

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

1.1. Background Study

Now it is obvious in any statistics that road accidents all over the world is on a rise with an ever-rising trend, as a matter of fact 70% of 1.17 million deaths occur each year worldwide due to road accidents occurred in developing countries such as Malaysia(Kareem, 2003). Statistically from 2005-20010, there is an increase of vehicle registration of 11% but the number of accidents shows an increase of 26% (Malaysia Institute of Road Safety, Feb 2014) and this increase have been recorded all over the world. In fact In 1990, death on road accidents remained in 9th rank; and by 2020 road accidents will be the third leading cause of death worldwide and among these the highest fatality belong to drivers around 20 years old (Kareem, 2003).

Information now plays a crucial and significant role in the advancement of the human civilization. Technology is constantly being upgraded to meet the demands of users. And with the creation of Smartphone, the growth of technology and information continue to increase. This little device brings people of all ages in any part of the world together; in work or play. Worldwide, owning a mobile phone is a necessity for purpose of convenience. Features and functionalities keep on being added to the smartphone to accommodate life's necessities. This in turn pushes the limits of the device's capabilities higher and higher as new models are introduced into the market. The hardware and software capacity continues to be enhanced rise and thus the cycle of upgrades continues to roll. But this growth of convenient features is now linked to the increase in accidents.

With the growth of the smartphones, mobile applications or commonly known as apps are growing rapidly in all devices. In Android market alone, 1.5 billion apps are being downloaded each month, and the numbers are growing. Although majority of apps are social apps, in second place is the utility and productivity apps; showing a growth of 149% in 2013. This shows more people are using their mobile phone to improve the quality of life (Intel, 2014).

1.2. Problem Statement

Plans like mobile data plans and internet downloads facilitate the updating of apps all the time and with ease. The risk arise when a notification is received the form of message, mails or chat while driving. Users may impatiently want to immediately view the new update and this will cause the driver to lose focus on his driving and be more prone to accidents. A risky and hazardous situation arises when they attempt to read or type a text reply. , Research shows that the recipient/driver will spent 400% more time looking away from the road (Nemme, H. E., & White, K. M., 2010). This will cause a significant delay in reaction time of the driver, thus increasing the possibility of an accident.

1.3. Objectives of study

The objective of this research is to

- i. develop a voice activated Android App to minimize the distraction that the mobile phone causes to the driver

1.4. Scope of study

The area of study will be focused on the distraction that the drivers faced

- i. The built in, available open source voice command and text to speech functionality in Android
- ii. Use existing tools such as Google Now Voice command and Google Text to Speech(TTS)
- iii. Dedicated for car drivers

CHAPTER 2

LITERATURE REVIEW

2. LITERATURE REVIEW

2.1 Accidents

Safety problems related to drivers' distraction have been foreseen to increase as technology facilitates life's convenience. The usage of mobile phone while driving is a practical example of such distraction. Surprisingly this issue of the mobile devices being a distraction has been researched since 1993 which was the early years of the mobile phones. Academicians suspect mobile devices that used by the population are actually a distraction to drivers (Violanti, J. M., & Marshall, J. R., 1996)., The mobile phone is said be a potential distraction for the driver when mobile phone becomes an activity, event or object that competes for the driver`s attention when he is driving. (Dragutinovic, N., & Twisk, D., 2005). Researches also show and prove that drivers using phones had a larger number of accidents in their driving histories. Furthermore, frequent usage of mobile phone while driving also tend to show traits like having additional traffics violations such as failure to use seat belts, aggressive driving, speeding, and other traffic violations behavior (Zhao, N. et al).

Generally, distractions of mobile phone are categorized into two (2) categories; the physical distraction and the cognitive distraction. In the physical distraction, especially hand-held phone, the driver will use one hand to handle the mobile phone while the other to operate the vehicle. During this period dual task, they may also need to divert their eyes from the road to find, or pick up the phone and diverts more if they are texting or reading. Even hands free cause distraction when the driver will have to at least press one button to activate but it is significantly less risky than the hand-held (The Royal Society for the Prevention of Accidents, RoSPA, 2003).

The second category of distraction is in the mental or cognitive distraction that occurs when the driver have to divide or switch his attention between his actions or task of driving to that of attending to the mobile phone. For example the dual mental attention of steering the vehicle which require road decision-making and talking on the mobile phone which may be of totally different subject matter. This lack of focus and attention caused by competing task will increase hazard and the probability of having an accident (RoSPA, 2003).

2.2 Voice Command

The first step in the voice command system is the conversion of Speech to Text or STT where an input of voice from the user will be captured and translated into actions for the system. Generally, all algorithms in speech recognition have the same basic steps involving feature extraction, acoustic models database, dictionary, language model and the speech recognition algorithm. It usually starts with sampling the analog speech signal based on amplitude and time axes. The result from the sampling will then be sliced into chunks of fixed interval and features of the speech will be compared to acoustics model available. The acoustics model will then be compared to a dictionary to determine the word structure used in the blocks. The results will then be run through the speech recognition model to determine the correct words in the speech (Reddy, B. R et al, Foong, O. M et al).

Different algorithms have different accuracy of speech recognition because of the differences in speaker dependence vs. independence, vocabulary size, and condition of the speech, task and language constraints. There are multiple existing voice recognition algorithm available such as Hidden Markov Model (HMM), Mel Frequency Cepstral Coefficient (MFCC), Dynamic Time Warping (DTW) and many more (Muda, L et al). The most widely used is the Hidden Markov Model (HMM) because it is said to be the most flexible and successful algorithm among them (Reddy, B. R et al). The HMM is also the algorithm used by the Google for their Google Now voice command system and is available for all devices on all platforms because of its reliability (Reddy, B. R et al, Foong, O. M et al).

Although the STT have been developed from 1936, even until today the existing algorithm is not perfect and as it is caused by the complexities of the language as well as the background noise that disrupts the capturing of information or speech. Due to recent advancement of technology like the cloud-computing, they have been research done to further improve the accuracy of STT by means of cloud-computing where the cloud computing will allow the system to check the accuracy of the speech recognition model. This is because the system will be able to counter check with the internet and the database for possible words which will in turn increase the overall accuracy of the system (Stefanovic, M et al).

2.3 Existing Works

There are existing researches done on the voice command to aid the society in daily life for example the Remote Computer Control System Using Speech Recognition Technologies of Mobile Devices where the target of this research is on the blind or physically disabled. Their disability becomes a barrier for these people to access computers because of the lack of flexibility to handle the keyboard or mouse which is usually the main controller for the computer. The system was designed to allow them to easily control a computer by the means of speech. The proposed product was a mobile device that is connected to a Google server; this device will be a remote access to the computer acting as a remote to the users. The device will allow them to basic thing like read and write mails, manage schedule, and even basic usage of the computer. Using the device, they will not have any trouble accessing their contents in the computer including documents as the system will allow them listen to their documents via the Text to Speech function in Google (Jeong, H. D et al).

CHAPTER 3

METHODOLOGY

3. METHODOLOGY

3.1 Research Methodology

The data collection was done by interviewing 20 respondents within the UTP community. The sample was chosen because most of people within the community; lecturers, staff and students, have driving experience either using their own vehicles or that of others. Lecturers and staffs generally drive more as they reside in the neighboring towns like Batu Gajah and Ipoh and have to commute to the university almost daily. The respondents to the survey/interview are lecturers, students and UTP staff. The interview was conducted within the timeframe of 5-10 minutes as the respondents are mostly occupied with their own work. Questions asked includes their background such as frequency of travel, distance travelled, the frequency of notifications received via mobile phone as well as the manner of response if any to the notification. The near miss or accidents that the respondents have encountered resulting from using the mobile phone while driving, was also asked in the interview. The interview was concluded by asking the respondents of their opinion on the prospect of the app application based on the respondent's responses throughout the interview.

