

School of Computing

Diploma in Infocomm Security Management

Internet of Things

ST0324

Assignment CA1

**Date of Submission:** 23nd June 2019

**Prepared For:** Ms Dora Chua

**Class:** DISM/FT/3A/31

**Submitted By:**

**Student ID Name**

1637876 Augustus Tan

Table of Contents

[1. Overview of Application 3](#_Toc12155972)

[2. Raspberry Pi Set-up 3](#_Toc12155973)

[3. Web Application 4](#_Toc12155974)

[4. Basic Requirements 7](#_Toc12155975)

[5. Bonus Feature 7](#_Toc12155976)

[6. Quick-Start Guide 7](#_Toc12155977)

[7. Hardware Checklist & Setup Instruction 7](#_Toc12155978)

[Checklist 7](#_Toc12155979)

[Special Hardware Setup 7](#_Toc12155980)

[8. Software Checklist 8](#_Toc12155981)

[9. Source Code 8](#_Toc12155982)

[./app.py 8](#_Toc12155983)

[./config/db\_access.py 8](#_Toc12155984)

[./config/hash.py 8](#_Toc12155985)

[./speech/recorder.py 9](#_Toc12155986)

[./speech/action.py 9](#_Toc12155987)

[./equipment/eq\_db.py 9](#_Toc12155988)

[./equipment/dht11.py 9](#_Toc12155989)

[./equipment/ldr.py 9](#_Toc12155990)

[./static/extrafunction.js 9](#_Toc12155991)

[./static/chart\_graph.js 9](#_Toc12155992)

[10. How It Works & Future Plans 10](#_Toc12155993)

[How It Works 10](#_Toc12155994)

[Future Plans 10](#_Toc12155995)

[11. Reference 10](#_Toc12155996)

# Overview of Application

HeyPi is an application that function similarly to many other smart voice assistant. HeyPi is easy to use and it provides an web interface to allow users to view logs, customise many different functions and add custom replies. To use HeyPi, users have to press and hold the button, then speak into the microphone. Release the button once the command is given. HeyPi will then process the voice into text, then reply to the user with a customisable voice stating that the function is done. Functions such as stating the temperature, brightness or toggling the LED.

The target audience for HeyPi is for anyone that have smart devices at home. It can be controlled with HeyPi simply by connecting to it. It provides convenience for home users to control their smart devices, and also crafting custom replies have never been easier. As compared to other smart assistant, HeyPi allows one to quickly add new conditions on the website in just a few clicks away.

1. Raspberry Pi Set-up

The Raspberry Pi set-up is messy, but it’ll be explained below.



