

American International University-Bangladesh (AIUB)  
**Department of Computer Science  
Faculty of Science &Technology (FST)  
Spring 22 23**

**Centralized Medical Application**

Software Requirement Engineering

Sec: **B**

Project submitted

By

*Nasib Ull Huq Shohan (19-39828-1)*

*Thouhida Tasnim (20-43051-1)*

*Kaniz Fatima (20-43137-1)*

*Simanto Podder (20-43480-1)*

**Checked By Industry Personnel**

Name:

Designation:

Company:

Sign:

Date:

1. **PROBLEM DOMAIN**
   1. **Background to the Problem**

Every citizen must go to the doctor lots of times throughout their lifetime. Also, scheduling an appointment to see the doctor and have the tests done takes a lot of time and presents difficulties for a variety of reasons. The citizen must manually make an appointment and then wait a long time for his meeting time. When a patient visits a doctor and the doctor prescribes him medicines or a test, the information is kept on a hard paper that is easily lost or damaged.

The result of this is that we have to go back to the doctor or have the subsequent tests done again, which is costly and time-consuming. These difficulties can be removed simultaneously by a digital web-based solution, which will save time and money.

* 1. **Solution to the Problem**

People frequently struggle with the decision of whether or not to go to the hospital when they consider the inconvenience of seeing a doctor. Today, getting a serial in a government hospital is significantly more challenging. More than an hour must be spent waiting in line. Then, after receiving the serial, some people engage in political maneuvering to meet the doctor early. Again, taking tests and other tasks are challenging. Also, the sufferer will endure great suffering if they lose their previous paperwork. Sometimes, doctors and hospital administrators overcharge patients for prescriptions, medications, procedures, etc. These problems can be solved by standardizing these rates. At that time, they go for a test and consult with the doctor.

These problems can be carefully resolved by a web-based application. It will help people do their tasks more quickly and reduce their worry of getting health care. At the same time, it will also decrease discrimination in regard to the cost of medical care, scheduling appointments with physicians, etc. The existing paper-based process takes a long time, causes a lot of issues, and causes citizens to be treated differently depending on their needs. The web-based solution will get rid of all the hassles and give people an effective means to get the care they need. Citizens will be able to use this method to diagnose mild illnesses without seeing a doctor. With the help of this system, discrimination between social classes will be eliminated, and everyone will receive equal treatment.

There are a few independent applications, including DIMS, DocTime, Arogga, SeekMed, and others. Each of them has unique qualities and was created to fulfill particular requirements. Also, they provide some unique features and offers, but these are insufficient to address these problems. We require a health application that will be created to address each of the issues mentioned.

1. **SOLUTION DESCRIPTION**
   1. **System Features**
2. **User Login**  
   **Functional Requirements:**
   1. The software shall allow users to login with their given username and password
   2. If the username and/or password has been inserted wrong for more than three times, the random verification code will be generated by the system to retry login.
   3. If the number of login attempt exceed its limit (5 times), the system shall block the user account login for one hour *[optional function]*

**Priority Level:** High  
**Precondition:** user have valid user id and password

1. **Doctor Appointment**

**Functional Requirements:**

**2.1** The system will recommend a doctor to the user based on their specific diseases.

**2.2** Additionally, a user can search for doctors manually by name.

**2.3** The user will then be able to view the doctor's name, degree, specialty, and

Qualifications and  appointment time.

**2.4** The user may book an appointment with any doctor by clicking "Book" next to the

doctor name.

**2.5** The user can also find information about hospitals and their doctors on this site.

**Priority Level:** High  
 **Precondition:** user must need to log in with valid id and password

1. **Doctor’s View**

**Functional Requirements:**

* 1. Can view patient list who are requested for appointment.
  2. Can check patient’s old medical history and reports.
  3. Can communicate with patient if necessary.

**Priority Level:** Medium  
 **Precondition:** user must need to log in with valid id and password.

1. **Disease prediction:**

**Functional Requirements:**

**4.1** Disease prediction requires inputs from the user, such as blood pressure and pulse.

**4.2** The user must select the checkbox next to the disease symptoms.

**4.3** The user is required to fill out the 'Duration' and 'Disease State' columns for any

symptoms for which the checkbox has been selected.

**4.4** The user must then select "Submit" for disease prediction, after which doctors will be

suggested based on the disease prediction.

**Priority Level:** Medium

**Precondition:** The user must accurately enter all required information.

1. **Custom Exercise Guide to Patients**

**Functional Requirements:**

**5.1** In this option, a doctor may advise the patient to engage in physical activity.

**5.2** The doctor can provide the patient with the necessary documents or videos.

**5.3** Patients can download or verify the available documents and videos.

**5.4** Additionally, the user is able to communicate with the doctor in real-time via live

chatting, and the doctor is able to monitor the patient's progress.

