GNG2101 Report Template

Project Deliverable F: Prototype II

Submitted by

C23, TalkBox

Victoria Jancowski, 300203985

Kaiyi Yuan, 8617972

Michael Hetu, 300209299

Chuanzhi Li, 300055864

Date February 21, 2021

University of Ottawa

Abstract

This report details the organization of the various prototypes and prototyping tests to develop a fully functional prototype that fulfills the previously identified needs of clients, Roy, and Anthony. It includes both the physical and software prototypes, and the testing of these individual components, as well as the feedback from both Anthony and Roy from the third client meet. The final prototype will be based of both the tests and the protypes realized in this report and will be combined with the feedback from the clients to final a completed design. Also included are further recommendations to the prototypes and specified levels of testing that need to be completed for the final product.

Table of Contents

[Abstract i](#_Toc66000446)

[Table of Contents ii](#_Toc66000447)

[List of Figures iv](#_Toc66000448)

[List of Tables v](#_Toc66000449)

[List of Acronyms vi](#_Toc66000450)

[1 Introduction 7](#_Toc66000451)

[2 Client Feedback Summary 8](#_Toc66000452)

[3 Prototype II 9](#_Toc66000453)

[3.1 Software Prototypes 9](#_Toc66000454)

[3.1.1 Graphical User Interface 9](#_Toc66000455)

[3.1.2 Joystick control code 12](#_Toc66000456)

[3.2 Physical Prototypes 14](#_Toc66000457)

[3.2.1 Raspberry Pi and Screen 14](#_Toc66000458)

[4 Prototype II Testing 16](#_Toc66000459)

[4.1 Software Testing 18](#_Toc66000460)

[4.1.1 Graphical User Interface Testing 18](#_Toc66000461)

[4.2 Physical Testing 21](#_Toc66000462)

[4.2.1 Raspberry Pi Operating System 21](#_Toc66000463)

[4.2.2 Software Test on the Hardware 22](#_Toc66000464)

[5 Conclusions and Recommendations for Future Work 24](#_Toc66000465)

[6 Bibliography 25](#_Toc66000466)

[APPENDICES 26](#_Toc66000467)

[APPENDIX I: Project Management 26](#_Toc66000468)

List of Figures

[Figure 3.1.1.1 - GUI Code 9](#_Toc66009708)

[Figure 3.1.1.2 - GUI Code 10](#_Toc66009709)

[Figure 3.1.1.3 - GUI Code 10](#_Toc66009710)

[Figure 3.1.1.4 - GUI Code 11](#_Toc66009711)

[Figure 3.1.1.5 - GUI Code 11](#_Toc66009712)

[Figure 3.1.1.6 - GUI Code 12](#_Toc66009713)

[Figure 3.1.2.1 - Example of Joystick Configuration – Provided by RaspberryPi Spi 13](#_Toc66009714)

[Figure 3.2.1.1 - Raspberry Pi with Installed Heatsinks 14](#_Toc66009715)

[Figure 3.2.1.2 - Raspberry Pi with Installed Heatsinks 15](#_Toc66009716)

[Figure 3.2.1.3 - LCD Screen on Raspberry Pi 15](#_Toc66009717)

[Figure 3.2.1.4 Side View of Encased Raspberry Pi and Screen 16](#_Toc66009718)

[Figure 4.1.1.1 - Main Menu Categories 18](#_Toc66009719)

[Figure 4.1.1.2 – Social Interactions Phrases 19](#_Toc66009720)

[Figure 4.1.1.3 - Smart Home Interaction Phrases 19](#_Toc66009721)

[Figure 4.1.1.4 Caretaker Interactions Phrases 20](#_Toc66009722)

[Figure 4.1.1.5 - Grocery Shopping Sub Menu 20](#_Toc66009723)

[Figure 4.1.1.6 - Doctors Sub Menu 21](#_Toc66009724)

[Figure 4.2.1.1 Raspberry Pi with OS and Screen On 22](#_Toc66009725)

[Figure 4.2.2.1 - Software Test With Raspberry Pi and Python Code 23](#_Toc66009726)

[Figure 4.2.2.2 Phrases Menu on the Assembled Hardware 23](#_Toc66009727)

List of Tables

[Table 2.1 - Client Meet Feedback 8](#_Toc66000469)

[Table 4.1 - Table of Cumulative Testing 16](#_Toc66000470)

List of Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| GUI | Graphical User Interface |
| LCD | Liquid Crystal Display |
| OS | Operating System |

# Introduction

This report outlines the integrated prototypes that were previously constructed into the beginnings of a fully functioning and comprehensive prototype. The software prototypes have been updated and the functions (text-to-speech, reading of the menu, navigating through menus) have been implemented into the main program. The physical components of the projects were combined by the installation of heatsinks to the raspberry pi, the uploading of the Raspbian operating system to the raspberry pi, and the connection of the LCD screen to the raspberry pi. As well prototype II includes the combination of the software components with the hardware components. This was achieved through uploading the current graphical user interface to the microSD card (inserted into the raspberry pi), that houses the operating system and uploading the Microsoft Excel sheet that contains both the categories and the phrases to the microSD as well.

Like the previous prototypes, prototype II was further tested and appraised based off of the client feedback provided in client meet three. Based on the analysis of this prototype, within the last few weeks of the design process, the final product is subject to debugging, completion of the joystick navigation system, and final adjustments. It will then be distributed to the clients for further testing.

# Client Feedback Summary

The feedback provided by Roy and Anthony from client meet three is organized into Table 2.1. This takes the suggestions from the observations of the clients, and organizes them for a simple way to view their meaning and intent behind what would improve the TalkBox device.

