Software Requirements Specification

for

<Hospital Management System>

Version 1.0 approved

Prepared by

<Mohaiminul Raju 2252421020

Shamoyeta Mourin Mouly 2252421036

Latifa Nishat Nishi 2252421062

Tahsina Tabassum Roza 2252421084

Raiyan Sarwar 2252421096>

<Bangladesh University of Professionals>

<7 May, 2025>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document outlines the software requirements for the **Hospital Management System (HMS), Version 1.0**. The HMS aims to automate the operational workflows within a hospital by automating administrative and clinical activities such as patient registration, doctor appointments, laboratory and pharmacy management, billing, and report generation. This SRS specifies the functionalities and constraints of the system and will serve as a reference for developers, testers, project managers, and stakeholders ensuring that the final product meets the needs of patients, healthcare providers, and administrative staff.

1.2 Document Conventions

- Requirements are labeled and numbered for easy identification.
- Keywords are bolded such as **shall**, **should**, and **may** denote levels of necessity.
- Functional requirements are listed with REQ-IDs (e.g., REQ-1, REQ-2).
- Diagrams and models will follow standard UML notations where applicable.

1.3 Intended Audience and Reading Suggestions

This document is intended for:

- **Developers** – to understand what features and behaviors to implement
- **Project Managers** – to plan and monitor project progress
- **Hospital Administration** – to understand the system capabilities
- **Regulatory Bodies or Auditors** – to review compliance with healthcare standards
- **Testers** – to validate the implementation against requirements

Readers are encouraged to begin with the Introduction (Section 1) and Overall Description (Section 2) before diving into the detailed requirements (Section 3).

1.4 Product Scope

The Hospital Management System (HMS) is a web-based software solution designed to digitalize and automate the core activities of hospital operations. The system will support tasks such as patient registration, appointment scheduling, laboratory and pharmacy management, doctor and staff coordination, billing, and medical report generation.

The main purpose of the HMS is to enhance operational efficiency, reduce manual workload, minimize errors, and ensure data accuracy across all departments. By providing centralized access to patient data and hospital records, the software ensures faster decision-making and improved patient care.

This product is aligned with institutional goals of adopting smart healthcare solutions and reducing administrative overhead. It helps hospitals transition toward paperless workflows, comply with healthcare regulations, and better allocate resources. If a separate Vision and Scope document is maintained for this project, it will detail long-term goals, scalability plans, and integration expectations beyond this SRS.

1.5 References

The following documents and resources have been referred to during the creation of this SRS:

1. IEEE Std 830-1998 – IEEE Recommended Practice for Software Requirements Specifications.
   Publisher: IEEE Standards Association, Version: 1998.
   URL: https://ieeexplore.ieee.org/document/720574
2. Hospital Management System – Vision and Scope Document,
   Prepared by: Project Team HMS, Version 1.0, April 2025.
   Location: Internal project repository.
3. Use Case Document for HMS,
   Author: Development Team, Version 1.1, March 2025.
   Location: /project_docs/hms/use_cases.docx
4. Government Guidelines for Health Information Systems,
   Issued by: Ministry of Health & Family Welfare, Government of [Your Country], 2023.
   URL: https://mohfw.gov.in

