Software Requirements Specification

for

MediBridge

Version: 1.0
Date: November 6, 2025

Prepared for:
Healthcare Technology Division

Prepared by:
Paarth Yadav and Karan Pratap Singh

Contents

1 Introduction

This document provides a detailed Software Requirements Specification (SRS) for the MediBridge system. It describes the system's purpose, scope, functional requirements, non-functional requirements, and external interfaces, serving as a foundational agreement between the development team and stakeholders.

1.1 Purpose

The purpose of this document is to provide a comprehensive description of the requirements for the MediBridge platform. It is intended for project managers, developers, testers, and stakeholders to understand the system's functionalities, constraints, and objectives. MediBridge aims to create a unified ecosystem to connect patients with suitable healthcare facilities in emergencies, ensuring timely access to care supported by readily available medical records.

1.2 Document Conventions

This document follows the IEEE 830-1998 standard for software requirements specifications. All requirements are uniquely identified with a prefix (e.g., FR for Functional Requirement) and a number for traceability.

1.3 Intended Audience and Reading Suggestions

- **Stakeholders & Project Managers:** Should review Sections 1 and 2 for a high-level overview and project scope.
- **Developers & Architects:** Must read the entire document, with a focus on Section 3 for detailed functional and non-functional requirements.
- **QA & Testers:** Should use Section 3 to create test cases and validation plans.

1.4 Product Scope

MediBridge is a comprehensive mobile and web-based platform designed to address critical gaps in emergency healthcare. Its core functionalities include:

- A real-time hospital locator with information on bed and specialist availability.
- A centralized Electronic Health Record (EHR) system for seamless medical history access.
- A secure chat-based guidance system for immediate assistance.
- A streamlined module for organ donation registration and coordination.

The system will serve patients, ambulance services, hospital administrators, and doctors, creating an interconnected healthcare network.

1.5 References

- IEEE Std 830-1998, *IEEE Recommended Practice for Software Requirements Specifications*.

2 Overall Description

2.1 Product Perspective

MediBridge is a new, self-contained product designed to integrate with existing healthcare infrastructures. It will act as an intermediary platform, pulling real-time data from registered hospitals and providing it to end-users. It will interface with external services like mapping APIs for navigation and potentially with government health databases for record verification.

2.2 Product Functions

The major functions of the MediBridge system are summarized as follows:

- **User Management:** Secure registration, authentication, and profile management for all user types.
- **Emergency Services:** Locate nearby hospitals based on real-time data, alert hospitals of incoming patients, and provide navigation.
- **Health Record Management:** Securely store, manage, and share electronic health records (EHR) with authorized medical personnel.
- **Communication Hub:** Provide a chat interface for users to get guidance and for doctors to communicate securely.
- **Organ Donation Registry:** A platform for users to register as organ donors and for hospitals to manage donor information.

2.3 User Characteristics

- **Patients/Family Members:** Individuals seeking emergency medical care. May be under stress and have varying levels of technical proficiency. The interface must be extremely simple and intuitive.
- **Doctors/Medical Staff:** Authorized healthcare professionals who need quick and secure access to patient medical histories to make informed decisions.
- **Hospital Administrators:** Staff responsible for updating their hospital's real-time status (bed availability, on-duty specialists, etc.).
- **Ambulance/Paramedic Crew:** First responders who need to transport patients to the most appropriate and available facility.

2.4 Constraints

- The system must comply with data privacy and security regulations equivalent to HIPAA for handling sensitive medical data.
- The mobile application must be available on both iOS and Android platforms.
- The system requires a stable internet connection for real-time data synchronization.
- Hospital data accuracy is dependent on timely updates from hospital administrators.

2.5 Assumptions and Dependencies

- It is assumed that hospitals will be willing to partner and provide real-time data.
- The system will depend on third-party mapping services (e.g., Google Maps API) for location and navigation functionalities.
- Users are assumed to have smartphones with GPS capabilities for location-based services.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The system will provide a clean, intuitive, and responsive user interface for both mobile and web platforms. The UI will feature high-contrast text, large buttons for critical functions (e.g., ”Find Nearest Hospital”), and a map-based view for navigation.

