DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Software Engineering [UE22CS341A]

Software Requirements Specification

Version 1.0

Hospital Records Management System

Samarth NN (PES1UG22CS509)

Vidhaan Viswas (PES1UG23CS835)

1. Introduction

1.1 Purpose

The purpose of this document is to specify the requirements for a Hospital Records Management System. This system is designed to automate the management of patient records, streamline administrative tasks, and ensure secure access to medical data. The document outlines the system's functionalities, performance, interface, and other critical factors necessary for development.

1.2 Scope

The Hospital Records Management System will be used in hospitals and healthcare facilities to handle day-to-day operations involving patient data, including patient registration, medical history tracking, appointment scheduling, billing, and reporting. The system will be accessible by hospital staff with varying levels of permissions based on their roles (e.g., Admin, Doctor, Nurse, Receptionist).

1.3 Definitions, Acronyms, and Abbreviations

- **SRS:** Software Requirements Specification
- **EMR**: Electronic Medical Record
- **EHR**: Electronic Health Record
- **GUI**: Graphical User Interface
- **DBMS:** Database Management System
- **CRUD:** Create, Read, Update, Delete

1.4 References

- HL7 (Health Level Seven) Standards Documentation
- HIPAA Compliance Guidelines
- Java Documentation: Oracle Java Documentation
- MySQL Documentation: MySQL Documentation

1.5 Overview

This document is structured to provide a comprehensive overview of the system, including detailed functional and non-functional requirements, system models, and interface descriptions.

2. Overall Description

2.1 Product Perspective

The Hospital Records Management System is a standalone application designed to integrate with the hospital's existing IT infrastructure. It is intended to replace manual record-keeping processes, thereby reducing errors and improving efficiency. The system will interact with a backend database to store and retrieve patient data.

2.2 Product Functions

- **User Authentication:** Secure login and role-based access control.
- **Patient Management:** Register new patients, update patient information, view medical history.
- **Appointment Scheduling:** Schedule, update, and cancel patient appointments.
- **Medical Records Management:** Store and retrieve patient medical records, including diagnosis, treatment, and prescriptions.
- **Billing:** Generate and print bills for patients, apply insurance details, and calculate totals.
- **Reporting**: Produce various reports for patient statistics, billing, and hospital operations.

2.3 User Classes and Characteristics

- **Admin:** Has full access to the system, including user management, patient management, and report generation.
- **Doctor:** Can access patient medical records, update diagnosis and treatment plans, and view relevant reports.
- **Nurse:** Can view and update patient vital signs and assist with patient care.
- **Receptionist:** Can manage patient registration, appointments, and basic patient information.

2.4 Operating Environment

- **Operating System**: Windows/Linux/MacOS
- **Database:** MySQL or PostgreSQL
- **Programming Language:** Java
- **Hardware:** Standard X-86 based work station with at least 8 GB RAM, 100 GB HDD

2.5 Design and Implementation Constraints

- The system must be developed using Java for the front-end and SQL for the back-end.
- The system should comply with HIPAA standards for data security and privacy.
- The application must ensure data security, particularly for patient and medical data.

2.6 Assumptions and Dependencies

- Users have basic knowledge of using a computer and can navigate a GUI.
- The system depends on a stable database connection for real-time operations.
- The hospital's network infrastructure must support client-server communication.

3. System Features

3.1 User Authentication

3.1.1 Description

The system will require users to log in with a username and password. User roles (Admin, Doctor, Nurse, Receptionist) will determine the level of access to system features.

3.1.2 Functional Requirements

- **REQ-1.1**: The system shall validate user credentials against the database.
- **REQ-1.2:** The system shall allow Admin users to manage user accounts.
- **REQ-1.3**: The system shall log user activities for audit purposes.

3.2 Patient Management

3.2.1 Description

The system will allow users to add, update, delete, and view patient information, including personal details, medical history, and insurance information.

3.2.2 Functional Requirements

- **REQ-2.1:** The system shall allow Admin and Receptionist users to register new patients.
- **REQ-2.2:** The system shall allow users to update existing patient details.
- **REQ-2.3:** The system shall display a list of patients with search and filter options.
- **REQ-2.4:** The system shall provide a detailed view of a patient's medical history.

3.3 Appointment Scheduling

3.3.1 Description

The appointment scheduling module will manage patient appointments, including booking, updating, and cancelling.

