GNG2101 Final Report

[VOICE CONTROL REMOTE]

Group C1A
Submitted By: Nicolas Salcedo-7193645
Juan Sebastian Guzman- 7615324
Bruce(Sike) Yin-8658847

Table of Figures

Table of Contents

Abstract

Throughout the Semester, we worked with our client to device, develop and design a solution for them. This project focuses on the social benefit of developing a vocal methodology of controlling devices rather than tactile interaction. Our objective is to design a internet-independent, voice and language barrier-free voice remote that will allow hospital patients with limited motors skills to operate TV and other devices using their voice.

After empathizing with our client, we began our ideation phases, in which we devised different design concepts of what our voice remote should be based on our client needs. From there, we began the development of our first prototype.

Eventually, we moved to a testing phase in which along with beneficial customer feedback we gained via constant demonstrations of our prototypes. Finally, after the testing phase was completed, we were able to improve our product design with client feedback to ensure that it follows the needs of our client.

Introduction

Patients and elders with limited motor skills have a difficult time performing actions throughout their daily lives. They cannot operate a TV because they have difficulty or do not have the ability of pressing the buttons on the tv remote controller. It is very stressful for them to not being able to enjoy watching TV due to these limitations. Our client Bocar Ndiaye, is an assistive technologist at the St Vincent's Hospital in Ottawa. He requested to build a voice remote controller which will allow the patients to control the TV using voice commands without the use of internet.

Bocar told us that he has tried using existing devices such as google Home and Alexa, but the issue is that both devices have trouble recognizing the patients voice due to their low tone and heavy accent. Thus, he requested that the voice remote would be able to pick up the patient's voice accent-free and accurately. This will give the hospital patients the opportunity to interact with the TV without worrying about their voice not being recognized. Lastly, Bocar also requested a voice remote that is affordable for all the patients. This is since many existing products are currently too expensive for the users.

With our design we aimed to satisfy all the customer requirements and needs. Thus, extended features were considered. An important feature that makes our product different than existing products is that it can control other devices other than just a TV. Which it will make it easier for the user to control other personal devices with it including those that relate to the tv.

Many of the patients do not want to use a device that is very expensive and that requires the use of internet so we made our voice remote operate using infrared-signals so that it can also operate different TV models. The following report entails how we used the design process to implement our voice remote controller.

Need Identification & Product Specification Process

Needs Identification & Problem Statement

All the Client Meeting information obtained was compiled into the following prioritized need statements. These need statements were used to device the problem statement of the voice remote.

a. Voice Remote allows users to control the TV using voice commands
b. Voice Remote uses speech recognition to translate the user's commands into control signals that will make the TV execute the desired task.
c. Voice Remote controls devices without using an Internet connection.
d. Voice Remote offers an easy-to-use user interface.

e. Voice Remote can be programmed with custom words and sentences to execute commands.
f. Voice Remote can be easily upgraded; the user can add, edit or delete commands
g. Voice Remote is compatible with add ons and improvements.
h. Voice Remote works great with simple user interactions.
i. Voice Remote detects a wide range of voice frequencies.
j. Voice Remote controls any type of TV.
k. Voice Remote identifies a wide range of words, accents and sentences
l. Voice Remote is affordable for all patients.

"There is a need for voice TV remote control for patients with limited motor skills in the Saint Vincent Hospital. According to Bocar N'diaye who is the technologist in the hospital, patients love to watch TV because of the limitation of their activity range. Patients have hard time to press the button of the TV remote because they barely can move their fingers, and some do not have mobility of their arms. However, most of patients can speak well; therefore, building voice control remote system for TV is the best solution. The voice control system shall be less$100 and do not need internet to support the operation. Also, the voice control system needs to recognize a wide range of words, accents and sentences."

Target Specifications and Metrics

A couple of constraints that we took into our design is that the voice remote controller will have a budget of only 100 dollars for the overall prototyping. Secondly, we had to see how big the patient's room in the hospital are to see how far the patients sit when watching TV. This gave us a clear idea of the required voice recognition distance the voice remote needs to be between 5-8 meters in order detect the voice of the patient.

For the voice remote to successfully detect and recognize the patients voice, we must take into consideration different possible tone levels the patients have and the quality of the microphone. Based on a general range of human voice levels, the voice remote must be able to pick up human voice level of 50 - 80 db.[1].

For the voice remote to accurately depict the patients voice, it should have a word error rate of a maximum of 20% percent. This takes into consideration if the patients have a speech impediment which will allow us to set a certain threshold on identifying which words the patients are able to utter out. If they cannot say a word, the voice remote will have to be customized to be able to pick up basic human natural sounds such as "aaa" "eee","oooh","uuu",etc to be able to transform them into commands.

The loudness level of the patient room is also important because it also plays a factor on the voice remote being able to pick up the patient’s voice. Therefore, the voice remote will have to able to pick up the patient's voice in different background noises by adapting to it.

Lastly, the frequency and preference of the patients use of tv will give us an idea of the range of TV commands the patient will require to be able to fully operate the tv as he/she desires. Taking into consideration basic commands such as “Turn on, Turn off, channel up/down volume down” and “mute”, the average voice commands required to be implemented will be at least 6-8 commands.

Benchmarking

Before designing our voice remote, we had to explore existing voice remote products to compare their performance and aspects that can be assimilated into our product such that we will be solving the problems these products currently produce.

Google Home is a smart speaker that allows clients to use their voice commands to interact with device such as TV and any other device. If a person says, “Hi Google, turn on my TV”, the TV will be turned on. This is one of the devices that patients are using in the hospital. However, patients that have strong accent have hard time to use it. The price of Google Home is around $130-$200 and monthly payments of $20-35. Most of patients would not able to afford it.

Figure 1:Gooogle Home

Harmony Hub is a device that can be placed near TV and cable box equipment so it can control the device by using smart phone, screen or remote. However, this device involves pressing a button, which does not fully eliminate tactile interaction as the client wants it to be. Lastly The price of Harmony is also over $100, which is slightly expensive to patients.

Figure 2 Harmony Hub

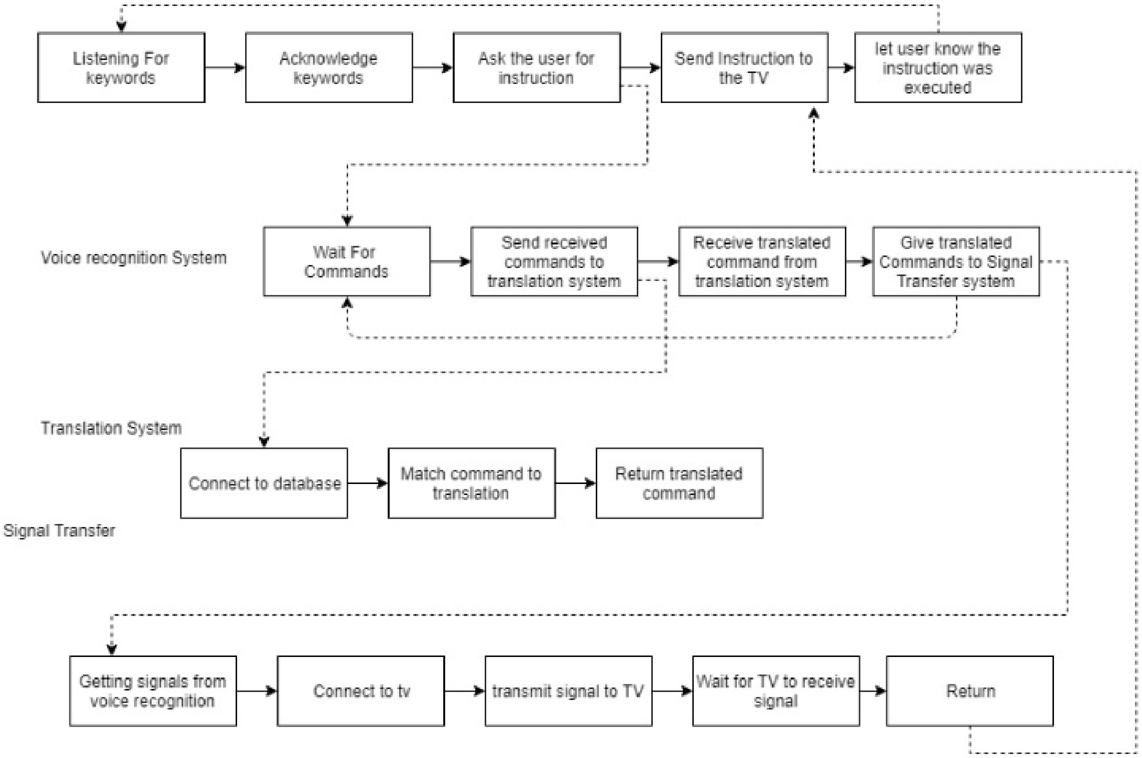
Judging on the products aspects, we can see that having internet connection is a very expensive alternative due to internet bill is considered. Both the google home and the harmony hub do not work in favor of the patients accent and voice level and are quite expensive.

