Software Requirements Specification (SRS)

Version 1.0

**Software Requirements Specification**

**For Usecase:02**

**Member Risk Stratification and Care Management**

**Project Title:**

**Hope Care: A Predictive Healthcare ROI Platform**

**Prepared by:**

* Pradeep Kumar A
* Suriya SK
* Kishore R
* Balamohan M
* Jayvanti S

**Organization:** Sri Manakula Vinayagar Engineering College

**Date Created**: 06/09/2025

**Table of Contents**

1. Introduction ......................................................................... 1

1.1 Purpose ............................................................................ 1

1.2 Document Conventions ..................................................... 1

1.3 Intended Audience and Reading Suggestions ...................... 2

1.4 Project Scope .................................................................. 2

1.5 References ....................................................................... 3

2. Overall Description ............................................................... 4

2.1 Product Perspective ........................................................ 4

2.2 Product Functions ............................................................ 5

2.3 User Classes and Characteristics ..................................... 5

2.4 Operating Environment .................................................... 6

2.5 Design and Implementation Constraints ............................ 6

2.6 Assumptions and Dependencies ....................................... 7

3. System Features ..................................................................... 8

3.1 Risk Stratification Module ............................................... 8

3.2 ROI Prediction Module .................................................... 9

3.3 Data Upload & Processing Module ................................... 9

3.4 Visualization Module ...................................................... 10

4. External Interface Requirements .......................................... 11

4.1 User Interfaces ............................................................... 11

4.2 Hardware Interfaces ....................................................... 11

4.3 Software Interfaces ........................................................ 12

4.4 Communications Interfaces ............................................ 12

5. Non-Functional Requirements ............................................ 13

5.1 Performance .................................................................. 13

5.2 Security ......................................................................... 13

5.3 Reliability & Availability ................................................. 14

5.4 Maintainability .............................................................. 14

5.5 Portability ...................................................................... 15

6. Other Requirements ............................................................. 16

## **1. Introduction**

### **1.1 Purpose**

The purpose of this Software Requirements Specification (SRS) is to define the functional, non-functional, and system-level requirements for the **Hope Care: A Predictive Healthcare ROI Platform**. This web-based decision-support system integrates **patient health risk Stratification** with **Return on Investment (ROI) forecasting** to quantify the financial benefits of proactive healthcare interventions.

This document serves as a guide for:

Developers (to implement system functionality),

* Testers (to validate requirements),
* Healthcare stakeholders (to understand expected outcomes),
* Researchers/students (to explore predictive analytics in healthcare).

### **1.2 Document Conventions**

* **Frontend Technologies**: React, Vite, Tailwind CSS
* **Backend Technologies**: Python, FastAPI, Uvicorn
* **Machine Learning Frameworks**: Scikit-learn, XGBoost, SHAP
* **Data Formats**: CSV (input), JSON (API responses)
* **Style Guidelines**: IEEE 830 SRS format

### **1.3 Intended Audience and Reading Suggestions**

* **Healthcare Providers/Insurers:** Review sections on Project Scope, System Features, and Visualization.
* **Developers:** Focus on System Features, External Interfaces, and Design Constraints.
* **Testers:** Focus on Functional Requirements, NFRs, and System Features.
* **Researchers/Students:** Review the entire document for predictive analytics use case insights.

### **1.4 Project Scope**

The system will:

* Ingest structured patient data (demographics, clinical variables).
* Predict patient risks for chronic conditions using a classifier chain ML model.
* Forecast proactive vs. reactive treatment costs with a Gradient Boosting Regressor ROI model.
* Present results in a web interface with tables, charts, and summary cards.
* Highlight financial savings to support decision-making in preventative care.

**Future Enhancements:**

* Integration of real-world anonymized patient and financial datasets.
* Advanced feature engineering for more precise predictions.
* Integration of secure databases (PostgreSQL/MySQL) for large-scale datasets.

### **1.5 References**

* IEEE 830 SRS Standard
* Scikit-learn Documentation
* XGBoost Documentation
* SHAP Documentation
* FastAPI Documentation
* React Documentation

## **2. Overall Description**

## **2.1 Product Perspective**

The system is a **modular full-stack web application** built as a **monorepo**:

* **Frontend (React + Vite):** Provides CSV upload, API integration, and visualization.
* **Backend (FastAPI + Python):** Hosts ML models, performs predictions, and exposes APIs.
* **ML Pipeline:** Two-stage prediction (Risk Stratification → ROI Calculation).

