



American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

PROJECT TITLE

Medi-Team Assist

A Software Engineering Project Submitted

By

Semester: Summer_21_22		Section:	Group Number:	
SN	Student Name	Student ID	Contribution (CO3+CO4)	Individual Marks
	HRIDOV SAHA	23-51060-1		
	TAFSIR RAHMAN	23-51061-1		
	MOST. SHUMAYA JAHAN	23-51070-1		
	GRONTHO CHANDRA ROY			

The project will be Evaluated for the following Course Outcomes

CO3: 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	
	[5 Marks]	
	Evidence of Argumentation regarding process model selection	
	[5Marks]	
CO4: Develop project management plan to manage software engineering projects following the principles of engineering management and economic decision process	Total Marks	
	[5Marks]	
	Develop the project plan, its components of the proposed software products	
	[5Marks]	
Identify all the activities/tasks related to project management and categorize them within the WBS structure. Perform detailed effort estimation correspond with the WBS and schedule the activities with resources	[5Marks]	
Identify all the potential risks in your project and prioritize them to overcome these risk factors.	[5Marks]	

Description of Student's Contribution in the Project work

<p>Student Name:</p> <p>Student ID:</p> <p>Contribution in Percentage (%):</p> <p><u>Contribution in the Project:</u></p> <ul style="list-style-type: none">▪ Contribution Description 1▪ Contribution Description 2 <hr/> <p>Signature of the Student</p>
<p>Student Name:</p> <p>Student ID:</p> <p>Contribution in Percentage (%):</p> <p><u>Contribution in the Project:</u></p> <ul style="list-style-type: none">▪ Contribution Description 1▪ Contribution Description 2 <hr/> <p>Signature of the Student</p>
<p>Student Name:</p> <p>Student ID:</p> <p>Contribution in Percentage (%):</p> <p><u>Contribution in the Project:</u></p> <ul style="list-style-type: none">▪ Contribution Description 1▪ Contribution Description 2 <hr/> <p>Signature of the Student</p>
<p>Student Name:</p> <p>Student ID:</p>

Contribution in Percentage (%):

Contribution in the Project:

- Contribution Description 1
- Contribution Description 2

Signature of the Student

Student Name:

Student ID:

Contribution in Percentage (%):

Contribution in the Project:

- Contribution Description 1
- Contribution Description 2

Signature of the Student

1. PROJECT PROPOSAL

2. Background to the Problem

Nowadays, many people face problems when they feel sick but can't quickly reach a doctor. Sometimes they search their symptoms online and try to take medicine on their own, which can be dangerous. This happens more often in rural areas where medical support is not always available. Because of this, people may take the wrong medicine, delay proper treatment, or face serious health risks.

Even though there are some health apps or websites, most of them just show general advice and don't give doctor-approved prescriptions. Also, many of these platforms are hard to use or don't connect patients directly with real doctors. That's why patients still suffer from wrong or late treatments.

Our project, **MediTeam Assist**, is designed to solve this problem. It helps users search for their symptoms and get proper prescriptions that are suggested and approved by real doctors. If no existing prescription is found for the given symptoms, a doctor can create and suggest a new one. The app also includes a secure payment system that shares earnings fairly between the doctor and the platform.

In addition, **MediTeam Assist supports 24-hour service**, so patients can get help anytime they need. The platform connects patients directly with real doctors, making the whole process fast, safe, and reliable.

Root Cause of the Problem:

The main cause of this problem is the **lack of easy and quick access to qualified doctors**, especially in rural or remote areas. Many people do not have the time, transportation, or nearby facilities to see a doctor when they feel sick. As a result, they try to solve their health issues by searching online or asking untrained people. This leads to **self-medication, wrong treatments, and even serious health risks**.

Another cause is that most existing health platforms only give **basic suggestions** instead of personalized, doctor-approved prescriptions. They also do not offer real-time connection with doctors or 24/7 support, which leaves patients without proper help when they need it most.

This Problem Is Important to Consider:

- Wrong medications can cause harmful side effects or long-term health issues
- Delays in proper treatment can make simple problems more serious

- Patients need access to real doctors, not just general advice
- Many people in remote areas have no other option, so a digital solution is necessary
- Safe and secure medical support is a basic human need

1.2 Solution to the Problem

The objective of **MediTeam Assist** is to provide a convenient, reliable, and doctor-verified digital platform where users can report symptoms and receive accurate prescriptions promptly. This system mitigates problems of healthcare access and unsafe self-treatment by integrating real-time doctor consultation, symptom analysis, and prescription management.

Proposed Solutions

- A user-friendly symptom input interface allowing users to describe or select their symptoms
- A database of doctor-verified prescriptions matched to common symptoms
- Automatic routing of unmatched cases to licensed doctors for personalized consultation and prescription creation
- 24/7 doctor availability to ensure timely assistance
- Secure authentication with role-based access for patients, doctors, and admins
- Integrated digital payment gateway with transparent revenue sharing between doctors and the platform
- History management and notifications to help users track their prescriptions and consultations

This solution is particularly appropriate because it directly targets the root causes of limited healthcare access and unsafe self-medication by providing verified, doctor-approved prescriptions rather than generic advice. The integration of real-time doctor interaction ensures reliability and trustworthiness, which are crucial for medical applications. The secure payment system aligns with business sustainability goals by fairly compensating doctors and maintaining platform operations.

The solution is feasible to meet the business objectives because it leverages mature, scalable technologies such as React.js, Node.js, Firebase, and modern payment gateways, which allow for rapid development and deployment. The model can easily scale to accommodate more users and doctors as demand grows, and the revenue sharing scheme ensures long-term financial viability.