2. Overall Description

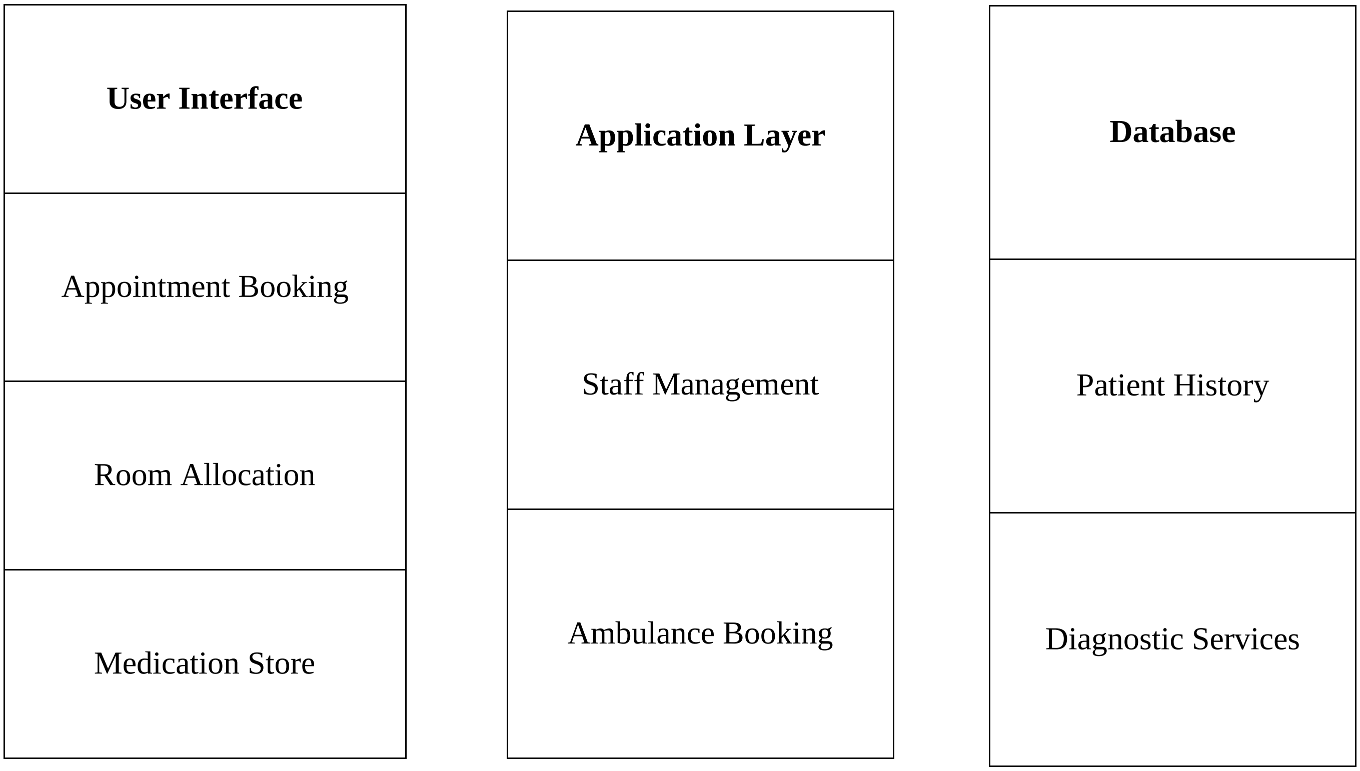
2.1 Product Perspective

The Hospital Management System (HMS) is a newly developed, self-contained software product designed to automate and streamline hospital operations. It replaces traditional manual processes and disparate record-keeping systems with an integrated digital platform that enhances efficiency, accuracy, and accessibility of hospital data.

This system encompasses multiple interrelated modules such as appointment booking, patient history management, staff management, room allocation, role-based login, ambulance booking, diagnostic services, and medication store management. These modules work cohesively to support the full spectrum of hospital administrative and clinical workflows.

The HMS is designed to operate independently but with the potential for future integration with external healthcare systems such as insurance providers, government health databases, and laboratory equipment. This modular approach ensures scalability and adaptability to the evolving needs of healthcare institutions.

System Overview Diagram:



User Interface

Appointment Booking

Room Allocation

Medication Store

Application Layer

Staff Management

Ambulance Booking

Database

Patient History

Diagnostic Services

2.2 Product Functions

The HMS will provide the following major functions, each designed to simplify hospital operations and improve user experience:

Appointment Booking:
Patients can schedule, reschedule, or cancel appointments with doctors through an intuitive interface. The system will display available time slots and send automated reminders via SMS or email to reduce no-shows.

Patient History Management:
Comprehensive medical records including patient demographics, previous visits, diagnoses, treatments, and test results will be securely stored and accessible to authorized medical personnel. This ensures continuity of care and informed decision-making.

Staff Management:
The system will maintain detailed profiles of hospital staff, including roles, schedules, qualifications, and attendance records. It will facilitate shift planning and ensure adequate staffing levels.

Room Allocation:
Based on patient needs and room availability, the system will assign rooms efficiently. It will track room occupancy status in real-time to optimize resource utilization.

Role-Based Login:
Secure authentication mechanisms will restrict access to system features according to user roles such as patient, doctor, nurse, pharmacist, or administrator, ensuring data privacy and compliance with healthcare regulations.

Ambulance Booking:
Patients or hospital staff can request ambulance services during emergencies. The system will manage ambulance availability, dispatch details, and track ambulance location if integrated with GPS.