**Priority Level:** Medium

**Precondition:** The user must log in with their valid email and password and must have

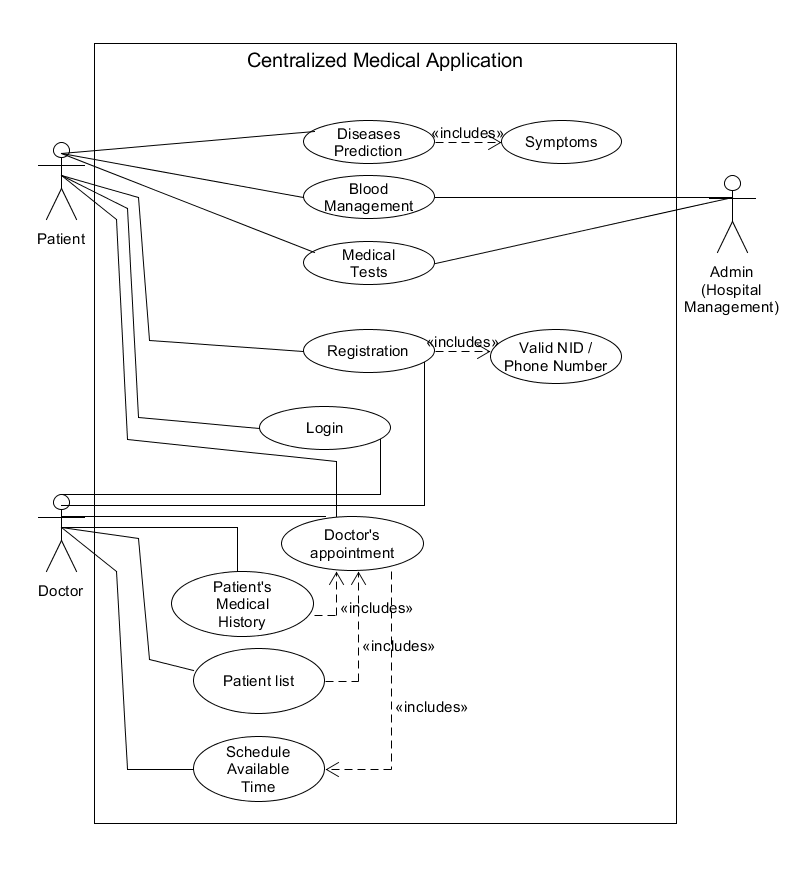
doctor’s consultation subscription.

## System Quality Attributes

1. **Availability:** The system must be 97% available during 8 p.m. to 8 a.m. hours and 99% available between the hours of 8 a.m. and 8 p.m. local time.
2. **Testability:** Software must have the ability to recognize when a system is at danger of failing. There shouldn't be any cyclomatic complexity greater than 15.
3. **Portability**: The web-based platform shall be used to run the system. Using a web browser on any device, the user can access the system.
4. **Maintainability:** It shouldn't take more than two hours for a maintenance programmer to make changes to an existing form. Any system problem must be fixed effectively by the maintenance programmers in less than three hours of manual effort.
5. **Flexibility:**The operation of this system will be easy and straightforward. If anything has to be added or updated, a maintenance programmer may work on the software and generate a new version, including code modifications and testing, in less than 3-4 hours of labor.
6. **Performance:** The system must be able to handle a large number of concurrent users without performance issues, with an average response time of two seconds and at least 1000 simultaneous users without crashing or slowing down.
7. **Efficiency:** The system should be designed to use resources efficiently and quickly to ensure an optimal user experience. It should be able to handle a large number of simultaneous users without slowing down or crashing. Performance metrics should be measured and optimized regularly to ensure peak efficiency.
8. **Security:** The system should have a well-defined authorization system to control user access to various system resources and restrict access to sensitive information.
9. **Reliability:** The system must be reliable, meaning that it should perform its functions correctly and consistently. It must be able to handle errors, prevent data loss, and recover quickly from any failure. The system must have a mean time between failures of at least 10,000 hours, and the mean time to repair must not exceed four hours.
10. **Usability:** The system must have a simple and understandable user interface. It needs to be simple to use, provide helpful feedback when something goes wrong, and be accessible to people with different abilities. Standardized usability testing requires a system usability score of 80% or above.
11. **Reusability:** The system should be built so that its parts can be simply reused and incorporated into other programmers. For maximum reusability, the system should be developed using common practices in the field, such as object-oriented design and the separation of concerns.
12. **Interoperability:** The system must be compatible with common data formats and protocols so that it can share information with other applications. Data import/export functionality and compatibility with other widely used applications and platforms are essential features.
    1. **UML Diagrams**