Table 2.1 - Client Meet Feedback

|  |  |
| --- | --- |
| Client Observation | Feedback |
| Too big a gap on the screen under welcome to TalkBox. | Remove welcome to talk box and the blank space under it. Fill it with the categories and phrases and make the font as big as possible. |
| No settings button. | Create a settings button so that people can change the joystick function, the size of the font, and the colors of the text(Black and Yellow, Day Mode, Night Mode). |
| Joystick will move fast. | Make sure the Joystick will be programmed to move slow. |
| Too much connecting with power supply. | Just use a USB connection to the wheelchair battery, for now just use computer power supplied from USB Port. |
| Bright screen on all the time can hurt eyes. | Consider an idle state. If someone has not used the program for a period of time, let the screen go black and awaken it with the joystick. |
| Gooseneck is too long and cumbersome. | Implement a smaller gooseneck in the design. (Locline or Rammount) |

# Prototype II

## Software Prototypes

### Graphical User Interface

The graphical user interface allows the user to navigate through the categories of phrases they would like to access, and upon selection shows the phrases that they can choose. When the desired category, or the desired phrase is selected, the text-to-speech function will be engaged, and a voice will speak the selection. At the end of the listed categories, there will be an exit application button, and on the phrases menu there will be a return to main menu button. This will allow the user the ability to navigate the program as well as speak through the program. Also included in this GUI is the caretaker modification ability that imports the desired phrases and categories from a modifiable Excel Document.

