The Open University of Hong Kong
School of Science and Technology
Bachelor of Computing with Honours in Internet Technology

COMPS456F – Software System Development Project

Final Year Project – Final Report

Elderly Mobile Life Assistant for Family Members with Extra Peace of Mind

Group Name: Whale Did You Go? Supervisor: Dr. Kelvin Kai Wing LEE

Student Name	Student ID Number	Signature
Leung Kwan Ho	12123629	
Au Chi Chung	12017765	
Lie Tze Wah	12123709	

We are declare that this is a group project and that no part of this submission has been copied from any other student's work or from any other source except where the due acknowledgment is made explicitly in the text, nor has any part been written for us by another.

Abstract

This project titled “Elderly mobile life assistant for family members with extra peace of mind” commences on 1-9-2019 under the supervision of Dr. Kelvin Kai Wing LEE. The project was initiated due to an observation that smartphone has a lot of features and applications which are not useful for the elderly, and the issue of elderly safety is common in Hong Kong. An Android application for the elderly can be developed with a special user interface that had large and few numbers of icons. Functions may include the single-touch dialing to another phone number until successful, instant messages, alarms, fall-detection, and alerts. Those features can build the connection and allow customizations to make it suitable for the use of the family members of the elderly, which also improving the elderly safety by using the mobile application. There were two parts to the study. Part one investigated the issue and the background of the project. Part two looked at the project development including program structure and requirements. The results indicated that system integration is feasible.

Declaration

We, Leung Kwan Ho (12123629), Au Chi Chung (12017765), Lie Tze Wah (12123709), and we have utilized the guidance of our supervisor in completing this project, and that the content which is not our own has been attributed and referenced properly. There should be no copyrighted content without permission to use. There should be no confidential data.

We declare that the description and information outlined in the individual team member reports are true reflection of the project status to the best of our knowledge.

Signature and Date

Leung Kwan Ho: _____

Au Chi Chung: _____

Lie Tze Wah: _____

Acknowledgement

We wish to express our deepest gratitude to our supervisor Dr. Kelvin Kai Wing LEE first. His knowledge of application development has been a valuable source of inspiration for the development of theories. Throughout the process, he provided some appropriate opinions to solve the confusion from us. After that, we wish to thank Dr. Andrew Kwok Fai LUI because of the teaching of the basic concept in the workshop for the final year project development including report writing and presentation skills. We also like to thank our classmates and parents who kindly helped to toke a set of unit testing. Our parents are the ultimate source of support so that we can complete the project at home for the mobile application development during the outbreak with Coronavirus Disease 2019. And their endless encouragement and patience have given us the confidence to overcome so many difficulties.

Table of Contents

Abstract 1

Declaration 2

Acknowledgement 3

List of Tables 3

List of Figures 4

1. Chapter 1 – Introduction 6

1.1. Overview 6

1.2. Project Aim 6

1.3. Project Objectives 7

1.4. Impact and Value Propositions 7

1.5. Project Structure 7

2. Chapter 2 – Background or Literature Review 8

2.1. Problem Analysis 8

2.2. Supporting Literature review for Related Technologies 8

2.3. Review of Existing or Related Solutions 9

3. Chapter 3 – Methodology 10

3.1. Overview of Methodologies 10

3.2. Requirements, Supporting Technologies, and the Technical Gap 11

3.2.1. Functional requirement 11

3.2.2. Non-functional requirement 13

3.3. System Limitation and Technical Gap 13

3.4. Design 14

3.4.1. Use case diagram 14

3.4.2. Component diagram 15

3.4.3. Data flow diagram 16

3.5. Key Technologies 17

3.6. Detailed Design of Selected Components 18

3.7. Tests, Performance, and Evaluation Method 20

3.7.1. Unit Test 20

3.7.2. User Evaluation 23

4. Chapter 4 – Results: Prototype Implementation 28

4.1. Issues about Implementation 28

4.2. Description of the Prototype 29

4.2.1. Setup of the prototype application 29

4.2.2. User Interfaces 35

4.2.2.1. Login Page 35

4.2.2.2. Main Page 36
4.2.2.3. Daily Schedule 37
4.2.2.4. Single Touch Dialing 38
4.2.2.5. Instant Message 39
4.2.2.6. Fall Detection 40
4.2.2.7. Other Functions 41
5. Chapter 5 – Results: Evaluation 42
5.1. Evaluation Study #1 46
5.2. Evaluation Study #2 47
6. Chapter 6 – Conclusion 48
References 49
Appendix A. Team Member's Roles and Responsibility 50
Appendix B1. Team Member Leung Kwan Ho's Final Report 52
Appendix B2. Team Member Au Chi Chung's Final Report 56
Appendix B3. Team Member Lie Tze Wah's Final Report 62

List of Tables

Table 1 – Review of existing of related solutions 9

Table 2 – Requirements: Functional requirement 11

Table 3 – Requirements: Non-functional requirement 13

Table 4 – Use cases: Actor description 14

Table 5 – Unit tests 20

Table 6 – Appendix: Team member's roles 50

Table 7 – Appendix: Team member's responsibility 50

Table 8 – Appendix: Team member Leung Kwan Ho's final report 52

Table 9 – Appendix: Team member Au Chi Chung's final report 56

Table 10 – Appendix: Team member Lie Tze Wah's final report 62

List of Figures

Figure 1 – e-See Find [6] 9
Figure 2 – Senior Safety App [7] 9
Figure 3 – Super Seniors 9
Figure 4 – Overview of methodologies: Waterfall development [8] 10
Figure 5 – Design: Use case diagram 14
Figure 6 – Design: Component diagram 15
Figure 7 – Design: Data flow diagram 16
Figure 8 – Key technologies: Cloud service in the system development 17
Figure 9 – Detailed design: Reference coordinate system for device orientation [9] 18
Figure 10 – Detailed design: Fall pattern [4] 18
Figure 11 – Detailed design: Fall detection algorithms for body attached accelerometers [10] 19
Figure 12 – User evaluation: Part 1 23
Figure 13 – User evaluation: Part 2 24
Figure 14 – User evaluation: Part 3 25
Figure 15 – User evaluation: Part 4 26
Figure 16 – User evaluation: Part 5 27
Figure 17 – Setup of the prototype application: Step 1.1 29
Figure 18 – Setup of the prototype application: Step 1.2 29
Figure 19 – Setup of prototype application: Step 2.1 30
Figure 20 – Setup of prototype application: Step 2.2 30
Figure 21 – Setup of prototype application: Step 3.1 31
Figure 22 – Setup of prototype application: Step 3.2 31
Figure 23 – Setup of prototype application: Step 3.3 32
Figure 24 – Setup of prototype application: Step 3.4 32
Figure 25 – Setup of prototype application: Step 4.1 33
Figure 26 – Setup of prototype application: Step 4.2 33
Figure 27 – Setup of prototype application: Step 5 34
Figure 28 – User interfaces: Login page 35
Figure 29 – User interfaces: Main page 36

Figure 30 – User interfaces: Daily schedule 37
Figure 31 – User interfaces: Single touch dialing 38
Figure 32 – User interfaces: Instant message 39
Figure 33 – User interfaces: Fall detection 40
Figure 34 – User interfaces: Other functions 41
Figure 35 – Evaluation results: Part 1 42
Figure 36 – Evaluation results: Part 2 43
Figure 37 – Evaluation results: Part 3 45
Figure 38 – Evaluation study 1: Before test 46
Figure 39 – Evaluation study 1: After test 46
Figure 40 – Evaluation study 2: Before test 47
Figure 41 – Evaluation study 2: After test 47

1. Chapter 1 – Introduction

1.1. Overview

This project is to develop the mobile application for targeting the severity of elderly safety in Hong Kong due to negligent care with family members. Due to the issue of falls will increase the risk of serious injuries, chronic pain, long-term disability, loss of independence, and psychological and social limitations due to institutionalization. Older adults are hospitalized for fall-related injuries. A fall can cause psychological damage even if the person did not suffer a physical injury. The mobile application of the fall detection system or urgent call system has become popular among the Elderly Care Service Centre or family use for providing help and contact the user's family to update the user's status.

Because of many existing applications or the user interface of the operating system is not focusing on the main users for the elderly, and it is not integrated for achieving the auxiliary use for them. For this situation, our application releases to the public that users can use this application with the simple interface to get more important information and the message alert with their family that improving personal safety.

1.2. Project Aim

This project aims to integrate the functions with a simple user interface that can focus on the problem of elderly safety.

In addition to the basic functions of fall detection GPS monitoring for an easy way to alert the users with family members during an occurrence, the information such as the instant message of notification and the single-touch dialing for connecting the elderly with their family members are also important to them because they want to get the latest status for providing help for their family members. In the case of an elderly facing accident like a slip and has life-threatening, the family members need to know the location of the elderly. And it gave the original aim of GPS supported navigation system is to guide the family members to reach their destinations when they receive the emergency notice from their elderly member, it seems imperative to include the above-mentioned real-time information. Of course, the application will provide some simple functions for helping the elderly to use this application such as the event calendar with the daily to-do list features.

1.3. Project Objectives

To achieve the aim, the main objective of the project is to develop the mobile application which focusing the target user of the elderly and their family members that can provide life assistant to the elderly and the updated status of the elderly for other family members.

The project has defined several sub-objectives as follows:

1. Provide different functions for focusing on elderly safety.
2. Provide the real-time in-app / system-based notification to remind the elderly about the daily matters and the urgent alert for their family members.

Here are some details of the application design of the system architecture before development:

- Login using with Google account via Firebase service and Google API that it is unnecessary to develop a user authentication with registration
  - The reason behind this is due to the usage of the project that we found the account register features have complicated the system and the function is not useful for the user. Then using Google account for login is based on the user because they are already set up the Google account in the Android mobile phone when they need to use Google service. The aim of this change is that Google account can get the information of username and Google calendar which are the need for the system. At the same time, the Google calendar allows multi-user for the edition that it meets the requirement for the to-do list function which lets teenage members edit the calendar for the elderly.

1.4. Impact and Value Propositions

For the family that who have not enough time to take care of their elderly members and hope to reduce the possibility of accidents for the elderly and help the elderly to use their smartphones more easily. Our mobile application with the integration of the functions that helping the elderly in handling the crisis and assisting them in their daily lives.

1.5. Project Structure

The remainder of the report is structured as follows. The next chapter is a literature review that discusses the use of a mobile application for the issue of elderly safety. Those problems are described in detail, and the solutions are discussed and critiqued. The report then outlines the design of the application which decided from different system requirements and introduces the major components. Also, the processes designed for the functional requirement and the methodology are also highlighted. The report finally presents the results of implementation and user evaluation of the mobile application and concludes with a discussion of the merits and limitations of the system.

2. Chapter 2 – Background or Literature Review

2.1. Problem Analysis

To review different applications that promote the safety of the elderly, we found that most of them having some of the problems. For example, the applications will waste users a lot of time dealing with the different complicated user-interface and that users cannot use it intuitional. Also, some applications cannot let the user choose whether the fall-detection is open or not. And that, some fall detection functions may no reaction or cannot sense the fall when the user killed the application. This is the serious problem that it often lets the user adapt or learn how to control the system, and the system may be misguided the user when they fully believe with the system. And we design the main direction for the system – "Safety" and "Self-managed". And we hope that those of the direction can help us for creating a clear concept of the system.

The project has defined several problems as follows:

1. Similar systems in the market
   The mobile application promoting the safety of the elderly is not a new idea, there are many existing similar functions in the market but those are just providing only one single function.

2. Language barrier
   The application must be compatible with different languages for different users. It is because the product may provide different languages for the users.

3. The user feels difficult to use
   Some users may do not know the basic technique to control mobile devices. And they may feel hard to adapt the application for the first time for use.

2.2. Supporting Literature review for Related Technologies

Mobile Application for Elderly Safety

Given different literature reports, we found that the following applications can solve the above problems. One of the news articles about how senior safety app now tracks senior falls and inactivity written by Devika Manghnani, which mentation that "As the population continues to age, falls have increasingly become a problem for seniors. As seniors grow older and more unsteady, they are more likely to suffer from a fall. Falls for seniors can lead to broken bones and blood clots, as well as Head and spinal injuries. Seniors who are affected by Parkinson's Disease, dementia, or stroke are even more likely to fall." [1]. Recently, some applications can allow the family members to detect a possible fall as the application will alert them if the phone falls to the ground. Also, family members can determine if a phone becomes inactive [2].

Solution for User Feels Difficult to Use of Mobile Applications

For another academic paper of search and it contains the solution for solving the problem that the user feels difficult to use of the mobile application. Regarding the works that study the usability for older people, the writer made a review of different studies in this sense. This study shows that older people require more time to complete tasks on mobile devices and it describes problems such as the size of the screen to

read information, the size of menus and interfaces to enter data such as virtual keyboards, functionalities such as the drag and drop, the size of the target (the older tend to make errors when tapping a small target), the gap between intended and actual touch locations (the older tend to miss their intended targets due to parallax and the large contact area of each finger) [3]. Besides, the writer mentation an experience from which they extract that for older people there are also other issues very important related to mobile devices. For example, characters in the screen easy to read, buttons easy to use, that the device was easy to learn and operate, keep in contact, good sound quality.

Theory of Fall-detection Sensor in Android Mobile Device

A typical Android smartphone is loaded with a variety of sensors. Accelerometer, magnetometer, proximity sensor, light sensors are some of the sensors available which can be used to detect postures and potentially detect events like fall [4]. The fall detection is used to monitor the elderly for possible falls so that they can be safe. It is used to detect and alert the caretakers about the area in which the fall occurred. All the Android devices are in-built with sensors like accelerometer and gyroscope. These can also be used as ankle monitors. The sensors are used to predict fall detection based on the change in the angle of the mobile [5]. The accelerometer of Android mobile devices used is a 3-axis accelerometer that can measure linear acceleration in x, y, z mutually perpendicular axes. The phone is assumed to be in a position for the algorithm to compute postures. Typically, the phone assumed to be in the waist pocket region. The algorithm is tuned to this position and thus all computation is based on this assumption [4].

2.3. Review of Existing or Related Solutions

Table 1 – Review of existing of related solutions

Logo / Name	Figure 1 – e-See Find [6] e-See Find by Senior Citizen Home Safety Association	Figure 2 – Senior Safety App [7] Senior Safety App by SENIORSAFETYAPP & 97 Technologies	Figure 3 – Super Seniors **Our Application** **Super Seniors** by Final Year Project Team Whale Did you Go
Features	▪ Can enquire for the elderly's device location anytime and anywhere	▪ Alert multiple people ▪ Automatic fall alerts ▪ Prevent wandering ▪ Inactivity alerts ▪ Low battery alert ▪ Prevent online abuse	▪ **Simple user interface** ▪ **Single touch dialing** ▪ **Fall detection with GPS Monitoring** ▪ **Daily calendar** ▪ **Instant messages**
Login Method	Phone number & Password		**Google account login**
Problem	Only can used by up to 5 designated emergency contacts for query the location	Need to setup the emergency, reporting and alerts with the complex steps at the first time	

3. Chapter 3 – Methodology

3.1. Overview of Methodologies

In this system integration project, the overall methodology is build-and-test and we decide to use Waterfall Development which is simple to understand and use.

Waterfall development allows for departmentalization and control and which can define stage and arrange tasks clearly. In the process model, it can set with deadlines for each stage of development and a product can proceed through the development process model phases one by one. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance.

The following illustration is a representation of phases for the Waterfall Model:

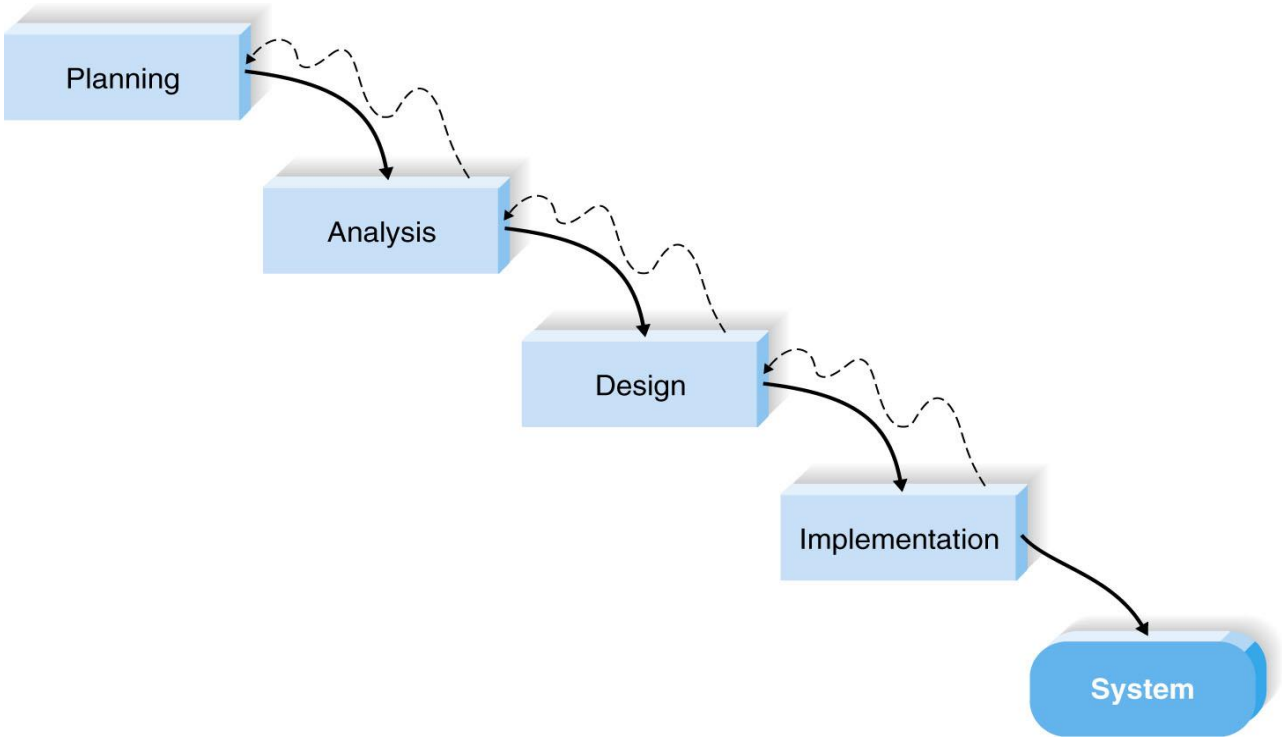


Figure 4 – Overview of methodologies: Waterfall development [8]

3.2. Requirements, Supporting Technologies, and the Technical Gap

3.2.1. Functional requirement

Table 2 – Requirements: Functional requirement

Functional Requirement	
Functions	*Function description*
Basic function	
1. User authentication (Sign-in and Sign-out)	Purposes: The user needs to use their Google account to login to the system before they need to use the functions of the system. And the authenticate using Firebase Auth user interface for the login that it can get the personal information of username, e-mail address, google calendar details via Google Service API, etc.
2. Multi-Language (English and Traditional Chinese)	Purposes: To compensate for the shortcomings of user-friendly and solve the problem of the language barrier, the application must be compatible with different languages for different users. We decided to provide some major language in the application that it can let users operating with the default language from their mobile or they can set up by the in-app option.
Main function	
1. Single touch dialing	Purposes: The elderly user can the single touch dialing function from the dashboard page for phone calls with the pre-saved number which contains the information of avatar, nickname, and contact number. Users can setup up to three contacts of the single touch dialing then take a phone call for the pre-saved contact. And that, this function can take a re-call action to the other party's phone Until the other party gets to the phone. Aims: ▪ Elderly daily life management.
2. Instant messages	Purposes: The Instant message connecting with the pre-saved contact from the function of single touch dialing. And the elderly user can use it to send an SMS message by selecting simple option buttons such as message details with the date time and some short sentence. It divided into three types of SMS, which are outgoing, shopping needs, and alerts. Aims: ▪ Elderly daily life management ▪ Elderly safety.

3. Daily calendar	Purposes: The daily calendar function connecting the Google calendar with Google account via Google Service API. It uses for the reminder to elderly users that it can create, update, or delete the event planning synchronously with the Google calendar. And it provides the to-do list function with the SMS notification that users can mark down the item he is already completed that it also avoids the accidents that elderly missing or misuse of drugs. Aims: ▪ Elderly daily life management ▪ Elderly safety
4. Fall-detection with GPS monitoring	Purposes: The fall-detection provides the alert with a ringtone from the user's phone when facing the fall that people nearby him can know the accident. It provides the on-off switch for the detection that the user can choose whether to turn on the detection. And that, it will send the SMS message to the family members in the pre-saved emergency contact list when the elderly facing fall accidents after countdown timer with no response by the user in 15 seconds. The SMS message containing the current location of the sender that other family members can get the location of him by clicking the link of Google map. Aims: ▪ Elderly safety

3.2.2. Non-functional requirement

Table 3 – Requirements: Non-functional requirement

Reliability and performance requirement	
Mobile Devices: ▪ must stay connecting with network service (Wi-Fi / 4G)	Purposes: Stay connecting with the network can confirm that the application user can already accept the urgent call, and the alert SMS.
Hardware requirement	
For development of the application: Operating System: Linux / MacOS X 10.5.8 or higher / Microsoft Windows XP or higher Platform: Android SDK Framework 10 or higher Tools: Android Studio, ADT for Android Studio Android Emulator: SDK API level 28 (Android 9) Minimum requirements of mobile devices: ▪ Android device ▪ Android 7.0 (API level 24 or higher version) ▪ 5.0 MB Minimum space to execute	Purposes: Base on the version requirement of the listener when using different built-in functions in Android mobile devices such as dial function, accelerometer, GPS, and calendar read/write function, etc. That it must be using the Android device on Android 7.0 or higher version. And the emergency contact and the contact information in a single touch dialing function will use the built-in memory in mobile.
Required implementation language and cloud service	
Programming language: ▪ Java and xml (Android Studio) Cloud service: ▪ Firebase with Google service API	Purposes: Using Android Studio for the system development that it only using the programming language of Java and XML for the development of this project. To using Google account for the system login, that we decided to use Google firebase could platform for the data storage at the same time.

3.3. System Limitation and Technical Gap

The project has defined several system limitations as follows:

1. Equipment restriction

   Users may have no mobile device, or they are using a different type of mobile with a different operating system that they cannot use the product.

2. Network restriction

   Users may not have internet service, or their mobile device cannot connect with the internet, that the function of the application will be affected. Such as the system cannot get the user status or in-app notification via the network.

