

KIKI: A CHATBOT APPLICATION FOR
OCD
IN ANDROID MOBILE APPLICATION

KONG KEI

BACHELOR OF COMPUTER SCIENCE
(SOFTWARE ENGINEERING) WITH
HONOURS

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : KONG KEI

Date of Birth

Title : KIKI: A CHATBOT APPLICATION FOR OCD
IN ANDROID MOBILE APPLICATION

Academic Session : SEMESTER I ACADEMIC SESSION 2022/2023

I declare that this thesis is classified as:

- CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)*
- RESTRICTED (Contains restricted information as specified by the organization where research was done)*
- OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

(Student's Signature)

(Supervisor's Signature)

New IC/Passport Number
Date: 08/02/2023

Dr. Anis Farihan Binti Mat Raffei
Name of Supervisor
Date: 08/02/2023

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Perpustakaan Universiti Malaysia Pahang,
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name
Thesis Title

Reasons	(i)
	(ii)
	(iii)

Thank you.

Yours faithfully,

(Supervisor's Signature)

Date: 08/02/2023

Stamp:

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.



SUPERVISOR'S DECLARATION

I/~~We~~* hereby declare that I/~~We~~* have checked this thesis/project* and in my/~~our~~* opinion, this thesis/project* is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Software Engineering) With Honours.

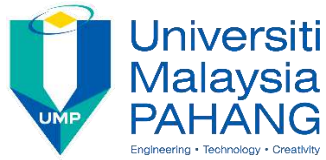
A handwritten signature in blue ink, appearing to read 'Anis Farihan Binti Mat Raffei', is written above a horizontal line.

(Supervisor's Signature)

Full Name : Dr. Anis Farihan Binti Mat Raffei

Position : Senior Lecturer

Date : 08/02/2023



STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

A handwritten signature in black ink, appearing to be 'KONG KEI', is written above a horizontal line.

(Student's Signature)

Full Name : KONG KEI

ID Number : CB19110

Date : 08/02/2023

KIKI: A CHATBOT APPLICATION FOR OCD
IN ANDROID MOBILE APPLICATION

KONG KEI

Thesis submitted in fulfillment of the requirements
for the award of the
Bachelor of Computer Science (SOFTWARE ENGINEERING)

Faculty of Computing
UNIVERSITI MALAYSIA PAHANG

JANUARY 2023

ACKNOWLEDGEMENTS

First of all, I would like to say thank you to my supervisor for this project, Dr. Anis Farihan Binti Mat Raffei, for willing to be my supervisor and give a lot of patients, guidance and advise to me to complete the project.

Next, I want to say thank you to my parents and family member who always support and encourage me during the project. Thank you for supporting me physically and mentally.

I would like to say thank you to the online community that create or share many useful tutorial or solution that help me a lot while I am facing problems in this project.

Last but not least, I would like to say thank you to all my dear friend for giving me advice, motivate me and helping me while the project is being developed.

ABSTRAK

Ramai orang mengalami masalah kesihatan mental. Walau bagaimanapun, jika gejala atau tanda-tanda yang berterusan menyebabkan tekanan yang kerap dan mengganggu prestasi anda, ia menjadi penyakit mental. Terdapat banyak penyakit mental yang biasa tetapi, kami akan memberi tumpuan kepada salah satu penyakit mental biasa iaitu gangguan obsesif-kompulsif (OCD) dalam projek ini. Gangguan obsesif-kompulsif (OCD) adalah penyakit mental yang menyebabkan pemikiran atau sensasi yang tidak diingini berulang (obsesi) atau keinginan untuk melakukan sesuatu berulang kali (paksaan). Seseorang boleh mempunyai kedua-dua obsesi dan paksaan. Dalam projek ini, saya akan mencadangkan aplikasi mudah alih chatbot yang memberi tumpuan kepada OCD yang percuma untuk dimuat turun dan tidak perlu membeli versi premium untuk mendapatkan semua ciri. Ini disebabkan selepas mengkaji tiga sistem yang sedia ada, kebanyakan sistem adalah untuk penyakit mental umum dan perlu membeli untuk versi premium untuk menggunakan ciri-ciri mereka. Ciri-ciri dalam sistem yang dicadangkan adalah chatbot, ujian diri, maklumat tambahan dan diari. Sistem yang dicadangkan melaksanakan metodologi pembangunan tangkas yang terdiri daripada enam fasa iaitu pengumpulan dan perancangan keperluan, reka bentuk, pembangunan, integrasi dan ujian, penggunaan and semakan. Ujian penerimaan pengguna (UAT) dijalankan untuk mengenal pasti kesilapan atau untuk menentukan sama ada sistem memenuhi keperluan. Kemudian sistem akan dikeluarkan dan maklum balas dikumpulkan daripada pengguna.

ABSTRACT

Many people experience mental health problems. However, if persistent symptoms or signs cause frequent stress and interfere with your performance, it becomes a mental illness. There are many common mental illnesses but, in this project, we will focus on one of the common mental illnesses which is obsessive-compulsive disorder (OCD). Obsessive-compulsive disorder (OCD) is a mental illness that causes repeated unwanted thoughts or sensations (obsessions) or the urge to do something over and over again (compulsions). Some people can have both the obsession and compulsions. In this project, I will propose a chatbot mobile application that focus on OCD that is free to download and no need to purchase for premium version to get all the features. This is because after studying the three existing system, most of the system are for general mental illness and need to purchase for the premium version to use their features. The features in the proposed system are chatbot, self-test, extra information and diary. The proposed system implements agile development methodology that consist of six phases which are requirement gathering and planning, design, development, integration and testing, deployment and review phases. The user acceptance testing (UAT) is conducted in order to identify the error or to determine whether the system meets the requirements. Then the system will be released and feedback are collected from the user.

TABLE OF CONTENT

DECLARATION	
TITLE PAGE	
ACKNOWLEDGEMENTS	ii
ABSTRAK	iii
ABSTRACT	iv
TABLE OF CONTENT	v
LIST OF FIGURE	ix
LIST OF TABLES	xii
LIST OF SYMBOLS	xiii
LIST OF ABBREVIATIONS	xiv
CHAPTER 1 INTRODUCTION	1
1.1 INTRODUCTION	1
1.2 PROBLEM STATEMENT	2
1.3 OBJECTIVE	3
1.4 SCOPE	3
1.5 SIGNIFICANCE OF PROJECT	3
1.6 REPORT ORGANIZATION	3
CHAPTER 2 LITERATURE REVIEW	5
2.1 INTRODUCTION	5
2.2 COMPARISON OF EXISTING SYSTEM	5
2.2.1 Inner Hour	5

2.2.2	Wysa	13
2.2.3	Nuna	18
2.3	SUMMARY OF COMPARISON	23
CHAPTER 3 METHODOLOGY		28
3.1	INTRODUCTION	28
3.2	PROJECT MANAGEMENT FRAMEWORK	28
3.2.1	Requirement Gathering and Planning Phase	29
3.2.2	Design Phase	30
3.2.3	Development Phase	30
3.2.4	Integration and Testing Phase	30
3.2.5	Deployment Phase	31
3.2.6	Review Phase	31
3.2.7	Software Requirement Specification (SRS)	31
3.2.8	Software Design Description (SDD)	31
3.3	PROJECT REQUIREMENT	32
3.3.1	Functional Requirement	32
3.3.2	Non-functional Requirement	33
3.3.3	Constraints	34
3.3.4	Limitations	34
3.4	PROPOSED DESIGN	35
3.4.1	Context Diagram	35
3.4.2	Use Case Diagram and Description	36
3.4.3	Activity Diagram	38
3.5	DATA DESIGN	39
3.5.1	Entity Relationship Diagram (ERD)	39

3.5.2	Data Dictionary	40
3.6	STORYBOARD	44
3.7	DIALOGFLOW	46
3.8	TESTING PLAN	46
3.9	POTENTIAL USE OF THE PROPOSED SOLUTION	50
CHAPTER 4 RESULTS AND DISCUSSION		51
4.1	INTRODUCTION	51
4.2	DEVELOPMENT ENVIRONMENT	51
4.2.1	SYSTEM FUNCTIONALITY	51
4.2.1.2	Manage Registration	52
4.2.1.3	Manage Profile	60
4.2.1.4	Manage Diary	72
4.2.1.5	Manage Information	78
4.2.1.6	Manage Chatting	85
4.2.1.7	Manage Self-Test	86
4.3	IMPLEMENTATION OF DIALOGFLOW	90
4.4	TESTING AND RESULT DISCUSSION	91
4.4.1	System Functionality	92
CHAPTER 5 CONCLUSION		93
5.1	INTRODUCTION	93
5.2	LIMITATION	93
5.3	FUTURE WORK	94
REFERENCES		95

APPENDIX A SOFTWARE REQUIREMENT SPECIFICATION (SRS)	96
APPENDIX B SOFTWARE DESIGN DESCRIPTION (SDD)	97
APPENDIX C GANTT CHART	98
APPENDIX D TEST REPORT	99

LIST OF FIGURE

Figure 2. 1 InnerHour Homepage Interface– Self-Assessment	7
Figure 2. 2 InnerHour Homepage Interface– Additional Courses, Recommended Activities	8
Figure 2. 3 InnerHour Homepage Interface– Mood Tracker, InnerHour Resources	9
Figure 2. 4 InnerHour Goals Interface	10
Figure 2. 5 InnerHour Chatbot Interface	11
Figure 2. 6 InnerHour Premium Interface	12
Figure 2. 7 InnerHour Profile Interface	13
Figure 2. 8 Wysa Homepage Interface	14
Figure 2. 9 Wysa Chatbot Interface	15
Figure 2. 10 Wysa Journey Interface	16
Figure 2. 11 Wysa Self-Care Interface	17
Figure 2. 12 Wysa Therapist Interface	18
Figure 2. 13 Nuna Homepage Interface	19
Figure 2. 14 Nuna Chatbot Interface	20
Figure 2. 15 Nuna Gratitude Journal Interface	21
Figure 2. 16 Nuna Insights Interface	22
Figure 2. 17 Nuna Toolbox Interface	23
Figure 3. 1 Agile Development Model(Kate Ta, 2020)	29
Figure 3. 2 Context diagram for Kiki Chatbot Mobile Application	36
Figure 3. 3 Use Case Diagram for Kiki Chatbot Mobile Application	37
Figure 3. 4 Activity Diagram for Kiki Chatbot Mobile Application	39
Figure 3. 5 Entity Relationship Diagram (ERD) for Kiki Chatbot Mobile Application	40
Figure 4. 1 Main Interface	52
Figure 4. 2 Login Interface	53
Figure 4. 3 Alert Error on Login Interface	54
Figure 4. 4 Pop Up Notification for Invalid Email or Password in Login Interface	54
Figure 4. 5 Reset Password Interface	55
Figure 4. 6 Alert in Rest Password Interface	56

Figure 4. 7 Register Interface	57
Figure 4. 8 Alert in Register Interface	58
Figure 4. 9 Dashboard Admin Interface	59
Figure 4. 10 Dashboard User Interface	59
Figure 4. 11 Profile Interface for User (Left) and Admin (Right)	60
Figure 4. 12 Edit Profile Prompt in Profile Interface	61
Figure 4. 13 Alert Error in Edit Profile Prompt	62
Figure 4. 14 User Feedback Interface	63
Figure 4. 15 Alert Error in Feedback Interface When Field is Blank.	64
Figure 4. 16 Alert Error in Feedback Interface When Words Exceed.	65
Figure 4. 17 Feedback List Interface	66
Figure 4. 18 Feedback Details Interface	67
Figure 4. 19 Feedback History Interface	68
Figure 4. 20 Search Interface in Feedback List Interface (Left) and Feedback Done Interface (Right)	69
Figure 4. 21 User List Interface	70
Figure 4. 22 Delete Prompt at User List Interface	71
Figure 4. 23 Search Interface in User List Interface	71
Figure 4. 24 Diary Interface	72
Figure 4. 25 Add Diary Interface	73
Figure 4. 26 Alert message on Null at Add Diary Interface	74
Figure 4. 27 Alert message on exceeding words in Add Diary Interface	75
Figure 4. 28 Details of Diary Interface in Diary Interface	76
Figure 4. 29 Delete prompt at Diary Interface	77
Figure 4. 30 Search in Diary Interface	78
Figure 4. 31 Admin Information Interface	79
Figure 4. 32 Add Information Interface	80
Figure 4. 33 Alert Message on Add Information Interface	81
Figure 4. 34 Details of information at Information Interface for admin	82
Figure 4. 35 Delete prompt at Information Interface for admin	83
Figure 4. 36 Information Interface for user	84
Figure 4. 37 Search Interface at Information Interface for admin(left) and user(right)	85
Figure 4. 38 Chatbot Interface for both the admin and user.	86
Figure 4. 39 Self-Assessment Interface for User and Admin	87
Figure 4. 40 Self-Assessment Test Interface for User and Admin	88

Figure 4. 41 Self-Assessment Result Interface for User and Admin	89
Figure 4. 42 Self-Assessment History Interface for Admin and User	90
Figure 4. 43 Intent Created in Dialogflow	91
Figure 4. 44 Example of Intent Symptoms	91

LIST OF TABLES

Table 2. 1 Summary of Comparison	27
Table 3. 1 Advantages and Disadvantages of Agile Development Model	29
Table 3. 2 Use case description for Kiki Chatbot Mobile Application	38
Table 3. 3 Profile	41
Table 3. 4 Information	41
Table 3. 5 Feedback	42
Table 3. 6 Diary	42
Table 3. 7 History	43
Table 3. 8 UAT form	49

LIST OF SYMBOLS

LIST OF ABBREVIATIONS

SRS	Software Requirement Specification
SDD	Software Design Document
OCD	Obsessive-Compulsive Disorder
UAT	User Acceptance Testing

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Many people experience mental health problems. However, if persistent symptoms or signs cause frequent stress and interfere with your performance, it becomes a mental illness. Mental illness, also called mental health disorders, refers to a wide range of mental health conditions — disorders that affect your mood, thinking and behavior (Mayo Clinic Staff, 2019). Mental illness can be influenced by several factors, such as genetics, environment, daily habit and biology. Examples of common mental illness includes bipolar disorder, persistent depressive disorder, generalized anxiety disorder (GAD), major depressive disorder (MDD), obsessive-compulsive disorder (OCD), post-traumatic stress disorder (PTSD) and so on.

In order for the people with mental illness to help themselves, seeking for professional and medication is the best way will do. There are many mental health services that provided in most of the country. However, many of the people with mental illness refused to go for these services. Some of them feel that it is a shame and embarrass to admit that they have mental illness and some of them might cannot afford the medical cost. So, there is a need to have a test for them to do it individually to test whether the need to seek for professional helps.

Besides self-test, getting information from a right way is also very important for those who want to get information about mental health problem. Many of the blogger or social media will exaggerate many information just to get attention from public. So, it is important to have an application that allow the user to gain correct information and can do a test to determine their OCD risk level.

1.2 PROBLEM STATEMENT

Living in this modern life that fills with technology causes a lot of mental illness among people nowadays. It is undeniable that technology has improved our life, but at the same time, it also brings some negative effect to us. For example, the use of technology has increased the exposure of us to social media. The survey showed, nearly all adults (99 percent) own at least one electronic device including a television (Kross et al., 2013). Furthermore, the research also shows that the percentage of American adults using social media increased from 7 percent to 65 percent between 2005 and 2015. Among young adults ages 18 to 29, the number is even higher — nine in 10 (90 percent) reported using social media in 2015, compared to 12 percent in 2005 (Kross et al., 2013). Mental illness such as stress runs higher for those who engage the usage of the technology device and social media frequently.

Obsessive-compulsive disorder (OCD) is a common mental illness that affects a lot of people today. Obsessive-compulsive disorder (OCD) is a mental illness that causes repeated unwanted thoughts or sensations (obsessions) or the urge to do something over and over again (compulsions). Some people can have both the obsession and compulsions (Smitha Bhandari, 2020). Besides obsessive thoughts and compulsive behaviors, other problems like depression, eating disorder, anxiety or substance abuse may happen with OCD. Stress is one of the factors that will trigger the OCD episodes or make them worse.

There are many existing chatbot application that help people to gain information with their mental illness. The chatbot is able to identify the question of the user input and give response to the user. The existing chatbot application can help people with a various of issue such as depression, anxiety, sleep and few more to count. It also will guide the user through some exercises to make them feel better. However, the vast majority of chatbots are designed to help people with general mental illnesses, and there is no specific OCD chatbot application. Moreover, most of the chatbot application for mental health will require the user to purchase for their membership to continue or extend the features for using the application. Therefore, this chatbot mobile application is specify for OCD and it is developed for free so that everyone can use it.

1.3 OBJECTIVE

The objective of this project are:

- i. To study the current limitation of the existing chatbot application for OCD.
- ii. To develop the OCD chatbot mobile application.
- iii. To test the functionality of the OCD mobile chatbot application.

1.4 SCOPE

The scopes of this project are as the following:

- i. This mobile application is developed for Android-based smartphone with internet connection.
- ii. This mobile application is free to download.

1.5 SIGNIFICANCE OF PROJECT

The significance of the project are as the following:

- i. This chatbot mobile application is develop to get information for OCD mental illness.
- ii. This chatbot mobile application is develop for free to download and use.
- iii. This chatbot mobile application is develop to reduce the chances of run off for the response or reply of chatbot.

1.6 REPORT ORGANIZATION

This thesis consists of five chapters, each of which will present a different aspect of the topic. The first chapter discusses the introduction to the project, which is mental health.

The chapter also consist of problem statement, objective of this project, scope and significance of the project.

Chapter two discuss about the literature review of the chatbot application for mental health. The existing chatbot application for mental health will be discussed and compared in this chapter.

Chapter three discuss about the methodology to do this application. This chapter will discuss the flow, requirement and the design of the system.

Chapter four will discuss about the result and discussion in this project. The developed interface, API used and testing done will be discuss in this chapter.

Chapter five is the conclusion of this project. The limitation and the future work on the project is discussed in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter will review the three existing mobile applications for mental health that had been chosen. In this chapter, the three existing mobile application will be explained, discussed, reviewed and compared. The existing system will be compared in several aspects such as advantages, disadvantages, features, platform and other relevant aspects. The existing mobile application will also be compared to the proposed mobile application.

2.2 COMPARISON OF EXISTING SYSTEM

There are several existing chatbot application for mental health published in the market. Three of the existing mobile applications is selected and will be compared in aspects of advantages, disadvantages, and features. This subchapter will also explain the information of the selected existing chatbot application and the proposed chatbot application.

2.2.1 Inner Hour

Inner Hour is a mobile application that help people to deal with their anxiety and depression that available in both Android and iOS device. It offers a digital therapy experience that you can access from the comfort of your home. The application consists of an AI relief chatbot, Allie, which is a text-based chatbot to give emotional support to the user.

InnerHour can be downloaded at Google Play Store in Android and App Store in Apple as free. This makes the application can be accessed by more people. If the user wants to subscribe for the premium version which offers more extra features, they will need to pay MYR46.99 annually. Some of the extra features are personalize your self-care activities and incorporate mindfulness into your daily routine. It required user to create account or login to access the account.

There are many features included inside InnerHour. The first feature is the chatbot Allie, which you can contact anytime, anywhere and it also can identify any mental health concerns that you might have(Relief Seeker, n.d.). Allie can also offer you some exercise during chat in order for you to overcome the feelings associated with anxiety, stress or depression. It also provides a self-assessment for the user to test so that they know their mental health condition. There are also recommended activity for the user to calm and relax themselves from stress, anxiety or depression. There are also some additional courses that user can follow to make them feel better. There are also a mood tracker features to allow the user to record their positive and also negative feelings. User can also get the additional resources in the application, for example, article, game and quotes bout the mental health. User is also able to set their goal to do the relaxation activities or physical activities every day.

Figure 2.1 present the Innerhour Homepage Interface for Self-Assessment. The user can take the self-assessment after clicking the “BEGIN” button. User can also click on the “Relief Bot” to start to chat with the chatbot.

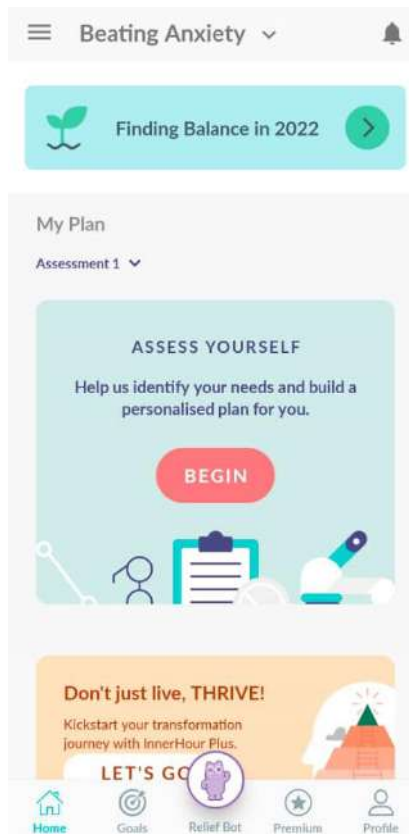


Figure 2. 1 InnerHour Homepage Interface– Self-Assessment

Figure 2.2 present the InnerHour Homepage Interface for Additional Courses and Recommended Activities. The additional courses consist of list activities that can be done to deal with certain mental illness while the recommended activities that can be done individually.

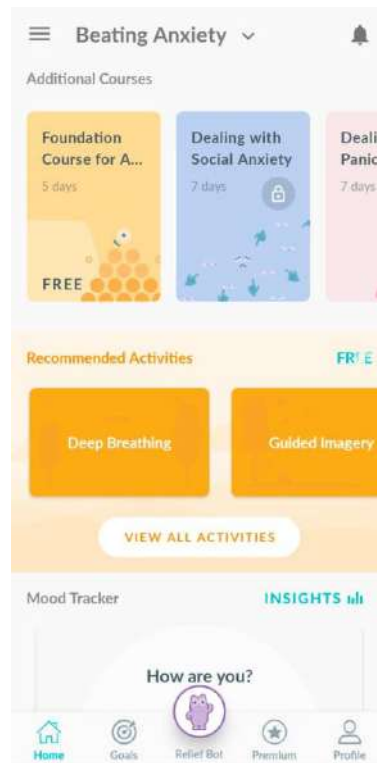


Figure 2. 2 InnerHour Homepage Interface– Additional Courses, Recommended Activities

Figure 2.3 present the InnerHour Homepage Interface for Mood Tracker and InnerHour Resources. User can choose their mood right now at the mood tracker and can click on the Innerhour Resources to view extra information.

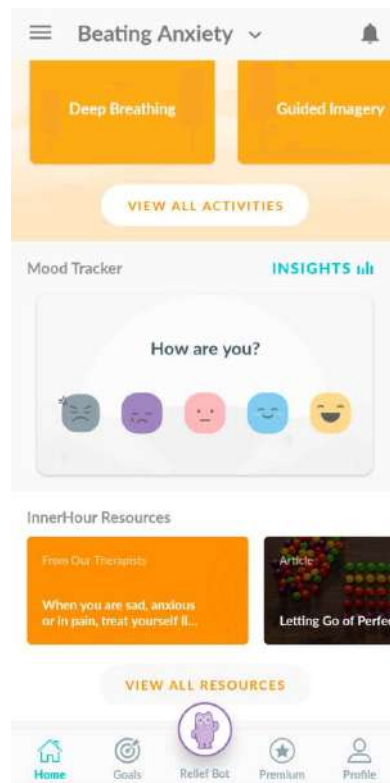


Figure 2. 3 InnerHour Homepage Interface– Mood Tracker, InnerHour Resources

Figure 2.4 present the InnerHour Goals Interface. User can add recommended activities as a goal to do at a particular date.



Figure 2. 4 InnerHour Goals Interface

Figure 2.5 present the InnerHour Chatbot Interface. User can chat with the chatbot Allie through text or choose options that provided at the bottom of the interface.

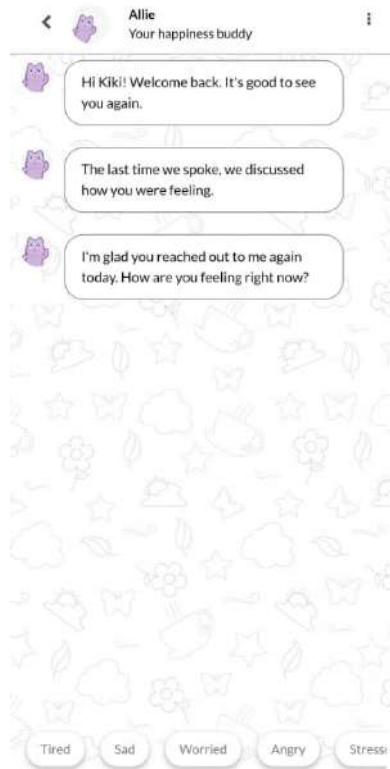


Figure 2. 5 InnerHour Chatbot Interface

Figure 2.6 present the InnerHour Premium Interface. The user can subscribe for premium version at this interface.

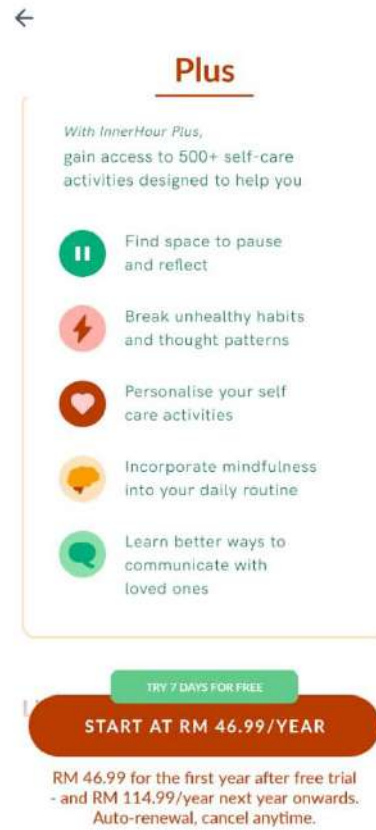


Figure 2. 6 InnerHour Premium Interface

Figure 2.7 present the InnerHour Profile Interface. The profile and summarization of user data will be presented at this interface.

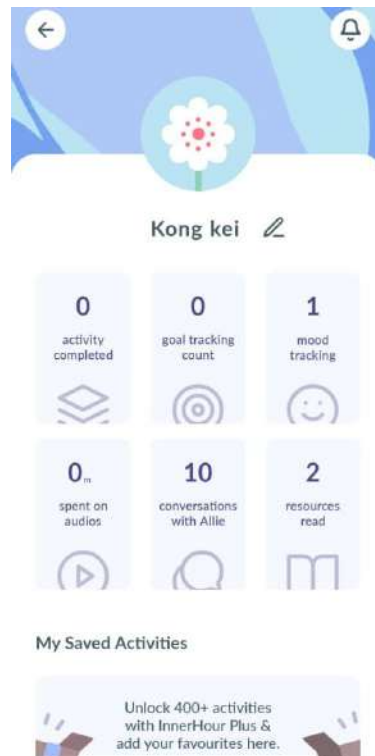


Figure 2. 7 InnerHour Profile Interface

2.2.2 Wysa

Wysa is an AI-based, emotionally intelligent chatbot that developed by Touchkin to help to manage thoughts and emotion via a combination of tools and techniques (Sohail Merchant, 2021). It is a text-based and speech-to-text chatbot application and available in both Android and iOS devices. The application is available in more than 30 countries.

Wysa can be downloaded at Google Play Store in Android and App Store in Apple as free and it is not required to login to access the application. This make the application can be accessed by more people and the data and conversation ore private and encrypted. If they want to find a real-time therapist in Wysa, they will need to pay MYR43.33 per session. That comes to a total of MYR519.99 for 12 sessions. User also need to subscribes premium version to get the full library of the sleep stories, sleep sound and meditations and more therapeutic exercises at the price of MYR144.99/year. Wysa implementing various tools and techniques such as Dialectical Behaviour Therapy (DBT), evidence-based Cognitive Behavioural Therapy (CBT) and guided meditation(Caitlin Bell, n.d.).

There are many features available in Wysa. One of the examples is the chatbot features. It will ask the user how is the feeling every day and it is able to chat with the user. Sometimes it will also reply the user with some interesting GIF. However, when chatting with the chatbot, sometimes the question will appear again and again which will make the user feels annoying. Besides, Wysa also provide the user with many exercises to cope with their mental health. For example, exercise to manage anxiety, beat stress, sleep stories and many others. It also provides a journey feature so that the user can view the chatting history with the chatbot.

Figure 2.8 present the Wysa Homepage Interface. The user can click the button label with “Talk” at the SELF-CARE section or click the “Talk Now” button to start to chat with the chatbot.

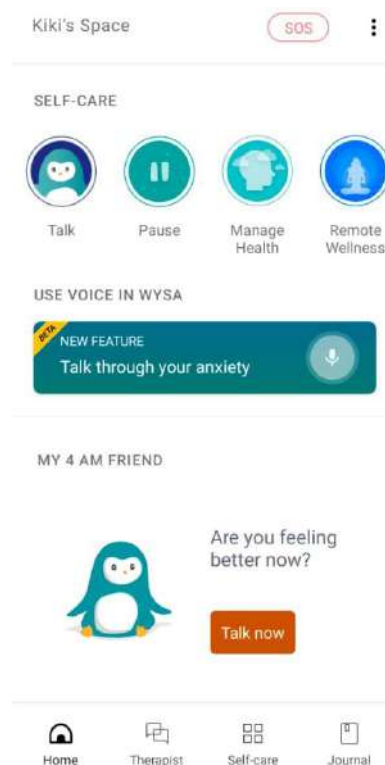


Figure 2. 8 Wysa Homepage Interface

Figure 2.9 present the Wysa Chatbot Interface. The user can chat with the chatbot using text or choose the options at the bottom of the interface. The user can also swipe the smiley face up or down to indicate the mood.



Figure 2. 9 Wysa Chatbot Interface

Figure 2.10 present the Wysa Journey Interface where all the chatting record will showed and can be viewed through this interface.

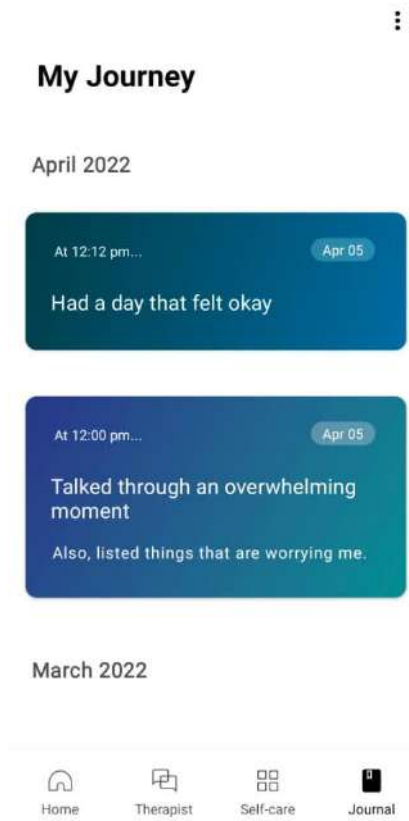


Figure 2. 10 Wysa Journey Interface

Figure 2.11 present the Wysa Self-Care Interface where the user can click into each topic to get the activities to do.



Figure 2. 11 Wysa Self-Care Interface

Figure 2.22 present Wysa Therapist Interface where the user can click “CONTINUE” button to pay to get a therapist.

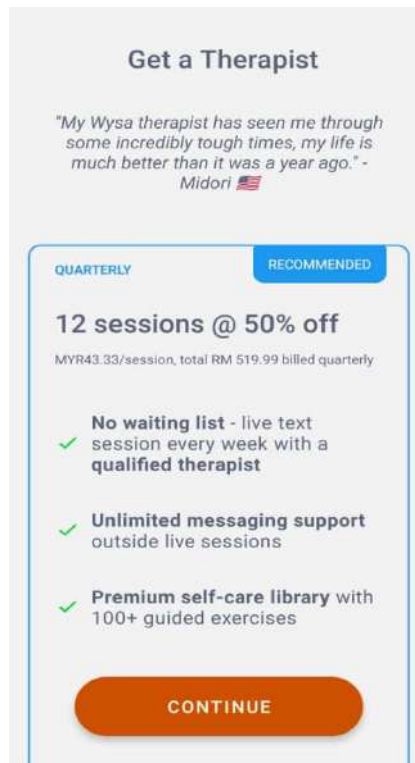


Figure 2. 12 Wysa Therapist Interface

2.2.3 Nuna

Nuna is an AI powered Cognitive Behavioral Therapy (CBT) that using full suite of tools, exercises, journaling, and Nuna's personalized chat(data.ai, n.d.). This application is built by a team of psychologists to create a safe space for early intervention for mental health challenges. Users learn to recognize and overcome stress, anxiety, depression and other mental health problems. It is a text-based chatbot application that available in both Android and iOS device.

Nuna can be downloaded for free from the Google App Store for Android and the App Store for iOS. It required the user to create account or login to access the application. However, if the user wants to subscribe for extra features, they need to subscribe for premium version for MYR 254.99 annually or MYR41.99 monthly. Some of the extra features are gratitude journal, tools or exercise for relaxation and also the chatbot.

There are many features inside the Nuna application but most of it only able to use after subscribe the premium version. The feature that can be used without subscribing the premium version is the Insights features which allow the user to record their feelings, give an overview about their feelings record and feelings chart. The features available for the premium version are the AI chatbot for user to chat with, the gratitude journal to record diary, toolbox which contain the organized exercise for different category, for example sleep, anxiety, stress, motivation and others.

Figure 2.13 present the Nuna Homepage Interface where the user can “Explore” to get more tools, click on the icon to record daily mood and also check in when clicking the “Quick Check-in” button.

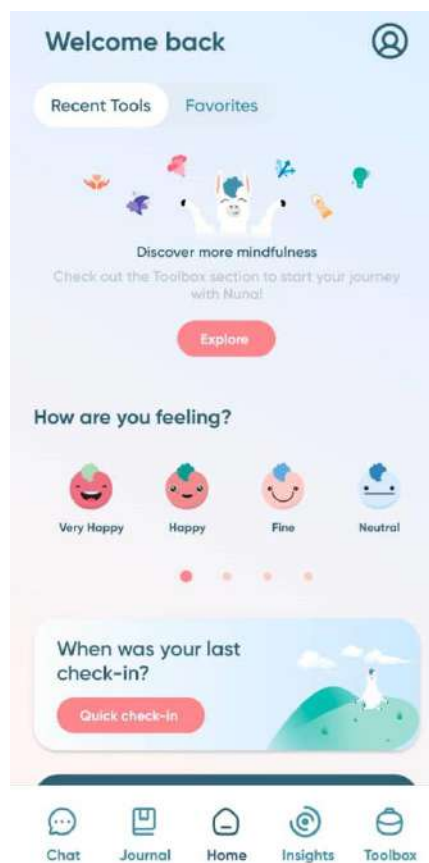


Figure 2. 13 Nuna Homepage Interface

Figure 2.14 present the Nuna Chatbot Interface where the user can chat with the chatbot by writing the message.

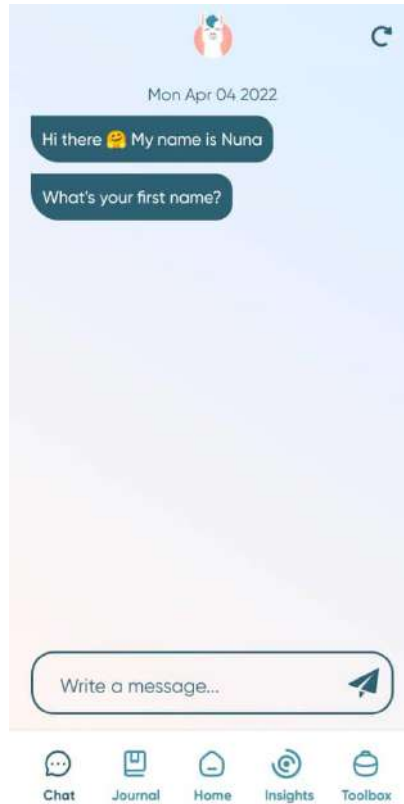


Figure 2. 14 Nuna Chatbot Interface

Figure 2.15 present Nuna Gratitude Journal Interface where the user can view the journal when subscribing to the premium version.



Figure 2. 15 Nuna Gratitude Journal Interface

Figure 2.16 present the Nuna Insights Interface where the user can check in for their daily mood and view the history of the check in.

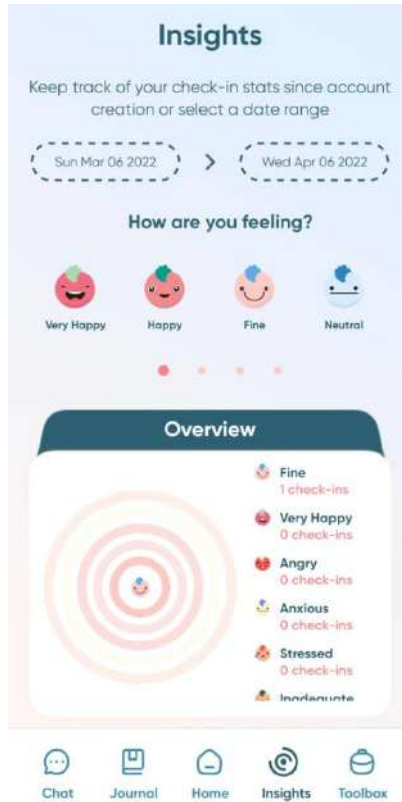


Figure 2. 16 Nuna Insights Interface

Figure 2.17 present Nuna Toolbox Interface where the user can view the activities in different category after subscribing the premium version.



Figure 2. 17 Nuna Toolbox Interface

2.3 SUMMARY OF COMPARISON

Specification	Inner Hour	Wysa	Nuna	Proposed System
Available On	<ul style="list-style-type: none"> • Google Play Store in Android and App Store in iOS. 	<ul style="list-style-type: none"> • Google Play Store in Android and App Store in iOS. 	<ul style="list-style-type: none"> • Google Play Store in Android and App Store in iOS. 	<ul style="list-style-type: none"> • Google Play Store in Android
Features	<ul style="list-style-type: none"> • Chatbot to chat and 	<ul style="list-style-type: none"> • Chatbot to chat with user. 	<ul style="list-style-type: none"> • Chatbot to chat with 	<ul style="list-style-type: none"> • Chatbot to chat

	<p>provide relaxation exercise to user.</p> <ul style="list-style-type: none"> • Recommended exercise to relax. • Mood tracker to record daily mood. • Set goals for relaxation exercise or physical exercise. • Self-assessment. • Additional course to follow every day. • Extra Resources about 	<ul style="list-style-type: none"> • Exercise for mental health which is being categorized. • Real-time coach (Premium version) • Journal that records the history chat with chatbot. 	<p>user. (Premium version)</p> <ul style="list-style-type: none"> • Tools or exercise for mental health which is being categorized. (Premium version) • Gratitude journal to record daily dairy. (Premium version) • Insights to record daily feelings, view overview of daily feelings 	<p>with user.</p> <ul style="list-style-type: none"> • Diary section to record diary. • Extra information regarding to the OCD. • Self-assessment for OCD.
--	--	--	--	---

	mental health.		and feelings chart.	
Advantages	<ul style="list-style-type: none"> • Free to download. • User friendly and clear interface. • Many features available in the application. • The user able to do the exercise according to the course or plan. 	<ul style="list-style-type: none"> • Free to download. • User friendly and clear interface. • Many features available. • No need to login to use the application. • Use a combination of various tools and techniques. • Interesting communication and interaction with chatbot (GIF and emoji). 	<ul style="list-style-type: none"> • Free to download. • User friendly and clear interface. • User able to record their daily feelings and view the overview within a specific duration. 	<ul style="list-style-type: none"> • Free to download. • User friendly and clear interface. • User able to record their daily feelings through diary. • Application specific for OCD.

		<ul style="list-style-type: none"> • Exercise is categorized and specified for different mental health. 		
Disadvantages	<ul style="list-style-type: none"> • Need to purchase for premium version to get more features. • Need to create account and login to access the application, the personal data might not be encrypted and protected. 	<ul style="list-style-type: none"> • Need to subscribe to premium version for real-time coach. • Need to subscribe for premium version to get more exercise library. • The same question will repeat again and again when chatting with the chatbot. 	<ul style="list-style-type: none"> • Need to purchase for premium version to get most of the features. • Need to create account and login to access the application, the personal data might not be encrypted and 	<ul style="list-style-type: none"> • Not available on iOS.

			<p>protected.</p> <ul style="list-style-type: none"> • Only the Insights features can be accessed if not subscribe for the premium version. 	
--	--	--	--	--

Table 2. 1 Summary of Comparison

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

The project management framework and the methodology that is used to develop this chatbot application for OCD will be discussed in this chapter. There are a few types of methodology to develop model in software engineering but only one methodology will be chosen for this project development. Besides, this chapter will also discuss about the project requirements, proposed design and database design of this project. Agile methodology is chosen to develop this project and further explanation is discussed in this chapter.

3.2 PROJECT MANAGEMENT FRAMEWORK

There are several methodologies can be used to develop a software project. Some of the examples are waterfall development methodology, agile development methodology, DevOps deployment methodology and rapid application development. Each of the methodology has their advantages and disadvantages. The methodology that is used in this project is the agile development methodology.

Agile methodology is an iterative and incremental approach that are anticipates the need for flexibility and applies a level of pragmatism to the delivery of the finished product(Kate Brush, 2019). It is said to be flexible due to its iterative an incremental approach which enable the implementation of changes to be done more easily. Figure 3.1 shows the process of an agile development model and the advantages and disadvantages of the agile development model is show as Table 3.1.

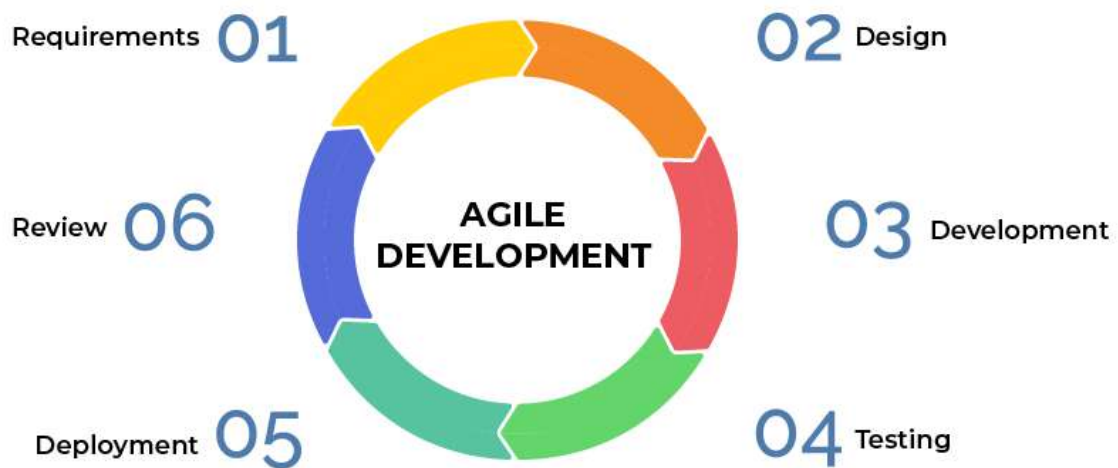


Figure 3. 1 Agile Development Model(Kate Ta, 2020)

Advantages	Disadvantages
System can be deployed faster.	Progress is difficult to measure.
Adapt to change and respond faster.	Documentation tends to get side-tracked.
Issue can be detected and fixed faster.	Actual cost is unknown.
Can get immediate feedback.	Product will lack of cohesion.

Table 3. 1 Advantages and Disadvantages of Agile Development Model

3.2.1 Requirement Gathering and Planning Phase

Requirement gathering is the first phase of the Software Development Life cycle (SDLC) for the agile development model. At this phase, the requirements should be gathered and defined. In this project, the requirement is done by collecting requirement from the existing system. The Software Requirement Specification (SRS) is prepared during this stage. The SRS document

include the details such as functionality and the interface design of this system. The details of the requirement documented in the Software Requirement Specification (SRS). Besides that, we need to collect information about the obsessive-compulsive disorder (OCD) such as causes, symptoms and the effect of having this mental illness to have a better understanding to the OCD.

3.2.2 Design Phase

Design phase is after requirement gathering and planning phase. In this phase, the basic function, programming language, syntax libraries and basic framework will be discussed and identified. Java is used to program this project and the real time database that used in this project is Firebase. All the related diagrams such as data flow diagram, use case diagram, context diagram and others such as interface design will be designed and presented in Software Requirement Specification (SRS) document. Further detail architecture design will be discussed in Software Design Description (SDD) document.

3.2.3 Development Phase

Development phase is the conversion of the structures and flows based on the requirements in Software Requirement Specification (SRS) and Software Design Description into coding. The interfaces and functions of the Kiki Chatbot Mobile Application that had been plan and design in previous phases will be implement in this phase.

3.2.4 Integration and Testing Phase

This phase is done after the development phase. In this phase, the product will be integrated and tested to ensure the final product is bug-free and it meet all the documented requirement of the system. A series of tests is conducted to check bugs, ensure the flow of the system is working as stated in the document and meets the requirements. The testing that conducted for the Kiki Chatbot Mobile Application is User Acceptance Testing (UAT). In Chapter 4, we will discuss in more details about the testing.

3.2.5 Deployment Phase

The Kiki Chatbot Mobile Application is then to be deployed after the integration and testing phase. The application is deployed on the server and made available to users and admins. The application will keep updating by adding new features or fixing bugs.

3.2.6 Review Phase

After the product being deployed, the reviews and feedbacks from the customer and stakeholders will be recorded and the development team will present to the customer the result. The developer of the Kiki Chatbot Mobile Application will present the idea to correct the issues that occurred during previous phases. All the reviews and feedbacks will be put into the consideration in order to facilitate the next iteration.

3.2.7 Software Requirement Specification (SRS)

Software Requirement Specification (SRS) is a document that describe the behavior and features of a system or application. It describes what a system can do and what is the expected performance that the system will provide. It is important as it form the basis of the entire project.

(Refer Appendix A)

3.2.8 Software Design Description (SDD)

Software Design Description (SDD) is a document that describe the design and the architecture of a system or application. It provides an overview to the development process on what or how to develop. Through this document, we can go through every challenge that we might face.

(Refer Appendix B)

3.3 PROJECT REQUIREMENT

Before the starting of the design and development phase, the project requirement should be gathered and define beforehand such as functional requirement and non-functional requirements. In this section, we will discuss about the functional, non-functional, constraints and limitations of this project.

3.3.1 Functional Requirement

User

- The system shall allow the user to register a new account using existing email.
- The system shall allow the user to login using the registered account.
- The system shall allow the user to logout to the system.
- The system shall allow the user to chat with chatbot using text.
- The chatbot shall be able to answer the user if the question is scripted.
- The system shall allow the user to do a self-assessment about the OCD and calculate the severity.
- The system shall allow the user to view their self-assessment history.
- The system shall provide user link to some extra information that is related to OCD.
- The system shall allow the user to record their emotion everyday through the diary.
- The system shall allow the user to view, edit and their account.
- The system shall allow the user to make feedback.

Admin

- The system shall allow the admin to login to the system using registered account.
- The system shall allow the admin to logout of the system.
- The system shall allow the admin to view and delete user.
- The system shall allow the admin to view, add, delete or edit the extra information.

- The system shall allow the admin to view and delete the feedback.
- The system shall allow the admin to chat with chatbot using text.
- The chatbot shall be able to answer the admin if the question is scripted.
- The system shall allow the admin to do a self-assessment about the OCD and calculate the severity.
- The system shall allow the admin to view their self-assessment history.
- The system shall provide admin link to some extra information that is related to OCD.

3.3.2 Non-functional Requirement

i. Availability

The system shall operate for 24 hours every day including overnight.

ii. Performance

The time for each of the interface to load shall not more than 6 seconds. The time for the chatbot to reply the user should not more than 5 seconds.

iii. Usability

The operation of the system shall be easy and clear for the user to operate.

iv. Reliability

The system shall be able to function well and reduce the critical failure time under normal usage.

v. Security

The system must implement security best practice to secure the user information.

3.3.3 Constraints

i. Time Constraint

In order to make a project to success, time management is essential and there are several of time constraints that will be faced during different phase of the project. Staying with the project timeline is very important to the outcome. If we fail to stay on the schedule, it could result in the failure on the whole project.

ii. Scope Constraint

Scope constraint in a project refers as the project's magnitude such as the quality, detail and deliverables. The project scope needs to be aware during each phase. If the scope increases, more time and cost will require to complete the project.

iii. Resources Constraint

Since the mobile application is designed for OCD, which is a mental health illness, so the resources must find from a valid and official source.

3.3.4 Limitations

i. Device limitation

The limitation of this system is that it is only applicable to mobile services and not web services since it is a mobile application. This means the application only can use and access through mobile with iOS and Android OS and not others gadget such as laptop, PC and television.

ii. Chatbot limitation

The limitation of the chatbot in this system is that the chatbot is only able to answer or respond to the question/reply from the user if the question/reply is scripted in the chatbot. If the question/reply is out of the script, the chatbot will confuse or undergo looping of the same reply.

iii. Emotions limitation

Chatbots have no emotions so they cannot really relate to the situation and emotions of someone. This means that the chatbot will not be able to establish a connection with the user, which is crucial especially as a mental health application.

3.4 PROPOSED DESIGN

3.4.1 Context Diagram

Figure 3.2 shows the context diagram of the application. Context diagrams show the interaction between a system and other actors that the system is designed to interface with, helping to understand the context to which the system belongs. The context diagram consists of two entities which are user and admin. First, in order to access the application and retrieve the data from the application, both the admin and user need to login to the application by entering a correct email and password. The user can edit profile, submit feedback, chat, do self-test, and write a diary. All these data will be sent and saved in the system. Then, the system will pass the updated profile, diary history, self-test result, self-test history, information, and chatting response to the user. The admin can edit profile, edit information, do self-test, view feedback, chat, and these data will be saved to the system. The system will return edited data to the admin.

