Software Requirements Specification

for

Health Hub

Version 1.0 approved

Prepared by: **Huzaifa Khalid**
21I-0527

Anas Imran
21I-2520

Nida Azam
21I-0433

23rd March, 2025

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to specify the software requirements for HealthHub version 1.0. HealthHub is a comprehensive healthcare platform aimed at revolutionizing personalized healthcare management. This Software Requirements Specification (SRS) outlines the scope of the product covered in this document, focusing on its key features and functionalities.

1.2 Document Conventions

The SRS follows standard typographical conventions, with requirements prioritized according to their significance in achieving higher-level objectives. Each requirement statement is assigned its own priority to ensure clarity and ease of reference.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, project managers, marketing staff, users, testers, and documentation writers involved in the development and deployment of HealthHub. It contains an overview of the product's features, detailed functional and non-functional requirements, and specifications for integration with existing healthcare systems. Readers are encouraged to begin with the overview sections and proceed through the sections most pertinent to their roles and responsibilities.

1.4 Product Scope

HealthHub is a groundbreaking healthcare platform designed to transform how individuals manage their well-being and interact with healthcare services. It offers personalized health assessments, telehealth consultations, medical records management, medication adherence and reminders, integration with remote monitoring devices, wellness challenges and incentives, AI-driven health predictions, and health education resources. The software aims to empower users to take charge of their health and promote preventive care strategies.

1.5 References

"HealthHub Vision and Scope Document," Version 1.0, Date: [Insert Date], Source: [Insert Source]

User Interface Style Guide: [Insert Title, Author, Version Number, Date, Source/Location]

Contract Documents: [Insert Title, Author, Version Number, Date, Source/Location]

System Requirements Specifications: [Insert Title, Author, Version Number, Date, Source/Location]

Use Case Documents: [Insert Title, Author, Version Number, Date, Source/Location]

These references provide additional context and guidance for the development and implementation of HealthHub.2.

Overall Description

2.1 Product Perspective

HealthHub is a standalone product designed to revolutionize personalized healthcare management. It is a new, self-contained platform developed to address the growing need for comprehensive and user-centric healthcare solutions. While HealthHub may integrate with existing healthcare systems through APIs and data exchange protocols, it primarily functions as an independent entity.

2.2 Product Functions

Personalized health assessments

Telehealth consultations

Medical records management

Medication adherence and reminders

Integration with remote monitoring devices

Wellness challenges and incentives

AI-driven health predictions

Health education resources

2.3 User Classes and Characteristics

HealthHub is intended for various user classes, including:

Individuals seeking personalized healthcare management

Healthcare professionals providing telehealth consultations and accessing patient data

Administrators managing system configurations and user accounts

Users may vary in technical expertise, with healthcare professionals requiring advanced knowledge of medical practices and individuals requiring user-friendly interfaces for easy navigation.

2.4 Operating Environment

HealthHub operates in a diverse environment, compatible with:

Web browsers (Chrome, Firefox, Safari)

Mobile devices (iOS, Android)

Desktop computers (Windows, macOS)

Internet connectivity for real-time data exchange

2.5 Design and Implementation Constraints

Compliance with regulatory standards (HIPAA, GDPR)

Compatibility with existing healthcare systems through standardized APIs Use

of secure encryption protocols to safeguard sensitive health data Adoption of

scalable architecture to accommodate future growth and expansion

2.6 User Documentation

User documentation components for HealthHub include:

User manuals

Online help resources

Tutorials and video guides

These components will be delivered in digital formats accessible through the HealthHub platform.

2.7 Assumptions and Dependencies

Assumption: Integration with third-party healthcare providers' systems will be feasible through standardized APIs.

Dependency: Availability of remote monitoring devices compatible with HealthHub's data integration protocols.

Assumption: Users will have access to reliable internet connections for telehealth consultations and data synchronization.

Dependency: Compliance with regulatory requirements and data privacy laws in all regions of operation.

3. External Interface Requirements

3.1 User Interfaces

The user interface of HealthHub is designed to be intuitive, user-friendly, and visually appealing. Key characteristics of the user interface include:

Sample screen images depicting various features and functionalities.

Compliance with GUI standards and accessibility guidelines.

Consistent screen layout across different devices (mobile, desktop).

Standard buttons and functions (e.g., help, settings) accessible from every screen.

Error message display standards ensuring clarity and helpfulness.

Keyboard shortcuts for common actions to enhance user efficiency.

Detailed design specifications for the user interface are documented in a separate User Interface Specification document.

3.2 Hardware Interfaces

HealthHub interacts with various hardware components, including:

Supported device types: Mobile devices (iOS, Android), desktop computers (Windows, macOS).

Nature of data and control interactions: Data synchronization between the software and remote monitoring devices via Bluetooth or Wi-Fi.

Communication protocols: Bluetooth Low Energy (BLE), Wi-Fi Direct, USB for device connectivity.

Physical characteristics of hardware interfaces ensure seamless integration and data exchange between HealthHub and external devices.

3.3 Software Interfaces

HealthHub integrates with several software components, including:

Database systems: MySQL for data storage and retrieval.

Operating systems: Compatibility with Windows, macOS, iOS, Android.

Tools and libraries: Use of machine learning libraries for personalized health assessments.

Integrated commercial components: Integration with third-party telehealth platforms for virtual consultations.

Data items shared across software components include user profiles, medical records, and health analytics. Detailed application programming interface (API) protocols are documented separately to ensure seamless integration and interoperability.

3.4 Communications Interfaces

HealthHub requires various communications functions, including:

Email: Sending notifications and alerts to users.

Web browser: Accessing the HealthHub platform from web browsers.

Network server communications protocols: HTTP for web-based interactions.

Electronic forms: Data collection for personalized health assessments.

Message formatting: JSON for data exchange between client and server.

Communication security: SSL/TLS encryption to ensure data security during transmission.

Data transfer rates and synchronization mechanisms: Real-time data synchronization for remote monitoring devices, ensuring up-to-date health information for users.

Communication standards and encryption mechanisms are implemented to safeguard user data and ensure secure communication channels.

4. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1 Personalized Health Assessments

4.1.1 View Health Analytics

Description and Priority:

This feature allows users to view their health analytics, including health status, risk factors, and recommendations. Priority: High.

Stimulus/Response Sequences:

Stimulus: User opens the health assessment menu.

Response: System displays options for viewing health analytics in a human-readable format.

Functional Requirements:

REQ-1: Display all health-related options clearly and comprehensively.

REQ-2: Ensure the presented data is understandable and relevant to the user's well-being.

4.1.2 Receive Health Status and Risk Factor

Description and Priority:

This feature enables users to receive their health status, risk factors, and personalized recommendations based on their data. Priority: High.

Stimulus/Response Sequences:

Stimulus: User accesses the health analytics option and fills in required data.

Response: System displays the user's health stats, risk factors, and recommendations in a visualized format.

Functional Requirements:

REQ-3: Provide users with clear visual representations of their health stats and risk factors.

REQ-4: Validate user-input data and display error messages for incorrect inputs.

4.1.3 Get Recommendations

Description and Priority:

This feature allows users to request personalized proactive recommendations based on their health stats. Priority: Medium.

Stimulus/Response Sequences:

Stimulus: User accesses the health analytics options and requests recommendations.

Response: System performs data analysis and provides personalized recommendations.

Functional Requirements:

REQ-5: Implement algorithms for analyzing user data and generating personalized recommendations.

REQ-6: Ensure recommendations are relevant and actionable for the user's health improvement.

4.2 Virtual Consultations through Telehealth

4.2.1 Access Healthcare Professionals

Description and Priority:

This feature enables users to access healthcare professionals through telehealth for virtual consultations. Priority: High.

Stimulus/Response Sequences:

Stimulus: User selects the telehealth feature.

Response: System displays a list of healthcare professionals with their contact information.

Functional Requirements:

REQ-7: Integrate a database of healthcare professionals accessible through the telehealth feature.

REQ-8: Display relevant contact information for each healthcare professional listed.

4.2.2 Schedule Virtual Consultations

Description and Priority:

This feature allows users to schedule virtual consultations with healthcare professionals through the HealthHub platform. Priority: High.

Stimulus/Response Sequences:

Stimulus: User selects a desired time slot for consultation.

Response: System confirms the appointment and notifies the user.

Functional Requirements:

REQ-9: Provide users with an interface to schedule appointments with healthcare professionals.

REQ-10: Send confirmation notifications to users upon successful appointment booking.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Performance Requirement 1:

Description: The system should respond to user actions within 2 seconds under normal load conditions.
Rationale: This ensures a responsive user experience and prevents frustration due to long wait times.
Performance Requirement 2:

Description: The system should be able to handle concurrent virtual consultations for up to 1000 users without significant degradation in performance.
Rationale: This ensures the system can scale effectively to accommodate a large user base.

5.2 Safety Requirements

Safety Requirement 1:
Description: The system must comply with HIPAA regulations to ensure the privacy and security of patient health information.
Rationale: Protecting patient confidentiality and data security is critical in healthcare applications to prevent unauthorized access and breaches.

5.3 Security Requirements

Security Requirement 1:
Description: User authentication must be performed using strong encryption protocols such as HTTPS for data transmission.
Rationale: Ensures secure communication between the user's device and the HealthHub servers, preventing unauthorized access to sensitive information.

5.4 Software Quality Attributes

Software Quality Attribute 1: Usability:

Description: The system should provide an intuitive user interface with clear navigation and understandable terminology.
Rationale: Enhances user satisfaction and adoption rates by making the system easy to use and navigate.
Software Quality Attribute 2: Reliability:

Description: The system should have a 99.9% uptime, ensuring users can access critical features without interruptions.
Rationale: Maintains user trust and confidence in the system's reliability for accessing healthcare services.