3.4. Design

3.4.1. Use case diagram

The following diagram represents the relationship between different system users and the application:

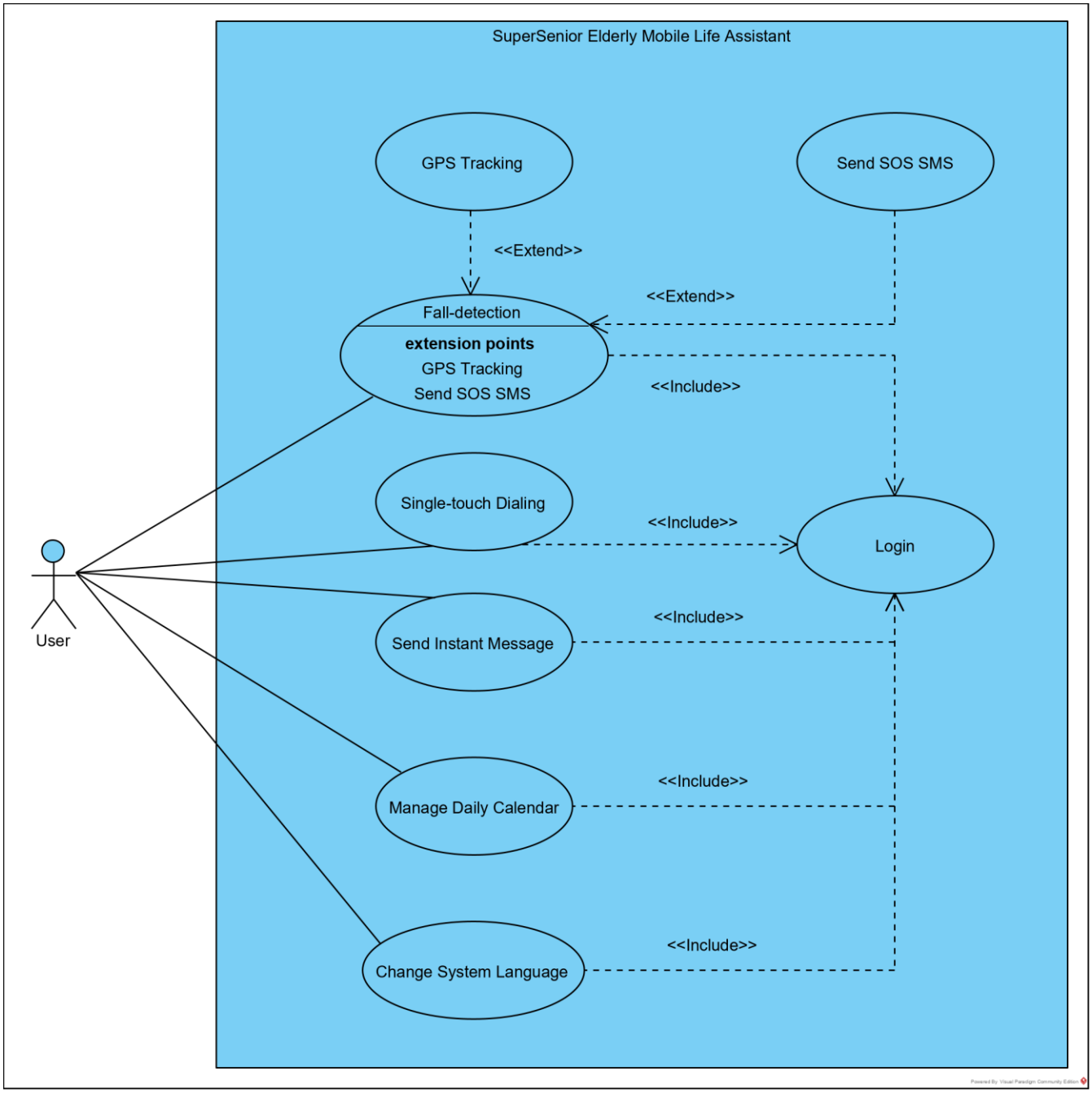


Figure 5 – Design: Use case diagram

Table 4 – Use cases: Actor description

Actor	*Definition*
User	Users can: ▪ Browse, create, update, delete calendar items ▪ Single touch dialing to pre-saved contact number ▪ Send an instant message to the pre-saved contact ▪ When the user has fall accident, the system will give off the alert and send out the urgent notification and GPS location to the family members ▪ Change the display language by an in-app preference setting

3.4.2. Component diagram

The following diagram describes the relationship between organizations and components in system:

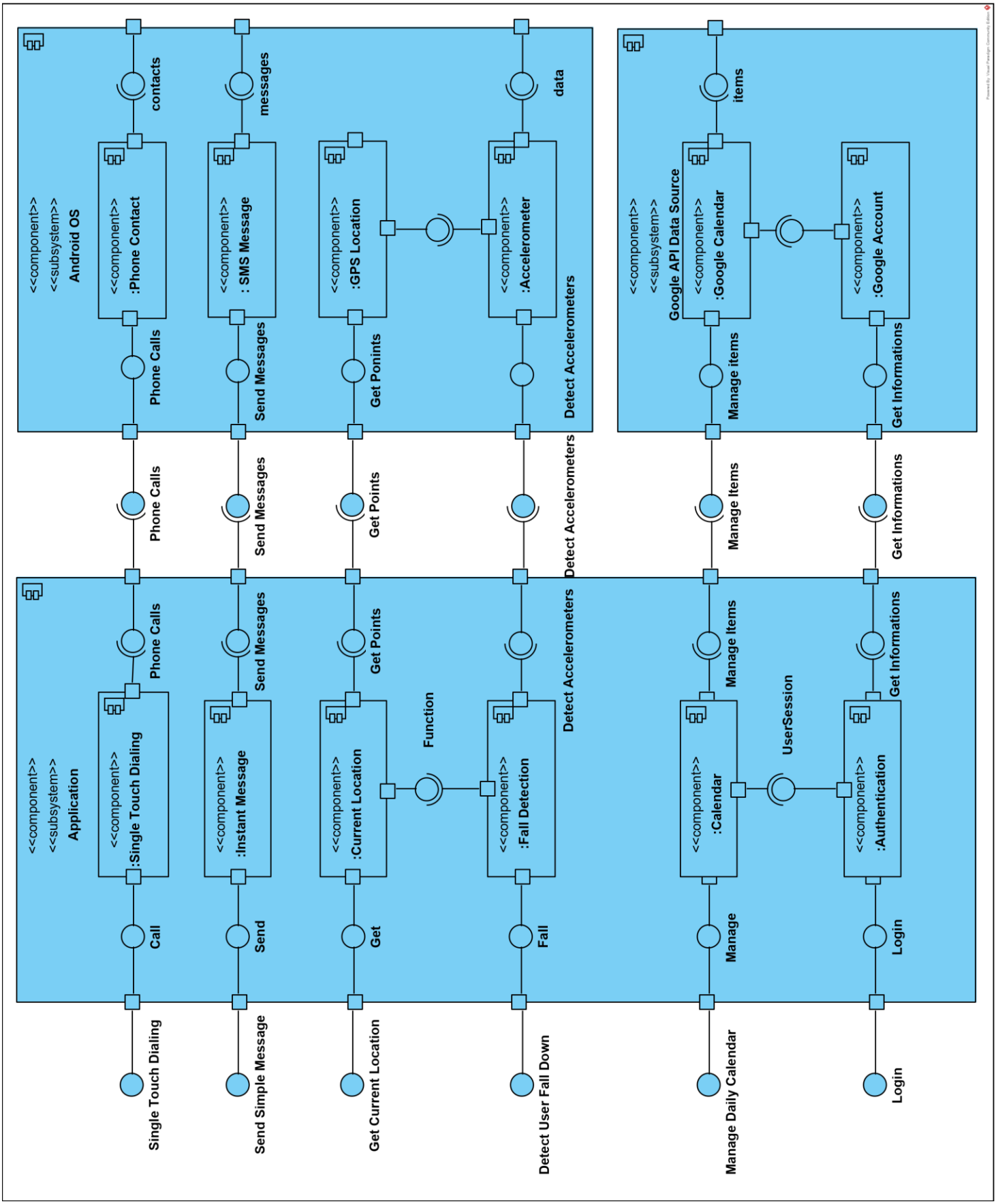


Figure 6 – Design: Component diagram

3.4.3. Data flow diagram

The following diagrams represent the flow of data in the whole system with different system user:

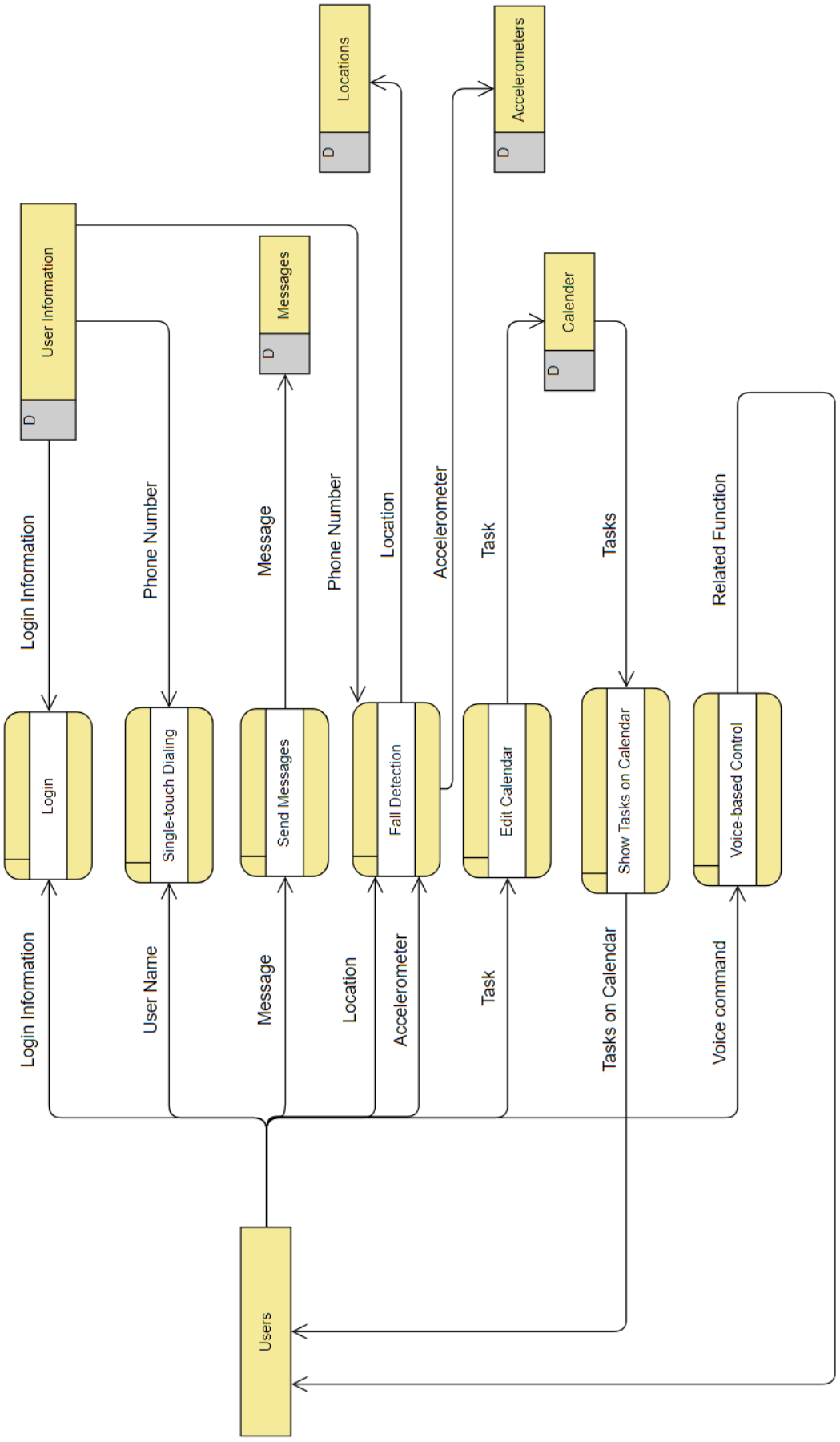


Figure 7 – Design: Data flow diagram

3.5. Key Technologies

For the programming language of the application development that we are using Android Studio for building with Java, xml, SQL, etc. To using Google account for the application login feature that we decided to use Google cloud platforms with Firebase at the same time, that the project prototype implementation is dependent on those structure.

Firebase is a set of mobile and web application-oriented development platforms provided by Google. It has various back-end functions such as account registration and third-party login, NoSQL database, audio, and video storage space, etc. And Firebase is more like a backend service that provides apps and web pages to quickly come up with a simple layout architecture, also it is easier to use comparing with Google Cloud Service (GCS).

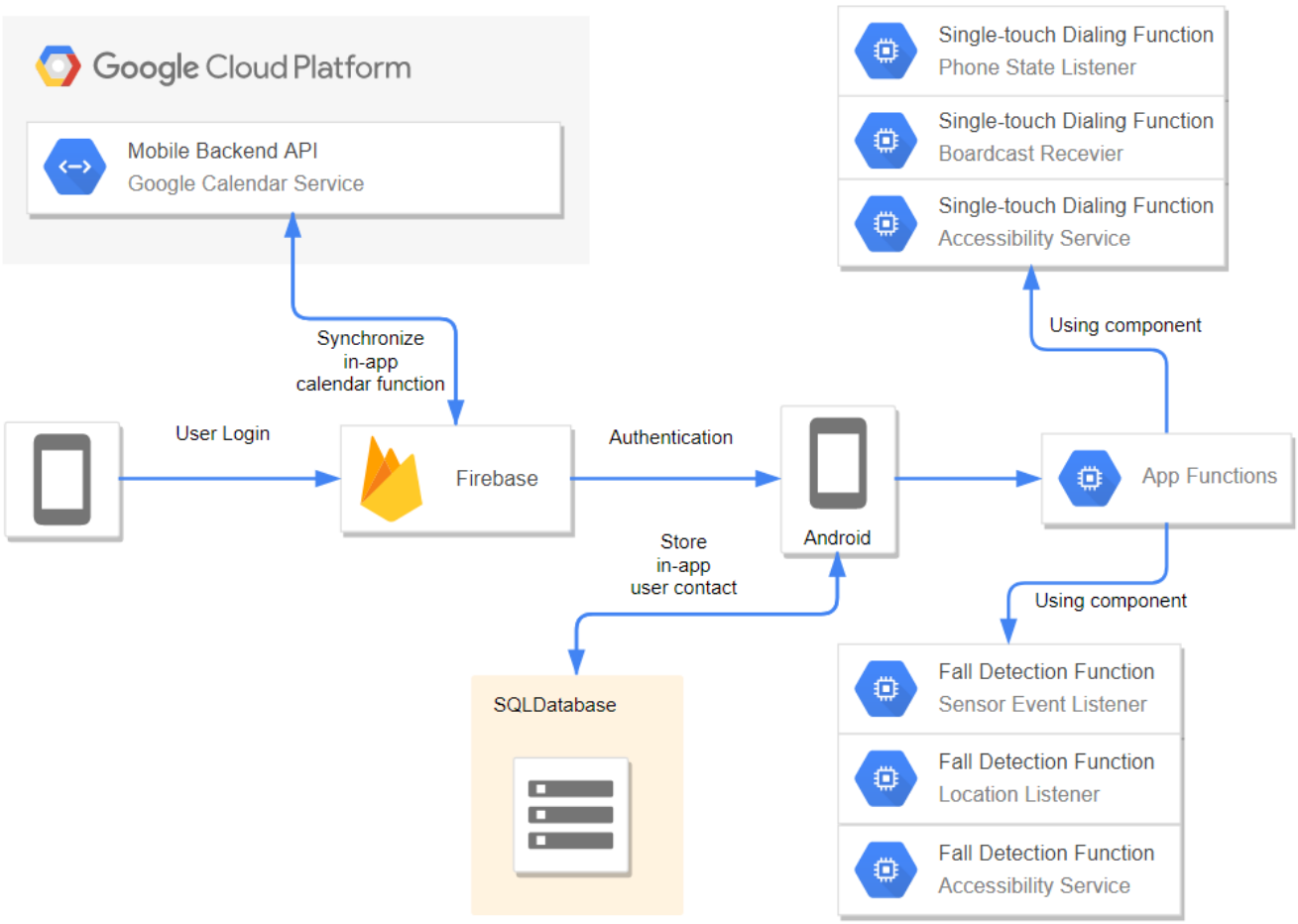


Figure 8 – Key technologies: Cloud service in the system development

3.6. Detailed Design of Selected Components

For detecting a fall event, the application monitors the Accelerometer and Magnetic field sensors in the background. A fall event is characterized by the detection of changes like fall patterns in both acceleration and orientation occurring within a short interval. Android API device sensors are exposed to a class called SensorManager. As we need to check the acceleration and orientation changes, the SensorManager is used to verify the accelerometer and magnetic field sensors. Upon the change in sensor values, an object of class SensorEvent is passed to the sensor event listener method [5].

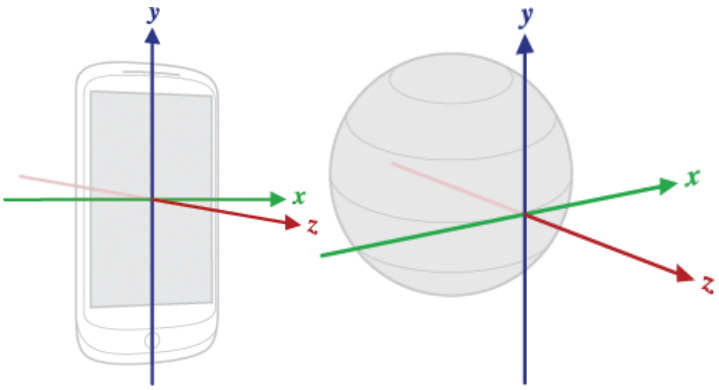


Figure 9 – Detailed design: Reference coordinate system for device orientation [9]

The SensorEvent class holds information like sensor type, sensor's data, event timestamp, and accuracy. We compute the norm of the Gx, Gy, and Gz that those are used by posture recognition and the fall detection module [4]. When received in the sensor listener contains an array of sensor values. For the accelerometer, the values that are returned are in m/s2 and Acceleration minus Gx (x-axis), Gy (y-axis), and Gz (z-axis). The acceleration applied to the divide is measured using the Accelerometer sensor and the measured acceleration is influenced by the force of gravity. Finally, the orientation of the device is calculated using the values obtained by both accelerometer and magnetic field sensors [5].

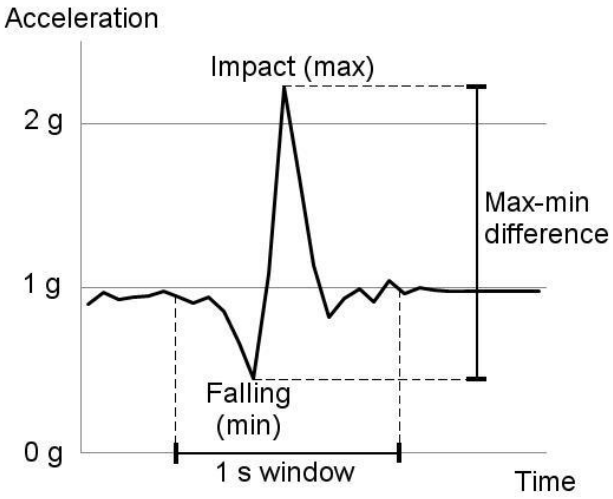


Figure 10 – Detailed design: Fall pattern [4]

If the difference between consecutive minima and maxima is greater the 2g that the output is decided as a fall. That the algorithm can calculate the event of a fall is based the output of both posture recognition and fall decision module. [5]

For the algorithm details on fall detection, it based upon the algorithm described in "Comparison of low-complexity fall detection algorithms for body attached accelerometers" authored by Maarit Kangas, Antti Konttila, Per Lindgren, Ilkka Winblad, Timo Jamsa and published in Gait & Posture 28 (2008) by Elsevier.

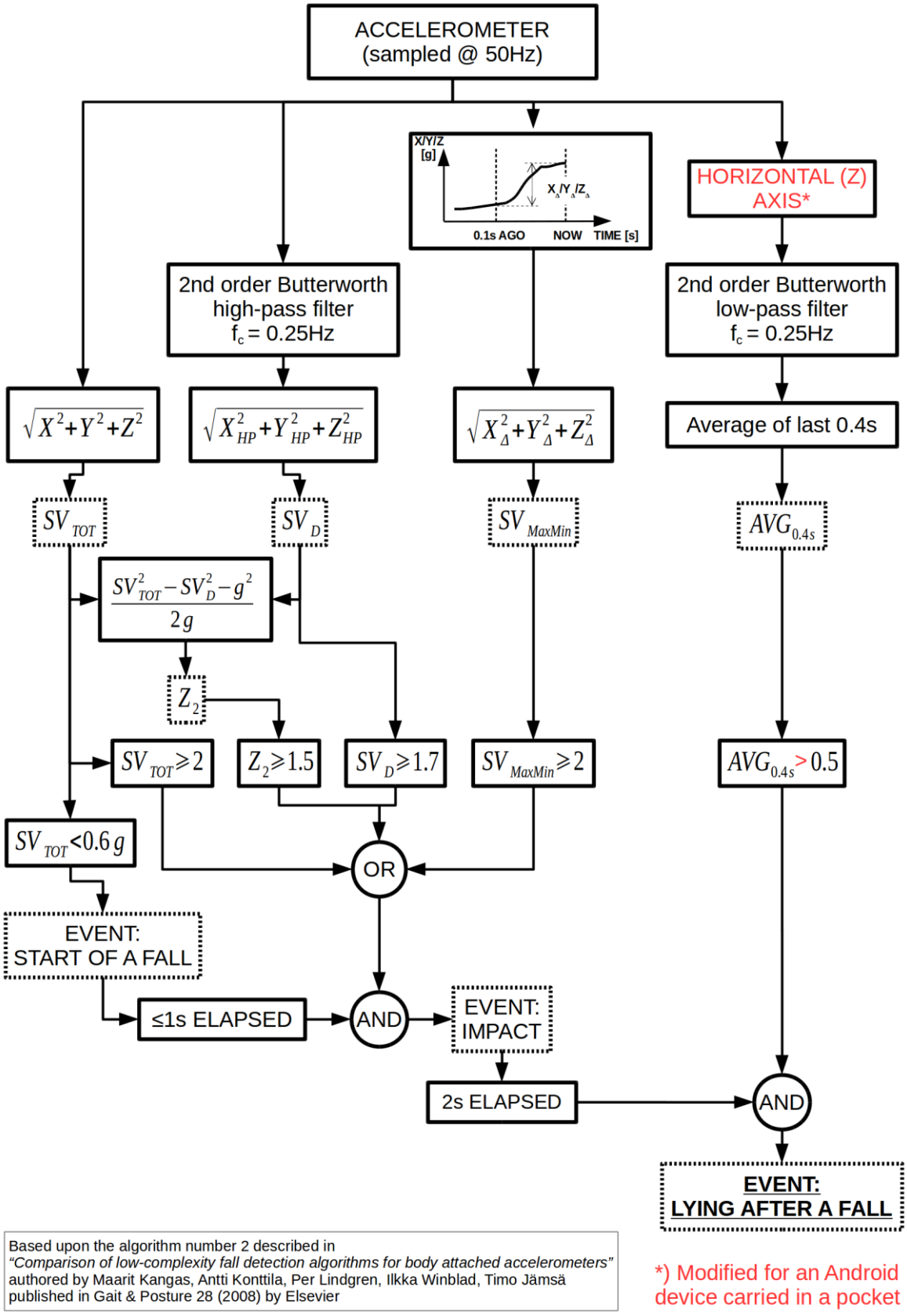


Figure 11 – Detailed design: Fall detection algorithms for body attached accelerometers [10]

3.7. Tests, Performance, and Evaluation Method

3.7.1. Unit Test

Table 5 – Unit tests

Function	Test Case	Expected Result	Observed Result	Result
Login Page	On load of startup screen	Display the login page after splash animation	Display the login page after splash animation	True
Login Page	Request different unapproved permission when the first installation	Request different unapproved permission	Request different unapproved permission	True
Login Page	After login with Google account	Show the main activity page and display the toast message with username and email address	Show the main activity page and display the toast message with username and email address	True
Login Page	Logout	Logout with the Google account then move to the login page	Logout with the Google account then move to the login page	True
Multi Language	Change the language to “English” using in-app option button (turn off the accessibility service)	Display the language by English	Display the language by English but no saved the setting after next open	False
Multi Language	Change the language to “Chinese” using in-app option button (turn off the accessibility service)	Display the language by Traditional Chinese	Display the language by Traditional Chinese but no saved the setting after next open	False
Multi Language	Change the language to “English” using in-app option button (turn on the accessibility service)	Display the language by English	Display the language by English and saved the setting after next open	True
Multi Language	Change the language to “Chinese” using in-app option button (turn on the accessibility service)	Display the language by Traditional Chinese	Display the language by Traditional Chinese but saved the setting after next open	True
Multi Language	Perform the language by user’s system default language in mobile	Perform the language by user’s system default language	Perform the language by user’s system default language	True
Calendar	After connected with the Google Calendars	Display the event items	Display the event items	True

Calendar	Create a new event	Item display in the fragment in schedule page with related date	Item display in the fragment in schedule page with related date	True
Calendar	Update the details of existing event	The details are updated	The details are updated	True
Calendar	Delete the existing event	The selected event is deleted	The selected event is deleted	True
Calendar	Time up for the event	Receive the SMS for the reminder	Receive the SMS for the reminder	True
Single Touch Dialing	Setup the contact with photo, name, phone number	The dashboard fragment displays the photo after saved	The dashboard fragment displays the photo after saved	True
Single Touch Dialing	Click the photo then update the contact details	The related details are updated	The related details are updated	True
Single Touch Dialing	Click the photo then take a single touch dialing	Move to the phone dialing interface then keep dialing until the other party answers the call	Move to the phone dialing interface then keep dialing until the other party answers the call	True
Instant Message	Select the related buttons with short sentence option in message dialog	Send the SMS message to the selected contact number in pre-saved contact list from single touch dialing	Send the SMS message to the selected contact number in pre-saved contact list from single touch dialing	True
Instant Message	Select with the space outside the dialog	No SMS action	No SMS action	True
Fall Detection	Create the emergency contact with name and phone number	The emergency contact list including the contact	The emergency contact list including the contact	True
Fall Detection	Click the button of list emergency contact	List the pre-saved contacts	List the pre-saved contacts	True
Fall Detection	Delete the emergency contact	The selected emergency contact is deleted in the real time	The selected emergency contact is deleted in the real time	True
Fall Detection	Turn off the fall detection by on-off switch	The fall detection is closed	The fall detection is closed	True
Fall Detection	Posture recognition: Walking	No fall alert	No fall alert	True
Fall Detection	Posture recognition: Standing	No fall alert	No fall alert	True
Fall Detection	Posture recognition: Sitting	No fall alert	No fall alert	True

Fall Detection	Posture recognition: Shaking	No fall alert	No fall alert	True
Fall Detection	Posture recognition: Fall	Perform fall alert and display count down timer	Perform fall alert and display count down timer	True
Fall Detection	Posture recognition: No Action	No fall alert	No fall alert	True
Fall Detection	Fall when the application showing in frontend	Perform fall alert and display count down timer	Perform fall alert and display count down timer	True
Fall Detection	Fall when the application is hiding (turn off the accessibility service)	Pop up the application then perform fall alert and display count down timer	Perform fall alert	False
Fall Detection	Fall when the application is hiding (turn on the accessibility service)	Pop up the application then perform fall alert and display count down timer	Pop up the application then perform fall alert and display count down timer	True
Fall Detection	Fall when the application is killed (turn off the accessibility service)	Pop up the application then perform fall alert and display count down timer	No state	False
Fall Detection	Fall when the application is killed (turn on the accessibility service)	Pop up the application then perform fall alert and display count down timer	Pop up the application then perform fall alert and display count down timer	True
Fall Detection	Fall occurred	Display the countdown timer from 15 seconds and alert with user's ringtone	Display the countdown timer from 15 seconds and alert with user's ringtone	True
Fall Detection	End of count down after fall action	Send SMS to pre-saved emergency contact and keep alert ringtone until the user close the alert manually	Send SMS to pre-saved emergency contact and keep alert ringtone until the user close the alert manually	True
Fall Detection	Sent the emergency SMS message	Message including the link of Google map with current location from sender	Message including the link of Google map with current location from sender	True

3.7.2. User Evaluation

In this project, we aim to solve the user problem that we need to design a solution based on technologies such as the integration of technologies into a prototype system. And the aim of user evaluation is to collect evidence to evaluate whether the user problem is solved or alleviated. For our user evaluation, user problems and experiences are where the objectives of the project are derived. In view of the situation, we decided to use the Pre-Post Study method via Google Form to select 8 users for a trial of the prototype application.

