Software Requirements Specification

for

Hospital Management System

Version 1.0 approved

Prepared by:

Sagar Puppala – 22bd1a661g

Rajiv Reddy - 22bd1a661k

Pradyumna Prahas – 22bd1a661j

J. Pradyumna – 22bd1a661e

Bharath Reddy – 22bd1a6613

Keshav Memorial Institute of Technology

18-10-2024

Table of Contents

Revision History

Name	Date	Reasons for Changes	Version
Week-1	26-09-2024	SRS Template Creation(Introduction)	1.0
Week -2	18-10-2024	SRS Document – Usecase & Class Diagram	2.0

1.Introduction

1.1 Purpose

The purpose of this system is to bring together hospitals, doctors, staff, patients, and other related parties into one unified platform. It aims to make interactions between these parties easier and more efficient, ultimately improving the quality of healthcare services. The system will help hospitals manage daily tasks more effectively by providing a global database that supports better decision-making, allowing users such as patients to have more options and hospitals to handle operations smoothly

1.2 Document Convention

Heading:

Font-Size:16

Font-Style: Bold

Font: Times New Roman

Subheading:

Font-Size:14

Font-Style: Bold

Font: Times New Roman

Content:

Font-Size:12

Font: Times New Roman.

1.3 Intended Audience and Reading Suggestions

This document is for several stakeholders:

- **Developers**: Focus on system requirements and architecture to design and build the Hospital Management System (HMS).
- **Project Managers**: Concentrate on timelines, resources, risks, and costs to keep the project on track and within budget.
- **Healthcare Administrators**: Understand how HMS improves workflows, manages patients, and enhances hospital operations.
- **Advertisers/Marketers**: Highlight features like seamless service integration and better patient care when promoting the system.

- **Testers**: Identify criteria for testing the system's functions and performance to ensure it meets quality standards.

1.4 Product Scope

The HMS will be a centralized platform that manages hospital operations and stores a global database of all related parties. Key features include a user-friendly login for patients and staff, access to patient services and records, admin control for database management, and integration of essential operations like patient registration, doctor schedules, billing, and reporting. The system will secure sensitive information while improving efficiency.

1.5 References

- HL7 (Health Level 7), HIPAA compliance, and ISO 13485 for medical systems.
- MySQL/PostgreSQL documentation, API guides for third-party integration, and software framework manuals (React, Spring Boot,C++ dev).
- Existing system design documents and market research on hospital management systems.
- National healthcare regulations
- Data protection laws.

2. Overall Description

2.1 Product Perspective

The HMS will act as a comprehensive, centralized system that automates hospital operations. It solves problems caused by disconnected systems that lead to inefficiencies and data duplication. HMS will integrate functions such as patient registration, doctor appointments, staff management, and billing, leading to better coordination and improved decision-making, which enhances patient care and hospital efficiency.

2.2 Product Functions

- **Patient Management**: Register and securely store patient personal and medical details.
- **Doctor Scheduling**: Manage doctor and staff schedules to optimize hospital resources and minimize wait times.
- **Medical Records Management**: Update and store medical histories for authorized access.
- **Billing and Payments**: Generate accurate bills and track real-time payments.
- **Patient Portal**: Provide patients with secure access to their medical reports and billing information.
- **Reporting and Analytics**: Generate reports for better decision-making regarding hospital operations and patient care.

2.3 Operating Environment

HMS will be accessible on desktops, tablets, and smartphones through browsers like Chrome and Firefox. It requires an internet connection to function in real time, and patient and hospital data will be securely stored in the cloud, ensuring scalability and secure management of sensitive data.

2.4 User Characteristics

- **Doctors:** Access and update medical records, schedule appointments.
- **Receptionist:** Handle administrative tasks, manage admissions, and billing.
- **Patients:** View medical reports and bills through the patient portal.
- **Administrators:** Monitor system performance, generate reports, and manage hospital operations.

2.5 Design and Implementation Constraints

The system must ensure data security (e.g., HIPAA compliance) and scalability to support growing hospital needs. It should update records in real-time, and different user roles must restrict access to sensitive data based on authorization levels.

2.6 User documentation

User guides will be available for staff, doctors, and patients. These guides will explain how to use key features like scheduling, billing, and accessing records. Training materials will also be provided to help users learn the system efficiently.

