American International University-Bangladesh (AIUB)

Department of Computer Science Faculty of Science & Technology (FST)

Hospital Management System

A Software Engineering Project Submitted By

Semester: Summer 24-25		Section: G	Group Number: 7	
SL	**Student Name**	**Student ID**	**Contribution (CO3 + CO4)**	**Individual Marks**
1	RASHED AHAMMOD BIN AZAM	23-51203-1		
2	ISRAT JAHAN CHOWDHURY	23-51219-1		
3	ASIFUR RAHMAN	23-51204-1		
4	NABIL MOHAMMED NASIM UDDIN	23-51206-1		
5	MARIYA MEHRIN	23-54092-3		

The project will be evaluated for the following Course Outcomes

CO3 (PO-g-1) ***Select appropriate software engineering models, project management roles and their associated skills for the complex software engineering project and evaluate the sustainability of developed software, taking into consideration the societal and environmental aspects***	Total Marks	
Selection of Software Engineering Models: Process model selection and presents sufficient evidence to support argument for the model selection	[5 Marks]	
Role identification and Responsibility Allocation: Well-planned project with proper role identification and responsibility allocation in the project management activities	[5Marks]	
Formatting and Submission: Submission, Defense, Completeness, Spelling, grammar, and Organization of the Project report	[5Marks]	
CO4 (PO-k-1) ***Apply engineering management principles and economic decision making to develop software engineering project management plan.***	Total Marks	
Project WBS and Testcases: Relevant WBS (project task list) and testcases for the proposed project are stated properly.	[5Marks]	
Effort Estimation and Scheduling: Project estimation was described using proper effort estimation or schedules based on available project resources	[5Marks]	
Risk Management: Sufficient and appropriate risks are identified, analyzed, and properly categorized or prioritized.	[5Marks]	

1. PROJECT PROPOSAL

1.1 Background to the Problem

A mid-sized public hospital has requested the development of a Hospital Management System to improve patient care and streamline hospital operations. At present, the hospital relies largely on manual processes and paper records for managing patients, departments, staff, and facilities. This approach has led to inefficiencies, delays in communication, and difficulties in tracking medical and administrative information. The hospital now requires a centralized system to better organize its operations, reduce paperwork, and ensure smoother coordination across all departments. Several roles have been identified for the system, each with specific responsibilities.

The Admin role is expected to oversee the hospital's operations at a high level. The Admin should be able to monitor information such as the number of users, departments, wards, rooms, medicines, and other facilities. It has been requested that the Admin should also have the authority to assign responsibilities to management staff, approve doctors and nurses for service, and handle complaints that may be raised by any user.

The Management staff should be able to handle admitted patients, including validating admissions through doctor referrals or patient applications. It has been requested that Management staff have well-defined responsibilities, so that some users may handle financial matters such as payments and transactions, while others may focus on areas like medicine stock or room allocation.

The Doctor role should allow doctors to manage their interactions with patients efficiently. Doctors are expected to be able to keep track of appointments, update the status of visits, issue prescriptions, and access patient histories to ensure continuity of care.

The Patient role should provide patients with self-service features. Patients should be able to find and book appointments with doctors, cancel appointments when needed, apply for admission to the hospital, view their medical history, and raise complaints if necessary. Patients should also have the ability to order medicines, book medical tests, and settle their hospital bills.

For those who are admitted, the role of Admitted Patient will provide additional access. These patients should be able to make use of different hospital facilities during their stay. Their care should be more closely managed by nurses and doctors, ensuring proper monitoring and treatment.

The Nurse role should allow nurses to support admitted patients effectively. Nurses should be able to view patient histories, record treatment notes, and manage patient releases once approved by doctors. Each nurse should be responsible for patients assigned to them based on ward allocation, and their working hours should also be taken into account.

By implementing this system, the hospital expects to modernize its operations, improve efficiency, strengthen communication among staff, and enhance the overall patient experience. The system is expected to help reduce paperwork, centralize hospital data, and provide both patients and staff with a smoother, more reliable, and well-coordinated experience.

1.2 Solution to the Problem and Process Model Selection

Users:

1. Admin
2. Management
3. Doctor
4. Nurse
5. Patient

Scopes:

1) Profile Management

Features:

a) All users can create their profiles with valid and complete information.
b) All users can log in using a valid username and password.
c) After successful login, each user can access their personalized dashboard.
d) All users can view their own profile details.
e) All users can update their own information with valid entries.
f) All users can change their password.
g) All users can recover their password after verification.

2) User Management

Features:

a) Admin can create, edit, update, and delete user profiles.
b) Admin can activate or deactivate user accounts (inactive accounts cannot log in but remain stored).
c) Admin can assign Management roles and Management subcategories.
d) Admin can approve doctor and nurse profiles.
e) Admin can view all user lists filtered by role.
f) Admin can manage department, ward, and room data.

3) Appointment Management

Features:

a) Patients can book and cancel appointments with available doctors.
b) Doctors can view and update the status of appointments (approved, rejected, completed).
c) Appointment history can be viewed by both doctors and patients.
d) Appointment status is updated and visible in real-time to both parties.
e) Notifications are sent to users regarding appointment status updates.

4) Admission Management

Features:

a) Patients can fill up admission request forms.
b) Doctors can refer patients for admission.
c) Management validates admissions based on doctor referrals or patient applications.
d) Management assigns rooms and wards to admitted patients.
e) Nurses are assigned to admitted patients ward-wise.
f) Admin and Management can monitor admission statuses.

5) Medical History & Prescription Management

Features:

a) Doctors can view patient history and assign prescriptions.
b) Nurses can view and update medical status of admitted patients.
c) Patients can view their own medical records and prescriptions.
d) Admitted patient history is updated regularly by nurses during hospital stay.
e) Doctors can discharge patients; nurses can process discharge after approval.

6) Complaint Management

Features:

a) Patients, doctors, nurses, and management users can submit complaints.
b) Admin can view and resolve complaints from all user types.
c) Complaint status is updated and notified to the user.
d) Complaint history can be accessed by both Admin and the complaint submitter.

7) Medicine and Facility Management

Features:

a) Management (if assigned) can add, update, or remove medicines from stock.
b) Medicine quantity and expiry details can be updated.
c) Admin and Management can view full medicine and facility lists.
d) Patients can view and order available medicines.
e) Admitted patients can rent available hospital facilities.
f) Facility availability status is updated in real time.

8) Billing and Test Management

Features:

a) Patients can pay hospital bills through their dashboard.
b) Test booking can be done by patients.
c) Management (if assigned) can manage transactions and verify payments.
d) Payment history and invoices are accessible by both patients and management.

9) Report and Dashboard Access

Features:

a) Admin can view total users, departments, wards, rooms, medicine stock, etc.
b) Role-based dashboards display relevant statistics to each user.

USER STORY BOARD

Table-1

As a/an	I want to	So that	Acceptance criteria
Patient	Create account	I can securely access healthcare services like booking appointments and viewing my medical history	Unique credentials and verification
	Login to the system	I can access my personal health information and manage my interactions with healthcare facilities.	Valid username and password
	View Dashboard	I can see all the actions that I can perform	After successful login
	Schedule Appointment	I can receive medical consultation at my preferred time	If the slot is available
	View Prescription/Test Results	I can understand my treatment plan and track my health progress	Results visible only after doctor uploads and patient logs in.
	Order Medication	I can adhere to my treatment regimen	Medicines are ordered only with valid prescription.
	Send/Get notification	I can stay informed about appointments, test results, medicine orders, and important updates	Patients get alerts only for valid events.
	View Doctors Profile	I can choose the right for my medical needs	Only when patients logged in

	Cancel Appointment	I can free up the slot if I'm unable to attend	Appointment can be canceled only before the scheduled time
	Get Admitted by filling up Admission form	I can get treatment from the hospital when admittance is required	If required fields are filled
	Edit and Update Profile	**I can** keep my contact details and medical information accurate and current.	Profile updates are saved only after login.
	File Complaints	I can report issues or provide feedback about the service	Allowed only for logged-in patients
	Pay Bill	I can settle my accounts with the healthcare facility	Bill paid only for own account with receipt.
	View History	I can keep track of my past treatments, appointments and diagnosis	Patients see only their own records in order
	Rent Facilities	I can have access to necessary amenities During my stay	If the requested resource is available and payment is completed
Doctor	Create Account on System	I can get a unique login, ensuring secure access to their own dashboard and patient data.	Users have a valid email or phone number.
	Login into the System	I can access my user interface and can do perform.	Only registered users can log in using valid credentials.
	View Dashboard	I can easily access all the information I need without switching between modules, by having everything visible on one screen	The user can access the dashboard immediately after login.

	View Profile	I can access my personal details anytime	All the information displayed is accurate and up to date
	Managing profile information	I can update my profile information on system	User has to input valid information
	Send Notification to Patient	I can handle essential communication directly and efficiently instead of relying on nurses or staff	The target user has an active account.
	Get Notification from Management	I can receive immediate alerts about changes in protocols, duty schedules, administrative policies, or emergency protocols.	The user has an active account.
	View Appointment List	I can see upcoming patient names and concerns that allow me to review medical history or test results in advance.	The system must display all upcoming and past appointments.
	Update Appointment Status	I can update the status of what patients perform on appointment days like completed or absent.	The user can change status immediately
	Assign Prescription	I can enter prescriptions digitally, eliminating issues related to illegible handwriting or manual errors.	The patient can view the assigned prescription.
	View Patient History	I can access past medication history during follow-ups.	The history shown belongs only to the selected patient.
	Access to View Patient Report	I can see an online copy of the patient report	The system displays the report with accurate and complete details.

	Feedback from Patient	I can review the outcome from the patient	The submitted feedback is visible to authorized users
	Send Leave application to management	I can view the status of the leave application (approved, pending, rejected).	The application is saved and linked to the correct staff/doctor profile.
Nurse	Create account into the system	I can access the system with my own secure login credentials.	Users must have a valid email to register.
	Login into the system	I can access my dashboard and perform my duties.	The user can log in successfully and reach their dashboard.
	View my dashboard	I can quickly see important updates, tasks, and patient info.	Dashboard displays important updates, tasks, and patient information.
	Release Admitted Patients	I can follow the Doctor's instructions to release a patient after he gives permission.	The user can release patients only after receiving doctor approval.
	Request for medicine refill	I can alert the Management that the ward has run out of certain medicines.	The system allows the user to send medicine refill requests to management
	Update Patient History	Tests or medications provided to patients are added to the history and updated	Tests and medications provided to patients are recorded and updated in their history.

	Get notification from Management	1. I know my designated ward and get assigned patients to take care 2. I can get immediate alerts about changes in protocols and duties	The user receives alerts about ward assignments and protocol changes.
	View Patient History	I can evaluate the patient's condition and provide accurate care	The user can access patient history to evaluate conditions accurately.
	Send a leave application to management	I can request time off officially and track approval.	The user can submit leave requests and track approval status.
	View the list of assigned admitted patients	I know which patients I am responsible for.	The user can see all patients assigned to their care.
	Track and follow my assigned ward schedule	I can manage my duties according to shifts and locations.	The user can view and manage duties based on shifts and ward locations.
	Send and receive notifications	I can stay updated and communicate effectively with the team.	The user can communicate and receive updates from the team in real time.

	Generate daily report for admitted patients	I can provide updates on patient status and activities each day.	The system allows the user to create and submit daily patient reports.
Admin	Login to the System	I can access all users' details and my scheduled activities	Valid user id and password then view dashboard
	Create Users' Profile	I can manually create personnel files from the admin end.	With proper verification
	View Users Profile	I can learn about users' identity, appointments, allocated doctors, and other details.	can View users all details
	Update Users Profile	I can control and update details manually	Changed all information and added to profile after update
	Delete Users Profile	I can manage profiles by removing inactive users or other unwanted profiles.	Delete User`s account from the system
	Manage and Access All Facilities	I have a clear idea about what the system contains, or I can edit what the Hospital offers and has.	Only available facilities can get access
	Send Notification to Users	Users can check all updates.	Users are notified with details after end of their activities.
	View Dashboard	I can see all the functionality of the system.	Visible after logging in to the account
	Handle Complains from Users	I can get feedback and resolve issues and handle the problem	Valid objections are the main concern from user
	View Reports	I can assess and get notified of the status of the Hospital.	After visiting doctor, user reports are available to view

	Override User Controls	I can manually control and override certain restrictions in the system for emergencies. (Force unlock accounts, reassign users, etc.)	Restrictions and rules are applied in the system for all types of users for security purpose.
	Manually create and assign Management accounts	Only authorized personnel can access Management accounts.	Only the designated person can assign if need
Management	Login to the system	I can access my user interface	Valid username and password required
	View Dashboard	I can easily access all the information I need without switching between modules, by having everything visible on one screen	Dashboard loads after login.
	Handle Inventory	I can update the required products' stock.	Only valid medicines from the catalog
	Handle Admission for patients	I can register and admit patients to the Hospital after validating them.	Admission requests visible. Management can approve or reject. On approval, patient status changes to "Admitted."
	Handle transaction and payments	I can clear patient's dues.	Payments linked only to the correct patient account. Valid invoices generated after payment confirmation.
	Send internal announcements	Doctors and nurses are updated about protocols or shifts.	Title and content must be entered. Recipients must be selected (Doctor, Nurse, or both). Announcements appear in recipients' dashboards and notifications.

	Assign and manage hospital facilities	Admitted patients can access necessary hospital amenities during their stay	Only available facilities can be assigned.
	Handle Complaints	I can view complaints filed by patients or staff and report them back to the Admin.	Complaint details visible. Management can add resolution notes and change status.
	Allocate Nurses their designations	Nurses get updated of their designations and respective wards	Nurses visible in allocation list with availability. Each nurse must be linked to a specific ward/room. Notifications sent to assigned nurses about their ward.
	Request Leave	I can get permission to get a leave	Leave request form requires valid dates and reason. Status of leave shown as Pending, Approved, or Rejected.
	Approve/Reject Leave Requests	leave requests of staff are documented and follows a procedure	Leave request form requires valid dates and reason. Status of leave shown as Pending, Approved, or Rejected.
	Oversee and approve patient test bookings	Lab resources are efficiently scheduled, and double booking is prevented	Test bookings visible with patient details, test type, date/time. Management can approve or reschedule with a new slot. Patients are notified immediately of approval or changes.

Fig-1: Shows User Story Board made using Trello.

Available existing software solutions:

Several hospital management and electronic medical record (EMR) systems already exist in the market, such as OpenMRS, an open-source platform for patient records and hospital workflows; MediTech, a commercial solution integrating billing, laboratory, and scheduling functions; SAP Healthcare Solutions, an enterprise-level system offering end-to-end hospital resource management; CareCloud, a cloud-based platform focused on patient management and billing; and HospitalRun, an open-source system designed for small and mid-sized hospitals with offline capability. While these solutions address many hospital needs, they are often either too costly, too complex, or require significant customization and technical expertise. As a result, they may not align with the requirements of a mid-sized public hospital that has not yet adopted software and needs a system that is both affordable and tailored to its specific operational needs.

