**Hospital Management System (HMS):**

**Problem Statement**

Hospital Management System (HMS): A large hospital needs software to automate its various book-keeping activities. The hospital has a set of consultation rooms, and a set of indoor rooms (called cabins) for resident patients. There are two types of doctors: regular and visiting consultants. The regular doctors sit in a consultation room for six hours every day. During 8 AM to 8 PM every day, at least three regular doctors are present. During 8 PM to 8 AM every day, at least one regular doctor is present. The software should provide facilities for the hospital administrator to input the names of the regular doctors employed by the hospital and their offered basic pay. The doctors also have a variable pay that is computed by multiplying the total number of patients they see over a month, multiplied by 100. The administrator should be able to generate pay slips for the doctors at the end of every month. The administrator should be able to delete the names of specific regular doctors who tender their resignation. The administrator also should be able to change the basic pay of a doctor. The administrator should be able to prepare a weekly roster for the doctors. The roster contains the visiting hours for every doctor and the specific consultation room allotted to them for each day. The roster once prepared by the administrator is automatically uploaded at the hospital web site and mails are sent to the doctors regarding their specific duty hours. The visiting consultants come for two-hour slots twice a week and their roster is also prepared by the administrator. The visiting consultants do not get basic pay but get Rs. 200 for every patient that they see. The patients can create their log-in account at the hospital website. The patients can see the specific doctors available for any day or over a week, and their qualifications and expertise. The patient can then proceed for booking an appointment with the doctor. The free slots available for the doctor would get displayed to him and he can book a slot of his convenience. But, for freezing the slot, he should pay a fee of Rs. 500 on-line. Once the patient visits the doctor at the appointed time, the doctor’s diagnosis and prescriptions are stored on-line for future reference. In a similar fashion, the patients can book the indoor residence rooms (cabins) depending on availability. But, before they can freeze their booking, they would have to pay Rs. 3000 per day of booking on-line. There are also a set of nurses, who get fixed salary. The names of the nurses and their salary is entered by the administrator. The administrator should be able to delete the names of specific nurses or change their salary. The duty allocation to the nurses for one week at a time is done by a head nurse. At any time, the administrator should be able to see the total number of patients who have visited the hospital for any given period, the total income, and the total salary outgo so far.

**SRS (Software Requirements Specification) document**

# Hospital Management System (HMS)

## Version 1.0

Date: 05/05/2025  
Author: Group11

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 30/01/2025 | 1.0 | SRS Document | Group11 |

## Table of Contents

1. Introduction

2. Overall Description

3. Specific Requirements

4. External Interface Requirements

5. Non-Functional Requirements

6. Supporting Information

# 1. Introduction

## 1.1 Purpose

This document defines the software requirements for the Hospital Management System (HMS), intended to support administrative, financial, and medical functions in a hospital environment. It serves as a reference for developers, testers, and stakeholders.

## 1.2 Scope

The HMS enables:  
- Patient registration, appointment scheduling, and online payments.  
- Doctor onboarding, rostering, and payroll management.  
- Nurse management including salary records and duty allocations.  
- Admin functionality for generating reports, managing users, and financials.  
- Secure web access with mobile-friendly user interface.

## 1.3 Definitions, Acronyms

HMS: Hospital Management System  
DB: Database  
GUI: Graphical User Interface  
Admin: System administrator

## 1.4 References

Hospital Workflow Guidelines  
Django REST Framework Documentation  
Payment Gateway API Docs

## 1.5 Overview

The document outlines functional modules, system behavior, constraints, interface specifications, and system architecture for HMS.

# 2. Overall Description

## 2.1 Product Perspective

This is a standalone web-based system using Django (backend), HTML/CSS (frontend), and MySQL (database).

## 2.2 User Classes

- Administrator: Manages users, reports, and configuration.  
- Doctor: Views appointments and updates medical records.  
- Patient: Books appointments, makes payments, views health history.  
- Nurse: Views duty schedule and assigned patient records.

