HEALTHCARE MANAGEMENT SYSTEM

Project Title: HealthCare Management System (HMS)

1. Introduction

1.1 Background

Effective healthcare management is more crucial than ever in our fast-paced world. But even after that a large number of clinics and hospitals continue to manage patient records, appointments, and billing using manual, paper based systems. Most clinics even use just a notepad to take appointments of patients. For both patients and employees, this frequently results in mistakes and needless workload. Smaller healthcare providers find it difficult to find accessible, user friendly digital solutions, while large hospitals may use enterprise grade software. In order to address this issue, our suggested "Healthcare Management System (HMS)" offers a straightforward, role-based & menu driven C application that is offering safe and effective patient care administration.

1.2 Problem Statement

In Bangladesh, managing healthcare operations especially in small to mid-sized clinics is still largely manual and outdated. Patients often face long waiting times for appointments, and they stay confusion about their treatment history, and difficulties communicating with doctors. It's not uncommon for patients to carry paper prescriptions or handwritten reports that are often getting lost or misinterpreted.

Doctors struggle with limited access to a patient's full medical history. And if the patient has visited many facilities, then many confusions arise. This

can lead to repeated tests, misdiagnosis or delays in treatment. Administrative staff often maintain registers manually, handle payments in cash, and schedule appointments without any centralized system, increasing the risk of errors and inefficiency.

For many rural clinics, the situation is even more difficult. They often lack the digital tools or budget to implement expensive hospital software. As a result, both patients and healthcare providers suffer from poor record-keeping, lack of coordination, and overall frustration. There is a clear need for a **low-cost, user-friendly, and reliable healthcare management system** that can bring digital services to these clinics.

1.3 Problem Statement

a) Developing a menu driven complete system where patients, doctors, and admins interact through role-based access.
b) Allow patients to register, log in, and book appointments.
c) Enable doctors to manage their appointments, give prescriptions, and order lab tests.
d) Provide admins the ability to manage patient records and maintain the system.
e) Ensure privacy and security of all sensitive medical data.
f) Build the entire system in C programming language to demonstrate low-level control and efficiency.

2. Project Scope

a) Patients can register, log in, and book doctor appointments online.
b) Patients can report symptoms, view doctor profiles, and confirm booking schedules.
c) Doctors can log in, review appointments, prescribe medication, and request lab tests.
d) Admins can manage patient records and perform post-treatment cleanup (Basically deleting data of already cleared patients).
e) The system supports prescription generation and digital billing.
f) Role-based access ensures that users only see features relevant to their role.
g) The system will be built for single-clinic or single-hospital use first.

Note: Future updates may include multi-hospital support, document uploads, and online payments and also a refined project structure.

3. Methodology

3.1 Tools and Techniques

- **Programming Language Used:** C
- **Development Tools:** GCC compiler, Visual Studio Code or Any Code Editor
- **Data Storage:** Plain-text files (for example users.txt)
- **Libraries:** Standard C libraries
- **Version Control:** Git and GitHub
- **Testing:** Manual testing, unit testing with assertions, peer code reviews

3.2 Phases of our project

Phase 1: Design and Planning

We will define the project's main modules, design the fundamental frameworks, and assess the system requirements during this phase. Diagrams such as flowcharts will be created. There will also be an outline of the time and resource plan used in the gantt chart.

Phase 2: Development

This is the core development stage where the complete functionality of the system will be implemented in C. Modules such as user authentication, appointment booking, and billing will be developed and tested one by one.

Phase 3: Deployment and Testing

To make sure the system functions properly, the last step entails integrating all modules and carrying out extensive manual testing. After testing is finished, the system will be prepared for deployment or demonstration.

4.Requirement Analysis

4.1 Functional Requirements

1. User Registration and Login for patients
2. Role-based login for doctors and admins (this will already be defined in staff.txt)
3. Profile management for patients (view their basic info)

4. Symptom input during appointment booking
5. Viewing available doctors with designation and time slots
6. Appointment booking and confirmation for patients
7. Appointment schedule viewing and management for doctors
8. Prescription creation by doctors
9. Lab test recommendations by doctors
10. Viewing prescriptions by patients
11. Downloading/print prescriptions by patients
12. Billing generation with payment options (bKash, Nagad, Rocket & Bank)
13. Booking cancellation for patients
14. Booking history display
15. Doctor remarks visible to patients after treatment
16. Admin ability to delete patient records after treatment
17. Admin ability to manage system users and clear data
18. Messaging system between patients and doctors

4.2 Non-Functional Requirements

1. Performance

- The system should load data files and respond to user actions within a reasonable time (should take about 2-3 seconds at most).
- Support multiple user sessions sequentially (one at a time).

2. Usability

- Simple command-line interface (CLI) that is easy to navigate.
- Clear prompts and instructions for user inputs.

3. Reliability

- Data stored in files must not be corrupted; proper file handling is essential.
- Prevent data loss on unexpected shutdowns by saving updates immediately.

4. Security

- Basic authentication with password protection.
- Sensitive data (like passwords) will be stored with basic level of encryption.
- Restrict access based on user roles.

5. Maintainability

- Code will be modular and organized.
- We will be using text files for easy access and debugging.

