

Mobile Application for The Visually Impaired

BY

DARREN TAN YUNG SHEN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology
(Kampar Campus)

JUNE 2020

REPORT STATUS DECLARATION FORM

Title: **Mobile Application for the Visually Impaired**

Academic Session: JUNE 2020

I DARREN TAN YUNG SHEN
(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

Darren

(Author's signature)



(Supervisor's signature)

Address:

3-8-7, JALAN GANGSA,

GREENLANE HEIGHTS,

11600 PENANG

LIM JIT THEAM

Supervisor's name

Date: 11/9/2020

Date: 11/9/2020

Mobile Application for The Visually Impaired

BY

DARREN TAN YUNG SHEN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology
(Kampar Campus)

JUNE 2020

DECLARATION OF ORIGINALITY

I declare that this report entitled “**Mobile Application for The Visually Impaired**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : *Darren*

Name : DARREN TAN YUNG SHEN

Date : 11/9/2020

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Mr. Lim Jit Theam who has given me this bright opportunity to engage in a mobile application project. It is my first step to establish a career in the field of developing mobile application. A million thanks to you.

Finally, I must say thanks to my friends and my family for their love, support and continuous encouragement throughout the course.

ABSTRACT

This is a mobile application project designed to assist visually impaired users. The core function of the application is to help users interpret the things in their surrounding without the need to get in much contact with it in order to know what it is. This function is more commonly known as object detection. Other than that, there would be additional features which can contribute in the visually impaired users' daily life. They will be able to experience a possible self-relying usability of a mobile application.

Developing this mobile application involves the usage of Java and Android Studio as its integrated development environment. The interface and some additional features would utilize much of Java programming. An android device will be used for testing the application constantly.

This project will be running on prototyping methodology, hence there will be lots of testing conducted on the application. The goal is to ensure our visually impaired users can experience an unsophisticated navigation within the application and frequent testing by prototyping can help in achieving the goal.

TABLE OF CONTENTS

FRONT COVER	i
REPORT STATUS DECLARATION FORM	ii
TITLE PAGE	iii
DECLARATION OF ORIGINALITY	iv
ACKNOWLEDGEMENTS	v
ABSTRACT	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	xi
LIST OF TABLES	xiv
LIST OF ABBREVIATIONS	xv
CHAPTER 1 INTRODUCTION	1
1.1 Problem Statement and Motivation	1
1.2 Background	2
1.3 Project Scope	3
1.4 Objectives	4
1.5 Impact, Significance and Contribution	5
1.6 Highlights of Project Achievement	6
1.7 Report Organization	7
CHAPTER 2 LITERATURE REVIEW	8
2.1 TapTapSee	8
2.2 Aipoly Vision: Sight for Blind & Visually Impaired	11
2.3 Google TalkBack	13
2.4 Be My Eyes	14
2.5 BlindTool	15
2.6 Seeing AI	17
2.7 Envision AI	19
2.8 Talking Goggles	21
2.9 Comparison between Reviewed Mobile Applications for The Visually Impaired including our Proposed Solution	24

CHAPTER 3 SYSTEM DESIGN	25
3.1 Methodology	25
3.2 Use Case Diagram	27
3.3 Activity Diagrams	28
3.3.1 Startup Application	28
3.3.2 Homepage Activity	29
3.3.3 Dial to Call Activity	30
3.3.4 Register Contact Activity	32
3.3.5 Recognize Object Activity	33
3.3.6 Recognize Text Activity	35
3.3.7 Voice Assistant Activity	37
3.3.8 Exit Application	38
3.4 Sequence Diagram	39
3.5 Collaboration Diagram	40
CHAPTER 4 INITIAL PROTOTYPE	41
4.1 Tools	41
4.1.1 Development Environment	41
4.1.2 Hardware	41
4.2 Users Characteristics	42
4.3 Requirements	42
4.4 Dial to Call	43
4.4.1 Application Logic	43
4.4.2 Application User Interface Design and Audio Files	45
4.5 Prototyping	46
4.5.1 Start Up	46
4.5.2 Homepage	47
4.5.3 Dial to Call	48
4.5.4 Register Contact	50
4.5.5 Recognize Object	51
4.5.6 Recognize Text	53
4.5.7 Voice Assistant	55

CHAPTER 5 CUSTOMER EVALUATION	56
5.1 Evaluation Method	56
5.2 Evaluation Outcome	56
5.2.1 What would be the best feedback a mobile application could allow visually impaired users to perceive?	57
5.2.2 Do you think navigation between pages like a "BACK" button or "EXIT" button is needed for a mobile application built for the visually impaired?	57
5.2.3 Is it important for a visually impaired person to be able to open the mobile application on his own?	57
5.2.4 If the application is also built for the partially impaired (those who can still see but vision is very blurry), which user interface do you think is more suitable for them?	58
CHAPTER 6 PROTOTYPE REFINEMENT	60
6.1 New Requirements	60
6.2 Screens of All User Interfaces in Refined Prototype	61
6.3 New Features of Refined Prototype	62
6.3.1 Integration of Audio and Vibration Feedback	62
6.3.2 Inclusion of Navigation Button on Every Necessary Activities	63
6.3.3 Unlocking phone starts up application	63
6.3.4 Design of User Interface	64
6.4 Post-Refinement of Prototype	65
6.4.1 Development	65
6.4.2 Testing	65
6.4.3 Maintenance	67
CHAPTER 7 CONCLUSION	68
7.1 Personal Insights	68
7.2 Project Review	68
7.3 Future Work	69

BIBLIOGRAPHY	70
APPENDIX A CUSTOMER EVALUATION OUTCOME	A-1
A.1 What would be the best feedback a mobile application could allow visually impaired users to perceive?	A-1
A.2 Do you think navigation between pages like a "BACK" button or "EXIT" button is needed for a mobile application built for the visually impaired?	A-1
A.3 Is it important for a visually impaired person to be able to open the mobile application on his own?	A-1
A.4 If the application is also built for the partially impaired (those who can still see but vision is very blurry), which user interface do you think is more suitable for them?	A-2
APPENDIX B POSTER	B-1
APPENDIX C PLAGIARISM CHECK RESULT	C-1
APPENDIX D FYP 2 CHECKLIST	D-1

LIST OF FIGURES

Figure Number	Title	Page
Figure 2.1.1	TapTapSee logo	8
Figure 2.1.2	Screenshot of TapTapSee recognizing wallet and key	9
Figure 2.1.3	Screenshot of TapTapSee recognizing laptop	9
Figure 2.1.4	Screenshot of TapTapSee’s requiring TalkBack accessibility setting	10
Figure 2.1.5	Screenshot of TapTapSee having trouble recognizing pillow	10
Figure 2.2.1	Aipoly Vision logo	11
Figure 2.2.2	Things that can be recognized by Aipoly Vision	11
Figure 2.2.3	Screenshot of Aipoly Vision settings page	12
Figure 2.3.1	Google TalkBack	13
Figure 2.4.1	Be My Eyes logo	14
Figure 2.4.2	Main page of Be My Eyes for the visually impaired	14
Figure 2.4.3	Volunteer’s view of video call with the visually impaired	14
Figure 2.5.1	BlindTool logo	15
Figure 2.5.2	Screenshot of Blind Tool recognizing sunglass	16
Figure 2.5.3	Screenshot of Blind Tool recognizing computer keyboard	16
Figure 2.5.4	Screenshot of Blind Tool’s failure in recognizing wastebasket	17
Figure 2.5.5	Screenshot of Blind Tool’s failure in recognizing toothbrush	17
Figure 2.6.1	SeeingAI logo	17
Figure 2.6.2	Image of SeeingAI describing a scene after recognition	18
Figure 2.6.3	Things that can be scanned and recognize by Seeing AI	19
Figure 2.7.1	Envision AI logo	19
Figure 2.7.2	Things which can be done with Envision AI	20
Figure 2.8.1	Talking Goggles logo	21
Figure 2.8.2	Screenshot of Talking Goggles’ failure to recognize laptop	22
Figure 2.8.3	Screenshot of Talking Goggles’ failure to recognize spectacles	22
Figure 2.8.4	Screenshot of Talking Goggles recognizing pillow and returning clustered outputs	23

Figure 2.8.5	Screenshot of Talking Goggles recognizing window blind and returning clustered outputs	23
Figure 3.1.1	Prototype Methodology	25
Figure 3.1.2	Screenshot of a review of Talking Goggles	26
Figure 3.1.3	Screenshot of a review of BlindTool	26
Figure 3.2	Use Case Diagram	27
Figure 3.3.1	Startup Application Activity Diagram	28
Figure 3.3.2	Homepage Activity	29
Figure 3.3.3	Dial to Call Activity Diagram	30
Figure 3.3.4	Register Contact Activity Diagram	32
Figure 3.3.5	Recognize Object Activity Diagram	33
Figure 3.3.6	Recognize Text Activity Diagram	35
Figure 3.3.7	Voice Assistant Activity Diagram	37
Figure 3.3.8	Exit Application Activity Diagram	38
Figure 3.4	Sequence Diagram	39
Figure 3.5	Collaboration Diagram	40
Figure 4.1.1	Logo of Android Studio	41
Figure 4.4.1.1	Project Structure of Initial Prototype	43
Figure 4.4.1.2	Object Detection Folder Structure	44
Figure 4.4.2	Application User Interface Design and Audio Files	45
Figure 4.5.1	Startup Screen of Initial Prototype	46
Figure 4.5.2	Homepage Screen of Initial Prototype	47
Figure 4.5.3.1	Dial to Call Screen of Initial Prototype	48
Figure 4.5.3.2	Manifest Call Permission Request	49
Figure 4.5.4.1	Register Contact Screen of Initial Prototype	50
Figure 4.5.4.2	Manifest Read/Write Contact Permission Request	51
Figure 4.5.5.1	Recognize Object Screen of Initial Prototype	51
Figure 4.5.5.2	Manifest Camera Permission Request	52
Figure 4.5.6.1	Recognize Text Screen of Initial Prototype without Text Recognition Result	53
Figure 4.5.6.2	Recognize Text Screen of Initial Prototype with Text Recognition Result	53
Figure 4.5.6.3	Text Recognition Image Edit Screen of Initial Prototype	54

Figure 4.5.6.4	Manifest Camera and Read/Write Storage Permission Request	54
Figure 4.5.7.1	Voice Assistant Screen of Initial Prototype	55
Figure 4.5.7.2	Manifest Record Audio Permission Request	55
Figure 5.2.4.1	Homepage interface prioritized on colour contrast	58
Figure 5.2.4.2	Homepage interface prioritized on beauty	58
Figure 6.2.1.1	Startup Screen of Refined Prototype	61
Figure 6.2.1.2	Homepage Screen of Refined Prototype	61
Figure 6.2.1.3	Dial to Call Screen of Refined Prototype	61
Figure 6.2.1.4	Register Contact Screen of Refined Prototype	61
Figure 6.2.1.5	Recognize Object Screen of Refined Prototype	61
Figure 6.2.1.6	Recognize Text Screen of Refined Prototype (without result)	61
Figure 6.2.1.7	Recognize Text Screen of Refined Prototype (with result)	62
Figure 6.2.1.8	Voice Assistant Screen of Refined Prototype	62
Figure 6.4.2	Question regarding sufficiency of refined prototype's features	66

LIST OF TABLES

Table Number	Title	Page
Table 2.9	Comparison between Reviewed Mobile Applications for The Visually Impaired including our Proposed Solution	24
Table 4.1.2	Technical Specifications of Hardware Used for Development	42
Table 4.2	Users Characteristics	42
Table 4.3	Requirements of initial prototype to be implemented	42
Table 6.1	Requirements of prototype to be refined	60

LIST OF ABBREVIATIONS

<i>e.g.</i>	exempli gratia (for example)
<i>et al.</i>	et alia (and others)
<i>AI</i>	Artificial Intelligence
<i>IDE</i>	Integrated Development Environment
<i>OCR</i>	Optical Character Recognition
<i>XML</i>	Extensible Markup Language
<i>OS</i>	Operating System
<i>CPU</i>	Central Processing Unit
<i>GPU</i>	Graphics Processing Unit
<i>GHz</i>	gigahertz
<i>RAM</i>	Random Access Memory
<i>ROM</i>	Read-only memory
<i>GB</i>	gigabyte
<i>URI</i>	Uniform Resource Identifier

Chapter 1

Introduction

1.1 Problem Statement and Motivation

The first problem statement is the inability of visually impaired users to perceive information by the sense of sight. Visually impaired people usually rely on their sense of touch, hearing, smell and taste. They have to challenge themselves daily in coping with their impaired vision in a universe where majority of the people perceives their environment with vision (Koestler 1976). Hence, our project will try to provide a solution to the problem.

Besides that, communication and access to information is a huge problem for the visually impaired people (Arati, Sayali, Sushanta & Harshata 2015), especially when it comes to terms of utilizing a smartphone to perform some actions. Performing a call or knowing your phone's certain status using a smartphone is difficult for visually impaired users. Many mobile applications developed for the visually impaired out there do not provide extra functionality like that for this category of users to access. Their sole focus is only on retrieving information of items interpreted by the mobile camera. There are several more aspects that are yet to be explored and has potential in assisting those who are visually impaired via mobile application.

Next, visually disabled people have limited accessibility to smart mobile devices' functionality (Dobosz 2017). Nowadays, most of the devices such as tablets and smartphones require users to only touch on the surface of the screens (Dobosz 2017). Besides facing difficulty in navigation throughout a mobile application, starting it is a very challenging task because it requires these visually impaired to seek for the exact position of the shortcut key or icon in order to turn on the application. Situations like that motivate this project to be developed to provide assistance to those in need.

Redundancy of objects under varying conditions is also among the problems in object recognition which many researchers are concerned of. This issue is often caused by the changes in camera angle, lighting, and sizes. This problem will definitely not benefit users who are visually impaired. Blind users are unable to know or see the object to estimate the distance of it away from the phone camera. It is also very arduous to fulfill the concept of inter-class similarity (e.g. computer and

television) and intra-class variability (e.g. different types of chairs) due to these factors (Jabnoun, Benzarti & Amiri 2017).

Another problem would be the constant reliance of some object recognition applications on an accessibility setting known as “VoiceOver” for iPhone users and “TalkBack” for Android users. These applications rely on the accessibility settings to return audio outputs to the users. Without it, the users will be left wondering whether they have opened the right application or have to navigated to the right interface. Due to the fact that the settings can only be configured by accessing the built-in accessibility option, visually impaired users will also have a hard time navigating all the way into the settings configuration themselves and require another person with proper vision to handle that.

1.2 Background

According to the World Health Organization, there are approximately 285 million people who have visual impairments, 39 million of them are blind and 246 million have decreased visual acuity. Close to 90% of the visually impaired are living in low-income nations. There are 30000 visually impaired people who have been identified in Tunisia; including 13.3% of them are blind. Impaired visions can cause severe consequences on some visual function capabilities (Jabnoun, Faouzi & Amiri 2015). These people will have trouble conducting daily living activities like walking. Vision will be required at a certain distance depending on how bad the person’s sight is. Read, write, communication would also be troublesome. They will also have problems in evaluating displacements and space. Every activity requires a prolonged and well maintained visual attention. It is why it is very essential for these group of people to understand their environment and knowing what objects or people are surrounding them. Knowing what is around them can also provide relief to them.

There are several kinds of aids which have been created by different researchers to help the visually impaired. There is mobility aid where motions and vibrations play a huge part in guiding the blind through obstacles. Its applications included guiding canes and wheelchairs. Another method would be computer vision or artificial intelligence.

The main focus of our project is to provide assistance for the low vision and the visually impaired community by creating and developing a mobile application with an

object recognition system implanted in it. However, given the available advancing technologies and algorithms, there are still existing issues which are bothering researchers.

Today, object recognition plays a very essential role in the innovation of computer vision and artificial intelligence. Modern research of computer vision had already existed in the early 1960s (Andreopoulos & Tsotsos 2013). Since then, object recognition has been put into good use for various fields of applications such as surveillance, automated vehicle systems and image retrieval (Vahab et al. 2019). Generally speaking, this method is used to identify or classify distinct objects by extracting several features using machine or deep learning algorithm. Several samples will be trained as input to allow the machine to learn up and remember a few unique characteristics of a certain object so it can be used to categorize and specifically identify other similar objects. Different AI algorithms and training models will be utilized in bringing success to this project.

The people with visual impairment clearly do not rely on their sense of sight. Therefore, this project optimizes blind users' auditory and tactile senses involvement. Besides the compulsory design of big buttons to help these users get contact with it, graphical user interface will not serve much purpose so it should not be prioritized much.

In order to provide assistance to those with visual impairment, the configuration or programming of a mobile device or application normally begins at the level of operating system. Many types of extra features have been implemented to simplify access for these group of people. Available application interface have always been utilized by developers to produce solutions (Dobosz 2017). Therefore, the people with visual impairment who want to use mobile devices are well supported. In this project, we will aim to provide a strong support for them with whatever best ways possible.

1.3 Project Scope

First of all, an object recognition function will be built into our mobile application. The mobile application should be able to detect certain items from the camera and return an audio output to announce what it is. In order to allow the mobile camera to recognize objects, machine learning has to be involved. Tensorflow will be utilized in this project because it serves as a platform for the expression and

implementation of machine learning algorithms. Tensorflow allows expressed algorithm computations to be executed on various heterogeneous systems including mobile applications (Abadi et al. 2015). On the other hand, Python programming language will be used to train and test object recognition models for further implementation into the project's mobile application based on a Tensorflow reference documentation which was first released in November 2015 (Abadi et al. 2015). Every object model trained will have its own label. Thus, the label text returned will be converted into speech through the Android's Text To Speech engine. The visually impaired will then perceive the detected object through listening.

Other than recognizing objects, the mobile application should also be able to detect text and return a speech output of it to the user. This enables the users to read words or sentences from anywhere including books or food packages. Android's Text To Speech is involved in the implementation of this function.

Besides that, users will be able to dial numbers from keypad to call from the mobile application but the function would have to take the visually impaired users' senses other than sight into account. The keypad would produce sounds or even vibrations to indicate the numbers or call button which are being pressed.

Moreover, there will be a voice assistant implanted into the application for users to speak out the information which they would like to acquire such as time and phone battery status. These are all programmed using Java and the codes are tested and executed in Android Studio.

This project is also very focused on providing visually impaired users an understandable navigation and unsophisticated accessibility while utilizing our mobile application. In order to achieve that, every function access button would return an audio output signifying the function which the users are about to access. It could save them time from asking for others' help to handle their device. This design could prevent the users from accessing the wrong function or application and getting themselves confused in finding the right way to navigate throughout the application. Java and Android Studio are also highly involved in this part of development.

1.4 Objectives

The first objective of this project is to provide assistance for the people with visual impairment. This application aims to bring convenience into these people's life.

It gives these users an opportunity to experience a different perception of their surroundings. The visually impaired can rely lesser on their other senses like touch or smell in order to recognize object with the existence of object recognition function in our application. They are able to save a little time and energy from moving closely to perceive the objects or words that they wish to have knowledge of.

Another objective is to optimize the usability of a smartphone application for the people with visual impairment. The development of this mobile application will focus mainly on enabling the users to perceive through the sense of auditory (hearing) and tactile (touch). The application is supposed to return audio output during the navigation throughout the application. For instance, if the user pressed a button, the button should indicate its function by audio.

Besides that, the project aims to integrate the object recognition function with other functions which are inaccessible by the blind users on a usual mobile application. It will be an additional feature which makes our mobile application unique as there currently close to one applications out there which combines these functions together. Rather than just allowing users to recognize object with the mobile application, this mobile application's development will include several simple functions like calling or saving contacts. For example, if they face trouble or whatsoever, they can simply access this mobile application and contact someone without worrying that they may not be able to access the phone's basic application like calling. This is because the usual mobile navigation will confuse and complicate the visually impaired users, hence wasting time and risking the users' life if it is an emergency.

1.5 Impact, Significance and Contribution

This project highly contributes to the society with visual impairment. Why? The first reason would be the development of the applications which takes the blind users' senses into consideration. The design aims to optimize the usage of their auditory and tactile senses. It is able to ease users in navigating throughout a mobile application and also save their time in accessing the function they which to utilize. Another reason is the existence of an object and text recognition function. The people with visual impairment will be able to 'see' or 'read' again without the need to get close

tactile contacts with the object. Once users know what it is, they can interpret information based on the recognition result.

The significant feature of this project is the integration of the object recognition function with other functions like calling and voice assistant. These integrated features are able to allow users to seek help from another party in order to solve a certain problem. Getting people involved in a blind users' life is essential because there would still be things that are unable to be carried out by a blind person. On the other hand, voice assistant could solve phone-related problems so users would know what to do next when information is retrieved through the audio output. The accessibility into those functions will be visually-impaired-oriented so users got nothing much to worry about getting lost during the navigation throughout the mobile application.