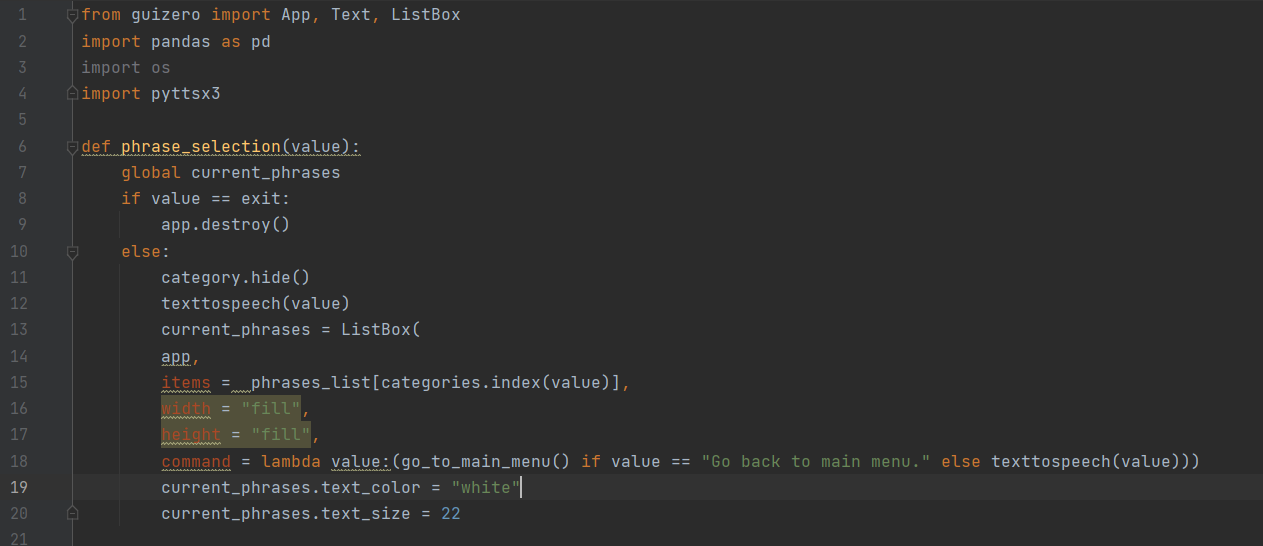


Figure 3.1.1.1 - GUI Code

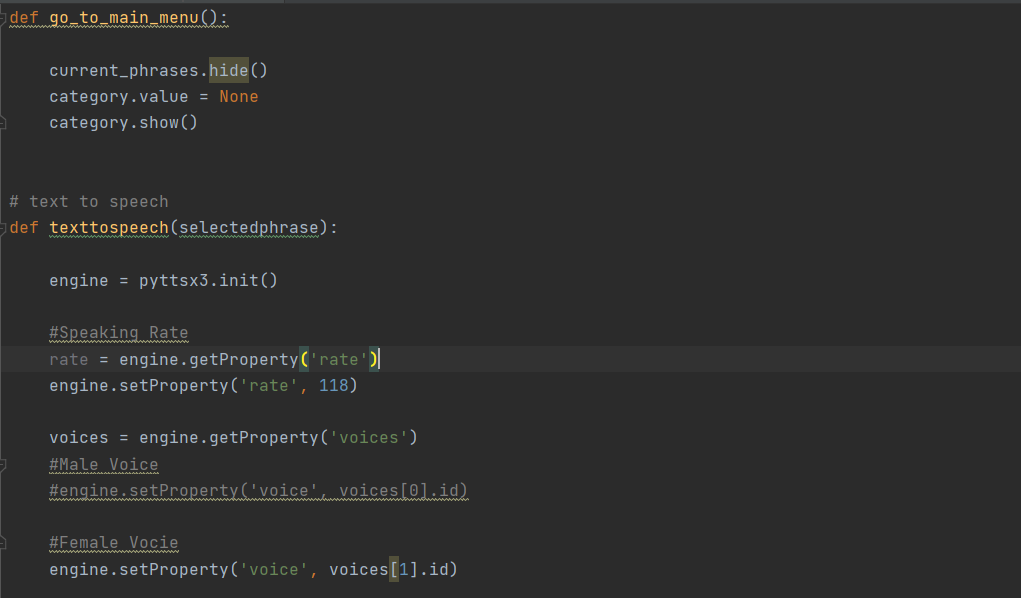


Figure 3.1.1.2 - GUI Code

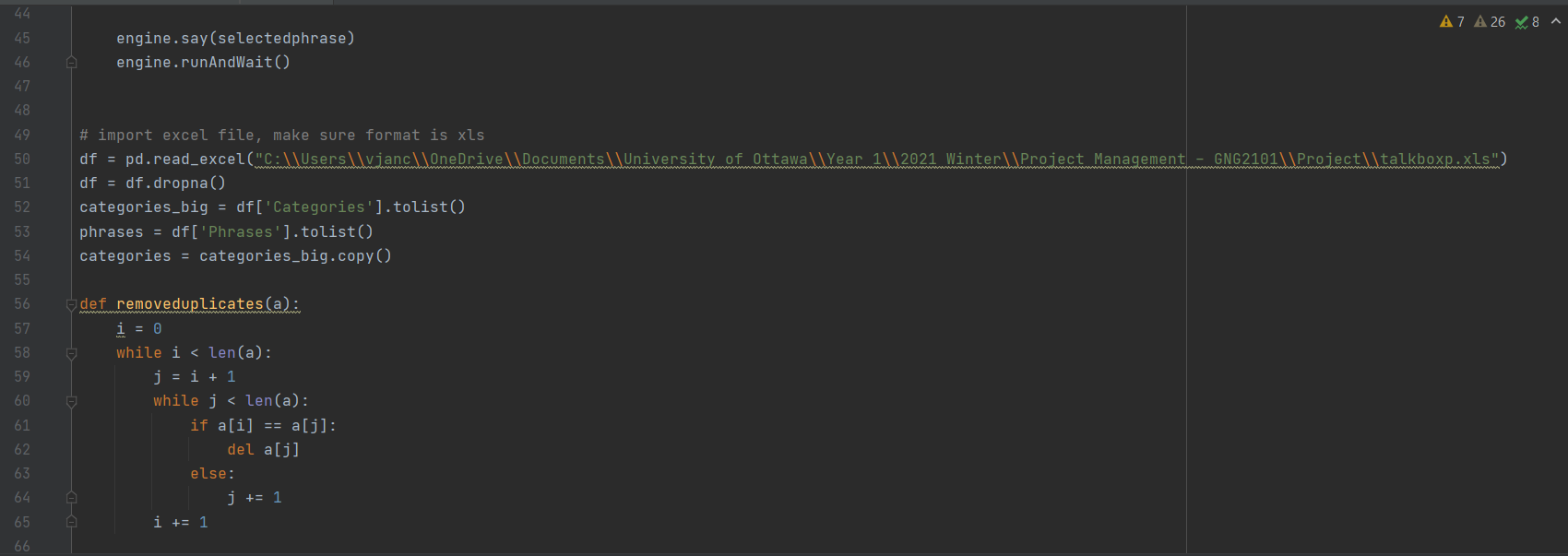


Figure 3.1.1.3 - GUI Code

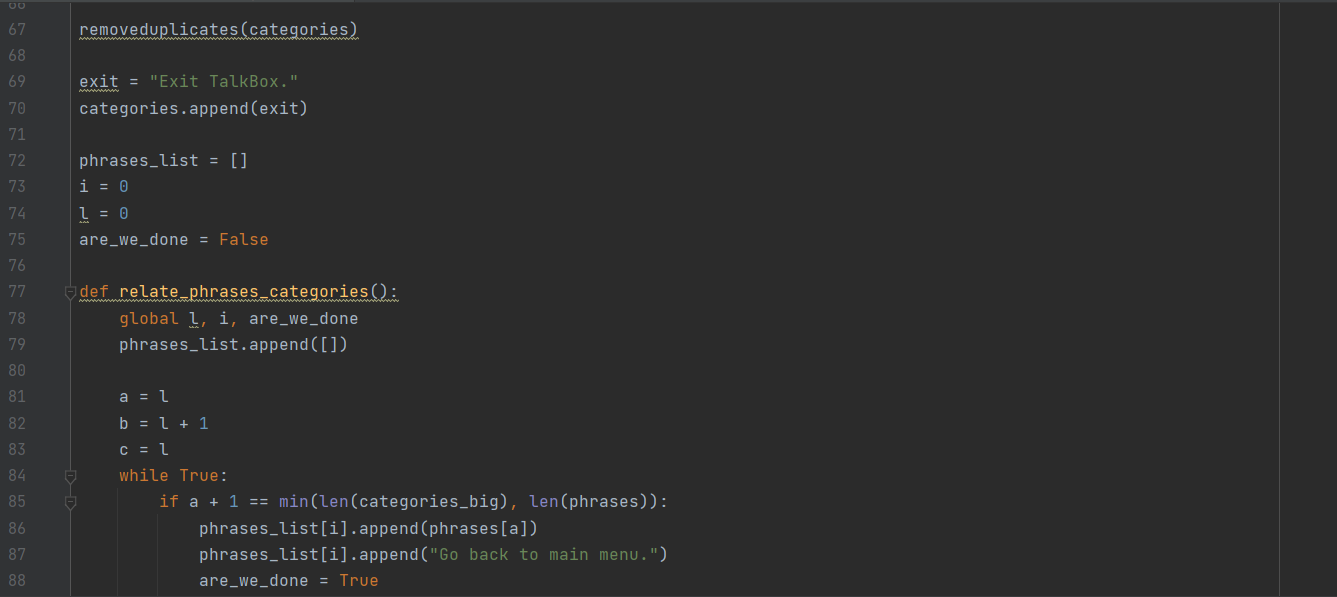


Figure 3.1.1.4 - GUI Code

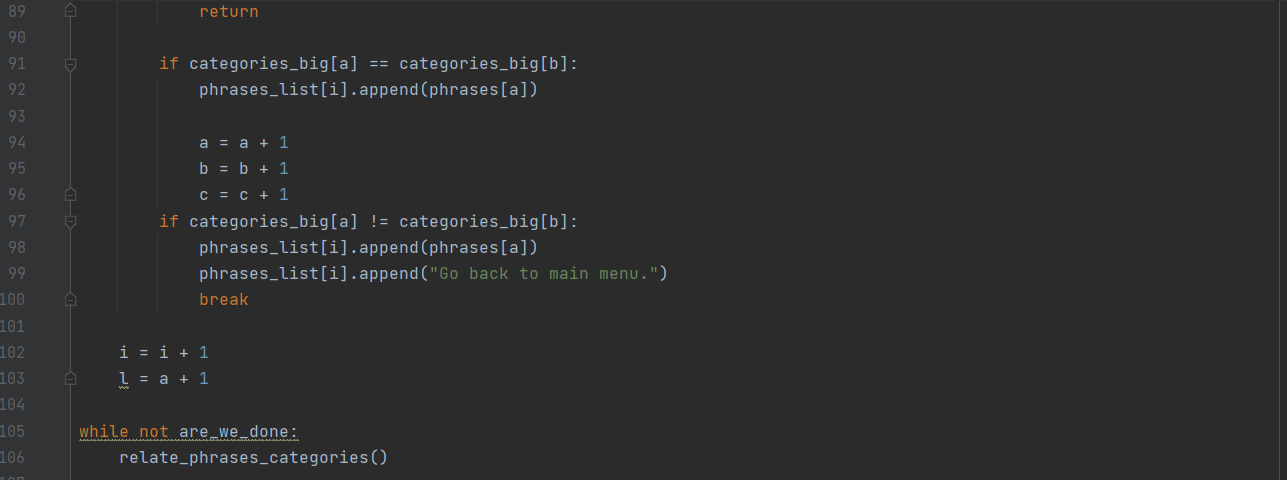


Figure 3.1.1.5 - GUI Code

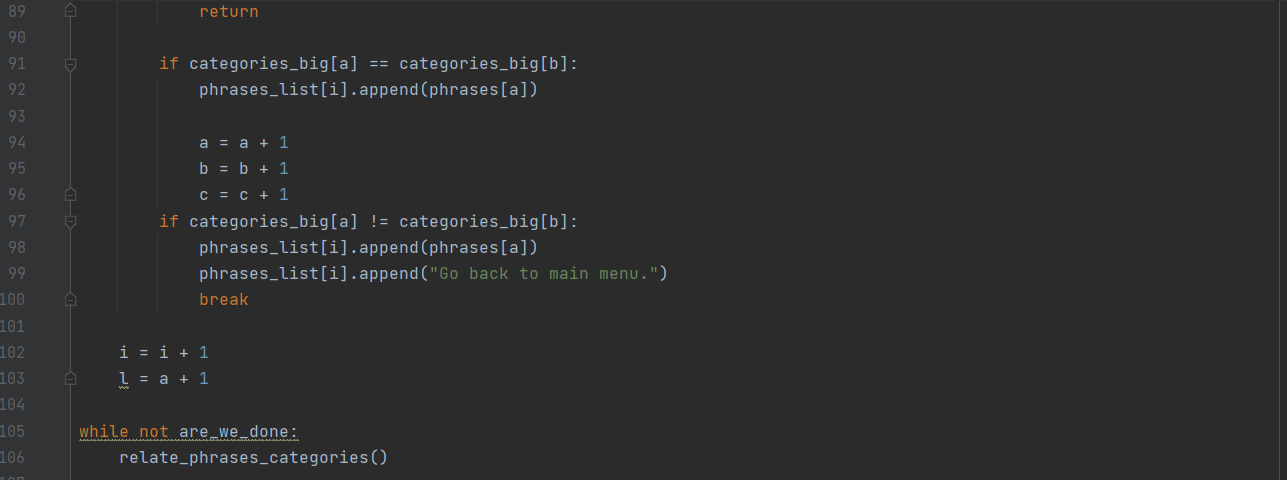
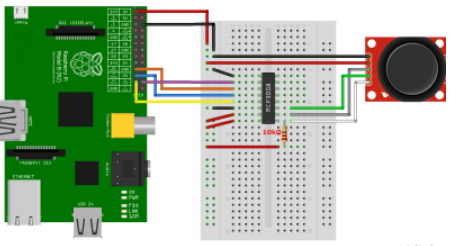


Figure 3.1.1.6 - GUI Code

### Joystick control code

This is the first attempt at creating a code that controls the joystick, and its functionality. The next step in prototyping with combine the code with the graphical user interface, so that the GUI will be completely navigable. The joystick will provide users with the ability to control the choices of categories and phrases that they desire to be spoken. And offer a way to scroll through the phrases and categories.

|  |
| --- |
| #!/usr/bin/python  import spidev #The spidev library is used in the script  import os  import time    # Define Axis Channels, Channel 3 to 7 can be assigned for more buttons or joysticks  swt\_channel = 0  vrx\_channel = 1  vry\_channel = 2  #Time delay, the time it takes to read out the value  delay = 0.5    # Spi oeffnen  spi = spidev.SpiDev()  spi.open(0,0)  spi.max\_speed\_hz=1000000  # Function for reading the MCP3008 channel between 0 and 7  def readChannel(channel):  val = spi.xfer2([1,(8+channel)<<4,0])  data = ((val[1]&3) << 8) + val[2]  return data      # endless loop during the opeartion of the joystick  while True:  # Determine position of the targeted point on the screen  vrx\_pos = readChannel(vrx\_channel)  vry\_pos = readChannel(vry\_channel)  # SW determine  swt\_val = readChannel(swt\_channel)  # output  print("VRx : {} VRy : {} SW : {}".format(vrx\_pos,vry\_pos,swt\_val))  # wait  time.sleep(delay)  #the MCP3008 returns a value between 0 and 1023. If the joystick is in the middle, around 512 will be returned. 0 is returned at one end of one axis and 1023 at the other end. |

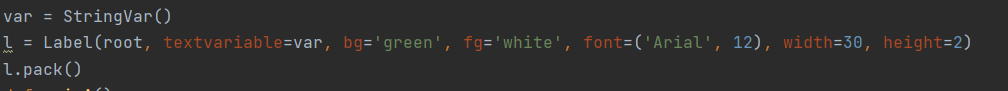


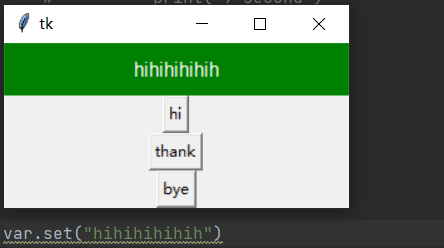
***3.1.2: Code for scan menu and test***

***High lighting***

We choose the label of tkinter as a highlighting way.