6. Portability

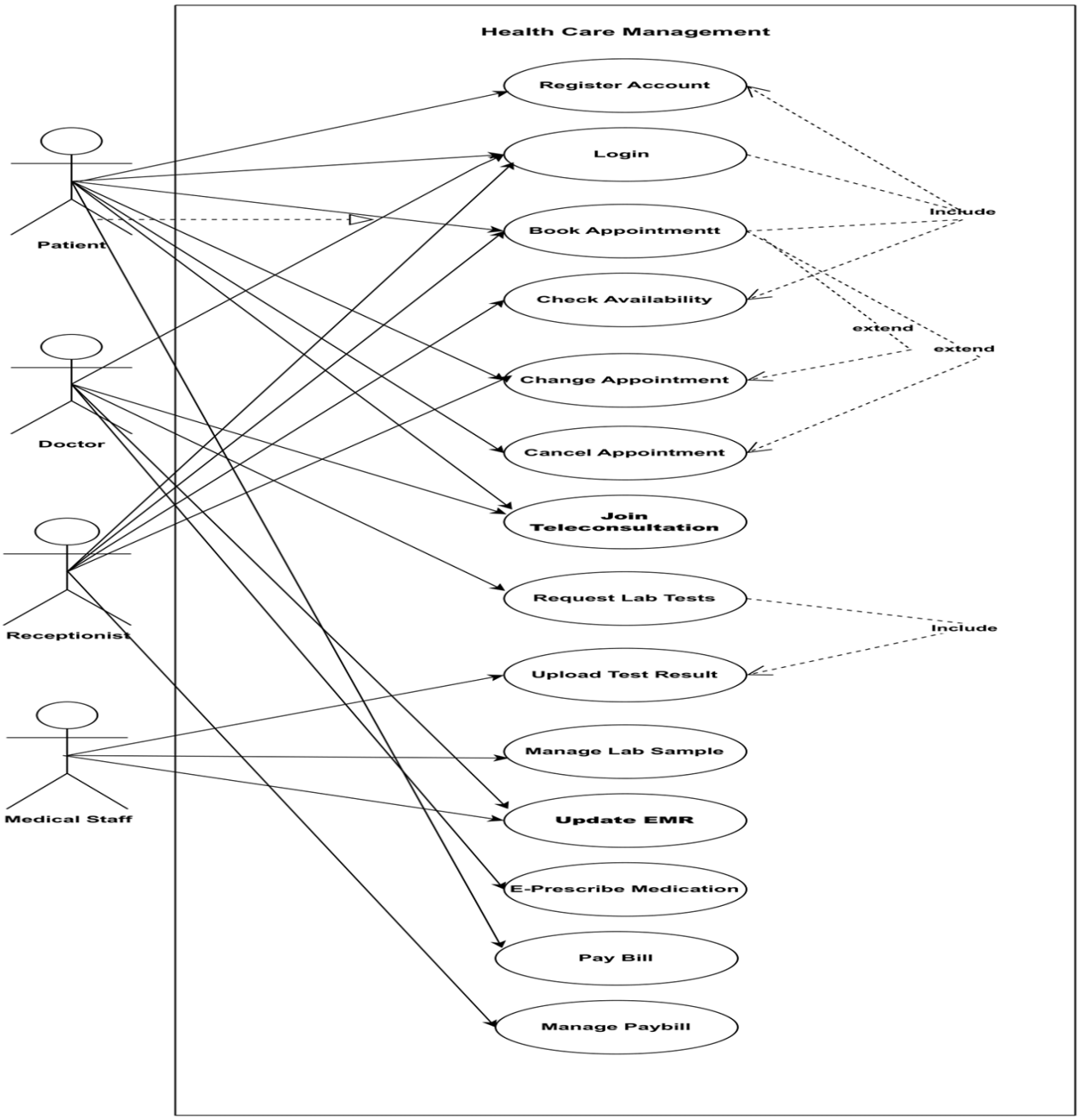
The system should run on any platform supporting standard C

7. Scalability (limited)

Should be able to handle growing number of patients, doctors, and appointments via file updates.

