



SECP3204: Software Engineering WBL

System Design Descriptions (SDD)

Student Health and Wellness Management
System

Version 1.0

21th June 2023

Faculty of Computing

Prepared by: Software

Table of Contents

| | | | |
|----------|---|--|----------------|
| 3 | System Architectural Design | | 3 - 7 |
| | 3.1 | Architectural Style and Rationale | 3 - 4 |
| | 3.2 | Component Model | 5 - 7 |
| 4 | Detailed Description of Components | | 8 - 55 |
| | 4.1 | Complete Package Diagram | 8 |
| | 4.2 | Detailed Description | 8 - 11 |
| | | 4.2.1 P001: <Authentication Module> Subsystem | 11 - 15 |
| | | 4.2.2 P002: <Account Management> Subsystem | 16 - 27 |
| | | 4.2.3 P003: <Patient Queue Management> Subsystem | 28 - 31 |
| | | 4.2.4 P004: <Health Tips> Subsystem | 32 - 36 |
| | | 4.2.5 P005: <Mental Health Support> Subsystem | 37 - 44 |
| | | 4.2.6 P006: <Dental Health Appointment > Subsystem | 45 - 51 |
| | | 4.2.7 P007: <Feedback and Reviews> Subsystem | 52 - 55 |
| 5 | Data Design | | 56 - 59 |
| | 5.1 | Data Description | 56 |
| | 5.2 | Data Dictionary | 56 - 59 |
| 6 | User Interface Design | | 60 - 88 |
| | 6.1 | Overview of User Interface | 60 - 61 |
| | 6.2 | Screen Images | |
| | | 6.2.1 Authentication System Interface | 62 - 63 |
| | | 6.2.2 Feedback and Reviews System Interface | 64 |
| | | 6.2.3 Patient Queue Manage System Interface | 65 - 66 |
| | | 6.2.4 Account Management System Interface | 67 - 74 |
| | | 6.2.5 Health Tips System Interface | 75 - 77 |

| | | | | |
|--|--|-------|--|---------|
| | | 6.2.6 | Mental Health Support System Interface | 78 - 83 |
| | | 6.2.7 | Dental Health Appointment System Interface | 84 - 88 |

3. System Architectural Design

3.1 Architecture Style and Rationale

For the development of the Student Health and Wellness Management app that is connected to the university clinic called PKU UTM in the University of Technology Malaysia (UTM), an appropriate architecture style to consider is a client-server architecture.

The client-server architecture is a widely used and effective approach for applications that require communication and data exchange between multiple entities, such as doctors, administrators, and students. This architecture consists of two main components: the client, which represents the user interface and interactions, and the server, which handles the business logic, data storage, and communication with various clients.

In this context, the client component of the architecture would encompass the mobile or web-based interface of the Student Wellness and Health Management app. This client application would provide functionalities for signing up, logging in, managing accounts, uploading health information, accessing health tips, setting up and managing appointments, giving and viewing feedback, and other relevant use cases outlined in the provided list.

The server component of the architecture would be responsible for processing and storing the data, managing user accounts and access rights, handling appointment scheduling and management, and facilitating communication between different users. It would also integrate with the existing systems of PKU UTM and the university clinic, ensuring seamless information exchange and coordination.

To implement this architecture, modern web technologies such as RESTful APIs (Representational State Transfer) can be utilized for communication between the client and server. The client application can be developed using frameworks such as ReactJS for web or

React Native for mobile platforms, while the server-side can employ frameworks like Node.js or Django to handle the business logic and data management.

Overall, a client-server architecture would provide a scalable and modular solution for the Student Wellness and Health Management app, allowing efficient communication and collaboration between doctors, administrators, and students from the KTDI hostel. It would enable streamlined processes for signing up, managing accounts, uploading health information, setting up appointments, accessing health tips, and more, contributing to the enhancement of student wellness and health management within UTM.

3.2 Component Model

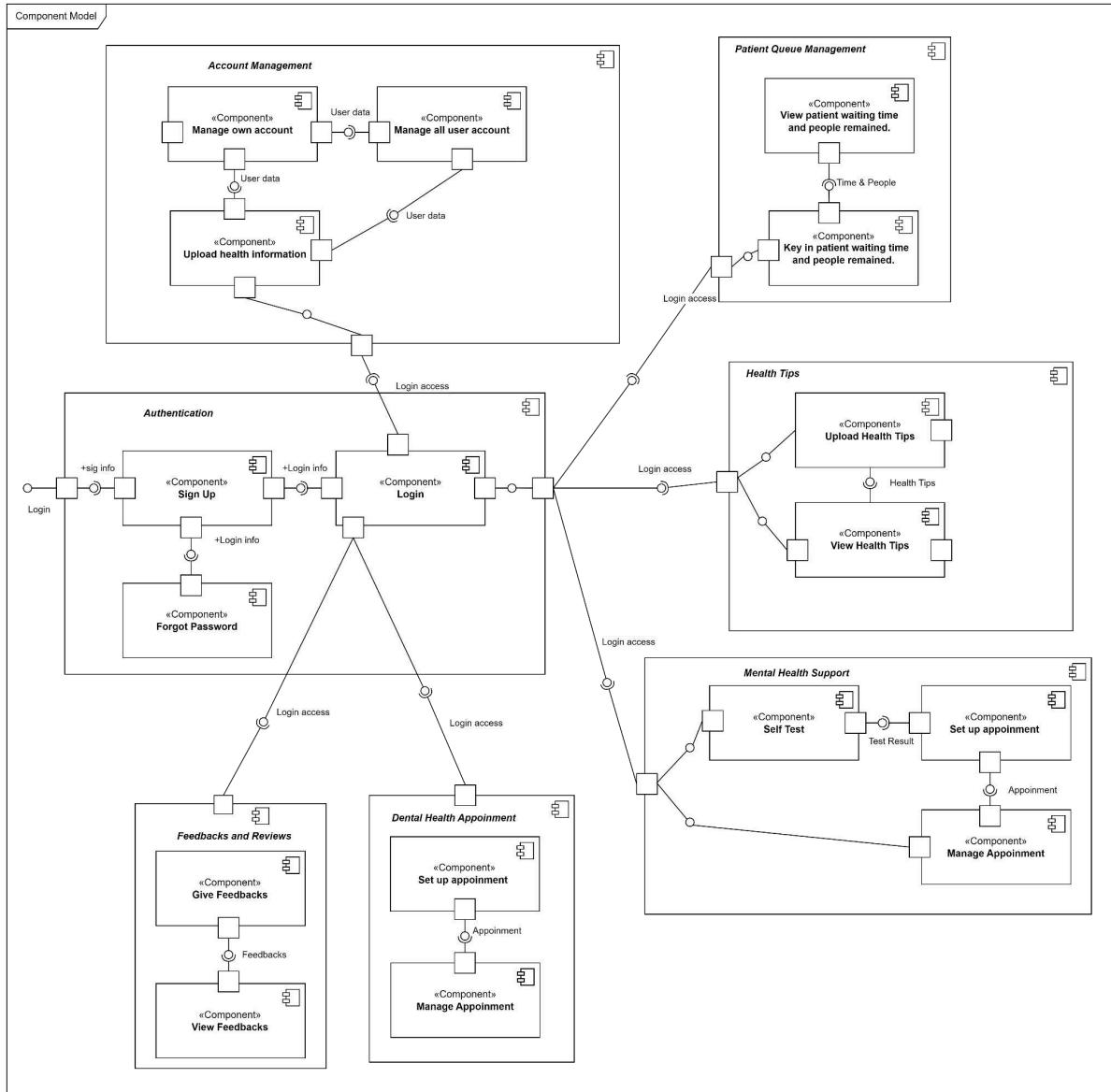


Figure 3.1: Component Model Diagram

The component model of the system encompasses various subsystems, each serving a specific purpose. The "Account Management" subsystem focuses on user accounts and includes components such as "Manage Own Account," "Manage all user accounts," and "Upload health information." These components enable users to modify their account details, allow administrators to handle multiple accounts, and facilitate the sharing of health-related data.

In the "Authentication" subsystem, secure access to the system is ensured through components like "Sign Up," "Login," and "Forgot Password." Users can create new accounts using the "Sign Up" component, while the "Login" component verifies credentials for access. Additionally, the "Forgot Password" component assists users in recovering their forgotten passwords.

Managing patient queues is the primary objective of the "Patient Queue Management" subsystem. It comprises components such as "View patient waiting time and people remained" and "Key in patient waiting time and people remained." These components enable users to monitor the waiting time and remaining patients, while authorized personnel can update the queue information.

The "Health Tips" subsystem aims to provide users with valuable health-related information. It includes components like "Upload Health Tips" and "View Health Tips." Authorized individuals can contribute informative content using the "Upload Health Tips" component, and users can access and read these tips through the "View Health Tips" component.

Addressing mental health concerns, the "Mental Health Support" subsystem consists of components like "Self Test," "Set up Appointment," and "Manage Appointment." Users can undergo mental health assessments using the "Self Test" component, which then informs the "Set up Appointment" component for scheduling further support. The "Manage Appointment" component allows users to handle their scheduled appointments, including rescheduling or canceling them.

Feedback and reviews play a crucial role in the "Feedbacks and Reviews" subsystem. Its components, "Give Feedbacks" and "View Feedbacks," enable users to submit their feedback and read reviews shared by others. This facilitates continuous improvement and provides valuable insights for users.

Additionally, the "Dental Health Support" subsystem caters specifically to dental care. It consists of components like "Set up Appointment" and "Manage Appointment." Users can

schedule dental appointments through the "Set up Appointment" component and manage their appointments using the "Manage Appointment" component.

It's worth noting that authentication, facilitated by the "Login" component in the "Authentication" subsystem, is essential for accessing several other subsystems. This ensures secure access and personalization throughout the system, enhancing user experiences.

4. Detailed Description of Components

4.1 Complete Package Diagram

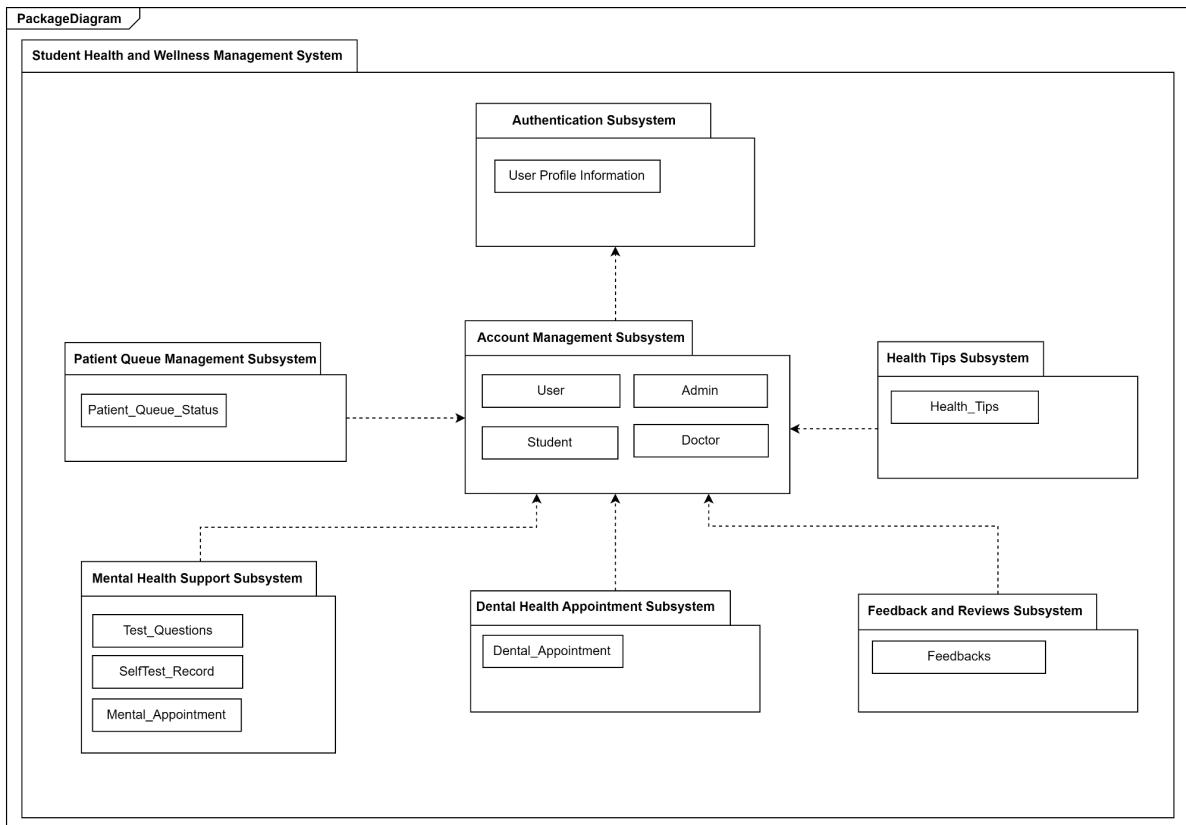


Figure 4.1: Complete Package Diagram

The diagram above shows every component's classes of the Student Health and Wellness Management System. Overall, it consists of 7 subsystems. The details of each subsystem is covered in the following section.

4.2 Detailed Description

| | | |
|------|----------------------------|--|
| P001 | <Authentication> Subsystem | |
| SD01 | Sign Up | It allows the student, doctor and administrator to sign up as a user for the application-based system. |

| | | |
|------|-----------------|---|
| SD02 | Login | It allows the student, doctor and administrator to login the application. |
| SD03 | Forgot Password | It allows the student, doctor and administrator to reset their passwords. |

| | | |
|------|--------------------------------|---|
| P002 | <Account Management> Subsystem | |
| SD04 | Manage Own Account | It allows the student, doctor and administrator to manage their accounts. |
| SD05 | Manage All Users Account | It allows the administrator to manage all users accounts. |
| SD06 | Upload Health Information | It allows the student, doctor and administrator to upload their health information including demographic information and lab results. |

| | | |
|------|---|---|
| P003 | <Patient Queue Management> Subsystem | |
| SD07 | Key in Patient Estimated Waiting Time and people remained | It allows the administrator to key in the estimated patient waiting time and people remaining (amount of pending patients) in PKU. |
| SD08 | View Patient Waiting Time and people remained | It allows the student and doctor to view the estimated patient waiting time and people remaining (amount of pending patients) in PKU. |

| | |
|------|-------------------------|
| P004 | <Health Tips> Subsystem |
|------|-------------------------|

| | | |
|------|--------------------|---|
| SD09 | Upload Health Tips | It allows the administrator to upload the health tips in the system. |
| SD10 | View Health Tips | It allows the student, doctor and administrator to view the health tips in the application. |

| | | |
|------|-----------------------------------|---|
| P005 | <Mental Health Support> Subsystem | |
| SD11 | Self-test | It allows the student to have a mental health self-test (DASS). |
| SD12 | Set Up Mental Appointment | It allows the student who scores high during the mental health self-test (DASS) to set up an appointment with the doctor of PKU in the application. |
| SD13 | Manage Mental Appointment | It allows the doctor and administrator to manage the appointment applied by the student. |

| | | |
|------|--|---|
| P006 | <Dental Health Appointment > Subsystem | |
| SD14 | Set Up Dental Appointment | It allows the student to set up an appointment with the doctor of PKU in the application. |
| SD15 | Manage Dental Appointment | It allows the doctor and administrator to manage the appointment applied by the student. |

| | | |
|------|----------------------------------|--|
| P007 | <Feedback and Reviews> Subsystem | |
| SD16 | Give Feedbacks | It allows the student and doctor to give feedback. |

| | | |
|------|----------------|---|
| SD17 | View Feedbacks | It allows the administrator to view feedback given by the student and doctor. |
|------|----------------|---|

4.2.1 P001: <Authentication> Subsystem

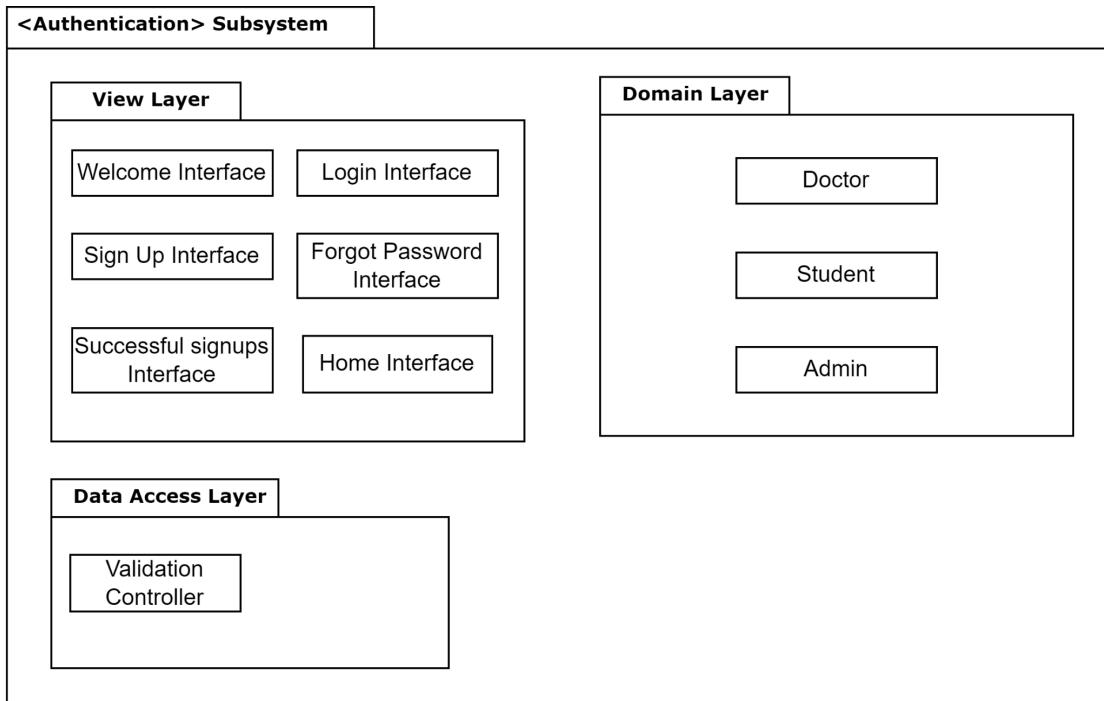


Figure 4.2.1.1: P001: <Authentication> Subsystem

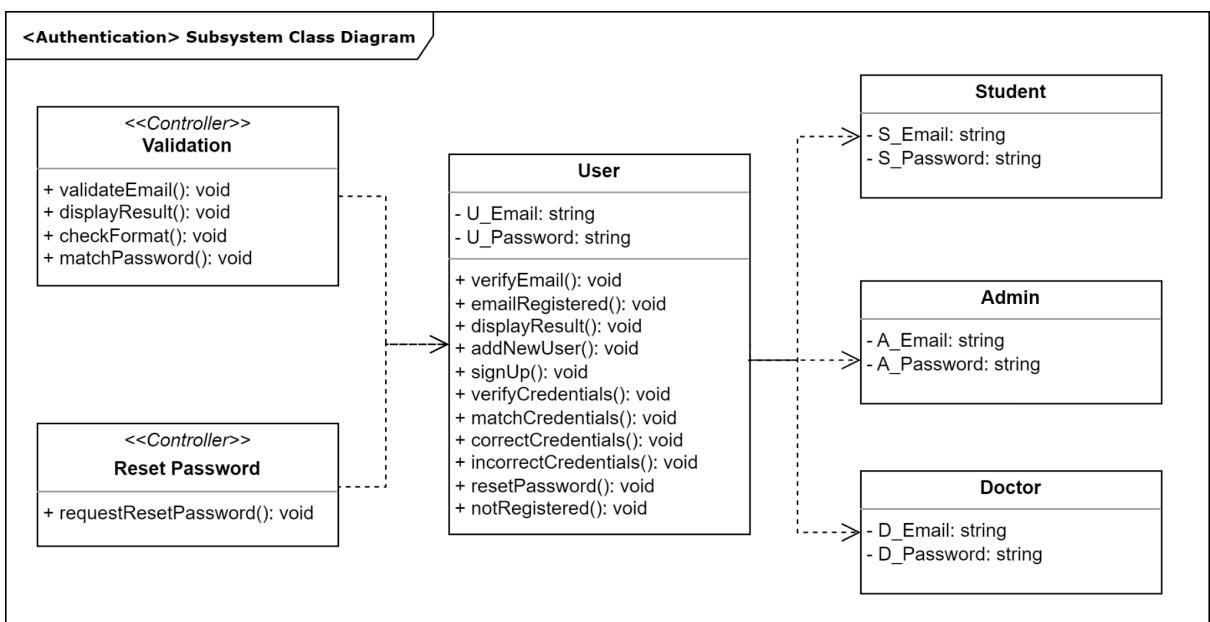


Figure 4.2.1.2: Class Diagram of P001: <Authentication> Subsystem

| | |
|--------------------|--|
| Entity Name | User (Student, Admin and Doctor) |
| Method Name | verifyEmail() |
| Input | email address of user |
| Output | System saves the user input |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. User input email address 3. System saves the input 4. End |

| | |
|--------------------|---|
| Entity Name | User (Student, Admin and Doctor) |
| Method Name | displayResult() |
| Input | user input |
| Output | System saves the result of finding a match for the email address in the database. |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. System starts to find a match for the user input in database 3. System saves the result 4. End |

| | |
|--------------------|---|
| Entity Name | User (Student, Admin and Doctor) |
| Method Name | addNewUser() |
| Input | email address and password of new user |
| Output | A new account is created |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. A new user input email address and password 3. The user submits their input 4. System saves the input into database 5. A new account is created 6. End |

| | |
|--------------------|--|
| Entity Name | User (Student, Admin and Doctor) |
| Method Name | signUp() |
| Input | Clicking on “Sign Up” button |
| Output | Successful SignUps Interface |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. The user clicks “Sign Up” after entering email address and password 3. The user will be directed to Successful Signups Interface 4. End |

| | |
|--------------------|--|
| Entity Name | User (Student, Admin and Doctor) |
| Method Name | verifyCredentials() |
| Input | Email address and password of user |
| Output | System saves the user input |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. The user clicks “Login” after entering email address and password 3. System saves the user input 4. End |

| | |
|--------------------|---|
| Entity Name | User (Student, Admin and Doctor) |
| Method Name | matchCredentials() |
| Input | user input |
| Output | System finds a match in database |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. System finds a match of user input in database 3. System saves the result 4. End |

| | |
|--------------------|---|
| Entity Name | User (Student, Admin and Doctor) |
| Method Name | correctCredentials() |
| Input | result of finding a match |
| Output | Home interface |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. System has found a match in database 3. The user will be directed to home interface 4. End |

| | |
|--------------------|---|
| Entity Name | User (Student, Admin and Doctor) |
| Method Name | incorrectCredentials() |
| Input | result of finding a match |
| Output | result |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. System did not find a match in database 3. System displays result 4. End |

| | |
|--------------------|--|
| Entity Name | User (Student, Admin and Doctor) |
| Method Name | resetPassword() |
| Input | request of password reset |
| Output | password reset link |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. User requests for password reset 3. System sends password reset link to user's email address 4. End |

| | |
|--------------------|---|
| Entity Name | User (Student, Admin and Doctor) |
| Method Name | notRegistered() |
| Input | email address |
| Output | result |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. User enters his registered email address 3. System did not find a match of email in database 4. System displays result 5. End |

