HOME AUTOMATION SYSTEM WITH SPEECH RECOGNITION

LOGESWARAN A/L RAJASEGARAN

This report is submitted in partial fulfillment of the requirements for the Bachelor of Computer Science (Artificial Intelligence)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS TESIS

JUDUL: HOME AUTOMATION SYSTEM WITH SPEECH RECOGNITION

SESI PENGAJIAN: Semester 2 – 2016 / 2017

Saya, LOGESWARAN A/L RAJASEGARAN,

(HURUF BESAR)

mengaku membenarkan tesis (PSM/~~Sarjana/Doktor Falsafah~~) ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.

2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.

3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.

4. ** Sila tandakan (/)

__________ SULIT (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

__________ TERHAD (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

_____/_____ TIDAK TERHAD

R.loges	
(TANDATANGAN PENULIS)	(TANDATANGAN PENYELIA)
Alamat tetap: No 23, jalan jerangor, Taman Chi Liung, 41200, Klang, Selagor	Dr. Yogan a/l Jaya Kumar Nama Penyelia
Tarikh: 15/8/2017	Tarikh: 15/8/2017

CATATAN: * Tesis dimaksudkan sebagai Laporan Akhir Projek Sarjana Muda (PSM)

** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa.

DECLARATION

I hereby declare that this project report entitled

HOME AUTOMATION SYSTEM WITH SPEECH RECOGNITION

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : LOGESWARAN A/L RAJASEGARAN Date: 15/8/2017

I hereby that I have read this project report and found this project report is sufficient in terms of the scope and quality for the award of

Bachelor of Computer Science (Artificial Intelligence) with Honours

SUPERVISOR: DR. YOGAN A/L JAYA KUMAR Date: 15/8/2017

DEDICATION

I dedicate my final year project report to my family, lecturers of the Department of Intelligent Computing and Analytic and also my friends. I would also like to show my gratitude to my project supervisor, Dr.Yogan Jaya Kumar, for helping and guiding me throughout the project timeline. I would like to show my deepest gratitude to my parents for being supportive in many aspects while completing this project. My family members have been very encouraging while giving constructive ideas for this project. Finally, I would like to dedicate this project to all my friends who helped me regarding this project.

ACKNOWLEDGEMENT

I would like to show my gratitude to the following respected people in helping me to complete this project. Firstly, I would like to show my gratitude to my project supervisor, Dr.Yogan Jaya Kumar and my project evaluator Dr. Halizah binti Basiron, who have given me constructive comments, guidance and support throughout this project. Next, I would like to thank my parents for being supportive in many aspects to finish this project successfully. I would like to thank my course mates for helping me to test and solve the technical and logical issues regarding my system.

ABSTRACT

The topic of this project is home automation system with speech recognition. Home automation is a residential extension which involves the control and automation of lighting, heating, ventilation, air conditioning and security, as well as home appliances such as washer/dryers, ovens or refrigerators that uses WiFi for remote monitoring. Home automation is the next step of revolution, where it makes people's lifestyle easy and also convenient.

Nowadays most of the new houses are equipped with automation system. These houses also known as smart homes. A smart home or building is a home or building, usually a new one, which is equipped with special structured wiring to enable occupants to remotely control or program an array of automated home electronic devices by entering a single command.

This project's purpose is to build a prototype of a home automation system with lighting and control them using speech commands, such as "On the light" and "Off the light". Raspberry Pi 3 Model B, microphone, LED lights, breadboard and jumper wires will be used to design the prototype of home automation system with speech recognition

ABSTRAK

Topik untuk projek ini adalah system automasi rumah dengan pengecaman suara. Automasi rumah adalah lanjutan tempat kediaman yang melibatkan kawalan dan automasi pencahayaan, pemanasan, pengudaraan, penghawa dingin dan keselamatan, serta peralatan rumah seperti mesin basuh / pengering, ketuhar atau peti sejuk yang menggunakan WiFi untuk pemantauan jarak jauh. Automasi rumah adalah revolusi langkah yang seterusnya, di mana ia akan memudahkan gaya hidup rakyat.

Pada masa kini, kebanyakan rumah baru dilengkapi dengan sistem automasi. Rumah-rumah ini juga dikenali sebagai rumah pintar. Sebuah rumah pintar atau bangunan biasanya dilengkapi dengan pendawaian berstruktur khas bagi membolehkan penghuni untuk mengawal peralatan elektronik rumah dari jauh dengan memasukkan arahan.

Tujuan ini projek ini adalah untuk membina sebuah prototaip sistem automasi rumah dengan lampu dan mengawal lampu-lampu tersebut menggunakan arahan bersuara, seperti "On the light" dan "Off the light". Raspberry Pi 3 Model B, mikrofon, lampu LED dan wayar akan digunakan untuk mereka bentuk prototaip sistem automasi rumah dengan pengecaman suara.

TABLE OF CONTENTS

CHAPTER 6 TESTING

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Home automation system is widely used in many home nowadays. Home automation system is a residential extension which involves the control and automation of lighting, heating, ventilation, air conditioning and security, as well as home appliances such as washer/dryers, ovens or refrigerators that uses WiFi for remote monitoring. Home automation system makes people's life easier, saves time and also convenient. This project's aim is to develop a prototype of a home automation system with lighting and control them using speech commands. We can give commands in the form of speech to ontrol the appliances, for example, turing on and off the lighing using "on the light" command and "of the light" command respectively**.** Raspberry Pi 3 Model B, microphone, LED lights, breadboard and jumper wires will be used to design the prototype of home automation system with speech recognition. The user will speak the command through the microphone, after receiving the audio, it will be sent to the Google Speech recognition API where the API will convert the speech into written text (Python strings). If the text is matching with the commands which have been initialized, for example, "On the light", the LED light which is connected to the Raspberry Pi will light up.

1.2 Problem statement

Physically challenged people face a lot of problem in their daily life. The basic stuffs such as getting out of bed, using the rest room, turning on and off the appliances at home, drinking, and eating and so on. To relate them with this project, they face difficulties in controlling home appliances such as light, fan, air conditioner, television and also opening and closing the door. Therefore, it is big problem for these individual when it comes to moving around and do stuffs physically. They need the help of technologies to make their life become easier. Besides, not only for the physically challenged people, as there are more and more home appliances are increasing, people feel difficult in controlling those appliances physically. Physically controlling the home appliances is time consuming and also is not convenient.

1.3 Objective

- To design a home automation system prototype with speech recognition.
- To investigate whether the home automation system is very helpful and easier to be used by people by letting people to test the system.
- To assess the performance of the recognition of speech in this system.

1.4 Scope and Limitation

- This project will be implemented using Raspberry Pi 3 Model B as this hardware can be used program and deploy system by ustilizing it general purpose input/output(GPIO) pins.
- This project focuses on the control of home appliances using speech recognition, thus the set of keywords for the actions will be clustered based on their phonetics to make the system to detect the keywords accurately.
- The system needs internet connection as it is using the Google Speech Recognition API.

1.5 Project Significance

As the technology is blooming, home automation system is becoming more popular nowadays, which make the life people easy. We can control home appliances using our mobile phone, IR remote, computers, laptops and also tablets. However, physically challanged people will face difficulties in controlling home appliances by using these gadgets. Therefore, the development of this home automation system with speech recognition able to help them to control their house appliances easily by speech. Besides, this system will also be helpful for other people who would like to build their own smart home.

1. 6 Expected output

The expected result of the project is this home automation system will be functioning well with speech commands. Besides, it would also be easier to be used by people especially physically challanged person without restiction.

1.7 Conclusion

This chapter has made the introduction about the Home automation system with speech recognition. The reader able to know how this system is going to be implemented. Purpose and the expected outputs are determined so that the problem that might be occur in near future can be solved and the desired output can be achieved.

CHAPTER 2

LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

Many researches have been done for the home automation system, including hand gesture based home automation system for visually challenged, smart home automation system for elderly, and handicapped people, home automation system using android for mobile phone and voice recognition based home automation system for paralyzed people. As this project focuses on the home automation system with speech recognition, there are several methods used to identify speech from users and process the speech given by the user. The method's used are, application which utilize the Microsoft Speech Application Programming Interface (SAPI), Speech Recognition System which utilizes MATLAB to process the voice commands, CMU Sphinx, an open source toolkit for speech recognition and lastly the Google Speech Recognition API, which converts spoken text (microphone) into written text (Python strings).

Compared to all of these methods, the Google Speech Recognition API considered as the best and easiest method to be implemented, as it can be set up easily compared to Microsoft Speech Application Programming Interface (SAPI), it doesn't need the aid of a software program to process the voice command as the Speech Recognition System which utilizes MATLAB, which may consume a lot of memory in the computer, and it can detect words effectively than CMU Sphinx, as the Google Speech Recognition API has a large database of words online. Moreover, the Google Speech Recognition API can be embedded easily with Home Automation System which implements speech recognition.

2.2 Facts and Findings

2.2.1 Internet of Things (IOT)

Internet of Thing (IOT) is the inter-networking of connected devices, home buildings, and items which are connected with electronics parts and software, where these objects can collect and exchange data between one another with an aid of network connection. The Internet of Thing (IOT) allows the objects, such as software, electronic devices, sensors and actuators to be sensed or controlled remotely across existing network connection. This allows the integration between physical world and computer based systems, which will result in the improvement of the accuracy, efficiency and reduce human intervention.

The concept of Internet of Things (IOT) was discussed in the early 1980's, specifically in the year 1982, whereby a coke machine has been modified in the Carnegie Mellon University. This coke machine declared as the first internet connected appliance where the machine is capable of reporting its inventory about the status of the newly loaded drinks. Besides, the concept of the Internet of Things became popular in the year 1999, where Radio Frequency Identification (RFID) was declared as one of the prerequisite for the Internet of Things. After that, Internet of Thing (IOT) became popular where a lot of technologies were introduced.

The Internet of Thing (IOT) can be applied in many areas, such as media, environmental monitoring, infrastructure management, manufacturing, medical and healthcare, transportation and building and automation. In building and home automation application, Internet of Thing (IOT) devices can be used to control and monitor electrical and electronic devices used in various types of buildings such as industrial buildings- and residential.

2.2.2 Home Automation

The home automation system is a residential extensions which involves the control and automation of lighting, ventilation, heating, air conditioning, air conditioning, security and home appliances such as washer/dryers, ovens, refrigerators and television that uses WiFi for remote monitoring. These appliances can be control by using our smart phone and also by using speech commands, as long as there is a network connection. The concept of home automation system became popular in the year 1975, where the first home automation network technology was developed. It was a communication protocol for electronic devices where it uses electric power transmission wiring for signal and control. By the year 2012, home automation system is widely used in residential around the world.

There are many applications of home automation system, such as lighting control system, heating, ventilation and air conditioning (HVAC), security, occupancy aware control system and home automation for the elderly and disabled. The Lighting control system is the system where the lightings can be controlled via remote monitoring, which is used widely on both indoor and outdoor lighting of commercial.

