

NIA, REMIND ME! - AI PERSONAL ASSISTANT SYSTEM

AININ SOFIYA HISHAM

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

1.1.1 Technique

1.1.1.1 Research

Research is an in-depth analysis of information that creates space for generating new questions, concepts, and understandings. The main objective of research is to explore the unknown and unlock new possibilities with a careful and detailed study of a particular problem or concern, using scientific methods. How many smart assistants have been built but how much of them are able or suite for everyone? Is it affordable? IT sector has been widely using in the world, we as Malaysian are not far behind from their super technology. Malaysian is the country that is in progress to develop countries, especially in technology. Many smart assistants built, but how much of them can suite each person in this world? Like Siri which only been built for Apple user, Cortana by Microsoft but not all regions are available to use it, Google Assistant which only available for certain language and does not include Malay language.

1.1.1.2 Artificial Intelligence Personal Assistant System Technique Used

- **Interactive AI**

Users can utilize this form of AI to automate communication without losing interactivity. Consider chatbots and smart personal assistants as examples of this sort of AI, whose capabilities range from answering pre-programmed queries to comprehending the context of a discussion. Another use for interactive AI is to improve a company's internal operations. For example, this project was dedicated to creating a smart personal assistant system to ease human's activities and scheduling daily tasks.

- **Text AI**

Productions of system that use text AI can enjoy machine translation text recognition, speech-to-text alteration and content generation skills. This little technique can still take advantage of this AI type even if a company is not any other giant company like Google, Amazon or others, that provides text AI as a service. Text AI, for example, may be used to fuel an internal corporate knowledge repository Unlike a traditional knowledge base, which relies on keyword searches, an AI-powered knowledge base can discover the document containing the most relevant response even if it lacks full keywords. This is made feasible by semantic search and natural language processing, which enable AI to construct semantic maps and distinguish synonyms in order to comprehend the context of the user's query. Text AI is used in this project to display what user command and what system able to understand from user command.

1.2 Project Methodology

1.2.1 System Development Life Cycle (Sdlc) Methodology Agile Methodology

As for completing this project, agile model approach has been used. Agile allow working product or project deliver quickly as each of iteration in each phase are eligible to be tested.

At planning phase, brainstorming the idea that can suite with current problem faced by people in organizing their life. Identify the user requirements and gathering all the requirement. Proceed on analyzing requirement and sperate user's needs and wants.

Design phase is where the system development will begin with the flow of the system. This include user that involved in project development and end-user.

Build phase is where system been implemented following the user and project requirements. Structure of the system program or code are using python library.

Proceed to testing phase, this should involve developer, system tester and end user. Produce a documentation user acceptance testing which count on expected result and actual result while testing the system.

Last but not least product review, review can achieve by feedback of system tester and end user. This is to show whether the project meet the user and project requirement or not. If both are acceptable, the product can be presented and launch, else proceed to the second sprint and third sprint. Planning stage may stay as planned or been changed due to user satisfaction of using or feedbacks for improvement phase.



FIGURE 1.2-1 AGILE METHODOLOGY PROCESS

1.3 Project Requirement

1.3.1 Software Requirement

TABLE 1.3-1 SOFTWARE REQUIREMENT

Requirement	Minimum	Recommended
RAM	2.5 GB of RAM	8 GB of RAM
CPU	1.8 GHz or faster processor.	Quad-core or better recommended
DISK SPACE	800MB up to 210 GB of available space, depending on features installed	Typical installations require 20-50 GB of free space.
MONITOR RESOLUTION	720p (1280 by 720)	Visual Studio will work best at a resolution of WXGA (1366 by 768) or higher.
OPERATING SYSTEM	<p>Windows 10 version 1703 or higher: Home, Professional, Education, and Enterprise (LTSC and S are not supported)</p> <p>Windows Server 2019: Standard and Datacenter</p> <p>Windows Server 2016: Standard and Datacenter</p> <p>Windows 8.1 (with Update 2919355): Core, Professional, and Enterprise</p> <p>Windows Server 2012 R2 (with Update 2919355): Essentials, Standard, Datacenter</p> <p>Windows 7 SP1 (with latest Windows Updates): Home Premium, Professional, Enterprise, Ultimate</p>	Latest 64-bit version of Windows, macOS, operating systems (64 bit recommended; ARM is not supported)

1.3.2 Hardware requirement

Table 1.3-2 Hardware requirement

Requirement	PC specification Hardware Requirement (ASUS TUF GAMING A15)
Processor	Processor AMD Ryzen 7 4800H with Radeon Graphics, 2900 Mhz, 8 Core(s), 16 Logical Processor(s)
RAM	16 GB
Storage (SSD)	512GB PCIe® Gen3 SSD
Operating System	OS Name Microsoft Windows 10 Home Single Language
Graphic or Video card	NVIDIA GeForce GTX 1650Ti 4GB GDDR6 VRAM

1.3.3 Other Requirements

There is no any other requirements included since this project being conducted by supervisor remotely due to Covid-19 pandemic.

1.4 Project Gantt Chart PSM 1 & PSM 2

TABLE 1.4-1 GANTT CHART PSM 1 & PSM 2

Task Name	Month																						
	Feb	Mac			Apr				May				June			July		August				September	
	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17	W18	W19	W20	W21	W22	W23
Chapter 1: Planning phase																							
Identify Project Title																							
Submit and Verify Proposal																							
Identify and verify Problem Statement																							
Identify Objectives and Scope																							
System Development module 1																							
Chapter 2: Literature Review and Methodology																							
Conduct Fact Findings																							
List Project Methodology																							
Identify Project Requirements																							
Gathering Requirements																							
System Development module 2																							
Chapter 3: Analysis Phase																							
Analyze Requirement																							
Produce and verify System Requirements Specifications																							

Table 1.4-2 Full Gantt chart one page

Task Name	Month																						
	Feb		Mac		Apr			May				June			July		August			September			
	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17	W18	W19	W20	W21	W22	W23
Chapter 1: Planning phase																							
Identify Project Title																							
Submit and Verify Proposal																							
Identify and verify Problem Statement																							
Identify Objectives and Scope																							
System Development module 1																							
Chapter 2: Literature Review and Methodology																							
Conduct Fact Findings																							
List Project Methodology																							
Identify Project Requirements																							
Gathering Requirements																							
System Development module 2																							
Chapter 3: Analysis Phase																							
Analyze Requirement																							
Produce and verify System Requirements Specifications																							
Produce Gantt Chart PSM 1 & 2																							
System Development module 3																							
Chapter 4: Design Phase																							
Design Interface																							
Create and verify Flowchart																							
Create use case diagram																							
System Development module 3																							
Project Demo																							
Project Presentation																							
Project Report Maintenance																							
Fix previous semester																							
Chapter 5: Implementation																							
Code Program Development																							
Reorganize program code																							
Full system development																							
Chapter 6: Testing																							
Produce test cases																							
Conduct testing																							
User Acceptance Testing																							
Chapter 7: Project Conclusion																							
Observation																							
Maintenance																							
Project Presentation																							
Project Report Maintenance																							

1.5 Conclusion

In literature review, information about the project development is gathered from previous research and developers of related domain obtained from articles, journals and website. All those references are used to determine the important elements needed in this project. In project methodology, the flow of the project is suitable planned in order to ensure smooth completion of the project within due deadline. The next chapter will discuss about the project analysis that includes problem analysis and requirement analysis.

Chapter 2: ANALYSIS

2.1 Introduction

In this chapter, analysis on project development will be conducted. This includes what problem occur during project development, how the system should work, the libraries that used to apply in the program to make it work and worth.

2.2 Problem Analysis

In developing this project, the main important is signal process. This relates to sound waves which happen from analog to digital and digital to analog. With built in microphone and speaker these should be no problem to do the conversion. But what the problem is speech recognition. In this world, there are many kinds of speech and accent. How can we make sure the system can understand user's speech? Developer use the ready-made vocabulary by Google as it has big dataset of vocabulary around the world. Developing this project basely using English language as it has simple term. The second problem is to make sure the voice command detection detects the correct and same as what user said it out and wants. The recognition of the speech by user sometime turns out to another terms, as the signal processing is not right due to background noise. Therefore, both of the problem could just be on the right track with correct and clear pronunciation by the user.

2.3 Requirement analysis

2.3.1 Data Requirement

The main input that should be in the system is voice command. The other input could be text, character or number that will be needed if voice command unable to be recognized correctly. The interface that involves might be the main section, reminder section, schedule display section, and command editor section. The type of data to store in database is more to text as user speak language converted to text and followed by number for date and time format. Characters might be included in the Notes space of reminder section.

Data Dictionary for Reminder Section

Table 2.3-1 Sample of Data Involve

Attributes	Description	Data Type	Field Size	Example
Day	The day that system need to remind user	Varchar	20	ISNIN; TUESDAY; FRIDAY; SUNDAY
Time	Time that system need to remind user	datetime	8	HH:MM:ss
Notes	What user want to remind or jot down about	varchar	100	Saya nak pergi keluar makan tengah hari; I want to do laundry
Sound	What sound alarm that user want to use for reminder	varchar	10	Any sound available
Recurring	Option for user to make the reminder recurring or not	varchar	10	Checked or Unchecked the Recurring checkbox
Active	Option for user to make the reminder is active or not	varchar	10	Checked or Unchecked the Active checkbox

2.3.2 Functional Requirements

- Speak able
- Set Reminders and Alarm
- Search information
- Auto-play system

2.3.3 Non-Functional Requirements

- Maintainability requirements: Changeability

No error shall need more than one person-days to identify and fix.

- Reliability requirements: Recoverability

In case of an error, the time needed to get the system up and running again shall not exceed one hour (MTTR).

2.3.4 Other requirement

Work of NIA (Nice In Assist)

- **Signal processing:**

It is the technology present in all stages of sound processing, from the conversion from analog to digital, to the application of filters or the gain control. Regarding Nia, it enables extracting the information within signals to translate it recognizable words.

- **Speech recognition:**

This is the most important part of this process, here the actual recognition is done. The feature vectors sequence is then decoded into a sequence of words. This is done by using algorithms such as python library speech recognition. The program includes big dictionary of popular words that exist in language using Google Vocabulary. Each feature vector should be matched against the sound.

- **Semantic interpretation:**

The algorithm will check to see if a given syllable can appear after another in the language. After that there will be grammar check which more to English vocab. The combination of words either it makes any sense or not also will be triggered. The way to decide how it can respond to the user is by using “command mode” semantic property.

- **Dialogue management:**

Mistakes are attempted to be corrected. The meaning of the combined words is then extracted, and the task at hand is completed. It is a framework that allows Nia talents to be authenticated. The conversation management produces a list of instructions for other sections of the conversation system, which is normally in the form of a semantic interpretation.

- **Response generation:**

Following the completion of a task, the response or outcome of the task is generated. The response can take the shape of a speech or a text message. It is chosen here the words to employ in order to optimize the user's understanding. The text to speech conversion method is employed if the response is to be supplied in the form of speech. In many cases Nia voice is not digitized at all. It is really Zira Voice or whoever else owns Nia voice in each country. In case, there is little no time for an answer to be recorded Nia can have her own response be readout by using her digitally generated Zira voice font.

Libraries or References

This page list out the libraries or references use to able the system work as needed. The main libraries used is definitely speech recognition as it act as an auto engine for reconition process. Below show some of additional references or library for speech recognition process and system project development using C# library/namespace.

Table 2.3-2: C# Reference for Classes Object

Libraries	Description
Speech.Recognition	This is the reference for speech recognition engine object wich allows the application to listen for and recognize spoken words or phrases by declaring the suitable class which is using System.Speech.Recognition.
Speech.Synthesis	This is the reference for speech synthesizer object which allow system to gives the application the ability to speak.
System.Globalization	“This references have classes that define culture-related information, including country/region, numbers, , calendars in use, format patterns for dates, currency, and language, and sort order for strings. These classes are useful for writing globalized (internationalized) applications. Classes such as StringInfo and TextInfo provide advanced globalization functionalities, including surrogate support and text element processing” (System.Globalization Namespace, n.d.).

System.Diagnostics	This reference provides classes that allow user to interact with system processes, performance counters, and event logs where user can process to start any special file or folder.
System.Media	This reference contains classes for playing sound files and accessing sounds that provided by the system which SoundPlayer will be used to declare the object.
System.IO	This reference have types that come out with basic file and directory support function. In this project, developer used types that allow system to do the writing(File.WriteLine) and reading(File.ReadAlllines) part to files and data streams.
System.Collection. Generic	“This reference have classes and interfaces that define generic collections, that allow users to create strongly typed collections that give better type safety and performance than non-generic strongly typed collections” (System.Collections.Generic Namespace, 2021).
Microsoft.Win32	This reference contains two types of classes which are those that handle events raised by the operating system and the one whoe that manipulate the system registry.

2.4 Conclusion

In conclusion, the problems and requirements have been analysed in order to achieve the objectives of this project smoothly. The analysis is done to obtain a clearer understanding on what this project should have and should be able to do. The next chapter will be on the design of the system.

Chapter 3: DESIGN/THE PROPOSED TECHNIQUE

3.1 Introduction

This chapter will explain on the design and proposed technique of the system. The design phase is needed as for visualise the system looks like and how it would function. Other than that, the design phase is also part of utilizing requirements mentioned in the analysis chapter previously.

3.2 High-Level Design

For Application Development Project

This system application or desktop application should be able to work offline for all tasks except for searching and retrieving high definition of information. This application gives first level of security where in order to run the application and use it, user need to insert the secret key. This achieves the first objective which is to design and develop intelligent voice recognition program with offline connection. The application should also be able to remind user by notify user based on their schedule added. The schedule is functional for seven days where user can choose on what days that they needed to be reminded and what is it about. This achieves the second objective which is to develop smart speaker assistant in reminding users. While user is having internet connection for the computer device, user should be able to speak to the system and ask something and it will lead to google.com browser as this might achieve the third objective of this project which is to allow user to receive relevant information by voice command.