Software Development Process Model:

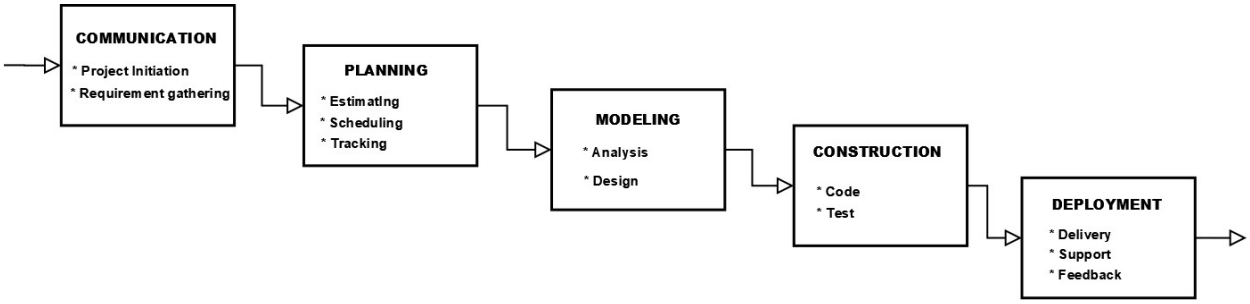

Fig-2: Waterfall Model

The Waterfall model was chosen for this project because it best aligns with the nature of the hospital's requirements and constraints. The client is a mid-sized public hospital that has not yet adopted a software solution, and its processes are already well-established and stable. Since the requirements are fixed and clearly defined from the beginning, a linear, phase-by-phase model ensures that these needs are addressed systematically without the risk of frequent changes or scope creep. The hospital also operates with limited resources and budget, which makes it important to follow a structured plan with predictable timelines and costs.

Waterfall's emphasis on thorough documentation at every stage makes it especially suitable for healthcare projects where compliance, accountability, and clarity are crucial. Each phase, from requirement gathering to planning, design, implementation, and testing, produces clear deliverables that can be reviewed and approved before moving forward. This reduces the chances of misunderstanding and ensures the hospital management has full visibility into the project's progress. Furthermore, testing is carried out once the entire system is built, allowing the solution to be evaluated as a complete and integrated product, which is critical when dealing with sensitive operations such as patient records, medicine stock, admissions, and billing.

In summary, the Waterfall model was chosen because it offers stability, predictability, strong documentation, and low risk, all of which are vital for delivering a reliable hospital management system. Its structured approach guarantees that the system is secure, accurate, and aligned with the hospital's operational needs, while remaining cost-effective and manageable within the hospital's resources.

Analysis of the Project Environment

The environment for this project is a mid-sized public hospital that has traditionally relied on manual processes for managing patient records, admissions, billing, and resource allocation. The hospital's workflows are already well-established and standardized, meaning the requirements for the new Hospital Management System are stable, clearly defined, and not expected to change frequently. The hospital management team has a clear understanding of what they need the system to achieve such as managing admissions, tracking medicine stock, handling billing, and improving communication between doctors, nurses, and patients, and these requirements have been agreed upon during the initial consultations.

Because the hospital operates within a fixed budget and limited technical capacity, there is little room for ongoing requirement changes or iterative experimentation. Instead, the hospital expects a solution that is reliable, cost-effective, and aligned with its current operations. This makes the project environment relatively predictable compared to projects in industries where client needs evolve continuously. The main focus is therefore on translating the hospital's established processes into a digital platform rather than redefining how the hospital operates.

In summary, the project environment is characterized by stable requirements, minimal expected changes, and a need for structured, step-by-step development that ensures accuracy, compliance, and efficiency in hospital operations.

Alignment of Model with Team Size, Communication, and Feasibility

The Waterfall model matches well with how our team is organized. Our team has five members, and each person is responsible for one role of the system (Admin, Management, Doctor, Patient, Nurse). Since the work is divided clearly, the step-by-step nature of Waterfall makes it easier for each member to focus on their own part without confusion. The model also depends on proper documentation at every stage, so instead of constant discussions, the team can rely on written plans and reports. This makes communication more straightforward and helps everyone stay on the same page.

For task coordination, Waterfall allows us to complete one stage fully before moving on to the next. For example, once the requirements are confirmed, the design can be finished, and only then does the development begin. This makes it clear what comes next and avoids mixing tasks. Each team member knows their responsibility, and progress can be tracked step by step. At the end, all parts are combined and tested together, which is important for a hospital system where everything must work correctly and reliably.

This solution is also practical for meeting the hospital's goals. The hospital's needs are already clear and not likely to change, which makes Waterfall suitable. The model ensures that the project stays within the limited budget, is delivered on time, and meets the hospital's main objectives: reducing paperwork, organizing data in one place, and improving how patients and staff interact. Overall, Waterfall fits our small team, supports communication and task management, and provides a realistic way to build a working system for the hospital.

Flexibility of the Model

The Waterfall model is generally not very flexible when it comes to adapting to changes in scope, technology, or user requirements once the project has started. Since it follows a strict step-by-step sequence, each phase must be completed before moving to the next. If major changes are introduced after the requirements or design phase, it would require going back and redoing earlier work, which can be costly and time-consuming. This makes Waterfall less adaptable compared to

iterative or Agile approaches. However, in our project context, the hospital's requirements are already stable, well-defined, and unlikely to change, so this limitation is less of a concern. On the positive side, the strong documentation and clear structure of Waterfall ensure that if small adjustments are needed, they can still be handled with proper planning and by updating relevant documents. In summary, while the model is not highly flexible, it is appropriate for this project because the scope and requirements are fixed, the technology is straightforward, and the hospital does not anticipate frequent changes.

Creative Solution to the Real-Life Problem

The main problem of the hospital is that most tasks like admissions, billing, medicine stock, and staff communication are still handled manually, which causes delays and mistakes. Our solution is to build a Hospital Management System (HMS) that brings all these tasks into one easy-to-use digital platform with role-based access. Each user will only see the features they need: Admins can oversee operations and handle approvals, Management can admit patients and allocate facilities or medicines, Doctors can manage appointments and prescriptions, Nurses can update patient records and request medicine refills, and Patients can book appointments, order medicines, pay bills, and view their history. Extra features such as digital facility allocation, nurse refill requests, and internal announcements for protocol changes make the system practical and efficient for real hospital use. This design reduces paperwork, speeds up decision-making, and improves patient service while keeping costs low, making it a creative yet realistic solution for a mid-sized public hospital.

Target Users and Benefits

The target users of the Hospital Management System are the staff and patients of the hospital, each with specific roles and needs. The Admin will benefit by having full visibility and control over hospital operations, being able to monitor resources, assign roles, and handle complaints efficiently. The Management staff will benefit from tools to validate admissions, allocate facilities, manage medicine stock, and send announcements, helping them organize resources more effectively. Doctors will gain easy access to patient histories, appointment schedules, and prescription tools, which improves treatment efficiency and record-keeping. Nurses will benefit by having clear ward assignments, the ability to update patient histories, request medicine refills, and manage patient releases with doctor approval, making their work faster and better coordinated. Patients will benefit from self-service features such as booking or cancelling appointments, ordering medicines, paying bills, and accessing their medical history without long waits, which improves convenience and satisfaction. For admitted patients, the solution will provide additional benefits like facility requests and closer monitoring. Overall, the system will benefit all user groups by reducing paperwork, improving communication, saving time, and ensuring better quality of care.

Contribution to Scientific Results

This project contributes to the development of scientific results by showing how a structured software engineering approach can be applied to solve a real-world problem in healthcare. The system design, requirements analysis, use case modeling, and testing are all carried out in a systematic and documented way, which makes the results reliable and reproducible. Each step of the project, from the problem statement to the model selection, class diagrams, activity diagrams, data flow diagrams, and test cases, has been clearly identified and recorded. This ensures that the project not only provides a working solution for the hospital but also serves as a well-documented

case study on how to design and plan an information system in a healthcare setting. The detailed documentation allows others to learn from the process, adapt the methods to similar projects, and build on the results in future research or practical implementations. In this way, the project contributes to both practical problem-solving and the academic understanding of applying software engineering principles in healthcare.

Managing Risks and Uncertainties

In the Waterfall model, project risks and uncertainties are managed through its structured, step-by-step phases. During the requirements stage, risks of misunderstanding the client's needs are reduced because all requirements are gathered, documented, and approved before moving forward. In the design stage, risks related to system structure or data handling are addressed early by preparing detailed diagrams and design documents, which lowers the chance of major issues appearing later. In the implementation stage, risks of coding errors are reduced because developers follow the approved design closely. In the testing stage, risks of failure in critical areas such as admissions, billing, or patient records are identified and corrected before deployment. Finally, in the deployment and maintenance stage, the risk of disruption is minimized by only releasing the system once it has passed all tests. Although the Waterfall model is less flexible for changing requirements, it manages risks by ensuring that every stage is completed, reviewed, and documented before moving to the next, which is especially important in a hospital setting where accuracy and reliability are critical.

Justification of Chosen Model

The Waterfall model is more suitable for this project than other alternatives because of the nature of the hospital's environment and requirements. The client is a mid-sized public hospital with well-defined processes that have remained stable over time, and the requirements for the system are clear, complete, and not expected to change. This makes a linear, phase-based model like Waterfall the most efficient choice.

Other traditional software process models were considered, but they were not chosen as they do not align as well with the project environment. The V-Model, while similar to Waterfall, places strong emphasis on verification and validation at every stage through parallel testing. Although this increases reliability, it also adds significant overhead, cost, and time, which would not be practical for a mid-sized public hospital with limited budget and resources. The Incremental Model was also reviewed, but it is better suited to projects where delivery of partial systems is required early and requirements may evolve over time. In our case, the hospital's requirements are stable, and the system is expected to be delivered as a complete product, not in separate increments. because it is most appropriate for large-scale, high-budget projects with high levels of uncertainty. Compared to these models, Waterfall is simpler, more cost-effective, and provides the clarity and documentation needed to deliver a reliable hospital management system in this environment.

Agile models such as Scrum or Iterative Development were also evaluated but found unsuitable. Agile works best in projects where requirements are flexible and evolve through ongoing client feedback. In this case, the hospital's requirements are fixed from the start, and the client does not have the resources or technical expertise to participate in frequent sprint reviews or continuous requirement refinements. Agile also requires highly collaborative teams and constant adaptability, which can increase costs and introduce instability, factors that conflict with the hospital's need for a cost-effective, predictable, and straightforward system.

1.3 Project Role Identification and Responsibilities

Main Roles Involved

The project requires a combination of management roles to oversee planning and progress, and development roles to execute technical tasks. The main roles are:

- **Project Manager**
- **System Analyst**
- **Designer (Software Architect)**
- **Developers (Programmers)**
- **Tester (Quality Assurance)**
- **Documentation & Configuration Manager**

Responsibilities in Key Stages

During the requirements gathering stage, the System Analyst leads by collecting the hospital's needs through discussions with the client and preparing documents such as the problem statement, scope, and use cases, while the Project Manager ensures that the requirements are realistic within the given time and budget, and the Documentation Manager records them properly.

In the design stage, the Designer prepares class diagrams, data flow diagrams, activity diagrams, and the system architecture, supported by the System Analyst who validates the design against requirements, while the Project Manager monitors progress.

In the implementation stage, the Developers build the assigned features for roles such as Admin, Management, Doctor, Patient, and Nurse, the Configuration Manager maintains version control using GitHub, and the Project Manager coordinates progress and resolves dependencies.

During the testing stage, the Tester executes the prepared test cases, verifies the results, and reports defects, while Developers fix the identified issues and the Project Manager ensures problems are resolved.

Finally, in the deployment stage, the Project Manager coordinates integration and release, the Documentation Manager prepares user manuals and training materials, and the Tester verifies that the deployed system fully meets the requirements.

Team Distribution and Justification

Responsibilities are divided step by step according to each phase. At the start, business analysts and system analysts talk with the client to collect and document all requirements, while the project manager checks that these are realistic for the time and budget. In the design stage, software architects and designers prepare system diagrams, database structures, and blueprints, with the project manager making sure the design matches the requirements. During implementation, developers write the actual code while configuration managers take care of version control. In the testing stage, quality assurance engineers run test cases to find problems, developers fix them, and the project manager ensures everything works properly. Finally, in the deployment stage, system administrators install the system, documentation specialists prepare user guides, and the project manager delivers the system to the client. This clear division of tasks makes sure that each part of the project is done by the right people and that work flows smoothly from one stage to the next.

2. SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS) / PRODUCT REQUIREMENTS DOCUMENT (PRD)

2.1 Functional Requirements

Table-2: Admin Functional Requirements

1.0	User home page-All features availability
1.1	**Registration**
1.1.1	Taking name
1.1.2	Taking email address
1.1.3	Taking username
1.1.4	Taking password
1.1.5	Taking confirm password
1.1.6	Taking gender
1.1.7	Taking phone number
1.1.8	Taking date of birth
1.1.9	Submitting the form
1.1.10	Resetting the form
1.2	**Login**
1.2.1	Taking user name
1.2.2	Taking password
1.2.3	Providing forgot password option
1.2.3.1	Submitting e-mail address and getting user info through our e-mail id
1.2.4	Submission of the login
1.3	**Dashboard**
1.3.1	Dashboard Page
1.3.1.1	Dashboard button [ref 1.3]

1.3.1.2	View profile button
1.3.1.2.1	Show name
1.3.1.2.2	Show e-mail
1.3.1.2.3	Show gender
1.3.1.2.4	Show phone number
1.3.1.2.5	Show date of birth
1.3.1.2.6	Show role (Admin)
1.3.1.2.7	Edit profile button
1.3.1.2.7.1	Show existing name and taking name
1.3.1.2.7.2	Show existing e-mail and taking e-mail
1.3.1.2.7.3	Show existing gender and taking gender
1.3.1.2.7.4	Show existing phone number and taking phone number
1.3.1.2.7.5	Show existing date of birth and taking date of birth
1.3.1.2.7.6	Submission of the updated info
1.3.1.2.7.7	View profile [ref 1.3.1.2]
1.3.1.2.8	Change profile picture button
1.3.1.2.8.1	Taking profile picture
1.3.1.2.8.2	Submission of the updated profile picture
1.3.1.2.8.3	View profile [ref 1.3.1.2]
1.3.1.3	Edit profile button [ref 1.3.1.2.7]
1.3.1.4	Change profile picture button [ref 1.3.1.2.8]
1.3.1.5	Change password button
1.3.1.5.1	Taking current password
1.3.1.5.2	Taking new password
1.3.1.5.3	Taking retype new password

1.3.1.5.4	Submitting the form
1.3.2	**User Management**
1.3.2.1	Create User Button
1.3.2.1.1	Take full name input
1.3.2.1.2	Take email input
1.3.2.1.3	Take username input
1.3.2.1.4	Take password input
1.3.2.1.5	Select role from dropdown (Management, Doctor, Nurse, Patient)
1.3.2.1.6	Assign department from dropdown (if applicable)
1.3.2.1.7	Take phone number input
1.3.2.1.8	Set account status (Active/Inactive)
1.3.2.1.9	Submit user creation
1.3.2.1.10	View list of users [ref 1.3.2.2]
1.3.2.2	**View Users List**
1.3.2.2.1	Show table of users with columns: Name, Role, Department, Status
1.3.2.2.2	Search bar to filter by name or role
1.3.2.2.3	Filter dropdown for account status
1.3.2.2.4	Edit user button [ref 1.3.2.3]
1.3.2.2.5	Delete user button [ref 1.3.2.4]
1.3.2.3	**Edit User Button**
1.3.2.3.1	Show existing name and take updated name input
1.3.2.3.2	Show existing email and take updated email input
1.3.2.3.3	Show existing role and allow role change (if applicable)
1.3.2.3.4	Show existing department and allow reassignment
1.3.2.3.5	Show existing status and allow change