Conceptual Design

After defining our problem and the expected target specifications the voice remote will carry out, we began our Ideation Phase. Where we devised the core functionality the remote will carry out for the implementation to be successful. Secondly, each of our members sketched their own design ideas of the voice remote to which as a team we decided which idea provided the best solution.

Functional Decomposition

The following Sketch shows three main important systems involved in the functionality of the voice remote. The Voice recognition system will be detecting and recognizing the user's voice, being ready to receive a command which is then send to the translation system to be mapped to its respective tv command. The translation system will see if the received command matches with any of the translations stored in the database. If the match is successful, the translated command is then sent to the tv using signal transfer and the tv perform the command received.
Otherwise. The voice remote will not perform any action until the user says a proper command.



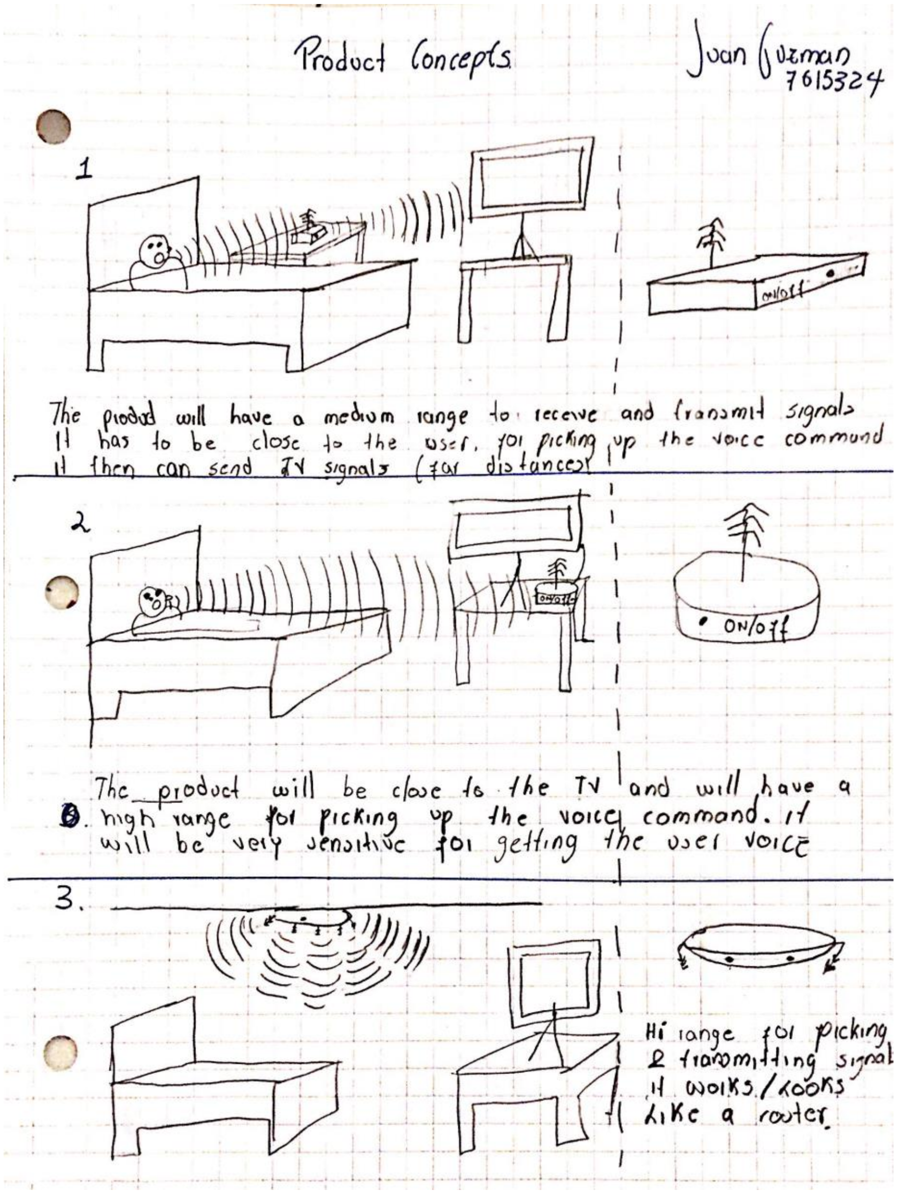
Listening For keywords
Acknowledge keywords
Ask the user for instruction
Send Instruction to the TV
let user know the instruction was executed
Voice recognition System
Wait For Commands
Send received commands to translation system
Receive translated command from translation system
Give translated Commands to Signal Transfer system
Translation System
Connect to database
Match command to translation
Return translated command
Signal Transfer
Getting signals from voice recognition
Connect to tv
transmit signal to TV
Wait for TV to receive signal
Return

Figure 3 :Voice Remote Functional Decomposition

Sketching Designs

Juan’s Designs

Juan sketches show three different designs. The first two designs consider having the voice remote being sending the voice command to the TV wirelessly with the difference being that the first design, the voice remote is being placed right next to the patient while the second design is being placed right next to the tv. The third design is that the voice remote behaving as a router that will be placed in the middle the ceiling of the patient's room. This will optimize the distance of the patient to the voice remote and the voice remote to the TV.


Product Concepts

Juan Guzman
7615324

The product will have a medium range to receive and transmit signals
It has to be close to the user, for picking up the voice command
it then can send TV signals (far distances)

The product will be close to the TV and will have a
high range for picking up the voice command. it
will be very sensitive for getting the user voice

Hi range for picking
& transmitting signal
it works/looks
like a router.

Figure 4:Juan's Sketches

Bruce's Designs

The first design consists on the voice remote being directly wired to the tv to send the received voiced commands. The voice remote will still be using a microphone to pick up the patients voice. The second design will consist of the voice remote being wirelessly connected to the TV and will be placed beside the tv.The last design consist of mounting the voice remote to the wall beside the patient bed and passing the wiring through the wall to reach the TV.

