Software Requirements Specification

For Health Management System

Version 2.0

Prepared by: Md. Abdul Mukit(241-35-121)

Date: [07/04/2025]

Table of Contents

- Related Work
- Gap Analysis

4. User Classes and Characteristics

- Patients
- Doctors & Nurses
- Receptionists & Front Desk Staff
- Hospital Administrators
- Pharmacy Staff
- Lab Technicians
- System Administrators

5. Design and Implementation Constraints

- 5.1 User Interface Technology
- 5.2 Implemented Tools and Platform

6. Requirement Specification

- 6.1 Functional Requirements
  - FR 01–18 (e.g., Patient Registration, Appointment Booking, Billing)
- 6.2 Non-Functional Requirements
  - 6.2.1 Performance Requirements
  - 6.2.2 Security Requirements
  - 6.2.3 Reliability Requirements
  - 6.2.4 Usability Requirements
  - 6.2.5 Maintainability Requirements
  - 6.2.6 Compliance Requirements
  - 6.2.7 Portability Requirements
  - 6.2.8 Environmental Requirements
- 6.3 Accessibility Requirements
- 6.4 Operational and Environmental Requirements
- 6.5 Legal Requirements

7. Use Case Diagram & Description

- 7.1 Use Case Diagram
- 7.2 Use Case Descriptions (e.g., Register Patient, View Medical Records, Pay Bills**)**

8. Activity Diagram

9. Requirement Engineering Process

- 9.1 Requirement Elicitation Techniques
- 9.2 Requirement Validation
- 9.3 Change Management

10. Project Plan

- 10.1 Timeline

1. Abstract

The **Healthcare Management System (HMS)** project aims to develop a comprehensive software solution that enhances the management of healthcare services within hospitals and clinics. The system is designed to streamline various administrative tasks such as patient registration, appointment scheduling, treatment tracking, blood bank inventory management, and billing. By integrating these key functionalities into a unified platform, HMS will improve efficiency, reduce human error, and ensure better care delivery. Additionally, it enhances the accessibility and security of sensitive patient data, making it easier for both medical professionals and administrative staff to manage patient information in a fast and secure

manner. Ultimately, the system will lead to improved patient satisfaction, optimized hospital operations, and a higher quality of healthcare service.

2. Introduction

2.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a comprehensive description of the **Healthcare Management System (HMS)**, a software solution developed to streamline and optimize healthcare services within hospitals and clinics. The system is designed to manage key healthcare operations, including patient registration, appointment scheduling, treatment tracking, blood bank inventory management, medical store operations, and billing. This document outlines the functional and non-functional requirements necessary for the development, testing, and deployment of the HMS. The goal is to ensure that the system is efficient, secure, scalable, and capable of meeting the diverse needs of healthcare providers while improving the quality of patient care, enhancing operational efficiency, and ensuring data security.

2.2 Project Scope

The **Healthcare Management System (HMS)** will enable healthcare providers to efficiently manage and streamline a variety of healthcare processes, ensuring the effective delivery of medical services. The system will provide a robust platform capable of handling key functions such as patient registration, appointment scheduling, treatment tracking, blood bank inventory management, medical store management, and billing. It will also support data security, ensuring that patient information is protected and accessible only to authorized personnel.

The system will support both individual healthcare professionals and administrative staff, with features designed to meet the operational needs of hospitals and clinics. It will be adaptable to various healthcare environments, capable of handling diverse medical records, patient details, and treatment histories. Additionally, the HMS will include performance tracking and reporting capabilities, allowing medical staff to monitor patient care progress and ensure optimal treatment outcomes.

The platform will be customizable to accommodate different hospital sizes, medical specialties, and regulatory requirements. It will also integrate with existing systems where necessary and allow for future enhancements, including potential database migration or integration with more advanced healthcare technologies.

Scope Includes:

- **Patient Management:** Secure registration, management, and updating of patient profiles, including personal information, medical history, and treatment details.

- **Appointment Scheduling:** Seamless booking, rescheduling, and cancellation of doctor appointments, with real-time availability and reminders for patients and healthcare providers.

- **Treatment and Prescription Management:** Detailed tracking of patient treatments, diagnoses, prescribed medications, and follow-up care.

- **Blood Bank Inventory Management:** Real-time monitoring and management of blood donations, blood type availability, and inventory levels to ensure proper allocation for patient needs.

- **Medical Store Management:** Efficient management of pharmacy inventory, including stock levels, medicine expiry dates, and reordering.

- **Billing and Invoicing:** Comprehensive billing system to generate invoices for services rendered, track payments, and handle insurance claims processing.

- **Data Security and Privacy:** Implementation of strict data protection measures to ensure the confidentiality and integrity of sensitive patient information in compliance with relevant regulations (e.g., HIPAA).

- **User Role Management:** Different access levels for various users (administrators, doctors, nurses, and patients), ensuring appropriate access to sensitive information.

- **Real-Time Reporting and Analytics:** Generation of performance metrics, reports on patient care, inventory levels, billing summaries, and other relevant healthcare statistics to assist in decision-making.

- **Scalable Deployment:** The system is designed to be scalable for deployment in both small clinics and large hospital networks, ensuring flexibility in adapting to different healthcare environments.

- **Integration with Existing Systems:** Ability to integrate with existing Electronic Health Record (EHR) systems, laboratory databases, or external healthcare applications to ensure seamless data flow across platforms.

2.3 Glossary

HMS (Healthcare Management System): The core software solution being developed to streamline healthcare operations, including patient management, appointment scheduling, treatment tracking, blood bank management, medical store operations, and billing.

EHR (Electronic Health Records): Digital version of a patient’s medical history, which includes data such as medical diagnoses, treatment plans, prescriptions, and previous health conditions.

EMR (Electronic Medical Records): A digital version of a patient's chart that contains the medical and treatment history provided by one healthcare provider.

Patient Management: The process of storing and managing patient details such as personal information, medical history, diagnoses, and treatment progress within the system.

Appointment Scheduling: The functionality that allows patients and healthcare providers to schedule, reschedule, or cancel appointments based on real-time availability.

Prescription Management: The feature in the system for tracking and managing medications prescribed to patients, ensuring timely refills, and monitoring drug interactions.

Blood Bank Inventory Management: A feature that tracks blood donations, available blood types, and inventory levels, ensuring that the necessary supplies are always available for patient needs.

Medical Store Management: The system functionality responsible for managing the inventory of medicines and supplies within the healthcare facility, including tracking expiry dates and stock levels.

Billing System: The module that generates invoices for medical services provided, processes payments, and supports insurance claims and reimbursement tracking.

Data Security: The measures and technologies employed to protect sensitive patient information from unauthorized access or breaches, ensuring compliance with privacy regulations such as HIPAA.

User Role Management: The system functionality that defines and manages different user access levels, such as administrators, doctors, nurses, and patients, to ensure appropriate access to data.

Reporting & Analytics: The functionality for generating reports and data analysis, including patient care outcomes, treatment efficacy, inventory levels, billing summaries, and more, to assist in hospital management.

SDLC (Software Development Life Cycle): The process followed during the design, development, testing, deployment, and maintenance of the Healthcare Management System to ensure high-quality software delivery.

HIPAA (Health Insurance Portability and Accountability Act): U.S. federal law that governs the protection and confidentiality of patient health information, ensuring compliance in systems managing healthcare data.

Cloud-Based Deployment: The ability to deploy the HMS on cloud servers, enabling access from multiple devices, scalability, and secure off-site storage for data.

Role-Based Access Control (RBAC): A security approach used in the system where users are assigned specific roles (e.g., doctor, nurse, admin) with defined permissions to access various system features and data.

2.4 References

IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications – A guideline for documenting the software requirements of a system, ensuring clarity, consistency, and completeness in the software design and development process.

HIPAA Privacy Rule: Health Insurance Portability and Accountability Act – U.S. federal law establishing standards for the protection of health information and ensuring patient privacy and security in health data management systems.

ISO/IEC 27001: Information Security Management Systems – A standard for establishing, implementing, operating, monitoring, reviewing, and improving information security management systems, ensuring the confidentiality and integrity of patient data in healthcare environments.

Research Paper: "Impact of Healthcare Information Systems on Clinical Efficiency" – A study exploring how healthcare management software can improve clinical workflows, patient care efficiency, and reduce errors in medical settings.

National Health Service (NHS) Data Security and Protection Toolkit – Guidelines and best practices on securing health data, ensuring that healthcare management systems comply with necessary security and privacy standards.

HealthIT.gov: U.S. Department of Health and Human Services – A government resource that provides standards and policies regarding the implementation and management of health information technologies, including Electronic Health Records (EHR) and Health Information Exchange (HIE).

Research Paper: "Benefits of Integrated Healthcare Management Systems in Hospitals" – A study analyzing how integrated healthcare management systems streamline operations, improve patient care, and enhance hospital performance

2.5 Overview

This document defines the functionalities and capabilities of the **Healthcare Management System (HMS)**, which aims to streamline and optimize healthcare operations within hospitals and clinics. It includes details about the system's key features, user roles, constraints, and requirements, offering a comprehensive guide to the development, testing, and deployment processes. The HMS is designed to manage critical healthcare activities such as patient registration, appointment scheduling, treatment tracking, blood bank inventory management, medical store operations, and billing.

The system's design focuses on improving the efficiency of healthcare delivery, ensuring the security and privacy of patient data, and providing valuable insights through reporting and analytics. The HMS will be scalable to accommodate the needs of both small healthcare

facilities and large hospital networks. This document serves as a reference for developers, testers, and stakeholders throughout the entire software development lifecycle to ensure that the system meets the required functionality, security standards, and operational expectations.

3. Literature Review {new}

3.1 Related Work

Several healthcare management systems (HMS) and electronic health record (EHR) solutions have been developed to digitize medical workflows. Examples include:

- **Epic Systems**: A widely used EHR platform for large hospitals, focusing on patient records and billing but often criticized for complex interfaces.
- **Cerner**: Specializes in clinical data management but lacks seamless integration with third-party diagnostic tools.
- **OpenMRS**: An open-source HMS for resource-limited settings, though it has limited scalability for large healthcare networks.
- **Practice Fusion**: Cloud-based EHR with strong usability but minimal support for advanced analytics.

While these systems address **specific healthcare functions** (e.g., EMRs, billing), they often:

- Operate in **silos** without unified patient dashboards.
- Lack **real-time interoperability** with labs, pharmacies, or insurance systems.
- Use **outdated UIs**, hindering adoption among non-technical staff.

3.2 Gap Analysis

Existing systems fail to provide:

1. **Holistic Integration**
    - No single platform combines **patient care, inventory, billing, and analytics**.
    - Example: Prescription systems rarely sync with pharmacy stock levels in real time.
2. **User-Centric Design**
    - Clinicians struggle with **clunky interfaces**, leading to workflow inefficiencies.
    - Patients lack **self-service portals** for appointments or test results.
3. **Interoperability**
    - Proprietary formats (e.g., non-HL7/FHIR) hinder data sharing across hospitals.
    - Limited APIs for telemedicine or wearable device integration.
4. **Advanced Analytics**
    - Few systems leverage **AI/ML** for predictive diagnostics or resource allocation.
5. **Scalability for Diverse Settings**
    - Most solutions are **hospital-centric**, neglecting clinics or rural healthcare providers

4. User Classes and Characteristics

1. Patients

Description: Primary users seeking healthcare services (e.g., appointments, prescriptions).
Demographics:

- Age: All age groups (pediatric to geriatric).
- Digital Literacy: Basic understanding of UIs (web/mobile).
  Responsibilities:
- Schedule/cancel appointments.
- View medical history and test results.
- Pay bills and receive reminders.
  Technical Requirements:
- Intuitive interface with minimal training needed.
- Secure access to personal health data (HIPAA/GDPR compliant).

---

2. Doctors & Nurses

Description: Medical professionals providing patient care.
Demographics:

- Qualifications: Licensed physicians, specialists, RNs.
- Workflow: High-paced clinical environments.
  Responsibilities:
- Access/update EMRs (Electronic Medical Records).
- Prescribe medications and treatments.
- Schedule follow-ups and monitor patient progress.
  Technical Requirements:
- Fast, reliable access to patient data.
- Integration with diagnostic tools (e.g., lab systems).

---

3. Receptionists & Front Desk Staff

Description: Handle patient onboarding and administrative tasks.
Demographics:

- Roles: Appointment coordinators, data entry operators.
- Skills: Multitasking, customer service.
  Responsibilities:
- Register patients and verify insurance.
- Manage appointment calendars.
- Handle walk-ins and emergencies.
  Technical Requirements:
- Bulk data entry support (e.g., CSV imports).
- Real-time scheduling conflicts detection.

---

4. Hospital Administrators

Description: Oversee facility operations and compliance.
Demographics:

- Expertise: Healthcare policies, resource management.
- Access: Full system privileges.
  Responsibilities:

- Generate operational/financial reports.
- Manage staff permissions and roles.
- Ensure regulatory compliance (HIPAA, JCI).
  Technical Requirements:
- Customizable dashboards and analytics.
- Audit trails for sensitive actions.

---

5. Pharmacy Staff

Description: Dispense medications and manage inventory.
Demographics:

- Qualifications: Pharmacists, pharmacy technicians.
- Workflow: Inventory-heavy with prescription verification.
  Responsibilities:
- Process e-prescriptions from doctors.
- Track drug stock/expiry dates.
- Handle patient medication queries.
  Technical Requirements:
- Barcode/QR scanning for inventory.
- Alerts for drug interactions/allergies.

---

6. Lab Technicians

Description: Conduct tests and upload diagnostic results.
Demographics:

- Roles: Pathologists, radiologists, lab assistants.
- Tools: Lab equipment interfacing (HL7/FHIR).
  Responsibilities:
- Upload test results (bloodwork, imaging).
- Flag critical findings to doctors.
- Maintain lab equipment logs.
  Technical Requirements:
- Secure DICOM/PDF uploads for reports.
- Integration with LIS (Lab Information Systems).

