WITHME ALARM APP

HANA FATIHA BINTI HADZRI
CD20135

BACHELOR OF COMPUTER SCIENCE (GRAPHICS & MULTIMEDIA TECHNOLOGY) WITH HONOURS

FACULTY OF COMPUTING (FK)

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : HANA FATIHA BINTI HADZRI

Date of Birth

Title : WITHME ALARM APP

Academic Session : 2022/2023 (SEMESTER II)

I declare that this thesis is classified as:

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

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

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

Certified by:

_____ (Student's Signature)

_____ (Supervisor's Signature)

SR. DR. NGAHZAIFA AB GHANI
Date: 25 JULY 2023

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the Bachelor of Computer Science (Graphics and Multimedia Technology) with Honours.

_____

(Supervisor's Signature)

Full Name : SR. DR. NGAHZAIFA AB GHANI

Position : SENIOR LECTURER

Date : 25 JULY 2023

STUDENT'S DECLARATION

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

_____

(Student's Signature)

Full Name : HANA FATIHA BINTI HADZRI

ID Number : CD20135

Date : 30 June 2023

WITHME ALARM APP

HANA FATIHA BINTI HADZRI

Thesis submitted in fulfillment of the requirements
for the award of
Bachelor of Computer Science (Graphics and Multimedia Technology)
with Honours

Faculty of Computing
UNIVERSITI MALAYSIA PAHANG

JULY 2023

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my supervisor, Sr. Dr. Ngahzaifa Ab Ghani, for her invaluable guidance and continuous encouragement throughout the completion of this project. Despite the fact that the field of this project is not directly related to her research group, she displayed remarkable patience and made efforts to assist me in organizing my thoughts and addressing concerns during the project development. I would also like to express my sincere gratitude to her for dedicating her time and sacrificing her days off to proofread my humble thesis.

I would like to extend my heartfelt appreciation to my resourceful and helpful friends, Nur Hafizah binti Mohd Nazam and Nurin Iman binti Mohammad Azmi. They generously allowed me to borrow their personal devices, such as smartphones and tablets, to facilitate the testing process of my app development for my final year project. I am especially grateful to Nur Atiqah binti Kamal and Noraisyah Asyikin binti Mat Lizan@Hasan for their involvement in the user testing process, which greatly contributed to my understanding of the project's development capabilities and limitations. I am truly fortunate to have such exceptional individuals as friends, and I sincerely appreciate their invaluable assistance and encouragement.

Finally, I am immensely grateful for the unwavering prayers and support of my parents. As the last child in the family, I carry the responsibility to bring pride and happiness by completing my bachelor's degree. Though I may not express it often, their health and well-being are always in my prayers. Their presence and belief in me provide the strength I need to overcome the challenges of this project. I am truly blessed to have such loving parents, and I am determined to make them proud.

ABSTRAK

Bagi memastikan keselamatan individu yang tersayang secara tradisional melibatkan fokus dalam pemerhatian atau bergenggaman tangan. Walau bagaimanapun, kejadian kehilangan individu yang tersayang masih berlaku, terutamanya di kalangan ibu bapa dan penjaga warga emas yang berkeperluan khas. Selain itu, di kawasan sesak seperti Bukit Bintang, Kuala Lumpur, Times Square, New York, atau Myeong-dong, Seoul, telah menjadi keperluan untuk sentiasa berwaspada dalam persekitaran yang sibuk untuk mencegah kehilangan individu. Bagi mengatasi cabaran ini, WithMe Alarm diperkenalkan sebagai aplikasi mudah alih yang menggunakan teknologi "geofencing". Ia membolehkan penjaga untuk mengesan dan memantau individu, termasuk kanak-kanak, warga emas, atau dewasa. Apabila individu yang dipantau keluar dari kawasan pengawasan, aplikasi ini akan mengeluarkan isyarat berbunyi untuk memberitahu penjaga. Projek ini mengikut model proses perisian "Agile", yang dipilih kerana keupayaannya menyesuaikan dengan perubahan keperluan pengguna dan sistem yang kerap berlaku dalam tempoh masa yang singkat. Proses ujikaji memberikan hasil yang positif, menegaskan kesiapan aplikasi ini untuk dikomersialkan. Projek ini berjaya mencapai objektifnya dalam lingkungan yang ditetapkan.

ABSTRACT

Ensuring the safety of individuals near us traditionally involves keeping them within sight or holding hands. However, incidents of people going missing still occur, particularly among parents and guardians of senior citizens with special needs. Additionally, in crowded areas like Bukit Bintang, Kuala Lumpur, Times Square, New York, or Myeong-dong, Seoul, remaining vigilant and navigating these bustling environments is crucial to prevent disorientation and loss. To address these challenges, the WithMe Alarm is introduced as a mobile application that employs geofencing technology. It allows registered monitors to track and monitor individuals, including children, senior citizens, or adults. When the monitored person goes outside radar, the app triggers an alarm to notify the monitor. The project development follows the Agile software process model, chosen for its adaptability to frequent changes in user and system requirements within a short duration. The testing process yields positive results, affirming the app's readiness for commercialization. The project successfully achieves its objectives within the defined scopes.

Table of Contents

CHAPTER 1

INTRODUCTION

1.1 Introduction

The majority of individuals now regularly utilise mobile devices wherever they are and whenever they choose(Zuva & Zuva, n.d.). Lately, the social community platform has also adopted the idea of location-based service (LBS) to allow location sharing such that the joint sharing of current whereabouts among members(Heiss & Gesellschaft für Informatik., 2011). That is how geofencing significantly useful because the user does not require to designate their location in push-based LBS(Zuva & Zuva, n.d.).

According to Namiot, geofence is a static or dynamic virtual perimeter for a realworld geographic region(Namiot, 2013). The study explained that the location-enable device of LBS user will receive a generated notification about the location of the device when the user enters or exits a geofence. It also can be described proactive LBS to geofence because location-based data is explicitly requested by the user. Based on other research, a geofencing service is a system which monitor coordinates of mobile objects and keep track continuously against geofences.

The driving force behind caretaker services is expected from combination of background tracking and geofencing(Heiss & Gesellschaft für Informatik., 2011). One of the best-known examples is child monitoring where the parents keep track the whereabouts of their children. Moreover, other broad target community such as elderly and patients also cover in caretaker services whereas they require ongoing monitoring and support by their doctors or the nursing staff.

1.2 Problem Statement

The traditional way of keeping someone near us fairly safe is by letting out the other person within our sight or holding hands to secure the person's whereabout. Other than that, if the person is gone in a glimpse and unintentionally, we have to call the person only when the mobile connection allowed through the building. Nevertheless, people missing would still happen and it matters the most among the parents and the guardian of senior citizen especially with special needs.

In addition, the number of Muslim people performing Umrah and Hajj has been steadily increasing, reaching millions since 2021, according to the Saudi General Authority for Statistics (https://www.stats.gov.sa). Based on this fact, a huge crowd is expected in both cities, Makkah and Madinah, and it is common for individuals to lose their way while performing religious activities. Given the significant influx of pilgrims and the vastness of the pilgrimage sites, it is crucial for individuals to stay vigilant and prepared to navigate the crowded surroundings effectively. The same holds true for other highly crowded hotspots in major cities, such as Bukit Bintang in Kuala Lumpur, Times Square in New York, or Myeong-dong in Seoul.

According to a feasibility study conducted with the use of global positional satellite (GPS) location concluded the lives of people with early dementia can be highly boosted thanks to GPS positioning(Milne et al., 2014). The study also included the guardian significantly spent reduced time to search with help of GPS that minimise their burden to monitor whereabout(Milne et al., 2014). Besides that, other similar research trial using GPS-enabled mobile phone for tracking patients with dementia and wandering issues has proven that the tracking technology was greatly reliable and precise when the user compliance is high(Miskelly, 2005). The study mentioned that low user compliance is due to incapable to utilize and correct their mobile setup that consequently bring frustration and rejection of the tracking system(Miskelly, 2005).

The proposed solution is called WithMe Alarm. It is a mobile application that implement geofences technology. It is when a registered user who chose to be the monitor will register another user ID who will be monitored such as a kid, a senior citizen even normal adult. If the person who being monitored is outside the radar, the app will trigger an alarm to alert the monitor. There are several advantages of WithMe Alarm can give to the public especially the guardian of senior citizens.

The advantages of WithMe Alarm are making use of the advanced technology nowadays, people own smartphone devices and short-range distance monitor. Firstly, we need to make use of the advanced technology for traditional problems. There are numerous open-source technology resources such as the Android Developers and Google Developers that enable the software developers to invent a safety application to help the local community. Secondly, today's people generation with vary ages from very young to old own a smartphone device. Hence, it is a smart way to utilize the smartphone to be used as radar monitor towards another person for safety reason. Thirdly, WithMe Alarm application is a short-range distance monitor. Therefore, it is suitable for short range distance location such as home, mall, exhibition hall etc.

In short, the people missing case whether to be a kid or a senior citizen can be reduced with the help of the proposed solution, WithMe Alarm only if it within short range distance and another person has the smartphone in their hand.

1.3 Objective

There are three (3) objectives of this project:

i. To study about the geofences technology and the existing family monitor mobile application in the market.

ii. To develop a geofencing mobile application for monitoring people and triggering alert.

iii. To evaluate the usability of the end product which is the geofencing mobile application.

1.4 Scope

i. The system is not restricted for monitoring the kid and the senior citizen.

ii. The system is developed for an Android smartphone device to geofence another Android smartphone device.

iii. The system is developed using open-source software technology.

1.5 Thesis Organization

There are total of five chapters in this thesis. Chapter 1 shall brief on the introduction to this project, WithMe Alarm. Chapter 2 will analyze and discuss about the existing system and related works to this project. Chapter 3 shall explain about the methodology in details to accomplish this project. Chapter 4 will elaborate on the implementation process until the outcome of this project. Last but not least, Chapter 5 shall summarize the completion of this project.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter discuss about four existing solutions done by previous work related with WithMe Alarm application. It is also cover all the content for analysis of comparison on existing systems and describe the scope, features, the technology and tools used, advantages and disadvantages of the existing works. The relevance of the comparison with the project of WithMe Alarm application is explained also the significants and impacts is mentioned. As a closure of Chapter 2, there is a summary that hightlight the key findings from the comparison reviewed.

2.2 Existing Systems/Works

There are four existing systems done by previous work related with WithMe Alarm application. The systems such that *Love Alarm*, *Find My Kids*, *Pingo by Find My Kids* and *Familo* have been analyzed thoroughly to get the information that can be reference for the project.

2.2.1 Love Alarm

Love Alarm is a mobile app released in beta version that constantly improving and originated from TV series and Webtoon known as 'Love Alarm'. It is an app which the user can give a gift only once every 24 hours by ringing someone's Love Alarm when someone inside the radar in case the user used to confess their love towards the someone(*LoveAlarm - 좋아하면 울리는 공식앱 - Apps on Google Play*, n.d.). The contents and user interfaces (UI) of Love Alarm application are described as below.

Figure 1. 1 Love Alarm Supplementary Page 1

Figure 1. 2 Love Alarm Supplementary Page 2

When a first-time user opens the application, these two pages above will appear and introduce briefly WHAT and HOW Love Alarm work as in Figure 1.1 and 1.2.

Figure 1. 3 Love Alarm Alert Box for Request Permission

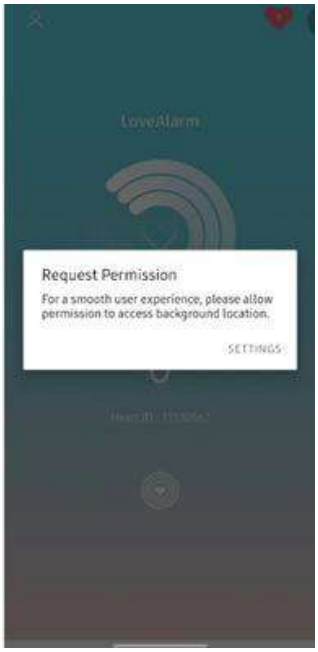


Figure 1. 4 Love Alarm Alert Box for Request Permission 2

The application requires the user to enable permission to access the user device background location as in Figure 1.3 and 1.4. This is due to verify the user entry into radius of the Love Alarm.

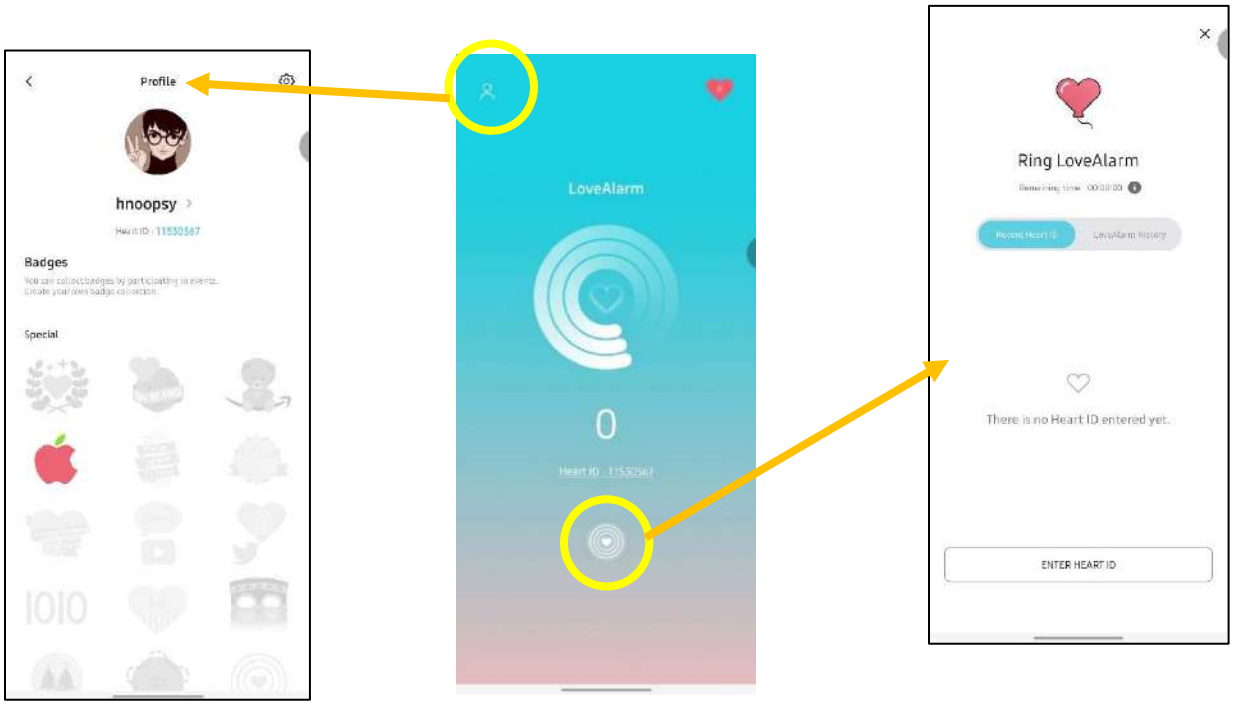


Figure 1. 5 Love Alarm Main Page

Figure 1.5 is the main page of Love Alarm app. It displays animated radar ripple and user unique Heart ID. It can navigate to two pages which are Profile and Ring LoveAlarm.

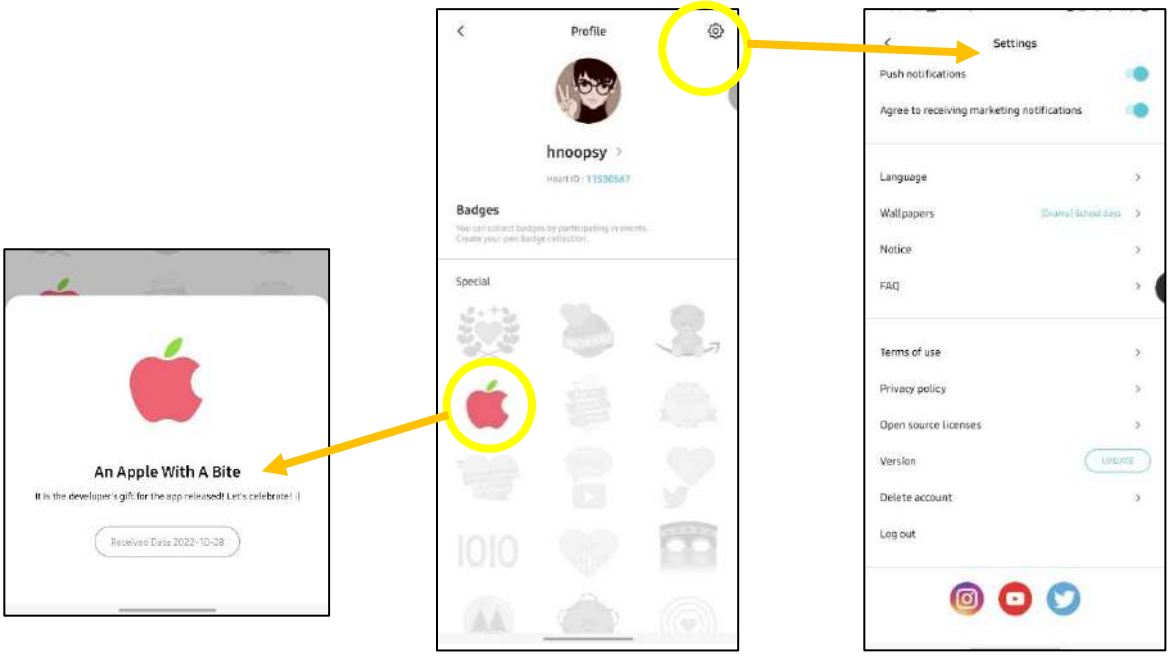


Figure 1. 6 Love Alarm Profile Page

Figure 1.6 depicted Profile page. It shows profile picture, username, Heart ID, and list of collectable badges by participating in events. It also can navigate to Settings page.

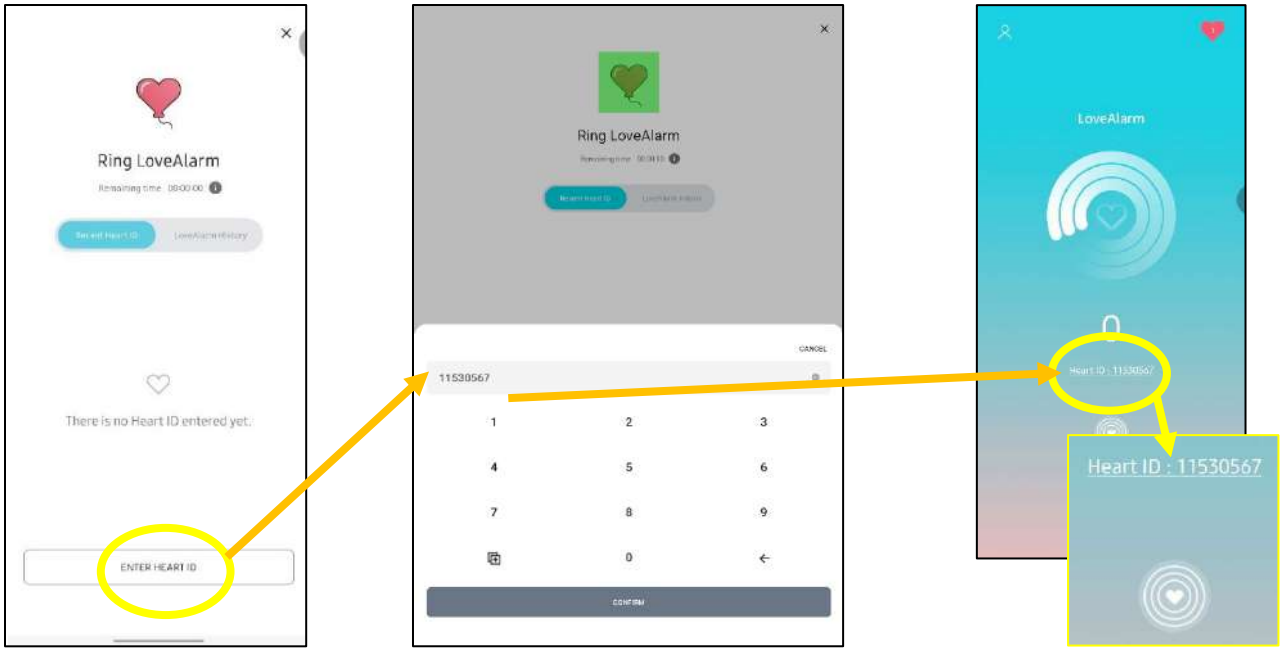


Figure 1. 7 Love Alarm Register Heart ID

Inside Ring LoveAlarm page, the user able to enter someone's Heart ID as in Figure 1.7. The Heart ID can be obtained from main page only if that someone share their ID.

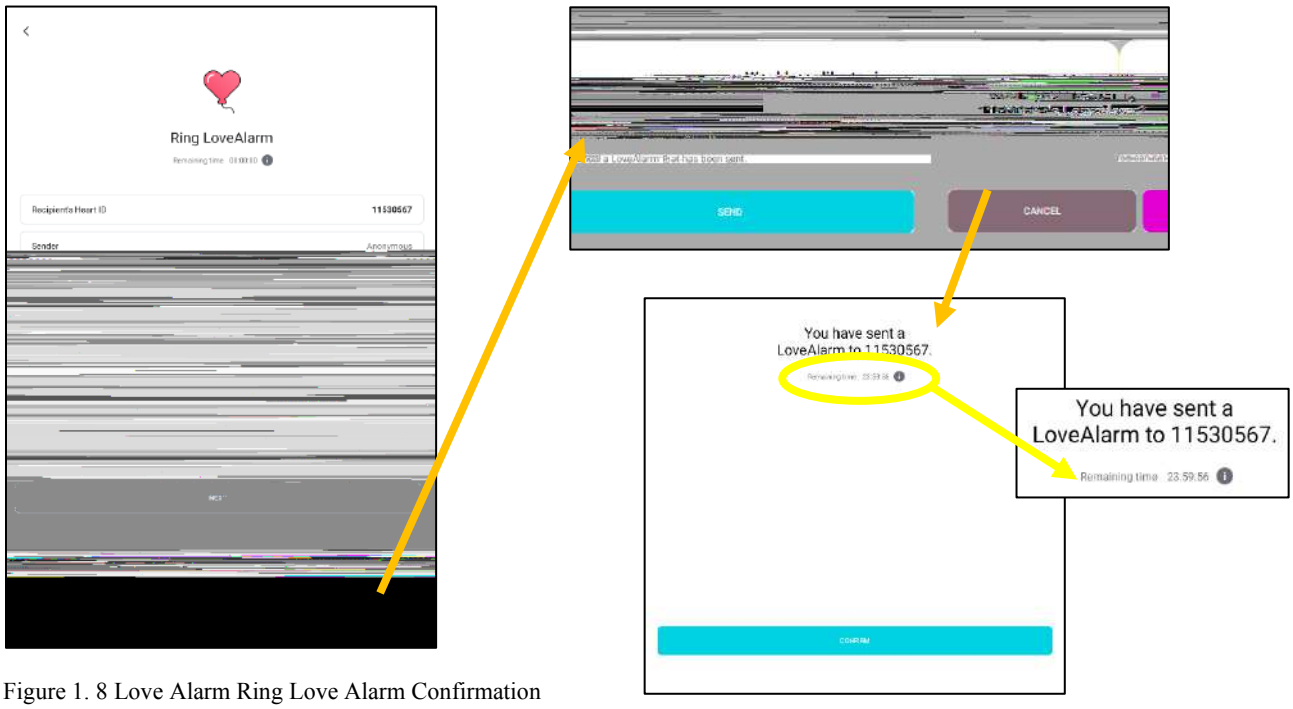


Figure 1. 8 Love Alarm Ring Love Alarm Confirmation

After CONFIRM button is clicked, the user will do confirmation to ring someone LoveAlarm in order to make sure correct recipient's Heart ID as in Figure 1.8. This is due to no return back when LoveAlarm ring sent within 24hours.

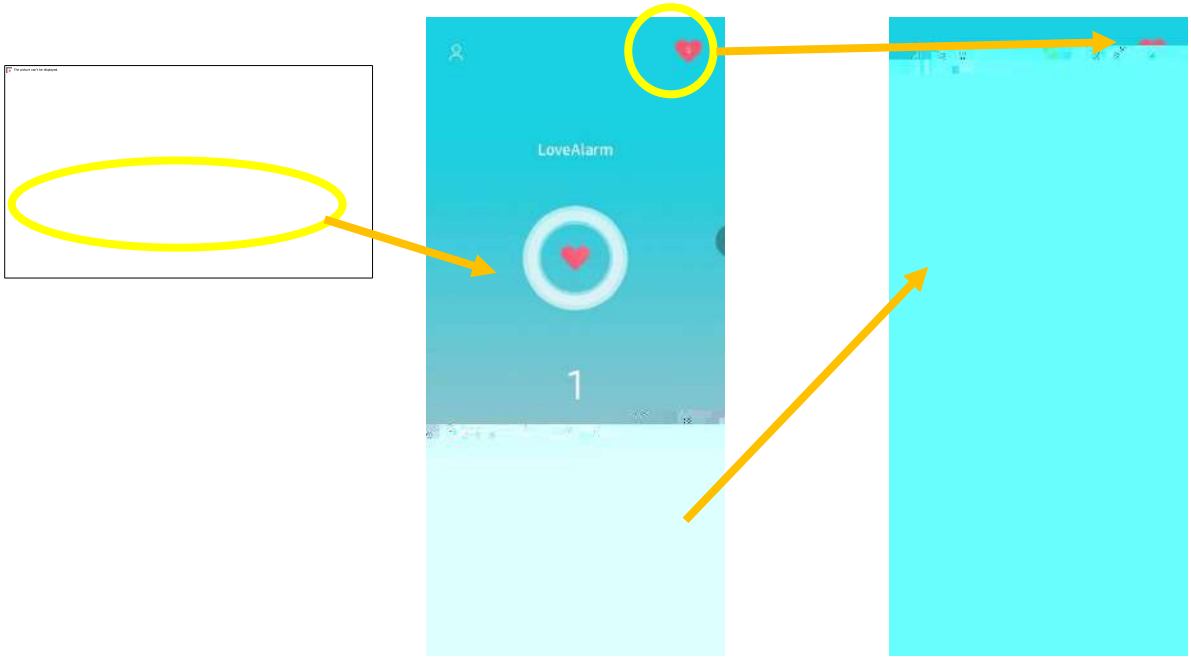


Figure 1. 9 Recipient Love Alarm Interface

Figure 1.9 is from recipient of LoveAlarm ring before. The recipient will get a notification from their device. When the recipient open, the radar diagram will stop animate and give a solid heart also displays number of LoveAlarm ring. After the recipient click on the camera icon to capture the heart, the main page returns normal and number count of hearts will increase shown at the top right corner.

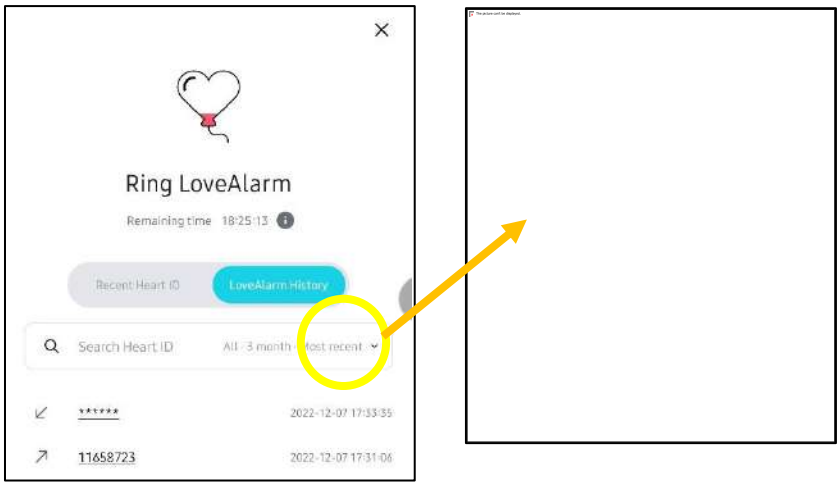


Figure 1. 10 Love Alarm History Interface

Figure 1.10 shows the LoveAlarm History from recipient before. The interface is similar as call history which the downward arrow depicts as receive ring meanwhile upward arrow indicates as send ring to someone. The user also can search previous Heart IDs and sort by order or filter by category and period.

2.2.2 Find My Kids

Find My Kids is a mobile application designed for child's safety and parental controls also known as a family GPS location tracker(*Find My Kids - Family Tracker - Apps on Google Play*, n.d.). This app works by collaborating with Pingo kid tracker app inside child's device such as smartphone or GPS watch. It has multifunctions built inside the app for instance GPS-family locator, battery check, family chat, phone locator, application control, loud signal, and listen in. The contents and user interfaces (UI) of Find My Kids application are described as below.

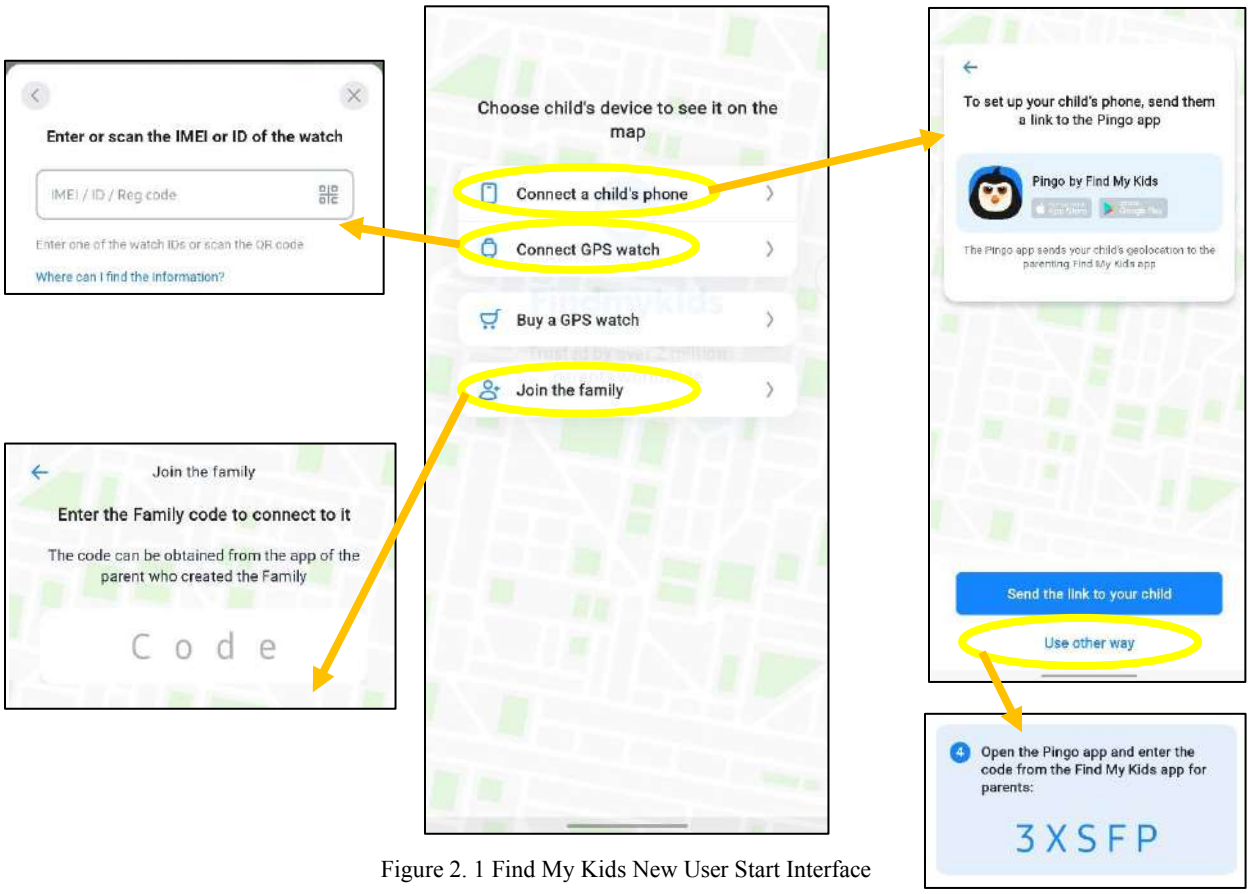


Figure 2. 1 Find My Kids New User Start Interface

For the start as in Figure 2.1, the parent or the guardian that installed this app, they can connect into kid tracker app by two ways which are connecting to a child's phone or to GPS watch. In case their child has not installed the app, the parent can send link for set up installation. Otherwise, the parent can use other way by open the Pingo app inside their child's device and enter the code '3XSFP' like in snapshot above. Other than that, the parent who already created a Family feature in the app, they can share code for other family members to join in family tracker.

Figure 2. 2 Find My Kids Start Setup Page

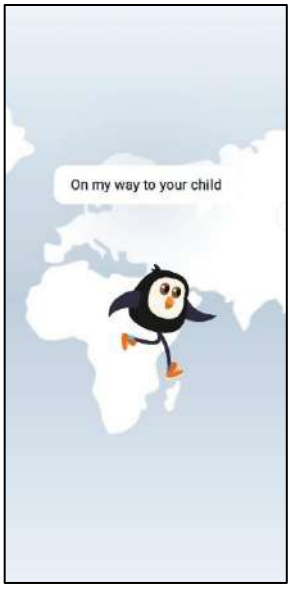


Figure 2. 3 Find My Kids Start Setup Page 2

After device connection finished in Pingo kid tracker app, the parent can continue set up their phone. When the parent clicked on 'Let's go' button as in Figure 2.2, the app will go retrieve the child's information such as location and device condition.

Figure 2. 4 Find My Kids Child Location

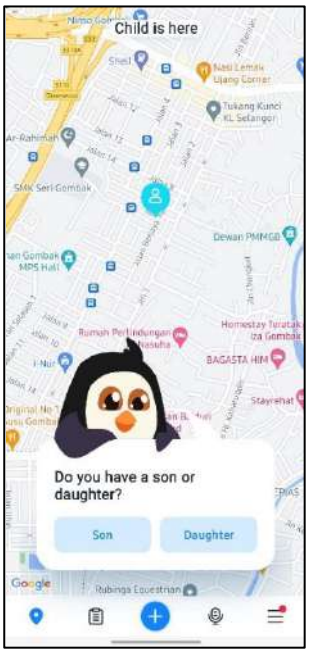


Figure 2. 5 Find My Kids Child Gender

Figure 2. 6 Find My Kids Child Name

After that, the app gained the geolocation and pins the child's location on the Google map as in Figure 2.4. Then, the app will ask for demographic information such as gender and name as shown in snapshots above as in Figure 2.5 and 2.6 respectively.

