



SECP3204: Software Engineering

System Requirement Specification (SRS)

Student Health and
Wellness Management System

Version 1.0

9th May 2023

Faculty of Computing

Prepared by: Software

Revision Page

a. Overview

In this version of system description, we have completed the System Requirement Specification (SRS) section.

b. Target Audience

The targeted audience of this system description are the developers, lecturers of UTM 2u2i Program and also the CEO of Mengaji One to One Program.

c. Project Team Members

Member Name	Role	Task	Status
Goh Jiale	Main Developer	Section 2.2.4	Complete
		Section 2.2.5	Complete
		Section 2.2.6	Complete
		Section 2.3	Complete
Lee Yik Hong	User Experience Designer	Section 2.2.7	Complete
		Section 2.2.8	Complete
		Section 2.2.16	Complete
		Section 2.2.17	Complete
		Section 2.4	Complete
Ong Yi Yan	Requirements Analyst	Section 2.1	Complete
		Section 2.2.1	Complete
		Section 2.2.2	Complete
		Section 2.2.3	Complete
Koh Su Xuan	Quality Assurance Tester	Section 2.2	Complete
		Section 2.2.9	Complete
		Section 2.2.10	Complete
		Section 2.2.14	Complete
		Section 2.2.15	Complete

Tang Yan Qing	Documentation Specialists	Section 1	Complete
		Section 2.2.11	Complete
		Section 2.2.12	Complete
		Section 2.2.13	Complete

d. **Version Control History (GOH)**

Version	Primary Author(s)	Description of Version	Date of Completed
1.0	Goh Jiale	This is the initial version of SRS documentation.	6 June 2023

Note:

This System Documentation (SD) template is adapted from IEEE Recommended Practice for Software Requirements Specification (SRS) (IEEE Std. 830-1998), Software Design Descriptions (SDD) (IEEE Std. 10161998 1), and Software Test Documentation (IEEE Std. 829-2008) that are simplified and customised to meet the need of SECJ2203 course at Faculty of Computing, UTM. Examples of models are from Arlow and Neustadt (2002) and other sources stated accordingly.

Table of Contents

1	Introduction		
	1.1	Purpose	5
	1.2	Scope	5
	1.3	Definitions, Acronyms and Abbreviation	6
	1.4	References	7
	1.5	Overview	8
2	Specific Requirements		
	2.1	User characteristics	9
	2.2	System Features	10
		2.2.1 UC01: Use Case <Sign Up>	14
		Use Case Specification of UC01	
		Activity Diagram of UC01	
		System Sequence Diagram of UC01	
		2.2.2 UC02: Use Case <Login>	19
		Use Case Specification of UC02	
		Activity Diagram of UC02	
		System Sequence Diagram of UC02	
		2.2.3 UC03: Use Case <Forgot Password>	23
		Use Case Specification of UC03	
		Activity Diagram of UC03	
		System Sequence Diagram of UC03	
		2.2.4 UC04: Use Case <Manage Own Account>	27
		Use Case Specification of UC04	
		Activity Diagram of UC04	
		System Sequence Diagram of UC04	

		2.2.5	UC05: Use Case <Manage All Users Account>	34
			Use Case Specification of UC05	
			Activity Diagram of UC05	
			System Sequence Diagram of UC05	
		2.2.6	UC06: Use Case <Upload Health Information>	38
			Use Case Specification of UC06	
			Activity Diagram of UC06	
			System Sequence Diagram of UC06 (GOH)	
		2.2.7	UC07: Use Case <Key in Patient Estimated Waiting Time and people remained>	44
			Use Case Specification of UC07	
			Activity Diagram of UC07	
			System Sequence Diagram of UC07	
		2.2.8	UC08: Use Case <View Patient Waiting Time and people remained>	48
			Use Case Specification of UC08	
			Activity Diagram of UC08	
			System Sequence Diagram of UC08	
		2.2.9	UC09: Use Case <Upload health tips>	52
			Use Case Specification of UC09	
			Activity Diagram of UC09	
			System Sequence Diagram of UC09	
		2.2.10	UC10: Use Case <View health tips>	57
			Use Case Specification of UC10	
			Activity Diagram of UC10	
			System Sequence Diagram of UC10	
		2.2.11	UC11: Use Case <Self-Test>	61

		Use Case Specification of UC11	
		Activity Diagram of UC11	
		System Sequence Diagram of UC11	
	2.2.12	UC12: Use Case <Set up appointment>	67
		Use Case Specification of UC12	
		Activity Diagram of UC12	
		System Sequence Diagram of UC12	
	2.2.13	UC13: Use Case <Manage appointment>	70
		Use Case Specification of UC13	
		Activity Diagram of UC13	
		System Sequence Diagram of UC13	
	2.2.14	UC14: Use Case <Set up appointment>	75
		Use Case Specification of UC14	
		Activity Diagram of UC14	
		System Sequence Diagram of UC14	
	2.2.15	UC15: Use Case <Manage appointment>	80
		Use Case Specification of UC15	
		Activity Diagram of UC15	
		System Sequence Diagram of UC15	
	2.2.16	UC16: Use Case <Give Feedbacks>	85
		Use Case Specification of UC16	
		Activity Diagram of UC16	
		System Sequence Diagram of UC16 (LEE)	
	2.2.17	UC17: Use Case <View Feedbacks>	89
		Use Case Specification of UC17	
		Activity Diagram of UC17	
		System Sequence Diagram of UC17 (LEE)	

	2.3	Software System Attributes, Performance and Other Requirements		
		2.3.1	Software System Attributes	
		2.3.2	Software System Performance	
		2.3.3	Other Requirements in Software	
	2.4	Design Constraints		
		2.4.1	Environmental Constraints	
		2.4.2	Hardware Constraints	
		2.4.3	Security Constraints	
		2.4.4	Compatibility Constraints	

1. Introduction

1.1 Purpose

This system documentation (SD) provides various elements that comprehensively explain and detail the system's architecture, functions and features. By referring to this document, a clear understanding of the system's objectives, features, interface and technical specifications can be attained by developers, stakeholders and users alike. Additionally, the documentation is prepared based on the hardware and software requirements gathered by initiating interview discussions with five separate stakeholders. Standardised data and guidelines will be accessible to all system users through this documentation, ultimately minimising the risk of any mistakes or discrepancies. Understanding how the system operates and how it can be improved is essential for developers, stakeholders and users who seek valuable sources in the future.

1.2 Scope

This System Documentation (SD) pertains to the software product "Student Health and Wellness Management System" and is designed to provide a clear understanding of the system development process. Introducing an unprecedented strategy to give college students direct access to medical support, health tracking and trustworthy health advice and counsel fulfils their need for top-notch well-being management. In order to empower students in KTDI to communicate and collaborate with professionals at the university's health centres, obtain medical counsel and plan visits for medical treatments, this system is made available to them. Additionally, this system offers university students a timely and seamless pathway to immediate aid in challenging moments. There were 5 stakeholders in this project: a student, a psychologist, a dentist, a medical officer and an administrator.

The following will be the boundaries of the Student Health and Wellness Management System:

1. Seamlessly integrating with the current student information system, the system is set to be crafted by a development team to ensure it is compatible with the administrators' goals. Close cooperation will ensure compatibility that will be easy.
2. The system will include 7 modules which are **Authentication system**, **Account Management system**, **Patient Queue Management system**, **Health Tips system**, **Mental Health Support system**, **Dental Health Appointment system** and **Feedback system**.
3. The security and privacy features of the system will follow all applicable laws and regulations such as the Personal Data Protection Act (PDPA).

1.3 Definitions, Acronyms and Abbreviation

Term	Definition
SHWMS	Student Health and Wellness Management System - A system developed with a series of features to assist students in taking care of their physical, mental and emotional health.
SRS	System Requirement Specification – a structured document that specified the detailed descriptions of the system's functions and operational constraints
SDD	System Design Document – a structured document that specified the technical specifications and design of a software system
STD	System Testing Document – a structured document that specified the testing strategy for the software system

UML	Unified Modelling Language - a graphical notation used to create visual models of software system
KTDI	Kolej Tun Dr. Ismail (Tun Dr. Ismail College) - a college for implementing Student Health and Wellness Management System as a sample
PKU	University Health Center - a health centre in UTM for implementing Student Health and Wellness Management System
DASS	Depression Anxiety and Stress Scale - a self-report scale created for measuring emotional states including depression, anxiety and stress
FE	Flow of Events
OTP	One Time Password that will send a 6 digit password to contact number or email.

1.4 References

1. IEEE. IEEE Std. 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

1.5 Overview

The requirements for the Student Health and Wellness Management System are described in this document's four main sections.

Section 1 introduces the user characteristics based on the different user types involved in this system. For example, PKU staff, students and administrators.

Besides, section 2 provides an overview of the system's features using use cases, activity and sequence diagrams.

Software system attributes, performance and other requirements are described in section 3. In this section, a detailed description of the functional and non-functional requirements of the software system, including performance, reliability and other attributes will be written here.

Lastly, the design constraints which contain any limitations or constraints that must be considered during the design and development of the software system are covered in the last section.

2. Specific Requirements

2.1 User characteristics

User	Characteristics
UTM Students	<ul style="list-style-type: none">• UTM students who are actively pursuing their studies.• Good level of digital literacy• Basic understanding of the English language
PKU Doctors	<ul style="list-style-type: none">• Trained medical professionals who possess the necessary expertise in their respective fields. They are• Technical proficiency required to navigate and utilise the functionalities of the application effectively.• Proficiency in understanding and utilising medical terms in English
PKU Administrator	<ul style="list-style-type: none">• Familiar with the operations and administrative procedures of PKU.• Fundamental understanding of the English language,• Technical skills to navigate its features and perform administrative tasks efficiently.

2.2 System Features

Student Health and Wellness Management System is a software system that operates on mobile devices, providing a means for UTM students to better manage their health on both physical and mental and at the same time ease the process of seeking health support from PKU UTM. It provides a centralised platform that streamlines various health-related processes, promotes wellness initiatives by implementing the Health Tips feature and facilitates effective communication in terms of making appointments between students and healthcare professionals.

The system features are illustrated in terms of modules in Figure 1.0 below. The detailed description of each module and functions is tabulated in Table 1.0.

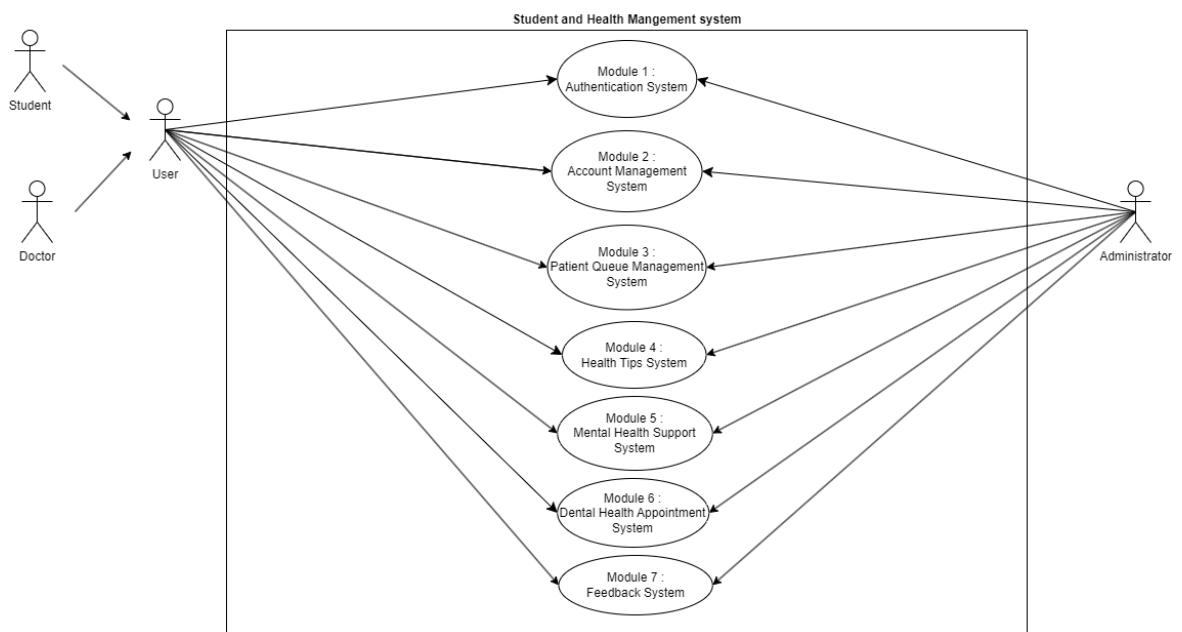


Figure 1.0: Use Case Diagram for <Student Health and Wellness System>

Module	Function	Description
Authentication Module	UC01 – Sign up	This use case allows the student, doctor and administrator to sign up as a user for the application-based system.
	UC02 - Login	This use case allows the student, doctor and administrator to login the application.
	UC03 - Forgot password	This use case allows the student, doctor and administrator to reset their passwords.
Account Management Module	UC04 – Manage own account	This use case allows the student, doctor and administrator to manage their accounts.
	UC05 – Manage all users accounts	This use case allows the administrator to manage all users accounts.
	UC06 – Upload health information	This use case allows the student, doctor and administrator to upload their health information including demographic information and lab results.
Patient Queue Management Module	UC07 – Key in patient waiting time and people remained	This use case allows the administrator to key in the estimated patient waiting time and people remaining (amount of pending patients) in PKU.
	UC08 – View patient waiting time and people remained	This use case allows the student and doctor to view the estimated patient waiting time and people remaining (amount of pending patients) in PKU.
Health Tips Module	UC09 – Upload health tips	This use case allows the administrator to upload the health tips in the system.
	UC10 – View health tips	This use case allows the student, doctor and administrator to view the health tips in the application.

Mental Health Support Module	UC11 – Self Test	This use case allows the student to have a mental health self-test (DASS).
	UC12 – Set up appointment	This use case 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.
	UC13 - Manage appointment	This use case allows the doctor and administrator to manage the appointment applied by the student.
Dental Health Appointment Module	UC14 – Set up appointment	This use case allows the student to set up an appointment with the doctor of PKU in the application.
	UC15 - Manage appointment	This use case allows the doctor and administrator to manage the appointment applied by the student.
Feedback and Reviews Module	UC16 – Give feedback	This use case allows the student and doctor to give feedback.
	UC17 – View feedback	This use case allows the administrator to view feedback given by the student and doctor.

Table 1.0: Description of Module and Functions for Student Health and Wellness

Management System

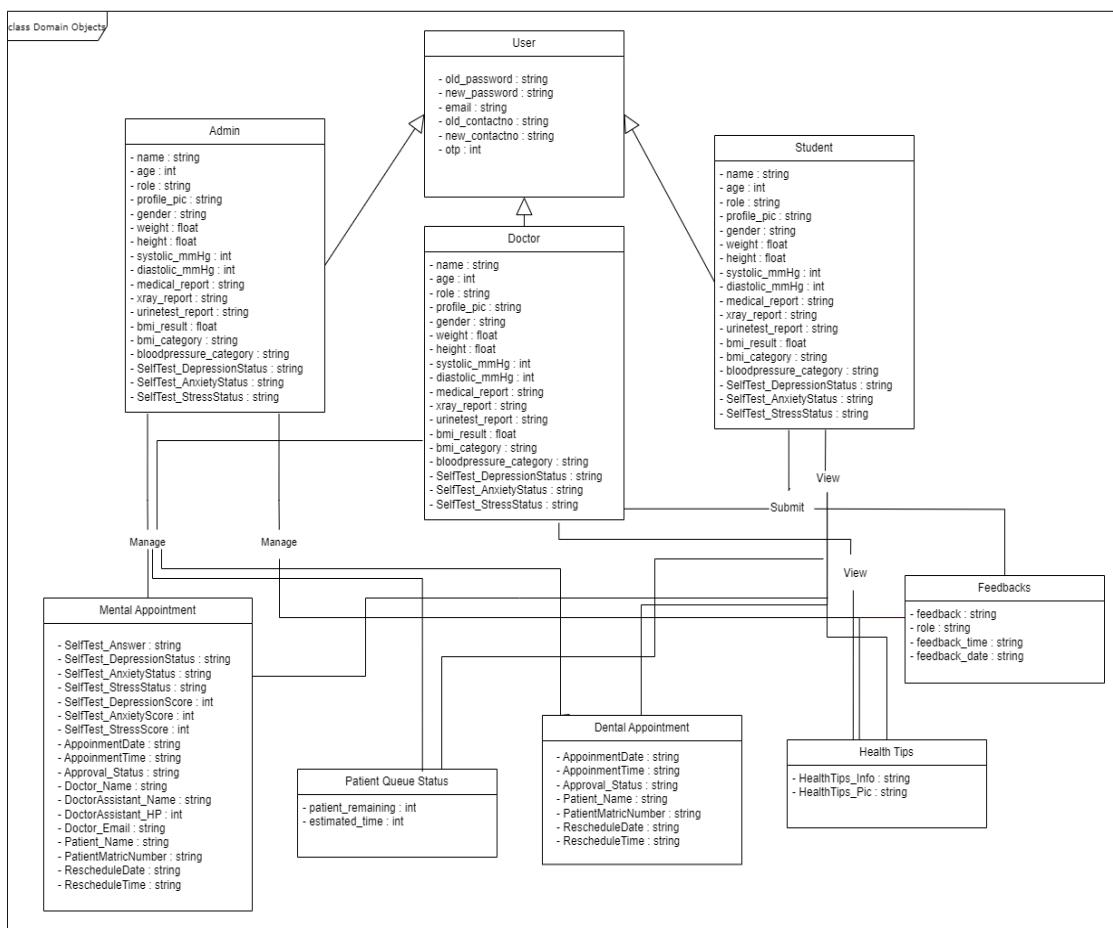


Figure 1.1: Domain Model for Student Health and Wellness Management System

2.2.1 Module 01: <Authentication System>

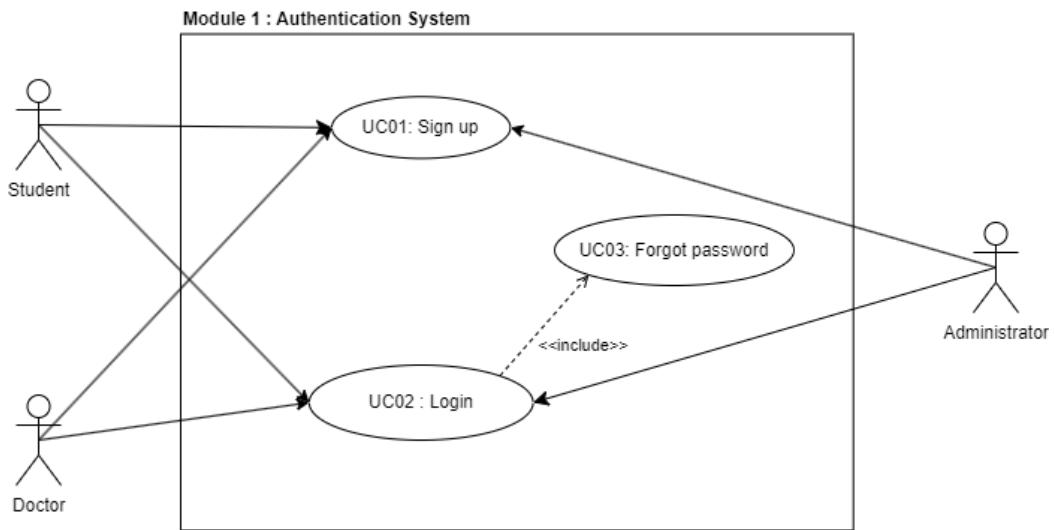


Diagram 2.2.1: Use Case Diagram of Submodule <Authentication System>

2.2.1.1 UC01: Use Case <Sign Up>

Table 2.2.1.1: Use Case Description for <Sign Up>

Use case: <Sign Up>
ID: UC01
Actors: Student
Preconditions: <ul style="list-style-type: none"> 1. The user has installed and opened the app on mobile devices 2. The user has active internet connection to the system 3. The user has a UTM organisation email address
Flow of events: <ul style="list-style-type: none"> 1. The user clicks ‘Sign Up’ 2. The user enters UTM organisation email address 3. The user enters password 4. The user enters password for second time 5. The user clicks ‘Sign Up’ 6. System displays ‘Thank you for signing up!’

- | |
|---------------------------------------|
| 7. The user is directed to login page |
|---------------------------------------|

Postconditions:

1. The user successfully signed up and his information is registered in the system.
2. The user receives a verification email for their successful sign-ups.
3. Verification is required before the user logs in.

Alternative flow 1:

1. The user clicks ‘Login’
2. The user clicks ‘Sign Up’ besides ‘Don’t have an account?’
3. The user enters UTM organisation email address
4. The user enters password
5. The user enters password for second time
6. The user clicks ‘Sign Up’
7. System displays ‘Thank you for signing up!’
8. The user is directed to the login page.

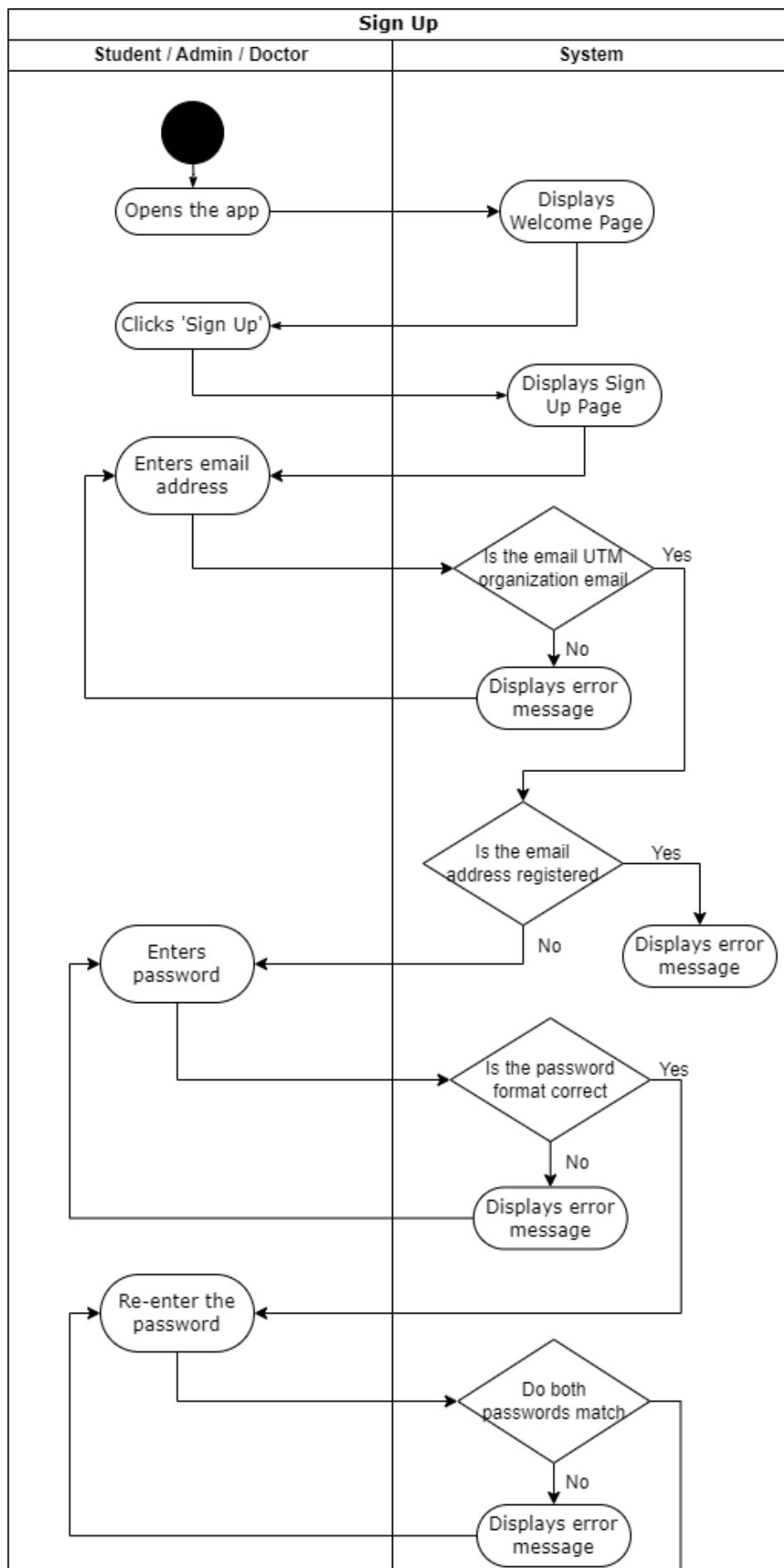
Postconditions:

1. The user successfully signed up and his information is registered in the system.
2. The user receives a verification email about their successful sign-ups.
3. Verification is required before the user logs in.