Besides, the home automation system for the elderly and disabled focuses on making the life easier and comfortable for the older adults and people with disabilities at home. For the home automation system for elderly and disabled, one of the method implemented to control the home lighting and appliances is via speech, where a particular elderly or disabled person doesn’t need to move around to control the appliances at home. All they have to do is to give commands through speech to control those appliances.

2.2.3 Speech Recognition

Speech recognition can be stated as the ability of a program of a machine to identify words in spoken language and convert them to machine readable format. The speech recognition works using through acoustic and language modelling. The acoustic modelling can be stated as the relationship between speech and audio signals, whereas language modelling can be stated as the process of matching sounds with words to help distinguish between words and the similar sound.

The concept of speech recognition started in the early 1950s, specifically in the year 1952, where a system for a single speaker digit recognition was built. The technology was limited to single speaker only with vocabularies around ten words. In late 1900s, a commercially successful speech recognition technologies were introduced. At that particular time, the vocabulary of a commercial speech recognition system was larger than the average human vocabulary. In this 21st century, the speech recognition system improved well as there are many application program interface (API) for the speech recognition, which helps in building software applications with speech recognition. These API's are pre-trained with model, methods and algorithms to improve the accuracy of speech recognition.

To improve the accuracy of a speech recognition system, there are several models methods or algorithms can be used such as the hidden Markov models, Dynamic time warping (DTW) algorithm and also neural network. Based on Huang X.D. and Ariki Y. (1990), University of Edinburgh, the hidden Markov model are used to recognize temporal patterns in speech as it is a statistical model in which a particular system is being modelled is assumed to be a process with hidden state. The Dynamic Time Warping (DTW) algorithm is used for measuring the similarity between two sequences of words that may vary in time or speed. This algorithm is suitable to use when the sentence speak by a user is long and to be recognized. Besides, neural network also have been used in speech recognition to classify the phoneme, to recognize isolated words and adapt the speaker who gives command.

The process of conversion of a stream of acoustic signals into a sequence of words is known as the Automatic Speech Recognition (ASR). ASR systems can be used to control machines, find online contents and also it contributes to generate contents. Figure 1 below shows the block diagram of an ASR system.

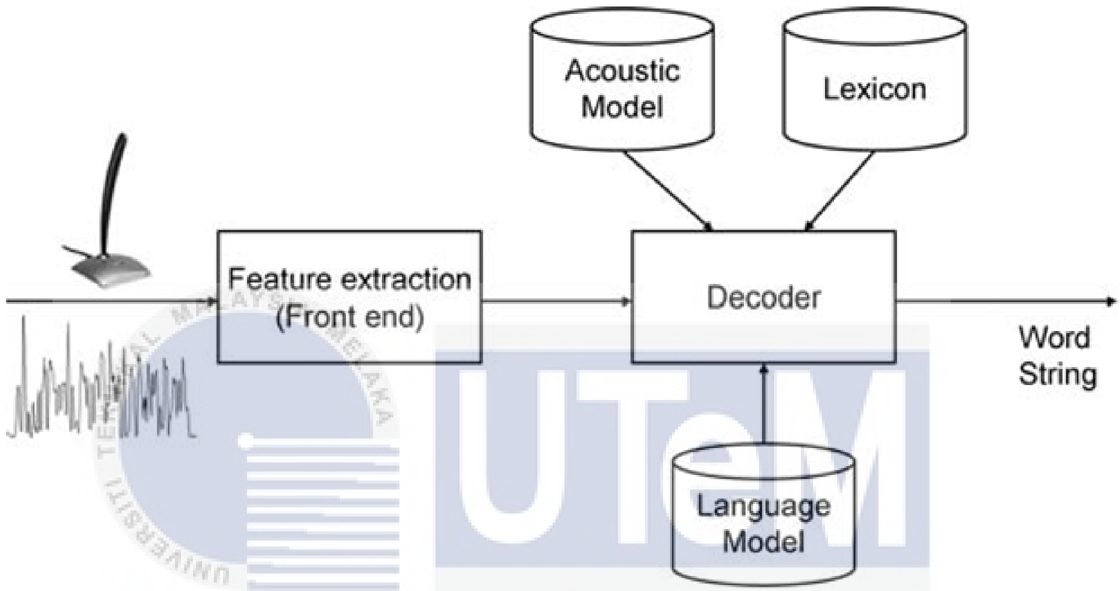


Figure 1: Block Diagram of an Automatic Speech Recognition (ASR) system

Based on Preeti Saini and Parneet Kaur, (2013), CSE Department, Kurukshetra University ACE, Haryana, India, a typical ASR system consists of two parts, which are the feature extraction (front end) and the decoder. When a user speaks or gives command to the microphone, the front end block will extract the sound spectrum which is in wave form. After extracting the sound wave, the decoder will search for the best matching of the words or commands spoken by the user from the Language model, Acoustic Model and also the lexicon. The language model consists of sequence words in various languages, such as English, French, German, Mandarin and so on. The acoustic model represent the relationship between the sound spectrum which captured in the front end block and the phonetics that make up a particular speech. Besides, Lexicon consists word vocabularies.

After searching for the best matching of words, the system will return the words in string form.

2.2.4 Phonetics

A phonetic algorithm is an algorithm that is used index words by their pronunciation. The algorithm has the ability to match two different words with same pronunciations to the same code. This allows phonetic similarity based word to set comparison and indexing. Based on Donald Fenna (2003), Computer Programs in Biomedicine, Vol 19, issue1, pp 31-36, a phonetic algorithm is useful in searching list of people in database and also useful in a spelling checker. These algorithms are usually used in combination with the algorithms of fuzzy search, which provides the users with a handy search by name. One of the well-known phonetics algorithm is the Soundex algorithm.

2.2.5 Soundex

Soundex is one of the earliest algorithm, invented in the 1910s by Robert Russell. Soundex is one of the fuzzy matching algorithm, where it matches words to the numerical index like S134, where the first letter of a particular word is stored and the subsequent letters are matched to digits, indicating different categories of sounds created by consonants following the first letter. The working principle of Soundex is based on the partition of consonants in groups with ordinal numbers, which are then compiled to the resulting value. One of the example of Soundex is, the word Washington is decoded as “W252”. Besides, certain words which are spelled different may have similar sound. Thus, these words will have the same Soundex representation. For example, the word GAUSS and GHOSH will have the same Soundex representation, which is “G200”, since the sound of both words are similar. Based on Alexander Beider & Stephen P.Morse (2010), Association of professional Genealogists Quarterly, Soundex is very helpful to identify the matching of surname of certain people without any problem, for example, “Schmidt” and “Smith”, since both have similar sound and pronunciation.

2.3 Existing System

Voice recognition based home automation system has been implemented by Mukesh Kumar, Shimi S.L in 2015 in the proceedings of International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE). The voice recognition based home automation system focuses mainly on paralyzed people. The proposed technique of speech recognition is, the Elechouse voice recognition module v3 was used for the voice recognition process. The speech signal is compared with the previous stored trained voice examples, after the voice recognition module receive the speech input from microphone. If there is a match between the speech input and the trained voice commands, the action for the command will take place. The voice recognition module v3 can store up to 80 commands in its library. But, out of the 80 commands, only 7 can be loaded into the recognizer for the recognition process. As only 7 commands are effective at a time, the existing commands has to be cleared in order to add another 7 new commands. This will be one of the constraints of this system.

Another existing system proposed by B.Mardiana, H. Hazura, which entitled “Home Appliances Controlled Using Speech Recognition in Wireless Network Environment”, in the proceedings of ICCTD International Conference on Computer Technology and Development. The proposed system is where it can control devices from large distance, as it utilizes WiFi. The voice command from user is converted into the symbols and they are sent to the server computer over internet connection. The computer is installed with voice recognition application developed with Microsoft Visual Basic. When the command is recognized successfully, the information is transferred to the control circuit, at the particular action is performed.

2.4 Project Methodology

The aim of this project is to develop a prototype of home automation system with speech recognition. Google Speech Recognition API is used to control the LED lights that are connected with the Raspberry Pi 3 Model B. The Google Speech Recognition API converts spoken text into written text, in other words, speech to text. We can speak via microphone and the Google Speech Recognition API will translate it into written text. First, the user will give the command to the microphone which is connected to the Raspberry Pi 3 Model B. That specific command will undergo preprocessing method and it will be sent to the Google Speech Server to recognize the word spoken by the user. After the recognition process, the word spoken by the user will be converted into text and it will be sent to the python program that has been coded in the Raspberry Pi 3 Model B. The converted text will be compared with commands in the python program, and if the converted text matches with the command, the LED light will perform its action, either to light up or turn itself off. Figure 2 below shows the architecture of the system.

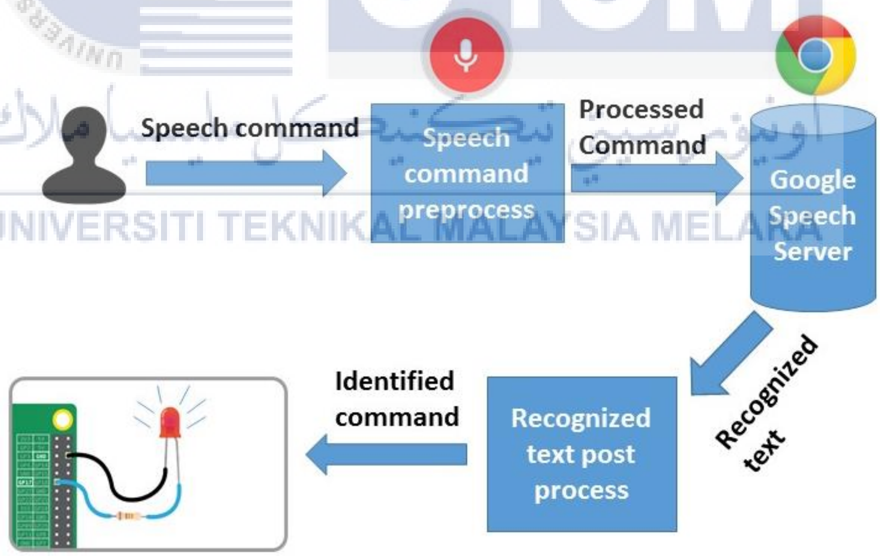


Figure 2: System architecture of Home automation system with speech recognition

2.5 Project Methodology model

The waterfall model is implemented for this project's methodology, in which the progress will be seen as flowing steadily downwards throughout the phase of requirements, design, implementation, testing and maintenance. In the requirement phase, we will finalize on what is needed in order to develop the prototype of Home Automation System with Speech Recognition. Raspberry Pi 3 Model B is the most important hardware which is required for this project. Besides, microphone and LED lights are also required to design the hardware. Besides hardware, a library for performing speech recognition is also required such as the Speech Recognition Library for Python.