1.6 Highlights of Project Achievement

The first highlight of this project is its ability to turn on automatically once users have unlocked their smartphone. It uses a service class by Android which will be explained later in further chapters. Visually impaired users are able to prevent turning on the application using the conventional method, which is by tapping on the application shortcut.

Next, this application includes the recognition of texts and objects which would be beneficial for those visually impaired. However, having the ability to recognize these things are not enough. These recognitions will return audio outputs of their results for users to perceive.

Besides that, users are able to call via this application. However, the keypad is designed for the perception of visually impaired users. Another feature is contact registration. This part might require the assistance of a third party to input the values currently but the buttons are still designed for the visually impaired. These features are created to help users with visual impairment carry out these functions without having to go through the conventional method of using a smartphone.

Moreover, there is a voice assistant which could help the user to carry out some functions or retrieve and understand some information. This activity will receive users' voice input and return results in audio form for users to perceive.

Final feature which plays a huge and important role in the application is its unique accessibility built for the users with visual impairment. There is a combination of audio and vibration feedback which can be triggered to notify users about their actions.

1.7 Report Organization

This report will comprise 7 chapters. Chapter 1 will be discussing about problem statements, background, project highlights, project scope and objectives. Literature reviews will be in Chapter 2. This chapter discusses about existing mobile applications that are assisting the visually impaired community.

Chapter 3 includes the project methodology and diagrams describing the flow of every activities in the mobile application developed in this project. The diagrams are use-case, activity, sequence and collaboration. Chapter 4 describes the initial prototype and requirements which needed to be fulfilled while developing it.

Customer evaluation will be discussed in Chapter 5. The outcome gained from the users' feedback published by Google Form will be discussed. Chapter 6 is about the refinement of the prototype. The design and implementation of the refined prototype will be explained.

Finally, Chapter 7 will conclude the entire project. Personal insights, project overview and future works will be discussed in this chapter.

Chapter 2

Literature Review

In this chapter, we will look deeply into the applications that are already developed as a solution to some of our problems. Those existing applications will be reviewed so the problems or mistakes of those applications would not be repeated in our project. Now, let us explore among those applications.

2.1 TapTapSee



Figure 2.1.1

Firstly, there is TapTapSee which is designed for the blind or low vision users in hope that it would help increase the users' daily independence. It is developed by CloudSight Inc., an organization which apply computer vision to provide detailed object recognition as one of its solutions (CloudSight 2019). TapTapSee utilizes mobile devices' camera for recognition of objects and screen reader functions from the devices' accessibility settings to speak the identifications of objects (TapTapSee 2013).

One of its substantial features is its ability in performing picture or scene recognition in less than a minute. This shows that it has a robust object recognition model. It also allows the previous identification to repeat in conjunction with the convenience of their target users. Users are also able to upload images from their phone gallery to identify objects. Flash can be automatically toggled in the application, just in case the environment is dim or dark. This feature helps to enhance the recognition operation by the illumination of environment.

TapTapSee is very beneficial to the community with visual impairment. It can be proven with its first strength which is its minimal design. Considering the applications

is for users who are not able to look at the application, it is specifically designed to be advertisement-free so it would not hinder the functionality of the application for the users (Crone 2015). Another strength would be its highly accurate computations. The application excel in determining the type and colour of objects, including its environment.

Figure 2.1.2 and Figure 2.1.3 depicting the interfaces after recognizing different objects on TapTapSee.



Figure 2.1.2

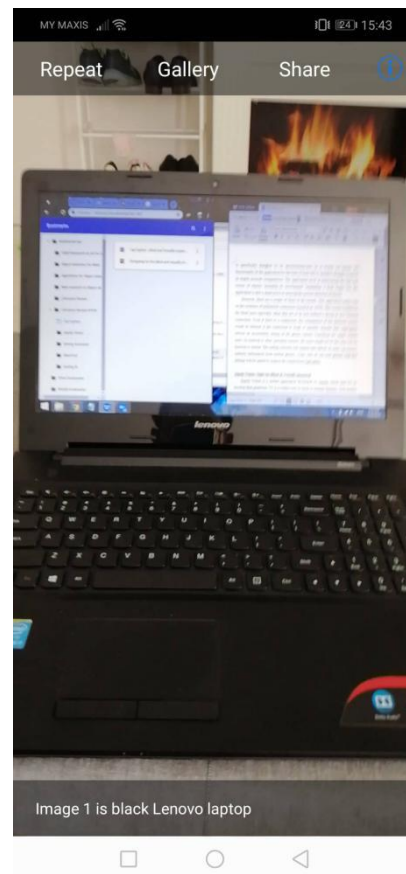


Figure 2.1.3

However, there are also some problems in the system which could potentially cause inconvenience for the users. Firstly, TapTapSee relies on an accessibility setting of the phone, namely VoiceOver for Apple mobile users (as shown in Figure 2.1.4) or TalkBack for Android users (TapTapSee 2013). The setting converts text outputs into speech so users can receive auditory information from mobile phones. Users who do not own phones with this settings will be unable to acquire the output from TapTapSee but that is not the main problem. Users with visual impairment are unable

to navigate into the phone settings to toggle the accessibility settings. That could cause confusion for the user because they need auditory or tactile response in order to understand their actions in a smartphone.

Another flaw is holding the camera at a certain distance away from an object might have the chance of not returning any result. An example can be shown in Figure 2.1.5. It is a screenshot of the application which is trying to recognize a pillow on a sofa but holding the camera too close to it will be difficult for it to return a result.

Besides that, since it is an application built for the people with visual impairment, it did not return audio output whenever the application has started up. Users are unable to verify whether it is time for them to perform any action in the application. This is a problem which our project should look into while developing.

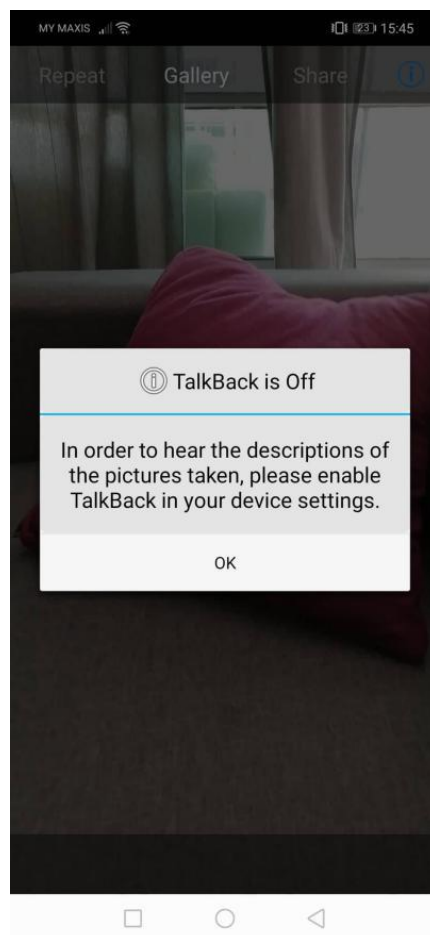


Figure 2.1.4



Figure 2.1.5

2.2 Aipoly Vision: Sight for Blind & Visually Impaired

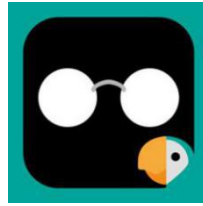


Figure 2.2.1

Aipoly Vision is a mobile application developed by V7 Ltd with the aim to help the visually impaired and possibly colour blind people to perceive the environment. Users only have to focus their phone camera at their interested object and press a button for the type of recognition they wish to perform. That will activate the artificial intelligence and return audio output for the users (V7 Ltd 2017).

This mobile application is able recognize plants, animals, texts, currency, food and colours but some of the features are only accessible if you are subscribed to the services which incurs monthly fee. Figure 2.2.2 shows some of the categories of objects the application is able to recognize. The application is also embedded with multiple languages like Italian, French and German, including English which allows users from different part of the world to experience the application (V7 Ltd 2017).

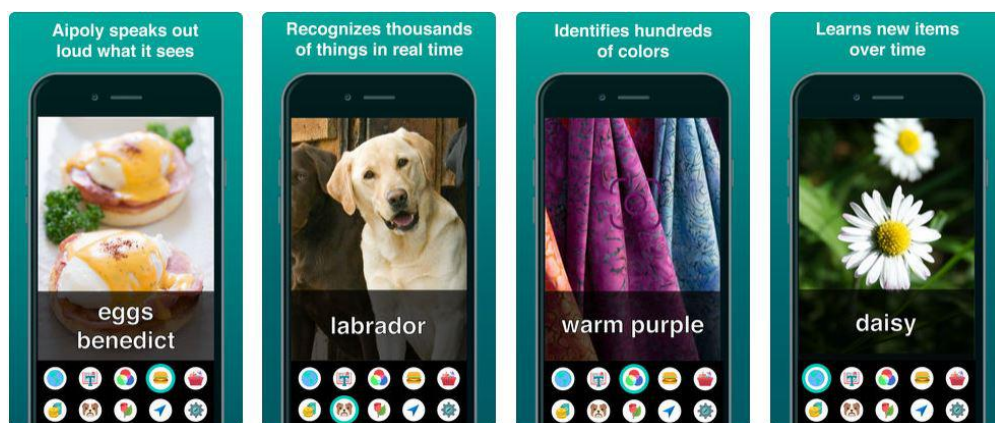


Figure 2.2.2

The application is able to recognize a large varieties of objects. To be more specific, over 1000 food dishes, over 1400 colours, over 2000 species of plants and animals including a thousand of essential items can be recognized using this application. Given the amount of objects the application is able to detect and

recognize, the datasets collected by the developers must be huge and often trained to evolve the existing models. This is beneficial for the visually impaired because they could now rely more on this application to recognize stuffs.

Besides, Aipoly Vision is also equipped with an Intelligent Torch (V7 Ltd 2017) which automatically turns on the phone's flash light whenever the surrounding is dark. It further improves accuracy in recognizing objects and colours in this sort of difficult conditions. Figure 2.2.3 displays the Aipoly Vision settings which include the automatic torch or better known as Intelligent Torch.

Furthermore, no accessibility settings like VoiceOver is needed to be used for the application to speak its output because a speech system has been embedded into the application as shown in the figure below. Users can either choose to enable the speaking voice or not.

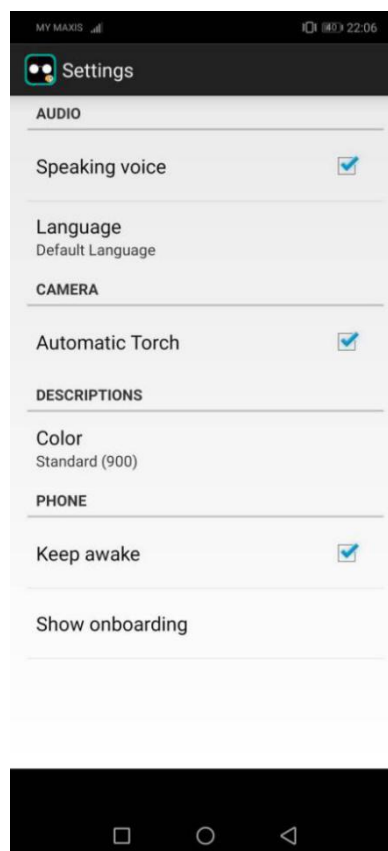


Figure 2.2.3

However, audio outputs are not created at necessary places or during certain navigation in Aipoly Vision. The blind users are unable to understand which interface they are accessing without an audio output which describes it. They also would not

know when the application has successfully started up. Therefore, this could be an aspect of Aipoly Vision which can be improved by the developers.

2.3 Google TalkBack



Figure 2.3.1

TalkBack is an accessibility service provided by Google (as displayed in Figure 2.3.1) with the purpose to assist those people with visual impairment in interacting with a smartphone device. Vibration and audible feedback are utilized in order to tell the users things that are displayed on the screen or what is the user touching and what can be done with it (Hildenbrand 2014).

The existence of this service enables users to be able to explore or navigate as comfortably as they want. Those with visual impairment do not need to worry about accidentally pressing something and accessing something that they do not wish to. TalkBack is also able to return audio output of whatever is written on the phone screen. Once TalkBack tells the users whatever they have touched, users will be able to act by double tapping or move on to other parts of the application without triggering anything (Hildenbrand 2014). If any of the TalkBack settings does not fit the users' requirement, it can still be configured in the settings.

However, the disadvantage of TalkBack would be the difficulty faced by the visually impaired users in activating the accessibility service themselves. They are

also unable to configure the settings which they desire. The toggling or configuration of TalkBack requires another person to activate. Although TalkBack is very beneficial for the users with visual impairment, it is still not the optimum solution in some ways as mentioned before.

2.4 Be My Eyes

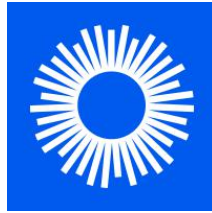


Figure 2.4.1

Be My Eyes is a mobile application that allows community of the visually impaired to connect with a group of volunteers. In this application, people are able to register themselves as either a volunteer or a visually impaired user. The application supports multilingual community (Be My Eyes 2020) so users do not need to worry about having trouble in conversing in their respective languages.

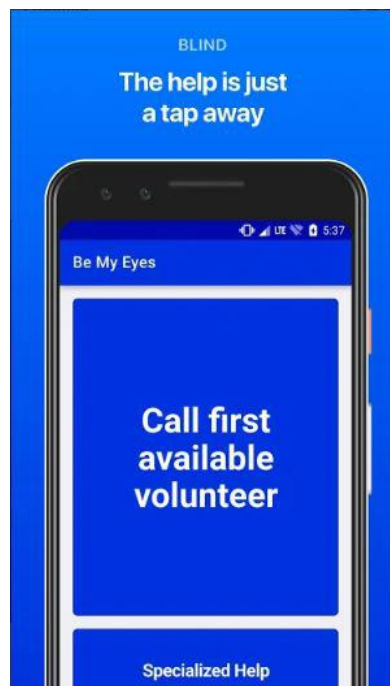


Figure 2.4.2

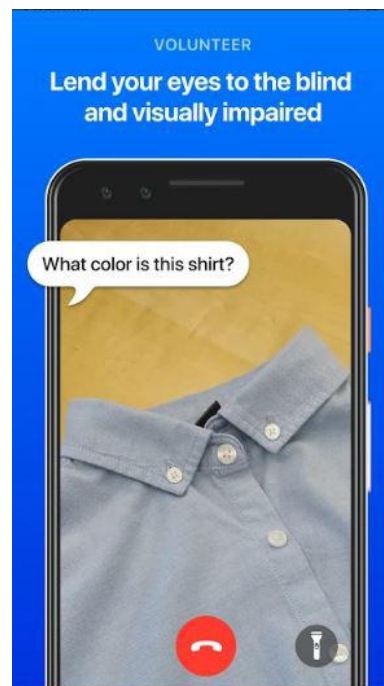


Figure 2.4.3

This application functions through the channel of video call. A visually impaired user will make a call by tapping a huge button that is written “Call first available volunteer” on it (depicted in Figure 2.4.2). Then, notification will be sent to an available user who can choose whether to answer the call or leave it to other volunteers.

The application is beneficial to the community of visually impaired because people who accept calls are equipped with proper vision and the recognition process will definitely be fast and accurate. The visually impaired will be able to read labels, television navigation, identify clothes, operating kiosks (Be My Eyes 2020) and many more possible actions which can be done by the human eye with the help of the volunteers.

The application’s calling function design also helps the visually impaired in a way that the call button is big enough for the blind users to be able to get in touch with. It can reduce the chances of the users accidentally pressing other parts of the device’s screen and navigate to wrong applications or interfaces.

However, the navigation throughout the application still seem difficult for a visually impaired user. These users will not be able to independently access and use the application without the help of audio or vibration outputs. In order to do so, accessibility settings like TalkBack has to be involved to provide assistance in this aspect.

2.5 BlindTool



Figure 2.5.1

BlindTool is a mobile application developed by Joseph Paul Cohen, a University of Massachusetts Boston student who holds a Ph.D Degree in Computer Science and Machine Learning (Cohen 2019). His current focus is on medical applications concerning deep learning. He designed the application with an objective of assisting

the blind to recognize objects. The CNN (Convolutional Neural Network) embedded in the application can understand up to a thousand ImageNet based classes (Cohen 2015).

While the technology behind this application may be complex, the usage are only very simple. Users only need to direct their phone camera onto the object they wish to detect. The application will then focus the camera onto a certain object. What makes this app extra great is that the output of the detected object is not only spoken aloud, it vibrates when an output is returned (Klotz 2016). The vibration depends on its confidence score. The more confident the application is in recognizing an object, the more vibration it will produce, allowing the visually impaired to also analyse and understand the object themselves.

Although the computation time is fast, the recognition is not precise. It often returns results which are irrelevant and incorrect. A huge database with numerous images will slow down the recognition process of the application for a smart phone, so Cohen opted for efficiency and speed over accuracy (Lasane 2016). There was even a quick test conducted in an office to verify the results' precision. Objects like sunglasses and keyboard were correctly identified during the first attempt but other objects such as watches and running shoes returned disappointing results. A toothbrush was never recognized accurately. The application must have wrongly vibrated when recognizing a wastebasket because it returned an a result of a toilet seat (Lasane 2016). Results like this make users to doubt in the application and its training models. The positive results are displayed in Figure 2.5.2 and Figure 2.5.3 while the false ones are in Figure 2.5.4 and Figure 2.5.5.

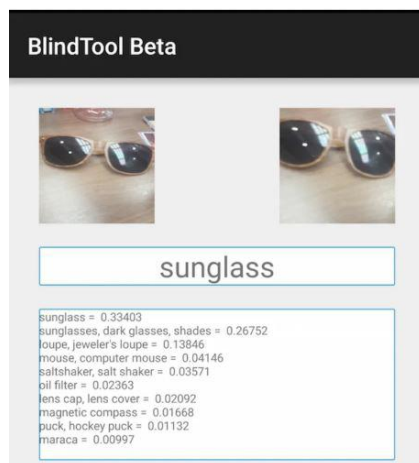


Figure 2.5.2



Figure 2.5.3

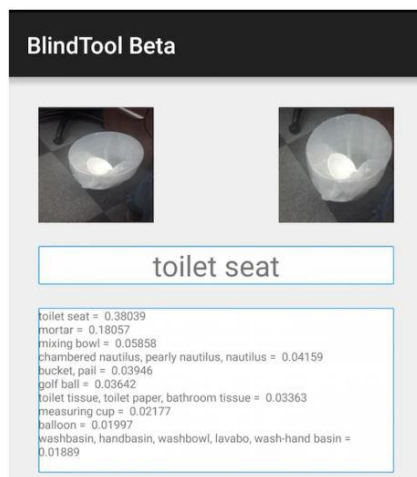


Figure 2.5.4

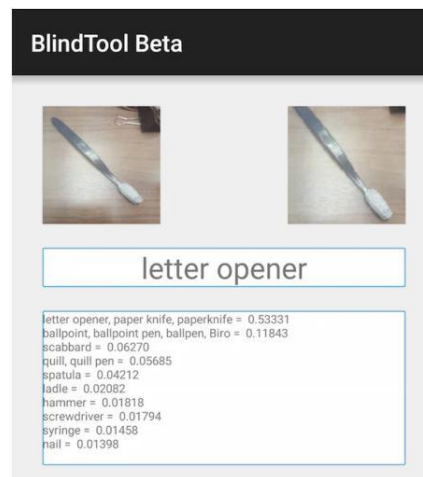


Figure 2.5.5

There is also another weakness in the application which plays an important role in a certain environment. The application does not turn on the phone's flashlight when the surrounding is dark (Marquart 2016). There is no single toggle button that can at least switch the flashlight no or off. Inability to turn on flashlight especially when in a dark environment normally is risky for people, but in this case, the application will not be able to detect and recognize objects around the blind users. Therefore, the visually impaired have to use their own other senses to detect stuffs and the application does not serve any purpose anymore.

2.6 Seeing AI



Figure 2.6.1

Seeing AI is a free mobile application created by Microsoft Corporation. It is currently only available for Apple users. The software is designed for the blind and low vision community. The power of Artificial Intelligence is being harnessed for this research project to widen the visual world (Microsoft Corporation 2019).

Seeing AI excels in having the capability to detect various things. The application that immediately speaks texts that appear in front of the phone camera. It also helps users to read documents and there is no need to worry if the document is not well-placed into the frame of the camera. This is because the application would provide audio guidance to capture the page of texts, understand its format and read the document aloud. Products for shopping can also be scanned via barcodes to return the result of the object name and possibly its details (Microsoft Corporation 2019).