3.1.2 Hardware Interfaces

The system does not directly interface with any specific hardware other than standard smartphones and web servers. It will utilize the smartphone’s GPS module for location tracking.

3.1.3 Software Interfaces

- **Mapping API:** The system shall interface with a mapping service (e.g., Google Maps API) to display hospital locations and provide turn-by-turn navigation.
- **SMS/Push Notification Gateway:** The system shall use a notification service to send alerts and updates to users.

3.1.4 Communications Interfaces

All communication between the client applications (mobile/web) and the backend server shall be encrypted using HTTPS/TLS protocol. Real-time chat will be implemented using WebSockets.

3.2 Functional Requirements

This section details the functional requirements of the MediBridge system.

3.2.1 Module 1: User Authentication and Profile Management

FR-1.1 The system shall allow users to register as a ’Patient’, ’Doctor’, or ’Hospital Administrator’.

FR-1.2 The system shall provide secure login using email/password.

FR-1.3 Patients shall be able to create and manage their profiles, including personal details and emergency contacts.

3.2.2 Module 2: Emergency Hospital Locator

FR-2.1 The system shall use the user’s current location to display a list and map of nearby hospitals.

FR-2.2 The system shall provide an estimated travel time to the selected hospital and integrate with a mapping application for navigation.

3.2.3 Module 3: Electronic Health Record (EHR) Management

FR-3.1 The system shall provide a centralized database for patients to upload and store their medical documents (prescriptions, lab reports, etc.).

FR-3.2 Patients shall have full control over their records.

FR-3.3 In a declared emergency, a paramedic or hospital can request temporary emergency access to a patient's critical information (blood type, allergies) and records.

3.2.4 Module 4: Chat-based Guidance

FR-4.1 The system shall provide a 24/7 chat service for users to receive non-diagnostic guidance and support.

FR-4.2 The chat system may be initiated with an AI-powered chatbot to handle common queries.

3.2.5 Module 5: Organ Donation Registry

FR-5.1 Users shall be able to register their consent to be an organ donor through the application.

FR-5.2 The organ donor status shall be indicated on the user's profile, visible only to authorized medical personnel in relevant situations.

3.3 Non-Functional Requirements

3.3.1 Security Requirements

NFR-1.1 All user data, especially personal health information (PHI), must be encrypted both in transit and at rest.

NFR-1.2 The system must implement Role-Based Access Control (RBAC) to ensure users can only access data and functions appropriate for their role.

NFR-1.3 The system shall be protected against common web and mobile vulnerabilities (e.g., SQL injection, XSS).

4 System Models

4.1 Use Case Diagram

The use case diagram below illustrates the primary actors (patients, doctors, hospital administrators, and ambulance personnel) and their interactions with the MediBridge system.