---

7. System Administrators

Description: Maintain HMS infrastructure and security.
Demographics:

- Skills: IT/network administration, cybersecurity.
- Access: Root-level system control.
  Responsibilities:
- Deploy updates and patches.
- Manage backups and disaster recovery.
- Monitor system performance/security breaches.
  Technical Requirements:
- Role-based access control (RBAC).
- Automated logging of system events.

Each user class has distinct roles and responsibilities within the healthcare system, ensuring smooth operation, patient care, and data security.

5. Design and Implementation Constraints

5.1 User Interface Technology

The **Healthcare Management System (HMS)** will rely on a user-friendly and intuitive interface designed to meet the needs of healthcare professionals, patients, and administrators. The platform will utilize a combination of modern front-end and back-end technologies to ensure seamless interaction, smooth workflows, and efficient management of healthcare tasks.

1. **Frontend Technologies**:

    - **React.js**: A popular JavaScript library for building dynamic and responsive user interfaces. React will be used to create interactive dashboards, patient management interfaces, and appointment scheduling forms that are intuitive and responsive across devices (desktop, tablet, mobile).

    - **Vue.js**: A progressive JavaScript framework for building user interfaces and single-page applications. Vue.js will enable smooth user interactions and transitions, providing an optimized experience for patient records, prescription management, and billing.

    - **HTML5 & CSS3**: Core web technologies for structuring and styling the web interfaces. These will be used in combination with responsive design frameworks like **Bootstrap** to ensure that the system is mobile-friendly and accessible across different screen sizes.

2. **Backend Technologies**:

    - **Node.js**: A JavaScript runtime environment used for building scalable and fast backend services. It will handle API requests, database operations, and real-time data processing, especially for appointment scheduling, medical records management, and billing systems.

    - **Express.js**: A minimal web framework for Node.js that will be used to build robust APIs for communicating between the frontend and backend systems. Express will enable seamless communication between the user interface and the underlying data storage.

3. **Database**:

- **MongoDB**: A NoSQL database will be used to store patient data, appointment schedules, treatment records, and billing information. MongoDB offers flexibility for unstructured data and scalability, making it a great choice for large healthcare environments.

- **PostgreSQL**: An alternative relational database for structured data storage such as billing information, staff records, and patient demographics. PostgreSQL ensures integrity and consistency of data across the system.

4. **Admin Interface**:

   - **Web Interface**: An intuitive, browser-based admin interface will be provided to hospital administrators and staff for managing patient data, appointments, treatment plans, and inventory. Technologies like **Vue.js** or **React** will power this interface, allowing admins to configure the system, track performance, and generate reports.

   - **Admin Dashboards**: Admins will have access to customizable dashboards using **Chart.js** or **D3.js** for visualizing key metrics such as patient volumes, treatment outcomes, and inventory levels.

5. **Security & Authentication**:

   - **OAuth 2.0** / **JWT (JSON Web Tokens)**: These technologies will be used to implement secure authentication and role-based access control (RBAC), ensuring that only authorized users can access sensitive healthcare data.

   - **SSL/TLS Encryption**: All communication between users and the system will be encrypted to ensure privacy and security of patient information.

6. **Mobile App Interface**:

   - **React Native**: A cross-platform mobile app framework for building native apps for both iOS and Android. This will be used to create mobile applications for patients to manage appointments, track treatments, and access their medical records on the go.

These technologies will work together to provide an efficient, scalable, and secure user interface for the healthcare management system, ensuring that it meets the needs of all users while delivering an optimal experience across platforms.

5.2 Implemented Tools and Platform

1. **Backend:**

    - **Node.js and Express.js**: For backend development, **Node.js** will be used as the JavaScript runtime environment, ensuring fast and scalable server-side applications. **Express.js**, a web application framework for Node.js, will handle API requests, manage routes, and support data processing tasks like appointment scheduling, billing, and patient management. This combination will enable smooth and efficient communication between the front-end interface and the database systems.

    - **MongoDB / PostgreSQL**: **MongoDB**, a NoSQL database, will be used to manage unstructured patient data, treatment records, and appointment schedules, allowing flexibility and scalability. Alternatively, **PostgreSQL**, a relational database, will store structured data like billing information, staff records, and patient demographics. Both databases ensure reliable data management and easy retrieval of patient and healthcare-related information.

    - **Apache/Nginx**: These web server platforms will handle HTTP requests, API integrations, and serve the front-end user interface. **Apache** or **NGINX** will be used to manage traffic efficiently, ensuring optimal performance for healthcare professionals accessing patient data and administrators managing hospital operations.

2. **Frontend:**

    - **React.js**: The front-end interface will be developed using **React.js**, a JavaScript library for building user interfaces. React will be utilized to design dynamic and responsive dashboards, forms for patient registration, appointment scheduling, and treatment management. It ensures an interactive, fast, and seamless experience for users such as doctors, nurses, and administrative staff.

    - **Vue.js**: As an alternative, **Vue.js** will be used to develop real-time interfaces, helping users manage appointments, view patient records, and process billing. Vue's flexibility in creating reactive components will ensure efficient and smooth interactions in the system for both healthcare staff and patients.

These tools and platforms, combined with modern front-end and back-end technologies, will deliver a scalable, secure, and responsive healthcare management system, enabling efficient operation and management within healthcare facilities.

6. Requirement Specification

6.1 Functional Requirements

FR 01	**Patient Registration**
Description	The system shall allow receptionists to register new patients by capturing details (ID, name, age, gender, contact) and storing them in a file-based database. Existing patients can be updated or deleted.
Stakeholder	Receptionists, Administrators

FR 02	**Patient Search**
Description	The system shall enable users to search for patients by ID or name and display their full records (demographics, appointment history).
Stakeholder	Receptionists, Doctors

FR 03	**Patient Record Update**
Description	The system shall allow authorized users to modify patient details (e.g., contact number, address) and save changes to the database.
Stakeholder	Receptionists

FR 04	**Appointment Booking**
Description	The system shall allow receptionists to book appointments by selecting a doctor, timeslot, and patient ID. A confirmation with a 30% advance payment will be generated.
Stakeholder	Receptionists, Patients

FR 05	**Appointment Cancellation**
Description	The system shall permit cancellation of appointments by patient ID and update the appointment records accordingly.

Stakeholder	Receptionists

FR 06	**Emergency Appointment Handling**
Description	The system shall prioritize emergency appointments by allowing immediate booking without advance payment.
Stakeholder	Doctors, Receptionists

FR 07	**Treatment Record Creation**
Description	The system shall enable doctors/lab technicians to add treatment records (e.g., X-ray, MRI) with test results and link them to patient IDs.
Stakeholder	Doctors, Lab Technicians

FR 08	**Treatment History Tracking**
Description	The system shall maintain a doubly linked list of treatments per patient for historical tracking and generate invoices.
Stakeholder	Doctors, Accountants

FR 09	**Blood Donor Registration**
Description	The system shall register blood donors (name, blood type, contact) and track donation dates/quantities.
Stakeholder	Lab Technicians

FR 10	**Blood Stock Management**
Description	The system shall monitor blood stock levels (by type) and alert staff when stock is low or expired.
Stakeholder	Lab Technicians

FR 11	**Medicine Inventory Management**

Description	The system shall allow pharmacists to add/update medicines (ID, name, price, expiry date) and flag expired/low-stock items
Stakeholder	Pharmacists

FR 12	**Medicine Search**
Description	The system shall support searching medicines by name, category, or ID.
Stakeholder	Pharmacists, Doctors

FR 13	**Bill Generation**
Description	The system shall generate itemized bills (consultation, medicines, tests) with tax/discount calculations.
Stakeholder	Accountants

FR 14	**Payment Processing**
Description	The system shall record payments (cash/card/insurance) and update due amounts in real-time.
Stakeholder	Accountants

FR 15	**Revenue Reporting**
Description	The system shall generate monthly revenue reports by aggregating paid bills.
Stakeholder	Administrators

FR 16	**Data Persistence**
Description	The system shall save all records (patients, appointments, treatments) to files (patients.txt, appointments.dat) on exit and reload them on startup.
Stakeholder	All Users

FR 17	**User Access Control**
Description	The system shall restrict features based on roles (e.g., only accountants can generate bills). *[Note: To be implemented in future versions.]*
Stakeholder	Administrators

FR 18	**Input Validation**
Description	The system shall validate inputs (e.g., date formats, numeric IDs) and reject invalid entries with error messages.
Stakeholder	All Users

6.2 non - functional requirements

6.2.1. Performance Requirements

NFR-P01: Response Time

- The system shall load patient records in <2 seconds for 95% of queries under normal load (1,000 concurrent users).

NFR-P02: Throughput

- The system shall process 50+ appointment bookings per minute during peak hours without degradation.

NFR-P03: Scalability

- The system shall support horizontal scaling to handle up to 500 concurrent users with linear performance.

6.2.2. Security Requirements

NFR-S01: Data Protection

- All patient data shall be encrypted at rest (AES-256) and in transit (TLS 1.3+).

NFR-S02: Access Control

- **Implement role-based access control (RBAC) with:**
  - 5 predefined roles (Admin, Doctor, Nurse, Receptionist, Patient)
  - Multi-factor authentication for clinical staff

NFR-S03: Audit Trail

- **Log all sensitive operations (e.g., record modifications) with:**
  - Timestamp
  - User ID
  - Action type
  - Retention period: 7 years

6.2.3. Reliability Requirements

NFR-R01: Availability

- System uptime shall be ≥99.5% during operational hours (6 AM - 10 PM daily).

NFR-R02: Fault Tolerance

- The system shall continue core operations (appointments, emergency care) during:
  - Database failures (switch to read-only mode)
  - Network outages (local caching for 4 hours)

NFR-R03: Backup

- **Automated daily backups with:**
  - 15-minute RPO (Recovery Point Objective)
  - 1-hour RTO (Recovery Time Objective)

6.2.4. Usability Requirements

NFR-U01: Accessibility

- **Comply with WCAG 2.1 AA standards:**
  - Screen reader support
  - Keyboard navigation
  - Color contrast ≥4.5:1

NFR-U02: Learnability

- New users shall complete basic workflows (appointment booking) within 3 minutes without training.

NFR-U03: Localization

- Support 3 languages (English, Spanish, French) for patient-facing interfaces.

6.2.5. Maintainability Requirements

NFR-M01: Modularity

- **System components shall be independently updatable with:**
    - Clear API contracts
    - Versioned endpoints

NFR-M02: Documentation

- **Provide:**
    - API documentation (OpenAPI 3.0)
    - Deployment runbooks
    - Troubleshooting guides

6.2.6. Compliance Requirements

NFR-C01: Regulatory

- **Adhere to:**
    - HIPAA (US)
    - GDPR (EU)
    - HL7/FHIR standards

NFR-C02: Certifications

- Obtain SOC 2 Type II certification within 12 months of launch.

6.2.7. Portability Requirements

NFR-PT01: Cross-Platform Support

- **Web interface compatible with:**
    - Chrome, Firefox, Safari (latest 2 versions)
    - Mobile browsers (iOS/Android)

NFR-PT02: Data Export

- **Allow export of patient records in:**
    - PDF (human-readable)
    - JSON/XML (machine-readable)

6.2.8. Environmental Requirements

NFR-E01: Deployment

- **Support deployment on:**
    - On-premise servers
    - AWS/GCP cloud
    - Hybrid configuration**s**

NFR-E02: Resource Usage

- Client-side RAM usage shall not exceed 500MB for core workflows

6.3 Accessibility Requirements

User Login/Logout

- **AR-1: Log in as a Patient**
- **AR-2: Log in as a Doctor**
- **AR-3: Log in as a Nurse**
- **AR-4: Log in as a Receptionist**
- **AR-5: Log in as a Hospital Administrator**
- **AR-6: Log in as a System Administrator**
- **AR-7: Log out from any role**

Central Authentication Mechanism

- The system shall provide a single sign-on (SSO) portal for all users (Patients, Medical Staff, Administrators).
- Authentication shall integrate with LDAP/Active Directory for hospital staff.

Password Security

- **Encryption:** All passwords shall be encrypted using bcrypt (or equivalent) with a minimum cost factor of 12**.**
- **Complexity Requirements:**
  - Minimum 8 characters
  - At least 1 uppercase, 1 lowercase, 1 number, and 1 special character
  - Password expiry every 90 days for staff accounts

Multi-Factor Authentication (MFA)

- **AR-8:** MFA shall be mandatory for:
  - Doctors accessing patient EMRs
  - Administrators performing system configurations
  - Pharmacy staff processing prescriptions

Session Management

- **AR-9:** Automatic session timeout after 15 minutes of inactivity.
- **AR-10:** Concurrent sessions shall be limited to 1 per user (except Patients, who may have 2).

Accessibility Compliance

- **AR-11:** The login interface shall comply with WCAG 2.1 AA, including:
  - Screen reader compatibility
  - Keyboard-only navigation
  - Color-contrast ratio ≥4.5:1

Self-Service Password Reset

- **AR-12:** Patients/staff shall reset passwords via:
    - Email OTP (One-Time Password)
    - Security questions (configurable)

6.4 Operational and Environmental Requirements

6.4.1 Expected Physical Requirements

- The HMS shall operate effectively within:
    - **Temperature range**: 10°C to 35°C
    - **Humidity levels**: 20% to 80% (non-condensing)
- Hardware must be placed in:
    - **Stable, vibration-free environments** (e.g., server rooms with shock-absorbing racks)
    - **Adequately ventilated spaces** (minimum 2m clearance around critical servers)

6.4.2 Interfacing with Adjacent Systems

- The HMS shall integrate seamlessly with:
    - **Hospital Information Systems (HIS)** via HL7/FHIR APIs
    - **Lab Information Systems (LIS)** for real-time test results
    - **Insurance claim portals** (e.g., CMS-1500 forms)
    - **Pharmacy databases** for e-prescription validation
- Data synchronization shall occur **every 5 minutes** with <1% error tolerance.