Figure : Overview

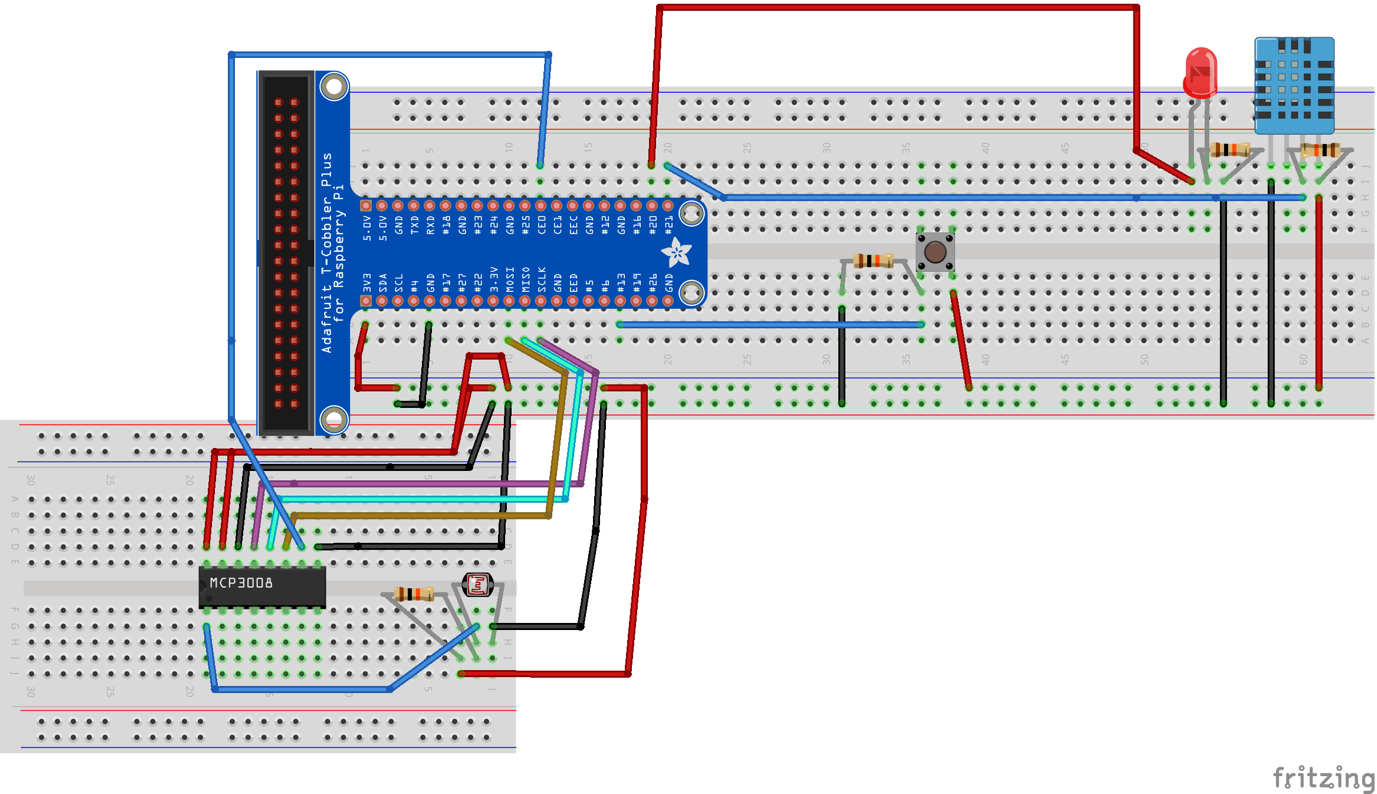


Figure : Fritzing Diagram

The basic set-up requires DHT11 sensor, LDR sensor, LED.

As for setting up the voice input and output, a microphone for input, and a speaker or earpiece for output is required. Unfortunately, through numerous tests and research, the Raspberry Pi only outputs audio to the 3.5mm jack. Therefore, an USB sound card does not work.

In my set-up, I used 2 earpiece (1 required to have microphone, another is optional). The earpiece with microphone is plugged into the USB sound card and the one without it, is plugged into the 3.5mm jack.

1. Web Application

The web application have a clean and sleek view. It contains 6 pages.

* Login
  + For users to login
* Home
  + User Dashboard contains dynamic text update
* Statistics
  + Provides live dynamic graphs data that are grabbed from the database
* Logs
  + Commands that were given by the users converted to text
* Register
  + Register a new user
* Settings
  + Add or delete new conditions and configure GPIO pins