## 2.3 Operating Environment

OS: Windows/Linux server  
Browser: Chrome, Firefox, Edge  
Backend: Django 4.x  
DB: MySQL 8.x

## 2.4 Design/Implementation Constraints

Web-based only  
Must comply with medical data security standards (HIPAA-like principles)

# 3. Specific Requirements

## 3.1.1 Patient Management

1. Register new patients with contact and demographic info.
2. Book appointments with doctors.
3. View and download prescriptions and diagnosis history.
4. Pay booking fee (INR 500) and room fee (INR 3000/day) online.

## 3.1.2 Doctor Management

1. Add/edit/delete regular and visiting doctors.
2. Manage payroll based on consultation count.
3. View roster and appointments.

## 3.1.3 Appointment Scheduling

1. Display doctor availability calendar.
2. Allow patients to book slots.
3. Notify doctors of bookings via email.

## 3.1.4 Room Booking

1. Allow admin to add/edit/delete rooms.
2. Patients can book rooms and pay online.
3. View available rooms with booking status.

## 3.1.5 Nurse Management

1. Add/edit/delete nurse records.
2. Generate duty schedules per week.

## 3.1.6 Financial & Administrative Reporting

1. View total income from appointments and rooms.
2. Display payroll expenses for doctors and nurses.
3. Export reports in PDF/Excel.

# 4. External Interface Requirements

## 4.1 User Interfaces

Web-based UI for desktop and mobile  
Role-based navigation menus  
Forms for booking, editing records, viewing reports

## 4.2 Hardware Interfaces

Server machine for deployment  
Client devices (PCs, tablets, phones)

## 4.3 Software Interfaces

Payment Gateway   
SMTP Email for notifications

## 4.4 Communication Interfaces

HTTPS protocol  
REST API for data access

# 5. Non-Functional Requirements

## 5.1 Performance

1. Up to 500 concurrent users
2. Max 3 seconds for appointment confirmation

## 5.2 Security

1. Role-based access
2. Encrypted password and patient data
3. CSRF and SQL injection protection

## 5.3 Usability

1. Accessible via desktop and mobile
2. Responsive design
3. Help and FAQ section available

## 5.4 Reliability

1. Daily DB backup
2. 99.9% uptime goal

## 5.5 Maintainability

1. Modular codebase (MVC)
2. Well-documented APIs and functions

# 6. Supporting Information

Use Case Diagrams  
ER Diagram  
Data Flow Diagrams  
Deployment Guide  
Test Plan Document (to be drafted separately)

**DFD (Data Flow Diagram) model**

**LEVEL 0:**

**A diagram of a diagram

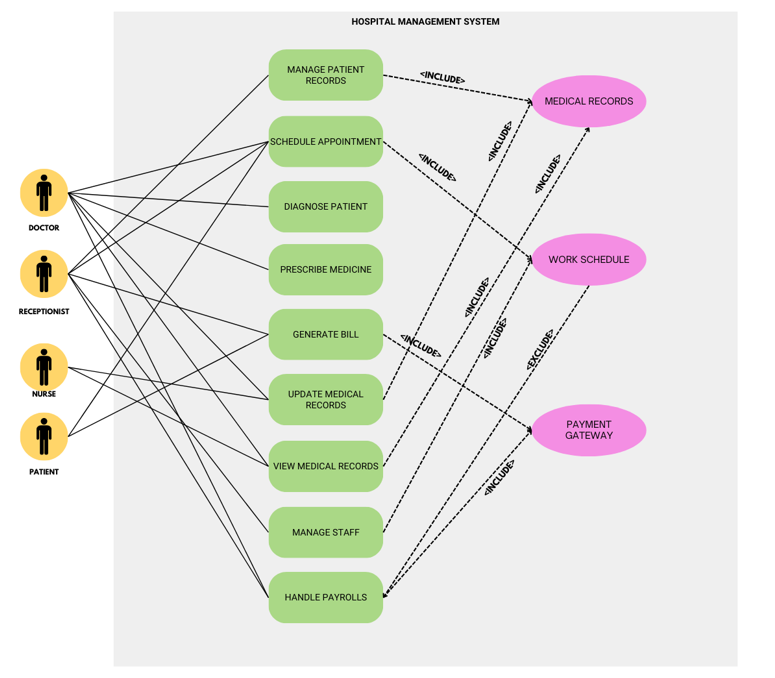
AI-generated content may be incorrect.**

