

American International University-Bangladesh (AIUB)  
**Department of Computer Science  
Faculty of Science & Technology (FST)**

**PROJECT TITLE**

A Software Engineering Project Submitted

By

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Semester: Summer 24-25** | | **Section:** | **Group Number:** | |
| **SL** | **Student Name** | **Student ID** | **Contribution (CO3+CO4)** | Individual Marks |
| 1 | Mahbub Hasan | 22-47419-2 | 25% |  |
| 2 | Mujtahid Tabassum Mohona | 22-47430-2 | 25% |  |
| 3 | MD. Zobayer Hosen | 23-50348-1 | 25% |  |
| 4 | MD. Tauhidul Islam | 22-49388-3 | 25% |  |
| 5 |  |  |  |  |

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] |  |
| Impact identification: Analysis of the impact of societal, health, safety, legal, and cultural issues | [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] |  |



# PROJECT PROPOSAL

## Background to the Problem

Healthcare access in Bangladesh, especially in Dhaka, is often challenging due to overcrowded hospitals, traffic congestion, and limited availability of doctors at the right time. Patients often face long waiting times to book appointments, and sometimes they have to travel long distances for basic diagnostic tests or to rent medical equipment. This delay can affect timely diagnosis and treatment. Moreover, there is no centralized platform for patients to easily find doctors, book appointments, order tests at home, or rent medical resources.

At the same time, hospitals and diagnostic centres face difficulties in managing patient requests efficiently. Doctors also have limited tools to communicate with patients, manage appointments, or share health information securely. Existing solutions in Bangladesh are either fragmented or limited in scope, such as booking platforms for only appointments without including lab tests, equipment rental, or home services. Hence, there is a strong need for a comprehensive healthcare management system that connects patients, doctors, and hospitals in a single, reliable platform.

## Solution to the Problem and Process Model Selection

**Project Scope and Features**

This project is a healthcare management software for Bangladesh, especially Dhaka city, where patients, doctors, hospitals, and admins can connect through one platform.

**Scope includes:**

**Patient Side:** Appointment booking, serial tracking, cancel, upload reports, order home tests, rent hospital cabins/ambulances/equipment, read blogs, chat with AI Doctor.

**Doctor Side:** Manage profile (time, fee, qualifications), write blogs, cancel patient serial, view ratings, chat with patients.

**Hospital Side:** Manage tests, cabins, ambulances, and medical equipment (CRUD), accept/reject requests, check ratings.

**User Story Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **As a…** | **I want to…** | **So that…** | **Priority** |
| US-01 | Patient | Search for nearby doctors by specialty | I can book an appointment easily | High |
| US-02 | Patient | Upload reports for doctor | They can review before my visit | High |
| US-03 | Patient | Order home diagnostic test | I can get my report without going outside | High |
| US-04 | Patient | Book/rent cabin, ambulance, equipment | I can get service from home | Medium |
| US-05 | Patient | Chat with AI Doctor | I can get early guidance and doctor suggestion | Medium |
| US-06 | Doctor | Set my profile with fee, time, and address | Patients can see accurate info | High |
| US-07 | Doctor | Cancel appointment if needed | I can manage my schedule | High |
| US-08 | Doctor | Write health blogs | Patients can learn from me | Medium |
| US-09 | Hospital | CRUD for tests, cabins, equipment | Patients see updated info | High |
| US-10 | Hospital | Accept/reject patient requests | I can control my resources | High |
| US-11 | Admin | Monitor users and see monthly signup stats | I can track platform growth | High |
| US-12 | Admin | Create, update, delete user accounts | I can manage the system effectively | High |

**User Story Board (Trello)**

**Backlog in Trello:**

A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.

**Existing Software Solutions**

**Practo (India based)**: Appointment booking, but not fully for Bangladesh.

**Doctorola (Bangladesh)**: Limited to doctor appointment booking only.

**Sheba Platform**: Provides some home services, but not fully integrated with doctors, hospitals, and equipment.

**Hospital Internal Systems**: Used inside hospital but not patient-friendly mobile apps.

**Gap:** None of these provide **one full ecosystem** for patients, doctors, hospitals, and admins in Bangladesh context. Our system will solve that.