1.3.2.3.6	Submit changes
1.3.2.3.7	View users list [ref 1.3.2.2]
1.3.2.4	**Delete User Button**
1.3.2.4.1	Confirmation prompt “Are you sure?”
1.3.2.4.2	If confirmed, remove user from system
1.3.2.4.3	View updated users list [ref 1.3.2.2]
1.3.3	**Management Role Assignment**
1.3.3.1	Assign Management Role Button
1.3.3.1.1	Select existing users from list [ref 1.3.2.2]
1.3.3.1.2	Choose sub-role from dropdown (Facility Allocation, Test Booking Oversight, Transactions)
1.3.3.1.3	Submit assignment
1.3.3.1.4	View updated users list [ref 1.3.2.2]
1.3.4	**Approvals**
1.3.4.1	Approve Doctor Button
1.3.4.1.1	View pending doctor requests
1.3.4.1.2	Click approve
1.3.4.1.3	Status changes to Active
1.3.4.1.4	View updated doctor list
1.3.4.2	Approve Nurse Button
1.3.4.2.1	View pending nurse requests
1.3.4.2.2	Click approve
1.3.4.2.3	Status changes to Active
1.3.4.2.4	View updated nurse list
1.3.5	**Complaint Handling**
1.3.5.1	View Complaints Button

1.3.5.1	View Complaints Button
1.3.5.1.1	Show list of complaints with columns: ID, User, Date, Status
1.3.5.1.2	Click complaint ID to view details
1.3.5.1.3	Add resolution notes
1.3.5.1.4	Mark as resolved
1.3.5.1.5	View updated complaints list
1.3.6	**Facility Management**
1.3.6.1	View Facilities Button
1.3.6.1.1	Show list of facilities with type, availability, status
1.3.6.1.2	Edit facility button [ref 1.3.6.2]
1.3.6.1.3	Add facility button [ref 1.3.6.3]
1.3.6.2	Edit Facility Button
1.3.6.2.1	Show existing details and take updated inputs
1.3.6.2.2	Submit changes
1.3.6.2.3	View facilities list [ref 1.3.6.1]
1.3.6.3	Add Facility Button
1.3.6.3.1	Take facility type input
1.3.6.3.2	Set capacity and availability
1.3.6.3.3	Submit facility creation
1.3.6.3.4	View facilities list [ref 1.3.6.1]
1.3.7	Logout Button
1.3.7.1	Show Login page [ref 1.0]

Table-3: Management Functional Requirements

1.0	User home page-All features availability
1.1	**Login**
1.1.1	Taking Username
1.1.3	Providing forgot password option
1.1.3.1	Submitting e-mail address and getting user info through our e-mail id [ref 1.1.3].
1.1.3.2	Provide a password reset link via email.
1.1.3.3	Reset password and return to login page.
1.1.4	Submission of the "Login".
1.1.5	System authenticates credentials and redirects to the Management Dashboard.
1.2	**Dashboard**
1.2.1	View profile button
1.2.1.1	Show name
1.2.1.2	Show e-mail
1.2.1.3	Show gender
1.2.1.4	Show designation
1.2.1.5	Show phone number
1.2.1.6	Show date of birth
1.2.1.7	Edit profile button
1.2.1.7.1	Show existing name and taking name
1.2.1.7.2	Show existing e-mail and taking e-mail
1.2.1.7.3	Show existing gender and taking gender
1.2.1.7.4	Show existing phone number and taking phone number
1.2.1.7.5	Show existing date of birth and taking date of birth
1.2.1.7.6	Submission of the updated info

1.2.1.7.7	View profile [ref 1.2.1]
1.2.1.8	Edit profile button [ref 1.2.1.7]
1.2.1.9	Change password button
1.2.1.9.1	Taking current password
1.2.1.9.2	Taking new password
1.2.1.9.3	Taking retype new password
1.2.1.9.4	Submitting the form
1.2.2	**Admitted Patient Management**
1.2.2.1	Validate Admission request button
1.2.2.1.1	Show pending admission requests list
1.2.2.1.2	Show request details (patient name, reason, referral, doctor assigned)
1.2.2.1.3	Approve request
1.2.2.1.4	Reject request with remarks
1.2.2.1.5	Show updated requests list
1.2.2.2	Assign Ward/Room Button
1.2.2.2.1	Select admitted patient from list
1.2.2.2.2	Choose ward from dropdown
1.2.2.2.3	Choose room from dropdown
1.2.2.2.4	Confirm assignment
1.2.2.2.5	Show updated patient details
1.2.3	**Facility Allocation Management**
1.2.3.1	Show Facilities Button
1.2.3.1.1	Show list of facilities with availability status
1.2.3.1.2	Search by facility type

1.2.3.1.3	Filter by availability
1.2.3.2	Assign facility button
1.2.3.2.1	Select admitted patient from list
1.2.3.2.2	Choose facility type from dropdown
1.2.3.2.3	Set duration of use
1.2.3.2.4	Confirm assignment
1.2.3.2.5	Show updated facility allocation list
1.2.3.3	Approve/Reject Facility Request Button
1.2.3.3.1	Show pending facility requests from admitted patients
1.2.3.3.2	Take request to show details
1.2.3.3.3	Approve request and update allocation
1.2.3.3.4	Reject request with remarks
1.2.4	**Test Booking Oversight**
1.2.4.1	Show Test Booking button
1.2.4.1.1	Show list of all booked tests with patient name, test type, date, time
1.2.4.1.2	Search by patient name or test type
1.2.4.1.3	Filter by date
1.2.4.2	Approve/Reschedule Test booking button
1.2.4.2.1	Select booking from list
1.2.4.2.2	Approve as scheduled
1.2.4.2.3	Reschedule by choosing new date/time from date-time picker
1.2.4.2.4	Confirm changes and notify patient
1.2.5	**Medicine Stock Management**
1.2.5.1	Show Medicine Stock button
1.2.5.1.1	Show list of medicines with name, quantity, expiry date and price

1.2.5.1.2	Search by medicine name
1.2.5.1.3	Filter by expiry date
1.2.5.2	Update Stock button
1.2.5.2.1	Select medicine from list
1.2.5.2.2	Update quantity value
1.2.5.2.3	Submit stock update
1.2.5.2.4	View updated stock list
1.2.6	**Internal Announcements Button**
1.2.6.1	Create Announcement button
1.2.6.1.1	Enter title of announcement
1.2.6.1.2	Enter description/content
1.2.6.1.3	Select recipients (Doctor, Nurse, or both)
1.2.6.1.4	Submit announcement
1.2.6.1.5	View announcements list
1.2.6.2	View Announcements Button
1.2.6.2.1	Show list of all announcements with tile, date, sender
1.2.6.2.2	Click to read full content
1.2.6.3	Send Message/Protocol Update Button
1.2.6.3.1	Enter message/protocol change title
1.2.6.3.2	Enter detailed instructions of message body/new protocol
1.2.6.3.3	Enter recipients' username/mail
1.2.6.3.4	Submit message
1.2.6.3.5	View message history

1.2.7	**Nurse Allocation Management**
1.2.7.1	View Nurse Allocation List Button
1.2.7.1.1	Show list of all wards and rooms with assigned nurses
1.2.7.1.2	Search by nurse name
1.2.7.1.3	Filter by ward or room
1.2.7.2	Assign Nurse Button
1.2.7.2.1	Select nurse from nurse list
1.2.7.2.2	Choose ward or room from dropdown
1.2.7.2.3	Confirm nurse assignment
1.2.7.2.4	Notify nurse about new assignment
1.2.7.2.5	View updated allocation list [ref 1.2.7.1]
1.2.7.3	Reassign Nurse Button
1.2.7.3.1	Select current allocation from list [ref 1.2.7.1]
1.2.7.3.2	Take new ward or room from dropdown
1.2.7.3.3	Confirm reassignment
1.2.7.3.4	Notify nurse about reassignment
1.2.7.3.5	View updated allocation list [ref 1.2.7.1]
1.2.7.4	Remove Nurse Allocation Button
1.2.7.4.1	Select nurse allocation from list [ref 1.2.7.1]
1.2.7.4.2	Confirm removal
1.2.7.4.3	Notify nurse about removal
1.2.7.4.4	View updated allocation list [ref 1.2.7.1]
1.2.8	Logout Button
1.2.8.1	Show Login page [ref 1.0]

Table-4: Doctor Functional Requirements

1.1	**Registration**
1.1.1	Taking name
1.1.2	Taking email address
1.1.3	Taking user name
1.1.4	Taking password
1.1.5	Taking confirm password
1.1.6	Taking phone number
1.1.7	Taking date of birth
1.1.8	Taking department
1.1.9	Submitting the form
1.1.10	Resetting the form
1.2	**Login**
1.3	**Dashboard**
1.3.1	Side Menu
1.3.1.1	View Appointment List Button
1.3.1.1.1	Show all appointment schedule patient is taken on that day
1.3.1.1.1.1	Show patient name
1.3.1.1.1.2	Show appointment slot
1.3.1.1.1.3	Show Appointment Status (By default it is “Scheduled”)
1.3.1.1.1.4	Update Appointment Status
1.3.1.1.1.4.1	Changes Status to Appointment “Completed” Or “Canceled”
1.3.1.1.1.4.2	Updated the Status
1.3.1.1.1.2	Search patient from appointment list by using their user id

1.3.1.1.1.3	Show All Appointment Schedule (ref 1.3.1.1.1)
1.3.1.2	Assign Prescription Button
1.3.1.2.1	Open a Prescription Form
1.3.1.2.1.1	Date will be set by default appointment date
1.3.1.2.1.2	Take patient user id
1.3.1.2.1.3	Take diagnosis
1.3.1.2.1.4	Take list of medicine
1.3.1.2.1.5	Take any instruction or comments
1.3.1.2.1.6	Completed prescription assign
1.3.1.2.1.7	Prescription Form (ref 1.3.1.2.1)
1.3.1.3	View Patient History Button
1.3.1.3.1	Show previous patient record
1.3.1.3.1.1	Search specific patient by their user id
1.3.1.4	Diagnostic Report Button
1.3.1.4.1	Show all reports that patient is tested on this hospital lab
1.3.1.4.2	Search specific patient by their user id
1.3.1.5	Send Confirmation Notification Button
1.3.1.5.1	Send messages to patient
1.3.1.6	Leave Application Request Button
1.3.1.6.1	Open leave application form
1.3.1.6.1.1	Take leave duration and specific date
1.3.1.6.1.2	Take the description of reason for leave
1.3.1.6.1.3	Submit the application form

1.3.1.6.1.4	Leave application form (ref 1.3.1.6.1)
1.3.1.7	Feedback Button
1.3.1.7.1	Show all feedback given by patient
1.3.2	Notification Icon Button on the top of the right corner
1.3.2.1	Show messages from authority
1.3.3	Account icon button on the right side of the notification button
1.3.3.1	My Account Button
1.3.3.1.1	View profile
1.3.3.1.1.1	Show name
1.3.3.1.1.2	Show email
1.3.3.1.1.3	Show contact number
1.3.3.1.1.4	Show department
1.3.3.1.1.5	Show date of birth
1.3.3.1.1.6	Show profile picture
1.3.3.1.1.7	Edit Button
1.3.3.1.1.7.1	Show existing name and taking name
1.3.3.1.1.7.2	Show existing email and taking email
1.3.3.1.1.7.3	Show existing contact number and taking contact number
1.3.3.1.1.7.4	Show existing date of birth and taking date of birth
1.3.3.1.1.7.5	Show existing department and taking department
1.3.3.1.1.7.6	Updated Info
1.3.3.1.1.7.7	View Profile (ref 1.3.3.1.1)
1.3.3.1.1.8	Change profile picture button
1.3.3.1.1.8.1	Taking profile picture

1.3.3.1.1.8.2	Submission of the updated profile picture
1.3.3.1.1.8.3	View Profile (ref 1.3.3.1.1)
1.3.3.2	Logout Button
1.3.2.1	Show Login page

Table-5: Patient Functional Requirements

1.0	User home page-All features availability
1.1	**Registration**
1.1.1	Taking full name
1.1.2	Taking email address
1.1.3	Taking username
1.1.4	Taking password
1.1.5	Taking confirm password
1.1.6	Taking gender
1.1.7	Taking phone number
1.1.8	Taking date of birth
1.1.9	Taking address
1.1.10	Taking postcode
1.1.11	Taking city
1.1.12	Taking division
1.1.13	Submitting the form
1.1.14	Resetting the form
1.2	**Login**
1.2.1	Taking username

1.2.2	Taking password
1.2.3	Forgot password option
1.2.3.1	Submitting email address and receiving reset link
1.2.4	Submission of login
1.3	**Dashboard**
1.3.1	Side menu
1.3.1.1	Dashboard button [ref 1.3]
1.3.1.2	View profile button
1.3.1.2.1	Show name
1.3.1.2.2	Show e-mail
1.3.1.2.3	Show gender
1.3.1.2.4	Show phone number
1.3.1.2.5	Show date of birth
1.3.1.2.6	Show address
1.3.1.2.7	Show postcode
1.3.1.2.8	Show city
1.3.1.2.9	Show division
1.3.1.2.10	Show profile picture
1.3.1.2.11	Edit profile button [ref 1.12]
1.3.1.3	Schedule appointment button [ref 1.4]
1.3.1.4	View prescription/test results button [ref 1.5]
1.3.1.5	Order medication button [ref 1.6]
1.3.1.6	Pay bill button [ref 1.7]

1.3.1.7	Get admitted button [ref 1.8]
1.3.1.8	Rent facilities button [ref 1.9]
1.3.1.9	View doctors' profile button [ref 1.10]
1.3.1.10	Cancel appointment button [ref 1.11]
1.3.1.11	File complaints button [ref 1.13]
1.3.1.12	Notifications button [ref 1.14]
1.3.1.13	View history button [ref 1.15]
1.3.1.14	Logout button
1.4	**Schedule Appointment**
1.4.1	Select department
1.4.2	Select doctor
1.4.3	Select date
1.4.4	Select time slot
1.4.5	Confirm details
1.4.6	Submit appointment request
1.4.7	View confirmation message
1.5	**View Prescription/Test Results**
1.5.1	Select type (prescription or test result)
1.5.2	View list in table format
1.5.3	Click view button for details
1.5.4	Option to download as PDF
1.6	**Order Medication**
1.6.1	Search for medicine by name

1.6.2	View medicine details (price, availability)
1.6.3	Add to cart
1.6.4	View cart
1.6.5	Remove items from cart
1.6.6	Proceed to checkout
1.6.7	Select payment method
1.6.8	Confirm order
1.6.9	View order confirmation
1.7	**Pay Bill**
1.7.1	View outstanding bills list
1.7.2	Select bill to pay
1.7.3	Choose payment method
1.7.4	Enter payment details
1.7.5	Confirm payment
1.7.6	View payment receipt
1.8	**Get Admitted**
1.8.1	Fill admission form (personal and medical details)
1.8.2	Select room type
1.8.3	Confirm admission request
1.8.4	View admission confirmation
1.9	**Rent Facilities (If admitted)**
1.9.1	View available facilities
1.9.2	Select facility