**LEVEL 1:**

**A diagram of a diagram

AI-generated content may be incorrect.**

**Use Case Diagram:**

****

USE CASE DOCUMENTATION

**HOSPITAL MANAGEMENT SYSTEM**

|  |  |
| --- | --- |
| **ID:** | **UC-01** |
| **Title:** | Manage Patient Records |
| **Description:** | Admin or Receptionist can add, update, or delete patient records in the system. |
| **Primary Actor:** | Admin, Receptionist |
| **Secondary Actor:** | None |
| **Preconditions:** | User must be logged in with appropriate permissions. |
| **Postconditions:** | Patient records are updated in the system. |
| **Dependency** | None |
| **Generalization** | None |
| **Main  Success Scenario:** | 1. Admin/Receptionist selects "Manage Patient Records." 2. System displays patient records. 3. Admin/Receptionist adds, updates, or deletes records.   System saves changes. |
| **Extensions or**  **Alternate Flow:** | If invalid data is entered, the system displays an error message. |
| **Frequency of Use:** | High |
| **Status:** | Completed |
| **Owner:** | Admin Team |
| **Priority:** | High |

|  |  |
| --- | --- |
| **ID:** | **UC-02** |
| **Title:** | Schedule Appointment |
| **Description:** | Patient or Receptionist schedules an appointment with a doctor. |
| **Primary Actor:** | Patient, Receptionist |
| **Secondary Actor:** | None |
| **Preconditions:** | Patient must be registered in the system. |
| **Postconditions:** | Appointment is scheduled and added to the system. . |
| **Dependency** | None |
| **Generalization** | None |
| **Main  Success Scenario:** | 1. Patient/Receptionist selects "Schedule Appointment." 2. System displays available slots. 3. Patient/Receptionist selects a slot and confirms. 4. System updates the appointment schedule.. |
| **Extensions or**  **Alternate Flow:** | If no slots are available, the system notifies the user. |
| **Frequency of Use:** | High |
| **Status:** | Completed |
| **Owner:** | Receptionist Team |
| **Priority:** | High |

|  |  |
| --- | --- |
| **ID:** | **UC-03** |
| **Title:** | Diagnose Patient |
| **Description:** | Doctor diagnoses a patient based on symptoms and test results.. |
| **Primary Actor:** | Doctor |
| **Secondary Actor:** | None |
| **Preconditions:** | Patient must have an active appointment. |
| **Postconditions:** | Diagnosis is recorded in the patient's medical history.. . |
| **Dependency** | None |
| **Generalization** | None |
| **Main  Success Scenario:** | 1. Doctor selects "Diagnose Patient." 2. System displays patient details and test results. 3. Doctor enters diagnosis and saves. 4. System updates patient's medical history. . |
| **Extensions or**  **Alternate Flow:** | If test results are incomplete, the system prompts the doctor to wait. |
| **Frequency of Use:** | Medium |
| **Status:** | Completed |
| **Owner:** | Doctor team |
| **Priority:** | High |

|  |  |
| --- | --- |
| **ID:** | **UC-04** |
| **Title:** | Prescribe Medicine |
| **Description:** | Doctor prescribes medicine based on the diagnosis. |
| **Primary Actor:** | Doctor |
| **Secondary Actor:** |  |
| **Preconditions:** | Diagnosis must be completed. |
| **Postconditions:** | Prescription is added to the patient's record. . |
| **Dependency** | None |
| **Generalization** | None |
| **Main  Success Scenario:** | 1. Doctor selects "Prescribe Medicine." 2. System displays available medicines. 3. Doctor selects medicine and dosage. 4. System updates patient's prescription record.   . |
| **Extensions or**  **Alternate Flow:** | If no slots are available, the system notifies the user. |
| **Frequency of Use:** | Medium |
| **Status:** | Completed |
| **Owner:** | Doctor Team |
| **Priority:** | High |