3.2 System Methodology

3.2.1 Planning

The project title was designed with the objective of reducing traffic accidents possibly cause by the usage of mobile phone while driving. The preliminary requirements of the system was drafted and from there the interview questions was developed to investigate what requirements users would like to have incorporated in the system to achieve the objective. The project scope and objective were strictly adhered in the drafting of the interview questions. This is to ensure that the interview will meets its specific target without going

beyond the scope. The timeline and deliverables was also marked to ensure the project is on course.

3.2.2 Analysis

The interview was conducted to collect information that is relevant to the system proposed. Based on the interview, an analysis of the project was then done. Based from the interview, the primary requirements and sub-requirements were marked as well as the suggestion and comments from the respondents. The opinion and remarks was noted to ensure error was minimized. The interview results were used as a base to proceed with the project. The project was also analyzed again to ensure all the required functionalities are accounted for.

3.2.3 Design

Based on the results obtained from the analysis, the functional and non-functional requirements are checked and incorporated in to designing the system. The design that was selected was the 3-tier architecture of application, Speech Kit and the Server. This design is in compliance with the design Android system. The design of the system does not require any additional hardware as all of the components are readily available in the mobile device itself.

The app will be using the Google Now as the default voice command app to receive the preset commands. Google Now will be a program that works in the background and will then translate the commands and turn the latest notification to be vocalized using the Google Text to Speech engine that will be available in the device. The Google Now and Google Text to Speech will be automatically downloaded to the device when this app is activated.

3.2.4 Implementation

The prototyping method will be used to implement the system; this will allow parts of the system to be developed while improving on the other functionality of the system. Regular internal testing will be done to ensure all parts of the system will function normally when the user testing comes at the end of the project.

3.3 System Architecture

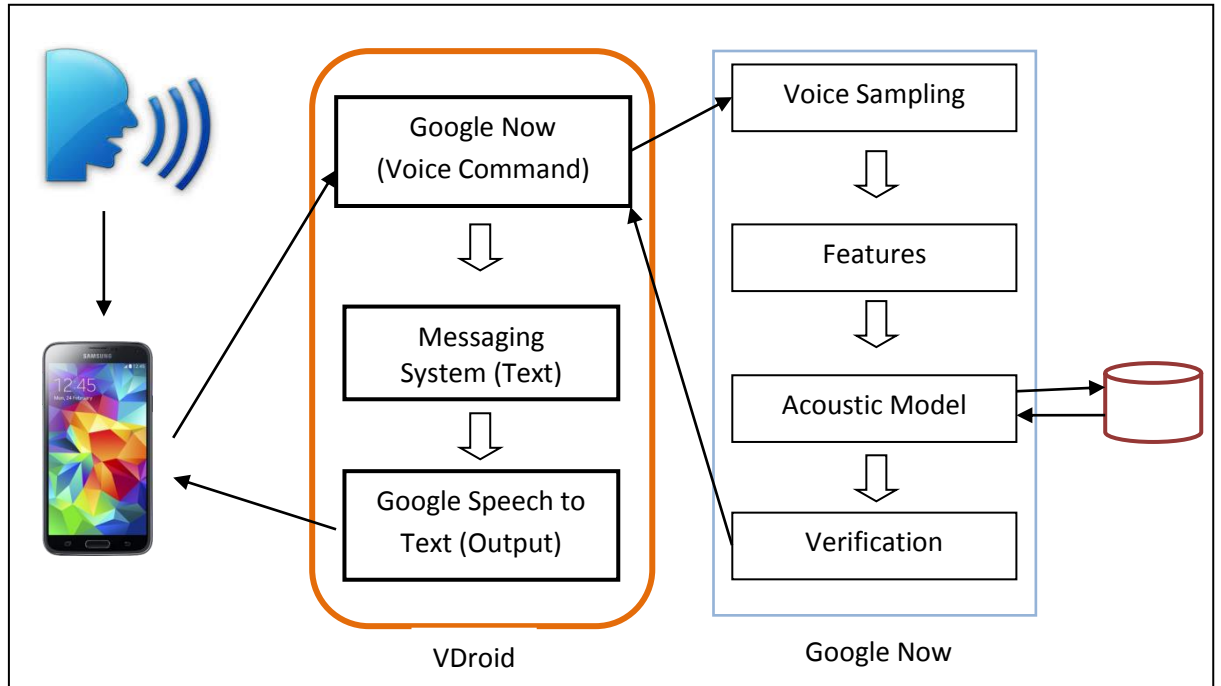


FIGURE 1: System Architecture

The system will be in an Android OS environment where there will be 3 layers. The first is the application layer which will display simple interaction with the user like when it is capturing the command or when it is reading out aloud. The interaction in the application layer will trigger the speech kit where the existing tool which is Google Now will take place. The tools will use its recognizer or microphone to capture the user input, and then data will be sent to the Google Now where its built in Speech to Text (STT) will interpret the command that it receives.

The command that the device will receive is a command that have already been preset in the system. Then VDroid will be responsible for the conversion of Text to Speech where the Google Text To Speech engine in the process will turn the text in the received SMS to speech for the user once it have confirm the command. The internet will only be used for to check the command speech. After the confirmation of order is verified, the VDroid will convert the text to speech and by using the Vocalizer, the content of the notification will be read aloud.