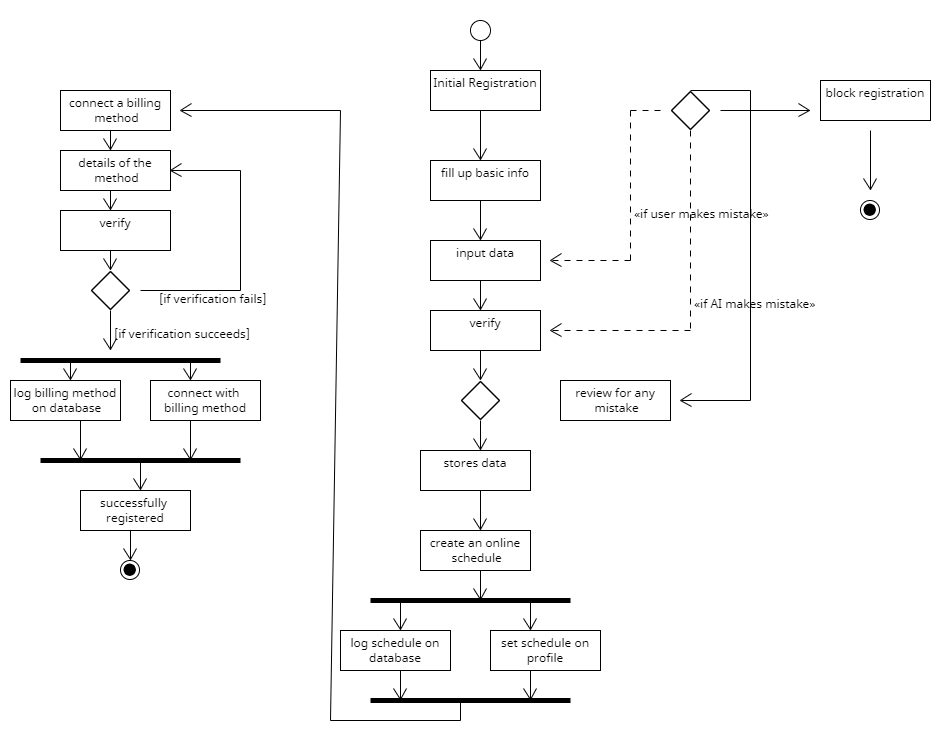
Here we use UMLET & DRAWIO online tool to draw these UML diagrams ( Use-case diagram, Activity diagram, ER diagram) for understanding the features and visualize the relationships.

