



INSTITUTE *of*
TECHNOLOGY

CARLOW

Visual Developer tool for Mycroft: Research Manual

by

Michal Wojnowski

C00213963

Software Development

Year 4

Supervisor: Joseph Kehoe

Date of Submission: 20/04/2020

Abstract

The aim of this project is to build a visual development tool in the vain of Scratch which would allow the users to easily build skills(apps) for Mycroft which is an open source voice assistant for Linux. This application should allow the users to create a skill without having to write any code by just dropping visual blocks on a canvas with prewritten code in the. The only thing that a user will have to do is answer a short questionnaire to create a skill template and select the file

Table of Contents

Abstract	1
1. Introduction	2
2. Mycroft	3
2.1 Architecture	5
2.2 Skills	6
3. Similar Apps	7
3.1 Alexa	7
3.2 Cortana	9
4. Scratch	10
5. Technologies Used	11
5.1 Framework	11
5.2 Database	13
6. Conclusion	14
7. Declaration	15
8. References	16

1. Introduction

This is a research document for the project of creating a visual development tool for Mycroft.

The project's aim is to develop an easy to use tool in the vein of Scratch which will allow the user to effortlessly build skills* for the platform without the need of having any prior programming knowledge of experience.

The document will explain what *is* Mycroft, an overview of similar applications on the market, other visual development tools that are currently on the market, the technology that are expected to be used in the project and finally an overview of what are the suitable libraries for this project.

*Skills is the name given by the developers of Mycroft and Amazon Alexa to the apps for their respective platform which allows the assistant to answer/perform new tasks. The reason that that the developers gave for the name *skill* is that the voice assistant doesn't just get a new app, it gains a skill which allows it to perform that task

2. Mycroft

Mycroft is a Python based free open-source (under the GPLv3 licence) voice assistant for Linux, Android and its own home device Mycroft Mark I and II. To understand what Mycroft is and how it came about, first, we need to know what a voice assistant is.

A voice assistant is an AI that uses voice recognition, natural voice procession and speech synthesis to allow users to use voice commands to their device in order to perform such functions as create events on a calendar, add an item to a shopping list, search for an answer to a given question, etc. How did Mycroft come about?

Mycroft was created by Ryan Sipes and Joshua Montgomery who's inspiration for it came from a Kansas City marketspace in which they saw a basic VI assistant.. In August 2015, the team launched a Kickstarter for creating Mycroft with the goal set at \$99,000 and to differentiate the product from other voice assistants like Alexa they promised privacy as a pivotal point of their sales pitch. To achieve their privacy claim Montgomery promised that Mycroft will protect user privacy through its open source machine learning platform. In the end they managed to raise \$127,520 and the physical product was shipped early May 2016.



Fig 1. Mycroft Mark 1

In January 2018 the Mycroft team launched a 2nd Kickstarter with the goal of \$50,000 in order to fund the creation of Mycroft Mark 2. The Mark 2 has several improvements over the Mark 1 and unlike its predecessor, which was targeted specifically at developers, the Mark 2 is targeted at the general audience so it was equipped with a screen which allows it to replay both audio as well as visual information as well as better microphones and speakers. The Kickstarter smashed its goal and ended up with \$394,572.



Fig 2. Mycroft Mark 2

2.1 Architecture

The Mycroft architecture is divided into four separate blocks, service, skills, voice and enclosure, they represent a process that's in memory of a linux system .