MediTeam Assist utilizes state-of-the-art technology to deliver a creative and impactful healthcare solution. Key functionalities include:

- Symptom-based Input & Matching: Advanced symptom entry options and AI-assisted matching with existing prescriptions to quickly guide users.
- Doctor-Guided Prescription: When symptoms do not match existing records, real doctors provide personalized prescriptions, ensuring medical accuracy.
- 24/7 Doctor Availability: Telemedicine with real-time chat or video support enhances patient care, especially in emergencies.
- Secure User Authentication & Role-Based Dashboards: Ensures privacy, data security, and customized interfaces for patients, doctors, and admins.
- Digital Payment Integration: Supports seamless, cashless transactions and fair revenue distribution, encouraging doctor participation.
- History & Notification Management: Users can review past prescriptions and receive reminders for follow-ups or medication schedules.

This system's societal impact is significant. It promotes public health by reducing risks of improper medication, improves access to quality healthcare for underserved populations, and encourages preventive care. By using digital prescriptions and teleconsultations, it enhances safety and reduces unnecessary hospital visits. The system also complies with legal and cultural norms by securing patient data and involving licensed medical professionals.

Target Users and Benefits

- **Patients:** Especially those in remote or underserved regions gain quick, affordable access to reliable medical advice and prescriptions without traveling or waiting in clinics.
- **Doctors:** Gain a platform to extend their services digitally with fair compensation, reaching more patients flexibly.
- **Healthcare Providers and Systems:** Benefit from reduced overcrowding and better-managed patient flow.
- **Community:** Gains healthier populations through early and accurate treatment, reducing disease spread and complications.

The users benefit from an accessible, trustworthy, and convenient healthcare channel that promotes safety and well-being while simplifying the prescription process.

This project advances scientific knowledge by demonstrating how telemedicine combined with symptom-based AI and secure payment integration can improve healthcare delivery. The data collected on symptoms, prescriptions, and consultations provides a valuable resource for epidemiological research and healthcare planning. This comprehensive, doctor-verified approach to symptom management and prescription issuance contributes to developing new telehealth models and policy frameworks, advancing medical informatics and digital health services.

Previous studies have explored telemedicine and symptom checker apps, noting improvements in access but often highlighting concerns about the reliability of non-doctor verified advice. Most existing apps offer generic symptom assessments but lack integrated real-time doctor prescriptions and secure payment mechanisms. Recent research stresses the importance of combining AI symptom analysis with professional medical validation to improve outcomes and trust.

Our project extends these studies by integrating:

- Real-time doctor consultations for unmatched symptoms
- Verified prescription management
- Secure role-based access and payment systems

This extension addresses gaps in trust, reliability, and financial sustainability identified in earlier works.

Existing platforms like WebMD, HealthTap, and Babylon Health provide symptom checking and some teleconsultation features but often lack real-time, doctor-issued prescriptions and integrated payment distribution. They may also have limited availability or accessibility in low-resource settings.

MediTeam Assist builds on these by offering:

- 24/7 doctor-guided prescriptions tailored to user symptoms
- A robust, secure payment system with transparent revenue sharing
- User-friendly interfaces optimized for diverse populations
- A scalable platform suited for developing regions

This comprehensive system improves user trust, access, and healthcare quality beyond current solutions.

2. SOFTWARE DEVELOPMENT LIFE CYCLE

2.1 Process Model: Extreme Programming (XP)

The proposal outlines MediTeam Assist, a symptom-based, doctor-guided prescription system aimed at addressing the lack of accessible and safe healthcare, especially in rural and underserved areas. The platform connects patients with real doctors for verified prescriptions based on symptoms, offering 24/7 support, secure payments, and reliable medical advice.

Now, let's analyze why Extreme Programming (XP) is the most suitable development methodology for this project and how it compares to other models like the Waterfall, Prototyping, V-Model, and Incremental Model.

XP Model Overview:

Extreme Programming (XP) is an Agile methodology that emphasizes customer involvement, rapid feedback, short development cycles, and continuous testing. XP promotes strong communication between developers and stakeholders and encourages frequent releases of functional software to ensure the end product meets evolving user needs.

Comparison with Other Models:

Waterfall Model:

Not Used Because: Waterfall is linear and rigid. MediTeam Assist requires flexibility and continuous updates as medical data and user needs evolve. Waterfall would not accommodate iterative prescription or symptom updates efficiently.

Prototyping Model:

Not Used Because: While useful for gathering early feedback, prototyping may delay real implementation and lead to design changes without a strong testing backbone, which could risk the platform's reliability in a healthcare context.

V-Model:

Not Used Because: V-Model emphasizes formal testing at each stage. While rigorous, it's not adaptive to changing requirements and may slow down delivery compared to XP's continuous testing and deployment.

Incremental Model:

Not Used Because: Though it allows partial system delivery, it doesn't emphasize developer-customer interaction as strongly as XP. Also, it lacks XP's focus on engineering best practices like TDD and pair programming, which are critical in our safety-sensitive system.

Advantages of XP for MediTeam Assist

Short Iterations: Quick delivery of usable features like symptom input, doctor chat, or payment modules.

Constant Feedback: Real-time updates from doctors and patients improve system reliability and trust.

Strong Testing Culture: Prevents release of faulty prescriptions or symptom logic.

Better Collaboration: Encourages teamwork and shared responsibility through practices like pair programming and daily stand-ups.