The second stage is designing the prototype. The prototype includes the rough idea on how the system works and all the functions that are about to be included in the system. On the hardware side, the Raspberry Pi 3 Model B will be connected with the LED light using jumper cables and a breadboard. Besides, the microphone also will be connected via the USB port of the Raspberry Pi 3 Model B. In the software side, the library for speech recognition will be installed into the Raspberry Pi 3 Model B, in order to perform speech recognition. After the installation, the design of the program is done, where the system is programmed in python language. The commands and keywords will be coded in the program. So, when a user speaks to the microphone, it will look for the keyword or the command in the program, and will execute the initialized action.

The third stage is implementation where the implementations of the projects is done in this stage. In this stage, the end user of this home automation system with speech recognition will be taken into consideration. During this phase, the documentations will be crated for the customer, which will be helpful for them during the usage of this system. The fourth stage will be the testing phase. In this phase, all the testing of the system will be done. For example, testing is done based on the speech recognition, where we speak through the microphone and wait for the system to perform the action based on the command. Besides, the testing also done in the hardware part, where we test out the LED lights to make sure those light are working perfectly without flaw.

The final stage will be the maintenance phase. In this phase, making changes to the hardware and software will involve. It also includes making changes to improve a system's performance, correct problems, and enhance the feature of the system, such as add more commands which can be embedded in a real home in the future. Figure 3 below shows the Waterfall Model Diagram of the system.

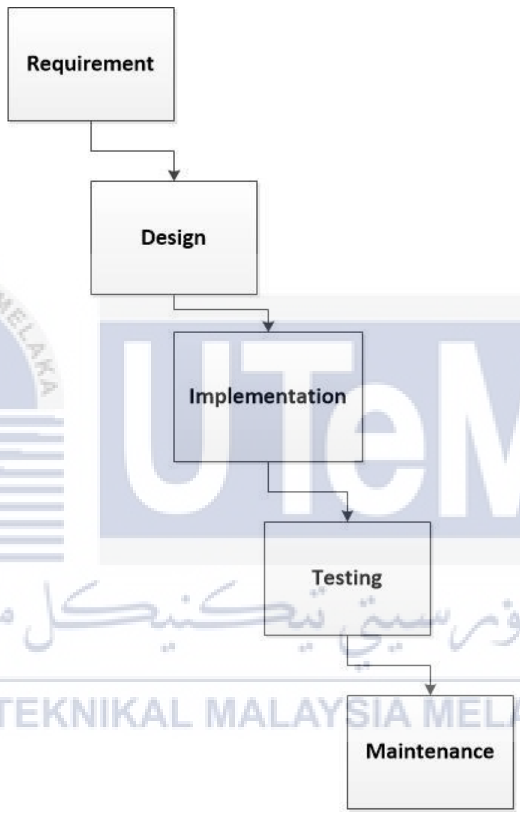


Figure 3: Waterfall Model Diagram of Home automation system with speech recognition

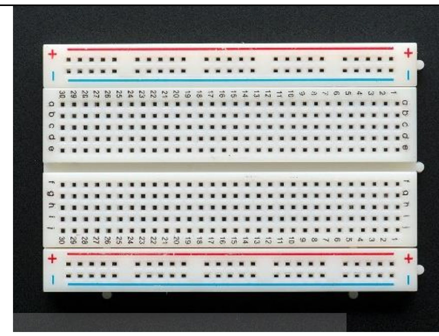
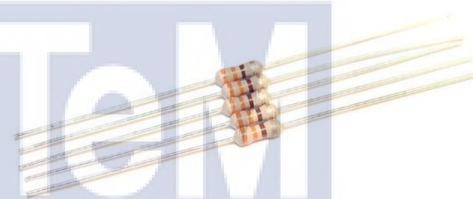
2.6 Project Requirements

2.6.1 Hardware Requirements

Table 1 below shows the hardware requirements of the system. These are the important hardware which are used in this project.

Table 1: Hardware requirements

Hardware	**Picture**
• Raspberry Pi 3 Model B	
• USB Microphone	
• LED lights	

• Jumper wires	
• Breadboard	
• Resistors	

2.6.2 Software requirements

- Python version 2.7
- Python Speech Recognition version 3.6.5
- PyAudio version 0.2.9
- Phonetics version 1.0.4
- Soundex version 1.1.3

2.7 Project Milestone

Project milestone is a tool used to mark specific points throughout a project timeline. It includes the project start, end date and also the task that should be carried out throughout the project. Table 2 below shows the Gantt Chart of the project where is shows the tasks completed in the period of time, which is from the beginning of the semester to the end of the semester.

Table 2: Gantt Chart of the overall project

Task	Feb 21-27	Mac 4-10	Mac 11-17	Mac 18-24	Mac 25-31	Apr 1-7	Apr 8-14	Apr 15-21	Apr 22-28	May 1-7	May 8-14	May 15-21	June 1-14
Proposal confirmation													
Correction on proposal													
Studies on Literature Review													
Work on chapter 1													
System development													
Work on chapter 2,3,4													
Work on testing and results													
Finalizing chapter 5,6,7													
Submission of chapter 1,2,3,4													
System demonstration													
Submission of final report													

2.8 Conclusion

This chapter focused on the literature review of the, existing system of the project and also the methodology. The literature review focused on the history of the related techniques used in this system and previous research. The existing system was discussed on the similar system that has been done which is related to this project title. Finally, the project methodology focused on the methodology used for the development of the project and also the Gantt Chart of the system which includes the milestones of the project that are about to be completed throughout the time frame of the project.

CHAPTER 3

ANALYSIS

3.1 Introduction

The analysis phase is the phase where the lifecycle of the project starts. The requirements of the projects are identified and discussed into more detailed project requirements. In this chapter, two important things will be discussed regarding the analysis of the system, which are the problem analysis of the system and the requirement analysis of the system. Firstly, the problem analysis of the system will be discussed, which includes the process of understanding the needs of the user in this system. There are five steps included in the problem analysis, first is define the problem statement of the system. The next step is to identify the root cause of the problem. Third step will be identify the target users, and the fourth step is to define the system boundary. The last step will be identifying the enforcement of the system. After the discussion of the problem analysis, the requirements analysis of this system will be discussed. There are several requirements need to be considered, which are functional requirement, non-functional requirement, data requirement and other requirements.

3.2 Problem analysis

Physically challenged people face a lot of problem in their daily life. The basic stuffs such as getting out of bed, using the rest room, turning on and off the appliances at home, drinking, and eating and so on. To relate them with this project, they face difficulties in controlling home appliances such as light, fan, air conditioner, television and also opening and closing the door. Therefore, it is big problem for these individual when it comes to moving around and do stuffs physically. They need the help of technologies to make their life become easier.

Besides, not only for the physically challenged people, as there are more and more home appliances are increasing, people feel difficult in controlling those appliances physically. Physically controlling the home appliances is time consuming and also is not convenient. So to solve this problem, home automation with speech recognition will be one of the best solution, as people and also physically challenged individuals can just control home appliances by giving command through speech. So the target users for this system will be the physically challenged individuals and also the people who would like to have a smart home. The home automation with speech recognition system will be very helpful and easier to be used by the people especially physically challenged people, as it will save time and increase the convenience of use. Figure 4 below shows the flowchart of the Home automation with speech recognition system prototype.

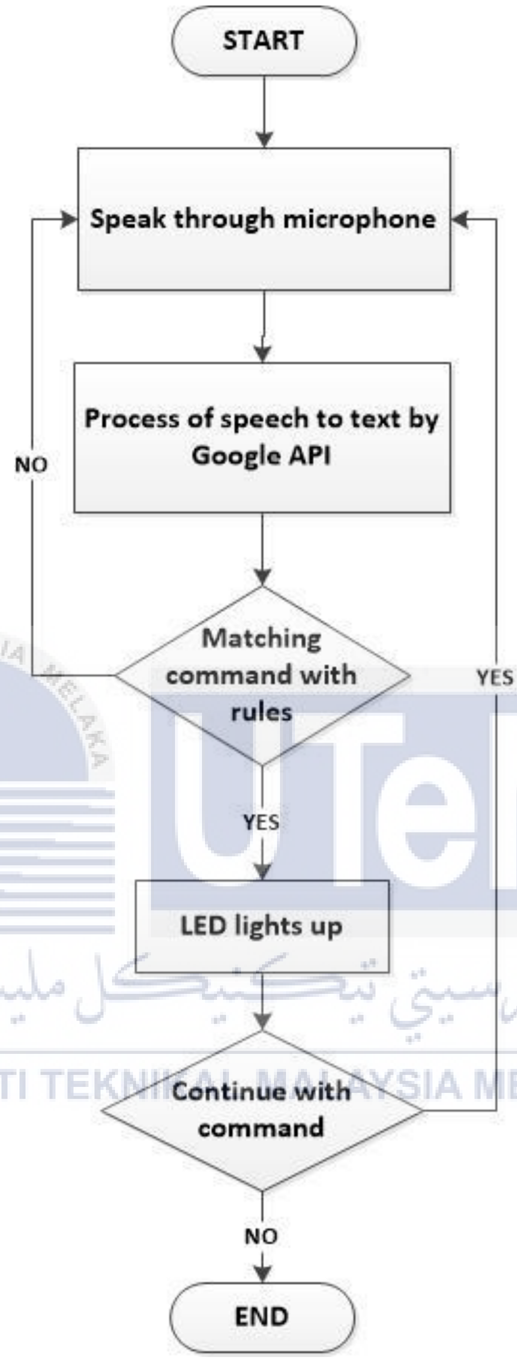


Figure 4: Flowchart of the Home automation with speech recognition system prototype

3.3 Requirement Analysis

3.3.1 Data Requirement

In this project, for the system to recognize the correct command from the user, the system should have a program with the keywords of commands. For example, for turning on the LED, the "on" command should be included in the program, and also for the turning off the LED, the "off" command should be included. Besides, as we are going to control home appliances, the name of the home appliances should also be included in the program such as "light", "fan" and also "radio". The user can speak to the microphone with only commands or with sentence, as the program will identify the keywords presented in those commands or sentence, and will perform the action to be done. As many people have different slang, their pronunciations of certain words will be different. So, the system might not be able to recognize the correct keyword even though the user gives the correct command. To overcome this problem, the concept of homonym was used. A homonym is a word that is said or spelled the same way as another word but has a different meaning. For example, "Light" and "Lite". So there will be a lot of homonym keywords included in order for the system to perform well.

3.3.2 Functional Requirement

Functional requirement is any requirement which specifies what the system should do. The functional requirement for the Home automation system with speech recognition prototype is the ability to perform set of actions, such as turning on or turning off the LED lights when the user gives the specific commands for those actions through the microphone.

3.3.3 Non Functional Requirement

Non-functional requirement is any requirement which specifies how the system performs a certain function. The non-functional requirement for this system is, the accuracy of the system to detect the commands. In order to obtain and analyze the accuracy of the system to detect the commands, 3 users will be testing the system by giving command and the accuracy will be recorded.

