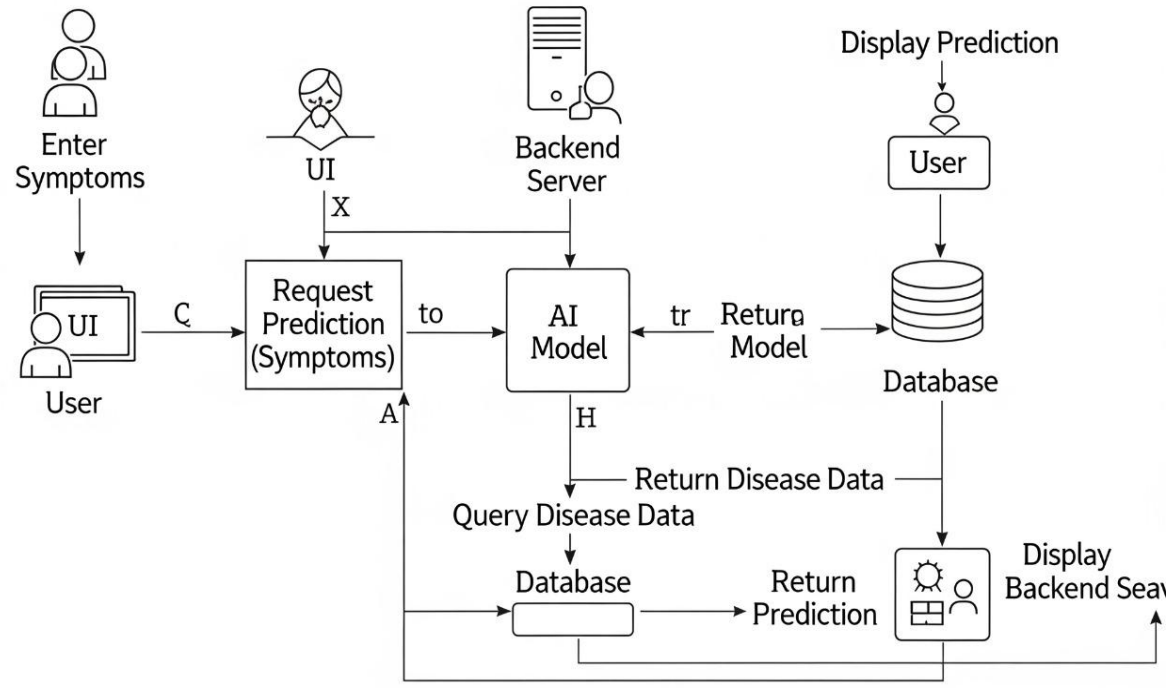


Fig 2.3 : Sequence Diagram

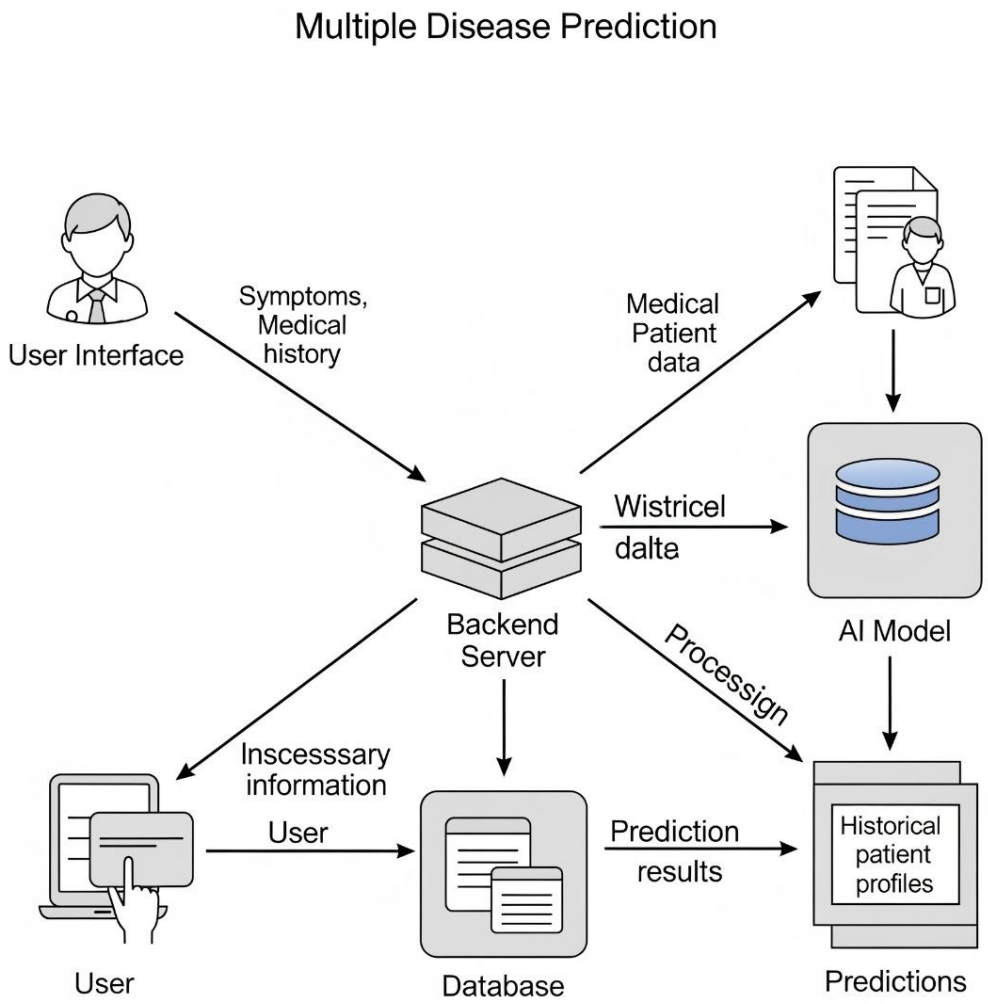


Fig 2.4 : Collaboration Diagram