PERSONAL MOBILE-HEALTH APPLICATION

BY

JEREMY LIM YEU SHUO

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS) INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2025

COPYRIGHT STATEMENT

ACKNOWLEDGEMENTS

I would like to express thanks and appreciation to my supervisor, Ts. Dr. Chai Meei Tyng and my moderator, Dr. Nur Syahirah Binti Roslan who have given me a golden opportunity to involve in the Mobile Application Development field study. Besides that, they have given me a lot of guidance in order to complete this project. When I was facing problems in this project, the advice from them always assists me in overcoming the problems. Again, a million thanks to my supervisor and moderator.

Next, I would like to thank all those who had helped me either directly or indirectly throughout this project. Without their motivational support, it would not be possible to complete this project. Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

Nowadays in the society, there is a phenomenon for raising health awareness and encouraging healthier lifestyles with the growing use of mobile health apps. However, the issues have been discovered such as ineffectiveness in managing the risks of chronic diseases, low user engagement, and insufficient features for individualized health insights. Through the application of innovative technological approaches and a user centred design approach, this project aims to fill these gaps. The Android-based mobile health application will employ ML-driven algorithms to diagnose the risk of chronic diseases, giving users personalized early warnings and proactive health management tools. The proposed mobile health application will be considered about user engagement and motivation of using the application which provides user engagement features like chatbot as personal health assistant to increase user engagement and motivation by achieve healthy lifestyle practice. The proposed application will also be leveraging the device's camera, various of assessment, text-sentiment analysis and AI emotion recognition technology that will evaluate the user's emotions before offering personalised insights and practical guidance to enhance mental and emotional health well-being. The application will provide user-friendly interfaces that make it simple to access and use personalized health information, goal monitoring and health comparisons of themself. In order to create a secure, scalable, and highly user interactive that enables users to efficiently manage their general health, the solution will be created in an Agile environment using tools like TensorFlow Lite, Firebase, and Kotlin as the primary of programming language in this development.

Area of Study (Minimum 1 and Maximum 2): Mobile Health Application

Keywords (Minimum 5 and Maximum 10): Mobile Health Application, ML-Driven Risk Assessment, User Engagement (Chatbot), Emotion Recognition, Personalized Health Insights

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

=	Equal Sign
x	Multiplication Sign
+	Addition Sign
-	Subtraction Sign
/	Division Sign

LIST OF ABBREVIATIONS

mHealth	Mobile Health Application
API	Application Programming Interface
WHO	World Health Organization
ICT	Information Communication and Technology
Covid-19	Coronavirus
ChatGPT	Chat Generative Pre-Trained Transformer
AI	Artificial Intelligence
CBT	Cognitive Behaviour Technique
EHR	Electronic Health Record
BMI	Body Mass Index
ML	Machine Learning
CNN	Convolutional Neural Network
SDK	Software Development Kit
XML	Extensible Markup Language
SDLC	Software Development Life Cycle
AWS	Amazon Web Services
XGBoost	eXtreme Gradient Boosting
CVD	Cardiovascular Disease
ERD	Entity Relationship Diagram
IDE	Integrated Development Environment
ONNX	Open Neural Network Exchange
ROC-AUC	Receiver Operating Characteristics – Area Under the Curve
HTTP	Hypertext Transfer Protocol
SHA	Hash Algorithm

Chapter 1

Introduction

Healthy lifestyle practices are goals that people worldwide strive to incorporate into their daily routines. These practices aim to improve overall well-being and extend life expectancy. Compared to previous generations, the current generation shows a heightened concern for their physical health, especially since the Covid-19 pandemic, which caused global panic [1]. During this time, the Ministry of Health Malaysia introduced several initiatives to educate Malaysians about adopting healthy lifestyle practices [2]. To maximize the implementation of healthy lifestyle practices, individuals increasingly seek information about their health [3]. The widespread availability of smartphones has provided opportunities for developers to introduce health-related applications, benefiting users and creating profitable markets. This includes the health sector, where mobile health applications gained popular during the Covid-19 pandemic [4]. However, there are variety of mobile health apps has led to a dilemma for users in choosing the most effective app for promoting a healthy lifestyle. Mobile health applications (mHealth apps) can be broadly defined as tools that provide health-related services, such as sharing health information with relevant parties, monitoring personal health, and preventing chronic diseases [5]. According to the World Health Organization (WHO), mHealth apps serve as public health guidelines supported by mobile devices like smartphones. These apps enable personal health monitoring and patient care in the healthcare sector [6]. They are categorized into education-focused apps for healthcare students, personal health trackers, fitness apps for tracking and managing physical activities, and telehealth applications for remote patient monitoring by healthcare professionals [7]. mHealth apps are developed due to helping individuals understand their health status and provide unique benefits that across various sectors. For instance, in education sector, these mHealth apps serve as training tools for medical students, enable them to simulating healthcare scenarios and offering access to virtual medical equipment typically found in clinics or hospitals [8],[9]. These medical tools allow students to develop decision-making skills and enhance their knowledge by making mHealth apps an integral part of healthcare education [10]. For personal use, mHealth apps are popular among fitness lovers and students which aiming to adopt healthier lifestyles. These mHealth apps will record the personal health metrics such as steps taken, calories consumed and recommended fitness routines for user [11]. The usage of mHealth apps for fitness lover popularity has increased during the Covid-19 pandemic when home-based exercise programs gained popular [12]. In this scenario, many fitness lover users were motivated to purchase smart fitness devices which these devices provided accurate tracking and increase the commitment to a healthier lifestyle [13]. The development of ICT and the internet have significantly boosted the adoption of mHealth apps in healthcare. These apps have created a transformative impact on public health by improving access to health information and supporting disease management [14],[15]. Globally, the popularity of mHealth apps has grown with approximately 99 million downloads on platforms like Google Play and the Apple App Store [16]. In Malaysia, smartphone usage has directly influenced the adoption of mHealth apps, with a new increase during the Covid-19 pandemic [17]. Although some groups remain unaware of these apps but they are generally demonstrating a positive attitude toward their purpose which indicating potential growth in their usage among Malaysians [18]. In conclusion, mHealth apps has offer various significant advantages in promoting healthy lifestyles and ensuring overall well-being. However, despite their ongoing development to making better, human health will still continue to face new challenges with the existing of new and currently for uncured diseases. Therefore, this highlights the need for continued awareness and innovation in healthy lifestyle practices even with the advancements in mHealth technology in current world.

1.1 Problem Statement and Motivation

Through the use of mobile health applications (mHealth) apps, individuals are becoming more aware of the importance of maintaining their health. In Malaysian society, mobile health applications have become a popular topic of discussion with usage increasing particularly among the urban residents [19]. However, despite to this positive trend, the overall health of Malaysians is getting worse. According to statistics from the World Health Organization (WHO), Malaysia's health index declined from 64.1% to 63.9%, representing a 0.2% decrease between 2000 and 2021 [20]. One of the primary contributors to this decline is due to chronic diseases which also remain the leading cause of death in Malaysia [20]. Chronic diseases are not just a national issue but also are considered a global health crisis by WHO due to their high prevalence and mortality rates in worldwide [20]. In Malaysia, the widespread prevalence of these chronic diseases is largely spread due to the lack of healthy lifestyle practices [20]. Certain groups in Malaysia, especially those from low-income families often disregard the severity of chronic diseases which viewing them as small matters in their daily lives [21]. In this scenario, it highlights that there is a lack of awareness and failure to adopt a healthy lifestyle are significant factors contributing to the prevalence of chronic diseases in Malaysia.

From the perspective of mHealth application, these tools are expected to play an important role in reducing chronic disease rates. However, challenges persist in achieving this goal. Recent studies have shown that mobile health applications are not yet efficient in mitigating the prevalence of chronic diseases [22]. Cost inefficiencies and the design of applications that address only specific diseases further exacerbate the problem [23]. Most mobile health apps for chronic diseases require users to unlock special features through subscription plans by having more accurate diagnostic results from the application and these apps are typically focus on just one chronic condition such as MySugr for diabetes, MyTherapy for hypertension and Google Fit for obesity. With this, it shows a clear trend which currently mobile health apps are built for specific diseases instead of offer diagnosing multiple chronic disease [24]. While this may benefit developers to create the apps in a way of faster, cost effective and provide a good experience to the user, however at the same time it also creates difficult for user. This is because in the perspective of user with several chronic conditions, they need to download multiple mHealth apps that are similar by just serving different purpose in order to manage their chronic condition which cause frustration to them [24].

Beyond chronic diseases, mHealth apps face additional challenges in helping users track personal health trends. Many of the technologies in the market lack essential features, such as personalized health reports like line chart that could significantly benefit users as they may understand about their current health condition [23]. In this case, user might difficult to track their health condition progress which they need to manually record it down their health condition periodically. Additionally, most applications are required users to manually access the feature in the mHealth apps, which increase the burden for users instead of using advanced technology such as voice command to mitigate it [23].

Another critical aspect is user interaction with mHealth apps. While these apps are designed to monitor personal health but they often lack interactive features such as reminders or push notifications that keep users continually engaged. Notifications are important in mHealth apps which not only they can help users keep track of their health condition and remind them of important tasks such as taking medication pills but also boost user engagement. Even though this is a small feature in the mHealth apps but it actually has a big positive impact on keeping users involved [25]. Furthermore, most of the mHealth apps are lacks of feature that user able to ask about health related inquires and also providing recommendation regarding to their current health condition. Users require to use reliable external sources like Google to answer their health information questions and find recommendations to managing their current health condition [26]. Therefore, this absence of interactive elements leads to boredom and eventually a decline in usage [27]. Moreover, mood plays a significant role in influencing healthy lifestyle practices. Individuals with a positive mood are more likely to adopt healthier lifestyles, whereas those with poor emotional states are less likely to do so [28]. Despite the significance of mood in promoting health, many mHealth apps fail to incorporate emotional well-being as part of their functionality [29]. Most apps prioritize physical activity as the sole measure of a healthy lifestyle while neglecting mental health aspects [27]. Without consider integrate mental health into mHealth apps, users might have difficulties of knowing overall body health condition.

Therefore, this project aims to address the challenges identified above by focusing on developing a comprehensive mHealth apps that promotes both physical and mental well-being.

Moreover, this study proposes improvements to existing mHealth apps by incorporating both internal and external physical aspects of health to promote an overall healthy lifestyle. The application will utilize ML model to diagnose the risk of common chronic diseases by providing users with valuable insights into their health status and increasing their awareness of these conditions. In addition to identifying chronic disease risks, the application will also allow users to make personal health comparisons and observe trends through descriptive statistics such as trend charts and generate a summary of comprehensive health reports. These features enable users to view all aspects of their health status in an organized and accessible format.

To enhance user engagement, the application will include AI technology such as chatbot which acts as personal health assistant to ensure the users maintain the right track of having healthy lifestyle and also motivate users to adopt and maintain healthy lifestyle practices. Furthermore, this application will incorporate mood analysis as a crucial aspect of healthy living in terms of mental health by leveraging AI technology together with text-sentiment analysis to detect and assess the user's emotional state

1.2 Objectives

Every project has specific objectives to achieve. Therefore, this project has four main objectives:

1. **To developed a machine learning model to diagnose the risk of chronic diseases and create awareness for users.**

This objective aims to implement machine learning model that able to analyse health data provided by users to predict the risk of developing chronic diseases specifically cardiovascular disease, obesity, diabetes and to provide early warnings for severe outcomes such as heart attack conditions identified as the most general chronic diseases in Malaysia by the Ministry of Health [31]. By offering personalized insights and early warnings, the application will enable users to take preventive action and manage their health proactively by addressing potential issues before they become worse.

2. **To create a personalized chatbot to increase user engagement with the mobile health application.**

This objective aims on developing a personalized chatbot that serves as a "personal health assistant" by addressing user inquiries regarding healthcare challenges and preventive measures for maintaining a healthy lifestyle. The chatbot will provide personalised responses based on user input and will analyse patterns from features like diabetes, journal prompt and face emotion data to offer real-time health insights. It will also remember the conversation history between users to encourage continuous engagement and support both mental and physical health.

3. **To detect user emotions by adopting various analytical tools to assess mental health and promote an overall healthy lifestyle.**

This objective involves integrating multiple methods such as text sentiment analysis, mental health assessment and potentially facial recognition via the device's camera to capture and assess users' emotional states accurately. By providing personalized advice based on real-time mood tracking, the application will help users manage their emotional health, promoting self-awareness and encouraging effective strategies to reduce the negative emotions. Users will have the option to express their emotions through text, assessment or images, which will allow the system to deliver more precise and context-sensitive recommendations.

4. **To develop a culturally adaptive, user-friendly Android application with a user-friendly interface and voice navigation by enhancing accessibility for diverse Malaysian users.**

This objective is to design an Android-based mobile health application that is easy to use and culturally adaptive for Malaysian users. The application will feature personalized health reports based on chronic disease risk assessments and mental health insights. It will include voice navigation for improved accessibility and given options to the user which providing support in bilingual languages to reduce barriers for different Malaysian communities. The interface will be designed to ensure a positive and engaging user experience while continually evolving based on user feedback.

1.3 Project Scope and Direction

This proposed mobile health application will offer a combination of health related and mental health application of an overall mobile health application to enhancing both physical and mental health. To provide personalized health recommendations, the app will integrate AI-driven mood detection with AWS Face Rekognition by scanning and presenting user emotion evaluations using a mobile device's camera and also having various assessment and comment that describe from users in order to get the accuracy of mood user. Additionally, it will use developed machine learning model to assess the risk of chronic diseases and offer actionable advice on how to manage their health, as well as provide early warnings based on the user provided health data. To encourage users to live a healthy lifestyle, the app will include features like a chatbot with ChatGPT that acts as a personal health assistant. Users can ask it about physical health and mental health concerns. Physical activity tracking such as step count and personalized health reports as the last feature of this application which will boost user awareness and participation into the practice of healthy lifestyle. This application targets the Malaysian community and offers English and Bahasa Malaysia languages. Users can switch between these languages based on their preference.

However, this project has some limitations. One major limitation is its emphasis on non-clinical, machine learning model that provide evaluations and health engagement methods, rather than providing qualified health diagnoses or treatment. The risk assessment in this application serves only as a reference to inform users about their current physical and mental health and to provide personalised recommendations based on the results while it is not suitable for official clinical use. Another drawback is this project application is exclusivity to the

Android operating system, which limits accessibility for iOS users and ensures easy development and optimization for Android devices. Additionally, the application must follow data privacy and security rules to ensure users' sensitive personal information is not used by third parties for their own benefit.

For the system architecture of this application, the project will use Agile methodology as the system architecture which this approach allows for rapid development and frequent updates until the application meets user requirements and the standards for a mobile health application. The deliverables of this project will include a working app with all module features that provides a basic overview of the mobile health application flow. Therefore, this deliverable will describe the outcomes related to both physical and mental health of a user and also identify the improvement need to be done in this mobile health application which able to ensuring users can practice and maintain healthy lifestyle habits.

1.4 Contributions

The proposed mobile health app aims to make a positive impact by encouraging healthier lifestyles and improving both mental and physical health. By using developed machine learning model, the application can analyse users' health information to identify their risk of chronic diseases. Based on the result it predicted, it will provide personalized advice and early warnings by helping users take action to prevent illnesses and support better overall public health.

The application also incorporates features for tracking personal health metrics and generating reports by providing users with a comprehensive view of their health. These tools can increase awareness and accountability, encouraging users to adopt healthier habits. To further engage users, engagement elements such as chatbot will serve as a personal health assistant which not only the user can ask about the health-related issues and also user can express their frustration to the chatbot which it acts as a listener and provide comfort the user.

Another society benefit comes from the application is the mood detection feature, which analyses users' emotional states using AI technology and various of technique such as text-sentiment analysis and mental health assessment to provide tailored feedback and support. This helps users identify potential mental health challenges early and offers actionable recommendations for improvement [30]. By integrating these advanced technologies, the application not only promotes physical activity but also contributes to mental well-being, produce a more fostering health awareness society.

In summary, the mobile health application combines with developed and trained machine learning model for health diagnostics, personalized health tracking, personal health assistant and AI-driven mood analysis to deliver a comprehensive tool for improving public health. By making health management accessible, engaging and personalized, the application has the potential to drive wide spready the adoption of healthier lifestyles and reduce the societal burden of chronic diseases and mental health issue.

1.5 Report Organization

This project report consists of 7 chapters and the details will be explained in the following chapters. In Chapter 1 Introduction, it introduces the application and discusses the project objectives, scope, motivation and contributions to users and society. In Chapter 2 Literature Review, there will be review existing mobile health application related background of chronic disease management and mental-health monitor by comparing their features, strength and weaknesses. Next, the overall system architecture diagram, wireframe design application, machine learning model and system design diagram will be proposed and explain in this Chapter 3 System Methodology. In Chapter 4 System Design, it will present and explain the application block diagram, flowchart diagram, ERD Diagram, database setup and the configuration of relevant external services. In Chapter 5 System Implementation will be discussed and presents about the software and hardware requirement, system operation with relevant screenshot, the application implementation issues and challenges and conclusion remark of each module. In Chapter 6 System Evaluation, it will be discussed the system setup and testing result, evaluation of the performance developed machine learning model, the project challenges, objectives evaluation and concluding remark based on the results obtain from each section. Lastly, Chapter 7 Conclusion and Recommendation will be discussed about the project conclusion and provide recommendation for improving the application in the future.

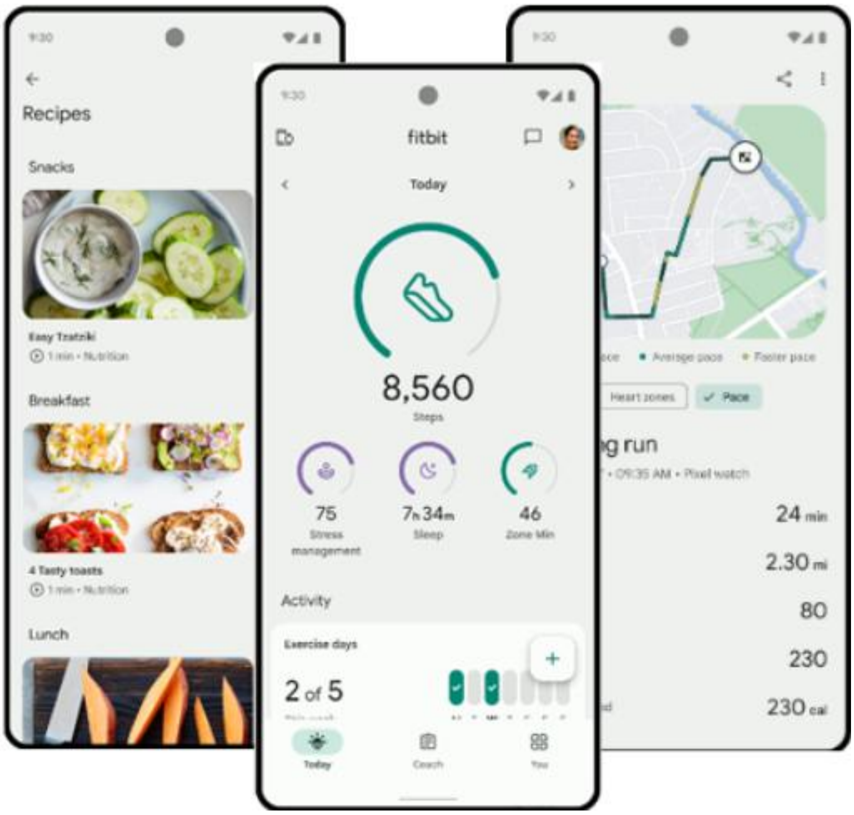
Chapter 2

Literature Review

In this literature review section, there will be two parts. The first part will focus on existing reviews of mobile applications and followed by a review of related research papers. This section will first present the review of mobile applications and then move on to the research paper review.

2.1 Mobile Health Application Review

2.1.1 Fitbit [32]



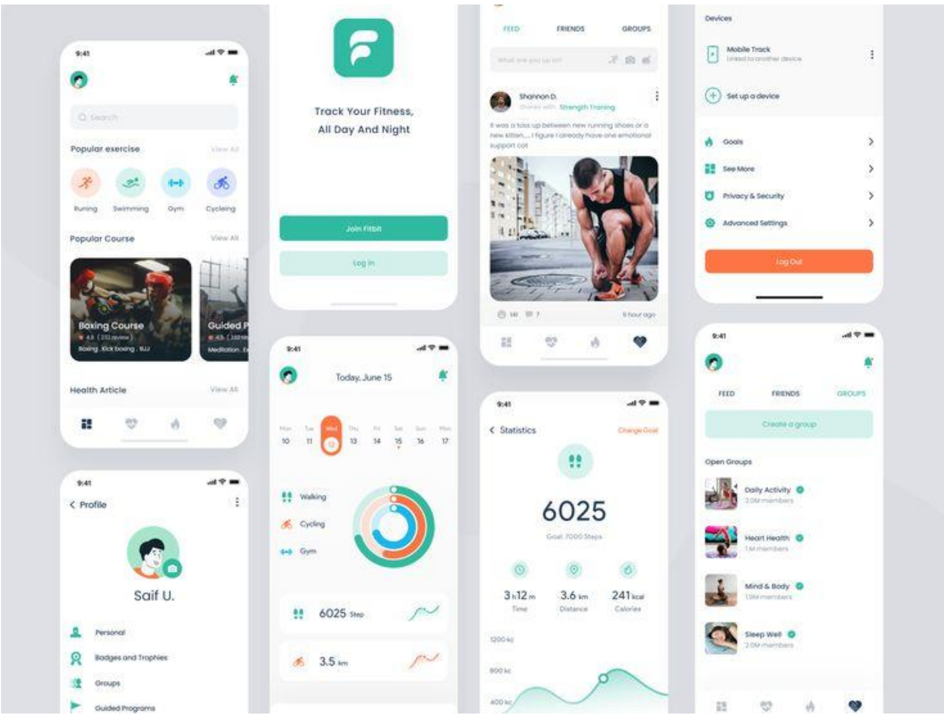


Figure 2.1.1 Fitbit Application

Google Fit is a popular health app in Google Play store in which offers features to user like heart rate monitoring, activity tracking and nutrition tracking. It follows the guidelines from the WHO to promote healthy habits. Not only that, the app introduces the programs like "Move Minutes" and "Heart Points" to encourage daily exercise which helping users stay engage and motivated.

Furthermore, Google Fit also lets users track their health over time which giving helpful insights into their progress based on the result obtain. This app also has extra benefits which its able works with other health apps to combine data from multiple sources by provide the experience smoother for users. However, one drawback of this application is that there are some features rely on these third-party app connections. Without them, certain advanced functions might be limited.

2.1.2 Garmin Connect [33]

Figure 2.1.2 Garmin Connect

Garmin Connect is a free health and fitness app available in Google Play Store that offer features to user like step counting, calorie tracking, and heart rate monitoring. Not only that, it let the users able to customize their workout plans easily. One unique feature that differentiated this app to the other app is this app has the ability to measure red blood oxygen levels and check cardiovascular fitness in real-time which giving users detailed information about their physical health based on the obtain result through user measurement. The app also allows users to share their fitness data with others which encouraging social interaction. By maintain the user engagement, the app introduces leaderboard feature which lets users to compare their progress with friend. Indirectly this will be creating a sense of competition and motivating regular use.

However, Garmin Connect has some drawbacks. One of the drawbacks in this app is the advanced features like detailed fitness metrics, can only be accessed by connecting to their Garmin fitness device which is not cost friendly for users. Not only that, the app sometimes faces server downtime by making it hard for users to access their data. Lastly, it mainly focuses on fitness and basic health tracking with limited features for monitoring medical conditions or overall, about a personal health.

2.1.3 EaseIt: An Android based Health Monitoring Mobile Application [34]

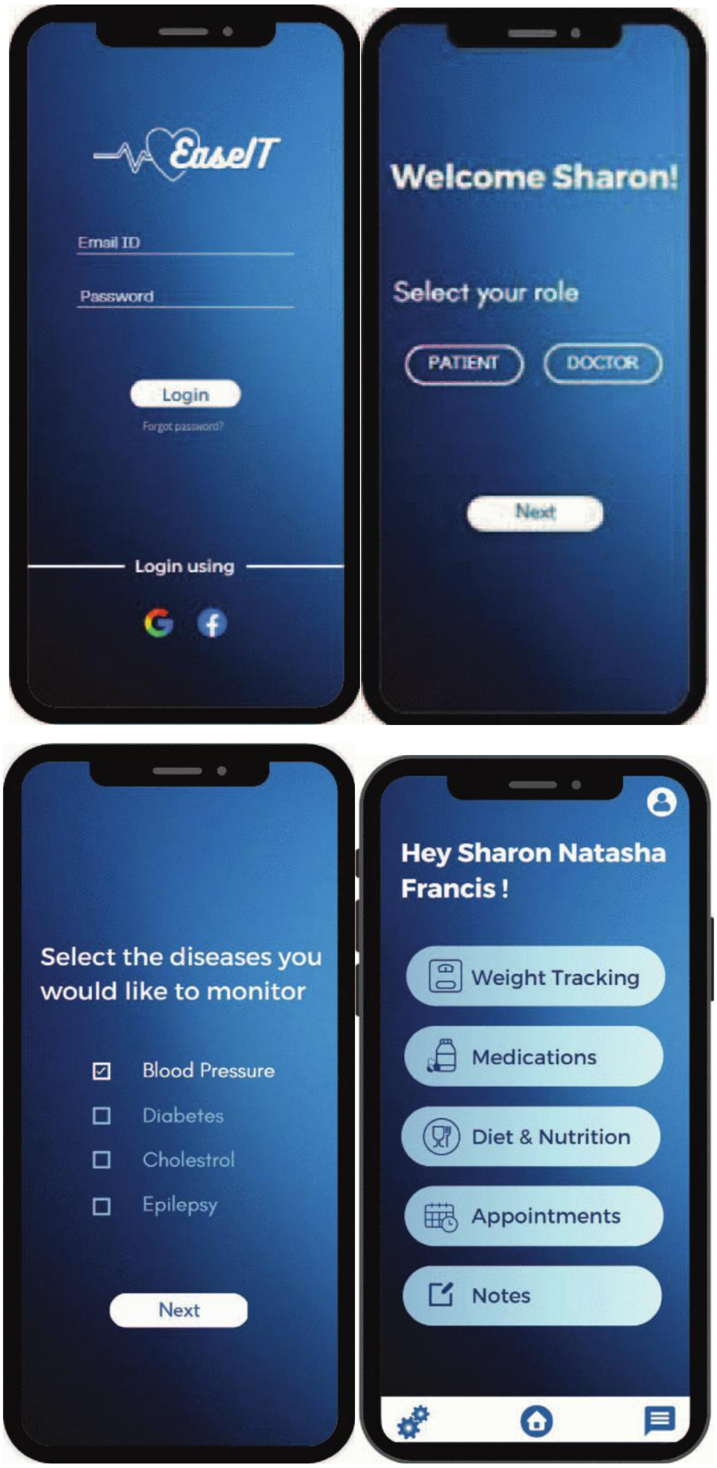


Figure 2.1.3 Easelt Mobile Health Application

This Android app called Easelt that specifically developed aims to reduce chronic diseases. Easelt helps users monitor their health by tracking key metrics like blood pressure, blood sugar and weight so that the user can take precaution and steps to prevent illness. The app also offers diet monitoring and provides exercise recommendations for the users which is to support healthy lifestyle. Additionally, Easelt also includes a chat feature that lets users to interact with health professional in real time and also able to schedule appointments with doctors based on their convenience.

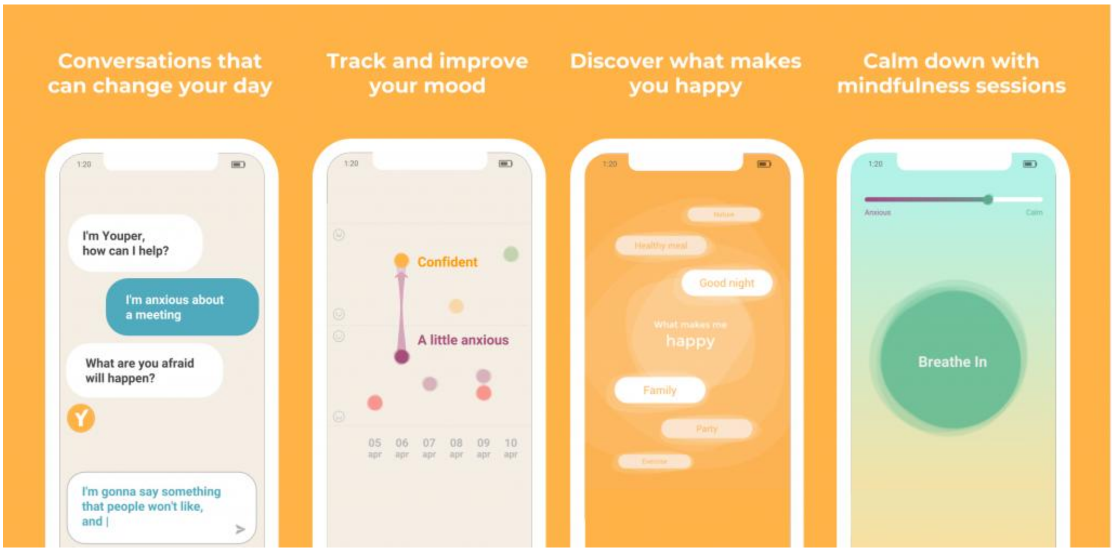
Despite these important features has provide benefits to users, however the app has limitation such as low user engagement and the lack of leverage of advanced AI technology for detailed analysis. Therefore, the researcher suggests that future improvements should include integrating AI technology and using cloud-based solutions for better data management and system scalability.

2.1.4 MOODIFY: Tailored, Personal and Multifaceted AI Assistant for Young Adult Mental Health Issues [35]

This proposed Android app in the researcher paper called Moodify which is to designed to help young adults with early-stage mental health problems. The app has the features of an AI facial emotion detection module that accurately recognizes facial expressions and recommends personalized music and podcasts for user based on the result to improve the user's mood. Indirectly, this real-time assessment helps users to better understand and manage their emotions. In addition, the app also has a feature called AI chatbot that uses deep learning technique to offer individual coping strategies, mental health support and empathetic conversation. Besides on that, Moodify also provides a social community platform where users can connect through anonymous group chats and build peer-to-peer relationships.

Despite the application contains many benefit features however the app has some drawbacks such as difficulty in detecting complex emotions through facial recognition which limited accessibility for people with environmental or physical challenges and also a lack of integration with mental health professionals.

2.1.5 Youper (Mental Health Application with chatbot) [36]



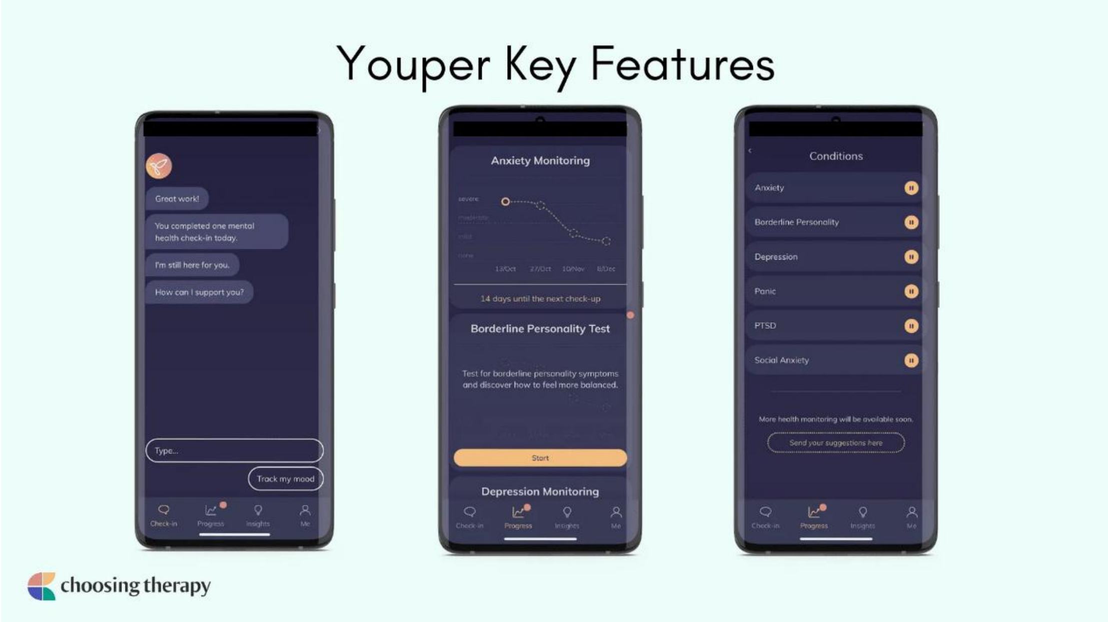


Figure 2.1.4 Youper Mental Health Application

Youper is a personalized AI-chatbot mental health assessment apps which designed to improve user mental-health being, aims to reduce the user's negative mental health issues. By reduce the negative mental health issues, Youper adopt Cognitive Behaviour Techinque (CBT) which one of the popular psychological treatments to identify the user's negative mood and provide recommendation as to reduce the negative mood from the users by lead to the users to having a positive mindset and also boost their motivation. This app consists of features such as personalized AI chatbot that offers effective personalized recommendation based on the user current mood, a user friendly-interface which makes the user found it interactively and increase engagement of using the app. It also provides a summary of user mood report by tracking the user's mood everyday consistently which allow user to identify their mood trends pattern and create awareness into it if the trends show not optimistic result.

Although it might seem these features has provided many benefits to the users, however this app contains limitation into it. This app is significantly requiring user engagement to provide their mental health by manually input. The chatbot will not notify or methods that aware the users to record their mood to understand their personal mental health growth. Not only that, Youper required subscription in order to utilize the usage of apps which are not cost efficiency for user. User may able to use it but able to use it in limited feature. Youper apps also require a stable internet connection as the chatbot relies on stable and good internet connection to provide analysis and suggestion from the reliable internet sources.

2.1.6 Wysa (Mental Health Application with Chatbot) [37]

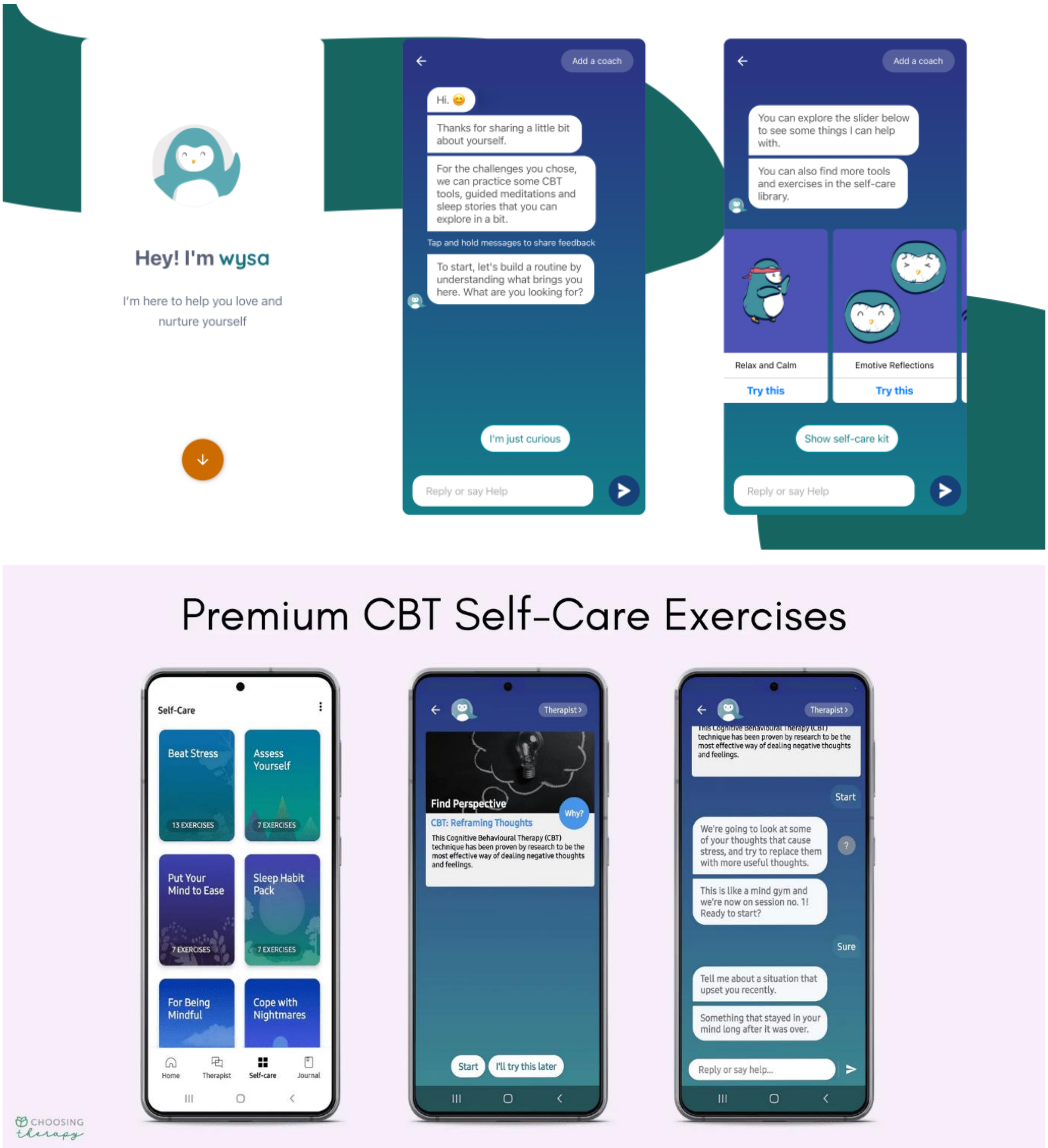
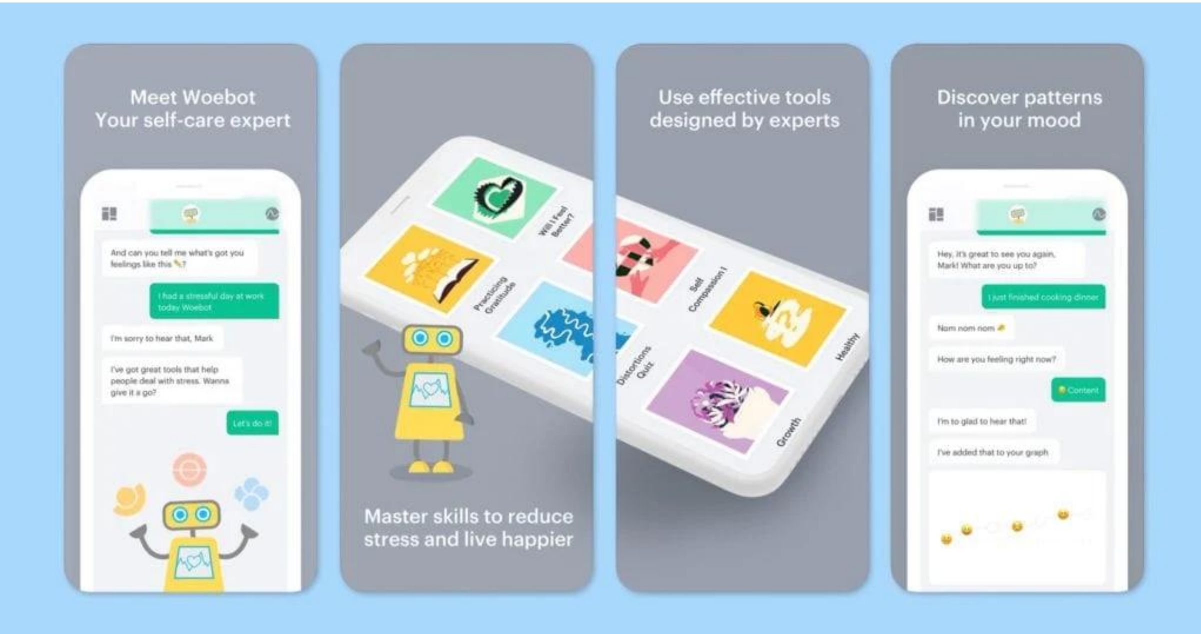


Figure 2.1.5 Wysa Mental Health Application

Wysa is an AI-chatbot personal mental health assessment apps which provide users a good recommendation insight of encounter mental health issues. It contains the AI chatbot of "penguin" avatar acts as personal mental health coach by provide CBT technique leads to positive mental health well-being. It consists of feature such as personalized AI-chatbot which will interactive with user to increase user engagement that able to access by anytime without concern about the time matter, mood tracking which user able to log their mood condition and track for their mood trends to understand their mood behaviour over time. It also ensures about data privacy while user will be acts as anonymous in this app without knowing by the third parties. Users able to express their current condition without concern about their information will be used by the external parties. Not only that, it also provides some simulation video such as meditation exercise which guide users step by step to aims to improve mental well-being.

Although seems provide useful feature to users, however it might have some limitations in this app. While Wysa consists of powerful AI-chatbot to assess the user mental health, however it does not guarantee that the assessment made by the AI chatbot will be accurate as the current user mental health. In other words, it highly depends on how user write their message to the chatbot to express their emotions or feeling. If the users are not good enough to express their emotion on the message, it might possibly have a wrong result from the AI-Chatbot. Not only that, it also requires user engagement of using the app to log their mood to the AI-Chatbot to do the analysis. Without user interaction, the AI-Chatbot could not produce an accuracy report based on the user mental health and AI-chatbot will also not notify if the user forgets to log in their mood on that day. The app is only available in English Language which is problem of language barrier for the user.

2.1.7 Woebot Health – Chatbot with Mental Health Application [38]



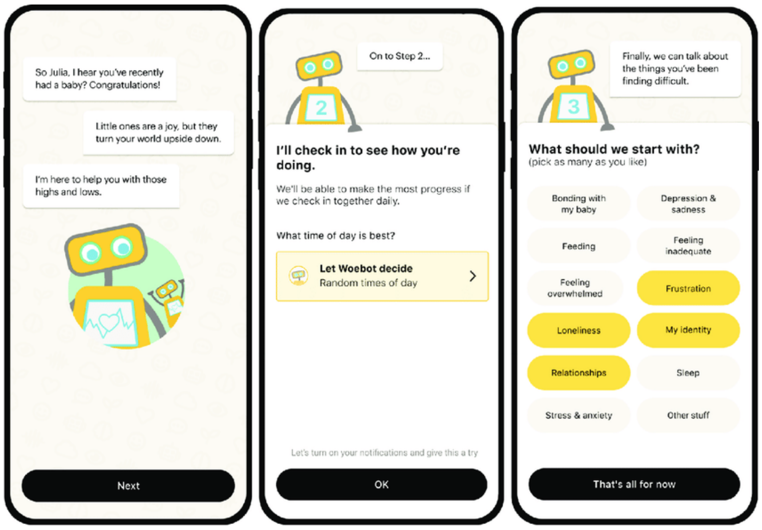


Figure 2.1.6 Woebot Health – Chatbot with Mental Health Application

Woebot health is a powerful AI-chatbot app that provide personalized mental health support to user by adopt CBT technique to improve user's mental health. This app consists features such as interactive AI-chatbot where users will input their emotions through the conversation with the chatbot and it will respond based on the personalize recommendation to the user of their emotion. The chatbot not only just act as friendly personal but it will also remind the users during their daily lives about their current mood which is the special features in this app. This AI-chatbot will analyse and track user emotion and provide a trend mood report for user to let them aware about the trend pattern of their mood. The chatbot will also provide the simulation video of improve positive mental health well-being if the user trend pattern is worse as to ensure the user mood is always maintain in the positive trend pattern. User able to chat with the chatbot anytime, anywhere as long as a stable internet connection which provide accessibility to the user. In this app, user will be act as anonymous where they can express their emotions freely with the chatbot without concern about offending someone as this app ensure the privacy conversation to the users.

It seems that this app has provide many significant benefits to the user maintain positive mental health well-being however it also has limitations into it. Woebot health relies on text-based message only where user able to express their emotions are through manual input only with the chatbot. In other words, there will be have a possible that the chatbot will not fully understand about the user emotions as the chatbot will analyse the text that input by the user which might occur accuracy problem. Not only that, this app has just available in English language which may occur language barrier problem as not all users have a strong fundamental English to understand it. This might be a problem for users who are in senior citizens age or who are just primary school as for their highest education level. The app will also need a stable and good internet connection which is not suitable for the users who live in the rural area.

2.1.8 Moodfit – Mental Health Application [39]

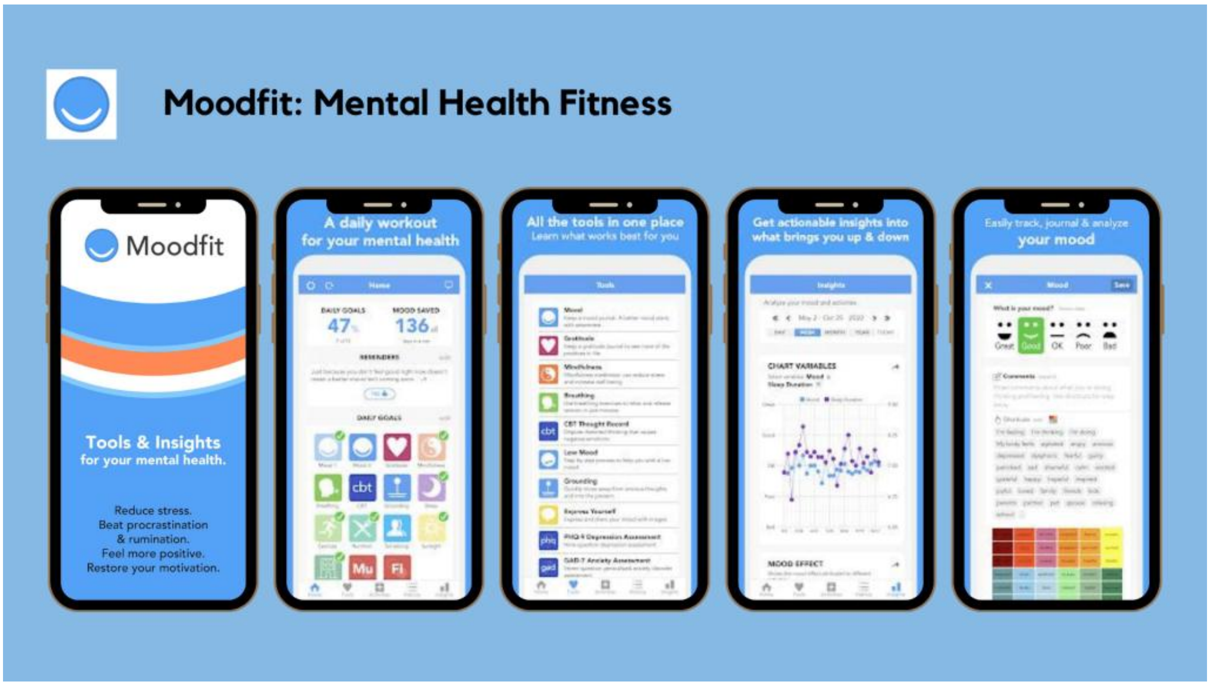


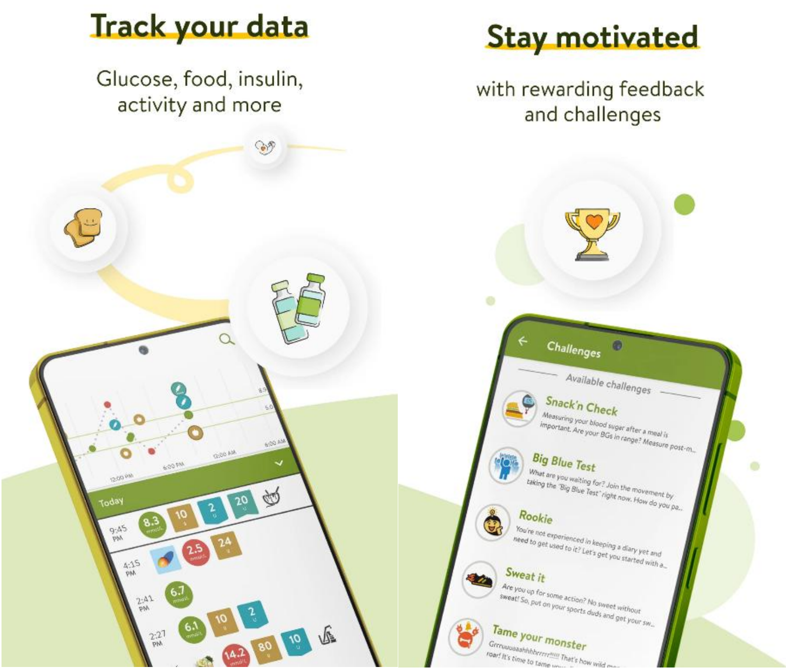
Figure 2.1.7 Moodfit – Mental Health Application

Moodfit is a mental health app that available on both iOS Apps Store and Android Google Play Store that helps users to track and improve their emotional health. The app includes features like record the mood data and daily mood assessments which are combined with other health data such as sleep and exercise. Next, Moodfit also in an easy-to-use dashboard shows changes in mood over time and highlights the connections between the user's personal lifestyle factors and emotions. This is because, Moodfit wants to makes it simple to track daily moods by offering customizable questions that encourage users to reflect on their feelings. In addition, it also supports syncs data from external sources like exercise apps and sleep monitors and uses this information to give personalized suggestions such as mindfulness exercises or fitness tips as to help users feel better.

