# Hospital management system

**1. Introduction**

**1.1. Project Overview**

The Hospital Management System (HMS) is designed to streamline the management of hospital operations by automating processes such as patient registration, appointment scheduling, billing, medical records management, Pharmacy. This system aims to improve the efficiency and quality of healthcare services.

**1.2. Objectives**

* To digitize hospital records, reducing paperwork and manual errors.
* To enhance patient care by providing real-time access to patient information.
* To facilitate efficient management of hospital resources, including staff.

**1.3. Scope**

The system will cover the following areas:

* Patient Registration and Management
* Appointment Scheduling
* Doctor and Staff Management
* Billing and Payment Processing
* Medical Records Management
* Pharmacy
* Role-Based Access Control

**2. Functional Requirements**

**2.1. Patient Management**

* **Registration**: Patients must register by providing personal details, contact information, and medical history.
* **Profile Management**: Patients must update their profiles, including contact details and insurance information.
* **Medical Records**: The system must store and retrieve patients' medical records, including diagnoses, and lab results.
* **Appointment History**: Patients must view their appointment history and upcoming appointments.

**2.2. Appointment Scheduling**

* **Booking Appointments**: Patients must book appointments with doctors online or through the hospital's reception.
* **Doctor Availability**: The system must display doctors' availability, allowing patients to choose suitable times.
* **Rescheduling/Cancellation**: Patients must reschedule or cancel appointments with automatic notification to the doctor.

**2.3. Doctor and Staff Management**

* **Doctor Profiles**: Doctors must manage their profiles, including specializations, availability, and contact information.
* **Staff Management**: Administrators must manage staff profiles, including nurses, technicians, and administrative personnel.
* **Duty Rosters**: The system must generate and manage duty rosters for doctors and staff.

**2.4. Billing and Payment Processing**

* **Invoice Generation**: The system must generate invoices for consultations, treatments, lab tests, and medications.
* **Payment Gateway Integration**: The system must integrate with payment gateways to process online payments.
* **Billing History**: Patients must view their billing history and print receipts.

**2.5. Medical Records Management**

* **Electronic Health Records (EHR)**: The system must store patients' medical records digitally, including prescriptions, lab results, and treatment plans.
* **Access Control**: Doctors and authorized staff must access and update medical records, with audit trails to track changes.

**2.6. Pharmacy Management**

* **Prescription Management**: Doctors must create digital prescriptions that can be sent directly to the hospital pharmacy.

**2.8. Role-Based Access Control (RBAC)**

* **User Roles**: The system must define user roles such as Admin, Doctor, Nurse, Receptionist, and Patient, each with specific permissions.
* **Access Control**: The system must restrict access to sensitive information based on user roles.
* **Audit Logs**: The system must maintain audit logs of user activities, particularly changes to medical records and billing information.

**3. Non-Functional Requirements**

**3.1. Performance**

* **Response Time**: The system must provide a response time of under 3 seconds for all user interactions.
* **Scalability**: The system must support the growth of hospital operations, handling an increasing number of patients, staff, and records.

**3.2. Security**

* **Data Encryption**: All sensitive data, including patient records and payment information, must be encrypted both in transit and at rest.
* **Authentication**: The system must use secure authentication methods, including multi-factor authentication (MFA) for critical access.

**3.3. Usability**

* **User Interface**: The system must have an intuitive, user-friendly interface for patients, doctors, and administrative staff.
* **Accessibility**: The system must be accessible to users with disabilities, adhering to relevant accessibility standards.

**3.4. Availability**

* **Uptime**: The system must ensure 99.9% uptime, with minimal downtime for maintenance.
* **Disaster Recovery**: The system must implement a disaster recovery plan, including regular backups and failover mechanisms.

**4. User Stories**

**4.1. Patient**

* As a patient, I want to book an appointment with my doctor online so that I can choose a convenient time.
* As a patient, I want to view my medical records so that I can track my treatment history.
* As a patient, I want to pay my bills online so that I don't have to visit the hospital just to settle payments.

**4.2. Doctor**

* As a doctor, I want to access my patients' medical records so that I can provide accurate diagnoses and treatment plans.
* As a doctor, I want to update my availability in the system so that patients can book appointments when I am free.
* As a doctor, I want to generate and send digital prescriptions to the pharmacy so that patients can get their medications quickly.