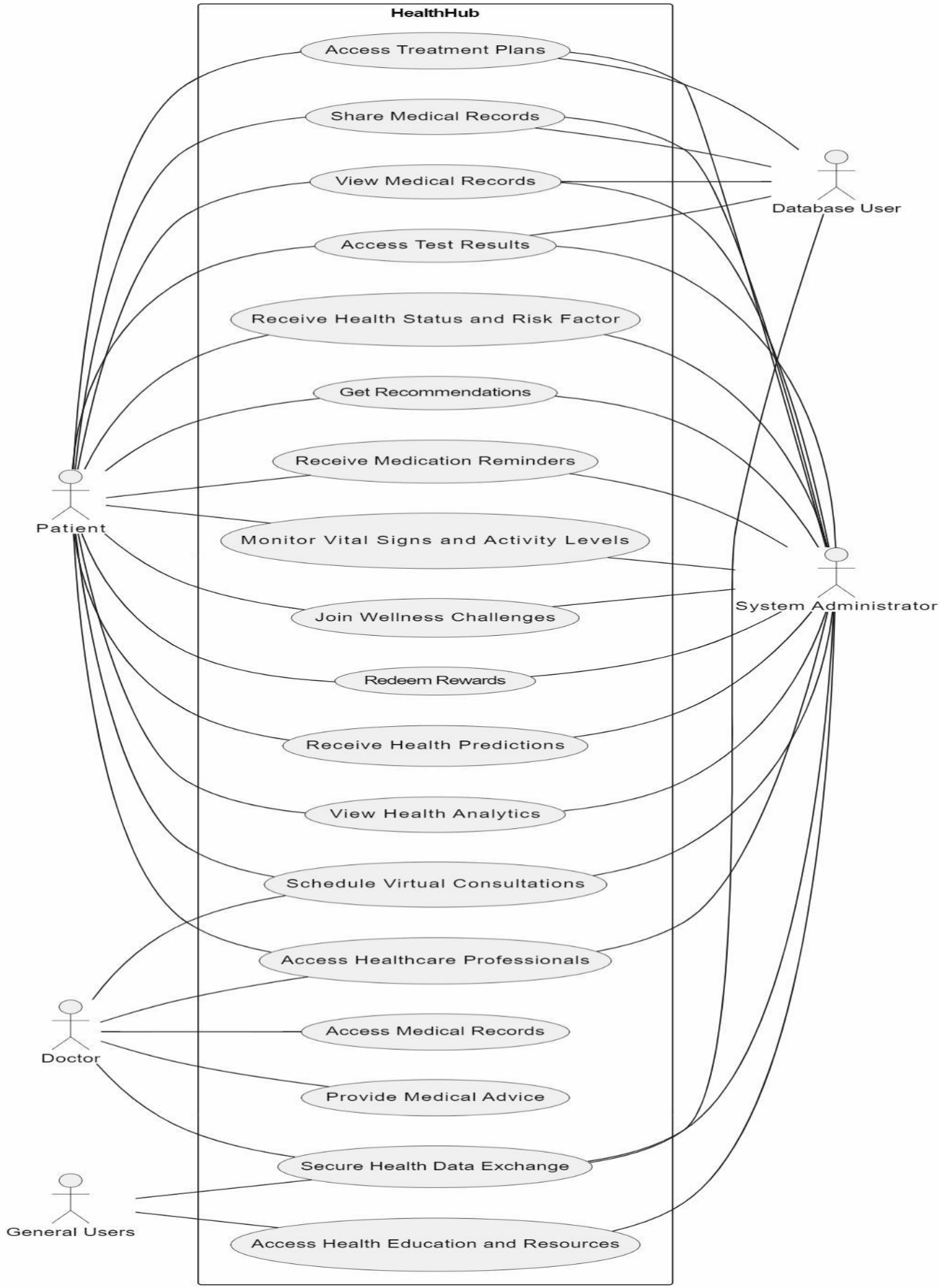
5.5 Business Rules

Business Rule 1:

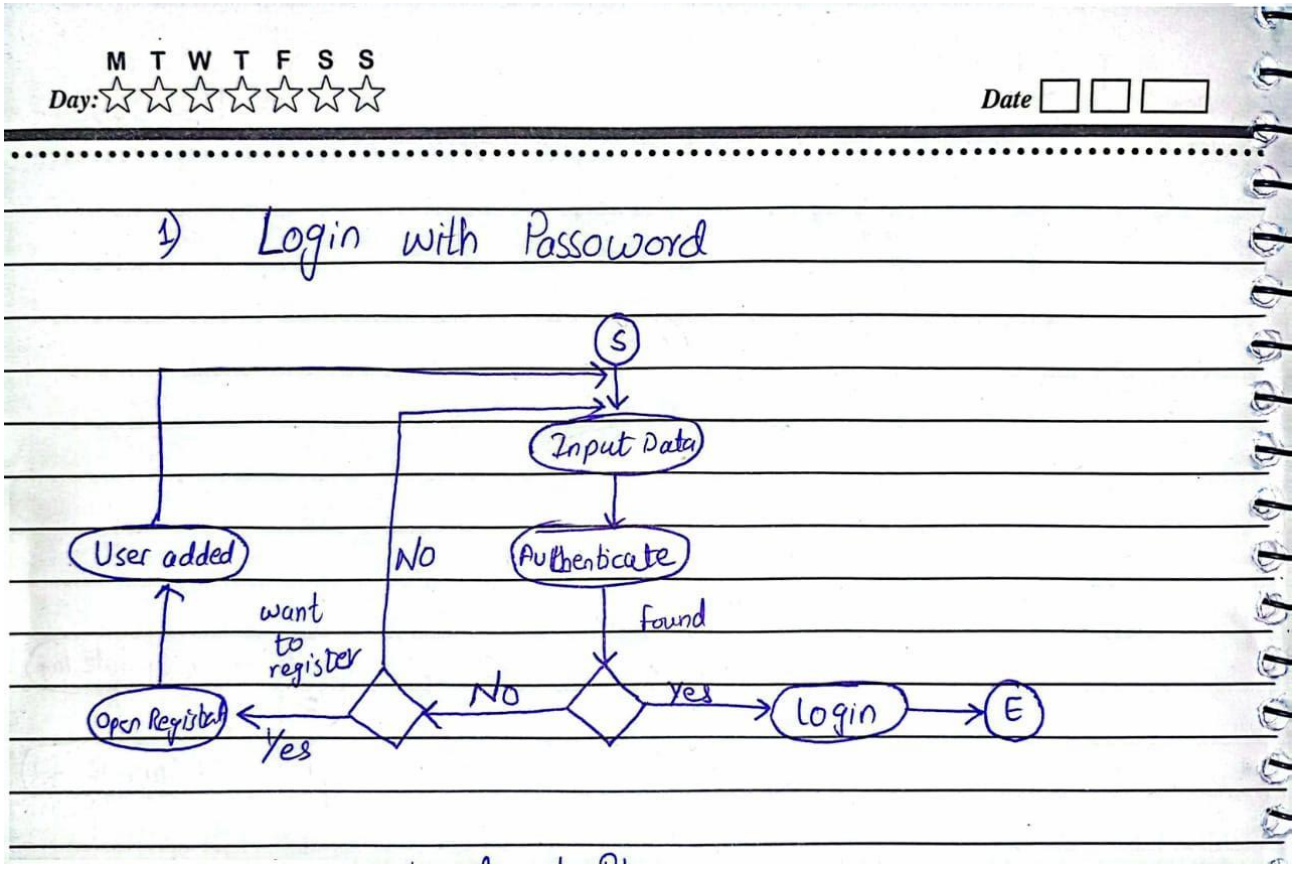
Description: Only authorized healthcare professionals should have access to patients' medical records and be able to prescribe treatments.
Implication: Requires implementing user authentication and role-based access control to enforce access restrictions.