Related hyperlink of the user evaluation form:
https://docs.google.com/forms/d/e/1FAIpQLScme723B2FfZGRyv4mcq3jZsJvxJBCZEM0QFFJyi_jv7TL4EA/viewform

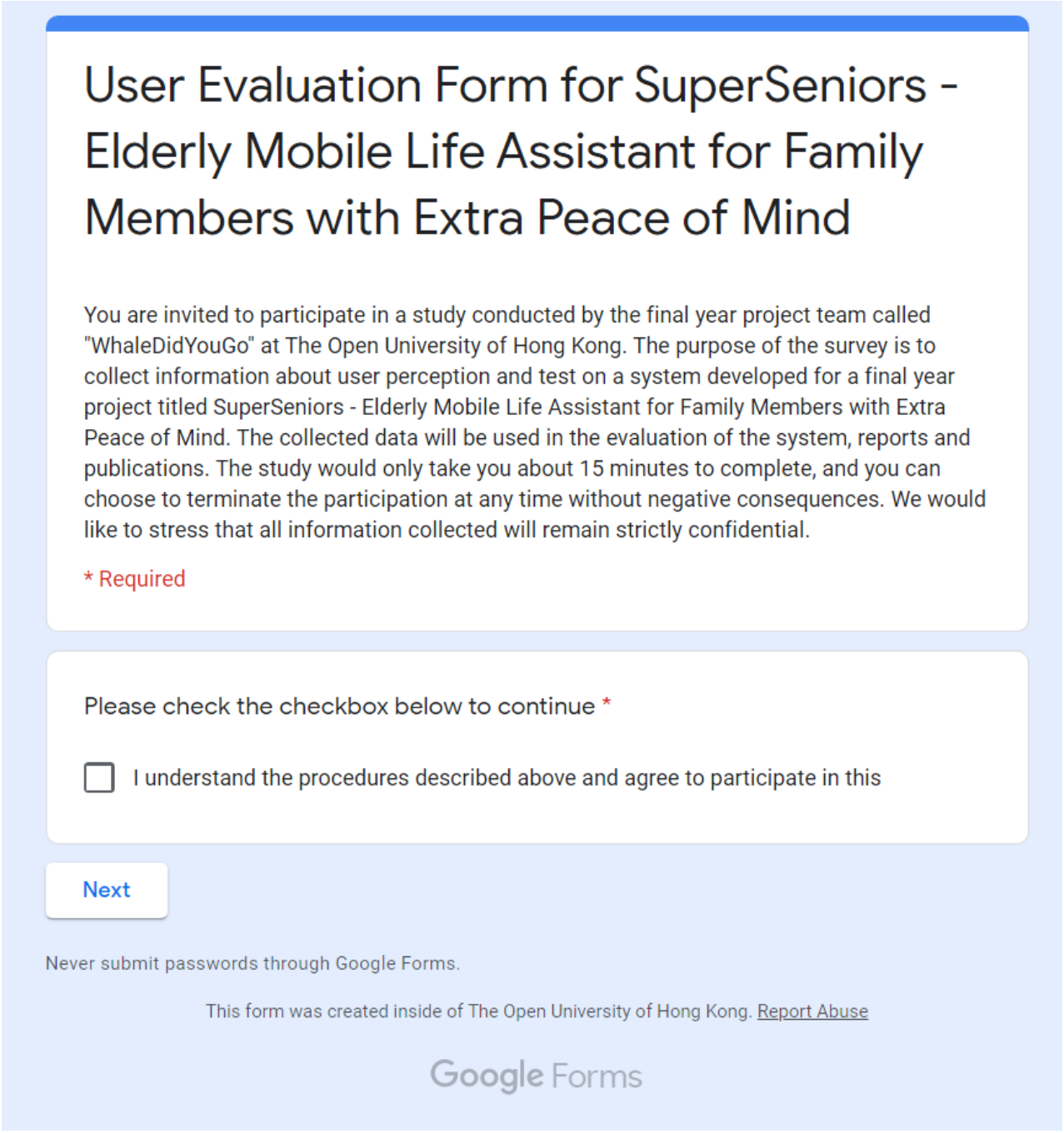


Figure 12 – User evaluation: Part 1

Then, the selected users need to take a survey with the sample question which asking their view before using the prototype application.

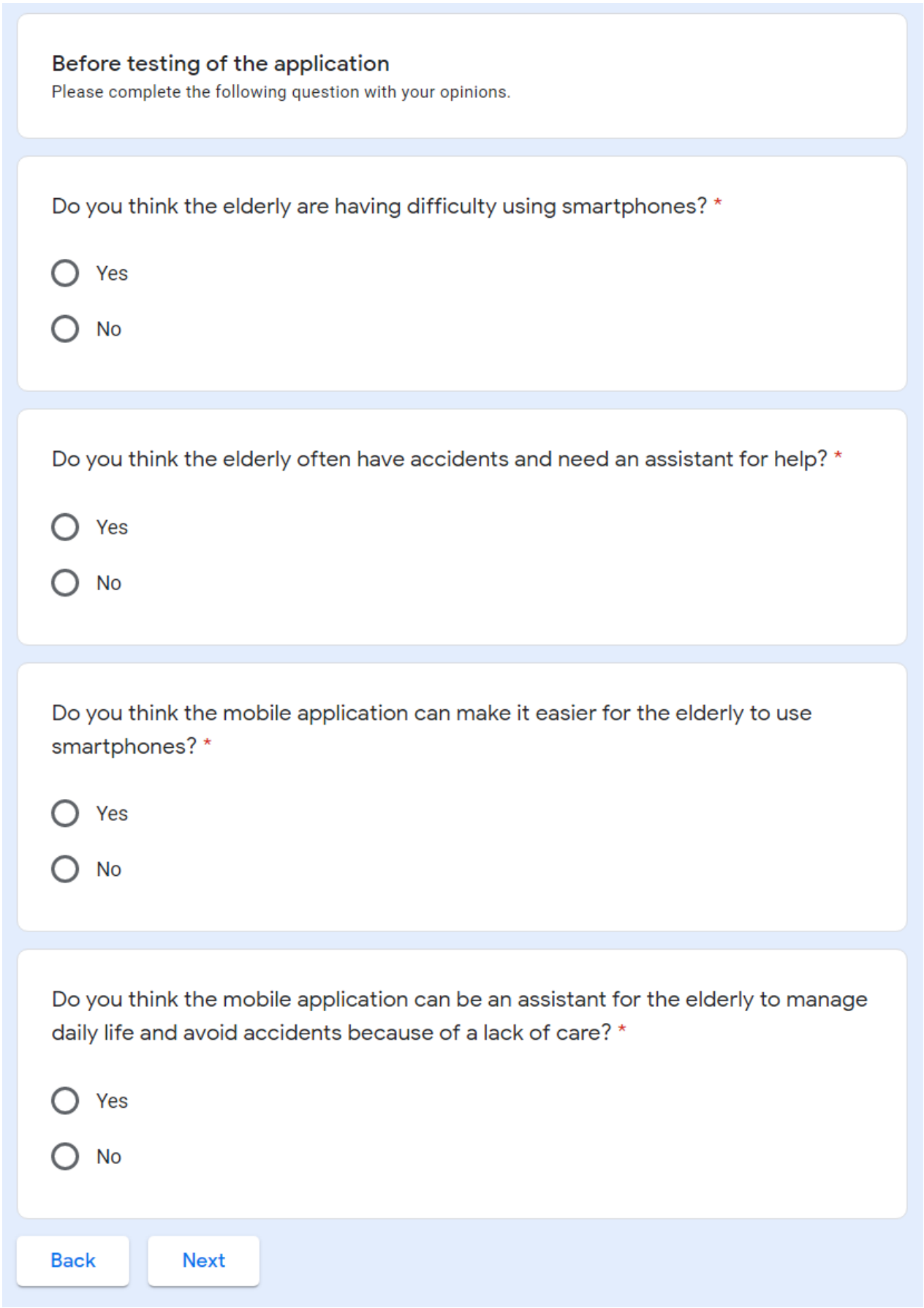


Figure 13 – User evaluation: Part 2

After that, they need to download the application from the hyperlink and test of the prototype application.

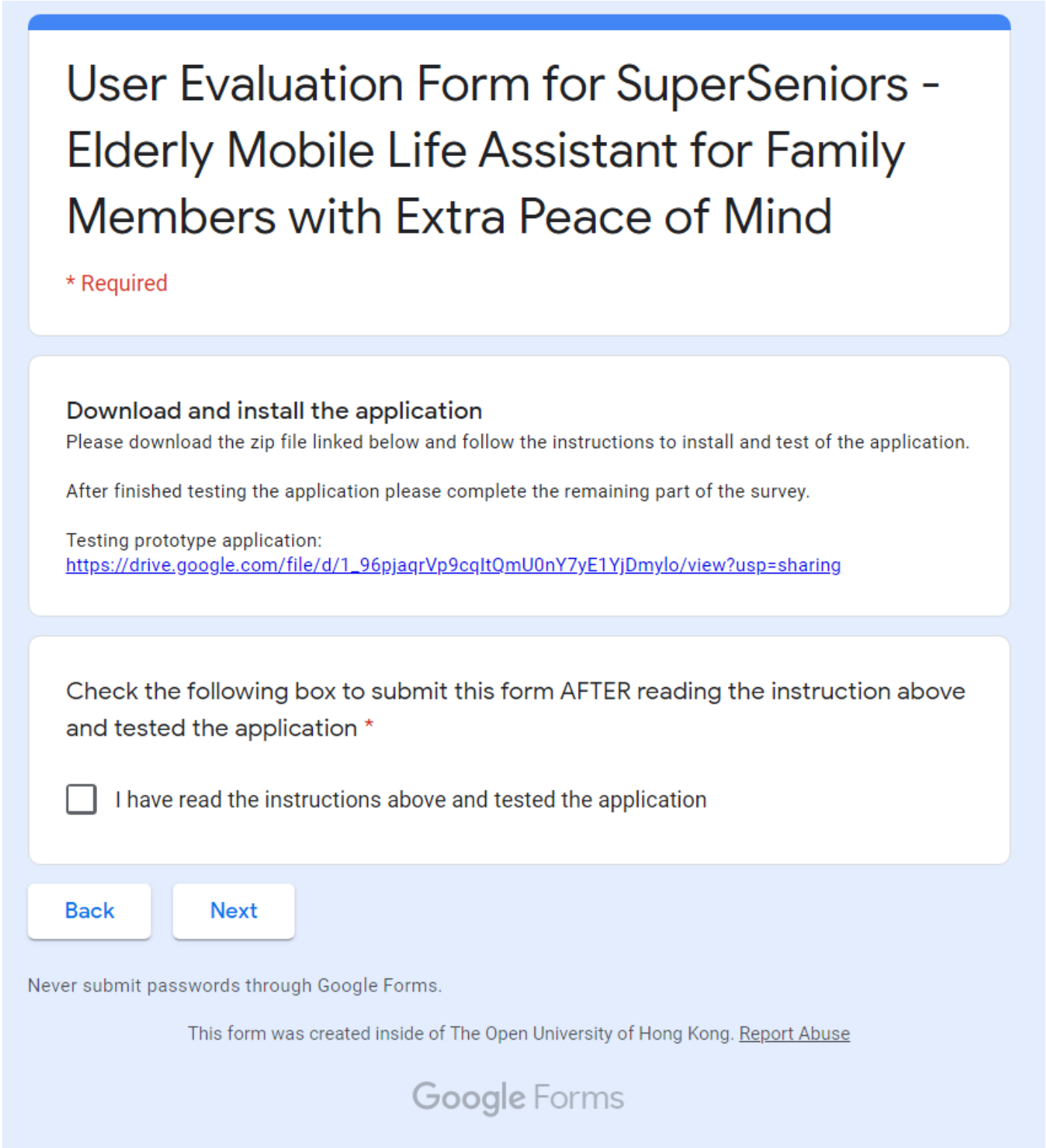


Figure 14 – User evaluation: Part 3

After the application test, they need to take the survey with the same question above (User evaluation form part 2) that the result can show the change and response for our application whether changed their mind.

Finally, they need to give a feedback for our application with the following questions:

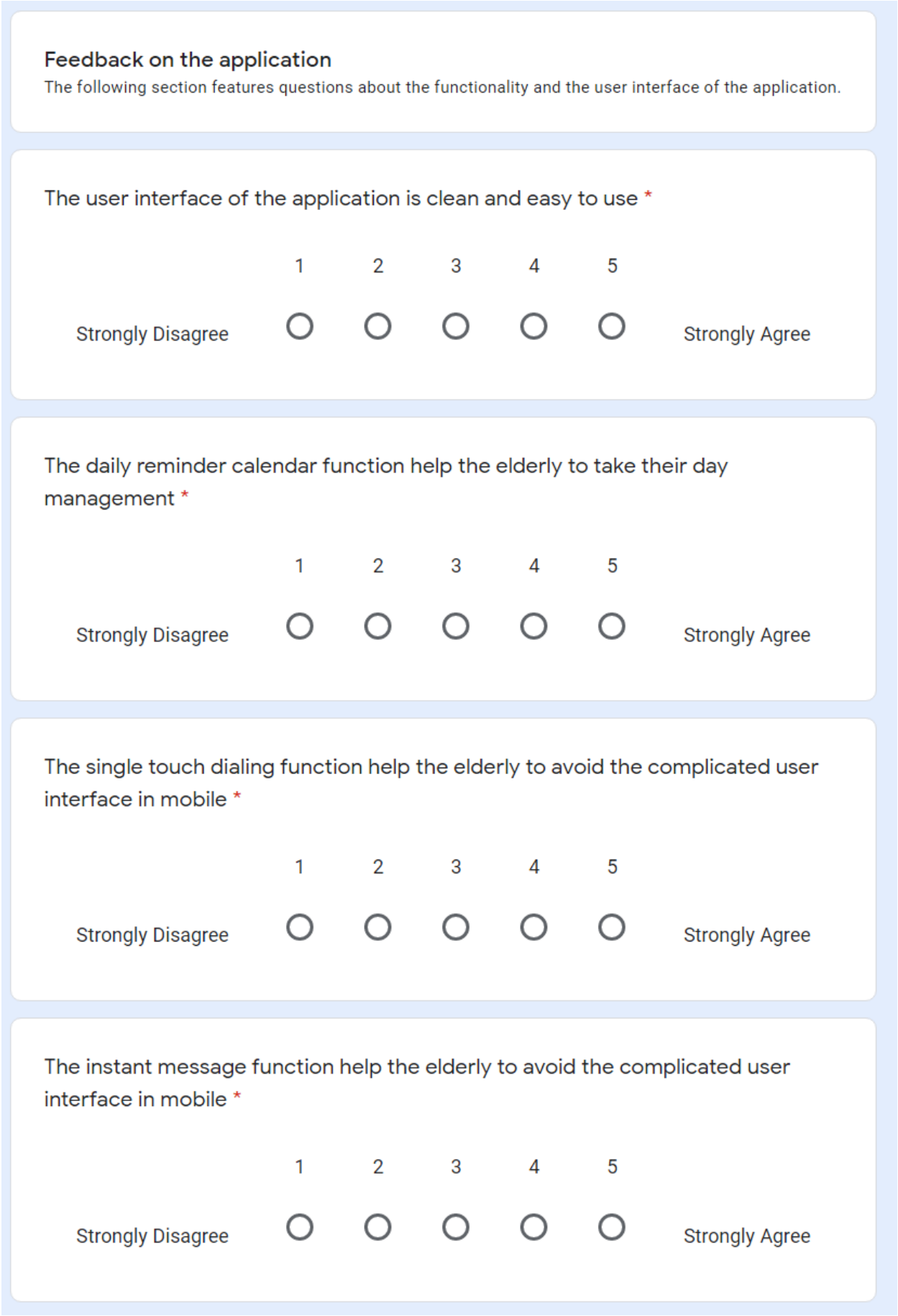


Figure 15 – User evaluation: Part 4

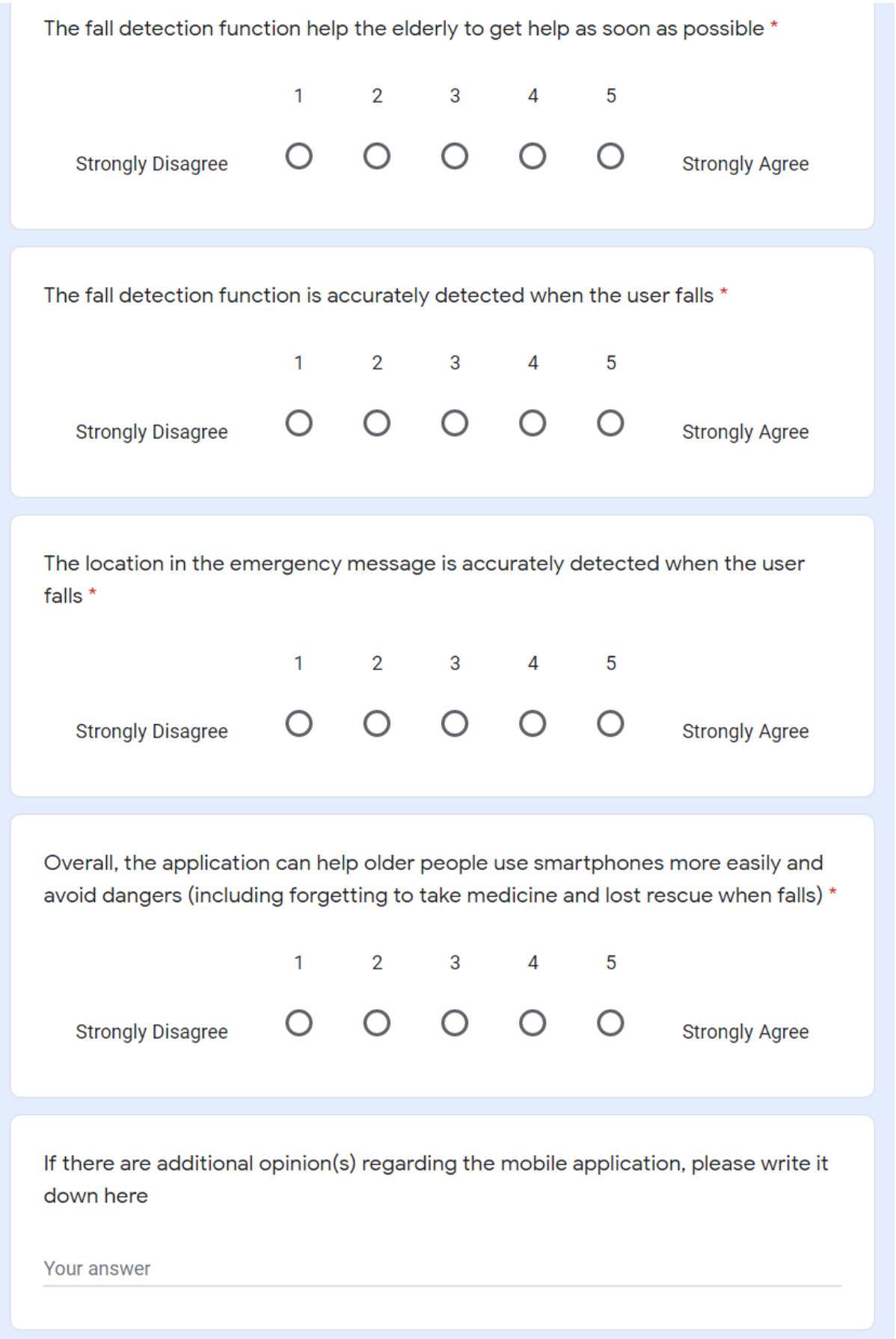


Figure 16 – User evaluation: Part 5

After the pre-post study via an online survey, the effect of the solution on users will be investigated. And we can get the response to measure users before and after using the prototype.

4. Chapter 4 – Results: Prototype Implementation

4.1. Issues about Implementation

For our application development, the Android application is developed by creating the project on Android Studio IDE. The emulator and the Logcat function are tools that aid in the development of Android applications to test and debug the application. The prototype application is developed using Android Studio version 3.6.3. Android SDK version Android 9 (API level 28) has been used for this application. The business is written in Java, and the user interface with string configuration is developed by XML files. The user interface is designed targeting Android mobile users rather than tablet users as this type of application is most widely used in motion and mobile devices are more comfortable for doing so. The navigation between all three main functions of the application makes it user-friendly and requires no system navigational instructions.

