



BANGLADESH UNIVERSITY OF PROFESSIONALS

FACULTY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (CSE)

Final Report

Submitted By:

Name	Roll
Mohaiminul Raju	2252421020
Shamoyeta Mourin Mouly	2252421036
Latifa Nishat Nishi	2252421062
Tahsina Tabassum Roza	2252421084
Raiyan Sarwar	2252421096

Section : B

Semester : 6th

Course Name : Software Engineering Laboratory

Course Code : CSE-3206

Date of Submission: 13.06.2025

Signature of Teacher

Table of Contents

CHAPTER 1: Introduction	4
1.1 Project Overview	4
1.2 Objective	4
1.3 Scope of the Project	5
CHAPTER 2: Requirement Analysis.....	6
2.1 Functional Requirements	6
2.1.1 Appointment & Scheduling System	6
2.1.2 Patient History / Electronic Health Record (EHR) Management	9
2.1.3 Medication Store	12
2.1.4 Diagnostic Services.....	14
2.1.5 Ambulance Booking	15
2.1.6 Role-Based Login	16
2.1.7 Staff Management	17
2.1.8 Room Allocation	18
2.1.9 Prescription Management	19
2.1.10 Billing and Payment System	19
2.2 Non-functional Requirements	20
2.2.1 Performance Requirements	21
2.2.2 Safety Requirements	21
2.2.3 Security Requirements	22
2.2.4 Software Quality Attributes	22
2.2.5 Business Rules	23
CHAPTER 3: System Design	25
3.1 UML Use Case Diagram.....	25
3.2 Extended Use Cases	28
3.3 UML Class Diagram	34
3.4 Data Flow Diagram (DFD)	39
3.5 Database Schema	43

CHAPTER 4: Tools Used	48
4.1 Trello	48
4.2 Discord.....	49
4.3 UML Diagram Tools	49
4.4 UI Design Tools	50
CHAPTER 5: Implementation Detail (UI design and Screenshots of UI)	52
CHAPTER 6: Testing.....	59
1. Appointment Booking Test Suite	59
2. Patient History Test Suite.....	62
3. Staff Management Test Suite	64
4. Room Allocation Test Suite	65
5. Medication Store Test Suite	67
6. Diagnostic Services Test Suite.....	68
7. Role-Based Login Test Suite.....	70
8. Ambulance Booking Test Suite.....	71
9. Prescription Management Test Suite.....	73
10. Billing & Payment System Test Suite.....	75
CHAPTER 7: Conclusion	77
References	78
Contribution Report	79

CHAPTER 1: INTRODUCTION

In the rapidly evolving field of healthcare, the efficient management of hospital operations has become increasingly important. The adoption of digital solutions is crucial for addressing the challenges faced by hospitals in maintaining high standards of patient care and operational efficiency. This project presents a comprehensive Hospital Management System aimed at transforming traditional hospital processes through technology.

1.1 Project Overview

A Hospital Management System (HMS) is an integrated information system designed to manage the administrative, financial, and clinical aspects of a hospital. The primary goal of an HMS is to streamline operations, improve patient care, and optimize resource utilization by digitizing and automating day-to-day hospital activities. This system centralizes data management, enabling efficient storage, retrieval, and processing of patient, staff, and hospital information.

With rising demands in healthcare, hospitals are expected to deliver prompt, high-quality services while ensuring accuracy and confidentiality of sensitive medical data. Manual record-keeping often results in inefficiencies, delays, and increased risk of errors. An HMS addresses these challenges by providing a unified platform for managing patient registration, appointments, billing, inventory, and reporting, as well as supporting clinical functions such as diagnosis and treatment tracking.

The Hospital Management System is not only beneficial to hospital administrators, but also to doctors, nurses, receptionists, and patients. It enables seamless coordination among departments, reduces paperwork, speeds up information flow, and supports evidence-based decision-making. Through comprehensive reporting and analytics, management can monitor hospital performance and implement improvements where necessary.

1.2 Objective

The main objectives of the Hospital Management System are:

- **To automate and digitize hospital workflows:** Replace manual, paper-based processes with efficient, computerized operations.
- **To maintain accurate and up-to-date records:** Ensure all patient, staff, and transaction data are securely stored and easily retrievable.
- **To improve patient care:** Facilitate better diagnosis, treatment, and follow-ups by providing healthcare professionals with timely access to complete medical histories.
- **To optimize resource utilization:** Enable better planning and allocation of hospital resources such as beds, staff, and equipment.

- **To enhance administrative efficiency:** Streamline billing, reporting, appointment scheduling, and inventory management.
- **To ensure data privacy and security:** Protect sensitive personal and medical information through robust authentication, authorization, and encryption mechanisms.
- **To provide a scalable and flexible platform:** Allow easy integration of additional modules (e.g., pharmacy, laboratory, radiology) and adaptation to the changing needs of the hospital.

1.3 Scope of the Project

The scope of this Hospital Management System encompasses the following key areas:

- **Patient Management:** Registration, admission, discharge, and maintenance of comprehensive patient records, including medical history, diagnosis, and treatments.
- **Appointment Scheduling:** Efficient handling of appointment bookings, cancellations, and reminders for both patients and doctors.
- **Staff and Doctor Management:** Management of personnel records, schedules, roles, and contact information.
- **Billing and Payments:** Automated generation of bills for consultations, procedures, and services, including payment tracking and receipt generation.
- **Medical Records Management:** Secure storage and retrieval of electronic medical records (EMR), prescriptions, lab results, and other clinical information.
- **Reporting and Analytics:** Generation of various operational and management reports, such as occupancy rates, revenue, patient demographics, and treatment outcomes.
- **User Authentication and Access Control:** Role-based access to system functionalities, ensuring that only authorized users can access or modify sensitive information.
- **System Administration:** Configuration, backup, and maintenance features to ensure system reliability and data integrity.

The project is designed to serve medium to large hospitals and can be customized or extended in the future to support specialized modules like pharmacy management, laboratory integration, or telemedicine. The HMS will be accessible to users through a secure, user-friendly interface, ensuring ease of use for all stakeholders, including hospital management, staff, and patients.

CHAPTER 2: REQUIREMENT ANALYSIS

A thorough requirement analysis lays the foundation for a successful Hospital Management System. Clearly defining what the system should do, how it should perform, and the boundaries within which it must operate ensure that all stakeholders' needs are addressed and that the system is robust, reliable, and adaptable to future changes. The following sections present both the functional and non-functional requirements, articulated to provide maximum clarity, traceability, and verifiability.

2.1 Functional Requirements

Functional requirements describe the specific behaviors and capabilities the Hospital Management System must provide to meet the operational needs of its users. Each requirement below is expressed in a user story format that includes explicit confirmation criteria, ensuring that every feature is both testable and traceable throughout development.

2.1.1 Appointment & Scheduling System

Each requirement below is expressed in a user story format that includes explicit confirmation criteria. Unique IDs (e.g., REQ-A1, REQ-A2) help ensure traceability and completeness.

- **REQ-A1: Patient Online Booking**

User Story: As a patient, I want to book an appointment online so that I can avoid long queues at the hospital.

Confirmation: When a patient submits a booking request with the required details (date, time, doctor), the system reserves the slot, displays a confirmation message with a unique appointment ID, and sends an SMS/email confirmation.

- **REQ-A2: View Real-Time Calendar Slots**

User Story: As a patient, I want to view a real-time calendar of available slots so that I can select a time that fits my schedule.

Confirmation: The system displays an up-to-date calendar reflecting current doctor availability when a patient accesses it.

- **REQ-A3: Search by Specialty, Location, and Rating**

User Story: As a patient, I want to search for doctors by specialty, location, and ratings so that I can choose the best care option.

Confirmation: Upon entering search criteria, the system filters and returns a list of doctors who meet the parameters, along with rating and location details.

- **REQ-A4: Digital Booking Confirmation Notification**

User Story: As a patient, I want to receive a confirmation notification (via email/SMS) after booking so that I know my appointment is confirmed.

Confirmation: After a successful booking, the system sends an automatic SMS/email that includes appointment details and a unique identifier.

- **REQ-A5: Easy Rescheduling/Cancellation**

User Story: As a patient, I want to cancel or reschedule an appointment online so that I can adapt my plans in case of emergencies or conflicts.

Confirmation: When a cancellation or rescheduling request is submitted, the system updates the appointment record, frees or adjusts the slot, and sends notifications confirming the change.

- **REQ-A6: Conflict Alerts**

User Story: As a patient, I want the system to flag scheduling conflicts so that I avoid double-booking my time.

Confirmation: The system automatically checks new booking requests against existing appointments and issues a clear warning if conflicts exist.

- **REQ-A7: Appointment History Review**

User Story: As a patient, I want to review my appointment history, including past visits and consultation notes, so that I have a detailed record of my care.

Confirmation: When accessing the history section, the system shows a chronological list of past appointments with relevant details.

- **REQ-A8: Caregiver Booking for Dependents**

User Story: As a caregiver, I want to book appointments for my dependents using individual profiles so that I manage their care efficiently.

Confirmation: Selecting a dependent's profile during booking results in the appointment being registered under that dependent's record and confirmed via notification.

- **REQ-A9: Real-Time Receptionist Management**

User Story: As a receptionist, I want to manage, edit, and update appointments in real time so that I can accommodate emergency or walk-in patients.

Confirmation: Edits made from the receptionist dashboard immediately update across all system views and trigger notifications as needed.

- **REQ-A10: Doctor's Daily/Weekly Schedule Display**

User Story: As a doctor, I want to view my daily and weekly appointment lists so that I can prepare adequately for each consultation.

Confirmation: The doctor's dashboard displays an accurate segmentation of appointments by day and week, auto-refreshing with any changes.

- **REQ-A11: Doctor Availability & Time Blocking**

User Story: As a doctor, I want to set my available time slots and block off personal time so that my schedule remains accurate and manageable.

Confirmation: The scheduling interface allows doctors to mark unavailable periods, ensuring these slots do not appear as available to patients.

- **REQ-A12: Nurse Pre-Appointment Preparation**
User Story: As a nurse, I want to view the upcoming appointment schedule so that I can prepare patient files and allocate rooms efficiently.
Confirmation: The nurse's dashboard lists upcoming appointments with patient names, doctor assignments, and visit types, updating in real time.
- **REQ-A13: Admin Tracking of No-Shows/Cancellations**
User Story: As an admin, I want to track metrics such as no-shows and cancellations so I can analyze booking trends and improve scheduling efficiency.
Confirmation: The system generates real-time dashboards and periodic reports displaying detailed statistics on cancellations and no-shows.
- **REQ-A14: Automated SMS/Email Reminders**
User Story: As a system, I want to send automated SMS/email reminders to patients ahead of their appointments so that missed visits are minimized.
Confirmation: Automated reminders are dispatched at pre-configured intervals (e.g., 24 hours prior), with each dispatch logged in the system.
- **REQ-A15: Doctor Reschedule Notifications for Patients**
User Story: As a patient, I want to be notified immediately if my doctor cancels or reschedules my appointment so that I can adjust my plans.
Confirmation: Any doctor-initiated change triggers an immediate notification to affected patients with details of the change and available alternatives.
- **REQ-A16: Waitlist Management**
User Story: As a receptionist, I want to manage a waitlist so that when an appointment slot opens unexpectedly, patients on the list are alerted.
Confirmation: When a booking is canceled, the system automatically checks for waitlisted requests and contacts the next eligible patient to fill the slot.
- **REQ-A17: Billing Integration**
User Story: As an admin, I want appointment data to integrate with the billing system so that invoices are generated automatically following an appointment.
Confirmation: At appointment completion, relevant booking data is sent to the billing module, and an invoice is generated and logged.
- **REQ-A18: Accessible Interface for All Users**
User Story: As a patient with accessibility needs, I want the booking system to support screen readers and high-contrast themes so that I can use it effectively.
Confirmation: The interface is tested with standard accessibility tools and adheres to WCAG guidelines to guarantee ease of use.
- **REQ-A19: Cross-Device Responsive Design**
User Story: As a mobile user, I want a responsive design that adapts seamlessly across devices so that I can manage my appointments on the go.

Confirmation: The user interface is verified on multiple resolutions (smartphones, tablets, desktops) to ensure consistent functionality and appearance.

- **REQ-A20: External Calendar Synchronization**

User Story: As an admin, I want the system to synchronize with external calendars (Google, Outlook) so that appointment data remains consistent across platforms.

Confirmation: Appointments created or updated within the system are automatically reflected in linked external calendars via an API integration, with confirmation logs generated.

- **REQ-A21: Analytics for Resource Planning**

User Story: As a system, I want to generate analytics on peak appointment times and doctor workloads so that hospital staffing and resource allocation can be optimized.

Confirmation: The system provides dashboards and exportable reports that include metrics such as appointment frequency and idle times, updated in near real-time.

The above appointment and scheduling requirements collectively ensure that all stakeholders—patients, caregivers, doctors, nurses, receptionists, and administrators—have the tools they need to make and manage appointments efficiently, transparently, and with minimal friction. The next sections outline similar user-story-based requirements for other key hospital modules, ensuring the HMS serves the complete operational workflow of a modern healthcare environment.

2.1.2 Patient History / Electronic Health Record (EHR) Management

Each requirement below is expressed as a user story with clear confirmation criteria. Unique IDs (e.g., REQ-E1, REQ-E2) ensure that all functionalities are fully traceable and verifiable.

- **REQ-E1: Comprehensive Medical Record Access**

User Story: As a doctor, I want to view a patient’s complete history—including diagnoses, prescriptions, and lab results—so that I can make informed treatment decisions.

Confirmation: On selecting a patient record, the system presents a segmented view covering all pertinent aspects (diagnoses, prescriptions, lab reports, and clinical notes).

- **REQ-E2: Patient Self-Access to Records**

User Story: As a patient, I want to access my full medical history online so that I can monitor my health trends and treatment progress.

Confirmation: After successful authentication, the patient’s portal displays a chronological and sectioned view of their health records with easy navigation.

- **REQ-E3: Document Upload Capability**

User Story: As a patient, I want to upload scanned copies of prescriptions, test reports, and other documents so that my record remains comprehensive and up-to-date.

Confirmation: The system accepts standard file formats (e.g., PDF, JPEG), assigns the document to the correct record section, and confirms the successful upload.

- **REQ-E4: Doctor Note Annotation**

User Story: As a doctor, I want to add notes and annotations to a patient's record after each consultation so that the treatment history evolves accurately.

Confirmation: Upon saving, the system timestamps each note and displays it as part of the patient's evolving record, with the option for later editing.

- **REQ-E5: Nurse Daily Updates**

User Story: As a nurse, I want to update patient vitals and daily observations in the EHR so that doctors have the most current data for decision-making.

Confirmation: Nurse inputs are recorded with corresponding date/time stamps and appear in a "Latest Observations" section, with an accompanying audit trail.

- **REQ-E6: Specialist Collaboration**

User Story: As a specialist, I want to review referral notes and previous treatments so that I can deliver consistent care without repeating unnecessary tests.

Confirmation: The system highlights referral summaries and past treatment notes in a dedicated section when the specialist accesses the patient record.

- **REQ-E7: Emergency Information Quick Access**

User Story: As an emergency responder, I want immediate access to critical patient data (e.g., allergies, medications, chronic conditions) so I can act quickly and safely.

Confirmation: Activating the emergency view mode presents a high-priority summary of critical data within three seconds, following the established security protocol.

- **REQ-E8: Role-Based Security for Data Access**

User Story: As a hospital admin, I want to manage access permissions for patient records so that only authorized personnel can view or modify sensitive information.

Confirmation: The system enforces role-based access controls (RBAC) and logs all access events; tests confirm that unauthorized attempts are blocked.

- **REQ-E9: Automated Audit Trail Logging**

User Story: As a system, I want to maintain a timestamped audit log of every access and modification of patient records to ensure accountability and compliance.

Confirmation: Each access or modification is logged with user details and timestamps; audit logs are queryable and exportable by authorized administrators.