Exception flow (if any):

1. Invalid email address
 - 1.1 Users enter their personal email address instead of UTM organisation email address
 - 1.2 Error message will be displayed

- 2. Email is registered before
 - 2.1 Users enter their UTM organisation email address but it is registered
 - 2.2 Error message will be displayed
 - 2.3 Users will be directed to login page
- 3. Incorrect password format
 - 3.1 Users enter password which does not meet the specified criteria
 - 3.2 Error message will be displayed
- 4. Both passwords do not match
 - 4.1 Users enter the second password which is not same as the first one
 - 4.2 Error message will be displayed



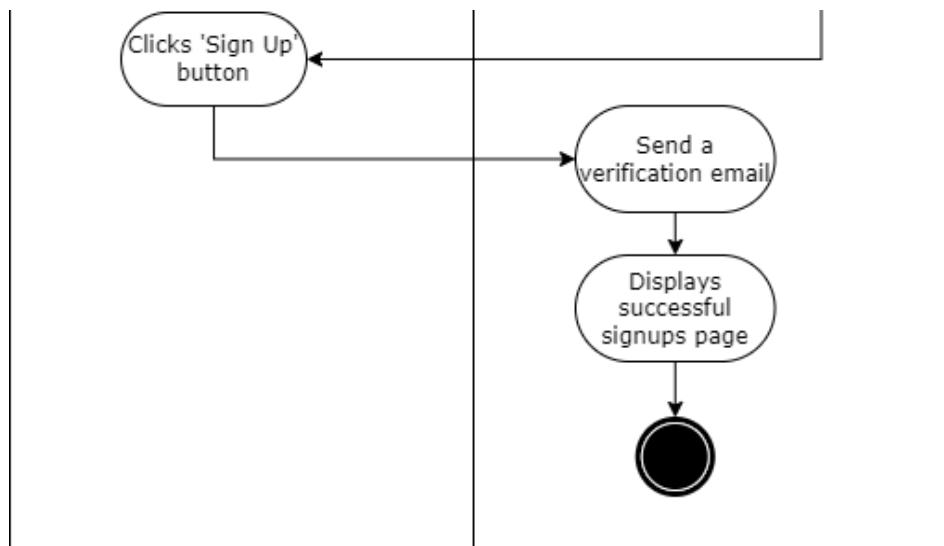


Figure 2.2.1.1: Activity Diagram for <Sign Up>

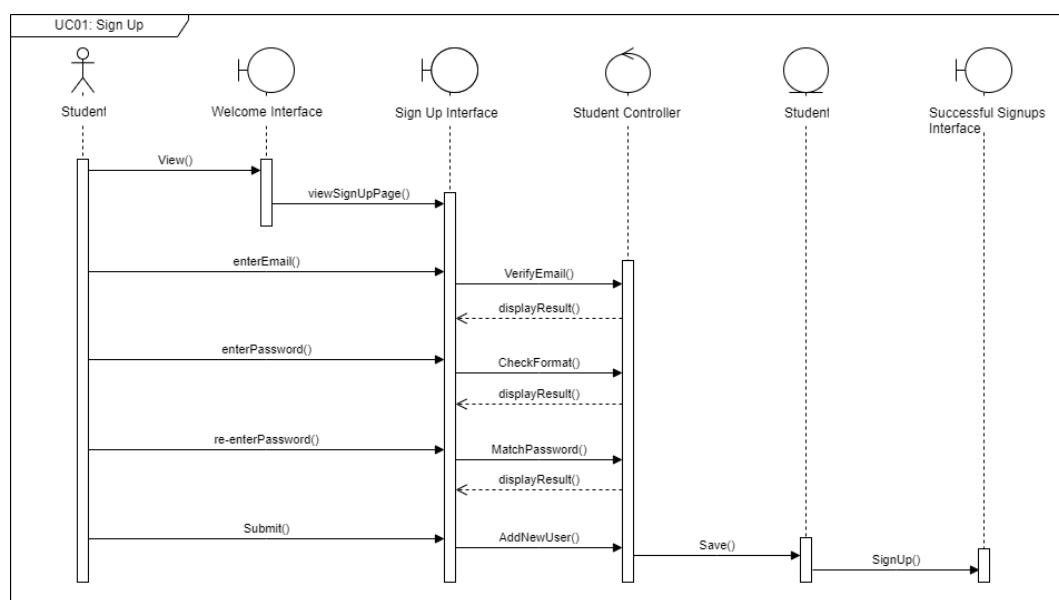


Figure 2.2.1.2: Sequence Diagram for <Sign Up>

2.2.1.2 UC02: Use Case <Login>

Table 2.2.1.2: Use Case Description for <Login>

Use case: <Login>
ID: UC02
Actors: Student, Doctor, Admin
<p>Preconditions:</p> <ol style="list-style-type: none"> 1. The user has installed and opened the app on mobile devices 2. The user has active internet connection to the system 3. The user has a registered account.
<p>Flow of events:</p> <ol style="list-style-type: none"> 1. The user clicks ‘Login’ 2. The user enters UTM organisation email address 3. The user enters password 4. The user clicks ‘Login’ 5. The user is redirected to home page
<p>Postconditions:</p> <ol style="list-style-type: none"> 1. The user is successfully authenticated and logged in. 2. The user's login status is maintained, allowing them to remain logged in even if the internet connection is temporarily lost.
<p>Alternative flow n:</p> <ol style="list-style-type: none"> 1. The user clicks ‘Sign Up’ 2. The user enters UTM organisation email address 3. System displays ‘Email is already registered’ 4. The user clicks ‘Login’ besides ‘Already have an account?’

- | |
|--|
| <ol style="list-style-type: none"> 5. The user enters UTM organisation email address 6. The user enters password 7. The user clicks 'Login' 8. The user is redirected to home page |
|--|

Postconditions:

- 1. The user is successfully authenticated and logged in.
- 2. The user's login status is maintained, allowing them to remain logged in even if the internet connection is temporarily lost.

Exception flow (if any):

- 1. Invalid email address
 - 1.1 Users enter their personal email address instead of UTM organisation email address
 - 1.2 Error message will be displayed
- 2. Incorrect password
 - 2.1 User forgets their password
 - 2.2 System displays 'Incorrect password'

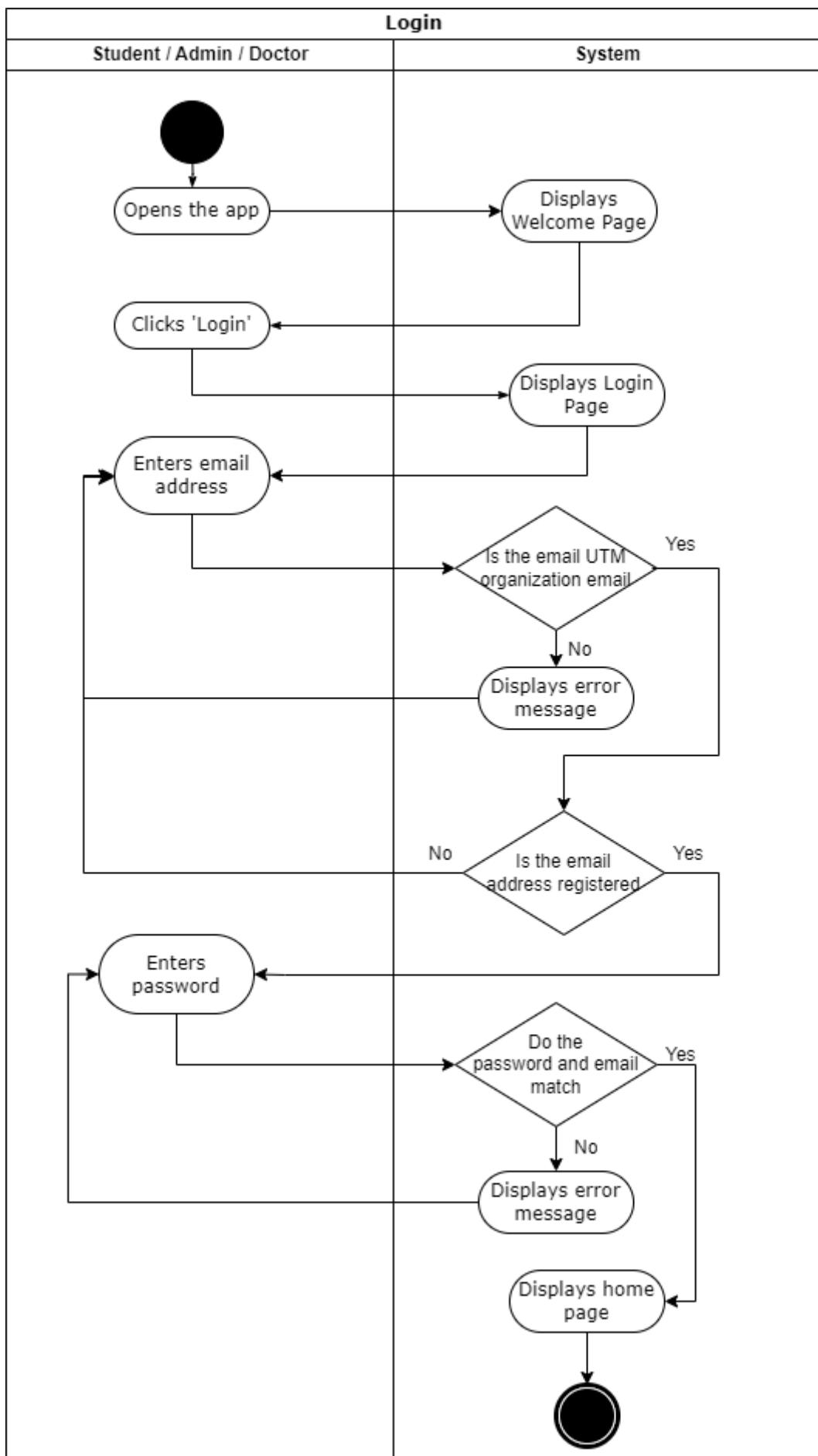


Figure 2.2.1.3: Activity Diagram for <Login>

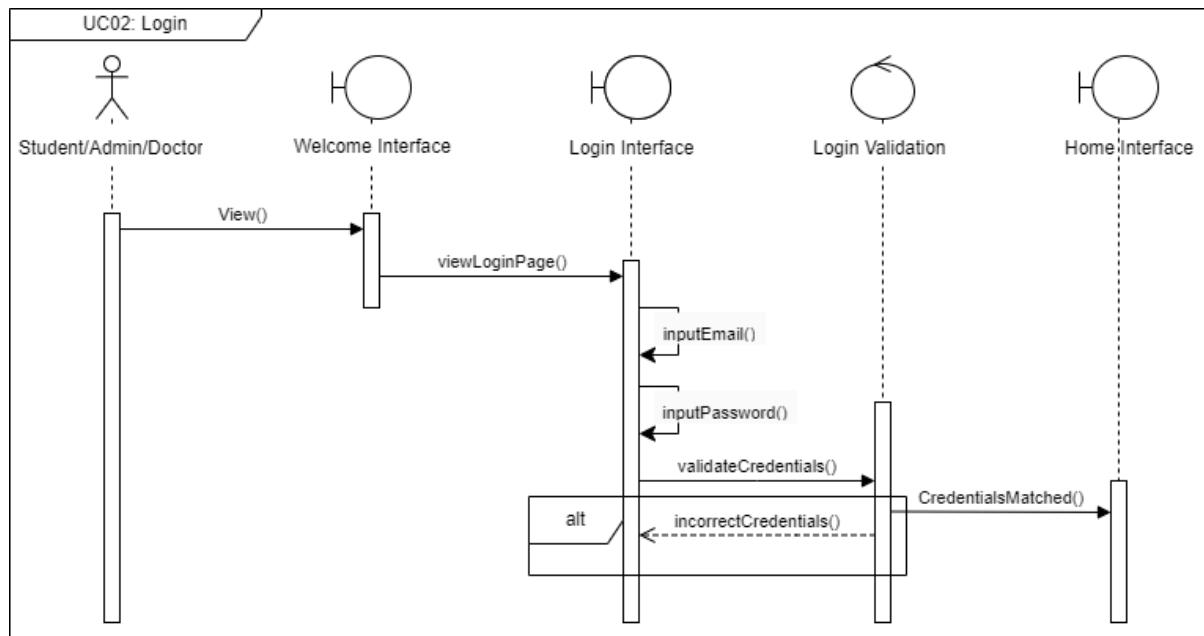


Figure 2.2.1.4: Sequence Diagram for <Login>

2.2.1.3 UC03: Use Case <Forgot Password>

Table 2.2.1.3: Use Case Description for <Forgot Password>

Use case: <Forgot Password>
ID: UC03
Actors: Student, Doctor, Admin
<p>Preconditions:</p> <ol style="list-style-type: none"> 1. The user has installed and opened the app on mobile devices. 2. The user has active internet connection to the system 3. The user has a registered account. 4. The user has forgotten their password or can not login due to password issues.
<p>Flow of events:</p> <ol style="list-style-type: none"> 1. The user clicks ‘Login’ 2. The user clicks ‘Forgot Password’ 3. The user enters registered UTM organisation email 4. The user clicks ‘Request Password Reset’ 5. The password reset link is sent to the user’s email address 6. The user resets password
<p>Postconditions:</p> <ol style="list-style-type: none"> 1. The user receives an email containing a link and instructions on how to reset their password. 2. The user successfully reset their password. 3. The user logs into the app using a new password.
<p>Exception flow (if any):</p> <ol style="list-style-type: none"> 1. Invalid email address <ol style="list-style-type: none"> 1.1 The email address users entered is not registered

1.2 Error message will be displayed

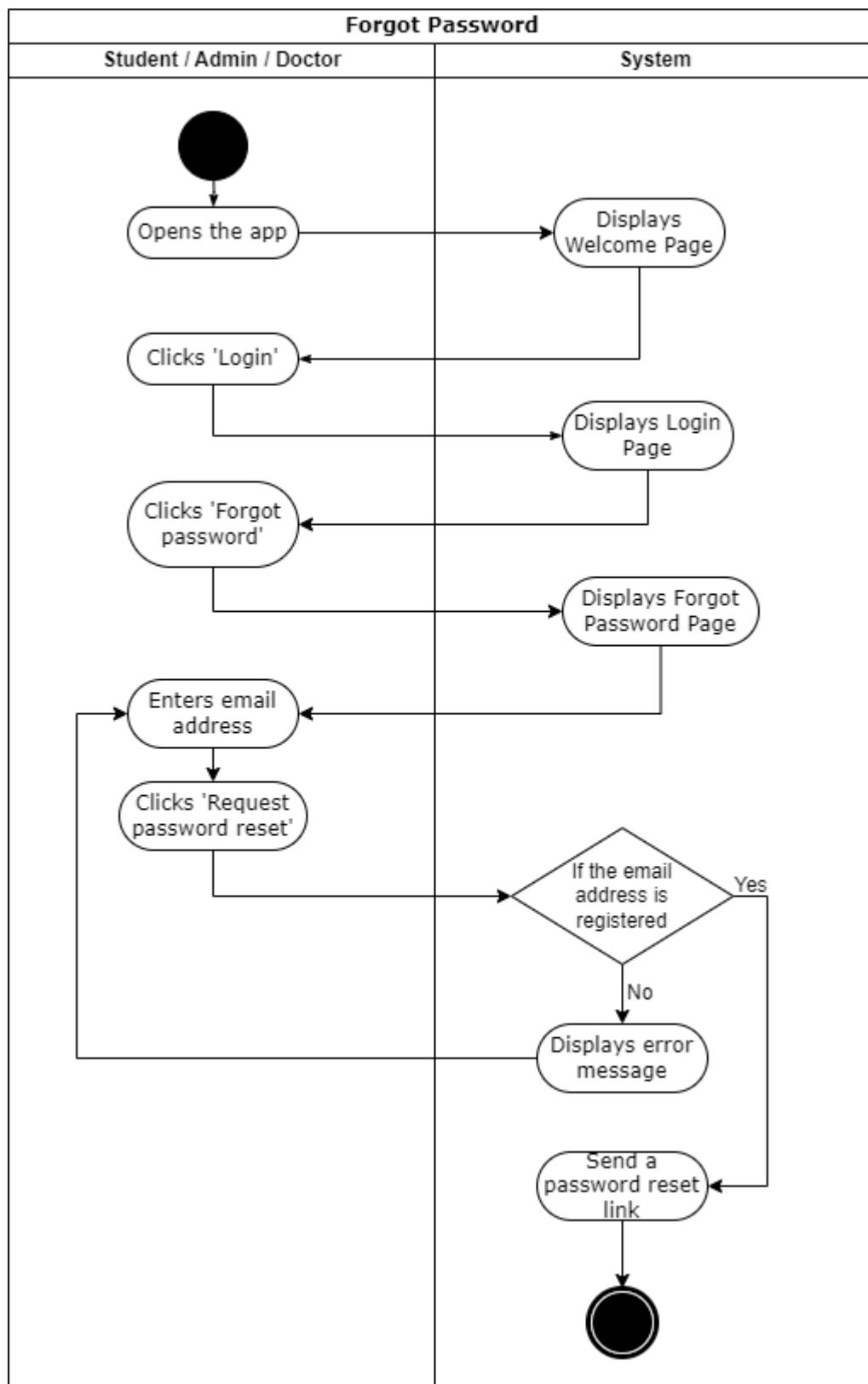


Figure 2.2.1.5: Activity Diagram for <Forgot Password>

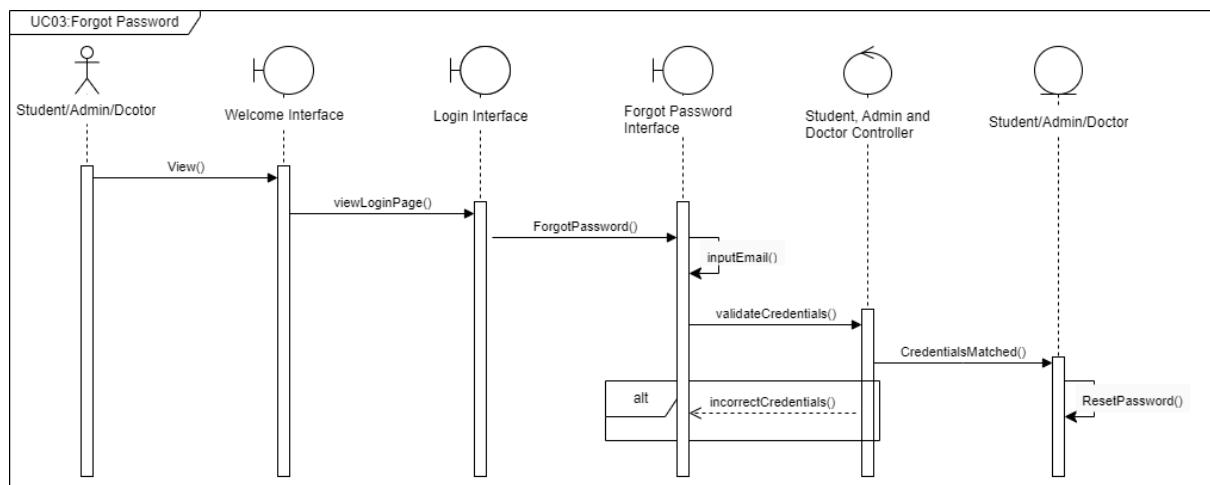


Figure 2.2.1.6: Sequence Diagram for <Forgot Password>

2.2.2 Module 02: <Account Management System>

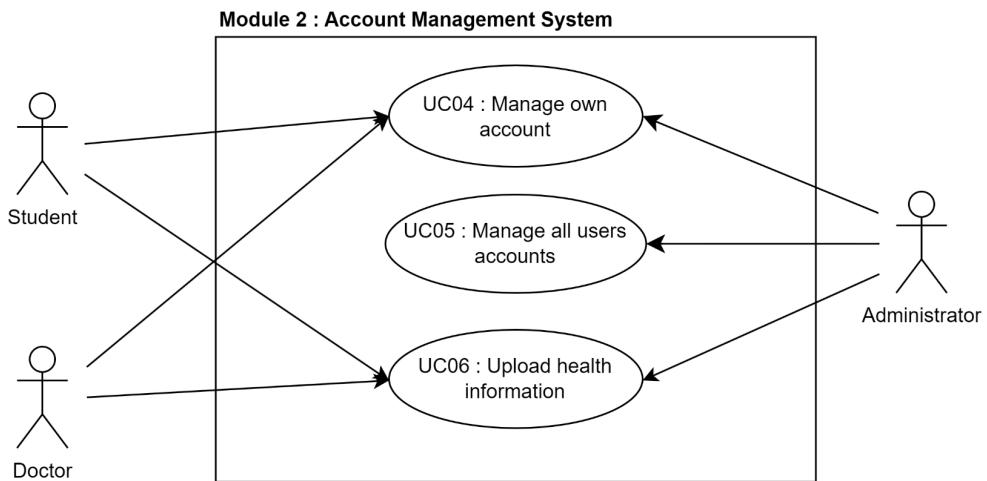


Diagram 2.2.2: Use Case Diagram of Submodule <Account Management System>

2.2.2.1 UC04: Use Case <Manage Own Account>

Table 2.2.2.1: Use Case Description for <Manage Own Account>

Use case: <Manage Own Account>	
ID:	UC04
Actors:	Student, Doctor, Admin
Preconditions:	<ol style="list-style-type: none"> 1. Users login to the system. 2. Users click on “Manage Profile” in the User Profile page.
Flow of events (FE):	<ol style="list-style-type: none"> 1. Users click on “Manage Profile” in the User Profile page. 2. If Users Edit Password <ol style="list-style-type: none"> 2.1. Click on the arrow at the right of the row of “Password” in the Manage Profile page. 2.2. Navigate to the “Password” page. 2.3. Users enter the original password and new password. 2.4. If the original password is correct, the user enters a contact number and otp. 2.5. If OTP is correct, the new password is saved.

- 2.6. If the password or OTP is wrong, users need to re-enter them.
3. If Users Edit Contact Number
 - 3.1. Click on the arrow at the right of the row of “Contact Number” in the Manage Profile page.
 - 3.2. Navigate to the “Contact Number” page.
 - 3.3. Enter a new Contact Number in the “Contact Number” page and otp.
 - 3.4. If OTP is correct, the new Contact Number is saved.
 - 3.5. If OTP is wrong, users need to re-enter them.
4. If Admin or Doctor Edit Role
 - 4.1. Click on the arrow at the right of the row of “Name” in the Manage Profile page.
 - 4.2. Navigate to the “Change Role” page.
 - 4.3. Choose “Role”.
 - 4.4. Save Role by clicking on the “Save” button at top-right corner.

Postconditions:

1. Students, doctors and admins successfully change their password for FE2.
2. Students, doctors and admins successfully change their contact number for FE3.
3. Doctors and admins successfully change their role for FE4.

Alternative flow 1:

At FE4.4, the user saves roles under an offline condition.

Postconditions:

Changes will be uploaded and synchronised when the device is under an online condition.

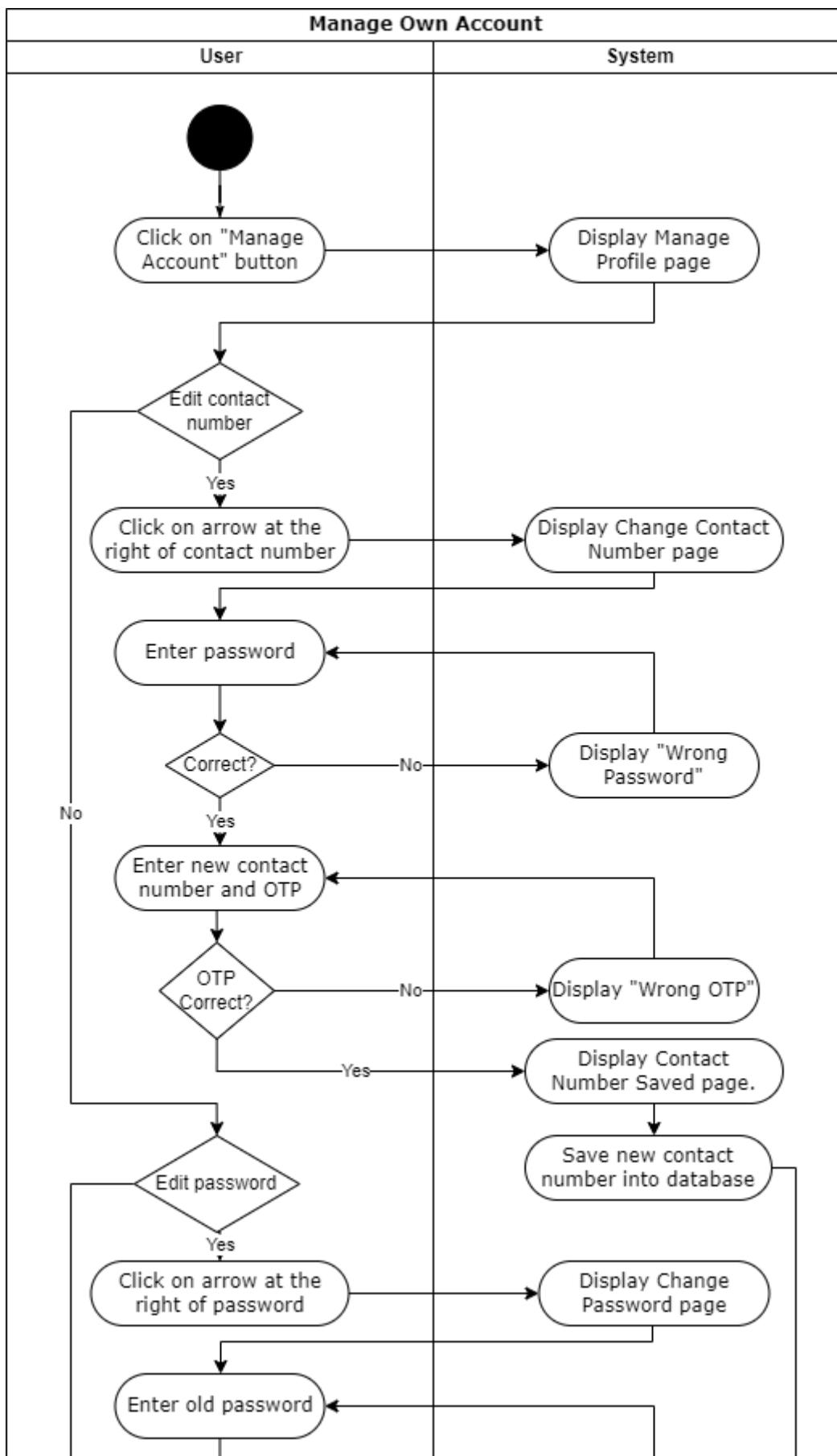
Exception flow:

- E.1. The textbox when editing the user account is empty.
 1. An error message will be displayed by the system.