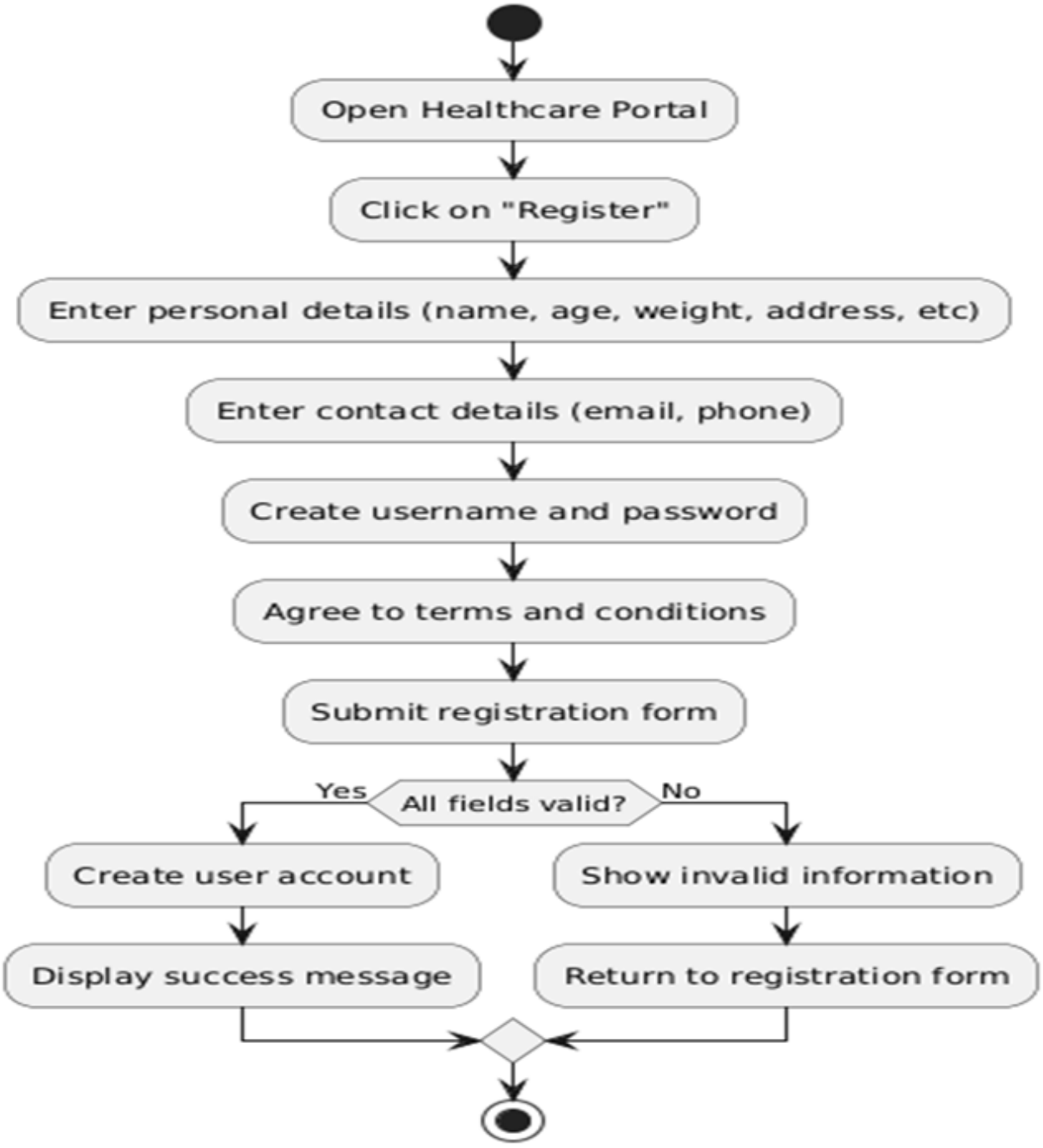
5. Diagrams

5.1 Use Case Diagram

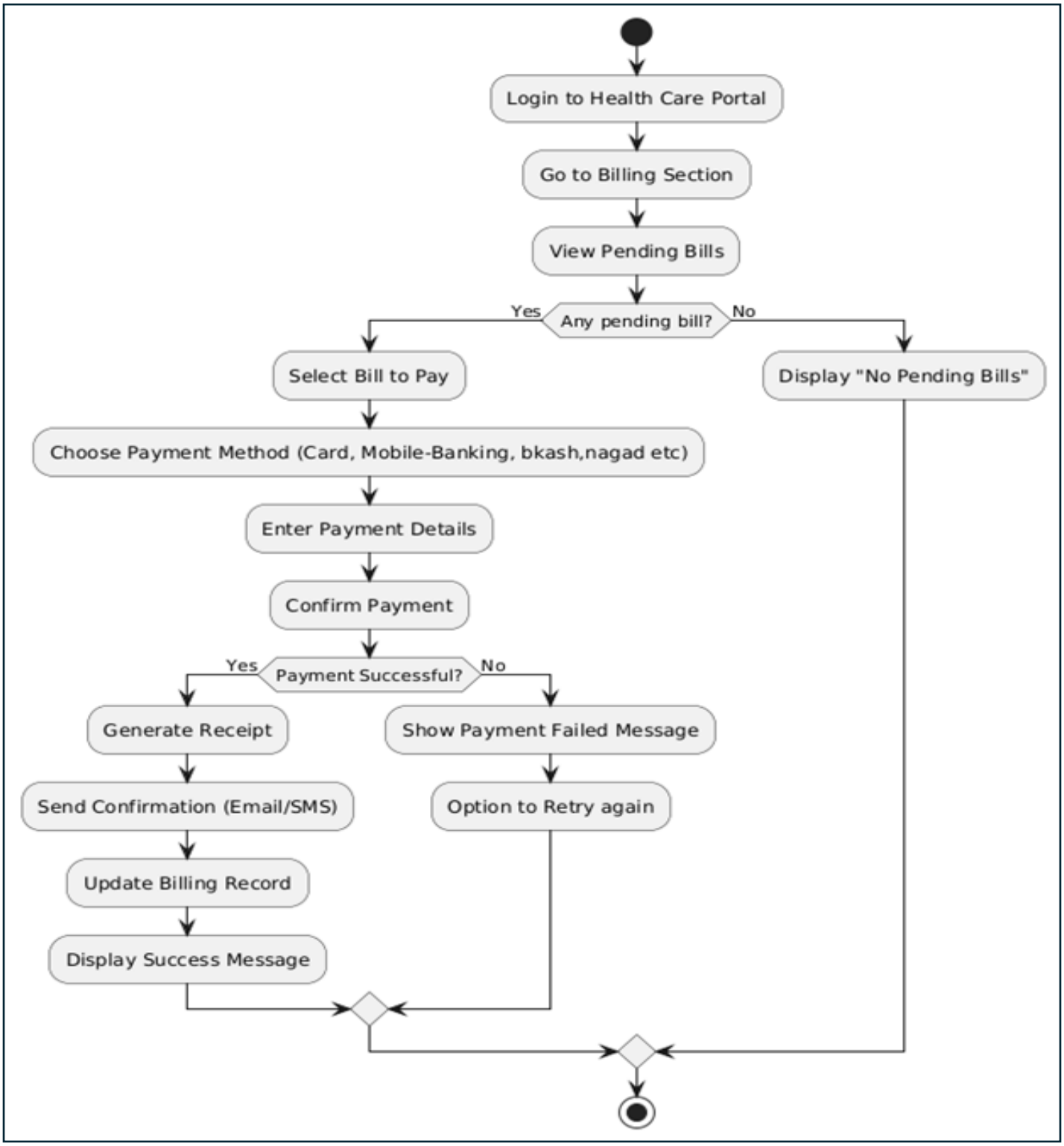


5.2 Activity Diagrams

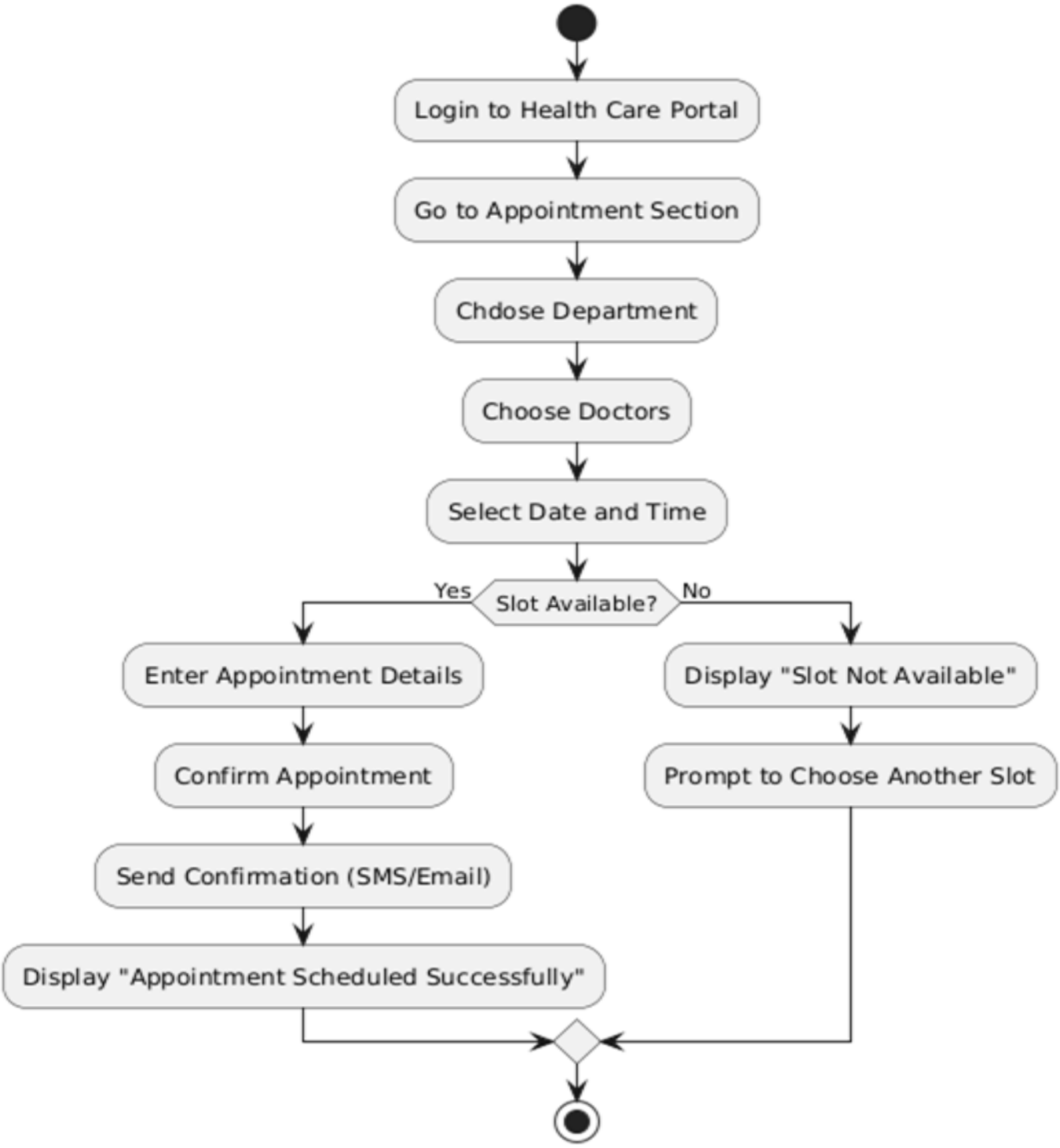
a) User Registration



b) **Billing and Payment**

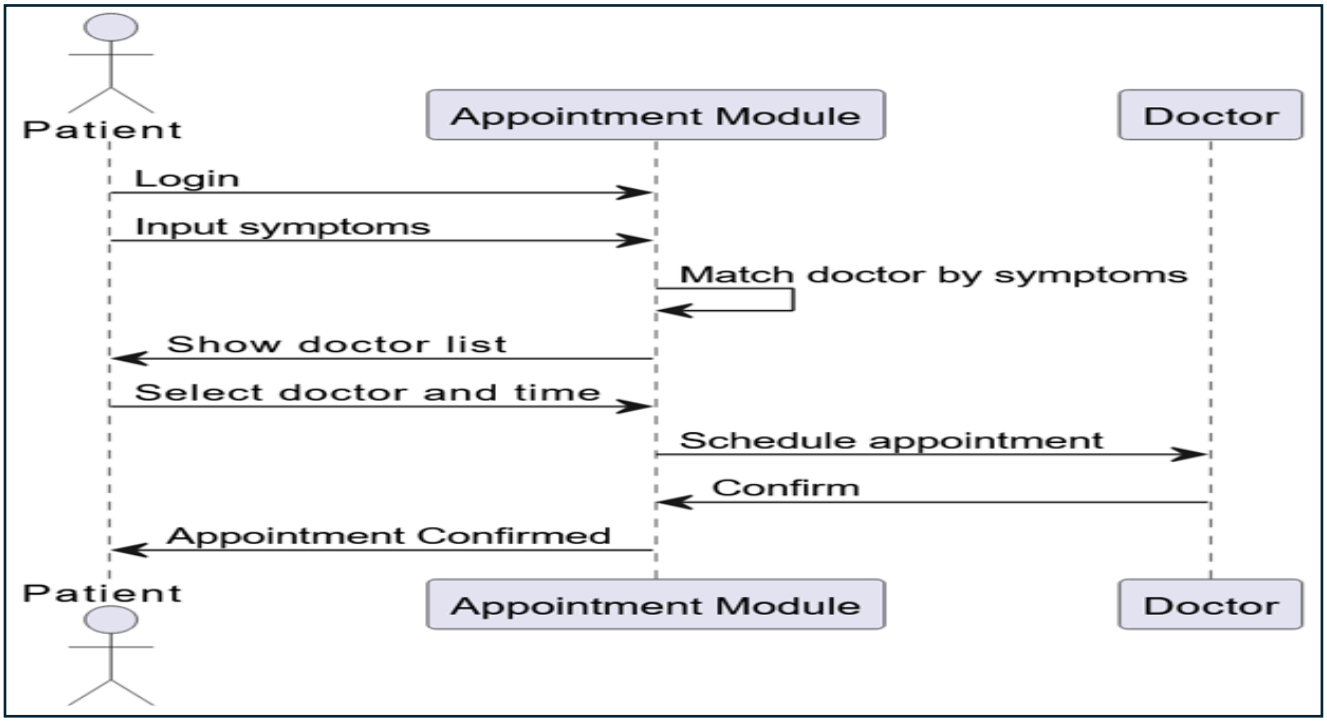


c) **Appointment Booking**

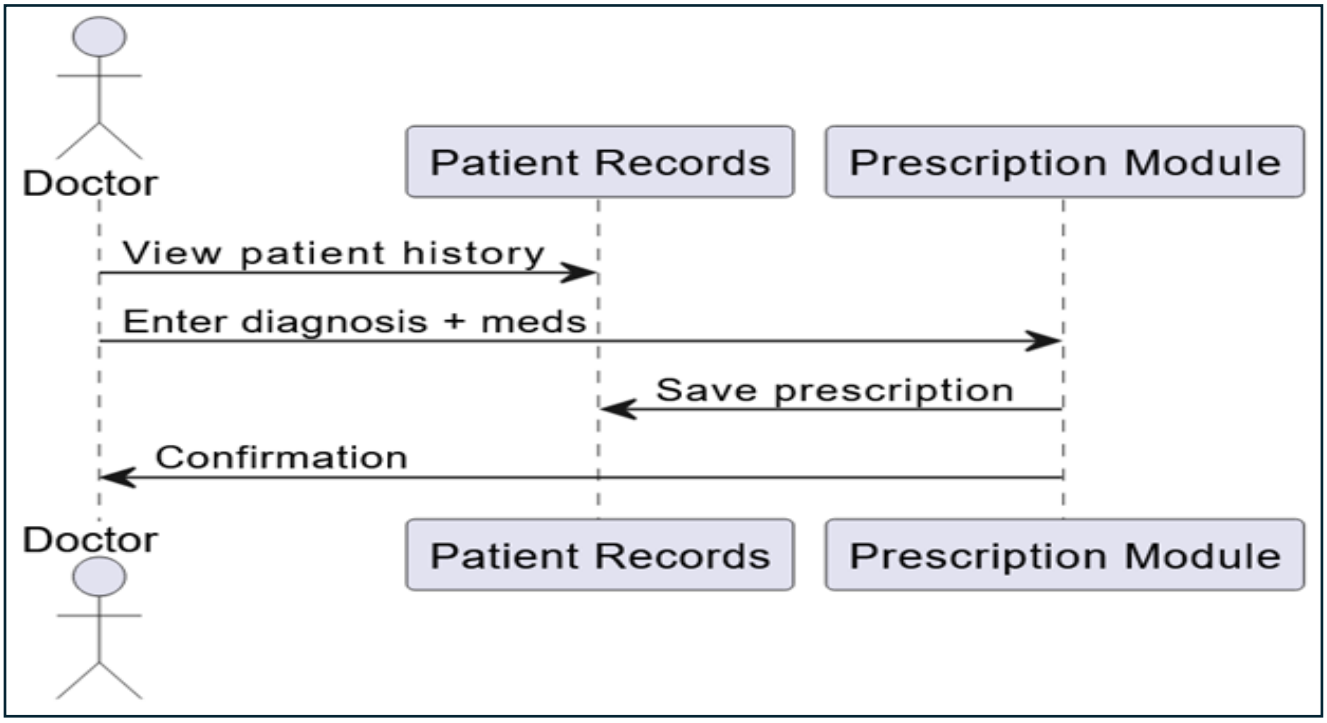


5.3 Sequence Diagrams

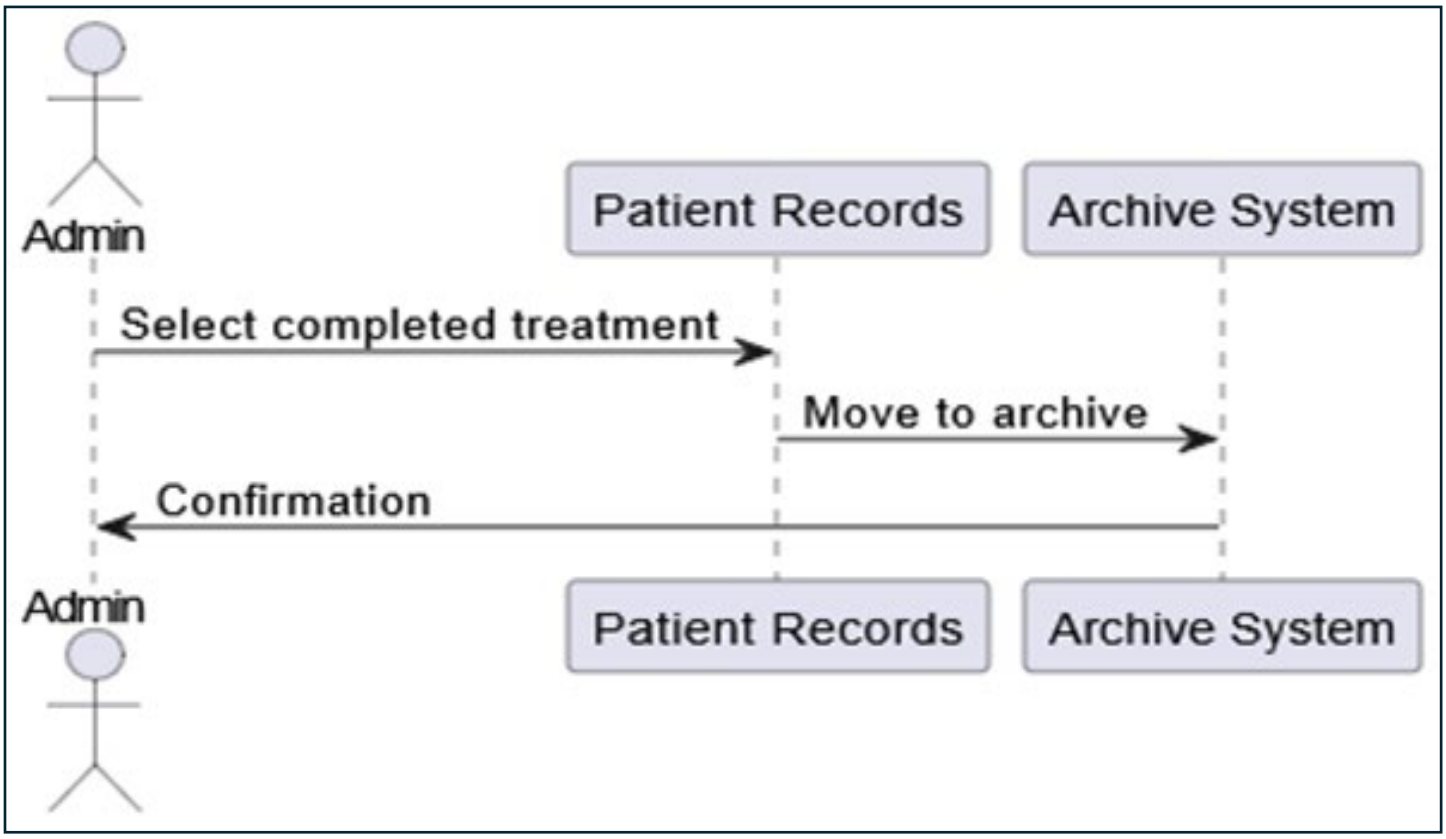
a) Appointment Booking



b) Prescribe Medicine



c) Admin Patient management



6. Use Case Descriptions

A. Patient Registration

Use Case	Patient Registration
Goal	Allow a new user to create an account in the HMS system.
Preconditions	User does not already have an account.
Success End Condition	Account is successfully saved in the system and user is redirected to login.
Failed End Condition	Registration is not completed due to input error or duplicate account.
Primary Actors	Patient
Secondary Actors	IT Support
Trigger	Patient initiates registration from the main menu.
Description / Main Success Scenario	1. Patient selects "Login". 2. System prompts for phone and password. 3. Credentials are matched from patients.txt. 4. On success, access is granted to dashboard.
Alternative Flows	a. Credentials mismatch → Show error and retry prompt
Quality Requirements	A) Login response within 2 seconds of input. B) Maximum 3 login attempts before lockout suggestion.
Security Requirements	A) Passwords must not be displayed during input. B) All login attempts must be logged.
Postconditions	On success: Patient is logged in. On failure: System stays on login screen.

B . Patient Login

Use Case	Patient Login
Goal	Authenticate the patient to access personal dashboard.