3.3.4 Other requirements

Software requirements

- Python version 2.7
- Python Speech Recognition version 3.6.5
- PyAudio version 0.2.9
- Phonetics version 1.0.4
- Soundex version 1.1.3

3.4 Conclusion

Analysis is the process of understanding the needs of the user in this home automation with speech recognition system. The Understandings of the problems faces by people and also the physically challenged individuals must be clear. Besides, the problems stated also must meet the objective of this system, before developing this system. Besides, all the requirements, including functional requirement, non-functional requirement, data requirement and also other requirement should be clearly specified before developing this system.

CHAPTER 4

DESIGN

4.1 Introduction

The system design and the functions will be explained in detail in this chapter. System design such as system architecture, hardware design and data of voice commands included in this system will be described based on high-level design.

Design phase is one of the critical stage in a product development. It utilizes the requirements mentioned in the analysis chapter and mapped into the system architecture. It will decide how the system will operate by involving several functions at the system requirements such as functional and non-functional requirements. The clear blueprint of the system will be presented in this chapter.

4.2 High Level Design

High level design is the overall system design which describes the relation between functions. It will mainly cover the system architecture. The system architecture diagram will explain us the overview of the entire system by identifying the main components that developed for the product. The diagram in figure 4.1 below describes the functionality of the system. Figure 5 below shows the architecture of the home automation system with speech recognition.

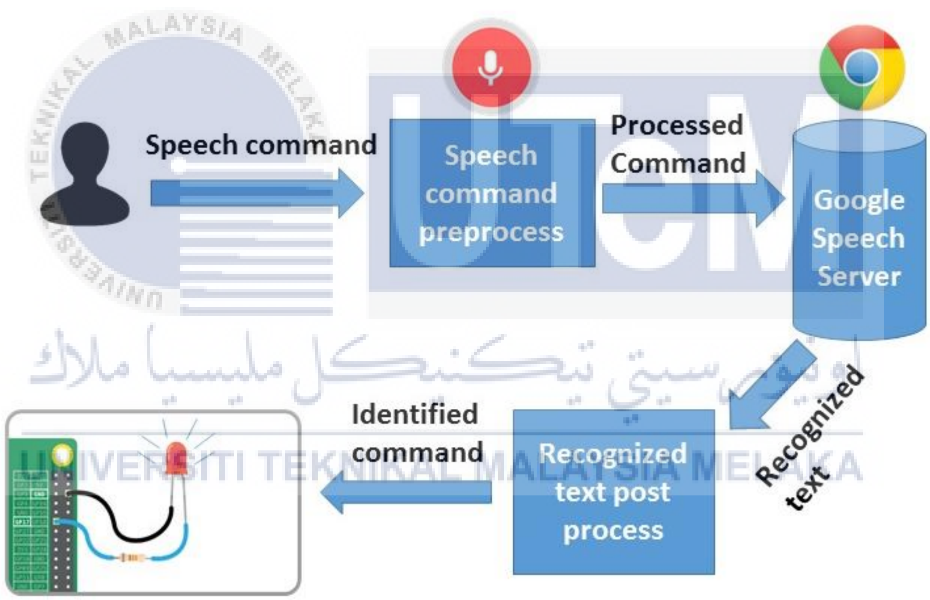


Figure 5: System architecture of Home Automtion System With Speech Recognition

4.2.2 Data of Voice commands

As mentioned in the data requirement at Chapter 3, for the system to recognize the correct command from the user, the system should have a program with the keywords of commands. The user will speak to the microphone which is connected to the Raspberry Pi 3 Model B with only command or sentence, as the program will identify the keywords presented in those command or sentence, and will perform the action to be done. Besides, the system might not be able to recognize the correct keyword, as many people have different slang. To overcome this problem, the concept of phonetics is used. The words which are sound similar or spelled the same way as another word, are clustered into the same group, so the system will generalize the words as one keyword which has been declared. Table 3 below shows the list of phonetics used based on the keywords. The keywords used in this system are, “on”, “off”, “light”, “fan” and “radio”.

Table 3: List of phonetics for each keywords

"light"	**"fan"**	**"radio"**
light	van	radio
like	pan	radial
black	pen	video
late	hang	meteor
lite	man	-
later	ban	-
lake	plan	-
line	been	-
laid	dance	-
life	fence	-
night	stand	-
right	Ben	-
lights	friend	-
write	and	-
-	ban	-
-	bend	-
-	fend	-
-	end	-
-	then	-
-	hand	-

4.2.3 Hardware Design

A hardware design is an abstract representation of an electronic device that is capable of running a fixed or changeable program. A hardware design model helps hardware designers to understand the way that the components fit in to a system architecture and provides the vital information to the software component designers for software development and integration. The hardware used for this project are Raspberry Pi 3 Model B, LED lights, breadboard, resistors, male-to-female jumper wires and also microphone. The LED lights and resistors are connected with the breadboard. For the current to flow from the Raspberry Pi Model 3 B to the LED lights through resistors, male-to-female wires are used to connect from the general purpose input/output (GPIO) pins on Raspberry Pi Model 3 B to the breadboard. Finally, the Microphone is connected to the USB port of the Raspberry Pi 3 Model B. Figure 6 below shows the hardware design of the system.

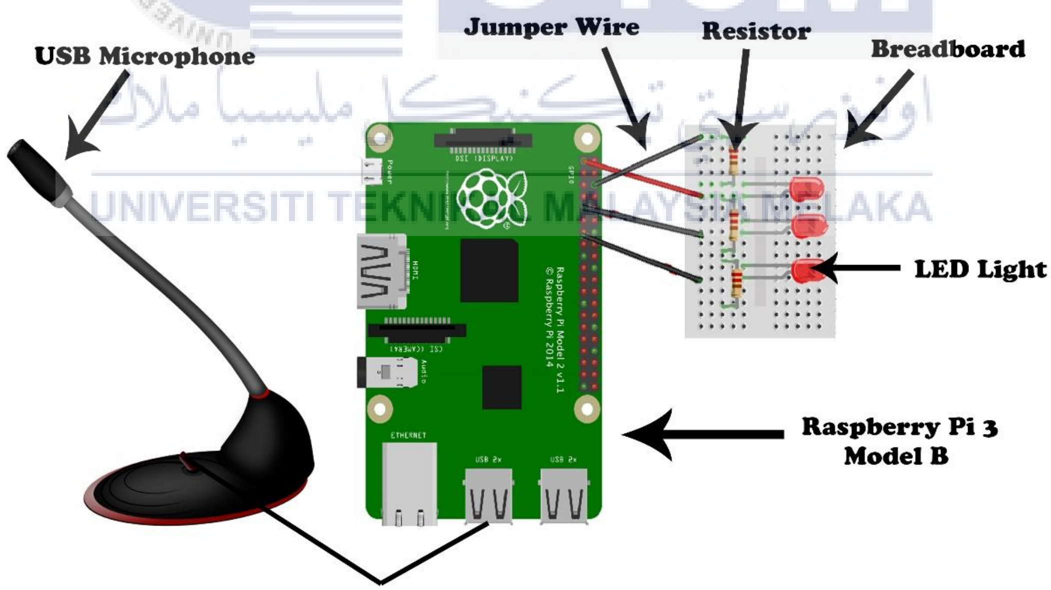


Figure 6: Hardware Design of Home Automtion System With Speech Recognition

4.3 Conclusion

This chapter provides an overview of the system, which includes the system architecture, data of the voice commands used alongside the homonyms, and also the hardware design of the system. The system architecture is vital as the user will understand the flow of the system. Besides, by clustering the homonyms of the commands used in this system, the system can perform effectively as it generalizes the homonyms into one keyword. The users will also would not face difficulties in using this system.

CHAPTER 5

IMPLEMENTATION

5.1 Introduction

The implementation of the system will be further discussed in this chapter. Implementation is the last process before the testing process, where all the information will be integrated from designing phase to production phase. The aim of this project is to implement speech to control the prototype of the home appliances, which are LED lights, with the aid of Google speech recognition API and also the phonetics python package to compute phonetics key of words spoken for indexing and fuzzy matching. To compute the phonetics key of words spoken, Soundex algorithm will be used as it will extract the sound of a word spoken and return it into string form, which will be used for decision making of the system.

5.2 Software Development and Environment Setup

This system will be developed using python programming language, as there are speech recognition packages exists, which can be used to develop an application which utilizes the speech recognition feature. The table 4 below shows the software and packages required to setup the speech recognition environment.

Table 4: Software and Packages for Speech Recognition Environment Setup

Software and packages	Version Required
Python	Version 2.7
Python Speech Recognition	Version 3.6.5
PyAudio	Version 0.2.9
Phonetics	Version 1.0.4
Soundex	Version 1.1.3

The steps for the installation of the software and packages are shown below.

- For Python Speech Recognition, type sudo pip install SpeechRecognition in the terminal.
- For PyAudio, Link to download: https://pypi.python.org/pypi/PyAudio/0.2.9
- For Phonetics, type sudo pip install phonetics in the terminal
- For Soundex, Link to download: https://pypi.python.org/pypi/soundex

Note: Python version 2.7 is pre-installed in the Raspbian Operating System of the Raspberry Pi.