Besides that, colours and handwriting can also be detected and recognized. The application also has an audible tone which can be generated corresponding to the surrounding's brightness. Another special characteristic of this software is its ability to recognize people whose images can be saved for a better identification. Once it is saved, the application is able to return the person's approximate age, gender and emotions when scanned by the phone camera (Microsoft 2019). The application is also able to describe scenes captured as shown in Figure 2.6.2.



Figure 2.6.2

Based on the number of classes and objects, there must be many inputs which are inserted into the training model of the application's learning machine. The different

classes recognized is depicted in Figure 2.6.3. This is advantageous because there will be less or not be any redundancy of objects while the recognition is performed.

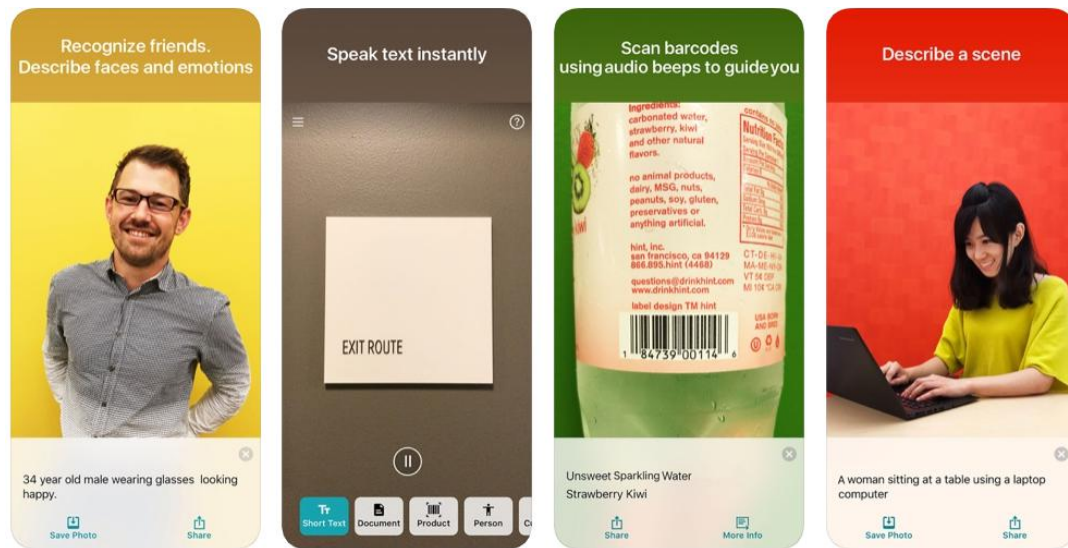


Figure 2.6.3

Unfortunately, VoiceOver, an accessibility setting of iPhone users is required for the application to return audio outputs for the visually impaired to perceive (Quantum 2018). In order to access the phone setting, the blind users will have to request assistance from another party to help them configure. The blind users will have to seek help from a person who is close to them physically. If the users are alone, they will also face a hard time because it is hard for them to perform a call to another person via a smartphone because of the difficult touchscreen navigation. They can only perform a call unless the VoiceOver setting is already configured to be functioning.

2.7 Envision AI

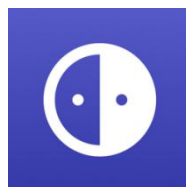


Figure 2.7.1

Envision AI is mobile application developed to assist the visually impaired by describing their surroundings. Artificial Intelligence and OCR (Optical Character

Recognition) are utilized in the core functions of this application (Envision Technologies BV 2020).

This application is very beneficial in the sense that it is able to detect and recognize various items such as texts, handwriting, colours and barcodes. Reading would not be a huge problem for those visually impaired who make use of this application. They can also easily purchase stuffs from possibly grocery stores with the help of the barcodes scanning function which is able to return product details as an output for the user to perceive. The color detection is not only able to help those who lose their sight, it can assist people who are color blind by returning the correct output of colours the application detects. Figure 2.7.2 portraits things which can be done by the application.

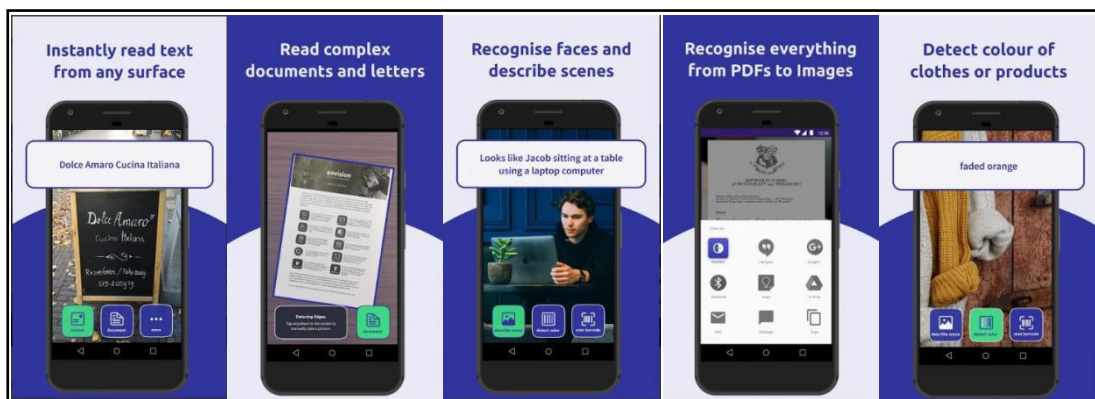


Figure 2.7.2

Besides that, this application has a special feature which is able to train the application to recognize faces which can be included during a scene description. It is able to save a little time for users while interpreting certain scenes. For example, instead of having the need to always ask about images' details which includes people in it, they can recognize themselves by using the recognition function of this application after training Envision to remember certain faces.

The weakness of this application would be its reliance on the support of TalkBack. Its interface is well-developed and the description of functions are clear. However, turning on TalkBack would be an extra and difficult step for them to take especially when there is nobody around to help them navigate throughout the application or mobile device.

2.8 Talking Goggles



Figure 2.8.1

Sparkling Apps developed Talking Goggles to speak out images of things around us which are recognized. Interestingly, this application do not distinguish the desired scanned objects (e.g. text, scene and people) by buttons. All users have to do is target the phone camera at the desired recognized item in order to return an output in the application. Landmarks, logos, books, texts and several more are able to be recognized with this application (Sparkling Apps 2019).

This application features a mode of video camera so users are allowed to identify an item after another without having the need to stop and take photos for recognition. The video camera button is located at the bottom right of the application. Whenever the application detects texts, OCR will be used to try recognizing and returning an audio output of it (Holton 2013).

This application also excels in its ability to return audible feedback to the users to indicate what it has detected. Besides, it can also return the output of the language you desire and all you have to do is configure.

Flash toggle is also one of the components in this application. As we know, flashlight benefit users big time especially in dark environments. It enhances users' vision to see in the dark so the accurate output would be produced. Besides benefiting users that way, this application is able to help their users read in the dark because this application is able to recognize texts as stated before. It can save users the trouble to go around in the dark looking for a light to switch on and off. The application also merged many languages into its database so users around the world are able to hear and understand their desired output.

In terms of weakness, this application has a high probability in returning errors of "No Close images Found" in recognizing everyday items (Holton 2013) like a laptop or a pair of reading glasses as shown in Figure 2.8.2 and Figure 2.8.3 respectively. If

this is an often occurrence, the users can hardly benefit from the object recognition application. It is as if the application does not even function at all.

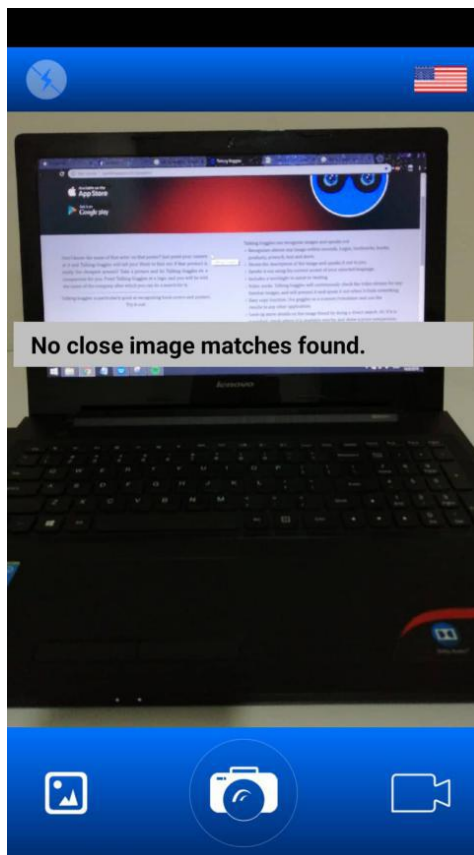


Figure 2.8.2



Figure 2.8.3

Besides that, there are times where the application would return texts as an output instead of the object after recognizing. Talking Goggles has also been programmed to read texts as we know, so OCR will recognize them whenever it sees texts. For example, the phrase “swash up” is returned instead of a screwdriver after scanning (Holton 2013). The phrase could be the brand of the screwdriver or just some random words it found on the screwdriver. It might be helpful sometimes but it still seems a little bit messy for the object recognition model to be doing that. Therefore, it is still better to design an object recognition model and interface which classifies the type of objects that users desire to recognize.

Another weakness in this application is there is no differentiation between the outputs which are returned to the users. It just return a jumbled up recognition output to the user without categorizing them to the users. In Figure 2.8.4, you can see the second result is colour. However, what if it was just a random text detected? The

visually impaired would not be able to know because they cannot perceive by sight. Figure 2.8.5 also displays a confusing result of recognized object.

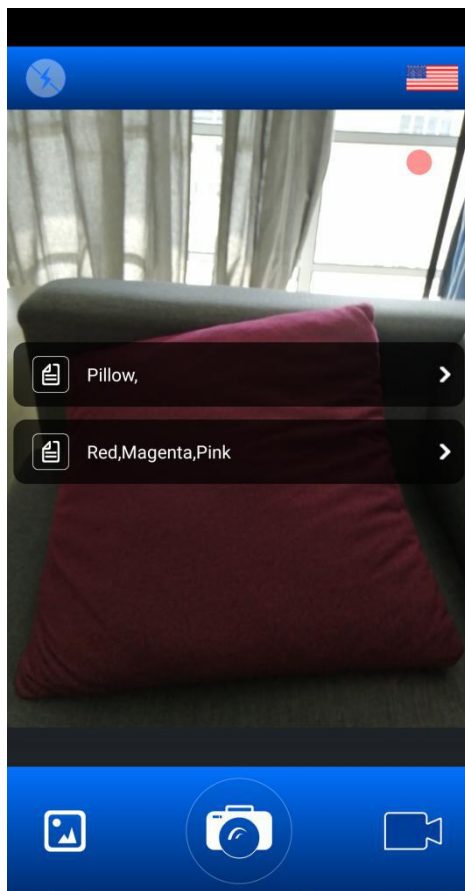


Figure 2.8.4

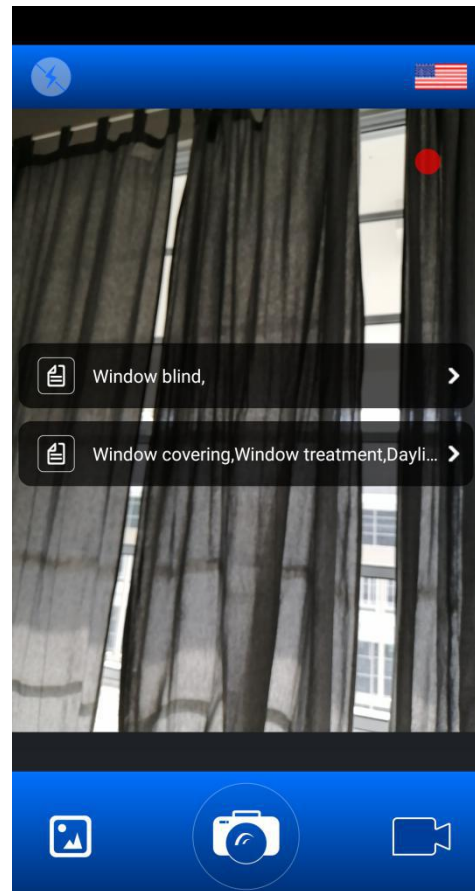


Figure 2.8.5

Another flaw of this application is its lack of guidance in telling the users which button does what functions. There should be at least an audio feedback returned to tell the users whenever they have pressed the button so they know what to do next. For example, the video streaming could return an audio output to indicate that the camera is currently recording and recognizing.

2.9 Comparison between Reviewed Mobile Applications for The Visually Impaired including our Proposed Solution

Features Applications	Object Recognition	Audio Feedback	Vibration Feedback	Call Function	Text Recognition	Barcode Recognition	Color Recognition
TapTapSee	✓						✓
Aipoly Vision	✓	✓			✓		✓
Google TalkBack		✓	✓				
Be My Eyes	✓			✓	✓		✓
BlindTool	✓	✓					
Seeing AI	✓				✓	✓	✓
Envision AI	✓				✓	✓	✓
Talking Goggles	✓				✓		✓
Proposed Solution	✓	✓	✓	✓	✓		

Table 2.9

Chapter 3

System Design

3.1 Methodology

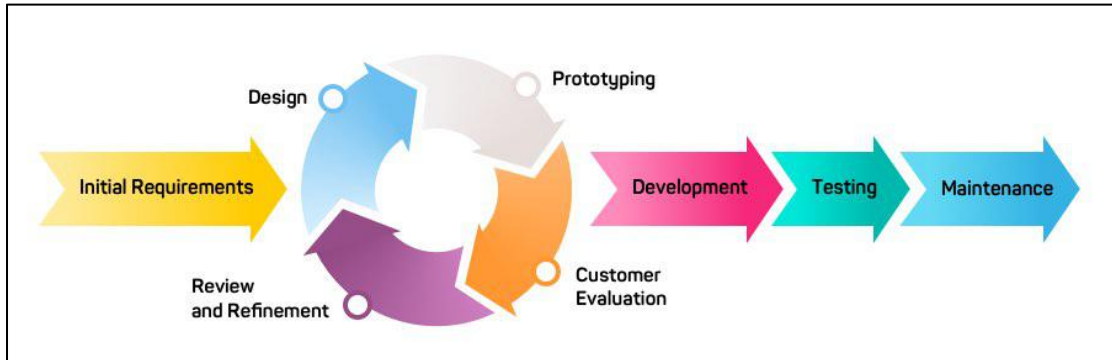


Figure 3.1.1

The methodology which is implemented for the realization of this project is prototyping. A representation is portrayed in Figure 3.1.1. It is a way of developing our mobile application in iteration to fulfill the requirements of the project's target users.

First step in this methodology is requirements analysis. This is the part where the project's target users will be studied in order to acquire an insight of the demand desired by the users. The mobile application prototype will be built according to the requirements gathered here. During the implementation of this phase, we looked into different applications which assist the visually impaired out there and studied the reviews given by those applications' users. Bad reviews provided by the users will be the developers' error which our project will aim to avoid repeating. An example of review taken into consideration is the review of Talking Goggles in Google Play Store as shown in Figure 3.1.1.2. Another example is a review of BlindTool. Figure 3.1.1.3 displays the review. The user reviewing BlindTool was clearly frustrated of the applications recognition results as the application does not seem to be beneficial at all. This gives the project a lesson to learn to ensure recognition results are accurate.

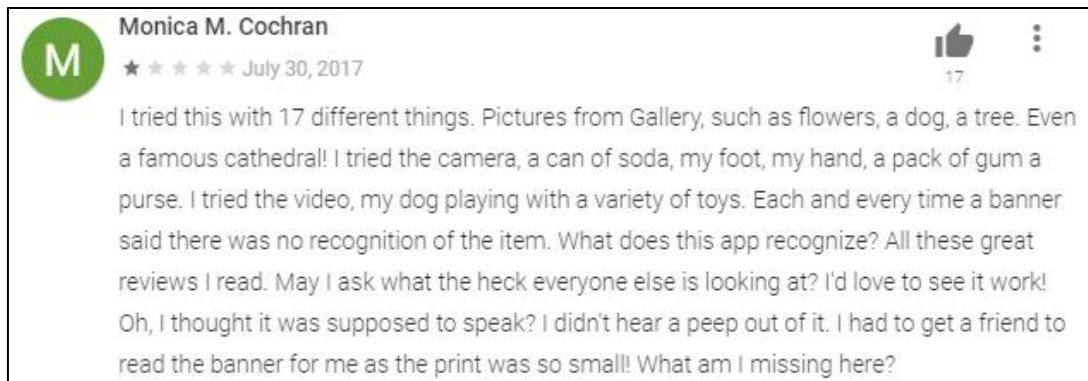


Figure 3.1.2

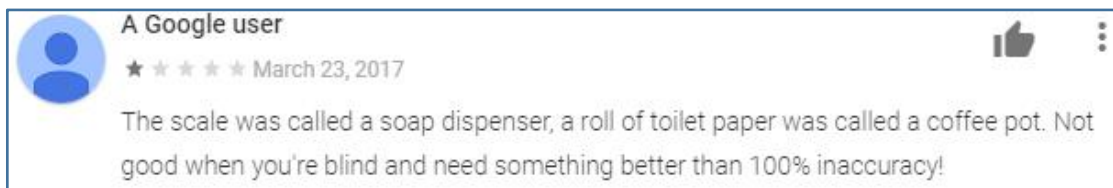


Figure 3.1.3

Next, a simple and quick design was created to portray the functionality and navigation which will exist in our mobile applications. As our mobile application is built for people with low to no vision, the buttons created will be larger so they will be able to cover more part of a certain button by their sense of touch. They do not need to worry about not being able to get in touch with a certain function due to a tiny button. Activity and sequence diagrams are also involved in representing an approximate application flow of our project. This designing phase will help in the prototype development.

Developing a prototype will only involve a lightweight development of basic functions in the application. The end result of the prototype has to provide the users an idea of how the application is going to work and what features will be included for further development.

After that, the prototype will be presented to the available users for evaluation. Users will have the chance to point out the missing aspects from our project and suggest ways to improve the application. Users' suggestions and feedback will be reviewed and work of refining the mobile application will be done according to beneficial reviews.

Design, prototype development, users' evaluation and project refinement will be iterative until users are almost satisfied with the application. When the prototyping

phase is done, the mobile application will finalize its development and testing. Maintenance of the application can be done from time to time to monitor the condition of the application.