1.9.3	Specify rental duration
1.9.4	Submit request
1.9.5	View confirmation
1.10	**View Doctors' Profile**
1.10.1	Search for doctor by name/specialty
1.10.2	View profile details (name, department, experience, contact)
1.10.3	View available appointment slots
1.11	**Cancel Appointment**
1.11.1	View upcoming appointments
1.11.2	Select appointment to cancel
1.11.3	Confirm cancellation
1.11.4	View cancellation confirmation
1.12	**Edit and Update Profile**
1.12.1	Show existing information (name, email, etc.)
1.12.2	Allow editing fields
1.12.3	Submit updated info
1.12.4	View updated profile
1.13	File Complaints
1.13.1	Open complaint form
1.13.2	Select complaint type
1.13.3	Enter complaint description
1.13.4	Submit complaint
1.13.5	View complaint status

1.14	**Send and Get Notifications**
1.14.1	View received notifications list
1.14.2	Click to view details
1.14.3	Send message/response if required
1.15	**View History**
1.15.1	Select history type (appointments, prescriptions, payments, admissions)
1.15.2	View history in table format
1.15.3	Filter and search history
1.15.4	Download history as PDF
1.16	Logout [ref 1.3.1.14]
1.16.1	Show Login page [ref 1.0]

Table-6: Nurse Functional Requirements

1.0	**User home page-All features availability**
1.1	**Registration**
1.1.1	Taking full name
1.1.2	Taking email address
1.1.3	Taking username
1.1.4	Taking password
1.1.5	Taking confirm password
1.1.6	Taking gender
1.1.7	Taking phone number
1.1.8	Taking date of birth
1.1.9	Taking address

1.1.10	Taking postcode
1.1.11	Taking city
1.1.12	Taking emergency contact
1.1.13	Submitting the form
1.1.14	Resetting the form
1.2	**Login**
1.2.1	Taking username
1.2.2	Taking password
1.2.3	Forgot password option
1.2.3.1	Submitting email address and receiving reset link
1.2.4	Submission of login
1.3	**Dashboard**
1.3.1	Side menu
1.3.1.1	Dashboard button [ref 1.3]
1.3.1.2	View profile button
1.3.1.2.1	Show name
1.3.1.2.2	Show e-mail
1.3.1.2.3	Show gender
1.3.1.2.4	Show phone number
1.3.1.2.5	Show date of birth
1.3.1.2.6	Show address
1.3.1.2.7	Show postcode
1.3.1.2.8	Show city

1.3.1.2.9	Show emergency contact
1.3.1.2.10	Show profile picture
1.3.1.2.11	Edit profile button [ref 1.12]
1.3.1.3	View assigned ward button [ref 1.4]
1.3.1.4	Update patient history button [ref 1.5]
1.3.1.5	Notifications button [ref 1.6]
1.3.1.6	Internal announcements button [ref 1.7]
1.3.1.7	Medicine refill request button [ref 1.9]
1.3.1.8	Complaint form button [ref 1.10]
1.3.1.9	Logout Button [ref 1.11]
1.4	**View Assigned Patients Page**
1.4.1	Show list of assigned patients with name, ward/room number, and admission date
1.4.2	Search bar for patient name or ID
1.4.3	Filter patients by ward or room
1.4.4	Select patient from list
1.4.5	View patient medical history [ref 1.5.1]
1.5	**Update Patient History Page**
1.5.1	Show patient's existing history with date, diagnosis, treatments, and notes
1.5.2	Add new record button
1.5.2.1	Enter date of entry
1.5.2.2	Enter diagnosis or update notes
1.5.2.3	Enter treatments given
1.5.2.4	Attach files or test results (optional)

1.5.2.5	Submit update for doctor approval
1.5.3	Release patient button
1.5.3.1	Confirm patient release request
1.5.3.2	System sends release request to Management for approval
1.5.3.3	Upon Management approval, system updates admission status to “Released”
1.6	**Notifications Page**
1.6.1	Show list of recent notifications with title, date, and brief message
1.6.2	Click to open full notification
1.6.3	Mark notification as read
1.7	**Internal Announcements Page**
1.7.1	Show list of announcements from Management with title, date, and content
1.7.2	Filter announcements by date or category (e.g., Protocol Changes, General Notice)
1.7.3	Click to view full announcement
1.8	**Edit Profile Page**
1.8.1	Show existing name and take updated name input
1.8.2	Show existing email and take updated email input
1.8.3	Show existing gender and take updated gender input
1.8.4	Show existing phone number and take updated phone number input
1.8.5	Show existing date of birth and take updated date of birth input
1.8.6	Show existing address and take updated address input
1.8.7	Show existing postcode and take updated postcode input
1.8.8	Show existing city and take updated city input
1.8.9	Show existing emergency contact and take updated emergency contact input

1.8.10	Submit updated profile information
1.8.11	View profile [ref 1.3.1.2]
1.9	**Medicine Refill Request Page**
1.9.1	Show list of medicines available in the nurse's assigned ward/room
1.9.2	Show current stock quantity for each medicine
1.9.3	Search bar for medicine name
1.9.4	Select medicine to request refill
1.9.5	Enter required quantity for refill
1.9.6	Add optional notes (e.g., urgency, usage reason)
1.9.7	Submit refill request to Management
1.9.8	View status of submitted requests (Pending, Approved, Rejected)
1.10	**Complaint Form Page**
1.10.1	Enter complaint subject/title
1.10.2	Select complaint category (e.g., Facility Issue, Staff Issue, Supply Issue)
1.10.3	Enter detailed description of the complaint
1.10.4	Attach supporting images or documents (optional)
1.10.5	Submit complaint to Admin
1.10.6	View status of submitted complaints (Pending, In Progress, Resolved)
1.11	**Logout**
1.11.1	Show Login page [ref 1.0]

2.2 Non-Functional Requirements

In addition to meeting its functional goals, the Hospital Management System must also satisfy several quality attributes to ensure that it is reliable, secure, and practical for everyday hospital use.

Performance:

The system should provide fast and efficient responses to user actions, with page loads and data retrieval occurring within 2–3 seconds under normal usage. It should be able to handle multiple users (staff and patients) accessing the system at the same time without performance issues, supporting at least a few hundred concurrent sessions in a mid-sized hospital environment.

Reliability:

The system must run smoothly with minimal downtime. Backup mechanisms should be in place to protect data in case of system crashes, and the system should be able to recover quickly from failures. Uptime of at least 99% is expected to ensure hospital operations are not disrupted.

Integrity/Security:

Data must be protected at all times. User authentication will ensure only authorized users can access the system, and role-based authorization will prevent unauthorized access to sensitive features. Patient information, billing data, and staff records should be encrypted and handled in compliance with data privacy standards to prevent misuse or loss.

Usability:

The system should be simple and intuitive so that hospital staff with limited technical training can use it effectively. Clear navigation, role-based dashboards, and user-friendly forms should ensure ease of use. Accessibility features should also be considered so that all users, including patients, can interact with the system comfortably.

Maintainability:

The system should be easy to maintain and update. Well-structured code, proper documentation, and use of version control (e.g., GitHub) will make it possible to add new features, fix bugs, or update modules without disturbing the rest of the system.

Scalability:

The system should be able to grow with the hospital's needs. It must support an increasing number of users, larger amounts of patient data, and additional features such as new departments or services in the future without requiring a complete redesign.

3. PROJECT ESTIMATION AND SCHEDULING

3.1 Effort and Cost Estimation

Scope of the Project

The Hospital Management System (HMS) will integrate different hospital activities into a single digital platform. It will provide secure login, role-based dashboards, and modules for Admin, Management, Doctor, Nurse, and Patient. The key features include profile management, appointment handling, admission management, medical history and prescriptions, billing, medicine stock and facility management, complaint handling, reporting, and test booking. The scope is large, with multiple modules requiring database design, secure authentication, real-time updates, and integration across all users.

The HMS will not include advanced features beyond the hospital's immediate needs such as AI-driven diagnostics, integration with national healthcare databases, or mobile apps in its first phase. The focus will remain on automating existing manual processes, reducing paperwork, and ensuring smooth communication between hospital staff and patients. This clear scope ensures that requirements are well-bounded, measurable, and achievable within the given budget, time, and resources, forming the basis for accurate effort and cost estimation.

The estimated size of the Hospital Management System is **65,000 Lines of Code (LOC)**

Productivity Rate =600 LOC per person-month.

Using LOC conventional method,

Effort (PM)= 65000/600 = 108 Person-Months

Using Dynamic multivariable model,

LOC = 65,000

Productivity parameter, P= 10

Project duration, t ≈ 14 months (from COCOMO result)

Productivity Factors, B= 1.1 (A mid-range B value is assumed)

$$\textbf{Effort (PM)} = [LOC * B^{0.333}/P]^3 * (1/t^4)$$

$$= [65000 * 27\text{^}0.333/10] \text{^}3 *(1/10\text{^}4)$$

$$= 7.4x10\text{^}8 \text{ Person-Months}$$

For the effort estimation of the Hospital Management System, we have used the Semi-Detached mode of the COCOMO model. This mode is appropriate because the project is of medium-to-large size (30K SLOC), involves multiple user roles and interconnected modules, and requires a team with mixed experience levels. Since our project requires careful integration but is not as rigid as embedded systems, the Semi-Detached approach provides the most realistic estimation of effort, development time, and staffing.

Using <u>Semi-Detached</u> mode,

Coefficient$_{\text{<Effort Factor>}}$= 3.0

Project complexity, P= 1.12

SLOC-dependent coefficient, T= 0.35

Effort (PM) = Coefficient$_{\text{<Effort Factor>}}$*(SLOC/1000) ^P

=3*(30,000/1000) ^1.12

= 135 Person-Months

Development time = DM = 2.50*(PM)^T

=2.50*(639.45) ^0.35

= 13.9 months

≈ 14 months

Required number of people = ST = PM/DM

= 135/14

= 10 people

3.2 Project Scheduling

Table-7: Task Breakdown

Activity	Assigned To	Start	End	Duration	Status	%Done
Task 1	Requirement Analyst	26-Jul	1-Aug	7	Complete	100%
Task 2	Requirement Analyst	2-Aug	8-Aug	7	Complete	100%
Task 3:	Designer	9-Aug	15-Aug	7	Complete	100%
Task 4	Team Leader+Designer	16-Aug	22-Aug	7	Complete	100%
Task 5	Backend Devloper	23-Aug	29-Aug	7	Complete	100%
Task 6	Fronted Devloper	30-Aug	5-Sep	7	Complete	100%
Task 7	Tester+ Devloper	6-Sep	12-Sep	7	Complete	100%
Task 8	Tester	13-Sep	19-Sep	7	In Progress	60%
Task 9	Full Team	20-Sep	22-Sep	3	In Progress	0%
Task 10	Full Team	23-Sep	26-Sep	4	In Progress	0%

Phase effort allocation (40–20–40 guideline):

Analysis & Design (40%): It includes gathering requirements, analyzing stakeholders, creating the Software Requirements Specification (SRS), designing system architecture and database, and making UI prototypes. The Business Analyst handles requirements, the System Architect works on architecture and database, and the UI/UX Designer creates prototypes.

Coding/Implementation (20%): It involves developing core modules like Authentication, Appointments, Admissions, Billing, Prescriptions, Inventory, Complaints, and Reports. Software Developers work on these modules under the Technical Lead

Testing & Deployment (40%): It includes unit and integration testing, user acceptance testing (with doctors, nurses, admin, and patients), performance and security checks, deployment, and user training. The QA Team manages testing, and the Project Manager coordinates deployment and training. Tasks are connected: requirements must be finished before design, design must be ready before coding, all modules must be done before testing, and successful testing leads to deployment.

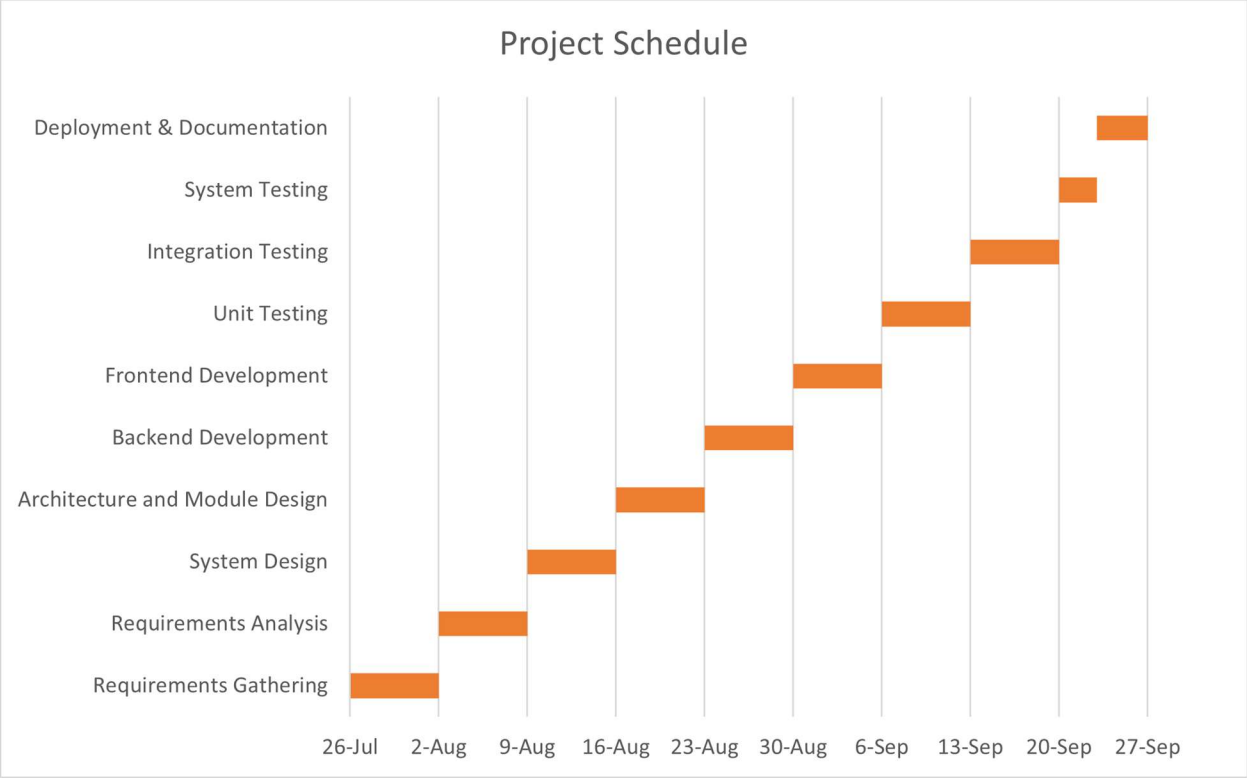

Fig-2: Shows a Gantt chart of the Project Schedule

Major Deliverables and Milestones

The project deliverables are tied to the milestones of the Waterfall development cycle, ensuring that outputs are clearly defined at each stage. During the Requirements Phase, the deliverables include the finalized Software Requirements Specification (SRS), problem statement, scope document, and use case models, marking the completion of requirement gathering. In the Design Phase, deliverables include system design diagrams such as class diagrams, data flow diagrams, activity diagrams, and the database schema, which together form the architecture blueprint. The Implementation Phase delivers completed functional modules for Admin, Management, Doctor, Nurse, and Patient roles, integrated into a working prototype of the Hospital Management System. The Testing Phase produces executed test cases, bug reports, and verification logs demonstrating that each requirement has been validated against the system. Finally, in the Deployment Phase, the deliverables include the deployed system in a test/live environment, user manuals, training materials, and the final project report. These milestones ensure that at every stage, tangible outputs are produced, reviewed, and approved before moving on, providing traceability and a clear measure of progress.