Figure 7 below shows the Deployment View Diagram of the system

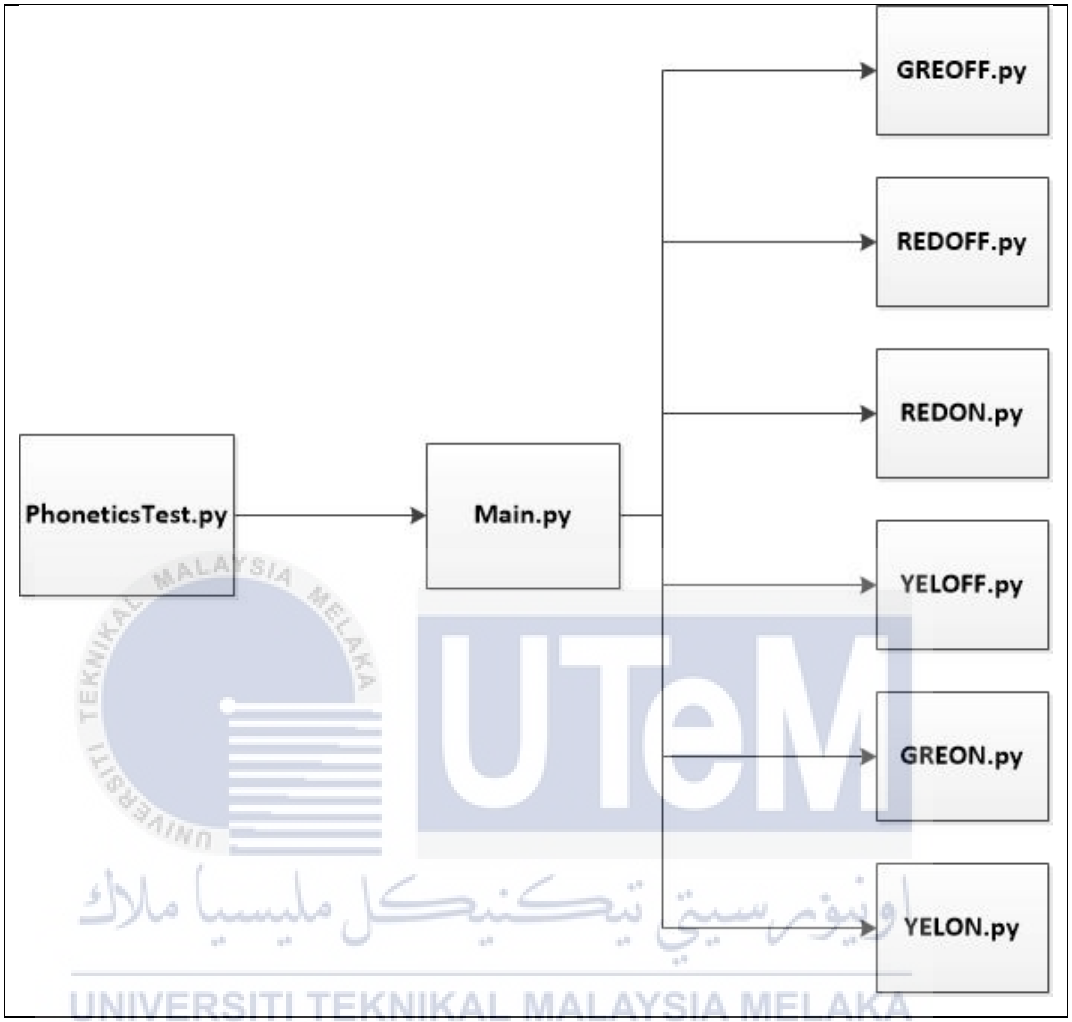


Figure 7: Deployment View Diagram

The phoneticsTest.py file contain the set of words which are similar in sound with the words of the home appliances used in this system (Light, Fan and TV). These set of words will undergo pre-processing phase, where the Soundex Algorithm will extract the sound of each words and return it in string form. Next, The Main.py file contains the code of the overall program, where it will receive the command spoken by a user. That specific command will undergo pre-processing method and it will be sent to the Google Speech Server to recognize the word spoken by the user. After the recognition process, the word spoken by the user will be converted into text and it will be returned to the program. Next, the text will be converted into string form using Soundex Algorithm and that string form will be compared with the string forms which have been returned in the PhoneticsTest.py file. Based on that, the decision will be made in order to turn on or off the LED lights which are connected to the raspberry pi. GREON.py file will turn on the green light, REDON.py file will turn on the red light, YELON.py file will turn on the yellow light, whereas GREOFF.py file will turn off the green light, REDOFF.py file will turn off the red light and YELON.py will turn off the yellow light.

5.3 Software Configuration management

Software configuration management is the work of tracking and regulating changes in software, which involves revision control and establishment of baseline, as the project progresses. Whenever there is a change in the process, the change has to be reported and recorded to make sure the status record of all modules in the system is well maintained.

To develop this home automation with speech recognition system, it involves a list of software and also packages to be installed in the Raspberry Pi. Since this system is developed using Python programming language, all the python packages which can support speech recognition is required, in order to implement the speech recognition feature.

5.3.1 Version Control Procedure

Throughout the time of the project, the documents and the program being developed may change over time. In order not to lose any important data, we have to keep the change in documents and program in record. It is vital that the changes in the system should be recorded so that we can have the most up to date version of a document. Thus, the redundant or draft version of documents can be deleted. Table 5 below shows the version control procedure of the project.

Table 5: Version Control Procedure

Type	Version
Draft Proposal	0.1
Finalized Proposal	0.2
Project report with minor change	1.0
Project report with major change	1.1
Finalized Project report	2.0

5.3.1.1 Backup Management

A backup management is the process of copying or archiving of data in a computer, so it may be used to restore the data if there is a data loss event. It is important to back up the files which includes the codes of the program, the documents of the project and also other important data which is related to the project. By doing this, we don't have to worry about the loss of data and also the corruption of the codes of the program. We can always restore back the original data. Besides, we should also backup frequently in order to keep the data up to date. Table 6 below shows the backup management of the project.

Table 6: Backup Management

Backup	Activity
Storage Scope	All
Backup Media	External Hard Disk & USB drive
Backup Frequency	weekly
Checking Backup Frequency	Once in 2 weeks

5.4 Implementation Status

The implementation process begins when a proposed technique or solution is approved. The implementation process involves planning, executing and deploying changes of the system. The planning phase involves the plan on setting the right decision for the system which satisfies the condition and the output of the system, which is to turn on or off the LED light. The execution phase involves testing of all the decisions to achieve the best result. Besides, changes will be done if the decisions didn't affect the output of the system. Figure 8 below shows the activity flowchart for the implementation phase.

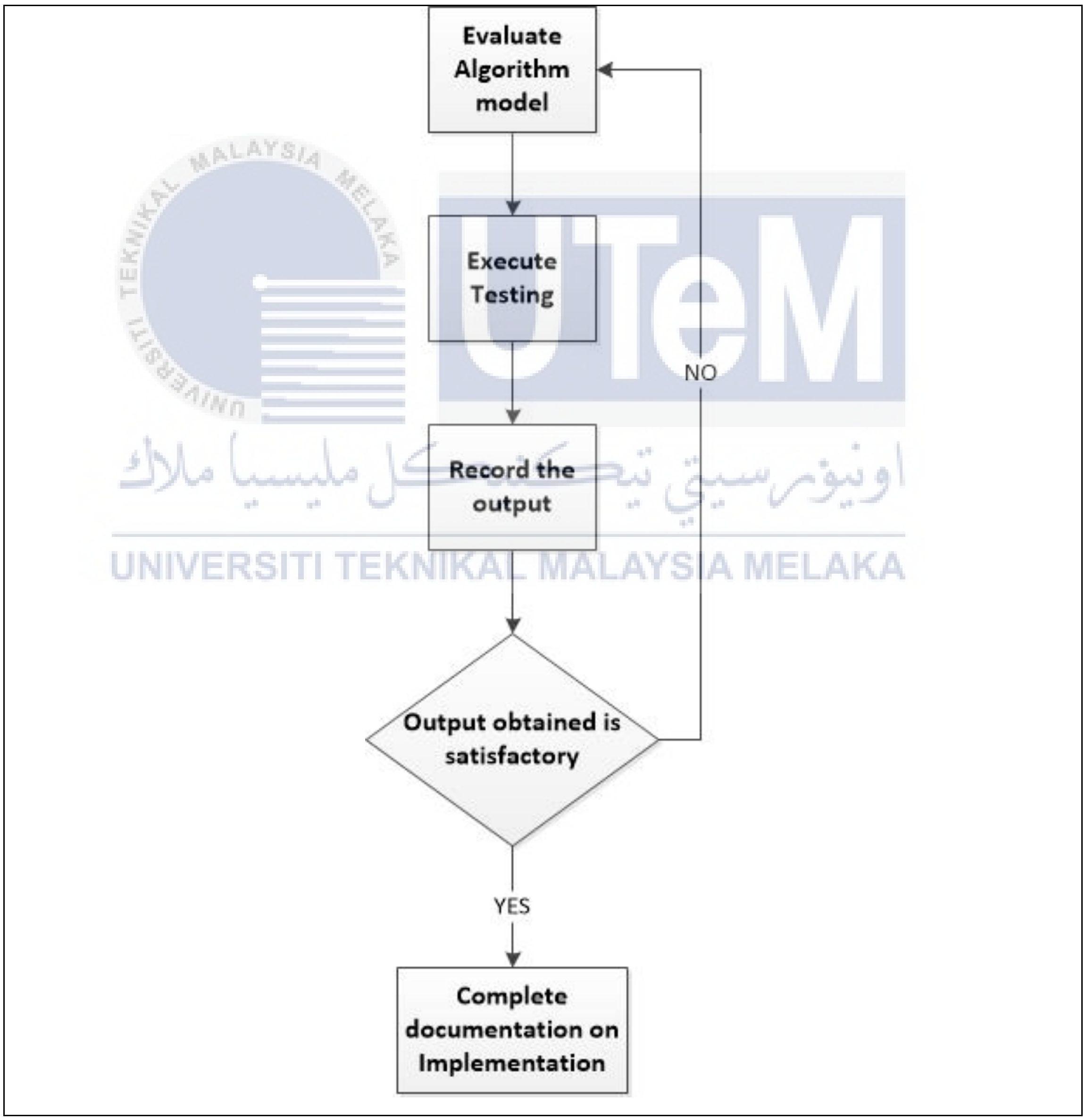


Figure 8: Flowchart for Implementation Phase

5.5 Conclusion

The main activity of the implementation phase is installing the system in the target environment. A lot of testing is required for the implementation of the speech recognition in the system. Besides, the version control is vital because it helps us to keep track on the changes that have been made on the system and also it serves as an evidence for all the progress that have been carried out throughout the project.

CHAPTER 6

TESTING

6.1 Introduction

Testing is the process to satisfy the requirements of the project. The program is executed in order to check whether it meets the project requirements. Besides, testing also is done in order to detect whether there are faulty codes which can affect the failure of the system. There are 2 types of tests, which are static test and dynamic test. Static test is where we carry out the test by finding defects without executing the code while dynamic test is where we carry out the test by finding defects by executing the code. As we are about to test the speech recognition of the system, dynamic test is used, so that we can obtain the optimal outcome for the speech recognition.

6.2 Test plan

6.2.1 Test environment and implementation

The system will be implemented and tested using Raspberry Pi with Raspbian operating system, where the operating system comes with python 2.7. So it will be easier to code in the raspberry pi and also easier to download the python speech recognition packages. Besides, the system will be tested in a closed quiet environment, so that there will be no interference when giving command to the microphone, as the microphone will record all the sounds in surrounding environment. Figure 9 below shows the hardware configuration of the system.