|  |  |
| --- | --- |
| **ID:** | **UC-05** |
| **Title:** | Generate Bill |
| **Description:** | Receptionist generates a bill for the patient's treatment. |
| **Primary Actor:** | Receptionist |
| **Secondary Actor:** | None |
| **Preconditions:** | Treatment must be completed.. |
| **Postconditions:** | Bill is generated and sent to the patient. . . |
| **Dependency** | None |
| **Generalization** | None |
| **Main  Success Scenario:** | 1. Receptionist selects "Generate Bill." 2. System calculates the total cost based on treatment and medicines. 3. Receptionist confirms and prints the bill. 4. System updates billing records. |
| **Extensions or**  **Alternate Flow:** | If payment is pending, the system marks the bill as unpaid |
| **Frequency of Use:** | High |
| **Status:** | Completed |
| **Owner:** | Receptionist Team |
| **Priority:** | High |

|  |  |
| --- | --- |
| **ID:** | **UC-06** |
| **Title:** | Update Medical Records |
| **Description:** | Updates test results for a patient. |
| **Primary Actor:** | Nurse, Doctor |
| **Secondary Actor:** | None |
| **Preconditions:** | Test must be completed. |
| **Postconditions:** | Test results are updated in the patient's record.. |
| **Dependency** | None |
| **Generalization** | None |
| **Main  Success Scenario:** | 1. selects "Update Test Results." 2. System displays pending tests. 3. enters results and saves.   4. System updates patient's medical history. . |
| **Extensions or**  **Alternate Flow:** | If results are abnormal, the system alerts the doctor. |
| **Frequency of Use:** | Medium |
| **Status:** | Completed |
| **Owner:** | Team |
| **Priority:** | Medium |

|  |  |
| --- | --- |
| **ID:** | **UC-07** |
| **Title:** | View Medical History |
| **Description:** | Patient or Doctor views the patient's medical history. |
| **Primary Actor:** | Patient, Doctor |
| **Secondary Actor:** | None |
| **Preconditions:** | Patient must be registered in the system. |
| **Postconditions:** | Medical history is displayed. . |
| **Dependency** | None |
| **Generalization** | None |
| **Main  Success Scenario:** | 1. Patient/Doctor selects "View Medical History." 2. System displays medical records. 3. User can view details or print records. |
| **Extensions or**  **Alternate Flow:** | If records are incomplete, the system notifies the user. |
| **Frequency of Use:** | High |
| **Status:** | Completed |
| **Owner:** | Patient/Doctor Team |
| **Priority:** | High |

|  |  |
| --- | --- |
| **ID:** | **UC-08** |
| **Title:** | Manage Staff |
| **Description:** | Admin adds, updates, or deletes staff records. |
| **Primary Actor:** | Admin |
| **Secondary Actor:** | None |
| **Preconditions:** | Admin must be logged in. |
| **Postconditions:** | Staff records are updated. . |
| **Dependency** | None |
| **Generalization** | None |
| **Main  Success Scenario:** | 1. Admin selects "Manage Staff." 2. System displays staff records. 3. Admin adds, updates, or deletes records. 4. System saves changes. . |
| **Extensions or**  **Alternate Flow:** | If invalid data is entered, the system displays an error message. |
| **Frequency of Use:** | Low |
| **Status:** | Completed |
| **Owner:** | Admin Team |
| **Priority:** | Medium |

|  |  |
| --- | --- |
| **ID:** | **UC-09** |
| **Title:** | Handle Billing |
| **Description:** | Receptionist handles billing and payment processing. |
| **Primary Actor:** | Receptionist |
| **Secondary Actor:** | None |
| **Preconditions:** | Bill must be generated. |
| **Postconditions:** | Payment is recorded in the system. . |
| **Dependency** | None |
| **Generalization** | None |
| **Main  Success Scenario:** | 1. Receptionist selects "Handle Billing." 2. System displays pending bills. 3. Receptionist processes payment. 4. System updates payment records. . |
| **Extensions or**  **Alternate Flow:** | If payment fails, the system marks the bill as unpaid. |
| **Frequency of Use:** | High |
| **Status:** | Completed |
| **Owner:** | Receptionist Team |
| **Priority:** | High |