3.4 Activity Diagram

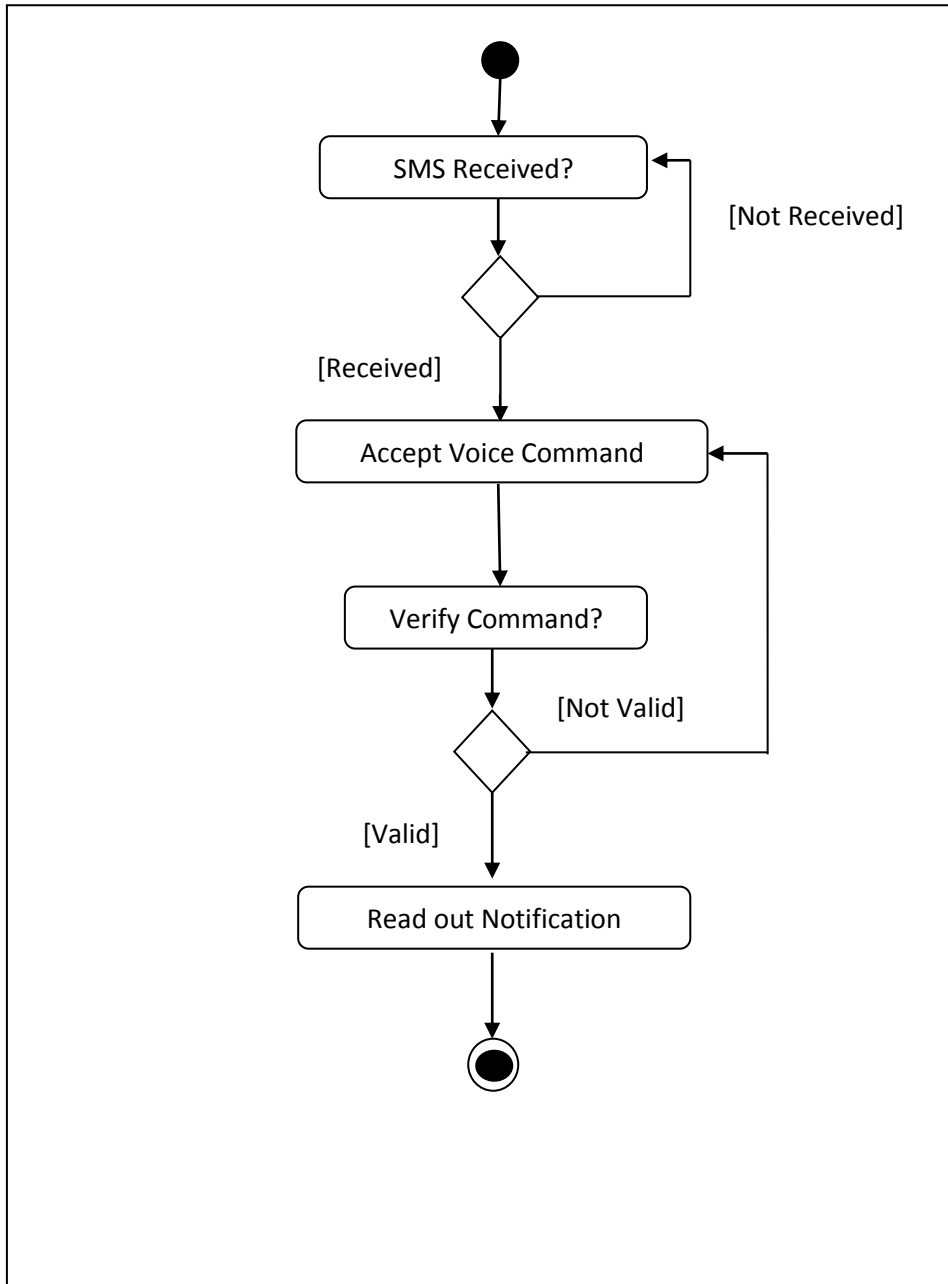


FIGURE 2: Activity Diagram

3.5 Requirements Analysis and Specification

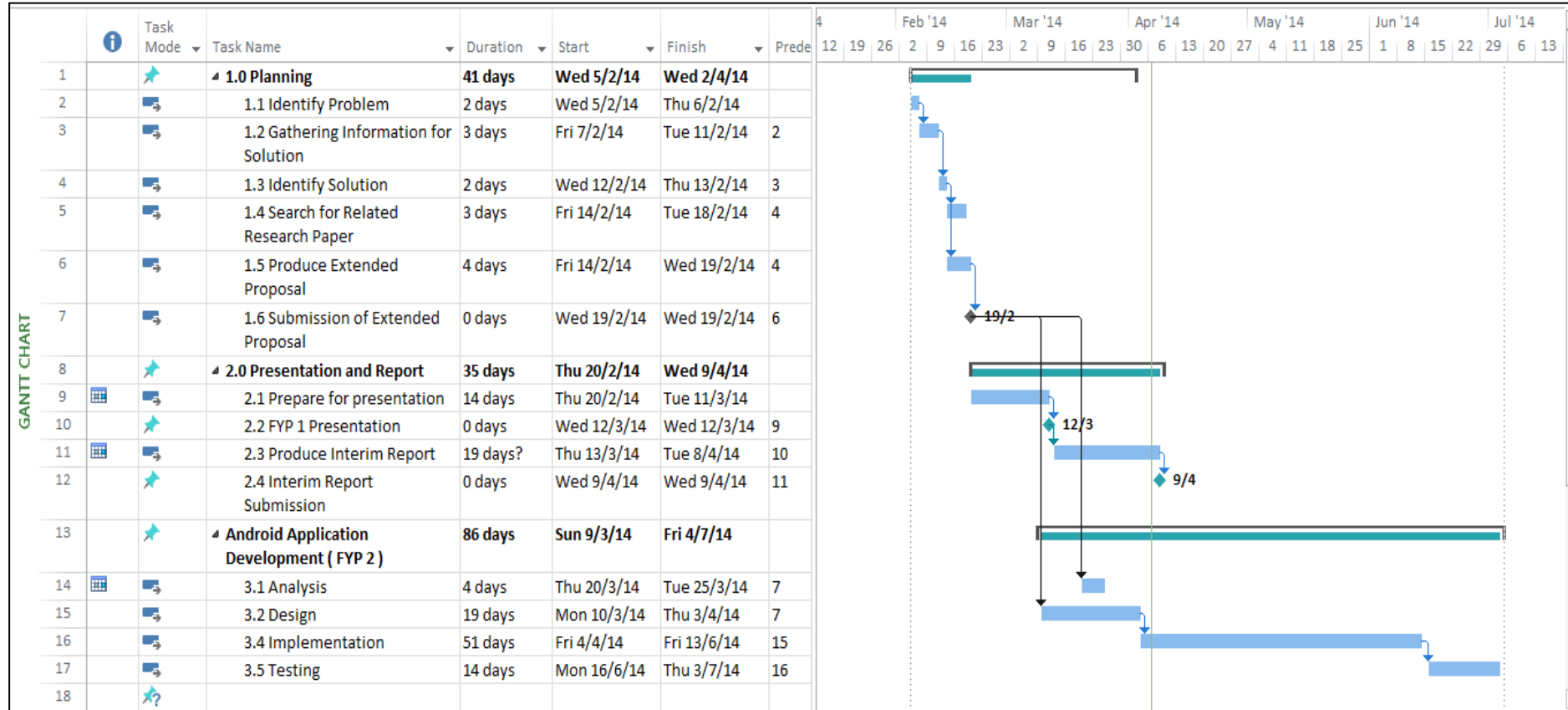
3.5.1 Functional Requirements

1. Remain on standby until SMS arrive
2. Capture the voice command after the SMS alert is rung
3. Verify the command received
4. Read the Contents of the SMS
5. Store Message
6. Allow old message to be read aloud again

3.5.2 Non-Functional Requirements

1. Run in Android platform

3.6 Gantt Chart



3.7 Requirements and Tools

Hardware

- Acer Aspire with Eclipse IDE for development
- Android Smartphone (Samsung Galaxy S2) for the mobile testing

Software

- Android Kit-Kat for the Android OS in the Samsung phone
- Android SDK for development on Android OS
- Cyanogen ROM for the android system used in Samsung S2
- Cyanogen UI to prototype User interface
- Google Now voice command to handle voice input
- Google Text To Speech Engine for the vocalization of notification