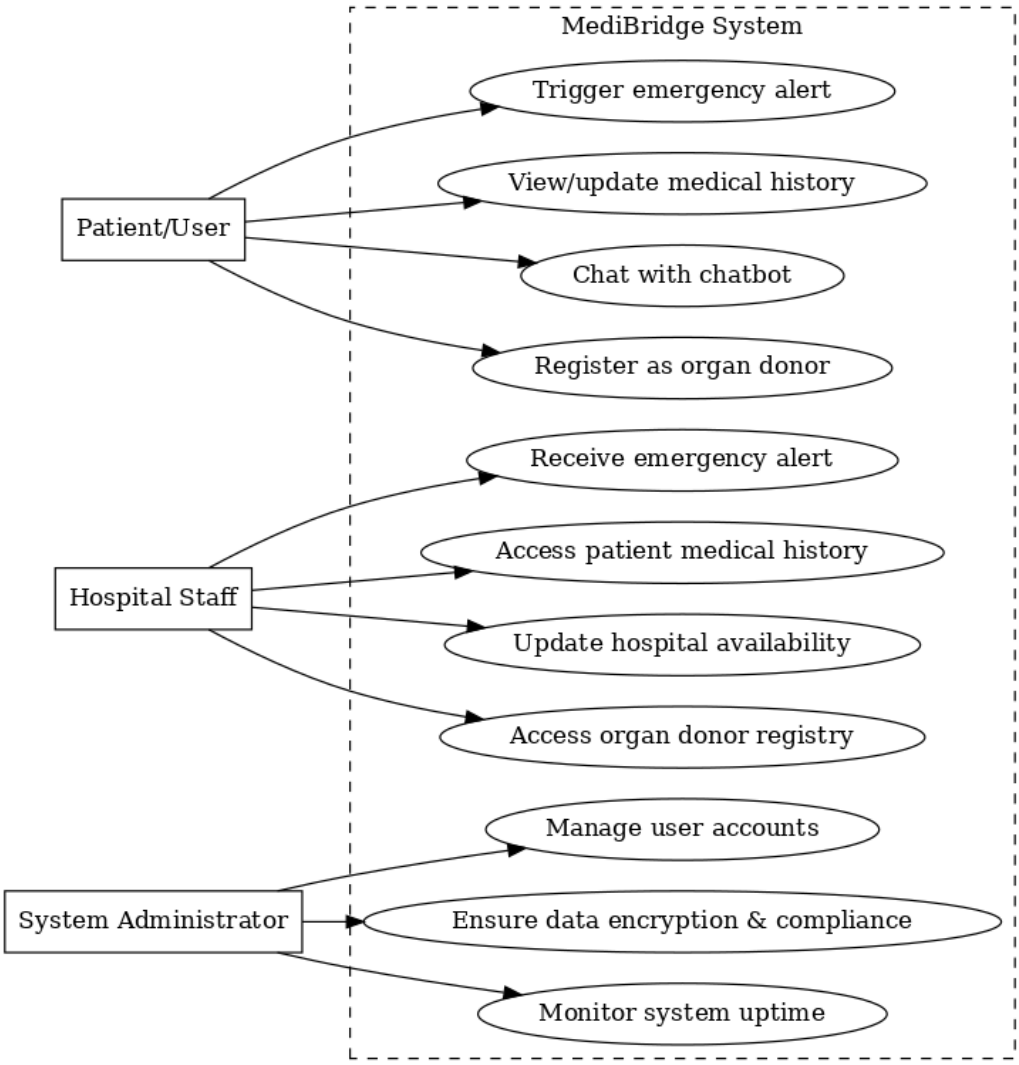

Figure 1: MediBridge Use Case Diagram

4.2 Class Diagram

The class diagram defines the structural relationships among the key system entities and classes within MediBridge, including user roles, EHR records, and hospital modules.