Diagnostic Services:
Medical staff can request diagnostic tests, view test results, and generate reports within the system. This module will support various diagnostic departments such as radiology, pathology, and lab tests.

Medication Store:
The pharmacy module will manage medicine inventory, track stock levels, process prescriptions, and record medicine dispensing. Alerts for low stock and expiry dates will help maintain optimal inventory.

2.3 User Classes and Characteristics

- Patients:
  - Primary users for appointment booking, ambulance requests, and viewing personal medical history.
  - May have varying levels of technical proficiency; the interface is designed to be user-friendly and accessible on multiple devices including smartphones and desktops.
- Doctors:
  - Access patient records, update medical histories, request diagnostics, and manage appointments.
  - Require quick, secure access to critical patient data and clinical tools.
- Nurses and Support Staff:
  - Assist with patient care by managing room allocations, administering medications, and facilitating diagnostic procedures.
  - Use the system to update patient status and coordinate with other departments.
- Pharmacists:
  - Manage medication inventory, process prescriptions, and ensure accurate dispensing.
  - Require detailed inventory tracking and reporting capabilities.
- Administrators:
  - Oversee hospital operations including staff management, system configuration, and reporting.
  - Require comprehensive access to all modules with tools for monitoring system usage and generating operational reports.

2.4 Operating Environment

Hardware:

The HMS will be hosted on hospital servers with sufficient processing power and storage capacity to handle concurrent users and large volumes of data.

Client devices include desktops, laptops, tablets, and smartphones used by hospital staff and patients.

Software:

The system will be a web-based application accessible through modern browsers such as Chrome, Firefox, and Edge.

The backend will run on stable server operating systems like Linux (Ubuntu, CentOS) or Windows Server.

A relational database management system (e.g., MySQL, PostgreSQL) will store all persistent data securely.

Network:

The system will operate within the hospital's secure local area network (LAN) with controlled internet access for remote patient use.

Network security protocols such as firewalls and VPNs will be employed to safeguard data transmission.

2.5 Design and Implementation Constraints

Regulatory and Compliance Constraints

The system must comply with data protection regulations such as the *Health Insurance Portability and Accountability Act (HIPAA)*, *GDPR*, or any relevant local data privacy laws to ensure the confidentiality and security of sensitive patient data.

Logging and audit trails must be maintained to monitor system access and modification of patient records, in accordance with medical data handling standards.

Hardware Limitations

The system will be deployed in environments with mid-range computing hardware; hence, the application must be optimized for performance with minimal resource consumption.

Server-side requirements must not exceed the capabilities of standard cloud or on-premise hosting environments used by medium-scale healthcare providers.

Software and Technology Stack Constraints

Front-end technologies: HTML5, CSS3, JavaScript (with frameworks such as React or Vue.js if needed).

Back-end technologies: Node.js/Express or Java/Spring Boot.

Database systems: MySQL or MongoDB, depending on the preference for structured or semi-structured data.

The software should support RESTful APIs for seamless integration with future third-party healthcare services (e.g., insurance systems, telemedicine platforms).

User Interface Design Standards

The application must follow accessibility standards (e.g., WCAG 2.1) to accommodate users with disabilities.

Consistent UI/UX design patterns should be applied across all modules to ensure usability by non-technical staff (e.g., hospital front-desk personnel, patients).

Security Constraints

All user roles must be authenticated using secure, role-based access controls (RBAC).

Data in transit must be encrypted using HTTPS and secure protocols such as TLS 1.2+.

Sensitive patient and staff data must be encrypted at rest using industry-standard algorithms (e.g., AES-256).

Development and Maintenance Constraints

The software must be developed in a modular and extensible manner to accommodate future enhancements without major structural changes.

Code must adhere to industry-standard programming conventions and be well-documented for ease of maintenance.

Version control using Git and CI/CD pipelines should be enforced for collaborative development and continuous deployment.

Operational Constraints

The system should support concurrent access by multiple users (e.g., doctors, administrative staff, patients) without performance degradation.

Downtime during updates or maintenance must be minimized and preferably scheduled during non-peak hours.

Integration Constraints

The application must interface with third-party services such as diagnostic equipment, SMS gateways for appointment reminders, and pharmacy management tools via APIs.

Data interoperability should be ensured using standard data formats like JSON or XML.