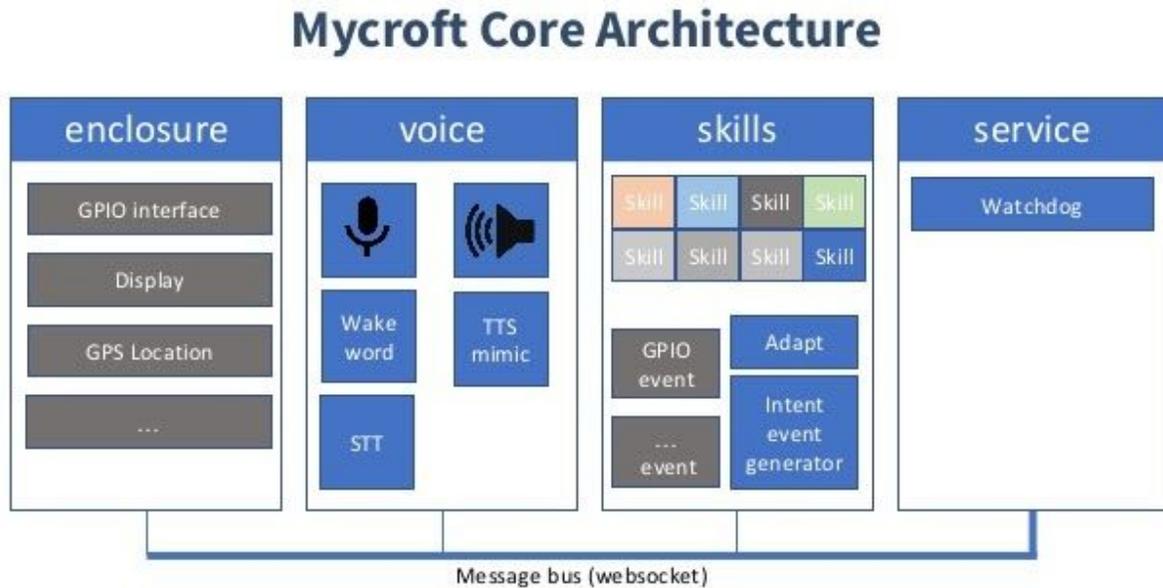


Fig 3. Mycroft Core Architecture Diagram

Function wise, the service block is the lowest level of the four blocks as it is responsible for establishing the message bus which allows all of the other blocks to connect and communicate with one another once the actual bus has been found.

The voice block is connected to the microphone and speakers of the system and is always looking for the Wake World key word, which in this case is “Mycroft”. Once it has been said, the device records the next part of the sentence and brings it into the system where it's sent to a cloud speech to text system which transcribes the answer from text to an audio file and sends it back to the voice service which then plays the audio file on its speakers. Once it's reached the voice service it is sent back through the bus as an utterance to the intent service in the skill block. This triggers the adapt query which searches if any of the skills on this device have an intent that matches the input utterance which invokes the skill. After the skill has been invoked it sends a message back to the voice block where the text to speech system changes it to audio which is then played.

The skill block is comprised of two systems, the first being responsible for loading all of the user made skills on startup. It does this by searching the file system for the opt Mycroft skills folder and when it finds the folder, it searches through it loading every skill inside. Meanwhile the second system is called the intent service which is connected to the instance of adapt. Every skill that launches registers its list of intents in the instance of adapt, with the purpose of allowing the system to check if the utterance spoken by the user matches with an intent of one of the custom skills.

Finally the enclosure block is responsible for talking to the hardware or software the program is running in right now, an example of that is the Mycroft Mark 1 model or the Linux environment Plasmoid.

2.2 Skills

Mycroft skills have three distinct parts that are utterance, dialogue and intent.

An utterance is a phrase that a user has to say after the Wake World has been spoken to invoke a skill. An example of an utterance is “ What’s the time in New York?”. Each skill will have its own set of utterances which allows the user to say the same thing, and invoke the skill in a myriad of different ways and not just the one, set in stone by the developer. There is nothing stopping two skills having the same utterance however there is a prioritization system in order to activate the most relevant skill to the users wish. A simple command that gives specific directions like “ play Michael Jackson on Pandora “ will simply go to Pandora and play the music. However, if the command is more complex the system receives a confidence score from each of the skills on how confident that skill is that it can provide the user with what the user asked for. An example would be if the user said “ play workout playlist “ and the user had a pre-made workout playlist made then the system would prioritize it over random Spotify playlists with that name.

