# **Software Requirements Specification (SRS)**

# **Hospital Management System**

## **1. Introduction**

### **1.1 Purpose**

This Software Requirements Specification (SRS) document outlines the requirements for the Hospital Management System (HMS). This system is designed to streamline the operations of a hospital by managing patient records, doctor appointments, medicine prescriptions, and inventory.

### **1.2 Scope**

The Hospital Management System will provide functionality for three main user roles: Receptionist (Admin), Doctor, and Store Manager. The system will handle patient registration, appointment scheduling, doctor assignments, prescription management, and medicine inventory control.

### **1.3 Definitions, Acronyms, and Abbreviations**

* **HMS**: Hospital Management System
* **In-patient**: Patients admitted to the hospital requiring a room assignment
* **Out-patient**: Patients visiting the hospital for consultation only
* **UI**: User Interface

## **2. Overall Description**

### **2.1 Product Perspective**

The Hospital Management System is a standalone web-based application that will be used within a hospital environment. It integrates multiple modules to provide comprehensive hospital management functionality.

### **2.2 Product Functions**

The primary functions of the HMS include:

* User authentication and authorization
* Patient registration and management
* Appointment scheduling
* Doctor assignment
* Prescription management
* Medicine inventory management
* Billing and charging

### **2.3 User Classes and Characteristics**

#### **2.3.1 Receptionist (Admin)**

The receptionist serves as the system administrator with the following responsibilities:

* Schedule appointments for in-patients and out-patients
* Enter and manage patient details
* Assign doctors to patients based on symptoms
* Allocate rooms for in-patients
* Add new doctors to the system

#### **2.3.2 Doctor**

Doctors are responsible for:

* Managing their appointments
* Accepting or rejecting appointment requests
* Prescribing medicines for patients

#### **2.3.3 Store Manager**

The store manager is responsible for:

* Managing medicine inventory
* Processing prescriptions
* Generating bills for prescribed medicines
* Monitoring medicine stock levels

### **2.4 Operating Environment**

* Web-based application
* Compatible with standard web browsers (Chrome, Firefox, Safari, Edge)
* Secure database for storing sensitive medical and personal information

### **2.5 Design and Implementation Constraints**

* The system must comply with healthcare data protection regulations
* The system must ensure secure storage of user credentials
* The interface should be intuitive and accessible to non-technical users

### **2.6 Assumptions and Dependencies**

* Users have basic computer literacy
* Reliable internet connection is available
* The system will have access to a secure and reliable database

## **3. Specific Requirements**

### **3.1 External Interface Requirements**

#### **3.1.1 User Interfaces**

1. **Login Page**
   * Email and password fields
   * Login button
   * Sign up option
2. **Receptionist Dashboard**
   * Patient registration form
   * Appointment scheduling interface
   * Doctor management section
   * Room allocation system
3. **Doctor Dashboard**
   * List of assigned patients
   * Appointment management interface
   * Prescription creation form
4. **Store Manager Dashboard**
   * Prescription fulfillment interface
   * Medicine inventory management
   * Billing generation
   * Low stock notifications

#### **3.1.2 Hardware Interfaces**

* Standard input devices (keyboard, mouse)
* Display screen with minimum resolution of 1280x720
* Network connection for data transmission

#### **3.1.3 Software Interfaces**

* Database Management System
* Web server
* Web browser

### **3.2 Functional Requirements**

#### **3.2.1 Authentication Module**

##### **3.2.1.1 User Login**

* **Description**: The system shall allow users to log in using their email and password.
* **Inputs**: Email, password
* **Processing**: Validate credentials against stored user data
* **Outputs**: Access to the appropriate dashboard based on user role
* **Error Handling**: Display appropriate error messages for invalid credentials

##### **3.2.1.2 User Registration**

* **Description**: The system shall allow new user registration with appropriate role assignment.
* **Inputs**: Name, email, password, role
* **Processing**: Validate input data and store in the database
* **Outputs**: Confirmation message and redirect to login page
* **Error Handling**: Display appropriate error messages for invalid inputs or existing email addresses