**Process Model Selection: Agile Scrum**

We will use the **Agile Scrum** model. This is because requirements are not fully stable. Patients, hospitals, and doctors may give feedback after each release, and we must update quickly.

**Why Agile Scrum is suitable:**

**Iterative development:** We can release core features (appointment booking, test order) first, then add advanced ones (AI chat, ambulance rental).

**Team communication:** Daily Scrum meetings help developers, testers, and designers stay on the same page.

**Flexibility:** Scope can be changed or extended sprint by sprint.

**Risk management:** If something fails in one sprint, we can fix in next sprint without big loss.

**Timely delivery:** Each sprint (2 weeks) delivers working features to real users.

**Analysis of Project Environment**

**Requirements:** Partly stable (appointment booking, user management) and partly changing (AI Doctor, chat features).

**Users:** Patients and hospitals in Dhaka may ask for more features after launch.

**Technology:** Using mobile apps + backend APIs. Future updates may require AI integration.

**Market environment:** High demand in Dhaka due to long queues in hospitals and busy traffic.

**How Agile Scrum Supports Team and Business Objective**

**Team Size:** Small(4 people). Scrum allows clear role distribution (Product Owner, Scrum Master, Developers, Testers).

**Communication:** Daily standups and sprint retrospectives ensure smooth task flow.

**Feasibility:** By delivering MVP first (appointments, test order), we can achieve business goals quickly and show value to investors/hospitals.

**Flexibility of Scrum**

Easy to adapt new features like online payment, telemedicine, or insurance integration.

If hospitals want new reporting features, it can be added in next sprint.

If patient feedback suggests UI changes, those can be applied quickly.

**Creative and Real-Life Solution Insight**

Our solution will save patients in Dhaka from wasting hours in traffic to just book an appointment or get a test. A patient sitting at home can order a test, get report online, and book the right doctor based on AI suggestions. Hospitals will reduce manual management workload. Doctors will gain visibility and better patient engagement.

By connecting **patient + doctor + hospital + AI support** in one platform, this project creates the first **all-in-one digital healthcare ecosystem for Bangladesh**.

**Contribution to scientific results**

Our project will contribute to scientific results in healthcare technology in Bangladesh. By collecting and analyzing real usage data, such as patient waiting time, number of appointments, test orders, and ambulance requests, we can identify patterns and challenges in Dhaka’s healthcare system. These data points can later be anonymized and documented to support future studies on urban healthcare access. In addition, our integration of an AI-powered assistant for basic health guidance will allow researchers to measure how artificial intelligence can improve early health decision-making in a developing country context. All findings will be documented in reports, technical papers, and datasets that can be reused for further scientific study.

**Evidence for model selection**

The Agile Scrum model is the most effective choice because of the nature of our problem and environment. In Dhaka, hospitals and patients may not fully know what they want until they see a working solution. Scrum allows us to deliver small usable features quickly, gather feedback, and make improvements without waiting for the entire system to be finished. This early delivery reduces risk and ensures stakeholders stay engaged. Evidence from many healthcare IT projects worldwide shows that iterative models like Scrum are more successful than rigid models when user needs are evolving. Thus, there is strong argument that Scrum is the right fit for our proposed solution.

**Managing project risks and uncertainties**

The Scrum model is highly effective in managing risks at different stages. In the early stage of the backlog, risks of unclear requirements are handled by refining stories with stakeholders. During development sprints, risks such as integration problems or low code quality are reduced through daily standups, code reviews, and automated testing. At the sprint review, stakeholders can identify risks of usability or acceptance, which are then fixed in the next sprint. This step-by-step approach ensures that risks are discovered early and solved before they become large problems.

**Relation to project schedule**

Scrum directly supports project scheduling by using fixed-length sprints. For example, in our project we may choose two-week sprints, meaning every two weeks we can deliver a working increment. This keeps the team focused and ensures that deadlines are respected. Even if the final scope changes, we will always have working software at the end of each sprint. For Dhaka healthcare providers, this means they can test and use features like booking or test ordering much earlier, rather than waiting for the whole platform to be finished. This predictable schedule also allows the management team to plan pilot launches on time.