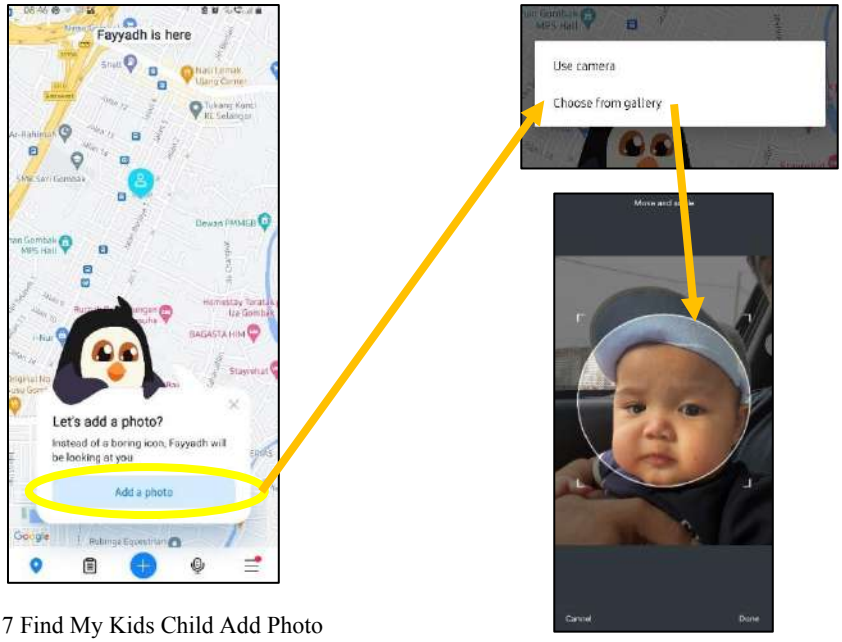


Figure 2. 7 Find My Kids Child Add Photo

This app also can offer the parent as a user to add a photo to the icon of child's location as in Figure 2.7. The parent can choose to use camera or from gallery. Then, the parent can adjust the photo to desired fit.

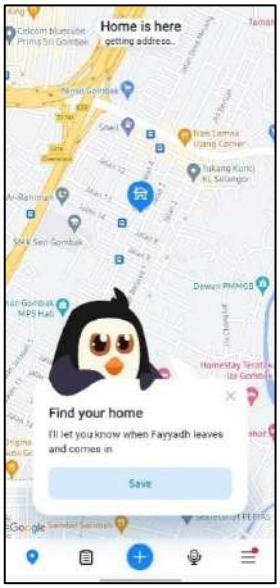


Figure 2. 8 Find My Kids Find Child Home

Figure 2. 9 Find My Kids Allow Important Notifications

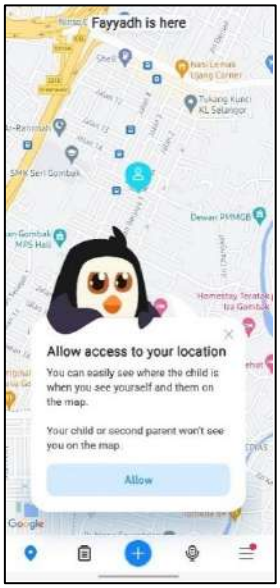


Figure 2. 10 Find My Kids Allow Location Access

Figures 2.8, 2.9 and 2.10 above show a few last steps of setup. The app said that it will let know the child leaves and comes in home. Next, the app request permission to allow the app send important notifications. Lastly, the app also requests the parent's location to easily see where the child is.

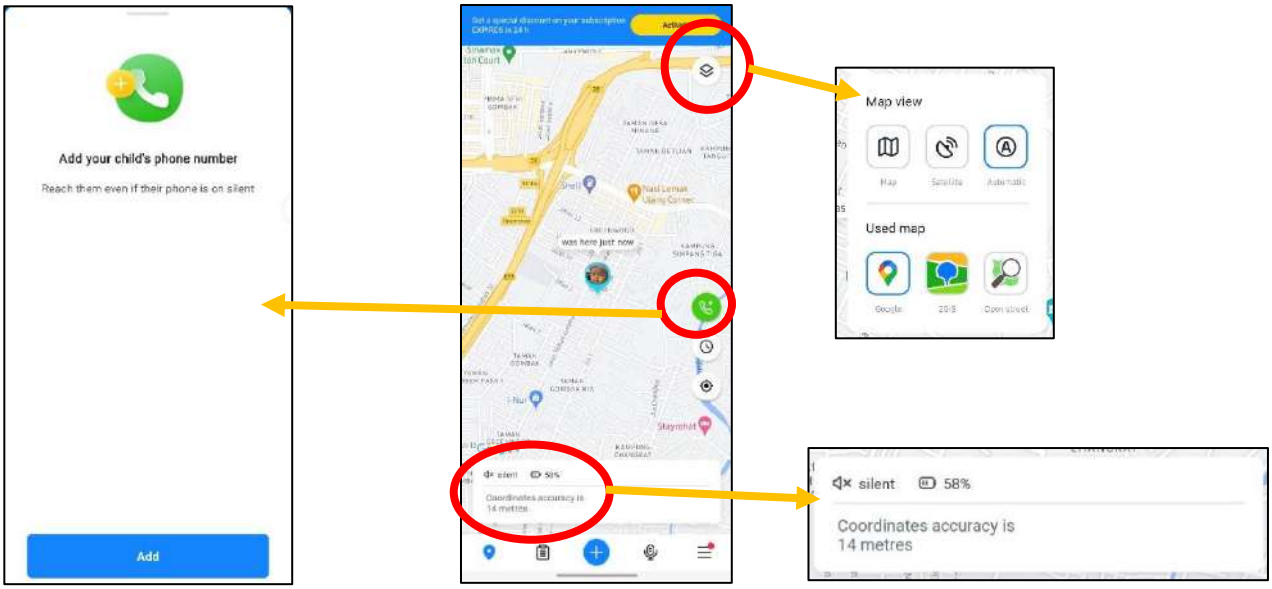


Figure 2. 11 Find My Kids Main Page

Figure 2.11 above is the main page of Find My Kids. The parent can see the pin location of the child and the parent can customize map view (Map, Satellite or Automatic) and used map (Goggle map, 2GIS, or Open street). The parent can add the child's phone number to reach them even in silent mode. The parent also can check the child's device sound, battery life and geolocation accuracy.

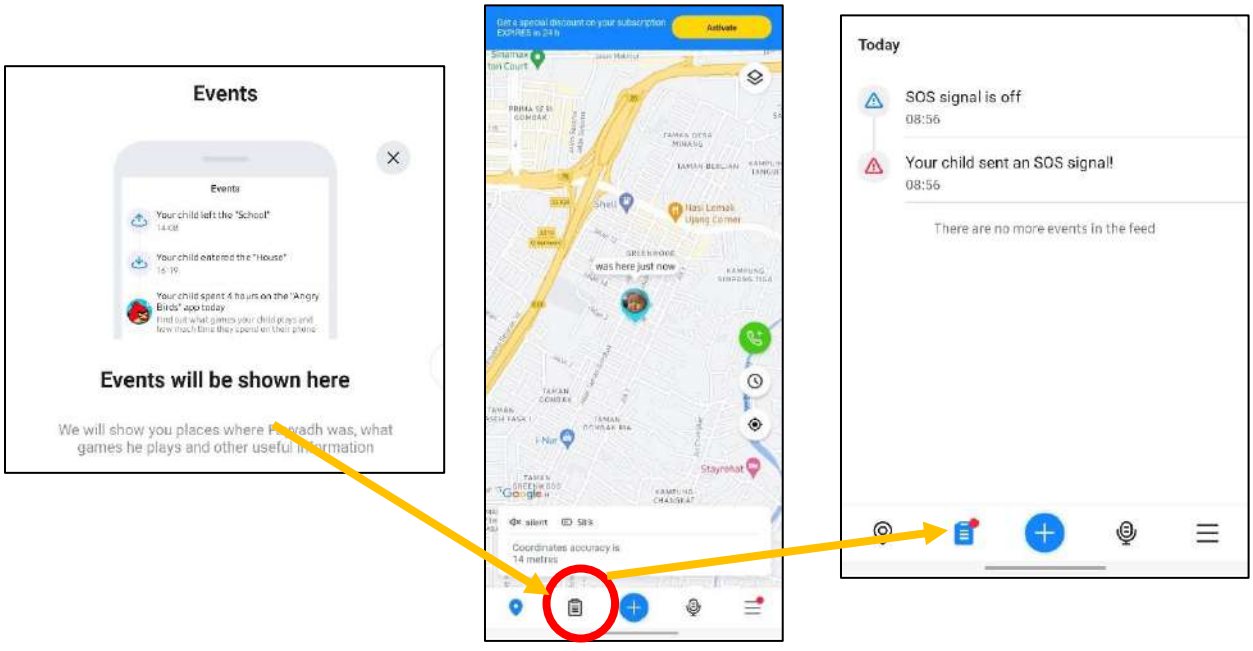


Figure 2. 12 Find My Kids Event Page

From main page, the parent can navigate to events page as in Figure 2.12. This is where the list of events history such as the child's whereabouts. For instance, an event recorded when the child sent an SOS signal as shown in the right snapshot above.

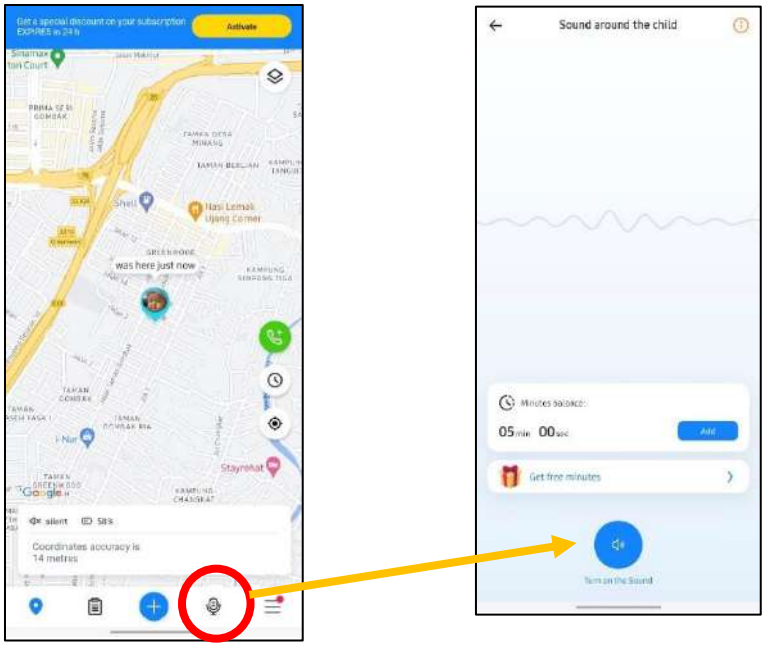


Figure 2. 13 Find My Kids Sound Around the Child

The parent can access to listen in the sound around the child for safety reason as in Figure 2.13. When the parent turns on the Sound button, the parent gets to listen the child surrounding within minutes given.

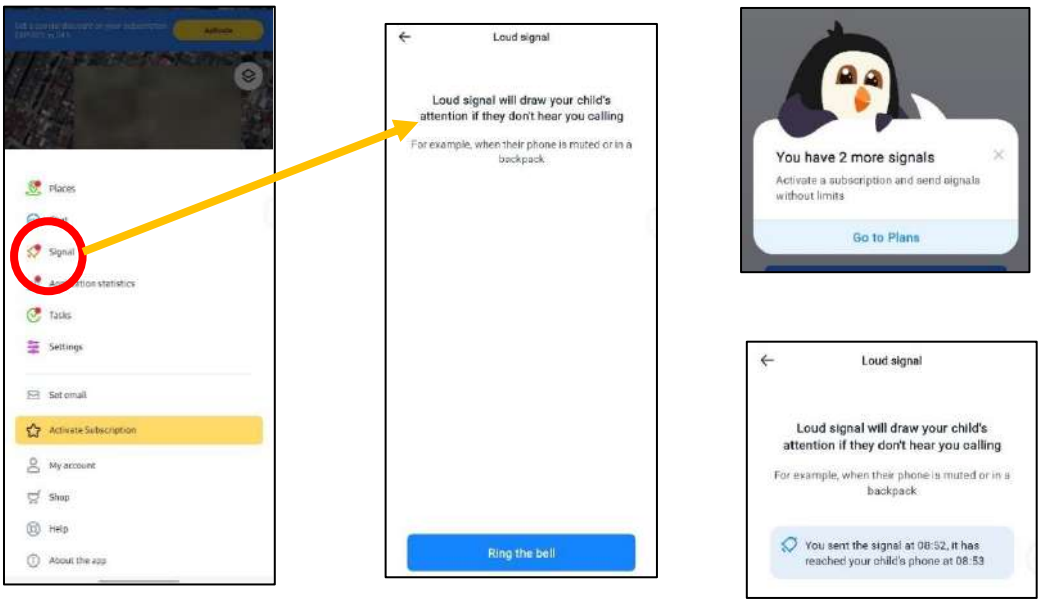


Figure 2. 14 Find My Kids Loud Signal

The app has a unique function where the parent can send loud signal or bell when urgent time, but the child's phone is muted or in the bag as in Figure 2.14. By default, the parent has 3 signals to send daily, and the signal can be stop in child's app. After signal sent, the app recorded the timeline usage of signal as shown in the snapshot above.

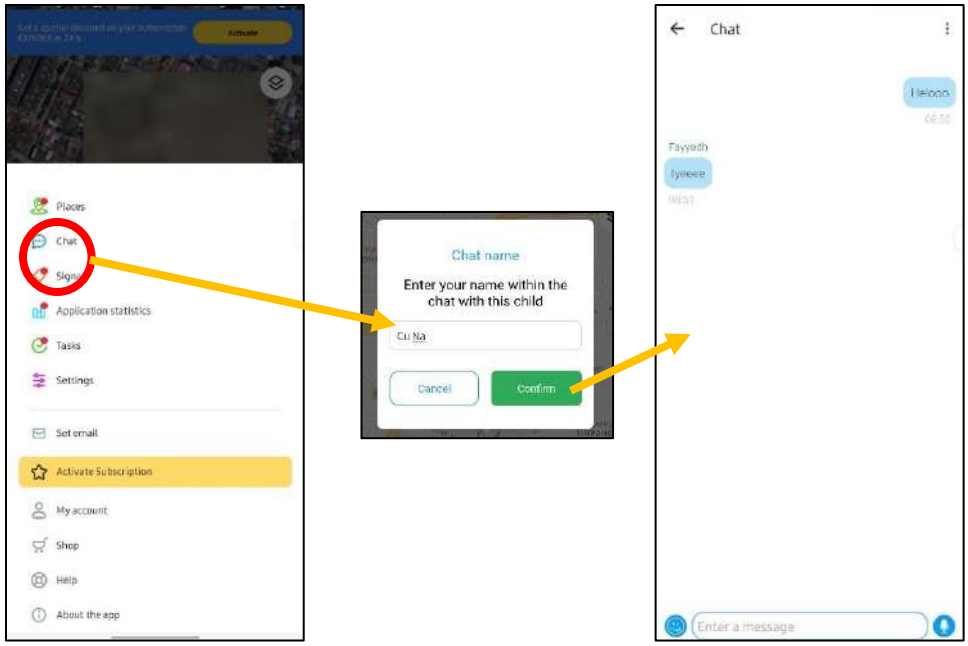


Figure 2. 15 Find My Kids Chat Feature

The app also has built-in chat feature as in Figure 2.15. The parent enters chat name before conversation. The interface of chat is similar with other chat message apps, so it is easy to use.

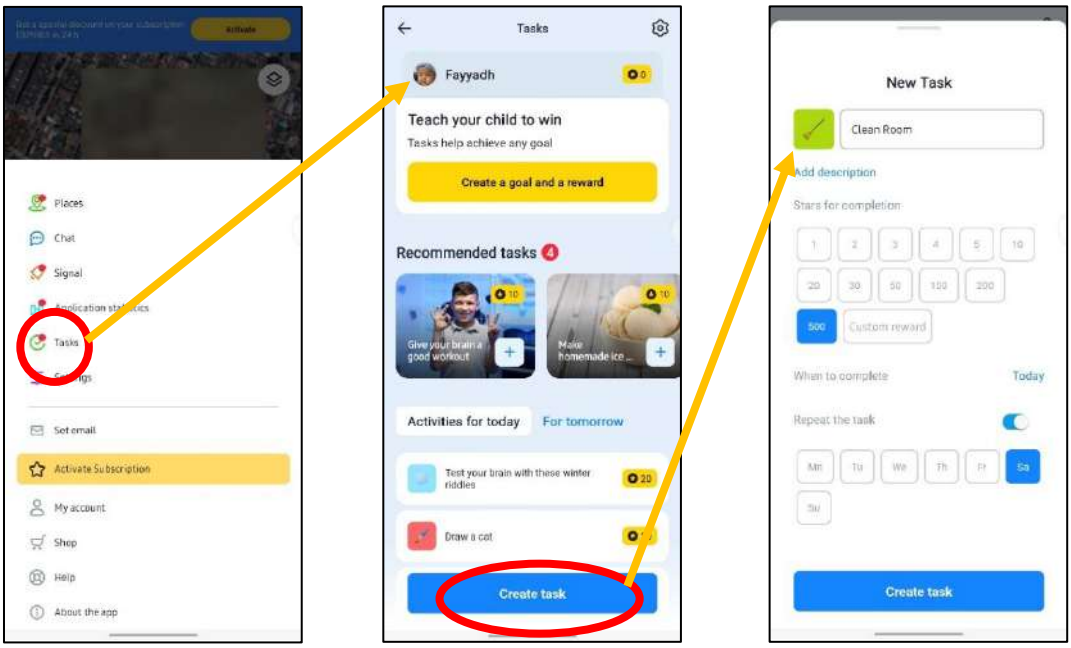


Figure 2. 16 Find My Kids Create Child Tasks

This app also other fun feature which the parent can create task and give rewards to the child. The parent can add task's title, icon, description, stars reward, duration of completion and task frequency. This feature ease the parent in child education and make activities with their child.

2.2.3 Pingo by Find My Kids

Pingo by Find My Kids is a mobile application targeted to a child or teenager for family location tracker(*Pingo by Findmykids - Apps on Google Play*, n.d.). This app is a companion app to Find My Kids which is for parents that has been explained previously. The multifunctions inside this app are similar but different scope with the parent's app for instance, GPS locator, Family chat, SOS signal, battery control and list of task activites created by the parent. The contents and user interfaces (UI) of Pingo by Find My Kids application are described as below.

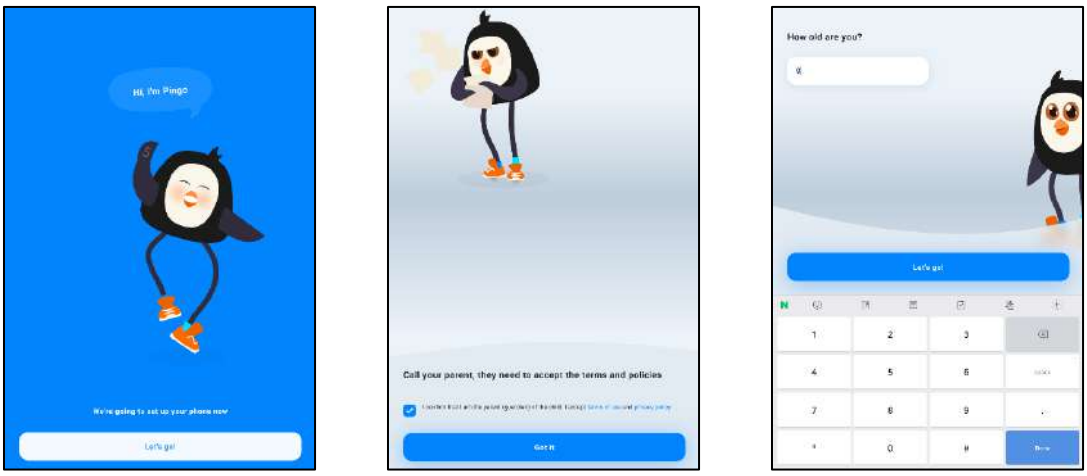


Figure 3. 1 Pingo Welcome Pages

Figure 3.1 above are snapshots of the welcome page for the child to setup this app. The parent needs to accept the terms and policies to continue the setup also asks for age of the child.

Figure 3. 2 Pingo Setup Code Process

Next, the app asks for the code which can be retrieved from the parent's app as shown in Figure 3.2. The app will verify the code in order to continue to setup.

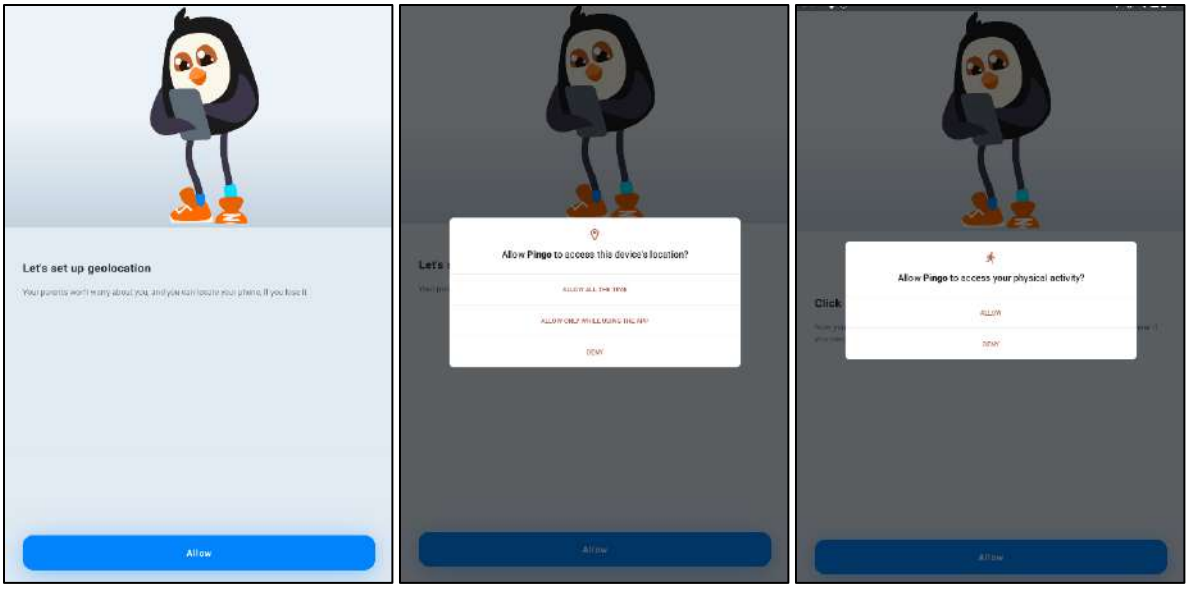


Figure 3. 3 Pingo Allow Permission for Geolocation Setup

After that, the app will request a few permissions to access from the child's device. The snapshots above show the app request to access the location and the physical activity of the child for geolocation setup.

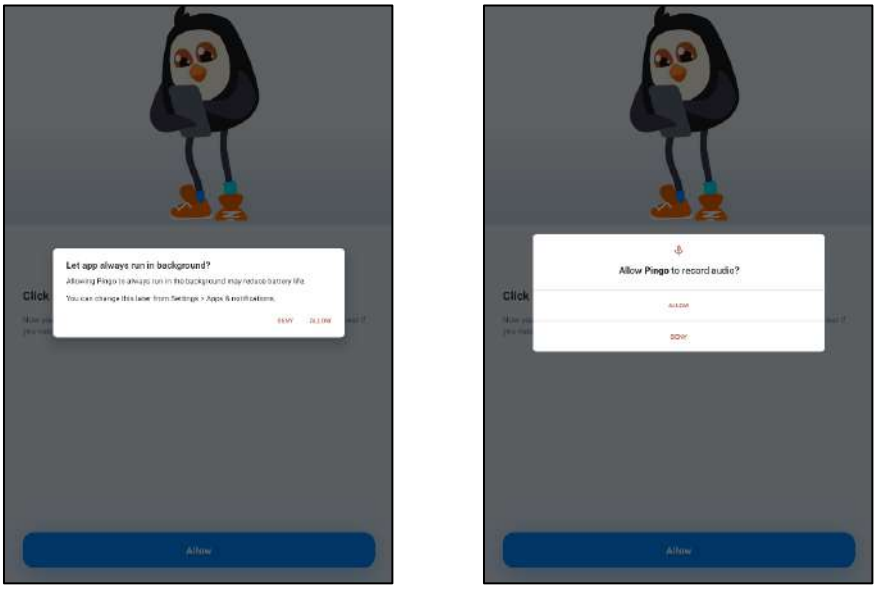


Figure 3. 4 Pingo Allow Permission for Run Background and Record Audio

Next, the app request to let the app always run in background and record audio. The purpose is to make sure the child's device is always accessible conveniently from the parent's app and the app able to track and record geolocation whenever for safety reason.

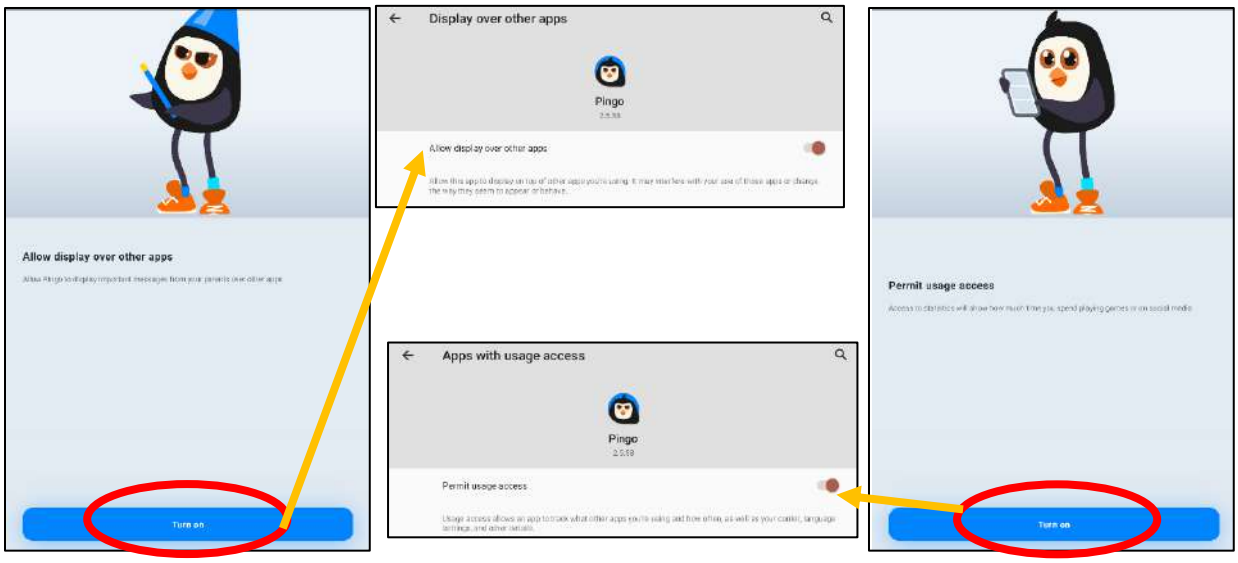


Figure 3. 5 Pingo Permission to display over other apps and child usage access

There are several more setup steps before main page appear. The app request for allowing to display important messages from the parent over other apps in child's device. Besides, the app also asks for permit usage access in order to track the child's phone usage for the parent's view.

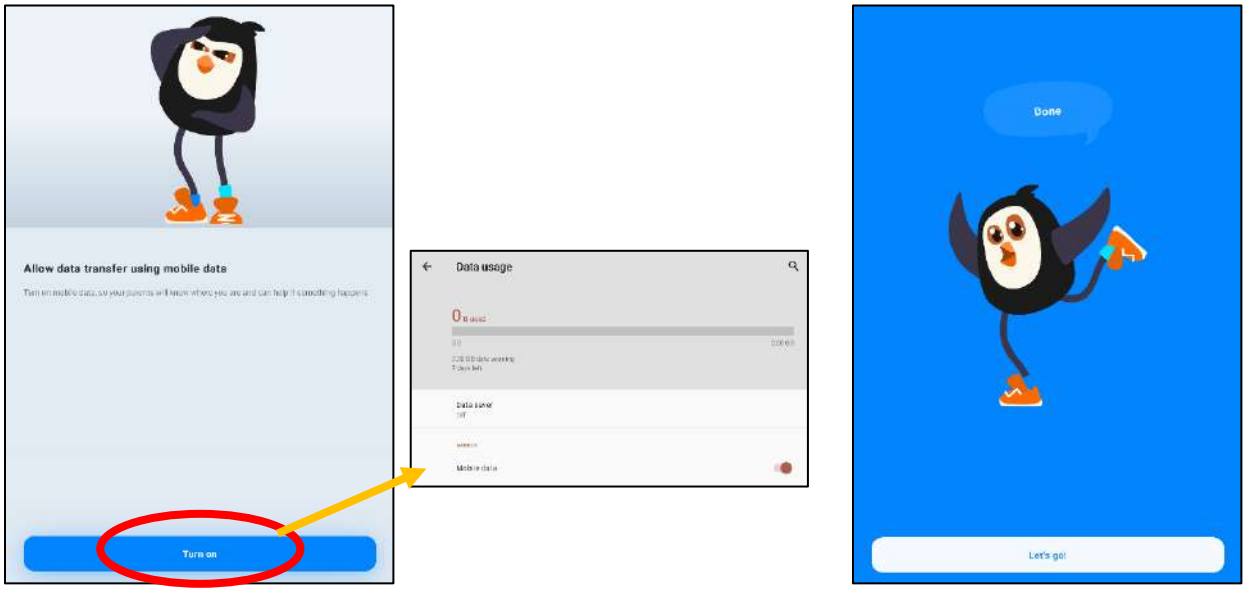


Figure 3. 6 Pingo Permission to use Mobile Data

Lastly, the app request to turn on the mobile data so the data transfer to the parent can be easy to access. This is due to safety reason so the parent can detect recent child's location and reach out to the child conveniently. Therefore, the setup of Pingo app has come to an end.

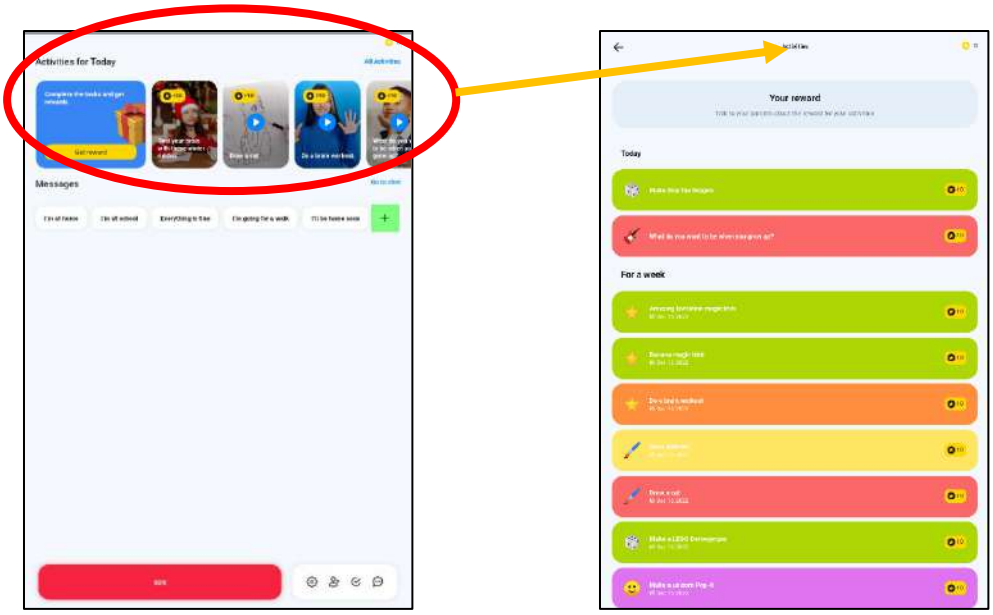


Figure 3. 7 Pingo Main Page

Figure 3.7 is the main page of Pingo application. From there, it can navigate to the activities page where the child can complete the list of tasks in order to receive rewards.

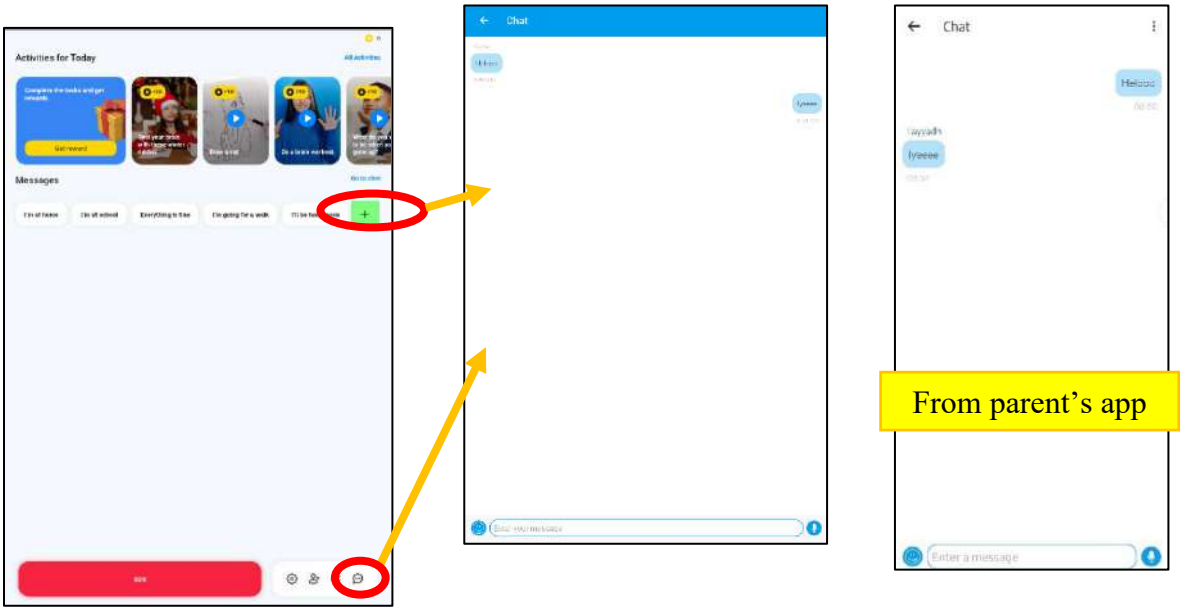


Figure 3. 8 Pingo Child Chat Message

The app also can navigate to chat page from main page as shown in Figure 3.8. There is also list of readymade messages which the child can choose plus free to create a new message. The UI of chat in child's app is similar with the parent's app and the minimalist design enable a child to teenager to utilize the chat feature.

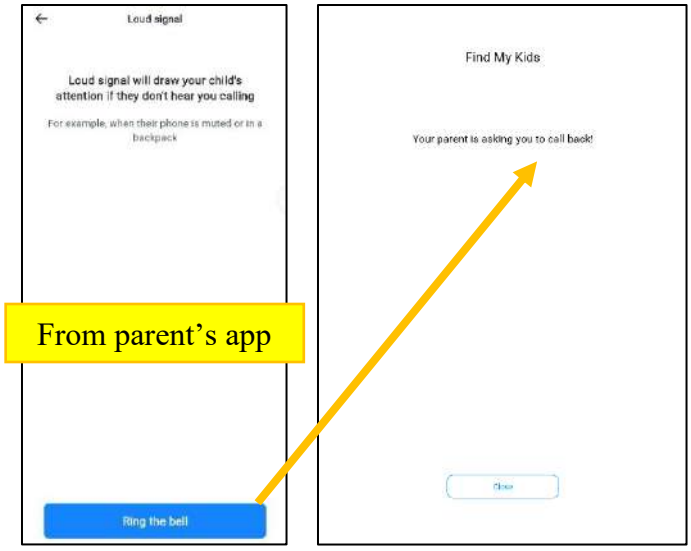


Figure 3. 9 Pingo Loud signal sent by parent

From parent's app (Find My Kids), when the parent ring the bell, an increasing loud signal sound come out from the child's device through Pingo app. The UI of child's app is as shown in Figure 3.9 above that inform the child to call back the parent. The loud signal will turn off when the child clicks on the 'Close' button.