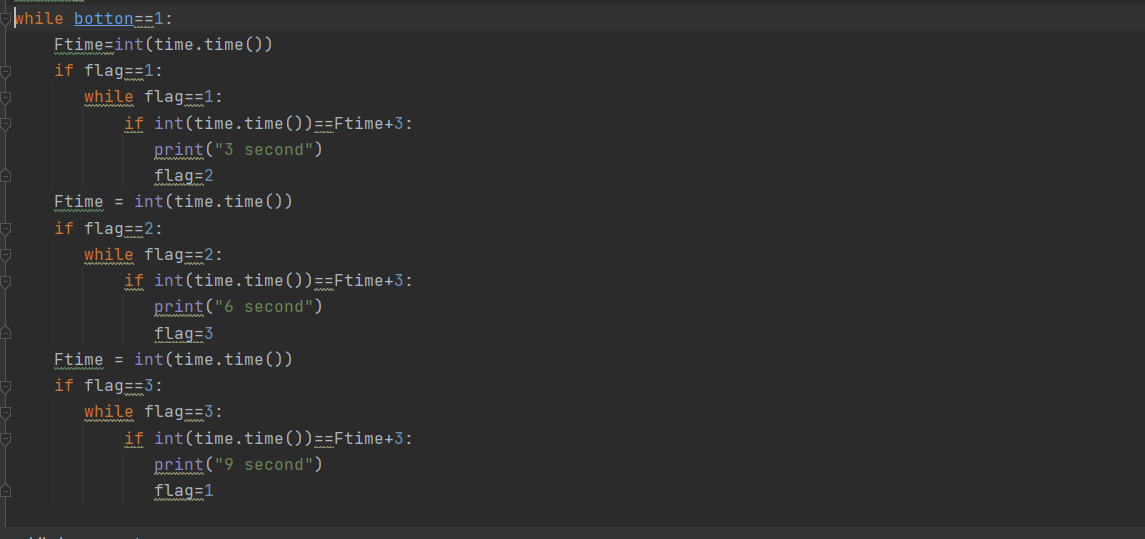
The label will show the ongoing the button



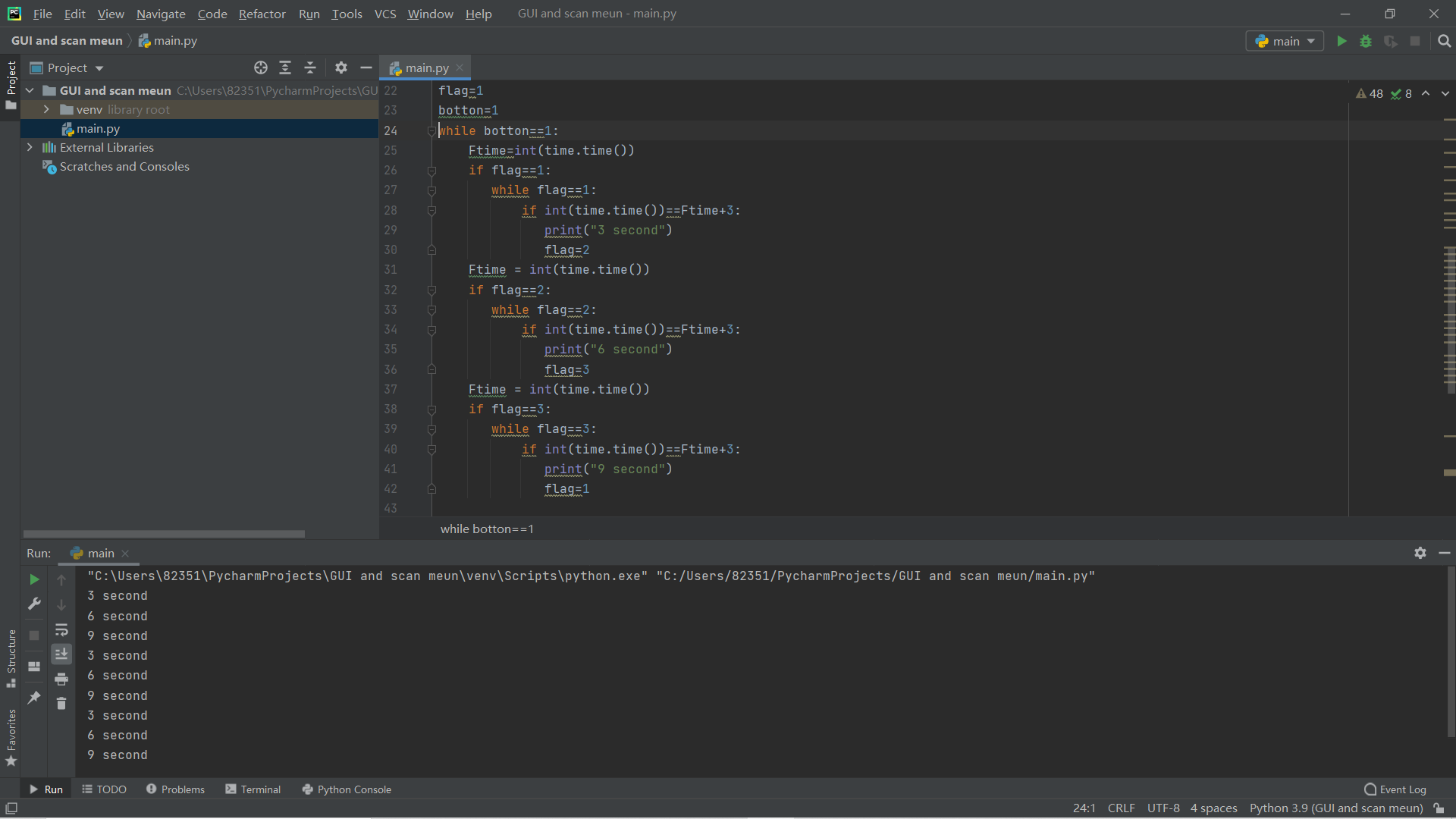


***Time counter***

We choose the time library to achieve the time counter



It can achieve that choosing the arbitrary time as one counter and counter loop of arbitrary phase.



The phase “3 second” etc. can be replaced to any phase we want to.

Also, the print function can be replaced by ver.set to make change on label of GUI (achieve the requirement). However, the code seems unable to work with GUI at the same time.