Overall, the discussion between these Ideas was that,

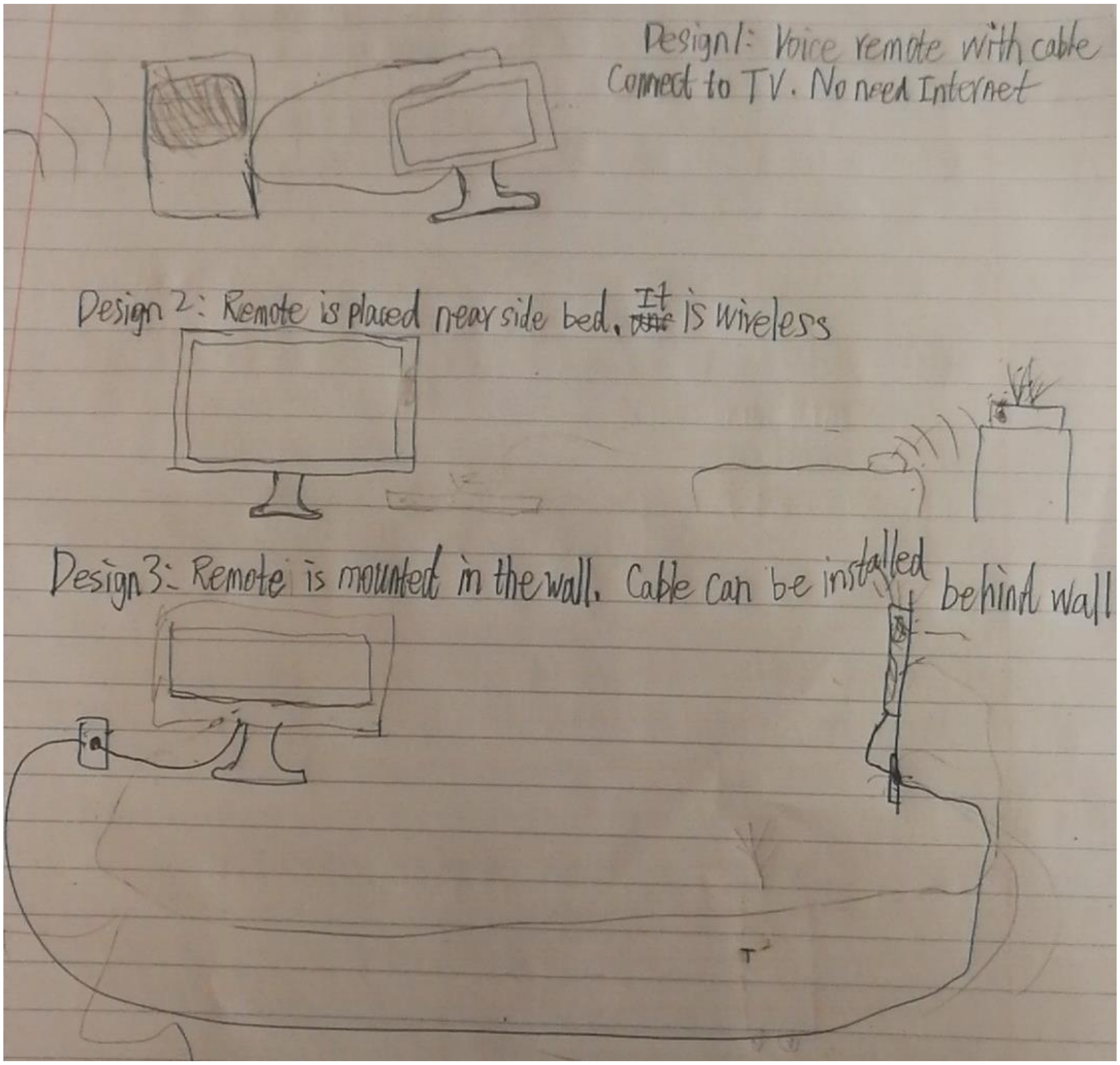
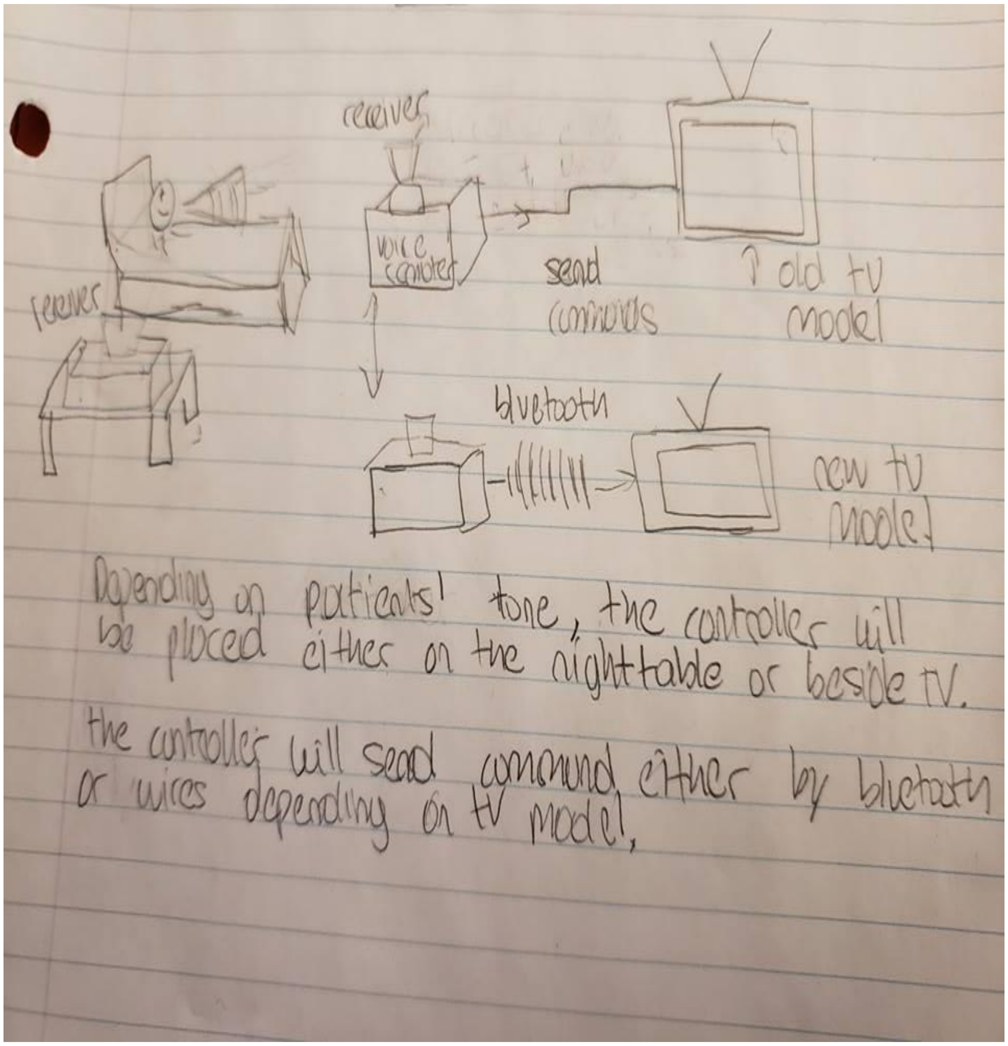

Design1: Voice remote with cable Connect to TV. No need Internet
Design 2: Remote is placed near side bed. It is wireless
Design3: Remote is mounted in the wall. Cable can be installed behind wall

Figure 5::Bruce Sketches

Nicolas's Design.

The following designs focuses on which method of signal transmission will be used based on the patient's' TV model. The First and second design are looking into placing the voice remote either closer to the patients or being placed right next to the TV depending on the patient's tone level. The difference however is that depending on the tv model, the voice remote will use HDMI wire to send the commands for modern models or will use Bluetooth for older models.



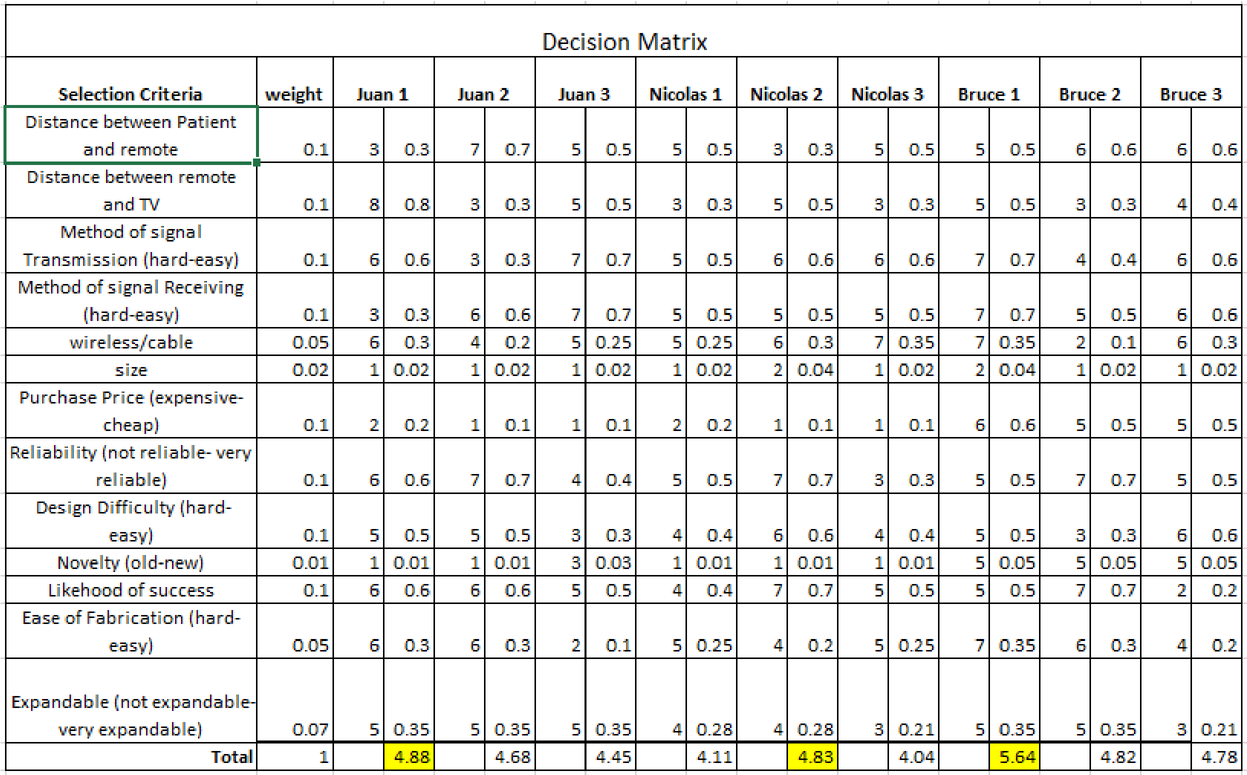
Depending on patients' tone, the controller will be placed either on the nighttable or beside tV.