3.2 Use Case Diagram

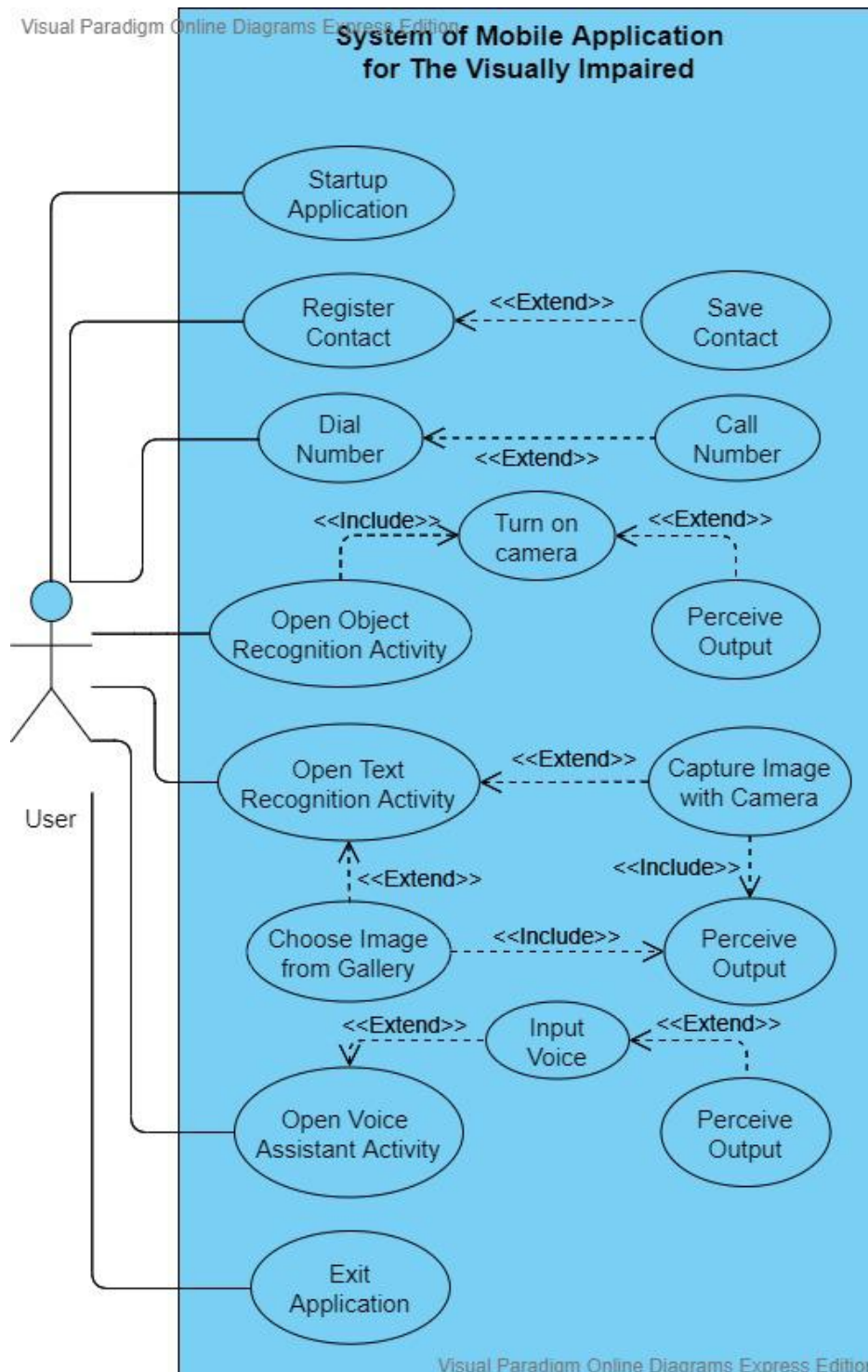


Figure 3.2

3.3 Activity Diagrams

3.3.1 Startup Application

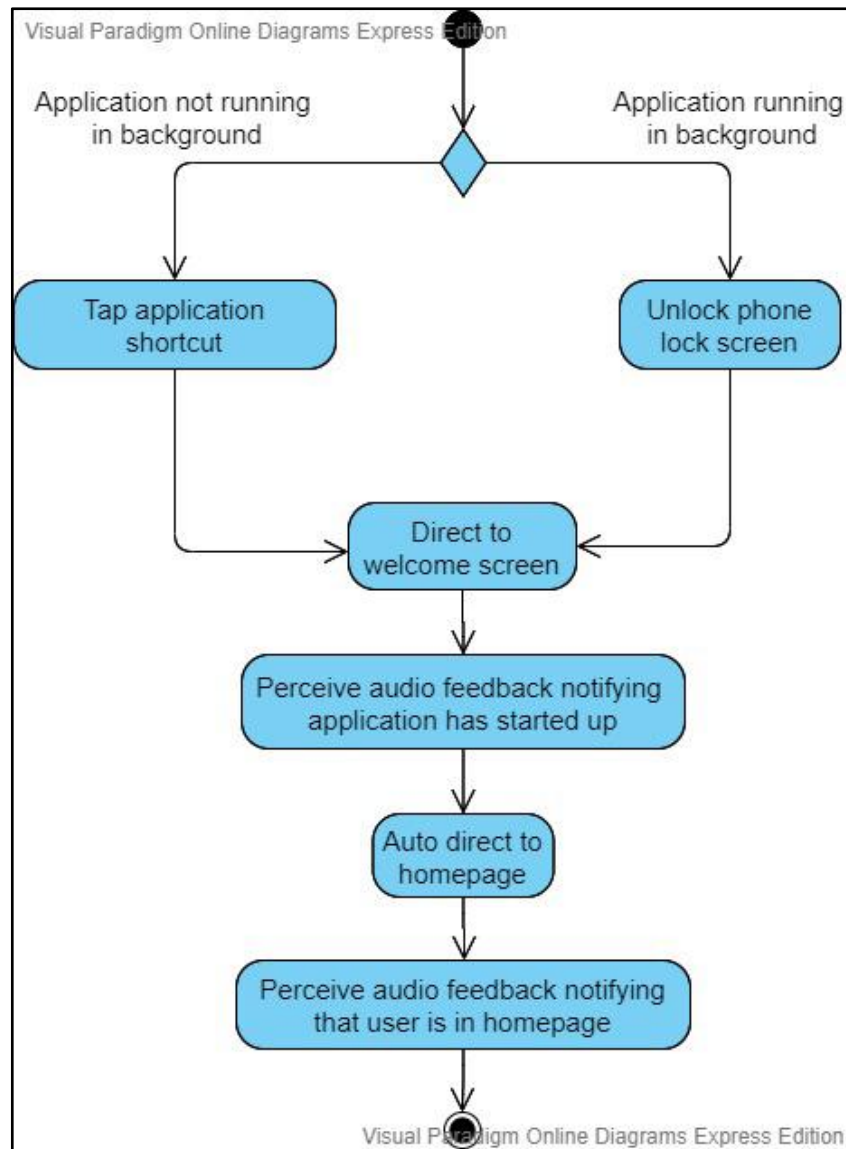


Figure 3.3.1

There are two ways a user can start up this mobile application. If users are downloading the application for the first time, they will need help from a third party to tap the application shortcut so the application can have a service running in the background. Once the service is running in the background, the user will then have a second option of opening the application. Ideally, the visually impaired users are able to start up the application by just unlocking their phone screen as long as the application is running in the background. Therefore, there is no need for them to worry about finding the application shortcut in order to open this application.

Once the application is started up, users will be directed to a splash screen (denoted by Android) or simply known as the welcome screen which will return an audio feedback saying “Welcome” to indicate users the application has started up. This screen is programmed to show for only 2.5 seconds.

After that, users are automatically directed to the homepage where the application will notify users by sounding the audio of “Homepage” to tell users that they are currently in the homepage of the application.

3.3.2 Homepage Activity

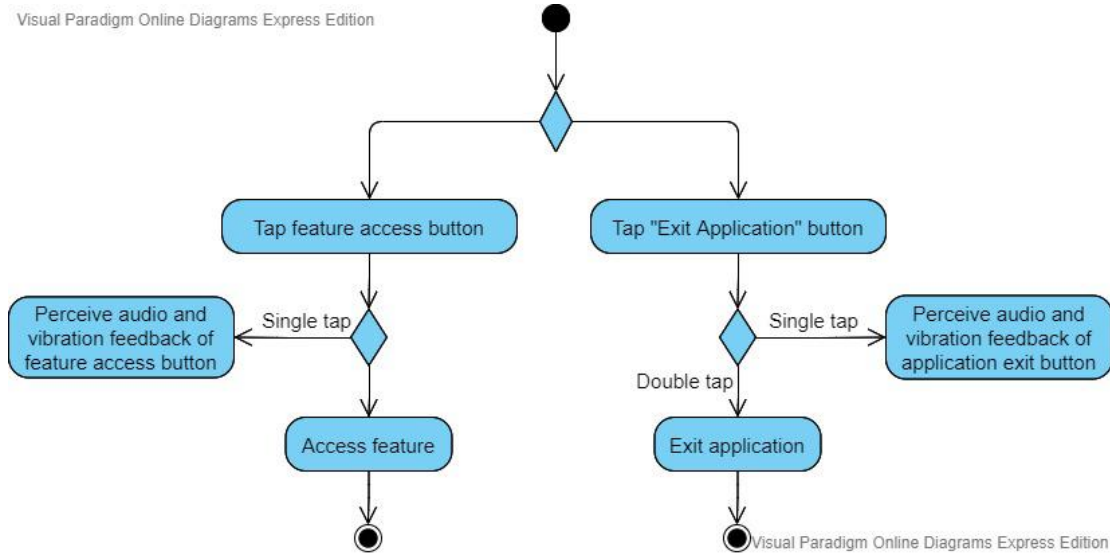


Figure 3.3.2

In homepage, users are able to choose whether to access and use the application’s features or exit the application.

In order to perform either both types of actions, users need to double tap to confirm performing the action. Otherwise, the users are able to tap once on either of the buttons to receive audio feedback of whether that is the action they would like to perform. Tapping once would also vibrate to notify users that they are still unable to perform that action. Meanwhile, after double tapping, the activity will not only provide audio feedback to tell user what feature it is, it will also sound a bell-like sound to let user know that they are directed through.

3.3.3 Dial to Call Activity

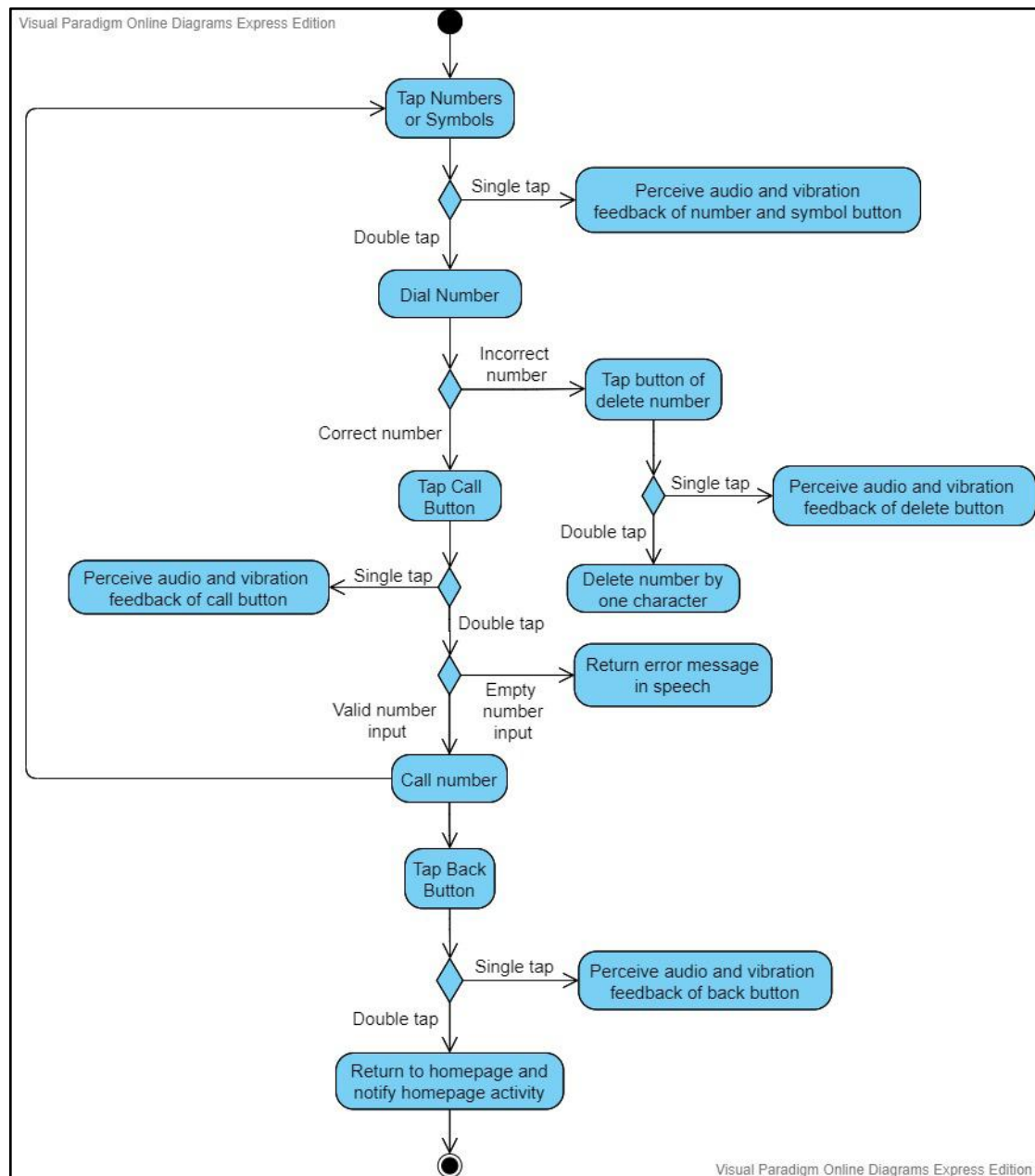


Figure 3.3.3

In this page, you will see an interface which will look like the regular call function of any other smartphones. Indeed, this is an activity of calling but it is designed for the people with visual impairment. Just like other call functions, users will first input the numbers then only hit the call button to perform call.

In our application, users will first singly tap either one of the numbers or symbols to interpret the button's representation. It could be the number 1, 2, 3, # or other number or symbols on the keypad. Tapping on one of those buttons will return

vibration and an audio feedback signifying the value of the keypad. For instance, if user tap one the button '1' once, the application will vibrate and sound the audio speaking '1'. This will only indicate the keypad value but not actually dial it into the number input for calling.

Now, if the users confirm the number is part of a certain string of a phone number, they can then double tap any of those numbers to actually dial them into the numbers input for calling. After double tapping, not only will the application return an audio feedback saying the number aloud, it also sounds a bell-like sound for confirmation of number input so users will know the value that they have entered. When users have confirmed the every number in the phone number string they wish to dial, they can proceed to double tap the numbers to perform the actual dial into the phone number input.

During the process of dialing, users are able to delete a character or perform a backspace function if they found out that they have inputted the wrong number into the string. Similar like other buttons before, single tap will indicate users the function of delete button while double tapping it will perform the actual deletion of a character from the string.

When users have finished inputting the correct number for calling, they can then double tap the call button for calling. If they have trouble finding the call button, they can definitely tap once slowly everywhere in order to tap the right button. This is because single tapping the buttons will indicate the buttons' functions including the call function.

Now, the call button function has a simple logic behind that prevents user from calling an empty number. Therefore, double tapping the call button with an empty input field of phone number will return an error to the users. The error will also speak itself to notify the users that the phone number is empty.

The following part of the calling activity of this application will be seen in the most of the further activities which will be describe in this report. This part is a feature called back. It is created to assist the visually impaired navigate themselves back into the homepage of the application where they could either choose to perform other actions or exit the application. The button functions similarly like any other buttons created in this application. Single tapping it will tell users that the button is for

directing back to homepage while double tapping it will actually navigate users back to homepage.

3.3.4 Register Contact Activity

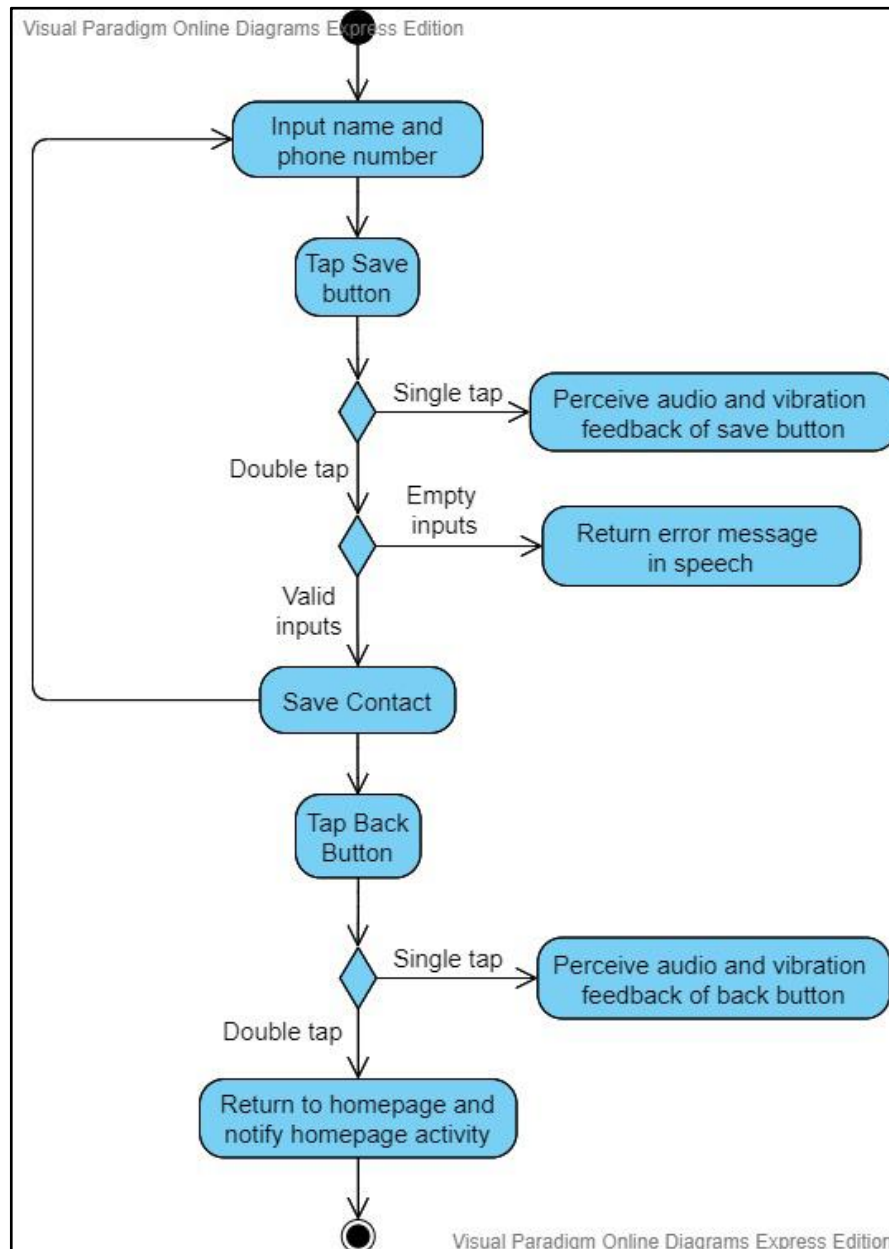


Figure 3.3.4

This activity allows users to register contact into their respective smartphones. The current flow of activity begins from inputting a set of name and phone number for further saving into the contact list of the phone. This part currently needs the

assistance of a third party to input while the further actions can be done and perceived by the users with visual impairment.

After finish inputting the fields, the users can then double tap the save button to save the name and phone number into their phones' contact list. If the name or phone number is empty, the application will notify an error in audio form let users perceive that the fields are empty. Single tapping the save button on the other hand is a method for letting users verify that the button is representing the save function that they would like to perform.

The final part of the activity would be to allow users to navigate back to the homepage of this application. Single tapping the back button will provide an audio and a vibration feedback to inform users that the button performs the function of navigating back to homepage. On the other hand, double tapping the back button will actually perform the function and a bell-like sound combined with an audio feedback saying "back" will be emitted to let users know that the function has already been performed.

3.3.5 Recognize Object Activity

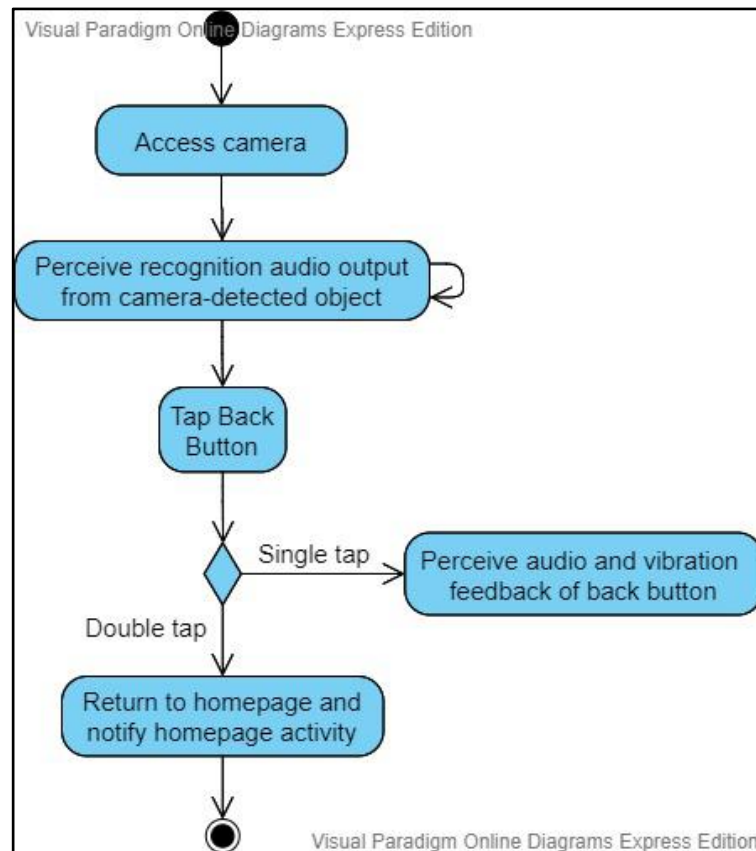


Figure 3.3.5

This activity allows users to recognize object using TensorFlow. Users are able to use this feature after they double tap the “Recognize Object” button on the homepage of the application.

Once they are navigated into this activity, the camera is then opened and accessed by the users automatically. The detection will constantly return results immediately after retrieving recognition results from the trained model. The users will be able to perceive the results of recognition via audio output returned by the application.