Programming Language

- Java
- SQL

CHAPTER 4

RESULTS AND FINDINGS

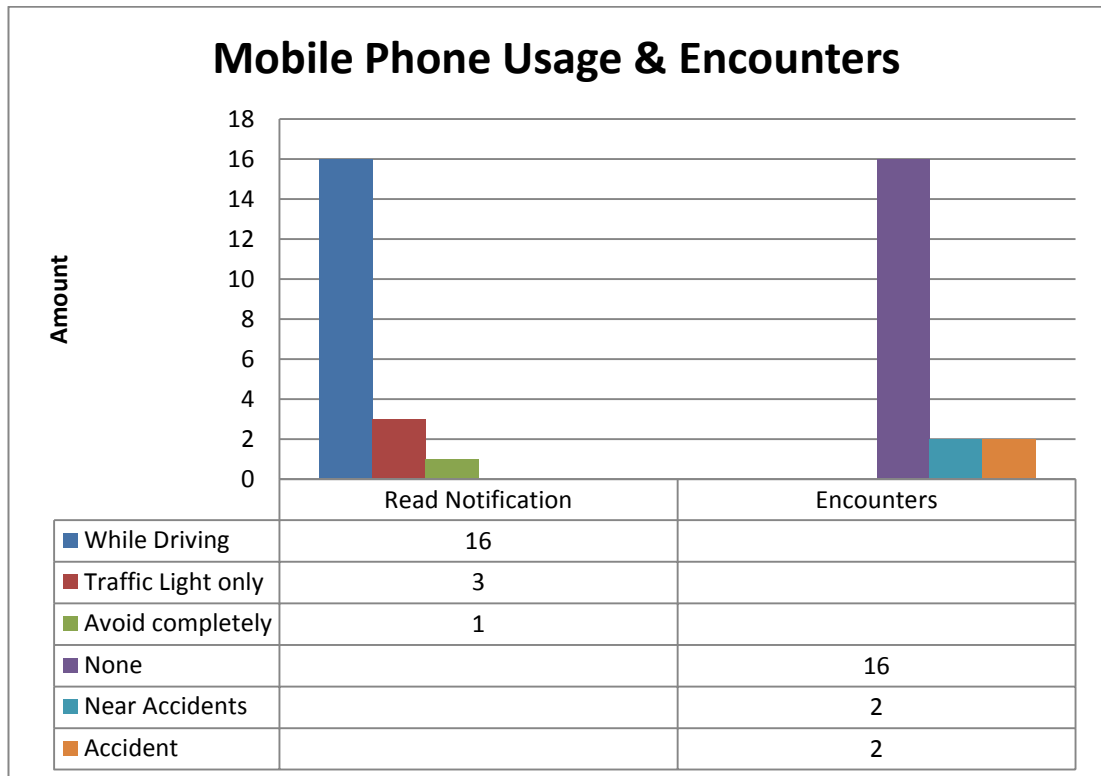
4. Results and Findings

4.1 Initial Data Gathering and Analysis

In the research, an interview was conducted to find the correlation between the increasing rate of accident and the usage of mobile phones while driving. The interview is to investigate the amount and behavior of people who use their mobile device while driving and their experience while using it. The respondents were a mixture of people in UTP community which include lecturers, staff and students. The respondents were a mixture of gender, age and background.

A total of 20 respondents were surveyed and the results were tabulated as follow

TABLE 1: Read Notification while Driving



From the table above we can see that 16 out of 20 respondents actually read their notification while driving while 3 only if their car is stationary and 1 avoid using mobile phone while driving. In term of encounters, 16 out of 20 have no encounters when they are using the phone but 2 have had close encounter while other 2 have experienced an accident with mobile phone as the cause.

4.2 Initial System Interface Design



FIGURE 3: Front Page



FIGURE 4: Capturing Content



FIGURE 5: Conversion Progress



FIGURE 6: Reading Content

4.3 Final Product

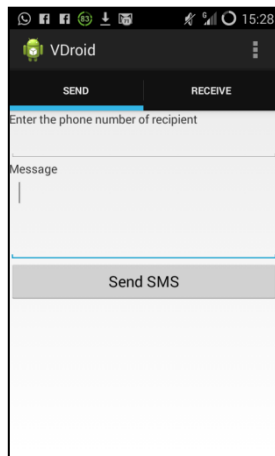


FIGURE 7: Send SMS

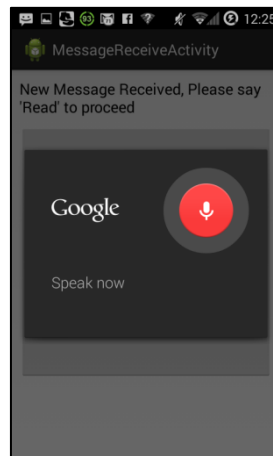


FIGURE 8: Voice

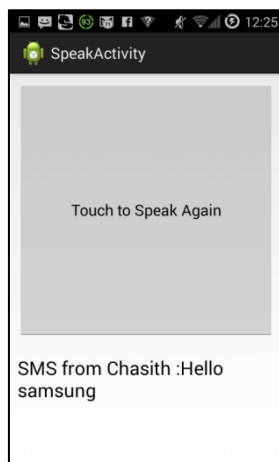


FIGURE 9: Read Out

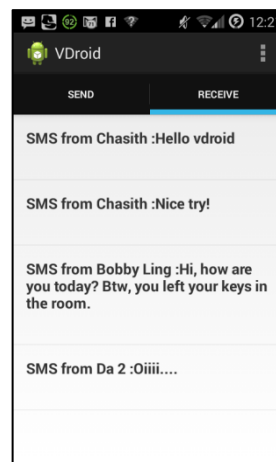


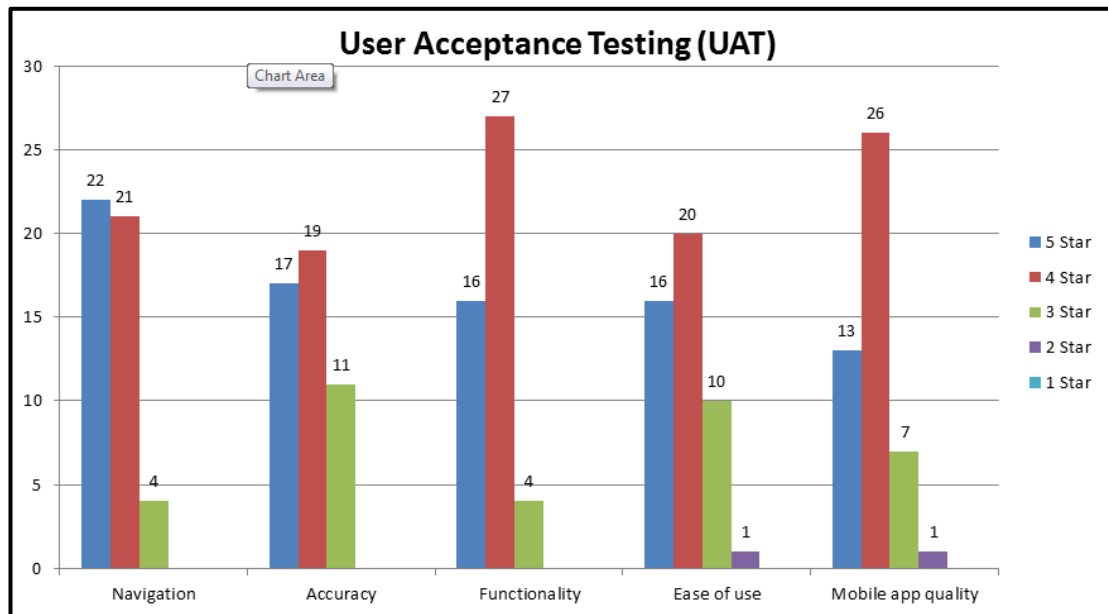
FIGURE 10: Inbox

4.4 User Acceptance Testing (UAT)

After the final Prototype was completed, a User Acceptance Test was conducted on 47 respondents from age 18 and above. 18 years old is the legal driving age in Malaysia evaluate if the VDroid will be able to complete its objective as well as the user reaction to the app. The tester was given a brief explanation and demo on the workings of the system and was allowed to play with the apps as they wished. After the users are satisfied with the testing, they are directed to the webpage of the online UAT form for the respondents to grade the prototype.