2.7 Assumptions and Dependencies

The system assumes proper training for users, a stable internet connection, and adherence to healthcare standards. It also depends on cloud infrastructure to ensure data security, scalability, and recovery in case of system failure.

3.External Interface Requirements

3.1 User Interfaces

The system will feature user-friendly login screens with an interactive GUI, incorporating intuitive icons, buttons, and clear fonts for an enhanced user experience. When accessing the database, information will be displayed in a well-organized, tabulated format for easy readability.

3.2 Hardware Interfaces

The HMS will work on computers and mobile devices with standard hardware specifications, supporting peripherals like monitors and input devices.

3.3 Software Interfaces

The system will use Java. MySQL will be used for the database, and third-party services like payment gateways will be integrated for financial transactions.

3.4 Communication Interfaces

The system will rely on internet protocols like HTTP and FTP for downloading medical reports and managing billing data, ensuring a seamless experience for users without requiring specific browsers.

4. System Features

4.1 Patient Registration and Admission

4.1.1. Description and Priority

This feature allows staff to register patients and manage admission details. It ensures that patients' information is securely stored and easily accessible for future care. **High Priority**

4.1.2. Stimulus/Response Sequences

- **Stimulus**: Admin/Receptionist inputs a new patient's details for registration.
- **Response**: The system assigns a unique patient ID and stores patient information linked to the ID.

4.1.3. Functional Requirements

- REQ-1: The system must generate a unique patient ID for each new patient.
- REQ-2: The system should allow hospital staff to add, view, update, or delete patient records.
- REQ-3: To ensure there is no duplicate records, the system must validate the patient information

4.2 Doctor Consultation and Scheduling

4.2.1. Description and Priority

Patients and staff can schedule appointments based on doctors' availability, helping avoid conflicts and streamlining care. **High Priority**

4.2.2. Stimulus/Response Sequences

- **Stimulus**: A patient requests an appointment with a doctor.
- **Response**: The system checks doctor availability, schedules the appointment, and notifies both the patient and the doctor.

4.2.3. Functional Requirements

- REQ-1: The system must display real-time doctor availability.
- REQ-2: The system must send notifications to patients and doctors when appointments are scheduled, rescheduled, or cancelled.

4.3 Medical Records Management

4.3.1. Description and Priority

Authorized users can access and update patient medical records, ensuring the information is current and accurate. **High Priority**

4.3.2. Stimulus/Response Sequences

- **Stimulus**: A doctor requests access to a patient's medical records.
- **Response**: The system retrieves the relevant patient data and displays it for the requested purpose.

4.3.3. Functional Requirements

- REQ-1: Only authorized personnel can view or modify medical records.
- REQ-2: The system should maintain a detailed trail of any updates made to medical records.
- REQ-3: The system must ensure secure access to medical records.

4.4 Billing and Financial Management

4.4.1. Description and Priority

The system will generate invoices, track payments, and handle financial records to ensure accurate billing. **High Priority**

4.4.2. Stimulus/Response Sequences

- **Stimulus**: Hospital services are rendered (e.g., doctor consultations, treatments, medications).
- **Response**: The system generates an invoice based on the services provided, updates the patient's financial record, and tracks payments.

4.4.3. Functional Requirements

- REQ-1: The system must automatically generate an invoice based on the services rendered.
- REQ-2: The system should support multiple payment methods (cash, card, insurance claims).

4.5 Staff and Role Management

4.5.1. Description and Priority

Administrators can manage staff details and schedules, ensuring smooth hospital operations. **Medium to Low Priority**

4.5.2. Stimulus/Response Sequences

- **Stimulus**: Admin enters new staff details or updates staff schedules.
- **Response**: The system updates the staff's roles, schedules, and personal details in the database.

4.5.3. Functional Requirements

- REQ-1: The system must maintain detailed records for all hospital staff, including roles, work hours, and shifts.
- REQ-2: The system should generate payroll reports based on staff roles and attendance.
- REQ-3: The system must allow for updating staff roles and schedules.

4.6 Patient Self-Service and Report Access

4.6.1. Description and Priority

Patients can securely access their medical records and reports through the system, ensuring data privacy. **High Priority**

4.6.2. Stimulus/Response Sequences

- **Stimulus**: A patient requests access to their medical report.
- **Response**: The system verifies the patient's identity and provides access to the report.