Furthermore, Moodfit has its own speciality which is its complete and user-friendly design. By merging objective lifestyle data with users' personal mood reports, Moodfit offers a full view of mental health and encourages proactive self-care to user as to maintain the user in practicing the healthy lifestyle. Not only that, its interactive design and daily check-ins help keep users engaged while the personalized guidance assists them in understanding and managing their emotions.

However, Moodfit also has some drawbacks. Firstly, Moodfit are relying on self-reported data which can lead to inconsistent data quality that may affect accuracy of the data result. Additionally, the app might not be suitable for those with serious mental illnesses as it is mainly designed for light to moderate emotional issues. Next, there are some advanced features require a subscription fee from user which can limit access for them and there are concerns from users about the privacy of sensitive personal information.

2.1.9 MySugr – Diabetes Management App (Chronic Disease) [40]



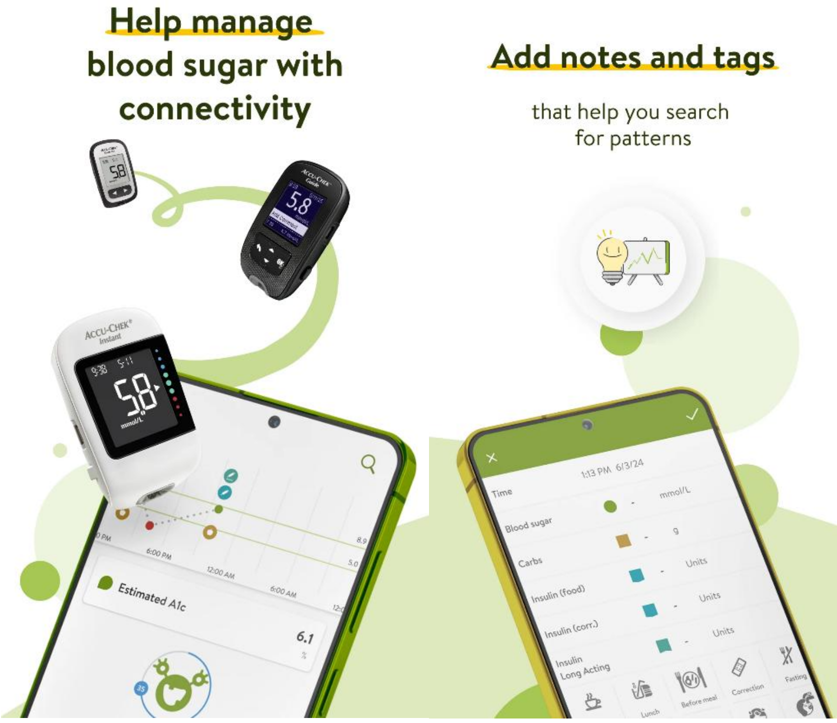


Figure 2.1.8 MySugr – Diabetes Management App (Chronic Disease)

MySugr is a popular mobile health application that available both in Android Google Play Store and also Apps Store for iOS user that designed specifically for diabetes management which allowing users to efficiently track and manage their condition. The application only supports a single platform for tracking vital health information such as blood sugar, insulin dosages, food intake and physical activity. The data in this application will be shown in simple visualizations like trend charts and comprehensive health reports and users have the option to enter the data themselves or use device integration such as wearable devices.

Not only that, MySugr also includes gamification features in the application such as challenges, awards and achievement tracking to encourage users' engagement and help users stick to their management schedules. Based on the user-inputted for logged information, the feedback is provided on an individual basis which enabling proactive daily activity changes and general health care. There is one speciality in this MySugr's application which is it has the integrated diabetes treatment model. The application provides more informative information about a user's status by integrating several health parameters into a single and user-friendly platform. This gives users the ability to combine their lifestyle and glucose control into a single and straightforward format.

Therefore, this app has been demonstrated that an engaging of gamification-based design that offers rewards for positive behaviour improves long-term user retention and treatment plan adherence. Additionally, the personalized insights and useful suggestions enable users to make informed choices that may improve their health.

Although this app may have many advantages, there are some drawbacks to this app as well. Firstly, if users are not careful when entering their data, the system will continue to process the mistake human entry which may result in inaccuracies and inconsistencies of data. In this situation, human entry is still a significant factor even if the program does provide connectivity with a wide range of devices for automatic data entry. Furthermore, while most of MySugr's primary features are free however there are still have premium feature that typically charged for which may limit some users' access. Given the sensitive nature of the health data being handled and the need for strong security measures, data privacy also plays a role in this situation. Lastly, even though MySugr works well for managing diabetes, its features and design are quite specific to this condition but it also at the same time making it difficult to use for other chronic conditions without significant changes.

2.1.10 MyTherapy (Chronic Disease App) [41]

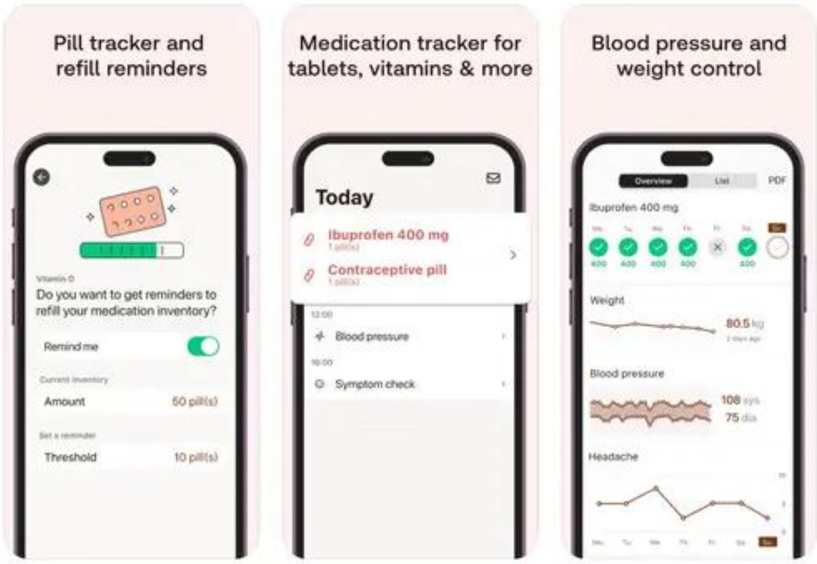


Figure 2.1.9 MyTherapy (Chronic Disease App)

MyTherapy is an integrated mobile health application that available in Android Google Play Store and iOS app store that uses integrated health status monitoring and improved medication compliance to help manage chronic conditions. MyTherapy has many features such as it able to customized adherence based on reminders, full dose logging, and reports on symptoms, mood and vital signs. In addition, in order to ease of use for users to keep accurate health records, the data collection makes it possible to create thorough health reports that are simple to share with medical professionals also facilitating a more educated and cooperative approach to disease management. In terms of functionality, MyTherapy has a speciality for its user-friendly, intuitive design and navigation which able to appeal to a wide range of users, including people who are not technical.

The app also has an ability to handle multiple languages and schedule that allows users of diverse backgrounds to customize the program and to meet their own specific requirements. The design of the app has focus on simplifies the user experience and provides a comprehensive picture of a person's health by combining multiple aspects of health monitoring such as medication adherence and daily health measurements into one system.

Although this app provides useful features to the user however there are some of existence limitations have been found in this app. Firstly, the app is reliance on human data entry which is the one of its primary drawbacks if the users submit their data incorrectly or at the wrong time, this might result in errors and inconsistencies. Furthermore, although the app is excellent at monitoring and reporting health data, it lacks of advanced artificial intelligence (AI) features that may be used to analyse trends, predict possible health problems or provide early warning signs. Additionally, the lack of AI-driven data prevents the app from providing proactive and personalized health suggestions which would make it a more useful early intervention tool. Furthermore, there is a lack of connection with more comprehensive electronic health record (EHR) systems which may make it more difficult for patients and their healthcare professionals to transmit critical health information.

2.2 Research Paper Review

2.2.1 Evaluation on the Persuasive System Design Model for Human Wellbeing in Mobile Health Application [42]

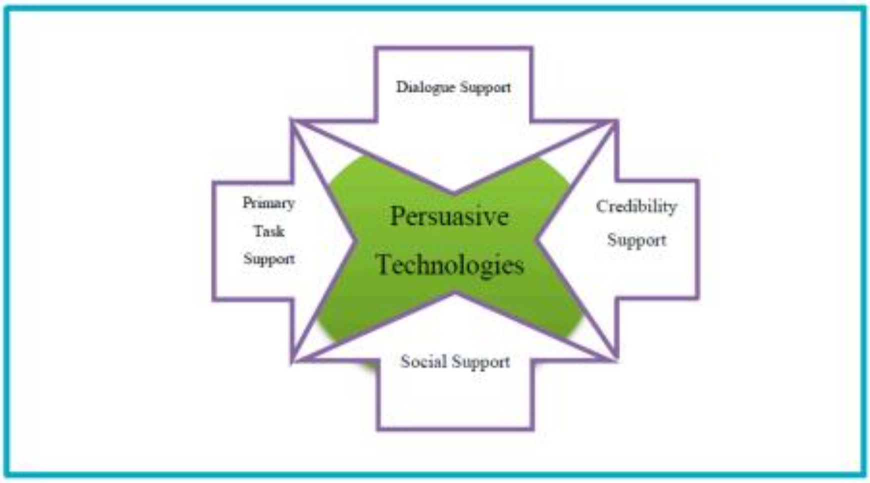


Figure 2.2.1 Persuasive System Design Model

This paper explores how persuasive technology can influence people to adopt healthier lifestyles, especially through mobile health apps. The researchers introduce a system called Persuasive System Design (PSD) that encourages users to add healthy habits to their daily routines. PSD serves as a motivator for regular exercise and overall wellness and includes four main parts.

The first part which called Primary Task Support is to help users meet their health goals by providing tools to allow them track their fitness and health data. The second part, is Dialogue Support which to keeps users engaged by offering interactive features such as rewards and badges for reaching milestones or chatbot as personal health assistant. The third part is the System Credibility Support which to builds trust by providing clear, accurate, and useful information for users to ensure there use the app without concern of data privacy issues. The fourth part of this design is called Social Support to encourages community by allowing users to connect with friends and family which boosts motivation and accountability of the user.

Although the system is effective in keeping users motivated with positive reinforcement, however the app also has some limitations such as not leverage of AI technology to analyse health data for deeper insights and also not considering the emotional side of a healthy lifestyle. Therefore, the researcher suggests that adding these features could make it even more effective and improve the overall user experience.

2.2.2 Use and Effectiveness of Mobile Health Applications for Stress Management and Emotional Self-Regulation in Adult Workers: A Systematic Review [43]

This research paper reviews the effectiveness of eleven mHealth apps available on the market for stress reduction and emotional self-regulation. These apps provide features such as structured modules, guided mindfulness exercises, educational content and helpful techniques like body scans, mindful movement, gratitude diaries and breath monitoring to boost users' emotional in a way better. Based on this paper the, the researcher mentioned that these apps lead to significant improvements in emotional self-regulation and work-related satisfaction and they help to reduce perceived stress, burnout and symptoms of anxiety and depression. Not only that, it also mentioned about the better user engagement and higher adherence were linked to features such as reminders, flexible session durations and user-friendly interfaces. Not only that, in this paper, the researcher also mentioned that some studies also reported that mHealth apps are cost-effective and easily scalable compared to traditional face-to-face interventions with mindfulness-based programs consistently showing positive outcomes.

However, the reviewed by the researcher also identified drawbacks such as the need for more personalized content and variations in effectiveness among different users. In conclusions, the researcher has highlighted the need for tailored features, focused interventions and intuitive app design to enhance the accessibility, usability and overall impact of these apps.

2.3 Summary Comparison Table for Mobile Health Application Review

Table 2.3 Comparison for Mobile Health Application Review

App Name	Type	Key Features	Strengths	Limitations
Google Fit	General fitness tracker	Heart rate, activity, nutrition, Move Minutes, Heart Points	Tracks over time, works with many other apps	Some features need third-party connections
Garmin Connect	Fitness and health tracker	Steps, calories, heart rate, SpO_2, cardio fitness	Real-time oxygen and fitness data, social leaderboards	Advanced metrics need Garmin device, occasional server downtime
EaseIt	Chronic disease monitor	Blood pressure, blood sugar, weight, diet and exercise tips, chat with health professional	Full health chat, doctor appointment booking	Low user engagement, no advanced AI for deep analysis
Moodify	Mental health assistant	Facial emotion detection, music or podcast suggestions, AI chatbot, anonymous group chat	Real-time mood help, social support	Hard to read complex emotions, no pro-therapist integration
Youper	Mental health chatbot	AI chatbot using CBT, mood tracking, daily mood report	Personalized CBT tips, mood trend charts	Manual mood input needed, subscription fee, needs internet
Wysa	Mental health chatbot	Penguin-avatar AI coach, CBT chats, mood log, meditation videos	Private, anonymous chat room, mood trends, guided exercises	Chat accuracy depends on user text, only English, needs internet
Woebot Health	Mental health chatbot	CBT-based AI chat, daily mood reminders, mood trend reports, guided videos	Always-on chat, mood tracking, privacy	Text-only input limits understanding, English only, needs internet

Moodfit	Mental health tracker	Mood log, daily assessments, dashboard linking mood to sleep or exercise, provide tips	Clear mood-lifestyle view, data sync	Self-report may be inconsistent, some features require subscription
MySugr	Chronic Disease Diabetes management	Blood sugar, insulin, diet, activity logs, charts, gamification features	Easy charts, fun challenges, integrated model	Manual data entry errors, premium features cost money
MyTherapy	Chronic Disease condition helper	Medication reminders, dose logging, symptom, mood or vital reports, multilingual language	Easy to use, shareable reports, multi-language	Manual entry only, no AI analysis, no EHR integration

2.4 Limitation of Previous Studies

Based on a review of existing mobile health apps, each of the application that contains own weaknesses and missing the features especially which does not include diagnosing for chronic disease and also mental health well-being. For example, Garmin Connect and Fitbit only track basic metrics like steps and heart rate. They do not predict or warn about chronic disease risks even though conditions like diabetes and CVD cause the most deaths globally in the world. Next, for Google Fit, while it is useful but it cannot easily share data with other health apps. This application has limited its ability to combine multiple data sources in order to provide a comprehensive picture of a user's health. EaseIt offers blood-pressure and blood-sugar tracking but it does not use AI technology to analyse trends or give personalized advice. Moodify, Youper, Wysa, and Woebot applications all focus on mental health chat features but they struggle to analyse complex emotions accurately. Moodfit and MySugr provide good dashboards and gamification for their application but they rely on manual data entry and often lock advanced features behind subscriptions. MyTherapy helps with medication reminders but lacks AI analysis and cannot connect to broader medical records.

In the academic studies, it also has similar gaps appear. Most papers on persuasive design or stress-management apps do not use AI to give deep and personalized health insights or predict chronic-disease risk. They also rarely measure a user's emotional state alongside physical metrics which missing a full view of a person overall health well-being. Research on gamification shows it can boost short-term engagement but few studies test whether badges, leaderboards, or challenges really keep users coming back over the long term. However, there is no proof for the researcher to proof it so the developers cannot know which game elements truly help people adopt healthier habits and also what are the methods to increase the user engagement to use the application.

2.5 Proposed Solutions

To address this problem, the proposed mobile health application will have several features that solving the existing problem encountered from the application and research paper.

Firstly, the app will use developed and trained ML algorithms to analyse users' health data and predict their risk of common chronic diseases which we target for cardiovascular disease (CVD) which can lead to heart attack, diabetes and obesity. We will train the ML algorithms on large and reliable datasets that include real patient measurements such as blood sugar level, BMI and frequency of meal consumption. When a user enters their personal health data with our application, the ML and AI will compare it to known the user patterns and deliver a clear risk score which are low, moderate or high risk along with simple and personalised advice on how to lower their risk.

Next, to keep the users engaged, the application will have a feature of friendly chatbot that acts as a personal health assistant. This chatbot will answer questions about related on healthcare and mental health. The chatbot remembers the conversation history which increases user satisfaction and makes the conversations feel more natural as if chatting with a personal virtual health assistant. Indirectly, this will improve user engagement which encouraging users to interact with the chatbot for their enquiries in the future. By talking naturally with the chatbot, users will feel supported and motivated to stick with healthy habits.

Moreover, the app will also track emotional well-being. The feature will be using on device AI camera-based facial analysis and also an optional of user to let user do various of mental health assessment and describe the user current mood by using text. In this case, it can detect how a user feels happy, stressed or sad and offer personalised tips to improve their mood. All mood result will be stored in the database so the user able to view trends over weeks or months by helping them understand how their feelings and physical health connect.

Finally, the app's interface will adapt to each user's language and needs. It allows users to choose the language they are most comfortable with when using the application, offer large with clear text and include voice navigation so that anyone even those with limited literacy can tap, speak or listen to guidance. This culturally sensitive, accessible design ensures that a wide range of Malaysian users can easily track their health, receive AI-driven insights and stay engaged for the long term

Chapter 3

System Methodology/Approach

During this project, each module will be developed by following the Software Development Life Cycle (SDLC) guidelines and using the Agile method as the development approach for this Mobile Health Application.

3.1 Agile Methodology

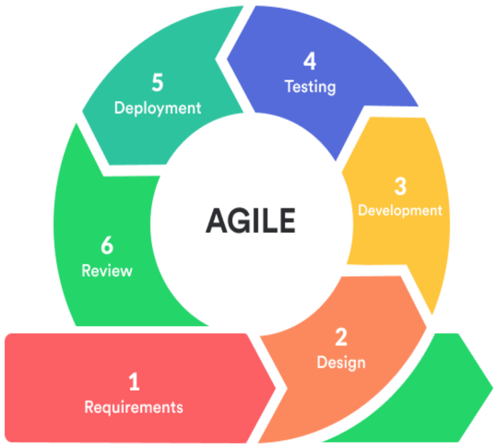


Figure 3.1.1.0 Agile Methodology Development Technique

1. **Requirement Phase**

In this requirement phase, we will start by gathering everything that we need to build the applications. In other words, during this phase, we have a laptop with Android Studio installed, setting up our Firebase database and also Firebase Storage, choosing Kotlin as the programming language and choose ML development tool which is TensorFlow Lite for our ML models.

2. **Design Phase**

Next, we turn those requirements into simple screen sketches and diagrams for system architecture design. For screen sketches, we will be using Figma to draw each screen such as login, dashboard, risk results, chat, mood check and settings. We also map out how data moves for example from Google Fit into our risk engine by using draw.io for system architecture design for our applications. Once we drawn finished, we review these sketches, fix anything confusing and only move on when the screens and also the diagram make sense.

3. **Development Phase**

In this phase, we will start to build the app in small steps called sprints. In each sprint we add one feature such as firstly add the risk checker, then the chatbot, after that is mood detection and finally the adaptable interface. We will write the code and connect with the ML models and make sure each part able to connect with Firebase correctly. Each sprint ends with a quick demo to show what works.

4. **Testing Phase**

After each sprint, we will test the new feature. We will pretend as the user and try the real actions such as logging in, syncing health data, running a risk check, chatting with chatbot and checking mood. Once we tested, we will write a simple test in code to catch errors early and fix any bugs right away. Our goal in this phase is to make sure each part works as expected before we build more.

5. **Deployment Phase**

During this phase, when all the features have passed, we will prepare an Android Device that already contains of our application for the test track. The testers might be from friends or relatives which will install the app and try key flows such as login, risk assessment, chat with chatbot and mood input. In this case, we will watch for crashes, slow load times or installation problems. Any issues found will be fixed before we open the app to more users.

6. **Review Phase**

Finally, we will gather feedback from our testers. We will ask easy questions like "Was the risk score clear?" or "Did the chatbot provide you the accurate respond?". Then, we will look at the app performance such as startup time, AI response speed and jot down the user suggestions. Once we review everything, we will make small fixes such as improve text, adjust AI settings or smooth out the UI. In this case, it will ensure the app is solid and user-friendly before the full release to the real user to use it the application.

3.2 System Design Diagram/Equation

3.2.1 Wireframe Interface Application

In this section, the wireframes design show how the application interface will look starting from the login screen to the user profile module.

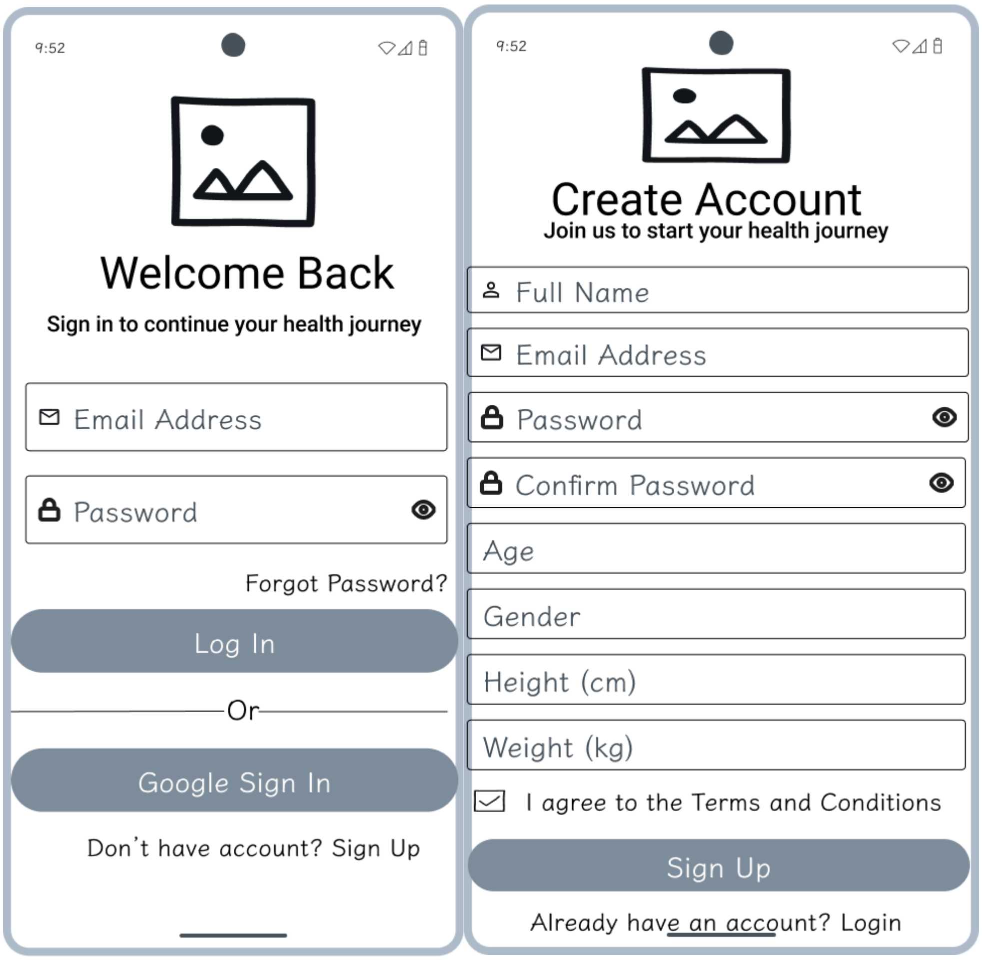


Figure 3.2.1.0 Wireframe Design (Sign up and Login Module)

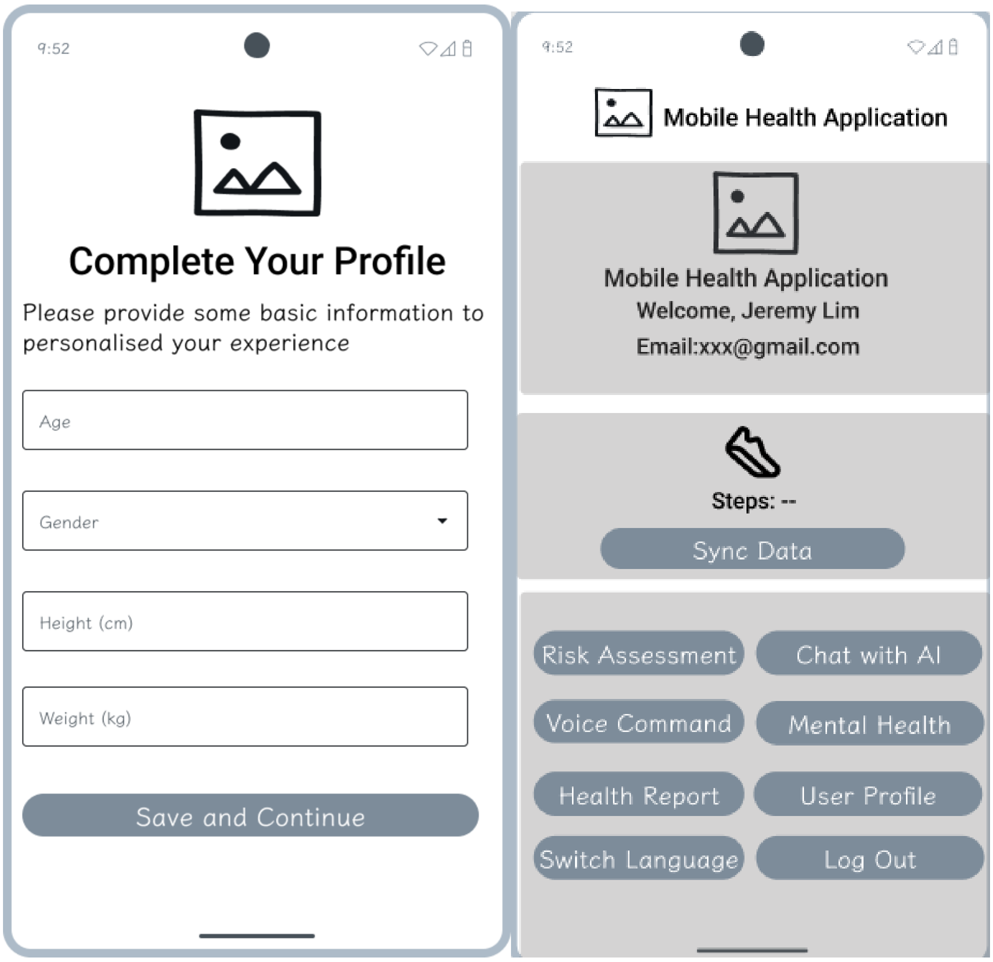


Figure 3.2.1.1 Wireframe Design

(Sign Up for Google Sign-in Method and Main Interface Screen)

Figure 3.2.1.2 Wireframe Design (Chronic Disease Risk and Mental Health Dashboard)

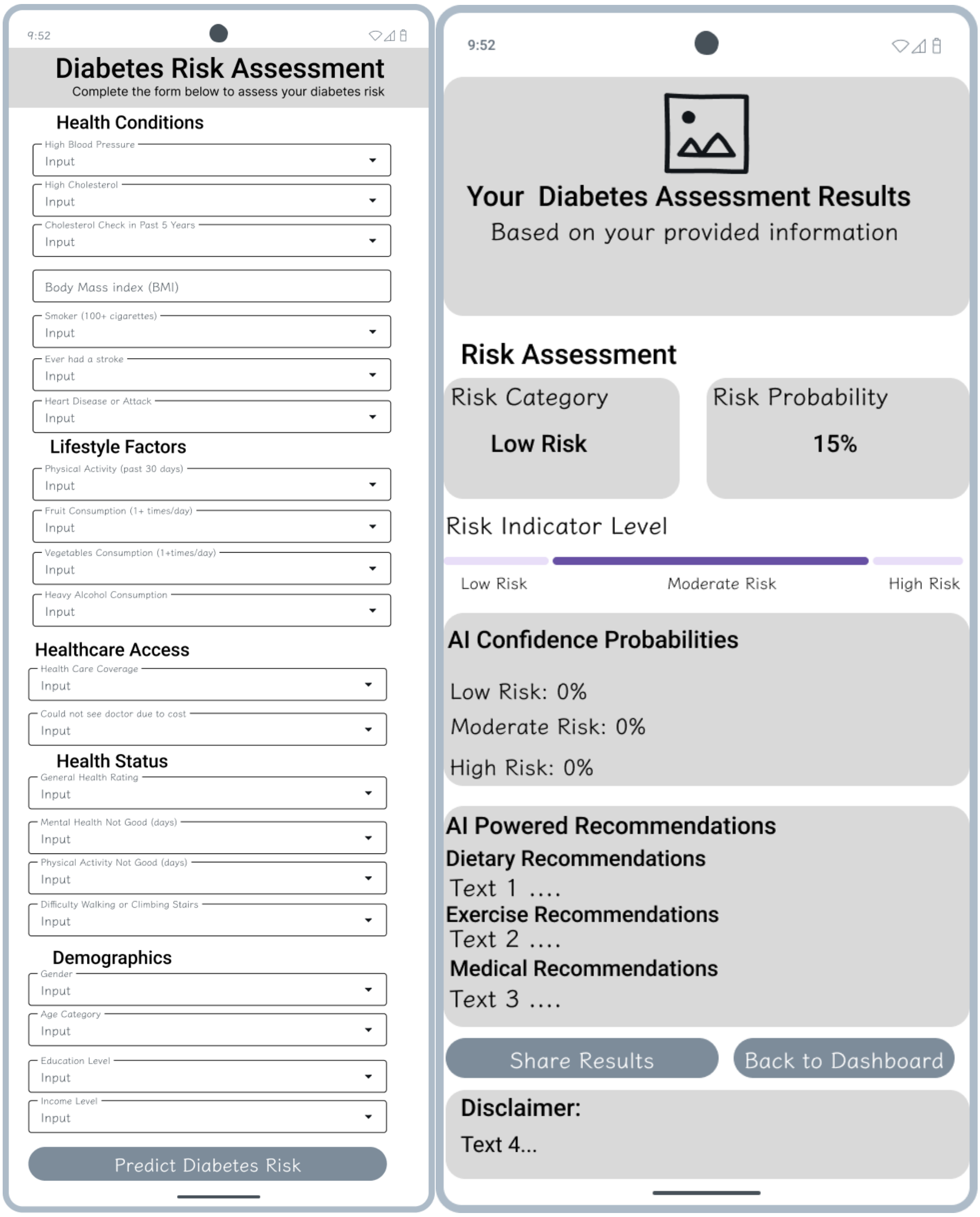


Figure 3.2.1.3 Wireframe Design (Diabetes Risk Interface and Result)

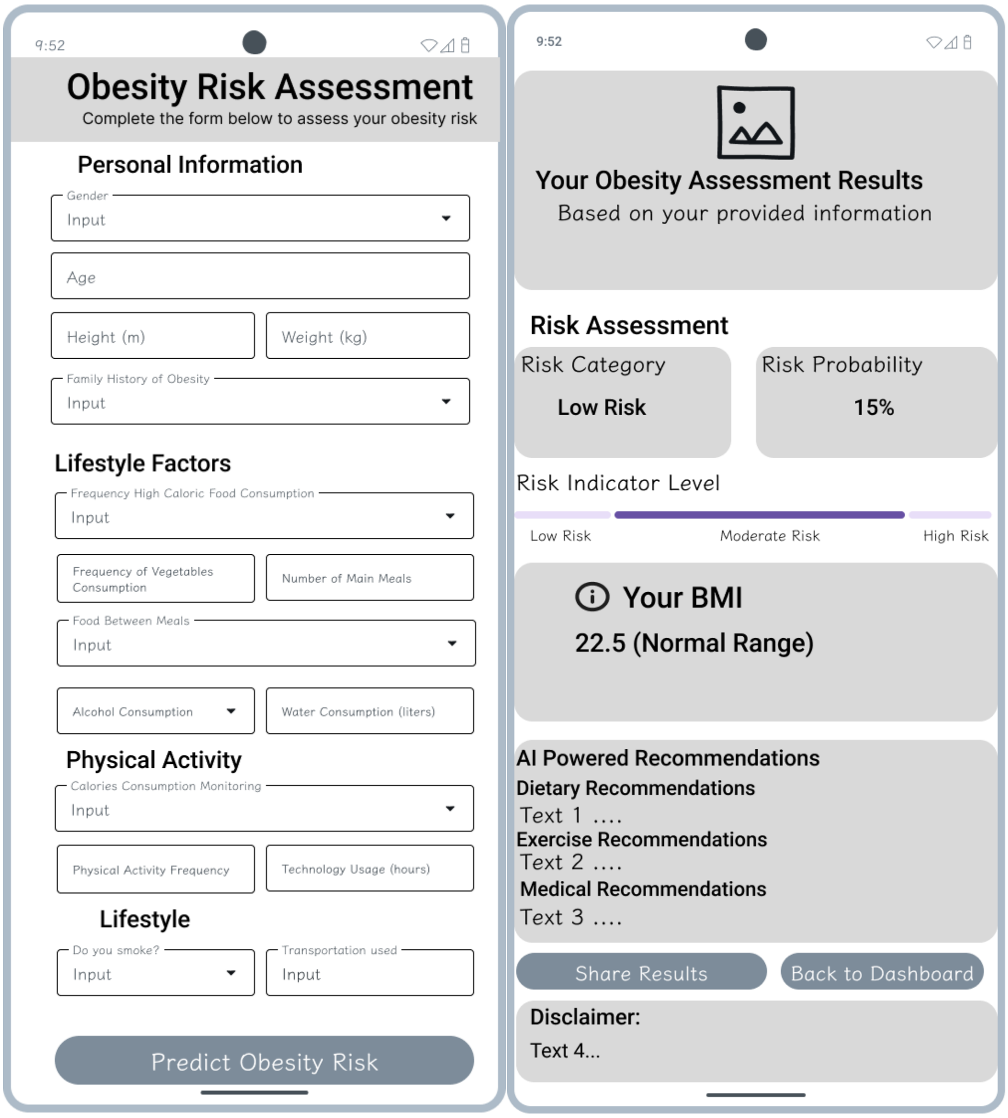


Figure 3.2.1.4 Wireframe Design (Obesity Risk Interface and Result)

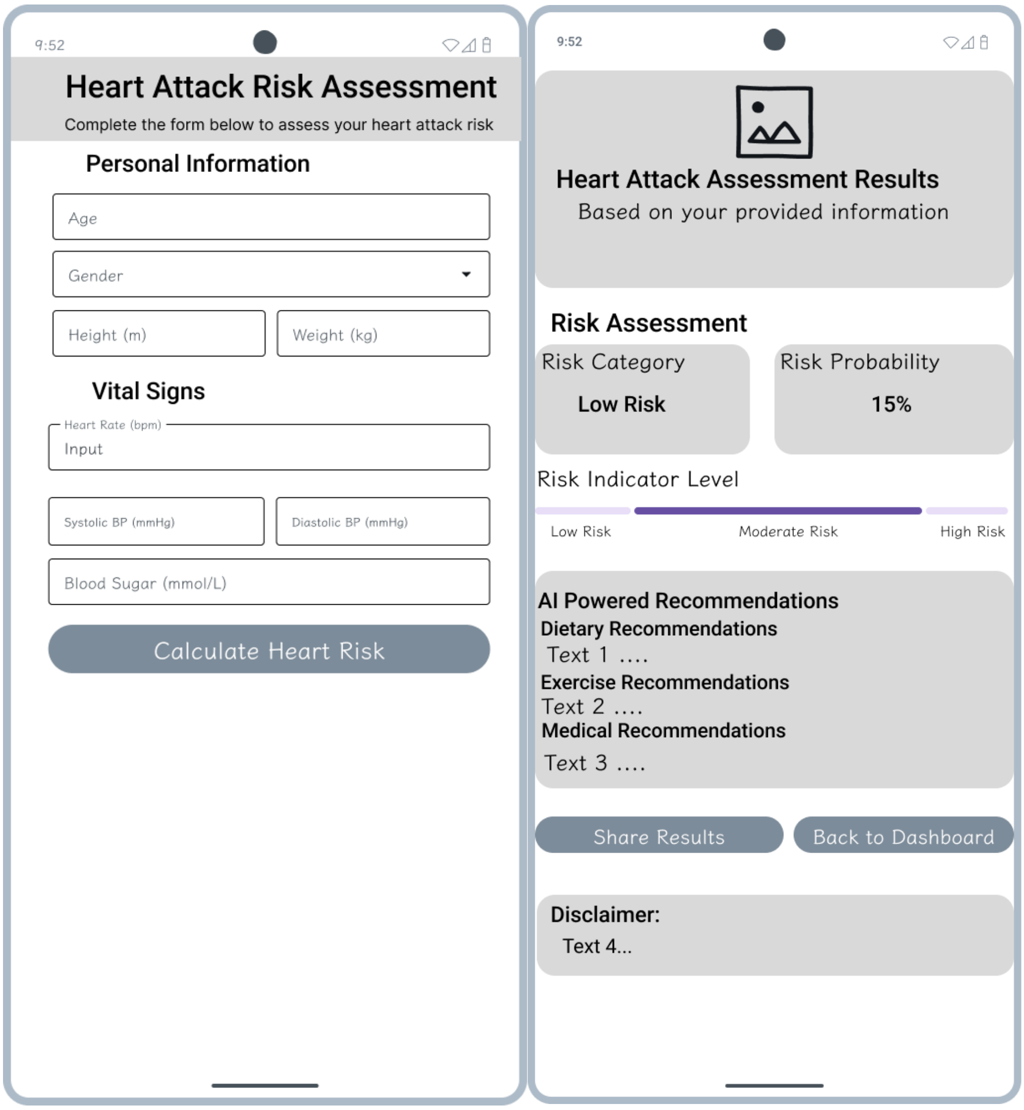


Figure 3.2.1.5 Wireframe Design (Cardiovascular Disease Risk Interface and Result)

Figure 3.2.1.6 Wireframe Design (Chatbot Interface)

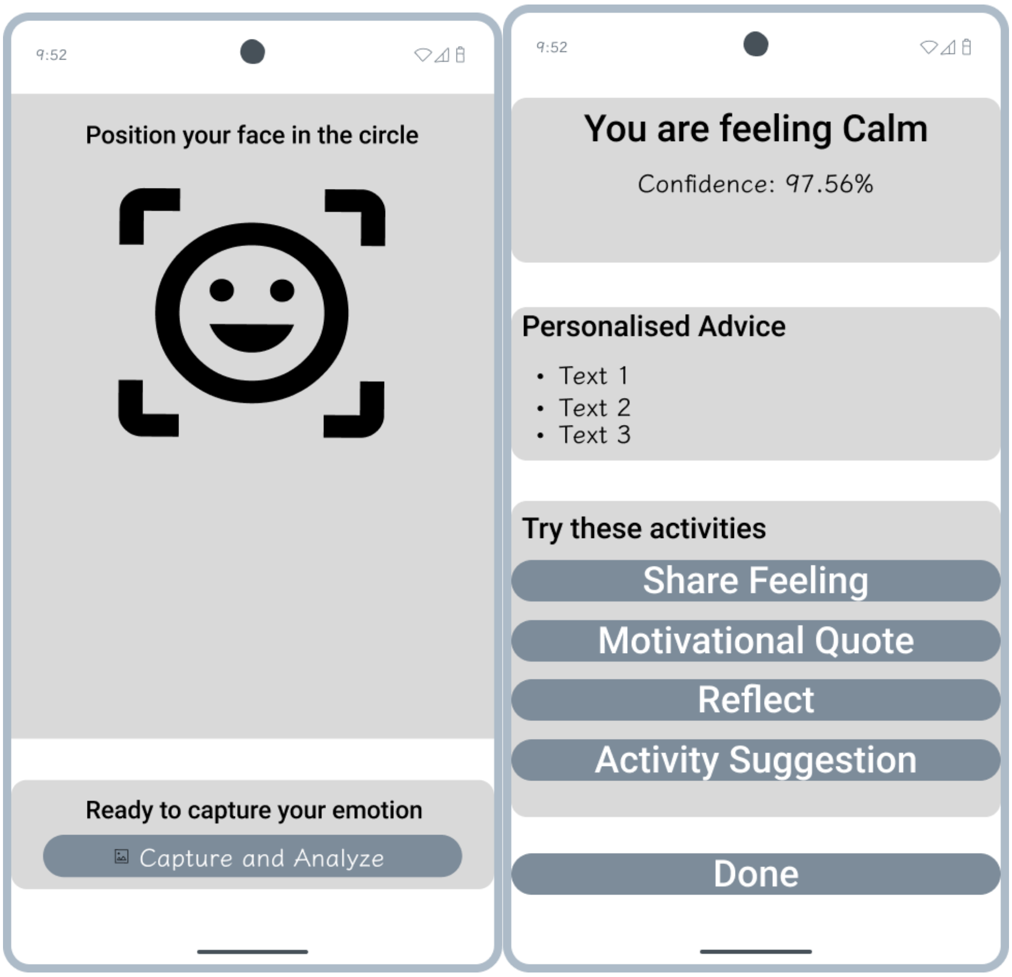


Figure 3.2.1.7 Wireframe Design (Face Recognition Interface and Result)

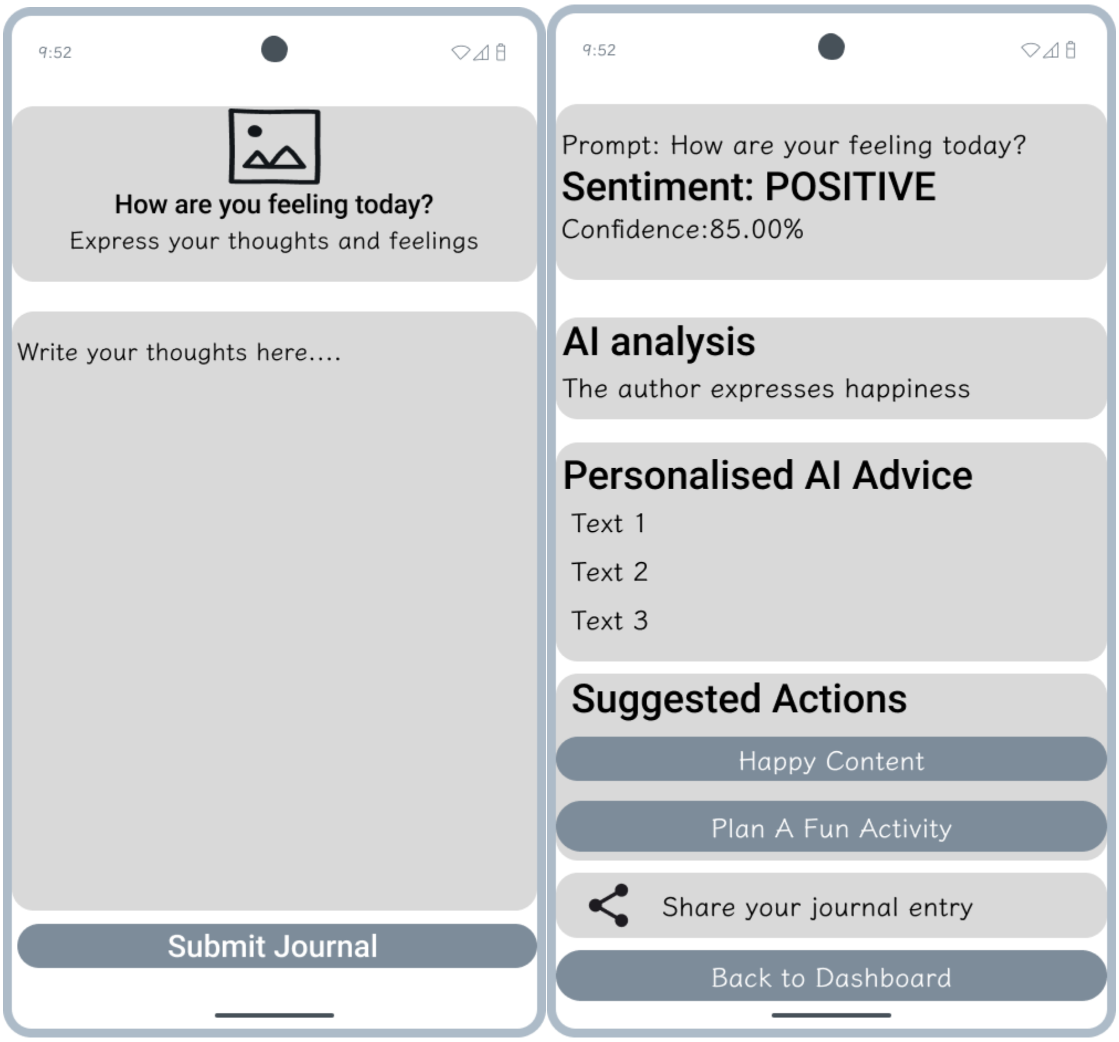


Figure 3.2.1.8 Wireframe Design (Journal Interface and Result)

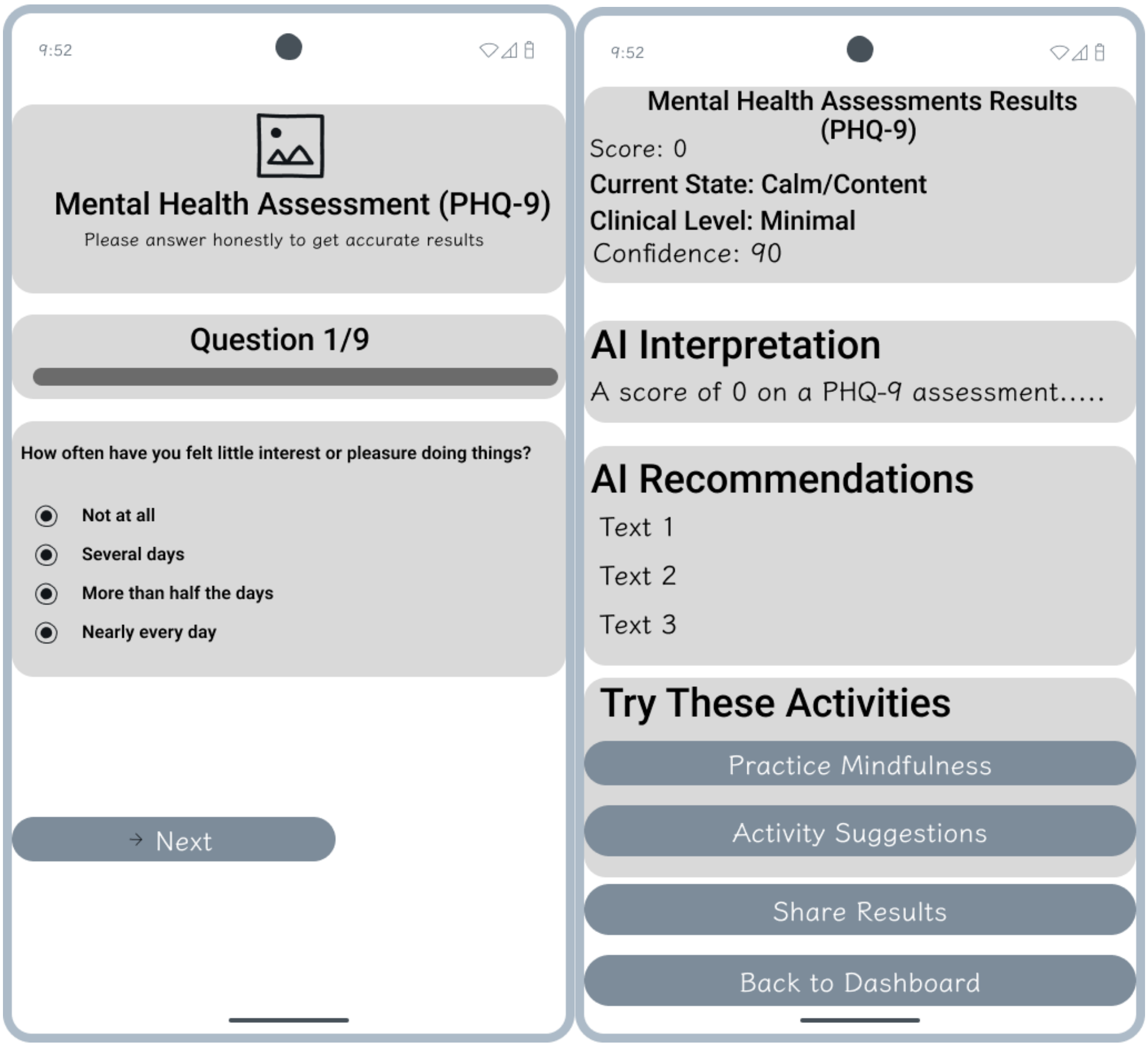


Figure 3.2.1.9 Wireframe Design (Mental Health Assessment Interface and Result)