The dialogue is the sentence or phrase that will be spoken by Mycroft as a response to the question after it executes its task, whether successfully or not which would impact the given response. Each skill will have its own set of dialogue. If we continue with the time example, the dialogue for it would be “ the.current.time.is.dialog”.

Finally the intent. Mycroft’s system matches the utterances spoken by the user with the skills registered in its instance of adapt, so when a user says “ Hey Mycroft, what’s the time in New York “ the system will search for a registered skill with a fitting intent. Once it fits it, the time

intent will be recognised and will match it to the time skill. If more than one skill has the same utterance then the prioritization process takes place in this step.

Alongside what was mentioned in the paragraph above, when creating a skill the developer also needs to keep in mind the name of skill, so it can be found by other users on the Mycroft Skill Marketplace. Furthermore it needs a response dialogue so the skill has a response of the user, a short 40 character description of the skill, a long description with an unlimited amount of characters, the authors name or github, what category the skill belongs to be it educational, quality of life or other. Note that if no category is specified then the skill will be set to the default category. And lastly, the skill needs tags which would make it easier for other users to find the skill in the marketplace.

In recent updates, the skill process for Mycroft has been made significantly easier. The user types in “ mycroft-msk create“ into their command prompt which will activate a short questionnaire that asks the user about the name of the skill, utterances, response dialog, short description, long description, author, categories and finally tags. The process creates all necessary files for a skill leaving the user to add the wanted functionality into the `__init__.py` file.

3. Similar Apps

The research started by looking at similar apps currently on the market. For Mycroft, the main competitors come from Amazon’s Echo (Alexa), followed by Apple’s Siri, Google’s Assistant and Microsoft’s Cortana. However as many of the applications are similar to each other only Alexa and Cortana will be covered as much of the information is overlapping for all of the products.

When it comes to similar apps to the planned visual development the closest tool available is Scratch. The tool included in the Lego Mindstorms series also draws resemblance to the planned tool however it’s design is focused on pictures rather than text which would be unfit for this project as each skill requires several variables which would limit the skill’s scope too much if they were predetermined.

3.1 Alexa

Alexa released in November of 2014 as the Amazon Echo to positive reviews and is currently one of, if not the, biggest voice assistant on the market. Over the years it received several new

variants with new limited edition special models as well as just standard models with upgraded hardware. Alexa is a complex device and so in order to cover it properly the explanations will be broken down into three sections, how does Alexa work and how do the skills for Alexa work.



Fig 4. Amazon Echo First Generation

How does Alexa work? Amazon hosts its own cloud service called Amazon Voice Service which will be referred to as the AVS from this point on. When the word “Alexa” is spoken the device’s Wake World Engine activates and the rest of the utterance following “Alexa...” is recorded and sent over to the AVS as an event. This event is handled by several layers of the AVS to determine what the user said and to perform the task. The first layer is called Automatic Speech Recognition (ASR), which determines what the user said. After that the event moves over to the second layer called Natural Language Understanding (NLU) so if for example the user said “Alexa, play music “ the AVS determines that “play music” is mapped to the music domain and if the user specifies the song that metadata would be added to the information, otherwise the song set as default would be played. Once the event has been mapped the service looks for a skill to perform the specified task. The final layer is Text to Speech (TTS), here for example the play directive and text to speech are sent back from the AVS to the device. During this process the text is also transformed into an MP3 which would be played as Alexa’s response and prelude to the song, thus completing the assigned task.

How do the skills work? The skills for Alexa have a similar structure to Mycroft. They are made up of 4 distinct parts, the invocation name, intent scheme, sample utterance and finally slot type definitions. The first and most basic component is the invocation name for example. This determines what someone has to say to invoke the skill. The second part of any skill is the intent

scheme, here it's determined what the user wants the skill to do.

