



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 24-25		Section: K	Group Number: 7	
SL	Student Name	Student ID	Contribution (CO3+CO4)	Individual Marks
1	HRIDOV SAHA	23-51060-1	25%	
2	TAFSIR RAHMAN	23-51061-1	25%	
3	MOST. SHUMAYA JAHAN	23-51070-1	25%	
4	GRONTHO CHANDRA ROY	23-51087-1	25%	

The project will be evaluated for the following Course Outcomes

CO3 (PO-g-1) <i>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</i>	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) <i>Apply engineering management principles and economic decision making to develop software engineering project management plan.</i>	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:

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, Medi-Team 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, Medi-Team 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 and Process Model Selection:

The objective of Medi-Team 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.

Medi-Team 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 work.

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.

Medi-Team 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.

Process Model: Extreme Programming (XP)

The proposal outlines Medi-Team 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 product meets evolving user needs.

Comparison with Other Models:

Waterfall Model:

Not Used Because: Waterfall is linear and rigid. Medi-Team 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 Medi-Team 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 Medi-Team 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.

1.3 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.
- Suggest 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 promote collaboration.
- Possesses deep knowledge of XP to mentor other team members.

Consultant

- Provides specialized technical expertise does not exist 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.

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

2.1 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 enables 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 should 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 should 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 allows 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.

2.2 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.

3. PROJECT ESTIMATION AND SCHEDULING

3.1 Effort and Cost Estimation:

Constructive Cost Model (COCOMO):

Let's assume Source Line of Code is 20000.

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

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

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 weeks)

3.2 Project Scheduling:

EVA Analysis:

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

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

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

Progress Evaluation:

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

Risk Management:

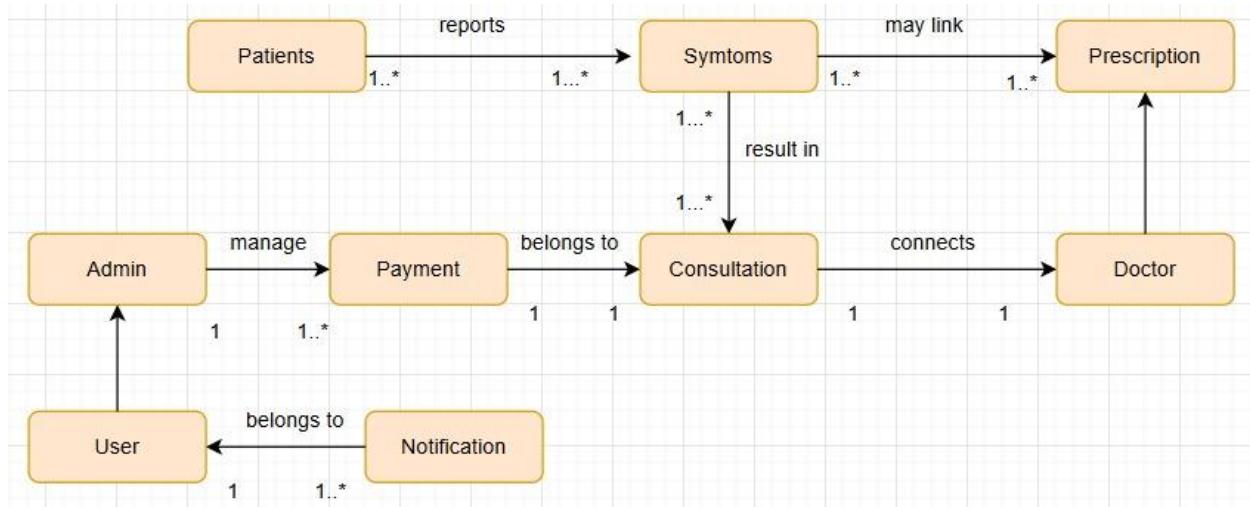
Risk ID	Risk Description	Probability	Impact	Risk Level	Mitigation Strategy
R1	Server downtime during peak consultation hours	High	High	Critical	Use cloud-based auto-scaling servers with backup instances and load balancing
R2	User fails OTP verification during registration	Medium	High	Major	Allow retry 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
-----------	---	-----	------	-------	--