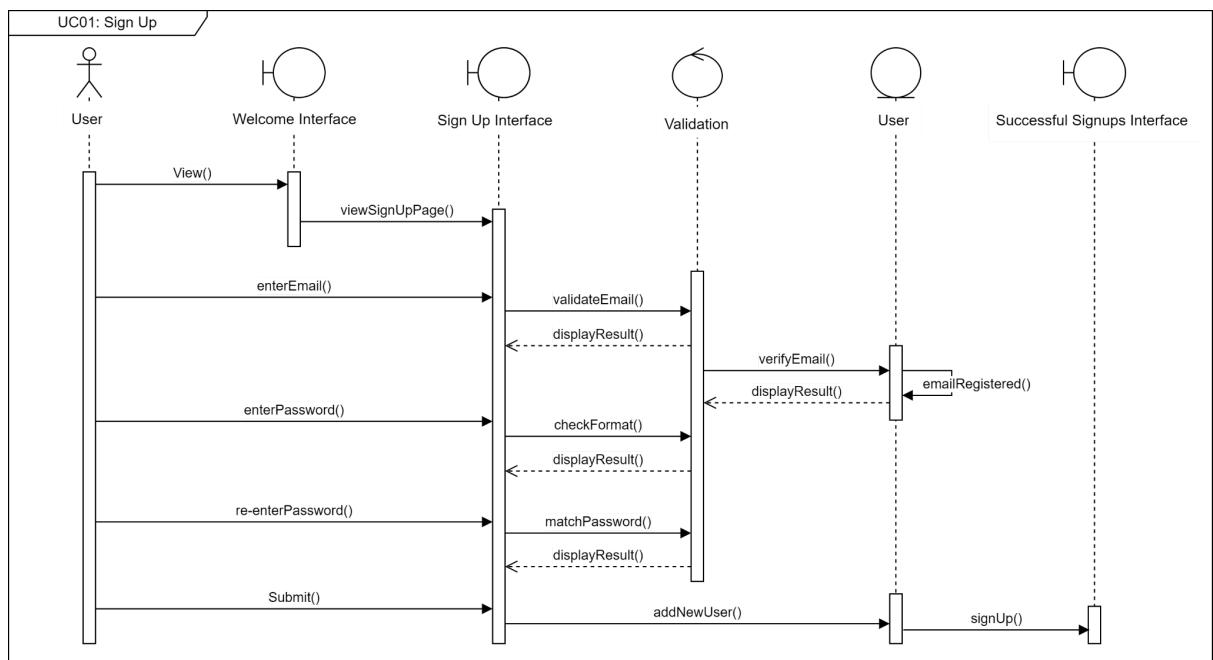


Figure 4.2.1.3: Sequence Diagram for <Sign Up>

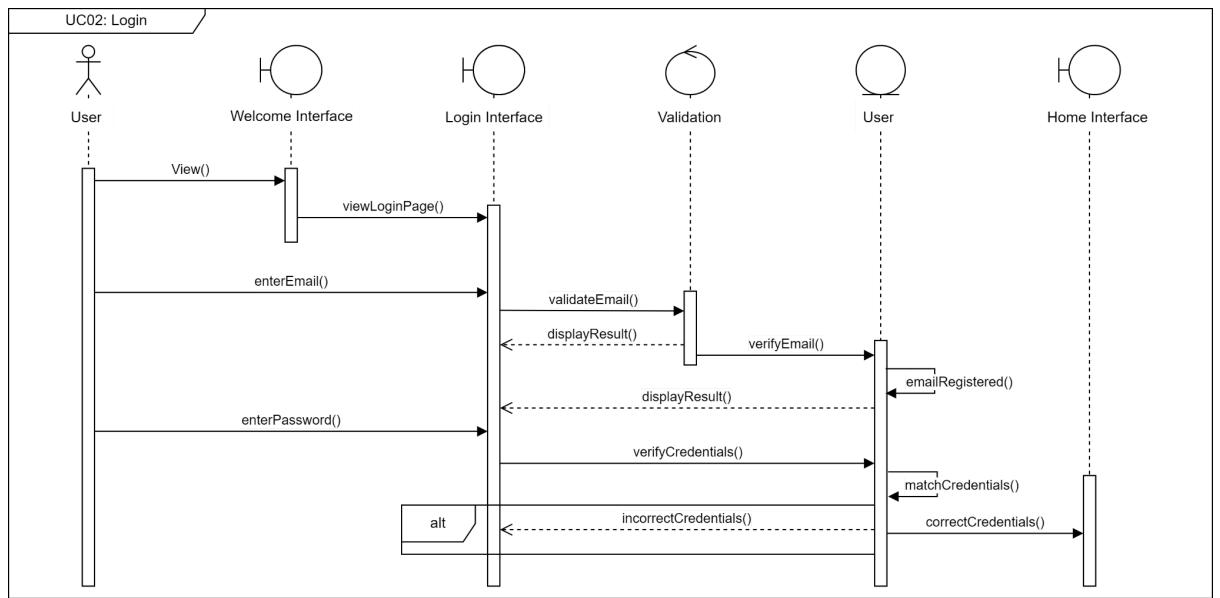


Figure 4.2.1.4: Sequence Diagram for <Login>

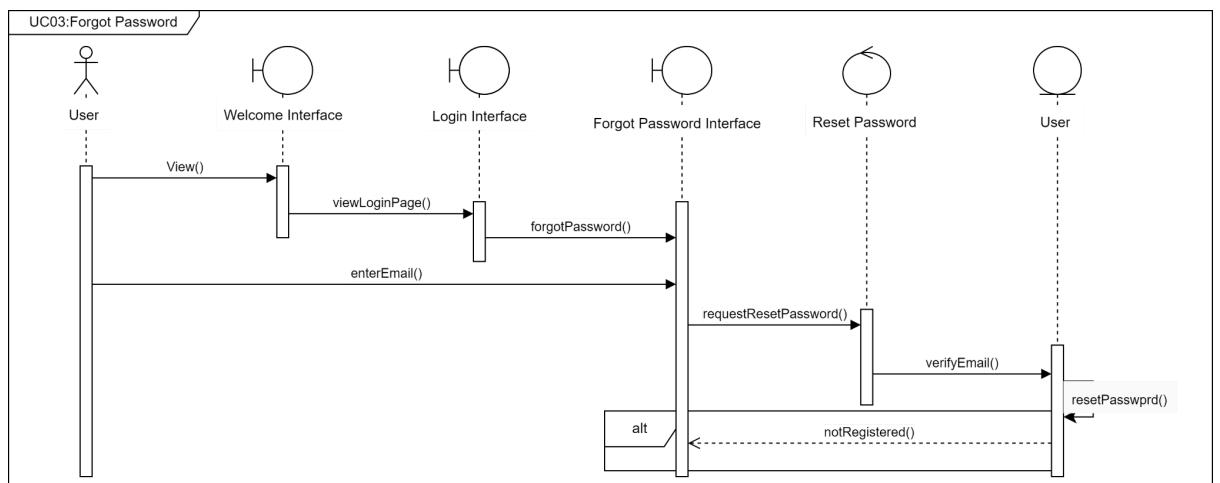


Figure 4.2.1.5: Sequence Diagram for <Forgot Password>

4.2.2 P002: <Account Management> Subsystem

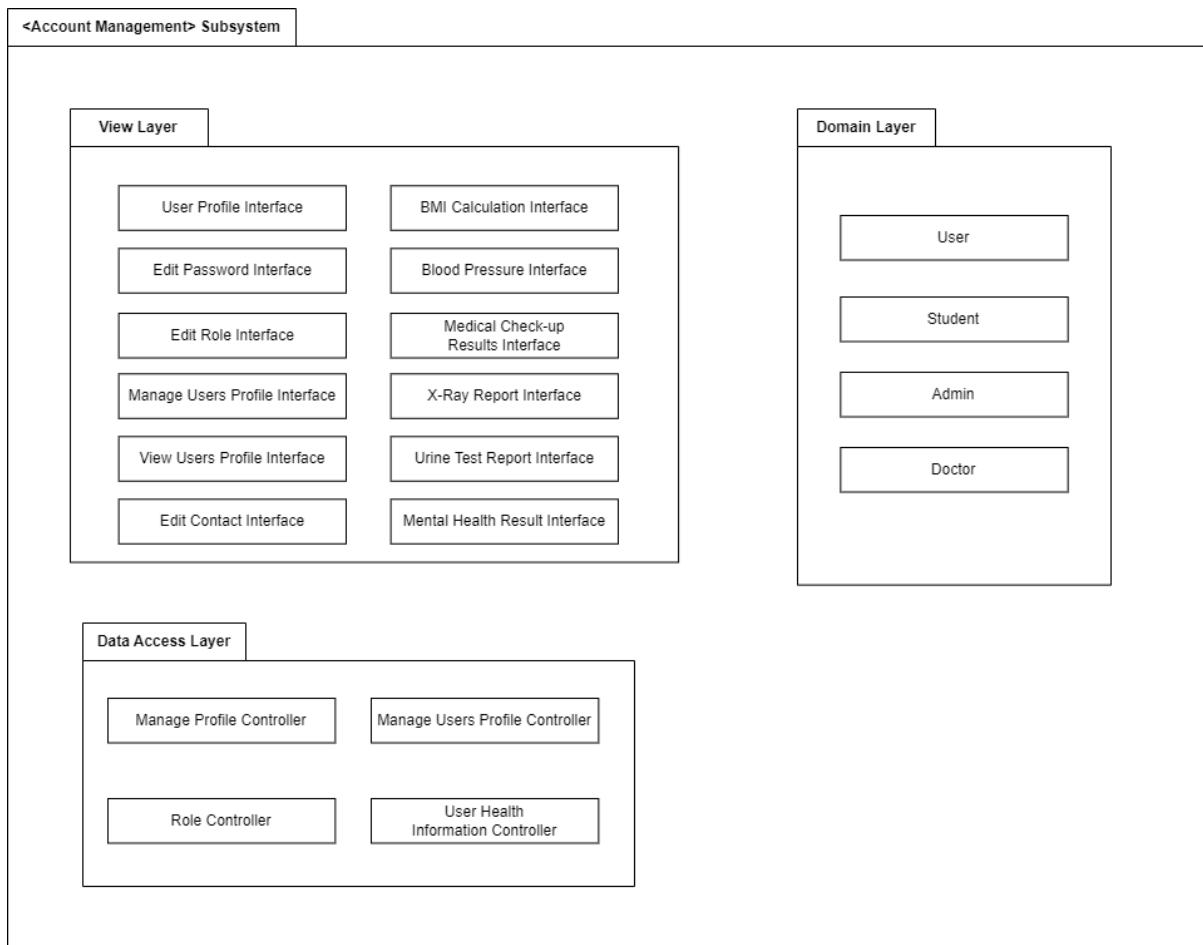


Figure 4.2.2.1: P002: <Account Management> Subsystem

4.2.2.1 Class Diagram

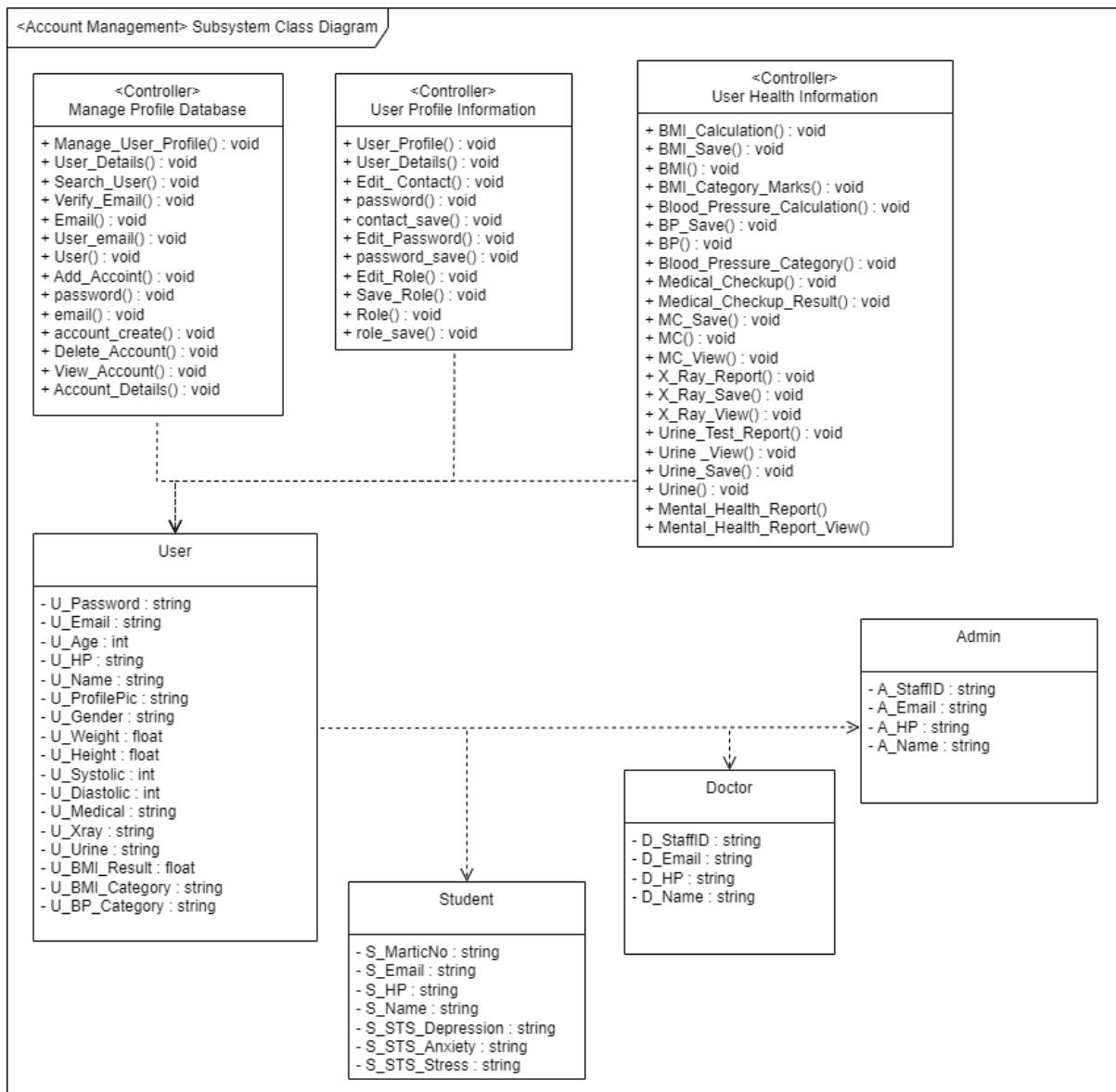


Figure 4.2.2.2: Class Diagram of P002: <Account Management> Subsystem

| | |
|--------------------|--|
| Entity Name | User |
| Method Name | Edit_Password |
| Input | 1. U_Password : string 2. U_HP : string |
| Output | password_save() for U_password |
| Algorithm | 1. Start 2. User input original password, new password, otp and contact number 3. System verify original password with U_password and otp 4. If original password same with U_password and otp correct, password saved 5. If original password different with U_password or otp wrong, display error message 6. End |

| | |
|--------------------|---|
| Entity Name | User, Student, Doctor and Admin |
| Method Name | Edit_Contact() |
| Input | U_HP : string |
| Output | contact_save() for U_HP |
| Algorithm | 1. Start 2. User input new_contactno and otp 3. System verify otp 4. If otp correct, new_contactno saved 5. If otp wrong, user re-enter otp 6. End |

| | |
|--------------------|---|
| Entity Name | User |
| Method Name | Edit_Role |
| Input | U_Role : string |
| Output | role_save() for U_Role |
| Algorithm | 1. Start 2. Doctor and Admin choose a new Role 3. Role save 4. End |

| | |
|--------------------|--|
| Entity Name | User, Student and Doctor |
| Method Name | Search_User() |
| Input | U_Email : string |
| Output | Search_User() based on U_Email. |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. Admin input email address of user account that wish to search 3. System search account based on email address 4. User account with the particular email address shown in user interface 5. End |

| | |
|--------------------|---|
| Entity Name | User, Student and Doctor |
| Method Name | Add_Account() |
| Input | <ol style="list-style-type: none"> 1. U_Email : string 2. U_Password : string |
| Output | New User Account created and saved in user entity. |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. Admin enter password and email for user to create a new user account 3. Account saved 4. End |

| | |
|--------------------|---|
| Entity Name | User, Student and Doctor |
| Method Name | Delete_Account |
| Input | - |
| Output | Account successfully deleted |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. Admin select user account and click delete 3. User account successfully deleted 4. End |

| | |
|--------------------|---|
| Entity Name | User, Student and Doctor |
| Method Name | View_Account |
| Input | - |
| Output | Account_Details() can be viewed |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. Admin select user account 3. View Users Profile Interface shown 4. End |

| | |
|--------------------|--|
| Entity Name | User |
| Method Name | BMI_Calculation() |
| Input | <ul style="list-style-type: none"> 1. U_Gender : string 2. U_Weight : float 3. U_Height : float 4. U_Age : int |
| Output | <ul style="list-style-type: none"> 1. U_Bmi_Result : float 2. U_Bmi_Category : string |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. Users choose gender and age, enter height and weight. 3. System calculates BMI based on input. 4. System displays BMI marks and BMI Category. 5. End |

| | |
|--------------------|--|
| Entity Name | User |
| Method Name | Blood_Pressure_Calculation() |
| Input | <ul style="list-style-type: none"> 1. U_Systolic : int 2. U_Diastolic : int |
| Output | U_BP_Category : string |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. Users enter systolic mmHg and diastolic mmHg 3. System calculates Blood Pressure Category based on input 4. System displays Blood Pressure Category 5. End |

| | |
|--------------------|---|
| Entity Name | User |
| Method Name | MC_Save() |
| Input | U_Medical : string |
| Output | - |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. Users click on the add button in the Medical Check-up interface 3. Users enter date of upload, upload report in image type and save 4. Medical Check-up report saved in system 5. End |

| | |
|--------------------|--|
| Entity Name | User |
| Method Name | MC_View() |
| Input | - |
| Output | - |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. Users choose the medical report that they wish to view. 3. System shows the medical report that selected 4. End |

| | |
|--------------------|--|
| Entity Name | User |
| Method Name | Urine_Save() |
| Input | U_Urine : string |
| Output | - |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. Users click on the add button in the Urine Test Report interface 3. Users enter date of upload, upload report in image type and save 4. Urine Test report saved in system 5. End |

| | |
|--------------------|--|
| Entity Name | User |
| Method Name | Urine_View() |
| Input | - |
| Output | - |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. Users choose the urine test report that they wish to view. 3. System shows the urine test report that selected 4. End |

| | |
|--------------------|--|
| Entity Name | User |
| Method Name | X_Ray_Save() |
| Input | U_Xray : string |
| Output | - |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. Users click on the add button in the X-Ray Report interface 3. Users enter date of upload, upload report in image type and save 4. X-Ray Report saved in system 5. End |

| | |
|--------------------|--|
| Entity Name | User |
| Method Name | X_Ray_View() |
| Input | - |
| Output | - |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. Users choose the X-Ray Report that they wish to view. 3. System shows the X-Ray Report that selected 4. End |

| | |
|--------------------|---|
| Entity Name | Student |
| Method Name | Mental_Health_Report_View() |
| Input | - |
| Output | - |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. Students choose mental health reports that they wish to view. 3. System shows mental health reports that selected 4. End |