Language and Localization Constraints

The initial version of the application will be developed in English. However, provisions must be made to support multilingual interfaces in the future, including right-to-left language support.

2.6 User Documentation

User Manuals:
Detailed step-by-step guides tailored for each user class (patients, doctors, nurses, administrators) explaining how to use system features effectively.

Online Help:
Context-sensitive help available within the application to assist users during their workflow.

Video Tutorials:
Short, focused videos demonstrating key tasks such as booking appointments, managing patient records, and processing prescriptions.

FAQs and Troubleshooting:
A collection of frequently asked questions and solutions to common issues to support users in self-service.

2.7 Assumptions and Dependencies

Availability of Stable Internet Connectivity
It is assumed that the client-side and server-side environments will operate in a stable internet environment, especially for features like cloud data backup, online appointment bookings, and SMS/email notifications.

Hardware Compatibility
It is assumed that hospitals and clinics using the system have the minimum required hardware infrastructure (e.g., PCs with modern browsers, tablets for staff, local server or cloud access).

Availability of Technical Staff
It is assumed that the deploying organization will have at least one IT administrator capable of configuring, maintaining, and troubleshooting the system post-deployment.

Standardized Data Format from External Systems
Any integration with third-party services (e.g., diagnostic labs, insurance systems, pharmacy tools) will follow standard APIs or well-documented formats (e.g., HL7, JSON, XML).

User Training Will Be Provided
It is assumed that sufficient training sessions will be arranged for medical and administrative staff to ensure proper utilization of the system.

No Legal or Regulatory Barriers
It is assumed that there are no unanticipated legal or data protection constraints that would prevent the system's implementation in the intended environment.

Single-Hospital Environment (for Phase 1)
The initial system is assumed to support a single hospital or clinic deployment. Multi-location support may be considered in future iterations.

Dependencies:

Third-Party Libraries and Frameworks
The system depends on external open-source or licensed frameworks and tools, such as:

- Node.js/Java Spring Boot (backend)
- React/Vue (frontend)
- MySQL/MongoDB (database)
- External APIs for email/SMS (e.g., Twilio, SendGrid)

Browser and OS Compatibility
The system is dependent on modern web browsers (e.g., Chrome, Firefox, Edge) and operating systems (Windows 10+, Linux distros). Functionality on outdated platforms is not guaranteed.

Authentication Services
The system may integrate with external authentication providers (e.g., OAuth, Google Sign-In) or internal LDAP systems, depending on deployment needs.

Hosting & Deployment Infrastructure
The system assumes availability of either:

- An on-premise server with sufficient resources, or
- Cloud platforms like AWS, Azure, or DigitalOcean for hosting the production environment.

Reuse of Utility Modules
Some backend utilities (such as log management, error handling, user notification services) may be reused from previously developed academic or commercial projects, provided compatibility and licensing compliance.

3. External Interface Requirements

3.1 User Interfaces

The Hospital Management System (HMS) will offer a dynamic and interactive web-based interface that ensures efficient access and usability for different user roles, including administrators, doctors, receptionists, staff, and public users. The interface will be accessed by modern web browsers and designed with consistent visual standards and composition principles for an uninterrupted user experience. The following modules will possess user interfaces:

Login/Authentication Interface:
Secure login for different user roles and error messages for failed login attempts

Dashboard Interface (Role-Based):
Quick links to primary modules

Department Management Interface:
View, add, edit, and delete hospital departments

Medicine Management Interface:
View current stock, low-stock alerts, and reorder tracking

User/Staff Management Interface:
Manage user roles and permissions, add/edit/remove users and credentials

Patient History Interface:
View patient medical history, insert and update patient visit history, diagnosis, and treatment

Appointment Booking Interface:
Book, reschedule, or cancel appointments with doctors, view time slots available and doctor timetables

Room Allocation Interface:
Assign patients to rooms or beds, view room availability and discharge status

Prescription Management Interface:
Generate and manage prescriptions, doctors can select medicine from inventory and prescribe dosage

Public Information Interface:
Display public information like department details, list of doctors, and visiting hours

Report Generation Interface:
Generate reports for departments, stock, appointments, and patients

Logout and Session Interface: Logout option for all users

GUI Standards & Style Guidelines:

Consistent Layout: All screens will have a top navigation menu, a main content section, and a footer.
Responsive Design: UI will be responsive to desktop and tablet resolutions (min 1024x768).
Color Scheme: Blue for main actions, green for success, red for errors/warnings.