Permission Request (AndroidManifest.xml):

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />
<uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.PROCESS_OUTGOING_CALLS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.PROCESS_OUTGOING_CALLS" />
<uses-permission android:name="android.permission.READ_CALENDAR"/>
<uses-permission android:name="android.permission.WRITE_CALENDAR"/>
```

Libraries (build. gradle (Module: app)):

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    implementation 'androidx.exifinterface:exifinterface:1.3.0-alpha01'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
    implementation 'androidx.lifecycle:lifecycle-extensions:2.2.0'
    implementation 'androidx.navigation:navigation-fragment:2.2.2'
    implementation 'androidx.navigation:navigation-ui:2.2.2'
    implementation 'androidx.preference:preference:1.1.1'
    implementation 'androidx.vectordrawable:vectordrawable:1.1.0'
    implementation 'com.google.firebase:firebase-analytics:17.4.1'
    implementation 'com.google.firebase:firebase-auth:19.3.1'
    implementation 'com.google.firebase:firebase-core:17.4.1'
    implementation 'com.google.firebase:firebase-firestore:21.4.3'
    implementation 'com.google.android.gms:play-services-auth:18.0.0'
    implementation 'com.google.android.gms:play-services-location:17.0.0'
    implementation 'com.google.android.material:material:1.2.0-alpha06'
    implementation 'cat.ereza:customactivityoncrash:2.3.0'
    testImplementation 'junit:junit:4.13'
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'
    androidTestImplementation 'androidx.test:runner:1.2.0'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
}
```

4.2. Description of the Prototype

4.2.1. Setup of the prototype application

For the need for testing with the project source, it must be a little bit set up for passing the authentication of the firebase login function. And the following steps are showing how to complete the authentication before testing the source code:

Step 1.1: Go to the Firebase console that we are pre-setup the account for the firebase management to you. (For the peoples from GitHub, you should setup your own firebase project by yourself)

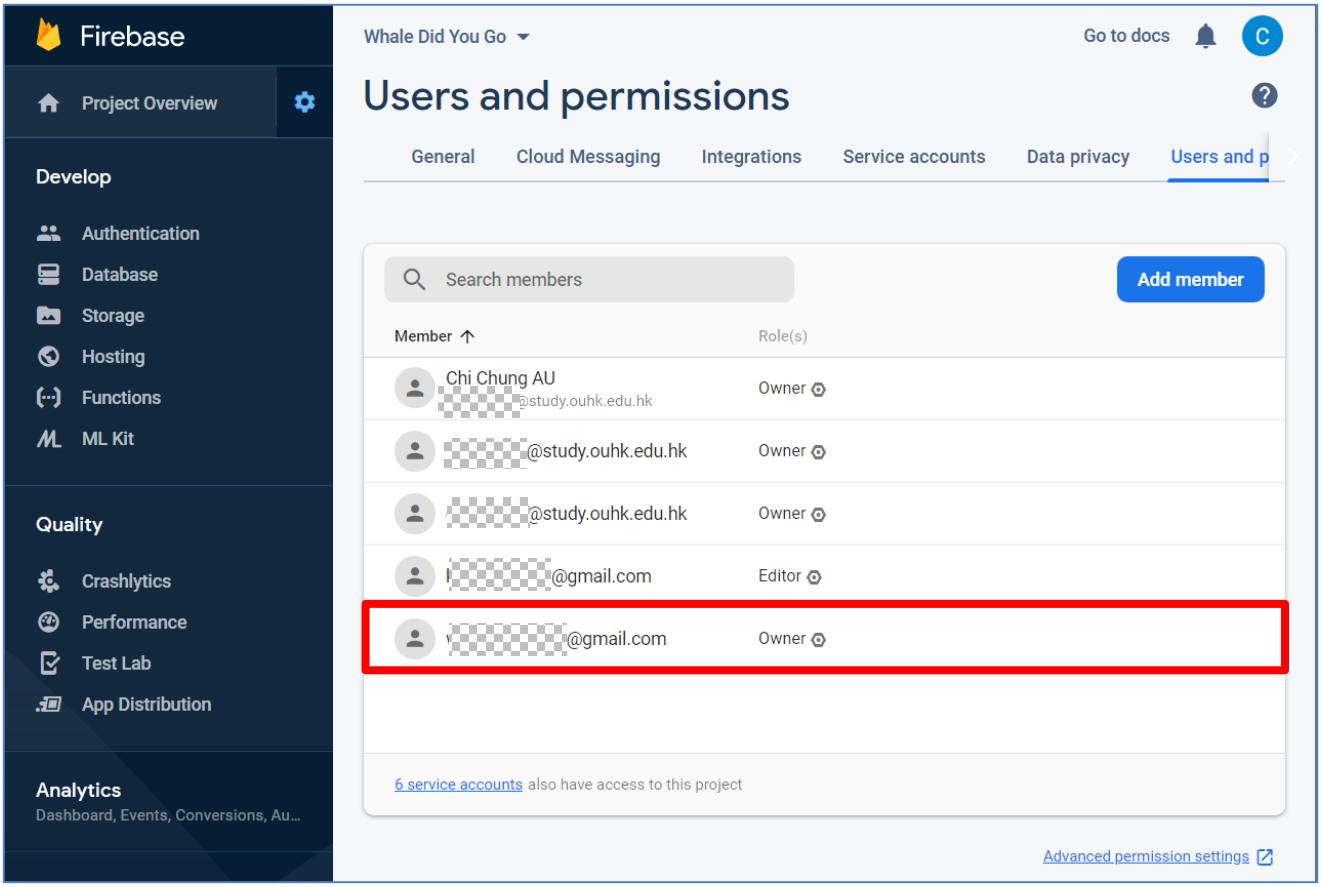


Figure 17 – Setup of the prototype application: Step 1.1

Step 1.2: Then, click into the firebase project (Whale Did You Go)

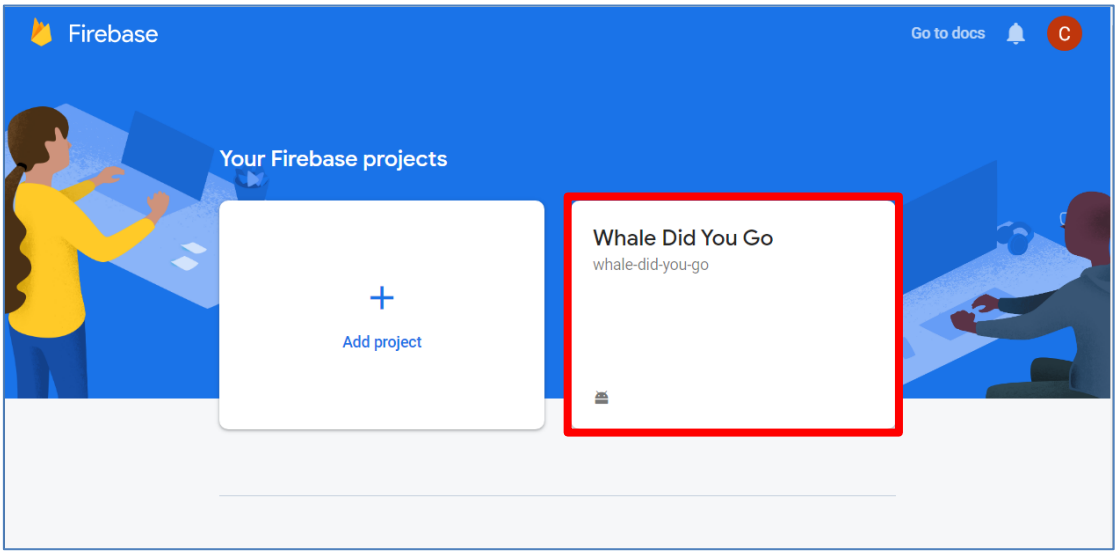


Figure 18 – Setup of the prototype application: Step 1.2

Step 2.1: Click the gearwheel button and select “Project settings”

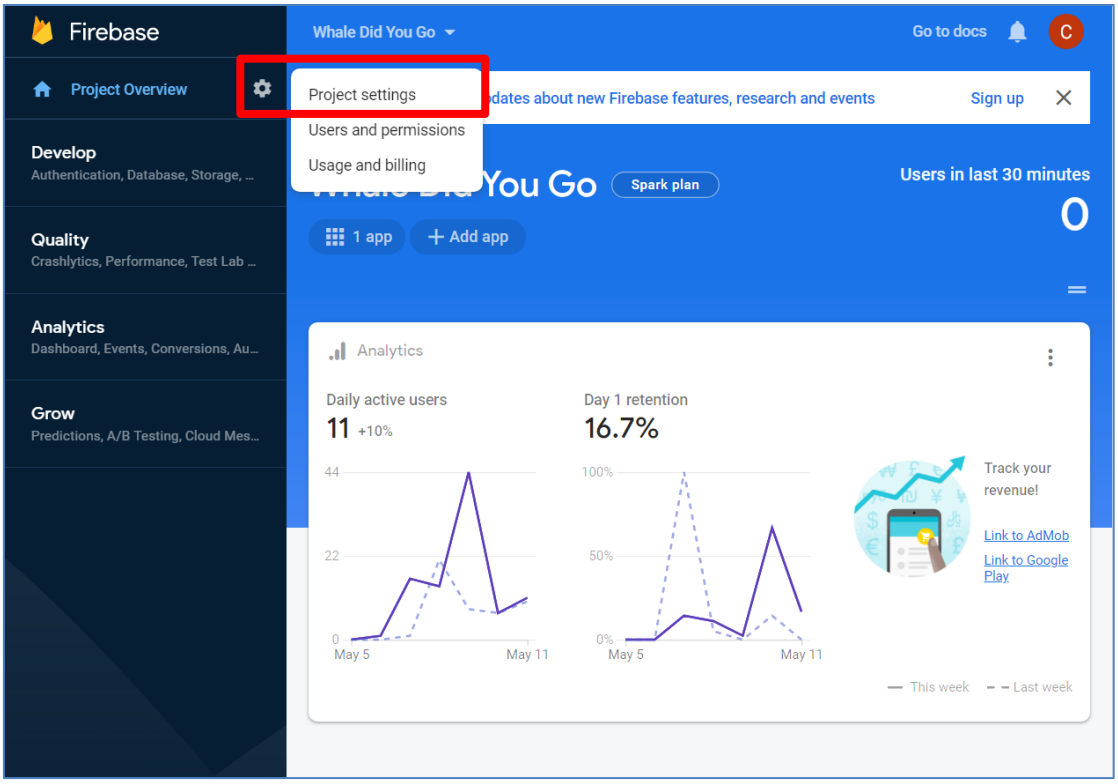


Figure 19 – Setup of prototype application: Step 2.1

Step 2.2: Then, you will see the firebase project setting page look like this

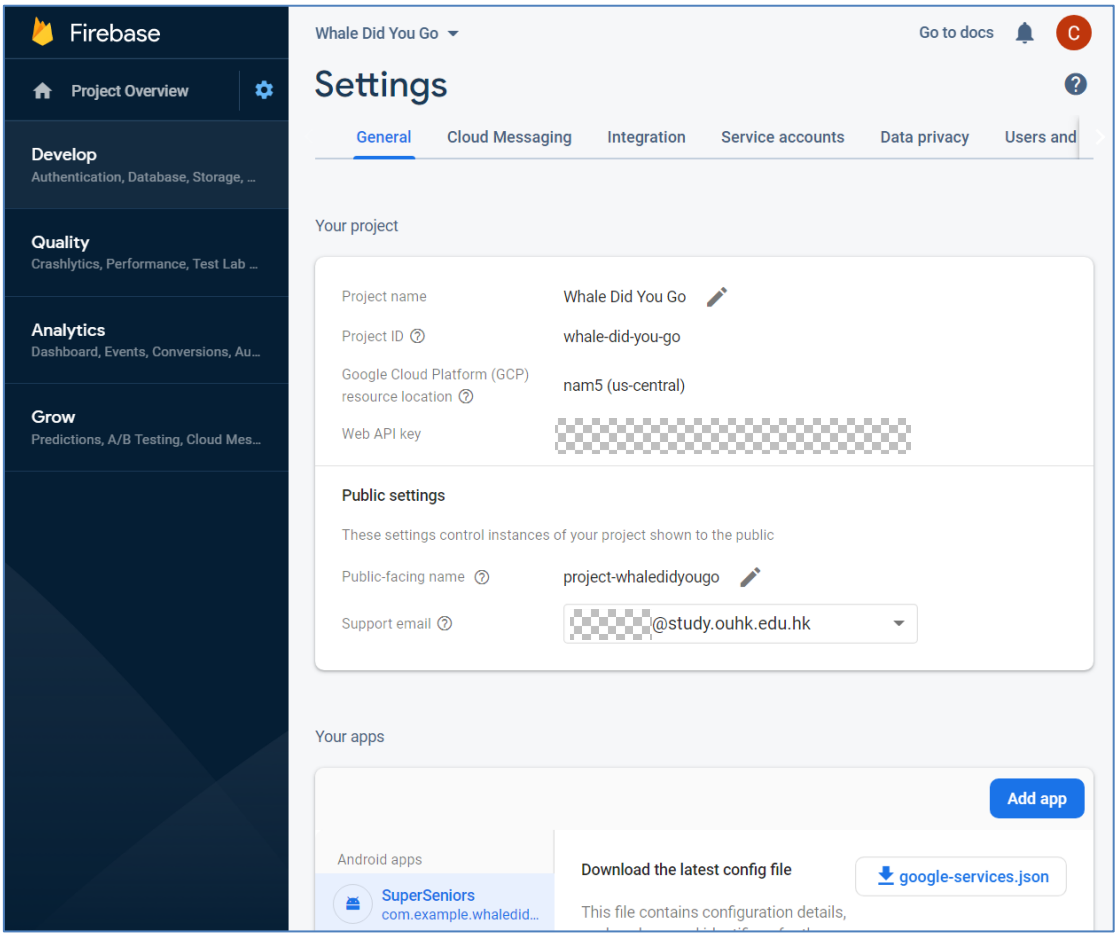


Figure 20 – Setup of prototype application: Step 2.2

Step 3.1: Scroll down the page to the title “Your apps” then you need to add the SHA certificate fingerprint which created by Android Studio

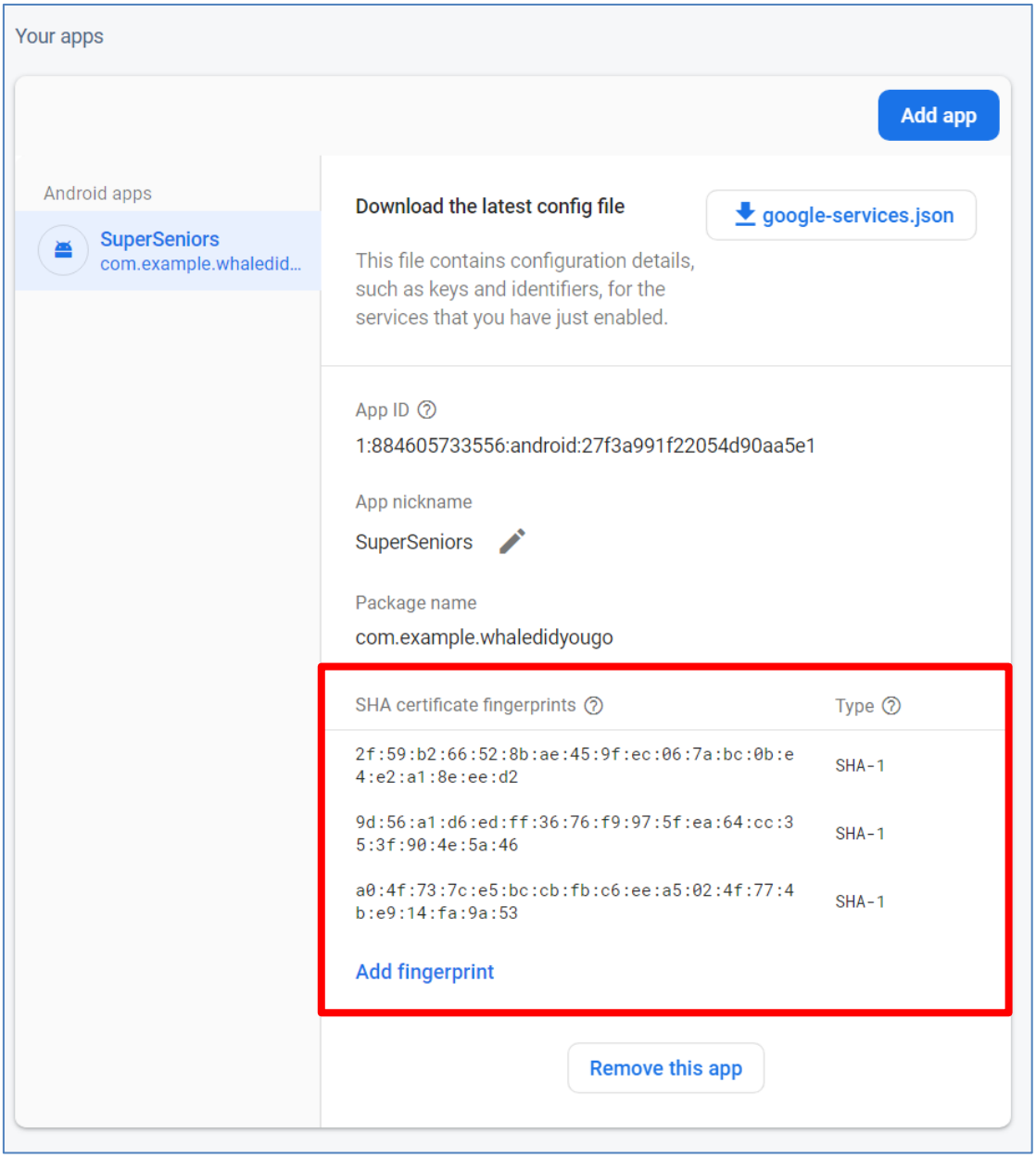


Figure 21 – Setup of prototype application: Step 3.1

Step 3.2: For the creation of the SHA certificate fingerprint, you need to open the Android Studio with the project source code file first, then click on grade (From the right-side panel, you will see Gradle bar)

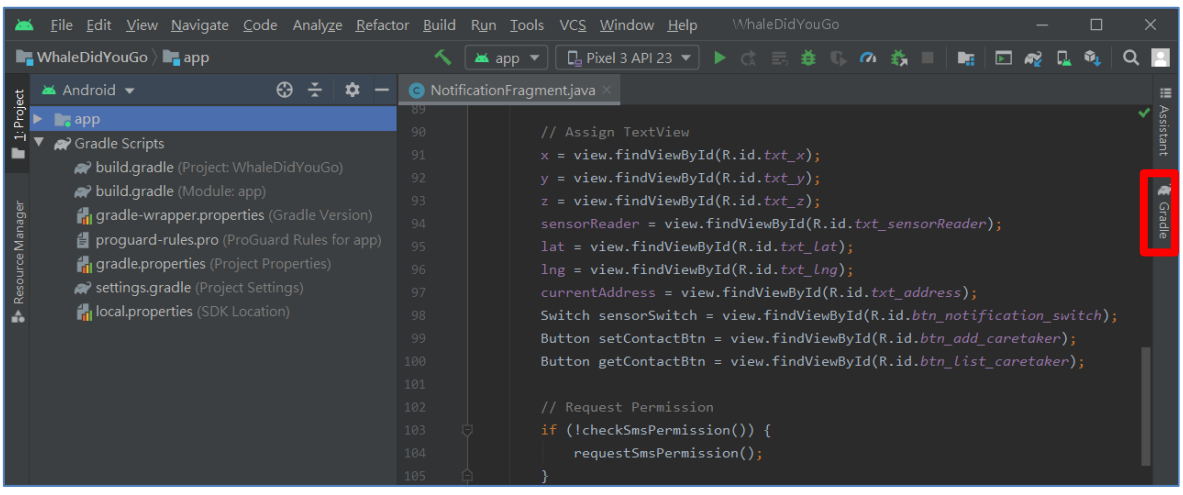


Figure 22 – Setup of prototype application: Step 3.2

Step 3.3: Click on “tasks”, then click on “Android” and double click on "signing report" (you will get SHA1 and MD5 in the run bar (Sometimes it will be in Gradle Console))

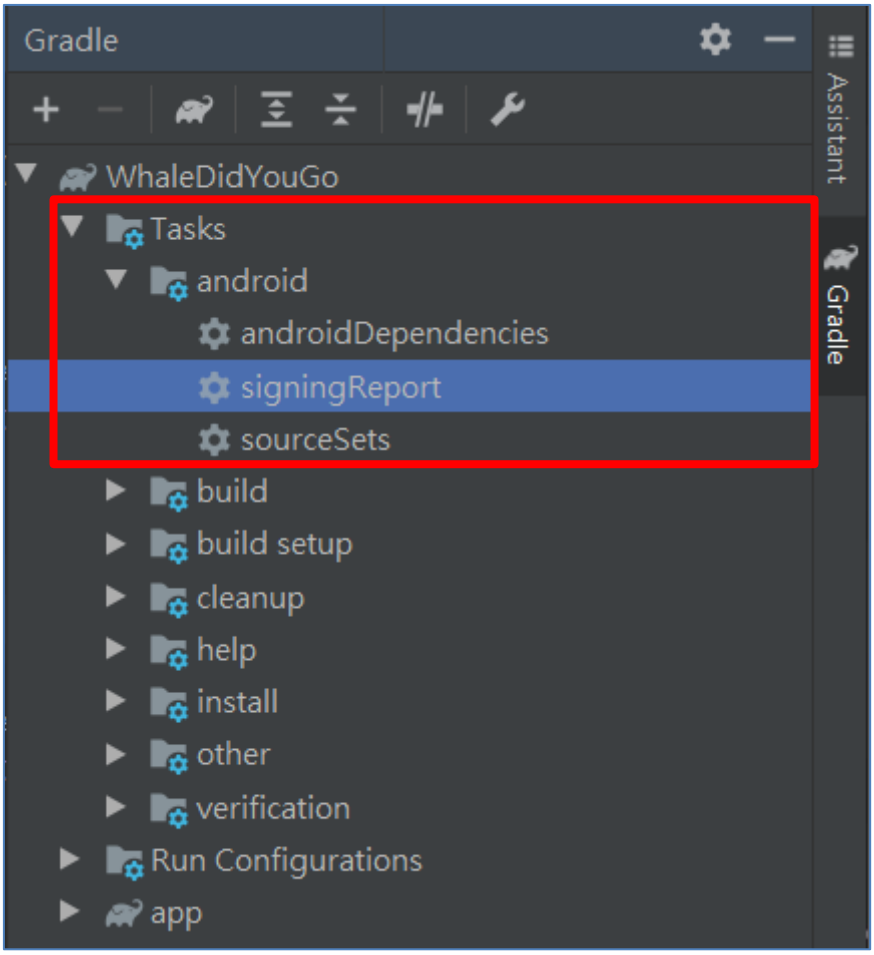


Figure 23 – Setup of prototype application: Step 3.3

Step 3.4: After that, the terminal in Android Studio will generate the SHA information look like this

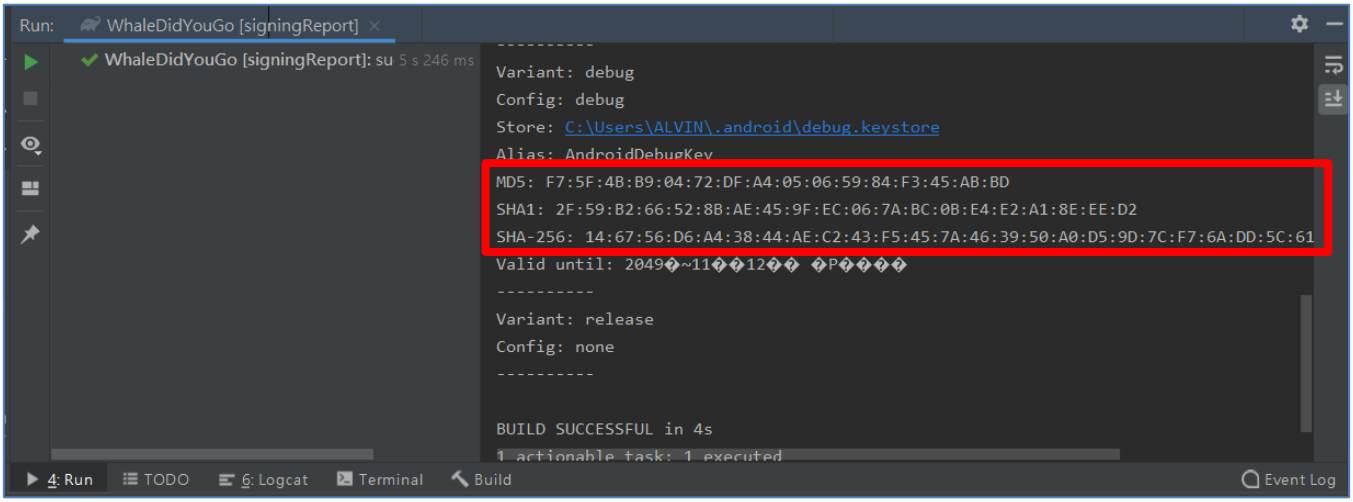


Figure 24 – Setup of prototype application: Step 3.4

Step 4.1: Copy the SHA1 key to the firebase console by using “Add fingerprint”