##### **3.2.1.3 Logout**

* **Description**: The system shall allow users to securely log out from any page.
* **Inputs**: User click on logout button
* **Processing**: Terminate user session
* **Outputs**: Redirect to login page
* **Error Handling**: N/A

#### **3.2.2 Receptionist Module**

##### **3.2.2.1 Patient Registration**

* **Description**: The system shall allow the receptionist to register new patients.
* **Inputs**: Patient details (name, age, gender, contact information)
* **Processing**: Validate and store patient information
* **Outputs**: Confirmation message and patient ID
* **Error Handling**: Display appropriate error messages for invalid inputs

##### **3.2.2.2 In-Patient Management**

* **Description**: The system shall allow the receptionist to manage in-patients.
* **Inputs**: Patient details, symptoms, doctor assignment, room allocation
* **Processing**: Create patient record, assign doctor, allocate room
* **Outputs**: Confirmation of in-patient registration
* **Error Handling**: Display appropriate error messages for unavailable doctors or rooms

##### **3.2.2.3 Out-Patient Management**

* **Description**: The system shall allow the receptionist to manage out-patients.
* **Inputs**: Patient details, symptoms, doctor assignment
* **Processing**: Create patient record, assign doctor
* **Outputs**: Confirmation of out-patient registration
* **Error Handling**: Display appropriate error messages for unavailable doctors

##### **3.2.2.4 Doctor Management**

* **Description**: The system shall allow the receptionist to add new doctors.
* **Inputs**: Doctor details (name, specialization, contact information)
* **Processing**: Validate and store doctor information
* **Outputs**: Confirmation message and doctor ID
* **Error Handling**: Display appropriate error messages for invalid inputs

##### **3.2.2.5 Appointment Scheduling**

* **Description**: The system shall allow the receptionist to schedule patient appointments.
* **Inputs**: Patient ID, doctor ID, date, time
* **Processing**: Check doctor availability and create appointment
* **Outputs**: Confirmation of appointment
* **Error Handling**: Display appropriate error messages for scheduling conflicts

#### **3.2.3 Doctor Module**

##### **3.2.3.1 View Assigned Patients**

* **Description**: The system shall allow doctors to view their assigned patients.
* **Inputs**: Doctor ID (from login session)
* **Processing**: Retrieve assigned patients from database
* **Outputs**: List of assigned patients
* **Error Handling**: Display message if no patients are assigned

##### **3.2.3.2 Appointment Management**

* **Description**: The system shall allow doctors to accept or reject appointments.
* **Inputs**: Appointment ID, decision (accept/reject)
* **Processing**: Update appointment status in database
* **Outputs**: Confirmation message
* **Error Handling**: N/A

##### **3.2.3.3 Prescription Creation**

* **Description**: The system shall allow doctors to create prescriptions.
* **Inputs**: Patient ID, list of medicines, dosage, duration
* **Processing**: Create prescription record
* **Outputs**: Prescription confirmation
* **Error Handling**: Display appropriate error messages for invalid inputs

#### **3.2.4 Store Manager Module**

##### **3.2.4.1 View Prescriptions**

* **Description**: The system shall allow the store manager to view prescriptions.
* **Inputs**: Store manager ID (from login session)
* **Processing**: Retrieve prescriptions from database
* **Outputs**: List of prescriptions
* **Error Handling**: Display message if no prescriptions are available

##### **3.2.4.2 Process Prescriptions**

* **Description**: The system shall allow the store manager to process prescriptions.
* **Inputs**: Prescription ID
* **Processing**: Update medicine inventory, generate bill
* **Outputs**: Bill with charges
* **Error Handling**: Display appropriate error messages for insufficient stock

##### **3.2.4.3 Inventory Management**

* **Description**: The system shall allow the store manager to manage medicine inventory.
* **Inputs**: Medicine details, quantity changes
* **Processing**: Update inventory records
* **Outputs**: Updated inventory status
* **Error Handling**: N/A

##### **3.2.4.4 Low Stock Notification**

* **Description**: The system shall notify the store manager of low stock items.
* **Inputs**: Inventory status (automatic check)
* **Processing**: Compare current stock with threshold values
* **Outputs**: Notification for items with stock below threshold
* **Error Handling**: N/A