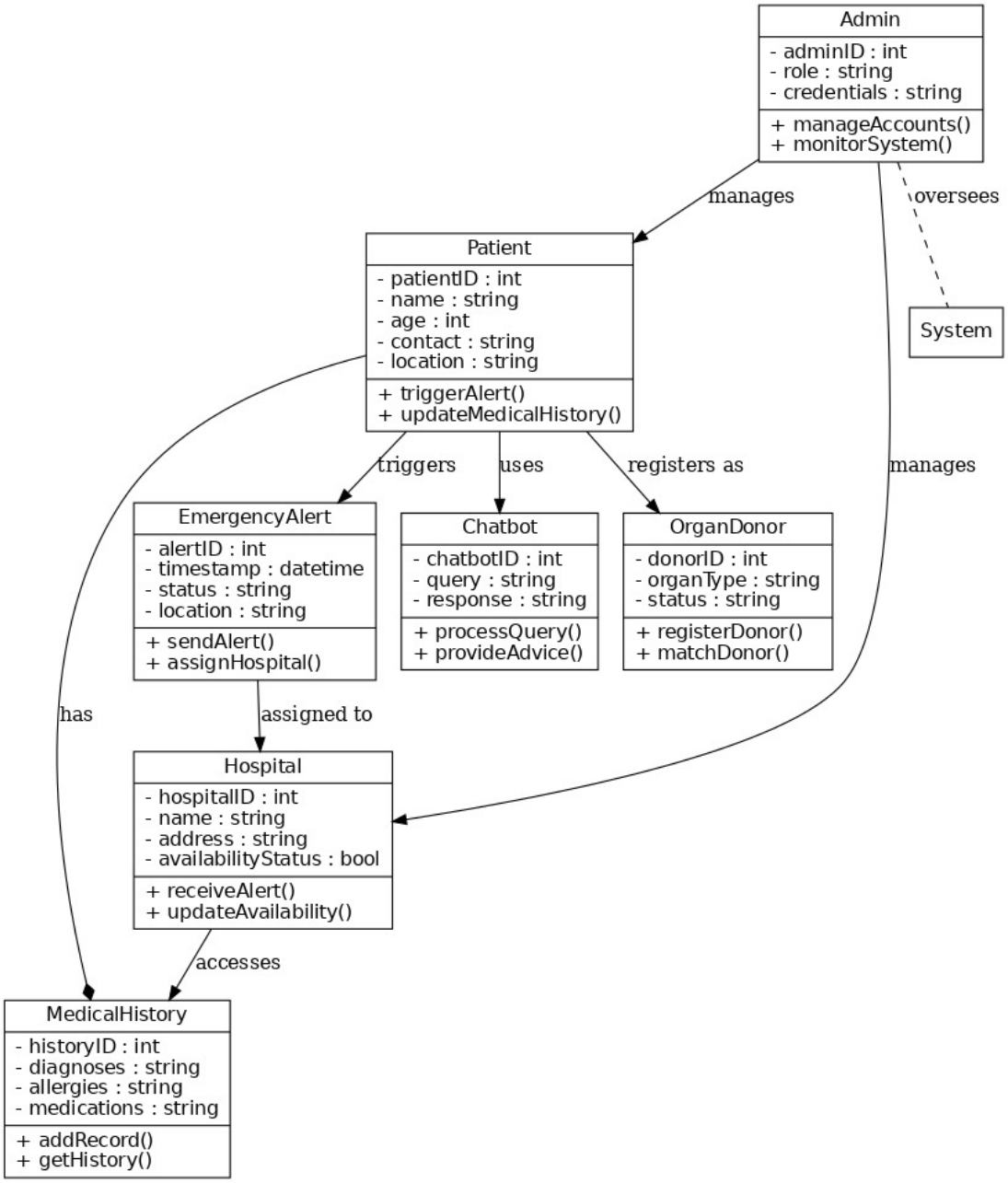

Figure 2: MediBridge Class Diagram

4.3 Sequence Diagram

The sequence diagram defines the structural relationships among the key system entities and classes within MediBridge, including user roles, EHR records, and hospital modules.

