MEDI FUSION

BY GROUP 3

Dhanasree Gidijala
Vinutna Boyapati
Malavika Badam
Ridhima Edimadakala
Tasneem Kousar
Afreen

INTRODUCTION

In today's rapid evolution of health care services, integration of technology

plays a crucial role in enhancing the problem solving….

Our project name itself means fusion of medical services with technology

In recent chaos our platform helps the users to find better

treatment by themselves

Problem Statement & Objectives

- Healthcare management today faces challenges like missed appointments, lack of timely reminders, limited awareness of disease outbreaks, and difficulty in accessing reliable healthcare feedback. Traditional systems often lack integration, making it hard for patients and providers to manage appointments and stay informed.
- **Medi fusion aims to:**

1. Improve accessibility and convenience in appointment scheduling.
2. Provide timely reminders to help patients take medicine .
3. Detect and alert users of potential disease outbreaks in their area.
4. Enable patients to leave feedback, helping others make informed healthcare choices.

Built using the Agile model, MediFusion continuously adapts and improves based on user feedback, aiming to create a more accessible, responsive, and efficient healthcare ecosystem.

Feasibility study

1. Technical Feasibility:

   Platform Compatibility: Ensure that Medi fusion is compatible with web and mobile platforms.

   Security: Assess the ability to implement strong encryption and secure data storage to comply with regulations.

Operational Feasibility:

User Adoption: Conduct surveys with potential users (patients and healthcare providers) to assess interest and willingness to use the platform.

Support Requirements: Identify the need for ongoing technical support and customer service for users to ensure satisfaction and address issues effectively.

3.Economic Feasibility:

Service	**Estimated Monthly Cost**
MongoDB Atlas (M10)	$57.60
Twilio Messaging	$90.00
Total	$147.60

- **Short-term Viability:** MediFusion could operate on the **free version of MongoDB** and **initial Twilio credit** for minimal or no cost, enabling early testing and feature validation.
- **Scaling Costs :** As user adoption increases, costs will scale with MongoDB and Twilio usage. For a growing user base, the **M10 MongoDB version** and **higher Twilio messaging plan** would be viable to support stable and reliable operation, with estimated monthly costs around **$147.60**.
- **Cost Justification:** The anticipated benefits—such as improved healthcare management, patient satisfaction, and potential revenue—justify these costs. With appropriate user growth and monetization, MediFusion has a feasible path to economic viability.

Functional Requirements for MediFusion:

1. **Appointment System:**
   1. A user shall be able to search the appointment lists for all clinics.
   2. The system shall allow users to book, modify, or cancel appointments.
2. **Reminder System:**
   1. The system shall send automated reminders to patients regarding their medicine intake at prescribed time via WhatsApp.
3. **Red Alerts:**
   1. The system shall notify users of potential disease outbreaks based on analyzed data from healthcare authorities.
4. **Feedback Mechanism:**
   1. A user shall be able to leave feedback regarding their healthcare experience, which shall be stored and summarized for provider review.
5. **User Authentication:**
   1. Each user of the system shall be uniquely identified by their email address or user ID, and the system shall require password authentication.
   2. Allows users to create and manage profiles, which store their health information.

Non-Functional Requirements

Property	**Measure**
Speed	• Processed transactions per second • User/event response time • Screen refresh time
Size	• Total storage used (Mbytes) • Number of database records
Ease of use	• Training time for users • Number of help resources available
Reliability	• Mean time to failure (MTTF) • Probability of unavailability • Rate of failure occurrence
Availability	• System uptime percentage
Robustness	• Time to restart after failure • Percentage of events causing failure • Probability of data corruption on failure
Portability	• Percentage of target-dependent code • Number of supported platforms

System Architecture

Client Layer: The frontend interface for patients, doctors, and admins, featuring modules for:

Booking Appointments with doctors.

Reminders for medication adherence.

Red Alerts for local disease outbreaks.

Review Submission for patients to leave verified feedback.

Backend Layer: The application server that contains microservices, each responsible for a specific task:

Appointment Service: Manages appointment scheduling and doctor availability.

Reminder Service: Sends medication reminders to patients.

Alert Service: Provides health alerts based on user location.

Review Service: Handles review submissions.