**Justification against other models**

Other process models were considered but found less suitable. The Waterfall model requires fixing all requirements at the start, which is unrealistic in Dhaka’s healthcare environment where hospitals and patients often change their needs after trying the system. V model is good for risk management but is too complex and resource-heavy for our small team. Scrum provides the right balance between flexibility, team coordination, and timely delivery. Therefore, compared to other models, Scrum is the most practical and effective approach for this project.

**Final justification**

In summary, our choice of Scrum is justified by clear evidence and alignment with the real environment of Bangladesh. It helps us adapt to changing needs, manage risks step by step, and deliver working results in short time frames. It ensures patients, doctors, and hospitals in Dhaka see early value and can guide the project direction. It also supports research contributions by documenting data and outcomes along the way. No other process model offers this strong balance of adaptability, stakeholder involvement, and predictable scheduling, making Scrum the best approach for developing our healthcare management system.

## Project Role Identification and Responsibilities

**Main Roles Involved**

Our project team will follow Agile Scrum roles along with additional technical roles to ensure smooth development and management. The main roles include:

* Product Owner
* Scrum Master
* Development Team (Frontend Developer, Backend Developer, Mobile App Developer, Database Engineer, AI Engineer)
* UI/UX Designer
* QA Tester
* System Administrator / DevOps Engineer

**Responsibilities in Key Stages**

**Requirements Gathering:** The Product Owner will collect requirements from stakeholders (patients, doctors, hospitals). The Scrum Master ensures the backlog is clear. Developers and designers may also give input during backlog refinement.

**Design:** The UI/UX Designer prepares mockups and wireframes. The Backend and Database Engineer design the system architecture. The Product Owner validates design decisions with stakeholders.

**Implementation:** The Frontend, Backend, Mobile App developers write the code. The Scrum Master ensures sprint progress.

**Testing:** The QA Tester checks functionality, performance, and security. Developers also do unit testing. The Product Owner validates features against acceptance criteria.

**Deployment:** The System Administrator/DevOps Engineer manages deployment to server or cloud. The Scrum Master ensures release is aligned with sprint goals.

**Decision Making, Quality Assurance, and Resource Management**

Decision Making: The Product Owner makes final decisions on features and priorities. Technical decisions are shared among the development team with the Scrum Master facilitating.

Quality Assurance: The QA Tester is directly responsible, but all developers must follow coding standards. The Scrum Master ensures quality checks are done each sprint.

Resource Management: The Scrum Master manages time and team resources, while the System Administrator manages technical resources like servers, databases, and cloud infrastructure.

**Distribution of Responsibilities and Justification**

**Product Owner** is chosen because they have domain knowledge of Bangladesh healthcare and can speak directly with patients, doctors, and hospitals.

**Scrum Master** is allocated to someone skilled in team coordination and Agile process management.

**Frontend & Mobile Developers** are responsible for building patient, doctor, and hospital apps because of their expertise in MERN and mobile frameworks.

**Backend & Database Engineers** handle APIs, databases, and integrations because they have strong system design knowledge.