3.2.1 System Architecture

Offline and Online System Architecture

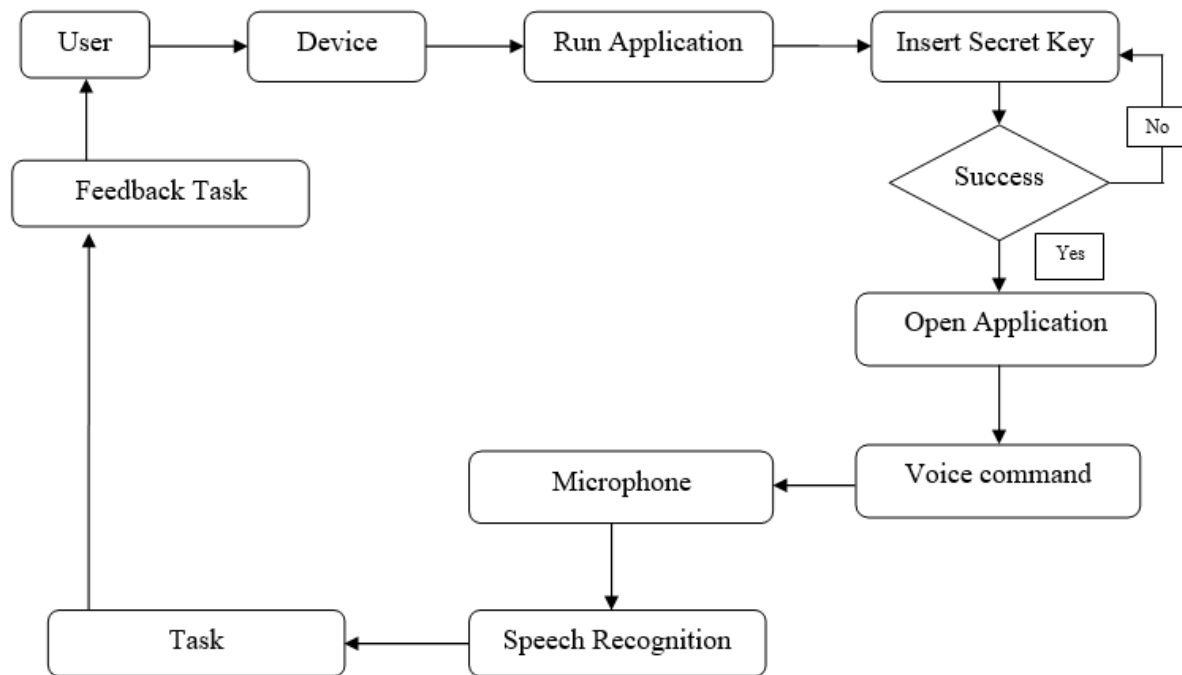


Figure 3.2-1 System architecture

User use and must has device(computer) in order to run the application. After application is opened, user need to enter the secret key. If the secret key is wrong, system will ask user to insert correct keyword else user can proceed. Through microphone, user can use either built-in or external in order to voice command for the system doing tasks. System then will do speech recognition and give feedback for the task. If user try to search for any information, system will ask to connect to the internet. If user already connected to the internet, system would do the task following command give feedback to user either by voice or user inter interface.

3.2.2 User Interface Design for expert system/DSS/simulation

In this section, the user interface design will be further explained in terms of navigation design, input design, and output design. The user interface consists of 8 forms where a few details are prompted from the user, another page that also prompts details from the user voice commands.

3.2.2.1 Navigation Design



Figure 3.2-2 Navigation Design for Secret Key Validation

This Figure 4.2.2.1.1 is navigation button for the first user interface of Secret Key section as the application is running. User need to enter the secret key or password and click this button in order to proceed to next page.



Figure 3.2-3 Navigation Design for Help User

This Figure 4.2.2.1-2 is navigation button or icon that used to help user looking for the list of commands available that able user to interact with the system using voice command. This navigation button also will lead to user interface of Commands List. User may later click the '+' button the dropdown all the commands phrase.



Figure 3.2-4 Navigation Design for Reminder or Schedule Section

This navigation button or icon indicate that user is using the reminder, or they have schedule. Figure 4.2.2.1-4 shows that this icon only appears when user to do scheduler set the reminder at Reminder section and it will shrink into icon in line with another symbol like internet connection, security, or battery icon. User may right click on this icon Reminder to lead to schedule list.



Figure 3.2-5 Navigation Design for Schedule Section

This navigation button is only will be appear when user want to schedule their activities at user interface Schedule/Reminder. Figure 4.2.2.1-4 shows there are 4 buttons available, and its function will lead to different path. 'Add' button is used and can be clicked by user to proceed adding the new reminder or editing part. 'Settings' button is used and can be clicked by user to do setting for the application. The more explanation about settings will be on the User Interface for Add/Edit reminder or schedule below.

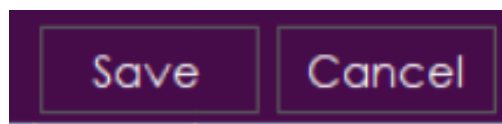


Figure 3.2-6 Navigation Design for Add/Edit Section

Figure 4.2.2.1-5 shows the navigation design for user interface Add/Edit section. User can click button 'Save' to proceed the reminder details and save it to list of schedules in Schedule User Interface. While 'Cancel' button used for user who do not want to save the reminder details to schedule list.

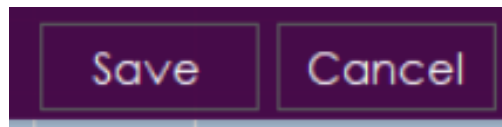


Figure 3.2-7 Navigation Design for Settings

This Figure 4.2.2.1-6 shows the navigation design for settings of the system application. 'Save' button is disabled if user does not make any changes to the settings and only can be clicked after done some changes. 'Cancel' button is used to cancel any settings made by user and will lead to the schedule list section.



Figure 3.2-8 Navigation Design Icon for Settings

This Figure 4.2.2.1-7 shows the navigation design icon for Settings. This icon will only appear at the windows taskbar in line with Internet connection icon, battery icon or security icon. This icon is used to lead user direct to Settings for application user interface after user right click on the icon for Reminder at Figure 4.2.2.1-3.

3.2.2.2 Input Design for expert system/DSS/simulation

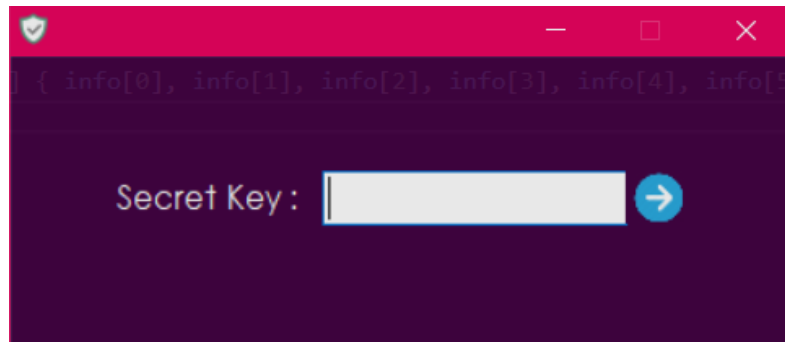


Figure 3.2-9 User Interface for Secret Key Input Design

Figure 4.2.2.2-1 shows the user interface input design for Secret Key Section. User may insert any possible words, numbers or characters that might be correct to proceed using the application.

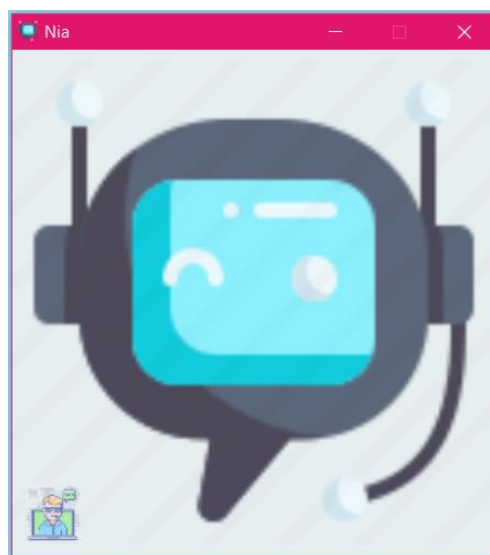


Figure 3.2-10 User Interface for Voice Command Input and Output Design

Figure 4.2.2.2-2 shows user interface input and output design for voice command. User may speak to give commands for ask something and system will reply to user by following the command. Also, there are available button input to view the Commands List user interface on Figure 4.2.2.4-2.

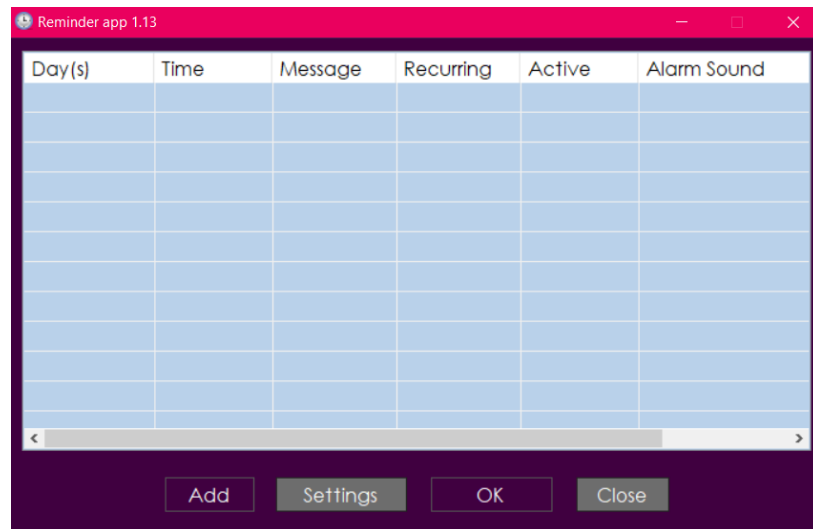


Figure 3.2-11 User Interface for Schedule List

Figure 4.2.2.2-3 shows the user interface for schedule list. Users need to input the schedule or reminder by clicking 'Add' button, click the 'Settings' button to do settings of the system application, click 'OK' button after done review schedule list and 'Close' button to close the button either schedule or reminder is added or not. All the buttons will navigate to the different output.

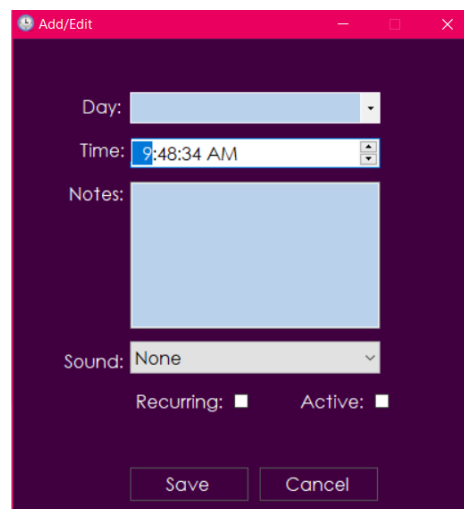


Figure 3.2-12 User Interface for Add/Edit reminder Input Design

Figure 4.2.2.2-4 shows the user interface for Add/Edit reminder. Users need to insert all the details needed and check whether user want the reminder is recurring and active in reminding them.

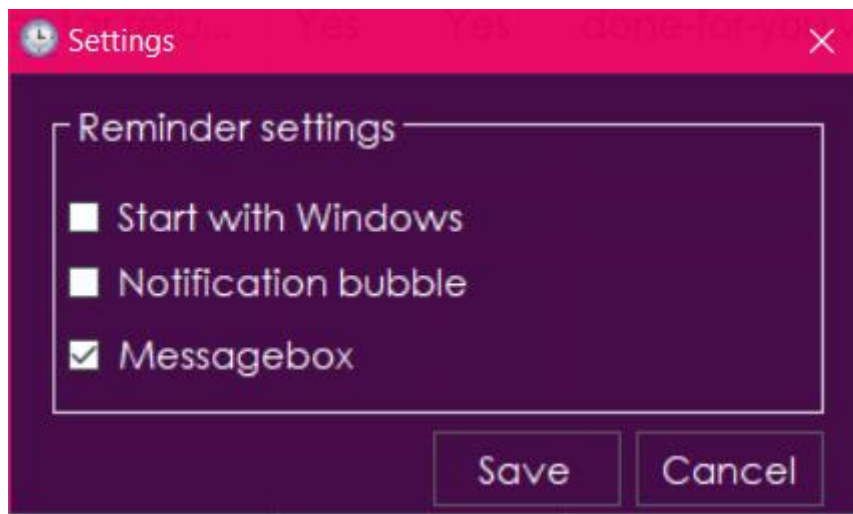


Figure 3.2-13 User Interface for Reminder Settings Input Design

Figure 4.2.2.2-5 shows the user interface for Reminder settings. User able to check the whitebox following choice for how the reminder should appear. MessageBox checkbox is for displaying the windows MessageBox for reminder. Notification bubble is for displaying reminder and it will pop up at the right bottom desktop for 3 second. While if user click the checkbox Start with Windows, the system application will as user switch on the computer. This means user no need to run application each time they want get the reminders.

3.2.2.3 Technical Design

The selected AI techniques are using Natural Language Processing (NLP) and Machine Learning. This can relate with the system development where NLP is implemented when user try to voice command all the possible and available phrases for the system to process and display what user is saying. Then, Machine Learning implemented while user voice command all the possible phrases one by one until the system able to recognize the correct phrases. Hence, for the next voice command user might not need to speak slowly or following the phonetic accent in order for the system to do recognition because the system already adapt with the voice, accent, or the way user talk.