4.6.3. Functional Requirements

- REQ-1: The system must allow patients to securely access their medical records and reports.
- REQ-2: The system must send notifications when new reports or updates are available.

4.7 Reporting and Analytics

4.7.1. Description and Priority

Admins can generate reports to analyze hospital performance, patient care, and finances.

High Priority

4.7.2. Stimulus/Response Sequences

- **Stimulus**: Admin requests a financial or operational report.
- **Response**: The system retrieves relevant data and generates a detailed report.

4.7.3. Functional Requirements

- REQ-1: The system must generate patient reports.
- REQ-2: The system must store historical data to facilitate trend analysis and performance evaluation.

5. Nonfunctional Requirements

5.1 Performance Requirements

The system should retrieve patient data within 2 seconds and handle 1000 concurrent users without slowdowns.

5.2 Safety Requirements

Data backups must be performed daily, and rollback features should allow correction of errors.

5.3 Security Requirements

The system must use strong encryption and implement role-based access controls. An audit trail will track any access or modifications to sensitive data.

5.4 Software Quality Attributes

- **Reliability:** 99.9% uptime(Servers should ideally be running at all times).
- **Scalability:** Must support growing hospital needs.
- **Maintainability:** Modular design for easy updates.
- **Usability:** Intuitive interface with minimal training needed.
- **Compatibility:** The system should integrate with third-party medical systems

5.5 Business Rules

The system must support various hospital billing models and comply with local accounting and healthcare regulations.

6. Other Requirements

6.1 Regulatory Compliance

The HMS must adhere to HIPAA, EHR regulations, and data protection laws to ensure secure patient data management.

6.2 Data Backup and Recovery

Daily automated backups and a disaster recovery plan must be in place to ensure continuity of service.

6.3 User Accessibility

The system must follow accessibility standards to accommodate all users, including those with disabilities.

6.4 Internationalization

HMS application should support multiple languages and currencies for a global user base.

6.5 Scalability

The system must scale to accommodate increased patient numbers and data as the user base increases.

6.6 Sessions

Strong session management is required to prevent unauthorized access, with features like session expiration and two-factor authentication (2FA). This maintains the integrity of the HMS.

6.7 Third-Party APIs

The HMS must easily integrate with necessary third-party APIS, while maintain the security and privacy of the patients, staff and doctors.

Appendix A: Glossary

- **EHR:** Electronic Health Record, a digital version of a patient's paper chart.
- **HMS:** Hospital Management System
- **2FA:** Two-Factor Authentication
- **API:** Application Programming Interface, which allows different systems to communicate.
- **SQL:** Structured Query Language

Appendix B: Analysis Models

Use Case ID	0987654321
Use Case Name	Hospital Management System
End Objective	To optimize the operations of a hospital or healthcare facility, ensuring efficient management of patient data, resources, workflows, and administrative tasks.

Created by	1. Pradyumna Jonnalagadda 2.Pradyumna Prahas 3. Sagar Puppala 4.Rajiv Reddy 5.Bharat Sashank Reddy	On (date):	October 18, 2024
User/Actor	Patient (User),Hospital Administration System		
Trigger	User (doctor, administrator, or patient) initiates a system action, such as scheduling an appointment, checking patient medical records, updating patient information, managing bed availability, processing billing or insurance claims, ordering lab tests, or managing hospital resources.		