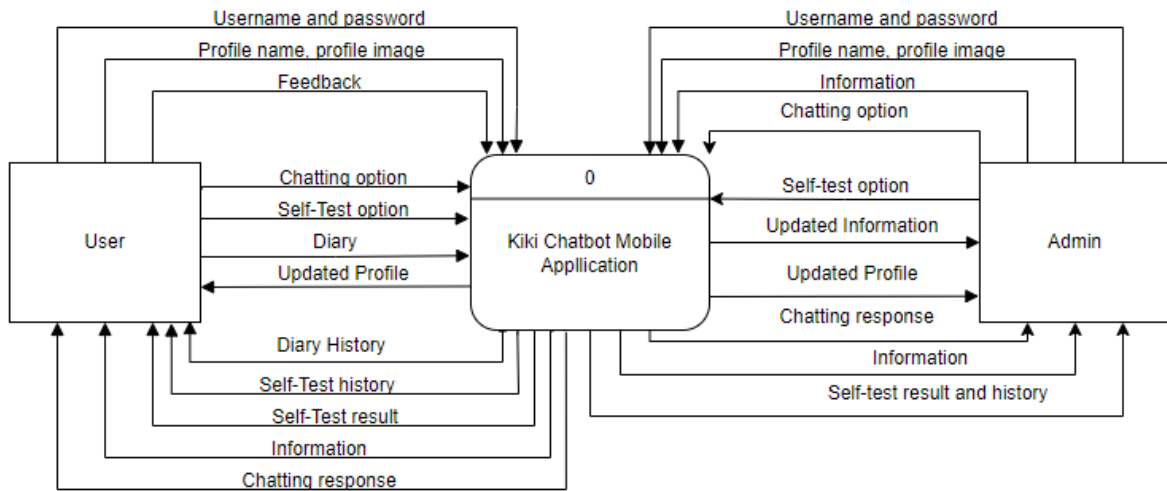


Figure 3. 2 Context diagram for Kiki Chatbot Mobile Application

3.4.2 Use Case Diagram and Description

A use case diagram shows all the interaction between the actor with the application and it is represented graphically. Figure 3.3 below show the use case diagram of the Kiki Chatbot Mobile Application. There are two actors in this application which are the user and the admin. Besides, six modules are included in this application which are manage registration, manage chatting, manage self-test, manage information, manage diary and manage profile.

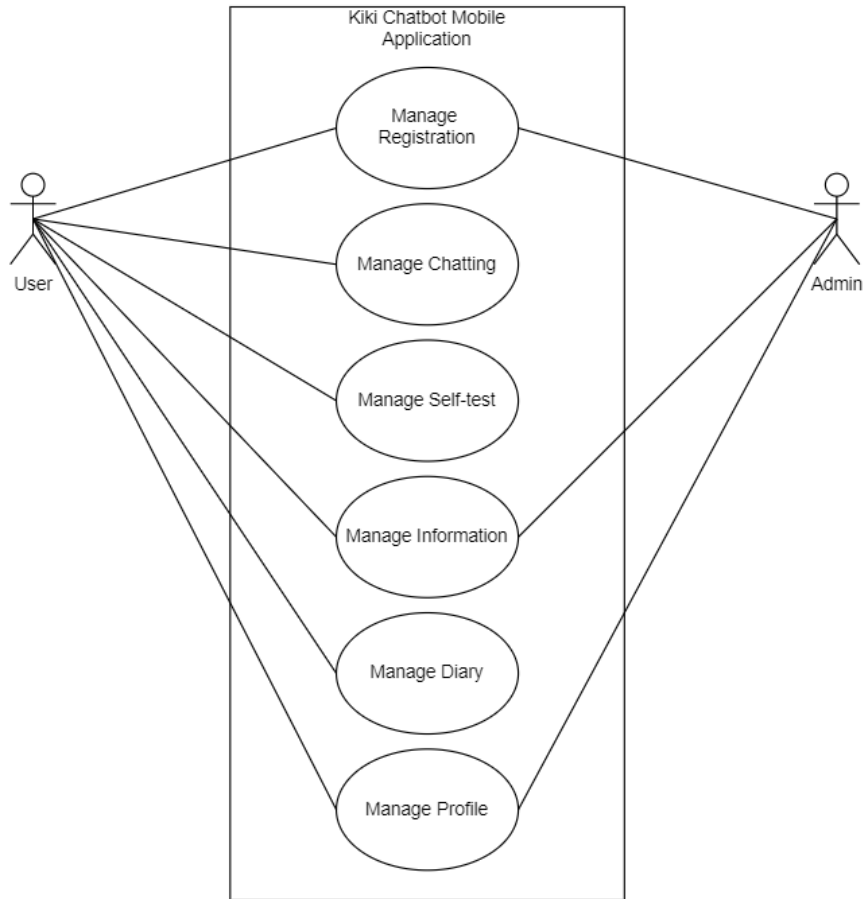


Figure 3. 3 Use Case Diagram for Kiki Chatbot Mobile Application

Use Case	Description
Manage Registration	The use case allows the user to register for an account in the application and login to the application.
Manage Chatting	The use case allows the user to chat using text through chatbot Kiki.
Manage Self-test	The use case allows the user to take a self-assessment to test their severity on OCD.

Manage Information	The use case allows the admin to add, edit and delete website information about OCD in the system.
Manage Diary	The use case allows the user to view, add or delete diary in the system.
Manage Profile	The use case allows the user and admin view and edit their profile, delete their account from the application. The use case allows the user to make feedback in the application. The use case allows the admin to view feedback made by user in the application. The system also allows the admin to delete a user account.

Table 3. 2 Use case description for Kiki Chatbot Mobile Application

3.4.3 Activity Diagram

The activity diagram describes the flow of the application from the starting to the end. Figure 3.3 show the overall activity diagram for the Kiki Chatbot Mobile Application.

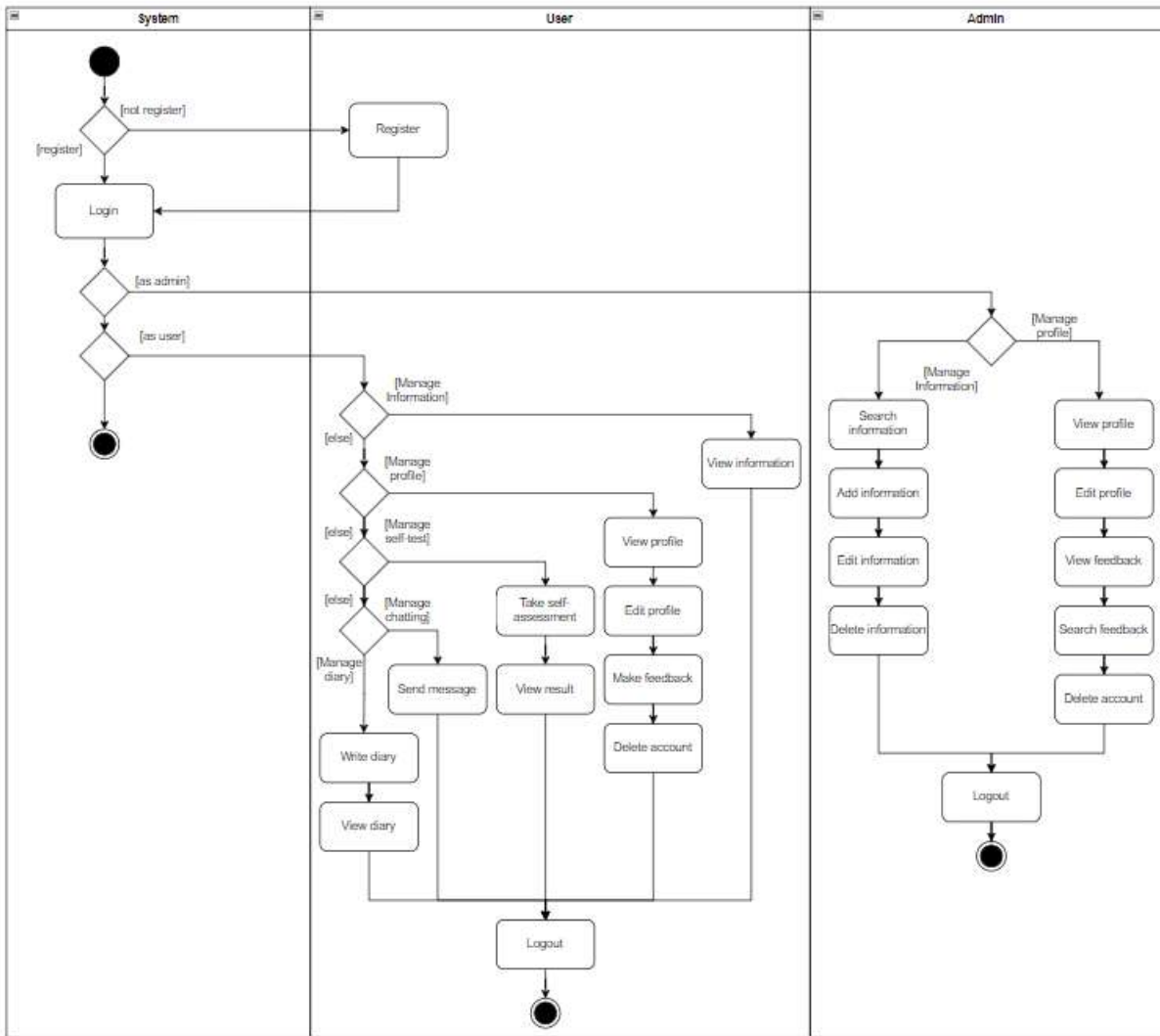


Figure 3. 4 Activity Diagram for Kiki Chatbot Mobile Application

3.5 DATA DESIGN

3.5.1 Entity Relationship Diagram (ERD)

Entity-Relationship Diagram (ERD) is a flow chart showing how entities in a system relate to each other in an application. Figure 3.5 shows the ERD for Kiki Chatbot Mobile Application. There are 5 entities in this system which are Profile, Information, Feedback, Diary and History.

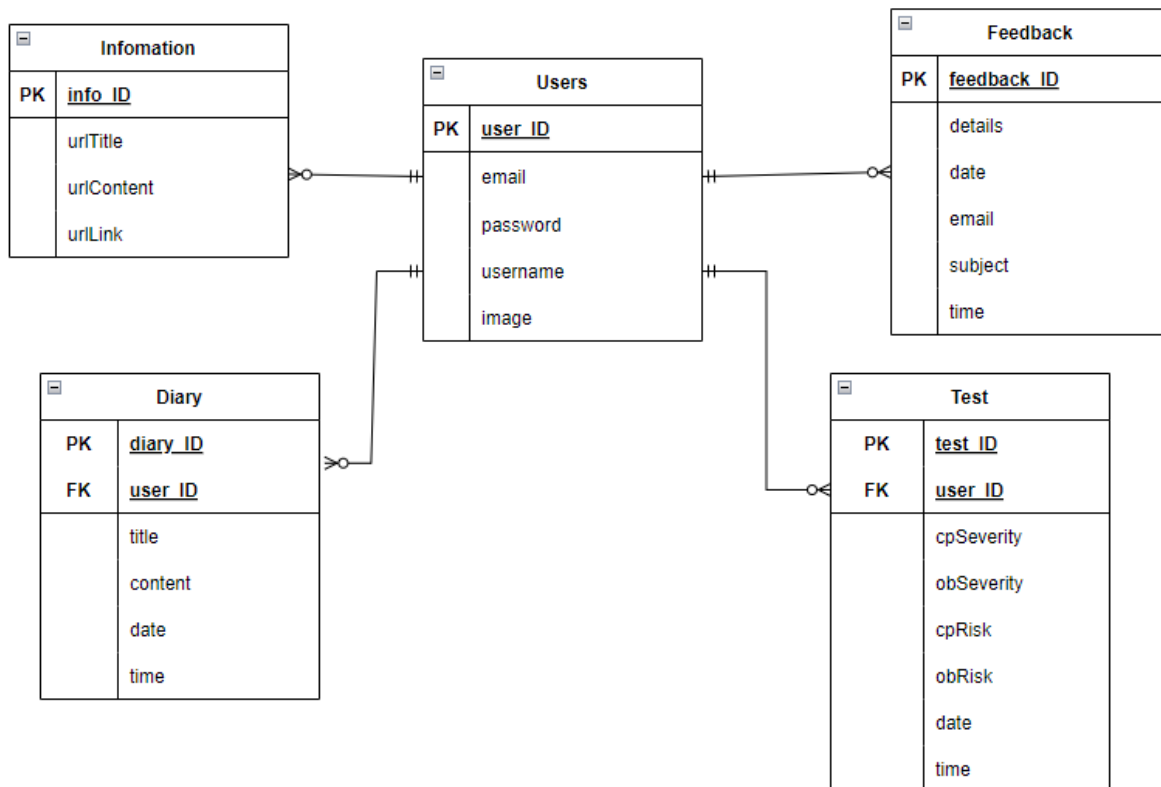


Figure 3. 5 Entity Relationship Diagram (ERD) for Kiki Chatbot Mobile Application

3.5.2 Data Dictionary

Users

Field Name	Description	Data Type	Constraint
user_ID	User account ID	VARCHAR(255)	PK, NOT NULL
email	User email	VARCHAR(255)	
password	Account password	VARCHAR(255)	

username	Username of user	VARCHAR(255)	
----------	------------------	--------------	--

Table 3. 3 Profile

Information

Field Name	Description	Data Type	Constraint
info_ID	Information ID	VARCHAR(255)	PK, NOT NULL
urlContent	Information title	VARCHAR(255)	
urlLink	URL link for the information	VARCHAR(255)	
urlTitle	Description of the information	VARCHAR(255)	

Table 3. 4 Information

Feedback

Field Name	Description	Data Type	Constraint
feedback_ID	Feedback ID	VARCHAR(255)	PK, NOT NULL
details	Feedback detail	VARCHAR(255)	
date	Date when user send the feedback	DATE	

time	Time when user send the feedback	TIME	
email	Email address that user fill for further contact	VARCHAR(255)	
subject	Feedback subject	VARCHAR(255)	

Table 3. 5 Feedback

Diary

Field Name	Description	Data Type	Constraint
diary_ID	Diary ID	VARCHAR(255)	PK, NOT NULL
user_ID	Account ID	VARCHAR(255)	FK
title	Diary title	VARCHAR(255)	
content	Diary detail	VARCHAR(255)	
date	Date when user save the diary	DATE	
time	Time when user save the diary	TIME	

Table 3. 6 Diary

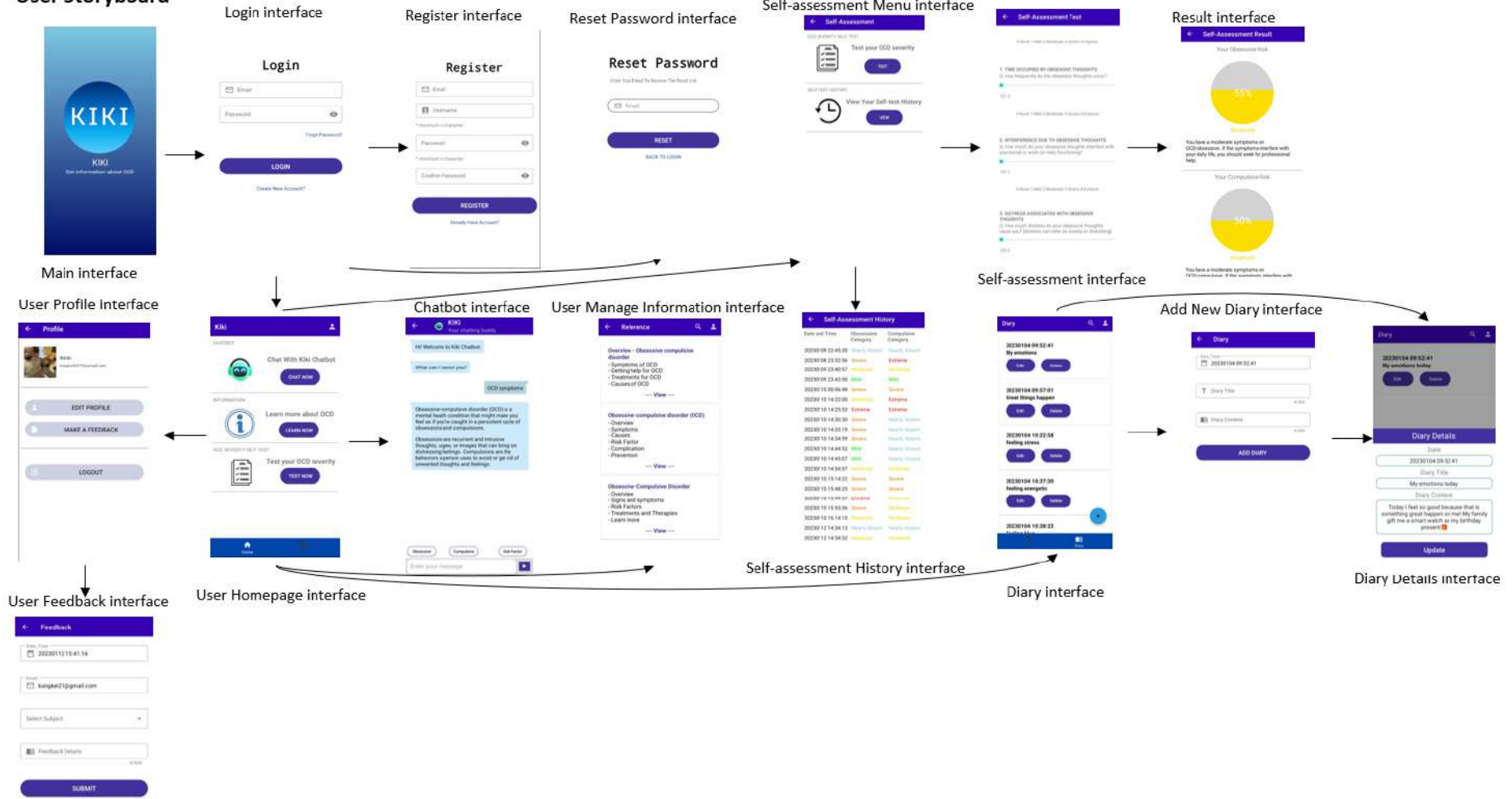
History

Field Name	Description	Data Type	Constraint
test_ID	History ID	VARCHAR(255)	PK, NOT NULL
user_ID	Account ID	VARCHAR(255)	FK
cpSeverity	The total score for compulsive.	FLOAT	
obSeverity	The total score for obsessive.	FLOAT	
cpRisk	The category for the calculated result score for compulsive.	VARCHAR(255)	
obRisk	The category for the calculated result score for obsessive.	VARCHAR(255)	
date	Date when user submit the test	VARCHAR(255)	
time	Time when user submit the test	VARCHAR(255)	

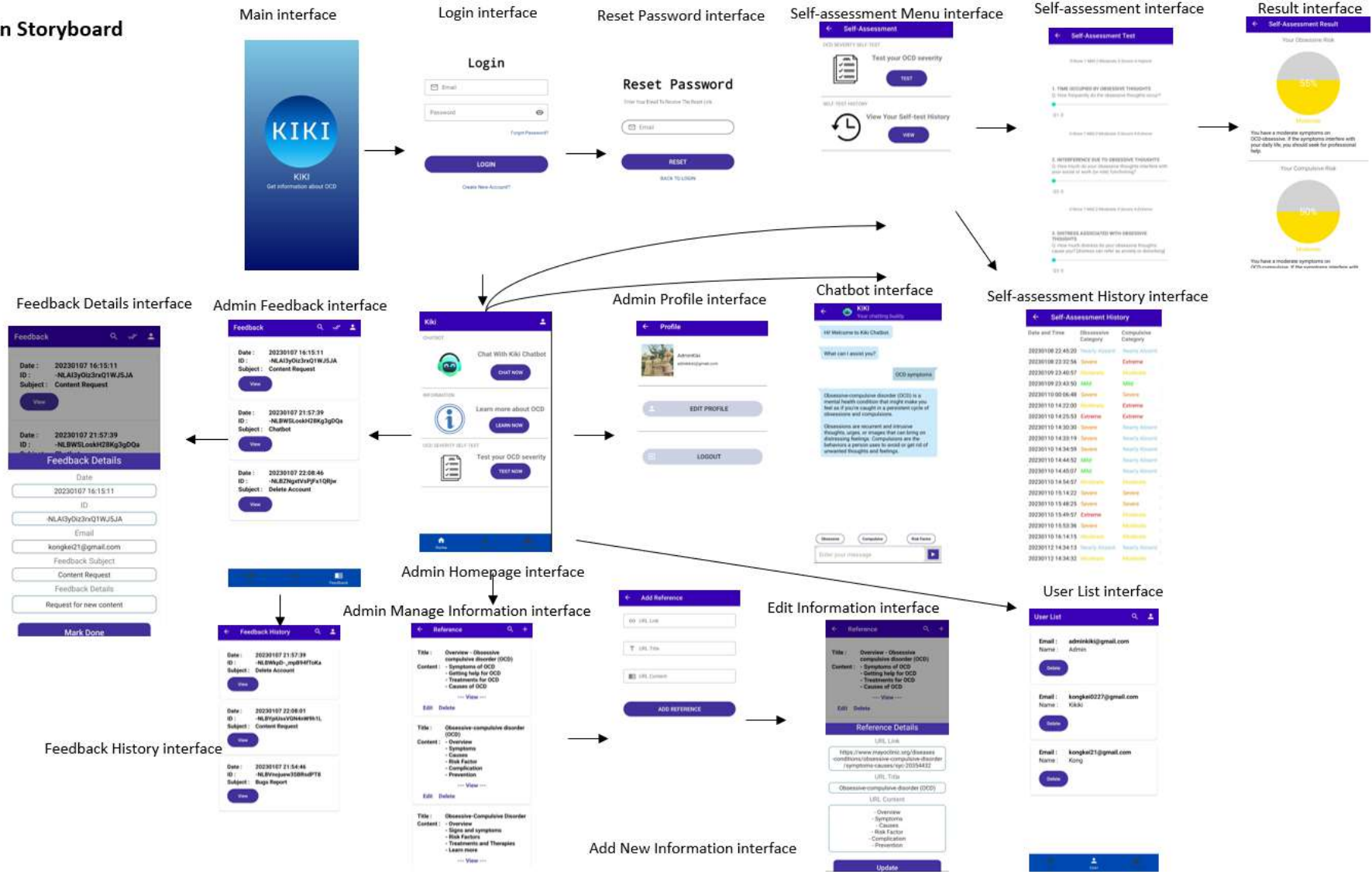
Table 3. 7 History

3.6 STORYBOARD

User Storyboard



Admin Storyboard



3.7 DIALOGFLOW

Dialogflow will be implemented in the design of the chatbot. Dialogflow is a natural language understanding platform that allows the easily design conversational user interfaces and integrate them into mobile apps, web applications, devices, bots, interactive voice response systems, and more. Dialogflow can provide users with new and engaging ways to interact with their products. Dialogflow can analyse different types of input, including text or voice input but in this project, only text input is implemented.

In order for the user to use the chatbot, they must login to the application. When the user press into the chatbot interface, the chatbot will first display a welcome message. Then, they can choose option from the bottom of the chatbot interface to ask for information for can type it manually. After accepting message from the user, Dialogflow will analyse and interpret the question by using the natural language understanding (NLU). If the question is scripted, the chatbot will extract the keyword of the message and give reply to the particular question. The chatbot will ask the user to ask one more time if it cannot understand and interpret the question. The conversation will end when the user clicks “That’s all” from the options.

3.8 TESTING PLAN

After the system is being developed, testing should be conducted to identify the error or to determine whether the system meets the requirements. In this project, User Acceptance Test (UAT) is planned to test the system after the Kiki Chatbot Mobile Application is being deployed. The purpose of conducting this UAT is to ensure the developed and deployed system meets the requirement that is stated in the Software Requirements Specification (SRS). The testing is also to ensure the system is accepted by the end user.

No.	Module	Activities	Status		Comments
1.	Manage Registration	User login	Yes <input type="checkbox"/>	No <input type="checkbox"/>	

2.		User register	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
3.		User reset password	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
4.		Admin login	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
5.		Admin reset password	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
6.	Manage Chatting	User and admin chat using text	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
7.		Chatbot give response	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
8.	Manage Self-test	User and admin do self-assessment	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
9.		User and admin get self-assessment result	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
10.		User and admin view self-assessment history	Yes <input type="checkbox"/>	No <input type="checkbox"/>	

11.	Manage Information	User view information	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
12.		User search information	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
13.		Admin view information	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
14.		Admin search information	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
15.		Admin add information	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
16.		Admin edit information	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
17.		Admin delete information	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
18.	Manage Diary	User add diary	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
19.		User delete diary	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
20.		User view diary details	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
21.	Manage Profile	User view profile	Yes <input type="checkbox"/>	No <input type="checkbox"/>	

22.		User edit profile	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
23.		User make feedback	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
24.		User logout	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
25.		Admin view profile	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
26.		Admin logout	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
27.		Admin view feedback list	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
28.		Admin search feedback	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
29.		Admin mark done feedback	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
30.		Admin view feedback history	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
31.		Admin view user list	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
32.		Admin delete user account	Yes <input type="checkbox"/>	No <input type="checkbox"/>	

Table 3. 8 UAT form

3.9 POTENTIAL USE OF THE PROPOSED SOLUTION

Due to different reasons, people nowadays tend to fill with stress and lead to mental illness. Some of the mental illness might be severe but some might be mild. Obsessive-compulsive disorder (OCD) is one of the common mental illnesses. There is no an existing chatbot application that is specifically designed for OCD. Thus, the Kiki Chatbot Mobile Application can be used by people who think themselves might get OCD, to chat with chatbot to get information, to record their daily dairy, to test their OCD severity, to get more information about OCD. The application is also useful for those are not an OCD patient but want to know more information about OCD.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 INTRODUCTION

This chapter will discuss about the implementation of Graphical User Interface (GUI) and features in this Kiki mobile chatbot application. Besides, the script that design to implement in the chatbot is also included in this chapter. Lastly, the testing that apply to this Kiki Mobile Chatbot Application which is the Unit Testing report is also included in this chapter.

4.2 DEVELOPMENT ENVIRONMENT

Kiki Mobile Chatbot Application is a mobile application that using Android Studio to develop. Android Studio is built on with JetBrains' IntelliJ IDEA software which are specifically used for Android development. It is also the official integrated development environment for Google's Android operating system. In this project, Android Studio is used to develop the Graphical User Interface (GUI) and also the function in the application. For the database, Firebase is used to store the data. Realtime database is used in this Firebase. It uses a NoSQL and also the real-time hosting of databases, social authentication, services, content or notification. In our project, the database in arranged in NoSQL and it is a real-time database that implemented using Firebase.

4.2.1 SYSTEM FUNCTIONALITY

In this section, the system functionality and the interaction of the graphical user interface (GUI) with the users. To ease the learning to use the application, a GUI should ne develop in a clear design so that the user can know how to use the application clearer. The application's interfaces with their respect functions will be discuss in this section.

4.2.1.2 Manage Registration


Figure 4.1 show the main interface when the user opened the Kiki Mobile Chatbot Application. The interface will be automatically directed to the Login Interface that show in Figure 4.2.



Figure 4. 1 Main Interface

Figure 4.2 show the Login Interface. In the login interface, the actor can login to the function by filling correct email and password and click on <<LOGIN>> button. When clicking on the “Forgot Password” text, the interface will redirect to Reset Password Interface while clicking on the “Create New Account?” text, the interface will redirect to register interface.

Login


[Forgot Password?](#)

[LOGIN](#)

[Create New Account?](#)

Figure 4. 2 Login Interface

Figure 4.3 show the alert from the login interface when the user enters an invalid email format or blank input field. When the actor blanks the email field and the password field or enter an invalid email format and click <<LOGIN>>, an alert will display in the respectively field.

Figure 4.4 show the interface with pop up when the actors enter invalid email or password. When the actor enters incorrect email and password to login to the system, a pop up will show to inform the actor failed to login. If the email and password are valid and correct, then the actor will login successfully and redirect to the dashboard.

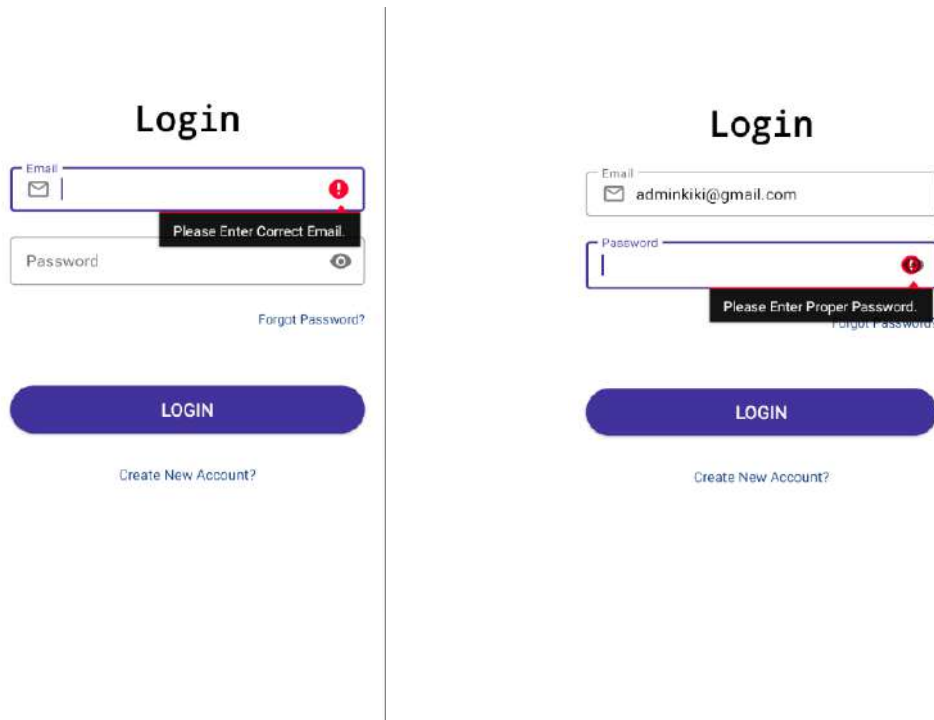


Figure 4. 3 Alert Error on Login Interface

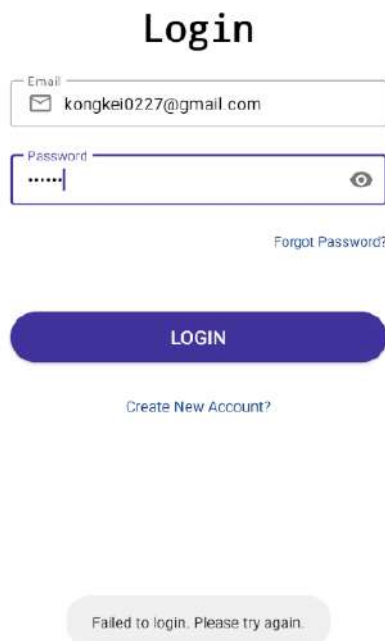


Figure 4. 4 Pop Up Notification for Invalid Email or Password in Login Interface

Figure 4.5 show the Reset Password Interface. The user can fill their registered email and click on <<RESET>> button to get their reset password email to reset their password.



The image shows a web interface for resetting a password. At the top, the title "Reset Password" is displayed in a large, bold, black font. Below the title, a subtitle reads "Enter Your Email To Receive The Reset Link". Underneath this is a text input field with a rounded rectangular border, containing a small envelope icon and the placeholder text "Email". Below the input field is a prominent, rounded rectangular button with a dark blue background and the word "RESET" in white, uppercase letters. At the bottom of the interface, the text "BACK TO LOGIN" is centered in a smaller, blue, uppercase font.

Figure 4. 5 Reset Password Interface

When the actor enters a blank or invalid email format and click on <<RESET>> at the Reset Password Interface, an alert will show beside the field as shown in Figure 4.6. Clicking on the “BACK TO LOGIN” text will redirect the user back to the Login Interface.

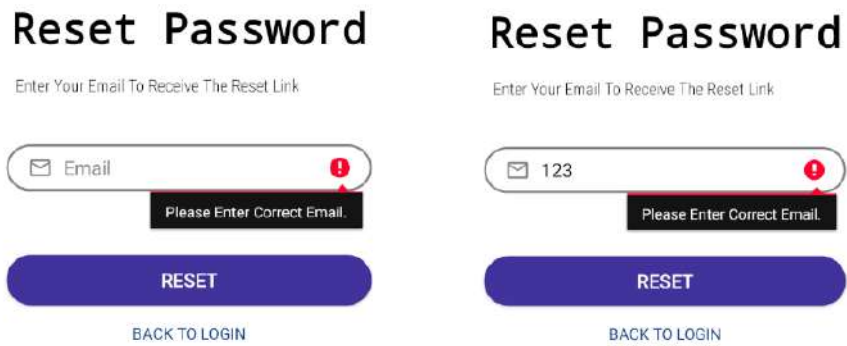


Figure 4. 6 Alert in Rest Password Interface

Figure 4.7 show the Register Interface. The user can register into the system by filling all the field and click on <<REGISTER>> button.

The image shows a registration form titled "Register". It contains four input fields: "Email" with an envelope icon, "Username" with a person icon and a note "* minimum 4 character", "Password" with an eye icon and a note "* minimum 6 character", and "Confirm Password" with an eye icon. Below the fields is a blue "REGISTER" button and a link "Already Have Account?".

Figure 4. 7 Register Interface

Figure 4.8 show the alert message when the actor enters a blank or invalid email format, blank username, blank password and confirm password or password or confirm password is not matching and click on <<REGISTER>>. If all the fields are correct and valid, then the user will be register into the database. Clicking on the “Already Have Account” will redirect the actor to the Login Interface.

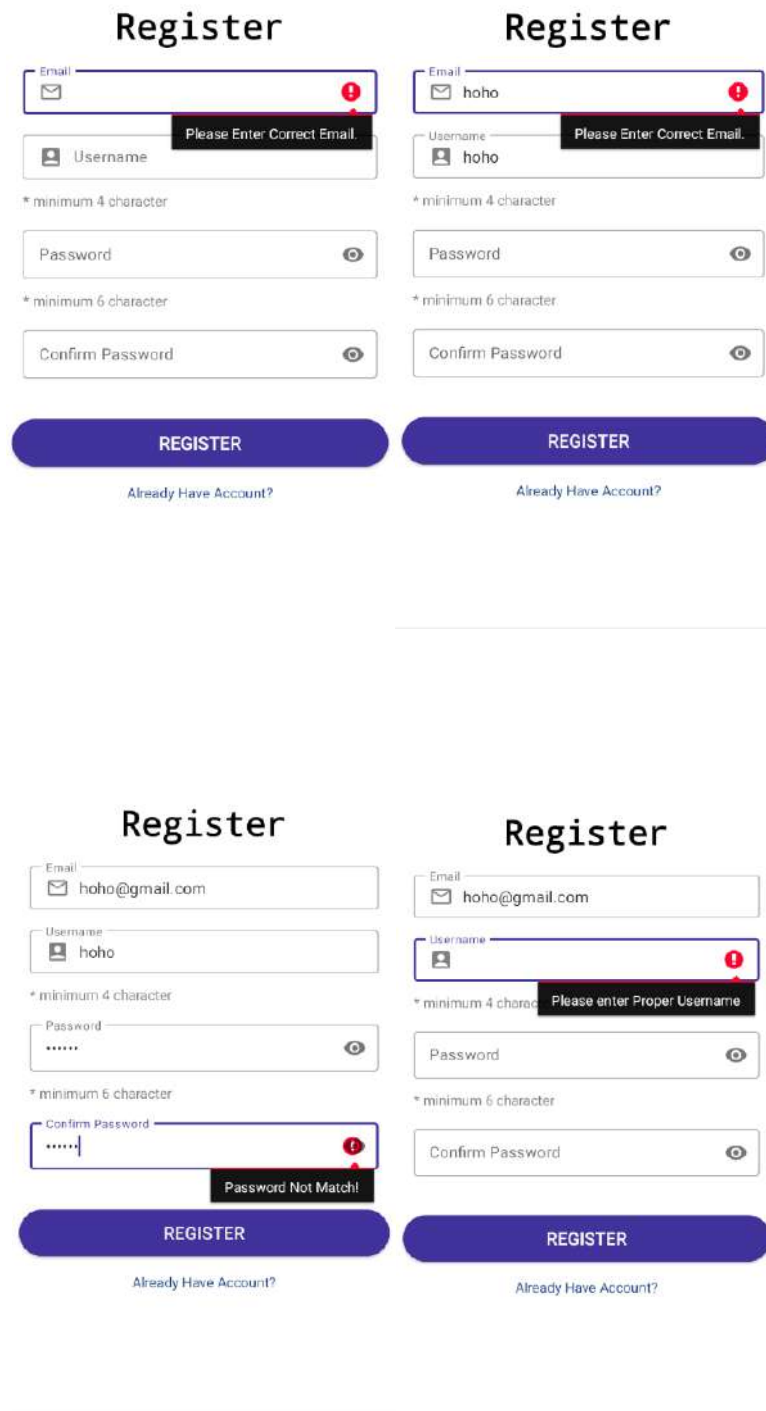


Figure 4. 8 Alert in Register Interface

Figure 4.9 show the dashboard interface when the admin successfully login into the application while Figure 4.10 show the dashboard interface for the user.

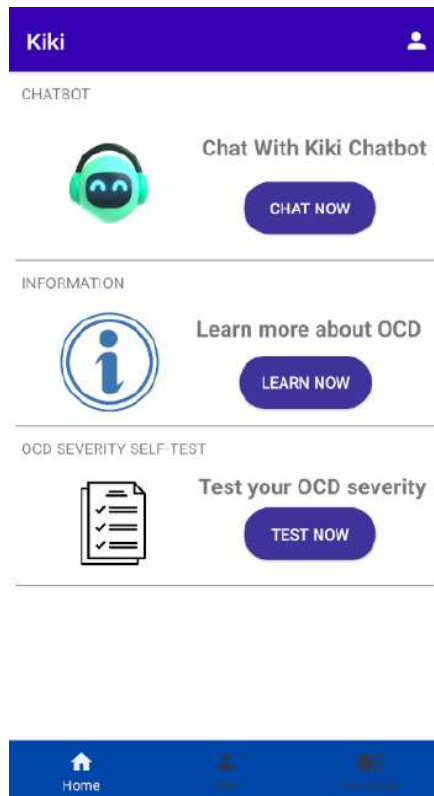


Figure 4. 9 Dashboard Admin Interface

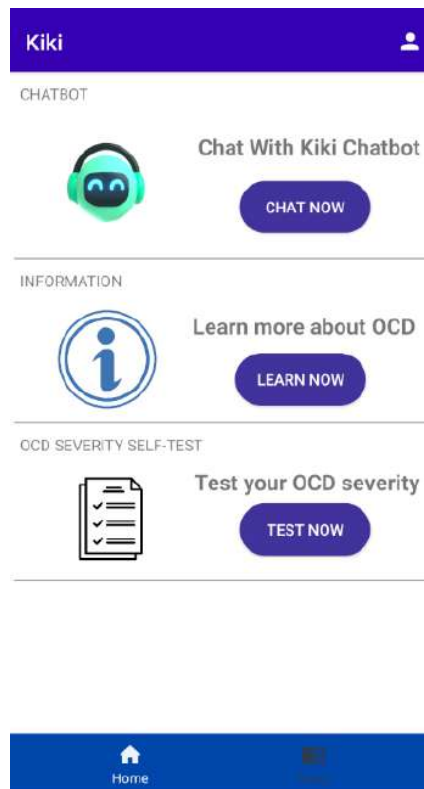


Figure 4. 10 Dashboard User Interface

4.2.1.3 Manage Profile

Figure 4.11 show the profile interface for both the admin and user when they click on the profile icon at the right top menu bar from the Dashboard Interface. When clicking on the <<Logout>> button, the admin and user will logout the system and redirect back to Main Interface. The actor and user also can change or add their profile picture by clicking on the profile image on the top left interface.

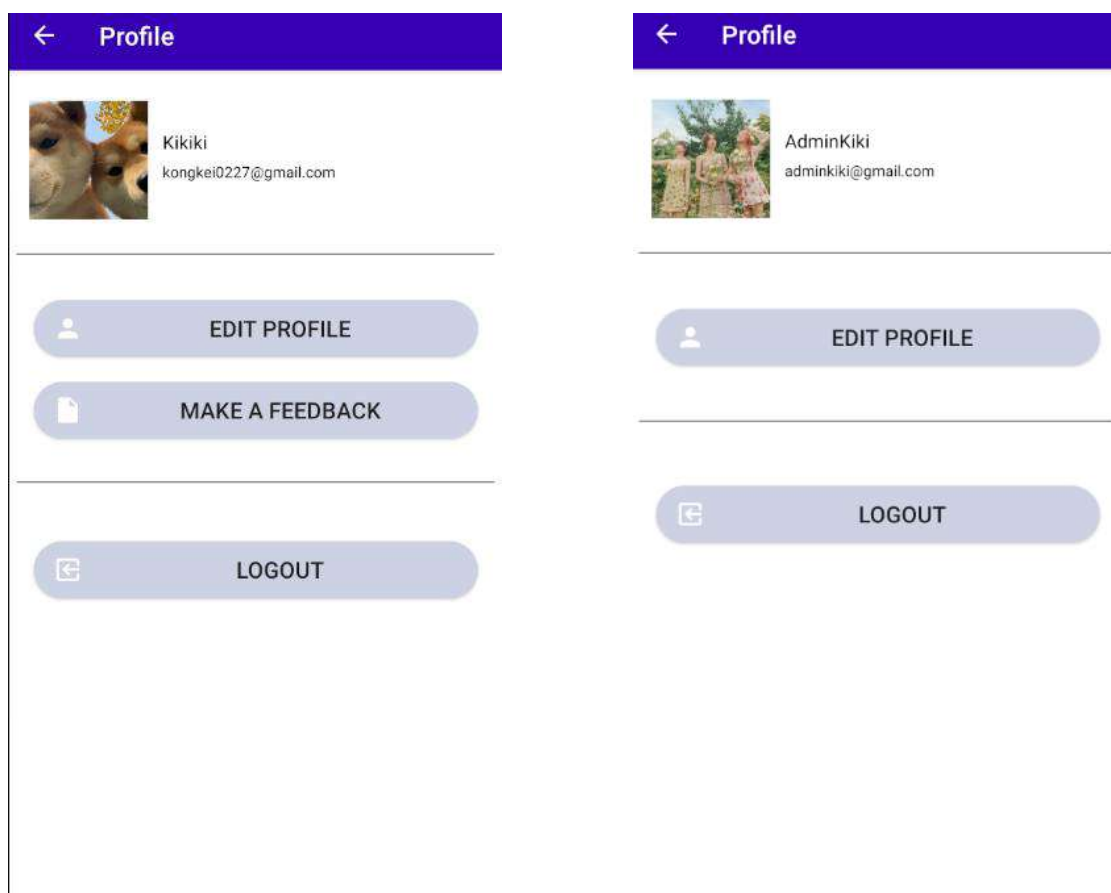


Figure 4. 11 Profile Interface for User (Left) and Admin (Right)

When clicking on the <<EDIT PROFILE>> button, a prompt will show as in Figure 4.12 that allow the user or admin to edit the username but the email is not editable. When the

username is valid and the actor click on the <<EDIT>>, then the username will change into the system and database.

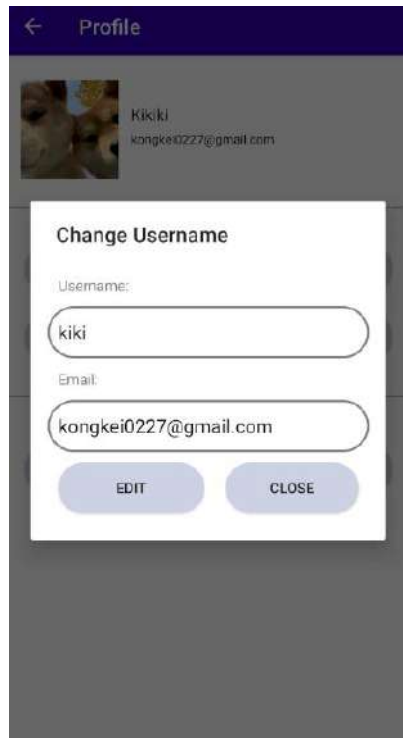


Figure 4. 12 Edit Profile Prompt in Profile Interface

When the username is blank and <<EDIT>> button is being clicked, an alert error will be display beside the username field as shown in Figure 4.13.

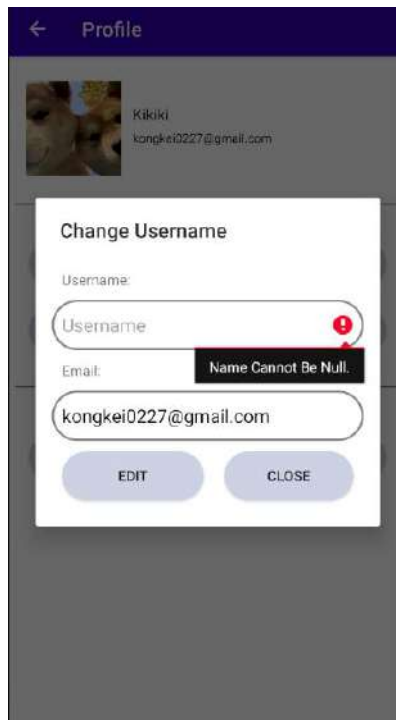
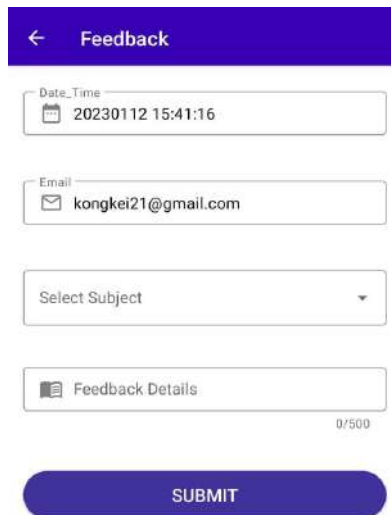


Figure 4. 13 Alert Error in Edit Profile Prompt

Figure 4.14 show the Feedback Interface for the user when the <<MAKE A FEEDBACK>> button in profile user is being clicked. The date and time field is generated automatically and is not editable, email field is not editable and the user can enter subject and feedback details and click on the <<SUBMIT>> button to make feedback.



The image shows a mobile application interface for providing feedback. At the top is a dark blue header with a white back arrow and the text "Feedback". Below the header are four input fields: 1. A date and time field labeled "Date_Time" with a calendar icon and the value "20230112 15:41:16". 2. An email field labeled "Email" with an envelope icon and the value "kongkei21@gmail.com". 3. A dropdown menu labeled "Select Subject" with a downward arrow. 4. A text area labeled "Feedback Details" with a document icon and a character count "0/500" at the bottom right. At the bottom of the form is a large, rounded, dark blue button with the text "SUBMIT" in white capital letters.

Figure 4. 14 User Feedback Interface

When the field is null and the user click on <<SUBMIT>>, an alert error will show at the respective field as shown in Figure 4.15. The text field box will turn red when the user exceeds the words.

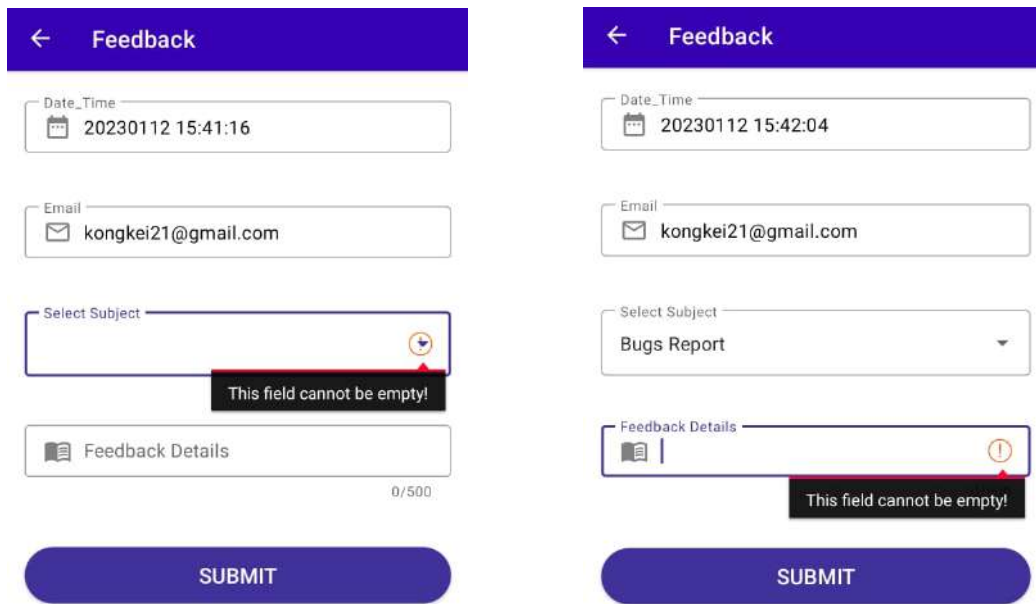


Figure 4. 15 Alert Error in Feedback Interface When Field is Blank.

Figure 4.16 show the alert error when the user exceeds the word limitation on details field and click <<SUBMIT>>.

The screenshot shows a feedback form with the following fields:

- Date_Time:** 20230112 15:42:15
- Email:** kongkei21@gmail.com
- Select Subject:** Chatbot
- Feedback Details:** A text area containing the message "chatbot not understand my question" repeated 15 times. A red border highlights the text, and a small error icon is visible on the left side of the text area. A character count "525/500" is shown at the bottom right of the text area.