Data Layer: MongoDB, as a NoSQL database, is flexible enough to handle structured data (like user profiles and appointments) as well as unstructured data (such as reviews and alerts) within the same system. This makes it an ideal choice for storing a variety of data types in a unified platform.

System Architecture

Integration Layer: Connects with third-party services to extend functionality:

- SMS/Email Services for reminders and alerts.

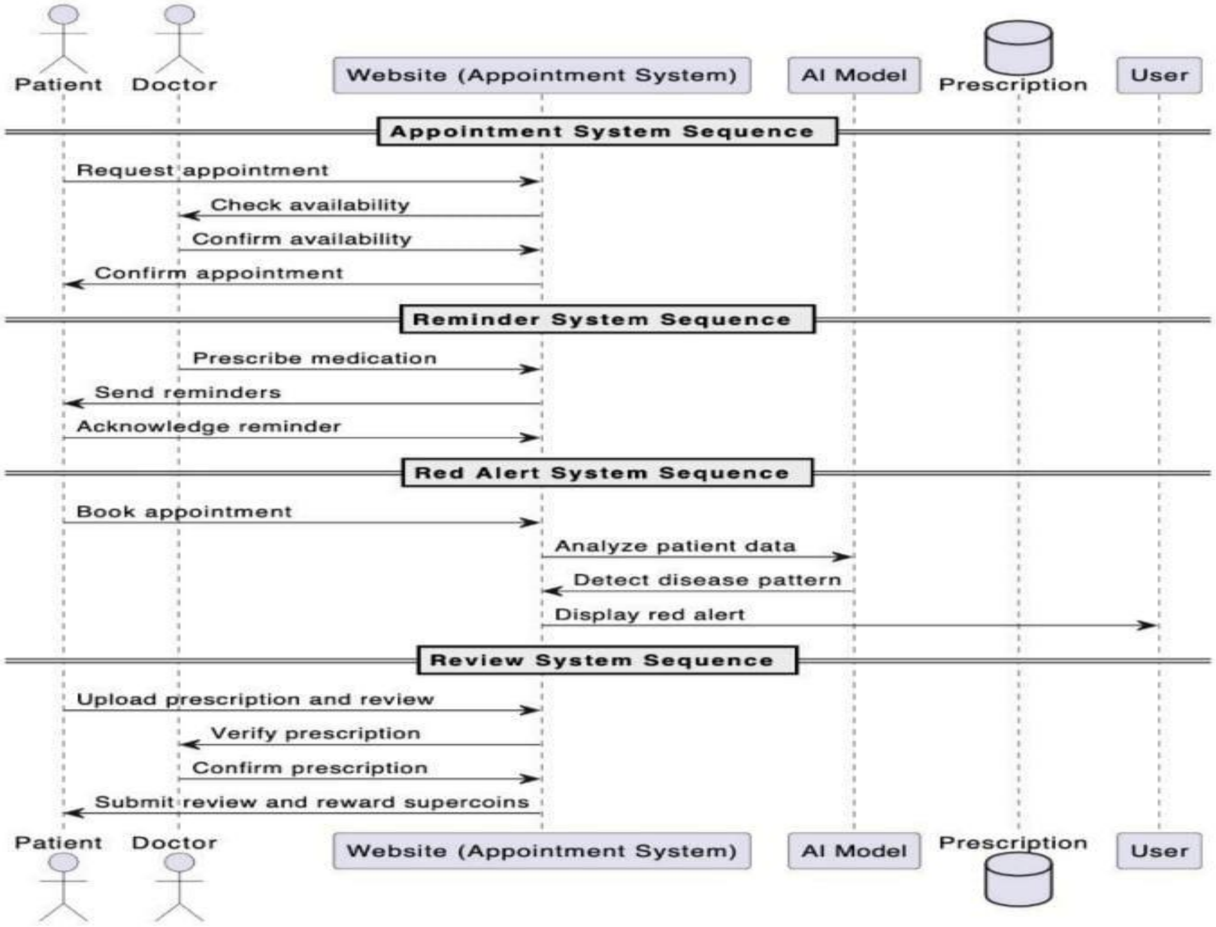
Security and Compliance Layer: Ensures data protection and privacy:

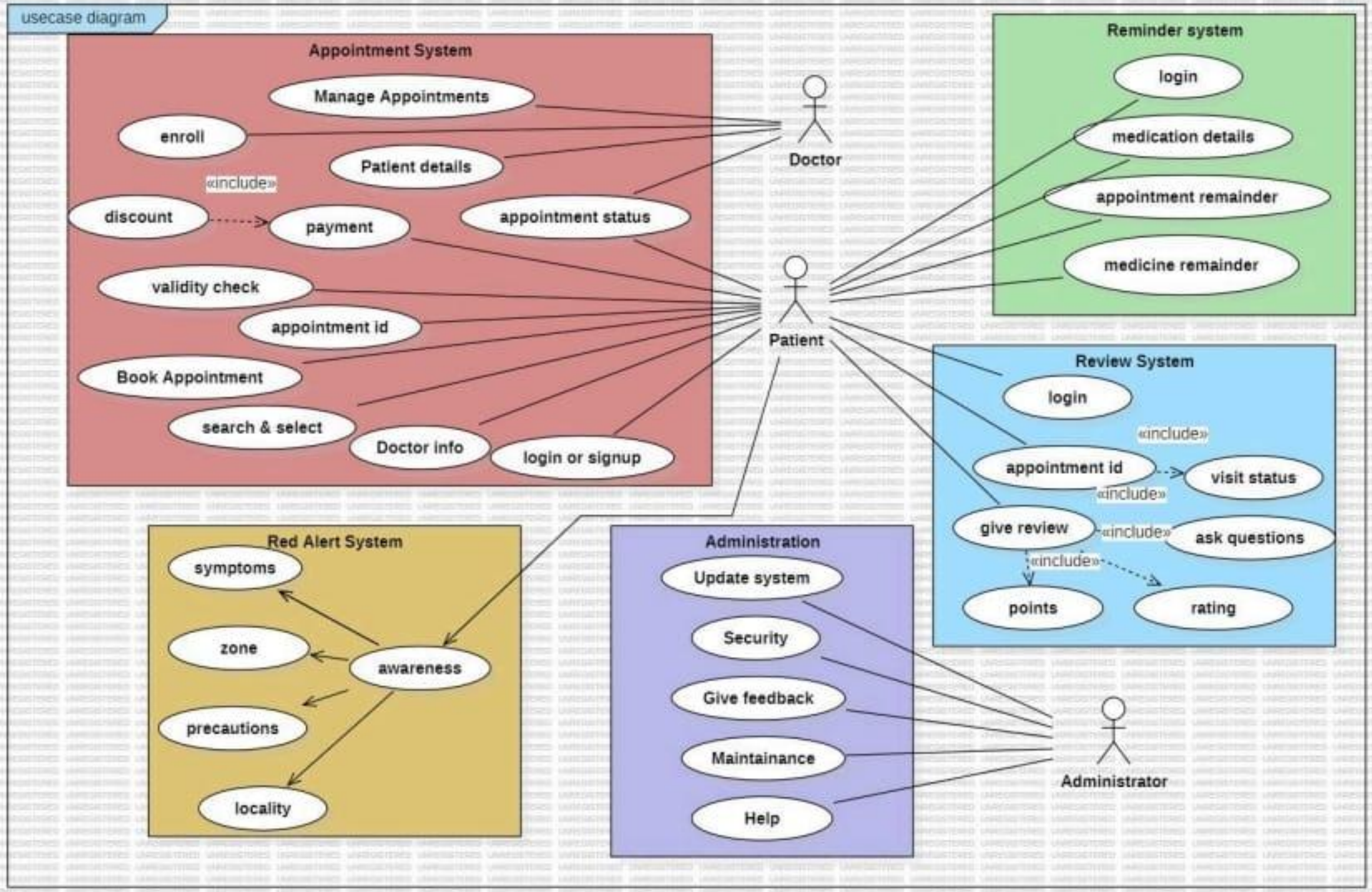
- Authentication and Authorization to control access to features.
- Data Encryption for secure storage and transfer.
- Compliance with healthcare regulations to protect sensitive patient information.