3.2.2.4 Output Design

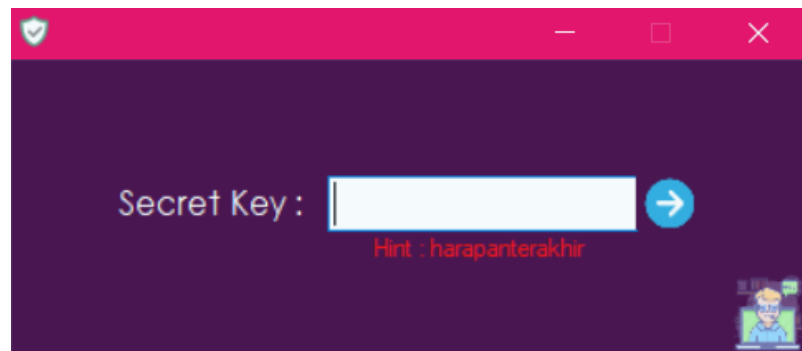


Figure 3.2-14 User Interface for Secret Key Output Design

Figure 4.2.2.4-1 shows the user interface for Secret Key. This interface will appear each user insert the wrong secret key. User can read the hint if forgot and mouse hover to the help icon to view on the secret key.

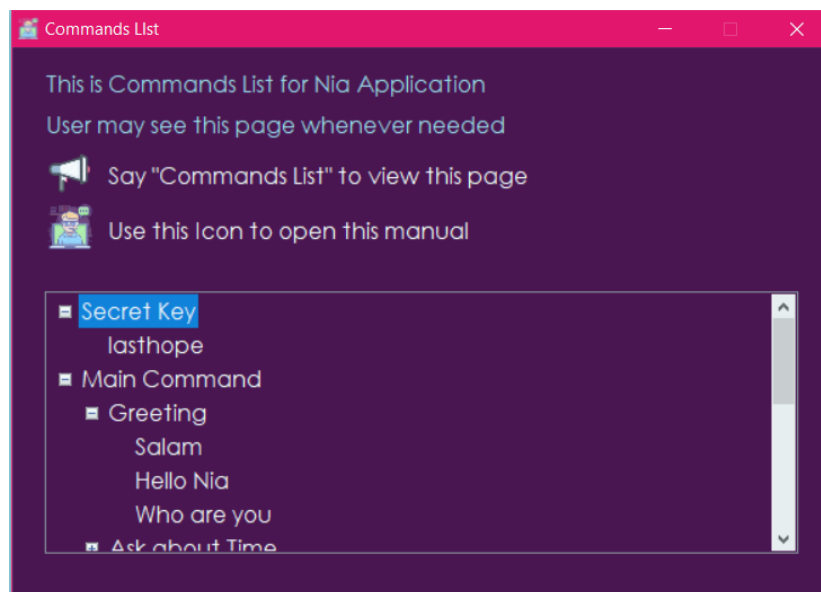
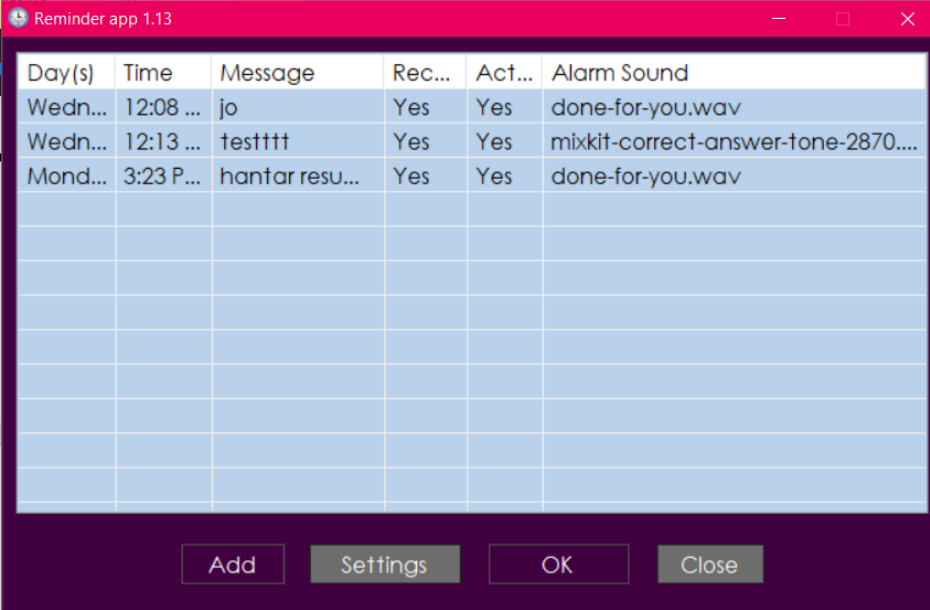


Figure 3.2-15 Help Section for List of Commands

Figure shows the user interface for help section for commands list. User may take a look on how to say to the system so the system able to understand and give a good feedback to user. User just only need to click '+' to dropdown the list.



Day(s)	Time	Message	Rec...	Act...	Alarm Sound
Wedn...	12:08 ...	jo	Yes	Yes	done-for-you.wav
Wedn...	12:13 ...	testttt	Yes	Yes	mixkit-correct-answer-tone-2870....
Mond...	3:23 P...	hantar resu...	Yes	Yes	done-for-you.wav

Buttons: Add, Settings, OK, Close

Figure 3.2-16 Schedule List

Figure shows the user interface output design for schedule. User can view the list of reminders create in this schedule page. User can select the any task to edit or delete. Also, by saying ‘remind me’ or ‘ingatkan saya’ this schedule will display at first so user would not make double activities or tasks.

3.2.3 Database Design

3.2.3.1 Conceptual and Logical Non-Database Design

This system speech recognition process conceptual design is using a dialogue pattern. This pattern being chose to as for showing the concept of the system. Dialogue between user and system is both been recognized each other. This happened when system try to greet user and user give feedback by voice command or if system is on speech but user want it to stay silent. The dialogue language is determined by developer and user can understand it.

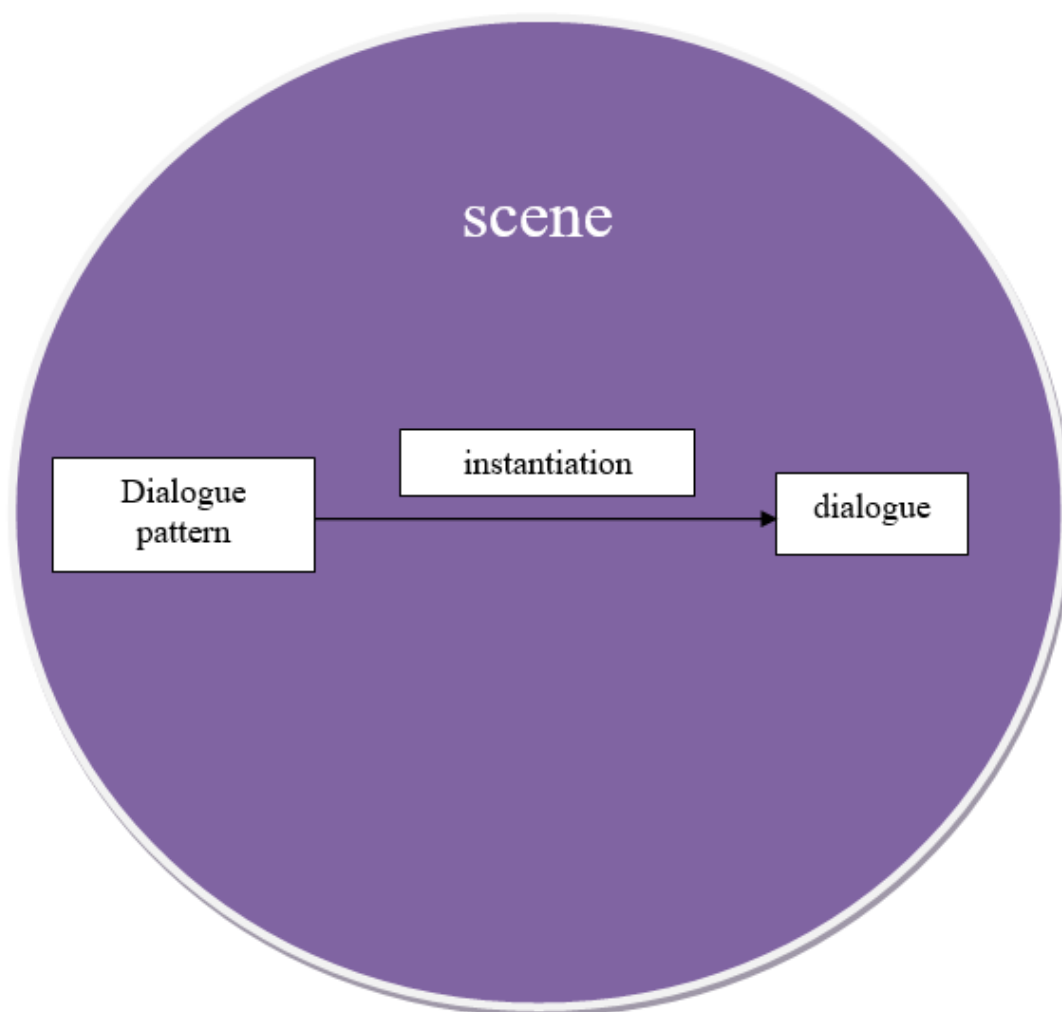


Figure 3.2-17 Diagram on User Interaction based on Dialogue Pattern Concept

3.3 Flow Process of Nia

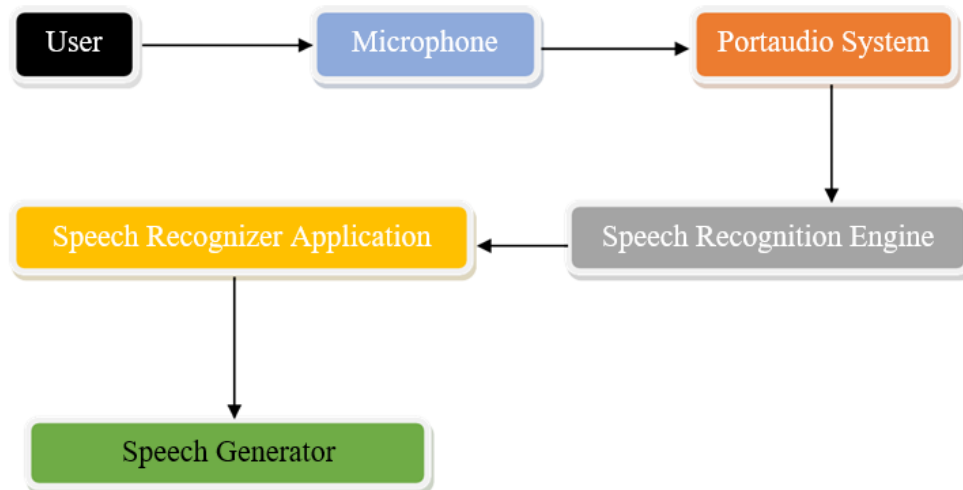


Figure 3.3-1 Flow chart process of system

- **User:** Users speak into the microphone. Example: “Play me some horror movie”
- **Microphone:** Capture sound waves and convert it to generate electrical impulse.
- **Portaudio System:** The cross-platform audio I/O library that allows playing or recording the audio received by user.
- **Speech Recognition Engine:** By using Google Speech Recognition, it will recognize speech input from the microphone. This is where the digital sound signal converted to phonemes.
- **Speech Recognizer Application:** Application process the recognized words as text input and display it.
- **Speech Generator:** Text input displayed will be converted to speech. This is where the system will do interactive AI by replying to the user while opening YouTube platform to display horror movie.

3.4 Features Design of Nia

- **Reminders**

The ability to set reminders is the main idea development of Nia, that also include the time based, people based & event location based. It also eligible for setting the alarms and voice over the current time. This includes for adding some notes for future reference.

- **Improved Search**

The Internet may be a ocean of information, but that is not the only resource available to Nia. Nia also able to play desire movies or music's videos just by asking her to play it and it will auto-play.

- **Offline function**

Nia able to be used while offline or no data. This could help user even they not online or data has been used out, as Nia could still remind the task or activities scheduled.

- **Open apps on computer device**

Nia could be used for opening any other apps or software like notepad, Microsoft office software, file folder and more. This functionality may save user's time as they are doing another thing, just voice over to open the desire apps, Nia will try to help by opening it.

3.5 Conclusion

In this chapter, design and architecture of the system are carefully explained and explained in order to achieve the objectives of this project. The design phase is done for obtaining a clearer view on how the system of the project should looked like. The next chapter will be on the implementation phase of this project development.

Chapter 4: IMPLEMENTATION

4.1 Introduction

After completing on the designing phase, and then this is where the project for Nia Remind Me! - AI Personal Assistant system will be implemented. In this chapter and phase is where the developer will start on writing the coding based on the entire requirement that been state in the previous stage of system development life cycle. The primary activity involve in this stage is on developing the small database for dictionary and code for the Nia Remind Me! - AI Personal Assistant system. The developer will create all the necessary data used in the database. Other than that, it will also look into on creating the suitable interface that needed so the system can be developed successful and fulfil all the system requirement and user needs. All the process on developing and implementing the Nia Remind Me! - AI Personal Assistant system will check thoughtfully so that this system will successfully being produce with high quality and useful to the users.

4.2 Software Development Environment setup

In term on implementing the Nia Remind Me! - AI Personal Assistant system, it requires to setup and installing all the necessary software needed for the system implementation. Before this, PyCharm Software was used in order to build the system without database or interface, and by using google speech api and pocketsphinx for recognize words. But for this final outcome, the needed software such as Visual Studio 2019 software and text file as portable database to show the user-friendly and offline mode.