2. Perform for FE3/FE4.

E.2. Students, doctors and admins may leave the edit page without clicking the save button.

1. The system will not save the edited information into the system database.
2. Perform for FE4.



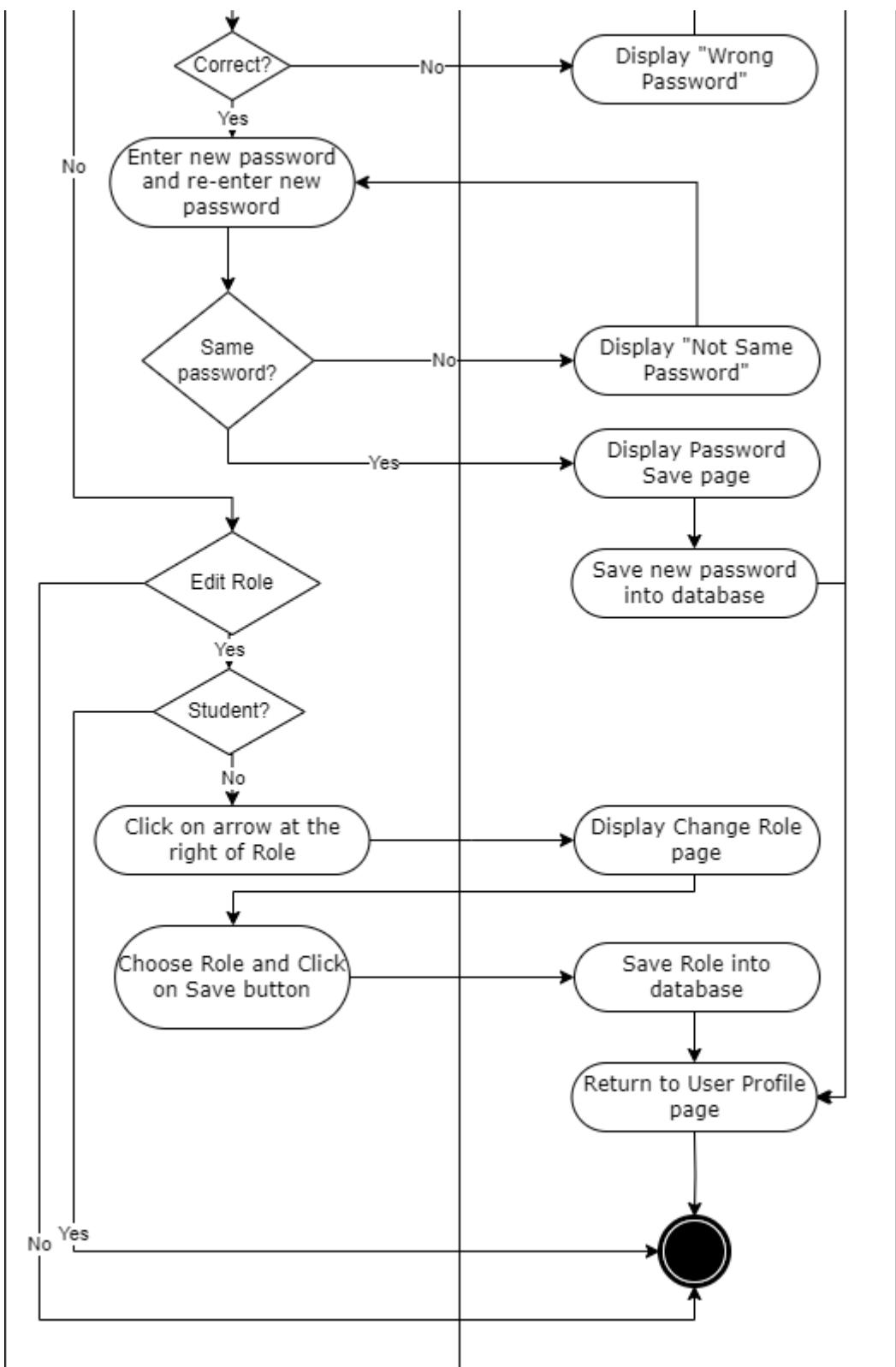


Figure 2.2.2.1: Activity Diagram for <Manage Own Account>

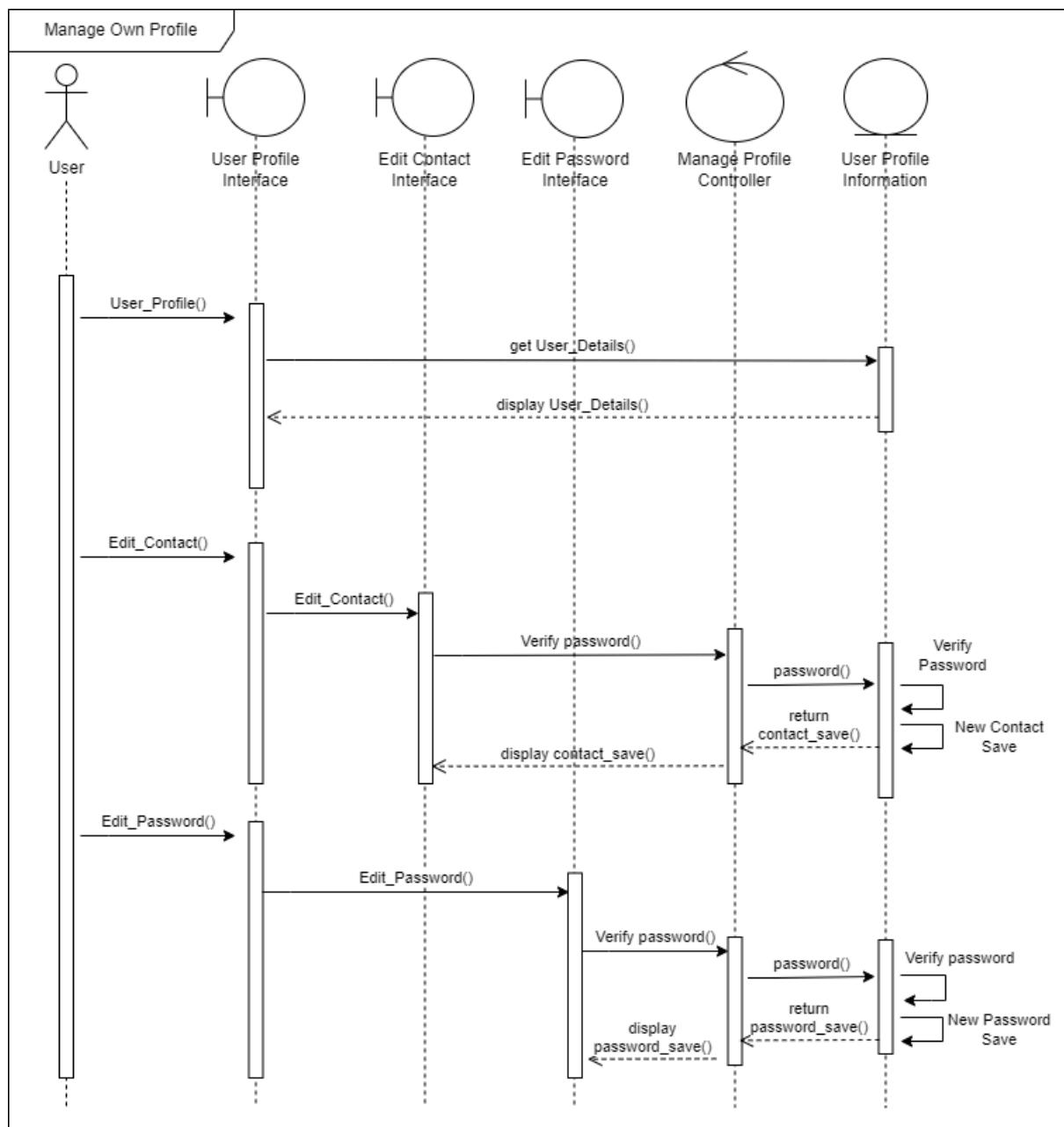


Figure 2.2.2.2: Sequence Diagram 1 for <Manage Own Account>

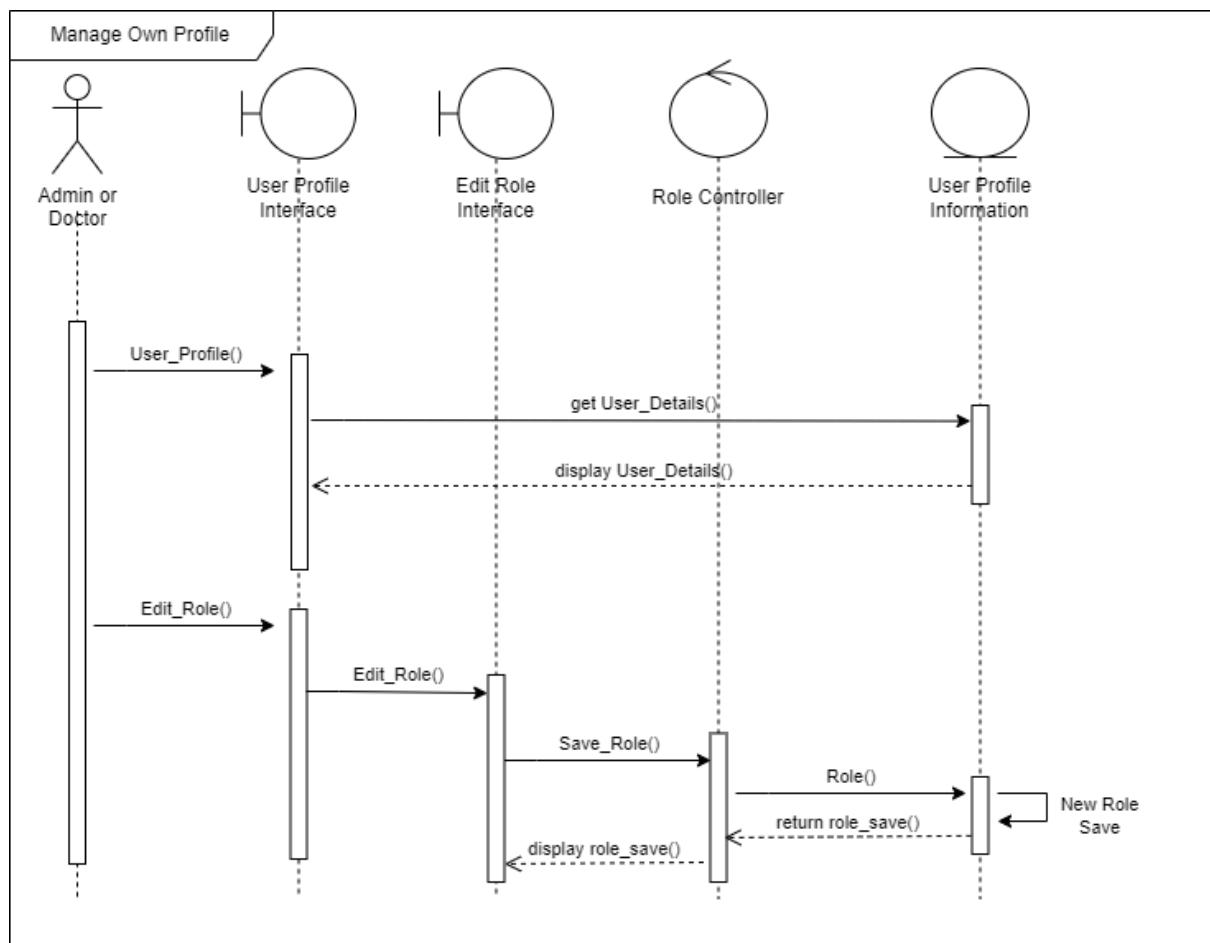


Figure 2.2.2.3: Sequence Diagram 2 for <Manage Own Account>

2.2.2.2 UC05: Use Case <Manage All Users Account>

Table 2.2.2.2: Use Case Description for <Manage All Users Account>

Use case: <Manage All Users Account>
ID: UC05
Actors: Admin
<p>Preconditions:</p> <ol style="list-style-type: none"> 1. Admin login to the system. 2. Admin click on the “Manage Users Profile” in the User Profile page.
<p>Flow of events:</p> <ol style="list-style-type: none"> 1. Admin go to “User Profile” and click on “Manage Users Profile”. 2. If admin search account <ol style="list-style-type: none"> a. Click on “Search by email” b. Key-in email address of users. c. Click Enter. d. The users with that email address shown. 3. If admin add an account <ol style="list-style-type: none"> a. Click on the add button. b. Key-in email address and password. c. Click on the create button. 4. If admin delete an account <ol style="list-style-type: none"> a. Click on the delete button. b. Click on the “Sure” button. 5. If admin view account information <ol style="list-style-type: none"> a. Click on the arrow at the right of the user's icon. b. View account details
<p>Postconditions:</p> <ol style="list-style-type: none"> 1. Users with email address that is the same as the key-in shown.

2. Admin successfully adds a user account.
3. Admin successfully deleted an user account.
4. Admin successfully views user account details.

Exception flow (if any):

- E.1. The entered email was not found in the system.
 1. An error message will be displayed by the system.
 2. Perform for FE2.
- E.2. The textbox when adding the user account is empty.
 1. An error message will be displayed by the system.
 2. Perform for FE3.

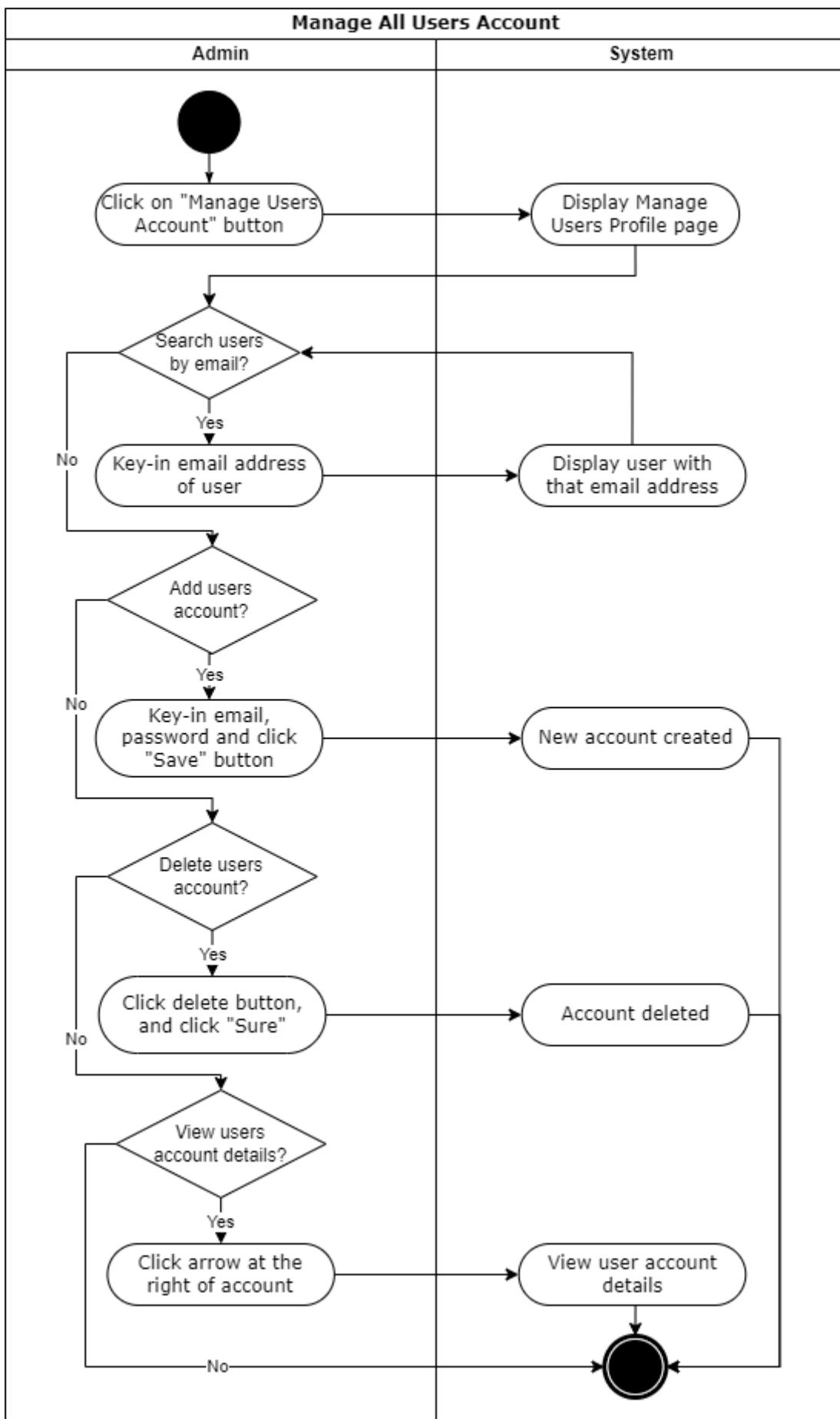


Figure 2.2.2.4: Activity Diagram for <Manage All Users Account>

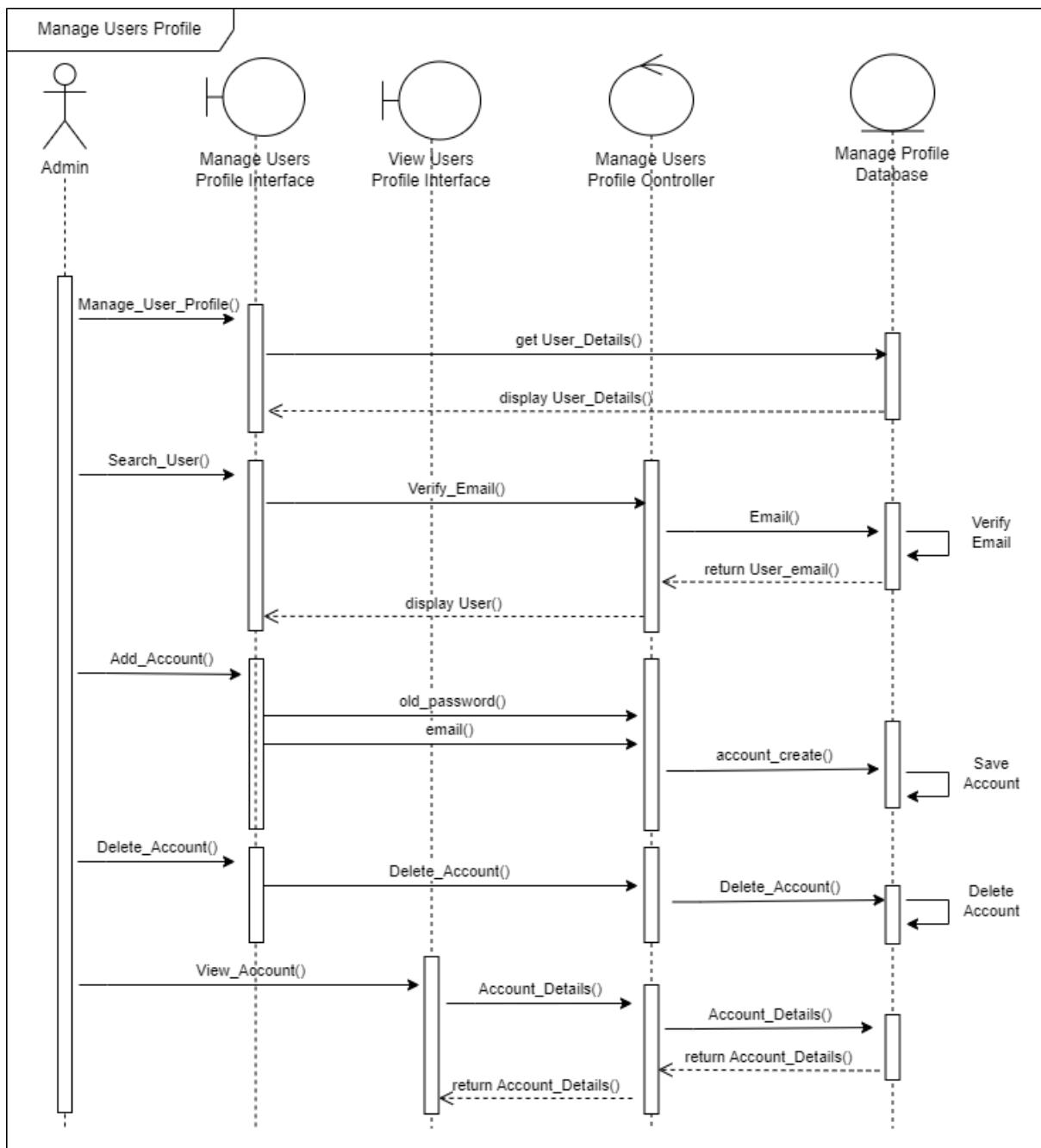


Figure 2.2.2.5: Sequence Diagram for <Manage All Users Account>

2.2.2.3 UC06: Use Case <Upload Health Information>

Table 2.2.2.3: Use Case Description for <Upload Health Information>

Use case: <Upload Health Information>
ID: UC06
Actors: Student, Doctor, Admin
Preconditions: <ol style="list-style-type: none"> 1. Users login to the system. 2. Users click on User Profile in the Home page.
Flow of events: <ol style="list-style-type: none"> 1. If users looking for BMI Calculation <ol style="list-style-type: none"> a. Go to “User Profile” and click on “BMI Calculation”. b. Edit by key-in Gender, Weight, Height and Age. c. System will calculate BMI and provide a kindly reminder for each scale of BMI. 2. If users looking for Medical Check-up Result / X-Ray Report / Urine Test Report / Mental Health Result (only for student) <ol style="list-style-type: none"> a. Go to “User Profile” and click on “Medical Check-up Result” / “X-Ray Report” / “Urine Test Report” / “Mental Health Result”. b. Upload a new Report. (Except for Mental Health Result) <ol style="list-style-type: none"> i. Clicking on the add logo. ii. Upload image for report. c. View report by clicking on the arrow at the right of the date for the report. 3. If users looking for Blood Pressure Report <ol style="list-style-type: none"> a. Go to “User Profile” and click on “Blood Pressure Report”.

- | |
|--|
| <p>b. Edit by key-in Systolic mmHg and Diastolic mmHg.</p> <p>c. System will calculate, show category and provide a kindly reminder for each category.</p> |
|--|

Postconditions:

1. Students, doctors and admins successfully calculate their BMI and view category for FE1.
2. Students, doctors and admins successfully add and view reports for FE2.
3. Students, doctors and admins successfully calculate their blood pressure category for FE3

Alternative flow 1:

At FE2, the user saves reports under an offline condition.