the controller will send command either by bluetooth or wires depending on tv model,

Figure 6: Nicolas's Sketches

Coming to Design Decision

Decision matrices can help us select the best option from the 9 concepts that were provided by each member of the team. We selected the selection criteria of the decision matrix based on the target specifications. We added the weight factor to each criterion; the weight factor was based on our knowledge of the product, the target specifications and the suggestions made by the client in the client meeting. Figure below displays the decision matrix.



Decision Matrix

Selection Criteria	weight	Juan 1		Juan 2		Juan 3		Nicolas 1		Nicolas 2		Nicolas 3		Bruce 1		Bruce 2		Bruce 3	
Distance between Patient and remote	0.1	3	0.3	7	0.7	5	0.5	5	0.5	3	0.3	5	0.5	5	0.5	6	0.6	6	0.6
Distance between remote and TV	0.1	8	0.8	3	0.3	5	0.5	3	0.3	5	0.5	3	0.3	5	0.5	3	0.3	4	0.4
Method of signal Transmission (hard-easy)	0.1	6	0.6	3	0.3	7	0.7	5	0.5	6	0.6	6	0.6	7	0.7	4	0.4	6	0.6
Method of signal Receiving (hard-easy)	0.1	3	0.3	6	0.6	7	0.7	5	0.5	5	0.5	5	0.5	7	0.7	5	0.5	6	0.6
wireless/cable	0.05	6	0.3	4	0.2	5	0.25	5	0.25	6	0.3	7	0.35	7	0.35	2	0.1	6	0.3
size	0.02	1	0.02	1	0.02	1	0.02	1	0.02	2	0.04	1	0.02	2	0.04	1	0.02	1	0.02
Purchase Price (expensive-cheap)	0.1	2	0.2	1	0.1	1	0.1	2	0.2	1	0.1	1	0.1	6	0.6	5	0.5	5	0.5
Reliability (not reliable- very reliable)	0.1	6	0.6	7	0.7	4	0.4	5	0.5	7	0.7	3	0.3	5	0.5	7	0.7	5	0.5
Design Difficulty (hard-easy)	0.1	5	0.5	5	0.5	3	0.3	4	0.4	6	0.6	4	0.4	5	0.5	3	0.3	6	0.6
Novelty (old-new)	0.01	1	0.01	1	0.01	3	0.03	1	0.01	1	0.01	1	0.01	5	0.05	5	0.05	5	0.05
Likehood of success	0.1	6	0.6	6	0.6	5	0.5	4	0.4	7	0.7	5	0.5	5	0.5	7	0.7	2	0.2
Ease of Fabrication (hard-easy)	0.05	6	0.3	6	0.3	2	0.1	5	0.25	4	0.2	5	0.25	7	0.35	6	0.3	4	0.2
Expandable (not expandable-very expandable)	0.07	5	0.35	5	0.35	5	0.35	4	0.28	4	0.28	3	0.21	5	0.35	5	0.35	3	0.21
Total	1		4.88		4.68		4.45		4.11		4.83		4.04		5.64		4.82		4.78

Figure 7:Decision Matrix

According to the decision matrix we have chosen concept 1 by Juan and concept 1 by Bruce as very promising solutions that we wish to develop.

The concept 1 by Juan describes a product that is not connected to the TV and that uses a microphone to get the voice commands from the user, then will translate the commands and transfer them via infrared to control the TV. Similarly, the concept 1 by Bruce uses a microphone to listen for the commands from the user but in this design the product is connected directly to the TV and will control the TV using the HDMI cable.

These solutions not only got the highest score in the decision matrix, but they scored very high on criteria that we consider critical, for instance; both concepts scored very high on the reliability and likelihood to success criteria. In our opinion those two criteria are very important because we want to build a product that is very consistent, dependable and we want to make sure that we can have a finished product that will succeed.

We chose the selection criteria based on the target specifications and the decision matrix has suggested the two concepts described before, we believe that they are options that will fulfil the client needs and we will be able to have either concept finished by the end of the semester.

Final Concept

The final concept that came out from the integration of the two concepts that scored the highest in the decision matrix will accomplish most of the target specifications. There are some specifications that we will be able to accomplish once we start building the device and start testing it. The final concept allows the user to control the TV using voice commands with no need of an internet connection, it controls any type of TV, it detects a wide range of voices and commands, it can be programmed with custom words and it is affordable.

The benefit of our final concept is that it can either send IR signals to control the TV wirelessly or it can be connected directly to the TV to control it. These two functionalities make the device not dependable on the internet. It also uses a microphone to capture the voice commands, which assures that the device will be able to listen to low frequency commands (user that speaks very low).

We believe that there are no drawbacks from our final concept, but we believe that other concepts could have been more versatile and better looking, but they were either not affordable, very hard to implement and very rare to succeed.

Group Design

We have decided to integrate both concepts, we will aim to build a product that will integrate both functionalities: first, being able to receive and send signals wirelessly and second, receiving signals wirelessly and transferring these signals to the TV through the HDMI cable.
So, we want to build a product that can execute the task either wirelessly or by being connected to the TV. We believe that this product will integrate all the needs of the client and will offer the client many types of advantages and convenience.

In conclusion we will build a product that will be always listening for voice commands from the user, once the commands are taken and translated the device can control the TV by sending control signals by either infrared methods or by being directly connected to the TV. The device can be placed next to the patient or can be left besides the TV.

The following sketch shows that the voice controller will be plugged to the TV at a medium-close distance to the Patient bed. The patient will be speaking and the voice controller’s antenna will pick up and recognize his/her voice to be processed into a command. This command signal will go the TV and will trigger an action based on the command.

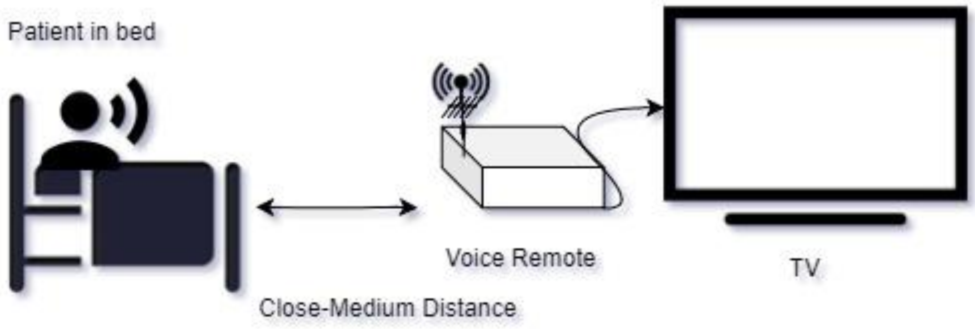


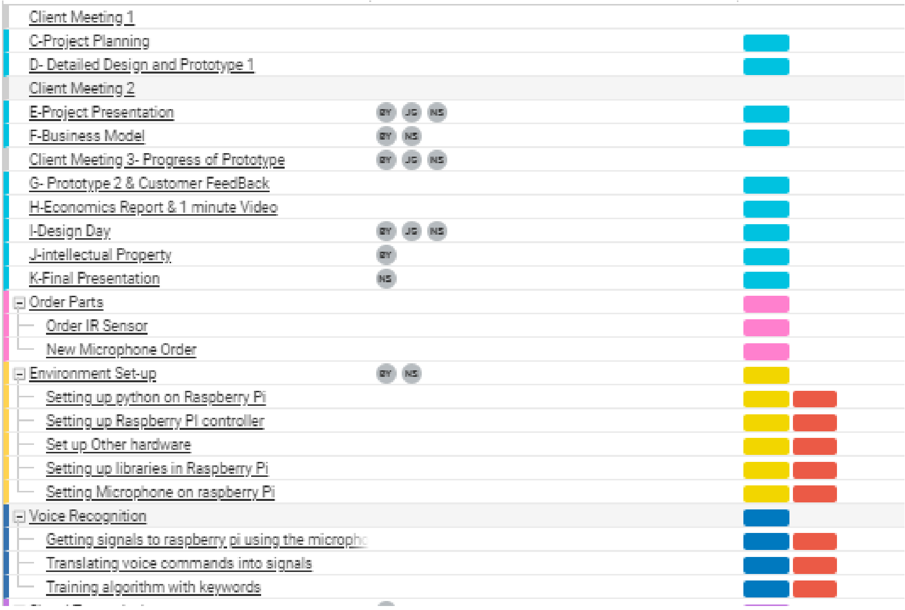
Figure 8:Group concept Diagram.