4.2.1 Visual Studio Installation Step

Visual Studio is a free integrated development environment (IDE) and developer tools that enable the user to build web applications, cloud services as well as modern application for Android, iOS, or Windows. Therefore, here will look into the easy step on setting and installing the Visual Studio 2019 software in the computer.

Step 1:

There are many versions of Visual Studio, any version can be downloaded at the Microsoft Visual Studio websites (<https://visualstudio.microsoft.com/vs/older-downloads/>). User just need to choose suitable version that match with software and hardware requirements. For this AI Personal Assistant system, visual studio 2019 would be used to build the desktop application that suitable with any CPU and operating system. Click the purple button “Free download” to start downloading the software.

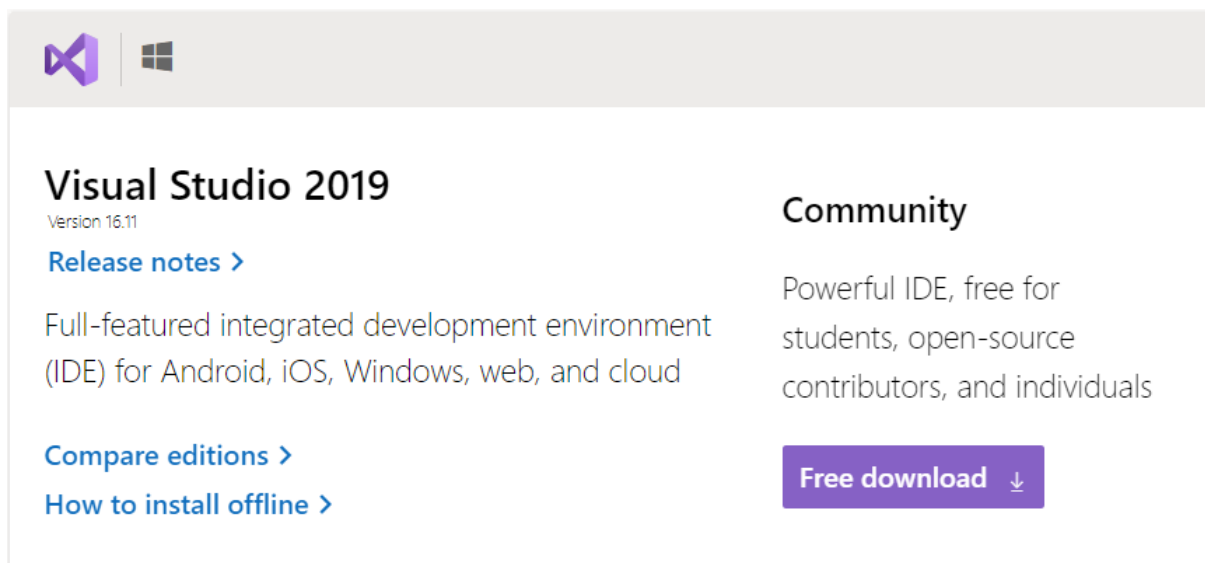


Figure 4.2-1 Download Visual Studio 2019

Step 2:

Start installation of Visual Studio 2019 by right click on the downloaded file and select “Run as administrator”. User then will be asked for permission to continue and click “Yes” to proceed the installation.

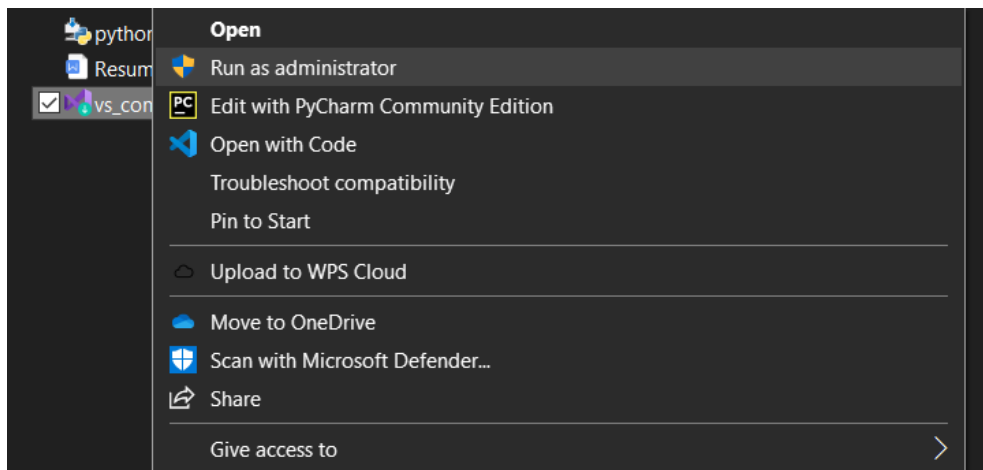


Figure 4.2-2 Visual Studio Run as Administrator

Step 3:

User will be asked to accept the Privacy Statement and Software License Terms, so just click “Continue”.

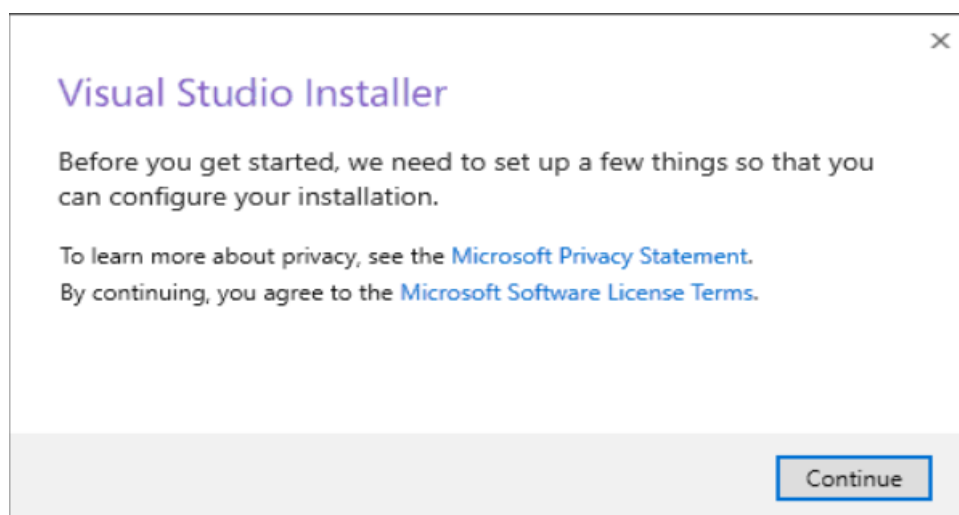


Figure 4.2-3 Visual Studio Installer

Step 4:

Installer start downloading and installing the required files for few minutes in figure 9. While figure 10 shows after the downloading and installing process is done and will proceed to the configuration.

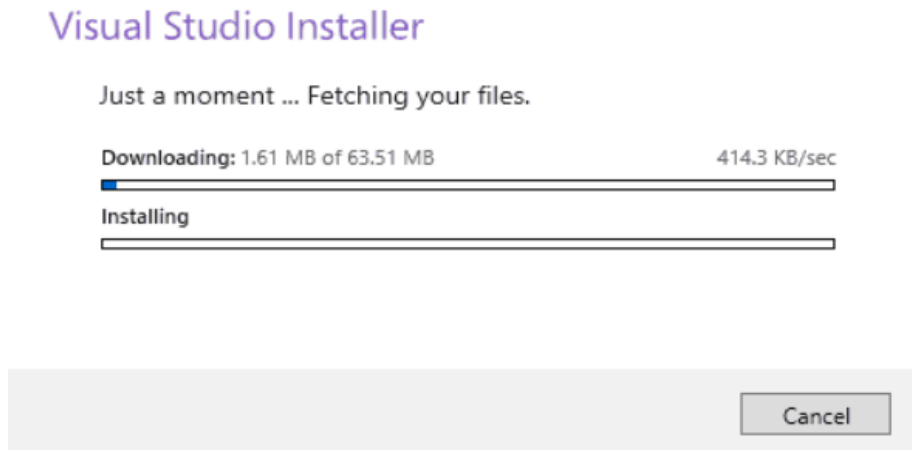


Figure 4.2-4 Visual Studio Installer Download and Installation

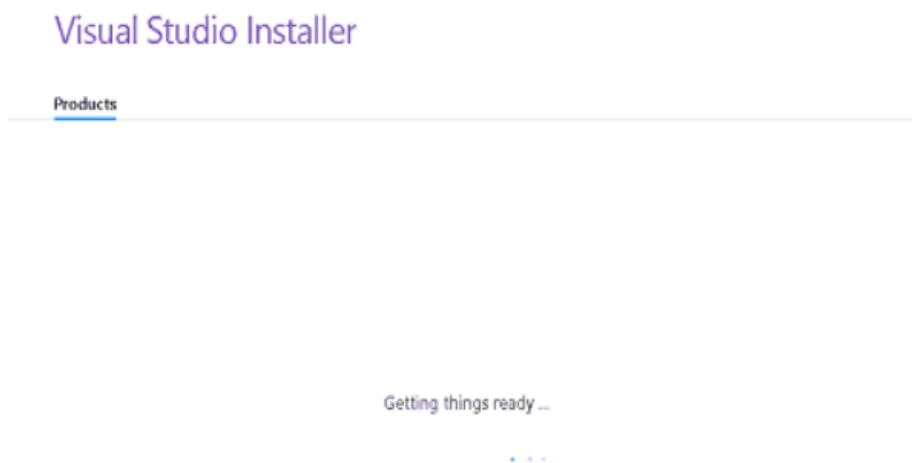


Figure 4.2-5 Visual Studio Installer Done Processing

Step 5:

Installer will ask user to install the features that interested to be used as there are four options.

1. Workloads
2. Individual Components
3. Language Packs
4. Installation Packs

In this case to build and run Nia Remind Me! AI Personal Assistant System, developer has chosen workload for .NET Desktop development and ASP.NET and web development (additional for pc that has sufficient storage space). Click install to proceed. After installation process done, user need to sign-in as community edition will expired after 30 days if user do not register or sign-in the Microsoft account.

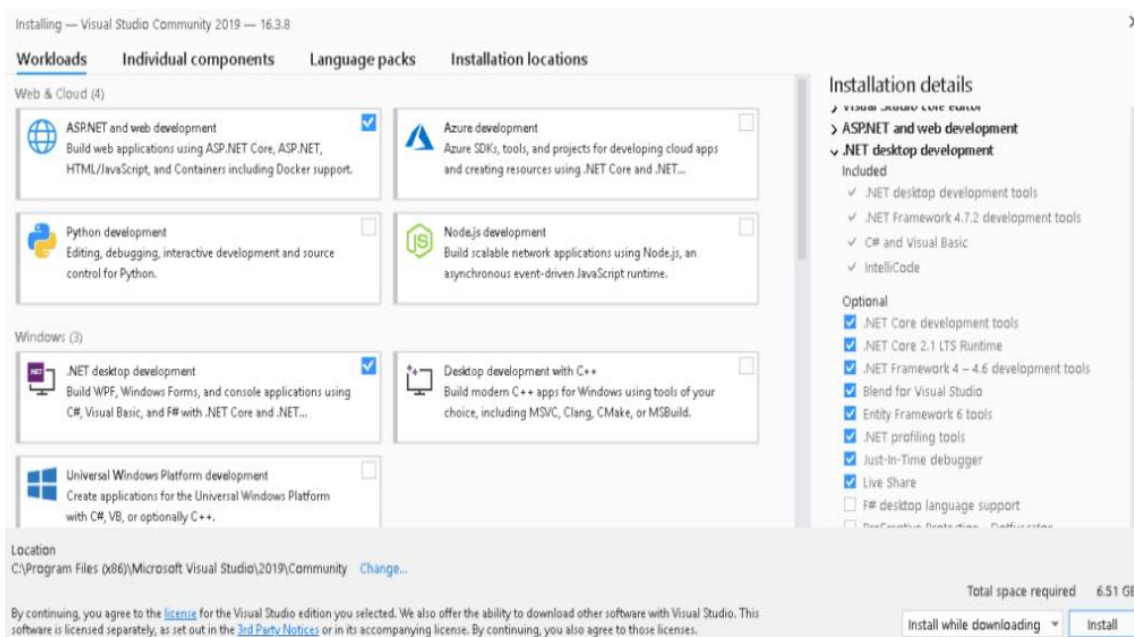


Figure 4.2-6: Main Menu Visual Studio Installer

Step 6:

Open Visual Studio 2019, the homepage will display and select create a new project. Developer may search for templates or project packages or choose Windows Forms App (.NET Framework) to start creating the Nia Remind Me! AI Personal Assistant System interface and program code. The programming language used to build this application is C#.NET.

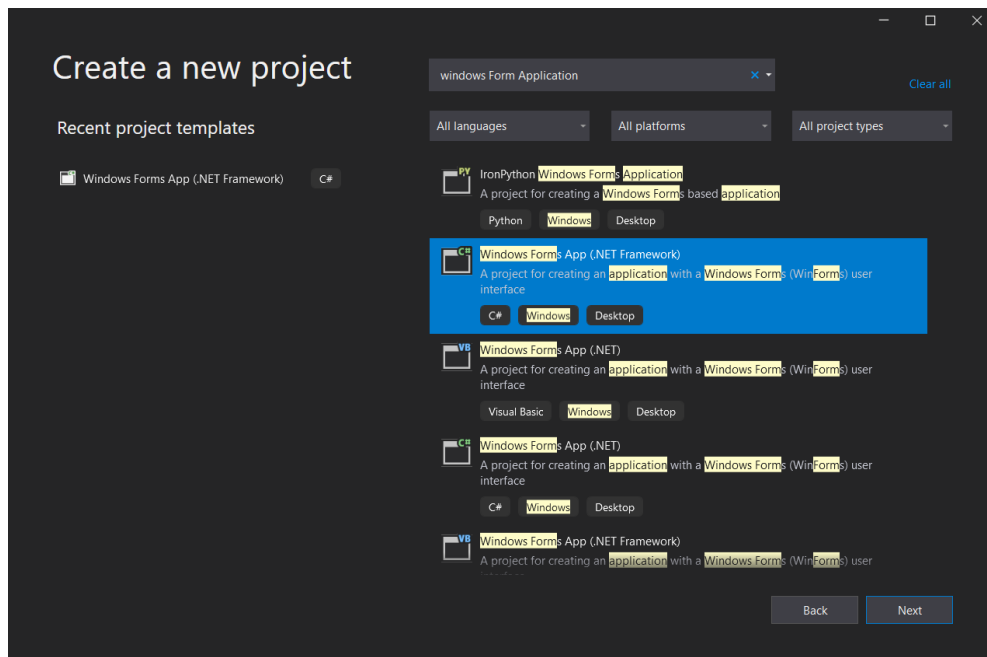


Figure 4.2-7 Create new project (Visual Studio 2019)

4.2.2 Text file Database

Text file is a file that able to store any data or information as it is also portable can be used by everyone except for encryption type file. By using this text file, combine with the system made in Visual Studio can be as database project. The database can be created from scratch or import any existing database. In this case, to build Nia Remind Me! AI Personal Assistant System, the database created to start from scratch as the database must contain all words or commands that will be used and understand by the system.