Postconditions:

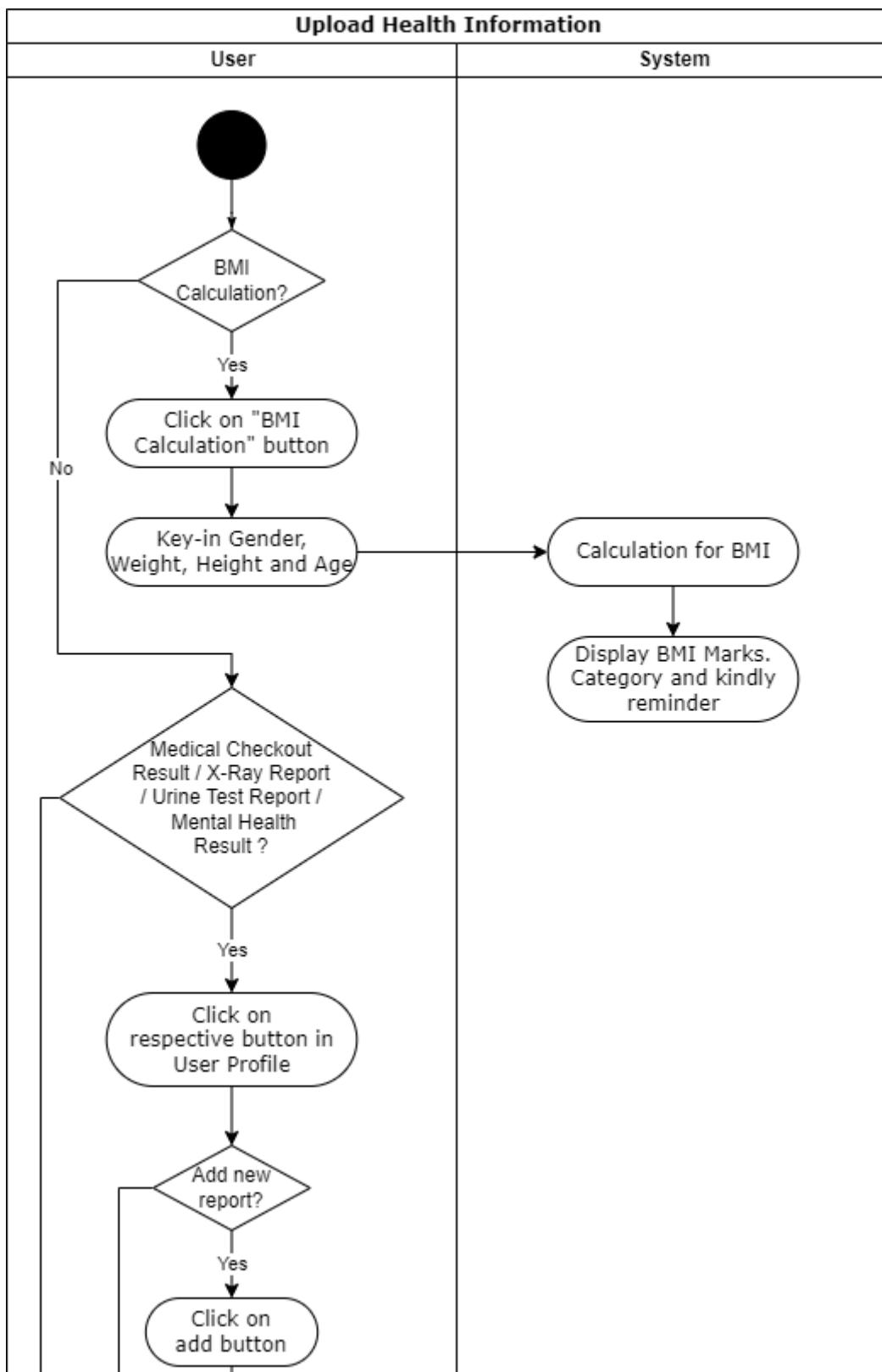
Changes will be uploaded and synchronised when the device is under an online condition.

Exception flow (if any):

E.1. The textbox when filling BMI Calculation and Blood Pressure

Report is empty.

1. An error message will be displayed by the system.
2. Perform for FE1/FE3.



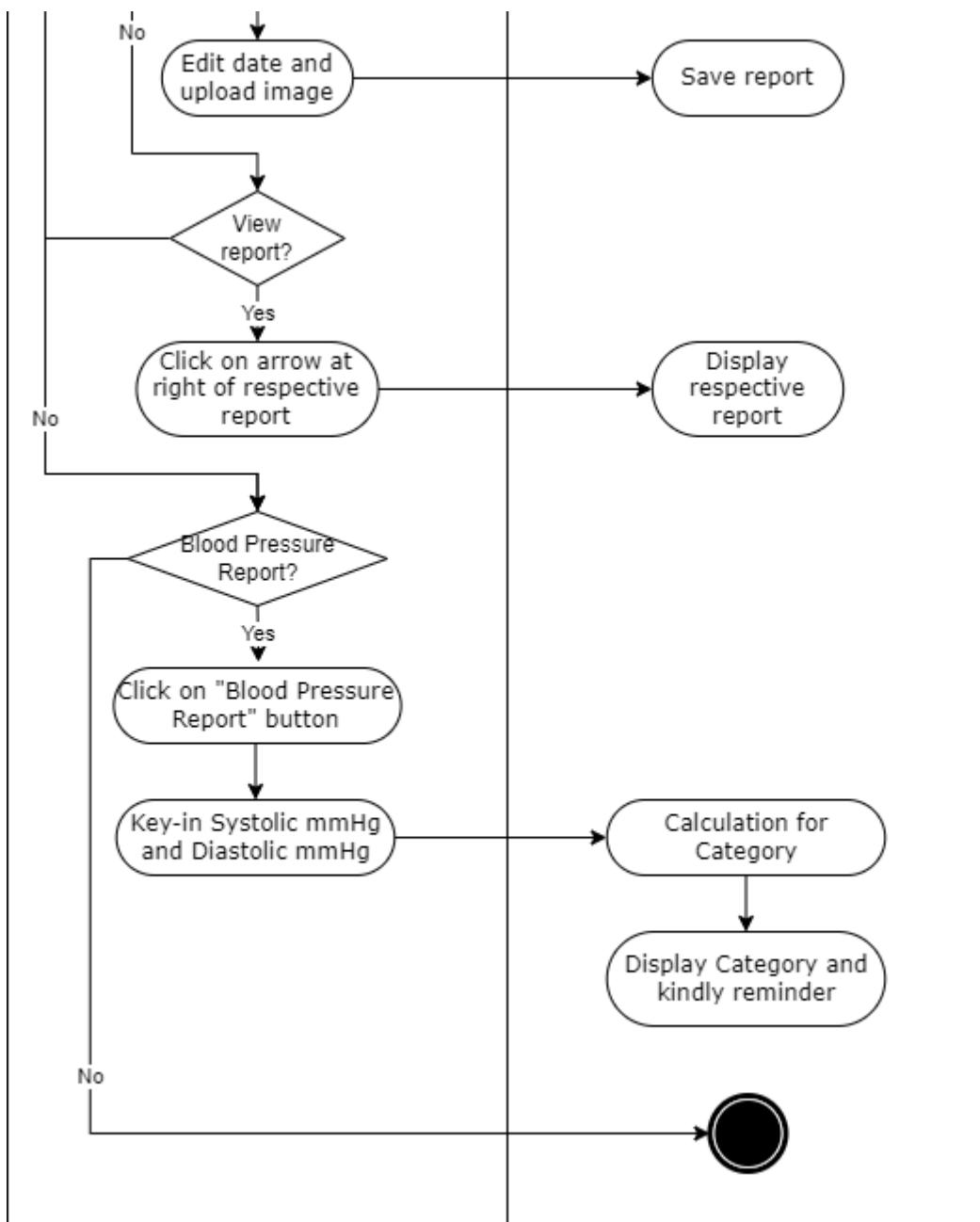


Figure 2.2.2.6: Activity Diagram for <Upload Health Information>

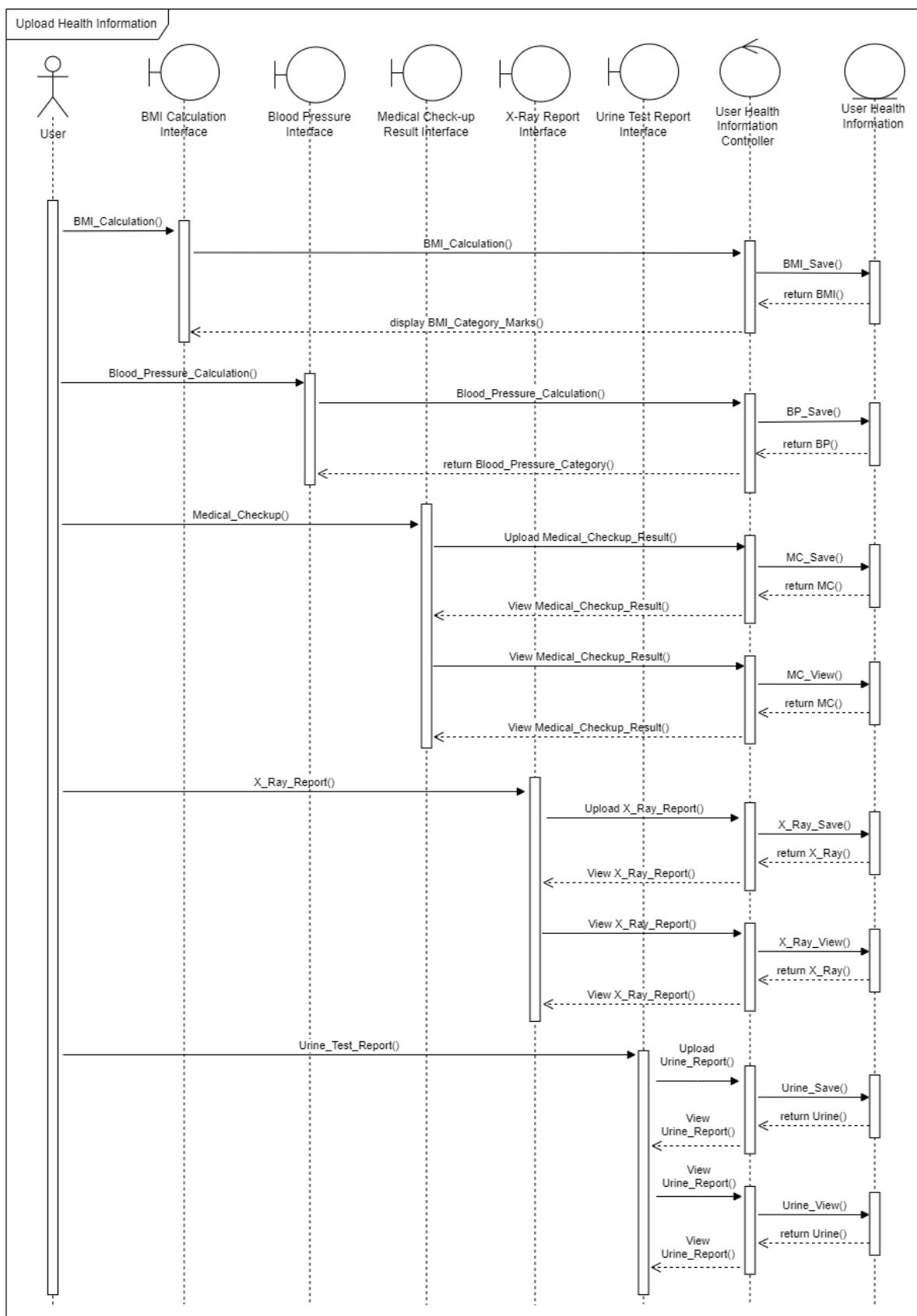
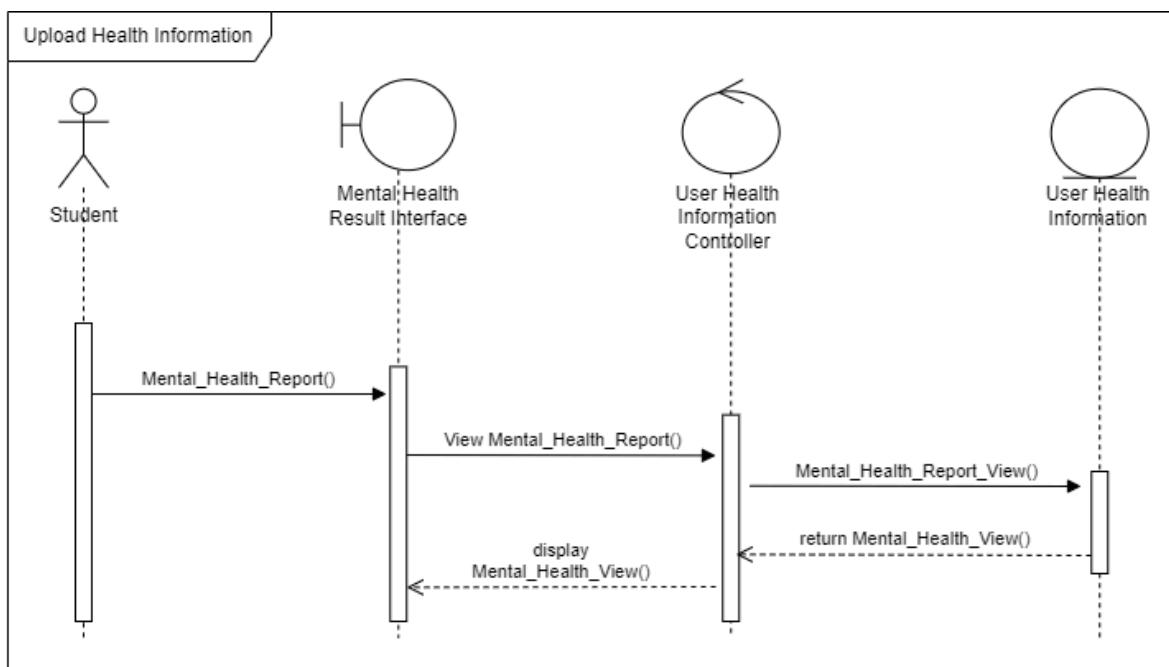


Figure 2.2.2.7: Sequence Diagram 1 for <Upload Health Information>



2.2.3 Module 03: <Patient Queue Management System>

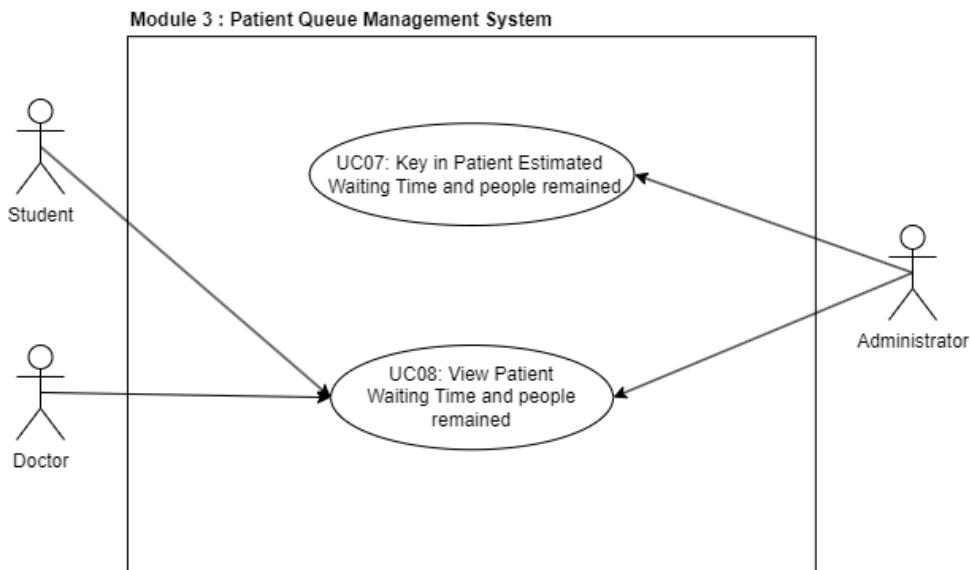


Diagram 2.2.3: Use Case Diagram of Submodule <Patient Queue Management System>

2.2.3.1 UC07: Use Case <Key in Patient Estimated Waiting Time and people remained>

**Table 2.2.3.1: Use Case Description for
<Key in Patient Estimated Waiting Time and people remained>**

Use case: <Key in Patient Estimated Waiting Time and people remained>
ID: UC07
Actors:
1. Administrator
Preconditions:
<ol style="list-style-type: none"> 1. Has active internet connection to the system. 2. Registered with the system. 3. Signed in system
Flow of events:
<ol style="list-style-type: none"> 1. View number of patient remaining and estimated time remaining

- a. System displays a “Patient Remaining” followed by numbers of patients, beside with “Estimated Time”.
- 2. Edit “Patient Remaining” and “Estimated Time”
 - a. The use case starts when the administrator 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. The user case ends.

Postconditions:

- 1. Patient remaining and estimated waiting time can be updated from time to time.

Alternative flow n:

- 1. Administrator can select the “Save” button to save the progress under an offline condition by storing the information in the application and device .

Postconditions:

- 1. Changes will be uploaded when the device is under an online condition

Exception flow (if any):

- 1. Changes discarded after a few minutes.

UC07: Key in Patient Estimated Waiting Time and people remained

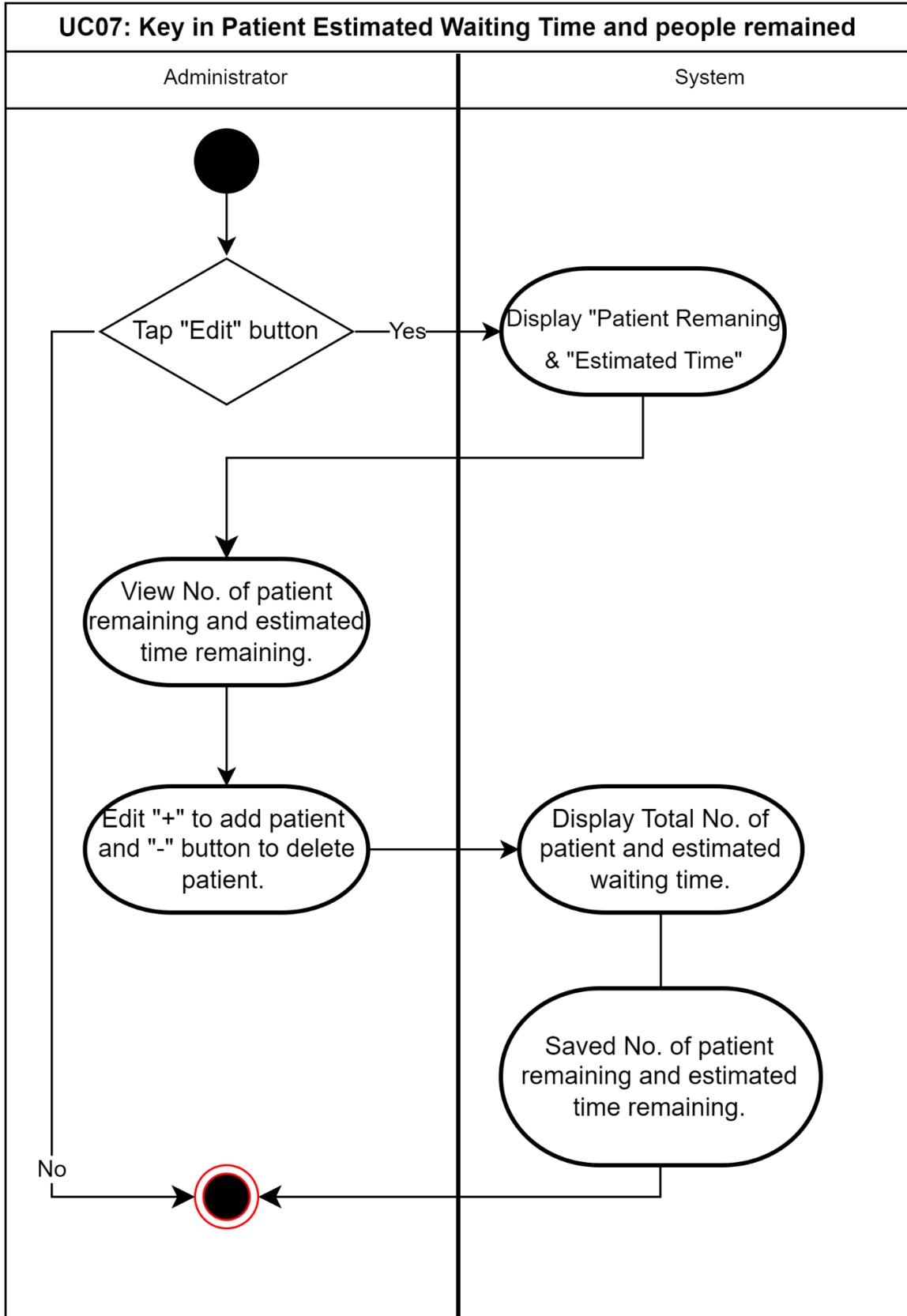


Figure 2.2.3.1: Activity Diagram for <Key in Patient Estimated Waiting Time and people remained>

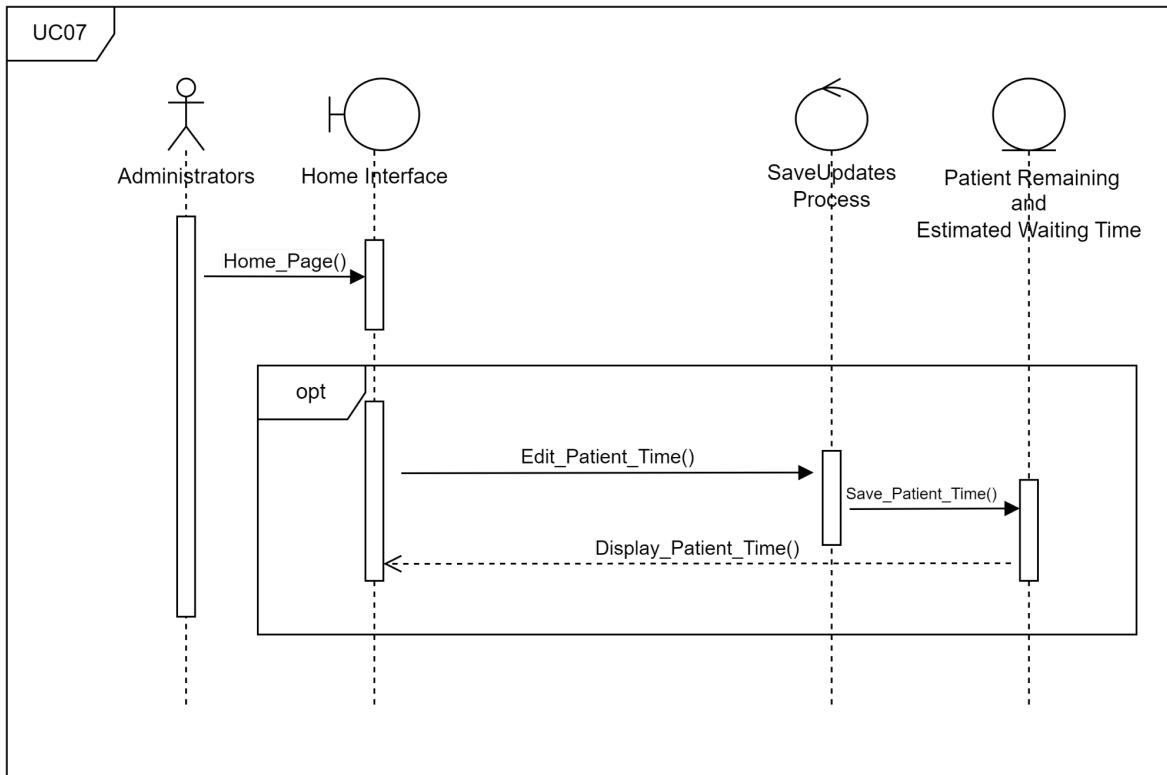


Figure 2.2.3.2: Sequence Diagram for <Key in Patient Estimated Waiting Time and people remained>

2.2.3.2 UC08: Use Case <View Patient Waiting Time and people remained>

Table 2.2.3.2.: Use Case Description for <View Patient Waiting Time and people remained>

Use case: <View Patient Waiting Time and people remained>
ID: UC08
Actors: <ul style="list-style-type: none"> 1. Student 2. Doctor 3. Admin
Preconditions: <ul style="list-style-type: none"> 1. Has active internet connection to the system. 2. Registered with the system. 3. Signed in system.
Flow of events: <ul style="list-style-type: none"> 1. “Patient Remaining” <ul style="list-style-type: none"> a. Number of patients can be viewed by users at the home interface. 2. “Estimated Time” <ul style="list-style-type: none"> a. Estimated waiting time can be viewed by users at the home interface. 3. The user case ends.
Postconditions: <ul style="list-style-type: none"> 1. Doctors and students can know the number of patients remaining and estimated waiting time needed before the next patient came in.
Alternative flow 1:

1. Doctor and Student can select the “Save” button to save the progress under an offline condition by storing the information in the application and device.