4.2.2.2 Sequence Diagram

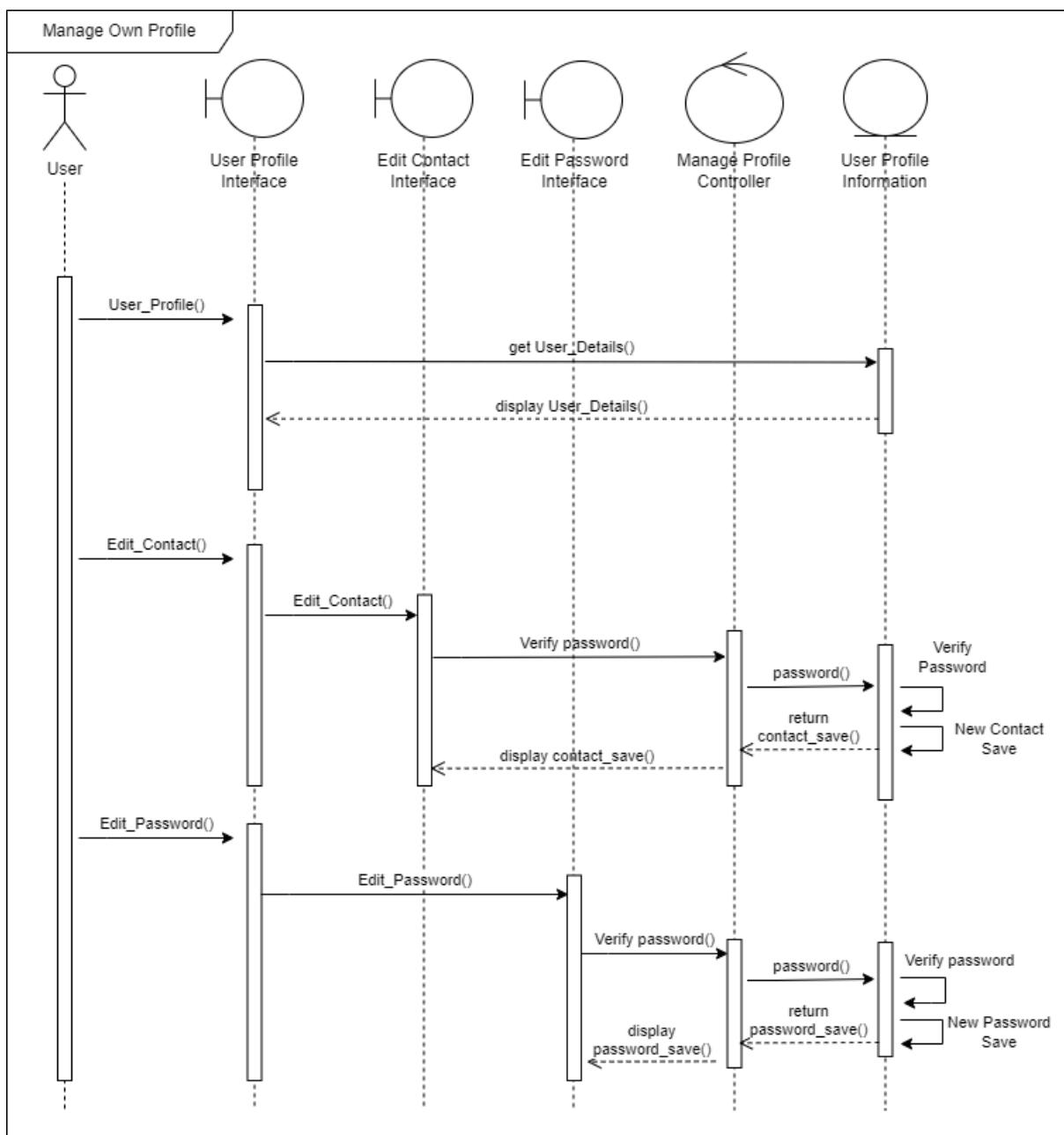


Figure 4.2.2.3: Sequence Diagram 1 for <Manage Own Account>

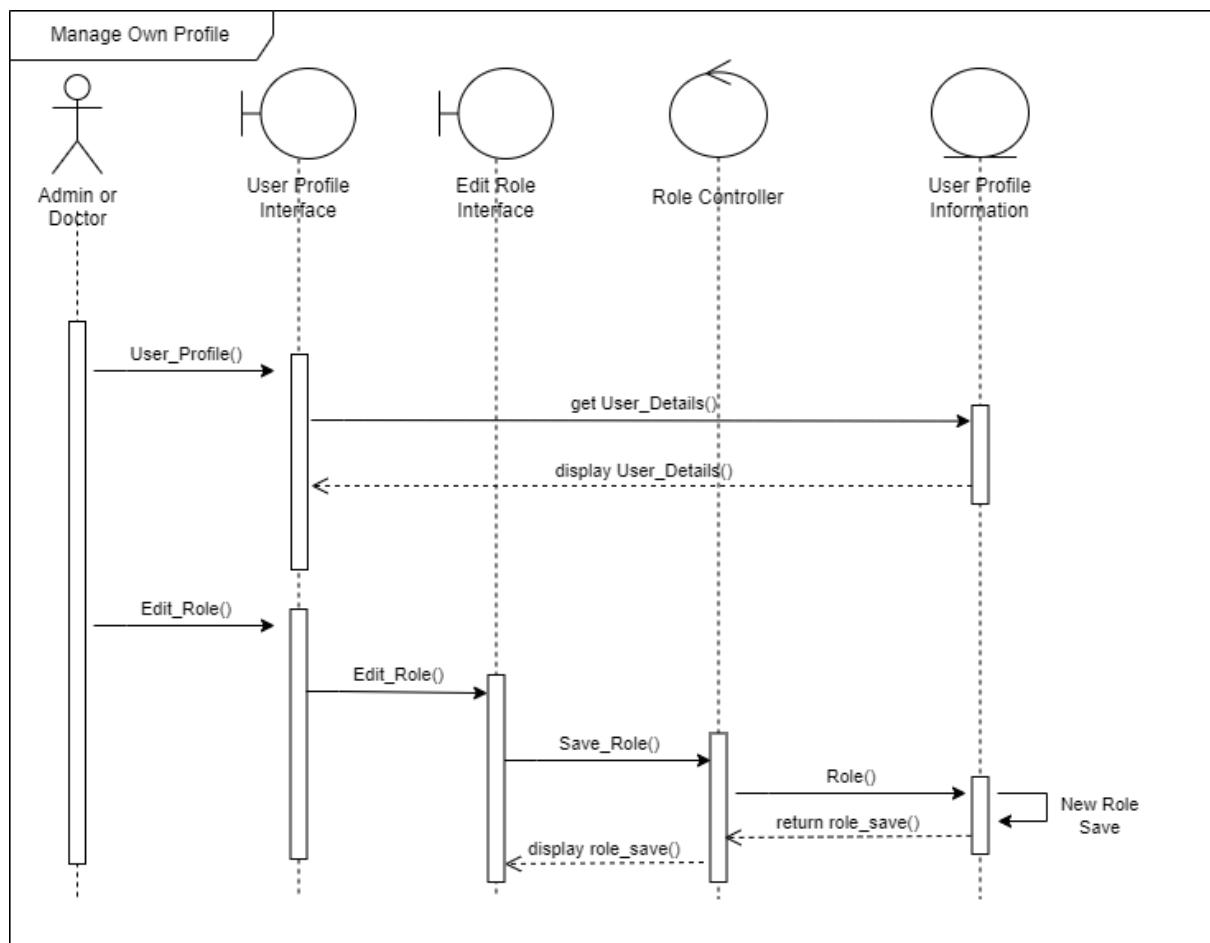


Figure 4.2.2.4: Sequence Diagram 2 for <Manage Own Account>

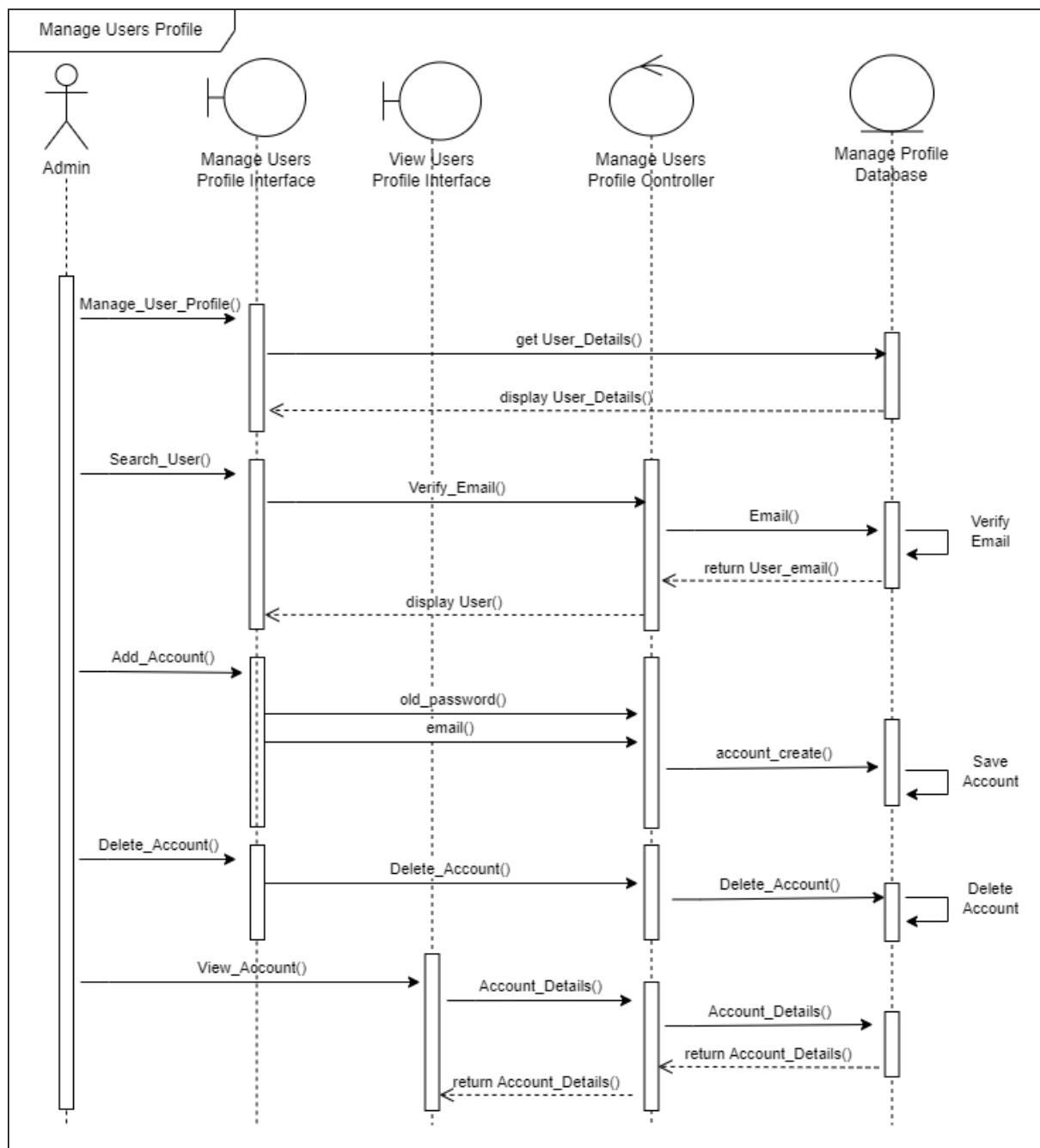
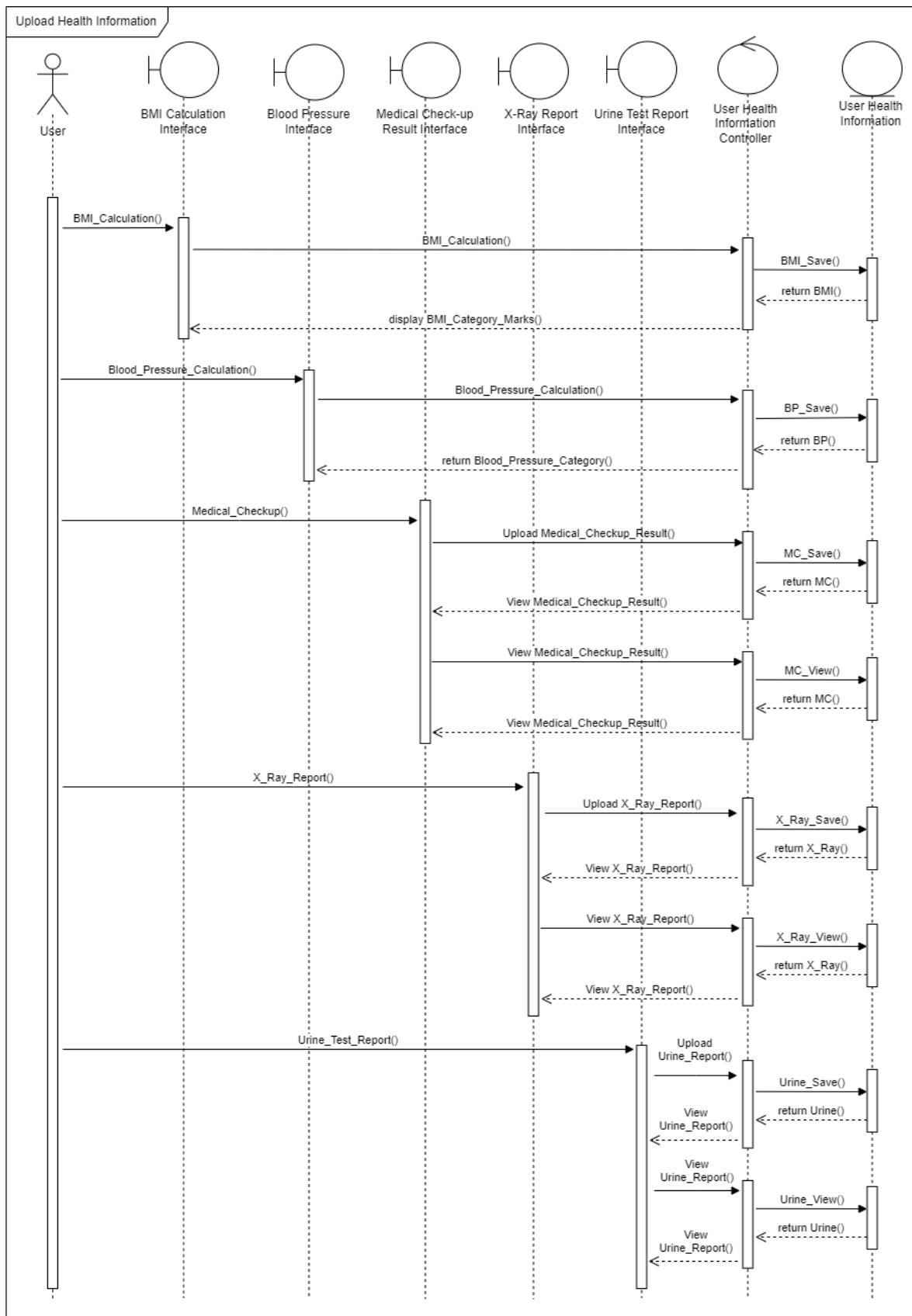


Figure 4.2.2.5: Sequence Diagram for <Manage All Users Account>



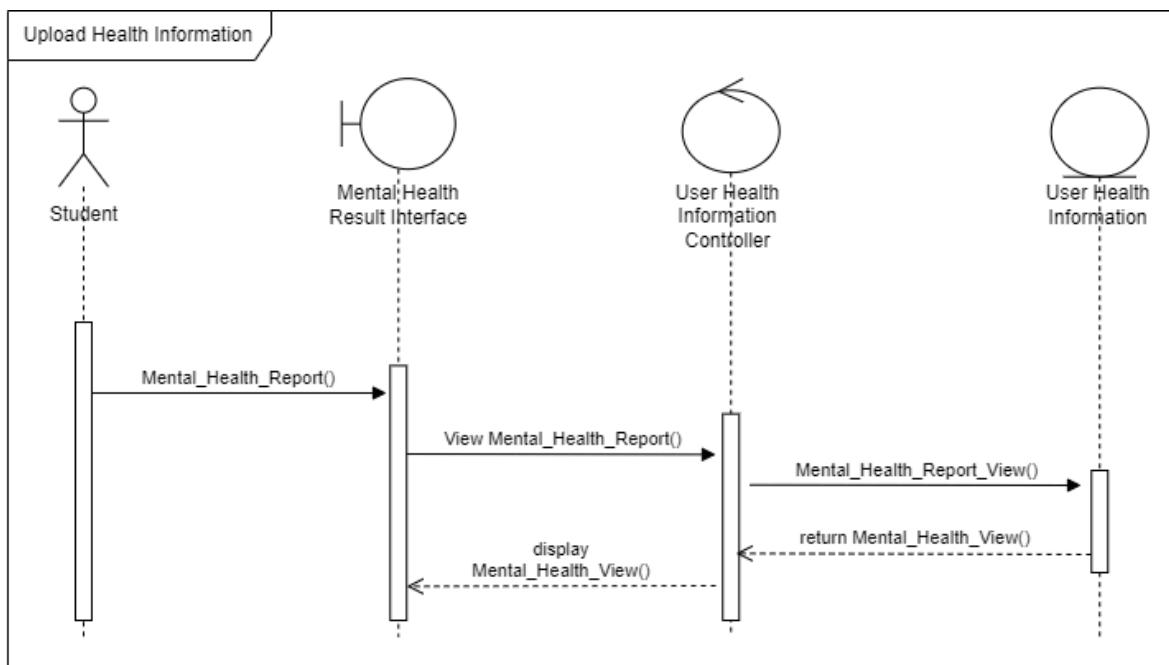


Figure 4.2.2.7: Sequence Diagram 2 for <Upload Health Information>

4.2.3 P003: <Patient Queue Management> Subsystem

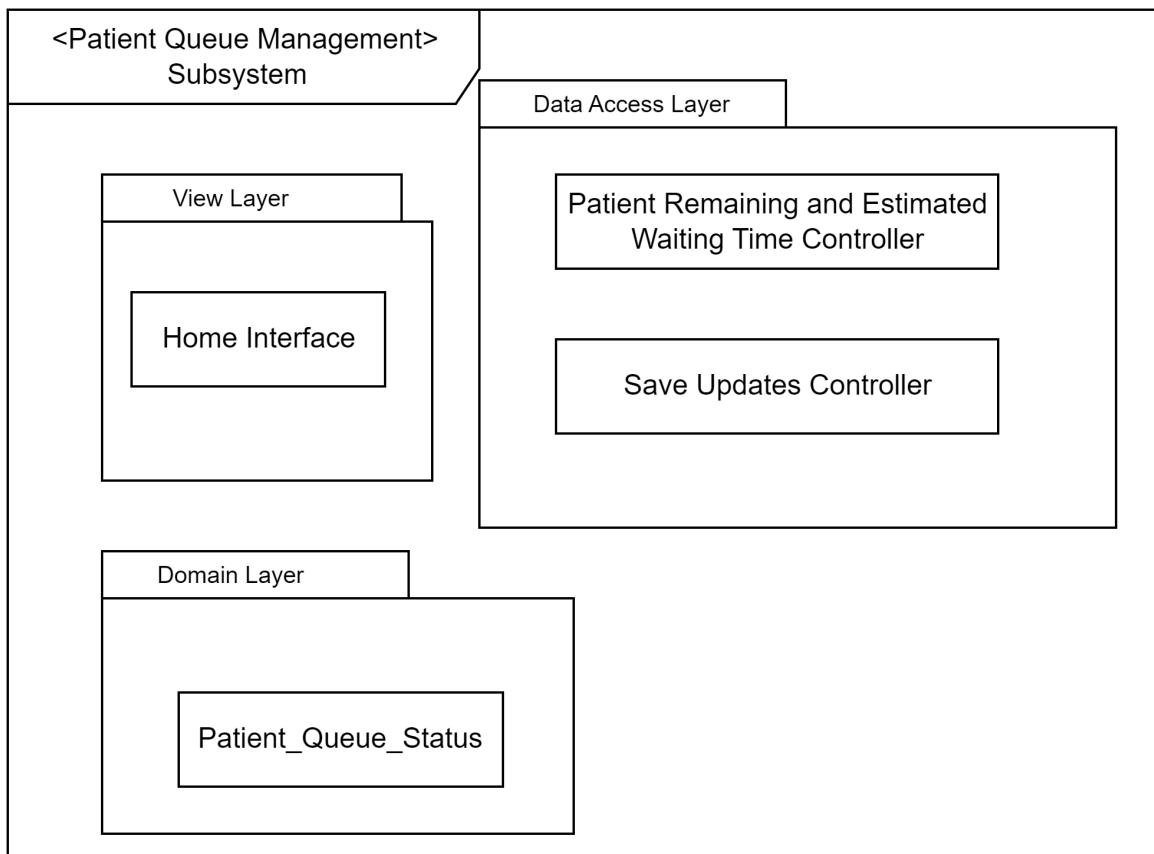


Figure 4.2.3.1: P003: <Patient Queue Management> Subsystem

4.2.3.1 Class Diagram

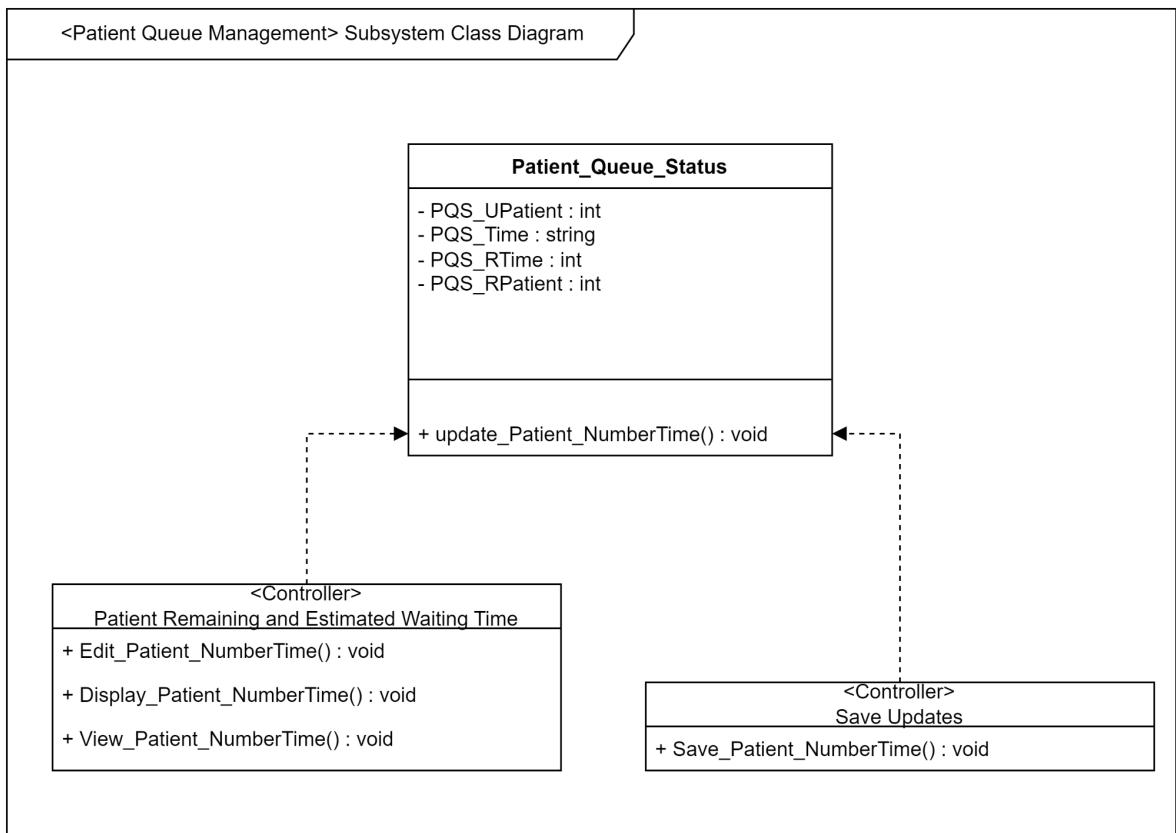


Figure 4.2.3.2: Class Diagram of P003: <Patient Queue Management> Subsystem

| | |
|--------------------|--|
| Entity Name | Patient_Queue_Status |
| Method Name | update_Patient_NumberTime |
| Input | Edit number of patient |
| Output | Update number of patient and estimated waiting time remaining |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. Taps on “Edit” at the home interface. 3. Administrator keys in numbers of patients or tap “+” to add patients, “-” to delete patients. 4. System will update the total number of patients and estimated waiting time remaining. 5. System displays an updated total number of patients and estimated waiting time remaining. 6. End |

4.2.3.2 Sequence Diagram

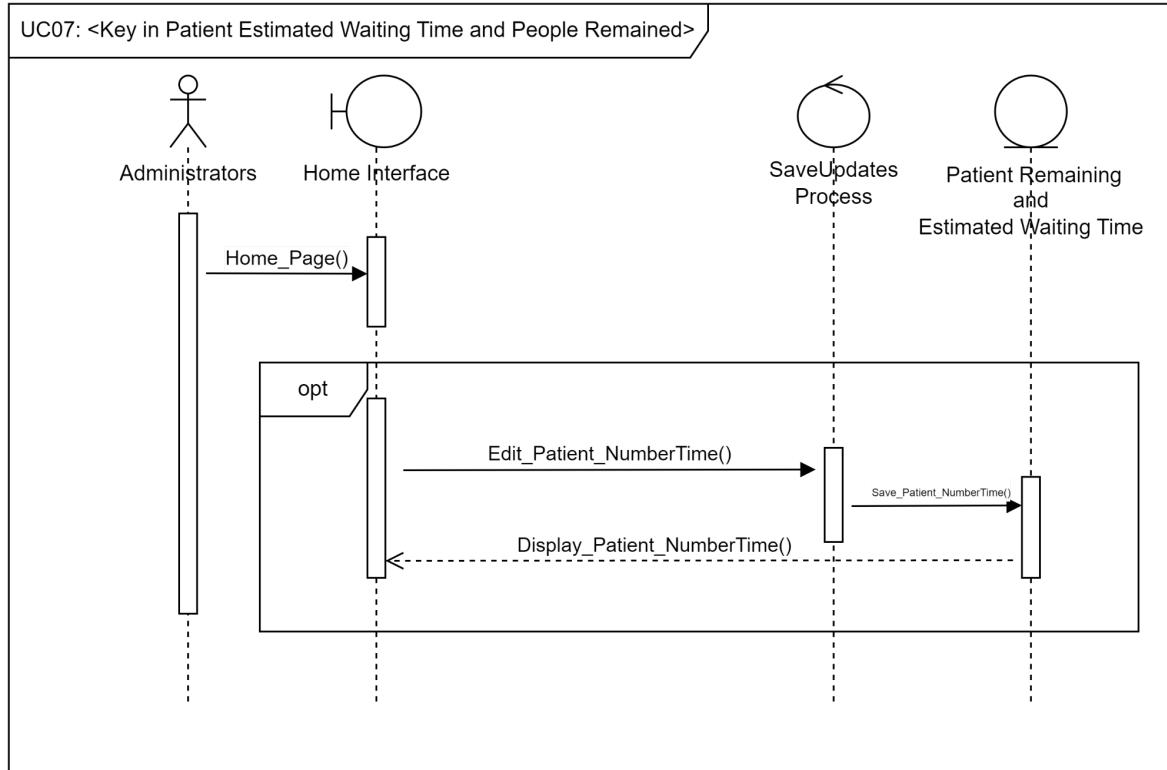


Figure 4.2.3.3: Sequence Diagram for <Key in Patient Estimated Waiting Time and people remained>

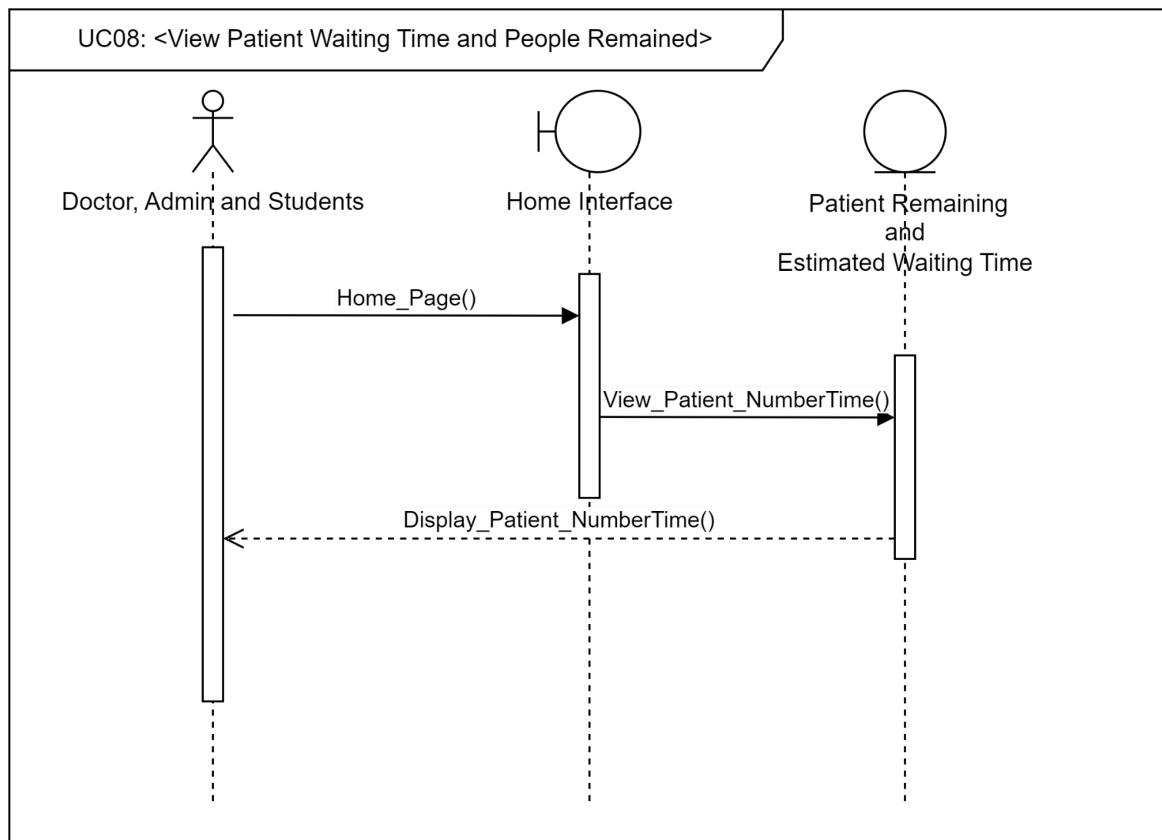


Figure 4.2.3.4: Sequence Diagram for <View Patient Waiting Time and people remained>

4.2.4 P004: <Health Tips> Subsystem

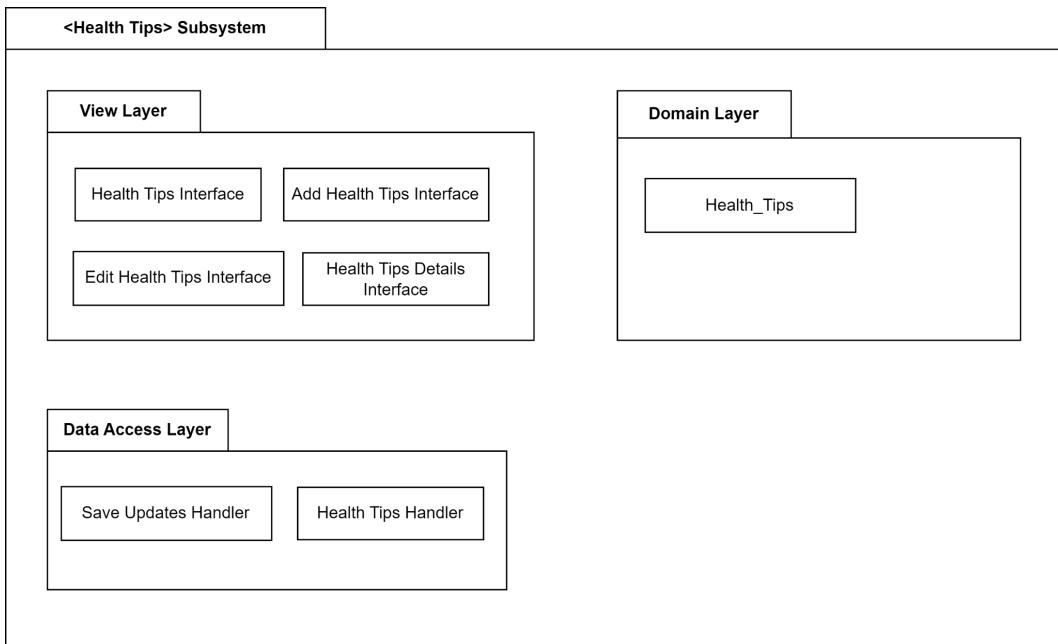


Figure 4.2.4.1: P004: <Health Tips> Subsystem

4.2.4.1 Class Diagram

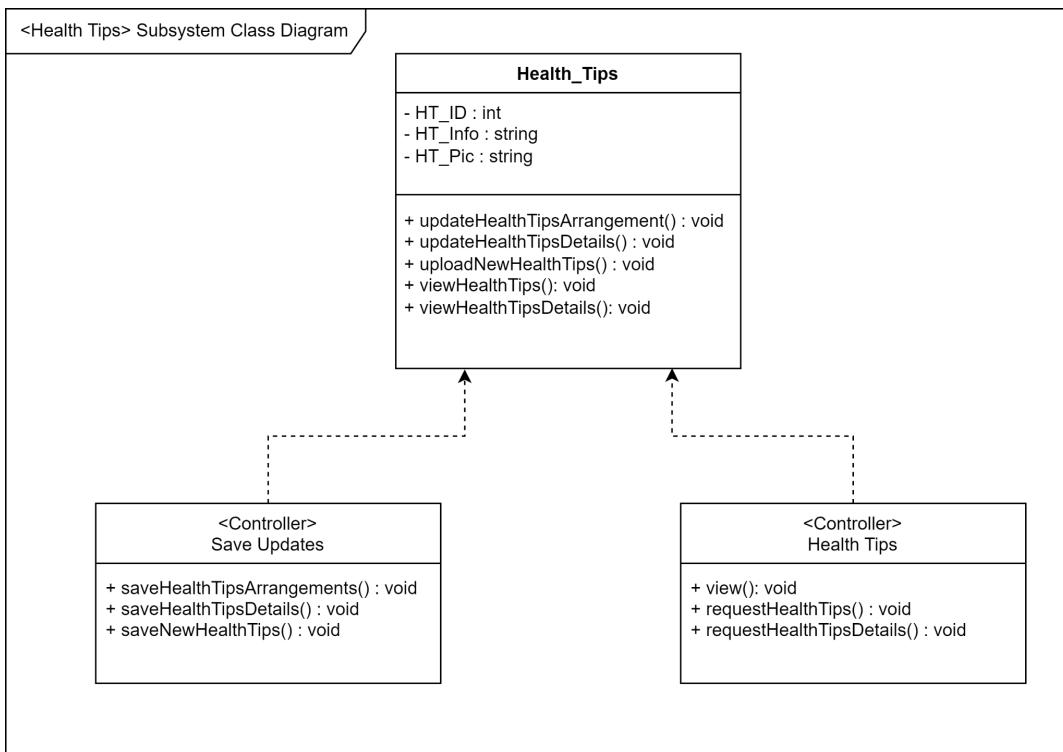


Figure 4.2.4.2: Class Diagram for <Health Tips> Subsystem

| | |
|--------------------|--|
| Entity Name | Health_Tips |
| Method Name | updateHealthTipsArrangement |
| Input | New arrangement of the existing health tips |
| Output | Display of new arrangement of the existing health tips |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. System displays the brief introduction of the health tips. 3. Administrator select “Edit” button to arrange the health tips that have been uploaded. 4. Administrator select “Save” button to save the edited arrangement of the existing health tips. 5. System saves the updates of the health tips arrangement. 6. System displays the new arrangement of the health tips. 7. End |

| | |
|--------------------|---|
| Entity Name | Health_Tips |
| Method Name | updateHealthTipsDetails |
| Input | Updates of the health tips details |
| Output | Updated health tips |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. System displays the brief introduction of the health tips. 3. Administrator selects the brief introduction of the desired edited health tip. 4. System displays the details of the health tip. 5. Administrator select “Edit” button to modify the details of the selected health tip. 6. Administrator select “Save” button to save the edited details of the health tip. 7. System saves the updates of the health tip details. 8. System displays the brief introduction of the health tips after updating the details of the health tip. 9. End |

| | |
|--------------------|--|
| Entity Name | Health_Tips |
| Method Name | uploadNewHealthTips |
| Input | New health tips |
| Output | Updated health tips |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. System displays the brief introduction of the health tips. 3. Administrator selects “Add” button to add the new health tips. 4. System displays a page for uploading new health tips. |

| | |
|--|---|
| | <p>5. Administrator uploads the new health tips.</p> <p>6. Administrator selects “Save” button to save the new health tips.</p> <p>7. System saves the new health tips.</p> <p>8. System displays the brief introduction of the health tips after uploading the new health tip.</p> <p>9. End</p> |
|--|---|

| | |
|--------------------|--|
| Entity Name | Health_Tips |
| Method Name | viewHealthTips |
| Input | Selection on the “Health Tips” button of Homepage |
| Output | Display of the brief introduction of the health tips |
| Algorithm | <p>1. Start</p> <p>2. System fetches the brief introduction of the health tips.</p> <p>3. System displays the brief introduction of the health tips.</p> <p>4. End</p> |

| | |
|--------------------|---|
| Entity Name | Health_Tips |
| Method Name | viewHealthTipsDetails |
| Input | Selection on the brief introduction of the health tip |
| Output | Display of the details of the health tips. |
| Algorithm | <p>1. Start</p> <p>2. System fetches the details of the selected health tip.</p> <p>3. System displays the details of the health tip.</p> <p>4. End</p> |