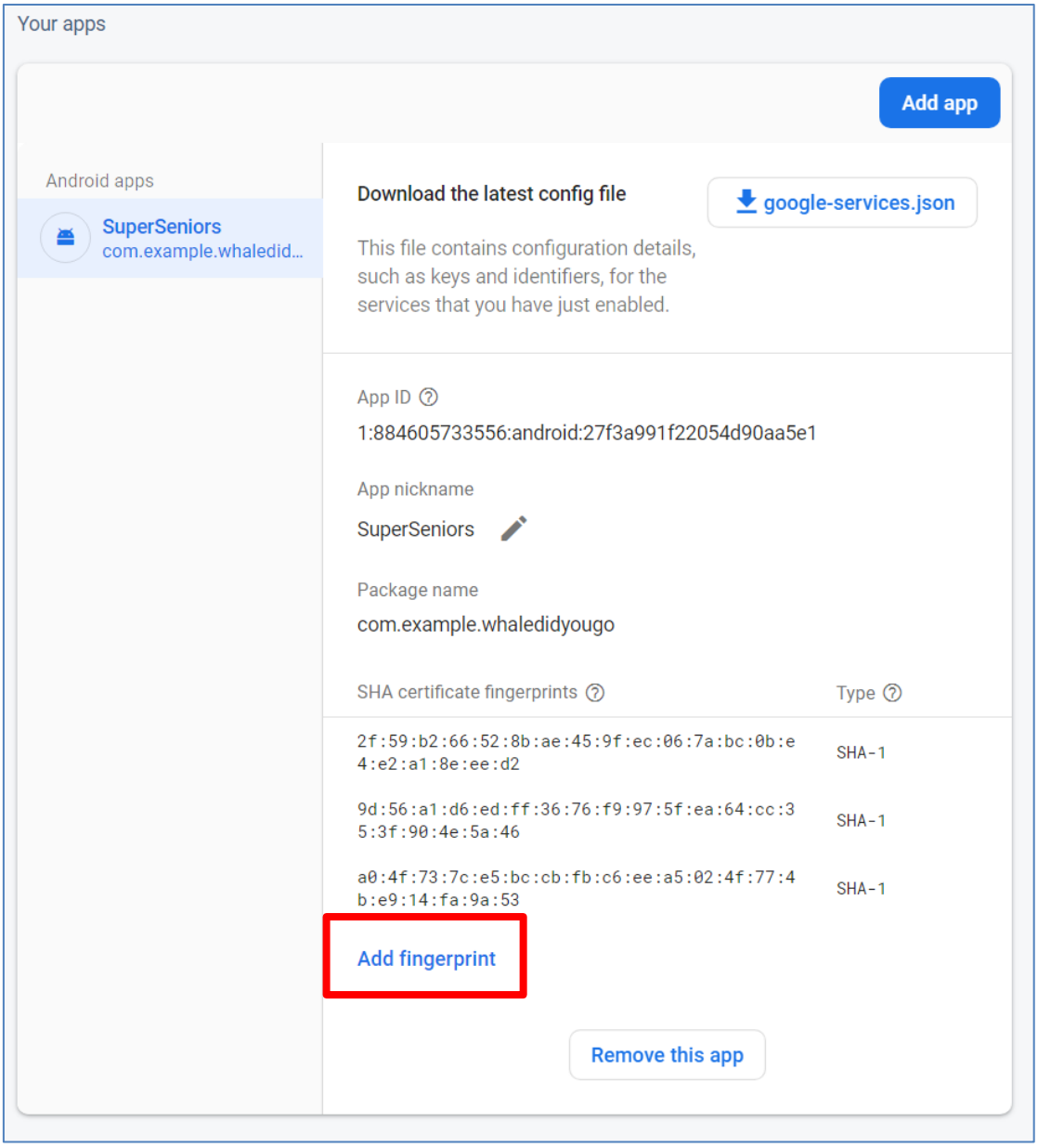


Figure 25 – Setup of prototype application: Step 4.1

Step 4.2: Remember to change back the compile item to “app”

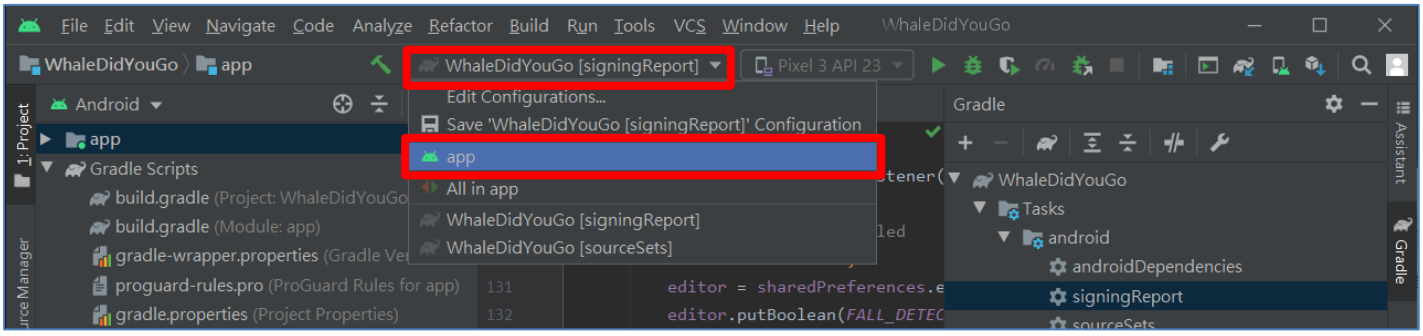


Figure 26 – Setup of prototype application: Step 4.2

Step 5: After installed the application, please turn on two of the Accessibility Service in “Settings” page

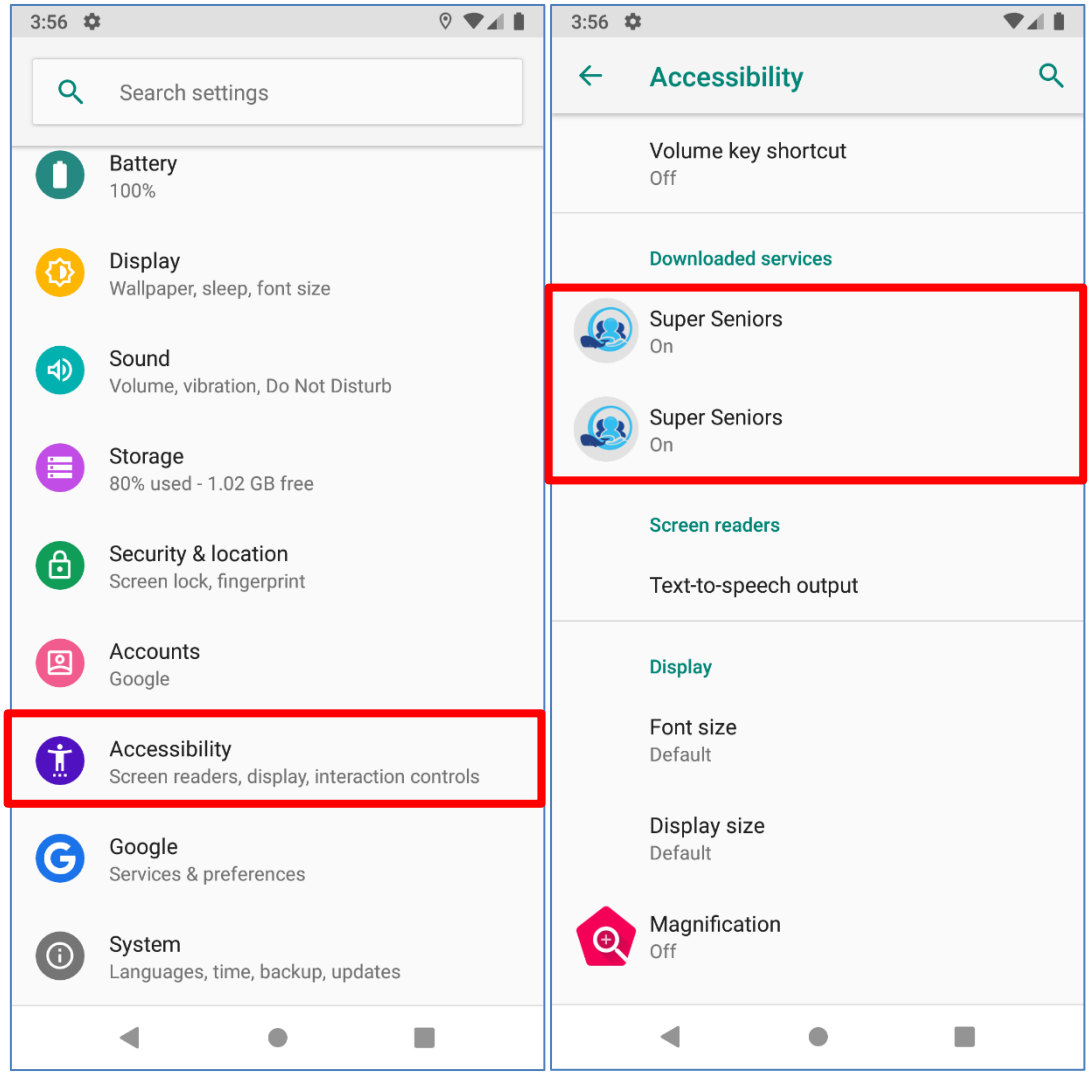


Figure 27 – Setup of prototype application: Step 5

Then, you can pass through the authentication of the google login function for testing the application.

Warning: Before the application test, please make sure to allow all permission for the application and turn on two of the accessibility setting. After the application test, please turn off the accessibility setting because it will affect your mobile device.

4.2.2. User Interfaces

4.2.2.1. Login Page

For the installation or application checked the unapproved permission, it will request all the needs in the login page. After the splash animation screen, users should log in with their Google account before using the application. And the user authentication from Firebase and Google Service API can provide the personal details from the user including username, email address, and Google calendar. Those of the information is used for the function of the daily schedule and the emergency message in the fall-detection function.

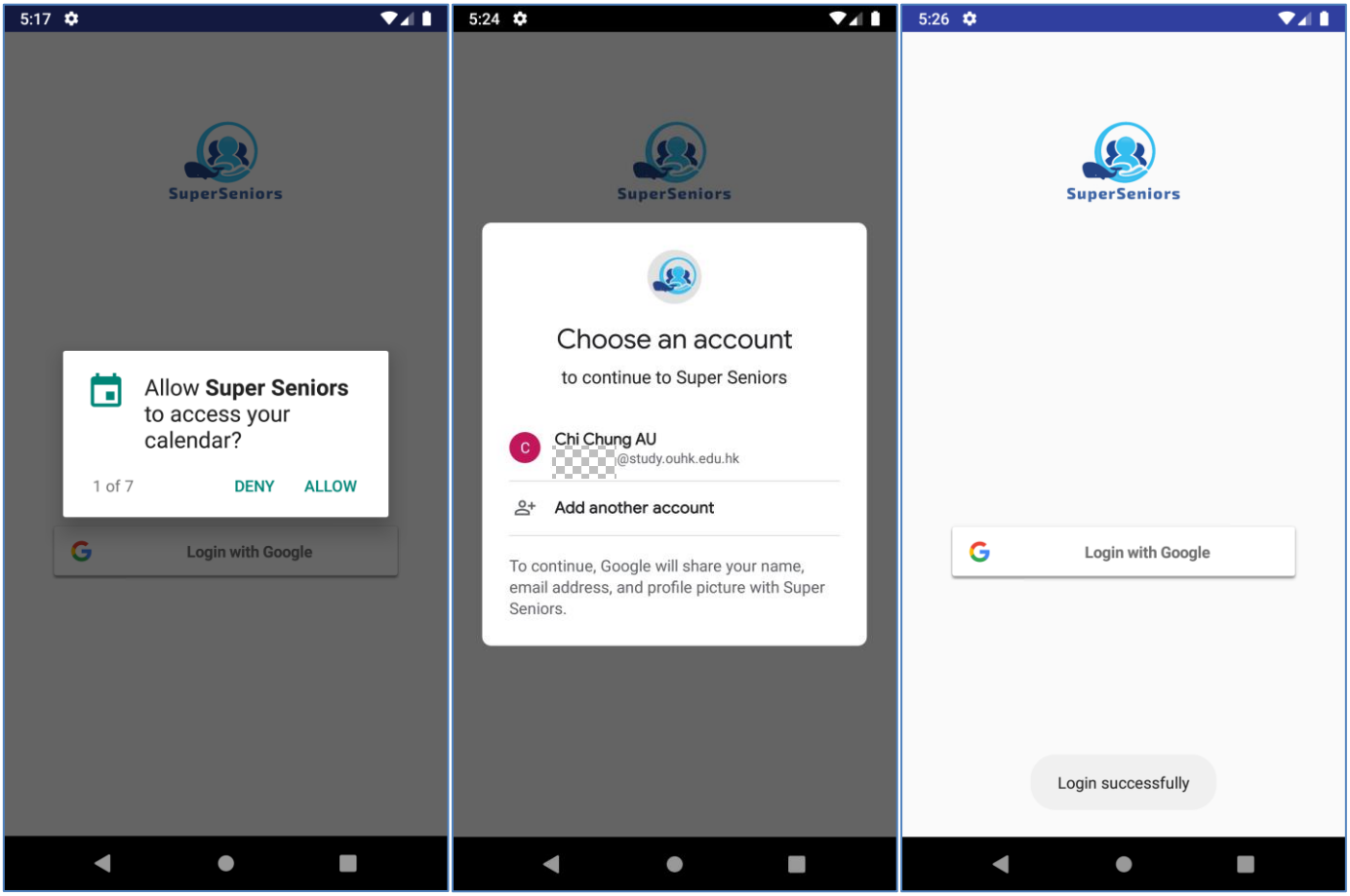


Figure 28 – User interfaces: Login page

4.2.2.2. Main Page

After the user login, divided into three areas (Schedule, Dashboard, Notification). And those areas performing the function of the daily schedule, single touch dialing & instant message, and fall detection. And the application provided some of the option buttons including logout, multi-language, and about page.

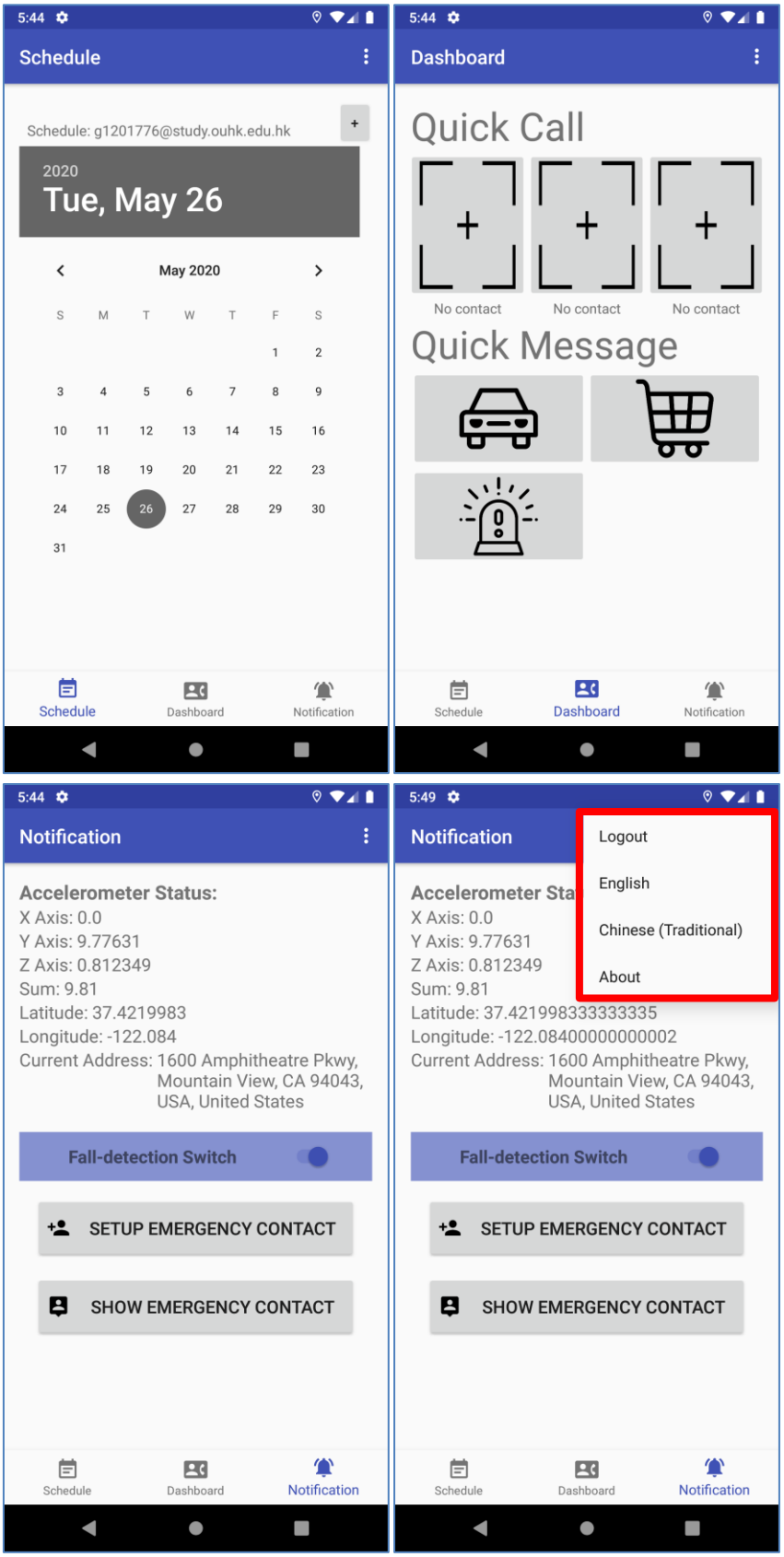


Figure 29 – User interfaces: Main page

4.2.2.3. Daily Schedule

The daily schedule function connecting with Google calendar based on user authentication in the previous login page. users can “browse”, “create”, “update”, and “delete” with the daily events, and it also includes the to-do list message which send by SMS for the reminder to the elderly whether the event times up. And the function solved the project objective of “Elderly daily life management”.

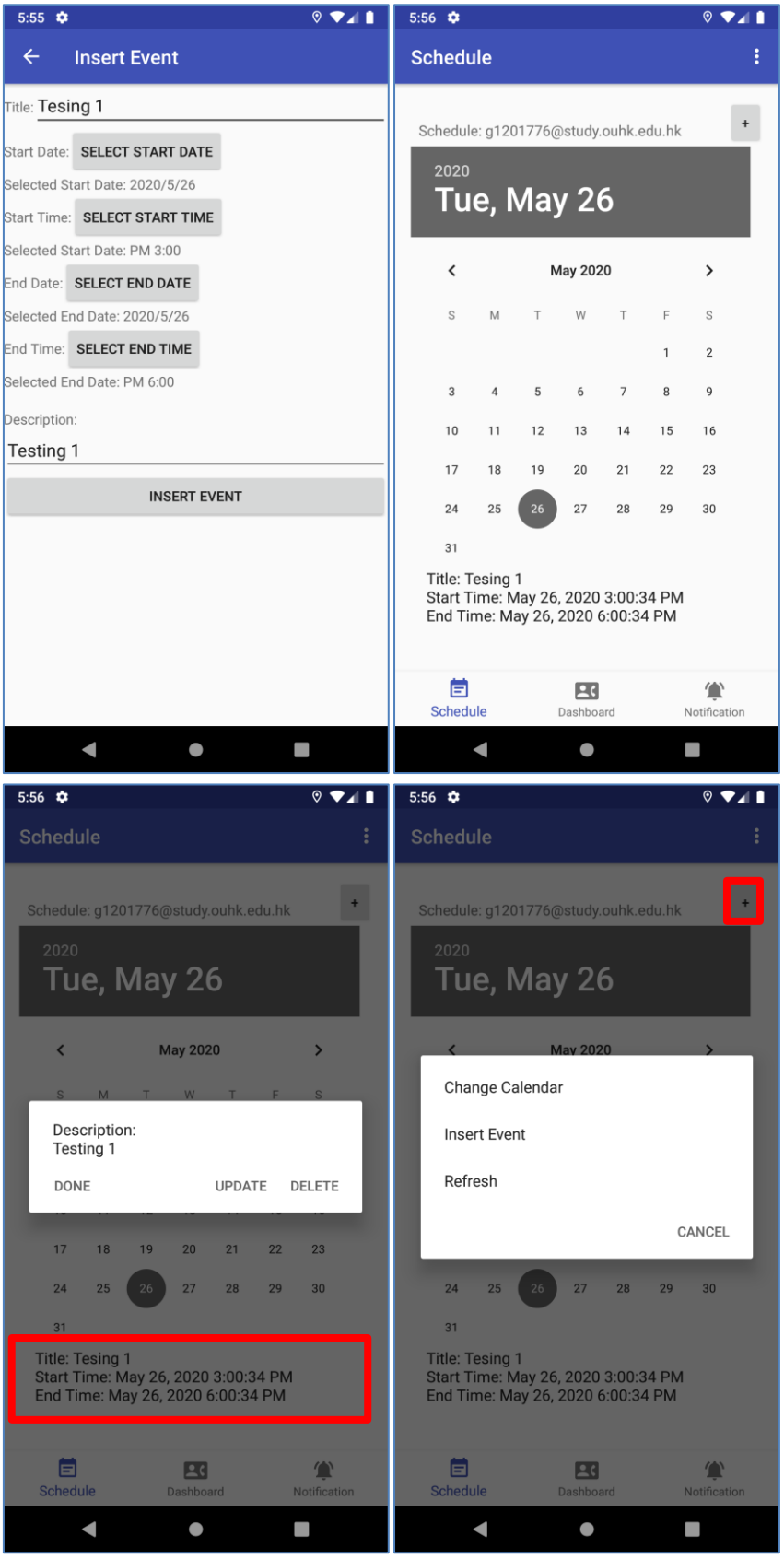


Figure 30 – User interfaces: Daily schedule

4.2.2.4. Single Touch Dialing

In the single touch dialing function, the user can insert most of the three contacts in the contact list. And user needs to enter some information for another party including photo (by image or camera), name, and phone number. And the function solved the project objective of “Elderly daily life management”.