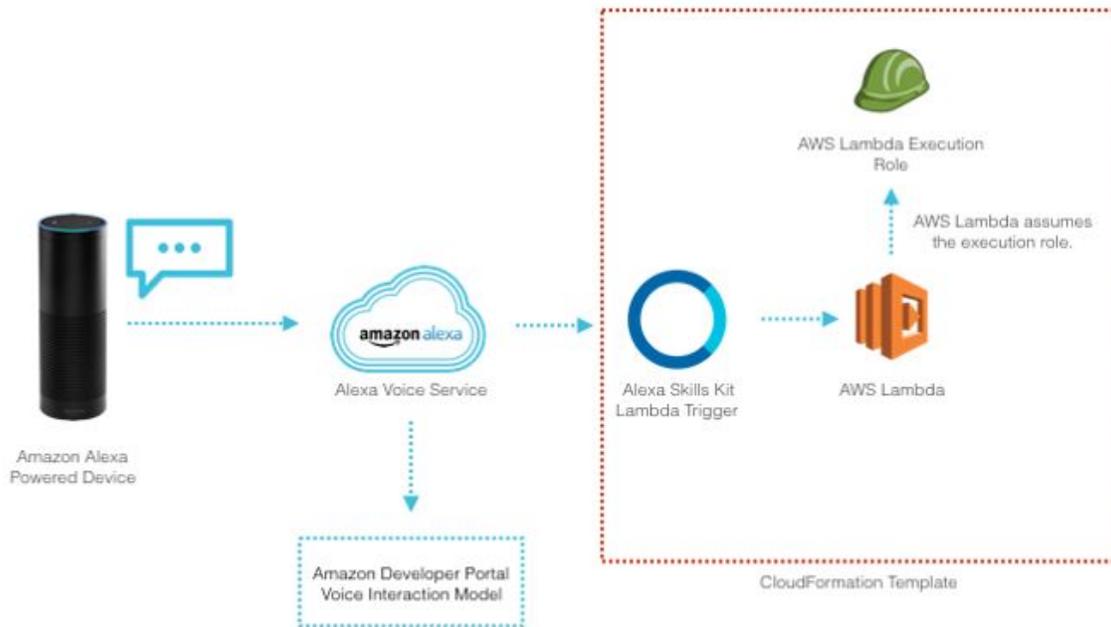


Fig 5 Alexa skill diagram

So for example the skill could have an intent called GetClosingTime which would check what time does, in this instance Tesco, closes at. But in order to achieve this the developer first need to give the intent scheme some sample utterances so it knows that's the intent the user wants to evoke as a skill can have several intents. In this case, the developer would add sample utterances like “ What time does it close ? “, “When does it close?”, “What is Tesco’s closing time?” and so on. For something more complex like getting the price of a chocolate bar, a slot would have to be put in which would allow the passing of variables. A sample utterance with a slot that a developer would give Alexa could be “ What is the price of {chocolate}?”. In order for utterances with slots to work a Slot Type with Slot Definitions would have to be created. The slot definitions would be the name of chocolate bars so “ Snickers”, “ Milky Way”, “Twix, “Kit Kat” and so on. Mentioning a bar that isn’t on the list will still make the skill trigger but if its an unusual name the program may have a problem finding it.

3.2 Cortana

Cortana is Microsoft’s voice assistant for Windows 10, Xbox, Android and IOS. It launched on Windows 10 in April 2014 and its use is very similar to Mycroft on Linux. Cortana appears on the taskbar beside the start button, the user can type in any question and Corta any question they

have and it will attempt to find and display the answer, if it fails then it will direct the user to the web search instead with Bing will provide more information.

The skill structure is fairly similar to Mycroft and Alexa with intents and utterances. Initially to get started on a skill the user has to create a bot in Azure Bot Service. They have the choice to select one of two options, do they want to create the bot for the skill in .NET or NodeJs. The developer has to provide the display name, invocation name, long description and short description. Once that is done and the bot has been created the user has access to the bot source code which they can edit to create the skill they desire.

