Hong Kong Institute of Vocational Education (Tsing Yi)
Department of Information Technology

Higher Diploma in Software Engineering (IT114105)

Final Year Project (ITE4116M)

Final Report

Title: Remote Health Monitoring System

Supervisor: CHEUNG KWOK FAI BILLY

Co-supervisor: Ms Chiu

Student: Wong Ho Shun Sang
CUI Tianlang
Yau Yuk ming
Chan Hing Chak

We declare that this is a group project and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgement is made explicitly in the text, nor has any part been written for us by another person.

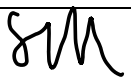
Student	Contribution to the project (%) (Total 100%)	Signature
Wong Ho Shun Sang	40%	
CUI Tianlang	26%	
Yau Yuk ming	18%	
Chan Hing Chak	16%	

Table of Contents

1.Abstract 7
2.Introduction 8
3.Driving Question 9
4. Function Requirement 11
4.1 User Registration 11
4.2 Login 11
4.3 Homepage 11
4.4 Chat Room 11
4.5 AI Bot Query 11
4.6 Remote Health Monitoring 13
4.7 Doctor-Patient Appointment 13
4.8 Real-time Chat for Diagnosis 13
4.9 Sensor Data Acquisition 13
5 Non-functional Requirements 14
5.1 Security 14
5.2 Performance 14
5.3 Usability 14
5.4 Reliability 14
5.5 Compatibility 14
5.6 Scalability 15
5.7 Compliance 15
5.8 Integration 15
6. Design of the proposed system 16
6.1 architecture of the proposed system 16
6.2 scope of the proposed system 17
6.3 description of functions provided 17
6.4 data processed by the system 18
7. Documentation for problem analysis 25
7.1 Actor description and Use case description 25
Actor description 25
8 Use case description 26
8.1 Login and Register System 26

8.2 Appointment System 27
8.3 Search system 28
8.4 Chat System 29
8.4 Sensor Data Acquisition System 30
9 Diagram 33
9.1 Use Case Diagram 33
Remote Health Monitoring System – Full System 33
Appointment System 35
Search and chatbot System 36
Chat System 37
Meeting Room System 38
Sensor Data Acquisition System 39
9.2 Class diagram 40
Login and Register System 40
Appointment System 40
Search System 41
Chat System 41
Sensor Data Acquisition System 42
9.3 Sequence diagram 43
Login and Register System 43
Appointment System 45
Chat System 49
Sensor Data Acquisition System 50
System level Sequence Diagram 52
9.4 State Machine diagram 53
Login and Register System 53
Appointment System 55
Chatbot 57
Search 58
Chat System 59
Sensor Data Acquisition System 60
10. Documentation for detailed design 62
10.1 Data design - ERD 62
10.2 Software architectural design 63

Software / Maintenances cost 69
10.3 Hardware architectural design (system design) 70
Cost of Hardware(s) and Service(s) 74
10.4 User interface design 75
Home Page 75
Login – Doctor 77
Login – User 77
Register – Doctor 78
Register – User 78
User - Home Page 79
User - Appointment 79
User - Healthy Test 81
User – Meeting Room 81
User – Chat Room 84
User – Healthy Information (English) 85
User – Healthy Information (Chinese) 86
User - Chat Bot 87
Doctor – Home Page 88
Doctor - Manage Appointment 88
Doctor – Chat Room 89
Doctor – Meeting Room 91
Doctor – Sensor Record 94
11. Critical Evaluation 95
11.1 Difficulties encountered 95
11.2 Imitations of the proposed system 95
11.3 Potential difficulties 95
12.All test cases and test data 96
12.1 Login 96
12.2 Make Appointment 97
12.3 Healthy Test 98
12.4 Chat Room 99
12.5 Meeting 100
12.6 Chatbot 101
Test Case ID 101

TC006 101
12.7 Search Page 101
12.8 Manage Appointment 102
12.9 View Sensor Record 103
13. Project plan 104
14. Conclusion 106
15. Project Log 108
16. References 110
17. Program listing 111
Home Page 111
Index.html 111
Home.js 113
Login 118
Customer.php 118
Doctor.php 120
RegisterDoctor.php 123
Customer.js 126
Customer.js 127
Meeting Room 131
Meeting Room.html 131
Chat Bot 133
ChatBot.html 133
ChatBot.css 134
Script.js 138
18. User guide and installation guide 140
Step 1: Download and zip the file 140
Step 2: Put the zipped file into the path 140
Step 3: Open the Xampp and Turn on the Apache and MySQL 141
Step4: Open the Admin and insert the mysql file 142
Step 5: Go to the browser and paste the link: http://127.0.0.1/FYP web/ 144
Step6: Click the button ‘Sign Up New’ to sign up 144
Step7: Login or sign up a new account (For Public) 145
Step8: Make Appointment (For Public) 147
Step9: Click the button chat to communicate with doctor (Public) 148

Step10: Click the button meeting to communicate with doctor (Public) 150
Step11: Click the Healthy Information to Search the information and use the chat bot 151
Step12: Login or sign up a new account (For Doctor) 153
Step13: Management appointment (Doctor) 155
Step14: Sensor Healthy test (Public) 156
Step15: View Sensor Record (Doctor) 157

1.Abstract

The objective of this project is to develop a comprehensive Remote Health Monitoring System that addresses the limitations of traditional healthcare systems. The system aims to provide individuals with a user-friendly platform for remote health monitoring, appointment scheduling, personalized health suggestions, and access to their health history. It also enables doctors to view extended patient health information and efficiently manage their patient list. The proposed solution follows a client-server architecture, with the client side developed using HTML, CSS, JavaScript, and jQuery, and the server side implemented using PHP and XAMPP. The system includes features such as a secure login system, sensor data acquisition, doctor information, appointment scheduling, online meetings, patient list management, health suggestions, and patient health history tracking. The project will be divided into various stages, including project initiation, requirements gathering and analysis, development, testing, deployment, documentation, and training. The main deliverables of the project include an SQL file for the database, a website for the user interface, and a comprehensive report documenting the project.

2.Introduction

This document provides a general description of the structure and objectives of the Remote Health Monitoring System project. The project aims to develop a comprehensive system that addresses the limitations of traditional healthcare systems by enabling remote health monitoring and improved communication between patients and doctors.

The ability to monitor certain aspects of a patient's health from their own home has become an increasingly popular telehealth option. Remote patient monitoring lets providers manage acute and chronic conditions. And it cuts down on patients' travel costs and infection risk.

The Remote Health Monitoring System is designed to provide individuals with a user-friendly platform for monitoring their health remotely. It allows users to track vital signs, such as heart rate, blood pressure, and temperature, using various sensors and devices. The system also includes features for appointment scheduling, personalized health suggestions, and access to the users' health history.

On the healthcare provider side, the system enables doctors to view extended patient health information and efficiently manage their patient list. It facilitates online meetings between doctors and patients, allowing for remote consultations and follow-ups. By leveraging technology, the system aims to enhance the efficiency and effectiveness of healthcare delivery.

The project follows a client-server architecture, with the client side developed using HTML, CSS, JavaScript, and jQuery. The server side is implemented using PHP and XAMPP, which provides a robust and secure foundation for data management and communication. The system utilizes a database to store patient information, sensor data, appointment details, and other relevant data.

By combining remote health monitoring, appointment management, and improved communication channels, the Remote Health Monitoring System aims to empower individuals to take a proactive role in managing their health while providing healthcare providers with valuable tools for delivering high-quality care remote.

3.Driving Question

Explain how the project answers this question: “How can Software Engineering techniques be used to develop software systems for supporting human activities?”

This system utilizes software engineering techniques to create a comprehensive solution that supports and enhances various human activities related to healthcare.

- The Remote Health Monitoring System addresses the need for individuals to remotely monitor their health and collaborate effectively with healthcare providers. By leveraging software engineering techniques, the system provides a user-friendly platform that supports human activities in the following ways:

- Remote Health Monitoring: The system utilizes sensors and devices to collect vital health data from individuals. Software engineering techniques are applied to design and develop the data acquisition and processing components, ensuring accurate and reliable measurement of health parameters. This enables individuals to actively monitor their health from the comfort of their homes or any remote location.

- Appointment Scheduling: The system incorporates appointment scheduling functionality, allowing users to conveniently book appointments with healthcare providers. Software engineering techniques are employed to develop a user-friendly interface for scheduling appointments, managing availability, and handling conflicts. This streamlines the process of setting up consultations and supports efficient coordination between patients and doctors.

- Personalized Health Suggestions: The system provides personalized health suggestions based on the collected health data. Software engineering techniques are utilized to analyze the data, apply algorithms and models, and generate meaningful insights and recommendations. This helps individuals make informed decisions regarding their health and well-being.

- Access to Health History: The system enables individuals to access and manage their health history. Software engineering techniques are employed to design and implement a secure database that stores patient health records. This allows users to review their medical history, track progress, and share relevant information with healthcare providers, supporting continuity of care.

- Communication and Collaboration: The system facilitates online meetings and communication channels between patients and doctors. Software engineering techniques are utilized to develop real-time communication features, ensuring secure and reliable interactions. This supports effective collaboration, remote consultations, and follow-ups, enhancing the overall healthcare experience.

Through the application of software engineering techniques, the Remote Health Monitoring System addresses the driving question by developing a software solution that supports and enhances human activities related to healthcare. It leverages technology to enable remote health monitoring, appointment scheduling, personalized health suggestions, and improved communication, ultimately empowering individuals to take an active role in managing their health while facilitating efficient healthcare delivery.

4. Function Requirement

4.1 User Registration

The system should provide a user-friendly registration process for patients, allowing them to enter their personal information accurately. The registration process should validate the entered information and ensure its integrity. Upon successful registration, patients should receive a confirmation notification or email, ensuring a smooth onboarding experience.

- The system should provide registration functionality for patients.
- Patients should register by providing necessary information through a hyperlink on the Login page.

4.2 Login

The login functionality should offer a secure and seamless access point for users (doctors and patients) to their respective accounts. Users should be able to enter their login credentials, such as username and password, and the system should authenticate the credentials to provide access. In case of incorrect credentials, appropriate error messages should be displayed to assist users in troubleshooting.

- The system should have a login functionality for both doctors and patients.
- Patients and doctors should be able to access their respective accounts by entering their credentials.

4.3 Homepage

The homepage should serve as a central hub for users, displaying relevant and up-to-date information in an easily accessible manner. It should provide a visually appealing layout, featuring healthcare news, articles, notifications, and other pertinent information. Users should be able to navigate through different sections of the homepage to find the desired information efficiently.

- The system should display a homepage HTML where users can view relevant news and information.
- Users should be able to access the homepage after successful login.

4.4 Chat Room

The Chat Room functionality should facilitate real-time communication between patients and doctors. It should offer an intuitive and user-friendly messaging interface, allowing users to exchange messages seamlessly. Users should be able to view message history, receive notifications for new messages, and have basic formatting options to enhance communication effectiveness.

- The system should provide a Chat Room functionality for patients and doctors to engage in remote real-time conversations.
- Patients should be able to access the Chat Room through the navigation bar after login.

4.5 AI Bot Query

The AI bot functionality should enable patients to obtain information and answers to their queries. The system should provide a conversational interface where patients can ask

questions or enter keywords related to their health concerns. The AI bot should employ natural language processing techniques to understand and provide relevant responses. It should support interactive conversations, allowing users to ask follow-up questions or seek clarifications.

- The system should allow patients to query information from an AI bot.
- Patients should be able to access the AI bot functionality through the navigation bar after login.

4.6 Remote Health Monitoring

The system should support the remote collection and storage of patients' health data. Patients should be able to upload their body readings or health measurements obtained from wearable devices or medical equipment. The system should securely store and associate the data with the respective patient's profile, ensuring privacy and data integrity. The uploaded data should be easily accessible for future reference by both patients and doctors.

- The system should enable patients to upload their body readings from remote devices.
- Patients should be able to log in to the system and upload their body readings for record-keeping.

4.7 Doctor-Patient Appointment

The appointment scheduling functionality should streamline the process of scheduling and managing appointments between doctors and patients. Patients should be able to view doctors' availability, select preferred dates and times, and book appointments based on their convenience. The system should provide confirmation notifications to patients and update doctors' schedules, accordingly, ensuring efficient appointment management.

- The system should facilitate doctor-patient appointments.
- Patients should be able to log in to the system and schedule appointments with doctors.

4.8 Real-time Chat for Diagnosis

The real-time chat functionality should enable doctors to engage in live conversations with patients for diagnosis and treatment purposes. Doctors should be able to ask pertinent questions, request additional information or tests, and provide medical advice. The chat interface should support multimedia elements, such as image sharing, to facilitate accurate diagnosis and enhance the doctor-patient communication experience.

- The system should enable doctors to have real-time conversations with patients for diagnosis purposes.
- Doctors should be able to use the Chat Room functionality to interact with patients and analyze their condition.

4.9 Sensor Data Acquisition

The system is integrated with sensors and devices to collect patient health data, including heart rate, temperature, and skin conductance. This data is then stored in a database, allowing doctors to remotely and in real-time monitor the patient's health status. This integration of the system with various sensors and equipment empowers healthcare providers to effectively track and manage the well-being of their patients, even from a distance, enabling more comprehensive and personalized care.

- The system should enable doctors to view patient health data instantly.
- Patients should be able to view their historical health data.

5 Non-functional Requirements

5.1 Security

The system should implement robust security measures to protect user data and ensure privacy. It should employ encryption techniques to secure data transmission and storage. Access to sensitive information should be restricted to authorized personnel, and the system should comply with relevant data protection regulations to safeguard user confidentiality.

- The system should ensure the security and confidentiality of user data, including personal information and health records.
- Data transmission and storage should be encrypted to protect patient privacy.

5.2 Performance

The system should deliver a responsive and high-performance experience, ensuring minimal delays in data retrieval, messaging, and other functionalities. It should be able to handle concurrent user interactions and maintain optimal performance even during peak usage periods, providing a smooth and efficient user experience.

- The system should provide a responsive and seamless user experience, ensuring minimal delays in chat conversations and data retrieval.
- The system should be capable of handling concurrent user interactions and processing data efficiently.

5.3 Usability

The user interface should be intuitive, visually appealing, and easy to navigate. Users should be able to understand and use the system's features without extensive training or assistance. Clear instructions, tooltips, and help sections should be available to guide users through different functionalities and aid in their overall usability.

- The system should have a user-friendly interface that is intuitive and easy to navigate.
- The system should provide clear instructions and guidance for users to perform actions effectively.

5.4 Reliability

The system should be reliable and available for users. It should have backup and recovery mechanisms in place to minimize downtime and ensure data integrity. Regular maintenance and updates should be performed to address any potential issues promptly and maintain system reliability.

- The system should be available and accessible to users reliably, with minimal downtime or system failures.
- The system should have backup and recovery mechanisms in place to prevent data loss in case of unexpected events.

5.5 Compatibility

The system should be compatible with various web browsers and devices, ensuring accessibility for a wide range of users. It should be designed to adapt to different screen sizes and resolutions, providing a consistent user experience across multiple platforms.

- The system should be compatible with various web browsers and devices to ensure accessibility for users.
- The system should be designed to support different screen sizes and resolutions.

5.6 Scalability

The system architecture should be scalable to accommodate a growing number of users and increasing data volume. It should be able to handle additional user load without significant performance degradation. Scalability measures, such as load balancing and cloud-based infrastructure, can be implemented to ensure efficient resource utilization and system stability.

- The system should be designed to accommodate a growing number of users and data without compromising performance.
- The system architecture should be scalable to handle increased user load and data volume.

5.7 Compliance

The system should comply with relevant data protection and privacy regulations, such as GDPR or HIPAA, depending on the jurisdiction. It should adhere to industry standards and best practices for healthcare information systems to ensure the secure handling and storage of sensitive user data.

- The system should comply with relevant data protection and privacy regulations, such as GDPR or HIPAA, depending on the jurisdiction.
- The system should adhere to industry standards and best practices for healthcare information systems.

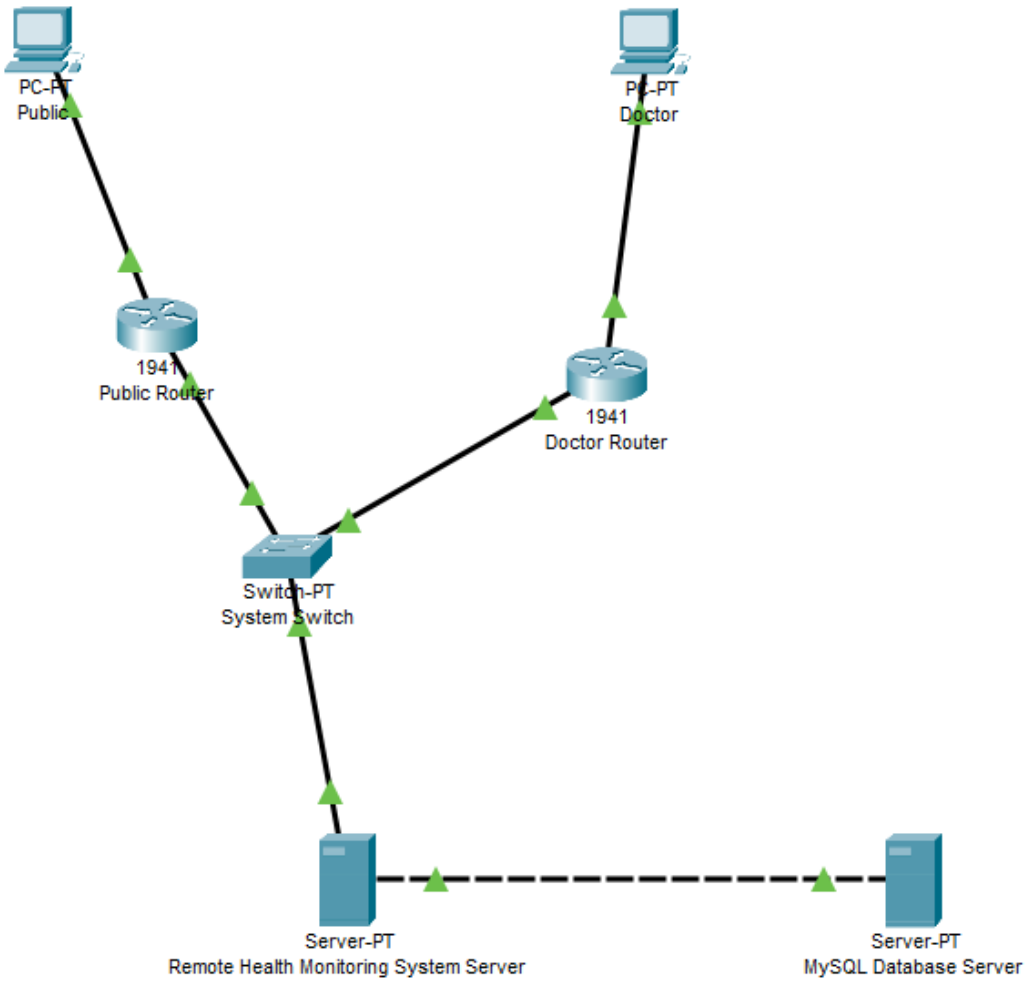
5.8 Integration

The system should be designed to integrate with external systems or devices, such as electronic health record systems or remote health monitoring devices. It should support seamless data exchange and interoperability, enhancing the overall functionality and usefulness of the system within the healthcare ecosystem.

- The system should be capable of integrating with external systems or devices for data exchange, such as remote health monitoring devices or electronic health record systems.

6. Design of the proposed system

6.1 architecture of the proposed system



There are two networks, one for the public and one for the private.

The Public Router connects to the internet via the Public IP address 1941.
The Public Router is connected to the Switch-PT, which is in turn connected to Server-PT and the Doctor's network via the Doctor Router.
The System Switch is connected to the Remote Health Monitoring System Server and the MySQL Database Server.
There are two servers connected to the System Switch. The first server is called Server-PT, which is the Remote Health Monitoring System Server. This server is responsible for collecting and storing health data from remote patients. The second server is called the MySQL Database Server, which stores the data collected by Server-PT.
The PL-PT link connects the Doctor's network to the Remote Health Monitoring System Server.

Overall, this network setup allows the doctor's network to connect to the Remote Health Monitoring System Server, which stores and manages health data for patients. The MySQL Database Server is used to store this data, and the System Switch and Switch-PT are used to connect various devices within the network. The Public Router connects the network to the internet, allowing remote access to the system.

6.2 scope of the proposed system

The scope of the proposed solution includes the development of a client-server architecture-based system. The client side is developed using HTML, CSS, JavaScript, and jQuery, providing a user-friendly interface for individuals to interact with the system. The server side is implemented using C# and XAMPP, ensuring robust data management and communication.

6.3 description of functions provided

1. Secure Login System: Users can securely log in to the system using their credentials, ensuring data privacy and security.
2. Doctor Information: Doctors have access to patient details, medical history, and appointment schedules for efficient patient management.
3. Appointment Scheduling: Users can schedule appointments with healthcare providers based on availability and convenience.
4. Online Meetings: The system facilitates online meetings between doctors and patients, enabling remote consultations and follow-ups.
5. Patient List Management: Doctors can efficiently manage their patient list, track appointments, and update patient information.
6. Health Suggestions: The system provides personalized health suggestions based on collected data and analysis.
7. Patient Health History Tracking: Users can access and manage their health history, including past appointments, test results, and medications.
8. Sensor Data Acquisition: The system integrates with sensors and devices to collect vital health data, such as heart rate, blood pressure, and temperature.

6.4 data processed by the system

doctor table

	#	名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註	額外資訊	動作
☐	1	doctorID	int(10)			否	無			修改 刪除 更多
☐	2	name	varchar(255)	utf8_general_ci		否	無			修改 刪除 更多
☐	3	types	varchar(255)	utf8_general_ci		否	無			修改 刪除 更多
☐	4	tel	int(8)			否	無			修改 刪除 更多
☐	5	email	varchar(255)	utf8_general_ci		否	無			修改 刪除 更多
☐	6	password	varchar(10)	utf8_general_ci		否	無			修改 刪除 更多

Mata data of doctor

Data Item Name	Data Type	Data length	Description of Data Item
doctorID	integer	10	Doctor Identification Number
name	character	255	Doctor **Name**
types	character	255	Doctor types
tel	integer	8	Doctor phone number
email	email	255	Doctor email
password	alphanumeric	10	Doctor Account Password

users table

	#	名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註	額外資訊	動作
☐	1	userID	int(10)			否	無			修改 刪除 更多
☐	2	userName	varchar(50)	utf8_general_ci		否	無			修改 刪除 更多
☐	3	Email	varchar(255)	utf8_general_ci		否	無			修改 刪除 更多
☐	4	tel	varchar(8)	utf8_general_ci		否	無			修改 刪除 更多
☐	5	password	varchar(8)	utf8_general_ci		否	無			修改 刪除 更多

Data Item Name	Data Type	Data length	Description of Data Item
userID	integer	10	Public Identification Number
userName	character	50	Public Name
Email	email	255	Public email
tel	integer	8	Public phone number

sensor table-GSR sensor

	#	名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註	額外資訊	動作
☐	1	id	int(11)			否	無		AUTO_INCREMENT	修改 刪除 更多
☐	2	gsr	int(11)			否	無			修改 刪除 更多
☐	3	datetime	datetime			否	current_timestamp()			修改 刪除 更多

Mata data of sensor table-GSR sensor

Data Item Name	Data Type	Data length	Description of Data Item
id	integer	11	Public Identification Number
gsr	integer	11	GSR sensor data
datetime	datetime		Current data & time

heartrate table- heart rate sensor

#	名稱	類型	編碼與排序	屬性	空值 (Null)	預設值	備註	額外資訊	動作
1	id	int(11)			否	無		AUTO_INCREMENT	修改 刪除 更多
2	rate	int(11)			否	無			修改 刪除 更多
3	datetime	datetime			否	current_timestamp()			修改 刪除 更多

Data Item Name	Data Type	Data length	Description of Data Item
id	integer	11	Public Identification Number
rate	integer	11	Heart rate sensor data
datetime	datetime		Current data & time

temperature_sensor table- temperature sensor

#	名稱	類型	編碼與排序	屬性	空值 (Null)	預設值	備註	額外資訊	動作
1	id	int(11)			否	無		AUTO_INCREMENT	修改 刪除 更多
2	Celsius	float			否	無			修改 刪除 更多
3	datetime	datetime			否	current_timestamp()			修改 刪除 更多

Mata data of temperature_sensor

Data Item Name	Data Type	Data length	Description of Data Item
id	integer	11	Public Identification Number
Celsius	float		temperature sensor data°C
datetime	datetime		Current data & time

history_sensor table- get data from 3 types of sensors(for using)

#	名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註	額外資訊	動作
1	id	int(11)			否	無		AUTO_INCREMENT	修改 刪除 更多
2	userName	varchar(50)	utf8mb4_general_ci		否	無			修改 刪除 更多
3	datetime	datetime			否	current_timestamp()			修改 刪除 更多
4	tem_C	float			否	無			修改 刪除 更多
5	heartrate	int(11)			否	無			修改 刪除 更多
6	gsr	int(11)			否	無			修改 刪除 更多

Mata data of history_sensor table

Data Item Name	Data Type	Data length	Description of Data Item
id	integer	11	Public Identification Number
userName	character	50	Public **Name**
datetime	datetime		Current data & time
tem_C	float		temperature sensor data°C
heartrate	integer	11	Heart rate sensor data
gsr	integer	11	GSR sensor data

Appointment table

#	名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註	額外資訊	動作
1	id	int(10)			否	無		AUTO_INCREMENT	修改 刪除 更多
2	name	varchar(255)	utf8mb4_general_ci		否	無			修改 刪除 更多
3	date	date			否	無			修改 刪除 更多
4	time	time(6)			否	無			修改 刪除 更多
5	VisitMethod	varchar(255)	utf8mb4_general_ci		否	無			修改 刪除 更多
6	doctorName	varchar(255)	utf8mb4_general_ci		否	無			修改 刪除 更多

Mata data of appointment

Data Item Name	Data Type	Data length	Description of Data Item
id	integer	10	Appointment Identification Number
name	character	255	**Username(Public)**
date	date		Meeting **Date**
time	time	6	Meeting **Time**
VisitMethod	character	255	Meeting method(face to face or online)
doctorName	character	255	Doctor Name

member Table

資料表結構 關聯檢視

#	名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註	額外資訊	動作
1	user_id	int(11)			否	無		AUTO_INCREMENT	修改 刪除 更多
2	unique_id	int(255)			否	無			修改 刪除 更多
3	fname	varchar(255)	utf8mb4_general_ci		否	無			修改 刪除 更多
4	lname	varchar(255)	utf8mb4_general_ci		否	無			修改 刪除 更多
5	email	varchar(255)	utf8mb4_general_ci		否	無			修改 刪除 更多
6	password	varchar(255)	utf8mb4_general_ci		否	無			修改 刪除 更多
7	img	varchar(255)	utf8mb4_general_ci		否	無			修改 刪除 更多
8	status	varchar(255)	utf8mb4_general_ci		否	無			修改 刪除 更多

Mata data of member

Data Item Name	Data Type	Data length	Description of Data Item
user_id	integer	11	user Identification Number
unique_id	int	255	Identify user
fname	character		User first name
lname	character	6	User last name
email	character	255	email
password	character	255	User password
img	character	255	User icon image
status	character	255	User status

messages table

資料表結構 關聯檢視

#	名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註	額外資訊	動作
1	msg_id	int(11)			否	無		AUTO_INCREMENT	修改 刪除 更多
2	incoming_msg_id	int(255)			否	無			修改 刪除 更多
3	outgoing_msg_id	int(255)			否	無			修改 刪除 更多
4	msg	varchar(1000)	utf8mb4_general_ci		否	無			修改 刪除 更多

Mata data of messages

Data Item Name	Data Type	Data length	Description of Data Item
msg_id	integer	11	User message Number
incoming_msg_id	int	255	Income message record id
outgoing_msg_id	int	255	Out go message record id
msg	character	1000	Message content

7. Documentation for problem analysis

7.1 Actor description and Use case description

Actor description

Doctor	Doctors rely on the system to streamline patient care, enhance communication with patients, and leverage data-driven insights to provide comprehensive healthcare services.
Public (Patient)	The public represents individuals who interact with the healthcare system as patients, caregivers, or general users. They utilize the system to access healthcare services, manage their health information, and engage in communication with healthcare providers.

8 Use case description

8.1 Login and Register System

Use case name	Register
Case ID	UC-100
actor	Public and Doctor
Brief description	All user can use the system to register an account. The system will save the account.
Preconditions	Users must follow the password format.
Post-conditions	Users can open the system
Flow of events	1. Users click button to their role signup page. 2. Users input the personal information 3. Users input the correct password format. 4. Users click the signup button.
Alternative flow and exception	The user may input wrong password format. The system will show the password format message. The user may miss an input data.
Priority	high
Non-behavioral requirement	The password should be kept secretly by the user.

Use case name	Login
Case ID	UC-200
actor	Public and Doctor
Brief description	All user can use their account to login to the system.
Preconditions	Users must have an account.
Post-conditions	User can login to the system and start using it
Flow of events	1. Users click button to their role login page. 2.Users input correct username. 3. Users input correct password.
Alternative flow and exception	The user may input wrong password or username. The system will alert a message to tell user
Priority	high
Non-behavioral requirement	The password should be kept secretly by the user.

8.2 Appointment System

Use case name	Make Appointment
Case ID	UC-300
actor	Public
Brief description	The public can fill the form to make an appointment
Preconditions	
Post-conditions	
Flow of events	1. Public login to the system. 2. Public click ‘Appointment’ button to appointment page. 3. Public view doctor information. 4.Public fill data in the form. 5. Public click submit button.
Alternative flow and exception	The public may select the wrong doctor.
Priority	high
Non-behavioral requirement	The public should ask the related doctor.

Use case name	Manage Appointment
Case ID	UC-400
actor	Doctor
Brief description	Doctor can manage (edit and delete) appointment record.
Preconditions	The public make an appointment.
Post-conditions	If any changes are made to the appointment record, such as editing the data or deleting the record, the public should be notified by the doctor or the system to ensure effective communication and minimize any confusion or inconvenience.
Flow of events	1.Doctor view appointment record. 2.Doctor edit the form data. 3.Doctor click submit button. 4.Doctor click delete button to delete the outdated record or canceled record.
Alternative flow and exception	The public may select the wrong doctor.
Priority	high
Non-behavioral requirement	In case of emergencies or unforeseen circumstances, doctors should promptly reach out to the public to reschedule appointments or provide alternative arrangements.

8.3 Search system

Use case name	Search
Case ID	UC-500
actor	public
Brief description	Public can search disease infromation
Preconditions	The public enter keywords.
Post-conditions	Accurate Information: The user should have been presented with accurate, up-to-date, and relevant information about the disease they searched for.
Flow of events	1.User enter keyword 2.User Click submit button 3.System display the information about the keyword
Alternative flow and exception	The system may not have the information about the keyword
Priority	middle
Non-behavioral requirement	The system errors

8.4 Chat System

Use case name	Chatbot
Case ID	UC-600
actor	public
Brief description	Public can ask anything to chatbot
Preconditions	Public enter something.
Post-conditions	Public should be polite to the chatbot
Flow of events	1.User enter keyword 2.User Click enter button 3.chatbot display something to user
Alternative flow and exception	
Priority	high
Non-behavioral requirement	The chatbot errors.

Use case name	Chat Room
Case ID	UC-700
actor	Patient, Doctor
Brief description	Public can ask anything to chatbot
Preconditions	Public enter something.
Post-conditions	This use case describes the process of engaging in remote real-time conversation between a patient and a doctor in the Chat Room.
Flow of events	1. The patient selects the Chat Room option from the navigation bar. 2. The system displays the Chat Room interface. 3. The patient enters a message in the input field. 4. The patient clicks the "Send" button. 5. The system sends the message to the doctor. 6. The doctor receives the message in their Chat Room interface. 7. The doctor reads the message. 8. The doctor composes and enters a response in the input field. 9. The doctor clicks the "Send" button. 10. The system sends the response to the patient. 11. The patient receives the response in their Chat Room interface. 12. The patient reads the response. 13. Steps 3-12 repeat as needed for the ongoing conversation.

	14. The patient can choose to end the conversation by closing the Chat Room interface.
Alternative flow and exception	• If the doctor is not available or not logged into the system: ▪ The patient receives a notification indicating the doctor's unavailability and is prompted to try again later. • If the patient or doctor encounters a technical issue (e.g., network problem): ▪ The system displays an error message and provides instructions for troubleshooting or contacting technical support.
Priority	Medium
Non-behavioral requirement	• The Chat Room interface should provide a real-time messaging experience with minimal delay. • The conversation history should be stored securely and accessible only to authorized users.

8.4 Sensor Data Acquisition System

Use case name	Sensor data Acquisition
Case ID	UC-800
actor	Public
Brief description	Public can use heart rate, temperature, gsr and sensors to obtain health data.
Preconditions	Sensors connected to database and Wi-Fi.
Post-conditions	
Flow of events	1. Open Arduino IDE 2. The sensor is connected to the computer port. 3. The sensor is connected to Wi-Fi. 4. The sensor is connected to the database. 5. Click the upload button to compile. 6. Sensor data will be uploaded to the database. 7. Repeat the above steps with the other sensors.
Alternative flow and exception	If you select the wrong port and card, an error will occur.
Priority	high
Non-behavioral requirement	The public should stop the connection, when the compilation is complete, the public should disconnect.

Use case name	Upload sensor data
Case ID	UC-900
actor	Public
Brief description	Public can upload sensor data to database of the history sensor table.
Preconditions	The sensor has been uploaded to its matching table.
Post-conditions	
Flow of events	1. Click upload button. 2. The system will upload these 3-type sensors to the history sensor table.
Alternative flow and exception	
Priority	high
Non-behavioral requirement	

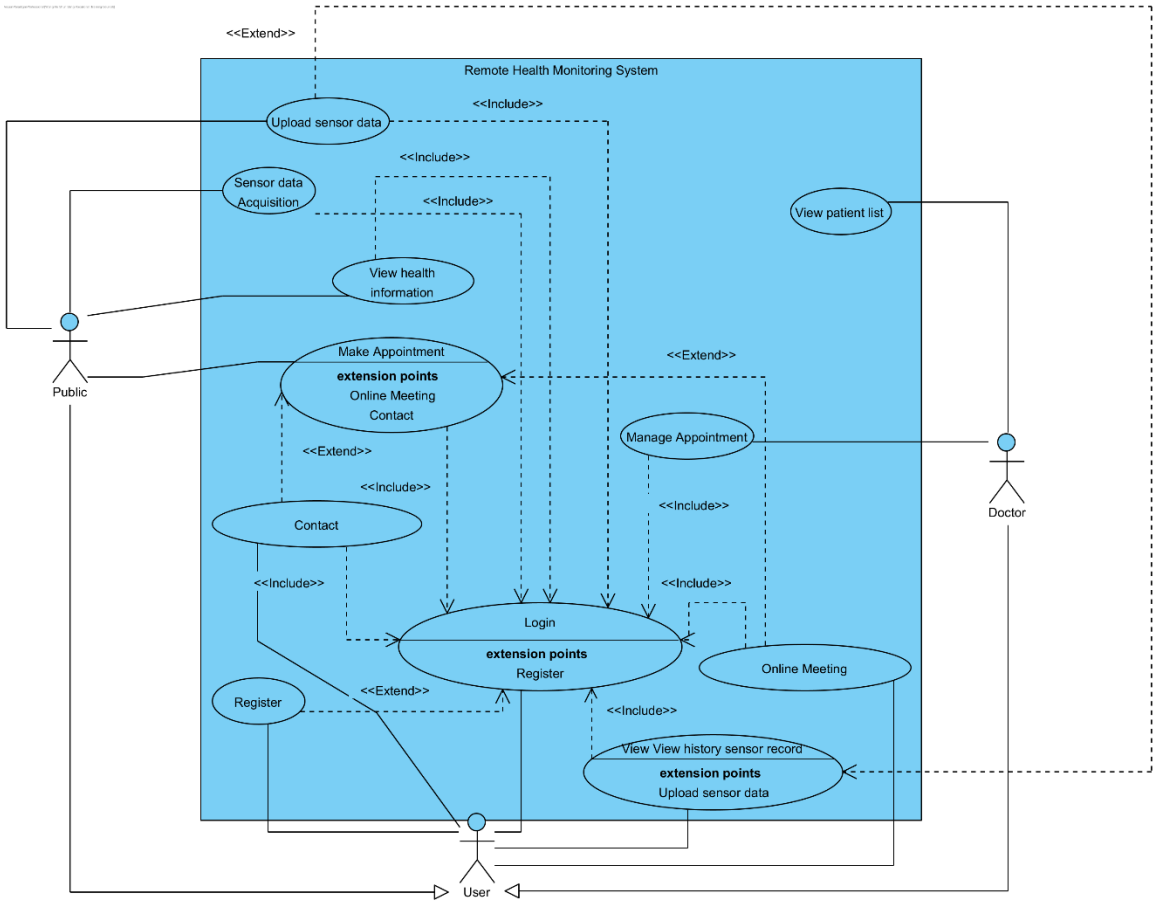
Use case name	View history sensor record
Case ID	UC-1000
actor	Public and Doctor
Brief description	Users can upload View history sensor record.
Preconditions	The public has uploaded sensor data to the historical sensor table.
Post-conditions	
Flow of events	1.View history sensor table . 2. The system returns the history sensor record.
Alternative flow and exception	
Priority	high
Non-behavioral requirement	

Use case name	View patient list
Case ID	UC-1100
actor	Doctor
Brief description	Doctor can view patient lists.
Preconditions	Patient has been registering an account.
Post-conditions	
Flow of events	1.View users table. 2. The system returns the patient list.
Alternative flow and exception	
Priority	high
Non-behavioral requirement	Doctor can contact patient.

9 Diagram

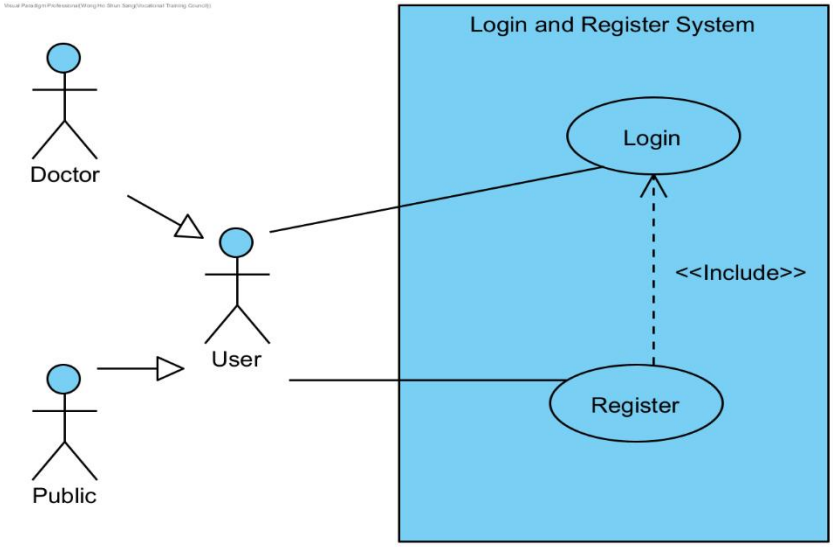
9.1 Use Case Diagram

Remote Health Monitoring System – Full System



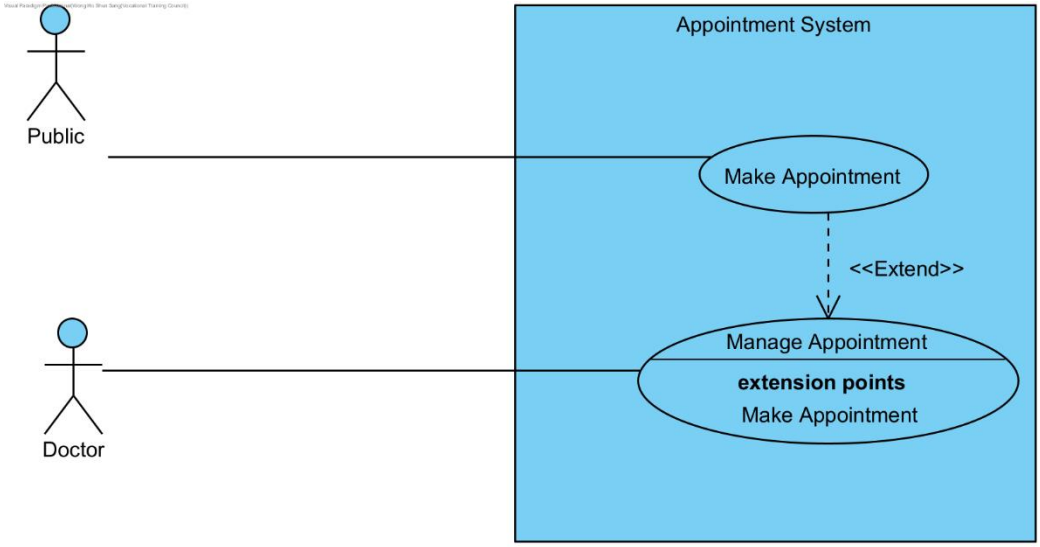
This is a use case diagram of the Remote Health Monitoring System. The diagram involves three primary actors: Patient, Doctor, and AI Bot. The Patient represents individuals utilizing the system for remote health monitoring, while the Doctor represents healthcare professionals engaging with patients through the system. The AI Bot acts as an artificial intelligence component providing automated responses and information. The diagram includes four main use cases: Register, Chat with Doctor, Query AI Bot, and Remote Health Monitoring. Users can register and access the system, engage in real-time communication with doctors for medical advice, interact with the AI Bot for information, and upload health data for remote monitoring by doctors. This use case diagram provides a concise overview of the system's main functionalities and the interactions between the actors and the system.