4.2.3 Database Environment Setup

Database is a collection of meaningful raw data. Database is one of the important keys in any project or system development that need to look into it. A lot of data and information need to be collected and managed properly. A good management of data can ensure the system effective, efficiency and productivity. Hence, it needs a database system that can hold and safely save all this important information. In this project on implementing the Nia Remind Me! AI Personal Assistant system, the tools that been use to manage the database is using text file as a database. Database can be created in two ways as below either step 1 or step 2 as both of them lead to the same result.

Step 1:

User can go to Solution Explorer and right click on the project name. Choose the add item so the many option will display, search for text file and rename it and create. After user create the file. User also need to go to the file folder of the project, select bin >> Debug and create new text file with the name as created in the visual studio. Hence, everything that we type in the text file at Visual Studio also will update in the text file at debug folder.

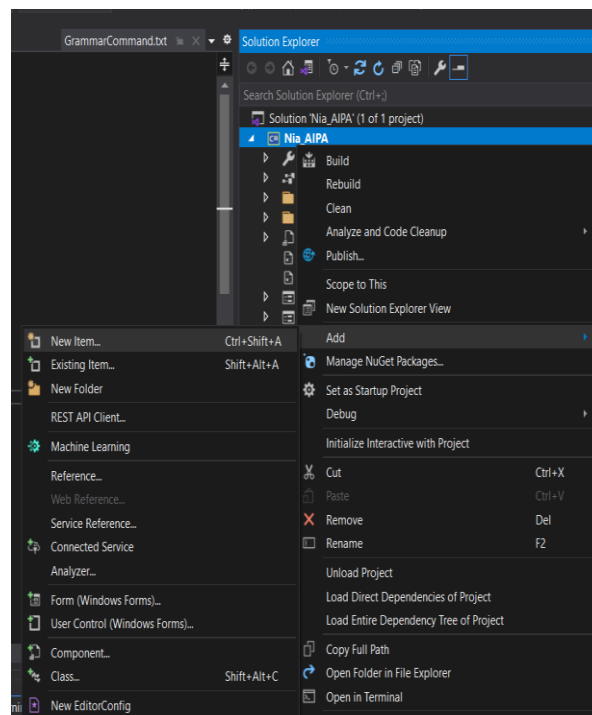


Figure 4.2-8 Step A to Create Textfile in Visual Studio

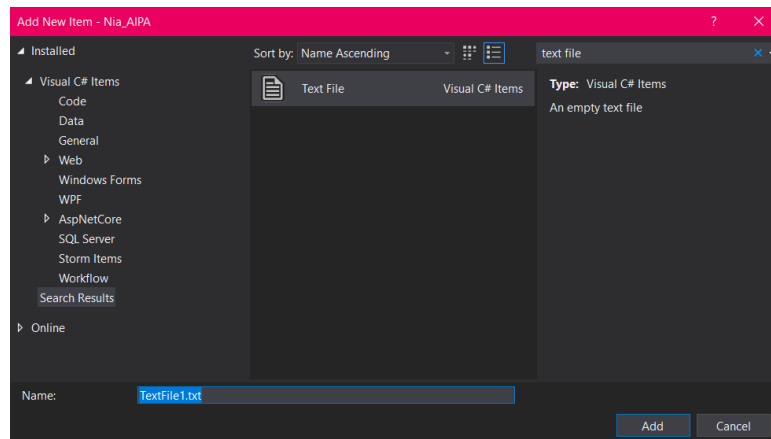


Figure 4.2-9 Step B to Create Textfile in Visual Studio

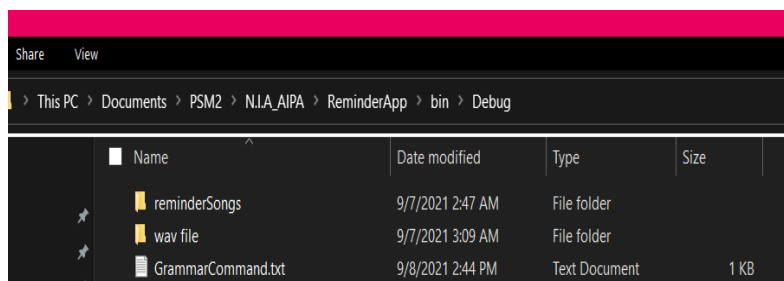


Figure 4.2-10 Step C to Create Text file in Visual Studio

Step 2:

User can directly create new file at the Debug folder, insert all the commands. But in the coding part user to declare the environment directory to that text file.

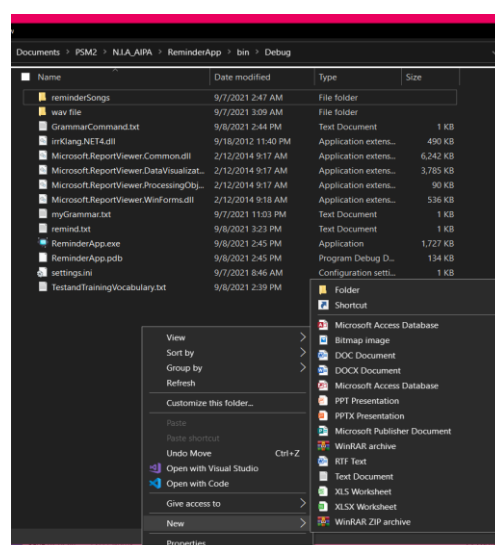


Figure 4.2-11 Step 2 Create Text file in Directory folder

4.3 Software Configuration Management

4.3.1 Configuration environment setup

In this project, Visual Studio is used to code a major part of the system, which consists of coding for the application and storing lists of commands for the use of system while interacting. The language used in Visual Studio is C#. There are about 6 additional using namespaces required during the development to declare some related objects such as using , using System.Speech to as reference for doing a speech process either text-to-speech or speech-to-text, using System.Diagnostics to interact with system processes, event logs, and performance counters that provides classes for user use, using System.Speech.Recognition to do the speech recognition process by declaring speech recognition engine object or class, using System.Speech.Synthesis to allow system to talk back to user or enable a two way communication process, using System.IO allow to reads and write data to files or data streams of the classes or object declared in the code, and using System.Media allow for playing sound files and accessing sounds provided by the system with related classes.

Text File that uses as database in the Visual Studio 2019 is used to store dictionary and vocabulary of words that will be as command to have conversation with the system. This kind of database contain vocabulary, dictionary as well as commands word to ease user interact with the system. If there are unrecognized word, user may add on its own the desire word so that system able to understand what user is talking about in the future. The configuration of database is only based on multiple tables as each table can only be used for certain features so that there will no mixed up and confusion occur.

Through these configurations, both Visual Studio and Text file Notepad as a database may allow user to use the system application in one set. The commands list also enable editing in any type of language that is available in the user's pc. The web browser used can be customized to any browser along with the setup of link and internet connection for searching more information. This application can run without the Internet if user would like to do things other than email or browsing process, where it can open document or other application in the computer, do the reminder, play video or music, shut down/restart the computer and many more. Internet connection is needed when only it is needed.

4.3.2 Version Control Procedure

For version control, the system uses the latest release for all dependencies in 2021. Microsoft Visual Studio Community 2019 version 16.8.4 is used for the Visual Studio IDE tools, C# 3.8.0 is used for the programming language and the Microsoft .NET Framework Version 4.8.04084. Text File Notepad version 10.0 used for portable database storage. The latest releases of Visual Studio are 16.11 are not chosen because the new features available are not required by the system implementation, and no extra configuration is needed to use the features. In conjunction with these versions, the system is up to date at the time of coding.

4.4 Implementation Status

Table 4.4-1 Modules Developed Status

Module	Description	Modification	Completion
Voice recognition	A module where the system should be able to recognize user's voice.	In PSM 1, system built with Python language and voice recognition done by Python library. In PSM 2, system built with C#.Net language and voice recognition done by system speech namespace and reference	Completed
Speech-To-Text & Text-To-Speech	A module where system able to convert speech voice by user to text and directly convert the speech to text by giving feedback to user	In PSM 1, system built with Python language and voice recognition done Python library for Speech-To-Text &Text-To-Speech operation. In PSM 2, system built with C#.Net language and voice recognition done by system speech namespace and reference for Speech-To-Text &Text-To-Speech operation	Completed
AI System Construction	A module where system able to able remind user at the correct day for meaning time	In PSM 1, system built with Python language and no reminder section were added. In PSM 2, system built with C#.Net language and reminder section is development that contain adding tasks and list of schedule.	Completed

Evaluation:

The components of this system are the main form, reminder form, open various application available in the computer, close any/current application that is not in used. Ask about simple task such as date, time, upcoming schedule, update the desire commands in the system and shut down or restart the computer.

The main form is a module in the form of interface that will appear at the first place for user to ask anything. Just voice command it out and the system should do the task. Implementation of this module took about 2-3 weeks in order to list out all the word using text document as draft before applying it to the local database.

As the main form were tested, configuration of all the words or commands listed also been updated from time to time in order to make it functional. In this form also user able to ask for simple task like open browser for searching some information, open any application in the computer such as Microsoft Office software, current date, day, and time, shut down or restart the computer or the Nia system itself. If user want to change the way it communicates with the system, user can ask for editing from the common commands to their desire commands. The thing is user can also insert multiple language of commands that use universal alphabet and numbers but not in characters.

The reminder form is a module in the form of interface that will appear only when user would like Nia to note something by voice command. And the note will be saved in the list of schedules. Users need to tell the time, date and what is event or activity of the day. User also able to cancel or reset the current notes or reminder. It took about 2 weeks to settle on voice command in the note space using custom made dictionary and vocabulary.

4.5 Conclusion

In this chapter, the tools used in completing this project and the development setup are carefully explained and elaborated in order to achieve the objectives of this project. The implementation phase is done to achieve the good end result of this project. The next chapter will be on the testing phase of this project.

Chapter 5: TESTING

5.1 Introduction

This chapter will explain about the testing phase of this project. This include how the test was implementation, where it happens and some kind of user acceptance testing. In the testing phase, a precision evaluation will be done on the system. The goal of this phase is to check for the practicability of the system, testing is done multiple times to verify that the outcome is maintained, and the system is ready to be used. From this result analysis, it can determine the reliability of the system where it will be included at the end of this chapter.

5.2 Test Plan

5.2.1 Test Organization

The personnel involved in the testing phase is me. In actual situation, anyone which is the user include developer will be as end user giving the voice command to do the tasks. In this case, developer-me myself and one user-my sister will evaluate the system following the questions stated in the User Acceptance Testing.

5.2.2 Test Environment

To test this system, a laptop connected to the Internet is needed to access the browser related to the suitable voice command. The environment of testing to be carried out is user's computer. To test the voice command precision, system can be tested with some background noise either fan noise, other people voice, or else.

5.2.3 Test Schedule

Testing implementation was about 10++ time for every command in order to test the commands and accurateness of speech recognition process. The system may learn by itself recognize the way human talk, their accent if many voice commands test out.

5.3 Test Strategy

The strategy is testing by using top-down approach where from knowledge source that being used are manually created as dictionary and vocabulary of the system. All the speech will be matched to the self-build dictation grammar and the system will recognize the words that only created in the database.

5.3.1 Classes of tests

Functionality test

- Open and close application in the computer
- Save Notes as Reminder
- Show schedule list
- Offline mode work

5.4 Test Implementation

5.4.1 Experimental / Test Description

Table 5.4-1 Test Case Main

Test Case ID	Test Case Description	Input Data	Expected Result
1	User says to greet Nia	Valid : “Hello Nia” ; “Hey Nia”;	System will reply “Hi” followed by user’s computer name.
		Invalid: “Hoy”; “Hola”; “halo”	No response. System will stay silence.
2	User says to open application in the computer	Valid : “Open Word”/ “Buka Word”/ “Word” ; “Open Notepad”/ “Notepad”/ “Buka Notepad”;	System will open the application based on user specified command word.
		Invalid : “Show word doc”; “Show notepad”	No response. System will stay silence.
3	User says to ask about current date	Valid: “what date is today”; “harini berapa hari bulan”; “date today” ; “today’s”	System will say the current date or today’s date.
		Invalid: “date” ; “current date”	No response. System will stay silence.
4	User says to ask about current time	Valid: “sekarang pukul berapa”; “what time now”; “now what time”	System will say the current time of the day.
		Invalid: “give me current time”; “show me time”	No response. System will stay silence.
5	User says to create a reminder or schedule	Valid: “remind me”; “ingatkan saya”; “ ”	System
		Invalid: “catat”; “tulis”, “write this”	No response. System will stay silence.