6.4.3 Release Requirements

- **Patch Management**:
    - Critical security updates deployed within **24 hours** of release
    - Non-critical updates rolled out **monthly** during maintenance windows (10 PM–2 AM local time)
- **Version Control**:
    - Backward compatibility maintained for **2 prior major versions**
    - Release notes documenting **database schema changes.**

6.5 Legal Requirements

6.5.1 Data Privacy & Protection

- **LR-01**: Comply with **HIPAA (US)** and **GDPR (EU)** for:
    - Patient data encryption (at rest/transit)
    - Right to access/erase personal data
    - Breach notification within **72 hours**
- **LR-02**: Implement **data minimization** – Collect only essential patient information.

6.5.2 Audit & Accountability

- **LR-03**: Maintain immutable logs for:
    - All EMR accesses (who, when, why)

    - Prescription changes
    - Billing modifications
    - **Retention period**: 7 years (10 years for pediatric records)

6.5.3 Consent Management

- **LR-04**: Obtain **digital consent** for:
    - Treatment plans
    - Data sharing with 3rd parties (e.g., labs, insurers)
    - Telemedicine consultations
- **LR-05**: Allow patients to **withdraw consent** via self-service portal.

6.5.4 Medical Compliance

- **LR-06**: Adhere to:
    - **HL7/FHIR** standards for interoperability
    - **FDA 21 CFR Part 11** (if handling electronic signatures)
    - Local **medical licensing laws** (e.g., physician credential verification)

6.5.5 Liability & Disclaimers

- **LR-07**: Display **clear disclaimers** for:
    - AI-assisted diagnostics (e.g., "For clinician review only")
    - Drug interaction warnings
- **LR-08**: System shall **not override** clinical decisions – Final authority rests with doctors.

6.5.6 Insurance & Billing

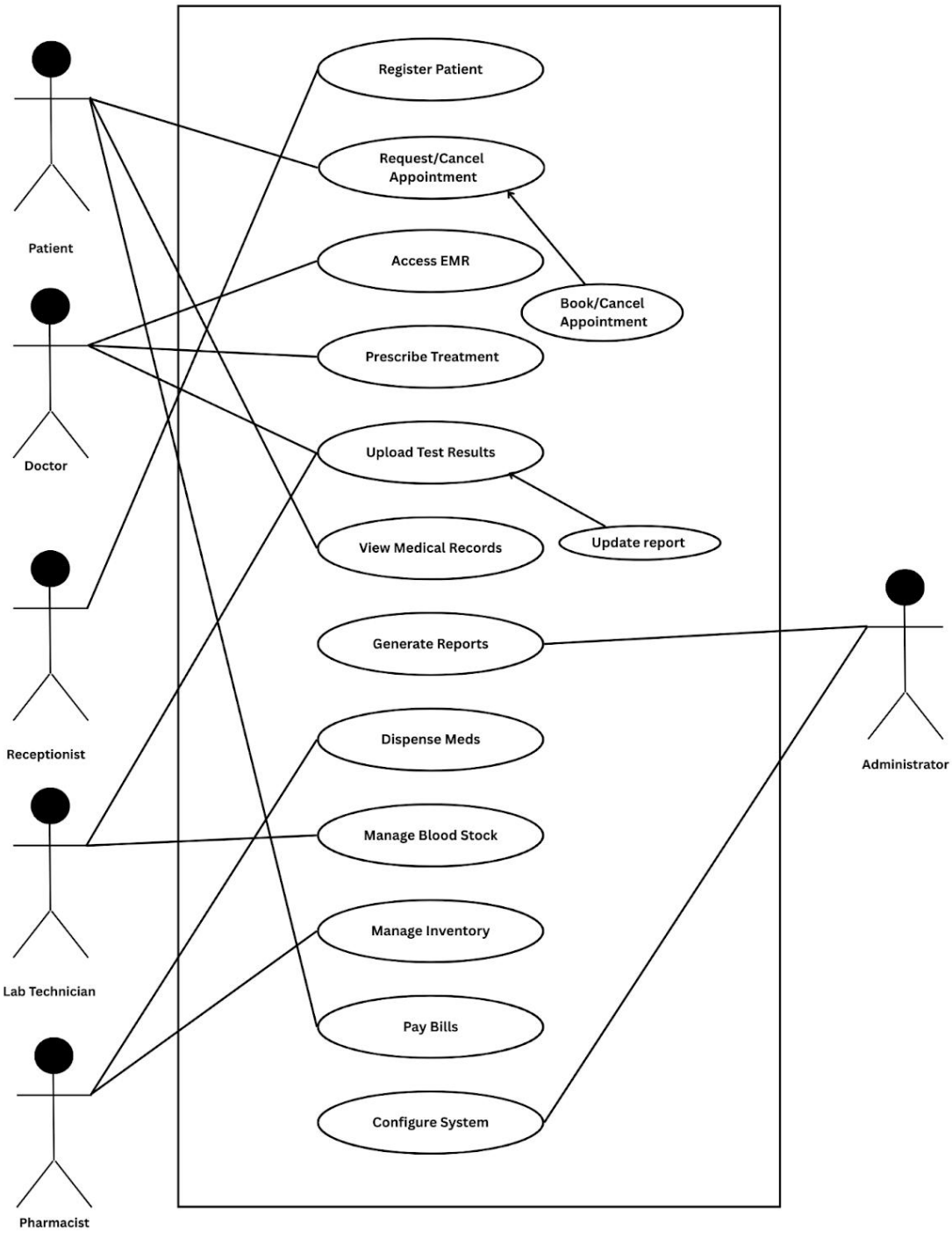
- **LR-09**: Automate **CMS-1500/UB-04** form generation for US insurance claims.
- **LR-10**: Validate **ICD-10/CPT codes** to prevent fraudulent billing.

6.5.7 Jurisdictional Adaptability

- **LR-11**: Support configurable rules for:
    - Minor consent age (varies by country/state)
    - Mandatory reporting (e.g., infectious diseases)

7. Use Case Diagram & Description

7.1 Use case Diagram



7.2 Use Case Description

<table>
<tr><td>Use case</td><td>Register Patient</td></tr>
<tr><td>Goal</td><td>Allow receptionists to register new patients in the system with their demographic and medical details.</td></tr>
<tr><td>Pre-condition</td><td>• The receptionist must have valid login credentials.
• The system must be operational and connected to the database.</td></tr>
<tr><td>Success End Condition</td><td>• The patient’s details are successfully stored in the database.
• A unique Patient ID is generated and displayed.</td></tr>
<tr><td>Failed End Condition</td><td>• The system rejects the registration due to invalid/missing data or database errors.
• The patient record is not created.</td></tr>
<tr><td>Primary Actor</td><td>• Receptionist</td></tr>
<tr><td>Secondary Actor</td><td>• System (validates and stores data)
• Database (stores patient records)</td></tr>
<tr><td>Trigger</td><td>• A new patient arrives at the healthcare facility and needs to be registered.</td></tr>
<tr><td>Main Flow</td><td>
<table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>The receptionist navigates to the "Register Patient" page</td></tr>
<tr><td>2</td><td>The system displays a form with fields: Name, Age, Gender, Contact, Address, Medical History (optional).</td></tr>
<tr><td>3</td><td>The receptionist enters the required details.</td></tr>
<tr><td>4</td><td>The system validates the inputs (e.g., age > 0, valid phone number format).</td></tr>
<tr><td>5</td><td>If valid, the system:
• Generates a unique Patient ID.
• Stores the record in the Patient Database (PostgreSQL/MongoDB).
• Displays a confirmation message: "Patient registered successfully! ID: [PID-1001]".</td></tr>
<tr><td>6</td><td>The receptionist provides the Patient ID to the patient.</td></tr>
</table>
</td></tr>
</table>

Alternative flow	Invalid Data Entry
	• Step 4: If validation fails (e.g., blank name, invalid phone number):
	  ◦ The system highlights the error (e.g., "Contact number must be 10 digits").
	  ◦ Allows the receptionist to correct and resubmit.

Quality Requirement	1. Performance:
	  ◦ Registration must complete within 3 seconds (95% of requests).
	2. Security:
	  ◦ Patient data must be encrypted (AES-256) in storage.
	  ◦ No PII (Personal Identifiable Information) displayed in error logs.
	3. Usability:
	  ◦ Form must auto-focus fields and support tab navigation.

Use case	**View Medical Records**
Goal	Allow authorized users (patients, doctors) to access patient medical records securely.
Pre-condition	• User is logged in with valid credentials. • Patient records exist in the database.
Success End Condition	System displays complete medical records (EMR) with test results, prescriptions, and treatment history.
Failed End Condition	Access denied due to insufficient permissions or missing records.
Primary Actor	• Patient (views own records) • Doctor (views patient records)
Trigger	User selects "View Medical Records" from dashboard.
Main Flow	<table><tr><td>Step</td><td>Action</td></tr><tr><td>1</td><td>User clicks "Medical Records" tab</td></tr><tr><td>2</td><td>System verifies user role and permissions</td></tr></table>

<table>
<tr><td></td><td>
<table>
<tr><td>3</td><td>For patients: System displays their own EMR.</td></tr>
<tr><td>4</td><td>For doctors: System prompts for Patient ID → displays EMR after validation.</td></tr>
<tr><td>5</td><td>Records show: Diagnoses, prescriptions, lab results, immunization history.</td></tr>
</table>
</td></tr>
<tr><td>Alternative flow</td><td>• A1: Patient requests record correction → System logs request for admin review.
• A2: Doctor accesses restricted notes → Additional MFA required.</td></tr>
<tr><td>Quality Requirement</td><td>• Load records in <2 seconds.
• HIPAA-compliant audit logging.</td></tr>
</table>

<table>
<tr><td>Use case</td><td>Request/Cancel Appointment</td></tr>
<tr><td>Goal</td><td>Enable patients to book or cancel appointments with healthcare providers.</td></tr>
<tr><td>Pre-condition</td><td>• Patient is registered in the system.
• Doctor schedules are available.</td></tr>
<tr><td>Success End Condition</td><td>Appointment is confirmed/canceled with notification to both parties</td></tr>
<tr><td>Failed End Condition</td><td>Slot unavailable or cancellation window expired</td></tr>
<tr><td>Primary Actor</td><td>Patient</td></tr>
<tr><td>Trigger</td><td>Patient selects "Book Appointment" or "Cancel Appointment".</td></tr>
<tr><td>Main Flow {Booking}</td><td>
<table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>Patient selects:
○ Doctor/specialty → Date/time → Reason for visit.</td></tr>
<tr><td>2</td><td>System checks real-time availability.</td></tr>
<tr><td>3</td><td>Confirms booking → Sends SMS/email confirmation</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Main Flow {Cancellation}</td><td>
<table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>Patient selects appointment → Clicks "Cancel".</td></tr>
<tr><td>2</td><td>System verifies cancellation policy (e.g., >24 hours notice).</td></tr>
<tr><td>3</td><td>Removes slot from schedule → Notifies provider.</td></tr>
</table>
</td></tr>
<tr><td>Alternative flow</td><td>• A1: Emergency booking → Overrides availability checks.
• A2: No-show → System auto-flags patient account after 3 incidents</td></tr>
<tr><td>Quality Requirement</td><td>• Support 50+ concurrent bookings during peak hours.
• Sync with Google Calendar/Outlook</td></tr>
</table>

<table>
<tr><td>Use case</td><td>Pay Bills</td></tr>
<tr><td>Goal</td><td>Allow patients to view and pay medical bills online.</td></tr>
<tr><td>Pre-condition</td><td>• Bill is generated and linked to patient ID.
• Payment gateway is operational.</td></tr>
<tr><td>Success End Condition</td><td>Payment processed → Receipt generated → EMR updated.</td></tr>
<tr><td>Failed End Condition</td><td>Payment declined or transaction timeout.</td></tr>
<tr><td>Primary Actor</td><td>Patient</td></tr>
<tr><td>Trigger</td><td>Patient selects "Pay Bills" from portal.</td></tr>
<tr><td>Main Flow</td><td>
<table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>System displays outstanding bills (consultation, tests, medications).</td></tr>
<tr><td>2</td><td>Patient selects payment method (credit card, insurance, PayPal).</td></tr>
<tr><td>3</td><td>System processes payment → Updates ledger → Sends e-receipt.</td></tr>
</table>
</td></tr>
</table>

Alternative flow	• A1: Partial payment → System calculates remaining balance. • A2: Insurance claim → Auto-files CMS-1500 form.
Quality Requirement	• PCI-DSS compliance for card payments. • Auto-apply discounts for early payments.

Use case	**Access EMR (Electronic Medical Record)**
Goal	Allow authorized medical staff to retrieve patient medical records.
Pre-condition	User is authenticated and has proper permissions
Primary Actor	Doctor
Secondary Actor	System
Trigger	Doctor selects "View Patient Record".
Main Flow	<table><tr><th>Step</th><th>Action</th></tr><tr><td>1</td><td>Doctor enters Patient ID or searches by name.</td></tr><tr><td>2</td><td>System verifies access permissions</td></tr><tr><td>3</td><td>System retrieves and displays complete EMR (diagnoses, treatments, allergies, test results)</td></tr><tr><td>4</td><td>Doctor reviews record and takes necessary action.</td></tr></table>
Alternative flow	If access denied, system logs attempt and notifies admin
Quality Requirement	• Load records in <1.5 seconds. • Audit all access attempts

Use case	**Prescribe Treatment**
Goal	Enable doctors to create and submit treatment plans.
Primary Actor	Doctor
Trigger	Doctor selects "New Prescription"

<table>
<tr><td>Main Flow</td><td><table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>System displays patient's current medications and allergies</td></tr>
<tr><td>2</td><td>1. Doctor enters:
○ Medication name, dosage, frequency
○ Duration of treatment
○ Special instructions</td></tr>
<tr><td>3</td><td>System checks for drug interactions.</td></tr>
<tr><td>4</td><td>Prescription is saved to EMR and sent to pharmacy.</td></tr>
</table></td></tr>
<tr><td>Alternative flow</td><td>If interaction found, system alerts doctor before submission</td></tr>
</table>

<table>
<tr><td>Use case</td><td>Update Test Results</td></tr>
<tr><td>Goal</td><td>Allow lab technicians to input diagnostic results</td></tr>
<tr><td>Primary Actor</td><td>Lab Technician</td></tr>
<tr><td>Trigger</td><td>Test results are ready.</td></tr>
<tr><td>Main Flow</td><td><table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>Technician selects patient and test type</td></tr>
<tr><td>2</td><td>System displays required fields for results</td></tr>
<tr><td>3</td><td>Technician enters values and uploads supporting documents.</td></tr>
<tr><td>4</td><td>System updates EMR and notifies ordering physician</td></tr>
</table></td></tr>
<tr><td>Quality Requirement</td><td>• Support various file formats (PDF, DICOM).
• Flag abnormal results for immediate attention.</td></tr>
</table>

Use case	**Book/Cancel Appointment**
Goal	Manage patient appointments.