Login and Register System



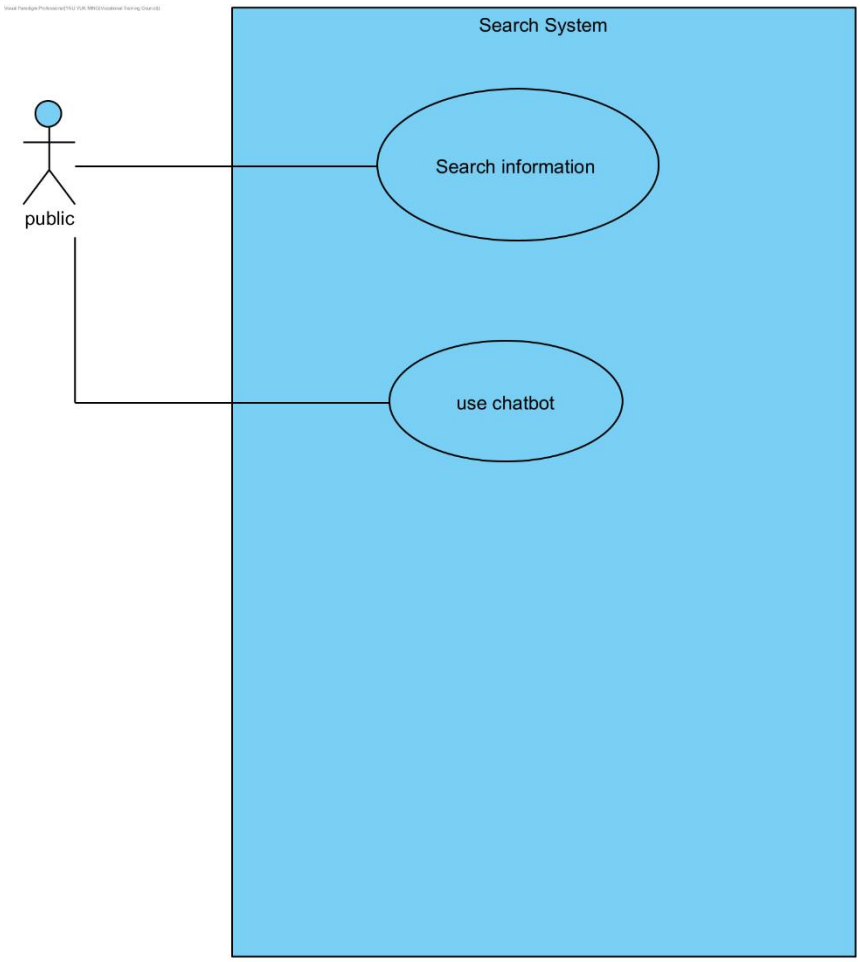
The use case diagram showcases the Login and Register functions of the Remote Health Monitoring System. The User actor interacts with the System actor to perform two main use cases. In the Register use case, the User provides personal information and submits it to the System for registration. The System validates the information, creates a new user account, and sends a confirmation notification. The Login use case allows the User to enter their login credentials, which are then verified by the System. If the credentials are valid, access is granted to the User. Otherwise, an error message is displayed. The diagram highlights the essential steps involved in user registration and login processes, emphasizing the interaction between the User and the System for accessing the Remote Health Monitoring System.

Appointment System



This use case diagram illustrates the appointment function of the Remote Health Monitoring System. The main actors in the diagram are the Patient and the Doctor, representing individuals who interact with the system. The diagram includes two primary use cases: "View Doctor Availability" and "Book Appointment." In the "View Doctor Availability" use case, the Patient can check the availability of different doctors within the system. This allows the Patient to see the doctors' schedules and identify suitable time slots for appointments. In the "Book Appointment" use case, the Patient selects a preferred date and time slot from the available options and requests an appointment with the chosen doctor. The system confirms the appointment and sends a notification to both the Patient and the Doctor. This use case diagram demonstrates the essential steps involved in the appointment booking process within the Remote Health Monitoring System, highlighting the interaction between the Patient, Doctor, and the system.

Search and chatbot System



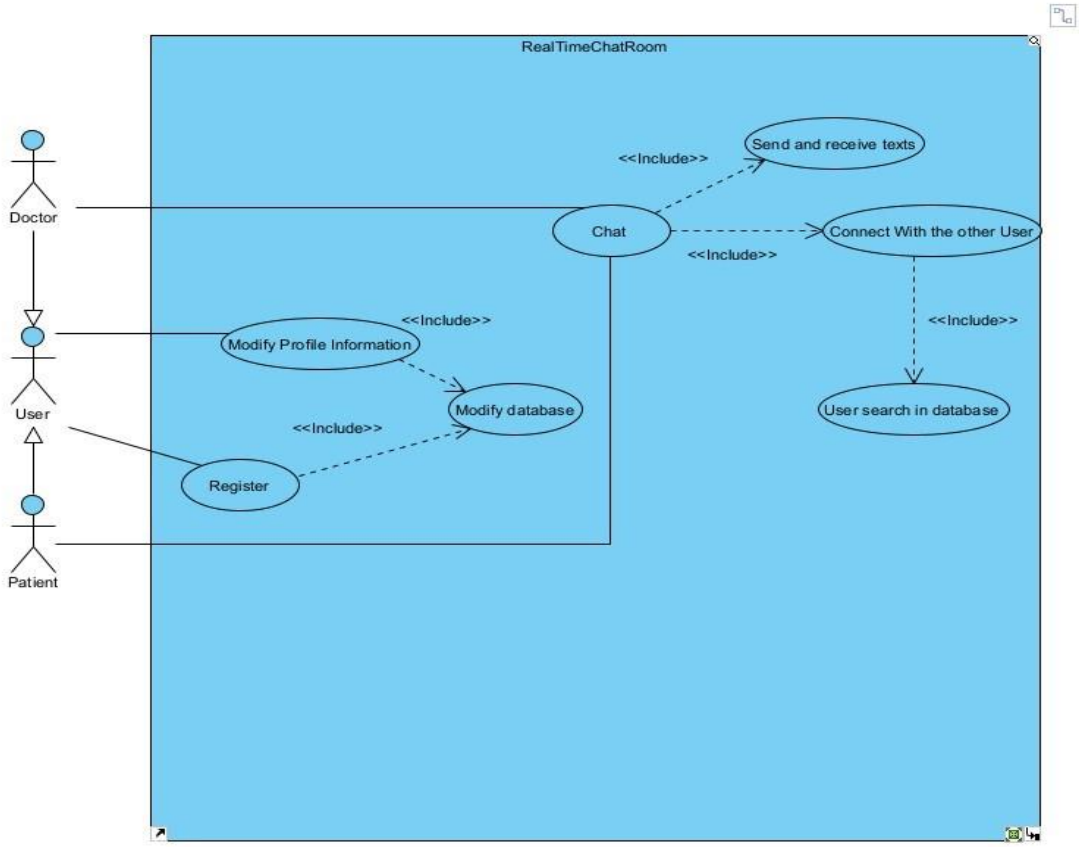
The use case diagram showcases the search and chat bot functions of the Remote Health Monitoring System. The diagram includes two primary actors: User and System. The User represents individuals interacting with the system, while the System represents the Remote Health Monitoring System itself.

The diagram incorporates two main use cases: "Search Health Information" and "Chat with Bot." In the "Search Health Information" use case, the User can enter specific keywords or queries to search for relevant health information within the system. The System retrieves and presents the search results to the User, providing them with access to a wide range of health-related knowledge.

In the "Chat with Bot" use case, the User initiates a conversation with the system's Chat Bot. The User can ask questions, seek medical advice, or request information on specific health topics. The Chat Bot utilizes predefined knowledge and natural language processing techniques to generate appropriate responses and engage in a dialogue with the User.

These two use cases demonstrate how the Remote Health Monitoring System allows Users to search for health information and interact with the Chat Bot for personalized support and guidance. The diagram illustrates the interaction between the User and the System, emphasizing the functionalities of search and chat bot within the system.

Chat System

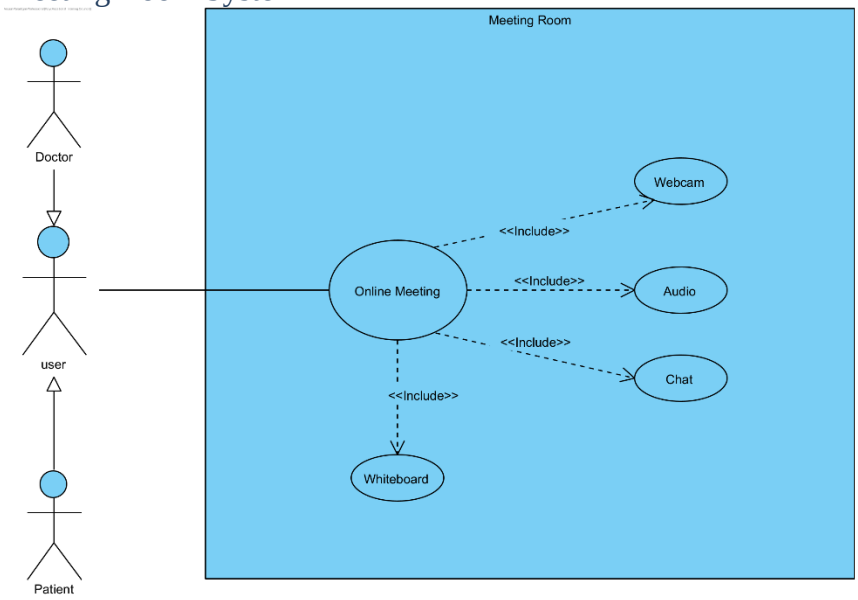


The use case diagram represents the chat system function of the Remote Health Monitoring System. The diagram includes three primary actors: User, Doctor, and System. The User represents individuals utilizing the system, the Doctor represents healthcare professionals, and the System represents the Remote Health Monitoring System itself.

The diagram consists of one main use case: "Chat with Doctor." In this use case, the User can initiate a conversation with the Doctor through the chat system provided by the Remote Health Monitoring System. The User can seek medical advice, ask questions, or discuss health concerns with the Doctor in real-time. The Doctor, on the other hand, responds to the User's messages, provides diagnoses, offers recommendations, or engages in a dialogue to address the User's medical needs.

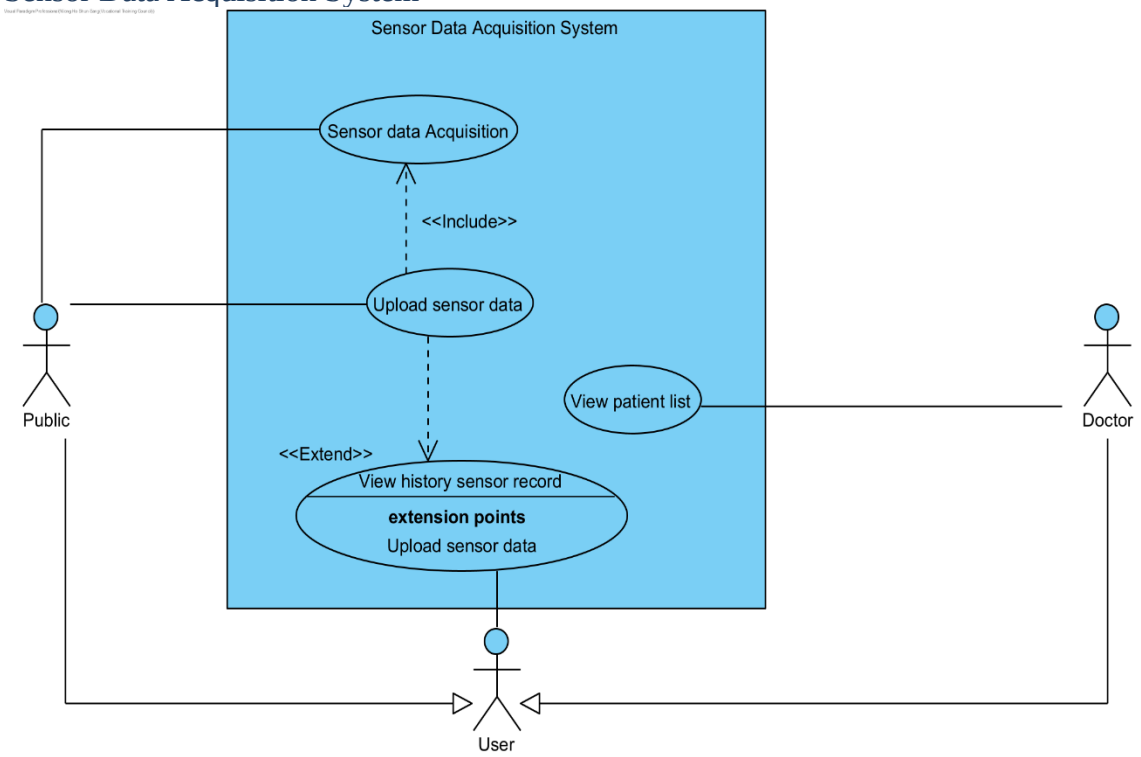
The chat system facilitates secure and convenient communication between the User and the Doctor, allowing for remote consultations and support. The system ensures that sensitive medical information remains confidential and provides a platform for effective healthcare communication. This use case diagram demonstrates the interaction between the User, Doctor, and the System, highlighting the chat system's function within the Remote Health Monitoring System.

Meeting Room System



The use case diagram represents the chat system function of the Remote Health Monitoring System. The diagram includes three primary actors: User, Doctor, and System. The User represents individuals utilizing the system, the Doctor represents healthcare professionals, and the System represents the Remote Health Monitoring System itself.
The diagram consists of one main use case: "Online Meeting". In this use case, the User can join a meeting to have a conversation through Webcam, Audio, Chat and Whiteboard. It provides variety way to let user communicate with each other.

Sensor Data Acquisition System

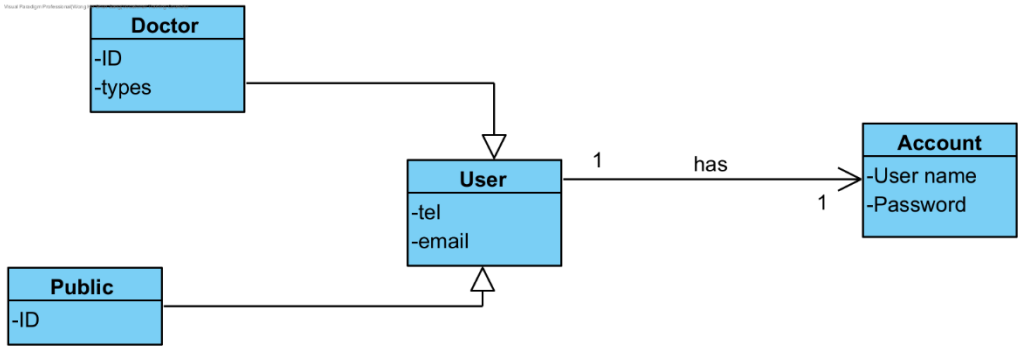


- "Sensor data Acquisition": This use case represents the system's ability to collect and acquire data from various sensors.
- "Upload sensor data": This use case describes the functionality of uploading the acquired sensor data to the system's database.
- "View history sensor record": This use case allows users to view the historical sensor data records for patients.
- The "Sensor data Acquisition" use case includes the functionality of acquiring sensor data.
- The "View history sensor record" use case extends the "Upload sensor data" use case, allowing users to view the uploaded sensor data.

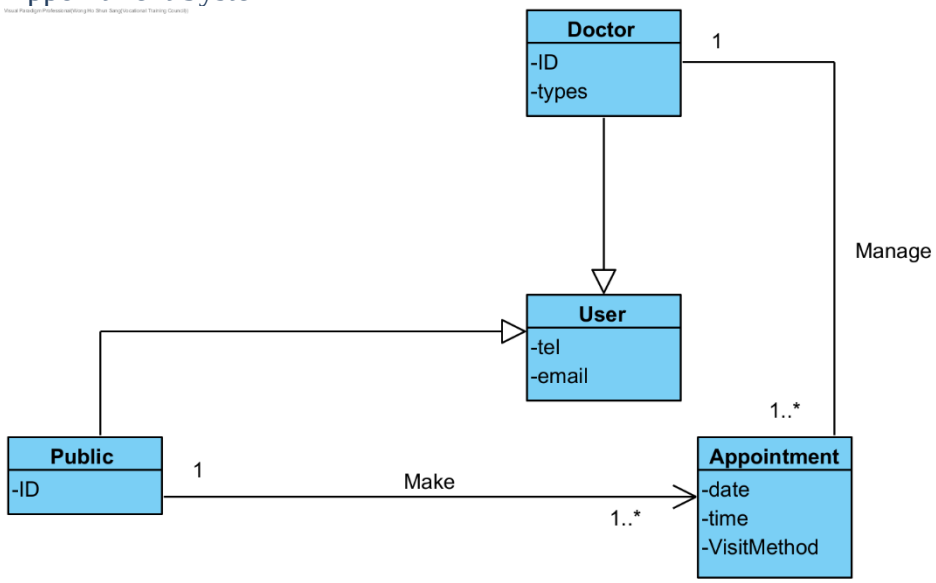
Overall, the use case diagram depicts the core functionalities of the Sensor Data Acquisition System, including data acquisition, data upload, and historical data viewing, as well as the interactions between the system and its various actors (Public, Doctor, and User).

9.2 Class diagram

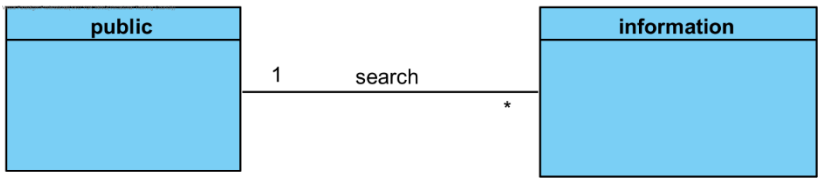
Login and Register System



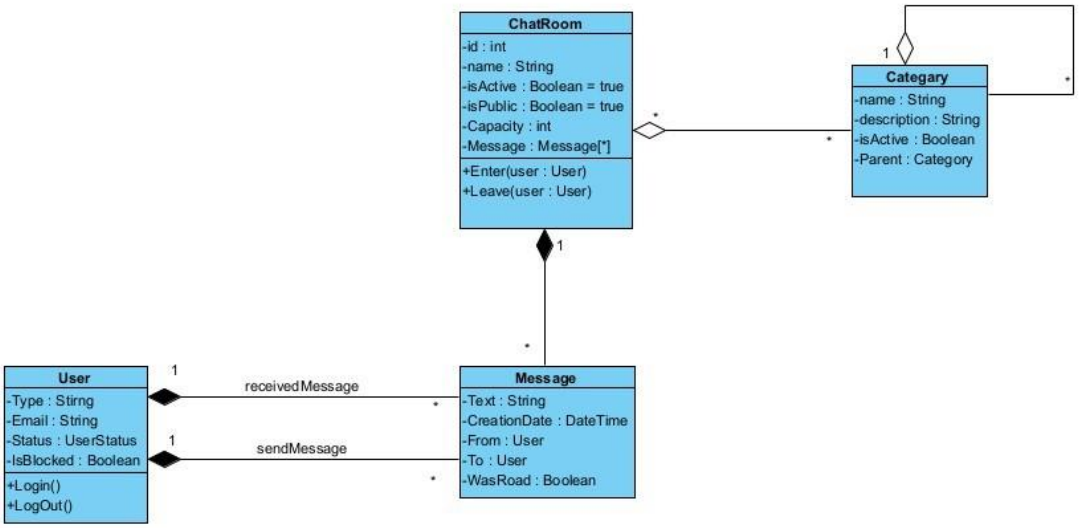
Appointment System



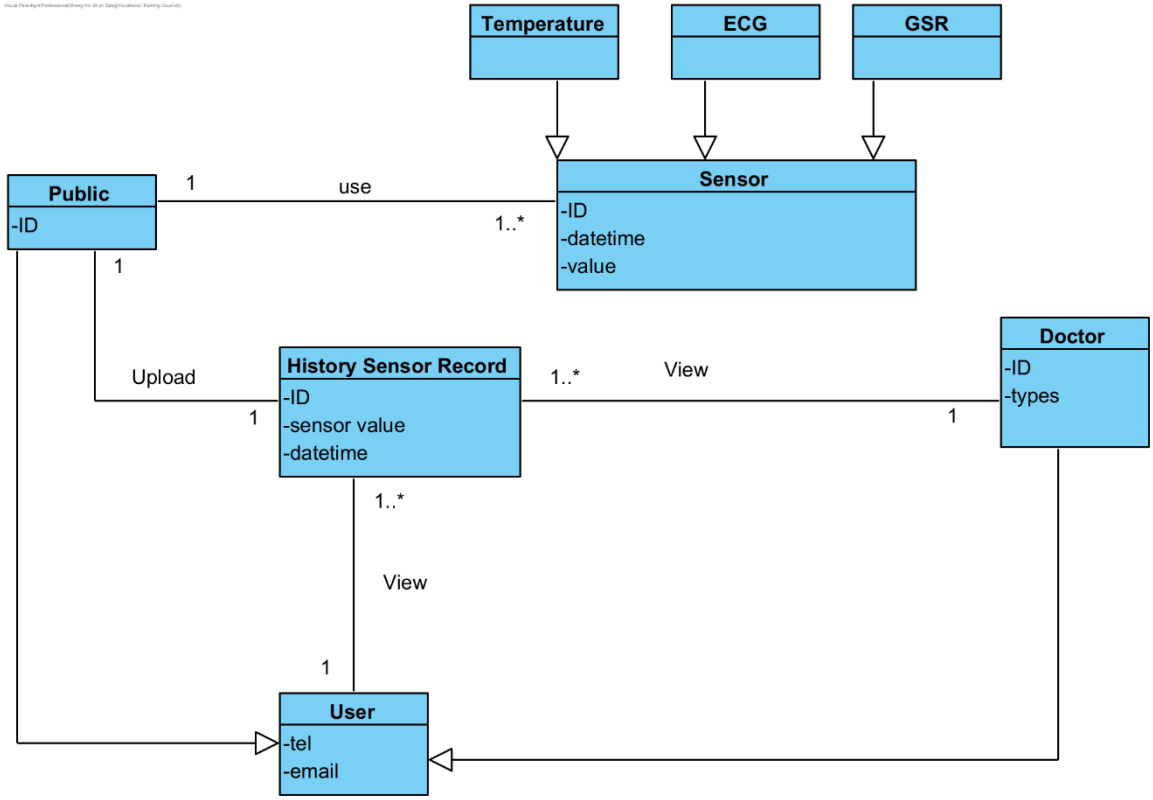
Search System



Chat System



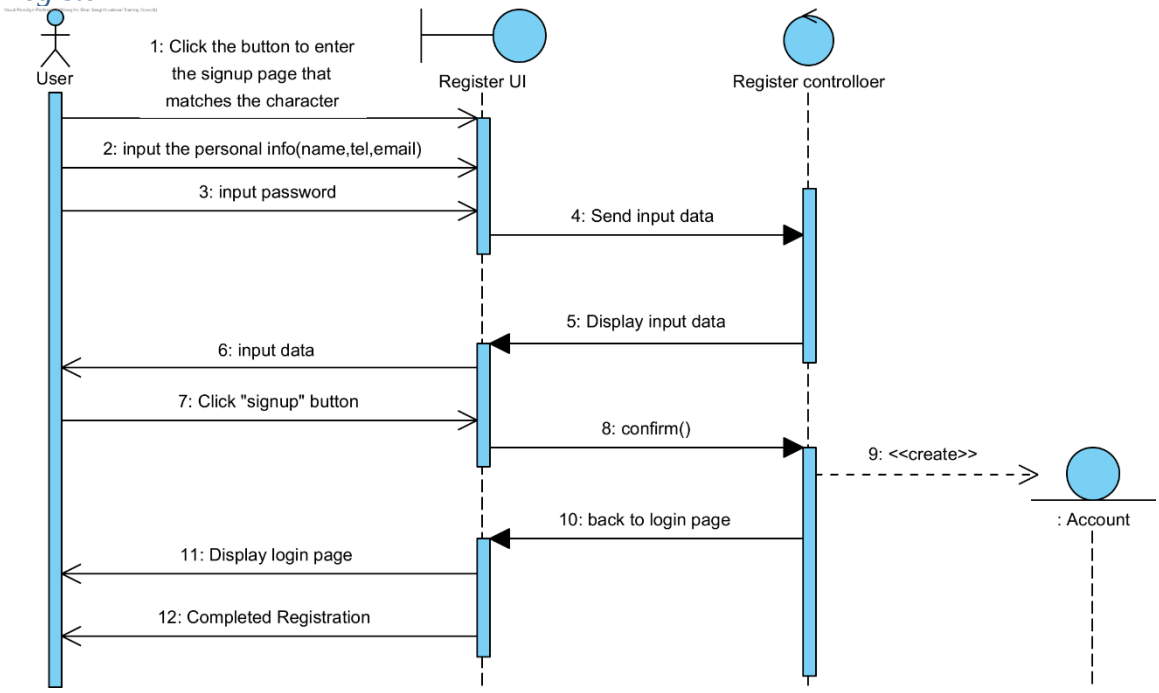
Sensor Data Acquisition System



9.3 Sequence diagram

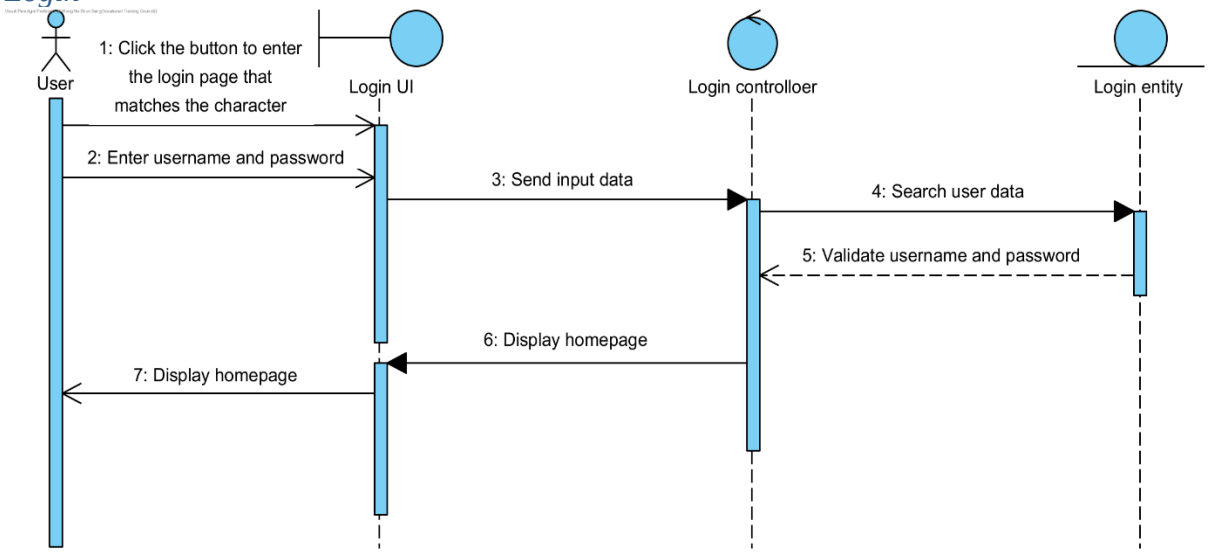
Login and Register System

Register



The sequence diagram depicts the Register system function of the Remote Health Monitoring System. The diagram shows the interactions between the User, RegisterController, and Database. It starts with the User sending a registration request to the RegisterController. The RegisterController validates the provided information and communicates with the Database to create a new user account. Once the account is successfully created, the RegisterController sends a registration confirmation back to the User. This sequence diagram illustrates the flow of actions involved in the Register process within the Remote Health Monitoring System.

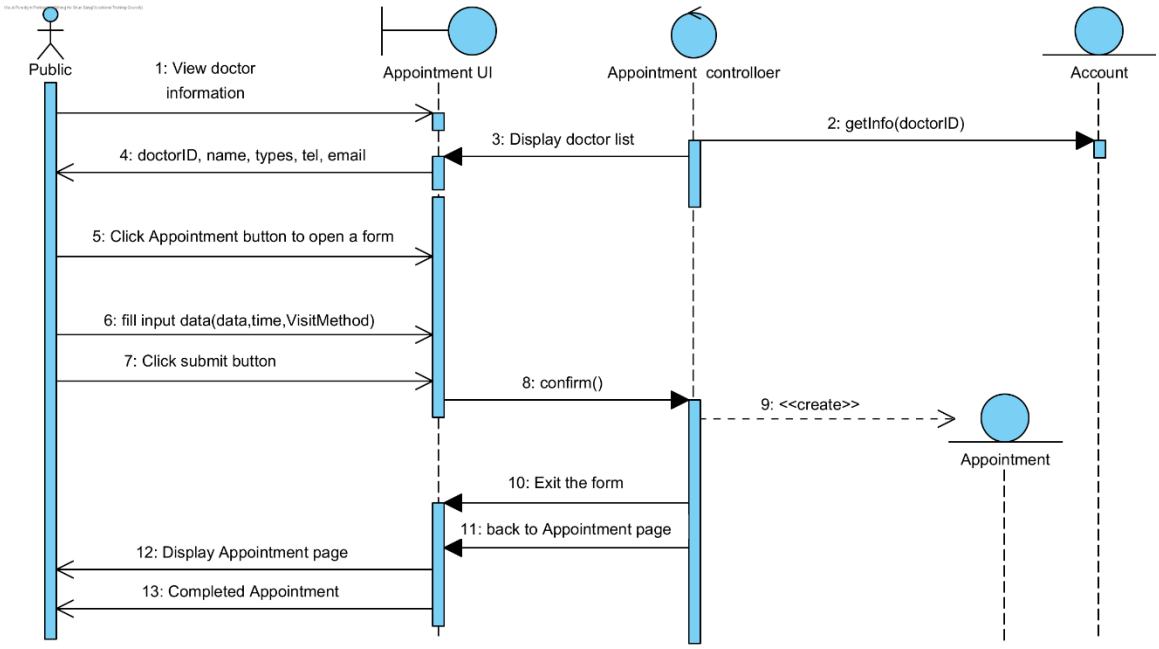
Login



The sequence diagram illustrates the Login system function of the Remote Health Monitoring System. The diagram showcases the interactions between the User, LoginController, and Database. It begins with the User sending a login request to the LoginController. The LoginController verifies the credentials provided by the User and communicates with the Database to retrieve the user's information. Upon successful verification, the LoginController sends a login response back to the User, allowing access to the system. This sequence diagram provides a visual representation of the flow of actions involved in the Login process within the Remote Health Monitoring System.

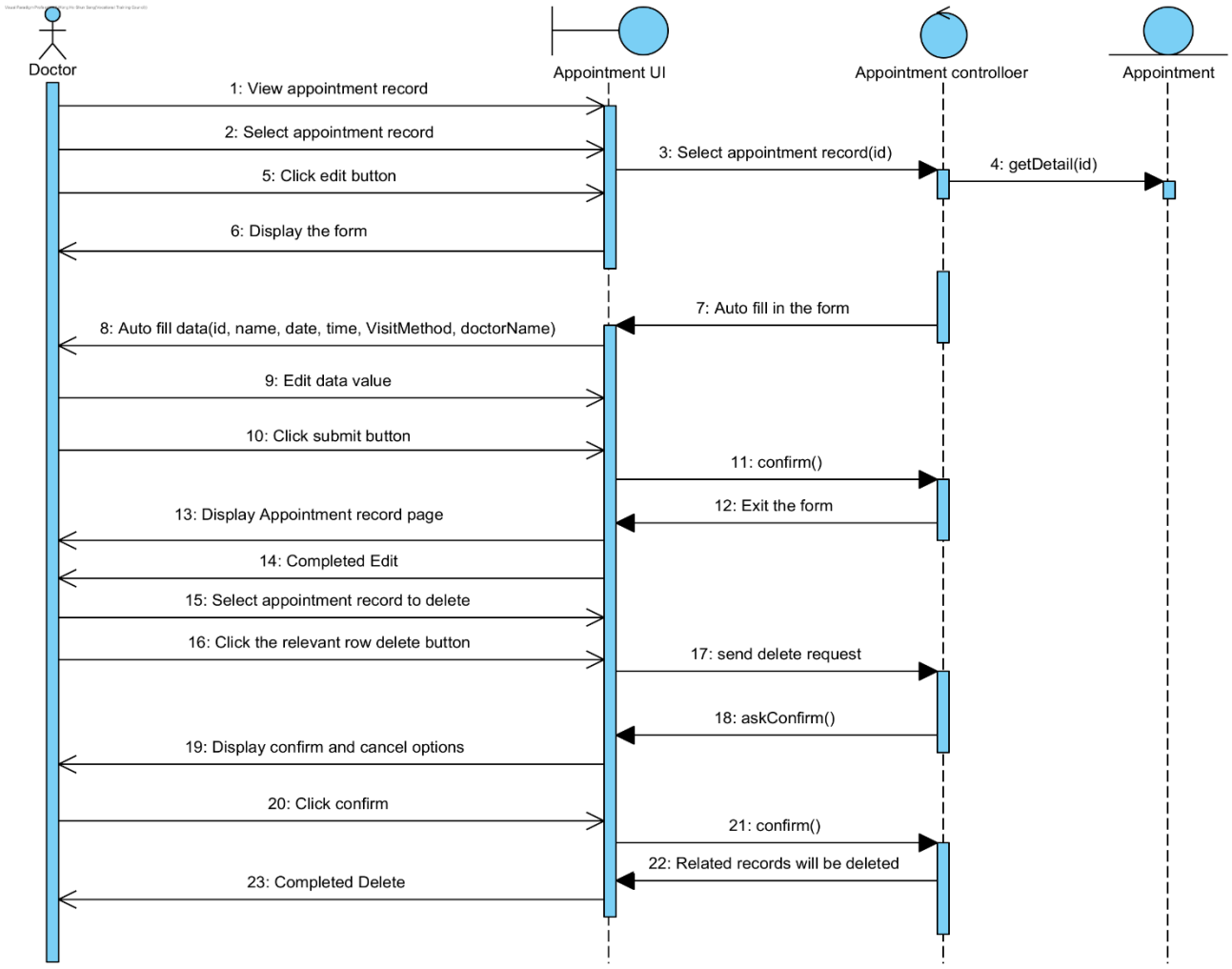
Appointment System

Make Appointment



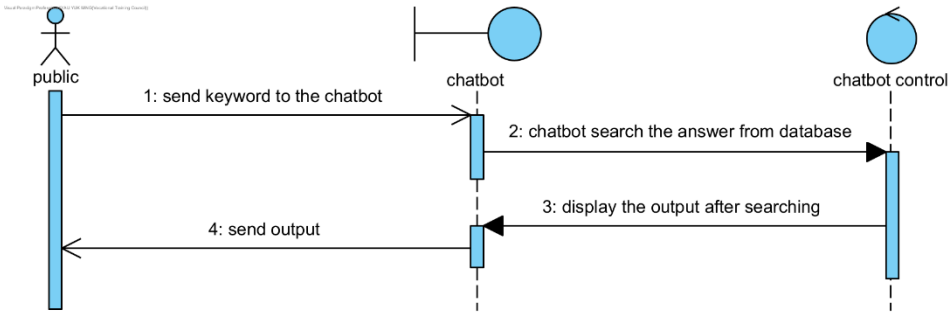
The sequence diagram illustrates the Appointment system's make appointment function in the Remote Health Monitoring System. The diagram depicts the interactions between the User, AppointmentController, Doctor, and Database. It begins with the User sending a request to make an appointment to the AppointmentController. The AppointmentController checks the Doctor's availability by communicating with the Database. If a suitable time slot is available, the AppointmentController confirms the appointment and notifies the User. Additionally, the AppointmentController notifies the Doctor about the new appointment. This sequence diagram showcases the flow of actions involved in the process of making an appointment within the Remote Health Monitoring System.

Manage Appointment



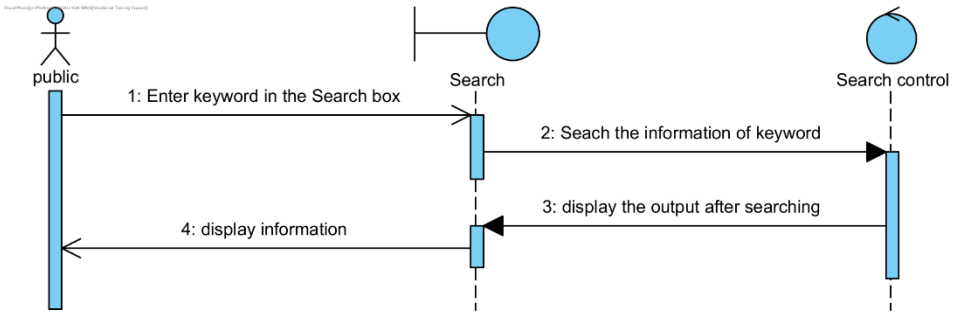
The sequence diagram illustrates the Appointment system's manage appointment function in the Remote Health Monitoring System. The diagram showcases the interactions between the User, AppointmentController, Doctor, and Database. It begins with the User requesting to manage an appointment, such as rescheduling or canceling, to the AppointmentController. The AppointmentController verifies the User's authorization and communicates with the Database to retrieve the appointment details. Once the necessary changes are made, the AppointmentController updates the appointment status and notifies both the User and the Doctor about the changes. This sequence diagram demonstrates the flow of actions involved in managing appointments within the Remote Health Monitoring System.

Chat bot system



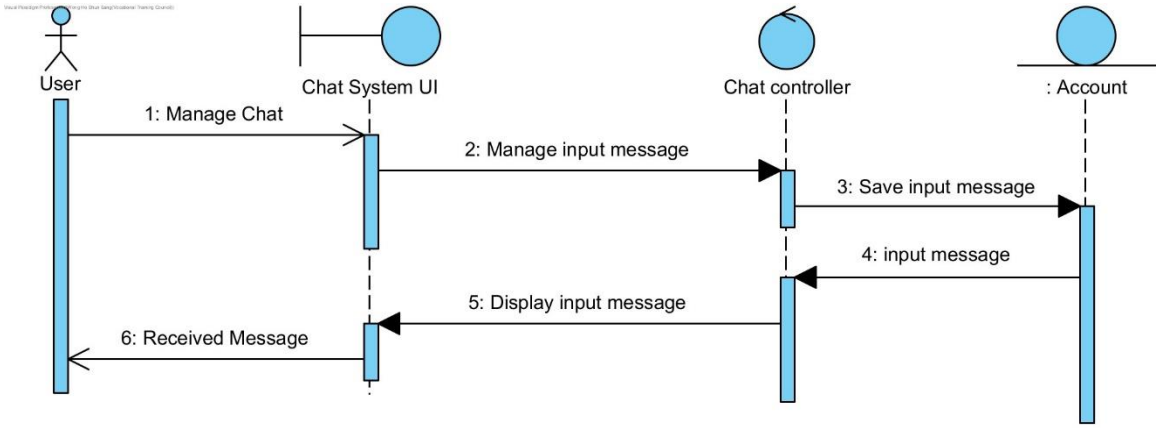
The sequence diagram represents the chat bot system function of the Remote Health Monitoring System. The diagram depicts the interactions between the User, ChatBot, and KnowledgeBase. It begins with the User sending a message to the ChatBot. The ChatBot analyzes the message and retrieves relevant information from the KnowledgeBase. Based on the analysis, the ChatBot generates a response and sends it back to the User. This sequence diagram illustrates the flow of actions involved in the chat bot system function, where the ChatBot utilizes the KnowledgeBase to provide intelligent and informative responses to the User's queries within the Remote Health Monitoring System.

Search system



The sequence diagram depicts the search system function of the Remote Health Monitoring System. The diagram shows the interactions between the User, SearchController, and Database. It begins with the User sending a search query to the SearchController. The SearchController processes the query and communicates with the Database to retrieve relevant health information. The SearchController then sends the search results back to the User. This sequence diagram illustrates the flow of actions involved in the search system function, where the User can search for and retrieve health information within the Remote Health Monitoring System.

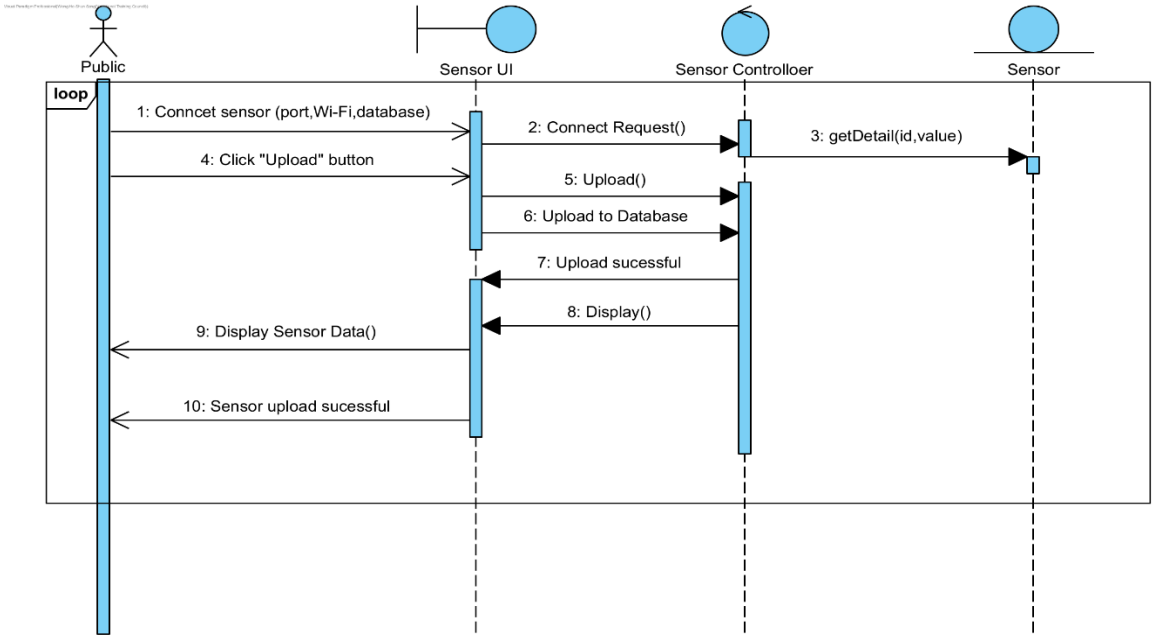
Chat System



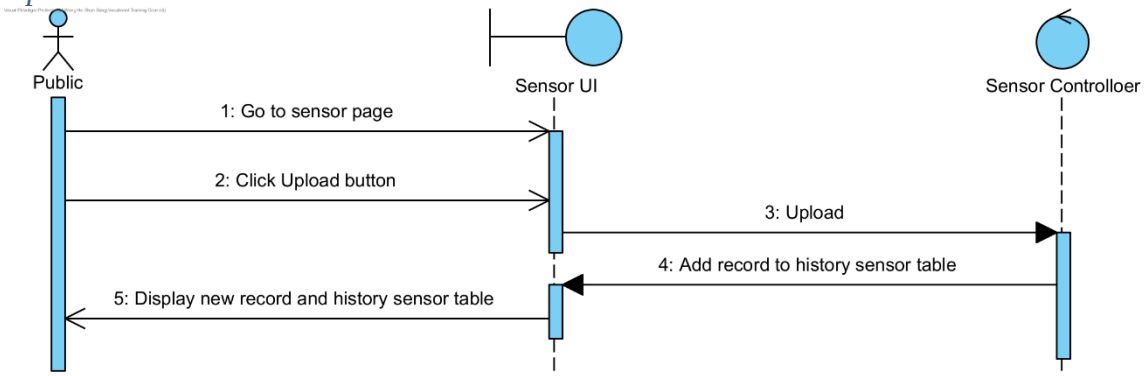
The sequence diagram depicts the chat system function of the Remote Health Monitoring System using a chat room for real-time communication. The diagram shows the interactions between Users and the Chat Room. Users join the chat room and send messages, which are broadcasted to all participants. The Chat Room receives the messages and distributes them to all connected Users. This sequence diagram illustrates the flow of real-time communication within the chat system, where Users can engage in conversations with each other through the shared chat room in the Remote Health Monitoring System.