5.4.2 Test Data

All data tested are in the portable text file as database. All the word listed database being tested by voice command. The accurateness can be assured after all the word and phrase tested during speech recognition process it where will be displayed in the testing box whether the spoken word displayed as expected. Total phrases that been tested are 341 which involved Bahasa Melayu has 154 phrases, English has 118 phrases and for main command involved both language has 67 phrases. The testing result analysis for accuracy is manual calculated since the phrases and dictionary are custom made. The result will be shown at Result and Analysis section below.

5.5 Test Results and Analysis

Test implementation has been conducted to test the system functionalities and test case below shows the test results.

Table 5.5-1 Test result and analysis

Test Case ID	Test Case Description	Input Data	Expected Result	Actual Result	Remarks
1	User says to greet Nia	Valid : “Hello Nia” ; “Salam”;	System will reply “Hi” followed by user’s computer name.	OK	System replied Hello followed by username of the computer device
		Invalid: “Hoy”; “Hola”; “halo”	No response. System will stay silence.	OK	No remarks
2	User says to open application in the computer	Valid : “Open Word”/ “Buka Word”/ “Word” ; “Open Notepad”/ “Notepad”/ “Buka Notepad”;	System will open the application based on user specified command word.	OK	System open Microsft Word immediately and followed by Notepad application
		Invalid : “Show word doc”; “Show notepad”	No response. System will stay silence.	OK	No remarks
3	User says to ask about current date	Valid: “what date is today”; “harini berapa hari bulan”; “date today” ; “tarikh harini”	System will say the current date or today’s date.	OK	System replied the correct current date of the day
		Invalid: “date” ; “current date”	No response. System will stay silence.	OK	No remarks
4	User says to ask about current time	Valid: “sekarang pukul berapa”; “what time now”; “now what time”	System will say the current time of the day.	OK	System replied the correct current time of the day
		Invalid: “give me current time”; “show me time”	No response. System will stay silence.	OK	No remarks
5	User says to create a reminder or schedule	Valid: “remind me”; “ingatkan saya”; “ ”	System will open UI Reminder.	OK	System replied what should be reminded and UI reminder opened

		Invalid: “catat”; “tulis”, “write this”	No response. System will stay silence.	OK	No remarks
6	User says to search Google	Valid: “Google”; “search Google”	System will ask what do you want to search for while display the search interface	OK	User suggest to make it in BM if possible for more friendly.
		Invalid: “search”, “cari jap”	No response. System will stay silence.	OK	No remarks
7	User insert secret key	Valid: lasthope	System will say access granted. And proceed to new interface bot.	OK	Use suggest to have this security while trying to open schedule.
		Invalid: Lasthope; LASTHOPE	System will say access denied. Ask user to insert correct secret key.	OK	No remarks
8	User set the notification for Messagebox & Info popup in Reminder Settings	Valid: “checked Messabox”, “checked Info Bubble”	System will remind user by popup info and messagebox	OK	System reminded user by giving popup info and messagebox
		Invalid: “unchecked Messabox”, “unchecked Info Bubble”	System will not give any popup notification	OK	No remarks
9	User click Save button to save reminder	Valid: Click Save button	System will say “Reminder Added” and display it in schedule list.	OK	System said “Reminder Added” and displayed it in schedule list.
		Invalid: Click Cancel	System will close the Add/Edit user interface	OK	No remarks
10	User right click on the list of schedule to edit	Valid: Right Click and click Edit	System will open the Add/Edit interface	OK	System will open the Add/Edit interface
		Invalid: Left click	System only highlight the selected task	OK	No remarks

User Acceptance Testing

Table 5.5-2 User Acceptance testing

Project Name:	Nia, Remind Me! – AI Personal Assistant System	Version Number:	1.0		Programmer’s Name:		Ainin Sofiya Hisham	Machine Specification:	<ul style="list-style-type: none">• OS Name: Microsoft Windows 10 Home Single Language• Microsoft Visual Studio Community 2019• SQL Server Local Database	
Client’s name:	Not specified				Test date:		30 August 2021			
Test Case Id	Test Case Description				Rating					Remark
					1	2	3	4	5	
TC S01	<u>COMPATIBILITY</u> Does the system able to perform effectively while sharing a common environment?									
TC S02	Does the system able to maintain the consistencies if there is any changes made in any of the hardware related with the system?									
TC S03	<u>FUNCTION SUITABILITY</u> Does the functions in the system covers all the specified tasks and user objectives?									

TC S04	Does the system provide correct and accurate output based on the inserted input?						
TC S05	Does the functions ease the user to accomplish their required task and objectives?						
TC S06	<u>PERFROMANCE EFFICIENCY</u> Does the system able to responds with appropriate processing time and throughput rates?						
TC S07	Does the resources used by the system especially when performing its functions meet the requirements?						
TC S08	Does the system have maximum limits of used?						
TC S09	<u>PORTABILITY</u> Does the system able to be adapted with the new changing environment?						
TC S10	Does the system can be installed easily and does it support the system?						
TC S11	Does the system able to be replaced with the current system?						

- **Compatibility**

As for compatibility of the system, TC S01 is the ability of the system to perform effectively while sharing a common environment. All two users are rating on box 3. For first remarks is because using new version of software application. This is because the Visual Studio is the tools that used to develop system and it has low minimum requirements where it perform effectively while sharing a common environment. Second remarks are commented no remarks because the user tester have nothing to say and for third remarks is compatible because using Visual Studio new version. Because with new version, it able to share with a common environment that also used the new version. TC S02 is the ability to maintain the consistencies if there are any changes made in any of the hardware related with system. All two users are rating on box 3. For first remarks is no comment because maybe the user has nothing to remarks and for second remarks is using new version and compatible with minimum requirements and third remarks is the software used is compatible to any kind of hardware. This is because while testing, the system is tested on other workstations and it still work smoothly.

- **Function Suitability**

As for function suitability, TCS03 is about functions of the system covers all the specified task and user objectives. 1 out of 2 are rate on box 2 while the rest is rate in box 1. The remarks both user that rate box 2 and box 1 commented on the same things where the system cannot retrieve the data from the database, and it cannot be saved as permanently in database. Once the system is exit, all the data saved are gone and refresh to 0. While the other user that rate box 2 comment that there is no help features. TCS04 is about the function of the system can provide correct and accurate output based on the inserted input. 1 out of 2 rates on box 4 and the remarks is good and accurate and no remarks only. This is because while testing the system able to function and do as told and correct output after speaking up the correct commands. TCS05 is about whether the function of the system can ease the user to accomplish their required task and objectives. 1 out of 2 rated on box 3 while the rest rated on box 2. For both user that rated on box 3, their remarks is yes. Easy to save, delete, edit trough database grid view. And for the one that rated on box 2 remarks where no. because not save the data permanently.

- **Performance Efficiency**

As for performance suitability, TCS06 is about the system able to respond with appropriate processing time and throughput rates. Both rate on box. The one that rate box remarks that yes. Time taken to process not too long. Take less time to open another form or any application. TCS07 is about the resources used by the system especially when performing its functions meet the requirements. It is about 1 out of 2 rated on box 2 and the rest rated on box 4. The one rated on box 2 remarks yes, but not all the data inserts are permanently saved. The other one rated on box 2 remarks is meet the requirements but not all & cannot retrieve data (the schedule or reminder have been set). TCS08 is about the system have maximum limits of used. All the user tester rated on box 3 where the remarks are all the same which is no remarks.

- **Portability**

TCS09 is about does the system able to be adapted with the new changing environment. It is about 2 out of 2 rated on box 4. This is because all of them think that the system can adapt with the new environment since using new version of software application and the rest is no comments or remarks. TCS10 is about does the system can be installed easily and whether it support the system. There are 2 out of 2 rated on box 4. The first remarks rated on box 4 is easy because using standard software application which requirement low specification. TCS11 is about the system able to be replaced with the current system. Each of user differently rated on the different box which are box 3, and box 4. This is because most of the user think that the proposed system can be replaced with current system because use common software application.

5.5.1 For fundamental and analysis project

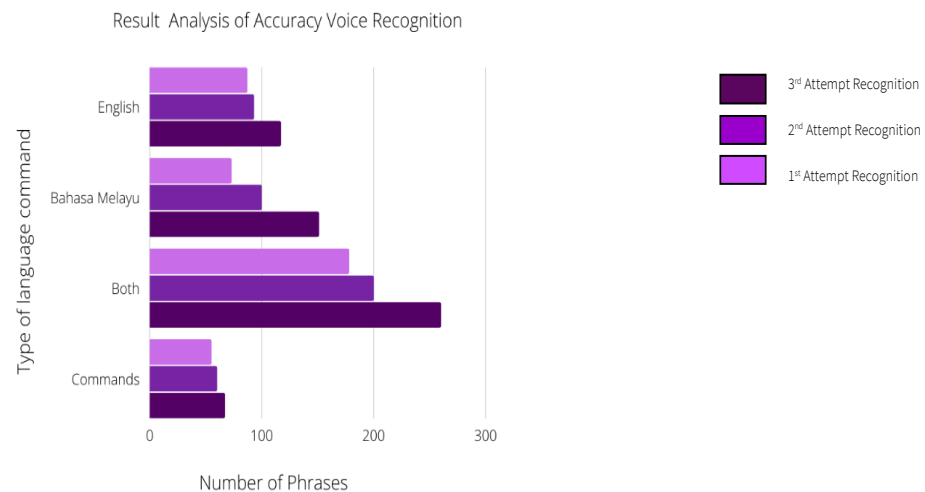


Figure 5.5-1 Accuracy of Voice Command In System Development

Above Figure shows the result analysis bar chart on type of language command vs number phrases been tested in order to calculate the accuracy of the system in voice recognition process. Total phrases for English are 118 and for the 1st attempt, the system only recognize 87 phrases correctly, 2nd attempt got 93 phrases detected correctly and 3rd attempt is 117 phrases. Total phrases for Bahasa Melayu are 154 phrases and for the 1st attempt, the system only recognize 73 phrases correctly, 2nd attempt got 100 phrases detected correctly and 3rd attempt is 151 phrases. Total phrases for both are 272 phrases and for the 1st attempt, the system only recognize 178 phrases correctly, 2nd attempt got 200 phrases detected correctly and 3rd attempt is 260 phrases. Total phrases for voice command are 67 phrases and for the 1st attempt, the system only recognize 55 phrases correctly, 2nd attempt got 60 phrases detected correctly and 3rd attempt is 67 phrases which is fully function. Based on the result, we can conclude that there is improvement for voice recognition as the system try to self-learning every single phrase and achieve above 90% average of accuracy. The accuracy calculated using the $\frac{[\text{overall total of phrases provided and used}]}{[\text{phrases detected by system}]}$.

5.6 Conclusion

In this chapter, the testing phase is described on how it is done to make sure the functional requirement run smoothly. From the analysis, the system works well with as long as user frequently give voice command and test their voices so the computer and system can learn to adapt with the accent and word language used.

Chapter 6: CONCLUSION

6.1 Observation on Weaknesses and Strengths

6.1.1 Strengths

System able to recognize voice in Bahasa Melayu. This is because some the command is inserted in Bahasa Melayu and the system able recognize correctly. System able to work while offline mode which mean no internet connection as long as no search or browsing action. User can test all the possible words at test and training section in order to for system to learn and have better reorganization on user's voice and accent.

6.1.2 Weaknesses

System cannot adapt on too loud background noise. This could unexpected activity to the system as system try to recognize all the sound surround it. For reminder section, user need to do typing a bit if system not able to recognize the correct word in command.

6.2 Propositions for Improvement

System may improve the language to more user friendly used. If in Malaysia user may use Bahasa Melayu commands. As developer, can provide more proper database all for each section like main section, reminder section, searching or browsing section where user can simply ask in Bahasa Melayu.

Related to the Bahasa Melayu commands, as for in reply to the user, it will be better if system can reply in Bahasa Melayu. So that, system can be more user friendly and adapt with current user situation.

This project system has only one official theme colour. In the future enhancement, the theme of the system should be able to customize following the user's mood or favourite.

Lastly, system can be improved in voice commands for typing in other application like word or PowerPoint. Hence, those application can be control or fully utilize by using voice commands.

6.3 Project Contribution

The contributions that this project can make are listed below:

- **Contribution to University**

This project can be a reference to other Final Year students that aim to develop a time-series forecasting project.

- **Contribution to Individual**

The project development can provide technique in save time, systematic life with organized daily's activity.

- **Contribution to End-user**

The system can help the end-user less stress with organized schedule activities. This project also might be useful to the disabilities (with no hands or arm or legs) to do computerised work using voice commands.

6.4 Conclusion

In this last chapter, the strengths and weaknesses of this project system has been listed out and a discussed-on improvements were elaborated to cater to the weakness mentioned. Although the project has its flaws, it can also contribute its knowledge to university, individuals, and the project's end-user. Furthermore, the project has met all 3 objectives set conclusively at the beginning of project development.