At the bottom of the form is a blue "SUBMIT" button.

Figure 4. 16 Alert Error in Feedback Interface When Words Exceed.

Figure 4.17 shows the interface for the Feedback List Interface. When clicking at the bottom right navigation bar at the Admin Dashboard, the admin can view all the new feedback that made by the user.

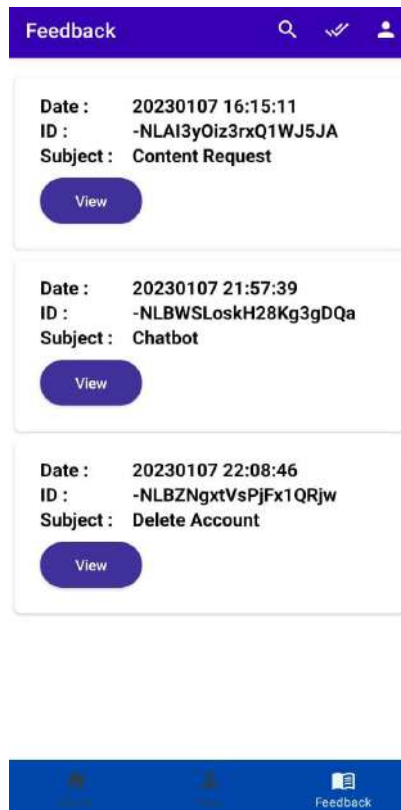


Figure 4. 17 Feedback List Interface

Figure 4.18 show when clicking on the <<View>> button on each feedback, a dialog interface will show and display the details of the respective feedback to let the admin to view the feedback. Admin can click on the <<Mark Done>> button when finish to read or settle the problem or issue that have been reported.

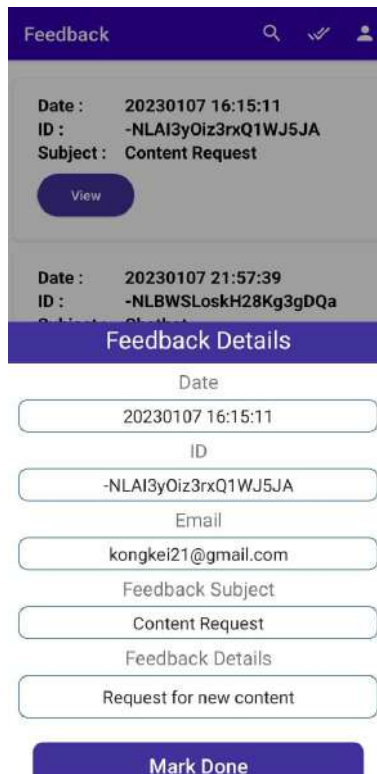


Figure 4. 18 Feedback Details Interface

Figure 4.19 show the Feedback history when clicking on the “Tick” icon at the top menu bar. Admin can view all the feedback that have been done at this interface.

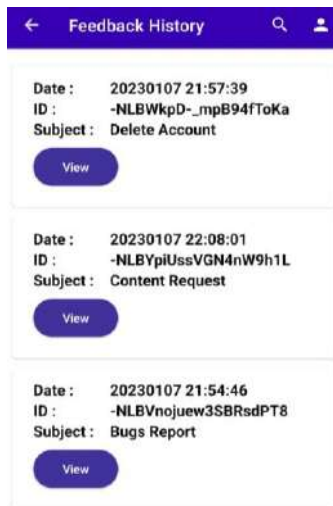


Figure 4. 19 Feedback History Interface

When clicking on the search icon at both the Feedback Admin and Feedback History Interface, the user can enter the subject title to search for the feedback. Figure 4.20 show the search interface for both the interface.

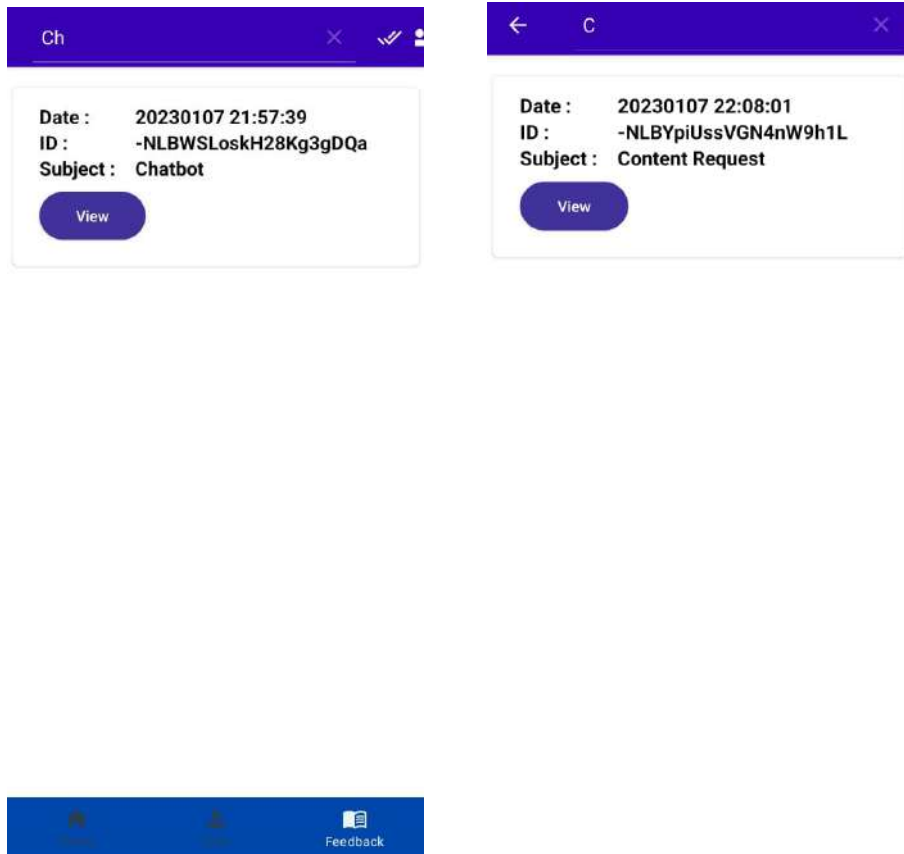


Figure 4. 20 Search Interface in Feedback List Interface (Left) and Feedback Done Interface (Right)

When the admin clicks on the <<User>> at the bottom navigation menu, the user list interface will show as Figure 4.21.

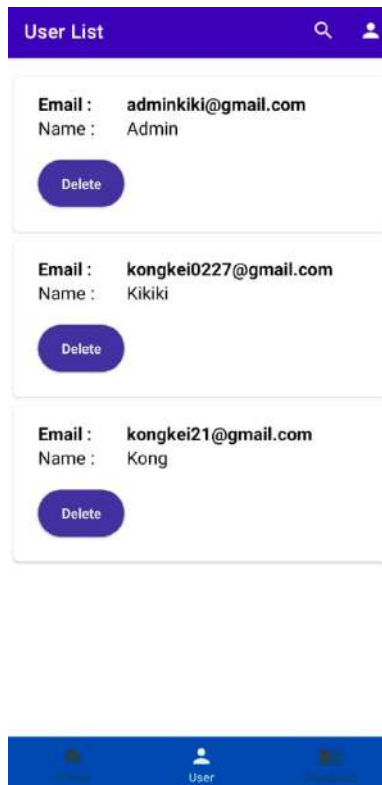


Figure 4. 21 User List Interface

When the admin clicks on the <<Delete>> button on each of the user information, a prompt will display to ask if the admin really want to deletes the user from the system as shown in Figure 4.22. If the admin clicks on Delete at the prompt, then the user data will be deleted from the system and database.

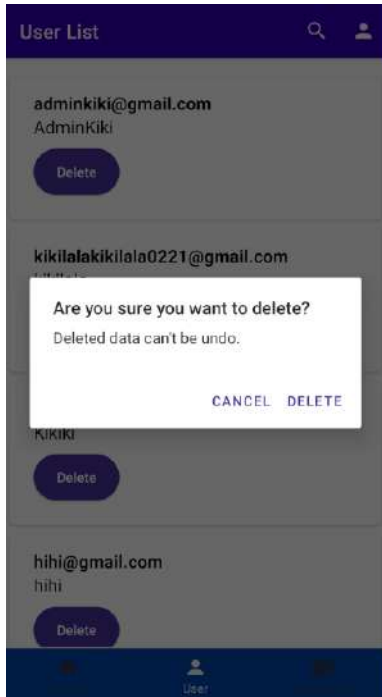


Figure 4. 22 Delete Prompt at User List Interface

Admin can search for specific user by clicking the “Search” icon at the top menu and entering the email of the user as shown in Figure 4.23.

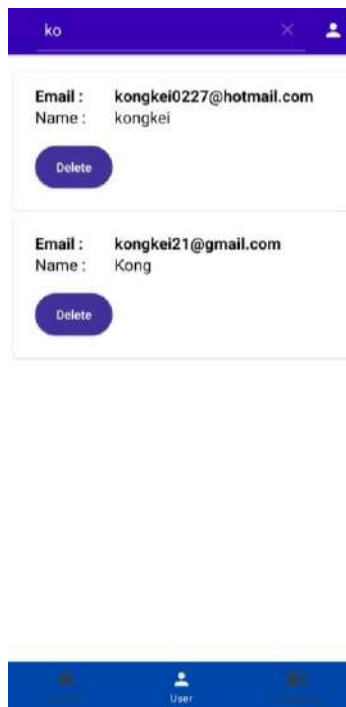


Figure 4. 23 Search Interface in User List Interface

4.2.1.4 Manage Diary

Figure 4.24 show the diary interface for the user. The user can be redirect to this interface by clicking the <<Diary>> icon and word at the bottom right of the menu. The user can add new diary by clicking the <<+>> button at the right bottom, view the details of the diary by clicking the <<Edit>> button, and delete the diary by clicking on the <<Delete>> button.

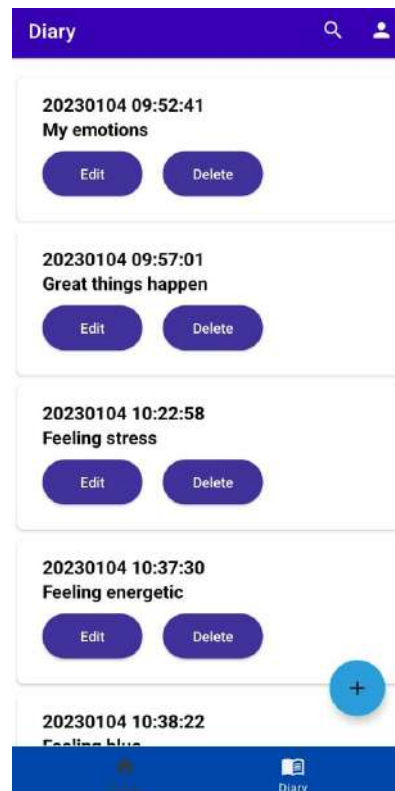


Figure 4. 24 Diary Interface

Figure 4.25 show the diary interface for the user to add a new diary. The user can add a new diary by entering the diary title, diary content and click on the <<ADD DIARY>> button. The Date_Time field is generate by the system and is not editable.

← Diary

Date_Time
20230104 09:52:41

T Diary Title
0/200

Diary Content
0/500

ADD DIARY

Figure 4. 25 Add Diary Interface

The diary title and diary content cannot be null or exceed 220 and 500 words respectively. If it is exceeding, an alert message will show at the respective field. Figure 4.26 show the null alerting while Figure 4.27 show the exceeding alerting.

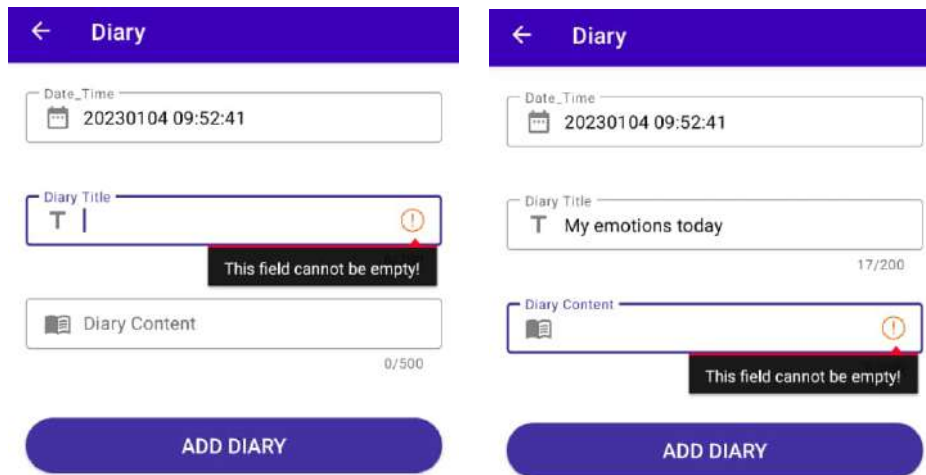


Figure 4. 26 Alert message on Null at Add Diary Interface

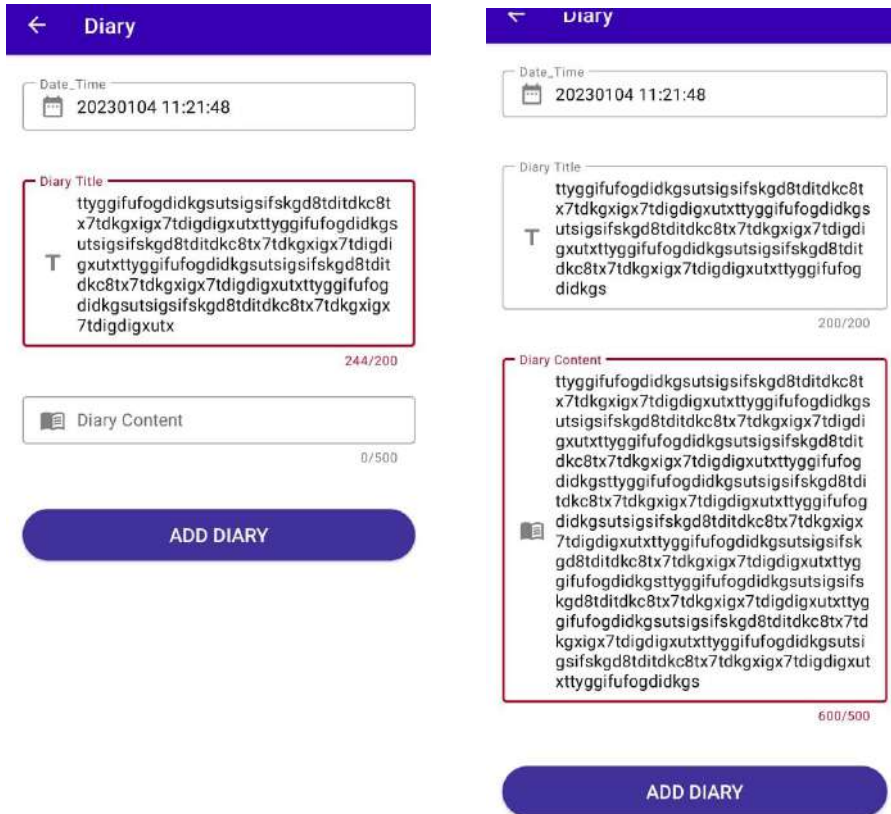


Figure 4. 27 Alert message on exceeding words in Add Diary Interface

Figure 4.28 show the details of the diary. User can update the diary after editing the field and clicks on the <<Update>> button. Date field is not editable.

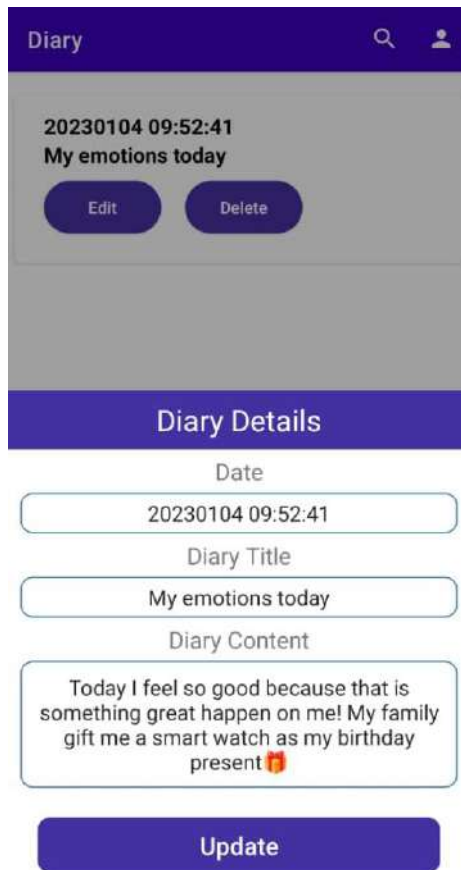


Figure 4. 28 Details of Diary Interface in Diary Interface

Figure 4.29 show the prompt when the user clicks on the <<Delete>> button to delete the specific diary. When user clicks on the <<DELETE>> button at the prompt. The specific diary will be deleted.

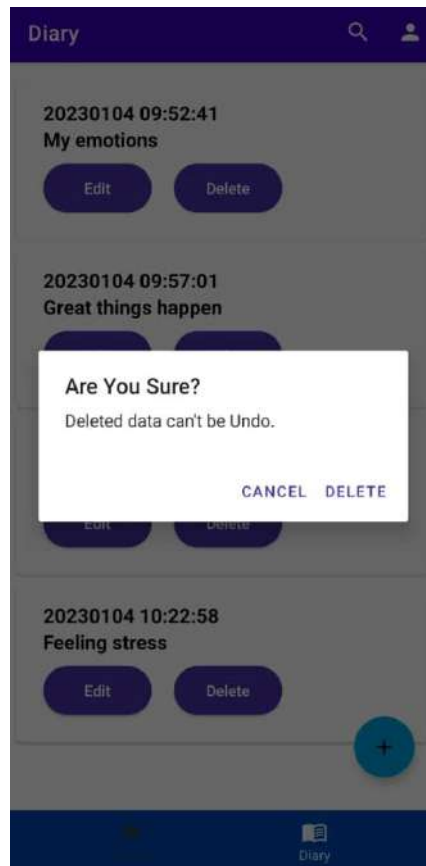


Figure 4. 29 Delete prompt at Diary Interface

User is able to search the diary by date by clicking on the search icon on the top right. Figure 4.30 show the interface for the search in diary interface.

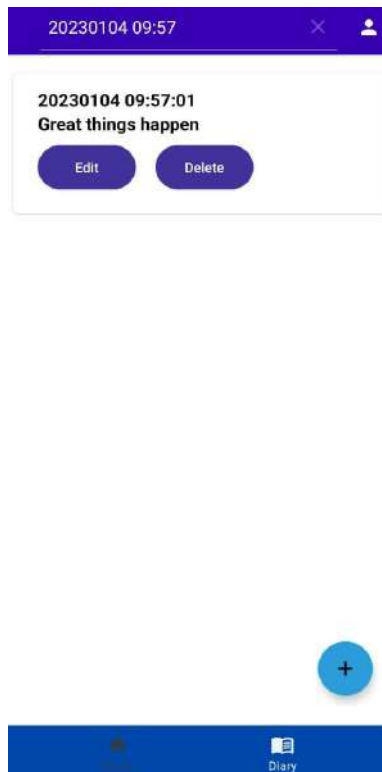


Figure 4. 30 Search in Diary Interface

4.2.1.5 Manage Information

Figure 4.31 show the Information Interface for admin. The admin can be redirect to this interface by clicking at the <<LEARN NOW>> button at the Admin Dashboard. The admin is able to add new information by clicking on the <<+>> button at the top right of the menu bar. The admin is also able to redirect tot the information website, edit and delete the information.

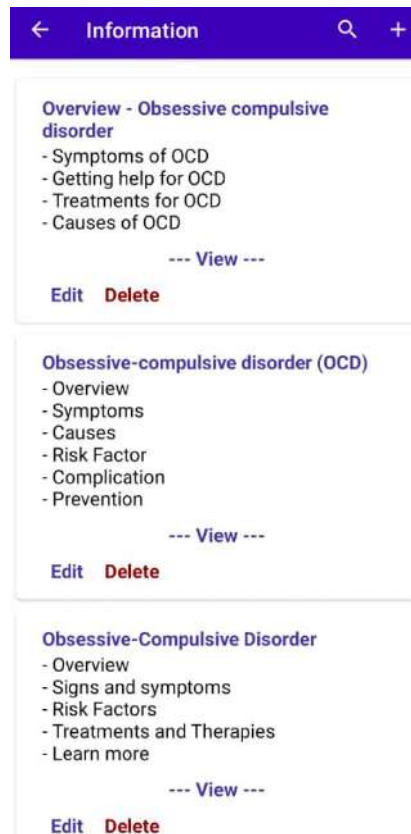


Figure 4. 31 Admin Information Interface

Figure 4.32 show the Add Information Interface for admin. The admin requires to enter URL link, title and content and click on the <<ADD INFORMATION >> button to add the new information.

← Add Information

🔗 URL Link

T URL Title

📄 URL Content

ADD INFORMATION

Figure 4. 32 Add Information Interface

All the field cannot be null. If the field is null or the URL link is not matching the format of a link, an alert message will show at the respective field. Figure 4.33 show the alert message on the field.

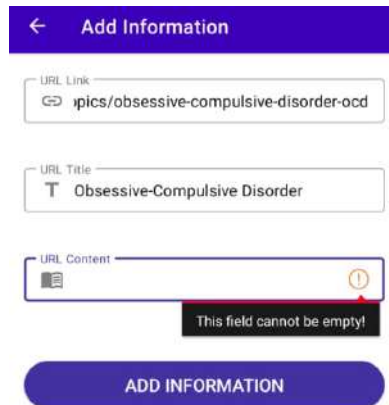
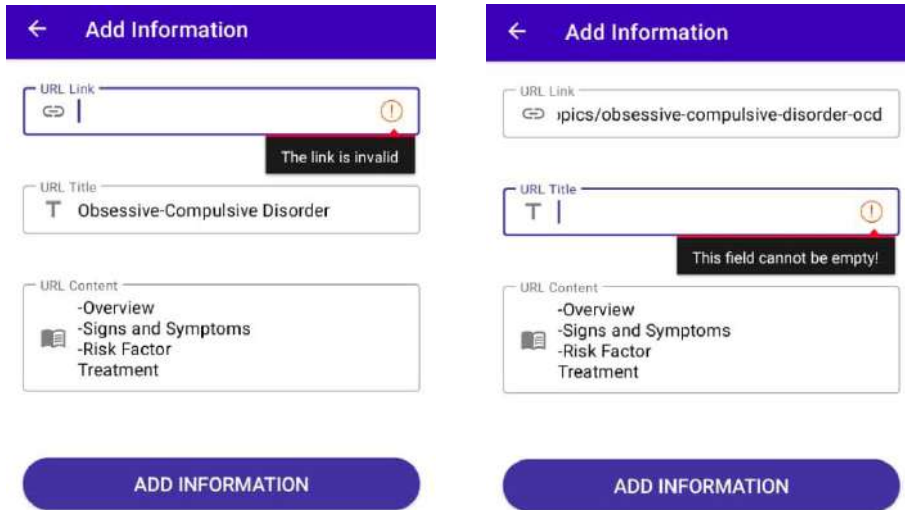


Figure 4. 33 Alert Message on Add Information Interface

Figure 4.34 show the details of the information and the admin are able to change the details of the information by clicking on the <<Edit>> button, editing the field and click on <<Update>> button.

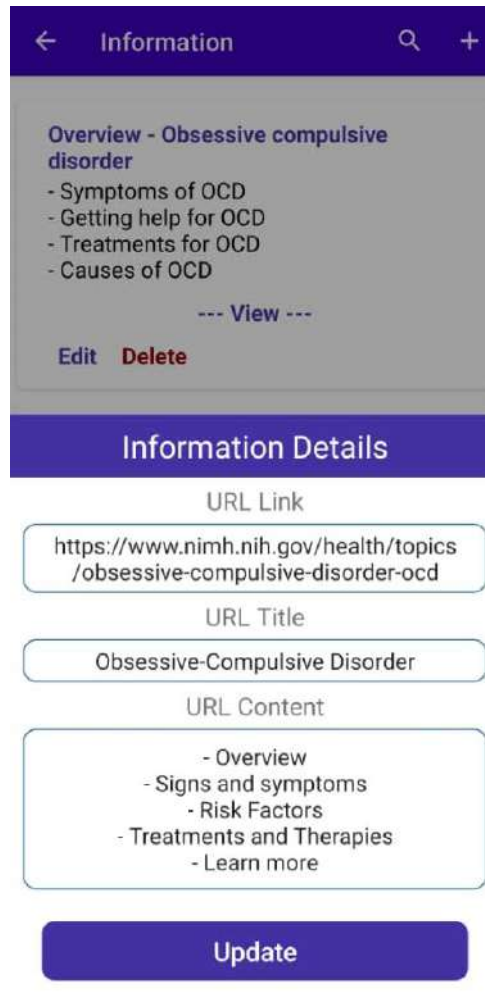


Figure 4. 34 Details of information at Information Interface for admin

Figure 4.35 show the delete prompt when the admin clicks on the <<Delete>> button.

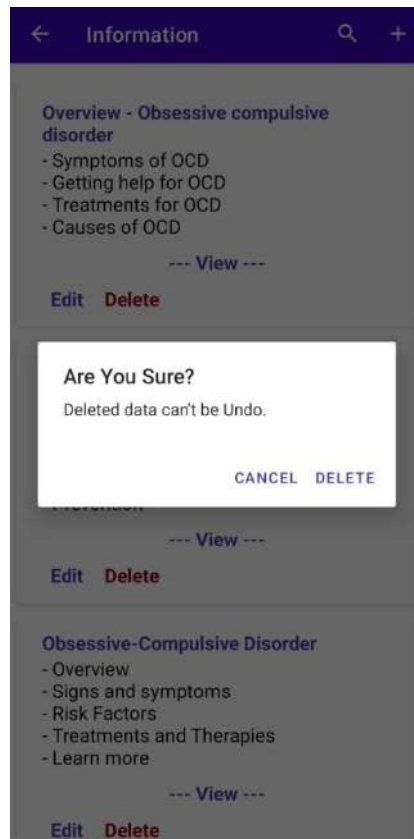


Figure 4. 35 Delete prompt at Information Interface for admin

Figure 4.36 show the Information Interface for the user. The user can click on the <<LEARN NOW>> button to redirect to this interface at the user dashboard. The user can click on the <<View>> button to redirect to the website for the information.

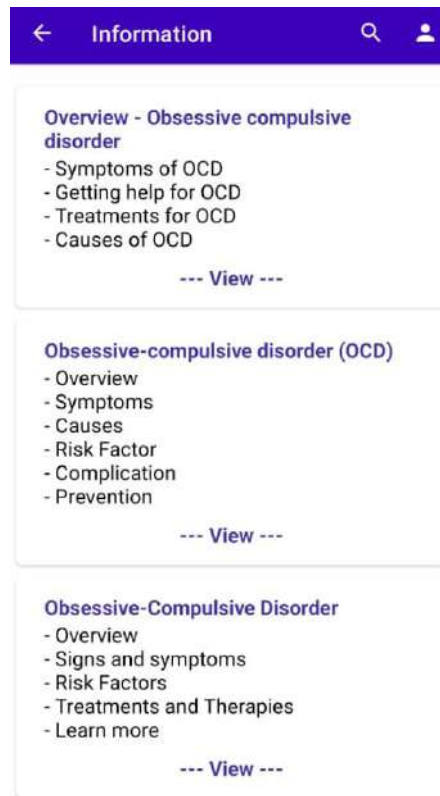


Figure 4. 36 Information Interface for user

Figure 4.37 show the search interface for the admin and user while they click on the search icon on the top right of the menu.

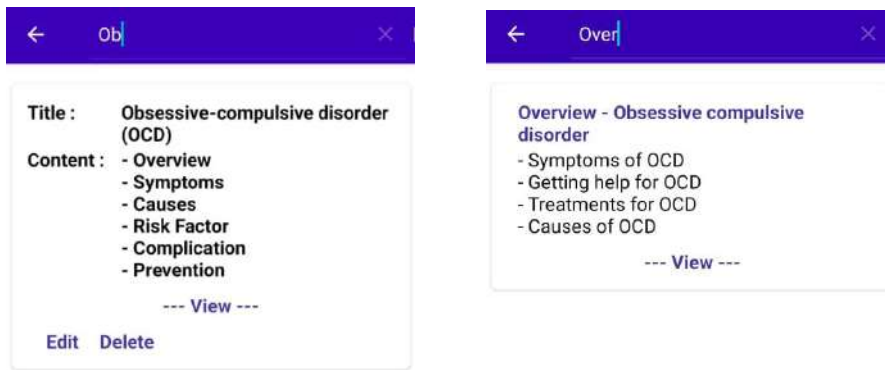


Figure 4. 37 Search Interface at Information Interface for admin(left) and user(right)

4.2.1.6 Manage Chatting

Figure 4.38 show the Chatbot Interface for both the admin and user when they click on the <<CHAT NOW>> button on the dashboard. User can type and click the send button to send message to the chatbot and get response from it. The user can also choose from the hint or suggestion above the text box to send question to the chatbot.

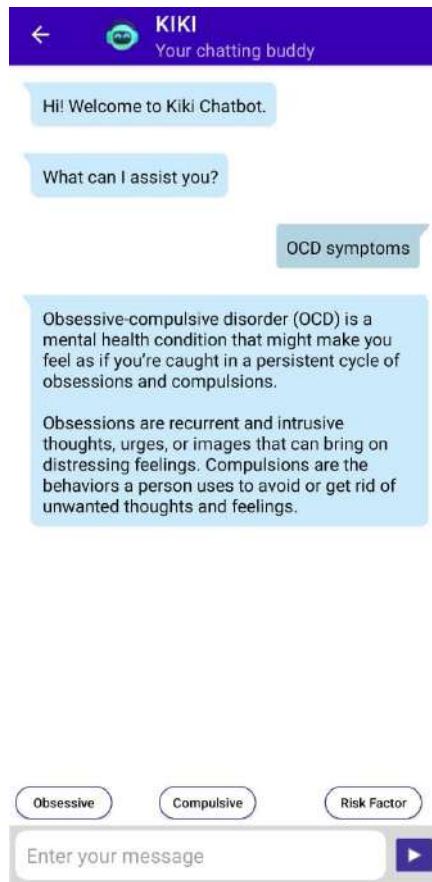


Figure 4. 38 Chatbot Interface for both the admin and user.

4.2.1.7 Manage Self-Test

Figure 4.39 show the Self-Assessment Interface for the user and the admin when the user or admin click on the <<TEST NOW>> button on the Dashboard.

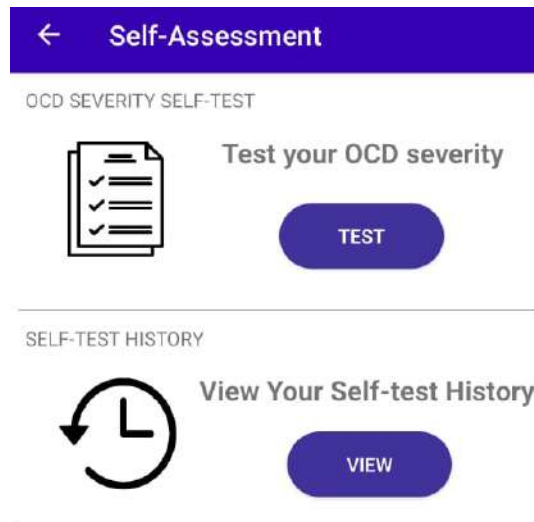


Figure 4. 39 Self-Assessment Interface for User and Admin

Figure 4.40 show the Self-Assessment Test Interface for the user and admin when they click on the <<TEST>> button from the Self-Assessment Interface. The user and admin can pull the answer bar to what the context of their feelings and click on the <<Submit>> button.

←
Self-Assessment Test

0-None 1-Mild 2-Moderate 3-Severe 4-Highest

1. TIME OCCUPIED BY OBSESSIVE THOUGHTS
 Q: How frequently do the obsessive thoughts occur?

Q1: 0

0-None 1-Mild 2-Moderate 3-Severe 4-Extreme

2. INTERFERENCE DUE TO OBSESSIVE THOUGHTS
 Q: How much do your obsessive thoughts interfere with your social or work (or role) functioning?

Q2: 0

0-None 1-Mild 2-Moderate 3-Severe 4-Extreme

3. DISTRESS ASSOCIATED WITH OBSESSIVE THOUGHTS
 Q: How much distress do your obsessive thoughts cause you? [distress can refer as anxiety or disturbing]

Q3: 0

8. DISTRESS ASSOCIATED WITH COMPULSIVE BEHAVIOR
 Q: How would you feel if prevented from performing your compulsion(s)? [Is prevented from performing your compulsion make you anxious?]

Q8: 0

0-Makes an effort to always resist, or symptoms so minimal doesn't need to actively resist
 1-Tries to resist most of the time
 2-Makes some effort to resist
 3-Yields to almost all compulsions without attempting to control them, but does so with some reluctance
 4-Completely and willingly yields to all compulsions

9. RESISTANCE AGAINST COMPULSIONS
 Q: How much of an effort do you make to resist the compulsions? [Only rate effort made to resist, not success or failure in actually controlling the compulsions.]

Q9: 0

0-Complete control
 1-Much control
 2-Moderate control
 3-Little control
 4-No control

10. DEGREE OF CONTROL OVER COMPULSIVE BEHAVIOR
 Q: : How strong is the drive to perform the compulsive behavior? [How much control do you have over the compulsions?]

Q10: 0

SUBMIT

Figure 4. 40 Self-Assessment Test Interface for User and Admin

Figure 4.41 show the Self-Assessment Result Interface for the user and the admin after they click on the <<Submit>> button from the Self-Assessment Test Interface. In this interface, the risk of the user or admin getting the obsessive and compulsive OCD will be shown in a graphical form and there are some advice to the user or admin. Pressing the back button will redirect the user back to the Self-Assessment Interface.

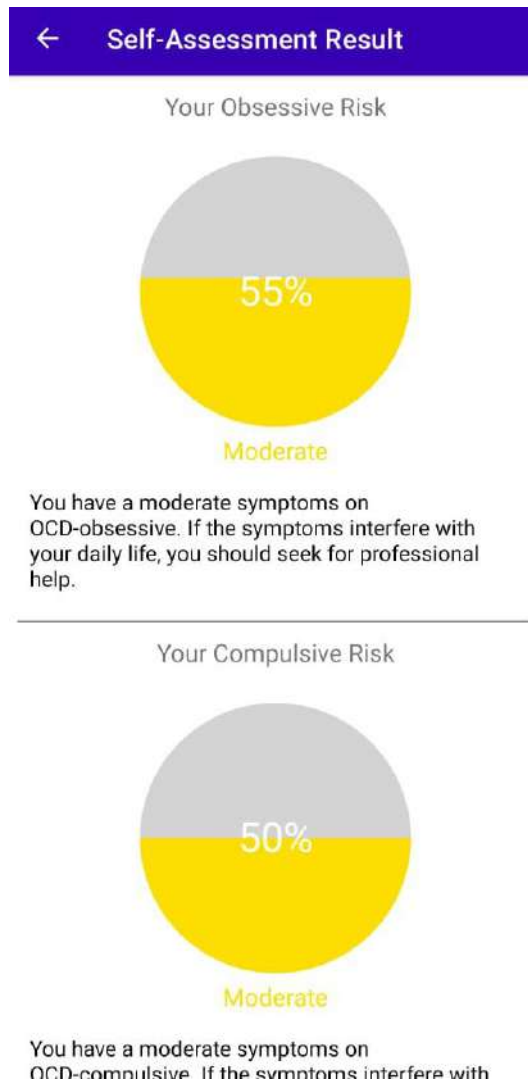


Figure 4. 41 Self-Assessment Result Interface for User and Admin

Figure 4.42 show the Self-Assessment History Interface when the user or admin click on the <<VIEW>> button on the Self-Assessment Interface. In this interface, the user are able to view all the history of their self-assessment test.

← Self-Assessment History		
Date and Time	Obsessive Category	Compulsive Category
20230108 22:45:20	Nearly Absent	Nearly Absent
20230108 23:32:56	Severe	Extreme
20230109 23:40:57	Moderate	Moderate
20230109 23:43:50	Mild	Mild
20230110 00:06:48	Severe	Severe
20230110 14:22:00	Moderate	Extreme
20230110 14:25:53	Extreme	Extreme
20230110 14:30:30	Severe	Nearly Absent
20230110 14:33:19	Severe	Nearly Absent
20230110 14:34:59	Severe	Nearly Absent
20230110 14:44:52	Mild	Nearly Absent
20230110 14:45:07	Mild	Nearly Absent
20230110 14:54:57	Moderate	Moderate
20230110 15:14:22	Severe	Severe
20230110 15:48:25	Severe	Severe
20230110 15:49:57	Extreme	Moderate
20230110 15:53:36	Severe	Moderate
20230110 16:14:15	Moderate	Moderate
20230112 14:34:13	Nearly Absent	Nearly Absent
20230112 14:34:32	Moderate	Moderate

Figure 4. 42 Self-Assessment History Interface for Admin and User

4.3 IMPLEMENTATION OF DIALOGFLOW

Google Dialogflow ES is implemented in this project and application. By inserting several possible user's question, Google Dialogflow ES will identify the keyword. When the user ask question that not inside but similar with the script, then the Dialogflow will able to capture the keyword and display the response or information to the user. There are several parts inside the script which included the symptom of OCD, Risk Factor of OCD and Treatment of OCD. The user will be redirected to the opening greeting when reaching the closing part.

Figure 4.43 show the overall intent created in Dialogflow ES to create the chatbot in this application. Figure 4.44 show an example how the chatbot is being trained.

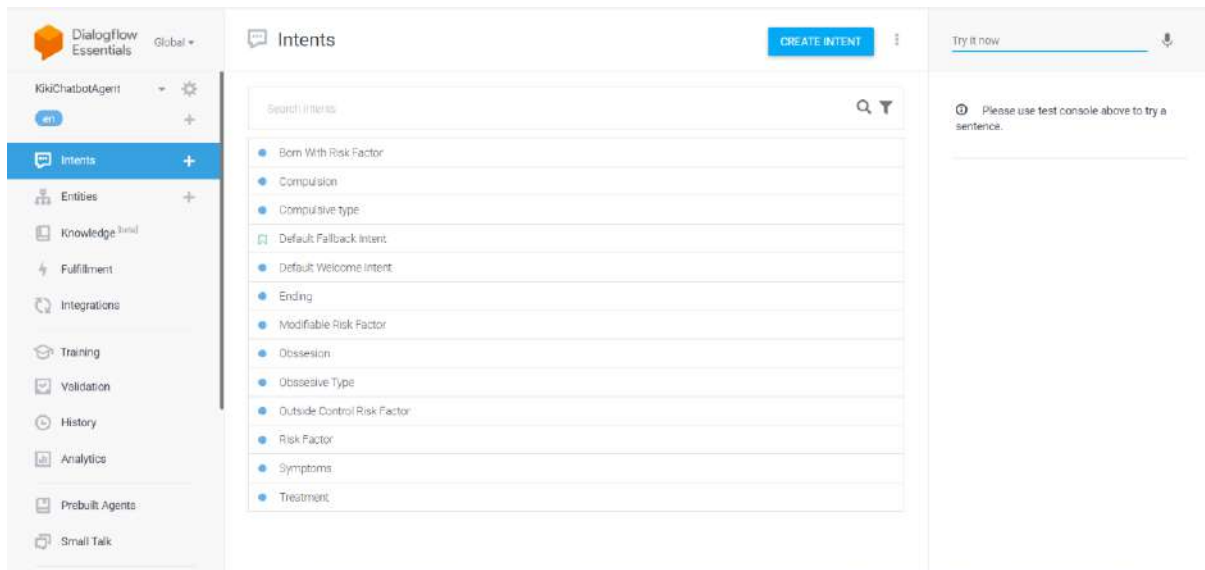


Figure 4. 43 Intent Created in Dialogflow

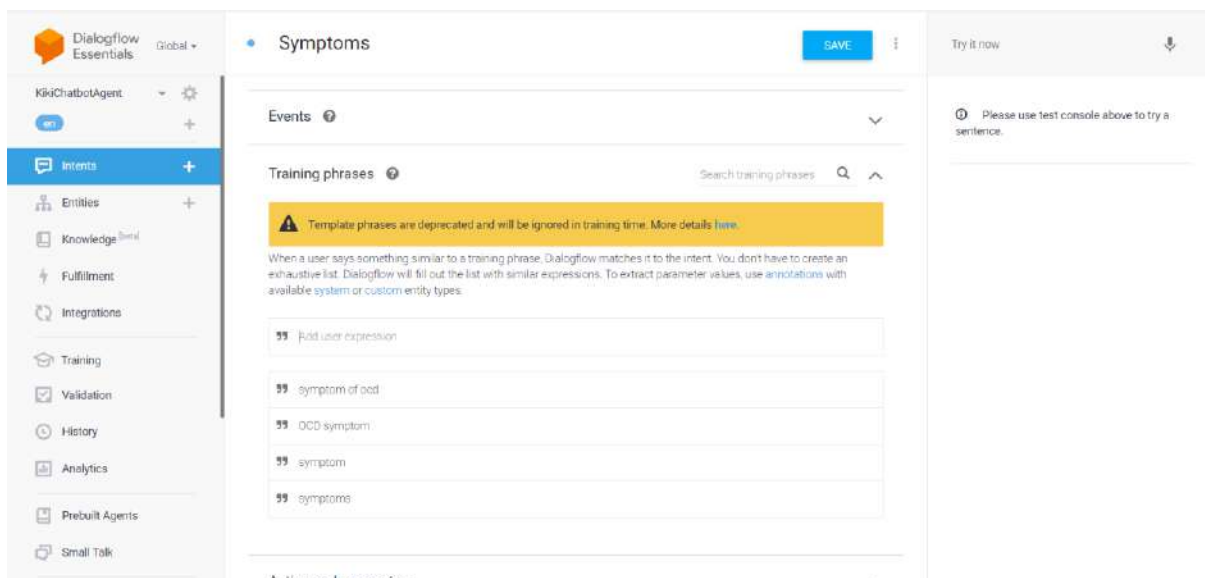


Figure 4. 44 Example of Intent Symptoms

4.4 TESTING AND RESULT DISCUSSION

Kiki Mobile Chatbot Application is developed using Android Studio, Java programming language, DialogFlow for training chatbot and Firebase as database. In the process of development, it is vital to have a testing activity throughout the development stages so that all the function in the app is error free and meets all the requirement. In our project, UAT testing is conducted to test the function of the application conforms with the requirement stated in

Software Requirement Specification (SRS). The UAT test report can be referred in Appendix D.

4.4.1 System Functionality

There are four types of functional testing that will be carry out in this testing activity that is unit testing, integration testing, system testing and user acceptance testing (UAT) to test Kiki Mobile Chatbot Application.

i. Unit Testing

Unit testing is carried out to test the smallest testable parts of an application individually and independently in the development process. The unit will be tested repeatedly until all the errors are being fixed.

ii. Integration Testing

Integration testing is the software testing that the individual software modules are combined and tested as a group. This testing is to evaluate the compliance the compliances of the components with the functional requirement.

iii. System Testing

System testing is a testing that conducted to test on the complete integrated system to evaluate the system's compliance with its specified requirement.

iv. User Acceptance Testing (UAT)

The purpose of conducting this UAT is to ensure the developed and deployed system meets the requirement that is stated in the Software Requirements Specification (SRS). The testing is also to ensure the system is accepted by the end user.

CHAPTER 5

CONCLUSION

5.1 INTRODUCTION

This chapter is used to conclude the development of the KIKI: A MOBILE CHATBOT APPLICATION FOR OCD. In conclusion, the target user for this application is those who suffer with OCD, who feel like having OCD and those who want to know more about OCD. Through this application, the user can write diary to express their feeling or emotion, chatting to the chatbot to get information, taking the OCD test to test their severity and getting reference of information from the website.

There are total of 5 chapter in this project. For chapter 1, the objectives, scope and problem stated are clarify. For chapter 2, application with similar function is compare and done through the literature review. Based on the literature review, most of the existing application require to pay for most of their function and there is no a specific chatbot application for OCD. For chapter 3, the methodology used in developing this project is discussed that is the agile methodology. All the system flow and requirements are discussed in this chapter. For chapter 4, it discussed about the output of the project. Output of the development of application is discussed in this chapter such as interface, function and coding of the application. Besides the development result, the testing result and conclusion is also discussed in this chapter. In conclusion, the project is done but there exists limitation and constraint, and it can be improved to make a better application in the future.

5.2 LIMITATION

There are some limitations in this KIKI: A MOBILE CHATBOT APPLICATION FOR OCD. The first limitation is the application is only applicable to mobile but not web service. The second limitation is this application is developed using android studio and it cannot run on the mobile that is not using Android. The third limitation is all the searching function in the

application, the user needs to key in accurate word, which means the capital letters or small letters alphabets cannot be ignore because it will affect the search result.

5.3 FUTURE WORK

The future work of this project can be done by improving the limitations stated above. For example, develop the application in others languages so that the application can be used in another operating system in mobile other than Android.

REFERENCES

- Caitlin Bell. (n.d.). *Wysa App Review 2022: Pros & Cons, Cost, & Who It's Right For - Choosing Therapy*. Retrieved January 17, 2023, from <https://www.choosingtherapy.com/wysa-app-review/>
- data.ai. (n.d.). *Nuna: Mental Health Companion App Ranking and Store Data | data.ai*. Retrieved May 31, 2022, from <https://www.data.ai/en/apps/google-play/app/io.aya/>
- Kate Brush. (2019). *What is Agile Software Development (Agile Methodologies)?* <https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development>
- Kate Ta. (2020). *Traditional vs Agile SDLC: How To Skyrocket Your Project With Agile Model*. <https://agiletech.vn/traditional-sdlc-vs-agile-sdlc/>
- Kross, E., Verduyn, P., Demiralp, E., Park, J., Lee, D. S., Lin, N., Shablack, H., Jonides, J., & Ybarra, O. (2013). Facebook Use Predicts Declines in Subjective Well-Being in Young Adults. *PLoS ONE*, 8(8). <https://doi.org/10.1371/journal.pone.0069841>
- Mayo Clinic Staff. (2019). *Mental illness - Symptoms and causes - Mayo Clinic*. <https://www.mayoclinic.org/diseases-conditions/mental-illness/symptoms-causes/syc-20374968>
- Relief Seeker. (n.d.). *Innerhour App Review (2021) Best Health Apps for Treating Depression and Anxiety*. Retrieved May 31, 2022, from <https://reliefseeker.com/reviews/innerhour>
- Smitha Bhandari, M. (2020). *OCD: Types, Symptoms, Causes, Diagnosis, Treatment, and Related Conditions*. <https://www.webmd.com/mental-health/obsessive-compulsive-disorder>
- Sohail Merchant. (2021). *These Chatbots Are Helping With Mental Health Right Now - AIM*. <https://www.aimblog.io/2021/03/02/these-chatbots-are-helping-with-mental-health-right-now/>
- YALE-BROWN OBSESSIVE COMPULSIVE SCALE (Y-BOCS)*. (n.d.).