While Cortana is nowhere near the most popular voice assistant out there, it serves a specific niche on Windows that Mycroft serves on Linux and due to the similarities of the two it was covered over some of the more popular voice assistants like Siri or Google Assistant.

4. Scratch

Scratch is a simple block based visual programming tool aimed at children to help them understand and get into programming. The tool has been developed by the MIT Media Lab and it is the foundation on which the visual tool for Mycroft will be based. Scratch uses UI blocks with simple text which represent more challenging programming conditions like if statements, for loops, printing strings and so on.

There are ten block colours with each colour representing a different category.

1. Medium blue blocks are called movement blocks and they represent sprite movement.
2. Purple blocks, called looks blocks, are used for sprite appearances.
3. Next are sound blocks which are used to control sound and MIDI functions, they are represented by magenta.
4. Sea green blocks are a pen extension which allows the sprites to draw on the map themselves, they are a new addition to Scratch and only came with Scratch 3, released on the 2nd of January 2019.
5. Control blocks are golden and are used for script control.
6. Brown blocks, called event blocks, are used for triggering scripts to run. Brown blocks are essential to any problem as without them projects would not be able to run unless manual scripts were being run.
7. Sensing blocks are coloured light blue and they are used for detecting different factors for the project like for example if the sprite touches the colour red.
8. Light green or operators blocks are used for scripting simple math equations and string handling.

9. Data blocks have two colour schemes, orange and deep red. The orange blocks are called variable blocks and are used for setting, deleting, hiding and showing variables meanwhile the deep red blocks are the list blocks and are responsible for handling lists.
10. Finally, my blocks. My blocks are an empty block and allows the user to customise them and they are colour coded pink. Thanks to the colour coded nature of the blocks it lets the users quickly memorise and recognise the blocks allowing for quicker learning.