### **2.2 Product Functions**

* Upload CSV file containing patient data.
* Parse and validate patient records.
* Predict health risks with XGBoost classifier chain.
* Forecast ROI via proactive vs. reactive cost predictions.
* Display results in structured tables and interactive charts.
* Enable multiple uploads and comparisons across patients.

### **2.3 User Classes and Characteristics**

* **Healthcare Providers (Doctors, Hospitals):** Require risk insights to design proactive care.
* **Insurance Companies:** Require financial justification for policy design and coverage.
* **Administrators:** Manage deployment, datasets, and ensure compliance.
* **Researchers/Students:** Use platform as reference for predictive healthcare analytics.

### **2.4 Operating Environment**

* **Frontend:** Chrome, Firefox, Edge (latest versions).
* **Backend:** Python 3.10+, FastAPI, Uvicorn server.
* **Supported OS:** Linux, Windows, macOS.
* **Database:** (Future scope) PostgreSQL/MySQL.

### **2.5 Design and Implementation Constraints**

* ROI model trained on **synthetic dataset** (no real patient cost data)
* Relies on **Python ML libraries** (scikit-learn, XGBoost, SHAP).
* Requires **modern browsers** for frontend rendering.
* Must comply with **data privacy laws (HIPAA/GDPR)** in future when real data is used.

### **2.6 Assumptions and Dependencies**

* Assumes CSV input data is clean and properly formatted.
* Requires availability of Python environment & Node.js runtime.
* Dependent on REST API communication between frontend and backend.

## **3. System Features**

### **3.1 Risk Stratification Module**

* **Description:** Predicts probability of chronic conditions.
* **Input:** Patient demographics, clinical features.
* **Output:** Risk scores, predicted conditions, contributing factors.
* **Model:** Classifier Chain with XGBoost + SHAP for explainability.

### **3.2 ROI Prediction Module**

* **Description:** Forecasts treatment costs.
* **Input:** Patient’s age & risk condition.
* **Output:** Predicted proactive cost, reactive cost, potential savings.
* **Model:** Gradient Boosting Regressor trained on synthetic dataset.

### **3.3 Data Upload & Processing Module**

* Upload CSV → Validate → Convert first row to JSON → API call.
* Validation: Ensures missing/invalid values are flagged.

### **3.4 Visualization Module**

* **Components:** PatientTable, RoiTable.
* **Features:**
  + Risk tiers with color codes.
  + Cost comparisons (reactive vs proactive).
  + Potential savings as summary cards.

## **4. External Interface Requirements**

### **4.1 User Interfaces**

* CSV upload button.
* /analysis page with real-time visualization.
* Error handling messages for invalid input.

### **4.2 Hardware Interfaces**

* Any device with modern browser.
* Backend: minimum **8GB RAM, Python 3.10 environment**.

### **4.3 Software Interfaces**

* Frontend ↔ Backend via Fast API.
* Data exchanged in **JSON format**.

### **4.4 Communications Interfaces**

* Protocol: **HTTP/HTTPS**.
* API Endpoint: **/predict/**.

## **5. Non-Functional Requirements**

### **5.1 Performance**

* Handle CSV uploads of up to **1000 patient records**.
* Response latency **< 2 seconds per patient**.

### **5.2 Security**

* Encrypted API communication (HTTPS).
* Authentication & role-based access (future scope).
* No PII (Personally Identifiable Information) stored.

### **5.3 Reliability & Availability**

* 95% uptime in production.
* Graceful error recovery from API failures.

### **5.4 Maintainability**

* Monorepo with modular structure.
* Clear separation of frontend and backend.
* Well-documented code and API contracts.

### **5.5 Portability**

* Cross-platform compatibility (Linux/Windows/macOS).
* Deployable on **AWS**

## **6. Other Requirements**

* Must comply with **HIPAA/GDPR** when handling real-world datasets.
* Integration with monitoring tools (Prometheus/Grafana) for production readiness (future scope

1. Introduction1.1 Purpose

This Software Requirements Specification (SRS) meticulously outlines the comprehensive functional, non-functional, and intricate system-level requirements for the **Hope Care: A Predictive Healthcare ROI Platform**. This sophisticated web -based decision-support system is engineered to seamlessly integrate **patient health risk Stratification** with robust **Return on Investment (ROI) forecasting**, thereby empowering healthcare stakeholders to quantitatively assess and comprehend the profound financial benefits derived from proactive healthcare interventions.