- **REQ-E10: Downloadable/Printable Records**

User Story: As a patient, I want the option to download or print my complete medical history so that I have a physical copy for personal use or consultations.

Confirmation: Upon request, the system generates a PDF of the complete record and initiates a download while also notifying the user on-screen.

- **REQ-E11: Mobile EHR Access**

User Story: As a mobile user, I want secure access to my medical records via a dedicated app so that I can review my history while away.

Confirmation: The mobile application offers secure multi-factor authentication and replicates the EHR view consistent with desktop functionality, confirmed via cross-platform testing.

- **REQ-E12: Multimedia Attachments**

User Story: As a doctor, I want to attach digital images (e.g., X-rays, MRIs, CT scans) to patient records so that visual data is easily accessible.

Confirmation: The system supports image uploads, displays them in a dedicated multimedia section with relevant metadata, and confirms successful attachment.

- **REQ-E13: Integrated View for Coordinated Care**

User Story: As a care coordinator, I want a consolidated view that integrates lab results, imaging reports, and clinical notes so that I can design holistic treatment plans.

Confirmation: The EHR interface aggregates data from multiple sources into a filterable summary view, updated in real time.

- **REQ-E14: Selective Data Sharing Options**

User Story: As a patient, I want to control and customize which parts of my history are shared with different providers so that my sensitive information remains private.

Confirmation: The system provides granular privacy settings that let patients toggle sharing options for each record section, with immediate effect and logged changes.

- **REQ-E15: Anonymized Data Access for Research**

User Story: As a researcher, I want access to anonymized patient data (with proper consent) so that I can study health trends without compromising privacy.

Confirmation: The system generates anonymized datasets—stripped of personally identifiable information—for approved research accounts, documented in an access log.

- **REQ-E16: Lab Result Integration**

User Story: As a lab technician, I want to upload test results directly into a patient's EHR so that healthcare providers can access updated information promptly.

Confirmation: After uploading, results are placed in the correct record section, a notification is sent to assigned providers, and a timestamp is logged.

- **REQ-E17: Insurance Verification Interface**

User Story: As an insurance officer, I want access to verified treatment histories (with patient consent) so that claims can be accurately processed.

Confirmation: The system presents a summarized, consent-verified treatment history to insurance users and logs every access event for audit purposes.

- **REQ-E18: Compliance Auditing Interface**

User Story: As a medical auditor, I want read-only access and summarized modification

logs of patient records so that I can verify compliance with healthcare standards.

Confirmation: The auditor interface provides a read-only view of records plus detailed, timestamped logs of all modifications, secured by proper authentication.

- **REQ-E19: Real-Time Patient Notifications on Updates**

User Story: As a patient, I want to receive alerts when new test results or updates are added to my record so that I remain informed about my health.

Confirmation: The system sends an immediate SMS/email alert upon record update, with the notification action logged for confirmation.

- **REQ-E20: Efficient Keyword-Based Search**

User Story: As a doctor, I want to search patient records using relevant keywords (e.g., diagnosis, treatment, medication) so that I can quickly locate key information during consultations.

Confirmation: Entered keywords trigger a search query that returns matching results with highlighted terms within two seconds.

- **REQ-E21: Long-Term Record Retention Policy Enforcement**

User Story: As a hospital admin, I want the system to enforce a long-term data retention policy so that patient records are securely preserved for the required duration (e.g., 10+ years).

Confirmation: The system automatically archives or retains records based on established policies, logs the archival actions, and provides secure retrieval or deletion mechanisms in compliance with regulations.

2.1.3 Medication Store

Each requirement below is expressed as a user story with clear confirmation criteria. Unique IDs (e.g., REQ-M1, REQ-M2) ensure that all functionalities are fully traceable and verifiable.

- **REQ-M1: View Medication Inventory**

User Story: As a pharmacist, I want to view the current inventory so that I can manage stock efficiently.

Confirmation: The system provides an interface showing all available medications, quantities, and expiration dates.

- **REQ-M2: Dispense Medication**

User Story: As a pharmacist, I want to dispense prescribed medications and update the inventory accordingly.

Confirmation: When a prescription is fulfilled, the system deducts the issued quantity and logs the transaction.

- **REQ-M3: Alert for Low Stock**

User Story: As a store admin, I want to receive alerts when any medication stock falls below the minimum threshold so that I can restock timely.

Confirmation: The system sends an automated notification listing low-stock items with reorder suggestions.

- **REQ-M4: Update Inventory Records**

User Story: As a store manager, I want to update medication entries to reflect restocking or removal so that records stay current.

Confirmation: The system logs inventory updates with medication name, batch, and quantity adjusted.

- **REQ-M5: Track Expiry Dates**

User Story: As a pharmacist, I want to track upcoming medication expirations so that I can manage safe disposal.

Confirmation: The system displays medications approaching expiry and generates weekly reports.

- **REQ-M6: Generate Inventory Reports**

User Story: As an admin, I want to generate reports of medication inflow and outflow over time so that I can assess usage patterns.

Confirmation: The system generates downloadable reports based on date, category, or medicine name.

- **REQ-M7: Barcode Scanning for Dispensing**

User Story: As a pharmacist, I want to scan barcodes for medicines during dispensing to ensure accuracy.

Confirmation: The system verifies scanned medicine and quantity against prescription and confirms the match.

- **REQ-M8: Role-Based Access Control**

User Story: As an admin, I want to restrict inventory management access to authorized users only.

Confirmation: Only pharmacists and admins can edit inventory; unauthorized users receive an access denied message.

- **REQ-M9: Medication Search Feature**

User Story: As a user, I want to search for medications by name, category, or manufacturer.

Confirmation: The system returns a filtered list matching the search criteria instantly.

2.1.4 Diagnostic Services

Each requirement below is expressed as a user story with clear confirmation criteria. Unique IDs (e.g., REQ-D1, REQ-D2) ensure that all functionalities are fully traceable and verifiable.

- **REQ-D1: Schedule Diagnostic Test**

User Story: As a doctor or nurse, I want to schedule diagnostic tests for patients so that the required tests are performed in a timely manner.

Confirmation: The system allows the user to select the patient, choose the test type and time, and confirms scheduling with test ID and appointment details.

- **REQ-D2: View Test Results**

User Story: As a patient or doctor, I want to view test results online so that diagnosis and treatment can proceed efficiently.

Confirmation: Upon test completion, the system uploads the result and makes it accessible via the patient portal and doctor's dashboard.

- **REQ-D3: Upload Diagnostic Report**

User Story: As a lab technician, I want to upload the completed diagnostic report so that doctors can review and take necessary action.

Confirmation: The system accepts only authorized uploads linked to a valid test ID and displays confirmation of successful attachment.

- **REQ-D4: Test Status Tracking**

User Story: As a doctor, I want to track the status of ordered diagnostic tests (e.g., pending, in progress, completed) so that I stay informed.

Confirmation: The system displays real-time status updates for all tests associated with a patient, accessible via the test log.

- **REQ-D5: Diagnostic Billing Integration**

User Story: As an accountant or system admin, I want diagnostic tests to automatically generate bills for patients so that billing remains accurate and transparent.

Confirmation: On test scheduling, the system auto-generates a bill entry in the patient's record with test name, cost, and payment status.

- **REQ-D6: Manage Diagnostic Test Catalog**

User Story: As an admin, I want to add, edit, or delete diagnostic test types and pricing so that the catalog remains updated.

Confirmation: The system reflects all changes immediately and logs the modification with user ID and timestamp.

- **REQ-D7: Notify Patients of Scheduled Tests**

User Story: As a patient, I want to receive notifications when a diagnostic test is scheduled so that I don't miss the appointment.

Confirmation: The system sends an SMS/email and dashboard notification with the test details upon successful scheduling.

- **REQ-D8: Generate Test Summary Reports**

User Story: As a lab technician or admin, I want to generate summary reports of all diagnostic tests done over a period so that I can monitor department performance.

Confirmation: The system provides downloadable PDF/Excel reports filtered by date range, test type, and status.

- **REQ-D9: Emergency Test Flagging**

User Story: As a doctor, I want to mark certain tests as urgent so that the lab prioritizes those results for faster diagnosis.

Confirmation: When flagged, the system highlights these tests in the lab dashboard and sends an alert to the responsible technician.

2.1.5 Ambulance Booking

Each requirement below is expressed as a user story with clear confirmation criteria. Unique IDs (e.g., REQ-AB1, REQ-AB2) ensure that all functionalities are fully traceable and verifiable.

- **REQ-AB1: Emergency Ambulance Booking**

User Story: As a patient or caregiver, I want to book an emergency ambulance so that I can receive immediate medical transportation when needed.

Confirmation: Upon request submission, the system identifies the nearest available ambulance, dispatches it within 2 minutes, and sends SMS/notification confirmations to both the patient and hospital.

- **REQ-AB2: Scheduled Ambulance Booking**

User Story: As a hospital staff member, I want to schedule a patient transfer in advance so that proper arrangements are made without delay.

Confirmation: The system provides a booking interface with time/date selection, assigns an ambulance based on availability, and confirms the booking via notification.

- **REQ-AB3: Booking History Log**

User Story: As a system, I want to maintain a log of all ambulance booking activities so that they can be audited for performance and accountability.

Confirmation: Each booking, cancellation, and dispatch is recorded with timestamps, user details, and booking outcome.

- **REQ-AB4: Booking Confirmation**

User Story: As a user, I want to receive a clear confirmation after booking an ambulance so that I know my request has been successfully recorded.

Confirmation: After completing the booking process, the system displays an on-screen

confirmation message including booking ID, assigned vehicle details, and estimated time of arrival. The confirmation is also saved in the user's booking history.

- **REQ-AB5: Role-Based Access to Booking Interface**

User Story: As a hospital admin, I want to ensure that only authorized staff can book or cancel ambulances so that misuse is prevented.

Confirmation: The system enforces role-based permissions and logs every action taken within the ambulance module.

- **REQ-AB6: Ambulance Availability Dashboard**

User Story: As a dispatcher, I want to view all available and occupied ambulances in real time so that I can manage dispatches efficiently.

Confirmation: The dashboard displays vehicle status (available, en route, occupied), driver info in a color-coded format.

2.1.6 Role-Based Login

Each requirement is expressed as a user story with confirmation criteria. IDs (e.g., REQ-L1, REQ-L2...) ensure traceability.

- **REQ-L1: Role-Based Login Access**

User Story: As a user of the hospital management system, I want to log in using credentials specific to my role so that I can access only the features relevant to my responsibilities.

Confirmation: Upon successful login, the system verifies credentials, determines the user's role, and grants access to the appropriate dashboard and modules.

- **REQ-L2: Unauthorized Access Prevention**

User Story: As a system, I want to prevent users from accessing modules not assigned to their roles so that sensitive data and functionalities remain protected.

Confirmation: Role-based restrictions are enforced, with blocked access attempts generating an error message and logged entries.

- **REQ-L3: Role Management by Admin**

User Story: As an admin, I want to create, assign, modify, or delete roles so that I can manage who has access to which parts of the system.

Confirmation: Admin users can manage role definitions via the admin panel. All role changes are recorded in an audit log.

- **REQ-L4: Password Protection and Security**

User Story: As a user, I want my login credentials to be securely handled so that my account is protected from unauthorized access.

Confirmation: The system stores passwords using encryption/hashing (e.g., bcrypt), and login endpoints are protected via HTTPS.

- **REQ-L5: Multi-Factor Authentication (Optional)**
User Story: As a system, I want to optionally support multi-factor authentication so that sensitive roles (e.g., admin, doctor) have enhanced login security.
Confirmation: When enabled, users are required to provide a second factor (OTP, email code, etc.) after entering the password.
- **REQ-L6: Login Attempt Limits**
User Story: As a system, I want to limit login attempts to prevent brute-force attacks and secure the platform.
Confirmation: After 5 failed login attempts, the account is temporarily locked or CAPTCHA is required; event is logged.
- **REQ-L7: Session Timeout Management**
User Story: As a system, I want to log out idle users after a certain time so that sessions are not misused.
Confirmation: Auto-log out inactive users after some idle duration (e.g., 15 minutes), with warning prior to logout.
- **REQ-L8: Login Activity Logs**
User Story: As an admin, I may view login activity history to monitor system usage and detect anomalies.
Confirmation: Login times, IP addresses, and success/failure indicators are logged by the system and provided to admins in reportable format.
- **REQ-L9: Forgot Password and Recovery**
User Story: As a user, I want to reset my password when I forget it so that I can safely return to the system.
Confirmation: The system facilitates a secure password recovery process via registered phone number or email with identity verification procedures.
- **REQ-L10: Cross-Role Login Restriction**
User Story: As a system, I want to disallow the users to be able to log in under multiple roles at the same time to maintain data integrity as well as responsibility.
Confirmation: The system disallows or warns against dual logins between various roles by the same user account.

2.1.7 Staff Management

Each requirement below is expressed as a user story with confirmation criteria. Unique IDs (e.g., REQ-SM1, REQ-SM2) ensure traceability.

- **REQ-SM1:** As an admin, I want to add and register new staff profiles so that I can manage all personnel centrally.

Confirmation: Upon submitting staff information, the system stores the profile with a unique ID and login credentials.

- **REQ-SM2:** As an admin, I want to assign staff roles and departments so that everyone has clearly defined responsibilities.

Confirmation: Roles and departments are assigned via dropdowns, saved with timestamps, and visible in the staff list.

- **REQ-SM3:** As a staff member, I want to view my assigned shifts and responsibilities so that I stay informed about my duties.

Confirmation: The staff dashboard displays current and upcoming schedules based on login credentials.

- **REQ-SM4:** As a supervisor, I want to track staff performance and attendance so that I can evaluate and plan better.

Confirmation: Performance logs are updated daily with attendance and task completion data, accessible via secure supervisor login.

- **REQ-SM5:** As a system, I want to notify staff of shift changes or new assignments so that everyone is aligned with real-time updates.

Confirmation: Whenever a change occurs, the system pushes a notification via SMS/email and updates the staff dashboard.

2.1.8 Room Allocation

Each requirement is expressed as a user story with confirmation criteria. Unique IDs (e.g., REQ-RA1, REQ-RA2...) ensure traceability.

- **REQ-RA1:** As a receptionist, I want to assign patients to available rooms so that the admission process is streamlined.

Confirmation: The system shows only vacant rooms, validates patient details, and locks the room upon successful assignment.

- **REQ-RA2:** As a doctor, I want to request specific room types (ICU/general) based on patient condition so that medical needs are met.

Confirmation: Doctors can select room types and see availability by category; requests are prioritized by urgency.

- **REQ-RA3:** As housekeeping staff, I want to update room cleanliness and maintenance status so that the system reflects real-time room readiness.

Confirmation: Staff can mark rooms as 'clean', 'occupied', or 'under maintenance'; updates are logged with timestamp and user ID.

- **REQ-RA4:** As a system, I want to auto-notify staff when rooms become available so that delays in patient accommodation are reduced.

Confirmation: When a room's status changes to 'available', notifications are sent to reception and admission dashboards.

- **REQ-RA5:** As a hospital admin, I want to view occupancy statistics and room usage reports so that I can plan resource allocation.

Confirmation: Reports show room turnover rates, peak times, and department-wise usage, updated in real time.

2.1.9 Prescription Management

- **REQ-P1: Create Digital Prescription**

User Story: As a doctor, I want to create digital prescriptions for patients so that treatment can be efficiently documented.

Confirmation: The system saves the prescription with drug names, dosage, frequency, and notes in the patient's profile.

- **REQ-P2: View Prescription History**

User Story: As a patient or pharmacist, I want to view past prescriptions so that medications can be verified and dispensed.

Confirmation: The system shows all prescriptions issued per patient with timestamps.

- **REQ-P3: Modify or Cancel Prescriptions**

User Story: As a doctor, I want to update or cancel a prescription if the patient's condition changes.

Confirmation: The system allows only the issuing doctor to modify or cancel, and logs changes for auditing.