APPENDIX A
SOFTWARE REQUIREMENT SPECIFICATION (SRS)


2022

SOFTWARE REQUIREMENT SPECIFICATION (SRS)

[KIKI: A CHATBOT APPLICATION FOR OCD
IN ANDROID MOBILE APPLICATION]



DOCUMENT APPROVAL

	Name	Date
Authenticated by:  _____ Name	KONG KEI	08/02/2023
Approved by: _____ Client	DR. ANIS FARIHAN BINTI MAT RAFFEI	08/02/2023

Software : Draw IO, Microsoft Word

Archiving Place : Google Drive

TABLE OF CONTENT

CONTENT	PAGE
DOCUMENT APPROVAL	ii
TABLE OF CONTENT.....	iii
LIST OF FIGURES	v
LIST OF TABLES	viii
LIST OF APPENDICES	ix
CHAPTER 1.....	1
1.1 PROJECT DESCRIPTION.....	1
1.2 SYSTEM IDENTIFICATION	1
1.3 CONTEXT DIAGRAM	1
1.4 DATA FLOW DIAGRAM.....	2
CHAPTER 2.....	4
2.1 USE CASE DIAGRAM AND DESCRIPTION	4
2.1.1 Manage Registration.....	5
2.1.2 Manage Chatting.....	8
2.1.3 Manage Self-test	9
2.1.4 Manage Information	11
2.1.5 Manage Diary	14
2.1.6 Manage Profile	16
2.2 SEQUENCE DIAGRAM.....	21
2.2.1 Manage Registration.....	21
2.2.2 Manage Chatting.....	23
2.2.3 Manage Self-test	24
2.2.4 Manage Information	26
2.2.5 Manage Diary	29
2.2.6 Manage Profile	31
CHAPTER 3.....	36
3.1 INTERFACE DESIGN	36
3.1.1 Manage Registration.....	36

3.1.2 Manage Profile	44
3.1.3 Manage Diary	55
3.1.4 Manage Information	61
3.1.5 Manage Chatting.....	68
3.1.1 Manage Self-Assesment	69
3.2 HARDWARE AND SOFTWARE SPECIFICATION.....	73

LIST OF FIGURES

Figure 1. 1: Context Diagram of Kiki Chatbot Mobile Application.....	2
Figure 1. 2: Data Flow Diagram of Kiki Chatbot Mobile Application.....	3
Figure 2. 1: Use Case Diagram for Kiki Chatbot Mobile Application	4
Figure 2. 2: Use Case for Manage Registration	5
Figure 2. 3: Use Case for Manage Chatting.....	8
Figure 2. 4: Use Case for Manage Self-test	9
Figure 2. 5: Use Case for Manage Information	11
Figure 2. 6: Use Case for Manage Diary	14
Figure 2. 7: Use Case for Manage Profile.....	16
Figure 2. 8 Manage Registration Basic Flow Sequence Diagram	21
Figure 2. 9 Manage Registration Alternative Flow Sequence Diagram	22
Figure 2. 10 Manage Registration Exception Flow Sequence Diagram	23
Figure 2. 11 Manage Chatting Basic Flow Sequence Diagram	24
Figure 2. 12 Manage Chatting Alternative Flow Sequence Diagram	24
Figure 2. 13 Manage Self-test Basic Flow Sequence Diagram	25
Figure 2. 14 Manage Self-test Alternative Flow Sequence Diagram	26
Figure 2. 15 Manage information Basic Flow Sequence Diagram	27
Figure 2. 16 Manage Information Alternative Flow Sequence Diagram.....	29
Figure 2. 17 Manage Diary Basic Flow Sequence Diagram.....	30
Figure 2. 18 Manage Diary Alternative Flow Sequence Diagram.....	31
Figure 2. 19 Manage Profile Basic Flow Sequence Diagram	32
Figure 2. 20 Manage Profile Alternative Flow Sequence Diagram.....	35
Figure 3. 1 Main Interface	36
Figure 3. 2 Login Interface.....	37

Figure 3. 3 Alert Error on Login Interface	38
Figure 3. 4 Pop Up Notification for Invalid Email or Password in Login Interface	38
Figure 3. 5 Reset Password Interface	39
Figure 3. 6 Alert in Rest Password Interface.....	40
Figure 3. 7 Register Interface	41
Figure 3. 8 Alert in Register Interface.....	42
Figure 3. 9 Dashboard Admin Interface	43
Figure 3. 10 Dashboard User Interface	43
Figure 3. 11 Profile Interface for User (Left) and Admin (Right)	44
Figure 3. 12 Edit Profile Prompt in Profile Interface	45
Figure 3. 13 Alert Error in Edit Profile Prompt	46
Figure 3. 14 User Feedback Interface	47
Figure 3. 15 Alert Error in Feedback Interface When Field is Blank.	48
Figure 3. 16 Alert Error in Feedback Interface When Words Exceed.	49
Figure 3. 17 Feedback List Interface.....	50
Figure 3. 18 Feedback Details Interface.....	51
Figure 3. 19 Feedback History Interface	51
Figure 3. 20 Search Interface in Feedback List Interface (Left) and Feedback Done Interface (Right).....	52
Figure 3. 21 User List Interface.....	53
Figure 3. 22 Delete Prompt at User List Interface	54
Figure 3. 23 Search Interface in User List Interface	54
Figure 3. 24 Diary Interface	55
Figure 3. 25 Add Diary Interface	56
Figure 3. 26 Alert message on Null at Add Diary Interface.....	57
Figure 3. 27 Alert message on exceeding words in Add Diary Interface	58

Figure 3. 28 Details of Diary Interface in Diary Interface	59
Figure 3. 29 Delete prompt at Diary Interface	60
Figure 3. 30 Search in Diary Interface	61
Figure 3. 31 Admin Information Interface	62
Figure 3. 32 Add Information Interface	63
Figure 3. 33 Alert Message on Add Information Interface	64
Figure 3. 34 Details of information at Information Interface for admin	65
Figure 3. 35 Delete prompt at Information Interface for admin.....	66
Figure 3. 36 Information Interface for user	67
Figure 3. 37 Search Interface at Information Interface for admin(left) and user(right).....	68
Figure 3. 38 Chatbot Interface for both the admin and user.....	69
Figure 3. 39 Self-Assessment Interface for User and Admin	70
Figure 3. 40 Self-Assessment Test Interface for User and Admin	71
Figure 3. 41 Self-Assessment Result Interface for User and Admin	72
Figure 3. 42 Self-Assessment History Interface for Admin and User.....	73

LIST OF TABLES

Table 2. 1 Relationship between module function and requirements	4
Table 2. 2 Use Case Description for Manage Registration.....	5
Table 2. 3 Use Case Description for Manage Chatting	8
Table 2. 4 Use Case Description for Manage Self-test.....	10
Table 2. 5: Use Case Description for Manage Information	11
Table 2. 6: Use Case Description for Manage Diary	14
Table 2. 7: Use Case Description for Manage Profile	16
Table 3. 1 Hardware Specification.....	74
Table 3. 2 Software Specification.....	74

LIST OF APPENDICES

CHAPTER 1

1.1 PROJECT DESCRIPTION

This is a project named Kiki chatbot which is a mobile application that provide the guidance and information of obsessive-compulsive disorder (OCD) to users. The project is to allow the user to chat with the chatbot inside the mobile application to get information, doing a self-assessment to test the severity of the OCD. The project also allows the user to get more information about the OCD in website form. The project will provide a diary section which allow the user to record their daily emotions. The Kiki chatbot mobile application consist of 6 modules which are manage registration, manage chatting, manage self-test, manage information, manage diary and manage profile.

1.2 SYSTEM IDENTIFICATION

This document uses the following convention:

System Identification Number: **SRS-KIKI-V01-22**

SRS: Software Requirement Specification

KIKI: Kiki chatbot mobile application

V01: Version 1

22: Year 2022

The format of recording the identification of the system is shown as the above information. SRS stands for Software Requirement Specification which is a type of document that organize and present the requirement of the project and how the project being executed. KIKI stands for the Kiki chatbot mobile application. V01 stands for Version 1 and 22 stands for Year 2022 which is the year the project being developed.

1.3 CONTEXT DIAGRAM

Figure 1.1 show the context diagram for the Kiki Chatbot Mobile Application.

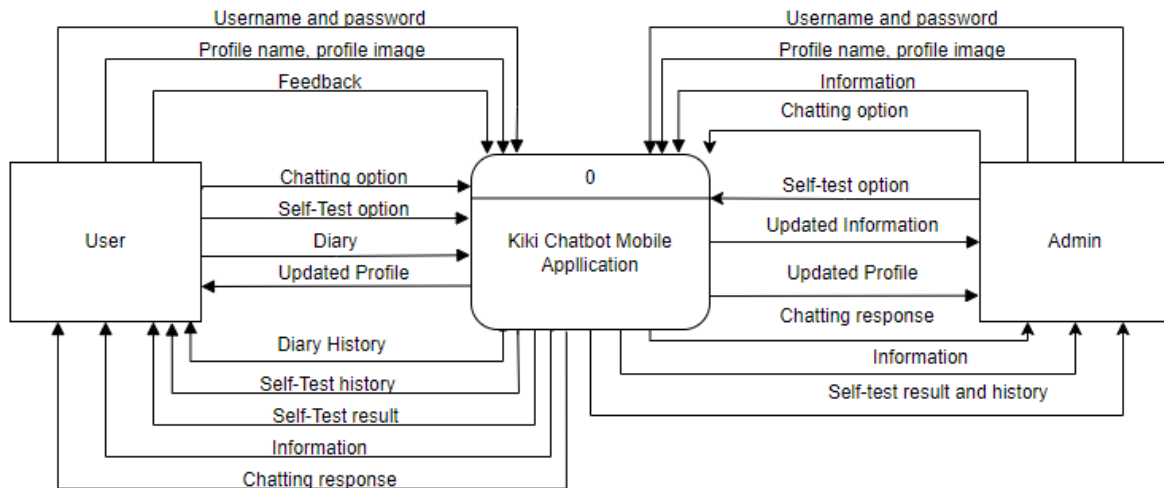


Figure 1. 1: Context Diagram of Kiki Chatbot Mobile Application

The context diagram consists of two entities which are user and admin. Both the admin and user need to enter correct username and password to access the system and retrieve data from the system. User can access profile, feedback, chat, self-test and diary. All these data will send and save in the system. Then, the system will pass the profile, diary, self-test, self-test history, information and chatting response data to the user. The admin can access profile, information, chatting, self-test, and these data will save to the system. The system will return information, profile, chatting, self-test, self-test history data to the admin.

1.4 DATA FLOW DIAGRAM

Figure 1.2 show the data flow diagram for the Kiki Chatbot Mobile Application. There are 5 process that involve in this system which are registration, profile, self-test, information and diary. Next, there are two entities in this system which are user and admin. Then, there are 5 data store which are users, feedback, information, test and diary. This data flow diagram shows how the data flow and the interaction between the processes, data stores and entities.

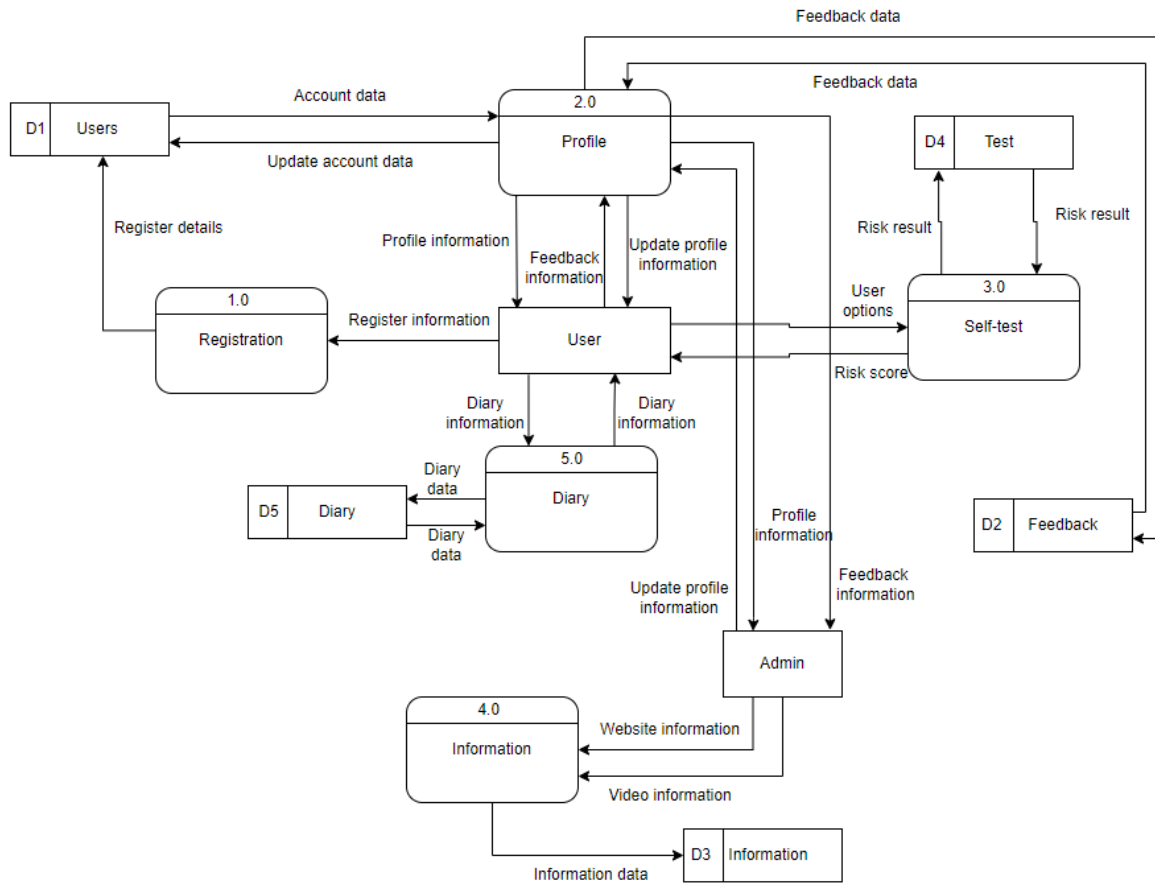


Figure 1. 2: Data Flow Diagram of Kiki Chatbot Mobile Application

CHAPTER 2

2.1 USE CASE DIAGRAM AND DESCRIPTION

The functionality of the Kiki Chatbot Mobile Application will be modeled in this section using the use case diagram and use case description. Figure 2.1 shows the overall use case diagram of this system. There are six modules in this system which are manage registration, manage chatting, manage self-test, manage information, manage diary, manage profile.

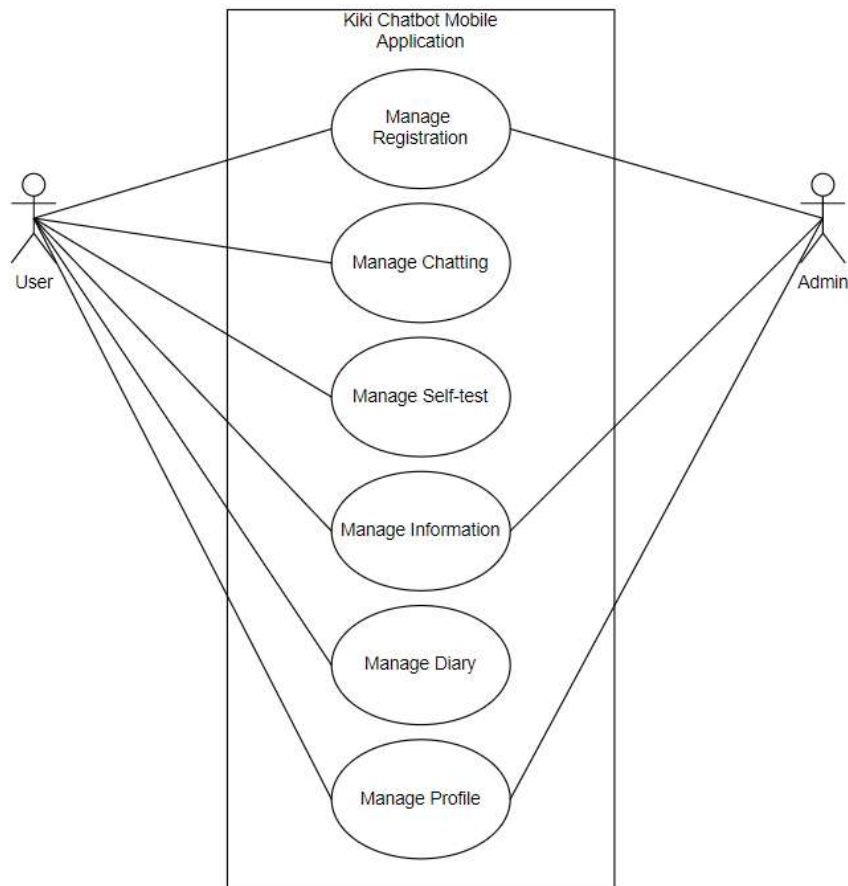


Figure 2. 1: Use Case Diagram for Kiki Chatbot Mobile Application

Table 2. 1 Relationship between module function and requirements

Module Name	Functions	Actor
Manage Registration	The use case allows the user to create account and allow the user and admin to login to the system.	User, Admin
Manage Chatting	The use case allows the user to chat using text through chatbot Kiki.	User

Manage Self-test	The use case allows the user to take a self-assessment to test their severity on OCD.	User
Manage Information	The use case allows both the admin and user to view the information and the admin to add, edit and delete website information about OCD in the system.	User, Admin
Manage Diary	The use case allows the user to view, add or delete diary in the system.	User
Manage Profile	The use case allows the user and admin view and edit their profile, delete their account. The system allows the user to make feedback in the system. The system allows the admin to view feedback made by user in the system.	User, Admin

2.1.1 Manage Registration

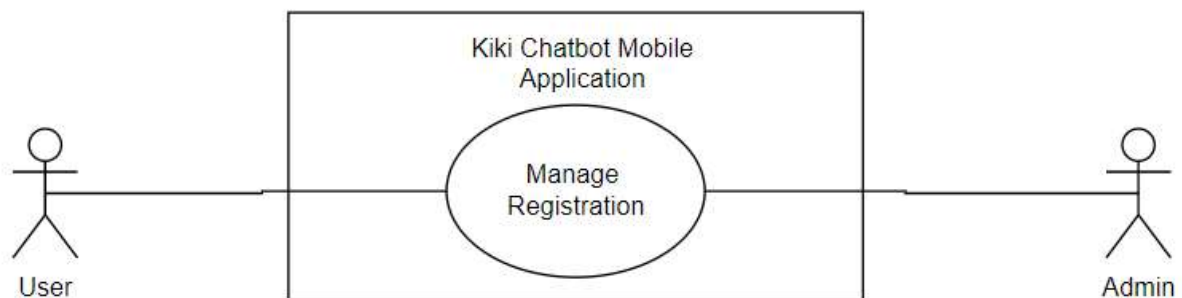


Figure 2. 2: Use Case for Manage Registration

Table 2. 2 Use Case Description for Manage Registration

Use Case ID	KIKI_UC_100
Use Case Name	Manage Registration
Brief Description	The use case allows the user to create account and allow the user and admin to login to the system.
Actor	User, Admin

Pre-Conditions	User do not log in to the system.
Basic Flow	<p>User</p> <ol style="list-style-type: none"> 1. The use case begins when the user opens the Kiki Chatbot Mobile Application. 2. The application displays the <<Main>> interface. 3. The application redirect to <<Login>> interface. 4. User enter email and password and clicks “Log in” button to login. [A1: No existing account.] [A2: Forgot password] 5. The application verifies the username and password. [E1: Invalid email and password] 6. The application redirect to <<User Homepage>> interface. 7. The use case end. <p>Admin</p> <ol style="list-style-type: none"> 1. The use case begins when the admin opens the Kiki Chatbot Mobile Application. 2. The application display the <<Main>> interface. 3. The application redirect to <<Login>> interface. 4. Admin enter email and password and clicks “Log in” button to login. [A2: Forgot password] 5. The application verifies the username and password. [E1: Invalid email and password] 6. The application redirect to <<Admin Homepage>>. 7. The use case end.
Alternative Flow	<p>A1: No existing account.</p> <ol style="list-style-type: none"> 1. User clicks the “Register” button. 2. The application redirects to <<Register>> interface. 3. User enter the required fields. 4. User clicks the “Register” button.

	<p>5. The application validates the registration data. [A3: Email taken]</p> <p>6. The application saves the data into the database.</p> <p>7. The application displays “Registered successfully” message.</p> <p>8. The use case continue to step 2 in User basic flow.</p> <p>A2: Forgot password</p> <p>1. User or admin clicks on “Forgot Password” button.</p> <p>2. The application redirects to <<Reset Password>> interface.</p> <p>3. User or admin enter the email that is in use and clicks “Send” button.</p> <p>4. The application sends a password reset link to the email.</p> <p>5. The application displays “A confirmation email to reset password is sent to your email. Please check your email.” message.</p> <p>6. User or admin clicks the link in the email and enter a new password.</p> <p>7. The use case continue to step 2 in User or Admin basic flow.</p> <p>A3: Email taken</p> <p>1. The application display a “Email already registered” message.</p> <p>2. The user or admin enter another email.</p> <p>3. The use case continue to step 4 in alternative flow A1.</p>
Exception Flow	<p>E1: Invalid email and password</p> <p>1. The application display “Invalid email or password” message.</p> <p>2. User or admin reenter email and password.</p> <p>3. The use case continue to step 5 in User or Admin basic flow.</p>
Post-Conditions	<p>The user registered their account successfully.</p> <p>The admin registered new admin account successfully.</p>

	The user's or admin's account are saved in the database.
Rules	-
Constraints	All the input fields in the registration interface must be fill before the user clicks on "Register" button.

2.1.2 Manage Chatting

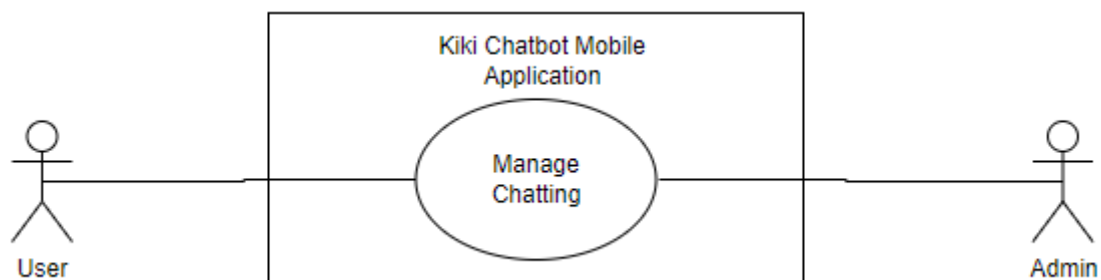


Figure 2. 3: Use Case for Manage Chatting

Table 2. 3 Use Case Description for Manage Chatting

Use Case ID	KIKI_UC_200
Use Case Name	Manage Chatting
Brief Description	This use case allows the user to chat using text with the chatbot Kiki.
Actor	User and Admin
Pre-Conditions	User logged into the system.
Basic Flow	<ol style="list-style-type: none"> 1. The use case start when the user clicks "Kiki Chatbot" button. 2. The application redirects to <<Chatbot>> interface. 3. The chatbot sends greetings message. 4. User reply the message by inputting text. [A1: Reply by options.]

	<p>5. User clicks “Send” button.</p> <p>6. The application analyses the message.</p> <p>7. The application retrieves the keyword from the message and search the script in Dialogflow.</p> <p>8. The application retrieves and reply with the closest answer found from the script.</p> <p>9. User close the application.</p> <p>10. The use case ends.</p>
Alternative Flow	<p>A1: Reply by options.</p> <p>1. User clicks the reply option provided above the text bar.</p> <p>2. The chosen reply send as user reply.</p> <p>3. The use case continues to step 6 in basic flow.</p>
Exception Flow	-
Post-Conditions	User get reply after sending message.
Rules	-
Constraints	-

2.1.3 Manage Self-test

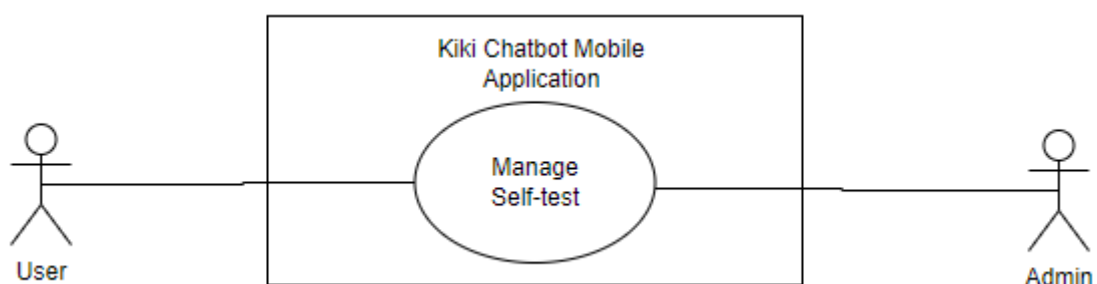


Figure 2. 4: Use Case for Manage Self-test

Table 2. 4 Use Case Description for Manage Self-test

Use Case ID	KIKI_UC_300
Use Case Name	Manage Self-test
Brief Description	This use case allows the user to do the OCD self- assessment to test the OCD severity.
Actor	User and Admin
Pre-Conditions	User logged into the system.
Basic Flow	<ol style="list-style-type: none"> 1. The use case begins when the user clicks “Test Now” button. 2. The application redirects to <<Self-assessment Menu>> interface. 3. User or admin clicks “New Self-assessment” button. [A1: Self-assessment History] 4. The application redirects to <<Self-assessment>> interface. 5. User choose answer for all of the question. 6. User clicks “Submit” button. 7. The application calculate the risk result. 8. The application save the risk result into the database. 9. The application display the risk result at <<Result>> interface. 10. The use case end.
Alternative Flow	<p>A1: Self-assessment History</p> <ol style="list-style-type: none"> 1. User clicks “Self-assessment History” button. 2. The application redirects to <<Self-assessment History>> interface. 3. The application retrieves risk result data from database. 4. The application display risk result list in table. 5. User close the interface. 6. The use case continue to step 10 in basic flow.
Exception Flow	-

Post-Conditions	The application displays the risk result of the user.
Rules	-
Constraints	-

2.1.4 Manage Information

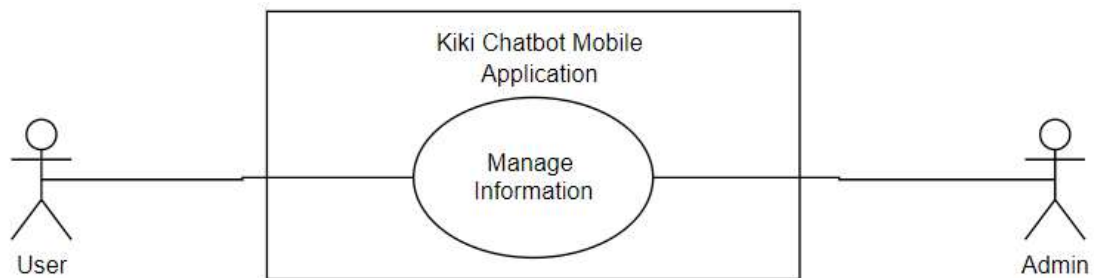


Figure 2. 5: Use Case for Manage Information

Table 2. 5: Use Case Description for Manage Information

Use Case ID	KIKI_UC_400
Use Case Name	Manage information
Brief Description	This use case allows the admin to add website and video information and allow user to view the website or video information.
Actor	User, Admin
Pre-Conditions	User or Admin logged into the system.
Basic Flow	<p>Admin</p> <ol style="list-style-type: none"> 1. The use case begins when admin clicks “Learn Now” button. 2. The application redirects to <<Manage Information>> interface. 3. The application retrieves all the information data from database. 4. The application display the information list in table.

	<p>5. Admin can:</p> <ol style="list-style-type: none"> a) Search information [A1: Search information] b) Add new information [A2: Add new information] c) Edit information [A3: Edit information] d) Delete information [A4: Delete information] <p>6. The use case end.</p> <p>User</p> <ol style="list-style-type: none"> 1. The use case starts when the user clicks “Learn Now” button. 2. The application redirect to <<Manage Information>> interface. 3. The application retrieves the information data from database. 4. The application displays the data in list of tables. 5. User clicks the particular information title from the list to view the information. [A1: Search information] 6. The application redirect to the particular website. 7. The use case end.
Alternative Flow	<p>A1: Search information</p> <ol style="list-style-type: none"> 1. User or admin enter title of information at the search bar to search. 2. The application retrieves all the data related from database. [A5: Search not found] 3. The application display the data. 4. The use case continues to step 5 in Admin basic flow or step 4 in User basic flow. <p>A2: Add new information</p> <ol style="list-style-type: none"> 1. Admin clicks “+” button. 2. The application redirects to <<Add New Information>> interface.

3. Admin enter all required field and clicks “Add” button.
4. The application saves the data into database.
5. The application display success message.
6. The application redirects to <<Manage Information>> interface.
7. The use case continue to step 5 in Admin basic flow.

A3: Edit information

1. Admin clicks “Edit” button.
2. The application retrieves the data of the selected reference from database.
3. The application displays the data in <<Edit Information>> interface.
4. Admin edit the fields that wish to make changes and clicks “Edit” button.
5. The application updates the data into database.
6. The application retrieves the updated data from database.
7. The application displays the data in <<Manage Information>> interface.
8. The use case continue to step 2 in Admin basic flow.

A4: Delete information

1. Admin clicks “Delete” button.
2. The application displays confirmation message.
3. Admin clicks “Confirm” button.
4. The application deletes the particular data from database.
5. The application display success notification.
6. The application redirect admin to <<Manage information>> interface.

	<p>7. The use case continues to step 5 in Admin basic flow.</p> <p>A5: Search not found</p> <p>1. The application could not query any related reference data from the database.</p> <p>2. The application display blank at <<Manage Information>> interface.</p> <p>3. The use case continues to step 1 in A1 alternative flow.</p>
Exception Flow	-
Post-Conditions	The information data is added, deleted or query from or to the database.
Rules	-
Constraints	-

2.1.5 Manage Diary

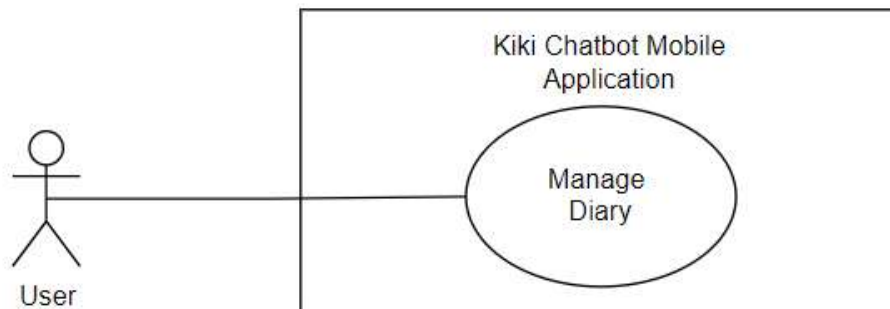


Figure 2. 6: Use Case for Manage Diary

Table 2. 6: Use Case Description for Manage Diary

Use Case ID	KIKI_UC_500
Use Case Name	Manage Diary

Brief Description	This use case allows user to record their diary.
Actor	User
Pre-Conditions	User logged into the system.
Basic Flow	<ol style="list-style-type: none"> 1. The use case begins when the user clicks “Diary” button. 2. The application redirects to <<Diary>> interface. 3. The application retrieve data from database and show all the diary. [A1: Add Diary] [A2: Delete Diary] [A3: Update Diary] 4. The use case end.
Alternative Flow	<p>A1: Add Diary</p> <ol style="list-style-type: none"> 1. User clicks “+” button on particular diary to view. 2. The application redirects to <<Add New Diary>> interface. 3. The application generate date for the diary. 4. User write the diary and click “Save” button. 5. The application saves the data to the database. 6. The application display a successful save message. 6. User click back button. 7. The use case continue to step 2 in basic flow. <p>A2: Delete Diary</p> <ol style="list-style-type: none"> 1. User clicks “Delete” button on particular diary to delete. 2. The application display a confirmation message. 3. User clicks “Confirm” button. 4. The application remove the data from database. 5. The use case continues to step 2 in basic flow. <p>A3: Update Diary</p> <ol style="list-style-type: none"> 1. User clicks “Update” button on particular diary to view.

	<ol style="list-style-type: none"> 2. The application redirect to <<Diary Details>> interface. 3. The application retrieves diary data for the particular diary. 4. The application display the data. 5. User update the field and close the interface. 6. The use case continues to step 2 in basic flow.
Exception Flow	-
Post-Conditions	The diary data is saved, updated or deleted to the database.
Rules	-
Constraints	-

2.1.6 Manage Profile

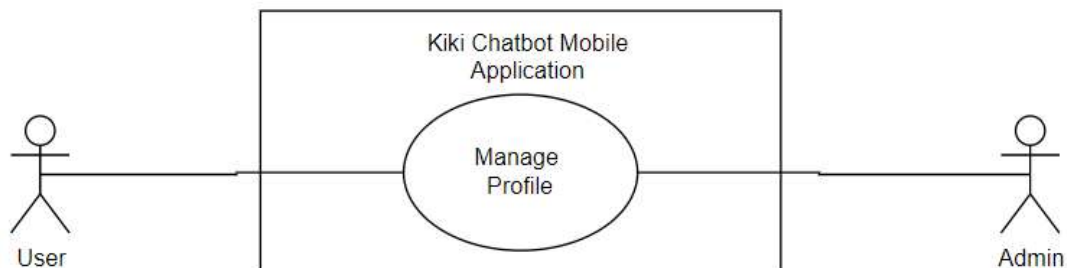


Figure 2. 7: Use Case for Manage Profile

Table 2. 7: Use Case Description for Manage Profile

Use Case ID	KIKI_UC_600
Use Case Name	Manage Profile
Brief Description	This use case allows the user and admin to view, edit their profile and delete their account. The user can send feedback and the admin can view the feedback.
Actor	User, Admin

Pre-Conditions	User and Admin logged into the system.
Basic Flow	<p>User</p> <ol style="list-style-type: none"> 1. The use case begins when the user redirects to the <<Profile>> interface by clicking on the “Profile” icon. 2. The application redirects to <<Profile>> interface. 3. User can: <ol style="list-style-type: none"> a) edit profile [A1: Edit profile] b) log out [A2: Logout] c) change profile picture [A3: Change profile picture] d) make feedback [A4: Make Feedback] 4. The use case end. <p>Admin</p> <ol style="list-style-type: none"> 1. The use case begins when the user redirects to the <<Profile>> interface by clicking on the “Profile” icon. [A5: View deedback] 2. The application redirects to <<Profile>> interface. 3. User can: <ol style="list-style-type: none"> a) edit profile [A1: Edit profile] b) log out [A2: Log out] c) change profile picture [A3: Change profile picture] 4. The use case end.
Alternative Flow	<p>A1: Edit profile</p> <ol style="list-style-type: none"> 1. User or admin clicks “Edit Profile” button. 2. The application retrieves the profile data from database. 3. The application display the data for particular profile at <<Edit Profile>> interface.

4. User or admin edit the field that wish to make changes.
5. User or admin click “Edit” button.
7. The application validates the data. [**A6: Invalid field.**]
8. The application updates the data into the database.
9. The application displays success notification.
10. User or admin close the interface.
11. The use case continues to step 2 in User or Admin basic flow.

A2: Log out

1. User or admin clicks “Logout” button.
2. User or admin clicks “Confirm” button.
3. The application logs the account out.
4. The application redirect to <<Main>> interface.
5. The use case end.

A3: Change profile picture

1. User or admin click on the profile picture.
2. User or admin select profile picture from phone storage.
3. User or admin click on the “crop” button on the top right at the menu.
4. The application save the profile picture to database.
5. The application redirect to <<Profile>> interface and retrieve the new profile picture to display.
6. The use case continue to step 2 in user or Admin Basic Flow.

A4: Make feedback

1. User clicks on “Feedback” button.

2. The application redirects to <<Feedback>> interface.
3. User enter all the required fields and clicks “Submit” button.
4. The application save the data into the database.
5. The application display “Feedback successfully” message.
6. The use case continues to step 2 in User basic flow.

A5: View Feedback

1. Admin clicks on “Feedback” button.
2. The application retrieves feedback data from database.
3. The application display the feedback list in a table at <<Feedback>> interface. **[A7: Search feedback] [A8: View feedback history]**
4. Admin click on the feedback that he/she want to view.
5. The application redirects to <<Feedback Details>> interface,
6. The application retrieves particular feedback data from database.
7. The application display the feedback details. **[A9: Mark done]**
8. The use case continues to step 1 in Admin basic flow.

A6: Invalid field.

1. The application display an error message.
2. User or admin reenter the data.
2. The use case continues to step 5 in A1 alternative flow.

A7: Search feedback

1. Admin search for particular feedback.
2. The application retrieves the related feedback from database. **[A10: Search not found]**
3. The application display the related feedback list.

	<p>4. The use case continue to step 4 in A5 alternative flow.</p> <p>A8: View feedback history</p> <ol style="list-style-type: none"> 1. Admin click the “tick” icon on the top right menu bar. 2. The application retrieve feedback history and display in <<Feedback History>> interface. 3. Admin click on the feedback that he/she want to view. 5. The application redirects to <<Feedback Details>> interface. 6. The application retrieves particular feedback data from database. 7. The application display the feedback details. 8. User click on the back button. 9. The use case continue to step 2 in A5 alternative flow. <p>A9: Mark done</p> <ol style="list-style-type: none"> 1. Admin click the “Mark done” button on the bottom. 2. The application save the feedback to feedback history. 3. The use case continue to step 2 in A5 alternative flow. <p>A10: Search not found</p> <ol style="list-style-type: none"> 1. The application could not query any related name of feedback from the database. 2. The application display “No Result Found” at <<Feedback>> interface. 3. The use case continues to step 1 in A7 alternative flow.
Exception Flow	-
Post-Conditions	<p>The profile is updated successfully.</p> <p>The account is deleted from the application.</p> <p>The feedback added into the application.</p>

Rules	-
Constraints	-

2.2 SEQUENCE DIAGRAM

2.2.1 Manage Registration

Figure 2.8 show the basic flow of the manage registration module. After showing the <<Main>> interface, the application will redirect to the <<Login>> interface from the <<Main>> interface. There is an alternative for user which is No Existing Account. After that, the actor enters the email and password and click “Login” button, The login data will send to the controller and verify by the database. Here is the alternative flow Forgot Password. After the data is being verified successfully, then the application will redirect to the actor homepage interface. There is an exception flow here which is the Invalid Email and Password. For the admin, there is 3 alternative flow which are search admin, add new admin and remove admin.

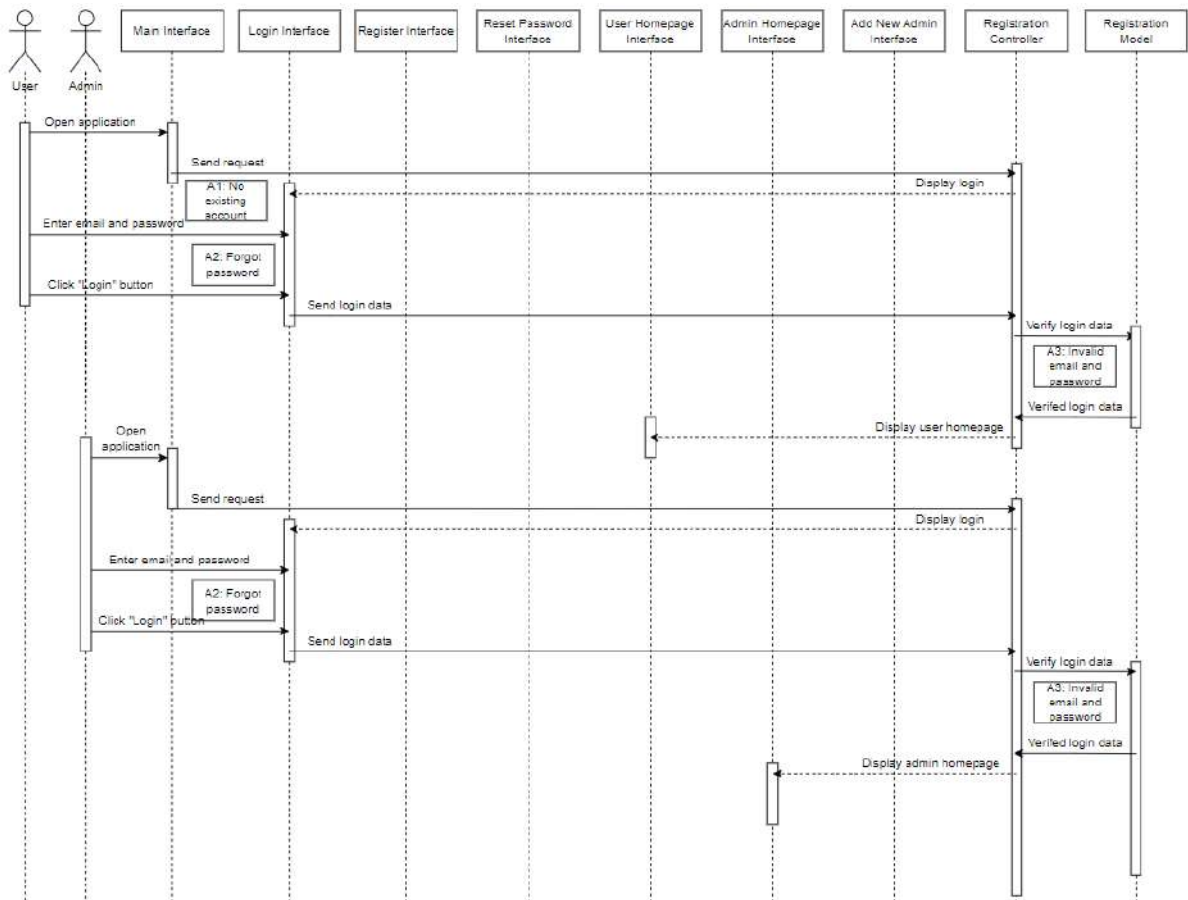


Figure 2. 8 Manage Registration Basic Flow Sequence Diagram

Figure 2.9 shows the alternative flow of this module. For the A1: No existing account, the user clicks the “Register” button and the application will redirect to <<Register>> interface and follow the register process. There is an alternative flow here which is Email Taken.

For A2: Forgot Password, the user and admin click “Forgot password” button and the application redirect to <<Reset Password>> interface and follow the reset password process.

For A3: Email taken, the user or admin required to enter another email.

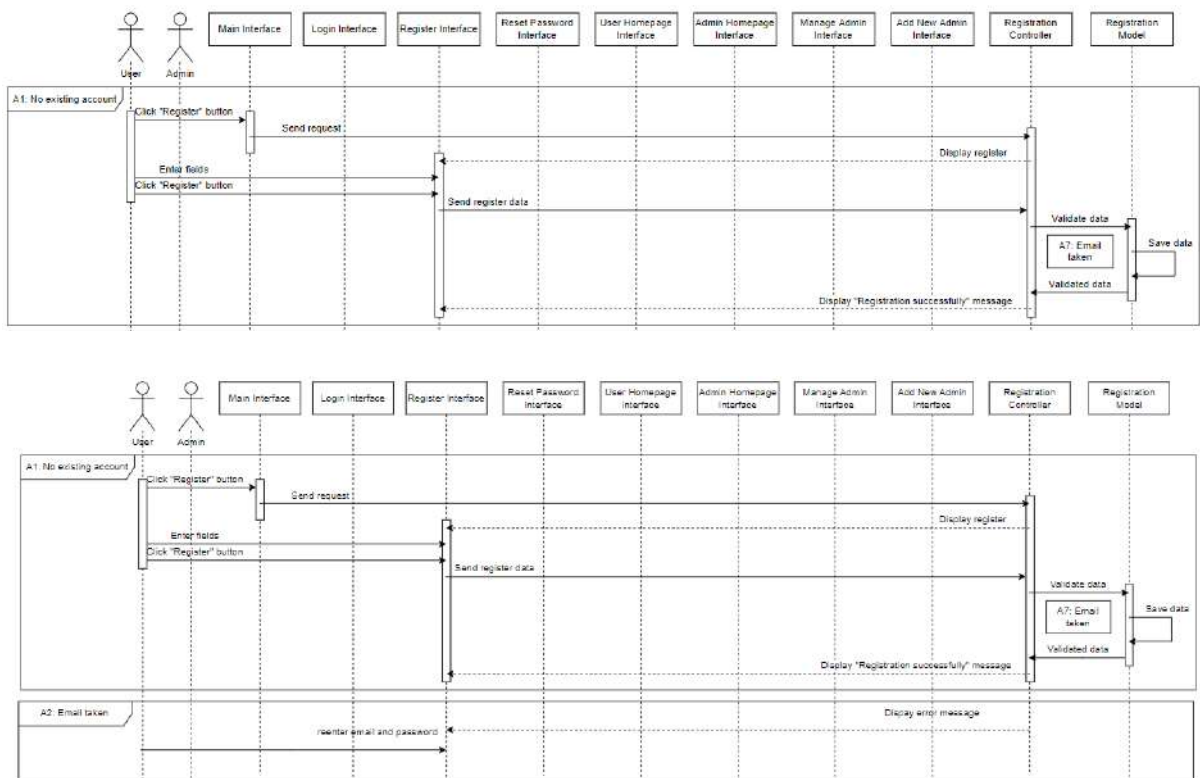


Figure 2. 9 Manage Registration Alternative Flow Sequence Diagram

Figure 2.10 show the exception flow of this module. When the user or admin enter the invalid email or password, the application will display an error message and the user or admin need to reenter the email and password.

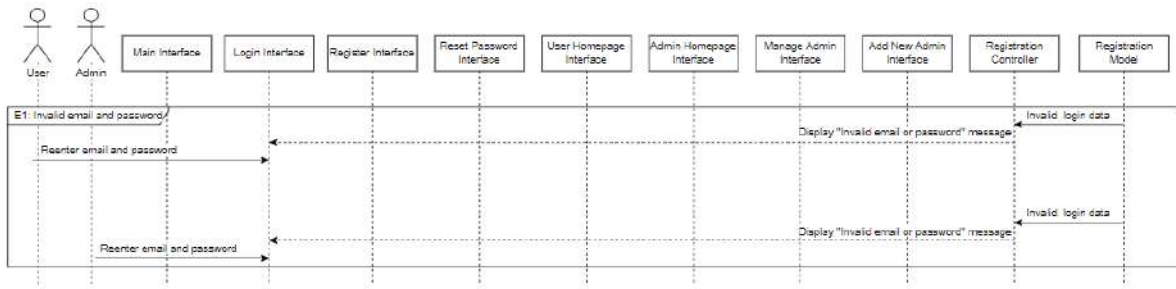


Figure 2. 10 Manage Registration Exception Flow Sequence Diagram

2.2.2 Manage Chatting

Figure 2.11 show the basic flow of the manage chatting module. After the user click the “Kiki Chatbot” button, the application will redirect to the <<Chatbot>> interface from the <<User Homepage>> interface. Then the chatbot will display greetings message, user can enter text to reply message or Reply by Options (alternative A1) and click “Send” button. After that, the message will be sent to the Controller and the response will be search to reply to the <<Chatbot>> interface.

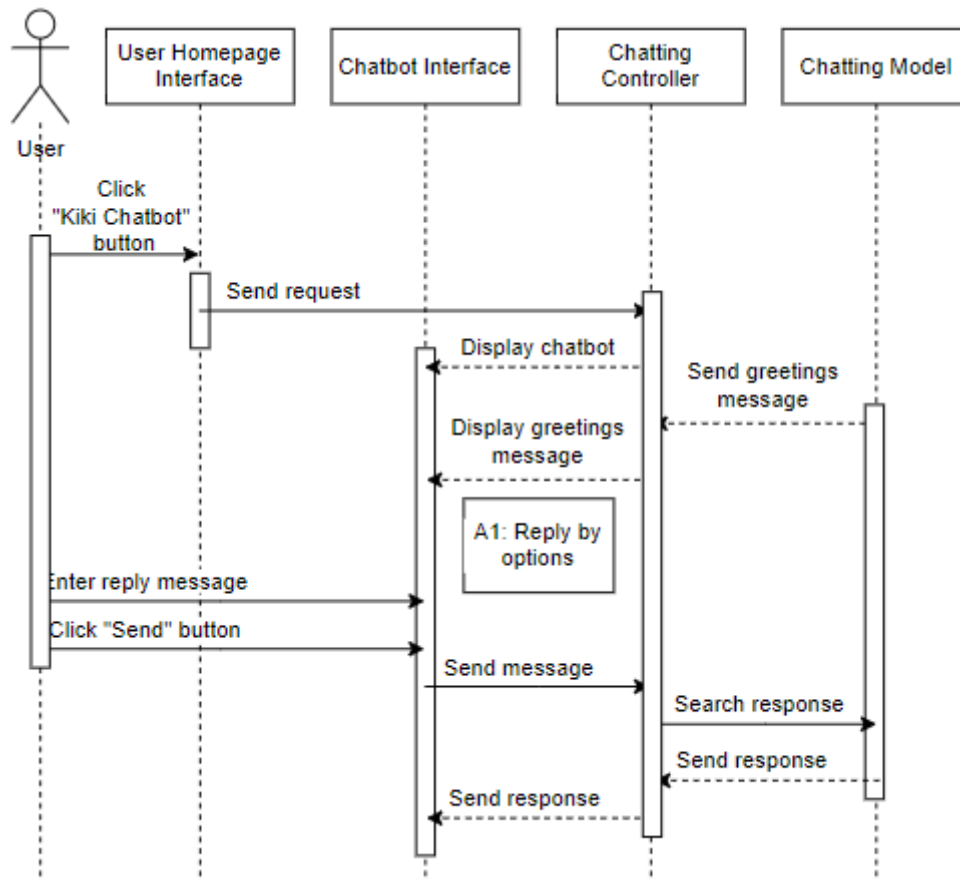


Figure 2. 11 Manage Chatting Basic Flow Sequence Diagram

Figure 2.12 show the alternative flow of this module. For A1: Reply by options, the user choose options above the textbox to reply.

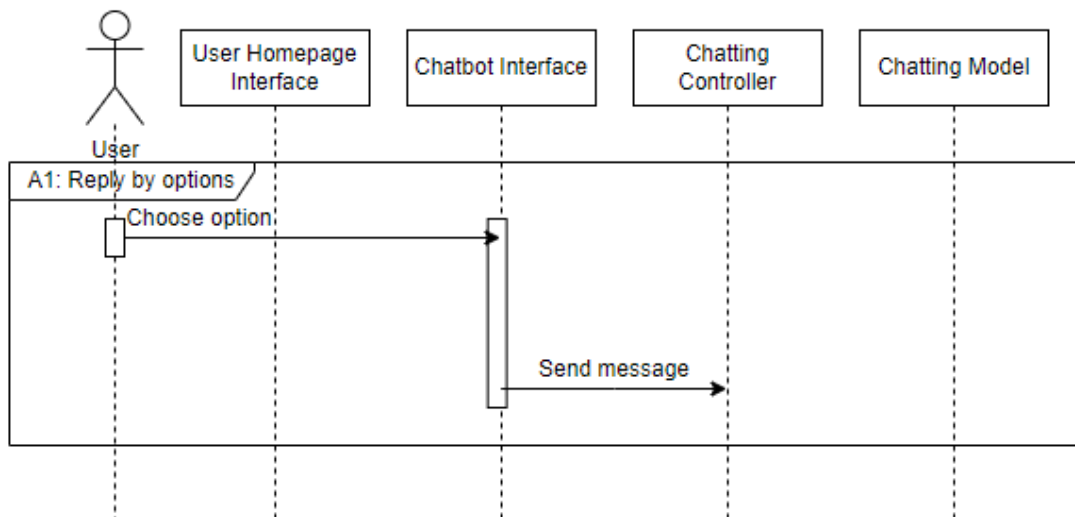


Figure 2. 12 Manage Chatting Alternative Flow Sequence Diagram

2.2.3 Manage Self-test

Figure 2.13 show the basic flow of the manage self-test module. After the user and admin click the “Test Now” button, the application will redirect to the <<Self-assessment Menu>> interface from the <<User Homepage>> interface. Then the user and admin can click the “New Self-assessment” button. Here is the alternative flow Self-assessment History. Then the application will direct to the <<Self -assessment>> interface for the user and admin to answer the question and click “Submit” button. The, the data will be sent to the controller and the result will be calculated and saved in the database and the result will be display at the <<Result>> interface.