<table>
<tr><td>Primary Actor</td><td>Receptionist/Patient</td></tr>
<tr><td>Trigger</td><td>Patient requests appointment.</td></tr>
<tr><td>Main Flow {Booking}</td><td>
<table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>System displays available slots based on:
Doctor's schedule
Urgency level</td></tr>
<tr><td>2</td><td>Receptionist/patient selects time and confirms</td></tr>
<tr><td>3</td><td>System sends confirmation to both parties.</td></tr>
</table>
</td></tr>
<tr><td>Main Flow {cancellation}</td><td>
<table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>User selects appointment to cancel.</td></tr>
<tr><td>2</td><td>System verifies cancellation policy</td></tr>
<tr><td>3</td><td>Slot is released and parties are notified</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Use case</td><td>Upload Test Results</td></tr>
<tr><td>Goal</td><td>Digitally store diagnostic reports.</td></tr>
<tr><td>Primary Actor</td><td>Lab Technician</td></tr>
<tr><td>Trigger</td><td>Test completion.</td></tr>
<tr><td>Main Flow</td><td>
<table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>1. System prompts for:
Patient ID
Test type
Result files</td></tr>
<tr><td>2</td><td>Technician uploads data.</td></tr>
<tr><td>3</td><td>System:</td></tr>
</table>
</td></tr>
</table>

	<table><tr><td></td><td> - Parses structured data (e.g., CSV) - Stores documents - Updates EMR </td></tr></table>
Quality Requirement	Support batch uploads for high-volume labs.

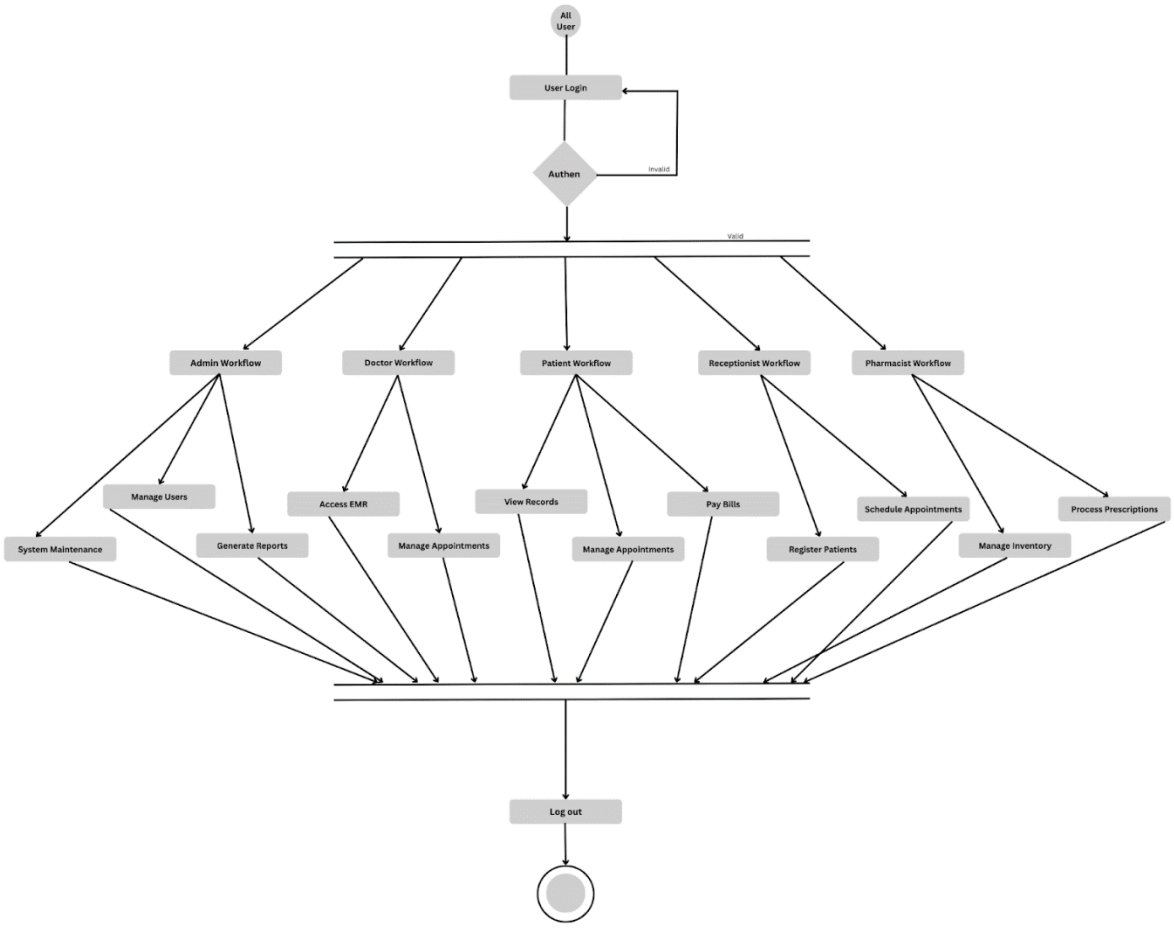
Use case	**Manage Blood Stock**
Goal	Track blood bank inventory.
Primary Actor	Lab Technician
Trigger	New donation or transfusion occurs.
Main Flow	<table><tr><td>Step</td><td>Action</td></tr><tr><td>1</td><td>System displays current stock levels by blood type</td></tr><tr><td>2</td><td>Technician records: - New donations (donor info, blood type, quantity) - Transfusions (patient ID, amount used) </td></tr><tr><td>3</td><td>System updates inventory and flags low stock</td></tr></table>
Alternative flow	If critical shortage, system alerts multiple staff members

Use case	**Manage Inventory**
Goal	Maintain pharmacy stock.
Primary Actor	Pharmacist
Trigger	New shipment arrives or medication is dispensed.
Main Flow	<table><tr><td>Step</td><td>Action</td></tr></table>

	<table><tr><td>1</td><td>System shows current medication quantities.</td></tr><tr><td>2</td><td>Pharmacist: - Adds new stock (scans barcode or manual entry) - Records dispensed medications </td></tr><tr><td>3</td><td>System: - Updates counts - Flags expiring medications </td></tr></table>

Use case	**Dispense Meds**
Goal	Process patient prescriptions.
Primary Actor	Pharmacist
Trigger	Prescription received.
Main Flow	<table><tr><td>Step</td><td>Action</td></tr><tr><td>1</td><td>System displays prescription details and checks stock.</td></tr><tr><td>2</td><td>Pharmacist: - Verifies patient identity - Prepares medication - Records dispensing in system </td></tr><tr><td>3</td><td>System: - Deducts from inventory - Updates patient's medication history </td></tr></table>
Alternative flow	If medication unavailable, system suggests alternatives.

8 Activity Diagram: (All User)



9. Requirement Engineering Process

9.1 Requirement Elicitation Techniques

Requirements were gathered through:

- **Interviews with:**
  - Doctors (to capture clinical workflows and EMR needs).
  - Hospital Administrators (for billing, compliance, and reporting).
  - Patients (to understand portal usability expectations).
- **Workshops to prototype:**
  - Appointment scheduling interfaces.
  - EMR dashboards for different specialties (e.g., cardiology vs. pediatrics).
- **Surveys distributed to:**
  - Nurses/Pharmacists (inventory management pain points).
  - IT Staff (infrastructure and security constraints).

9.2 Requirement Validation

Validation ensures the HMS meets healthcare standards and user needs:

1. **Peer Reviews**
    - Stakeholders: Doctors, nurses, admins, and developers review requirements for:
        - Clinical accuracy (e.g., drug interaction checks in prescriptions).
        - Regulatory compliance (HIPAA/GDPR alignment).
    - Output: Refined FRs/NFRs (e.g., added "flag abnormal lab results" as FR8).
2. **Prototype Testing**
    - Early prototypes tested on:
        - Receptionists: Appointment booking flow (validated FR4).
        - Doctors: EMR access speed (<2 sec load time, per NFR-P01).
    - Tools: Figma mockups, React.js clickable demos.
3. **Simulation Testing**
    - Scenarios validated:
        - Peak load: 200+ concurrent users booking appointments (validated NFR-P02).
        - Emergency mode: Override for urgent appointments (FR6).
4. **Stakeholder Feedback Loops**
    - Monthly check-ins with:
        - Clinical staff: Adjust UI for faster charting (e.g., voice-to-text).
        - Patients: Simplify bill payment portal (added "one-click pay" option).

9.3 Change Management

Controlled process for adapting HMS requirements:

1. **Change Identification**
    - Sources: User feedback (e.g., nurses request barcode scanning for meds), new regulations (e.g., ICD-11 codes).
    - Documentation: Jira tickets with tags (e.g., #compliance, #usability).
2. **Impact Assessment**
    - Evaluated for:
        - System performance (e.g., adding AI diagnostics may increase server load).
        - Training needs (e.g., new pharmacy module requires staff training).
3. **Approval Process**
    - Change Control Board (CCB):
        - Members: Chief Medical Officer, IT Lead, Legal Advisor.
        - Criteria: Alignment with project scope/budget (e.g., rejected blockchain audit logs due to cost).
4. **Implementation & Testing**
    - Staging environment: Test changes with synthetic patient data.
    - UAT (User Acceptance Testing): Nurses/doctors validate new features.
5. **Deployment**
    - Phased rollouts:

        - Pilot deployment in outpatient clinic before hospital-wide.
    - Rollback plan: Revert to last stable version if critical bugs found.
6. **Monitoring & Feedback**
    - Post-deployment metrics:
        - API response times (ensure NFR-P07 compliance).
        - User error rates (e.g., failed login attempts).
7. **Documentation**
    - Updated artifacts:
        - SRS document (revised FRs/NFRs).
        - Training manuals (new workflows for staff).

10. Project Plan for Healthcare Management System (HMS)

10.1 Timeline

Phase	Duration	Key Activities
Requirement Analysis	14 days	Stakeholder interviews, SRS documentation
System Design	16 days	Architecture planning, UI/UX prototyping
Development	20 days	Core modules (EMR, appointments, billing
Testing	14 days	Unit, integration, UAT
Deployment	6 days	Pilot launch (1 clinic), staff training
Maintenance	Ongoing	Bug fixes, updates

10.2 Deliverables

1. **Fully Functional HMS with:**
    - Electronic Medical Records (EMR)
    - Appointment scheduling (incl. emergency slots)
    - Billing & insurance claim processing
    - Pharmacy/lab integration
2. **Technical Documentation:**
    - System architecture diagrams
    - API specifications (HL7/FHIR)
3. **Testing Reports:**
    - Performance benchmarks (e.g., 200 concurrent users)
    - Security audit results (HIPAA compliance)

10.3 Resources

Category	Resources
Software	React.js, Node.js, MongoDB, PostgreSQL, Docker
Team	Developers (4), QA Engineers (2), UI/UX Designer (1), Project Manager
Hardware	Cloud servers (AWS/GCP), HIPAA-compliant backup storage

11 Budget

11.1 Cost Estimates

Item	Cost (USD)
Development (1 months)	$120,000
Cloud Infrastructure	$15,000/year
Training & Deployment	$10,000
Total	**$145,000**

11.2 Funding Sources

- Hospital IT budget (70%)
- Government healthcare grants (30%)

---

12. Evaluation

12.1 Assessment Criteria

1. **Functionality:**
    - Handles 500+ patient records without errors.
    - Processes 50+ appointments/minute during peak hours.
2. **Usability:**
    - 90% of staff rate system as "intuitive" in surveys.
    - Average task completion time <2 minutes (e.g., booking appointments).
3. **Security:**
    - Zero critical vulnerabilities in penetration tests.
    - 100% encryption of PHI (Protected Health Information).

12.2 Evaluation Methods

- User Surveys: Nurses/doctors rate ease of use (Likert scale).
- Load Testing: Simulate 1,000 concurrent users.
- Security Audits: Third-party HIPAA compliance checks.

---

13. Conclusion

This HMS will transform healthcare delivery by:

✔ Digitizing workflows (reducing paper-based errors).
✔ Enhancing patient care through real-time EMR access.
✔ Ensuring compliance with HIPAA/GDPR regulations.

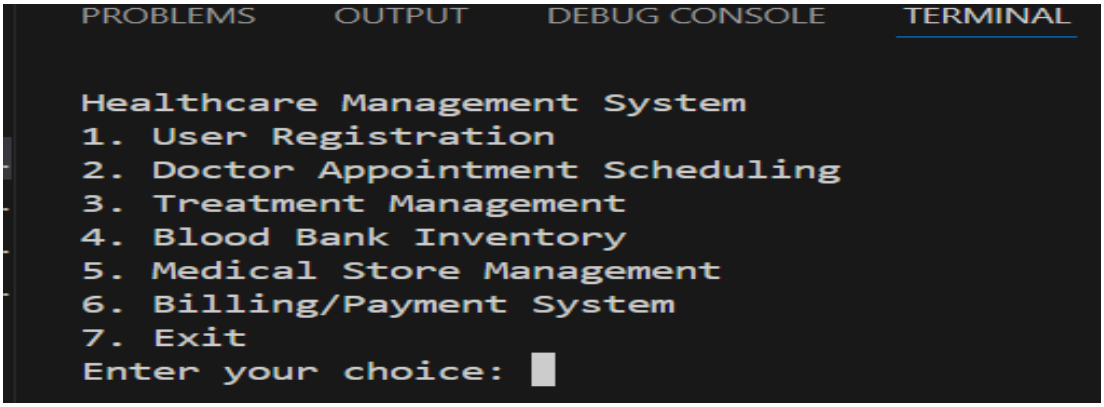
Future Work

1. AI Integration: Predictive analytics for patient readmissions.
2. IoT Compatibility: Wearable device data syncing.
3. Telemedicine: Video consultation module.