4.2.4.2 Sequence Diagram

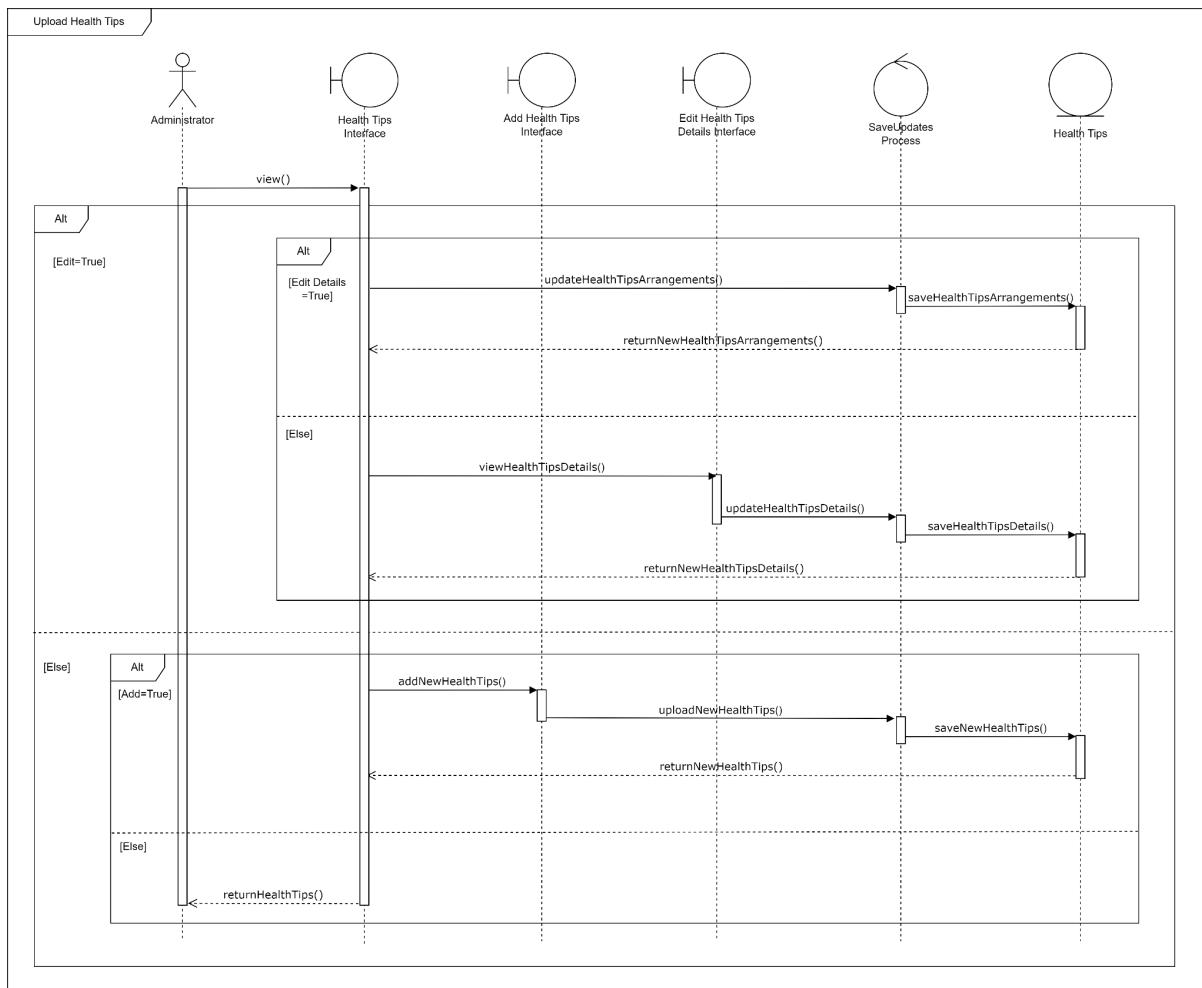


Figure 4.2.4.3: Sequence Diagram for <Upload Health Tips>

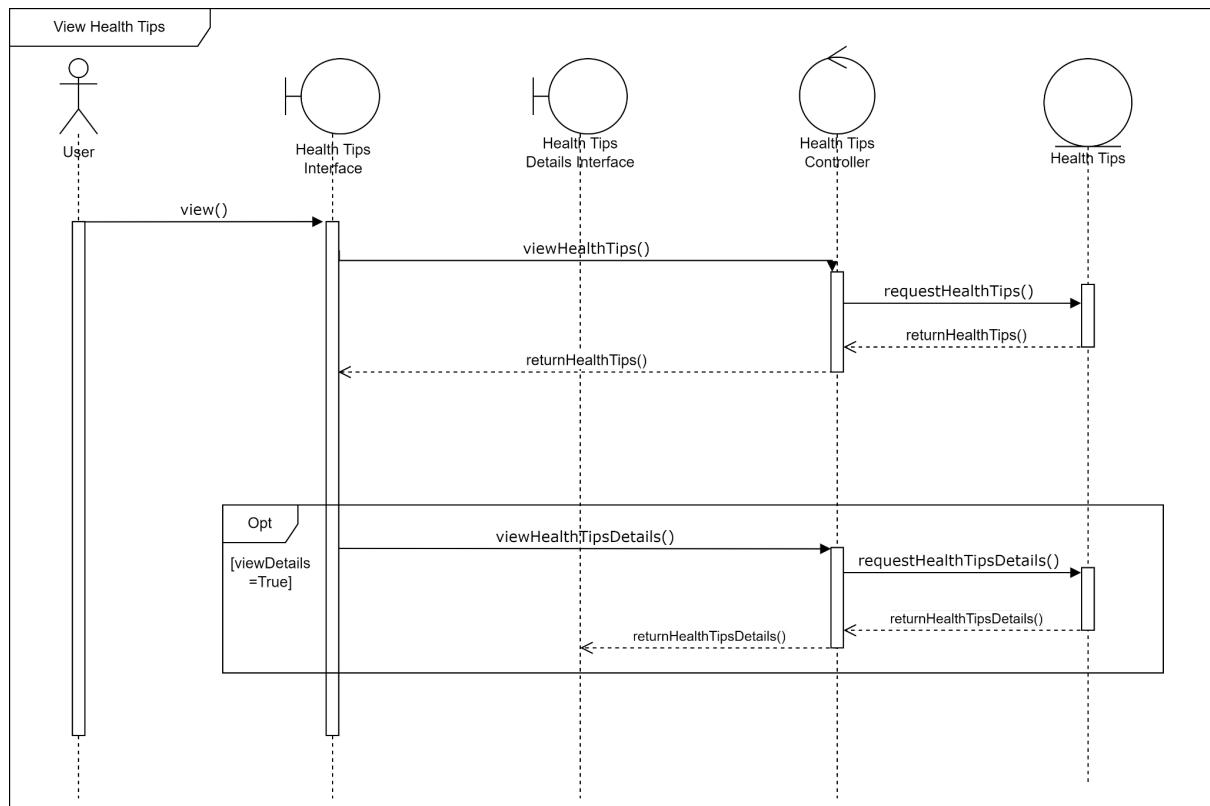


Figure 4.2.4.4: Sequence Diagram for <View Health Tips>

4.2.5 P005: <Mental Health Support> Subsystem

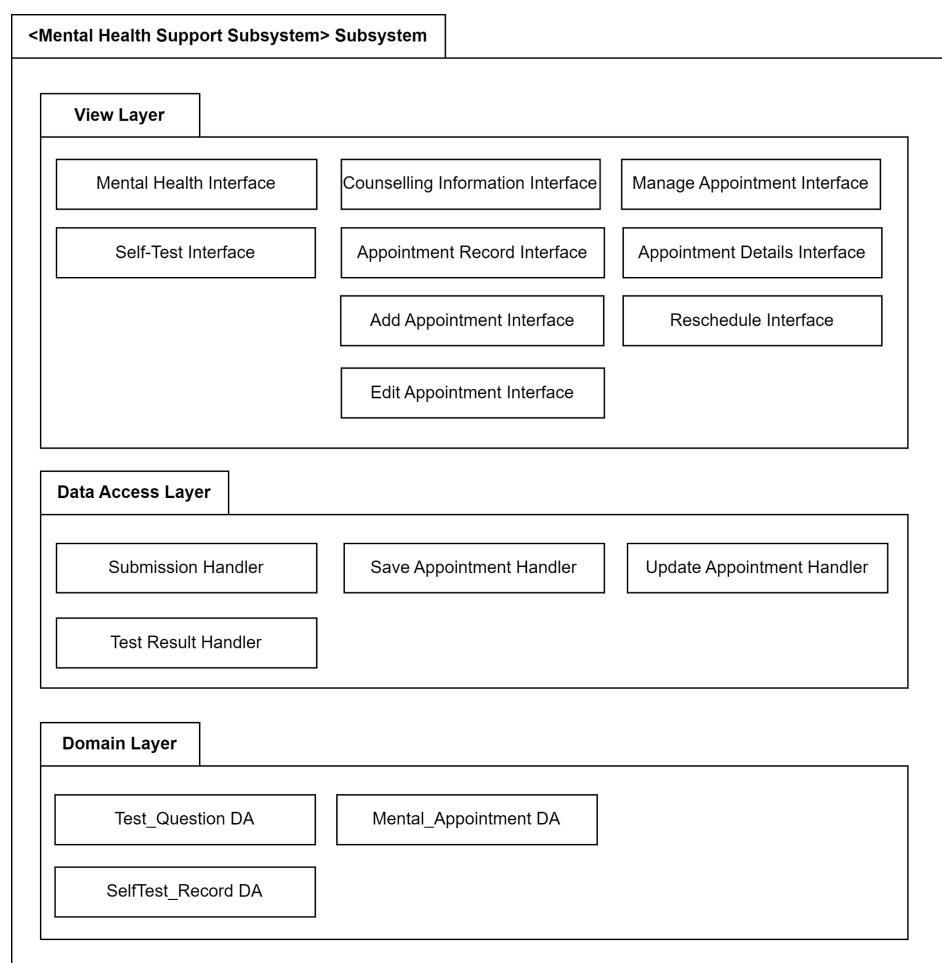


Figure 4.2.5.1: P005: <Mental Health Support> Subsystem>

4.2.5.1 Class Diagram

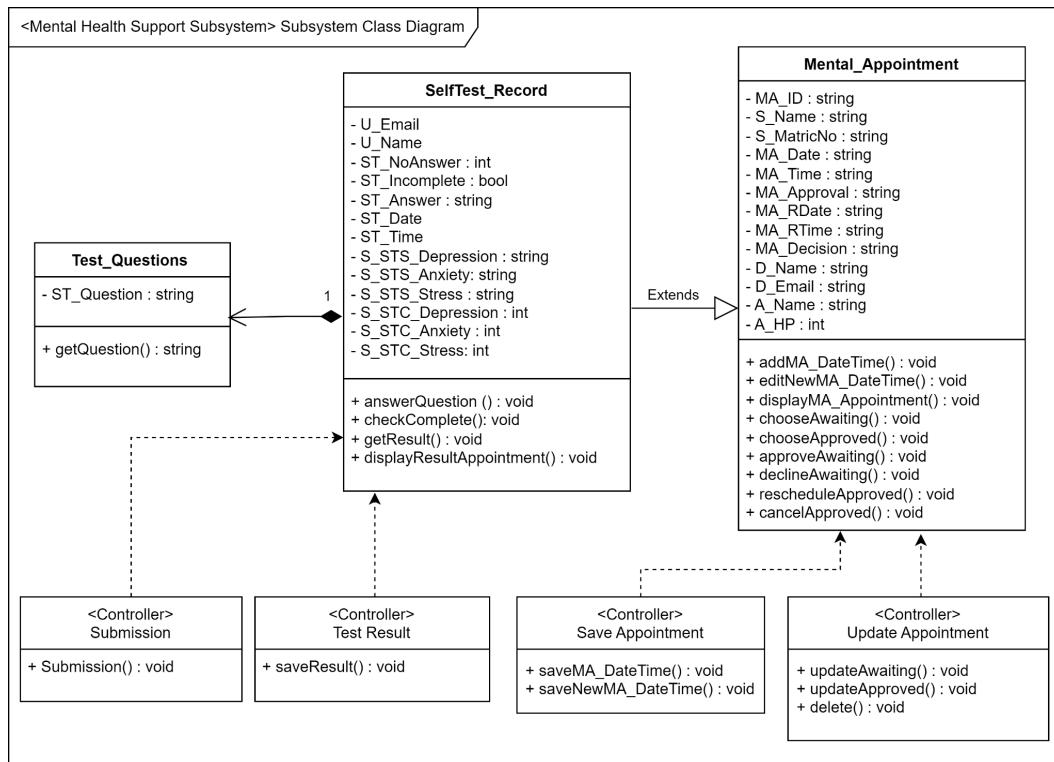


Figure 4.2.5.2: Class Diagram for <Mental Health Support> Subsystem

| | |
|--------------------|--|
| Entity Name | Test_Questions |
| Method Name | getQuestions |
| Input | - |
| Output | Self-test questions |
| Algorithm | <ol style="list-style-type: none"> Start System fetches information of the self-test questions. System displays the self-test questions in the self-test page. End |

| | |
|--------------------|---|
| Entity Name | SelfTest_Record |
| Method Name | answerQuestion |
| Input | Answer of self-test questions |
| Output | - |
| Algorithm | <ol style="list-style-type: none"> Start Students answer the self-test questions. Students click the “Submit” button. End |

| | |
|--------------------|---|
| Entity Name | SelfTest_Record |
| Method Name | checkComplete |
| Input | Number of questions that has been answered |
| Output | - |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. System checks the number of questions that have been answered. 3. If all questions have been answered, the system adds the answer to the database. 4. If any one of the questions has not been answered, the system prompts the student to answer all the questions. 5. End |

| | |
|--------------------|---|
| Entity Name | SelfTest_Record |
| Method Name | getResult |
| Input | Answer of self-test questions |
| Output | - |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. System calculates the score of the self-test in the Anxiety part. 3. System calculates the score of the self-test in the Depression part. 4. System calculates the score of the self-test in the Stress part. 5. System determines the status of Anxiety. 6. System determines the status of Depression. 7. System determines the status of Status. 8. System adds the score and status of the Anxiety part in the database. 9. System adds the score and status of the Depression part in the database. 10. System adds the score and status of the Stress part in the database. 11. End |

| | |
|--------------------|---|
| Entity Name | SelfTest_Record |
| Method Name | displayResultAppointment |
| Input | - |
| Output | Status and score of the Anxiety part, status and score of the Depression part, status and score of the Stress part |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. System fetches information of the Depression, Anxiety and Stress part. |

| | |
|--|---|
| | <p>3. System displays the score and status in the counseling information page.</p> <p>4. If status is severe or extremely severe, display the "Appointment" button.</p> <p>5. End</p> |
|--|---|

| | |
|--------------------|---|
| Entity Name | Mental_Appointment |
| Method Name | addMA_DateTime |
| Input | Appointment date, Appointment time |
| Output | - |
| Algorithm | <p>1. Start</p> <p>2. Students select the appointment date.</p> <p>3. Students select the appointment time.</p> <p>4. System adds the date to the database.</p> <p>5. System adds the time to the database.</p> <p>6. End</p> |

| | |
|--------------------|---|
| Entity Name | Mental_Appointment |
| Method Name | editNew_MADateTime |
| Input | Appointment date, Appointment time |
| Output | - |
| Algorithm | <p>1. Start</p> <p>2. Students select the new appointment date.</p> <p>3. Students select the new appointment time.</p> <p>4. System adds the new date to the database.</p> <p>5. System adds the new time to the database.</p> <p>6. End</p> |

| | |
|--------------------|---|
| Entity Name | Mental_Appointment |
| Method Name | displayMA_Appointment |
| Input | - |
| Output | Appointment date and time for appointments that are awaiting approval, Appointment date and time for appointments that have already been approved |
| Algorithm | <p>1. Start</p> <p>2. System fetches information of appointment date and time for appointments that are awaiting approval and appointments that have already been approved.</p> <p>3. System displays each appointment date and time.</p> <p>4. End</p> |

| | |
|--------------------|--|
| Entity Name | Mental_Appointment |
| Method Name | chooseAwaiting |
| Input | Selection on the appointments that are awaiting approval |
| Output | Details of appointments that are awaiting approval |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. System fetches information of appointment details for appointments that are awaiting approval 3. The system displays the details of the appointment. 4. End |

| | |
|--------------------|--|
| Entity Name | Mental_Appointment |
| Method Name | chooseApproved |
| Input | Selection on the appointments that have already been approved |
| Output | Details of appointments that have already been approved |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. System fetches information of appointment details for appointments that have already been approved. 3. The system displays the details of the appointment. 4. End |

| | |
|--------------------|---|
| Entity Name | Mental_Appointment |
| Method Name | approveAwaiting |
| Input | Selection on the “Approve” button of the appointments that are awaiting approval |
| Output | Updated appointments list |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. The system saves the updated approval status of the appointment. 3. The appointment disappears from the list of appointments that are awaiting approval and is displayed in the list of appointments that are already approved. 4. The system displays the appointments list that have been updated. 5. End |

| | |
|--------------------|--|
| Entity Name | Mental_Appointment |
| Method Name | declineAwaiting |
| Input | Selection on the “Decline” button of the appointments that are awaiting approval |

| | |
|------------------|--|
| Output | Updated appointments list |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. The system saves the updated approval status of the appointment. 3. The appointment disappears from the list of appointments that are awaiting approval. 4. The system displays the appointments list that have been updated. 5. End |

| | |
|--------------------|---|
| Entity Name | Mental_Appointment |
| Method Name | rescheduleApproved |
| Input | Selection on the “Reschedule” button of the appointments that are already approval |
| Output | Updated appointment list |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. Doctor or Administrator selects the date and time to reschedule the appointment. 3. Doctor or Administrator selects the “Save” button. 4. The system saves the rescheduled date and time of the appointment. 5. The system displays the appointments list that have been updated. 6. End |

| | |
|--------------------|---|
| Entity Name | Mental_Appointment |
| Method Name | cancelApproved |
| Input | Selection on the “Cancel” button of the appointments that are already approval |
| Output | Updated appointments list |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. The system saves the updated approval status of the appointment. 3. The appointment will disappear from the list of appointments that have already been approved. 4. The system displays the appointments list that have been updated. 5. End |