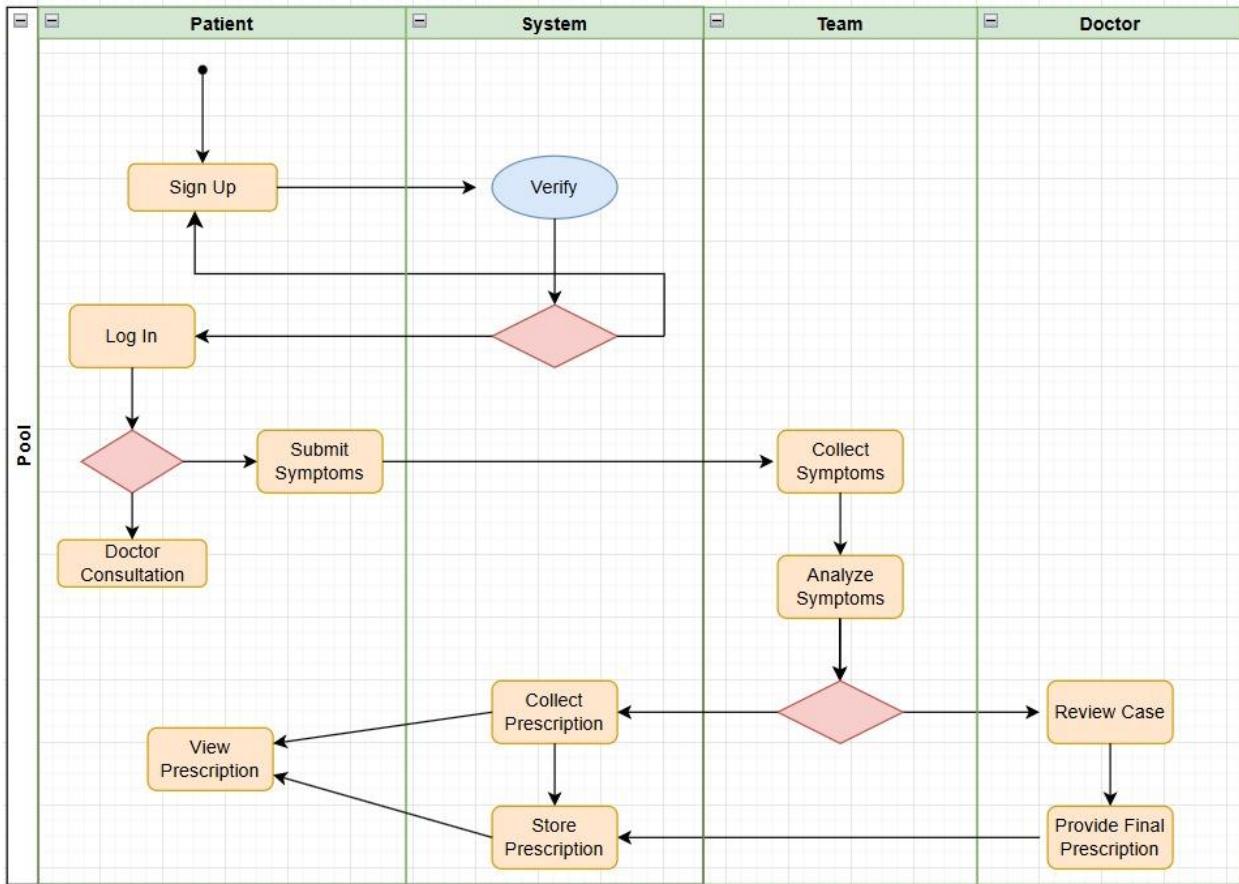
4. SOFTWARE DESIGN

4.1 System Design:

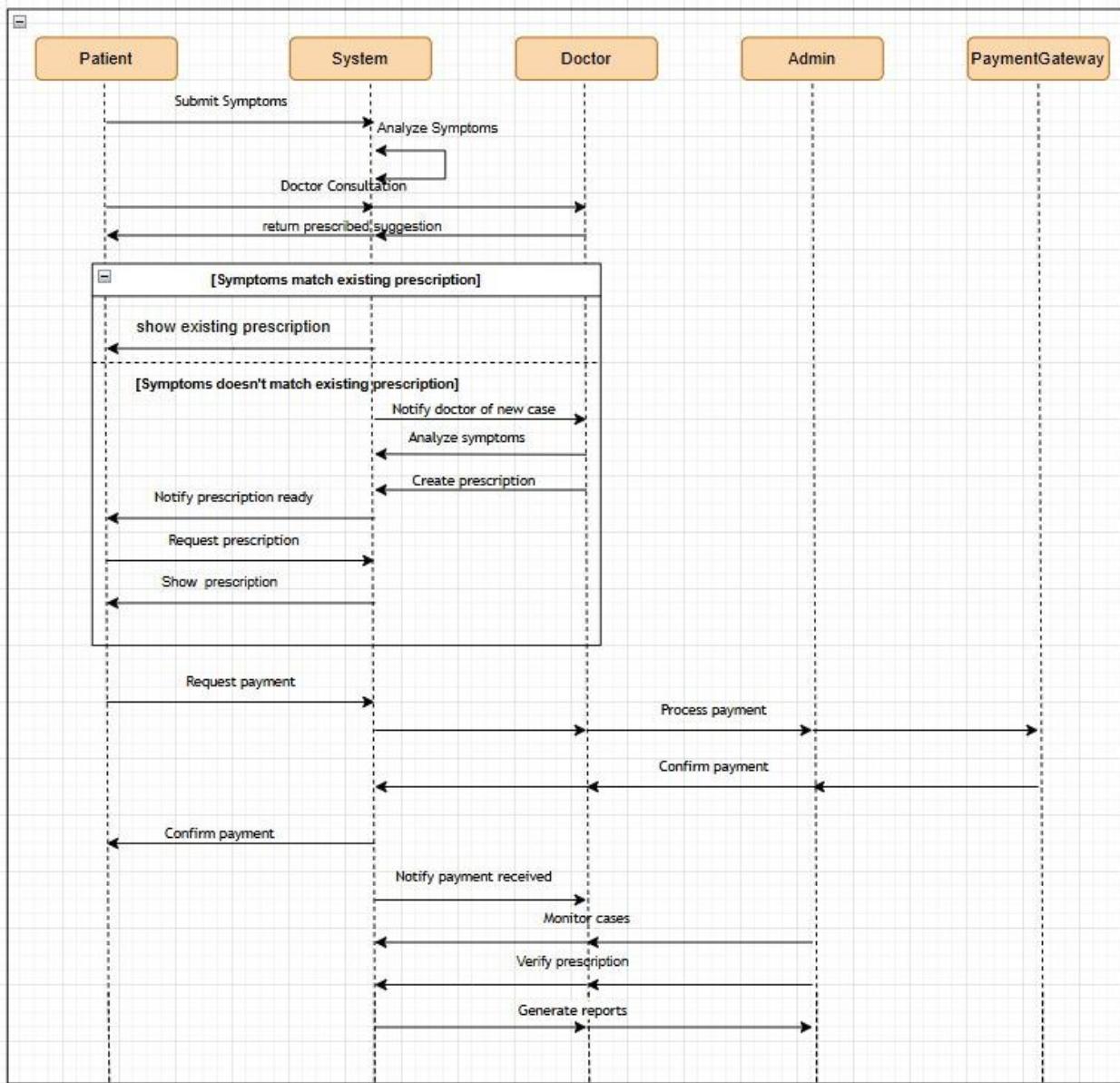
Class Diagram



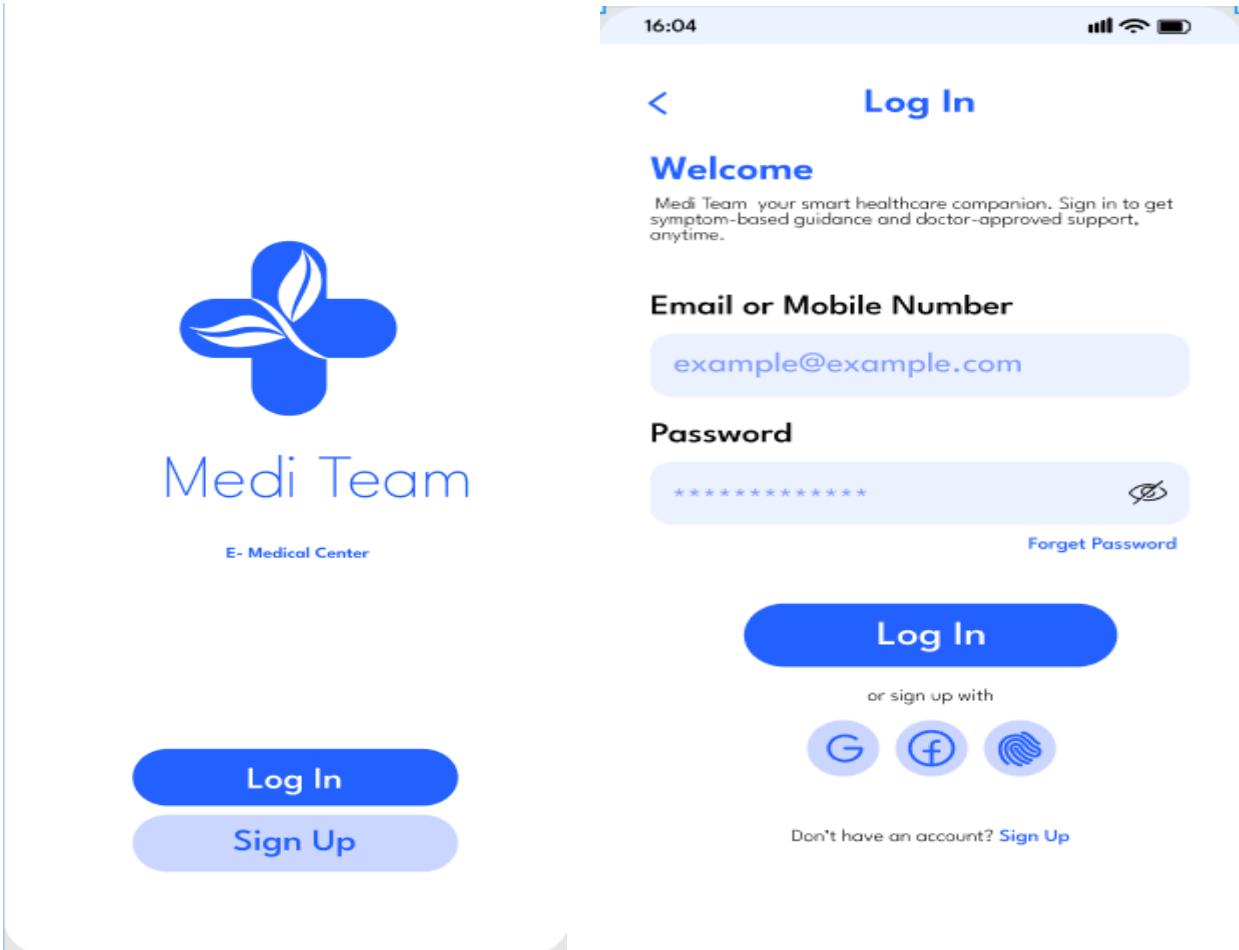
Activity Diagram



Sequence Diagram



4.2 UI / Wireframe Design



16:04

New Account