The final part of the activity would be to allow users to navigate back to the homepage of this application. Single tapping the back button will provide an audio and a vibration feedback to inform users that the button performs the function of navigating back to homepage. On the other hand, double tapping the back button will actually perform the function and a bell-like sound combined with an audio feedback saying “back” will be emitted to let users know that the function has already been performed.

3.3.6 Recognize Text Activity

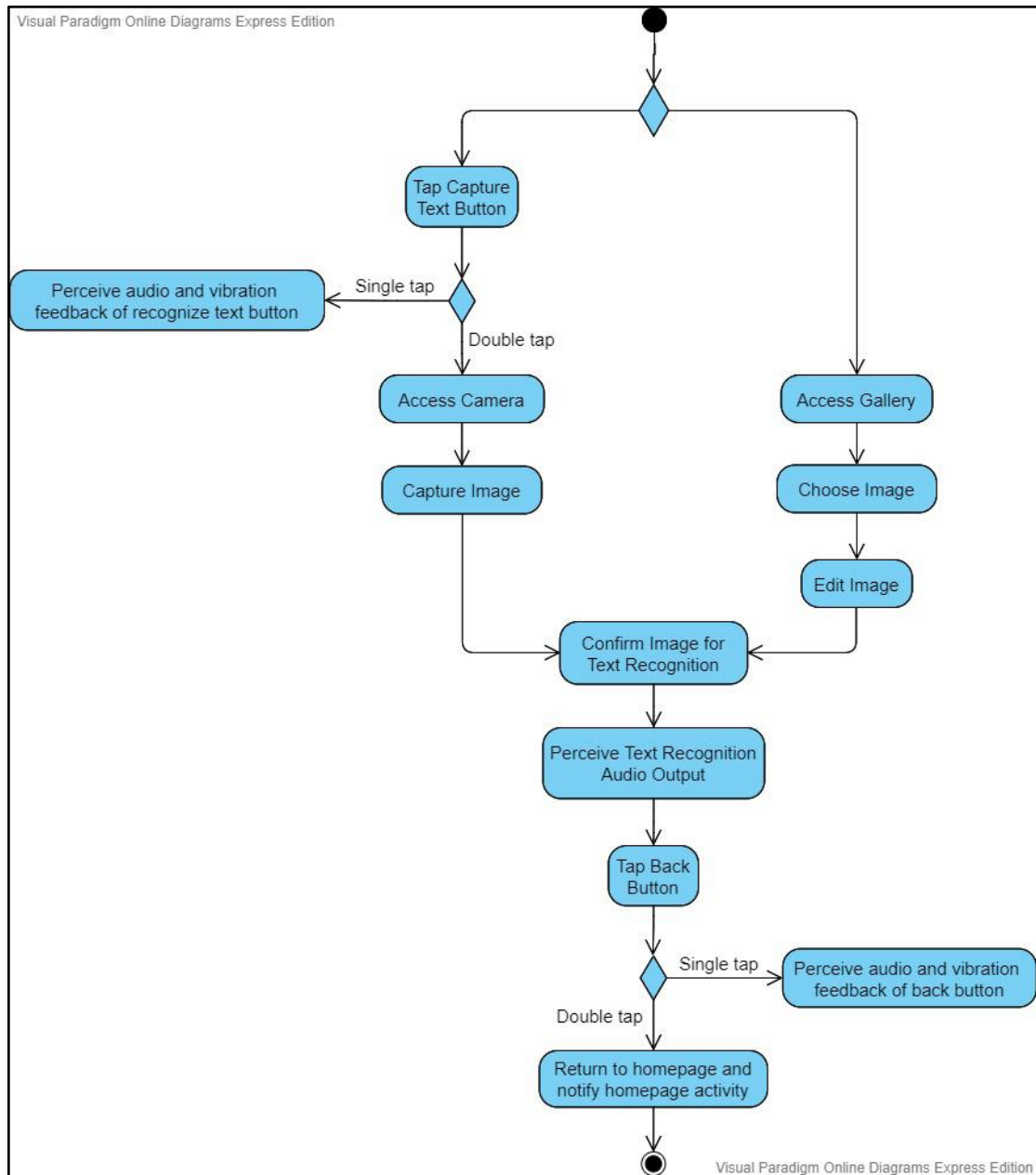


Figure 3.3.6

Recognize text activity enables users to capture documents, books, sentences, words and several more characters which can be interpreted by the text to speech synthesizer. In our application, users can either choose to synthesize the text by capturing an image from camera directly or choosing a picture from the gallery.

The flow of capturing text with the phone camera starts with users double tapping the middle image button which indicates snapping a picture with a camera. Single tapping the button will return audio and vibration feedback telling users that the

button is for capturing texts. Now, after double tapping, users will be directed to the phone camera. Then, users will capture the image and confirm the image for recognition. An audio output of the texts will be returned for the users to perceive.

Another flow would have a high chance of requiring a third party to assist the visually impaired because the current design requires a visually healthy person to navigate throughout the phone and application. First, users will access the gallery of the phone and select the desired image for text recognition. Then, they will be able to edit the images for later text recognition. The end result would also be similar like the other flow. Audio outputs of the text recognition result will be returned for the users' perception.

These two flows will lead to users choosing whether they would like to navigate back to the application homepage. Single tapping the back button will provide an audio and a vibration feedback to inform users that the button performs the function of navigating back to homepage. On the other hand, double tapping the back button will actually perform the function and a bell-like sound combined with an audio feedback saying "back" will be emitted to let users know that the function has already been performed.

3.3.7 Voice Assistant Activity

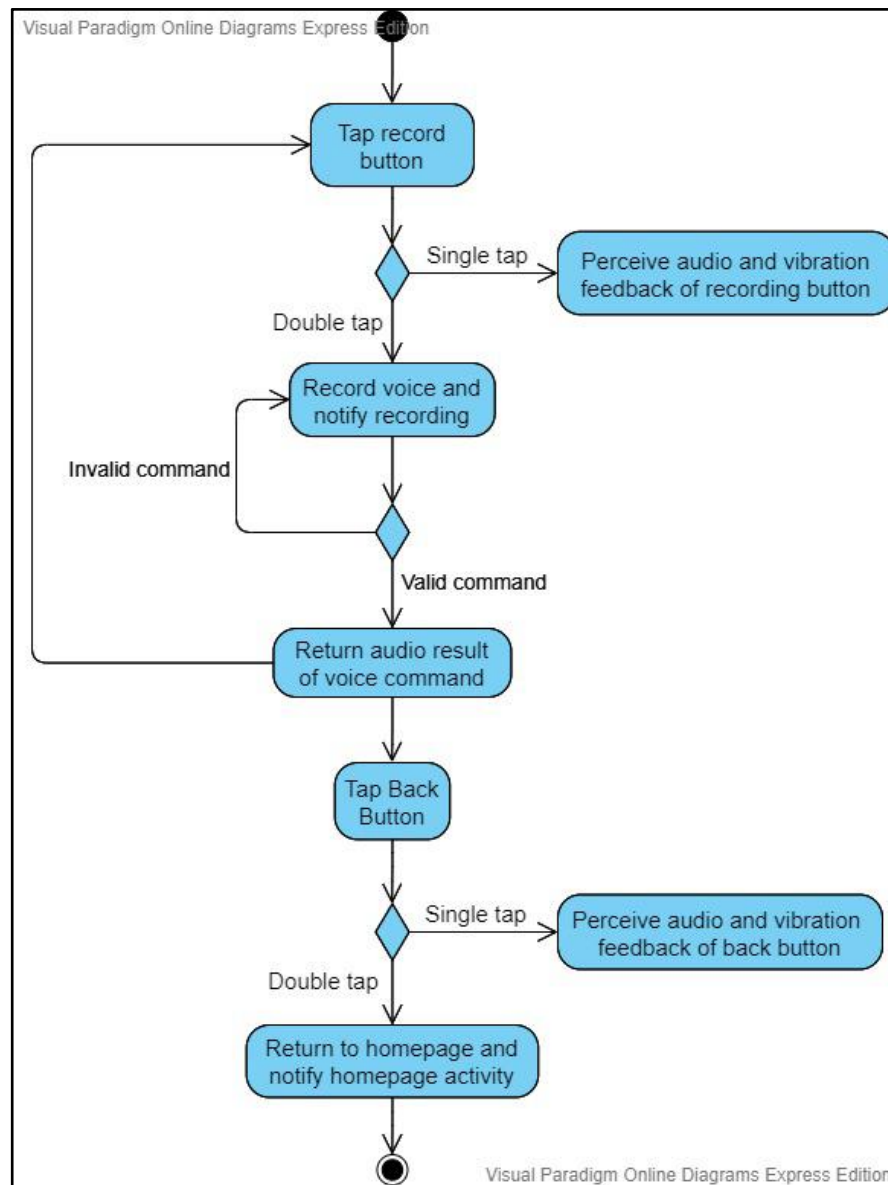


Figure 3.3.7

The main function of this activity is tapping the recording button. Tapping once would indicate its position for the users to perceive. Vibration and an audio feedback will be returned to users to tell users that they are tapping the recording button. The microphone (recording) image button has been designed large enough to be in touch for the users so they do not need to keep tapping for the right spot. Double tapping the microphone button will begin recording users' voice input. They can then voice out the command of the functions that they wish to perform. The image button has the notification sound to indicate the beginning and end of recording so users will know

when the voice input has been received. The results returned after the voice command will be in audio form too so users can perceive and understand the outcome.

3.3.8 Exit Application

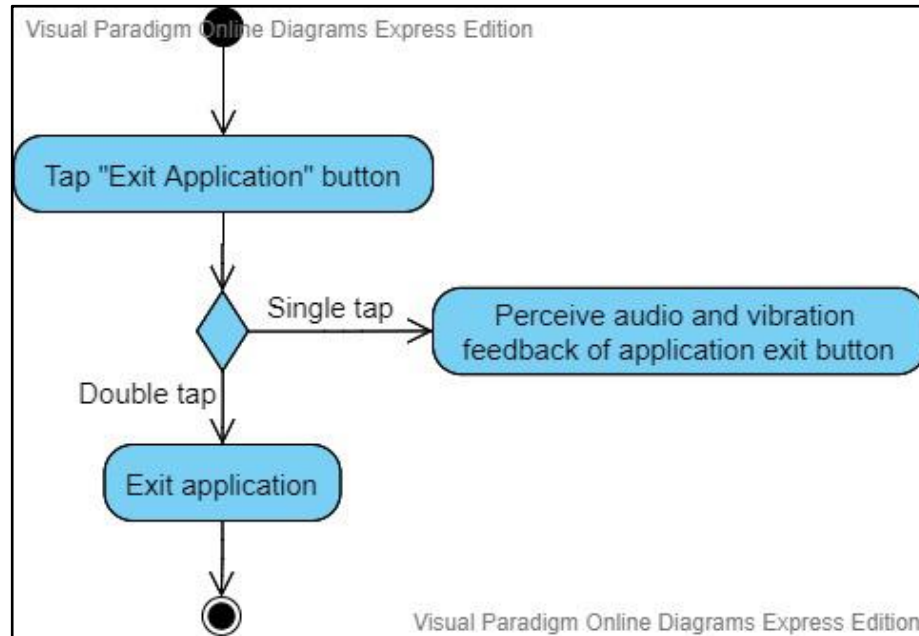


Figure 3.3.8

In order to exit the application, users will have to be in the homepage of the application. Users can look for the exit button by tapping once in order to perceive the function of buttons. The exit button should sound “Exit Application” for the users to listen. Users can then double tap the button to actually perform the function of exiting the application.

3.4 Sequence Diagram

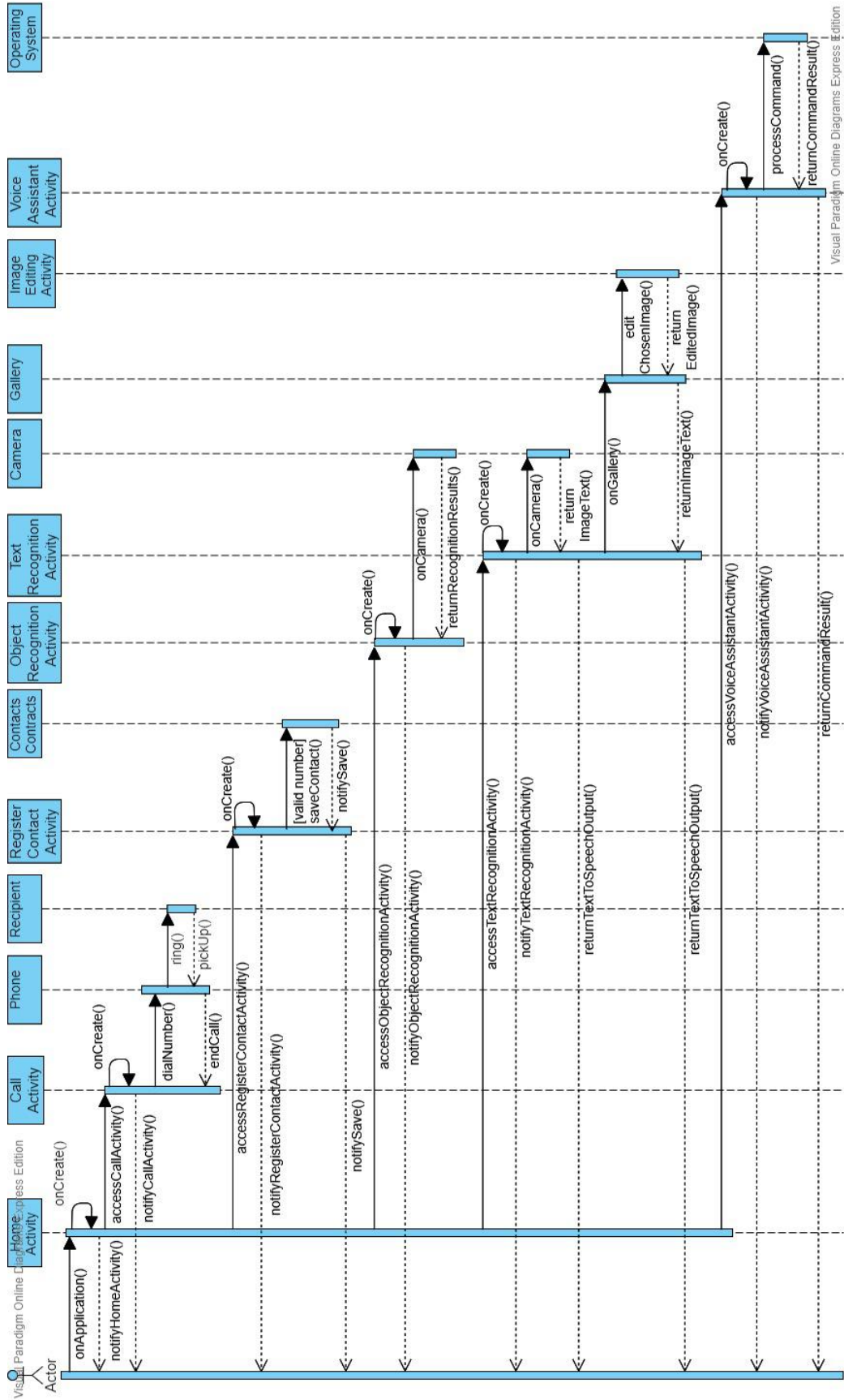


Figure 3.4

3.5 Collaboration Diagram

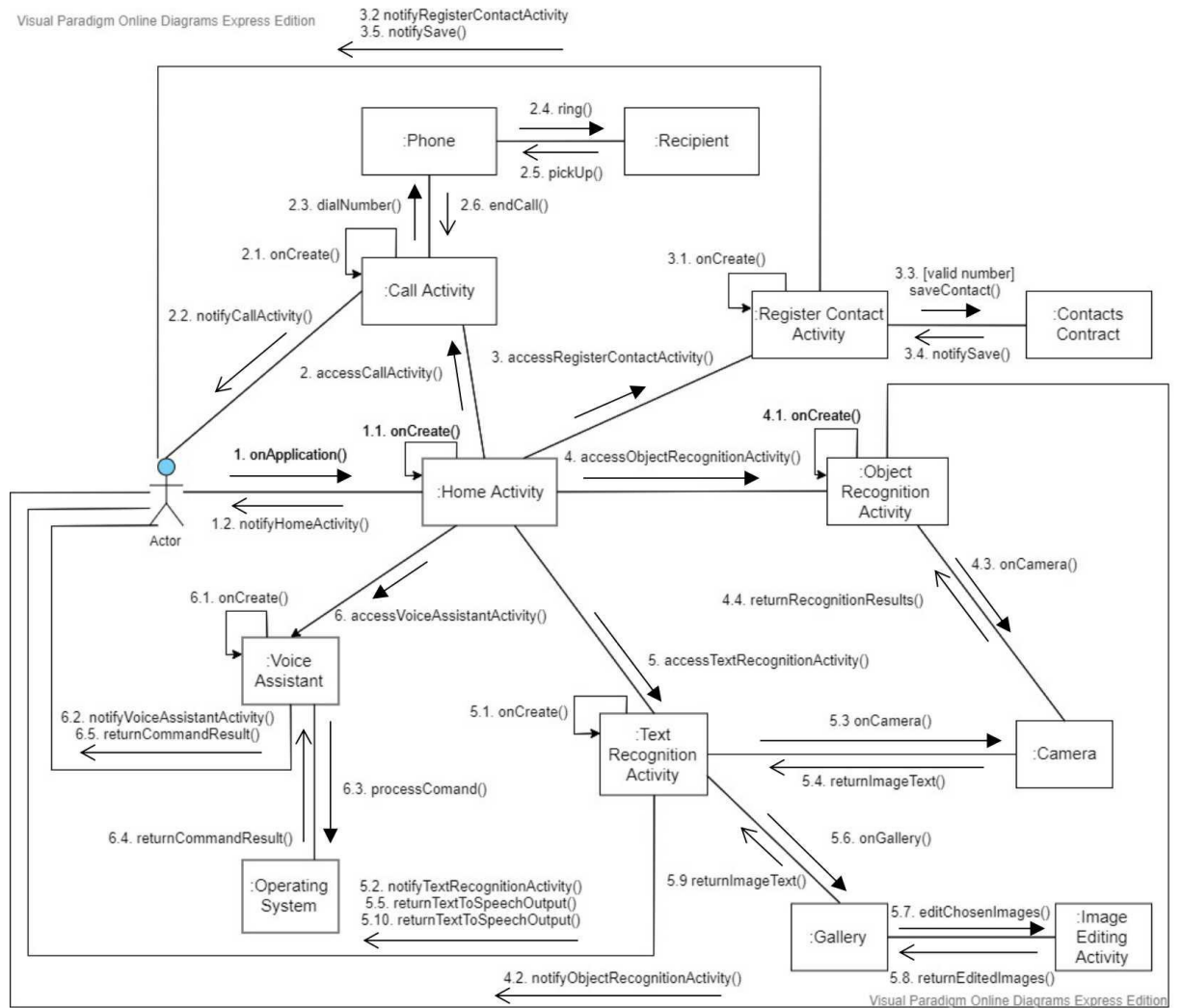


Figure 3.5

Chapter 4

Initial Prototype

Based on the literature review of various applications and studying several customer reviews on some of the applications, requirements had been gathered for this project. A proper direction had been set in ensuring the creation of a useful and efficient application which can really offer assistance to the people with visual impairment. As mentioned in chapter 1, this project's methodology is prototyping. Therefore, a lot of priority will be placed on building and releasing prototypes for users to evaluate. This chapter will describe the prototype which was built in the beginning of the entire prototyping phase.

4.1 Tools

Before describing the initial prototype built, the tools used for building the prototype and following prototypes will be specified.

4.1.1 Development Environment



Figure 4.1.1

The IDE (Integrated Development Environment) used to build this mobile application is Android Studio. It is designed specifically for Android development. The main programming language used in the development of this application's logic and features is Java. The design of the application's user interface is done by XML (Extensible Markup Language).

4.1.2 Hardware

The phone used to build this application is Huawei Nova 3i. The specifications are listed in Table 4.1.2.

Technical Specifications	Specification Details
OS	Android 9 (Pie)
Model	INE-LX2R
CPU	Octa-core (4x2.2 GHz Cortex-A73 & 4x1.7 GHz Cortex-A53)
GPU	Mali-G51 MP4
Internal Memory	4GB RAM, 128GB ROM
External Memory	microSD, up to 256 GB
Dimension	157.6 x 75.2 x 7.6 mm

Table 4.1.2

4.2 Users Characteristics

This section describes the users who will use our application.

Role	Description
User	User with visual impairment or users who have partial visual impairment.