Adaptive Planning: Easily incorporates changes based on real medical feedback or new policy changes.

Disadvantages of XP:

Not Suitable for Large, Distributed Teams: XP is best for small teams with close collaboration, which is suitable in this case.

Requires High Commitment from Stakeholders: Doctors and users must be actively involved, which we have accounted for.

Less Emphasis on Documentation: However, critical modules like prescriptions and payments will still be documented for legal and security reasons.

XP Process Model to MediTeam Assist

Exploration Phase:

Patients and doctors help define requirements via user stories (e.g., "As a patient, I want to input symptoms and get a verified prescription").

Planning Phase:

User stories are estimated and selected for the upcoming iteration based on priority and feasibility.

Iteration to Release Phase:

Pair programming, TDD, and continuous integration ensure high quality.

Features like symptom search, doctor chat, prescription creation, and payment processing are built in short cycles (1-2 weeks).

Regular feedback is collected and implemented.

Productionizing Phase:

A small release (e.g., MVP with core features) is deployed for real users.

Doctors verify prescription workflows before full rollout.

Maintenance Phase:

Bug fixes and feature enhancements are made continuously.

The system evolves based on usage patterns and medical updates.

Death Phase:

Once all essential features are stable and tested, the final product is launched.

Long-term maintenance continues for data integrity, updates, and support.

Conclusion:

Extreme Programming (XP) is the most suitable process model for MediTeam Assist because it:

Supports rapid and flexible development

Encourages continuous collaboration with patients and doctors

Promotes high code quality and early issue detection

Delivers working software in short, manageable cycles

Its focus on real-time feedback, test-driven development, and fast iteration perfectly aligns with the critical and evolving nature of healthcare technology.

2.2 Project Role Identification and Responsibilities

Customer

- Acts as the voice of the end-user or stakeholder.
- Writes and prioritizes **user stories** (requirements).
- Provides clarification on features and feedback throughout the iterations.
- Makes business decisions such as feature scope, deadlines, and priorities.
- Tests the product regularly to confirm it meets business needs.

Programmer

- Writes clean, efficient, and test-driven code.
- Participates in **pair programming** and **continuous integration**.
- Implements the user stories provided by the Customer.
- Refactors code frequently to improve design.
- Collaborates closely with the Tester and other team members.

Tester

- Works alongside programmers to create and automate tests.
- Ensures that every feature meets its acceptance criteria.
- Performs **unit, integration, and acceptance testing**.
- Identifies bugs and helps maintain a reliable codebase.
- Provides test reports and helps with quality assurance.

Tracker

- Monitors the team's estimation accuracy (e.g., effort and time).
- Provides feedback to improve future estimations.
- Tracks progress during each iteration.
- Evaluates whether goals can be achieved within current time and resource constraints.
- Suggests adjustments if goals seem unreachable.

Coach

- Oversees the entire XP process and ensures adherence to its principles.
- Guides the team in following XP practices correctly.
- Helps resolve process-related issues and promotes collaboration.
- Possesses deep knowledge of XP to mentor other team members.

Consultant

- Provides specialized technical expertise not present within the team.
- Offers advice or hands-on help with complex or unfamiliar technologies.
- Typically brought in temporarily for expert support.

Manager (Big Boss)

- Holds the highest decision-making authority in the project.
- Sets overall direction, goals, and priorities.
- Allocates resources and approves major changes or deliverables.

3. REQUIREMENT ANALYSIS

3.1 Functional Requirements

1. The system allows patients to register and log in using email/password or social login.
2. The system shall allow patients to input their symptoms through a text or form-based interface.
3. The system shall match symptoms with an existing database of doctor-verified prescriptions.
4. The system shall route unmatched symptoms to licensed doctors for **personalized consultation**.
5. The system allows **real-time online consultation** via chat, audio, or video between doctors and patients.
6. The system enable doctors to create, approve, and manage prescriptions after consultation.
7. The system shall securely store patient medical history, prescriptions, and consultation records.
8. The system shall integrate a digital payment gateway for consultation fees and revenue sharing.
9. The system shall provide a role-based dashboard (Patient, Doctor, Admin).
10. The system shall notify patients about new prescriptions, medication reminders, consultation schedules, and follow-ups.
11. The system shall ensure patients cannot edit or delete doctor-issued prescriptions.
12. The system shall allow admins to manage doctors, patients, consultations, and overall platform activities.
13. The system shall support **24/7 availability** for emergency online consultations.
14. The system shall allow doctors to set their availability status (online/offline) and consultation timings.
15. The system shall provide patients with a rating and feedback option after consultation.

3.2 Non-Functional Requirements

1. **Security:** The system shall use AES-256 encryption and secure authentication (JWT/Firebase Auth) to protect patient and doctor data.
2. **Performance:** The system shall handle at least 50,000 concurrent users and consultations without degradation.
3. **Availability:** The system shall ensure 99.9% uptime for uninterrupted online medical support.
4. **Scalability:** The system shall be cloud-based and horizontally scalable to support increasing patients, doctors, and consultations.
5. **Usability:** The system shall provide an intuitive, easy-to-use interface optimized for both patients and doctors (mobile & web).