Common UI Elements:

Search bars, filters, data tables
Icons for edit, delete, print, and download
Required fields indicated with (*) and simple form validation

Error Messages:

Displayed below input fields or as pop-ups
Red for errors, green for success

Keyboard Shortcuts:

Not implemented currently; may be added in future releases.

3.2 Hardware Interfaces

The Hospital Management System is a web-based application and does not interface directly with hardware devices other than standard input/output and networking devices. However, it does rely on standard hardware infrastructure for deployment and usage. The following are the hardware interface considerations:

Client Devices: Desktop PCs, laptops, tablets, and smartphones

Input/Output Devices: Keyboard, mouse, monitors, and printers (for prescriptions/reports)

Server Requirements:

Quad-core processor, 8GB RAM, 256GB SSD
Network connectivity via LAN or Wi-Fi

Network:

LAN for internal access
Internet for remote/cloud access
Communication over HTTP/HTTPS (preferably HTTPS for security)

3.3 Software Interfaces

The Hospital Management System interacts with several key software components to execute its functionalities effectively. They include the operating system, database management system, web technologies, libraries, and external tools. The key software interfaces are as follows:

Operating System:
Preferably Windows 10+ or Linux (Ubuntu 20.04+)

Database Management System:
MySQL is used to store and manage data related to patients, appointments, inventory, rooms, reports, and prescriptions.

- Input: Patient info, appointments, prescriptions, stock updates, etc.
- Output: Records, availability status, reports, etc.
- Communication: SQL queries via backend scripts (e.g., PHP)

Web Technologies:

- Frontend: HTML5, CSS3, JavaScript
- Backend: PHP (or Node.js, as applicable)
- Web Server: Apache HTTP Server
- Communication: Data exchanged over HTTP/HTTPS

3.4 Communications Interfaces

The Hospital Management System will follow standard communication protocols for data transmission between the server and users. The system will be accessed through web browsers using the HTTP/HTTPS protocol. Form data (e.g., patient registration, appointment booking) will be submitted through HTML forms using the POST method.

Email support may be added to send confirmations of appointments through SMTP. The data in transit would be protected using HTTPS. The supported message formats would be plain text or HTML. The system would operate over a TCP/IP network with a speed to accommodate concurrent users. There is no need for any external synchronization mechanisms or FTP.

4. System Features

This document organizes the functional requirements by system features. Each feature is broken down into a brief description with priority ratings, stimulus/response sequences (illustrating user interactions), and a list of detailed functional requirements written in user story format with confirmation (acceptance criteria) statements. Each requirement is uniquely identified for traceability.

4.1 Appointment & Scheduling System

4.1.1 Description and Priority

Description: This feature enables patients, caregivers, and hospital staff to manage appointments online—from booking and viewing real-time calendar slots to rescheduling, cancellation, and real-time administrative adjustments. It streamlines patient flow, improves resource allocation, and integrates with billing and external calendar systems.

Priority: High Component Ratings (Example):

- Benefit: 9
- Penalty for Non-Compliance: 9
- Implementation Cost: 6
- Risk: 8

4.1.2 Stimulus/Response Sequences

- **Patient Booking: Stimulus:** A patient navigates to the booking module, selects a date, time, and doctor, then submits the appointment request. **Response:** The system validates the request, ensures there are no conflicts, reserves the slot, and sends a confirmation via email/SMS.
- **Rescheduling/Cancellation: Stimulus:** A patient or caregiver initiates a cancellation or rescheduling action from their user portal. **Response:** The system updates the appointment status, frees the previously reserved slot, adjusts other connected schedules, and sends updated notifications.
- **Real-Time Management: Stimulus:** A receptionist or doctor logs into their dedicated dashboard. **Response:** The system presents an up-to-date view of current appointments, enabling immediate edits, confirmations, or cancellations.

4.1.3 Functional Requirements

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.

4.2 Patient History / Electronic Health Record (EHR) Management

4.2.1 Description and Priority

Description: This feature manages comprehensive electronic health records by enabling secure access, updates, and sharing of patient data among authorized healthcare professionals. It supports multimedia attachments, role-based access controls, audit logging, and data integration from various sources, ensuring complete, accurate, and privacy-compliant records.