Monitoring and Logging Layer: Tracks system health and performance:

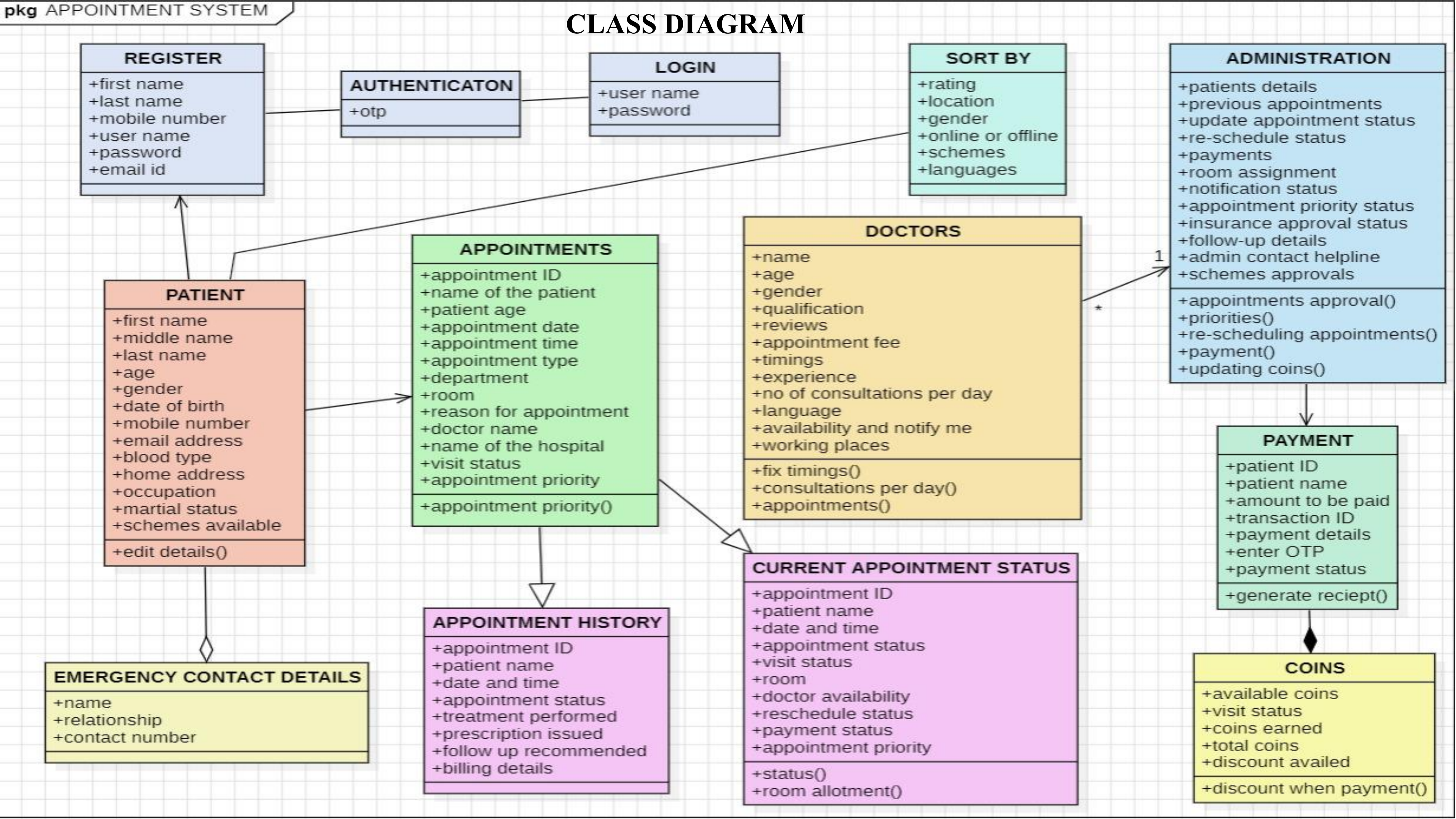
- Monitoring Tools to measure response times and identify issues.
- Logging and Alerts to capture errors and notify admins for real-time troubleshooting.