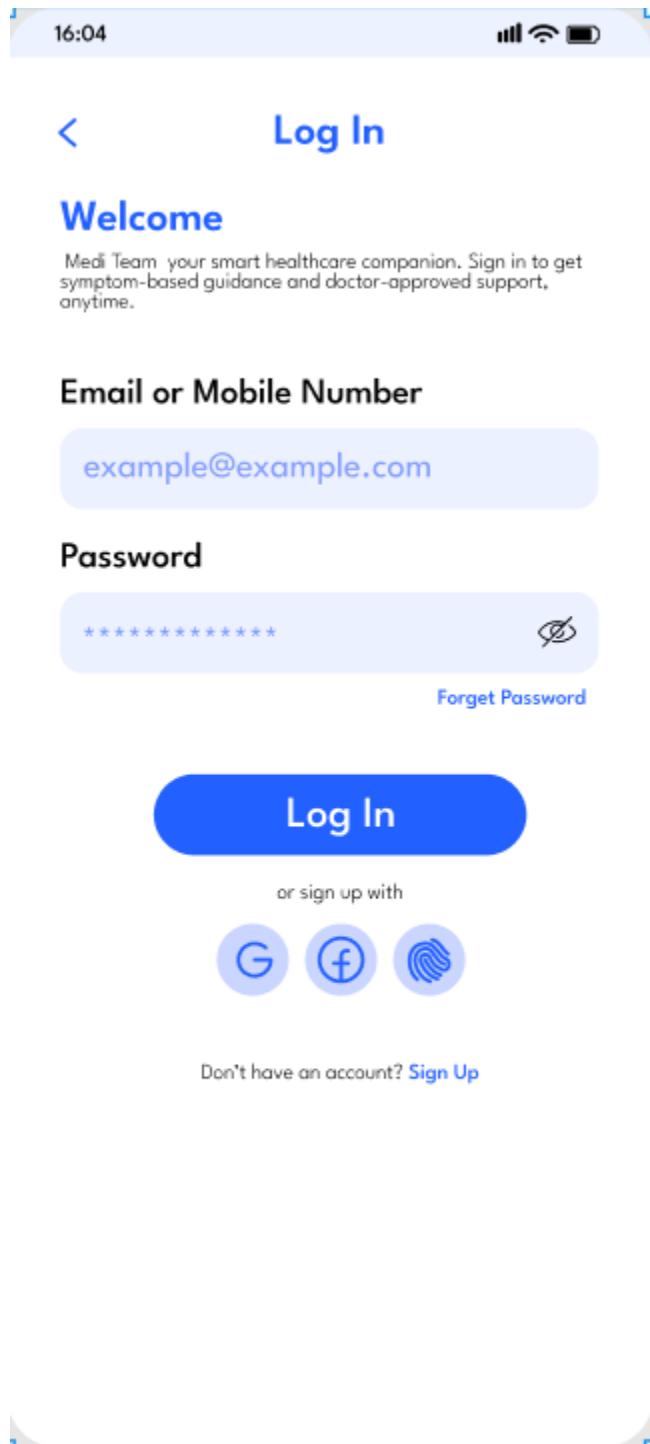
6. **Accessibility:** The system shall comply with WCAG 2.1 standards to support users with disabilities.
7. **Compliance:** The system shall comply with healthcare regulations (HIPAA/GDPR equivalent for data protection).
8. **Auditability:** The system shall maintain secure logs of all consultations, prescriptions, and payments for auditing.
9. **Maintainability:** The system shall follow modular and API-driven architecture for easy updates and third-party integration.
10. **Disaster Recovery:** The system shall perform automated daily backups and enable recovery within 30 minutes in case of system failure.

4. Prototype Designing (Figma)

1. Splash Screen



2. Log In



3. Sign In

16:04

< **New Account**

Full name
example@example.com

Password

Email
example@example.com

Mobile Number
example@example.com

Date Of Birth
DD / MM /YYY

By continuing, you agree to
[Terms of Use](#) and [Privacy Policy](#).

Sign Up

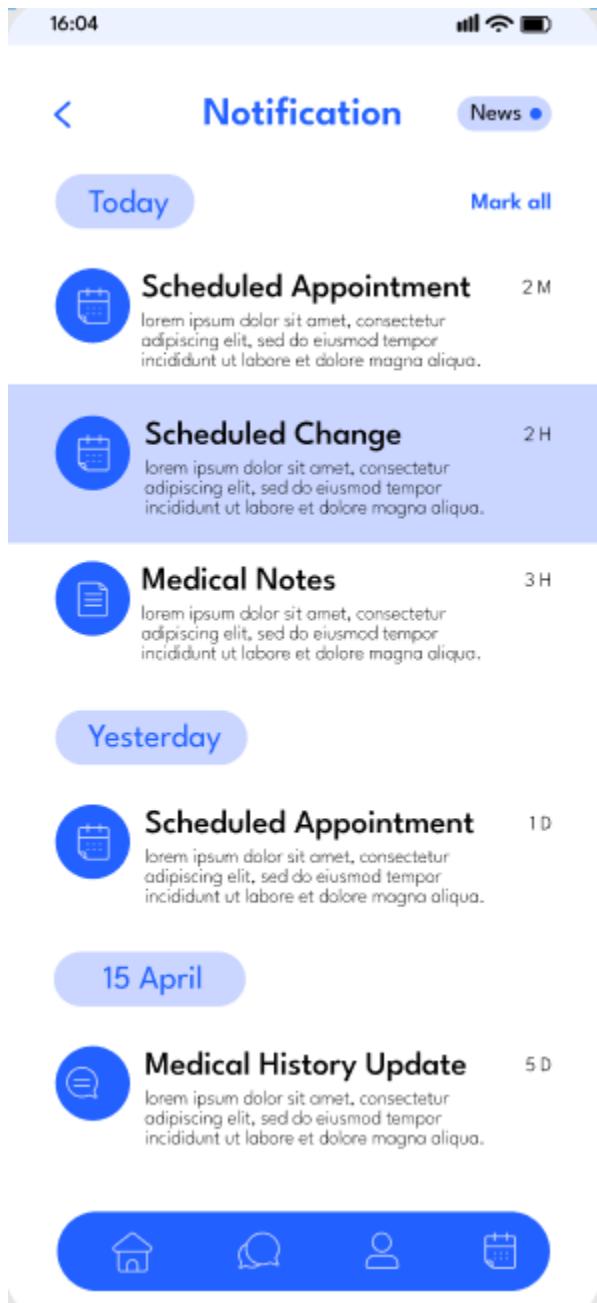
or sign up with

already have an account? [Log in](#)