- **REQ-P4: Prescription Forwarding**

User Story: As a doctor, I want prescriptions to be automatically forwarded to pharmacy and lab modules for fulfillment.

Confirmation: Once submitted, the system sends the prescription details to the respective departments.

- **REQ-P5: Print and Download Option**

User Story: As a user, I want to print or download a copy of my prescription for reference.

Confirmation: The system generates a downloadable and printable PDF of the prescription.

2.1.10 Billing and Payment System

- **REQ-B1: Automatic Bill Generation**

User Story: As an admin, I want bills to be auto-generated when services are rendered so

that no charges are missed.

Confirmation: The system compiles consultation, test, and medication fees into one bill upon checkout.

- **REQ-B2: Multi-Mode Payment Support**

User Story: As a patient, I want to pay using cash, card, or mobile payments for flexibility.

Confirmation: The system supports different payment gateways and logs transactions.

- **REQ-B3: Apply Discounts and Insurance**

User Story: As an admin, I want to apply discounts or insurance so that patients are billed correctly.

Confirmation: The system updates the payable amount and records applied policies or discount details.

- **REQ-B4: View and Download Invoice**

User Story: As a patient, I want to view and download my invoices for future reference.

Confirmation: The system allows patients to access and download a PDF of any past invoice.

- **REQ-B5: Billing Dashboard and Reports**

User Story: As an accountant, I want to generate reports of all payments and dues to manage finances.

Confirmation: The system provides filters by date, department, or patient for detailed billing reports.

- **REQ-B6: Payment Reminder Alerts**

User Story: As a system, I want to send payment reminders to patients with pending dues.

Confirmation: The system sends email/SMS notifications with due amount and payment links.

With these functional requirements, the Hospital Management System is architected to deliver comprehensive, end-to-end coverage of all hospital processes, ensuring efficiency, accuracy, and a seamless experience for all users. The requirements are structured to provide a clear and actionable development roadmap, and to ensure that every critical workflow within the hospital is addressed.

2.2 Non-functional Requirements

Non-functional requirements define the quality attributes, system constraints, and operational characteristics that ensure the Hospital Management System is robust, reliable, and fit for its mission-critical role in healthcare. These requirements are as vital as the functional ones, as they govern the system's usability, security, and maintainability.

2.2.1 Performance Requirements

The Hospital Management System (HMS) must ensure consistently high performance across all modules and user scenarios, as it serves critical healthcare functions. The system should be able to simultaneously handle a minimum of 200 active users without noticeable degradation in speed or responsiveness. These users may include doctors, administrative staff, pharmacists, diagnostic technicians, and patients accessing the platform.

Each functional operation—such as patient registration, appointment booking, room allocation, ambulance request, and billing—must be completed in under 2 seconds in normal operating conditions. During peak hours, such as early mornings or emergencies, response time should not exceed 5 seconds. This ensures timely delivery of services, especially in life-critical operations like prescription generation or ambulance dispatch.

Diagnostic services and patient history retrieval operations, which involve larger data access, must be optimized for faster query processing using appropriate indexing and database normalization techniques. These should not exceed 3 seconds even during concurrent access.

The system must support real-time updates for sensitive tasks such as ambulance tracking, room status changes, or medication stock adjustments. These updates should reflect on all connected terminals within 1 second of the action.

Server uptime must meet 99.9% availability, allowing for minimal unplanned downtime. Automated nightly backups should not affect system responsiveness and must be completed within a 30-minute window. Additionally, the system must handle load spikes (e.g., during pandemic or disaster scenarios) with graceful degradation by limiting certain non-critical modules temporarily and queuing less-urgent operations.

Performance testing should be conducted before deployment and periodically afterward to ensure compliance with these standards.

2.2.2 Safety Requirements

Safety is of paramount importance in a hospital setting, where incorrect or delayed information can lead to critical consequences. The system must ensure the integrity and reliability of patient-related data, especially during prescription handling, diagnostics, medication management, and emergency services.

All patient data must be automatically validated upon entry. For example, prescription inputs must be verified against dosage guidelines and known patient allergies. Medication store alerts must be triggered when stock for essential drugs reaches a critical threshold to avoid unavailability during emergencies.

Room allocation systems must prevent assignment conflicts, such as double-booking or unauthorized placement in specialized care units like ICU. Similarly, ambulance booking must confirm driver and vehicle availability and prevent overlapping or duplicate assignments.

Safety protocols must also include redundancy measures: in the event of a system failure, a manual override system should activate within 5 minutes. Data backups (encrypted and tested for restorability) must be stored in both on-site and cloud-based servers, with a recovery time objective (RTO) of under 30 minutes and a recovery point objective (RPO) of under 5 minutes.

Physical access to servers and hospital terminals should be restricted. Critical features such as diagnostic result uploads, medication dispatch, and prescription changes must only be accessible by authorized personnel and logged for accountability.

All safety mechanisms must comply with national and international medical safety standards, including local health authority regulations and international protocols such as HL7 or ISO/IEC 62304 (for software lifecycle processes in medical devices).

2.2.3 Security Requirements

Due to the sensitivity of medical and financial data handled by the HMS, the system must implement industry-standard security protocols at every layer of interaction. Data confidentiality, integrity, and availability must be ensured for all users.

Role-based access control (RBAC) must be enforced to restrict data and functionality based on user roles. Doctors may generate and view prescriptions but cannot process payments. Pharmacists can update stock records but cannot view patient diagnostics. Patients may access only their own data, such as appointment schedules, bills, and history.

All users must authenticate with a unique username and strong password combination. High-privilege users (e.g., hospital admin, financial officers) should be required to use two-factor authentication (2FA) using email or phone-based OTPs. Sessions must automatically time out after 10 minutes of inactivity to reduce the risk of unauthorized access from unattended terminals.

All data transactions between the client and server must occur over secure HTTPS connections with TLS encryption. Sensitive data such as diagnostic records, billing details, and staff personal information must be encrypted at rest using AES-256 or equivalent algorithms.

The system must log all user activity, especially those involving changes to sensitive records (e.g., test results, patient status, prescription history). These logs must be immutable and retained for a minimum of 2 years, with proper tools for forensic audit.

Security compliance must align with regional healthcare data regulations such as the Bangladesh Digital Security Act and global standards like HIPAA (if applicable for international use). Periodic penetration testing and vulnerability assessments should be conducted to proactively address threats.

2.2.4 Software Quality Attributes

The HMS must meet high standards in various software quality dimensions to support sustainable, long-term use and adaptability to changing requirements.

- **Reliability:** The system must ensure 99.9% uptime, with seamless handling of core features such as prescription management, room allocation, billing, and ambulance dispatch. The system must continue to operate even if non-critical components fail.
- **Maintainability:** The modular design of the system will allow updates to be made to individual modules (e.g., billing or diagnostics) without affecting the rest of the system. Proper documentation and versioning must be maintained for all modules.
- **Usability:** The user interface must be designed to accommodate both tech-savvy professionals and non-technical staff. A uniform layout, clear icons, and guided workflows must reduce the learning curve to under 2 hours for basic use.
- **Interoperability:** The system must support integration with third-party diagnostic equipment, government health portals, and insurance systems using standard protocols such as RESTful APIs or HL7 where applicable.
- **Scalability:** The system must be scalable both vertically and horizontally. As patient volume increases or new departments are added, the system should accommodate additional users, features, and data without re-architecting.
- **Portability:** The system must be browser-based and mobile responsive, enabling access across desktop and mobile devices. It should function on all major browsers (Chrome, Firefox, Edge) without performance degradation.
- **Robustness:** Error-handling mechanisms must ensure that exceptions (e.g., invalid data entries or server communication failures) do not crash the system. All such events should be logged and recoverable.
- **Testability:** Unit, integration, and system-level testing should be supported. Automated test scripts should be written for critical operations like patient registration, prescription validation, and billing.

2.2.5 Business Rules

The HMS must enforce a range of business logic and operational protocols aligned with real-world hospital workflows. These rules define how and when specific functionalities may be accessed or triggered.

- Only registered doctors can create, modify, or cancel prescriptions. Once submitted, prescriptions are digitally signed and cannot be altered without authorization.
- Billing processes are managed exclusively by the finance department. Automated billing is triggered after appointment completion or diagnostic test submission.

- Appointment booking must align with the doctor's availability and departmental working hours. Patients cannot double-book overlapping times.
- Room allocation must follow priority-based logic. Emergency patients and ICU cases override normal admission sequences based on physician recommendations.
- Ambulance booking must check the availability of both vehicles and driver. A booking is not confirmed until it is explicitly approved by the ambulance desk.
- Staff management functions (scheduling, leave tracking, role assignment) are accessible only by HR or administrative personnel.
- Medication stock updates require dual verification: one from the store manager and another from a pharmacist. Alerts must be raised automatically for expiring or low-stock items.
- Only lab technicians can upload diagnostic results. These results are locked after submission and viewable by patients and physicians, but not editable.
- Each user has a unique role and dashboard experience. No cross-role functionality should be available unless explicitly granted through administrative controls.

These business rules ensure operational efficiency, legal compliance, and service integrity across all departments and user groups within the hospital.

In summary, the requirements detailed in this chapter serve as the blueprint for the HMS, ensuring that both the business objectives and the operational realities of healthcare are addressed. The next chapter will focus on the system design, outlining how these requirements are translated into effective, scalable, and maintainable architecture that supports the evolving needs of a modern hospital.

CHAPTER 3: SYSTEM DESIGN

System design is the process of defining the architecture, components, modules, interfaces, and data flow of a software system to meet specified requirements. It focuses on creating a blueprint that outlines how the system will function, how different parts will interact, and how the system will achieve its objectives in a scalable, efficient, and maintainable manner.

3.1 UML Use Case Diagram

The Hospital Management System is designed to streamline healthcare operations by integrating patient management, clinical services, logistics, and administrative tasks into a unified platform. The use case diagrams serve as a visual blueprint that maps out all major functionalities and interactions between the system and its key external actors. Together, the diagrams emphasize both high-level system requirements and the nuanced relationships between various processes.

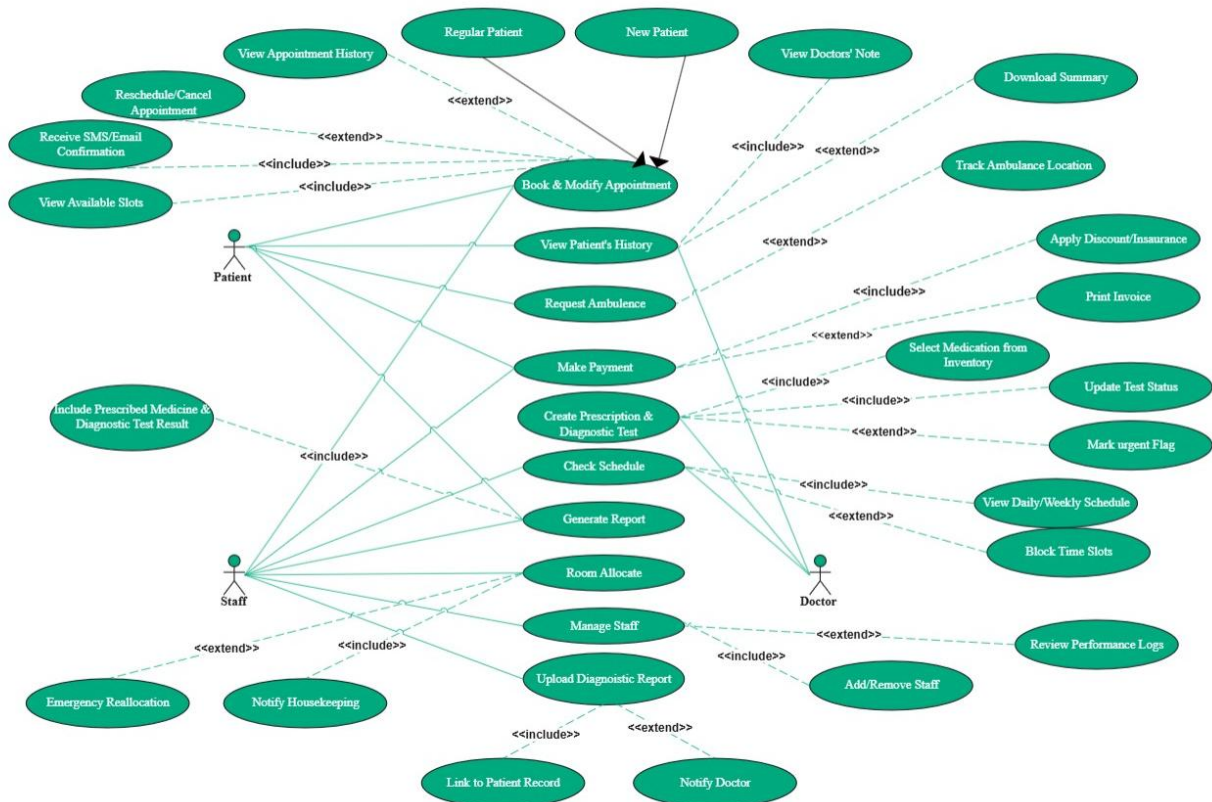


Fig 1: Use Case Diagram

The provided use case diagram visually represents the major functionalities and user interactions within the Hospital Management System. It features three primary actors—Patient, Doctor, and Staff—each with distinct roles and system interactions, as detailed below.

Actors and Their Roles

1. Patient:

Patients primarily use the system to manage their own care. They can:

- Book, modify, and cancel appointments
- View available appointment slots and appointment history
- Receive SMS/email confirmations
- Request ambulance services
- Make payments for services
- Access scheduled treatment plans and generate related reports

2. Doctor:

Doctors are central to the clinical workflow. They utilize the system to:

- Access comprehensive patient histories and notes
- Prescribe medications and order diagnostic tests
- Block or view time slots for consultations
- Download visit and treatment summaries
- Apply discounts or insurance adjustments
- Communicate and coordinate with medical teams via notifications

3. Staff:

Staff members—including administrators and technicians—support hospital operations by:

- Allocating rooms and managing admissions
- Uploading and linking diagnostic reports to patient profiles
- Managing staff schedules and overseeing room availability
- Handling emergency reallocations and notifying housekeeping
- Generating reports and overseeing hospital resource management

Functional Components and Use Cases

The diagram organizes the system's capabilities into well-defined use cases, each representing a functional requirement:

- **Appointment Management:** Patients can schedule, view, modify, or cancel appointments, while staff and doctors can check schedules and manage bookings. The system includes real-time slot viewing, confirmation notifications, and appointment history access.

- **Emergency Services:** The system supports ambulance requests and real-time ambulance location tracking, extending care beyond routine scenarios to urgent medical needs.
- **Medical Documentation and Reporting:** Doctors and staff can generate and download reports, update medical records, annotate diagnostic results, and link data to specific patient records.
- **Financial Transactions:** Integrated billing allows patients to make payments, print invoices, and utilize discounts/insurance. Doctors and staff can apply financial adjustments and monitor transactions.
- **Resource and Staff Management:** Staff manage room allocation, staff schedules, and performance logs, with the ability to handle emergency reallocations for operational efficiency.

Relationships and Process Dependencies

The diagram highlights two key types of relationships:

- **Include Relationships (<<include>>):** These indicate that certain use cases always incorporate the functionality of another. For example, booking an appointment always includes viewing available slots and receiving confirmation.
- **Extend Relationships (<<extend>>):** These show optional or conditional behaviors. For instance, booking an appointment can be extended to include automated reminders or real-time doctor notifications.

Such relationships ensure essential processes are consistently executed, while optional enhancements are flexibly integrated according to workflow needs.

Integrated System View

This use case diagram provides a cohesive, high-level overview of the Hospital Management System. It illustrates:

- The responsibilities and privileges of each actor
- The critical pathways for care management, resource allocation, and administration
- The interdependence between core and auxiliary processes

By mapping out these interactions, the diagram serves as an essential reference for developers, system architects, and hospital administrators. It guides system design and implementation, ensuring all operational requirements are addressed and the system remains robust, scalable, and user centered.