Table 4.2

4.3 Requirements

The important required features that the initial prototype should have are being listed down before development as shown in Table 4.3.

Function ID	Function	Description	Accessible Role
F001	Notify Start Up	To notify user that the application has started up.	User
F002	Call	To allow user to perform call function.	
F003	Register Contact	To allow user to register contact into their phone.	
F004	Recognize Object	To allow user to recognize object through phone camera.	
F005	Recognize Text	To allow user to recognize text through phone camera.	
F006	Voice Assistant	To allow user to input voice command to get information.	
F007	Return audio feedback	To allow user to perceive function of every button and recognition results along with voice assistant results in audio form.	

Table 4.3

4.4 Project Structure

4.4.1 Application Logic

This section will allow this report's readers to understand the project structure created in the process of developing in Android Studio.

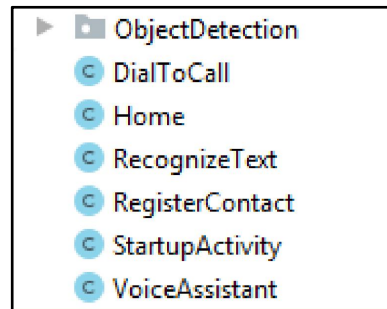


Figure 4.4.1.1

Figure 4.2 depicts the features that are developed in the project. As can be seen in the figure, the name is exactly like what they are supposed to mean.

DialToCall contains the code for the interface where users are allowed to use a specially-designed keypad to perform calling function. *Home* denotes the code implementation of the application homepage. *RecognizeText* has the text to speech synthesizer programmed in it. *RegisterContact* is developed for the ability to allow users to perform the function of contact registration. *StartupActivity* is the activity which includes a splash screen as mentioned before where a welcome screen interface is directed once the application starts up and an audio feedback will be returned to notify user that the application has started. *VoiceAssistant* contains the code of allowing users to input voice command to get results or perform functions that they desire.

The *ObjectDetection* folder contains Java codes which are designed to work with some of TensorFlow framework's features. Figure 4.2.2 shows the structure of the folder.

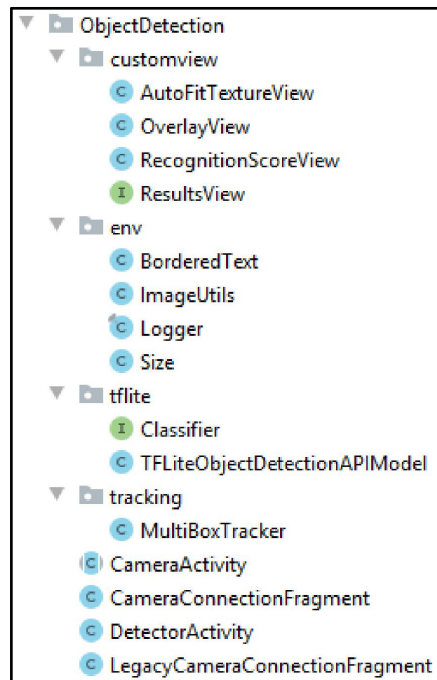


Figure 4.4.1.2

In order to simply understand the folder, the folder contents will be described briefly. The folder *customview* contains features which will be implemented into *CameraConnectionFragment* or *DetectorActivity* for the purpose of ensuring a proper capturing of objects to fit into box trackers for recognition. The *env* folder contains several parameters which can be set to fit with the trackers of the recognition model. The folder of *tflite* represents TensorFlow Lite which means the model and label will be configured into it. This folder will play a huge role in the adjustments of outcome and accuracy of the object detection. *MultiBoxTracker* in *tracking* designs the box trackers as mentioned before so they could appear on the camera after performing object recognition to indicate the object names and recognition confidence scores. The activities outside those folders like *CameraActivity* and *DetectorActivity* designs the interface which will be displayed for the users and also implements the object recognition model onto the camera where results will returned to let user know the recognition results.

4.4.2 Application User Interface Design and Audio Files

This section describes the folder structure where the user interface designs and audio files used for returning feedback are stored.

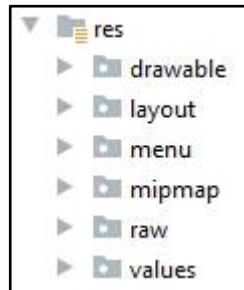


Figure 4.4.2

The folders used for the user interfaces designs are *drawable*, *layout*, *menu*, *mipmap* and *value*. On the other hand, the audio files are stored in *raw*. The folders *drawable*, *values* and *mipmap* basically stores parameters which will be used in the main design of the user interface where all the files are located in *layout* or *menu*.

4.5 Prototyping

This section will explain the way that the required activities are implemented in the development of codes.

4.5.1 Start Up



Figure 4.5.1

Android's *AppCompatActivity* library has a function known as *onCreate* where every activity will be created according to the designed layout which will be linked in the function's settings. All of the activities which will be displayed as a user interface on the mobile application will have to use that function. The function will set a content view linking to the startup layout designed in the *layout* folder. The design for the startup interface for this prototype will look as displayed in Figure 4.5.1.

As this is only a welcome (startup) screen to notify users have successfully started the application, this screen will be designed to have a timeout of 2.5 seconds. After 2.5 seconds, this screen will start the homepage activity where users will be able to select actions they would like to perform. Besides, during this startup activity,

Android library's *MediaPlayer* will play an audio welcoming users to the application to notify those visually impaired.

4.5.2 Homepage

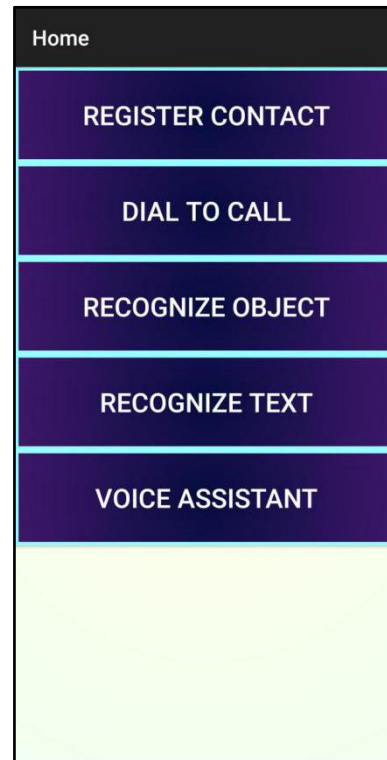


Figure 4.5.2

The above figure displays the user interface of the application's homepage after the first prototype has been developed. Every button allows users to access the respective functions as labeled. For this prototype, tapping once on the button will sound the button's function and direct users into the selected function.

The activity and interface as mentioned before is started by the *onCreate* function by an Android library known as *AppCompatActivity*. Tapping the buttons will start the respective activities by calling those activity classes shown in Figure 4.4.1.1. The audio feedback returned to the user is also handled by *MediaPlayer* library of Android. The *MediaPlayer* value or item is declared in the *onCreate* function and the audio will be triggered or started (coding logic) when the screen detects users' touch on the button.

The file of the homepage interface design is located under *layout* directory and named *activity_home.xml*. In this phase, the priority is not places on the user interface so a random design will be made. One of the designs from this interface is the usage of a value created in *drawable* to be implemented onto the background of every button to make it look more pleasant. As you can see from Figure 4.5.2, the border of every button is light blue and the background is dark blue, a minor contrast that can be seen but buttons cannot be distinguished easily for those who are partially blind. The fonts are designed to be white to contradict the colour of dark blue in the background of the button to make the labels clearer. The background of this interface (behind the button) is randomly set to be in light blue colour.

4.5.3 Dial to Call



Figure 4.5.3.1

This is the activity which allows users to call directly from this application. The keypads and accessibility are designed specifically to fit the visually impaired users' requirements so these users do not need to go through a conventional way to perform a call which is very difficult from their perspectives.

As you can see from the above figure, users are able to dial the numbers just like the usual numbers which exist in a normal keypad. However, there is an extra delete button which can help users to perform backspace action without the need to find the backspace button from the phone keyboard. Tapping on every button from the keypad will return audio feedback telling users their respective representations so users are able to interpret themselves. The way the audio return feedback is also by using the library of Android known as *MediaPlayer*. Every inputted number will be appended into a string which will be displayed in the text field above the keypad. Once the number is confirmed, users are able call by tapping the call button. Users should also take note that a logic has been programmed to prevent users from calling an empty phone number because it is not possible. Therefore, an error message in audio form will be returned for the users to listen if the field of phone number is empty.

This part of the application will need to request permission to call before calling for the very first time to allow Android phones users to call from the application. Therefore, the logic in requesting permission for calling from the application if it has never been requested before is important to be included in the program. The permission has to also be declared in the manifest file of the project, namely *AndroidManifest.xml* under the directory called *manifest*. The permission declaration is shown in Figure 4.5.3.2.

```
<!--Phone calling permission will be required-->  
<uses-permission android:name="android.permission.CALL_PHONE" />
```

Figure 4.5.3.2

The call can be performed by coding an intent of Android known as *ACTION_CALL*. The data (phone number) will then be formatted and parsed into a new URI object and then set to be ready for call. Once users tap the call button, the calling activity will then be started and the call is performed.

4.5.4 Register Contact

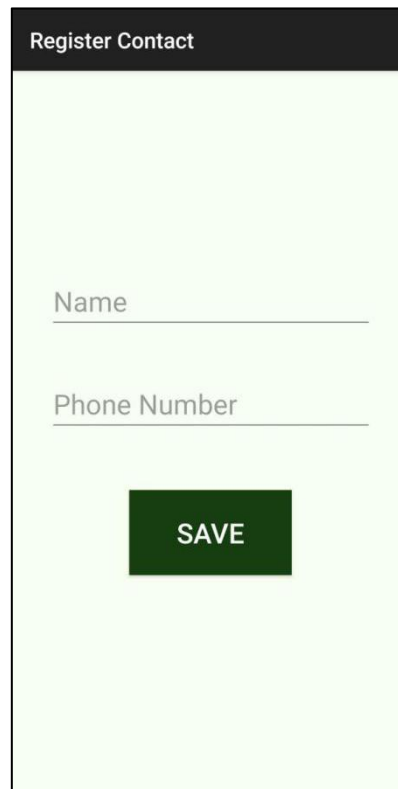
The image shows a mobile application screen titled "Register Contact". The screen has a light green background. At the top, there is a dark grey header with the text "Register Contact" in white. Below the header, there are two input fields. The first is labeled "Name" and the second is labeled "Phone Number". Both labels are in a light grey font and are positioned above a horizontal line representing the input field. Below these two fields, there is a dark green rectangular button with the word "SAVE" written in white capital letters.

Figure 4.5.4.1

This activity uses the Android classes of *ContactsContract* and *ContentProviderOperation* to perform a direct contact registration from this application into the phone's contact list.

ContentProviderOperation will be the party which inserts the inputted name and phone number into the lists of contact handled by *ContactsContract*. While inserting the new contact, a new contact ID will be created to let *ContactsContract* distinguish the new contact with the others. The type of input data are specified so the values of name and phone number will be stored in its correct column. The name input shown in Figure 4.5.4 will be stored as a display name of the contact while the phone number will be stored as a phone number of the contact.

If the input of name and phone number are not inputted, the application will return an error message in audio form notifying users that the name and phone number must not be empty. The "Save" button will sound when pressed and the way it is designed is the same as the past described buttons.

Performing this activity also require the requests of permission in reading and writing contact of Android phones. The manifest written in the project is shown in the figure below.

```
<!--Contact accessing permission will be required-->  
<uses-permission android:name="android.permission.WRITE_CONTACTS" />  
<uses-permission android:name="android.permission.READ_CONTACTS" />
```

Figure 4.5.4.2

4.5.5 Recognize Object

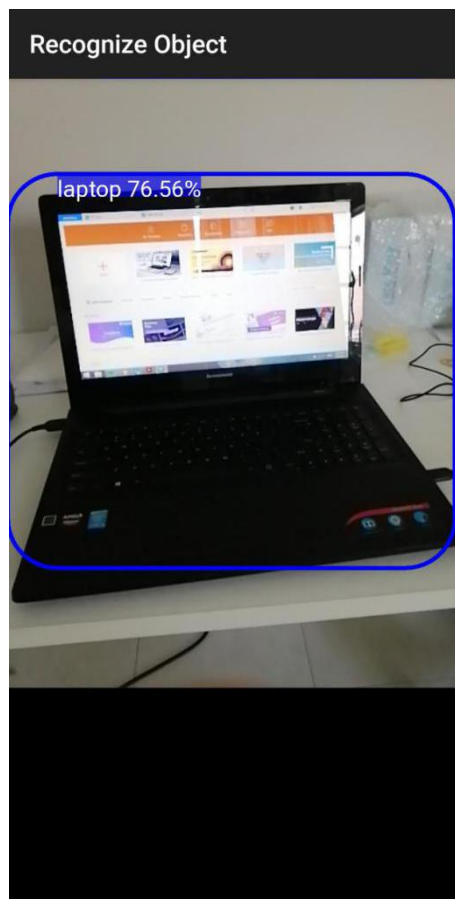


Figure 4.5.5.1

This function comprises the creation of various classes as described before in Section 4.4.1. Now, the way to implement TensorFlow Lite technique will be described in more detail.

TensorFlow Lite uses a *.tflite* model file to perform object recognition. This file type is converted from a model trained using Python. The conversion will be made via a Python API. After converting, the model is deployed to an Android device and ran locally using an interpreter of TensorFlow Lite. This file will be implemented through the TensorFlow Lite API designed using Java into the *DetectorActivity*. The *.tflite* file is located under the directory named *assets*.

In order to retrieve recurring recognition results for the users to perceive, the collaboration between *CameraActivity* and *DetectorActivity* is very important. Two of them are needed for extraction of real-time images for immediate recognitions.

For first time access into this activity, permission to use camera will be requested for the users. The application logic is programmed in the *CameraActivity* class. The manifest file for requesting the camera permission is shown below.

```
<!--Camera and storage permission will be required-->  
<uses-permission android:name="android.permission.CAMERA" />  
<uses-feature android:name="android.hardware.camera" />  
<uses-feature android:name="android.hardware.camera.autofocus" />
```

Figure 4.5.5.2

The purposes of other classes has been explained in section 4.4.1 so kindly refer to that.

4.5.6 Recognize Text

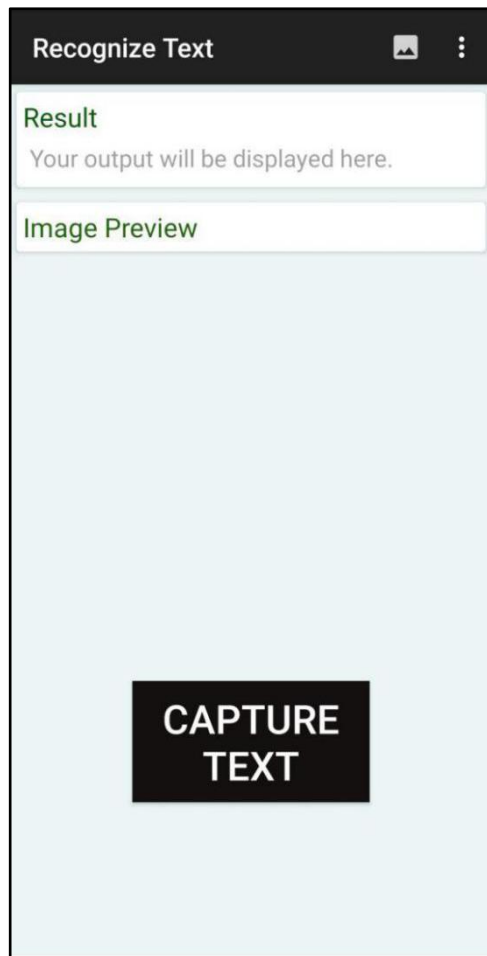


Figure 4.5.6.1

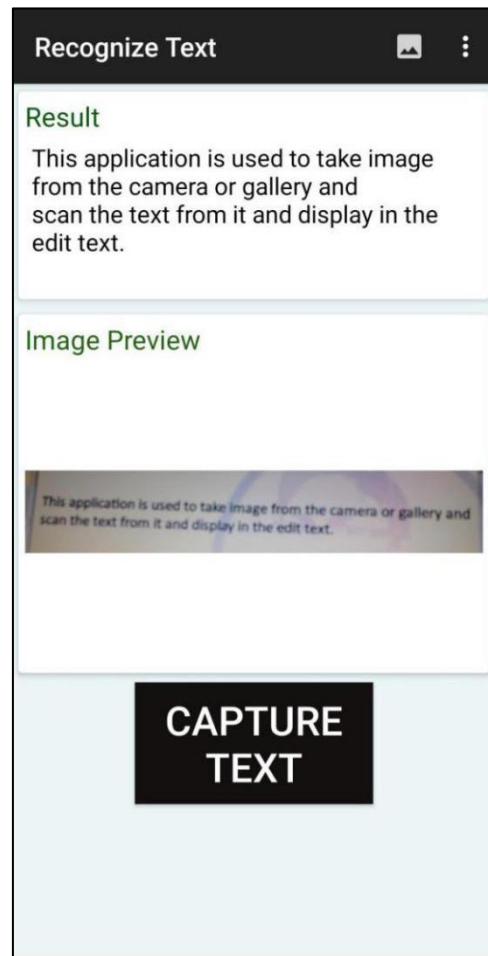


Figure 4.5.6.2

We will start off describing the interface by mentioning that clicking “Recognize Button” in the homepage will direct users to interface displayed in Figure 4.5.6.1. Now, when users are in this page, they can choose either to capture text or access gallery. In order to access the camera to capture text, users have to just simply click “Capture Text” which can also sound audio when clicked to notify the user. Then, users will access the camera and take an image of the text and an output will be returned just like the display in Figure 4.5.6.2.

On the other hand, users have to click the scene icon on the top right of the page to access gallery. Once chosen a picture from the gallery, users are able to edit the image. The image editing interface is portrayed in Figure 4.5.6.3. When it is done edited, the output will also be returned as similar as in Figure 4.5.6.2. However, these steps obviously can only be handled by a sighted user. Sighted users are able to help

the visually impaired to choose images of perhaps a book or news to be read aloud by the application. It is able to save time of sighted users from utilizing time to always read for the visually impaired. Sometimes, it is better this way.

This activity is mostly handed by text to speech which converts the extracted text(s) from the image into voice output.

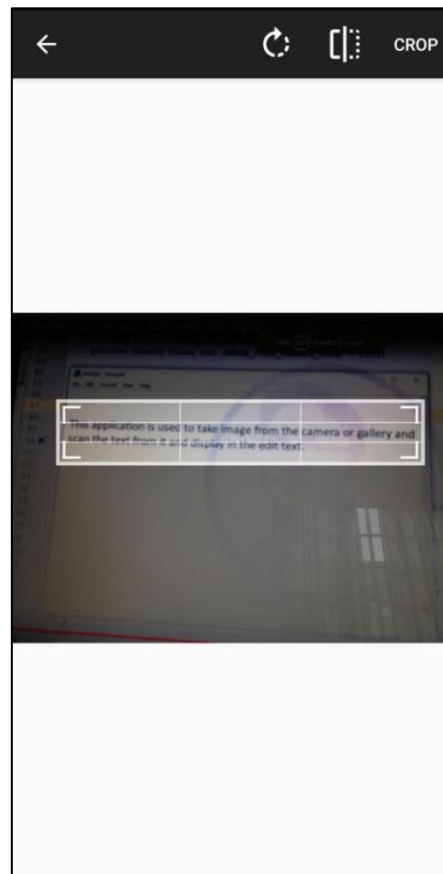


Figure 4.5.6.3

A manifest permission is also needed for the purpose of accessing storage to retrieve images from gallery. Another manifest permission need is the use of camera to directly capture text from images. The permission is written as shown below.

```
<!--Camera and storage permission will be required-->
<uses-permission android:name="android.permission.CAMERA" />
<uses-feature android:name="android.hardware.camera" />
<uses-feature android:name="android.hardware.camera.autofocus" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Figure 4.5.6.4

4.5.7 Voice Assistant

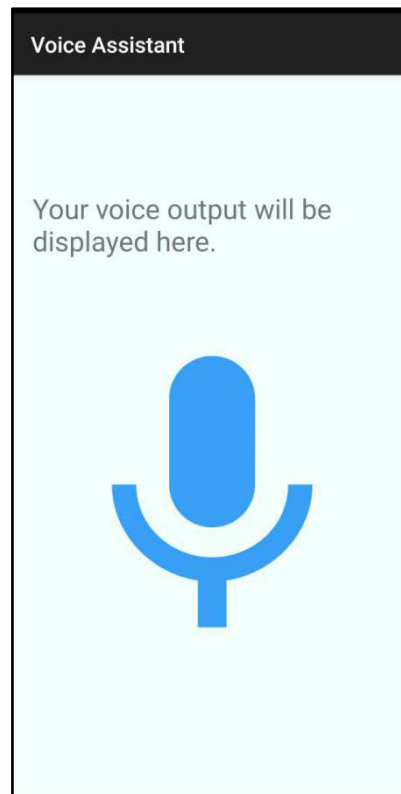


Figure 4.5.7.1

This activity is known as a Voice Assistant. Clicking the microphone image button will simply starts recording the users' voice input. Android's speech recognizer will play a part in detecting voice inputs. When the microphone button is pressed, it has a sound signifying its beginning of recording. The end of receiving the voice input will also sound.

The output of the voice command will be displayed on screen and also spoken aloud so users can perceive it. Android's text to speech synthesizer is used to allow those displayed output texts to be spoken aloud. So far, users can only request for the time and the phone's battery status.

A permission of recording audio is needed to be written in the manifest file of the project though. The code can be seen as shown below.