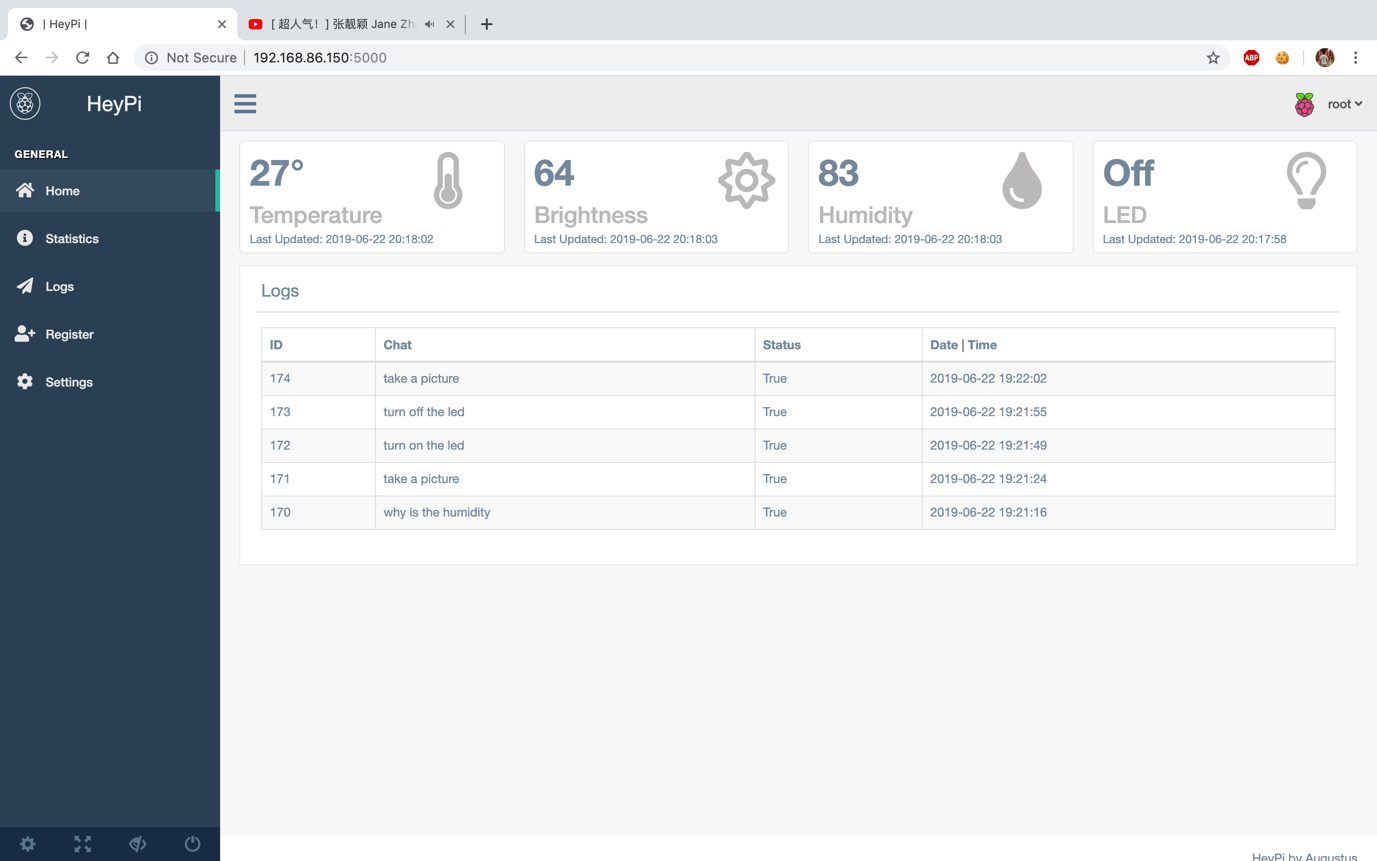


Figure : Dashboard

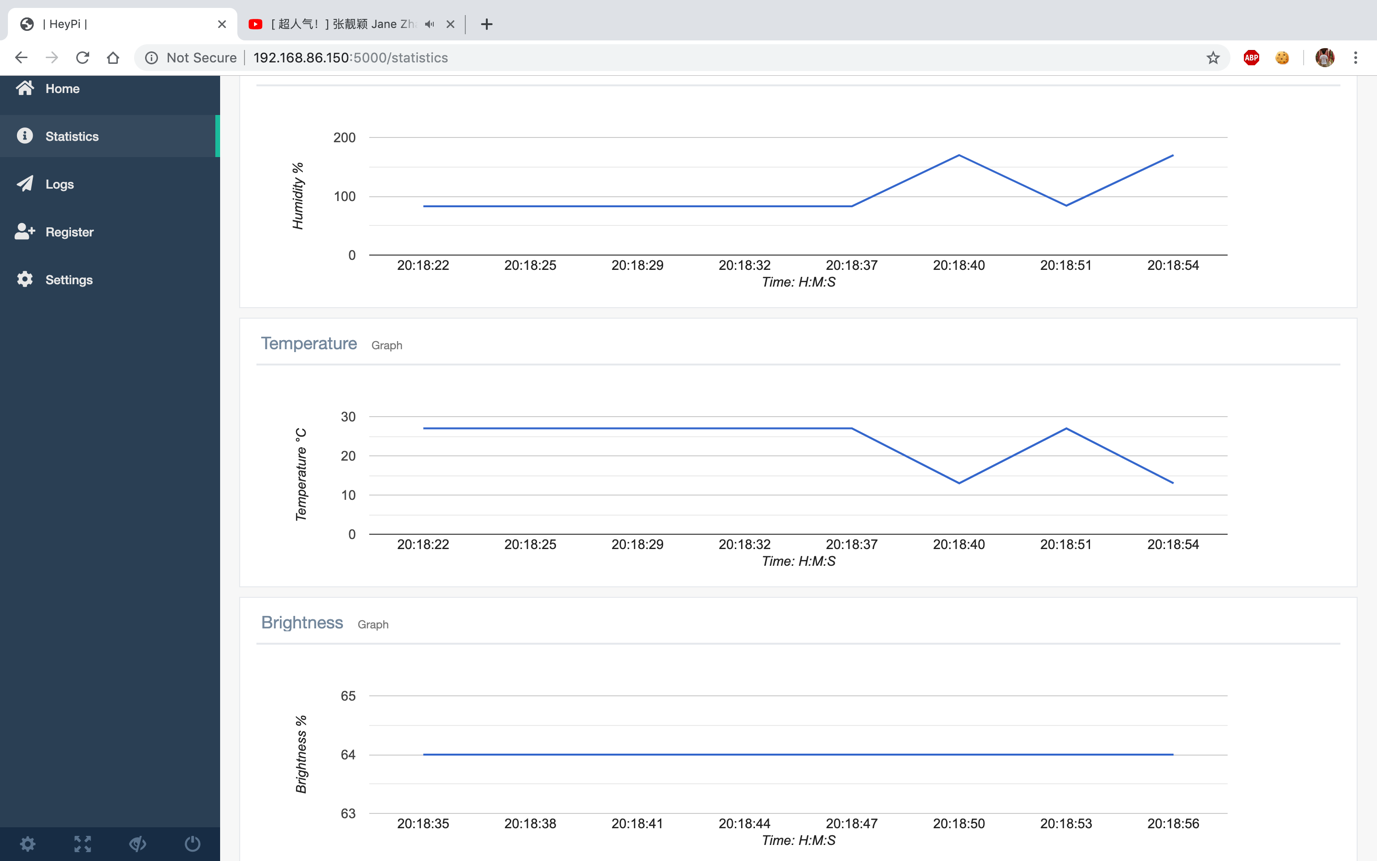


Figure : Statistics

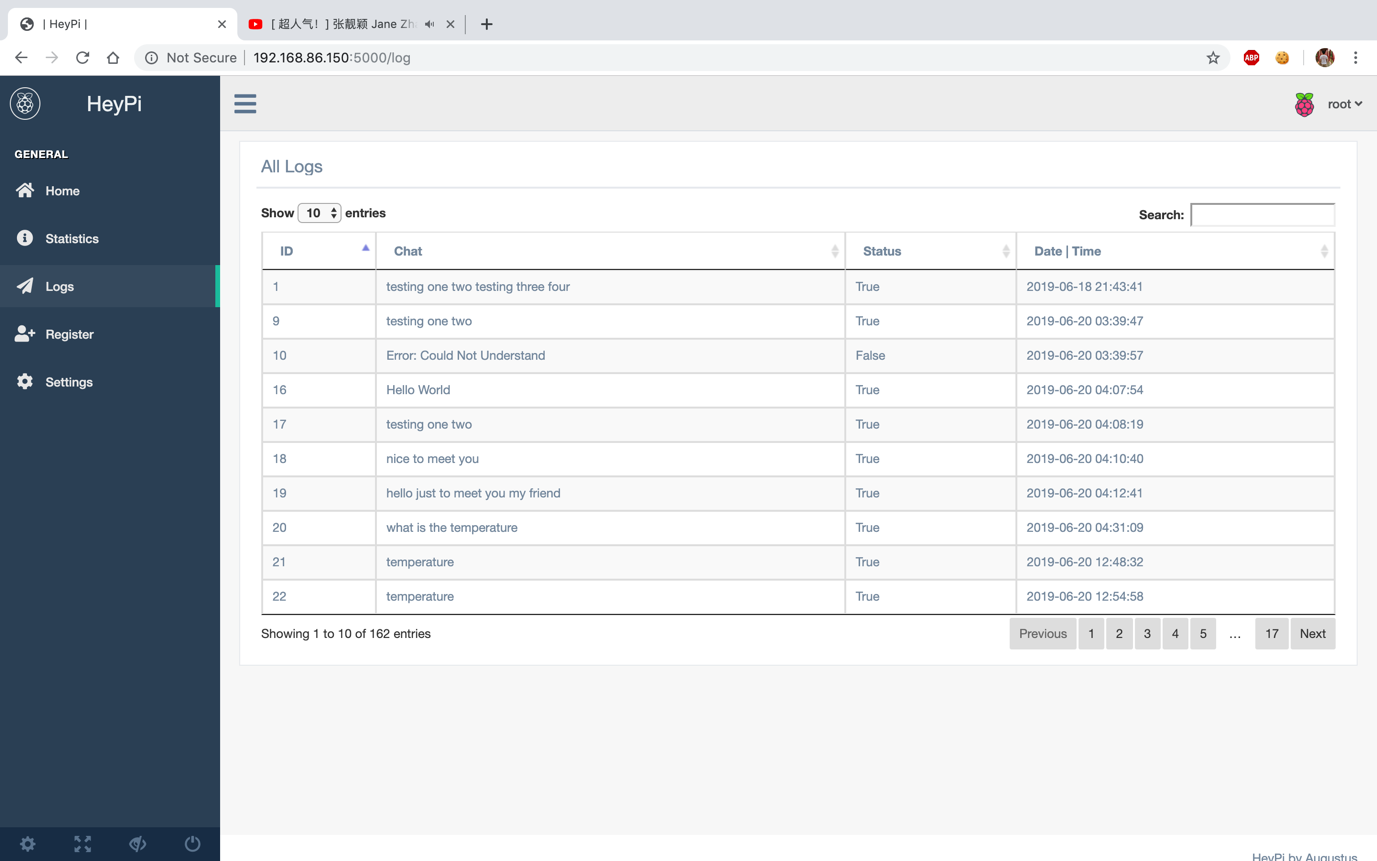


Figure : Speech Logs

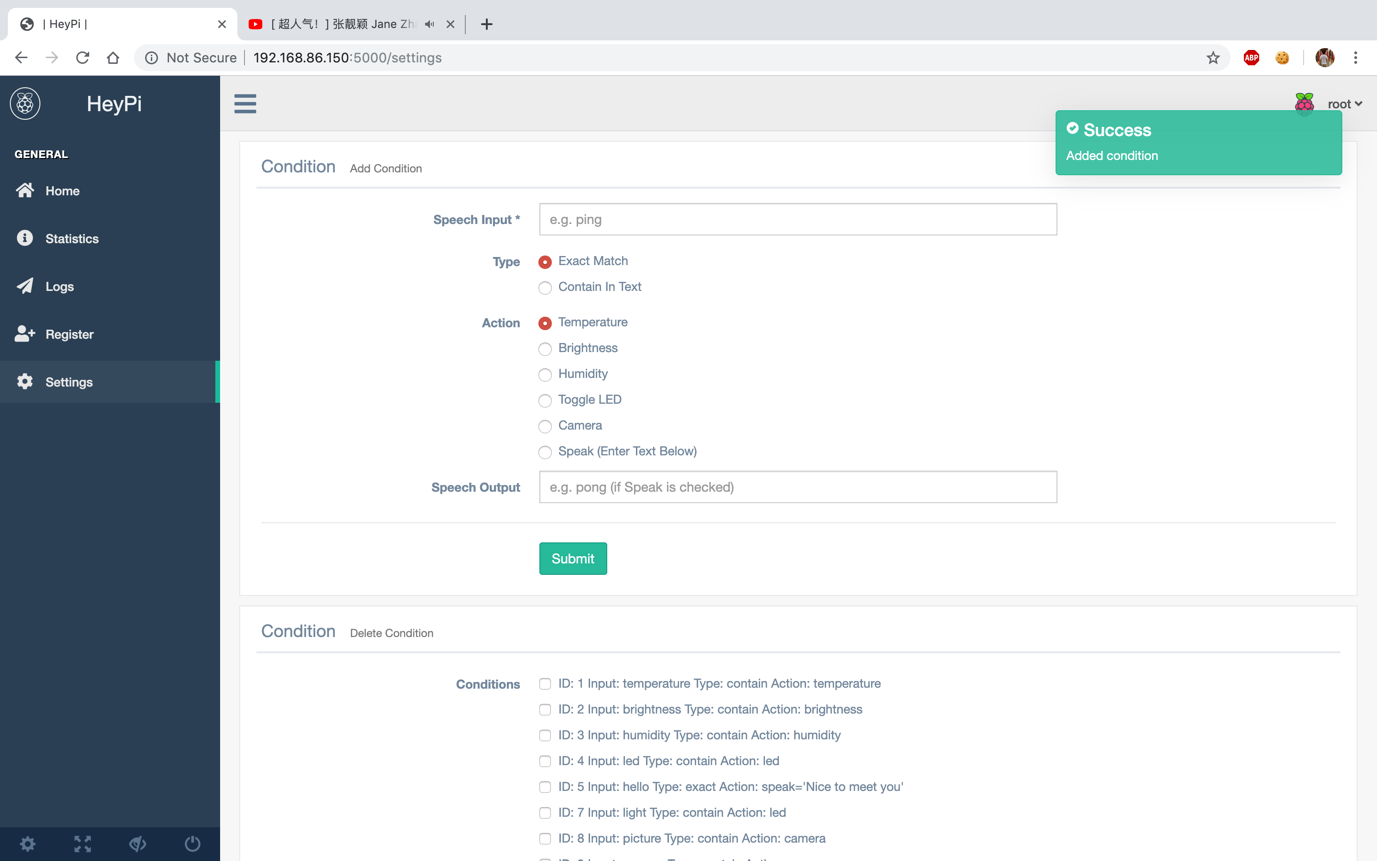


Figure : Settings

1. Basic Requirements

|  |  |
| --- | --- |
| Requirements | Evidence |
| Used Two Sensors | Used DHT11 & LDR |
| Provide Real-Time Values | Showed the real-time value of temperature, brightness and humidity |
| Provide Historic Values | Showed Chat, Temperature, Humidity and Brightness logs |
| Control Actuator | Button on website is able to control the LED |

1. Bonus Feature

* Raspberry Pi is able to take in audio and output audio
* Asynchronous voice commands to control LED, get sensors value and take picture
  + Pictures are saved at (script location)/picture
* Audio Input & Output is highly customisable. GPIO Pins are customisable too
* Password is hashed for security
* A login system

1. Quick-Start Guide
2. Connect an USB audio input
3. Connect a 3.55mm jack audio output