This comprehensive use case diagram lays the groundwork for structured system development, supporting the hospital’s goal of delivering efficient, coordinated, and high-quality healthcare services to all stakeholders.

3.2 Extended Use Cases

This section details the primary extended use cases that collectively define the core operational workflows of the Hospital Management System. Each use case describes a real-world scenario involving one or more actors, their step-by-step interactions with the system, exception handling, and the conditions before and after the process. The scenarios span a wide range of hospital operations, from appointment booking and patient record management to staff administration, resource allocation, pharmacy operations, diagnostics, user authentication, emergency ambulance requests, prescription processing, and billing.

The use cases are designed to capture not only the standard flows but also the exception paths, ensuring the system is robust and user centric. For instance, the system handles exceptions such as unavailable appointment slots, unauthorized access, system errors, and payment failures. Preconditions and postconditions are specified for each use case to clarify the necessary setup and the expected outcomes, improving traceability and testability during system development.

By mapping out these extended use cases, the Hospital Management System ensures comprehensive coverage of the hospital’s day-to-day and critical operations, supporting all stakeholders—including patients, doctors, staff, administrators, and support personnel—through an integrated, efficient, and secure digital platform.

1. Appointment Booking

Field	Details
Use Case	Appointment Booking
Actor(s)	Patient
Scenario	<ol style="list-style-type: none">1. Patient logs into the system.2. Navigates to the appointment booking module.3. Selects a doctor and an available time slot.4. Enters required details (reason for visit, contact info).5. Submits the booking request.6. System validates and confirms the appointment.

Exceptions	<ul style="list-style-type: none"> - Selected time slot is unavailable - Missing mandatory fields - System error or downtime
Preconditions	<ul style="list-style-type: none"> - Patient is registered and logged in - Doctor has available time slots
Post Conditions	<ul style="list-style-type: none"> - Appointment is booked and confirmed - A confirmation email/notification is sent - The selected time slot is marked as booked

2. Patient History Management

Field	Details
Use Case	Patient History Management
Actor(s)	Patient, Doctor, Medical Staff
Scenario	<ol style="list-style-type: none"> 1. Authorized user logs into the system. 2. Navigates to the Patient History module. 3. Searches for and opens a patient record. 4. Views complete medical history. 5. Optionally adds or updates history entries.
Exceptions	<ul style="list-style-type: none"> - Unauthorized access attempt - No existing records for a new patient - Database issues
Preconditions	<ul style="list-style-type: none"> - Patient record exists - User is authorized to access/edit the data
Post Conditions	<ul style="list-style-type: none"> - History displayed or updated - No records found” message for new patients

3. Staff Management

Field	Details
-------	---------

Use Case	Staff Management
Actor(s)	Admin, HR Staff
Scenario	<ol style="list-style-type: none"> 1. Admin logs in 2. Navigates to Staff Management 3. Adds/Modifies/Removes staff records 4. Assigns roles and departments 5. Saves changes
Exceptions	<ul style="list-style-type: none"> - Missing details - Unauthorized modification - System error during save
Preconditions	- Admin is logged in
Post Conditions	<ul style="list-style-type: none"> - Staff records updated - Changes logged in audit trail

4. Room Allocation

Field	Details
Use Case	Room Allocation
Actor(s)	Receptionist, Hospital Admin
Scenario	<ol style="list-style-type: none"> 1. Receptionist logs in 2. Searches for available rooms 3. Selects and assigns a room 4. Saves the allocation
Exceptions	<ul style="list-style-type: none"> - Room already occupied - System error during allocation
Preconditions	<ul style="list-style-type: none"> - Patient admitted - At least one room available
Post Conditions	- Room marked as “occupied” and linked to patient

5. Medication Store Management

Field	Details
Use Case	Medication Store Management
Actor(s)	Pharmacist, Admin
Scenario	1. Pharmacist logs in 2. Reviews stock 3. Adds/updates quantity, expiration 4. Saves changes
Exceptions	- Stock over-dispensed - Invalid data entry - Inventory update failure
Preconditions	- User is authorized
Post Conditions	- Inventory updated - Low stock alerts triggered

6. Diagnostic Services

Field	Details
Use Case	Diagnostic Services
Actor(s)	Doctor, Lab Technician
Scenario	1. Doctor orders a test 2. Order goes to lab 3. Technician performs and uploads results 4. Doctor reviews results
Exceptions	- Unauthorized access - System error during result upload
Preconditions	- Diagnostic order exists - User is authorized

Post Conditions	- Results stored and accessible in patient record
-----------------	---

7. Role-Based Login

Field	Details
Use Case	Role-Based Login
Actor(s)	Admin, Doctor, Receptionist, Patient
Scenario	<ol style="list-style-type: none"> 1. User opens login page 2. Enters credentials 3. System validates and assigns role 4. User redirected to dashboard
Exceptions	<ul style="list-style-type: none"> - Incorrect credentials - Unauthorized access attempts
Preconditions	- Valid user account exists
Post Conditions	<ul style="list-style-type: none"> - User redirected to correct dashboard - Failed attempts show error

8. Ambulance Booking

Field	Details
Use Case	Ambulance Booking
Actor(s)	Patient, Hospital Staff
Scenario	<ol style="list-style-type: none"> 1. Accesses booking module 2. Inputs location, contact, urgency 3. Submits request 4. System assigns ambulance
Exceptions	<ul style="list-style-type: none"> - No ambulance available - System booking error

Preconditions	<ul style="list-style-type: none"> - Emergency service active - Ambulance availability confirmed
Post Conditions	<ul style="list-style-type: none"> - Ambulance booked and dispatched - Notification sent

9. Prescription Management

Field	Details
Use Case	Prescription Management
Actor(s)	Doctor, Pharmacist, Patient
Scenario	<ol style="list-style-type: none"> 1. Doctor logs in 2. Enters medication, dosage, instructions 3. Prescription sent to pharmacy and stored
Exceptions	<ul style="list-style-type: none"> - Invalid prescription - Medication unavailable
Preconditions	<ul style="list-style-type: none"> - Valid consultation exists - Doctor is authorized
Post Conditions	<ul style="list-style-type: none"> - Prescription recorded - Pharmacy notified

10. Billing & Payment System

Field	Details
Use Case	Billing & Payment System
Actor(s)	Patient, Admin, Billing Staff
Scenario	<ol style="list-style-type: none"> 1. Bill auto-generated post service 2. Patient views bill

	3. Chooses payment method 4. Payment processed and receipt issued
Exceptions	- Payment gateway failure - Insufficient funds - Billing error
Preconditions	- Billing record exists - Payment method available
Post Conditions	- Payment marked "paid" - Receipt issued

The extended use cases presented here form the backbone of the Hospital Management System, outlining the essential processes and interactions that drive hospital operations. By addressing both typical scenarios and potential exceptions, these use cases ensure the system is prepared for real-world complexities. Clear definition of actors, workflows, and outcomes provides a strong foundation for system design, implementation, and continuous improvement, ultimately enabling the hospital to deliver high-quality, patient-centered care through streamlined, reliable, and secure processes.

3.3 UML Class Diagram

The class diagram provides a comprehensive structural overview of the Hospital Management System, illustrating the main entities (classes), their attributes, methods, and the relationships between them. This diagram captures both the inheritance hierarchy for users and the core domain interactions that support hospital operations.

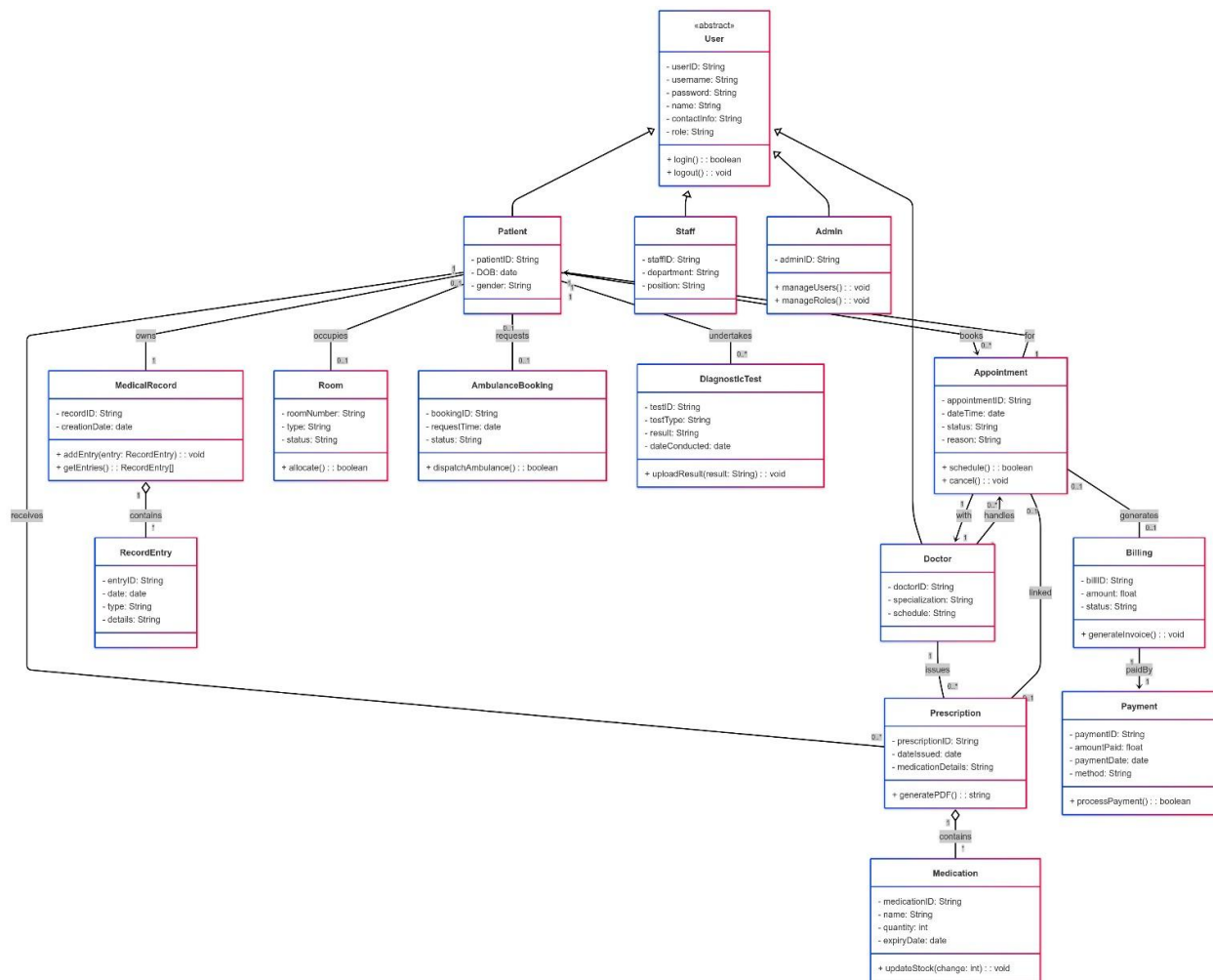


Fig 2: Class Diagram of Hospital Management System

1. User Hierarchy

- **User (abstract)**

- *Attributes:* userID, username, password, name, contactInfo, role
- *Methods:* login(), logout()
- *Description:* Serves as an abstract base class for all system users, encapsulating authentication and identification.

- **Patient**

- *Inherits from:* User
- *Attributes:* patientID, DOB, gender

- *Associations:*
 - **MedicalRecord** (1:1): Each patient has one medical record.
 - **Appointment** (1..*): A patient can book multiple appointments.
 - **AmbulanceBooking** (0..*): A patient may request ambulance services.
 - **Room** (0..1): A patient may occupy one room.
 - **Billing & Payment** (through Appointment): Patient pays for services rendered.
- **Staff**
 - *Inherits from:* User
 - *Attributes:* staffID, department, position
 - *Description:* Represents any hospital staff member. Staff can undertake diagnostic tests and manage rooms.
- **Admin**
 - *Inherits from:* User
 - *Attributes:* adminID
 - *Methods:* manageUsers(), manageRoles()
 - *Description:* Administrator with permissions to manage other users and system roles.

2. Core Operational Classes

- **MedicalRecord**
 - *Attributes:* recordID, creationDate
 - *Methods:* addEntry(), getEntries()
 - *Associations:*
 - **Patient** (1:1): Each patient has one medical record.
 - **RecordEntry** (1..*): A medical record contains multiple entries (e.g., diagnoses, treatments).
- **RecordEntry**
 - *Attributes:* entryID, date, type, details
 - *Description:* Represents individual entries in a patient's medical record.
- **Room**

- *Attributes:* roomNumber, type, status
- *Methods:* allocate()
- *Associations:*
 - **Patient** (0..1): A room may be occupied by a patient.
- *Description:* Manages room allocation and tracks occupancy status.
- **AmbulanceBooking**
 - *Attributes:* bookingID, requestTime, status
 - *Methods:* dispatchAmbulance()
 - *Associations:*
 - **Patient** (0..*): Patients can make ambulance bookings.
- **Appointment**
 - *Attributes:* appointmentID, dateTime, status
 - *Methods:* schedule(), cancel()
 - *Associations:*
 - **Patient** (1..*): Patients book appointments.
 - **Doctor** (1): Each appointment involves a doctor.
 - **Billing** (1): Each appointment generates a billing record.
 - *Description:* Orchestrates the scheduling and management of consultations.
- **Doctor**
 - *Inherits from:* User
 - *Attributes:* doctorID, specialization, schedule
 - *Associations:*
 - **Appointment** (0..*): Doctors have multiple appointments.
 - **Prescription** (0..*): Doctors issue prescriptions.
 - **DiagnosticTest** (0..*): Doctors order diagnostic tests.
- **DiagnosticTest**
 - *Attributes:* testID, testType, result, dateConducted
 - *Methods:* uploadResult()
 - *Description:* Represents lab or imaging tests conducted for a patient.

3. Treatment & Medication

- **Prescription**
 - *Attributes:* prescriptionID, dateIssued, medicationDetails
 - *Methods:* generatePDF()
 - *Associations:*
 - **Doctor** (1): Issued by a doctor.
 - **Medication** (1..*): Contains one or more medications.
- **Medication**
 - *Attributes:* medicationID, name, quantity, expiryDate
 - *Methods:* updateStock()
 - *Description:* Represents medicine inventory, including stock management.

4. Financial Management

- **Billing**
 - *Attributes:* billID, amount, status
 - *Methods:* generateInvoice()
 - *Associations:*
 - **Appointment** (1): Each appointment leads to a billing entry.
 - **Payment** (1): Each bill is settled by a payment.
- **Payment**
 - *Attributes:* paymentID, amount, paymentDate, method
 - *Methods:* processPayment()
 - *Description:* Tracks the payment of bills and supports multiple payment methods.

5. Relationships Overview

- **Inheritance:** All users (Patient, Staff, Doctor, Admin) inherit from the abstract User class, ensuring consistent authentication and role management.
- **Associations:**
 - Patients are linked to appointments, medical records, rooms, ambulance bookings, and billing.

- Doctors are connected to appointments, prescriptions, and diagnostic tests.
- Staff manage rooms, perform diagnostic tests, and oversee operational activities.
- Admins oversee user and role management.
- **Composition/Aggregation:**
 - MedicalRecord contains multiple RecordEntry objects.
 - Prescription contains one or more Medication objects.

This class diagram encapsulates the foundational data structure of the Hospital Management System. It reflects real-world hospital entities and workflows, ensuring that all interactions—be it care delivery, resource allocation, or financial transactions—are logically modeled and interconnected. By adhering to object-oriented principles, the system supports extensibility, maintainability, and provides a clear blueprint for implementation and future enhancements.

3.4 Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) is a tool used in system design to represent the flow of information within a system. It helps break down a system into smaller components, allowing both technical and non-technical stakeholders to understand how data moves, is processed, and is stored. In this project, we use DFDs to visualize the operations of a Hospital Management System.