14 User Manual

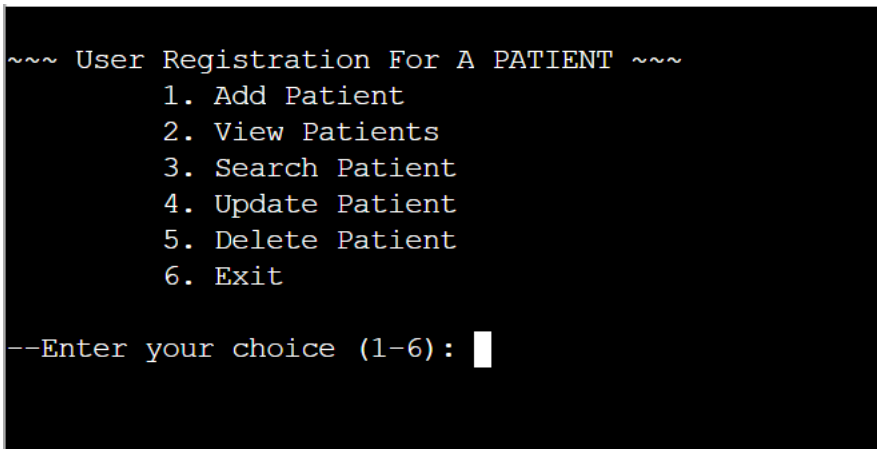
- ***Start:***

  This is the main menu of a Healthcare Management System. It provides options to register a user, Doctor Appointment Scheduling, treatment management, Blood Bank Inventory, Medical Store Management, Billing/payment System, or exit the system. Users select a choice by entering the corresponding number.



1. User Registration:

1.1 Start: This is the main menu of a **Patient Registration System**. It provides options to **add a patient**, **view registered patients**, **search a patient by ID**, **update patient details**, **delete a patient**, or **exit** the system. The user selects a choice by entering the corresponding number.



1.2 Add Patient: This option allows a user to add a new patient into the system.
The system asks for the following information from the user**:**

- Patient ID (e.g., 101)
- Patient Name (e.g., Eren Yeager)
- Patient Age (e.g., 25)
- Patient Gender (e.g., Male/Female/Other)
- Patient Contact Number (e.g., 01712345678)

Once the details are provided, the system saves the information in a text file named **patients.txt**, and shows the message

“Patient registered successfully!”

```
--Enter your choice (1-6): 1

 [ ADD NEW PATIENT ]
--Enter Patient ID (example:101): 10
--Enter Patient Name (example: Eren Yeager): afnan
--Enter Patient Age (example: 25): 21
--Enter Patient Gender (example: Male/Female/Other): female
--Enter Patient Contact (example: 01712345678): 01888043839

Patient registered successfully!ᓚᘏᗢ..^ᓄ♡
```

1.3 View Patients: This option allows the user **to view all the registered patients.** If no patients are registered yet, it will show a message indicating that **the list is empty.** When selected, the system reads from the patients.txt file and displays the full list of patients with the following information:

- ID
- Name
- Age
- Gender
- Contact

```
--Enter your choice (1-6): 2

 [ REGISTERED PATIENTS ]

ID: 10, Name: zarin, Age: 22, Gender: female, Contact: 01888043839

ID: 10, Name: afnan, Age: 21, Gender: female, Contact: 01888043839

End of patient listᓚᘏᗢ..^ᓄ♡.
```

1.4 Search Patient: This option allows the user **to search for a patient by their ID**. If no match is found, the system displays a message

that **the patient was not found.**
When selected, the user is prompted to enter the Patient ID (e.g., 101). The system then searches the patients.txt file. If a match is found, the system shows:

- Patient ID
- Name
- Age
- Gender
- Contact

```
 [ SEARCH PATIENT ]
--Enter Patient ID to search (example:101): 10

Patient Found!(^. .^)∫
~ID: 10
,
~Name: afnan,
~Age: 21,
~Gender: female,
~Contact: 01888043839
```

If no match is found, the system displays a message that the patient was not found.

```
 [ SEARCH PATIENT ]
--Enter Patient ID to search (example:101): 9

Patient with ID 9 not found!(ノ^..^)ノ♡
 Please check the ID and try again.
```

1.5 Update Patient: This option allows the user to update the details of an existing patient.

The user is prompted to enter the Patient ID to be updated. If found, the system allows the user to re-enter the new:

- Name
- Age
- Gender
- Contact Number

The system then **updates the record** in the file, and confirms:

“Patient details updated successfully!”

If the ID does not exist, it informs the user that the **Patient ID was not found**.

```
--Enter your choice (1-6): 4

 [ UPDATE PATIENT ]
--Enter Patient ID to update (example:101): 10
--Enter new details for Patient ID 10 (example:101):
--Enter Name (example: Zeke Yeager): zarin
--Enter Age (example: 25) : 22
--Enter Gender (example: Male/Female/Other): female
--Enter Contact (example: 01712345678): 01888043839

Patient details updated successfully⊂(◉‿◉)つ♡!
```

1.6 Delete Patient: This option allows the user to delete a patient from the system by entering their Patient ID.
If the ID exists, the record is removed from the **patients.txt** file, and a message is shown:

“Patient with ID [ID] deleted successfully!”

If the ID is not found, it displays a **not found message** and keeps the data unchanged.

```
 [ DELETE PATIENT ]
--Enter Patient ID to delete (example:101): 10

Patient with ID 10 deleted successfully!ς(ᵔᵕᵔ)ɔ♡
```

```
--Enter your choice (1-6): 5

 [ DELETE PATIENT ]
--Enter Patient ID to delete (example:101): 9

Patient with ID 9 not found!ς(ᵔᵕᵔ)ɔ♡ Please check the ID and try again.
```

1.7 Exit: This option allows the user to exit the patient registration system.

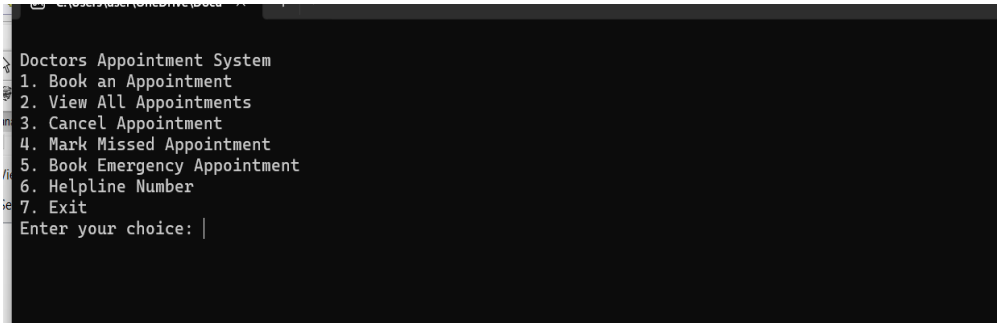
Upon exiting, a farewell message is displayed:

“Exited. Thank you for using our

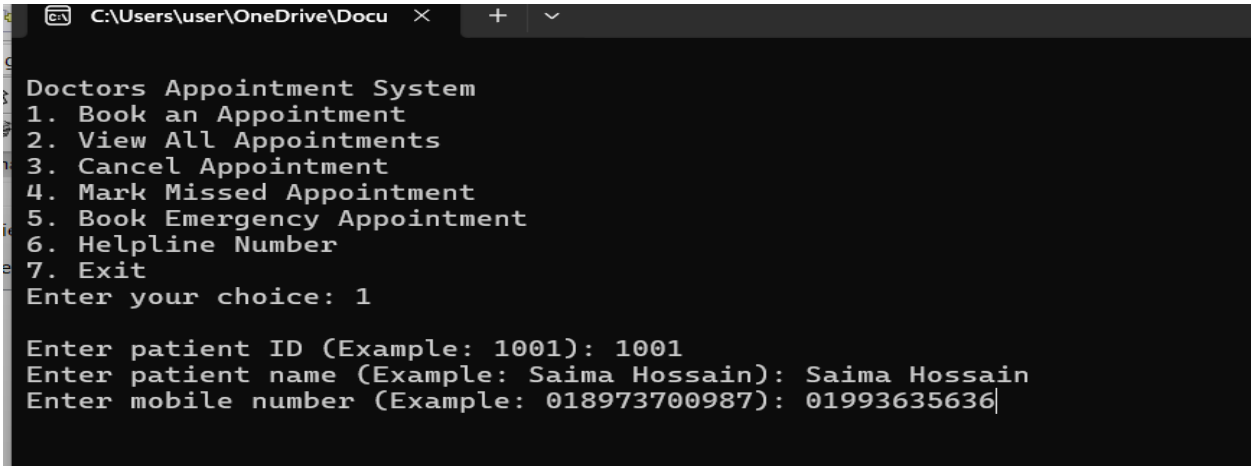
```
--Enter your choice (1-6): 6

Exited. ς(ᵔᵕᵔ)ɔ♡
Thank you for using our system! °❀⋆!
```

2. Doctor's Appointment System:

2.1 Start: This is the main menu of a doctor's Appointment system. It provides options to book an appointment, view all appointments, cancel an appointment, mark a missed appointment, book an emergency appointment, helpline number & exit.



2.2 Book an appointment: This option allows a patient to book an appointment with the system. The patient gives their ID, name, and mobile number. The system then gives a doctor a list.



2.3 Select Doctor and schedule: The patient selects the doctor, schedules an appointment, and gets it. The system then shows that the Appointment was booked successfully.

```
16. Dr. Muhammad Mehrab Hussain - Dermatology (Fee: 800.00)
17. Dr. Shahida Akhtar - Dermatology (Fee: 800.00)
18. Dr. Zakir Hussain - Dermatology (Fee: 800.00)
19. Dr. Mohammed Masud - Dermatology (Fee: 800.00)
20. Dr. Liton Das - Dermatology (Fee: 800.00)
21. Dr. Iqbal Choudhary - Pediatrics (Fee: 1000.00)
22. Dr. Sukumar Kanti Das - Pediatrics (Fee: 1000.00)
23. Dr. Saila Hussain - Pediatrics (Fee: 1000.00)
24. Dr. Sharmin Akter - Pediatrics (Fee: 1000.00)
25. Dr. Turna Barua - Pediatrics (Fee: 1000.00)
Select doctor by number (1-25): 5

Choose time slot:
1. 9:00-11:00 AM
2. 3:00-6:00 PM
3. 8:00-11:00 PM
Enter time slot (1-3): 3

Appointment booked successfully!
Payment of 300.00 made (30% of doctor's fee).
```

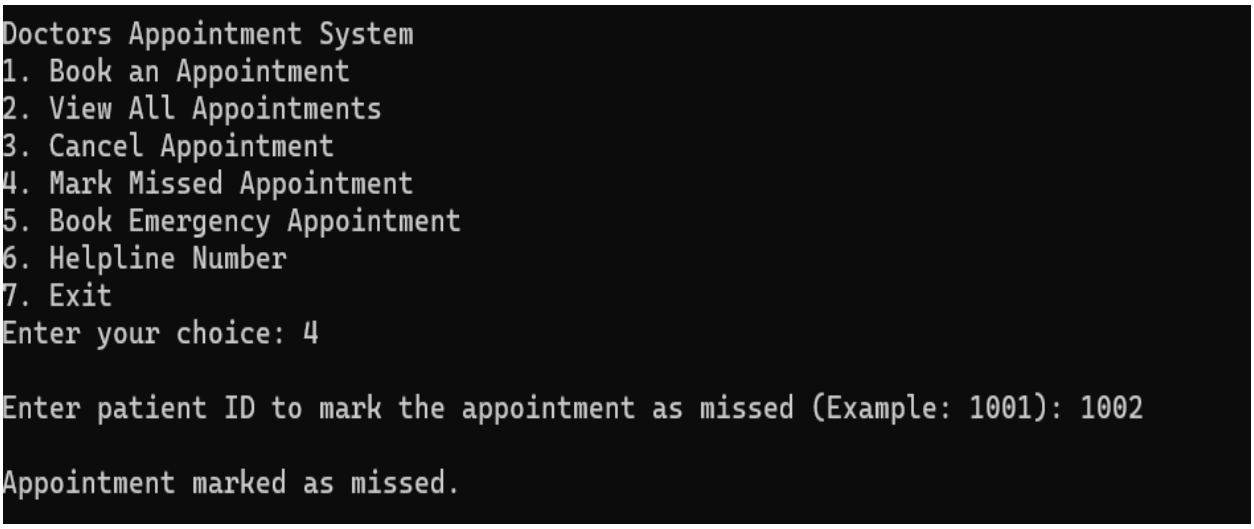
2.4 View all Appointments: When choosing the option, this system provides all appointment records.

```
Doctors Appointment System
1. Book an Appointment
2. View All Appointments
3. Cancel Appointment
4. Mark Missed Appointment
5. Book Emergency Appointment
6. Helpline Number
7. Exit
Enter your choice: 2

List of Appointments:
Patient ID: 1002, Name: Saima Hossain, Doctor: Arors Ahmed, Time Slot: 1, Payment: 450.00, Missed: No, Emergency: Yes
Patient ID: 1001, Name: Saima Hossain, Doctor: Rupashree Bisshas, Time Slot: 3, Payment: 300.00, Missed: No, Emergency:
No
```

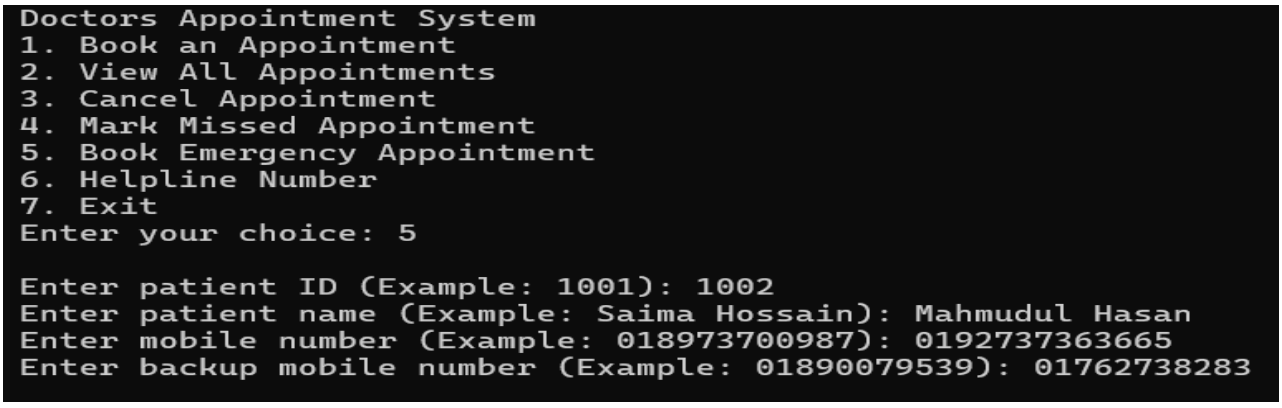
2.5 Cancel Appointments: This option allows you to cancel appointments. When the patient selects the option and gives the patient id then the appointment is canceled successfully.

```
Doctors Appointment System
1. Book an Appointment
2. View All Appointments
3. Cancel Appointment
4. Mark Missed Appointment
5. Book Emergency Appointment
6. Helpline Number
7. Exit
Enter your choice: 3

Enter patient ID to cancel the appointment (Example: 1001): 1001

Appointment canceled successfully!
```

2.6 Mark Missed Appointment: This option allows you to mark a missed appointment. When the patient selects the option and gives the patient ID, the system shows that the appointment is marked as missed successfully.