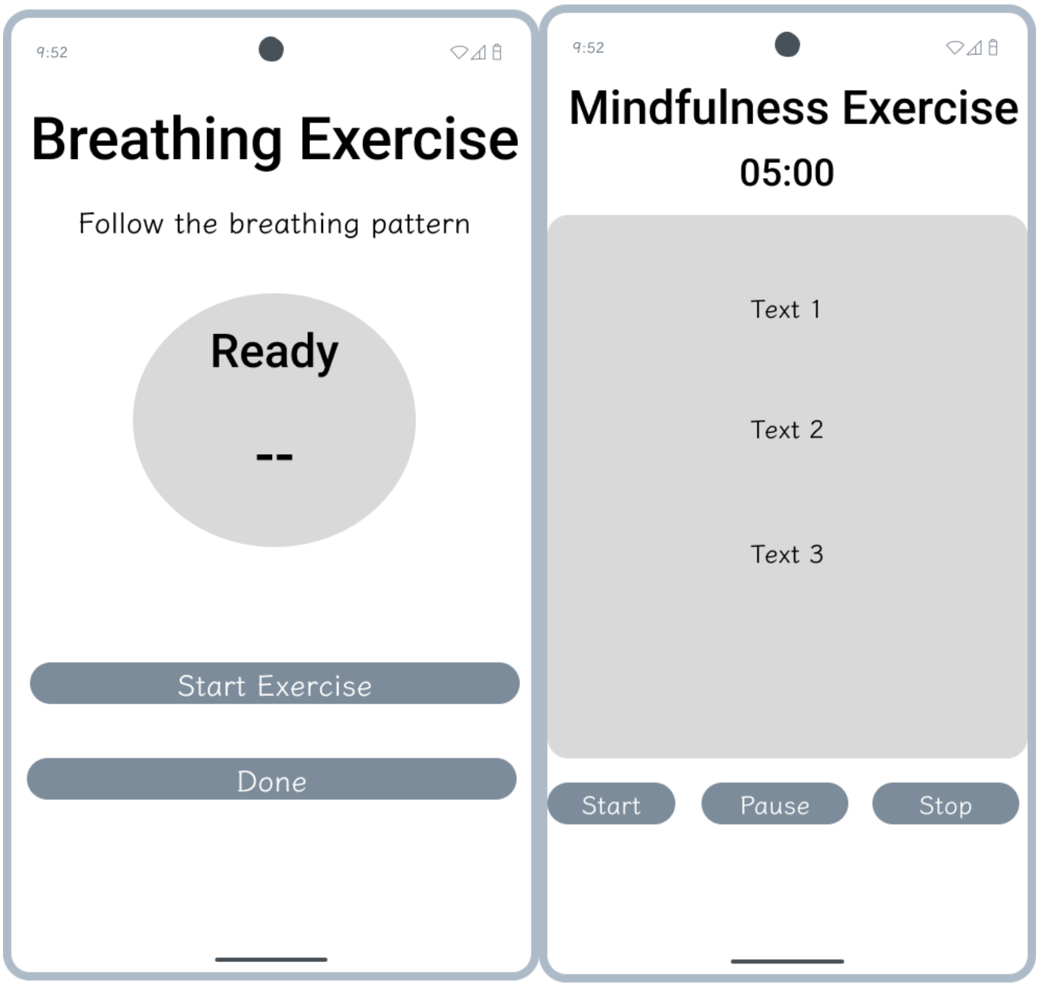


Figure 3.2.1.10 Wireframe Design (Breathing and Mindfulness Exercise Interface)

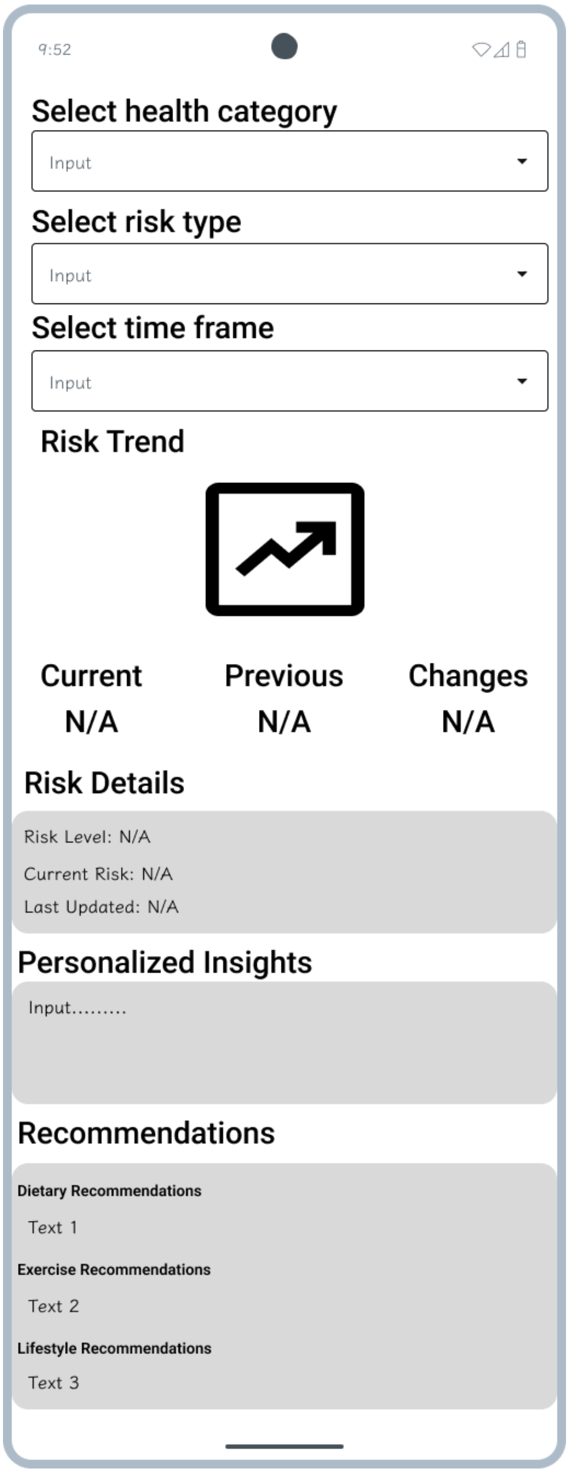


Figure 3.2.1.11 Wireframe Design (Health Report Interface)

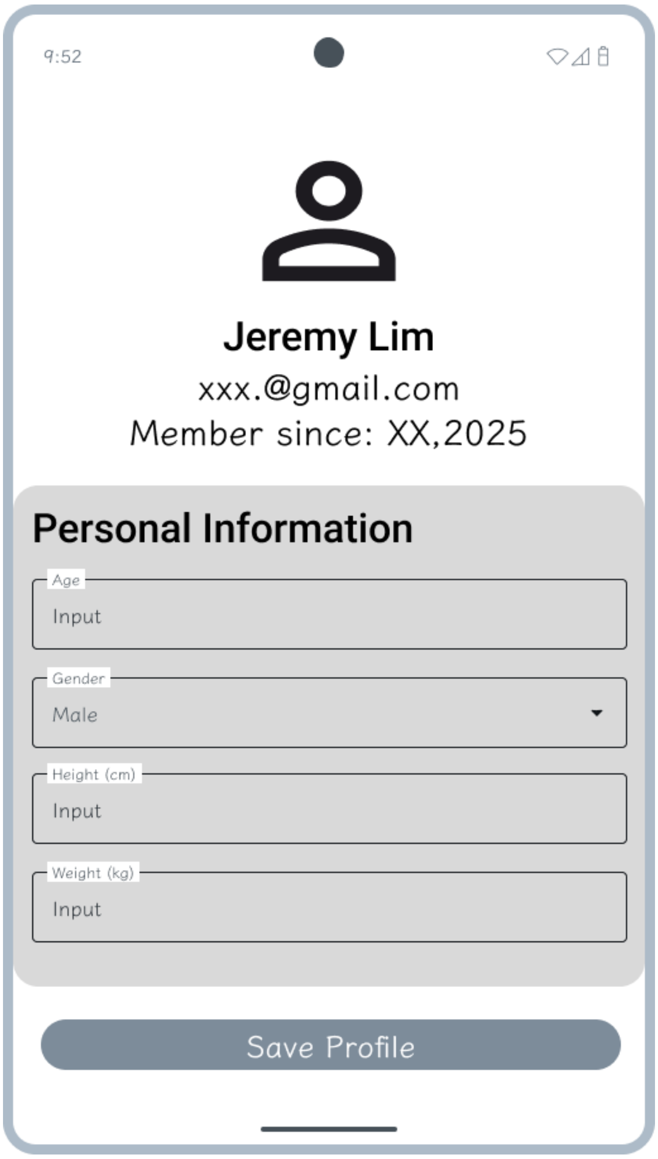


Figure 3.2.1.12 Wireframe Design (Update Profile Interface)

3.2.2 System Equation

3.2.2.1 Body Mass Index (BMI) [44]

Formula

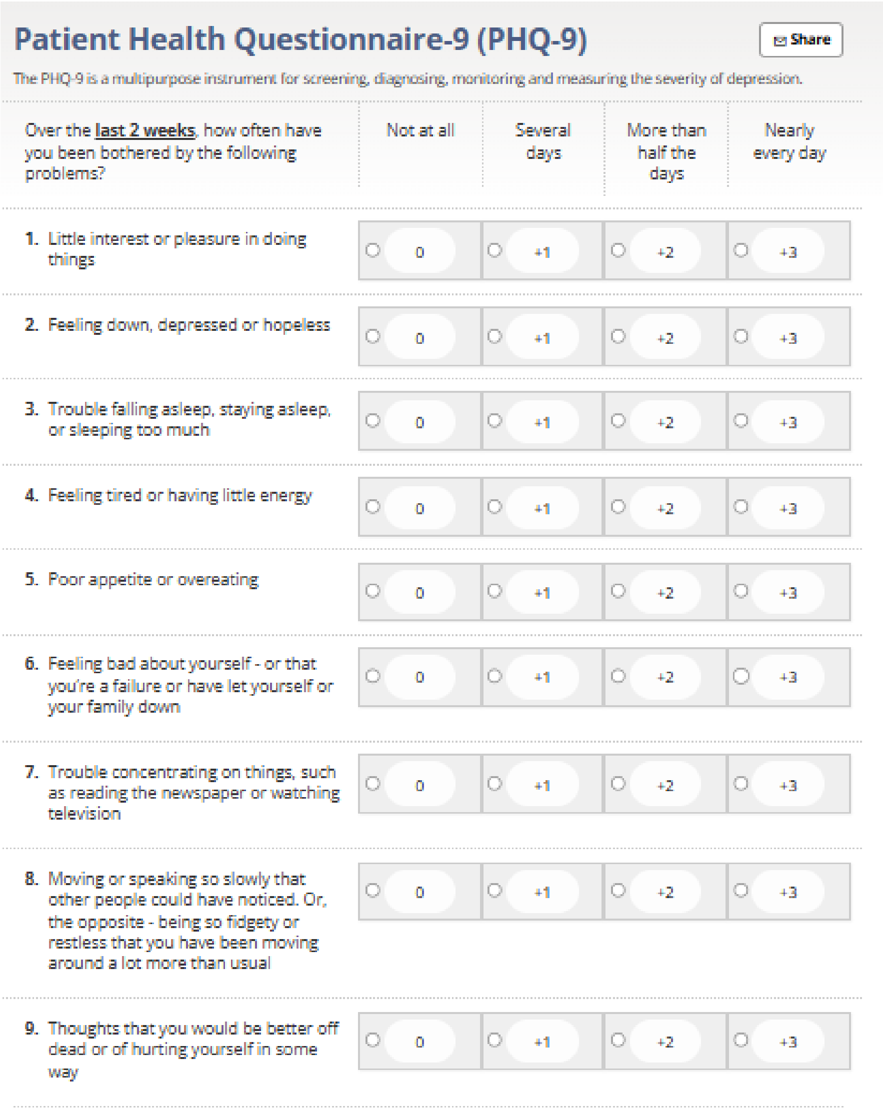
$$\frac{\boldsymbol{weight}(\boldsymbol{kg}) \times \boldsymbol{weight}(\boldsymbol{kg})}{\boldsymbol{height}(\boldsymbol{cm}) \times \boldsymbol{height}(\boldsymbol{cm})}$$

BMI Category	BMI Range (kg/m^2)
Underweight	Less than 18.5
Healthy Weight	18.5 to less than 25
Overweight	25 to less than 30
Obesity	30 or greater
Class 1 Obesity	30 to less than 35
Class 2 Obesity	35 to less than 40
Class 3 Obesity (Severe Obesity)	40 or greater

Figure 3.2.2.1 BMI Category

BMI is a one of the most famous and simple way to check if your weight is healthy based on your current height. It is calculated by dividing your weight (in kilograms) by the square of your height (in meters). The purpose of using BMI is to quickly estimate whether you are underweight, normal weight, overweight or in obesity category which can help identify potential health risks related to weight, such as heart disease or diabetes.

3.2.2.2 Patient Health Questionnaire-9 (PHQ-9) [45]



Patient Health Questionnaire-9 (PHQ-9) Share

The PHQ-9 is a multipurpose instrument for screening, diagnosing, monitoring and measuring the severity of depression.

Over the **last 2 weeks**, how often have you been bothered by the following problems?	Not at all	Several days	More than half the days	Nearly every day
1. Little interest or pleasure in doing things	0	+1	+2	+3
2. Feeling down, depressed or hopeless	0	+1	+2	+3
3. Trouble falling asleep, staying asleep, or sleeping too much	0	+1	+2	+3
4. Feeling tired or having little energy	0	+1	+2	+3
5. Poor appetite or overeating	0	+1	+2	+3
6. Feeling bad about yourself - or that you're a failure or have let yourself or your family down	0	+1	+2	+3
7. Trouble concentrating on things, such as reading the newspaper or watching television	0	+1	+2	+3
8. Moving or speaking so slowly that other people could have noticed. Or, the opposite - being so fidgety or restless that you have been moving around a lot more than usual	0	+1	+2	+3
9. Thoughts that you would be better off dead or of hurting yourself in some way	0	+1	+2	+3

Figure 3.2.2.2.1 Patient Health Questionnaire (PHQ-9)

Interpretation

Provisional Diagnosis and Proposed Treatment Actions		
PHQ-9 Score	**Depression Severity**	**Proposed Treatment Actions**
0 – 4	None-minimal	None
5 – 9	Mild	Watchful waiting; repeat PHQ-9 at follow-up
10 – 14	Moderate	Treatment plan, considering counseling, follow-up and/or pharmacotherapy
15 – 19	Moderately Severe	Active treatment with pharmacotherapy and/or psychotherapy
20 – 27	Severe	Immediate initiation of pharmacotherapy and, if severe impairment or poor response to therapy, expedited referral to a mental health specialist for psychotherapy and/or collaborative management

Figure 3.2.2.2.2 Patient Health Questionnaire Scoring (PHQ-9)

The PHQ-9 is a standard clinical tool used to diagnose and measure the severity of depression. It features 9 questions scored from 0 to 3 with a total score of defining severity levels from Minimal to Severe. This allows users to assess their symptoms and monitor changes in their mental health over time [reference]

3.2.2.3 Accuracy [46]

Formula

$$\frac{\boldsymbol{TP + TN}}{\boldsymbol{TP + FP + TN + FN}}$$

where

TP = True Positive

TN= True Negative

FP= False Positive

FN= False Negative

The accuracy formula above is to measures the overall correctness of the ML model. It is important because it helps us evaluate whether the ML model is suitable to integrate into our application.

3.2.2.4 Precision [46]

Formula

$$\frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}}$$

where

TP= True Positive

FP= False Positive

The precision formula above shows how often the ML model positive predictions are correct. It helps us see how many predicted positives are actually wrong and decide whether the ML model performance is good enough.

3.2.2.5 Recall [46]

Formula

$$\frac{TP}{TP+FN}$$

where

TP= True Positive

FN= False Negative

The recall formula above shows the ability of how well ML model finds the actual real positives. A higher recall is better because it means the model correctly detects more of the real positive cases.

3.2.2.6 F1-Score [46]

Formula

$$\frac{\boldsymbol{2\times Precision\times Recall}}{\boldsymbol{Recall+Precision}}$$

The F1-score formula combines precision and recall to measure how well the ML model finds true predictions while avoiding false predictions. It is useful because it shows if the ML model can handle an imbalanced dataset and helps us decide if the model is good enough to use in our application.

3.2.2.7 AUC-ROC Curves

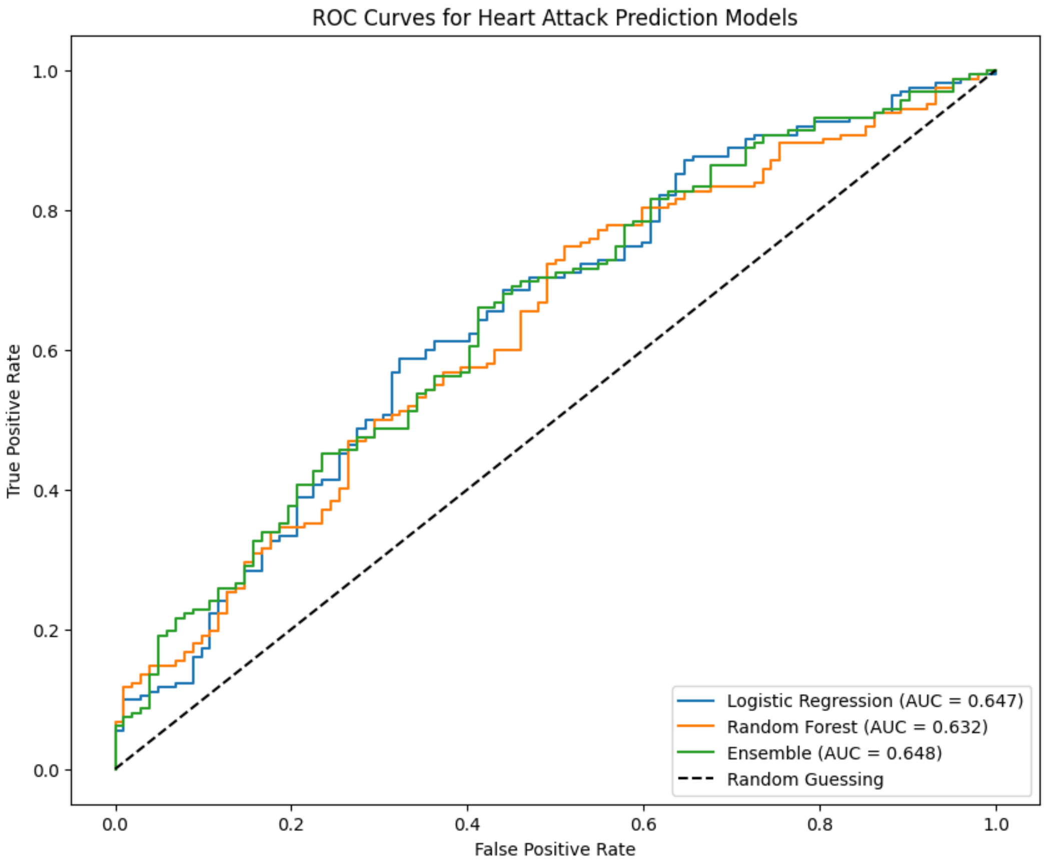


Figure 3.2.2.6.1 Sample Screenshot of ROC Curves

The AUC-ROC curve value shows how well an ML model can tell the difference between correct and wrong predictions. A higher AUC value means the model is better at making this difference and can be considered to have good and stable performance.

3.2.3 System Architecture Diagram

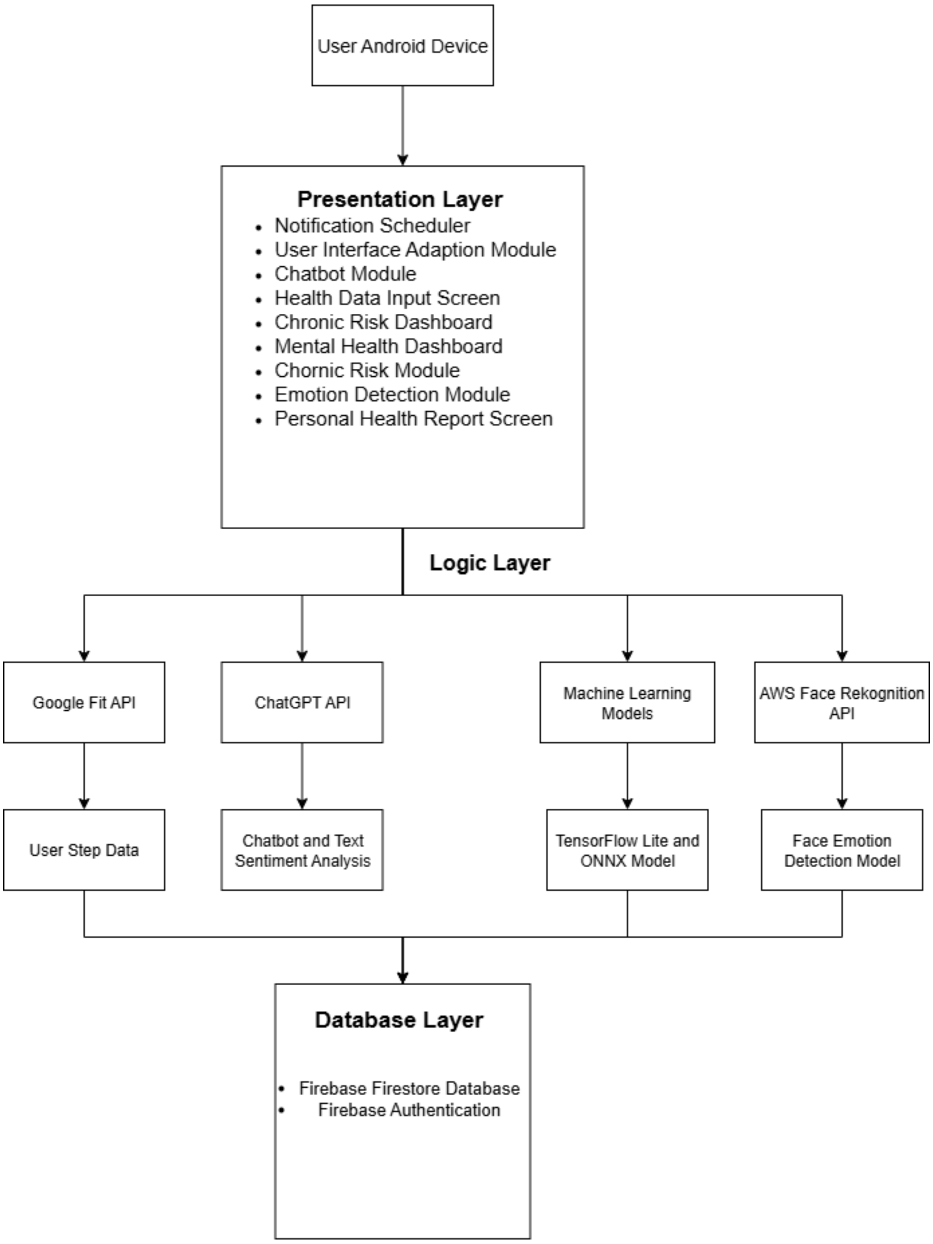


Figure 3.2.3.1 Three-Tier Architecture Diagram

Based on the figure above shows the architecture diagram which use Three-Tier Architecture Diagram for Mobile Health Application. When the user opens the app, the interface will adapts based on the user device and preferences and a built-in scheduler decides when to send gentle reminders for example, “Your mood matters! Consider tracking how you're feeling today”. The login screen uses Firebase Authentication method so that each person’s data stays private. The user’s step count will first be taken from the Google Fit API. If no data is found, then it will check Firebase Firestore to see if the user’s step count data is available there. Next, the user can enter health measurements such as heart rate, blood sugar levels, blood pressure levels and these inputs will feed directly into a tiny machine learning model which called TensorFlow Lite file and ONNX Model file that instantly calculates risk scores for CVD as the leading cause of heart attack, diabetes and obesity risk and displays them in clear, color-coded warnings with personalised advice that obtain from ChatGPT and the data will be stored into Firebase Firestore.

At the same time, the app offers a chat screen powered by ChatGPT via the OpenAI API. Whenever the user asks questions like "Give me some health tips?" or "Suggest an exercise routine for me?", the app will send the text with to ChatGPT with the valid API key service and responses with friendly and personalized reply. Not only that, the same service also handles text sentiment checks such as any free-form message the user types are sent with a prompt asking "Is this positive, neutral or negative?" for simple mood identification. The data will be stored into the Firebase Firestore as the user does not need to worry about past messages because the chatbot remembers the conversation history. This makes the conversation smooth and natural which makes the user feels like they are talking to their own personal health assistant.

For emotion detection, the app provides three methods which giving users options to choose from. First, the user can capture an image using their device camera which later will be sent to AWS Face Rekognition through the API service for emotion analysis. Once the analysis is done, AWS Face Rekognition will sends back the result and the device displays it to the user. If the user does not capture their face, they can share their feelings or thoughts through the journal prompt module. This uses ChatGPT to analyze the text and provide personalized advice along with a motivational message if the user is feeling down. The user also has the option to take a mental health assessment that know about their current mental health condition. After completing the questions, the system calculates the score and sends it to ChatGPT to provide personalized advice. Based on the score, the system will also suggest suitable activities for the user. All the result data will be stored in Firebase Firestore which allowing users to track their mental health over time.

3.2.4 Use Case Diagram and Description

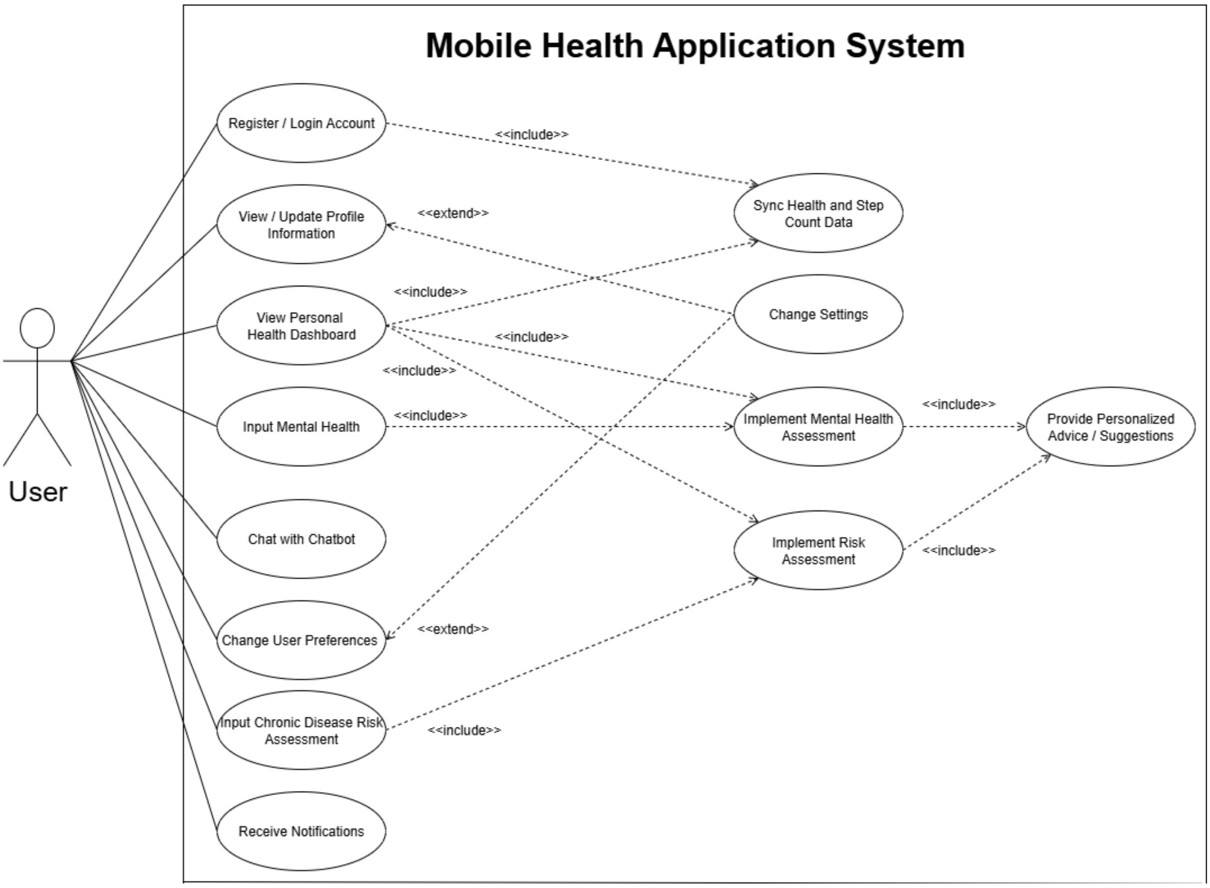


Figure 3.2.4.1 Use Case Diagram

The use case diagram above illustrates the main methods of how a user interacts with the mobile health application by showing each feature as a use case and how they connect to each other. The diagram has a single **User** actor. Every user must first **Register or Login Account** which this includes syncing their health, sleep and step-count data from Google Fit API so that their dashboard is always able up to date. Once the user logged in, the users can **View or Update Profile Information**, receive notifications from mobile health application and changing settings like language preferences that extends from this use case.

From the user profile or dashboard, users **View Personal Health Dashboard** which an action that includes both data sync and running the core features in this mobile health application which are **Implement Risk Assessment** and **Implement Mood Assessment** of their routines. When the dashboard displays risk scores and mood results, it is also **Provide Personalized Advice or Suggestions** based on those assessments. Users may also able to choose **Input Mental Health condition** which this directly includes the mood assessment use case.

For conversational support, users can **Chat with the Chatbot** and ask questions related to healthcare and mental health. The chatbot will respond based on the user's questions and users can also follow up because the chatbot remembers the conversation history. Finally, users can **Change User Preferences** at any time which this extends from viewing or updating their profile information and switching the application language based on their preferences.

In conclusion, the use case diagram shows that every high-level action which are dashboard viewing, mood input, chatbot, risk check that relies on underlying services which are data sync, assessments, advice and that changing settings or profiles can extend or modify those behaviours. This clear structure diagram ensures each feature is connected to the right processes and that users are always get up-to-date data, personalized insights and full control over their preferences.

3.2.5 Activity Diagram

3.2.5.1 Sign-Up Module

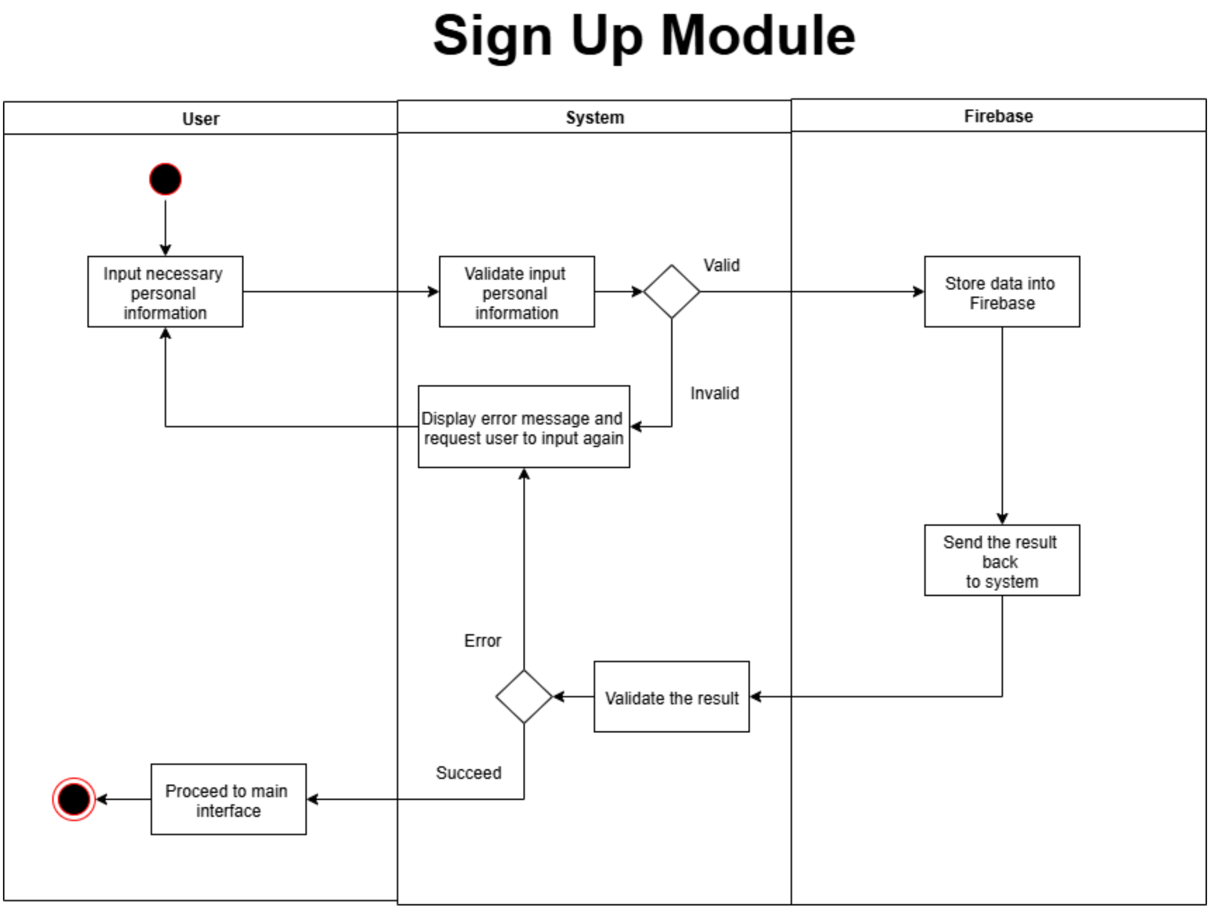


Figure 3.2.5.1 Activity Diagram (Sign Up Module)

The activity diagram above shows the actors involved in the Sign-Up module and the process of creating account in this application. In this module, the User, System and Firebase will interact each other in order to complete the sign-up process. Each actor has its own functions and decisions to ensure data flows smoothly between them. Regardless of the sign-up method such as Google Sign-In, the account creation will go through this Sign-Up module which to ensure the efficiency of the application.

3.2.5.2 Log-In Module

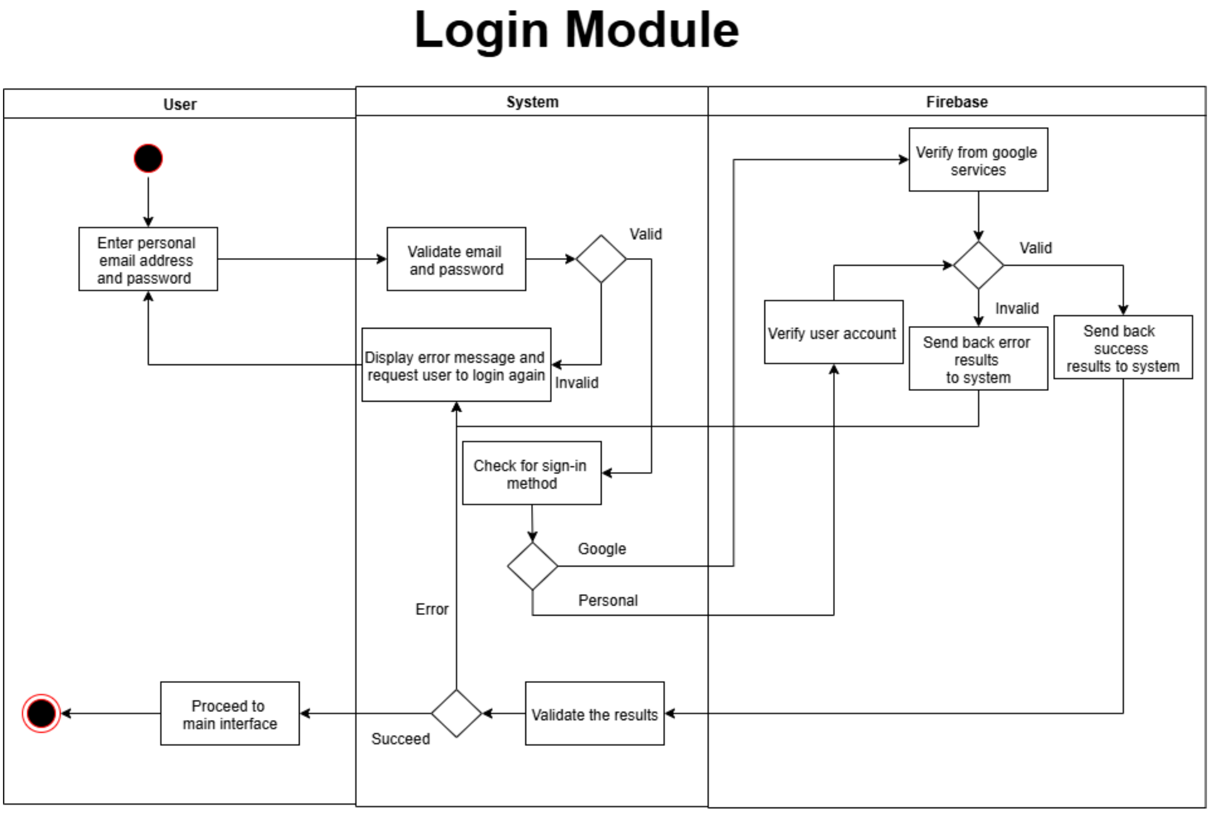


Figure 3.2.5.2 Activity Diagram (Login Module)

The activity diagram above shows the authentication process that verifies users before they access the application. In this module, the actors interact to ensure authentication runs smoothly and to prevent anonymous users from entering and misusing the app. Each actor has specific functions and decision points to keep the system and data flow efficient. All login methods will follow this authentication flow as to ensure the system able to work smoothly.

3.2.5.3 User Profile Module

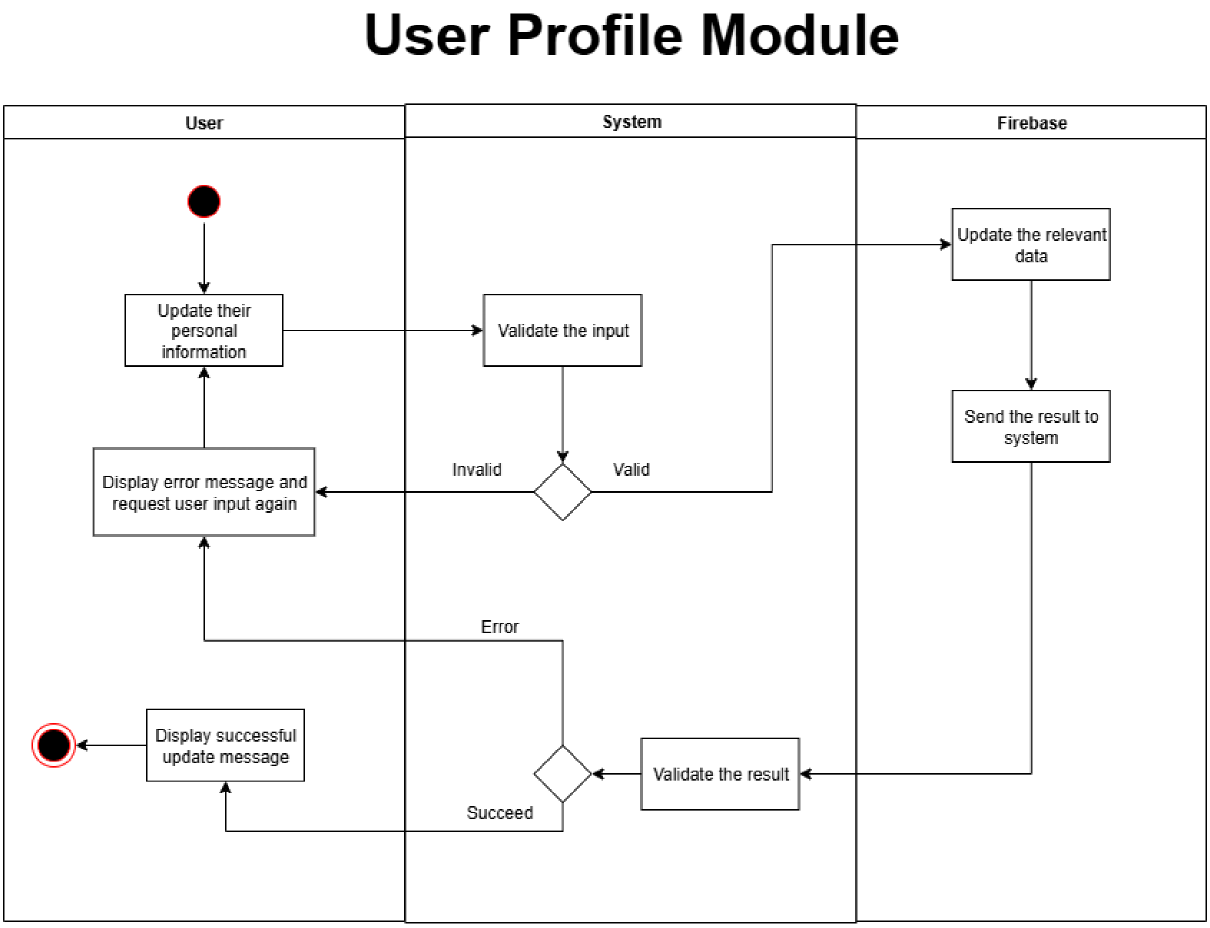


Figure 3.2.5.3 Activity Diagram (User Profile Module)

The activity diagram shows the update personal information process when the user updates their personal information in the application. The user, system and Firebase interact each other to ensure a smooth update and data flow process. Each actor performs its specific tasks and handles errors during the profile update.

3.2.5.4 Diabetes Module

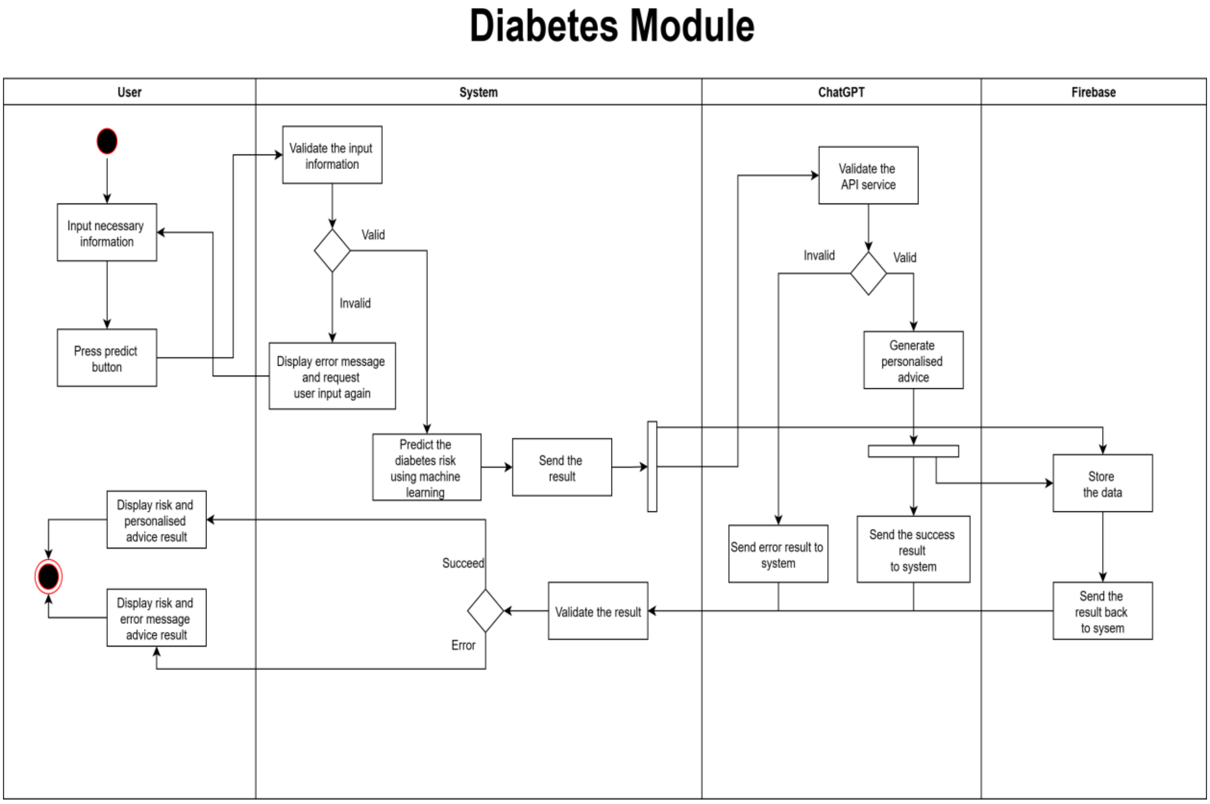


Figure 3.2.5.4 Activity Diagram (Diabetes Module)

The activity diagram above shows the process and data flow for the Diabetes module. In this module, the user, the system, ChatGPT as an external service and Firebase interact to ensure a smooth user experience. The Diabetes module uses a developed machine-learning model to predict the user's diabetes risk either is in low, moderate or high risk based on the information provided. The prediction results are sent to ChatGPT to generate a personalised advice message and to Firebase for storage and tracking so users can monitor their diabetes status at any time. Each actor has specific functions and error to be handle to keep the process smooth and efficient of the application.

3.2.5.5 Cardiovascular Disease Module

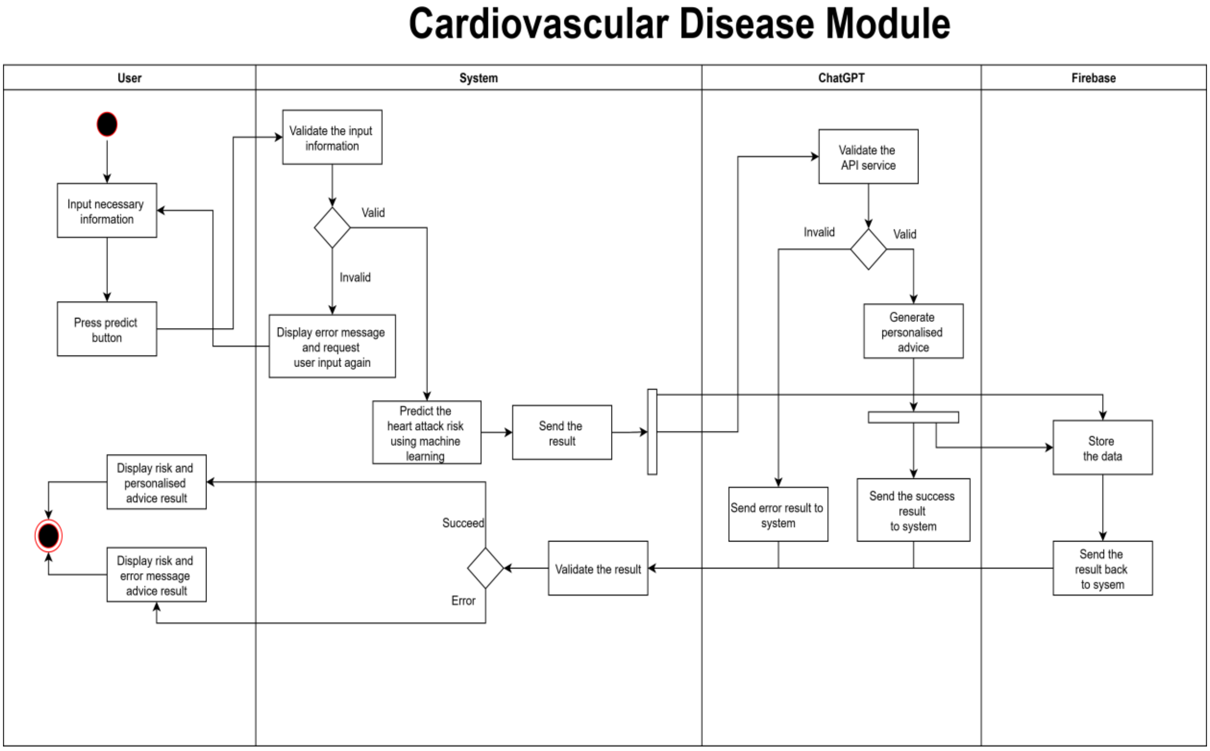


Figure 3.2.5.5 Activity Diagram (Cardiovascular Disease Module)

The activity diagram above shows the process and data flow for the Cardiovascular Disease module. The prediction workflow starting from user input to system display of results is similar to the Diabetes module including the error handling also. The main difference is that each module uses its own trained machine-learning model and a disease-specific dataset. The Cardiovascular Disease module predicts the user's risk level whether in low, moderate or high risk based on the information provided and the result is processed and stored in the same way as the Diabetes module.