Levels of DFD

A Data Flow Diagram (DFD) is typically developed in multiple levels to represent a system in increasing detail. The three common levels of DFD are Level 0, Level 1, and Level 2.

The Level 0 DFD, also known as the context diagram, provides the most abstract view of the system. It represents the entire system as a single process and shows its interaction with external entities such as users, administrators, or other systems. This level focuses only on the input and output data flows between the system and its environment, without revealing internal processes or data stores.

Level 1 DFD expands this single process into a set of sub-processes, each representing a major function within the system. It introduces internal data flows and data stores to show how data moves and is transformed within the system. This level provides a clearer understanding of the system's functional components and how they interact with each other and with external entities.

Level 2 DFD takes this a step further by zooming into a specific process from Level 1 to provide more detailed insights. It breaks down a particular process into smaller, logical sub-processes, often with their own data flows and supporting data stores. This level is especially useful for

understanding the step-by-step operation of complex features within the system. Together, these three levels of DFD offer a structured and layered approach to system analysis and design, moving from a general overview to detailed functionality.

Level 0 DFD

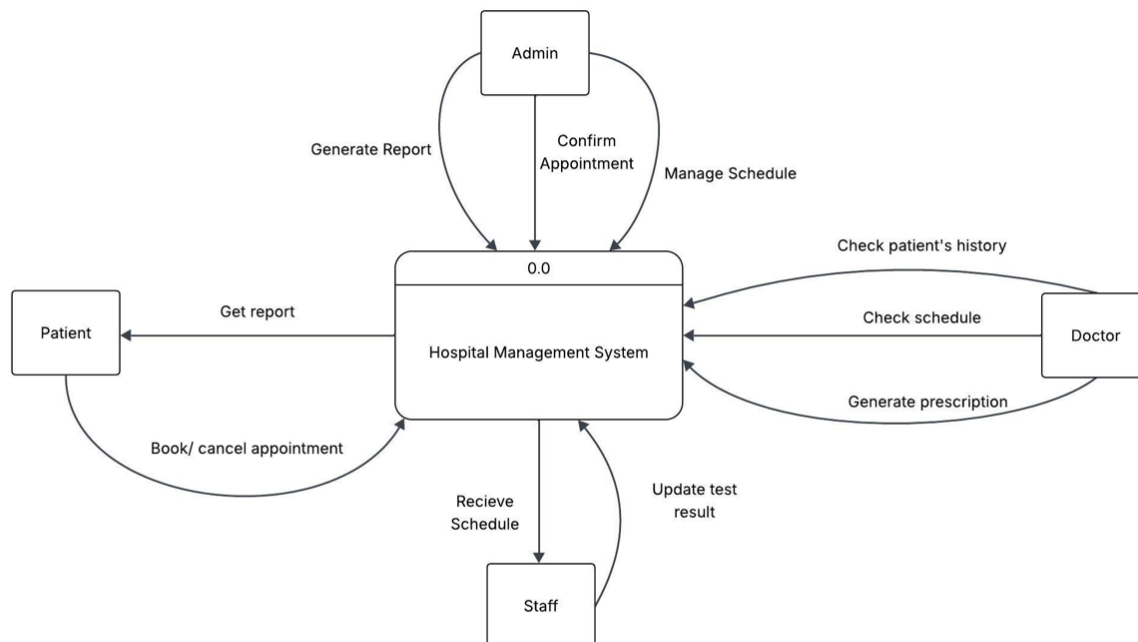


Fig: Level-0 DFD of "Hospital Management System"

The Level 0 DFD provides the most abstract, top-level view of the Hospital Management System. Here, the entire system is depicted as a single process (Process 0.0), showing its interactions with key external entities: Patient, Doctor, Admin, and Staff. Data flows such as booking appointments, generating reports, confirming appointments, updating test results, and managing schedules are illustrated as simple input and output arrows between the system and its users. At this level, the diagram focuses only on how the system as a whole communicates with its environment, without exposing any internal workings.

Level 1 DFD

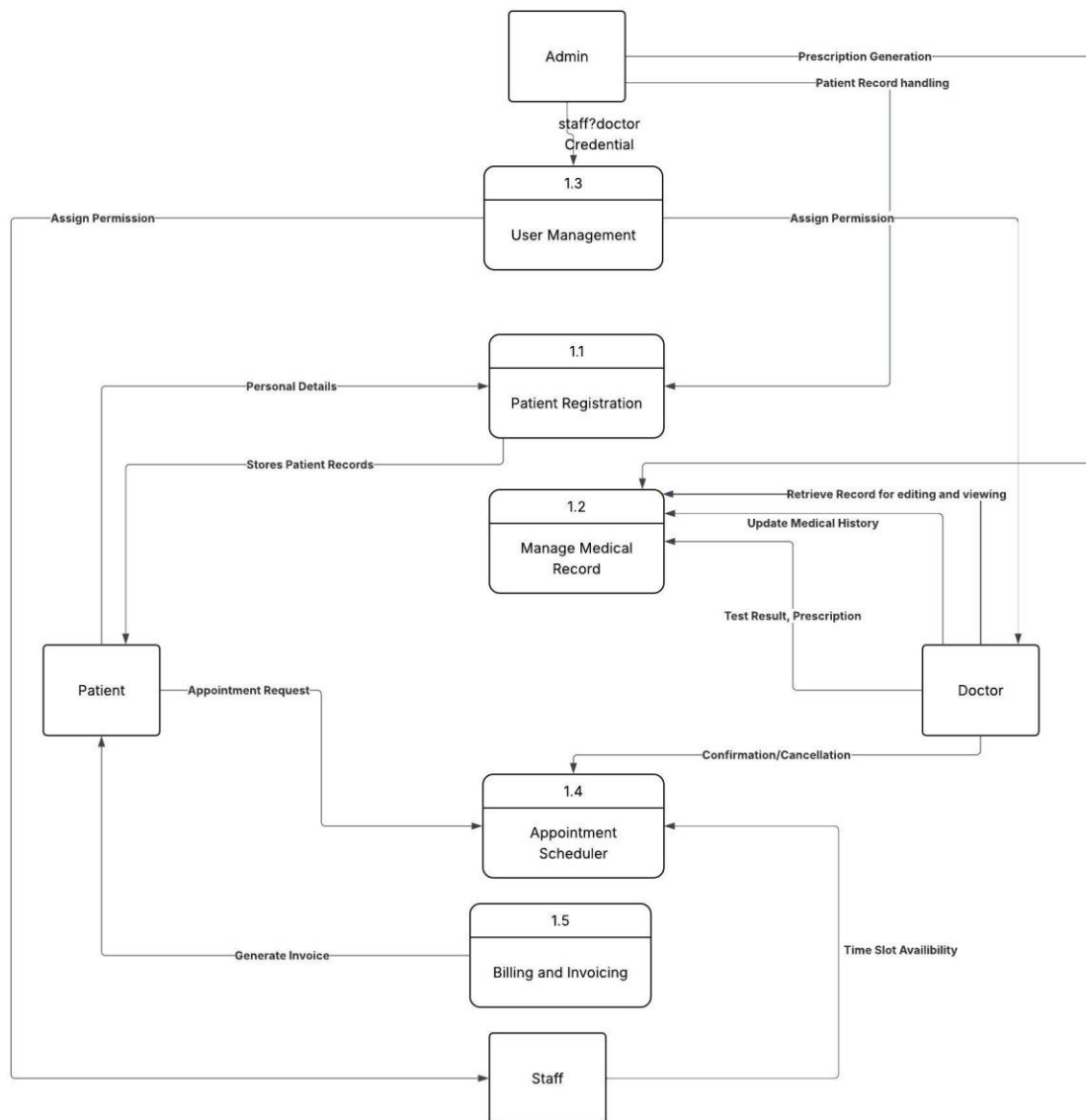


Fig: Level-1 DFD "Hospital Management System"

The Level 1 DFD breaks down the single process from Level 0 into major functional sub-processes. These include Patient Registration, Manage Medical Record, User Management, Appointment Scheduler, and Billing and Invoicing. Each sub-process shows its data exchanges with users (Patient, Doctor, Admin, Staff) and with each other. For example, patients may register and submit appointment requests, doctors handle prescriptions and medical history, and admins manage user permissions. This level introduces data stores and illustrates how data flows between

modules, providing a more detailed understanding of the system's main functions and their interactions.

Level 2 DFD

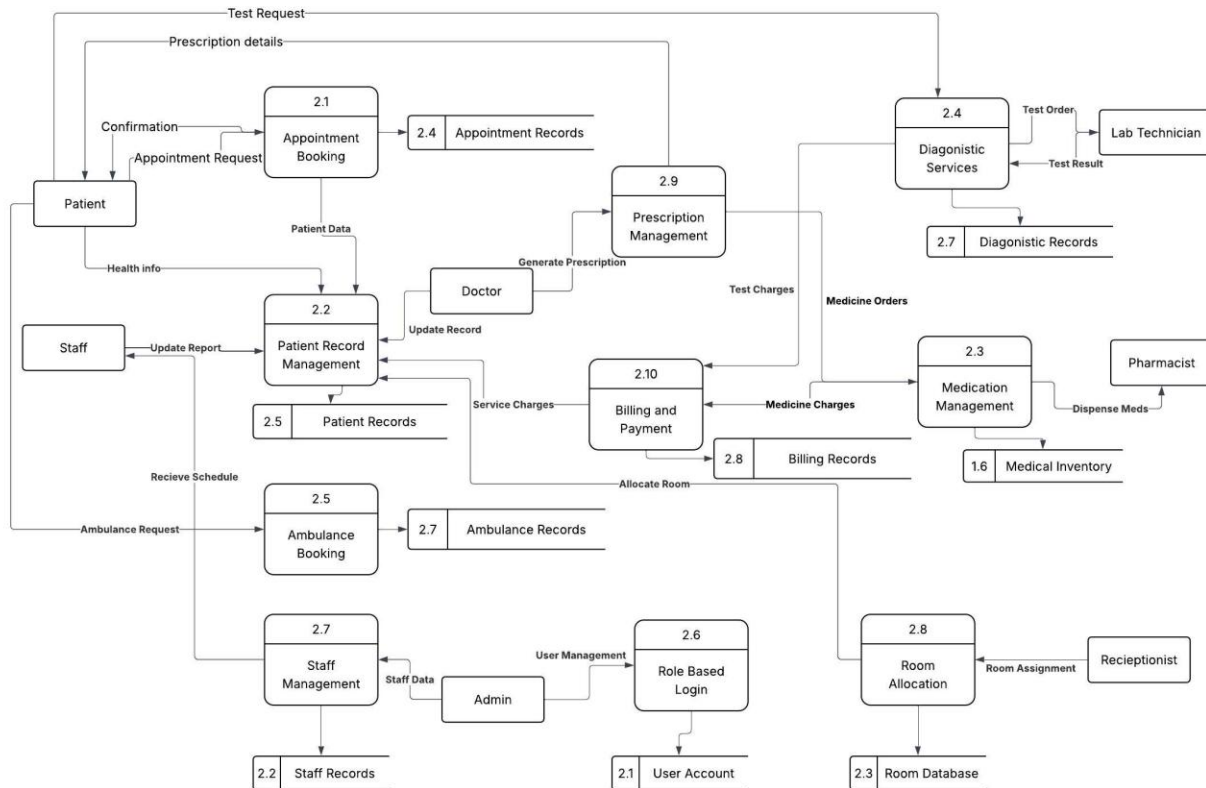


Fig: Level-2 DFD "Hospital Management System"

The Level 2 DFD zooms into the major processes from Level 1, breaking them down into even finer sub-processes. It details specific modules such as Appointment Booking, Patient Record Management, Diagnostic Services, Medication Management, Ambulance Booking, Room Allocation, Role-Based Login, Staff Management, and Billing and Payment. The diagram maps out how data (like health info, appointment requests, test results, prescriptions, billing records, staff records, and user accounts) flows between these modules, their supporting data stores, and the actors (Patient, Doctor, Staff, Pharmacist, Lab Technician, Receptionist, Admin). This level reveals the step-by-step operations and dependencies between all system components, showing precisely how information is processed, stored, and retrieved across the hospital's digital infrastructure.

3.5 Database Schema

A well-structured database schema is fundamental to the reliability, efficiency, and scalability of the Hospital Management System. The schema must accurately reflect the real-world entities, relationships, and processes that occur within a hospital environment, facilitating robust data storage, retrieval, consistency, and security. Below is a detailed relational database schema designed to support all critical modules of the Hospital Management System, including user management, patient records, appointments, billing, prescriptions, diagnostics, ambulance bookings, medication inventory, room allocation, and staff management.

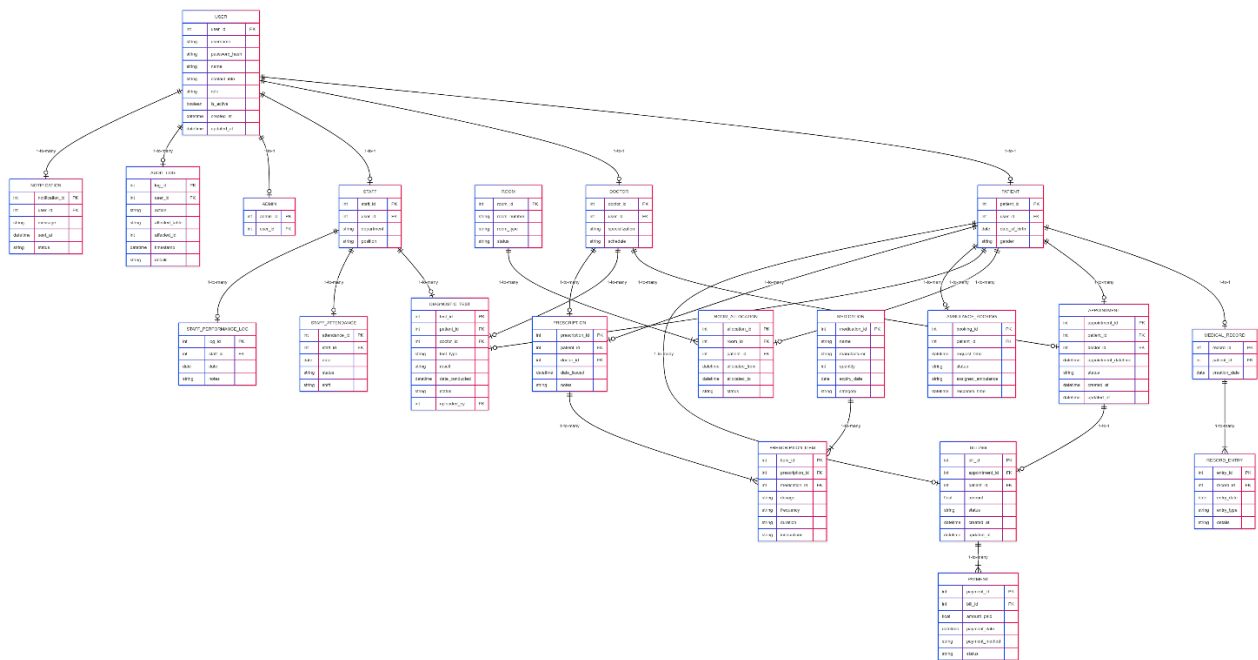


Fig 3: Database Schema

The schema is organized into modular tables—each representing a primary entity or relationship in the hospital’s workflow. Foreign key constraints ensure referential integrity, while ENUMs and status flags support workflow management. This normalized design aids clarity and future extensibility for new modules or regulatory requirements.

Entity Tables and Relationships

User and Role Management

- **User**

Stores all system users, supporting multiple roles (patient, doctor, staff, admin).