Full name

example@example.com

Password

***** 

Email

example@example.com

Mobile Number

example@example.com

Date Of Birth

DD / MM / YYYY

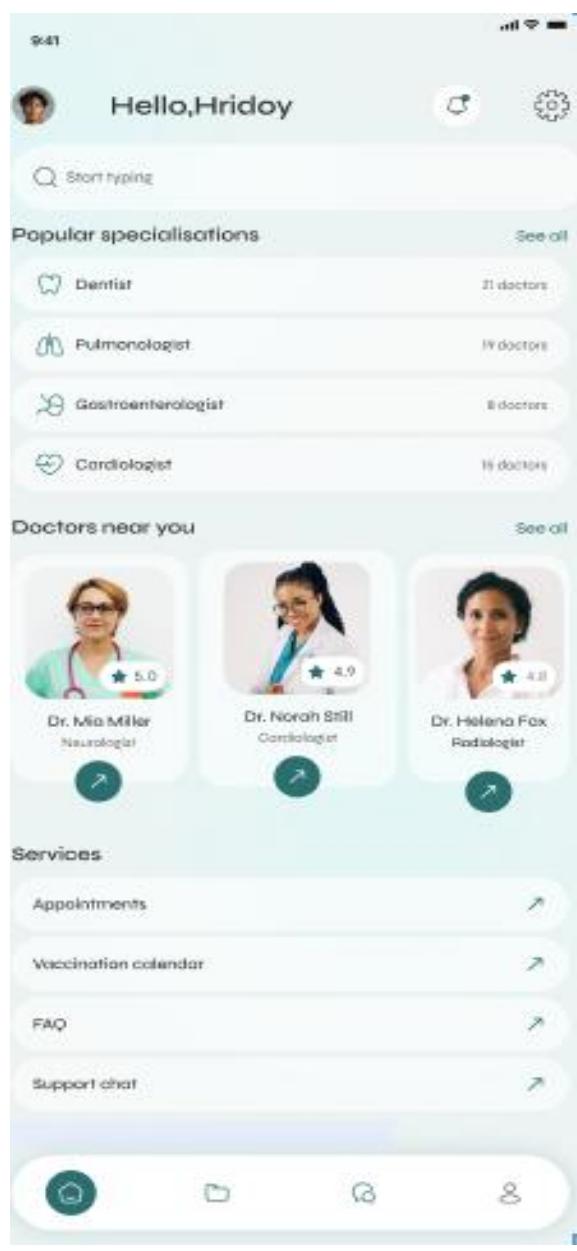
By continuing, you agree to
[Terms of Use](#) and [Privacy Policy](#).