The results of the 47 User Acceptance Testing is tabulated in the table below

FIGURE 11: User Acceptance Testing Result Graph



From the Figure above, we can conclude that that the almost all the testers are satisfied with the navigation of the VDroid, this criteria checks on the ease of navigation for the user when using this app. Navigation scored 5 stars by 22 people and 4 star from 21 people and it is due to the simplicity of the Graphical User Interface (GUI). Whereas in accuracy most testers give a score of 4 star with 19 people followed by 17 people giving 5 star. This lower score is seen due to the inaccurate capturing of command by the system cause by high background noise and interference which cause the need to repetitively give commands to get the correct response. The functionality scored mostly 4 stars with users being excited with the functionality introduced. The ease of used scored mostly 5 stars and 4 stars with the system being design for drivers, thus making it easy to use is crucial. In terms of app quality majority scored 4 stars and this is due to the lack of more contrasting color and design in the app. In a whole, the VDroid is well accepted by the users with positive remarks and praise with a high average of 4.19/5.0.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

As a conclusion, there are many applications and functionality in the market and in the device itself, which are currently being used for life's convenience. The integration of Voice To Text and Text To Voice is actually a quite common application in the world however, its application as per the proposed system is still not available. This system will potentially increase the safety on the road for all users since drivers will have less distraction to take care. This VDroid app is only meant to reduce the probability and not to completely eliminate the hazard. In the pilot study, all the respondents gave positive comments on the system and majority of them are awaiting for the completion of the system. The system was regarded as a very convenient tool for drivers as it will help in their daily driving necessity. During the UAT process, many positive remarks were given on the application with mainly being useful and convenient. The practicality and the driver friendliness is welcomed by the testers.

5.2 Future Works

In the pilot study most of the respondents gave ideas to further improve the system like giving more functionality in all sectors. An example is the security, where the user will be prompted with the sender's tone of the notification to allow the driver to decide whether to read the SMS or not. Future features on the system is the possibility for the user to directly reply by using their voice that will be converted to text. This will allow the driver to both read and reply to the notification. Another proposed future work is to upgrade the language selection as the current one only supports English language.

REFERENCES

- Backer-Grøndahl, A., & Sagberg, F. (2011). Driving and telephoning: Relative accident risk when using hand-held and hands-free mobile phones. *Safety Science*, 49(2), 324-330. doi: <http://dx.doi.org/10.1016/j.ssci.2010.09.009>
- Dragutinovic, N., & Twisk, D. (2005). Use of mobile phones while driving—effects on road safety. *SWOV Institute for Road Safety Research, Leidschendam, The Netherlands, Report, 12*, 2005.
- Foong, O. M., Low, T. J., & La, W. W. (2009). V2S: Voice to Sign Language Translation System for Malaysian Deaf People. In *Visual Informatics: Bridging Research and Practice* (pp. 868-876). Springer Berlin Heidelberg.
- Jeong, H. D. J., Sang-Kug, Y., Jiyong, L., Ilsun, Y., Wooseok, H., & Hee-Kyoung, S. (2013, 3-5 July 2013). *A Remote Computer Control System Using Speech Recognition Technologies of Mobile Devices*. Paper presented at the Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013 Seventh International Conference on.
- Kastelan, I.; Kukolj, S.; Pekovic, V.; Marinkovic, V.; Marceta, Z., "Extraction of text on TV screen using optical character recognition," *Intelligent Systems and Informatics (SISY), 2012 IEEE 10th Jubilee International Symposium on* , vol., no., pp.153,156, 20-22 Sept. 2012 doi: 10.1109/SISY.2012.6339505
- Muda, L., Begam, M., & Elamvazuthi, I. (2010). *Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques*. arXiv preprint arXiv:1003.4083.
- Nemme, H. E., & White, K. M. (2010). Texting while driving: Psychosocial influences on young people's texting intentions and behaviour. *Accident Analysis & Prevention*, 42(4), 1257-1265. doi: <http://dx.doi.org/10.1016/j.aap.2010.01.019>
- Pöysti, L., Rajalin, S., & Summala, H. (2005). Factors influencing the use of cellular (mobile) phone during driving and hazards while using it. *Accident Analysis & Prevention*, 37(1), 47-51. doi: <http://dx.doi.org/10.1016/j.aap.2004.06.003>
- Reddy, B. R., & Mahender, E. *Speech to Text Conversion using Android Platform*. Vol. 3, Issue 1, January -February 2013, pp.253-258
- Stefanovic, M., Cetic, N., Kovacevic, M., Kovacevic, J., & Jankovic, M. (2012, 20-22 Nov. 2012). *Voice control system with advanced recognition*. Paper presented at the Telecommunications Forum (TELFOR), 2012 20th.
- the, T. R. S. f., & Accidents, P. o. (2003). *The Risk of Using a MobilePhone While Driving*. Retrieved 5 March, 2014

Törnros, J. E. B., & Bolling, A. K. (2005). Mobile phone use—Effects of handheld and handsfree phones on driving performance. *Accident Analysis & Prevention*, 37(5), 902-909. doi: <http://dx.doi.org/10.1016/j.aap.2005.04.007>

Violanti, J. M., & Marshall, J. R. (1996). Cellular phones and traffic accidents: An epidemiological approach. *Accident Analysis & Prevention*, 28(2), 265-270. doi: [http://dx.doi.org/10.1016/0001-4575\(95\)00070-4](http://dx.doi.org/10.1016/0001-4575(95)00070-4)

Zhao, N., Reimer, B., Mehler, B., D'Ambrosio, L. A., & Coughlin, J. F. (2013). Self-reported and observed risky driving behaviors among frequent and infrequent cell phone users. *Accident Analysis & Prevention*, 61(0), 71-77. doi: <http://dx.doi.org/10.1016/j.aap.2012.07.019>

APPENDICEX A

Initial Data Gathering Interview Questions

- 1) How often do you drive? Is it long distance or short distance mostly?
- 2) How often you will receive updates when driving?
- 3) What is usually the notification status? Is it usually urgent?
- 4) Have you attempted to receive call while driving? If yes, do you do it often?
- 5) Have you attempted to read or reply a text while driving? If yes, do you do it often?
- 6) Have you experience any close encounter while using the mobile? Or Accident?
- 7) What is your opinion on the application?
- 8) Do you think it will increase your cellphone usage on the road?

User Acceptance Testing Online Form

User Acceptance Testing

1 How old are you?

- <20
- 20-30
- 31-40
- >41

2* How often do you drive?

- Everyday
- Most Of The Time
- Sometimes
- Occasionaly
- Never

3* VDroid Acceptance Test (How is the App in your opinion?)