- user_id (PK)
- username, password_hash, name, contact_info, role, is_active

- created_at, updated_at
- **Patient**
 - patient_id (PK)
 - user_id (FK → User)
 - date_of_birth, gender
- **Doctor**
 - doctor_id (PK)
 - user_id (FK → User)
 - specialization, schedule
- **Staff**
 - staff_id (PK)
 - user_id (FK → User)
 - department, position
- **Admin**
 - admin_id (PK)
 - user_id (FK → User)

Medical Records

- **MedicalRecord**
 - record_id (PK)
 - patient_id (FK → Patient)
 - creation_date
- **RecordEntry**
 - entry_id (PK)
 - record_id (FK → MedicalRecord)
 - entry_date, entry_type, details

Appointments

- **Appointment**
 - appointment_id (PK)
 - patient_id (FK → Patient)
 - doctor_id (FK → Doctor)

- appointment_datetime
- status (ENUM: scheduled, completed, cancelled, no-show)
- created_at, updated_at

Room Management

- **Room**

- room_id (PK)
- room_number, room_type
- status (ENUM: vacant, occupied, maintenance, cleaning)

- **RoomAllocation**

- allocation_id (PK)
- room_id (FK → Room)
- patient_id (FK → Patient)
- allocated_from, allocated_to
- status (ENUM: active, completed)

Diagnostic Services

- **DiagnosticTest**

- test_id (PK)
- patient_id (FK → Patient)
- doctor_id (FK → Doctor)
- test_type, result, date_conducted
- status (ENUM: ordered, in-progress, completed)
- uploaded_by (FK → Staff)

Ambulance Booking

- **AmbulanceBooking**

- booking_id (PK)
- patient_id (FK → Patient)
- request_time
- status (ENUM: requested, dispatched, completed, cancelled)
- assigned_ambulance
- response_time

Prescriptions and Medication

- **Prescription**
 - prescription_id (PK)
 - patient_id (FK → Patient)
 - doctor_id (FK → Doctor)
 - date_issued, notes
- **PrescriptionItem**
 - item_id (PK)
 - prescription_id (FK → Prescription)
 - medication_id (FK → Medication)
 - dosage, frequency, duration, instructions
- **Medication**
 - medication_id (PK)
 - name, manufacturer, quantity, expiry_date, category

Billing and Payment

- **Billing**
 - bill_id (PK)
 - appointment_id (FK → Appointment)
 - patient_id (FK → Patient)
 - amount, status (ENUM: pending, paid, cancelled, overdue)
 - created_at, updated_at
- **Payment**
 - payment_id (PK)
 - bill_id (FK → Billing)
 - amount_paid, payment_date, payment_method (ENUM: cash, card, mobile, insurance)
 - status (ENUM: successful, failed, pending)

Staff Management

- **StaffAttendance**
 - attendance_id (PK)

- staff_id (FK → Staff)
- date, status (ENUM: present, absent, on-leave, late)
- shift
- **StaffPerformanceLog**
 - log_id (PK)
 - staff_id (FK → Staff)
 - date, notes

Audit and Notifications

- **AuditLog**
 - log_id (PK)
 - user_id (FK → User)
 - action, affected_table, affected_id, timestamp, details
- **Notification**
 - notification_id (PK)
 - user_id (FK → User)
 - message, sent_at, status (ENUM: sent, pending, failed)

Design Features

- **Referential Integrity:** All foreign keys enforce relationships between key entities (e.g., each appointment links to a valid patient and doctor).
- **Scalability:** The modular approach enables easy addition of new modules (e.g., laboratory, pharmacy, insurance).
- **Security:** User roles and audit logs support fine-grained access control and traceability.
- **Performance:** Indexed primary and foreign keys allow efficient queries for large datasets.

This relational database schema forms the backbone of the Hospital Management System, ensuring reliable data organization for all core hospital operations—from patient care to financial transactions and resource management. Its structure supports robust reporting, auditability, security, and smooth integration with other systems, enabling healthcare providers to deliver effective and efficient services.

CHAPTER 4: TOOLS USED

In the development and management of the Hospital Management System project, a variety of tools and platforms were utilized to enhance collaboration, streamline task management, and support design and documentation activities. Below is an overview of the primary tools and their roles in the project lifecycle:

4.1 Trello

Purpose: Task and Communication Management

Link: [Trello Board - G-3 Hospital Management System](#)

Trello was used extensively as the central platform for tracking tasks, assigning responsibilities, and ensuring transparent project progress. Each module and feature was represented as a card, allowing the team to:

- Organize tasks in columns (e.g., To Do, In Progress, Done) for clear workflow visualization.
- Assign members to specific tasks, set deadlines, and prioritize work.
- Comment directly on cards for module-specific discussions and clarifications.
- Attach relevant resources, such as documents, links, and images, to maintain centralized access to information.
- Track overall project status and milestones at a glance.

This approach fostered effective team collaboration and enabled real-time updates on project activities, reducing miscommunication and improving accountability.

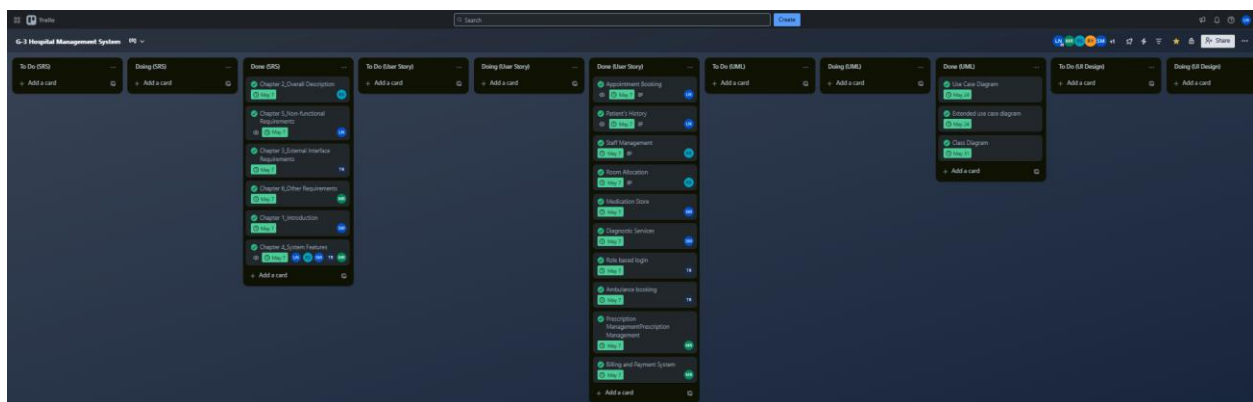


Fig 4: Trello Board

4.2 Discord

Link: [Discord Channel](#)

Purpose: Real-Time Communication and Quick Discussions

Discord served as the team's main channel for instant communication, facilitating:

- Real-time discussions on design decisions and technical challenges.
- Voice and text channels for different sub-teams (e.g., UI/UX, backend, documentation).
- Quick troubleshooting and brainstorming sessions.
- Sharing of resources and meeting reminders.

Using Discord helped bridge communication gaps, ensuring that all team members could quickly resolve issues and coordinate their work, particularly during critical phases of the project.

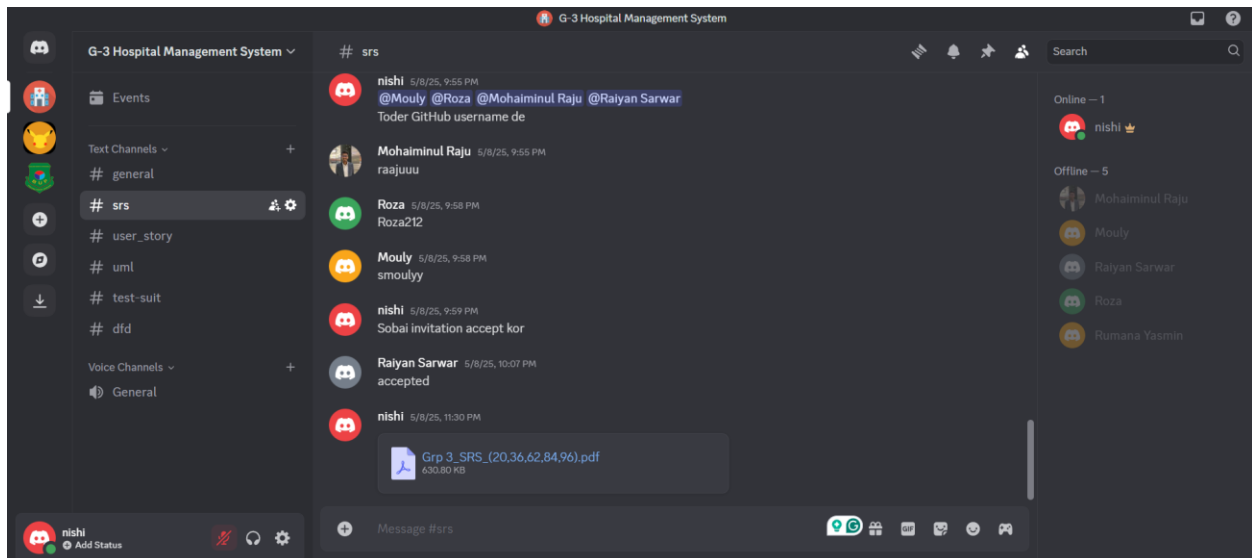


Fig 5: Discord

4.3 UML Diagram Tools

Purpose: System Design and Visualization

Link: [Class Diagram](#)

To properly design and document the system architecture, the team used specialized UML (Unified Modeling Language) diagram tools. These included:

- **draw.io (diagrams.net):** For creating class diagrams, use case diagrams, sequence diagrams, and ERDs.

- **Lucidchart:** For collaborative diagramming, especially when multiple team members needed to contribute simultaneously.
- **StarUML:** For generating and exporting complex UML diagrams in various formats.
- **MermaidChart:** For generating diagrams using the Mermaid syntax, which is easy to maintain and integrate into documentation.

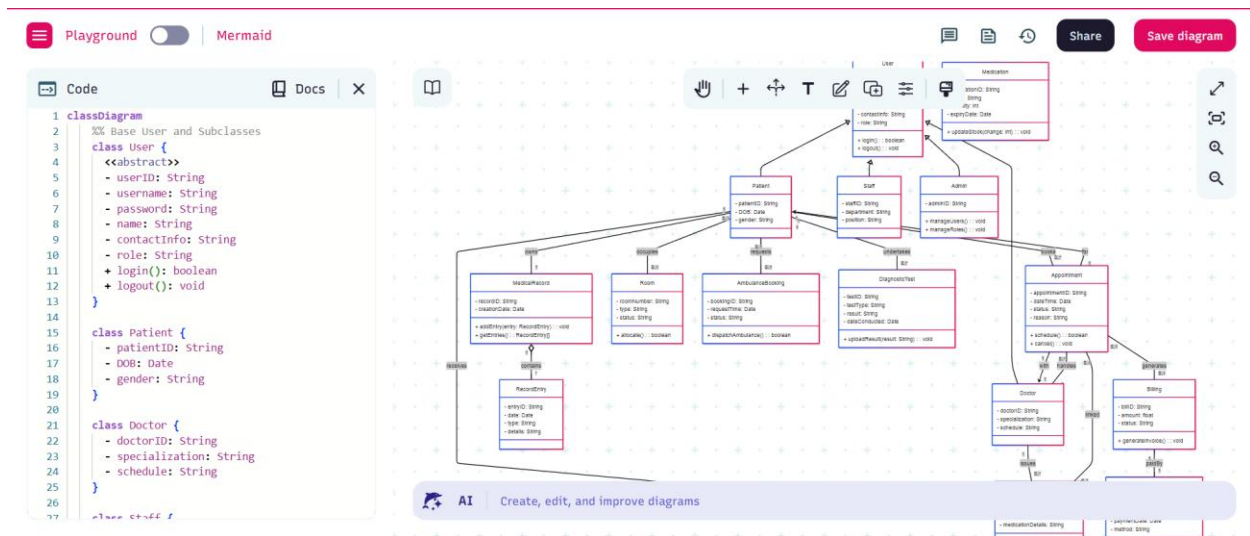


Fig 6: Mermaid Chart

4.4 UI Design Tools

Purpose: User Interface Prototyping and Mockups

For designing and prototyping the user interface of the Hospital Management System, the following tools were used:

- **Figma:** Used for collaborative UI/UX design, prototyping, and sharing interactive mockups with the team. Figma allowed the team to visualize layouts, experiment with different design ideas, and gather early feedback from stakeholders before development began.
- **HTML and CSS:** Used to implement the user interface designs in the web application. HTML provided the structure of each page, while CSS was used to style and layout the UI components according to the designs created in Figma.

These tools enabled the team to:

- These tools and technologies enabled the team to:

- Create user-friendly and intuitive interfaces.
- Ensure consistency in design across different modules.
- Efficiently translate design prototypes into working web pages.

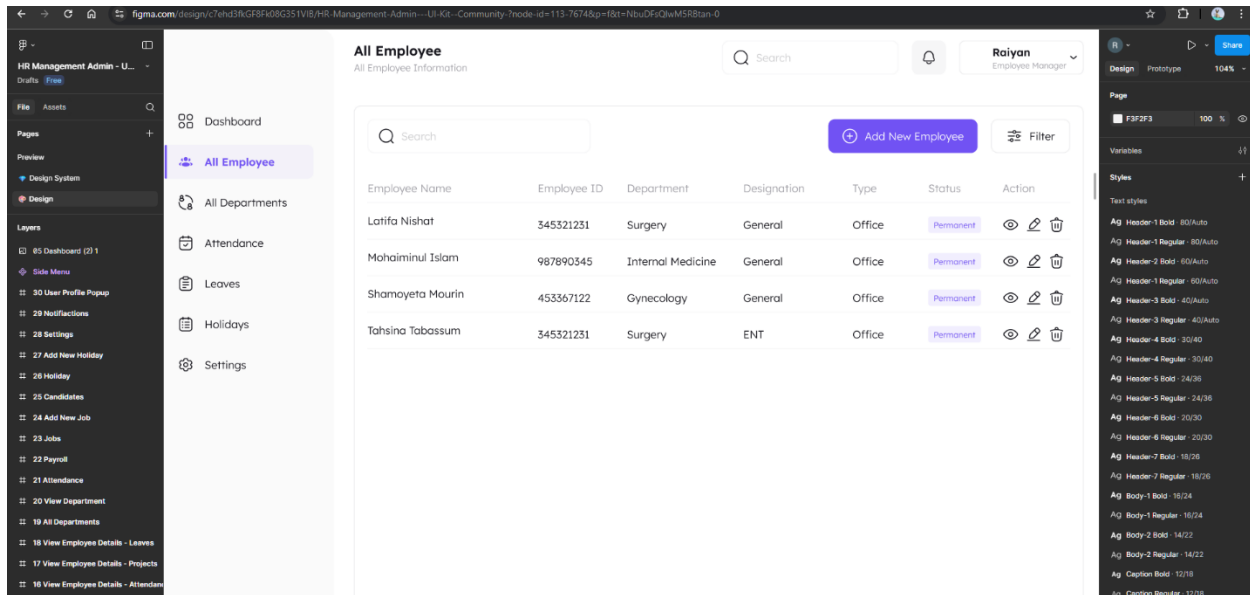


Fig: Figma

The combination of these tools—Trello for task management, Discord for communication, UML diagram tools for system design, and UI design tools for interface prototyping—greatly contributed to the successful execution of the Hospital Management System project. By leveraging these platforms, the team maintained clear communication, organized workflows, and produced well-documented, user-centric deliverables.