Preconditions	Patient is already registered.

Use Case	Patient Login
Success End Condition	Patient is logged in and taken to their dashboard.
Failed End Condition	Invalid credentials; login attempt fails.
Primary Actors	Patient
Secondary Actors	IT Support
Trigger	Patient selects "Login" from the menu.
Description / Main Success Scenario	1. Patient selects "Login". 2. System prompts for phone and password. 3. Credentials are matched from patients.txt. 4. On success, access is granted to dashboard.
Alternative Flows	Step 3a. Credentials mismatch → Show error and retry prompt.
Quality Requirements	Login response within 2 seconds of input.
Security Requirements	Maximum 3 login attempts before lockout suggestion.
Postconditions	On success: Patient is logged in. On failure: System stays on login screen.

C. Doctor/Admin Login

Use Case	Doctor/Admin Login
Goal	Authenticate predefined staff (Doctor/Admin) using stored credentials.
Preconditions	User exists in staff.txt.
Success End Condition	User sees a role-specific welcome screen.
Failed End Condition	Access denied due to incorrect info.
Primary Actors	Doctor, Admin
Secondary Actors	IT Support

Use Case	Doctor/Admin Login
Trigger	User selects role and attempts login.
Description / Main Success Scenario	1. Staff selects their role (Doctor/Admin). 2. Enters username and password. 3. System checks credentials in staff.txt. 4. If valid, shows respective welcome screen.
Alternative Flows	Step 3a. Mismatch → Display error and retry option.
Quality Requirements	Login validation within 2 seconds.
Security Requirements	1. Role-based access control must be enforced. 2. Staff login attempts should be logged.
Postconditions	1. On success: Access granted to staff module. 2. On failure: User remains on login screen.

