

This is the functional specification for my voice assistant project which was completed on the 22/11/2018.

Voice Assistant

Functional Specification

Alex Randles
15302766



Table of Contents

Table of Contents

1.Introduction	1
1.1 Overview	1
1.2 Business Context	4
1.3 Glossary	5
2.General Description	7
2.1 Product/ System Function	7
2.2 User Characteristics & Objectives	11
2.3 Operational Scenarios	12
2.4 Constraints	15
3.Functional Requirements	19
3.1 Interpret user's voice	19
3.2 Process user's voice	20
3.3 Verify command	22
3.4 Access Command Library	23
3.5 Search Command Library	24
3.6 Execute command	25
4. System Architecture	27
4.1 System Architecture Diagram	27
4.2 System Architecture Description	27
5. High Level Design	29
5.1 Data Flow Diagram	29
5.2 Context Diagram	30
5.3 News use case Diagram	31
5.4 State machine Diagram	32
5.5 Component Diagram	32
6. Preliminary Schedule	33
6.1 GANTT Chart	33
7. Appendices	34
7.1 Software Resources	34
7.2 Hardware Resources	34
7.3 Research Resources	34

Introduction

1.1 Overview

- **Description**

The system I'm intending on making is a voice assistant. This system will work through voice commands from the users and interact with the user through speech from a speaker within my system. The interaction between the user and system will be solely with voice. Voice commands will be issued to and from the user in order to carry out a certain function. The system software will be stored on a Raspberry Pi 3 Model B+ running the Noobs Operating system. The reason for using a raspberry pi to run the required software is the size and portability of the device. It also has all the necessary hardware requirements such as expandable storage, Wifi for connecting to the internet wireless, GPIO pins for connecting to the matrix creator board, Bluetooth for the speaker and 4 usb ports for any usb device that I may need to use/power. The matrix creator development board which I'm attaching to it, will be used to record the voice of the user and interact with the user.

Introduction

- **Need for my system**

1. People who are unable to use the internet could find a need for this system as it doesn't require any knowledge of how the system works but simple of which commands are available.
2. Elderly people who have poor mobility could find a need for it. For example, instead of looking around for a newspaper they could get the news from my system.
3. People who are unable to read could also find a need for this system as they would be unable to read the news/weather etc but could still give voice commands to my system.
4. People with poor dexterity could find a need for this system as it would be a lot easier to use this system over a computer. For example, it may take them a long time to google search their song rather due to their poor ability using a keyboard instead they could ask my system to play their song with a voice command.

Introduction

- **Functions**

The functionality of my system is they interact with the system through issuing voice commands and the system interacts with them by artificial speech. I'm hoping to have voice commands that carry out the following functions at a minimum. I hope to add more functions to the project throughout its life cycle.

- Play music through Spotify/Youtube
- Access the news
- Remote control TV/TV box
- Find out the weather forecast
- Tell the time

- **Interaction with other systems**

The system will interact with several different online systems, this interaction will mainly happen through API's which will retrieve information from there system and return it in a JSON format that will be then be used by my system.

Introduction

1.2 Business Context

- This system could be possible sold to a retailer as a life assistant for the elderly or people with disabilities that may make it difficult for them to use the internet.
- It could be sold online as a device to be used at home or offices as a general assistant for daily needs.
- A nursing home/disability chain could possible roll them out across their chain to be used in each room a person stays in, this could give them a lot more freedom and interaction with the outside world compared what they may have currently have.
- It could also be sold to primary schools/secondary schools as an assistant for students who have learning or interaction disabilities.
- There is a huge number of different contexts the system could be used in, it really depends on what the users end goal is, this is why I think it has a huge business prospect.

Introduction

1.3 Glossary

- **Raspberry Pi**

The Raspberry Pi is a low cost, credit-card sized computer.

- **Matrix creator board**

MATRIX Creator is a fully-featured development board and daughter board for the Raspberry Pi that includes led Array, sensors, microphone array, wireless communications, and a field-programmable gate array.

- **GPIO**

A General purpose input/output pins on the Raspberry Pi, the function of a GPIO pin is customizable and can be controlled by software.

- **NOOBS**

New Out of the Box Software. It's an easy operating system installation manager for the Raspberry Pi.

- **Raspbian**

Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware.

Introduction

- **IR**

Infrared uses light to carry signals. Mainly used in TV remote controls.