```
<!--Audio recording permission will be required-->  
<uses-permission android:name="android.permission.RECORD_AUDIO" />
```

Figure 4.5.7.2

Chapter 5

Customer Evaluation

5.1 Evaluation Method

This evaluation is conducted online due to the ongoing pandemic. The sample size used for this evaluation is 40 respondents. The respondents vary from visually healthy and vice versa. Those who are visually healthy are required to close their eyes while testing out the mobile application prototype, whereas the respondents who have weak eyesight are required to take off their spectacles or any other visual aid while trying out this mobile application. However, those who are visually unhealthy but still can perceive through sight well without visual aid are required to close their eyes while conducting this evaluation.

Moreover, the target respondents chosen are only those who own an Android mobile device because the application prototype is currently only designed for Android users. An *apk* file will be shared via Google Drive link for respondents to download and install into their local devices.

Once the respondents are ready, guides will be provided for the first time they use the application prototype. When they are familiar and ready to use the application, they are required to try navigating throughout the application and try out those recognition and other application functions themselves. 15 minutes will be provided for respondents to complete the process.

After finish testing the mobile application, the respondents are then required to fill in a feedback form via Google Form. The form will attempt to gain feedback from respondents regarding their preferences and the optimal application features.

5.2 Evaluation Outcome

The following subsections will describe the outcome of the feedback based on the response by all 40 respondents after testing out the prototype of the mobile application.

5.2.1 What would be the best feedback a mobile application could allow visually impaired users to perceive?

The pie chart depicting the results for the answers to this question is located in Appendix A-1. This question is asked based on their preferences and judgement on the accessibility throughout the entire application prototype.

Three choices of answer are provided for selection. They are audio, vibration or both audio and vibration. 87.5% of the people have chosen to have the integration of audio and vibration to be built into the application. Meanwhile, only 12.5% of the people are satisfied with the prototype's only audio feedback. Vibration took in zero responses.

As can be seen from the results, the prototype has to be modified to suit the users' preferences. Audio and vibration must be used together to provide feedback for users to perceive.

5.2.2 Do you think navigation between pages like a "BACK" button or "EXIT" button is needed for a mobile application built for the visually impaired?

The pie chart depicting the results for the answers to this question is located in Appendix A-2. This question requests for the respondents' opinions on the necessity of implementing a navigation button into the application for the visually impaired users who will actually utilize the application.

Majority (65%) of them feel that it is necessary for these navigation buttons to exist in the application. 12.5% of the respondents answered maybe so this segment of data cannot be taken into consideration while the rest of the respondents (22.5%) disagreed with the question asked.

With the data collected, it is clear that navigation throughout the application has to be prioritized too during prototype development.

5.2.3 Is it important for a visually impaired person to be able to open the mobile application on his own?

The pie chart depicting the results for the answers to this question is located in Appendix A-3. This part asks respondents on the importance regarding the ability of a visually impaired person to start up the application by himself or herself.

35 respondents which take up 87.5% of the sample size answered yes to the importance. 3 respondents answered maybe while the leftover 2 respondents think that it is not important.

Based on the outcome, it is very clear that the application should be able to be conveniently started by the visually impaired users. Therefore, this requirement should be listed as part of the development in later prototype developments.

5.2.4 If the application is also built for the partially impaired (those who can still see but vision is very blurry), which user interface do you think is more suitable for them?

The pie chart depicting the results for the answers to this question is located in Appendix A-3.

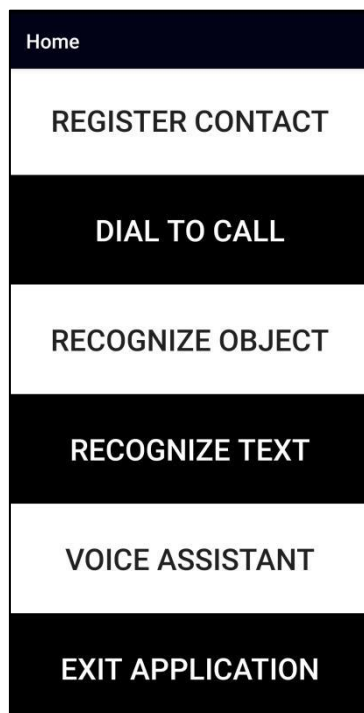


Figure 5.2.4.1

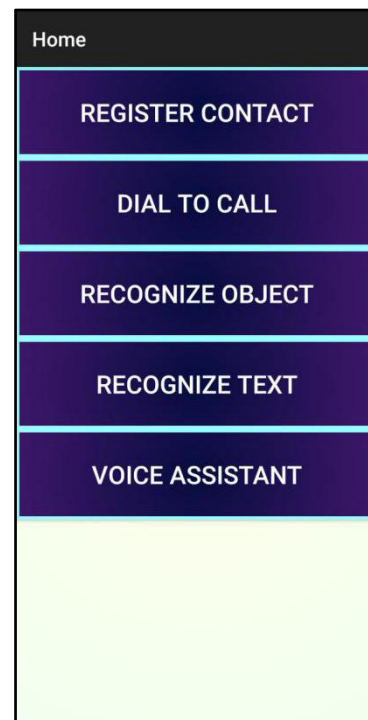


Figure 5.2.4.2

Figure 5.2.4.1 is a screenshot of an interface where color contrast is being prioritized while designing. On the other hand, Figure 5.2.4.2 is a picture where the beauty of interface is being prioritized while developing. Both are screenshots of the homepage taken from the application for the purpose of displaying for respondents before they answer the question. The question asks the for the selection of interface which can suit the partially visually impaired community who uses this application.

CHAPTER 5 CUSTOMER EVALUATION

29 (72.5%) respondents chose the interface with much more color contrast while the rest have chosen the beautiful interface.

Results have shown that it is more important to create an interface where there is much more distinction between colours so users can properly select the items (buttons) which they desire.

Chapter 6

Prototype Refinement

This chapter will disclose and explain the newly implemented application designs and logic. This new development has been done based on the evaluation provided by the customers (alpha users).

6.1 New Requirements

Function ID	Function	Description	Accessible Role
F101	Return audio and vibration feedback	To allow users to perceive and understand buttons for every tap even better.	User (Visually Impaired)
F102	Add navigation buttons in all necessary activities	To allow user to navigate throughout the application conveniently and suitably according to their perceptions.	
F103	Start up application running in background	To allow user to access application automatically once the phone is unlocked.	
F104	Design interfaces to have distinguishable contrast between colours	To allow partially visually impaired users to perceive the sight of the user interfaces without trouble.	
F105	Return audio feedback of activities being resumed	If user went idle after using activity halfway, user is able to unlock phone and resume activity and not restart application.	

Table 6.1

6.2 Screens of All User Interfaces in Refined Prototype

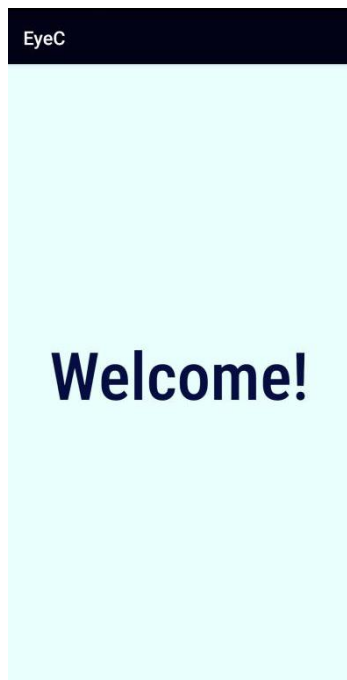


Figure 6.2.1.1

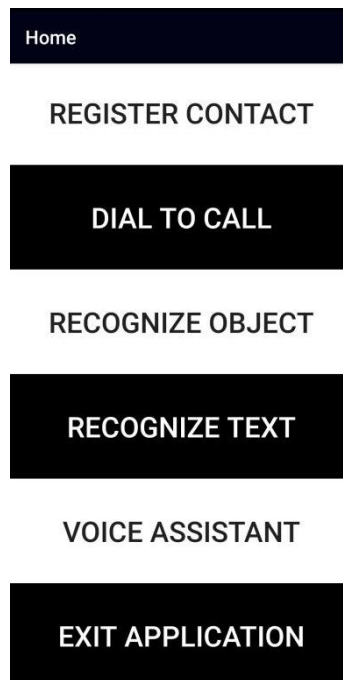


Figure 6.2.1.2

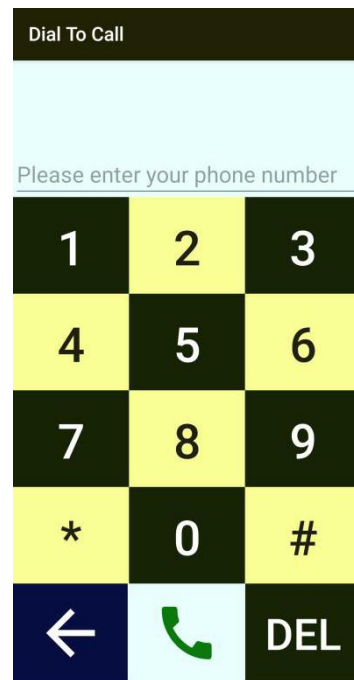


Figure 6.2.1.3

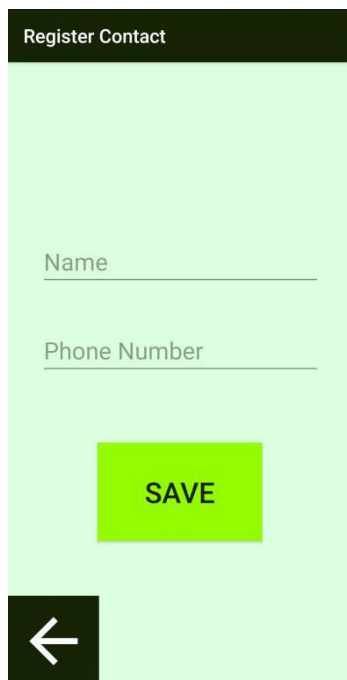


Figure 6.2.1.4

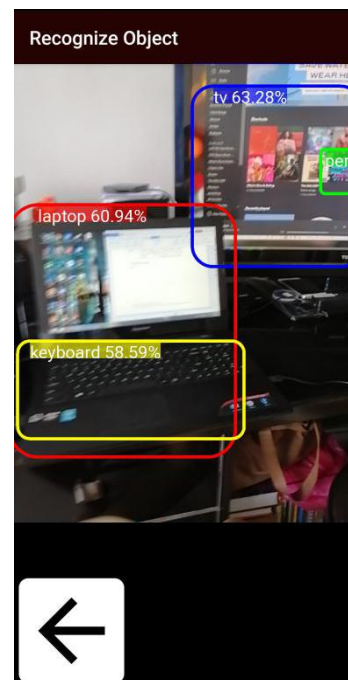


Figure 6.2.1.5

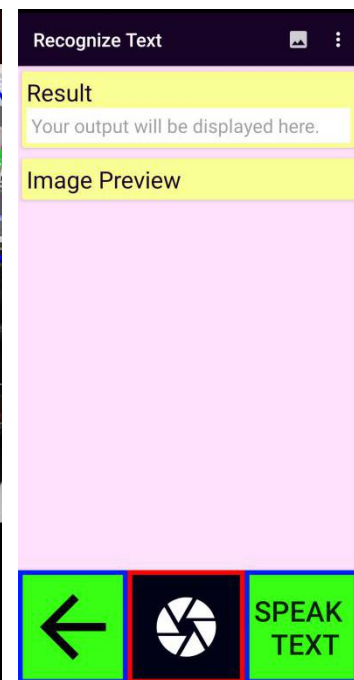


Figure 6.2.1.6

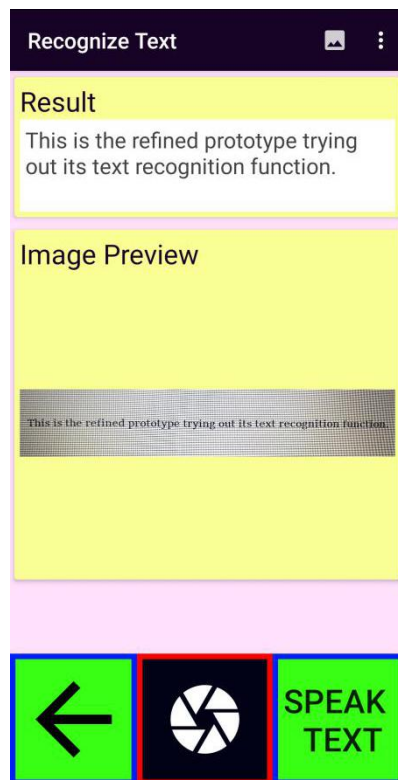


Figure 6.2.1.7

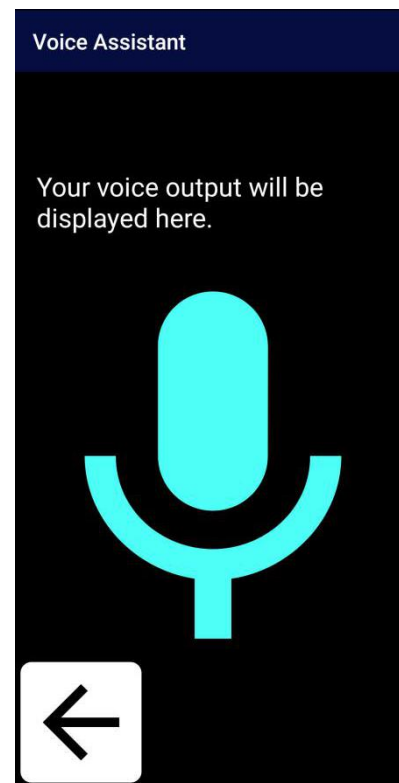


Figure 6.2.1.8

6.3 New Features of Refined Prototype

6.3.1 Integration of Audio and Vibration Feedback

Every single button in this new application prototype will not only return audio feedback for users to perceive, it also returns vibration feedback. How does it actually work since both types of feedback are utilized?

Here is the explanation. In order to first understand a certain button's function, users will have to tap once onto the button. Once tapped, the application will return a vibration indicating that user still has not performed any action and an audio feedback specifying the button's function. Users will have to double tap the button in order to actually carry out the function or access a certain feature. Double tapping will also sound an audio speaking the function of the button and a bell-like audio to notify users have already performed the action.

The coding implementation of this feature requires the utilization of an Android library called *GestureDetector* where it has a built-in function detecting single tap and double on screen and hence inserting the desired action into the functions respectively. The vibration feedback on the other hand uses an Android OS library to get a certain phone's vibration service to perform vibration when triggered. The vibration duration

and settings can be configured external of the *onCreate* function coding-wise. The function containing the settings can then be called to be triggered when vibration service is called. The current configuration of vibration service is suitable for Android 8.0 and above only so users have to be aware of this.

6.3.2 Inclusion of Navigation Button on Every Necessary Activities

After the studying the results of the evaluation conducted from Section 5.2.2, it is understood it is important for navigation buttons to be designed on the activity pages in order to return audio feedback to guide them to wherever page they desire to be direct to.

The buttons function similarly as explained in Section 6.3.1. The design of back buttons can be observed in some pages displayed in Section 6.2. Single tapping the back buttons will notify users that it is for the function of navigating back, which all direct users back to homepage. This is because most of our features only contains one page after homepage and not more. Double tapping would actually navigate users back to the previous page. Users also do not need to worry if they are in the homepage because the application will also return audio feedback speaking the page they are in. All the back buttons currently in this application direct users back to homepage so a bell-like sound and the homepage audio should play once users are directed there.

Users are also able to exit the application. The function is also similar like the other buttons. Single tap would return audio and vibration output while double tapping would actually perform the function which is exiting the application. When this application is exited, the process is not entirely terminated as there is a background service running which bring us to our next feature in Section 6.3.3.

6.3.3 Unlocking phone starts up application

This feature is an important feature in assisting the visually impaired user begin using the application. None of the other features matter if the visually impaired users cannot even access the application. Now, let us take a look at a clearer picture of the feature.

If users are accessing the application for the very first time, assistance is required from a third party to start up the application. Once the application has started up, a background service will start running. Now, users can exit the application via the exit

button. After exiting, this background service will not be terminated but continue running in the background. Next, users can try locking and unlocking their phone screen. The action of unlocking the phone screen will automatically turn on the application. This is how visually impaired users are able to start up the application by themselves. There is no need for them to look for the application shortcut to actually tap it which is too difficult for this category of users.

However, a logic is programmed into this feature because there exists a bug which is very troublesome for users. Notice that now the application will startup every time users unlock their phone. If there is no logic in this feature, the application will restart when users unlock the phone even if they lock their screen in the middle of using the application. This is not a rational function an application should have. Therefore, the background service has been programmed to only start the application when the application is not on pause.

If a third party assisting the visually impaired notice a high battery consumption on a certain phone due to this application. They can simply remove the application from the task manager to terminate the application together with its background service. If they wish to help the visually impaired use the application again, simply just turn it on again to trigger the background service once again.

The code implementation of this feature involves classes of Android's *Service* and *BroadcastReceiver*. The *BroadcastReceiver* is responsible in receiving the trigger of phone unlock in order to start up the application while *Service* is responsible in ensuring the application has a service running in the background to detect the action of *BroadcastReceiver*. This *Service* class is also designed to display a notification on the notification panel of the phones so the third party users who are visually healthy are able to understand the application condition and its battery consumption if they need to.

6.3.4 Design of User Interface

In the initial prototyping phase, there was not much priority placed on designing user interface. However, it is discovered over the period that this application can also be built for partially blind people so the development of refinement will take this feature into consideration.

In the refined prototype, the buttons have been designed to contradict each others' and background's color. That way, partially blind people can still distinguish items displayed on the screen. If there are anymore changes that need to be made to the design of the application to fulfill the users' requirements, there will definitely be no hesitation in performing the modification.

6.4 Post-Refinement of Prototype

6.4.1 Development

Now, the leftover developments are the assurance of existing functioning core features. The important one left is returning object recognition audio output for users to perceive.

As explained before, the recognition algorithm is done by Tensorflow Lite so the technique left is to implement Android's Text To Speech synthesizer at a suitable location where the labels read from recognition can be spoken aloud for the users to listen.

Besides that, flashlight toggle will also be implemented into Voice Assistant. It allows users to read objects or texts in dark environments. Users only have to simply command to "turn on flashlight" to trigger the phone's flashlight to switched on and vice versa. This implementation involves using an Android's library named *CameraManager*. This library handles the flashlight for toggling where there's a function known as *setTorchMode*. Setting one of the function's parameters called *enabled* to true will turn on the flashlight while setting it to false will do the opposite.

6.4.2 Testing

All the features and accessibility of the mobile application are fully tested once again to ensure there is no bugs or features which do not meet the requirements. Then, a further test or better called as evaluation is conducted through the same method as mentioned before in Section 5.1.

A question will be published to the 40 respondents who have previously evaluated the prototype. The reason they are chosen back again is because of their familiarity on the system. Therefore, they would clearly understand what is lacking from the application system. The question is published in Google form. It only asks a general question with a display of an updated user interface of the application's

homepage. Users are requested to give their opinion on the sufficiency of the features that exist in the application. The question is can be seen displayed in Figure 6.4.2.

Are these buttons (or features) sufficient for the people with visual impairment?