Sensor Data Acquisition System

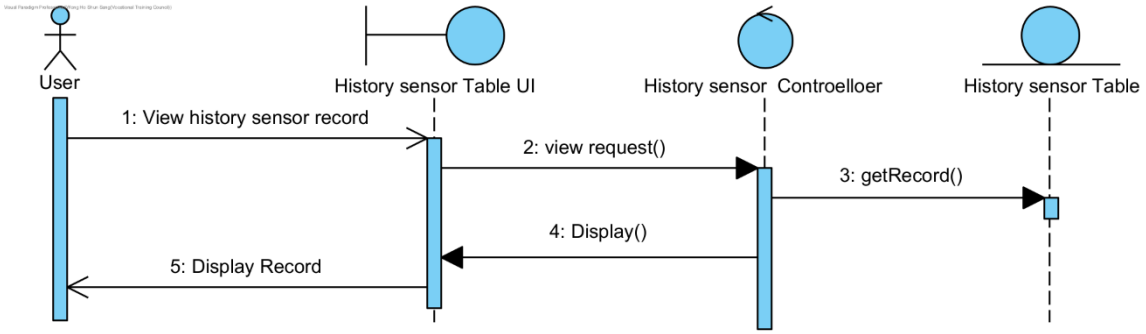
Sensor data Acquisition



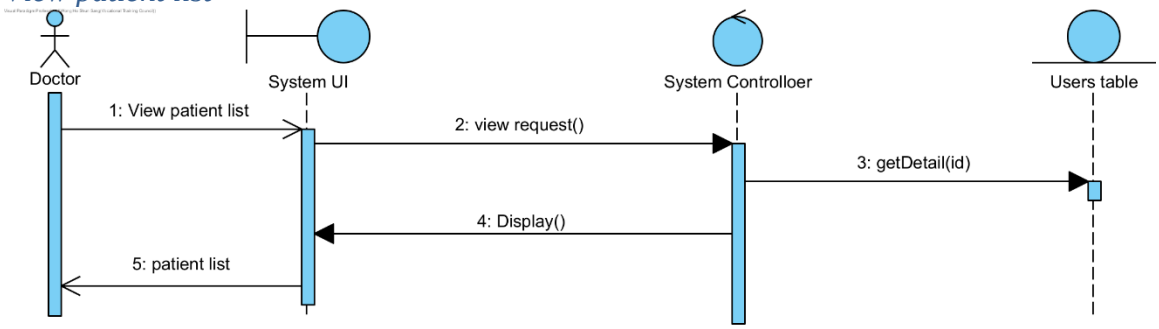
Upload sensor data



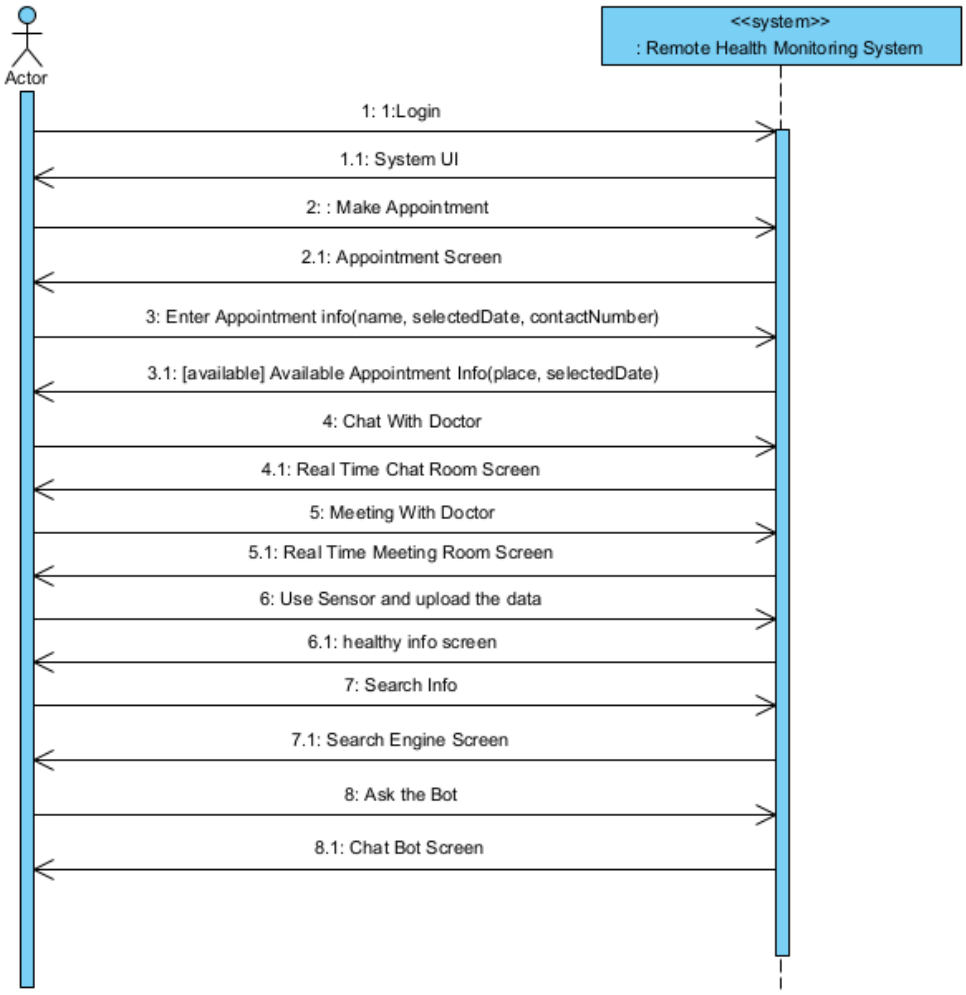
View history sensor record



View patient list



System level Sequence Diagram

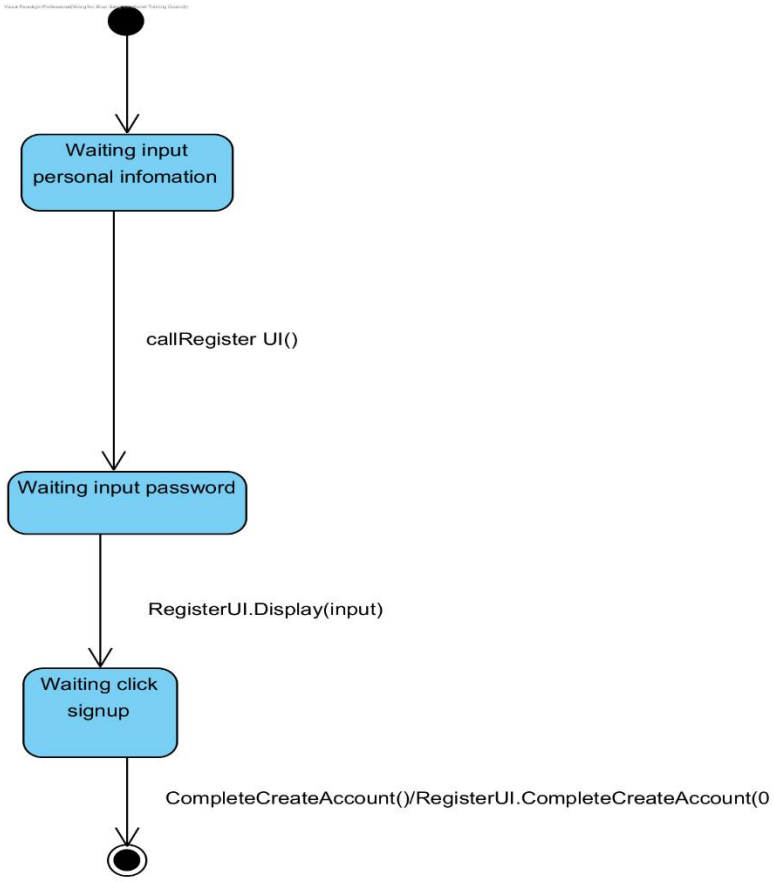


The System level sequence diagram of the Remote Health Monitoring System depicts the flow of interactions between the User, Login, Chat, ChatBot, Appointment, and Database. It shows how users' login to the system, engage in real-time chats facilitated by the Chat receive intelligent responses from the ChatBot based on the KnowledgeBase, and manage appointments through the Appointment, which communicates with the Database to update and retrieve appointment information. This diagram provides an overview of the key functionalities of the Remote Health Monitoring System and how different components interact to deliver a seamless user experience.

9.4 State Machine diagram

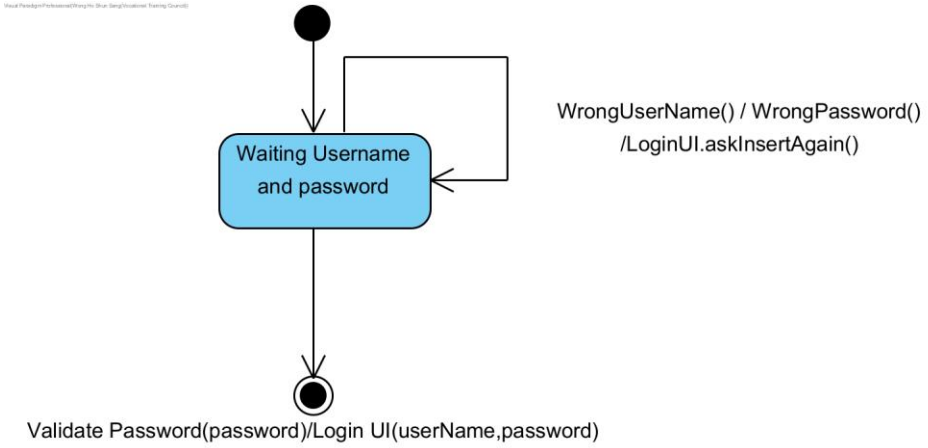
Login and Register System

Register



The state machine diagram illustrates the register system function of the Remote Health Monitoring System, demonstrating the different states and transitions involved in the registration process.

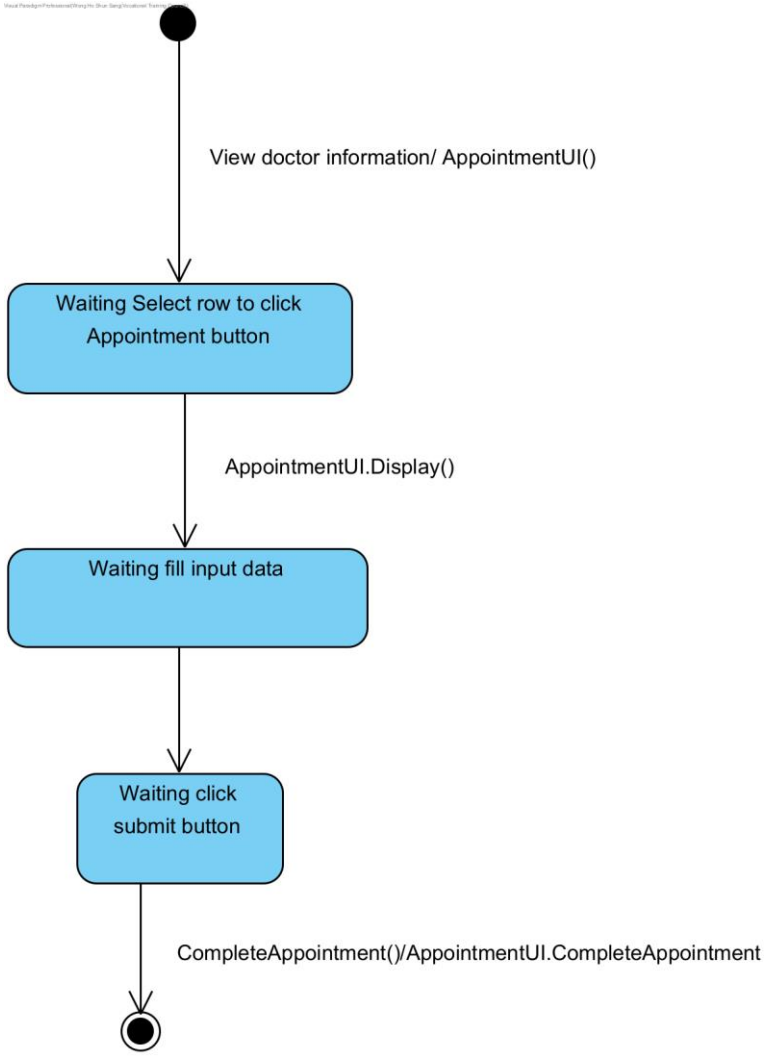
Login



The state machine diagram represents the login system function of the Remote Health Monitoring System, showcasing the various states and transitions involved in the login process.

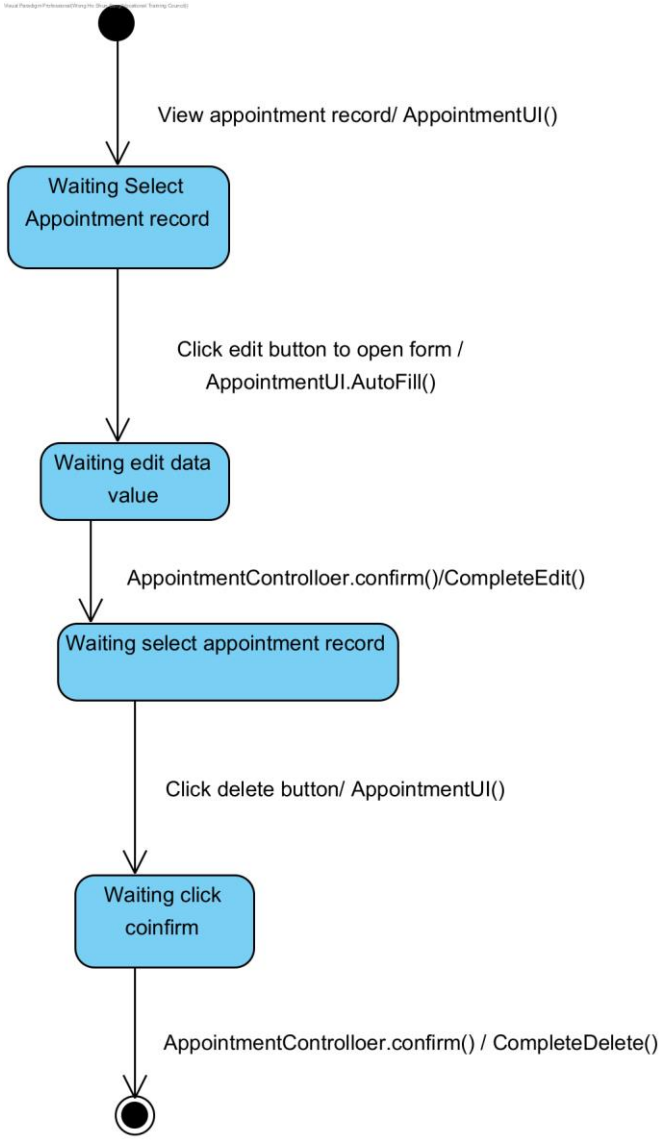
Appointment System

Make Appointment



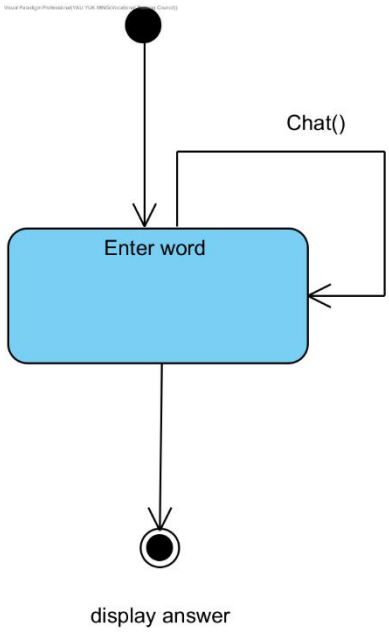
The state machine diagram represents the make appointment function of the Appointment system in the Remote Health Monitoring System, showcasing the different states and transitions involved in the process of creating an appointment.

Manage Appointment



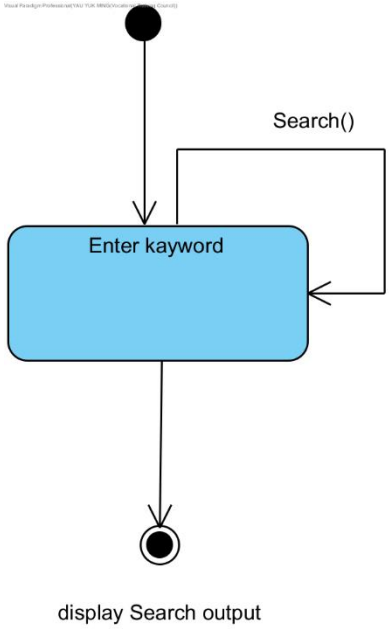
The state machine diagram depicts the manage appointment function of the Appointment system in the Remote Health Monitoring System, illustrating the various states and transitions involved in the process of modifying or canceling an existing appointment.

Chatbot



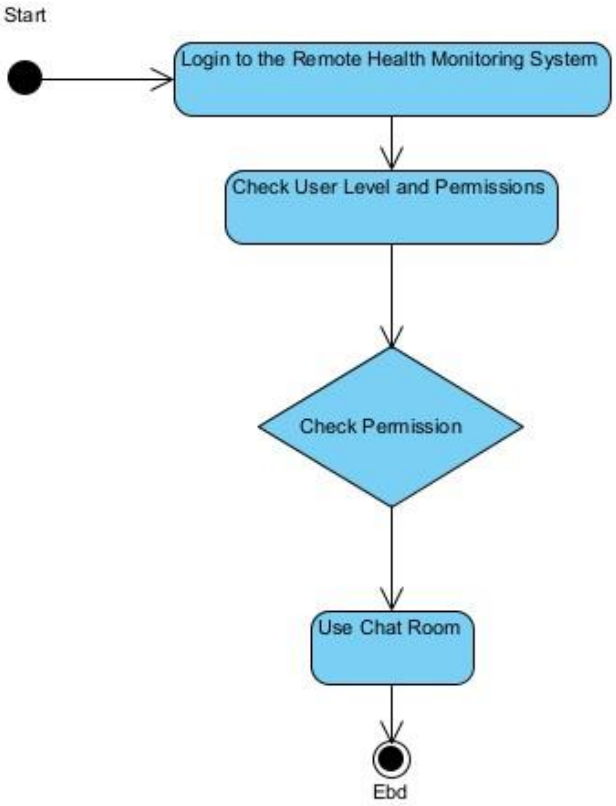
The state machine diagram represents the chat bot system function of the Remote Health Monitoring System, showcasing the different states and transitions involved in the interaction between the user and the chat bot for providing responses and gathering information.

Search



The state machine diagram illustrates the search system function of the Remote Health Monitoring System, highlighting the various states and transitions involved in the process of searching and retrieving health information based on user queries.

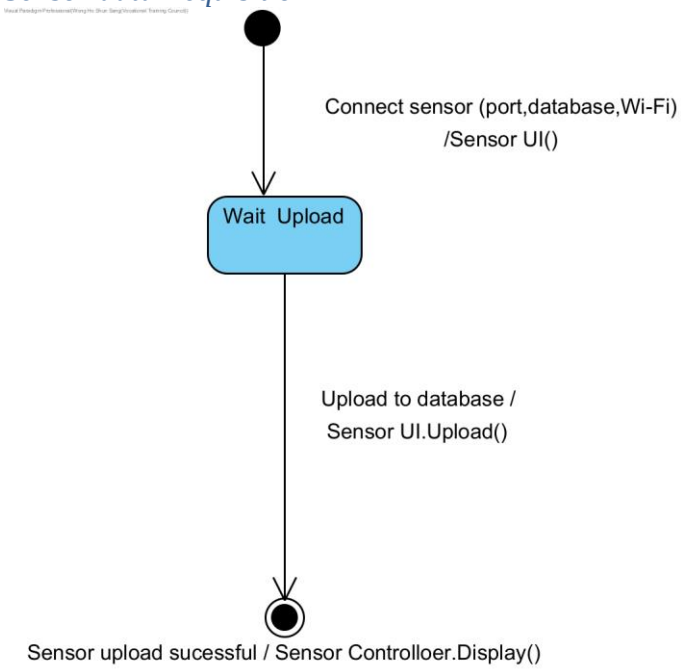
Chat System



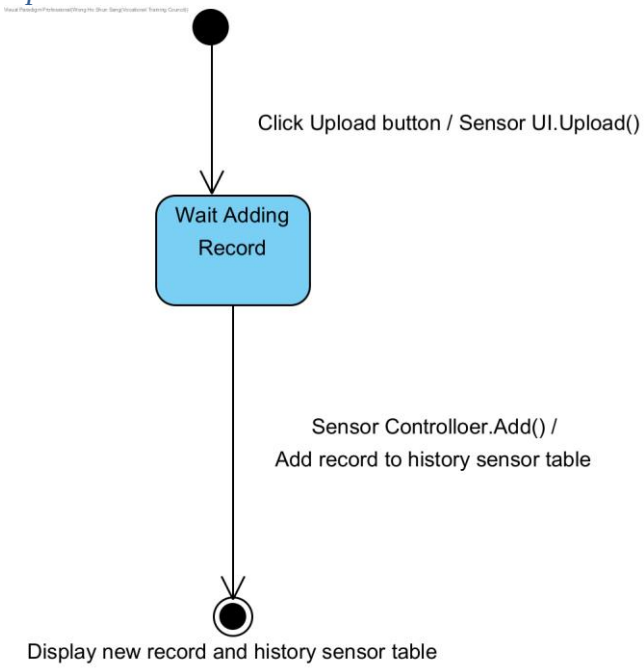
The state machine diagram represents the chat room system function of the Remote Health Monitoring System, showcasing the different states and transitions involved in the real-time communication and message exchange within the shared chat room among multiple users.

Sensor Data Acquisition System

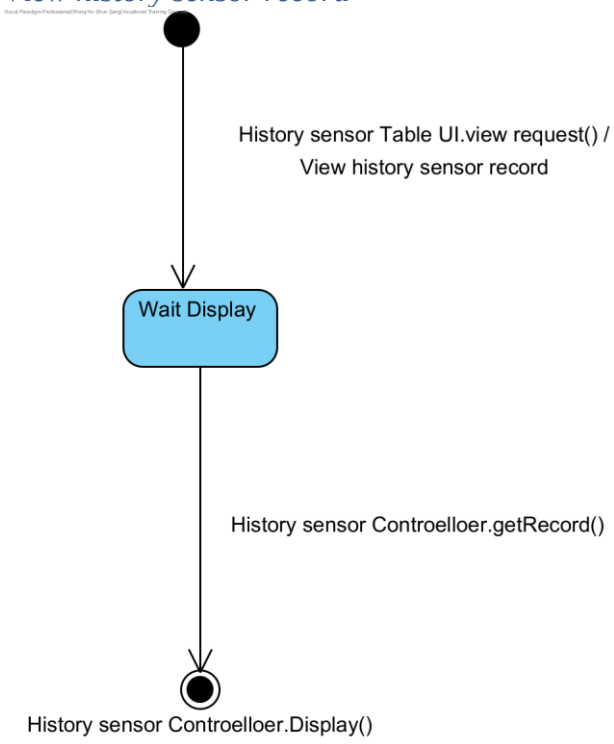
Sensor data Acquisition



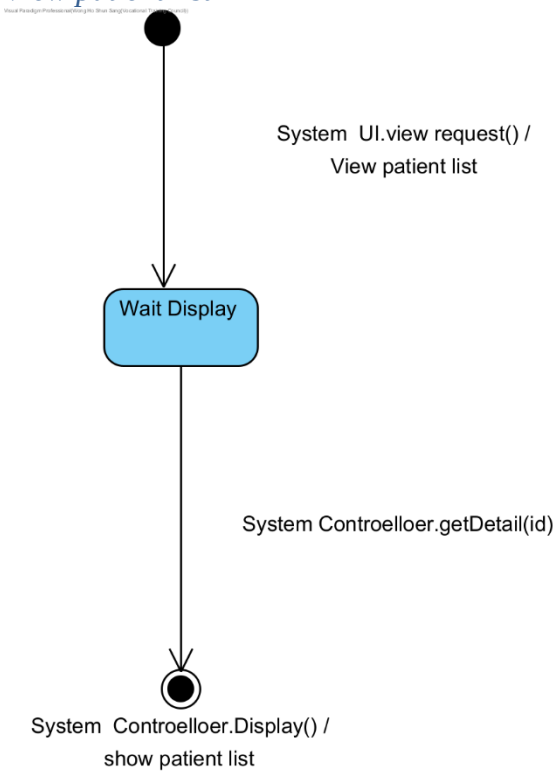
Upload sensor data



View history sensor record

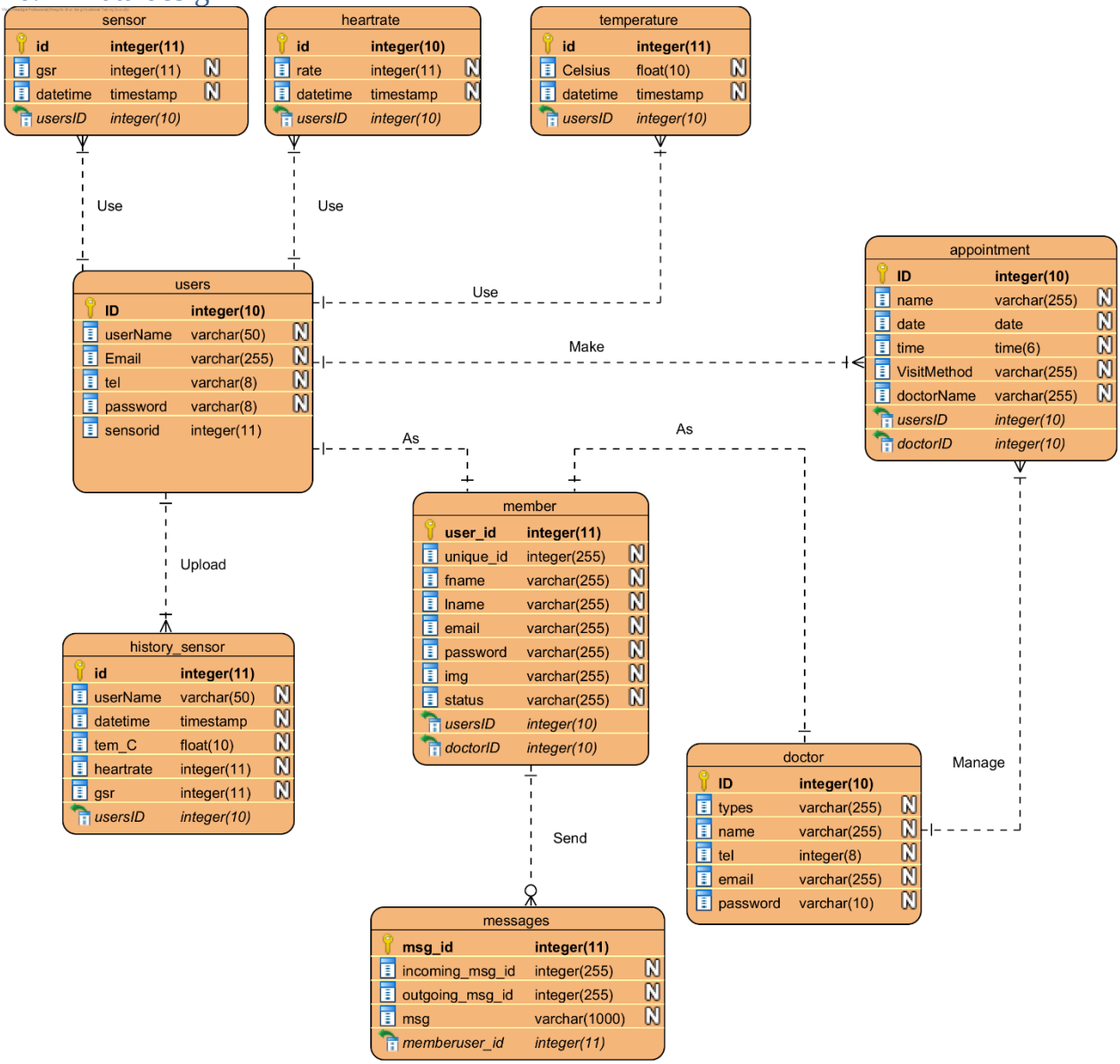


View patient list



10. Documentation for detailed design

10.1 Data design - ERD



10.2 Software architectural design

Google Chrome
is a web browser developed by Google. It was first released in 2008 for Microsoft Windows, built with free software components from Apple WebKit and Mozilla Firefox. Versions were later released for Linux, macOS, iOS, and also for Android, where it is the default browser.The browser is also the main component of ChromeOS, where it serves as the platform for web applications.

PhpStorm is a development tool for PHP and Web projects. It's a perfect PHP IDE for working with Laravel, Symfony, Drupal, WordPress, and other frameworks.

DroidCam turns your phone into a camera source on your computer.

Oracle database system

MySQL, developed by Oracle, is a renowned open-source relational database management system (RDBMS) extensively utilized for constructing reliable and scalable applications. It encompasses a broad array of functionalities that aid developers in efficiently creating, storing, managing, and retrieving data.

Xampp

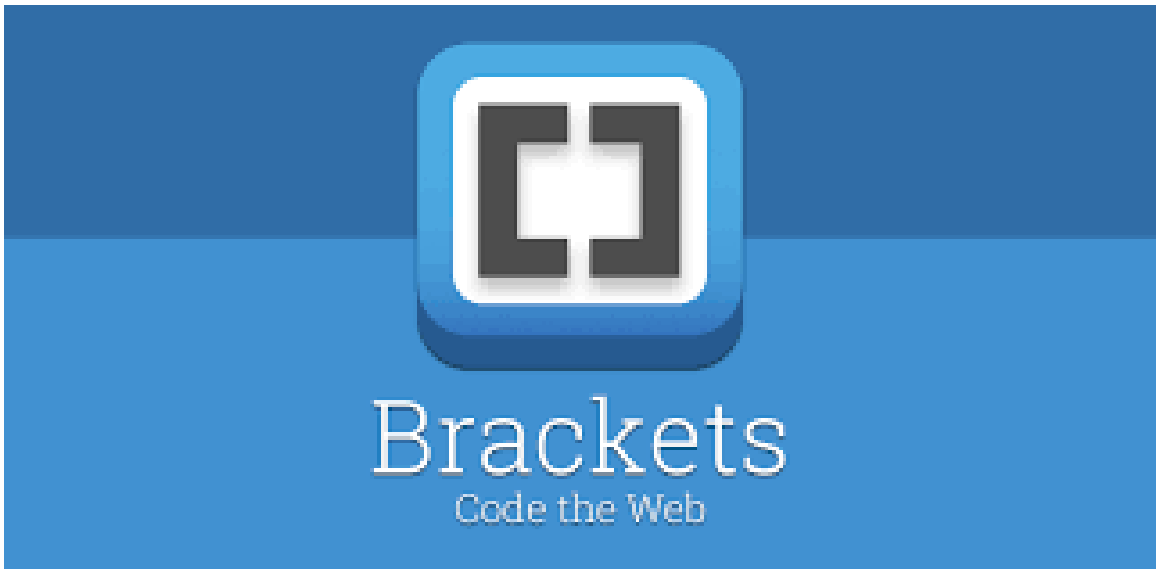
XAMPP is a widely used open-source software package that provides a local development environment for web applications. It includes Apache as the web server, MySQL as the database management system, PHP as the scripting language, and Perl for additional flexibility. XAMPP simplifies the setup and configuration of these components, allowing developers to quickly create and test web applications on their local machines before deployment.

Microsoft Office

Microsoft Office is a comprehensive collection of productivity applications that offers a diverse set of tools for creating, organizing, and collaborating on various types of documents, spreadsheets, presentations, and other tasks related to enhancing productivity.

Visual Studio Code

Visual Studio Code, commonly known as VS Code, is a lightweight and highly customizable source code editor developed by Microsoft. It provides a user-friendly interface and a vast ecosystem of extensions, enabling developers to write, edit, and debug code efficiently across multiple programming languages. With its extensive features and intuitive design, Visual Studio Code enhances the coding experience and boosts productivity for developers.



Bracket

Brackets is a lightweight and open-source code editor designed specifically for web development. It offers a clean and minimalistic interface with a focus on front-end development, providing features like inline editing, live preview, and quick editing of CSS and HTML. Brackets aims to streamline the web development workflow and make coding for the web more accessible and efficient.

Arduino IDE

Arduino is an open-source integrated development environment (IDE) that makes it easy to write code and upload it to the board. Using the Arduino software, you can work with any Arduino board. The software provides a library manager that makes it convenient to install library files, with thousands of official and contributed libraries available for direct download. Sample code is also provided. The installation process is quick and straightforward, whether you're using Windows, macOS, or Linux operating systems.

Microsoft Windows 10

Microsoft Windows 10 is an operating system developed by Microsoft and is the latest version of the Windows family. It provides a user-friendly interface, enhanced security features, and a wide range of applications and tools for various tasks. Windows 10 offers a seamless user experience with features like the Start Menu, Cortana virtual assistant, Microsoft Edge web browser, and compatibility with a vast range of software and hardware devices. It is designed to be versatile, efficient, and suitable for both personal and professional use. Here are some of the key functions of Microsoft Windows:

- User Interface
- File Management
- Security
- Device Drivers
- Networking
- System Maintenance

PyCharm

PyCharm is a powerful and user-friendly integrated development environment (IDE) specifically designed for Python programming. It offers a range of features such as code completion, debugging tools, intelligent code analysis, and project management capabilities. PyCharm provides a seamless coding experience, helping developers write, test, and debug Python code efficiently. With its intuitive interface and extensive set of tools, PyCharm is widely used by developers to enhance their productivity in Python development projects.

Software / Maintenances cost

Title	Cycle	Price	Subtotal	Software / Maintenances	Total
Hardware check	Per 3 months	388	388	M	1522 per year
System update(s)	Per week	180	180	M	9360 per year
License of Anti-virus software	Per year	1280	1280	S	1280 per year
Oracle database system	Per version	7600	7600	S	7600
License of Microsoft Office	Per version	1488	1488	S	1488
License of Microsoft Windows	Per version	2980	2980	S	2980

10.3 Hardware architectural design (system design)

Workstation PC

A workstation PC is a high-performance computer designed for professional use in demanding tasks such as graphic design, video editing, 3D modeling, scientific simulations, and software development. It typically features powerful hardware components like multi-core processors, ample RAM, dedicated graphics cards, and fast storage options. Workstation PCs are optimized for reliability, stability, and efficient workflow, often including specialized software and tools. They provide the processing power and resources required to handle complex workloads and deliver superior performance for professionals in various industries.

Router

A router is a networking device that connects multiple devices within a network and directs data packets between them. It acts as a central hub, allowing devices to communicate with each other and with devices on other networks. Routers use IP addresses to determine the best path for data transmission and ensure that information reaches its intended destination. They provide functions like network address translation (NAT), firewall protection, and quality of service (QoS) to manage network traffic effectively. Routers are essential in homes, offices, and other networked environments to enable internet connectivity and facilitate communication between devices.

Sensor

GSR Sensor

The GSR (Galvanic Skin Response) sensor measures the electrical conductivity of the skin, which can be influenced by factors such as sweat and moisture. It is commonly used to assess the level of emotional or physiological arousal in individuals.

1. High Value: A high value in the GSR sensor reading typically indicates a lower skin conductivity or higher skin resistance. This can occur when the skin is dry or when the individual is in a state of low arousal or relaxation.

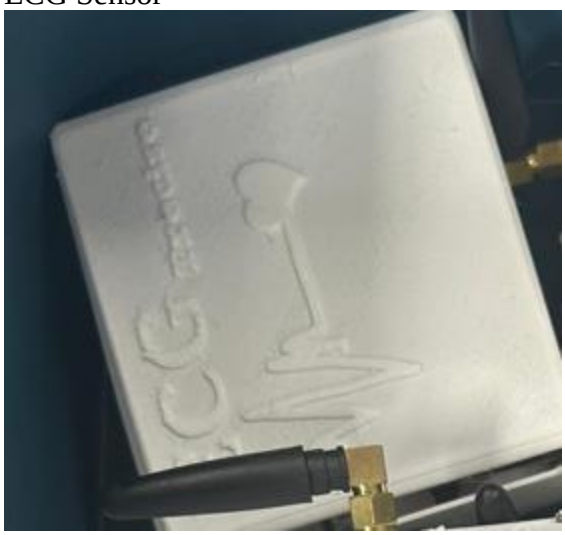
2. Low Value: Conversely, a low value in the GSR sensor reading generally indicates higher skin conductivity or lower skin resistance. This can happen when the skin is moist or when the individual is experiencing increased arousal or emotional stimulation.

Temperature Sensor

A temperature sensor, also known as an infrared thermometer or non-contact thermometer, is a device used to measure the temperature of an object or person without making physical contact. It utilizes infrared technology to detect and measure the thermal radiation emitted by the target.

The temperature readings provided by a temperature gun sensor are typically displayed in Celsius (°C).

ECG Sensor



An ECG (Electrocardiogram) sensor is a device used to measure and record the electrical activity of the heart. It detects the electrical signals generated by the heart muscle during each heartbeat and provides valuable information about the heart's rhythm, rate, and overall cardiac health.

To calculate the heart rate from an ECG signal, the number of heartbeats occurring within a specific time interval needs to be determined. The most common time interval used is 60 seconds (1 minute). The ECG sensor records the electrical activity of the heart continuously, and by analyzing the recorded data, the number of heartbeats occurring within the 60-second interval can be counted.

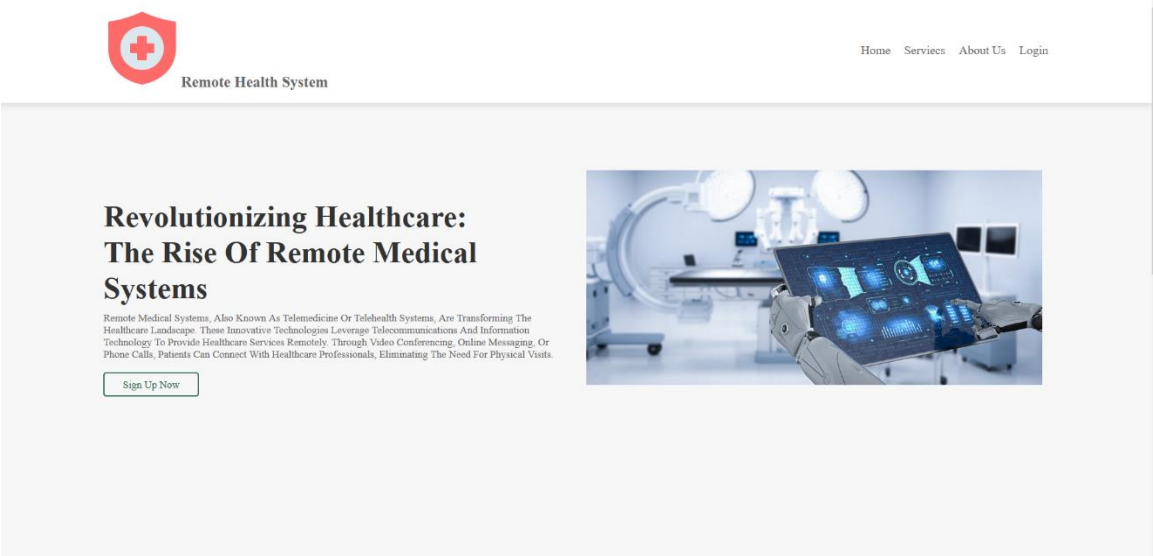
To obtain the heart rate, the number of heartbeats is divided by the duration of the interval (60 seconds) and multiplied by 60 to convert it to beats per minute (bpm). This calculation provides an estimation of the heart rate based on the ECG signal recorded by the sensor.