Priority: High Component Ratings (Example):

- Benefit: 9
- Penalty for Non-Compliance: 9
- Implementation Cost: 7
- Risk: 8

4.2.2 Stimulus/Response Sequences

- **Doctor Record Access: Stimulus:** A doctor selects a patient's record on the EHR interface. **Response:** The system retrieves and displays the patient's comprehensive history—including diagnoses, lab results, prescriptions, and clinical notes—in a consolidated view.
- **Patient Record Update**: **Stimulus:** A patient uploads a new test result or prescription via their portal. **Response:** The system validates the upload, attaches the document to the correct record section, and displays a confirmation message.
- **Emergency Quick View**: **Stimulus:** An emergency responder activates the emergency access mode. **Response:** The system immediately displays a condensed view with critical information (e.g., allergies, medications, chronic conditions), bypassing non-essential details while ensuring secure access.

4.2.3 Functional Requirements

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.

4.3 Medication Store

4.3.1 Description and Priority

Description: This feature enables the hospital staff to manage medication inventory, including adding, updating, and tracking medicines. It ensures real-time stock updates and alerts for low-stock or expired medicines.

Priority: High Component Ratings (Example):

- Benefit: 9
- Penalty (if fails): 8
- Implementation Cost: 3
- Risk: 2

4.3.2 Stimulus/Response Sequences

- **Medication Inventory Access:** Stimulus: A pharmacist logs into the system and selects the inventory management module. **Response:** The system displays the current stock levels of all medicines, categorized by type, brand, and expiry date.
- **Add New Stock Entry:** Stimulus: A pharmacist enters details of a new batch of medicines, including name, quantity, batch number, and expiry date. **Response:** The system validates the inputs, updates the inventory, and confirms successful stock addition.
- **Dispense Medication:** Stimulus: A doctor or nurse selects a patient and prescribes medication through the system. **Response:** The system deducts the prescribed amount from the inventory and logs the transaction under the patient's record with dosage and time details.
- **Low Stock Alert:** Stimulus: Inventory of a particular medication falls below the pre-set threshold. **Response:** The system triggers an automatic alert to the admin and pharmacy staff, prompting reorder actions.

4.3.3 Functional Requirements

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.

4.4 Diagnostic Services

4.4.1 Description and Priority

Description: This feature allows patients to book diagnostic tests, and enables lab staff to manage test schedules, update reports, and notify patients.

Priority: High Component Ratings (Example):

- Benefit: 8
- Penalty (if fails): 7
- Implementation Cost: 4
- Risk: 3

4.4.2 Stimulus/Response Sequences

- **Test Scheduling:** Stimulus: A doctor places an order for a diagnostic test for a patient. Response: The system schedules the test, assigns it to an available lab technician, and notifies the patient.
- **Test Report Upload:** Stimulus: A lab technician completes a diagnostic test and uploads the result. Response: The system stores the result, links it to the patient's record, and makes it accessible to the assigned doctor.
- **Test Result Review:** Stimulus: A doctor accesses a patient's file to review diagnostic results. Response: The system displays all past and current diagnostic reports, clearly sorted and dated.
- **Billing Generation:** Stimulus: A diagnostic test is confirmed and completed. Response: The system automatically generates a billing entry linked to the test.

4.4.3 Functional Requirements

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.

4.5 Ambulance booking

4.5.1 Description and Priority

Description: This feature facilitates on-time ambulance service bookings for patients and healthcare professionals through a secure, user-friendly interface. It facilitates real-time monitoring of availability, priority dispatching (e.g., emergency vs. non-emergency). The system offers timely, transparent, and well-documented ambulance logistics.

Priority: High Component Ratings (Example):

- Benefit: 8
- Penalty for Non-Compliance: 7
- Implementation Cost: 6
- Risk: 7

4.5.2 Stimulus/Response Sequences

- **Patient Emergency Request:** Stimulus: The caregiver or patient requests an emergency ambulance via the web or mobile portal.

Response: The system identifies the nearest available ambulance, classifies the level of emergency, dispatches the vehicle, and notifies the medical staff.

- **Scheduled Transfer Request:** Stimulus: A member of hospital staff requests an ambulance for a scheduled patient transfer.

Response: The system reserves the appointment, assigns a vehicle according to availability, and sends confirmation to the requester and the driver.

- **Cancellation of Booking:** Stimulus: The requester cancels a booked ambulance service.