This meticulously crafted document serves as an indispensable compass, guiding various key stakeholders:

* **Developers:** This SRS provides the architectural blueprint and precise specifications necessary for the meticulous implementation of every system functionality, ensuring alignment with design objectives.
* **Testers:** It offers a definitive reference point for the rigorous validation of all specified requirements, enabling thorough testing and quality assurance procedures.
* **Healthcare Stakeholders:** This document elucidates the anticipated outcomes and tangible benefits of the platform, fostering a clear understanding of its value proposition for improved patient care and financial efficiency.
* **Researchers/Students:** It serves as a valuable academic resource, offering profound insights into the practical application and exploration of advanced predictive analytics within the dynamic landscape of healthcare.

1.2 Document Conventions

To ensure clarity, consistency, and unambiguous interpretation throughout this document, the following conventions are adopted:

* **Frontend Technologies**: The user interface and client-side logic will be developed using **React**, a declarative, component-based JavaScript library, bundled efficiently with **Vite**, a next-generation frontend tooling that provides an extremely fast development experience. Styling will be managed using **Tailwind CSS**, a utility-first CSS framework that enables rapid UI development.
* **Backend Technologies**: The server-side logic, API endpoints, and machine learning model serving will be implemented in **Python**, a versatile and widely-used programming language. The web framework of choice is **FastAPI**, renowned for its high performance and automatic interactive API documentation. The application will be served by **Uvicorn**, an ASGI server.
* **Machine Learning Frameworks**: For the core predictive analytics, the platform leverages industry-standard machine learning libraries: **Scikit-learn** for various supervised and unsupervised learning algorithms, **XGBoost** for highly efficient and scalable gradient boosting, and **SHAP (SHapley Additive exPlanations)** for model interpretability and explainability.
* **Data Formats**: Input data for analysis will primarily be accepted in **CSV (Comma Separated Values)** format, a widely accessible and interoperable standard. All API responses and internal data exchanges will conform to the **JSON (JavaScript Object Notation)** format, ensuring lightweight and readable data transmission.
* **Style Guidelines**: This SRS adheres strictly to the **IEEE 830 SRS format**, a globally recognized standard for software requirements specifications, ensuring comprehensive coverage, clarity, and maintainability of the document.

1.3 Intended Audience and Reading Suggestions

Recognizing the diverse readership of this SRS, specific reading suggestions are provided to optimize comprehension and relevance for each audience segment:

* **Healthcare Providers/Insurers:** To grasp the strategic value and practical applications of the platform, it is recommended to intensely review sections such as **Project Scope** (for understanding what the system will achieve), **System Features** (for detailed functionalities), and **Visualization** (for insights into data presentation and decision support).
* **Developers:** For comprehensive technical understanding and implementation guidance, developers should concentrate their efforts on **System Features** (to understand core functionalities), **External Interfaces** (for API interactions and integration points), and **Design Constraints** (for technical limitations and architectural considerations).
* **Testers:** To formulate effective test plans and validate system behavior, testers should meticulously examine **Functional Requirements** (detailing what the system must do), **Non-Functional Requirements (NFRs)** (addressing quality attributes like performance and security), and **System Features** (for comprehensive test case development).
* **Researchers/Students:** For an holistic understanding of the predictive analytics use case in healthcare, encompassing theoretical and practical aspects, researchers and students are encouraged to undertake a thorough review of the **entire document**.

1.4 Project Scope

The **Predictive Healthcare ROI Analysis Platform** is designed with a clearly defined scope, focusing on key functionalities to deliver immediate value:

The system will:

**Ingest Structured Patient Data:** The platform will facilitate the seamless ingestion of structured patient demographic and clinical variables, typically provided in CSV format, serving as the foundational dataset for predictive modeling.