3.2.5.6 Obesity Module

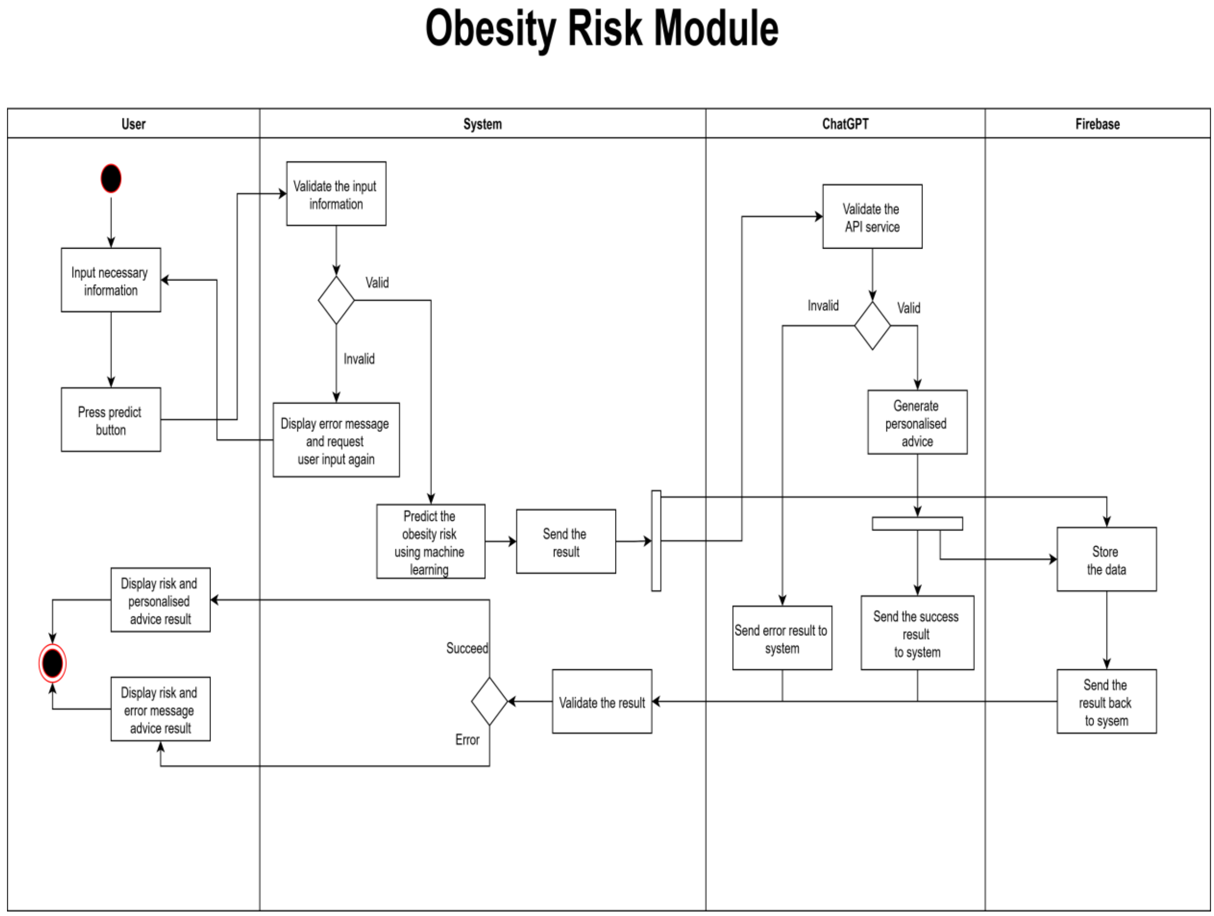


Figure 3.2.5.6 Activity Diagram (Obesity Module)

The activity diagram shows the process and data flow for the Obesity module. The overall workflow process starting from user input to result display which is similar to the Diabetes and Heart Attack modules. The main difference is that the Obesity module uses its own trained machine-learning model to predict obesity risk. The obesity result links the risk levels either in low, moderate or high risk to BMI categories that specifically used by the obesity module. For example, a user with low obesity risk would falls into the Normal Weight BMI category.

3.2.5.7 Chatbot Module

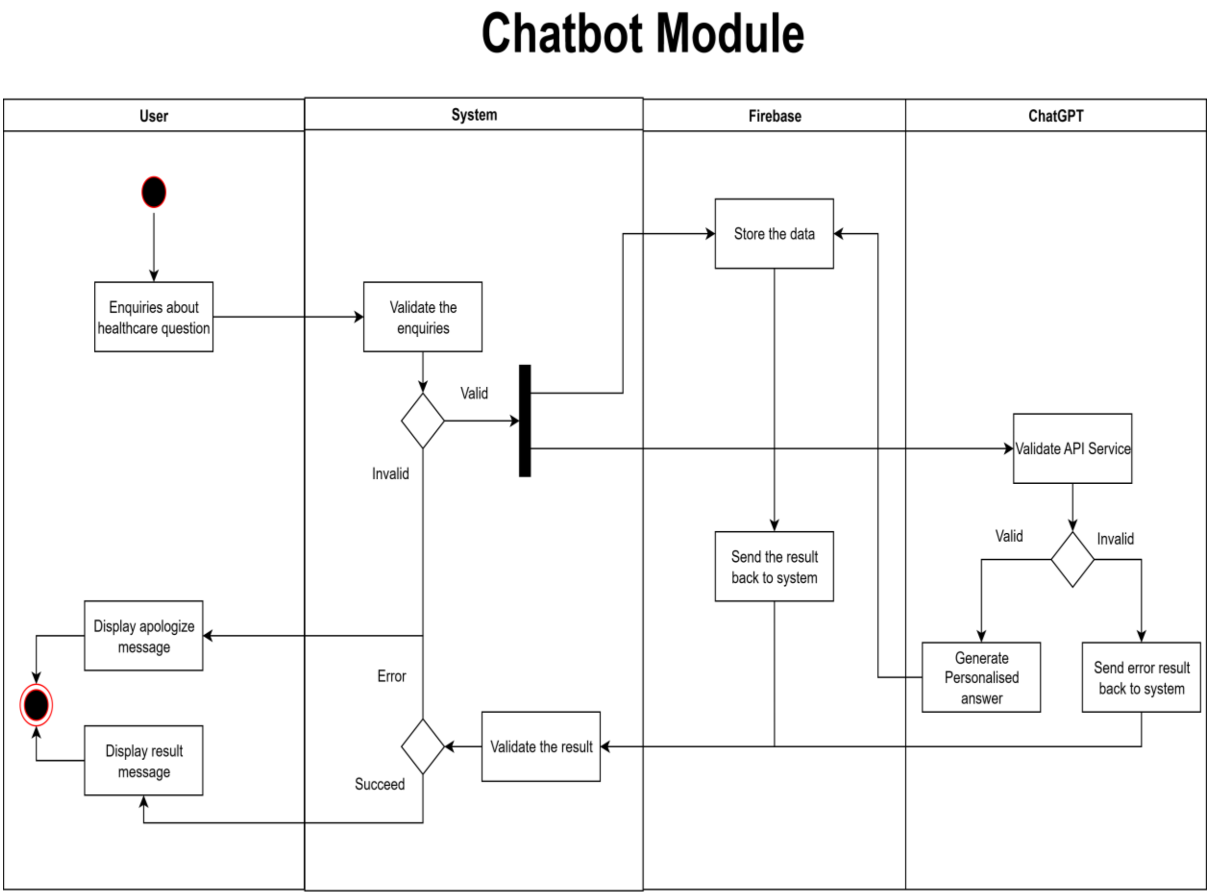


Figure 3.2.5.7 Activity Diagram (Chatbot Module)

The activity diagram above shows the chatbot module overall process flow. The chatbot involves the user, system, Firebase, and ChatGPT service and they interact between each other to handle user queries. When users ask healthcare questions, the system validates the input. If the input is valid, the system sends it to Firebase and ChatGPT at the same time otherwise it will show an error message. Firebase and ChatGPT perform their tasks and return results to the system. The system then validates the results and displays an appropriate response to the user.

3.2.5.8 Face Recognition Module

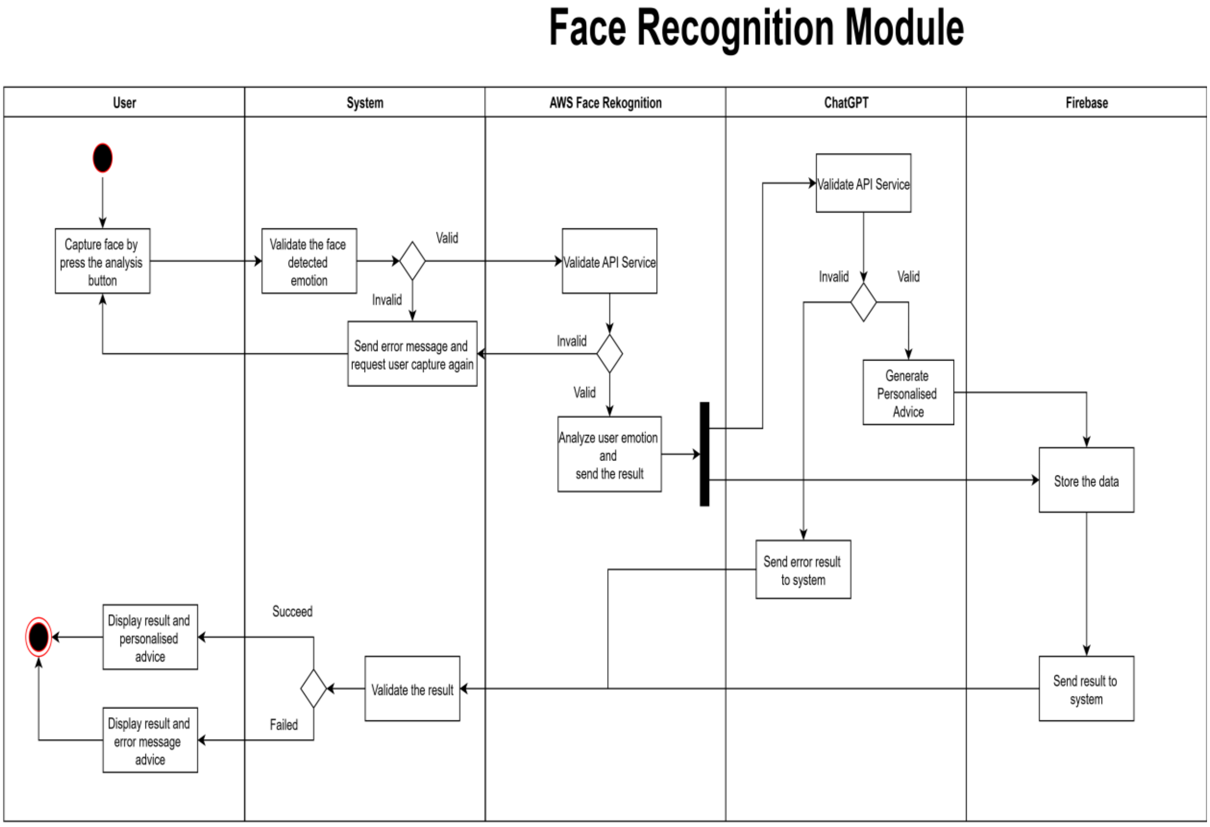


Figure 3.2.5.8 Activity Diagram (Face Recognition Module)

The activity diagram shows the face recognition module's process and data flow. Actors include the user, the system, AWS Rekognition, ChatGPT and Firebase. The user captures a face image with the device camera and sends it to the system for validation. If the user image is valid, the system will send the image to AWS Rekognition which verifies the API and the image. If AWS Rekognition validate the image, it would perform emotion analysis and returns the result otherwise it sends an error and the system will request the user to capture the image again. ChatGPT then generates a personalised advice message based on the emotion analysis. Lastly, the result and advice will return to the system, stored in Firebase and displayed to the user.

3.2.5.9 Journal Prompt Module

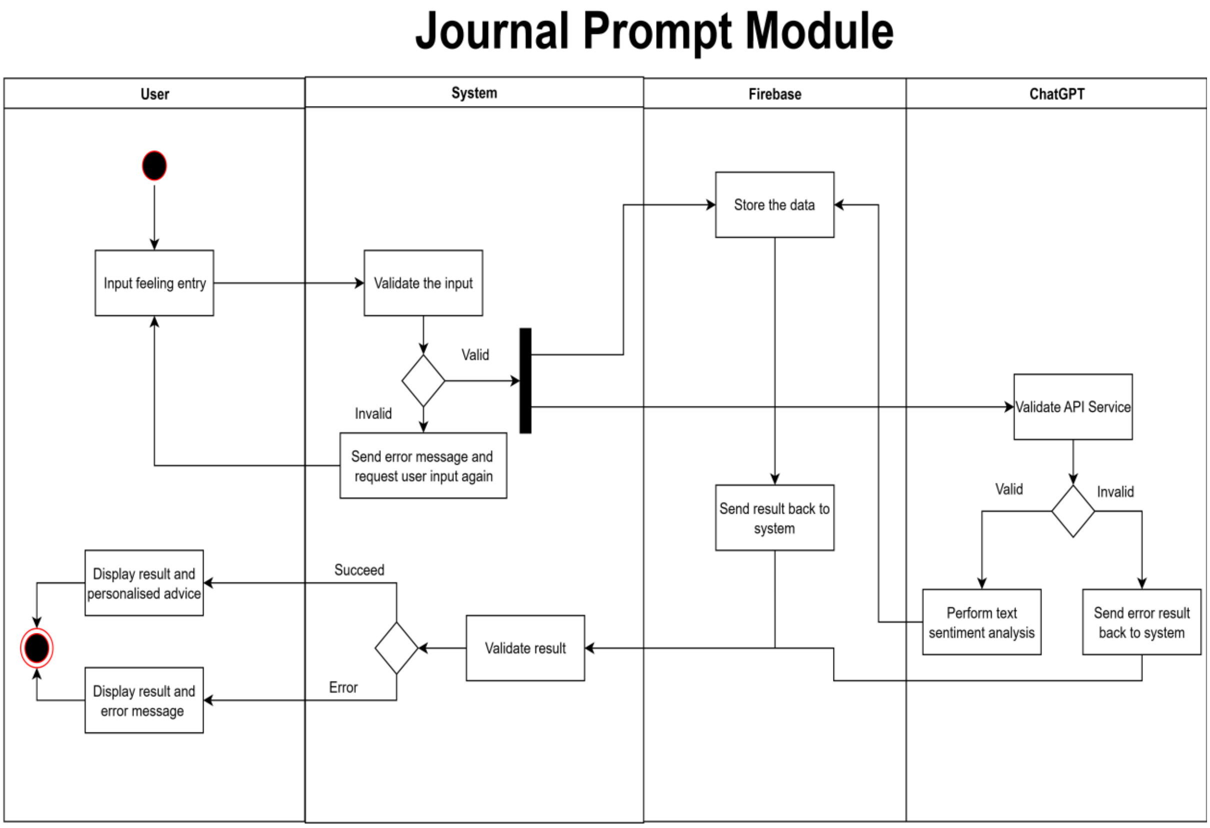


Figure 3.2.5.9 Activity Diagram (Journal Prompt Module)

The activity diagram shows the data and process flow between the User, System, Firebase and ChatGPT. The user inputs their feelings or expression and sends it to the system. The system validates the input and then stores it in Firebase and sends it to ChatGPT for text-sentiment analysis. ChatGPT will verifies the API service, analyzes the text and returns the result to Firebase for stored data purpose and to the system. The system validates the result and displays the text sentiment analysis result, personalised advice and suggested activities to the user. If any error occurs, the system will show an error message or ask the user to submit their input thoughts again.

3.2.5.10 Mental Health Assessment Module

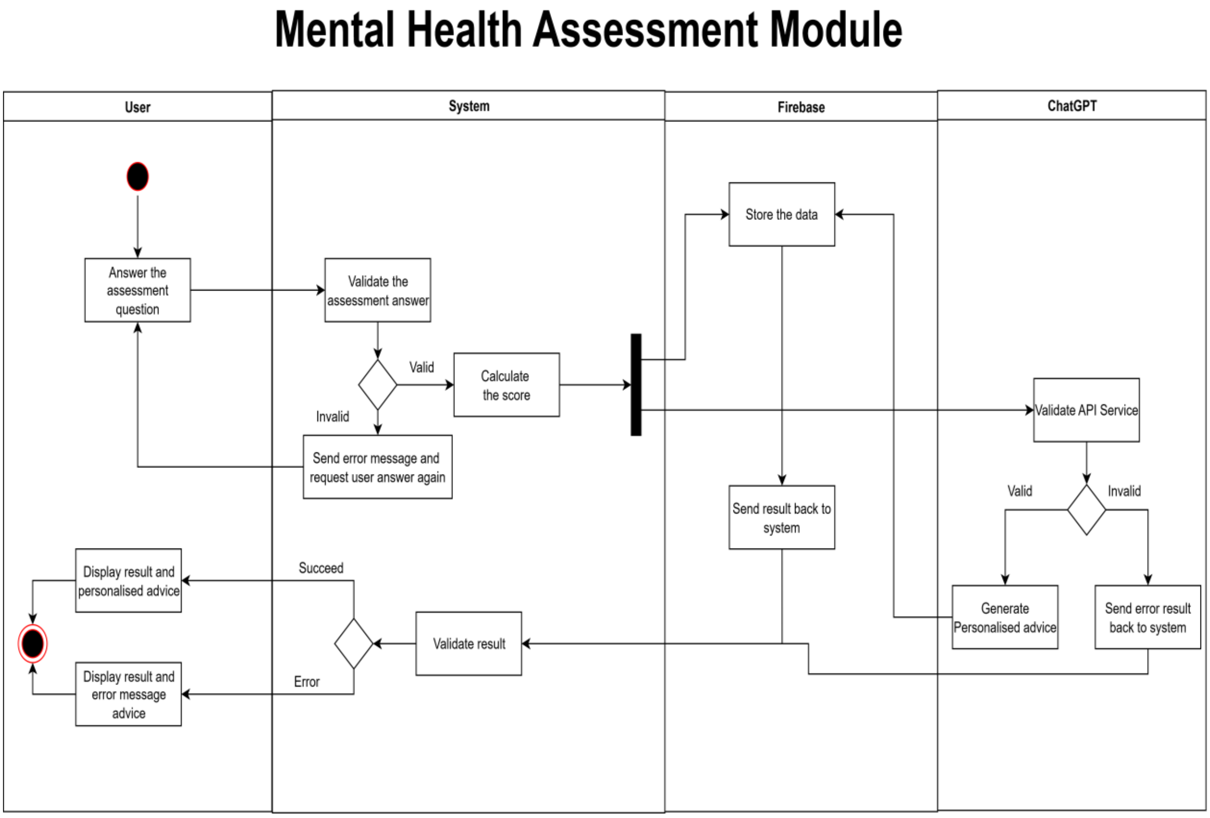


Figure 3.2.5.10 Activity Diagram (Mental Health Assessment Module)

The activity diagram above shows the process and data flow for the Mental Health Assessment module. The user, system, Firebase and ChatGPT interact each other to ensure the process runs smoothly and efficiently. Each actor has specific functions, logic and decision points made to complete the module. The user answers the assessment questions based on PHQ-9 and sends the results to the system. The system validates the answers, calculates the score based on PHQ-9 scoring system and sends the score to Firebase for store data purpose and to ChatGPT to generate personalised advice. ChatGPT verifies API access, creates the personalised advice and returns it to the system together with Firebase also stores the advice. The system then validates the returned results and displays the score, appropriate messages and personalised advice to the user.

3.2.5.11 Personalised Health Report Module

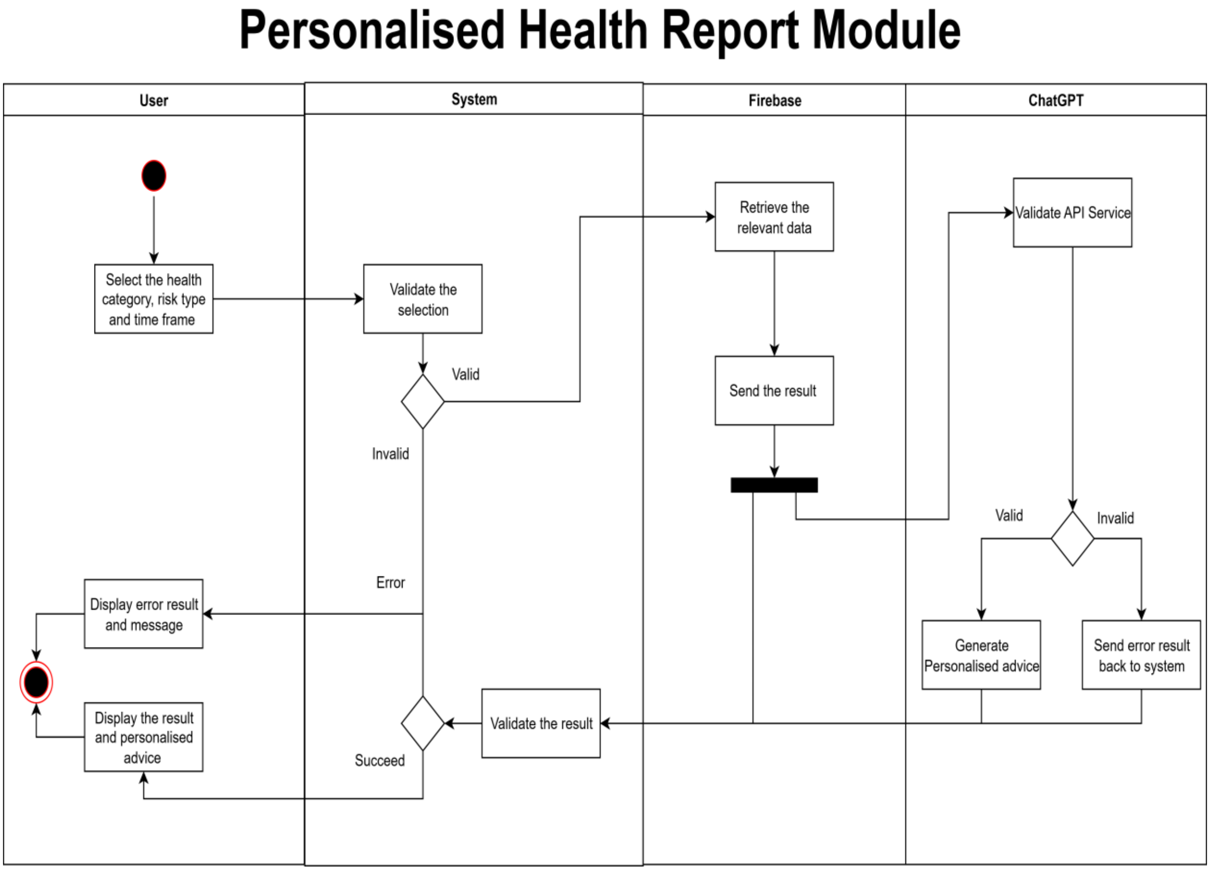


Figure 3.2.5.11 Activity Diagram (Personalised Health Report Module)

The activity diagram above shows the actors which are User, System, Firebase and ChatGPT and how they interact in the overall process, data and logic flow of the Personalised Health Report module. Each actor performs specific functions and makes decisions to ensure the module runs smoothly and efficiently. The user selects the health category, risk type and time frame then submits the selection to the system for validation. After the validation has been performed, the system requests the needed data from Firebase. Then, the Firebase returns the data to the system and also sends it to ChatGPT to generate personalised advice if the API is available. ChatGPT sends the personalised advice back to the system which the system will validates the results and displays the appropriate data, messages and personalised advice to the user.

3.3 System Model

In this section will introduces and explains the machine learning model involves for comparison and used for the Chronic Disease Risk Assessment module.

3.3.1 Logistic Regression Model

Logistic Regression is a widely used prediction model, especially in the healthcare and statistics sectors for predicting binary outcome events. It is a good candidate for comparing models and predicting chronic disease risk in this module because it estimates the probability that a user is at high risk for a condition. The results of this model are easy to interpret and are often used by healthcare professionals to support clinical decisions [47].

3.3.2 Random Forest Model

Random Forest is a type of machine learning algorithm that uses many decision trees to make a prediction. Each tree is trained on a different random part of the data and the final prediction will be based on the majority vote from all the trees. This method makes the model very accurate and reliable because it does not rely on just one decision tree opinion. It is excellent for predicting chronic disease risk because it can handle complex health data and identify the most important risk factors such as blood sugar level or BMI that contribute to the particular chronic disease [48].

3.3.3 XGBoost Model (eXtreme Gradient Boosting)

XGBoost is a fast and popular machine learning algorithm that is designed for obtaining efficiency and high-performance result. The model will build many decision trees sequentially as each new decision tree tries to fix the errors made by previous trees and all the decision trees results are combined to produce the final prediction. XGBoost often produce high accuracy and strong performance when it is well tuned. It is widely used on healthcare data such as predicting chronic disease risk because it can handle complex and unclean datasets and provides tools to reduce overfitting and deal with class imbalance [49]. Therefore, XGBoost is a good candidate to compare and selection for predicting chronic disease risk in this mobile health application.

3.3.4 Calibrated Ensemble Model

A calibrated ensemble model is a machine-learning algorithm that combines the predictions of two or more ML models and then adjusts the output so that the predicted probabilities are reliable. The combination approach helps improve overall accuracy and robustness while calibration ensures that the probability values reflect to the true confidence levels. For example, if the model predicts a 70% of people getting diabetes high risk, then it means that about 70 out of 100 similar cases should truly have the disease. Therefore, this makes calibrated ensembles suitable for healthcare sector such as predicting chronic disease risk because they not only provide a risk score but also indicate how confident the prediction is [50]. In other words, a higher calibrated probability means a higher likelihood that the predicted risk level is real. Therefore, this model is suitable to become as a comparison and candidate selection of becoming the machine learning model that be used in this mobile health application.

3.4 Dataset for Chronic Disease Risk Assessment

3.4.1 Diabetes Chronic Disease Dataset [51]

Available Dataset Link:

https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset/data

This dataset is based on the **Behavioral Risk Factor Surveillance System (BRFSS)** from the United States which collects the citizen personal health information such as lifestyle habits, eating patterns and history of chronic diseases. The dataset is considered reliable and authentic as other researchers have also used it for predicting diabetes risk [52].

3.4.2 Cardiovascular Disease (Heart Attack) Dataset [53]

Available Dataset Link:

https://data.mendeley.com/datasets/wmhctcrt5v/1

This dataset was collected and published by **Zheen Hospital in Erbil, Iraq** which containing the patient health information from **January to May 2019**. Therefore, it can be considered authentic and reliable which making it suitable for training machine learning models.

3.4.3 Obesity Chronic Disease Dataset [54]

Available Dataset Link:

https://www.kaggle.com/datasets/ruchikakumbhar/obesity-prediction

This dataset was collected from people in **Mexico, Peru and Colombia**. The information of this dataset includes the people eating habits, lifestyle habits and family history of chronic diseases. The dataset is considered reliable and authentic for training machine learning model as other researchers have also used it to predict obesity risk [55].

3.5 Gantt Chart

3.5.1 Gantt Chart (Final Year Project 1)

Name	Begin date	End date	Predecessors
Review Proposal Report	10/02/2025	10/02/2025	
Revise Literature Review	11/02/2025	19/02/2025	1
Revise Project Objectives	11/02/2025	14/02/2025	1
Identify development envrionment tools and technique	20/02/2025	26/02/2025	2,3
Setup development environment	27/02/2025	05/03/2025	4
Identify the requirements and implementation details for the application modules	17/02/2025	20/02/2025	3
Create System Architecture Design	21/02/2025	03/03/2025	6
Implement prototype first module	04/03/2025	12/03/2025	6,7
Implement prototype second module	13/03/2025	21/03/2025	6,7,8
Implement prototype third module	24/03/2025	01/04/2025	9
Implement prototype fourth module	02/04/2025	10/04/2025	10
Validate all modules prototype	11/04/2025	21/04/2025	8,9,10,11
Conduct final testing for all prototype module	22/04/2025	24/04/2025	12
Prepare Final Year Project 1 Report	25/04/2025	29/04/2025	13
Submit Draft Report to Supervisor	30/04/2025	30/04/2025	14
Submission Report to Faculty	01/05/2025	01/05/2025	15
Present and demostrate system the prototype to supervisor and moderator	05/05/2025	13/05/2025	

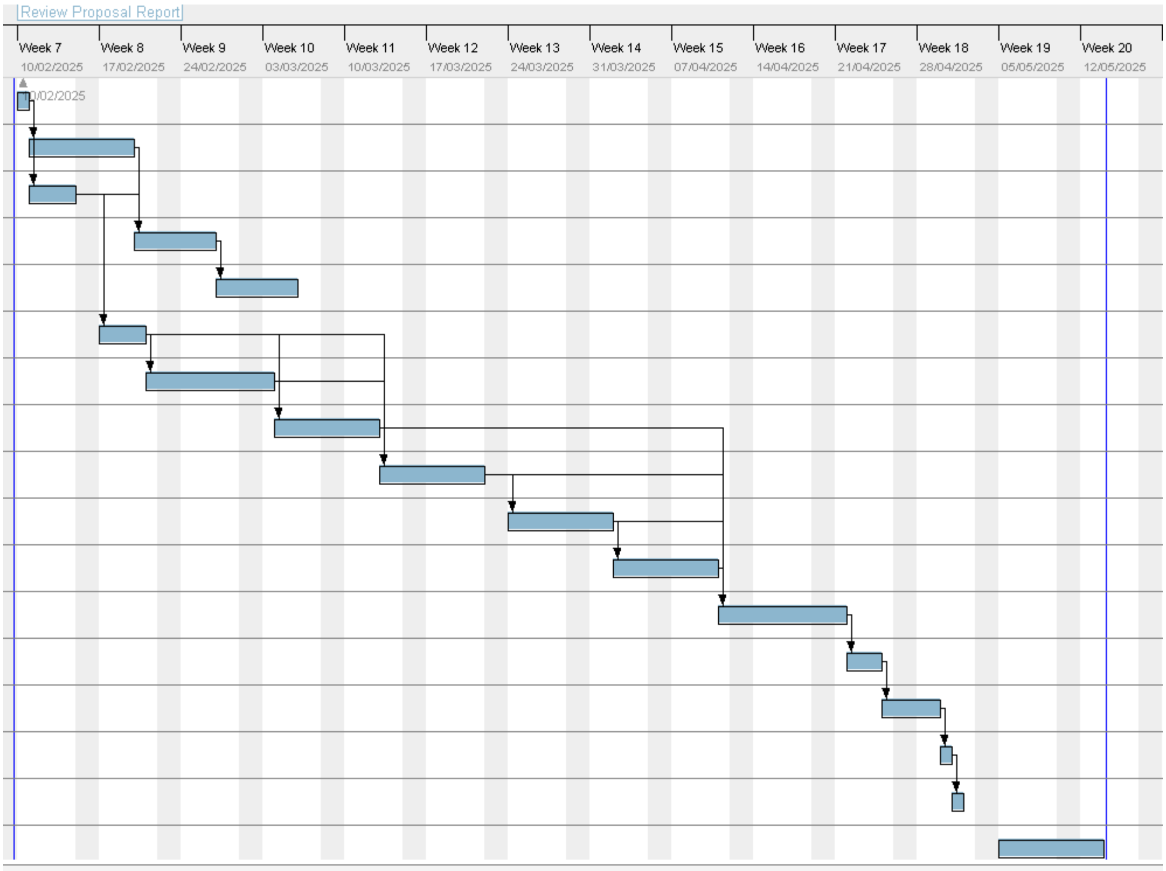


Figure 3.5.1 Gantt Chart (Final Year Project 1)

3.5.2 Gantt Chart (Final Year Project 2)

Name	Begin date	End date	Predecessors
Review Final Year Project 1	23/06/2025	23/06/2025	
Revise Project Scope and Objectives	24/06/2025	26/06/2025	1
Revise Flow Chart, Use Case and ERD Diagram	27/06/2025	27/06/2025	2
Revise Architecture and Block Diagram	30/06/2025	30/06/2025	3
Draw Activity Diagram	01/07/2025	01/07/2025	5
Draw Wireframe Application	02/07/2025	02/07/2025	6
Setup Database Environment	03/07/2025	03/07/2025	7
Setup Amazon Web Services Environment	04/07/2025	04/07/2025	8
Complete First Module	04/07/2025	23/07/2025	8
Complete Second Module	24/07/2025	06/08/2025	8,10
Complete Third Module	07/08/2025	22/08/2025	8,9,11
Complete Fourth Module	25/08/2025	02/09/2025	8,9,10,11,12
Conduct All Modules Testing	03/09/2025	09/09/2025	10,11,12,13
Writing Final Year Project 2 Report	10/09/2025	18/09/2025	15
Submission Draft Report to Supervisor	19/09/2025	19/09/2025	16
Submission Final Year Project 2 Report to Faculty	22/09/2025	22/09/2025	17
Present the Final System to Supervisor and Moderator	23/09/2025	26/09/2025	18

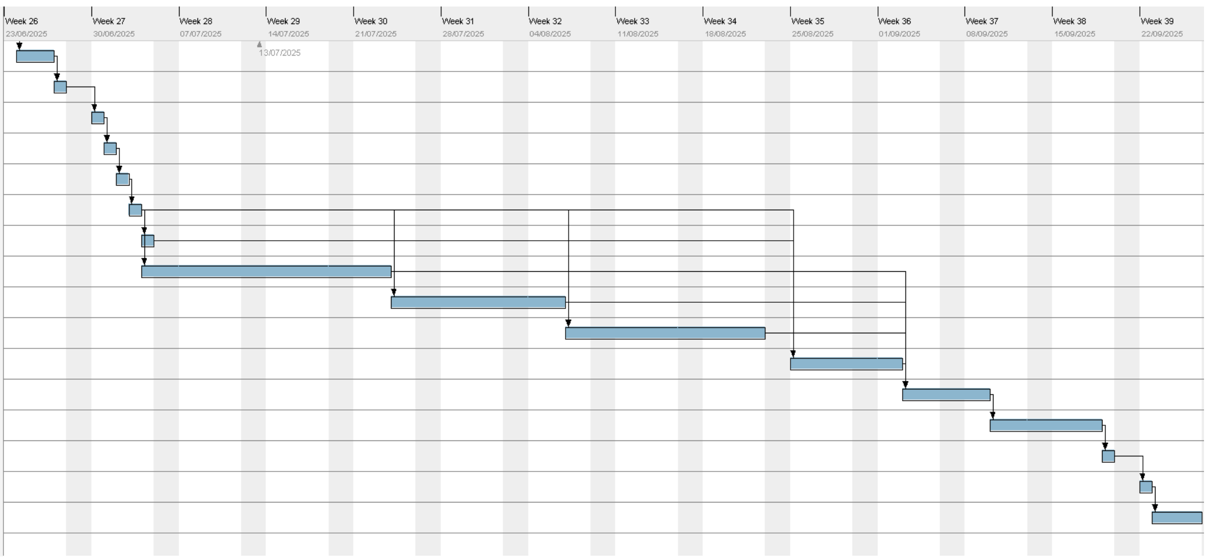


Figure 3.5.2 Gantt Chart (Final Year Project 2)

Chapter 4

System Design

4.1 System Block Diagram

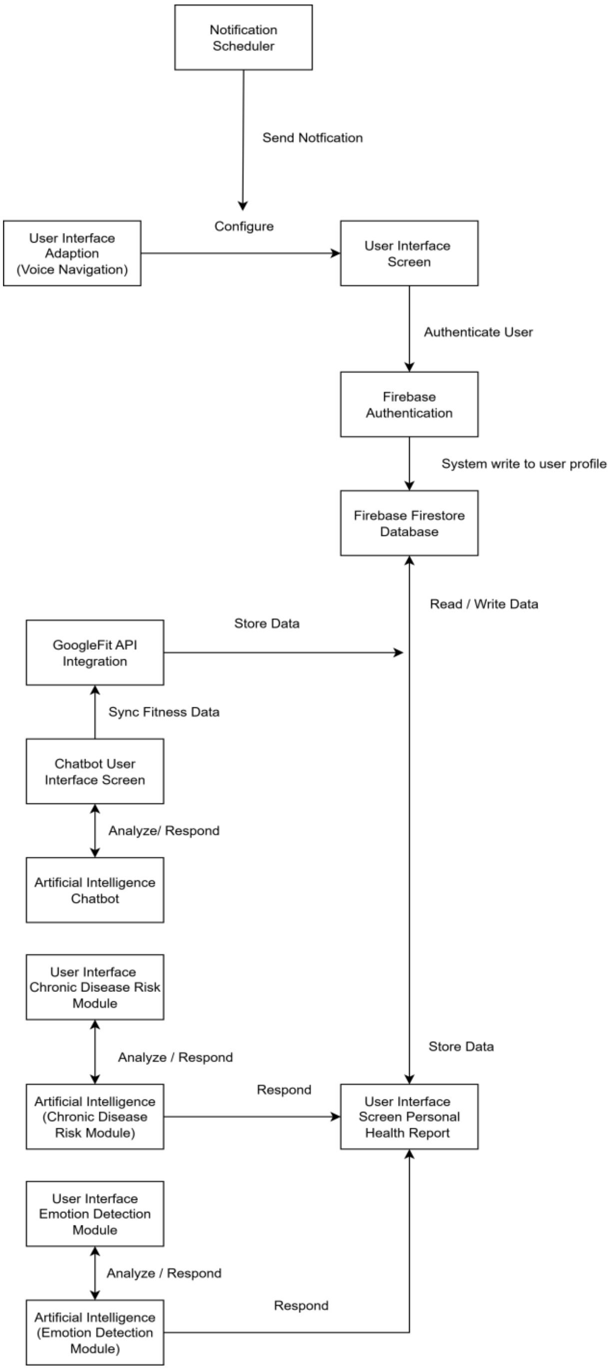


Figure 4.1.1 System Block Diagram

Based on the Block Diagram above, at the very top, the **Notification Scheduler** will configure and sends reminders or alerts to the application main interface. Before those notifications appear, the **User Interface Adaptation** module will make sure the screen is ready for voice navigation or other special controls. At the same time, these two features will prepare and configure of what the user will see and hear when they open the application.

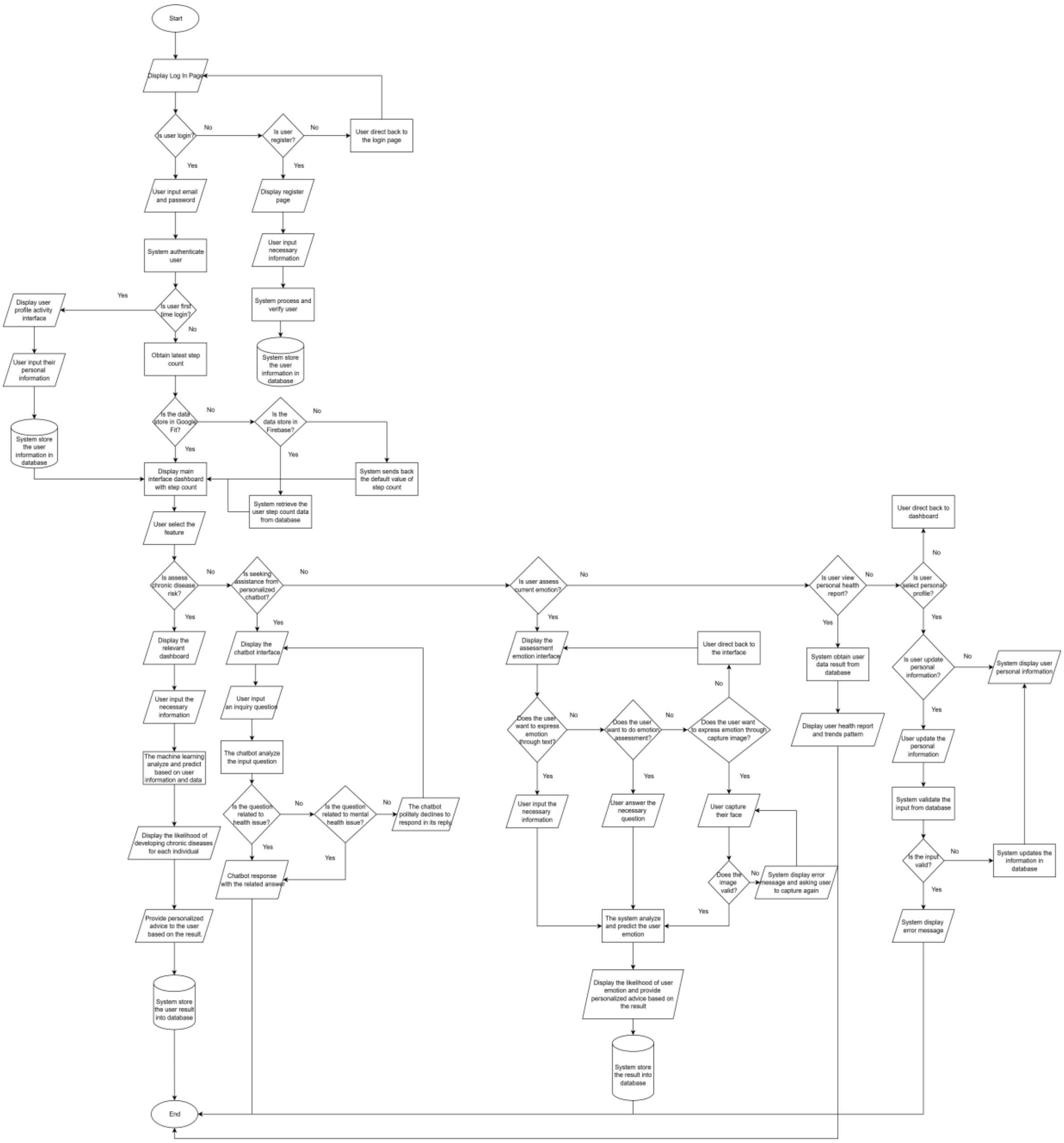
When the user launches the **User Interface Screen**, the first thing they do is log in or sign up through the **Firebase Authentication**. Once they are verified, the application records their basic profile information into **Firebase Firestore Database**. Based on the process, the application can both read existing data such as past health records and write new data such as fresh measurements to Firebase whenever it needs to. For **Google Fit API integration**, the application will connect to Google Fit and sync the user's data to **Firebase Firestore** to keep records updated. The latest step count will be shown in the user interface when available.

Next, the **Chatbot User Interface Screen** involves the **Artificial Intelligence Chatbot** to do analyse of user health data or answer the user's health questions. Any new or updated information from these conversations such as suggestion an exercise routine will also store in Firebase which is for the future use.

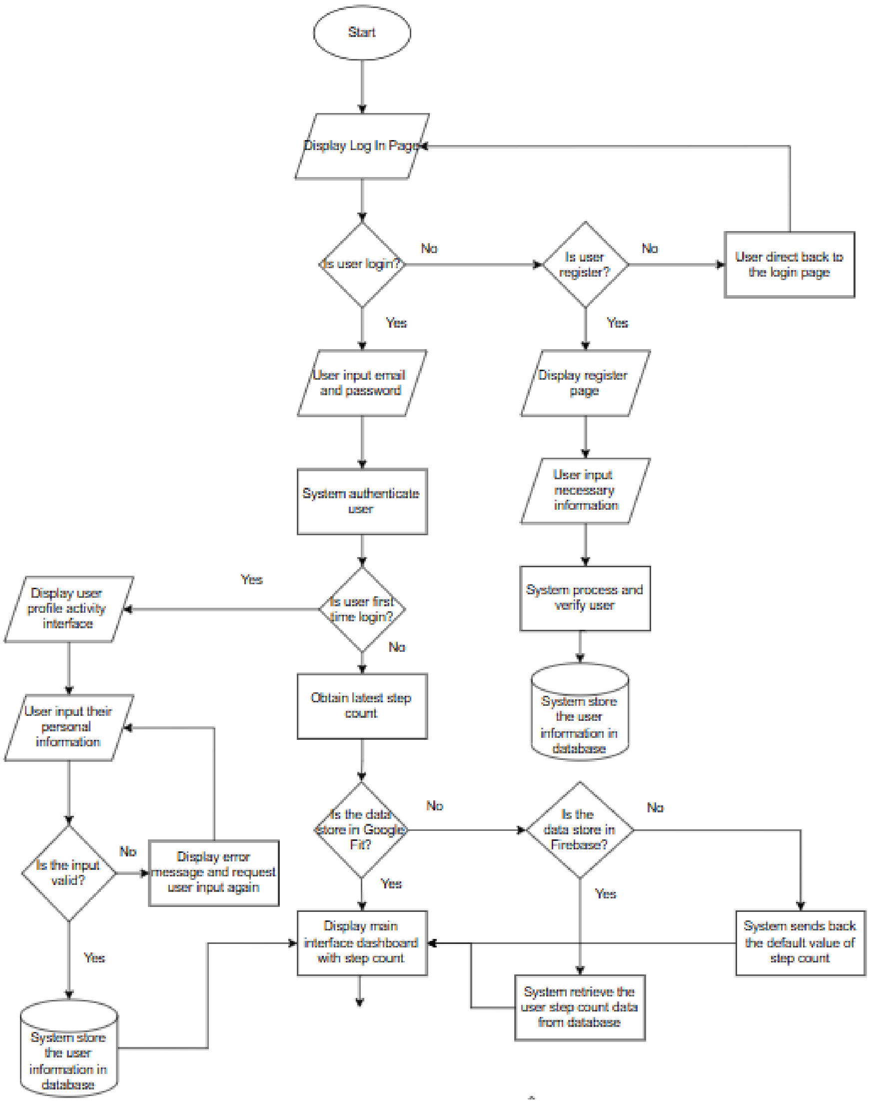
Meanwhile, another branch sends the user information into the **Chronic Disease Risk Module** where the **ML engine** will check for things like obesity or diabetes risk. The results of this analysis are sent to the User Interface Screen for Personal Health Report which giving the user a clear and color-coded summary of their chronic disease risks. This report will also store in Firebase so that the user can retrieve the data at any time.

Finally, the application includes an **Emotion Detection Module**. The user able to using the camera on their device then the AI scans the user's face to guess their current mood. The emotional feedback will also go into the **Personal Health Report** screen which letting the user see not only their physical health metrics but also how they are feeling today. Like the other parts, emotion results are also stored in Firebase and can be used by the chatbot for more provide better personalized advice to the user.