SEQUENCE DIAGRAM FOR THE WHOLE PROJECT

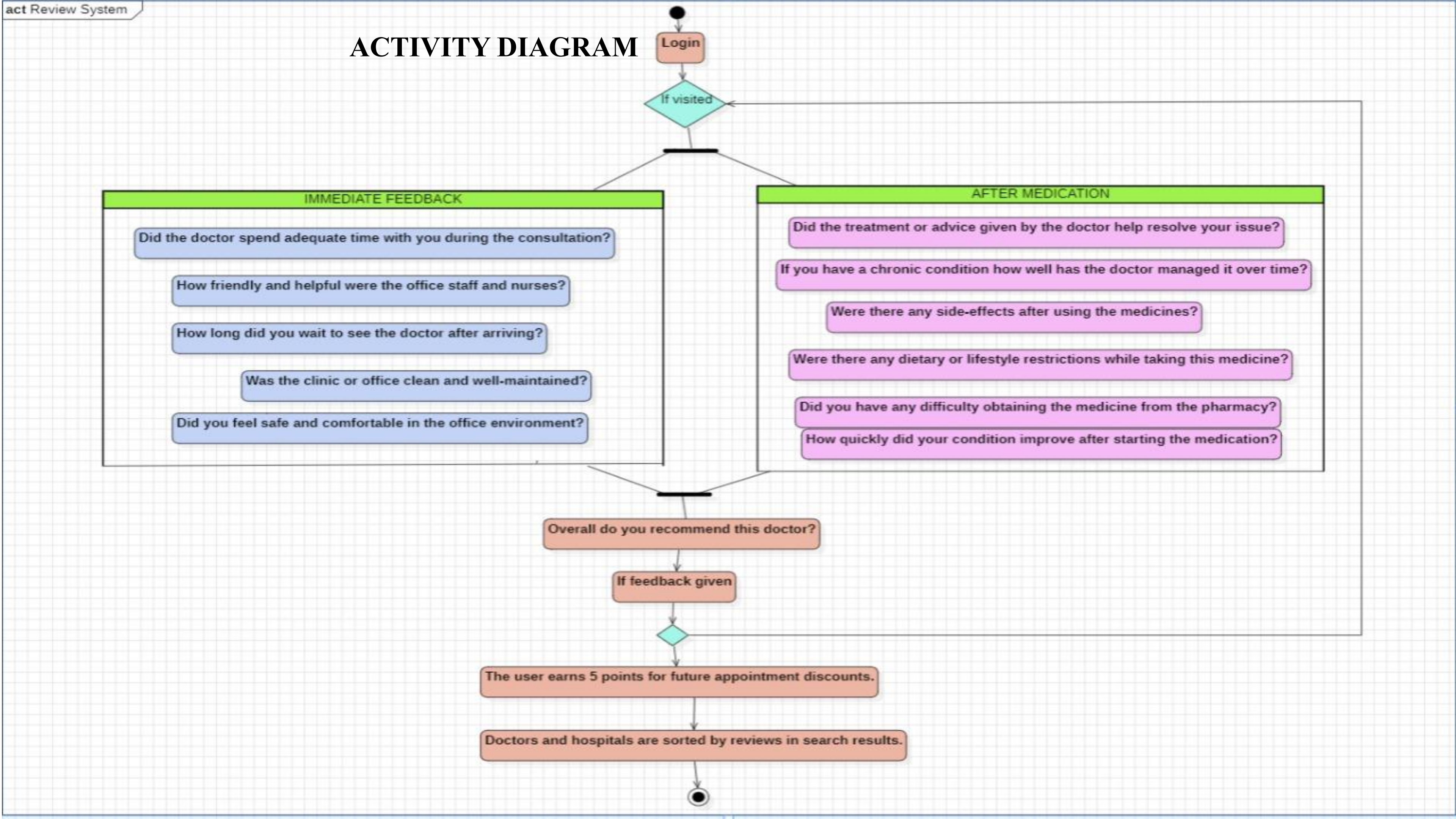


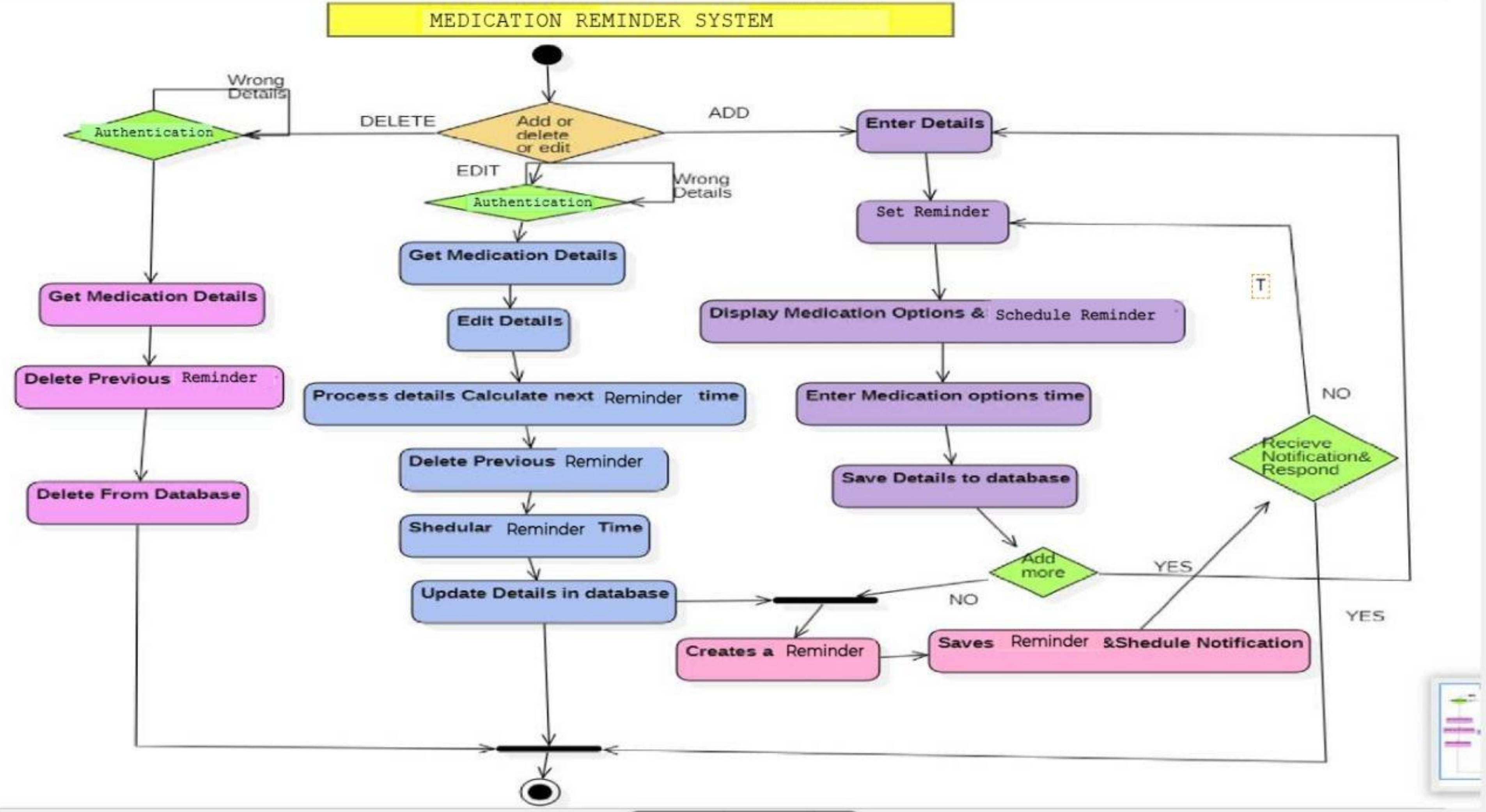

usecase diagram
Appointment System
Manage Appointments
enroll
Patient details
«include»
discount
payment
appointment status
validity check
appointment id
Book Appointment
search & select
Doctor info
login or signup
Doctor
Patient
Reminder system
login
medication details
appointment remainder
medicine remainder
Text
Review System
login
«include»
appointment id
visit status
«include»
give review
«include»
ask questions
«include»
points
rating
Red Alert System
symptoms
zone
awareness
precautions
locality
Administration
Update system
Security
Give feedback
Maintainance
Help
Administrator