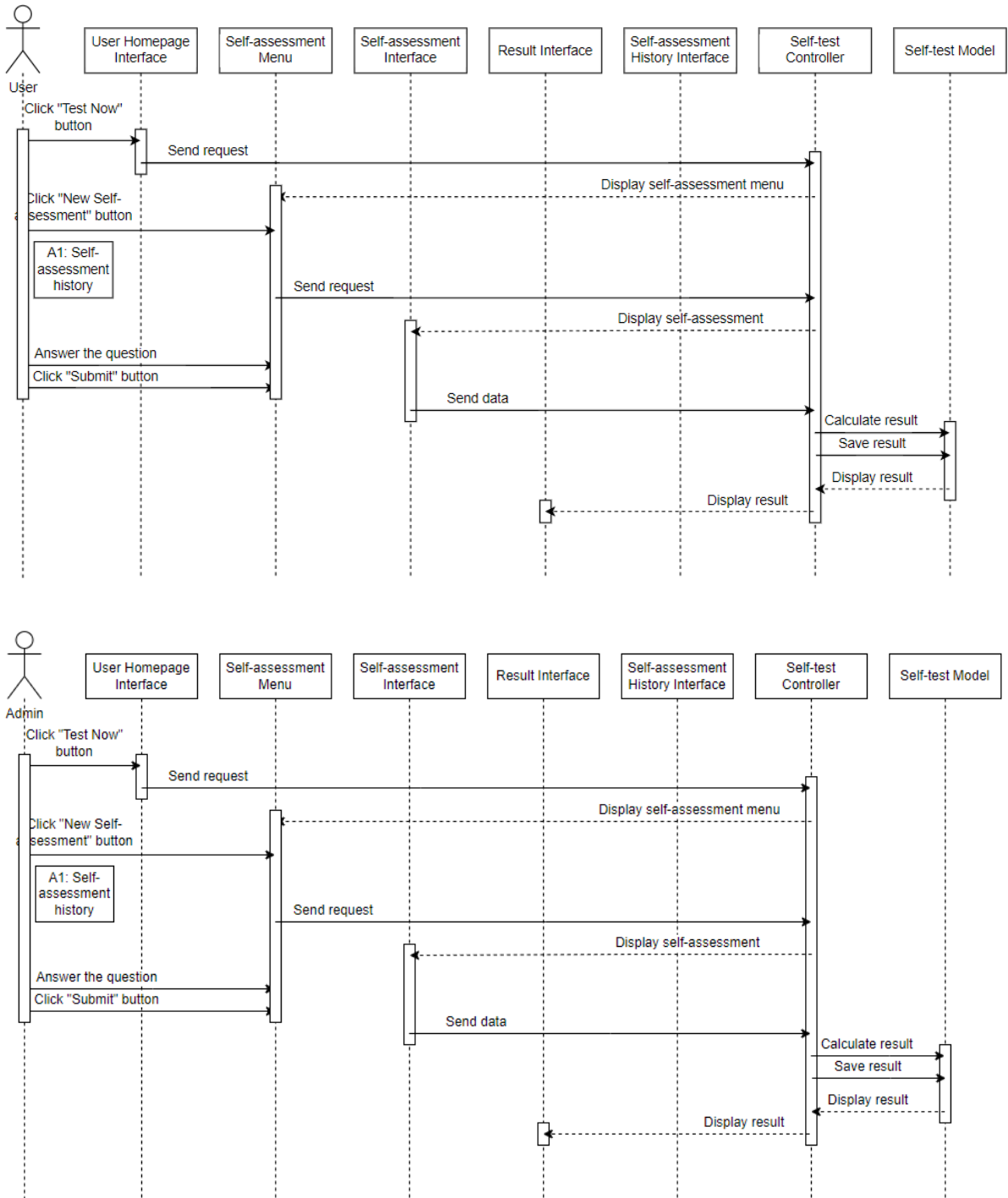


Figure 2. 13 Manage Self-test Basic Flow Sequence Diagram

Figure 2.14 show the alternative flow of this module. A1: Self-assessment history, the user click “Self-assessment history” button and the controller will retrieve the data from the database and display at the << Self-assessment History >> interface.

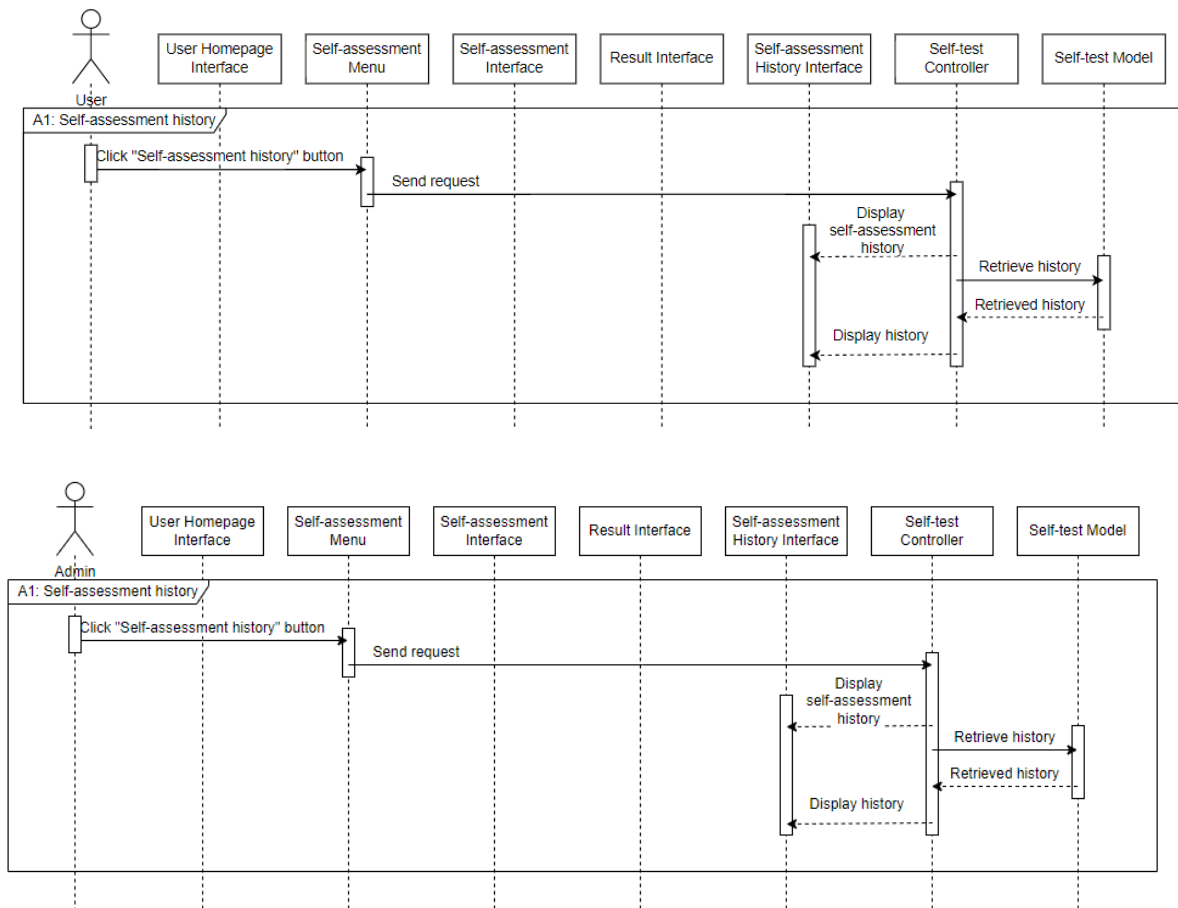


Figure 2. 14 Manage Self-test Alternative Flow Sequence Diagram

2.2.4 Manage Information

Figure 2.15 show the basic flow of the manage information module. After the user click the “Learn Now” button, the application will redirect to the <<Manage Information>> interface and display the information list from the <<User Homepage>> interface. Then the user can click the “View” button to view the website. There is a alternative flow where the user can search for specific information by title.

For the admin, after clicking the “Learn Now” button, the application will redirect to the <<Manage Information>> interface and display the information list from the <<User Homepage>> interface. Then here is the alternative flow, which are Search Information, Add New Information, Edit Information and Delete Information.

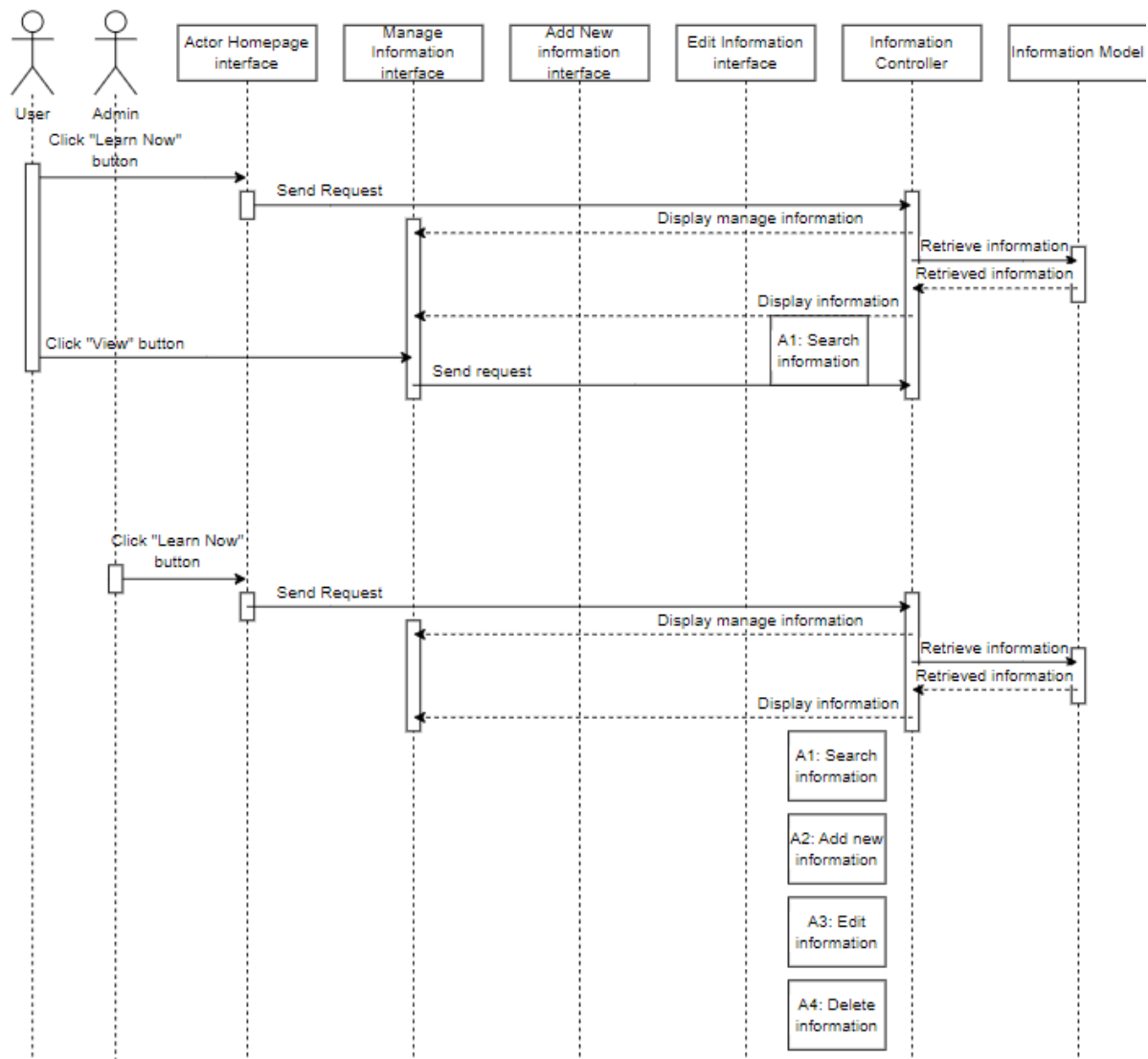


Figure 2. 15 Manage information Basic Flow Sequence Diagram

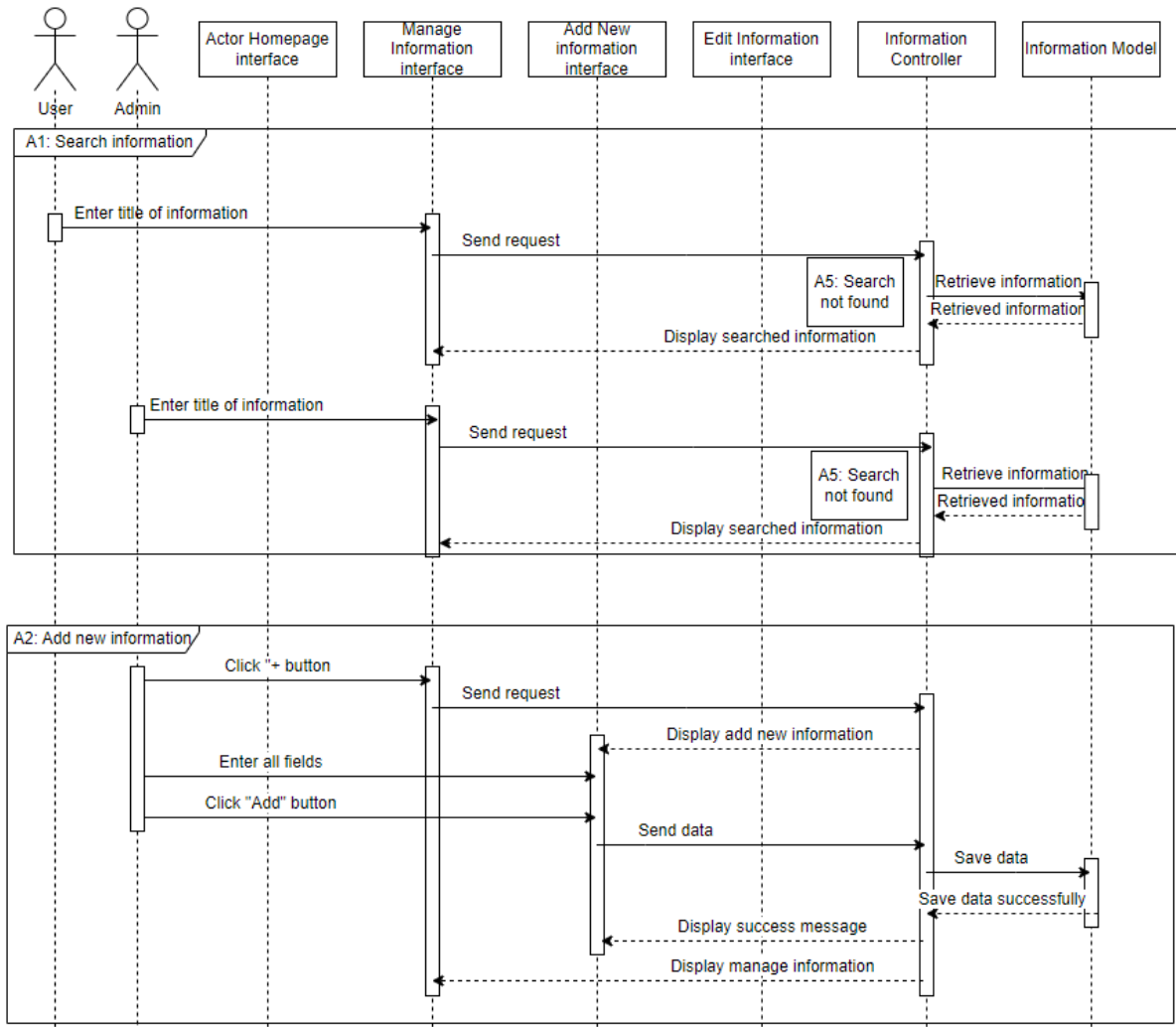
Figure 2.16 shows the alternative flow for this module. For A1: Search information, the user and admin enter title of the information at the search bar and the application will retrieve and display the data from the database. Here is the alternative flow Search not Found.

For A2: Add new information, the admin click “Add New Information” button and fill in the field and click the “Add” button. Then the data will saved in the database.

For A3: Edit information, the admin click “Edit information” button, edit the field then click “Edit” button. The edited data will update to the database.

For A4: Delete information, the admin click “Delete” button and “Confirm” button from the confirmation prompt out. Then, the data will be deleted from the database.

For A5: Search not found, the application will displayed “No Result Found”.



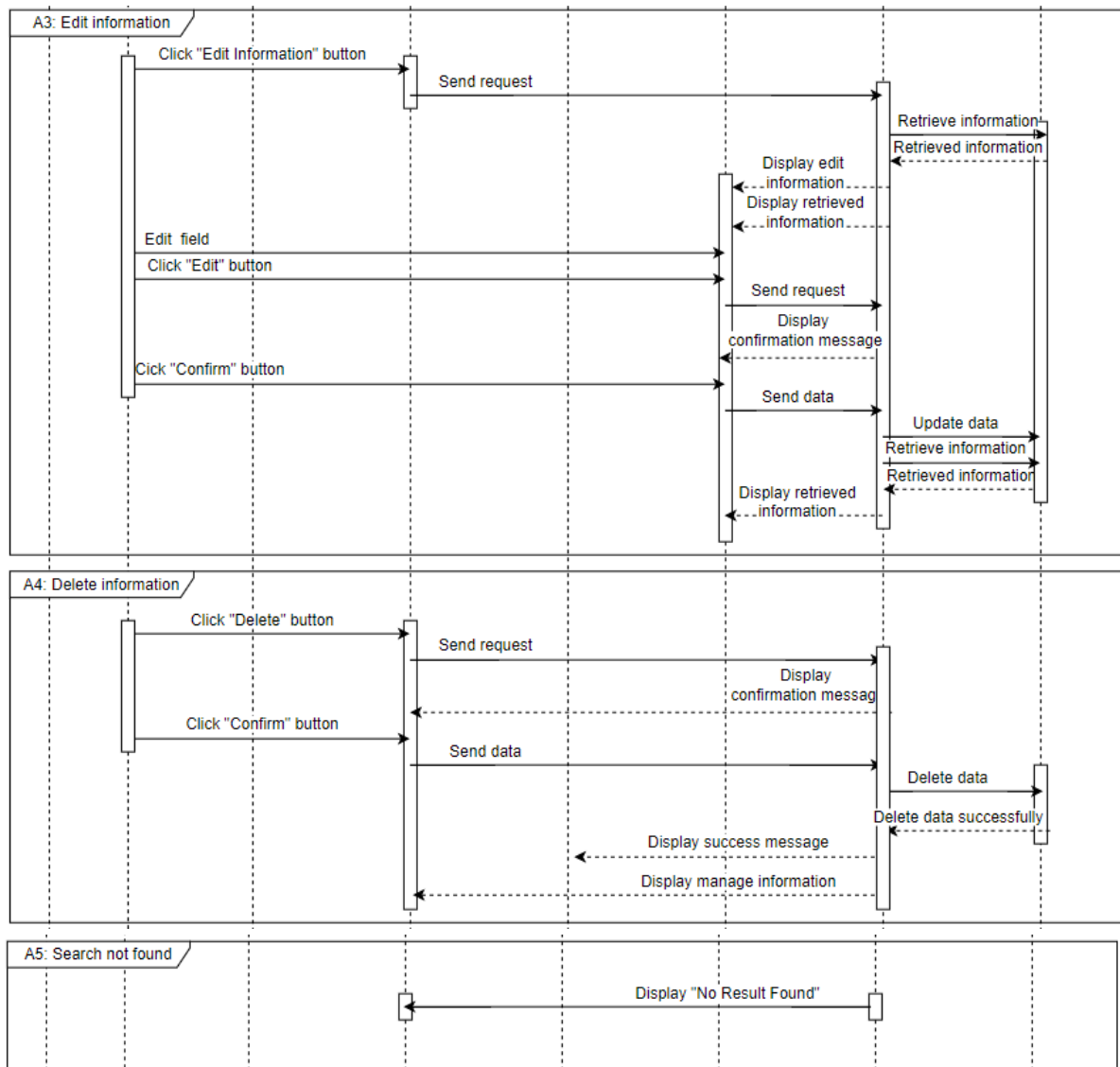


Figure 2. 16 Manage Information Alternative Flow Sequence Diagram

2.2.5 Manage Diary

Figure 2.17 show the basic flow of the manage diary module. After the user click the “Diary” button, the application will redirect to the <<Diary>> interface. All the diary history will be shown here. Here is the alternative flow, Add Diary, Delete Diary and Update Diary.

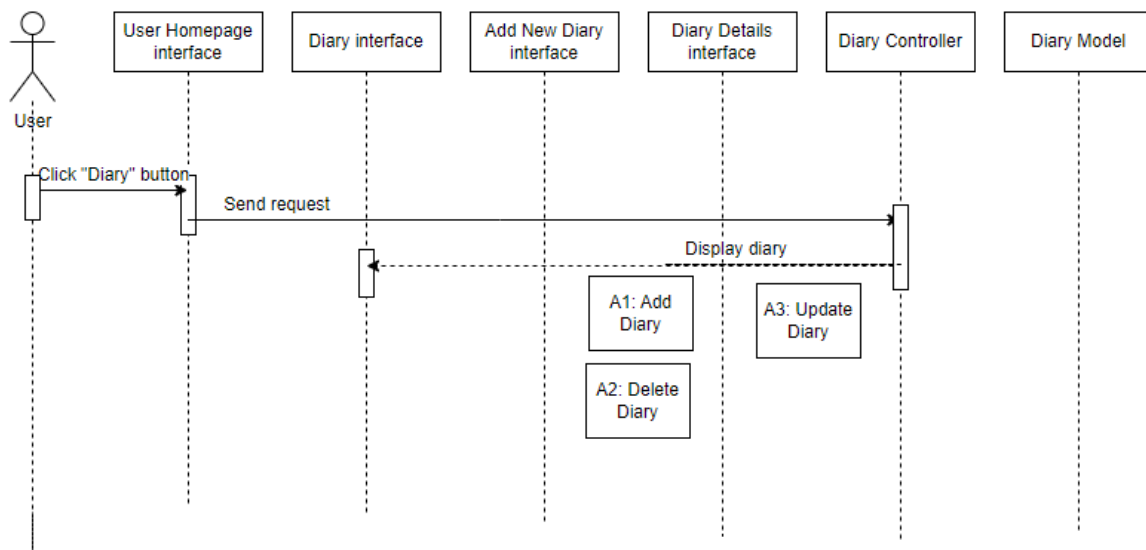


Figure 2. 17 Manage Diary Basic Flow Sequence Diagram

Figure 2.18 show the alternative of this module. For A1: Add diary, the user clicks “+” and redirect to <<Add Diary>> interface to add new diary.

For A2: Delete Diary, the user click “Delete” button and “Confirm” button on the confirmation prompt out. Then , the data will be deleted from the database.

For A3: Update Diary, the user edit the field and click “Update” button.

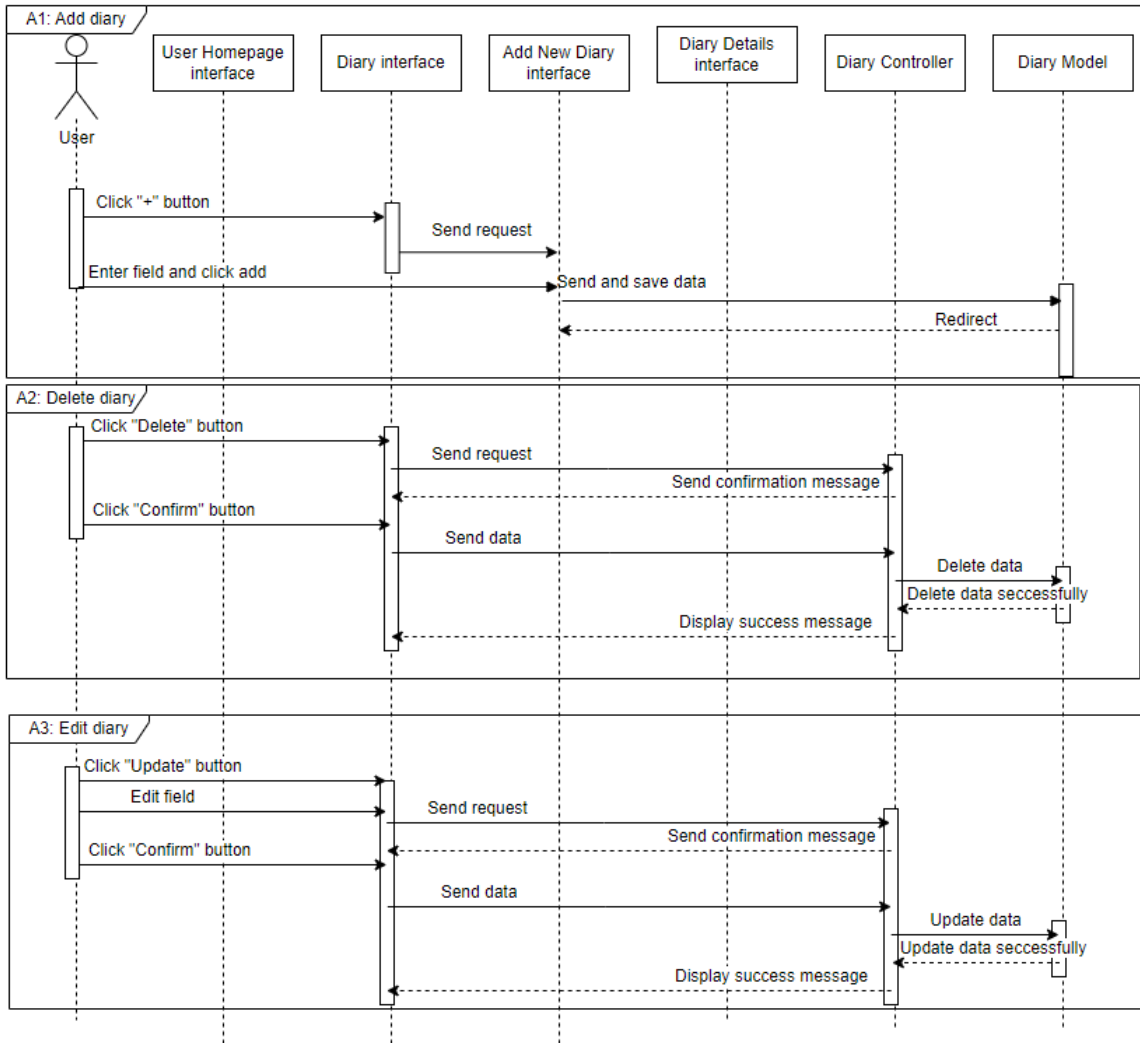


Figure 2. 18 Manage Diary Alternative Flow Sequence Diagram

2.2.6 Manage Profile

Figure 2.19 show the basic flow of the manage profile module. After the user click the “Profile” icon, the application will redirect to the <<Profile>> interface. Here is the alternative flow for admin which is View Feedback. Then, at the profile interface, there are alternative flow for both the user and admin which are Edit Profile, Logout and Change Profile Picture. And there is one more alternative for the user which is Make Feedback.

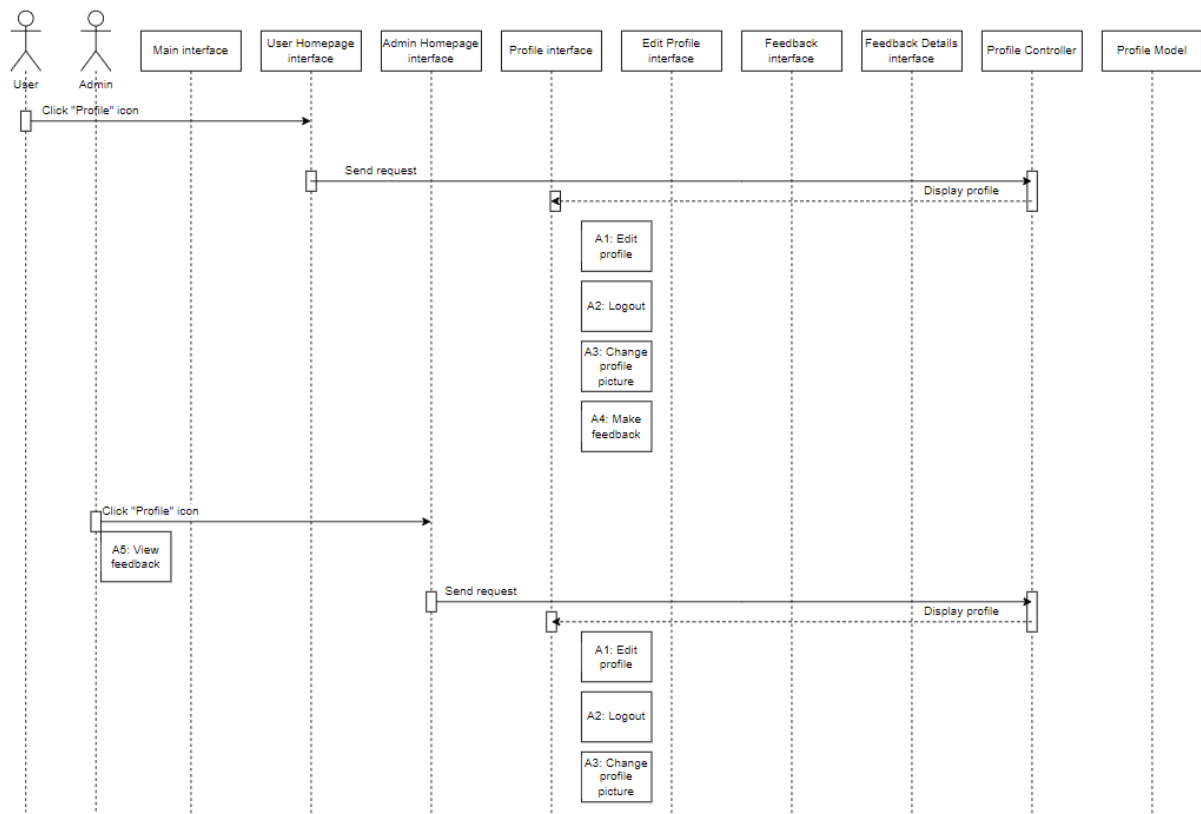


Figure 2. 19 Manage Profile Basic Flow Sequence Diagram

Figure 2.20 show the alternative flow of this module. For A1:Edit profile, these user and admin clicks “Edit Profile” button and follow the steps to edit the profile. Here is the alternative Invalid Field. The edited data will be saved in the database.

For A2: Logout, the user and admin click the “Logout” button at the <<Profile>> interface to logout.

For A3: Change profile picture, the user and admin click the profile picture to change the new profile picture.

For A4: Make feedback, the user clicks “Make Feedback” button at the <<Profile>> interface and fills the feedback form and clicks the “Submit” button. The data will be saved in the database.

For A5: View feedback, the admin clicks “Feedback” icon and then the “View” button, the controller will retrieve the data and display the data. Here is the alternative flow Search Feedback.

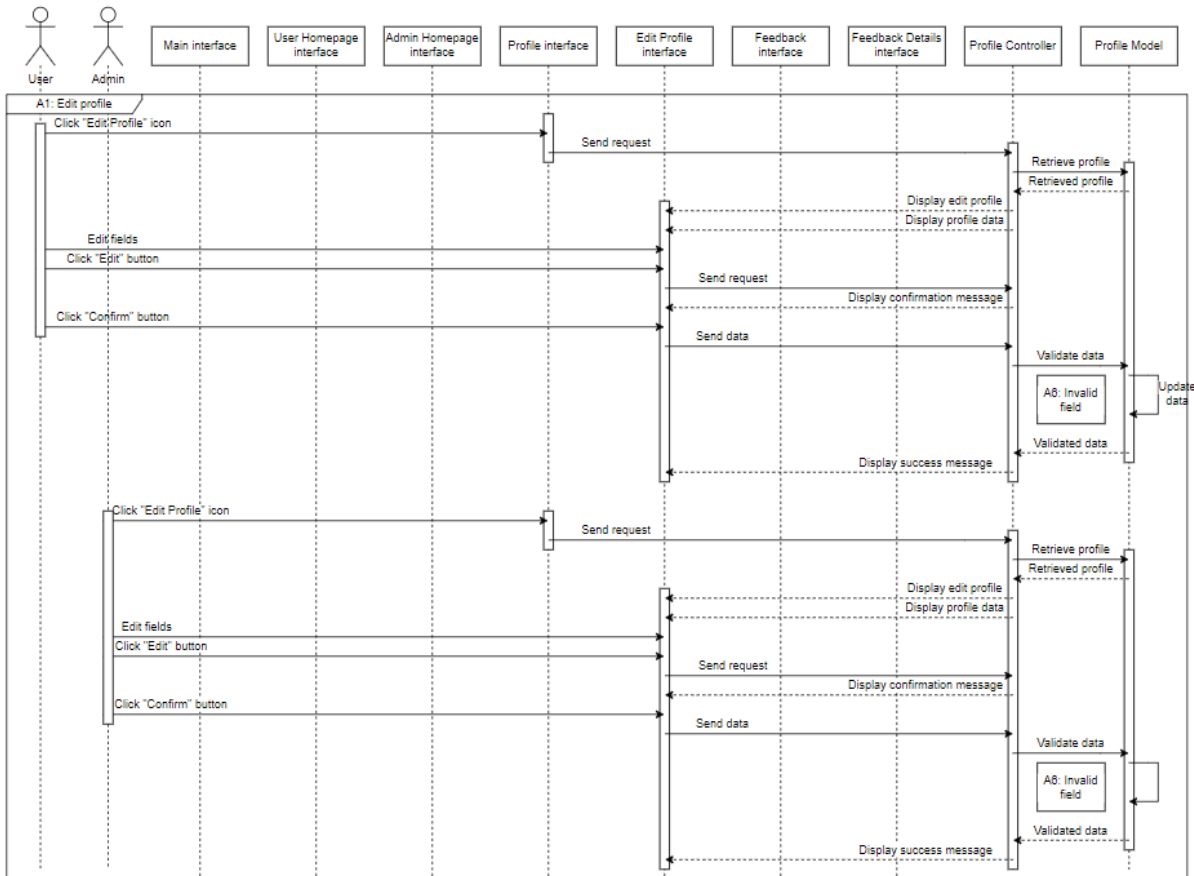
For A6: Invalid field, an error message will be displayed and the user or admin needs to reenter the field.

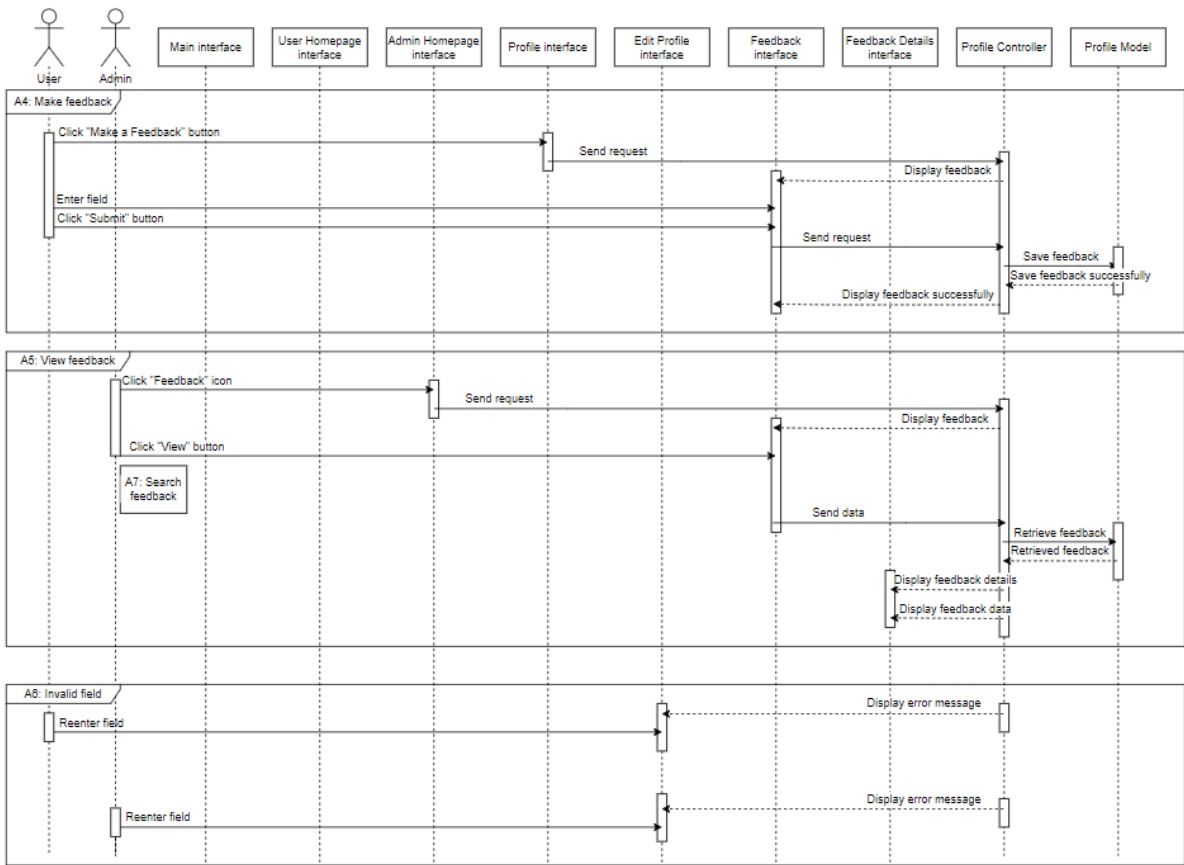
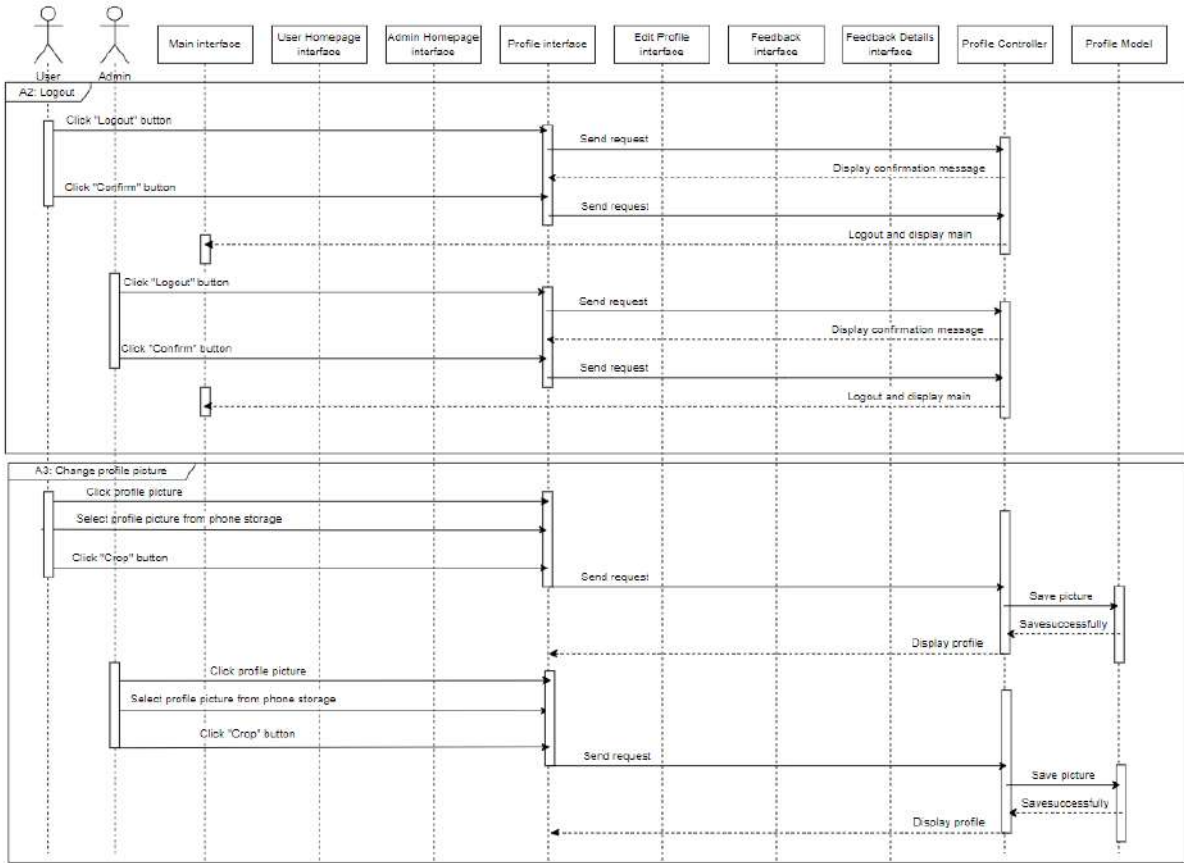
For A7: Search feedback, the admin enter the feedback at the search bar to search and the controller will retrieved and display the related data. Here is the alternative flow, Search Not Found.

For A8: View feedback history, the admin redirect to view feedback history.

For A9: Mark done, the admin mark done when the feedback is done.

For A10: Search not found, the controller will display the “Search Not Found” message.





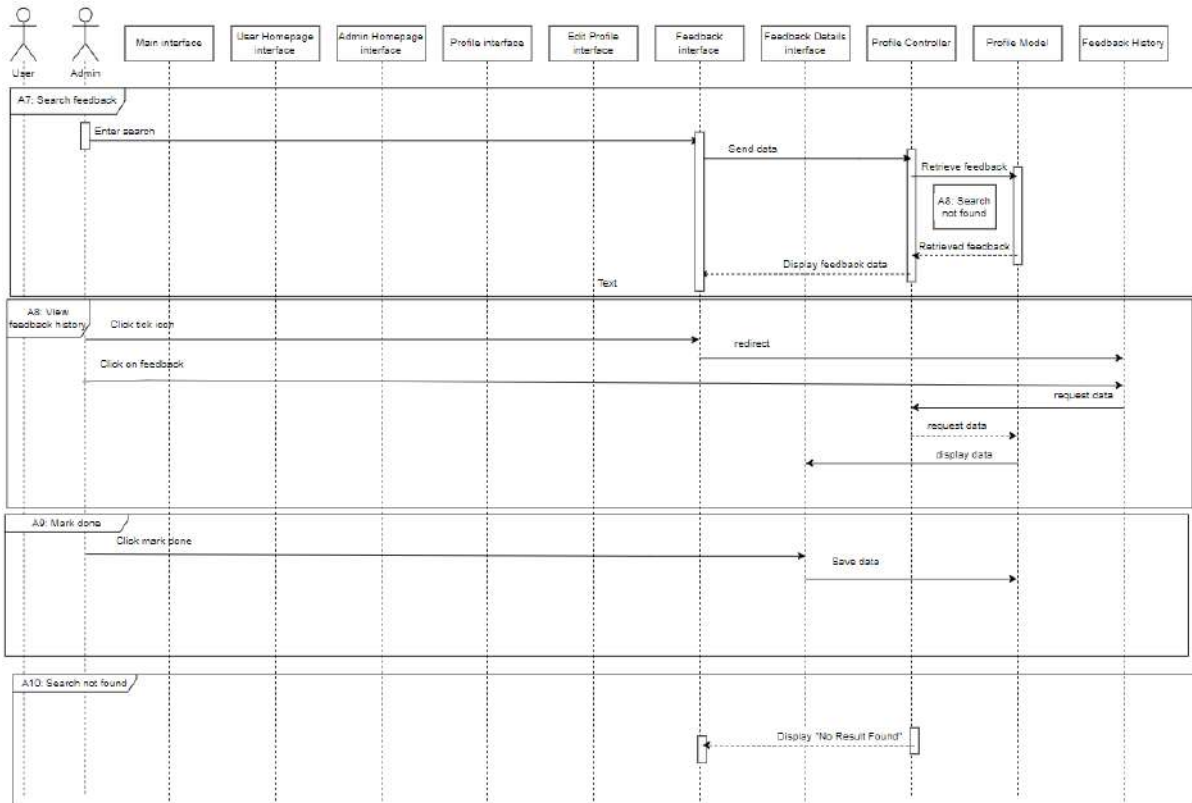


Figure 2. 20 Manage Profile Alternative Flow Sequence Diagram

CHAPTER 3

3.1 INTERFACE DESIGN

3.1.1 Manage Registration

Figure 3.1 show the main interface when the user opened the Kiki Mobile Chatbot Application. The interface will be automatically directed to the Login Interface that show in Figure 3.2.



Figure 3. 1 Main Interface

Figure 3.2 show the Login Interface. In the login interface, the actor can login to the function by filling correct email and password and click on <<LOGIN>> button. When clicking on the “Forgot Password” text, the interface will redirect to Reset Password Interface while clicking on the “Create New Account?” text, the interface will redirect to register interface.

The image shows a login interface with the following elements:

- A title "Login" centered at the top.
- An input field for "Email" with an envelope icon on the left.
- An input field for "Password" with a toggle icon (an eye) on the right.
- A link "Forgot Password?" located below the password field.
- A blue rounded rectangular button labeled "LOGIN" centered below the fields.
- A link "Create New Account?" located below the login button.

Figure 3. 2 Login Interface

Figure 3.3 show the alert from the login interface when the user enters an invalid email format or blank input field. When the actor blanks the email field and the password field or enter an invalid email format and click <<LOGIN>>, an alert will display in the respectively field.

Figure 3.4 show the interface with pop up when the actors enter invalid email or password. When the actor enters incorrect email and password to login to the system, a pop up will show to inform the actor failed to login. If the email and password are valid and correct, then the actor will login successfully and redirect to the dashboard.

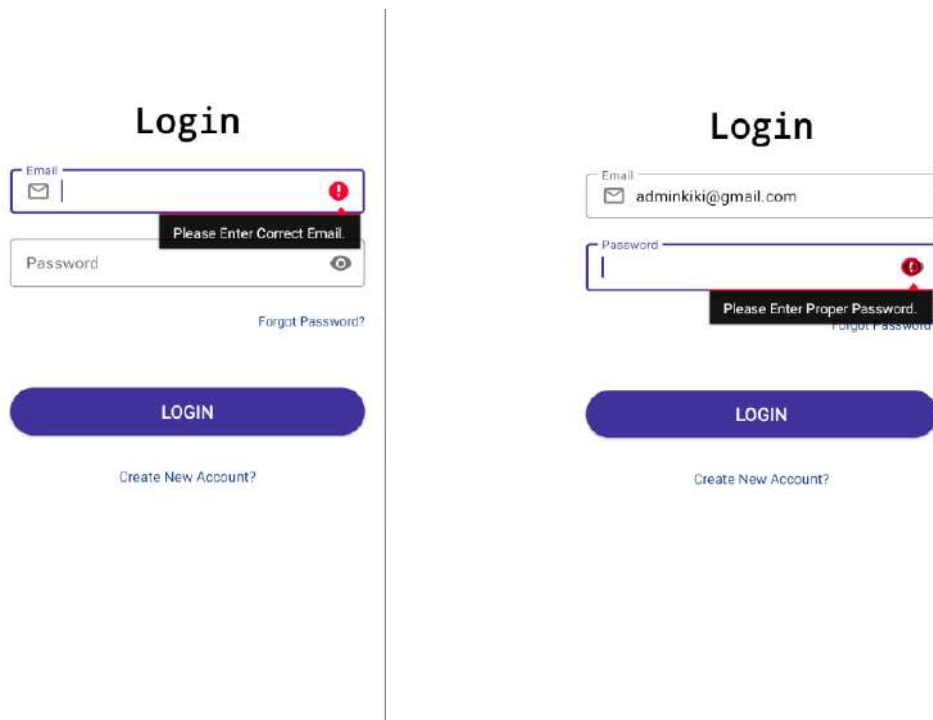


Figure 3. 3 Alert Error on Login Interface

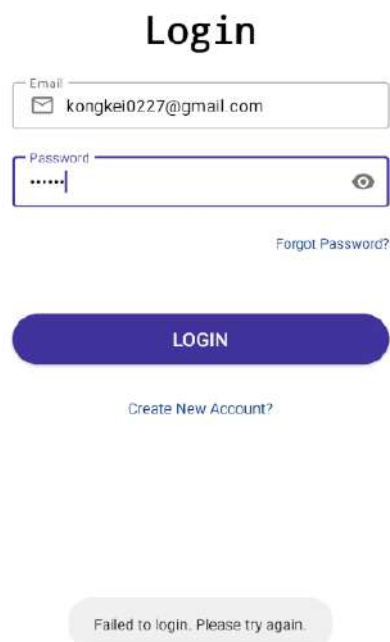


Figure 3. 4 Pop Up Notification for Invalid Email or Password in Login Interface

Figure 3.5 show the Reset Password Interface. The user can fill their registered email and click on <<RESET>> button to get their reset password email to reset their password.

Reset Password

Enter Your Email To Receive The Reset Link

[BACK TO LOGIN](#)

Figure 3. 5 Reset Password Interface

When the actor enters a blank or invalid email format and click on <<RESET>> at the Reset Password Interface, an alert will show beside the field as shown in Figure 3.6. Clicking on the “BACK TO LOGIN” text will redirect the user back to the Login Interface.