**Use-Case Diagram:**

****

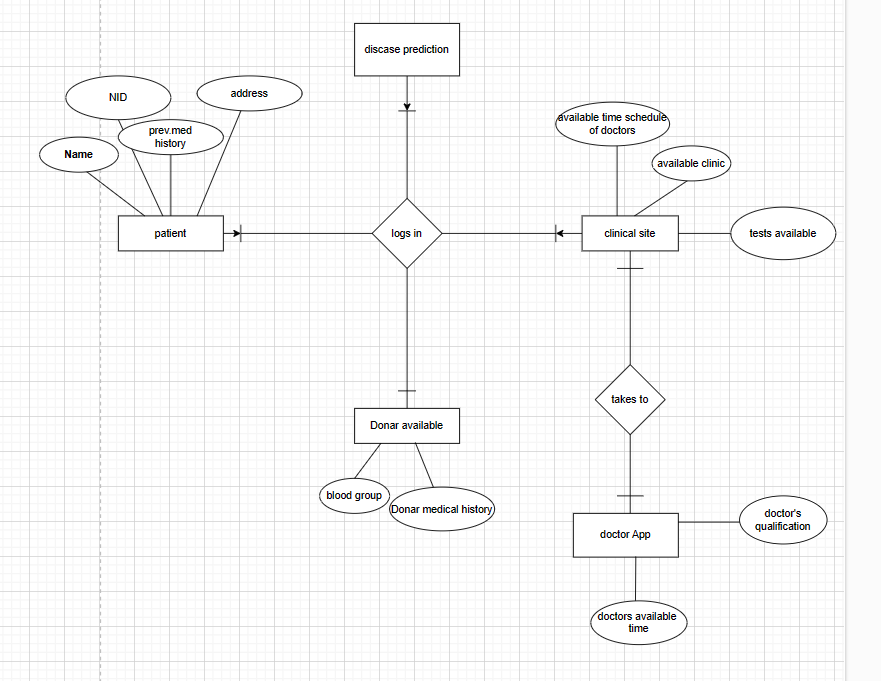
**Fig:** Use case diagram for CMA

**Activity Diagram:**



**Fig:** Activity Diagram for CMA

**ER Diagram:**

****

**Fig:** ER diagram for CMA

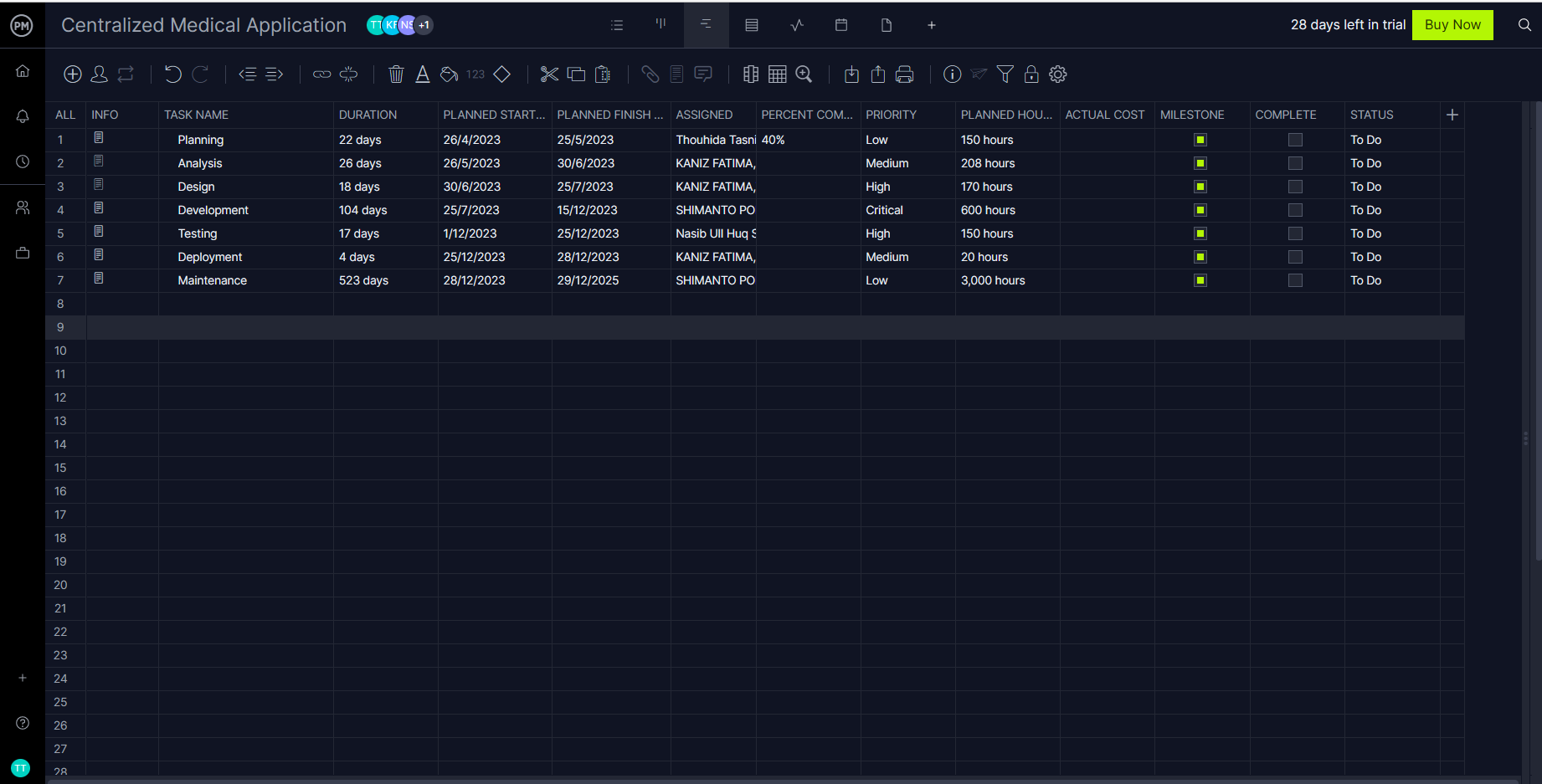
1. **Social Impact**

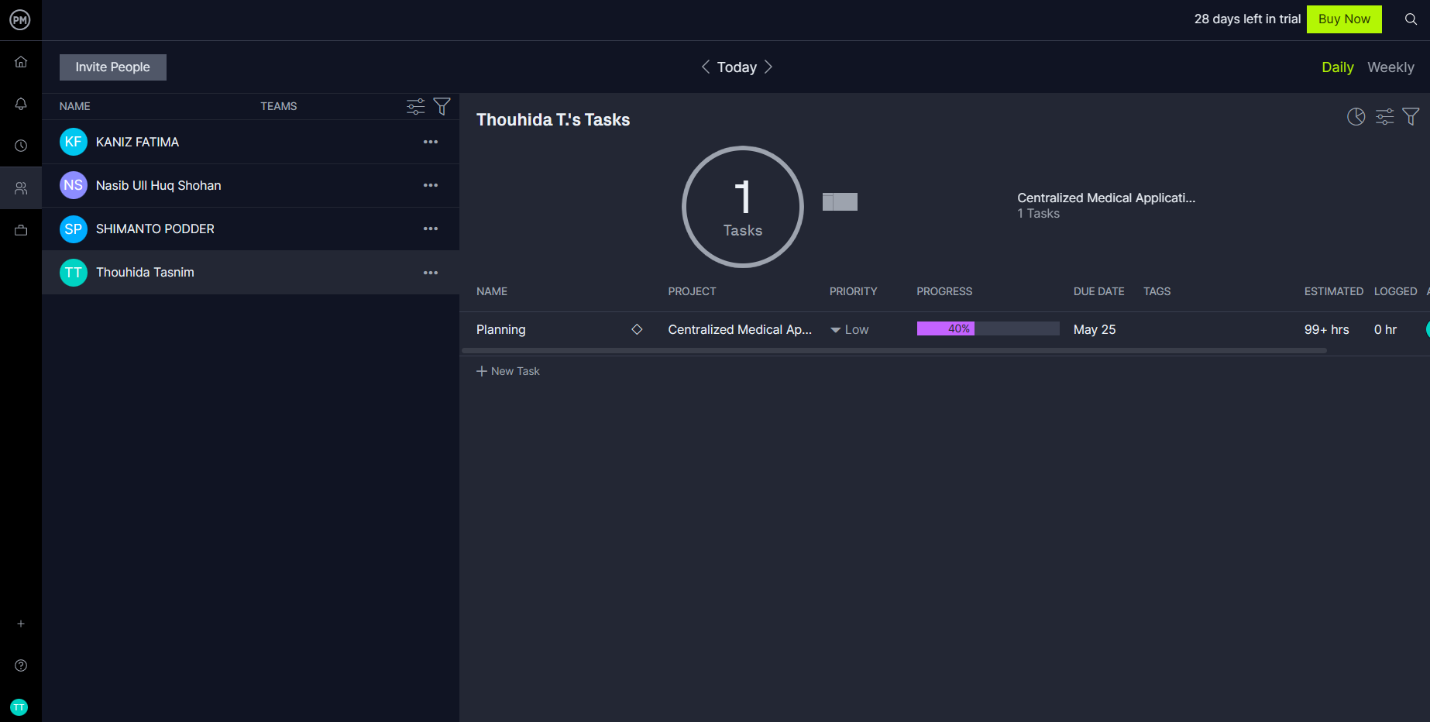
A Centralized Medical Application of social impact could be a platform that connects patients in underserved communities with healthcare providers, resources, and information. The goal of the application would be to improve access to healthcare for individuals who might not have the means to travel to a healthcare facility or have access to adequate medical care in their local communities.