REFERENCES

- [1] Cherry, K., 2021. Reasons Why People Forget. Verywell Mind, [online] accessed 5 March 2021, <<https://www.verywellmind.com/explanations-for-forgetting-2795045>>
- [2] Bekker, A., 2021. 5 Types of AI to Propel Your Business. ScienceSoft., [online], accessed 6 March 2021, <<https://www.scnsoft.com/blog/artificial-intelligence-types#:~:text=You%20can%20opt%20for%20any,or%20wisely%20combine%20several%20ones>>
- [3] Schrader, J., 2021. Why We Forget | The Benefits of Not Remembering. Psychologytoday., [online], accessed 11 April 2021, <<https://www.psychologytoday.com/us/blog/definingmemories/201706/why-we-forget>>
- [4] Johari, J., 2007. Chapter 3 Project Methodology. Dspace.unimap.edu.my., [online], accessed 11 April 2021, <<http://dspace.unimap.edu.my/xmlui/bitstream/handle/123456789/3057/Methodology.pdf?sequence=4>>
- [5] Thomas, M., 2021. 7 Dangerous Risks of Artificial Intelligence. Built In., [online], accessed 16 April 2021, <<https://builtin.com/artificial-intelligence/risks-of-artificial-intelligence>>
- [6] Marr, B., 2020. What Are The Negative Impacts Of Artificial Intelligence (AI)?, [online], accessed 17 April 2021, BernardMarr. <<https://bernardmarr.com/default.asp?contentID=1827>>
- [7] O'Boyle, B., 2021. What is Alexa and what can Amazon Echo do?. Pocket-lint. [online], accessed 18 April 2021, <<https://www.pocket-lint.com/smart-home/news/amazon/138846-what-is-alexa-how-does-it-work-and-what-can-amazons-alexa-do>>
- [8] admin@topfaida.com, 2019. Advantages and disadvantages of google assistant., Topfaida.Com., [online] Accessed 7 May 2021, <<https://topfaida.com/advantages-and-disadvantages-of-google-assistant/>>

- [9] Pocket-lint. 2020. What is Siri and how does Siri work?. [online], accessed 9 May 2021, <<https://www.pocket-lint.com/apps/news/apple/112346-what-is-siri-apple-s-personal-voice-assistant-explained>>
- [10] Contributor, G., 2018. 9 reasons to choose Agile Methodology for Mobile App Development. Packt Hub., [online], accessed 15 May 2021, <<https://hub.packtpub.com/9-reasons-to-choose-agile-methodology-for-mobile-app-development/>>
- [11] Gonfalonieri, A., 2018. How Amazon Alexa works? Your guide to Natural Language Processing (AI). Medium., [online], accessed 25 May 2021, <<https://towardsdatascience.com/how-amazon-alexa-works-your-guide-to-natural-language-processing-ai-7506004709d3>>
- [12] Marigold, S., 2019. Voice Command System Using C# .Net Application with Artificial Intelligence. 7th ed. , Blue Eyes Intelligence Engineering & Sciences Publication, p.3., [ebook], accessed 7 September 2021, <<https://www.ijrte.org/wp-content/uploads/papers/v7i6s4/F10850476S419.pdf>>
- [13] Adegeo, A., Nschonni, N., Wagner, B. and Gewarren, G., 2017. How to: Extract the Day of the Week from a Specific Date. Docs.microsoft.com., [online], accessed 6 August 2021, <<https://docs.microsoft.com/en-us/dotnet/standard/base-types/how-to-extract-the-day-of-the-week-from-a-specific-date>>
- [14] Docs.microsoft.com. 2021. System.Diagnostics Namespace. [online], accessed 12 August 2021, <<https://docs.microsoft.com/en-us/dotnet/api/system.diagnostics?view=net-5.0>>
- [15] www.javatpoint.com. 2021. C# FileStream - javatpoint. [online], accessed 12 August 2021, <<https://www.javatpoint.com/c-sharp-filestream>>
- [16] www.javatpoint.com. 2021. C# System.IO Namespace - javatpoint. [online], accessed 14 August 2021, <<https://www.javatpoint.com/c-sharp-system-io>>
- [17] A, W., 2017. Adding a timer on a game - C# - C Sharp - Countdown., Youtube.com., [online] accessed 16 August 2021 <<https://www.youtube.com/watch?v=DDXC0juDBhg>>

- [18] Thalheim, B., 2010. *Integration of Dialogue Patterns into the Conceptual Model of Storyboard Design*. ResearchGate., [online], accessed 22 August 2021, <https://www.researchgate.net/figure/Instantiation-of-dialogue-patterns_fig2_221269855>
- [19] McCaffrey, J., 2015. Voice Recognition - Speech Recognition with .NET Desktop Applications. Docs.microsoft.com., [online] accessed 23 August 2021, <<https://docs.microsoft.com/en-us/archive/msdn-magazine/2014/december/voice-recognition-speech-recognition-with-net-desktop-applications>>
- [20] Codegrepper.com. 2020. c# open file Code Example. [online] , accessed 26 August 2021, <<https://www.codegrepper.com/code-examples/csharp/c%23+open+file>>
- [21] Youtube.com. 2020. How To Make Reminder App Using Visual Studio C#. [online], accessed 27 August 2021 <https://www.youtube.com/watch?v=Ez5vxXbP_qA>
- [22] C#, R. and Raddi, A., 2017. Reminder using NotifyIcon and Timer C#. Stack Overflow, [online]., accessed 28 August 2021 <<https://stackoverflow.com/questions/43853260/reminder-using-notifyicon-and-timer-c-sharp>>
- [23] Docs.microsoft.com. n.d. System.Globalizaton Namespace. [online], accessed 28 August 2021, <<https://docs.microsoft.com/en-us/dotnet/api/system.globalization?view=net-5.0>>
- [24] Docs.microsoft.com. 2021. System.Collections.Generic Namespace. [online], Accessed 29 August 2021, <<https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic?view=net-5.0>>
- [25] GitHub. 2020. Build software better, together. [online], accessed 9 September 2021, <<https://github.com/topics/reminder?l=c%23>>

APPENDICES

Appendix A: Secret Key Section Form and Codes

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Speech.Recognition;
using System.Speech.Synthesis;
using System.Speech;
using System.IO;
using System.Media;
using System.Data.SqlClient;

namespace ReminderApp
{
    public partial class FrmSecretKey : Form
    {
        SpeechRecognitionEngine listener = new SpeechRecognitionEngine(); //if set the
        language by own used null;
        SpeechSynthesizer MyNia = new SpeechSynthesizer();
        public FrmSecretKey()
        {
            InitializeComponent();
            try
            {
                SoundPlayer sound = new
                SoundPlayer(@"C:\Users\user\Documents\PSM2\N.I.A_AIPA\ReminderApp\bin\Debug\wav
                file\mixkit-correct-answer-tone-2870.wav");
                sound.Play();
                MyNia.Speak("Please type in our secret key");}
            catch (Exception error)
            {
                MessageBox.Show(error.Message);
                MyNia.SpeakAsync(error.Message); }}
        private void LoadGrammarAndCommands()
        {
            string seckey = secretkeytxtbx.Text;
            try{
                if (seckey == "lasthope")
                {
                    this.Hide();
                    FrmNia main = new FrmNia();
                    main.Show();}
                else
                {
                    secretkeytxtbx.Clear();
                    lblHint.Visible = true;
                    pictureHelp.Visible = true;
                    MyNia.SpeakAsync("Access Denied");
                    MyNia.Speak("Please enter correct secret key");}}
            catch (Exception error)
            {
                MyNia.SpeakAsync(error.Message); }}

        private void hopIn_Click(object sender, EventArgs e)
        {LoadGrammarAndCommands();}
    }
}

```

```

ToolTip t1 = new ToolTip();
private void pictureHelp_Click(object sender, EventArgs e)
{
    t1.Show("Command List", pictureHelp);
    FrmCommandList cmdlist = new FrmCommandList();
    cmdlist.Show();
}
}
}

```

Appendix B: Nia Voice Command Initial Form and Codes

```

SpeechRecognitionEngine listener = new SpeechRecognitionEngine(); //if set the
language by own used null;
SpeechSynthesizer MyNia = new SpeechSynthesizer();
SpeechRecognitionEngine startlistening = new SpeechRecognitionEngine();//for
detecting
Random rmd = new Random();
string namE = Environment.UserName; //username of the person who currently
logon into the windows system
System.Windows.Forms.Timer stopListeningTimer = new
System.Windows.Forms.Timer();
public static Speak speak = new Speak();
public FrmNia()
{InitializeComponent();
    try
    {MyNia.SpeakAsync("OK. Access granted");
        SoundPlayer sound = new
SoundPlayer(@"C:\Users\user\Documents\PSM2\N.I.A_AIPA\ReminderApp\bin\Debug\wav
file\done-for-you.wav");
        sound.Play();
        greeting();
        listener.LoadGrammarAsync(new Grammar(new GrammarBuilder(new
Choices(File.ReadAllLines(@"GrammarCommand.txt")))));
        //Eventhandler for recognized text
        listener.SpeechRecognized += new
EventHandler<SpeechRecognizedEventArgs>(nia_SpeechRecognized);
        listener.SetInputToDefaultAudioDevice();
        listener.RecognizeAsync(RecognizeMode.Multiple);
        MyNia.SelectVoiceByHints(VoiceGender.Neutral);}
    catch (Exception error)
    {MessageBox.Show(error.Message);
        MyNia.SpeakAsync(error.Message);}
    stopListeningTimer.Tick += new EventHandler(time_Tick);
    stopListeningTimer.Interval = 1000;}
private void nia_SpeechRecognized(object sender, SpeechRecognizedEventArgs e)
{//statement and command in here
    string speak = e.Result.Text; //executed text after give commands (the
output)

    switch (e.Result.Text)
    {case "Hello Nia":
        case "Salam":
            MyNia.SpeakAsync("Hello. Salam " + namE);
            break;
            //duguya
        case "who are you":
            MyNia.SpeakAsync("I am Nia. Your personal assistant");
            MyNia.SpeakAsync("i can help to do several task in your computer,
reminding you and do schedule for you, search web for you if you have internet
connection");
            break;
            //asking owner name
        case "what is my name":
            MyNia.SpeakAsync(namE);
    }
}
}

```

```

        break;
//time
case "what time is it":
case "now what time":
case "what time now":
case "pukul berapa sekarang":
case "sekarang pukul berapa":
    MyNia.SpeakAsync("Now is " + DateTime.Now.ToString("t"));
    break;
//date
case "what date is today":
case "date today":
case "tarikh harini":
case "harini berapa hari bulan":
    MyNia.SpeakAsync("today is " + DateTime.Now.ToString("ddd dd MMM
yyyy"));
    break;
//shutdown computer
case "shutdown my computer":
    MyNia.Speak("OK. shutting down the computer");
    Process.Start("shutdown", "/s /t 0");
    break;
//restart computer
case "restart my computer":
    MyNia.Speak("OK. restarting the computer");
    break;
//exit Nia application
case "Nia exit":
case "tutup Nia":
    MyNia.Speak("OK. Have a good day. bye-bye!");
    Application.Exit();
    break;
//open application in the computer
case "open Notepad"://notepad app
case "Notepad":
case "buka Notepad":
    MyNia.Speak("opening notepad..");
    Process.Start("NOTEPAD.exe");
    break;
case "open Word"://ms word app
case "Microsoft Word":
case "buka Word":
    MyNia.Speak("opening word...");
    Process.Start("WinWord.exe");
    break;
case "open Powerpoint"://ms powerpoint
case "Ms Powerpoint":
case "buka Powerpoint":
    MyNia.Speak("opening powerpoint..");
    Process.Start("POWERPNT.exe");
    break;
//pause Nia
case "Nia stop talking":
case "Nia pause":
    MyNia.Speak("Alright!");
    if (MyNia.State == SynthesizerState.Paused)
        MyNia.Resume();
    MyNia.SpeakAsyncCancelAll();
    break;
//resume Nia
case "Nia resume":
case "Nia comeback":

```

```

        SoundPlayer sound = new
SoundPlayer(@"C:\Users\user\Documents\PSM2\N.I.A_AIPA\ReminderApp\bin\Debug\wav
file\mixkit.wav");sound.Play();
        if (MyNia.State == SynthesizerState.Speaking)
            MyNia.Pause();
        break;
//reminder Nia
case "ingatkan saya":
case "remind me":
case "schedule please":
case "note me":
case "what is my schedule":
    MyNia.SpeakAsync("OK, Please add day, time and select to make it
active and recurring or not");
    this.Hide();
    FrmLoad load = new FrmLoad();
    load.Show();
    break;
//reminder Nia
case "Commands list":
case "Show Commands list":
case "Show Commands":
case "bagithu commands":
    FrmCommandList commandList = new FrmCommandList();
    commandList.Show();
    break;
case "Close commands":
case "Tutup commands":
    stopListening();
    break;
case "stop listening":
    stopListening();
    break;
case "search":
case "search for":
case "find":
    MyNia.Speak("OK");
    googleSearch();
    break;
case "all processes":
case "show processes":
    showProcesses();
    break;
case "Google":
case "search Google":
    googleSearch();
    break;
case "exit chrome":
case "close chrome":
    exitChromewindows();
    break;
case "open chrome":
    openChromewindow();
    break;
case "test my voice":
    FrmTestandTraining test = new FrmTestandTraining();
    test.Show();
    break;}}
int time = 60;
public void stopListening()
{time = 60;
    MyNia.SpeakAsync("If you need me,call me");

```