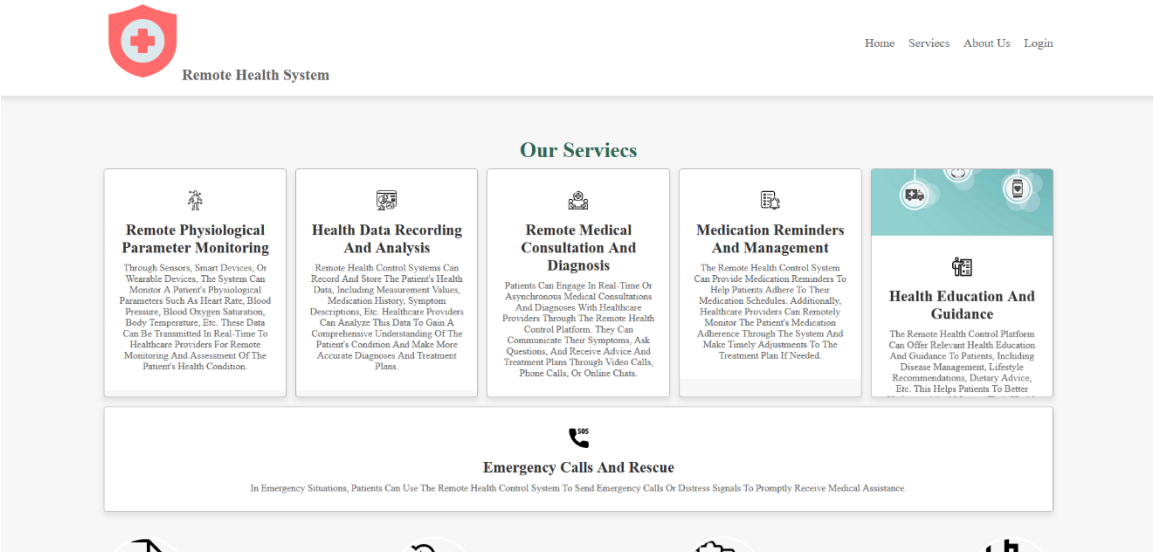
Cost of Hardware(s) and Service(s)

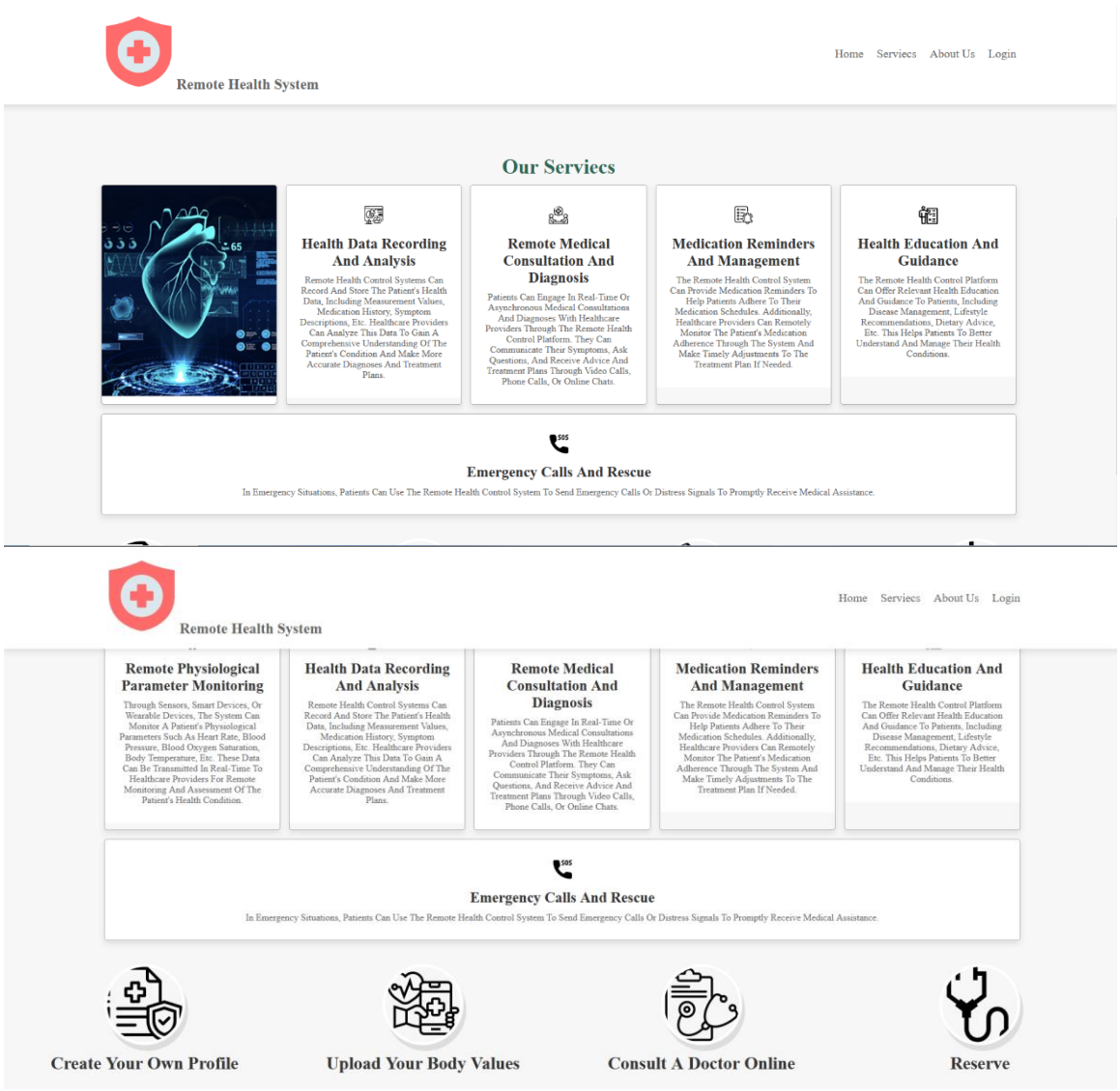
Title	Quantity	Unit Price	Subtotal	
Router (TP-Link Archer AX11000)	2	2380 HKD	4760 HKD	Router connected to the internet for system able to communicate with the server
Network fees (HKBN 10G)	1	528 HKD	528 HKD	Fees for the internet access, charged from broadband company
24/7 technical support	1	328 HKD	328 HKD	One-time payment of 24/7 technical support through online chat, phone call
Workstation PC (ThinkStation P360 Ultra - Intel)	2	9896.4 HKD	19792.8 HKD	Workstation PC for running the software and daily uses

10.4 User interface design

Home Page

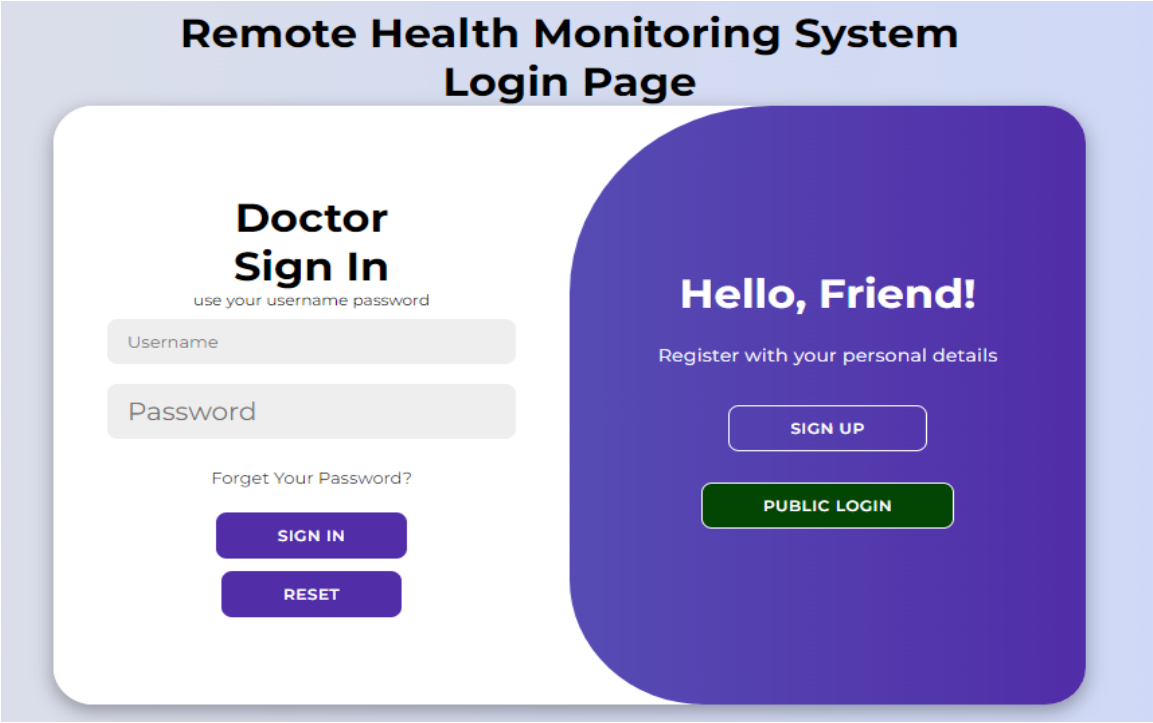




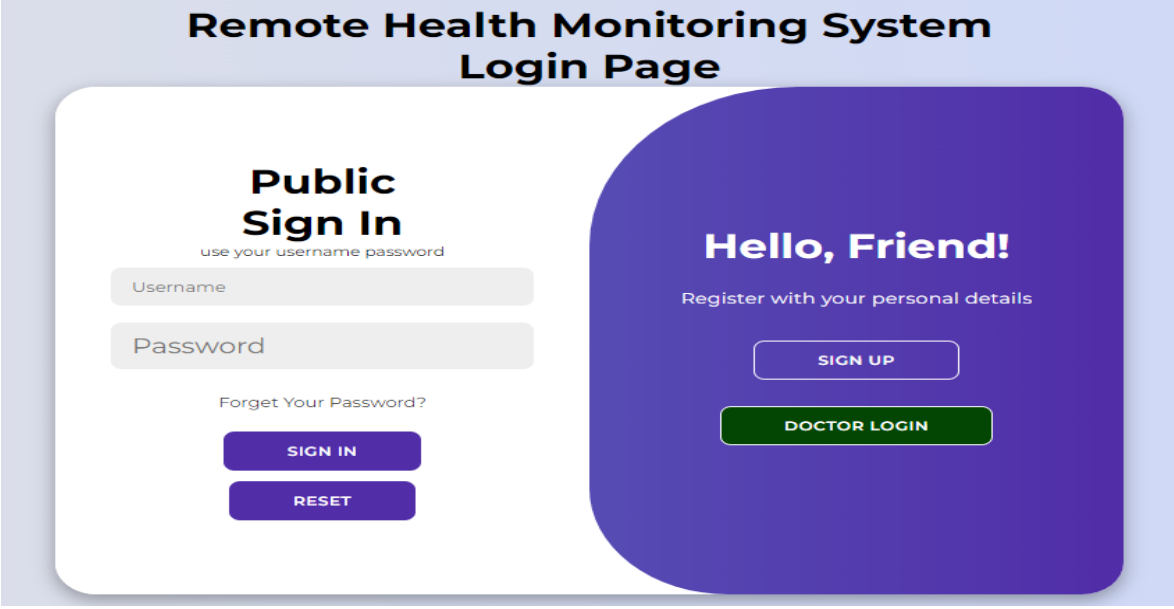

Remote Health System
Home
Serviecs
About Us
Login
Our Serviecs
Health Data Recording And Analysis
Remote Health Control Systems Can Record And Store The Patient's Health Data, Including Measurement Values, Medication History, Symptom Descriptions, Etc. Healthcare Providers Can Analyze This Data To Gain A Comprehensive Understanding Of The Patient's Condition And Make More Accurate Diagnoses And Treatment Plans.
Remote Medical Consultation And Diagnosis
Patients Can Engage In Real-Time Or Asynchronous Medical Consultations And Diagnoses With Healthcare Providers Through The Remote Health Control Platform. They Can Communicate Their Symptoms, Ask Questions, And Receive Advice And Treatment Plans Through Video Calls, Phone Calls, Or Online Chats.
Medication Reminders And Management
The Remote Health Control System Can Provide Medication Reminders To Help Patients Adhere To Their Medication Schedules. Additionally, Healthcare Providers Can Remotely Monitor The Patient's Medication Adherence Through The System And Make Timely Adjustments To The Treatment Plan If Needed.
Health Education And Guidance
The Remote Health Control Platform Can Offer Relevant Health Education And Guidance To Patients, Including Disease Management, Lifestyle Recommendations, Dietary Advice, Etc. This Helps Patients To Better Understand And Manage Their Health Conditions.
Emergency Calls And Rescue
In Emergency Situations, Patients Can Use The Remote Health Control System To Send Emergency Calls Or Distress Signals To Promptly Receive Medical Assistance.

Remote Health System
Home
Serviecs
About Us
Login
Remote Physiological Parameter Monitoring
Through Sensors, Smart Devices, Or Wearable Devices, The System Can Monitor A Patient's Physiological Parameters Such As Heart Rate, Blood Pressure, Blood Oxygen Saturation, Body Temperature, Etc. These Data Can Be Transmitted In Real-Time To Healthcare Providers For Remote Monitoring And Assessment Of The Patient's Health Condition.
Health Data Recording And Analysis
Remote Health Control Systems Can Record And Store The Patient's Health Data, Including Measurement Values, Medication History, Symptom Descriptions, Etc. Healthcare Providers Can Analyze This Data To Gain A Comprehensive Understanding Of The Patient's Condition And Make More Accurate Diagnoses And Treatment Plans.
Remote Medical Consultation And Diagnosis
Patients Can Engage In Real-Time Or Asynchronous Medical Consultations And Diagnoses With Healthcare Providers Through The Remote Health Control Platform. They Can Communicate Their Symptoms, Ask Questions, And Receive Advice And Treatment Plans Through Video Calls, Phone Calls, Or Online Chats.
Medication Reminders And Management
The Remote Health Control System Can Provide Medication Reminders To Help Patients Adhere To Their Medication Schedules. Additionally, Healthcare Providers Can Remotely Monitor The Patient's Medication Adherence Through The System And Make Timely Adjustments To The Treatment Plan If Needed.
Health Education And Guidance
The Remote Health Control Platform Can Offer Relevant Health Education And Guidance To Patients, Including Disease Management, Lifestyle Recommendations, Dietary Advice, Etc. This Helps Patients To Better Understand And Manage Their Health Conditions.
Emergency Calls And Rescue
In Emergency Situations, Patients Can Use The Remote Health Control System To Send Emergency Calls Or Distress Signals To Promptly Receive Medical Assistance.
Create Your Own Profile
Upload Your Body Values
Consult A Doctor Online
Reserve

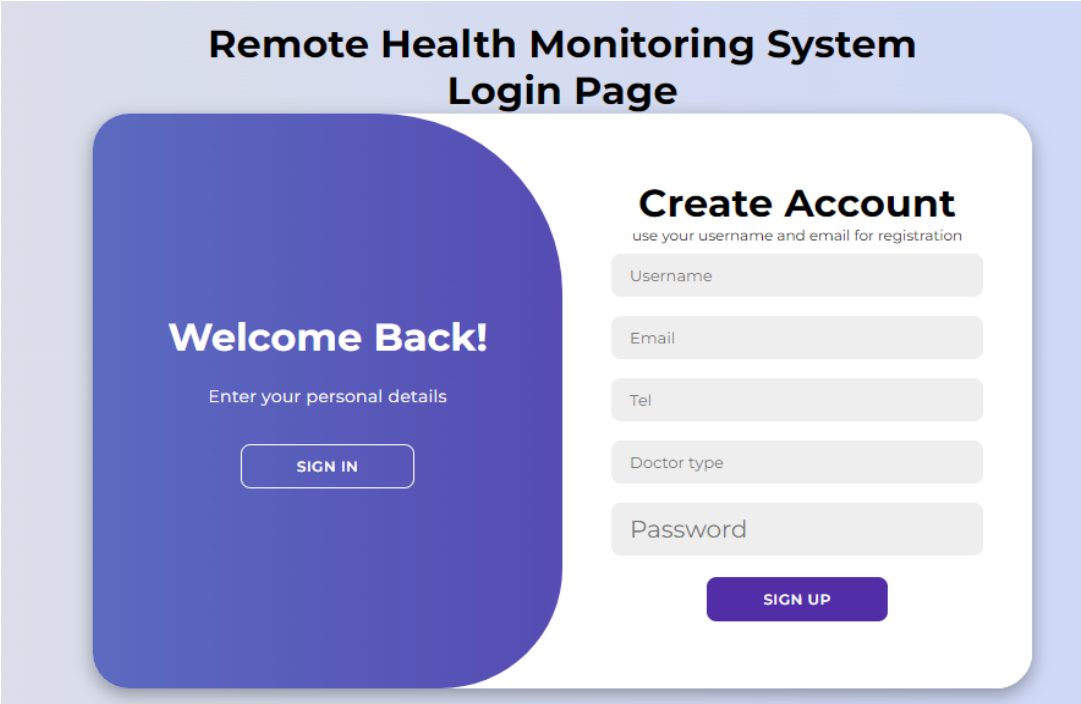
Login – Doctor



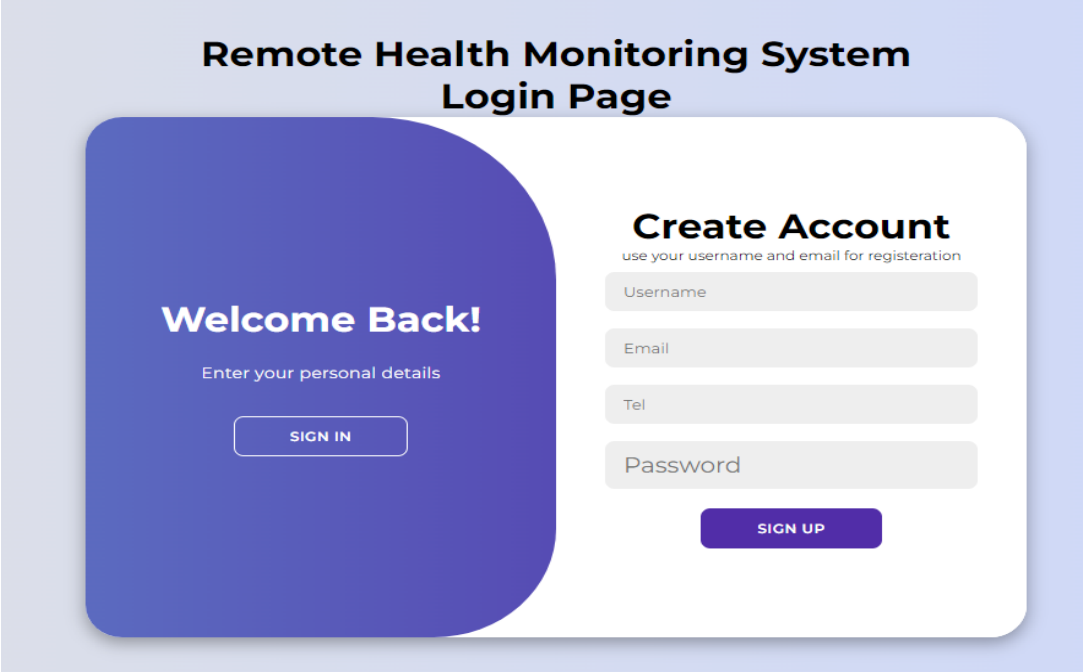
Login – User



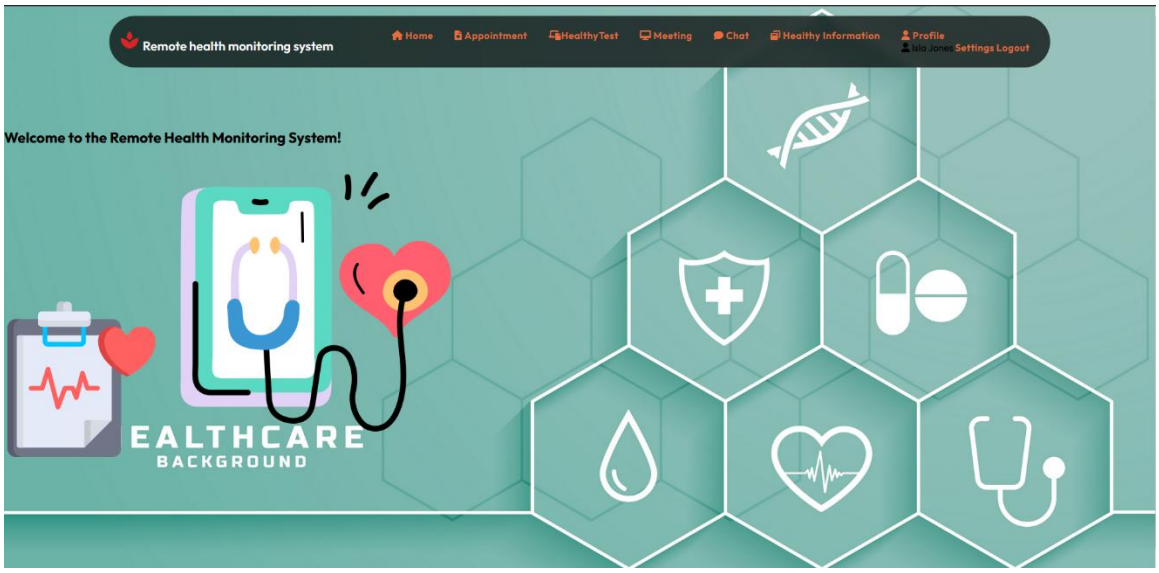
Register – Doctor



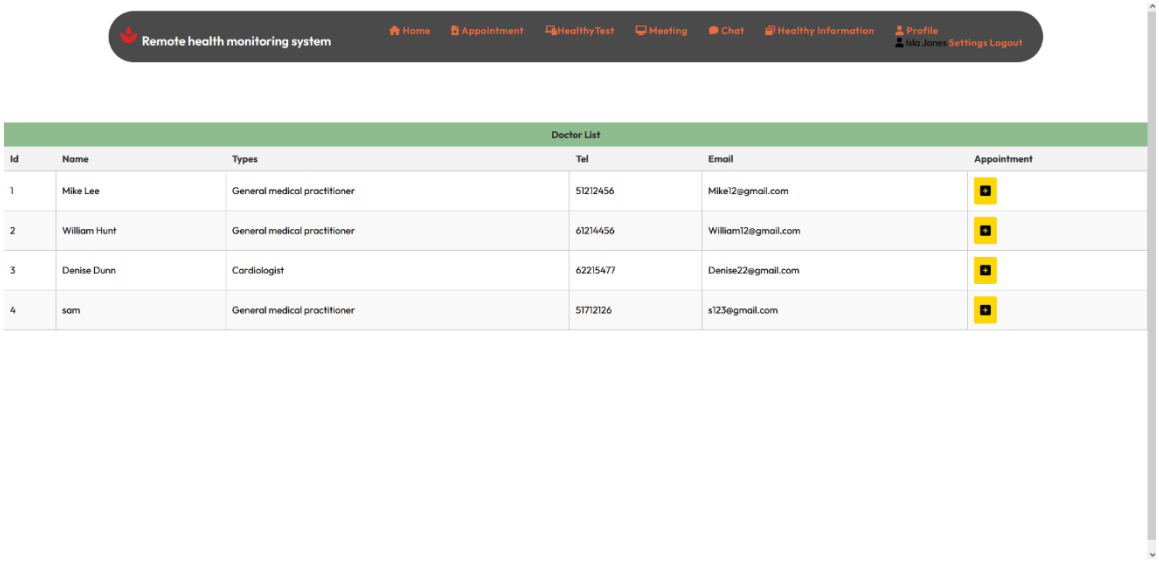
Register – User

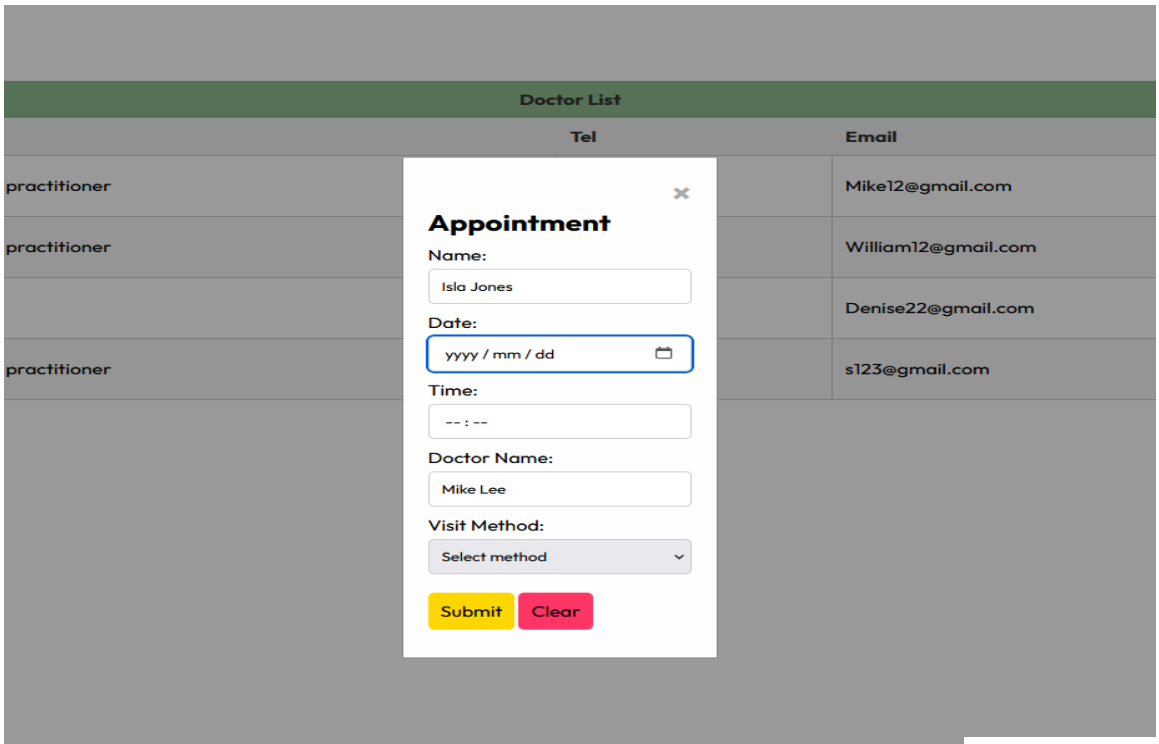


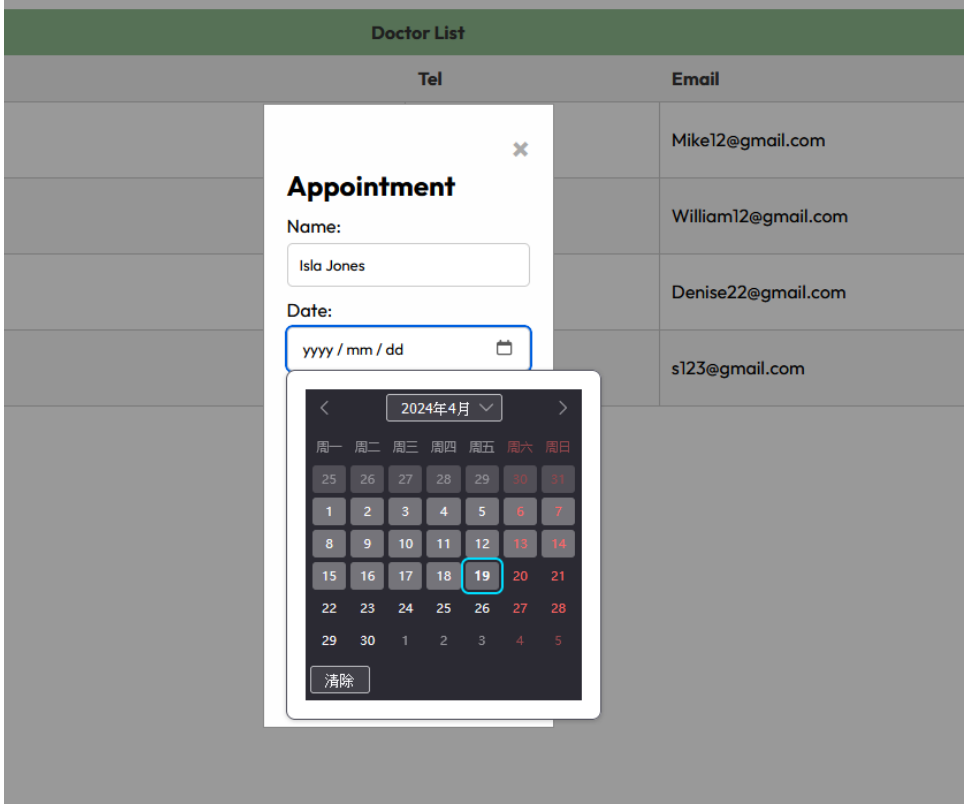
User - Home Page



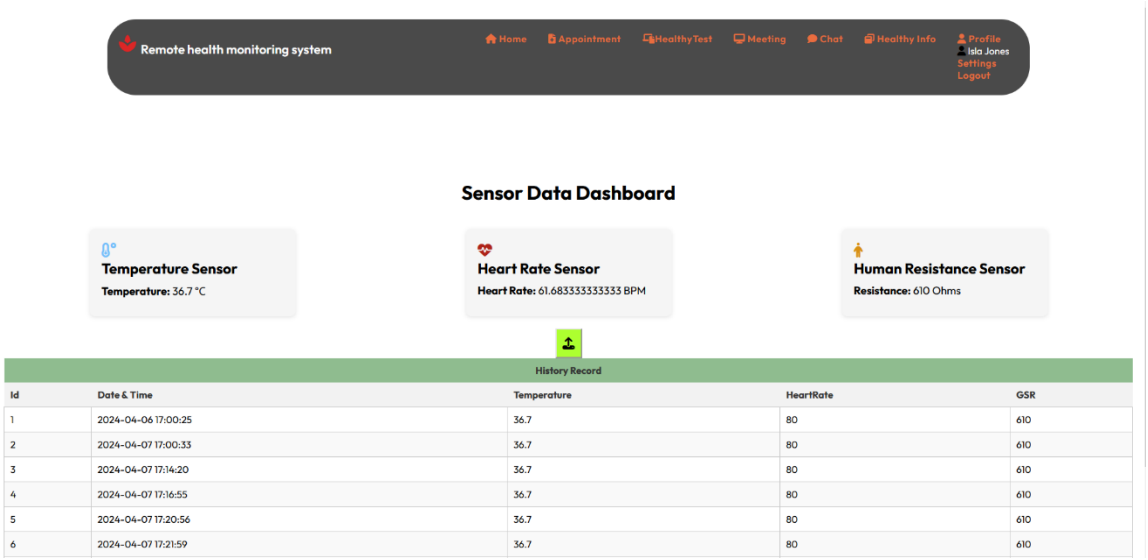
User - Appointment




Doctor List
Tel
Email
practitioner
Mike12@gmail.com
practitioner
William12@gmail.com
Denise22@gmail.com
practitioner
s123@gmail.com
Appointment
Name:
Isla Jones
Date:
yyyy / mm / dd
Time:
-- : --
Doctor Name:
Mike Lee
Visit Method:
Select method
Submit
Clear

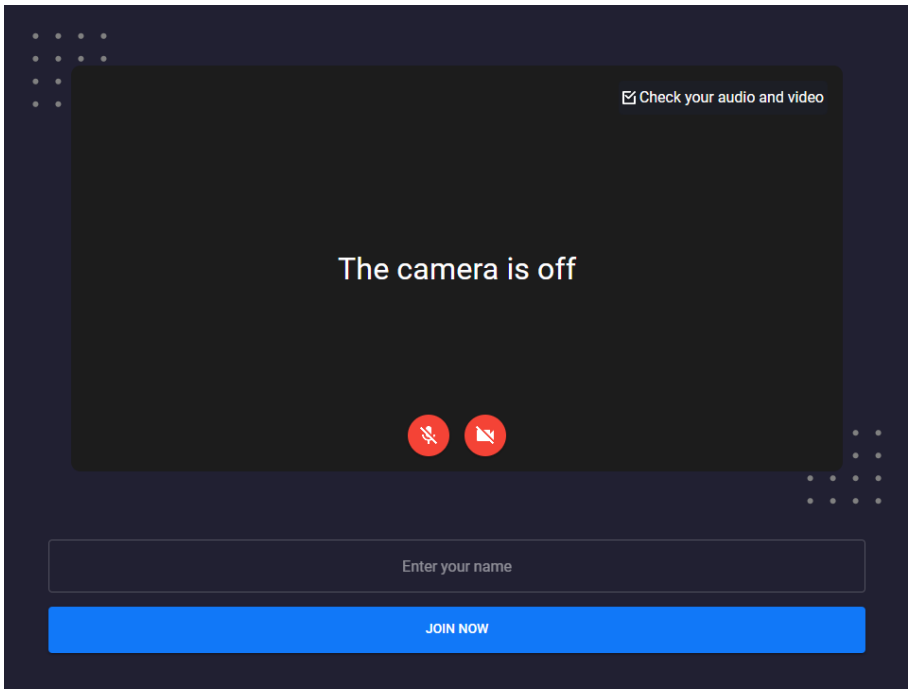

Doctor List
Tel
Email
Mike12@gmail.com
William12@gmail.com
Denise22@gmail.com
s123@gmail.com
Appointment
Name:
Isla Jones
Date:
yyyy / mm / dd
2024年4月
周一 周二 周三 周四 周五 周六 周日
清除

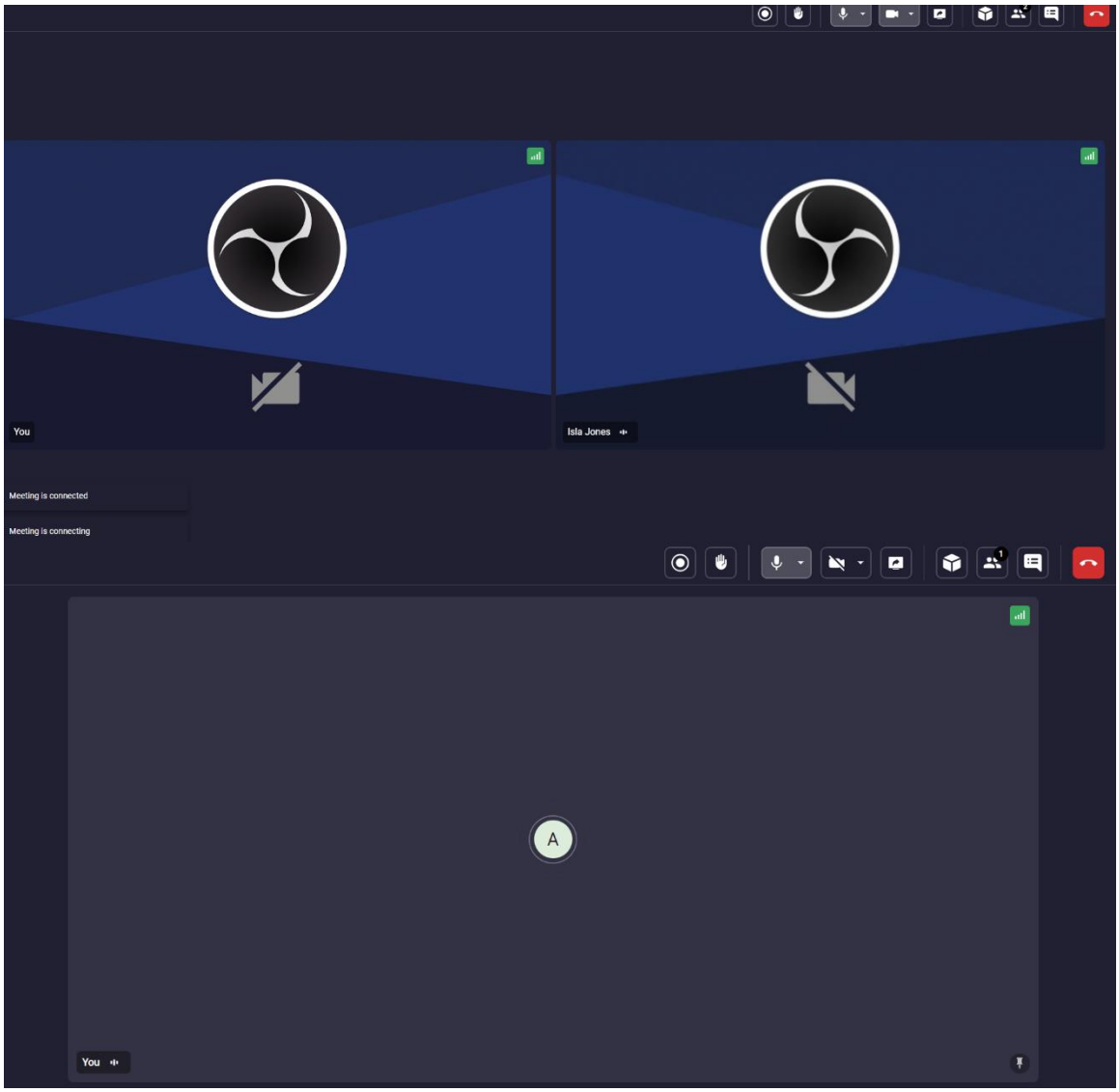
User - Healthy Test

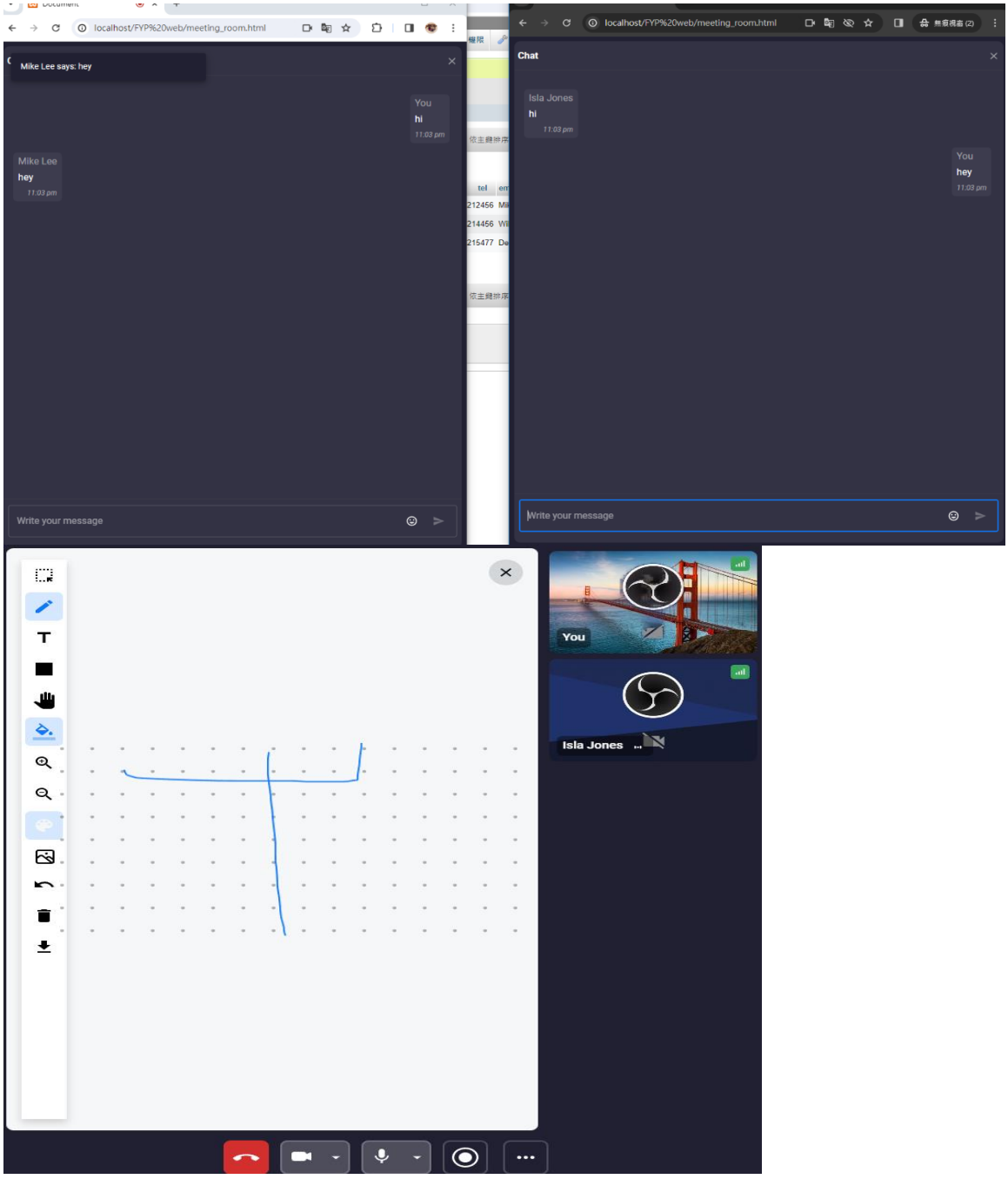


Id	Date & Time	Temperature	HeartRate	GSR
1	2024-04-06 17:00:25	36.7	80	610
2	2024-04-07 17:00:33	36.7	80	610
3	2024-04-07 17:14:20	36.7	80	610
4	2024-04-07 17:16:55	36.7	80	610
5	2024-04-07 17:20:56	36.7	80	610
6	2024-04-07 17:21:59	36.7	80	610