D. Appointment Booking

Use Case	Appointment Booking
Goal	Patient books an appointment with a doctor.
Preconditions	Patient is logged in and profile is complete.
Success End Condition	Appointment is confirmed and saved.
Failed End Condition	Time slot unavailable or input error.
Primary Actors	Patient
Secondary Actors	Doctor, Scheduling Module
Trigger	Patient chooses "Book Appointment".
Description / Main Success Scenario	1. Patient selects "Book Appointment". 2. Enters symptoms.

Use Case	Appointment Booking
	3. Views list of available doctors. 4. Selects doctor and available time. 5. System confirms and saves appointment.
Alternative Flows	Step 4a. No slots → Display message and reschedule option.
Quality Requirements	Slot check and confirmation should complete within 3 seconds.
Security Requirements	Appointment data must be stored securely and protected from unauthorized edits.
Postconditions	On success: Appointment is saved and visible to both doctor and patient. On failure: No changes; retry possible.

E. Appointment Management

Use Case	Appointment Management
Goal	Doctor views and manages upcoming appointments.
Preconditions	Doctor is logged in.
Success End Condition	Appointments are updated and saved.
Failed End Condition	Updates are not saved or access denied.
Primary Actors	Doctor
Secondary Actors	Appointment Scheduler
Trigger	Doctor selects “Manage Appointments”.
Description / Main Success Scenario	1. Doctor logs in to the system. 2. System displays today’s scheduled appointments. 3. Doctor selects a patient case from the list. 4. Doctor marks patient as “Seen” or “Treated”. 5. System updates appointment status and saves it.