Postconditions:

1. Changes will be uploaded when the device is under an online condition.

Exception flow (if any):

1. Changes discarded after a few minutes.

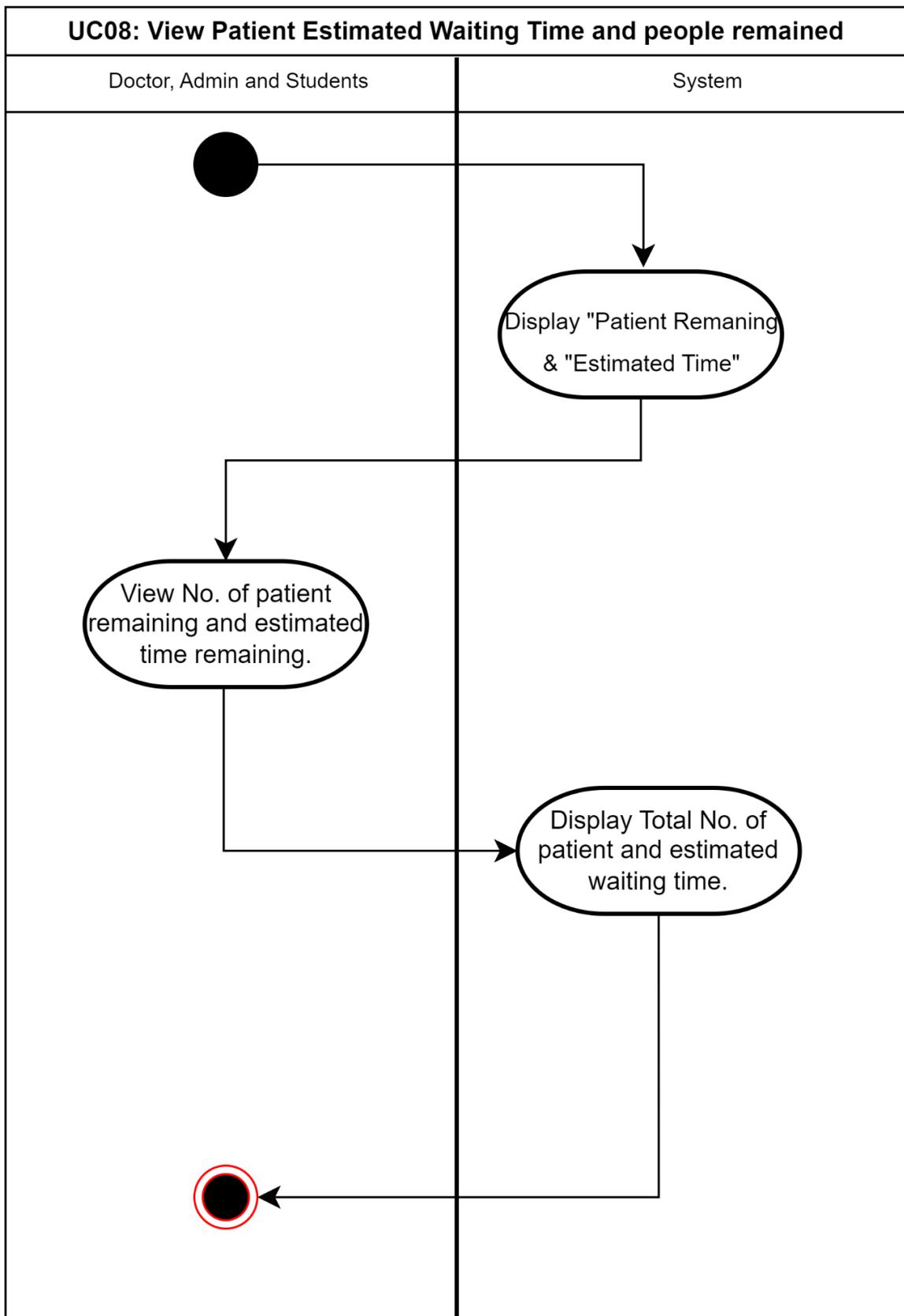


Figure 2.2.3.3: Activity Diagram for <View Patient Waiting Time and people remained>

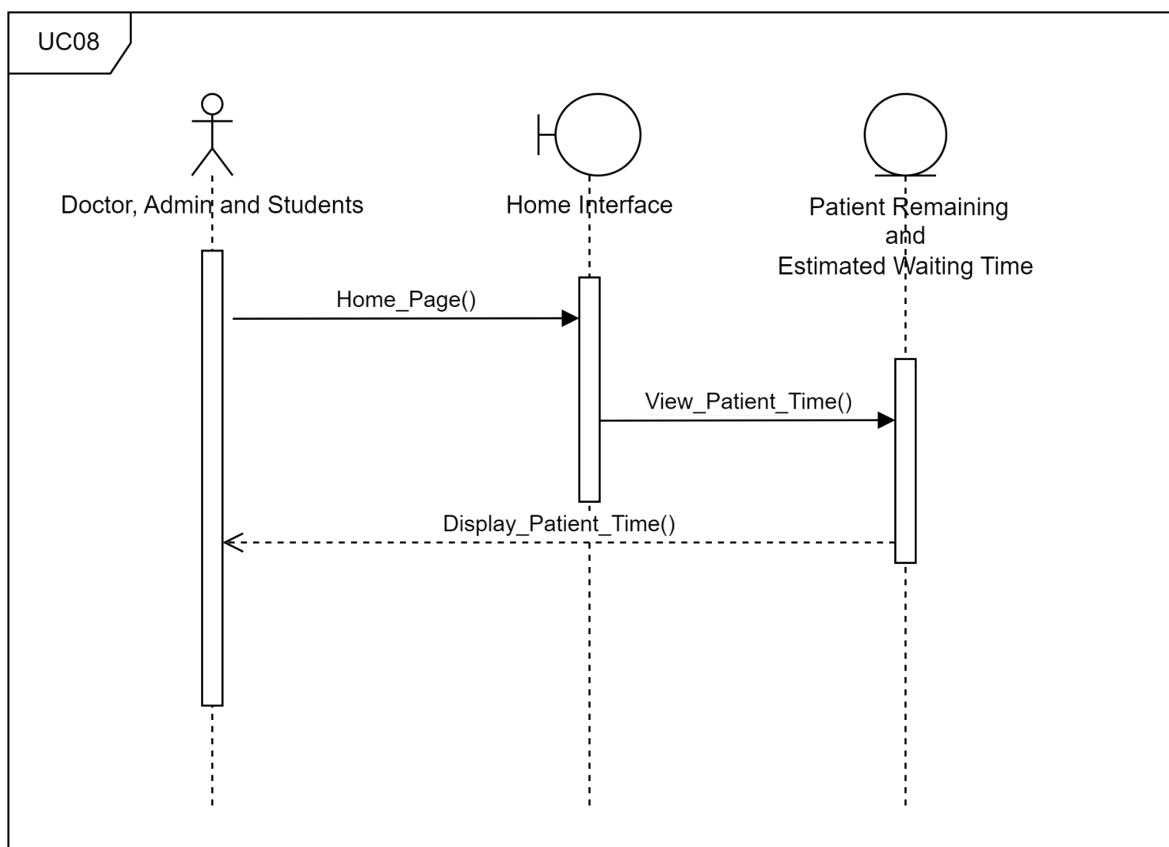


Figure 2.2.3.4: Sequence Diagram for <View Patient Waiting Time and people remained>

2.2.4 Module 04: <Health Tips System>

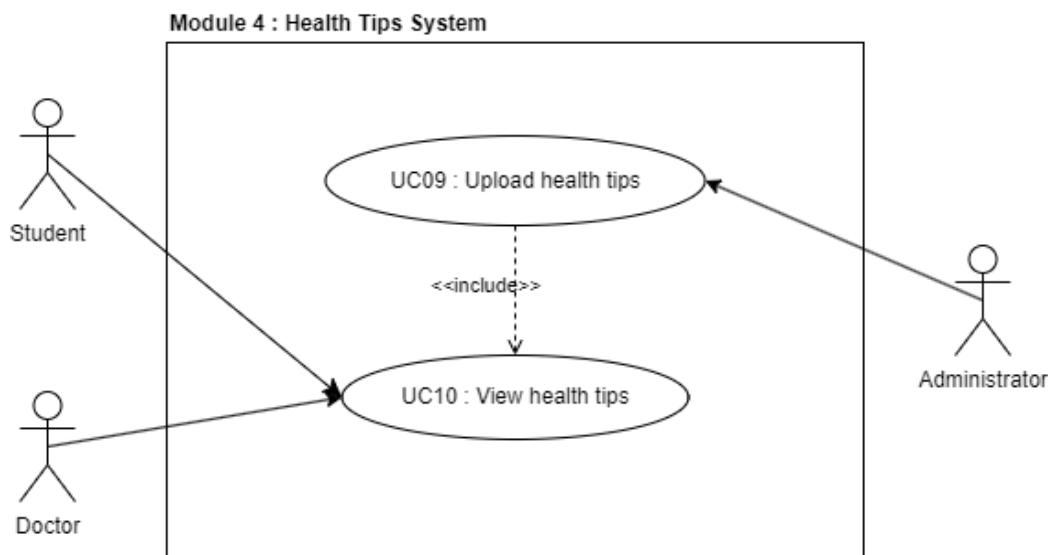


Diagram 2.2.4: Use Case Diagram of Submodule <Health Tips System>

2.2.4.1 UC09: Use Case <Upload health tips>

Table 2.2.4.1: Use Case Description for <Upload health tips>

Use case: <Upload health tips>
ID: UC09
Actors:
1. Administrator
Preconditions:
1. A valid administrator is logged on to the system with correct credentials.
Flow of events:
<ol style="list-style-type: none"> 1. Administrator selects the “Health Tips” button from the navigation bar. It will be directed to the “Health Tips” interface. 2. If the Administrator selects the “Edit” button. <ol style="list-style-type: none"> 2.1. Administrator arranges the health tips that have been uploaded.

- 2.2. Administrator selects the "Save" button to save the changes of arrangement of health tips.
3. If the Administrator selects the “Add” button. It will be directed to the “Add Health Tips” interface.
- 3.1. Administrator uploads the health tips.
 - 3.2. Administrator selects the "Save" button to save the uploaded health tips.
4. If the Administrator selects the brief introduction of the health tips. Details of the health tips will be displayed.
- 4.1. If the Administrator selects the "Edit" button.
 - 4.2. Administrator updates and modifies the health tips.
 - 4.3. Administrator selects the "Save" button to save the updated and modified health tips.

Postconditions:

The health tips information has been uploaded, arranged and saved.

Alternative flow 1:

Administrator can select the “Save” button to save the progress under an offline condition by storing the information in the application and device .

Postconditions:

Changes will be uploaded when the device is under an online condition.

Exception flow:

1. Administrator leaves the health tips editing interface without pressing the save button.
 - 1.1. Changes discarded after a few minutes.

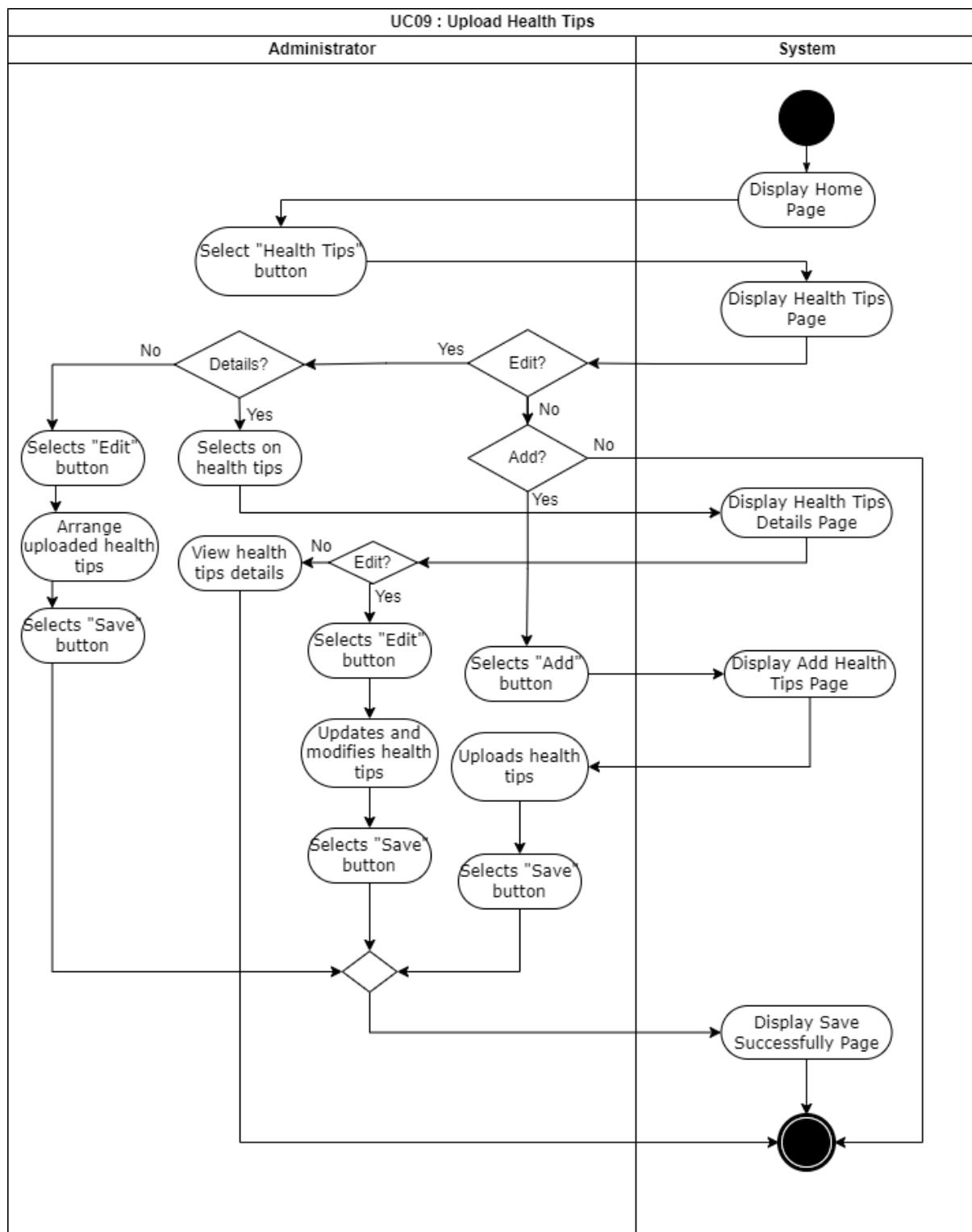


Figure 2.2.4.1: Activity Diagram for <Upload health tips>

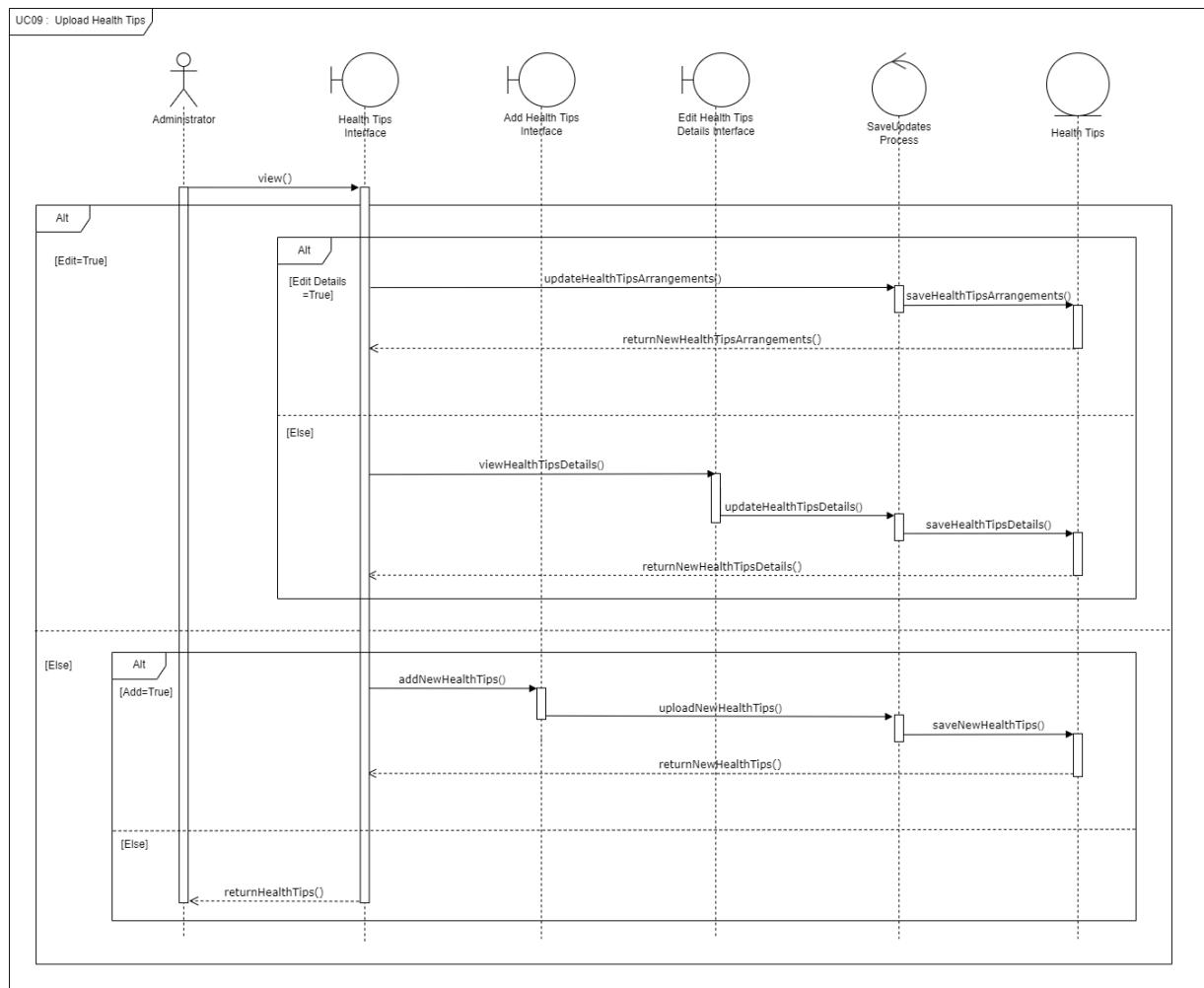


Figure 2.2.4.2: Sequence Diagram for <Upload health tips>

2.2.4.2 UC10: Use Case <View health tips>

Table 2.2.4.2: Use Case Description for <View health tips>

Use case: <View health tips>
ID: UC10
Actors: <ul style="list-style-type: none"> 1. Student 2. Doctor 3. Administrator
Preconditions: <ul style="list-style-type: none"> 1. A valid student is logged on to the system with correct credentials. 2. A valid doctor is logged on to the system with correct credentials. 3. A valid administrator is logged on to the system with correct credentials.
Flow of events: <ul style="list-style-type: none"> 1. Users select the “Health Tips” button from the navigation bar. It will be directed to the “Health Tips” interface. 2. Users select on the health tips to read further and detailed information.
Postconditions: The health tips information has been viewed.
Alternative flow 1: Users can select nothing and view the briefing of the health tips information at the “Health Tips” interface.
Postconditions: The health tips information has been briefly viewed.

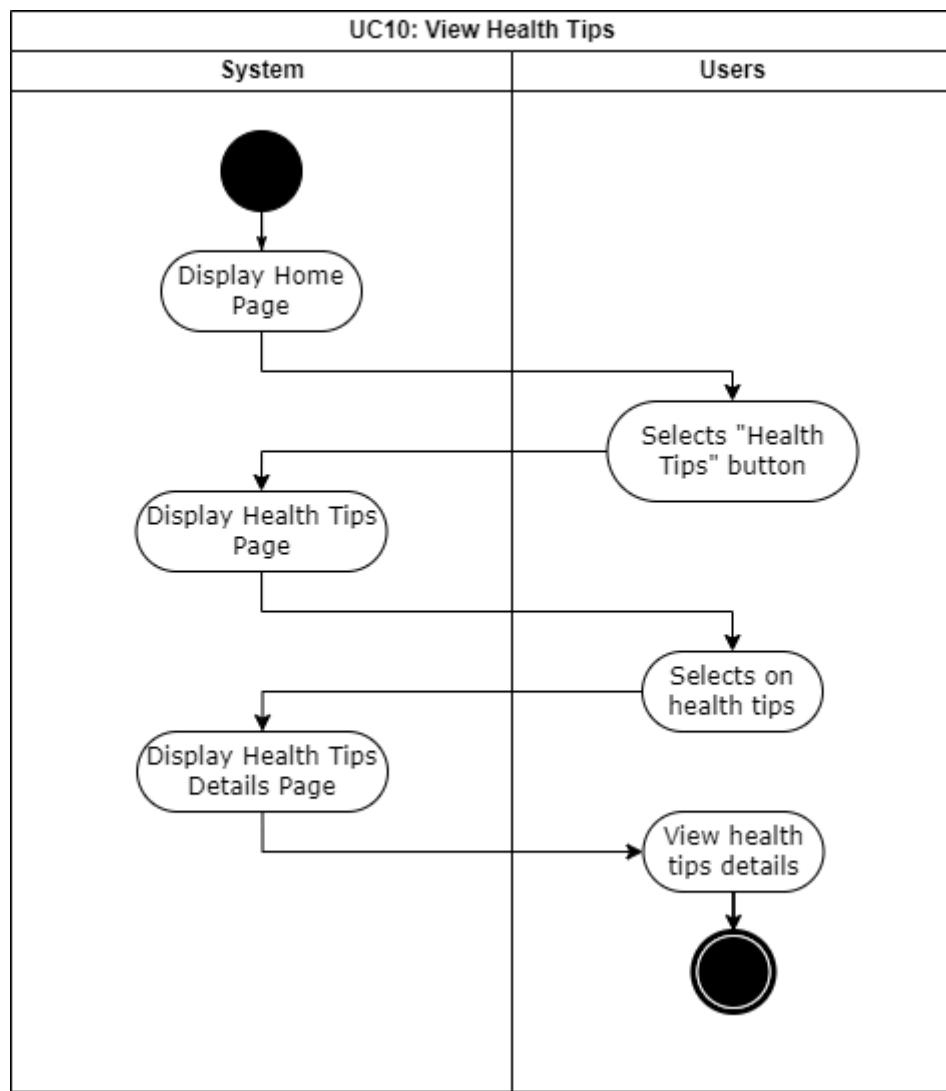


Figure 2.2.4.3: Activity Diagram for <View health tips>

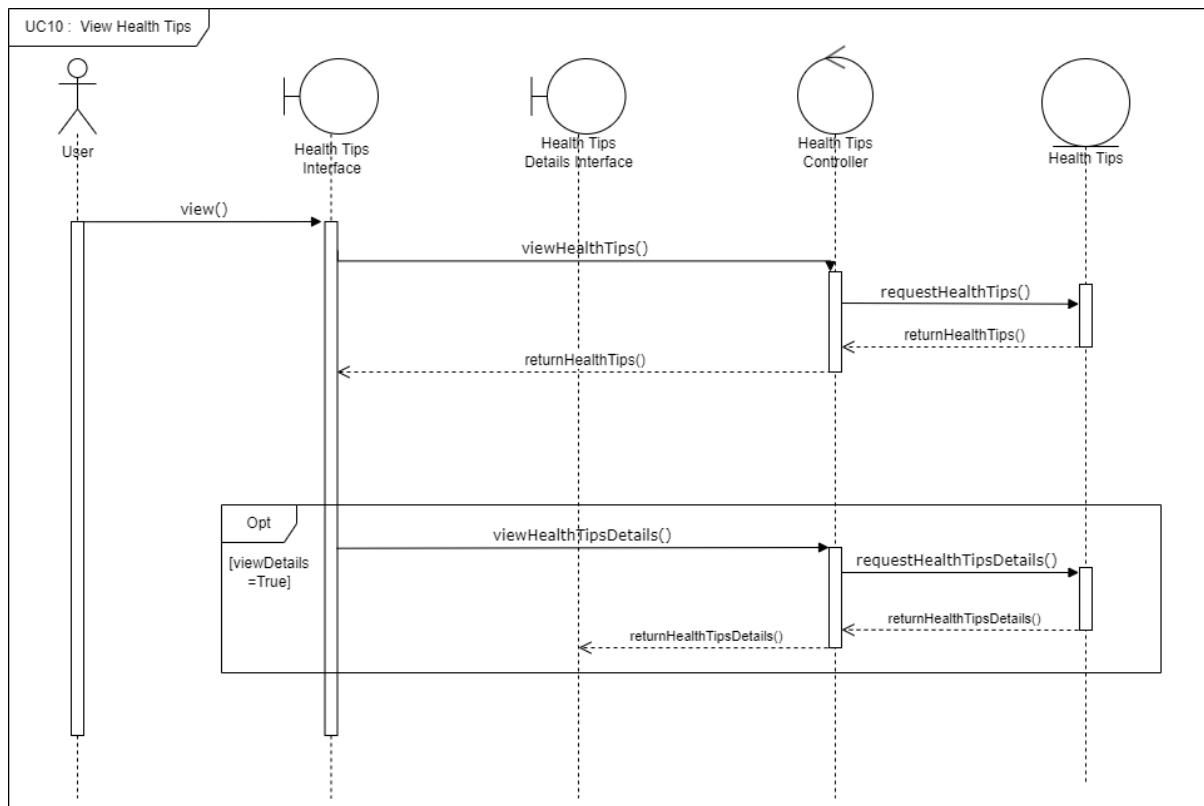


Figure 2.2.4.4: Sequence Diagram for <View health tips>

2.2.5 Module 05: <Mental Health Support System>

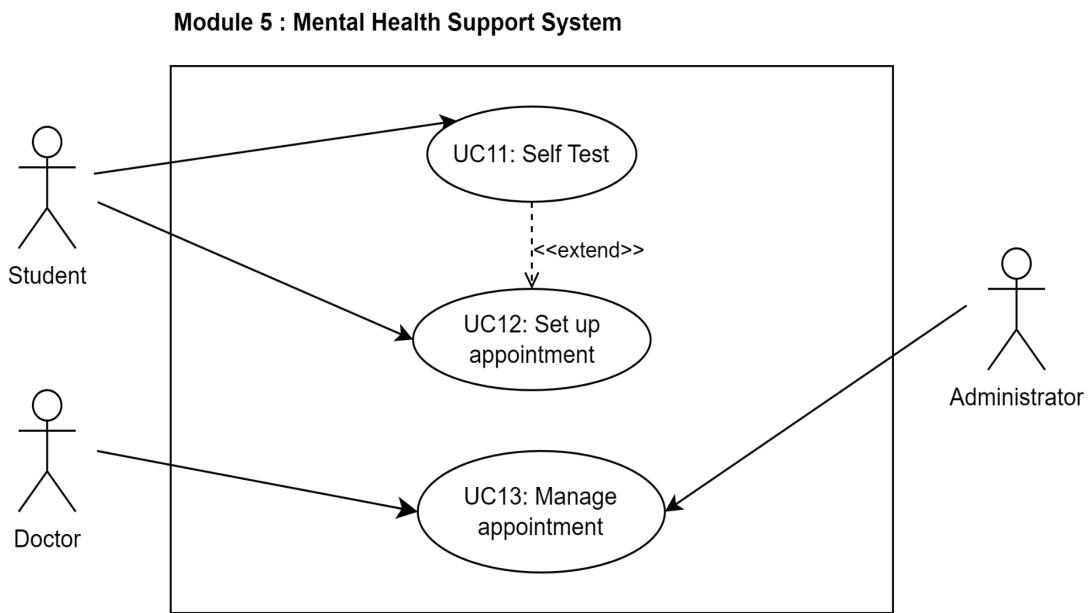


Diagram 2.2.5: Use Case Diagram of Submodule <Mental Health Support System>

2.2.5.1 UC11: Use Case <Self-Test>

Table 2.2.5.1: Use Case Description for <Self-Test>

Use case: <Self-Test>	
ID:	UC11
Actors:	1. Student
Preconditions:	1. A valid student is logged on to the system with correct credentials.
Flow of events:	1. Student selects the “Mental Health” button from the navigation bar. It will be directed to the “Mental Health” interface.