**AI Engineer** focuses on the AI Doctor feature since it requires special expertise in machine learning and NLP.

**UI/UX Designer** ensures that the app is simple enough for Bangladeshi users who may not be very tech-savvy.

**QA Tester** ensures features meet requirements before release.

**System Administrator / DevOps Engineer** manages deployment pipelines and keeps the system online 24/7.

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

## 2.1 Functional Requirements

**Patient User Role:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Item No. | Feature | Specification | User Story | Screen Definition | Acceptance Criteria |
| 1 | Welcome Screen | Splash Screen | As a user, I want an engaging splash screen while the app loads. | Splash screen with logo during app startup | Blue background splash screen with centered logo. Display for 2s minimum or until DOM loads. Clear branding on app launch. |
| 2 | Onboarding | Swipe Animation with Information | As a first-time user, I want to learn about the app through an informative carousel. | First-time user carousel with app introduction | Three-slide carousel for first-time users with headings and images. 'Get Started' button on final slide saves cookie to prevent re-display. |
| 3 | Authentication | Login or signup | As a user, I want to easily sign in or create an account with multiple options and secure verification. | Login/signup screen with social media and email options | Login options: Gmail, Facebook, email/password. 'Forgot Password' link. Signup form with email, password (8+ chars), name, phone. OTP verification via email. |
| 4 | Profile Setup | Upload profile picture | As a new user, I want to optionally add a profile picture to personalize my account. | Profile picture upload during onboarding | Profile setup screen after registration. Upload profile picture option with skip functionality. Accepts JPEG/PNG formats. |
| 5 | Profile Setup | User basic information entry | As a new user, I want to provide basic information or skip to continue using the app. | Additional profile information collection | Basic information form: gender, age, address. Skip option available. Data saved to user profile. |
| 6 | Location Service | Location permission or manual selection | As a user, I want to share my location for relevant local healthcare information. | Location access for personalized content | Location permission prompt or manual map selection. Homepage inaccessible until location provided. Location-based content display. |
| 7 | Search | Search bar with filtering | As a user, I want to search and filter healthcare services to find what I need quickly. | Homepage search with filtering capabilities | Search bar for doctors, hospitals, tests with real-time results. Filter button with options: doctor name, hospital, disease type, specialist, test name. |
| 8 | Content Discovery | Popular healthcare carousel | As a user, I want to see popular healthcare options in my area for quick selection. | Carousel displaying popular local healthcare options | Location-based carousel below search bar showing popular hospitals, services, doctors, tests. Swipeable navigation with clear labels. |
| 9 | Favorites | Favorite items functionality | As a user, I want to save preferred healthcare options for easy future access. | Favorite marking system for healthcare items | Love icon on each carousel item. Click to favorite and save to profile. Visual indication for favorited items. SQLite3 storage. |
| 10 | Content Navigation | Complete service listings | As a user, I want to see comprehensive lists of healthcare options in each category. | Full list view for each healthcare category | 'View All' button in each carousel section. Redirects to complete location-based list of hospitals, services, doctors, or tests. |
| 11 | Navigation | Home Screen navigation | As a user, I want easy navigation between different app sections. | Main navigation between app sections | Navigation bar with Home, Reviews, Blogs, Account icons. Home icon navigates to location-based content and services carousel. |
| 12 | Health Management | Medicine Reminder system | As a user, I want to manage medicine reminders with flexible scheduling options. | Medicine reminder with alarm functionality | Alarm-style medicine reminder interface. 'Add New' popup for time, weekdays, duration, snooze settings. Long-press for edit/remove options. Toggle on/off. |
| 13 | Content | Blog Page with engagement | As a user, I want to read health articles and engage with content through comments and reactions. | Health blog platform with interaction features | Blog list with clickable titles. Category filter bar. Sort by date or doctor name. React to posts and comment. Reply to comments. |
| 14 | AI Service | AI Doctor chat | As a user, I want to consult an AI doctor and access my conversation history. | AI doctor consultation chat interface | GPT-like chat interface with AI doctor. Side panel shows chat history. Real-time health advice responses. |
| 15 | Account Management | User profile management | As a user, I want to manage my profile information and account settings. | User profile editing and account control | Profile management: edit name, password, email, phone, address, picture. Email/phone changes need OTP verification. Account delete/deactivate options. |
| 16 | Order Management | Order History access | As a user, I want to view my service history and easily reorder or review past services. | Past service usage tracking | Order history list with dates. Review button and 'Place Again' option for each service. Clear chronological display. |
| 17 | Appointment Tracking | Track Appointment details | As a user, I want to track my appointment status and waiting times. | Current appointment status monitoring | Scheduled appointments list with date, serial number, estimated wait time, running serial. Real-time updates. |
| 18 | Service Management | Active Orders monitoring | As a user, I want to monitor my active rental services and their status. | Current rental service tracking | Active rental bookings (ambulance, oxygen, cabin) with status, duration, service details. Real-time updates. |
| 19 | Navigation | Navigation history stack | As a user, I want intuitive navigation with swipe gestures and history tracking. | Page navigation history and gesture controls | Navigation history stack. Swipe left/right navigation. Back button on each page. iOS gesture support. Session persistence. |
| 20 | Payment Processing | Payment Option selection | As a user, I want flexible payment options appropriate for different healthcare services. | Service-specific payment method handling | Payment options: mobile banking, card, bank payment for all services. Cash-on-hand for appointments, ambulance, oxygen. Advance payment only for cabins. |
| 21 | Review System | Doctor/hospital reviews | As a user, I want to rate and review healthcare providers to help others make informed decisions. | Healthcare provider feedback and rating system | Doctor/hospital profile pages with rating and comment sections. 1-5 star ratings with text comments. Public feedback display. |
| 22 | Provider Information | Doctor/hospital profile display | As a user, I want detailed information about healthcare providers to make informed choices. | Detailed healthcare provider profiles | Tap doctor/hospital cards to view profiles. |