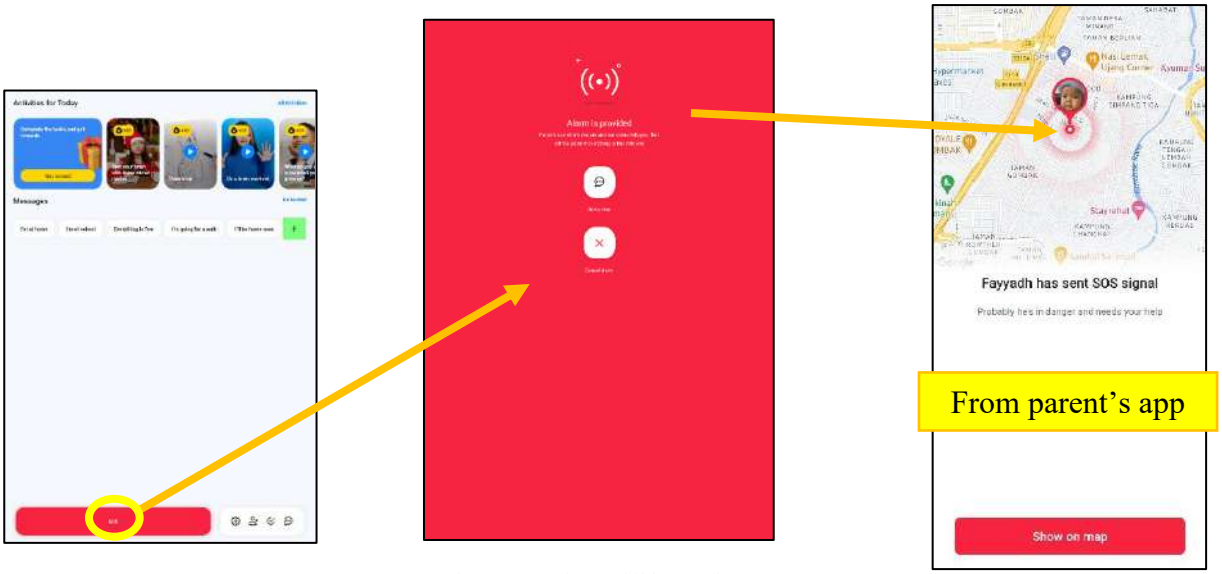


Figure 3. 10 Pingo Child SOS alarm

For the parent's app, it has unique function which is loud signal to reach the child meanwhile for child's app, it has SOS alarm to send to parent's app. The child can access to the SOS button from their main page at the bottom page as shown in Figure 3.10. There will be a very loud alarm sound from child's app and the parent will get an SOS notification. The child can stop the alarm or chat the parent for help while at it. From parent's app, the parent will be able to immediately check the child's location from the map and reach out to the children to rescue.

2.2.4 Familo

Familo is a mobile application for family GPS locator also a 'find my phone' app that can be share with family members. This app has helpful functions and quite distinct from the existing systems mentioned before. This app able to locate family real-time on map, know when the family members leave or arrive, user can own several private group chat, and user can be anonymous inside the app(*Familo: Find My Phone Locator - Apps on Google Play,* n.d.).The contents and user interfaces (UI) of Familo application are described as below.

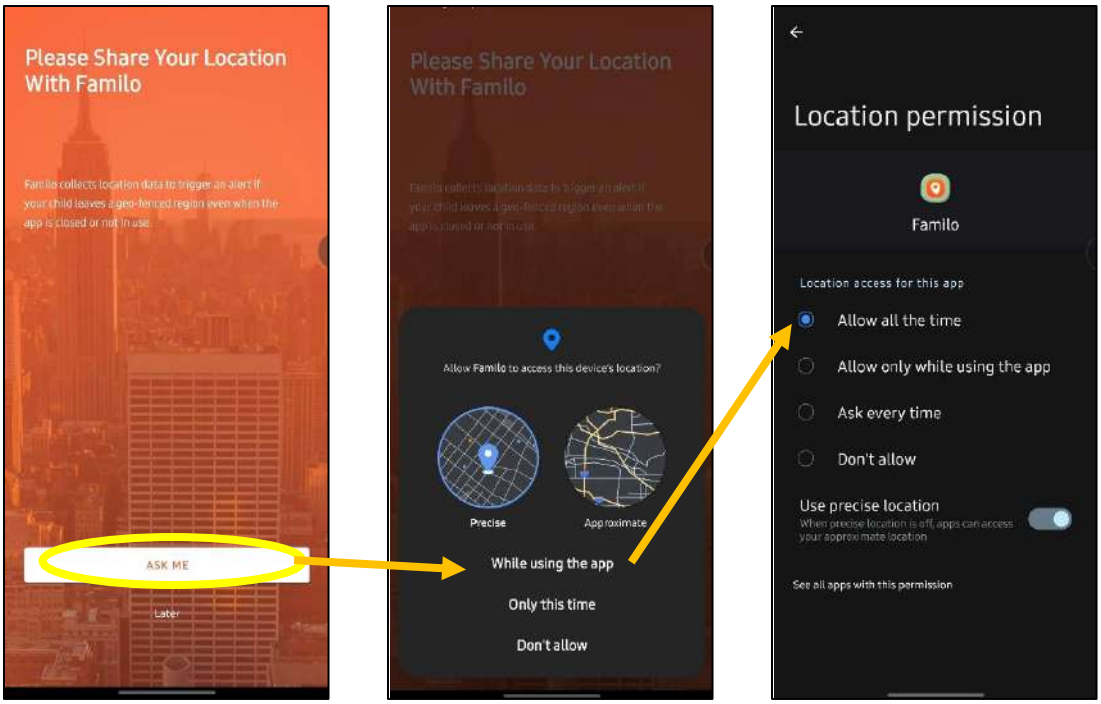


Figure 4. 1 Familo Welcome Pages

The welcome page is as shown in Figure 4.1 above where the app asks for permission to allow the app to access the user's device location. The user can adjust the location permission in the device settings.

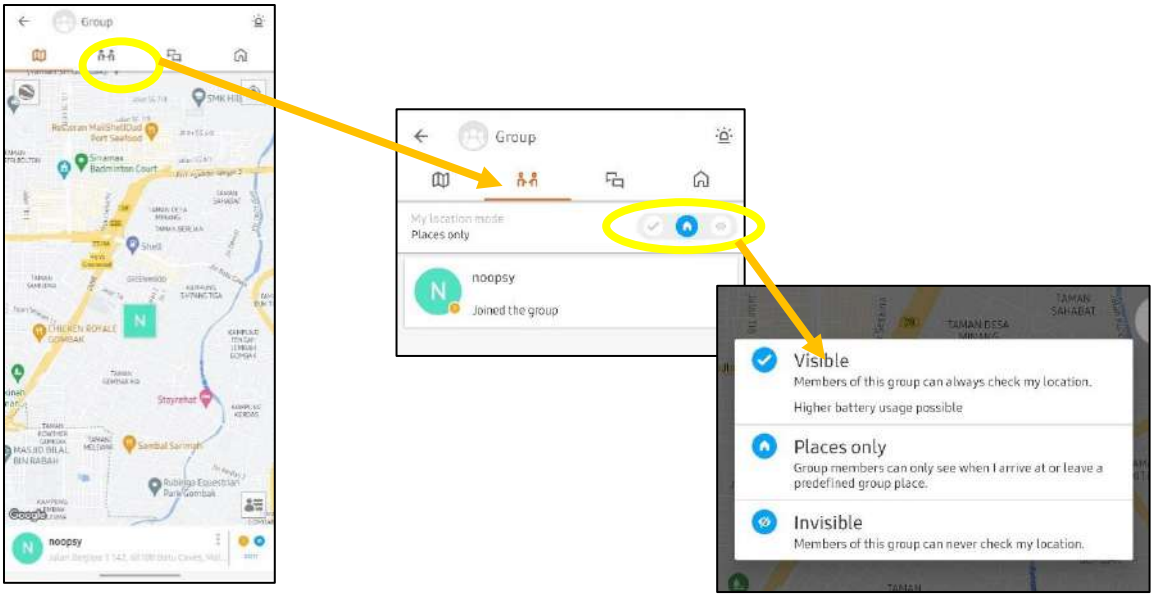


Figure 4. 2 Famlio Main Page

Figure 4.2 is the main page of Famlio app. The next page is where the list of group members. There are three location modes above the list which the user can choose such as 'Visible'(group members can always check user's location), 'Place only'(group members can see when the user arrives or leave) or 'Invisible'(group members forbid from view the user's location).

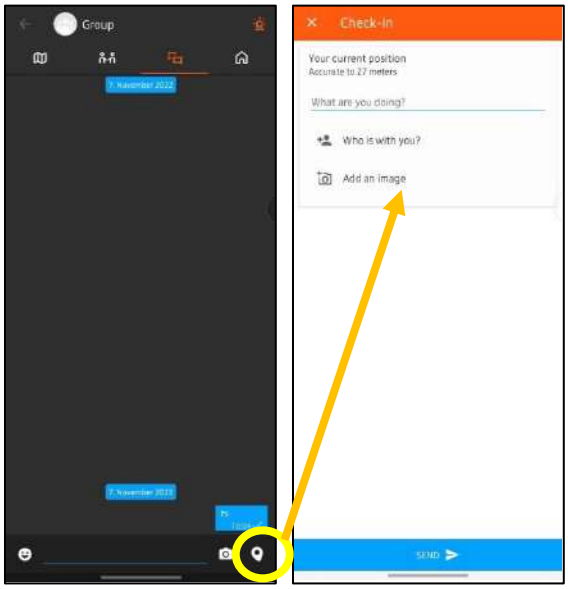


Figure 4. 3 Famlio Check-in Chat

Next is the private group chat. The unique feature in the chat is that the user can send Check-in information into the chat. The user can share what activity, with who and insert an image to group members as in Figure 4.3.

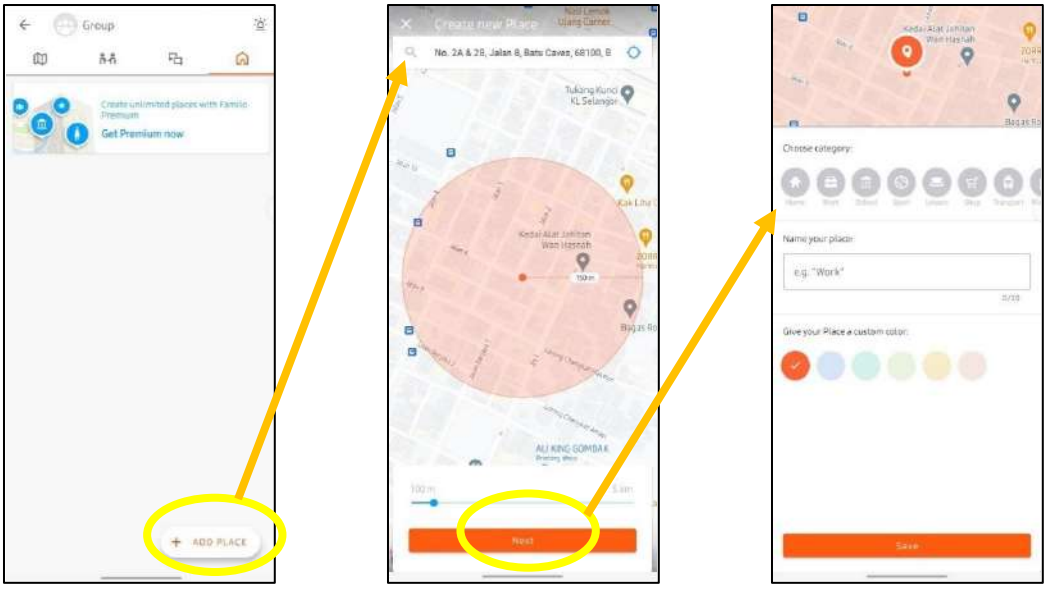


Figure 4. 4 Familo Group Map

The user can create a place into map for the group as depict in Figure 4.4. The user can set how wide radius of the place for members can pin their location on the map. The user also can set the category, name of place and custom color before save the new place.

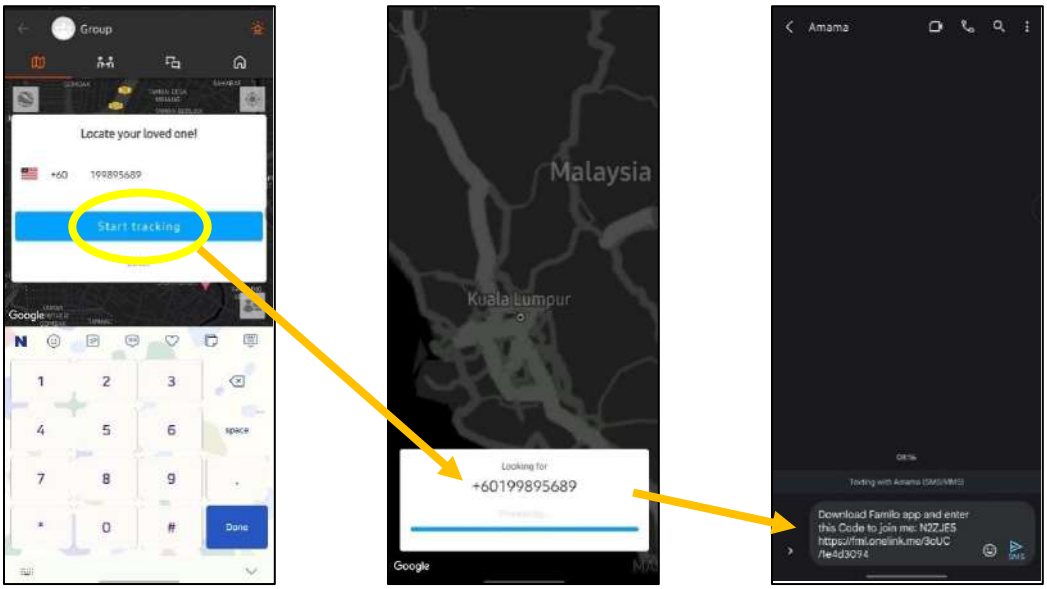


Figure 4. 5 Familo Add by Phone Number

The user is able to track their loved one by inserting phone number to retrieve the device location. The app will pin the location on the map if the loved one has installed the app otherwise it will send invitation link with group code to join through SMS as shown in the snapshot above.

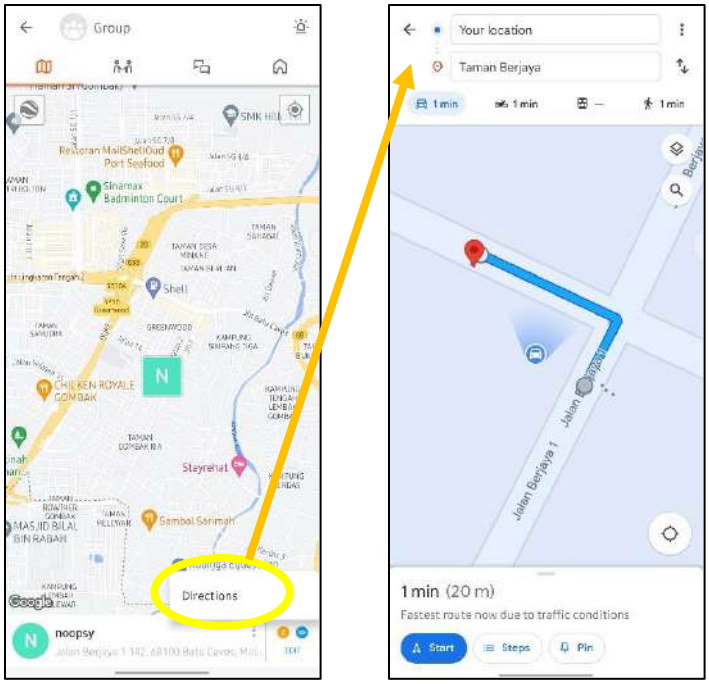


Figure 4. 6 Famillo Member Google Map Direction

The user can get directions to go towards the group member with the location pinned as shown in Figure 4.6. The user will be sent into Google Map along their GPS location and the app will give directions to the destination. Therefore, the user will be less hassle to paste location of other member into Google Map or similar apps.

2.3 Analysis/ Comparison of Existing System

2.3.1 Analysis of comparison on existing system.

Table 1 shows the analysis comparison of four existing systems based on the scope, features, access permissions, tools and technology used, advantages and disadvantages.

Existing Work	Scope	Special Features	Access Permissions	Tools/ Technology	Advantages	Disadvantages
Love Alarm	-For love interest -Not suitable for minor kids -Radius range up until 100 meters	-Ring LoveAlarm Module -Register Heart ID Module	-Background location, important notifications	Open-source tools and projects Android Extension Library (AndroidX) **Programming Language:** Java & Kotlin	-Simple and minimalistic UI. -High familiarity in features and direct instructions. -Free access to all features. -The app able to quickly detect person inside radar.	-Incapable to detect precise location of ring sender and recipient.
Find My Kids	-For parent usage -No radius required	-Events Module -Listen Around Module -Loud Signal Module	-Background refresh, important notifications	**ServerHost**: Yandex AppMetrics **Database**: Firebase Facebook Analytics	-The app has loud signal to reach out the child when device is muted. -The app capable to record sound around the child.	-Only capable to give location on GPS map not inside building. -Requires constant internet connection and paid subscription to use all features.
Pingo by Find My Kids	-For child usage -No radius restriction	-SOS alarm Module -Task Module	-Background refresh, location, physical activity, record audio, app usage access, display over other app, mobile data		- The app has SOS alert for help and safety. -Kids-friendly UI.	-Requires constant internet connection to always update activity and location. -Consumes high battery usage for app background refresh.
Familo	-No age restrictions -Radius range from 100 meters to 5 kilometers	-Track User Module -Add New Member Module -Create Place Module	-Background refresh, location, microphone, contacts, notification, mobile data	**Database**: Firebase, **MapView**: Google Map, **ServerHost**: Google Cloud, Nexmo SMS API	-The app able to give GPS direction to another user instantly. -Simple and minimalistic UI.	-Only capable to give location on GPS map not inside building. -Requires paid subscription to fully utilize features in the app.

Table 1 Analysis of comparison on existing system

2.3.2 Relevance of comparison with project title

All of four existing applications mentioned before having one similar purpose which is to track people or retrieve geolocation of another user from the app. Most of the applications mentioned also require the user's device to give access on location, app background refresh and mobile data in order to send and receive data of the user. The comparison is relevant to the proposed project title (WithMe Alarm) as it is also requiring geolocation technology. The chosen existing applications were to consider as reference for development of WithMe Alarm app based on the scope, features, access permissions, tools and technology used, advantages and disadvantages.

The significant is most of the user interface of the existing applications are minimalist and user-friendly design. The applications like Find My Kids and Familo are focus on the use of map view which is the most significant feature in the app in order to track location. The Love Alarm app has minimized data input and has interface that directly focus on the main function which is located in the main page. The target user of Pingo app is a child so the interfaces are very kid-friendly and wise in colors which is vibrant color. Each of the applications has unique features especially in Find My Kids and Pingo which have record audio of surrounding feature and loud signal alarm sound for emergency respectively. WithMe Alarm can apply this kind of features with regard to give alarm while monitor the people in the radar as given name of this project title.

The user experience of the related applications must have brought positive impacts due to the number of high ratings and reviews received from the app market. This is due to each of the applications has its own target users, purpose and solutions onto the problem they tried to solve. There are several disadvantages of the related applications but the most frequent mentioned is the app only capable to give location on GPS map. Therefore, with this project title (WithMe Alarm) can turn the disadvantage from the related applications mentioned before into a solution. Moreover, the special features and the significant of apps also can be a good reference to implement inside the project development.

2.4 Summary

To summarize, Chapter 2 is about four existing mobile applications that related with the project title. The four applications are *Love Alarm*, *Find My Kids*, *Pingo by Find My Kids* and *Familo* which have been analyzed into table in regards to get the information that can be reference for the project. The analysis of comparison is based on the scope, features, access permissions, tools and technology used, advantages and disadvantages. The high relevance of the comparison with the project title also explained along with the significant of the special features inside related apps and the good impacts is as well mentioned.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter describe about the methodology adopted for WithMe Alarm application development. The details cover the process in phases to complete the project development with the suitable project management. Besides that, this chapter also consists of WithMe Alarm Project and User Requirements, WithMe Alarm Propose Design, WithMe Alarm Data Design, WithMe Alarm Proof of Initial Concept, WithMe Alarm Testing Plan and Potential Use of WithMe Alarm solution.

3.2 Project Management Methodology

The methodology of WithMe Alarm application for project development adopted is based on Agile software process model. It is the best suitable model for this project development due to flexibility to frequent change of user and system requirements in short duration with low budget(*IJCSC,* n.d.). It is also an iterative and incremental model that requires continuous customer feedback in order to deliver amply satisfied product and secures reliability. The development of WithMe Alarm application will go through phases such that (1) Plan (2) Design (4) Develop (4) Integrate and Test (5) Release and Feedback.

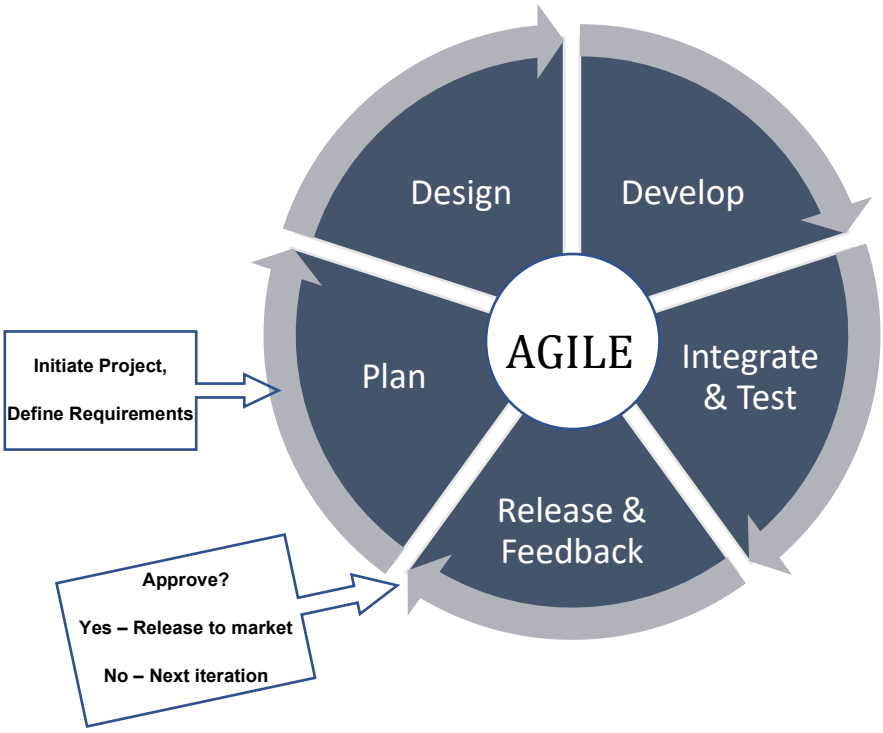


Figure 5. 1 Agile Methodology for WithMe Alarm App

Plan

This phase is the start where to initiate the project development. The background of the WithMe Alarm app is predetermined, and the problems is underlined during plan phase. The objective and scope of the project are also defined in order to accomplish the project development. All of this information is documented in the Chapter 1. There are three existing related works have been reviewed before defining the requirements for WithMe Alarm app. The reviews are for identify the user and system requirements applied. The requirements in those three related works can be as reference and do implementation into WithMe Alarm app. The reviews have been discussed in Chapter 2. After that, project management methodology is identified to be the best suitable for mobile app development which is Agile model. The project requirements such as hardware, software requirements and how should WithMe Alarm app work like functional, non-functional requirements and constraints are described in this Chapter 3.

Design

The design phase is where proposed design related with the project requirements of WithMe Alarm app are created to show the flow of running app. From the requirements, system designs are illustrated in form of flowchart, entity relationship diagram, storyboard and more. All the information of design depicted in Chapter 3.

Develop

This phase is where the WithMe Alarm app start to develop relied on the specifications set in this project. It is involving actual coding and object-oriented programming implementation to WithMe Alarm application. All the information will be compiled into project documentation in Chapter 4.

Integrate and Test

Once WithMe Alarm app development is completed, it is ready to perform integration and testing. It is to ensure all of individual modules are combined and tested as a whole system. It is also tested to meet all the project requirements and follow testing plan defined in previous phase.

Release and Feedback

After testing plan is passed smoothly, WithMe Alarm app can be released to targeted users in order to receive feedback. If the targeted users satisfied, then WithMe Alarm app is fully ready to release to the market. Otherwise, the WithMe Alarm app development will go through next number of phase iteration along with the feedback received that will be analyse for refining the project requirements.

3.3 Project Requirements

Hardware Requirements

- System: Intel core i5 CPU @ 1.60gHz 1.80gHz
- Required space: 50GB
- RAM: 12GB

Software Requirements

- IDE (Integrated Development Environment): Android Studio Dolphin (2021.3.1)
- Database: Google Firebase
- Programming Language: Kotlin

User Requirements

Functional Requirement

a. WithMe Alarm User Registration Module
    - This module is for users who wants to retain their data while utilizing the application.
    - The module must allow the user to register by their personal email and new password when clicked on 'SIGN UP' button.

b. WithMe Alarm Login Module
    - This module uses Firebase for account authentication for enable the user to log into the app.
    - The module must validate the user's email and password for successful log in.

c. WithMe Alarm User Profile Module
    - This module displays user's profile inside the app such as name, email, password and WithMe Alarm unique ID.
    - The module is for the user to edit and insert additional information.

d. Add and Delete WithMe Alarm user ID Module
    - This module enable user to add and delete WithMe Alarm user ID of the other who they chose to monitor.
    - The module must allow the user to register new WithMe Alarm user ID to trigger radar alarm and delete unwanted the registered ID anytime they want.

e. WithMe Alarm Trigger Module

- This module functions once the user set with their targeted WithMe Alarm user ID and obtained range of location, WithMe Alarm app will trigger the alarm sound and notification on screen to indicate the targeted user is outside of the radar.

Non-Functional Requirement

a. The WithMe Alarm mobile app use of storage space shall be below 100MB. This is due to the app does not have to load high-definition multimedia elements like images also it is not an app purposed to be watch daily. Thus, the app may take minimal amount of device's storage space to store data.
b. The WithMe Alarm mobile app shall consume efficient energy to run all modules. For the reason that, the app will implement high accuracy priority of location services request. It is a setting using GPS to determine location. However, in order to guard against battery drain, the app will apply high interval location data request and set a reasonable timeout such as after 1-hour geofence location request should stop.

Constraints and limitations

a. The WithMe Alarm mobile app depends on constant internet connection in order to utilize some modules such as login module.

b. The WithMe Alarm mobile app may drain the battery life when user location is always turned on deliberately.

3.4 Propose Design

i. Flowchart

a. User flow for Guardian role

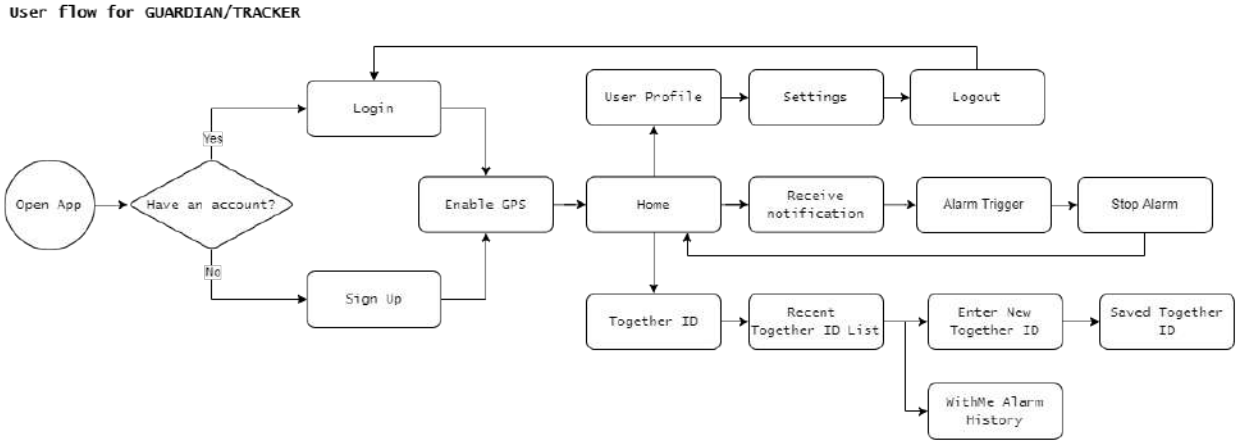


Figure 5. 2 Flowchart for Guardian role in WithMe Alarm App

The user flow starts from the app opening. Then, the user who has an account can login otherwise they have to sign up. The user has to enable GPS function in order to fully utilize the functions in the app. After that, the user can go to Home interface, User Profile interface and Together ID interface. Inside User Profile, the user can access to Settings interface where the user can logout. Inside Together ID, the user can view and manage Recent Together ID list also access WithMe Alarm History. When the WithMe Alarm triggered, the user will receive notification from their device and stop the alarm in the app.

b. User flow for Non-Guardian role

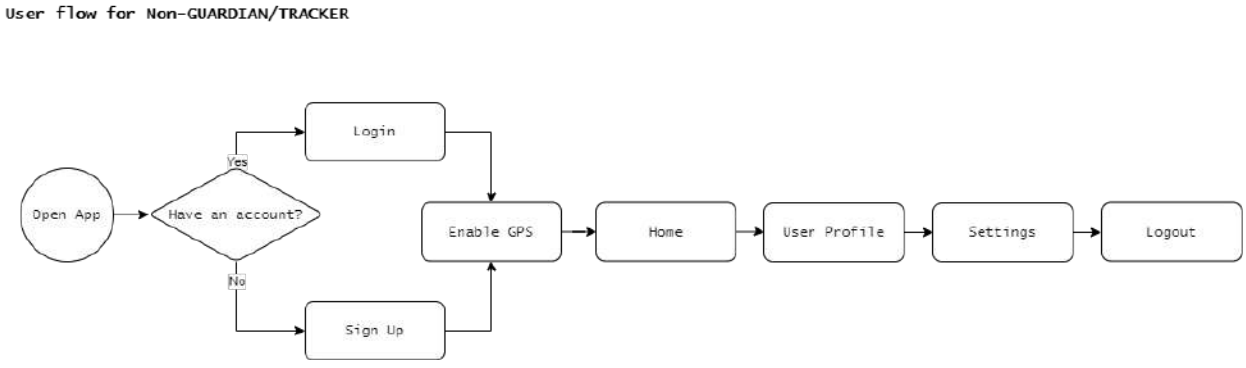


Figure 5. 3 Flowchart for Non-Guardian role in WithMe Alarm App

The user flow is the more simpler than the user flow for the guardian. This is because the system app for the role is to frequently update current location for the guardian usage.

ii. Context Diagram

- Guardian side of system interactions are user login, user sign up, user logout, manage user profile, notification and alarm trigger functions, manage recent Together ID and WithMe history.
- Non-Guardian side of system interactions are user login, user sign up, user logout, and manage user profile.

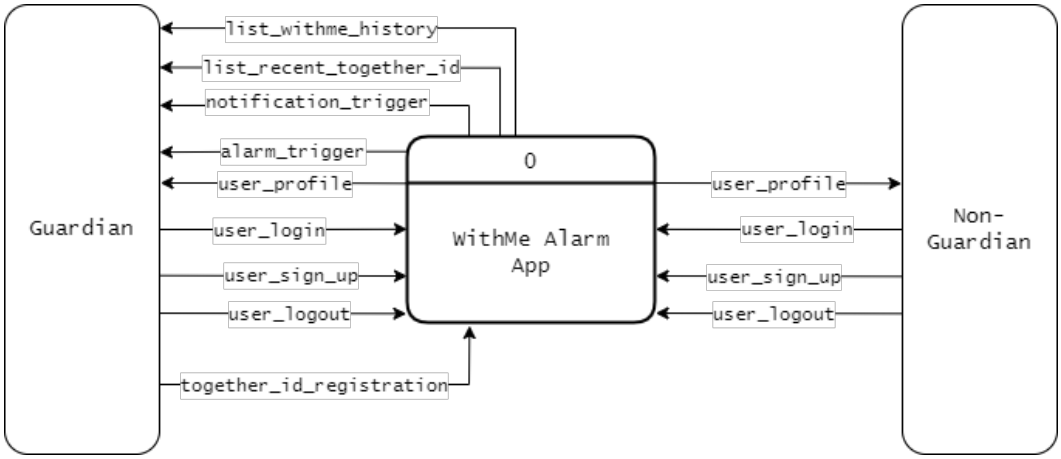


Figure 5. 4 Context Diagram for WithMe Alarm App

iii. Use Case Diagram & description

a. Use Case Diagram

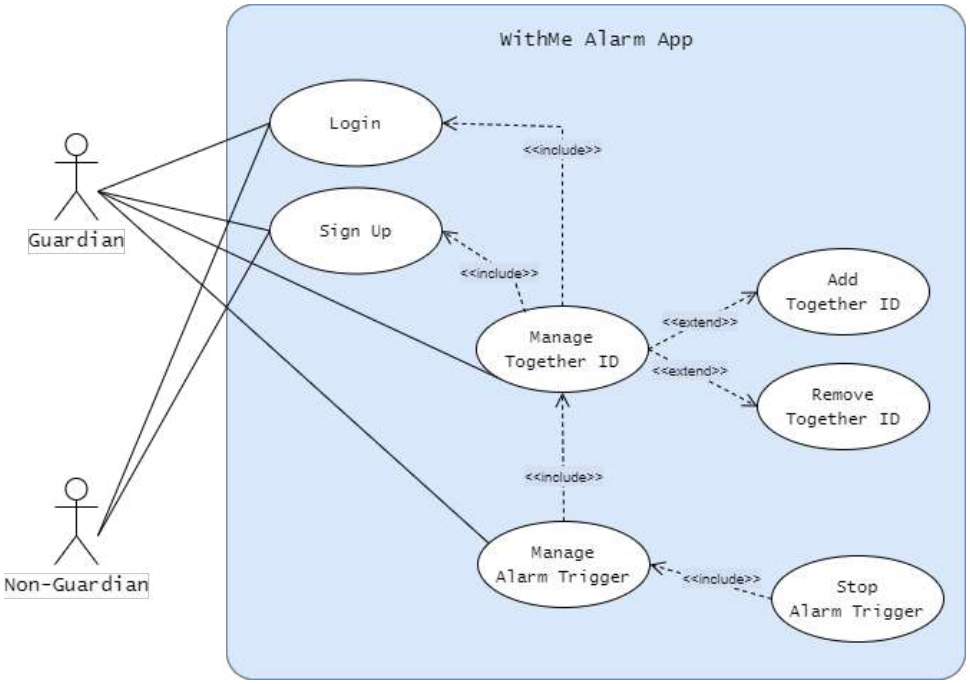


Figure 5. 5 Use Case Diagram of WithMe Alarm App

b. Use Case Description

Table 2.1 shows the description of Sign Up use case.