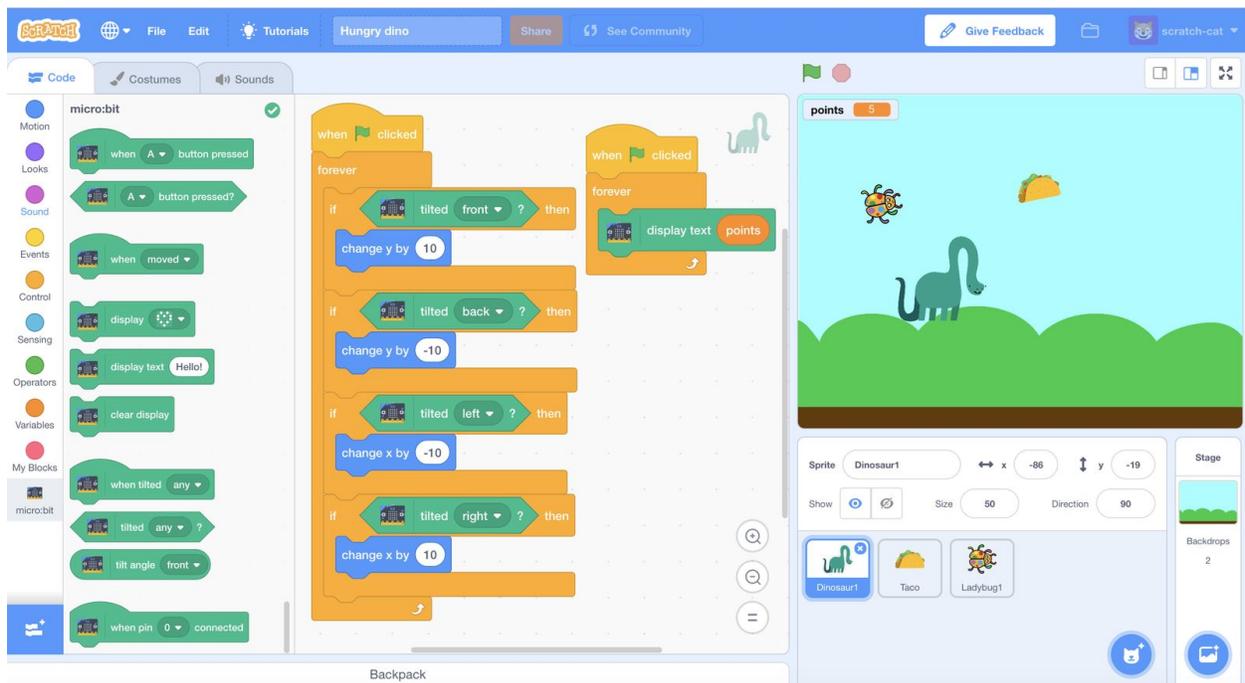


Fig 6. Example of Scratch blocks

Scratch was publicly released on the 8th of January 2007, and over the twelve years it has grown and thought millions of kids every day. The simplicity of its design is carried the project for long and that's why the visual development tool for Mycroft is based on it

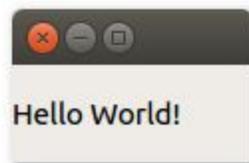
5. Technologies Used

This project will require several tools and libraries working together in order to achieve the desired end result. The language chosen is Python 3.7 as Mycroft is written in Python. This section will cover what framework is expected to be used as well as the choice of the database

5.1 Framework

Two frameworks are being considered for this project, namely PyQt5 and Tkinter. Both of the frameworks offer a unique set of tools to tackle the challenges set by the project.

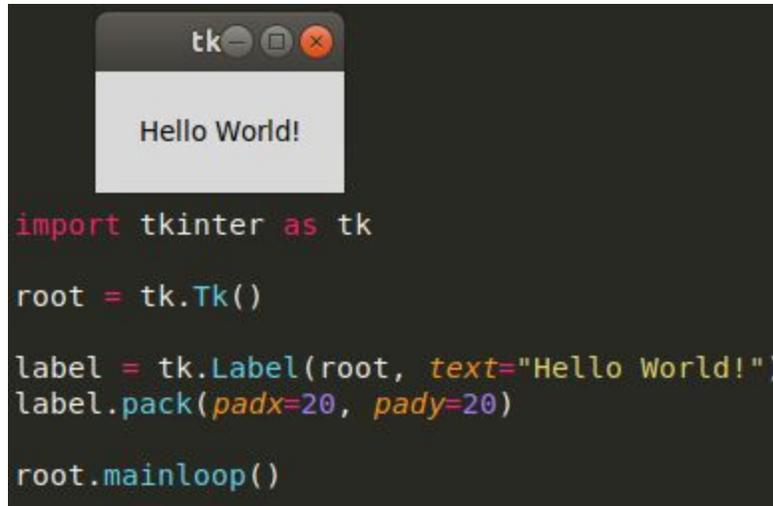
What is PyQt5? In the words of Riverbank Computing, the creators of PyQt, “*PyQt is a set of Python v2 and v3 bindings for The Qt Company's Qt application framework and runs on all platforms supported by Qt ... The bindings are implemented as a set of Python modules and contain over 1,000 classes.*” (Riverbankcomputing.com, n.d.). Why use PyQt5? Qt is written in C++ which allows the user to quickly build applications in Python without losing the speed that C++ provides. Moreover the framework itself comes with a plethora of GUI widgets in it's library and it provides the developer “*with abstract network sockets, threads, Unicode, regular expressions, SQL databases, SVG, OpenGL, XML, a fully functional web browser, a help system, a multimedia framework*” (Riverbankcomputing.com, n.d.). PyQt5 is one of, if not the most, popular GUI frameworks for Python on the market and it is easy to see why it reached such a position.



```
from PyQt5.QtWidgets import *
app = QApplication([])
label = QLabel('Hello World!')
label.show()
app.exec_()
```

Fig 7. PyQt5 example

Tkinter is the defacto GUI toolkit for Python and it is included with the standard Python installs on Windows, Linux and Mac OS. The benefit that Tkinter outside of the vast toolkit that it provides, it's also the default GUI framework for Python it has an abundance of resources like books and tutorials which allows for a quick solution to any obstacles encountered during development.