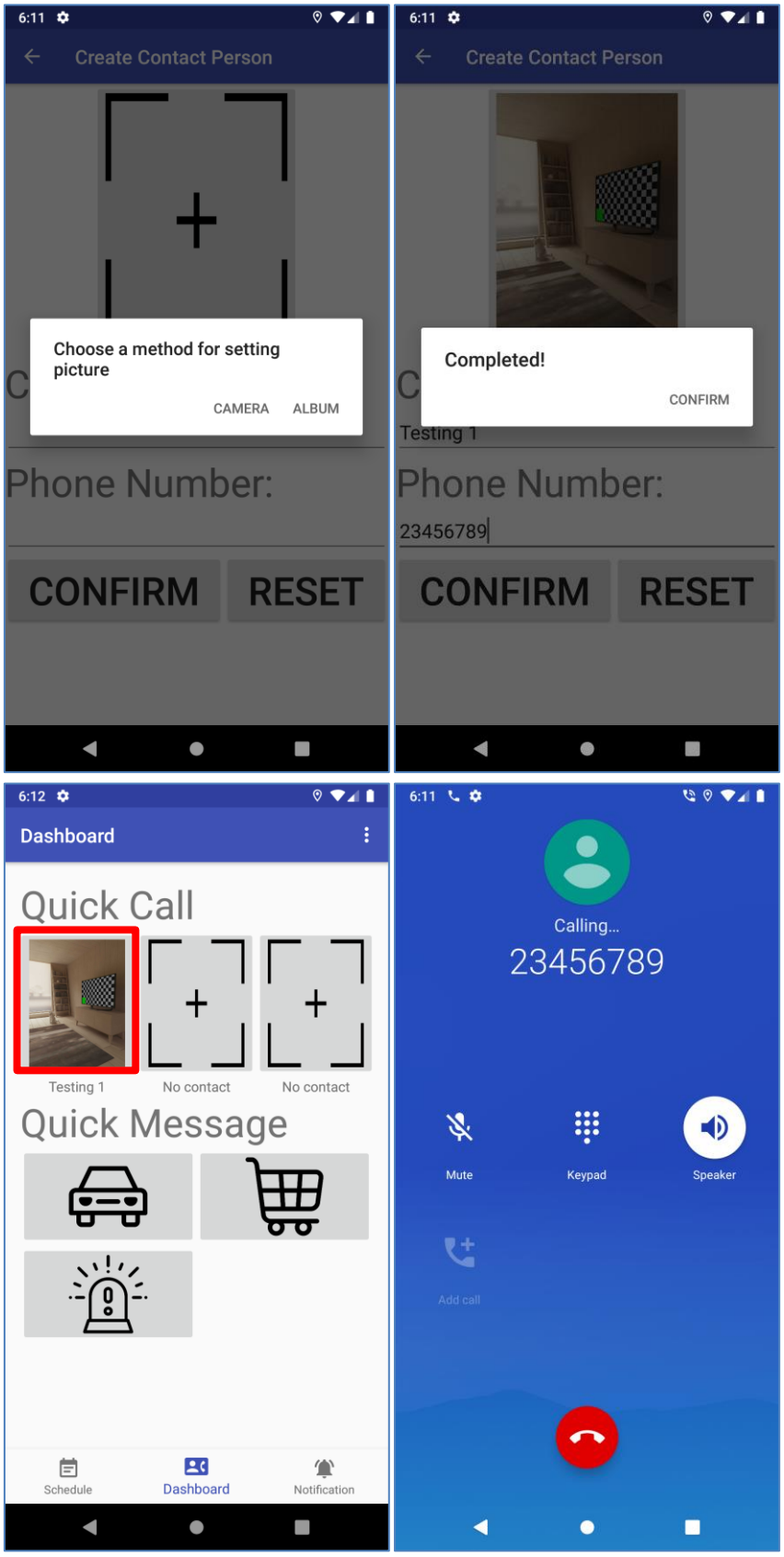


Figure 31 – User interfaces: Single touch dialing

4.2.2.5. Instant Message

In the instant message function, the user can send an SMS message by selecting some simple short sentence to another person in the contact list. It divided three types of messages including “Outing”, “Shopping”, and “Emergency”. And the function solved the project objective of “Elderly daily life management” and “Elderly safety”.

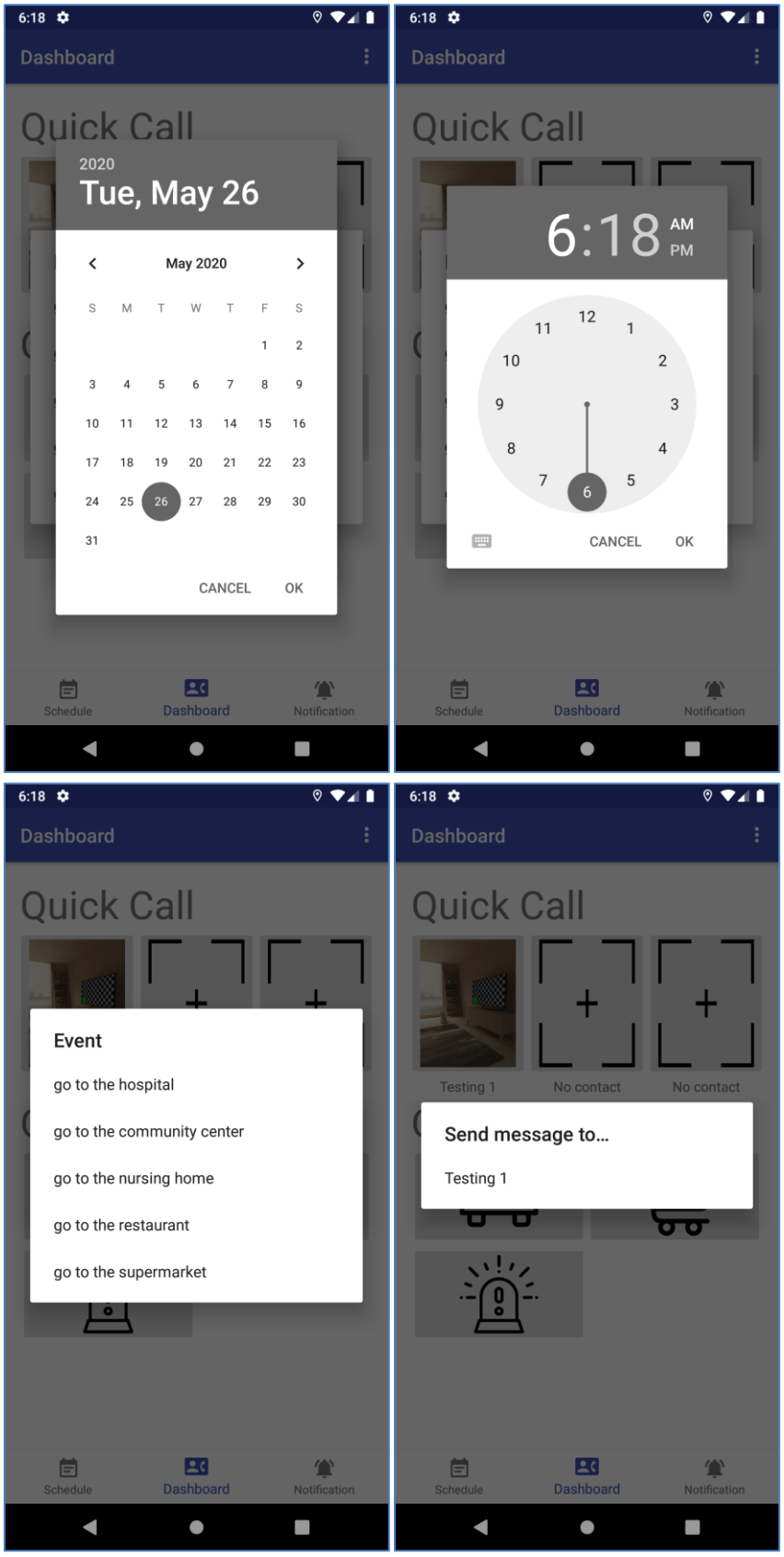


Figure 32 – User interfaces: Instant message

4.2.2.6. Fall Detection

In the fall detection function, the user can select to turn on or turn off the function by the switch button. After created the emergency contact and facing the fall accident, the application will pop up the countdown timer with ringtone alert then send the emergency message to the emergency contact in the emergency contact list. And the function solved the project objective of “Elderly safety”.

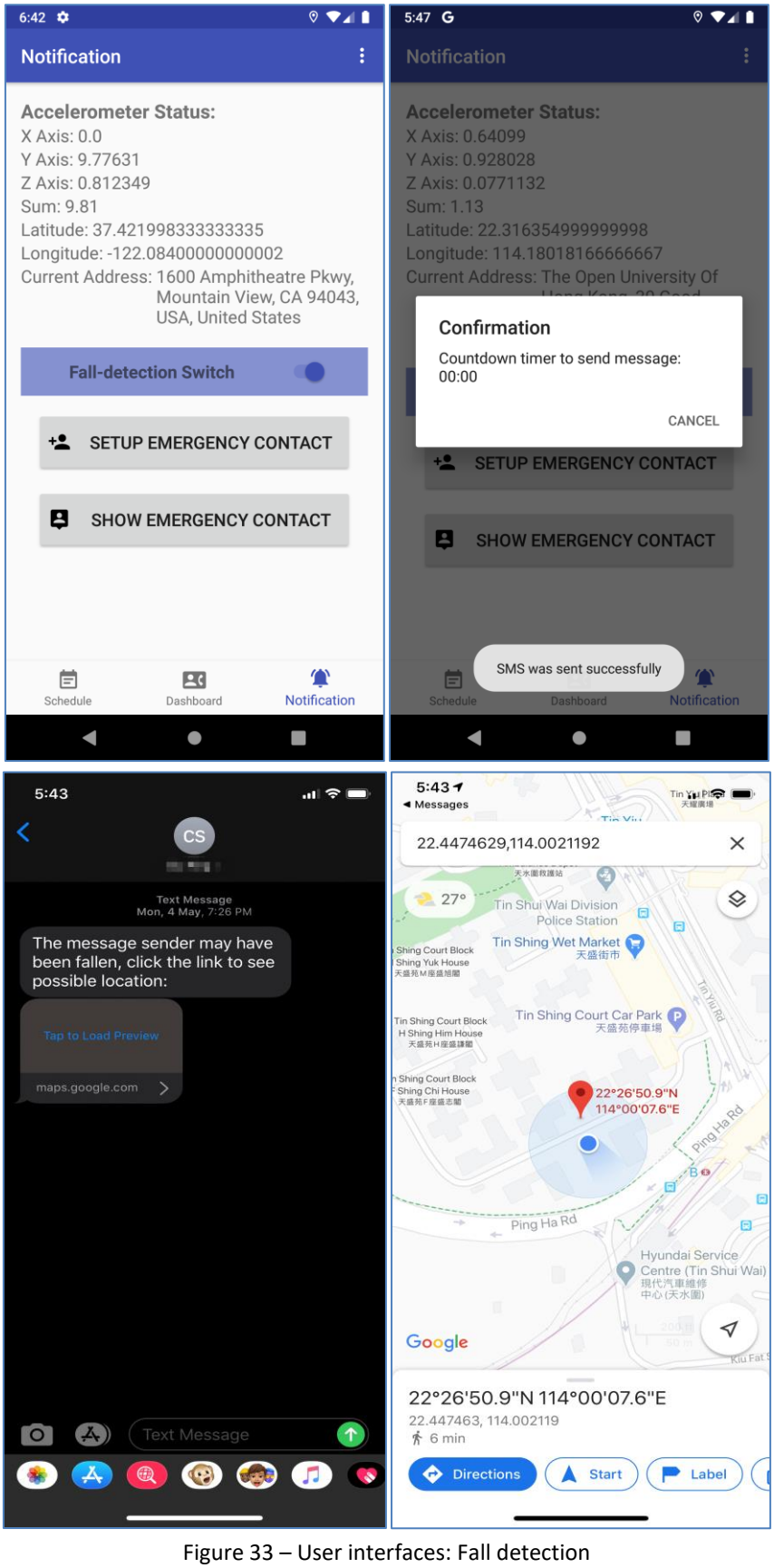


Figure 33 – User interfaces: Fall detection

4.2.2.7. Other Functions

In addition to the above three main functions, we added some of the interfaces such as multi-language setting with English and Traditional Chinese, splash screen animation before the login page, crash screen when facing any unknown error then provide a button for restart application, and about us page to introduce our aims and usage for the application. That we hope to make the application more complete.

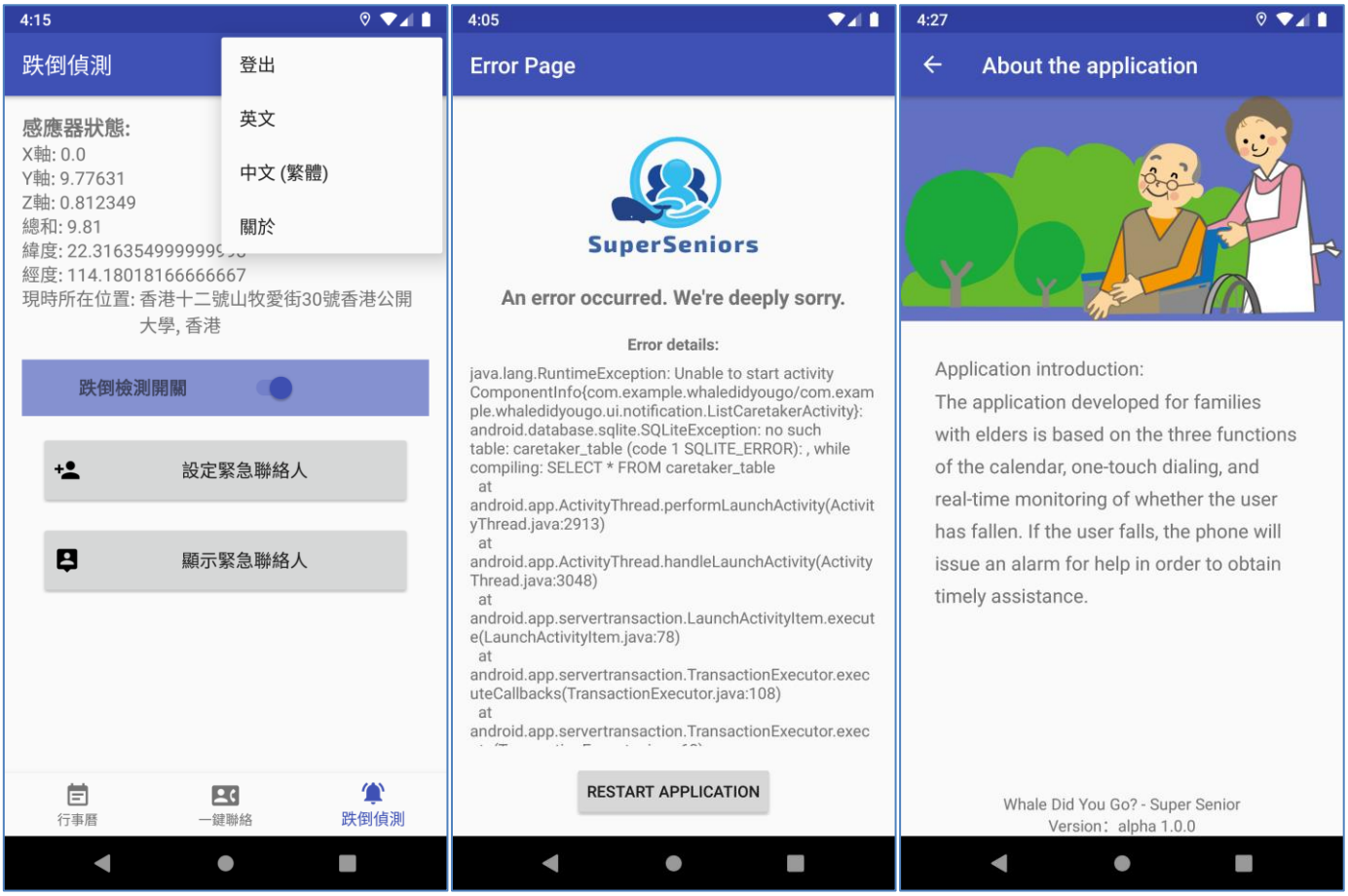


Figure 34 – User interfaces: Other functions

5. Chapter 5 – Results: Evaluation

After the survey of user evaluation via Google Form, the response summary by 8 users are following:

- Before testing of the application

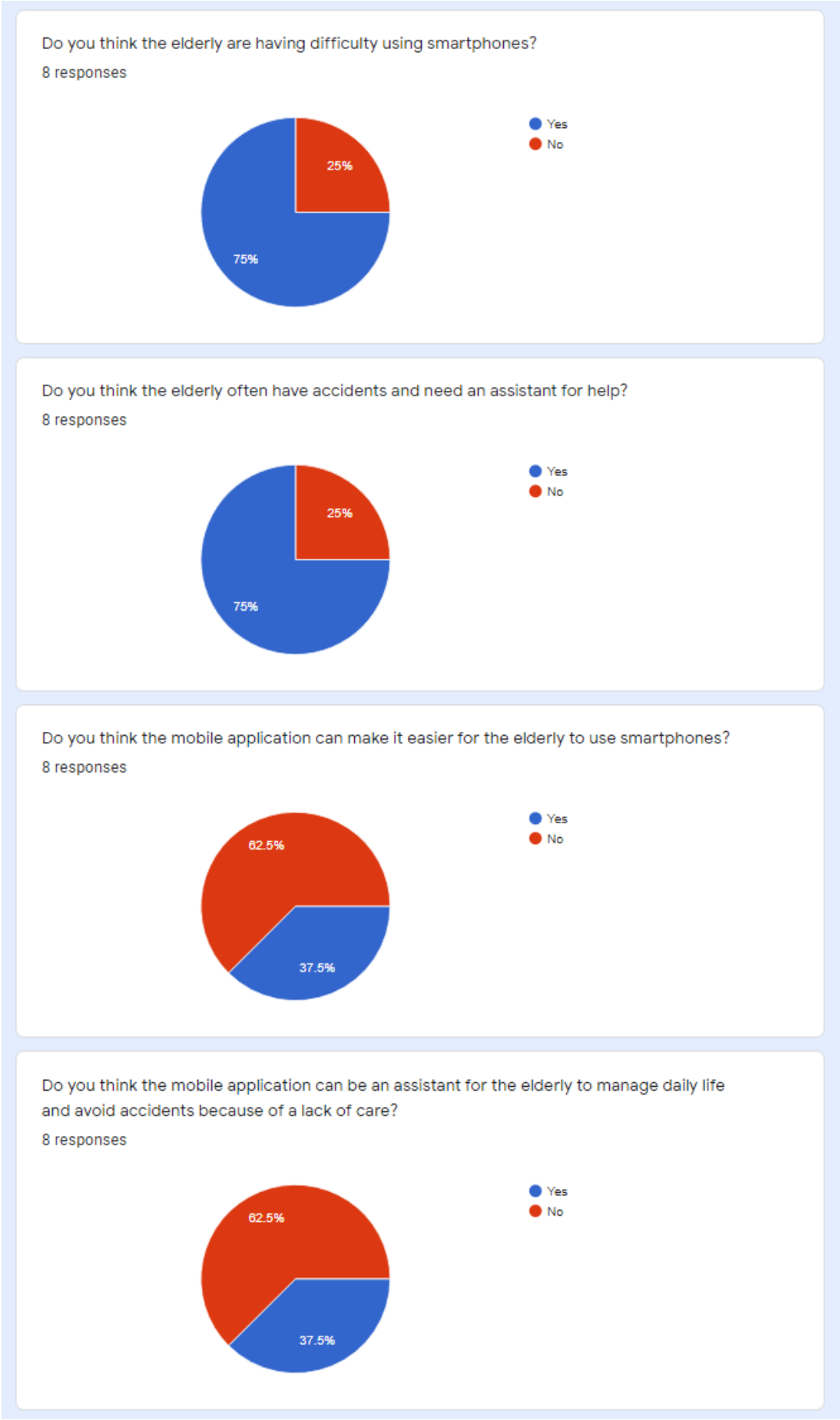


Figure 35 – Evaluation results: Part 1

- After testing of the application

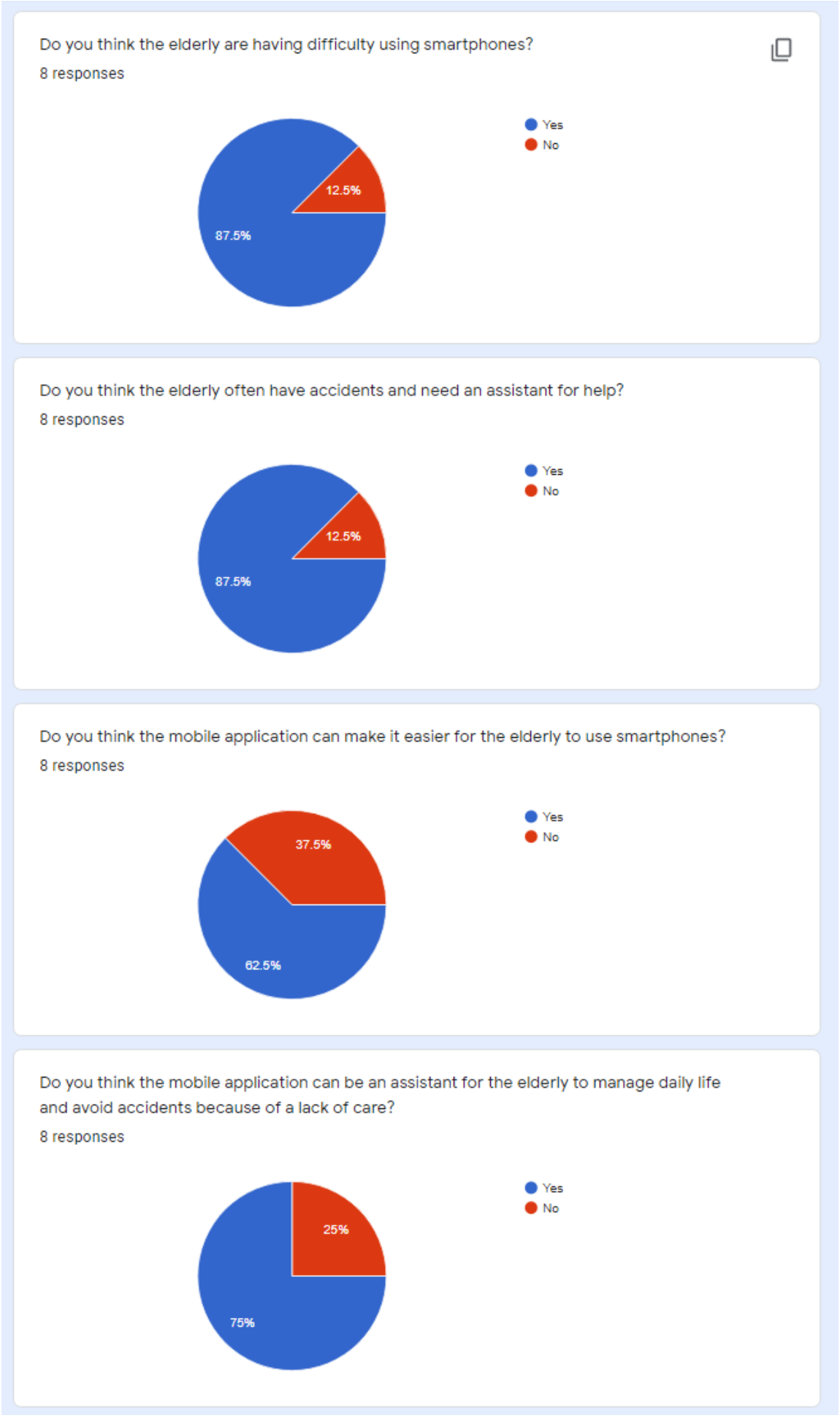
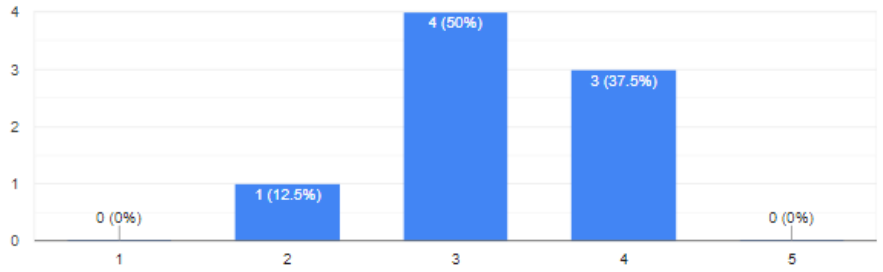


Figure 36 – Evaluation results: Part 2

- Feedback on the application

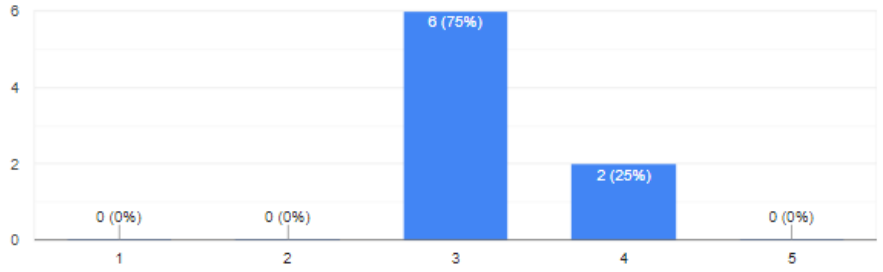
The user interface of the application is clean and easy to use