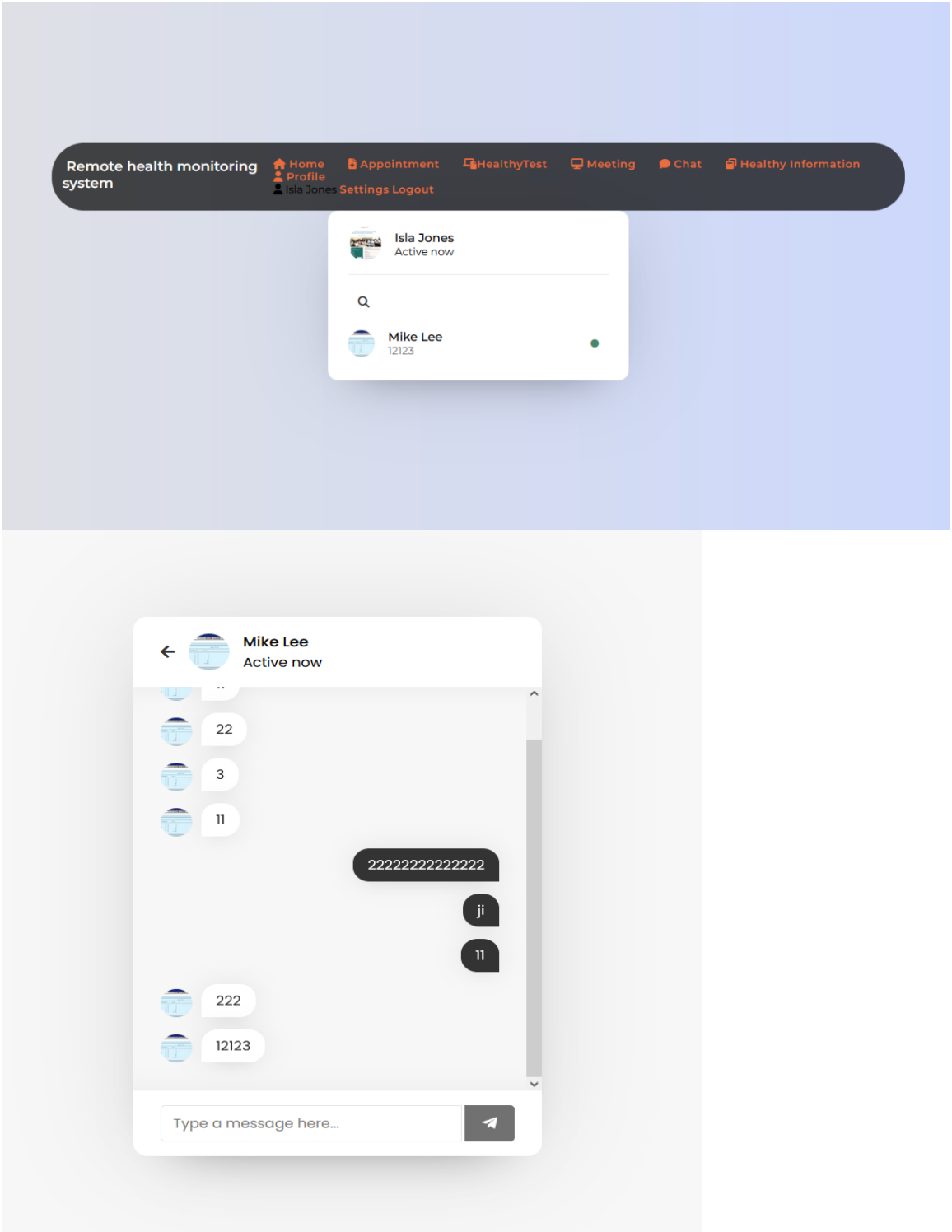
User – Meeting Room



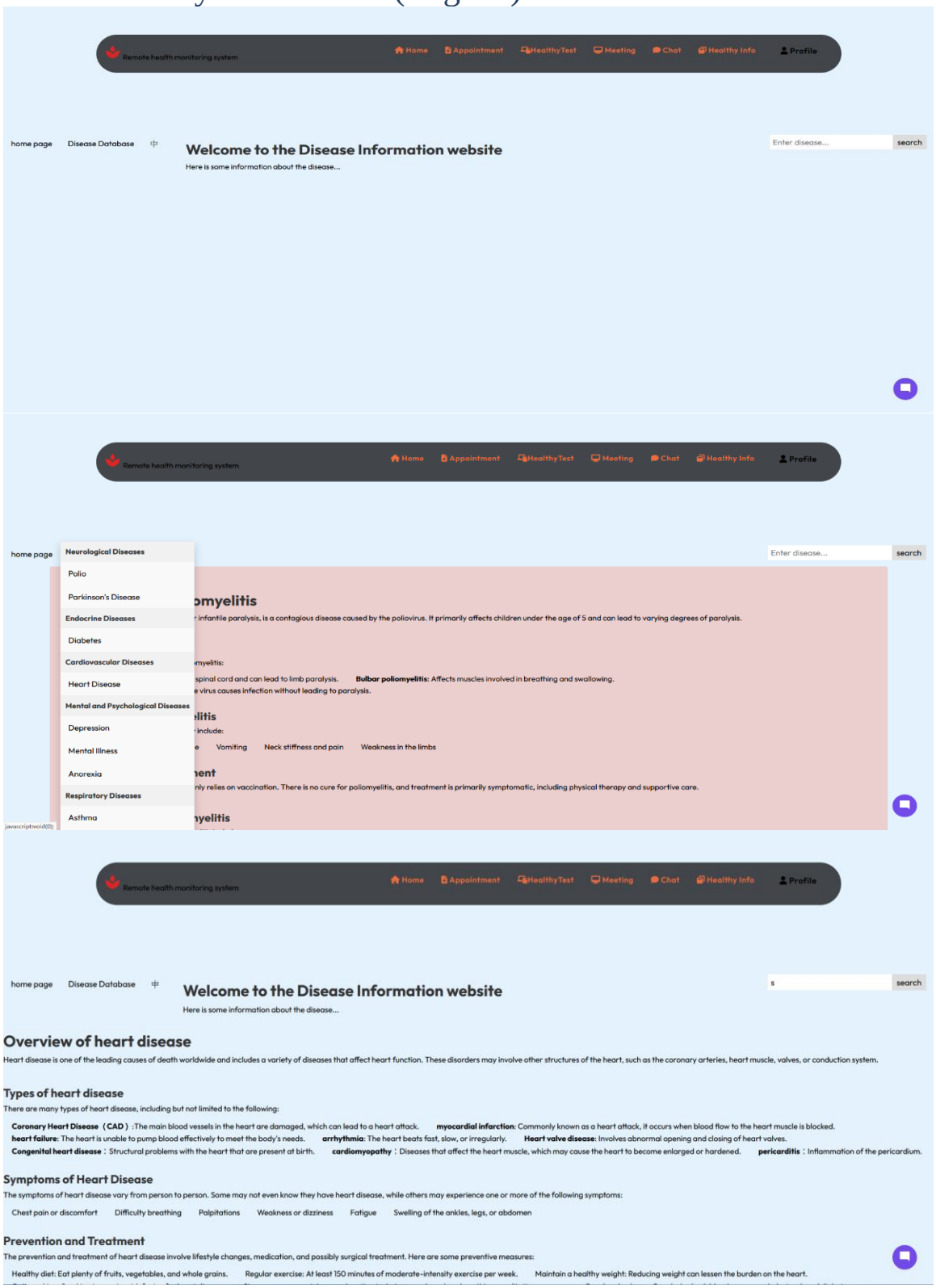

You
Isla Jones
Meeting is connected
Meeting is connecting
A
You


localhost/FYP%20web/meeting_room.html
Mike Lee says: hey
You
hi
11:03 pm
Mike Lee
hey
11:03 pm
Write your message
Chat
Isla Jones
hi
11:03 pm
You
hey
11:03 pm
Write your message
You
Isla Jones

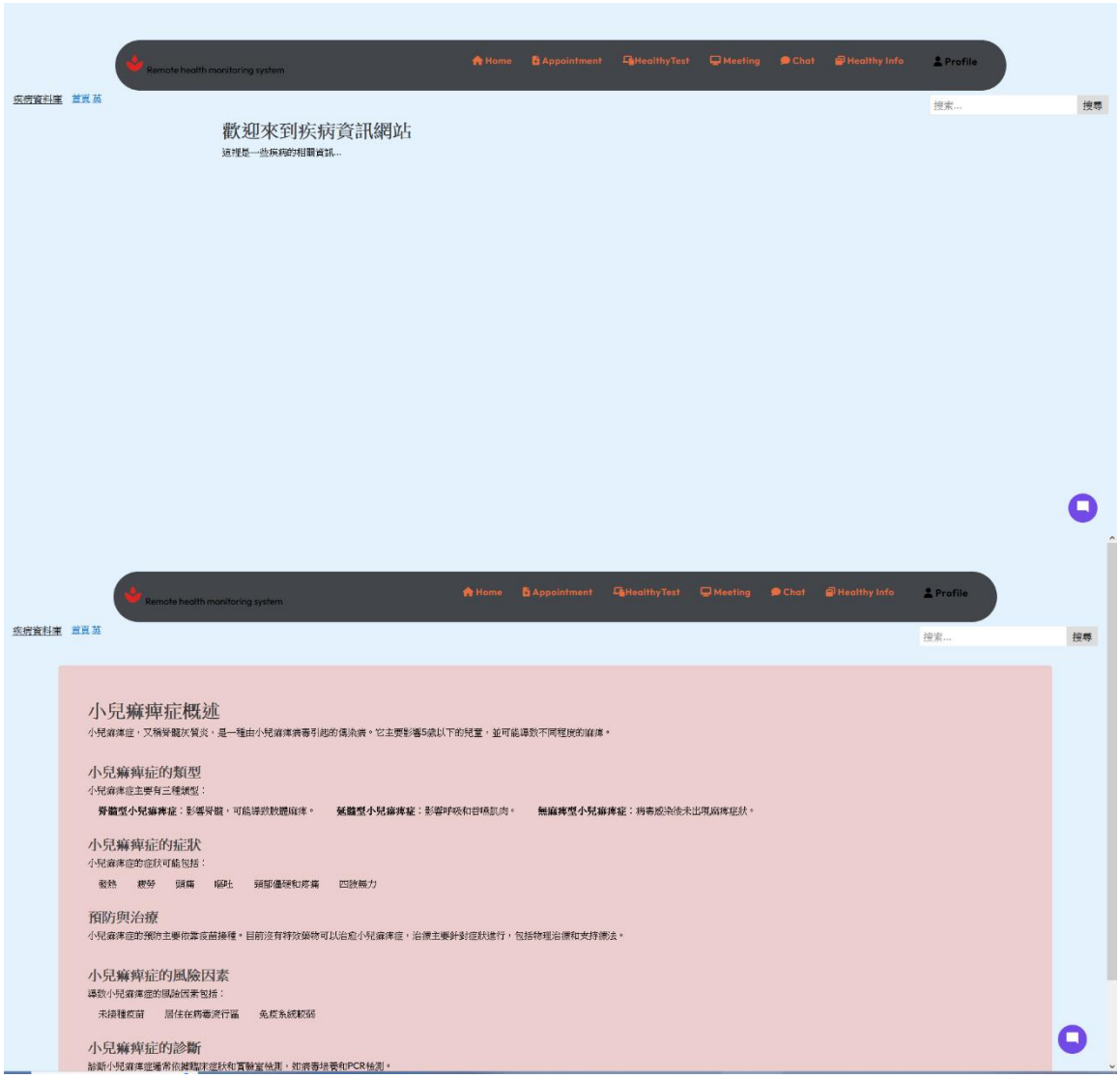
User – Chat Room



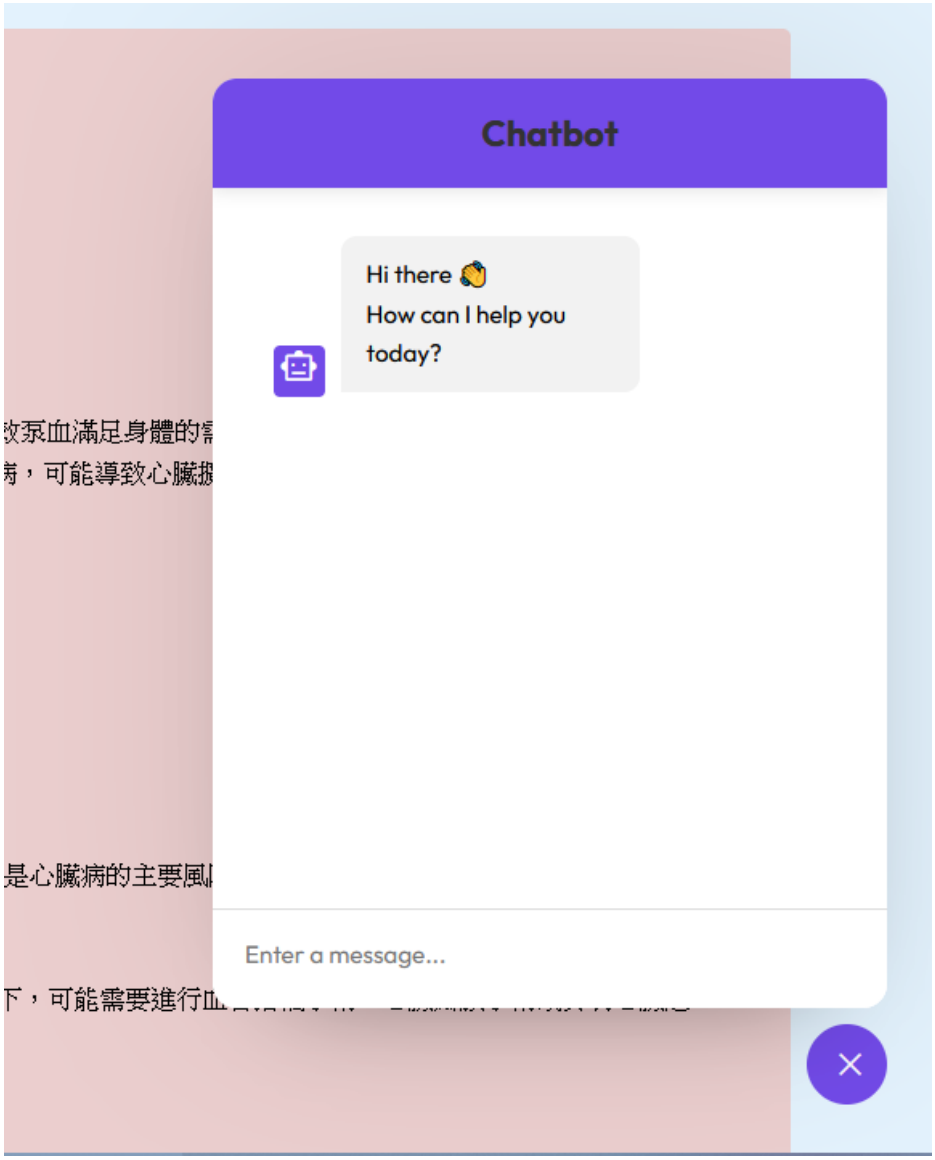
User – Healthy Information (English)



User – Healthy Information (Chinese)

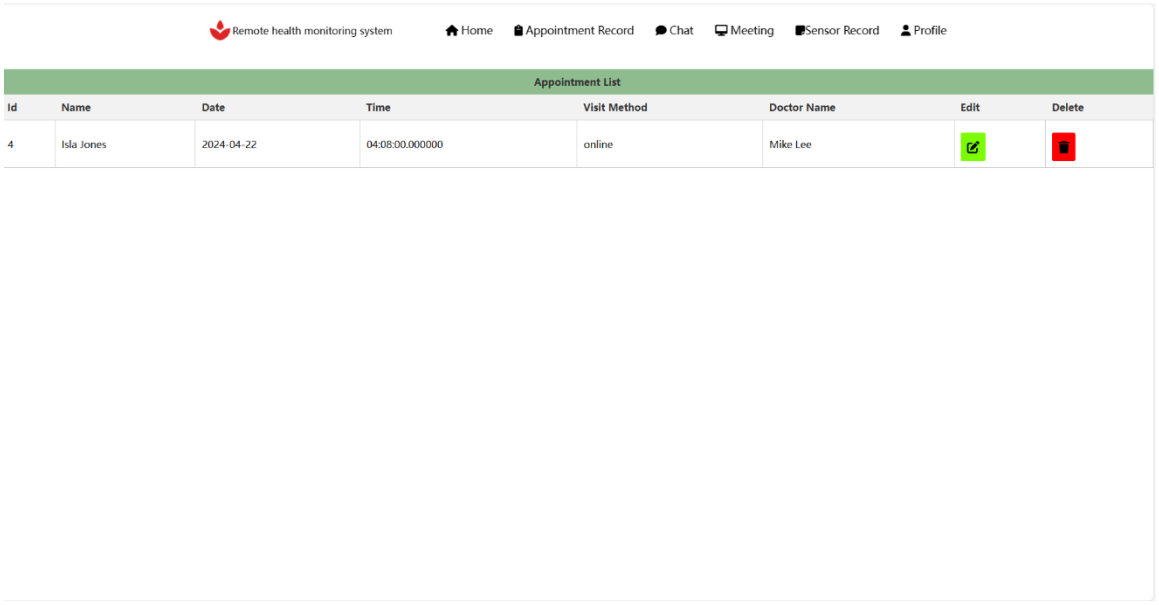


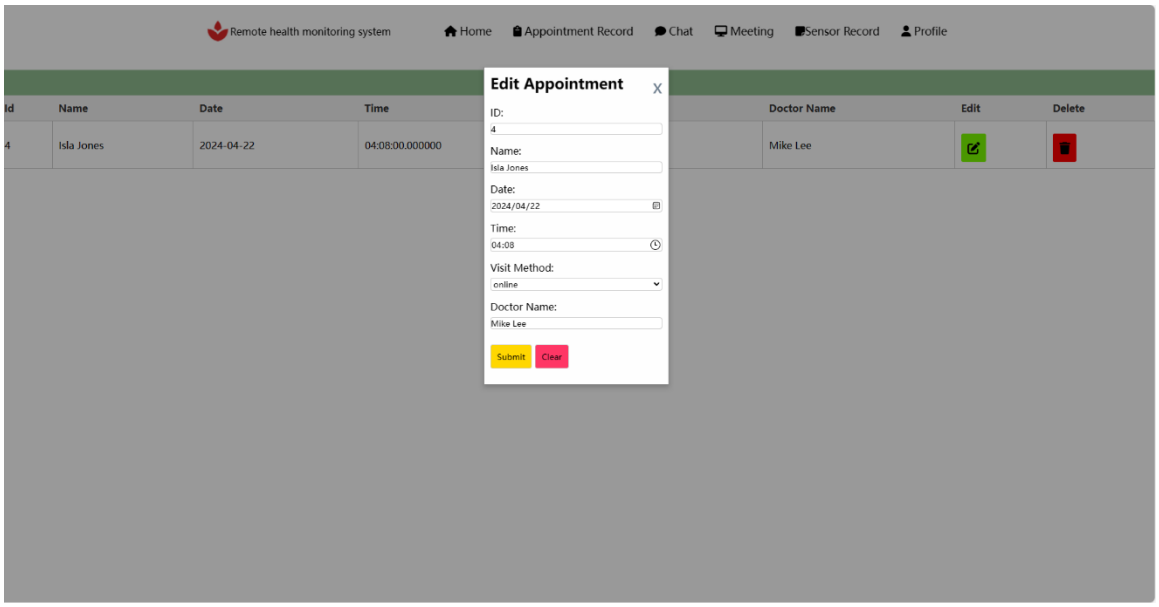
User - Chat Bot



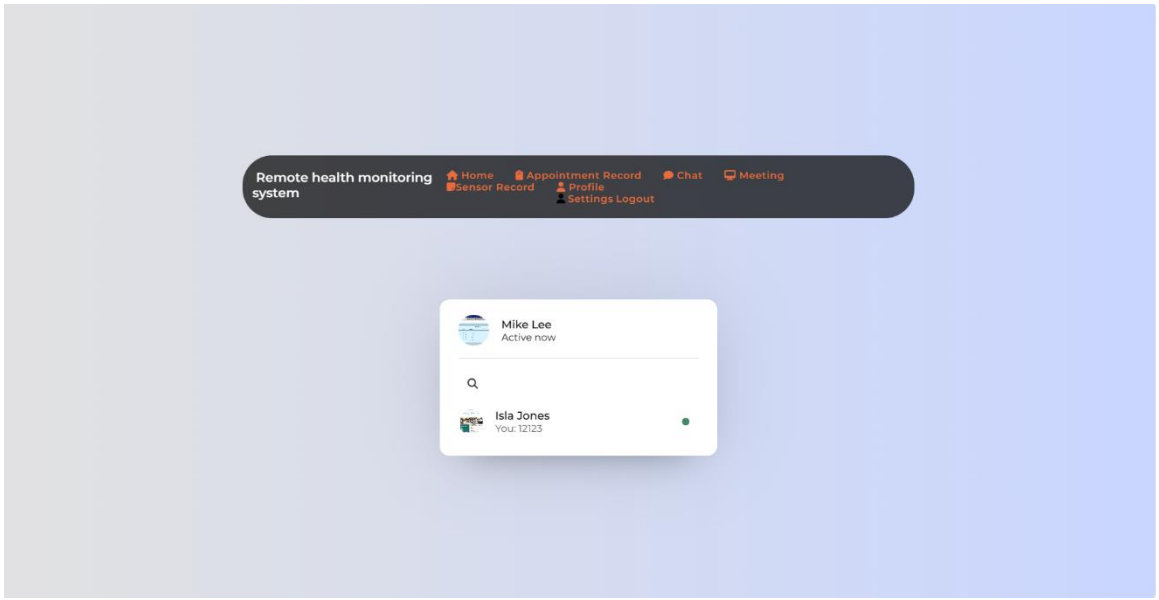
Doctor – Home Page

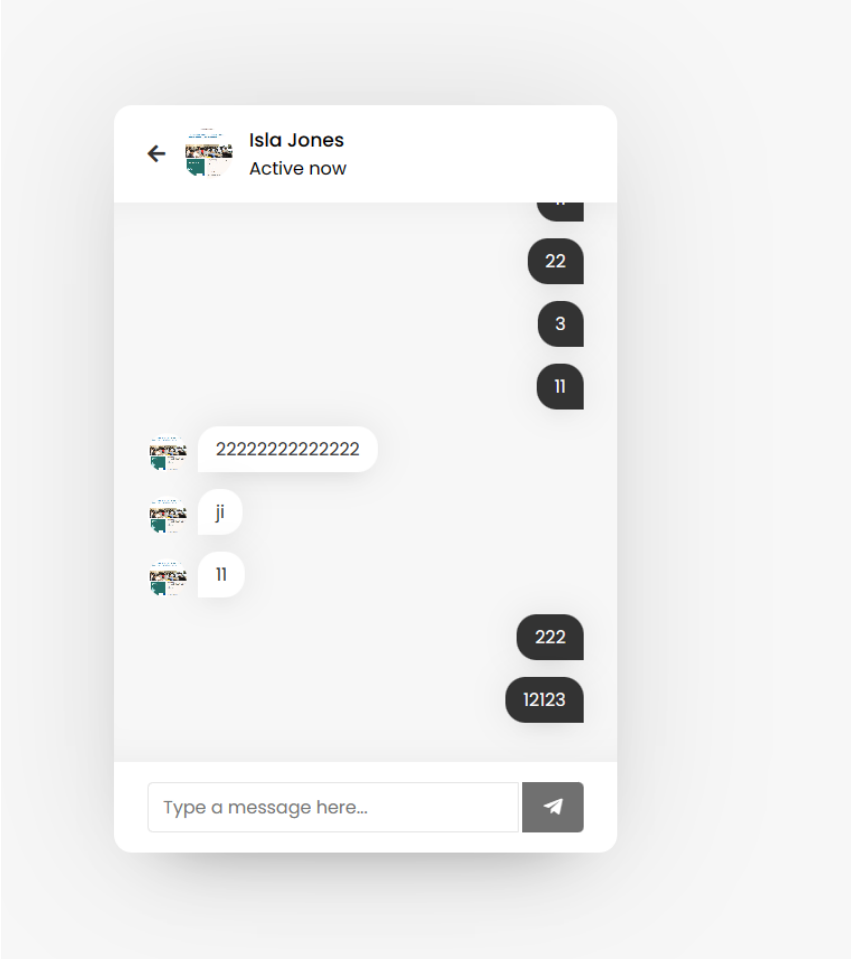
Doctor - Manage Appointment



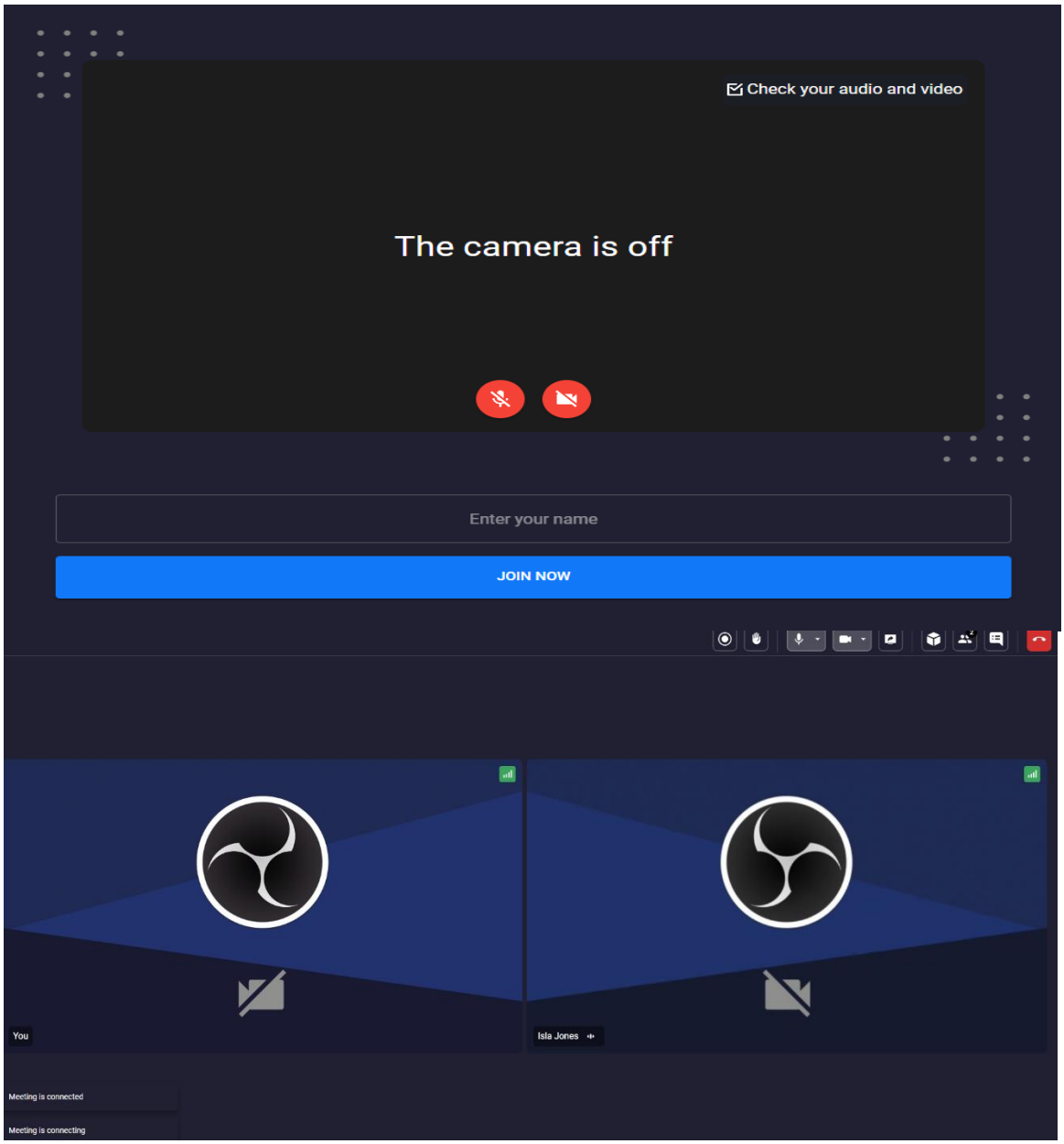


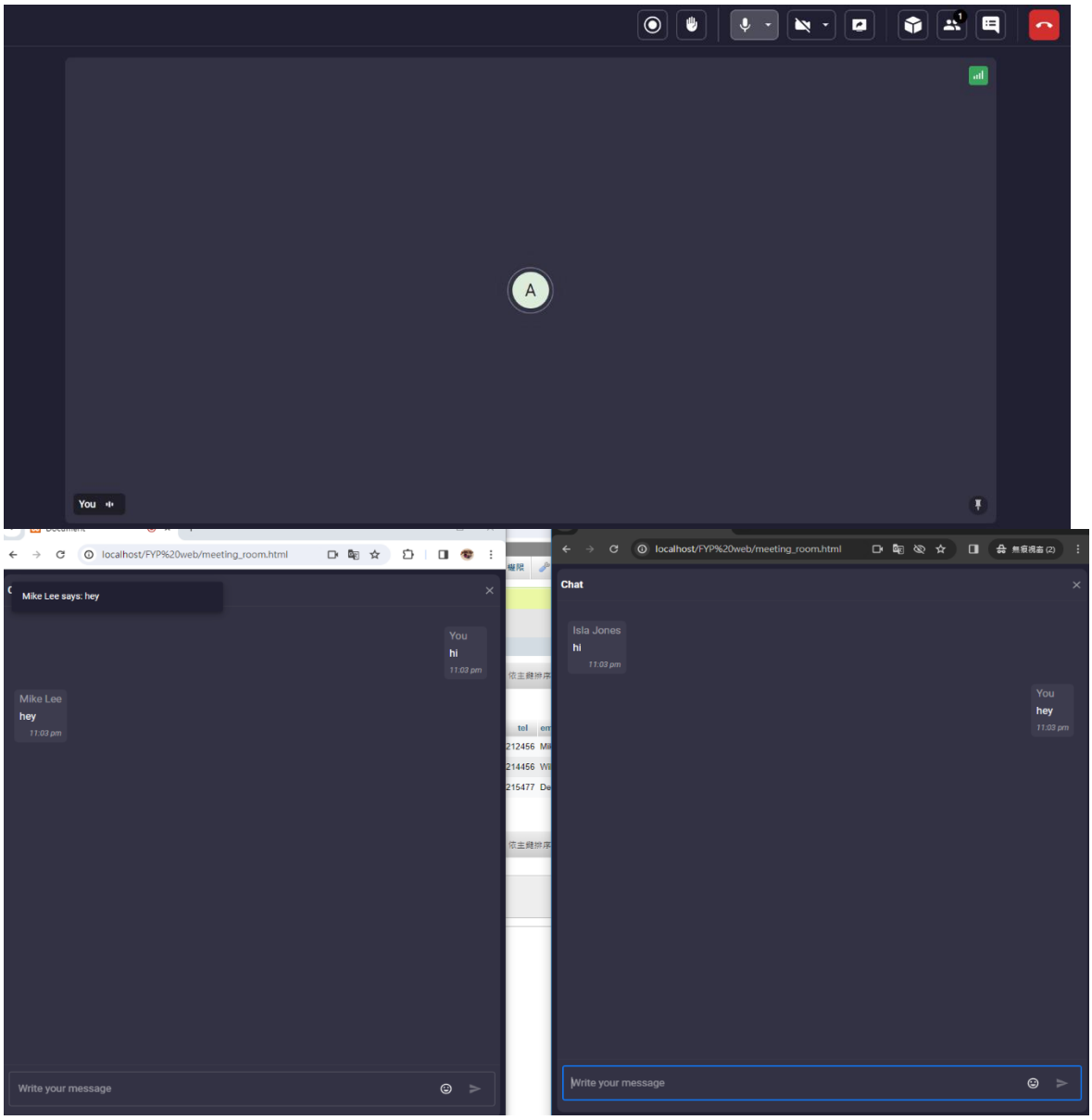
Doctor – Chat Room

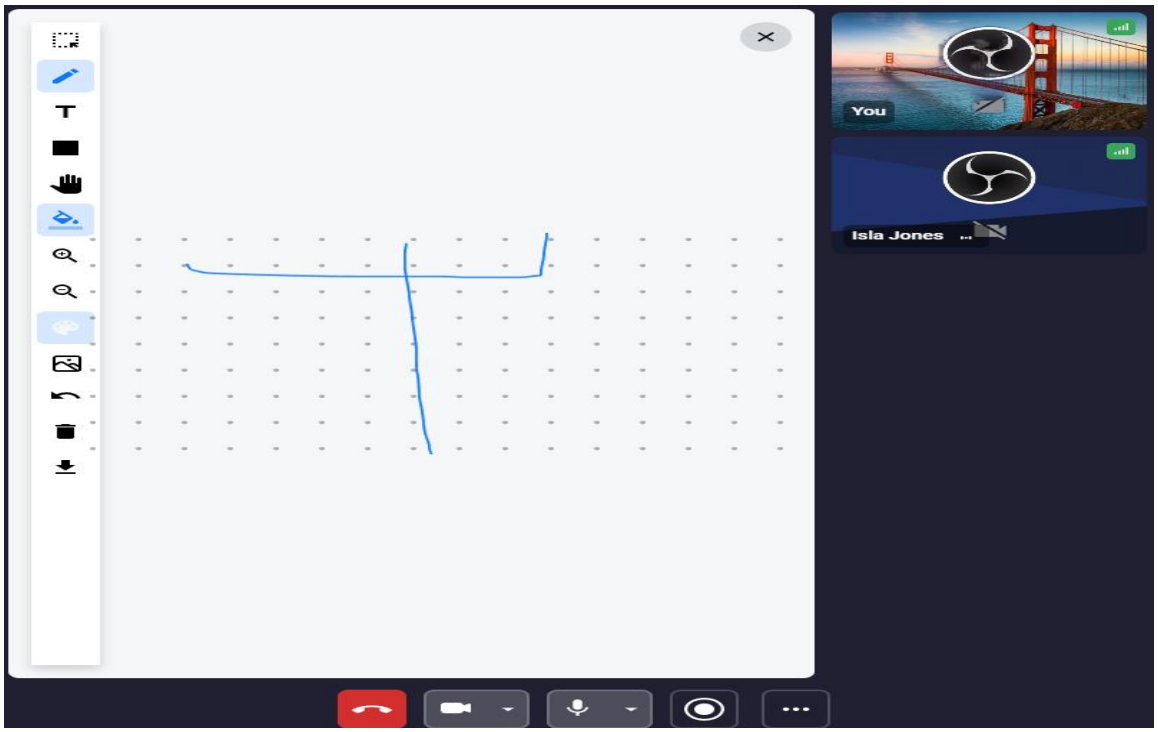



Isla Jones
Active now
22
3
11
22222222222222
ji
11
222
12123
Type a message here...

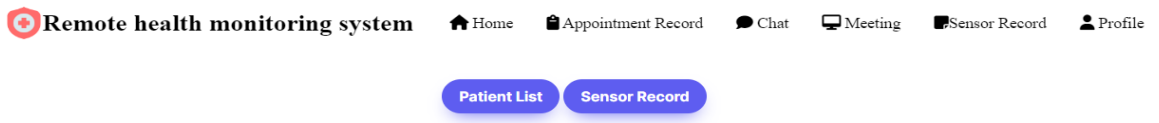
Doctor – Meeting Room



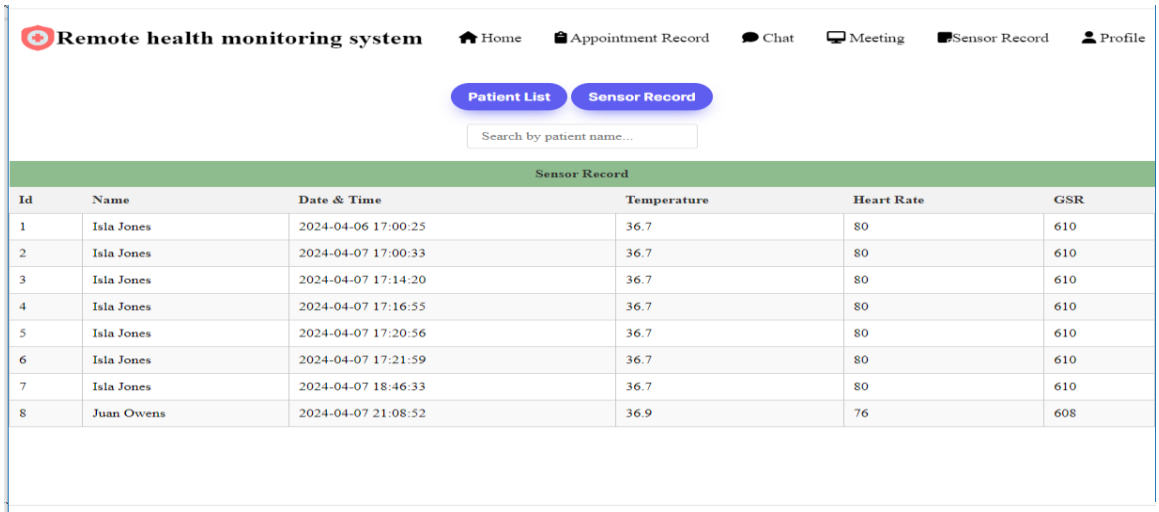

A
You
localhost/FYP%20web/meeting_room.html
Mike Lee says: hey
You
hi
11:03 pm
Mike Lee
hey
11:03 pm
Write your message
localhost/FYP%20web/meeting_room.html
Chat
Isla Jones
hi
11:03 pm
You
hey
11:03 pm
Write your message


You
Isla Jones

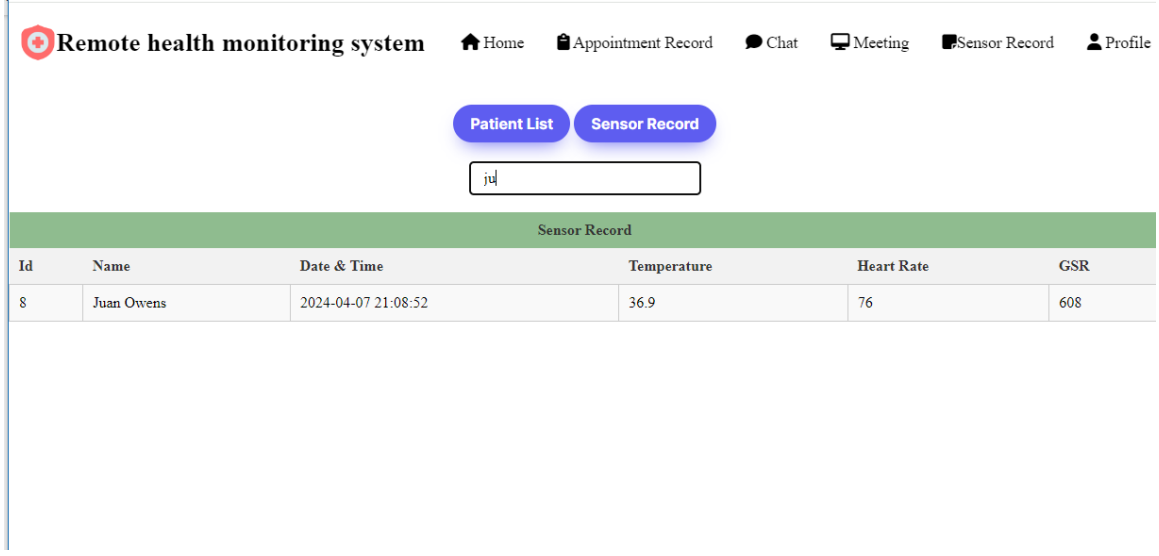
Doctor – Sensor Record



Patient List			
Id	Name	Email	Tel
1	Isla Jones	Isla2j@gmail.com	55678899
2	Juan Owens	Juow11@gmail.com	65446322
3	TAM	s123@gmail.com	51712126
4	sssaa	s123@gmail.com	51234561



Sensor Record					
Id	Name	Date & Time	Temperature	Heart Rate	GSR
1	Isla Jones	2024-04-06 17:00:25	36.7	80	610
2	Isla Jones	2024-04-07 17:00:33	36.7	80	610
3	Isla Jones	2024-04-07 17:14:20	36.7	80	610
4	Isla Jones	2024-04-07 17:16:55	36.7	80	610
5	Isla Jones	2024-04-07 17:20:56	36.7	80	610
6	Isla Jones	2024-04-07 17:21:59	36.7	80	610
7	Isla Jones	2024-04-07 18:46:33	36.7	80	610
8	Juan Owens	2024-04-07 21:08:52	36.9	76	608



Sensor Record					
Id	Name	Date & Time	Temperature	Heart Rate	GSR
8	Juan Owens	2024-04-07 21:08:52	36.9	76	608

11. Critical Evaluation

11.1 Difficulties encountered

1. Technical Challenges: The implementation of certain features, such as login system and appointment system, required extensive testing and debugging to ensure smooth functionality. Dealing with compatibility issues, software conflicts, or integrating different technologies posed challenges that required additional time and effort.

2. Resource Limitations: Limited availability of resources, both in terms of hardware and personnel, impacted the development process. The system's reliance on XAMPP for the server infrastructure posed constraints on scalability and performance.

11.2 Imitations of the proposed system

1. Dependency on XAMPP for Server: The system relies on XAMPP, a local server solution, for hosting the application. This dependency may limit scalability and accessibility since it requires the system to be hosted on a local machine rather than a dedicated server or cloud-based infrastructure.

2. Lack of Financial Resources for Setting up a Server: The proposed system may face challenges in terms of funding to set up a dedicated server infrastructure. This limitation can impact the system's performance, reliability, and availability, as it may not be able to handle many users or provide consistent service uptime.

3. Restricted Usage to a Single Computer: Due to the limitations of the system's infrastructure, it may only allow for simultaneous usage of two webpages on the same computer. This restriction can hinder scalability and limit the number of concurrent users who can access the system simultaneously.

11.3 Potential difficulties

1. User Adoption and Acceptance: Introducing a new system in the healthcare domain may face resistance or hesitancy from both healthcare providers and patients. Overcoming user resistance, addressing concerns about usability and trust, and effectively promoting the benefits of the system are essential for successful user adoption and acceptance.

2. Stakeholder Collaboration: Building a successful remote health monitoring system requires collaboration among various stakeholders, including healthcare providers, technology vendors, regulatory bodies, and patients.

3. Regulatory and legal considerations: Remote health monitoring systems must comply with various regulatory requirements, such as data protection laws, medical device regulations, and telemedicine guidelines. Navigating through these regulations and ensuring compliance can be complex and time-consuming, potentially delaying the implementation process.

