AI-Powered Triage System
System Design Document (SDD)
Version 2

Group 17
Esmeralda Amado
Jessica Parker
Jonathan Su
Nneoma Ajawara
Yvan Michel Kemsseu Yobeu

December 2025

Contents

Version	Description	Date Added
1.0	Final version for Start Snapshot submission.	12/05/2025
2.0	Added full database design, updated workflow, added CRUD operations and implemented core triage features.	12/08/2025

Version Description

1 Introduction

1.1 Purpose

The purpose of Version 2 of this Software Design Document (SDD) is to describe the newly implemented database design and core triage workflow for the AI-Powered Triage System. This version expands on the initial framework introduced in Version 1.

1.2 Intended Audience

This document is intended for:

- The instructor and the graders of the CS 3338 course.
- Developers responsible for implementing system components.
- Testers working with TestRail and Jira.
- Future maintainers/contributors of the system, in later snapshots.

1.3 System Overview

The AI-Powered Triage System now includes a defined database structure and a functional workflow for:

- Storing patient information.
- Retrieving patient information.

The AI system is still not implemented, but the system architecture has been expanded to support patient data entry and basic triage record management.

2 System Architecture

2.1 Workflow

The workflow of the AI Triage System is as follows:

1. The user logs in through the secure authentication portal.
2. The front-end client requests patient data from the back-end API.
3. Data is retrieved from the database and displayed to the user.
4. The user selects a patient or creates a new patient record.
5. The user updates the patient's visit history or completes forms.
6. The user can later retrieve and review stored visit information (notes).

2.2 Component Breakdown

The AI Triage System will consist of the following components:

- **Client-Side (Front-End)** A web interface built using HTML/CSS/JavaScript or a front-end framework like React.
- **Server-Side (Back-End)** A REST API built using Python Flask, FastAPI, or Node.js for:
  - User authentication
  - Database interactions
  - Passing data to/from the AI model
- **Database Layer** A relational database (MySQL) has been defined for storing patient demographics and visit records. The database doesn't interact with AI yet, but supports basic database organization and retrieval.
- **AI Service Layer** This is not yet implemented and will be introduced in future versions.

3 User Interface

3.1 How to Use

Users will be able to interact with the system, using the following components:

- **Patient Profile Page**: Verified healthcare personnel can also enter visit notes, and view their visit history.
- **Dashboard**: To display a list of patients, recent activity, and quick links.
- **Visit Documentation Form**: A structured form for healthcare personnel to complete patient encounter records.

3.2 Database Design Overview

The database will contain the following tables:

- **Staff** Columns: staff_id, name, email, role, password_hash
- **Patients** Columns: patient_id, name, age, sex, date_of_birth
- **Visits** Columns: visit_id, patient_id, visit_date, condition, treatment, outcome

3.3 UI Screenshots

Screenshots of the tentative UI design may be added in later versions.

4 Glossary

AI - Artificial Intelligence

API - Application Programming Interface

DB - Database

SDD - System Design Document

UI - User Interface

5 References

Current references:

- CS 3338 Lab 14 Instructions (Canvas).
- CS 3338 Final Project Instructions (Canvas).
- Example Final Projects From Past Group (Github).
- Overleaf Documentation: `https://www.overleaf.com/learn`
- HIPAA Overview: `https://www.hhs.gov/hipaa/`