3.3.2 Functional Requirements

- **REQ-3.1:** The system shall allow users to schedule new appointments.
- **REQ-3.2:** The system shall allow users to update and cancel existing appointments.
- **REQ-3.3:** The system shall send reminders for upcoming appointments.

3.4 Medical Records Management

3.4.1 Description

The system will manage patient medical records, including diagnosis, treatment plans, and prescriptions.

3.4.2 Functional Requirements

- **REQ-4.1:** The system shall allow Doctors to create and update medical records.
- **REQ-4.2:** The system shall provide a view of a patient's complete medical history.
- **REQ-4.3:** The system shall ensure that only authorized users can access sensitive medical information.

3.5 Billing

3.5.1 Description

The billing module will handle patient billing, insurance claims, and payment processing.

3.5.2 Functional Requirements

- **REQ-5.1:** The system shall generate bills based on services provided and applicable insurance.
- **REQ-5.2:** The system shall allow users to apply insurance details and calculate the patient’s out-of-pocket costs.
- **REQ-5.3:** The system shall generate printable and electronic billing statements.

3.6 Reporting

3.6.1 Description

The reporting module will provide reports for patient statistics, billing, and operational analysis.

3.6.2 Functional Requirements

- **REQ-6.1:** The system shall generate daily, weekly, and monthly reports on patient admissions and discharges.
- **REQ-6.2:** The system shall generate billing and payment reports.
- **REQ-6.3:** The system shall allow reports to be exported in PDF and Excel formats.

4. External Interface Requirements

4.1 User Interfaces

- The system will have a GUI developed using Java Swing or JavaFX.
- The main interface will include menus for navigating different modules (e.g., Patient Management, Appointments, Billing, Reports).
- Forms will be used for data entry with validation for mandatory fields.

4.2 Hardware Interfaces

- The system will interface with standard input devices (keyboard, mouse) and output devices (monitor, printer).

4.3 Software Interfaces

- The system will interface with a MySQL or PostgreSQL database for data storage and retrieval.
- The system will generate reports that can be exported to third-party applications like Excel and PDF viewers.

4.4 Communications Interfaces

- The system will communicate with the database over TCP/IP if deployed in a client-server environment.

5. Other Non-Functional Requirements

5.1 Performance Requirements

- The system should handle up to 500 patient records per day without performance degradation.
- The system should load the patient list within 3 seconds for up to 1000 records.

5.2 Safety Requirements

- The system should ensure data integrity by using transactions to prevent data loss in case of a failure during data operations.
- Regular backups should be scheduled to prevent data loss.

5.3 Security Requirements

- User passwords should be stored in the database using encryption.
- The system should enforce role-based access control to restrict unauthorized access to certain features.
- The system should comply with HIPAA regulations for patient data privacy and security.

5.4 Software Quality Attributes

- **Usability:** The system should have an intuitive interface, making it easy for users to perform tasks with minimal training.
- **Reliability:** The system should have an uptime of 99.9% and handle errors gracefully without crashing.
- **Maintainability:** The system's codebase should be modular and well-documented to facilitate easy updates and maintenance.

5.5 Business Rules

- All patients must have a unique identifier (Patient ID).
- Only Admin users can delete patient records or modify user roles.
- Insurance claims should not exceed the amount specified by the patient's insurance provider.

6. Other Requirements

6.1 Database Backup

- The system should automatically back up the database daily and store the backup in a secure location.

6.2 System Documentation

- The system should include a user manual and technical documentation covering installation, configuration, and troubleshooting.

6.3 Training

- Training sessions should be provided for hospital staff to familiarize them with the system's features.

7. Requirements Traceability Matrix (RTM)

Requirement ID	Requirement Description	Source	Design Document	Development	Test Case ID	Status	Comments
R1	System must secure patient record management.	Business Analyst	Design Doc v1	Dev Team A	TC01	In Progress	
R2	System should allow scheduling of patient appointment.	Business Analyst	Design Doc v1	Dev Team B	TC02	Completed	
R3	System must generate detailed medical reports.	Business Analyst	Design Doc v2	Dev Team C	TC03	Not Started	
R4	System should integrate with insurance for billing.	Business Analyst	Design Doc v3	Dev Team A	TC04	In Progress	
R5	System must provide patient statistics reporting.	Business Analyst	Design Doc v2	Dev Team B	TC05	Not Started	