Software Requirements Specification (SRS) Document

Project Title: HealthSphere – An Electronic Health Record (EHR) & Telemedicine Platform

---

1. Introduction

1.1 Purpose

The purpose of this document is to define the software requirements for *HealthSphere*, a web-based Electronic Health Record (EHR) and Telemedicine platform that enables secure doctor–patient interaction, health data management, and remote consultations.

1.2 Scope

HealthSphere provides a centralized, HIPAA-compliant system for healthcare providers and patients.

- **Doctors** can manage patient records, schedule appointments, prescribe medications, and conduct video consultations.
- **Patients** can view their health records, book appointments, and attend telemedicine sessions.
  The system emphasizes **security**, **scalability**, and **usability**.

1.3 Definitions, Acronyms, and Abbreviations

Term	Description
EHR	Electronic Health Record
RBAC	Role-Based Access Control
HIPAA	Health Insurance Portability and Accountability Act
MFA	Multi-Factor Authentication
API	Application Programming Interface
WebRTC	Web Real-Time Communication

---

2. Overall Description

2.1 Product Perspective

HealthSphere is a **full-stack web application** built with:

- **Frontend:** React.js (responsive UI)
- **Backend:** Spring Boot (microservice architecture)
- **Database:** MySQL/PostgreSQL
- **Video Calls:** WebRTC
- **Authentication:** Spring Security with JWT and RBAC

2.2 Product Functions

- User authentication and authorization (Doctor, Patient, Admin)
- Electronic health record (EHR) management
- Appointment scheduling and management
- Telemedicine via secure video conferencing
- E-prescription creation and management
- Notification and reminder system
- Role-based dashboards

2.3 User Characteristics

User Type	Description
Patient	Books appointments, views records, attends online consultations
Doctor	Manages patient records, appointments, consultations, and prescriptions
Admin	Manages system users, roles, and audits for security

2.4 Constraints

- Must comply with **HIPAA** data protection rules.
- Accessible through modern browsers.
- Stable internet connection required for telemedicine.
- Backend and frontend integration using RESTful APIs.

2.5 Assumptions and Dependencies

- Users have registered accounts.
- All doctors are verified healthcare professionals.
- Internet connection available for teleconsultation.
- External services (like email or video APIs) remain active and stable.

---

3. Specific Requirements

3.1 Functional Requirements

ID	Requirement Description
FR1	The system shall allow registration and login using MFA.
FR2	The system shall implement RBAC for Doctor, Patient, and Admin.
FR3	The system shall allow doctors to create, update, and delete patient EHR data.

ID	Requirement Description
FR4	The system shall allow patients to view their medical records.
FR5	The system shall provide appointment scheduling, rescheduling, and cancellation features.
FR6	The system shall support secure video consultation between doctor and patient (WebRTC).
FR7	The system shall allow doctors to generate e-prescriptions and share them digitally.
FR8	The system shall notify patients about upcoming appointments or new prescriptions.
FR9	The system shall allow admins to manage users and maintain audit logs.

3.2 Non-Functional Requirements

Type	Requirement
Performance	The system shall respond within 2 seconds for most operations.
Security	Data shall be encrypted using SSL/TLS. All APIs protected by JWT.
Scalability	The system shall support at least 500 concurrent users.
Usability	UI shall be responsive and intuitive for all age groups.
Availability	The system shall maintain 99% uptime during active hours.
Maintainability	Code shall follow modular microservice architecture.
Compatibility	Compatible with Chrome, Edge, Safari, and Firefox.

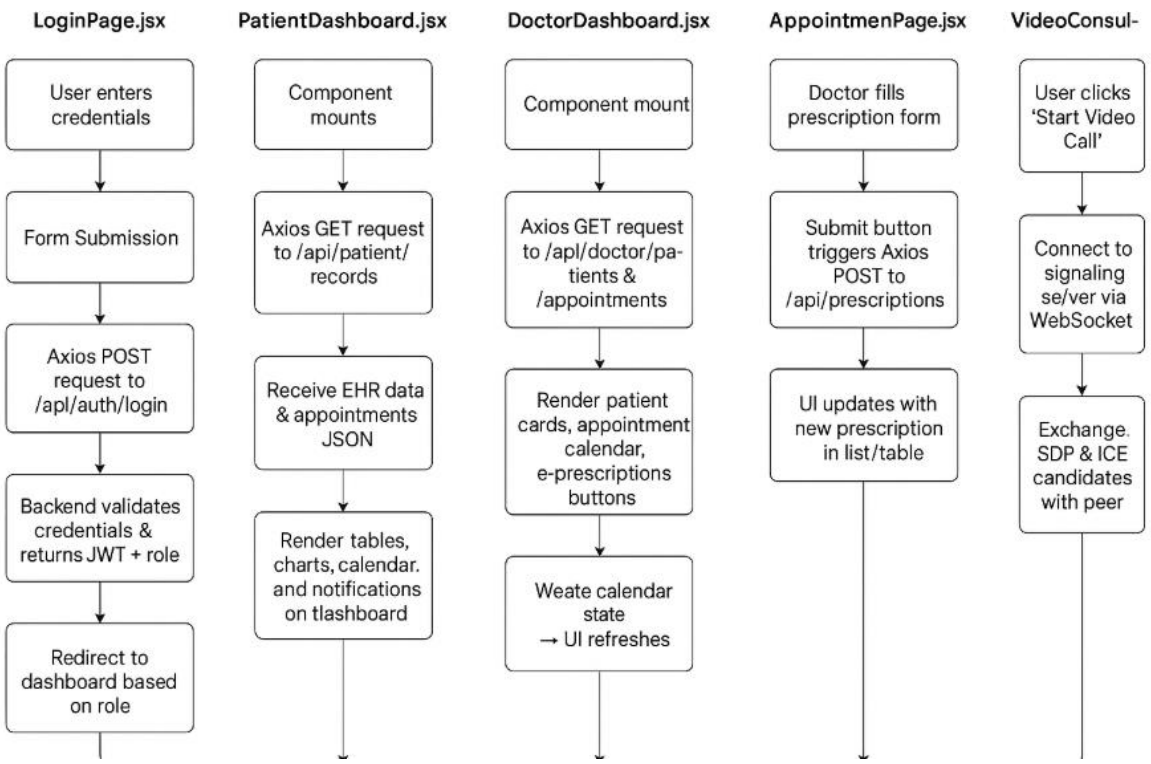
---

4. System Design Overview

4.1 Architecture

- **Frontend:** React.js SPA
- **Backend:** Spring Boot microservices
- **Database:** MySQL
- **Video Calls:** WebRTC with signaling via Spring server
- **Authentication:** JWT + RBAC
- **Notifications:** Email/SMS or Web push

4.2 Data Flow Overview



- User logs in → Token generated → Role identified (Patient/Doctor/Admin)
- API calls handle CRUD for EHR, Appointments, Prescriptions
- Video calls established using WebRTC via signaling server

---

5. Future Enhancements

- AI-based symptom checker or health insights.
- Integration with wearable device data (e.g., Fitbit, smartwatch).
- Cloud-based scalability using Kubernetes or Docker.
- Integration with pharmacy and lab networks.

---

6. Appendix

- **Tools Used:** IntelliJ, VS Code, Postman, GitHub, MySQL Workbench
- **Languages:** Java, JavaScript, HTML, CSS
- **Frameworks:** Spring Boot, React.js