### **3.3 Non-Functional Requirements**

#### **3.3.1 Performance Requirements**

* The system shall load pages within 3 seconds
* The system shall support at least 100 concurrent users
* Database transactions shall complete within 2 seconds

#### **3.3.2 Security Requirements**

* User passwords shall be stored using secure hashing algorithms
* Access to patient data shall be restricted based on user roles
* The system shall automatically log out inactive users after 15 minutes
* All data transmission shall be encrypted

#### **3.3.3 Reliability Requirements**

* The system shall have an uptime of at least 99.5%
* The system shall perform regular data backups
* The system shall provide error recovery mechanisms

#### **3.3.4 Usability Requirements**

* The user interface shall be intuitive and user-friendly
* The system shall provide help documentation
* Error messages shall be clear and informative

#### **3.3.5 Maintainability Requirements**

* The system shall be modular to facilitate future enhancements
* The code shall be well-documented
* The system shall log errors for debugging purposes

## **4. Data Requirements**

### **4.1 Logical Data Model**

The system shall maintain the following data entities:

* Users (Receptionist, Doctor, Store Manager)
* Patients (In-patient, Out-patient)
* Appointments
* Prescriptions
* Medicines
* Rooms

### **4.2 Data Dictionary**

#### **4.2.1 User Data**

* UserID (Primary Key)
* Name
* Email
* Password (hashed)
* Role (Receptionist, Doctor, Store Manager)
* Contact Information

#### **4.2.2 Patient Data**

* PatientID (Primary Key)
* Name
* Age
* Gender
* Contact Information
* Type (In-patient, Out-patient)
* Symptoms
* AssignedDoctorID (Foreign Key)
* RoomID (Foreign Key, for in-patients only)
* Registration Date

#### **4.2.3 Appointment Data**

* AppointmentID (Primary Key)
* PatientID (Foreign Key)
* DoctorID (Foreign Key)
* Date and Time
* Status (Scheduled, Accepted, Rejected, Completed)

#### **4.2.4 Prescription Data**

* PrescriptionID (Primary Key)
* PatientID (Foreign Key)
* DoctorID (Foreign Key)
* Date
* Status (New, Processed)

#### **4.2.5 Prescription Item Data**

* ItemID (Primary Key)
* PrescriptionID (Foreign Key)
* MedicineID (Foreign Key)
* Dosage
* Duration
* Instructions

#### **4.2.6 Medicine Data**

* MedicineID (Primary Key)
* Name
* Description
* Current Stock
* Minimum Stock Threshold
* Price

#### **4.2.7 Room Data**

* RoomID (Primary Key)
* Room Number
* Type
* Status (Available, Occupied)
* Rate per Day

## **5. System Architecture**

### **5.1 High-Level Architecture**

The Hospital Management System will follow a three-tier architecture:

1. **Presentation Layer**: User interfaces for different roles
2. **Application Layer**: Business logic and functionality
3. **Data Layer**: Database management and storage

### **5.2 Component Diagram**

The system will consist of the following components:

* Authentication Component
* Receptionist Component
* Doctor Component
* Store Manager Component
* Database Component

## **6. Verification and Validation**

### **6.1 Testing Approach**

* Unit Testing for individual components
* Integration Testing for component interactions
* System Testing for the complete application
* User Acceptance Testing with stakeholders

### **6.2 Validation Methods**

* Requirements review with stakeholders
* Prototype demonstration
* User feedback collection
* Compliance verification

## **7. Appendices**

### **7.1 Glossary**

* **Hospital Management System (HMS)**: A software system designed to manage hospital operations
* **In-patient**: A patient who is admitted to the hospital and assigned a room
* **Out-patient**: A patient who visits the hospital for consultation without being admitted
* **Prescription**: A written instruction from a doctor for medication
* **Inventory**: The stock of medicines available in the hospital store

### **7.2 References**

* Healthcare Data Protection Regulations
* Standard Web Application Development Practices
* User Interface Design Guidelines