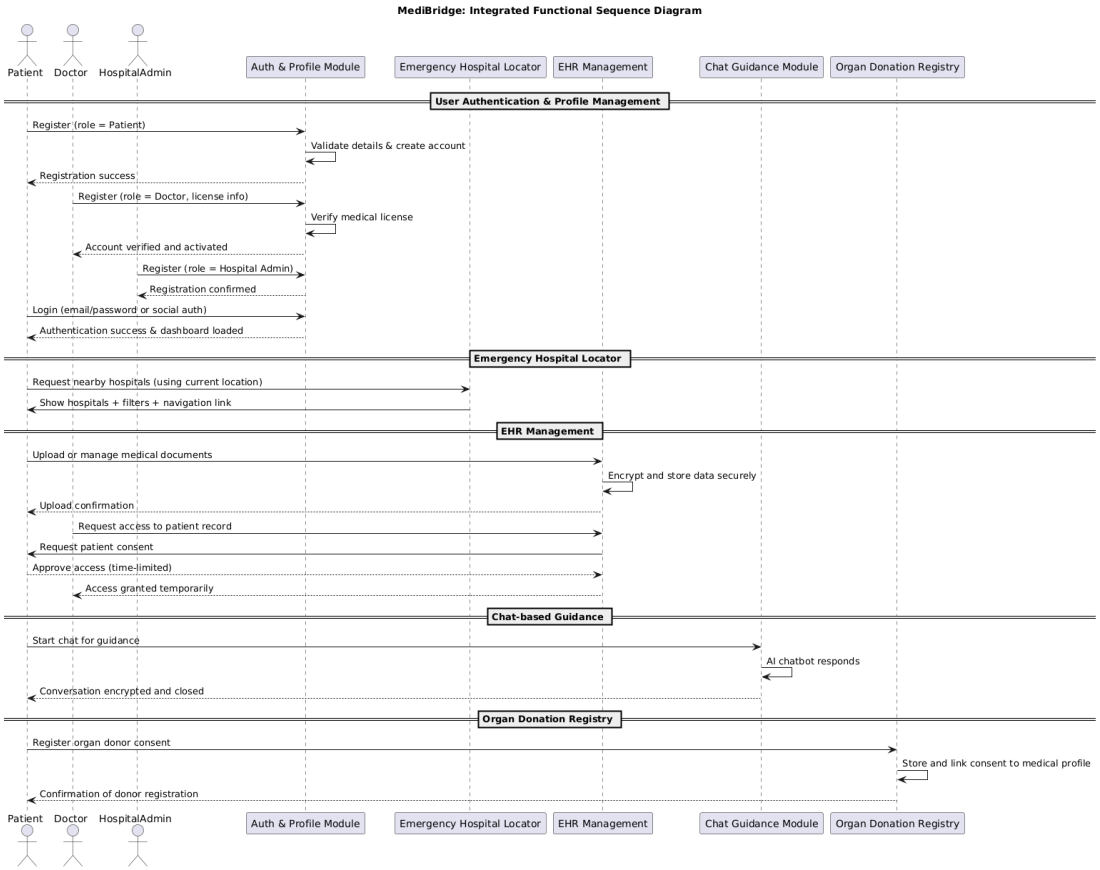

Figure 3: MediBridge Class Diagram

4.4 Activity Diagram

The activity diagram outlines the sequence of actions in the MediBridge system for operations such as user authentication, appointment handling, and electronic health record updates.

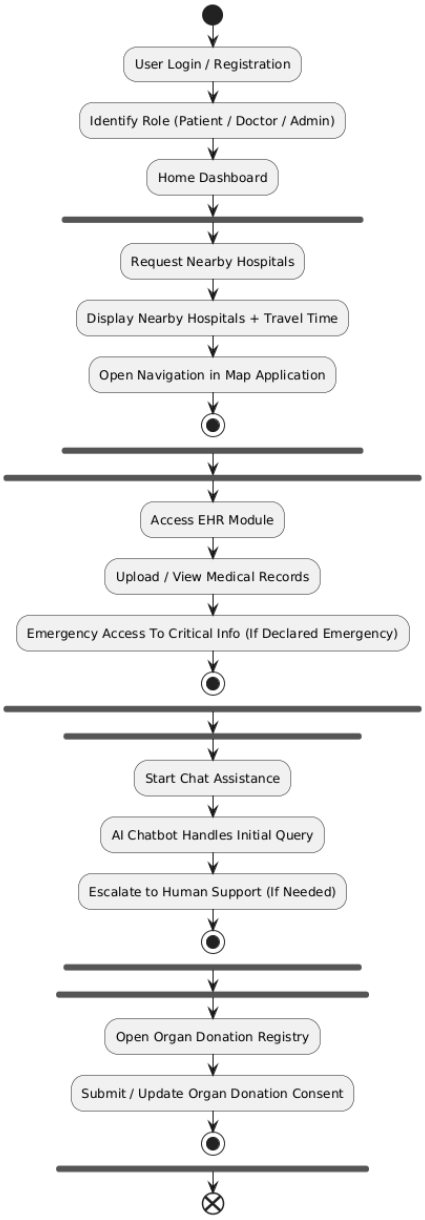

Figure 4: Activity Diagram for Core System Operations

4.5 Entity Relationship (ER) Diagram

The ER diagram illustrates the structure of the database, showing key entities such as Patient, Doctor, Hospital, Appointment, and EHR Records, along with their relationships.