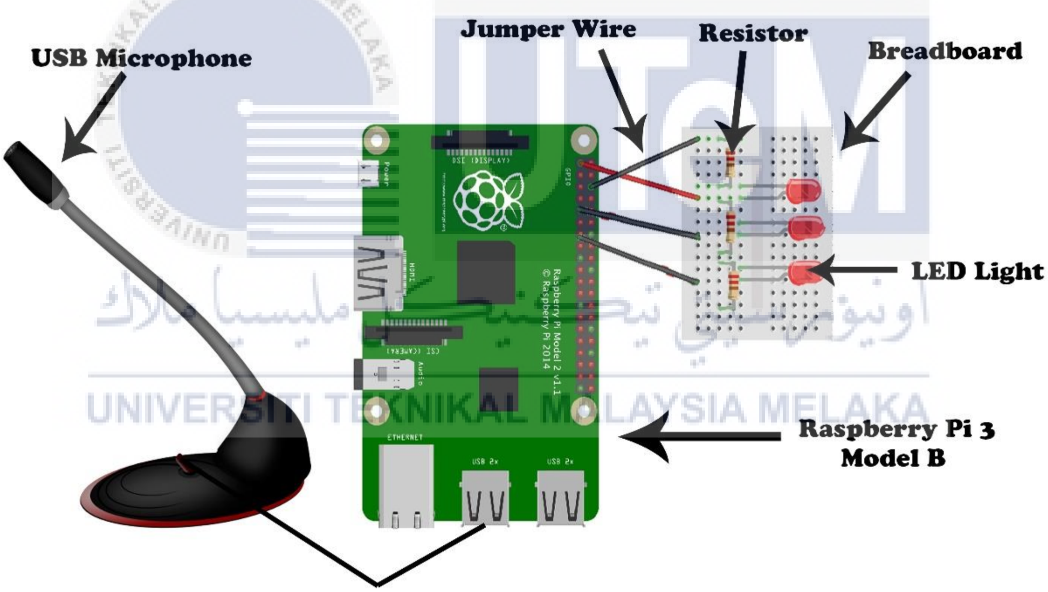


Figure 9: Hardware configuration of the system

6.2.2 Test Schedule

Each of the commands will be tested for 25 times. The user will give command by speaking to the microphone. The commands that can be used in this system are “light on”, “light off”, “on light”, “off light”, “fan on”, “fan off”, “on fan”, “off fan”, “radio on”, “radio off”, “on radio” and “off radio”. Upon testing each of these commands for 25 times, the accuracy of the system to detect these commands correctly, will be recorded.

6.3 Test Strategy

6.3.1 Dynamic strategy

Dynamic test strategy is used in this project as this dynamic strategy helps us to find defects easily during the speech recognition process. Dynamic strategy helps to identify defects before the program is coded completely. Besides, this strategy also allows the programmer to find out the defects before proceeding to another function of the program.

6.3.2 Variable coverage

Variable coverage is a measure of all variables that are used in the program and the variables must not overlap with another variable. In this project, there are many variables used, such as for the array stores, extraction of words from array and variables for the extraction of sound in string form. Each variables declared are unique and did not overlap with another variable, as conflict will happen in the system is the variables are not unique, and this will lead to a faulty system. Figure 10 below shows a part of code of the overall program, which have unique variables.

```
try:
    recognised=r.recognize_google(audio)
    split = recognised.split()
    print split
    index1 = [0]
    index2 = [1]
    q=[split[x] for x in index1]
    w=[split[x] for x in index2]
    z= " ".join(str(x) for x in q)
    c= " ".join(str(x) for x in w)
    print z
    print c
    j = phonetics.soundex(z)
    k = phonetics.soundex(c)
    print j
    print k
    m = [j,k]
    print m
```


Figure 10: Variables used in the program

6.4 Test Implementation

6.4.1 Test Description

The test activities of this project, which is giving commands by the user, will be recorded to compare which commands are easier to be detected and understood by the system. Figure 11 below shows the example of expected result for the command "light on" or "on light". The '102032' and 'o500' are the extracted sound in string form. Based on these string form, decision will be made by the system to turn on or turn off the LED light.

```
speak!
Recognised!
l02032
o500
['l02032', 'o500']
turning on the light
```

Figure 11: Example of expected result for "light on" or "on light" command

Figure 12 below shows the example of expected result for the command "fan on" or "on fan".

```
speak!
Recognised!
h052
o500
['h052', 'o500']
turning on the fan
```

Figure 12: Example of expected result for "fan on" or "on fan" command

Figure 13 below shows the example of expected result for the command “radio on” or “on radio”.

```
speak!
Recognised!
v030
o500
['v030', 'o500']
turning on the radio
```

Figure 13: Example of expected result for “radio on” or “on radio” command

Figure 14 below shows the example of expected result for the command “light off” or “off light”.

```
speak!
Recognised!
l02032
o100
['l02032', 'o100']
turning off the light
```

Figure 14: Example of expected result for “light off” or “off light” command

Figure 15 below shows the example of expected result for the command “fan off” or “off fan”.

```
speak!
Recognised!
f053
o100
['f053', 'o100']
turning off the fan
```

Figure 15: Example of expected result for “fan off” or “off fan” command.

Figure 16 below shows the example of expected result for the command “radio off” or “off radio”.

```
speak!
Recognised!
r030
o100
['r030', 'o100']
turning off the radio
```

Figure 16: Example of expected result for “radio off” or “off radio” command.

6.4.2 Test Data

Table 7 below shows the set of data of phonetics used based on the keywords. The keywords used in this system are, “light”, “fan” and “radio”.

Table 7: Phonetics of the keywords

“light”	“fan”	“radio”
light	van	radio
like	pan	radial
black	pen	video
late	hang	meteor
lite	man	-
later	ban	-
lake	plan	-
line	been	-
laid	dance	-
life	fence	-
night	stand	-
right	Ben	-
lights	friend	-
write	and	-
-	ban	-
-	bend	-
-	fend	-
-	end	-
-	then	-
-	hand	-

Each word in the table above will undergo preprocess phase, where the sound of each words will be extracted and will be returned in string forms. Unittest module in python will be used in this case, so that it is easier to detect bugs in the code in early stage. Figure 17 below shows the extracted sound of the words in class “light” which returned in string form.

```
Python 2.7.9 (default, Sep 17 2016, 20:26:04)
[GCC 4.9.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
For class light:
l0203
l020
b402
l030
l030
l0306
l020
l050
l030
l010
n0203
r0203
l02032
w6030


.
----------------------------------------------------------------------
Ran 1 test in 0.144s

OK
>>> 
```

Ln: 28 Col: 4

Figure 17: Extracted sound of the words in class “light”

Figure 18 below shows the extracted sound of the words in class "fan" which returned in string form.

```
Python 2.7.9 (default, Sep 17 2016, 20:26:04)
[GCC 4.9.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
For class fan:
f050
v050
p050
p050
h052
m050
b053
p405
b050
d0520
f0520
s3053
B050
f6053
a530
b050
b053
t050
f053
h053
e530


.
----------------------------------------------------------------------
Ran 1 test in 0.193s

OK
>>> |
```

Figure 18: Extracted sound of the words in class "fan"

Figure 19 below shows the extracted sound of the words in class "fan" which returned in string form.

```
Python 2.7.9 (default, Sep 17 2016, 20:26:04)
[GCC 4.9.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
For class Radio:
r030
r0304
v030
m0306


.
----------------------------------------------------------------------
Ran 1 test in 0.059s

OK
>>> |
```

Figure 19: Extracted sound of the words in class "fan"

Table 8 below shows the set of data of extracted sound in string form, which are similar to the keywords used in this system. These set of data will be used for the decision making of the system in turning on and off the LED lights.

Table 8: Data of extracted sound in string form

"light"	**"fan"**	**"radio"**
l0203	f050	r030
l020	v050	r0304
b402	p050	v030
l030	h052	m0306
l0306	m050	-
l020	b053	-
l050	p405	-
l010	b050	-
n0203	d0520	-
r0203	f0520	-
l02032	s3053	-
w6030	B050	-
-	f6053	-
-	a530	-
-	t050	-
-	f053	-
-	h053	-
-	e530	-

6.5 Test Result and Analysis

The test of the system has been performed by three individuals, including myself. The testers will be testing each commands for 25 times in the system, and the accuracy of the system to detect these commands correctly, will be recorded. The test includes before the implementation of Soundex Algorithm and also after the implementation of Soundex algorithm. Table 9 below shows the testers' identification.

Table 9: Testers' Identification

	Name	Age	Relationship
Tester 1	Logeswaran a/l Rajasegaran	23	Myself
Tester 2	Tan Eong Wei	23	Friend
Tester 3	Muhammad Izham	23	Friend

The List below shows the performance measure of the system, which will be used to rate the performance of the system:

- Less than 0.5 will be low accuracy
- More than 0.5 and less than 0.8 will be average accuracy
- More than 0.8 will be high accuracy

6.5.1 Test for commands before the implementation of Soundex Algorithm

6.5.1.1 Test for command “light on”

Table 10 below shows the accuracy for the test of command “light on”.

Table 10: Accuracy for the test of command “light on”

	Correctly Identified	Accuracy
Tester 1	20 times	0.80
Tester 2	21 times	0.84
Tester 3	22 times	0.88
Average		0.84

Based on the table above, the average accuracy of the system to detect the command correctly is high, which is 0.84. All the testers are able to speak the command correctly for more than 20 times.

6.5.1.2 Test for command “fan on”

Table 11 below shows the accuracy for the test of command “fan on”.

Table 11: Accuracy for the test of command “fan on”

	Correctly Identified	Accuracy
Tester 1	21 times	0.84
Tester 2	19 times	0.76
Tester 3	18 times	0.72
Average		0.77

Based on the table above, the average accuracy of the system to detect the command "fan on", correctly is average, which is 0.77. Only tester 1 is able to speak the command correctly for more than 20 times.

6.5.1.3 Test for command "radio on"

Table 12 below shows the accuracy for the test of command "radio on".

Table 12: Accuracy for the test of command "radio on"

	Correctly Identified	Accuracy
Tester 1	20 times	0.80
Tester 2	21 times	0.84
Tester 3	21 times	0.84
Average		0.83

Based on the table above, the average accuracy of the system to detect the command correctly is high, which is 0.83. All the testers are able to speak the command correctly for more than 20 times.

6.5.1.4 Test for command “light off”

Table 13 below shows the accuracy for the test of command “light off”.

Table 13: Accuracy for the test of command “light off”

	Correctly Identified	Accuracy
Tester 1	19 times	0.76
Tester 2	19 times	0.76
Tester 3	18 times	0.72
Average		0.75

Based on the table above, the average accuracy of the system to detect the command correctly is average, which is 0.75. Non of the testers are able to speak the command correctly for more than 20 times.

6.5.1.5 Test for command “fan off”

Table 14 below shows the accuracy for the test of command “fan off”.

Table 14: Accuracy for the test of command “fan off”

	Correctly Identified	Accuracy
Tester 1	17 times	0.68
Tester 2	19 times	0.76
Tester 3	18 times	0.72
Average		0.72

Based on the table above, the average accuracy of the system to detect the command correctly is average, which is 0.72. Non of the testers are able to speak the command correctly for more than 20 times.

6.5.1.6 Test for command “radio off”

Table 15 below shows the accuracy for the test of command “radio off”.

Table 15: Accuracy for the test of command “radio off”

	Correctly Identified	Accuracy
Tester 1	20 times	0.80
Tester 2	22 times	0.88
Tester 3	20 times	0.80
Average		0.83

Based on the table above, the average accuracy of the system to detect the command correctly is high, which is 0.83. All the testers are able to speak the command correctly for more than 20 times.