Response: The system releases the booked ambulance slot, notifies the driver, and logs the cancellation with a reason.

4.5.3 Functional Requirements

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.

1. **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.

2. **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.

3. **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.

4. **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.

5. **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.

6. **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 .

4.6 Role based login

4.6.1 Description and Priority

Description: This feature ensures secure access to the hospital management system by enforcing role-based login credentials for different user categories (e.g., admin, doctor, nurse, patient, receptionist, lab technician). Each role is assigned specific access rights to features and data, helping maintain privacy, streamline workflows, and prevent unauthorized usage.

Priority: High Component Ratings (Example):

- Benefit: 9
- Penalty for Non-Compliance: 8
- Implementation Cost: 5
- Risk: 6

4.6.2 Stimulus/Response Sequences

- **User Login Attempt:** Stimulus: A user enters their credentials on the login screen.

Response: The system verifies the credentials, determines the user’s role, and grants access to the corresponding dashboard and features.

- **Unauthorized Role Access Attempt**: Stimulus: A user attempts to access a module or function outside their assigned role.

Response: The system denies access, displays an authorization error message, and logs the attempt.

4.6.3 Functional Requirements

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.

4.7 Staff Management

4.7.1 Description and Priority

Description: This feature enables the administration to manage hospital staff efficiently, including recruitment, scheduling, role assignment, and performance tracking. It ensures that all staff details are securely maintained and updated, allowing for effective communication, operational planning, and compliance tracking.

Priority: High Component Ratings (Example):

- Benefit: 9
- Penalty for Non-Compliance: 8
- Implementation Cost: 6
- Risk: 7

4.7.2 Stimulus/Response Sequences

- **Staff Addition:** Stimulus: An admin adds a new staff member via the staff module. Response: The system records all staff details and assigns a unique ID, role, and login credentials.

- **Schedule Viewing:** Stimulus: A staff member checks their upcoming schedule. Response: The system fetches and displays shift timings, role assignments, and relevant alerts.

- **Performance Review:** Stimulus: A supervisor accesses staff performance logs. Response: The system displays historical activity, performance metrics, and incident reports.

4.7.3 Functional Requirements

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.

4.8 Room Allocation

4.8.1 Description and Priority

Description: This feature manages the allocation and tracking of hospital rooms for patients, ensuring efficient room utilization, cleanliness status updates, and department-specific availability. It also supports emergency reallocation and room maintenance alerts.

Priority: High Component Ratings (Example):

- Benefit: 9
- Penalty for Non-Compliance: 9
- Implementation Cost: 5
- Risk: 6

4.8.2 Stimulus/Response Sequences

- **Room Assignment:** Stimulus: A patient is admitted to the hospital. Response: The system assigns an available room based on patient type, availability, and cleanliness status.
- **Emergency Transfer:** Stimulus: A critical patient requires immediate room reassignment. Response: The system finds the nearest available ICU/emergency bed and reallocates resources accordingly.
- **Room Cleaning Update**: Stimulus: Housekeeping updates a room's status. Response: The system marks the room as clean/available or under maintenance and notifies admission staff.

4.8.3 Functional Requirements

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.

4.9 Prescription Management

4.9.1 Stimulus/Response Sequences

Description:
This feature allows doctors to create, edit, and manage digital prescriptions for patients. It streamlines the treatment process by ensuring prescriptions are directly linked to patient records and accessible by pharmacy and diagnostic departments for fulfillment and coordination.

Priority: High
Component Ratings:

- **Benefit:** 9
- **Penalty (if fails):** 8
- **Implementation Cost:** 3
- **Risk:** 3

4.9.2 Stimulus/Response Sequences

Create Prescription

- **Stimulus:** A doctor accesses a patient record during consultation and creates a new prescription.
- **Response:** The system validates entries and stores the prescription under the patient’s profile.

View Prescriptions

- **Stimulus:** A pharmacist or patient opens the prescription section of the system.
- **Response:** The system retrieves and displays prescriptions based on user role with medication details and issue date.

Update/Cancel Prescription

- **Stimulus:** A doctor selects an existing prescription and edits or voids it.
- **Response:** The system logs the changes and updates the prescription history with version tracking.

Link to Pharmacy and Diagnostics

- **Stimulus:** A prescription includes medicine and diagnostic test recommendations.
- **Response:** The system forwards medicine requests to the pharmacy and test requests to the diagnostics module.