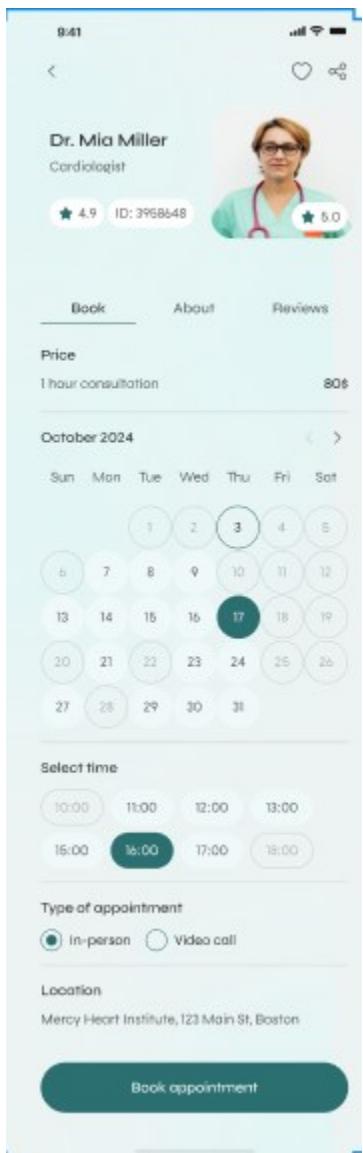
4. Home page



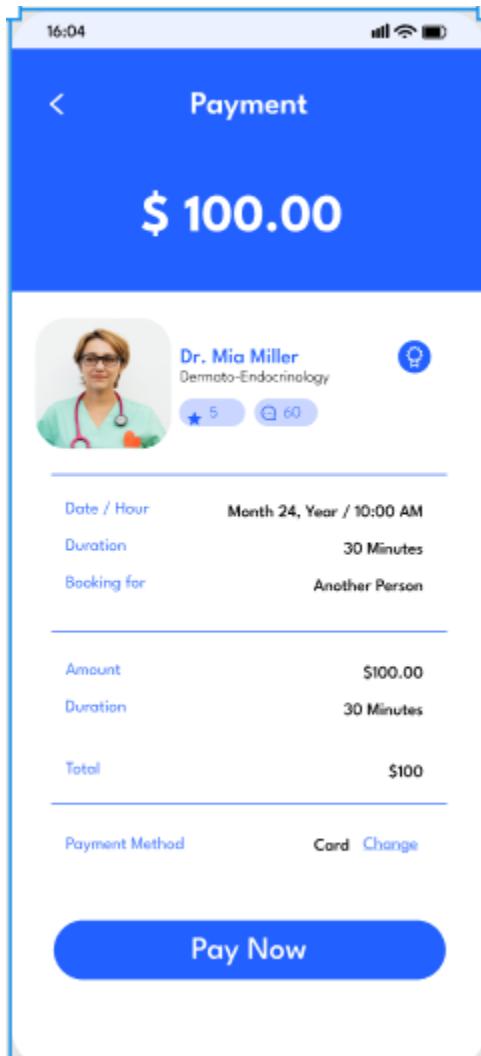
5. Notification



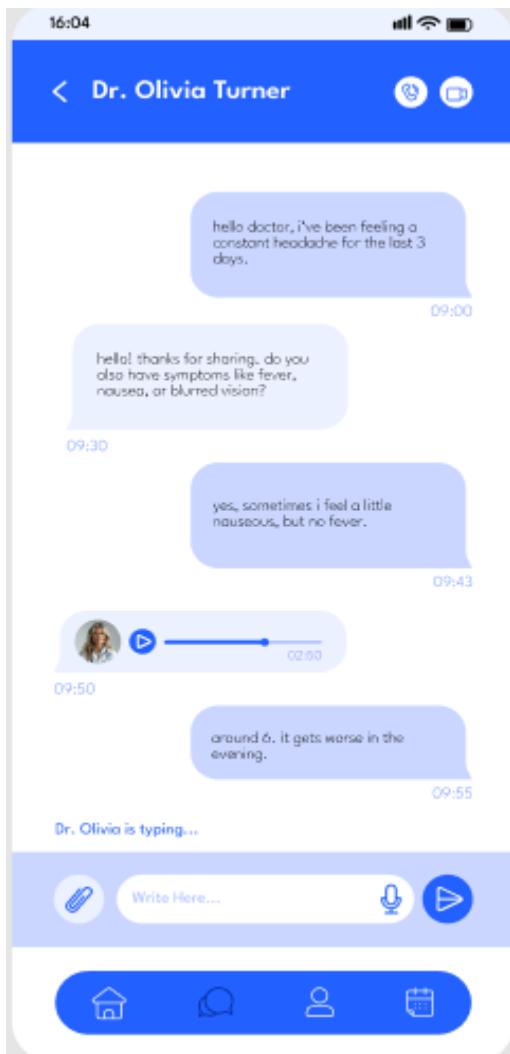
6. Doctor profile/book



7. Payment summary

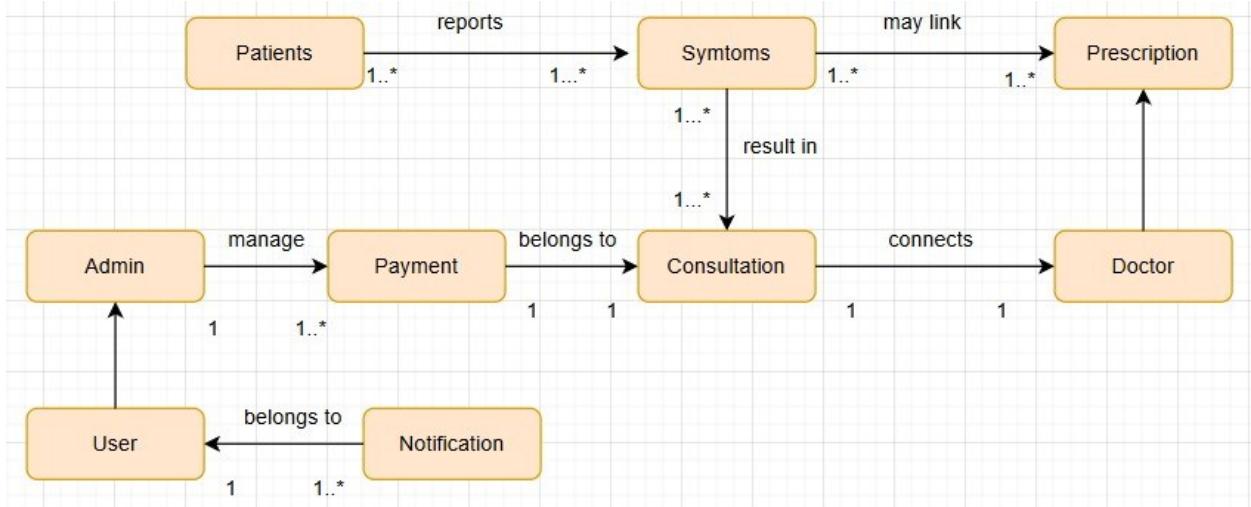


8. Message With Doctor

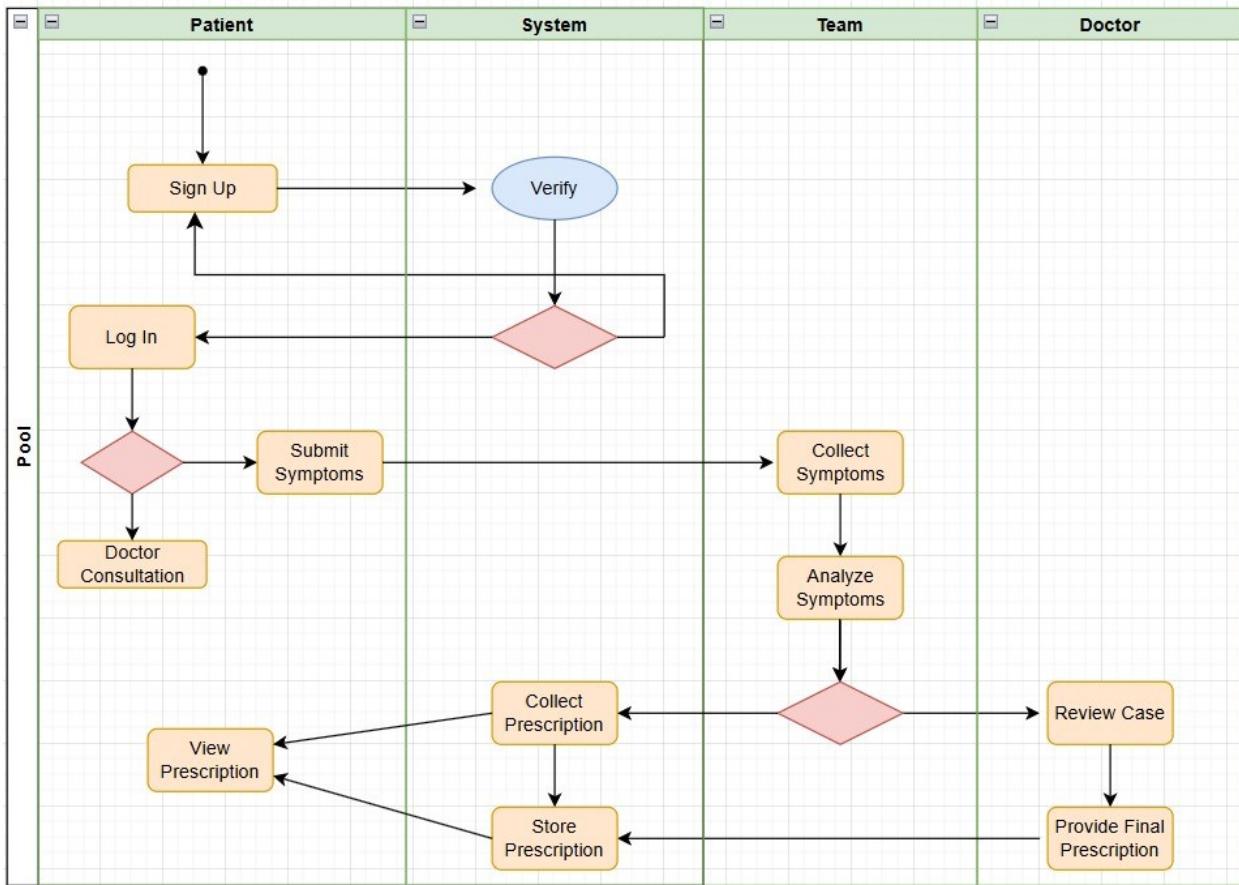