4.2.5.2 Sequence Diagram

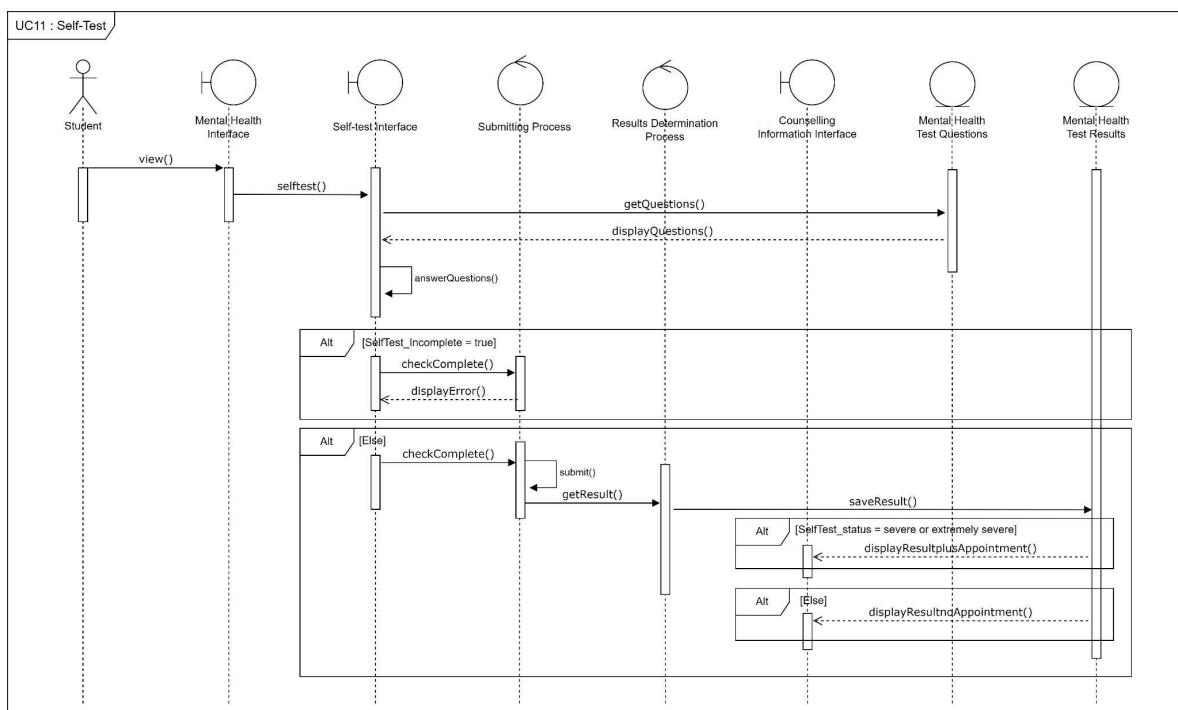


Figure 4.2.5.3: Sequence Diagram for <Self-Test>

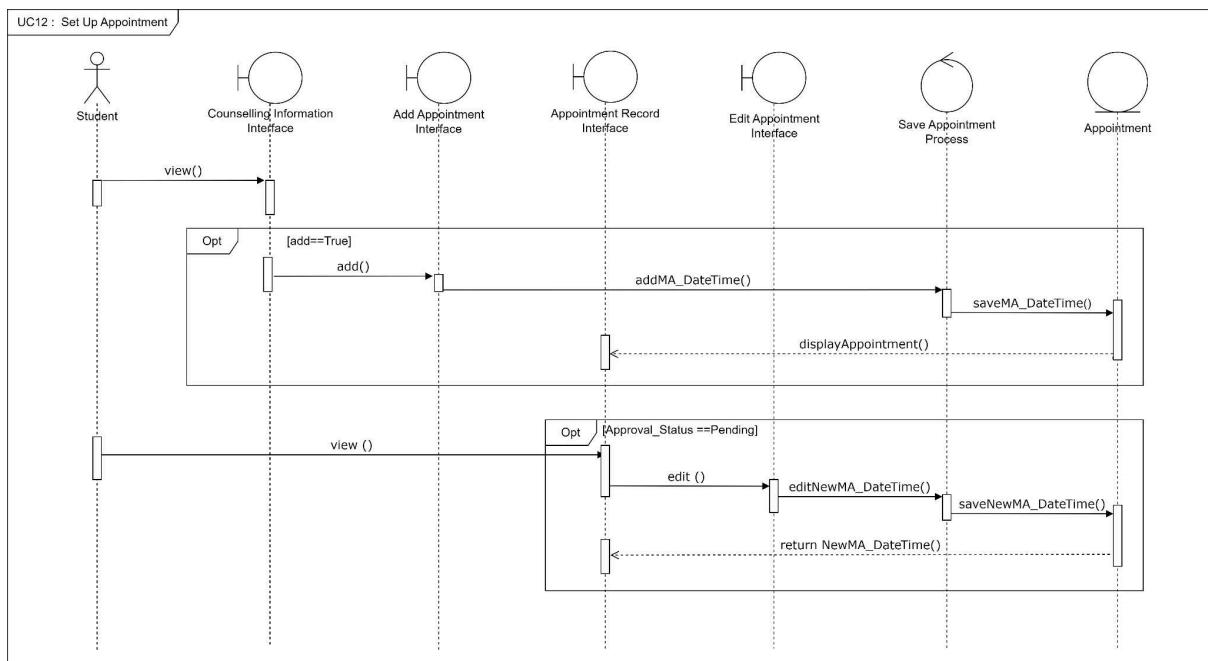


Figure 4.2.5.4: Sequence Diagram for <Set Up Mental Appointment>

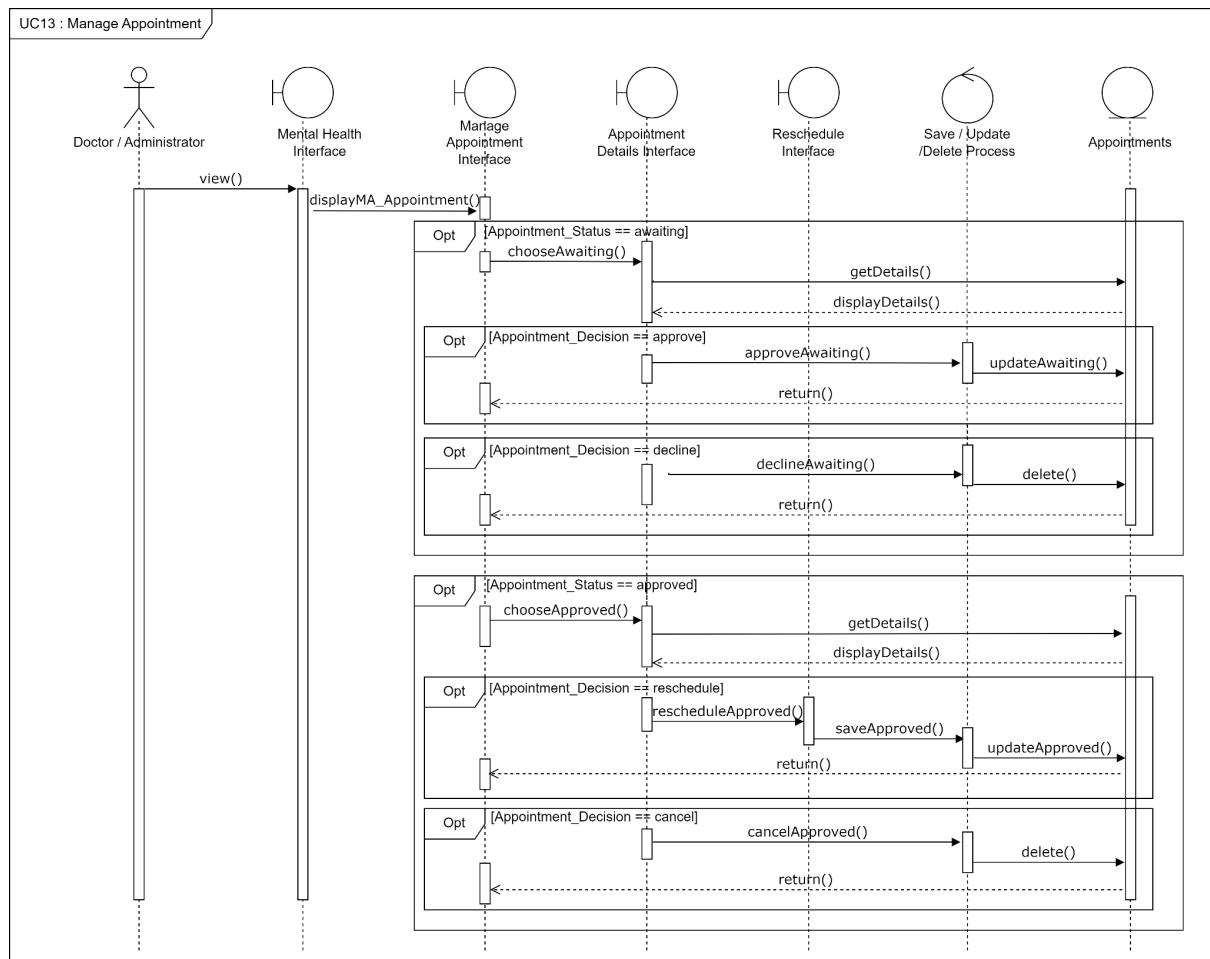


Figure 4.2.5.5: Sequence Diagram for <Manage Mental Appointment>

4.2.6 P006: <Dental Health Appointment> Subsystem

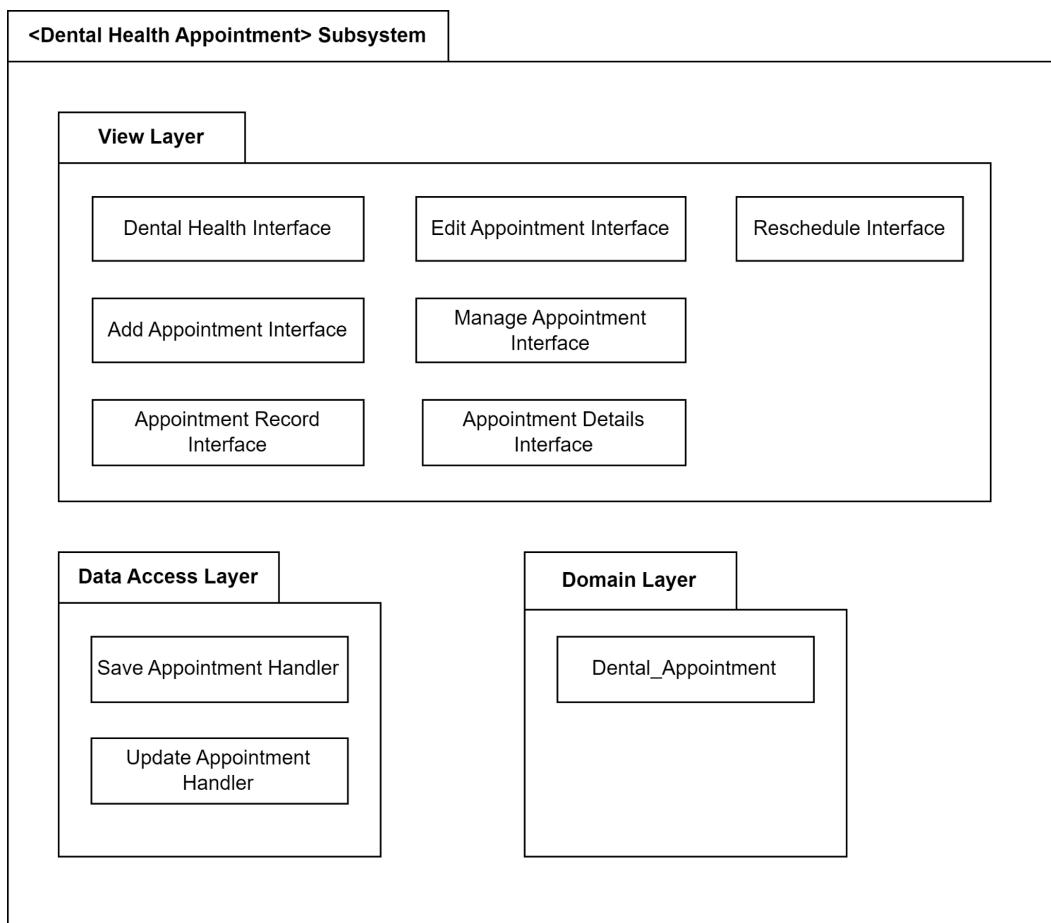


Figure 4.2.6.1: P006: <Dental Health Appointment> Subsystem

4.2.6.1 Class Diagram

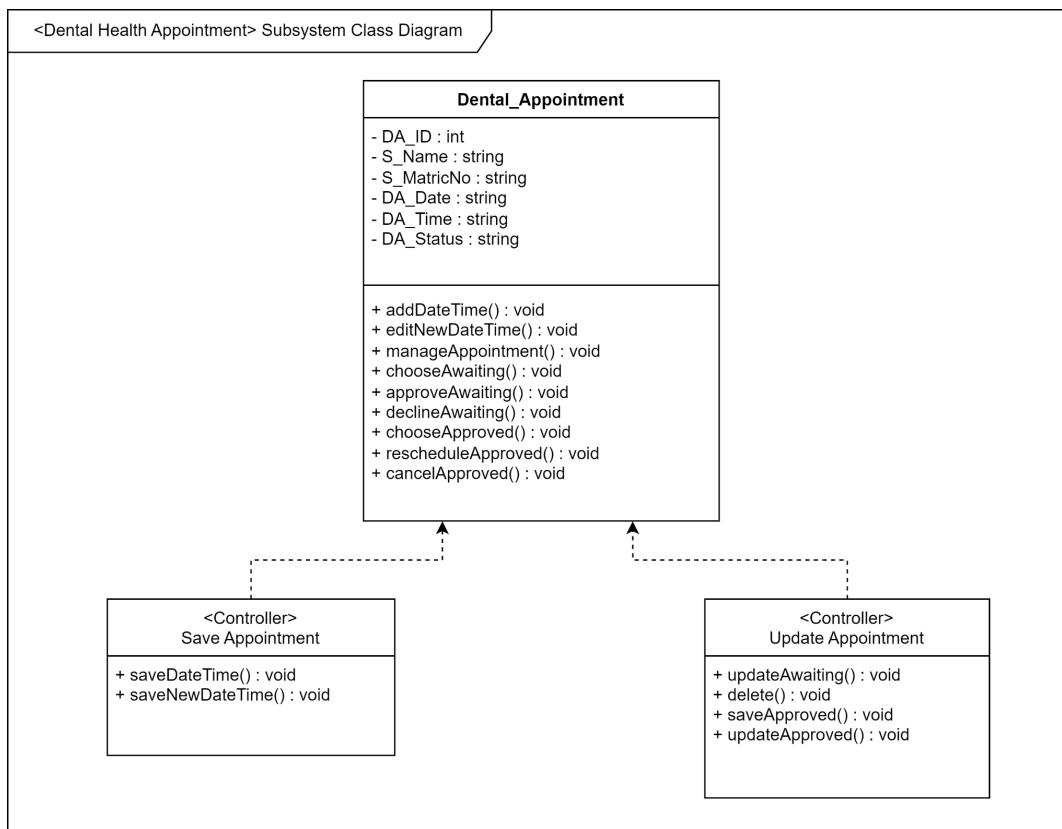


Figure 4.2.6.2: Class Diagram for <Dental Health Appointment> Subsystem

| | |
|--------------------|---|
| Entity Name | Dental_Appointment |
| Method Name | addDateTime |
| Input | Date and time of desired appointment |
| Output | Success of Application |
| Algorithm | <ol style="list-style-type: none"> 6. Start 7. System displays the dental health interface. 8. Student selects “Add Your Appointment” button to add the appointment. 9. System displays the add appointment interface. 10. Student selects the date and time desired for an appointment. 11. Student selects the “Submit” button to apply for the appointment. 12. System saves the application of the appointment. 13. System displays a page with the message of submit successfully and await for approval. 14. End |

| | |
|--------------------|--|
| Entity Name | Dental_Appointment |
| Method Name | editNewDateTime |
| Input | New date and time of desired appointment |
| Output | Success of Saving Reschedule |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. System displays the dental health interface. 3. Student selects “Edit Your Appointment” button to add the appointment. 4. System displays the appointment records. 5. If the approval status is shown “Pending” and the student selects the “Edit” button, the system displays the edit appointment interface. 6. Student selects a new date and time to reschedule the appointment. 7. Student selects the “Save” button to apply for the appointment. 8. System saves the application of the appointment. 9. System displays a page with the message of save successfully and await for approval. 10. End |

| | |
|--------------------|--|
| Entity Name | Dental_Appointment |
| Method Name | manageAppointment |
| Input | Selection on the “Appointment” button |
| Output | List of the appointments that are awaiting approval and already approved |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. The system displays the appointments list for the appointments that are awaiting approval at the upper part of the interface and the appointments list for the appointments that have already been approved at the lower part of the interface according to the date. 3. End |

| | |
|--------------------|---|
| Entity Name | Dental_Appointment |
| Method Name | chooseAwaiting |
| Input | Selection on the appointments that are awaiting approval |
| Output | Details of the appointments |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. The system fetches the details of the appointments. 3. The system displays the details of the appointments. 4. End |

| | |
|--------------------|---|
| Entity Name | Dental_Appointment |
| Method Name | approveAwaiting |
| Input | Selection on the “Approve” button of the appointments that are awaiting approval |
| Output | Updated appointments list |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. The system saves the updated approval status of the appointment. 3. The appointment disappears from the list of appointments that are awaiting approval and is displayed in the list of appointments that are already approved. 4. The system displays the appointments list that have been updated. 5. End |

| | |
|--------------------|--|
| Entity Name | Dental_Appointment |
| Method Name | declineAwaiting |
| Input | Selection on the “Decline” button of the appointments that are awaiting approval |
| Output | Updated appointments list |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. The system saves the updated approval status of the appointment. 3. The appointment disappears from the list of appointments that are awaiting approval. 4. The system displays the appointments list that have been updated. 5. End |

| | |
|--------------------|---|
| Entity Name | Dental_Appointment |
| Method Name | chooseApproved |
| Input | Selection on the appointments that are already approved |
| Output | - |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. The system fetches the details of the appointments. 3. The system displays the details of the appointments. 4. End |

| | |
|--------------------|--|
| Entity Name | Dental_Appointment |
| Method Name | rescheduleApproved |
| Input | Selection on the “Reschedule” button of the appointments that are already approval |
| Output | Updated appointment list |

| | |
|------------------|---|
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. Doctor or Administrator selects the date and time to reschedule the appointment. 3. Doctor or Administrator selects the “Save” button. 4. The system saves the rescheduled date and time of the appointment. 5. System displays a page with the message of save successfully. 6. The system displays the appointments list that have been updated. 7. End |
|------------------|---|

| | |
|--------------------|---|
| Entity Name | Dental_Appointment |
| Method Name | cancelApproved |
| Input | Selection on the “Cancel” button of the appointments that are already approval |
| Output | Updated appointments list |
| Algorithm | <ol style="list-style-type: none"> 1. Start 2. The system saves the updated approval status of the appointment. 3. The appointment will disappear from the list of appointments that have already been approved. 4. System displays a page with the message of cancel successfully. 5. The system displays the appointments list that have been updated. 6. End |

4.2.6.2 Sequence Diagram

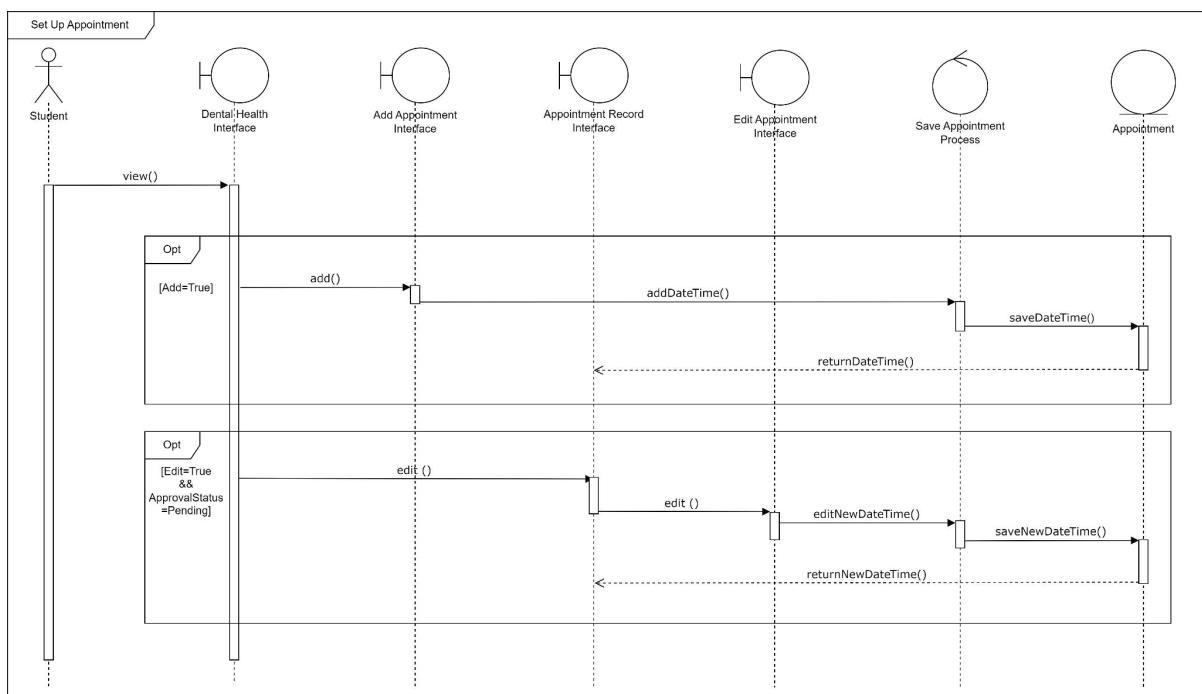
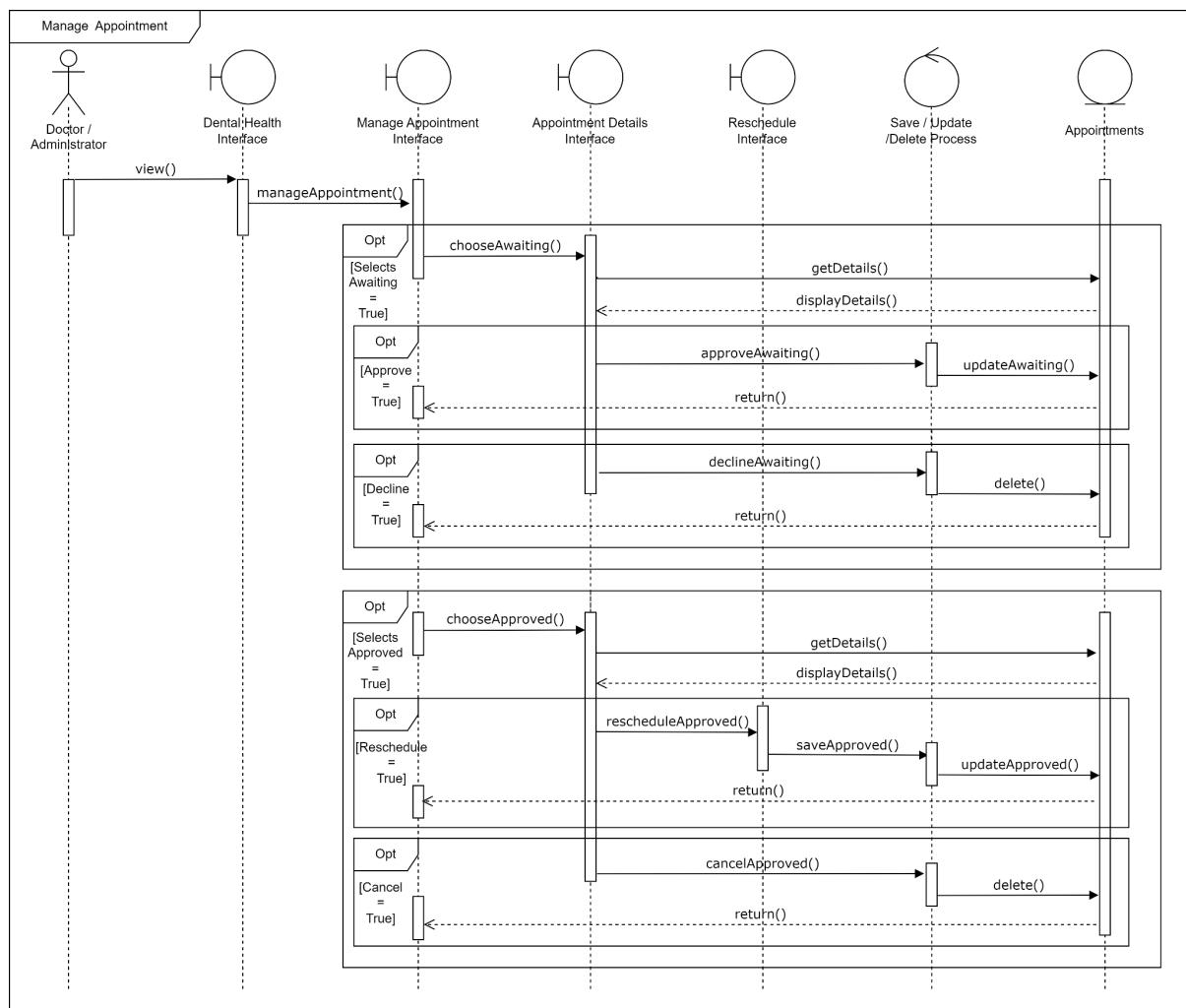


Figure 4.2.6.3: Sequence Diagram for <Set Up Dental Appointment>



4.2.7 P007: <Feedback and Reviews> Subsystem

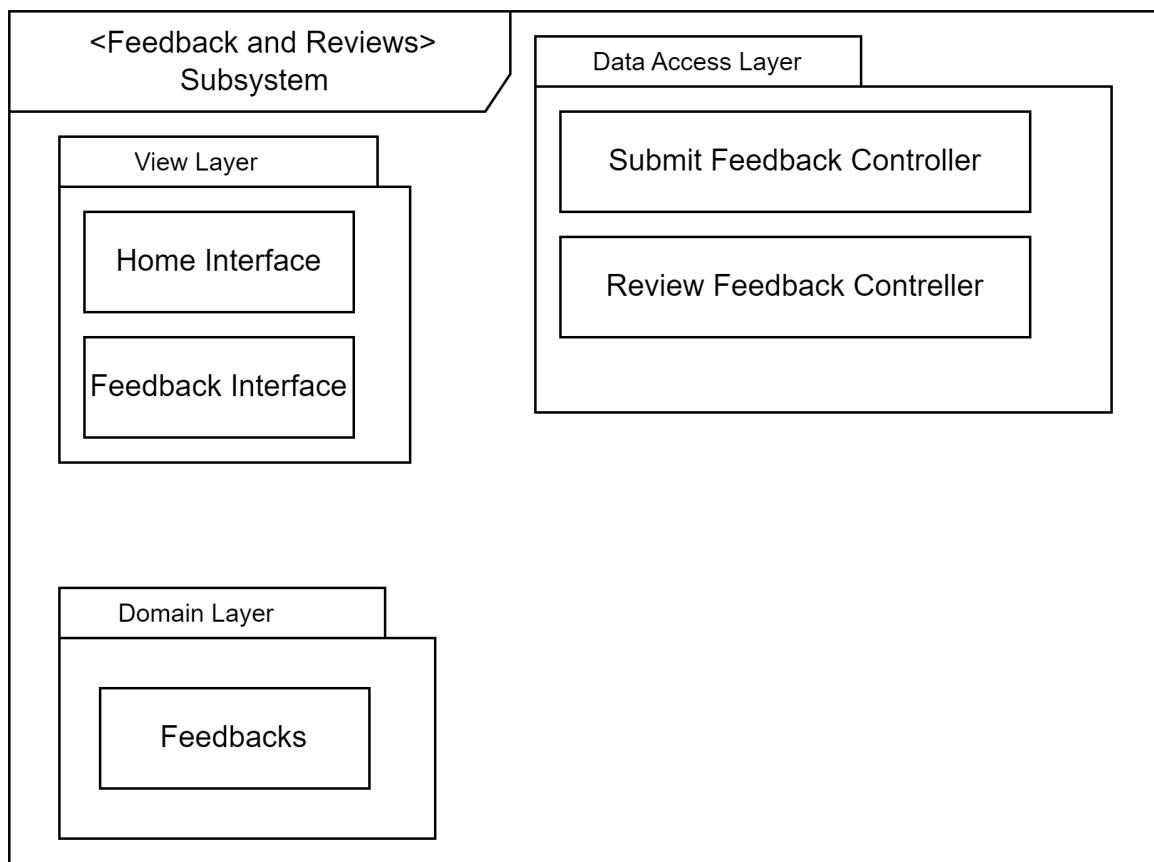


Figure 4.2.7.1: P007: <Feedback and Reviews> Subsystem

4.2.7.1 Class Diagram

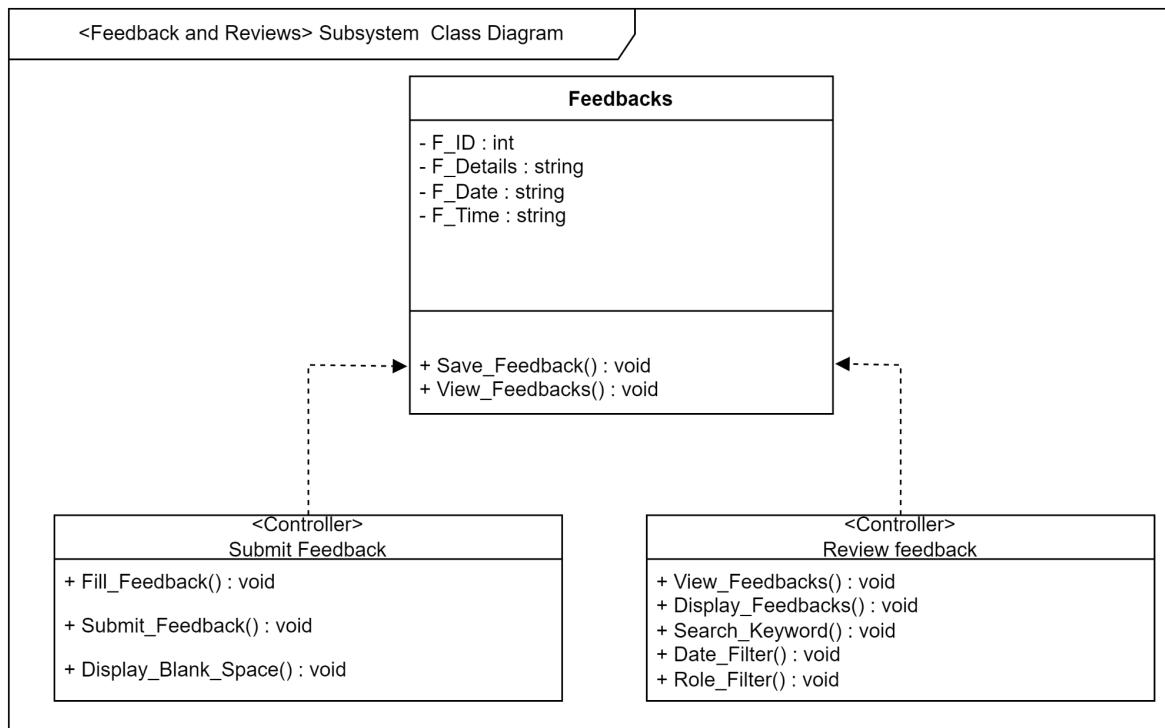


Figure 4.2.7.2: Class Diagram of P007: <Feedback and Reviews> Subsystem

| | |
|--------------------|--|
| Entity Name | Feedbacks |
| Method Name | Save_Feedback |
| Input | Submission of feedbacks |
| Output | Success of feedbacks |
| Algorithm | <ol style="list-style-type: none"> Start Fill in the feedback into the display blank space at the bottom of the homepage interface. Student and Doctors selects “Submit” button to submit feedback. System saved feedback. System displays a page with the message of submit successfully. End |

| | |
|--------------------|---|
| Entity Name | Feedbacks |
| Method Name | View_Feedbacks |
| Input | Search and filter keyword of feedbacks |
| Output | View needed feedbacks |
| Algorithm | <ol style="list-style-type: none"> Start |

- | | |
|--|---|
| | <ol style="list-style-type: none"> 2. Administrators tap the home page dashboard “Feedbacks” button. 3. Administrators tap and navigate to a new page to view a feedback list that provides feedback, role, time and date submitted. 4. Administrator type keywords to search or filter with date, time, role for the feedback. 5. Administrators can view the feedback they want. 6. end. |
|--|---|