Earned Value Analysis

The Hospital Management System project has 25 planned work tasks that are estimated to require 582 person-days to complete. At the present checkpoint, 12 tasks have been completed, whereas according to the baseline schedule, 15 tasks were expected to be finished by this time.

Table-8

Task	Planned Effort	Actual Effort
1	12.0	12.5
2	15.0	11.0
3	13.0	17.0
4	8.0	9.5
5	9.5	9.0
6	18.0	19.0
7	10.0	10.0
8	4.0	4.5
9	12.0	10.0
10	6.0	6.5
11	5.0	4.0
12	14.0	14.5
13	16.0	-
14	6.0	-
15	8.0	-

Here,
Given total task= 25
BAC=582 Person Day
Budgeted cost of work scheduled (BCWS)=156.50
Budgeted cost of work performed (BCWP)=126.50
Actual cost of work performed (ACWP)= 127.50

Schedule performance index, SPI = BCWP/BCWS
= 126.50/156.50
= 0.808307

Schedule variance, SV = BCWP – BCWS
= 126.50-156.50
= -30 person-day

Cost performance index, CPI = BCWP/ACWP
= 126.50/127.50
= 0.99

Cost variance, CV = BCWP – ACWP
= 126.50-127.50
= -1 person-day

4. SOFTWARE DESIGN

4.1 System Design

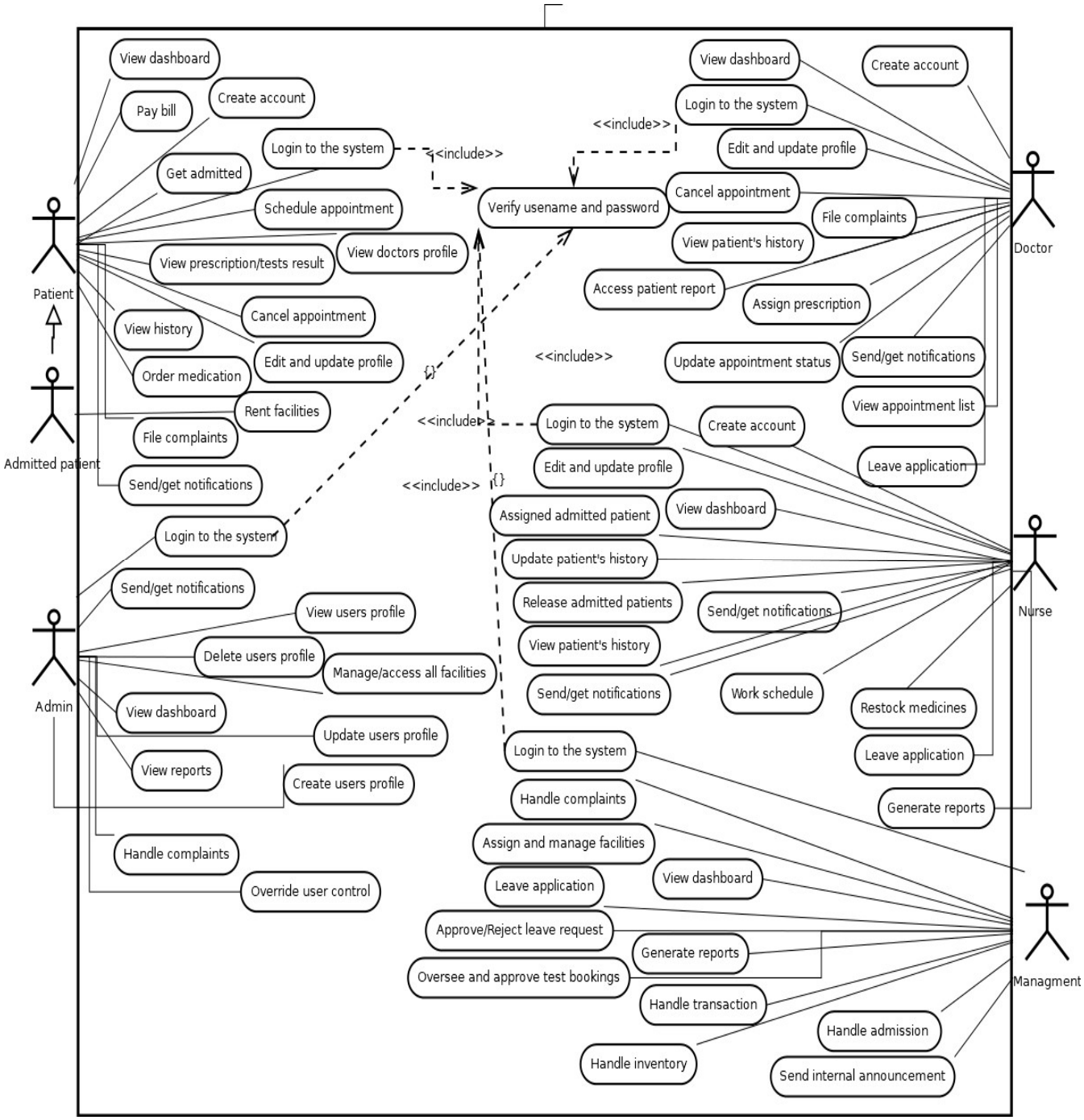

Fig-3: Shows the Use Case Diagram

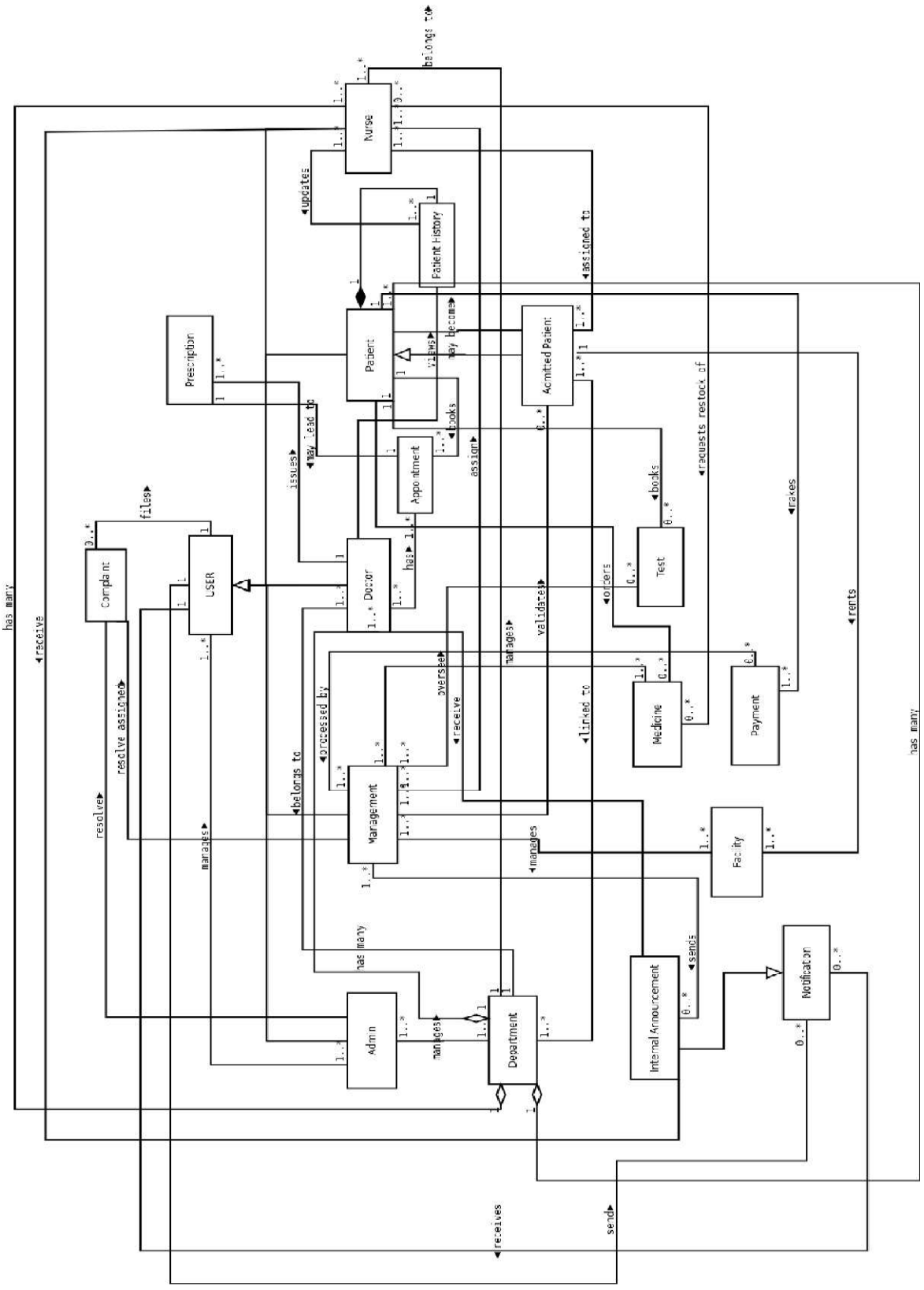

Fig-4: Shows the Class Diagram

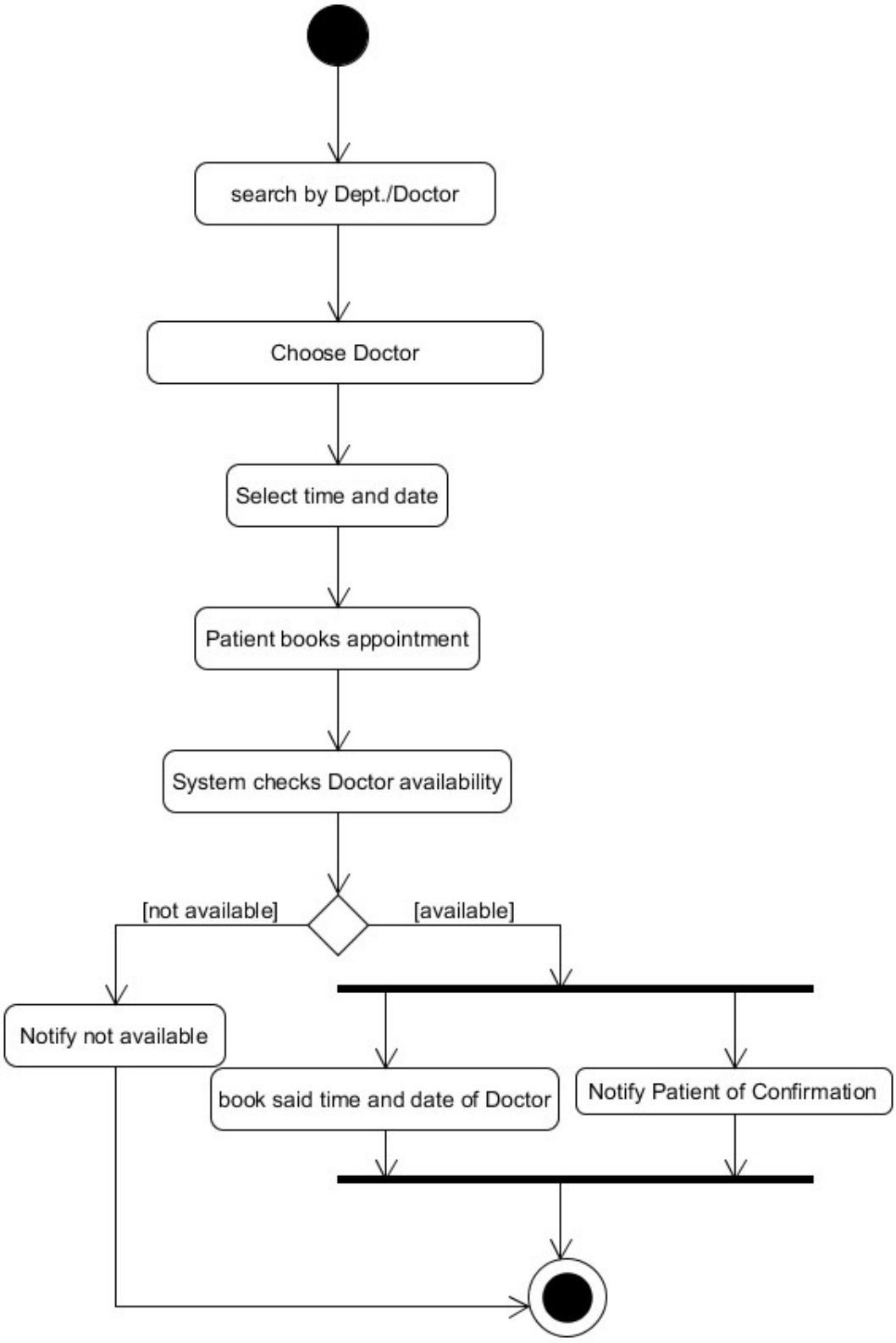

Fig-5: Shows the Activity Diagram for a Patient booking a doctor's appointment

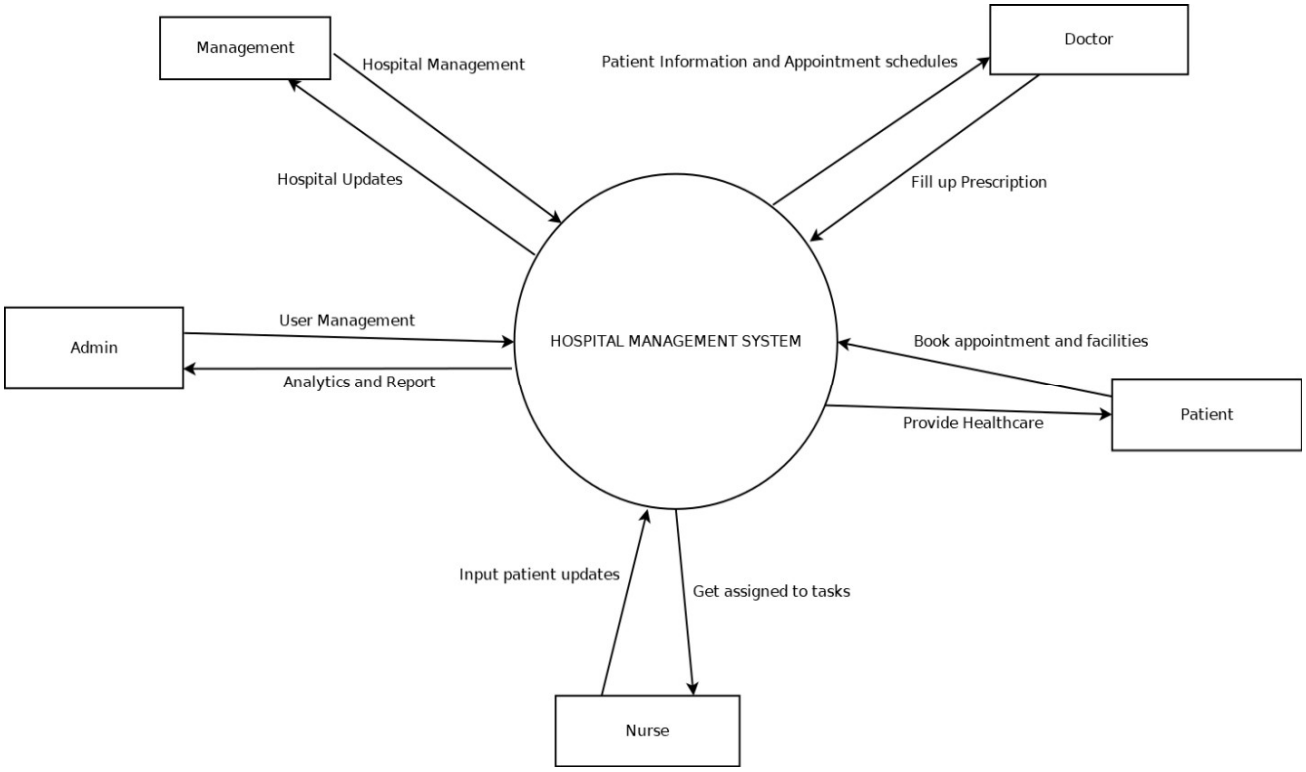

Fig-6: Shows the Data Flow Diagram

Consistency Across Diagrams