Use Case	Sign Up
Actors	Guardian/Non-Guardian
Summary	The guardian or non-guardian wants to create account of the app.
Precondition	No valid account registered inside database.
Description	The use case begins when the actor inserts the sign-up form with username, email, password and role. It ends when the actor is submitted the sign-up form or cancel to sign up.
Postcondition	The actor sign-up successfully. Home interface appears.
Extends	-
Includes	Manage Together ID

Table 2. 1 Sign Up Use Case

Table 2.2 shows the description of Login use case.

Use Case	Login
Actors	Guardian/Non-Guardian
Summary	The guardian or non-guardian wants to log into app.
Precondition	A valid account registered inside database.
Description	The use case begins when the actor inserts the login form with email and password. It ends when the actor is logged in or cancel to login.
Postcondition	The actor log in successfully. Home interface appears.
Extends	-
Includes	Manage Together ID

Table 2. 2 Login Use Case

Table 2.3 shows the description of Manage Together ID use case.

Use Case	Manage Together ID
Actors	Guardian
Summary	The guardian manages the registered Together ID list.
Precondition	The actor must login the app.
Description	The use case begins when the actor views the registered Together ID list. The actor can add others Together ID or remove any unwanted registered Together ID. It ends when the actor is finished viewing or cancel to manage the Together ID.
Postcondition	There will be new addition or a smaller number even no changes of Together ID in the list.
Extends	-Add Together ID and Delete Together ID
Includes	Manage Alarm Trigger

Table 2. 3 Manage Together ID Use Case

Table 2.4 shows the description of Manage Alarm Trigger use case.

Use Case	Manage Alarm Trigger
Actors	Guardian
Summary	The guardian manages the alarm trigger in the app.
Precondition	-The actors must login the app. -There is at least one registered Together ID. - The background location is always permitted. -The non-guardian is outside geofence.
Description	The use case begins when the actor be alert with the alarm trigger. It ends when the actor is stopping the alarm trigger.
Postcondition	There will be alarm history recorded.
Extends	-
Includes	Stop Alarm Trigger

Table 2. 4 Manage Alarm Trigger Use Case

iv. Activity diagram

For a start, the guardian and non-guardian users open the WithMe Alarm app. Then, the app asks the users to login if have an account otherwise, the user has to sign up for an account. After that, the app requests for background location permission and both users must accept it. In case of guardian user, they require to add a Together ID of non-guardian user to start geofence activity. Upon that, the app will check the non-guardian user's geolocation. If it is true the non-guardian user exit the geofence, the app will trigger and push an urgent notification and alarm towards both users. They can stop their alarm if it safe meanwhile the non-guardian user must send safety message which will forward to the guardian.

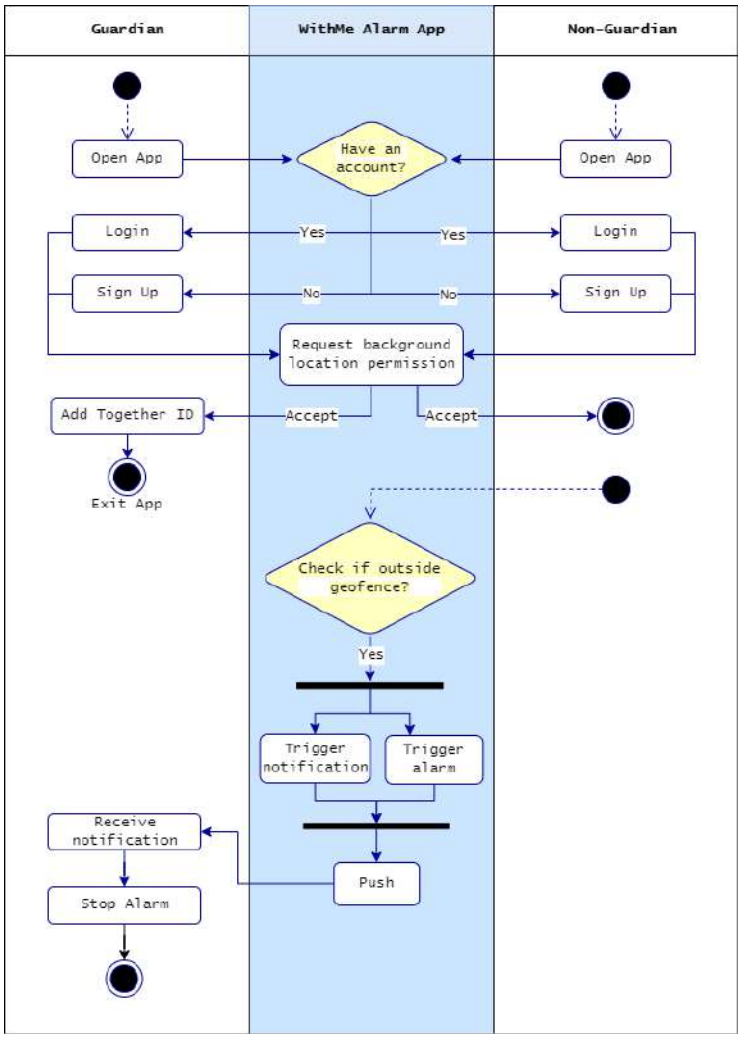


Figure 5. 6 Activity Diagram WithMe Alarm App

v. Storyboard

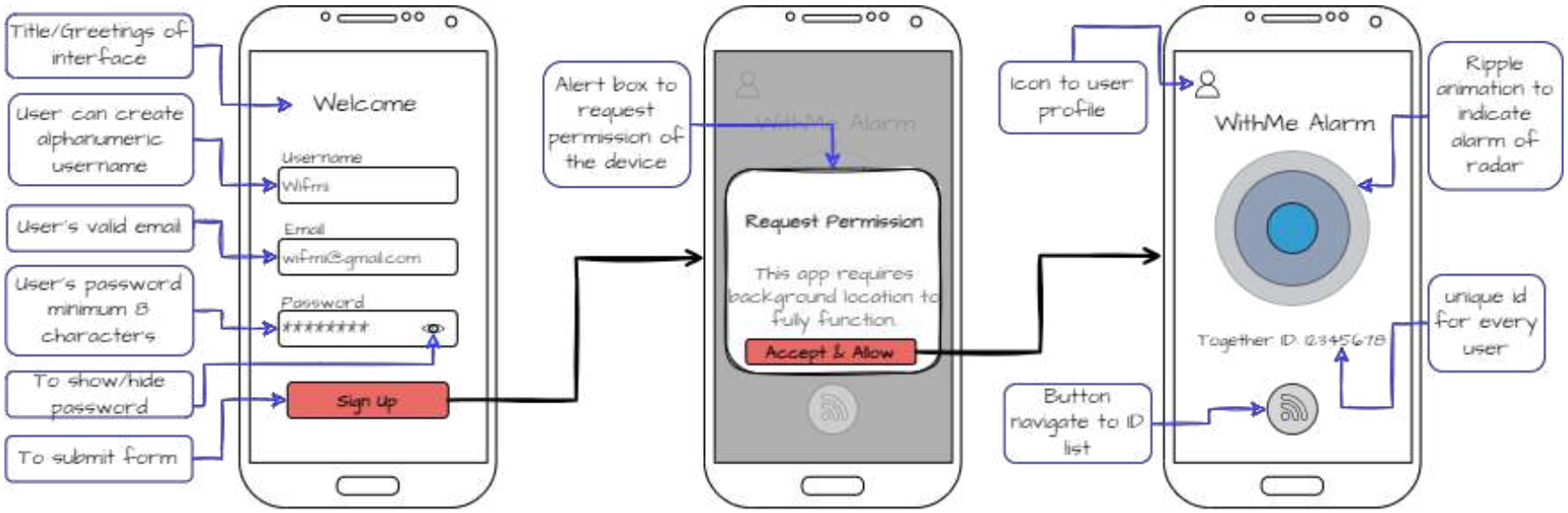


Figure 5. 7 New User Flow

Firstly, a new user will be welcomed by a Sign Up form in order to create an account. The new user has to enter with a username, an email, a password. The username can be an alphanumeric and the password length is minimum 8 characters. Then, the new user clicked on the Sign Up button to submit form. Secondly, the new user will see an alert box which request permission from the new user to accept and allow the background location so the app can be fully function. Thirdly, the new user is welcomed with the Home interface of WithMe Alarm App. From the top, there are a user profile icon, a ripple animation diagram to indicate alarm of radar, the new user's Together ID, and button to navigate to Manage WithMe Alarm interface.

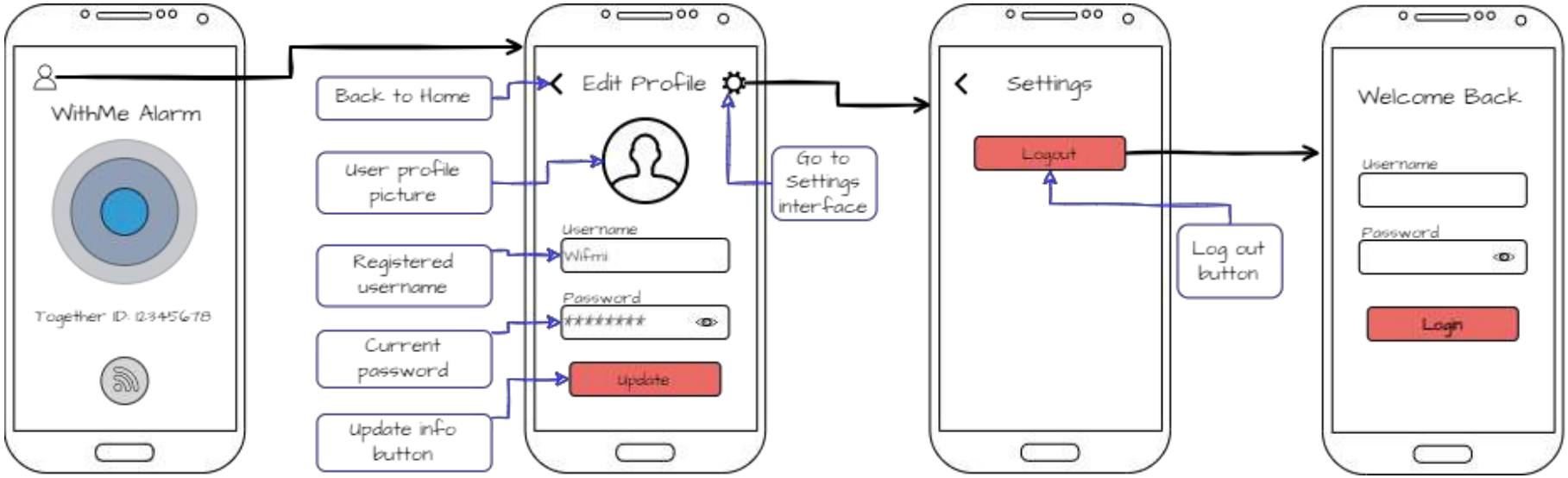


Figure 5. 8 Navigation from Home page

Fourthly, the user clicks on the user profile icon which navigate them to edit profile. Inside Edit Profile interface, there are back to home icon, Settings icon, user profile picture, user's registered username and password, also update button to submit Edit Profile form if there are changes. Fifthly, the user click on the Settings icon which navigate them to the Settings interface. From there, the user can logout by using Logout button which getting them to Login interface. The user may fill the textfield of username and password in order to log in back into the WithMe Alarm app.

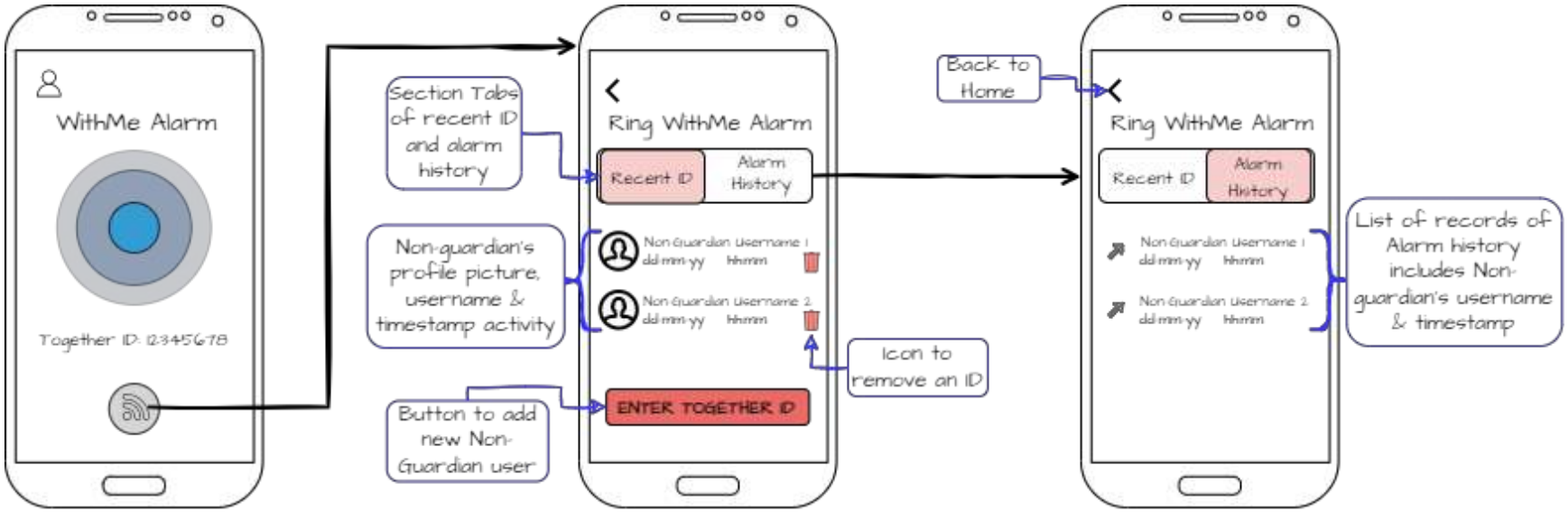


Figure 5. 9 Navigation to Manage WithMe Alarm

Sixthly, the user clicks on the Manage WithMe Alarm button on the bottom of Home page. This button will navigate the user to Manage WithMe Alarm interface. Inside of interface, there are section tabs of Recent ID and Alarm History. Seventhly, on Recent ID section contains list of Non-Guardian Together IDs entered by the user who is a Guardian. The list consists of profile picture, username, timestamp activity and delete icon to remove an ID. The user can register new Non-Guardian Together ID to monitor by clicking the ENTER TOGETHER ID button at the bottom page. Eighthly, the user can navigate to Alarm History from Recent ID section. The user can see the records of Alarm History after the alarm event happened. It contains of Non-Guardian's username and timestamp. The user can go back to Home page using the Back to Home icon located at top left corner.

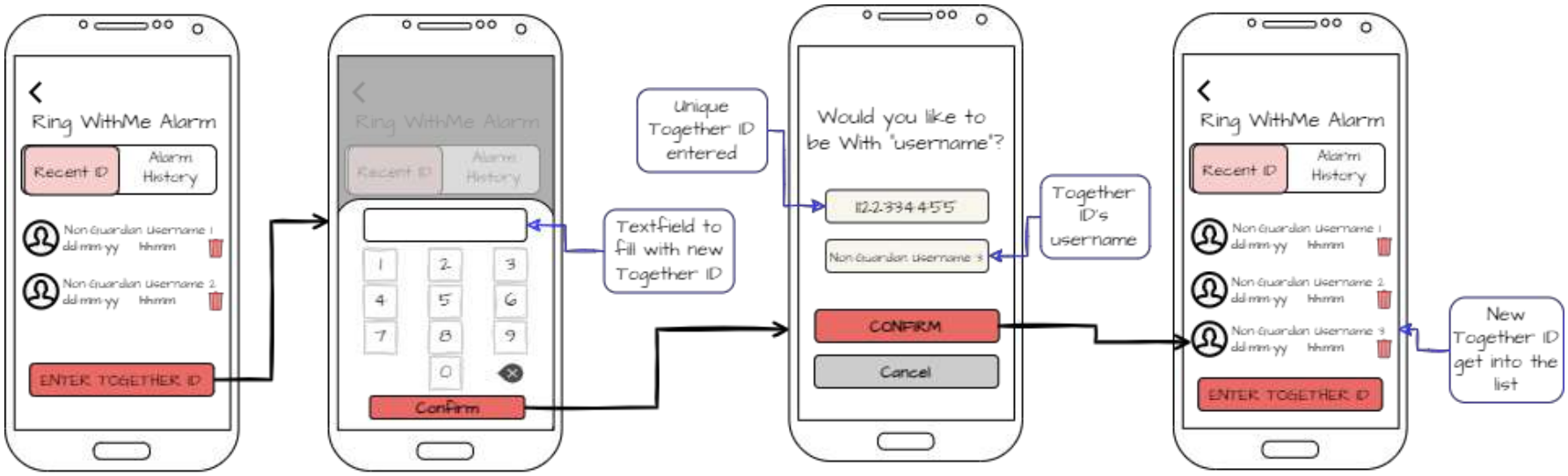


Figure 5. 10 Add New Together ID

Ninthly, the flow of adding new Non-Guardian to monitor. The user can click the ENTER TOGETHER ID button at section tab of Recent ID. There will be half page form for the user to fill with new Together ID. Their ID can be taken from the Home page as mentioned in Figure 5.7. After that, the app will ask for confirmation by Together ID entered and username. The user clicks CONFIRM button and the user will connect with the new ID. Then, new Together ID will get into the list of Recent ID. The user can go back if it is incorrect user when the user clicks the Cancel button.

3.5 Data Design

i. Entity Relationship Diagram (ERD)

The ERD in Figure 5.12 can be described as there are three collections and these are Users, Saved_togetherID and WithMeAlarm_History. The Users collection will have several documents that each of them contains data of togetherID, username, email, password and role. The Saved_togetherID collection has document with data of togetherID and username. The WithMeAlarm_History collection will contain document which data includes togetherID, dateEvent, distance, time, userBLatitude, userBLongitude and userEvent. The relationship between the three collections is one user can have many saved Together ID and each of them can have many WithMe history.

WithMe Alarm app implements NoSQL database which is Realtime Database by Firebase provider to store data and the data structure is a JSON tree. Thus, ERD of WithMe Alarm app does not have primary key and foreign key instead using node. This implies the first data of a document of a collection. In this case, the Users collection will get reference to retrieve data based on togetherID which same goes with Saved_togetherID and WithMeAlarm_History. This signifies the database more manageable as togetherID is unique for each user and will has similar purpose as primary key in SQL database.

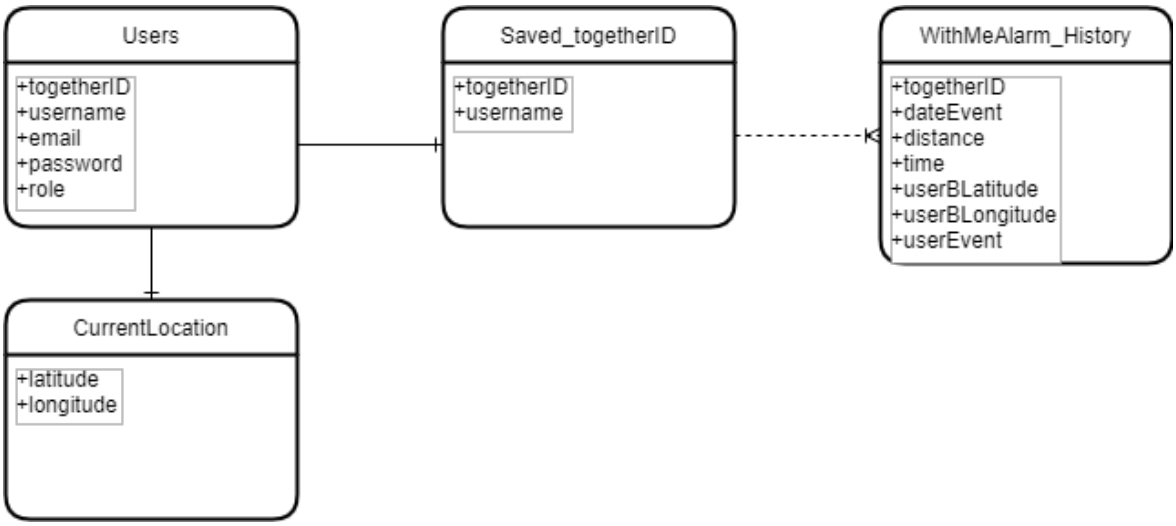


Figure 5. 11 ERD of WithMe Alarm App

ii. Database Dictionary

Table 2. 5 describes the database dictionary for Users Collection with field name, field type and descriptions.

Users Collection		
Field_name	**Field_type**	**Descriptions**
togetherID	Integer	The unique ID of the registered guardian user.
username	Text string	The username created by the registered guardian user.
email	Text string	The email to create an account for authentication.
password	Text string	The password to login the app.
role	Boolean	The authentication key for each user.

Table 2. 5 Database Dictionary Users Collection

Table 2.6 describes the database dictionary for Saved_TogetherID Collection with field name, field type and descriptions.

Saved_TogetherID Collection		
Field_name	**Field_type**	**Descriptions**
togetherID	Integer	The unique ID of the registered non-guardian user.
username	Text string	The username of the registered non-guardian user.

Table 2. 6 Database Dictionary Saved_TogetherID Collection

Table 2. 7 describes the database dictionary for WithMeAlarm_History Collection with field name, field type and descriptions.

WithMeAlarm_History Collection		
Field_name	**Field_type**	**Descriptions**
togetherID	Integer	The unique ID of the registered non-guardian user.
dateEvent	Date	The date when the alarm activity occurred.
distance	Double	The distance between user A and user B.
time	Time	The time when the alarm activity occurred.
userBLongitude	Double	User B current longitude.
userBLongitude	Double	User B current latitude.
userEvent	String	"Exit" to indicate alarm event.

Table 2. 7 Database Dictionary WithMeAlarm_History Collection

Table 2. 8 describes the database dictionary for CurrentLocation Collection with field name, field type and descriptions.

CurrentLocation		
Field_name	**Field_type**	**Descriptions**
latitude	Float	The current latitude of current login user.
longitude	Float	The current longitude of current login user.

Table 2. 8 Database Dictionary CurrentLocation Collection

3.6 Proof of Initial Concept

Figure 5. 12 WithMe Alarm Splash Screen

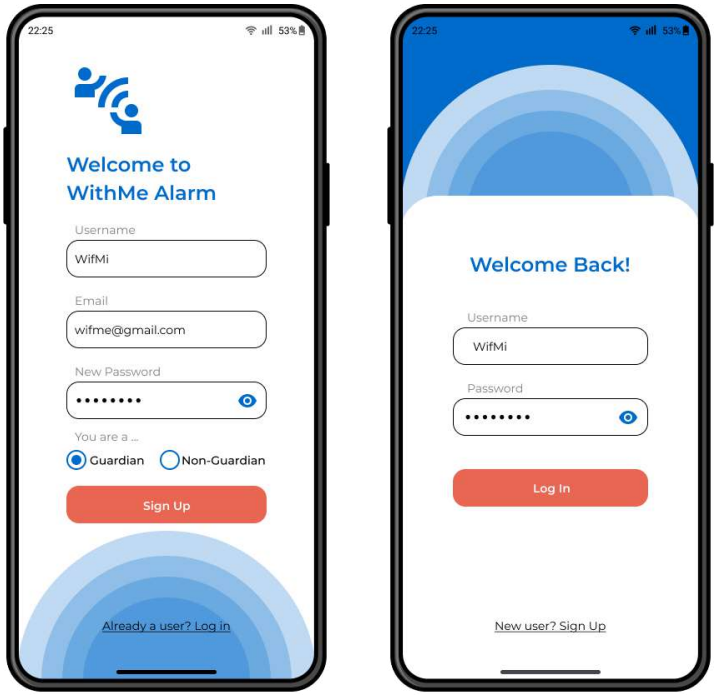


Figure 5. 13 WithMe Alarm Sign Up Interface(Left) and Login Interface(Right)

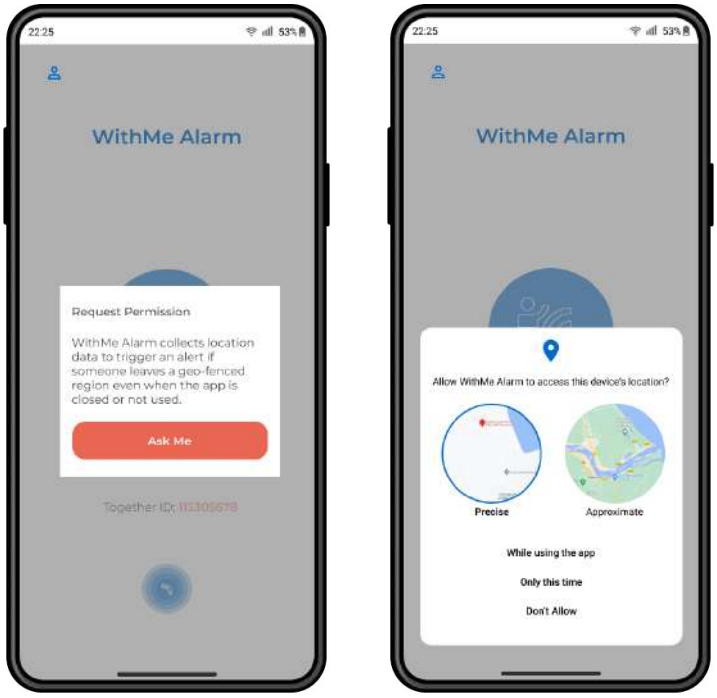


Figure 5. 14 WithMe Alarm Alert Box of Request Permission (Left) and Allow Location access options (Right)

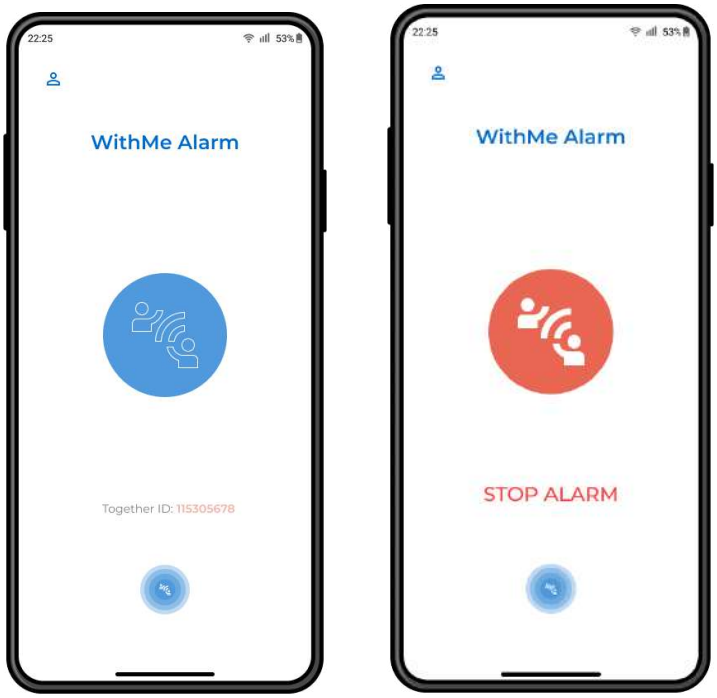


Figure 5. 15 Homepage of Normal WithMe Alarm Indicator (Left) and Warning WithMe Alarm Indicator (Right)

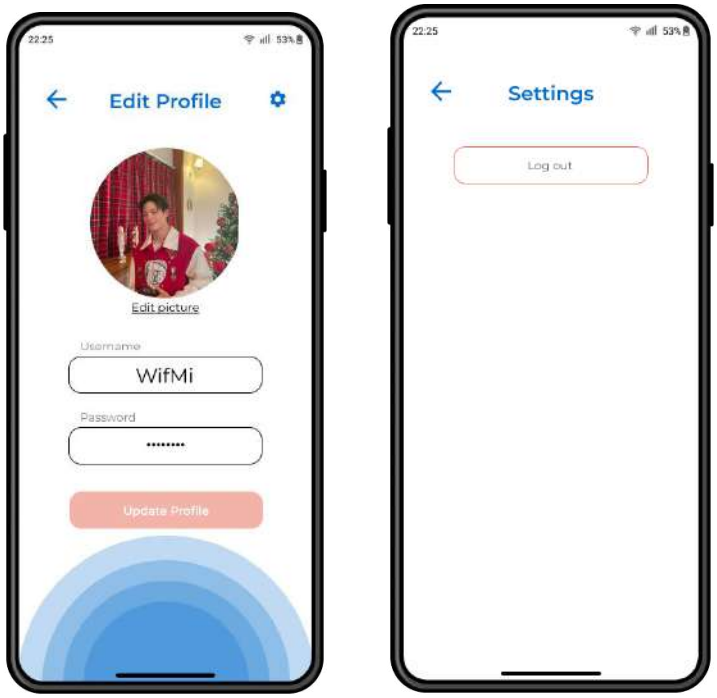


Figure 5. 16 WithMe Alarm Edit Profile Interface (Left) and Settings Interface (Right)

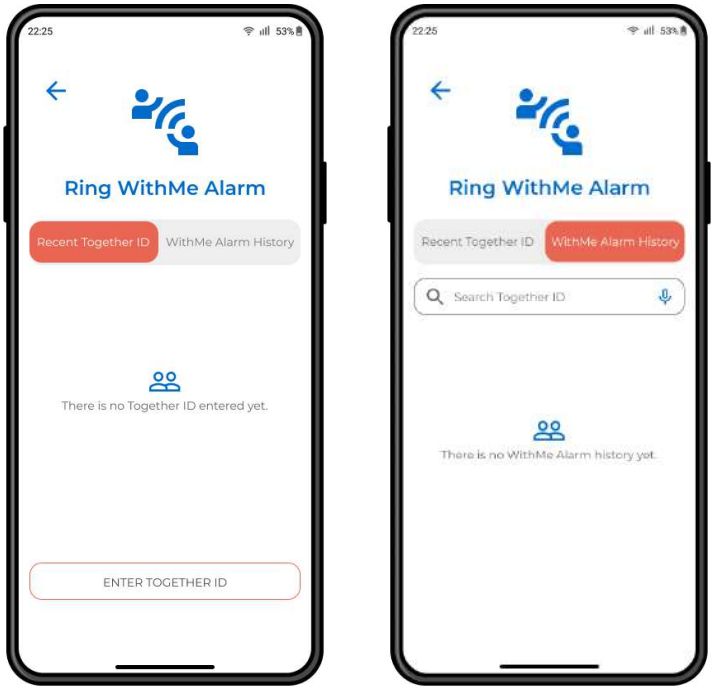


Figure 5. 17 WithMe Alarm Recent Together ID Interface (Left) and WithMe Alarm History Interface (Right)

Figure 5. 18 WithMe Alarm Add New Together ID form

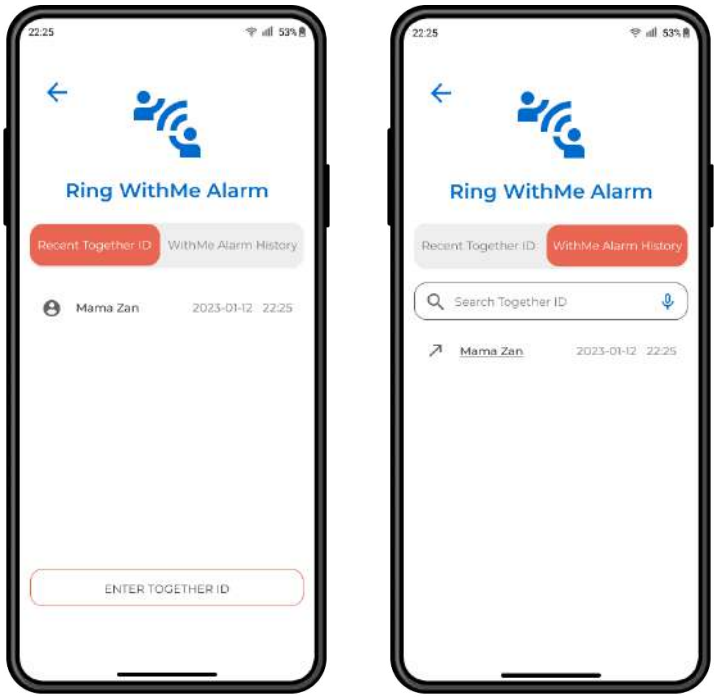


Figure 5. 19 WithMe Alarm Recent Together ID List and WithMe Alarm History Record

3.7 Testing/Validation Plan

i. Functional Test Cases. It is to ensure the functional requirements fulfill and WithMe Alarm app will be test based on Table 2.9. The test cases will run by the developer of WithMe Alarm app.

Table 2.9 shows the Functional Test Cases for WithMe Alarm App that includes Test Case ID, Test Case Objective, Prerequisite, Steps, Input Data, Expected Output, Actual Output and Status.

Test Case ID	Test Case Objective	Prerequisite	Steps	Input Data	Expected Output	Actual Output	Status	Comment
TC_01	Test sign up form	No valid account	1) Insert username 2) Insert email 3) Insert password 4) Choose role 5) Hit sign up button	Username: hana78 Email: hanahdzr@gmail.com Password: *******	Sign up successful			
TC_02	Test login form	A valid account	1) Insert email 2) Insert password 3) Hit login button	Email: hanahdzr@gmail.com Password: *******	Login successful			
TC_03	Test new Together ID form	N/A	1) Insert Together ID 2) Hit submit button	Together ID: 1536698874	Together ID saved successfully			

TC_04	Test Together ID removal	At least one saved Together ID	1) Slide left the together ID 2) Hit delete button	N/A	Together ID removed successfully			
TC_05	Test geofence alarm (Guardian user)	- Registered at least one Together ID -Turned on background location -Other device location is outside geofence	1) Click on the notification 2) Click 'Stop Alarm'	N/A	Geofence alarm successful and alarm stopped			

Table 2. 9 Functional Test Cases of WithMe Alarm App

ii. System Usability Scale(SUS). It is for the user to measure the perceived usability of WithMe Alarm app. The SUS form is as in Table 2.10 below. The form will fill by the potential users of WithMe Alarm app.

Table 2.9 shows the SUS plan for WithMe Alarm App that includes ten item questionnaires with five scales option for the respondents.

No.	SUS Questionnaire	Strongly Disagree				Strongly Agree
		1	2	3	4	5
1.	I think that I would like to use this app frequently					
2.	I found the app unnecessarily complex					
3.	I though the app was easy to use					
4.	I think I would need support of technical person to be able to use this app					
5.	I found the various functions in this app were well integrated					
6.	I thought there was too much inconsistency in this app					
7.	I would imagine that most people would learn to use the app very quickly					
8.	I found the app very awkward to use					
9.	I felt very confident using the app					
10.	I need to learn a lot of things before I could get going with this system					

Table 2. 10 System Usability Scale form for WithMe Alarm app

3.8 Potential Use of Proposed Solution

There are several potentials of WithMe Alarm app to elevate the issues in daily life as a caring guardian. This is due to minimalist user interface and user experience through the app. It is not very difficult to manage and less cost to spend timely because of the available functions built-in the WithMe Alarm app.