6.5.1.7 Test for command "on light"

Table 16 below shows the accuracy for the test of command "on light".

Table 16: Accuracy for the test of command "on light"

	Correctly Identified	Accuracy
Tester 1	18 times	0.72
Tester 2	17 times	0.68
Tester 3	17 times	0.68
Average		0.69

Based on the table above, the average accuracy of the system to detect the command correctly is average, which is 0.69. All the testers are not able to speak the command correctly for more than 20 times.

6.5.1.8 Test for command "on fan"

Table 17 below shows the accuracy for the test of command "on fan".

Table 17: Accuracy for the test of command "on fan"

	Correctly Identified	Accuracy
Tester 1	17 times	0.68
Tester 2	16 times	0.64
Tester 3	16 times	0.64
Average		0.65

Based on the table above, the average accuracy of the system to detect the command correctly is average, which is 0.65. All the testers are not able to speak the command correctly for more than 20 times.

6.5.1.9 Test for command “on radio”

Table 18 below shows the accuracy for the test of command “on radio”.

Table 18: Accuracy for the test of command “on radio”

	Correctly Identified	Accuracy
Tester 1	19 times	0.76
Tester 2	20 times	0.80
Tester 3	19 times	0.76
Average		0.77

Based on the table above, the average accuracy of the system to detect the command correctly is average, which is 0.77. Only tester 2 is able to speak the command correctly for more than 20 times.

6.5.1.10 Test for command “off light”

Table 19 below shows the accuracy for the test of command “off light”.

Table 19: Accuracy for the test of command “off light”

	Correctly Identified	Accuracy
Tester 1	15 times	0.60
Tester 2	16 times	0.64
Tester 3	15 times	0.60
Average		0.61

Based on the table above, the average accuracy of the system to detect the command correctly is average, which is 0.61. All the testers are not able to speak the command correctly for more than 20 times.

6.5.1.11 Test for command “off fan”

Table 20 below shows the accuracy for the test of command “off fan”.

Table 20: Accuracy for the test of command “off fan”

	Correctly Identified	Accuracy
Tester 1	16 times	0.64
Tester 2	15 times	0.60
Tester 3	17 times	0.68
Average		0.64

Based on the table above, the average accuracy of the system to detect the command correctly is average, which is 0.64. All the testers are not able to speak the command correctly for more than 20 times.

6.5.1.12 Test for command “off radio”

Table 21 below shows the accuracy for the test of command “off radio”.

Table 21: Accuracy for the test of command “off radio”

	Correctly Identified	Accuracy
Tester 1	18 times	0.72
Tester 2	18 times	0.72
Tester 3	20 times	0.80
Average		0.76

Based on the table above, the average accuracy of the system to detect the command correctly is average, which is 0.76. Only tester 3 is able to speak the command correctly for more than 20 times.

6.5.1.13 Average results for the commands in the system before the implementation of Soundex Algorithm.

Table 22 below shows the overall average results for each commands used in this system before the implementation of Soundex Algorithm.

Table 22: Average results before implementation of Soundex Algorithm

Commands	Average accuracy
“light on”	0.84
“fan on”	0.77
“radio on”	0.83
“light off”	0.75
“fan off”	0.72
“radio off”	0.83
“on light”	0.69
“on fan”	0.65
“on radio”	0.77
“off light”	0.61
“off fan”	0.64
“off radio”	0.76

6.5.2 Test for commands after the implementation of Soundex Algorithm

6.5.2.1 Test for command "light on"

Table 23 below shows the accuracy for the test of command "light on".

Table 23: Accuracy for the test of command "light on"

	Correctly Identified	Accuracy
Tester 1	21 times	0.84
Tester 2	22 times	0.88
Tester 3	22 times	0.88
Average		0.87

Based on the table above, the average accuracy of the system to detect the command correctly is high, which is 0.87. All the testers are able to speak the command correctly for more than 20 times. From the result above, we can conclude that the command "light on" is easier for the system to detect, based on the technique used in this system.

6.5.2.2 Test for command "fan on"

Table 24 below shows the accuracy for the test of command "fan on".

Table 24: Accuracy for the test of command "fan on"

	Correctly Identified	Accuracy
Tester 1	21 times	0.84
Tester 2	20 times	0.80
Tester 3	19 times	0.76
Average		0.80

Based on the table above, the average accuracy of the system to detect the command "fan on", correctly is high, which is 0.8. Tester 1 and Tester 2 are able to speak the command correctly for more than 20 times. From the result above, we can conclude that the command "fan on" is easier for the system to detect, based on the technique used in this system.

6.5.2.3 Test for command "radio on"

Table 25 below shows the accuracy for the test of command "radio on".

Table 25: Accuracy for the test of command "radio on"

	Correctly Identified	Accuracy
Tester 1	23 times	0.92
Tester 2	23 times	0.92
Tester 3	22 times	0.88
Average		0.91

Based on the table above, the average accuracy of the system to detect the command correctly is high, which is 0.91. All the testers are able to speak the command correctly for more than 20 times. From the result above, we can conclude that the command "radio on" is easier for the system to detect, based on the technique used in this system.

6.5.2.4 Test for command “light off”

Table 26 below shows the accuracy for the test of command “light off”.

Table 26: Accuracy for the test of command “light off”

	Correctly Identified	Accuracy
Tester 1	20 times	0.80
Tester 2	19 times	0.76
Tester 3	21 times	0.84
Average		0.80

Based on the table above, the average accuracy of the system to detect the command correctly is high, which is 0.8. Tester 1 and Tester 3 are able to speak the command correctly for more than 20 times. From the result above, we can conclude that the command “light off” is easier for the system to detect, based on the technique used in this system.

6.5.2.5 Test for command “fan off”

Table 27 below shows the accuracy for the test of command “fan off”.

Table 27: Accuracy for the test of command “fan off”

	Correctly Identified	Accuracy
Tester 1	19 times	0.76
Tester 2	21 times	0.84
Tester 3	19 times	0.76
Average		0.79

Based on the table above, the average accuracy of the system to detect the command correctly is average, which is 0.79. Only tester 2 is able to speak the command correctly for more than 20 times. From the result above, we can conclude that the command "fan off" is easier for the system to detect, based on the technique used in this system as the average accuracy almost approaches 0.8.

6.5.2.6 Test for command "radio off"

Table 28 below shows the accuracy for the test of command "radio off".

Table 28: Accuracy for the test of command "radio off"

	Correctly Identified	Accuracy
Tester 1	21 times	0.84
Tester 2	22 times	0.88
Tester 3	22 times	0.88
Average		0.87

Based on the table above, the average accuracy of the system to detect the command correctly is high, which is 0.87. All the testers are able to speak the command correctly for more than 20 times. From the result above, we can conclude that the command "radio off" is easier for the system to detect, based on the technique used in this system.

6.5.2.7 Test for command “on light”

Table 29 below shows the accuracy for the test of command “on light”.

Table 29: Accuracy for the test of command “on light”

	Correctly Identified	Accuracy
Tester 1	18 times	0.72
Tester 2	19 times	0.76
Tester 3	17 times	0.68
Average		0.72

Based on the table above, the average accuracy of the system to detect the command correctly is average, which is 0.72. All the testers are not able to speak the command correctly for more than 20 times. From the result above, we can conclude that the command “on light” is a little harder for the system to detect, based on the technique used in this system.

6.5.2.8 Test for command “on fan”

Table 30 below shows the accuracy for the test of command “on fan”.

Table 30: Accuracy for the test of command “on fan”

	Correctly Identified	Accuracy
Tester 1	19 times	0.76
Tester 2	19 times	0.76
Tester 3	18 times	0.72
Average		0.75

Based on the table above, the average accuracy of the system to detect the command correctly is average, which is 0.75. All the testers are not able to speak the command correctly for more than 20 times. From the result above, we can conclude that the command “on fan” is a little harder for the system to detect, based on the technique used in this system.

6.5.2.9 Test for command “on radio”

Table 31 below shows the accuracy for the test of command “on radio”.

Table 31: Accuracy for the test of command “on radio”

	Correctly Identified	Accuracy
Tester 1	19 times	0.76
Tester 2	20 times	0.80
Tester 3	20 times	0.80
Average		0.79

Based on the table above, the average accuracy of the system to detect the command correctly is average, which is 0.79. Tester 2 and tester 3 are able to speak the command correctly for more than 20 times. From the result above, we can conclude that the command “on radio” is easier for the system to detect, based on the technique used in this system as the average accuracy almost approaches 0.8.

6.5.2.10 Test for command "off light"

Table 32 below shows the accuracy for the test of command "off light".

Table 32: Accuracy for the test of command "off light"

	Correctly Identified	Accuracy
Tester 1	17 times	0.68
Tester 2	18 times	0.72
Tester 3	18 times	0.72
Average		0.71

Based on the table above, the average accuracy of the system to detect the command correctly is average, which is 0.71. All the testers are not able to speak the command correctly for more than 20 times. From the result above, we can conclude that the command "off light" is a little harder for the system to detect, based on the technique used in this system.

6.5.2.11 Test for command "off fan"

Table 33 below shows the accuracy for the test of command "off fan".

Table 33: Accuracy for the test of command "off fan"

	Correctly Identified	Accuracy
Tester 1	17 times	0.68
Tester 2	18 times	0.72
Tester 3	17 times	0.68
Average		0.69

Based on the table above, the average accuracy of the system to detect the command correctly is average, which is 0.69. All the testers are not able to speak the command correctly for more than 20 times. From the result above, we can conclude that the command “off fan” is a little harder for the system to detect, based on the technique used in this system.

6.5.2.12 Test for command “off radio”

Table 34 below shows the accuracy for the test of command “off radio”.

Table 34: Accuracy for the test of command “off radio”

	Correctly Identified	Accuracy
Tester 1	20 times	0.80
Tester 2	19 times	0.76
Tester 3	20 times	0.80
Average		0.79

Based on the table above, the average accuracy of the system to detect the command correctly is average, which is 0.79. Tester 1 and tester 3 are able to speak the command correctly for more than 20 times. From the result above, we can conclude that the command “off radio” is easier for the system to detect, based on the technique used in this system, as the average accuracy almost approaches 0.8.

6.5.2.13 Average results for the commands in the system after the implementation of Soundex algorithm

Table 35 below shows the overall average results for each commands used in this system after the implemetation of Soundex Algorithm.

Table 35: Average results after the implementation of Soundex Algorithm

Commands	Average accuracy
“light on”	0.87
“fan on”	0.80
“radio on”	0.91
“light off”	0.80
“fan off”	0.79
“radio off”	0.87
“on light”	0.72
“on fan”	0.75
“on radio”	0.79
“off light”	0.71
“off fan”	0.69
“off radio”	0.79

Table 36 below shows the comparison of average results before and after implementing the Soundex Algorithm.