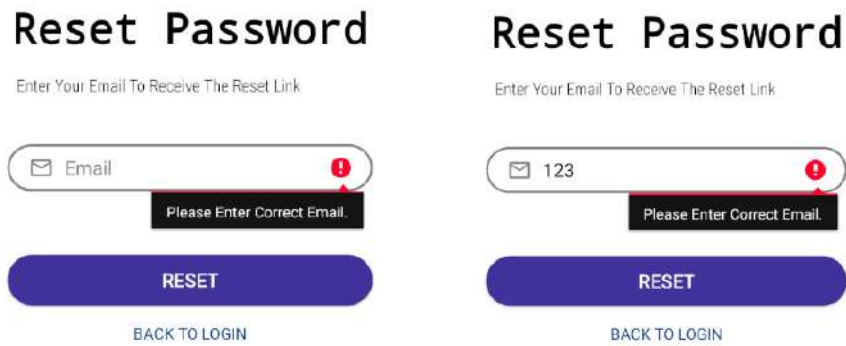
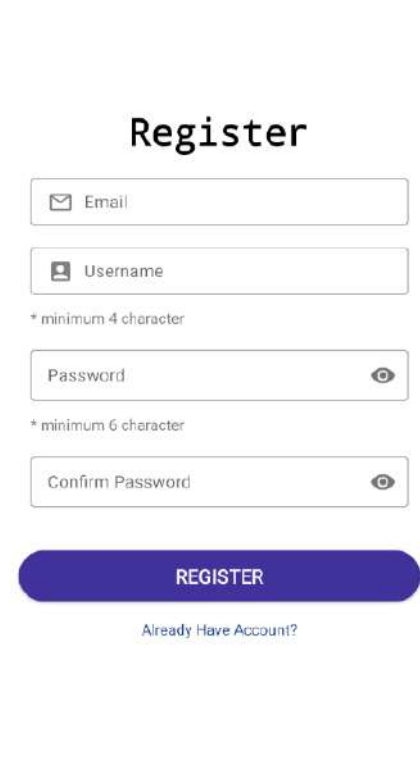


Figure 3. 6 Alert in Rest Password Interface

Figure 3.7 show the Register Interface. The user can register into the system by filling all the field and click on <<REGISTER>> button.



The image shows a registration form titled "Register". It contains four input fields: "Email" with an envelope icon, "Username" with a person icon, "Password" with a lock icon, and "Confirm Password" with a lock icon. Below the "Username" field is a note: "* minimum 4 character". Below the "Password" field is a note: "* minimum 6 character". At the bottom of the form is a blue "REGISTER" button and a link "Already Have Account?".

Figure 3. 7 Register Interface

Figure 3.8 show the alert message when the actor enters a blank or invalid email format, blank username, blank password and confirm password or password or confirm password is not matching and click on <<REGISTER>>. If all the fields are correct and valid, then the user will be register into the database. Clicking on the “Already Have Account” will redirect the actor to the Login Interface.

The figure displays four screenshots of a 'Register' form, arranged in a 2x2 grid, illustrating various validation errors. Each form has the following fields: Email, Username, Password, and Confirm Password. Below the fields are a 'REGISTER' button and a link 'Already Have Account?'. The error messages are as follows:

- Top Left:** The Email field contains an empty input. A red exclamation mark icon is present. A black alert box displays the message: "Please Enter Correct Email."
- Top Right:** The Username field contains the text "hoho". A red exclamation mark icon is present. A black alert box displays the message: "Please Enter Correct Email."
- Bottom Left:** The Confirm Password field contains the text ".....". A red exclamation mark icon is present. A black alert box displays the message: "Password Not Match!"
- Bottom Right:** The Username field contains an empty input. A red exclamation mark icon is present. A black alert box displays the message: "Please enter Proper Username"

Figure 3. 8 Alert in Register Interface

Figure 3.9 show the dashboard interface when the admin successfully login into the application while Figure 3.10 show the dashboard interface for the user.

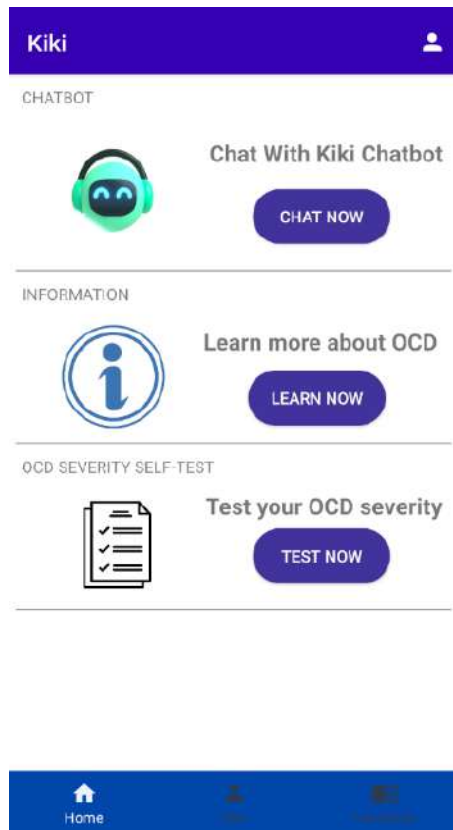


Figure 3. 9 Dashboard Admin Interface

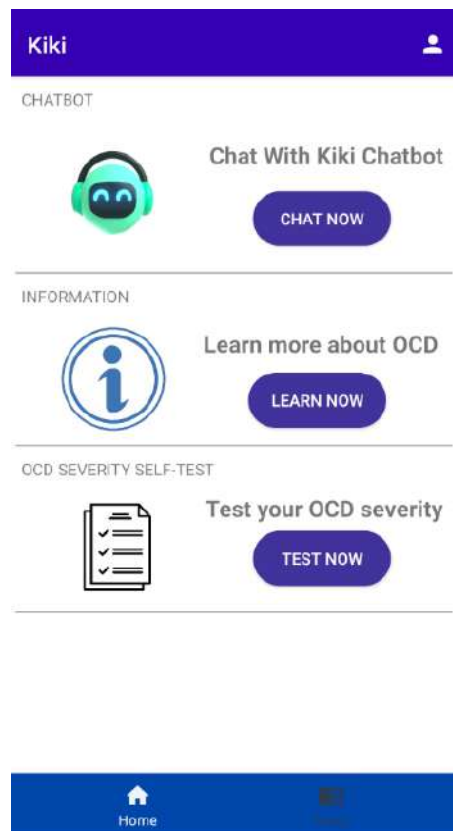


Figure 3. 10 Dashboard User Interface

3.1.2 Manage Profile

Figure 3.11 show the profile interface for both the admin and user when they click on the profile icon at the right top menu bar from the Dashboard Interface. When clicking on the <<Logout>> button, the admin and user will logout the system and redirect back to Main Interface. The actor and user also can change or add their profile picture by clicking on the profile image on the top left interface.

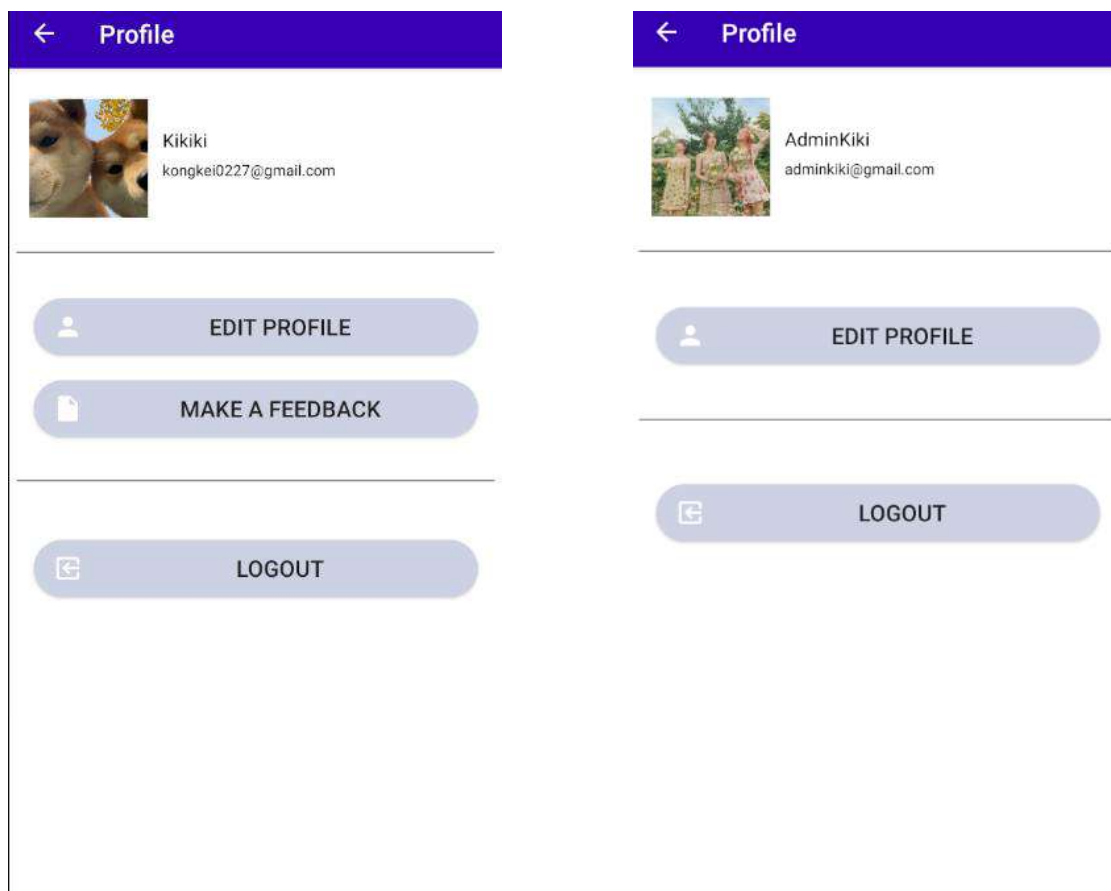


Figure 3. 11 Profile Interface for User (Left) and Admin (Right)

When clicking on the <<EDIT PROFILE>> button, a prompt will show as in Figure 3.12 that allow the user or admin to edit the username but the email is not editable. When the username is valid and the actor click on the <<EDIT>>, then the username will change into the system and database.

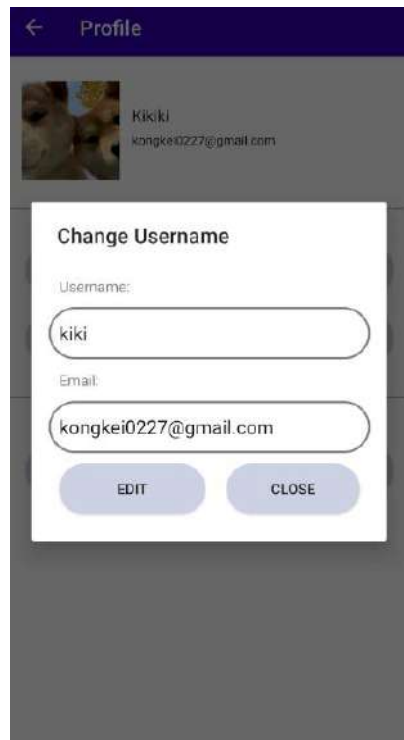


Figure 3. 12 Edit Profile Prompt in Profile Interface

When the username is blank and <<EDIT>> button is being clicked, an alert error will be display beside the username field as shown in Figure 3.13.

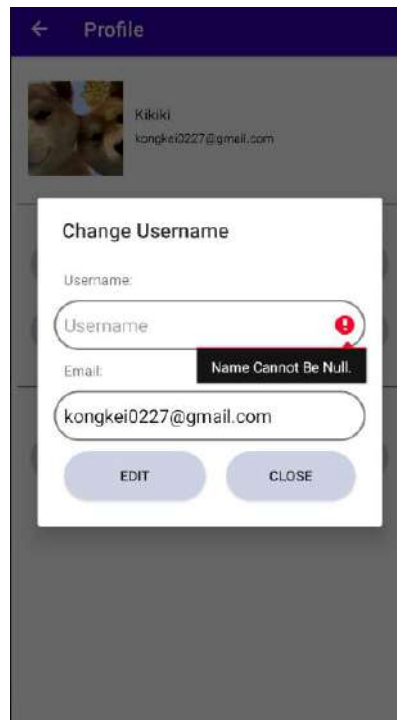
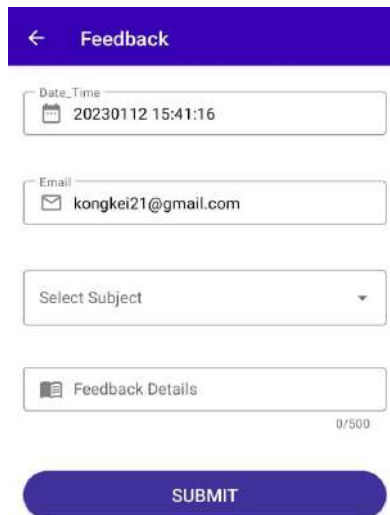


Figure 3. 13 Alert Error in Edit Profile Prompt

Figure 3.14 show the Feedback Interface for the user when the <<MAKE A FEEDBACK>> button in profile user is being clicked. The date and time field is generated automatically and is not editable, email field is not editable and the user can enter subject and feedback details and click on the <<SUBMIT>> button to make feedback.



The image shows a mobile application interface for providing feedback. At the top, there is a blue header bar with a white left-pointing arrow and the text "Feedback". Below the header, there are four input fields stacked vertically. The first field is labeled "Date_Time" and contains the text "20230112 15:41:16". The second field is labeled "Email" and contains the text "kongkei21@gmail.com". The third field is a dropdown menu labeled "Select Subject". The fourth field is labeled "Feedback Details" and has a character count "0/500" below it. At the bottom of the form is a large, rounded blue button with the text "SUBMIT" in white.

Figure 3. 14 User Feedback Interface

When the field is null and the user click on <<SUBMIT>>, an alert error will show at the respective field as shown in Figure 3.15. The text field box will turn red when the user exceeds the words.

The image displays two side-by-side screenshots of a feedback form titled "Feedback". Both screenshots show the same form fields: "Date_Time", "Email", "Select Subject", and "Feedback Details", each with a "SUBMIT" button below. In the left screenshot, the "Date_Time" field contains "20230112 15:41:16", the "Email" field contains "kongkei21@gmail.com", and the "Select Subject" field is empty. A red error message "This field cannot be empty!" is shown below the "Select Subject" field. In the right screenshot, the "Date_Time" field contains "20230112 15:42:04", the "Email" field contains "kongkei21@gmail.com", and the "Select Subject" field is set to "Bugs Report". The "Feedback Details" field is empty, and a red error message "This field cannot be empty!" is shown below it.

Figure 3. 15 Alert Error in Feedback Interface When Field is Blank.

Figure 3.16 show the alert error when the user exceeds the word limitation on details field and click <<SUBMIT>>.

The screenshot shows a mobile application interface for a feedback form. At the top, there is a blue header bar with a back arrow and the text "Feedback". Below the header, there are three input fields: "Date_Time" with a calendar icon and the value "20230112 15:42:15", "Email" with an envelope icon and the value "kongkei21@gmail.com", and "Select Subject" with a dropdown menu showing "Chatbot". Below these fields is a "Feedback Details" section with a red border, containing a text area with the repeated text "chatbot not understand my question" and a character count "525/500". At the bottom, there is a blue "SUBMIT" button.

Figure 3. 16 Alert Error in Feedback Interface When Words Exceed.

Figure 3.17 shows the interface for the Feedback List Interface. When clicking at the bottom right navigation bar at the Admin Dashboard, the admin can view all the new feedback that made by the user.

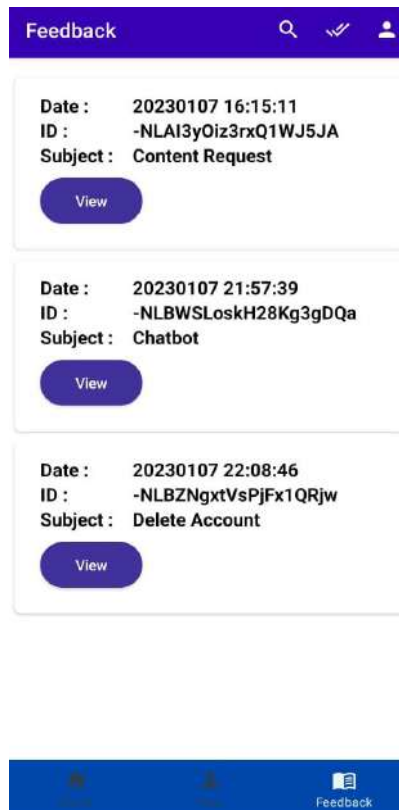


Figure 3. 17 Feedback List Interface

Figure 3.18 show when clicking on the <<View>> button on each feedback, a dialog interface will show and display the details of the respective feedback to let the admin to view the feedback. Admin can click on the <<Mark Done>> button when finish to read or settle the problem or issue that have been reported.

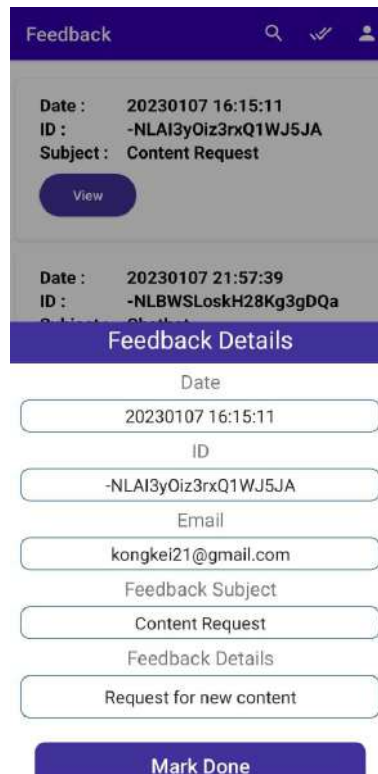


Figure 3. 18 Feedback Details Interface

Figure 3.19 show the Feedback history when clicking on the “Tick” icon at the top menu bar. Admin can view all the feedback that have been done at this interface.

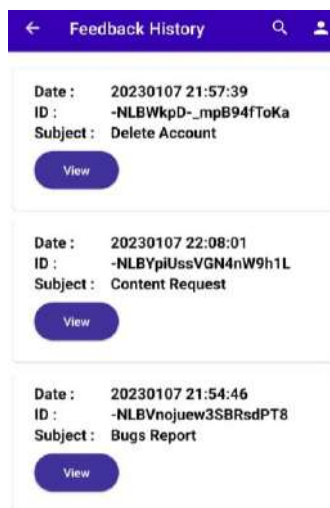


Figure 3. 19 Feedback History Interface

When clicking on the search icon at both the Feedback Admin and Feedback History Interface, the user can enter the subject title to search for the feedback. Figure 3.20 show the search interface for both the interface.

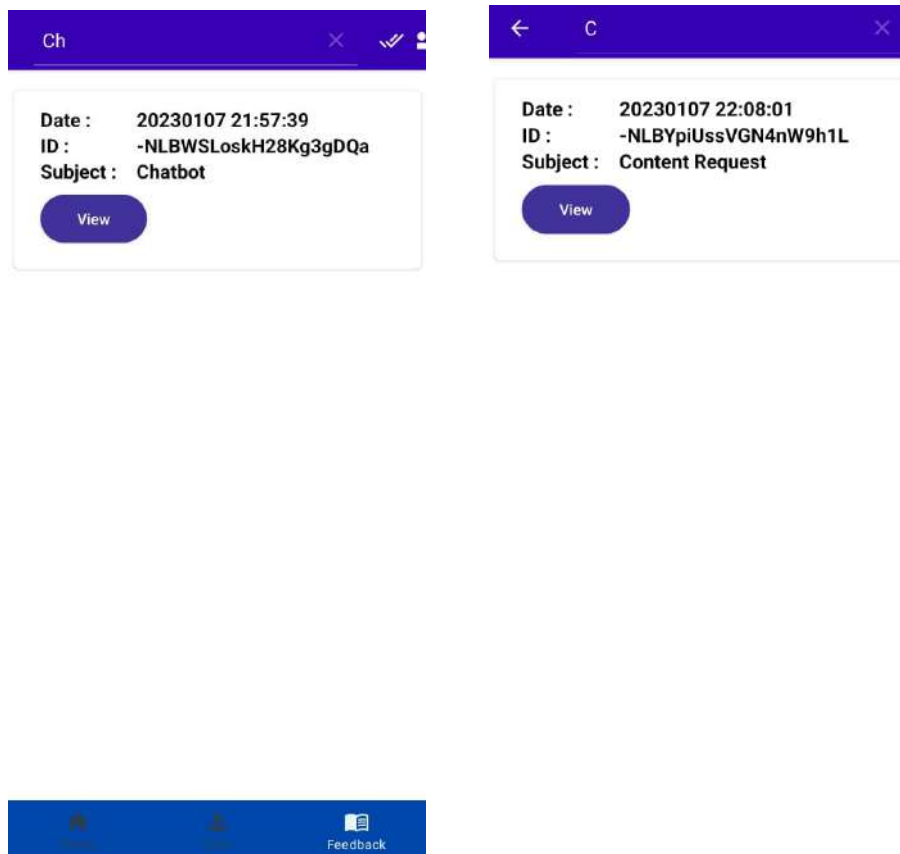


Figure 3. 20 Search Interface in Feedback List Interface (Left) and Feedback Done Interface (Right)

When the admin clicks on the <<User>> at the bottom navigation menu, the user list interface will show as Figure 3.21.

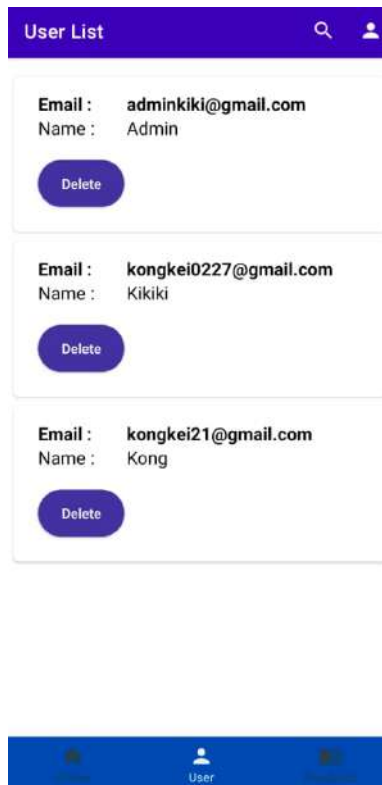


Figure 3. 21 User List Interface

When the admin clicks on the <<Delete>> button on each of the user information, a prompt will display to ask if the admin really want to deletes the user from the system as shown in Figure 3.22. If the admin clicks on Delete at the prompt, then the user data will be deleted from the system and database.

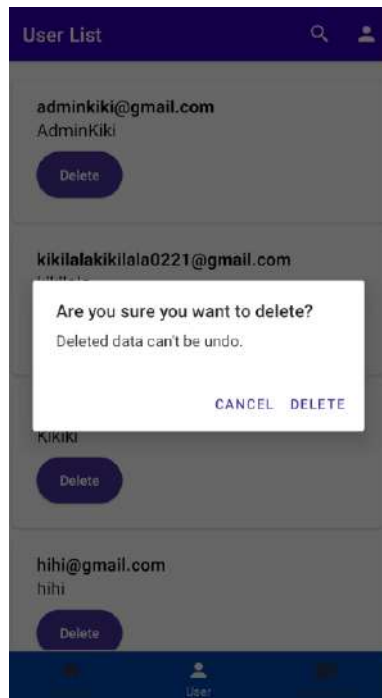


Figure 3. 22 Delete Prompt at User List Interface

Admin can search for specific user by clicking the “Search” icon at the top menu and entering the email of the user as shown in Figure 3.23.

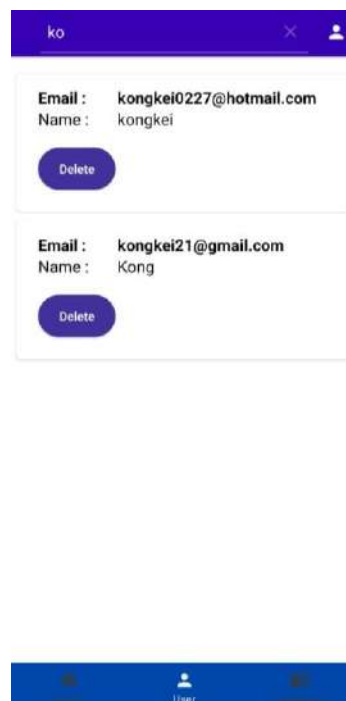


Figure 3. 23 Search Interface in User List Interface

3.1.3 Manage Diary

Figure 3.24 show the diary interface for the user. The user can be redirect to this interface by clicking the <<Diary>> icon and word at the bottom right of the menu. The user can add new diary by clicking the <<+>> button at the right bottom, view the details of the diary by clicking the <<Edit>> button, and delete the diary by clicking on the <<Delete>> button.

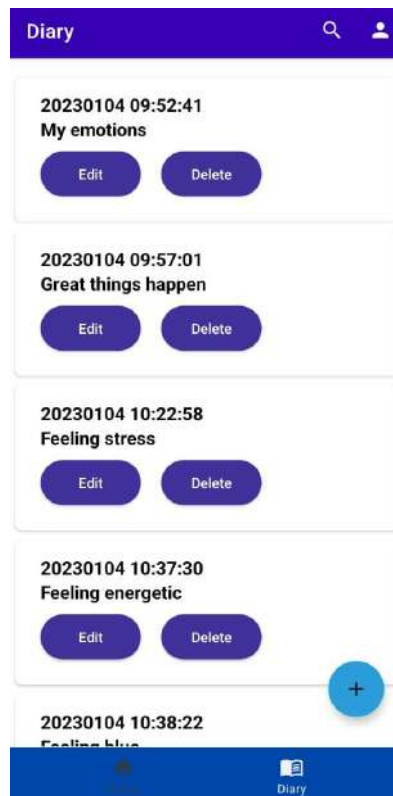


Figure 3. 24 Diary Interface

Figure 3.25 show the diary interface for the user to add a new diary. The user can add a new diary by entering the diary title, diary content and click on the <<ADD DIARY>> button. The Date_Time field is generate by the system and is not editable.

← Diary

Date_Time
20230104 09:52:41

T Diary Title
0/200

Diary Content
0/500

ADD DIARY

Figure 3. 25 Add Diary Interface

The diary title and diary content cannot be null or exceed 220 and 500 words respectively. If it is exceeding, an alert message will show at the respective field. Figure 4.26 show the null alerting while Figure 3.27 show the exceeding alerting.

The figure displays two versions of the 'Add Diary' form. Both forms have a purple header with a back arrow and the text 'Diary'. The 'Date_Time' field is a text input with a calendar icon and the value '20230104 09:52:41'. The 'Diary Title' field is a text input with a red alert message 'This field cannot be empty!' and a character count of 17/200. The 'Diary Content' field is a text input with a document icon and a character count of 0/500. A blue 'ADD DIARY' button is located at the bottom of each form.

Figure 3. 26 Alert message on Null at Add Diary Interface

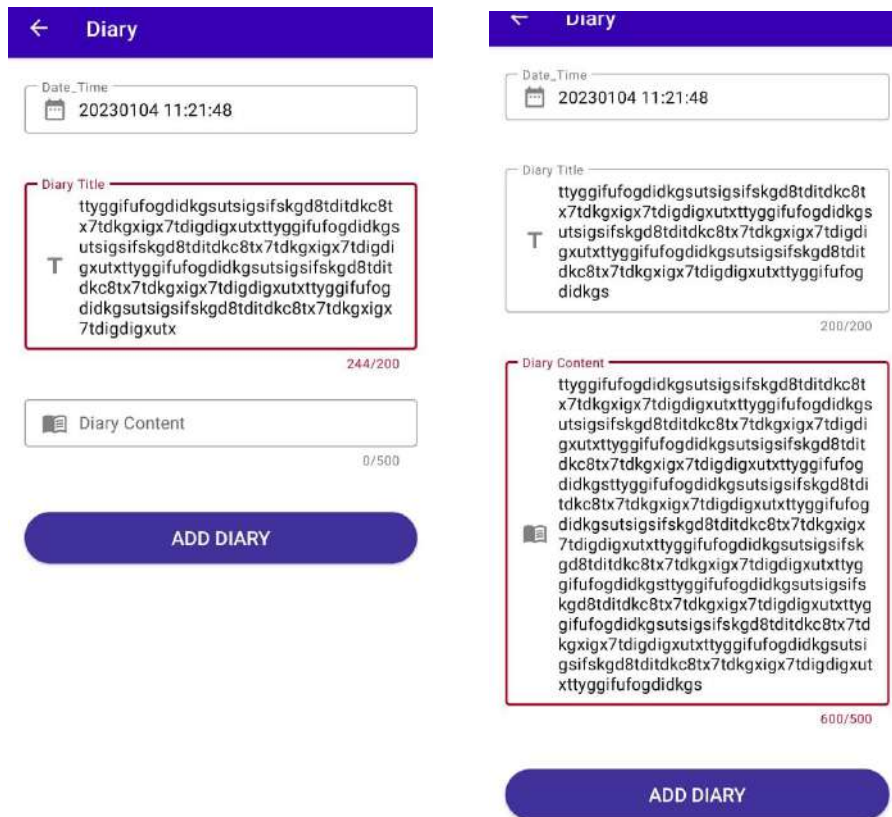


Figure 3. 27 Alert message on exceeding words in Add Diary Interface

Figure 3.28 show the details of the diary. User can update the diary after editing the field and clicks on the <<Update>> button. Date field is not editable.

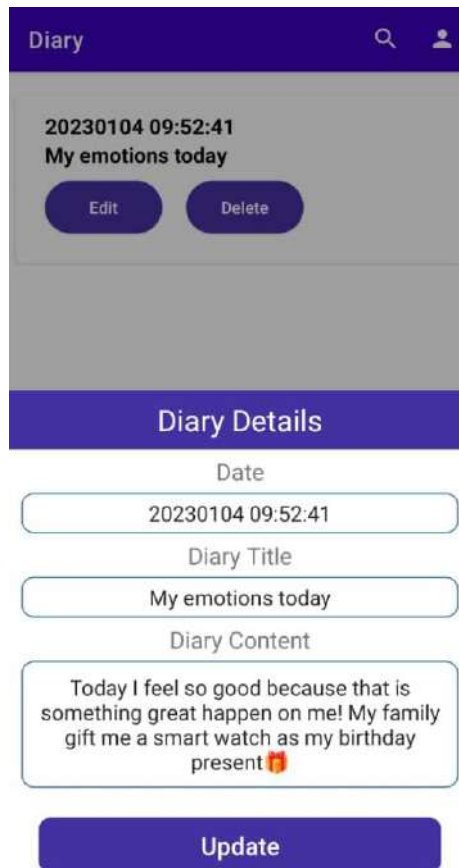


Figure 3. 28 Details of Diary Interface in Diary Interface

Figure 3.29 show the prompt when the user clicks on the <<Delete>> button to delete the specific diary. When user clicks on the <<DELETE>> button at the prompt. The specific diary will be deleted.

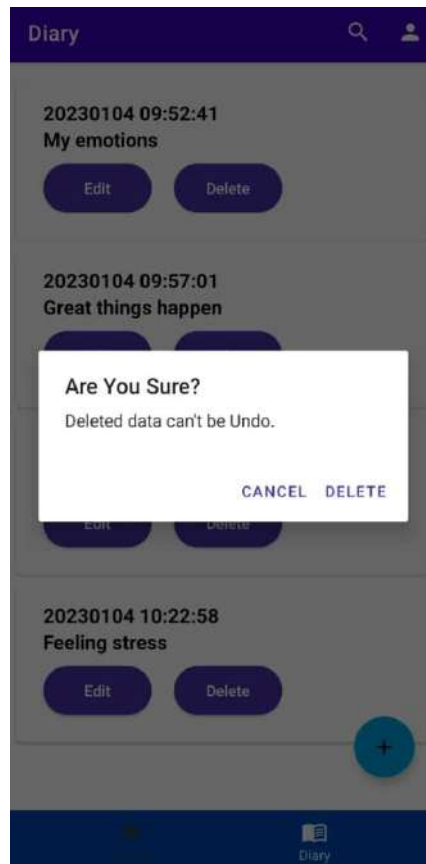


Figure 3. 29 Delete prompt at Diary Interface

User is able to search the diary by date by clicking on the search icon on the top right. Figure 3.30 show the interface for the search in diary interface.

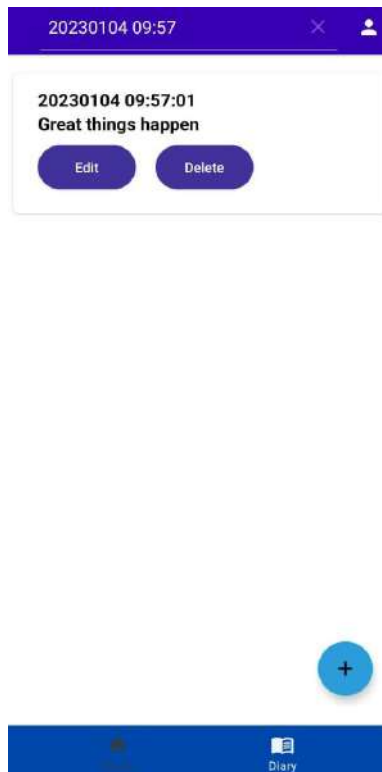


Figure 3. 30 Search in Diary Interface

3.1.4 Manage Information

Figure 3.31 show the Information Interface for admin. The admin can be redirect to this interface by clicking at the <<LEARN NOW>> button at the Admin Dashboard. The admin is able to add new information by clicking on the <<+>> button at the top right of the menu bar. The admin is also able to redirect tot the information website, edit and delete the information.

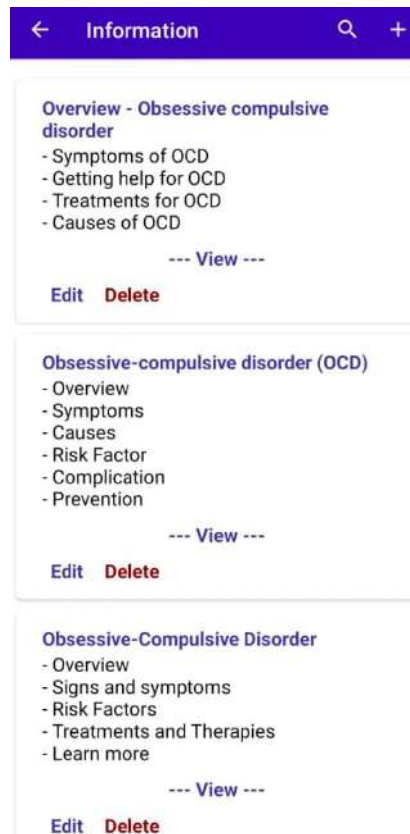
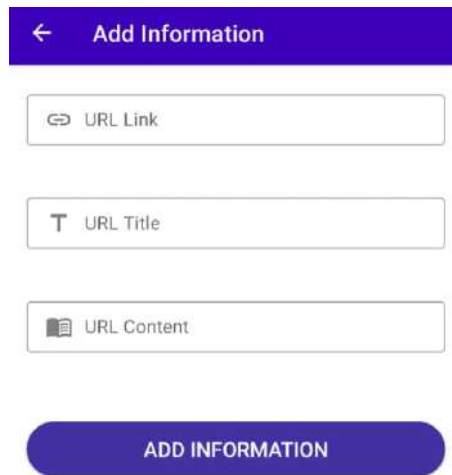


Figure 3. 31 Admin Information Interface

Figure 3.32 show the Add Information Interface for admin. The admin requires to enter URL link, title and content and click on the <<ADD INFORMATION>> button to add the new information.



The image shows a mobile application interface for adding information. At the top, there is a dark blue header bar with a white left-pointing arrow and the text "Add Information". Below the header are three white input fields with rounded corners and thin grey borders. The first field is labeled "URL Link" and contains a small grey link icon. The second field is labeled "URL Title" and contains a small grey text icon. The third field is labeled "URL Content" and contains a small grey document icon. At the bottom of the form is a dark blue rounded rectangular button with the white text "ADD INFORMATION".

Figure 3. 32 Add Information Interface

All the field cannot be null. If the field is null or the URL link is not matching the format of a link, an alert message will show at the respective field. Figure 3.33 show the alert message on the field.

The figure displays two side-by-side screenshots of the 'Add Information' interface. Both screenshots show a purple header bar with a back arrow and the text 'Add Information'. Below the header, there are three input fields: 'URL Link', 'URL Title', and 'URL Content'. The 'URL Content' field contains a list of items: '-Overview', '-Signs and Symptoms', '-Risk Factor', and 'Treatment'. In the left screenshot, the 'URL Link' field is empty and has a red error message 'The link is invalid' below it. In the right screenshot, the 'URL Link' field contains the text 'pics/obsessive-compulsive-disorder-ocd', the 'URL Title' field is empty and has a red error message 'This field cannot be empty!' below it, and the 'URL Content' field is filled with the same list of items. At the bottom of each screenshot is a purple button with the text 'ADD INFORMATION'.

The figure displays a single screenshot of the 'Add Information' interface. It shows a purple header bar with a back arrow and the text 'Add Information'. Below the header, there are three input fields: 'URL Link', 'URL Title', and 'URL Content'. The 'URL Link' field contains the text 'pics/obsessive-compulsive-disorder-ocd'. The 'URL Title' field contains the text 'Obsessive-Compulsive Disorder'. The 'URL Content' field is empty and has a red error message 'This field cannot be empty!' below it. At the bottom of the screenshot is a purple button with the text 'ADD INFORMATION'.

Figure 3. 33 Alert Message on Add Information Interface

Figure 3.34 show the details of the information and the admin are able to change the details of the information by clicking on the <<Edit>> button, editing the field and click on <<Update>> button.



Figure 3. 34 Details of information at Information Interface for admin

Figure 3.35 show the delete prompt when the admin clicks on the <<Delete>> button.

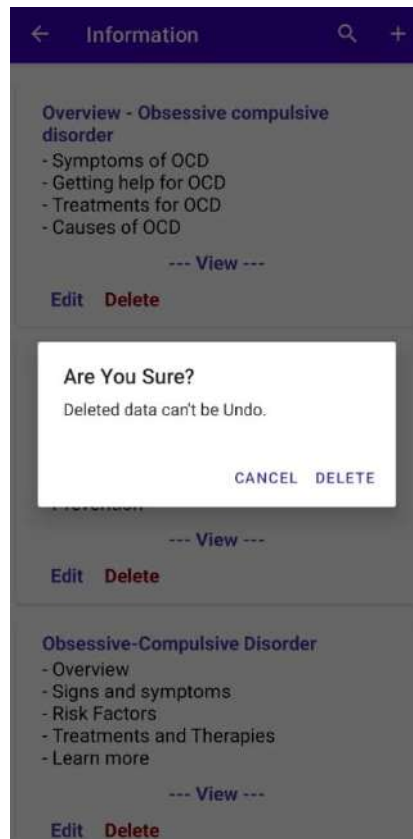


Figure 3. 35 Delete prompt at Information Interface for admin

Figure 3.36 show the Information Interface for the user. The user can click on the <<LEARN NOW>> button to redirect to this interface at the user dashboard. The user can click on the <<View>> button to redirect to the website for the information.

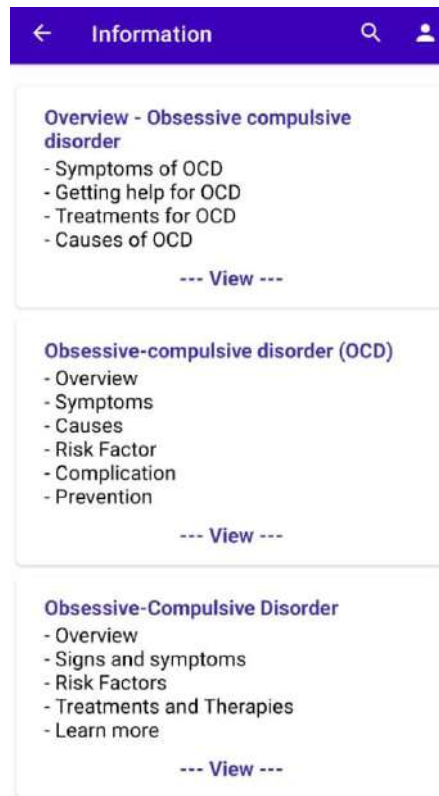


Figure 3. 36 Information Interface for user

Figure 3.37 show the search interface for the admin and user while they click on the search icon on the top right of the menu.

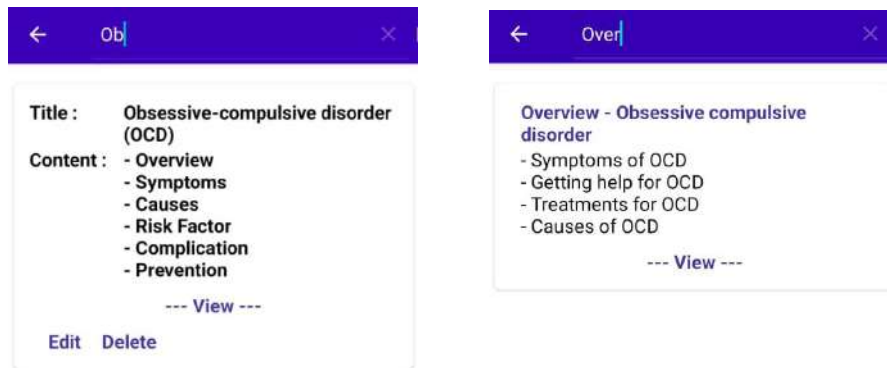


Figure 3. 37 Search Interface at Information Interface for admin(left) and user(right)

3.1.5 Manage Chatting

Figure 3.38 show the Chatbot Interface for both the admin and user when they click on the <<CHAT NOW>> button on the dashboard. User can type and click the send button to send message to the chatbot and get response from it. The user can also choose from the hint or suggestion above the text box to send question to the chatbot.

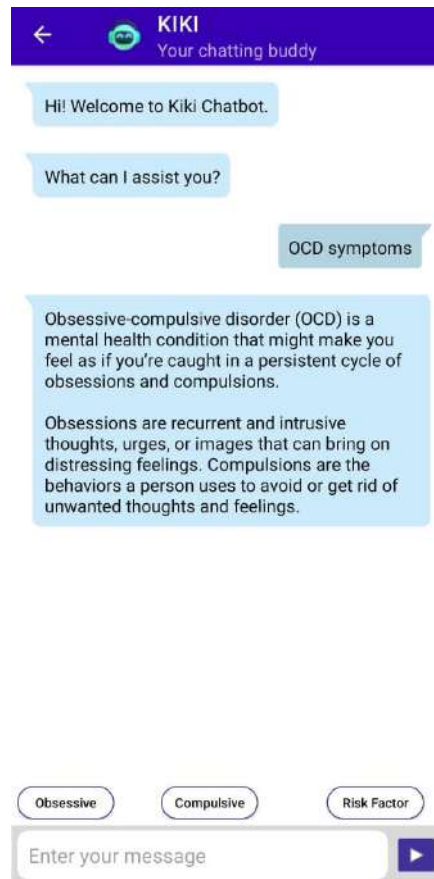


Figure 3. 38 Chatbot Interface for both the admin and user.

3.1.1 Manage Self-Assesment

Figure 3.39 show the Self-Assessment Interface for the user and the admin when the user or admin click on the <<TEST NOW>> button on the Dashboard.

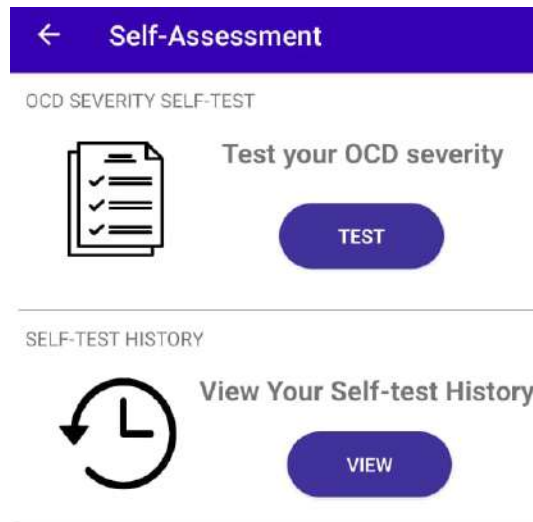


Figure 3. 39 Self-Assessment Interface for User and Admin

Figure 3.40 show the Self-Assessment Test Interface for the user and admin when they click on the <<TEST>> button from the Self-Assessment Interface. The user and admin can pull the answer bar to what the context of their feelings and click on the <<Submit>> button.

←
Self-Assessment Test

0-None 1-Mild 2-Moderate 3-Severe 4-Highest

1. TIME OCCUPIED BY OBSESSIVE THOUGHTS
 Q: How frequently do the obsessive thoughts occur?

Q1: 0

0-None 1-Mild 2-Moderate 3-Severe 4-Extreme

2. INTERFERENCE DUE TO OBSESSIVE THOUGHTS
 Q: How much do your obsessive thoughts interfere with your social or work (or role) functioning?

Q2: 0

0-None 1-Mild 2-Moderate 3-Severe 4-Extreme

3. DISTRESS ASSOCIATED WITH OBSESSIVE THOUGHTS
 Q: How much distress do your obsessive thoughts cause you? [distress can refer as anxiety or disturbing]

Q3: 0

8. DISTRESS ASSOCIATED WITH COMPULSIVE BEHAVIOR
 Q: How would you feel if prevented from performing your compulsion(s)? [Is prevented from performing your compulsion make you anxious?]

Q8: 0

0-Makes an effort to always resist, or symptoms so minimal doesn't need to actively resist
 1-Tries to resist most of the time
 2-Makes some effort to resist
 3-Yields to almost all compulsions without attempting to control them, but does so with some reluctance
 4-Completely and willingly yields to all compulsions

9. RESISTANCE AGAINST COMPULSIONS
 Q: How much of an effort do you make to resist the compulsions? [Only rate effort made to resist, not success or failure in actually controlling the compulsions.]

Q9: 0

0-Complete control
 1-Much control
 2-Moderate control
 3-Little control
 4-No control

10. DEGREE OF CONTROL OVER COMPULSIVE BEHAVIOR
 Q: : How strong is the drive to perform the compulsive behavior? [How much control do you have over the compulsions?]

Q10: 0

SUBMIT

Figure 3. 40 Self-Assessment Test Interface for User and Admin

Figure 3.41 show the Self-Assessment Result Interface for the user and the admin after they click on the <<Submit >> button from the Self-Assessment Test Interface. In this interface, the risk of the user or admin getting the obsessive and compulsive OCD will be shown in a graphical form and there are some advice to the user or admin. Pressing the back button will redirect the user back to the Self-Assessment Interface.

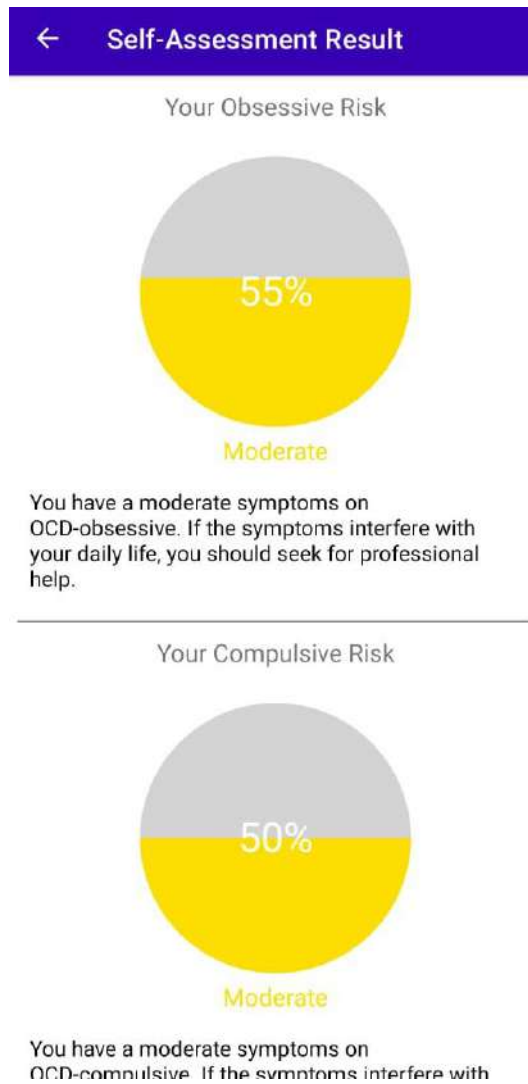


Figure 3. 41 Self-Assessment Result Interface for User and Admin

Figure 3.42 show the Self-Assessment History Interface when the user or admin click on the <<VIEW>> button on the Self-Assessment Interface. In this interface, the user are able to view all the history of their self-assessment test.

← Self-Assessment History		
Date and Time	Obsessive Category	Compulsive Category
20230108 22:45:20	Nearly Absent	Nearly Absent
20230108 23:32:56	Severe	Extreme
20230109 23:40:57	Moderate	Moderate
20230109 23:43:50	Mild	Mild
20230110 00:06:48	Severe	Severe
20230110 14:22:00	Moderate	Extreme
20230110 14:25:53	Extreme	Extreme
20230110 14:30:30	Severe	Nearly Absent
20230110 14:33:19	Severe	Nearly Absent
20230110 14:34:59	Severe	Nearly Absent
20230110 14:44:52	Mild	Nearly Absent
20230110 14:45:07	Mild	Nearly Absent
20230110 14:54:57	Moderate	Moderate
20230110 15:14:22	Severe	Severe
20230110 15:48:25	Severe	Severe
20230110 15:49:57	Extreme	Moderate
20230110 15:53:36	Severe	Moderate
20230110 16:14:15	Moderate	Moderate
20230112 14:34:13	Nearly Absent	Nearly Absent
20230112 14:34:32	Moderate	Moderate

Figure 3. 42 Self-Assessment History Interface for Admin and User

3.2 HARDWARE AND SOFTWARE SPECIFICATION

i. Hardware Specification

Item	Quantity	Specification	Purpose
Laptop	1	<ul style="list-style-type: none"> - 12 GB RAM - Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz - 64-bit Microsoft® Windows® 10 	<ul style="list-style-type: none"> - Use to run android studio and others software to develop the application - Use to prepare all the software documentation
Smartphone	1	<ul style="list-style-type: none"> - 2.4 GHz Snapdragon 765G Octa-core - 8 GB RAM 	<ul style="list-style-type: none"> - Use to run the application that developed.