5. Diagram

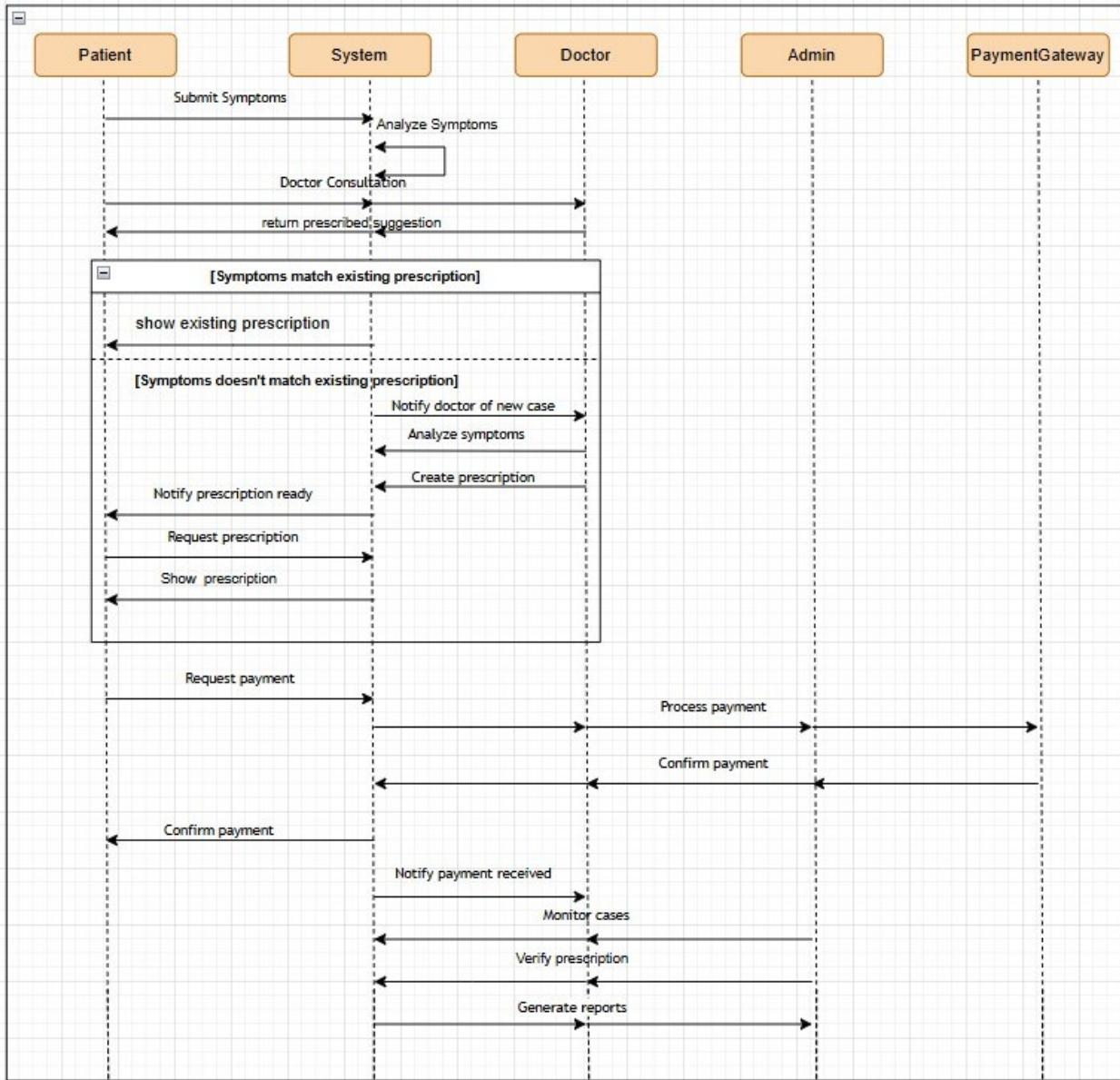
5.1 Class Diagram



5.2 Activity Diagram



5.3 Sequence Diagram



6. Effort Estimation

1. Constructive Cost Model (COCOMO):

Let's assume Source Line of Code is 20000.

So, effort will be, PM = Coefficient<Effort Factor>*(SLOC/1000)P = $3 * (20000/1000)^{1.05} = 85.955$

Development time, DM = $2.50 * (PM)T = 2.50 * (85.955)^{0.35} = 11.88 = 12(\text{week})$

Required number of people, ST = PM/DM = $85.955/11.88 = 7.23 = 7$

That means we need to work for =12 weeks. (Total weeks in 3 months as DM is 12 week)