Table 36: Comparison of results before and after Soundex Algorithm implementation

Command	Average before Soundex	Average after Soundex
“light on”	0.84	0.87
“fan on”	0.77	0.80
“radio on”	0.83	0.91
“light off”	0.75	0.80
“fan off”	0.72	0.79
“radio off”	0.83	0.87
“on light”	0.69	0.72
“on fan”	0.65	0.75
“on radio”	0.77	0.79
“off light”	0.61	0.71
“off fan”	0.64	0.69
“off radio”	0.76	0.79

Based on the table above, we can clearly see that the average results for each commands after implementing the Soundex Algorithm is higher compared to the average results for each commands before implementing the Soundex Algorithm. Each words spoken by the user is processed by the system with the aid of the Soundex Algorithm and the decision is made by the system whether to turn on or turn of the LED lights. Based on the results, we can conclude that the technique used, which is utilizing the Soundex Algorithm can make the system to perform better.

6.6 Conclusion

Testing is one of the important part to meet the requirement of the project. This system is tested with multiple commands which can be used to control the LED lights on the Raspberry Pi. We can reduce unexpected errors by doing testing, before we move on to the next stage of the project. Besides, testing has to be done for each funtion of the program, to ensure that all the variables and conditions are well defined and does not overlap with each other. Based on the testing done, we can conclude that the technique used, which is utilizing the Soundex Algorithm can make the system to perform better.

CHAPTER 7

CONCLUSION

7.1 Observation on weaknesses and strengths

Based on the test run, the system have strengths and weaknesses. The system can perform well with the technique used, which is by using the Soundex algorithm to extract the sound of word spoken by the user, and convert it into string form, which will be used by the system to make decision based on the conditions. One of the weaknesses of the system is the surrounding noise of the environment. If the environment is noisy, the system will not perform well as the microphone will capture all the sounds in the surroundings. So the system have to run in a quiet environment in order to perform well without any fault. Besides, another weakness will be the internet connection. If the internet connection is low or lost, the system is unable to run, as the system uses Google speech recognition API to return the command spoken by the user. So, the internet connection must be stable in order for the system to run well.

7.2 Proposition for improvement

Form the hardware side, one of the suggestion to improve the system is using a good quality microphone. Using a good quality microphone can make the system to detect the command of the user more easily and clearly. Besides, a good quality microphone with noise cancelling feature can make the system to perform better as it able to omit sounds in the surrounding and only detect the command of the user based on the threshold.

Besides, to improve the accuracy of the system to detect command even more, we can collect more data of keywords which are similar in sound with the command and process them. By having more data of keywords, the accuracy of the system to detect the correct command will be high.

7.3 Project contribution

This project is completed with the contribution of Universiti Teknikal Malaysia Melaka (UTeM) under the Faculty of Information and Communication Technology for providing an adequate place and also facilities to conduct this final year project. This project is also completed under the supervision of Dr. Yogan Jaya Kumar who contributed a lot of ideas and solutions for this project. Comments from the evaluator, Dr Halizah binti Basiron also have contributed a lot in improving this project.

7.4 Conclusion

The objectives of this project are met where it's able to design a home automation system prototype with speech recognition with the aid of Google Speech Recognition API alongside with the python speech recognition packages. Besides, the investigation of whether the home automation system is very helpful and easier to be used by people has been done and we can conclude that the users can give the appropriate command easily as it only consists of two words. Moreover, the performance of the recognition of speech in this system has been assessed and the accuracy of the system to detect the command from user is high in most cases. So, the technique used can make the system to perform well.

APPENDICES

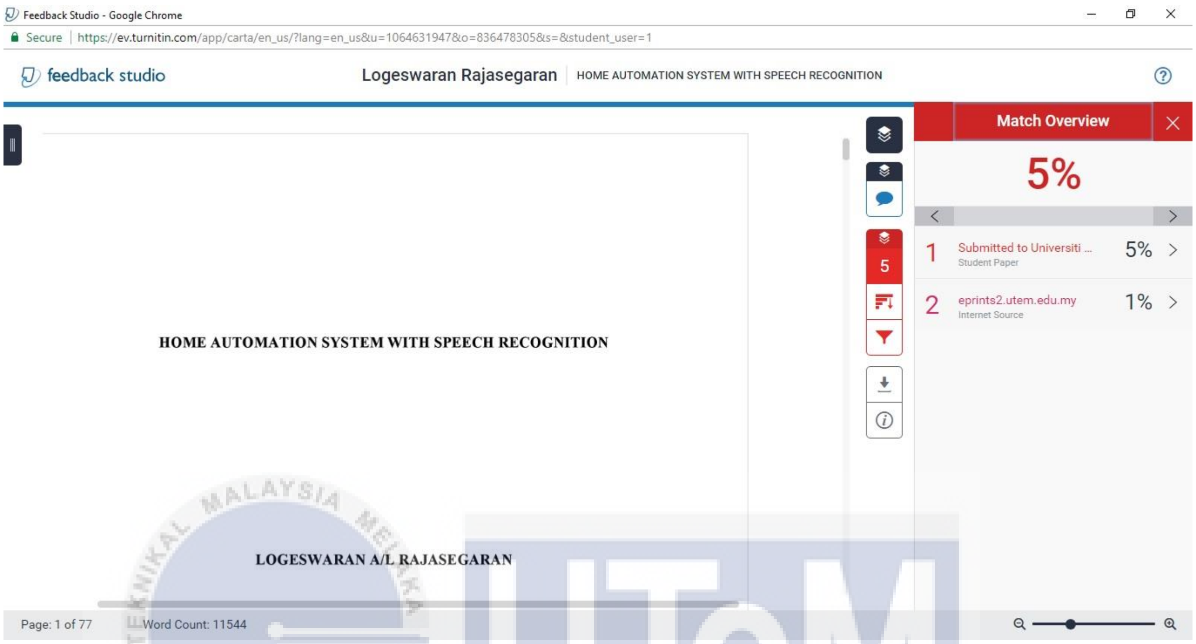


Figure A1: Turnitin report for the final year project report

Figure A2: Hardware used for the system

Figure A3: Illumination of red LED for command “light on” or “on light”

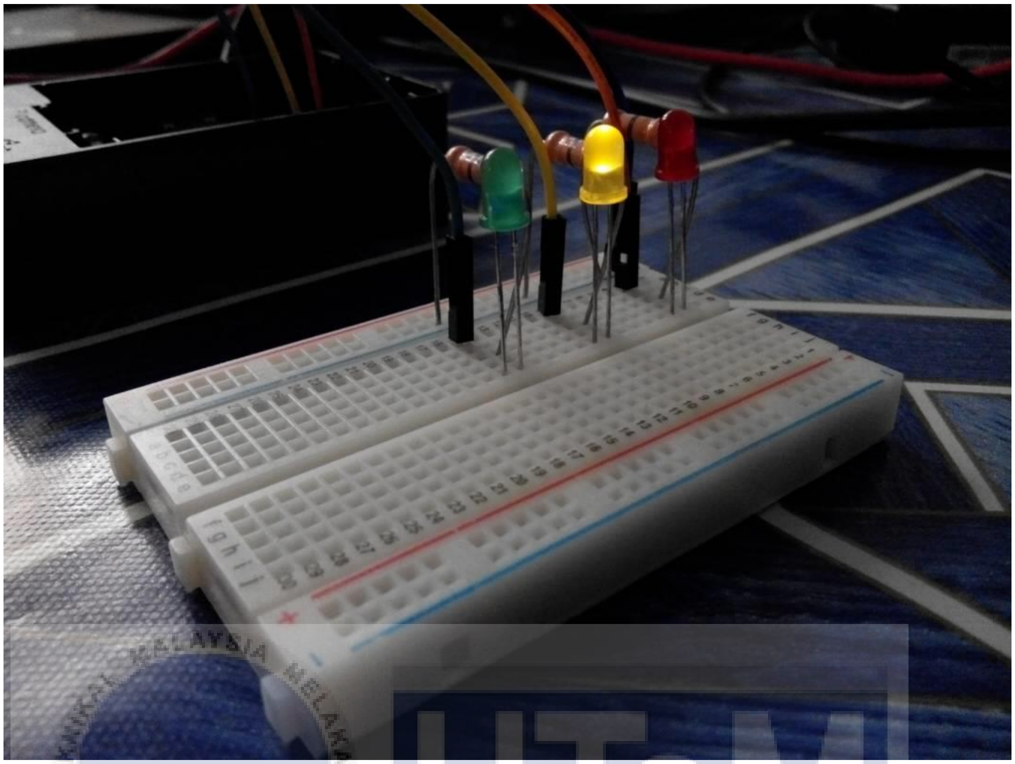


Figure A4: Illumination of yellow LED for command “radio on” or “on radio”

Figure A5: Illumination of green LED for command “fan on” or “on fan”

```
*Python 2.7.9 Shell*
File  Edit  Shell  Debug  Options  Windows  Help
Python 2.7.9 (default, Sep 17 2016, 20:26:04)
[GCC 4.9.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
speak!
Recognised!
b050
o500
['b050', 'o500']
turning on the fan
speak!
Recognised!
v030
o500
['v030', 'o500']
turning on the radio
speak!
Recognised!
l0203
o500
['l0203', 'o500']
turning on the light
speak!
Recognised!
turning on the light
```

Figure A6: Sample output of the system

REFERENCES

Mukesh Kumar, Shimi S.L. (2015). "Voice recognition based home automation system" Proceedings of International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE).

Mardiana.B, Hazura.H. (2009). "Home Appliances Controlled Using Speech Recognition in Wireless Network Environment." Proceedings of ICCTD International Conference on Computer Technology and Development.

Faisal Baig, Muhammad Fahad Khan. (2012), "Controlling Home Appliances Remotely through Voice Command", International Journal of Computer Applications, Vol. 48.

Amrutha S, Aravind S, (2015), "Speech Recognition Based Wireless Automation of Home Loads- E Home" International Journal of Engineering Science and Innovative Technology (IJESIT), Vol. 4.

Lee K.F, Hon H.W, Reddy R, (1990), "An overview of the SPHINX speech recognition system" IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 38.

Doug Hellmann, (2012), "Using Fuzzy Matching to Search by Sound with Python", The Soundex Indexing System, U.S National Archives.

Alexander Beider, Stephen P.Morse, (2010), "Phonetics Matching: A Better Soundex", Association of Professional Genealogists Quarterly.

Donald Fenna, (2003), "Phonetics reduction of names" Computer Programs in Biomedicine, Vol. 19, issue 1, pp. 31-36.

X.D. Huang, Y. Ariki (1990), "Hidden Markov Models for Speech Recognition", Edinburgh University Press, Edinburgh Information Technology Series 7.

Preeti Saini, Parneet Kaur (2013), "Automatic Speech Recognition: A Review", International Journal of Engineering Trends and Technology, Vol 4 Issue 2.