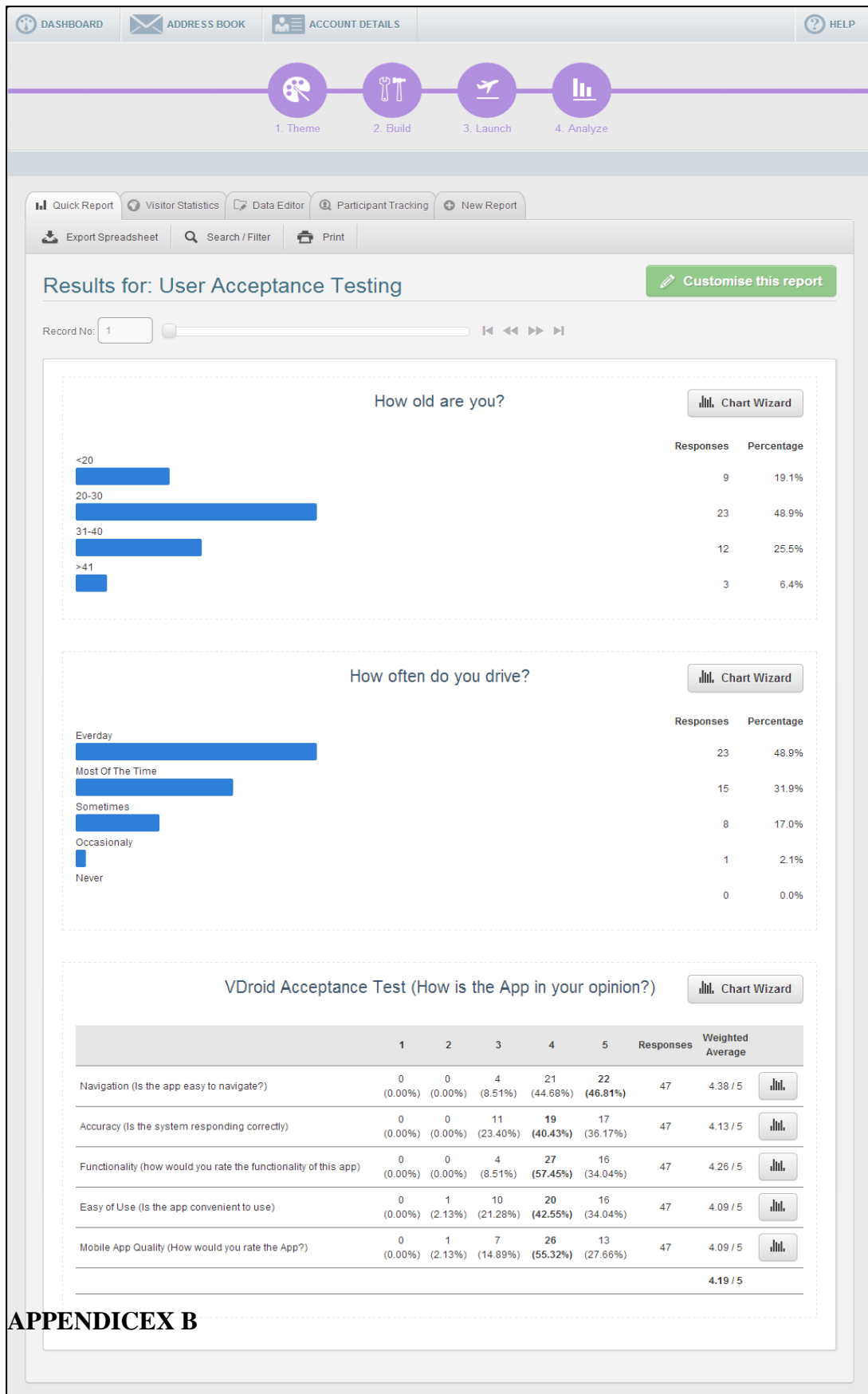
Navigation (Is the app easy to navigate?)	★★★★★
Accuracy (Is the system responding correctly)	★★★★★
Functionality (how would you rate the functionality of this app)	★★★★★
Easy of Use (Is the app convenient to use)	★★★★★
Mobile App Quality (How would you rate the App?)	★★★★★

Give you Comments

User Acceptance Testing Online Hyperlink

<http://kwiksurveys.com/s.asp?sid=oj83qqoujdw8mmt394046>

User Acceptance Testing Results



APPENDICEX B

Coding of System

Main Activity

```
package fyp.app.vdroid;

import java.util.Locale;

import android.app.ActionBar;
import android.app.FragmentTransaction;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentActivity;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;
import android.support.v4.view.ViewPager;
import android.view.Menu;

public class MainActivity extends FragmentActivity implements
        ActionBar.TabListener {

    SectionsPagerAdapter mSectionsPagerAdapter;

    ViewPager mViewPager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Set up the action bar.
        final ActionBar actionBar = getActionBar();

        actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);

        mSectionsPagerAdapter = new SectionsPagerAdapter(
            getSupportFragmentManager());

        // Set up the ViewPager with the sections adapter.
        mViewPager = (ViewPager) findViewById(R.id.pager);
        mViewPager.setAdapter(mSectionsPagerAdapter);

        // When swiping between different sections, select the corresponding
        // tab. We can also use ActionBar.Tab#select() to do this if we have
        // a reference to the Tab.
        mViewPager
            .setOnPageChangeListener(new
ViewPager.SimpleOnPageChangeListener() {
                @Override
                public void onPageSelected(int position) {

                    actionBar.setSelectedNavigationItem(position);
                }
            })
    }
```

```

    });

    // For each of the sections in the app, add a tab to the action bar.
    for (int i = 0; i < mSectionsPagerAdapter.getCount(); i++) {
        actionBar.addTab(actionBar.newTab()

.setText(mSectionsPagerAdapter.getPageTitle(i))
                .setTabListener(this));
    }

    actionBar.setSelectedNavigationItem(1);

    //testing code here
    //Intent i = new Intent(this,MessageReceiveActivity.class);
    //startActivity(i);

}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

@Override
public void onTabSelected(ActionBar.Tab tab,
        FragmentTransaction fragmentTransaction) {
    // When the given tab is selected, switch to the corresponding page in
    // the ViewPager.
    mViewPager.setCurrentItem(tab.getPosition());
}

@Override
public void onTabUnselected(ActionBar.Tab tab,
        FragmentTransaction fragmentTransaction) {
}

@Override
public void onTabReselected(ActionBar.Tab tab,
        FragmentTransaction fragmentTransaction) {
}

/**
 * A {@link FragmentPagerAdapter} that returns a fragment corresponding
to
 * one of the sections/tabs/pages.
 */
public class SectionsPagerAdapter extends FragmentPagerAdapter {

```

```

public SectionsPagerAdapter(FragmentManager fm) {
    super(fm);
}

@Override
public Fragment getItem(int position) {
    Fragment fragment;
    if(position == 0){
        fragment = new SendFragment();
    }
    else{
        fragment = new ReceiveFragment();
    }

    return fragment;
}

@Override
public int getCount() {
    return 2;
}

@Override
public CharSequence getPageTitle(int position) {
    Locale l = Locale.getDefault();
    switch (position) {
        case 0:
            return
getString(R.string.title_section1).toUpperCase(l);
        case 1:
            return
getString(R.string.title_section2).toUpperCase(l);
    }
    return null;
}
}
}