The potential such that is making use of the advanced technology nowadays where people own smartphone devices and the evolving technology like geofencing. Such technology will be applied inside the WithMe Alarm app which is described as a short-range distance monitor. By developing this app, it is capable to solve the problem mentioned in Problem Statement of Chapter 1 which is the guardian's burden to monitor the family members all the time.

Particularly for someone who goes travel oversea or perform Umrah with their family members. The WithMe Alarm app can be handy to avoid the family members lost from the group in unfamiliar places. Besides, there is no need constant internet connection for the WithMe Alarm app as the geofencing technology will rely on the device's GPS. Thus, this gives big prospective to WithMe Alarm app during travelling oversea if not subscribed to any cellular service.

Other than that, the WithMe Alarm app has wide potential to be commercialize towards the travel agency industry or caregiver agencies. This is due to WithMe Alarm app will be developed as a safety monitor application to help the local community with vary ages. Plus, the WithMe Alarm app is very user-friendly with less interaction but powerful to monitor people with less cost to spend. Therefore, the people lost case whether to be an average adult or senior citizen can be reduced with the help of the proposed solution.

In short, WithMe Alarm app is a simple proposed solution. The explanation above makes the WithMe Alarm app owns many potentials either in daily use or commercialization.

3.9 Gantt Chart

Figure 5.23 below shows the Gantt Chart for WithMe Alarm App project development which consists of Plan phase, Design phase, Develop phase, Integrate & Test phase and Release & Feedback phase. The project duration starts from 17 October 2022 and will be expecting to finish first sprint by 7 April 2023.

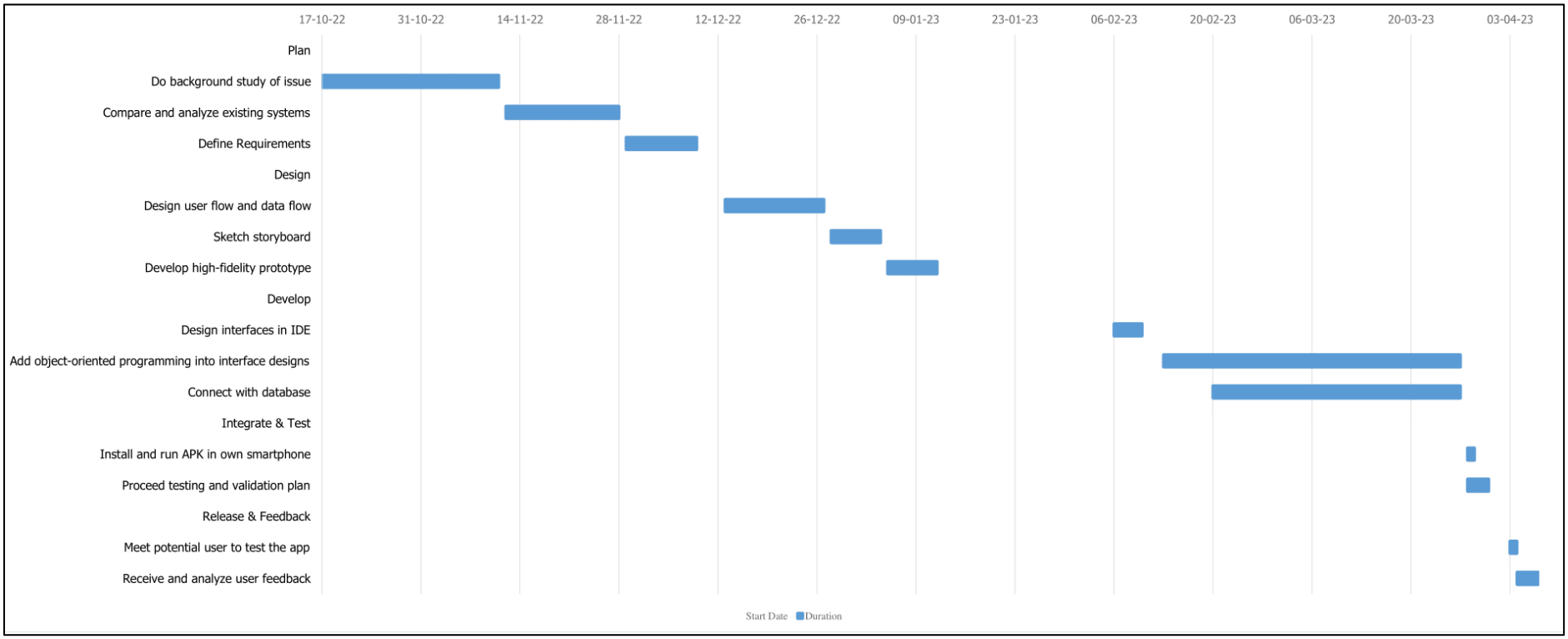


Figure 5. 20 WithMe Alarm App Project Gantt Chart

CHAPTER 4

IMPLEMENTATION, RESULT AND DISCUSSION

4.1 Introduction

This chapter discuss about the implementation of the project. This chapter contains the final product of proof of initial concept. It's also including the explanation of the mobile application development that shows the requirements of the project is fulfilled.

4.2 Implementation Process

In the previous chapter, below is the initial software requirements for the app development.

Software Requirements

- IDE (Integrated Development Environment): Android Studio Dolphin (2021.3.1)
- Database: Google Firebase

As development gradually progresses, there are several additional user permissions, libraries and plugins that are required in order to fulfill the modules requirements. These are shown in subtopic 4.2.1 of Software Requirements. For final product's interface design, geofence system implementation and database design are described in subtopic 4.2.2, 4.2.3 and 4.2.4 respectively.

4.2.1 Software Requirements

a) User permissions applied in Android Manifest file of Android Studio Project.

1. PERMISSIONS	2. PURPOSES
android.permission.INTERNET	Required for user authentication and data manipulation.
android.permission.ACCESS_COARSE_LOCATION	Required for collecting user location and geofence implementation.
android.permission.ACCESS_FINE_LOCATION	
android.permission.ACCESS_BACKGROUND_LOCATION	
android.permission.POST_NOTIFICATION	Required for triggering notification upon geofence transition.
android.permission.SCHEDULE_EXACT_ALARM	Required for triggering alarm along notification upon geofence transition equal to EXIT.

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\AndroidManifest.xml

```
 1 <?xml version="1.0" encoding="utf-8"?>
 2 <manifest xmlns:android="http://schemas.android.com/
   apk/res/android"
 3     xmlns:tools="http://schemas.android.com/tools">
 4     <!-- permissions -->
 5     <uses-permission android:name="android.permission
   .INTERNET" />
 6     <uses-permission android:name="android.permission
   .ACCESS_COARSE_LOCATION" />
 7     <uses-permission android:name="android.permission
   .ACCESS_FINE_LOCATION" />
 8     <uses-permission android:name="android.permission
   .ACCESS_BACKGROUND_LOCATION" />
 9     <uses-permission android:name="android.permission
   .POST_NOTIFICATIONS" />
10
11     <uses-permission android:name="android.permission
   .SCHEDULE_EXACT_ALARM" />
12
```

Figure 4. 7 User permissions implementation in Android Manifest file

b) Build Gradle Module

- There are two additional plugins applied in the app project which are `org.jetbrains.kotlin.android` and `com.google.gms.google-services`.
- `org.jetbrains.kotlin.android` is used because the programming language of the app currently in development is Kotlin.
- `com.google.gms.google-services` is used because the app requires dependencies from Google Play API Location in order to collect user's geolocation.
- The min SDK version that can run the app is 29 which is Android 10 due to APIs implemented for app development.
- The target SDK version is 33 due to current developed app compilation is installed on a device with Android 13 operating system.

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\build.gradle

```
plugins {
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android'
    id 'com.google.gms.google-services'
}

android {
    namespace 'com.example.withmealarm'
    compileSdk 33

    defaultConfig {
        applicationId "com.example.withmealarm"
        minSdk 29
        targetSdk 33
        versionCode 1
        versionName "1.0"

```

Figure 4. 8 Build Gradle setup

- Figure below is the list of dependencies libraries that are implemented for fulfilling the modules requirements. The dependencies are included Firebase Database and Authentication, Google Play API Location and Ripple Background Animation.

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\build.gradle

```
40 dependencies {
41
42     implementation 'androidx.core:core-ktx:1.7.0'
43     implementation 'androidx.appcompat:appcompat:1.6.
   1'
44     implementation 'com.google.android.material:
   material:1.8.0'
45     implementation 'androidx.constraintlayout:
   constraintlayout:2.1.4'
46     // Import the BoM for the Firebase platform
47     implementation platform('com.google.firebase:
   firebase-bom:31.2.1')
48     implementation 'com.google.firebase:firebase-
   database-ktx:20.2.0'
49     implementation 'com.google.firebase:firebase-auth
   -ktx:21.3.0'
50
51     testImplementation 'junit:junit:4.13.2'
52     androidTestImplementation 'androidx.test.ext:
   junit:1.1.5'
53     androidTestImplementation 'androidx.test.espresso
   :espresso-core:3.5.1'
54     //ripple animation
55     implementation 'com.skyfishjy.ripplebackground:
   library:1.0.1'
56     //google play API location
57     implementation('com.google.android.gms:play-
   services-location:21.0.1')
58     implementation('com.google.android.gms:play-
   services-maps:18.1.0')
59     implementation ('com.firebase:geofire-android:3.1
   .0')
60     //gson library to convert hashmap for db use //
   not use
61     implementation 'com.google.code.gson:gson:2.8.8'
62
63 }
```

Figure 4. 9 List of Dependencies in App Project

4.2.2 Development of Interface Design

Interface Designs of Functional Requirement

a) WithMe Alarm User Registration Module

- Product of Proof of Initial Concept

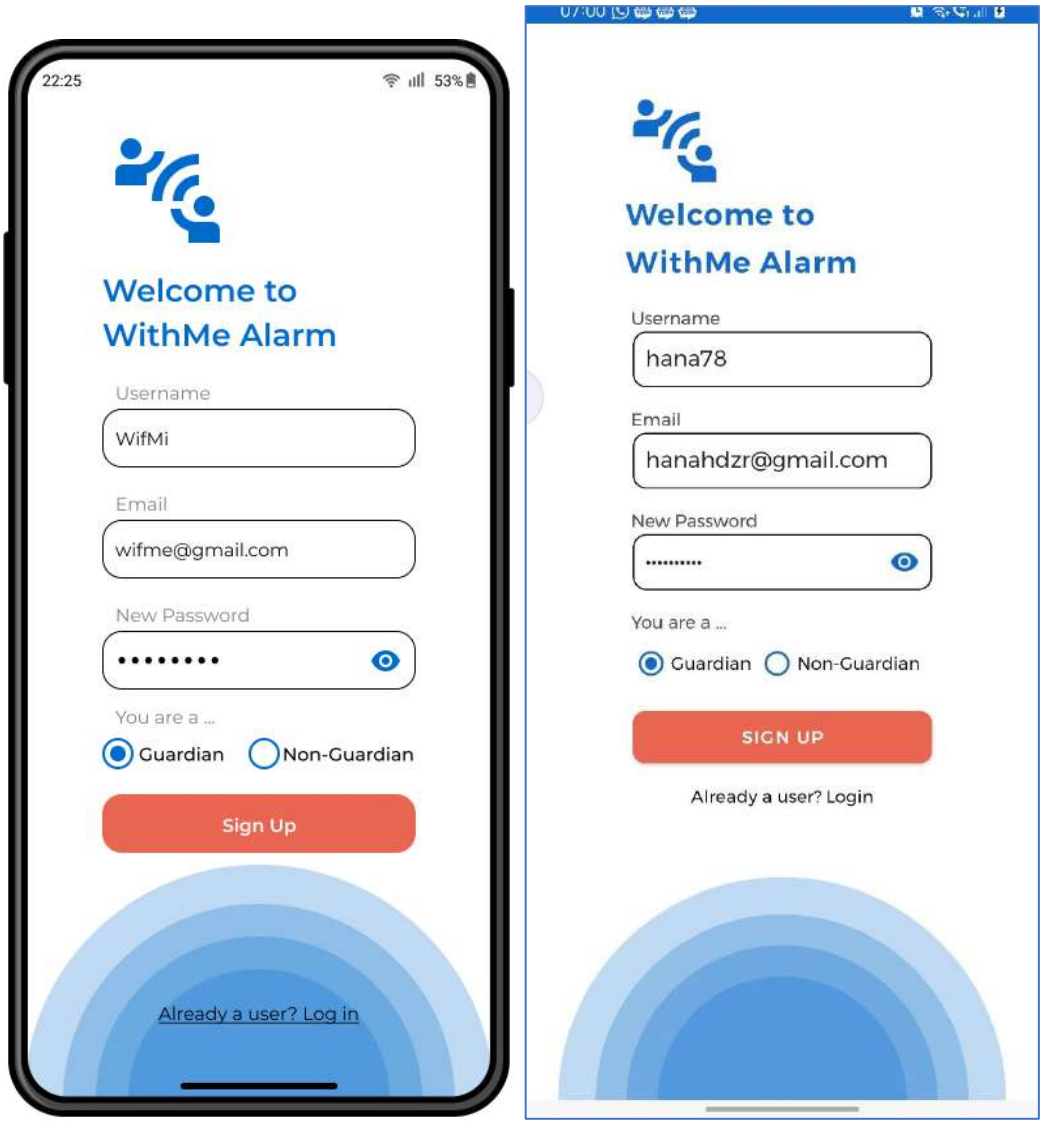


Figure 4. 10 Proof of Initial Concept (Left) and Product of User Registration Module (Right)

- Code Implementation

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\res\layout\activity_signup.xml

```
 1 <?xml version="1.0" encoding="utf-8"?>
 2 <LinearLayout xmlns:android="http://schemas.android.
   com/apk/res/android"
 3     xmlns:app="http://schemas.android.com/apk/res-
   auto"
 4     xmlns:tools="http://schemas.android.com/tools"
 5     android:layout_width="match_parent"
 6     android:layout_height="match_parent"
 7     android:orientation="vertical"
 8     tools:context=".SignUp">
 9
10     <androidx.cardview.widget.CardView
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:layout_margin="30dp"
14         app:cardElevation="0dp">
15
16         <LinearLayout
17             android:layout_width="match_parent"
18             android:layout_height="wrap_content"
19             android:layout_gravity="center_horizontal
   "
20             android:orientation="vertical"
21             android:padding="24dp">
22
23             <ImageView
24                 android:layout_width="wrap_content"
25                 android:layout_height="wrap_content"
26                 android:layout_marginStart="25dp"
27                 android:src="@drawable/
   ic_baseline_connect_without_contact_24" />
28
29             <TextView
30                 android:layout_width="wrap_content"
31                 android:layout_height="wrap_content"
32                 android:layout_marginStart="25dp"
33                 android:fontFamily="@font/
   montserrat_semibold"
34                 android:letterSpacing="0.03"
35                 android:text="Welcome to"
36                 android:textColor="@color/blue"
```

Page 1 of 5

Figure 4. 11 User Registration Module Layout Code (Page 1)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\res\layout\activity_signup.xml

```
37                  android:textSize="24dp" />
38
39              <TextView
40                  android:layout_width="wrap_content"
41                  android:layout_height="wrap_content"
42                  android:layout_marginStart="25dp"
43                  android:fontFamily="@font/
   montserrat_semibold"
44                  android:letterSpacing="0.03"
45                  android:text="WithMe Alarm"
46                  android:textColor="@color/blue"
47                  android:textSize="24dp" />
48
49              <TextView
50                  android:layout_width="wrap_content"
51                  android:layout_height="wrap_content"
52                  android:layout_marginStart="30dp"
53                  android:layout_marginTop="15dp"
54                  android:fontFamily="@font/montserrat"
55                  android:text="Username"
56                  android:textColor="@color/graythin"
57                  android:textSize="14dp" />
58
59              <EditText
60                  android:id="@+id/etUname"
61                  android:layout_width="240dp"
62                  android:layout_height="45dp"
63                  android:layout_gravity="center"
64                  android:fontFamily="@font/montserrat"
65                  android:background="@drawable/
   custom_edittext"
66                  android:inputType="textPersonName"
67                  android:padding="10dp" />
68
69              <TextView
70                  android:layout_width="wrap_content"
71                  android:layout_height="wrap_content"
72                  android:layout_marginStart="30dp"
73                  android:layout_marginTop="15dp"
74                  android:fontFamily="@font/montserrat"
75                  android:text="Email"
```

Page 2 of 5

Figure 4. 12 User Registration Module Layout Code (Page 2)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\res\layout\activity_signup.xml

```
76                 android:textColor="@color/graythin"
77                 android:textSize="14dp" />
78
79             <EditText
80                 android:id="@+id/etEmail"
81                 android:layout_width="240dp"
82                 android:layout_height="45dp"
83                 android:layout_gravity="center"
84                 android:fontFamily="@font/montserrat
   "
85                 android:background="@drawable/
   custom_edittext"
86                 android:inputType="textEmailAddress"
87                 android:padding="10dp" />
88
89             <TextView
90                 android:layout_width="wrap_content"
91                 android:layout_height="wrap_content"
92                 android:layout_marginStart="30dp"
93                 android:layout_marginTop="15dp"
94                 android:fontFamily="@font/montserrat
   "
95                 android:text="New Password"
96                 android:textColor="@color/graythin"
97                 android:textSize="14dp" />
98
99             <EditText
100                android:id="@+id/etPw"
101                android:layout_width="240dp"
102                android:layout_height="45dp"
103                android:layout_gravity="center"
104                android:fontFamily="@font/montserrat
   "
105                android:background="@drawable/
   custom_edittext"
106                android:drawableEnd="@drawable/
   ic_baseline_remove_red_eye_24"
107                android:drawablePadding="8dp"
108                android:inputType="textPassword"
109                android:padding="10dp" />
110
```

Page 3 of 5

Figure 4. 13 User Registration Module Layout Code (Page 3)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\res\layout\activity_signup.xml

```
111             <TextView
112                 android:layout_width="wrap_content"
113                 android:layout_height="wrap_content"
114                 android:layout_marginStart="30dp"
115                 android:layout_marginTop="15dp"
116                 android:fontFamily="@font/montserrat
    "
117                 android:text="You are a ..."
118                 android:textColor="@color/graythin"
119                 android:textSize="14dp" />
120
121             <RadioGroup
122                 android:layout_width="240dp"
123                 android:layout_height="45dp"
124                 android:layout_marginStart="30dp"
125                 android:orientation="horizontal"
126                 android:id="@+id/radioGroup">
127
128                 <RadioButton
129                     android:id="@+id/rbG"
130                     android:layout_width="
    wrap_content"
131                     android:layout_height="
    wrap_content"
132                     android:buttonTint="@color/blue"
133                     android:fontFamily="@font/
    montserrat"
134                     android:text="Guardian" />
135
136                 <RadioButton
137                     android:id="@+id/rbNG"
138                     android:layout_width="
    wrap_content"
139                     android:layout_height="
    wrap_content"
140                     android:layout_marginStart="5dp"
141                     android:buttonTint="@color/blue"
142                     android:fontFamily="@font/
    montserrat"
143                     android:text="Non-Guardian" />
144             </RadioGroup>
```

Page 4 of 5

Figure 4. 14 User Registration Module Layout Code (Page 4)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\res\layout\activity_signup.xml

```
145
146             <Button
147                 android:id="@+id/btnSignUp"
148                 android:layout_width="240dp"
149                 android:layout_height="wrap_content"
150                 android:layout_gravity="center"
151                 android:layout_marginTop="10dp"
152                 android:backgroundTint="@color/
    peachred"
153                 android:fontFamily="@font/
    montserrat_medium"
154                 android:padding="10dp"
155                 android:text="Sign Up"
156                 app:cornerRadius="10dp" />
157
158             <TextView
159                 android:id="@+id/loginLink"
160                 android:layout_width="wrap_content"
161                 android:layout_height="wrap_content"
162                 android:layout_gravity="center"
163                 android:layout_marginTop="10dp"
164                 android:fontFamily="@font/montserrat
    "
165                 android:textColor="@color/black"
166                 android:text="Already a user? Login"
    />
167         </LinearLayout>
168     </androidx.cardview.widget.CardView>
169
170     <ImageView
171         android:layout_width="360dp"
172         android:layout_height="360dp"
173         android:src="@drawable/component_2"
174         android:layout_gravity="center"/>
175 </LinearLayout>
```

Page 5 of 5

Figure 4. 15 User Registration Module Layout Code (Page 5)

b) WithMe Alarm Login Module

- Product of Proof of Initial Concept

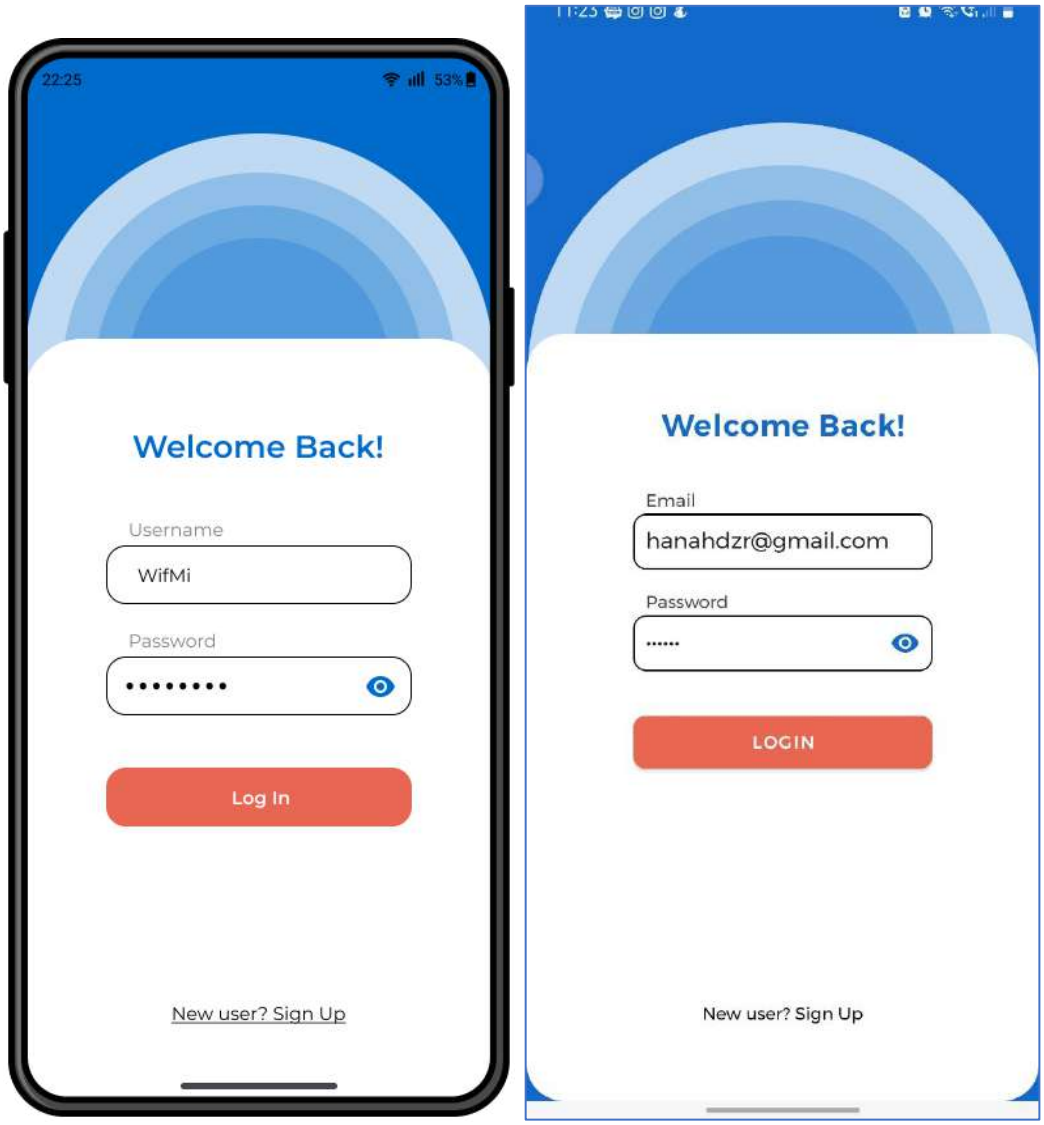


Figure 4. 16 Proof of Initial Concept (Left) and Product of Login Module (Right)

- Code Implementation

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\res\layout\activity_login.xml

```
 1 <?xml version="1.0" encoding="utf-8"?>
 2 <LinearLayout xmlns:android="http://schemas.android.
   com/apk/res/android"
 3     xmlns:app="http://schemas.android.com/apk/res-
   auto"
 4     xmlns:tools="http://schemas.android.com/tools"
 5     android:layout_width="match_parent"
 6     android:layout_height="match_parent"
 7     tools:context=".Login"
 8     android:background="@color/blue"
 9     android:orientation="vertical">
10
11     <RelativeLayout
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content">
14     <ImageView
15         android:layout_width="410dp"
16         android:layout_height="410dp"
17         android:src="@drawable/component_2"
18         android:layout_centerHorizontal="true"
19         android:layout_marginTop="80dp"/>
20
21     <androidx.cardview.widget.CardView
22         android:layout_width="match_parent"
23         android:layout_height="match_parent"
24         android:layout_marginTop="250dp"
25         app:cardCornerRadius="30dp"
26         app:cardElevation="0dp">
27
28         <LinearLayout
29             android:layout_width="wrap_content"
30             android:layout_height="wrap_content"
31             android:layout_gravity="center_horizontal
   "
32             android:orientation="vertical"
33             android:padding="24dp">
34
35             <TextView
36                 android:layout_width="wrap_content"
37                 android:layout_height="wrap_content"
38                 android:layout_marginTop="30dp"
```

Page 1 of 4

Figure 4. 17 Login Module Layout Code (Page 1)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\res\layout\activity_login.xml

```
39                  android:layout_gravity="center"
40                  android:fontFamily="@font/
   montserrat_semibold"
41                  android:letterSpacing="0.03"
42                  android:text="Welcome Back!"
43                  android:textColor="@color/blue"
44                  android:textSize="24dp" />
45
46              <TextView
47                  android:layout_width="wrap_content"
48                  android:layout_height="wrap_content"
49                  android:layout_marginStart="10dp"
50                  android:layout_marginTop="30dp"
51                  android:fontFamily="@font/montserrat"
52                  android:text="Email"
53                  android:textColor="@color/graythin"
54                  android:textSize="14dp" />
55
56              <EditText
57                  android:id="@+id/etEmail"
58                  android:layout_width="240dp"
59                  android:layout_height="45dp"
60                  android:layout_gravity="center"
61                  android:fontFamily="@font/montserrat"
62                  android:background="@drawable/
   custom_edittext"
63                  android:inputType="textEmailAddress"
64                  android:padding="10dp" />
65
66              <TextView
67                  android:layout_width="wrap_content"
68                  android:layout_height="wrap_content"
69                  android:layout_marginStart="10dp"
70                  android:layout_marginTop="15dp"
71                  android:fontFamily="@font/montserrat"
72                  android:text="Password"
73                  android:textColor="@color/graythin"
74                  android:textSize="14dp" />
75
76              <EditText
77                  android:id="@+id/etPw"
```

Page 2 of 4

Figure 4. 18 Login Module Layout Code (Page 2)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\res\layout\activity_login.xml

```
78                  android:layout_width="240dp"
79                  android:layout_height="45dp"
80                  android:layout_gravity="center"
81                  android:fontFamily="@font/montserrat
    "
82                  android:background="@drawable/
    custom_edittext"
83                  android:drawableEnd="@drawable/
    ic_baseline_remove_red_eye_24"
84                  android:drawablePadding="8dp"
85                  android:inputType="textPassword"
86                  android:padding="10dp" />
87
88              <Button
89                  android:id="@+id/btnLogin"
90                  android:layout_width="240dp"
91                  android:layout_height="wrap_content"
92                  android:layout_gravity="center"
93                  android:layout_marginTop="30dp"
94                  android:backgroundTint="@color/
    peachred"
95                  android:fontFamily="@font/
    montserrat_medium"
96                  android:padding="10dp"
97                  android:text="Login"
98                  app:cornerRadius="10dp" />
99
100             <TextView
101                 android:id="@+id/signUpLink"
102                 android:layout_width="wrap_content"
103                 android:layout_height="wrap_content"
104                 android:layout_gravity="center"
105                 android:layout_marginTop="180dp"
106                 android:fontFamily="@font/montserrat
    "
107                 android:textColor="@color/black"
108                 android:text="New user? Sign Up"/>
109         </LinearLayout>
110     </androidx.cardview.widget.CardView>
111     </RelativeLayout>
112
```

Page 3 of 4

Figure 4. 19 Login Module Layout Code (Page 3)

c) Add WithMe Alarm user ID Module

- Product of Proof of Initial Concept

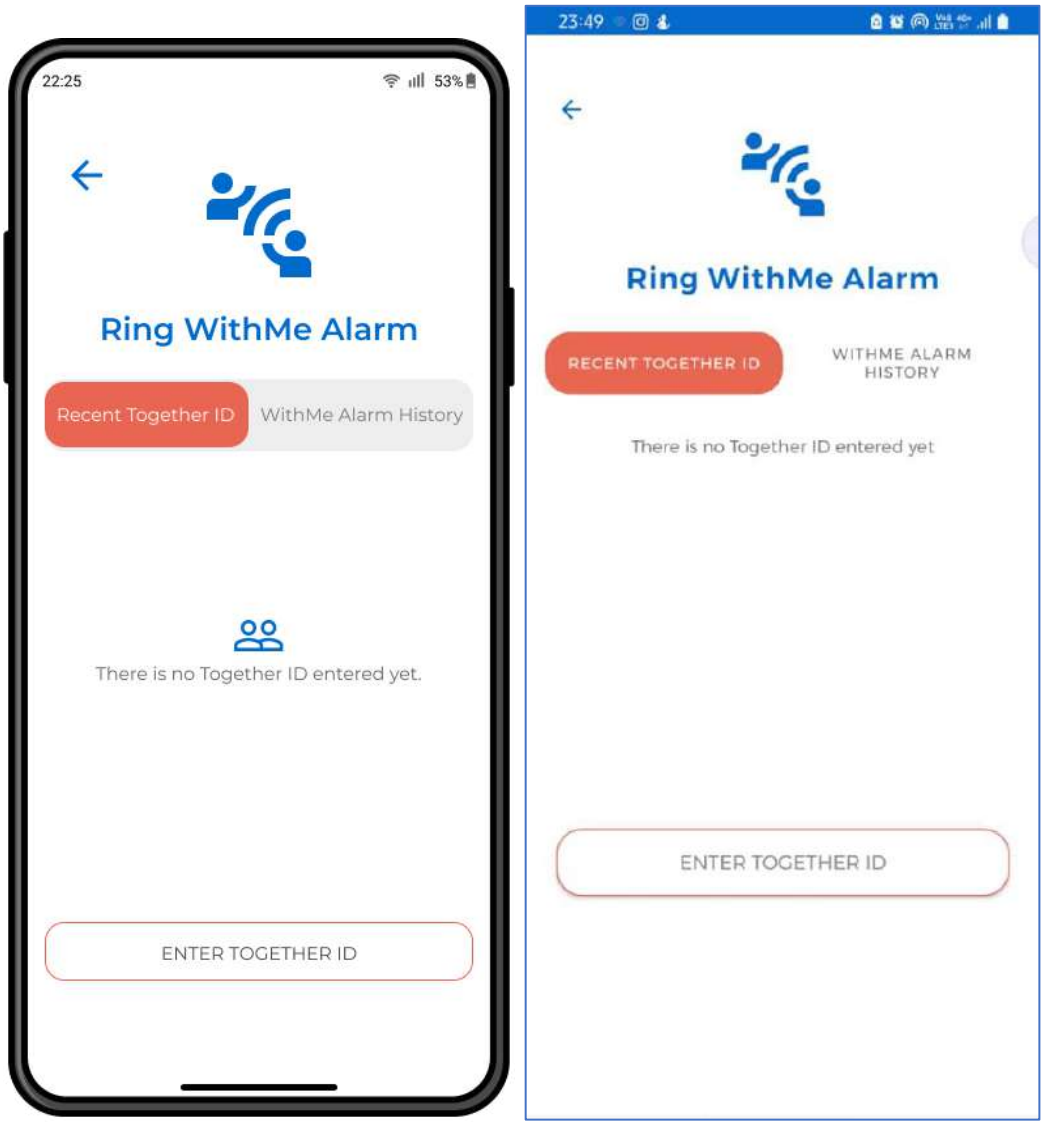


Figure 4. 20 Proof of Initial Concept (Left) and Product of Add New ID Module (Right) Part 1

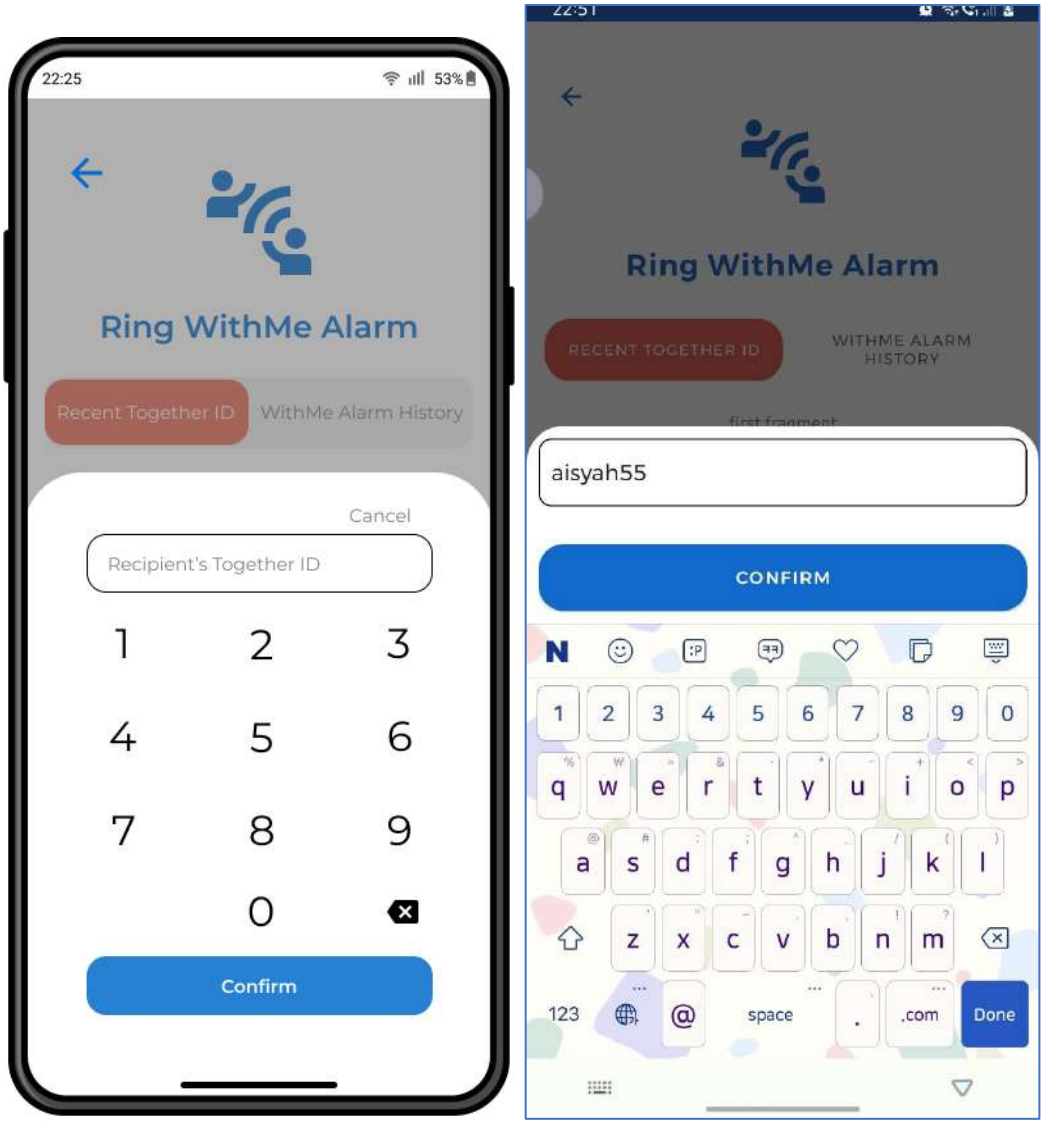


Figure 4. 21 Proof of Initial Concept (Left) and Product of Add New ID Module (Right) Part 2