Basic/Normal Flows

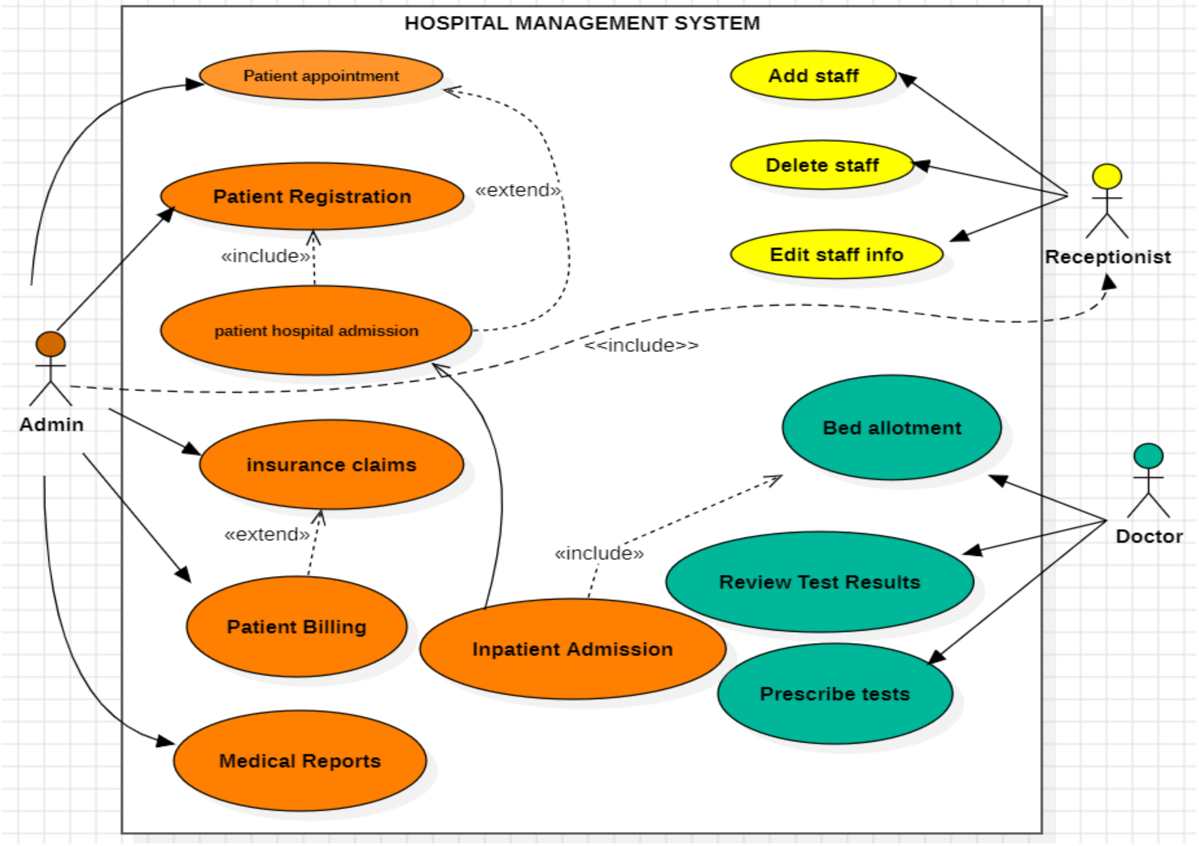
User Actions	System Actions
The **Receptionist** schedules a patient appointment.	The system creates the appointment and assigns it to a doctor.
The **Receptionist** registers a patient.	The system saves the patient’s information and generates a medical record.
The **Receptionist** submits an insurance claim.	The system processes the claim and updates the billing information.
The **Doctor** reviews test results.	The system fetches and displays the test results for the doctor.
The **Doctor** prescribes tests for a patient.	The system records the test prescriptions and sends the request to the lab.
The **Doctor** assigns bed allotment for a patient.	The system allocates the appropriate bed based on availability.
The **Record System Administrator** adds /updates/deletes staff.	The system creates a new/deletes/updates staff profile with the provided details.