If no, what is your suggestion for the additional feature? *

Home

REGISTER CONTACT

DIAL TO CALL

RECOGNIZE OBJECT

RECOGNIZE TEXT

VOICE ASSISTANT

EXIT APPLICATION

Your answer

Figure 6.4.2

Let's take a look at the outcome of the response. 29 respondents answered that the application's features are sufficient for the community with visual impairment. However, this might be enough for the meantime only. Therefore, we have to also study the critiques in order to further improve the application.

Some of the important critiques and judgement have been filtered and analyzed. There are some critiques speaking about the interface, specifically the button. One of them mentioned that the button fonts could be bigger. Another mentioned that a more vibrant colour is needed to distinguish the buttons and that the buttons do not look very much like a button. This will be an aspect which will be looked into during the next phase.

Besides that, 3 of the respondents also mentioned about building features that functions like Siri. Currently, there is one which is Voice Assistant. However, the

function that could be done by Voice Assistant is limited. Perhaps in the next phase, the Voice Assistant can be programmed to call features like Siri (Google Assistant for Android Users) locally.

6.4.3 Maintenance

Now, the maintenance phase will focus on fixing existing bugs and possibly adding features which users require. The comments from a question published during testing phase will be taken into consideration for further developments or maintenance.

Chapter 7

Conclusion

7.1 Personal Insights

It has been a great journey developing this application. A lot of knowledge have been gained and a lot of problems have been solved.

Java and Android would be the most insightful knowledge that have been gained and understood. Design flow and prototypes development are definitely great technological lessons that have been learnt throughout the entire development process.

One of the problems that hinder the development is understanding the TensorFlow Lite folder. Online documentations need to be studied so the files are understandable and further actions can be done with it such as implementing the text to speech synthesizer along with the framework.

Another problem faced would be to understand the functions of *BroadcastReceiver* and *Service* classes. Android released its own documentation for developers to study. However, technical terms used in the document made it difficult to simply understand. Without understanding the functions, code implementations cannot be done. Therefore, research continued to look for forums like StackOverflow where the most approved comments or answers are being studied in order to develop this project using them.

7.2 Project Review

This project can be concluded by its successful achievement of all its initially stated objectives and scopes. The end mobile application product is able to efficiently solve the problems that was initially written down.

Firstly, the project has solved the problem of users facing trouble to access some settings or phone applications conventionally. This mobile application is built to provide a totally new and comfortable environment to the community with visual impairment. Besides that, the recognition of object function is also working very well. There is no redundancy of object while recognizing things except for objects which are not trained in the model. Visually impaired users can also communicate by calling

via this application. Accessing to information like time and battery status also is not a huge problem. The ability for this application to be started up once phone is unlocked has clearly also solved the problem of the non-necessity for conventional accessibility method that is super troublesome for the users.

7.3 Future Work

This application obviously still has room for improvement. The first aspect which could be added in the future is the removal of the bottom navigation bar of Android devices when the application has started up. Every Android devices has a default settings which displays the navigation keys every time they are browsing throughout the phone and other applications. The navigation bar usually contain a task manager button, a home button and a back button. The removal of this bar can reduce the chance of the users accidentally pressing them and navigating to pages leaving the mobile application, where they would be lost as smartphones' screens has very sensitive touches.

Another feature which could be added in performing call function via Voice Assistant activity. The idea of the flow would be to detect voice input such as "Call Emily". The speech would then be converted to text where the name "Emily" would be appended to match with the string in the contact list. After matching, the call can then be performed. However, if the name is not found in the contact, an error in audio form should be returned for the users to perceive. Besides implanting this feature in the Voice Assistant activity, many more other features can be created. Voice Assistant could be developed to play a huge part in this application because it solely relies on speaking and listening. Both these actions are strong perceptions of the visually impaired users.

Bibliography

- Koestler, F. A. (1976) *The unseen minority: A social history of blindness in the United States*, New York, NY: David McKay.
- Arati, K., Sayali, A., Sushanta, D., & Harshata, A. (2015) ‘Object recognition in mobile phone application for visually impaired users’, *Journal of Computer Engineering*, vol.1, no.3, pp. 30–33.
- Dobosz, K. (2017) ‘Designing Mobile Applications For Visually Impaired People’, *Visually Impaired: Assistive Technologies, Challenges and Coping Strategies*, pp.103-126
- Jabnoun, H., Benzarti, F. and Amiri, H. (2015), ‘Object detection and identification for blind people in video scene’, *2015 15th International Conference on Intelligent Systems Design and Applications (ISDA)*, Marrakech, Morocco, 14-16 December.
- Abadi et al. (2015), ‘TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems’, *arXiv:1603.04467*, Available from: arxiv.org/abs/1603.04467.
- Vahab et al. (2019), ‘Applications of Object Detection System’, *International Research Journal of Engineering and Technology (IRJET)*, vol.6, no.4, pp.4186-4192.
- Andreopoulos, A. and Tsotsos, J.K. (2013) ‘50 years of object recognition: Directions forward’, *Computer Vision and Image Understanding*, vol.117, no.8, pp.827-891
- TapTapSee (2020), *TapTapSee - Assistive Technology for the Blind And Visually Impaired*. [Online] Available from: <https://taptapseeapp.com/> [Accessed 15 March 2020].
- Crone, J. (2015). *Designing for the blind and visually impaired — TapTapSee* [Online]. Available from: <https://medium.com/@jordanthelast/designing-for-learning-v2-taptapsee-a875f4f84073> [Accessed 15 March 2020].
- CloudSight Inc. (2019). *TapTapSee* (Version 3.3.4.3) [Mobile application software]. Retrieved from <https://play.google.com/store/apps/details?id=com.msearcher.taptapsee.android>

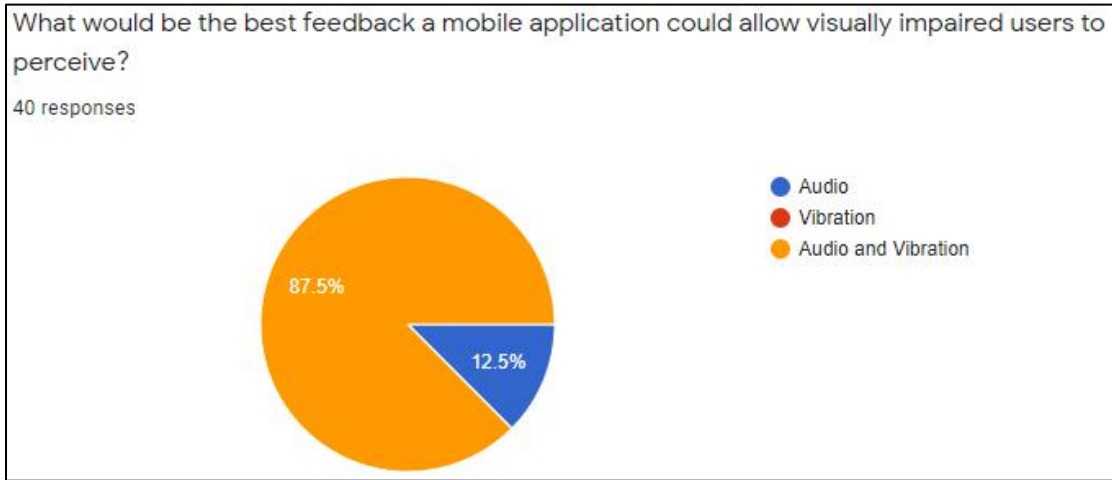
BIBLIOGRAPHY

- V7 Ltd. (2017). *Aipoly Vision* (Version 1.0.2) [Mobile application software]. Retrieved from <https://play.google.com/store/apps/details?id=com.aipoly.vision&hl=en>
- Hildenbrand, J. (2014). *Whats is Google TalkBack?* [Online]. Android Central. Available from: <https://www.androidcentral.com/what-google-talk-back> [Accessed 15 March 2020].
- Be My Eyes (2020). *Be My Eyes - See the world together* (Version 1.3.16) [Mobile application software]. Retrieved from : <https://play.google.com/store/apps/details?id=com.bemyeyes.bemyeyes>
- Cohen, J. P. (2015). *BlindTool - A mobile app that gives a 'sense of vision' to the blind with deep learning* [Online]. Joseph Paul Cohen. Available from: <https://josephpcohen.com/w/blindtool/> [Accessed 15 March 2020]
- Klotz, B. (2016). *Eyes in Your Pocket: "BlindTool" App Represents the New Frontier of Assistive Technology* [Online]. INSIGHT. Available from: <https://mabvi.wordpress.com/2016/01/13/eyes-in-your-pocket-blindtool-app-represents-the-new-frontier-of-assistive-technology/> [Accessed 15 March 2020].
- Marquart, S. (2016). *BlindTool: This Free App Verbally Identifies Objects For the Blind* [online]. Futurism. Available from: <https://futurism.com/kickstarter-blindtool-v2> [Accessed 15 March 2020].
- Lasane, A. (2016). *This Smartphone App Aims to Help the Blind See* [Online]. Mental Floss. Available from: <https://www.mentalfloss.com/article/73303/smartphone-app-aims-help-blind-see> [Accessed 15 March 2020].
- Cohen, J. (2020). *Joseph Paul Cohen* [Online]. Available from: <https://josephpcohen.com/w/> [Accessed 15 March 2020].
- Microsoft Corporation (2019). *Seeing AI* (Version 3.3.2) [Mobile application software]. Retrieved from: <https://apps.apple.com/us/app/seeing-ai/id999062298#?platform=iphone>
- Quantum (2018). *Seeing AI and OrCam MyEye: A Comparison* [Online]. Available from: <https://www.microlinkpc.com/wp-content/uploads/Seeing-AI-and-OrCam-MyEye-A-Comparison.pdf> [Accessed 16 March 2020]

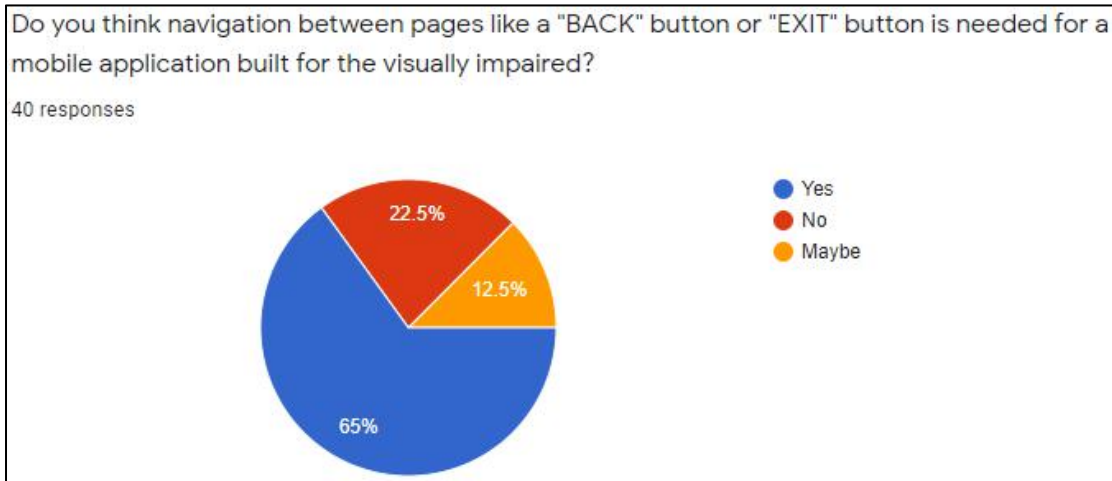
BIBLIOGRAPHY

- Envision Technologies BV (2020). *Envision AI* (Version 1.4.1) [Mobile application software]. Retrieved from <https://play.google.com/store/apps/details?id=com.letsenvision.envisionai&hl=en>
- Holton, B. (2013). *A Review of the TapTapSee, CamFind, and Talking Goggles Object Identification Apps for the iPhone* [Online]. Available from: <https://www.afb.org/aw/14/7/15675> [Accessed 16 March 2020].
- Sparkling Apps (2019). *Talking Goggles* (Version 1.37) [Online]. Retrieved from: <https://play.google.com/store/apps/details?id=com.sparkling.talkinggogglesfree> [Accessed 14 Aug 2019].

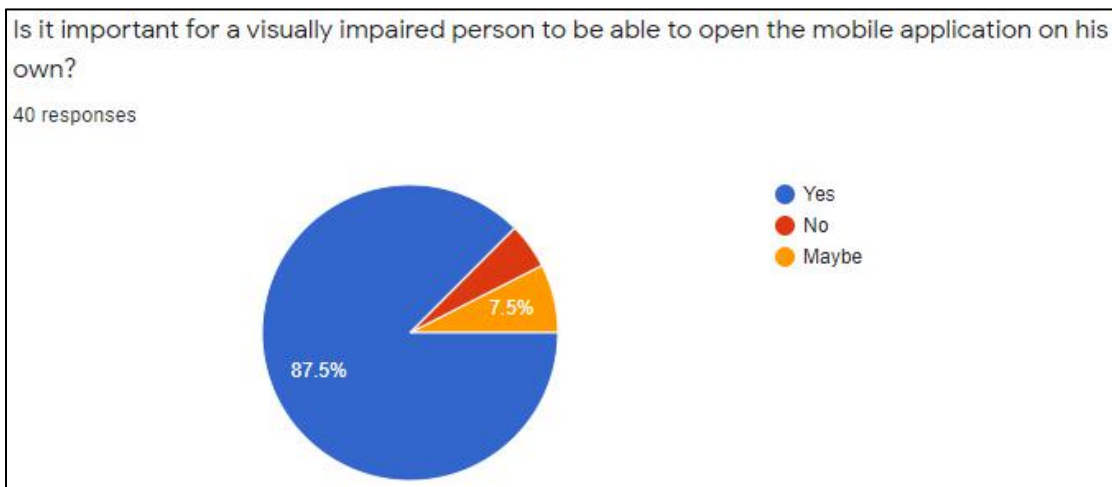
Customer Evaluation Outcome



A.1

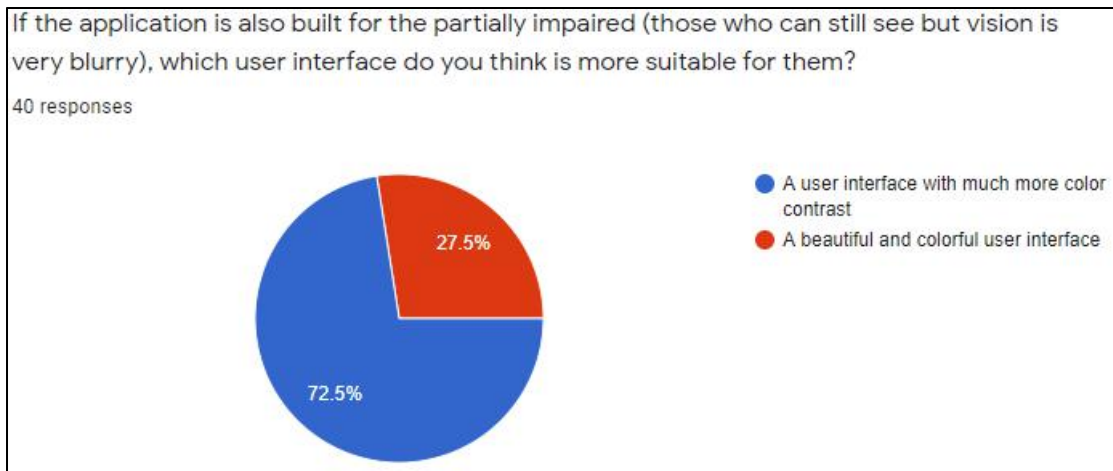


A.2



A.3

APPENDIX A



A.4

Poster

Mobile Application for the Visually Impaired

What features do this application have?

- Dial to Call
- Register Contact
- Recognize Object
- Recognize Text
- Voice Assistant
- Unique Accessibility



Project Objectives

- Provide visual aid for the visually impaired
- Optimize smartphone usability for the users
- Allow users to communicate with others



Methodology

1. Review existing applications
2. Design application flow and workflows
3. Develop application using Android Studio and Java



Conclusion

- The novel aspect of the project is the integration of extra functions with object recognition function which can assist the visually impaired.
- The project will aim to improve the usability and accessibility of the application.

by Darren Tan Yung Shen

Plagiarism Check Result

Feedback Studio - Google Chrome

ev.turnitin.com/app/carta/en_us/?student_user=1&u=1100902824&lang=en_us&to=1383708097&ts=

Darren TAN | Mobile Application for the Visually Impaired [FYP 2]

Match Overview

2%

Rank	Source	Match Percentage
1	Hanan Jabnoun, Faouz... Publication	<1%
2	ntrs.nasa.gov Internet Source	<1%
3	eprints.utar.edu.my Internet Source	<1%
4	www.phonestablets.co... Internet Source	<1%
5	digital.library.unt.edu Internet Source	<1%
6	scholar.uwindsor.ca Internet Source	<1%
7	josephcohen.com Internet Source	<1%
8	www.dtic.mil Internet Source	<1%

Chapter 1

Introduction

1.1 Problem Statement and Motivation

The first problem statement is the inability of visually impaired users to perceive information by the sense of sight. Visually impaired people usually rely on their sense of touch, hearing, smell and taste. They have to challenge themselves daily in coping with their impaired vision in a universe where majority of the people perceives their environment with vision (Koestler 1976). Hence, our project will try to provide a solution to the problem.

Besides that, communication and access to information is a huge problem for the visually impaired people (Arati, Sayali, Sushanta & Harshata 2015), especially when it comes to terms of utilizing a smartphone to perform some actions. Performing a call or knowing your phone's certain status using a smartphone is difficult for visually impaired users. Many mobile applications developed for the visually impaired out there do not provide extra functionality like that for this category of users to access. Their sole focus is only on retrieving information of items interpreted by the mobile camera. There are several more aspects that are yet to be explored and has potential in assisting those who are visually impaired via mobile application.

Next, visually disabled people have limited accessibility to smart mobile devices' functionality (Dobosz 2017). Nowadays, most of the devices such as tablets and smartphones require users to only touch on the surface of the screens (Dobosz 2017).

Turnitin

turnitin.com/newreport_classic.asp?lang=en_us&oid=1383708097&ft=1&bypass_cv=1

Turnitin Originality Report

Document Viewer

Processed on: 10-Sep-2020 23:32 +08
ID: 1383708097
Word Count: 14240
Submitted: 1

Mobile Application for the Visually Impaired ...
By Darren TAN

Similarity Index	Similarity by Source
2%	Internet Sources: 2% Publications: 0% Student Papers: 0%

exclude quoted | exclude bibliography | exclude small matches | mode: quickview (classic) report | Change mode

print | download

- <1% match (publications)
[Hanan Jabnoun, Faouzi Benzarti, Hamid Amiri, "Object detection and identification for blind people in video scene", 2015 15th International Conference on Intelligent Systems Design and Applications \(ISDA\), 2015](#)
- <1% match (Internet from 04-Jul-2019)
<https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19980007312.pdf>
- <1% match (Internet from 09-Sep-2018)
<https://www.phonestablets.co.ke/product/huawei-p-smart-plus-nova-3i-4gb-price-specs-in-kenya>
- <1% match (Internet from 24-Jun-2020)
<http://eprints.utar.edu.my>
- <1% match (Internet from 04-Jun-2020)
https://digital.library.unt.edu/ark:/67531/metadc716532/m2/1/high_res_d/769024.pdf
- <1% match ()
<http://scholar.uwindsor.ca>
- <1% match (Internet from 13-Apr-2019)

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1 of 1



**FACULTY OF INFORMATION AND COMMUNICATION
TECHNOLOGY**

Full Name(s) of Candidate(s)	DARREN TAN YUNG SHEN
ID Number(s)	16ACB03924
Programme / Course	COMPUTER SCIENCE
Title of Final Year Project	MOBILE APPLICATION FOR THE VISUALLY IMPAIRED

Similarity	Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)
Overall similarity index: <u>2%</u> Similarity by source Internet Sources: <u>2%</u> Publications: <u>0%</u> Student Papers: <u>0%</u>	
Number of individual sources listed of more than 3% similarity: <u>0</u>	
Parameters of originality required and limits approved by UTAR are as follows: (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: Lim Jit Theam

Date: 11/9/2020

Signature of Co-Supervisor

Name: _____

Date: _____



UNIVERSITI TUNKU ABDUL RAHMAN


FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	16ACB03924
Student Name	DARREN TAN YUNG SHEN
Supervisor Name	LIM JIT THEAM

TICK (√)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Front Cover
√	Signed Report Status Declaration Form
√	Title Page
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

<p>I, the author, have checked and confirmed all the items listed in the table are included in my report.</p> <p style="text-align: center;"><i>Darren</i></p> <hr style="width: 20%; margin-left: auto; margin-right: auto;"/> <p>(Signature of Student) Date: 11/9/2020</p>	<p>Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction.</p> <div style="text-align: center;">  </div> <hr style="width: 20%; margin-left: auto; margin-right: auto;"/> <p>(Signature of Supervisor) Date: 11/9/2020</p>
---	--