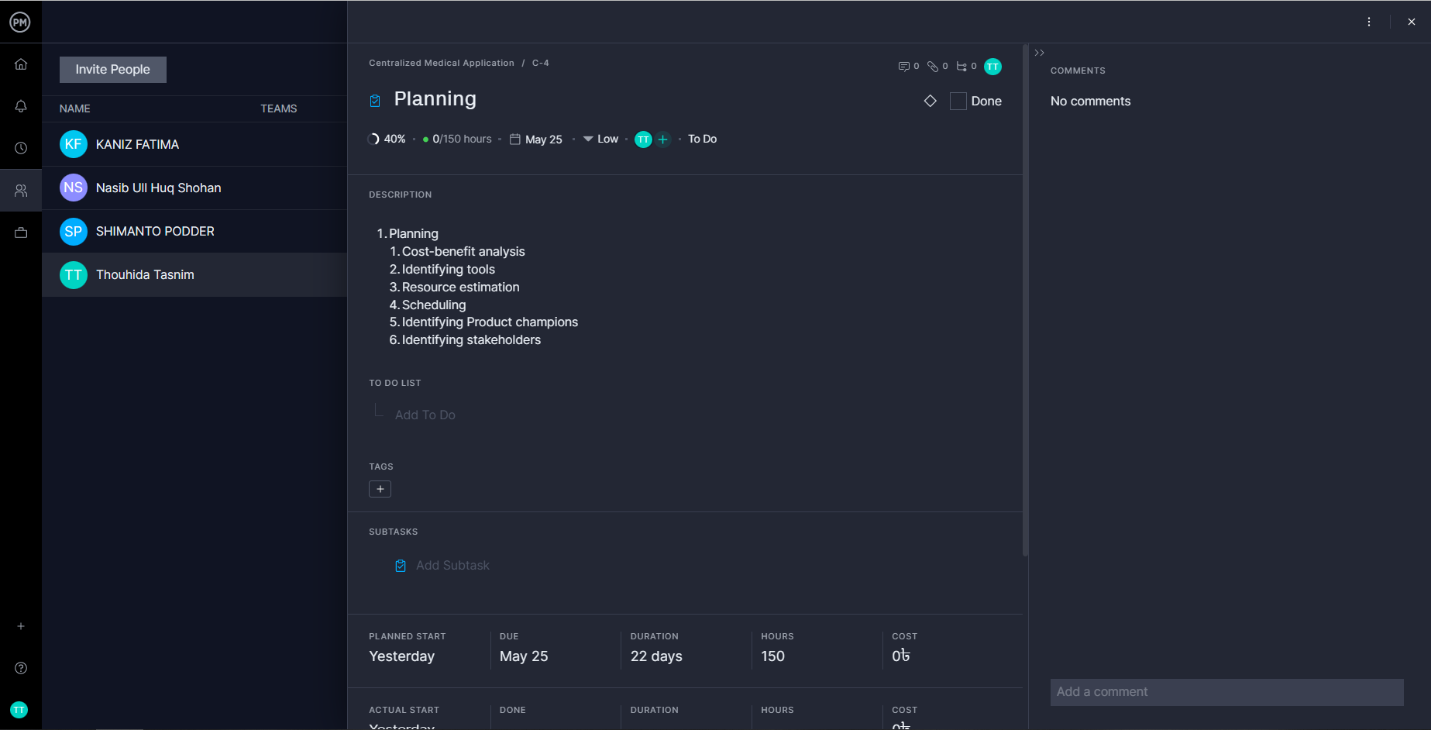
The application could include features such as telemedicine, where patients can consult with healthcare providers remotely, as well as a directory of local clinics, hospitals, and pharmacies. The application could also provide information on health topics such as disease prevention, symptoms, and treatment options.

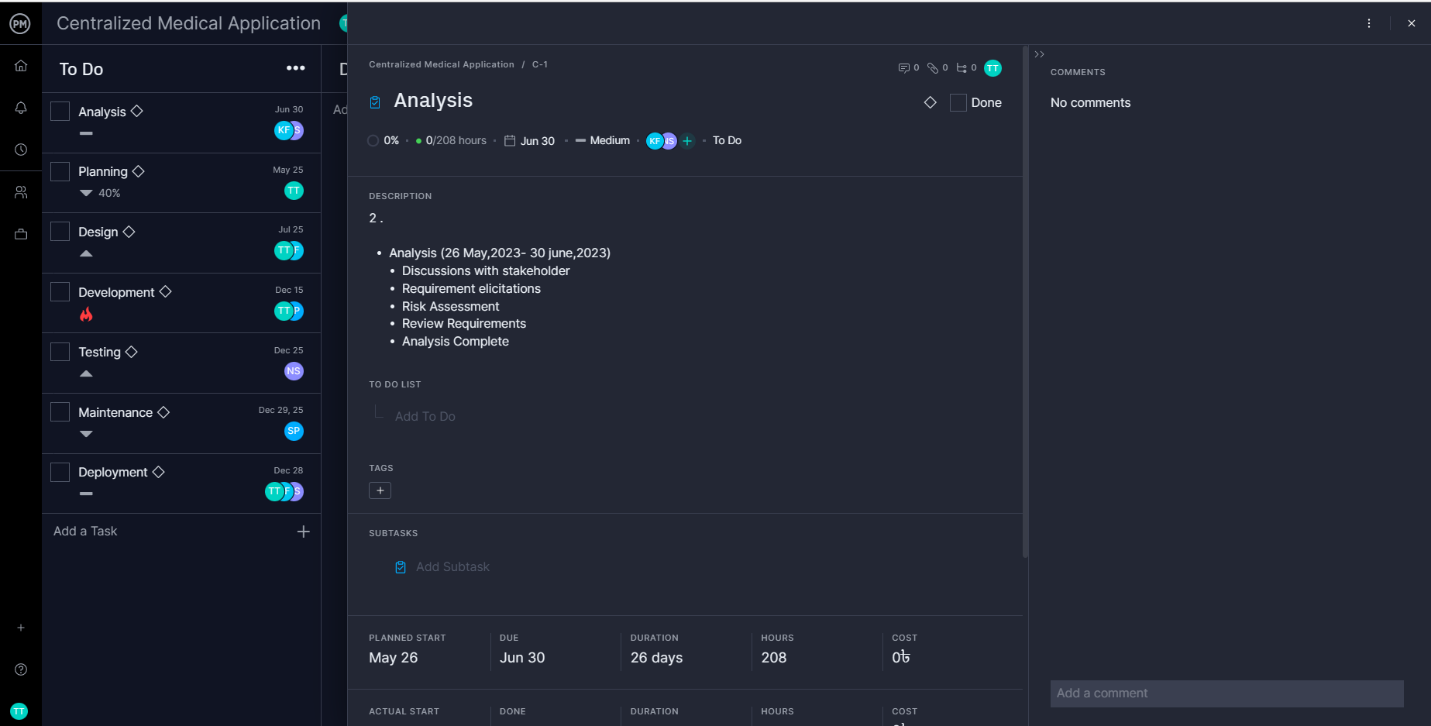
To increase social impact, the application could be designed to prioritize patients who are low-income or uninsured. Additionally, the application could be designed to integrate with electronic health records to provide healthcare providers with a comprehensive view of a patient's medical history and support continuity of care. By facilitating communication and collaboration between patients and healthcare providers, the application could help to improve health outcomes and reduce healthcare disparities in underserved communities.