Exception Flows

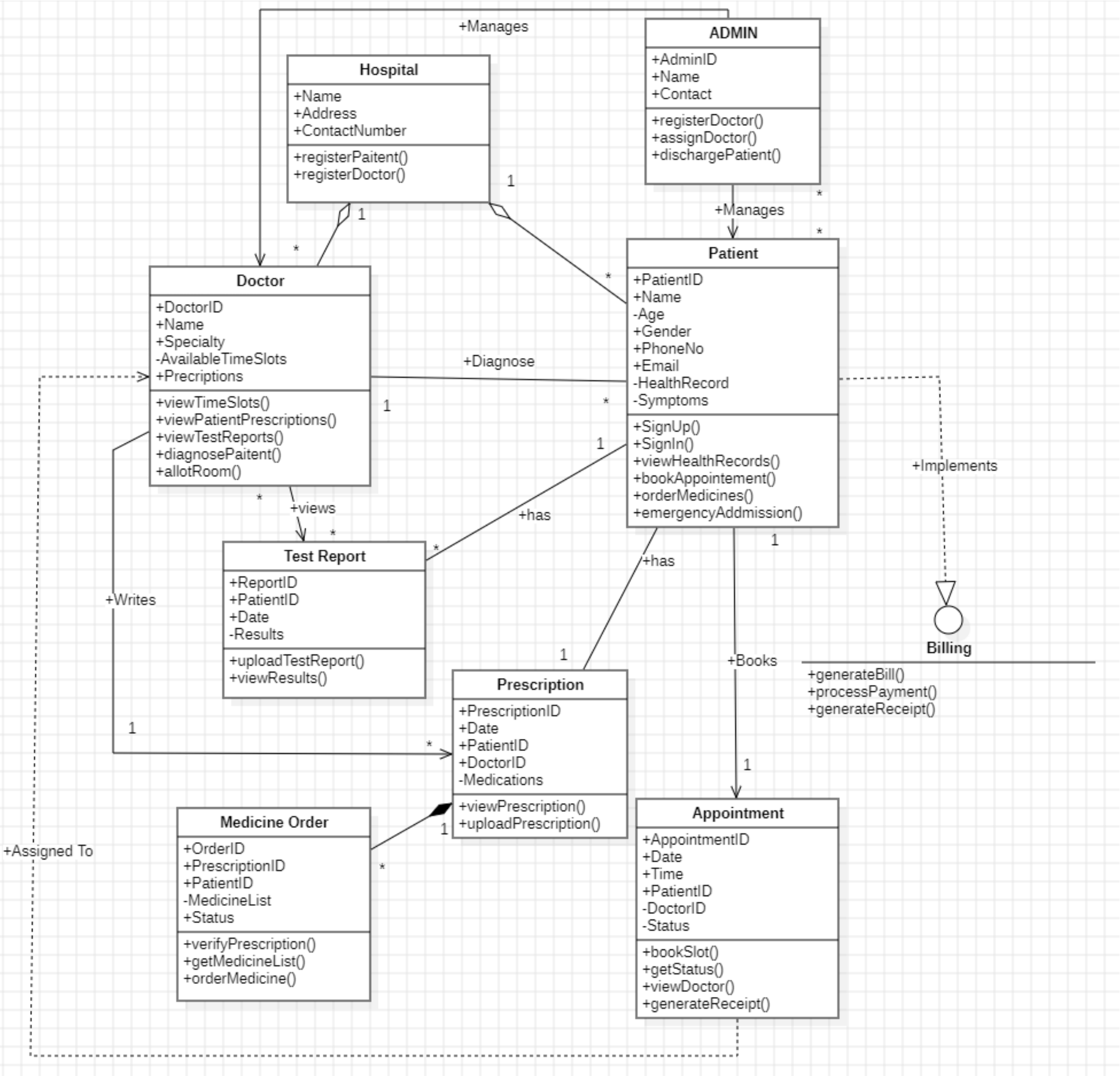
User Actions	System Actions
The **Receptionist** tries to schedule a patient appointment, but the selected doctor is unavailable.	The system notifies the receptionist that the doctor is unavailable and suggests alternative time slots or doctors.
The **Receptionist** tries to register a patient, but required patient information is incomplete or invalid.	The system prompts the receptionist to provide or correct the missing or invalid patient details before proceeding with registration.

The **Receptionist** initiates billing, but there are errors in calculating charges (e.g., missing service entries).	The system generates an error message indicating incomplete billing data and requests correction.
The **Doctor** prescribes tests, but the required tests are unavailable in the system.	The system alerts the doctor that the specific test is not available and suggests alternative tests or prompts for manual entry.
The **Record System Administrator** tries to add a new staff member, but the staff ID is already in use.	The system prevents the addition of the new staff member and informs the administrator about the duplicate staff ID, suggesting a new one.
The **Record System Administrator** tries to delete staff information, but the staff member is linked to active hospital processes (e.g., assigned patients or tasks).	The system blocks the deletion and notifies the administrator that the staff member cannot be deleted due to ongoing assignments.

Use Case Diagram:



Class Diagram:



Appendix C: To Be Determined List

- Third-party lab systems to integrate.
- Final roles and permissions for staff.
- Performance benchmarks for peak usage.
- Mobile Application Requirements.
- Decide whether analytics of reports is done manually or by AI modules