2.7 Book Emergency Appointment: This option allows a patient to book an emergency appointment through the system. A patient gives their ID, name, and mobile number. After the system gives a doctor a list. The patient selects the doctor and schedules, and gets an emergency appointment. Then the system shows that the Appointment was booked successfully.



```
16. Dr. Muhammad Mehrab Hussain - Dermatology (Fee: 800.00)
17. Dr. Shahida Akhtar - Dermatology (Fee: 800.00)
18. Dr. Zakir Hussain - Dermatology (Fee: 800.00)
19. Dr. Mohammed Masud - Dermatology (Fee: 800.00)
20. Dr. Liton Das - Dermatology (Fee: 800.00)
21. Dr. Iqbal Choudhary - Pediatrics (Fee: 1000.00)
22. Dr. Sukumar Kanti Das - Pediatrics (Fee: 1000.00)
23. Dr. Saila Hussain - Pediatrics (Fee: 1000.00)
24. Dr. Sharmin Akter - Pediatrics (Fee: 1000.00)
25. Dr. Turna Barua - Pediatrics (Fee: 1000.00)
Select doctor by number (1-25): 5

Choose time slot:
1. 9:00-11:00 AM
2. 3:00-6:00 PM
3. 8:00-11:00 PM
Enter time slot (1-3): 3

Appointment booked successfully!
Payment of 300.00 made (30% of doctor's fee).
```

2.8 Helpline Number: This option allows you to get the helpline number. When a patient selects the option, the system gives the helpline number.

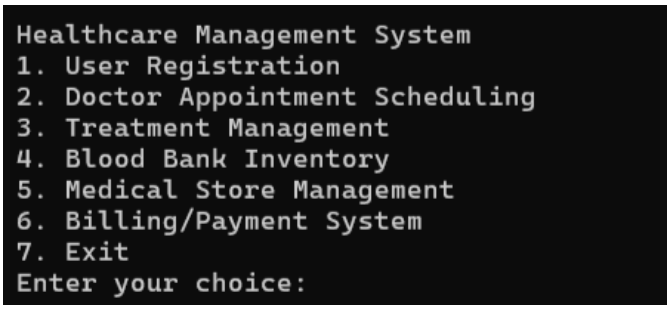
```
Doctors Appointment System
1. Book an Appointment
2. View All Appointments
3. Cancel Appointment
4. Mark Missed Appointment
5. Book Emergency Appointment
6. Helpline Number
7. Exit
Enter your choice: 6

Helpline Numbers:
9990930, 89000979
```

2.9 Exit: This option allows the user to exit the system. Upon exiting, a message confirming that data is saved successfully is displayed, and the system closes.

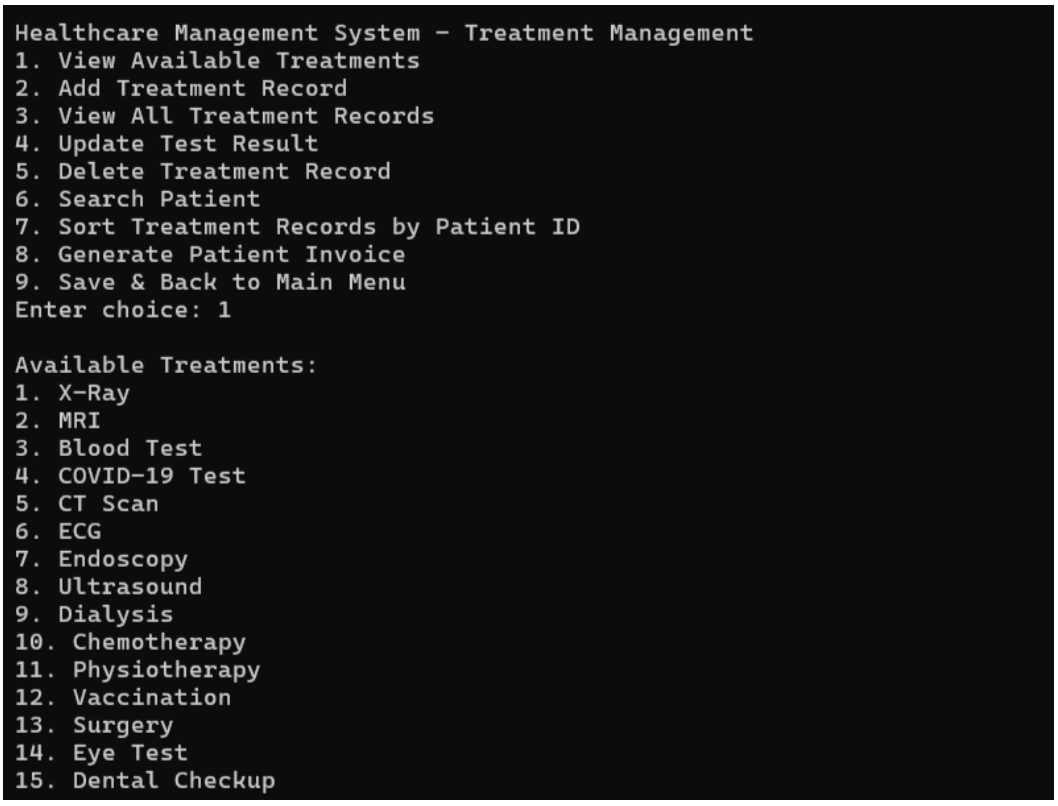
```
Doctors Appointment System
1. Book an Appointment
2. View All Appointments
3. Cancel Appointment
4. Mark Missed Appointment
5. Book Emergency Appointment
6. Helpline Number
7. Exit
Enter your choice: 7
Exiting...
```

3. Treatment Management System:

This module manages patient treatment records, including adding new treatments, updating test results, searching patient records, generating invoices, and sorting records by patient ID. It also tracks treatment history and provides alerts for pending tests.



3.1: View Available Treatments: Displays a comprehensive list of all medical treatments and diagnostic services available at the healthcare facility. This includes common procedures like X-rays, MRIs, blood tests, and specialized treatments. The list is presented in a numbered format for easy reference, showing both the treatment name and its category. This feature helps medical staff quickly verify available services before recommending or recording treatments.



3.2: Add Treatment Record: Allows healthcare providers to create new treatment entries for patients. The system prompts for essential details, including patient ID, name, age, and contact information. Staff can select from the available treatments list and specify the treatment date. Each new record is automatically marked with a "Pending" status until test results are updated. The feature ensures all required fields are completed before saving.

```
Healthcare Management System - Treatment Management
1. View Available Treatments
2. Add Treatment Record
3. View All Treatment Records
4. Update Test Result
5. Delete Treatment Record
6. Search Patient
7. Sort Treatment Records by Patient ID
8. Generate Patient Invoice
9. Save & Back to Main Menu
Enter choice: 2
Enter Patient ID: 1234
Enter Name: Mukit
Enter Age: 22
Enter Mobile Number: 01796124352

Available Treatments:
1. X-Ray
2. MRI
3. Blood Test
4. COVID-19 Test
5. CT Scan
6. ECG
7. Endoscopy
8. Ultrasound
9. Dialysis
10. Chemotherapy
11. Physiotherapy
12. Vaccination
13. Surgery
14. Eye Test
15. Dental Checkup
Select Treatment Type (Enter Number): 3
Enter Treatment Date (DD/MM/YYYY): 12/12/2024

Treatment record added successfully!
```

3.3: View All Treatment Records: Provides a complete overview of all treatment records in the system. Displays information in a tabular format showing patient ID, name, treatment type, date, and current status. The view can be scanned quickly to identify pending treatments or review historical procedures. Data is presented in chronological order by default, with clear column headers for easy navigation.

```
Healthcare Management System - Treatment Management
1. View Available Treatments
2. Add Treatment Record
3. View All Treatment Records
4. Update Test Result
5. Delete Treatment Record
6. Search Patient
7. Sort Treatment Records by Patient ID
8. Generate Patient Invoice
9. Save & Back to Main Menu
Enter choice: 3

ID         Name                Age   Mobile          Treatment           Date            Result
-------------------------------------------------------------------------------------------
4567       Fahim               19    01765235489     Surgery             30/10/2025      Pending
9876       Sakib               25    01521736424     ECG                 20/03/2025      Pending
1234       Mukit               22    01796124352     Blood Test          12/12/2024      Pending
```

3.4: Update Test Result: Enables medical staff to modify treatment outcomes and status. Authorized users can search for treatments by patient ID and change results from "Pending" to "Completed" or other statuses. The system allows adding detailed findings and maintains an audit trail of all updates. Critical result updates can trigger notifications to relevant healthcare providers when integrated with a messaging system.

```
Healthcare Management System - Treatment Management
1. View Available Treatments
2. Add Treatment Record
3. View All Treatment Records
4. Update Test Result
5. Delete Treatment Record
6. Search Patient
7. Sort Treatment Records by Patient ID
8. Generate Patient Invoice
9. Save & Back to Main Menu
Enter choice: 4
Enter Patient ID to update result: 1234
Enter new result: Negative

Result updated successfully!
```

3.5: Delete Treatment Record: Provides functionality to remove incorrect or duplicate treatment entries from the system. Requires confirmation before deletion to prevent accidental data loss. The feature completely erases all associated treatment data while

maintaining database integrity. A log of deletions is maintained for administrative review and compliance purposes.

```
Healthcare Management System - Treatment Management
1. View Available Treatments
2. Add Treatment Record
3. View All Treatment Records
4. Update Test Result
5. Delete Treatment Record
6. Search Patient
7. Sort Treatment Records by Patient ID
8. Generate Patient Invoice
9. Save & Back to Main Menu
Enter choice: 5
Enter Patient ID to delete record: 4567

Record deleted successfully!
```

3.6: Search Patient: Offers flexible search capabilities to locate specific treatment records. Supports queries by both patient ID and name, with partial match functionality. Search results display all matching records with key details for verification. This feature significantly reduces time spent locating patient histories in large databases.

```
Healthcare Management System - Treatment Management
1. View Available Treatments
2. Add Treatment Record
3. View All Treatment Records
4. Update Test Result
5. Delete Treatment Record
6. Search Patient
7. Sort Treatment Records by Patient ID
8. Generate Patient Invoice
9. Save & Back to Main Menu
Enter choice: 6
Enter Patient ID or Name to search: 1234
Patient Found: Mukit
```

3.7: Sort Treatment Records by Patient ID: Reorganizes the treatment database numerically by patient identifier. This sorting method groups all treatments for individual patients together in chronological order. Particularly useful for generating comprehensive

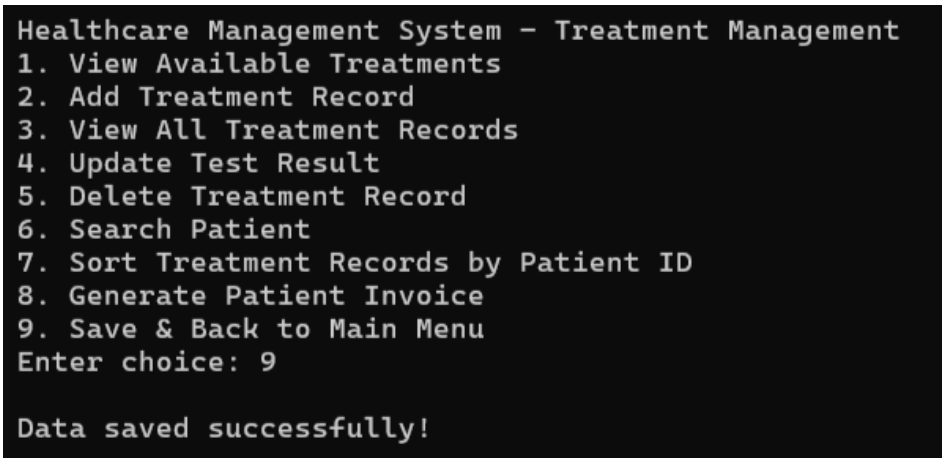
patient histories or preparing for medical reviews. Maintains consistent organization even as new records are added.

```
Healthcare Management System - Treatment Management
1. View Available Treatments
2. Add Treatment Record
3. View All Treatment Records
4. Update Test Result
5. Delete Treatment Record
6. Search Patient
7. Sort Treatment Records by Patient ID
8. Generate Patient Invoice
9. Save & Back to Main Menu
Enter choice: 7

Treatments sorted by Patient ID.
```

3.8: Generate Patient Invoice: Creates detailed, professional invoices for all treatments administered to a specific patient. Automatically calculates totals based on treatment types and includes breakdowns of individual procedures. The standardized format includes patient details, treatment dates, costs, and payment status. Invoices can be printed or saved digitally for records.