**For now, I am unable to combine it together because of some technical issues.**

Figure 3.1.2.1 - Example of Joystick Configuration – Provided by RaspberryPi Spi

## Physical Prototypes

### Raspberry Pi and Screen

The combined components of the Raspberry Pi and the LCD allow the user to see the screen and make modifications to the Excel Document through touch. Both were attached together and secured using a case, encapsulating the two. A microSD card was also used to load the operating system (Raspbian), the python files, and the Excel Document. Two heat sinks were also installed: one on the CPU and one on the network card.



Figure 3.2.1.1 - Raspberry Pi with Installed Heatsinks



Figure 3.2.1.2 - Raspberry Pi with Installed Heatsinks



Figure 3.2.1.3 - LCD Screen on Raspberry Pi



Figure 3.2.1.4 Side View of Encased Raspberry Pi and Screen

# Prototype II Testing

Below is the table illustrating the current tests done, and to be done, and what has been tested and verified up until now. The testing conducted over the course of prototype II was of the continuously updated software, and of the construction of the physical device. It also tests the combination of the software and the hardware all in one.

Table 4.1 - Table of Cumulative Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Metric | Unit | Target Value | Actual Value | Justification | Verified Through Testing |
| Device will provide user with text on screen. | Y/N | Y | Y | Device will be easy to use and provide user with visual cues. | Yes |
| Button/switch will reset when user has lifted hand. | N | 3.5 | N | Important that the button functions for all types of users with all types of dexterity. | Not Yet |
| Software will be easily modified by users through settings button. | Y/N | Y | N | Important that software is easy to use for all persons. | Not Yet |
| Voice Interaction with Smart Devices | Y/N | Y | Y | Client requested that interaction with smart devices be available. | Yes |
| Device functions offline. | Y/N | Y | Y | Offline ability will allow user to operate anywhere. | Yes |
| Text-to-Speech Ability | Y/N | Y | Y | Prototype will have ability to provide user with simplified communication. | Yes |
| Requires a simple switch or a button. | Y/N | Y | N | Allows the user to navigate the program easily. | Not Yet |
| Simple User Interface | Y/N | Y | Y | Requested by the client for function ability. | Yes |
| Adaptable User Functionality | Y/N | Y | N | Allows user to modify the program according to individual needs. | Not Yet |
| Device possess large, diverse phrase bank. | kB | 1.5 | Y | Program will allow user to use as many or as little phrases as they desire. | Yes |
| The device is portable through mount. | Y/N | Y | N | Client requested the device be portable through use of wheelchair. | Not Yet |
| The device will charge through wheelchair. | kWh | .015 | Y | Client requested the device be chargeable through use of wheelchair battery. | Yes |
| Device is durable. | Y/N | Y | N | Device will be drop tested in order to ensure durability | Not Yet |
| Device is inexpensive. | $ | 0 | N | The cost for the user will be low. | Not Yet |
| Device is Eco-Friendly. | Y/N | Y | N | The device will provide a neutral or negative carbon footprint. | Not Yet |

## Software Testing

### Graphical User Interface Testing

Below are the images of the graphical user interface being tested. The prototype meets the expected results by compiling successfully and completing the tasks designed for the TalkBox. The program navigates in the proper way upon selection of the categories and of the phrases and uses text-to-speech on the selected phrases and categories. The application also goes back to the previous menu upon selection and exits the app upon selection as well.