- Code Implementation

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\res\layout\activity_crud_id.xml

```
 1 <?xml version="1.0" encoding="utf-8"?>
 2 <LinearLayout xmlns:android="http://schemas.android.
   com/apk/res/android"
 3     xmlns:app="http://schemas.android.com/apk/res-
   auto"
 4     xmlns:tools="http://schemas.android.com/tools"
 5     android:layout_width="match_parent"
 6     android:layout_height="match_parent"
 7     android:orientation="vertical"
 8     tools:context=".CRUD_id">
 9
10     <LinearLayout
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:layout_marginTop="48dp"
14         android:orientation="horizontal">
15
16         <ImageView
17             android:id="@+id/linkBack"
18             android:layout_width="wrap_content"
19             android:layout_height="wrap_content"
20             android:layout_marginStart="24dp"
21             android:src="@drawable/
   ic_baseline_arrow_back_24" />
22     </LinearLayout>
23
24     <ImageView
25         android:layout_width="wrap_content"
26         android:layout_height="wrap_content"
27         android:layout_gravity="center"
28         android:src="@drawable/
   ic_baseline_connect_without_contact_24" />
29
30     <TextView
31         android:layout_width="wrap_content"
32         android:layout_height="wrap_content"
33         android:layout_gravity="center"
34         android:layout_marginTop="24dp"
35         android:fontFamily="@font/montserrat_semibold
   "
36         android:letterSpacing="0.03"
```

Page 1 of 3

Figure 4. 22 Add New ID Module Layout Code (Page 1)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\res\layout\activity_crud_id.xml

```
37          android:text="Ring WithMe Alarm"
38          android:textColor="@color/blue"
39          android:textSize="24dp" />
40
41      <com.google.android.material.tabs.TabLayout
42          android:id="@+id/tabCRUD"
43          android:layout_marginTop="24dp"
44          android:layout_marginRight="15dp"
45          android:layout_marginLeft="15dp"
46          android:layout_width="match_parent"
47          android:layout_height="50dp"
48          app:tabMode="fixed"
49          app:tabIndicatorGravity="stretch"
50          app:tabIndicatorAnimationMode="elastic"
51          app:tabSelectedTextColor="@color/white"
52          app:tabTextAppearance="@style/TextAppearance.
    App.Button"
53          app:tabIndicatorColor="@null"
54          app:tabIndicator="@drawable/
    custom_tab_indicator_box"/>
55
56
57      <androidx.viewpager2.widget.ViewPager2
58          android:id="@+id/viewPager2"
59          android:layout_marginTop="24dp"
60          android:layout_marginRight="15dp"
61          android:layout_marginLeft="15dp"
62          android:layout_width="match_parent"
63          android:layout_height="wrap_content"/>
64
65      <androidx.appcompat.widget.AppCompatButton
66          android:id="@+id/btnAddId"
67          android:layout_width="match_parent"
68          android:layout_height="wrap_content"
69          android:layout_gravity="center"
70          android:layout_margin="24dp"
71          android:layout_marginTop="10dp"
72          android:background="@drawable/
    custom_btn_stroke"
73          android:fontFamily="@font/montserrat"
74          android:padding="15dp"
```

Page 2 of 3

Figure 4. 23 Add New ID Module Layout Code (Page 2)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\res\layout\activity_crud_id.xml

```
75          android:text="enter together id"
76          android:textColor="@color/graycancel"
77          android:textSize="16dp"/>
78
79  </LinearLayout>
```

Page 3 of 3

Figure 4. 24 Add New ID Module Layout Code (Page 3)

d) WithMe Alarm Trigger Module

- U_Home

  ➢ Product of Proof of Initial Concept

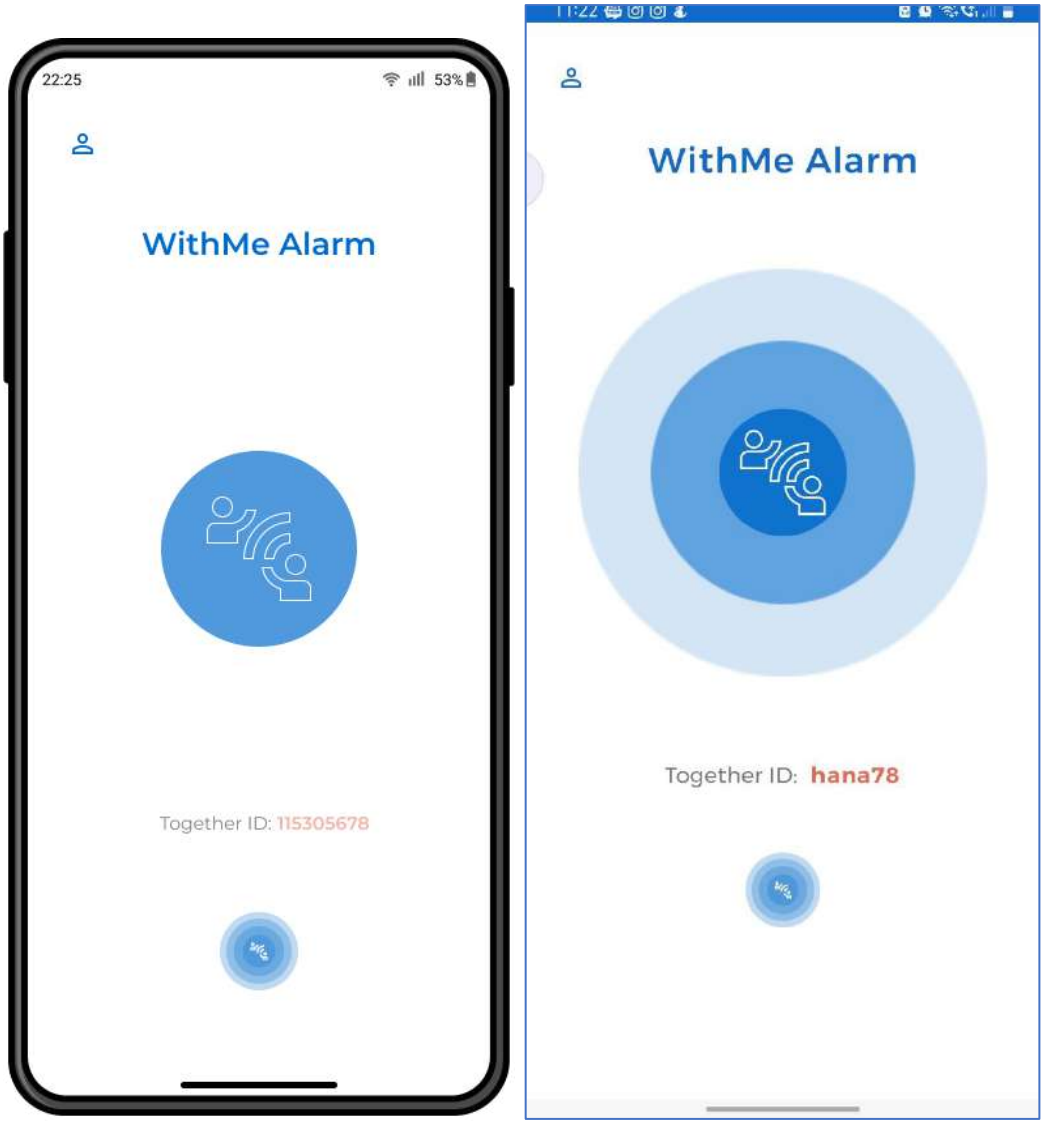


Figure 4. 25 Proof of Initial Concept (Left) and Product of Home (Right)

➢ Code Implementation

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\res\layout\activity_home.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".U_Home"
    android:orientation="vertical">

    <ImageView
        android:id="@+id/linkUserProfile"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_baseline_person_outline_24"
        android:layout_marginTop="32dp"
        android:layout_marginStart="24dp"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="WithMe Alarm"
        android:layout_marginTop="32dp"
        android:fontFamily="@font/montserrat_medium"
        android:textColor="@color/blue"
        android:textSize="28dp"
        android:letterSpacing="0.03"
        android:layout_gravity="center"/>

    <androidx.cardview.widget.CardView
        android:layout_width="410dp"
        android:layout_height="410dp"
        android:layout_gravity="center"
        android:layout_marginTop="24dp"
        android:layout_marginBottom="24dp"
        app:cardElevation="0dp">
        <com.skyfishjy.library.RippleBackground
            android:layout_width="match_parent"
            android:layout_height="match_parent"
```

Page 1 of 3

Figure 4. 26 Home Layout Code (Page 1)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\res\layout\activity_home.xml

```
39              android:id="@+id/content"
40              app:rb_color="@color/blue"
41              app:rb_radius="32dp"
42              app:rb_rippleAmount="4"
43              app:rb_duration="3000"
44              app:rb_scale="6">
45              <ImageView
46                  android:layout_width="wrap_content"
47                  android:layout_height="wrap_content"
48                  android:src="@drawable/
   ic_outline_connect_without_contact_24"
49                  android:layout_centerVertical="true"
50                  android:layout_centerHorizontal="true
   "/>
51          </com.skyfishjy.library.RippleBackground>
52      </androidx.cardview.widget.CardView>
53      <LinearLayout
54          android:layout_width="wrap_content"
55          android:layout_height="wrap_content"
56          android:layout_gravity="center">
57          <TextView
58              android:layout_width="wrap_content"
59              android:layout_height="wrap_content"
60              android:text="Together ID:"
61              android:textSize="18dp"
62              android:fontFamily="@font/montserrat" />
63          <TextView
64              android:id="@+id/tvTogetherId"
65              android:layout_width="wrap_content"
66              android:layout_height="wrap_content"
67              android:textSize="18dp"
68              android:fontFamily="@font/
   montserrat_semibold"
69              android:letterSpacing="0.05"
70              android:layout_marginStart="10dp"
71              android:textColor="@color/peachred"/>
72      </LinearLayout>
73
74      <ImageView
75          android:id="@+id/btnTogetherId"
76          android:layout_width="wrap_content"
```

Page 2 of 3

Figure 4. 27 Home Layout Code (Page 2)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\res\layout\activity_home.xml

```
77          android:layout_height="wrap_content"
78          android:src="@drawable/component_4"
79          android:layout_gravity="center"
80          android:layout_marginTop="48dp"/>
81
82  </LinearLayout>
```

Page 3 of 3

Figure 4. 28 Home Layout Code (Page 3)

4.2.3 Development of Geofence System

a) WithMe Alarm User Registration Module.

- ❖ After the user clicks on the Submit button of the Sign-Up form, the app will verify there are no empty fields. If there is, a short Toast appears to notify the user to fill up the form completely.

- ❖ After that, the data submitted will be registered into Firebase Authentication and Realtime Database. Then, the user will be navigated to Home page accordingly by their chosen role along with a short Toast appears to welcome the new user.

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\SignUp.kt

```
 1 package com.example.withmealarm
 2
 3 import android.content.ContentValues.TAG
 4 import android.content.Intent
 5 import android.os.Bundle
 6 import android.util.Log
 7 import android.widget.Toast
 8 import androidx.appcompat.app.AppCompatActivity
 9 import com.example.withmealarm.databinding.
   ActivitySignupBinding
10 import com.google.firebase.auth.FirebaseAuth
11 import com.google.firebase.auth.FirebaseUser
12 import com.google.firebase.auth.ktx.auth
13 import com.google.firebase.database.FirebaseDatabase
14 import com.google.firebase.ktx.Firebase
15
16 class SignUp : AppCompatActivity() {
17
18     private lateinit var binding:
   ActivitySignupBinding
19     private var role: String = ""
20     private lateinit var firebaseAuth: FirebaseAuth
21
22
23     override fun onCreate(savedInstanceState: Bundle
   ?) {
24         super.onCreate(savedInstanceState)
25         binding = ActivitySignupBinding.inflate(
   layoutInflater)
26         setContentView(binding.root)
27
28         //2) Initialize Firebase Auth
29         firebaseAuth = Firebase.auth
30         val user = firebaseAuth.currentUser
31
32         binding.radioGroup.setOnCheckedChangeListener
    { group, checkedId ->
33             role = if (R.id.rbG == checkedId) "1"
   else "2"  //1-guardian 2-non-g.
34             Toast.makeText(
35                 baseContext,
```

Page 1 of 5

Figure 4. 29 User Registration Module Code (Page 1)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\SignUp.kt

```
36                  role,
37                  Toast.LENGTH_SHORT,
38              ).show()
39          }
40
41          binding.btnSignUp.setOnClickListener {
42
43              //4)
44              val email = binding.etEmail.text.toString
    ()
45              val password = binding.etPw.text.toString
    ()
46              val username = binding.etUname.text.
    toString()
47
48              if (email.isNotEmpty() && password.
    isNotEmpty() && username.isNotEmpty())
49              {
50                  firebaseAuth.
    createUserWithEmailAndPassword(email, password)
51                      .addOnCompleteListener(this) {
    task ->
52                          if (task.isSuccessful) {
53                              // Sign in success,
    update UI with the signed-in user's information
54                              Log.d(TAG, "
    createUserWithEmail:success")
55                              Toast.makeText(
56                                  baseContext,
57                                  "Authentication
    succeed.",
58                                  Toast.LENGTH_SHORT,
59                              ).show()
60
61                              if(user != null)
62                              {
63                                  val uid = user.uid
64                                  //add new user into
    DB
65                                  writeNewUser(uid,
    username, email, role)
```

Page 2 of 5

Figure 4. 30 User Registration Module Code (Page 2)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\SignUp.kt

```
66                              }
67
68                              updateUI(user)
69
70                          } else {
71                              // If sign in fails,
   display a message to the user.
72                              Log.w(TAG, "
   createUserWithEmail:failure", task.exception)
73                              Toast.makeText(
74                                  baseContext,
75                                  "Authentication
   failed.",
76                                  Toast.LENGTH_SHORT,
77                              ).show()
78                              updateUI(null)
79                          }
80                      }
81              }else {
82                  Log.w(TAG, "createUserWithEmail:
   failure")
83                  Toast.makeText(this, "Fields cannot
   be empty", Toast.LENGTH_SHORT).show()
84              }
85          }
86
87          binding.loginLink.setOnClickListener {
88              val loginIntent = Intent(this, Login::
   class.java)
89              startActivity(loginIntent)
90          }
91      }
92
93      private fun updateUI(user: FirebaseUser?) {
94
95          if (user != null) {
96              if(role == "1")
97              {
98                  val homeIntent = Intent(this, U_Home
   ::class.java)
99                  startActivity(homeIntent)
```


Figure 4. 31 User Registration Module Code (Page 3)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\SignUp.kt

```
100             }
101             else
102             {
103                 val home2Intent = Intent(this,
    EU_Home::class.java)
104                 startActivity(home2Intent)
105             }
106         }
107         else
108         {
109             val loginIntent = Intent(this, Login::
    class.java)
110             startActivity(loginIntent)
111         }
112     }
113
114     //realtime db implementation
    --------------------------->
115     private fun writeNewUser( uid: String,
    togetherID: String, email: String, role: String) {
116         val user = User( togetherID, email, role)
117
118         val databaseUrl = "https://withmealarm-
    default-rtdb.asia-southeast1.firebasedatabase.app/"
119         val database = FirebaseDatabase.getInstance(
    databaseUrl)
120
121         database.getReference("Users").child(uid).
    setValue(user)
122             .addOnCompleteListener { task ->
123                 if (task.isSuccessful) {
124                     Toast.makeText(
125                         baseContext,
126                         "Welcome Aboard!",
127                         Toast.LENGTH_SHORT,
128                     ).show()
129                 }
130             }.addOnFailureListener { e ->
131             Toast.makeText(
132                 this,
133                 e.message.toString(),
```

Page 4 of 5

Figure 4. 32 User Registration Module Code (Page 4)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\SignUp.kt

```
134                 Toast.LENGTH_SHORT
135             ).show()
136         }
137     }
138
139 }
```

Figure 4. 33 User Registration Module Code (Page 5)

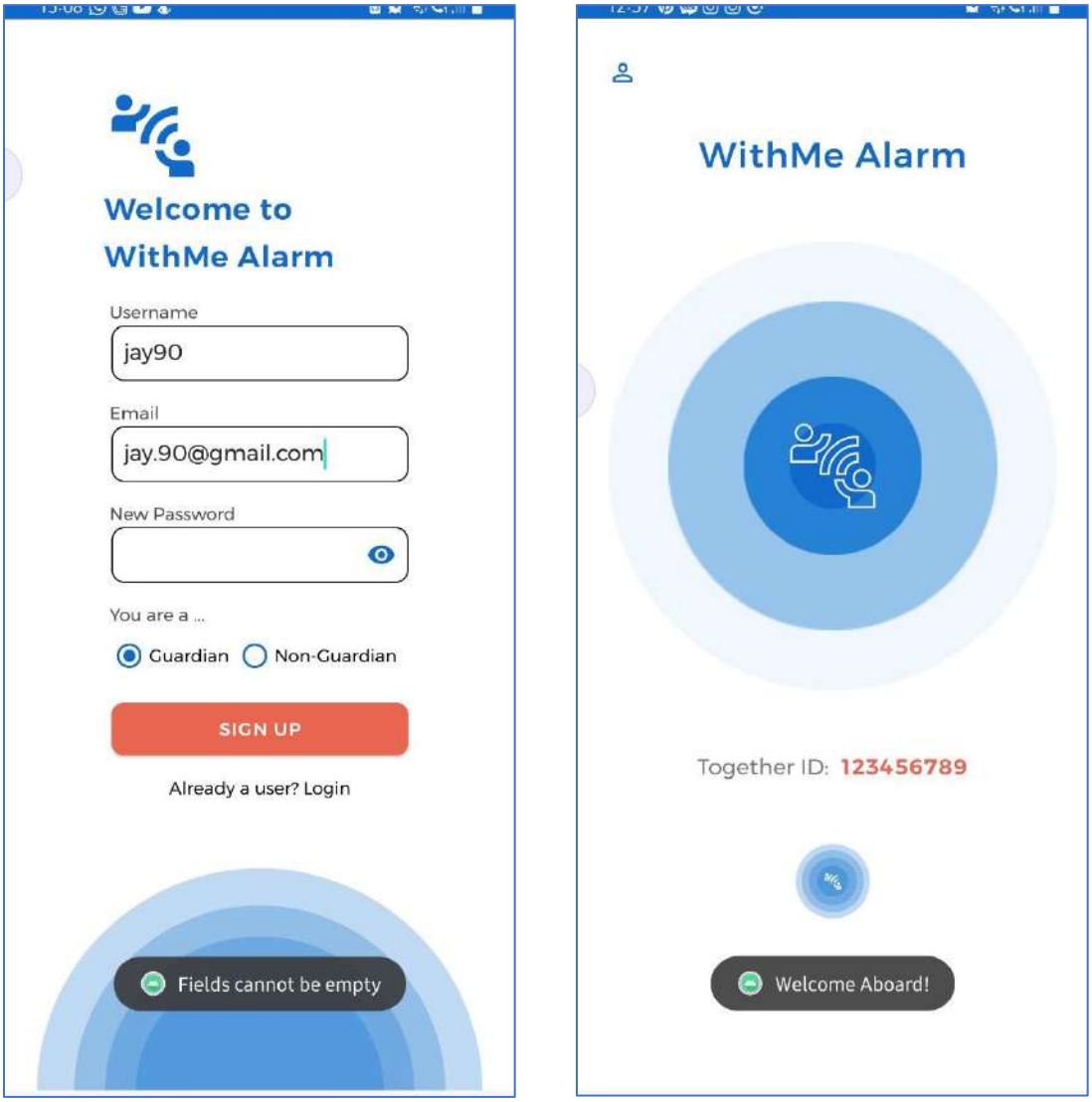


Figure 4. 34 Toast of Fields Empty (Left) and Toast of Welcome (Right)

b) WithMe Alarm Login Module

- After the user clicks on the Login button of the Login form, the app will verify there are no empty fields. If there is, a short Toast appears to notify the user to fill up the form completely.

- After that, the data will be verified by Firebase Authentication and retrieved the user unique ID (UID). By using the current user UID, the app will try to retrieve the role of the user from Firebase Realtime Database. Then, the user will be navigated to Home page accordingly by their chosen role.

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\Login.kt

```
 1 package com.example.withmealarm
 2
 3 import android.content.ContentValues.TAG
 4 import android.content.Intent
 5 import android.os.Bundle
 6 import android.util.Log
 7 import android.widget.Toast
 8 import androidx.appcompat.app.AppCompatActivity
 9 import com.example.withmealarm.databinding.
   ActivityLoginBinding
10 import com.google.firebase.auth.FirebaseAuth
11 import com.google.firebase.auth.ktx.auth
12 import com.google.firebase.database.DataSnapshot
13 import com.google.firebase.database.DatabaseError
14 import com.google.firebase.database.FirebaseDatabase
15 import com.google.firebase.database.
   ValueEventListener
16 import com.google.firebase.ktx.Firebase
17
18 class Login : AppCompatActivity() {
19
20     private lateinit var binding:
   ActivityLoginBinding
21     private lateinit var firebaseAuth: FirebaseAuth
22
23     override fun onCreate(savedInstanceState: Bundle
   ?) {
24         super.onCreate(savedInstanceState)
25         binding = ActivityLoginBinding.inflate(
   layoutInflater)
26         setContentView(binding.root)
27
28         // Initialize Firebase Auth
29         firebaseAuth = Firebase.auth
30         val user = firebaseAuth.currentUser
31         val databaseUrl = "https://withmealarm-
   default-rtdb.asia-southeast1.firebasedatabase.app/"
32         val db = FirebaseDatabase.getInstance(
   databaseUrl).getReference("Users")
33
34         binding.btnLogin.setOnClickListener {
```

Page 1 of 4

Figure 4. 35 Login Module Code (Page 1)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\Login.kt

```
35              val email = binding.etEmail.text.toString
   ()
36              val password = binding.etPw.text.toString
   ()
37
38              if (email.isNotEmpty() && password.
   isNotEmpty())
39              {
40                  firebaseAuth.
   signInWithEmailAndPassword(email,password).
   addOnCompleteListener{
41                      if (it.isSuccessful)
42                      {
43                          if(user != null) {
44                              val uid = user.uid
45
46                              val childRef = db.child(
   uid)
47
48                              // Retrieve the user's
   email address from the Realtime Database
49                              childRef.
   addValueEventListener(object : ValueEventListener {
50                                  override fun
   onDataChange(dataSnapshot: DataSnapshot) {
51
52                                      // Loop through
   the children of the DataSnapshot
53                                      for (
   childSnapshot in dataSnapshot.children) {
54                                          // Get the
   values from the child snapshot and assign them to
   variables
55                                          val role =
   dataSnapshot.child("role").getValue(String::class.
   java)
56
57                                          // Do
   something with the variables
58                                          Log.d(TAG, "
   User Role: $role")
```

Page 2 of 4

Figure 4. 36 Login Module Code (Page 2)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\Login.kt

```
59                                              updateUI(role
   )
60                                          }
61                                      }
62
63                                      override fun
   onCancelled(databaseError: DatabaseError) {
64                                          // Handle
   database error
65                                      }
66                                  })
67                              }
68
69                              Toast.makeText(this, "Login
   Successfully", Toast.LENGTH_SHORT).show()
70                          }else {
71                              Toast.makeText(this, it.
   exception.toString(), Toast.LENGTH_SHORT).show()
72                          }
73                      }
74                  }else {
75                      Toast.makeText(this, "Fields cannot
   be empty", Toast.LENGTH_SHORT).show()
76                  }
77              }
78
79              binding.signUpLink.setOnClickListener {
80                  val signUpIntent = Intent(this, SignUp::
   class.java)
81                  startActivity(signUpIntent)
82              }
83          }
84
85          private fun updateUI(role: String?) {
86
87              if (role == "1") {
88                  val homeIntent = Intent(this, U_Home::
   class.java)
89                  startActivity(homeIntent)
90                  finish()
91              } else {
```


Figure 4. 37 Login Module Code (Page 3)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\Login.kt

```
92              val home2Intent = Intent(this, EU_Home::
   class.java)
93              startActivity(home2Intent)
94              finish()
95          }
96      }
97  }
```

Page 4 of 4

Figure 4. 38 Login Module Code (Page 4)

c) Add WithMe Alarm user ID Module

- When the user clicks on the ENTER TOGETHER ID button, a dialog window appears. The dialog is used as a form for the user to enter the end-user's Together ID. It is to collect end-user's current location by their ID in order to monitor them within geofence area.

- After the user submits the form, the data is passed by intent to an extension code file called `Connect Id`. It is to save the end-user's Together ID as user's collection in database.

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\CRUD_id.kt

```
 1 package com.example.withmealarm
 2
 3 import android.app.Activity
 4 import android.app.Dialog
 5 import android.content.Intent
 6 import android.graphics.Color
 7 import android.graphics.drawable.ColorDrawable
 8 import android.os.Bundle
 9 import android.view.Gravity
10 import android.view.ViewGroup
11 import android.view.ViewGroup.LayoutParams
12 import android.view.Window
13 import android.widget.Button
14 import android.widget.EditText
15 import androidx.appcompat.app.AppCompatActivity
16 import androidx.viewpager2.widget.ViewPager2
17 import com.google.android.material.tabs.TabLayout
18 import com.google.android.material.tabs.TabLayout.Tab
19
20 class CRUD_id : AppCompatActivity() {
21
22     private lateinit var tabLayout: TabLayout
23     private lateinit var viewPager2: ViewPager2
24     private lateinit var adapter: FragmentPageAdapter
25
26     companion object {
27         const val REQUEST_CODE_CUSTOM_DIALOG = 1
28         const val EXTRA_NEW_TOGETHER_ID = "com.
   example.withmealarm.NEW_TOGETHER_ID"
29     }
30
31
32     override fun onCreate(savedInstanceState: Bundle
   ?) {
33         super.onCreate(savedInstanceState)
34         setContentView(R.layout.activity_crud_id)
35
36         val btnAddNewId = findViewById<Button>(R.id.
   btnAddId)
37
38         btnAddNewId.setOnClickListener { view->
```

Page 1 of 4

Figure 4. 39 Add New ID Module Code (Page 1)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\CRUD_id.kt

```
39              showCrudDialog()
40          }
41
42
43          tabLayout = findViewById(R.id.tabCRUD)
44          viewPager2 = findViewById(R.id.viewPager2)
45
46          //initialize
47          adapter = FragmentPageAdapter(
    supportFragmentManager, lifecycle)
48
49          tabLayout.addTab(tabLayout.newTab().setText("
    Recent Together ID"))
50          tabLayout.addTab(tabLayout.newTab().setText("
    WithMe Alarm History"))
51
52          //send above initialization into viewpager2
53          viewPager2.adapter = adapter
54
55          tabLayout.addOnTabSelectedListener(object :
    TabLayout.OnTabSelectedListener{
56              override fun onTabSelected(tab: Tab?) {
57                  if (tab != null) {
58                      viewPager2.currentItem = tab.
    position
59                  }
60              }
61
62              override fun onTabUnselected(tab: Tab?) {
63
64              }
65
66              override fun onTabReselected(tab: Tab?) {
67
68              }
69
70          })
71
72          viewPager2.registerOnPageChangeCallback(
    object : ViewPager2.OnPageChangeCallback(){
73              override fun onPageSelected(position: Int
```


Figure 4. 40 Add New ID Module Code (Page 2)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\CRUD_id.kt

```
73 ) {
74                 super.onPageSelected(position)
75                 tabLayout.selectTab(tabLayout.
   getTabAt(position))
76             }
77         })
78 
79     }
80 
81     private fun showCrudDialog()
82     {
83         val dialog = Dialog(this)
84         dialog.requestWindowFeature(Window.
   FEATURE_NO_TITLE)
85         dialog.setContentView(R.layout.
   activity_crud_bottom_dialog)
86 
87         val btnSubmitNewId = dialog.findViewById<
   Button>(R.id.btnConfirm)
88 
89         btnSubmitNewId.setOnClickListener {
90             val newId = dialog.findViewById<EditText
   >(R.id.etTogetherID).text.toString()
91             val intent = Intent()
92             intent.putExtra(EXTRA_NEW_TOGETHER_ID,
   newId)
93             dialog.dismiss()
94             setResult(Activity.RESULT_OK, intent)
95             // Create an Intent to start Activity B
96             val intentB = Intent(this, ConnectId::
   class.java)
97             intentB.putExtra(EXTRA_NEW_TOGETHER_ID,
   newId)
98             startActivity(intentB)
99         }
100 
101 
102         /*
103         dialog.setOnDismissListener {
104             finish()
105         }*/
```

Page 3 of 4

Figure 4. 41 Add New ID Module Code (Page 3)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\CRUD_id.kt

```
106
107         dialog.show()
108         dialog.window?.setLayout(ViewGroup.
    LayoutParams.MATCH_PARENT,LayoutParams.WRAP_CONTENT)
109         dialog.window?.setBackgroundDrawable(
    ColorDrawable(Color.TRANSPARENT))
110         dialog.window?.attributes?.windowAnimations
     = R.style.DialogAnimation
111         dialog.window?.setGravity(Gravity.BOTTOM)
112
113         //startActivityForResult(intent,
    REQUEST_CODE_CUSTOM_DIALOG)
114     }
115
116     override fun onActivityResult(requestCode: Int,
    resultCode: Int, data: Intent?) {
117         super.onActivityResult(requestCode,
    resultCode, data)
118         if (requestCode ==
    REQUEST_CODE_CUSTOM_DIALOG && resultCode == Activity
    .RESULT_OK) {
119             val newId = data?.getStringExtra(
    EXTRA_NEW_TOGETHER_ID)
120             // Do something with the new ID here
121
122         }
123     }
124
125
126 }
```


Figure 4. 42 Add New ID Module Code (Page 4)

- Connect Id Extension File.
    - The app initializes the current user UID from Firebase Authentication which is for saving the end-user's Together ID under the user UID.
    - Then, the app will find the UID of end-user's Together ID by attaching a listener to loop through the child nodes to find UID for child node that contains the end-user's Together ID.
    - If there is a match, the app will retrieve or snapshot the specific data needed of the end-user to save under user UID.

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\ConnectId.kt

```
 1 package com.example.withmealarm
 2
 3 import android.content.Intent
 4 import android.os.Bundle
 5 import android.util.Log
 6 import android.widget.Toast
 7 import androidx.appcompat.app.AppCompatActivity
 8 import com.google.android.gms.location.
   FusedLocationProviderClient
 9 import com.google.android.gms.location.
   GeofencingClient
10 import com.google.firebase.auth.FirebaseAuth
11 import com.google.firebase.auth.ktx.auth
12 import com.google.firebase.database.DataSnapshot
13 import com.google.firebase.database.DatabaseError
14 import com.google.firebase.database.FirebaseDatabase
15 import com.google.firebase.database.
   ValueEventListener
16 import com.google.firebase.ktx.Firebase
17
18 //this page is to saved new id
19 class ConnectId : AppCompatActivity() {
20
21     lateinit var fusedLocationProviderClient:
   FusedLocationProviderClient
22     lateinit var geofencingClient: GeofencingClient
23     private lateinit var firebaseAuth: FirebaseAuth
24
25     companion object {
26         const val EXTRA_NEW_TOGETHER_ID = "com.
   example.withmealarm.NEW_TOGETHER_ID"
27     }
28
29     fun onDialogResult(newId: String) {
30         // Do something with the new ID here
31
32     }
33
34     override fun onCreate(savedInstanceState: Bundle
   ?) {
35         super.onCreate(savedInstanceState)
```

Page 1 of 5

Figure 4. 43 Connect ID Extension Code (Page 1)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\ConnectId.kt

```
36          setContentView(R.layout.activity_connect_id)
37
38          val newId = intent.getStringExtra(
    EXTRA_NEW_TOGETHER_ID)
39          if (newId != null) {
40              // Do something with the newId value here
41          }
42          val usernameToFind = newId
43          Log.d("TAG", "retrieve $newId")
44
45          firebaseAuth = Firebase.auth
46          val user = firebaseAuth.currentUser
47          val m_uid = user?.uid
48          val convUid = m_uid.toString()
49
50          //db setup
51          val databaseUrl = "https://withmealarm-
    default-rtdb.asia-southeast1.firebasedatabase.app/"
52
53          // Get a reference to the child node where
    the username is stored
54          val db = FirebaseDatabase.getInstance(
    databaseUrl).getReference("Users")
55
56          //find uid by username
57          // Attach a listener to read the data stored
    in the node
58          db.addListenerForSingleValueEvent(object :
    ValueEventListener {
59              override fun onDataChange(snapshot:
    DataSnapshot) {
60                  var uid: String? = null
61                  // Loop through the child nodes under
     the "Users" node to find the UID for the child node
    that contains the entered username
62                  for (childSnapshot in snapshot.
    children) {
63                      val username = childSnapshot.
    child("togetherID").getValue(String::class.java)
64                      if (username == usernameToFind) {
65                          uid = childSnapshot.key
```


Figure 4. 44 Connect ID Extension Code (Page 2)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\ConnectId.kt

```
66                          Log.d("TAG", "match $
   username")
67
68                          if (uid != null) {
69                              // A matching UID was
   found, use it to query the child nodes under that
   UID
70                              val uidRef = db.child(
   uid)
71                              uidRef.
   addListenerForSingleValueEvent(object :
   ValueEventListener {
72                                  override fun
   onDataChange(snapshot: DataSnapshot) {
73                                      // Loop through
   the child nodes under the UID to retrieve the
   desired information
74                                      for (
   childSnapshot in snapshot.children) {
75                                          val value =
   childSnapshot.getValue(String::class.java) //error
   here
76                                          Log.d("TAG"
   , "$childSnapshot.key: $value")
77                                          writeNewId(
   convUid, uid, username)
78                                      }
79                                  }
80
81                                  override fun
   onCancelled(error: DatabaseError) {
82                                      Log.d("TAG", "
   Database error occurred: ${error.message}")
83                                  }
84                              })
85                          } else {
86                              // No matching UID was
   found, display an error message to the user
87                              Log.d("TAG", "Username
   not found!")
88                          }
```

Page 3 of 5

Figure 4. 45 Connect ID Extension Code (Page 3)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\ConnectId.kt