Use Case	Appointment Management
Alternative Flows	A) Step 2a. No appointments → Show "No Appointments Today". B) Step 5a. File error → Show error message and allow retry.
Quality Requirements	A) Schedule must load within 2 seconds. B) Appointment status update must be saved within 1 second.
Security Requirements	A) Only authorized doctors can access and modify their appointments. B) Activity logs must track all changes.
Postconditions	A) On success: Appointment is updated in the system. B) On failure: No changes made; retry possible.

F. Prescription Creation

Use Case	Prescription Creation
Goal	Doctor creates a prescription for a patient.
Preconditions	Appointment exists.
Success End Condition	Prescription is saved in prescriptions.txt.
Failed End Condition	Prescription not saved due to input or file error.
Primary Actors	Doctor
Secondary Actors	Patient Record System
Trigger	Doctor selects "Create Prescription" after consultation.
Description / Main Success Scenario	1. Doctor opens patient appointment. 2. Inputs diagnosis and medicines. 3. Optionally adds lab test recommendations. 4. System saves prescription in prescriptions.txt.

Use Case	Prescription Creation
	5. Confirmation message shown to doctor.
Alternative Flows	A) Step 2a. No input → Show validation error. B) Step 4a. File save fails → Retry or alert admin.
Quality Requirements	A) Save operation must complete within 2 seconds.
Security Requirements	A) Prescriptions must be accessible only by assigned doctor and the patient. B) Prescription entries should be timestamped and not editable once saved.
Postconditions	A) On success: Prescription is stored for future viewing. B) On failure: Error; re-entry required.