**Doctor User Role:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Item No. | Feature | Specification | Screen Definition | User Story | Acceptance Criteria |
| 23 | Doctor Profile | Doctor profile header with picture and name | Welcome page with doctor's basic information display | As a doctor, I want my profile and name visible on the welcome page to confirm correct account access. | Profile picture and full name prominently displayed in header. Reliable image loading without errors. Name matches logged-in account exactly. |
| 24 | Patient Feedback | Patient comments management with ratings | Recent patient comments with detailed view and reply options | As a doctor, I want to see and respond to patient feedback to manage my reputation and improve service. | Display 3-5 recent comments by default. 'View All' button expands to complete feedback history. Reply button on each comment opens text input. Rating breakdown by categories. Load within 2 seconds. |
| 25 | Appointment Management | Serial appointment request handling | Patient appointment requests with approval controls | As a doctor, I want to review and manage appointment requests to control my schedule effectively. | Show appointment requests with patient name, date, time. Accept/Reject buttons process immediately. Auto-notify patients of doctor's decision. 'View All' shows complete history. |
| 26 | Navigation | Welcome page navigation shortcuts | Quick access icons for key doctor features | As a doctor, I want quick access to important features for efficient navigation between sections. | Welcome icon redirects to homepage. Quick-access buttons for account, reviews, blog management. All buttons respond within 1 second of click. |
| 27 | Account Management | Account settings and profile updates | Doctor account options and profile management | As a doctor, I want to manage my account details and control my visibility and security settings. | Profile editing with auto-save and confirmation. Account delete/deactivate requires password confirmation. Hospital listing request with pending status. Logout redirects to login screen. |
| 28 | Review Management | Reviews tab with filtering capabilities | Patient reviews with filtering and display options | As a doctor, I want to view and filter patient reviews to track performance and satisfaction levels. | All reviews load within 2 seconds. Filter by 1-5 star ratings and comment keywords. Display reviewer name, date, star rating, full comment. Filter results show under 0.5s delay. |
| 29 | Content Management | Blog page with post editor | Blog post creation and management interface | As a doctor, I want to create and manage blog posts to share medical knowledge with patients. | 'Add New' button opens blank editor. Tap-and-hold reveals Edit/Delete options. Auto-save drafts every 10 seconds. Published posts appear instantly in listing. |
| 30 | Patient History | Patient visit history tracking | Chronological patient visit history with search functionality | As a doctor, I want to view patient visit history to track consultations and enable follow-ups. | Display 50 recent visits by default. Show patient name, visit date/time, consultation reason. Real-time search by name/date under 500ms. Complete history loads within 3 seconds. |