4.2 Flow Chart Diagram

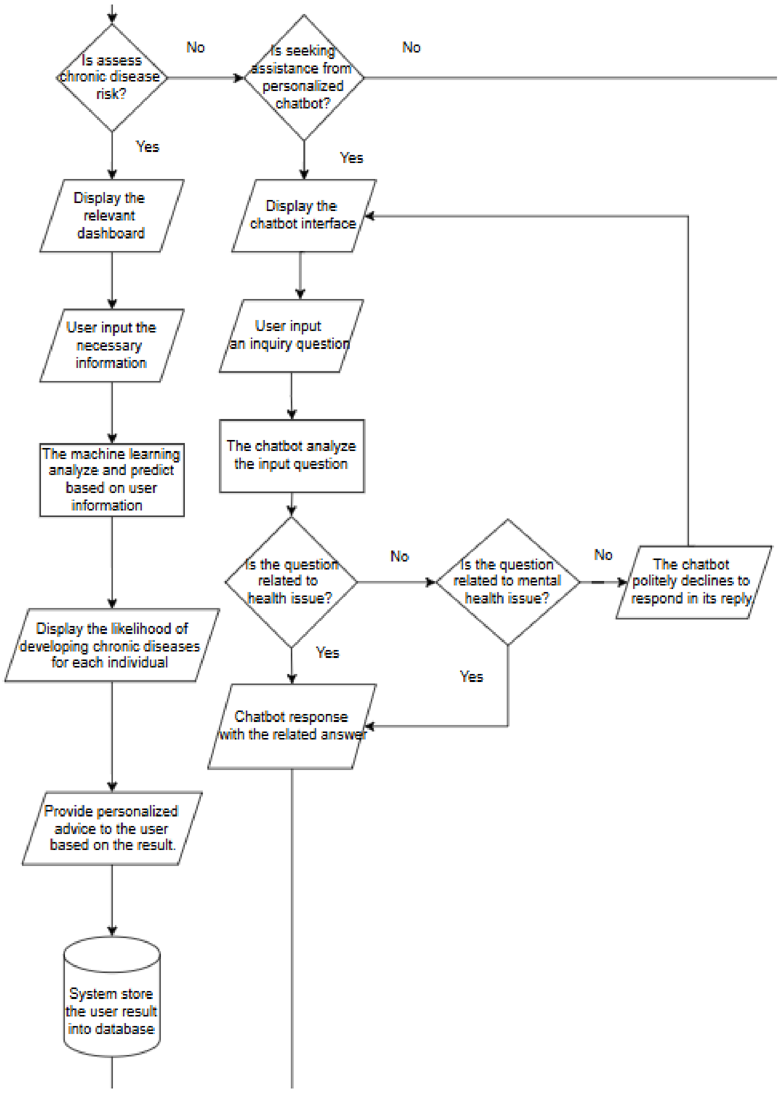


4.2.1.1 Overall Flow Chart Diagram



4.2.1.2 Flowchart Diagram (Sign Up and Login Module)

Based on the diagram above shows the overall flowchart diagram of the mobile health application. The flowchart starts at the login screen where the user must choose to either sign in or register an account. If they select login, they are required to enter their email and password and the system will immediately verify the input by compare the data that stored in the firebase. If verification succeeds, the system checks whether this is the user's first login. First-time users who sign in with Google must complete their personal information before they can access to the application. The system will validate the user input and if it is valid, the user information will be stored into firebase and then they will direct to the home dashboard, otherwise a failure message returns them to the personal activity interface to request the user input again. If they choose to register, they need to provide the required details on the registration page. Once completed, the system creates a new user record and verifies the user's step count by retrieving data from Google Fit or from Firebase, then shows it in the main dashboard.



4.2.1.3 Flowchart Diagram (Chronic Disease Risk and Chatbot Module)

In the main dashboard, the user can select one of the modules which are assessing chronic-disease risk, chatting with the personalized health assistant, do assessment to verify their current emotional state, viewing their overall health report and trends and also updating their profile information. Each selection has its own mini–flow process to be complete. Based on this diagram, the risk assessment module will retrieve the user's information, runs it through a developed machine-learning model to make analysis and predict the risk of getting chronic disease, displays percentage-based risk scores for heart attack, diabetes and obesity, offers tailored advice and finally saves the results back to the database for enable user to view their health trend.

Next, if the user chooses for the chatbot, they can enter any health-related or mental health question and the chatbot will analyses whether the question is related to health-related issues or not. If the question is valid, the chatbot will answer the question that based on the user and provide the advice to the user if it is negative question about health-related issues or mental health. The chatbot will also politely to decline the user queries if the question is not related.

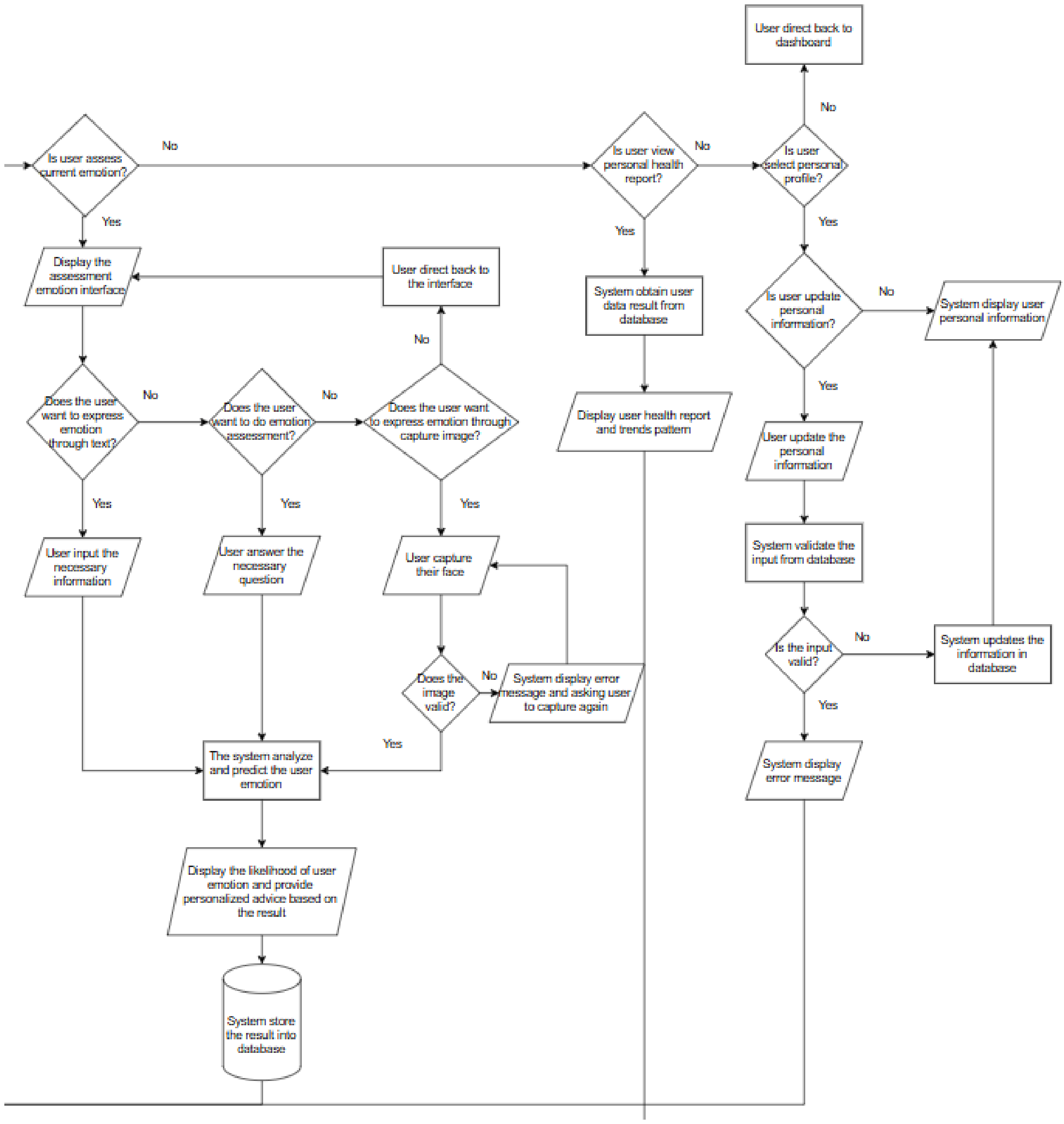


Figure 4.2.1.4 Flowchart Diagram

(Mental Health, Personalised Health Report and Update Profile Module)

For mental health module, the user can choose to briefly describe their current feeling and answer a various of assessment question that follows PHQ method or capture a photo which the system then applies facial-analysis models to analyze and detect the user current mood and displays a probability breakdown of emotions. Once the result has been analyzing regarding user to choose what method to detect their emotion, the system will offer personalised advice to the user at the same time it also stores the result into the firebase. If they choose to viewing their personal health report, the app fetches all past risk and emotion assessments data result and generate them into visually which is trend lines and also allows the user to return to the dashboard at any time. In the personal profile module, users can update their personal information by entering the latest details. The application validates the input and updates the data in Firebase if the input is valid. Otherwise, an error message will be display and asks the user try to input again.

Throughout every module in this mobile health application, the system assures about data integrity and also able to have a meaningful error handling. Invalid inputs such as incorrect login credentials, duplicate profile entries, invalid chatbot questions or invalid face captured image will trigger a clear error message and guide the user back to the appropriate screen for correction. In result, the application able to delivers accurate personalized health insights while at the same time also maintaining a smooth and engaging user experience.

4.3 Entity Relationship Diagram (ERD Diagram - NO SQL)

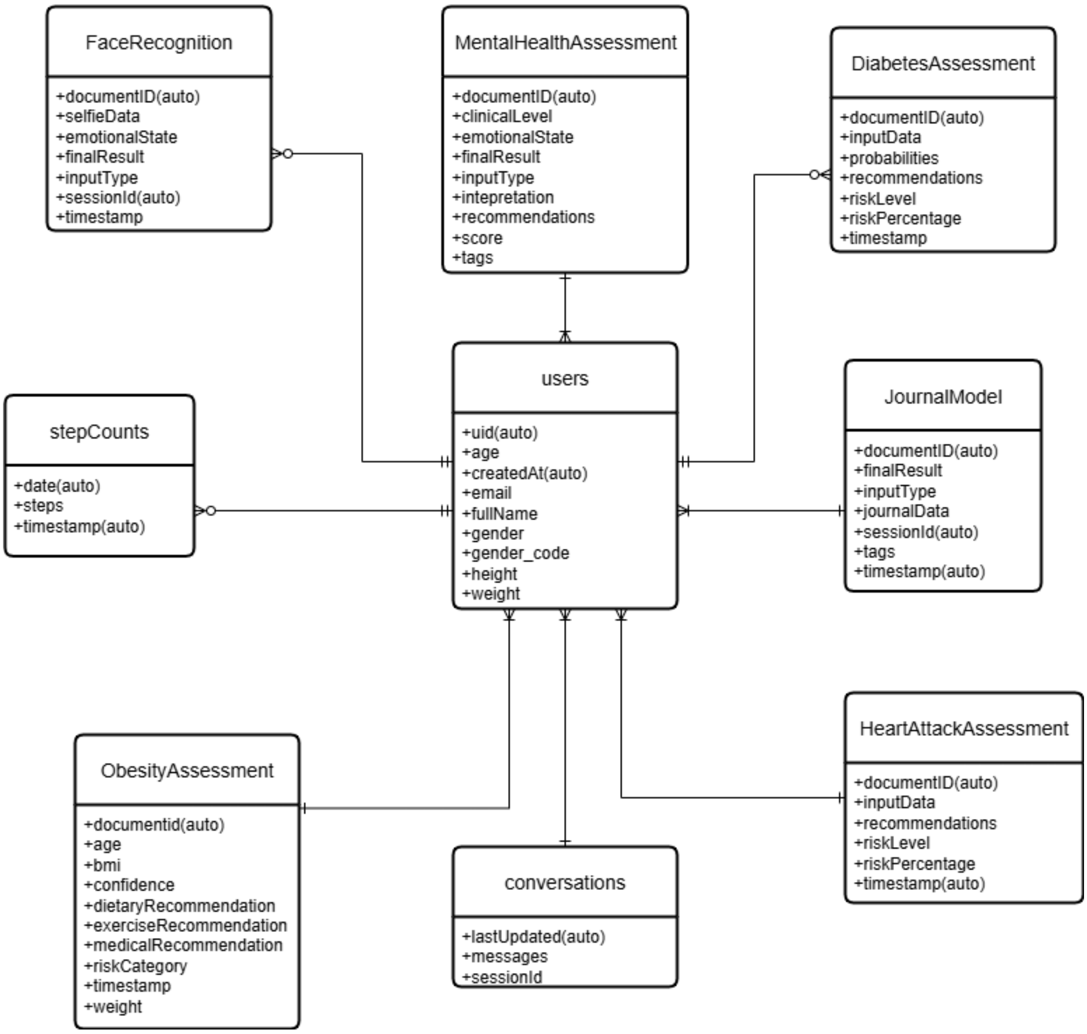


Figure 4.3.1 Entity Relationship Diagram (NO SQL)

The ERD diagram above shows the all-data tables with the related attributes and how they connect each other in the Mobile Health Application. Based on the diagram, the centre of the ERD is the **users** table which it holds each person's profile information such as their full name, email, gender and also connected to all other tables which required the users table to display the relevant information to the user in the application.

The left-hand table, **stepCounts**, stores each user's total steps, the date of the steps was recorded and a timestamp of the most recent update. Next, the **FaceRecognition** table auto-generates a document_id to ensure uniqueness of personal data for each user. It stores selfie_data to record input validation, the user's detected emotional_state using AWS Rekognition, the final_result produced by the analysis, the input_type such as phone camera, a session_id and a timestamp that is automatically generated from the capture date and time. Furthermore, the **ObesityAssessment** table record the user obesity risk state. It stores the user's age, weight, BMI, the ML model's confidence value, dietary, exercise, and medical_recommendations generated by AI from ChatGPT, the risk_category, and the assessment timestamp and also auto-generates a document_id.

The **MentalHealthAssessment** table stores the results of the user's depression evaluation. It records the clinical severity level such as Mild, Moderate which derived from the PHQ-9 score and contrasts it with the more user-friendly emotional state such as Calm/Content and Stressed. It also includes the calculated PHQ-9 score itself, the input type will mark as 'assessment', an AI-generated interpretation and a list of personalized recommendations from ChatGPT. Each record is also tagged with a category for module identification.

Below the users table, the **conversations** table stores the chat history between users and the chatbot. It contains the message log, a last-updated timestamp, and an auto-generated session_id to guarantee the uniqueness of each conversation.

To the right of the users table, the **DiabetesAssessment** table stores the results of a user's diabetes risk evaluation. It contains the user's inputData from the assessment, the ML model analysis which including the Probability, riskPercenatage and riskLevel, the AI-generated recommendations from ChatGPT and a timestamp of when the assessment was taken.

Below this, the **JournalModel** table stores the user's journal or feeling entries. It contains the journal input data, a sentiment analysis result generated by ChatGPT and a timestamp for the entry. To ensure data integrity, it uses an auto-generated document_id as a unique key and a session_id to group related entries. The inputType is marked as 'journal' and entries are tagged for categorization, such as 'how are you feeling today'.

Further down, the **HeartAttackAssessmen**t table stores the results from the cardiovascular disease (CVD) risk module, which predicts the user's likelihood of a heart attack. It contains the user's input data from the assessment, the ML model's output which including the riskLevel and riskPercentage, AI-generated recommendations from ChatGPT and a timestamp of the assessment. A unique document_id is auto-generated for each record to ensure data integrity.

In conclusion, each of these tables links back to the user via the shared userId which is the primary key of the users table by ensuring that all the tables' data is related to the correct user. This design shows the data is easy to fetch and update and supports features like personalized advice of risk warnings and emotion-based advice and keeps the application data consistent and reliable.

4.4 System Components Specifications

In this section, it will outline several important software tools and services were used to build for this mobile health application. Each one was chosen for its specific strengths as to make the application work effectively.

4.4.1 Android Studio (Android Environment)

Android Studio is the official IDE for building Android apps. It includes useful tools like emulators, debuggers, and code editors. It is great for creating mobile health application because it connects easily with the Android SDK, helping to build feature-rich and high-performance applications. Its built-in profiling tools also help to improve app performance, which is important for real-time health tracking. The drag-and-drop UI designer also makes it simple to create user-friendly layouts.

4.4.2 Firebase (Database Environment)

Firebase is a real-time database that provides features like live data syncing and secure user authentication. It also includes many useful tools for app development. Firebase's cloud features allow for serverless actions such as creating reports or sending notification reminders to help users meet their daily health goals. It is suitable to become the database selection for this application because it can easily scale to support more users as the app grows and works across different platforms.

4.4.3 TensorFlowLite (Machine Learning Converter)

TensorFlow Lite is a lightweight tool that allows machine learning models to run directly on mobile devices. It is used in this application to provide fast predictions for health risks such as diabetes or cardiovascular disease without needing an internet connection. This means users can get immediate results while keeping their data private and secure on their own device. Therefore, TensorFlowLite is chosen for this application because it helps the application work quickly, save battery life and protect user privacy by processing everything locally in the application.

4.4.4 ONNX Model (Machine Learning Converter)

ONNX (Open Neural Network Exchange) is a format that helps make machine learning models compatible across different platforms. It is used in this application to convert models created in other frameworks like PyTorch or scikit-learn, into a format that can work efficiently on mobile devices. This gives more flexibility to choose the best model for each health prediction task. Therefore, ONNX was included in this application because it allows the application to use a wider variety of models, making the predictions more accurate and the overall system more adaptable.

4.4.5 Google Fit (Fitness Data Service)

The Google Fit API allows the app to access a user's health data such as step count that stored in their Google Fit account. By using the official Google Fit SDK, the app can securely read this data once the user gives permission. This saves development time and ensures the application works well on many Android devices. Therefore, Google Fit is chosen for this mobile health application because it provides a reliable and standardized way to access high quality health data without needing to build complex tracking features from scratch.

4.4.6 ChatGPT (AI Recommendations and Chatbot)

ChatGPT is used in this application to provide personalized health advice and to serves as a healthcare chatbot to the user. After a user completes an assessment, their results are sent to the ChatGPT API, which returns easy to understand explanations and useful recommendations. The chatbot also allows users to ask health-related questions and it can remember the conversation history to make the discussion more natural and continuous. Therefore, ChatGPT is chosen for this mobile health application because it helps the application offer more engaging and supportive interactions, making users feel like they are talking to a helpful assistant rather than just using a regular app.

4.4.7 AWS Face Rekognition (Emotion Detection)

AWS Face Rekognition is a service used to detect emotions from a user's selfie. The application sends the photo to the AWS API, which analyzes it and returns the detected emotional state like happy, sad, calm, or angry along with how confident it is in each result. The service is suitable for this mobile health application because it gives the application another way to check on the user's well-being without making them answer more questions. It provides an objective look at their current mood, which can be combined with their other assessment results to give a more complete picture of how they are feeling. This makes the overall health analysis richer and helps offer better and more personalized support.

Chapter 5
System Implementation

5.1 Hardware Setup

The hardware devices used in this project will include a laptop and an Android mobile device. The laptop will serve as the primary development platform which used for coding the application, testing and debugging. Next, the laptop will involve installing fundamental programming tools such as Android Studio, as well as implement each module in this application. The requirements for this Android app development project include a laptop with a current processor, 16 GB of RAM, a GeForce RTX 3050 GPU, and sufficient storage space.

Table 5.1.1 Specifications of laptop

Description	Specifications
Model	Acer Nitro 5 AN515-58
Processor	Intel Core i5-12500H
Operating System	Windows 11 Home
Graphic	NVIDIA GeForce RTX 3050 GPU
Memory	16 GB DDR4 RAM
Storage	512GB NVMe SSD

An Android smartphone will be used to test and validate all the modules in this mobile health application which ensuring that compatibility, usability and performance when it deploys on the real world. By testing on the Android mobile device, it will verify that the application meets the requirement of the application and to shows the output of each module.

Table 5.1.2 Specifications of Mobile Device

Description	**Specifications**
Model	Huawei Y9s Series
Processor	Kirin 710F (Octa-Core CPU)
Operating System	Android 9 (Pie) with EMUI 9.1
Graphic	Mali-G51 MP4 GPU
Memory	6GB RAM
Storage	128 GB Internal Storage

5.2 Software Setup

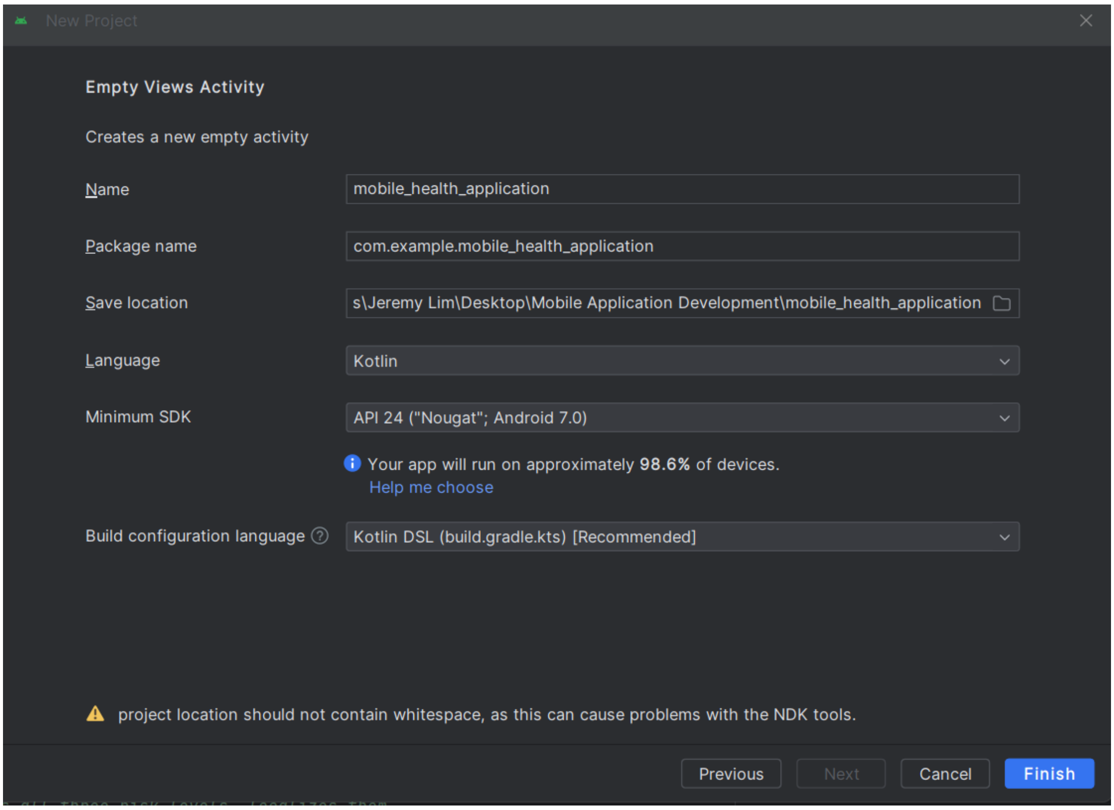


Figure 5.2.1 Android Studio Setup

Before developing the application, we need to configured the project to use Kotlin as the programming language and set the minimum target SDK to API 24 (Android 7.0 or higher) and used Kotlin DSL (build.gradle.kts) for the build configuration.

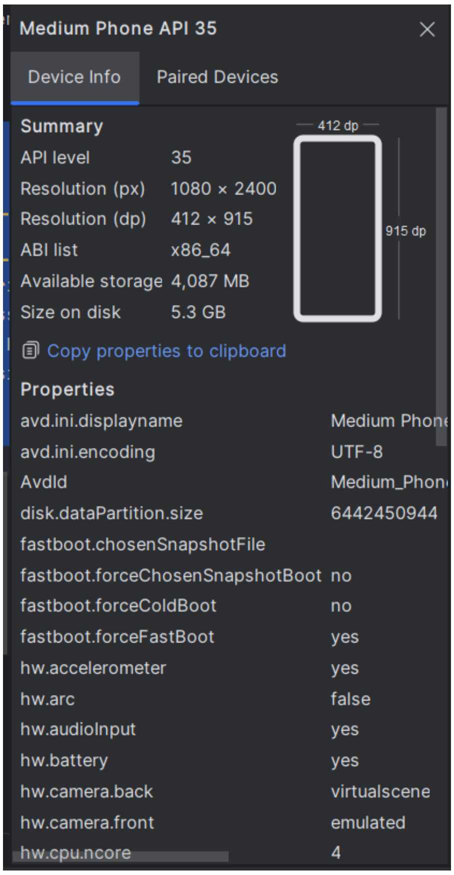


Figure 5.2.2 Android Studio Configuration Emulator Device

For the emulator device, we need to choose API 35 level to run the application as it able to test the result of each module with showing the output in the screen.

5.3 Setting and Configuration

5.3.1 Android Studio

build.gradle (mobile_health_application)

You can use the Project Structure dialog to view and edit your project configuration Open (Ctrl

```
// Top-level build file where you can add configuration options common to all sub-projects/modules.
plugins {
    id 'com.android.application' version '8.2.2' apply false
    id 'org.jetbrains.kotlin.android' version '1.9.0' apply false
    id 'com.google.android.libraries.mapsplatform.secrets-gradle-plugin' version '2.0.1' apply false
    //For Firebase Connection
    id("com.google.gms.google-services") version "4.4.1" apply false
}
```

Figure 5.3.1.1 Screenshot build.gradle(Project Level)

In Android Studio, we need to add the Firebase plugin to allow our application to connect with the Firebase Firestore Database.

build.gradle (:app)

You can use the Project Structure dialog to view and edit your project configuration

```
plugins {
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android' version '1.9.0'
    id 'com.google.android.libraries.mapsplatform.secrets-gradle-plugin'
    id("kotlin-kapt") //For Face Recognition Feature
    id("com.google.gms.google-services") // For Firebase Connection
    id("kotlin-parcelize")// For Mental Health Model Result Activity

}
```

```
dependencies {
    //GoogleFit API and Authorization
    implementation 'com.google.android.gms:play-services-fitness:21.1.0'
    implementation 'com.google.android.gms:play-services-auth:20.7.0'

    //For test data step count and sleep data pattern
    implementation 'com.google.android.gms:play-services-fitness:21.0.1'

    // For schedule periodic (notifiations)
    implementation "androidx.work:work-runtime-ktx:2.7.1"

    //For getting the latest icon
    implementation "androidx.compose.material:material-icons-extended:<latest-version>"

    //For navigation to other module
    implementation("androidx.navigation:navigation-compose:2.7.6")


    //For GSON Implementation
    implementation("com.google.code.gson:gson:2.8.9")
    // WorkManager core + KTX
    //implementation "androidx.work:work-runtime-ktx:2.9.0"

    //For implement of extend Material Icon
    implementation "androidx.compose.material3:material3:1.0.0"
    implementation 'com.google.android.material:material:1.11.0'

```

```
    // For additional icons
    implementation "androidx.compose.material:material-icons-extended:1.4.3"

    // Retrofit for networking
    implementation 'com.squareup.retrofit2:retrofit:2.9.0'
    // Gson converter
    implementation 'com.squareup.retrofit2:converter-gson:2.9.0'
    // Coroutines
    implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-android:1.7.1'
    //Interceptor
    implementation 'com.squareup.okhttp3:logging-interceptor:4.9.3'


    implementation 'androidx.core:core-ktx:1.12.0'
    implementation 'androidx.lifecycle:lifecycle-runtime-ktx:2.6.2'
    implementation 'androidx.activity:activity-compose:1.8.0'


    implementation platform('androidx.compose:compose-bom:2023.08.00')
    implementation 'androidx.compose.ui:ui'
    implementation 'androidx.compose.ui:ui-graphics'
    implementation 'androidx.compose.ui:ui-tooling-preview'
    implementation 'androidx.compose.material3:material3'
    implementation 'com.google.android.gms:play-services-maps:19.2.0'
    implementation 'androidx.appcompat:appcompat:1.7.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.2.1'
    implementation 'com.google.android.material:material:1.12.0'
```

```
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
    androidTestImplementation platform('androidx.compose:compose-bom:2023.08.00')
    androidTestImplementation 'androidx.compose.ui:ui-test-junit4'
    debugImplementation 'androidx.compose.ui:ui-tooling'
    debugImplementation 'androidx.compose.ui:ui-test-manifest'

    // AndroidX basics
    implementation("androidx.core:core-ktx:1.13.1")
    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.12.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")

    // Lifecycle + coroutines
    implementation("androidx.lifecycle:lifecycle-runtime-ktx:2.7.0")
    implementation("androidx.lifecycle:lifecycle-viewmodel-ktx:2.7.0")
    implementation("org.jetbrains.kotlinx:kotlinx-coroutines-android:1.7.3")
    implementation("androidx.lifecycle:lifecycle-livedata-ktx:2.6.1")
    implementation("androidx.fragment:fragment-ktx:1.6.1")

    // Retrofit + OkHttp + Moshi for ChatGPT API
    implementation("com.squareup.retrofit2:retrofit:2.9.0")
    implementation("com.squareup.retrofit2:converter-moshi:2.9.0")
    implementation("com.squareup.okhttp3:okhttp:4.11.0" )
    implementation("com.squareup.okhttp3:logging-interceptor:4.11.0" )
    implementation("com.squareup.moshi:moshi:1.15.0")
```

```
    implementation("com.squareup.moshi:moshi-kotlin:1.15.0")
    kapt("com.squareup.moshi:moshi-kotlin-codegen:1.15.0")

    // CameraX
    implementation("androidx.camera:camera-core:1.3.3")
    implementation("androidx.camera:camera-camera2:1.3.3")
    implementation("androidx.camera:camera-lifecycle:1.3.3")
    implementation("androidx.camera:camera-view:1.3.3")
    implementation("androidx.camera:camera-extensions:1.3.3")

    // Testing dependencies (keep existing if any, or add these)
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")

    //For ONNX Runtime dependency
    implementation("com.microsoft.onnxruntime:onnxruntime-android:1.17.1")

    // TensorFlow Lite dependencies
    implementation("org.tensorflow:tensorflow-lite:2.10.0")
    implementation("org.tensorflow:tensorflow-lite-support:0.4.0")
    implementation("org.tensorflow:tensorflow-lite-metadata:0.4.0")

    //For Firebase Connection
    // Import the Firebase BoM
    implementation(platform("com.google.firebase:firebase-bom:32.8.1"))
```

```
    // TODO: Add the dependencies for Firebase products you want to use
    // When using the BoM, don't specify versions in Firebase dependencies
    implementation("com.google.firebase:firebase-analytics")

    //Firebase authentication
    //implementation 'com.google.firebase:firebase-auth-ktx:24.0.1'
    implementation("com.google.firebase:firebase-auth-ktx")

    //Firebase Firestore
    implementation 'com.google.firebase:firebase-firestore-ktx:24.9.1'

    //For Mental Health Data Class FinalResult.kt
    implementation("org.jetbrains.kotlin:kotlin-stdlib:1.8.0")

    // Chart library for visualization
   // implementation("com.github.PhilJay:MPAndroidChart:v3.0.3")
    implementation("com.diogobernardino:williamchart:3.11.0")

    // For date formatting
    implementation("org.jetbrains.kotlinx:kotlinx-datetime:0.4.0")

    //Firebase Cloud Messaging (Notification)
    implementation 'com.google.firebase:firebase-messaging:23.2.1'
    implementation 'com.google.firebase:firebase-installations:17.1.3'
}
```

Figure 5.3.1.2 Screenshot build.gradle(Application Level)

In the app-level build.gradle file, we need to add the required highlighted plugins and dependencies in our mobile health application.

```
assets
    diabetes_model_final.onnx
    heart_risk_model.tflite
    obesity_model.onnx
    preprocessor_info.json
    scaler_params.json
```

Figure 5.3.1.3 Screenshot assets folder

We add the machine learning models and associated JSON files to the assets folder so the application can access them.

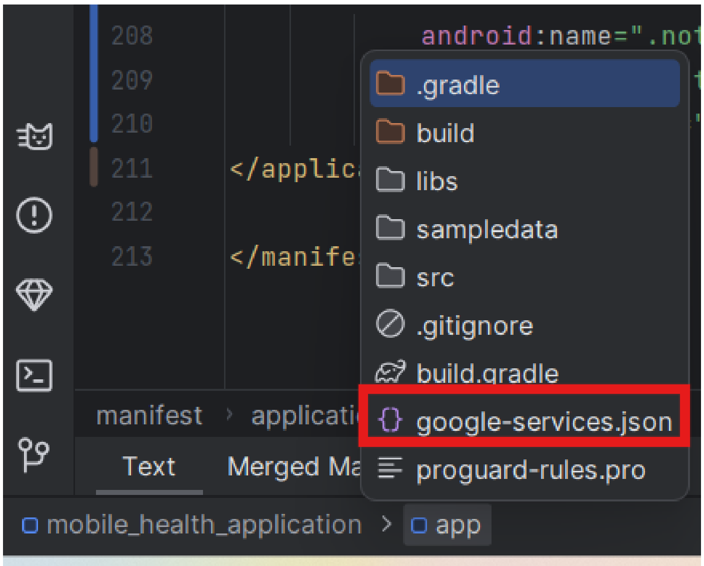


Figure 5.3.1.4 Screenshot app folder

The google-services.json file must be added to the app folder to enable Firebase connectivity. This file is generated in the Firebase console once the application is added to the Firebase project.

```
<uses-feature
    android:name="android.hardware.camera"
    android:required="false" />
<uses-feature android:name="android.hardware.camera.autofocus" />

<uses-permission android:name="android.permission.INTERNET" /> <!-- For new version androi
<uses-permission android:name="android.permission.ACTIVITY_RECOGNITION" /> <!-- For older
<uses-permission android:name="com.google.android.gms.permission.ACTIVITY_RECOGNITION" />
<uses-permission android:name="android.permission.POST_NOTIFICATIONS" /><!-- For notificat
<uses-permission android:name="android.permission.RECORD_AUDIO"/><!-- For record voice inp
<uses-permission android:name="android.permission.CAMERA" /> <!-- For camera uses -->
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" /> <!-- For access
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" /><!-- For detect
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" /><!-- For detec
<uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCATION" /><!-- For d
<!-- For Health Connect -->
<uses-permission android:name="android.permission.health.READ_STEPS" />
<uses-permission android:name="android.permission.health.READ_SLEEP" />
<!-- Use Phone Hardware to detect step count -->
<uses-feature android:name="android.hardware.sensor.stepcounter" />
<uses-feature android:name="android.hardware.sensor.stepdetector" />
<!-- For Sending Notification to user -->
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.SCHEDULE_EXACT_ALARM" />
<uses-permission android:name="android.permission.USE_EXACT_ALARM" />
```

Figure 5.3.1.5 Screenshot AndroidManifest.xml

These are the user permissions we need to add to our AndroidManifest.xml file and request from user to make sure the modules and services in our application able to work effectively.

5.3.2 Firebase Database Setup

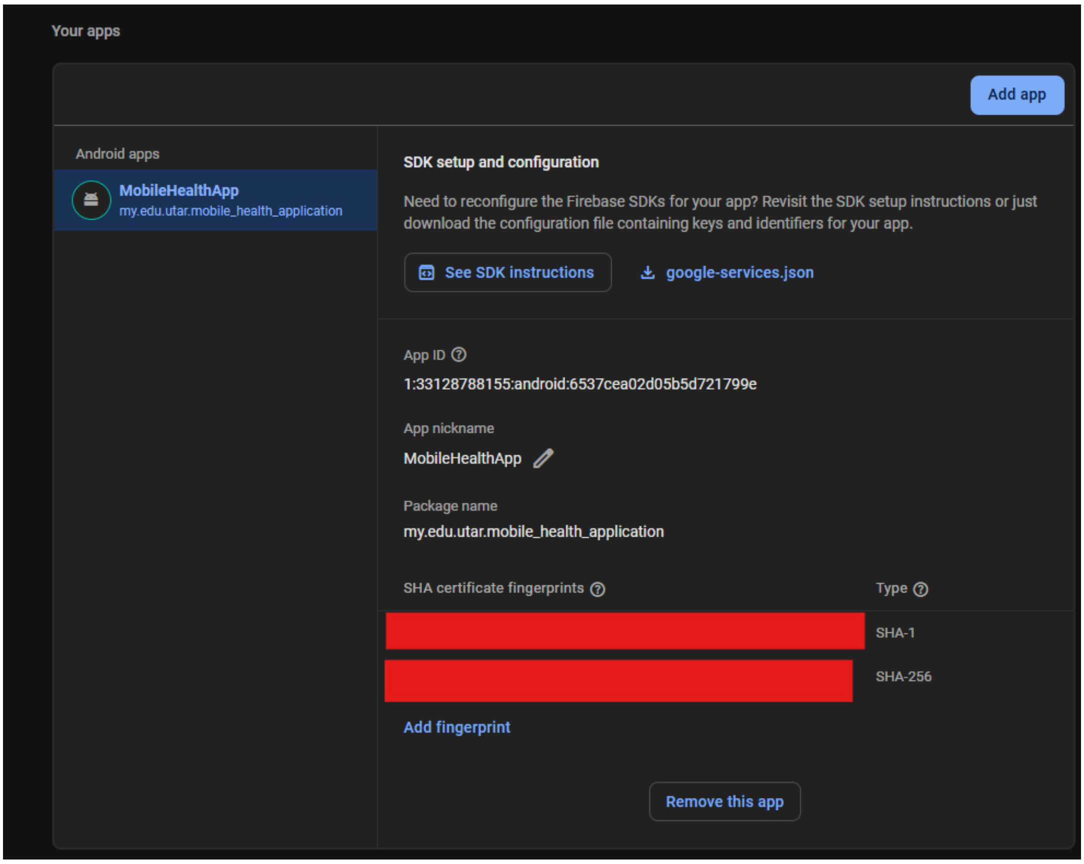


Figure 5.3.2.1 Screenshot Firebase Console add application

Based on the figure, we need to add our mobile application to the Firebase Console within our Firebase project. By clicking the 'Add App' button, we can follow the provided steps to register the application. This process requires providing the app's nickname, package name and SHA-1 along with SHA-256 signing certificate fingerprints. Due to privacy concerns, the figure has hidden the SHA-1 and SHA-256 values.

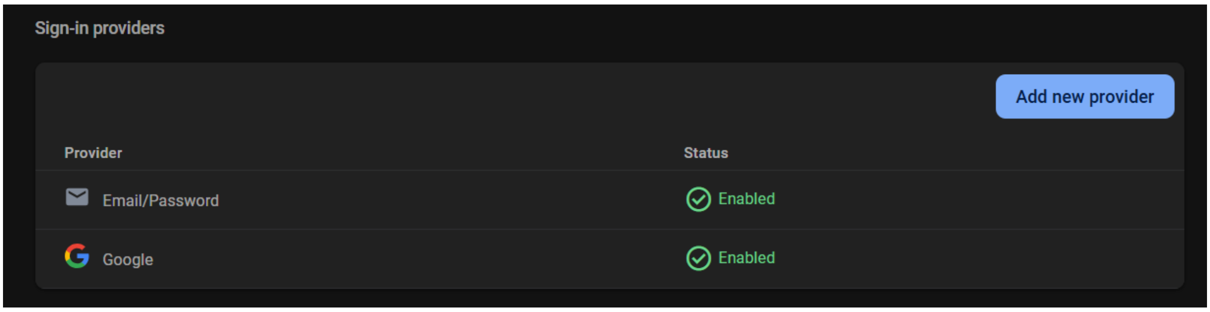


Figure 5.3.2.2 Screenshot Firebase Console sign-in providers

Based on the figure, we need to add these two sign-in providers in our Firebase project by clicking the 'Add new provider' button. This allows users to log in to the app using either their personal email address that authenticated by Firebase or Google Sign-In.

```
rules_version = '2';

service cloud.firestore {
  match /databases/{database}/documents {
    // Allow read/write access to all users under any conditions

    match /{document=**} {
      allow read, write: if true;
    }
  }
}
```

Figure 5.3.2.3 Screenshot Firebase Firestore Rules

In our Firebase Firestore Database, we need to configure the security rules as shown in the figure above to allow users to read and write data in our application.

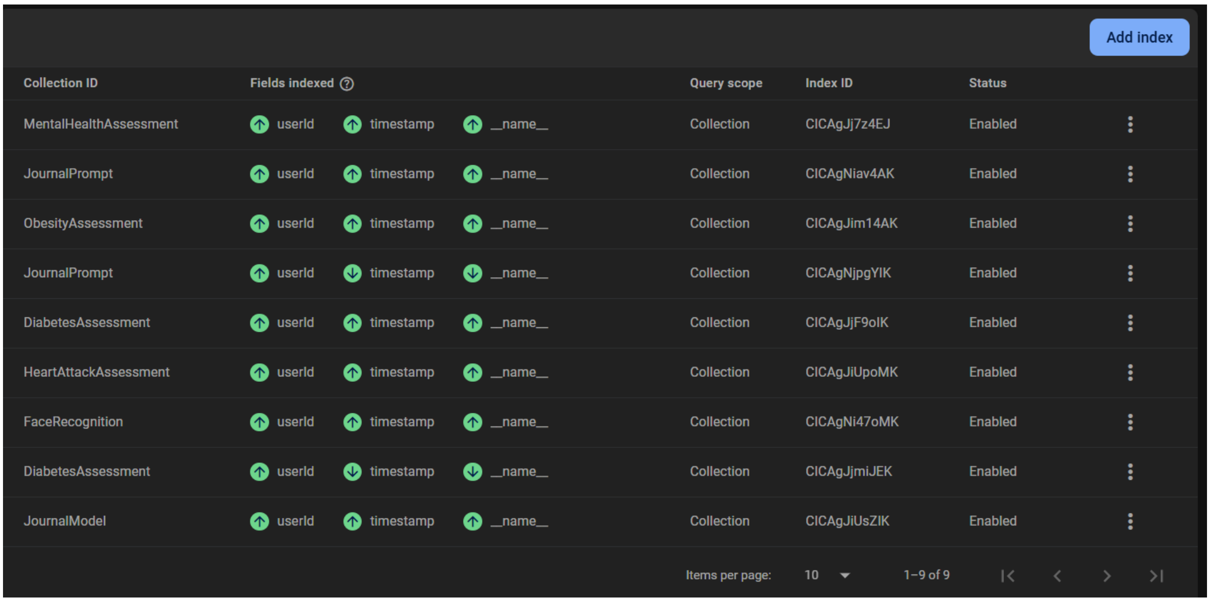


Figure 5.3.2.4 Screenshot Firebase Firestore Indexes

In Firebase Firestore, we need to create all the indexes shown in the figure. To add an index, click the 'Create Index' button and follow the steps provided. The required information includes the collection ID and the fields to be indexed which must match the attributes defined in the Firestore database entities. This is to allow user able to retrieve the correct data from the entity and display data in visualization format in the application.

5.3.3 AWS Face Rekognition Setup

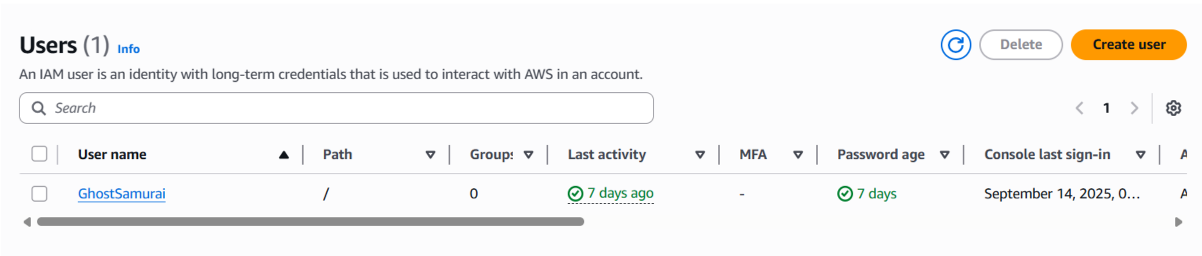


Figure 5.3.3.1 Screenshot IAM User

In the AWS Management Console, we need to create a user with permissions to access the AWS Face Rekognition service for our application. To create the user, click the 'Create user' button and follow the guided steps to set up the account.

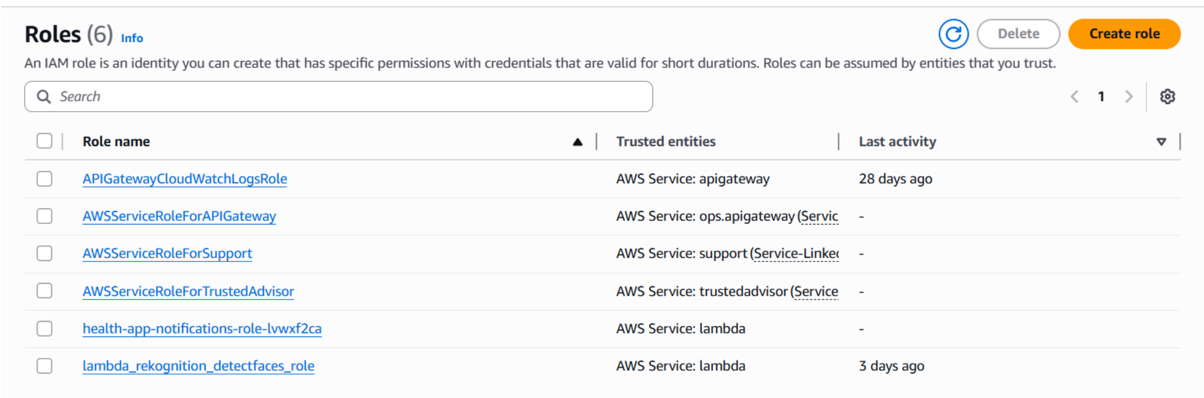


Figure 5.3.3.2 Screenshot User Roles

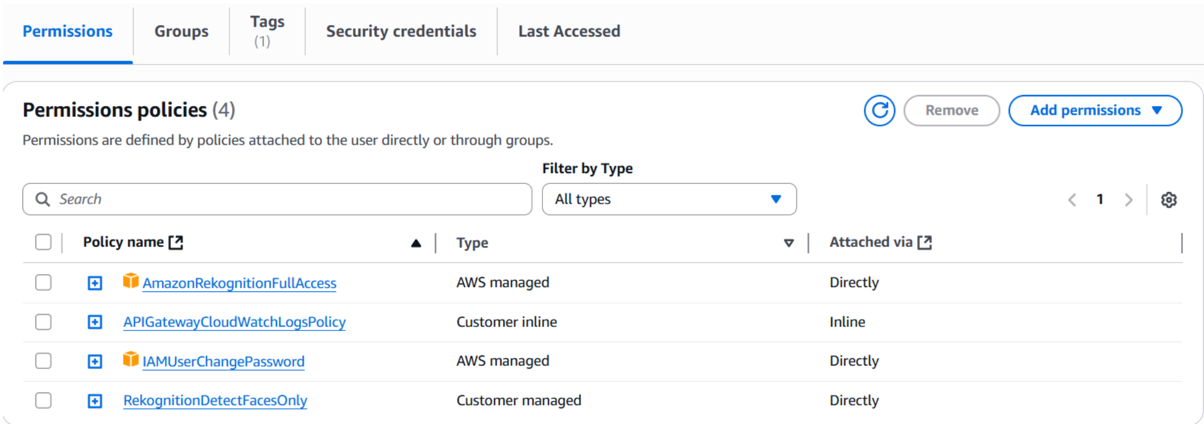


Figure 5.3.3.3 Screenshot User Permission Policies

In the AWS Management Console, we need to define a role with permissions for AWS Rekognition and assign it to the user we created. To create the role, click the 'Create role' button and follow the provided steps. During setup, we must specify the trusted entities as shown in Figure 5.3.3.2 and attach the necessary permissions policy to access the AWS Face Rekognition service.

To add permission policies to the user, click 'Add permissions' and follow the steps guided by AWS. The required information when adding the policies includes the policy name and type as shown in Figure 5.3.3.3.