2. Student clicks the “Self-Test” button on the “Mental Health” interface.
3. Student answers all the questions from the DASS questionnaire.
4. Student clicks the “Submit” button after completing all the questions.
5. The system will display the self-test score that the student obtained for each section (D, A, S) and the levels of depression, anxiety and stress (Normal, Mild, Moderate, Severe, Extreme Severe) based on the score. The system will also display the DASS scoring guide.
6. If all of the levels of depression, anxiety or stress is normal, mild or moderate.
 - 6.1 The system will display the counselling information below the self-test score section.
7. If any of the levels of depression, anxiety or stress is severe or extremely severe.
 - 7.1 The system will display the counselling information along with a “Apply Appointment” button below the self-test score section.
 - 7.2 Go <Set up Appointment>.

Postconditions:

The self-test result will be stored in the user profile along with the date of the self-test.

Alternative flow 1:

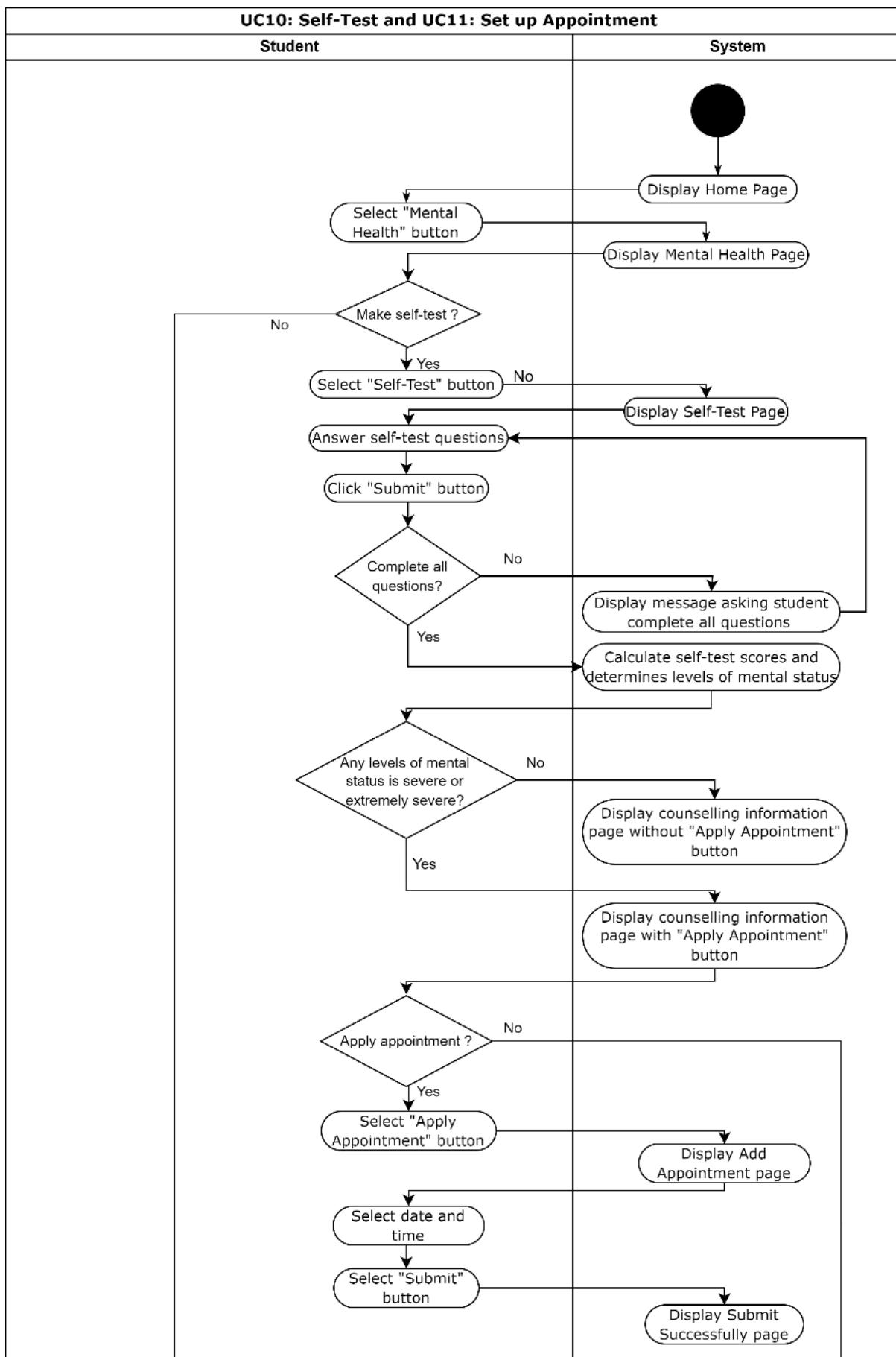
Student clicks the “Submit” button after completing all the questions under an offline condition.

Postconditions:

Self-test results will be uploaded when the device is under an online condition.

Exception flow (if any):

1. If the student clicks the “Submit” button without completing all the questions, a message will be displayed asking the student to answer all questions.
2. If the student leaves the “Self-Test” interface before successfully clicks the “Submit” button, answers to the questions will be discarded.



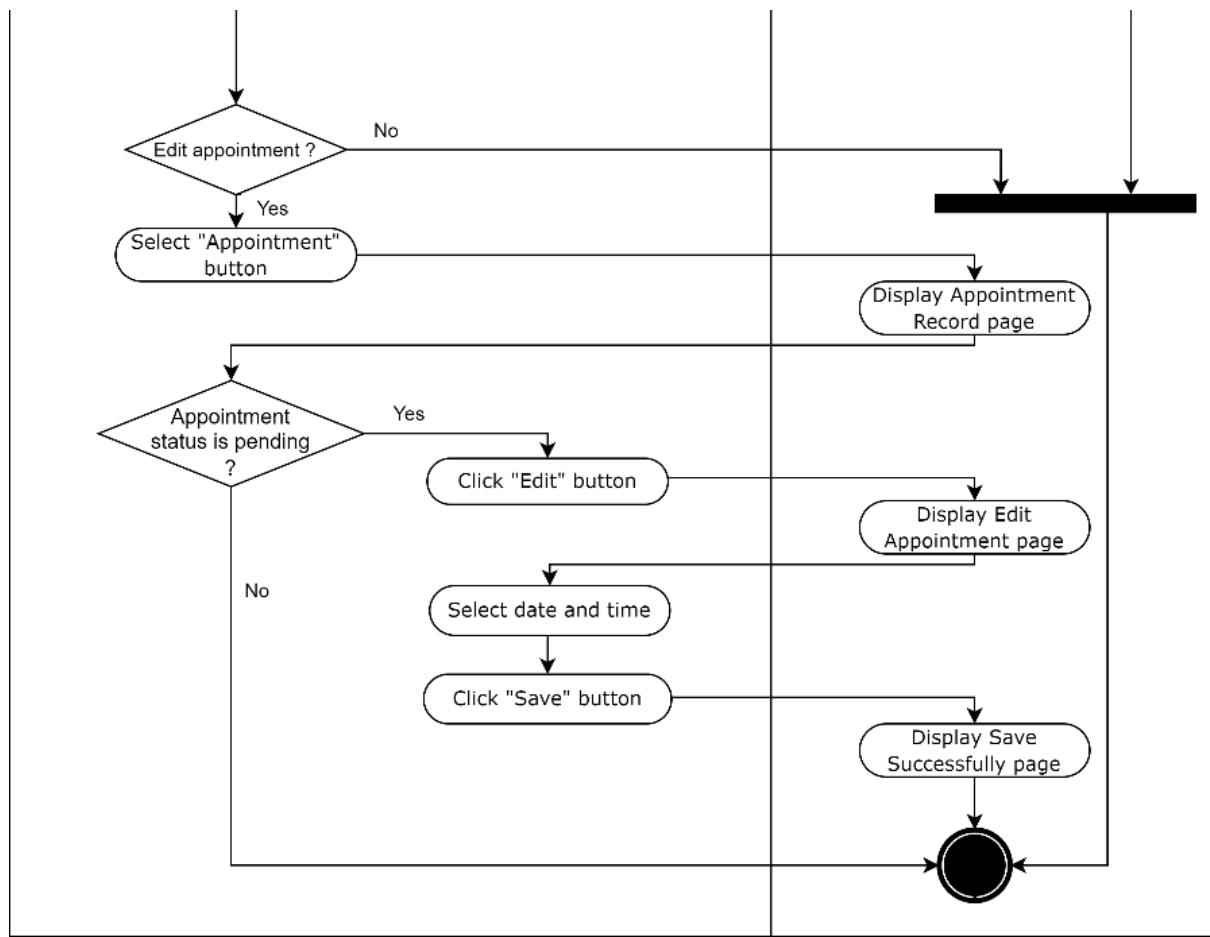


Figure 2.2.5.1: Activity Diagram for <Self-Test>

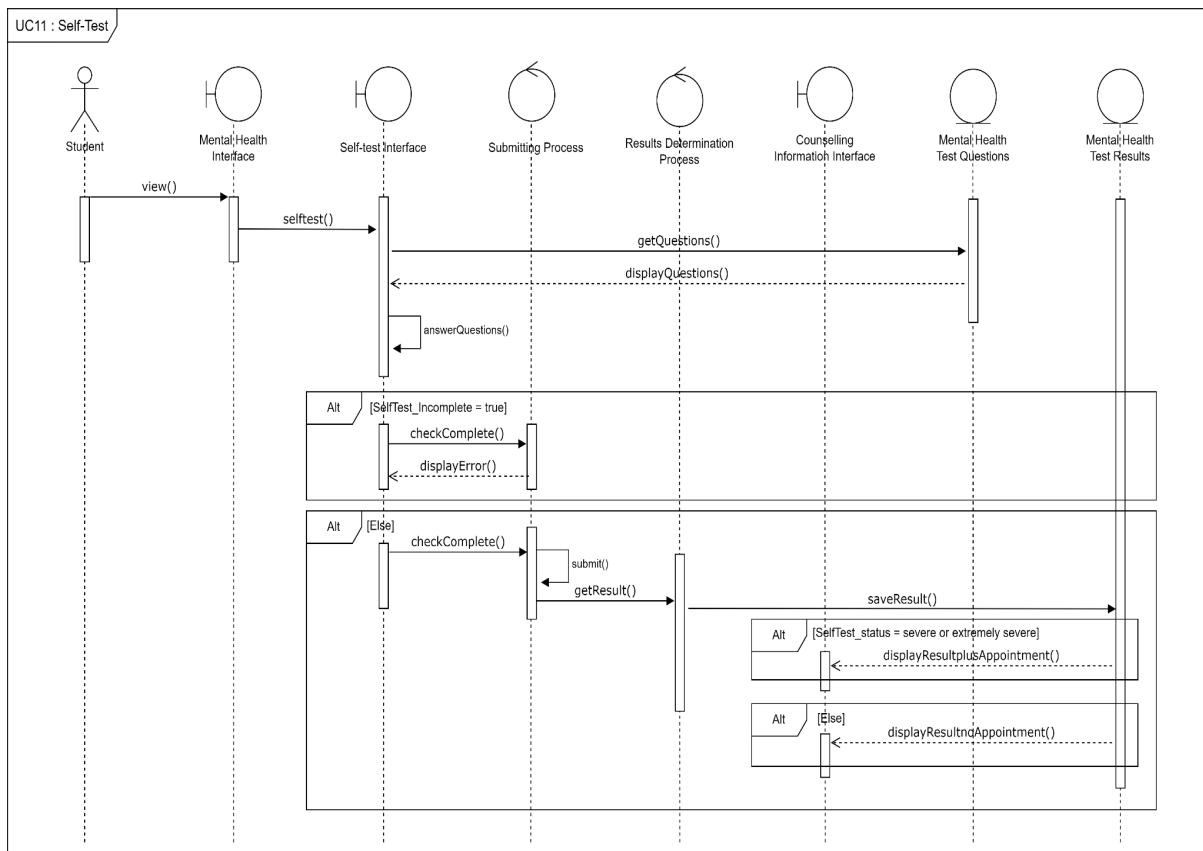


Figure 2.2.5.2: Sequence Diagram for <Self-Test>

2.2.5.2 UC12: Use Case <Set up appointment>

Table 2.2.5.2: Use Case Description for <Set up appointment>

Use case: <Set up appointment>
ID: UC12
Actors:
1. Student
Preconditions:
<ol style="list-style-type: none"> 1. A valid student is logged on to the system with correct credentials. 2. Student has answered the self-test, reached a severe or extremely severe level in one of the sections of depression, anxiety or stress and has clicked on the “Apply Appointment” button in the “Self-Test Result” interface.
Flow of events:
<ol style="list-style-type: none"> 1. At the “Add Appointment” interface, the student selects the date and time desired for an appointment. 2. Student selects the “Submit” button to apply for the appointment. 3. The system will display “Submit Successfully” and “Await for Approval”. 4. If the student goes back to the “Home” interface, clicks the “Mental Health” button and then clicks the “Appointment” button, it will be directed to the “Appointment Record” interface. <ul style="list-style-type: none"> 4.1 The system will display the details of appointment records 4.2 If the approval status is “Pending” and the student clicks the "Edit" button, it will be directed to the “Edit Appointment” interface.

<p>4.2.1 Student selects a new date to reschedule the appointment.</p> <p>4.2.2 Student selects a time to reschedule the appointment.</p> <p>4.2.3 Student selects the “Save” button to save the update of the rescheduling of the appointment.</p> <p>4.2.4 The system will display “Save Successfully” and “Await for Approval”.</p>
<p>Postconditions:</p> <ol style="list-style-type: none"> 1. The appointment will await doctor's approval. 2. When the appointment is approved, the system will show “Approved” at approval status. 3. When the appointment is declined, the system will show “Unsuccessful” at approval status. 4. The student can call the doctor's assistant if they want to reschedule after their appointment has been approved.
<p>Alternative flow 1:</p> <p>Student clicks the “Submit” button after selecting the date and time for the appointment under an offline condition.</p>
<p>Postconditions:</p> <p>Changes will be uploaded when the device is under an online condition.</p>
<p>Alternative flow 2:</p> <p>Student clicks the “Save” button after editing the date and time for the appointment under an offline condition.</p>
<p>Postconditions:</p> <p>Changes will be uploaded when the device is under an online condition.</p>

Exception flow (if any):

1. If the student leaves the “Apply Appointment” interface without pressing the “Submit” button, changes to the appointment date and time will be discarded.
2. If the student selects the date and time when editing the appointment without pressing the “Save” button, changes to the date and time will be discarded.

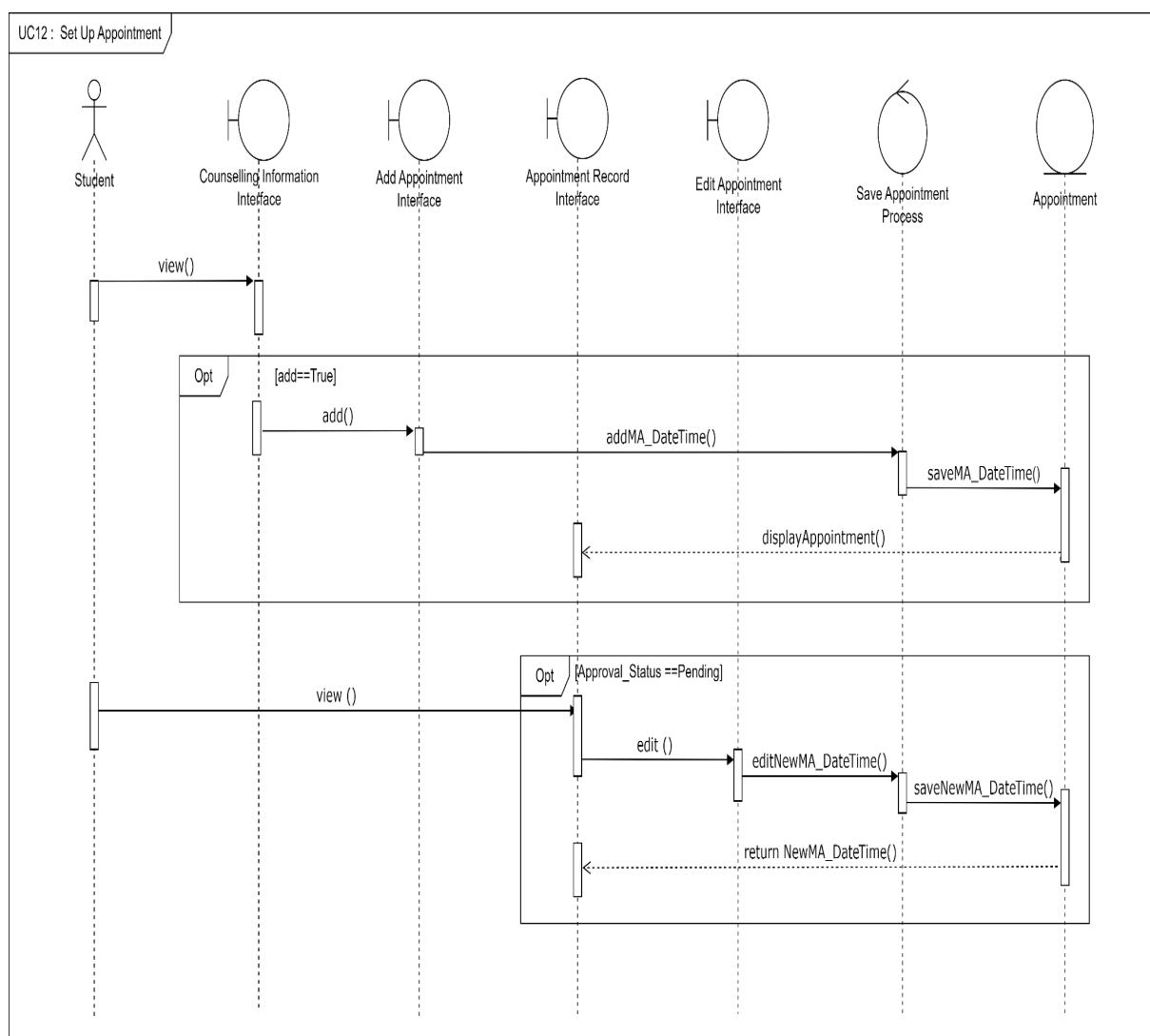


Figure 2.2.5.3: Sequence Diagram for <Set up appointment>

2.2.5.3 UC13: Use Case <Manage appointment>

Table 2.2.5.3: Use Case Description for <Manage appointment>

Use case: <Manage appointment>
ID: UC13
Actors: <ul style="list-style-type: none"> 1. Doctor 2. Administrator
Preconditions: <ul style="list-style-type: none"> 1. A valid doctor is logged on to the system with correct credentials. 2. A valid administrator is logged on to the system with correct credentials.
Flow of events: <ol style="list-style-type: none"> 1. Doctor or Administrator selects the “Mental Health” button from the navigation bar. It will be directed to the “Mental Health” interface. 2. Doctor or Administrator clicks the “Appointment” button on the “Mental Health” interface. 3. The system will display 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. 4. If the doctor or administrator clicks any of the appointments that are awaiting approval. <ol style="list-style-type: none"> 4.1 The system will display the details of the appointment. 4.2 If the doctor or administrator clicks the “Approve” button. <ol style="list-style-type: none"> 4.2.1 The appointment will disappear from the list of appointments that are

	<p>awaiting approval and display in the list of appointments that are already approved.</p>
4.3	<p>If the doctor or administrator clicks the “Decline” button.</p>
4.3.1	<p>The appointment will disappear from the list of appointments that are awaiting approval.</p>
5.	<p>If the doctor or administrator clicks any of the appointments that are already approved.</p>
5.1.	<p>The system will display the details of the appointment.</p>
5.2.	<p>If the doctor or administrator clicks the “Reschedule” button.</p>
5.2.1	<p>Doctor or Administrator selects the date to reschedule the appointment.</p>
5.2.2	<p>Doctor or Administrator selects the time to reschedule the appointment.</p>
5.2.3	<p>Doctor or Administrator clicks the “Save” button to save the update of the rescheduling of the appointment.</p>
5.2.4	<p>The system will display “Save Successfully”.</p>
5.3.	<p>If the doctor or administrator clicks the “Cancel” button.</p>
5.3.1	<p>The appointment will disappear from the list of appointments that have already been approved.</p>
5.3.2	<p>The system will display “Cancel Successfully”</p>
Postconditions:	

The appointment has been approved, rescheduled or cancelled.
Alternative flow 1: Doctor or Administrator clicks the “Approve”, “Decline” button under an offline condition.
Postconditions: Changes will be uploaded when the device is under an online condition.
Alternative flow 2: Doctor or Administrator clicks the “Save” button after selecting the date and time for rescheduling under an offline condition.
Postconditions: Changes will be uploaded when the device is under an online condition.
Exception flow (if any): If the doctor or administrator selects the date and time for rescheduling without pressing the “Save” button, changes to the date and time for rescheduling will be discarded.