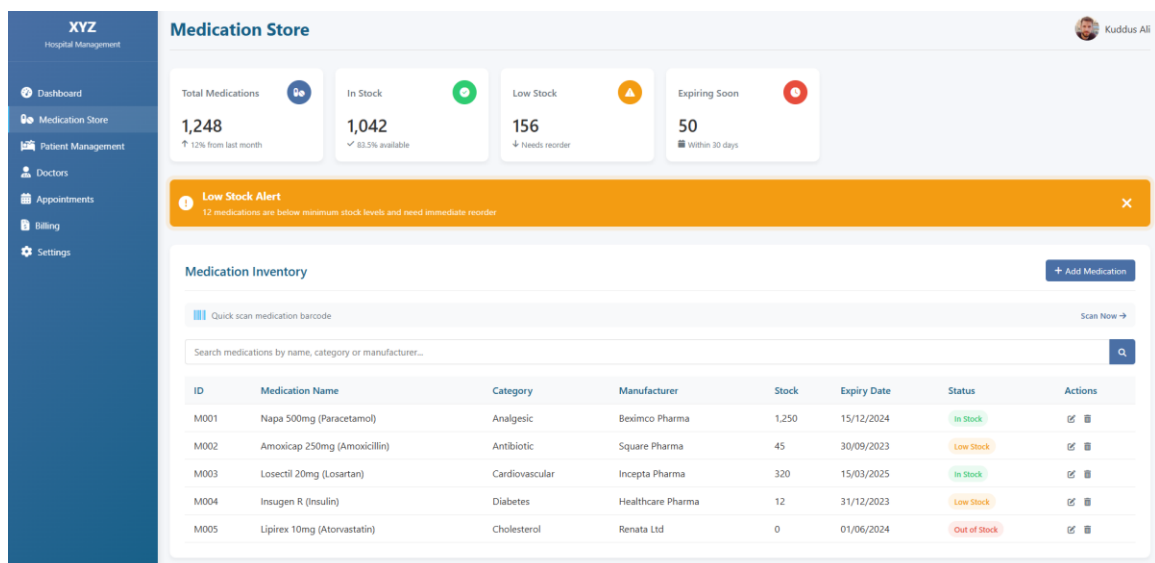
CHAPTER 5: IMPLEMENTATION DETAIL (UI DESIGN AND SCREENSHOTS OF UI)

UI design (User Interface design) is the process of creating visually appealing and user-friendly interfaces for software applications. It focuses on the layout, look, and interactive elements to ensure users can easily navigate and interact with the system, enhancing usability and overall user experience.

1. Billing & Payment System



2. Medication Store



3. Appointment Booking

Patient Dashboard

Logout

Book Appointment

My Appointments

Doctor Schedule

Book an Appointment

Select Specialty:

General Practitioner

Select Doctor:

Dr. A (Mon, Wed, Fri) - Rating: 4.0

Select Date:

dd - - - - - yyyy

Booking for Dependent?

☒

Booking for Dependent?

Available Slots:

10:00 AM

11:00 AM

3:00 PM

Confirm Appointment

Patient Dashboard

Logout

Book Appointment

My Appointments

Doctor Schedule

My Appointments

ID	Date	Time	Doctor	Status	Actions
APT1	2025-06-15	10:00 AM	Dr. A (Mon, Wed, Fri) - Rating: 4.0	Scheduled	<div>Cancel</div> <div>Reschedule</div>
APT2	2025-06-20	2:30 PM	Dr. B (Tue, Thu, Sat) - Rating: 4.5	Rescheduled	<div>Cancel</div> <div>Reschedule</div>

Patient Dashboard

Logout

Book Appointment

My Appointments

Doctor Schedule

Doctor Schedule

Doctor	Specialty	Available Days	Rating
Dr. A	General Practitioner	Mon, Wed, Fri	4.0
Dr. B	Cardiology	Tue, Thu, Sat	4.5
Dr. C	Dermatology	Mon, Thu, Sun	3.8

4. Role Based Login

Hospital Management System

Home About Contact Ambulance

Welcome Back

Access the Hospital Management System to manage patients, appointments, and medical records efficiently.

Your Health Data, Protected

24/7 Access to Medical Data

Sign In

Please sign in to access your account

Select Your Role

Doctor

Patient

Admin

Staff

Username or Email

Password

Remember me

Forgot password?

Sign In

© 2025 Hospital Management System. All rights reserved.

Terms of Service Privacy Policy Support Ambulance

Hospital Management System

Home About Contact Ambulance

Join Our Team

Register to access the Hospital Management System and begin your journey.

Secure Registration

Access Medical Data

Create Account

Please fill out the form to register

Select Your Role

Doctor

Patient

Admin

Staff

Doctor ID

Phone Number

Email Address

Password

Confirm Password

Sign Up

Already have an account? Sign in

© 2025 Hospital Management System. All rights reserved.

Terms of Service Privacy Policy Support

5. Staff Management

Dashboard

All Employee

All Departments

Attendance

Leaves

Holidays

Settings

All Employee

All Employee Information

Search

Notifications

Raiyan Employee Manager

Search

Add New Employee

Filter

Employee Name	Employee ID	Department	Designation	Type	Status	Action
Latifa Nishat	345321231	Surgery	General	Office	Permanent	View Edit Delete
Mohaiminul Islam	987890345	Internal Medicine	General	Office	Permanent	View Edit Delete
Shamoyeta Mourin	453367122	Gynecology	General	Office	Permanent	View Edit Delete
Tahsina Tabassum	345321231	Surgery	ENT	Office	Permanent	View Edit Delete

Showing

04

Showing 1 to 04 out of 20 records

[1](#) [2](#) [3](#) [4](#)

6. Ambulance Booking

Ambulance Booking

Dr. Sarah Haque

Emergency ambulance services are available 24/7. Average response time in Dhaka: 15-20 minutes.

New Ambulance Request

Patient Name

Contact Number

Pickup Location

Destination

Emergency Type

Priority Level

Additional Notes

Request Ambulance

Recent Ambulance Requests

Filter Export

Request ID	Patient Name	Pickup Location	Time	Emergency Type	Priority	Status	Actions
AMB-DHK-1023	Md. Rahman Ali	Mirpur-10, Dhaka	Today, 10:30 AM	Heart Attack	Code Red	In Transit	View Edit
AMB-DHK-1022	Ayesha Begum	Dhanmondi-15, Dhaka	Today, 9:15 AM	Dengue Complications	Code Yellow	Completed	View Edit
AMB-DHK-1021	Kamal Hossain	Savar, Dhaka	Yesterday, 3:45 PM	Road Accident	Code Red	Completed	View Edit
AMB-DHK-1020	Fatema Akter	Uttara Sector-7, Dhaka	Yesterday, 1:20 PM	Childbirth Emergency	Code Red	Completed	View Edit
AMB-DHK-1019	Abdul Karim	Narayanganj	Yesterday, 11:30 AM	Stroke	Code Red	Completed	View Edit

7. Patients' History

Patient History



Dr. Sneha Hauque

Patient records are securely stored and HIPAA compliant.

Search Patients

Patient Name or ID

Search by name or patient ID

Date Range

dd-----yyyy



to

dd-----yyyy



Condition

All Conditions

Reset

Search


Patient Records

+ New Patient

Patient ID	Name	Age	Gender	Blood Type	Last Visit	Conditions	Actions
PT-5001	Shamoyeta Mourin	32	Female	A+	2023-06-15	Hypertension Diabetes	History Edit
PT-5002	Mohaiminul Islam	45	Male	O-	2023-06-14	Asthma Hypertension	History Edit
PT-5003	Naimul Hasan	28	Male	B+	2023-06-10	Migraine Allergies	History Edit

8. Prescription Management

Prescription Management

 Dr. Sneha Haque

⚠️ Always verify patient allergies before prescribing medication.

New Prescription

Patient

Select Patient

Date

dd- - - - - yyyy

Diagnosis

Enter diagnosis

Medications

Medication name

Dosage

Frequency

Duration

+

Paracetamol - 10mg - Once daily - 30 days

Napa - 500mg - Twice daily - 30 days

Additional Instructions

Special instructions for the patient

Print Preview

Save Prescription


Recent Prescriptions


Filter


Export

Prescription ID	Patient	Date	Diagnosis	Medications	Prescribed By	Actions
RX-2023-0789	Latifa Nishat	2023-06-15	Hypertension, Type 2 Diabetes	2 medications	Dr. Sneha Haque	<div>View</div> <div>Print</div>
RX-2023-0788	Tahsina Tabassum	2023-06-14	Asthma exacerbation	3 medications	Dr. Sneha Haque	<div>View</div> <div>Print</div>
RX-2023-0787	Raiyan Sarwar	2023-06-10	Migraine with aura	1 medication	Dr. Sneha Haque	<div>View</div> <div>Print</div>

9. Room Allocation

 Room Allocation

 Dr. Farhana Ahmed

 Currently 15 beds available (out of 60 total beds)

Allocate Room

Select Patient

Select Patient

▼

Room Type

Select Room Type

▼

Admission Date

dd- - - - - yyyy

📅

Expected Discharge

dd- - - - - yyyy

📅

Special Instructions

Any special requirements

Allocate Room

Current Room Status

Filter

Print

Room 101 (AC Cabin)

Available

Ward: Medicine

Rate: ₹5000/day

Facilities: AC, TV, Attached Toilet

Assign Patient

Room 205 (Non-AC Cabin)

Occupied

Ward: Surgery

Patient: Mohammad Ali

Admitted: 14-06-2023

View Details

ICU 3

Occupied

Patient: Sumaiya Akter

Admitted: 15-06-2023

Doctor: Dr. Rahman

View Details

Bed 312 (General Ward)

Available

Ward: Orthopedics

Rate: ₹1000/day

Capacity: 4 patients

Assign Patient

10. Diagnostic Services

Diagnostic Services

Dr. Sarah Haque

Our diagnostic center operates 24/7. Reports are typically ready within 6-12 hours.

New Test Request

Patient Name

Enter patient name

Patient ID

Enter patient ID

Test Type

Select Test Type

Priority

Routine

Clinical Indications

Enter clinical indications for this test

Submit Request

Test ID	Patient Name	Test Type	Date	Status	Result	Actions
T-2023-1056	Ayesha Rahman	Complete Blood Count	15-06-2023	Completed	✓ Ready	View Download
T-2023-1055	Kamal Hossain	X-Ray (Chest)	15-06-2023	In Progress	Processing	View
T-2023-1054	Nusrat Jahan	Ultrasound (Whole Abdomen)	14-06-2023	Completed	✓ Ready	View Download

Page | 58

CHAPTER 6: TESTING

A comprehensive test suite is essential for ensuring the reliability, correctness, and robustness of the Hospital Management System (HMS). The test suite is designed to validate all critical functionalities of the system, covering both typical use cases and edge scenarios. It includes a variety of test types—unit tests, integration tests, system tests, and user acceptance tests (UAT)—to thoroughly assess modules such as user authentication, appointment booking, patient record management, billing, diagnostics, medication inventory, and reporting. Automated testing frameworks are leveraged where possible to enable frequent and consistent validation throughout the development lifecycle.

This test suite provides comprehensive coverage of the Hospital Management System's core modules. Each feature (appointment booking, patient history, staff management, room allocation, medication store, diagnostics, role-based login, ambulance booking, prescription management, billing & payment) is validated through dedicated test cases targeting both common and edge scenarios. The suite ensures functional correctness, data integrity, security, error handling, and positive user experience across all workflows.

1. APPOINTMENT BOOKING TEST SUITE

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
AB-1	Valid Appointment Booking	<ul style="list-style-type: none">- Patient is registered and logged in- The chosen doctor and time slot are available	<ol style="list-style-type: none">1. Log in using valid patient credentials.2. Navigate to the appointment booking module.3. Select an available doctor.4. Choose an available date and time slot.5. Fill in all required fields (e.g., reason for visit, contact	<ul style="list-style-type: none">- A confirmation page with appointment details is displayed.- A confirmation email or notification is sent to the patient.- The selected time slot is marked as booked in the schedule.

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
			details). 6. Submit the booking request.	
AB-2	Appointment Booking with Missing Required Fields	- Patient is logged in and accesses the appointment booking form.	1. Open the appointment booking form. 2. Leave one or more required fields (e.g., appointment date/time, contact details) blank. 3. Attempt to submit the form.	- Submission is prevented with error messages indicating which field(s) are missing. - No appointment is created.
AB-3	Booking When the Selected Time Slot Is Unavailable	- The selected time slot is fully booked.	1. Log in as a patient. 2. Navigate to the appointment booking module. 3. Select a doctor and choose a time slot that is already booked. 4. Complete the booking form and submit.	- The system notifies the patient that the time slot is unavailable and prompts for an alternative.
AB-4	Modification of an Existing Appointment	- A valid appointment exists in the patient's history.	1. Log in and access the appointment history. 2. Select the	- The updated appointment is validated, applied, and

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
		- Patient is logged in.	appointment to modify. 3. Edit the desired fields (e.g., new time or doctor). 4. Submit the modifications.	confirmed via notification.
AB-5	Appointment Cancellation	- A cancelable appointment exists. - Patient is logged in.	1. Log in and navigate to appointment history. 2. Select the appointment to cancel. 3. Initiate and confirm cancellation.	- The appointment is marked as cancelled with the time slot released for rebooking. - A cancellation confirmation is sent.
AB-6	Prevention of Overlapping Appointments	- The patient already has an appointment. - An attempt is made to book a second appointment with overlapping timing.	1. Log in as a patient. 2. Attempt to create a new appointment that overlaps with the existing one. 3. Submit the booking request.	- A warning indicates the scheduling conflict, and the new booking is not processed until resolved.

2. PATIENT HISTORY TEST SUITE

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
PH-1	Retrieving Patient History with Date Range Filtering	<ul style="list-style-type: none"> - Multiple history records exist for a patient, spanning various dates. 	<ol style="list-style-type: none"> 1. Log in with appropriate credentials (patient or authorized staff). 2. Navigate to the Patient History module. 3. Set a start and end date in the filter options. 4. Execute the search/filter. 	<ul style="list-style-type: none"> - Only the records falling within the specified date range are displayed, correctly ordered by date.
PH-2	Unauthorized Access Attempt for Patient History Data	<ul style="list-style-type: none"> - A patient history exists; a user with insufficient privileges attempts access. 	<ol style="list-style-type: none"> 1. Log in using credentials that lack permission (e.g., an unverified user or one with a restricted role). 2. Attempt to access a patient's history. 	<ul style="list-style-type: none"> - The system denies access, displays a "Permission Denied" or similar error message, and logs the attempt for security auditing.
PH-3	Exporting Patient History in Multiple Formats	<ul style="list-style-type: none"> - The patient has multiple history records. - The user has export privileges. 	<ol style="list-style-type: none"> 1. Log in and navigate to the Patient History section. 2. Select the export option and choose PDF 	<ul style="list-style-type: none"> - Both exported files are generated successfully and contain all relevant history data in the correct format.

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
			format. Save or view the output. 3. Repeat the export process for CSV format.	
PH-4	Handling New or Empty Patient History	- A new patient record exists with no historical entries.	1. Log in and open the Patient History module for the new patient. 2. Observe how the system handles the empty state.	- A friendly message is displayed (e.g., "No records found. Please add a new entry."), along with suggestions or links to help begin the record-keeping process.
PH-5	Editing an Existing Patient History Record	- The patient history contains at least one record that is eligible for editing. - The user (doctor/authorized staff) is logged in with editing privileges.	1. Log in and access the specific patient history record. 2. Select the record to be edited and click "Edit". 3. Modify the necessary fields (e.g., update diagnosis details or treatment information). 4. Submit the changes.	- The updated record is saved, and the changes are immediately reflected in the patient history. - An audit log entry is created (if applicable) to track the update.

3. STAFF MANAGEMENT TEST SUITE

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
SM-1	Valid Staff Member Addition	- An admin is logged in with access to the Staff Management module.	<ol style="list-style-type: none"> 1. Navigate to the Staff Management panel. 2. Click "Add New Staff". 3. Enter all required details (name, department, position, contact info). 4. Submit the form. 	- A new staff member is added to the system, the list updates, and an audit log is recorded.
SM-2	Staff Role Modification	<ul style="list-style-type: none"> - Staff member exists in the system. - Admin is logged in. 	<ol style="list-style-type: none"> 1. In the Staff Management module, select an existing staff member. 2. Change the role or department. 3. Confirm and submit the changes. 	- The staff member's details are updated and a confirmation is logged.
SM-3	Invalid Staff Data Submission	- Admin is on the "Add New Staff" form.	<ol style="list-style-type: none"> 1. Leave required fields blank or enter invalid data (e.g., numeric values in the name field). 	- Error messages are displayed and no record is created until corrections are made.