**Hospital User Role:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Item No. | Feature | Specification | Screen Definition | User Story | Acceptance Criteria |
| 31 | Hospital Profile | Hospital profile header with picture and name | Welcome page with hospital's basic information display | As a hospital manager, I want my profile visible on the welcome page to confirm correct account access. | Hospital name and profile picture visible in header. Image loads without errors. Name matches logged-in account details. |
| 32 | Patient Feedback | Patient comments management with ratings | Recent patient comments with detailed view and reply options | As a hospital manager, I want to see and respond to patient feedback to manage reputation and improve service. | Display 3-5 recent comments by default. 'View All' expands to complete feedback history. Reply button on each comment. Rating breakdown by categories. Load within 2 seconds. |
| 33 | Appointment Management | Appointment request handling | Patient appointment requests with approval controls | As a hospital manager, I want to review and manage appointment requests to control scheduling. | Show patient name, requested date/time for each appointment. Accept/Reject buttons respond instantly. Auto-notify patients of decisions. |
| 34 | Navigation | Welcome page navigation shortcuts | Quick access icons for key hospital management features | As a hospital manager, I want quick access to important features for efficient navigation. | Welcome icon redirects to homepage. Quick-access buttons for account, reviews, blog management. All buttons respond within 1 second. |
| 35 | Account Management | Account settings and profile updates | Hospital account options and profile management | As a hospital manager, I want to manage account details and control visibility and security settings. | Profile edits save instantly with confirmation. Delete/deactivate requires password confirmation. Hospital listing shows pending status. Logout redirects to login screen. |
| 36 | Review Management | Reviews tab with filtering capabilities | Patient reviews with filtering and display options | As a hospital manager, I want to view and filter patient reviews to track performance and satisfaction. | All reviews load within 2 seconds. Filter by 1-5 star ratings and comment keywords. Display username, date, star rating, full comment. Filters apply instantly. |
| 37 | Content Management | Blog page with post editor | Blog post creation and management interface | As a hospital manager, I want to create and manage blog posts to share medical knowledge with patients. | 'Add New' button opens blank editor. Tap-and-hold reveals Edit/Delete options. Auto-save drafts every 10 seconds. Published posts appear immediately in list. |
| 38 | Doctor Management | CRUD operations for doctor listings | Paginated doctor list with approval workflow | As a hospital manager, I want to manage doctor listings to ensure only qualified practitioners are onboarded. | Real-time new requests display. Approvals add doctors to whitelist immediately. Rejected applicants receive email notification. Search returns results within 2 seconds. |
| 39 | Doctor Approval | Doctor whitelist management | Doctor request queue with approval controls | As a hospital manager, I want to manage doctors to keep our team qualified. | New requests appear instantly. One-click approve/reject functionality. Fast search result delivery. |
| 40 | Cabin Management | Cabin booking and availability system | Visual floor plan with color-coded cabin status markers | As a hospital manager, I want to monitor and update cabin statuses to optimize hospital capacity. | Status changes save instantly. New cabins appear on map real-time. Prevents double-booking. Tablet/desktop compatible interface. |
| 41 | Equipment Tracking | Medical equipment management | Sortable table of medical devices with checkout system | As a hospital manager, I need to track device usage to prevent shortages. | Real-time availability display. Auto-alerts for overdue items. Scanner-compatible checkout process. |
| 42 | Ambulance Tracking | Real-time ambulance monitoring | Live GPS map with ambulance locations and status panel | As a hospital manager, I need real-time ambulance tracking to ensure rapid response. | Location updates every 10 seconds. Single-tap status changes. Network outage functionality. Clear maintenance schedules. |
| 43 | Patient Support | Patient inquiry management system | Ticket-format patient query interface with reply system | As support staff, I need to respond to patient inquiries to improve service. | New queries blink for visibility. Auto-archive resolved tickets after 7 days. Search by keywords and ticket ID. |

## 2.2 Non-Functional Requirements

**Patient User Role**

# 3. PROJECT ESTIMATION AND SCHEDULING

## 3.1 Effort and Cost Estimation

* Define the scope of the project clearly before estimation.
* Apply **Lines of Code (LOC)** estimation by predicting the size of the system and converting it into effort using productivity rates.
* Apply the **COCOMO model** by selecting the appropriate project mode and calculating effort, development time, and staffing.
* Present results from all three estimation methods to show effort in person-hours or person-months.
* Mention assumptions and possible variation in results, since estimation always involves some level of uncertainty.