4. Run app.py (sudo python3 app.py)
   * The entire program is packaged for convenient!
5. Hold the button
6. Start Talking!
7. Release the button to get reply
8. Hardware Checklist & Setup Instruction

## Checklist

* 1x Audio Input (Microphone) \* USB
* 1x Audio Output (Earpiece/Speaker) \* 3.5mm jack
* 1x DHT11 Sensor
* 1x LED
* 1x Light Dependent Resistor (LDR)
* 1x MCP3008 (Analog to Digital)
* 1x Button
* Camera

## Special Hardware Setup

* Audio Input MUST be USB
  + Raspberry Pi default does not have an audio input. The 3.5mm jack only supports audio output.
* Audio Output MUST be 3.5mm jack
  + Through multiple test, the Raspberry Pi is configured to output audio to only either the 3.5mm jack or Bluetooth.

1. Software Checklist

Do run the following commands before using HeyPi

|  |
| --- |
| sudo apt-get install python3  sudo apt-get install espeak  sudo apt-get install flac  sudo pip3 install –upgrade pip  sudi pip3 install flask  sudo pip3 install SpeechRecognition  sudo pip3 install python3-dev  sudo pip3 install python3-pyaudio  sudo pip3 install adafruit\_dht  sudo pip3 install gpiozero  sudo pip3 install mysql-connector-python  sudo pip3 install espeak  sudo pip3 install picamera |

HeyPi is coded in Python3 instead of Python2 due to Python2 is going to be depreciated in 2020.

1. Source Code

Double Click The Icons To View The Source Code File

## ./app.py



## ./config/db\_access.py



## ./config/hash.py



## ./speech/recorder.py



## ./speech/action.py



## ./equipment/eq\_db.py



## ./equipment/dht11.py



## ./equipment/ldr.py



Source code for website is not provided due to too much file. However, the custom JavaScript that was written by myself will be provided below.

## ./static/extrafunction.js



## ./static/chart\_graph.js



1. How It Works & Future Plans

## How It Works

1. When the button is pressed, HeyPi will spawn a thread to start recording the audio with the help of PyAudio (a library that allows recording with python)
   * If the thread isn’t spawn, the entire program will stop working
2. Once the button is released, HeyPi will output the audio to a file called output.wav
3. The program will then process the audio file and interpret the speech from the user
4. Once the speech is confirmed, it will move on to compare the speech with conditions that was given by the user
5. Lastly, once the condition is met, an action will be done to responds to the user.

## Future Plans

HeyPi is a program that can be develop even much more. However, raspberry pi have limited processor and ram memories. If possible, the implementation of AI and natural processing language will be much better.

1. Reference

Amos, D. (2018). *The Ultimate Guide To Speech Recognition With Python – Real Python*. [online] Realpython.com. Available at: https://realpython.com/python-speech-recognition/ [Accessed 20 Jun. 2019].

Pham, H. (n.d.). *PyAudio Documentation — PyAudio 0.2.11 documentation*. [online] People.csail.mit.edu. Available at: https://people.csail.mit.edu/hubert/pyaudio/docs/ [Accessed 20 Jun. 2019].