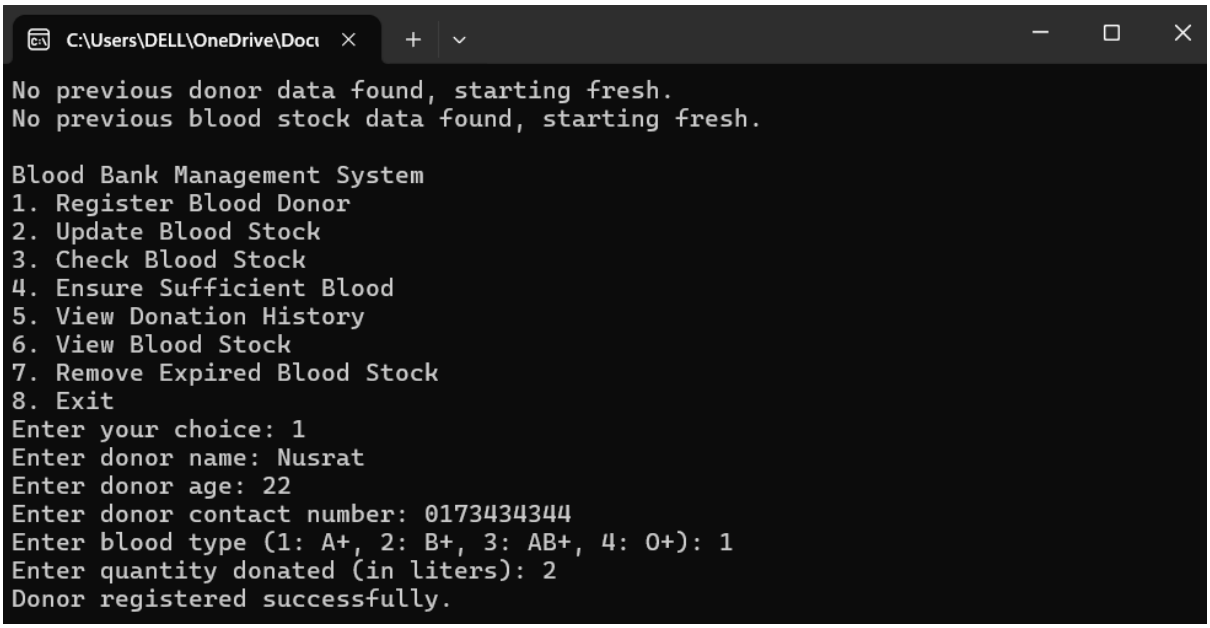
```
Healthcare Management System - Treatment Management
1. View Available Treatments
2. Add Treatment Record
3. View All Treatment Records
4. Update Test Result
5. Delete Treatment Record
6. Search Patient
7. Sort Treatment Records by Patient ID
8. Generate Patient Invoice
9. Save & Back to Main Menu
Enter choice: 8
Enter Patient ID for Invoice: 1234


                        The Healthcare Management System
                        24/7 Hotline: 017-12345678
                        Email: info@healthcare-system.com
================================================================================

                                PATIENT INVOICE

  Patient ID: 1234
  Name: Mukit
  Age: 22
  Mobile: 01796124352
--------------------------------------------------------------------------------
  Treatment History:
  +--------------------+--------------+---------------+
  | Treatment Type     | Date         | Result        |
  +--------------------+--------------+---------------+
  | Blood Test         | 12/12/2024   | Negative      |
  +--------------------+--------------+---------------+

================================================================================
  Important Notes:
  1. Please bring this invoice for any future references
  2. Contact our hotline for any emergency or query
  3. Payment should be made within 7 days of treatment
  4. Keep your patient ID safe for future appointments
================================================================================
```

3.9: Save & Back to Main Menu: Securely commits all treatment data to the system database before exiting. Provides visual confirmation that records have been successfully saved and any unsaved changes are preserved. Returns users to the main healthcare management interface while maintaining system state. Includes automatic backup functionality to prevent data loss during transitions.

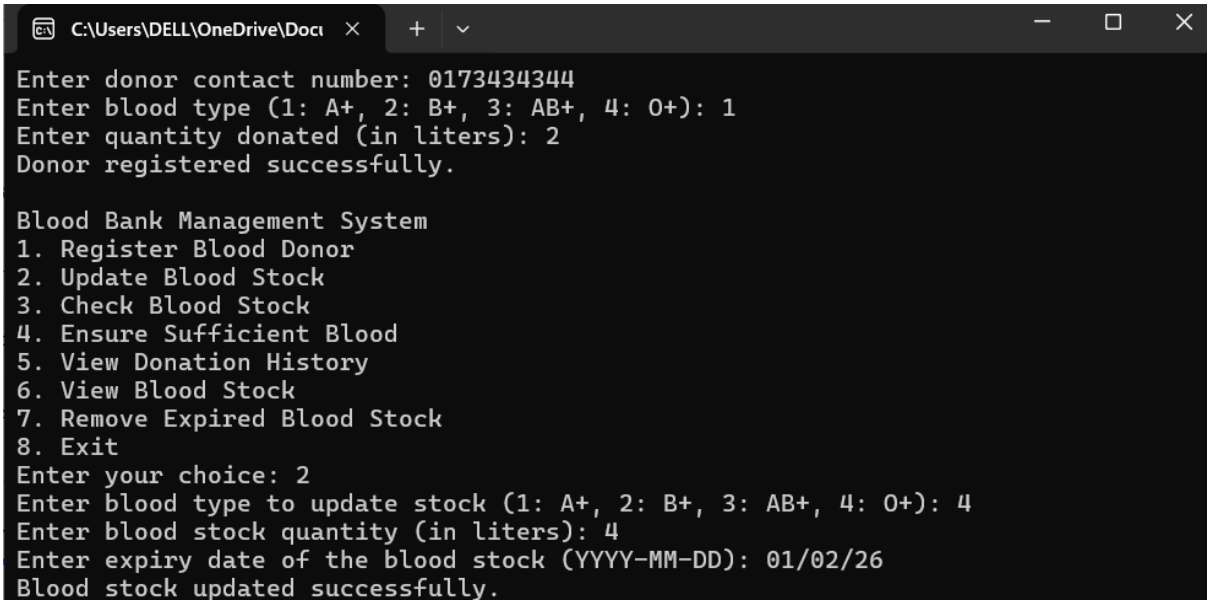


4. Blood Bank :

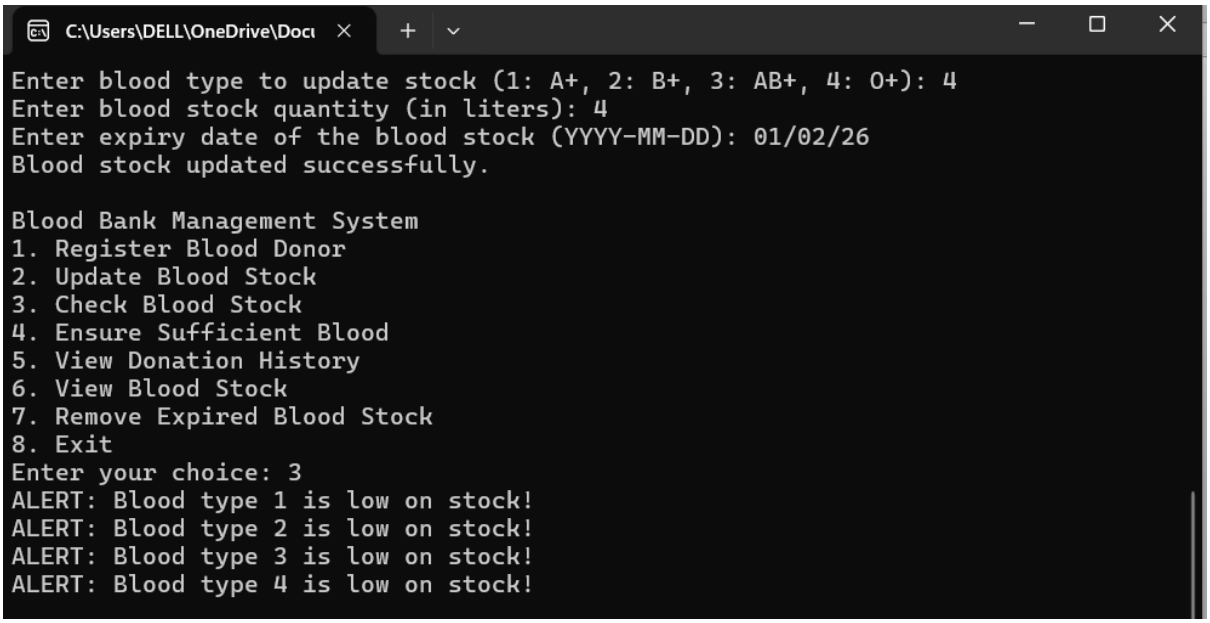
4.1 Register Blood Doner: "Register blood donor" generally refers to the process of signing up someone as a blood donor in an official system or database. Information usually required: Name, Age, Contact number, blood group etc.



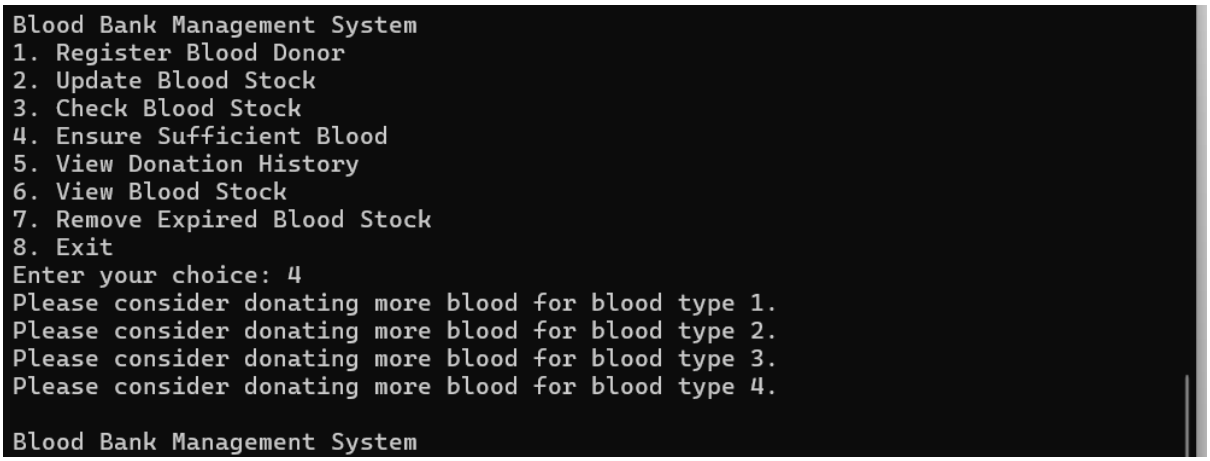
4.2 Update Blood Stock: "Update blood stock" refers to the process of assessing, monitoring, and adjusting the inventory of blood available for use in hospitals, clinics, and medical centers. Blood stock includes different types of blood (e.g., whole blood and red blood cells) that are collected from blood donors.



4.3 Check Blood Stock: "Check blood stock" refers to the process of reviewing and assessing the current inventory of blood available at a blood bank or donation center. This involves evaluating the amount, types, and freshness of the blood products (such as whole blood and red blood cells) stored and ready for use.



4.4 Ensure Sufficient Blood: "Ensure sufficient blood" refers to the efforts and actions taken to make sure that there is enough supply of blood available to meet the needs of patients requiring transfusions, medical treatments, or surgeries. It involves maintaining a steady, reliable stock of donated blood at blood banks, hospitals, or clinics so that it can be used whenever needed.



4.5 View Donation History: "View donation history" refers to the ability to check or access a record of all previous blood donations made by an individual. This history typically includes information such as:

1. **Dates of Donations**
2. **Type of Blood Donated**
3. **Quantity Donated**

```
Blood Bank Management System
1. Register Blood Donor
2. Update Blood Stock
3. Check Blood Stock
4. Ensure Sufficient Blood
5. View Donation History
6. View Blood Stock
7. Remove Expired Blood Stock
8. Exit
Enter your choice: 5

Donation History:
Name    Age     Contact         Blood Type      Quantity (L)    Donation Date
Nusrat  22      0173434344      A+      2.00            2025-04-06
```

4.6 View Blood Stock: "View blood stock" refers to the process of checking the current inventory of blood available at a blood bank, hospital, or donation center. This stock includes various types of blood and blood components (such as whole blood and red blood cells) that are ready for use in medical treatments, surgeries, and emergencies.

```
Blood Bank Management System
1. Register Blood Donor
2. Update Blood Stock
3. Check Blood Stock
4. Ensure Sufficient Blood
5. View Donation History
6. View Blood Stock
7. Remove Expired Blood Stock
8. Exit
Enter your choice: 6

Blood Stock:
Blood Type: A+, Stock: 0.00 liters, Expiry Date:
Blood Type: B+, Stock: 0.00 liters, Expiry Date:
Blood Type: AB+, Stock: 0.00 liters, Expiry Date:
Blood Type: O+, Stock: 4.00 liters, Expiry Date: 01/02/26
```

4.7 Remove Expired Blood Stock: "Remove expired blood stock" refers to the process of identifying and discarding blood that has passed its expiration date and is no longer safe or effective for medical use. Blood and its components (such as whole blood and red blood

cells) have a limited shelf life, and once they expire, they must be removed from the inventory to ensure patient safety.