The image shows a Tkinter window titled 'tk' with a white background and a grey title bar containing the text 'tk' and standard window control buttons (minimize, maximize, close). The window displays the text 'Hello World!' in a simple font. Below the window, the Python code used to create it is shown in a dark-themed code editor. The code is as follows:

```
import tkinter as tk

root = tk.Tk()

label = tk.Label(root, text="Hello World!")
label.pack(padx=20, pady=20)

root.mainloop()
```

Fig 8. Tkinter example

It is expected that Tkinter will be used for the project as it is better suited to smaller apps with simple designs, however this could change once production on the project has begun.

5.2 Database

It is unclear if this project will use a database however databases are used in any application that uses data as they allow a fast and reliable way of storing data as well as give a framework of how the data should be stored. There are two databases that are being considered to be used in the project, PostgreSQL and MySQL. These two databases were selected as they provide a plethora of features and are the two most used databases in the industry and so they have become the industry standard for open source databases

PostgreSQL is a database that emphasizes flexibility and standard compatibility. It is ACID compliant as well as it follows the SQL format it carefully which means that it is easy to migrate to a closed source database like Oracle. Furthermore, the database boasts a large number of features such as user-defined types, table inheritance, locking mechanism, foreign key referential integrity and more. It is the world's most advanced open source database and it is easy to see why it gained and maintained its popularity.

MySQL is currently ranked as the second most popular database just being Oracle and the most popular open source database in the world. It has a somewhat easier learning curve for beginners

however it has less features than PostgreSQL. It provides high speed performance and security to the data only a few databases can match. While it is the most popular open source database in the world, its feature set and benefits lack behind some of its competitors.

PostgreSQL is expected to be used for this project as its larger feature set and strict standards policy will allow easier and more streamlined development of the visual tool.

6. Conclusion

Developing a visual tool for Mycroft will be a challenge and will provide many obstacles along the way; however the analysis of its skill structure as well a look on how Scratch operates and the design decisions that were taken by MIT to keep the project going for so long provides a valuable foundation of how the visual development tool should be built.

To aid me in overcoming the challenge of this task I have chosen Tkinter and PostgreSQL. The Tkinter framework will provide a fast and sufficient way for the user interface to be built and with its vast popularity learning resources such as guides, tutorials and problem solving forum threads will be plentiful. This will allow for more time to be spent developing the project rather than researching problems. Meanwhile, if need be, the PostgreSQL database ticks all of the same boxes as Tkinter with wealth of resources which will once again allow to focus on the development and not problem solving

7. Declaration

- I declare that all material in this submission e.g. thesis/essay/project/assignment is entirely my/our own work except where duly acknowledged.
- I have cited the sources of all quotations, paraphrases, summaries of information, tables, diagrams or other material; including software and other electronic media in which intellectual property rights may reside.
- I have provided a complete bibliography of all works and sources used in the preparation of this submission.
- I understand that failure to comply with the Institute's regulations governing plagiarism constitutes a serious offense.

Student Name: Michal Wojnowski

Student Number: C00213963

Signature: 

Date: 20/04/2020

8. References

Penrod, S. (2017). *Mycroft Architecture: Current and future capabilities*. [video] Available at: <https://www.youtube.com/watch?v=pyc3wWYoI8E>. [Accessed 28 Oct. 2019].