G. Patient Record Management

Use Case	Patient Record Management
Goal	Admin removes or archives patient data after treatment completion.
Preconditions	Treatment marked completed.
Success End Condition	Patient data removed or backed up.
Failed End Condition	File operation fails; data remains unchanged.
Primary Actors	Admin
Secondary Actors	File System
Trigger	Admin selects "Manage Records".
Description / Main Success Scenario	1. Admin logs in. 2. Selects “Completed Patients” list.

Use Case	Patient Record Management
	3. Chooses a patient to archive or delete. 4. System confirms and performs the action. 5. Record is removed from active list or stored in archive.
Alternative Flows	A) Step 4a. Confirmation declined → No action taken. B) Step 5a. File error → Retry or notify support.
Quality Requirements	A) Deletion or archiving must complete within 2 seconds.
Security Requirements	A) Only Admins can perform deletions or archiving. B) Archived data must be stored in a secure folder.
Postconditions	On success: Record is removed from active patient list. On failure: Error displayed and operation retried.

H. Teleconsultation

Use Case	Teleconsultation
Goal	Enable patients and doctors to interact remotely.
Preconditions	Appointment exists and both users are available.
Success End Condition	Conversation is completed and consultation saved.
Failed End Condition	Session fails or one party does not join.
Primary Actors	Patient, Doctor
Secondary Actors	IT Support
Trigger	Patient selects “Start Consultation”.
Description / Main Success Scenario	1. Patient selects “Start Consultation”. 2. Doctor joins the session (CLI chat interface).

Use Case	Teleconsultation
	3. They communicate regarding the case. 4. Doctor writes note and creates prescription if needed. 5. System stores the chat or marks the session as completed.
Alternative Flows	A) Step 2a. Doctor unavailable → Notify patient and reschedule. B) Step 3a. Technical issue → Retry or suggest in-person visit.
Quality Requirements	A) Session should start within 3 seconds of both parties being online. B) Chat must support real-time text within 1-second delay.
Security Requirements	A) Communication must be encrypted. B) Only authorized participants can join the session.
Postconditions	A) On success: Session marked completed and saved. B) On failure: Error logged and session flagged as incomplete.

I . Billing and Payment

Use Case	Billing and Payment for Appointment Scheduling
Goal	To enable patients or authorized staff to complete billing and payment processes related to scheduled healthcare appointments, ensuring payment is collected before or after services are rendered.
Preconditions	User (patient or staff) must be authenticated and logged into the system. An appointment must be scheduled or in the process of being scheduled. Service and pricing details must be available in the system.
Success End Condition	Payment is successfully processed for the appointment and the billing record is updated. Patient receives confirmation and receipt.

Use Case	Billing and Payment for Appointment Scheduling
Failed End Condition	Payment is not completed or billing fails to generate; appointment may remain unpaid or in pending status.
Primary Actors	Patient, Front Desk Staff
Secondary Actors	Payment Gateway, Insurance Provider System, Billing Department, IT Support
Trigger	Patient or staff initiates appointment scheduling that requires billing or payment confirmation.
Description / Main Success Scenario	Step 1. Patient or staff accesses the Appointment Scheduling module. Step 2. Selects desired service, provider, date, and time. Step 3. System calculates cost of service and displays billing summary. Step 4. Patient selects a payment method (self-pay, insurance, etc.). Step 5. If self-pay, patient proceeds to online payment. Step 6. System processes payment via secure gateway. Step 7. System updates appointment status to "Confirmed" and sends receipt/confirmation. Step 8. Billing record is stored and linked to the appointment.
Alternative Flows	Step 3a. Service pricing not found → Display error; prompt staff to verify service list. Step 4a. Insurance selected → Trigger Use Case 'Submit Insurance Claim' Step 5a. Payment skipped (e.g., due at time of visit) → Mark appointment as "Payment Pending" Step 6a. Payment fails → Display error and allow retry or alternative method
Quality Requirements	Step 3. Cost estimate must be displayed within 2 seconds. Step 6. Payment confirmation must occur within 10 seconds of submission. Step 7. Appointment status update and receipt delivery should complete within 3 seconds.
Security Requirements	All payment transactions must be processed over encrypted connections. Sensitive billing data must be stored securely and access-limited to authorized personnel. System must log all billing activities.
Postconditions	On success: Appointment is confirmed with associated billing record. On failure: Appointment may remain tentative or unconfirmed; issue is logged for follow-up.

7. Gantt Chart

Gantt chart for **Healthcare Management System**

Task / Week	Jun W1	Jun W2	Jun W3	Jun W4	Jul W1	Jul W2	Jul W3	Jul W4	Aug W1	Aug W2	Aug W3	Aug W4
1. Requirement Analysis												
2. System Design												
3. File Structure Planning												
4. Module: User Management												
5. Module: Appointment System												
6. Module: Billing												
7. Admin Controls												
8. Testing & Debugging												
9. Documentation												
10. Final Report & Delivery												

8. Software Requirement Specification

Software Requirements Specification Document of Our HMS

FR001	User Registration
Description	After entering the URL in any internet browser, in the software index page, a new patient must register by providing name, age, gender, contact information, and password. All user data shall be stored in a file (e.g., patients.txt). Registration is mandatory for new patients.
Stakeholder	Patient, Admin

FR002	User Login
Description	After registration, users (patients or staff) must log in using stored credentials. Only verified users can access the system. Credentials are matched from file records before granting access.
Stakeholder	Patient, Staff, Admin

FR003	Book Appointment
Description	After log-in, patients or receptionists can book an appointment by selecting doctor, department, date, and time. Appointment details shall be stored in a file (e.g., appointments.txt).
Stakeholder	Patient, Receptionist, Admin

FR004	Manage Appointment
Description	Patients or receptionists can view, reschedule, or cancel existing appointments. The system updates or

	deletes appointment records in the file as required.
Stakeholder	Patient, Receptionist, Admin

FR005	View and Pay Bill
Description	Patients can view their billing details and process payment (simulated). Payment status shall be updated in the billing file (e.g., billing.txt).
Stakeholder	Patient, Admin

FR006	Conduct Teleconsultation
Description	The system simulates teleconsultation by displaying doctor availability. Consultation notes are recorded and stored in the patient's record.
Stakeholder	Patient, Doctor, Admin

FR007	E-Prescription Generation
Description	After consultation, doctors can enter and save prescription details linked to the patient's file for future reference.
Stakeholder	Doctor, Patient, Admin

FR008	Lab Test Request & Result Upload
Description	Doctors can request lab tests for patients. Lab technicians can upload and store test results linked to the patient ID.
Stakeholder	Doctor, Lab Technician, Patient, Admin

FR009	Pharmacy Module
Description	The pharmacist dispenses medication based on prescription ID. Delivery details are recorded in a file.
Stakeholder	Pharmacist, Patient, Admin

FR010	Medical Record Update
Description	Doctors can update the patient's electronic medical record with history, diagnosis, and other details, stored securely in the system file.
Stakeholder	Doctor, Admin

9. User Profiles

Hospital Admin

Use cases: Manage hospital resources, schedule appointments, oversee staff, track patient records, and generate operational reports.