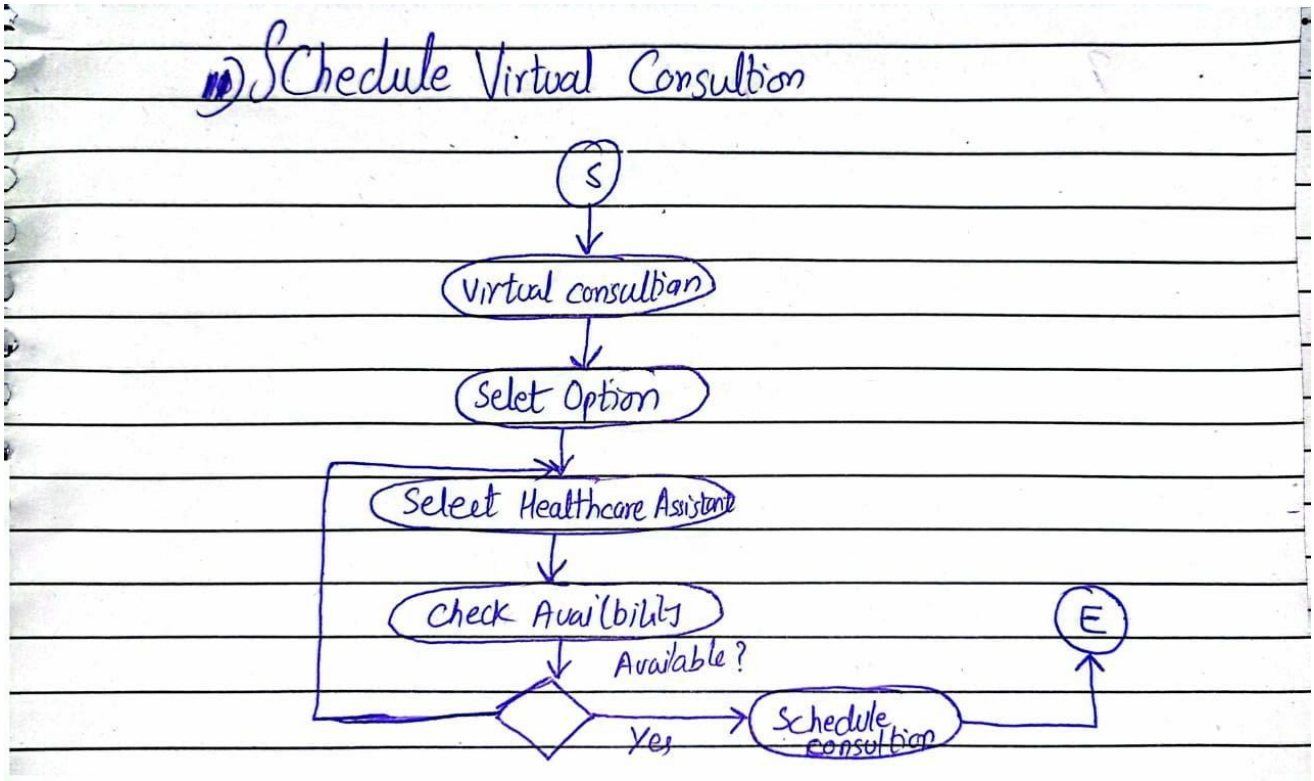
6. Diagrams

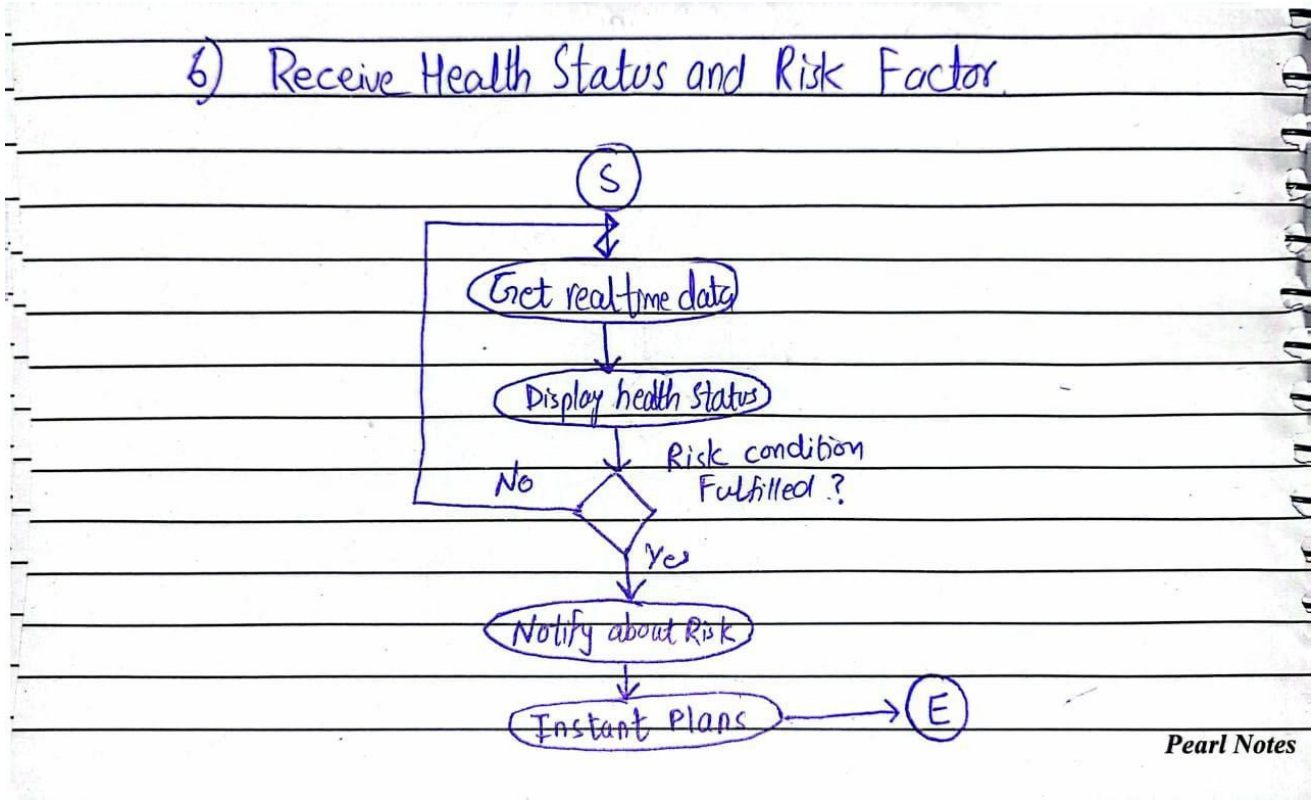
6.1 Use Case Diagram


HealthHub
Access Treatment Plans
Share Medical Records
View Medical Records
Access Test Results
Receive Health Status and Risk Factor
Get Recommendations
Receive Medication Reminders
Monitor Vital Signs and Activity Levels
Join Wellness Challenges
Redeem Rewards
Receive Health Predictions
View Health Analytics
Schedule Virtual Consultations
Access Healthcare Professionals
Access Medical Records
Provide Medical Advice
Secure Health Data Exchange
Access Health Education and Resources
Patient
Doctor
General Users
Database User
System Administrator

6.2 Activity Diagram



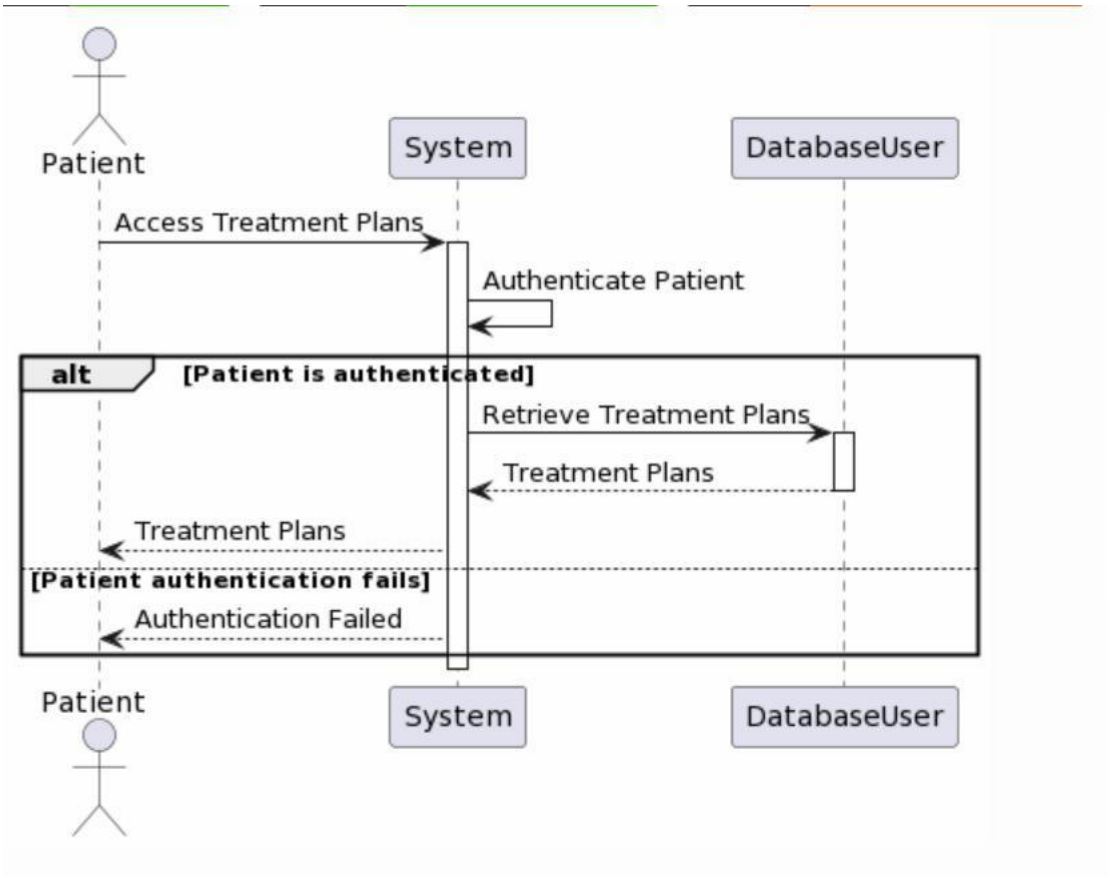

5) Schedule Virtual Consultion
S
Virtual consultion
Selet Option
Seleet Healthcare Assistant
Check Availbility
Available?
Yes
Schedule consultion
E


6) Receive Health Status and Risk Factor
S
Get real time data
Display health status
Risk condition Fulfilled?
No
Yes
Notify about Risk
Instant Plans
E

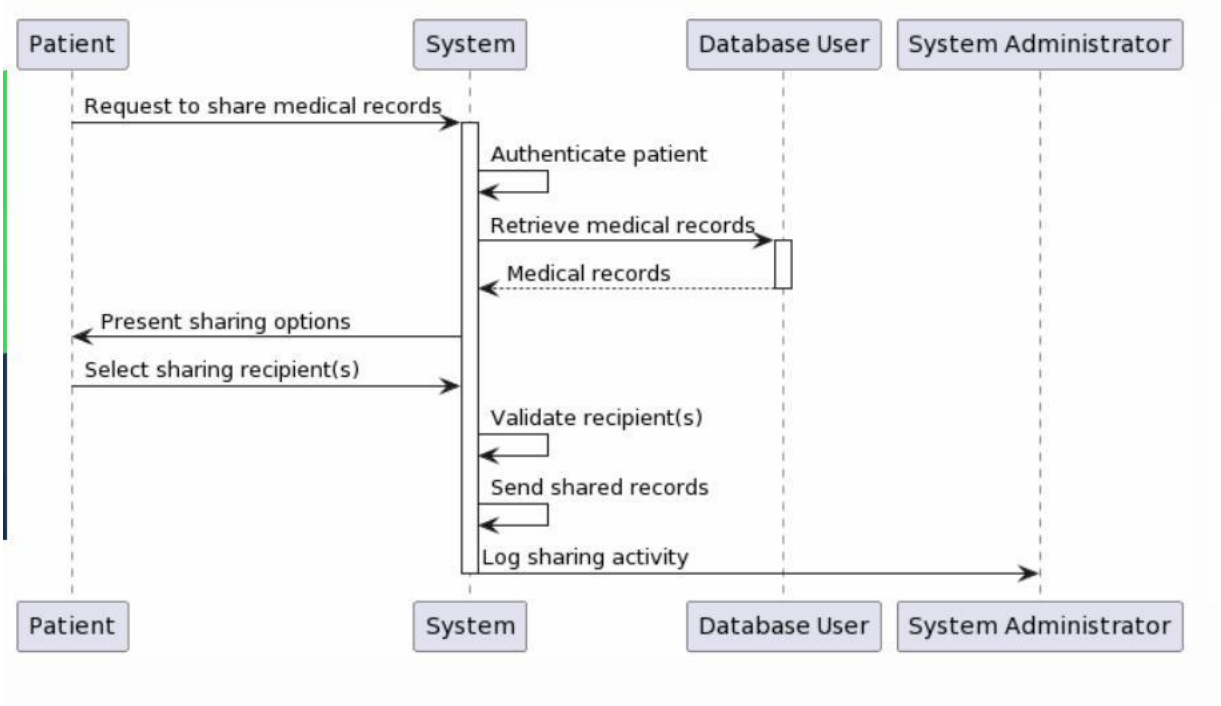
6.3 Sequence Diagram

System Diagrams:

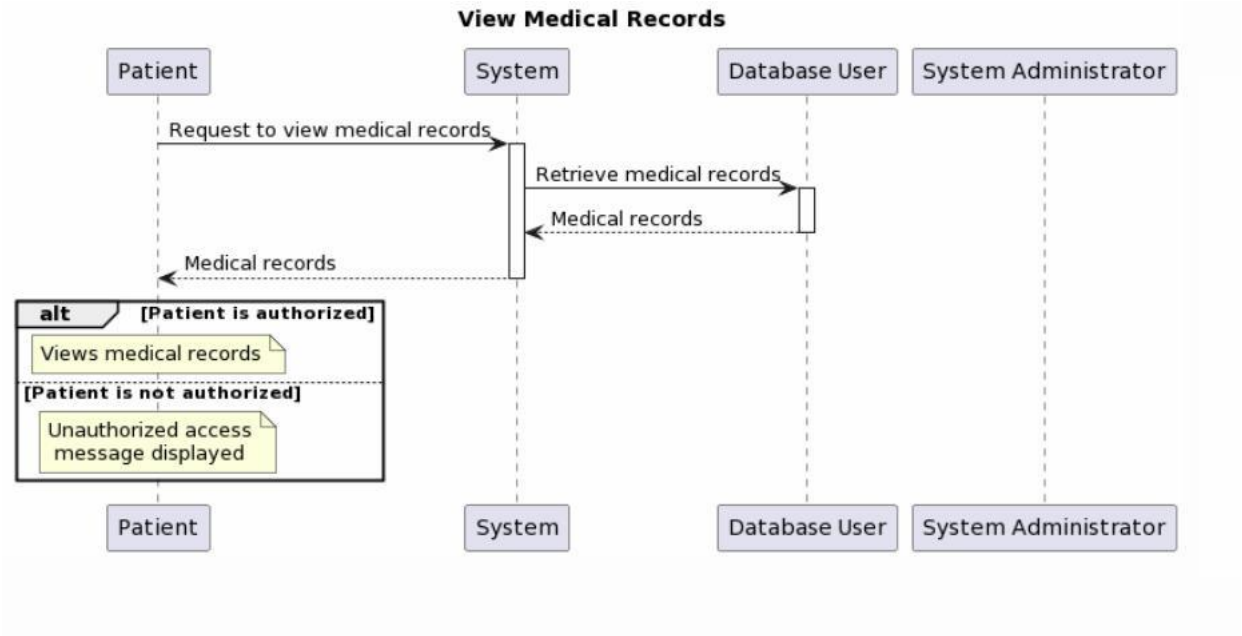
1) Access Treatment plans:



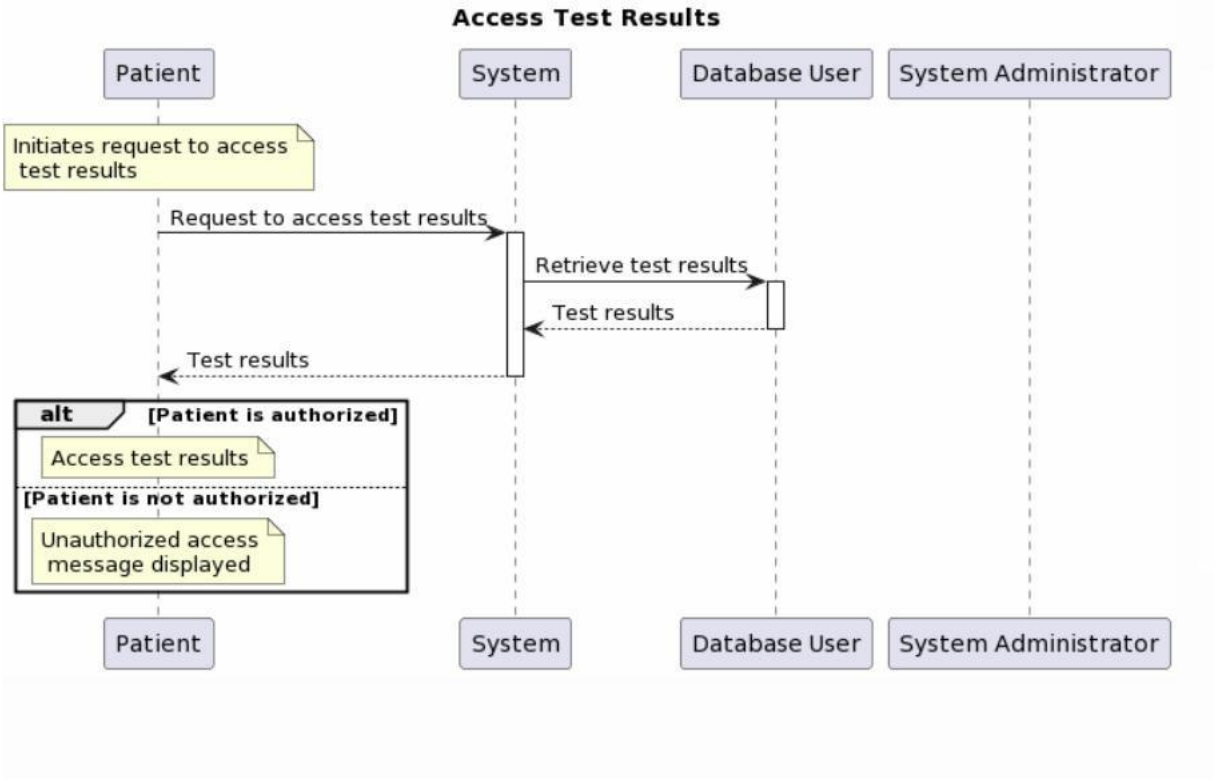
2) Share Medical Records:



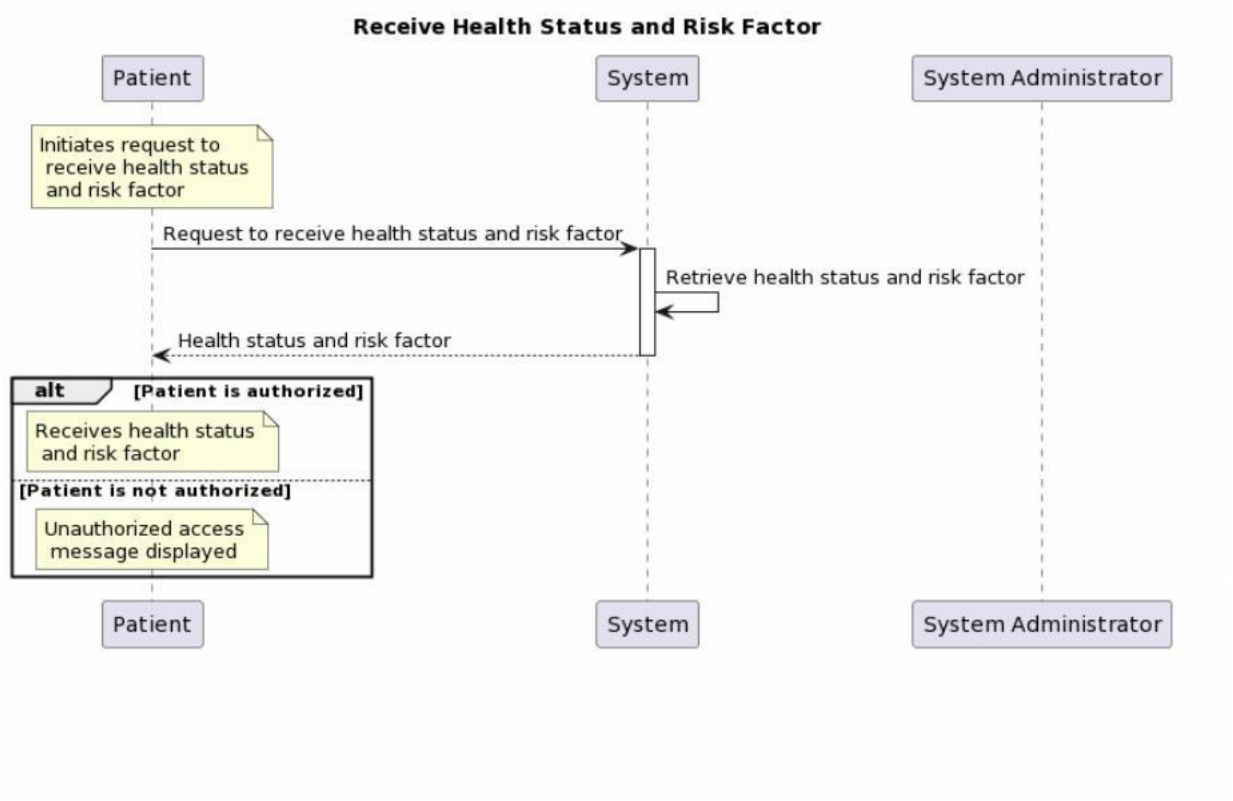
3) View Medical Records:



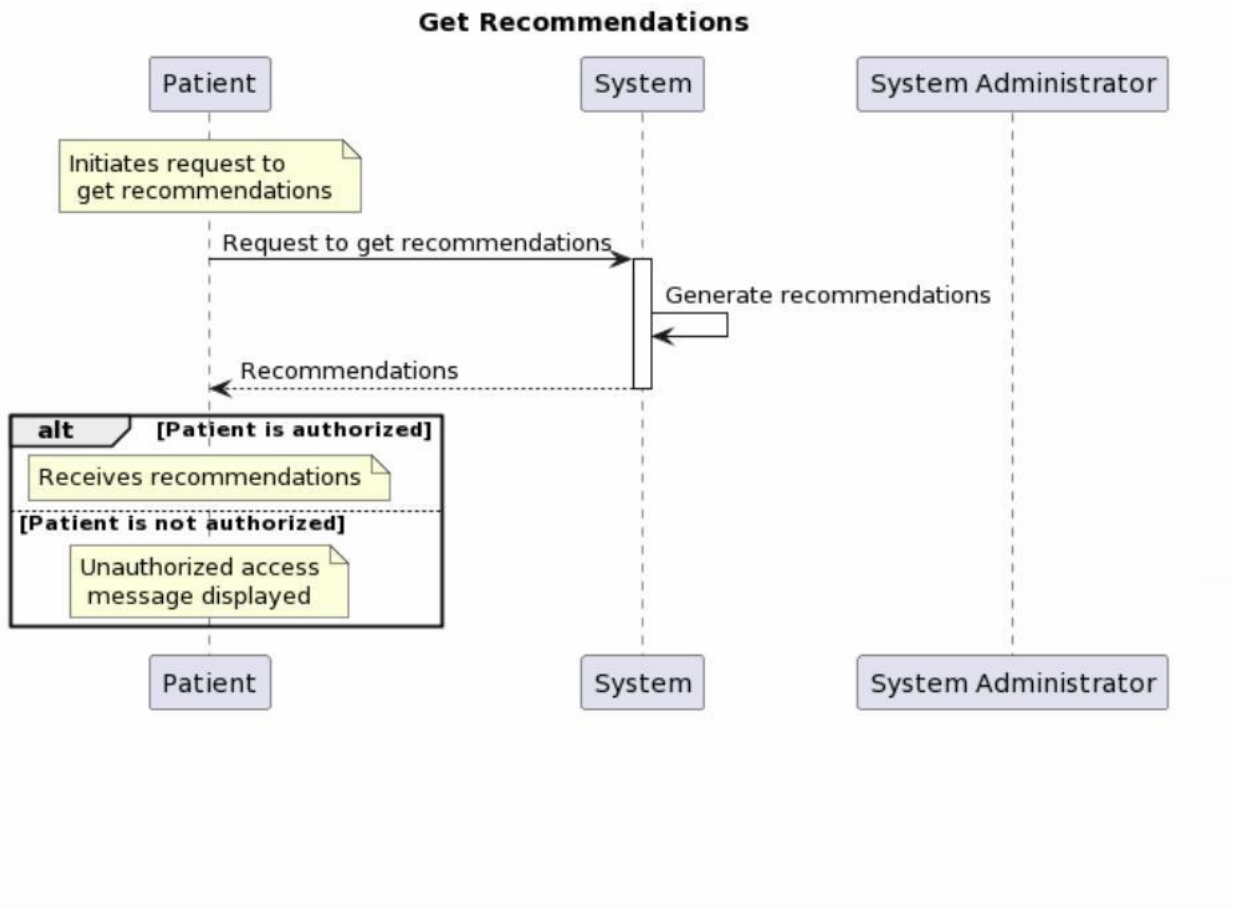
4) Access Test Results:



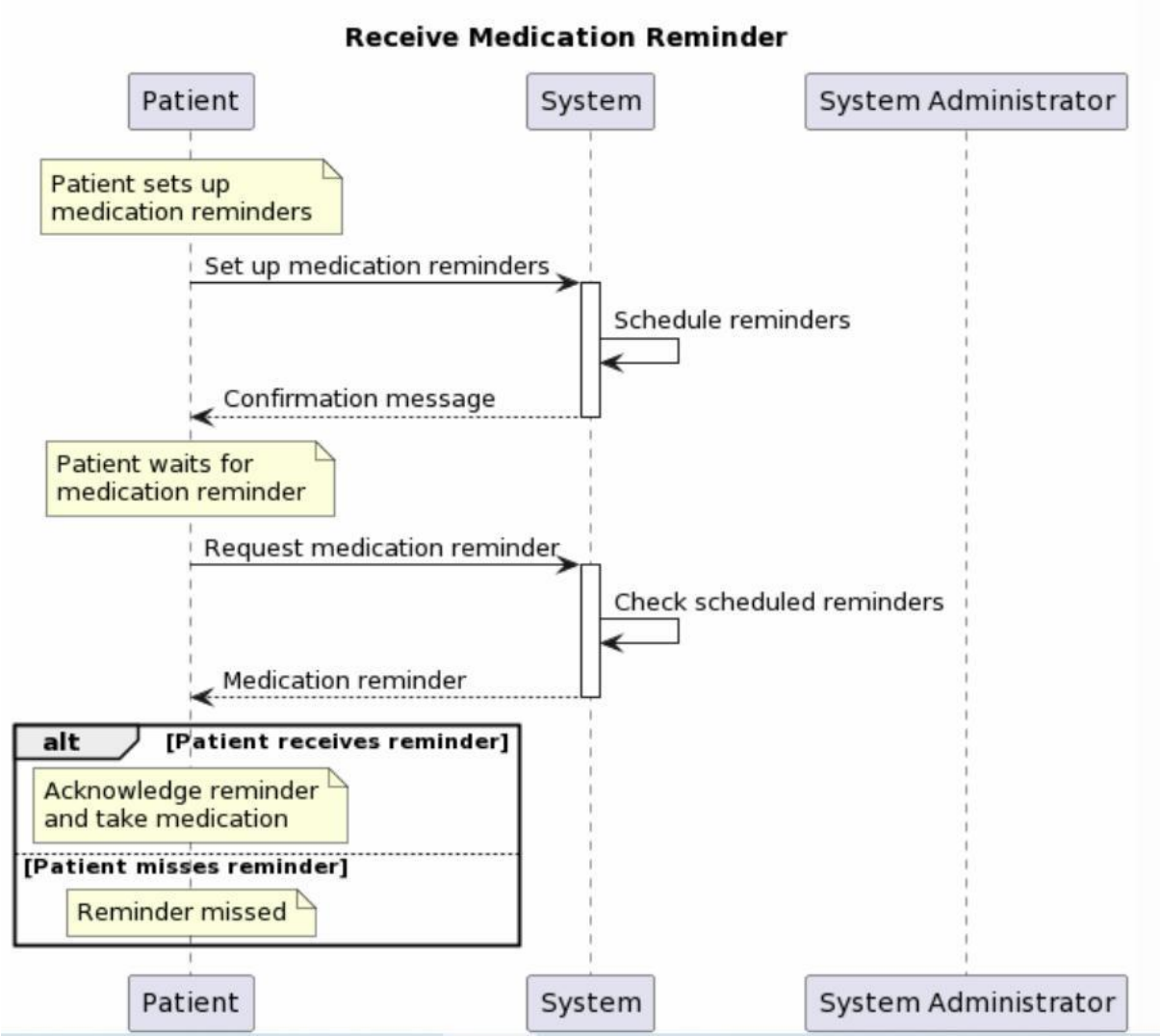
5) Receive Health Status and Risk Factor:



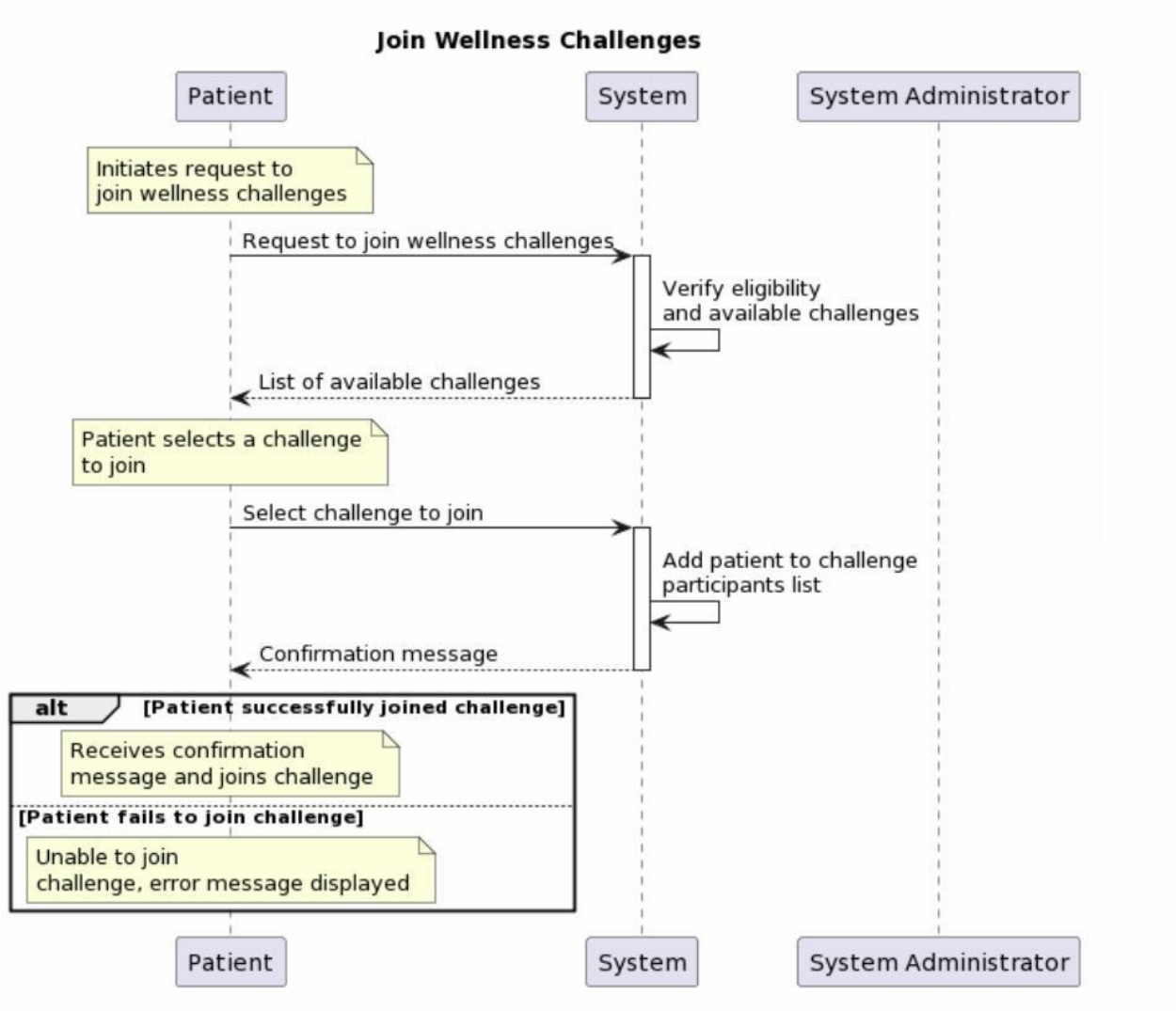
6) Get Recommendations:



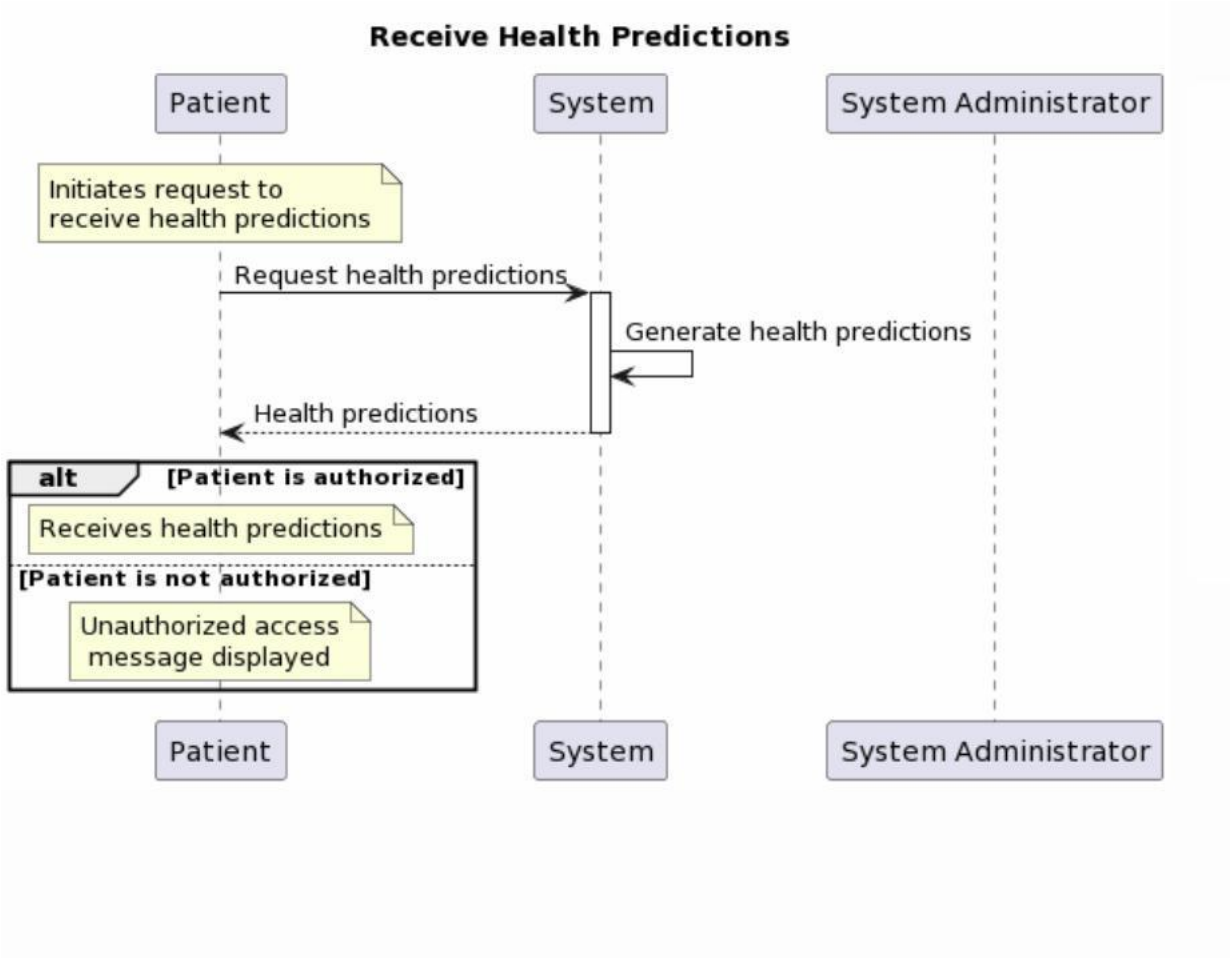
7) Receive Medication Reminders:



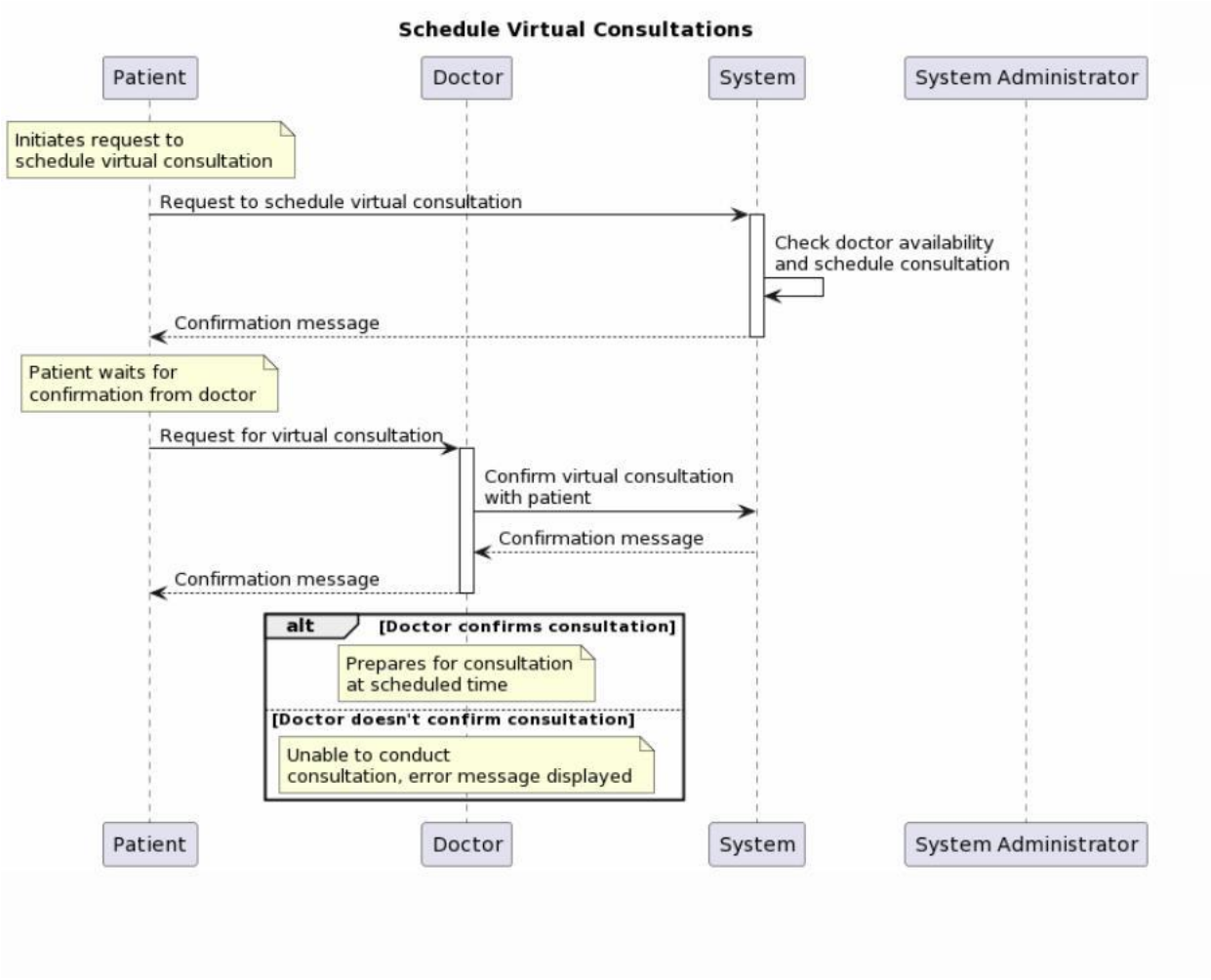
8) Join Wellness Challenges:



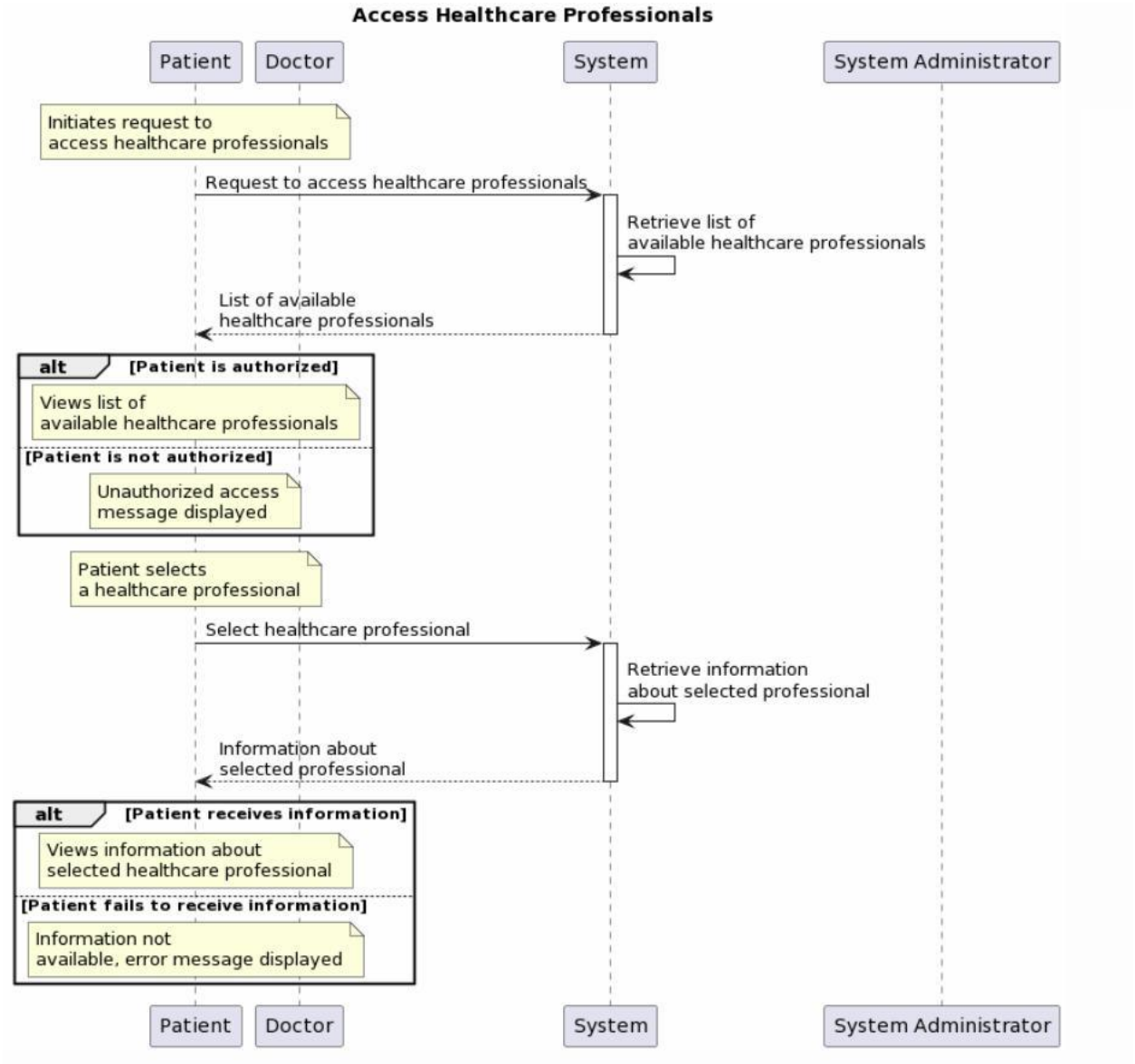
9) Receive Health Predictions:



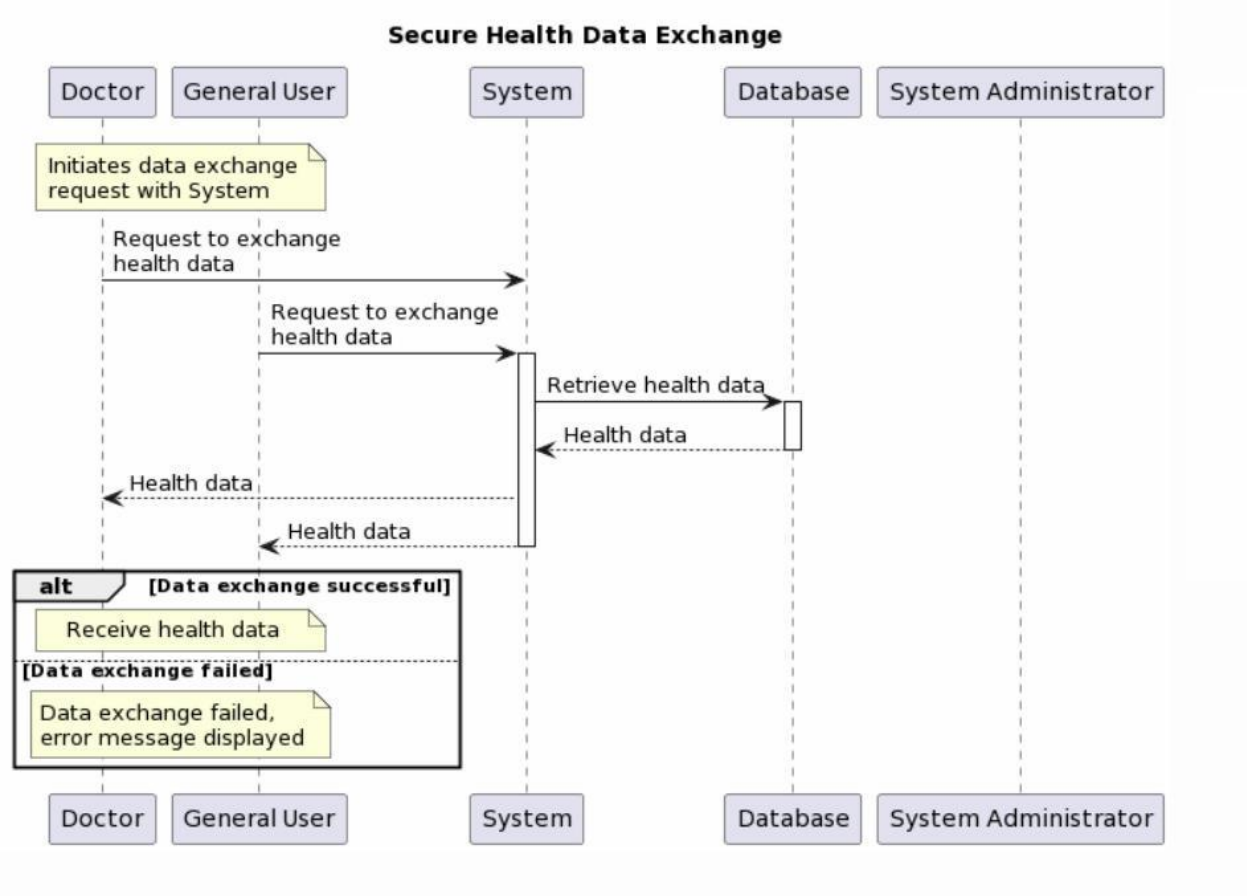
10) Schedule Virtual Consultations:



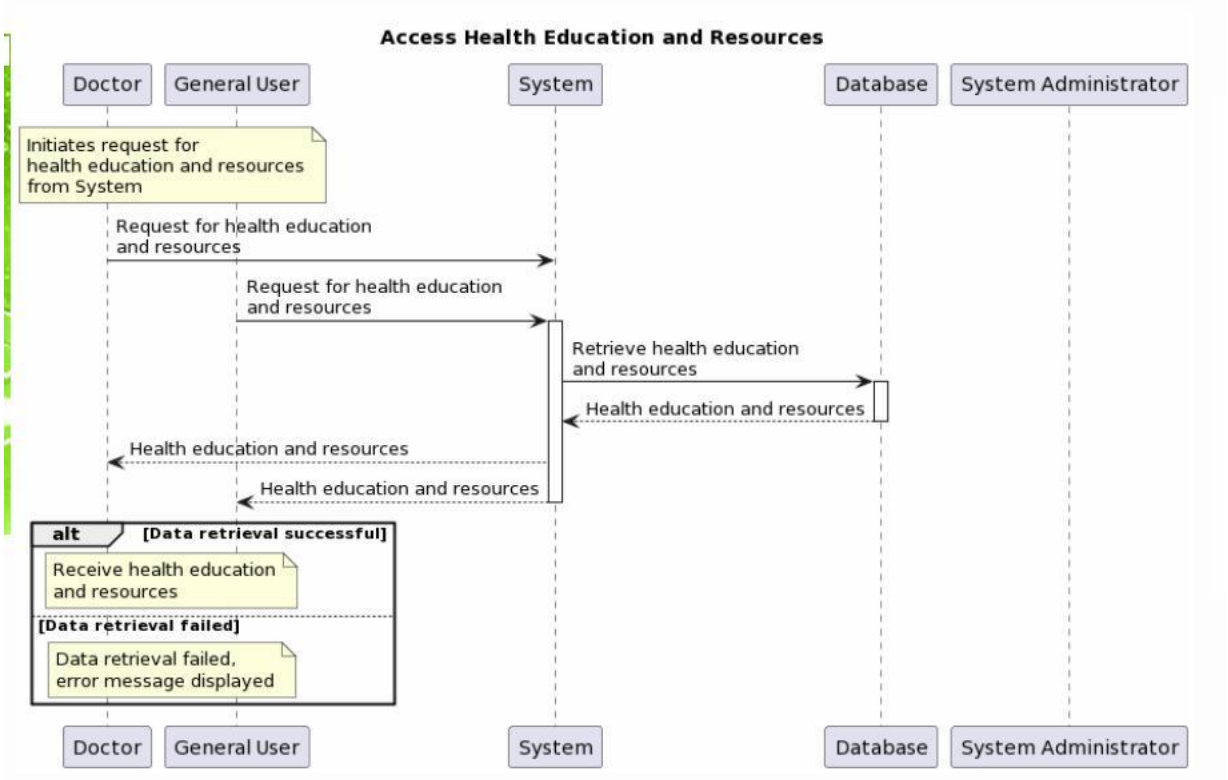
11) Access Healthcare Professionals:



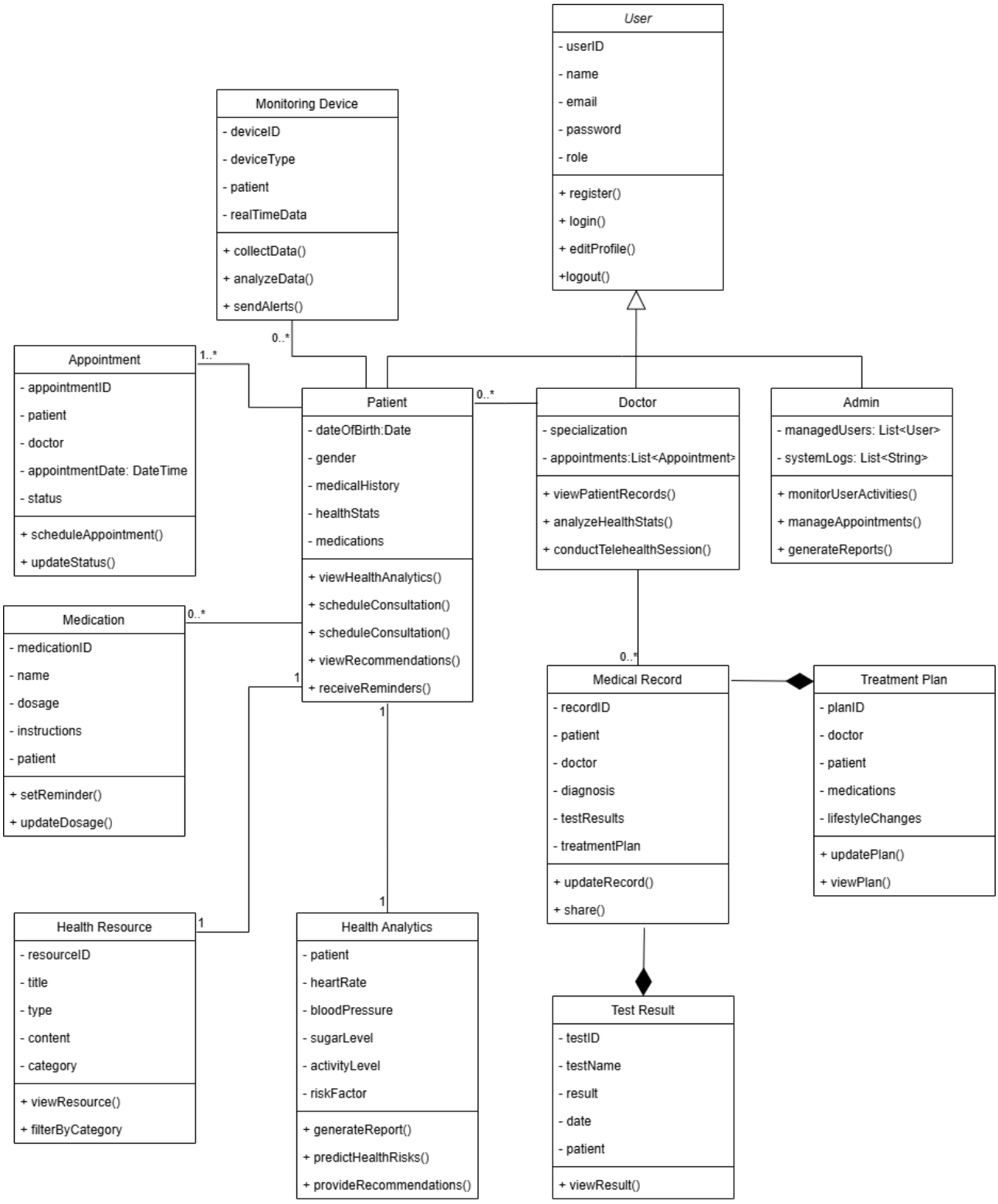
12) Secure Health Data Exchange:



13) Access Health Education and Resources:



6.4 Class Diagram



Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>

1. Sprint Backlog for Sprint 1

a. User Registration

b. User Story
As a user, I want to register on Health Hub so that I can access and view my health analytics, enabling me to make informed decisions and maintain my well-being effectively
Sub User Story
- As a user, I want to enter my email, username, and password during registration so that I can create an account.
- As a user, I want to receive a verification email so that I can confirm my account and start using Health Hub.
- As a user, I want to log in after registration using my credentials so that I can access my account.

a. Login with Specified Role

b. User Story

As a user of Health Hub, I want to log in with my specified role and password so that I can access personalized features and functionalities based on my role within the healthcare ecosystem.

Sub User Story
- As a user, I want to enter my email and password so that I can securely log in to my account.
- As an admin, I want to access admin dashboard after logging in so that I can manage users and monitor system activity.
- As a doctor, I want to access patient records and health analytics after logging in so that I can provide informed medical advice.
- As a patient, I want to see my personal health analytics after logging in so that I can track my health progress.

a. View Health Analytics

b. User Story

As a user of HealthHub, I want the platform to utilize advanced technologies to provide tailored health recommendations based on predictive health analytics.

Sub User Story

- As a user, I want HealthHub to analyze my past health data and lifestyle habits so that it can generate personalized health recommendations.
- As a user, I want the system to notify me if my health trends indicate a risk of chronic disease so that I can consult a doctor in advance.
- As a user, I want HealthHub to suggest diet and exercise plans based on my health condition so that I can improve my well-being.
- As a user, I want the system to provide mental health insights based on my sleep and stress patterns so that I can manage my mental well-being effectively.
- As a user, I want to compare my current health analytics with past data so that I can track improvements or deteriorations over time.

a. Receive Health Status and Risk Factors

b. User Story

As a user of Health Hub, I want to view my health status and risk factors so that I can understand my current health condition and take preventive measures accordingly.
Sub User Story

- As a user, I want to input my basic health details (age, weight, height, etc.) so that the system can assess my health status.
- As a user, I want to see my calculated BMI and general health category so that I can understand if I am underweight, normal, or overweight.
- As a user, I want to enter my medical history so that the system can provide me with personalized health insights.
- As a user, I want to receive an analysis of potential health risks (e.g., heart disease, diabetes, etc.) so that I can take preventive measures.
- As a doctor, I want to access my patient's health status and risk factors so that I can provide informed medical advice.
- As a user, I want my health data to be stored securely so that I can track my progress over time.

a. Access Telehealth Services

b. User Story

As a user, I want to access healthcare professionals through telehealth so that I can book appointments without needing to leave my home.

Sub User Story
- As a user, I want to browse a list of available doctors and specialists so that I can choose the right professional for my needs.
- As a user, I want to see available time slots for each doctor so that I can book a convenient appointment.
- As a user, I want to book an appointment through an easy-to-use scheduling system so that I can secure a consultation without hassle.
- As a user, I want the option to cancel or reschedule my telehealth appointment so that I can adjust my schedule if needed.
- As a user, I want to securely share my past medical records and test results with the doctor during the telehealth session so that they can provide accurate advice.

a. User Telehealth Consultation Scheduling

b. User Story

As a user of HealthHub, I want to schedule virtual consultations with healthcare professionals through the platform's telehealth feature to obtain medical advice and treatment conveniently.

Sub User Story
- As a user, I want to browse available healthcare professionals based on specialization and availability so that I can find the right doctor for my needs.
- As a user, I want to view a doctor's profile, including qualifications and experience so that I can make an informed decision before scheduling a consultation.
- As a user, I want to check available time slots for each doctor so that I can book a consultation at a convenient time.
- As a user, I want the ability to reschedule or cancel my telehealth consultation so that I can adjust my appointment if necessary.

- As a user, I want to join my telehealth consultation via a secure video call within the platform so that I can consult with my doctor remotely.

- As a user, I want to receive a summary of my consultation, including prescriptions and treatment recommendations so that I can follow up on my healthcare plan.
- As a user, I want to view a history of my past telehealth consultations so that I can track my medical interactions over time.

a. Securely Store Medical Records

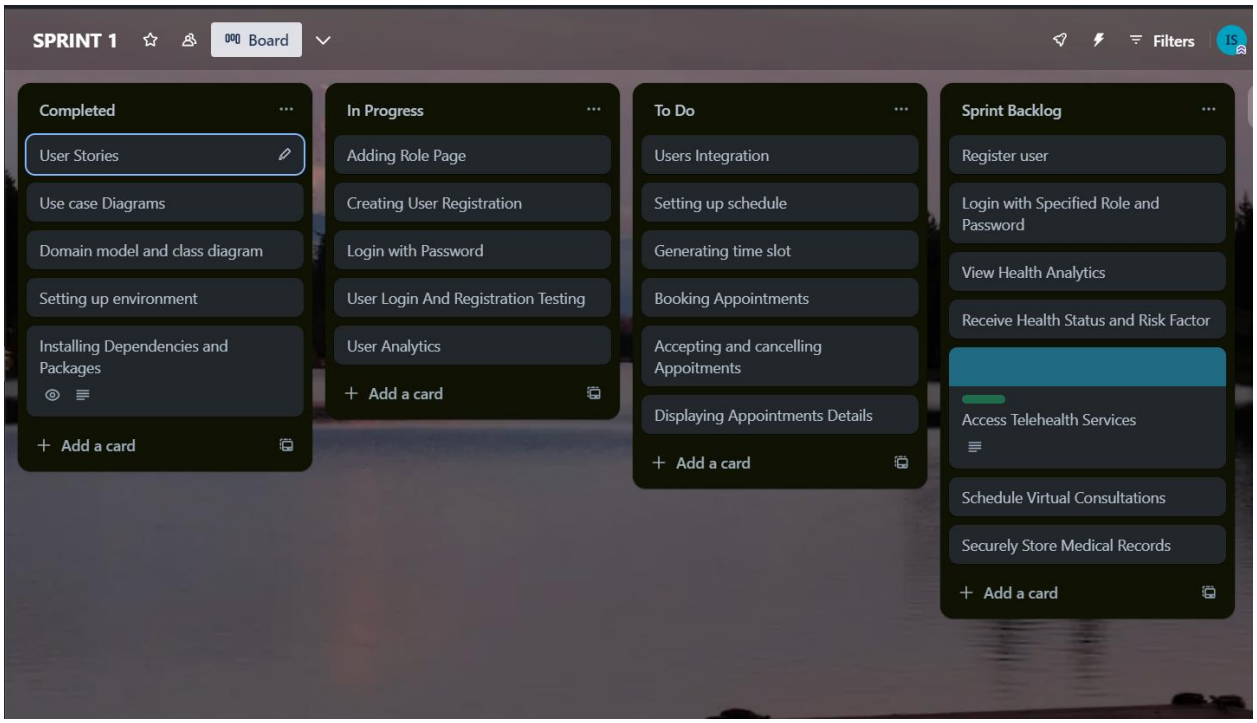
b. User Story

As a user of Health Hub, I want to securely store my medical records so that I can access them anytime and share them with healthcare providers when needed.