```

Message Receive Coding

```
package fyp.app.vdroid;

import java.util.ArrayList;

import fyp.app.controller.SQLController;

import android.app.Activity;
import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.view.View;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

public class MessageReceiveActivity extends Activity {

    protected static final int RESULT_SPEECH = 1;

    private ImageButton btnSpeak;
    private TextView txtText;
    SQLController dbcon;

    String message;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_message_receive);

        txtText = (TextView) findViewById(R.id.txtText);

        message = getIntent().getStringExtra("message");

        //insert information into the sqlite
        dbcon = new SQLController(this);
        dbcon.open();
        dbcon.insertData(message);

        btnSpeak = (ImageButton) findViewById(R.id.btnSpeak);

        btnSpeak.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {

                Intent intent = new Intent(

                RecognizerIntent.ACTION_RECOGNIZE_SPEECH);

                intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, "en-US");
```

```

        try {
            startActivityForResult(intent, RESULT_SPEECH);
            txtText.setText("");
        } catch (ActivityNotFoundException a) {
            Toast t = Toast.makeText(getApplicationContext(),
                "Ops! Your device doesn't support
Speech to Text, Please install Google Search",
                Toast.LENGTH_SHORT);
            t.show();
        }
    });

    Intent intent = new Intent(
        RecognizerIntent.ACTION_RECOGNIZE_SPEECH);

    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, "en-
US");

    try {
        startActivityForResult(intent, RESULT_SPEECH);
        txtText.setText("");
    } catch (ActivityNotFoundException a) {
        Toast t = Toast.makeText(getApplicationContext(),
            "Ops! Your device doesn't support Speech to Text,
Please install Google Search",
            Toast.LENGTH_SHORT);
        t.show();
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    switch (requestCode) {
        case RESULT_SPEECH: {
            if (resultCode == RESULT_OK && null != data) {

                ArrayList<String> text = data

                .getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);

                txtText.setText("\n" + text.get(0) + "\n is captured, please
say \"Read\" to proceed");

                if(text.get(0).equals("read")){
                    Intent i = new Intent (this, SpeakActivity.class);
                    i.putExtra("message", message);
                    startActivity(i);
                }
                else{

                    Intent intent = new Intent(

```

```

RecognizerIntent.ACTION_RECOGNIZE_SPEECH);

intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, "en-US");

        try {
            startActivityForResult(intent,
RESULT_SPEECH);
            txtText.setText("");
        } catch (ActivityNotFoundException a) {
            Toast t =
Toast.makeText(getApplicationContext(),
            "Ops! Your device doesn't
support Speech to Text, Please install Google Search",
            Toast.LENGTH_SHORT);
            t.show();
        }
    }
}
break;
}
}
}

@Override
public void onBackPressed(){
    Intent i = new Intent(this, MainActivity.class);
    i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    startActivity(i);
    finish();
}
}
}

```

Read Aloud Message Code

```

package fyp.app.vdroid;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class SpeakActivity extends Activity implements TextToSpeech.OnInitListener {

    private TextToSpeech mTts;
    // This code can be any value you want, its just a checksum.

```

```

private static final int MY_DATA_CHECK_CODE = 1234;

String message;

Button again;

TextView messageText;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_speak);

    message = getIntent().getStringExtra("message");

    again = (Button)findViewById(R.id.btnSpeakAgain);
    messageText = (TextView)findViewById(R.id.messageText);

    again.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            mTts.speak(message, TextToSpeech.QUEUE_FLUSH, null);
        }
    });

    // Fire off an intent to check if a TTS engine is installed
    Intent checkIntent = new Intent();
    checkIntent.setAction(TextToSpeech.Engine.ACTION_CHECK_TTS_DATA);
    startActivityForResult(checkIntent, MY_DATA_CHECK_CODE);
}

public void onInit(int i)
{
    messageText.setText(message);//setting text to message

    mTts.speak(message, TextToSpeech.QUEUE_FLUSH, // Drop all pending entries in
the playback queue.
    null);

}

public void onActivityResult(int requestCode, int resultCode, Intent data)
{
    if (requestCode == MY_DATA_CHECK_CODE)
    {
        if (resultCode == TextToSpeech.Engine.CHECK_VOICE_DATA_PASS)
        {
            // success, create the TTS instance
            mTts = new TextToSpeech(this, this);
        }
        else
        {
            // missing data, install it
            Intent installIntent = new Intent();

```

```

        installIntent.setAction(
            TextToSpeech.Engine.ACTION_INSTALL_TTS_DATA);
        startActivity(installIntent);
    }
}

//shut down the speech machine
@Override
public void onDestroy()
{
    // Don't forget to shutdown!
    if (mTts != null)
    {
        mTts.stop();
        mTts.shutdown();
    }
    super.onDestroy();
}
}

```

Receive Activity

```

package fyp.app.vdroid;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemLongClickListener;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;
import android.widget.TextView;
import android.widget.Toast;
import fyp.app.controller.DBhelper;
import fyp.app.controller.SQLController;

public class ReceiveFragment extends Fragment {

    ListView lv;
    SQLController dbcon;
    TextView memID_tv, messageContent;
    View rootView;
}

```

```

Long message_delete_id;

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    rootView = inflater.inflate(R.layout.fragment_receive, container, false);

    dbcon = new SQLController(getActivity());
    dbcon.open();
    lv = (ListView) rootView.findViewById(R.id.messageList);

    // Attach The Data From DataBase Into ListView Using Crusor Adapter
    Cursor cursor = dbcon.readData();
    String[] from = new String[] { DBHelper.MESSAGE_ID,
DBHelper.MESSAGE_CONTENT };
    int[] to = new int[] { R.id.message_id, R.id.message_content };

    SimpleCursorAdapter adapter = new SimpleCursorAdapter(
        getActivity(), R.layout.message_list_item, cursor, from,
to,0);

    adapter.notifyDataSetChanged();
    lv.setAdapter(adapter);

    // OnClickListiner For List Items
    lv.setOnItemClickListener(new OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view,
            int position, long id) {
            memID_tv = (TextView)
view.findViewById(R.id.message_id);
            messageContent = (TextView)
view.findViewById(R.id.message_content);

            //pass value to the new activity
            //String msg_id = memID_tv.getText().toString();
            String msg_content = messageContent.getText().toString();

            Intent i = new Intent(getActivity(),SpeakActivity.class);
            i.putExtra("message", msg_content);
            startActivity(i);
        }
    });

    lv.setOnItemLongClickListener(new OnItemLongClickListener() {
        @Override
        public boolean onItemLongClick(AdapterView<?> arg0, View arg1,
int pos, long id) {
            memID_tv = (TextView)
arg1.findViewById(R.id.message_id);
            messageContent = (TextView)
arg1.findViewById(R.id.message_content);

```

```

        //pass value to the new activity
        String msg_id = memID_tv.getText().toString();
        String msg_content = messageContent.getText().toString();

        //Toast.makeText(getActivity(), msg_id + " " + msg_content,
Toast.LENGTH_LONG).show();

        alertDelete(msg_id);

        return false;
    }
});

return rootView;
}

private void alertDelete(String msg_id) {
    message_delete_id = Long.parseLong(msg_id);

    AlertDialog.Builder alertDialog = new AlertDialog.Builder(getActivity());
    alertDialog.setTitle("Delete");
    alertDialog.setMessage("Delete this message from the inbox?");
    alertDialog.setPositiveButton("Cancel", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int which) {
            dialog.cancel();
        }
    });
    alertDialog.setNegativeButton("Yes",new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            dbcon.deleteData(message_delete_id);

            //refresh activity to remove the unwanted list item
            Intent i = new Intent(getActivity(),MainActivity.class);
            i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(i);
            getActivity().finish();

        }
    });
    // Showing Alert Message
    alertDialog.show();
}
}