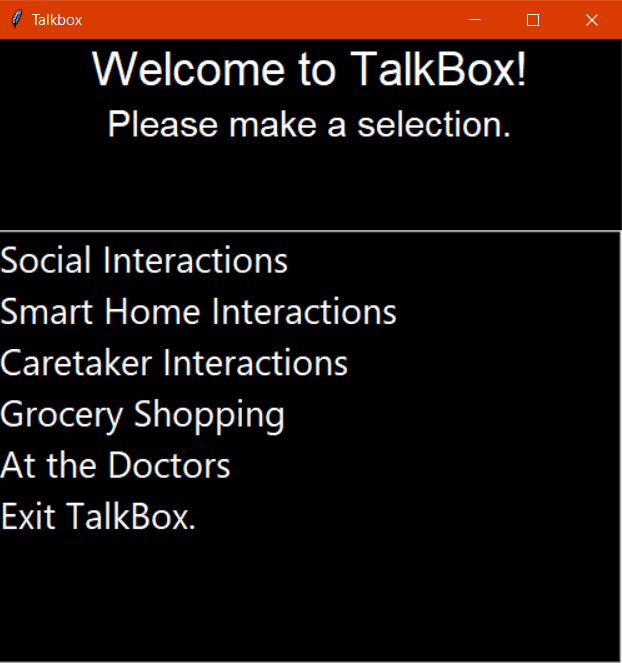


Figure 4.1.1.1 - Main Menu Categories

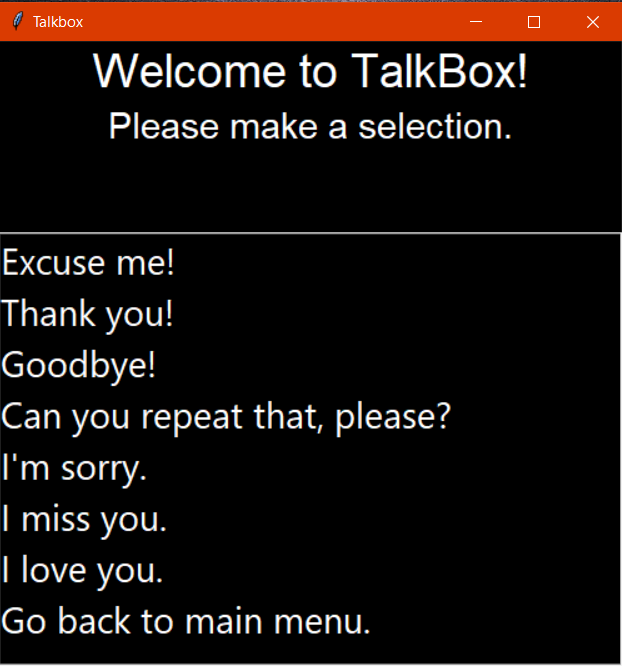


Figure 4.1.1.2 – Social Interactions Phrases

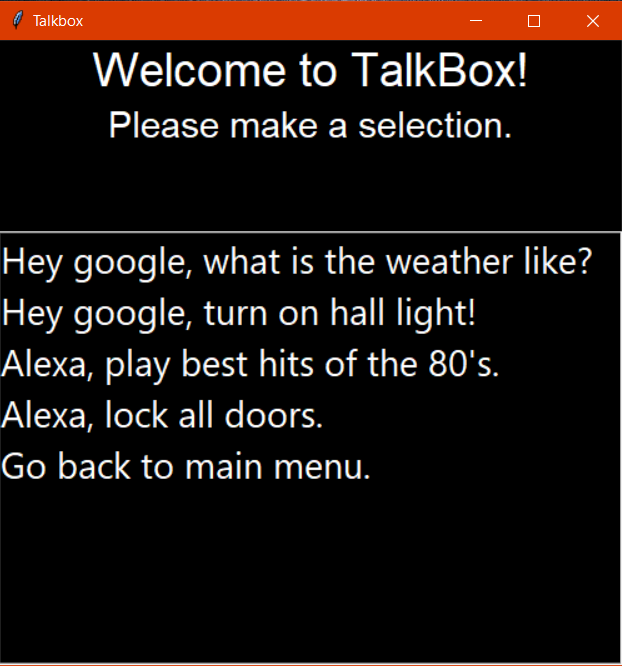


Figure 4.1.1.3 - Smart Home Interaction Phrases

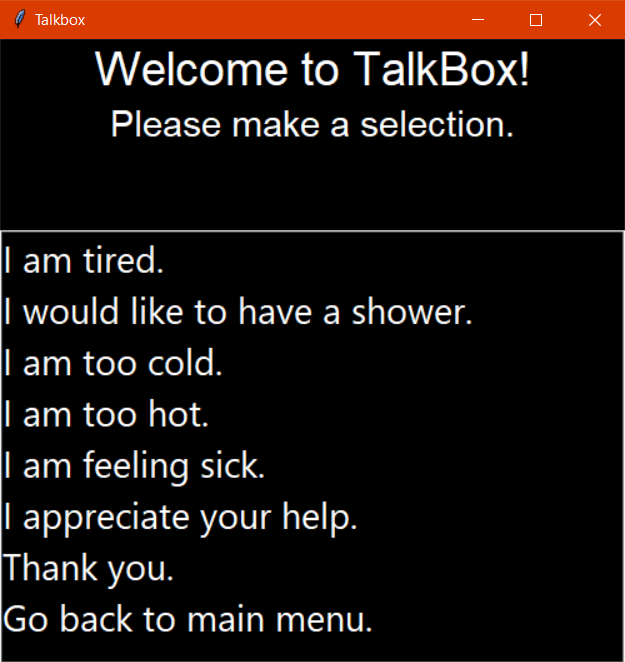


Figure 4.1.1.4 Caretaker Interactions Phrases

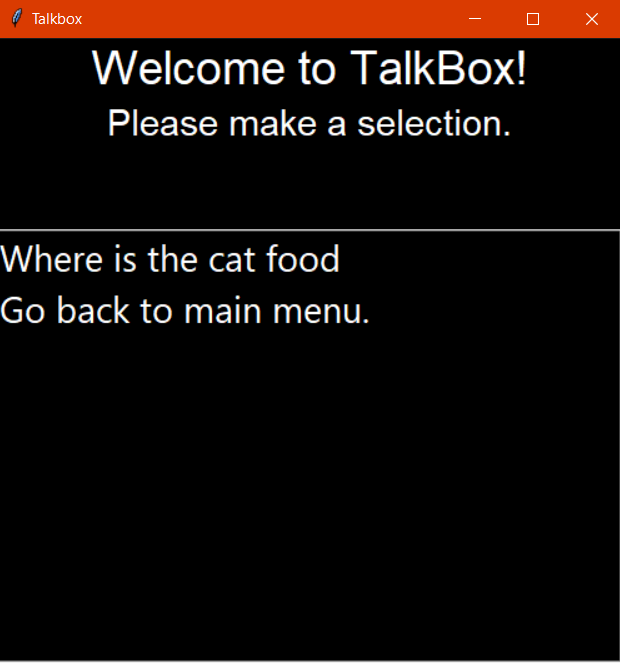


Figure 4.1.1.5 - Grocery Shopping Sub Menu

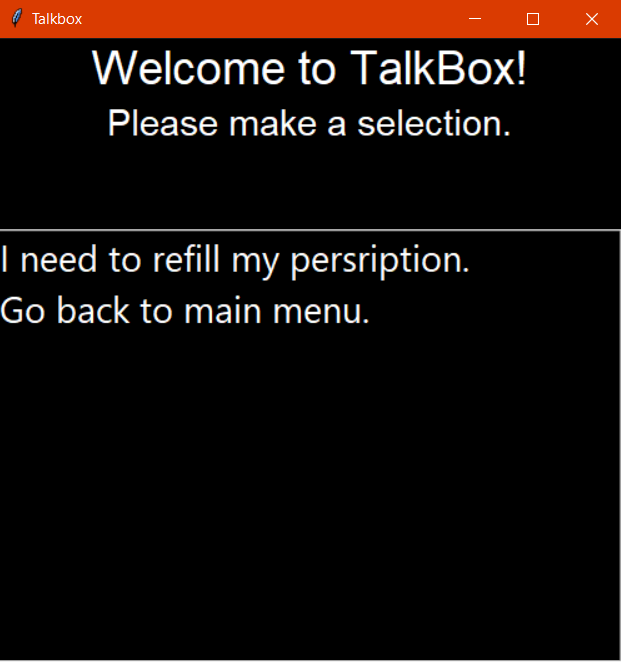


Figure 4.1.1.6 - Doctors Sub Menu

## Physical Testing

### Raspberry Pi Operating System

The first step in the physical testing was making sure all of the components worked together, and that the system turned on after installing all of the appropriate drivers and the operating system. This took multiple attempts due to missing drivers for the LCD screen and the need for a monitor to begin to program the raspberry pi.