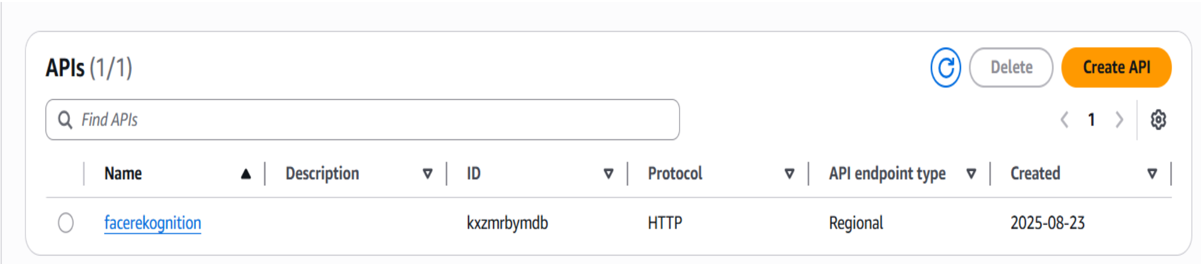


Figure 5.3.3.4 Screenshot API Gateway

In the AWS API Gateway, we need to create a new API to facilitate the connection between our application and the AWS Face Rekognition service. To create the API, click the 'Create API' button and follow the steps provided by AWS. During the creation process, we must specify the protocol type as shown in Figure above used to build the API gateway.

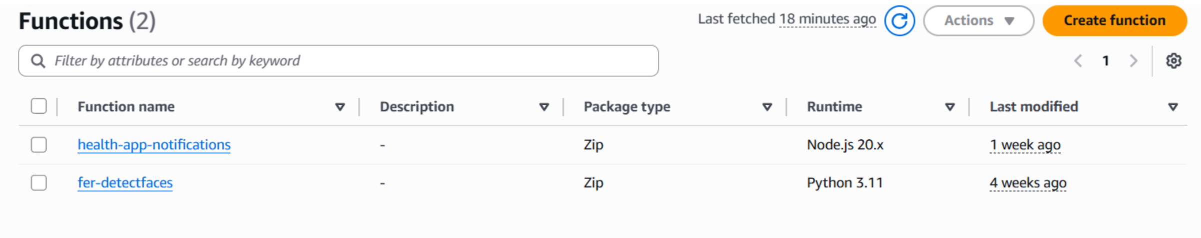


Figure 5.3.3.5 Screenshot Lambda Function

In AWS Lambda, we need to create a function to manage the HTTP communication between our application and the AWS Face Rekognition service. This is done by clicking the 'Create function' button and following AWS guided setup process where we need to define the function name, select a runtime environment like Node.js based on the figure above and write the function logic to handle requests and responses. The function needs to be programmed to return specific HTTP status codes which are STATUS 200 for a successful request, STATUS 400 for a bad request due to invalid input and STATUS 500 for an internal server error that prevents the request from being processed.

5.3.4 ChatGPT Service Setup

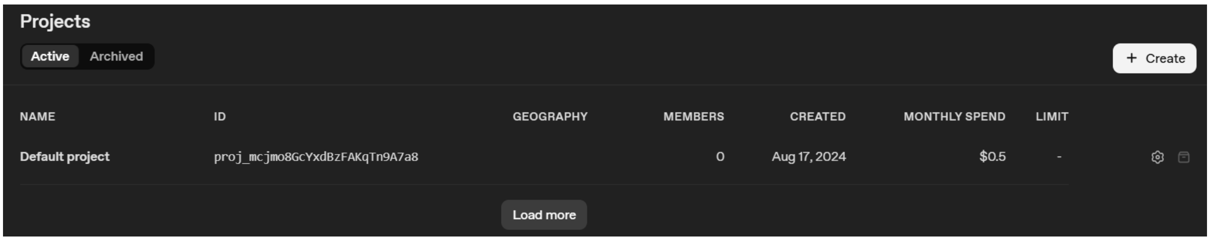


Figure 5.3.4.1 Screenshot OpenAI API Project

On the OpenAI API platform, we need to create a project to use the ChatGPT service in our application. To create the project, click the 'Create' button and define your project name.

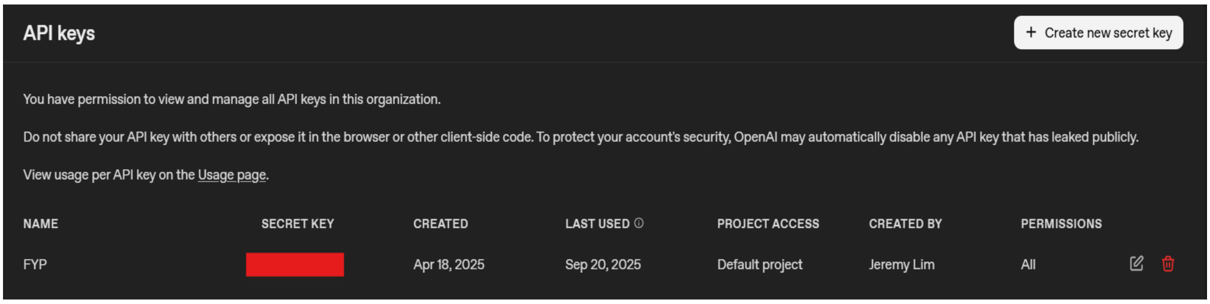


Figure 5.3.4.2 Screenshot OpenAI API Keys

On the OpenAI platform, we need to create an API key to allow our application to use the ChatGPT service. To do this, click the 'Create new secret key' button and follow the guided steps. The required information includes a name for the API key and selecting the project created earlier as shown in Figure 5.3.4.1. Once successfully generated, the API key will be provided by OpenAI. Due to privacy concerns, the API key has been hidden in the figure above.

5.3.5 Google Fit Service Setup

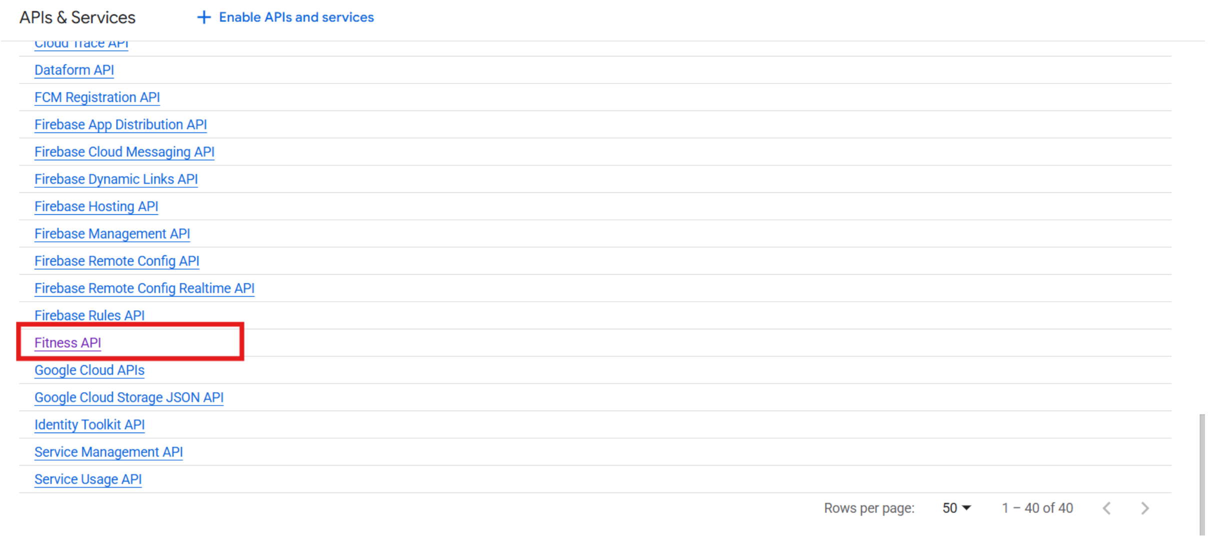


Figure 5.3.5.1 Screenshot Google Fit API

Based on the figure above, we need to enable the Google Fit API for our project. This is done in the Google Cloud Console by clicking 'Enable APIs and Services,' searching for 'Fitness API' and then clicking the 'Enable' button. This configuration is performed in the Google Cloud Console project that is automatically linked to your Firebase project when you create it and add your mobile health application.

5.4 System Operation

5.4.1 Login Interface Screen

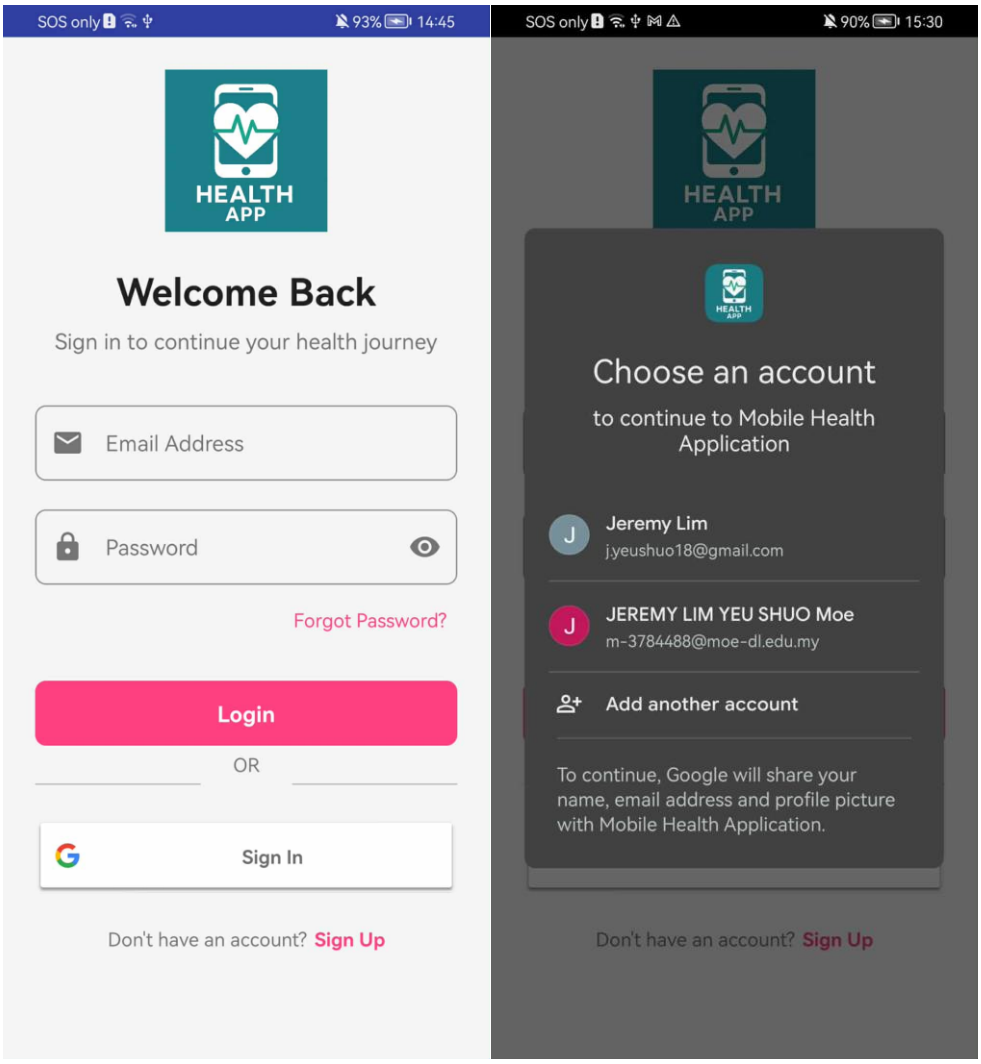


Figure 5.4.1 Login Interface Screen

On the login screen, users can either sign in with their email and password which are verified by Firebase Authentication or use Google Sign-In which prompts them to select a Google account for authentication.

5.4.2 Reset Password Interface Screen

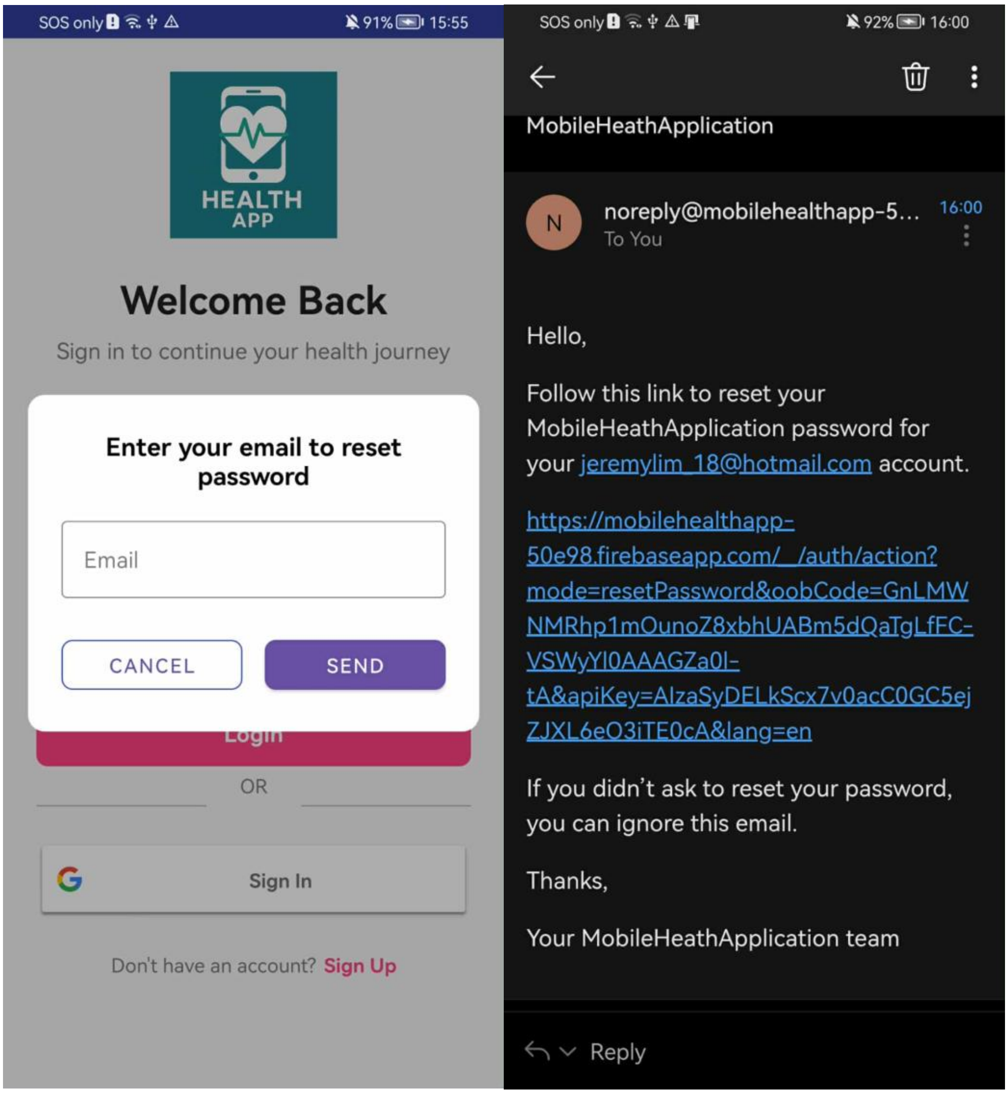


Figure 5.4.2.1 Reset Password Interface Screen

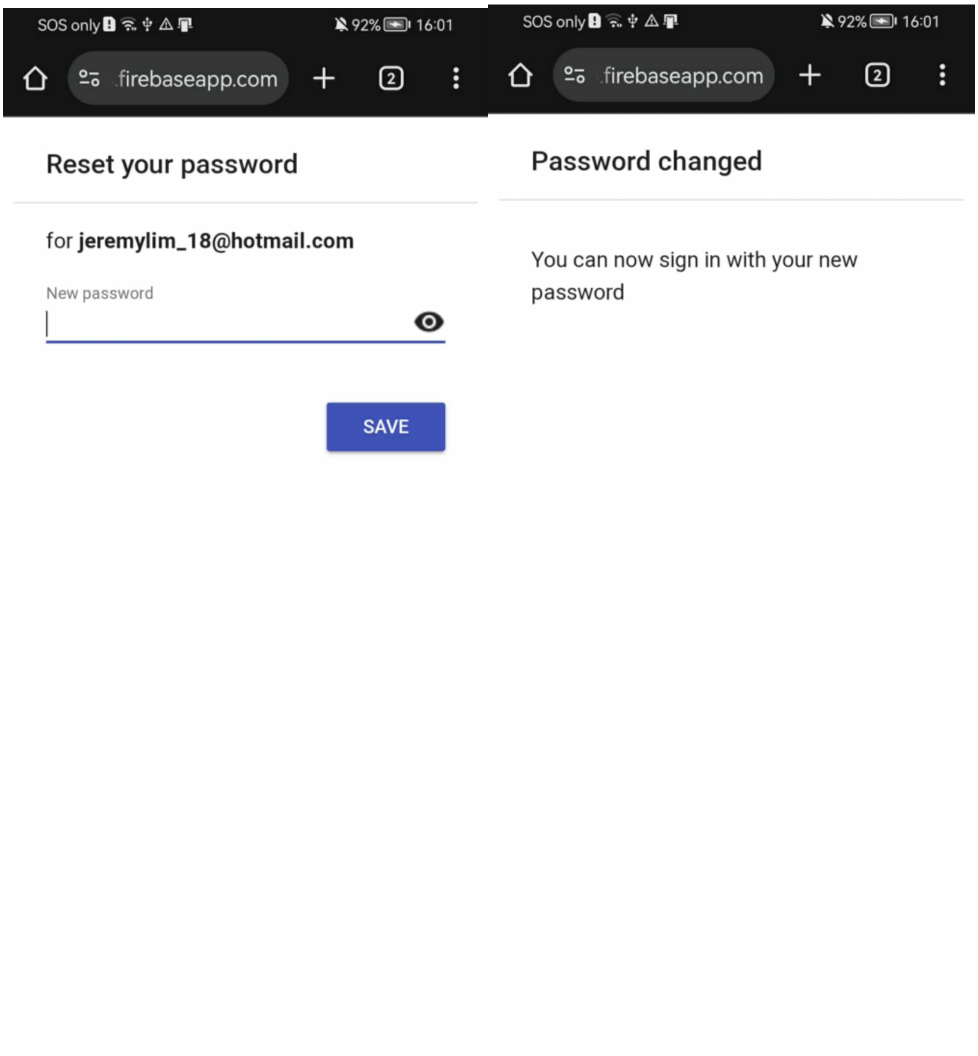


Figure 5.4.2.1 Reset Password Interface Screen

If the user forgets their password, they can click the 'Forgot Password' button, which will open a dialog interface prompting them to enter their email address. After clicking 'Send' button, Firebase will automatically send a password reset link to the provided email. The user must then check their inbox, click the reset link and set a new password. Once completed, they can return to the application and log in using their email address and the newly created password.

5.4.3 Sign Up Interface Screen

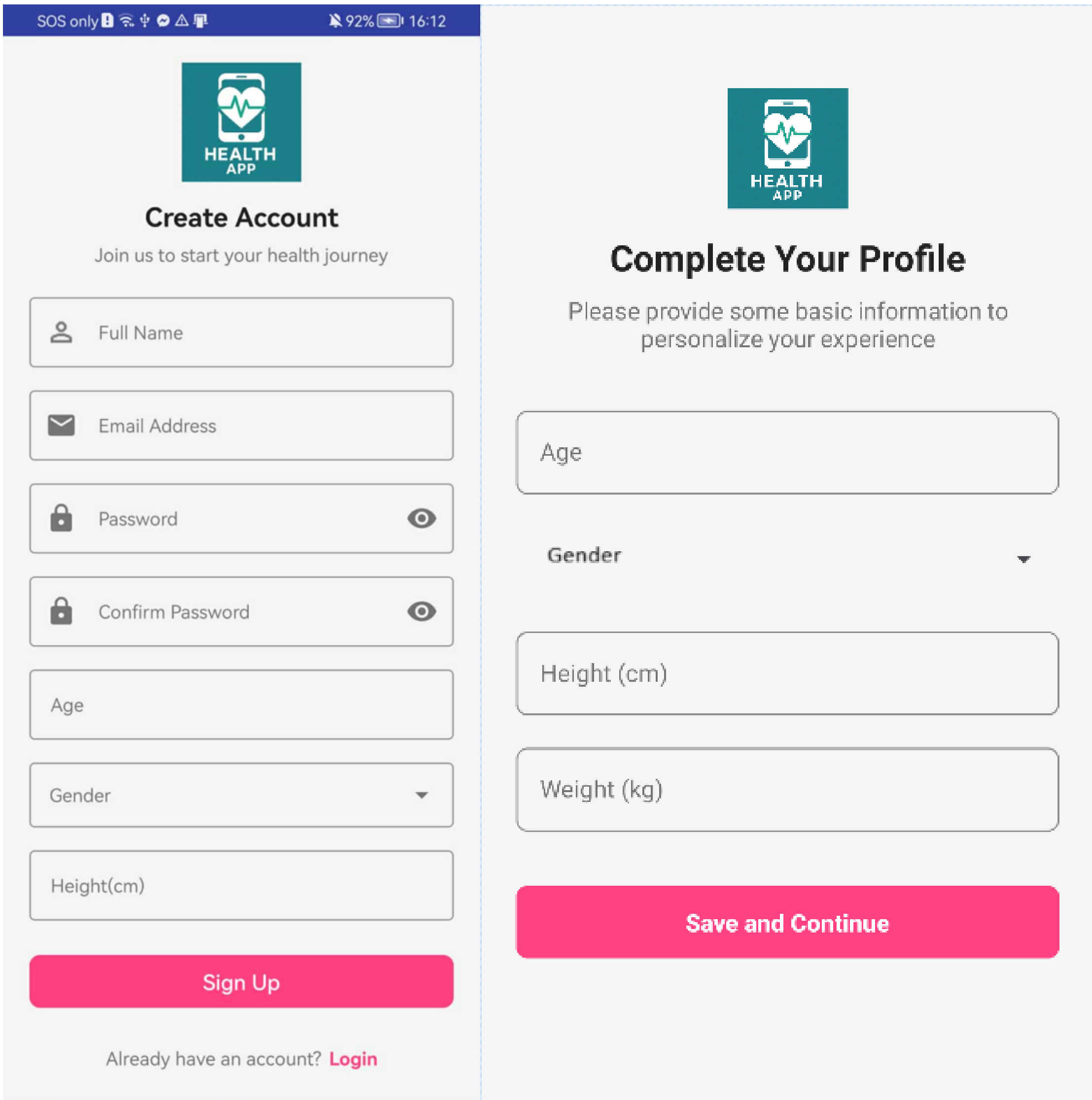


Figure 5.4.3.1 Sign Up Interface Screen

If a user does not have an account, they can click the 'Sign Up' button on the login interface and input the necessary information, as shown in the figure above. The interface on the right side of the figure will appear if the user logs in for the first time using the Google Sign-In method. In this case, the user must also complete their profile by providing the required information on the 'Complete Your Profile' screen. After clicking the 'Sign Up' or 'Save and Continue' button, the user will be automatically logged into the application.

5.4.4 Main Interface Screen

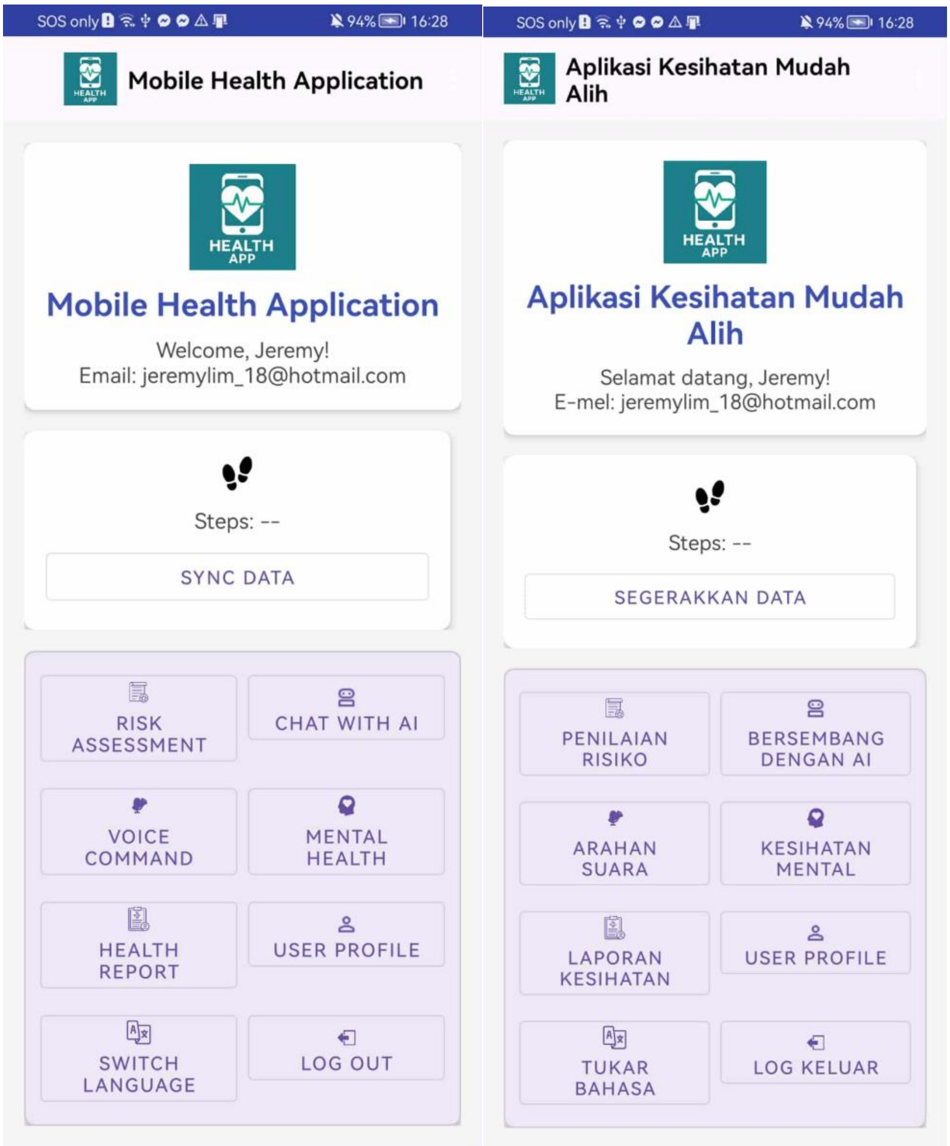


Figure 5.4.4.1 Main Interface Screen

The main interface screen contains six buttons that provide access to the application's various modules. When the user clicks the voice command button, they can speak the name of a module and the application will navigate to the relevant interface. For example, if the user says 'open chatbot' then the application will directly open the chatbot interface.

Users can manually synchronize their step count data by clicking the 'Sync Data' button. The application will check Firebase for step count data associated with the user, retrieve the information and display it on the main interface.

The right side of the interface shows the Bahasa Melayu translation which is designed specifically for Malaysian users who may not be comfortable with English. By clicking the 'Switch Language' button and selecting Bahasa Melayu, users can change the application's language from English to Malay.

5.4.5 Chronic Disease Risk Dashboard Interface Screen

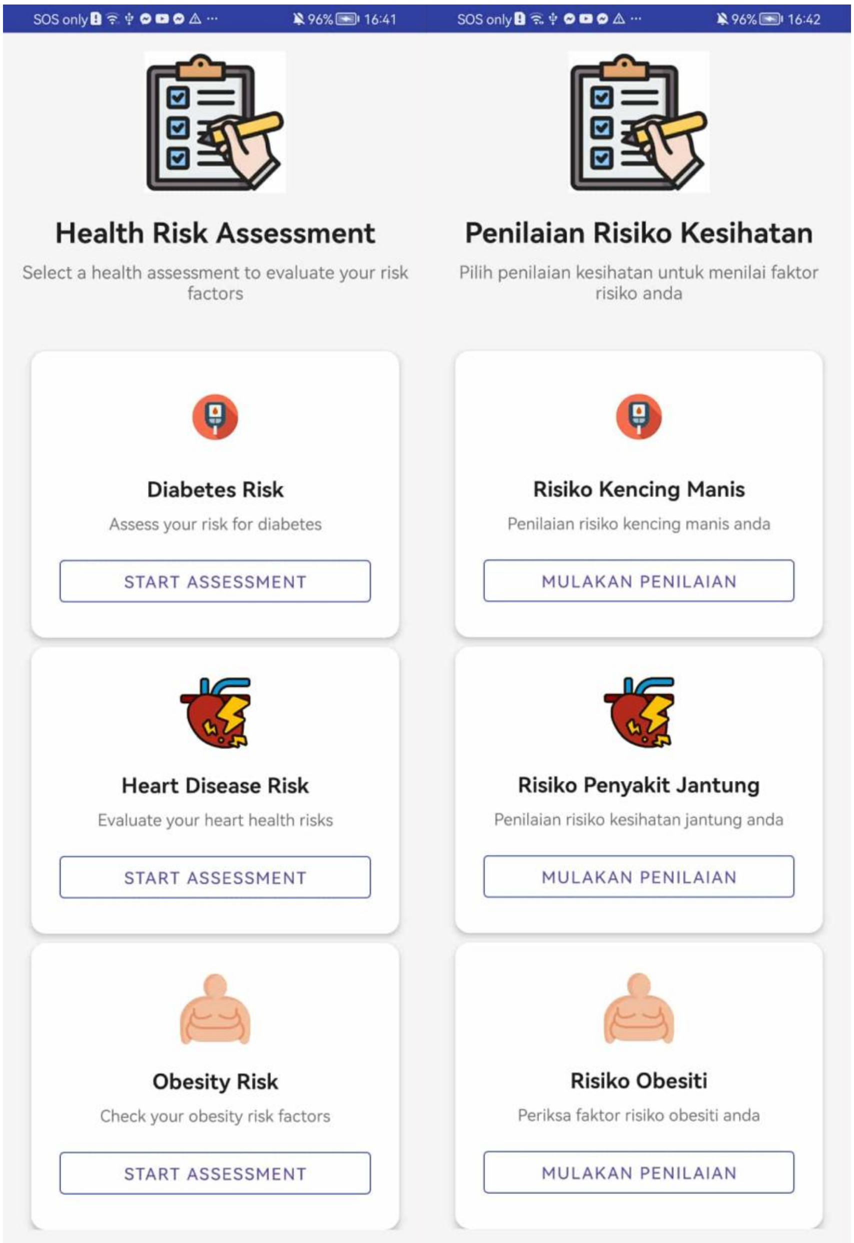


Figure 5.4.5.1 Chronic Disease Risk Assessment Dashboard Screen

Based on the figure, the dashboard displays three chronic disease risk assessments which are Diabetes Risk, Heart Disease Risk which representing cardiovascular disease and Obesity Risk. Users may select any of these options to proceed with the relevant risk assessment.

5.4.6 Diabetes Risk Assessment Interface Screen

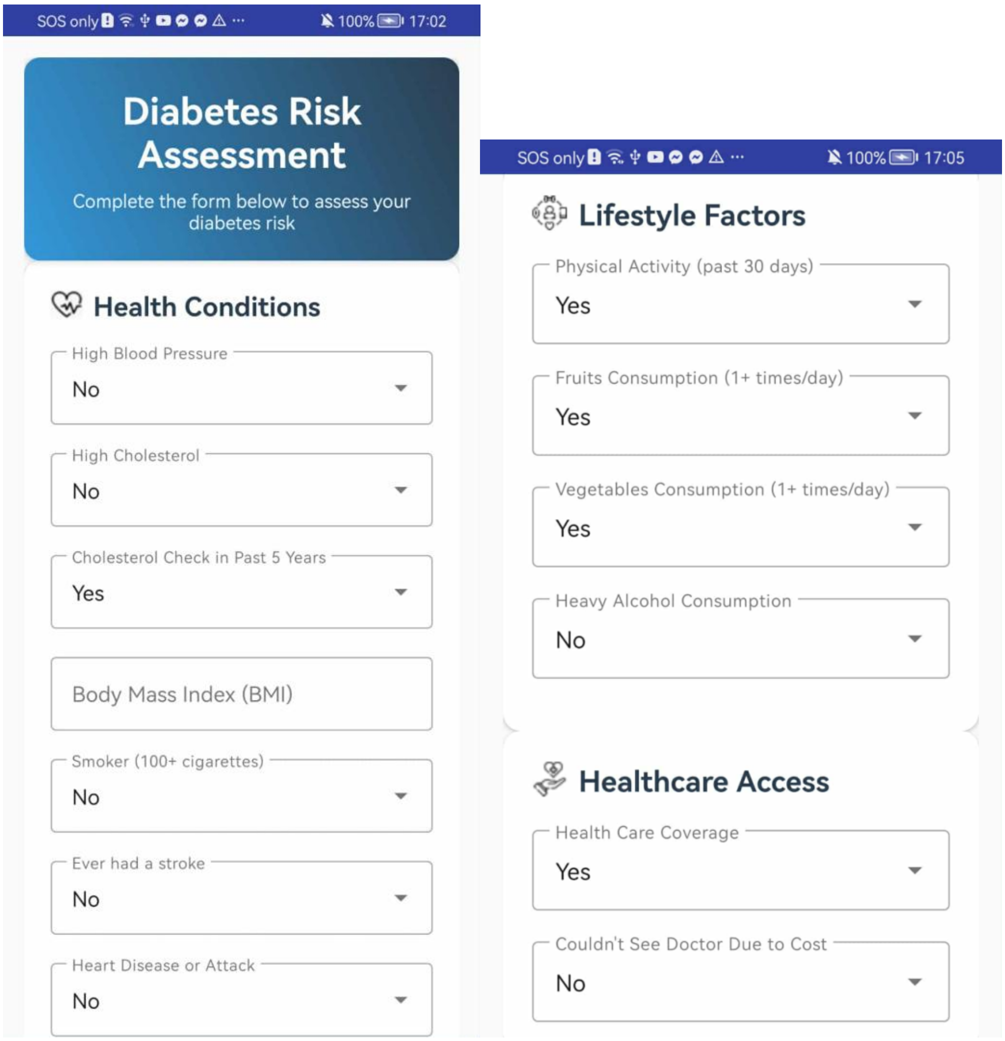


Figure 5.4.6.1 Diabetes Risk Assessment Interface Screen

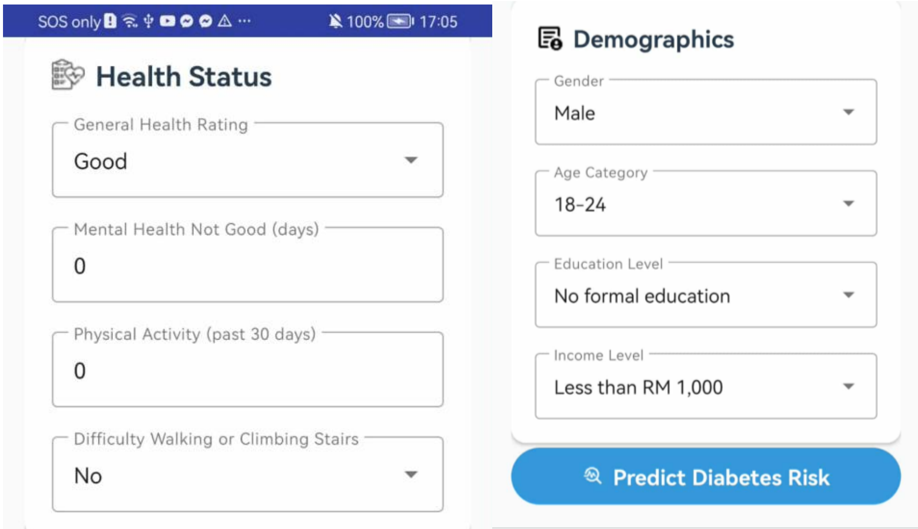


Figure 5.4.6.1 Diabetes Risk Assessment Interface Screen

Based on the figure above, users need to provide the necessary information in the diabetes assessment interface to allow the application to predict their diabetes risk. Once the user clicks the 'Predict Diabetes Risk' button, the application validates the input. If validation is successful, the data is sent to the diabetes machine learning model for prediction. After the prediction is complete, the results are forwarded to ChatGPT to generate personalized recommendations which are then displayed on the results interface screen as shown in the next figure.

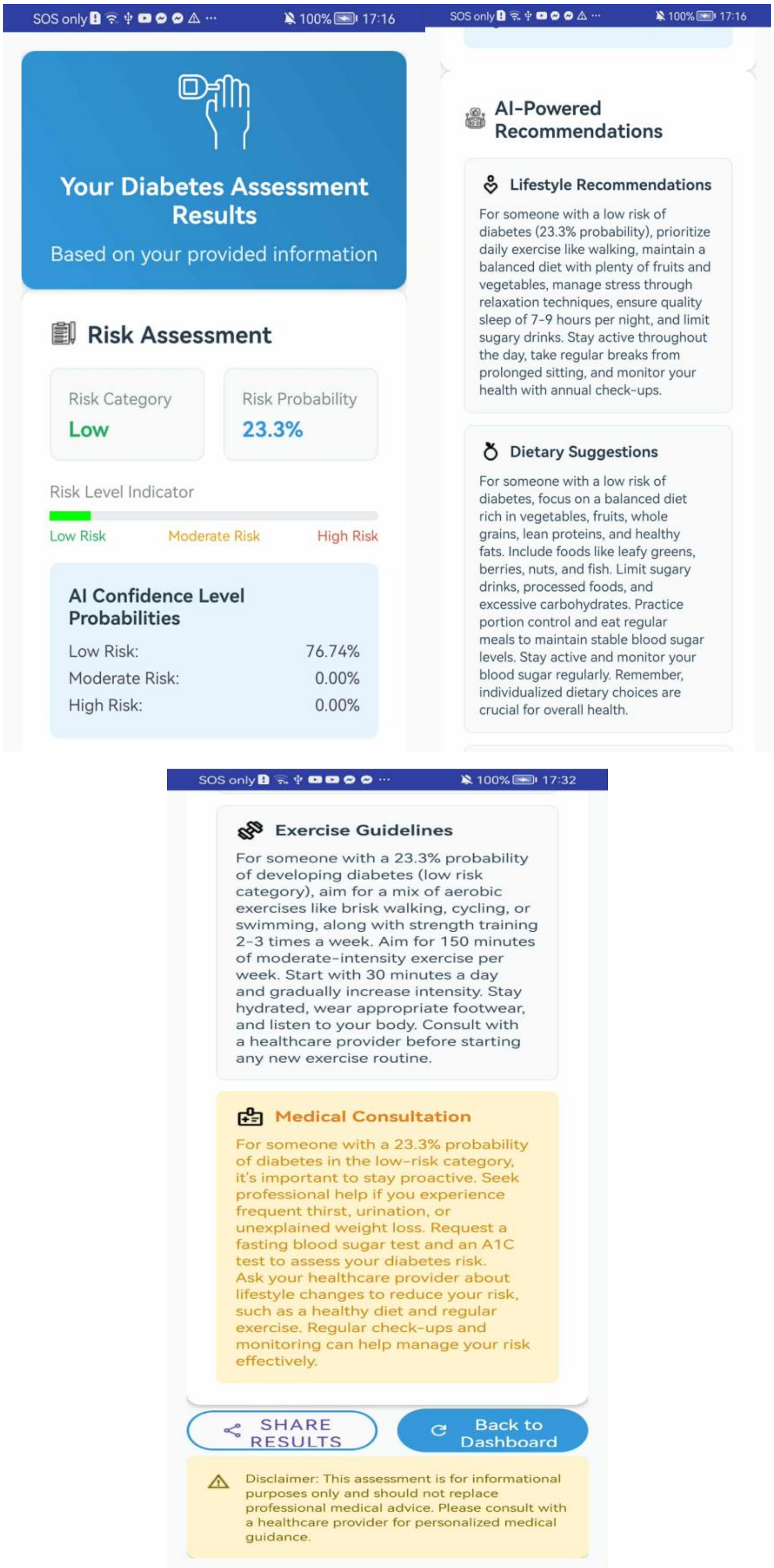


Figure 5.4.6.2 Diabetes Risk Assessment Result Screen

Based on the figure above, the diabetes risk assessment results screen displays the user's risk category which are Low, Moderate, or High-risk probability percentage, AI confidence level for the prediction, and AI-powered lifestyle, dietary, exercise and medical consultation recommendations generated by ChatGPT. A disclaimer is included to inform users that the results are for informational purposes only and advises them to visit a clinic or hospital for accurate medical diagnosis. The results of each diabetes risk assessment are stored in Firebase, allowing users to track their health condition over time.

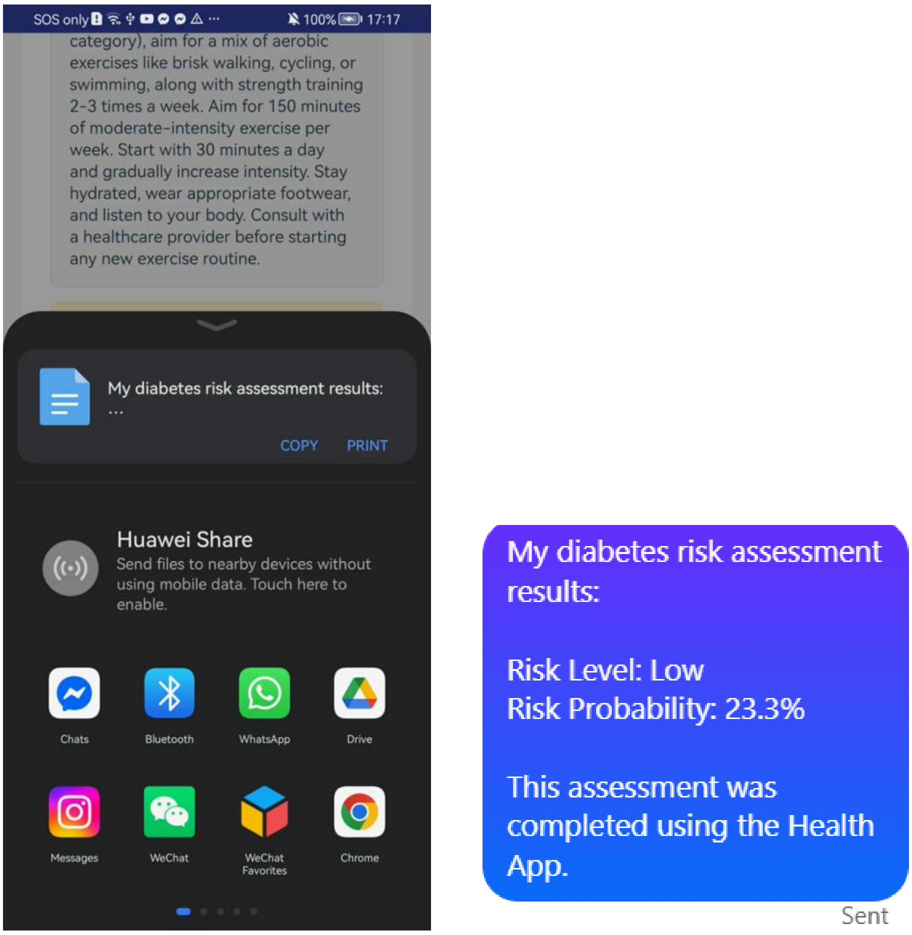


Figure 5.4.6.3 Diabetes Risk Assessment Share Result

Users can share their diabetes assessment results by pressing the 'Share Results' button and selecting their preferred social media platform. As shown in the figure above, the results can be shared via messaging apps like Messenger and sent to friends.

5.4.7 Cardiovascular Disease (Heart Attack) Assessment Interface Screen

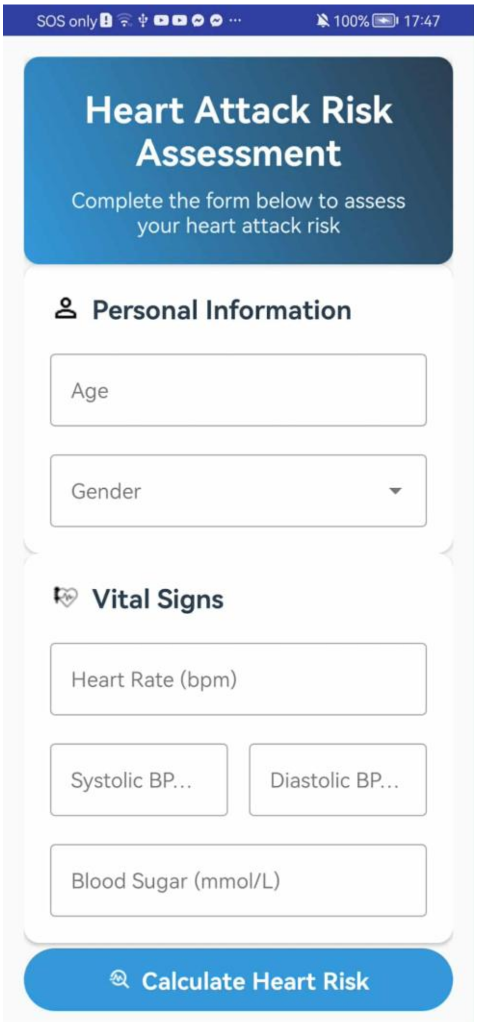


Figure 5.4.7.1 Cardiovascular Disease (Heart Attack) Assessment Interface

Based on the figure above, if the user chooses to assess their cardiovascular disease which is to predict the heart attack risk outcome, they need to provide the necessary information for the application to predict their cardiovascular disease risk level. Once the user presses the "Calculate Heart Risk" button, the application will validate the input data, send it to the machine learning model for prediction and then forward the results to both the application interface and ChatGPT to generate personalized recommendations. The results interface will be displayed as shown in the following figure.

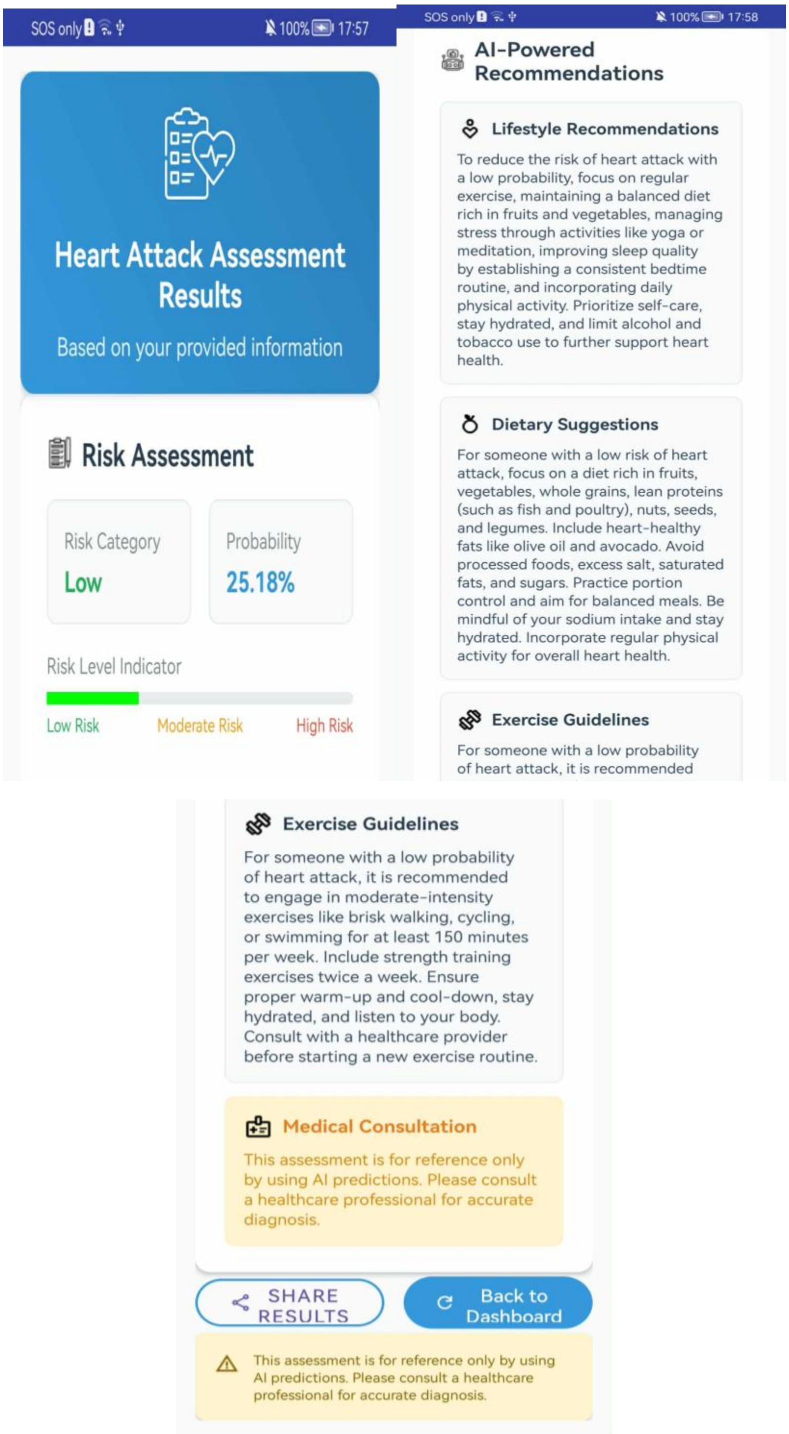


Figure 5.4.7.2 Cardiovascular Disease (Heart Attack) Assessment Result Interface

Based on the figure above, the assessment result interface will display results in a pattern similar to the diabetes module's output. Personalized recommendations generated by ChatGPT will be displayed according to the risk category and risk probability provided by the application.

