# Software Requirements Specification (SRS)

AI-Powered Triage System

CS 3338 – Group 17

Esmeralda Amado
Jessica Parker
Jonathan Su
Nneoma Ajawara
Yvan Michel Kemsseu Yobeu

November 2025

## Contents

# 1 Introduction

## 1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to describe the functional and non-functional requirements for the AI-Powered Triage System. This document will be updated at each snapshot checkpoint to reflect the latest progress and system enhancements.

## 1.2 Scope

The system provides healthcare workers with a secure platform that includes:

- A user login system for verified healthcare staff.
- Access to patient demographic and visit history information.
- An AI assistant capable of reading patient records and summarizing relevant medical information.
- The ability to update and store new visit documentation.
- AI-assisted completion of post-visit forms using chat logs.

## 1.3 Definitions and Acronyms

- **AI** — Artificial Intelligence.
- **User** — Verified healthcare staff.
- **Database** — Storage for patient demographic and visit history.
- **Snapshot** — Project development milestone.

## 1.4 References

- CS 3338 Final Project Instructions (Canvas).
- HIPAA documentation (concept reference, not fully implemented).

### 1.5 Document Overview

This SRS includes:

- System description
- Functional requirements
- Non-functional requirements
- AI behavior
- Planning sections for Snapshots 1–4

## 2 Overall Description

### 2.1 Product Perspective

The system is a web application composed of:

- A front-end website for interacting with patients and the AI.
- A secure login portal.
- A back-end API to manage data and AI connections.
- A database storing patient information.
- An AI model integrated to assist medical staff.

### 2.2 User Classes and Characteristics

- **Healthcare Staff**: Must be authenticated users; familiar with patient care workflow.
- **System Administrator**: Manages user permissions and system configuration.

### 2.3 Operating Environment

- Web browser (Chrome, Firefox, Safari, Edge).
- Backend server (Node.js or Python-based).
- Database (PostgreSQL or MySQL).

## 2.4 Design and Implementation Constraints

- Limited scope due to academic timeline.
- HIPAA compliance cannot be fully implemented.
- AI features must be explainable and safe.

## 2.5 Assumptions and Dependencies

- AI API availability.
- Stable internet connection for users.
- Valid healthcare employment verification.

# 3 System Features

## 3.1 Feature F1: User Authentication

### Description

Healthcare staff must log in to access patient data.

### Functional Requirements

- F1.1 The system shall allow users to log in with a username and password.
- F1.2 The system shall validate user credentials securely.
- F1.3 The system shall restrict access for unauthorized users.
- F1.4 Login errors shall be displayed clearly.

## 3.2 Feature F2: Patient Record Management

### Description

Users can view, search, and update patient records.

**Functional Requirements**

- F2.1 Users shall search for patients by name or ID.
- F2.2 Users shall view demographic info (name, age, gender).
- F2.3 Users shall view previous visit history.
- F2.4 Users shall add new medical visits to the patient record.
- F2.5 Users shall update existing visit notes.

## 3.3 Feature F3: AI-Assisted Triage

**Description**

An AI assistant helps summarize patient information and provide triage support.

**Functional Requirements**

- F3.1 The system shall provide an AI chat interface for staff.
- F3.2 The AI shall access patient history data.
- F3.3 The AI shall summarize previous conditions, treatments, and patterns.
- F3.4 The AI shall suggest severity indicators but not diagnose a condition.

## 3.4 Feature F4: Visit Documentation Assistance

**Description**

AI assists staff by generating post-visit documentation.

**Functional Requirements**

- F4.1 The system shall provide a structured visit form.
- F4.2 The AI shall extract useful information from chat logs.
- F4.3 Users shall edit AI-generated content before saving.
- F4.4 The final documentation shall be stored in the database.

# 4 Non-Functional Requirements

## 4.1 Performance

- NFR1: Pages shall load within 3 seconds on average.
- NFR2: AI responses shall generate within 5 seconds.

## 4.2 Security

- NFR3: All sensitive data must be encrypted.
- NFR4: Passwords stored as hashes.
- NFR5: Only authenticated users may access patient records.

## 4.3 Usability

- NFR6: UI shall require minimal training for healthcare staff.
- NFR7: Information shall be presented clearly and consistently.

## 4.4 Reliability

- NFR8: System shall handle at least 20 simultaneous users.
- NFR9: AI must not produce harmful medical claims.

# 5 Snapshots (Planned Updates)

## 5.1 Snapshot 1

- Establish project scope.
- Build initial SRS and SDD structure.
- Create basic repository structure.
- Begin login system outline.

### 5.2 Snapshot 2

- Implement database structure.
- Connect UI to database.
- Add TestRail testing document for this feature.
- Update SRS with new requirements.

### 5.3 Snapshot 3

- Implement AI assistant feature.
- Connect AI to patient record database.
- Add TestRail document for AI testing.
- Update SRS requirements.

### 5.4 Snapshot 4

- Finalize UI and polish system.
- Add future enhancement section.
- Add TestRail document for UI workflow.
- Prepare final SRS revision.

## 6 Appendix

This document will be updated continuously during development and at each snapshot stage.