General Description

2.1 Product/System Functions

Figure 1.1 User interaction with system

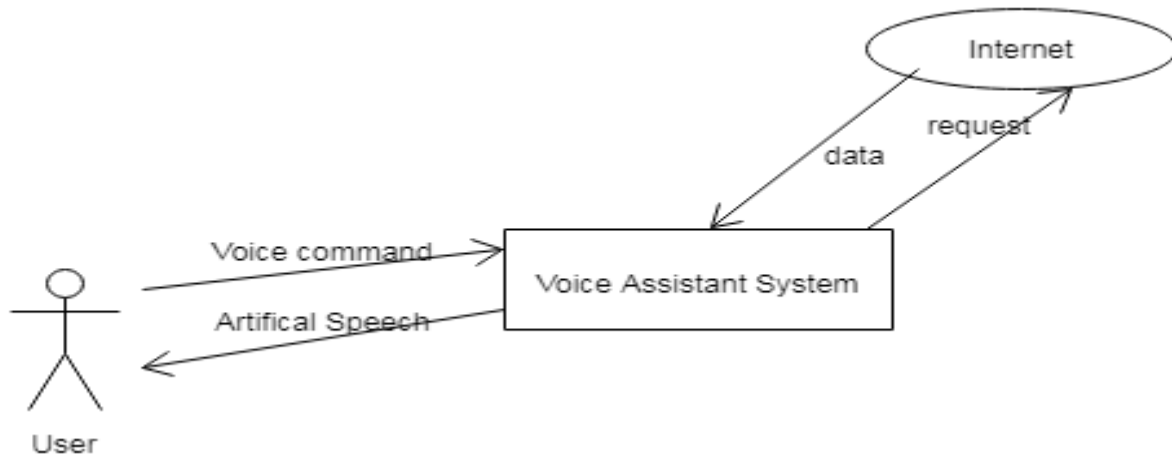
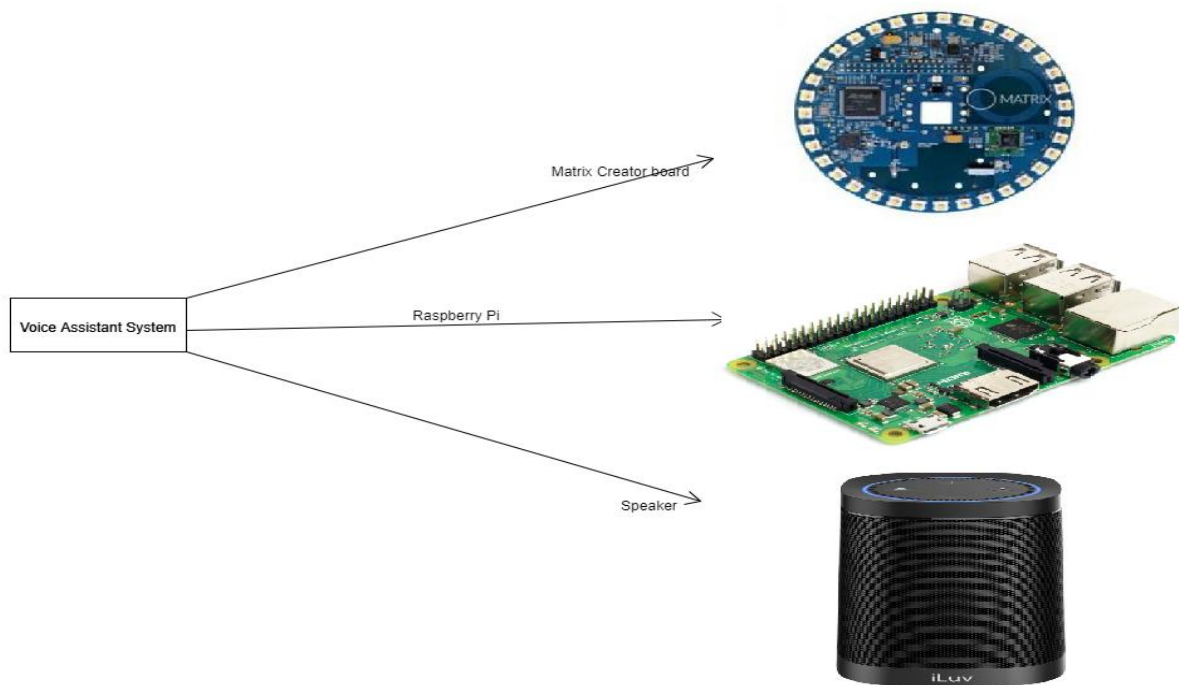


Figure 1.2 Voice Assistant System



General Description

Major components of the project:

1. **Raspberry Pi 3 Model B+** used to store software, interact with matrix creator board and access the internet.
2. **Matrix Creator Board** used mainly for recording the voice commands but also has additional functionality I may use such as LEDs and wireless communications.
3. **Speaker** will be used to play audio and speak to the end user.

Baseline Functions:

- Play music through Spotify/Youtube - using the built-in speaker.
- Access the news - read out the news to the user, mostly the headline news.
- Remote control TV/TV box – will have the ability to control a device TV/TV box that uses an IR blaster remote to control it using the Matrix Creators IR blaster.
- Find out the weather forecast – brief description of the current weather and forecast.
- Tell the time

General Description

The general functionality of my system is broken into the following steps:

- The users voice will be detected by the microphone array on the matrix creator board.
- A hotword will be set in my system that makes the system listen for a command e.g 'Alexa'
- Using a hotword speeds up efficiency and would be too complex to decipher sentences for commands.
- Now the system is listening for a command, the system records what the user says in the next few seconds. I've yet to decide on the length of this Recording. This will be decided once the I've completed all of the voice commands for the system and I will do testing onto what the maximum time it would take a user to say the longest command.
- Their recording will then be feed into a speech to text converter stored on the Raspberry pi.
- The text will then be checked against a library of commands I've created to see if it matches any of them. If not, the system will tell the user, it has not said a valid command.

General Description

- If a valid command is heard a script will start running from this library I've created, the type of script will depend on the command.
- Commands that require speaking to the user through artificial speech will use a text to speech converter.
- Commands that require constantly updating data such as the news/weather will use an API to grab this data from the internet.
- The artificial speech will then be generated and spoken to the user.

General Description

2.2 User characteristics and Objectives

The users of my system will generally be people with little expertise in using software systems. They require no knowledge of how the system works but only knowledge of what voice commands are available to them. This could be provided in a paper print out or a settings command. When using the internet, the user requires knowledge of how computers and search engines work and this can be a difficult task for some people. Elderly people would fit mainly into this category. This could be seen as an independent home care assistant.