8 responses



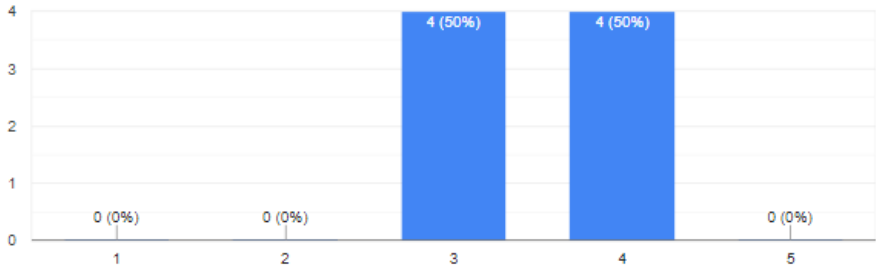
The daily reminder calendar function help the elderly to take their day management

8 responses



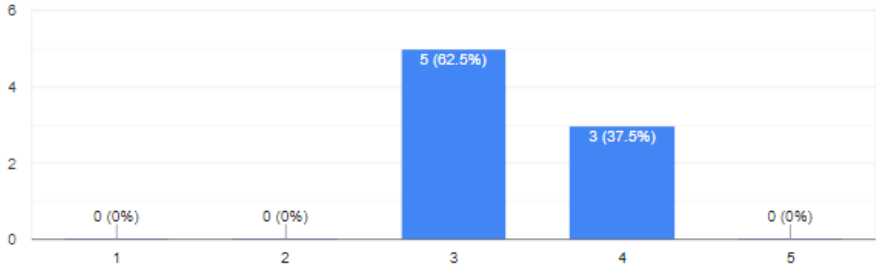
The single touch dialing function help the elderly to avoid the complicated user interface in mobile

8 responses



The instant message function help the elderly to avoid the complicated user interface in mobile

8 responses



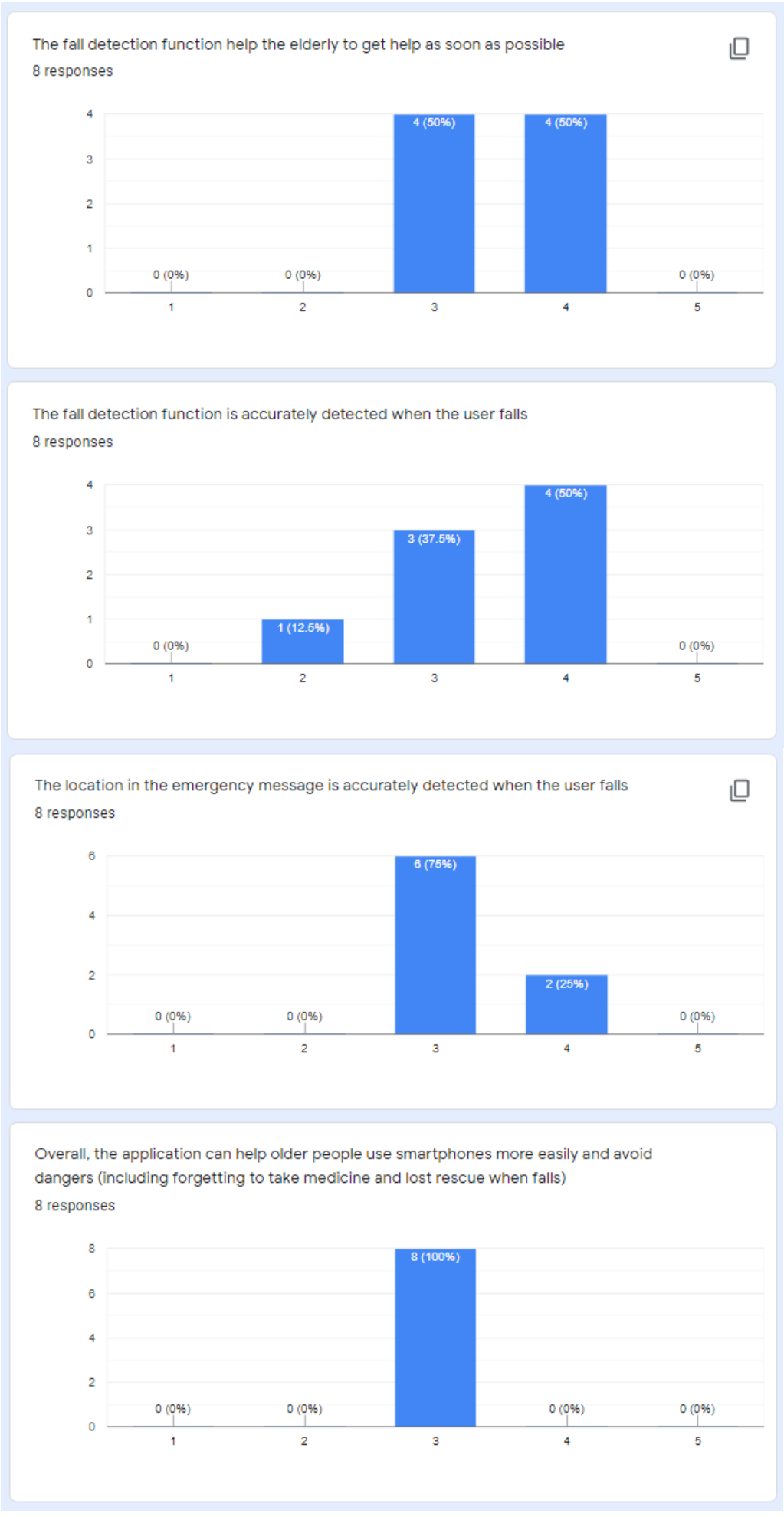


Figure 37 – Evaluation results: Part 3

5.1. Evaluation Study #1

Before:

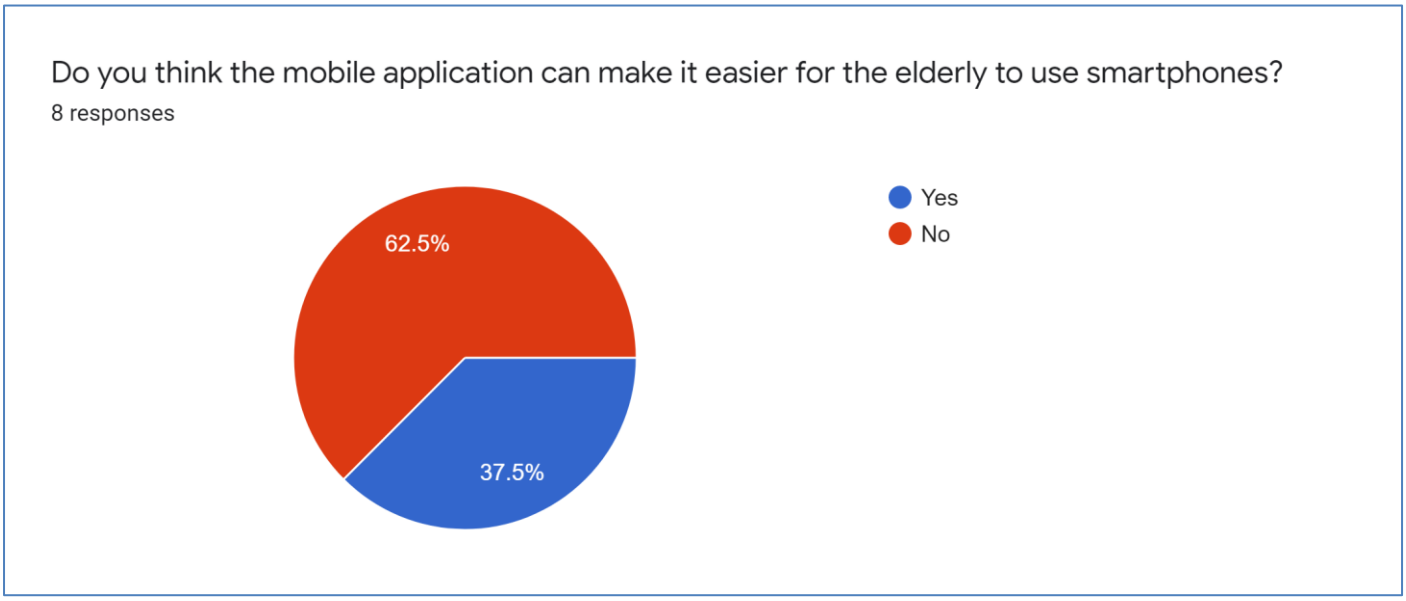


Figure 38 – Evaluation study 1: Before test

After:

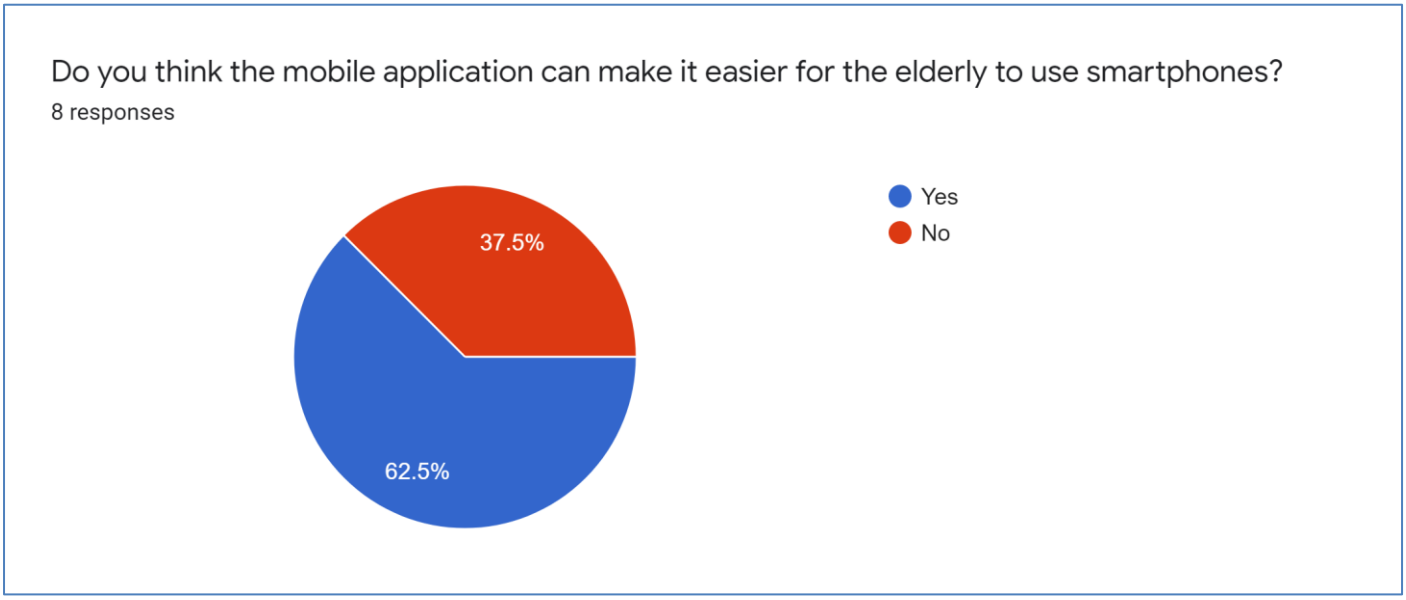


Figure 39 – Evaluation study 1: After test

To review the above responses, we can see that the view of "believe elderly are having difficulty using smartphones" and "believe the mobile application can make it easier for the elderly to use smartphones" are inverse proportion that means they think that mobile application is not useful. After the pre-post study method in user evaluation, we can see that some users are changed their minds and we can see an improvement in the above situation. That means, our application can meet the project objective of easy to use changed their view for the application.

5.2. Evaluation Study #2

Before:

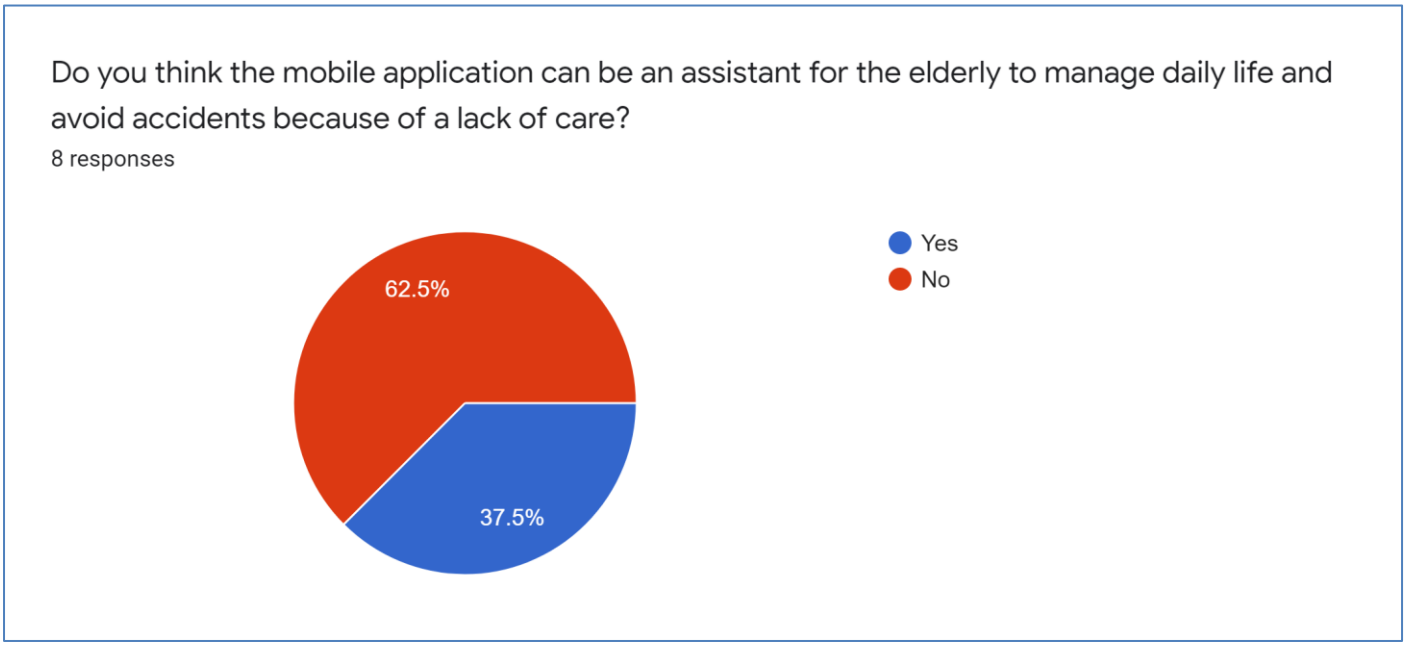


Figure 40 – Evaluation study 2: Before test

After:

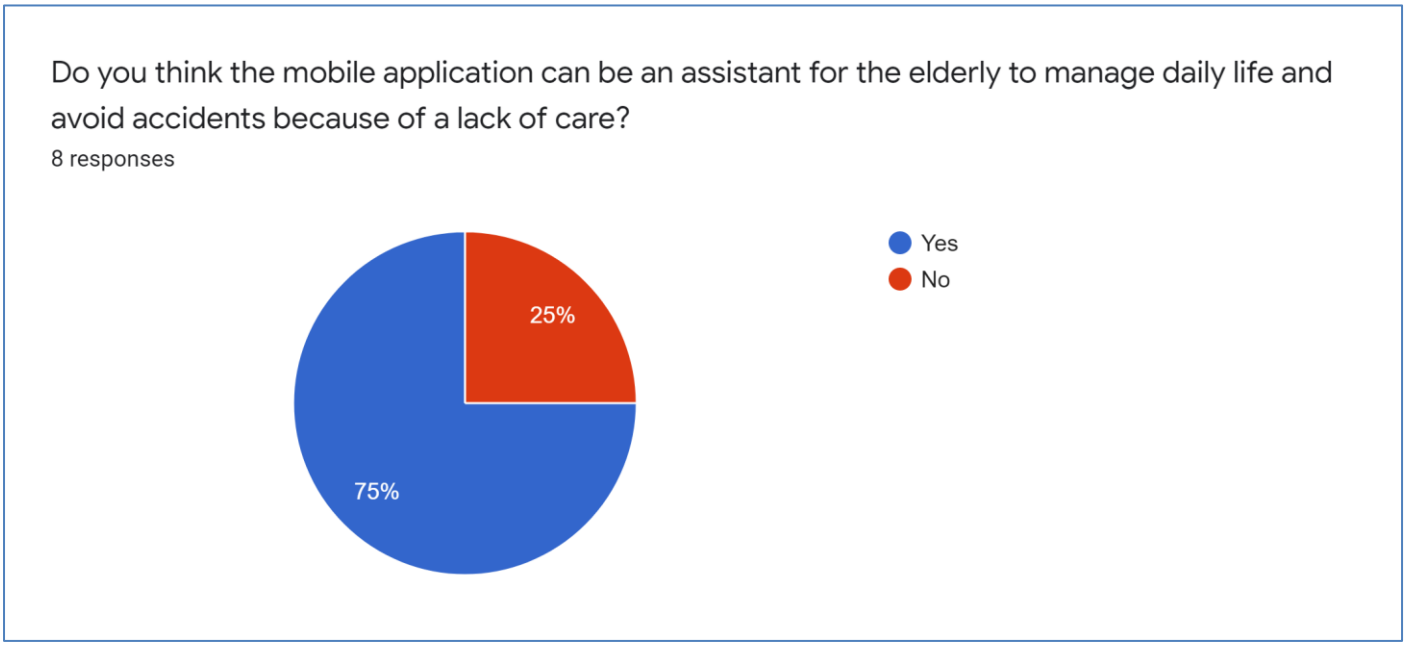


Figure 41 – Evaluation study 2: After test

The response is very similar to the above case in evaluation study 1. After the pre-post study method in user evaluation, we can see that some users are changed their minds and we can see a visible improvement in this case. That means, most of the users believing a mobile application can help the elderly to manage daily life and avoid accidents because of a lack of care.

6. Chapter 6 – Conclusion

For the whole of the application development, most of the aim given in the project background have been already satisfied and the application has been implemented on the real devices. The project enabled us to learn more about Android application development and it helped us in understanding the concept of the different modules including phone state listener and broadcast receiver for the single touch dialing function, the sensor event listener with accelerometers, and location listener for the fall detection function. Even in this project, we have also learned about how to measure the importance of different functions that prioritize the most important functions and take the system implementation as the main topic.

In fact, the system prototype can be improved in different ways and can be extended to support different devices such as tablets and iOS devices. After the survey of user evaluation, we get more ideas for the application. And the following are some of the possible extensions to extends the application. These services include:

1. The user interface can be more simply
2. Google map: provide the map interface for tracking the current location to users
3. In-app notification: some message can display via in-app notification and substitute to an SMS message, such as the notifications of the to-do list in daily calendar function, and the instant message functions
4. Voice control: single touch dialing, and instant message function can operate via voice control
5. Provide preference settings page that user can set up all the application settings in a single page

References

[1] D. Manghnani. (2018, May 22). Senior Safety App Now Tracks Senior Falls and Inactivity. [Online]. Available: http://www.reuters.com/brandfeatures/venture-capital/article?id=37090

[2] B. R. Prasad Babu., S. Patil., T. Gayathri. (2014, May). Smart Mobile Application for the Safety of Women and Elderly Population. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1022.7881&rep=rep1&type=pdf

[3] F. J. García-Peñalvo., M. Conde-González., V. Matellán. (2014, June). Mobile apps for older users. The development of a mobile app repository for older people. [Online]. Available: http://www.researchgate.net/publication/286178979_Mobile_Apps_for_Older_Users_-_The_Development_of_a_Mobile_Apps_Repository_for_Older_People

[4] B. Sreenivasan. (2012). Fall Detection with Posture recognition on Android Smart phone. [Online]. Available: https://github.com/BharadwajS/Fall-detection-in-Android/blob/master/Fall%20detection.pdf

[5] T. Rudraraju. (2011). Elderly support - android application for fall detection and tracking. [Online]. Available: https://krex.k-state.edu/dspace/handle/2097/17866

[6] Senior Citizen Home Safety Association (2019, August 28). e-Care Link Mobile App. [Online]. Available: http://www.schsa.org.hk/en/services/safe_services/outdoor_safe_services/service_2/introduction/index.html

[7] Senior Safety App (2019). Senior Safety App. [Online]. Available: http://www.seniorsafetyapp.com/

[8] A. Dennis., B. H. Wixom., D. Tegarden. (2019). System Analysis & Design. [Online]. Available: http://www.saigontech.edu.vn/faculty/huynq/SAD/Systems_Analysis_Design_UML_5th%20ed.pdf

[9] Android Developers. (2013, Feb 13). SensorEvent - Android SDK. [Online]. Available: https://stuff.mit.edu/afs/sipb/project/android/docs/reference/android/hardware/SensorEvent.html

[10] M. Kangas., A. Konttila., P. Lindgren., I. Winblad., T. Jamsa. (2011). Gait & Posture 28. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S096663620800026X

[11] A. A. P. Wai., S. S. Devi., J. Biswas., S. K. Panda. (2011). Pervasive Intelligence System to Enable Safety and Assistance in Kitchen for Home-Alone Elderly. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-21535-3_41

[12] Centre for Health Protection (2010). Unintentional Injury Survey - “Accidental Injury Survey 2008”. [Online]. Available: http://www.chp.gov.hk/files/pdf/injury_survey_eng.pdf

[13] T. Claburn. (2009). Court asked to disallow warrantless GPS tracking. [Online]. Available: http://www.informationweek.com/architecture/court-asked-to-disallow-warrantless-gps-track-ing/d/d-id/1077257

[14] S. Jung., S. Qin. (2014, May 10). Development of safe-driving-system features for elderly drivers. [Online]. Available: http://www.tandfonline.com/doi/full/10.1080/21642583.2014.924083#aHR0cHM6Ly93d3cudGFuZGZvbmxpbmUuY29tL2RvaS9wZGYvMTAuMTA4MC8yMTY0MjU4My4yMDE0LjkyNDA4Mz9uZWVkQWNjZXNzPXRydWVAQEAw

Appendix A. Team Member's Roles and Responsibility

Table 6 – Appendix: Team member's roles

Roles	*Member(s)*	*Remarks*
Team Leader	Leung Kwan Ho	Push forward for the system development progress and coordinate of the supervisor meeting
Team Coordinator	Au Chi Chung	Manages the project records, reports, presentation slide, and other documents in order
Secretary	Au Chi Chung	Meeting agenda and minutes
System Analyst & Designer	All Members	Including System Architecture and User Interface Design
Programmer	All Members	/
Tester and Evaluator	Leung Kwan Ho, Lie Tze Wah	/
Development Expert	Lie Tze Wah	Focus on the application development and debug

Table 7 – Appendix: Team member's responsibility

Tasks	*Responsible Member(s)*
Project planning	All Members
Technology test of resource and library	All Members
First screening presentation slide	Au Chi Chung
First screening presentation	Leung Kwan Ho
Initial report	Au Chi Chung
Design system architecture	All Members
Design user interface	All Members
System login function	Au Chi Chung
Single touch dialing function	Lie Tze Wah
Fall-detection function	Au Chi Chung & Lie Tze Wah
Daily schedule function	Leung Kwan Ho & Lie Tze Wah
Instant message function	Lie Tze Wah
Second screening presentation slide	Au Chi Chung
Interim report	Au Chi Chung
Experiment setup and coordination	Leung Kwan Ho & Lie Tze Wah
Experimental data analysis	Leung Kwan Ho & Lie Tze Wah
Final presentation slide	Au Chi Chung
Final Presentation	All Members
Final Report	Au Chi Chung

The Open University of Hong Kong
School of Science and Technology
Bachelor of Computing with Honours in Internet Technology

COMPS456F – Software System Development Project

Final Year Project – Team Member Final Report

Elderly Mobile Life Assistant for Family Members with Extra Peace of Mind

Group Name: Whale Did You Go? Supervisor: Dr. Kelvin Kai Wing LEE

Student Name	Student ID Number	Signature
Leung Kwan Ho	12123629	

Appendix B1. Team Member Leung Kwan Ho's Final Report

Table 8 – Appendix: Team member Leung Kwan Ho's final report

Declaration Statement

I, Leung Kwan Ho (12123629), certify that the description and information included in this team member's report is true to the best of my knowledge.