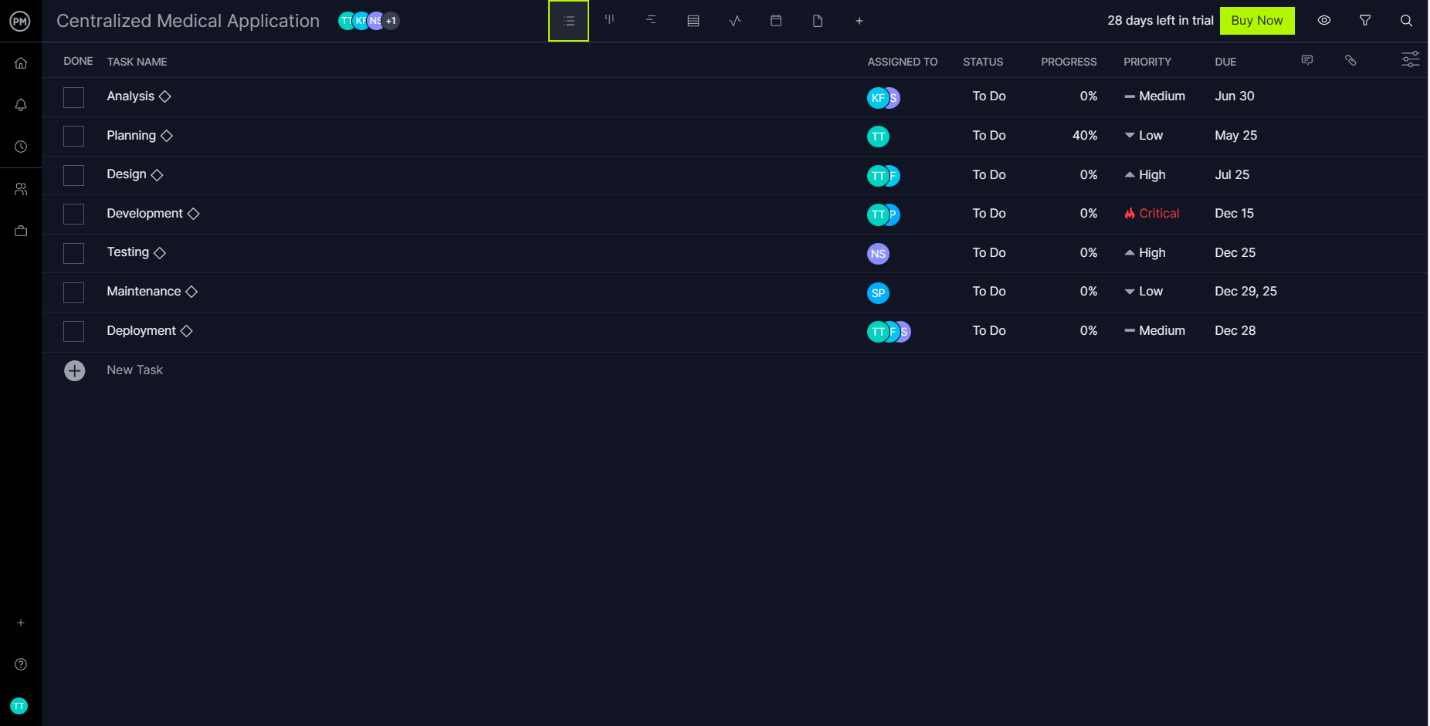
1. **Development Plan with Project Schedule**
2. Planning (25 April,2023 - 25 may,2023)
   1. Cost-benefit analysis
   2. Identifying tools
   3. Resource estimation
   4. Scheduling
   5. Identifying Product champions
   6. Identifying stakeholders
3. Requirement Analysis (26 May,2023 - 30 june,2023)
   1. Discussions with stakeholder
   2. Requirement elicitations
   3. Risk Assessment
   4. Review Requirements
   5. Analysis Complete
4. Design (30 June,2023 - 25 July,2023)
   1. UML diagram design
   2. Interface design
   3. Database Design
   4. Business Rules
5. Development (25 july,2023 - 15 dec,2023)
   1. Front-end development
   2. Back-end development
6. Testing & Fixing (1 dec,2023 - 25 dec,2023)
   1. Check for bugs
   2. Check for errors
   3. Quality analysis
7. Deployment & Implementation (25 decv,2023 - 30 dec,2023)
   1. Deploy the product
   2. Review the product
   3. Release The product
8. Maintenance (25 dec,2023 - 25dec, 2025)
   1. Continuous Support & Bug Fixing
   2. Assess the system’s fitness
   3. Define Marketing strategy