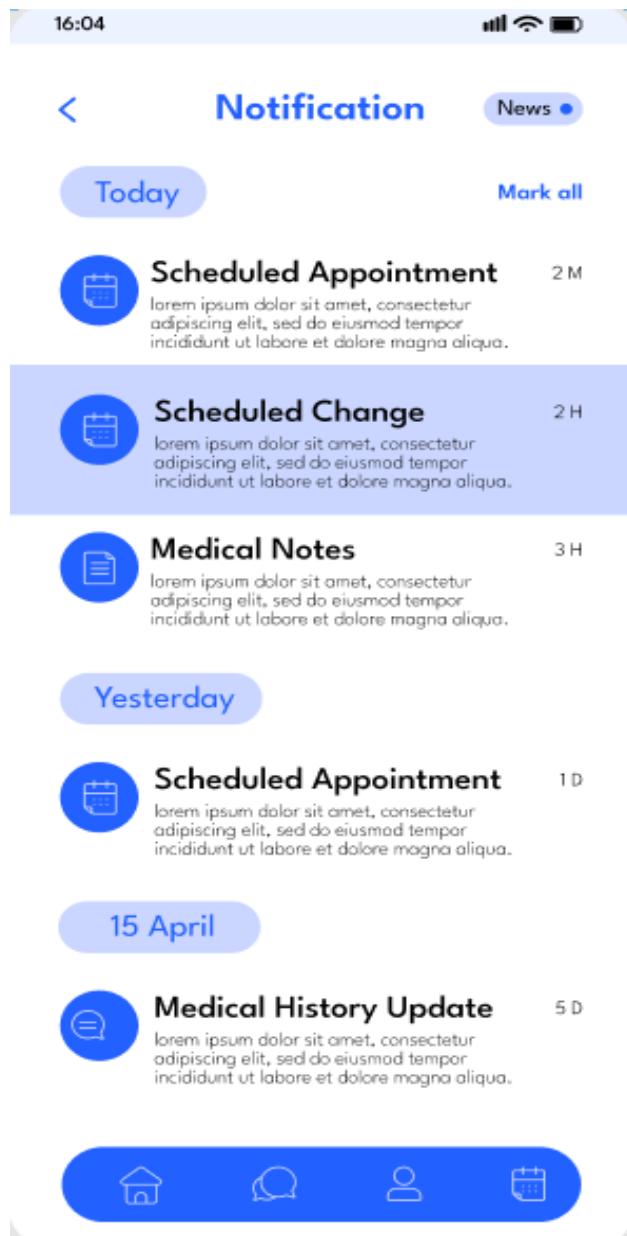
Sign Up

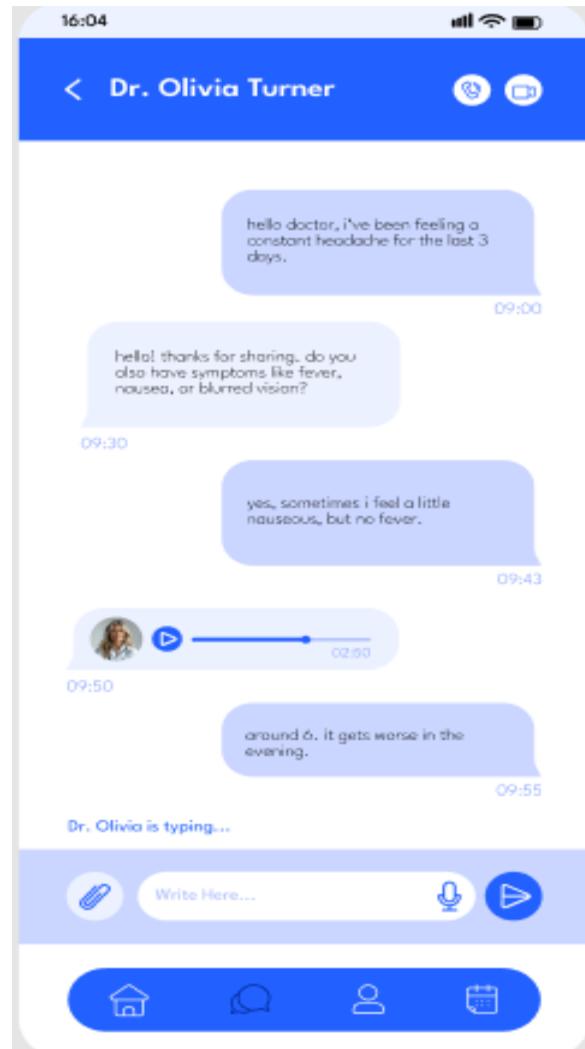
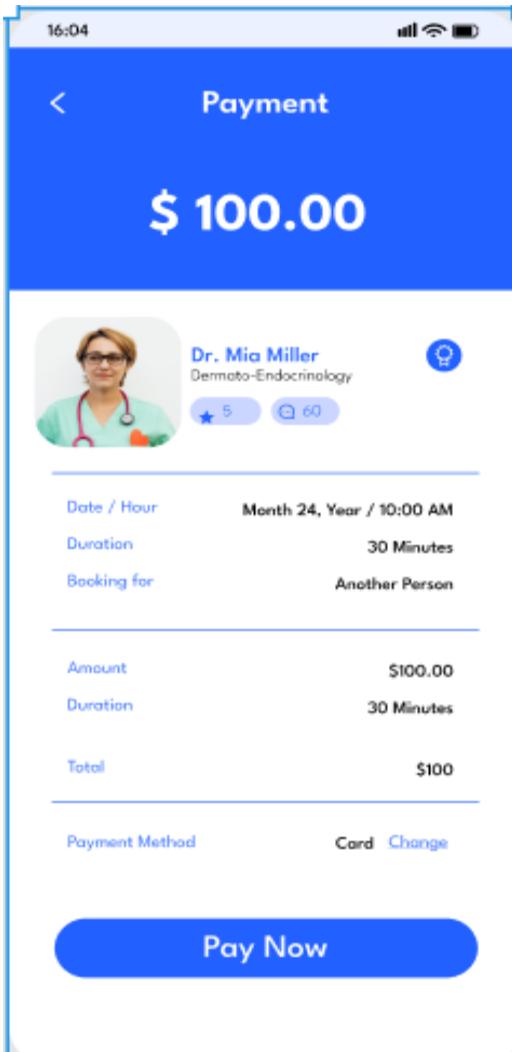
or sign up with

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







5. GIT WORKFLOW

GitHub Hridoy

The screenshot shows a GitHub repository page for 'E-medical'. The repository is public and has 3 branches and 0 tags. It contains two files: 'E-medical.docx' and 'Software-project.docx', both committed 3 weeks ago. There is a section to 'Add a README'.

File	Commit Message	Time
E-medical.docx	Change Docx	3 weeks ago
Software-project.docx	chore: initial commit with project file	3 weeks ago

About
No description, website, or topics provided.
Activity: 3 commits
0 stars
0 watching
0 forks

Releases
No releases published
Create a new release

Packages
No packages published
Publish your first package

The screenshot shows the 'Branches' page for the 'E-medical' repository. It lists three branches: 'main', 'feature/T-14', and 'hotfix/critical-readme-type'. The 'main' branch is the default. The 'feature/T-14' branch is active, showing it is 3 weeks old and 3 commits behind the main branch. The 'hotfix/critical-readme-type' branch is also 3 weeks old and 1 commit ahead of the main branch. A 'New branch' button is visible at the top right.