Characteristic	**Notes on Characteristic**	**Implied Requirement**
Type of User	Hospital administrator	Role-based access with administrative privileges.
Age Range	35–60 years	Clear, professional UI with emphasis on data visibility and readability.
Frequency of Use	Daily (full-time role)	Real-time dashboard with updates and alerts.
Mandatory/Discretionary	Mandatory (critical to operations)	High system reliability, minimal downtime.
Computer Experience	Intermediate to advanced	Advanced features for staff, appointments, and reporting management.

Characteristic	Notes on Characteristic	Implied Requirement
Education	Bachelor's degree or higher (e.g., healthcare administration)	Use of industry-specific terms; interface can assume familiarity with healthcare terms.
Goals	Ensure smooth hospital operations, reduce wait times, monitor KPIs	Tools for scheduling, reporting, patient tracking, and analytics.
Language Skills	Professional English; may require local language support	Option to customize or localize system language.
Number of Users	Medium (multiple admins, department heads)	Multi-user access with customizable permission levels.
Training	Required (formal system training necessary)	Comprehensive onboarding, manuals, and help resources.
Ways of Working	Collaborative with medical/support staff	Integrated modules across departments (e.g., scheduling, HR, patient records, billing).

Doctor

Use cases: View patient history, update medical records, manage appointments, prescribe medication, and communicate with other departments.

Characteristic	Notes on Characteristic	Implied Requirement
Type of User	Licensed medical professional	Secure and fast access to accurate patient records.
Age Range	30–65 years	Interface should accommodate both younger and older professionals.
Frequency of Use	Daily (during working hours)	Minimal clicks to access patient history and lab results.
Mandatory/Discretionary	Mandatory	Streamlined interface to support uninterrupted workflow.
Computer Experience	Moderate to high	Support for tools like ICD code search and e-prescriptions.

Characteristic	Notes on Characteristic	Implied Requirement
Education	Advanced medical degree (e.g., MBBS, MD)	Integration of medical terminology and diagnostic tools.
Goals	Efficient diagnosis and treatment tracking	Fast-loading charts; integration with lab and imaging systems.
Language Skills	Professional English; possibly multilingual	Clear documentation with optional translation support.
Number of Users	High	Real-time synchronization and reliable data updates.
Training	Basic onboarding needed	Short tutorials focused on EMR usage and prescription handling.
Ways of Working	Independent but collaborates with nurses and lab teams	Role-based notifications and secure interdepartmental data sharing.

Patient

Use cases: Schedule appointments, view prescriptions and reports, make payments, track treatment history, and receive notifications.

Characteristic	Notes on Characteristic	Implied Requirement
Type of User	General outpatient user	Clean, mobile-friendly portal with appointment booking.
Age Range	18–70 years	Adjustable font sizes and accessibility features (e.g., screen reader support).
Frequency of Use	Occasional (based on treatment needs)	Easy login, appointment scheduling, and reminder features.
Mandatory/Discretionary	Discretionary (use encouraged by hospital)	Incentivize portal use with value additions (e.g., easy access to lab reports).
Computer Experience	Ranges from basic to moderate	Intuitive design with simple icons and clear navigation.
Education	Varies widely	Avoid technical jargon; communicate using plain, clear language.

Characteristic	Notes on Characteristic	Implied Requirement
Goals	Manage appointments, view reports, track health info	One-click access to prescriptions, test results, and visit summaries.
Language Skills	Local language; some users understand English	Multilingual or localized interface support.
Number of Users	Very high	Scalable system with load balancing and high user concurrency.
Training	None	Visual onboarding, FAQs, and guided tips for first-time users.
Ways of Working	Independent	24/7 mobile and desktop access to personal health dashboard.

Medical Staff (Receptionist)

Characteristics	Notes on characteristics	Requirement implied
Type of user	Frontline hospital staff (receptionist)	Easy-to-use interface for multitasking under pressure.
Age range	22–50 years	Touchscreen-friendly UI for quick data input.
Frequency of use	Daily	Always-on systems with auto-save to avoid data loss.
Mandatory/discretionary	Mandatory	Role-based permissions with quick navigation.
Computer experience	Basic to intermediate	Pre-set forms and dropdowns to minimize manual entry.
Education	Nursing diploma or basic administrative training	Visual indicators for patient statuses.
Goals	Keep records accurate, ensure patient flow	User alerts for pending tasks or status updates.
Language skills	Local language, some English	Multi-language interface or labeling.

Characteristics	Notes on characteristics	Requirement implied
Number of users	Medium to high (all departments)	Simultaneous access from multiple devices and terminals.
Training	Required (short, hands-on training)	In-system tooltips and support chat.
Ways of working	Collaborative with doctors and patients	Task assignment tools and alert dashboards.

10. Requirement Elicitation Technique

Technique	Purpose	Stakeholders Involved
Interview	Understand workflows	Doctors, Admins
Questionnaire	Collect user expectations	Patients, Staff
Observation	Discover unspoken requirements	All
Document Analysis	Identify current processes	Admin
Brainstorming	Decide features to implement	Developer Team (Rayan, Bappi & Tanvir)

11. Conclusion

This proposed “Healthcare Management System” balances ambitious functionality with the practical constraints for our basic capstone project. By choosing the C programming language, me and my team are showcasing low-level design mastery while delivering a feature rich application. The menu driven approach, modular design, security first mindset and clear development map helps us foresee the success completion and real world relevance of our capstone project. Me and my team are hoping that we are going to improve it more and more by sequentially testing and maintaining our project.