## 3.2 Project Scheduling

* Break the project into smaller tasks, show how the tasks are connected, and assign responsibilities.
* Allocate effort across phases such as analysis/design, coding, and testing (for example, using the 40–20–40 guideline).
* Develop a schedule with milestones and deadlines, represented in a timeline or Gantt chart.
* Include major deliverables tied to milestones and ensure outputs are clearly defined.
* Track progress through regular meetings, milestone reviews, and schedule checks.
* Use **Earned Value Analysis (EVA)** with measures such as Planned Value (BCWS), Earned Value (BCWP), Actual Cost (ACWP), Schedule Performance Index (SPI), Cost Performance Index (CPI), and variances (SV, CV) to evaluate progress.
* Recognize common causes of delay such as unrealistic deadlines, changing requirements, risks, technical or human issues, and miscommunication, and plan accordingly.

# SOFTWARE DESIGN

**4.1 System Design**

* Draw the system design for your project using **Draw.io** or **LucidChart**.
* Prepare a **Use Case Diagram** by first defining all users (actors) and their roles. Show each actor’s interactions with the system through use cases inside a system boundary. Include relationships like include or extend where needed.
* Prepare a **Class Diagram** by identifying the main classes from your project. Add attributes and operations for each class, and show associations, generalizations, aggregations, or compositions between them.
* Prepare an Activity Diagram that visually represents the workflow of a system or process.
* Prepare a **Data Flow Diagram (DFD)** starting with the **Context Diagram** (system as one process with external entities).
* Ensure consistency in naming actors, classes, processes,and data flows across all diagrams so they support each other.

## UI / Wireframe Design

* Design the user interface of your project individually using **Figma** or **Balsamiq**.
* Build the wireframes directly from your **PRD table**. Each functional requirement defined in the PRD should have a corresponding screen or component in the UI.
* Use the **Trello user board** (or equivalent task board) as a reference to decide which features and user flows need to be represented in the prototype.
* Create clickable wireframes that connect the main screens together, showing how a user will navigate through the system.
* Ensure the wireframe matches the actors, roles, and functionalities described in the PRD and system design diagrams.
* Export the prototype and include screenshots or a link to your design as part of the report submission.

# GIT WORKFLOW

* Create a central repository for the project on GitHub and set the **master (or main) branch** as the primary branch for integration.
* Each member should clone the repository and create their own **feature branches** for assigned tasks. Work on new features or fixes within these branches.
* Add files, stage them and commit changes with clear messages that describe the purpose of each update.
* Push commits from the feature branches to the remote repository so other members can see progress.
* Use **pull** to fetch and integrate changes from the remote repository into local copies, ensuring everyone stays updated.
* Merge feature branches into the **master/main branch** only after the work is tested and reviewed, resolving any conflicts that occur.
* Show evidence of collaboration by maintaining a clear commit history (using logs) with multiple commits, merges and contributions from all group members.
* Keep the repository organized with a clean history that tracks the project workflow from initialization to completion.

# TESTING

* The goal is to show how testing ensures quality and requirements conformance.
* Identify some testing methods that you want to use in the testing phase later for your project.
* Prepare **test cases** using a manual test case template which template taught you in the class.

# SOFTWARE PRODUCT METRICS

* Apply software product metrics to evaluate your project.
* Use **Function-based Metrics (Function Points)** by identifying inputs, outputs, files, interfaces, and inquiries to measure the functionality delivered by the system.
* Apply **Object-Oriented and Class Metrics** such as size, complexity, coupling, cohesion, and inheritance measures (e.g., WMC, DIT, NOC, CBC, LCOM) to assess the structure and quality of your design.
* Include **Operation-oriented Metrics** by measuring average operation size, complexity, and number of parameters per operation.
* Apply **Maintenance Metrics** such as the **Software Maturity Index (SMI)** to check the stability of your project across changes and updates.

# CONCLUSION AND FUTURE WORK

* Write a conclusion and future work based on your project.

**Text Format:**

* Style: Times New Roman
* Size: 12
* Space: 1.15
* Alignment: Justify