4.9.3 Functional Requirements

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.

4.10 Billing and Payment System

4.10.1 Stimulus/Response Sequences

Description:
This feature manages all billing operations for hospital services including consultation, medication, diagnostic tests, and room charges. It automates invoice generation, supports multiple payment methods, and tracks financial records.

Priority: High
Component Ratings:

- **Benefit:** 10
- **Penalty (if fails):** 9
- **Implementation Cost:** 4
- **Risk:** 3

4.10.2 Stimulus/Response Sequences

Generate Bill

- **Stimulus:** A patient completes a consultation, test, or receives medication.
- **Response:** The system compiles all charges and generates a consolidated bill.

Make Payment

- **Stimulus:** A patient opts to pay for the services received.

- **Response:** The system displays the total bill, processes payment via selected method, and issues a receipt.

Apply Discounts or Insurance

- **Stimulus:** An admin applies a discount or insurance coverage to a patient's bill.
- **Response:** The system recalculates the payable amount and updates the invoice accordingly.

View Billing History

- **Stimulus:** A patient or admin accesses the billing module.
- **Response:** The system shows all past transactions and billing summaries.

4.10.3 Functional Requirements

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.

5. Other Nonfunctional Requirements

5.1 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 complete 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.

5.2 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).

5.3 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.

5.4 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.

5.5 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 a priority-based logic. Emergency patients and ICU cases override normal admission sequences based on physician recommendations.
- Ambulance booking must check availability of both vehicle 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.

6. Other Requirements

This section defines additional requirements not covered in previous sections of the SRS.

6.1 Database Requirements

- The system will use a relational database management system such as **MySQL** or **PostgreSQL**.
- The database will store user information, crop data, market prices, forecasts, and system logs.
- Data integrity will be ensured through primary and foreign key constraints.
- Regular database backups will be scheduled to prevent data loss.
- Passwords and sensitive data will be encrypted using industry-standard hashing techniques (e.g., SHA-256).

6.2 Internationalization Requirements

- The application will initially support only **English**.
- Unicode (UTF-8) encoding will be implemented to support multiple languages in future releases.
- The system interface will be designed with placeholders to accommodate dynamic language switching.

6.3 Legal Requirements

- The system must comply with local and national **data privacy laws**.
- Users must consent to a **privacy policy** before account creation.
- Market forecasts will include disclaimers about the **non-guaranteed accuracy** of predictions.

6.4 Reusability Objectives

- Key components like the **login module**, **forecasting engine**, and **report generator** will be modular for potential reuse.
- Components will follow standard design patterns to promote integration into future agricultural analytics projects or educational systems.

6.5 Performance Requirements

- The system should handle at least **100 concurrent users** without performance degradation.
- Data queries for reports or market prices should respond in **under 3 seconds** on average.

Appendix A: Glossary

Term	Definition
SRS	Software Requirements Specification
UI	User Interface
UX	User Experience
DFD	Data Flow Diagram
UML	Unified Modeling Language
CRUD	Create, Read, Update, Delete
SQL	Structured Query Language
Forecasting	Predicting future market conditions using historical data
Role-Based Access Control	Security model that assigns permissions based on user roles
TBD	To Be Determined

Appendix B: Analysis Models

The following models have been included as part of system analysis:

- **Level 0 DFD**: Displays the main system process and how it interacts with external entities such as users and data sources.
- **Level 1 DFD**: Breaks down the central processes like "Monitor Prices," "Input Forecast Data," and "Generate Reports" into more specific subprocesses.
- **Class Diagram**: Shows key classes such as `User`, `Crop`, `MarketPrice`, and `Forecast`, along with their attributes and relationships.
- **Sequence Diagrams**: Illustrate interactions for core functionalities like user login and generating crop forecasts.
- **Activity Diagrams**: Detail the workflow of major processes such as data input, validation, and report generation.

Appendix C: To Be Determined List

#	TBD Item	Description
1	Forecasting Algorithm	The specific algorithm (e.g., ARIMA, LSTM) to be used for forecasting is not finalized.
2	Data Source APIs	External APIs or databases for real-time market prices are yet to be selected.
3	Hosting Environment	Decision pending on whether the system will be hosted on cloud or local servers.
4	Localization Languages	Languages for future internationalization support have not been determined.