Figure 4.2.1.1 Raspberry Pi with OS and Screen On

### Software Test on the Hardware

The software test on the assembled hardware was very buggy. There were multiple imports that needed to be done, a reworking of the code that provides the location of the excel file, and voice alterations to make the voice seem more human. The importing of the various installs proved very complicated an elusive as multiple exceptions were brought around in operating in Linux versus windows. This took lots of time of researching and installing and uninstalling different modules and packages. The voice sounds a bit too robotic, but Linux only runs espeak for the pyttsx3 module that is being used as the text to speech function. In the future, work arounds will be explored to make the voice seem more human. The placement of the online widgets will also need to be moved around for better visibility and accessibility while using the product.

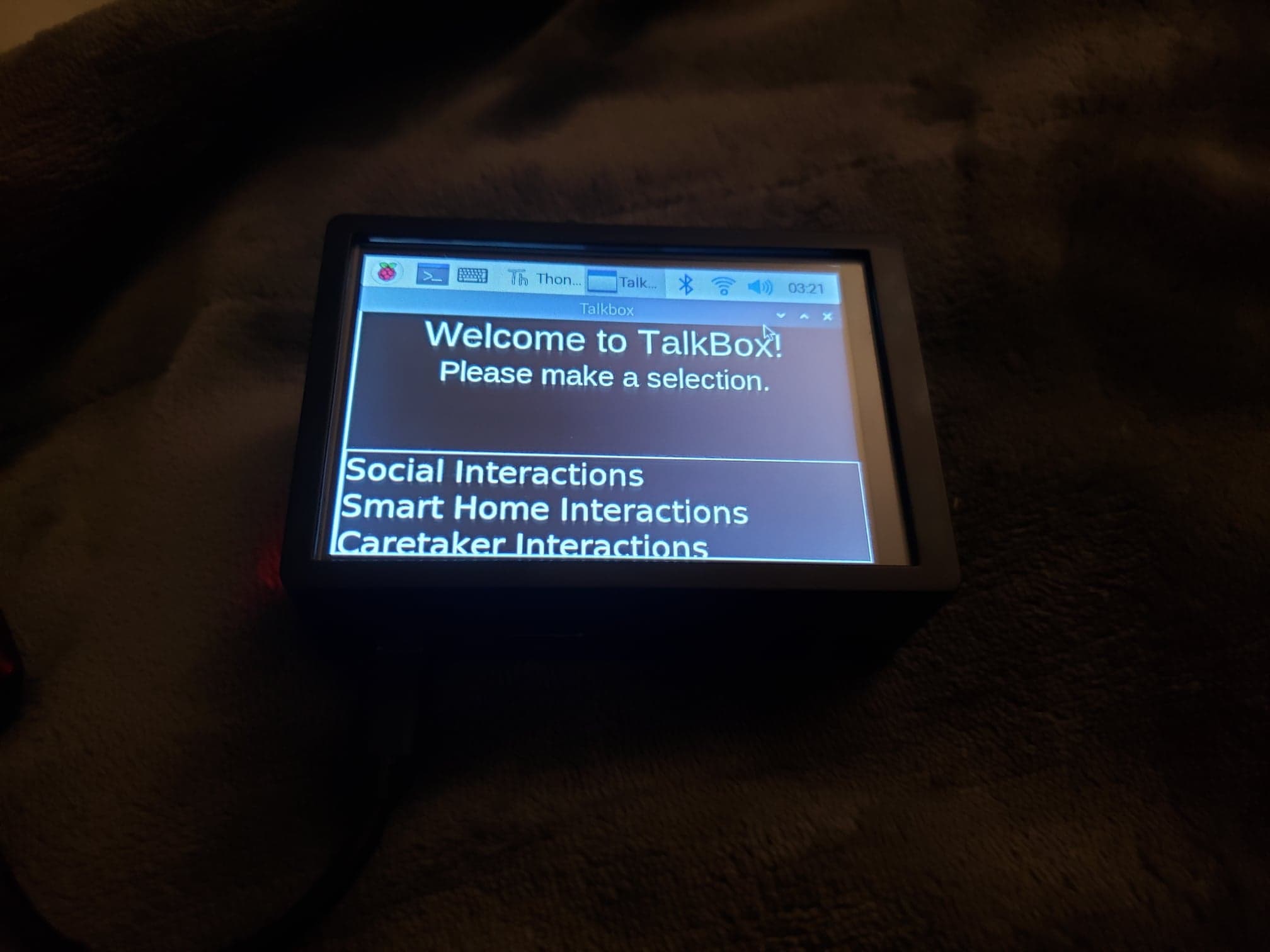


Figure 4.2.2.1 - Software Test With Raspberry Pi and Python Code



Figure 4.2.2.2 Phrases Menu on the Assembled Hardware

# Conclusions and Recommendations for Future Work

In conclusion, by creating and following a thorough a rapid iterative prototype plan, both software and physical prototypes were created. The conceived final prototype as outlined in this document, is a high-fidelity comprehensive prototype of an application that fulfills the needs of the clients, Patrick, and Roy. The implementation of feedback received from the first prototype resulted in the development of the second and final prototype which exceeds the targeted metrics in ease of navigation and user experience.

In the future, the joystick with be created and the joystick programming will be added to the final program. The team will also refine the enclosure of the physical components and alter the aesthetics of both the software and physical hardware.

# Bibliography

Man, Average, et al. “Using A Joystick On The Raspberry Pi Using An MCP3008.” *Raspberry Pi Spy*, 25 Feb. 2016, [www.raspberrypi-spy.co.uk/2014/04/using-a-joystick-on-the-raspberry-pi-using-an-mcp3008/](http://www.raspberrypi-spy.co.uk/2014/04/using-a-joystick-on-the-raspberry-pi-using-an-mcp3008/).

APPENDICES

APPENDIX I: Project Management

Here is the snapshot of the Gannt Chart.

<https://www.wrike.com/frontend/ganttchart/index.html?snapshotId=c6R3Q39PywG1fFWp8bk0aW8R8aR5jQSC%7CIE2DGNBUHAYDALSTGE3A>