The user may have many objectives when using this system and this can vary greatly from user to user. This system will be an assistant for anyone that is not very tech savvy. They require no technical knowledge only the ability to hear and speak. The user's goals may vary greatly when using the system, the user may simple want to tell the time as there unable to see the clock on the wall. The user could be unable to use Spotify to play a song so they just ask my system to play it for them. The objectives of the user are only limited by the number of commands available in the command library

General Description

Another user which may find my system useful is a person who could be tech savvy but could also have a disability that affects their ability to use a computer. These disabilities could include sight, dexterity and mobility. This system could greatly benefit them as they only require their hearing and speech to use my system.

Their objectives when using my system could also vary greatly depending on the task they wish to complete.

2.3 Operational Scenarios

Actors – User

Red – Systems words

Blue – Users words

Scenario 1: Play music through Spotify/Youtube

- User says hotword – **“Snowboy”**
- System will respond with an acknowledgement - **‘I’m now listening’**
- User will say command - **“Play music”**
- System will prompt user for the song title – **“What song would you like me to play?”**

General Description

- User tells system the song they'd like played – **"Summer by Calvin Harris"**
- System searches for song and if found plays it.
- Otherwise system tells the user the song has not been found – **"Song not found"**
- If the user would like to stop the song playing before it's over they say the hotword – **"Snowboy"**, followed by the command **"Stop playing music"**.
- Otherwise the song will stop playing automatically once the song is over.

Scenario 2: Access the news

- User says hotword – **"Snowboy"**
- System will respond with an acknowledgement – **"I'm now listening"**
- User responds with a sentence variation that contains the word news – **"Tell me the news"** or **"What is the news"** etc.
- System accesses the news and reads it out – **"The news for today is [news] "**

Scenario 3: Remote Control TV/TV Box

- User says hotword – **"Snowboy"**

General Description

- System will respond with an acknowledgement – **“I’m now listening”**
- User responds with the command they would like to be executed which will be some sentence variation consisting of turn, TV/TV box and off – **“Turn off the [TV/TV Box]”**.
- System uses the Matrix Creator to turn off the TV and respond with an acknowledgement – **“Turning off the TV/TV Box”**
- The TV/TV Box will now be turned off.

Scenario 4: Access the weather forecast

- User says hotword – **“Snowboy”**
- System will respond with an acknowledgement – **“I’m now listening”**
- User responds with a sentence variation that consists of the word weather – **“What is the weather?”** or **“Tell me the weather”**
- System will then retrieve the weather through a weather API and read it out to the user – **“The weather is...[weather]”**

General Description

Scenario 5: Tell the time

- User says hotword – **“Snowboy”**
- System will respond with an acknowledgement – **“I’m now listening”**
- User responds with a sentence variation that consist of the word time – **“Tell me the time” or “What is the time”**
- System retrieves the time and reads out the time to the user – **“The time is...[time]”**

2.4 Constraints

Speed Constraints

- There is a huge constraint of speed when it comes to my project as the user shouldn’t have to wait a long period of time for the system to respond. The speed at which the system responds to a command should be less than the time it would take that user to carry out the command manually. For example, the news function should be able to read the top 10 news headlines in under 1 minute 10 seconds as this was the time it took me to unlock my iPhone, search RTE news and read the top 10 headlines. The optimal response time should be at about half this.

General Description

Software constraints

There is a number of software constraints when it comes to this project. The main reason there is constraints on the software I can use is I'm using the Raspberry Pi to store my software. When setting up the Raspberry Pi I used a Pre-Installed SD card with NOOBS to install the Raspbian operating system. This is the official Raspberry Pi operating system and it comes with Java, Python and plenty of packages pre-installed on it. I couldn't use an operating system such as Windows as the current version which is available for the Raspberry Pi is unstable, unfinished and has very poor performance.

Size Constraint

There is a constraint when it comes to size for my system as I'd like to be as small as possible as well as being efficient. This system should be portable and not take up a lot of space. It could be used on office desk which mostly don't have a huge amount of space.

General Description

Hardware Constraints

- There are a number of hardware constraints when it comes to my project.
 - The hardware platform must be capable of running Python, Java/JavaScript and Unix commands. These are the main languages I'll be using for my system as they are most suitable for my project and I'm most familiar with them.
 - The hardware platform must also be able to communicate with the Matrix Creator development board I'm using. The Matrix creator communicates through the GPIO on the it. The Raspberry Pi is perfect for this as it has a GPIO header.
 - The hardware must also have some sort of Audio output available such as Bluetooth or Aux preferably. It will also need to have at least one USB port to power the speaker.
 - The hardware platform must also have enough memory to store all the required software I need for my system.
 - WIFI capability is also necessary.

General Description

User Requirement

- The user must have a basic understanding of the systems commands and the be able to speak and hear. Without this they would be unable to use my system correctly.

Time Constraint

- DCU has given a final delivery date of the 19th of May 2019. This a strict deadline and I have all my project materials must be done by this date.

Industry Protocols

- The users for this system will vary greatly and there is not one definite set of users. The system must not use any speech that may be found offensive and must have the correct copyright privileges to play music if it were to be sold.

Functional Requirements

3.1 Interpret user's voice

Description

The system must be able to understand what the user is saying to it. This means I must use a proven and tested way of Speech Recognition. A lot of research must be done into finding out which method is most effective for this. There is a huge variation between each person's speech so this must be done in the most effective way.

Criticality

This is one of the most critical aspects of the system as without the system being able to understand what that specific user is saying they will be unable to use the system as it works based on speech interaction.