Branch	Updated	Check status	Behind	Ahead	Pull request
main	3 weeks ago	Default			
feature/T-14	3 weeks ago		3 2		
hotfix/critical-readme-type	3 weeks ago		1 0	1 0	

```
C:\Windows\System32\cmd.e  X  +  ▾

Microsoft Windows [Version 10.0.26100.6584]
(c) Microsoft Corporation. All rights reserved.

D:\8TH Semester>git clone https://github.com/hridoy-saha1/E-medical.git
Cloning into 'E-medical'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 15 (delta 2), reused 13 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (15/15), 660.00 KiB | 1.21 MiB/s
Receiving objects: 100% (15/15), 1.02 MiB | 1.43 MiB/s, done.
Resolving deltas: 100% (2/2), done.

D:\8TH Semester>
```

GitHub Grontho

 Medi-Team-Assist Public

[Pin](#) [Watch 0](#) [Fork 0](#) [Star 0](#)

[main](#) [4 Branches](#) [0 Tags](#) [t](#) [Add file](#) [Code](#)

 **Grontho-Roy-Nirob** Merge pull request #1 from Grontho-Roy-Nirob/critical-readm... 7a0d86b · 3 weeks ago [3 Commits](#)

 Medi-Team Assist_23-51087-1.docx	Initial Commit	3 weeks ago
 Medi-Team Assist_23-51087-1.pdf	Initial Commit	3 weeks ago
 README.md	Implementation	3 weeks ago

About
No description, website, or topics provided.

 Activity
 0 stars
 0 watching
 0 forks

Releases
No releases published
[Create a new release](#)

Packages
No packages published
[Publish your first package](#)

 © 2025 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Community](#) [Docs](#) [Contact](#) [Manage cookies](#) [Do not share my personal information](#)

Branches

New branch

Overview Yours Active Stale All

Search branches...

Default

Branch	Updated	Check status	Behind	Ahead	Pull request
main	3 weeks ago	Default			

Your branches

Branch	Updated	Check status	Behind	Ahead	Pull request
feature/T-14	3 weeks ago	0 1	#2		
dev	3 weeks ago	0 0			
hotfix/critical-readme-typo	3 weeks ago	1 0	#1		

Active branches

Branch	Updated	Check status	Behind	Ahead	Pull request
feature/T-14	3 weeks ago	0 1	#2		
dev	3 weeks ago	0 0			
hotfix/critical-readme-typo	3 weeks ago	1 0	#1		

```
HP@LAPTOP-8KHCK6S9 MINGW64 /d/CSE DEPT DOCUMENT/8th Semester/SOFTWARE ENGINEERING[K]/Assignment (hotfix/critical-readme-typo)
$ git push -u origin hotfix/critical-readme-typo
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 283 bytes | 283.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'hotfix/critical-readme-typo' on GitHub by visiting:
remote:   https://github.com/Grontho-Roy-Nirob/Medi-Team-Assist/pull/new/hotfix/critical-readme-typo
remote:
To https://github.com/Grontho-Roy-Nirob/Medi-Team-Assist.git
 * [new branch]      hotfix/critical-readme-typo -> hotfix/critical-readme-typo
branch 'hotfix/critical-readme-typo' set up to track 'origin/hotfix/critical-readme-typo'.
```

```
HP@LAPTOP-8KHCK6S9 MINGW64 /d/CSE DEPT DOCUMENT/8th Semester/SOFTWARE ENGINEERING[K]/Assignment (hotfix/critical-readme-typo)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

HP@LAPTOP-8KHCK6S9 MINGW64 /d/CSE DEPT DOCUMENT/8th Semester/SOFTWARE ENGINEERING[K]/Assignment (main)
$ git pull origin main
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (1/1), 922 bytes | 153.00 KiB/s, done.
From https://github.com/Grontho-Roy-Nirob/Medi-Team-Assist
 * branch    main      -> FETCH_HEAD
   db774c2..7a0d86b  main      -> origin/main
Updating db774c2..7a0d86b
Fast-forward
 README.md | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
```

```

HP@LAPTOP-8KHCK6S9 MINGW64 /d/CSE DEPT DOCUMENT/8th Semester/SOFTWARE ENGINEERING[K]/Assignment (main)
● $ git ls-remote --heads origin
966a68f39a7b55b73214e9e94e05fa5b24bcfa9d      refs/heads/hotfix/critical-readme-typo
7a0d86b2b95ae5a75500e184e214fa2175cfb2f3      refs/heads/main

HP@LAPTOP-8KHCK6S9 MINGW64 /d/CSE DEPT DOCUMENT/8th Semester/SOFTWARE ENGINEERING[K]/Assignment (main)
● $ git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.

HP@LAPTOP-8KHCK6S9 MINGW64 /d/CSE DEPT DOCUMENT/8th Semester/SOFTWARE ENGINEERING[K]/Assignment (main)
● $ git pull origin main
From https://github.com/Grontho-Roy-Nirob/Medi-Team-Assist
 * branch            main       -> FETCH_HEAD
Already up to date.

HP@LAPTOP-8KHCK6S9 MINGW64 /d/CSE DEPT DOCUMENT/8th Semester/SOFTWARE ENGINEERING[K]/Assignment (main)
● $ git checkout -b dev
Switched to a new branch 'dev'

HP@LAPTOP-8KHCK6S9 MINGW64 /d/CSE DEPT DOCUMENT/8th Semester/SOFTWARE ENGINEERING[K]/Assignment (dev)
● $ git branch -d hotfix/critical-readme-typo
Deleted branch hotfix/critical-readme-typo (was 966a68f).

```

```

HP@LAPTOP-8KHCK6S9 MINGW64 /d/CSE DEPT DOCUMENT/8th Semester/SOFTWARE ENGINEERING[K]/Assignment (dev)
● $ git clone https://github.com/Grontho-Roy-Nirob/Medi-Team-Assist.git
Cloning into 'Medi-Team-Assist'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 8 (delta 2), reused 6 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (8/8), 5.25 MiB | 52.00 KiB/s, done.
Resolving deltas: 100% (2/2), done.

HP@LAPTOP-8KHCK6S9 MINGW64 /d/CSE DEPT DOCUMENT/8th Semester/SOFTWARE ENGINEERING[K]/Assignment (dev)
● $ cd Medi-Team-Assist

HP@LAPTOP-8KHCK6S9 MINGW64 /d/CSE DEPT DOCUMENT/8th Semester/SOFTWARE ENGINEERING[K]/Assignment/Medi-Team-Assist (main)
● $ git fetch origin

HP@LAPTOP-8KHCK6S9 MINGW64 /d/CSE DEPT DOCUMENT/8th Semester/SOFTWARE ENGINEERING[K]/Assignment/Medi-Team-Assist (main)
● $ git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.

HP@LAPTOP-8KHCK6S9 MINGW64 /d/CSE DEPT DOCUMENT/8th Semester/SOFTWARE ENGINEERING[K]/Assignment/Medi-Team-Assist (main)
● $ git pull origin main
From https://github.com/Grontho-Roy-Nirob/Medi-Team-Assist
 * branch            main       -> FETCH_HEAD
Already up to date.