2. Scheduling Grant Chart



3. EVA Analysis

4. Project Gantt Summary (Planned Effort)

Task	Phase / Task	Start Date	End Date	Days
1	Conduct And Analysis	01-Jul-25	03-Jul-25	3
2	Describe Project Scope	04-Jul-25	06-Jul-25	3
3	Client meeting & requirement gathering	07-Jul-25	11-Jul-25	4
4	Define target audience	12-Jul-25	15-Jul-25	4
5	Create sitemap & website structure	16-Jul-25	20-Jul-25	4
6	Diagram Design	21-Jul-25	24-Jul-25	4
7	UI/UX Design	25-Jul-25	03-Aug-25	10
8	Frontend Development	04-Aug-25	17-Aug-25	14
9	Backend Development	18-Aug-25	03-Sep-25	14
10	Testing & Bug Fixing	04-Sep-25	10-Sep-25	7
11	Deployment & Final Review	11-Sep-25	17-Sep-25	7
12	Risk Handling	18-Sep-25	25-Sep-25	8

5. Planned vs Actual Efforts

Task	Planned Effort (days)	Actual Effort (days)
1	3	3
2	3	3
3	4	4
4	4	4
5	4	4
6	4	4
7	10	10
8	14	14
9	14	14
10	7	7
11	7	7
12	8	8

6. EVA Metrics

Metric	Formula	Result	Interpretation
SPI (Schedule Performance Index)	BCWP / BCWS	$82 / 82 = 1.00$	On Schedule
SV (Schedule Variance)	BCWP – BCWS	0	No schedule variance
CPI (Cost Performance Index)	BCWP / ACWP	$82 / 82 = 1.00$	On Budget
CV (Cost Variance)	BCWP – ACWP	0	No cost variance

7. Progress Evaluation

Metric	Formula	Result
% Schedule Completed	BCWS / BCWP	$82 / 95 = 86.31\%$
% Work Completed	BCWP / BAC	$82 / 95 = 86.31\%$

8. Risk Management

Risk ID	Risk Description	Probability	Impact	Risk Level	Mitigation Strategy
R1	Server downtime during peak consultation hours	High	High	Critical	Use cloud-based auto-scaling servers with backup instances and load balancing
R2	User fails OTP verification during registration	Medium	High	Major	Allow retry attempts, support both SMS and email OTP, provide helpdesk fallback
R3	Duplicate or incorrect symptom entries	Medium	Medium	Moderate	Validate symptom inputs, provide guided symptom checklist, and use AI-based suggestion
R4	Delay in real-time doctor consultation (long queue)	High	Medium	Major	Implement queue management, notify users of wait time, and increase doctor availability during peak hours
R5	User confusion or lack of awareness in using app features	High	Medium	Major	Provide in-app tutorials, multi-language support (Bangla/English), and FAQs
R6	Payment failure during consultation fee transaction	Medium	High	Major	Support multiple payment gateways, auto-retry failed payments, and provide manual payment confirmation
R7	System overload with too many concurrent users	Medium	High	Major	Conduct load testing, use horizontal scaling, and optimize database queries
R8	Legal/regulatory compliance issues (health data laws)	Low	High	Major	Ensure compliance with GDPR/local health data laws, maintain documentation, and regular audits

Rubric for Project Assessment (CO3)

Criteria	Marks distribution (Max 3X5= 15)				Acquired Marks
	Inadequate (1-2)	Satisfactory (3)	Good (4)	Excellent (5)	
Selection of Software Engineering Models	Does not articulate a position or argument of choosing appropriate model. Does not present any evidence to support the arguments for the choice of the model	Articulates a position or argument for choosing models that is unfocused or ambiguous. Presents incomplete/vague evidence to support argument for model choice	Articulates a position or argument of choosing models that is limited in scope. Does not present enough evidence to support the argument for the choice of the model	Clearly articulates a position or argument for the choosing software engineering models. Presents sufficient amount of evidence to support argument for the model selection	
Role identification and Responsibility Allocation	The project has poor project management plans for identifying roles and assigning the responsibilities	Identify few roles in the project management where some of the roles are left alone with any project responsibilities	Identify most of the roles in the project management and assign their responsibilities	Well planned project with proper role identification and responsibility allocation in the project management activities	
Impact identification					

Formatting and Submission	Project report is not complete and Several errors in spelling and grammar. Present a Confusing organization of concepts, supporting arguments, and real-life example. Sentences rambling, and details are repeated.	Some errors in spelling and grammar. Some problems of organizing the answer in a logical order of defining, elaborating, and providing real-life examples.	Few errors in spelling and grammar. Presents most of the details in a logical flow of organization in definition, details, and example.	Project report is complete and No errors in spelling and grammar. Consistently presents a logical and effective organization of definition, details, and real-life example of the topic.	
Acquired marks:					
CO Pass / Fail:					

Rubric for Project Assessment (CO4)

Marks Distribution (Maximum 3X5=15)					
Marking Criteria	Inadequate (1-2)	Satisfactory (3)	Good (4)	Excellent (5)	Acquired Marks
Project Planning	No background information regarding the project is given; project goals and benefits are missing.	Insufficient background information is given; project goals and benefits are poorly stated	Sufficient background information is given; the purpose and goals of the project are explained.	Thorough and relevant background information is given; project goals are clear and easy to identify.	
Effort Estimation and Scheduling	Student vaguely discuss the impact of societal, health, safety, legal and cultural issues in their project	Student provided with partial relevance to the impact of societal, health, safety, legal and cultural issues in their project	Student fairly provided the analysis to the impact of societal, health, safety, legal and cultural issues in their project	Student comprehensively provided the analysis to the impact of societal, health, safety, legal and cultural issues in their project	

		cultural issues in their project			
Risk Management	Ambiguous representative example.	Partially identify / indicate towards real-life example.	Real-life example is fairly connected towards the definition.	Comprehensively defend with real life example.	
Acquired Marks:					
CO Pass / Fail:					