12.All test cases and test data

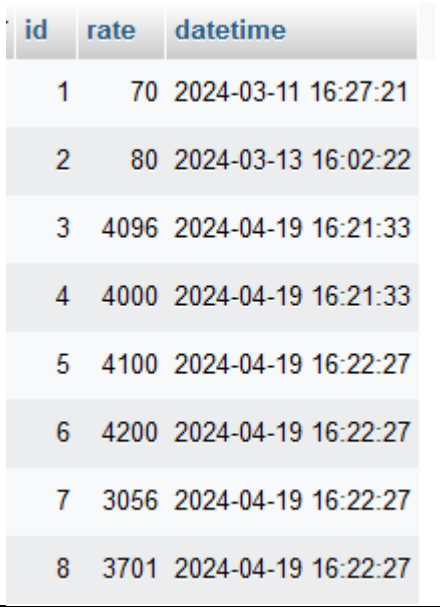
12.1 Login

<table>
<tr><td>Test Case ID</td><td>TC001</td></tr>
<tr><td>Test Description</td><td>Verify that the login functionality works with valid credentials</td></tr>
<tr><td>Preconditions</td><td>The user must be at the login page</td></tr>
<tr><td>Test Steps</td><td>1.Enter the username in the username field.
2.Enter the password in the password field.
3.Click the login button.</td></tr>
<tr><td>Test Data</td><td>User
<table><tr><th>userID</th><th>userName</th><th>Email</th><th>tel</th><th>password</th></tr><tr><td>1</td><td>Isla Jones</td><td>Isla2j@gmail.com</td><td>55678899</td><td>Jo123@66</td></tr><tr><td>2</td><td>Juan Owens</td><td>Juow11@gmail.com</td><td>65446322</td><td>Juan123@</td></tr><tr><td>3</td><td>TAM</td><td>s123@gmail.com</td><td>51712126</td><td>Ss123456</td></tr><tr><td>4</td><td>sssaa</td><td>s123@gmail.com</td><td>51234561</td><td>Ss123456</td></tr></table>
Doctor
<table><tr><th>doctorID</th><th>name</th><th>types</th><th>tel</th><th>email</th><th>password</th></tr><tr><td>1</td><td>Mike Lee</td><td>General medical practitioner</td><td>51212456</td><td>Mike12@gmail.com</td><td>Mike11l@</td></tr><tr><td>2</td><td>William Hunt</td><td>General medical practitioner</td><td>61214456</td><td>William12@gmail.com</td><td>Wi22sh7@</td></tr><tr><td>3</td><td>Denise Dunn</td><td>Cardiologist</td><td>62215477</td><td>Denise22@gmail.com</td><td>Dencar2@</td></tr><tr><td>4</td><td>sam</td><td>General medical practitioner</td><td>51712126</td><td>s123@gmail.com</td><td>Ss123456</td></tr></table></td></tr>
<tr><td>Expected Result</td><td>The user is redirected to the dashboard after successful login</td></tr>
<tr><td>Actual Result</td><td>Login successful</td></tr>
<tr><td>Status</td><td>(To be filled after execution)</td></tr>
<tr><td>Comments</td><td>Null</td></tr>
</table>

12.2 Make Appointment

Test Case ID	TC002
Test Description	Make the Appointment
Preconditions	The user must be at the login page
Test Steps	1. Click the button ‘Make appointment’ 2.Choose value the Date. 3.Choose the type of appointment. 4.Choose the Doctor.
Test Data	id | name | date | time | VisitMethod | doctorName 2 | Isla Jones | 2024-01-25 | 00:29:00.000000 | face to face | William Hunt 3 | Isla Jones | 2024-04-23 | 22:43:00.000000 | face to face | sam 4 | Isla Jones | 2024-04-22 | 04:08:00.000000 | online | Mike Lee
Expected Result	The user can see the successful message after successful making appointment
Actual Result	Make Appointment successful
Status	(To be filled after execution)
Comments	Null

12.3 Healthy Test

Test Case ID	TC003
Test Description	Do the Healthy Test and upload the data to the system
Preconditions	The user must be at the login page
Test Steps	1.Set up the sensor. 2. Click the button “Healthy Test” 3.Use the sensor to check the body data. 4.Click the green button to upload the data
Test Data	
Expected Result	The user can see check and upload the data after using the sensor to check the body information
Actual Result	View and upload the data
Status	(To be filled after execution)
Comments	Null

id	rate	datetime
1	70	2024-03-11 16:27:21
2	80	2024-03-13 16:02:22
3	4096	2024-04-19 16:21:33
4	4000	2024-04-19 16:21:33
5	4100	2024-04-19 16:22:27
6	4200	2024-04-19 16:22:27
7	3056	2024-04-19 16:22:27
8	3701	2024-04-19 16:22:27

12.4 Chat Room

<table>
<tr><td>Test Case ID</td><td>TC004</td></tr>
<tr><td>Test Description</td><td>Communicate with each other successfully</td></tr>
<tr><td>Preconditions</td><td>The user(doctor) must be at the login page</td></tr>
<tr><td>Test Steps</td><td>1.Enter the Chat Room.
2.Click the Send button.</td></tr>
<tr><td>Test Data</td><td>Message
<table>
<tr><th>msg_id</th><th>incoming_msg_id</th><th>outgoing_msg_id</th><th>msg</th></tr>
<tr><td>1</td><td>1572804435</td><td>1110990992</td><td>11</td></tr>
<tr><td>2</td><td>1572804435</td><td>1110990992</td><td>22</td></tr>
<tr><td>3</td><td>1572804435</td><td>1110990992</td><td>3</td></tr>
<tr><td>4</td><td>1572804435</td><td>1110990992</td><td>11</td></tr>
<tr><td>5</td><td>1110990992</td><td>1572804435</td><td>22222222222222</td></tr>
<tr><td>6</td><td>1110990992</td><td>1572804435</td><td>ji</td></tr>
<tr><td>7</td><td>1110990992</td><td>1572804435</td><td>11</td></tr>
<tr><td>8</td><td>1572804435</td><td>1110990992</td><td>222</td></tr>
<tr><td>9</td><td>1572804435</td><td>1334909215</td><td>sss</td></tr>
<tr><td>10</td><td>1110990992</td><td>1334909215</td><td>sss</td></tr>
<tr><td>11</td><td>1572804435</td><td>1110990992</td><td>12123</td></tr>
</table>
</td></tr>
<tr><td>Expected Result</td><td>The user(doctor) can chat with others successfully</td></tr>
<tr><td>Actual Result</td><td>Send and Received Successfully</td></tr>
<tr><td>Status</td><td>(To be filled after execution)</td></tr>
<tr><td>Comments</td><td>Null</td></tr>
</table>

12.5 Meeting

Test Case ID	TC005
Test Description	Verify that the meeting room functionality works with valid credentials
Preconditions	The user(doctor) must be at the meeting room page
Test Steps	1.Enter the username in the username field. 2.Click the join meeting button.
Test Data	User userName Isla Jones Juan Owens TAM sssaa Doctor name Mike Lee William Hunt Denise Dunn sam
Expected Result	The user(doctor) is redirected to the meeting room after enter username and click join meeting button
Actual Result	Enter to the meeting room successful
Status	(To be filled after execution)
Comments	Null

12.6 Chatbot

Test Case ID	TC006
Test Description	Verify that the chatbot functionality works with valid credentials
Preconditions	The users must be at the search information page
Test Steps	1.Enter the questions or information in the chat bot 2.Click the enter button to wait for the reply of the chatbot
Test Data	Keywords of the questions or information
Expected Result	The chatbot reply to the user with relevant answer or information
Actual Result	Users get the answer or information with high accuracy
Status	(To be filled after execution)
Comments	Null

12.7 Search Page

Test Case ID	TC007
Test Description	Search the information of the symptom
Preconditions	The user must be at the login page
Test Steps	1.Enter the Search Page. 2.Enter the keywords 3.Click the Search button.
Test Data	Search Data • Fever • Cough • Headache • Muscle pain • Sore throat • Fatigue • Nausea • Vomiting • Diarrhea • Nasal congestion • Runny nose
Expected Result	The user searches the correct information
Actual Result	Show the information about the symptom
Status	(To be filled after execution)
Comments	Null

12.8 Manage Appointment

Test Case ID	TC008
Test Description	Manage the Appointment which is applied from the users
Preconditions	The user must be at the login page
Test Steps	1.Click the button “Appointment Record” 2.View the Appointment Record. 3.Click the Green button to edit (can change the time of appointment) 4.Click the Red button to delete.
Test Data	id | name | date | time | VisitMethod | doctorName 2 | Isla Jones | 2024-01-25 | 00:29:00.000000 | face to face | William Hunt 3 | Isla Jones | 2024-04-23 | 22:43:00.000000 | face to face | sam 4 | Isla Jones | 2024-04-22 | 04:08:00.000000 | online | Mike Lee
Expected Result	The user searches the correct information
Actual Result	Show the information about the symptom
Status	(To be filled after execution)
Comments	Null

12.9 View Sensor Record

Test Case ID	TC009
Test Description	Doctor can view the record and its details
Preconditions	The user must be at the login page
Test Steps	1.Click the button “Sensor Record” 2.View the Sensor Record. 3.Click the ‘Patient List’ button to view the information of the patient 4.Click the ‘Sensor Record’ button to view all the record.
Test Data	
Expected Result	The user searches the correct information
Actual Result	Show the information about the symptom
Status	(To be filled after execution)
Comments	Null

id	userName	datetime	tem_C	heartrate	gsr
1	Isla Jones	2024-04-06 17:00:25	36.7	80	610
2	Isla Jones	2024-04-07 17:00:33	36.7	80	610
3	Isla Jones	2024-04-07 17:14:20	36.7	80	610
4	Isla Jones	2024-04-07 17:16:55	36.7	80	610
5	Isla Jones	2024-04-07 17:20:56	36.7	80	610
6	Isla Jones	2024-04-07 17:21:59	36.7	80	610
7	Isla Jones	2024-04-07 18:46:33	36.7	80	610
8	Juan Owens	2024-04-07 21:08:52	36.9	76	608
9	Isla Jones	2024-04-19 22:16:57	36.7	3701	610
10	Isla Jones	2024-04-19 22:32:02	36.7	62	610
11	Isla Jones	2024-04-19 22:53:08	36.7	62	610

13. Project plan

1. Project Schedule:

The project will be divided into several phases, each with specific tasks and milestones. The estimated timeline for the project is as follows:

Phase 1: Project Initiation
- Define project scope and objectives
- Formulate project team and roles
- Identify project stakeholders

Phase 2: Requirements Gathering and Analysis
- Conduct user interviews and research
- Define system requirements and specifications
- Identify hardware and software dependencies

Phase 3: Development
- Design the user interface
- Implement client-side using HTML, CSS, JavaScript, and jQuery
- Implement server-side using PHP and XAMPP
- Integrate sensor data acquisition and processing components
- Implement functionality

Phase 4: Testing
- Perform unit testing on individual components
- Conduct integration testing
- Perform system testing and bug fixing

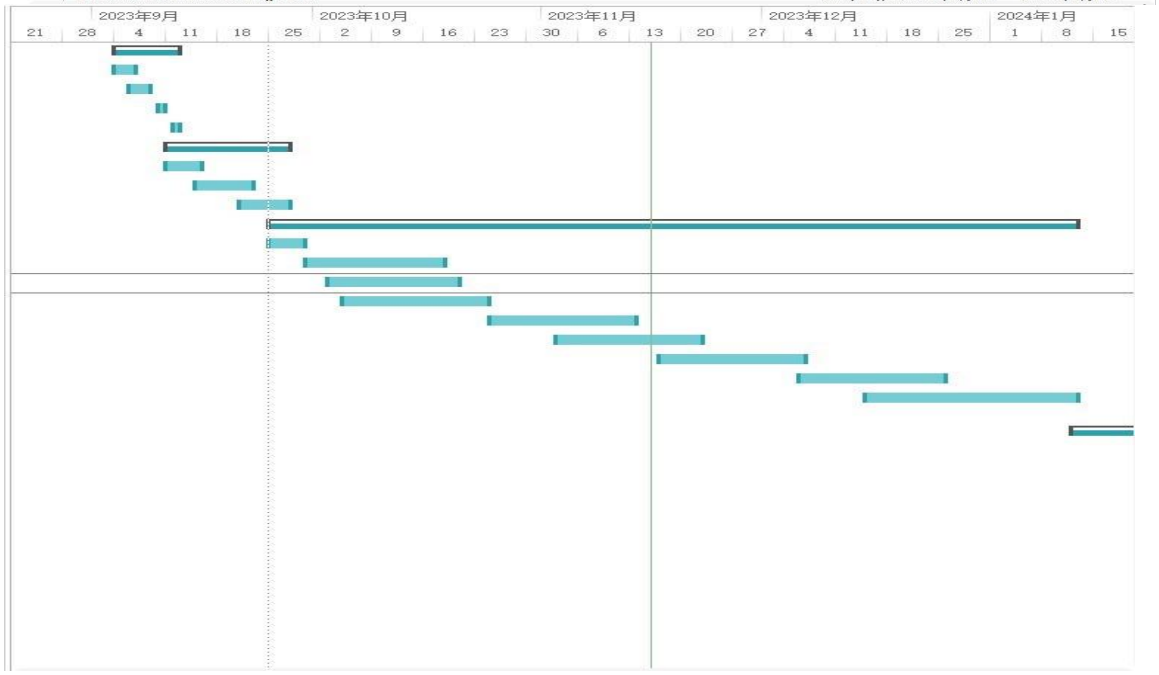
Phase 5: Deployment
- Set up the server environment
- Deploy the system on the server
- Perform final testing and troubleshooting

Phase 6: Documentation and Training
- Prepare comprehensive project documentation
- Create user manuals and guides

日程表

甘特图

	任务名称	工期	开始时间	完成时间
1	**Project Initiation Phase**	**7 个工作日**	**2023年9月4日**	**2023年9月12日**
2	Define project scope and objectives	3 个工作日	2023年9月4日	2023年9月6日
3	Identify stakeholders and their roles.	3 个工作日	2023年9月6日	2023年9月8日
4	Create a project plan and timeline.	1 个工作日	2023年9月10日	2023年9月10日
5	Develop a project charter.	1 个工作日	2023年9月12日	2023年9月12日
6	**Requirements Gathering and Analysis Phase**	**14 个工作日**	**2023年9月11日**	**2023年9月27日**
7	Define functional and non-functional requirements.	5 个工作日	2023年9月11日	2023年9月15日
8	Create use cases and user stories.	7 个工作日	2023年9月15日	2023年9月22日
9	Develop system architecture and design.	5 个工作日	2023年9月21日	2023年9月27日
10	**Development Phase**	**80 个工作日**	**2023年9月25日**	**2024年1月12日**
11	Set up the development environment.	5 个工作日	2023年9月25日	2023年9月29日
12	Implement the client-side using HTML, CSS, JavaScript, and jQuery.	14 个工作日	2023年9月30日	2023年10月18日
13	Implement the server-side using C# and XAMPP.	14 个工作日	2023年10月3日	2023年10月20日
14	Develop algorithms for authentication, data retrieval, and data processing.	14 个工作日	2023年10月5日	2023年10月24日
15	Implement the login system for patients and doctors.	14 个工作日	2023年10月25日	2023年11月13日
16	Design and implement the SQL database for storing patient and doctor information.	14 个工作日	2023年11月3日	2023年11月22日
17	Establish a connection with the sensor for data acquisition.	14 个工作日	2023年11月17日	2023年12月6日
18	Create a user-friendly interface with a user drop-down menu.	14 个工作日	2023年12月6日	2023年12月25日
19	Develop features for doctor information, contact, appointment scheduling, online meetings, patient list management, healthy suggestions, doctor count, and patient health history tracking.	21 个工作日	2023年12月15日	2024年1月12日
20	**Testing Phase**	**21 个工作日**	**2024年1月12日**	**2024年2月9日**
21	Conduct unit testing for individual components.			
22	Perform integration testing to ensure all modules work together.			
23	Test the system's functionality, performance, and security.			
24	Identify and fix any bugs or issues.			
25	Conduct user acceptance testing with selected users.			
26	**Deployment Phase**	**21 个工作日**	**2024年2月10日**	**2024年3月8日**
27	Prepare the system for deployment.			
28	Set up hosting environment and configure servers.			
29	Deploy the system to production.			
30	Perform final testing in the live environment.			
31	Address any last-minute issues or concerns.			
32	**Documentation and Training Phase**	**21 个工作日**	**2024年3月9日**	**2024年4月5日**



14. Conclusion

The Remote Health Monitoring System (RHMS) project has successfully demonstrated its capability to revolutionize patient care by leveraging digital technologies to facilitate real-time health monitoring and communication between doctors and patients. This system, equipped with functionalities like registration, login, appointment scheduling, health record management, and live communication, has addressed critical gaps in remote healthcare management.

System Features and Benefits

1. **User-Friendly Interface**: The RHMS provides a seamless interface for users to register and login, ensuring that even those with minimal technical knowledge can easily access health services.
2. **Enhanced Doctor-Patient Interaction**: By incorporating features such as online chat and video meetings, the system has significantly improved the way doctors interact with their patients, offering convenience and reducing the need for physical visits.
3. **Efficient Appointment Management**: Doctors can manage appointments effectively, view patient health records at a glance, and make informed decisions quickly, which is crucial in medical practice.
4. **Real-Time Health Monitoring**: The ability for users to check their heart rate using devices and upload the data online has been a game-changer. This functionality not only helps in continuous monitoring but also alerts healthcare providers promptly about any critical changes in the patient's condition.
5. **Accessible Health Records**: Patients have the benefit of viewing their own health records whenever needed. This transparency helps in fostering trust and encourages patients to take an active role in managing their health.
6. **Automated Assistance**: The integration of a chatbot provides immediate answers to common queries, which enhances user experience and reduces the workload on healthcare professionals.

Impact

The RHMS has shown potential in enhancing the efficiency of healthcare delivery. It has reduced the time and cost associated with traditional healthcare processes by minimizing physical consultations and optimizing the scheduling and management of appointments. Moreover, the system has empowered patients, giving them control over their health management, and has ensured that doctors can provide timely and effective treatment.

In conclusion, the Remote Health Monitoring System has successfully demonstrated how digital innovation can be harnessed to enhance the accessibility and quality of healthcare services. It signifies a step forward in the field of telemedicine, offering a robust framework that not only meets current healthcare demands but also anticipates future needs.

15. Project Log

Overview This log details the weekly schedule and milestones for the final year project from September 4, 2023, to April 22, 2024. Weekly meetings with the supervisor are scheduled every Tuesday, with project work sessions every Wednesday and Saturday.

Weekly Breakdown with Dates

Weekly Breakdown with Dates

Week	Date (Tuesday)	Date (Wednesday)	Date (Saturday)	Activity	Notes
1	Sep 5, 2023	Sep 6, 2023	Sep 9, 2023	Initial meeting with supervisor	Discuss project scope and objectives
				Kick-off session	Setup project workspace, initial research
				Continue research	Focus on literature review
2	Sep 12, 2023	Sep 13, 2023	Sep 16, 2023	Progress update with supervisor	Present research findings
				Start developing project outline	Outline major components
				Refine project outline and start detailed planning	Create detailed task list
3	Sep 19, 2023	Sep 20, 2023	Sep 23, 2023	Review outline with supervisor	Adjust based on feedback

4	Sep 26, 2023	Sep 27, 2023	Sep 30, 2023	Progress update and feedback	Review initial results
				Implement supervisor's feedback	Modify design/Approach
				Continue development	Further coding and testing
5	Oct 3, 2023	Oct 4, 2023	Oct 7, 2023	Mid-project review	Comprehensive review of project status
				Address any critical issues	Make necessary adjustments
				Proceed with refined plans	Continue with implementation
...	...	...	...	...	...
29	Apr 16, 2024	Apr 17, 2024	Apr 20, 2024	Final meeting with supervisor	Review project journey, discuss any follow-up actions
				Project handover (if applicable)	Transfer materials, code, and documentation
				Celebration and reflection session	Recognize team effort, discuss future opportunities

Additional Notes

Ensure all logs and documentation are updated promptly after each session. Prepare questions and discussion points for each meeting with the supervisor to maximize the effectiveness of the feedback. Utilize Wednesday and Saturday sessions to address any pending issues, ensuring continuous progress on the project. This structured approach helps maintain regular progress checks and ensures that the project remains on track throughout the academic year.

16. References

SYNERTONE
http://www.synertone.net/synertone/article/index/id/161

GovHK
https://www.gov.hk/tc/residents/health/healthadvice/

17. Program listing

Home Page

Index.html

C: > xampp > htdocs > FYP web > index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Remote Health System</title>
  <link rel="stylesheet" type="text/css" href="Home.css">
</head>

<body>
  <header>
  <a href="#" class="logo"><img src="image/logo.png" alt="">Remote Health System</a>

  <nav class="navbar">
    <a href="#home">Home</a>
    <a href="#Serviecs">Serviecs</a>
    <a href="">About us</a>
  </nav>
</header>

<!-- Home Section-->
 <section class="home" id="home">
    <div class="content">
      <h3>Revolutionizing Healthcare:</h3><h3>The Rise of Remote Medical Systems</h3>
      <p>Remote Medical Systems, also known as telemedicine or telehealth systems, are transforming the healthcare landscape. The
      <a href="Register_Login/Customer.php" class="btn">Sign Up Now</a>
    </div>

    <div class="image">
      <img src="image/4.jpg" alt="">
    </div>
 </section>

<section class="Serviecs" id="Serviecs">
  <h1 class="heading"><span>Our Serviecs</span></h1>
  <div class="box-container">
    <div class="box">
      <img src="image/5.jpg" alt="" class="image">
      <div class="content">
        <img src="image/endocrine-system.png" alt="" class="img">
        <h3>Remote physiological parameter monitoring</h3>
        <p>Through sensors, smart devices, or wearable devices, the system can monitor a patient's physiological parameters suc
      </div>
    </div>
      <div class="box">
      <img src="image/6.jpg" alt="" class="image">
      <div class="content">
        <img src="image/analytics.png" alt="" class="img">
        <h3>Health data recording and analysis</h3>
        <p>Remote health control systems can record and store the patient's health data, including measurement values, medicati
      </div>
```

```
    </div>
        <div class="box">
        <img src="image/3.jpg" alt="" class="image">
        <div class="content">
          <img src="image/consultation.png" alt="" class="img">
          <h3>Remote medical consultation and diagnosis</h3>
          <p>Patients can engage in real-time or asynchronous medical consultations and diagnoses with healthcare providers through the remote healt
        </div>
    </div>
            <div class="box">
        <img src="image/7.jpg" alt="" class="image">
        <div class="content">
          <img src="image/reminder.png" alt="" class="img">
          <h3>Medication reminders and management</h3>
          <p>The remote health control system can provide medication reminders to help patients adhere to their medication schedules. Additionally,
        </div>
    </div>
            <div class="box">
        <img src="image/8.jpg" alt="" class="image">
        <div class="content">
          <img src="image/guidance.png" alt="" class="img">
          <h3>Health education and guidance</h3>
          <p>The remote health control platform can offer relevant health education and guidance to patients, including disease management, lifestyl
        </div>
    </div>
            <div class="box">
        <img src="image/9.jpg" alt="" class="image">
        <div class="content">
          <img src="image/emergency-call.png" alt="" class="img">
          <h3>Emergency calls and rescue</h3>
          <p>In emergency situations, patients can use the remote health control system to send emergency calls or distress signals to promptly rece
        </div>
    </div>
  </div>
</section>

<section class="benifit">
  <div class="box">
    <img src="image/health-insurance.png" alt="">
    <h3>Create Your Own Profile</h3>
  </div>
  <div class="box">
    <img src="image/healthcare.png" alt="">
    <h3>Upload Your Body Values</h3>
  </div>
    <div class="box">
    <img src="image/check-up.png" alt="">
    <h3>Consult A Doctor Online</h3>
  </div>
            <div class="box">
            <img src="image/stethoscope.png" alt="">
            <h3>Reserve</h3>
          </div>
        </section>

        <script src="Home.js"></script>
    </body>
    </html>
```

Home.js

```
let menu = document.querySelector('#Serviecs-bar');
let navbar = document.querySelector('.navbar');

menu.onclick = () =>{
  menu.classList.toggle('fa-times');
  navbar.classList.toggle('active');
}

window.onscroll = () =>{
  menu.classList.remove('fa-times');
  navbar.classList.remove('active');
  if(window.scrollY > 60){
    document.querySelector('#scroll-top').classList.add('active');
  }else{
    document.querySelector('#scroll-top').classList.remove('active');
  }
}

// 获取搜索按钮和输入框的引用
var searchButton = document.getElementById("searchButton");
var searchInput = document.getElementById("searchInput");

// 添加点击事件监听器
searchButton.addEventListener("click", function(event) {
  event.preventDefault(); // 阻止表单提交
  var searchText = searchInput.value;
  console.log("搜索内容：" + searchText);
  // 可以在这里执行搜索相关的操作
});

const imageLinks = document.querySelectorAll('.image-link');

imageLinks.forEach(function(link) {
  link.addEventListener('click', function(event) {
    event.preventDefault();
    window.location.href = this.getAttribute('href');
  });
});
```

Home.css

```css
@import url('https://fonts.googleapis.com/css2?family=Agbalumo&display=swap');

:root{
  --green:#306754;
}

*{
  font-family: 'Agbalumo', Regular 400;
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  outline: none;
  border: none;
  text-decoration: none;
  text-transform: capitalize;
  transition: all .2s linear;
}

*::selection{
  background:var(--green);
  color:#fff;
}
section{
  padding:2rem 9%;
}

html{
  font-size: 62.5%;
  overflow-x: hidden;
  scroll-behavior: smooth;
  scroll-padding-top: 6rem;
}

body{
  background:#f7f7f7;
}

header{
  position: fixed;
  top:0; left: 0; right:0;
  z-index: 1000;
  display: flex;
  align-items: center;
  justify-content: space-between;
  background:#fff;
  padding:2rem 9%;
  box-shadow: 0 .5rem 1rem rgba(0,0,0,.1);
}

header .logo{
  font-size: 2.5rem;
  font-weight: bolder;
  color:#666;
```

```
}

header .navbar a{
  font-size: 2rem;
  margin-left: 2rem;
  color:#666;
}
header .navbar a:hover{
  color:lightblue;
}

/*home*/
#menu-bar{
  font-size: 3rem;
  cursor: pointer;
  color:#666;
  border:.1rem solid #666;
  border-radius: .3rem;
  padding:.5rem 1.5rem;
  display: none;
}

.btn{
  display: inline-block;
  padding:.8rem 3rem;
  border:.2rem solid var(--green);
  color:var(--green);
  cursor: pointer;
  font-size: 1.7rem;
  border-radius: .5rem;
  position: relative;
  overflow: hidden;
  z-index: 0;
  margin-top: 1rem;
}
.btn::before{
  content: '';
  position: absolute;
  top:0; right: 0;
  width:0%;
  height:100%;
  background:var(--green);
  transition: .3s linear;
  z-index: -1;
}
.btn:hover::before{
  width:100%;
  left: 0;
}
.btn:hover{
  color:#fff;
}

```

```css
.home{
  display: flex;
  flex-wrap: wrap;
  grid-gap:1.5rem;
  min-height: 100vh;
  align-items: center;
  background:url(../images/home-bg.jpg) no-repeat;
  background-size: cover;
  background-position: center;
}
.home .content{
  flex:1 1 40rem;
  padding-top: 6.5rem;
}
.home .image{
  flex:1 1 40rem;
}
.home .image img{
  width:100%;
  padding:1rem;
}

@keyframes float{
  0%, 100%{
    transform: translateY(0rem);
  }
  50%{
    transform: translateY(3rem);
  }
}

.home .content h3{
  font-size: 5rem;
  color:#333;
}
.home .content p{
  font-size: 1.7rem;
  color:#666;
  padding:1rem 0;
}
/*menu*/
.heading{
  text-align: center;
  font-size: 3.5rem;
  padding:1rem;
  color:#666;
}
.heading span{
  color:var(--green);
}

.Serviecs .box-container{
  display: flex;
```

```
  flex-wrap: wrap;
  grid-gap:1.5rem;
}
.Serviecs .box-container .box{
  flex:1 1 30rem;
  position: relative;
  overflow: hidden;
  box-shadow: 0 .5rem 1rem rgba(0,0,0,.1);
  border:.1rem solid rgba(0,0,0,.3);
  cursor: pointer;
  border-radius: .5rem;
}
.Serviecs .box-container .box .image{
  height:100%;
  width:100%;
  object-fit: cover;
  position: absolute;
  top:-100%; left:0;
}
.Serviecs .box-container .box .content{
  text-align: center;
  background:#fff;
  padding:2rem;
}
.Serviecs .box-container .box .content img{
  margin:1.5rem 0;
}
.Serviecs .box-container .box .content h3{
  font-size: 2.5rem;
  color:#333;
}
.Serviecs .box-container .box .content p{
  font-size: 1.6rem;
  color:#666;
  padding:1rem 0;
}
.Serviecs .box-container .box:hover .image{
  top:0;
}
.Serviecs .box-container .box:hover .content{
  transform: translateY(100%);
}

/*benefit*/
.benifit{
  display: flex;
  flex-wrap: wrap;
  gap:1.5rem;
  padding:1rem;
}
.benifit .box{
  flex:1 1 25rem;
  padding:1rem;
    padding:1rem;
    text-align: center;
  }
  .benifit .box img{
    border-radius: 50%;
    border:0.5rem solid #fff;
    box-shadow: 0.5rem 1rem rgba(0,0,0,.1);
  }
  .benifit .box h3{
    font-size: 3rem;
    color:#333;
    padding:1rem;
  }
```

Login

Customer.php

```php
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css">
    <link rel="stylesheet" href="Customer.css">
    <title>Customer Login Page </title>
    <style>
        input[type='password'] {
            font-size: 20px; /* Increase text size for password fields */
        }
    </style>
</head>

<body>
<?php
session_start();
require_once('../conn.php');
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['signUp'])) {
    $userName = $_POST['name'];
    $Email = $_POST['email'];
    $tel = $_POST['tel'];
    $password = $_POST['pw'];

    // Get the maximum userID
    $maxIDQuery = "SELECT MAX(userID) FROM `users`";
    $result = mysqli_query($conn, $maxIDQuery);
    $row = mysqli_fetch_array($result);
    $maxID = $row[0];

    // Determine the new userID
    $newUserID = ($maxID === NULL) ? 1 : $maxID + 1;

    // Insert new user into the database
    $sql = "INSERT INTO users (userID, userName, Email, tel, password) VALUES (?, ?, ?, ?, ?)";

    // Prepare statement
    $stmt = mysqli_prepare($conn, $sql);
    // Bind parameters
    mysqli_stmt_bind_param($stmt, 'issss', $newUserID, $userName, $Email, $tel, $password);
    // Execute query
    mysqli_stmt_execute($stmt) or die(mysqli_error($conn));

    // Close statement
    mysqli_stmt_close($stmt);
    // Redirect or handle the successful registration as needed
}
?>
<?php
require_once('../conn.php'); // Make sure this path is correct

```

```php
// Check if we have a POST request
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['signIn'])) {
    $userName = $_POST['name'];
    $password = $_POST['pw'];

    // Prepare SQL to prevent SQL injection
    $sql = "SELECT * FROM users WHERE userName = ? AND password = ?";

    // Prepare the statement
    $stmt = mysqli_prepare($conn, $sql);

    // Bind parameters
    mysqli_stmt_bind_param($stmt, "ss", $userName, $password);

    // Execute the statement
    mysqli_stmt_execute($stmt);

    // Store the result so we can check if the account exists in the database.
    mysqli_stmt_store_result($stmt);

    if (mysqli_stmt_num_rows($stmt) == 1) {
        // Account exists, now we verify the password.
        // Assuming you want to start a new session with user details
        $_SESSION['userName'] = $userName;
        // Redirect to user home page
        header("Location: ../User/home.php");
        exit();
    } else {
        // Username or password is incorrect
        echo '<script>alert("Incorrect username or password!");</script>';
    }

    // Close statement
    mysqli_stmt_close($stmt);
}
?>
    <h1>Remote Health Monitoring System</h1>
    <h1>Login Page</h1>
    <div class="container" id="container">
        <div class="form-container sign-up">
            <form method="POST">
                <h1>Create Account</h1>
                <span>use your username and email for registeration</span>
                <input type="text" id="name" name="name" placeholder="Username" required>
                <input type="email" id="email" name="email" placeholder="Email" required>
                <input type="text" id="tel" name="tel" placeholder="Tel" required>
                <input type="password" name="pw" placeholder="Password" pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}"
                    title="Must contain at least one number and one uppercase and lowercase letter, and at least 8 or more characters" required>
                <button id="sign-up" name="signUp">Sign Up</button>
            </form>
        </div>
        <div class="form-container sign-in" >
            <form method="post" action="Customer.php">
                <h1>Public</h1>
                <h1>Sign In</h1>
                <span>use your username password</span>
                <input type="text"  name="name" placeholder="Username">
                <input type="password" name="pw" placeholder="Password">
                <a href="#">Forget Your Password?</a>
                <button id="sign-in" name="signIn">Sign In</button>
                <button>reset</button>
            </form>
        </div>
        <div class="toggle-container">
            <div class="toggle">
                <div class="toggle-panel toggle-left">
                    <h1>Welcome Back!</h1>
                    <p>Enter your personal details</p>
                    <button class="hidden" id="login">Sign In</button>
                </div>
                <div class="toggle-panel toggle-right">
                    <h1>Hello, Friend!</h1>
                    <p>Register with your personal details</p>
                    <button class="hidden" id="register">Sign Up</button>
                    <a href="Doctor.php"><button class="restaurant">Doctor Login</button></a>
                </div>
            </div>
        </div>
    </div>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script src="Customer.js"></script>
</body>
</html>
```

Doctor.php

```php
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css">
    <link rel="stylesheet" href="Customer.css">
    <title>Customer Login Page </title>
    <style>
        input[type='password'] {
            font-size: 20px; /* Increase text size for password fields */
        }
    </style>
</head>

<body>
<?php
session_start();
require_once('../conn.php');

if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['signUp'])) {
    // Collect and sanitize the input data
    $name = mysqli_real_escape_string($conn, $_POST['name']);
    $email = mysqli_real_escape_string($conn, $_POST['email']);
    $tel = mysqli_real_escape_string($conn, $_POST['tel']);
    $types = mysqli_real_escape_string($conn, $_POST['types']);
    $password = mysqli_real_escape_string($conn, $_POST['pw']); // 您要求不使用密码哈希

    // Retrieve the maximum doctorID and add 1 to it to create a new ID
    $result = mysqli_query($conn, "SELECT MAX(doctorID) FROM doctor");
    $row = mysqli_fetch_array($result);
    $newId = str_pad((int) $row[0] + 1, 10, "0", STR_PAD_LEFT); // 假设doctorID是一个数字，需要转换成字符串

    // SQL query to insert data into the doctor table with the new doctorID
    $sql = "INSERT INTO doctor (doctorID, name, types, tel, email, password) VALUES (?, ?, ?, ?, ?, ?)";

    // Prepare the SQL statement
    if ($stmt = mysqli_prepare($conn, $sql)) {
        // Bind variables to the prepared statement as parameters
        mysqli_stmt_bind_param($stmt, "ssssss", $newId, $name, $types, $tel, $email, $password);
        // Execute the prepared statement
        if (mysqli_stmt_execute($stmt)) {
            echo "New record created successfully with ID " . $newId;
        } else {
            echo "Error: " . mysqli_error($conn);
        }

        // Close statement
        mysqli_stmt_close($stmt);
    } else {
        echo "Error: " . mysqli_error($conn);
    }
```

```php
    // Close connection
    mysqli_close($conn);
}
?>
<?php
require_once('../conn.php'); // Make sure this path is correct

// Check if we have a POST request
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['signIn'])) {
    $userName = $_POST['name'];
    $password = $_POST['pw'];

    // Prepare SQL to prevent SQL injection
    $sql = "SELECT * FROM doctor WHERE name = ? AND password = ?";

    // Prepare the statement
    $stmt = mysqli_prepare($conn, $sql);

    // Bind parameters
    mysqli_stmt_bind_param($stmt, "ss", $userName, $password);

    // Execute the statement
    mysqli_stmt_execute($stmt);

    // Store the result so we can check if the account exists in the database.
    mysqli_stmt_store_result($stmt);

    if (mysqli_stmt_num_rows($stmt) == 1) {
        // Account exists, now we verify the password.
        // Assuming you want to start a new session with user details
        $_SESSION['name'] = $userName;
        // Redirect to user home page
        header("Location: ../Doctor/home.php");
        exit();
    } else {
        // Username or password is incorrect
        echo '<script>alert("Incorrect username or password!");</script>';
    }

    // Close statement
    mysqli_stmt_close($stmt);
}
?>

<h1>Remote Health Monitoring System</h1>
<h1>Login Page</h1>
<div class="container" id="container">
    <div class="form-container sign-up">
        <form method="POST">
            <h1>Create Account</h1>
            <span>use your username and email for registration</span>
            <input type="text" id="name" name="name" placeholder="Username" required>
```

```
            <input type="email" id="email" name="email" placeholder="Email" required>
            <input type="tel" id="tel" name="tel" placeholder="Tel" required>
            <input list="types" id="types-input" name="types" placeholder="Doctor type" required/>
            <datalist id="types">
                <option value="General medical practitioner">
                <option value="Cardiologist">
                <option value="Gynocologist">
                <option value="Pediatrician">
            </datalist>
            <input type="password" name="pw" placeholder="Password" pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}"
                    title="Must contain at least one number and one uppercase and lowercase letter, and at least 8 or more characters" required>
            <button id="sign-up" name="signUp">Sign Up</button>
        </form>
    </div>
    <div class="form-container sign-in" >
        <form method="post" action="Doctor.php">
            <h1>Doctor</h1>
            <h1>Sign In</h1>
            <span>use your username password</span>
            <input type="text"  name="name" placeholder="Username">
            <input type="password" name="pw" placeholder="Password">
            <a href="#">Forget Your Password?</a>
            <button id="sign-in" name="signIn">Sign In</button>
            <button>reset</button>
        </form>
    </div>
    <div class="toggle-container">
        <div class="toggle">
            <div class="toggle-panel toggle-left">
                <h1>Welcome Back!</h1>
                <p>Enter your personal details</p>
                <button class="hidden" id="login">Sign In</button>
            </div>
            <div class="toggle-panel toggle-right">
                <h1>Hello, Friend!</h1>
                <p>Register with your personal details</p>
                <button class="hidden" id="register">Sign Up</button>
                <a href="Customer.php"><button class="restaurant">Public Login</button></a>
            </div>
        </div>
    </div>
</div>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script src="Customer.js"></script>
</body>
</html>
```

RegisterDoctor.php

```php
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>Register</title>

    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script src="https://kit.fontawesome.com/61b76a306e.js" crossorigin="anonymous"></script>
    <script src="DoctorNext.js"></script>
    <script src="password.js"></script>
    <script src="inputValidation.js"></script>
    <link rel="stylesheet" href="index.css">
    <link rel="stylesheet" href="styles.css">
    <link rel="stylesheet" href="termsConditions.css">
    <script src="function.js"></script>
</head>
<script>
    function navigate() {
        var dropdown = document.getElementById("role");
        var selectedOption = dropdown.options[dropdown.selectedIndex].value;

        if (selectedOption !== "") {
            window.location.href = selectedOption;
        }
    }

    // Function to toggle between registration form and password form
    function toggleForms() {
        var registrationForm = document.getElementById('registrationForm');
        var passwordForm = document.getElementById('passwordForm');
        var nextButton = document.getElementById('next');

        // When the Next button is clicked
        document.getElementById('nextButton').addEventListener('click', function(event){
            event.preventDefault(); // Prevent the form from submitting

            // Hide the registration form and show the password form
            registrationForm.style.display = 'none';
            passwordForm.classList.remove('hidden');
        });
    }

    // Call the function to activate form toggling
    toggleForms();

</script
<body>
<?
require_once("../conn.php");
// Check if the form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Get the maximum doctorID from the table
```

```php
    $query = "SELECT MAX(doctorID) as maxID FROM doctor";
    $result = mysqli_query($conn, $query);
    $row = mysqli_fetch_assoc($result);
    $maxId = $row['maxID'];
    $newId = (int)$maxId + 1; // Increment the doctorID

    // Get the form data
    $name = mysqli_real_escape_string($conn, $_POST['name']);
    $types = mysqli_real_escape_string($conn, $_POST['types']);
    $tel = mysqli_real_escape_string($conn, $_POST['tel']);
    $email = mysqli_real_escape_string($conn, $_POST['email']);
    $password = mysqli_real_escape_string($conn, $_POST['password']); // Consider hashing the password

    // Prepare an insert statement
    $query = "INSERT INTO doctor (doctorID, name, types, tel, email, password) VALUES (doctorID, name, types, tel, email, password)";

    if($stmt = mysqli_prepare($conn, $query)){
        // Bind variables to the prepared statement as parameters
        mysqli_stmt_bind_param($stmt, "ississ", $newId, $name, $types, $tel, $email, $password);

        // Execute the statement
        if(mysqli_stmt_execute($stmt)){
            echo "Record created successfully.";
        } else {
            echo "ERROR: Could not execute query: $query. " . mysqli_error($conn);
        }
    } else {
        echo "ERROR: Could not prepare query: $query. " . mysqli_error($conn);
    }

    // Close statement
    mysqli_stmt_close($stmt);
}

// Close connection
mysqli_close($conn);
?>
<div class="form">
    <h1>Doctor Register</h1>

    <!-- personal information form -->
    <form id="registrationForm" method="post" action="RegisterDoctor.php">
        <div class="field username-field">
            <div class="input">
                <input type="text" name="name" id="username" pattern="[A-Z][a-z]*" required placeholder="start with an uppercase letter and contain lowercase letters">
                <label for="username">Username:</label>
            </div>
            <span class="error username-error">
            <p class="error-text"><i class="fa-solid fa-circle-exclamation" style="color: #e61405;"></i> start from uppercase and contain lowercases</p>
        </span>
        </div>
        <div class="field types-field" id="types-field">
            <label for="types">Doctor Type:</label>
```

```php
                <div class="input" name="types">
                    <input list="types" />
                    <datalist id="types">
                        <option value="General medical practitioner" />
                        <option value="Cardiologist" />
                        <option value="Gynocologist" />
                        <option value="Pediatrician" />
                    </datalist>

                </div>

            </span>
            </div>
            <div class="field email-password">
                <div class="input">
                    <input type="email" name="email" id="email" required>
                    <label for="email">Email:</label>
                </div>
                <span class="error email-error">
                <p class="error-text"> <i class="fa-solid fa-circle-exclamation" style="color: #e61405;"></i> Please enter a valid email</p>
            </span>

            </div>

            <div class="field tel-field">
                <div class="input">
                    <input type="text" name="tel" id="tel" required pattern="^852-\d{8}" value="852-">
                    <label for="tel">Tel:</label>
                </div>
                <span class="error tel-error">
                <p class="error-text"> <i class="fa-solid fa-circle-exclamation" style="color: #e61405;"></i> tel format: 852- 8 numbers.</p>
            </span>
            </div>
            <div class="field role-field">
                <div class="input">
                    User Type:
                    <select id="role" onchange="navigate()">
                        <option value="" disabled selected>Select your role</option>
                        <option value="RegisterUser.php">Public</option>
                        <option value="Doctor" selected>Doctor</option>
                    </select>
                </div>
                <span class="error role-error">
            <p class="error-text"> <i class="fa-solid fa-circle-exclamation" style="color: #e61405;"></i> Please select a role</p>
        </span>
            </div>
            already have an account? <a href="../index.html">Login</a>

            <button type="button" id="next"><i class="fa-solid fa-arrow-right"></i> Next</button>
        </form>

        <!-- password form -->
        <form id="passwordForm" class="hidden" method="post" action="RegisterDoctor.php">
```

```
        <div class="field create-password">
            <div class="input">
                <input type="password" name="password" id="password" minlength="8" required>
                <label for="password">Password:</label>
                <span class="eye" onclick="Actpassword('password')">
            <i id="password-hide" class="fa-regular fa-eye-slash"></i>
            <i id="password-show" class="fa-regular fa-eye" style="display: none;"></i>
            </span>
            </div>

            <span class="error password-error">
            <i class="bx bx-error-circle error-icon"></i>
            <p class="Starterror-text">
              <i class="fa-solid fa-circle-exclamation" style="color: #e61405;"></i> Start with an uppercase letter.
            </p>
            <p class="Minerror-text">
              <i class="fa-solid fa-circle-exclamation" style="color: #e61405;"></i> Minimum one lowercase letter and one digit.
            </p>
            <p class="Lengtherror-text">
              <i class="fa-solid fa-circle-exclamation" style="color: #e61405;"></i> Length must be between 8 to 10 characters.
            </p>
          </span>
          </div>

          <div class="field confirm-password">
            <div class="input">
                <input type="password" id="confirmPassword" required>
                <label for="confirmPassword">Confirm Password:</label>
                <span class="eye" onclick="Actpassword('confirmPassword')">
            <i id="confirmPassword-hide" class="fa-regular fa-eye-slash"></i>
            <i id="confirmPassword-show" class="fa-regular fa-eye" style="display: none;"></i>
          </span>
            </div>

            <div id="passwordError" class="error-message">Passwords do not match.</div>
            <span class="error cPassword-error">
          <p class="error-text"> <i class="fa-solid fa-circle-exclamation" style="color: #e61405;"></i> Passwords don't match</p>
        </span>
        </div>
        <input type="checkbox" required>I agree to the <a id="termsLink" href="#">terms and conditions</a>
        <br>
        <button id="back"><i class="fa-solid fa-arrow-left"></i> Back</button>
        <button type="submit" id="register">Register</button>
    </form>
</div>

<!-- Terms and conditions overlay -->
<div class="overlay">
    <div class="modal">
        <span class="modal-close">&times;</span>
        <h2>Terms and Conditions</h2>
        <p>1. Account Registration</p>
        <p>
```

```
            1.1. To use our food delivery platform, you must register an account with accurate and complete information.</p>
        <p>
            1.2. You are responsible for maintaining the confidentiality of your account credentials and for all activities that occ
        <p>
            1.3. You must promptly notify us of any unauthorized use of your account or any other security breach.</p>
        <p>2. Placing Orders</p>
        <p> 2.1. By placing an order through our platform, you agree to provide accurate and up-to-date information, including your
        <p> 2.2. You are responsible for reviewing and confirming your order details before submitting the order.</p>
        <p> 2.3. Once an order is placed, it is considered final and cannot be modified or canceled, except as permitted by our canc