To maintain clarity and avoid confusion, the project ensures that all actors, classes, processes, and data flows are named consistently across every diagram and document. For example, the roles Admin, Management, Doctor, Nurse, Patient, and Admitted Patient are used with the same labels in the SRS, use case diagrams, class diagrams, activity diagrams, and data flow diagrams. Similarly, system processes such as Appointment Management, Admission Management, Complaint Handling, Medicine and Facility Management, and Billing are referenced with identical terminology throughout the models. Data items such as Patient ID, Appointment ID, Medicine Stock, and Facility Allocation follow uniform naming conventions across ER models, DFDs, and class diagrams. This consistency ensures that each diagram supports the others, making the system design easy to understand, traceable, and logically connected from requirements to implementation. By avoiding mismatched labels or ambiguous terms, the documentation becomes more reliable and the system easier to develop, test, and maintain.

4.2 UI / Wireframe Design using Figma

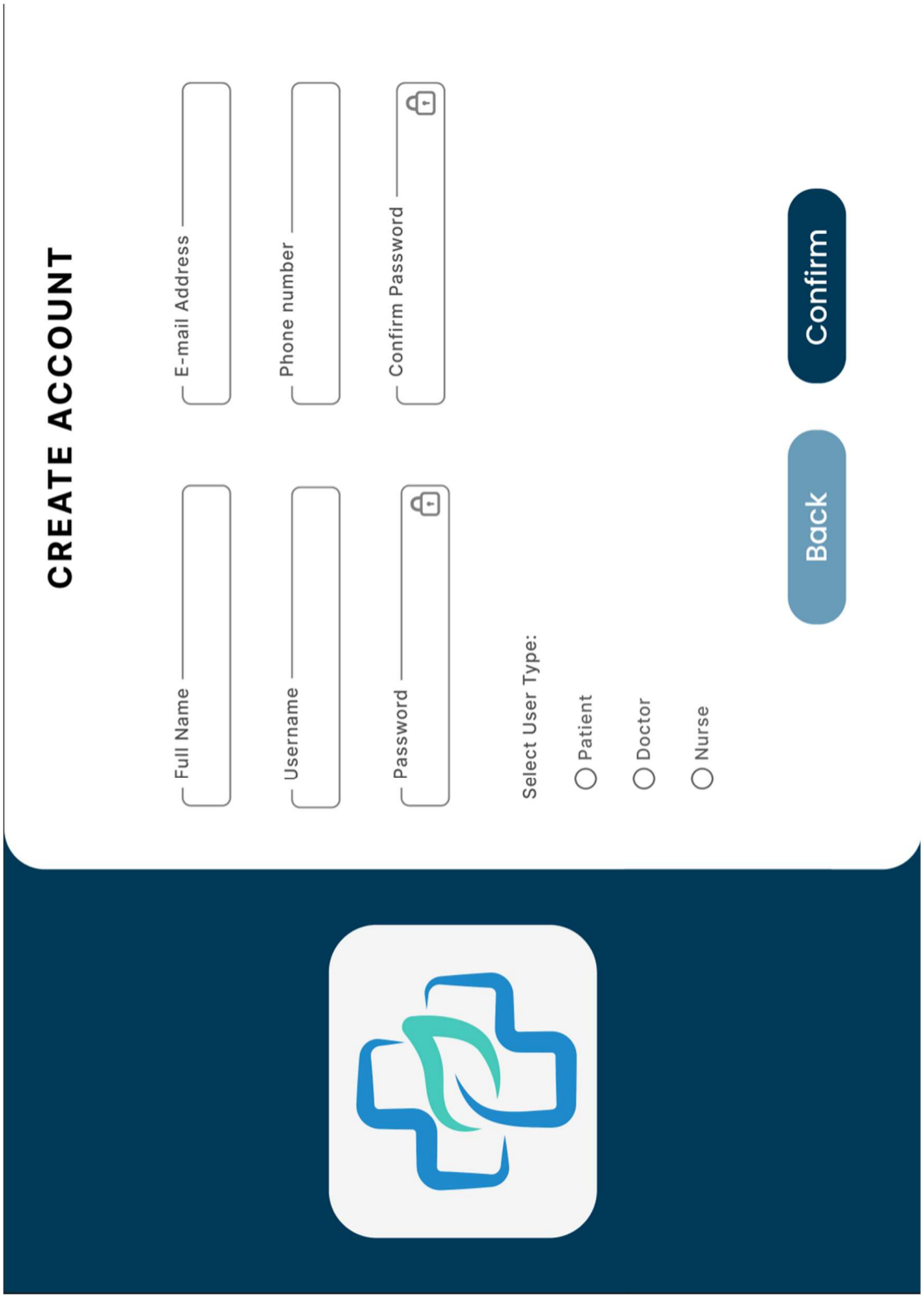

Fig-7: Shows Wireframe Design for Registration Page (part 1)

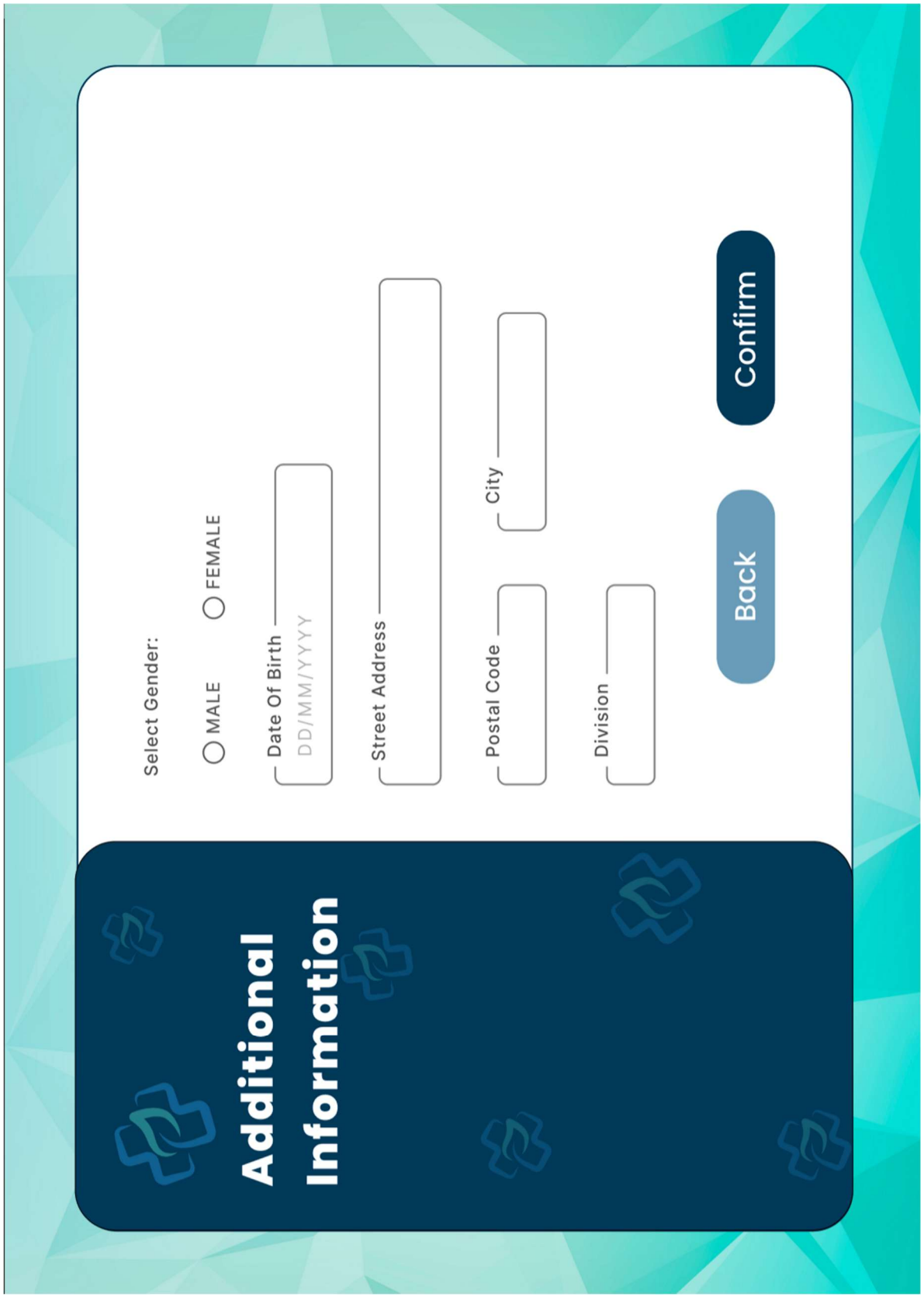

Fig-8: Shows Wireframe Design for Additional Information for Registration (part 2)

Fig-9: Shows Wireframe Design for Login Page

Fig-10: Shows Wireframe Design for Patient Dashboard

5. GIT WORKFLOW

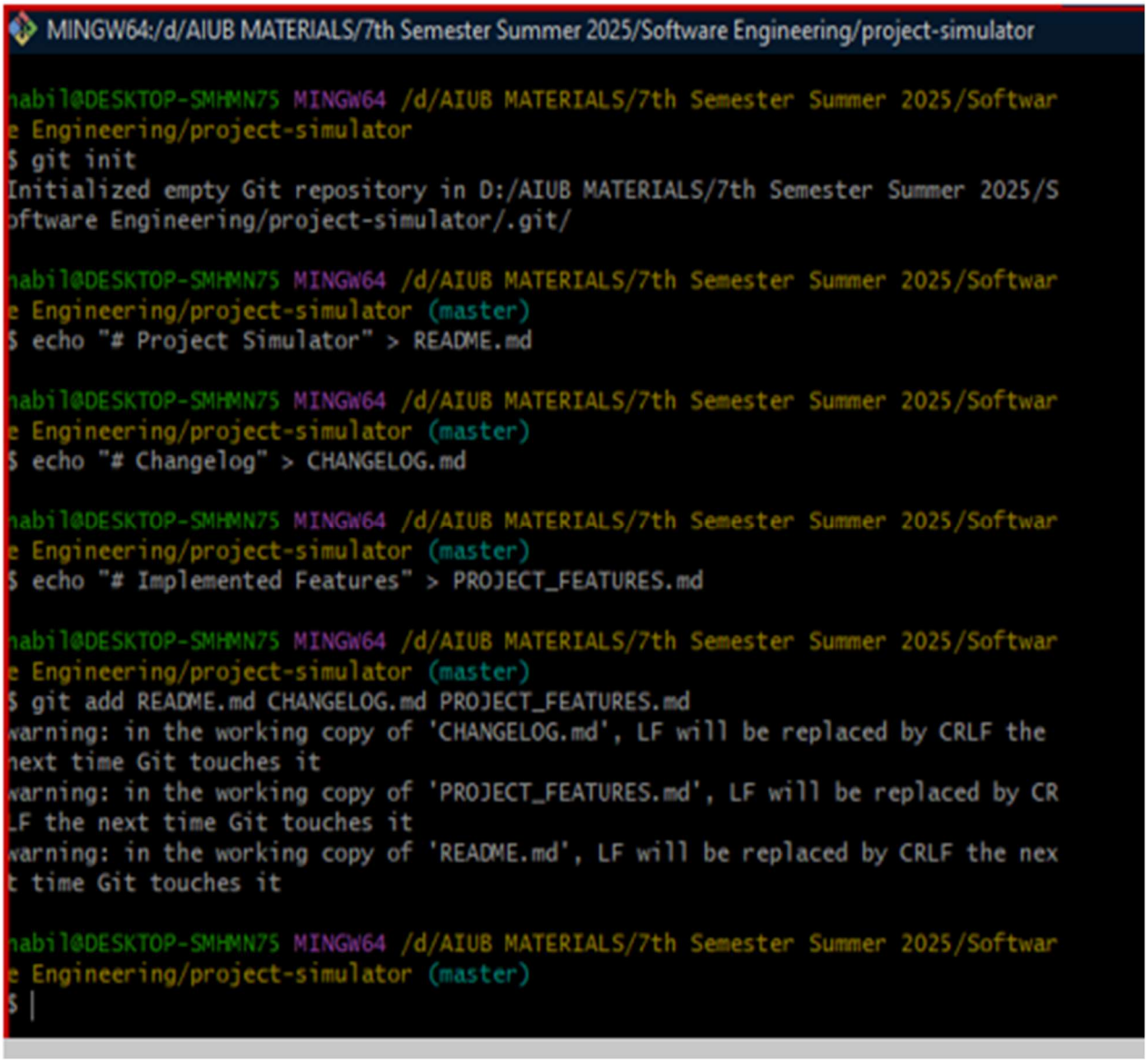

Fig-11: Shows Git Bash commands to create CHANGELOG, PROJECT FEATURES and README file in the master branch