```

SendSMS

```
package fyp.app.vdroid;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemLongClickListener;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;
import android.widget.TextView;
import android.widget.Toast;
import fyp.app.controller.DBhelper;
import fyp.app.controller.SQLController;

public class ReceiveFragment extends Fragment {

    ListView lv;
    SQLController dbcon;
    TextView memID_tv, messageContent;
    View rootView;

    Long message_delete_id;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        rootView = inflater.inflate(R.layout.fragment_receive, container,
false);

        dbcon = new SQLController(getActivity());
        dbcon.open();
        lv = (ListView) rootView.findViewById(R.id.messageList);

        // Attach The Data From DataBase Into ListView Using Crusor
Adapter
        Cursor cursor = dbcon.readData();
        String[] from = new String[] { DBhelper.MESSAGE_ID,
DBhelper.MESSAGE_CONTENT };
        int[] to = new int[] { R.id.message_id, R.id.message_content };
```

```

SimpleCursorAdapter adapter = new SimpleCursorAdapter(
    getActivity(), R.layout.message_list_item, cursor,
from, to,0);

adapter.notifyDataSetChanged();
lv.setAdapter(adapter);

// OnClickListiner For List Items
lv.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view,
        int position, long id) {
        memID_tv = (TextView)
view.findViewById(R.id.message_id);
        messageContent = (TextView)
view.findViewById(R.id.message_content);

        //pass value to the new activity
        //String msg_id = memID_tv.getText().toString();
        String msg_content =
messageContent.getText().toString();

        Intent i = new
Intent(getActivity(),SpeakActivity.class);
        i.putExtra("message", msg_content);
        startActivity(i);
    }
});

lv.setOnItemLongClickListener(new OnItemLongClickListener() {
    @Override
    public boolean onItemLongClick(AdapterView<?> arg0,
View arg1,
        int pos, long id) {
        memID_tv = (TextView)
arg1.findViewById(R.id.message_id);
        messageContent = (TextView)
arg1.findViewById(R.id.message_content);

        //pass value to the new activity
        String msg_id = memID_tv.getText().toString();
        String msg_content =
messageContent.getText().toString();

        //Toast.makeText(getActivity(), msg_id + " "+
msg_content, Toast.LENGTH_LONG).show();

        alertDelete(msg_id);

        return false;
    }
});

```

```

        });
    }

    return rootView;
}

private void alertDelete(String msg_id) {
    message_delete_id = Long.parseLong(msg_id);

    AlertDialog.Builder alertDialog = new
AlertDialog.Builder(getActivity());
    alertDialog.setTitle("Delete");
    alertDialog.setMessage("Delete this message from the inbox?");
    alertDialog.setPositiveButton("Cancel", new DialogInterface.OnClickListener()
{
        public void onClick(DialogInterface dialog, int which) {
            dialog.cancel();
        }
    });
    alertDialog.setNegativeButton("Yes",new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            dbcon.deleteData(message_delete_id);

            //refresh activity to remove the unwanted list item
            Intent i = new Intent(getActivity(),MainActivity.class);
            i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(i);
            getActivity().finish();

        }
    });
    // Showing Alert Message
    alertDialog.show();
}
}

```

ReceiveSMS

```

package fyp.app.vdroid;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.Contacts;
import android.telephony.gsm.SmsMessage;
import android.widget.Toast;

```

```

public class SmsReceiver extends BroadcastReceiver
{
    @Override
    public void onReceive(Context context, Intent intent)
    {
        //---get the SMS message passed in---
        Bundle bundle = intent.getExtras();
        SmsMessage[] msgs = null;
        String str = "";
        if (bundle != null)
        {
            //---retrieve the SMS message received---
            Object[] pdus = (Object[]) bundle.get("pdus");
            msgs = new SmsMessage[pdus.length];
            for (int i=0; i<msgs.length; i++){
                msgs[i] = SmsMessage.createFromPdu((byte[])pdus[i]);
                str += "SMS from " + getContactName(msgs[i].getOriginatingAddress(), context);
                str += " :";
                str += msgs[i].getMessageBody().toString();
                str += "\n";
            }
            //---display the new SMS message---
            Toast.makeText(context, str, Toast.LENGTH_SHORT).show();

            //goto the speak activation activity
            Intent i = new Intent(context,MessageReceiveActivity.class);
            i.putExtra("message", str);
            i.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            context.startActivity(i);
        }
    }

    public String getContactName(final String phoneNumber, Context context)
    {
        Uri uri;
        String[] projection;
        Uri mBaseUri = Contacts.Phones.CONTENT_FILTER_URL;
        projection = new String[] { android.provider.Contacts.People.NAME };
        try {
            Class<?> c =Class.forName("android.provider.ContactsContract$PhoneLookup");
            mBaseUri = (Uri) c.getField("CONTENT_FILTER_URI").get(mBaseUri);
            projection = new String[] { "display_name" };
        }
        catch (Exception e) {
        }
    }

    uri = Uri.withAppendedPath(mBaseUri, Uri.encode(phoneNumber));
    Cursor cursor = context.getContentResolver().query(uri, projection, null, null, null);

    String contactName = "";

    if (cursor.moveToFirst())
    {
        contactName = cursor.getString(0);
    }
}

```

```

    }

    cursor.close();
    cursor = null;

    return contactName;
}
}

```

SQLHandling

```

package fyp.app.vdroid;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;

public class SQLController {

    private DBHelper dbHelper;
    private Context ourcontext;
    private SQLiteDatabase database;

    public SQLController(Context c) {
        ourcontext = c;
    }

    public SQLController open() throws SQLException {
        dbHelper = new DBHelper(ourcontext);
        database = dbHelper.getWritableDatabase();
        return this;
    }

    public void close() {
        dbHelper.close();
    }

    public void insertData(String name) {
        ContentValues cv = new ContentValues();
        cv.put(DBHelper.MESSAGE_CONTENT, name);
        database.insert(DBHelper.TABLE_MESSAGE, null, cv);
    }

    public Cursor readData() {
        String[] allColumns = new String[] { DBHelper.MESSAGE_ID,
            DBHelper.MESSAGE_CONTENT };
        Cursor c = database.query(DBHelper.TABLE_MESSAGE,
allColumns, null,

```

```

        null, null, null, null);
    if (c != null) {
        c.moveToFirst();
    }
    return c;
}

public int updateData(long memberID, String memberName) {
    ContentValues cvUpdate = new ContentValues();
    cvUpdate.put(DBhelper.MESSAGE_CONTENT, memberName);
    int i = database.update(DBhelper.TABLE_MESSAGE, cvUpdate,
        DBhelper.MESSAGE_ID + " = " + memberID, null);
    return i;
}

public void deleteData(long memberID) {
    database.delete(DBhelper.TABLE_MESSAGE,
DBhelper.MESSAGE_ID + "="
        + memberID, null);
}

} // outer class end

```

Database

```

package fyp.app.vdroid;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DBhelper extends SQLiteOpenHelper {

    // TABLE INFORMATION
    public static final String TABLE_MESSAGE = "message";
    public static final String MESSAGE_ID = "_id";
    public static final String MESSAGE_CONTENT = "content";

    // DATABASE INFORMATION
    static final String DB_NAME = "MEMBER.DB";
    static final int DB_VERSION = 1;

    // TABLE CREATION STATEMENT
    private static final String CREATE_TABLE = "create table "
        + TABLE_MESSAGE + "(" + MESSAGE_ID
        + " INTEGER PRIMARY KEY AUTOINCREMENT, "
        + MESSAGE_CONTENT + " TEXT NOT NULL);";

    public DBhelper(Context context) {
        super(context, DB_NAME, null, DB_VERSION);
    }
}

```

```
@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL(CREATE_TABLE);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
{
    // TODO Auto-generated method stub

    db.execSQL("DROP TABLE IF EXISTS " + TABLE_MESSAGE);
    onCreate(db);
}
}
```