Problems Encountered Prior to prototyping

HDMI cable connection was taken into consideration to send tv commands to the TV. However, even with the TV having an HDMI port, it is still required to have the HDMI-CEC protocol built-in for it to be able to receive the incoming commands-turned signals from the voice remote. Since not many TV models in the hospital are very modern, the voice remote would yield a very limited connectivity. Therefore, the method of communication between the voice remote and the tv was changed to using only Infrared signals. Which will provide universal connection with any TV model.

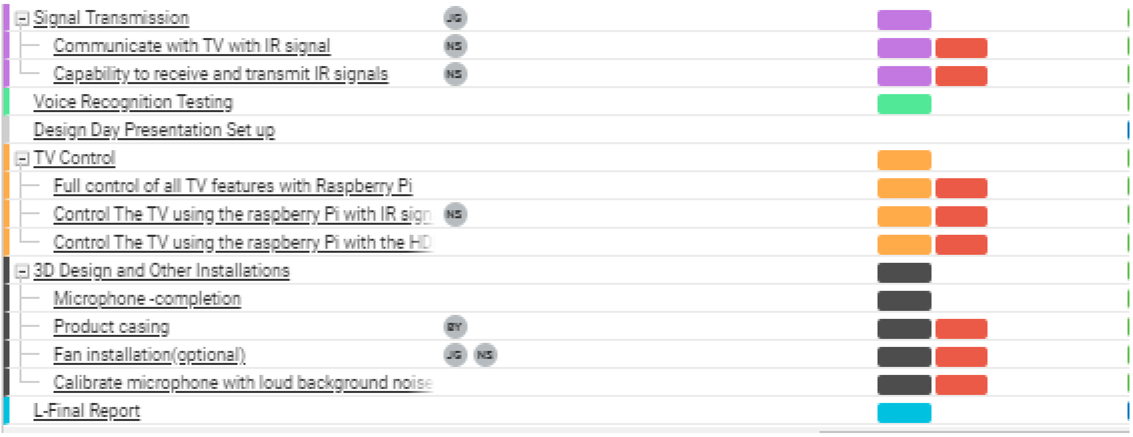
Project Planning and Feasibility Study

The following two pictures shows that our Project planning. We listed 5 major tasks and arranged their deadlines. Every major task represents a kind of color. We also came up with bunch of sub-tasks and their deadlines. Each sub-task has one or more relative major tasks. This way of planning is beneficial for us to maintain the track of time and planning management.



Client Meeting 1	
C-Project Planning	
D- Detailed Design and Prototype 1	
Client Meeting 2	
E-Project Presentation	BY JG NS
F-Business Model	BY NS
Client Meeting 3- Progress of Prototype	BY JG NS
G- Prototype 2 & Customer FeedBack	
H-Economics Report & 1 minute Video	
I-Design Day	BY JG NS
J-intellectual Property	BY
K-Final Presentation	NS
⊟ Order Parts	
Order IR Sensor	
New Microphone Order	
⊟ Environment Set-up	BY NS
Setting up python on Raspberry Pi	
Setting up Raspberry PI controller	
Set up Other hardware	
Setting up libraries in Raspberry Pi	
Setting Microphone on raspberry Pi	
⊟ Voice Recognition	
Getting signals to raspberry pi using the microph	
Translating voice commands into signals	
Training algorithm with keywords	

Figure 9:Project Planning 1



⊟ Signal Transmission	JG
Communicate with TV with IR signal	NS
Capability to receive and transmit IR signals	NS
Voice Recognition Testing	
Design Day Presentation Set up	
⊟ TV Control	
Full control of all TV features with Raspberry Pi	
Control The TV using the raspberry Pi with IR sign	NS
Control The TV using the raspberry Pi with the HD	
⊟ 3D Design and Other Installations	
Microphone -completion	
Product casing	BY
Fan installation(optional)	JG NS
Calibrate microphone with loud background noise	
L-Final Report	

Figure 10:Project Planning 2

The following shows the Gantt Chart of when our team will start and how long our team plans to spend for each of the design progress steps. The First sequence (colored blue represents our deliverables present during our project process while the others represent our critical path sequence on steps that needed to be done in order to successfully reach a final prototype. First, we will be ordering the parts required for our prototype. We will then step into setting the environment and doing the voice recognition. Then, we will be doing signal Transmission and TV control. Finally, we will be testing the prototype to ensure it meets target specifications.

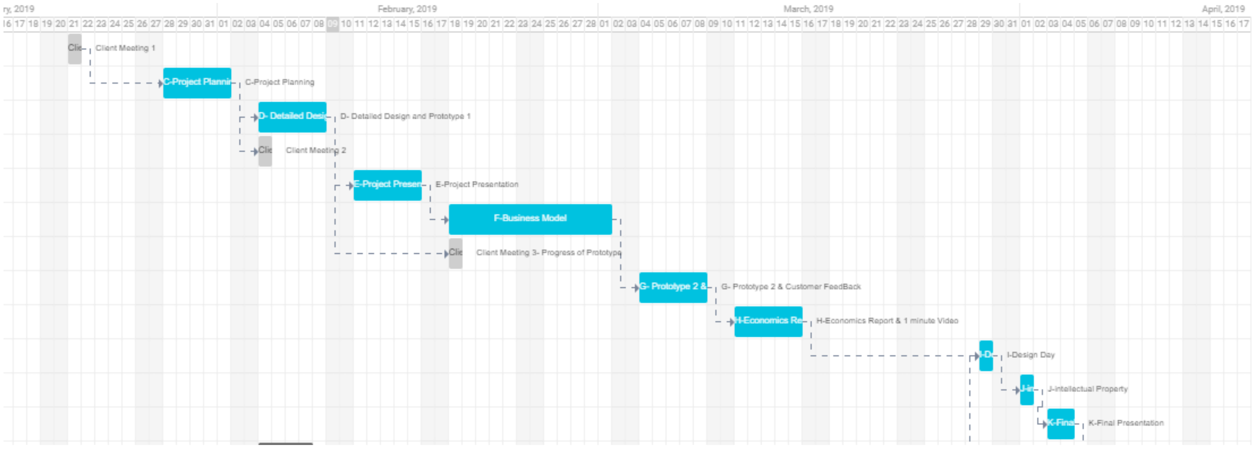


Figure 11:Gantt Chart (a)