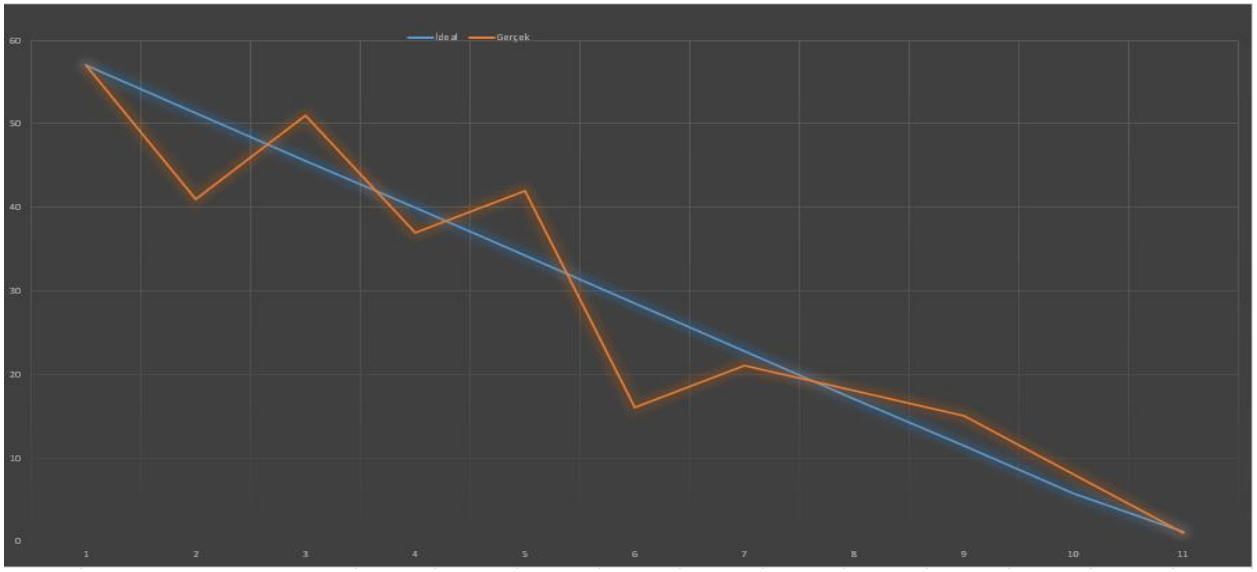
Sub User Story

- As a user, I want to upload my medical records (prescriptions, test results, reports, etc.) so that I can keep them in one place for easy access.
- As a user, I want my medical records to be stored securely so that my sensitive health information remains private.
- As a user, I want to categorize my records (e.g., Diagnosis, Prescriptions) so that I can find specific documents easily.
- As a doctor, I want to request access to my patient's medical records so that I can review their medical history and provide informed treatment.
- As a user, I want to update or delete my outdated medical records so that my stored information remains relevant and accurate.
- As a user, I want to be able to back up my medical records so that I don’t lose important health information due to system failures.
- As an admin, I want to ensure compliance with healthcare data protection regulations so that user data remains legally protected.

Trello Board



Product Burndown Chart



Code Screenshots (In Progress)

Login

```
import React, { useState } from 'react';
import axios from 'axios';
import { Link } from 'react-router-dom';
import { useNavigate } from 'react-router-dom';
import Button from '@mui/material/Button';
import TextField from '@mui/material/TextField';
import Card from '@mui/material/Card';
import CardContent from '@mui/material/CardContent';
import Box from '@mui/material/Box';
import { makeStyles } from '@mui/styles';
import Typography from '@mui/material/Typography';

const useStyles = makeStyles({
    root: {
      minHeight: '100vh',
      display: 'flex',
      justifyContent: 'center',
      alignItems: 'center',
      backgroundColor: '#001f3f',
    },
    card: {
      backgroundColor: '#f5f5f5',
      boxShadow: '0px 0px 20px rgba(0, 0, 0, 0.1)',
      borderRadius: '10px',
      maxWidth: '400px',
      padding: '20px',
      textAlign: 'center',
      transition: 'transform 0.3s ease-in-out',
      '&:hover': {
        transform: 'scale(1.05)',
      },
    },
  });


const Login = ({ selectedOption }) => {
    const classes = useStyles();
    const [formData, setFormData] = useState({
        email: '',
```

```
        password: ''
    });

    const navigate = useNavigate();

    const handleLogin = async (e) => {
        e.preventDefault();

        try {
            const response = await axios.post('http://localhost:3000/api/login',
formData);


            const userData = response.data;
            console.log('User Data:', userData);

            let optval = "/" + selectedOption + "Profile";
            console.log(optval);

            navigate(optval, { state: { user: userData } });

        } catch (error) {

            console.error('Login error occurred:', error);

            alert('Login failed. Please check your credentials.');
        }
    };

    const handleSignup = () => {
        let optval = "/" + selectedOption + "Signup";
        console.log(optval);
        navigate(optval, { replace: true });
    };

    return (
        <div className={classes.root}>
            <Card className={classes.card}>
                <CardContent>
                <Typography variant="h1" style={{ color: '#1976d2', marginBottom:
'20px',fontSize:'35px' }}>Login</Typography>
                    <p>Selected Option from Options component: {selectedOption}</p>
                    <form>
                        <div>
                            <TextField
                                id="email"
                                label="Email"
```

```
                                type="email"
                                value={formData.email}
                                onChange={(e) => setFormData({ ...formData, email:
e.target.value })}
                                required
                            />
                        </div>
                        <div>
                            <TextField
                                id="password"
                                label="Password"
                                type="password"
                                value={formData.password}
                                onChange={(e) => setFormData({ ...formData,
password: e.target.value })}
                                required
                            />
                        </div>
                        <Button variant="contained" onClick={handleLogin}
style={{marginTop:'11px'}}>Login</Button>
                        <Button variant="outlined" onClick={handleSignup}
style={{marginTop:'11px'}}>Sign Up</Button>
                    </form>
                </CardContent>
            </Card>
        </div>
    );
};

export default Login;
```

Admin SignUp

```
import React, { useState } from 'react';
import axios from 'axios';
import Card from '@mui/material/Card';
import CardContent from '@mui/material/CardContent';
import Button from '@mui/material/Button';
import TextField from '@mui/material/TextField';
import Typography from '@mui/material/Typography';
import { makeStyles } from '@mui/styles';

import { useNavigate } from 'react-router-dom';
```

```
import { Link } from 'react-router-dom';

const useStyles = makeStyles({
  root: {
    minHeight: '100vh',
    display: 'flex',
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#001f3f',
    paddingLeft: '10px',
    paddingRight: '10px',
  },
  card: {
    backgroundColor: '#f5f5f5',
    marginTop: '35px',
    boxShadow: '0px 0px 20px rgba(0, 0, 0, 0.1)',
    borderRadius: '10px',
    maxWidth: '400px',
    padding: '20px',
    textAlign: 'center',
    transition: 'transform 0.3s ease-in-out',
    '&:hover': {
      transform: 'scale(1.05)',
    },
  },
  title: {
    color: '#1976d2',
    fontSize:'35px',
    marginBottom: '20px',
  },
  input: {
    marginBottom: '15px',
  },
  button: {
    marginTop: '15px',
  },
});

const AdminSignup = () => {
  const navigate = useNavigate();
  const classes = useStyles();

  const [formData, setFormData] = useState({
    fullName: '',
    email: '',
    password: '',
    confirmPassword: '',
```

```
    role: 'Admin',
    contactNumber: '',
    termsAndConditions: false,
  });

  const handleChange = (e) => {
    const { name, value, type, checked } = e.target;
    setFormData((prevData) => ({
      ...prevData,
      [name]: type === 'checkbox' ? checked : value,
    }));
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    console.log('Admin Signup Data:', formData);

    try {
      const response = await axios.post('http://localhost:3000/api/signup/admin',
formData);
      console.log('Signup successful:', response.data)

      if (formData.password !== formData.confirmPassword) {
        alert('Passwords do not match!');
        return;
      }
      setFormData({
        fullName: '',
        email: '',
        password: '',
        confirmPassword: '',
        role: 'Admin',
        contactNumber: '',
        termsAndConditions: false,
      });

      navigate('/login');
    } catch (error) {
      console.error('Signup error:', error.response ? error.response.data :
error.message);
    }
  };

  return (
    <div className={classes.root}>
      <Card className={classes.card}>
        <CardContent>
```

```
          <Typography variant="h1" className={classes.title}>Admin
Signup</Typography>

          <TextField
            type="text"
            name="fullName"
            label="Full Name"
            value={formData.fullName}
            onChange={handleChange}
            required
            className={classes.input}
          />

          <TextField
            type="email"
            name="email"
            label="Email"
            value={formData.email}
            onChange={handleChange}
            required
            className={classes.input}
          />

          <TextField
            type="password"
            name="password"
            label="Password"
            value={formData.password}
            onChange={handleChange}
            required
            className={classes.input}
          />

          <TextField
            type="password"
            name="confirmPassword"
            label="Confirm Password"
            value={formData.confirmPassword}
            onChange={handleChange}
            required
            className={classes.input}
          />

          <TextField
            type="text"
            name="role"
            label="Role"
```

```
            value={formData.role}
            readOnly
            className={classes.input}
          />

          <TextField
            type="text"
            name="contactNumber"
            label="Contact Number"
            value={formData.contactNumber}
            onChange={handleChange}
            required
            className={classes.input}
          />

          <div>
            <label>
              <input
                type="checkbox"
                name="termsAndConditions"
                checked={formData.termsAndConditions}
                onChange={handleChange}
                required
              />
              Agree to Terms and Conditions
            </label>
          </div>

          <Button type="submit" variant="contained" className={classes.button}
onClick={handleSubmit} component={Link} to="/" >Signup</Button>
        </CardContent>
      </Card>
    </div>
  );
};

export default AdminSignup;
```

GitHub Repository

[huzxaifa/SE-Project]