Mycroft-ai.gitbook.io. (n.d.). *Introduction to Skill Development*. [online] Available at: <https://mycroft-ai.gitbook.io/docs/skill-development/introduction> [Accessed 28 Oct. 2019].

Mycroft-ai.gitbook.io. (n.d.). *Your First Skill*. [online] Available at: <https://mycroft-ai.gitbook.io/docs/skill-development/your-first-skill> [Accessed 28 Oct. 2019].

Mycroft Mark 1. (2017). [image] Available at: <https://www.mycroft.ai/wp-content/uploads/2017/02/mark1.png> [Accessed 28 Oct. 2019].

Mycroft Mark 2. (2018). [image] Available at: https://upload.wikimedia.org/wikipedia/en/c/c0/Mycroft_Mark_II_voice_assistant.png [Accessed 16 Dec. 2019]

Mycroft Architecture [image] Available at: <https://image.slidesharecdn.com/aiiotandvoiceasanaturalinterface-171016213751/95/ai-io-t-and-voice-as-a-natural-interface-13-638.jpg?cb=1508189935> [Accessed 16 Dec. 2019]

Mycroft Community Forum. 2020. *Skills Which Have The Same Trigger*. [online] Available at: <https://community.mycroft.ai/t/skills-which-have-the-same-trigger/5880> [Accessed 4 April 2020].

The first-generation Amazon Echo. (2014). [image] Available at: https://upload.wikimedia.org/wikipedia/commons/thumb/5/5c/Amazon_Echo.jpg/800px-Amazon_Echo.jpg [Accessed 31 Oct. 2019].

Developer.amazon.com. 2020. *Rapidly Create Your Alexa Skill Backend With AWS Cloudformation*. [online] Available at: <https://developer.amazon.com/blogs/post/Tx27NAUCY0KQ34D/Rapidly-Create-Your-Alexa-Skill-Backend-with-AWS-CloudFormation> [Accessed 4 April 2020].

Alexa Developers (2017). *Build Voice-Enabled Devices with Amazon Alexa*. [video] Available at: <https://www.youtube.com/watch?v=t58fUp15YB4> [Accessed 27 Oct. 2019].

Decypher Media (2016). *Coding the Amazon Echo (Alexa) - How to create custom Alexa skills from scratch with AWS Lambda*. [video] Available at: <https://www.youtube.com/watch?v=zt9WdE5kR6g> [Accessed 27 Oct. 2019].

Docs.microsoft.com. (n.d.). *Getting started with Cortana skills - Cortana*. [online] Available at: <https://docs.microsoft.com/en-us/cortana/skills/get-started> [Accessed 31 Oct. 2019].

Brown, D. (2018). *Cortana skills development: Get started*. [video] Available at: <https://www.youtube.com/watch?v=BQ40xzk-XVc> [Accessed 31 Oct. 2019].

En.scratch-wiki.info. (n.d.). *Blocks - Scratch Wiki*. [online] Available at: <https://en.scratch-wiki.info/wiki/Blocks> [Accessed 30 Oct. 2019].

Example of Scratch blocks. (2019). [image] Available at: <https://microbit.org/assets/posts/2019-01-14-scratch-meets-microbit.png> [Accessed 30 Oct. 2019].

Riverbankcomputing.com. (n.d.). *What is PyQt?*. [online] Available at: <https://riverbankcomputing.com/software/pyqt/intro> [Accessed 30 Oct. 2019].

DB-Engines. (n.d.). *DB-Engines Ranking*. [online] Available at: <https://db-engines.com/en/ranking> [Accessed 31 Oct. 2019].

Postgresqtutorial.com. (n.d.). *What is PostgreSQL*. [online] Available at: <http://www.postgresqtutorial.com/what-is-postgresql/> [Accessed 31 Oct. 2019].

GitHub. 2020. *Pyqt/Examples*. [online] Available at: https://github.com/pyqt/examples/tree/_/src/01%20PyQt%20QLabel [Accessed 4 April 2020].