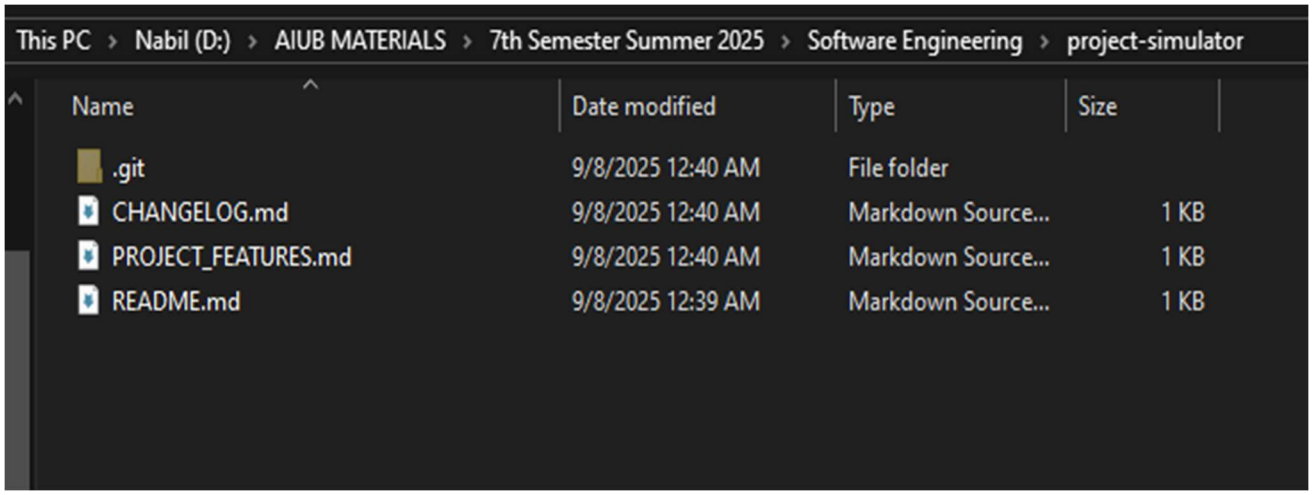

Fig-12: Shows Local Repository after git commands were executed

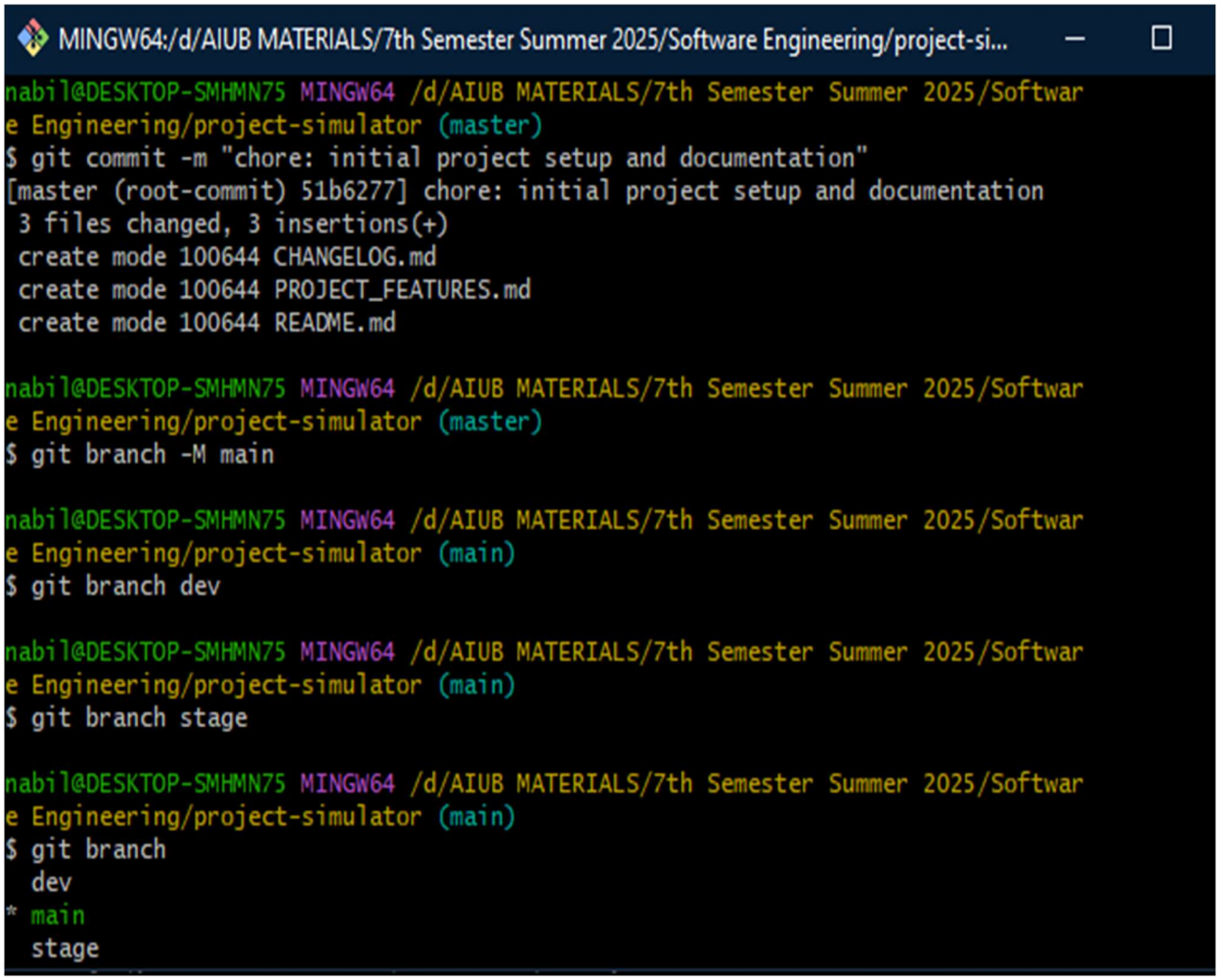

Fig-13: Shows Git Bash commands to rename ‘master’ branch to ‘main’ and create additional branches (dev and stage)

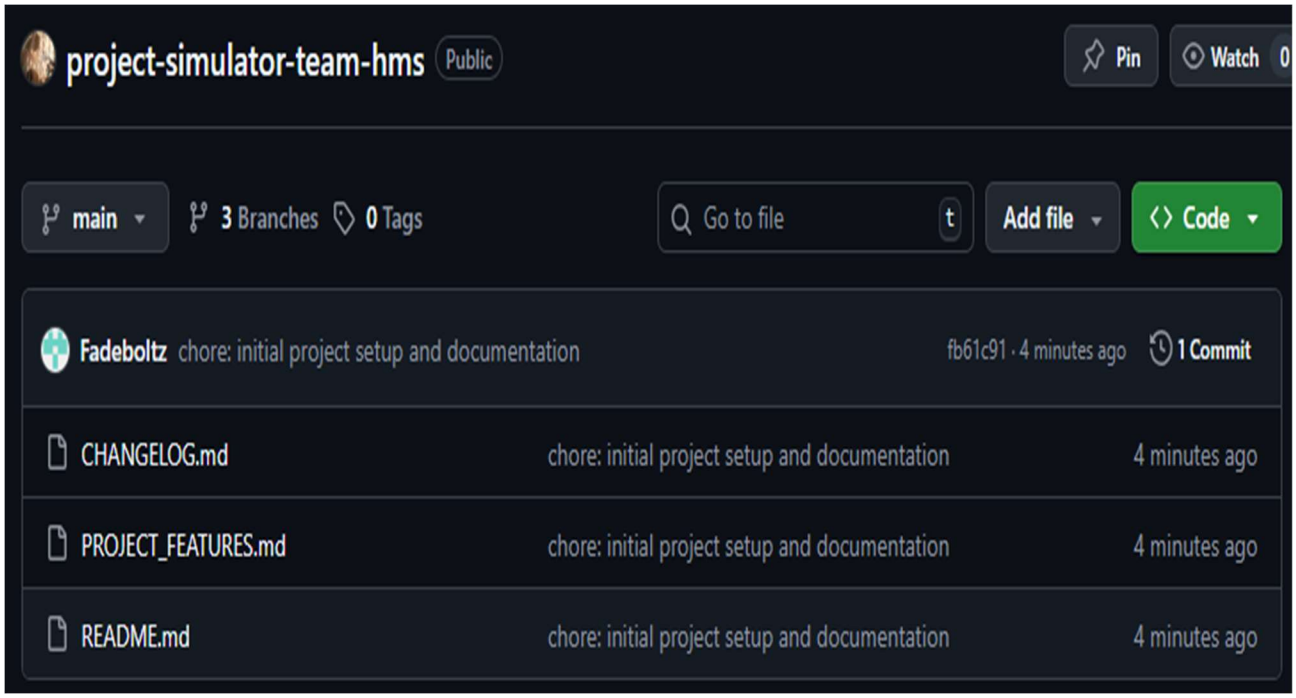

Fig-14: Shows Remote Repository after git commands were executed and branches were created

Fig-15: Shows Remote Repository after all feature branches were created by group members and pushed

Fig-16: Shows User Story Board in Trello after features get pushed to development (In Progress).

6. TESTING

The main goal of testing in our project is to ensure that the Hospital Management System is of high quality and that it meets all the requirements defined earlier. Testing helps to detect errors before deployment, verify that each function performs as intended, and validate that the system as a whole aligns with the client's needs. It also provides confidence that the solution is stable, secure, and ready to support hospital operations. In short, testing guarantees both requirements conformance and software quality assurance.

For our project, we plan to use a combination of testing methods:

- **Unit Testing:** Each module (Admin, Management, Doctor, Nurse, Patient) will be tested separately to confirm that its functions, such as booking an appointment or approving admissions, work correctly in isolation.
- **Integration Testing:** After individual modules are verified, they will be combined and tested together to ensure smooth interaction (e.g., patient booking an appointment that the doctor can later confirm).
- **System Testing:** The complete system will be tested in a simulated hospital environment to check overall functionality, performance, and stability.
- **Performance & Security Testing**: Before we release the system to the hospital, runing performance and security tests is necessary. Healthcare data and uptime are critical: test response times under expected load, concurrent user behavior, plus authentication, role-based access, encryption and common security checks. These must be validated as a formal pre-deployment step in Waterfall.
- **White-Box Testing:** White-box testing (unit tests, code-path checks, database query checks) is performed during the implementation phase by developers. It reduces defects early, makes later system testing smoother, and ensures internal logic and data handling are correct and efficient.

Table-9

Project Name: HOSPITAL MANAGEMENT SYSTEM			**Test Designed by: Mariya Mehrin**	
Test Case ID: TC_ADMIN_01			Test Designed date: 02/7/2025	
Test Priority (Low, Medium, High): High			Test Executed by: Mariya Mehrin	
Module Name: User Profile Management			Test Execution date: 22/08/2025	
Test Title: Create a New User Profile				
Description: Verify that the Admin can successfully create a new user profile (Doctor, Nurse, Management, Patient).				
Precondition: Admin account exists and is logged in. No duplicate username or email for the new user.				
Dependencies: Database connectivity must be active.				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Login as Admin** 2. **Click User Management** 3. **Click 'Create New User'** 4. **Fill in required fields** 5. **Submit form**	New user role: Doctor Name: Username: Email: Phone: Department:	User profile created and active in system.	Account created successfully	Pass

Table-10

Project Name: HOSPITAL MANAGEMENT SYSTEM			**Test Designed by: Mariya Mehrin**	
Test Case ID: TC_ADMIN_02			Test Designed date: 02/7/2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: Mariya Mehrin	
Module Name: Account Approval			Test Execution date: 22/08/2025	
Test Title: Approve Doctor Account for Login				
Description: Verify that Admin can approve pending doctor accounts, enabling them to log in.				
Precondition: Doctor profile exists but status = 'Pending'. Admin is logged in.				
Dependencies: Notification system must be active to notify approved user.				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Go to "Pending Approvals"** 2. **Select Dr. Chopper** 3. **Click approve** 4. **Click submit**	Doctor ID: DOC-202 Name: Dr. Tony Tony Chopper Current Status: Pending	Doctor account is active and can log in.	Doctor status = "Active"	Pass

Table-11

Project Name: HOSPITAL MANAGEMENT SYSTEM			**Test Designed by: Mariya Mehrin**	
Test Case ID: TC_ADMIN_03			Test Designed date: 02/7/2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: Mariya Mehrin	
Module Name: Complaint Management			Test Execution date: 22/08/2025	
Test Title: Review and Resolve Patient Complaint				
Description: Verify that Admin can review a submitted complaint and mark it as resolved.				
Precondition: At least one complaint exists in the system with status = Pending.				
Dependencies: Complaint subsystem operational.				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Go to "Complaints"** 2. **Select CMP-501** 3. **Click "Resolve" and enter note** 4. **Mark complaint as resolved**	Complaint ID: CMP-501 Submitted by: PAT-302 (Patient Luffy) Complaint: "Medicine delivery delayed"	Complaint CMP-501 marked as Resolved in database.	Complaint updated to "Resolved"	Pass

Table-12

Project Name: HOSPITAL MANAGEMENT SYSTEM			**Test Designed by: Mariya Mehrin**	
Test Case ID: TC_ADMIN_04			Test Designed date: 02/7/2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: Mariya Mehrin	
Module Name: User Profile Management			Test Execution date: 22/08/2025	
Test Title: Deactivate a User Profile				
Description: Verify that Admin can deactivate a user account, preventing login without deleting the profile.				
Precondition: User profile exists in Active state.				
Dependencies: Authentication system must not let users with inactive status to login.				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Go to User Management** 2. **Search for Nico Robin** 3. **Select "Deactivate"** 4. **Click confirm**	Nurse ID: NUR-210 Name: Nico Robin Current Status: Active	User profile remains stored, but login disabled.	User denied login.	Pass

Table-13

Project Name: HMS			**Test Designed by: Asifur Rahman**	
Test Case ID: TC_MGMT_01			Test Designed date: 2/7/2025	
Test Priority: Medium			Test Executed by: Asifur Rahman	
Module Name: Admission Management			Test Execution date: 22/08/2025	
Test Title: Validate Admission Request from Patient				
Description: Ensure Management can validate an admission request submitted by a patient				
Precondition: Patient has submitted an admission form				
Dependencies: Doctor referral may be required				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Go to "Admission Requests"** 2. **Select a patient request** 3. **Click Validate** 4. **Click submit**	Patient ID Patient Name Admission Form Referral (optional)	Admission Request being Approved	As expected	Pass

Table-14

Project Name: HMS			**Test Designed by: Asifur Rahman**	
Test Case ID: TC_MGMT_02			Test Designed date: 2/7/2025	
Test Priority: Medium			Test Executed by: Asifur Rahman	
Module Name: Facility Allocation Management			Test Execution date: 22/08/2025	
Test Title: Assign ICU to Admitted Patient				
Description: Allocation of available ICU beds to an admitted patient by Management				
Precondition: Patient is admitted; At least one ICU with status "Available"				
Dependencies: Facility list is up-to-date; Notifications are operational; Billing subsystem functional for immediate charge updates				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Go to the Facility Allocation** 2. **Select Admitted Patient** 3. **Choose Available ICU** 4. **Click Confirm**	Admitted Patient ID ICU Facility ID (status = Available) Start time	Patient profile and admission details are displayed System prompts to confirm assignment System sets particular ICU to "Occupied" Allocation list updated and searchable	As expected	Pass

Table-15

Project Name: HMS			**Test Designed by: Asifur Rahman**	
Test Case ID: TC_MGMT_03			Test Designed date: 2/7/2025	
Test Priority: Medium			Test Executed by: Asifur Rahman	
Module Name: Medicine Stock Management			Test Execution date: 22/08/2025	
Test Title: Approve Medicine Refill Request from Nurse				
Description: Verify that Management can review and approve a medicine refill request submitted by a nurse				
Precondition: Nurse has submitted a refill request with required quantity. Medicine item exists in catalog.				
Dependencies: Medicine stock records must reflect current quantities.				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Go to Medicine Stock Requests** 2. **Open to view details** 3. **Click Approve** 4. **Click confirm**	Nurse ID Request ID Medicine ID Requested quantity Current Stock	Stock records are updated, and Nurse is notified about approval and expected delivery timeline	As expected	Pass