* **Predict Patient Risks for Chronic Conditions:** Leveraging a sophisticated **classifier chain Machine Learning (ML) model**, the system will accurately predict patient risks across a spectrum of chronic conditions, identifying individuals at higher propensity for specific health issues.
* **Forecast Proactive vs. Reactive Treatment Costs:** Utilizing a powerful **Gradient Boosting Regressor ROI model**, the platform will forecast the potential costs associated with both proactive (preventative) and reactive (curative) treatment approaches, enabling a clear financial comparison.
* **Present Results in a Web Interface:** All analytical outcomes and predictions will be intuitively presented within a user-friendly web interface, utilizing interactive elements such as detailed tables, informative charts (e.g., bar charts for cost comparisons, pie charts for risk distribution), and concise summary cards (e.g., total potential savings, highest risk categories).
* **Highlight Financial Savings:** A core objective of the platform is to explicitly highlight and quantify the financial savings achievable through proactive healthcare interventions, directly supporting data-driven decision-making in preventative care strategies.

**Future Enhancements:** While the initial release focuses on core functionalities, the platform is envisioned with a robust roadmap for future enhancements, expanding its capabilities and scalability:

* **Integration of Real-World Anonymized Patient and Financial Datasets:** A critical future step involves integrating the platform with anonymized, real-world patient health records and corresponding financial datasets, significantly enhancing the accuracy and practical applicability of the predictions.
* **Advanced Feature Engineering for More Precise Predictions:** Continuous improvement of the machine learning models will involve exploring and implementing advanced feature engineering techniques to extract richer insights from data, leading to even more precise and reliable predictions.
* **Integration of Secure Databases (PostgreSQL/MySQL) for Large-Scale Datasets:** To handle large volumes of patient and financial data efficiently and securely, the system will integrate with robust relational databases like PostgreSQL or MySQL, ensuring data integrity and fast retrieval.

1.5 References

The development and documentation of this platform are informed by widely accepted industry standards and reputable technical resources:

* **IEEE 830 SRS Standard:** The overarching framework and guidelines for this Software Requirements Specification are based on the IEEE 830 standard, ensuring a comprehensive and well-structured document.
* **Scikit-learn Documentation:** Official documentation for the scikit-learn machine learning library, serving as a primary reference for algorithm implementations and best practices.
* **XGBoost Documentation:** Comprehensive documentation for the XGBoost library, guiding its efficient and effective application within the predictive models.
* **SHAP Documentation:** Reference for the SHAP library, used for model interpretability and explaining the predictions generated by the machine learning models.
* **FastAPI Documentation:** Official documentation for the FastAPI web framework, providing guidance on API development, routing, and data validation.
* **React Documentation:** The authoritative source for React development, supporting the creation of the interactive and responsive user interface.

2. Overall Description2.1 Product Perspective

The Predictive Healthcare ROI Analysis Platform is architected as a **modular full-stack web application**, meticulously structured as a **monorepo**. This architectural choice fosters cohesive development, simplified dependency management, and streamlined deployment. The system is comprised of three distinct yet interconnected components:

* **Frontend (React + Vite):** This layer is responsible for the user-facing interface, offering intuitive functionalities such as CSV file upload mechanisms, seamless integration with backend APIs for data submission and retrieval, and dynamic visualization of analytical results. Its primary role is to provide a rich, interactive, and responsive user experience.
* **Backend (FastAPI + Python):** Serving as the core computational engine, the backend is responsible for hosting the advanced machine learning models, performing complex predictive analytics on submitted data, and exposing well-defined RESTful APIs for communication with the frontend. It acts as the brain of the operation, processing data and generating insights.
* **ML Pipeline:** The heart of the platform's intelligence lies in its two-stage prediction Machine Learning (ML) pipeline. The first stage involves **Risk Stratification**, where patient data is analyzed to predict health risks. The subsequent stage performs **ROI Calculation**, leveraging the risk predictions to forecast treatment costs and potential savings. This integrated pipeline ensures a holistic analysis from risk identification to financial quantification.

2.2 Product Functions

The platform provides a comprehensive suite of functions designed to facilitate seamless data analysis and insightful decision-making:

* **Upload CSV File Containing Patient Data:** Users can easily upload CSV files containing structured patient data, initiating the analytical process. This function is a primary entry point for data into the system.
* **Parse and Validate Patient Records:** Upon upload, the system intelligently parses the CSV file and performs rigorous validation of patient records, ensuring data integrity and flagging any missing or invalid values.
* **Predict Health Risks with XGBoost Classifier Chain:** Leveraging the pre-trained XGBoost classifier chain model, the platform accurately predicts individual patient health risks for various chronic conditions, providing