        <p>3. Delivery</p>
        <p>3.1. We strive to deliver orders promptly and within the estimated delivery time. However, delivery times may vary depend
        and order volume.</p>
        <p> 3.2. You agree to provide clear and accurate delivery instructions to ensure smooth delivery of your order.</p>
        <p> 3.3. In the event of any issues or delays in delivery, we will make reasonable efforts to inform you and resolve the sit
        <p> 4. Payment</p>
        <p>4.1. Payment for orders placed through our platform must be made using the available payment methods.</p>
        <p> 4.2. All prices displayed on our platform are in the local currency and include applicable taxes and fees, unless otherw
        <p> 4.3. We reserve the right to update prices, fees, and payment methods at any time without prior notice.</p>

        <p>5. User Conduct</p>
        <p> 5.1. You agree to use our food delivery platform responsibly and in compliance with all applicable laws and regulations.
        <p>5.2. You shall not engage in any fraudulent, unlawful, or abusive activities while using our services.</p>
        <p>
            5.3. Any misuse or unauthorized use of our platform may result in the suspension or termination of your account.</p>
        <p>6. Intellectual Property</p>
        <p> 6.1. All content, trademarks, logos, and intellectual property rights on our platform are owned or licensed by us.</p>
        <p>6.2. You may not reproduce, distribute, modify, or create derivative works of any content from our platform without our p

        <p>7. Limitation of Liability</p>
        <p> 7.1. We strive to provide accurate information and a seamless experience. However, we do not guarantee the availability,
        services provided.</p>
        <p>
            7.2. In no event shall we be liable for any direct, indirect, incidental, consequential, or punitive damages arising out
            </p>
        <p>8. Modifications to Terms and Conditions
            8.1. We reserve the right to modify these terms and conditions at any time without prior notice.</p>
        <p>
            8.2. It is your responsibility to review these terms periodically for any changes.</p>
        <p>By using our food delivery platform, you acknowledge that you have read, understood, and agreed to these terms and condit
        contact our customer support.</p>
        <p>Please note that this is just a sample terms and conditions content, and it's important to consult with legal professiona
        comply with applicable laws and regulations.</p>

    </div>
</div>


</body>

</html>
```

Customer.js

```
const container = document.getElementById('container');
const registerBtn = document.getElementById('register');
const loginBtn = document.getElementById('login');

registerBtn.addEventListener('click', () => {
    container.classList.add("active");
});

loginBtn.addEventListener('click', () => {
    container.classList.remove("active");
});

$(document).ready(function() {


    $('#reset').click(function() {
      $('#username').val('');
      $('#welcome-message').empty();
    });

    $('#sign-up').click(function() {
      var name = $('#name').val();
      if (name !== '') {
        alert('You have successfully registered. Please verify your email.');
      }
    });
  });


  $(document).ready(function() {
    $('login').addClass('striped');

    $('tr').hover(
      function() {
        $(this).addClass('over');
      },
      function() {
        $(this).removeClass('over');
      }
    );
  });

```

Customer.js

```css
@import url('https://fonts.googleapis.com/css2?family=Montserrat:wght@300;400;500;600;700&display=swap');

*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Montserrat', sans-serif;
}

body{
    background-color: #c9d6ff;
    background: linear-gradient(to right, #e2e2e2, #c9d6ff);
    display: flex;
    align-items: center;
    justify-content: center;
    flex-direction: column;
    height: 100vh;
}

.container{
    background-color: #fff;
    border-radius: 30px;
    box-shadow: 0 5px 15px rgba(0, 0, 0, 0.35);
    position: relative;
    overflow: hidden;
    width: 768px;
    max-width: 100%;
    min-height: 480px;
}

.container p{
    font-size: 14px;
    line-height: 20px;
    letter-spacing: 0.3px;
    margin: 20px 0;
}

.container span{
    font-size: 12px;
}

.container a{
    color: #333;
    font-size: 13px;
    text-decoration: none;
    margin: 15px 0 10px;
}

.container button{
    background-color: #512da8;
    color: #fff;
    font-size: 12px;
    padding: 10px 45px;
```

```
    border: 1px solid transparent;
    border-radius: 8px;
    font-weight: 600;
    letter-spacing: 0.5px;
    text-transform: uppercase;
    margin-top: 10px;
    cursor: pointer;
}

.container button.hidden{
    background-color: transparent;
    border-color: #fff;
}

.container button.restaurant{
    background-color: rgb(3, 70, 3);
    border-color: #fff;
}

.container button.delivery{
    background-color: rgb(104, 4, 4);
    border-color: #fff;
}

.container form{
    background-color: #fff;
    display: flex;
    align-items: center;
    justify-content: center;
    flex-direction: column;
    padding: 0 40px;
    height: 100%;
}

.container input{
    background-color: #eee;
    border: none;
    margin: 8px 0;
    padding: 10px 15px;
    font-size: 13px;
    border-radius: 8px;
    width: 100%;
    outline: none;
}

.container button:hover {
    background-color: #80bfff;
 }

.form-container{
    position: absolute;
    top: 0;
    height: 100%;
```

```
    transition: all 0.6s ease-in-out;
}

.sign-in{
    left: 0;
    width: 50%;
    z-index: 2;
}

.container.active .sign-in{
    transform: translateX(100%);
}

.sign-up{
    left: 0;
    width: 50%;
    opacity: 0;
    z-index: 1;
}

.container.active .sign-up{
    transform: translateX(100%);
    opacity: 1;
    z-index: 5;
    animation: move 0.6s;
}

@keyframes move{
    0%, 49.99%{
        opacity: 0;
        z-index: 1;
    }
    50%, 100%{
        opacity: 1;
        z-index: 5;
    }
}

.toggle-container{
    position: absolute;
    top: 0;
    left: 50%;
    width: 50%;
    height: 100%;
    overflow: hidden;
    transition: all 0.6s ease-in-out;
    border-radius: 150px 0 0 100px;
    z-index: 1000;
}

.container.active .toggle-container{
    transform: translateX(-100%);
    border-radius: 0 150px 100px 0;
}
```

```
161
162 .toggle{
163     background-color: #512da8;
164     height: 100%;
165     background: linear-gradient(to right, #5c6bc0, #512da8);
166     color: #fff;
167     position: relative;
168     left: -100%;
169     height: 100%;
170     width: 200%;
171     transform: translateX(0);
172     transition: all 0.6s ease-in-out;
173 }
174
175 .container.active .toggle{
176     transform: translateX(50%);
177 }
178
179 .toggle-panel{
180     position: absolute;
181     width: 50%;
182     height: 100%;
183     display: flex;
184     align-items: center;
185     justify-content: center;
186     flex-direction: column;
187     padding: 0 30px;
188     text-align: center;
189     top: 0;
190     transform: translateX(0);
191     transition: all 0.6s ease-in-out;
192 }
193
194 .toggle-left{
195     transform: translateX(-200%);
196 }
197
198 .container.active .toggle-left{
199     transform: translateX(0);
200 }
201
202 .toggle-right{
203     right: 0;
204     transform: translateX(0);
205 }
206
207 .container.active .toggle-right{
208     transform: translateX(200%);
209 }
```

Meeting Room

Meeting Room.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>Document</title>

</head>

<body>
    <script>
        var script = document.createElement("script");
        script.type = "text/javascript";

        script.addEventListener("load", function(event) {
            const config = {
                name: "",
                meetingId: "fyp",
                apiKey: "34d073b0-96b3-44df-8a26-7d2450faf7e3",

                containerId: null,

                micEnabled: true,
                webcamEnabled: true,
                participantCanToggleSelfWebcam: true,
                participantCanToggleSelfMic: true,

                chatEnabled: true,
                raiseHandEnabled: true,
                screenShareEnabled: true,
                whiteboardEnabled: true,

                joinScreen: {
                    visible: true,
                },

                recording: {
                    enabled: true,
                    webhookUrl: "https://www.videosdk.live/callback",
                    autoStart: false,
                    theme: "DARK", // DARK || LIGHT || DEFAULT

                    layout: {
                        type: "SIDEBAR",
                        priority: "PIN",
                        gridSize: 3,
                    },
                },

                permissions: {
                    pin: true,
                    toggleRecording: true,
                    askToJoin: false,
                    toggleParticipantWebcam: false,
                    toggleParticipantMic: false,
                    toggleParticipantScreenshare: false,
                    removeParticipant: true,
                    endMeeting: true,
                    drawOnWhiteboard: true,
                    toggleWhiteboard: true,
                },
```

```
                leftScreen: {
                    rejoinButtonEnabled: true,
                },
            };

            const meeting = new VideoSDKMeeting();
            meeting.init(config);
        });

        script.src =
            "https://sdk.videosdk.live/rtc-js-prebuilt/0.3.31/rtc-js-
            prebuilt.js";
        document.getElementsByTagName("head")[0].appendChild(script);

    </script>
</body>

</html>

```

Chat Bot

ChatBot.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Chatbot in JavaScript | CodingNepal</title>
    <link rel="stylesheet" href="ChatBot.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,wght,FILL,GRAD@48,400,0,0" />
    <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Material+Symbols+Rounded:opsz,wght,FILL,GRAD@48,400,1,0" />
    <script src="script.js" defer></script>
  </head>
  <body>
    <h1>Remote Health Monitoring System Chat Bot</h1>
    <button class="chatbot-toggler">
      <span class="material-symbols-rounded">mode_comment</span>
      <span class="material-symbols-outlined">close</span>
    </button>
    <div class="chatbot">
      <header>
        <h2>Chatbot</h2>
        <span class="close-btn material-symbols-outlined">close</span>
      </header>
      <ul class="chatbox">
        <li class="chat incoming">
          <span class="material-symbols-outlined">smart_toy</span>
          <p>Hi there 👋<br>How can I help you today?</p>
        </li>
      </ul>
      <div class="chat-input">
        <textarea placeholder="Enter a message..." spellcheck="false" required></textarea>
        <span id="send-btn" class="material-symbols-rounded">send</span>
      </div>
    </div>

  </body>
</html>
```

ChatBot.css

```css
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&display=swap');
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Poppins", sans-serif;
}
body {
  background: #E3F2FD;
}
.chatbot-toggler {
  position: fixed;
  bottom: 30px;
  right: 35px;
  outline: none;
  border: none;
  height: 50px;
  width: 50px;
  display: flex;
  cursor: pointer;
  align-items: center;
  justify-content: center;
  border-radius: 50%;
  background: #724ae8;
  transition: all 0.2s ease;
}
body.show-chatbot .chatbot-toggler {
  transform: rotate(90deg);
}
.chatbot-toggler span {
  color: #fff;
  position: absolute;
}
.chatbot-toggler span:last-child,
body.show-chatbot .chatbot-toggler span:first-child  {
  opacity: 0;
}
body.show-chatbot .chatbot-toggler span:last-child {
  opacity: 1;
}
.chatbot {
  position: fixed;
  right: 35px;
  bottom: 90px;
  width: 420px;
  background: #fff;
  border-radius: 15px;
  overflow: hidden;
  opacity: 0;
  pointer-events: none;
  transform: scale(0.5);
  transform-origin: bottom right;
  box-shadow: 0 0 128px 0 rgba(0,0,0,0.1),
              0 32px 64px -48px rgba(0,0,0,0.5);
  transition: all 0.1s ease;
}
body.show-chatbot .chatbot {
  opacity: 1;
  pointer-events: auto;
  transform: scale(1);
}
```

```css
.chatbot header {
  padding: 16px 0;
  position: relative;
  text-align: center;
  color: #fff;
  background: #724ae8;
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);
}
.chatbot header span {
  position: absolute;
  right: 15px;
  top: 50%;
  display: none;
  cursor: pointer;
  transform: translateY(-50%);
}
header h2 {
  font-size: 1.4rem;
}
.chatbot .chatbox {
  overflow-y: auto;
  height: 510px;
  padding: 30px 20px 100px;
}
.chatbot :where(.chatbox, textarea)::-webkit-scrollbar {
  width: 6px;
}
.chatbot :where(.chatbox, textarea)::-webkit-scrollbar-track {
  background: #fff;
  border-radius: 25px;
}
.chatbot :where(.chatbox, textarea)::-webkit-scrollbar-thumb {
  background: #ccc;
  border-radius: 25px;
}
.chatbox .chat {
  display: flex;
  list-style: none;
}
.chatbox .outgoing {
  margin: 20px 0;
  justify-content: flex-end;
}
.chatbox .incoming span {
  width: 32px;
  height: 32px;
  color: #fff;
  cursor: default;
  text-align: center;
  line-height: 32px;
  align-self: flex-end;
  background: #724ae8;
  border-radius: 4px;
  margin: 0 10px 7px 0;
}
```

```
}
.chatbox .chat p {
  white-space: pre-wrap;
  padding: 12px 16px;
  border-radius: 10px 10px 0 10px;
  max-width: 75%;
  color: #fff;
  font-size: 0.95rem;
  background: #724ae8;
}
.chatbox .incoming p {
  border-radius: 10px 10px 10px 0;
}
.chatbox .chat p.error {
  color: #721c24;
  background: #f8d7da;
}
.chatbox .incoming p {
  color: #000;
  background: #f2f2f2;
}
.chatbot .chat-input {
  display: flex;
  gap: 5px;
  position: absolute;
  bottom: 0;
  width: 100%;
  background: #fff;
  padding: 3px 20px;
  border-top: 1px solid #ddd;
}
.chat-input textarea {
  height: 55px;
  width: 100%;
  border: none;
  outline: none;
  resize: none;
  max-height: 180px;
  padding: 15px 15px 15px 0;
  font-size: 0.95rem;
}
.chat-input span {
  align-self: flex-end;
  color: #724ae8;
  cursor: pointer;
  height: 55px;
  display: flex;
  align-items: center;
  visibility: hidden;
  font-size: 1.35rem;
}
.chat-input textarea:valid ~ span {
  visibility: visible;
}
```

```
@media (max-width: 490px) {
  .chatbot-toggler {
    right: 20px;
    bottom: 20px;
  }
  .chatbot {
    right: 0;
    bottom: 0;
    height: 100%;
    border-radius: 0;
    width: 100%;
  }
  .chatbot .chatbox {
    height: 90%;
    padding: 25px 15px 100px;
  }
  .chatbot .chat-input {
    padding: 5px 15px;
  }
  .chatbot header span {
    display: block;
  }
}
```

Script.js

```
const chatbotToggler = document.querySelector(".chatbot-toggler");
const closeBtn = document.querySelector(".close-btn");
const chatbox = document.querySelector(".chatbox");
const chatInput = document.querySelector(".chat-input textarea");
const sendChatBtn = document.querySelector(".chat-input span");

let userMessage = null;
const API_KEY = "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";
const inputInitHeight = chatInput.scrollHeight;

const createChatLi = (message, className) => {
    const chatLi = document.createElement("li");
    chatLi.classList.add("chat", `${className}`);
    let chatContent = className === "outgoing" ? `<p></p>` : `<span
    class="material-symbols-outlined">smart_toy</span><p></p>`;
    chatLi.innerHTML = chatContent;
    chatLi.querySelector("p").textContent = message;
    return chatLi;
}

const generateResponse = (chatElement) => {
    const API_URL = "https://api.openai.com/v1/chat/completions";
    const messageElement = chatElement.querySelector("p");

    const requestOptions = {
        method: "POST",
        headers: {
            "Content-Type": "application/json",
            "Authorization": `Bearer ${API_KEY}`
        },
        body: JSON.stringify({
            model: "gpt-3.5-turbo",
            messages: [{role: "user", content: userMessage}],
        })
    }

    fetch(API_URL, requestOptions).then(res => res.json()).then(data => {
        messageElement.textContent = data.choices[0].message.content.trim();
    }).catch(() => {
        messageElement.classList.add("error");
        messageElement.textContent = "Oops! Something went wrong. Please try
        again.";
    }).finally(() => chatbox.scrollTo(0, chatbox.scrollHeight));
}

const handleChat = () => {
    userMessage = chatInput.value.trim();
    if(!userMessage) return;

    chatInput.value = "";
    chatInput.style.height = `${inputInitHeight}px`;

    chatbox.appendChild(createChatLi(userMessage, "outgoing"));
    chatbox.scrollTo(0, chatbox.scrollHeight);