```
 89                     }
 90                     else {
 91                         Log.d("TAG", "not match1 $
    usernameToFind")
 92                         Log.d("TAG", "not match2 $
    username")
 93                     }
 94                 }
 95
 96             }
 97             override fun onCancelled(error:
    DatabaseError) {
 98                 Log.d("TAG", "Database error
    occurred: ${error.message}")
 99             }
100         })
101     }
102     //realtime db implementation
    --------------------------->
103     private fun writeNewId( convUid:String, euid:
    String, togetherID: String?) {
104         val newId = Saved_Id(euid, togetherID)
105
106         val databaseUrl = "https://withmealarm-
    default-rtdb.asia-southeast1.firebasedatabase.app/"
107         val database = FirebaseDatabase.getInstance(
    databaseUrl)
108         val existingNodeReference  = database.
    getReference("Users")
109
110         val savedIdMap = HashMap<String, Any>()
111         savedIdMap["savedEuid"] = newId
112
113         existingNodeReference.child(convUid).child("
    Saved_togetherID").setValue(savedIdMap)
114             .addOnCompleteListener { task ->
115                 if (task.isSuccessful) {
116                     Toast.makeText(
117                         baseContext,
118                         "Saved New Id",
119                         Toast.LENGTH_SHORT,
```

Page 4 of 5

Figure 4. 46 Connect ID Extension Code (Page 4)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\ConnectId.kt

```
120                     ).show()
121
122                     //home
123                     val homeIntent = Intent(this,
    U_Home::class.java)
124                     startActivity(homeIntent)
125                     finish()
126                 }
127             }.addOnFailureListener { e ->
128                 Toast.makeText(
129                     this,
130                     e.message.toString(),
131                     Toast.LENGTH_SHORT
132                 ).show()
133
134                 //back to CRUD_id page
135                 val cancelIntent = Intent(
    applicationContext, CRUD_id::class.java)
136                 startActivity(cancelIntent)
137
138                 finish()
139             }
140     }
141
142     //auth
143     public override fun onStart() {
144         super.onStart()
145         //3) Check if user is signed in (non-null)
    and update UI accordingly.
146         val user = Firebase.auth.currentUser
147         if (user != null) {
148             // User is signed in
149         } else {
150             // No user is signed in
151             val loginIntent = Intent(this, Login::
    class.java)
152             startActivity(loginIntent)
153         }
154     }
155 }
```


Figure 4. 47 Connect ID Extension Code (Page 5)

d) WithMe Alarm Trigger Module

❖ This module requires three separate files which are the U_Home(user's Home page), Geofence Broadcast Receiver and Notification Trigger.

i. U_Home

➢ The geofence setup, notification and alarm trigger setup are inside user's Home page instead of end-user.

➢ Every time the user logs in; the app will collect the current location of the current user to save them in the database. At the same time, the app will use the current location to create or update geofence area.

➢ If the user is new, the app will ask for user permission on location access. Therefore, the user has to allow precise location tracker as shown in Figure 4.42.

➢ Using the saved end-user's Together ID from database, the app will monitor the geofence transition of them. There are geofence transition enter and exit. Both can trigger notification to the current logged in user when there is transition. Besides, the alarm will trigger when the end-user exits the geofence setup by user.

➢ Every transition will call the `getGeofencingRequest` method which is Geofence Broadcast Receiver file.

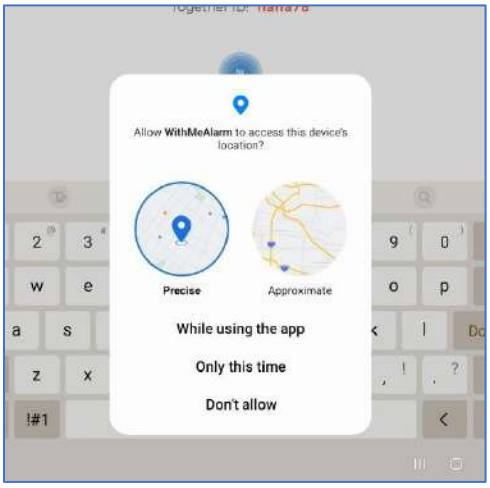


Figure 4. 48 Location Access Permission Dialog Box

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\U_Home.kt

```
 1 package com.example.withmealarm
 2
 3 import android.Manifest
 4 import android.annotation.TargetApi
 5 import android.app.PendingIntent
 6 import android.content.Intent
 7 import android.content.IntentSender
 8 import android.content.pm.PackageManager
 9 import android.os.Bundle
10 import android.util.Log
11 import android.widget.Toast
12 import androidx.appcompat.app.AppCompatActivity
13 import androidx.core.app.ActivityCompat
14 import com.example.withmealarm.databinding.
   ActivityHomeBinding
15 import com.firebase.geofire.GeoFire
16 import com.firebase.geofire.GeoLocation
17 import com.google.android.gms.common.api.
   ResolvableApiException
18 import com.google.android.gms.location.*
19 import com.google.firebase.auth.FirebaseAuth
20 import com.google.firebase.auth.ktx.auth
21 import com.google.firebase.database.DataSnapshot
22 import com.google.firebase.database.DatabaseError
23 import com.google.firebase.database.FirebaseDatabase
24 import com.google.firebase.database.
   ValueEventListener
25 import com.google.firebase.ktx.Firebase
26
27
28 lateinit var binding: ActivityHomeBinding
29
30 class U_Home : AppCompatActivity() {
31
32     var TAG: String = "location"
33     val geofenceList: MutableList<Geofence> =
   mutableListOf()
34     var latitude: Double = 0.0
35     var longitude: Double = 0.0
36
37     lateinit var fusedLocationProviderClient:
```


Figure 4. 49 Home Module Code (Page 1)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\U_Home.kt

```
37 FusedLocationProviderClient
38     lateinit var geofencingClient: GeofencingClient
39     private lateinit var firebaseAuth: FirebaseAuth
40
41     private val databaseUrl = "https://withmealarm-
   default-rtdb.asia-southeast1.firebasedatabase.app/"
42     private val database = FirebaseDatabase.
   getInstance(databaseUrl)
43
44     //check if device is running Android Q(API 29) or
    later
45     private val runningQOrLater = android.os.Build.
   VERSION.SDK_INT >= android.os.Build.VERSION_CODES.Q
46
47     //request persmission from user to access
   location if not granted
48     private val
   REQUEST_FOREGROUND_AND_BACKGROUND_PERMISSION_RESULT_C
   ODE = 3 // random unique value
49     private  val
   REQUEST_FOREGROUND_ONLY_PERMISSIONS_REQUEST_CODE = 4
50     private val REQUEST_TURN_DEVICE_LOCATION_ON = 5
51
52     override fun onCreate(savedInstanceState: Bundle
   ?) {
53         super.onCreate(savedInstanceState)
54         binding = ActivityHomeBinding.inflate(
   layoutInflater)
55         setContentView(binding.root)
56
57         val rippleBackground = binding.content
58         rippleBackground.startRippleAnimation()
59
60         binding.btnTogetherId.setOnClickListener {
61             val CRUD_Intent = Intent(this, CRUD_id::
   class.java)
62             startActivity(CRUD_Intent)
63         }
64
65         binding.linkUserProfile.setOnClickListener {
66             val userProfile = Intent(this,
```

Page 2 of 12

Figure 4. 50 Home Module Code (Page 2)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\U_Home.kt

```
66 UserProfile::class.java)
67             startActivity(userProfile)
68         }
69 
70         //2) Initialize Firebase Auth
71         firebaseAuth = Firebase.auth
72         val user = firebaseAuth.currentUser
73 
74         val db = FirebaseDatabase.getInstance(
   databaseUrl).getReference("Users")
75 
76     //GEOFENCE IMPLEMENTATION
   ---------------------------------->
77         fusedLocationProviderClient =
   LocationServices.getFusedLocationProviderClient(this
   )
78 
79         if(user != null) {
80             val uid = user.uid
81 
82             val childRef = db.child(uid)
83 
84             // Retrieve the user's together ID from
   the Realtime Database
85             childRef.addValueEventListener(object :
   ValueEventListener {
86                 override fun onDataChange(
   dataSnapshot: DataSnapshot) {
87 
88                     // Loop through the children of
   the DataSnapshot
89                     for (childSnapshot in
   dataSnapshot.children) {
90                         // Get the values from the
   child snapshot and assign them to variables
91                         val togetherId =
   dataSnapshot.child("togetherID").getValue(String::
   class.java)
92 
93                         //set together id at
   homepage
```


Figure 4. 51 Home Module Code (Page 3)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\U_Home.kt

```
 94                          binding.tvTogetherId.text =
    togetherId
 95                      }
 96                  }
 97
 98                  override fun onCancelled(
    databaseError: DatabaseError) {
 99                      // Handle database error
100                  }
101              })
102
103              checkLocation(uid)
104          }
105
106          //add geofencing client as main entry point
    for interacting w/ geofencing APIs
107          geofencingClient = LocationServices.
    getGeofencingClient(this)
108
109
    requestForegroundAndBackgroundLocationPermissions()
110
111          //notification trigger
112          createChannel(this)
113
114      }//end fun oncreate
115
116      //function to check the user permission and set
    geofence with fetched last known location by lat/
    long
117      private fun checkLocation(uid:String)
118      {
119          val task = fusedLocationProviderClient.
    lastLocation
120
121          if(ActivityCompat.checkSelfPermission(this,
    android.Manifest.permission.ACCESS_FINE_LOCATION)
122              != PackageManager.PERMISSION_GRANTED &&
123              ActivityCompat.checkSelfPermission(this
    , android.Manifest.permission.ACCESS_COARSE_LOCATION
    )
```


Figure 4. 52 Home Module Code (Page 4)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\U_Home.kt

```
124             != PackageManager.PERMISSION_GRANTED)
125         {
126             ActivityCompat.requestPermissions(this,
    arrayOf(android.Manifest.permission.
    ACCESS_FINE_LOCATION), 101)
127             return
128         }
129         task.addOnSuccessListener {
130             if(it != null){
131 
132                 //SET GEOFENCE
    -------------------------->
133                 geofenceList.add(Geofence.Builder()
134                     // Set the request ID of the
    geofence. This is a string to identify this
135                     // geofence.
136                     .setRequestId("entry.key")
137 
138                     // Set the circular region of
    this geofence.
139                     .setCircularRegion(
140                         it.latitude,
141                         it.longitude,
142                         180f //Constants.
    GEOFENCE_RADIUS_IN_METERS
143                     )
144 
145                     // Set the expiration duration
    of the geofence. This geofence gets automatically
146                     // removed after this period of
    time.
147                     .setExpirationDuration(Geofence.
    NEVER_EXPIRE) //Constants.
    GEOFENCE_EXPIRATION_IN_MILLISECONDS
148 
149                     // Set the transition types of
    interest. Alerts are only generated for these
150                     // transition. We track entry
    and exit transitions in this sample.
151                     .setTransitionTypes(Geofence.
    GEOFENCE_TRANSITION_ENTER or Geofence.
```

Page 5 of 12

Figure 4. 53 Home Module Code (Page 5)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\U_Home.kt

```
151 GEOFENCE_TRANSITION_EXIT) //or Geofence.
    GEOFENCE_TRANSITION_EXIT
152                     // Create the geofence.
153                     .build())
154
155                 Toast.makeText(applicationContext, "
    ${it.latitude} ${it.longitude}", Toast.LENGTH_SHORT
    ).show()
156
157                 //save data
158                 writeUserCurrentLocation(uid, it.
    latitude, it.longitude)
159             }
160         }
161     }
162
163     private fun writeUserCurrentLocation( uid:String
    , lat: Double, lng: Double) {
164
165         // Update your database with the location
    data
166         val databaseReference = database.
    getReference("Users").child(uid)
167         val geoFire = GeoFire(databaseReference)
168         geoFire.setLocation("current_location",
    GeoLocation(lat, lng))
169
170         Toast.makeText(
171             baseContext,
172             "Current Location Saved",
173             Toast.LENGTH_SHORT,
174         ).show()
175     }
176
177     //2 step closer to check geofence transition
178     @TargetApi(29)
179     private fun
    foregroundAndBackgroundLocationPermissionApproved
    (): Boolean {
180         val foregroundLocationApproved = (
181                 PackageManager.PERMISSION_GRANTED ==
```

Page 6 of 12

Figure 4. 54 Home Module Code (Page 6)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\U_Home.kt

```
182                         ActivityCompat.
    checkSelfPermission(this,
183                             android.Manifest.
    permission.ACCESS_FINE_LOCATION))
184         val backgroundPermissionApproved =
185             if (runningQOrLater) {
186                 PackageManager.PERMISSION_GRANTED ==
187                         ActivityCompat.
    checkSelfPermission(
188                             this, android.Manifest.
    permission.ACCESS_BACKGROUND_LOCATION
189                         )
190             } else {
191                 true
192             }
193         return foregroundLocationApproved &&
    backgroundPermissionApproved
194     }
195
196     //1 step closer to check geofence transition
197     private fun
    requestForegroundAndBackgroundLocationPermissions
    () {
198         if (
    foregroundAndBackgroundLocationPermissionApproved())
199             return
200         var permissionsArray = arrayOf(android.
    Manifest.permission.ACCESS_FINE_LOCATION)
201         val resultCode = when {
202             runningQOrLater -> {
203                 permissionsArray += android.Manifest
    .permission.ACCESS_BACKGROUND_LOCATION
204
    REQUEST_FOREGROUND_AND_BACKGROUND_PERMISSION_RESULT_
    CODE
205             }
206             else ->
    REQUEST_FOREGROUND_ONLY_PERMISSIONS_REQUEST_CODE
207         }
208         Log.d(TAG, "Request foreground only location
     permission")
```

Page 7 of 12

Figure 4. 55 Home Module Code (Page 7)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\U_Home.kt

```
209         ActivityCompat.requestPermissions(
210             this,
211             permissionsArray,
212             resultCode
213         )
214     }
215 
216     //3 step closer to check geofence transition
217     //once user responds to permission request, app
    should process their response this method
218     override fun onRequestPermissionsResult(
219         requestCode: Int,
220         permissions: Array<String>,
221         grantResults: IntArray
222     ) {
223         Log.d(TAG, "onRequestPermissionResult")
224         super.onRequestPermissionsResult(requestCode
    , permissions, grantResults)
225 
226         if (requestCode ==
    REQUEST_FOREGROUND_AND_BACKGROUND_PERMISSION_RESULT_
    CODE ||
227             requestCode ==
    REQUEST_FOREGROUND_ONLY_PERMISSIONS_REQUEST_CODE) {
228             if (grantResults.isNotEmpty() && (
    grantResults[0] == PackageManager.PERMISSION_GRANTED
    )) {
229                 validateGadgetAreaInitiateGeofence()
230                 //no problem here but crashed
    afterwards
231             }
232         }
233     }//end fun onRequestPermissionsResult
234 
235     //4 step closer to check geofence transition
236     private fun validateGadgetAreaInitiateGeofence(
    resolve:Boolean = true) {
237 
238         val locationRequest = LocationRequest.
    Builder(Priority.PRIORITY_HIGH_ACCURACY, 100).build
    ()
```

Page 8 of 12

Figure 4. 56 Home Module Code (Page 8)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\U_Home.kt

```
239
240         val builder = LocationSettingsRequest.
    Builder().addLocationRequest(locationRequest)
241         val settingsClient = LocationServices.
    getSettingsClient(this)
242         val locationSettingsResponseTask =
243             settingsClient.checkLocationSettings(
    builder.build())
244         locationSettingsResponseTask.
    addOnFailureListener { exception ->
245             if (exception is ResolvableApiException
     && resolve){
246                 try {
247                     exception.
    startResolutionForResult(this,
248
    REQUEST_TURN_DEVICE_LOCATION_ON)
249                 } catch (sendEx: IntentSender.
    SendIntentException) {
250                     Log.d(TAG, "Error getting
    location settings resolution: " + sendEx.message)
251                 }
252             } else {
253
    validateGadgetAreaInitiateGeofence()
254                 }
255
256         }
257         locationSettingsResponseTask.
    addOnCompleteListener {
258             if (it.isSuccessful)
259             {
260                 addGeofence()
261             }
262         }
263     }
264
265     //5 step closer to check geofence transition
266     override fun onActivityResult(requestCode: Int,
    resultCode: Int, data: Intent?) {
267         super.onActivityResult(requestCode,
```


Figure 4. 57 Home Module Code (Page 9)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\U_Home.kt

```
resultCode, data)
        validateGadgetAreaInitiateGeofence(false)
    }

    //6)create pending intent to handle geofence transitions
    private val geofencePendingIntent: PendingIntent by lazy {
        val intent = Intent(this, GeofenceBroadcastReceiver::class.java)
        PendingIntent.getBroadcast(this, 0, intent, PendingIntent.FLAG_MUTABLE)
    }

    //7)function to specify the geofence to monitor and to set how related geofence events are triggered
    private fun getGeofencingRequest(userBuid: String): GeofencingRequest {
        return GeofencingRequest.Builder().apply {
            setInitialTrigger(GeofencingRequest.INITIAL_TRIGGER_ENTER or GeofencingRequest.INITIAL_TRIGGER_EXIT)
            addGeofences(geofenceList)
        }.build().also { geofencingRequest ->
            val intent = Intent(this, GeofenceBroadcastReceiver::class.java)
            intent.putExtra("userBuid", userBuid)
        }
    }

    //8)To associate a geofence with a pendingIntent
    private fun addGeofence(){
        //current user
        val user = firebaseAuth.currentUser
        val uid = user?.uid.toString()

        //pass intently User B uid
        val dbRef = database.getReference("Users").child(uid).child("Saved_togetherID").child("savedEuid")

```

Page 10 of 12

Figure 4. 58 Home Module Code (Page 10)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\U_Home.kt

```
297         dbRef.addListenerForSingleValueEvent(object
     : ValueEventListener {
298             override fun onDataChange(snapshot:
    DataSnapshot) {
299                 // Retrieve the saved id from the
    snapshot
300                 val savedId = snapshot.child("uid").
    getValue(String::class.java)
301 
302                 //check permission
303                 if (ActivityCompat.
    checkSelfPermission(
304                         applicationContext, Manifest
    .permission.ACCESS_FINE_LOCATION
305                     ) != PackageManager.
    PERMISSION_GRANTED
306                 ) {
307                     return
308                 }
309 
310                 // Do something with the saved id
311                 geofencingClient.addGeofences(
    getGeofencingRequest("$savedId"),
    geofencePendingIntent).run {
312                     addOnSuccessListener {
313                         Toast.makeText(baseContext,
    "Geofence(s) added", Toast.LENGTH_SHORT).show()
314                         Log.d("TAG", "Geofence
    addded")
315                     }
316                     addOnFailureListener {
317                         Toast.makeText(baseContext,
    "Failed to add geofence(s)", Toast.LENGTH_SHORT).
    show()
318                         Log.d("TAG", "Geofence
    failed")
319                     }
320                 }
321             }
322 
323             override fun onCancelled(error:
```

Page 11 of 12

Figure 4. 59 Home Module Code (Page 11)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\U_Home.kt

```
DatabaseError) {
                // Handle the error if the operation is cancelled
                Log.d("TAG", "Database error occurred: ${error.message}")
            }
        })
    }

//auth
    public override fun onStart() {
        super.onStart()
        //3) Check if user is signed in (non-null) and update UI accordingly.
        val user = Firebase.auth.currentUser
        if (user != null) {
            // User is signed in
        } else {
            // No user is signed in
            val loginIntent = Intent(this, Login::class.java)
            startActivity(loginIntent)
        }
    }
}
```

Page 12 of 12

Figure 4. 60 Home Module Code (Page 12)

ii. Geofence Broadcast Receiver

- This code file includes Notification Manager and Alarm Manager.
- At first, the app will retrieve the end-user's UID which passed by intent. With that, the app will collect the end-user's location saved from database.
- After that, the Geofence Event will handle the geofence transition type. If the transition is equal to ENTER, then the notification manager will call method `sendGeofenceEnteredNotification`. Otherwise, it will call method `sendGeofenceExitedNotification` along with alarm manager with specific setup.

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\GeofenceBroad

```
 1 package com.example.withmealarm
 2
 3 import android.app.AlarmManager
 4 import android.app.NotificationManager
 5 import android.app.PendingIntent
 6 import android.content.BroadcastReceiver
 7 import android.content.Context
 8 import android.content.Intent
 9 import android.util.Log
10 import androidx.core.content.ContextCompat
11 import com.google.android.gms.location.Geofence
12 import com.google.android.gms.location.
   GeofenceStatusCodes
13 import com.google.android.gms.location.
   GeofencingEvent
14 import com.google.firebase.database.DataSnapshot
15 import com.google.firebase.database.DatabaseError
16 import com.google.firebase.database.FirebaseDatabase
17 import com.google.firebase.database.
   ValueEventListener
18
19 class GeofenceBroadcastReceiver : BroadcastReceiver
   () {
20
21     private val databaseUrl = "https://withmealarm-
   default-rtdb.asia-southeast1.firebasedatabase.app/"
22     private val database = FirebaseDatabase.
   getInstance(databaseUrl)
23
24     override fun onReceive(context: Context, intent:
   Intent) {
25
26         val geofencingEvent = GeofencingEvent.
   fromIntent(intent)
27         if (geofencingEvent != null) {
28             if (geofencingEvent.hasError()) {
29                 val errorMessage =
   GeofenceStatusCodes.getStatusCodeString(
   geofencingEvent.errorCode)
30                 Log.e(TAG, errorMessage)
31                 return
```

Page 1 of 4

Figure 4. 61 Geofence Broadcast Receiver Extension Code (Page 1)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\GeofenceBroad

```
32              }
33              Log.d("TAG", "Geofence started")
34          }
35
36          //notification manager setup
37          val notificationManager = ContextCompat.
    getSystemService(
38              context,
39              NotificationManager::class.java
40          ) as NotificationManager
41
42          // Retrieve the user B's uid passed from the
    intent
43          val userBuid = intent.getStringExtra("
    userBUid") ?: return
44
45          // Retrieve User B's location from the
    database
46          val userBLocationRef = database.getReference(
    "Users").child(userBuid)
47          userBLocationRef.addValueEventListener(object
     : ValueEventListener {
48              override fun onDataChange(snapshot:
    DataSnapshot) {
49                  if (snapshot.exists()) {
50                      val latitude = snapshot.child("
    latitude").value as Double
51                      val longitude = snapshot.child("
    longitude").value as Double
52                      // do something with the location
     data
53
54                      // Handle the geofencing event
    based on the transition type
55                      if (geofencingEvent?.
    geofenceTransition == Geofence.
    GEOFENCE_TRANSITION_ENTER) {
56                          Log.i(TAG, "User B entered
    the geofence.")
57
58                          // Do something when User B
```


Figure 4. 62 Geofence Broadcast Receiver Extension Code (Page 2)

```
58 enters the geofence
59                          notificationManager.
   sendGeofenceEnteredNotification(
60                              context
61                          )
62
63                      } else if (geofencingEvent?.
   geofenceTransition == Geofence.
   GEOFENCE_TRANSITION_EXIT) {
64                          Log.i(TAG, "User B exited the
    geofence.")
65                          // Do something when User B
   exits the geofence
66                          notificationManager.
   sendGeofenceExitedNotification(
67                              context
68                          )
69
70                          //Alarm Manager
   implementation
71                          // Set up the alarm to be
   triggered after the specified delay
72                          val alarmIntent = Intent(
   context, AlarmReceiver::class.java)
73                          val alarmPendingIntent =
   PendingIntent.getBroadcast(
74                              context,
75                              ALARM_ID,
76                              alarmIntent,
77                              PendingIntent.
   FLAG_UPDATE_CURRENT
78                          )
79
80                          //added permission in android
    manifest: SCHEDULE_EXACT_ALARM
81                          val alarmManager = context.
   getSystemService(Context.ALARM_SERVICE) as
   AlarmManager
82                          alarmManager.setExact(
83                              AlarmManager.RTC_WAKEUP,
84                              System.currentTimeMillis
```


Figure 4. 63 Geofence Broadcast Receiver Extension Code (Page 3)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\GeofenceBroad

```
84 () + ALARM_DELAY_MS,
85                             alarmPendingIntent
86                         )
87                     }
88                 }
89             }
90
91             override fun onCancelled(databaseError:
   DatabaseError) {
92                 Log.e(TAG, "Failed to retrieve User
   B's location: ${databaseError.message}")
93             }
94         })
95     }
96
97     companion object {
98         private const val TAG = "
   GeofenceBroadcastReceiver"
99         private const val ALARM_DELAY_MS = 0 // Set
   the alarm delay to 0 to avoid triggering another
   alarm //300000 = 5 min [longer delay - user forget
    || shorter delay - user annoy]
100        private const val ALARM_ID = 78 // or any
   unique integer value
101    }
102 }
```

Page 4 of 4

Figure 4. 64 Geofence Broadcast Receiver Extension Code (Page 4)

iii. Notification Trigger

- There are two similar methods but with different notification contents text.
- `sendGeofenceEnteredNotification` is triggered when the end-user entered into geofence area.
- `sendGeofenceExitedNotification` is triggered when the end-user exited into geofence area.

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\NotificationUtils

```
 1 package com.example.withmealarm
 2
 3 import android.app.NotificationChannel
 4 import android.app.NotificationManager
 5 import android.app.PendingIntent
 6 import android.content.Context
 7 import android.content.Intent
 8 import android.os.Build
 9 import androidx.core.app.NotificationCompat
10
11 private const val NOTIFICATION_ID = 33
12 private const val CHANNEL_ID = "GeofenceChannel"
13
14 fun createChannel(context: Context) {
15     if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.
   O) {
16         val notificationChannel =
17             NotificationChannel(CHANNEL_ID, "Channel1
   ", NotificationManager.IMPORTANCE_HIGH)
18         val notificationManager = context.
   getSystemService(NotificationManager::class.java)
19         notificationManager.createNotificationChannel
   (notificationChannel)
20     }
21 }
22
23 fun NotificationManager.
   sendGeofenceEnteredNotification(context: Context) {
24
25     //Opening the Notification
26     val contentIntent = Intent(context, U_Home::class
   .java)
27     val contentPendingIntent = PendingIntent.
   getActivity(
28         context,
29         NOTIFICATION_ID,
30         contentIntent,
31         PendingIntent.FLAG_MUTABLE
32     )
33     //Building the notification
34     val builder = NotificationCompat.Builder(context
```

Page 1 of 3

Figure 4. 65 Notification and Alarm Trigger Extension Code (Page 1)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\NotificationUtils

```
34 , CHANNEL_ID)
35         .setContentTitle(context.getString(R.string.
   app_name))
36         .setContentText("Someone have entered a
   geofenced area")
37         .setSmallIcon(R.drawable.
   ic_baseline_connect_without_contact_24)
38         .setPriority(NotificationCompat.PRIORITY_HIGH
   )
39         .setContentIntent(contentPendingIntent)
40         .build()
41
42     this.notify(NOTIFICATION_ID, builder)
43 }
44
45 fun NotificationManager.
   sendGeofenceExitedNotification(context: Context) {
46
47     //Opening the Notification
48     val contentIntent = Intent(context, U_Home::class
   .java)
49     val contentPendingIntent = PendingIntent.
   getActivity(
50         context,
51         NOTIFICATION_ID,
52         contentIntent,
53         PendingIntent.FLAG_MUTABLE
54     )
55     //Building the notification
56     val builder = NotificationCompat.Builder(context
   , CHANNEL_ID)
57         .setContentTitle(context.getString(R.string.
   app_name))
58         .setContentText("Someone have exited a
   geofenced area")
59         .setSmallIcon(R.drawable.
   ic_baseline_connect_without_contact_24)
60         .setPriority(NotificationCompat.PRIORITY_HIGH
   )
61         .setContentIntent(contentPendingIntent)
62         .build()
```

Page 2 of 3

Figure 4. 66 Notification and Alarm Trigger Extension Code (Page 2)

File - C:\Users\Hana Fatiha\AndroidStudioProjects\WithMeAlarm\app\src\main\java\com\example\withmealarm\NotificationUtils