```

6. TESTING

Project Brief Introduction:

Medi-Team Assist is a digital healthcare platform designed to connect patients with doctors for symptom-based, verified prescriptions. It offers 24/7 support, secure payments, and role-based dashboards for patients, doctors, and administrators.

Testing Scope:

This test plan focuses exclusively on **Black-box Testing** to validate the external behaviour of the system against its functional requirements, without considering the internal code structure.

Goals of Black-Box Testing:

- Ensure all **functional requirements** work correctly from an end-user perspective.
- Detect errors in the categories of: incorrect/missing functions, interface errors, external database access, performance issues, and initialization/termination errors.
- Verify that the system is secure, reliable, and user-friendly for patients, doctors, and admins.

People Involved in Testing:

- **1 QA Lead / Test Engineer:** Prepares black-box test cases, manages test execution, and reports defects.
- **1 Project Manager:** Oversees the testing process and ensures alignment with project goals.
- **2 End-User Representatives (e.g., a mock Patient and a mock Doctor):** Perform User Acceptance Testing (UAT) based on the black-box test cases to validate real-world usability.

Black-Box Testing Strategy for Medi-Team Assist:

The testing will focus on the input/output behaviour of the system's key features as derived from the functional requirements.

1. Authentication & Authorization

- **Focus:** Incorrect functions, interface errors, initialization errors.
- **Example Test Cases:**
 - Verify that a patient can log in successfully with valid credentials.
 - Verify that login fails with an incorrect password and a clear error message is displayed.
 - Verify that a logged-in Doctor is redirected to the Doctor Dashboard and not the Patient Dashboard (role-based access).

- Verify the "Logout" function correctly terminates the session and requires a new login to access protected pages.

2. Symptom Input & Prescription Matching

- **Focus:** Incorrect/missing functions, errors in database access, behavior errors.
- **Example Test Cases:**
 - Submit a set of common symptoms (e.g., fever, cough) and verify the system returns a matching, doctor-verified prescription from the database.
 - Submit a set of unique or complex symptoms and verify the system correctly routes the case to the "Unmatched" queue for doctor consultation.
 - Verify that the symptom input form validates for empty submissions and provides an appropriate error message.

3. Doctor Consultation & Prescription Management

- **Focus:** Incorrect functions, interface errors, database access errors.
- **Example Test Cases:**
 - Verify a doctor can view their queue of unmatched symptom cases.
 - Verify a doctor can create a new prescription for a patient and that it is saved correctly to the patient's medical history.
 - Verify that a patient cannot edit or delete a doctor-issued prescription (as per FR-11).
 - Test the real-time chat interface: send a message as a patient and verify it appears in the doctor's chat window.

4. Payment Gateway Integration

- **Focus:** Interface errors, behaviour/performance errors, initialization/termination errors.
- **Example Test Cases:**
 - Initiate a consultation payment and verify the system correctly redirects to the secure payment gateway (e.g., SSLCommerz).
 - Simulate a successful payment and verify that both the platform and the doctor's ledger are updated with the correct revenue share.
 - Simulate a failed payment (e.g., insufficient funds) and verify the system handles it gracefully, notifying the user without booking the consultation.

- Verify that the payment process times out after a period of inactivity and terminates correctly.

5. Medical History & Notifications

- **Focus:** Incorrect functions, database access errors.
- **Example Test Cases:**
 - Verify a patient can view their complete history of consultations and prescriptions.
 - Verify that the system sends a notification (e.g., email, SMS) to a patient when a new prescription is issued.
 - Verify that medication reminders are triggered at the correct times.

6. Admin Dashboard & Platform Management

- **Focus:** Interface errors, incorrect functions.
- **Example Test Cases:**
 - Verify an admin can view lists of all doctors and patients.
 - Verify an admin can deactivate a doctor's account and that the doctor can no longer log in.

7. Risks & Mitigation (Black-Box Focus)

- **Risk:** Payment transaction fails during a critical consultation.
 - **Mitigation:** Perform extensive black box testing on the payment interface using mock gateway simulations for various failure scenarios (declined, timeout, network error).
- **Risk:** System fails to match symptoms correctly, leading to all cases being routed to doctors and causing long queues (performance error).
 - **Mitigation:** Test the symptom-matching function with a wide range of input data to verify its accuracy and behaviour under different conditions.
- **Risk:** Patient data is displayed to another user due to an interface or authorization error.
 - **Mitigation:** Rigorously test role-based access by trying to access patient-specific URLs and data while logged in as a different patient or a doctor.

Entry & Exit Criteria:

- **Entry Criteria:**
 - A stable build of the "MediTeam Assist" application is available for testing.

- The test environment with a mock database is ready.
- **Exit Criteria:**
 - All critical black-box test cases have been executed.
 - All defects related to incorrect or missing functions, critical interface errors, and payment failures have been resolved.
 - The application passes key user acceptance scenarios for patients and doctors.

Deliverables:

- **Black-Box Test Case Document** (Detailed steps for each scenario above).
- **Test Execution Report** (Log of passed/failed tests).
- **Bug Reports** from the black box testing cycle.
- **Final QA Sign-Off** for the functional aspects of the system.

Conclusion:

This black-box test plan ensures that "Medi-Team Assist" is validated against its core functional requirements from an end-user's perspective. By focusing on finding errors in functions, interfaces, database access, and system behaviour, we ensure the delivered platform is reliable, secure, and meets the critical needs of its users before deployment.

Test Case

Test Case 1: Verify Login Page Elements Load Correctly	
Project Name: mediteam Assist	Test Designed By: Grontho Chandra Roy
Test Case ID: MTA-LGIN-001	Test Designed date: 24/09/2025
Test Priority (Low, Medium, High): High	Test Executed By: Hridoy Saha
Module Name: Authentication	Test Execution Date: 01/10/2025

Test Title: Verify Login Page Elements Load Correctly							
Description: TO ENSURE THE LOGIN PAGE LOADS WITH ALL REQUIRED ELEMENTS							
Precondition: USER IS ON THE APPLICATION'S LANDING PAGE							
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)			
1. Open the application URL.	Application URL: https://mediteamassist.com	The page displays "Medi Team", "E-Medical Center", "Log In", and "Sign Up".	Valid	Pass			
2. Verify all elements are visible and correctly displayed.	Expected Elements: <ul style="list-style-type: none">• App Logo• Page Title ("Medi-Team Assist")• Email / Username field• Password field• "Log In" button• "Sign Up" link or button• Footer text (if any)	All text is clear and buttons/l	Valid	Pass			
Post Condition: User is on the main authentication page and can see the options to Log In or Sign Up.							