Table-16

Project Name: HMS			**Test Designed by: Asifur Rahman**	
Test Case ID: TC_MGMT_04			Test Designed date: 2/7/2025	
Test Priority: Medium			Test Executed by: Asifur Rahman	
Module Name: Internal Announcements & Protocol Updates			Test Execution date: 22/08/2025	
Test Title: Send Protocol Change Announcement to Nurses and Doctors				
Description: Verify Management can compose, target, and send internal announcements (protocol updates) to selected staff groups and that recipients receive notifications				
Precondition: Management user is logged-in and has permission to send announcements				
Dependencies: Notification subsystem is operational				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Go to Internal Announcements** 2. **Enter title and body** 3. **Enter the recipients (roles or personal)** 4. **Click send**	Announcement title Announcement content Recipients' data	Announcements are visible in announcements history	As expected	Pass

Table-17

Project Name: HOSPITAL MANAGEMENT SYSTEM			**Test Designed by: Rashed Ahammod**	
Test Case ID: TC_DOC_01			Test Designed date: 02/7/2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: Rashed Ahammod	
Module Name: Appointment Status			Test Execution date: 22/08/2025	
Test Title: Modify the appointment status				
Description: Test the ability of system to update the current status of target user's appointment.				
Precondition: Target users have to take appointment.				
Dependencies: When target user takes appointment, the status should be by default "Scheduled".				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Go to Side Menu** 2. **Click Appointment List Button** 3. **Select user** 4. **Change Status** 5. **Click Update**	Status in "Completed" or "Cancelled"	The changes of the status should be successfully updated to "Completed" or "Cancelled".	Update Status applicable	Pass

Table-18

Project Name: HOSPITAL MANAGEMENT SYSTEM			**Test Designed by: Rashed Ahammod**	
Test Case ID: TC_DOC_02			Test Designed date: 02/7/2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: Rashed Ahammod	
Module Name: Assign Prescription			Test Execution date: 22/08/2025	
Test Title: Verify user can prescribe to target user				
Description: Test user can be able to assign a prescription for target user where by default user id is assigned and today date is auto generated.				
Precondition: The target user has a registered account.				
Dependencies: None				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Go to the Side Menu** 2. **Click Assign Prescription** 3. **Enter necessary information** 4. **Click submit**	System auto assigned user id on prescription. Issue date auto generated.	The user should assign prescription to the target user where expected user id and issue date should display form the target user perspective.	User performs Assign prescription session as per requirement	Pass

Table-19

Project Name: HOSPITAL MANAGEMENT SYSTEM			**Test Designed by: Rashed Ahammod**	
Test Case ID: TC_DOC_03			Test Designed date: 02/7/2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: Rashed Ahammod	
Module Name: Patient History			Test Execution date: 22/08/2025	
Test Title: Verify the user can view the history of target user.				
Description: Test the user can able to see the history of old patients during the follow up.				
Precondition: The target user must be completed appointment to the user minimum once.				
Dependencies: None				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Go to Side Menu** 2. **Click Patient History Button** 3. **Enter user id** 4. **Click Search**	User Id: *******	When the user searches by a specific User Id, the system should display only the target user history related to that User Id.	Users see any old target user history	Pass

Table-20

Project Name: HOSPITAL MANAGEMENT SYSTEM			**Test Designed by: Rashed Ahammod**	
Test Case ID: TC_DOC_04			Test Designed date: 02/7/2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: Rashed Ahammod	
Module Name: Diagnostics Report			Test Execution date: 22/08/2025	
Test Title: Verify the user can access to see the diagnostics report.				
Description: Test the user has access to see the online copy of diagnostics report through the system				
Precondition: The target user must perform their test on this hospital.				
Dependencies: Clear all payment related to test				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Go to Side Menu** 2. **Click Diagnostics Report Button** 3. **Enter user id** 4. **Click Search**	User Id: *******	When the user searches by a specific User Id, the system should display only the target user report related to that User Id.	User able to see online report copy of the selected target user	Pass

Table-21

Project Name: HMS			**Test Designed by: Israt Jahan Chowdhury**	
Test Case ID: TC_PAT_01			Test Designed date: 02/7/2025	
Test Priority (Low, Medium, High): High			Test Executed by: Israt Jahan Chowdhury	
Module Name: Appointment Management			Test Execution date: 22/08/2025	
Test Title: Verify patient can schedule an appointment				
Description: Ensure that a patient can book an appointment with a doctor by selecting department, doctor, date and time slot.				
Precondition: Patient must be logged in.				
Dependencies: Doctor must be available with free slots				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Go to Dashboard** 2. **Click "Schedule Appointment"** 3. **Select department** 4. **Select doctor** 5. **Select date and time** 6. **Confirm booking**	Department: Cardiology Doctor: Dr. Rahman Date and time: 12/09/2025, 10:30 AM	Patients can successfully book an appointment. Confirmation messages are displayed with details (doctor, date, time).	Appointment booked and shows up in Patient's upcoming appointments	Pass

Table-22

Project Name: HMS			**Test Designed by: Israt Jahan Chowdhury**	
Test Case ID: TC_PAT_02			Test Designed date: 02/7/2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: Israt Jahan Chowdhury	
Module Name: View Prescription/Test Results			Test Execution date: 22/08/2025	
Test Title: Verify patient can view prescriptions and test results				
Description: Ensure that a patient can access and download prescriptions/test results from the dashboard.				
Precondition: Patient must have completed at least one appointment or test				
Dependencies: Prescription/test results must be uploaded by the doctor				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Go to Dashboard** 2. **Click "View Prescription/Test Results"** 3. **Select type** 4. **Open details** 5. **Download file**	Prescription Prescription ID: PR001 / Test Test ID: TT001	Patients should be able to see prescriptions and/or test results in table format. On selecting one, full details should be displayed. File download option should generate a PDF.	Prescriptions/ tests can be seen.	Pass

Table-23

Project Name: HMS			**Test Designed by: Israt Jahan Chowdhury**	
Test Case ID: TC_PAT_03			Test Designed date: 02/7/2025	
Test Priority (Low, Medium, High): High			Test Executed by: Israt Jahan Chowdhury	
Module Name: Order Medication			Test Execution date: 22/08/2025	
Test Title: Verify patients can order medicine online				
Description: Ensure that a patient can search, add medicines to cart, and confirm order.				
Precondition: Patient must be logged in.				
Dependencies: Medicine must be available in stock.				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Go to Dashboard** 2. **Click "Order Medication"** 3. **Search for medicine** 4. **Add to cart** 5. **Proceed to checkout**	Paracetamol 500mg Qty: 2 Payment: Card	Patients can search medicine by name and see details. Selected medicine gets added to cart. Checkout and payment confirmation is shown. Order confirmation message is displayed.	Medicines are ordered and new stock quantity is updated in the database.	Pass

Table-24

Project Name: HMS			**Test Designed by: Israt Jahan Chowdhury**	
Test Case ID: TC_PAT_04			Test Designed date: 02/7/2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: Israt Jahan Chowdhury	
Module Name: File Complaints			Test Execution date: 22/08/2025	
Test Title: Verify patient can submit complaints				
Description: Ensure that a patient can file a complaint about hospital services and track complaint status.				
Precondition: Patient must be logged in.				
Dependencies: Admin must be available to handle complaint.				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Go to Dashboard** 2. **Click "File Complaint"** 3. **Enter complaint type** 4. **Enter description** 5. **Submit form**	Service Issue "Doctor was late for appointment."	Patients can select complaint type and enter details. Complaints should be submitted successfully. Complaint status should appear as "Pending" until handled by Admin.	Complaints are successfully sent.	Pass

Table-25

Project Name: HOSPITAL MANAGEMENT SYSTEM			**Test Designed by: NABIL NASIM**	
Test Case ID: TC_NUR_1			Test Designed date: 02/7/2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: NABIL	
Module Name: Login			Test Execution date: 22/08/2025	
Test Title: Verify nurse login with valid credentials				
Description: Ensure that the nurse can log into the system with a valid username and password.				
Precondition: Nurse must have a valid registered account.				
Dependencies: User credentials exist in the system database.				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Go to the site** 2. **Enter username** 3. **Enter password** 4. **Click login**	Username: nurse001 Password: test@123	Nurse is successfully logged in and redirected to the dashboard	As expected	Pass

Table-26

Project Name: HOSPITAL MANAGEMENT SYSTEM			**Test Designed by: NABIL NASIM**	
Test Case ID: TC_NUR_2			Test Designed date: 02/7/2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: NABIL	
Module Name: Patient History			Test Execution date: 22/08/2025	
Test Title: Verify nurse can update admitted patient's history				
Description: Test that nurse is able to update an admitted patient's medical history.				
Precondition: Patient must be admitted and assigned to the nurse.				
Dependencies: Patient record exists in system.				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Login as nurse** 2. **Navigate to Patient History section** 3. **Select assigned patient** 4. **Add/update medical notes** 5. **Save changes**	Patient ID: P12345 History update: 'Patient given antibiotics at 10:00 AM'	System successfully updates patient's history and stores new entry.	As expected	Pass

Table-27

Project Name: HOSPITAL MANAGEMENT SYSTEM			**Test Designed by: NABIL NASIM**	
Test Case ID: TC_NUR_3			Test Designed date: 02/7/2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: NABIL	
Module Name: Notifications			Test Execution date: 22/08/2025	
Test Title: Verify nurse can send and receive notifications				
Description: Check that the nurse can successfully send and receive system notifications.				
Precondition: Nurse is logged into the system.				
Dependencies: Other users like Doctor, Admin must exist in the system.				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Login as nurse** 2. **Go to Notifications** 3. **Type a new message for doctor** 4. **Click Send** 5. **Verify notification is sent** 6. **Check inbox for received notification**	Message: 'Patient P12345 requires medicine restock.	Message is delivered successfully and appears in recipient's notification list.	As expected	Pass

Table-28

Project Name: HOSPITAL MANAGEMENT SYSTEM			**Test Designed by: NABIL NASIM**	
Test Case ID: TC_NUR_4			Test Designed date: 02/7/2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: NABIL	
Module Name: Medicine Stock			Test Execution date: 22/08/2025	
Test Title: Verify nurse can request medicine restock				
Description: Ensure that the nurse can send restock requests for medicines when supply is low.				
Precondition: Nurse must be logged in.				
Dependencies: Medicine stock module available.				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. **Login as nurse** 2. **Navigate to Restock Medicines** 3. **Select medicine with low stock** 4. **Enter restock quantity** 5. **Click 'Submit Request'**	Medicine: Paracetamol Quantity: 50 units	System successfully generates a restock request and sends it to management.	As expected	Pass

7. SOFTWARE PRODUCT METRICS

The quality of the Hospital Management System (HMS) design has been evaluated using object-oriented software product metrics. These metrics including size, complexity, coupling, cohesion, and inheritance serve as key indicators to assess the structural integrity, maintainability, and overall effectiveness of the design.

Selected Classes in HMS:

- **User**
- **Admin**
- **Management**
- **Doctor**
- **Patient**
- **Nurse**
- **Appointment**
- **Admission**
- **MedicalHistory**
- **Prescription**
- **MedicineStock**
- **Complaint**
- **Billing**
- **Facility**
- **Report**

Number of Classes (NOCS)=15 classes

<u>Metrics and Calculations</u>

- **Size Metrics**

Methods per Class (MPC):

- Admin → 18 methods
- Patient → 12 methods
- Doctor → 10 methods
- Nurse → 8 methods
- Management → 12 methods

Average MPC = (18+12+10+8+12) ÷ 5 = 12 methods/class

<u>Weighted Methods per Class (WMC):</u>

WMC = Σ (complexity of each method)

Assuming average complexity = 2

- Admin = 18×2 = 36
- Patient = 12×2 = 24
- Doctor = 10×2 = 20
- Nurse = 8×2 = 16
- Management = 12×2 = 24
- Others (Appointment, Billing, etc.) → 5–10 methods each.

Coupling

Coupling Between Classes (CBC): Number of other classes a class is connected to.

- Admin = 6 (heavily connected: users, facility, complaints, billing).
- Patient = 5 (appointments, doctor, billing, facility, complaints).
- Doctor = 4 (patient, appointment, prescription, report).
- Nurse = 3 (patient, admission, report).
- Management = 5 (admission, facility, stock, billing, test booking).

Cohesion

Lack of Cohesion in Methods (LCOM):

- Admin: High LCOM (low cohesion) since it handles unrelated tasks (approvals, complaints, reports).
- Management: Moderate LCOM (admission, billing, stock, announcements).
- Doctor, Patient, Nurse: Low LCOM (good cohesion, tasks strongly related).

Patient class has 12 methods, the total possible unique pairs of methods is:

Total pairs=C(12,2) =m*(m−1)/2

=12*(12-1)/2

= 66

We assume,

- **Admission group:** 5 methods (work with admission data)
- **Billing group:** 4 methods (work with billing/finance data)
- **Stock/Announcement group:** 3 methods (work with supplies and announcements)

Number of method pairs that do share at least one instance variable, **Q**= C(5,2) + C(4,2) + C(3,2)

= 16 + 1+3 = 19

Number of method pairs that do not share any instance variable, **P** = 66-19=47

Since P >Q, LCOM = |P|-|Q|

=47-19

=28

Management LCOM = 28

Inheritance

- **Depth of Inheritance Tree (DIT):**

User →subclasses (Admin, Management, Doctor, Nurse, Patient)

DIT(User) = 0

DIT(children) = 1

Other classes (Appointment, Billing, etc.) = DIT = 0.

- **Number of Children (NOC):**
  - NOC(User) = 5
  - All others = 0

8. CONCLUSION AND FUTURE WORK

The Hospital Management System (HMS) developed in this project addresses the key challenges faced by a mid-sized public hospital that currently relies on manual processes. By introducing a centralized digital platform, the system ensures better organization of hospital operations, reduces paperwork, improves patient care, and strengthens communication between staff and patients. Through careful requirement gathering, design, implementation planning, and testing strategies, the system provides role-based features tailored to Admins, Management, Doctors, Nurses, and Patients. The use of the Waterfall model ensured a structured and disciplined approach, producing clear documentation and minimizing risks. Overall, the project demonstrates how software engineering principles can be applied to solve a real-world healthcare problem with a practical, affordable, and sustainable solution.

Although the current HMS design covers the essential needs of the hospital, there are opportunities for expansion in future iterations. Mobile application support can be added to improve accessibility for patients and staff on the go. Integration with external systems such as national health databases, laboratory information systems, and insurance providers could further enhance functionality. Advanced features like analytics dashboards, automated reporting, and AI-driven decision support (e.g., predicting patient flow or medicine shortages) may also be introduced to improve hospital efficiency. Additional security layers, such as biometric authentication and multi-factor login, can be considered to further safeguard sensitive data. Finally, future work could explore migrating the system to a scalable cloud-based infrastructure to support larger user bases and ensure seamless expansion as hospital needs grow.