		- Android version 12	
--	--	----------------------	--

Table 3. 1 Hardware Specification

ii. Software Specification

Software	Purpose
Microsoft Office Word 2019	To prepare and create documentation.
Draw.io	To design and draw all the related diagram.
Android Studio	To develop the mobile application using Java language.
Android emulator	To simulate the Android mobile phone.
Firebase	To enable real time syncing of data.
Windows 10	Operating system that run all the software.
Dialogflow	To script the script for chatbot.

Table 3. 2 Software Specification

APPENDIX B
SOFTWARE DESIGN DESCRIPTION (SDD)


2022

SOFTWARE DESIGN DESCRIPTION (SDD)

[KIKI: A CHATBOT APPLICATION FOR OCD
IN ANDROID MOBILE APPLICATION]



DOCUMENT APPROVAL

	Name	Date
Authenticated by:  _____ Name	KONG KEI	08/02/2023
Approved by: _____ Client	DR. ANIS FARIHAN BINTI MAT RAFFEI	08/02/2023

Software : Draw IO, Microsoft Word

Archiving Place : Google Drive

TABLE OF CONTENT

CONTENT	PAGE
DOCUMENT APPROVAL	ii
TABLE OF CONTENT.....	iii
LIST OF FIGURES	vi
LIST OF TABLES	vii
LIST OF APPENDICES	x
1.1 PROJECT DESCRIPTION.....	1
1.2 SYSTEM IDENTIFICATION	1
1.3 ARCHITECTURE / BLUE PRINT	2
1.3.1 General Architecture	2
1.3.2 Package Module	2
1.3.2.1 Application Layer	3
1.3.2.2 Manage Registration	3
1.3.2.3 Manage Chatting	4
1.3.2.4 Manage Self-test.....	4
1.3.2.5 Manage Information	4
1.3.2.6 Manage Diary.....	5
1.3.2.7 Manage Profile.....	5
1.3.3 Business Service Layer	6
1.3.3.1 Controller	6
1.3.3.2 Model	7
1.4 ARCHITECTURE / BLUEPRINT DESCRIPTION	7
1.4.1 Manage Registration.....	7
1.4.2 Manage Chatting.....	8
1.4.3 Manage Self-test	8
1.4.4 Manage Information	8
1.4.5 Manage Diary	8
1.4.6 Manage Profile	9
2.1 DETAILED DESCRIPTION	10

2.1.1	Manage Registration	10
2.1.1.1	Main	10
2.1.1.2	Login	10
2.1.1.3	Register	11
2.1.1.4	Reset Password	11
2.1.1.5	User Homepage	12
2.1.1.6	Admin homepage	12
2.1.1.1	Manage Registration Controller	13
2.1.1.1	Manage Registration Model	14
2.1.2	Manage Chatting	16
2.1.2.1	Chatbot	16
2.1.2.2	Chatbot Controller	16
2.1.3	Manage Self-test	18
2.1.3.1	Self-assessment Menu	18
2.1.3.2	Self-assessment	18
2.1.3.3	Result	19
2.1.3.4	Self-assessment History	19
2.1.3.5	Self-test Controller	20
2.1.3.6	Self-test Model	22
2.1.4	Manage Information	23
2.1.4.1	Manage Information	24
2.1.4.2	Add New Information	24
2.1.4.3	Edit Information	25
2.1.4.4	Manage Information Controller	25
2.1.4.5	Manage Information Model	27
2.1.5	Manage Diary	30
2.1.5.1	Diary	30
2.1.5.2	Add New Diary	31
2.1.5.3	Diary Details	32
2.1.5.4	Manage Diary Controller	32
2.1.5.5	Manage Diary Model	34
2.1.6	Manage Profile	36
2.1.6.1	Profile	36

2.1.6.2	Edit Profile	37
2.1.6.3	Feedback.....	37
2.1.6.4	Feedback details.....	38
2.1.6.5	Manage Profile Controller	38
2.1.6.6	Manage Profile Model.....	41
2.2	DATA DICTIONARY.....	44

LIST OF FIGURES

Figure 1. 1 General Architecture for Kiki Chatbot Mobile Application.....	2
Figure 1. 2 Package Module for Kiki Chatbot Mobile Application	2
Figure 1. 3 Application Layer for Kiki Chatbot Mobile Application	3
Figure 1. 4 Business Service Layer for Kiki Chatbot Mobile Application.....	6
Figure 1. 5 Class Diagram for Kiki Chatbot Mobile Application.....	7
Figure 2. 1 Manage Registration Module	10
Figure 2. 2 Manage Chatting Module	16
Figure 2. 3 Manage Self-test Module	18
Figure 2. 4 Manage Information Module.....	23
Figure 2. 5 Manage Diary Module.....	30
Figure 2. 6 Manage Profile Module.....	36

LIST OF TABLES

Table 1. 1 Application Layer – Manage Registration.....	4
Table 1. 2 Application Layer – Manage Chatting.....	4
Table 1. 3 Application Layer – Manage Self-test	4
Table 1. 4 Application Layer – Manage Information	5
Table 1. 5 Application Layer – Manage Diary	5
Table 1. 6 Application Layer – Manage Profile	5
Table 1. 7 Business Service Layer – Controller	6
Table 1. 8 Business Service Layer – Model	7
Table 1. 9 Manage Registration Controller Class.....	8
Table 1. 10 Manage Chatting Controller Class.....	8
Table 1. 11 Manage Self-test Controller Class	8
Table 1. 12 Manage Information Controller Class	8
Table 1. 13 Manage Diary Controller Class	9
Table 1. 14 Manage Profile Controller Class.....	9
Table 2. 1 Main.....	10
Table 2. 2 Login	11
Table 2. 3 Register.....	11
Table 2. 4 Reset Password	12
Table 2. 5 User Homepage.....	12
Table 2. 6 Admin Homepage.....	12
Table 2. 7 Manage Registration Controller	14
Table 2. 8 Manage Registration Model.....	16
Table 2. 9 Chatbot	16
Table 2. 10 Chatbot Controller	17
Table 2. 11 Self-assessment Menu	18

Table 2. 12 Self-assessment	19
Table 2. 13 Result	19
Table 2. 14 Self-assessment History	20
Table 2. 15 Self-test Controller	22
Table 2. 16 Self-test Model	23
Table 2. 17 Manage Information	24
Table 2. 18 Add New Information	24
Table 2. 19 Edit Information	25
Table 2. 20 Manage Information Controller	27
Table 2. 21 Manage Information Model	30
Table 2. 22 Diary	31
Table 2. 23 Add New Diary	31
Table 2. 24 Diary Details	32
Table 2. 25 Manage Diary Controller	34
Table 2. 26 Manage Diary Model	36
Table 2. 27 Profile	37
Table 2. 28 Edit Profile	37
Table 2. 29 Feedback	38
Table 2. 30 Feedback Details	38
Table 2. 31 Manage Profile Controller	41
Table 2. 32 Manage Profile Model	44
Table 2. 33 Profile	44
Table 2. 34 Information	45
Table 2. 35 Feedback	45
Table 2. 36 Diary	46
Table 2. 37 History	47

LIST OF APPENDICES

CHAPTER 1

1.1 PROJECT DESCRIPTION

The Kiki Chatbot Mobile application is an application that provide information and guidance for the obsessive-compulsive disorder (OCD) mental illness. The project is to allow the user to chat with the chatbot inside the mobile application to get information, doing a self-assessment to test the severity of the OCD. The project also allows the user to get more information about the OCD in form of website. The project will provide a diary section which allow the user to record their daily emotions. This document is intended for the use of the Kiki Application developer, project manager, management team, system user, system testers and documentation writer.

The Kiki chatbot mobile application consist of 6 modules which are manage registration, manage chatting, manage self-test, manage information, manage diary and manage profile. Manage registration module allows the user to create account in the system. The system also allows the admin to register a new admin account in the system. Manage chatting module allow allows the user to chat using text through chatbot Kiki. Manage self-test module allow the user to take a self-assessment to test their severity on OCD. Manage Information allow both the admin and user to view the information and the admin to add, edit and delete website information about OCD in the system. Manage diary allow user to view, add or delete diary in the system. Manage profile allow the user and admin view and edit their profile, delete their account. The system allows the user to make feedback in the system. The system allows the admin to view feedback made by user in the system.

1.2 SYSTEM IDENTIFICATION

This document uses the following convention:

System Identification Number: **SDD-KIKI-V01-22**

SDD: Software Design Description

KIKI: Kiki chatbot mobile application

V01: Version 1

22: Year 2022

The format of recording the identification of the system is shown as the above information. SDD stands for Software Design Description which is a type of document that is a representation of a software design that is to be used for recording design information, addressing various design concerns, and communicating that information to the design's stakeholders. KIKI stands for the Kiki chatbot mobile application. V01 stands for Version 1 and 22 stands for Year 2022 which is the year the project being developed.

1.3 ARCHITECTURE / BLUE PRINT

1.3.1 General Architecture

The system uses MVC architecture where it consists of three layers, M stands for Model, V stands for Views and C stands for controllers. Figure below shows the general architecture of Kiki Chatbot Mobile Application.

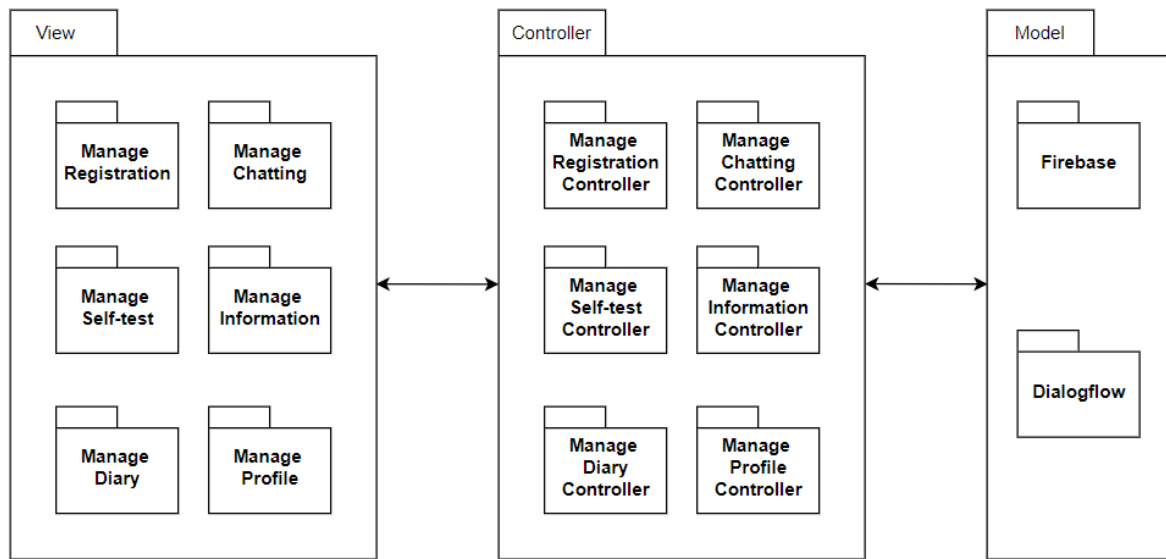


Figure 1. 1 General Architecture for Kiki Chatbot Mobile Application

1.3.2 Package Module

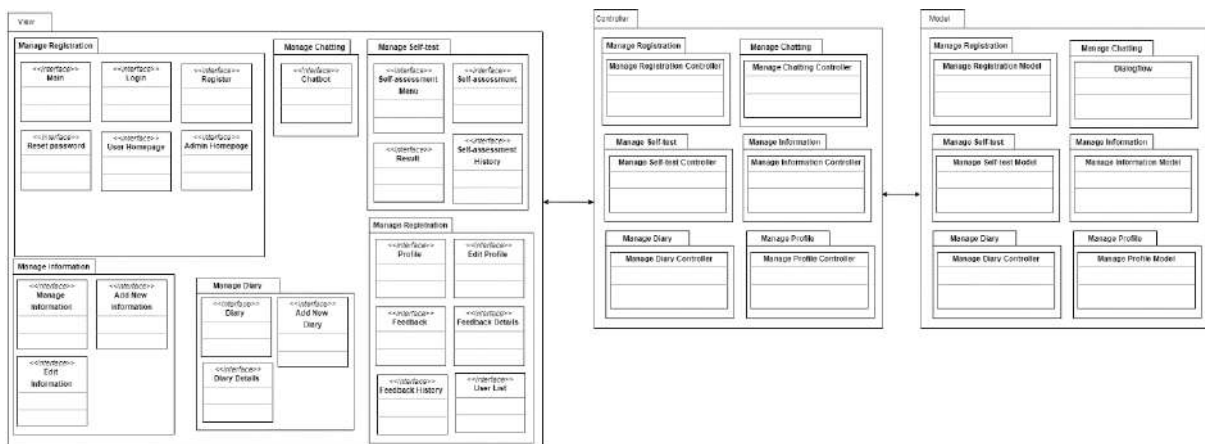


Figure 1. 2 Package Module for Kiki Chatbot Mobile Application

1.3.2.1 Application Layer

The application layer of Kiki Chatbot Mobile Application is as shown in Figure 1.3. The interfaces are categorized into six modules which are manage registration, manage chatting, manage self-test, manage information, manage diary and manage profile. The descriptions of the interfaces are as tables below.

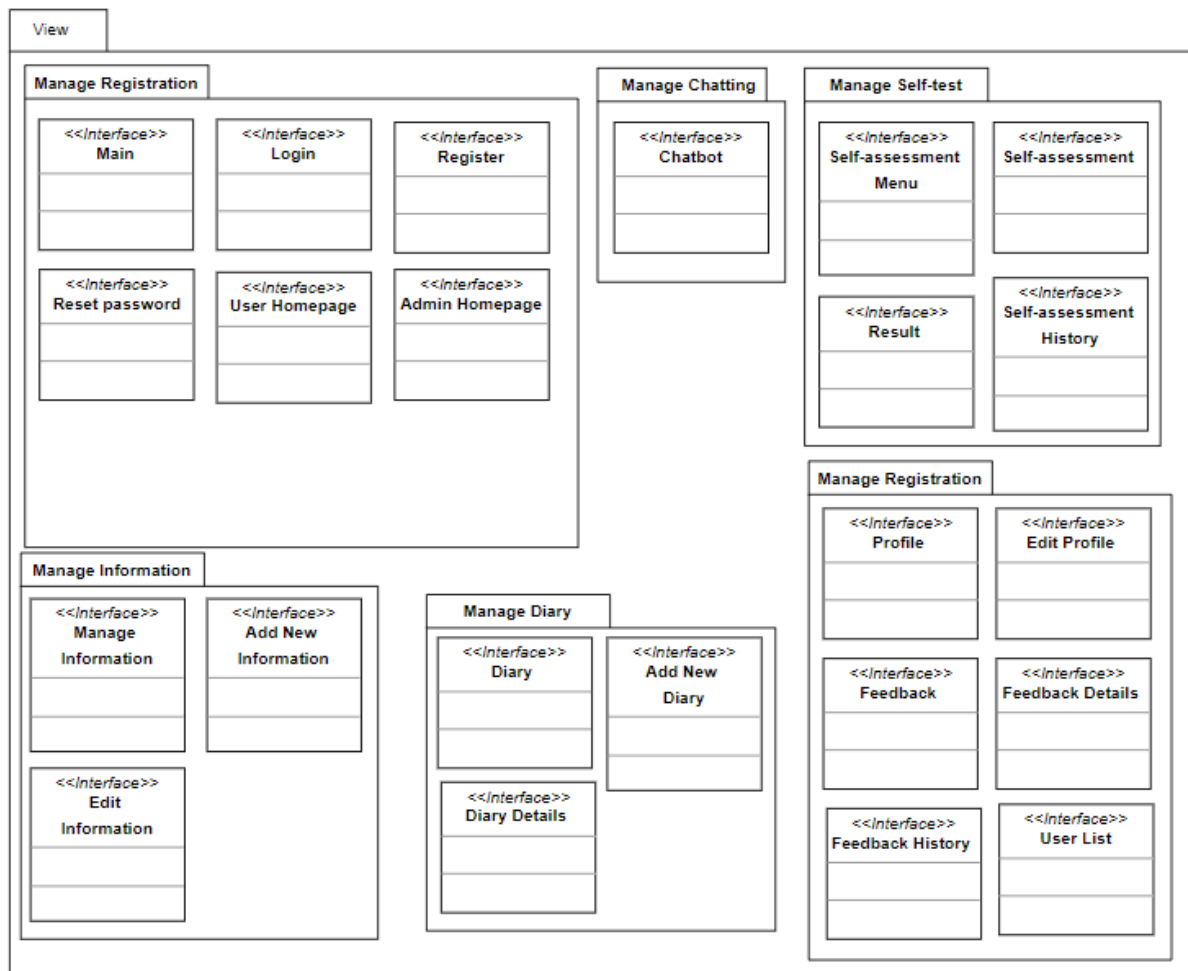


Figure 1. 3 Application Layer for Kiki Chatbot Mobile Application

1.3.2.2 Manage Registration

Class Name	Description
------------	-------------

Main	This interface shows up after the user open the application.
Login	This interface allows the user and admin to login.
Register	This interface allows the user to register their account.
Reset Password	This interface allows the user and admin to reset their password.
User Homepage	This interface shows up after the user login into the application.
Admin Homepage	This interface shows up after the admin login into the application.

Table 1. 1 Application Layer – Manage Registration

1.3.2.3 Manage Chatting

Class Name	Description
Chatbot	This interface allows the user to chat using text with the chatbot.

Table 1. 2 Application Layer – Manage Chatting

1.3.2.4 Manage Self-test

Class Name	Description
Self-assessment Menu	This interface allows the user to view the self-assessment menu.
Self-assessment	This interface allows the user to do the self-test on OCD.
Result	This interface allows the user to view their self-test result.
Self-assessment History	This interface allows the user to view their self-assessment history.

Table 1. 3 Application Layer – Manage Self-test

1.3.2.5 Manage Information

Class Name	Description
Manage Information	This interface allows the user and admin to view the information list.
Add New Information	This interface allows the admin to add new information.

Edit Information	This interface allows the admin to edit information.
------------------	--

Table 1. 4 Application Layer – Manage Information

1.3.2.6 Manage Diary

Class Name	Description
Diary	This interface allows the user to view their diary list.
Add New Diary	This interface allows the user to add new diary.
Diary Details	This interface allows the user to view their diary details.

Table 1. 5 Application Layer – Manage Diary

1.3.2.7 Manage Profile

Class Name	Description
Profile	This interface allows the user and admin to view their profile.
Edit Profile	This interface allows the user and admin to edit their profile.
Feedback	This interface allows the user to make feedback and the admin to view the feedback list.
Feedback Details	This interface allows the admin to view the feedback details.
Feedback History	This interface allows the admin to view feedback history.
User List	This interface allows the admin to view the user list.

Table 1. 6 Application Layer – Manage Profile

1.3.3 Business Service Layer

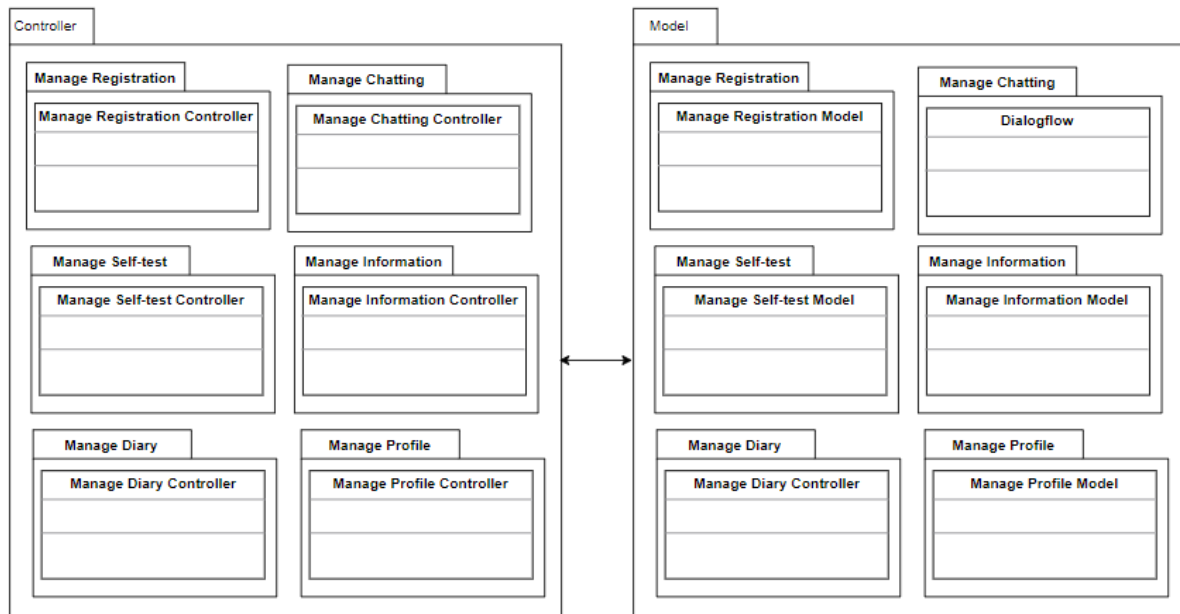


Figure 1. 4 Business Service Layer for Kiki Chatbot Mobile Application

1.3.3.1 Controller

Class Name	Description
Manage Registration Controller	To manage and process the registration data.
Manage Chatting Controller	To manage and process the chatting data.
Manage Self-test Controller	To manage and process the self-test data.
Manage Information Controller	To manage and process the information data.
Manage Diary Controller	To manage and process the diary data.
Manage Profile Controller	To manage and process the profile data.

Table 1. 7 Business Service Layer – Controller

1.3.3.2 Model

Class Name	Description
Manage Registration Model	To store the registration data.
Dialogflow	To store the intent in Dialogflow.
Manage Self-test Model	To store the self-test data.
Manage Information Model	To store the information data.
Manage Diary Model	To store the diary data.
Manage Profile Model	To store the profile data.

Table 1. 8 Business Service Layer – Model

1.4 ARCHITECTURE / BLUEPRINT DESCRIPTION

Figure below show the class diagram for Kiki Chatbot Mobile Application that consists of six modules, which are manage registration, manage chatting, manage self-test, manage information, manage diary and manage profile.

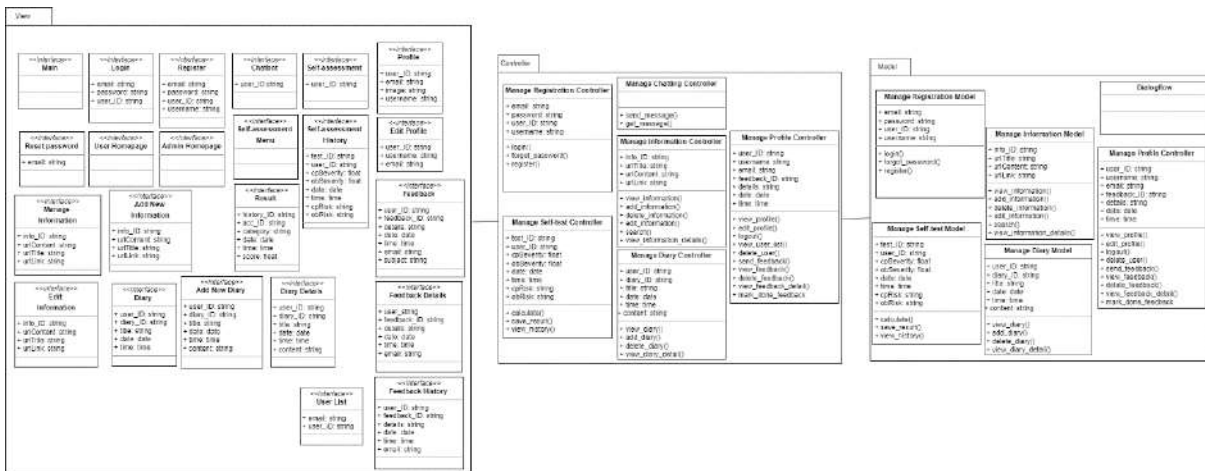


Figure 1. 5 Class Diagram for Kiki Chatbot Mobile Application

1.4.1 Manage Registration

Manage Registration Controller	
login()	Allow the user and admin to login to the application.
forgot_password()	Allow the user and admin to reset password.

register()	Allow the user to register their account.
------------	---

Table 1. 9 Manage Registration Controller Class

1.4.2 Manage Chatting

Manage Chatting Controller	
send_message()	Allow the user to send message.
get_message()	Allow the user to receive reply message.

Table 1. 10 Manage Chatting Controller Class

1.4.3 Manage Self-test

Manage Self-test Controller	
calculate()	To calculate the risk for the test of the user.
save_result()	To save the test result.
view_history()	Allow the user to view their test history.

Table 1. 11 Manage Self-test Controller Class

1.4.4 Manage Information

Manage Information Controller	
view_information()	Allow the user and the admin to view the information list.
add_information()	Allow the admin to add new information.
delete_information()	Allow the admin to delete information.
edit_information()	Allow the admin to edit the information.
search()	Allow the user and admin to search for a information.
view_information_details()	Allow the user and admin to view the information website.

Table 1. 12 Manage Information Controller Class

1.4.5 Manage Diary

Manage Diary Controller	
--------------------------------	--

view_diary()	Allow the user to view their diary list.
add_diary()	Allow the user to add new diary.
delete_diary()	Allow the user to delete diary.
view_diary_detail()	Allow the user to view their diary details.

Table 1. 13 Manage Diary Controller Class

1.4.6 Manage Profile

Manage Profile Controller	
view_profile()	Allow the user and admin to view their profile.
edit_profile()	Allow the user and admin to edit their profile.
logout()	Allow the user and admin logout the application.
delete_account()	Allow the user and admin to delete their account.
view_user_list	Allow the admin to view the user list.
send_feedback()	Allow the user to send feedback.
view_feedback()	Allow the admin to view feedback list.
delete_feedback()	Allow the admin to delete the feedback.
view_feedback_detail()	Allow the admin to view feedback details.
mark_done_feedback()	Allow the user to mark the feedback done.

Table 1. 14 Manage Profile Controller Class

CHAPTER 2

2.1 DETAILED DESCRIPTION

2.1.1 Manage Registration

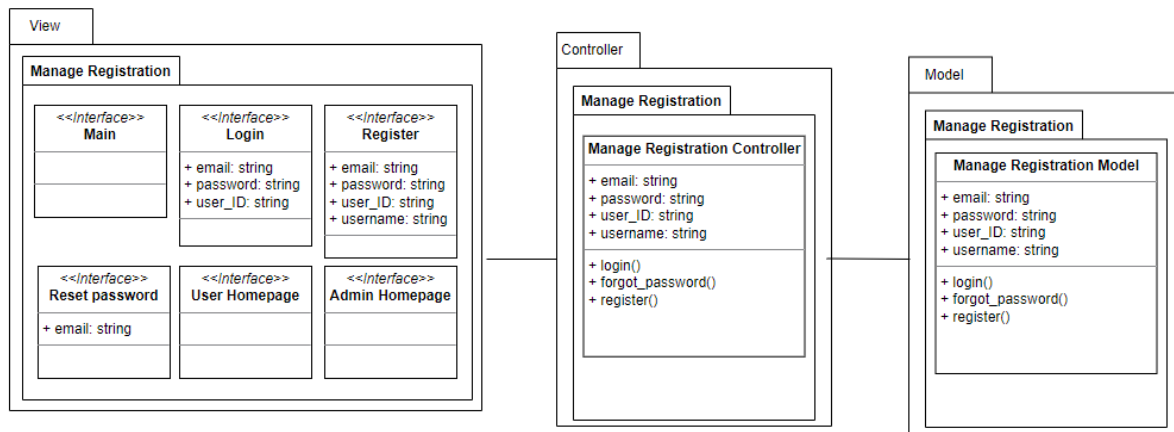


Figure 2. 1 Manage Registration Module

2.1.1.1 Main

Class Type	Boundary Class	
Responsibility	This interface shows up after the user open the application.	
Attributes	Attributes Name	Attributes Type
	N/A	N/A
Methods	Methods Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 1 Main

2.1.1.2 Login

Class Type	Boundary Class
------------	----------------

Responsibility	This interface allows the user and admin to login.	
Attributes	Attributes Name	Attributes Type
	email	String
	password	String
	user_ID	String
Methods	Methods Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 2 Login

2.1.1.3 Register

Class Type	Boundary Class	
Responsibility	This interface allows the user to register their account.	
Attributes	Attributes Name	Attributes Type
	email	String
	password	String
	user_ID	String
	username	String
Methods	Methods Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 3 Register

2.1.1.4 Reset Password

Class Type	Boundary Class	
Responsibility	This interface allows the user and admin to reset their password.	
Attributes	Attributes Name	Attributes Type

	email	String
Methods	Methods Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 4 Reset Password

2.1.1.5 User Homepage

Class Type	Boundary Class	
Responsibility	This interface shows up after the user login into the application.	
Attributes	Attributes Name	Attributes Type
	N/A	N/A
Methods	Methods Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 5 User Homepage

2.1.1.6 Admin homepage

Class Type	Boundary Class	
Responsibility	This interface shows up after the admin login into the application.	
Attributes	Attributes Name	Attributes Type
	N/A	N/A
Methods	Methods Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 6 Admin Homepage

2.1.1.1 Manage Registration Controller

Class Type	Controller Class	
Responsibility	To manage and process the registration data.	
Attributes	Attributes Name	Attributes Type
	email	String
	password	String
	user_ID	String
	username	String
Methods	Methods Name	Description
	login()	Allow the user and admin to login to the application.
	forgot_password()	Allow the user and admin to reset password.
	register()	Allow the user to register their account.
Algorithm	<pre> login() BEGIN IF username and password correct THEN redirect to User or Admin Homepage ELSE display error message END forgot_password() BEGIN Click forgot password? Redirect to Reset Password, fill in email and click "Send" END </pre>	

	<pre> register() BEGIN IF username missing input THEN display 'Name is required' error message Else IF email missing input THEN display 'Email is required' error message Else IF email already exist THEN display 'Email already in use' error message Else IF password missing input THEN display 'Password is required' error message Else send username, email and password to model. END </pre>
--	--

Table 2. 7 Manage Registration Controller

2.1.1.1 Manage Registration Model

Class Type	Model Class	
Responsibility	To store the registration data.	
Attributes	Attributes Name	Attributes Type
	email	String
	password	String
	user_ID	String
	username	String
Methods	Methods Name	Description
	login()	Allow the user and admin to login to the application.
	forgot_password()	Allow the user and admin to reset password.

	register()	Allow the user to register their account.
	search()	Allow the admin to search for a particular admin.
Algorithm	<pre> login() BEGIN IF username and password correct THEN redirect to User or Admin Homepage ELSE display error message END forgot_password() BEGIN Click forgot password? Redirect to Reset Password, fill in email and click "Send" END register() BEGIN IF username missing input THEN display 'Name is required' error message Else IF email missing input THEN display 'Email is required' error message Else IF email already exist THEN display 'Email already in use' error message Else IF password missing input THEN display 'Password is required' error message </pre>	

	Else send username, email and password to model. END
--	---

Table 2. 8 Manage Registration Model

2.1.2 Manage Chatting

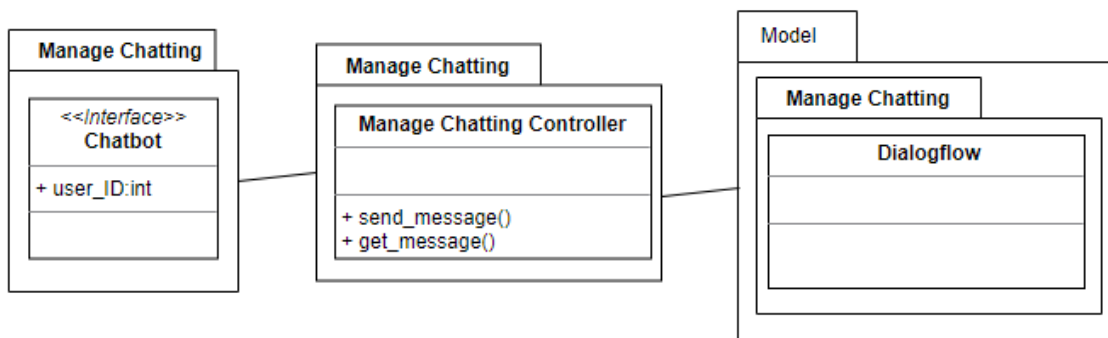


Figure 2. 2 Manage Chatting Module

2.1.2.1 Chatbot

Class Type	Boundary Class	
Responsibility	This interface allows the admin to add new admin.	
Attributes	Attributes Name	Attributes Type
	user_ID	String
Methods	Methods Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 9 Chatbot

2.1.2.2 Chatbot Controller

Class Type	Controller Class
------------	------------------

Responsibility	To manage and process the chatting data.	
Attributes	Attributes Name	Attributes Type
	N/A	N/A
Methods	Methods Name	Description
	send_message()	Allow the user to send message.
	get_message()	Allow the user to receive reply message.
Algorithm	<pre> send_message() BEGIN IF click send THEN send input message ELSE display error notification END get_message() BEGIN Chatbot understanding the message Reply to the message END </pre>	

Table 2. 10 Chatbot Controller

2.1.3 Manage Self-test

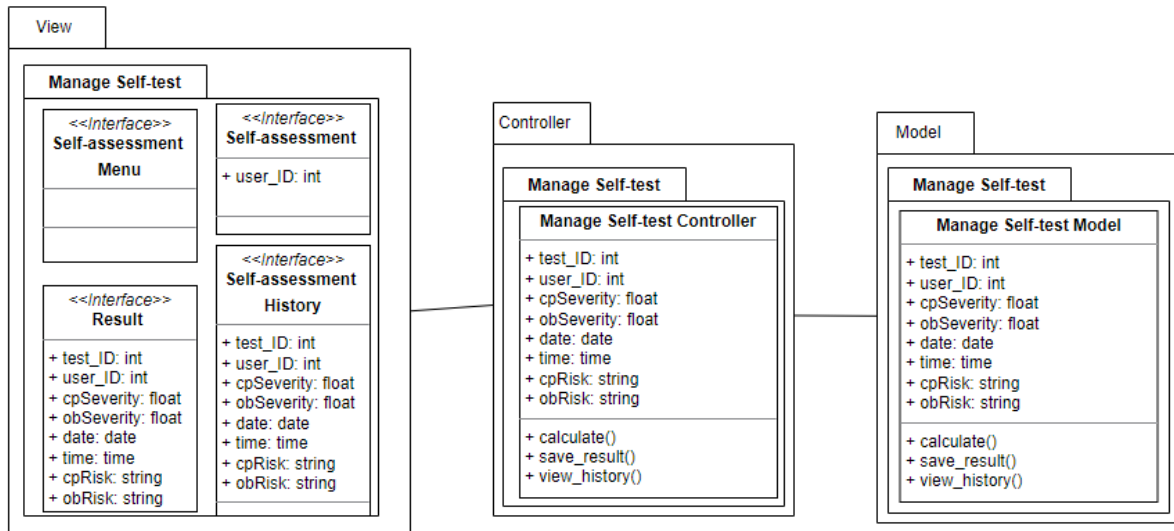


Figure 2. 3 Manage Self-test Module

2.1.3.1 Self-assessment Menu

Class Type	Boundary Class	
Responsibility	This interface allows the user to view the self-assessment menu.	
Attributes	Attributes Name	Attributes Type
	N/A	N/A
Methods	Methods Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 11 Self-assessment Menu

2.1.3.2 Self-assessment

Class Type	Boundary Class	
Responsibility	This interface allows the user to do the self-test on OCD.	
Attributes	Attributes Name	Attributes Type
	user_ID	String

Methods	Methods Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 12 Self-assessment

2.1.3.3 Result

Class Type	Boundary Class	
Responsibility	This interface allows the user to view their self-test result.	
Attributes	Attributes Name	Attributes Type
	test_ID	String
	user_ID	String
	cpSeverity	Float
	obSeverity	Float
	date	String
	time	String
	cpRisk	String
obRisk	String	
Methods	Methods Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 13 Result

2.1.3.4 Self-assessment History

Class Type	Boundary Class
Responsibility	This interface allows the user to view their self-assessment history.

Attributes	Attributes Name	Attributes Type
	test_ID	String
	user_ID	String
	cpSeverity	Float
	obSeverity	Float
	date	String
	time	String
	cpRisk	String
	obRisk	String
Methods	Methods Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 14 Self-assessment History

2.1.3.5 Self-test Controller

Class Type	Controller Class	
Responsibility	To manage and process the self-test data.	
Attributes	Attributes Name	Attributes Type
	test_ID	String
	user_ID	String
	cpSeverity	Float
	obSeverity	Float
	date	String
	time	String
	cpRisk	String

	obRisk	String
Methods	Methods Name	Description
	calculate()	To calculate the risk for the test of the user.
	save_result()	To save the test result.
	view_history()	Allow the user to view their test history.
Algorithm	<pre> calculate() BEGIN Receive all required input Calculate risk END save_result() BEGIN IF result calculated THEN save result to the self-test model ELSE display error notification END view_history() BEGIN IF click "View" button THEN get all result data Display the data ELSE display error notification END </pre>	

Table 2. 15 Self-test Controller

2.1.3.6 Self-test Model

Class Type	Model Class	
Responsibility	To store the self-test data.	
Attributes	Attributes Name	Attributes Type
	test_ID	String
	user_ID	String
	cpSeverity	Float
	obSeverity	Float
	date	String
	time	String
	cpRisk	String
	obRisk	String
Methods	Methods Name	Description
	calculate()	To calculate the risk for the test of the user.
	save_result()	To save the test result.
	view_history()	Allow the user to view their test history.
Algorithm	<pre> calculate() BEGIN Receive all required input Calculate risk END save_result() </pre>	

	<pre> BEGIN IF result calculated THEN save result to the self-test model ELSE display error notification END view_history() BEGIN IF click "View" button THEN get all result data Display the data ELSE display error notification END </pre>
--	--

Table 2. 16 Self-test Model

2.1.4 Manage Information

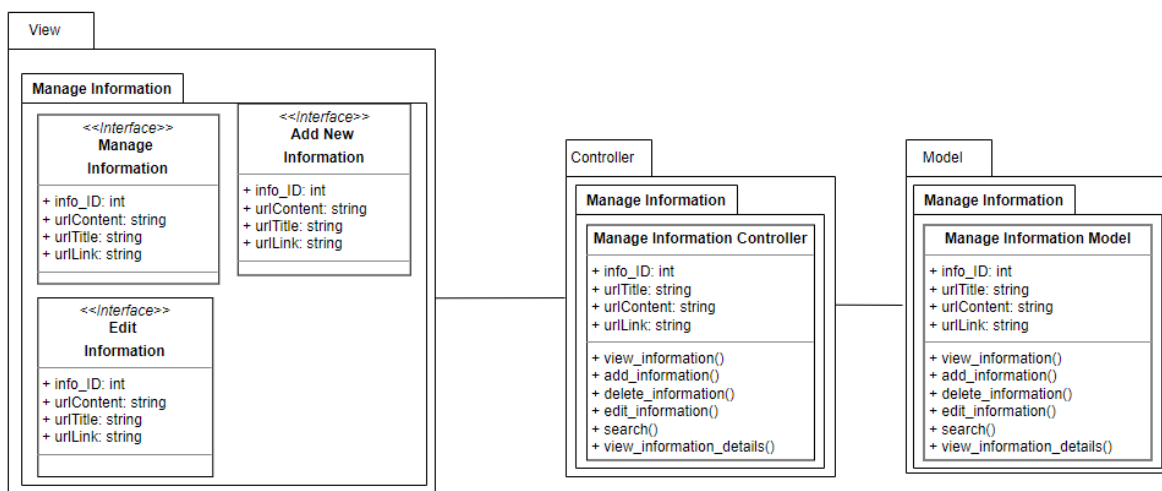


Figure 2. 4 Manage Information Module

2.1.4.1 Manage Information

Class Type	Boundary Class	
Responsibility	This interface allows the user and admin to view the information list.	
Attributes	Attributes Name	Attributes Type
	info_ID	String
	urlTitle	String
	urlContent	String
	urlLink	String
Methods	Methods Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 17 Manage Information

2.1.4.2 Add New Information

Class Type	Boundary Class	
Responsibility	This interface allows the admin to add new information.	
Attributes	Attributes Name	Attributes Type
	info_ID	String
	urlTitle	String
	urlContent	String
	urlLink	String
Methods	Methods Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 18 Add New Information

2.1.4.3 Edit Information

Class Type	Boundary Class	
Responsibility	This interface allows the admin to edit information.	
Attributes	Attributes Name	Attributes Type
	info_ID	String
	urlTitle	String
	urlContent	String
	urlLink	String
Methods	Methods Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 19 Edit Information

2.1.4.4 Manage Information Controller

Class Type	Controller Class	
Responsibility	To manage and process the information data.	
Attributes	Attributes Name	Attributes Type
	info_ID	String
	urlTitle	String
	urlContent	String
	urlLink	String
Methods	Methods Name	Description
	view_information()	Allow the user and the admin to view the information list.
	add_information()	Allow the admin to add new information.

	delete_information()	Allow the admin to delete information.
	edit_information()	Allow the admin to edit the information.
	search()	Allow the user and admin to search for a information.
	view_information_details()	Allow the user and admin to view the information website.
Algorithm	<pre> view_information() BEGIN IF click "Learn Now" THEN send request to retrieve data to model. Redirect to manage information interface ELSE display error notification END add_information() BEGIN IF click add information THEN send related data to model ELSE display error notification END update_information() BEGIN IF click edit THEN update related data to model </pre>	

	<pre> ELSE display error notification END delete_information() BEGIN If click delete THEN send info_ID to model and delete data ELSE display error notification END search() BEGIN IF enter keyword and click search THEN pass keyword to model ELSE display error notification END view_information_details() BEGIN Click view button Redirect to website END </pre>
--	--

Table 2. 20 Manage Information Controller

2.1.4.5 Manage Information Model

Class Type	Controller Class
------------	------------------

Responsibility	To store the information data.	
Attributes	Attributes Name	Attributes Type
	info_ID	String
	urlTitle	String
	urlContent	String
	urlLink	String
Methods	Methods Name	Description
	view_information()	Allow the user and the admin to view the information.
	add_information()	Allow the admin to add new information.
	delete_information()	Allow the admin to delete information.
	edit_information()	Allow the admin to edit the information.
	search()	Allow the user and admin to search for a information.
	view_information_details()	Allow the user and admin to view the information website.
Algorithm	<pre> view_information() BEGIN IF click "Learn Now" THEN send request to retrieve data to model. Redirect to manage information interface ELSE display error notification END </pre>	

```
add_information()
BEGIN
IF click add information
    THEN send related data to model
ELSE display error notification
END

update_information()
BEGIN
IF click edit
    THEN update related data to model
ELSE display error notification
END

delete_information()
BEGIN
If click delete
    THEN send info_ID to model and delete data
ELSE display error notification
END

search()
BEGIN
IF enter keyword and click search
    THEN pass keyword to model
ELSE display error notification
```

	<p>END</p> <p>view_information_details()</p> <p>BEGIN</p> <p>Click view button</p> <p>Redirect to website</p> <p>END</p>
--	--

Table 2. 21 Manage Information Model

2.1.5 Manage Diary

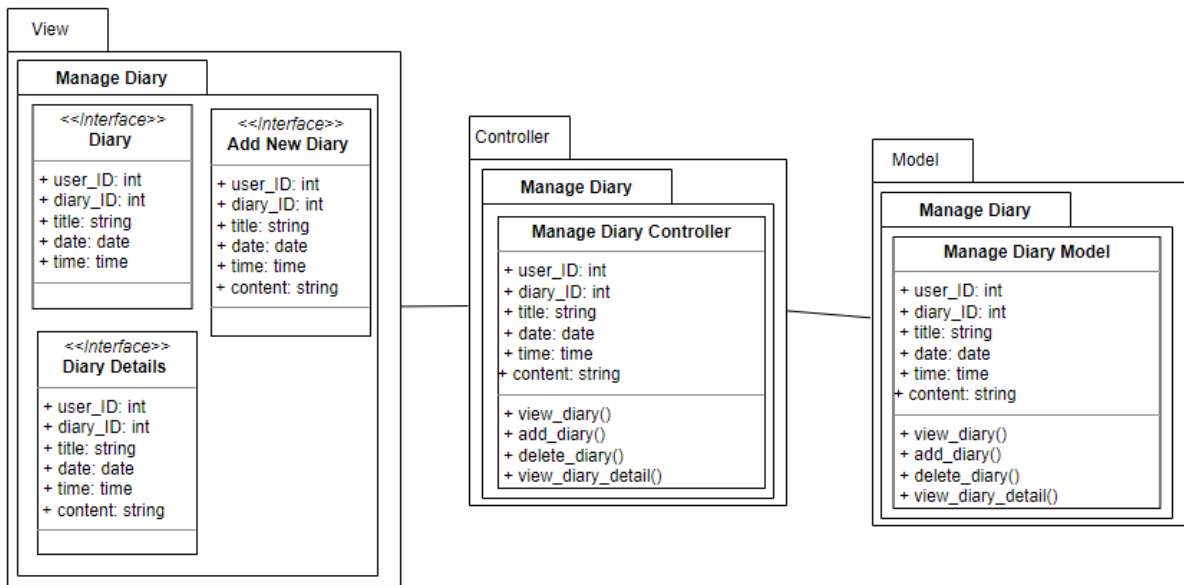


Figure 2. 5 Manage Diary Module

2.1.5.1 Diary

Class Type	Boundary Class	
Responsibility	This interface allows the user to view their diary list.	
Attributes	Attributes Name	Attributes Type

	user_ID	String
	diary_ID	String
	title	String
	date	Date
	time	Time
Methods	Methods Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 22 Diary

2.1.5.2 Add New Diary

Class Type	Boundary Class	
Responsibility	This interface allows the user to add new diary.	
Attributes	Attributes Name	Attributes Type
	user_ID	String
	diary_ID	String
	title	String
	date	Date
	time	Time
	content	String
Methods	Methods Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 23 Add New Diary

2.1.5.3 Diary Details

Class Type	Boundary Class	
Responsibility	This interface allows the user to view their diary details.	
Attributes	Attributes Name	Attributes Type
	user_ID	String
	diary_ID	String
	title	String
	date	Date
	time	Time
	content	String
Methods	Methods Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 24 Diary Details

2.1.5.4 Manage Diary Controller

Class Type	Controller Class	
Responsibility	To manage and process the information data.	
Attributes	Attributes Name	Attributes Type
	user_ID	String
	diary_ID	String
	title	String
	date	Date
	time	Time
	content	String
Methods	Methods Name	Description

	view_diary()	Allow the user to view their diary list.
	add_diary()	Allow the user to add new diary.
	delete_diary()	Allow the user to delete diary.
	view_diary_detail()	Allow the user to view their diary details.
Algorithm	<pre> view_diary() BEGIN IF click "Diary" icon THEN send request to retrieve data to model. Redirect to diary interface ELSE display error notification END add_diary() BEGIN IF click add diary THEN send related data to model ELSE display error notification END delete_diary() BEGIN If click delete THEN send diary_ID to model and delete data ELSE display error notification </pre>	

	<p>END</p> <p>view_diary_details()</p> <p>BEGIN</p> <p>Click view button</p> <p>Send request to retrieve data to model</p> <p>END</p>
--	---

Table 2. 25 Manage Diary Controller

2.1.5.5 Manage Diary Model

Class Type	Controller Class	
Responsibility	To store the information data.	
Attributes	Attributes Name	Attributes Type
	user_ID	String
	diary_ID	String
	title	String
	date	Date
	time	Time
	content	String
Methods	Methods Name	Description
	view_diary()	Allow the user to view their diary list.
	add_diary()	Allow the user to add new diary.
	delete_diary()	Allow the user to delete diary.
	view_diary_detail()	Allow the user to view their diary details.