CLASS DIAGRAM

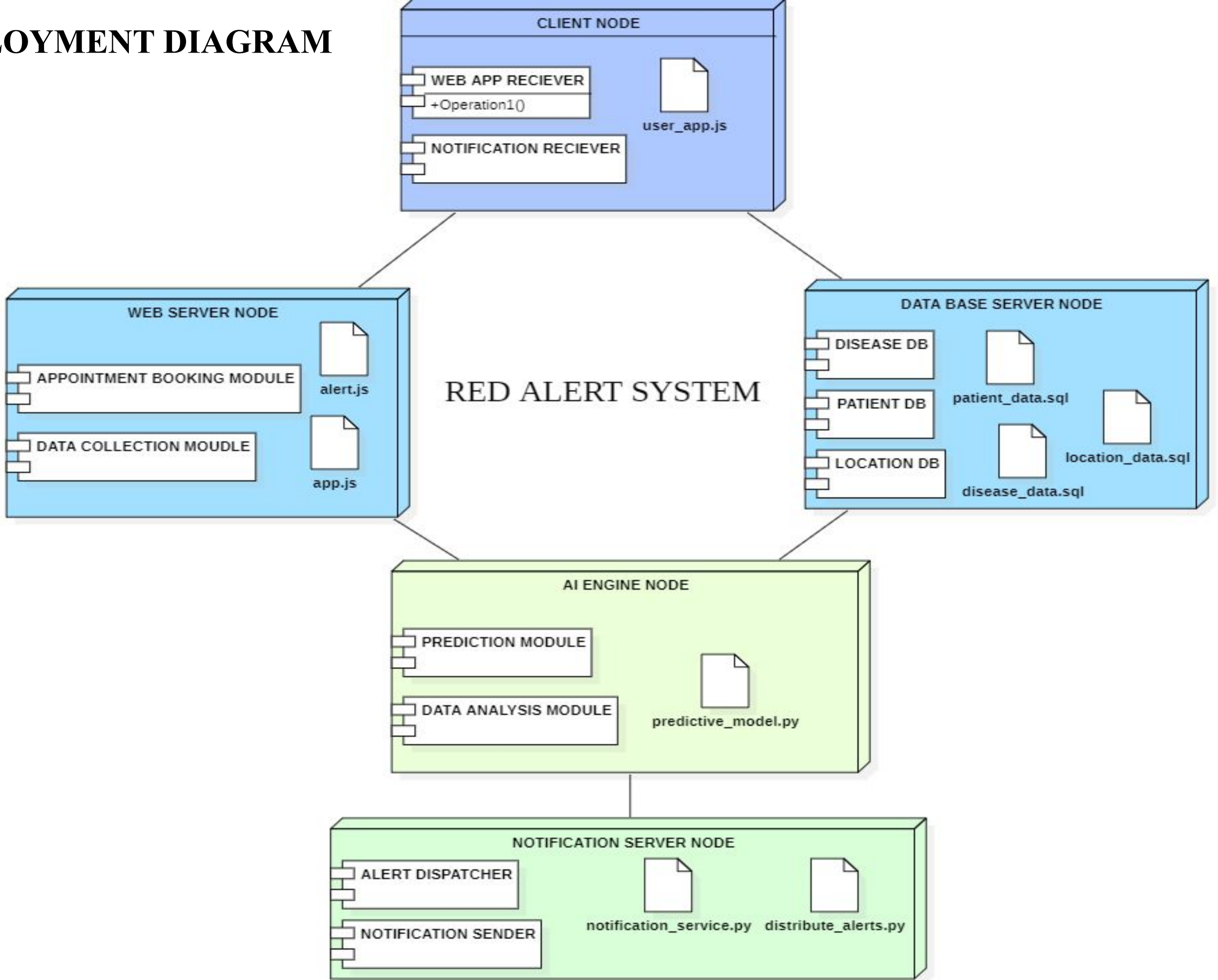


ACTIVITY DIAGRAM


act Review System
Login
If visited
IMMEDIATE FEEDBACK
Did the doctor spend adequate time with you during the consultation?
How friendly and helpful were the office staff and nurses?
How long did you wait to see the doctor after arriving?
Was the clinic or office clean and well-maintained?
Did you feel safe and comfortable in the office environment?
AFTER MEDICATION
Did the treatment or advice given by the doctor help resolve your issue?
If you have a chronic condition how well has the doctor managed it over time?
Were there any side-effects after using the medicines?
Were there any dietary or lifestyle restrictions while taking this medicine?
Did you have any difficulty obtaining the medicine from the pharmacy?
How quickly did your condition improve after starting the medication?
Overall do you recommend this doctor?
If feedback given
The user earns 5 points for future appointment discounts.
Doctors and hospitals are sorted by reviews in search results.