Technical Issues

The systems design will only allow a person speaking English to use it. This is because most of the high-end speech recognition packages don't support other languages and I only understand English. The rate of speech could also be an issue when implementing this requirement as some users may speak too slow and the system could time out waiting on a response.

Functional Requirements

Dependences with other requirements

The main dependencies with fulfilling this requirement are the microphone recording quality and the speech recognition reliability. If the microphone quality is not good or cannot hear the speech correctly, the speech recognition software will be unable to process the speech of user thus making it impossible to understand what they are saying.

3.2 Process user's speech

Description

The system must be capable of processing the user's speech into a text form which can then be used to activate a specific command. The command which is activated will depend on what the user says.

Criticality

This is a quite important part of the system as this is used to activate the command they wish to activate. If this is not done correctly an incorrect command or no command found could be activated.

Functional Requirements

Technical Issues

One issue when satisfying this requirement is the speech recognition software being used. I will carry out a lot of testing on the speech recognition software to check which format it translates the text into. Upper case, lower case and spaces will have to be dealt with correctly within my design.

Dependencies with other requirements

This requirement will rely on the speech recognition understanding what the user is saying. If the speech recognition doesn't properly recognise what the user is saying this requirement will also not be fulfilled as the system will process it into incorrect text and not activate the correct command. The speed at which its processed is also important for this requirement as this will affect the overall speed and performance of the system.

Functional Requirements

3.3 Verify Command

Description

Once the user's speech is translated into a text format, this must be checked against a library of commands to find if the text relates to a command and what command it relates to. This will be done through string manipulation and must be done to a high standard.

Criticality

This is a very important part of the system as the systems reliability will be effected by mismatched commands. The system should be able to match the user's speech to the correct command nearly all of the time. The whole system relies on this as incorrect code will be run if this requirement is fulfilled to a high standard.

Technical Issues

A huge implementation issue when fulfilling this requirement is, the different word combinations a user may say to activate a specific command. I will have to design the system in such a way that I consider these different combinations. For example, a user may say 'Tell me the news' vs 'What is the news?'.

Functional Requirements

These variations will have to be dealt with when designing my system to ensure each user is able to use the different functions of the system.

Dependencies with other requirements

This heavily relies on the user's speech being translated correctly because of this part is not done correctly, the system will be unable match their speech to the correct command and this could lead to the incorrect command being activated or a command not found error being activated.

3.4 Access command library

Description

Once the user's speech has been matched to a certain command the system must access the command library which will contain the scripts for all the commands the system contains.

Criticality

This is quite important as the system must be able to find all the data related to the specific command. This will be needed for when the command is run.

Functional Requirements

Technical Issues

The design of the command library must ensure correct structure so that easy and quick access time is accomplished by the system. The layout of the library must ensure that each command has the same structure.

Dependencies with other requirements

The speed at which the library is accessed will rely on the performance speed of the hardware I'm using. Correct directory structure will ensure that the correct command is accessed each time.

3.5 Search Command Library

Description

After the system has accessed the command library it must then find the correct code for the specific command the user wishes to run.

Criticality

The design of the system must be one that allows each searching through the library and the file names must follow the same standard.

Functional Requirements

Dependencies with other requirements

This requirement heavily depends on the system accessing the command library in the first place. If the system doesn't access this library, the system will be unable to search through it.

3.6 Execute command

Description

Once the system has successfully searched through the command library and found the code for that specific command, it will have to execute all the code related to that command. This must be done in the correct order to ensure the user receives the response they were expecting.

Criticality

If the code related to that command is ran incorrectly the user will either receive no response or an incorrect response. This will make the user perceive the system as unreliable and could make them stop using the system all together.

Functional Requirements

Technical Issues

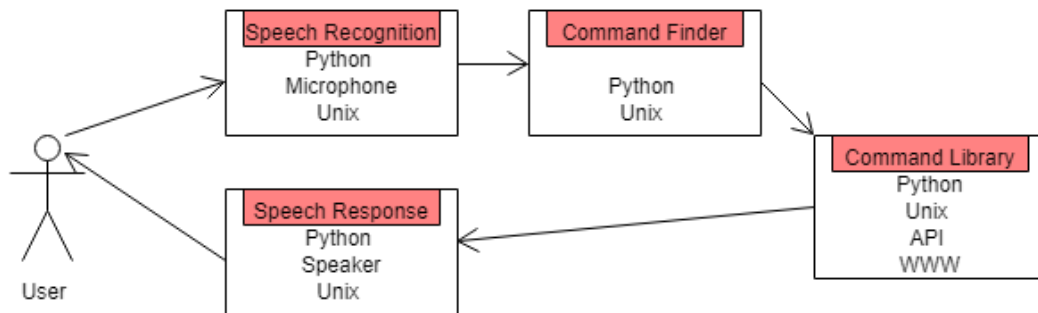
The response the user receives should be what they were expecting. The design of the system should allow for the user to receive the information they require and in an easy to understand format. The rate at which the system interacts with the user should be such that every user is able to understand what it is saying. The design of the system should ensure that the speaker quality is adequate enough for each user to hear.

Dependencies with other requirements

This requirement relies on the execution speed of the system being quick enough to give the user the feedback requirement in a timely fashion. The speaker quality is also important as the user may have poor hearing ability.

System architecture

4.1 System Architecture Diagram



4.2 System Architecture Description

Speech Recognition

- This will include all the hardware and software that recognises the speech and translates into a string format. This is the module which the user interacts with.

Command Finder

- This will be feed the string created by the speech recognition module and carry out string manipulation to find the best matched command.