4.2.7.2 Sequence Diagram

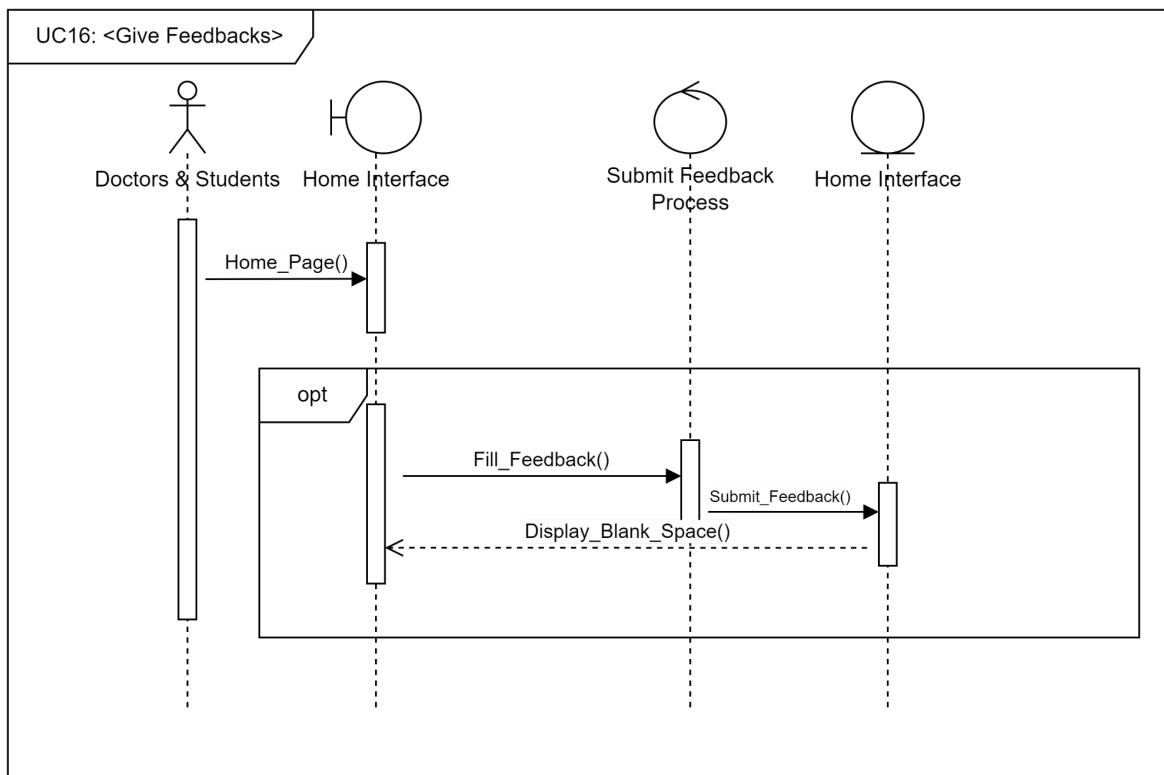


Figure 4.2.7.3: Sequence Diagram for <Give Feedbacks>

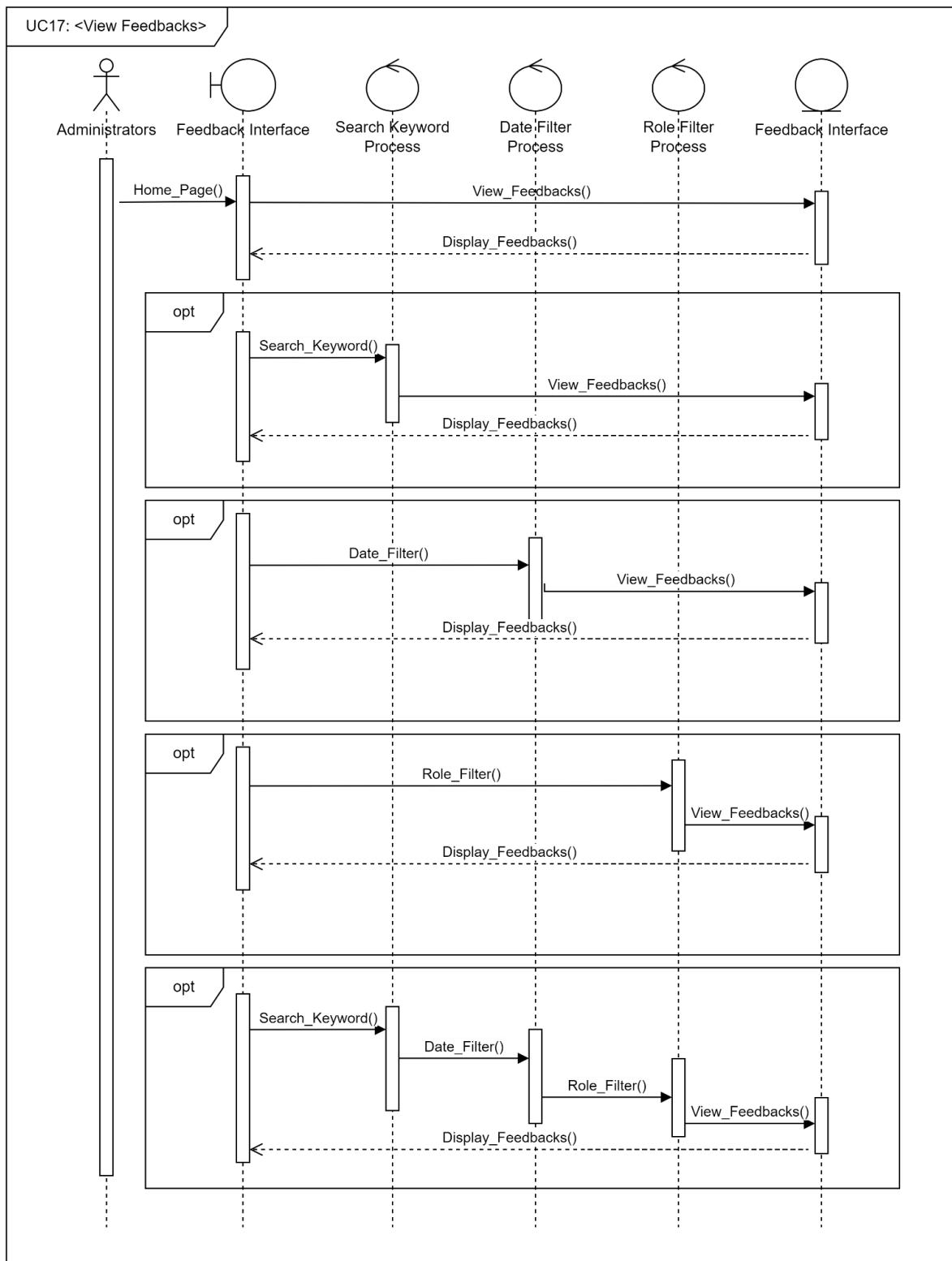


Figure 4.2.7.4: Sequence Diagram for <View Feedbacks>

5. Data Design

5.1 Data Description

Table 5.1: Description of Entities in the Database

| No. | Entity Name | Description |
|-----|----------------------|--|
| 1 | User | saves all information about user |
| 2 | Student | saves all information about student |
| 3 | Doctor | saves all information about doctor |
| 4 | Admin | saves all information about admin |
| 5 | Patient_Queue_Status | saves all information about patient queue status |
| 6 | SelfTest_Record | saves all information about self test records |
| 7 | Mental_Appointment | saves all information about mental appointment |
| 8 | Dental_Appointment | saves all information about dental appointment |
| 9 | Feedbacks | saves all information about feedbacks |
| 10 | Health_Tips | saves all information about health tips |

5.2 Data Dictionary

5.2.1 Entity: <User>

| Attribute Name | Type | Description |
|----------------|--------|--|
| U_Email | string | Email address of user |
| U_Password | string | Encrypted password of user |
| U_HP | string | Phone number of user |
| U_Name | string | Full name of user |
| U_Role | string | Role of user |
| U_Age | int | Age of user |
| U_ProfilePic | string | Profile picture of user |
| U_Gender | string | Gender of user |
| U_Weight | float | Weight of user |
| U_Height | float | Height of user |
| U_Systolic | int | Systolic blood pressure of user in mmHg |
| U_Diastolic | int | Diastolic blood pressure of user in mmHg |
| U_Medical | string | Medical report of user |
| U_Xray | string | X-ray report of user |

| | | |
|----------------|--------|---------------------------------|
| U_Urine | string | Urine test report of user |
| U_BMI_Result | float | BMI result of user |
| U_BMI_Category | string | BMI category of user |
| U_BP_Category | string | Blood pressure category of user |

5.2.2 Entity: <Student>

| Attribute Name | Type | Description |
|------------------|--------|---|
| S_MatricNo | string | Matric number of student |
| S_Email | string | Email address of student |
| S_HP | string | Phone number of student |
| S_Name | string | Name of student |
| U_Email | string | Email address of user (foreign key) |
| U_HP | string | Phone number of user (foreign key) |
| U_Name | string | Name of user (foreign key) |
| S_STS_Depression | string | Self test status of depression of student |
| S_STS_Anxiety | string | Self test status of anxiety of student |
| S_STS_Stress | string | Self test status of stress of student |
| S_STC_Depression | int | Self test score of depression of student |
| S_STC_Anxiety | int | Self test score of anxiety of student |
| S_STC_Stress | int | Self test score of stress of student |

5.2.3 Entity: <Doctor>

| Attribute Name | Type | Description |
|----------------|--------|-------------------------------------|
| D_StaffID | string | Staff ID of doctor |
| U_Email | string | Email address of user (foreign key) |
| U_HP | string | Phone number of user (foreign key) |
| U_Name | string | Name of user (foreign key) |
| D_Email | string | Email address of doctor |
| D_HP | string | Phone number of doctor |
| D_Name | string | Name of doctor |

5.2.4 Entity: <Administrator>

| Attribute Name | Type | Description |
|----------------|--------|-------------------------------------|
| A_StaffID | string | Staff ID of admin |
| U_Email | string | Email address of user (foreign key) |
| U_HP | string | Phone number of user (foreign key) |

| | | |
|---------|--------|--------------------------------|
| U_Name | string | Name of user (foreign key) |
| A_Email | string | Email address of administrator |
| A_HP | string | Phone number of administrator |
| A_Name | string | Name of administrator |

5.2.5 Entity: <Patient_Queue_Status >

| Attribute Name | Type | Description |
|----------------|--------|---|
| PQS_UPatient | int | Numbers of updated patient queue status |
| PQS_Time | string | Time of updated patient queue status |
| PQS_RTime | int | Estimated waiting time for walk-in patients |
| PQS_RPatient | int | Amount of remaining patients ahead |

5.2.6 Entity: <SelfTest_Record>

| Attribute Name | Type | Description |
|------------------|--------|---|
| ST_ID | int | ID of self test |
| U_Email | string | Email address of users who did self test |
| U_Name | string | Name of users who did self test |
| ST_Date | string | Date of completing self test |
| ST_Time | string | Time of completing self test |
| S_STC_Depression | int | Self test score of depression of student |
| S_STC_Anxiety | int | Self test score of anxiety of student |
| S_STC_Stress | int | Self test score of stress of student |
| S_STS_Depression | string | Self test status of depression of student |
| S_STS_Anxiety | string | Self test status of anxiety of student |
| S_STS_Stress | string | Self test status of stress of student |

5.2.7 Entity: <Mental_Appointment>

| Attribute Name | Type | Description |
|----------------|--------|---|
| MA_ID | int | ID of mental appointment |
| S_Name | string | Name of the student who made the appointment |
| S_MatricNo | string | Matric number of the student who made the appointment |
| MA_Date | string | Mental appointment date |
| MA_Time | string | Mental appointment time |
| MA_Status | string | Mental appointment status (Pending, Approved, Declined) |

5.2.8 Entity: <Dental_Appointment>

| Attribute Name | Type | Description |
|----------------|--------|--|
| DA_ID | int | ID of dental appointment |
| S_Name | string | Name of the student who made the appointment |
| S_MatricNo | string | Matric number of the student who made the appointment |
| DA_Date | string | Dental appointment date |
| DA_Time | string | Dental appointment time |
| DA_Status | string | Dental appointment status (Pending, Approved, Declined) |

5.2.9 Entity: <Feedbacks>

| Attribute Name | Type | Description |
|----------------|--------|----------------------|
| F_ID | int | ID of feedbacks |
| F_Details | string | Details of feedbacks |
| F_Date | string | Feedback date |
| F_Time | string | Feedback time |

5.2.10 Entity: <Health_Tips>

| Attribute Name | Type | Description |
|----------------|--------|----------------------------|
| HT_ID | int | ID of health tips |
| HT_Info | string | Information of health tips |
| HT_Pic | string | Picture of health tips |

6. User Interface Design

6.1 Overview of User Interface

The interface includes a general design for every user and a specified interface type for the dedicated users. First and foremost, the interface that is displayed to every single user is the sign up and login page where the users will dive into the system with their respective credentials. Subsequently, there's a user forgetting password and reset password pages in case the user has forgotten their password and would like to reset it. Moreover, after login to each account, there is a home page that can navigate to each main page of the system, which include functionality such as log-out, view patient remaining and estimated waiting time, user profile, mental health, dental health, health tips and feedback pages. For admin, they can edit the patient remaining in the Home page which can help students to save their time since they can avoid going to PKU at peak period.

Nevertheless, for the user profile page, there is an overview for your own profile and some navigation to other pages such as manage account page, medical check-up result page, blood pressure page, X-Ray report page, Urine Test Report page, Mental Health Report page and BMI Calculation page. For the Manage Account page, users are allowed to view their email address, change password and contact number if the user had changed their contact number and wish to change his/her current password to strengthen their respective password. For Admin and Doctor, they can also change their role through the Edit Role page in Manage Profile page to avoid the changing in role not updated in this mobile system. Nonetheless, there are some functions that can be used by users to ensure their current health status in the User Profile page that stated previously. For Admin, they are allowed to manage user profiles by Manage User Profile page for the purpose of viewing, creating and deleting user accounts to avoid any mistake in creating accounts for users.

In addition, through the Mental Health page, students are allowed to do a set of questions in the Self-Test page that aim to check the current mental status of students. If students get high marks in self-test, it will be recommended to make an appointment with PKU Mental Health Doctor through the Set Up Appointment page. There is also counseling information for those students that think they are stressful although their marks are not high enough for setting up appointments in the Counseling Information page. For students, there is also an Appointment Record page that can edit their appointment if PKU admin hasn't approved the appointment. Students are only allowed to edit appointments before the appointment is approved. For admin and doctor, they are allowed to view, approve, cancel

and reschedule the appointment via Appointment Record page, Pending Appointment page, Approved Appointment page and Reschedule Appointment page if the date and time does not fit with their schedule.

Next, for the Dental Health page, students are allowed to set up appointment and edit appointment via Set Up Appointment page, Appointment Record page and Edit Appointment Page. For add appointments, students can choose the date and time for the appointment that they wish to have through the Set Up Appointment page. This can reduce the workload of PKU to make dental appointments for each patient that walks-in to PKU physically. Furthermore, students are allowed to edit their dental appointment before the doctor approves through the Edit Appointment page. This can avoid students simply editing their appointment after the doctor approves. For admin and doctor, they are allowed to view, approve, cancel and reschedule the appointment via Appointment Record page, Pending Appointment page, Approved Appointment page and Reschedule Appointment page if the date and time does not fit with their schedule.

Other than that, there is a Health Tips page that aims to increase the health awareness among students. Students and doctors are allowed to view the Health tips via the Health Tips page and Health Tips Details page. However, admin not only can view the Health tips, but they can also edit and add new Health Tips via Create Health Tips and Edit Health Tips page. In fact, health tips will be approved by the doctor physically and uploaded to this mobile application by admin.

Last but not the least, students and doctors are allowed to give feedback for the system in the Home page to ensure the continuous improvement culture in this system. Through feedback, the admin can view it in the Feedback page and use feedback as an idea when updating the system and debugging for the system

6.2.1.1 General Interface

6.2 Screen Images

6.2.1 Authentication System



Figure 5.1: Welcome Page

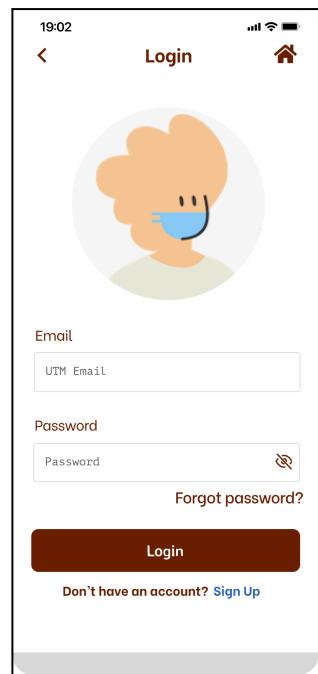


Figure 5.3: Login Page

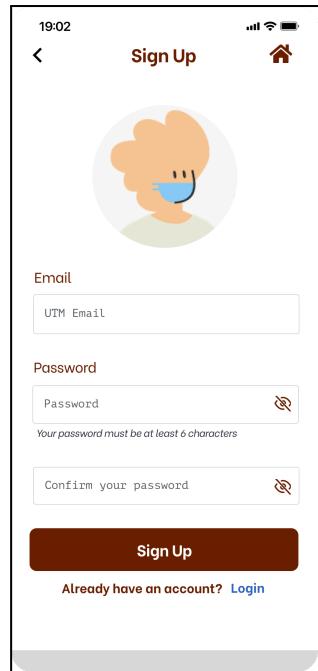


Figure 5.2: Sign Up Page

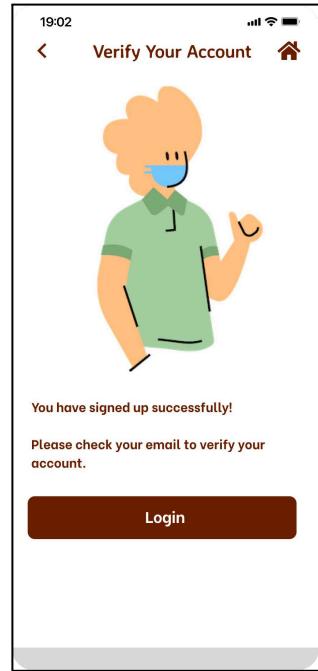


Figure 5.4: Successfully sign up Page

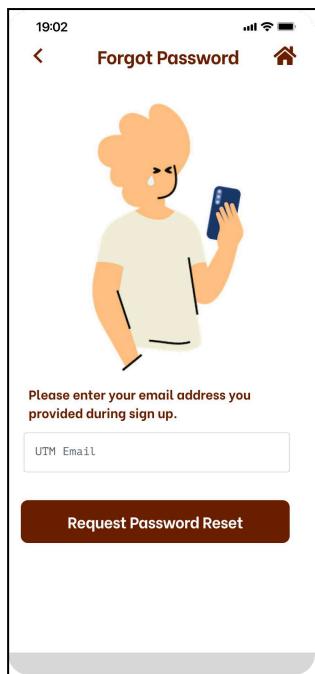


Figure 5.5: Forgot Password Page

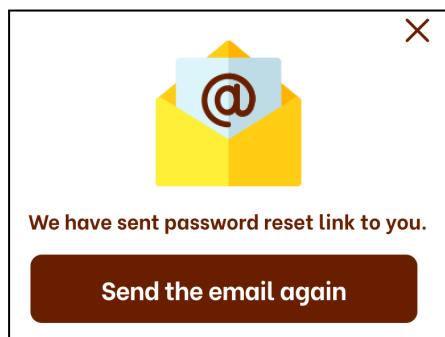


Figure 5.6: Pop-up for request password reset

6.2.2 Feedback and Reviews System

6.2.2.1 Student and Doctor

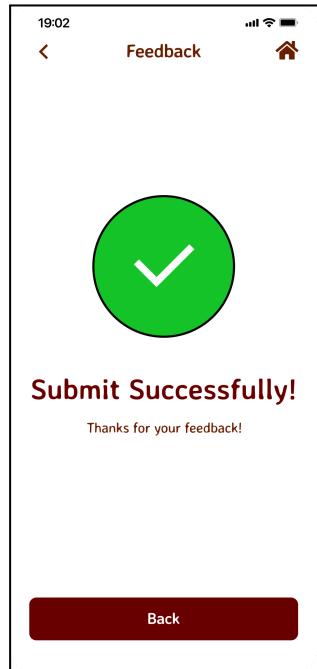


Figure 5.7: Feedback Submit Successfully Page

6.2.2.2 Admin

| Date | Time | Role | Feedback |
|------------|-------|---------|---|
| 02/05/2023 | 14:30 | Student | I think that the..... |
| 25/04/2023 | 09:34 | Doctor | I think that the..... |
| 20/04/2023 | 11:05 | Doctor | I think that the..... |
| 05/04/2023 | 23:45 | Student | I cannot apply appointment after |

Figure 5.8: Feedback View Page

6.2.3 Patient Queue Manage System

6.2.3.1 Student and Doctor

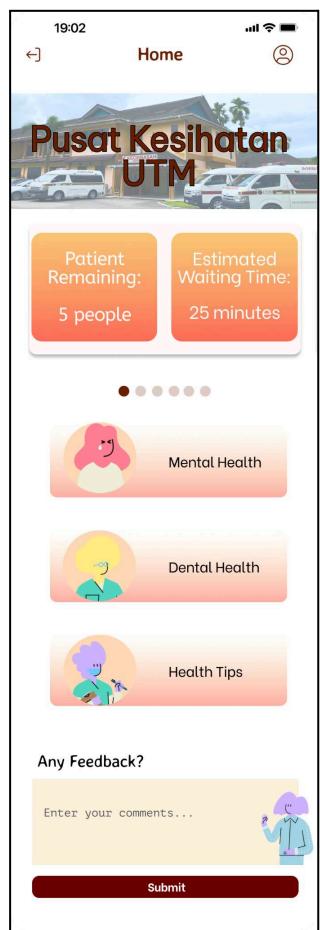


Figure 5.9: Home Page

6.2.3.2 Admin

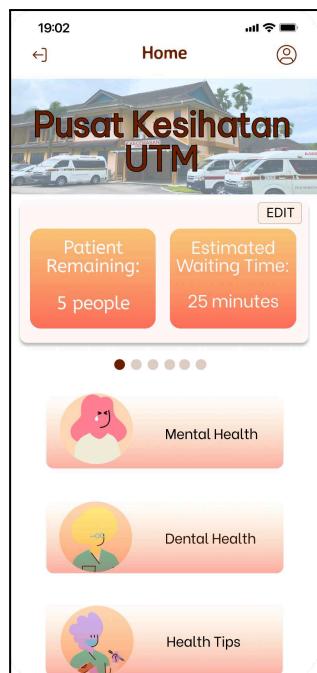


Figure 5.10: Edit Patient Remaining Page 1

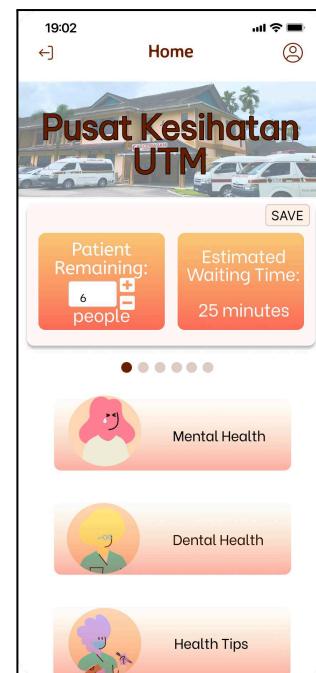


Figure 5.12: Edit Patient Remaining Page 3

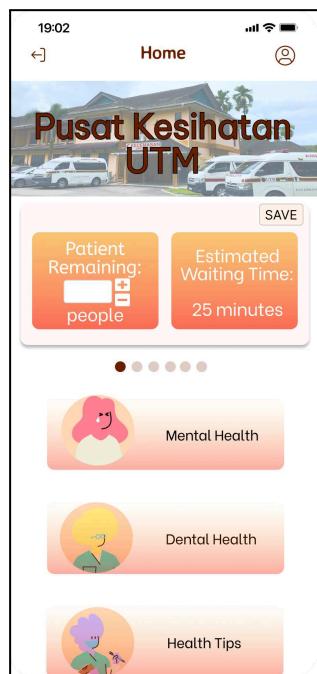


Figure 5.11: Edit Patient Remaining Page 2

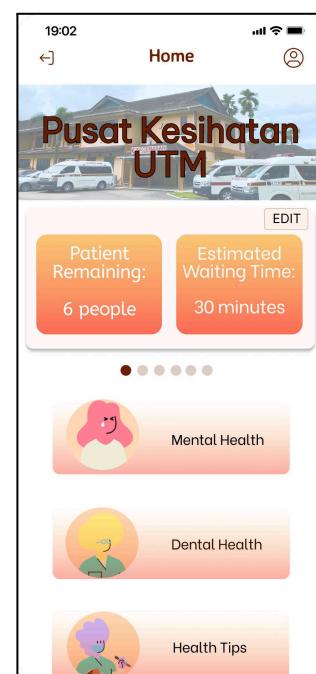


Figure 5.13: Save Patient Remaining Page

6.2.4 Account Management System

6.2.4.1 General Interface

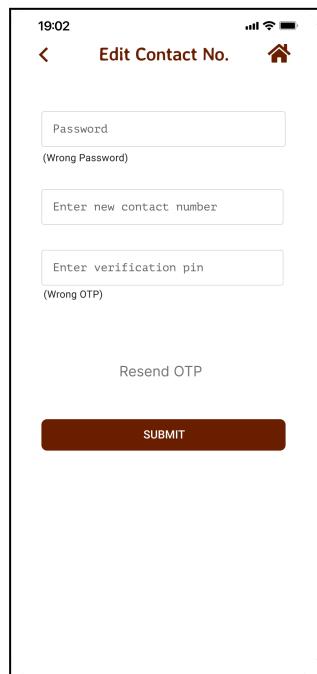


Figure 5.14: Edit Contact Number Page

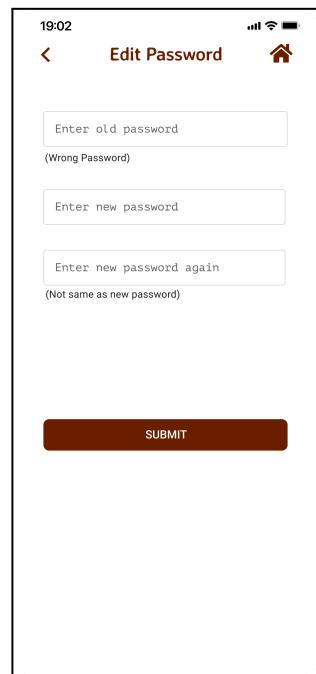


Figure 5.16: Edit Password Page

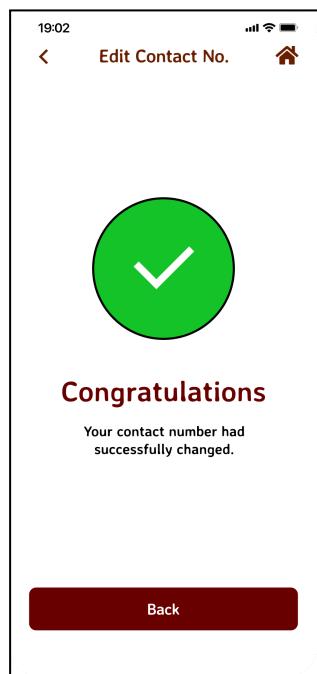


Figure 5.15: Contact Number Saved Page

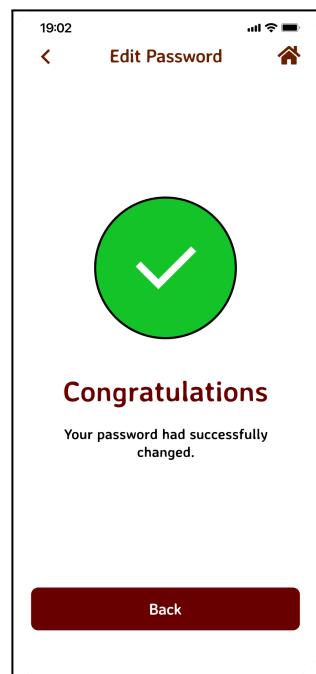


Figure 5.17: Password Saved Page

BMI Calculation

Gender : Gender

Age : 21

Weight (in kg) : 45.3

Height (in cm) : 170

BMI: 19.0

Category: Normal

Body Mass Index

| | |
|-----|-------------|
| 130 | Obese |
| 120 | Overweight |
| 110 | Normal |
| 100 | Underweight |
| 90 | |
| 80 | |
| 70 | |
| 60 | |
| 50 | |
| 40 | |

Height in cm

Kindly reminder
Please drink more water, maintain healthy lifestyle and sleep more to be more healthy and maintain current BMI.