5.4.8Obesity Assessment Interface Screen

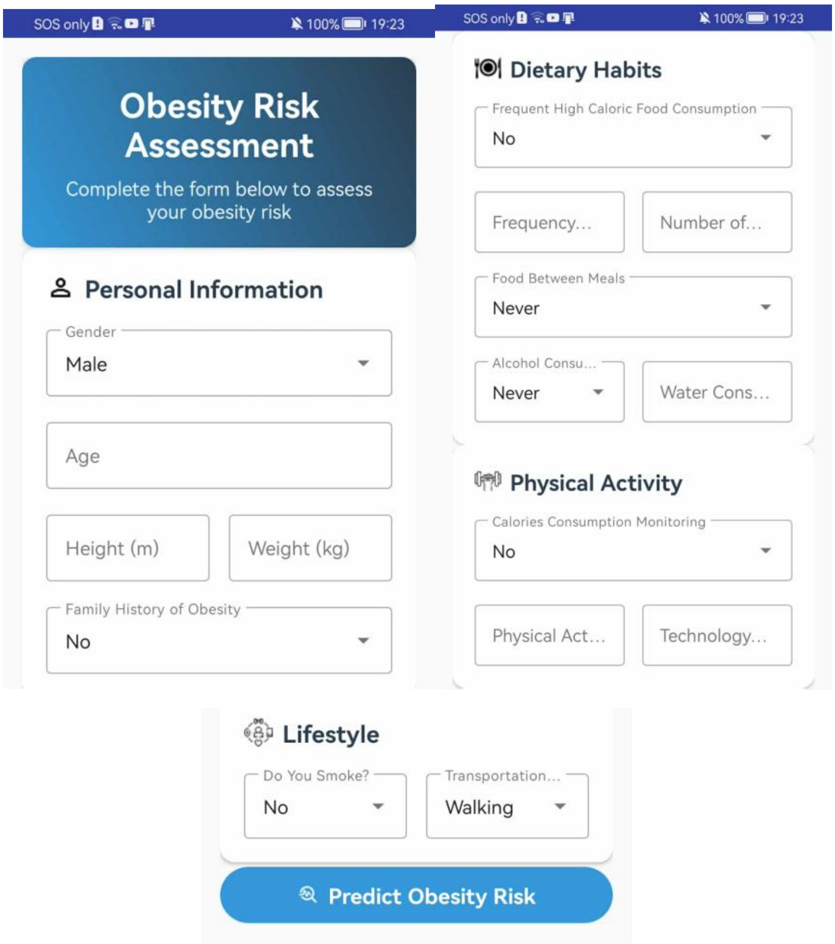


Figure 5.4.8.1 Obesity Assessment Interface

Based on the figure above, the user needs to provide the necessary information in the interface to assess their obesity risk. Once the user presses the "Predict Obesity Risk" button, the application will validate the input and send it to the obesity machine learning model for prediction. The model will then assess the user's obesity risk and send the results to both the application interface and ChatGPT to generate personalized recommendations based on the outcome. The results display interface will be shown in the following figure.

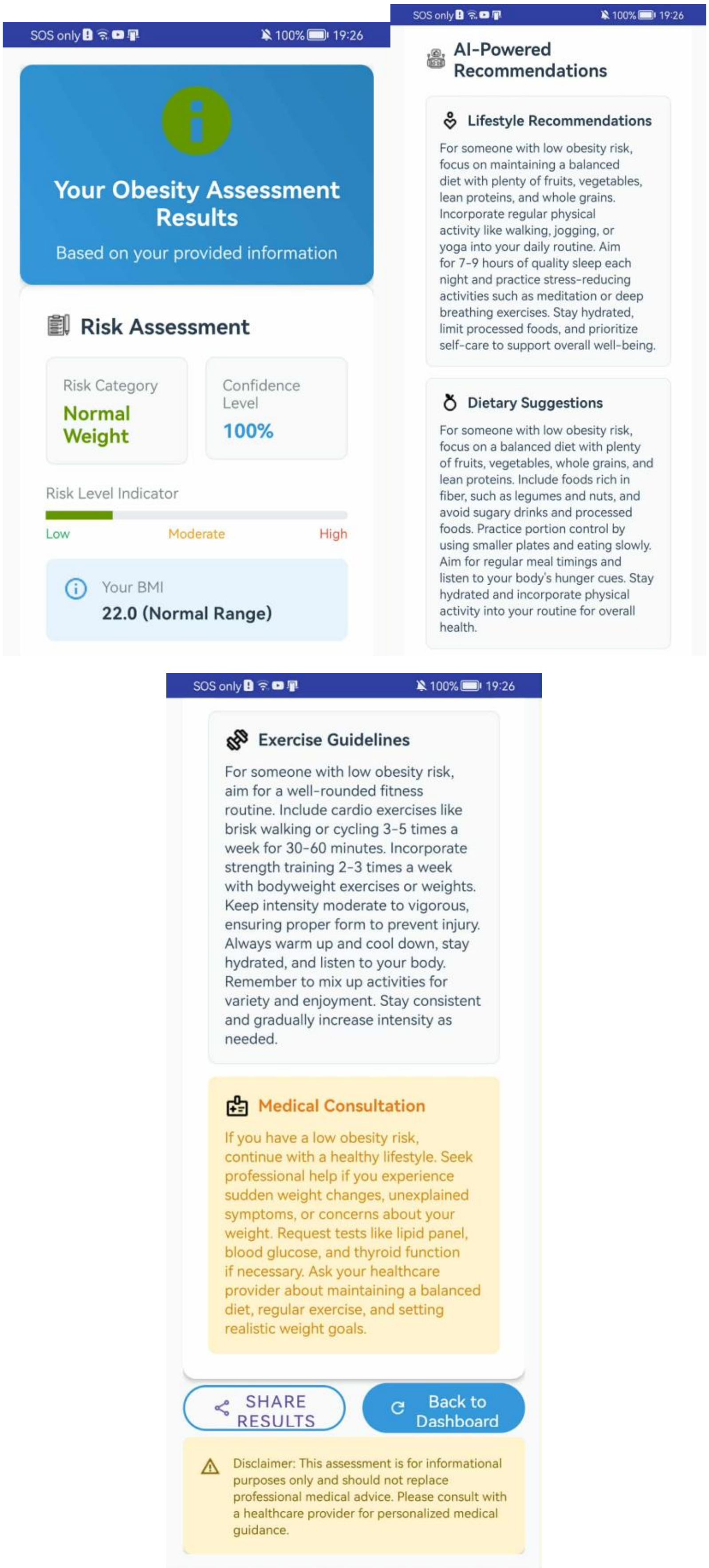


Figure 5.4.8.2 Obesity Assessment Result Interface

Based on the figure, the obesity assessment result interface displays the user's risk category, the machine learning model's confidence level, BMI range and personalized recommendations generated by ChatGPT with following a similar format to the diabetes and cardiovascular disease result interfaces. A disclaimer statement is also included to aware the user about the risk assessment purposes.

5.4.9 Chatbot Interface Screen

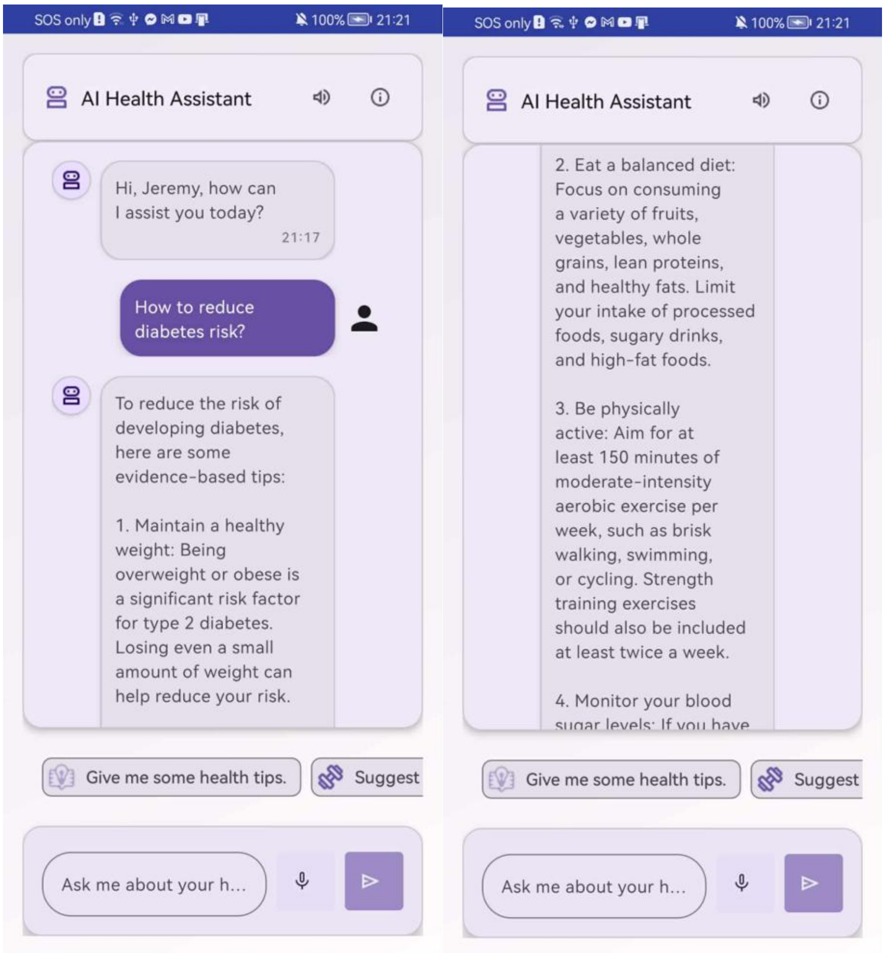


Figure 5.4.9.1 Chatbot Interface

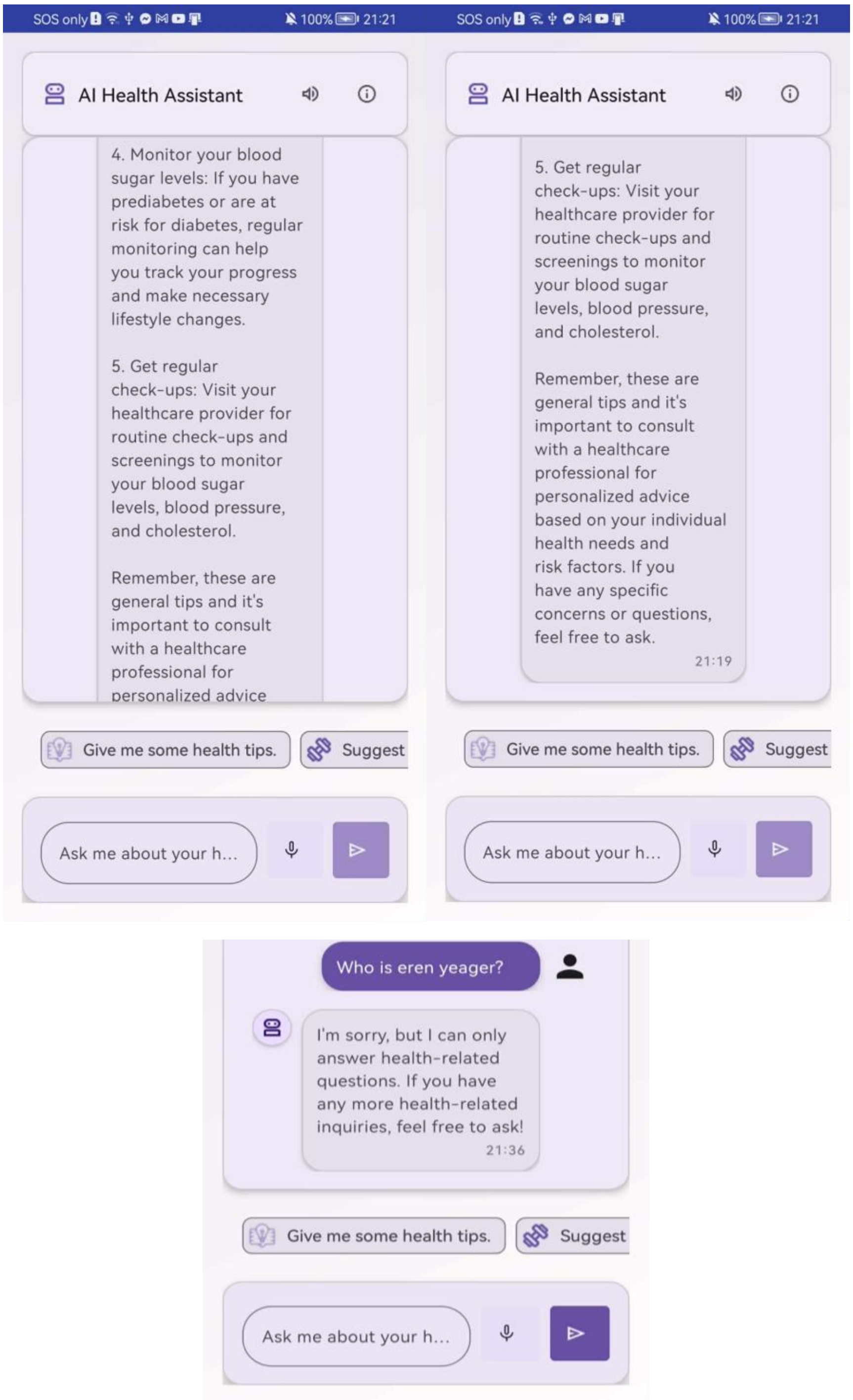


Figure 5.4.9.1 Chatbot Interface

Based on the figure above, this is the chatbot module by using ChatGPT service where users can ask healthcare-related questions. The chatbot will provide answers based on the user's queries and the answer is limited on healthcare-related questions only. If the user enables the toggle button, the responses will be delivered as voice messages. Oppositely, if the user disables the toggle button, the voice feature will be turned off and responses will be displayed as text only.

5.4.10 Mental Health Module Dashboard Interface

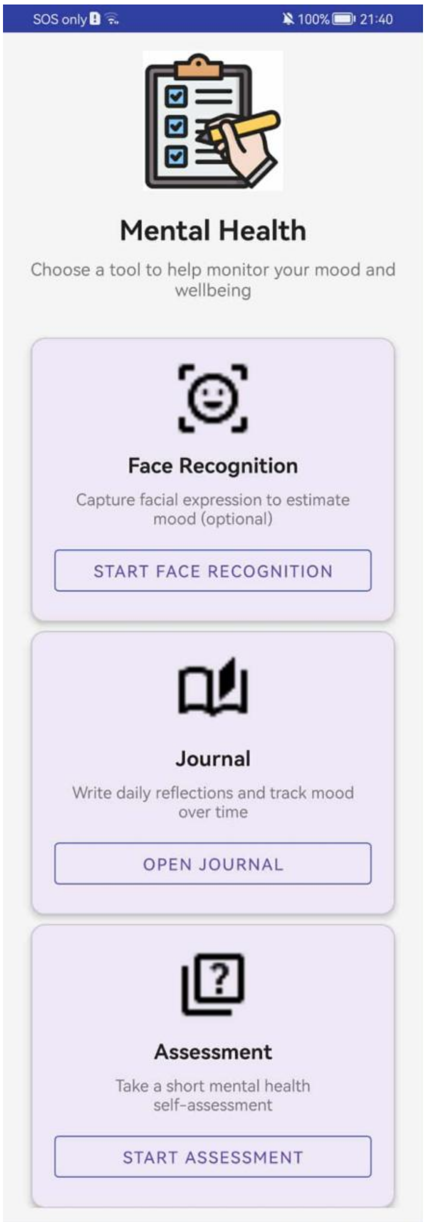


Figure 5.4.10.1 Mental Health Dashboard

The figure above shows the mental health dashboard, which includes three modules, the Face Recognition, Journal and Mental Health Assessment. Users can select any of these modules to navigate to the corresponding interface.

5.4.11 Face Recognition Module

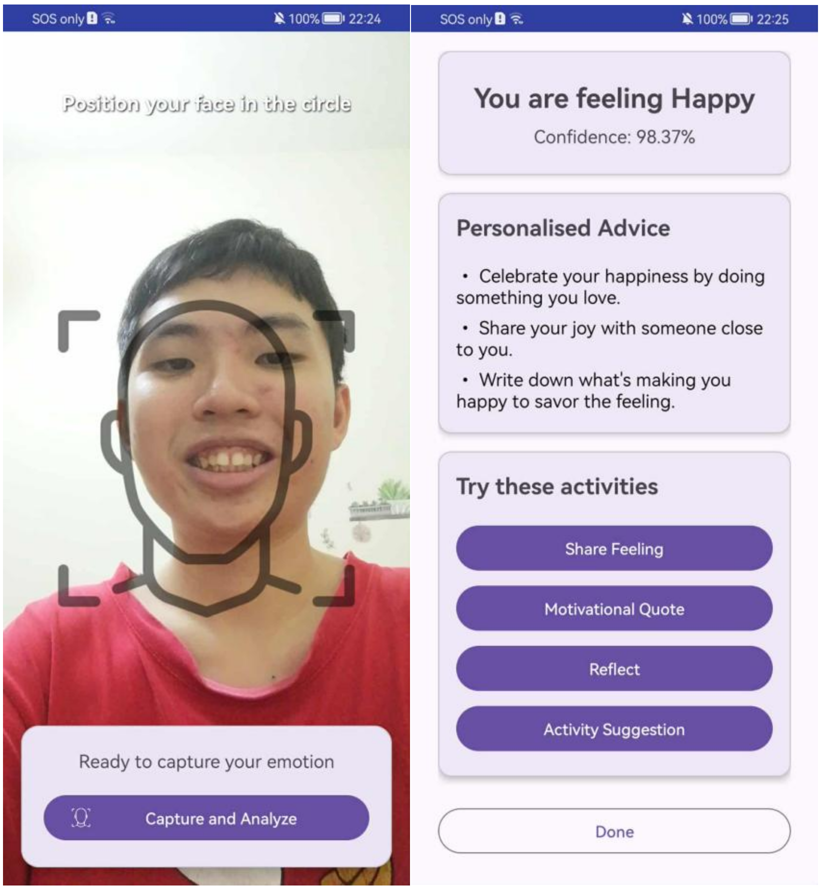


Figure 5.4.11.1 Face Recognition Interface and Result

The figure above shows the Face Recognition module feature. Users are prompted to position their face within the frame while expressing their current mood. After clicking the 'Capture and Analyze' button, the application validates the input image and submits it to AWS Face Recognition for emotional analysis. The results are then processed by ChatGPT to generate personalized advice and activity suggestions.

The results screen as shown on the right displays the user's detected emotional state along with a confidence score, accompanied by the personal tailored advice. Not only that, users can explore these suggested activities to help improve negative emotions or maintain positive ones.

Similar to the diabetes module, this feature includes a 'Share Results' option which allowing users to share their emotional assessment with friends or family through social media platforms.

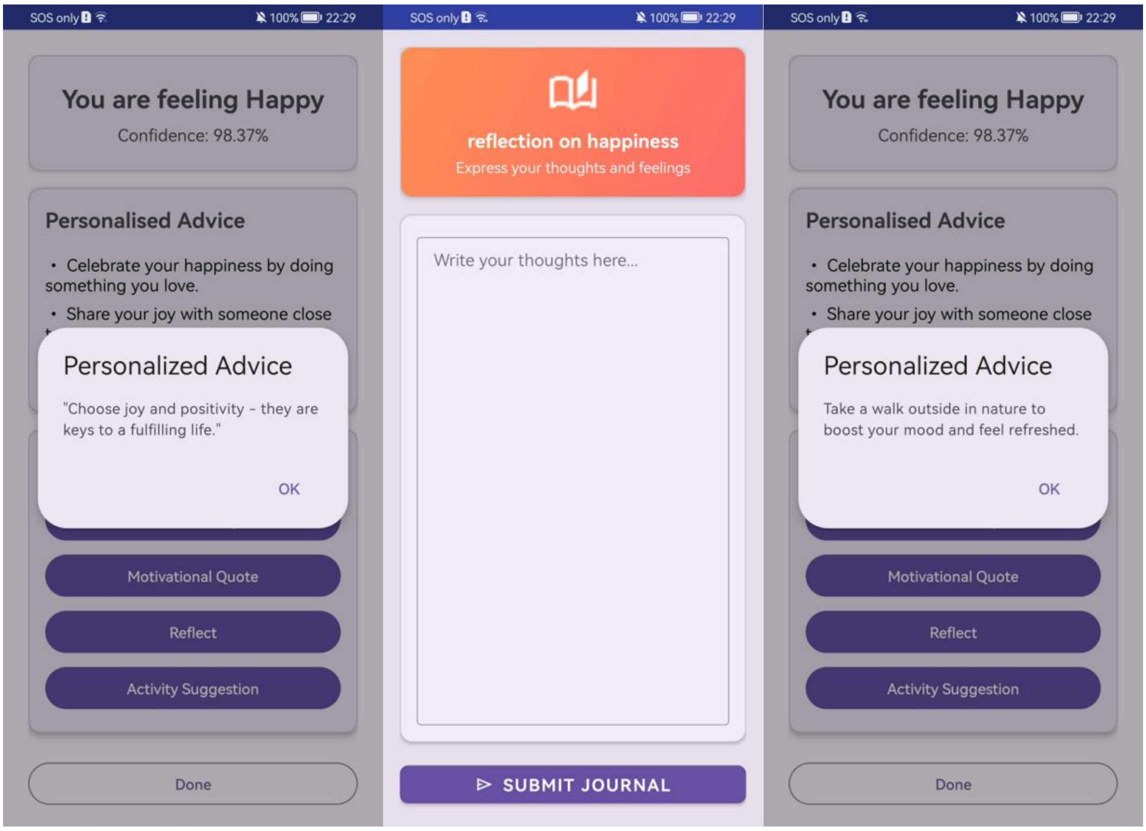


Figure 5.4.11.2 Face Recognition Activity Suggestion

When the user clicks the 'Motivational Quote' button as shown in the left-hand figure, the application generates a personalized motivational quote using ChatGPT. Similarly, clicking the 'Activity Suggestion' button as shown in the right-hand figure triggers ChatGPT to produce a customized activity recommendation.

If the user selects the 'Reflect Activity' button as shown in the center figure, they are redirected to the journal module, where they can record their current feelings. The journal entry is then analyzed by ChatGPT which performs text-sentiment analysis on the input to provide personalized insights.

5.4.12 Journal Prompt Interface

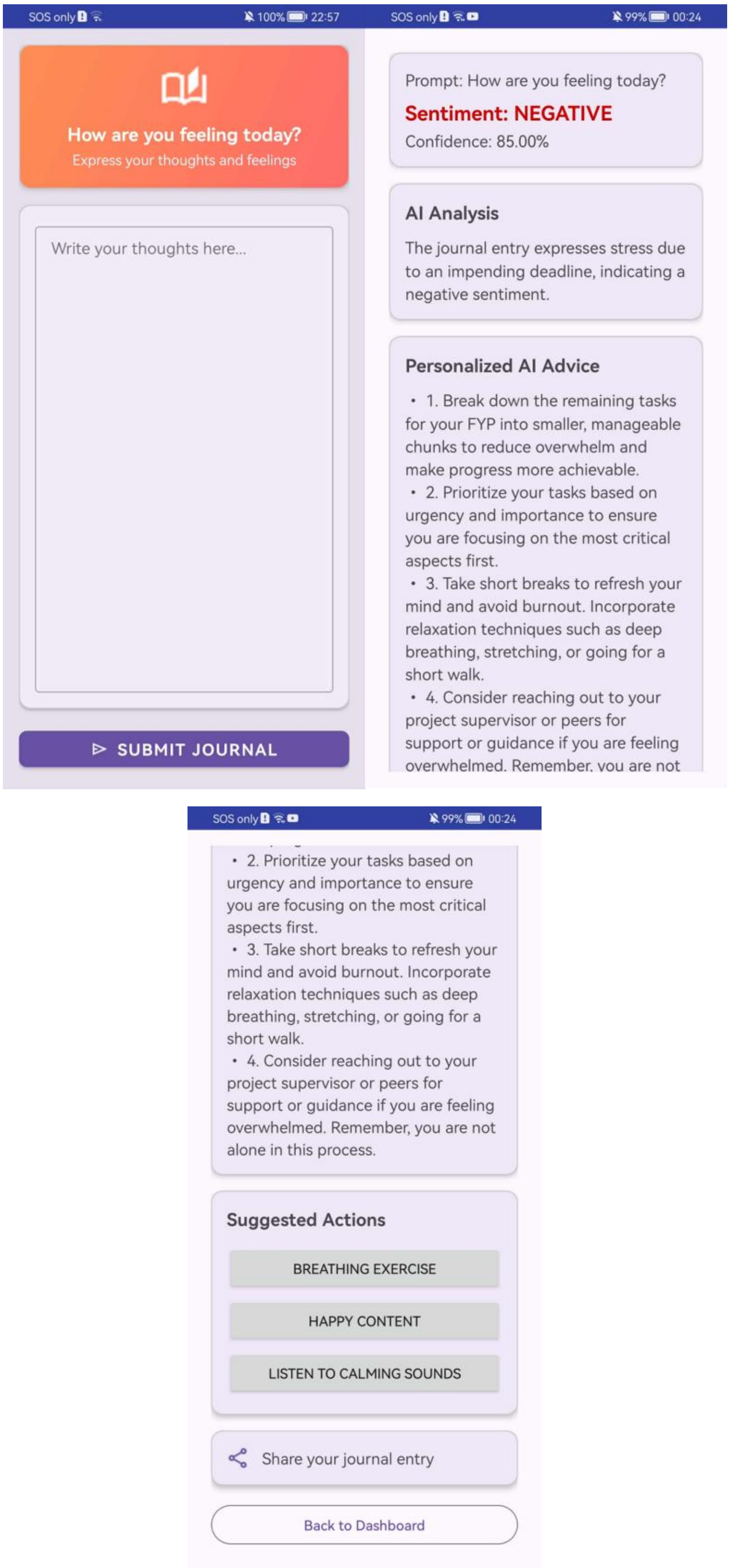


Figure 5.4.12.1 Journal Module Interface and Result

Based on the figure above, users can write about their current feelings in the provided input box. Upon clicking the 'Submit' button, the application validates the input and sends the journal entry to ChatGPT for text-sentiment analysis. The analyzed results which including the sentiment classification, confidence value, and AI-generated interpretation are returned to the application and displayed in the results interface.

For example, as shown in the figure, a user mentioned feeling stressed due to an upcoming FYP deadline. ChatGPT processed this input and generated personalized advice, which is displayed on the right-side results screen. The application also suggests activities similar to those in the Face Recognition module to help the user manage their emotions.

Additionally, the 'Share Your Journey' button allows users to share their journal insights with friends or family via social media applications which similar to the functionality in the Diabetes module.

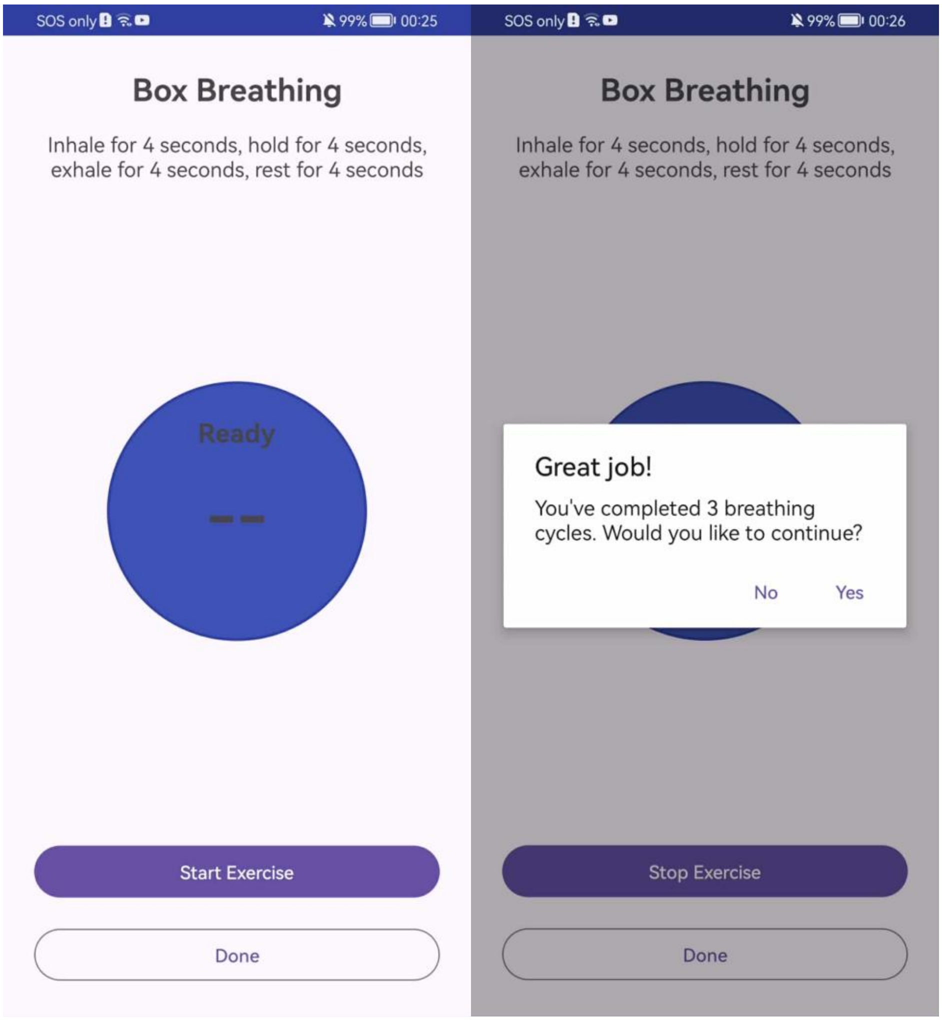


Figure 5.4.12.2 Breathing Exercise Interface

When the user presses the breathing exercise button, the application displays the interface shown in the figure, which provides instructions for the user to follow. The user can press the 'Start Exercise' button to begin the breathing exercise. After completing three cycles, the application will prompt the user to decide whether to continue. If the user chooses 'Yes,' the exercise continues, otherwise, it stops.

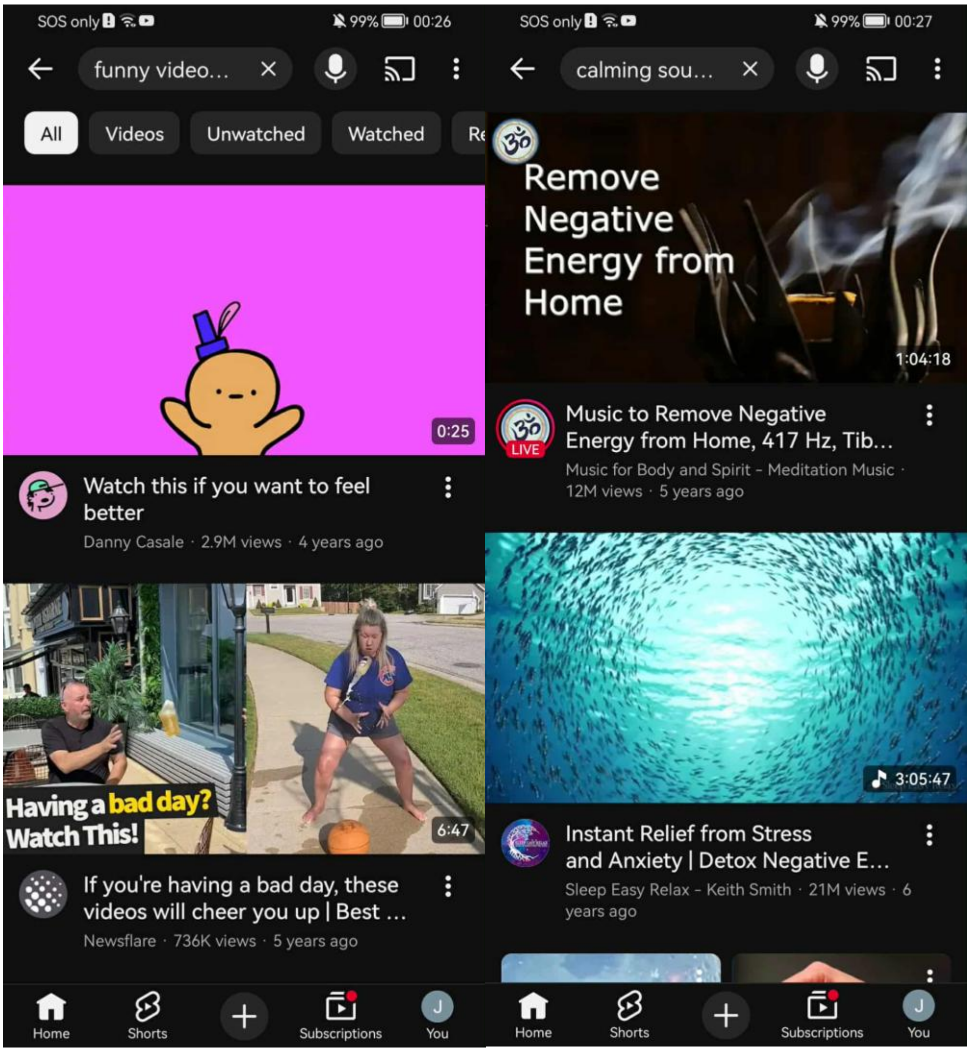


Figure 5.4.12.3 Funny Content and Calming Sound Interface

Based on the figure above, pressing the Happy Content button as shown in left-hand figure triggers ChatGPT to generate and search for funny videos on YouTube by helping to reduce the user's negative emotions. Similarly, pressing the Calming Sound button prompts ChatGPT to find calming sounds on YouTube which can also help reduce negative emotions.

5.4.13 Mental Health Assessment Interface

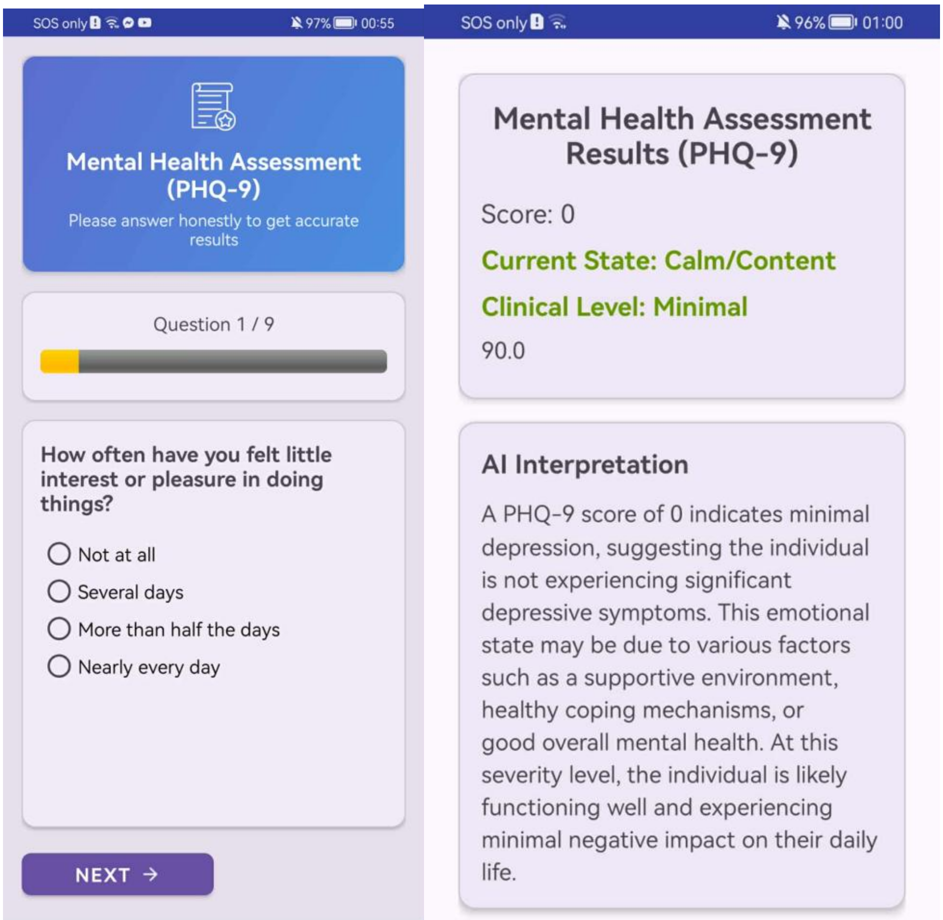


Figure 5.4.13.1 Mental Health Assessment Interface and Result

Based on the figure above, the left-hand side shows the mental health assessment interface. When the user selects this option, they are required to answer questions based on the PHQ-9 standard which helps evaluate their current level of depression severity.

After completing all questions, the application calculates a score according to the PHQ-9 scoring system. This score is then sent to ChatGPT to generate an AI interpretation as shown in the right-hand figure and personalized recommendations that will be displayed in the next figure.

The results will be presented to the user include their total score, current emotional state, clinical depression severity level and related activity suggestions.

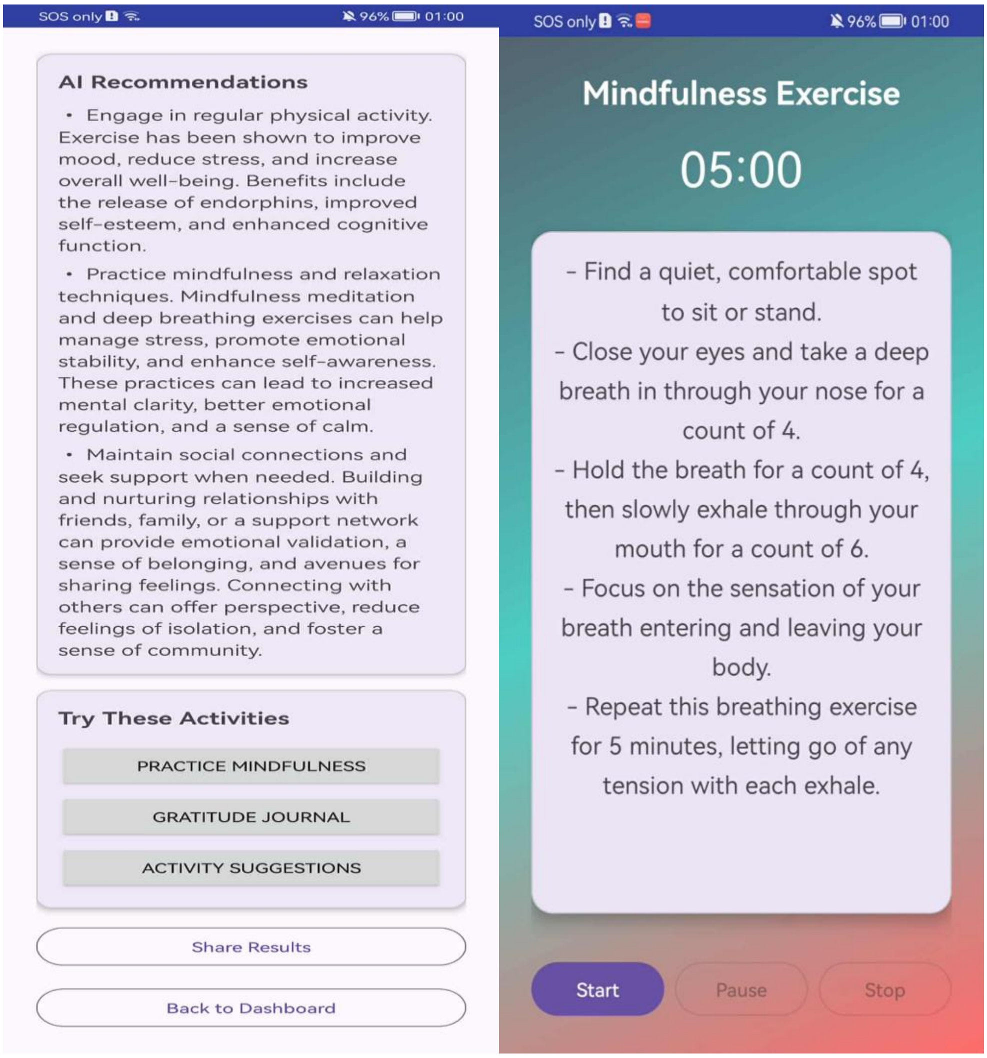


Figure 5.4.13.2 Recommendations and Mindfulness Exercise Interface

Based on the figure, users can engage with suggested activities such as the gratitude journal and activity recommendations, which function similarly to the Face Recognition module. The 'Share Results' button also similar to the functionality in the diabetes assessment module which allowing users to share their results with friends or family via social media.

When the user presses the 'Practice Mindfulness' button, the interface shown in the right-hand figure is displayed. During the mindfulness exercise, instructions are uniquely generated each time by ChatGPT. Users can begin the exercise by clicking the 'Start' button which initiates a 5-minute timer. The exercise can be paused at any time or ended completely by selecting the 'Stop' button.

5.4.14 Health Report Interface

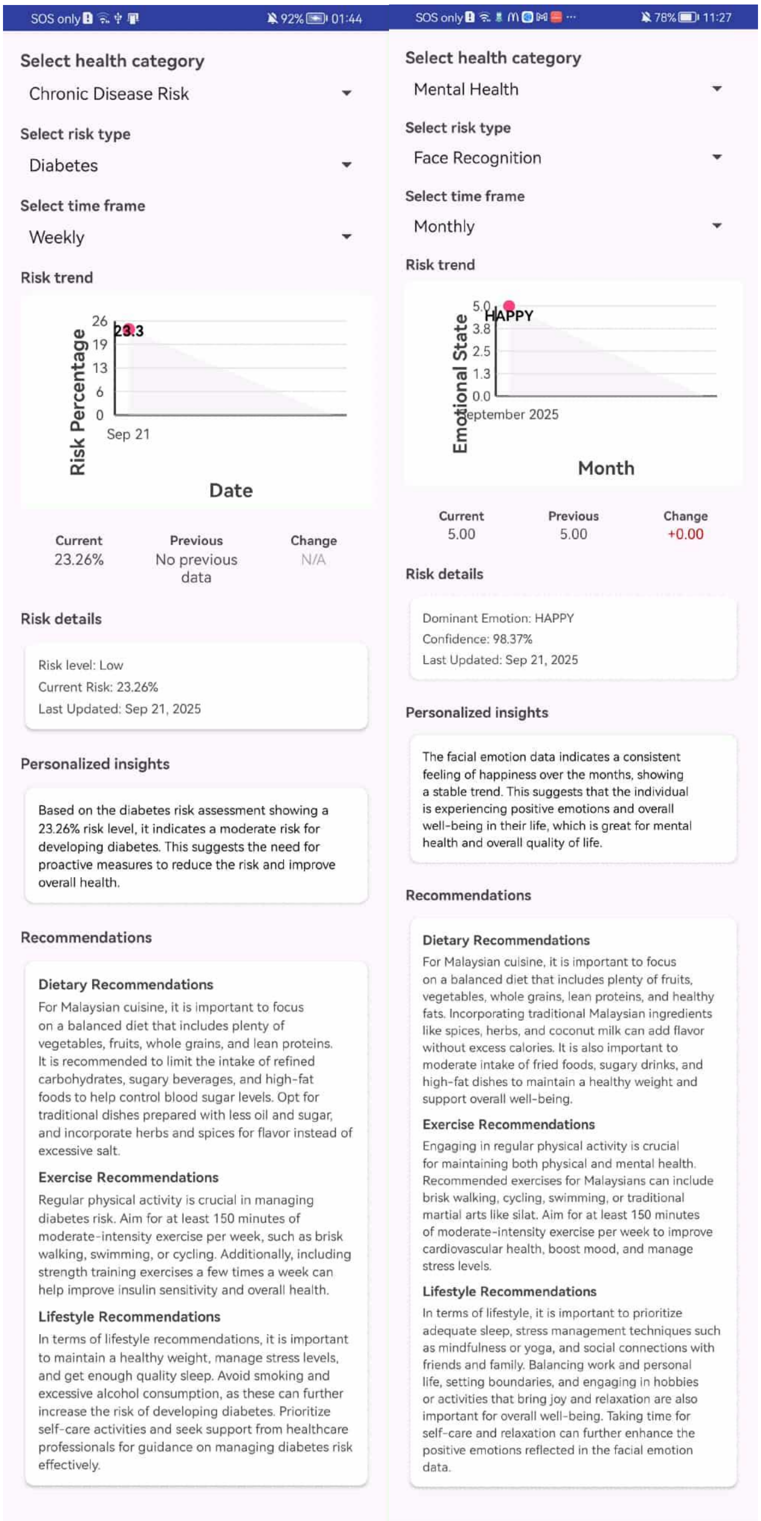


Figure 5.4.14.1 Health Report Interface

Based on the figure above, users can generate a health report by selecting their desired health category, risk type and timeframe. Once all selections are made, the application displays the relevant historical data in a line chart and shows risk details based on the user's most recent assessment from the corresponding module.

The report also provides personalized insights and recommendations generated by ChatGPT and offering tailored advice based on the analyzed health data. This feature enables users to track both their physical and mental health conditions over time which helping them to monitor trends and make informed decisions about their wellbeing.

5.4.15 Update Profile Interface

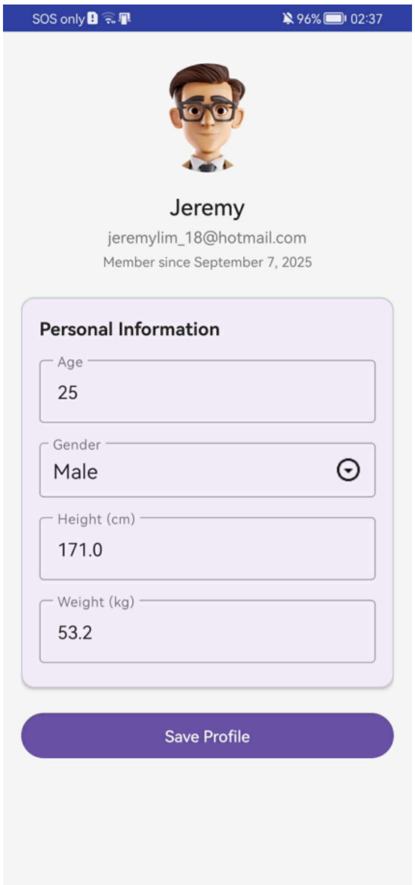


Figure 5.4.15.1 Update Profile Interface

Based on the figure above, users can update their personal information by selecting the user profile button from the main interface. Within the profile update screen, users may modify relevant data such as their age, gender, height, and weight. After clicking the 'Save Profile' button, the application validates the input and updates the user's profile by writing the latest information to the Firebase Firestore database.

5.5 Implementation Issues and Challenges

The development of this mobile health application reviews several implementation challenges across its various modules, especially in integrating on-device processing for chronic disease risk assessment and emotion detection.

Firstly on-device machine learning is the primary challenge in this mobile health application. Implementing on-device models for chronic disease risk prediction which are diabetes, obesity, and cardiovascular disease and emotion detection via AWS Face Recognition involved significant challenges. The primary difficulty was balancing model size and speed to ensure accurate predictions without exceeding the memory and processing constraints of an Android device. While larger models may offer better accuracy but they often run too slowly or consume excessive memory. Techniques such as converting models to TensorFlow Lite or ONNX format and applying quantization were essential however they required careful testing to avoid a substantial loss in prediction accuracy.

Next, managing three distinct AI pipelines which are chronic disease risk assessment, text sentiment analysis for journals module and facial emotion detection has added a layer of complexity to the application. Each model requires unique input preprocessing, resource allocation and output handling. A major concern was optimizing battery usage and maintaining user privacy by processing data locally on the device whenever possible.