Leung Kwan Ho: _____ (Signature and Date)

Task Assigned to the Author and their Status

The above table outlines the task assigned to me in this project:

Tasks	*Responsible Member(s)*	*Target Date*	*Rate*	*Remarks*
First Screening Presentation (Present)	Leung Kwan Ho Au Chi Chung	3rd Oct 2019	100%	/
Initial Report (Component diagram) (Data flow diagram)	Leung Kwan Ho Au Chi Chung Lie Tze Wah	24th Oct 2019	100%	/
Interim Demonstration (Recorded video of prototype)	Leung Kwan Ho Au Chi Chung Lie Tze Wah	25th Feb 2020	100%	/
Final Demonstration and Presentation (Recorded video of prototype) (Video 2: Project Outline, Objectives)	Leung Kwan Ho Au Chi Chung Lie Tze Wah	20th May 2020	100%	/
Daily schedule function (Google calendar)	Leung Kwan Ho	/	100%	Debug with Lie Tze Wah

Review and Appraisal of Code Section

1.

```
cur = cr.query(builder.build(),
        INSTANCE_PROJECTION,
        selection,
        eventSelectionArgs,
        CalendarContract.Events.DTSTART + " ASC");
if (cur != null) {
...
    while (cur.moveToNext()) {
        ...
        eventID = cur.getLong(PROJECTION_ID_INDEX);
        beginVal =
DateFormat.getDateTimeInstance().format(cur.getLong(PROJECTION_BEGIN_INDEX));
        title = cur.getString(PROJECTION_TITLE_INDEX);
        endVal = DateFormat.getDateTimeInstance().format(cur.getLong(PROJECTION_END_INDEX));
        description = cur.getString(PROJECTION_DESCRIPTION_INDEX);
        ...
    }
    cur.close();
}
```

This part of code is showing some simple steps of how to query events from a Google Calendar by using Google API and cursor. There are 5 elements we need to put into the query. First, builder is created with Content URL of Calendar. The start and end date will also be set to the builder. Second, INSTANCE_PROJECTION is a string array to keep the column name of those records. Third, Selection is the condition of the query. Fourth, eventSelectionArgs is the variable for the Selection. Fifth, that is the column that you want to order the query result by.

After query events, if the cursor is not null, use a while loop to get the data of each column of each record by using the PROJECTION_ INDEX. This index is the order of the string array INSTANCE_PROJECTION.

2.

```
long eventId = Long.valueOf(eventIDArray[listPosition]);
String doneTitle = eventTitleArray[listPosition] + "(done)";
ContentResolver cr = getActivity().getContentResolver();
ContentValues values = new ContentValues();
values.put(CalendarContract.Events.TITLE, doneTitle);
int permissionCheck = ContextCompat.checkSelfPermission(getContext(),
        Manifest.permission.WRITE_CALENDAR);
if (permissionCheck == PackageManager.PERMISSION_GRANTED) {
    Uri uri = ContentUris.withAppendedId(CalendarContract.Events.CONTENT_URI, eventId);
    cr.update(uri, values, null, null);
}

String starttimeformsn =
DateFormat.getDateTimeInstance().format(eventBeginArray[listPosition]);
String endtimeformsn = DateFormat.getDateTimeInstance().format(eventEndArray[listPosition]);
for (int i = 0; i < contact_phone.length; i++) {
    String message = getString(R.string.done_sms) + ":\n" + eventTitleArray[listPosition];
    String smsTo = contact_phone[i];
    SmsManager smsManager = SmsManager.getDefault();
    smsManager.sendTextMessage(smsTo, null, message, null, null);
}
```

And this part is the actions that will be taken after elderly had finished an event and clicked the done button. It will get that event's title from the title array and set that event's title with done. Then, it will update to the Google Calendar. After that, it will send a message to the family members which are the contact persons of that elderly and report to them that what event elderly had finished.

Short Essay on Solving a Problem related to a Task Assigned to the Author

Nowadays, Elderly has many problems in their daily life, such as falling down, spraining, getting lost and so on. And a server problem, which is always happen on elderly, is forgetting what they need to do in a specific time every day. The most serious case is that elderly, who has server illness, forget using drugs. It may cause big problem. That is why I also want to include a function that the application can also let elderly know about what they need to do in a day. So that elderly can follow the event time list to do all the events in that day.

To start working on solving this problem, I designed using a Google Calendar as a share calendar to Elderly and their family members. It is because Google Calendar is a complete existing solution and it is useful for us to help users save all elderly's events day by day. To use Google Calendar function in our Android application, I have studied how to use the Google Calendar API in Android, so that I can use this API to

write codes in Android Studio to create functions for our application to access all Google Calendars of a Google email account with read and write functions.

And for the design of the calendar function, there is a page for the calendar function and all events in that day will be shown in that page. Those events will be changed when date is changed. There is also a date picker for users to pick a specific day and change to list out all events in the picked day. And there is insert function inside the application so that user can create event to the calendar easily through the application. For each event in the event list, only title, date and time will be shown so that the interface will not be so complicated. To view the description of each event, users can just click an event in the list. Then a dialog will pop up and description of that event will be shown. Also, there are three buttons for three different functions. First, users can delete the clicked event by clicking the delete button. Second users can update the details of the clicked event by clicking the update button. Finally, elderly can click the done button if they finished the clicked event. A message will be sent to report to their contact persons when elderly finished an event and clicked the done button. Moreover, the title of that event will also be updated as done so that their family members can also check the status of those events in the application.

As a result, with the above functions inside the application, elderly can be noticed about what they need to do every day by checking the calendar inside the application. Their family members can also be noticed about the status of their elderly's events by message and checking the calendar inside the application. And they can have a strong connection between each other.

Self-Appraisal of Contributions

I had helped a lot on the analysis part, system design part and development part of the project.

First, in the analysis part, we had thought about many problems that elderly commonly has nowadays. And the calendar function was the idea that I brought out. I think this idea because if we just think about the ideas of helping on caring the safety of the elderly, there are just too little ideas and finally the application will be too simple with too little functions. Also, another reason is that I think caring daily life of elderly is also a very important idea that we can also include to our application. That's why I brought out this idea and this part is also mainly handled by me.

Second, in the system design part, in Initial stage, we had also thought together about where we can save the data of our application. And it is related to another problem that if we need to create database or not. That's why for my calendar part, I start think about what existing solution can I use to help on my part. And finally, I use Google Calendar function as the calendar for saving the events of elderly. For other parts, like contact or fall detection, we also think that no need to create another database for our application to use. Also, I had also discussed to my groupmates if we really need to use firebase as Google Login function in our application. In addition, I had also helped on designing some parts of interface of the application.

Third, in the development part, I design and develop and whole calendar function. I had studied the Google Calendar API and used it in the calendar function. All problems and most of the bugs are also fixed by me. And I have also helped other groupmates about some of the coding problems, like the coding part of permission of the application.

The Open University of Hong Kong
School of Science and Technology
Bachelor of Computing with Honours in Internet Technology

COMPS456F – Software System Development Project

Final Year Project – Team Member Final Report

Elderly Mobile Life Assistant for Family Members with Extra Peace of Mind

Group Name: Whale Did You Go? Supervisor: Dr. Kelvin Kai Wing LEE

Student Name	Student ID Number	Signature
Au Chi Chung	12017765	

Appendix B2. Team Member Au Chi Chung's Final Report

Table 9 – Appendix: Team member Au Chi Chung's final report

Declaration Statement

I, Au Chi Chung (12017765), certify that the description and information included in this team member's report is true to the best of my knowledge.

Au Chi Chung: _____ (Signature and Date)

Task Assigned to the Author and their Status

The above table outlines the task assigned to me in this project:

Tasks	*Responsible Member(s)*	*Target Date*	*Rate*	*Remarks*
First Screening Presentation (Slide)	Leung Kwan Ho Au Chi Chung	3rd Oct 2019	100%	/
Initial Report	Leung Kwan Ho Au Chi Chung Lie Tze Wah	24th Oct 2019	100%	/
Second Screening Presentation (Slide)	Au Chi Chung	21st Oct 2019	100%	/
Interim Report	Au Chi Chung	6th Feb 2020	100%	/
Interim Demonstration (Recorded video of prototype)	Leung Kwan Ho Au Chi Chung Lie Tze Wah	25th Feb 2020	100%	/
Final Demonstration and Presentation (Slide) (Recorded video of prototype) (Video 2: Evaluation Plan, Conclusion)	Leung Kwan Ho Au Chi Chung Lie Tze Wah	20th May 2020	100%	/
Final Report	Au Chi Chung	8th Jun 2020	100%	/
Login function with splash screen (Firebase)	Au Chi Chung	/	100%	/
Fall detection function	Au Chi Chung	/	100%	Debug with Lie Tze Wah
Other functions (Multi-language, Crash screen, About)	Au Chi Chung	/	100%	/

Review and Appraisal of Code Section – Section 1 (Code: \ui\notification\NotificationFragment.java)

Fall detection to sense the value change accelerometers of in mobile (line 240 - 273):

```
@SuppressLint("SetTextI18n")
@Override
public void onSensorChanged(final SensorEvent event){
    x.setText("" + event.values[0]);
    y.setText("" + event.values[1]);
    z.setText("" + event.values[2]);

    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
        double sum = Math.sqrt( Math.pow(event.values[0], 2) + Math.pow(event.values[1], 2) +
Math.pow(event.values[2], 2) );
        DecimalFormat precision = new DecimalFormat("0.00");
        double accRound = Double.parseDouble(precision.format(sum));
        sensorReader.setText("" + accRound);

        if(switchState) {
            if(sharedPreferences.getBoolean(IS_FALL_DOWN, false)) {//if (accRound <= 2.0) {
                if (isAdded()) {
                    editor.putBoolean(IS_FALL_DOWN, true);
                    editor.apply();
                    mSensorManager.unregisterListener(this);
                    if(!sharedPreferences.getBoolean(ALREADY_FALLDOWN, false)) {
                        requireActivity().startService(new Intent(getActivity(),
AlertService.class));
                        showTimerDialog();
                    } else { }
                }
            }
        }
    }
}
```

Show count down timer dialog when fall detected (line 275 - 333):

```
private void showTimerDialog() {
    final AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(getContext());
    alertDialogBuilder.setTitle(R.string.notification_timer_confirm_message);
    alertDialogBuilder.setMessage("").setPositiveButton(R.string.notification_timer_cancel,
new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    dialog.cancel();
                    dialog_is_show = false;
                    editor.putBoolean(IS_FALL_DOWN, false);
                    editor.apply();
                    editor.putBoolean(ALREADY_FALLDOWN, false);
                    editor.commit();
                }
    });

    final AlertDialog alertDialog = alertDialogBuilder.create();
    dialog_is_show = true;
    editor.putBoolean(ALREADY_FALLDOWN, true);
    editor.commit();
    alertDialog.show();
    timer = new CountDownTimer(15000, 1000) {
        @Override
        public void onTick(long millisUntilFinished) {
            int seconds = (int) (millisUntilFinished / 1000);
            int minutes = seconds / 60;
            alertDialog.setMessage(getString(R.string.notification_timer_message) + " " +
String.format(Locale.getDefault(), "%02d:%02d", minutes, seconds));
```

```
            alertDialog.setOnCancelListener(new DialogInterface.OnCancelListener() {
                @Override
                public void onCancel(DialogInterface dialog) {
                    timer.cancel();
                    requireActivity().stopService(new
Intent(getActivity(),AlertService.class));
                    onResume();
                }
            });
        }

        @Override
        public void onFinish() {
            try {
                sendMessage();
                dialog_is_show = false;
                editor.putBoolean(IS_FALL_DOWN, false);
                editor.apply();
                editor.putBoolean(ALREADY_FALLDOWN, false);
                editor.commit();
            } catch (Exception e) {
                requireActivity().stopService(new Intent(getActivity(), AlertService.class));
            }
        }
    }.start();
}
```

Review and Appraisal of Code Section – Section 2 (Explain the design and operation of the code)

For the function of fall detection which called from Main Activity and operating by the code in Notification folder and performs the user interface by fragment_notification.xml. The user interface is divided into three areas including the information display of accelerometer on the top with real-time update, function switch button in the middle, and the area for managing emergency contacts at the bottom. I set up the onSensorChanged() method from the SesorEventListener with the fall detection algorithms for body attached accelerometers, when the value which is the sum of the value x, y, and z from the accelerometers are equals 2.0 that it decided to be a fall situation. Then, it will call the showTimerDialog() method for perform the ringtone alert and the SMS message method after the timer is time up.

Review and Appraisal of Code Section – Section 3 (Report any issue, problem, or mistake)

The most difficult part in fall detection is how to let the function working when the application is hiding (not showing in the front) or the application is closed (killed by task manager). And I have tried many ways for the performance including the foreground service in Android, which runs independently from other application components (like activities) but displays a persistent notification to the user if it is running. The original intention is to make the application unclosed and cannot kill by task manager that it can perform the fall detection when the application is hiding or killed, but the idea did not work.

After the changes by using Accessibility Service in Android, the problem is solved at all.

Short Essay on Solving a Problem related to a Task Assigned to the Author

For the task selected for discussion, we were assigned to the project of "Android App for The Elderly and Their Family Members", and the main purposes is to develop an application with a special home page having large and few numbers of icons for the elderly (Features may include the following: single-touch dialing to several phone numbers until successful, instant messages, alarms, voice-based control, voice or video monitoring, fall-detection and alerts, etc.).

We ensure the project aims of an application to help the elderly to avoid accidents with lack of care and handle their daily life management. Then, we decide the teammate handles the functions proposed from him including the new add-on functions after the first meeting with the supervisor. After the final decision, our team leader Leung Kwan Ho be responsible for the daily schedule function, which is the proposed, and teammate Lie Tze Wah be responsible for the single touch dialing function. Because I have some experience in the final year project, that I am responsible for the documentation and handle the fall detection function, which hoping to reduce the workload in the application development for other teammates.

As the situation in the appraisal of the code section, the prototype of fall-detection with alert and SMS message is created quickly because of the references and improvements, but the problem appears after the application test, the application cannot work when the application is hiding or the application has been killed by task manager. I have tried the method of foreground service in Android, which runs independently from other application components like activities but displays a persistent notification to the user if it is running. I hope to make the application unclosed and cannot kill by the task manager, but the idea did not work. After the long discussion and debug works with the teammate (Lie Tze Wah), we found the way that it can solve the above problem which called Accessibility Service.

To applying the accessibility service for the use of running applications in the background, it has larger changes for the program including adding a new class for handle the accessibility service and implements with the sensor event listener of the accelerometer. After that, we created some of the shared preference values for carrying the state of the application that it is running in the foreground and running in the background. Finally, the problem is already solved but it is more sensitive when detecting the fall when turning on the accessibility service, that it is the best way to perform the function by our team.

Self-Appraisal of Contributions

List of contributions:

- Documentation: all reports and presentation slides, literature research and content analysis
- System design: revise for the system diagrams
- Application development: base design, login, fall detection, crash screen, multi-language

For me, I think my contributions also have a certain weight in this project. In addition to my paperwork including evaluation plan, reports, and PowerPoint slides for those presentations, I have also built with the application to reduce the teammate's workload. The contents in the reports are also searched or base on the idea by me. After the discussion and the final decision of the application basic design, I have improved the system diagram from my teammates. At the same time, I often need to remind team members to improve the progress rate of the application development even in WhatsApp or Discord because the progress is not satisfactory. I dare not say that my contribution is the greatest in this team because teammate Lie Tze Wah helps for the debug works for other teammates in the application development without his own works, but I have also handled the base development including the bottom navigation layout, login page with animation, fall detection, crash screen, multi-language, about us page, etc. And the suggestion of a grade for me, I think B+ or higher mark is not an excessive suggestion because of my contributions.

The Open University of Hong Kong
School of Science and Technology
Bachelor of Computing with Honours in Internet Technology

COMPS456F – Software System Development Project

Final Year Project – Team Member Final Report

Elderly Mobile Life Assistant for Family Members with Extra Peace of Mind

Group Name: Whale Did You Go? Supervisor: Dr. Kelvin Kai Wing LEE

Student Name	Student ID Number	Signature
Lie Tze Wah	12123709	

Appendix B3. Team Member Lie Tze Wah's Final Report

Table 10 – Appendix: Team member Lie Tze Wah's final report

Declaration Statement

I, Lie Tze Wah (12123709), certify that the description and information included in this team member's report is true to the best of my knowledge.

Lie Tze Wah: _____ (Signature and Date)

Task Assigned to the Author and their Status

The above table outlines the task assigned to me in this project:

Tasks	*Responsible Member(s)*	*Target Date*	*Rate*	*Remarks*
Initial Report (Use case diagram)	Leung Kwan Ho Au Chi Chung Lie Tze Wah	24th Oct 2019	100%	/
Interim Demonstration (Recorded video of prototype)	Leung Kwan Ho Au Chi Chung Lie Tze Wah	25th Feb 2020	100%	/
Final Demonstration and Presentation (Recorded video of prototype) (Video 2: Methodology)	Leung Kwan Ho Au Chi Chung Lie Tze Wah	20th May 2020	100%	/
Single touch dialing function	Lie Tze Wah	/	100%	/
Instant message function	Lie Tze Wah	/	100%	/
Accessibility Service	Lie Tze Wah	/	100%	/
Fall detection function (Debug)	Lie Tze Wah	/	100%	Debug
Daily schedule function (Debug)	Lie Tze Wah	/	100%	Debug

Review and Appraisal of Code Section

In the Super Seniors application, quick contact function has 2 parts. One is single touch dialing, the second is instant message. Because of the single touch dialing is the main function of quick contact, and it contains many techniques and problems in the development, I will focus on introduce about this.

Here is a part of the code for single touch dialing:

```
@Override
public void onAccessibilityEvent(AccessibilityEvent event) {

   if (event.getEventType() == AccessibilityEvent.TYPE_WINDOW_CONTENT_CHANGED) {
      AccessibilityNodeInfo info = event.getSource();

      if (info != null && info.getText() != null) {
         String duration = info.getText().toString();
         String zeroSeconds = String.format("%02d:%02d", new Object[]{Integer.valueOf(0), Integer.valueOf(0)});
```

```
        String firstSecond = String.format("%02d:%02d", new Object[]{Integer.valueOf(0), Integer.valueOf(1)});

        if (zeroSeconds.equals(duration) || firstSecond.equals(duration)) {
          c.cancel();
          editor.putBoolean(REDIAL_REMINDER, false);
          editor.commit();
          count = 3;
        }

        info.recycle();
      }
    }
```

In Android call system, there are 2 main statuses of outgoing call are "CALL_STATE_OFFHOOK" (when you are dialing or the call is answered), and -> "CALL_STATE_IDLE" (end of the dialing or call). However, we cannot know whether the call is answered in the status of "CALL_STATE_OFFHOOK". Based on Android does not provide a clear call status for developer to know what the current status of outgoing call is. I use AccessibilityService to monitor the screen change. If the word "dialing" become to "00:00", "00:01", … on the screen, that means the outgoing call is answered.

In the onAccessibilityEvent(), if the service detected the change of "dialing" to time format on the screen, that means the outgoing call is answered. And it call c.cancel() to stop the redial countdown timer and reset the redial times to 3. It solved the problem of cannot know whether the call is answered in status "CALL_STATE_OFFHOOK". And onAccessibilityEvent() is keep running when user turn on the AccessibilityService in setting.

```
public class MyBroadcastReceiver extends BroadcastReceiver {

  @Override
  public void onReceive(final Context context, final Intent intent) {

      if(intent.getAction().equals(Intent.ACTION_NEW_OUTGOING_CALL)) {

        curr_calling_no = getResultData();
        c = new CountDownTimer(20000, 1000) {

         @Override
         public void onTick(long millisUntilFinished) { }

         @Override
         public void onFinish() {
           try {
             TelephonyManager telephonyManager = (TelephonyManager)
getSystemService(Context.TELEPHONY_SERVICE);
             Class classTelephony = Class.forName(telephonyManager.getClass().getName());
             Method methodGetITelephony = classTelephony.getDeclaredMethod("getITelephony");
             methodGetITelephony.setAccessible(true);
             Object telephonyInterface = methodGetITelephony.invoke(telephonyManager);
             Class telephonyInterfaceClass = Class.forName(telephonyInterface.getClass().getName());
             Method methodEndCall = telephonyInterfaceClass.getDeclaredMethod("endCall");
             methodEndCall.invoke(telephonyInterface);

             count--;
             editor.putBoolean(REDIAL_REMINDER, true);
             editor.commit();

           } catch (Exception e) { }
         }
```

```
        }.start();
    }

}
```

In the MyBroadcastReceiver, if it knows the current action is outgoing call, it creates a 20 seconds countdown timer. If the outgoing call is not answered in 20 seconds, it automatically and compulsory cut off the dialing and reduce once redial time. In here, I call a method named "endCall" in class "Telephony" to stop the outgoing call. It uses try and catch because this method provides by TelephonyManager hidden API, Android does not provide it in clear. The status "CALL_STATE_OFFHOOK" become to "CALL_STATE_IDLE". It automatically redials in 10 seconds again.

Short Essay on Solving a Problem related to a Task Assigned to the Author

Quick contact function is mainly solving the problem of elderly daily life management. We noticed that some elderly cannot contact their family when they are in emergency. The main reason may be the UI/UX design of phone contact is difficult to use for them. Also, elderly's nervousness is another reason to make this problem. So, I redesign the contact method for elderly to make the operation become more easily. The design direction is simple for use and direct to see.

In the traditional contact method, elderly need to find out the phone a number in the number set when who want to call someone. If elderly want to send a message to someone, he/she also needs to hand-write the message because many elderlies do not know the input method. The handwriting also has many problems such as mobile phone cannot identify the words or elderly do not know how to delete the text. However, elderly need redoes the above step when the call is not answered. So, we see the traditional contact method is complicated and not user friendly for them.

In the new contact method, we provide a maximum of 3 emergency contact people and some type of message template. For the single touch dialing, elderly need saves the emergency contact number before using it. The contact's photo and name will display in a conspicuous place. When elderly want to call with their family, just single touch the icon that can finish the whole step for calling. Also, when the out-going call is not answered, the application will automatically redial twice.

Compare with the traditional method, elderly skip to find out the phone number in the set and it will automatically redial. Elderly can see the contact people immediately and the operation also is simpler. So, the new contact method is meeting my design direction. Improved the elderly difficult to contact with their family.

For the instant message, our template covers some daily life activities such as going out, shopping, and emergency. Elderly click the message button and the date, time, and event selector will popup. Elderly just single touch the list item that can send the message to the emergency contact people easily.

Compare with the traditional method, elderly do not need to hand-write the message and find the contact. Elderly just easily tap the item and follow the step that can send the message. It is convenient for elderly's daily life.

So, the new contact method provides a simple and direct environment for elderly, make them easy to operate. In the new method design, we consider all possible problems when elderly is using their phone.

As much as possible integration and simplify the step of operation, therefore the single touch dialing, and instant message function were born.

Self-Appraisal of Contributions

In this project, I am mainly in system design, system implementation and testing, and problem analysis and formulation.

In the problem analysis and formulation, we discuss the background and problem of this issue. We found out there are 3 main topics of elderly's daily life problems. They are elderly's daily activity reminded, contact in emergency and daily injuries. Based on these situations, we provide some solution such as calendar, quick contact and fall detection. Among them, quick contact is proposed by me. It based on my experience, many elderlies think the smartphone is too difficult to use such as they do not know how to send a message, spend too much time finding the contact and do not know some button icon's meaning. So, I think "single touch" is important for them. Also, it is a new way of design for quick contact.

In system design, our application data mostly are contact's name, image, phone number and elderly's location. These data not as sensitive as credit card information, it does not need to verify and operate in server. So, I think the data store in local storage is the most effective. On the calendar function, we choose to use google calendar because it is an online platform. Elderly can share a calendar with their family, anyone can mark some event on google calendar and it will update immediately. And it is easy to unify smartphone of different companies and implement. So, user need to login when the first time to use.

In system implementation and testing, I am mainly in backend system. Single touch dialing was designed, developed, and debugged by me such as I find some problems in listening to the phone status and some smartphones cannot use single touch dialing function. For the malfunction of single touch dialing, it works in my Samsung Note5 with Android 7.0 and Android VM with Android 9.0. But it fails on my Samsung Note8 with Android 9.0, I think it may relate to some smartphone models cannot monitor to the screen change. At the same time, I find AccessibilityService can detect phone's situation without Super Seniors is running. So, I share this information for my group member and rewrite the fall detection in a little bit that it works when Super Seniors was killed. On the other hand, I help my group member debug and test their function.