```
63
64      this.notify(NOTIFICATION_ID, builder)
65  }
```

Figure 4. 67 Notification and Alarm Trigger Extension Code (Page 3)

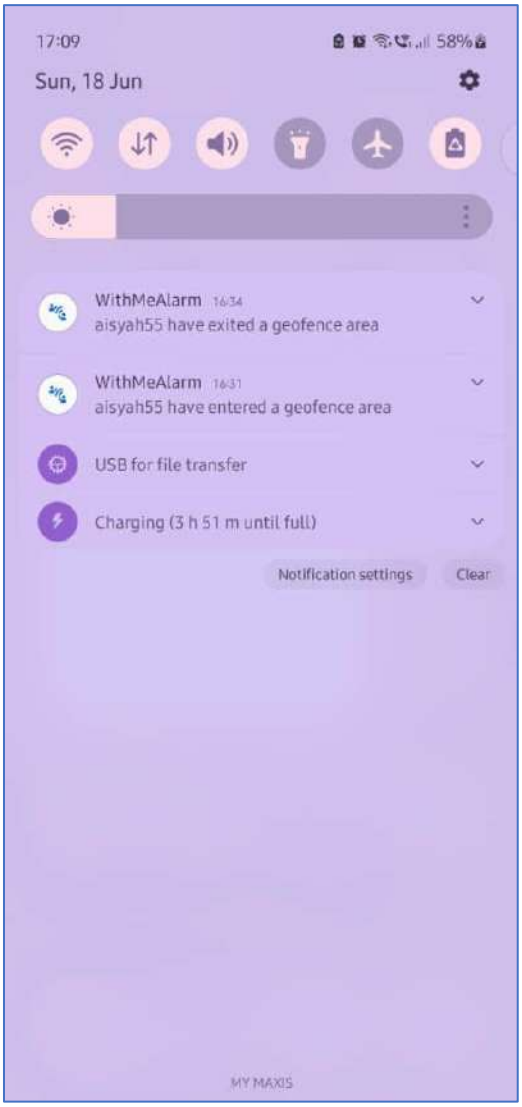


Figure 4. 68 Example of notification triggered when a user entered/exited a geofence area

4.2.4 Database Design Implementation

Firebase Authentication helps to handle the user authentication using email address and password. The User UID is uniquely generated by the Firebase which therefore it saved in Realtime Database too. This is due to UID is used for most module in app project.

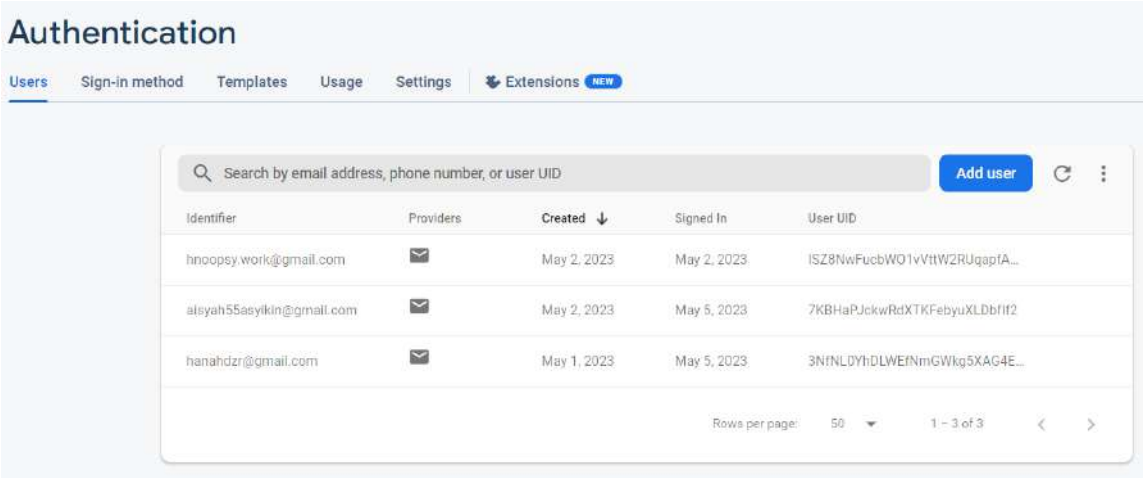


Figure 4. 69 Firebase Authentication Implementation

Within Realtime Database of app project, all user's data saved under a collection called `Users` with User UID as child node. This child node become parent node to other collections such as `Saved_togetherID` for end-user's UID and `current_location` for user's current location.

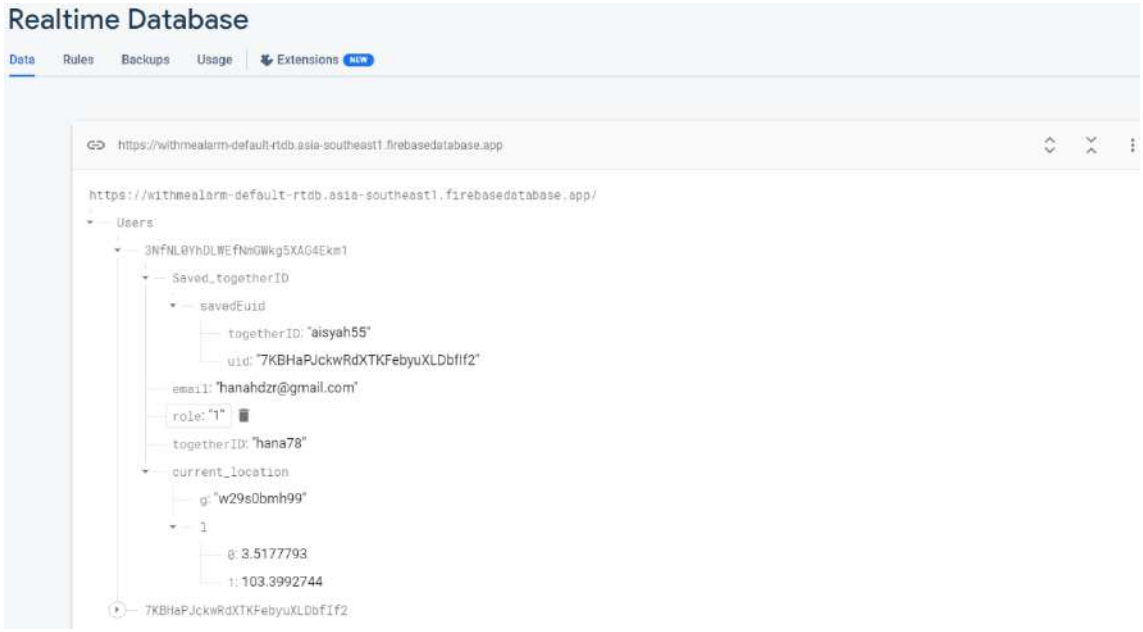


Figure 4. 70 Firebase Realtime Database Implementation

Under `Saved_togetherID` node, there is a child node named `WithMeAlarm_Log` which saved the history of the alarm was triggered when the end-user exited the geofence area. It saved the date of the event, distance between of end-user and the user, the clock time when the alarm triggered, end-user's activity which is exit, and end-user's current longitude also latitude. This data is displayed in the WitheMe History tab section as shown in Figure 4.72.

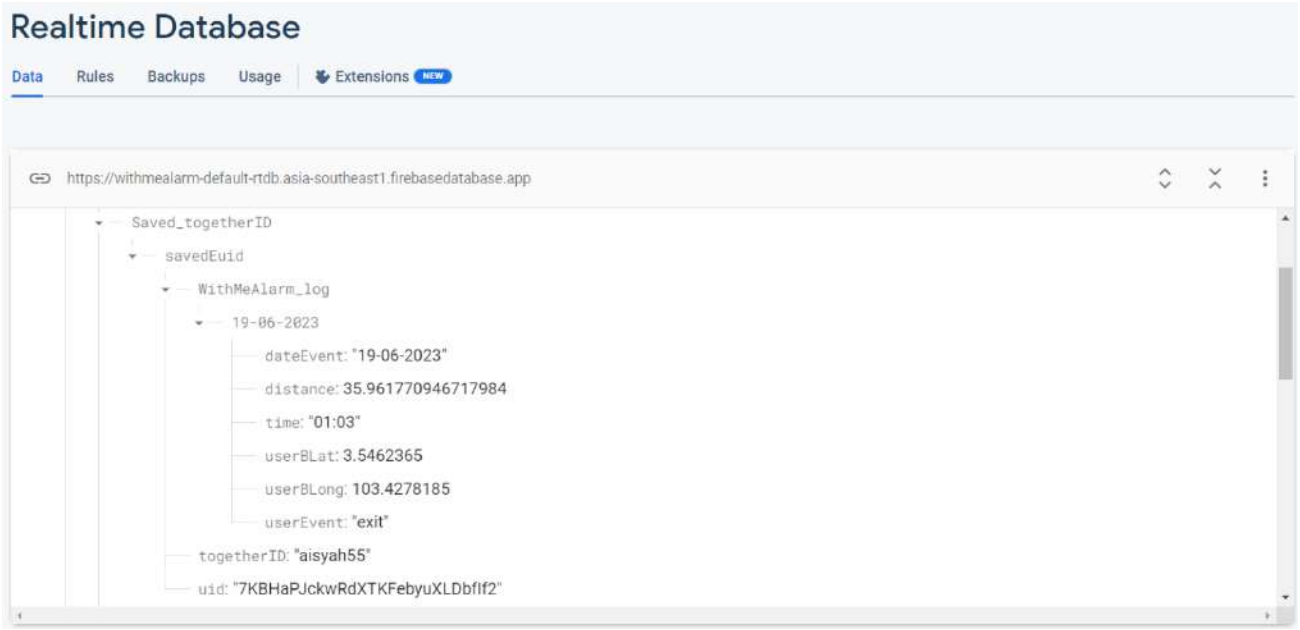


Figure 4. 71 WithMeAlarm_Log Data Structure

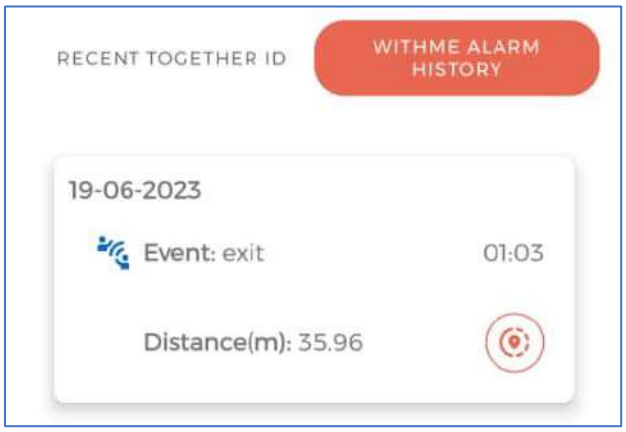


Figure 4. 72 WithMe ALarm History

4.3 Testing

i. **Functional Test Cases**. The test cases are done by the developer of WithMe Alarm app.

The testing process is started from the app is built into the developer's device. The device requirements are Android 13 operating system, background location and notification permissions are allowed also the device must has internet connection. The app is tested according to Functional Test Cases form as in APPENDIX A.

ii. **System Usability Scale (SUS)**. The form is filled by the potential users of WithMe Alarm app.

The testing process is involved with two interested users of WithMe Alarm app. They installed WithMe Alarm APK into their personal device. Both of them were acting as a guardian and non-guardian respectively. After they went through the app, they are required to give feedback at Google Form given as in APPENDIX B.

4.4 Result Discussion

Based on the SUS feedback, both users provided mostly positive ratings for the ten questions. Their detailed feedback can be found in APPENDIX C. Below is a summary of the result discussion for each question:

1. "I think that I would like to use this app frequently."
    - Both users rated this question highly, indicating that the app can be considered necessary in daily life.
2. "I found the app unnecessarily complex."
    - One user rated this question highly, while the other user provided the lowest rating.
3. "I thought the app was easy to use."
    - Both users rated this question highly, indicating that the app is user-friendly and can be easily learned.

4. "I think I would need support from a technical person to be able to use this app."
    - One user rated this question highly, while the other user provided a medium rating.
5. "I found the various functions in this app were well integrated."
    - Both users rated this question highly, suggesting that the app's modules are well-functioning and seamlessly integrated.
6. "I thought there was too much inconsistency in this app."
    - Both users provided the lowest rating, indicating agreement that the elements in the app's modules are consistently designed.
7. "I would imagine that most people would learn to use the app very quickly."
    - Both users rated this question highly, indicating agreement that the app is well-developed for public users and has a short learning curve.
8. "I found the app very awkward to use."
    - Both users rated this question with a low score, indicating that they found the app easy to use despite the introduction of new concepts.
9. "I felt very confident using the app."
    - Both users gave the highest rating, suggesting a high level of confidence in using the app.
10. "I need to learn a lot of things before I could get going with this system."
    - One user rated this question highly, while the other user provided a medium rating.

Overall, both users expressed highly positive opinions about the system usability of the WithMe Alarm app. However, in question 2, 4, and 10, one user expressed the need for additional assistance while interacting with the app. Therefore, it is recommended that the developer consider incorporating documentation or tutorial elements within the app to facilitate user navigation and enhance overall efficiency.

CHAPTER 5

CONCLUSION

5.1 Introduction

This last chapter of thesis conclude about the project development and implementation. WithMe Alarm app serves similar as “FindMyPhone” apps in the market but it is more towards caring people. The most significance of the app is due to geofence implementation. The technology is not taught in faculty’s subject therefore it is a new learning experience to be applied for a final year student’s project.

Based on the analysis of collected feedback and test cases, the developed app in this project has demonstrated potential in effectively solving the identified problem. Throughout the project's development, the app has undergone refinement and improvement, reaching a stage where it shows substantial capability. However, further evaluation and optimization may still be necessary.

The methodology of WithMe Alarm application for project development adopted is based on Agile software process model. It is the best suitable model for this project development due to flexibility to frequent change of user and system requirements in short duration with low budget(*IJCSC*, n.d.). As proven, there are quiet few changes has been taken due to limitation of time and these are described in project constraint.

Based on project constraint and bright future hold by WithMe Alarm mobile application, there are about five suggestions to enhancement of the project development. Some enhancements are from the initial planning of the project and some are for confidently market the application to the public user.

5.2 Project Constraint

i. Time

In the early planning, the end-user has interactivity elements in WithMe Alarm mobile application. The element such as the end-user can stop alarm alert and send message to notify the user. However, during the project development is more focusing on the user side for geofence implementation and geofence is a new topic that is not teach in the faculty syllabus. There are many try and error in geofence development for user side therefore the end-user is only act as a place to save and updates current location for monitoring the geofence.

ii. Coding/Scripting Error

Consequently, the WithMe Alarm app experiences various logic programming errors, resulting in glitches. Additionally, the scripting files lack organization, making it challenging to trace the bugs that are affecting the app's functionality. The most significant errors primarily stem from the geofence implementation. Unfortunately, there is a scarcity of tutorials available for geofence implementation, particularly tailored to this project's specific architecture and requirements.

5.3 Future Work

WithMe Alarm mobile application has wide opportunity to have improvements and additional elements as listed below:

i. WithMe Alarm mobile application can be developed to support in iOS and Huawei service.

ii. Developer can do adjustment in geofence development to improve battery usage efficiency and location accuracy to avoid battery drainage.

iii. WithMe Alarm mobile application can be developed using Flutter framework for better code organization and easier to theme according Malaysia's holidays.

iv. Developer can add element which the user receives notification with location's name and direction of end-user's recent whereabout.

v. Developer able to add interactivity for the end-user side where they can send short message after stopping the alarm alert. The user will receive the short message from notification.

REFERENCES

Familo: Find My Phone Locator - Apps on Google Play. (n.d.). Retrieved December 8, 2022, from https://play.google.com/store/apps/details?id=net.familo.android

Find My Kids ・Family tracker - Apps on Google Play. (n.d.). Retrieved December 8, 2022, from https://play.google.com/store/apps/details?id=org.findmykids.app

Heiss, H.-U., & Gesellschaft für Informatik. (2011). *Informatik 2011 : Informatik schafft Communities ; Beiträge der 41. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 4. - 7.10.2011 in Berlin*. Ges. für Informatik.

IJCSC. (n.d.).

LoveAlarm - 좋아하면 울리는 공식앱 - Apps on Google Play. (n.d.). Retrieved December 8, 2022, from https://play.google.com/store/apps/details?id=net.ky.lovealarm

Milne, H., van der Pol, M., McCloughan, L., Hanley, J., Mead, G., Starr, J., Sheikh, A., & McKinstry, B. (2014). The use of global positional satellite location in dementia: A feasibility study for a randomised controlled trial. *BMC Psychiatry*, *14*(1). https://doi.org/10.1186/1471-244X-14-160

Miskelly, F. (2005). Electronic tracking of patients with dementia and wandering using mobile phone technology [1]. In *Age and Ageing* (Vol. 34, Issue 5, pp. 497–499). https://doi.org/10.1093/ageing/afi145

Namiot, D. (2013). GeoFence services. *International Journal of Open Information Technologies*, *1*(9), 30–33. http://www.injoit.ru/index.php/j1/article/view/51

Pingo by Findmykids - Apps on Google Play. (n.d.). Retrieved December 8, 2022, from https://play.google.com/store/apps/details?id=org.findmykids.child

Zuva, K., & Zuva, T. (n.d.). *Tracking of Customers using Geofencing Technology*.

APPENDIX A

FUNCTIONAL TEST CASES (FOR DEVELOPER)

Appendix A shows the Functional Test Cases for WithMe Alarm App that includes Test Case ID, Test Case Objective, Prerequisite, Steps, Input Data, Expected Output, Actual Output and Status. The test case is executed by developer during the project development.

Date test taken: 10 June 2023

Test Case ID	Test Case Objective	Prerequisite	Steps	Input Data	Expected Output	Actual Output	Status	Comment
TC_01	Test sign up form	No valid account	1) Insert username 2) Insert email 3) Insert password 4) Choose role 5) Hit sign up button	Username: hana78 Email: hanahdzr@gmail.com Password: *******	Sign up successful	*As expected*	*Pass*	-
TC_02	Test login form	A valid account	1) Insert email 2) Insert password 3) Hit login button	Email: hanahdzr@gmail.com Password: *******	Login successful	*As expected*	*Pass*	*Glitch refresh at homepage.*
TC_03	Test new Together ID form	N/A	1) Insert Together ID 2) Hit submit button	Together ID: aisyah55	Together ID saved successfully	*As expected*	*Pass*	-

TC_04	Test Together ID removal	At least one saved Together ID	1) Slide left the together ID 2) Hit delete button	N/A	Together ID removed successfully	*Not expected*	*Fail*	*Investigate code*
TC_05	Test geofence alarm (Guardian user)	- Registered at least one Together ID -Turned on background location -Other device location is outside geofence	1) Click on the notification 2) Click 'Stop Alarm'	N/A	Geofence alarm successful and alarm stopped	*Not expected*	*Fail*	*Investigate code*

Date test taken: 18 June 2023

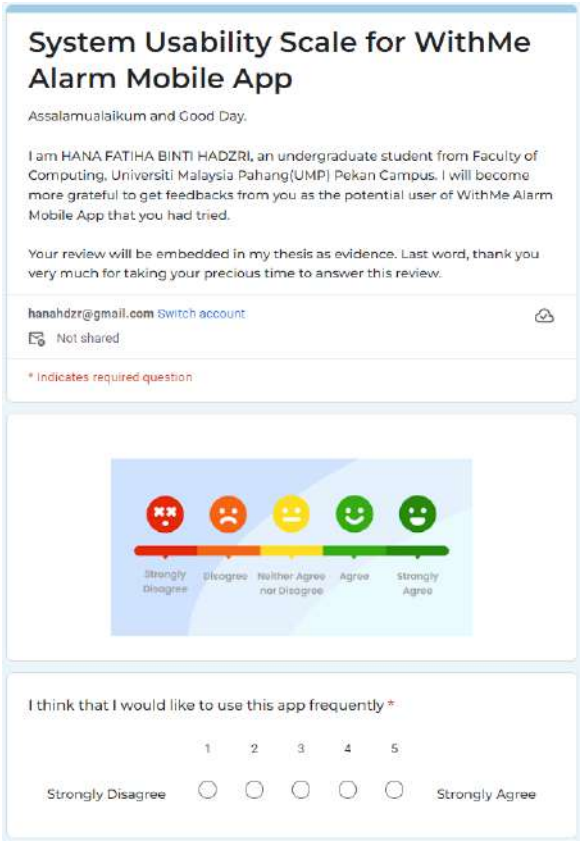
Test Case ID	Test Case Objective	Prerequisite	Steps	Input Data	Expected Output	Actual Output	Status	Comment
TC_01	Test sign up form	No valid account	1) Insert username 2) Insert email 3) Insert password 4) Choose role 5) Hit sign up button	Username: hana78 Email: hanahdzr@gmail.com Password: *******	Sign up successful	*As expected*	*Pass*	-
TC_02	Test login form	A valid account	1) Insert email 2) Insert password 3) Hit login button	Email: hanahdzr@gmail.com Password: *******	Login successful	*As expected*	*Pass*	*Glitch refresh at homepage.*
TC_03	Test new Together ID form	N/A	1) Insert Together ID 2) Hit submit button	Together ID: aisyah55	Together ID saved successfully	*As expected*	*Pass*	-
TC_04	Test Together ID removal	At least one saved Together ID	1) Slide left the together ID 2) Hit delete button	N/A	Together ID removed successfully	*As expected*	*Pass*	-
TC_05	Test geofence alarm (Guardian user)	- Registered at least one Together ID -Turned on background	1) Click on the notification 2) Click 'Stop Alarm'	N/A	Geofence alarm successful and alarm stopped	*As expected*	*Pass*	-

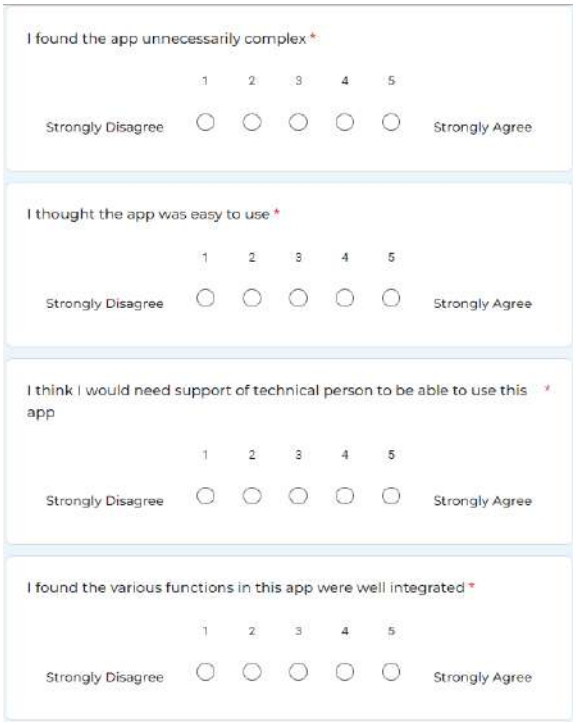
		location -Other device location is outside geofence						

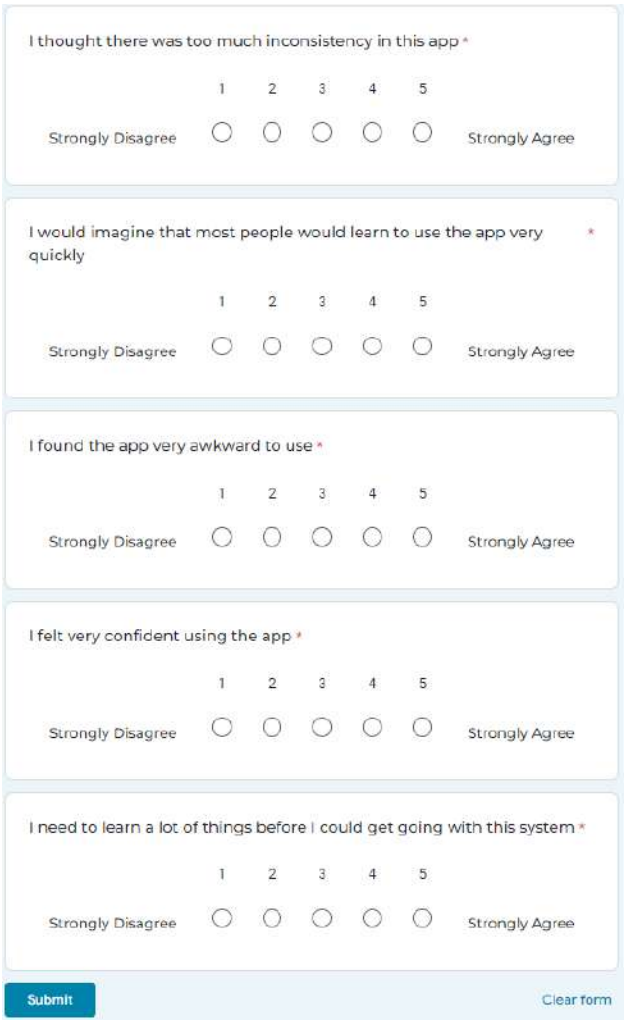
APPENDIX B

SYSTEM USABILITY SCALE FORM (FOR USERS)

Appendix B shows the System Usability Scale for WithMe Alarm App that includes 10 questionnaires. The form is used by the potential users that had tried the app.



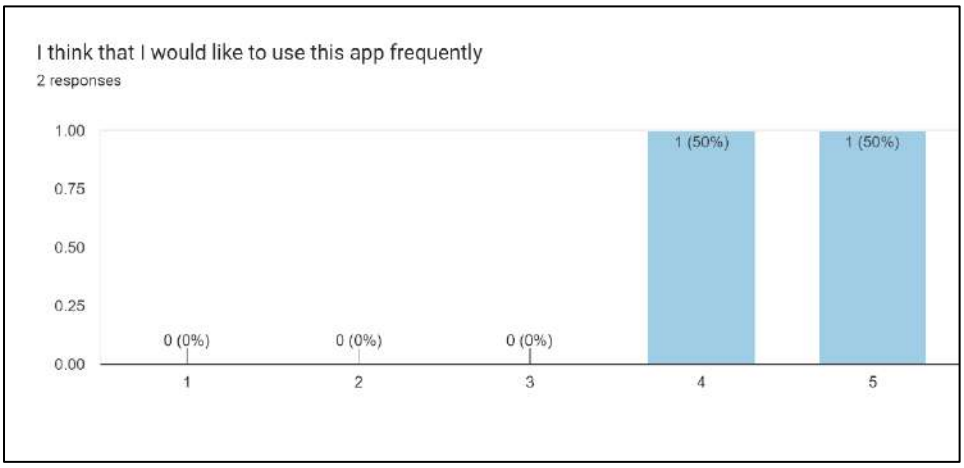


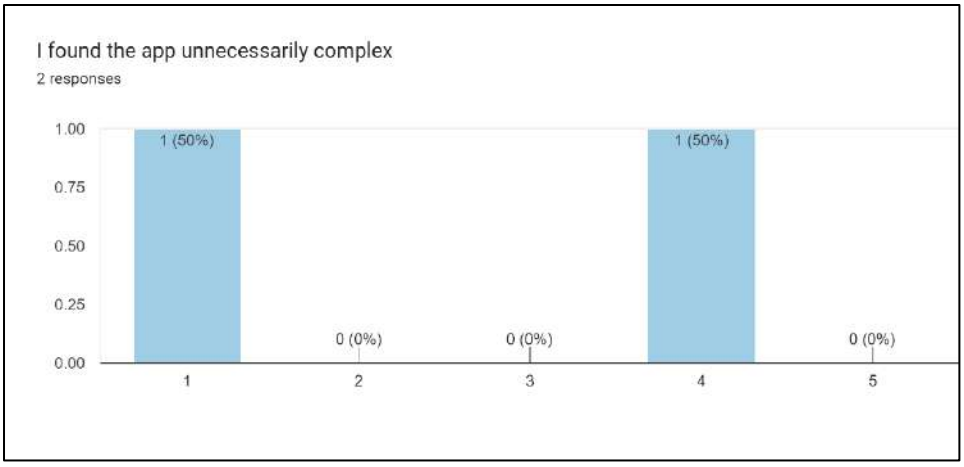

I thought there was too much inconsistency in this app *
1 2 3 4 5
Strongly Disagree
Strongly Agree
I would imagine that most people would learn to use the app very quickly *
1 2 3 4 5
Strongly Disagree
Strongly Agree
I found the app very awkward to use *
1 2 3 4 5
Strongly Disagree
Strongly Agree
I felt very confident using the app *
1 2 3 4 5
Strongly Disagree
Strongly Agree
I need to learn a lot of things before I could get going with this system *
1 2 3 4 5
Strongly Disagree
Strongly Agree
Submit
Clear form

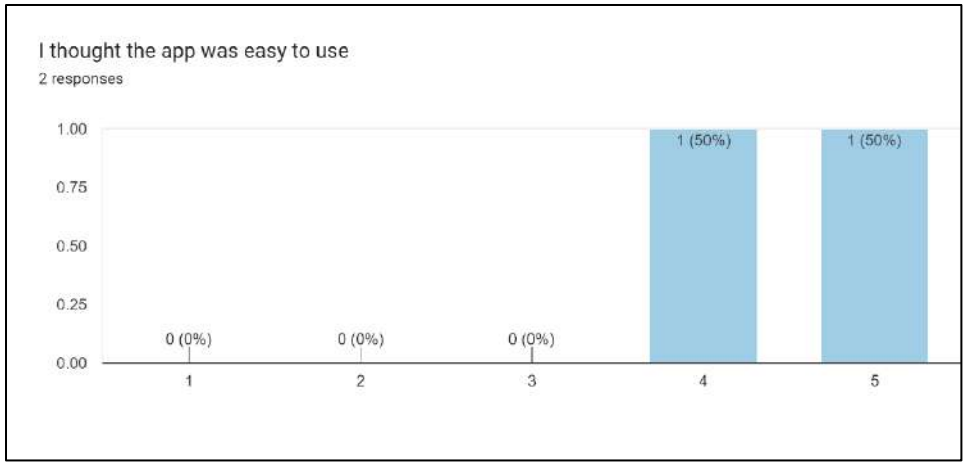
APPENDIX C

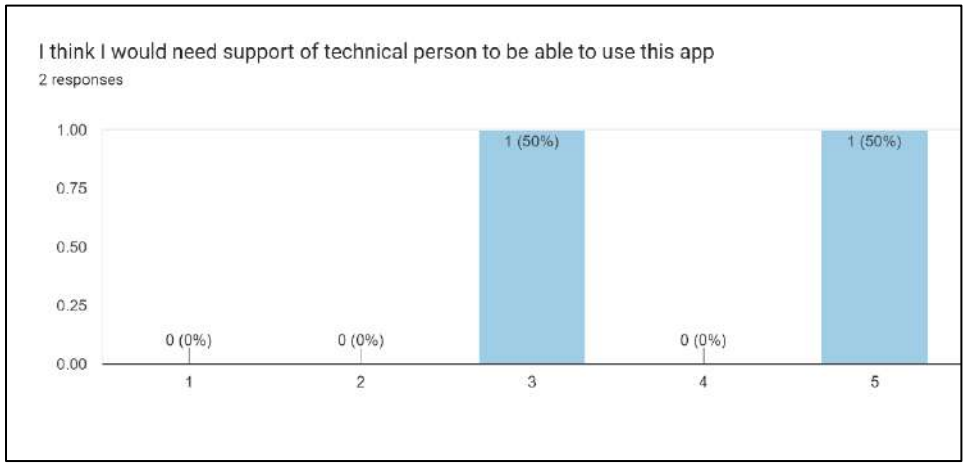
SYSTEM USABILITY SCALE FEEDBACKS (FOR USERS)

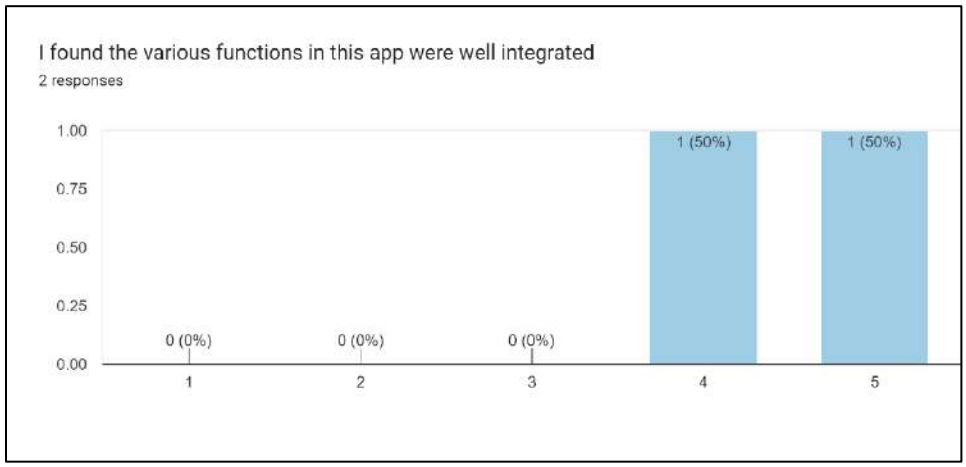
Appendix C shows the System Usability Scale feedbacks for WithMe Alarm App. The form is answered by the potential users that had tried the app.

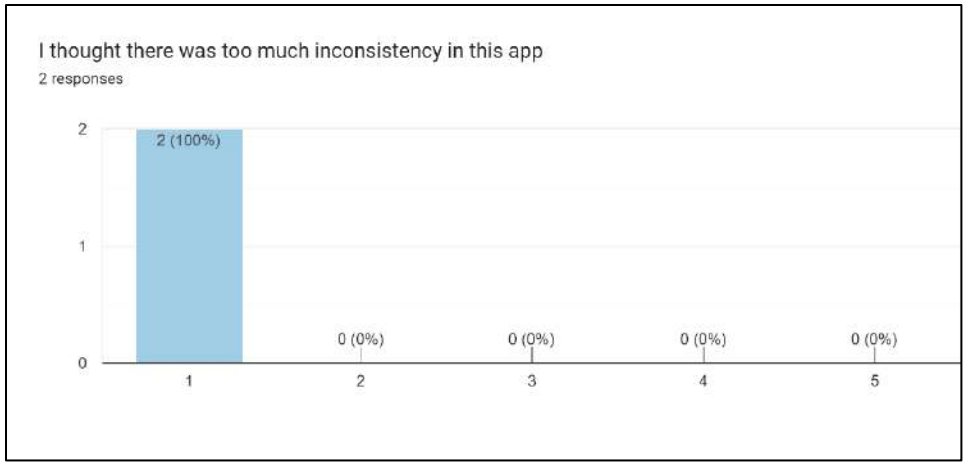


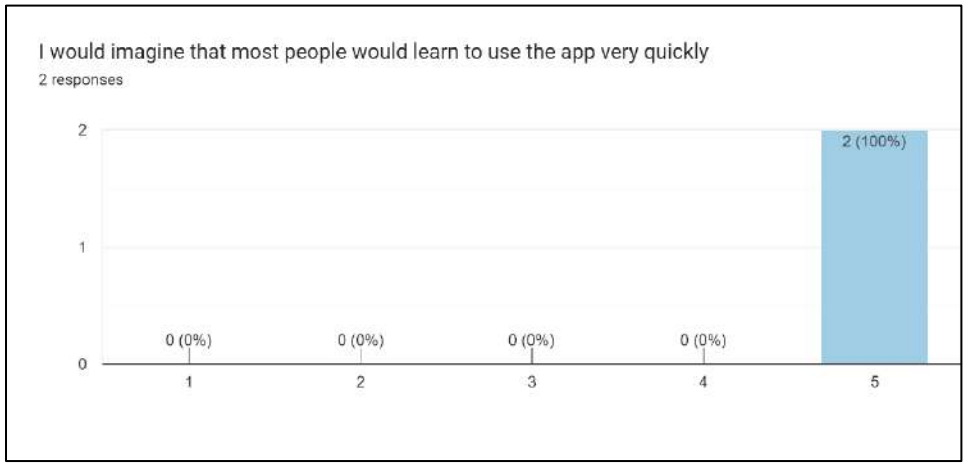


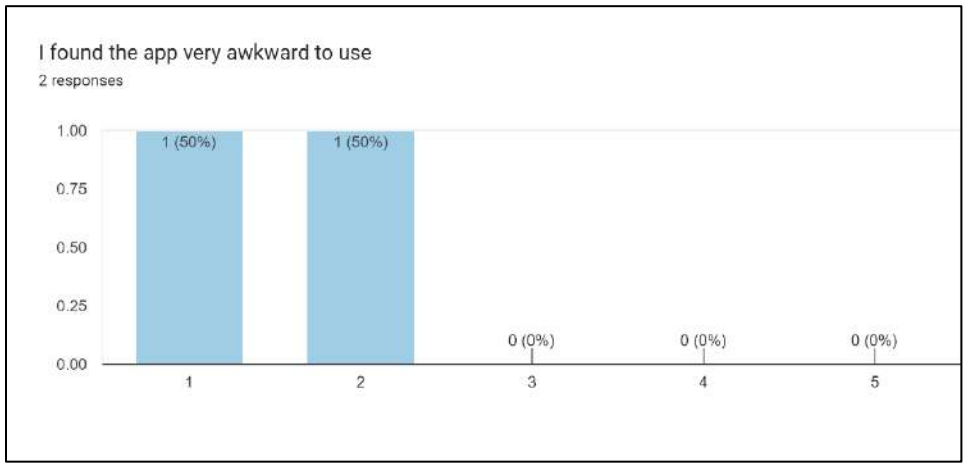

I thought the app was easy to use
2 responses
1.00
0.75
0.50
0.25
0.00
0 (0%)
0 (0%)
0 (0%)
1 (50%)
1 (50%)
1
2
3
4
5

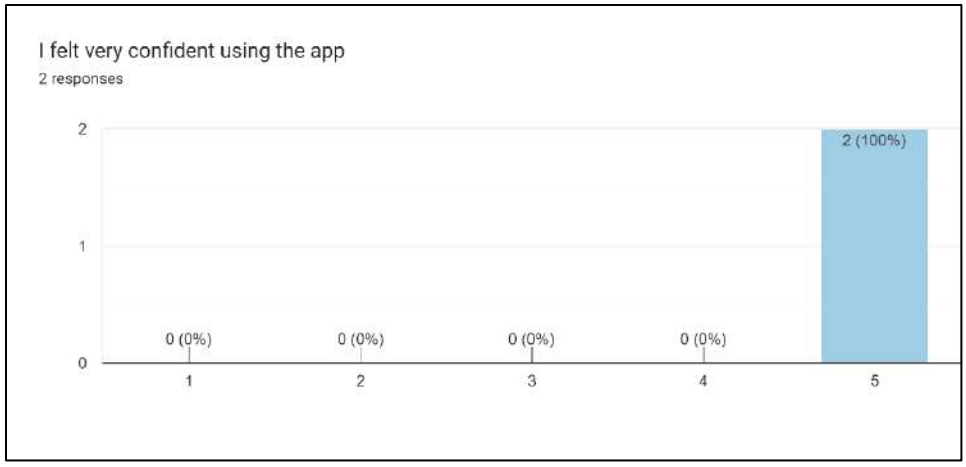

I think I would need support of technical person to be able to use this app
2 responses
1.00
0.75
0.50
0.25
0.00
0 (0%)
0 (0%)
1 (50%)
0 (0%)
1 (50%)
1
2
3
4
5

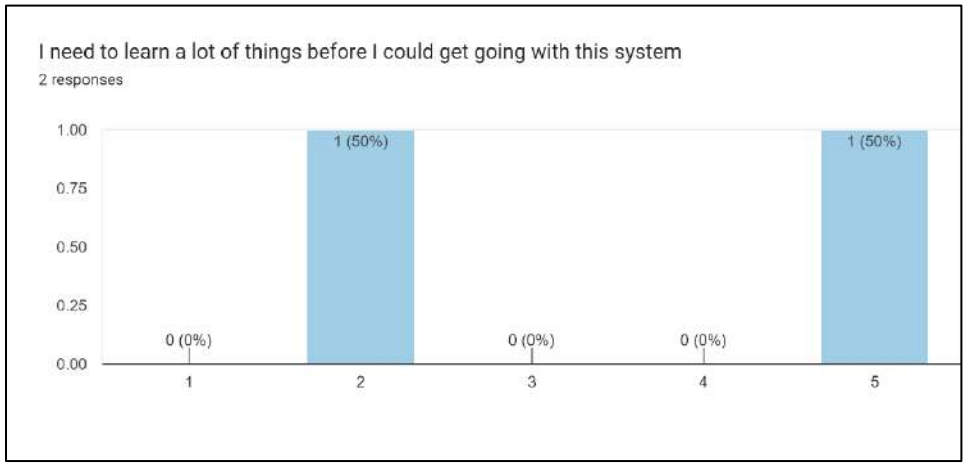

I found the various functions in this app were well integrated
2 responses
1.00
0.75
0.50
0.25
0.00
0 (0%)
0 (0%)
0 (0%)
1 (50%)
1 (50%)
1
2
3
4
5


I thought there was too much inconsistency in this app
2 responses
2
1
0
2 (100%)
0 (0%)
0 (0%)
0 (0%)
0 (0%)
1
2
3
4
5


I would imagine that most people would learn to use the app very quickly
2 responses
2
1
0
0 (0%)
0 (0%)
0 (0%)
0 (0%)
2 (100%)
1
2
3
4
5


I found the app very awkward to use
2 responses
1.00
0.75
0.50
0.25
0.00
1 (50%)
1 (50%)
0 (0%)
0 (0%)
0 (0%)
1
2
3
4
5


I felt very confident using the app
2 responses
2
1
0
2 (100%)
0 (0%)
0 (0%)
0 (0%)
0 (0%)
1
2
3
4
5


I need to learn a lot of things before I could get going with this system
2 responses
1.00
0.75
0.50
0.25
0.00
1 (50%)
1 (50%)
0 (0%)
0 (0%)
0 (0%)
1
2
3
4
5