    setTimeout(() => {
        const incomingChatLi = createChatLi("Thinking...", "incoming");
        chatbox.appendChild(incomingChatLi);
        chatbox.scrollTo(0, chatbox.scrollHeight);
        generateResponse(incomingChatLi);
    }, 600);
}
```

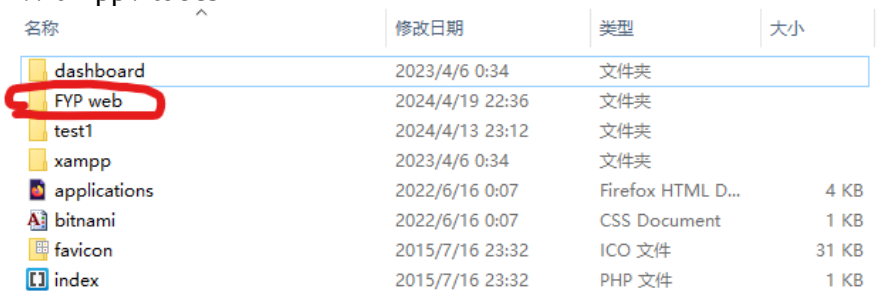
```
61
62 ▼ chatInput.addEventListener("input", () => {
63       chatInput.style.height = `${inputInitHeight}px`;
64       chatInput.style.height = `${chatInput.scrollHeight}px`;
65   });
66
67 ▼ chatInput.addEventListener("keydown", (e) => {
68 ▼     if(e.key === "Enter" && !e.shiftKey && window.innerWidth > 800) {
69           e.preventDefault();
70           handleChat();
71       }
72   });
73
74   sendChatBtn.addEventListener("click", handleChat);
75   closeBtn.addEventListener("click", () => document.body.classList.remove("show-
     chatbot"));
76   chatbotToggler.addEventListener("click", () =>
     document.body.classList.toggle("show-chatbot"));
```

18. User guide and installation guide

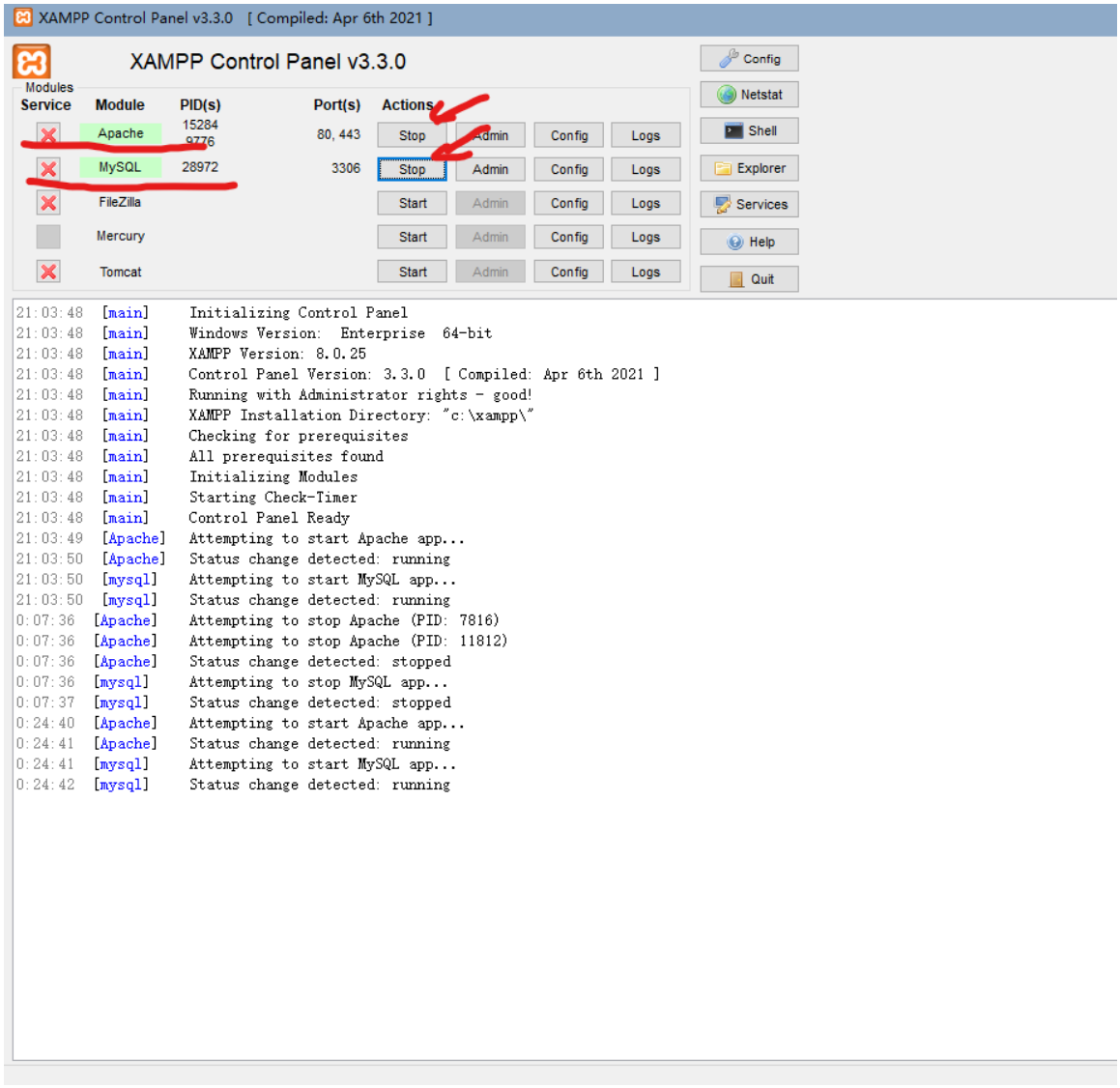
Step 1: Download and zip the file

Step 2: Put the zipped file into the path

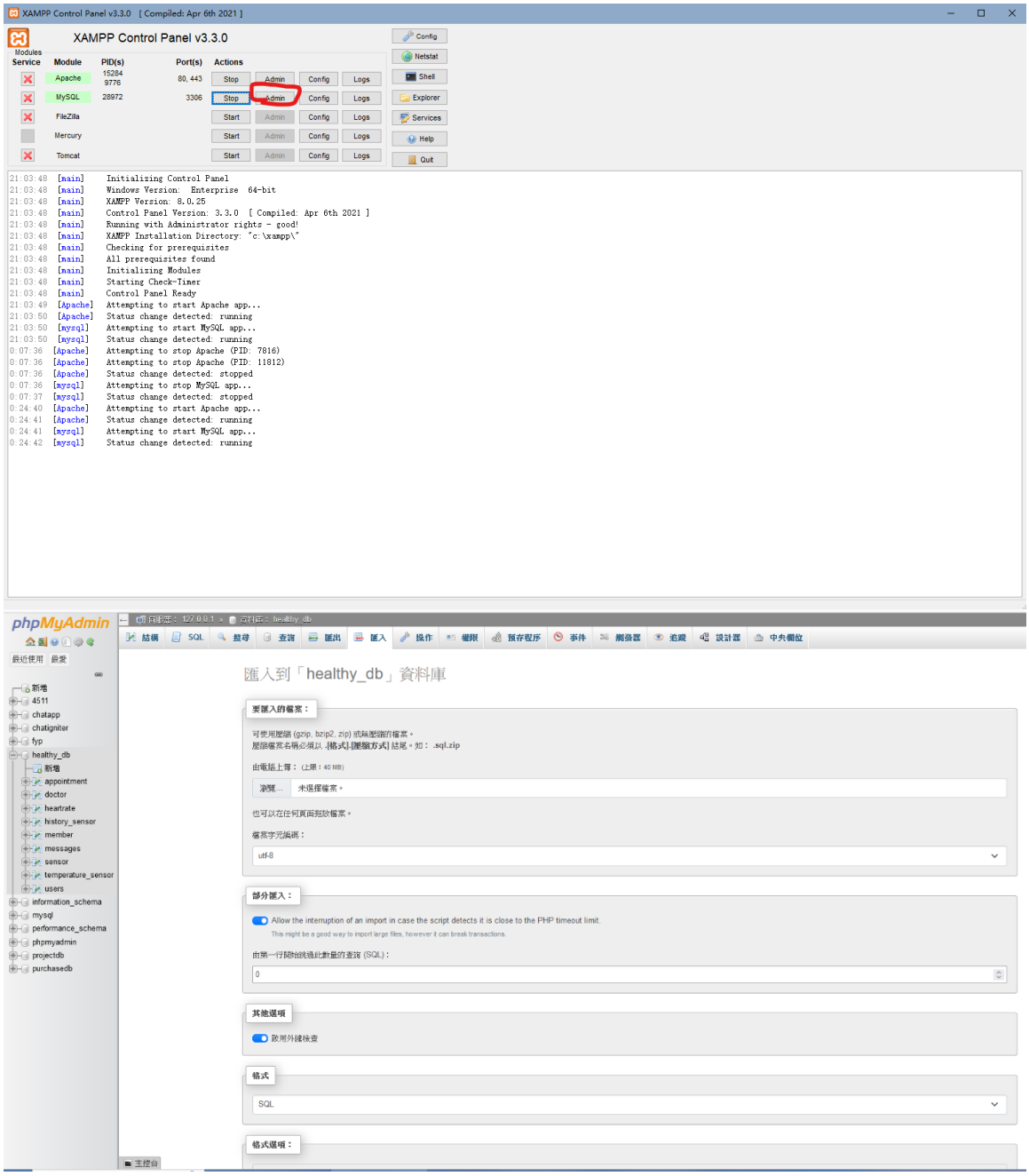
X:\xampp\htdocs

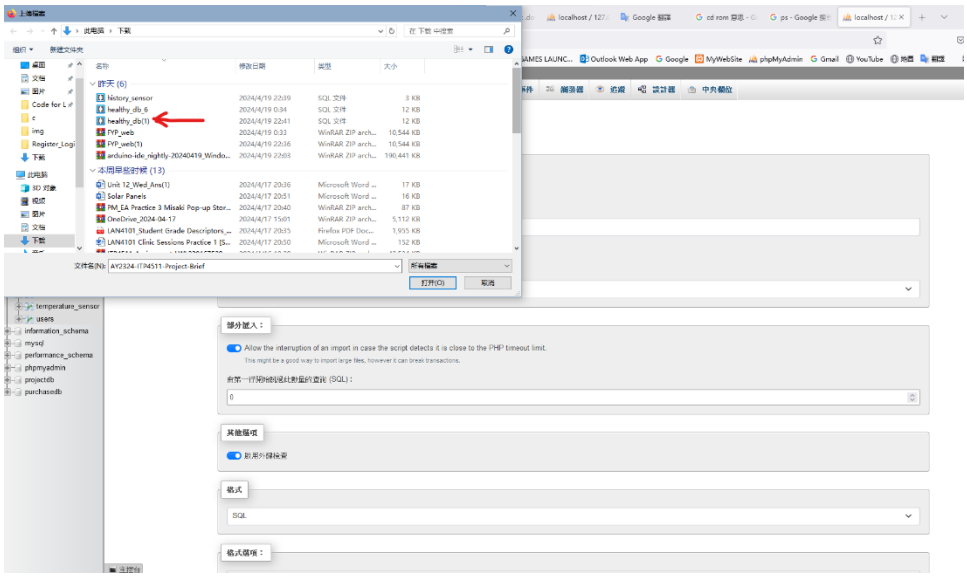


Step 3: Open the Xampp and Turn on the Apache and MySQL

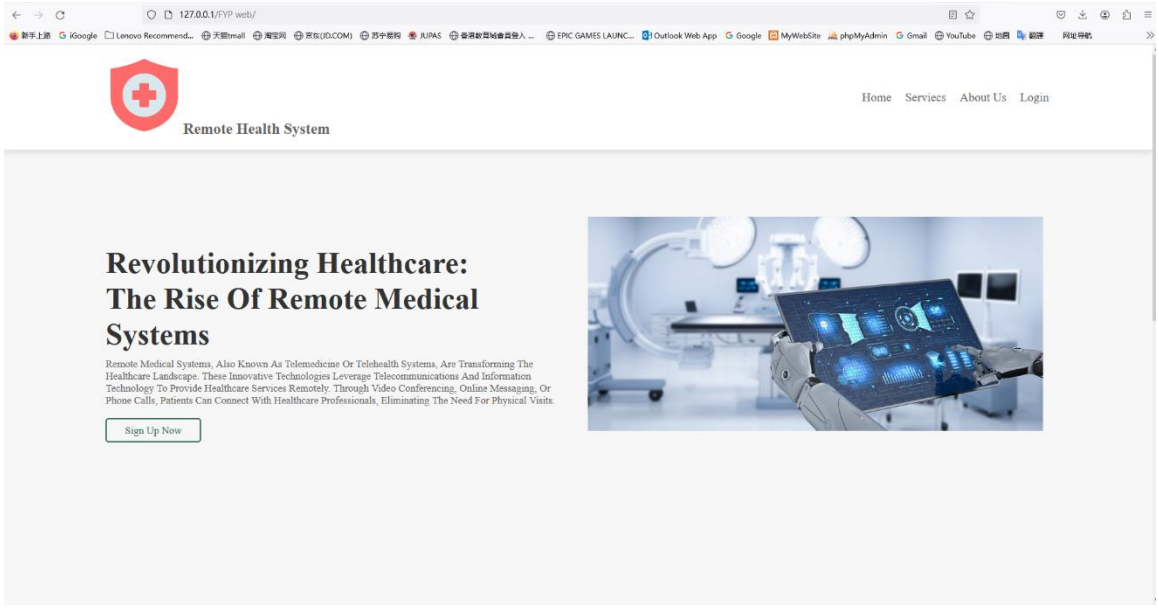


Step4: Open the Admin and insert the mysql file

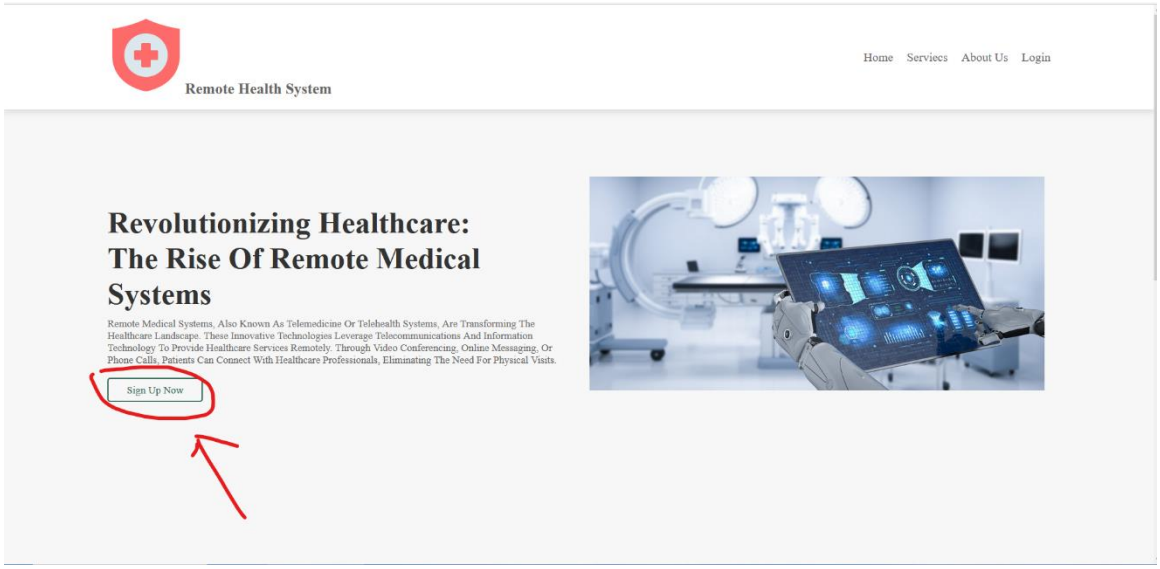




Step 5: Go to the browser and paste the link: http://127.0.0.1/FYP web/

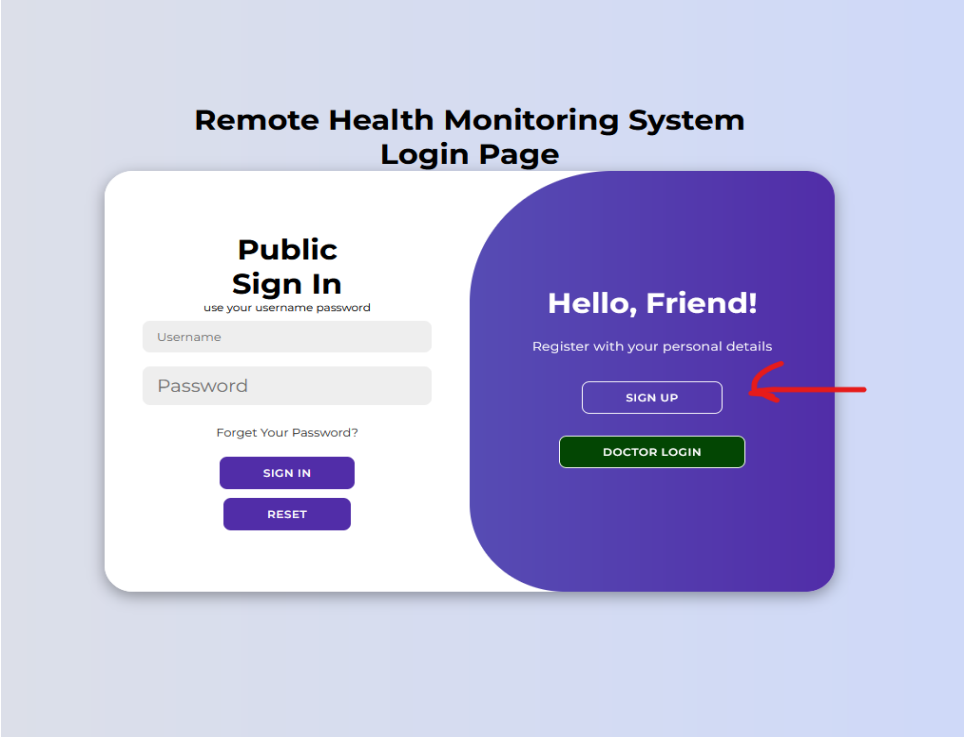


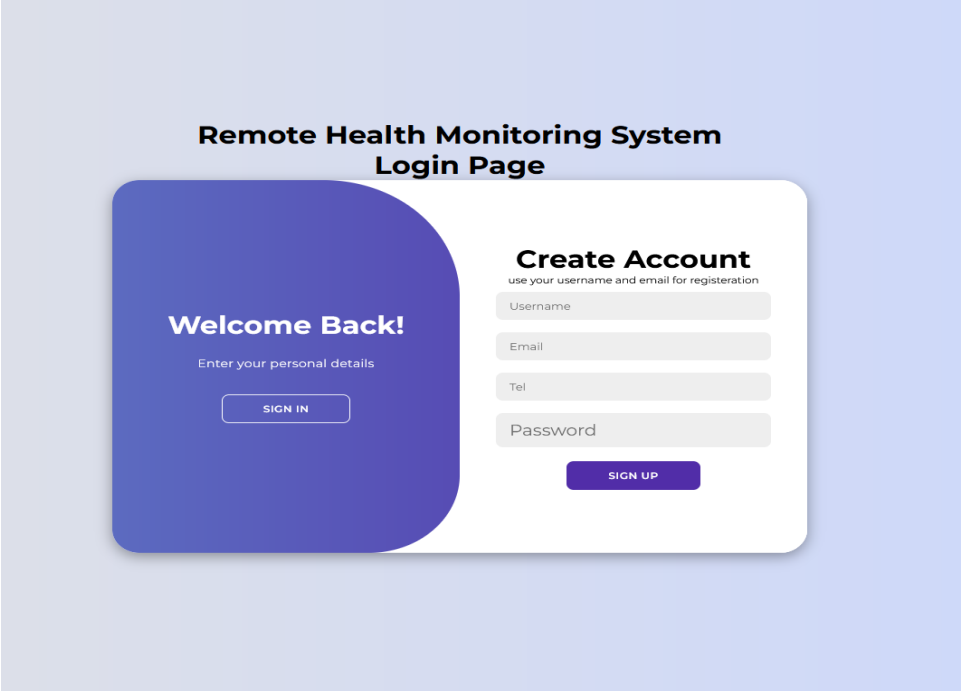
Step6: Click the button ‘Sign Up New’ to sign up

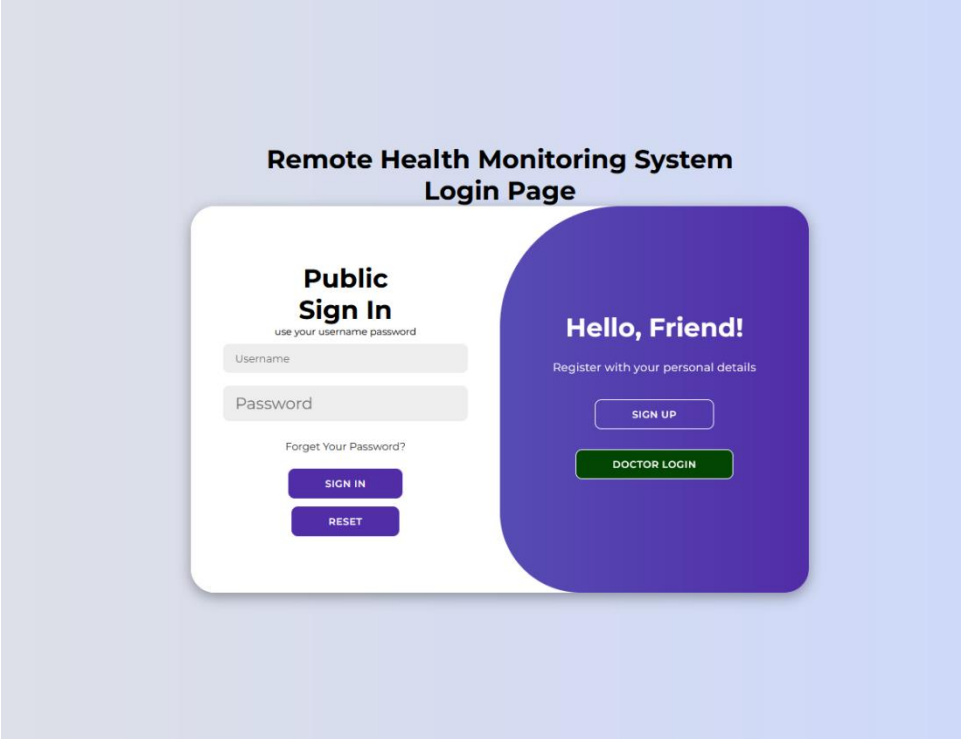


Choose your character. If you are a public user follow step 7, otherwise follow step 12.

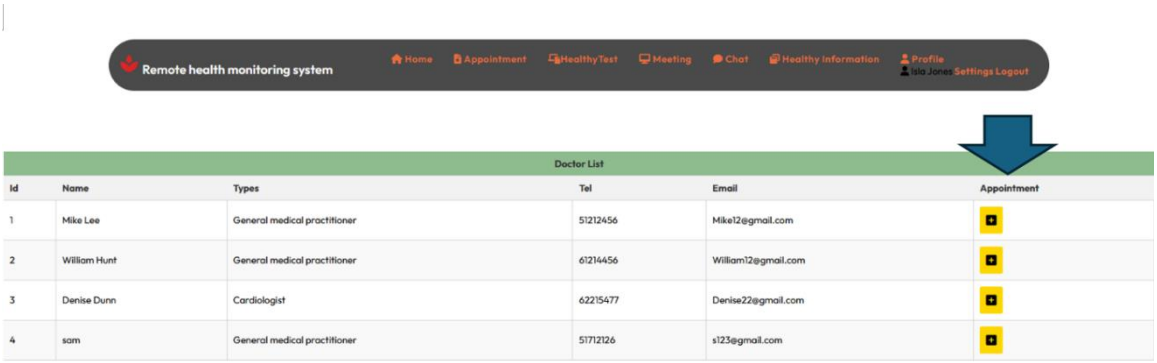
Step7: Login or sign up a new account (For Public)



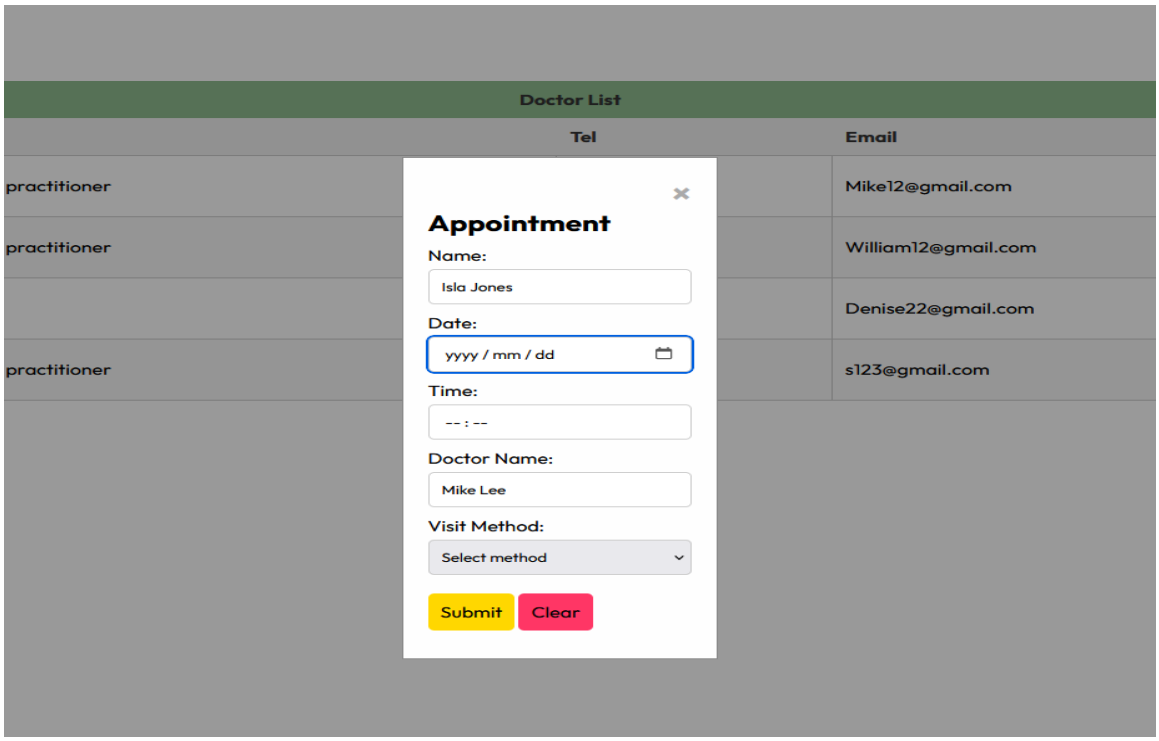



Remote Health Monitoring System
Login Page
Public
Sign In
use your username password
Username
Password
Forget Your Password?
SIGN IN
RESET
Hello, Friend!
Register with your personal details
SIGN UP
DOCTOR LOGIN

Step8: Make Appointment (For Public)

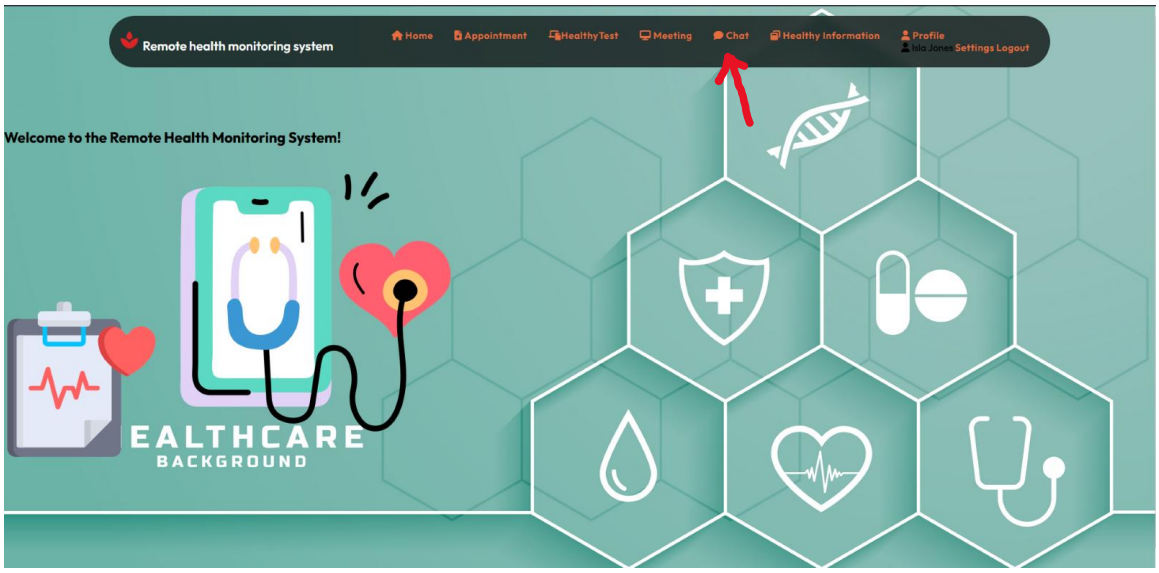


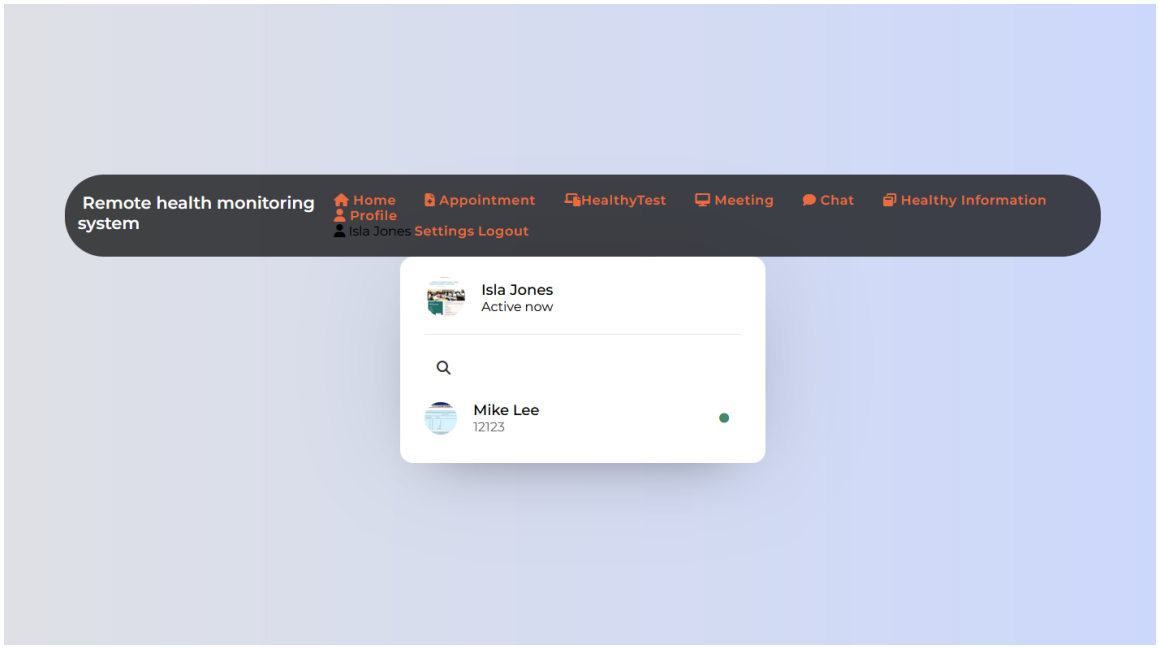
1. Choose the doctor.
2. Click the related row appointment button to open the form

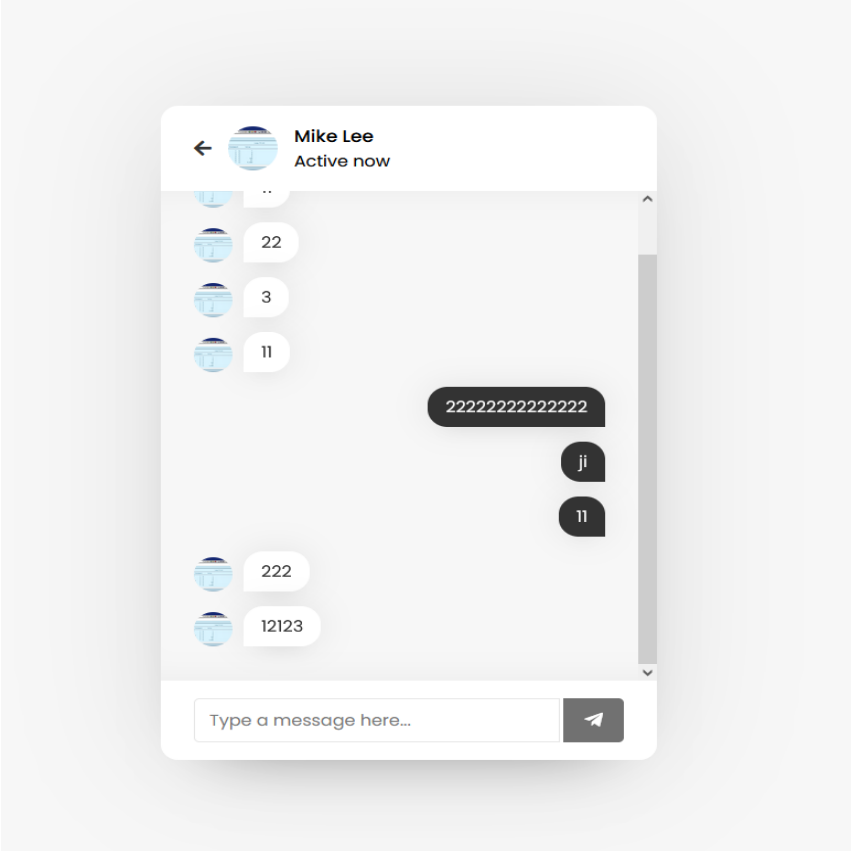


3. Fill the form
4. Click submit button

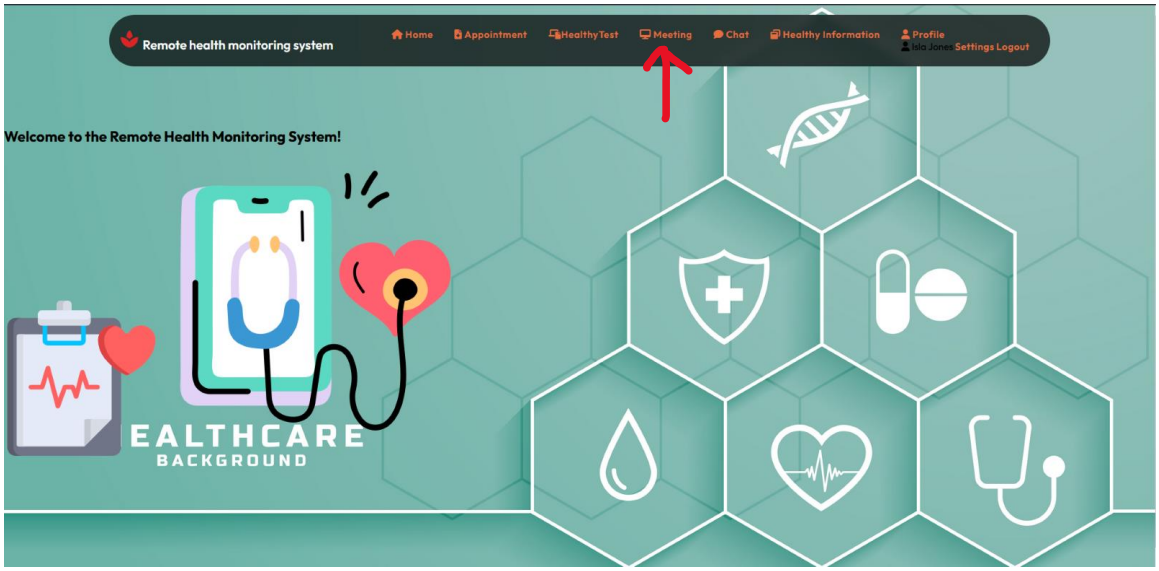
Step9: Click the button chat to communicate with doctor (Public)

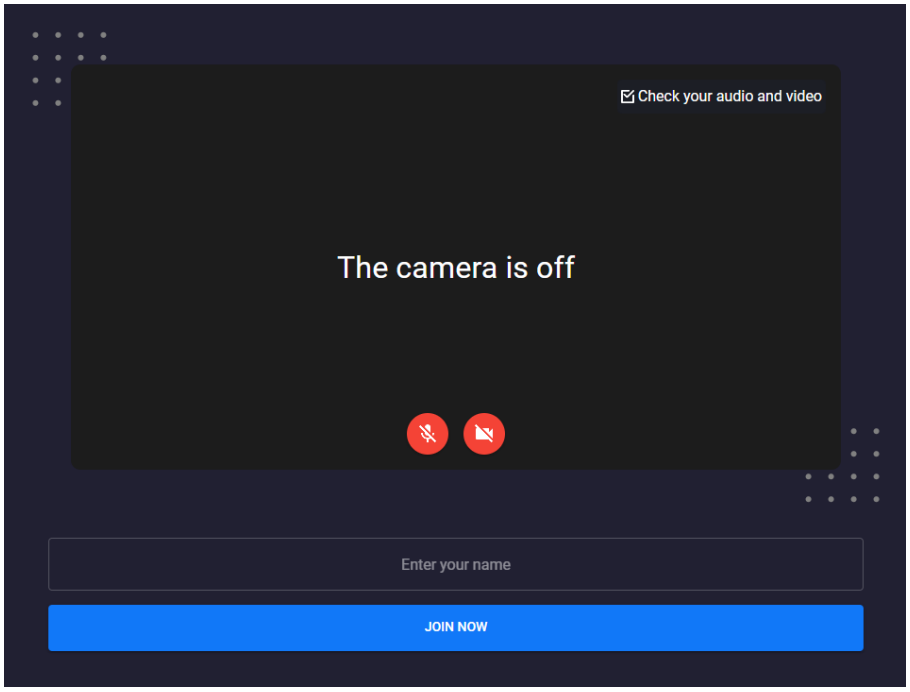




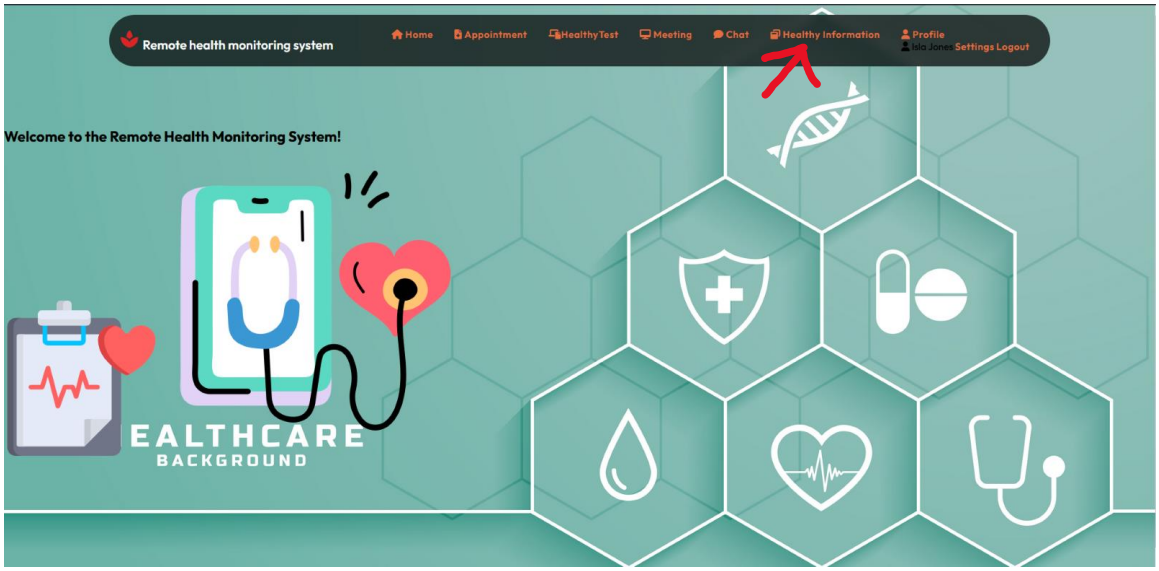

Mike Lee
Active now
22
3
11
22222222222222
ji
11
222
12123
Type a message here...

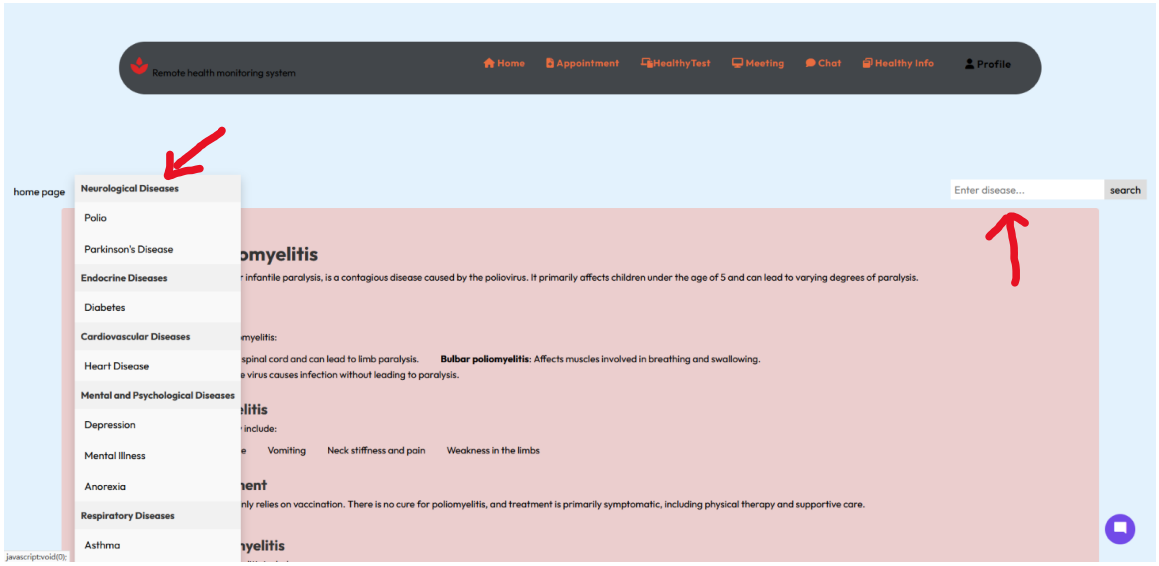
Step10: Click the button meeting to communicate with doctor (Public)

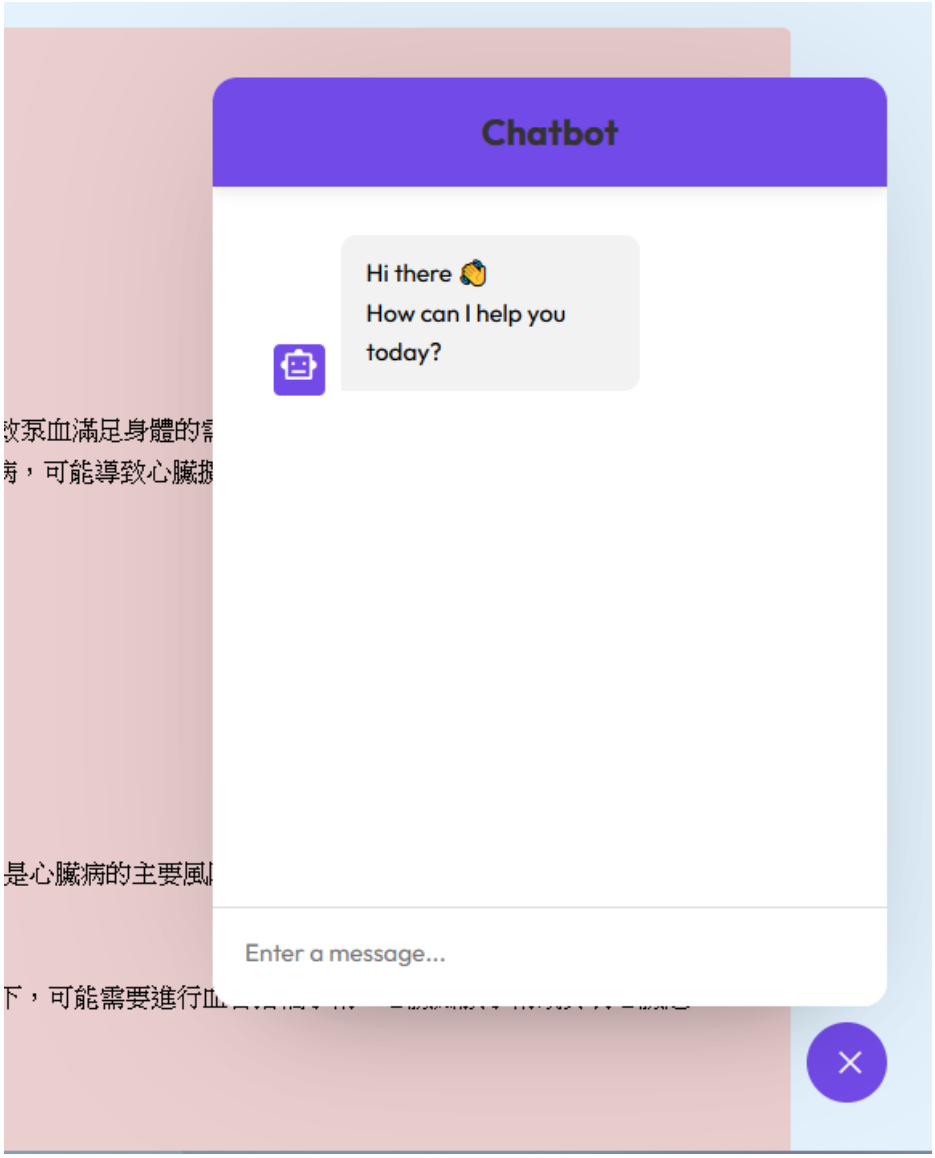




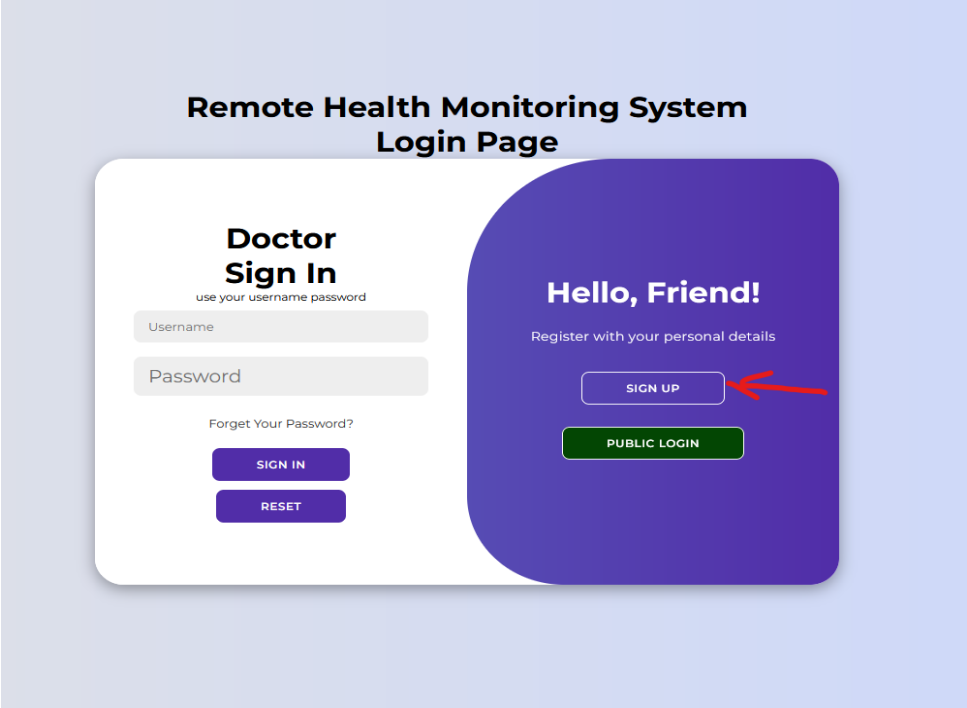
Step11: Click the Healthy Information to Search the information and use the chat bot

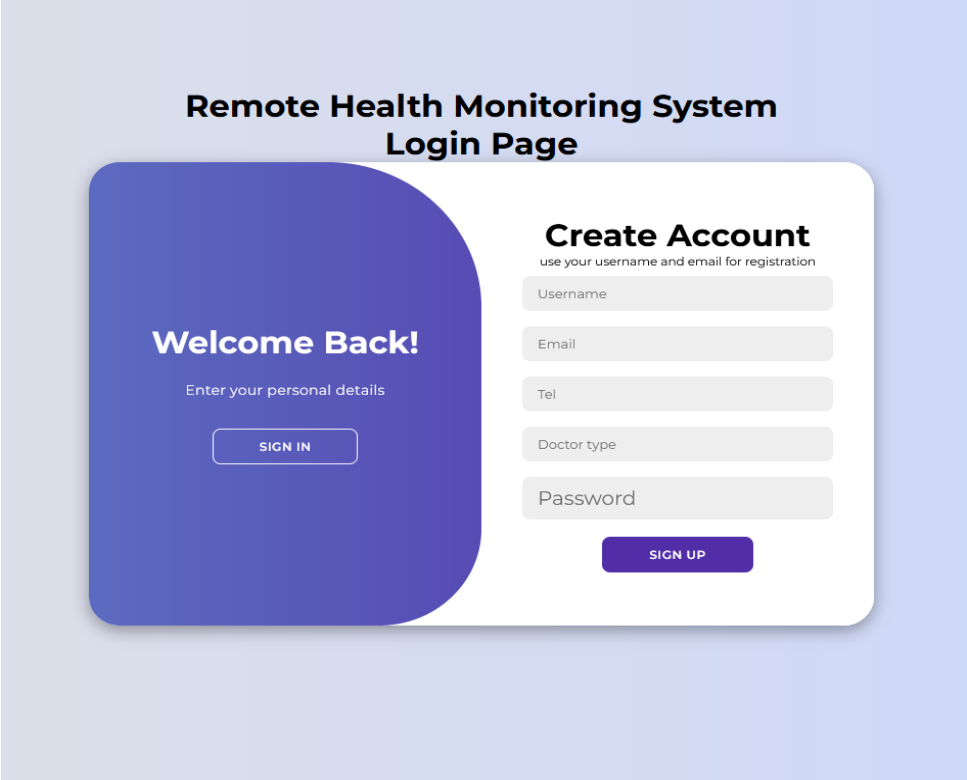


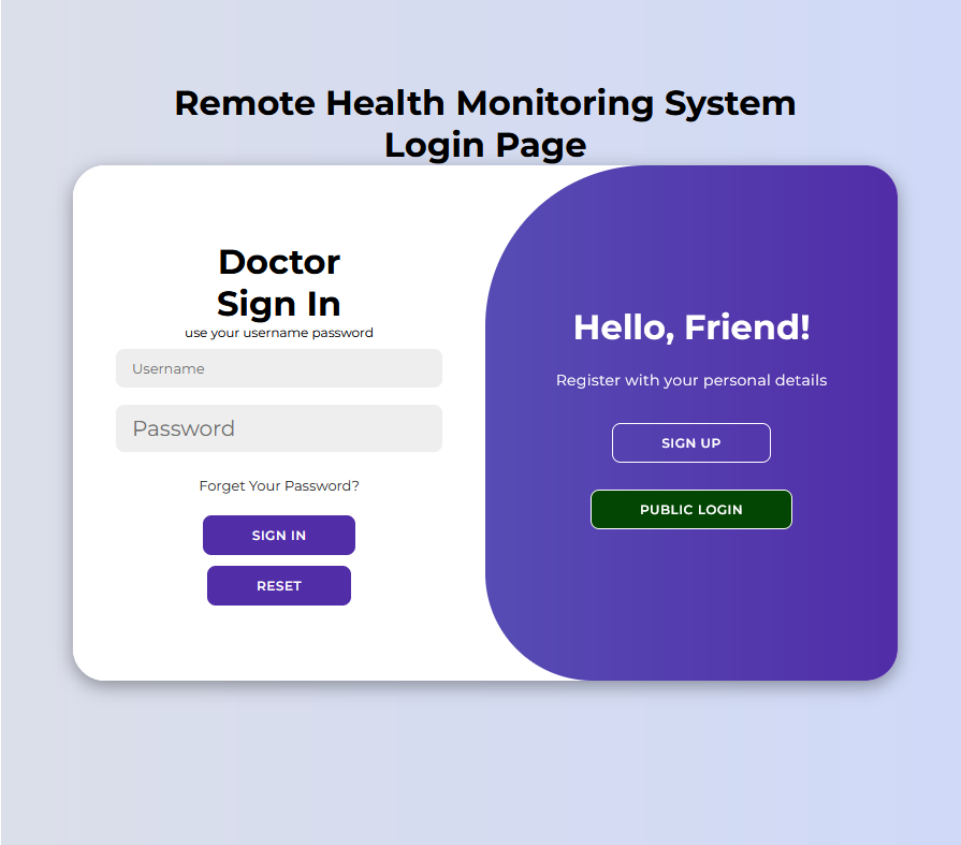



Chatbot
Hi there 👋
How can I help you today?
Enter a message...

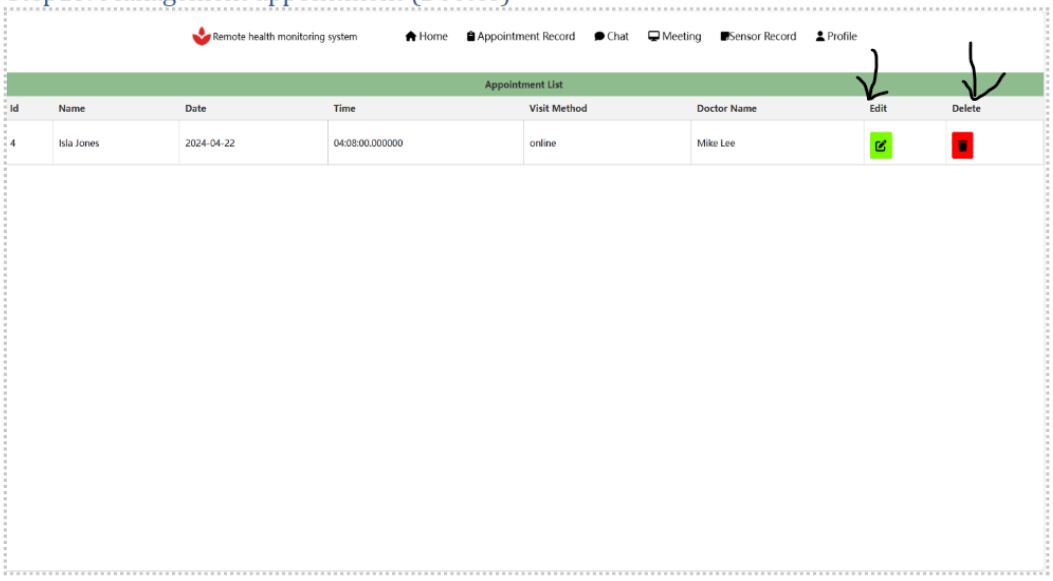
Step12: Login or sign up a new account (For Doctor)



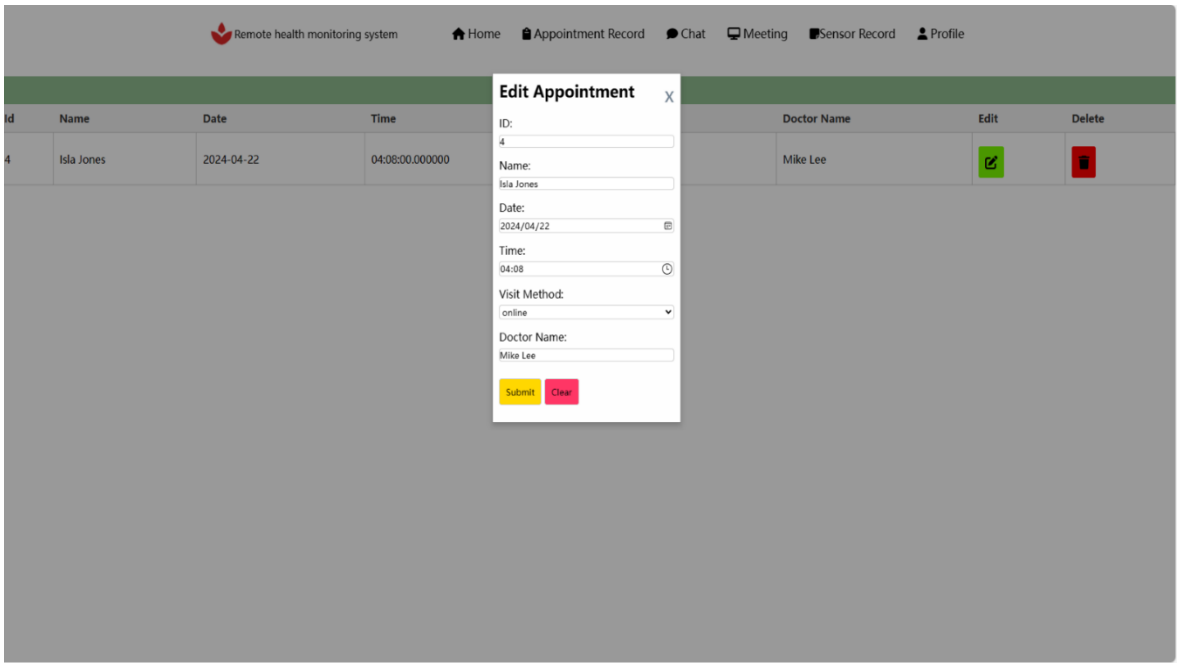



Remote Health Monitoring System
Login Page
Doctor
Sign In
use your username password
Username
Password
Forget Your Password?
SIGN IN
RESET
Hello, Friend!
Register with your personal details
SIGN UP
PUBLIC LOGIN

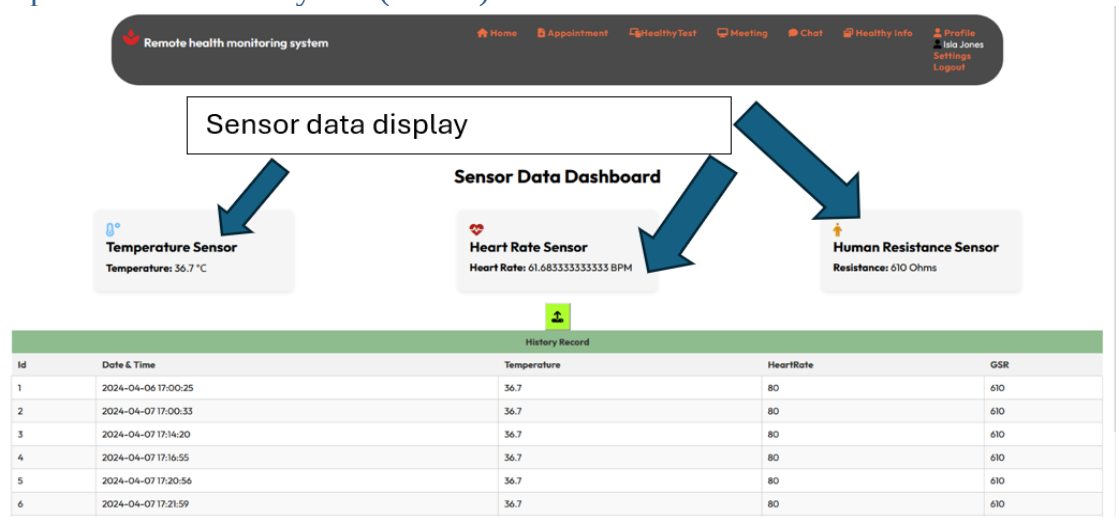
Step13: Management appointment (Doctor)



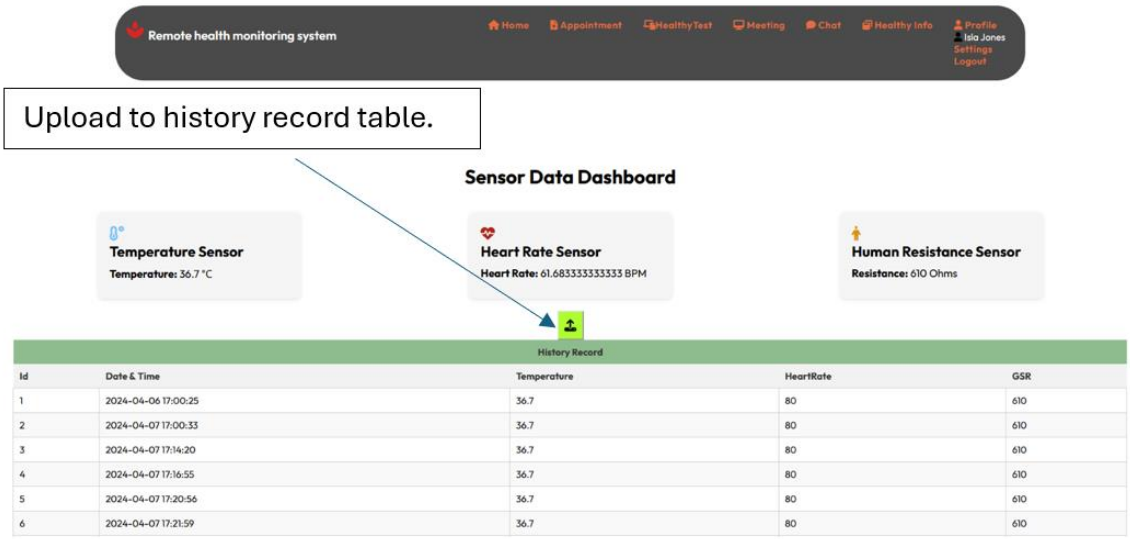
Doctor can edit the related row record or delete



Step14: Sensor Healthy test (Public)

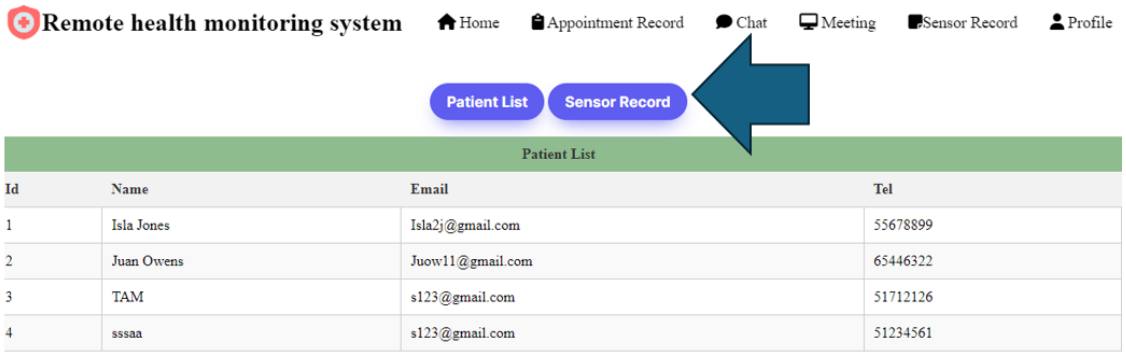


Id	Date & Time	Temperature	HeartRate	GSR
1	2024-04-06 17:00:25	36.7	80	610
2	2024-04-07 17:00:33	36.7	80	610
3	2024-04-07 17:14:20	36.7	80	610
4	2024-04-07 17:16:55	36.7	80	610
5	2024-04-07 17:20:56	36.7	80	610
6	2024-04-07 17:21:59	36.7	80	610

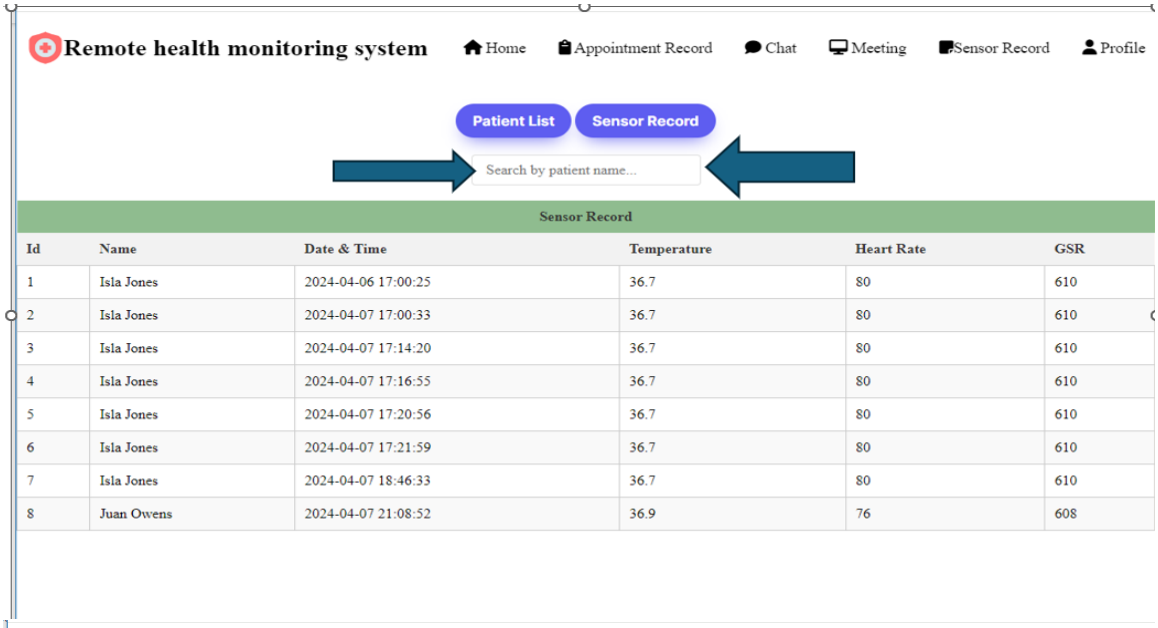


Id	Date & Time	Temperature	HeartRate	GSR
1	2024-04-06 17:00:25	36.7	80	610
2	2024-04-07 17:00:33	36.7	80	610
3	2024-04-07 17:14:20	36.7	80	610
4	2024-04-07 17:16:55	36.7	80	610
5	2024-04-07 17:20:56	36.7	80	610
6	2024-04-07 17:21:59	36.7	80	610

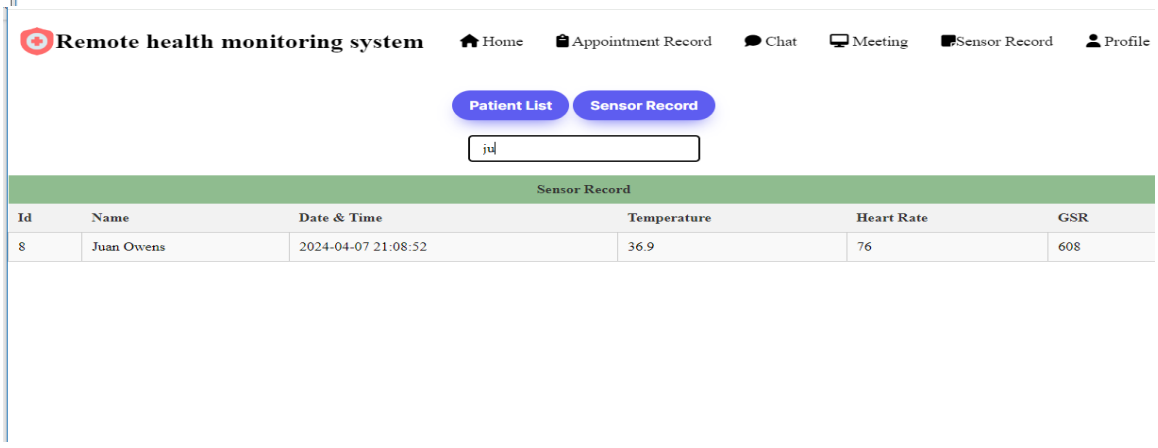
Step15: View Sensor Record (Doctor)



Patient List			
Id	Name	Email	Tel
1	Isla Jones	Isla2j@gmail.com	55678899
2	Juan Owens	Juow11@gmail.com	65446322
3	TAM	s123@gmail.com	51712126
4	sssaa	s123@gmail.com	51234561



Sensor Record					
Id	Name	Date & Time	Temperature	Heart Rate	GSR
1	Isla Jones	2024-04-06 17:00:25	36.7	80	610
2	Isla Jones	2024-04-07 17:00:33	36.7	80	610
3	Isla Jones	2024-04-07 17:14:20	36.7	80	610
4	Isla Jones	2024-04-07 17:16:55	36.7	80	610
5	Isla Jones	2024-04-07 17:20:56	36.7	80	610
6	Isla Jones	2024-04-07 17:21:59	36.7	80	610
7	Isla Jones	2024-04-07 18:46:33	36.7	80	610
8	Juan Owens	2024-04-07 21:08:52	36.9	76	608



Sensor Record					
Id	Name	Date & Time	Temperature	Heart Rate	GSR
8	Juan Owens	2024-04-07 21:08:52	36.9	76	608