Test Case 2: User Login with Valid Credentials		
Project Name: Medi-Team Assist	Test Designed By: Grontho Chandra Roy	
Test Case ID: MTA-LGIN-002	Test Designed date: 24/09/2025	
Test Priority (Low, Medium, High): High	Test Executed By: Hridoy Saha	
Module Name: Authentication	Test Execution Date: 01/10/2025	

Test Title: User Login with Valid Credentials							
Description: USER NEEDS TO LOGIN TO THE APPLICATION USING VALID EMAIL AND PASSWORD							
Precondition: USER MUST HAVE A VALID REGISTERED EMAIL AND PASSWORD							
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)			
1. Enter Email or Mobile Number	user@example.com	Field accepts the input	Valid	Pass			
2. Enter Password	*****	Password field masks the input	Valid	Pass			
3. Click "Log In" Button	Press the button	User is successfully logged in and redirected to dashboard	Valid	Pass			

Post Condition: User is successfully logged in and redirected to the application dashboard/home page.

Test Case 3: User Registration with Valid Information	
Project Name: MediTeam Assist	Test Designed By: Grontho Chandra Roy
Test Case ID: MTA-REG-003	Test Designed date: 24/09/2025
Test Priority (Low, Medium, High): High	Test Executed By: Hridoy Saha
Module Name: Registration	Test Execution Date: 01/10/2025
Test Title: User Registration with Valid Information	

Description: VERIFY NEW USER CAN CREATE ACCOUNT USING VALID PERSONAL INFORMATION

Precondition: USER IS ON THE REGISTRATION PAGE AND HAS VALID PERSONAL DETAILS				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Enter Full Name	John Smith	Field accepts alphabetic input	Valid	Pass
2. Enter Password	SecurePass123!	Password field masks the input	Valid	Pass
3. Enter Email	john.smith@email.com	Field accepts valid email format	Valid	Pass
4. Enter Mobile Number	+8801712345678	Field accepts numeric input	Valid	Pass
5. Enter Date of Birth	15/05/1990	Date picker accepts valid date format	Valid	Pass
6. Click "Sign Up" Button	Press the button	Account created successfully and user redirected to verification/dashboard	Valid	Pass

Post Condition: New user account is created successfully, and user is redirected to application dashboard or verification page.

Test Case 4: Homepage Dashboard Navigation and Features							
Project Name: MediTeam Assist		Test Designed By: Grontho Chandra Roy					
Test Case ID: MTA-HOME-004		Test Designed date: 24/09/2025					
Test Priority (Low, Medium, High): High		Test Executed By: Hridoy Saha					
Module Name: Home Dashboard		Test Execution Date: 01/10/2025					
Test Title: Homepage Dashboard Navigation and Features							
Description: VERIFY HOMEPAGE DASHBOARD LOADS ALL SECTIONS AND NAVIGATION WORKS CORRECTLY							
Precondition: USER IS SUCCESSFULLY LOGGED INTO THE APPLICATION							
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)			
1. Verify dashboard loads after login	N/A	"Hello, [Day]" greeting displayed with all dashboard sections	Valid	Pass			
2. Check "Popular specialisations" section	Dentist, Pulmonologist, etc.	All specialisation cards displayed with doctor counts	Valid	Pass			
3. Click "See all" in specialisations	N/A	Navigates to full specialisations list page	Valid	Pass			
4. Verify "Doctors near you" section	Dr. Mia Miller, etc.	Doctor cards with names and locations displayed	Valid	Pass			

5. Check Services section	Appointments, Vaccination, etc.	All service options displayed and accessible	Valid	Pass
---------------------------	---------------------------------	--	-------	------

Post Condition: User can view all dashboard sections and navigate through different features successfully.

Test Case 5: Notification Center Display and Organization	
Project Name: MediTeam Assist	Test Designed By: Grontho Chandra Roy
Test Case ID: MTA-NOTIF-005	Test Designed date: 24/09/2025
Test Priority (Low, Medium, High): High	Test Executed By: Hridoy Saha
Module Name: Notification Center	Test Execution Date: 01/10/2025
Test Title: Notification Center Display and Organization	

Description: VERIFY NOTIFICATION CENTER CORRECTLY DISPLAYS AND ORGANIZES ALL USER NOTIFICATIONS

Precondition: USER IS SUCCESSFULLY LOGGED IN AND HAS EXISTING NOTIFICATIONS

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to Notification section	N/A	Notification page loads with "Notification" header	Valid	Pass
2. Verify Today's notifications	Scheduled Appointment	Today's notifications grouped under "Today" section with correct timestamp (2M)	Valid	Pass

3. Check older notifications	Yesterday, 15 April	Notifications properly grouped by date with correct time indicators (3H, 1D, 5D)	Valid	Pass
4. Verify notification content	Lorem ipsum text	Each notification displays relevant title, timestamp, and description text	Valid	Pass
5. Check notification categories	Appointment, Notes, Medical History	Different notification types are clearly distinguished	Valid	Pass

Post Condition: All notifications are properly displayed, organized by date, and show correct content with appropriate timestamps.

Test Case 6: Doctor Appointment Booking Process	
Project Name: MediTeam Assist	Test Designed By: Grontho Chandra Roy
Test Case ID: MTA-APPT-001	Test Designed date: 24/09/2025
Test Priority (Low, Medium, High): High	Test Executed By: Hridoy Saha
Module Name: Appointment Booking	Test Execution Date: 01/10/2025
Test Title: Doctor Appointment Booking Process	
Description: VERIFY USER CAN SUCCESSFULLY BOOK AN APPOINTMENT WITH A DOCTOR	
Precondition: USER IS LOGGED IN AND VIEWING DOCTOR'S PROFILE PAGE	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Verify doctor profile information	Dr. Mia Miller, Cardiologist	Doctor's name, specialty, rating (4.9), and ID displayed correctly	Valid	Pass
2. Select appointment date	October 2024 calendar	User can select an available date from the calendar	Valid	Pass
3. Select appointment time	11:00	Time slot can be selected from available options	Valid	Pass
4. Choose appointment type	Video call	Appointment type (In-person/Video call) can be selected	Valid	Pass
5. Verify location details	Mercy Heart Institute address	Location information displayed for in-person appointments	Valid	Pass
6. Click "Book appointment" button	N/A	System processes booking and shows confirmation	Valid	Pass

Post Condition: Appointment is successfully booked and user receives confirmation.