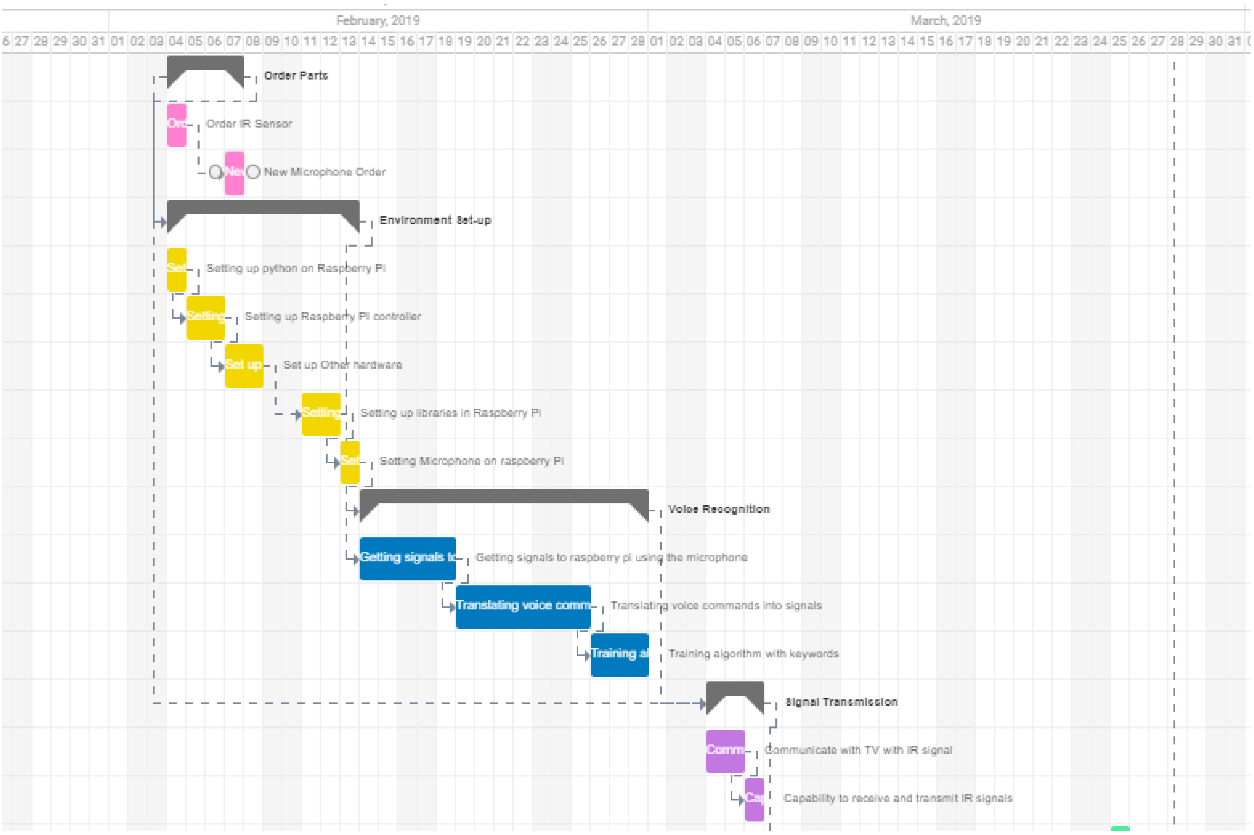


Figure 12:Gantt Chart (b)

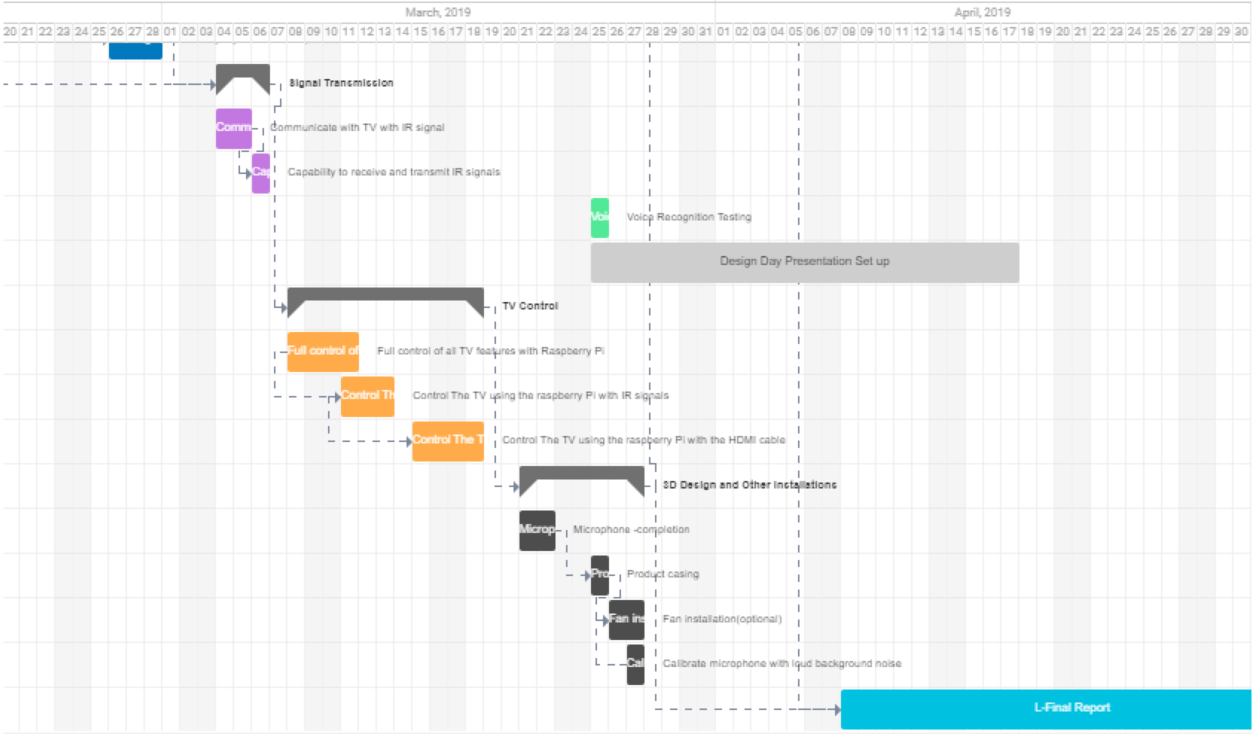


Figure 13:Gantt Chart(c)

Bill of Materials (BOM)

The following bill of Materials Table entails all material that were bought to be able to design our voice remote. During the beginning of the design process, we went for cheaper alternatives(IR LED and USB microphone when building our first prototypes. As the design progressed, we bought components with same functionality but better performance as previously bought.eg (IR LED circuit replaced with IR transceiver).As a result, we exceeded the budget required by about 15 dollars.

Item Number	Part Name	Description	Quantity	Unit Cost	Extended Cost
1	Raspberry Pi 3	Main part of voice control remote	1	$44.88	$44.88
2	HDMI Cable	Cable that is used to connect TV	1	$11	$11
3	Wires	Make a Circuit	10	$0.2	$10
4	Microphone	Receive Voice	1	$2	$2

5	330 ohm resistors	Resistors that will be used to maintain proper current	5	$0.05	$0.25
6	NPN Transistor	Transmit signal to Raspberry pi	4	$0.96	$3.84
7	Breadboard	A board that is used to build circuit	1	$9.38	$9.38
8	IR LED	Transmit signal to TV	2	$0.41	0.82
9	IR Transceiver	Second Alternative to transmit signal TO TV	1	$12.39	$12.39
10	Professional Cardioid Condenser Microphone With Tripod Stand	To speak to send voice to voice remote for command mapping	1	$24.95	$24.95
				Total	$119.51

Feasibility Study

Technical: Does your team have enough expertise and technical resources?
In order to successfully design the voice controller, it has to be taking into account the technical aspects required to build it. Therefore, for voice recognition, signal processing, integrated controllers and computer programing expertise is required. Fortunately, a raspberry pi microcontroller is an excellent resource as it allows us as a team to work with received signal by writing and running computer programs in it.
Every team member has experience writing computer programs. Therefore, voice recognition algorithm will be challenging task. we have two team members in 5th electrical engineering and they have adept experience on handling circuitry or working with signal processing. Therefore, integrating every component together with any external devices will not a difficult task on the electrical aspect.

Economic: Can the cost of your project be reasonable?
Most of the voice controller voice recognition and signal transmittal is covered by the raspberry pi microcontroller due to the fact it has a built-in receiver/transmitter and programming scripts can be uploaded in it. Therefore, spending about $50 on the raspberry pi is very efficient and fair. Secondly, sending the processed command signal to the tv through wire, it is logical that only an

HDMI wire is required, thus $12 dollars is a fair price. Lastly, external circuitry and the use of microphone will be taken into consideration to optimize the accuracy of voice recognition, therefore $15 is a fair price since the microphone, breadboard, circuit wires and other electrical components are very cheap.

Legal: Are there any legal issues with releasing your solution to the public

The Voice controller will only send commands to the TV when the patient says a specific instruction. It will not pick up any other information. Thus, the confidentiality policy in the hospital will be respected. Thus, there will not be any legal issues on releasing our design to the public.