MEDICATION REMINDER SYSTEM
Wrong Details
Authentication
DELETE
Add or delete or edit
ADD
Enter Details
EDIT
Authentication
Wrong Details
Set Reminder
Get Medication Details
Get Medication Details
Edit Details
Display Medication Options & Schedule Reminder
T
Delete Previous Reminder
Process details Calculate next Reminder time
Enter Medication options time
NO
Recieve Notification& Respond
Delete Previous Reminder
Save Details to database
Delete From Database
Shedular Reminder Time
Add more
YES
Update Details in database
NO
YES
Creates a Reminder
Saves Reminder &Shedule Notification

DEPLOYMENT DIAGRAM



Implementation Details

Technology Stack:

- **Frontend:** React.js
- **Backend:** Node.js and Express.js
- **Database:** MongoDB
- **Communication Service:** Twilio's Messaging API

Key Features Implemented:

- **User Registration & Authentication:**
  - Secure registration and login.
- **Appointment Management:**
  - Users can book, modify, and cancel appointments.
- **Medication Reminder System:**
  - Utilizes Twilio API to send SMS reminders to patients for taking medications on time.

Feedback Mechanism:

- Allows users to submit feedback, which is stored and analyzed.

SMS Notification:

- Twilio is configured to send SMS reminders for medication schedules.
- Each user can set up their medication schedule, and the system triggers reminders based on this schedule.

Database Structure:

- **User Collection:** Stores user information (name, email, password hash, phone number).
- **Appointment Collection:** Stores appointment details (user ID, date, time, clinic).
- **Medication Collection:** Stores medication schedules (medication name, dosage, reminder time).
- **Feedback Collection:** Stores feedback submissions (user ID, comments, ratings).

TESTCASES

User Registration & Authentication:

- **Test Case 1:** Verify that users can register with valid credentials.
- **Test Case 2:** Ensure that duplicate user registrations are not allowed.
- **Test Case 3:** Validate that users can log in with correct credentials.
- **Test Case 4:** Confirm that incorrect login attempts are handled gracefully.

Appointment Management:

- **Test Case 5:** Check if users can book an appointment successfully.
- **Test Case 6:** Verify that admins can delete an existing user.
- **Test Case 7:** Ensure that doctors can approve an appointment and that it updates in the database.

Medication Reminder System:

- **Test Case 8:** Verify that users can set up their medication schedule.
- **Test Case 9:** Check if SMS reminders are sent to patients at the scheduled time.
- **Test Case 10:** Ensure that reminders are sent only for active medication schedules.
- **Test Case 11:** Validate that reminders are not sent for deleted medications.

Red Alerts:

- **Test Case 12:** Check if Red alerts are shown when there is an outbreak in the locality.
- **Test Case 13:**Check if Red alerts are not shown when there is no outbreak in the locality.

Feedback Mechanism:

- **Test Case 14:** Ensure that users can submit feedback successfully.
- **Test Case 15:** Validate that feedback is stored correctly in the database.

TEST REPORT

Test Case ID	Description	Expected Outcome	Status
1	Verify that users can register with valid credentials.	Registration is successful with a confirmation message.	Passed
2	Ensure that duplicate user registrations are not allowed.	Registration fails with an error message.	Passed
3	Validate that users can log in with correct credentials.	Login is successful, and the user is redirected to the dashboard.	Passed
4	Confirm that incorrect login attempts are handled gracefully.	Login fails with an error message: "Incorrect password."	Passed
5	Check if users can book an appointment successfully.	Appointment booking is successful, and the appointment appears in the user's list.	Passed
6	Verify that admins can delete an existing user.	User deletion is successful.	Passed
7	Ensure that doctors can approve an appointment.	Appointment status is updated to "Approved" in the system and visible to users.	Passed

Test Case ID	Description	Expected Outcome	Status
8	Verify that users can set up their medication schedule.	Medication schedule is saved and appears in the reminders section.	Passed
9	Check if SMS reminders are sent to patients at the scheduled time.	SMS reminder is received by the patient at the scheduled time	Passed
10	Check if Red alerts are shown when there is an outbreak in the locality.	Red alert is displayed to users in the affected area.	Passed
11	Attempt to register with an already existing email address.	Registration fails with an error message.	Passed
12	Check if Red alerts are not shown when there is no outbreak in the locality.	Red alert is not displayed to users in the affected area.	Passed
13	Attempt to submit feedback without entering required fields.	Submission fails with an error message: "Please fill in all required fields."	Passed
14	Verify system behavior when the database is down during registration.	Error message: "Unable to connect to the database. Please try again later."	Queued

LIVE DEMO

Thank you