1. **Marketing Plan**

Our target market is healthcare providers, including private practices, clinics, hospitals, and other medical facilities. We aim to target medical professionals who are looking for a cost-effective and efficient way to manage their patients' records and appointments. Our goals are to increase brand awareness by 50% within the first six months of launch and achieve a customer satisfaction rating of 90% or higher.

**Marketing Strategy:**

1. **Short Term Plan:**

**Email Marketing:**

We will create a targeted email marketing campaign to reach potential customers. We will use email marketing to promote our software, offer demos, and share success stories from current customers.

**Referral Program:**

We will establish a referral program that rewards current customers for referring new customers to us. This will help us generate new leads and increase our customer base.

1. **Long Term Plan:**

**Email Marketing:**

We will create a targeted email marketing campaign to reach potential customers. We will use email marketing to promote our software, offer demos, and share success stories from current customers.

**Industry Events:**

We will attend industry events and conferences to showcase our software and network with potential customers. We will also consider sponsoring events to increase our visibility and brand recognition.

1. **Continuous Plan:**

**Online Advertising:**

We will use targeted online advertising on social media platforms and search engines to reach potential customers. We will use keywords such as "medical software," "patient management," and "medical billing" to attract potential customers.

**Content Marketing:**

We will create a blog on our website and publish regular articles related to healthcare and medical software. This will help establish us as a thought leader in the industry and drive traffic to our website.

1. **Cost and Profit Analysis**

**Time:**

Constructive Cost Model:

* Project type : Organic (Basic COCOMO Model)
* Coefficient<effort factor>  : 2.40 [P=1.05, T=0.38]
* SLOC :10000 Lines
* Effort : (2.40\* 101.05) = 26.93
* Dev. time, DM : (2.50\* 26.930.38) = 8.74 = 9 Months = 2160 WH

**Budget:**

Required People, ST: Effort/DM = 26.93/8.74 = 3.80 = 4

Developer & Tester Salary in 9 months:

Per Developer salary Per working Hour = 700 Taka

Total Developer salary = 700\*2160 = 15,12000 Taka

Requirement Analysis:

Time Needed: 5 weeks (25 working days = (25\*12) =300 WH)

Req Analyst Person’s Hourly wage = 300 Taka

Total Req. Analyst salary = 300\*300= 90,000 Taka

Hardware Expense Estimation: 100000 Taka

Rent Expense:

Office space per Month = 50,000 Taka

Total in 9 Months = 450,000 Taka

Total Utilities in 9 Months (including miscellaneous): 20,000 Taka

Maintenance (Till 2 years after Delivery):

Expense per Hour: 1000 Taka

Total Estimated Time needed for Maintenance 2400 Hours

Total Estimated Maintenance Expense = 2400\*1000 = 2400,000 Taka

Advertisement Marketing Cost (Annual) :

Package that includes a total of 30 minutes advertisement 3,00,000Taka

Social Media Sponsored Post:

Facebook/Instagram sponsored post cost per month 25,000Taka

Sponsored post cost in 9 months = 25,000 \* 8 = 225,000/-

Total Estimated Expense:

1512000 + 90000 + 100000 + 450,000 + 20,000 + 2400,000 + 300,000 + 225,000 = 50,97,000Taka

Profit:

Reverse Expense = 15%\*50,97,000 = 764,550 Taka

**Project Budget:**

50,97000 + 764,550 = **58,61,550 Taka**

1. **Reference**
2. <https://www.softwareengineerinsider.com/careers/software-requirements-engineering.html>
3. <https://www.javatpoint.com/software-engineering-requirement-engineering>
4. <https://www.umlet.com/>
5. <https://app.diagrams.net/>
6. <http://www.processimpact.com/goodies.shtml>
7. [ProjectManager](https://t7cci2owjxh.app.projectmanager.com/project/plan/C)