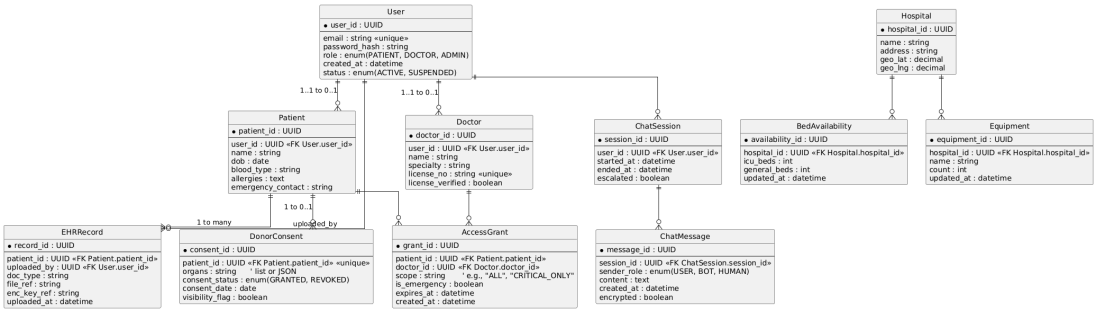

Figure 5: Entity Relationship Diagram

4.6 Data Flow Diagram (DFD)

The DFD represents how data moves through the system, highlighting the interactions between users, system processes, and data storage components.

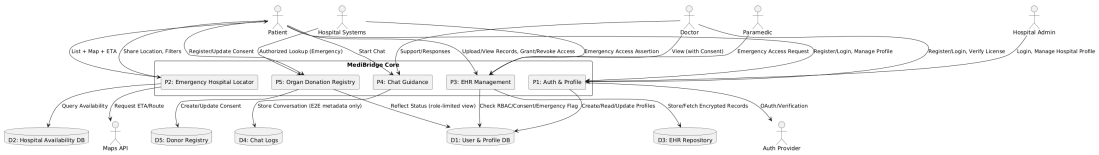

Figure 6: Data Flow Diagram

4.7 Object Diagram

The object diagram shows a snapshot of system objects at a particular moment, reflecting real runtime interactions between instances of system classes.