**4.3. Administrator**

* As an administrator, I want to manage hospital staff schedules so that we can always ensure adequate coverage.
* As an administrator, I want to generate reports on hospital performance so that we can make informed decisions.

**5. Acceptance Criteria**

**5.1. Patient Management**

* Patients must successfully register, log in, and update their profiles without errors.
* Medical records must be securely stored and accessible only to authorized users.

**5.2. Appointment Scheduling**

* Patients must book, reschedule, and cancel appointments seamlessly, with automatic updates sent to doctors.
* Doctors' availability must be accurately reflected in the system.

**5.3. Billing and Payments**

* Invoices must be generated accurately, reflecting all the services provided.
* Online payments must be processed securely, with immediate confirmation to patients.

**5.4. Security**

* All user data, including medical records and payment information, must be encrypted.
* Unauthorized access attempts must be blocked, and audit logs maintained for all sensitive actions.
* enhancements.

## **6** Key Entities

1. **Patient entity can descript by:**
   * PatientID (Primary Key)
   * FirstName
   * LastName
   * age
   * DateOfBirth
   * Gender
   * ContactDetails 🡪(phone, email)
   * Address
2. **Doctor**
   * DoctorID (Primary Key)
   * FirstName
   * LastName
   * Specialization
   * Availability
   * ContactDetails 🡪(phone, email)
   * ProfileInformation
3. **Appointment** 
   * AppointmentID (Primary Key)
   * AppointmentDate
   * AppointmentTime
   * AppointmentStatus
4. **MedicalRecord**
   * RecordID (Primary Key)
   * Diagnosis
   * Created Date
   * TreatmentPlans
   * RefillInformation: Details on whether the prescription can be refilled and how many times
   * LabResults
5. **Prescription**:
   * PrescriptionID (Primary Key)
   * Dosage
   * DateIssued
   * duration
6. Medication:
   * MedicationCode (Primary Key)
   * Name
   * DosageForm (e.g., tablet, capsule)
   * Strength (e.g., 500 mg
   * Manufacturer
   * Price
7. ActiveSubstances:
   * ActiveSubstanceID
   * Name
   * Side Effects 🡪 multi-values
8. **Invoice**:
   * InvoiceID (Primary Key)
   * InvoiceDate
   * TotalAmount
   * PaymentStatus
9. **Staff**:
   * StaffID (Primary Key)
   * FirstName
   * LastName
   * ContactDetails
10. **Role**
    * RoleID (Primary Key)
    * RoleName (e.g., Admin, Doctor, Nurse, Patient)
    * Permissions
11. Pharmacy
    * PharmcyID (Primary Key)
    * Name
    * Contact Details 🡪(phone, email)
12. User
    * UserID (Primary Key)
    * Username
    * Password (hashed and securely stored)
    * ContactDetails 🡪(phone, email)
13. **DutyRosters**
    * RosterID (Primary Key)
    * ShiftStart
    * ShiftEnd
    * DutyDate
14. AuditLog
    * LogID (Primary Key)
    * Timestamp
    * ActionType (e.g., CREATE, UPDATE, DELETE)
    * EntityType (e.g., Patient, Appointment, MedicalRecord)

## Relationships

1. A Patients can have multiple appointments
2. patient can have many prescriptions
3. A patient can be allergic to specific active substances. (**Allergy-Active Substance**)
4. A single patient can have multiple invoices over time.
5. A doctor can have multiple appointments, and each appointment is with one doctor.
6. A doctor manages the medical records of the patients they treat.
7. A doctor can issue multiple prescriptions.
8. A single medical record can have multiple prescriptions.
9. A prescription specifies a particular ActiveSubstance
10. A medication can contain one or more active substances
11. Each prescription is associated with a medical record that has a Prescription Date.
12. Multiple prescriptions can be associated with a pharmacy.
13. A staff member can have multiple entries in the duty roster.
14. A doctor can have multiple entries in the duty roster.
15. A user can have multiple audit log entries.
16. multiple users can share the same role.
17. Each staff member has a user account.
18. Each doctor has a user account.
19. Each patient is associated with a single user account.
20. ActiveSubstance interaction whit other ActiveSubstance descript by interactionDegrea