Algorithm	<pre>view_diary() BEGIN IF click "Diary" icon THEN send request to retrieve data to model. Redirect to diary interface ELSE display error notification END add_diary() BEGIN IF click add diary THEN send related data to model ELSE display error notification END delete_diary() BEGIN If click delete THEN send diary_ID to model and delete data ELSE display error notification END view_diary_details() BEGIN Click view button Send request to retrieve data to model</pre>
-----------	---

	END
--	-----

Table 2. 26 Manage Diary Model

2.1.6 Manage Profile

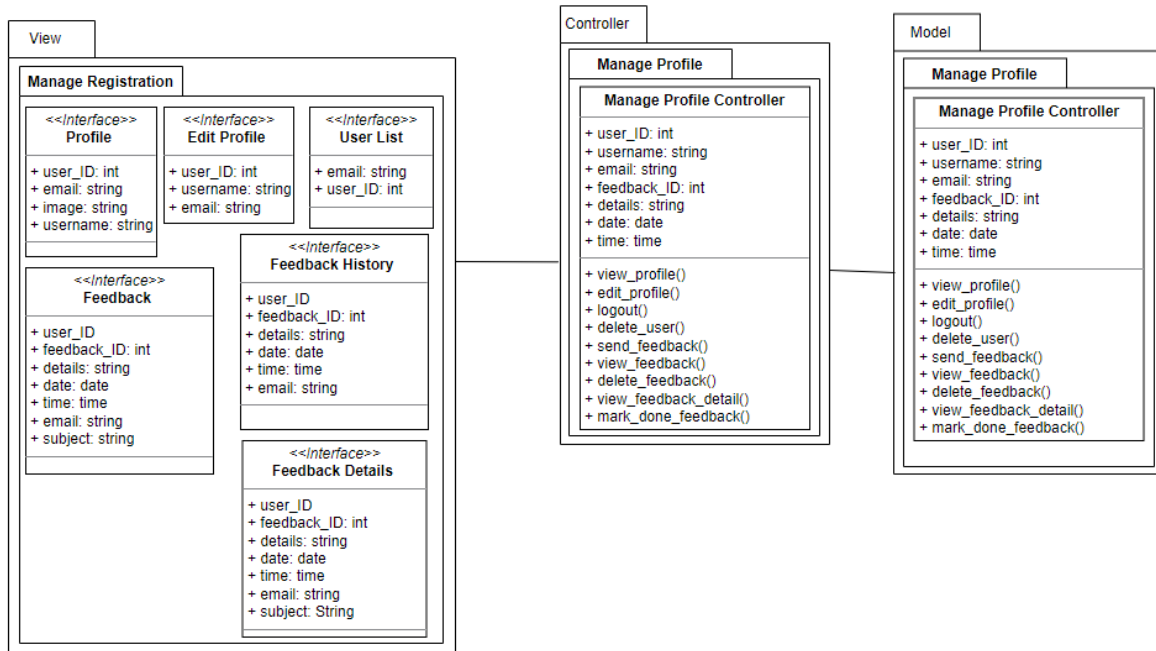


Figure 2. 6 Manage Profile Module

2.1.6.1 Profile

Class Type	Boundary Class	
Responsibility	This interface allows the user and admin to view their profile.	
Attributes	Attributes Name	Attributes Type
	user_ID	String
	email	String
	image	String
	username	String
Methods	Methods Name	Description

	N/A	N/A
Algorithm	N/A	

Table 2. 27 Profile

2.1.6.2 Edit Profile

Class Type	Boundary Class	
Responsibility	This interface allows the user and admin to edit their profile.	
Attributes	Attributes Name	Attributes Type
	user_ID	String
	username	String
	email	String
Methods	Methods Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 28 Edit Profile

2.1.6.3 Feedback

Class Type	Boundary Class	
Responsibility	This interface allows the user to make feedback and the admin to view the feedback list.	
Attributes	Attributes Name	Attributes Type
	user_ID	String
	feedback_ID	String
	details	String
	date	Date
	time	Time
	email	String

	subject	String
Methods	Methods Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 29 Feedback

2.1.6.4 Feedback details

Class Type	Boundary Class	
Responsibility	This interface allows the admin to view the feedback details.	
Attributes	Attributes Name	Attributes Type
	user_ID	String
	feedback_ID	String
	details	String
	date	Date
	time	Time
	email	String
	subject	String
Methods	Methods Name	Description
	N/A	N/A
Algorithm	N/A	

Table 2. 30 Feedback Details

2.1.6.5 Manage Profile Controller

Class Type	Controller Class	
Responsibility	To manage and process the profile data.	
Attributes	Attributes Name	Attributes Type

	user_ID	String
	feedback_ID	String
	details	String
	date	Date
	time	Time
	email	String
	subject	String
Methods	Methods Name	Description
	view_profile()	Allow the user and admin to view their profile.
	edit_profile()	Allow the user and admin to edit their profile.
	logout()	Allow the user and admin logout the application.
	delete_account()	Allow the user and admin to delete their account.
	send_feedback()	Allow the user to send feedback.
	view_feedback()	Allow the admin to view feedback list.
	delete_feedback()	Allow the admin to delete the feedback.
	view_feedback_detail()	Allow the admin to view feedback details.
	Mark_done_feedback()	Allow the admin to mark the feedback done
Algorithm	view_profile() BEGIN Retrieve account data based on acc_ID.	

END

edit_profile()

BEGIN

Update account data based on acc_ID

END

logout()

BEGIN

Logout account

END

delete_account()

BEGIN

Delete account data based on acc_ID

END

send_feedback()

BEGIN

Save feedback.

END

view_feedback()

BEGIN

Retrieve feedback data

Display data

END

	<pre> delete_feedback() BEGIN Delete feedback data based on feedback_ID END view_feedback_detail() BEGIN Retrieve feedback data based on feedback_ID Display data. END Mark_feedback_done() BEGIN Click "Mark Done" button Data save to database END </pre>
--	--

Table 2. 31 Manage Profile Controller

2.1.6.6 Manage Profile Model

Class Type	Controller Class	
Responsibility	To store the profile data.	
Attributes	Attributes Name	Attributes Type
	acc_ID	Int
	username	String
	email	String
	feedback_ID	String

	feedback_detail	String
	date	Date
	time	Time
	feedback_email	String
Methods	Methods Name	Description
	view_profile()	Allow the user and admin to view their profile.
	edit_profile()	Allow the user and admin to edit their profile.
	logout()	Allow the user and admin logout the application.
	delete_account()	Allow the user and admin to delete their account.
	send_feedback()	Allow the user to send feedback.
	view_feedback()	Allow the admin to view feedback list.
	delete_feedback()	Allow the admin to delete the feedback.
	view_feedback_detail()	Allow the admin to view feedback details.
		Mark_done_feedback()
Algorithm	<pre> view_profile() BEGIN Retrieve account data based on acc_ID. END edit_profile() BEGIN </pre>	

```
Update account data based on acc_ID
```

```
END
```

```
logout()
```

```
BEGIN
```

```
Logout account
```

```
END
```

```
delete_account()
```

```
BEGIN
```

```
Delete account data based on acc_ID
```

```
END
```

```
send_feedback()
```

```
BEGIN
```

```
Save feedback.
```

```
END
```

```
view_feedback()
```

```
BEGIN
```

```
Retrieve feedback data
```

```
Display data
```

```
END
```

```
delete_feedback()
```

```
BEGIN
```

```
Delete feedback data based on feedback_ID
```

```
END
```

	<pre> view_feedback_detail() BEGIN Retrieve feedback data based on feedback_ID Display data. END Mark_feedback_done() BEGIN Click "Mark Done" button Data save to database END </pre>
--	--

Table 2. 32 Manage Profile Model

2.2 DATA DICTIONARY

Users

Field Name	Description	Data Type	Constraint
user_ID	User account ID	VARCHAR(255)	PK, NOT NULL
email	User email	VARCHAR(255)	
password	Account password	VARCHAR(255)	
username	Username of user	VARCHAR(255)	

Table 2. 33 Profile

Information

Field Name	Description	Data Type	Constraint

info_ID	Information ID	VARCHAR(255)	PK, NOT NULL
urlContent	Information title	VARCHAR(255)	
urlLink	URL link for the information	VARCHAR(255)	
urlTitle	Description of the information	VARCHAR(255)	

Table 2. 34 Information

Feedback

Field Name	Description	Data Type	Constraint
feedback_ID	Feedback ID	VARCHAR(255)	PK, NOT NULL
details	Feedback detail	VARCHAR(255)	
date	Date when user send the feedback	DATE	
time	Time when user send the feedback	TIME	
email	Email address that user fill for further contact	VARCHAR(255)	
subject	Feedback subject	VARCHAR(255)	

Table 2. 35 Feedback

Diary

Field Name	Description	Data Type	Constraint
diary_ID	Diary ID	VARCHAR(255)	PK, NOT NULL
user_ID	Account ID	VARCHAR(255)	FK

title	Diary title	VARCHAR(255)	
content	Diary detail	VARCHAR(255)	
date	Date when user save the diary	DATE	
time	Time when user save the diary	TIME	

Table 2. 36 Diary

Test

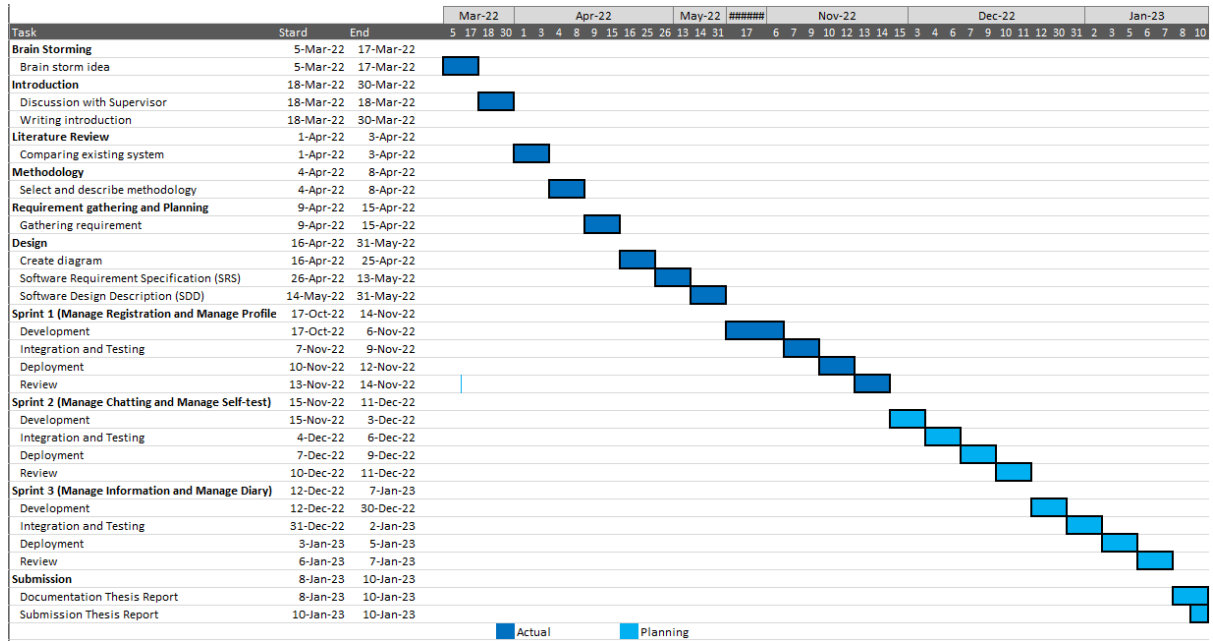
Field Name	Description	Data Type	Constraint
test_ID	History ID	VARCHAR(255)	PK, NOT NULL
user_ID	Account ID	VARCHAR(255)	FK
cpSeverity	The total score for compulsive.	FLOAT	
obSeverity	The total score for obsessive.	FLOAT	
cpRisk	The category for the calculated result score for compulsive.	VARCHAR(255)	
obRisk	The category for the calculated result score for obsessive.	VARCHAR(255)	
date	Date when user submit the test	VARCHAR(255)	
time	Time when user submit the test	VARCHAR(255)	

Table 2. 37 History

APPENDIX C
GANTT CHART

Gantt Chart

No	Task	Duration(Day)	Start Date	End Date
	Brain Storming	13	5/3/2022	17/3/2022
1	Brain storm idea	13	5/3/2022	17/3/2022
	Introduction	13	18/3/2022	30/3/2022
2	Discussion with Supervisor	1	18/3/2022	18/3/2022
3	Writing introduction	13	18/3/2022	30/3/2022
	Literature Review	3	1/4/2022	3/4/2022
4	Comparing existing system	3	1/4/2022	3/4/2022
	Methodology	5	4/4/2022	8/4/2022
5	Select and describe methodology	5	4/4/2022	8/4/2022
	Requirement gathering and Planning	7	9/4/2022	15/4/2022
6	Gathering requirement	7	9/4/2022	15/4/2022
	Design	46	16/4/2022	31/5/2022
7	Create diagram	10	16/4/2022	25/4/2022
8	Software Requirement Specification (SRS)	18	26/4/2022	13/5/2022
9	Software Design Description (SDD)	18	14/5/2022	31/5/2022
	Sprint 1 (Manage Registration and Manage Profile)	29	17/10/2022	14/11/2022
10	Development	21	17/10/2022	6/11/2022
11	Integration and Testing	3	7/11/2022	9/11/2022
12	Deployment	3	10/11/2022	12/11/2022
13	Review	2	13/11/2022	14/11/2022
	Sprint 2 (Manage Chatting and Manage Self-test)	27	15/11/2022	11/12/2022
14	Development	19	15/11/2022	3/12/2022
15	Integration and Testing	3	4/12/2022	6/12/2022
16	Deployment	3	7/12/2022	9/12/2022
17	Review	2	10/12/2022	11/12/2022
	Sprint 3 (Manage Information and Manage Diary)	27	12/12/2022	7/1/2023
18	Development	19	12/12/2022	30/12/2022
19	Integration and Testing	3	31/12/2022	2/1/2023
20	Deployment	3	3/1/2023	5/1/2023
21	Review	2	6/1/2023	7/1/2023
	Submission	2	8/1/2023	10/1/2023
22	Documentation Thesis Report	2	8/1/2023	10/1/2023
23	Submission Thesis Report	1	10/1/2023	10/1/2023



**APPENDIX D
TEST REPORT**

2023

USER ACCEPTANCE TEST (UAT)

[KIKI: A CHATBOT APPLICATION FOR OCD
IN ANDROID MOBILE APPLICATION]



DOCUMENT APPROVAL

	Name	Date
<p>Authenticated by:</p> <p>_____</p> <p>Name</p>		
<p>Approved by:</p> <p>_____</p> <p>Client</p>		

Software :

Archiving Place :

TABLE OF CONTENT

CONTENT	PAGE
DOCUMENT APPROVAL	ii
TABLE OF CONTENT.....	iii
TEST REPORT.....	1

TEST REPORT

The UAT testing is conducted to ensure the developed system is acceptable by the user and conform the requirement stated in Software Requirement Specification (SRS). This testing is conducted after the functional of the system testing is done.

The end user and client are involved to participate in this testing. This testing will take place after the unit testing, integration testing and system testing.

Table 1.1: Registration Function Testing

Module	Manage Registration				
Objective	To test registration function of the system.				
Test ID	Test Cases	Test Data	Actual Result	Expected Result	Test Status
TC-01-01	Register user with blank input	No input	Display error message	Display error message	PASS
TC-01-02	Register user with existing email	Existing email registered	Display error message	Display error message	PASS
TC-01-03	Register user with invalid password format	“1234” “1234”	Display error message	Display error message	PASS
TC-01-04	Register user with different password confirmation.	“123456” “654321”	Display error message	Display error message	PASS

TC-01-05	Register user with invalid username	“abc”	Display error message	Display error message	PASS
TC-01-06	Register user with valid email, username, password and password confirmation	“kiki@gmail.com” “Kiki” “123456” “123456”	Register successfully	Register successfully	PASS

Table 1.2: Login Function Testing

Module	Manage Registration				
Objective	To test login function of the system.				
Test ID	Test Cases	Test Data	Actual Result	Expected Result	Test Status
TC-02-01	Login with blank email input	No input	Display error message	Display error message	PASS
TC-02-02	Login with email not registered	Email not registered	Display error message	Display error message	PASS
TC-02-03	Login with blank password input	“kiki@gmail.com” “”	Display error message	Display error message	PASS

TC-02-04	Login with wrong password	“kiki@gmail.com” “456123”	Display error message	Display error message	PASS
TC-02-05	Login with valid email and password	“kiki@gmail.com” “123456”	Login successfully	Login successfully	PASS

Table 1.3: Forget Password Testing

Module	Manage Registration				
Objective	To test reset password function.				
Test ID	Test Cases	Test Data	Actual Result	Expected Result	Test Status
TC-03-01	Reset password with blank email input	No input	Display error message	Display error message	PASS
TC-03-02	Reset password with invalid email input	“abc”	Display error message	Display error message	PASS
TC-03-03	Reset password with valid email input	“kiki@gmail.com”	Reset password email sent	Reset password email sent	PASS

Table 1.4: Profile Function Testing

Module	Manage Profile				
Objective	To test the profile function.				
Test ID	Test Cases	Test Data	Actual Result	Expected Result	Test Status
TC-04-01	Display username and email correctly after login.	No input	Username and email show correctly	Username and email show correctly	PASS
TC-04-02	Display username and email correctly after clicking on <<Edit Profile>> button.	No input	Username and email correctly show after clicking on <<Edit Profile>> button.	Username and email correctly show after clicking on <<Edit Profile>> button.	PASS
TC-04-03	Enter invalid username	“kk”	Show error message	Show error message	PASS
TC-04-04	Enter valid username	“kiki”	Username changed successfully	Username changed successfully	PASS
TC-04-05	Click on <<Log Out>> button	No input	Logout successfully	Logout successfully	PASS

Table 1.5: Feedback Function Testing

Module	Manage Profile				
Objective	To test the feedback function.				
Test ID	Test Cases	Test Data	Actual Result	Expected Result	Test Status
TC-05-01	Submit with null selection for subject.	No input	Display error message	Display error message	PASS
TC-05-02	Submit with empty details.	No input	Display error message	Display error message	PASS
TC-05-03	Submit details exceeding 500 characters.	501 characters	Display error message	Display error message	PASS
TC-05-04	Submit with selection for subject and valid details	“Chatbot” “There is a bug.”	Feedback sent	Feedback sent	PASS
TC-05-05	Show all feedback that have not marked done	No input	Feedback show	Feedback show	PASS
TC-05-06	Show all feedback that marked done	No input	Feedback show	Feedback show	PASS
TC-05-07	Show the details of	No input	Feedback details show	Feedback details show	PASS

	each feedback				
TC-05-08	Search feedback by subject	“Chatbot”	Search result show	Search result show	PASS

Table 1.6: User List Function Testing

Module	Manage Profile				
Objective	To test the user list function				
Test ID	Test Cases	Test Data	Actual Result	Expected Result	Test Status
TC-06-01	User List for all user display	No input	User List displayed	User List displayed	PASS
TC-06-02	User deleted while click on <<Delete>> button	No input	User deleted	User deleted	PASS
TC-06-03	Search user by email	“kong”	Search result show	Search result show	PASS

Table 1.7: Registration Function Testing

Module	Manage Diary
Objective	To test the diary function.

Test ID	Test Cases	Test Data	Actual Result	Expected Result	Test Status
TC-07-01	List all the diary record when click on the diary tab	No input	Diary record show	Diary record show	PASS
TC-07-02	Add new diary with null title or content.	No input	Display error message	Display error message	PASS
TC-07-03	Add new diary with title over 200 character or content over 500 characters.	200 character and 500 characters	Display error message	Display error message	PASS
TC-07-04	Add new diary with valid title and content	“Feels good today” “Good thing happen today”	Diary added successfully	Diary added successfully	PASS
TC-07-05	View diary details	No input	Diary details show correctly	Diary details show correctly	PASS
TC-07-06	Edit diary details with	No input	Display error message	Display error message	PASS

	null title or content				
TC-07-07	Edit Diary with valid title and content	“Feels good today” “Good thing happen today”	Diary edited successfully	Diary edited successfully	PASS
TC-07-08	Delete Diary when click on <<Delete>> button	No input	Diary deleted	Diary deleted	PASS

Table 1.8: Information Function Testing

Module	Manage Information				
Objective	To test the information function.				
Test ID	Test Cases	Test Data	Actual Result	Expected Result	Test Status
TC-08-01	Information listed when clicking into the inforamtion interface.	No input	Information showed	Information showed	PASS
TC-08-02	Add new information with null	No input	Display error message	Display error message	PASS

	link, title or content.				
TC-08-03	Add new information with valid title and content	“OCD” “Symptoms of OCD”	Information added successfully	Information added successfully	PASS
TC-08-04	Edit information details with null title or content	No input	Display error message	Display error message	PASS
TC-08-05	Edit information with valid title and content	“Feels good today” “Good thing happen today”	Information edited successfully	Information edited successfully	PASS
TC-08-06	Delete information when click on <<Delete>> button	No input	Information deleted	Information deleted	PASS
TC-08-07	Redirect to the website when click on the “View”	No input	Redirected to website	Redirected to website	PASS

TC-08-08	Add new information with invalid link, title or content.	“abc” “” “”	Display error message	Display error message	PASS
TC-08-09	Search information by title	“Overview”	Search result show	Search result show	PASS

Table 1.9: Chatting Function Testing

Module	Manage Chatting				
Objective	To test the chatting function.				
Test ID	Test Cases	Test Data	Actual Result	Expected Result	Test Status
TC-09-01	User type message and send	“Hi”	Chatbot show reply	Chatbot show reply	PASS
TC-09-02	User type question scripted in script	“OCD”	Chatbot show reply	Chatbot show reply	PASS
TC-09-03	User send with null input	No input	“Please enter text” message show	“Please enter text” message show	PASS
TC-09-04	User click on the suggestion	No input	Chatbot show reply	Chatbot show reply	PASS

	bubble option				
TC-09-05	User type question not scripted in script	“Mental health”	Chatbot show not understand reply	Chatbot show not understand reply	PASS

Table 1.10: Self-Test Function Testing

Module	Manage Self-Test				
Objective	To test the self-test function.				
Test ID	Test Cases	Test Data	Actual Result	Expected Result	Test Status
TC-10-01	OCD test question display to user in Self-Assessment Test interface.	No input	Question show	Question show	PASS
TC-10-02	Self-Assessment Result show accurately after user submit the test.	Answer for each question	Self-assessment result show	Self-assessment result show	PASS

TC-10-03	Self-assessment history show in the Self-Assessment History interface.	No input	Self-assessment history show	Self-assessment history show	PASS
----------	--	----------	------------------------------	------------------------------	------

UAT TEST RESULT

Display error message when register user with blank input

15 responses



● PASS
● FAIL

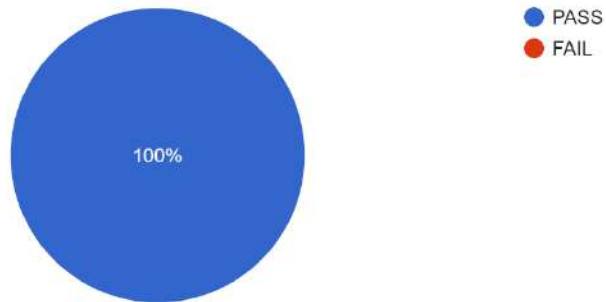
Display error message when register user with existing email

15 responses

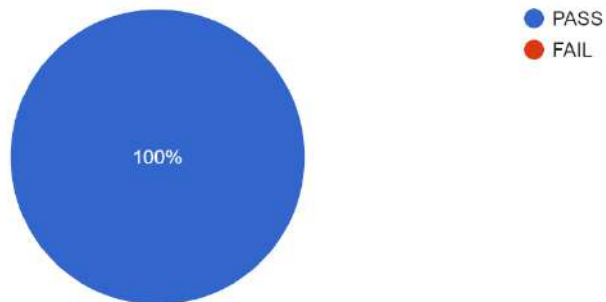


● PASS
● FAIL

Display error message when register user with invalid password format
 15 responses



Display error message when register user with different password confirmation.
 15 responses

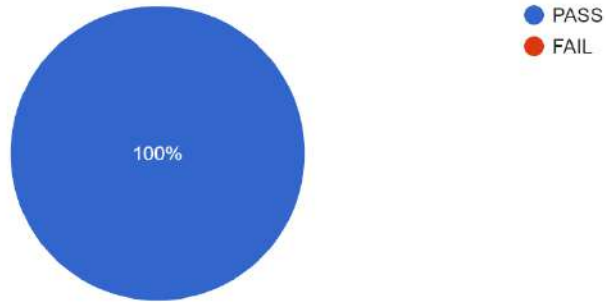


Display error message when register user with invalid username
 15 responses



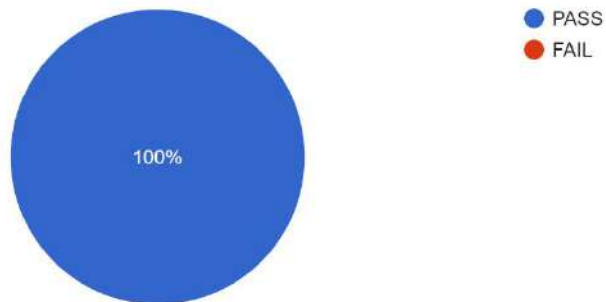
Register successfully with valid email, username, password and password confirmation.

15 responses



Display error message when login with blank email input

15 responses



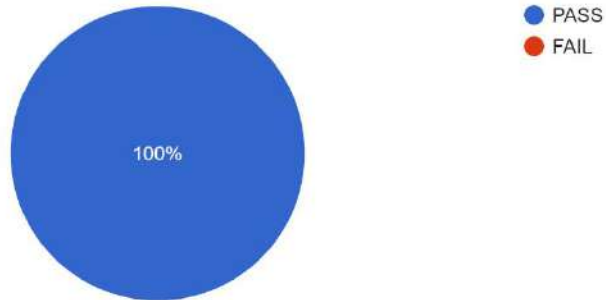
Display error message when login with email not registered

15 responses



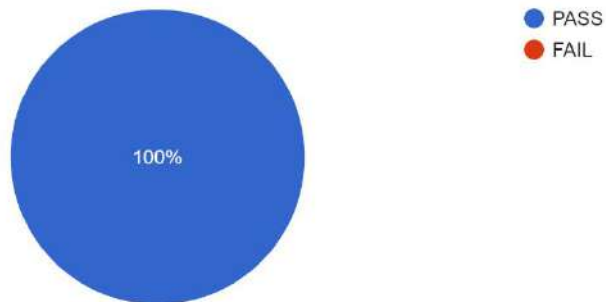
Display error message when login with blank password input

15 responses



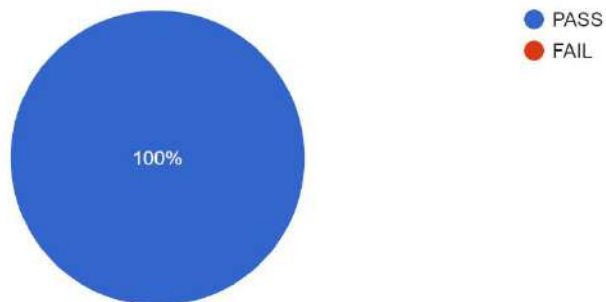
Display error message when login with wrong password

15 responses



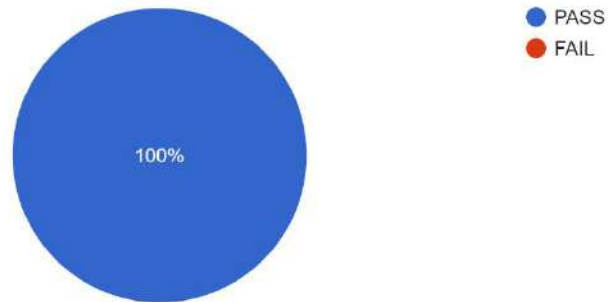
Display error message when login with valid email and password

15 responses



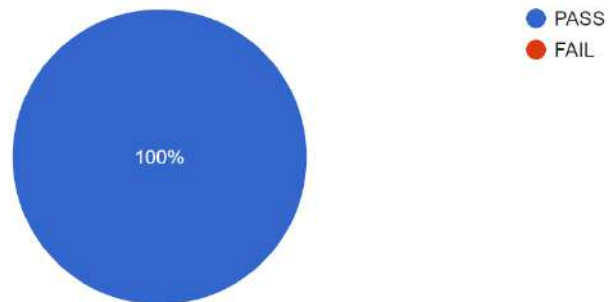
Login successfully when login with valid email and password

15 responses



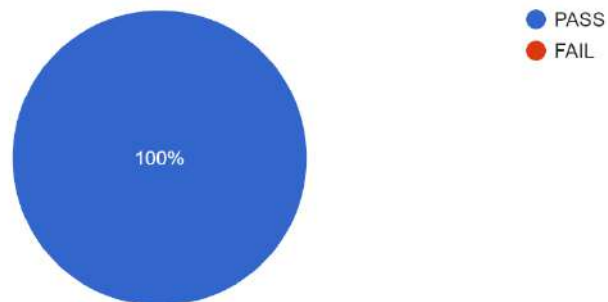
Display error message when reset password with blank email input

15 responses



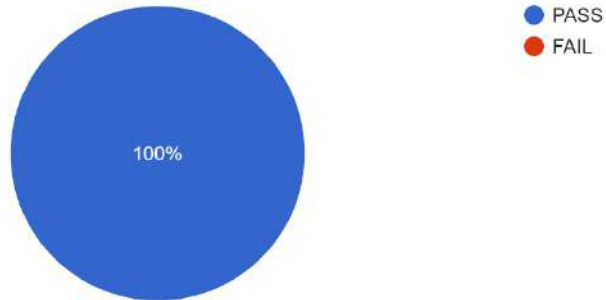
Display error message when reset password with invalid email input

15 responses



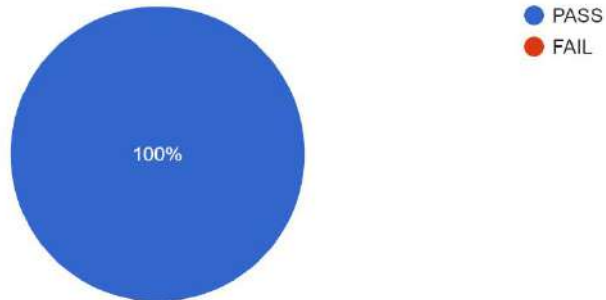
Reset password successfully when reset password with valid email input

15 responses



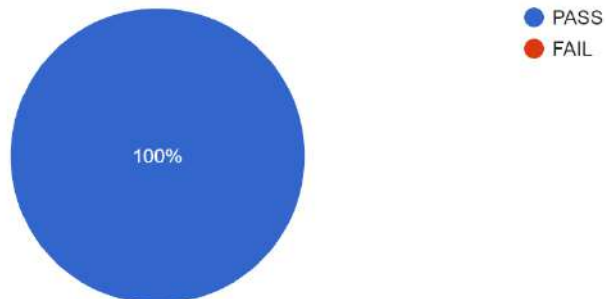
Display username and email correctly after login.

15 responses



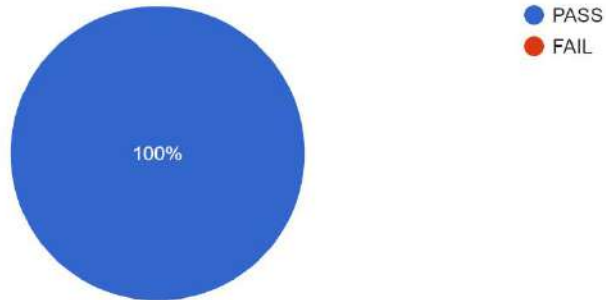
Display username and email correctly after clicking on <<Edit Profile>> button.

15 responses



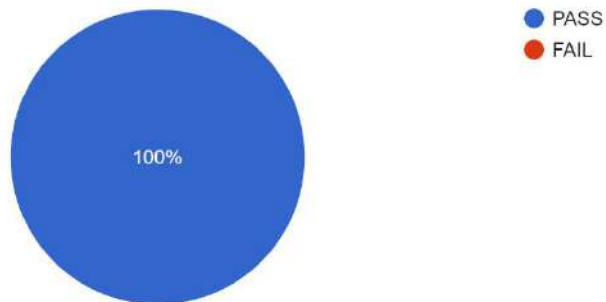
Display error message when enter invalid username

15 responses



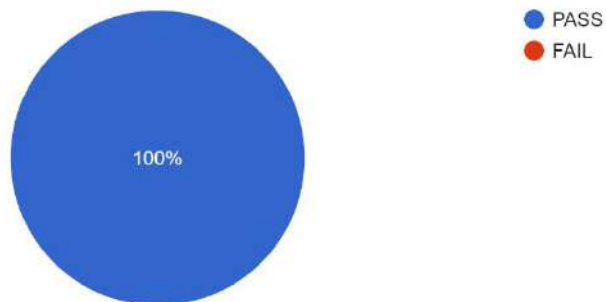
Username changed successfully when enter invalid username

15 responses

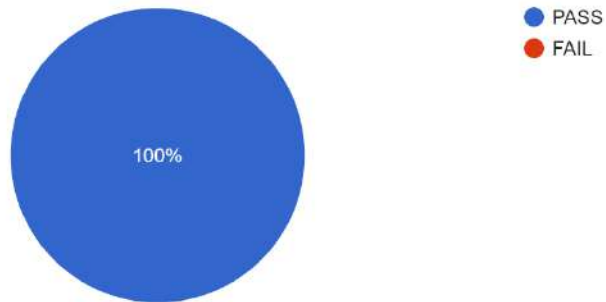


Logout successfully when click on <<Log Out>> button

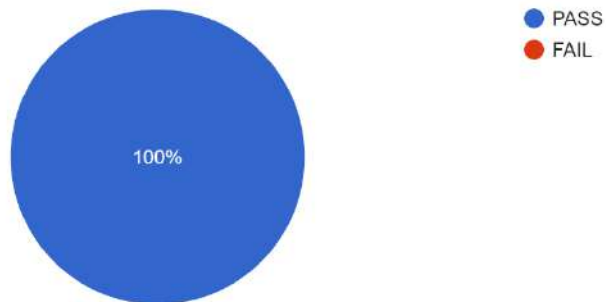
15 responses



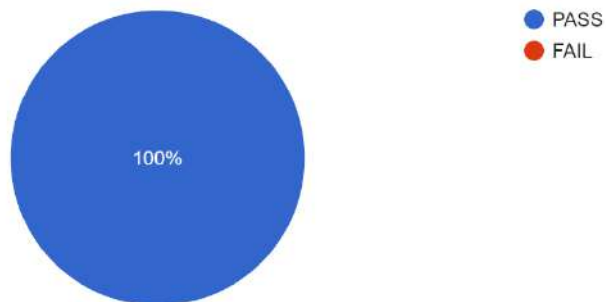
Diaplay error message when submit with null selection for subject in feedback.
15 responses



Diaplay error message when submit with empty details in feedback.
15 responses

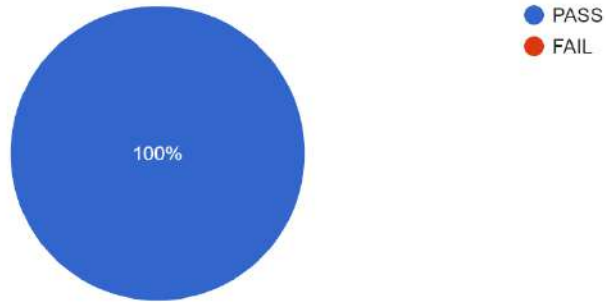


Diaplay error message when details exceeding 500 characters in feedback.
15 responses



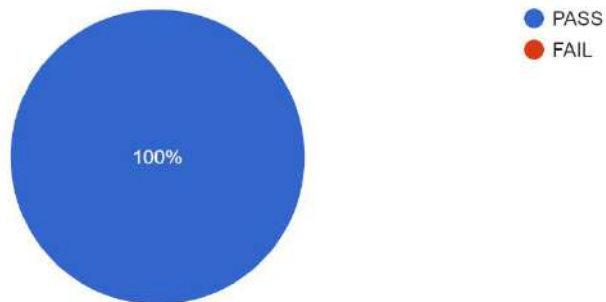
Submit successfully when submit with selection for subject and valid details.

15 responses



Show all feedback that have not marked done

15 responses



Show all feedback that marked done

15 responses



Show the details of each feedback

15 responses



● PASS
● FAIL

Show searched feedback by subject.

15 responses



● PASS
● FAIL

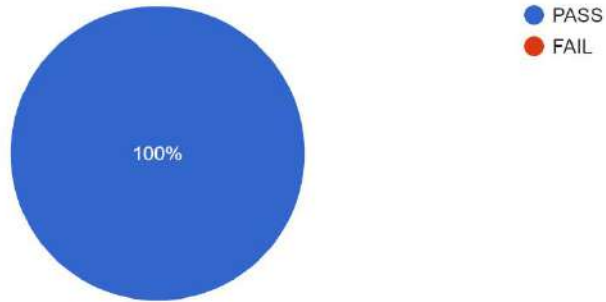
Display user List for all user.

15 responses

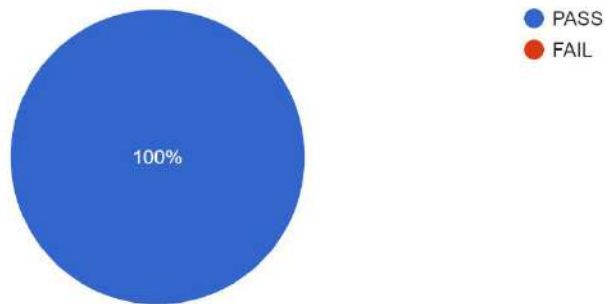


● PASS
● FAIL

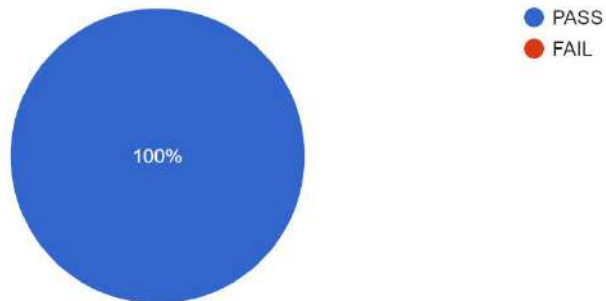
User deleted while click on <<Delete>> button.
15 responses



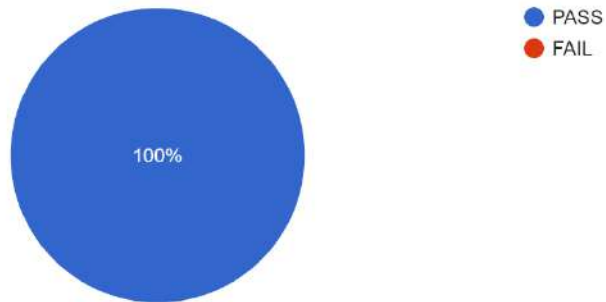
Show searched user by email
15 responses



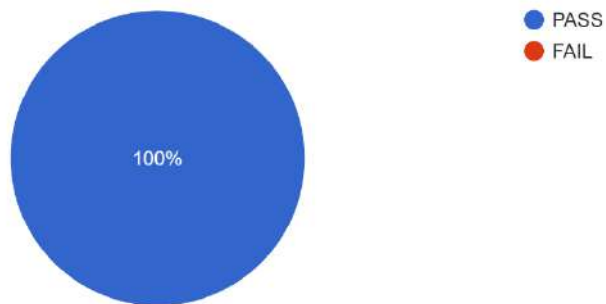
List all the diary record when click on the diary tab
15 responses



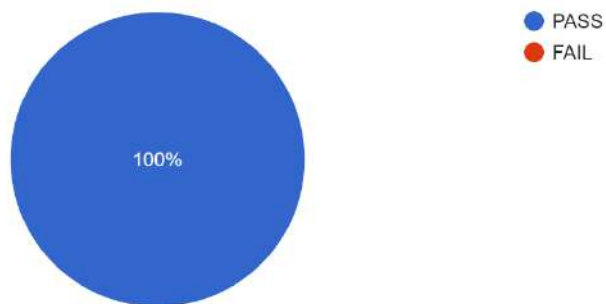
Display error message when add new diary with null title or content.
15 responses



Display error message when add new diary with title over 200 character or content over 500 characters.
15 responses

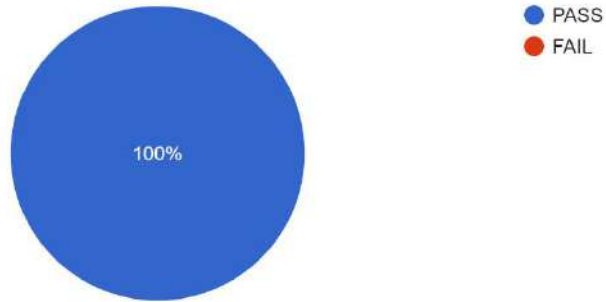


Diary added succesfully when add new diary with valid title and content
15 responses



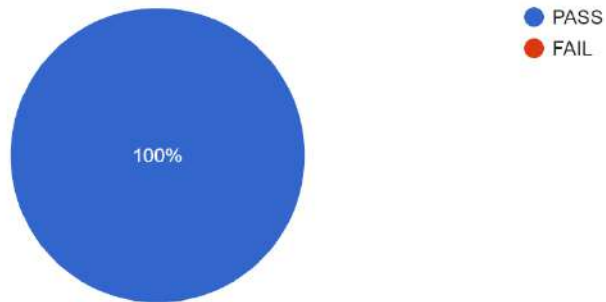
Diary details show correctly when view diary details

15 responses



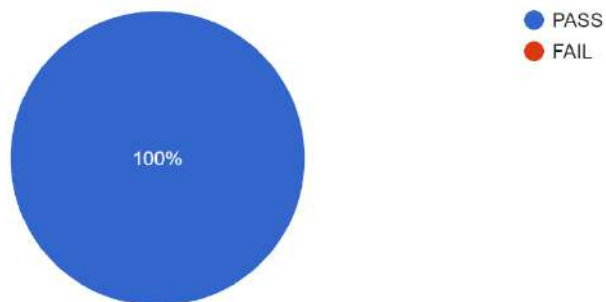
Display error message when edit diary details with null title or content

15 responses



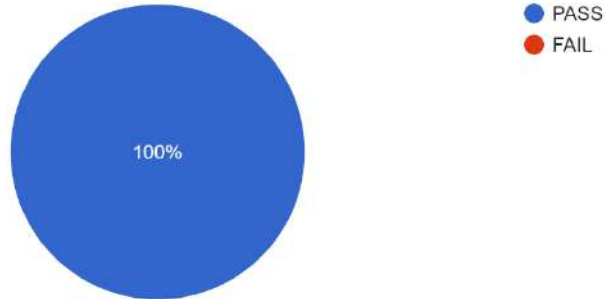
Diary edited successfully when edit Diary with valid title and content

15 responses



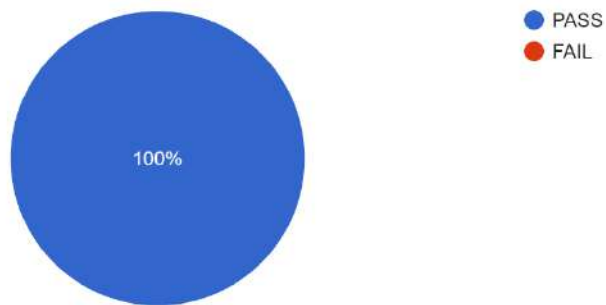
Diary deleted when delete Diary when click on <<Delete>> button

15 responses



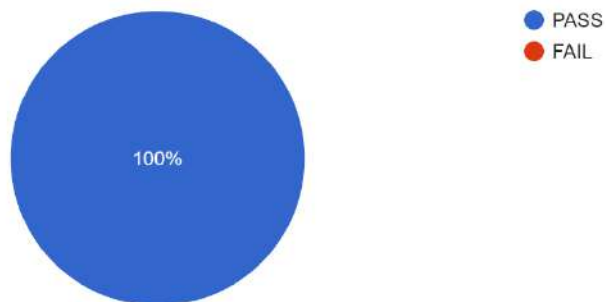
Information listed when clicking into the information interface.

15 responses



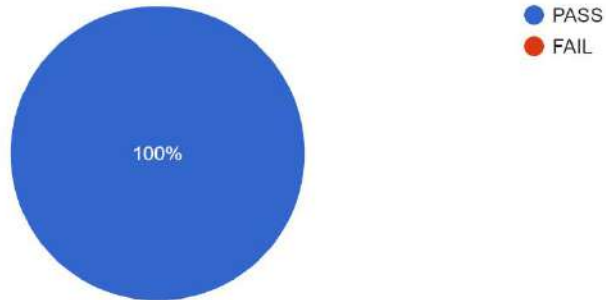
Display error message when add new information with null link, title or content.

15 responses



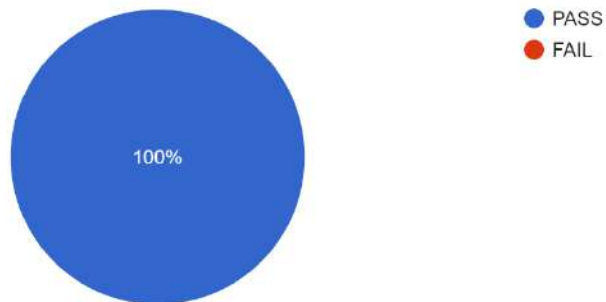
Information added successfully with valid title and content

15 responses



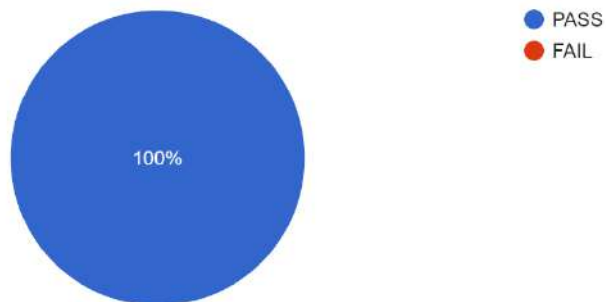
Display error message when edit information details with null title or content

15 responses

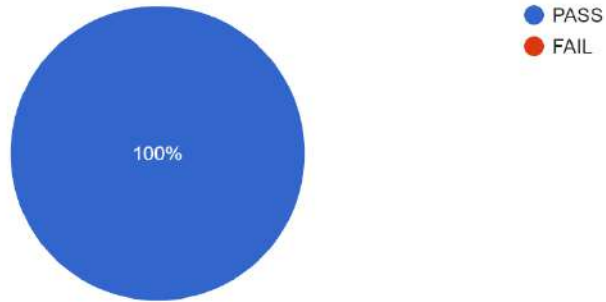


Information edited when edit information with valid title and content

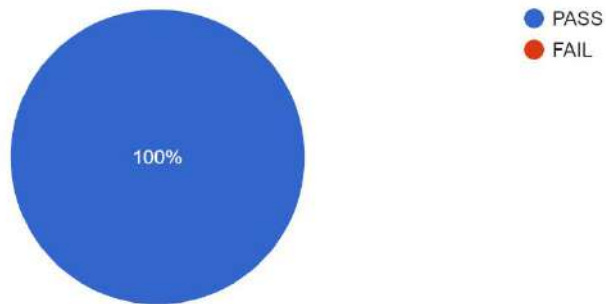
15 responses



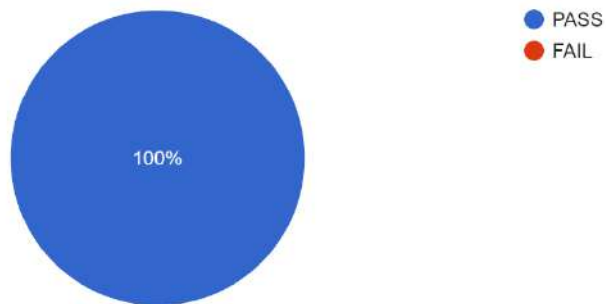
Information deleted when delete information when click on <<Delete>> button
 15 responses



Redirect to the website when click on the "View"
 15 responses



Display error message when add new information with invalid link, title or content.
 15 responses



Display searched information by title
15 responses



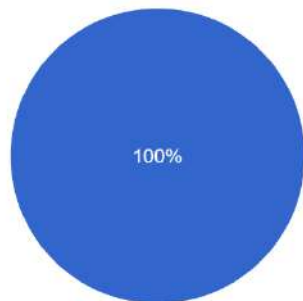
● PASS
● FAIL

Chatbot show reply when user type message and send
15 responses



● PASS
● FAIL

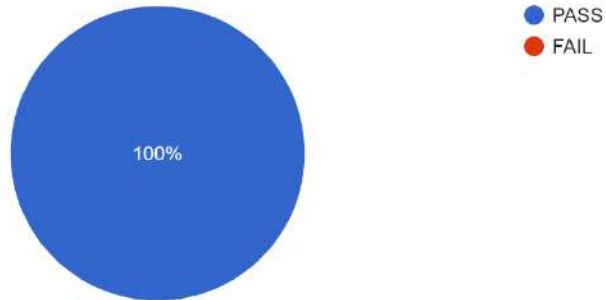
Chatbot show reply when user type question scripted in script
15 responses



● PASS
● FAIL

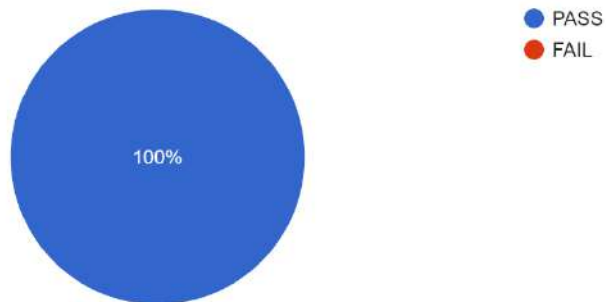
"Please enter text" message show when user send with null input

15 responses



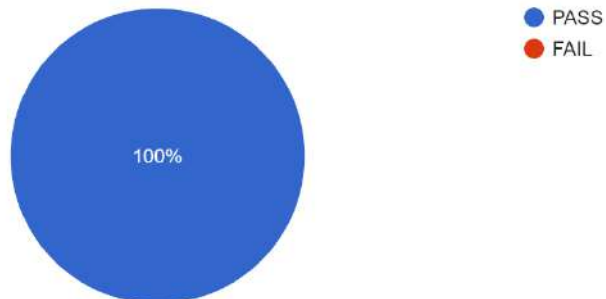
Chatbot show not understand reply when user type question not scripted in script

15 responses



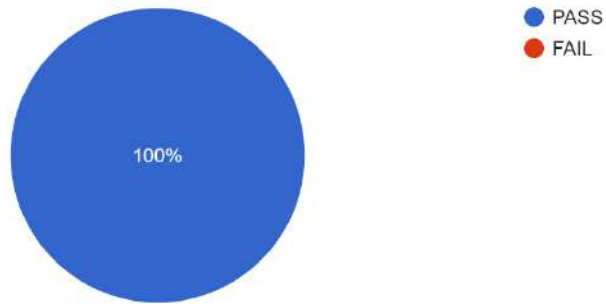
OCD test question display to user in Self-Assessment Test interface.

15 responses



Self-Assessment Result show accurately after user submit the test.

15 responses



Self-assessment history show in the Self-Assessment History interface.

15 responses