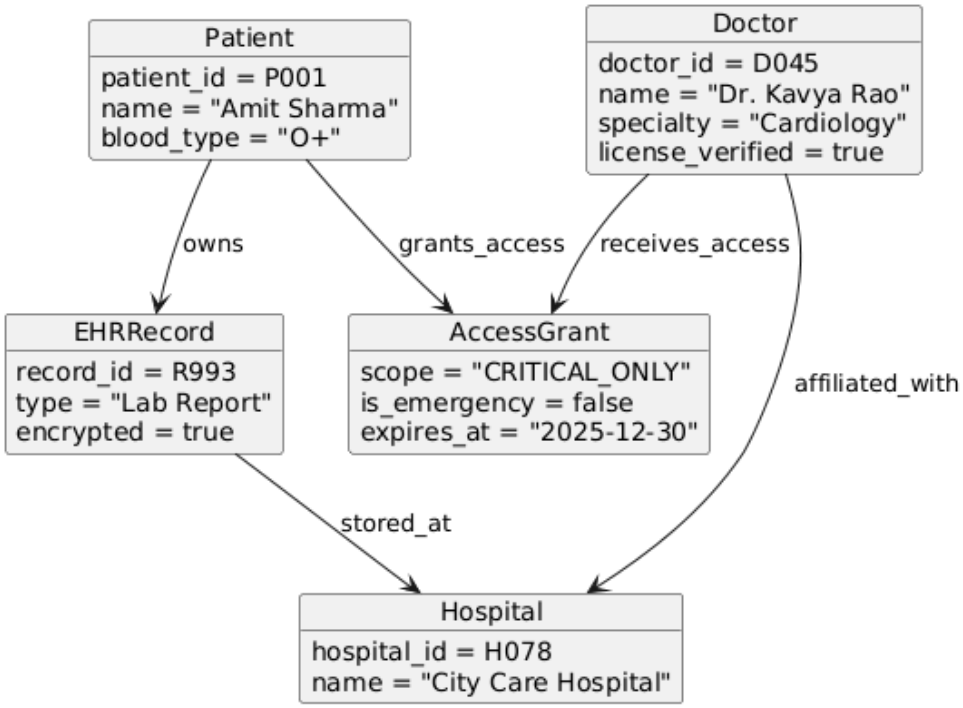

Figure 7: Object Diagram

4.8 State Chart Diagram

The state chart captures how specific system entities, such as a patient record, transition across various states through system operations.

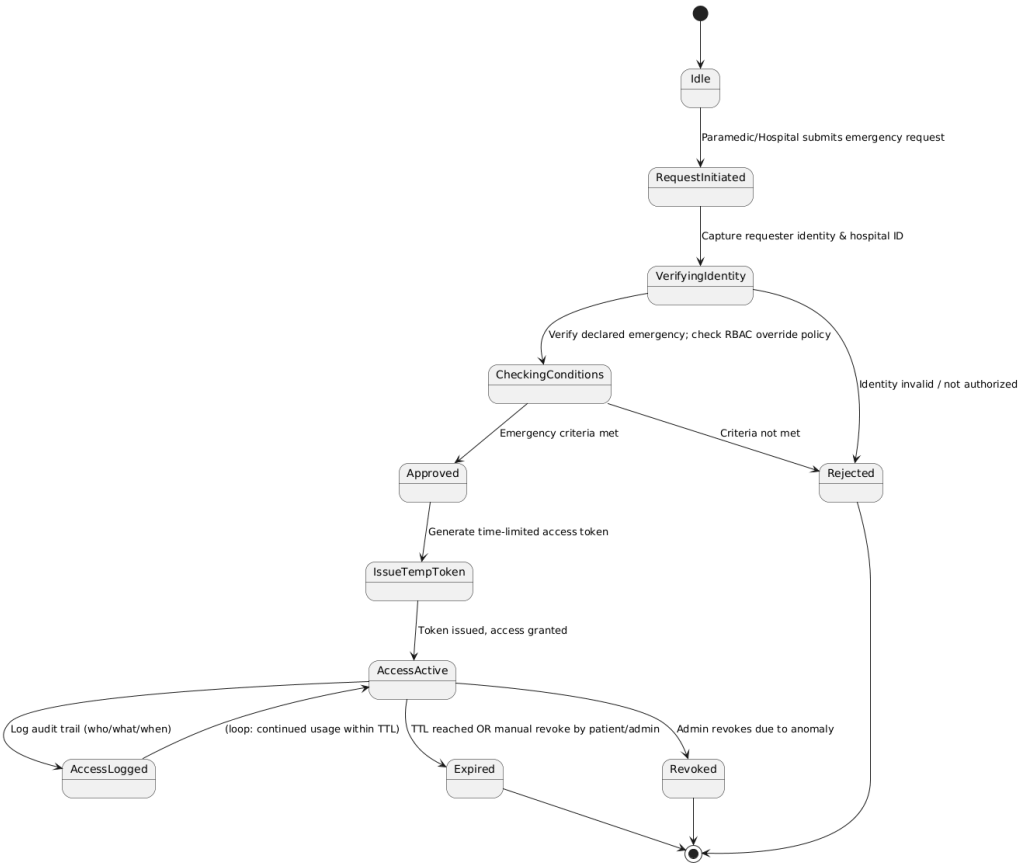

Figure 8: State Chart Diagram

4.9 Component Diagram

The component diagram breaks the system into modular components such as User Interface, Authentication Service, EHR Module, and Database Layer, showing how they interact.