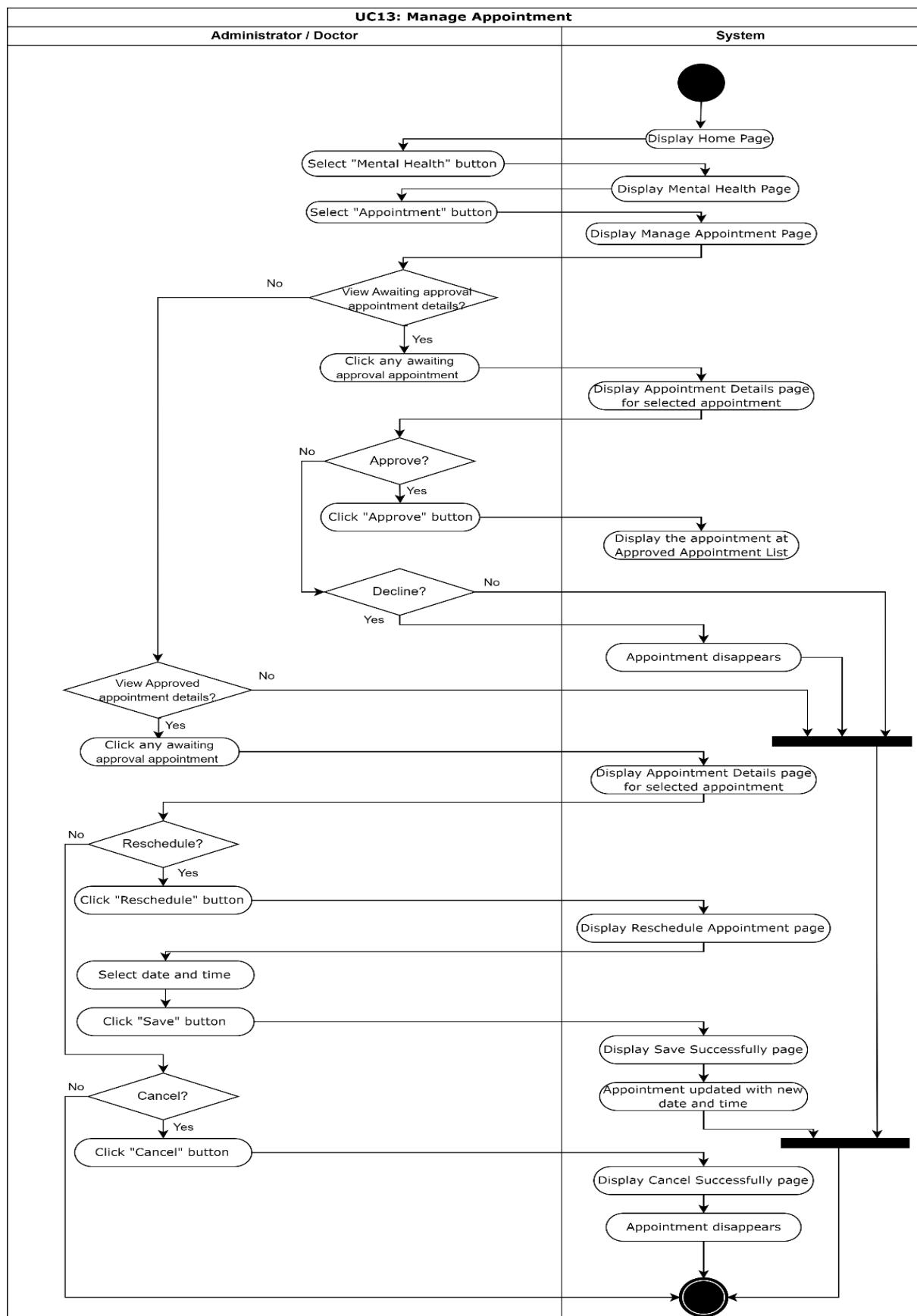


Figure 2.2.5.4: Activity Diagram for <Manage appointment>

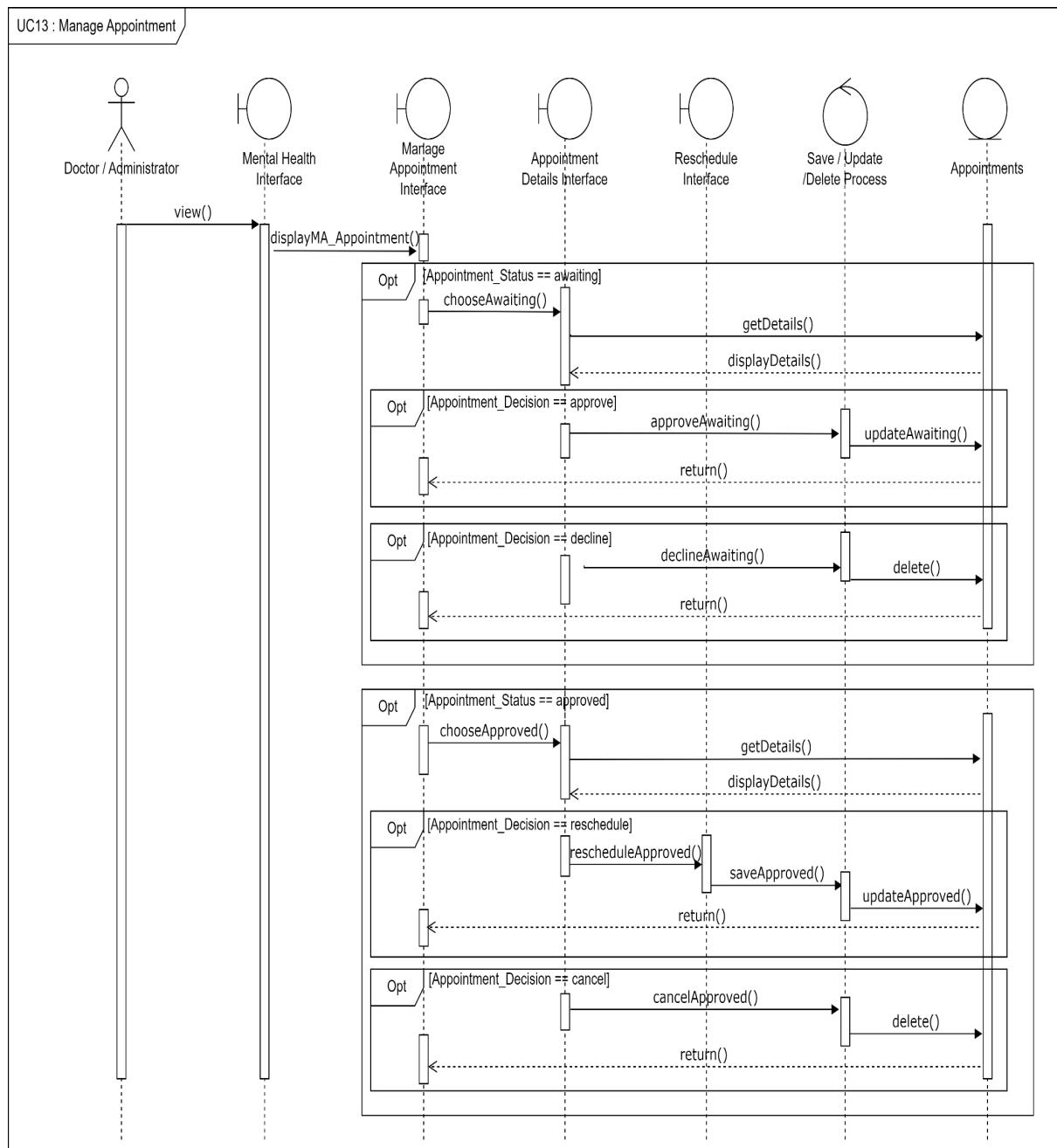


Figure 2.2.5.5: Sequence Diagram for <Manage appointment>

2.2.6 Module 06: <Dental Appointment System>

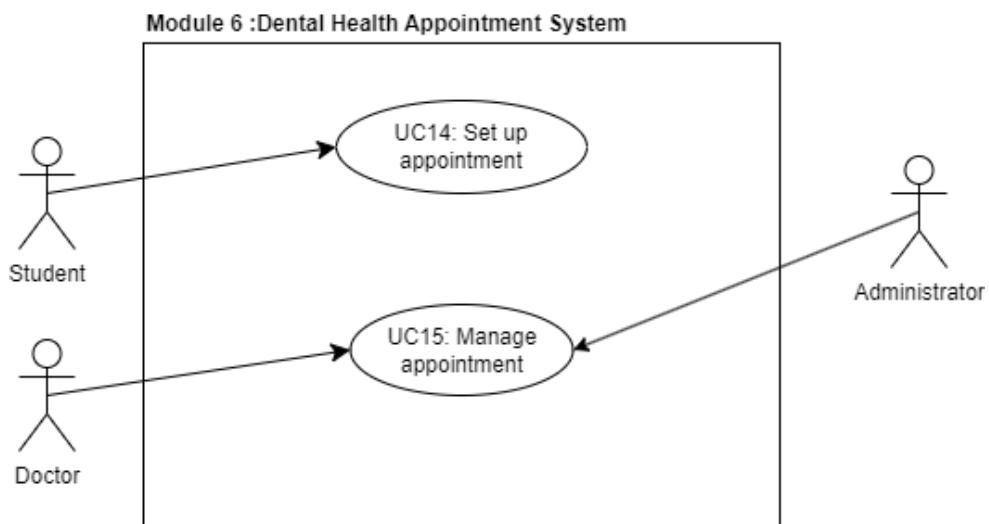


Diagram 2.2.6: Use Case Diagram of Submodule <Dental Health Appointment System>

2.2.6.1 UC14: Use Case <Set up appointment>

Table 2.2.6.1: Use Case Description for <Set up appointment>

Use case: <Set up appointment>
ID: UC14
Actors:
1. Student
Preconditions:
1. A valid student is logged on to the system with correct credentials.
Flow of events:
<ol style="list-style-type: none"> 1. Student selects the “Dental Health” button from the navigation bar. It will be directed to the “Dental Health” interface. 2. If the student selects the “Add Your Appointment” button. It will be directed to the “Add Appointment” interface.

- | |
|--|
| <p>2.1. Student selects the date and time desired for an appointment.</p> <p>2.2. Student selects the “Submit” button to apply for the appointment.</p> <p>2.3. The system will display “Submit Successfully” and “Await for Approval”.</p> <p>3. If the student selects the “Edit Your Appointment” button. It will be directed to the “Appointment Record” interface.</p> <p>3.1. The system will display the details of appointment records</p> <p>3.2. If the approval status is “Pending” and the student clicks the "Edit" button, it will be directed to the “Edit Appointment” interface.</p> <p>3.2.1. Student selects a new date to reschedule the appointment.</p> <p>3.2.2. Student selects a time to reschedule the appointment.</p> <p>3.2.3. Student selects the “Save” button to save the update of the rescheduling of the appointment.</p> <p>3.2.4. The system will display “Save Successfully” and “Await for Approval”.</p> |
|--|

Postconditions:

1. The appointment has been set up(added) or/and edited.
2. The appointment will await doctor's approval.
3. When the appointment is approved, the system will show “Approved” at approval status.
4. When the appointment is declined, the system will show “Unsuccessful” at approval status.

Alternative flow 1:

<p>Student clicks the “Submit” button after selecting the date and time for the appointment under an offline condition.</p>
<p>Postconditions:</p> <p>Changes will be uploaded and synchronised when the device is under an online condition.</p>
<p>Alternative flow 2:</p> <p>Student clicks the “Save” button after editing the date and time for the appointment under an offline condition.</p>
<p>Postconditions:</p> <p>Changes will be uploaded and synchronised when the device is under an online condition.</p>
<p>Exception flow :</p> <ol style="list-style-type: none"> 1. If the student leaves the “Add Appointment” interface without pressing the “Submit” button. <ol style="list-style-type: none"> 1.1. Changes to the appointment date and time will be discarded. 2. If the student selects the date and time when editing the appointment without pressing the “Save” button. <ol style="list-style-type: none"> 2.1. Changes to the date and time will be discarded.

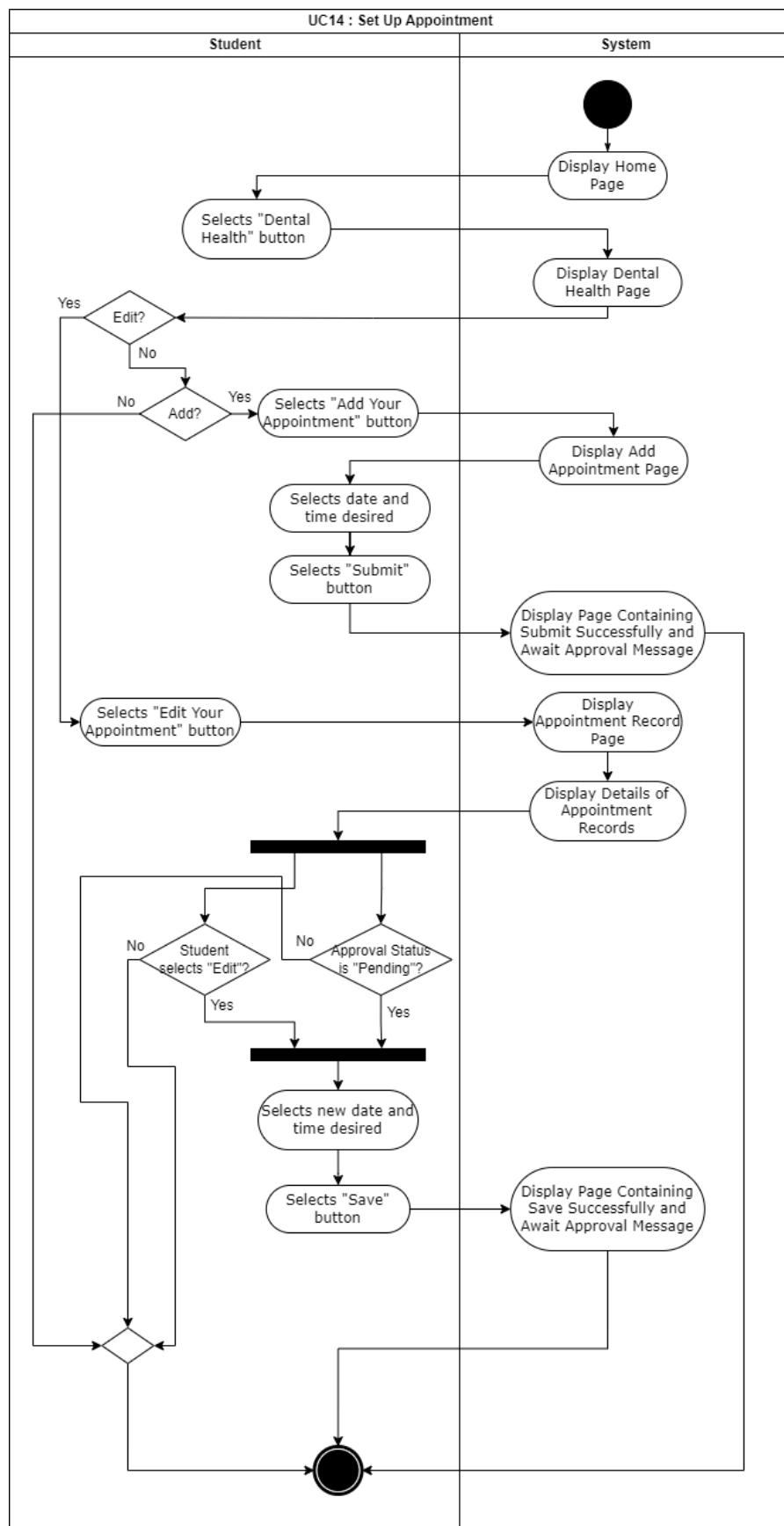


Figure 2.2.6.1: Activity Diagram for <Set up appointment>

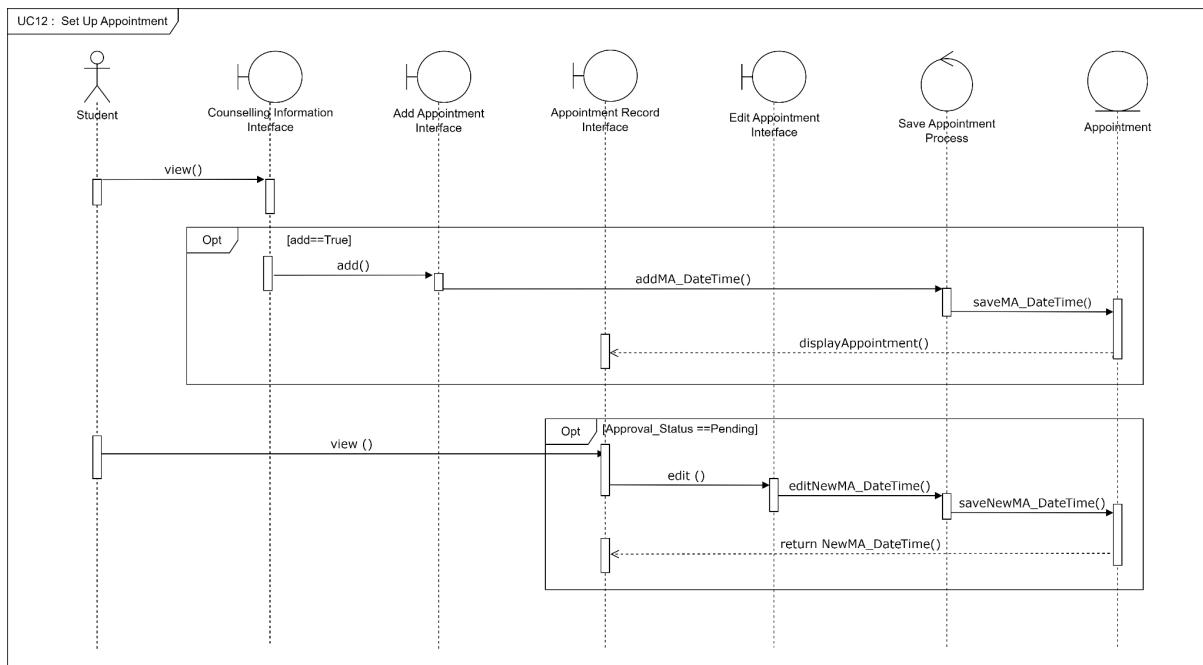


Figure 2.2.6.2: Sequence Diagram for <Set up appointment>

2.2.6.2 UC15: Use Case <Manage appointment>

Table 2.2.6.2: Use Case Description for <Manage appointment>

Use case: <Manage appointment>
ID: UC15
Actors: 1. Doctor 2. Administrator
Preconditions: 1. A valid doctor is logged on to the system with correct credentials. 2. A valid administrator is logged on to the system with correct credentials.
Flow of events: 1. Doctor or Administrator selects the “Dental Health” button from the navigation bar. It will be directed to the “Dental Health” interface. 2. Doctor or Administrator selects the “Appointment” button on the “Dental Health” interface. 3. The system will display 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. 4. If the doctor or administrator selects any of the appointments that are awaiting approval. 4.1. The system will display the details of the appointment. 4.2. If the doctor or administrator selects the “Approve” button. 4.2.1. The appointment will disappear from the list of appointments that are awaiting approval and be

	<p>displayed in the list of appointments that are already approved.</p> <p>4.3. If the doctor or administrator selects the “Decline” button.</p> <p>4.3.1. The appointment will disappear from the list of appointments that are awaiting approval.</p>
5.	<p>If the doctor or administrator selects any of the appointments that are already approved.</p> <p>5.1. The system will display the details of the appointment.</p> <p>5.2. If the doctor or administrator selects the “Reschedule” button.</p> <p>5.2.1. Doctor or Administrator selects the date to reschedule the appointment.</p> <p>5.2.2. Doctor or Administrator selects the time to reschedule the appointment.</p> <p>5.2.3. Doctor or Administrator clicks the “Save” button to save the update of the rescheduling of the appointment.</p> <p>5.2.4. The system will display “Save Successfully”.</p> <p>5.3. If the doctor or administrator selects the “Cancel” button.</p> <p>5.3.1. The appointment will disappear from the list of appointments that have already been approved.</p> <p>5.3.2. The system will display “Cancel Successfully”.</p>
	<p>Postconditions:</p> <p>The appointment has been approved, rescheduled or cancelled.</p>
	<p>Alternative flow 1:</p> <p>Doctor or Administrator selects the “Approve”, “Decline” button under an offline condition.</p>
	<p>Postconditions:</p>

<p>Changes will be uploaded and synchronised when the device is under an online condition.</p>
<p>Alternative flow 2:</p> <p>Doctor or Administrator selects the “Save” button after selecting the date and time for rescheduling under an offline condition.</p>
<p>Postconditions:</p> <p>Changes will be uploaded and synchronised when the device is under an online condition.</p>
<p>Exception flow :</p> <ol style="list-style-type: none"> 1. Doctor or Administrator selects the date and time for rescheduling without pressing the “Save” button. <ol style="list-style-type: none"> 1.1. Changes to the date and time for rescheduling will be discarded.