System architecture

Command Library

- This will contain the code for all the commands available to the user. This will execute the code for the specific command or generate a command not found error.

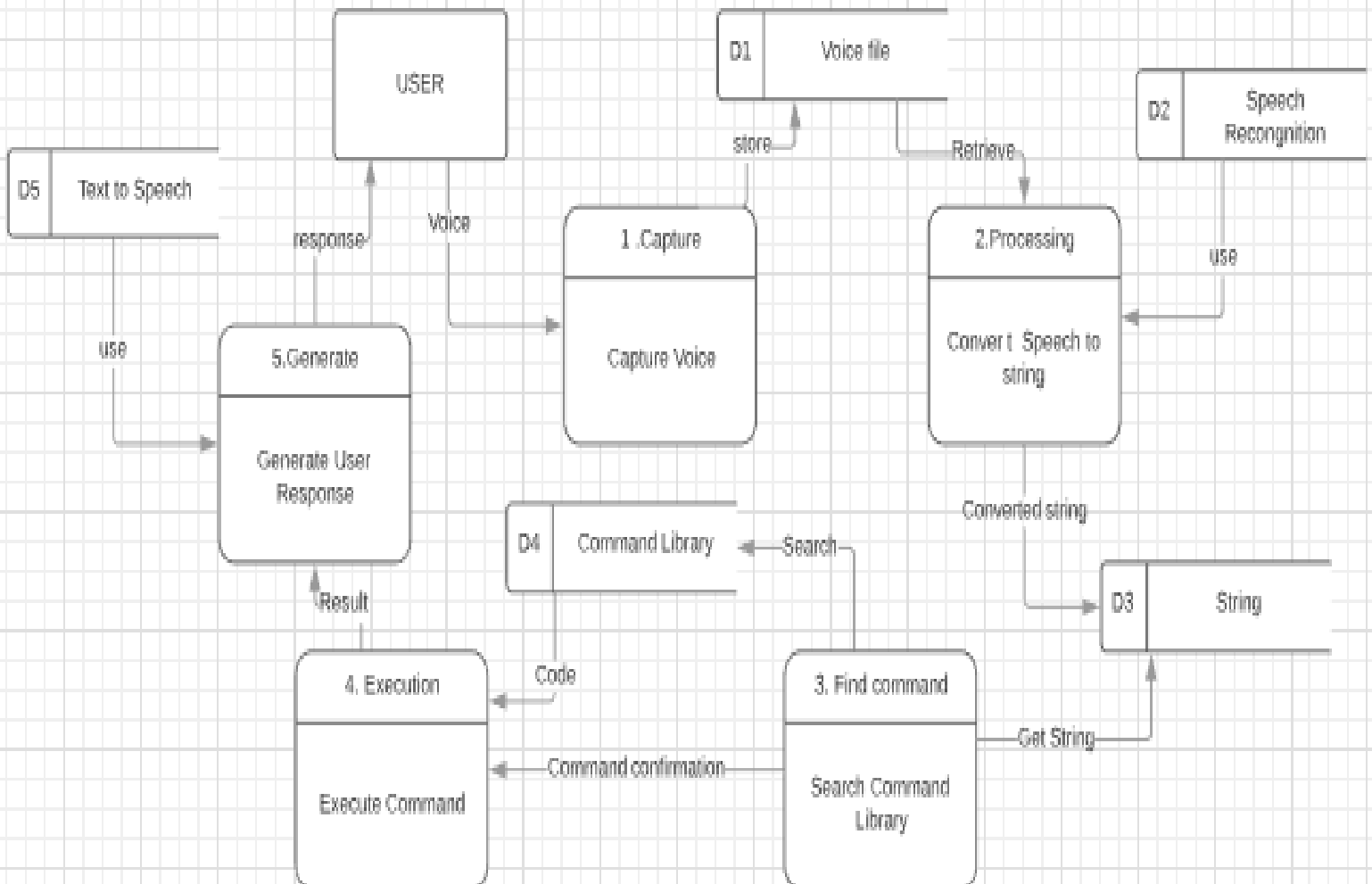
Speech Response

- This will receive the information generated from the command library and return it to the user in a spoken format which is easy for them to understand.