Test Case 7: Payment Processing for Appointment Booking

Project Name: MediTeam Assist	Test Designed By: Grontho Chandra Roy
Test Case ID: MTA-PAY-007	Test Designed date: 24/09/2025
Test Priority (Low, Medium, High): High	Test Executed By: Hridoy Saha
Module Name: Payment System	Test Execution Date: 01/10/2025
Test Title: Payment Processing for Appointment Booking	

Description: VERIFY PAYMENT CAN BE SUCCESSFULLY PROCESSED FOR DOCTOR APPOINTMENT

Precondition: USER HAS SELECTED APPOINTMENT DETAILS AND NAVIGATED TO PAYMENT PAGE

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Verify payment amount	\$100.00	Total amount displayed correctly	Valid	Pass
2. Check doctor information	Dr. Mia Miller, Dermato-Endocrinology	Doctor details and specialty displayed accurately	Valid	Pass
3. Verify appointment details	Month 24, Year/10:00 AM, 30 Minutes	Date, time and duration shown correctly	Valid	Pass

4. Confirm booking type	Another Person	Booking type displayed as selected	Valid	Pass
5. Select payment method	Card	Payment method can be selected/changed	Valid	Pass
6. Click "Pay Now" button	N/A	Payment processes successfully and shows confirmation	Valid	Pass

Post Condition: Payment is successfully processed and appointment is confirmed.

Test Case 8: Real-Time Chat with Doctor	
Project Name: MediTeam Assist	Test Designed By: Grontho Chandra Roy
Test Case ID: MTA-CHAT-001	Test Designed date: 24/09/2025
Test Priority (Low, Medium, High): High	Test Executed By: Hridoy Saha
Module Name: Messaging System	Test Execution Date: 01/10/2025
Test Title: Real-Time Chat with Doctor	

Description: VERIFY REAL-TIME CHAT FUNCTIONALITY BETWEEN PATIENT AND DOCTOR WORKS CORRECTLY

Precondition: USER HAS AN ACTIVE APPOINTMENT WITH DOCTOR AND IS IN CHAT INTERFACE

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Verify chat history displays	Previous messages	All previous messages show with correct timestamps (09:00, 09:30, etc.)	Valid	Pass

2. Check message formatting	Patient/Doctor messages	Messages are properly aligned and distinguishable	Valid	Pass
3. Verify typing indicator	"Dr. Olivia is typing..."	Typing indicator appears when doctor is composing message	Valid	Pass
4. Type new message in input field	"My headache pain level is 6"	Text input field accepts new message	Valid	Pass
5. Send new message	N/A	Message sends successfully and appears in chat history	Valid	Pass

Post Condition: Chat messages are sent and received in real-time with proper formatting and timestamps.

7. SOFTWARE PRODUCT METRICS

Function-Based Metrics (Function Points):

Function Point (FP) analysis measures the system's functionality from the user's perspective. Key system elements for Medi-Team Assist:

Component Type	Count	Weight	Adjusted FP
External Inputs (EI) – Patient registration, login, symptom entry, doctor prescription entry, payment submission	5	4	20
External Outputs (EO) – Prescription results, consultation summaries, payment receipts, notifications	4	5	20

External Inquiries (EQ) – Searching symptoms, viewing doctor availability, chat inquiry	3	4	12
Internal Logical Files (ILF) – User profiles, doctor data, prescription records, transaction records	4	10	40
External Interface Files (EIF) – Payment gateway, authentication API (Firebase/Google), cloud database	3	7	21

Object-Oriented and Class Metrics:

Metric	Description	Observation for Medi-Team Assist
WMC (Weighted Methods per Class)	Average number of methods per class	≈ 6 methods per class → Moderate complexity ; manageable for maintainability
DIT (Depth of Inheritance Tree)	Maximum inheritance depth	2 levels (e.g., User → Doctor / Patient) → Good abstraction without excessive complexity
NOC (Number of Children)	Average subclasses per parent class	User has 2 children (Doctor, Patient) → Low coupling and clear hierarchy
CBO (Coupling Between Objects)	Number of other classes a class depends on	≤ 4 → Well-structured, low interdependency
LCOM (Lack of Cohesion of Methods)	Measures relatedness of methods within a class	~0.3 → High cohesion , ensuring modular design

Operation-Oriented Metrics:

Metric	Description	Medi-Team Assist Observation
Average Operation Size (LOC/Method)	Total LOC / Total Methods	≈ 30–40 LOC per method (moderate)
Operation Complexity (Cyclomatic Complexity)	Count of decision paths	Avg. CC ≈ 4 → Moderate , acceptable for maintainability
Average Parameters per Operation	Measures data coupling	≈ 2 parameters per function (e.g., userId, symptom)

Maintenance Metrics:

Software Maturity Index (SMI) helps assess software stability over releases:

$$SMI = \frac{MT - (Fa + Fc + Fd)}{MT}$$

Where:

- MT = Total modules in the current release = 40
- Fa = Modules added = 3
- Fc = Modules changed = 4
- Fd = Modules deleted = 1

$$SMI = \frac{40 - (3 + 4 + 1)}{40} = \frac{32}{40} = 0.80$$

SMI = 0.80 indicates the system is stable with controlled updates and consistent maintenance.

8. CONCLUSION AND FUTURE WORK

The Medi-Team Assist project successfully showcases the creation of a comprehensive digital healthcare platform that effectively bridges the gap between patients and medical professionals. Through systematic black-box testing, all critical functional requirements were rigorously validated to ensure seamless performance from an end-user perspective. The testing confirmed a robust authentication system, ensuring secure login and registration; a user-friendly interface with responsive and intuitive navigation; and reliable core functionalities such as symptom reporting, doctor consultations, prescription handling, and payment processing. Furthermore, the secure transaction system was thoroughly tested for accuracy and safety, guaranteeing protected financial exchanges. By focusing on external functionality rather than internal code, black-box testing helped identify and resolve issues in UI, workflows, and integration, ensuring the platform meets user expectations for reliability and precision. Overall, Medi-Team Assist stands as a ready-to-deploy healthcare solution, providing 24/7 medical support, reducing unnecessary hospital visits, and promoting safe, accessible, and efficient healthcare for users worldwide.