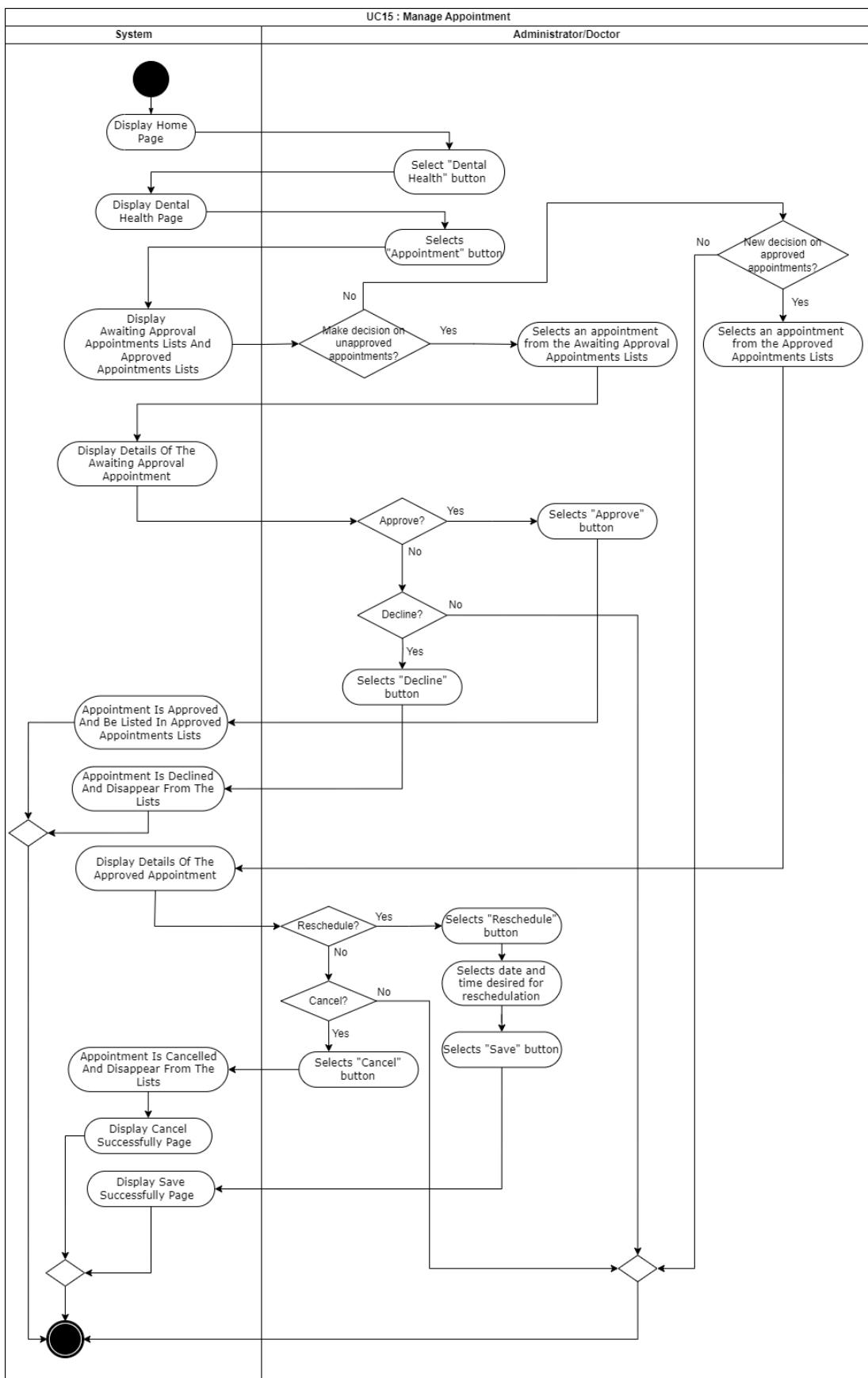


Figure 2.2.6.3: Activity Diagram for <Manage appointment>

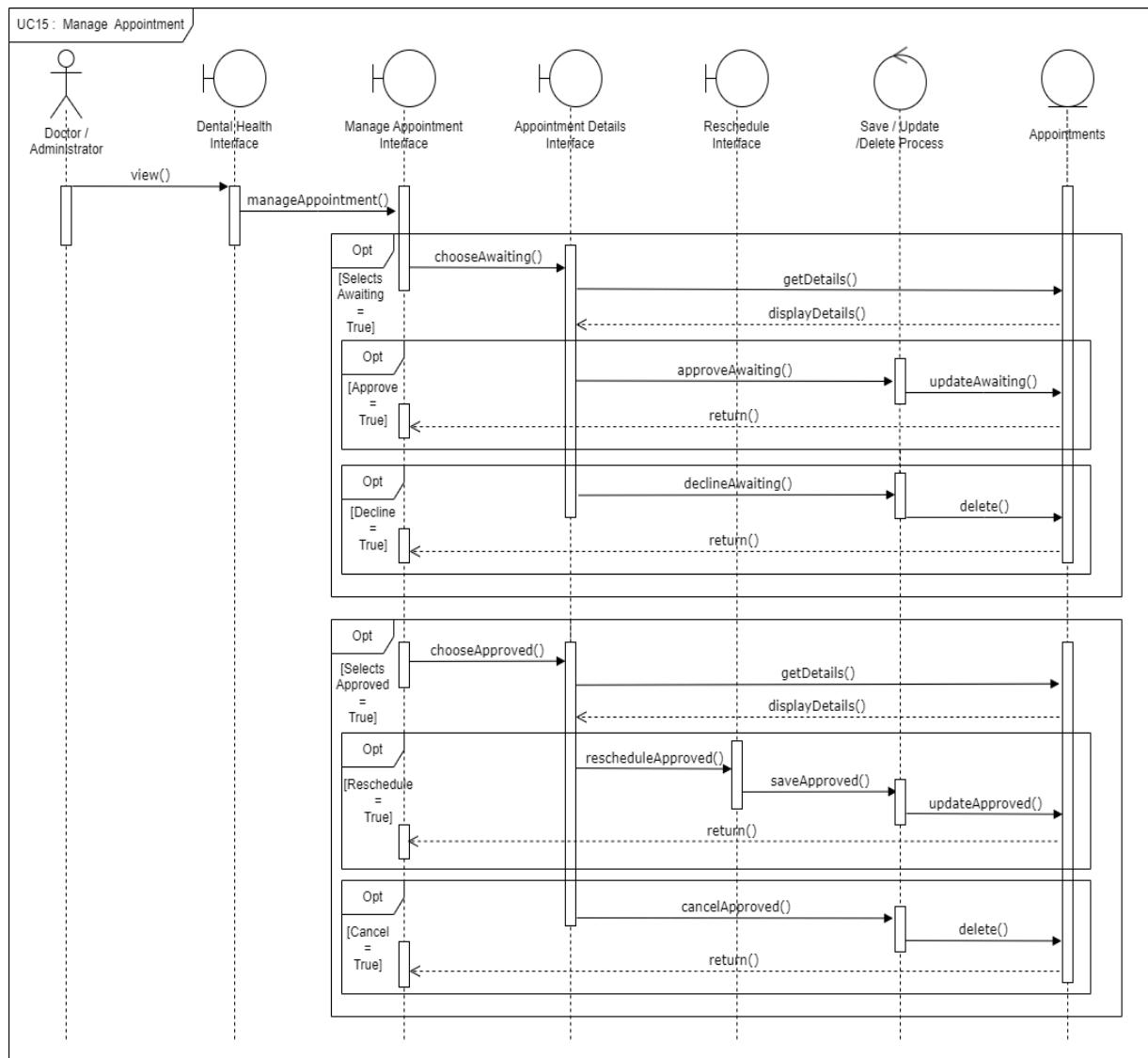


Figure 2.2.6.4: Sequence Diagram for <Manage appointment>

2.2.7 Module 07: <Feedback System>

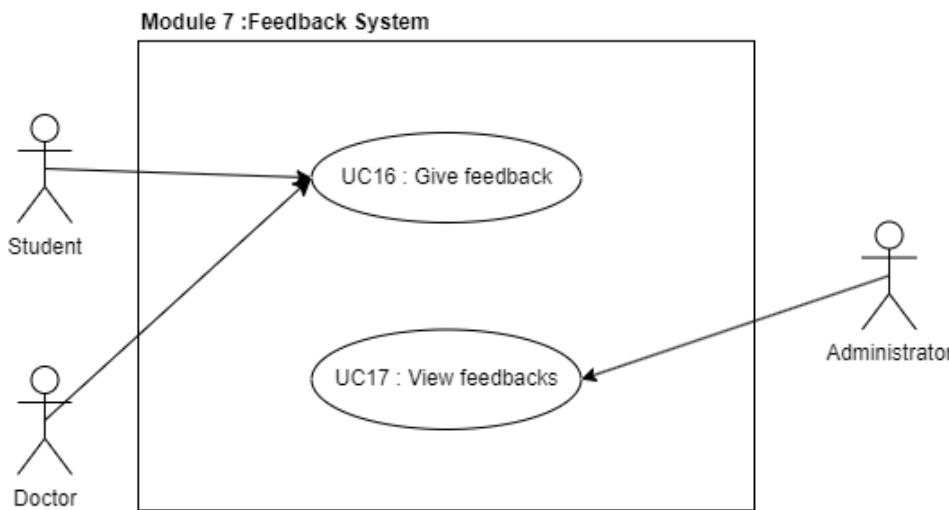


Diagram 2.2.7: Use Case Diagram of Submodule <Feedback System>

2.2.7.1 UC16: Use Case <Give Feedbacks>

Table 2.2.7: Use Case Description for <Give Feedbacks>

Use case: <Give Feedbacks>
ID: UC16
Actors: 1. Students 2. Doctors
Preconditions: 1. Has active internet connection to the system. 2. Registered with the system. 3. Signed in system.
Flow of events: 1. Doctors and students navigate to the bottom of the home page. 2. “Feedbacks” blank box is provided to fill in a short answer or paragraph for doctors and students.

3. Doctors and students tap and fill.
 4. Doctors and students submit the feedbacks
 - a. Tap “Submit”.
 5. User case end.
- ...

Postconditions:

1. Students and doctors are available to give feedback about servicing and challenges.

Alternative flow 1:

1. Students and doctors can select the “Save” button to save the progress under an offline condition by storing the information in the application and device .

Postconditions:

1. Doctor or student selects the date and time for rescheduling without pressing the “Save” button.

Exception flow (if any):

1. Changes to the date and time for rescheduling will be discarded.

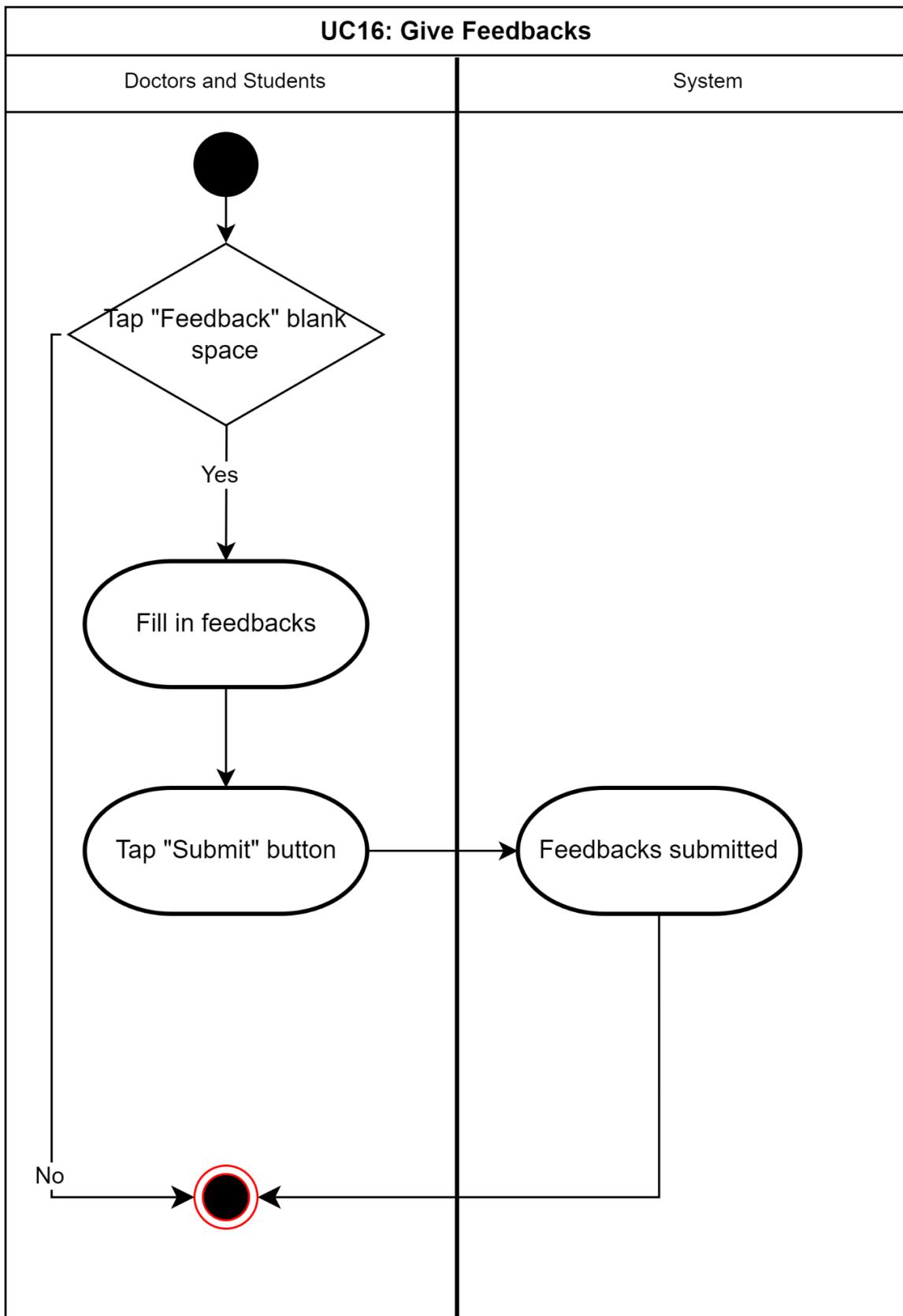


Figure 2.2.7.1: Activity Diagram for <Give Feedbacks>

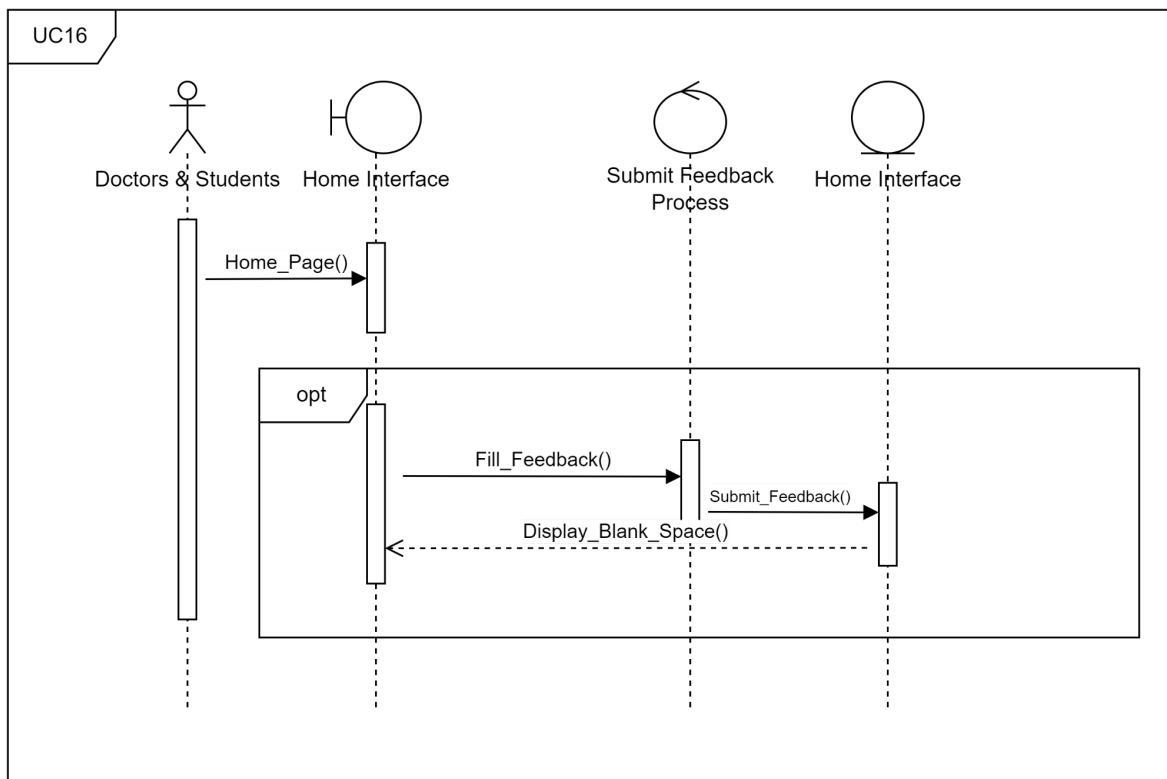


Figure 2.2.7.2: Sequence Diagram for <Give Feedbacks>

2.2.7.2 UC17: Use Case <View Feedbacks>

Table 2.2.7.2: Use Case Description for <View Feedbacks>

Use case: <View Feedbacks>
ID: UC17
Actors: <ul style="list-style-type: none"> 1. Administrator
Preconditions: <ul style="list-style-type: none"> 1. Has active internet connection to the system. 2. Registered with the system. 3. Signed in system.
Flow of events: <ul style="list-style-type: none"> 1. Administrators tap the home page dashboard “Feedbacks” button. 2. Administrators tap and navigate to a new page to view a feedback list that provides feedback, role, time and date submitted. 3. Administrator type keywords to search or filter with date, time, role for the feedback. 4. Administrators can view the feedback they want. 5. User case end. <p>...</p>
Postconditions: <ul style="list-style-type: none"> 1. Administrator able to allocate the problems of the system or service from the feedback.
Alternative flow n: <ul style="list-style-type: none"> 1. Administrators can select the “Save” button to save the progress under an offline condition by storing the information in the application and device .

Postconditions:

1. Administrator selects the date and time for rescheduling without pressing the “Save” button.

Exception flow (if any):

1. Changes to the date and time for rescheduling will be discarded.

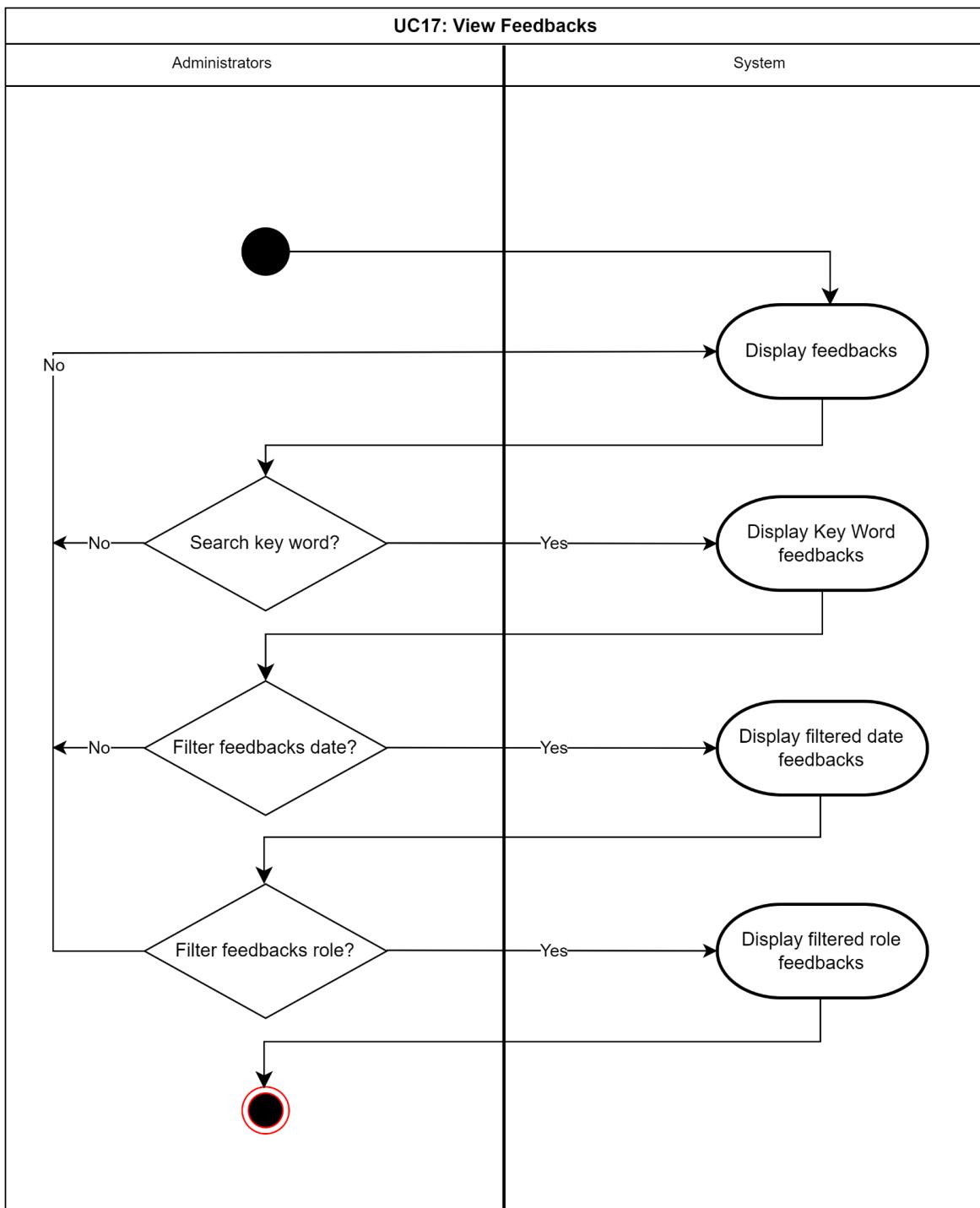


Figure 2.2.7.3: Activity Diagram for <View Feedbacks>

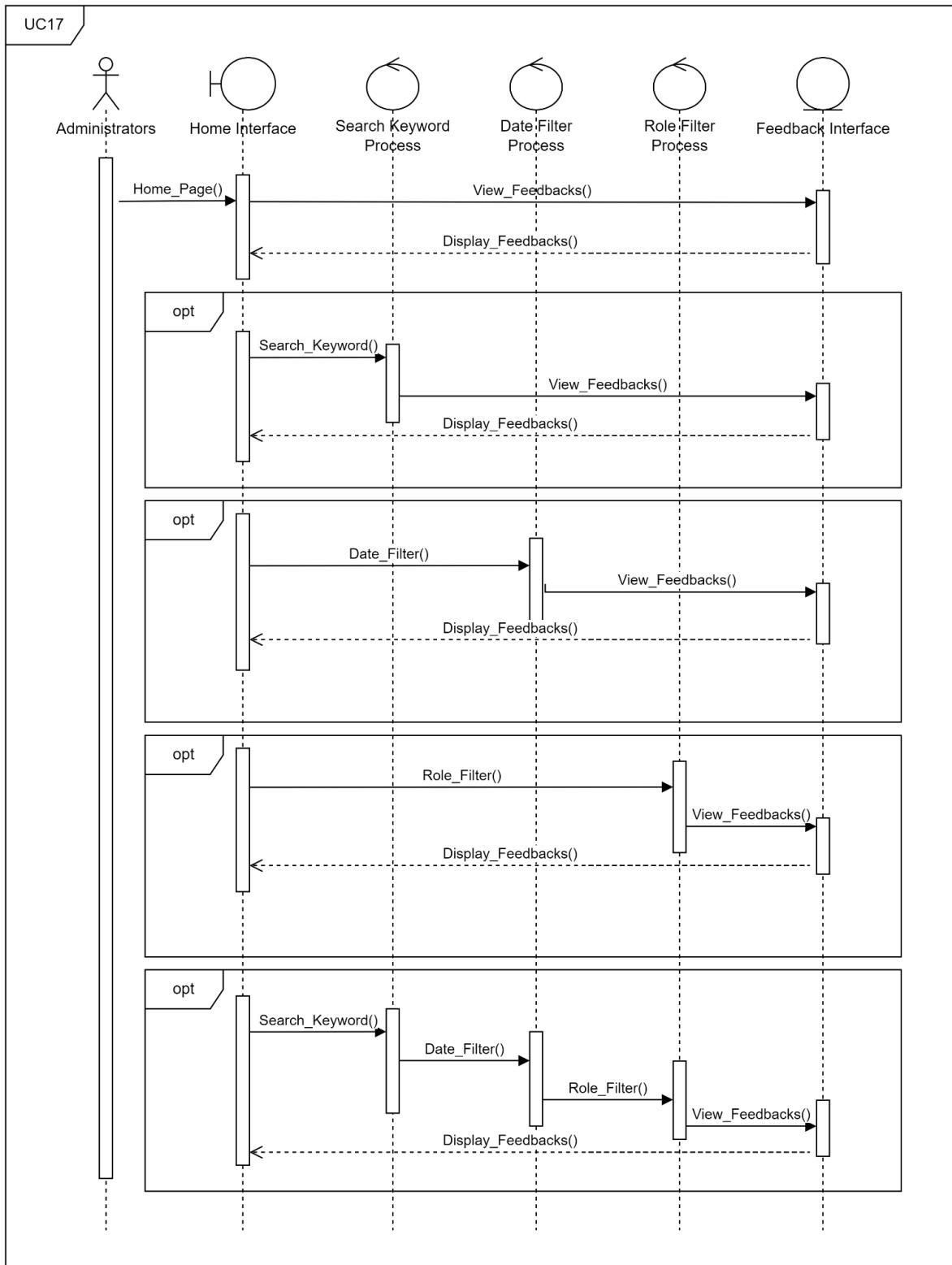


Figure 2.2.7.4: Sequence Diagram for <View Feedbacks>

2.3 Software System Attributes, Performance and Other Requirements

2.3.1 Software System Attributes

Software system attributes, also known as software quality attributes or software characteristics, describe the overall qualities or characteristics of the software system. These attributes are crucial for ensuring that the system meets the stakeholders' needs and expectations. These attributes will help to improve user requirements & satisfaction, clearer software design, and ultimately greater end product quality. Here are some software system attributes that need to be followed along this software production:

1. Usability : The system should have an intuitive and user-friendly interface, making it easy for students to navigate, book appointments, access information, and utilise health tracking tools.
2. Reliability : The system should be robust and reliable, ensuring that it performs its intended functions accurately and consistently. It should minimise errors, crashes, and downtime, providing a stable and dependable user experience.
3. Maintainability: The software system should be designed in a modular and well-structured manner, making it easy to update, modify, and maintain. It should have clear documentation and code organisation to facilitate future enhancements or bug fixes.
4. Portability: The system should be developed using technologies and frameworks that support cross-platform compatibility. It should be able to run on different web browsers and operating systems commonly used by students and PKU.
5. Compatibility: The software system should integrate seamlessly with existing systems and databases used by PKU and other relevant school organisations. It should have interoperability with external systems for data exchange and integration.

2.3.2 Software System Performance

Performance requirements define the system's capability to respond to user requests and handle data efficiently. They address the system's responsiveness, throughput, capacity, and availability. Here are some key performance-related requirements:

1. Response Time: The system should have fast response times, ensuring that user requests are processed and displayed within an acceptable time frame. Actions such as submitting health forms, viewing medical records, and accessing health resources should be prompt.
2. Throughput: The system should be able to handle a large number of simultaneous users without significant degradation in performance. It should support the expected user load during peak usage times, such as registration periods or when critical health information is released.
3. Capacity: The system should have sufficient capacity to store and manage a substantial volume of student health data, including medical records, health assessments, and wellness resources. It should scale efficiently to accommodate future growth in user numbers and data size.
4. Availability: The system should be highly available, with minimal planned or unplanned downtime. It should be accessible to users 24/7, with appropriate backup and recovery mechanisms to ensure data integrity and system availability.

2.3.3 Other Requirements in Software

1. Security: The system should employ robust security measures to protect sensitive student health data, adhering to industry best practices and relevant data protection regulations. It should implement user authentication, encryption, access controls, and audit logs to ensure confidentiality and prevent unauthorised access.
2. Safety: The system should be designed to promote the safety and well-being of students. It should provide accurate and reliable health information, guidance, and alerts. It should also facilitate emergency communication and provide access to relevant emergency contact information.
3. Legal and Regulatory: The system should comply with applicable laws, regulations, and standards related to data privacy, health information management, and accessibility. It should follow guidelines set by regulatory bodies such as HIPAA (Health Insurance Portability and Accountability Act) or GDPR (General Data Protection Regulation).
4. Environmental: The system should have minimal impact on the environment, such as by optimising resource usage and minimising energy consumption. It should also promote eco-friendly practices, such as reducing paper usage through digital forms and document management.

2.4 Design Constraints

2.4.1 Environmental Constraints

1. Power supply: The student health and management system designed to have high compatibility and work efficiently with the existing power infrastructure at KTDI and PKU UTM. Ensure no excessive consumption of electrical systems.
2. Network connectivity: The student health and management system should be able to connect to the existing network infrastructure at KTDI and PKU UTM to ensure seamless communication and data exchange.

2.4.2 Hardware Constraints

1. Device compatibility: The student health and management system should be compatible with different types of hardware devices available to students, doctors, and administrators such as computers, laptops, tablets, and smartphones.
2. Storage capacity: The student health and management system should avoid excessive consumption of storage space by providing efficient data storage mechanisms to and limit the storage of the devices.
3. Processing power: The student health and management system should be optimised to run on the available hardware devices to maintain and improve significant performance.
4. Input/output devices: The student health and management system should be compatible with the available input/output devices at KTDI and PKU UTM. For example, keyboards, mice, touchscreens, printers, and scanners.

2.4.3 Security Constraints

1. User authentication: The student health and management system should provide secure authentication mechanisms to verify the identities of students, doctors, and administrators while preventing unauthorised access to the system.
2. Data confidentiality: The student health and management system should ensure that only authorised personnel have access to confidential student health data.
3. Secure communication: The student health and management system should protect data during transmission between different system components by using secure protocols and encryption techniques to protect data during transmission between different system components.

2.4.4 Compatibility Constraints

1. Data exchange standards: The student health and management system should adhere to standard data exchange formats and protocols to facilitate interoperability and data sharing with external systems or databases.
2. Software compatibility: The student health and management system should be designed to work with different operating systems and software versions used by students, doctors, and administrators, ensuring compatibility across various platforms.

These constraints should be considered during the design and implementation of the student health and management system to ensure it meets the specific requirements and limitations of the hostel and clinic environment.