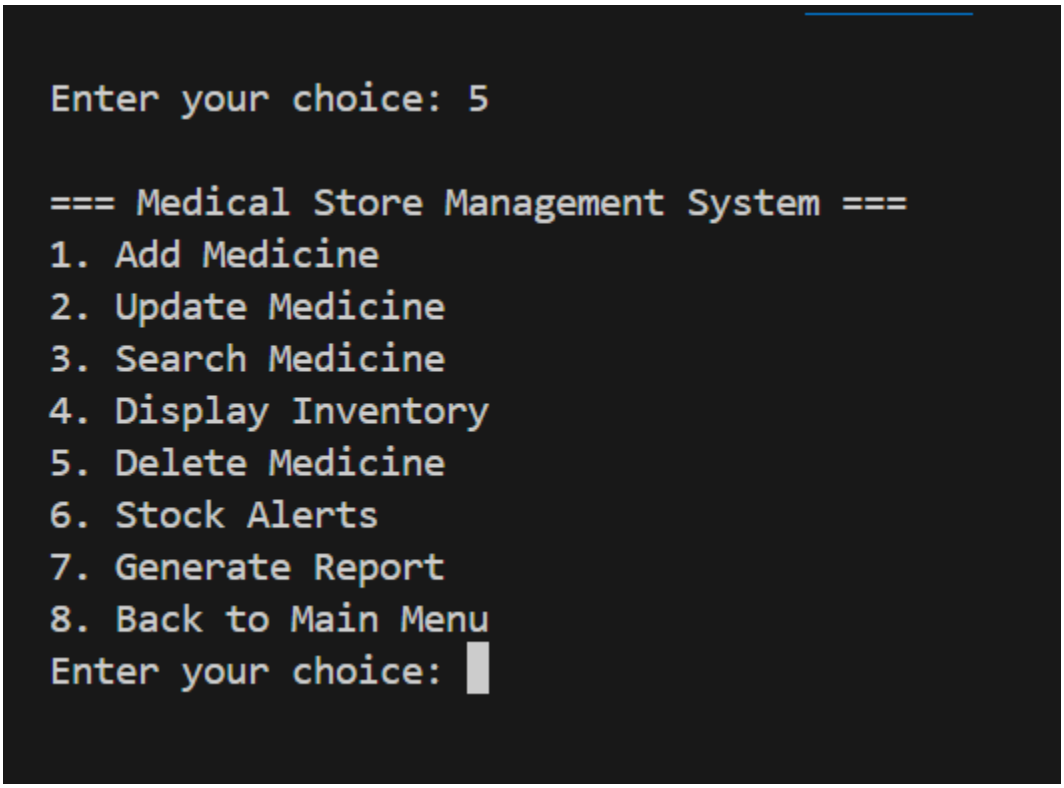
```
Blood Bank Management System
1. Register Blood Donor
2. Update Blood Stock
3. Check Blood Stock
4. Ensure Sufficient Blood
5. View Donation History
6. View Blood Stock
7. Remove Expired Blood Stock
8. Exit
Enter your choice: 7
Blood type 2 has expired. Removing from stock.
Blood type 3 has expired. Removing from stock.
Blood type 4 has expired. Removing from stock.
```

4.8 Exit: In the context of a blood bank, the term **"exit"** typically refers to the **process of blood leaving the blood bank** for distribution or use in medical settings.

```
Blood Bank Management System
1. Register Blood Donor
2. Update Blood Stock
3. Check Blood Stock
4. Ensure Sufficient Blood
5. View Donation History
6. View Blood Stock
7. Remove Expired Blood Stock
8. Exit
Enter your choice: 8
Exiting program.

Process returned 0 (0x0)   execution time : 199.809 s
Press any key to continue.
```

5. Medical Store Management:

5.1 Medical store management: This option allows a user to add medicine, update medicine, search medicine, display inventory, delete medicine, set stock alerts, generate reports & and return to the main menu.



5.2 Add Medicine: This option allows the user to add new medicine into the inventory by providing the medicine ID, the Medicine Name, the Medicine category, Quantity, Price, Manufacturing Date, & Expiry Date.

```
Enter your choice: 1
Enter Medicine ID: 102
Enter Medicine Name: napa
Enter Medicine Category: fever
Enter Quantity: 35
Enter Price: 350
Enter Manufacturing Date (DD-MM-YYYY): 20-05-2028
Enter Expiry Date (DD-MM-YYYY): 30-06-2032
Medicine added successfully!
```

5.3 Update Medicine: This option allows the user to update medicines in the inventory by providing the medicine ID, then adding the new Quantity of the & the new price of the medicine.

```
Enter Medicine ID to update: 102
Current Details:
Name: napa
Category: fever
Quantity: 35
Price: 350.00
Expiry: 30-06-2032
Enter new quantity (>= 0): 04
Enter new price (> 0): 40
Medicine updated successfully!
```

5.4 Search Medicine: This option allows the user to search for medicines in the inventory by providing the medicine Name, otherwise the Medicine category & the medicine ID. It helps to find our needed medicine easily with short details(like quantity & expiry dates).

```
Search by:
1. Name
2. Category
3. ID
Choice: 1
Enter medicine name: napa

ID: 102
Name: napa
Category: fever
Quantity: 4
Price: 40.00
Expiry: 30-06-2032
```

5.5 Display Inventory: This option allows the user to Display medicines in the inventory by selecting the option & it shows everything how many medicines & price, quantity, expiry date, etc.

```
Enter your choice: 4

ID    Name                Category        Quantity   Price      Expiry
-------------------------------------------------------------
102   napa                fever           4          $40.00     30-06-2032
103   tafnil              headache        45         $560.00    30-06-2035
```

5.6 Delete Medicine: This option allows the user to delete medicines from the inventory by providing the medicine ID then it deletes the medicine with its full details from the inventory.

```
Enter your choice: 5
Enter Medicine ID to delete: 103
Medicine deleted successfully!
```

5.7 Stock Alerts: This option allows the user to stock alert medicines in the inventory by selecting the option & it shows how many medicines. If the medicine quantity is below 5, then it shows in the low stocks; otherwise, it shows nothing.

```
Enter your choice: 6

Stock Alerts:
 [LOW STOCK] napa (ID: 102) - 4 remaining
```

5.8 Generate Report: This option allows the user to generate report of medicines in the inventory by selecting the option & it shows the Inventory valuation report. If the medicine quantity is below 5, then it shows in the low stock. The total inventory value& if any item has expired, then it is also showing that item.

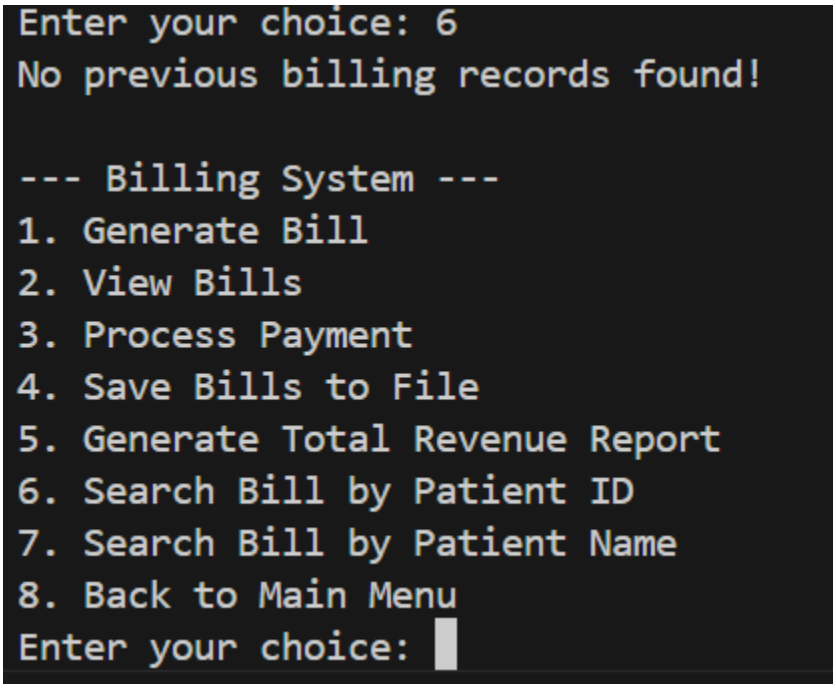
```
Enter your choice: 7

Inventory Valuation Report:
Medicine             Quantity   Unit Price Total Value  Status
napa                 4          $40.00     $160.00      LOW STOCK

Summary:
Total Inventory Value: $160.00
Expired Items: 0
Low Stock Items: 1
```

5.9 Back to Main Menu: This option allows the user back to main menu of the healthcare management system if they finish their work in medicine management or they can easily go back to the main menu by clicking this option.

```
Enter your choice: 8
Returning to main menu...

Healthcare Management System
1. User Registration
2. Doctor Appointment Scheduling
3. Treatment Management
4. Blood Bank Inventory
5. Medical Store Management
6. Billing/Payment System
7. Exit
Enter your choice:
```

6.Billing/Payment system:

6.1 Billing/Payment System: This option allows a user to generate a Bill, View Bills, Process Payment, Save Bills to File, generate Total Revenue Report, Search Bill by Patient ID, Search Bill by Patient Name & and return to the main menu.



6.2 Generate Bill: This option allows a user to generate a Bill by providing Patient ID, Patient Name, Patient Address, Patient Age, Patient Phone Number, Patient Gender, Consultation Fee, Treatment Charges, Test Charges, Medicine Charges, Discount Amount, Tax Percentage .this billing system is complete the healthcare system billing process a patients easily give their bill of their treatment or other fees through this system.

```
Enter your choice: 1
Enter Patient ID: 102
Enter Patient Name: nadim
Enter Patient Address: uttara 13,sector13,Dhaka,Bangladesh
Enter Patient Age: 35
Enter Patient Phone Number: 01745970934
Enter Patient Gender: male
Enter Consultation Fee: 1000
Enter Treatment Charges: 500
Enter Test Charges: 300
Enter Medicine Charges: 250
Enter Discount Amount: 0000
Enter Tax Percentage: 5%

Bill generated successfully!
```

6.3 View Bills: This option allows a user to view a Bill by choosing this option on this function, then it shows the billing details of a patient it showing how much a patient needs to pay, and the pending method, it shows the whole billing records of the patients.

```
Enter your choice: 2

--- Billing Records ---

Bill ID: 1
Patient ID: 102
Total Amount: 2152.50
Paid Amount: 0.00
Due Amount: 2152.50
Payment Method: Pending

Bill ID: 2
Patient ID: 0
Total Amount: 6.12
Paid Amount: 0.00
Due Amount: 6.12
Payment Method: Pending
```

6.4 Process Payment: This option allows a user to pay a Bill by choosing the billing ID of a patient, then he/she show the billing details he/she can pay by his/her bill by cash/card/insurance. After successfully paying the bill. They receive an invoice & the system shows payment processed successfully.

```
Enter your choice: 3
Enter Bill ID for payment: 1
Enter Payment Amount: 2152
Enter Payment Method (Cash/Card/Insurance): cash
Payment processed successfully!
```

6.5 Save Bills to File: This option allows a user to Save Bills to File by choosing this option. After successfully paying the bill. The

system can successfully store the details of billing & we can store the details on the hospital server.

```
Enter your choice: 4
Bills saved successfully!
```

6.6 Generate Total Revenue Report: This option allows a user to generate a Total Revenue Report by choosing this option. Hospital authority can check their revenue easily by clicking this option to see how much balance is on the hospital's inventory. How much payment get from the patients & show the total revenue report.

```
Enter your choice: 5

--- Total Revenue Report ---
Total Revenue: 2152.00
```

6.7 Search Bill by Patient ID: This option allows a user to search for a Bill by patient ID choosing this option. then, input the ID of the patient, the system shows every detail of the patient. The system creates an invoice & the system shows how the payment was processed successfully. How much bill is left to pay.

```
Enter your choice: 6
Enter Patient ID to search: 102

--- Invoice ---
Bill ID: 1
Patient ID: 102
Patient Name: nadim
Patient Address: uttara 13,sector13,Dhaka,Bangladesh
Patient Age: 35
Patient Phone: 01745970934
Patient Gender: male
Consultation Fee: 1000.00
Treatment Charges: 500.00
Test Charges: 300.00
Medicine Charges: 250.00
Discount: 0.00
Tax: 5.00%
Total Amount: 2152.50
Paid Amount: 2152.00
Due Amount: 0.50
Payment Method: cash
```

6.8 Search Bill by Patient Name: This option allows a user to search for a Bill by patient Name choosing this option. then, input the Name of the patient, the system shows every detail of the patient. The system creates an invoice & the system shows how the payment was processed successfully. How much bill is left to pay. Same as the previous option, if the user wants to select & search by name, then this option works otherwise, we can use the previous option. depends on patients also searching for their bill by their name.

```
Enter your choice: 6
Enter Patient ID to search: 102

--- Invoice ---
Bill ID: 1
Patient ID: 102
Patient Name: nadim
Patient Address: uttara 13,sector13,Dhaka,Bangladesh
Patient Age: 35
Patient Phone: 01745970934
Patient Gender: male
Consultation Fee: 1000.00
Treatment Charges: 500.00
Test Charges: 300.00
Medicine Charges: 250.00
Discount: 0.00
Tax: 5.00%
Total Amount: 2152.50
Paid Amount: 2152.00
Due Amount: 0.50
Payment Method: cash
```

6.9 Back to Main Menu: This option allows the user to return to the main menu of the healthcare management system if user finish the work in the billing system, or they can easily go back to the main menu by clicking this option.

```
Enter your choice: 8
Bills saved successfully!
Returning to main menu...
```

6.10 Exit: This option allows the user to exit the healthcare management system if the user finishes the work in their individual system, or they can easily go back by clicking this option.

```
7. Exit
Enter your choice: 7
Exiting Healthcare Management System...
```

7. Exit:

7. Exit: This option allows the user to exit the healthcare management system after completing tasks related to various available functions, or easily navigate back to the previous menu with a single click.


```
7. Exit
Enter your choice: 7
Exiting Healthcare Management System...
```