High Level Design

5.1 Data Flow Diagram

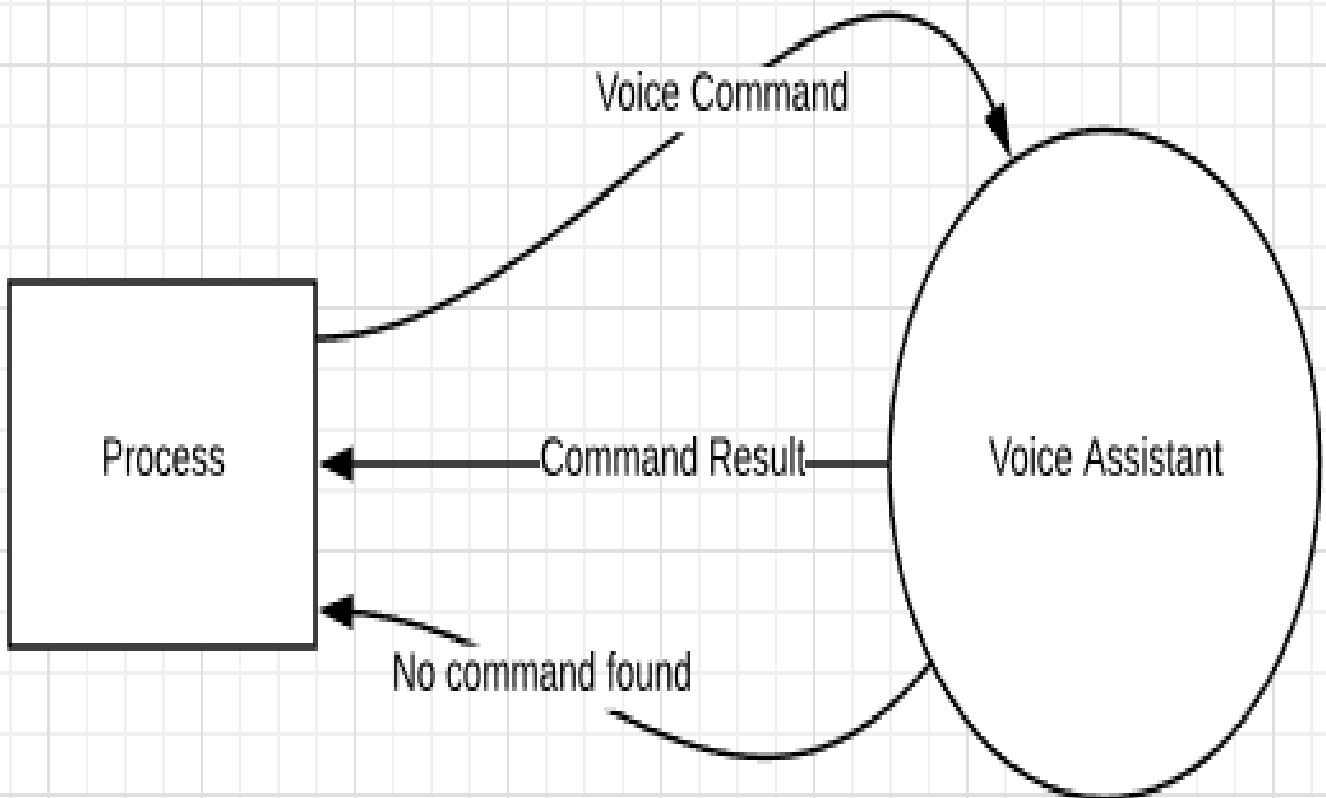
DATA FLOW DIAGRAM



High Level Design

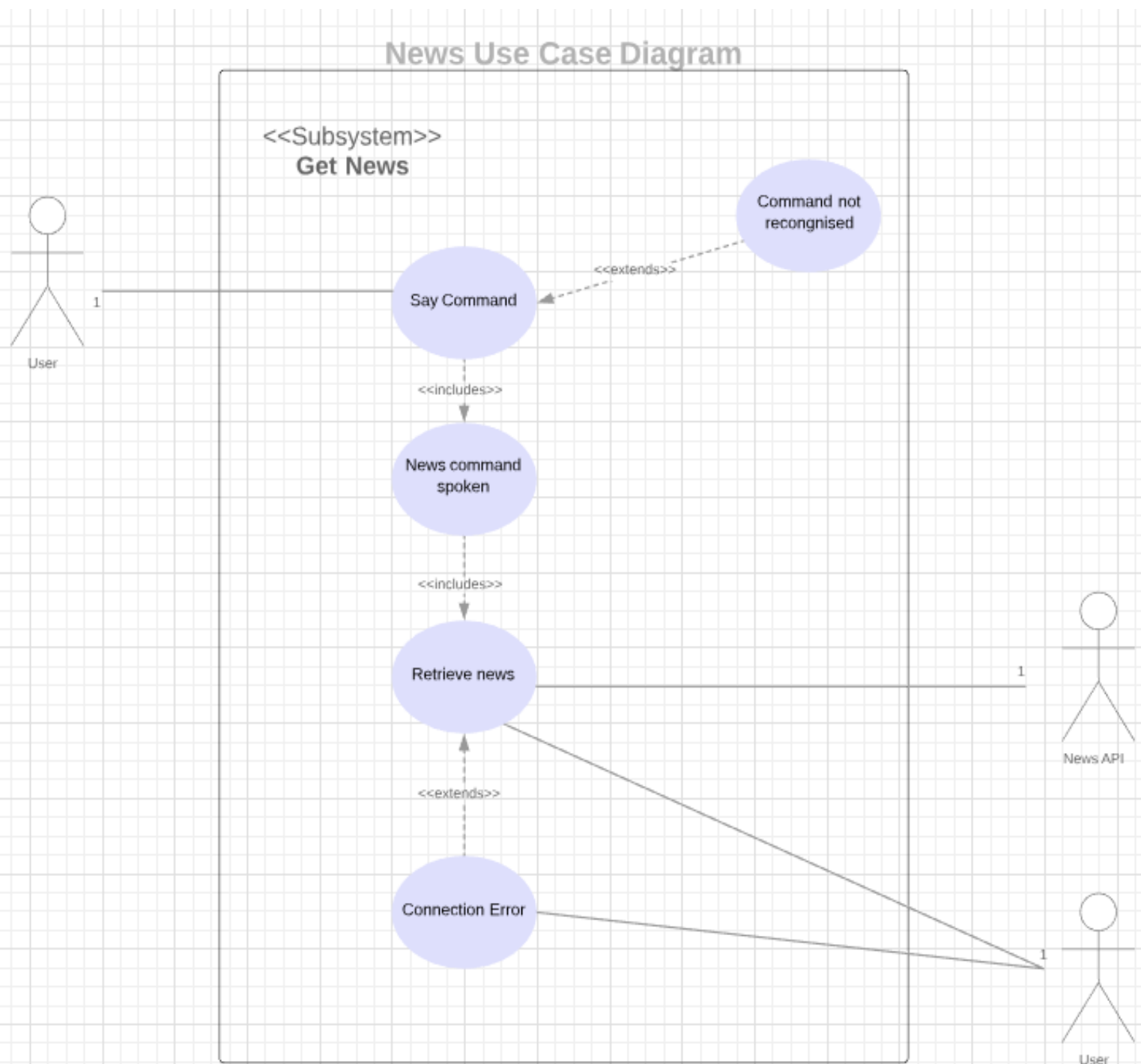
5.2 Context Diagram

CONTEXT DIAGRAM



High Level Design

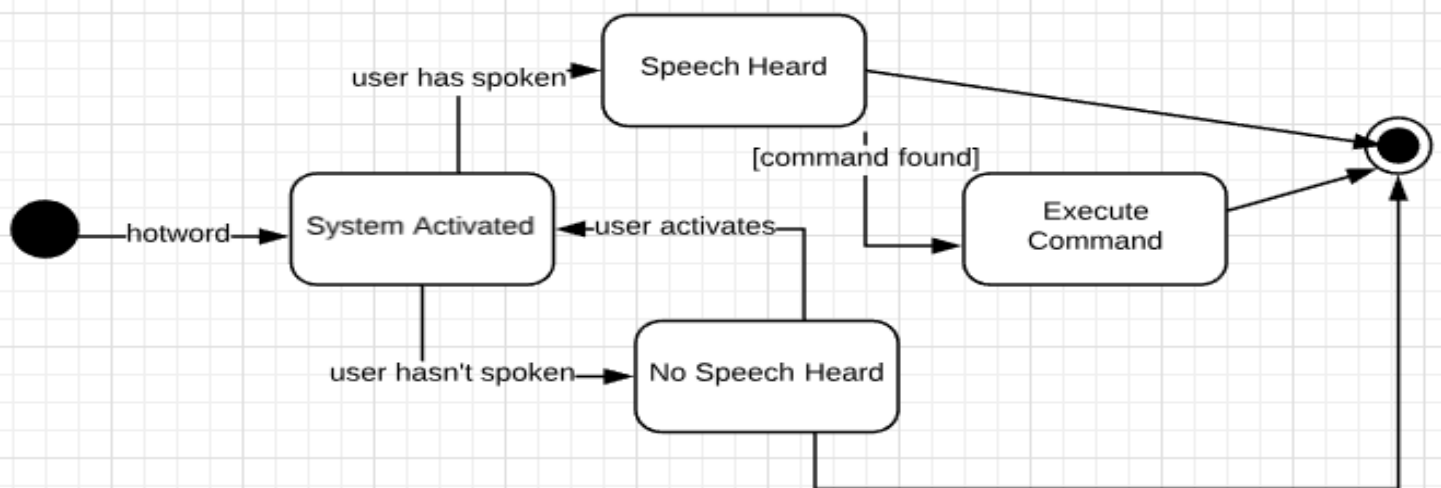
5.3 News use Case Diagram



High Level Design

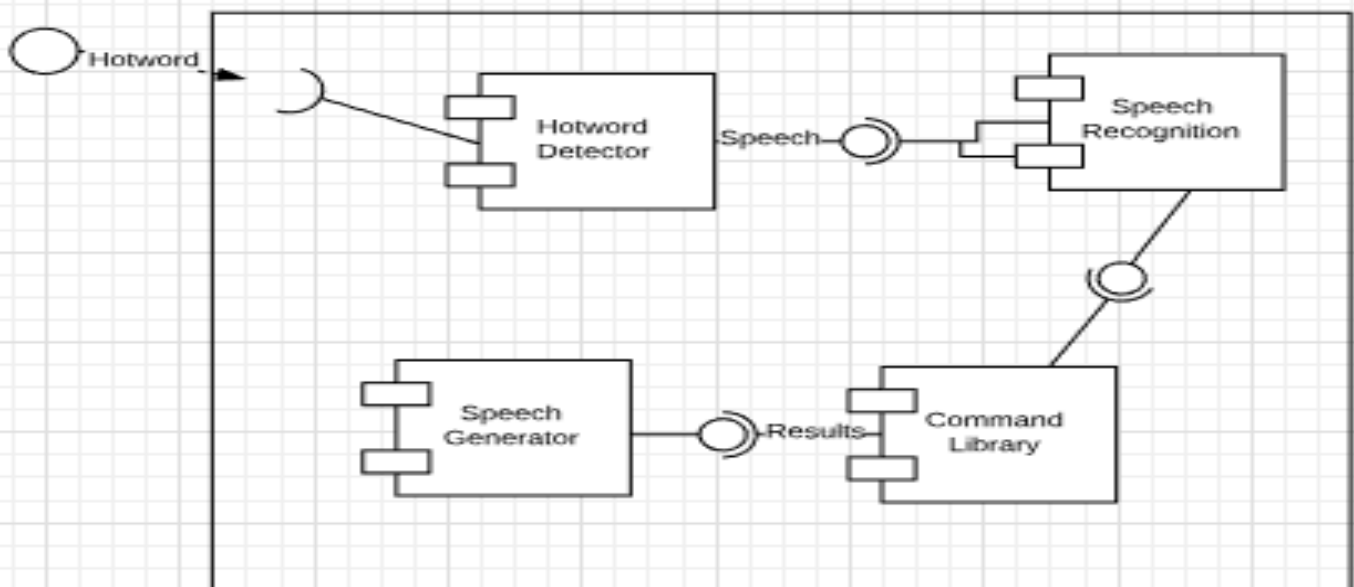
5.4 State Machine Diagram

State Machine Diagram



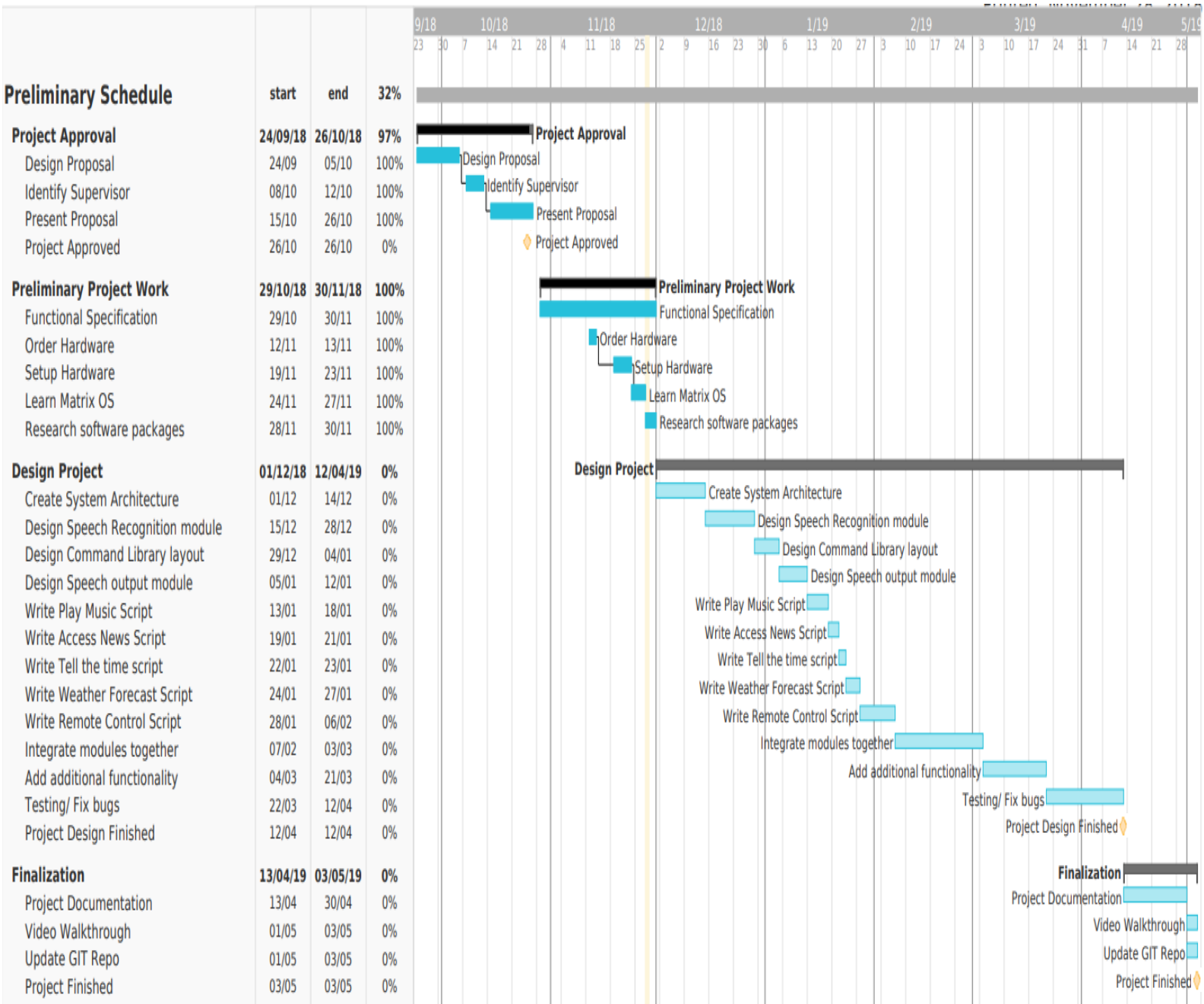
5.5 Component Diagram

Component Diagram



Preliminary Schedule

6.1 GANTT Chart



Key



Software: Most of the software will be free and easy to install Python and UNIX packages. Only specialised software needed is for the Matrix Creator operating system, this is available through their website.

Hardware: Raspberry Pi 3 Model B+, Micro SD card, Speaker & Matrix Creator development board.

Appendices

7.1 Software Resources

Hotword detector

- <https://snowboy.kitt.ai/>
- <https://pypi.org/project/snowboy/>

Speech Recognition/Response

- <https://pypi.org/project/gTTS/>
- <https://pypi.org/project/SpeechRecognition/>
- <https://pythonprogramminglanguage.com/text-to-speech/>

Other

- <https://newsapi.org/>
- <https://openweathermap.org/api>
- <https://developers.google.com/youtube/>
- <https://developer.spotify.com/documentation/web-api/>

7.2 Hardware Resources

- <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>
- <https://www.matrix.one/products/creator>



**Matrix Creator connected to
Raspberry Pi**

7.3 Research Resources

- <https://google.com>
- <https://wikipedia.com>
- <http://asr.cs.cmu.edu/spring2011/>
- <https://maker.pro/raspberry-pi/tutorial/the-best-voice-recognition-software-for-raspberry-pi>
- <https://www.techradar.com/how-to/how-to-control-the-raspberry-pi-with-your-voice>