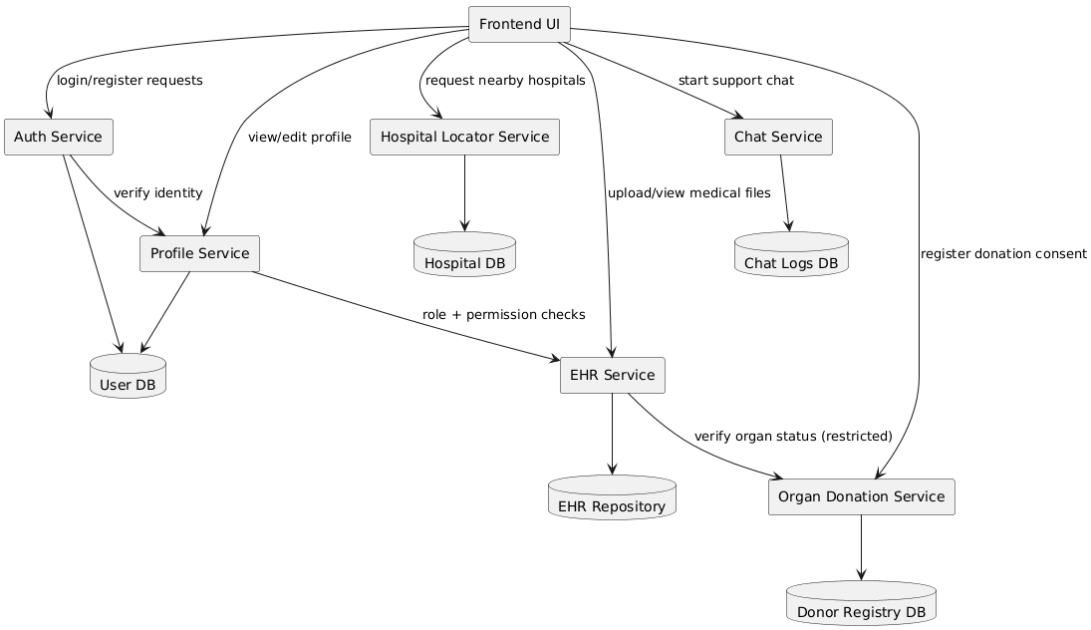

Figure 9: Component Diagram

4.10 Deployment Diagram

The deployment diagram shows how system components are deployed across physical or cloud-based servers, describing execution nodes and their communication links.

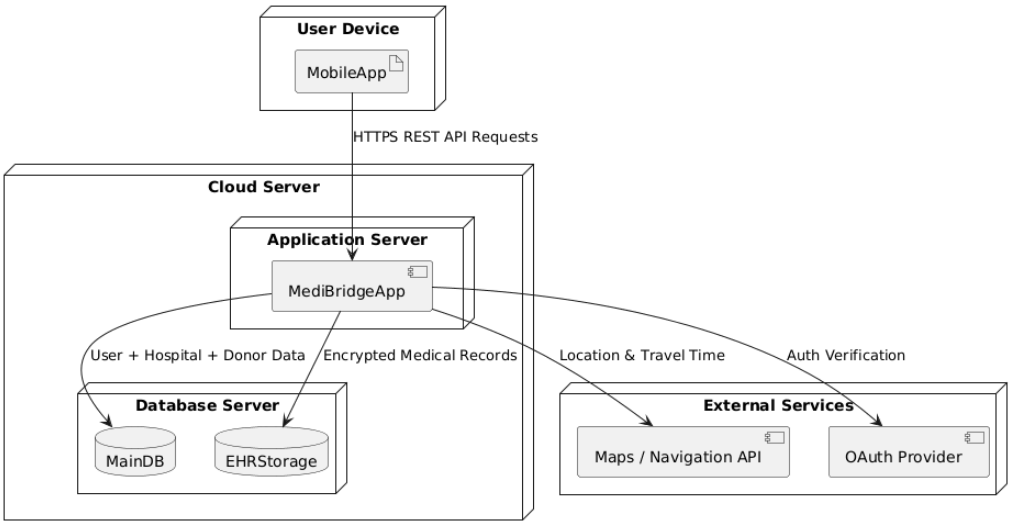

Figure 10: Deployment Diagram