Operational: Are there any organizational constraints that will prevent your success

The distance between the patient bed and the TV is something important to be taken into consideration. Depending on where the TV was installed in the patient room, if it is already placed at a medium-far distance from the patient's bed, the voice remote could have inaccuracies in picking up the patient's voice. Also, the voice control remote is not depended on using internet; therefore, there exists problems with system updating. Picking up voice with accent is also a constraint because accent is not normally can be picked up by voice recognition system.

Scheduling: What are the deadlines and are they reasonable for your solution

There are few deadlines:

1. Detailed Design and Prototype 1, Feb 10
2. Environment set-up & Voice Recognition Feb 27
3. Signal Transmission & TV control March 1
4. Prototype 2 & Customer Feedback, March 10
5. Design and other installations: March 17
6. Design Day, March 29

Analysis, Prototyping, Testing and Customer Validation.

In this section we will show all the prototypes that we built, their evolution, the tests performed on each prototype and the validation of each design. This process directed us to our final product.

First Prototype

The first prototype consisted of a very cheap USB microphone, SOPARE software for the voice recognition system and transceiver/receiver mounted on a breadboard.
Figure 9- displays our first prototype.

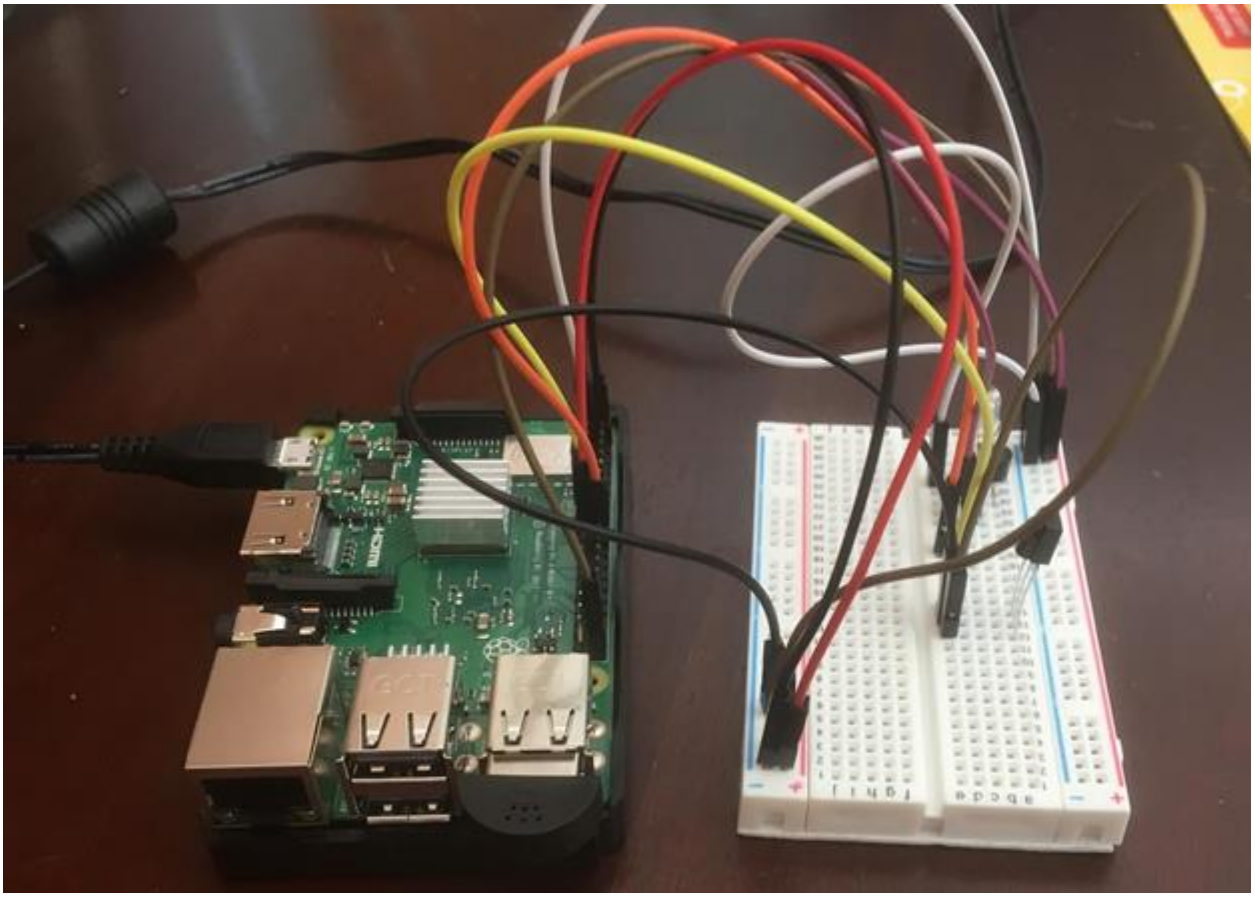


Figure 14:First Prototype

For our first prototype we wanted to make sure that we were able to train some word commands, once those commands had been trained then we would say those commands and our prototype was able to recognize those commands and send the correct signal to the TV.
Basically, we wanted to make sure that we were able to say a command word and the prototype would control the tv accordingly; for instance, saying the word "Power" would turn the TV off or On.
The tests that we performed for this first prototype was 5 sets of words (Power, Volume Up, Volume Down, Mute, Source and Exit) and we were controlling a RCA TV
We were able to get the IR codes from the TV using the LIRC module for the transceiver hardware.
Table 1 shows the results obtained from the tests performed.

Target specification	**Expected Result**	**Actual Result**
Word Distinction	5 words out of 5 words	3 words out of 5 words
Word recognition	100%	75%
IR Transmission distance	3m	3 m
IR Receiver commands	Receiving all commands from remote control	Receiving all commands from remote control
Voice recognition with quiet environment	100%	90%
Voice recognition with noisy environment	100%	30%

Table 1: Testing Results for First Prototype.

From the table above, we can see that the results obtained for our first prototype were not very satisfactory, however we had things that were working very well like the IR Receiver and transmitter but things that were working very poorly like the voice recognition in noisy environment and the word distinction.

For the next prototype we wanted to make sure to improve the voice recognition and increase the IR transmitting distance.

Second Prototype

As stated before, our first prototype was working poorly with the voice recognition in noisy environment and the word recognition and distinction were not completely satisfactory, so we decided to invest in a better microphone to improve those aspects. We decided to get a microphone with better fidelity and a wider range of pick up frequencies. The microphone that we selected took around 1 week and a half to arrive. We did not want to wait for that long to work on our second prototype, so we decided to work on our IR system and improve it the best we could.

For our second prototype we decided to get an IR transceiver module and integrated to the raspberry pi. Figure 10 displays our second prototype.

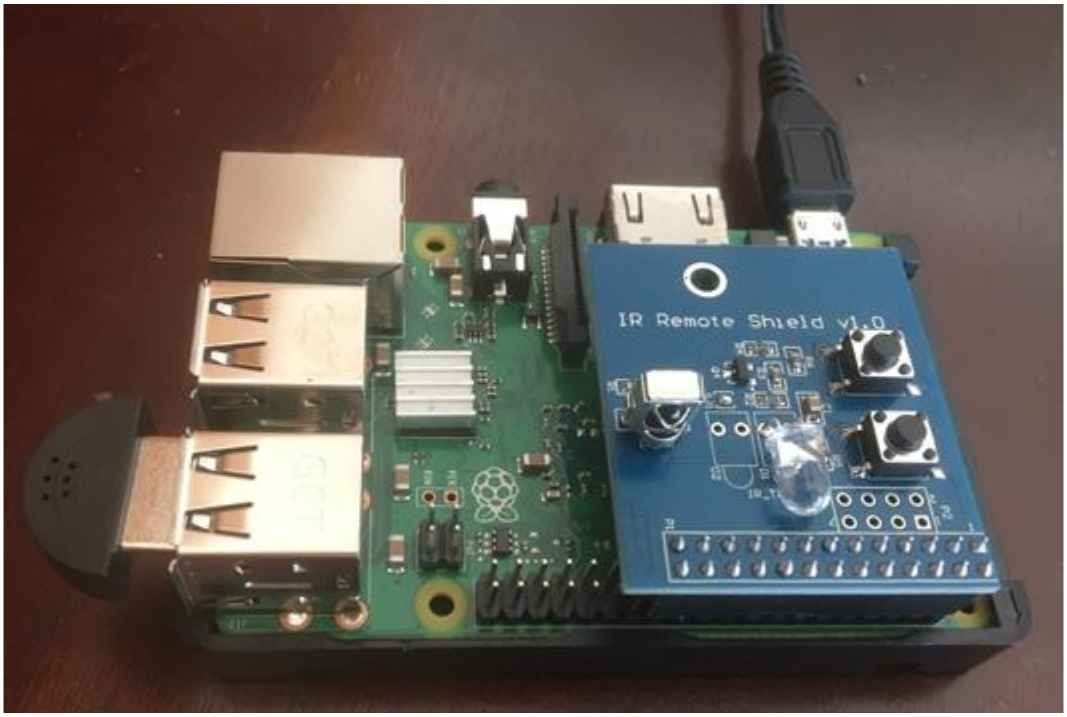


Figure 15:Second Prototype.