Figure 5.18: BMI Calculation Page

Medical Check-up

6 June 2023 2:30 p.m.

6 June 2023 2:30 p.m.

6 June 2023 2:30 p.m.

+

Figure 5.20: Medical Check-up Main Page

Blood Pressure

Systolic mmHg : 110

Diastolic mmHg : 70

Category: Normal

Blood Pressure Classification

| BLOOD PRESSURE CATEGORY | SYSTOLIC mm Hg (upper number) | DIASTOLIC mm Hg (lower number) | |
|---|-------------------------------|--------------------------------|-----------------|
| NORMAL | LESS THAN 120 | and | LESS THAN 80 |
| ELEVATED | 120 - 129 | and | LESS THAN 80 |
| HIGH BLOOD PRESSURE (HYPERTENSION) STAGE 1 | 130 - 139 | or | 80 - 89 |
| HIGH BLOOD PRESSURE (HYPERTENSION) STAGE 2 | 140 OR HIGHER | or | 90 OR HIGHER |
| HYPERTENSIVE CRISIS (consult your doctor immediately) | HIGHER THAN 180 | and/or | HIGHER THAN 120 |

Kindly reminder
Please reduce the amount of salt you eat and have a generally healthy diet to maintain your current blood pressure category.

Figure 5.19: Blood Pressure Page

Date

1902

FOREIGN SERVICE OF THE PHILIPPINES
MEDICAL EXAMINATION OF VISA APPLICANTS

Place: Date: 19.06.2023

At the request of the Philippine Consulate General in _____, Country: _____

I certify that on the above date I was examined and that under the Philippine Immigration Regulations I am not suffering from any disease or disability which would render me liable to be deported.

Name: ANDI THAHA Age: 21 Sex: M Citizenship: INDONESIA

Class A: DANGEROUS CONTAGIOUS DISEASE

Class B: DANGEROUS NON-CONTAGIOUS DISEASE

Class C: DANGEROUS CONDITION

MEDICAL RECORDS

- Patient Medical History
- Sightings by Physician
- Present Complaints (e.g., pain, fever, etc.)
- Laboratory Examinations (Attach laboratory reports)

EXAMINING PHYSICIAN: Signature: ADDRESS: Date: 19.06.2023

SAVE

Figure 5.21: Add Medical Check-up Report Page

19:02

6 June 2023

FOREIGN SERVICE OF THE PHILIPPINES
MEDICAL EXAMINATION OF VISA APPLICANTS

Place: Date: 6 JUN 2023
At the request of the Philippine Consul at: City: Country: Photo (2x2)

I certify that on the above date I was examined.

Name: ANDI THAHL Age: 37 Sex: M Citizenship: ENGL-IND-THA

and then under the Philippine Immigration Regulation or applicable laws for classified as follows: (check all that apply)

Class A DANGEROUS CONTAGIOUS DISEASE

Chronic Gonorrhea, Gonococcal Inflammation, Syphilis, Lymphogranuloma Venereum, Leptospiral Infection, Tuberculosis (Active), Latent, Latent Tuberculosis, Latent Tuberculosis (Active), Mental Retardation, Mental Deficiency, Insanity, Psychotic Condition, One or More Attacks of Insanity, Anti-Social Personality, Mental Defects, Chronic Alcoholism, Chronic Drug Addiction.

Class B MEDICAL DISORDERS

Persons having physical defects, disease or disability which may impair their health or future ability to earn a living and to make a normal life for themselves.

Class C MINOR CONDITIONS

1. Past Medical History: Normal Case
2. Significant Physical Examination: Normal Case
3. Chest X-ray Report (Age: 11 yrs. and above): Normal Case
4. Laboratory Examinations (Attach laboratory reports):
a. Blood Smear: RBC/Hb/VFC: (Age: 15 yrs. and above)
b. Urine Test: (Age: 1 yr. and above)
c. Stool (Ova and Parasite): (Age: 1 yr. and above)
d. Other examinations if necessary: Not physically and mentally defective or diseased

EXAMINING PHYSICIAN: *Dr. A. S. M.* ADDRESS: *12345678901234567890*

Figure 5.22: View Medical Check-up Report Page

19:02

Date: 6 JUN 2023

FOREIGN SERVICE OF THE PHILIPPINES
MEDICAL EXAMINATION OF VISA APPLICANTS

Place: Date: 6 JUN 2023
At the request of the Philippine Consul at: City: Country: Photo (2x2)

I certify that on the above date I was examined.

Name: ANDI THAHL Age: 37 Sex: M Citizenship: ENGL-IND-THA

and then under the Philippine Immigration Regulation or applicable laws for classified as follows: (check all that apply)

Class A DANGEROUS CONTAGIOUS DISEASE

Chronic Gonorrhea, Gonococcal Inflammation, Syphilis, Lymphogranuloma Venereum, Leptospiral Infection, Tuberculosis (Active), Latent, Latent Tuberculosis, Latent Tuberculosis (Active), Mental Retardation, Mental Deficiency, Insanity, Psychotic Condition, One or more Attacks of Insanity, Anti-Social Personality, Mental Defects, Chronic Alcoholism, Chronic Drug Addiction.

Class B MEDICAL DISORDERS

Persons having physical defects, disease or disability which may impair their health or future ability to earn a living and to make a normal life for themselves.

Class C MINOR CONDITIONS

1. Past Medical History: Normal Case
2. Significant Physical Examination: Normal Case
3. Chest X-ray Report (Age: 11 yrs. and above): Normal Case
4. Laboratory Examinations (Attach laboratory reports):
a. Blood Smear: RBC/Hb/VFC: (Age: 15 yrs. and above)
b. Urine Test: (Age: 1 yr. and above)
c. Stool (Ova and Parasite): (Age: 1 yr. and above)
d. Other examinations if necessary: Not physically and mentally defective or diseased

EXAMINING PHYSICIAN: *Dr. A. S. M.* ADDRESS: *12345678901234567890*

SAVE

Figure 5.24: Add X-Ray Report Page

19:02

X-Ray Report

6 June 2023 2:30 p.m.

6 June 2023 2:30 p.m.

6 June 2023 2:30 p.m.

+

Figure 5.23: X-Ray Report Main Page

19:02

6 June 2023

FOREIGN SERVICE OF THE PHILIPPINES
MEDICAL EXAMINATION OF VISA APPLICANTS

Place: Date: 6 JUN 2023
At the request of the Philippine Consul at: City: Country: Photo (2x2)

I certify that on the above date I was examined.

Name: ANDI THAHL Age: 37 Sex: M Citizenship: ENGL-IND-THA

and then under the Philippine Immigration Regulation or applicable laws for classified as follows: (check all that apply)

Class A DANGEROUS CONTAGIOUS DISEASE

Chronic Gonorrhea, Gonococcal Inflammation, Syphilis, Lymphogranuloma Venereum, Leptospiral Infection, Tuberculosis (Active), Latent, Latent Tuberculosis, Latent Tuberculosis (Active), SERIOUS MENTAL DISORDER

Class B MEDICAL DISORDERS

Persons having physical defects, disease or disability which may impair their health or future ability to earn a living and to make a normal life for themselves.

Class C MINOR CONDITIONS

1. Past Medical History: Normal Case
2. Significant Physical Examination: Normal Case
3. Chest X-ray Report (Age: 11 yrs. and above): Normal Case
4. Laboratory Examinations (Attach laboratory reports):
a. Blood Smear: RBC/Hb/VFC: (Age: 15 yrs. and above)
b. Urine Test: (Age: 1 yr. and above)
c. Stool (Ova and Parasite): (Age: 1 yr. and above)
d. Other examinations if necessary: Not physically and mentally defective or diseased

EXAMINING PHYSICIAN: *Dr. A. S. M.* ADDRESS: *12345678901234567890*

Figure 5.25: View X-Ray Report Page

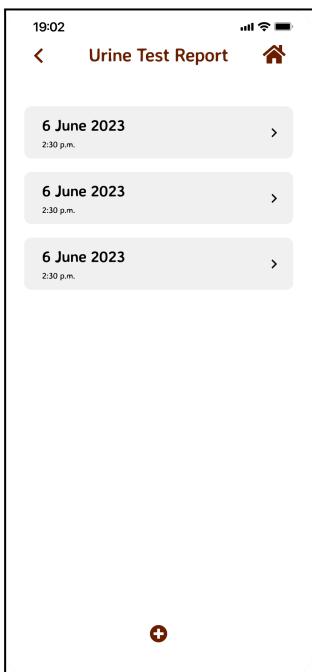


Figure 5.26: Urine Test Report Main Page

| | |
|---|--------------------|
| FOREIGN SERVICE OF THE PHILIPPINES | |
| MEDICAL EXAMINATION OF VISA APPLICANTS | |
| Place | Date (in 12/12/12) |
| All the request of the Philippine Consul at | City Country |
| Photo (2x2) | |
| I certify that the above date I was examined and that under the Philippine Immigration Regulation the applicant should be classified as follows: (Mark the appropriate class) | |
| Name: ANDI THAHA Age: 39 Sex: M Citizenship: INDONESIAN Photo (2x2): Class A DANGEROUS CONTAGIOUS DISEASE Chancroid, Gonorrhoea, Granuloma Inguinale, Leprosy (Infectious), Lymphogranuloma Venereum,梅毒 (Infectious), Plague, Smallpox, Typhus, Syphilis Infectious Stage, Tuberculosis (Active) Class B DISEASES WHICH MAY BE INFECTIVE OR CONTAGIOUS Mental Retardation, Mental Deficiency, Inanity, Insanity, Epilepsy, Seizure, Epileptic Seizure, Healthy, Anti-Social Personality, Mental Defect, Chronic Alcoholism, Chronic Drug Addiction Class C OTHER DISEASES Persons having physical defects, disease or deformities which may be disfiguring or which will impair their ability to earn a living or to make them likely to be a public charge. Minor Conditions: Not physically and mentally defective or diseased | |
| MEDICAL RECORDS | |
| 1. Patient Medical History <i>Normal Case</i> 2. Significant Physical Examination <i>Normal Case</i> 3. Chest X-ray Report (Age 11 yrs. and above) <i>Normal Case</i> 4. Laboratory Examination (Attach laboratory report) a. Blood Sediment: RBC/DL (Age: 15 yrs. and above) b. Urine: Urine Analysis (Age: 15 yrs. and above) c. Stool (One and Female) (Age: 1 yr. and above) d. Other examinations if needed <i>Normal Case</i> e. Not physically and mentally defective or diseased | |
| EXAMINING PHYSICIAN: <i>Andrea</i> ADDRESS: <i>Brgy. Poblacion, San Jose, Batangas</i> | |

Figure 5.28: View Urine Test Report Page

| | |
|---|--------------------|
| FOREIGN SERVICE OF THE PHILIPPINES | |
| MEDICAL EXAMINATION OF VISA APPLICANTS | |
| Place | Date (in 12/12/12) |
| All the request of the Philippine Consul at | City Country |
| Photo (2x2) | |
| I certify that the above date I was examined and that under the Philippine Immigration Regulation the applicant should be classified as follows: (Mark the appropriate class) | |
| Name: ANDI THAHA Age: 39 Sex: M Citizenship: INDONESIAN Photo (2x2): Class A DANGEROUS CONTAGIOUS DISEASE Chancroid, Gonorrhoea, Granuloma Inguinale, Leprosy (Infectious), Lymphogranuloma Venereum,梅毒 (Infectious), Plague, Smallpox, Typhus, Syphilis Infectious Stage, Tuberculosis (Active) Class B DISEASES WHICH MAY BE INFECTIVE OR CONTAGIOUS Mental Retardation, Mental Deficiency, Inanity, Insanity, Epilepsy, Seizure, Epileptic Seizure, Healthy, Anti-Social Personality, Mental Defect, Chronic Alcoholism, Chronic Drug Addiction Class C OTHER DISEASES Persons having physical defects, disease or deformities which may be disfiguring or which will impair their ability to earn a living or to make them likely to be a public charge. Minor Conditions: Not physically and mentally defective or diseased | |
| MEDICAL RECORDS | |
| 1. Patient Medical History <i>Normal Case</i> 2. Significant Physical Examination <i>Normal Case</i> 3. Chest X-ray Report (Age 11 yrs. and above) <i>Normal Case</i> 4. Laboratory Examination (Attach laboratory report) a. Blood Sediment: RBC/DL (Age: 15 yrs. and above) b. Urine: Urine Analysis (Age: 15 yrs. and above) c. Stool (One and Female) (Age: 1 yr. and above) d. Other examinations if needed <i>Normal Case</i> e. Not physically and mentally defective or diseased | |
| EXAMINING PHYSICIAN: <i>Andrea</i> ADDRESS: <i>Brgy. Poblacion, San Jose, Batangas</i> | |

Figure 5.27: Add Urine Test Report Page

6.2.4.2 Student Interface

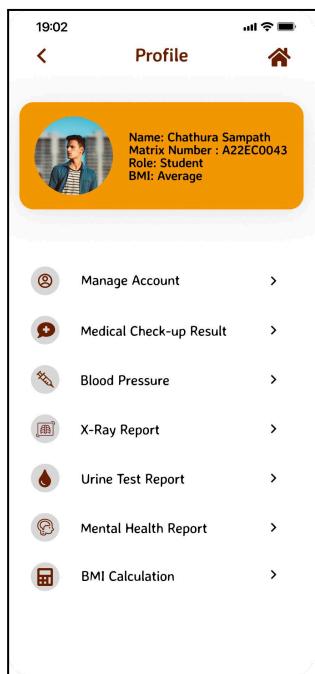


Figure 5.29: Student User Profile Page



Figure 5.31: Mental Health Report Main Page

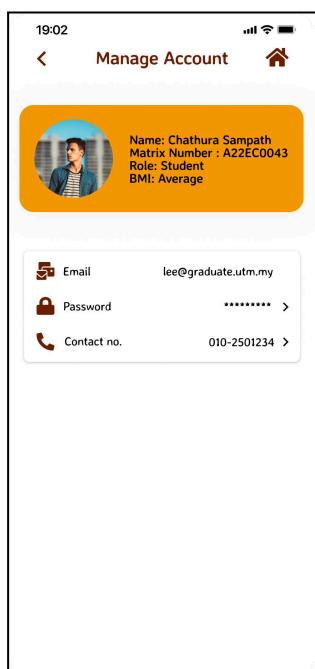


Figure 5.30: Student Edit User Profile Page

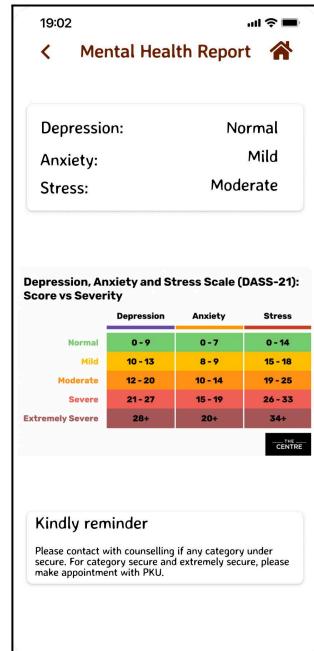


Figure 5.32: Mental Health Report Result Page

6.2.4.3 Doctor Interface

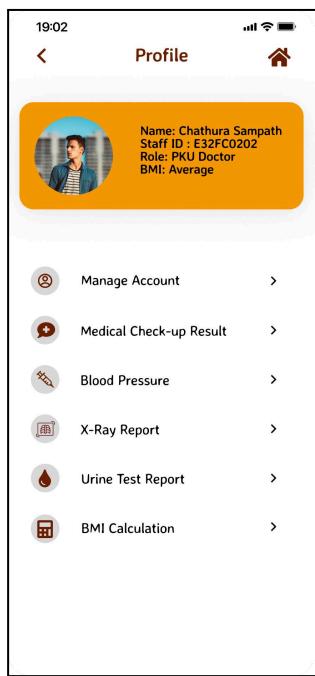


Figure 5.33: Doctor User Profile Page

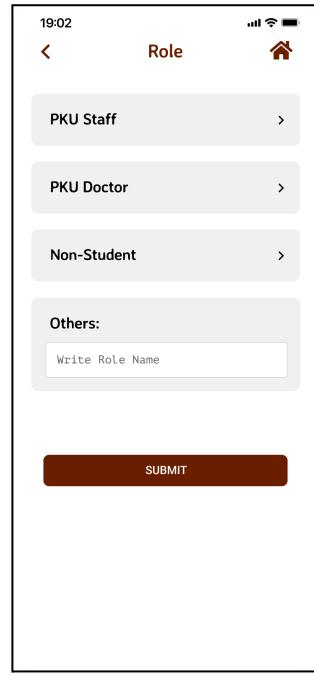


Figure 5.35: Edit Role Page

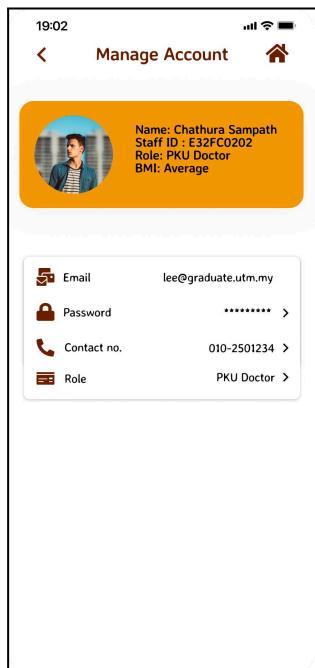


Figure 5.34: Doctor Edit User Profile Page



Figure 5.36: Role Saved Page

6.2.4.4 Admin Interface

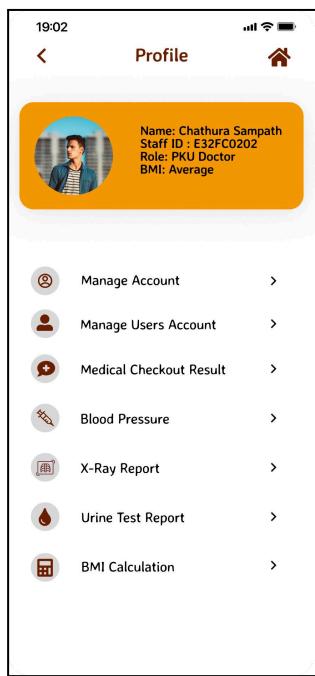


Figure 5.37: Admin User Profile Page

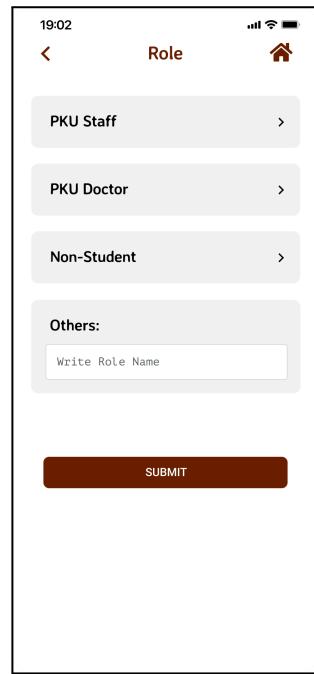


Figure 5.39: Edit Role Page



Figure 5.38: Admin Edit User Profile Page



Figure 5.40: Role Saved Page

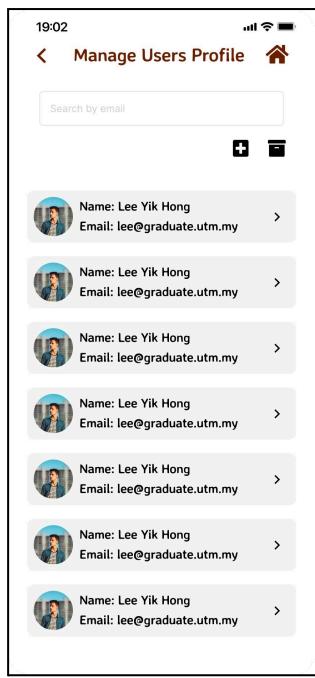


Figure 5.41: Manage Users Profile Page

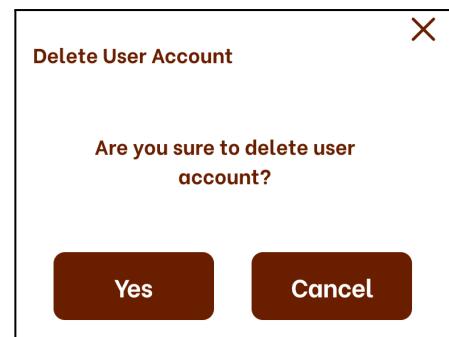


Figure 5.43: Delete User Account Pop-up

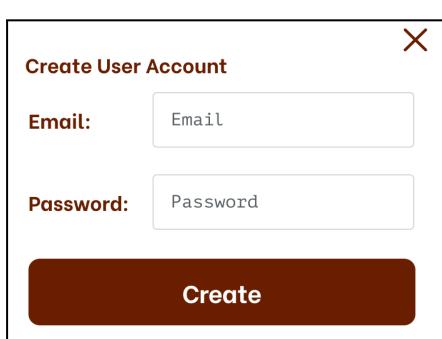


Figure 5.42: Create User Account Pop-up

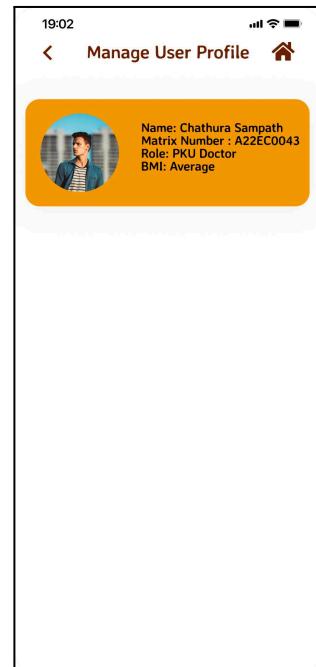


Figure 5.44: View Users Profile Page

6.2.5 Health Tips System

6.2.5.1 Student and Doctor Interface



Figure 5.45: Health Tips Main Page

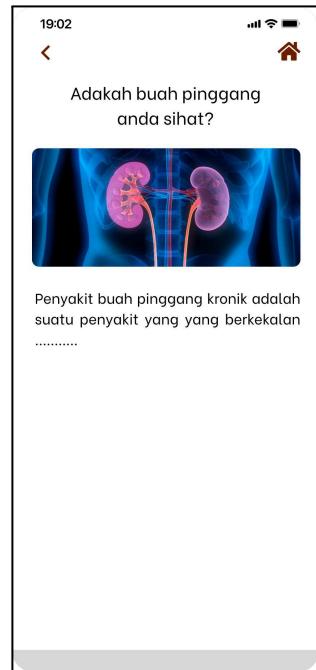


Figure 5.47: Health Tips Details Page 2

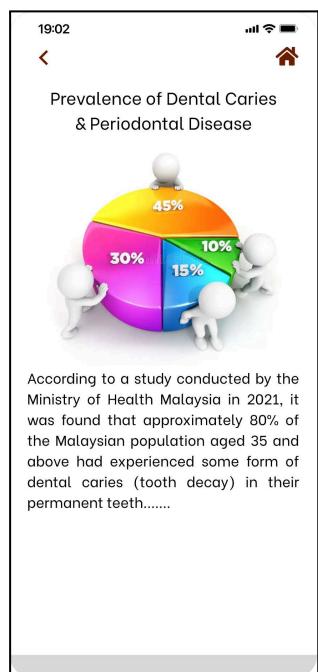


Figure 5.46: Health Tips Details Page 1

6.2.5.2 Admin Interface

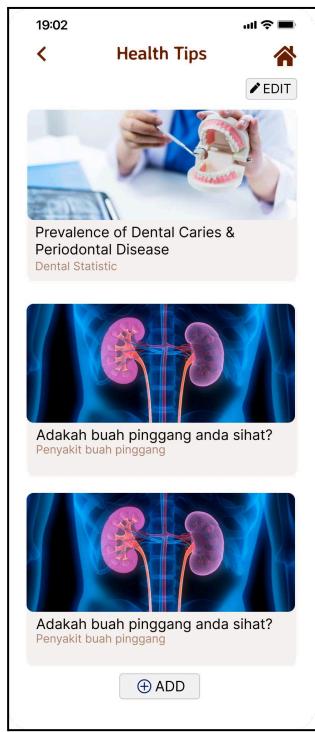


Figure 5.48: Health Tips Edit Page

This screenshot shows the 'Add Health Tips' page. At the top, there is a header with a back arrow, the title 'Add Health Tips', and a home icon. Below the header, there is a form with fields for 'Title' and 'Brief Description'. There is also a placeholder image area with a 'Change Photo' button. At the bottom right of the screen is a 'SAVE' button.

Figure 5.50: Add Health Tips Page



Figure 5.49: Health Tips Save Page

This screenshot shows the 'Edit Health Tips' page for the first item. At the top, there is a header with a back arrow, the title 'Please enter the details', and a home icon. Below the header, there is a form with fields for 'Title' (set to 'Prevalence of Dental Caries & Periodontal Disease') and 'Brief Description' (set to 'Dental Statistic'). There is also a placeholder image area with a 'Change Photo' button. A large image of a dental professional working on a dental model is displayed above the form. Below the image, there is a pie chart titled 'Prevalence of Dental Caries & Periodontal Disease' with the following data:

| Category | Percentage |
|----------|------------|
| 45% | 30% |
| 10% | 15% |

At the bottom of the screen is a text block: 'According to a study conducted by the Ministry of Health Malaysia in 2021, it was found that approximately 80% of the Malaysian population aged 35 and above had experienced some form of dental caries (tooth decay) in their permanent teeth.....'.