Not only that, creating a cohesive, personalized health assistant that seamlessly integrates quantitative health metrics, mood readings from emotion analysis and natural-language dialogue via ChatGPT was another considerable challenge. The system requires sophisticated logic to determine whether a user's query is health-related and to provide context-aware responses. Additionally, ensuring the interface supports multiple languages and voice commands further increased the development complexity.

5.6 Concluding Remark

In conclusion, it has shown the complete process of building the mobile health application, from setting up the tools to making everything work together. We successfully configured the software environment and connected important services like Firebase for the database and user login, AWS Rekognition for face emotion detection and ChatGPT for giving personalised advice and answering healthcare-related questions.

The machine learning models for predicting disease risk were converted to work efficiently on a mobile phone using TensorFlow Lite and ONNX formats, though it was challenging to keep them both fast and accurate. Additionally, combining all the different features such as health risk assessments, the journal and the chatbot into one smooth application was complex, but was ultimately achieved in this mobile health application.

In the end, all the main functions of the application are working well. Users can predict their chronic disease risk, understand their mental health condition, chat with the chatbot and clearly view both their physical and mental health reports displayed in line charts along with personalized advice. Although we faced some difficulties with performance and integration, however we found alternative solutions to ensure the application is helpful and easy to use. This project provides a strong foundation that can be improved in the future with more features or even better models.

Chapter 6
System Evaluation and Discussion

6.1 System Testing and Performance Metrics

6.1.1 Diabetes Machine Learning Model Performance Results

Table 6.1.1 Machine Learning Model Result for Diabetes Module

Machine Learning Model	Precision	Recall	F1 Score	Accuracy
Logistic Regression	44.16%	52.14%	42.13%	63.87%
Random Forest	43.91%	42.05%	42.65%	82.94%
XGBoost (Chosen)	**30.26%**	**84.05%**	**44.49%**	**66.95%**

We had trained and tested three different machine learning models which are Logistic Regression, Random Forest and XGBoost to see which one is best for predicting diabetes risk. Based on the results, we looked at their scores for Precision, Recall, F1-Score and Accuracy.

Based on the table, we can see that the Random Forest model had the highest Accuracy score (82.94%), meaning it was correct most often. But its Recall score was low (42.05%) which means it was not very good at finding all the people who really might have diabetes risk.

Even though the XGBoost model had lower Accuracy (66.95%) however it had a very high Recall score (84.05%). This means it is very good at finding most of the people who are actually at risk. Therefore, we chose this model because in a health application, it is more important to warn someone who might be at risk even if sometimes it is a false alarm rather than to miss someone who is actually at risk.

6.1.2 Obesity Machine Learning Model Performance Results

Table 6.1.2 Machine Learning Model Result for Obesity Module

Machine Learning Model	Precision	Recall	F1 Score	AUC-ROC	Accuracy
Logistic Regression	76.53%	77.07%	76.41%	93.36%	77.07%
Random Forest	94.73%	94.09%	94.25%	99.51%	94.09%
XGBoost (Chosen)	**95.68%**	**95.51%**	**95.55%**	**99.71%**	**95.51%**

We had trained and tested three machine learning models which are Logistic Regression, Random Forest and XGBoost to find the best one for predicting obesity risk. The models were evaluated on several metrics which are Precision, Recall, F1-Score, AUC-ROC and Accuracy.

The Logistic Regression model performed reasonably well, with accuracy and recall both around 77%. However, the Random Forest model performed much better, achieving above 94% in all metrics which is showing it is a strong and reliable model.

Ultimately, we selected the XGBoost model because it performed the best across every single metric. It achieved the highest scores in Precision (95.68%), Recall (95.51%), F1-Score (95.55%), AUC-ROC (99.71%) and Accuracy (95.51%). This means XGBoost is the most accurate and consistent model for predicting obesity risk in our application.

Therefore, we chose XGBoost for the obesity module to give users the most reliable health predictions possible.

6.1.3 Cardiovascular Disease (Heart Attack) Machine Learning Model Performance Results

Table 6.1.3

Machine Learning Model Result for Cardiovascular Disease (Heart Attack) Module

Machine Learning Model	Precision	Recall	F1 Score	AUC-ROC	Accuracy
Logistic Regression	73%	64%	68%	65.01%	63.26%
Random Forest	70%	72%	71%	65.92%	64.39%
Calibrated Ensemble (Chosen)	**70%**	**72%**	**71%**	**67.05%**	**64.02%**

We have trained and tested three machine learning models which are Logistic Regression, Random Forest and Calibrated Ensemble to predict cardiovascular disease (heart attack) risk. The models were compared using Precision, Recall, F1-Score, AUC-ROC and Accuracy.

The Logistic Regression model showed moderate performance, with Precision at 73% but recall at only 64%, meaning it was better at avoiding false alarms but missed many true at-risk cases. Next, the Random Forest model improved Recall to 72% and achieved slightly better overall balance (F1-Score: 71%) and Accuracy (64.39%).

We selected the Calibrated Ensemble model because it offered the highest AUC-ROC (67.05%), indicating better overall ability to distinguish between patients with and without heart attack risk. Even though its Accuracy (64.02%) and F1-Score (71%) were similar to Random Forest, but the higher AUC-ROC value suggests greater reliability in real-world clinical scenarios where correctly identifying true positives (actual heart attack risk) is critically important. Therefore, this model provides a balanced and trustworthy foundation for heart attack risk prediction in the application.

6.2 Testing Setup and Result

In our application we will use Black-box testing as our testing result for each module in our application. Black-box testing involves testing a module's input compared to output behavior without concern about its internal code structure. This guarantees that the application satisfies the functional specifications and reacts adequately to different user interactions. Testing includes confirming that the module satisfies all predetermined requirements while preserving data integrity and dependability, as well as validating actions like modifying or retrieving personal health data and information.

6.2.1 Login Module

Figure 6.2.1 Login Module Test Case

Test Cases	Test Detail	Test Data	Expected Result	Actual Result	Test Result
1	Sign in the application using personal email address and password	Insert email: jeremylim_18@hotmail.com password: abc123	The application authenticates from Firebase and user successfully login into the application	As expected	Pass
2	Sign in the application using Google-Sign In method	Select the gmail account: j.yeushuo18@gmail.com	The application authenticates from Google service and user successfully login into the application	As expected	Pass
3	Sign in the application using wrong personal email address	Insert email: jeremylim_19@hotmail.com password: abc123	The application blocks the user to login the application and display invaid email or password message	As expected	Pass
4	Sign in the application using wrong password	Insert email: jeremylim_18@hotmail.com password: abc456	The application blocks the user to login the application and display invaid email or password message	As expected	Pass

6.2.2 Sign Up Module

Figure 6.2.2 Sign Up Module Test Case

	Sign up module				
Test Cases	**Test Detail**	**Test Data**	**Expected Result**	**Actual Result**	**Test Result**
1	User create account and input the necessary input information	Insert full name: john lim new email:john@yahoo.com new password: Abcd123 age:23 gender: male height:171 weight 75	The application successfully created the account and store it in the Firebase and user direct login into the application	As expected	Pass
2	User does not follow the password criteria	Insert full name: john lim new email:john@yahoo.com new password: abb age:23 gender: male height:171 weight 75	The application block the user to create account and display invalid password format message which password must be at least 6 characters long	As expected	Pass
3	User empty the email input	Insert full name: john lim new email: new password: Abcd123 age:23 gender: male height:171 weight 75	The application block the user to create account and display invalid email input message which email cannot be empty	As expected	Pass
4	User empty the password input	Insert full name: john lim new email: new password: age:23 gender: male height:171 weight 75	The application block the user to create account and display invalid password input message which password cannot be empty	As expected	Pass
5	User empty the age input	Insert full name: john lim new email: new password: age: gender: male height:171 weight 75	The application block the user to create account and display invalid age input message which age cannot be empty	As expected	Pass
6	User empty the height input	Insert full name: john lim new email: new password: age: gender: male height: weight 75	The application block the user to create account and display invalid height input message which height cannot be empty	As expected	Pass
7	User empty the weight input	Insert full name: john lim email:jeremylim_18@hotmail.com password:Abbcc123 age:25 gender: male height:175 weight:	The application block the user to create account and display invalid weight input message which weight cannot be empty	As expected	Pass
8	User create the account using existing email address	Insert full name: john lim email:jeremylim_18@hotmail.com password:Abbcc123 age:25 gender: male height:175 weight:61	The application block the user to create account and display invalid email input message which found email exist in the application	As expected	Pass

6.2.3 Diabetes Module

Figure 6.2.3 Diabetes Module Test Case

Test Cases	Test Detail	Test Data	Expected Result	Actual Result	Test Result
1	User input the information with the low risk result	A 25-29 year old female with a Bachelor's degree and high income (RM20,000 or more), who has no significant medical history (e.g., no high BP, high cholesterol, or heart disease), maintains a healthy lifestyle (normal BMI, physically active, consumes fruits and vegetables, does not smoke or drink heavily), and reports excellent general, mental, and physical health.	The application machine learning model successfully predict the user in low risk	As expected	Pass
2	User input the information with the moderate risk result	A 35-39 year old male with a college diploma and moderate income, who has no critical medical history (e.g., no high BP, high cholesterol, heart disease, or stroke) but reports occasional mental and physical health concerns. He maintains several protective lifestyle factors (healthy BMI, physical activity, fruit/vegetable consumption, no smoking or heavy drinking) but has avoided seeing a doctor due to cost in the past.	The application machine learning model successfully predict the user in moderate risk	As expected	Pass
3	User input the information with the high risk result	A 45-49 year old male with high blood pressure, high cholesterol, and a slightly overweight BMI (28.0), who reports moderate mental and physical health concerns. He has no critical heart-related history but maintains a less active lifestyle with no physical activity and low fruit/vegetable consumption, despite having healthcare access.	The application machine learning model successfully predict the user in high risk	As expected	Pass
4	User empty the BMI input	A 45-49 year old male with high blood pressure, high cholesterol, and without BMI input, who reports moderate mental and physical health concerns. He has no critical heart-related history but maintains a less active lifestyle with no physical activity and low fruit/vegetable consumption, despite having healthcare access.	The application block the user to predict the diabetes risk and display invalid BMI input message which BMI value cannot be empty	As expected	Pass

6.2.4 Cardiovascular Disease (Heart Attack) Module

Figure 6.2.4 Cardiovascular Disease Module Test Case

	Cardiovascular Disease				
Test Cases	**Test Detail**	**Test Data**	**Expected Result**	**Actual Result**	**Test Result**
1	User input the information with the low risk result	A 30-year-old female with excellent vital signs: a healthy heart rate (65 bpm), normal blood pressure (110/70 mmHg), and optimal blood sugar levels (4.5 mmol/L).	The application machine learning model successfully predict the user in low risk	As expected	Pass
2	User input the information with the moderate risk result	A 55-year-old male with elevated vital signs indicating potential risk: high blood pressure (145/95 mmHg), borderline blood sugar (6.0 mmol/L), and a slightly elevated heart rate (80 bpm).	The application machine learning model successfully predict the user in moderate risk	As expected	Pass
3	User input the information with the high risk result	A 70-year-old male with critically elevated vital signs: severe hypertension (180/110 mmHg), tachycardia (95 bpm), and hyperglycemia (8.5 mmol/L blood sugar), indicating a high-risk cardiovascular and metabolic profile.	The application machine learning model successfully predict the user in high risk	As expected	Pass

6.2.5 Obesity Module

Figure 6.2.5 Obesity Module Test Case

Test Cases	Test Detail	Test Data	Expected Result	Actual Result	Test Result
1	User input the information with the normal weight result	A 25-year-old female with a healthy lifestyle—including regular physical activity, a balanced diet with frequent vegetable consumption, no smoking or alcohol, and calorie monitoring—is predicted to be in the Normal_Weight category with a BMI of ~22.0.	The application machine learning model successfully predict the user in normal weight	As expected	Pass
2	User input the information with the overweight result	A 35-year-old male with moderate-risk habits—such as frequent high-calorie food consumption, low vegetable intake, sedentary behavior, and family history—is predicted to be classified as Overweight_Level_I/II with a BMI of ~27.8.	The application machine learning model successfully predict the user in overweight	As expected	Pass
3	User input the information with the obesity result	A 45-year-old male with multiple high-risk factors—including obesity-level BMI (~41.5), sedentary lifestyle, smoking, frequent alcohol consumption, and poor dietary habits—is predicted to fall into the Obesity_Type_I/II/III category.	The application machine learning model successfully predict the user in obesity	As expected	Pass
4	User input the information with the underweight result	A 20-year-old female with notably low body weight (BMI ~15.6), infrequent meal patterns, and high physical activity levels is predicted to be classified as Insufficient_Weight.	The application machine learning model successfully predict the user in underweight	As expected	Pass

6.2.6 Chatbot Module

Figure 6.2.6 Chatbot Module Test Case

Test Cases	Test Detail	Test Data	Expected Result	Actual Result	Test Result
1	User enquries about the healthcare related questions	User ask how to reduce diabetes risk	The chatbot responses the way to reduce the diabetes risk	As expected	Pass
2	User enquries about the non-healthcare related questions	User ask who is Naruto	The chatbot responses the rejected politely message to user	As expected	Pass

6.2.7 Face Recogntion Module

Figure 6.2.7 Face Recognition Module Test Case

Test Cases	Test Detail	Test Data	Expected Result	Actual Result	Test Result
1	User smile on the camera and capture it	User smile	The AWS Face Rekogntion sucessfully analyzed the user face with happy emotion	As expected	Pass
2	User makes an angry expression on the camera and capture it	User angry face	The AWS Face Recognition successfully analyzed the user face with ANGRY emotion	As expected	Pass
3	User maintains a neutral expression on the camera and capture it	User neutral face	The AWS Face Recognition successfully analyzed the user face with CALM emotion	As expected	Pass
4	User makes a confused expression on the camera and capture it	User confused face	The AWS Face Recognition successfully analyzed the user face with CONFUSED emotion	As expected	Pass
5	User makes a disgusted expression on the camera and capture it	User disgusted face	The AWS Face Recognition successfully analyzed the user face with DISGUSTED emotion	As expected	Pass
6	User makes a fearful expression on the camera and capture it	User fearful face	The AWS Face Recognition successfully analyzed the user face with FEAR emotion	As expected	Pass
7	User makes a sad expression on the camera and capture it	User sad face	The AWS Face Recognition successfully analyzed the user face with SAD emotion	As expected	Pass
8	User makes a surprised expression on the camera and capture it	User surprised face	The AWS Face Recognition successfully analyzed the user face with SURPRISED emotion	As expected	Pass

.

6.2.8 Journal Prompt Module

Figure 6.2.8 Journal Prompt Module Test Case

Test Cases	Test Detail	Test Data	Expected Result	Actual Result	Test Result
1	User submits a journal entry with positive sentiment	"I am thrilled about my new job! I had a wonderful first day and my team is incredibly supportive. The future looks bright!"	The system successfully analyzes the journal entry and returns a POSITIVE sentiment result.	As expected	Pass
2	User submits a journal entry with negative sentiment	"I feel completely defeated today. Nothing went right, and I'm overwhelmed with all my responsibilities. I just want to be alone."	The system successfully analyzes the journal entry and returns a NEGATIVE sentiment result.	As expected	Pass
3	User submits a journal entry with neutral sentiment	"This is a log of my tasks. I went to the store to buy groceries. The meeting is scheduled for 3 PM tomorrow."	The system successfully analyzes the journal entry and returns a NEUTRAL sentiment result.	As expected	Pass
4	User submits a complex entry with mixed but ultimately positive words	"The project was challenging and stressful at times, but we persevered as a team and the final result was a huge success. I'm proud of what we accomplished."	The system successfully analyzes the journal entry and returns a POSITIVE sentiment result.	As expected	Pass
5	User submits an empty entry	""	The system successfully analyzes the journal entry and returns a NEUTRAL sentiment result.	As expected	Pass
6	User submits a very short, ambiguous entry	"Okay."	The AWS Face Recognition successfully analyzed the user face with FEAR emotion	As expected	Pass

6.2.9 Mental Health Assessment Module

Figure 6.2.9 Mental Health Assessment Module Test Case

Test Cases	Test Detail	Test Data	Expected Result	Actual Result	Test Result
1	Minimal Depression - User reports no significant symptoms	[0, 0, 0, 0, 0, 0, 0, 0, 0]	Total Score: 0. Severity: Minimal depression.	As expected	Pass
2	Mild Depression - User reports minor symptoms on several days	[1, 1, 2, 1, 0, 1, 2, 0, 1] (Total = 9)	Total Score: 9. Severity: Mild depression.	As expected	Pass
3	Moderate Depression - User reports symptoms more than half the days	[2, 2, 1, 2, 2, 2, 1, 2, 1] (Total = 15)	Total Score: 15. Severity: Moderate depression.	As expected	Pass
4	Moderately Severe Depression - User reports significant symptoms nearly every day	[3, 2, 3, 3, 2, 3, 2, 3, 1] (Total = 22)	Total Score: 22. Severity: Moderately severe depression.	As expected	Pass
5	Severe Depression - User reports severe symptoms nearly every day	[3, 3, 3, 3, 3, 3, 3, 3, 3] (Total = 27)	Total Score: 27. Severity: Severe depression.	As expected	Pass
6	Functional Impairment Logic - User scores high but reports no functional impairment	Answers: [3, 3, 3, 3, 3, 3, 3, 3, 3] + "No difficulty at all" on functional question.	Total Score: 27. Severity: Severe depression. (The functional question is for clinical context but does not change the PHQ-9 score).	As expected	Pass
7	Functional Impairment Logic - User scores low but reports significant functional impairment	Answers: [1, 0, 1, 0, 1, 0, 1, 0, 0] (Total=4) + "Very difficult" on functional question.	Total Score: 4. Severity: Minimal depression. (The score is based on the 9 items; functional impact is a separate data point).	As expected	Pass
8	Boundary Test: Incomplete Form - User submits the form without answering all questions	Answers: [0, 1, , 2, 0, , 1, 0, 1] (Questions 3 & 6 are missing)	The system displays a validation error message: "Please answer all questions before submitting." The score is not calculated.	As expected	Pass
9	Boundary Test: Score Calculation - Mixed frequencies for precise addition	Answers: [0, 1, 3, 2, 1, 0, 2, 3, 0] (Total = 0+1+3+2+1+0+2+3+0 = 12)	Total Score: 12. Severity: Moderate depression.	As expected	Pass

6.2.10 Health Report Module

Figure 6.2.10 Health Report Module Test Case

Test Cases	Test Detail	Test Data	Expected Result	Actual Result	Test Result
1	Mental Health -> PHQ-9 -> Weekly View	1. Select Category: Mental Health 2. Select Type: Mental Health Assessment (PHQ-9) 3. Select Time Frame: Weekly *User has 4 PHQ-9 scores from the last 3 weeks.*	A line chart loads showing 3-4 data points (one for each assessment) on the X-axis (date) and the score (0-27) on the Y-axis. The chart title reflects the filters. A text-based recommendation is shown (e.g., "Your symptoms have remained stable. Consider checking in weekly.").	As expected	Pass
2	Mental Health -> Journal Sentiment -> Monthly View	1. Select Category: Mental Health 2. Select Type: Journal Analysis 3. Select Time Frame: Monthly User has 20 journal entries from the last month.	A line chart loads showing a trend line for sentiment over the month (e.g., from Negative to Positive). The chart may aggregate data by week or show all entries. A recommendation is shown based on the overall trend (e.g., "Your overall mood has improved this month!").	As expected	Pass

3	Mental Health -> Face Recognition -> Weekly View	1. Select Category: Mental Health 2. Select Type: Face Recognition Analysis 3. Select Time Frame: Weekly User has 5 facial analysis results from the last week.	A line chart loads showing the trend of a specific emotion (e.g., "HAPPY" confidence score) over the week. A recommendation is shown (e.g., "You've been showing more positive expressions towards the end of the week.").	As expected	Pass
4	Chronic Disease -> Diabetes Risk -> Monthly View	1. Select Category: Chronic Disease Risk 2. Select Type: Diabetes 3. Select Time Frame: Monthly User has logged blood sugar readings 3 times a week for a month.	A line chart loads showing the trend of blood sugar levels (or a diabetes risk score) over the month. The Y-axis shows the relevant metric. A recommendation is shown (e.g., "Your levels are within normal range. Maintain your current diet.").	As expected	Pass
5	Chronic Disease -> Obesity Risk -> Weekly View	1. Select Category: Chronic Disease Risk 2. Select Type: Obesity 3. Select Time Frame: Weekly User has logged weight and meals daily for a week.	A line chart loads showing a trend like BMI or calorie intake over the week. A recommendation is shown (e.g., "Your calorie intake was lower this week. Great job!").	As expected	Pass
6	Chronic Disease -> Heart Attack Risk -> Monthly View	1. Select Category: Chronic Disease Risk 2. Select Type: Heart Attack 3. Select Time Frame: Monthly User has logged blood pressure and activity data throughout the month.	A line chart loads showing a trend like resting heart rate or a calculated risk score over the month. A recommendation is shown (e.g., "Your cardiovascular fitness is improving. Keep up the regular exercise!").	As expected	Pass
7	No Data for Selected Filter	1. Select Category: Mental Health 2. Select Type: Mental Health Assessment 3. Select Time Frame: Weekly *User has no PHQ-9 assessments for the last week.*	The line chart area displays a message: "No data available for the selected filters." The recommendations section shows a generic message (e.g., "Unable to load your recommendations").	As expected	Pass

6.2.11 Profile Activity Module

Figure 6.2.11 Profile Activity Module Test Case

Test Cases	Test Detail	Test Data	Expected Result	Actual Resul	Test Result
1	Successful Update - Valid Data	Age: 30, Gender: Male, Height: 175, Weight: 68	The system saves the new information. A success message is displayed.	As expected	Pass
2	Validation Error - Invalid Age (Negative)	Age: -5 (Other fields valid)	The system does not save. Error message: "Please enter a valid age."	As expected	Pass
3	Validation Error - Invalid Weight (Zero)	Weight: 0 kg (Other fields valid)	The system does not save. Error message: "Please enter a valid weight."	As expected	Pass
4	Validation Error - Required Field Empty	Age: (blank) (Other fields valid)	The system does not save. Error message: "Age is required."	As expected	Pass
5	Partial Update	update Weight to 70 kg. Leave other fields uncha	The system saves only the new Weight. Other values remain unchanged.	As expected	Pass
6	Data Persistence	Update profile, log out, log back in	The updated profile information is successfully retrieved and displayed.	As expected	Pass

6.3 Project Challenges

A major challenge in this application was balancing the need for accurate predictions with the limited processing power of a mobile device. Implementing on-device machine learning for the chronic disease risk required us to convert models into efficient formats like TensorFlow Lite and ONNX model and use techniques like quantization. While this was essential for making the application feasible on Android hardware but it introduced a trade-off where some level of prediction accuracy was potentially sacrificed to achieve the necessary speed and avoid excessive memory use which was a key limitation of the chosen approach.

Next, managing three separate AI pipelines for chronic disease, journal sentiment and facial emotion detection also added significant complexity to the system. A major concern was ensuring that running these models did not lead to high battery consumption or slow down the device especially if multiple processes were running at once. Processing data locally was a priority for user privacy but optimizing the resource allocation for each unique model to maintain a smooth user experience was a considerable challenge.

Furthermore, creating a single cohesive system that could intelligently combine quantitative health data, mood analysis from journals and face recognition and natural-language conversation from ChatGPT was another difficult task. The application required sophisticated logic to decide when to use each data source and how to provide a context-aware response to the user. Ensuring that all these different features worked together seamlessly within one interface while also supporting multiple languages which greatly increased the development complexity and represents a current limitation on the depth of analysis the system can perform.

Finally, a necessary and important challenge was designing a system that makes automated health suggestions without being a "black box." It is difficult to transparently show the user exactly how a specific recommendation was generated from their combined data, which could impact how much they trust the advice. While the application successfully displays trends in line charts and provides general advice which fully explaining the reasoning behind its composite recommendations remains a complex challenge for the current system.

6.4 Objectives Evaluation

Based on the first objective of this project which is to develop a machine learning model to diagnose the risk of chronic diseases and create awareness for users, this objective was successfully achieved. Machine learning models for predicting the risk of cardiovascular disease, obesity, and diabetes were developed and implemented within the Android application. The models were converted into TensorFlow Lite and ONNX format to enable on-device processing, allowing users to input their health data and receive an immediate risk assessment without an internet connection. This functionality provides personalized insights and early warnings that used ChatGPT to generated which directly addresses the project's aim of enabling users to take preventive action and proactively manage their health by identifying potential issues before they become more severe.

Next, the second objective which is to create a personalized chatbot to increase user engagement with the mobile health application, this objective was successfully met through the integration of a ChatGPT-based chatbot. The chatbot serves as a "personal health assistant" within the application, addressing user inquiries related to healthcare challenges and preventive measures for maintaining a healthy lifestyle. It was designed to provide personalised responses by analysing user input and patterns from other application features, including chronic disease risk data, journal entries and emotion analysis. Furthermore, the chatbot maintains conversation history to encourage continuous engagement, thereby effectively supporting both the mental and physical health of the user.

Furthermore, the third objective which is to detect user emotions by adopting various analytical tools to assess mental health and promote an overall healthy lifestyle that this objective was fulfilled through the implementation of a multi-modal approach to emotion detection. The application integrates three distinct analytical tools which are a text sentiment analysis feature for the journal module, a standardized PHQ-9 mental health assessment and facial emotion recognition via the AWS Rekognition API using the device's camera. By combining these methods, the system can accurately assess a user's emotional state and provide personalized advice based on real-time mood tracking. This comprehensive strategy helps users manage their emotional health, promotes self-awareness, and encourages effective strategies to improve their wellbeing.

Moreover, the fourth objective to develop a culturally adaptive, user-friendly Android application with a user-friendly interface and voice navigation by enhancing accessibility for diverse Malaysian users that this objective was achieved through the careful design and development of an Android application that prioritizes accessibility and cultural adaptation. The application features a user-friendly interface that presents personalized health reports based on both chronic disease risk assessments and mental health insights. To enhance accessibility for diverse Malaysian users, the application includes voice navigation capabilities and provides support in bilingual languages. These features were specifically designed to reduce language barriers and ensure a positive and engaging user experience for different communities across Malaysia.

In conclusion, all four primary objectives of the project have been successfully met and integrated into a single and cohesive mobile application. The core functionalities for chronic disease risk prediction, mental health assessment, personalized chatbot assistance, and culturally adaptive design are fully operational in the final product.

6.5 Concluding Remark

In conclusion, this chapter has provided a thorough evaluation of the complete mobile health application system. The system testing and performance metrics have shown that the chosen machine learning models while sometimes requiring a trade-off between different metrics are fit for their purpose. The XGBoost model for diabetes was selected for its high recall to ensure most at-risk users are warned, while the XGBoost model for obesity and the Calibrated Ensemble model for cardiovascular disease were chosen for their strong overall performance and high AUC-ROC scores which providing reliable predictions for users.

Furthermore, the comprehensive black-box testing conducted on all application modules, from login and sign-up to the complex health report generator has confirmed that all functional requirements have been successfully met. Each module reacts correctly to user input and produces the expected output which ensuring the application is reliable and stable for end-users.

Although the project faced significant challenges, particularly in balancing on-device performance with model accuracy and managing multiple AI pipelines, these were overcome through technical solutions like model quantization and efficient resource management. Finally, the evaluation confirms that all four primary project objectives have been fully achieved. The application successfully delivers a functional, integrated and user-friendly platform for chronic disease risk assessment, mental health analysis, and personalized health assistance, providing a strong foundation that can be built upon with future enhancements.

Chapter 7
Conclusion and Recommendations

7.1 Conclusion

This project was developed to overcome the shortcomings of existing mobile health applications, specifically the lack of integrated early warning systems for chronic diseases, low user engagement, and insufficient focus on mental well-being. Motivated by the high prevalence of diabetes, obesity, and heart disease in Malaysia, this project aimed to create a unified application that addresses both physical and mental health proactively.

To solve these problems, a comprehensive Android application was successfully built that fulfilling all initial objectives through its core modules. Firstly, the objective of chronic disease risk diagnosis was achieved by developing and integrating three on-device machine learning models for diabetes, obesity and cardiovascular disease. These models analyze user-input health data to provide an immediate risk assessment categorized as “low,” “moderate,” or “high,” along with personalized advice, fulfilling the goal of creating awareness and enabling preventive action.

Next, the engagement objective was met by implementing a ChatGPT-powered chatbot. This module acts as a personal health assistant, answering user queries and offering context-aware support based on the user's own health data from other modules, thereby encouraging continuous interaction.

Furthermore, the objective of detecting user emotions to assess mental health was accomplished through a multi-modal approach. This was realized by integrating three specific modules which are a journal prompt with text sentiment analysis, a standardized PHQ-9 mental health assessment and facial-expression recognition via AWS Rekognition. This combination allows users to track their emotional state through multiple methods and receive personalised recommendations.

Moreover, the objective of a culturally adaptive application was fulfilled by developing a user-friendly Android interface that includes bilingual support language and voice navigation. The health report module synchronize data from all other features into clear visual charts and summaries, making complex health information accessible and understandable for a diverse user base.

Lastly, the novelty of this project lies in the seamless integration of these four pillars which are chronic disease prediction, an AI chatbot, multi-modal emotion detection, and an adaptive interface into a single, privacy-focused application. By processing all data on the device, the application ensures security and provides instant results without an internet connection. This holistic approach, which combines physical and mental health tracking with personalized, accessible support in one application was successfully addresses the gaps identified in current mobile health solutions and provides a strong foundation for future development.

7.2 Recommendations

Based on this mobile health application project, there are several recommendations are proposed to enhance the application's capabilities, accuracy, and user experience in the future.

Firstly, the performance of the chronic disease prediction models especially for diabetes and cardiovascular disease could be significantly improved by training them on a larger and more localized dataset. Future work should focus on collecting more health data from Malaysian users to retrain and validate the models. This would help increase the accuracy, precision and reliability of the risk predictions by making the application a more trusted tool for users.

Moreover, the application currently focuses on diabetes, obesity, and cardiovascular disease. In the future, the application can be extended to include predictions for other common chronic conditions in Malaysia, such as hypertension, chronic kidney disease or high cholesterol. This would make the application a more comprehensive health assessment tool for users.

Next, the current ChatGPT integration provides general health advice, future development could focus on making the chatbot more deeply integrated with the user's data. For example, the chatbot could proactively give advice based on a user's latest health report or remind them to check their risk if they haven't done in a while. This would make the chatbot feel more like a true personal health assistant.

Furthermore, to further increase user engagement, the application could include a section for short educational articles or video tips about managing chronic diseases and improving mental health. Adding features like setting health goals, tracking progress, and earning achievements could also motivate users to maintain a healthy lifestyle and use the application regularly.

In conclusion, these recommendations provide a clear roadmap for enhancing the application's functionality, accuracy and user engagement. By focusing on these areas, future developers can build upon this project's strong foundation to create an even more powerful and effective mobile health tool.

REFERENCES

[1] A. Ahmad, M. R. Shahril, N. Wan-Arfah, W. A. Mohd Abu Bakar, C. Piernas, and P. L. Lua, “Changes in health-related lifestyles and food insecurity and its association with quality of life during the COVID-19 lockdown in Malaysia,” *BMC Public Health*, vol. 22, no. 1, Jun. 2022, doi: https://doi.org/10.1186/s12889-022-13568-0.

[2] Norbaidurah Ithnain *et al.*, “Lifestyle and preventive behaviors among adults during the early phase of the COVID-19 movement control order (MCO) in Malaysia.,” *PubMed*, vol. 12, pp. 174–174, Jan. 2023, doi: https://doi.org/10.4103/jehp.jehp_1521_22.

[3] F. Mohd and N. I. Elanie Mustafah, "“Hello, Dr”: A Heatlhcare Mobile Application," 2021 4th International Symposium on Agents, Multi-Agent Systems and Robotics (ISAMSR), Batu Pahat, Malaysia, 2021, pp. 20-23, doi: 10.1109/ISAMSR53229.2021.9567764. keywords: {Heating systems; COVID-19;Pandemics;Hospitals;Mobile applications;Robots;Monitoring;smart mobile healthcare;mobile application;self-diagnose},

[4] L. A. Lima, M. D. M. Danilo, C. B. Vaz and A. I. Pereira, "Using m-health apps in oncology : A review from 2015 to 2022," 2023 18th Iberian Conference on Information Systems and Technologies (CISTI), Aveiro, Portugal, 2023, pp. 1-6, doi: 10.23919/CISTI58278.2023.10211467. keywords: {Pandemics;Hospitals;Databases;Collaboration;Oncology;Mobile applications;Prognostics and health management;m-health;mobile apps;oncology;cancer patients},

[5] R. N. Parvin and M. Shahjahan, "Knowledge, attitude and practice on eHealth among doctors working at selected private hospitals in Dhaka, Bangladesh," J. Int. Soc. Telemed. eHealth, vol. 4, no. 1, pp. e15-1, 2016.

[6] S. Ryu, "Book review: MHealth: New horizons for health through mobile technologies: Based on the findings of the second global survey on eHealth

(Global Observatory for eHealth series, volume 3)," Healthc. Inform. Res., vol. 18, no. 3, pp. 231, 2012.

[7] M. N. K. Boulos, A. C. Brewer, C. Karimkhani, D. B. Buller, and R. P. Dellavalle, "Mobile medical and health apps: State of the art, concerns, regulatory control and certification," Online J. Public Health Inform., vol. 5, no. 3, p. 229, 2014.

[8] V. P. Chandran, A. Balakrishnan, M. Rashid, G. PaiKulyadi, S. Khan, E. S. Devi, et al., "Mobile applications in medical education: a systematic review and meta-analysis," PLoS One, vol. 17, no. 3, p. e0265927, 2022.

[9] C. Quant, L. Altieri, J. Torres, and N. Craft, "The self-perception and usage of medical apps amongst medical students in the United States: A cross-sectional survey," Int. J. Telemed. Appl., vol. 2016, p. 3929741, 2016.

[10] C. L. Ventola, "Mobile devices and apps for health care professionals: Uses and benefits," P T, vol. 39, no. 5, pp. 356–364, 2014.

[11] J. V. J. Jembai *et al.*, “Mobile health applications: awareness, attitudes, and practices among medical students in Malaysia,” *BMC Medical Education*, vol. 22, no. 1, Jul. 2022, doi: https://doi.org/10.1186/s12909-022-03603-4.

[12] M. Mutz, J. Müller, and A. K. Reimers, "Use of digital media for home-based sports activities during the COVID-19 pandemic: results from the German SPOVID survey," Int. J. Environ. Res. Public Health, vol. 18, no. 9, p. 4409, 2021.

[13] D. A. Kinney, L. A. Nabors, A. L. Merianos, and R. A. Vidourek, "College students’ use and perceptions of wearable fitness trackers," Am. J. Health Educ., vol. 50, no. 5, pp. 298–307, 2019.

[14] N. Khan, M. I. Qureshi, I. Mustapha, S. Irum, and R. N. Arshad, “A Systematic Literature Review Paper on Online Medical Mobile Applications in

Malaysia,” *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 16, no. 01, p. 63, Jan. 2020, doi: https://doi.org/10.3991/ijoe.v16i01.12263.

[15] C. Qiang, M. Yamamichi, V. Hausman, R. Miller, and D. Altman, "Mobile applications for the health sector," World Bank, 2012. [Online]. Available:https://documents1.worldbank.org/curated/en/751411468157784302/pdf/726040WP0Box370th0report00Apr020120.pdf. [Accessed: Dec. 1, 2024].

[16] A. Firdaus, N. A. A. Aziz, W. A. W. Asyraf, and A. Fairuz, "Explorative Survey on the Usage and Needs of Mobile Health Applications (mHealth) amongst Caregivers in Taking Care of Stroke Survivors," Med. Health, vol. 17, no. 2, pp. 85-100, Dec. 2022. Available: https://doi.org/10.17576/MH.2022.1702.07.

[17] H. Mohd, I. Fatihah, None Norsuriani Samsudin, A. Halim, M. Valli, and N. Mohd, “Predicting the Intention to Use Mobile Health Applications Among Young Adults in Malaysia,” *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 37, no. 2, pp. 48–57, Jan. 2024, doi: https://doi.org/10.37934/araset.37.2.4857.

[18] J. Y. Lee, C. P. Wong, and S. W. H. Lee, "m-Health views and perception among Malaysians: findings from a survey among individuals living in Selangor," Mhealth, vol. 6, p. 6, 2020. PubMed PMID: 32190617. [Online]. Available: https://doi.org/10.21037/mhealth.2019.09.16.

[19] B. Kc *et al.*, “The Types and Pattern of Use of Mobile Health Applications Among the General Population: A Cross-Sectional Study from Selangor, Malaysia,” *Patient Preference and Adherence*, vol. 15, pp. 1755–1762, Aug. 2021, doi: https://doi.org/10.2147/PPA.S325851.

[20] World Health Organization, “Malaysia data | World Health Organization,” *data.who.int*, 2023. https://data.who.int/countries/458

[21] A. S. B. A. Bakar, H. A. Rahman, A. I. N. M. Nazan, and H. K. Shahar, "Understanding the Determinant Factors of Chronic Disease Behaviour among Malaysian Low-Income Workers: An Integrated Model Approach," Int. J. Acad. Res. Bus. Soc. Sci., vol. 12, no. 14, pp. 255-270, 2022.

[22] A. Triantafyllidis, H. Kondylakis, K. Votis, D. Tzovaras, N. Maglaveras, and K. Rahimi, “Features, outcomes, and challenges in mobile health interventions for patients living with chronic diseases: A review of systematic reviews,” *International Journal of Medical Informatics*, vol. 132, p. 103984, Dec. 2019, doi: https://doi.org/10.1016/j.ijmedinf.2019.103984.

[23] H. Kharrazi, R. Chisholm, D. VanNasdale, and B. Thompson, “Mobile personal health records: An evaluation of features and functionality,” *International Journal of Medical Informatics*, vol. 81, no. 9, pp. 579–593, Sep. 2012, doi: https://doi.org/10.1016/j.ijmedinf.2012.04.007.

[24] P. Krebs and D. T. Duncan, “Health App Use Among US Mobile Phone Owners: A National Survey,” *JMIR mHealth and uHealth*, vol. 3, no. 4, p. e101, Nov. 2015, doi: https://doi.org/10.2196/mhealth.4924.

[25] L. Zhou, J. Bao, I. M. A. Setiawan, A. Saptono, and B. Parmanto, “The mHealth App Usability Questionnaire (MAUQ): Development and Validation Study,” *JMIR mHealth and uHealth*, vol. 7, no. 4, p. e11500, Apr. 2019, doi: https://doi.org/10.2196/11500.

[26] Zhang Yanjun and R. A. Santos, “Understanding User Engagement Assessment and Modernization in a Chronic Disease Management App on A-B-C Platform,” *The QUEST Journal of Multidisciplinary Research and Development*, vol. 2, no. 3, Dec. 2023, doi: https://doi.org/10.60008/thequest.v2i3.160.

[27] J. Knitza *et al.*, “Mobile Health Usage, Preferences, Barriers, and eHealth Literacy in Rheumatology: Patient Survey Study,” *JMIR mHealth and uHealth*, vol. 8, no. 8, p. e19661, Aug. 2020, doi: https://doi.org/10.2196/19661.

[28] "Healthcare | Free Full-Text | Influencing Factors of Acceptance and Use Behavior of Mobile Health Application Users: Systematic Review," *Mdpi.com*, 2020. https://www.mdpi.com/2227-9032/9/3/357/review_report (accessed Dec. 09, 2024).

[29] J. Sarris *et al.*, "Multiple lifestyle factors and depressed mood: a cross-sectional and longitudinal analysis of the UK Biobank (N = 84,860)," *BMC Medicine*, vol. 18, no. 1, Nov. 2020, doi: https://doi.org/10.1186/s12916-020-01813-5.

[30] J. Melzner, J. Heinze, and T. Fritsch, "Mobile Health Applications in Workplace Health Promotion: An Integrated Conceptual Adoption Framework," *Procedia Technology*, vol. 16, pp. 1374–1382, 2014, doi: https://doi.org/10.1016/j.protcy.2014.10.155.

[31] National strategic plan for non-communicable diseases, https://www.moh.gov.my/moh/resources/Penerbitan/Rujukan/NCD/National%20Strategic%20Plan/FINAL_NSPNCD.pdf (accessed Mar. 11, 2025).

[32] Fitbit LLC, "Fitbit," *Google.com*, 2021. https://play.google.com/store/apps/details?id=com.fitbit.FitbitMobile&hl=en&pli=1 (accessed Dec. 09, 2024).

[33] G. L. or its subsidiaries, "Garmin Connect Mobile | Apps | Garmin Malaysia," *Garmin Malysia*. https://www.garmin.com.my/products/apps/garmin-connect-mobile/

[34] S. C. P and S. N. Francis, "EaseIt: An Android Based Health Monitoring Mobile Application," vol. 6, pp. 865–872, Jun. 2023, doi: https://doi.org/10.1109/icpcsn58827.2023.00148.

[35] Janak Limbachia, Y. Damani, S. Dave, and Vidya Sagvekar, "MOODIFY: Tailored, Personal and Multifaceted AI Assistant for Young Adult Mental Health Issues," Dec. 2023, doi: https://doi.org/10.1109/icast59062.2023.10455044.

[36] I. Youper, “Youper: AI Mental Health,” *App Store*, Jan. 11, 2016. https://apps.apple.com/us/app/youper-ai-mental-health/id1060691513 (accessed Apr. 29, 2025).

[37] Touchkin, “Wysa: Anxiety, therapy chatbot,” *Google.com*, 2021. https://play.google.com/store/apps/details?id=bot.touchkin&hl=en (accessed Apr. 29, 2025).

[38] “Woebot: The Mental Health Ally,” *App Store*, Jan. 16, 2024. https://apps.apple.com/us/app/woebot-the-mental-health-ally/id1305375832 (accessed Apr. 29, 2025).

[39] Moodfit, “Moodfit | Tools & Insight for Your Mental Health,” *moodfit*. https://www.getmoodfit.com/ (accessed Apr. 29, 2025).

[40] “mySugr US - Make Diabetes Suck Less,” *www.mysugr.com*. https://www.mysugr.com/en (accessed Apr. 29, 2025).

[41] MyTherapy, “Medication Reminder and Pill Tracker App MyTherapy,” *Mytherapyapp.com*, 2020. https://www.mytherapyapp.com/ (accessed Apr. 29, 2025).

[42] A. Hussain, A. Mateen, F. Amin, M. Abid, and S. Ullah, "Health Monitoring Apps: An Evaluation of the Persuasive System Design Model for Human Wellbeing," Information, vol. 14, no. 7, p. 412, 2023. doi: 10.3390/info14070412.

[43] G. Vertola *et al.*, “Use and Effectiveness of Mobile Health Applications for Stress Management and Emotional Self-Regulation in Adult Workers: A Systematic Review,” *IEEE Xplore*, Sep. 01, 2022. https://ieeexplore.ieee.org/document/10081617

[44] Centers for Disease Control and Prevention, “Adult BMI categories,” *CDC*, Mar. 19, 2024. https://www.cdc.gov/bmi/adult-calculator/bmi- categories.html (accessed Apr. 29, 2025).

[45] National HIV Curriculum, “Patient Health Questionnaire-9 (PHQ-9) - Mental Disorders Screening - National HIV Curriculum,” *National HIV Curriculum*, 2025. https://www.hiv.uw.edu/page/mental-health-screening/phq-9

[46] Nasim, Nadire Cavus, P. Esmaili, Boran Sekeroglu, and Süleyman Aşır, “Detecting emotions through EEG signals based on modified convolutional fuzzy neural network,” *Scientific reports*, vol. 14, no. 1, May 2024, doi: https://doi.org/10.1038/s41598-024-60977-9.

[47] M. E. Shipe, S. A. Deppen, F. Farjah, and E. L. Grogan, “Developing prediction models for clinical use using logistic regression: an overview,” *Journal of Thoracic Disease*, vol . 11, no. S4, pp. S574–S584, Mar. 2019, doi: https://doi.org/10.21037/jtd.2019.01.25.

[48] F. M. Delpino, Â. K. Costa, S. R. Farias, A. D. P. Chiavegatto Filho, R. A. Arcêncio, and B. P. Nunes, “Machine learning for predicting chronic diseases: a systematic review,” *Public Health*, vol. 205, pp. 14–25, Apr. 2022, doi: https://doi.org/10.1016/j.puhe.2022.01.007.

[49] T. Chen and C. Guestrin, “XGBoost: a Scalable Tree Boosting System,” *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’16*, vol. 1, no. 1, pp. 785–794, Aug. 2016, doi: https://doi.org/10.1145/2939672.2939785.

[50] S. Fan *et al.*, “Applying probability calibration to ensemble methods to predict 2-year mortality in patients with DLBCL,” *BMC Medical Informatics and Decision Making*, vol. 21, no. 1, Jan. 2021, doi: https://doi.org/10.1186/s12911-020-01354-0.

[51] “Diabetes Health Indicators Dataset,” *www.kaggle.com*. https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset/data

[52] Z. Xie, O. Nikolayeva, J. Luo, and D. Li, “Building Risk Prediction Models for Type 2 Diabetes Using Machine Learning Techniques,” *Preventing Chronic Disease*, vol. 16, Sep. 2019, doi: https://doi.org/10.5888/pcd16.190109.

[53] T. A. Rashid and B. Hassan, "Heart Attack Dataset," *data.mendeley.com*, vol. 1, Nov. 2022, doi: https://doi.org/10.17632/wmhctcrt5v.1.

[54] RK, "Obesity Prediction," *Kaggle.com*, 2025. https://www.kaggle.com/datasets/ruchikakumbhar/obesity-prediction

[55] F. M. Palechor and A. de la H. Manotas, "Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from Colombia, Peru and Mexico," *Data in Brief*, vol. 25, p. 104344, Aug. 2019, doi: https://doi.org/10.1016/j.dib.2019.104344.

.

POSTER

FACULTY OF INFORMATION COMMUNICATION AND TECHNOLOGY

PERSONAL MOBILE-HEALTH APPLICATION

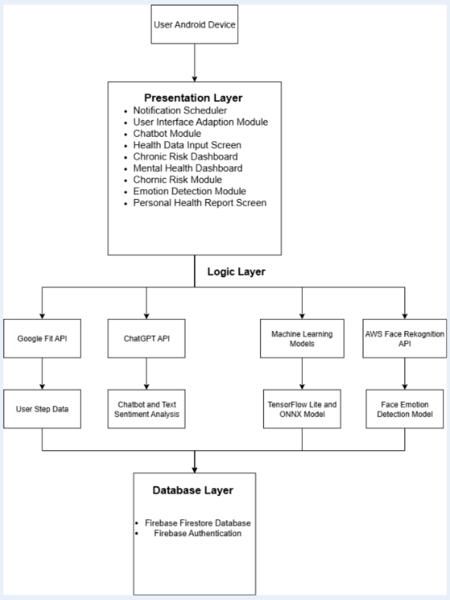
INTRODUCTION

THIS MOBILE HEALTH APP IS VERY DIFFERENT FROM OTHER WHICH IT CAN PREDICTS DIABETES, CARDIOVASCULAR DISEASE AND OBESITY RISKS, OFFERS A CHATBOT HEALTH ASSISTANT, DETECTS YOUR MOOD FROM TEXT, SELFIE AND ADAPTS TO YOUR LANGUAGE. THE USER CAN NOW TAKE CARE FOR BOTH BODY AND MENTAL HEALTH IN JUST THIS SIMPLE APP.

APPLICATION FEATURES

1. Use AI to predict diabetes, cardiovascular disease and obesity risks
2. Offer a personalized chatbot health assistant
3. Detect user emotions (text, assessment and selfie) for mental-health support
4. Build an adaptive, multilingual Android UI with voice navigation

DIAGRAM



DISCUSSIONS

1. The app interface will adjusts to your device and sends gentle reminders to keep user active.
2. It secure login to protects user data and let user sync health info from trusted sources.
3. Built-in AI on user phone instantly assesses chronic-disease risks and shows clear with color-coded warnings.
4. A smart chatbot answers user health questions while simple text, assessment and selfie checks track user mood and offer personalized advice for user.

CONCLUSION

This app combines MLrisk checks, a health chatbot and mood tracking in one tool. It gives color-coded warnings, personalized advice and mood insights via text, assessment or selfie and provide convenient to user with just easy voice and language controls to help user manage physical and mental health every day.