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
			2. Attempt submission.	
SM-4	Deletion of a Staff Member	<ul style="list-style-type: none"> - A staff member exists. - Admin is logged in. 	<ol style="list-style-type: none"> 1. Navigate to the Staff Management module. 2. Select an existing staff member. 3. Click "Delete" and confirm the action. 	<ul style="list-style-type: none"> - The staff member is removed, and the deletion is logged.
SM-5	Staff Search and Filter Performance	<ul style="list-style-type: none"> - Multiple staff records exist. - Admin is logged in. 	<ol style="list-style-type: none"> 1. Use the search functionality to filter staff based on specific criteria (name or department). 	<ul style="list-style-type: none"> - Matching staff records are displayed promptly and accurately.

4. ROOM ALLOCATION TEST SUITE

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
RA-1	Valid Room Allocation	<ul style="list-style-type: none"> - A patient requires room allocation and at least one room is available. - Administrator or receptionist is logged in. 	<ol style="list-style-type: none"> 1. Navigate to the Room Allocation module. 2. Select an available room. 3. Assign the room to the 	<ul style="list-style-type: none"> - The room's status is updated to "occupied," and the patient's record reflects the assignment.

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
			patient. 4. Submit the allocation request.	
RA-2	Room Unavailability Handling	- A specific room is already occupied.	1. Attempt to allocate the same room to another patient.	- An error message indicates the room is unavailable; allocation is halted.
RA-3	Updating Room Status After Discharge	- A patient is occupying a room.	1. Mark the patient as discharged. 2. Update the room allocation status.	- The room status changes from "occupied" to "available".
RA-4	Room Cleaning Status Update	- A room is marked as occupied or under maintenance.	1. A housekeeping user logs into the Room Allocation module. 2. Update the status to "clean" after servicing the room.	- The room status is updated and notifications are sent to reception if needed.
RA-5	Special Room Request Handling (e.g., ICU)	- A patient requires admission to a special room and	1. Initiate a room allocation request specifying ICU or special requirements.	- The system prioritizes and assigns an ICU room according

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
		such rooms exist with priority rules.	2. Submit the request.	to availability and workflow rules.

5. MEDICATION STORE TEST SUITE

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
MS-1	Valid Medication Stock Update	<ul style="list-style-type: none"> - A medication exists in inventory. - A pharmacist is logged in. 	<ol style="list-style-type: none"> 1. Navigate to the Medication Inventory module. 2. Select a medication and enter the restocked quantity. 3. Submit the update. 	<ul style="list-style-type: none"> - The medication quantity is updated, and a confirmation message is displayed.
MS-2	Low Stock Alert Generation	<ul style="list-style-type: none"> - A medication's quantity is below the minimum threshold. 	<ol style="list-style-type: none"> 1. Verify the inventory dashboard for low-stock alerts. 	<ul style="list-style-type: none"> - An alert or notification highlights the low stock, prompting reorder.
MS-3	Prevent Dispensing Over Available Stock	<ul style="list-style-type: none"> - Medication exists with a known quantity. 	<ol style="list-style-type: none"> 1. Initiate a medicine dispensing operation requesting a 	<ul style="list-style-type: none"> - The system rejects the request with an error message about insufficient stock.

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
			quantity greater than available.	
MS-4	Barcode Scanning Integration	- Barcode scanning is enabled and medications have valid barcodes.	1. Scan the barcode during dispensing or inventory update.	- The medication is correctly identified and record updates occur as expected.
MS-5	Medication Inventory Report Generation	- Multiple medication records exist. - The pharmacist or admin is logged in.	1. Click on the "Generate Report" option in the inventory module. 2. Select parameters such as date range or category.	- A report (PDF/Excel) is generated that accurately reflects inventory details.

6. DIAGNOSTIC SERVICES TEST SUITE

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
DS-1	Schedule Diagnostic Test	- Patient record exists and doctor is logged in.	1. Access the Diagnostic Services module. 2. Select a patient and test type along with a designated date/time.	- The test is scheduled, a test ID is generated, and the patient is notified.

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
			3. Submit the test order.	
DS-2	Successful Upload of Diagnostic Report	<ul style="list-style-type: none"> - A diagnostic test is scheduled. - Lab technician is logged in. 	<ol style="list-style-type: none"> 1. Locate the scheduled test in the Diagnostic Services module. 2. Upload the diagnostic report file. 	<ul style="list-style-type: none"> - The report is attached to the test order, a timestamp is recorded, and a confirmation is sent.
DS-3	Test Status Tracking	<ul style="list-style-type: none"> - A diagnostic test order is active. 	<ol style="list-style-type: none"> 1. Monitor the test order status as it changes through its lifecycle. 	<ul style="list-style-type: none"> - The status is updated in real time with appropriate notifications to the patient and doctor.
DS-4	Handling Invalid File Format for Report Upload	<ul style="list-style-type: none"> - A lab technician attempts to upload a diagnostic report. 	<ol style="list-style-type: none"> 1. Attempt to upload a file in an unsupported format (e.g., .exe or .txt). 	<ul style="list-style-type: none"> - An error message is displayed, and the file is rejected with a prompt to use a valid format.
DS-5	Diagnostic Test Rescheduling	<ul style="list-style-type: none"> - A diagnostic test order exists with a scheduled date/time. - The remediating user (doctor or lab coordinator) is logged in. 	<ol style="list-style-type: none"> 1. Select a scheduled test order in the Diagnostic Services module. 2. Choose a different date/time. 3. Resubmit the 	<ul style="list-style-type: none"> - The test order is updated with the new schedule and notifications are sent to the relevant parties.

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
			scheduling details.	

7. ROLE-BASED LOGIN TEST SUITE

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
RL-1	Successful Login for a Valid User	- A valid user account exists.	1. Enter correct username and password. 2. Click "Login".	- Successful authentication and redirection to the user-specific dashboard.
RL-2	Login Attempt with Incorrect Credentials	- A valid user account exists.	1. Enter an incorrect username or password. 2. Attempt to log in.	- An error message (e.g., "Invalid credentials") is displayed, and access is denied.
RL-3	Role-Specific Access Control	- A user logged in (e.g., a patient) does not have permissions to access administrative modules.	1. Attempt to navigate to an admin or doctor-specific URL/module.	- Access is blocked with an "Access Denied" message, and the action is logged.

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
RL-4	Password Recovery Functionality	- The user has a registered email or phone number.	<ol style="list-style-type: none"> 1. Click on "Forgot Password" on the login page. 2. Enter the registered contact information. 3. Submit the request. 	- A password reset link or code is sent, allowing the user to recover access.
RL-5	Session Timeout Handling	- A user is logged in.	<ol style="list-style-type: none"> 1. Leave the session idle for the configured timeout duration (e.g., 15 minutes). 	- The system automatically logs out the user, requiring a fresh login.

8. AMBULANCE BOOKING TEST SUITE

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
AB-1	Valid Emergency Ambulance Booking	<ul style="list-style-type: none"> - Patient/caregiver is logged in. - At least one ambulance is available. 	<ol style="list-style-type: none"> 1. Navigate to the Ambulance Booking module. 2. Select "Emergency" and complete the required fields (location, contact details). 	- The system confirms a dispatch of the ambulance with immediate notification to the user.

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
			3. Submit the request.	
AB-2	Scheduled Ambulance Booking	- Patient or staff is planning a scheduled transfer.	<ol style="list-style-type: none"> 1. Choose "Scheduled" mode in the Ambulance Booking module. 2. Select a future date/time and provide necessary details. 3. Submit the booking request. 	- The booking is recorded for the specified time, and a confirmation is sent.
AB-3	Handling No Ambulance Availability	- All ambulances are currently in use (booked or en route).	<ol style="list-style-type: none"> 1. Attempt an ambulance booking during a high-demand period. 	- An error message is displayed stating "No ambulance available at this time," with alternative suggestions if applicable.
AB-4	Ambulance Booking Cancellation	- A valid ambulance booking exists.	<ol style="list-style-type: none"> 1. Navigate to the Ambulance Booking history. 2. Select the booking to cancel. 3. Confirm cancellation. 	- The booking is marked "Cancelled" and a confirmation is sent.

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
AB-5	Ambulance Rebooking After Cancellation	- A cancelled ambulance booking exists.	<ol style="list-style-type: none"> 1. Cancel an existing ambulance booking. 2. Return to the booking module and initiate a new booking. 3. Provide any updated details and submit. 	- A new booking is created successfully with confirmation, and availability is rechecked.

9. PRESCRIPTION MANAGEMENT TEST SUITE

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
PM-1	Creating a Prescription	- Doctor is logged in and a valid patient record is available.	<ol style="list-style-type: none"> 1. Navigate to the Prescription Management module. 2. Select the patient's profile. 3. Fill in details (medication, dosage, instructions). 4. Submit the new prescription. 	- A prescription with a unique ID is generated and a confirmation is displayed.

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
PM-2	Modifying an Existing Prescription	<ul style="list-style-type: none"> - A prescription exists for the patient. - Doctor is logged in. 	<ol style="list-style-type: none"> 1. Access the patient's prescription history. 2. Select a prescription to modify. 3. Update relevant fields. 4. Submit the changes. 	<ul style="list-style-type: none"> - The prescription is updated, changes are logged, and a confirmation message is displayed.
PM-3	Link Prescription to Medication Inventory	<ul style="list-style-type: none"> - Valid medication data exists in the inventory. 	<ol style="list-style-type: none"> 1. In the prescription creation module, select medications from a provided list (or via barcode scanning). 2. Complete and submit the prescription. 	<ul style="list-style-type: none"> - The prescription displays correct medication details and reflects current inventory status.
PM-4	Download Prescription as PDF	<ul style="list-style-type: none"> - A valid prescription exists. 	<ol style="list-style-type: none"> 1. Navigate to the prescription details page. 2. Click "Download PDF". 	<ul style="list-style-type: none"> - A PDF file is generated and downloaded successfully.
PM-5	Prescription Forwarding to Pharmacy Notification	<ul style="list-style-type: none"> - A valid prescription is issued. 	<ol style="list-style-type: none"> 1. Create and submit the prescription. 2. Verify that the prescription is forwarded 	<ul style="list-style-type: none"> - The pharmacy receives the prescription details and a

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
			automatically to the pharmacy module.	notification is logged.

10. BILLING & PAYMENT SYSTEM TEST SUITE

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
BP-1	Automatic Bill Generation Post-Appointment	- A completed appointment exists with associated services.	<ol style="list-style-type: none"> 1. Complete an appointment. 2. Allow the system to process the billing generation. 	- A billing record with a unique bill ID is generated, and its status is set to "unpaid".
BP-2	Successful Multi-Mode Payment Processing	- A billing record exists for a completed service.	<ol style="list-style-type: none"> 1. Navigate to the Payment section for the bill. 2. Select a payment method (credit card, cash, online transfer). 3. Enter required information and process the payment. 	- The payment is processed successfully, the bill status updates to "paid," and a receipt is generated.
BP-3	Applying Discounts	- A billing record exists; discount/insurance	1. During payment processing, input valid	- The final bill reflects the discount/insurance

Test Case ID	Objective	Preconditions	Test Steps	Expected Outcome
	and Insurance	options are available.	discount/insurance codes. 2. Submit and recalculate the amount.	adjustment accurately and the updated invoice is displayed.
BP-4	Handling Payment Gateway Failures	- A billing record exists; simulate an error (e.g., incorrect payment info or gateway outage).	1. Initiate the payment with faulty details.	- An error message is displayed, the transaction is aborted, and no update is made to the billing record.
BP-5	Invoice Download and Billing History	- Previous billing records with generated invoices exist.	1. Navigate to the Billing History module. 2. Select a past invoice and click "Download Invoice".	- A downloadable PDF invoice is generated and matches the on-screen billing details.

By systematically validating every major system module and user interaction, this test suite supports a reliable, user-friendly, and compliant Hospital Management System. The feature-wise tabular organization enables focused testing, easier maintenance, and clear traceability to requirements, ensuring all stakeholders' needs are met and the system remains robust as it evolves.

CHAPTER 7: CONCLUSION

The Hospital Management System project represents a significant step toward the digital transformation of healthcare operations. Through a structured approach encompassing requirements analysis, system design, implementation, and comprehensive testing, the team successfully developed a robust and user-friendly platform. The system addresses core hospital needs such as appointment scheduling, patient records, medication management, diagnostics, billing, and more—streamlining processes for both staff and patients.

Key factors contributing to the project's success include:

- Clear definition of functional and non-functional requirements.
- Effective use of collaboration and design tools (Trello, Discord, Figma, UML tools).
- Iterative development and regular stakeholder feedback.
- Thorough testing of all modules to ensure reliability and usability.

The resulting system is scalable, secure, and adaptable to future requirements, providing a foundation for enhanced healthcare service delivery. This project demonstrates the value of teamwork, modern software engineering practices, and the integration of technology in critical domains like healthcare.

References

1. Hospital Management System Project in Software Development (with ER diagrams and concepts) – GeeksforGeeks
<https://www.geeksforgeeks.org/hospital-management-system-project-in-software-development/>
2. Hospital Management System: A Step-by-Step Development Guide – MindInventory
<https://www.mindinventory.com/blog/hospital-management-software-development/>
3. How to Develop Hospital Management System in 2025: The Full Guide – Cleveroad
<https://www.cleveroad.com/blog/how-to-develop-hospital-management-system/>
4. How To Develop Hospital Management System: A Detailed Guide – The Algo
<https://www.the-algo.com/post/how-to-develop-hospital-management-system-a-detailed-guide>
5. Hospital Management Software Development: A Step-by-Step Guide – DjangoStars
<https://djangostars.com/blog/hospital-management-software-development/>
6. Trello Board - G-3 Hospital Management System
<https://trello.com/b/iYxKoPF9/g-3-hospital-management-system>
7. Discord Communication Channel
<https://discord.com/channels/1360408648886522158/1360409297783357703>
8. draw.io (diagrams.net)
<https://www.diagrams.net/>
9. Figma
<https://www.figma.com/>
10. MermaidChart Project Workspace
<https://www.mermaidchart.com/>
11. LucidChart
<https://www.lucidchart.com/pages>
12. Course lecture notes and recommended software engineering textbooks.
13. Software Engineering (10th Edition) by Ian Sommerville.
ISBN: 978-0133943030

Contribution Report

Team Member	Major Topic(s) Contributed
Mohaiminul Islam Raju (2252421020)	Trello & Discord documentation, SRS (Chapter 4: System Features, Chapter 6: Other Requirements), UI Design & User Story (Billing & Payment System, Prescription Management), Database Schema, Class Diagram
Shamoyeta Mourin Mouly (2252421036)	Trello & Discord documentation, Presentation, SRS (Chapter 1: Introduction, Chapter 4: System Features), UI Design & User Story (Medication Store, Diagnostic Services), Database Schema, Test Suite, DFD
Latifa Nishat Nishi (2252421062)	Project Overview, Trello & Discord documentation, Final Report, SRS (Chapter 4: System Features, Chapter 5: Non-functional Requirements), UI Design & User Story (Appointment Booking, Patient's History), Use Case Diagram, Extended Use Case, Test Suite
Tahsina Tabassum Roza (2252421084)	Trello & Discord documentation, Presentation, SRS (Chapter 3: External Interface Requirements, Chapter 4: System Features), UI Design & User Story (Role-Based Login, Ambulance Booking), Class Diagram, DFD, Test Suite
Raiyan Sarwar (2252421096)	Trello & Discord documentation, SRS (Chapter 2: Overall Description, Chapter 4: System Features), UI Design & User Story (Room Allocation, Staff Management), Database Schema, Class Diagram