```

        listener.RecognizeAsyncCancel();
        stopListeningTimer.Start();}
private void time_Tick(object sender, EventArgs e)
{time = time - 1;
    Console.WriteLine(time.ToString());
    if (time == 0)
    {listener.RecognizeAsync(RecognizeMode.Multiple);
        MyNia.Speak("You may speak");
        stopListeningTimer.Stop();}}
public void greeting()
{int hour = System.DateTime.Now.Hour;
    if (hour >= 5 && hour < 12)
    {MyNia.SpeakAsync("Good Morning " + name);}
    else if (hour >= 12 && hour < 18)
    { MyNia.SpeakAsync("Good Afternoon " + name);}
    else if (hour < 5)
        MyNia.SpeakAsync("Hello " + name + ", you still awake you should go to
sleep, it's getting late");
    else
        MyNia.SpeakAsync("Good Evening " + name);}
private void openChromewindow()
{Process.Start("chrome.exe");}
private void exitChromewindows()
{ endProcess("chrome");}
private void endProcess(string process)
{Process[] processes = Process.GetProcesses();
    foreach (Process currentProcess in processes)
    { if
(currentProcess.ProcessName.ToString().ToUpper().Contains(process.ToUpper()))
        {Console.WriteLine("Process: {0} ID: {1}", currentProcess.ProcessName,
currentProcess.Id);currentProcess.Kill();}}}
//.....Google search.....
public static SpeechRecognitionEngine whatToGoogle = new
SpeechRecognitionEngine();
public static FrmGoogle google = new FrmGoogle();
private void googleSearch()
{google.Show();
    Thread searchForThread = new Thread(new ThreadStart(() =>
speak.searchFor()));
    searchForThread.IsBackground = true;
    searchForThread.Start();
    listener.RecognizeAsyncCancel();
    whatToGoogle.SetInputToDefaultAudioDevice();
    Choices mygrammar = new Choices();
    string[] text = File.ReadAllLines(Environment.CurrentDirectory +
"//myGrammar.txt");
    mygrammar.Add(text);
    Grammar grammar = new Grammar(new GrammarBuilder(mygrammar));
    whatToGoogle.LoadGrammar(grammar);
    whatToGoogle.SpeechRecognized += new
EventHandler<SpeechRecognizedEventArgs>(whatToGoogle_SpeechRecognized);
    whatToGoogle.RecognizeAsync(RecognizeMode.Multiple);}
private void whatToGoogle_SpeechRecognized(object sender,
SpeechRecognizedEventArgs e)
{if (e.Result.Text.ToUpper().Equals("SEARCH"))
    {google.searchGoogle();
        Thread loading = new Thread(new ThreadStart(() => speak.loading()));
        loading.IsBackground = true;
        loading.Start();
        try
        {listener.RecognizeAsync(RecognizeMode.Multiple);}
        catch (System.Exception ex)

```

```

        {MessageBox.Show(ex.ToString());}
        whatToGoogle.RecognizeAsyncStop();}}

//.....WHAT IS COMPUTER PROCESSING NOW.....
private void showProcesses()
{try {Process.Start("Taskmgr.exe");} //what is currently running (not
background running)
catch (System.Exception exce)
{MessageBox.Show(exce.ToString());}}
protected override void OnFormClosing(FormClosingEventArgs e)
{base.OnFormClosing(e);
//Close application for real if windows is shutting down
if (e.CloseReason == CloseReason.WindowsShutDown)
return;
//If user close the window minimize
if (e.CloseReason == CloseReason.UserClosing)
{e.Cancel = true;
this.Hide();}}
private void pichelpcmd_MouseHover(object sender, EventArgs e)
{FrmCommandList frmCommandList = new FrmCommandList();
frmCommandList.Show();}}
private void pichelpcmd_MouseLeave(object sender, EventArgs e)
{FrmCommandList frmCommandList = new FrmCommandList();
frmCommandList.Close();}}

```

Appendix C : Google Search Section Form and Codes

```

public partial class FrmGoogle : Form
{
    public FrmGoogle()
    {InitializeComponent();}
    public void searchGoogle()
    {string toSearch = txtbxGoogleSearch.Text;
    string searchTemplate = "www.google.com/search?q=";
    Process.Start(searchTemplate + toSearch);
    this.Hide();}}

```

Appendix D: Reminder Settings

```

public partial class FrmSettings : Form
{
    FrmMain frm;
    RegistryKey autostartRegKey;
    public FrmSettings(FrmMain frm)
    {InitializeComponent();
    autostartRegKey =
    Registry.CurrentUser.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run",
    true);

    this.frm = frm;}
    private void FrmSettings_Load(object sender, EventArgs e)
    { //Read settings and set the check boxes to right state
    checkAutostart.Checked = autostartRegKey.GetValue("ReminderApp") != null;
    if (frm.settings.GetBool("Bubble"))
        checkBubble.Checked = true;
    if (frm.settings.GetBool("Messagebox"))
        checkMBox.Checked = true;
    btnSaveSet.Enabled = false;}
    private void button1_Click_1(object sender, EventArgs e) //button SAVE SETTINGS
    { //Autostart check
    if (checkAutostart.Checked)
        autostartRegKey.SetValue("ReminderApp", Application.ExecutablePath);
    else

```

```

        autostartRegKey.DeleteValue("ReminderApp", false);
//Set the values in dictionary
try
{ if (checkBubble.Checked)
    frm.settings.SetBool("Bubble", true);
  else
    frm.settings.SetBool("Bubble", false);

  if (checkMBox.Checked)
    frm.settings.SetBool("Messagebox", true);
  else
    frm.settings.SetBool("Messagebox", false);
  frm.settings.Save();}
catch (Exception ex)
{ MessageBox.Show("Error: " + ex);}
this.Close();}
private void checkAutostart_CheckedChanged_1(object sender, EventArgs e)
{btnSaveSet.Enabled = true;}
private void btnCancelSet_Click(object sender, EventArgs e)
{this.Close();}
private void checkBubble_CheckedChanged(object sender, EventArgs e)
{ btnSaveSet.Enabled = true;}
private void checkMBox_CheckedChanged_1(object sender, EventArgs e)
{btnSaveSet.Enabled = true;}

```

Appendix E: Add/Edit Section Form and Codes

```

public partial class FrmAddReminder : Form
{
    SpeechSynthesizer MyNia = new SpeechSynthesizer();
    FrmMain frm;
    bool edit;
int index;
    string filename = "remind.txt";

    public FrmAddReminder(FrmMain main, bool edit, int index)
    {
        InitializeComponent();
        frm = main;
        this.edit = edit;
        this.index = index;}
    private void FrmAddReminder_Load(object sender, EventArgs e)
    {
        //Disblae the dropdown
        timePicker.ShowUpDown = true;
        //If not in edit mode set timepicker value to current time
        if (!edit)
        {
            timePicker.Value = DateTime.Now;}

        string[] daysArr = { "Monday", "Tuesday", "Wednesday",
                            "Thursday", "Friday", "Saturday",
                            "Sunday", "Workdays", "Weekend",
                            "Everyday" };
        for (int i = 0; i < daysArr.Length; i++)
        {
            string item = daysArr[i];
            dayBox.Items.Add(item);}
        comboBox1.Items.Clear();
        DirectoryInfo dinfo = new DirectoryInfo("./reminderSongs/");
        FileInfo[] files = dinfo.GetFiles("*.wav");
        comboBox1.Items.Add("None");
        comboBox1.SelectedIndex = 0;
    }
}

```



```

foreach (FileInfo file in files)
{
    comboBox1.Items.Add(file.Name);}}
private void button1_Click_1(object sender, EventArgs e)//button SAVE
{
    MyNia.SpeakAsync("Reminder added");
    //Get the short time of the timepicker
    DateTime dt = timePicker.Value;
    String timeNow = dt.ToShortTimeString();

    //If not edit mode just add the reminder to the end of remind.txt
    if (dayBox.Text != null)
        if (!edit)
        {
            if (!textBox1.Text.Contains(";"))
            {
                try
                {
                    using (StreamWriter sw = File.AppendText(filename))
                    {
                        string recurr;
                        string active;
                        string sound = "";

                        if (recurringBox.Checked)
                        {
                            recurr = "Yes";}
                        else
                        {
                            recurr = "No";}

                        if (activeBox.Checked)
                        {
                            active = "Yes";}
                        else
                        {
                            active = "No";}
                        if (comboBox1.SelectedItem.ToString() != "None")
                            sound = comboBox1.SelectedItem.ToString();
                        //Write the new reminder to the remind.txt
                        sw.WriteLine(dayBox.Text + ";" + timeNow + ";" +
textBox1.Text + ";" + recurr + ";" + active + ";" + sound);
                        sw.Close();}}
                    catch (Exception ex)
                    {
                        MessageBox.Show("Error: " + ex);}

                        //Reload the reminders in the main form and application
                        frm.AddReminders();
                        this.Close();}
                else
                {
                    MessageBox.Show("No ; sign is allowed in message");}
            else
            {
                //Re read all the reminders into memory
                List<String> remindList = new List<String>();
                try
                {using (StreamReader sr = new StreamReader(filename))
                    {string line;
                        while ((line = sr.ReadLine()) != null)
                        { remindList.Add(line);}}

```

```

        string recurr;
        string active;
        string sound = "";

        if (recurringBox.Checked)
        {recurr = "Yes";}
        else
        {recurr = "No";}

        if (activeBox.Checked)
        {active = "Yes";}
        else
        {active = "No";}

        if (comboBox1.SelectedItem.ToString() != "None")
            sound = comboBox1.SelectedItem.ToString();
        //Split and change the information for remind with index
        String[] remind = remindList[index].Split(';');
        remind[0] = dayBox.Text;
        remind[1] = timeNow;
        remind[2] = textBox1.Text;
        remind[3] = recurr;
        remind[4] = active;
        string reminder = remind[0] + ";" + remind[1] + ";" +
remind[2] + ";" + remind[3] + ";" + remind[4] + ";" + sound;
        remindList[index] = reminder;

        //Rewrite the remind.txt file with changes
        File.WriteAllLines(filename, remindList.ToArray());}
    catch (Exception ex)
    { MessageBox.Show("Error: " + ex);}
    frm.AddReminders();
    this.Close();}}
private void btnCancel_Click(object sender, EventArgs e)
{this.Close();}}

        if (comboBox1.SelectedItem.ToString() != "None")
            sound = comboBox1.SelectedItem.ToString();
        //Write the new reminder to the remind.txt
        sw.WriteLine(dayBox.Text + ";" + timeNow + ";" +
textBox1.Text + ";" + recurr + ";" + active + ";" + sound);
        sw.Close();}}
    catch (Exception ex)
    {
        MessageBox.Show("Error: " + ex);}

        //Reload the reminders in the main form and application
        frm.AddReminders();
        this.Close();}
    else
    {
        MessageBox.Show("No ; sign is allowed in message");}
else
{
    //Re read all the reminders into memory
    List<String> remindList = new List<String>();
    try
    {using (StreamReader sr = new StreamReader(filename))
        {string line;
            while ((line = sr.ReadLine()) != null)
            { remindList.Add(line);}}
        string recurr;

```

```

        string active;
        string sound = "";

        if (recurringBox.Checked)
        {recurr = "Yes";}
        else
        {recurr = "No";}

        if (activeBox.Checked)
        {active = "Yes";}
        else
        {active = "No";}

        if (comboBox1.SelectedItem.ToString() != "None")
            sound = comboBox1.SelectedItem.ToString();
        //Split and change the information for remind with index
        String[] remind = remindList[index].Split(';');
        remind[0] = dayBox.Text;
        remind[1] = timeNow;
        remind[2] = textBox1.Text;
        remind[3] = recurr;
        remind[4] = active;
        string reminder = remind[0] + ";" + remind[1] + ";" +
remind[2] + ";" + remind[3] + ";" + remind[4] + ";" + sound;
        remindList[index] = reminder;

        //Rewrite the remind.txt file with changes
        File.WriteAllLines(filename, remindList.ToArray());}
    catch (Exception ex)
    { MessageBox.Show("Error: " + ex);}
    frm.AddReminders();
    this.Close();}}
private void btnCancel_Click(object sender, EventArgs e)
{this.Close();}}

```

Appendix E : Testing and training user voice

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Speech.Recognition;
using System.Speech.Synthesis;
using System.IO;

namespace ReminderApp
{
    public partial class FrmTestandTraining : Form
    {
        SpeechRecognitionEngine listener = new SpeechRecognitionEngine(); //if set the
language by own used null;
        SpeechSynthesizer MyNia = new SpeechSynthesizer();
        PromptBuilder pBuilder = new PromptBuilder();
        public FrmTestandTraining()
        {InitializeComponent();
            try
            {
                listener.LoadGrammarAsync(new Grammar(new GrammarBuilder(new
Choices(File.ReadAllLines(@"TestandTrainingVocabulary.txt")))));

```

```

        listener.RequestRecognizerUpdate();
        listener.SpeechRecognized += new
EventHandler<SpeechRecognizedEventArgs>(testengine_SpeechRecognizedBM);
        // use the system's default microphone
        listener.SetInputToDefaultAudioDevice();
        // start listening
        listener.RecognizeAsync(RecognizeMode.Multiple);
        MyNia.SelectVoiceByHints(VoiceGender.Neutral);}
    catch (Exception)
    {
        MyNia.SpeakAsync("Something when wrong. Please try again");}}

    private void testengine_SpeechRecognizedBM(object sender,
SpeechRecognizedEventArgs e)
    {
        rtbSpeechTest.Text = rtbSpeechTest.Text + " \n " +
e.Result.Text.ToString();}

    private void FrmTestandTraining_Load(object sender, EventArgs e)
    {
        MyNia.Speak("Test your voice for better recognition you may speak in
Bahasa Melayu or English language");}}}

```

Appendix F: Voice Commands List

GrammarCommand.txt	GrammarCommand.txt
1 Hello Nia	37 buka Notepad
2 Salam	38 Notepad
3 who are you	39 open Access
4 what is name	40 open Powerpoint
5 nope	41 Ms Powerpoint
6 yes	42 buka Powerpoint
7 tidak	43 open Visual Studio
8 tiada	44 open Sticky Notes
9 takpelah	45 sticky Notes
10 Nia exit	46 search Google
11 tutup Nia	47 Google
12 Nia resume	48 Google please
13 Nia comeback	49 close chrome
14 shutdown my computer	50 exit chrome
15 restart my computer	51 open chrome
16 what time is it	52 testing
17 now what time	53 show
18 what time now	54 stop listening
19 sekarang pukul berapa	55 search
20 harini berapa hari bulan	56 search for
21 what date is today	57 find
22 date today	58 SEARCH
23 tarikh harini	59 all processes
24 what is my schedule	60 show processes
25 remind me	61 Show Commands list
26 ingatkan saya	62 Commands list
27 note for me	63 Show Commands
28 note me	64 bagithu commands
29 schedule please	65 Close commands
30 open Youtube	66 Tutup commands
31 open Music	67 test my voice
32 open Document	
33 open Word	
34 buka Word	
35 Microsoft Word	
36 open Notepad	
37 buka Notepad	
38 Notepad	

Appendix Figure 1 Voice Commands List