Figure 5.51: Edit Health Tips Page 1

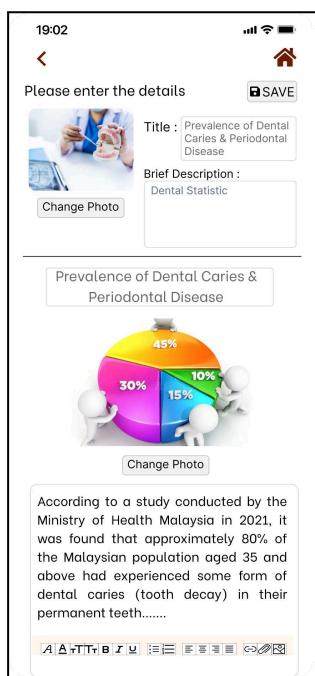


Figure 5.52: Save Health Tips Page 1



Figure 5.54: Save Health Tips Page 2



Figure 5.53: Edit Health Tips Page 2

6.2.6 Mental Health Support System

6.2.6.1 Student Interface

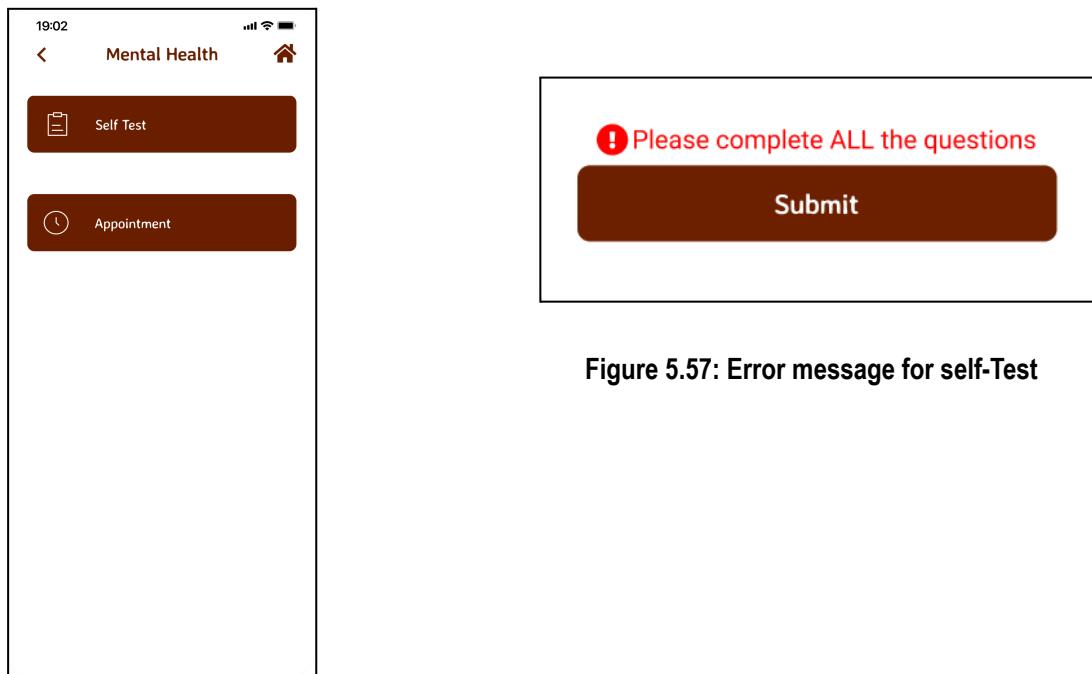


Figure 5.57: Error message for self-Test

Figure 5.55: Mental Health Main Page

The image consists of two screenshots of a mobile application. The left screenshot, labeled 'Figure 5.56: Self-Test Page', shows the 'Self Test' page with a timestamp of 19:02. It includes an 'Instruction' section with a detailed description of the DASS-21 scale. Below the instruction are four numbered questions, each with a list of response options from 'ALMOST ALWAYS' to 'NEVER'. The right screenshot, labeled 'Figure 5.58: Counseling Information Page', shows the 'Counselling Information' page with a timestamp of 19:02. It displays a summary of the test score: 'Your Test Score Are : D A S' with values '13(Severe)', '10(Extreme Severe)', and '14(Severe)'. It also includes a table of DASS-21 Scoring ranges for Normal, Mild, Moderate, and Severe levels across Depression, Anxiety, and Stress dimensions. The page also contains contact information and an 'Apply Appointment' button.

Figure 5.58: Counseling Information Page

Figure 5.56: Self-Test Page

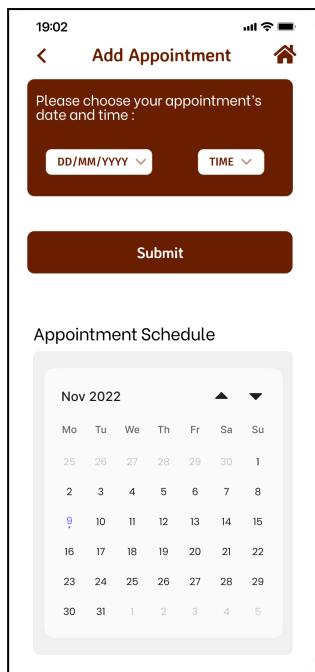


Figure 5.59: Add Mental Health Appointment Page

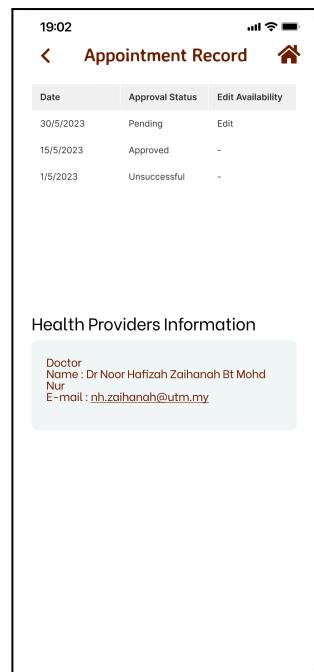


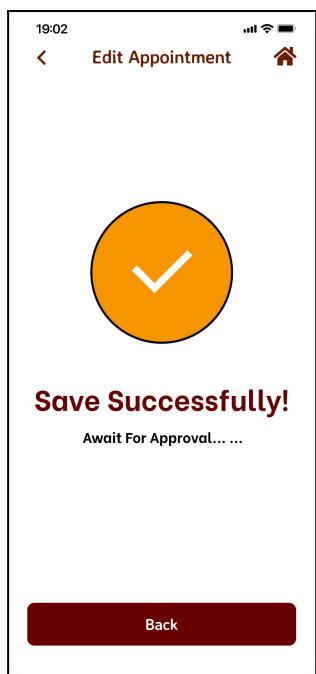
Figure 5.61: Mental Health Appointment Record Page



Figure 5.60: Mental Health Submit Successfully Page



Figure 5.62: Mental Health Edit Appointment Page



**Figure 5.63: Mental Health Edit
Appointment Successfully Page**

6.2.6.2 Doctor and Admin Interface

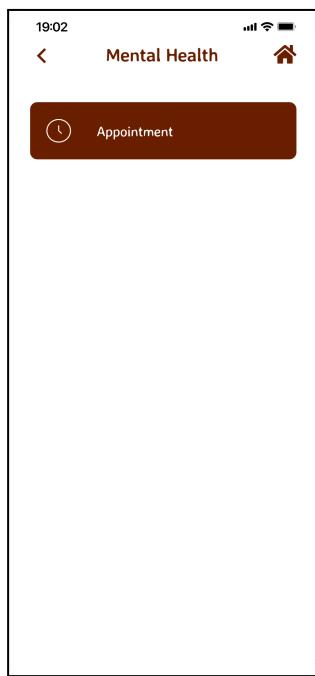


Figure 5.64: Mental Health Main Page

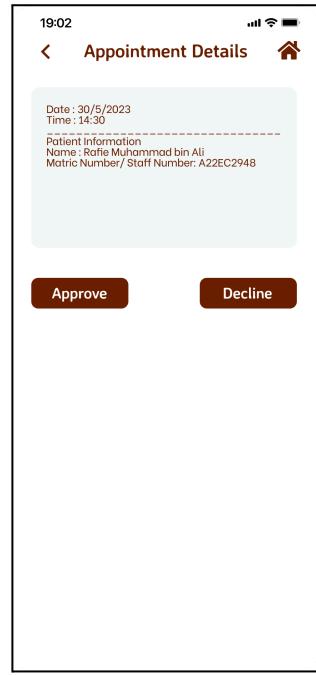


Figure 5.66: Pending Appointment Page 1

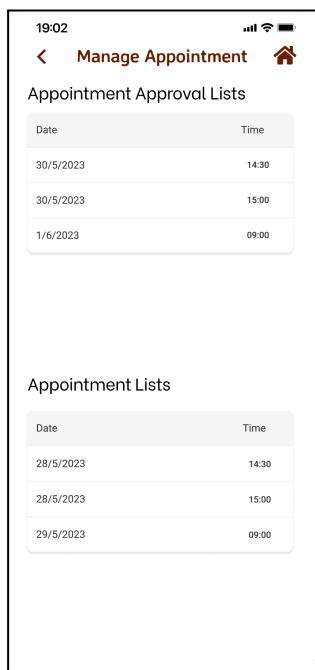


Figure 5.65: Manage Mental Health

Appointment Main Page

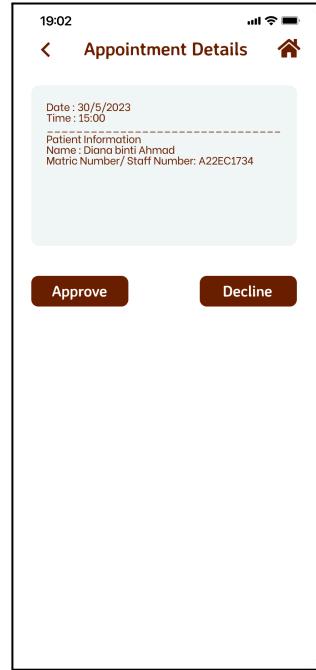


Figure 5.67: Pending Appointment Page 2

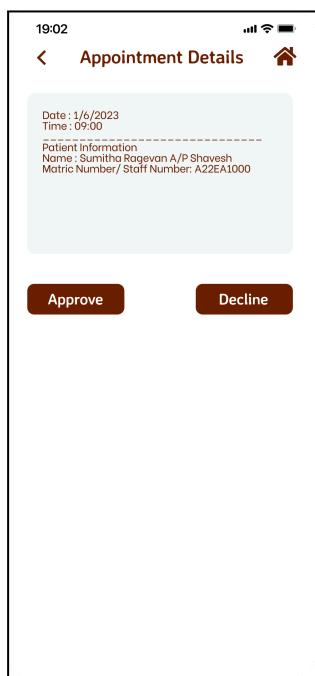


Figure 5.68: Pending Appointment Page 3

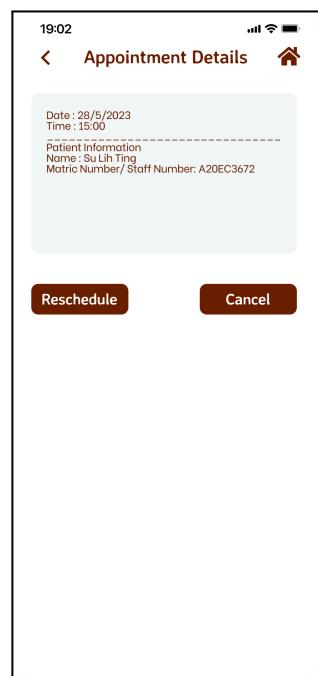


Figure 5.70: Approved Appointment Details Page 2

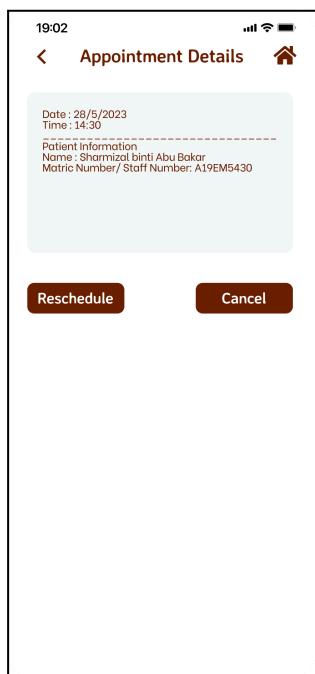


Figure 5.69: Approved Appointment Details Page 1

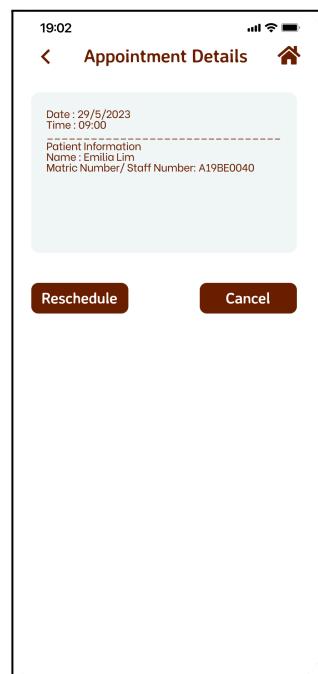


Figure 5.71: Approved Appointment Details Page 3

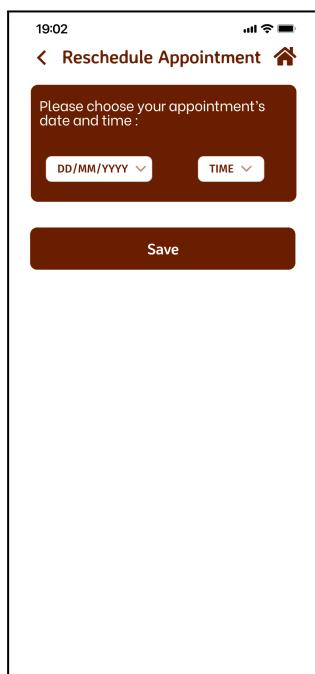


Figure 5.72: Reschedule Appointment Page

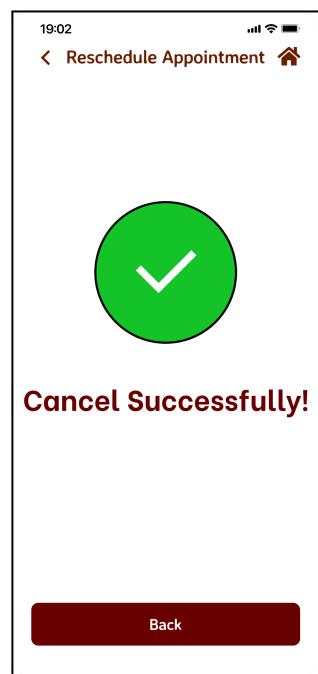


Figure 5.74: Reschedule Appointment
Canceled Page



Figure 5.73: Reschedule Appointment
Saved Page

6.2.7 Dental Health Appointment System

6.2.7.1 Student Interface

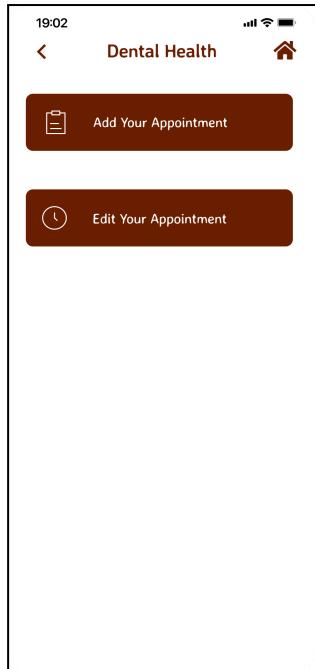


Figure 5.75: Dental Health Main Page

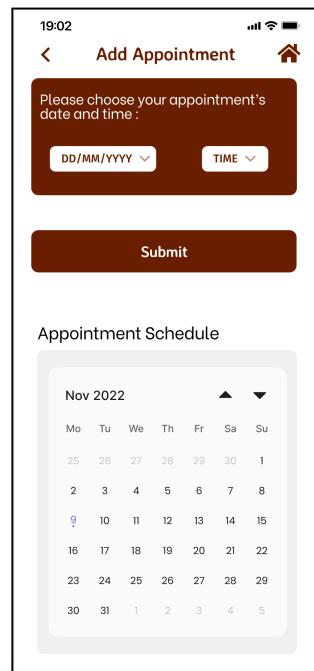


Figure 5.76: Add Dental Health Appointment Page

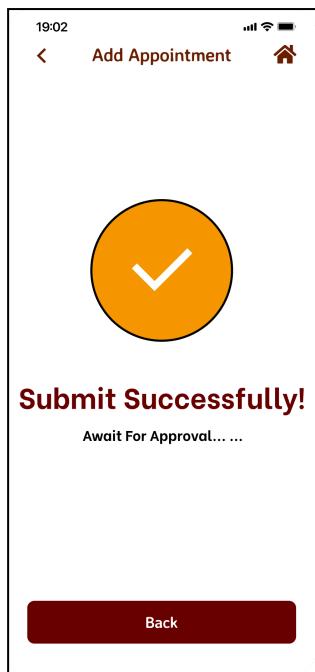


Figure 5.77: Dental Health
Appointment Submitted Page

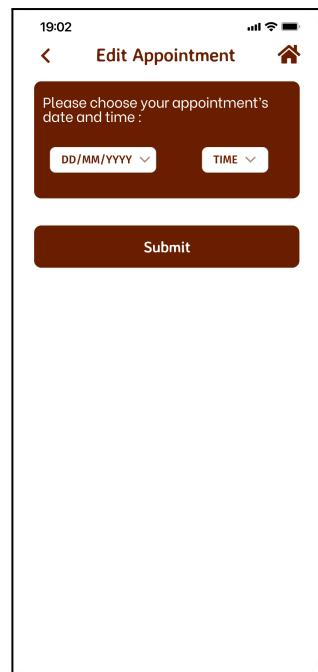


Figure 5.79: Edit Dental Health
Appointment Page

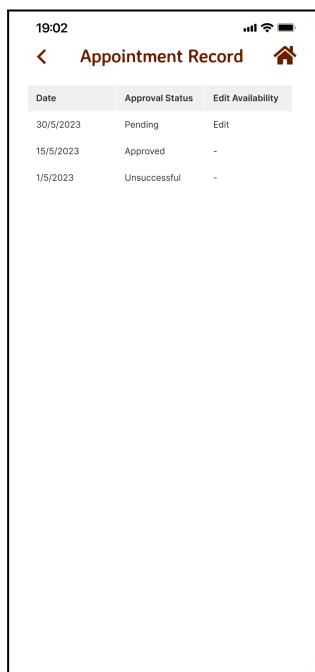


Figure 5.78: Dental Health
Appointment Record Page



Figure 5.80: Edit Dental Health
Appointment Saved Page

6.2.7.2 Admin and Doctor Interface

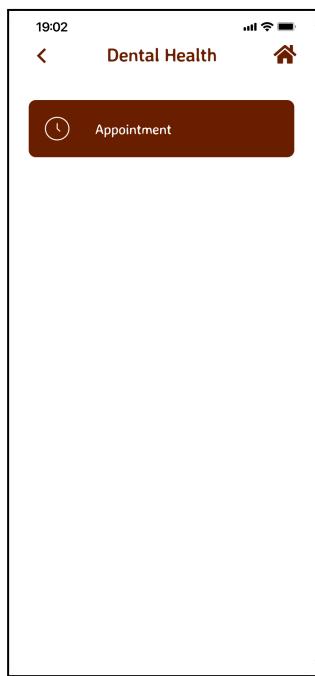


Figure 5.81: Dental Health Main Page

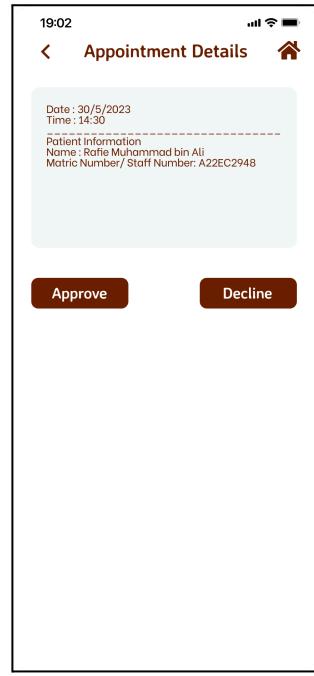


Figure 5.83: Pending Appointment Page 1

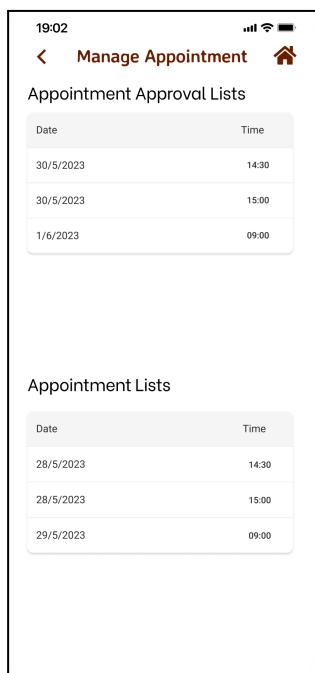


Figure 5.82: Manage Dental Health
Appointment Main Page

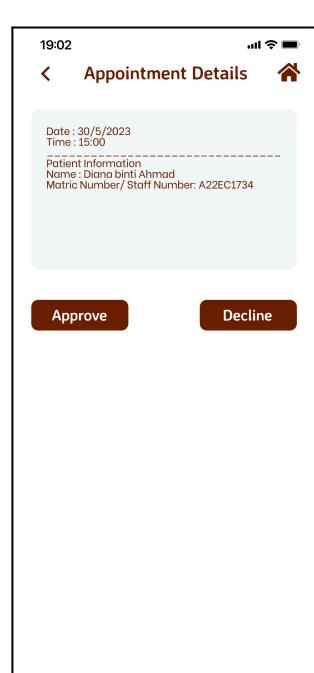


Figure 5.84: Pending Appointment Page 2

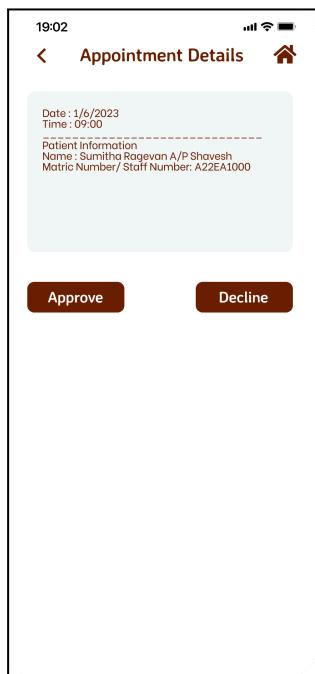


Figure 5.85: Pending Appointment Page 3

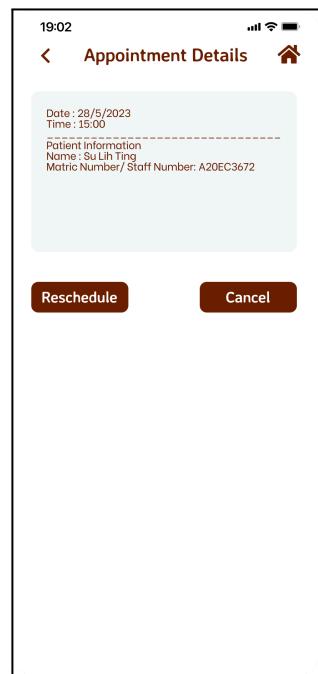


Figure 5.87: Approved Appointment Details Page 2

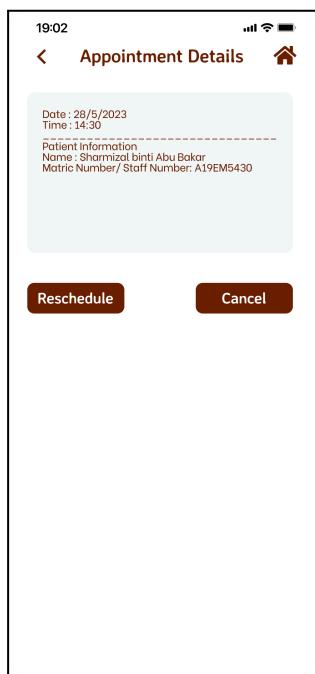


Figure 5.86: Approved Appointment Details Page 1

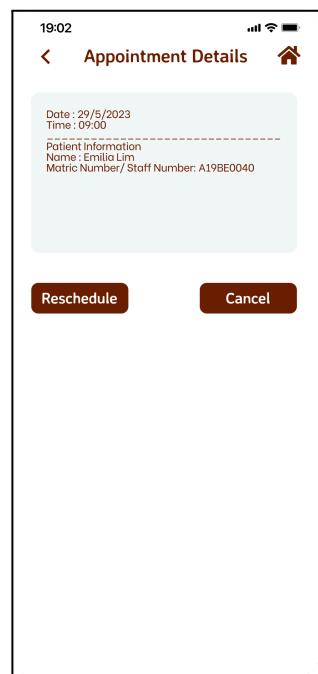


Figure 5.88: Approved Appointment Details Page 3

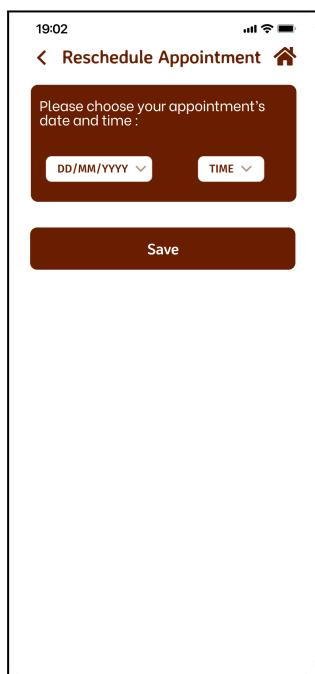


Figure 5.89: Reschedule Appointment Page

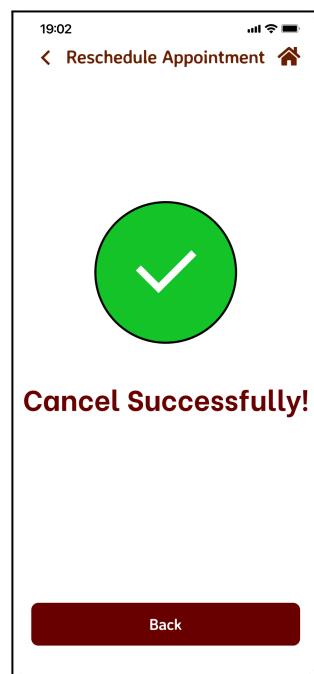


Figure 5.91: Reschedule Appointment
Canceled Page



Figure 5.90: Reschedule Appointment
Saved Page