As you can see from the figure above, the IR Remote shield for the transceiver module was much smaller and much simpler than the one we had built in the breadboard for our first prototype. We run some tests for this prototype, the tests were more related to the IR receiver and transmitter rather than voice recognition.

Table 2 displays the results from the tests

Target specification	**Expected Result**	**Actual Result**
Word Distinction	5 words out of 5 words	3 words out of 5 words
Word recognition	100%	75%
IR Transmission distance	8m	8m
IR Receiver commands	Receiving all commands from remote control	Receiving all commands from remote control

IR Transmitting	100%	100%
Number of Electronic devices that can be controlled	4 Devices	4 Devices
Voice recognition with quiet environment	100%	90%
Voice recognition with noisy environment	100%	30%

Table 2:Test results for Second Prototype.

From the table above we are able to see that our IR system improved very much (we can now transmit signals from 8 meters of distance) and the IR remote shield allowed us to control more than one electronic device, so thanks to the IR remote shield with our second prototype we were able to control not only one TV but multiple TVs or Light systems, sound systems any electronic device that has a remote controller could be controlled by our IR remote shield. This was a big improvement and a new feature for our product.

We were very satisfied with our IR module but still we needed to improve our voice recognition system.

Third Prototype

At this point we were very satisfied with our IR transceiver module, it was working perfectly, and it had given us a new feature for our product, the ability to control more than 1 electronic device.

The voice recognition system was not very satisfactory and around this time we were able to get the microphone that we had ordered. We integrated this better microphone to our product and execute some test.

Figure 11 displays our third prototype.

From the figure you can see that we had a much better USB microphone, the IR Transceiver module and a 3D printed casing for the Raspberry pi.

As stated before the main goal that we wanted to achieve with this third prototype was to improve the voice recognition system by being able to identify each word from all the words in the training set, improve precision and accuracy of word detection in quiet and very noisy environments.

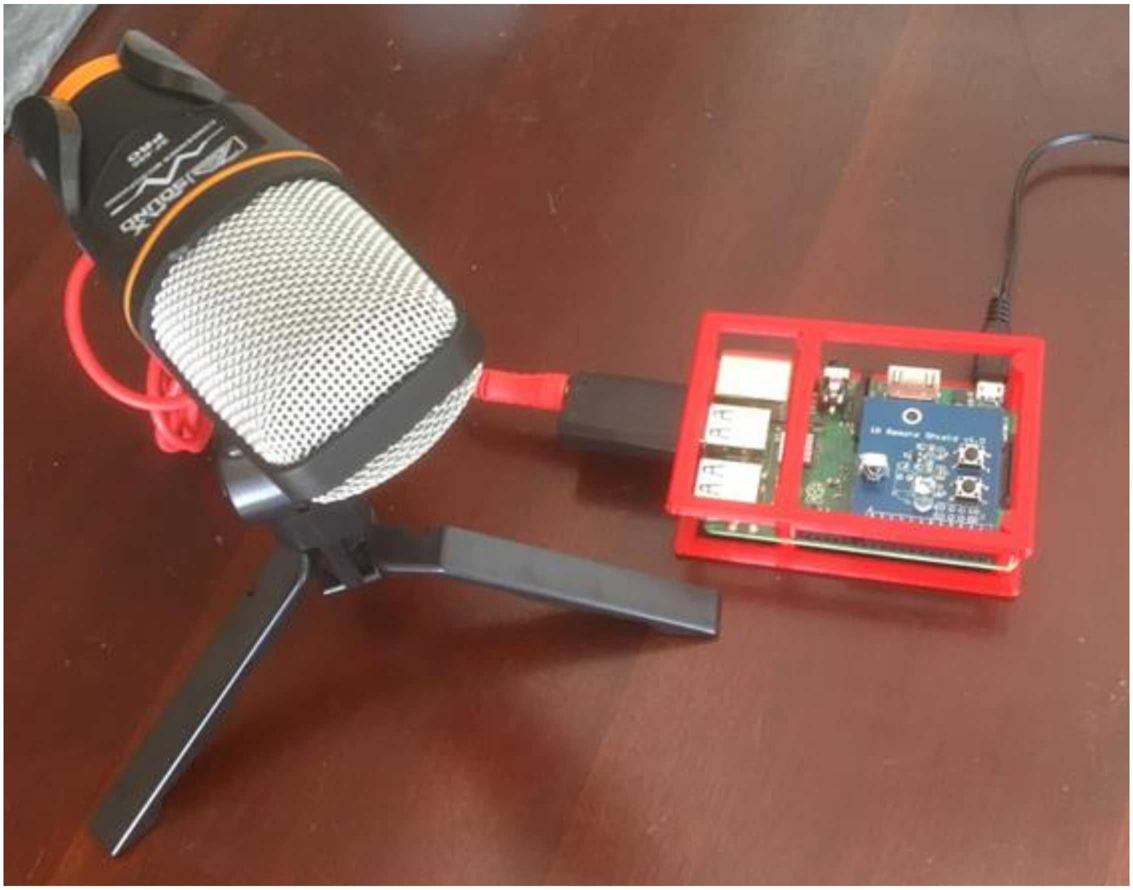


Figure 16:Third Prototype

Table 3 displays the results from the tests performed on the third prototype.

Target specification	**Expected Result**	**Actual Result**
Word Distinction	5 words out of 5 words	5 words out of 5 words
Word recognition	100%	100%
IR Transmission distance	8m	8m
IR Receiver commands	Receiving all commands from remote control	Receiving all commands from remote control
IR Transmitting	100%	100%

Number of Electronic devices that can be controlled	4 Devices	4 Devices
Voice recognition with quiet environment	100%	100%
Voice recognition with noisy environment	100%	75%

Table 3:Test results for prototype 3

As it can be seen in the table above the choice of microphone really improved the performance of the voice recognition system, this prototype was able to distinguish every word from a set of 5 words and the word recognition was 100% in a quiet environment. The microphone really helped us improved the performance of our product.

At this point we were very satisfied with our product, however we wanted to make sure that our product was able to recognize a bigger set of words, also we wanted to improve the voice recognition in a noisy environment. We wanted our final product to have those features working in a very satisfactory standard.

So, for our final product we needed to improve our voice recognition in a noisy environment, which had to be done with adjusting the SOPARE software and making a perfect work when training the words. For our IR module we were very satisfied, so we decided to leave that module as it was for our final product. Finally, we wanted to improve our casing for the raspberry pi and include a fan to cool down the controller of the raspberry pi, to make sure it would never shut down due to temperatures issues.

Our goal was to improve the described issues for our final product.

Final Solution

From our prototyping we were very satisfied with our IR module, so we decided to leave it as it was for our final product. The voice recognition system needed a better performance when performing in a noisy environment as well as increasing the number of words that the voice recognition system was able to detect. And our product needed a better and more attractive casing. These aspects were our main goal to improve for our final product.

In order to fix the voice recognition system, we became experts with the SOPARE software and its configurations; we found out that for better performance the command words have to be set in the environment that the product will be performing; for instance if the product is going to be used on a place that has a lot of noise and sounds (a house close to a construction site, or a common room with lots of TVs and many people talking) the words should be tested and trained in that environment.

We adjusted the configurations of the SOPARE software and trained the words on a very loud and noisy environment to test our product performance.

We were also to increase the amount of recognizable words from a set of 5 to a set of 15 different words which was a very big improvement. Generally, to fully control a TV you will need no more than 10 commands, so being able to differentiate between a set of 15 words was a very good feature for our final product.

We decided to create an MDF casing for our product with open compartments for the product to be able to transmit the IR signal and to prevent the raspberry pi to get hot. We also included a fan to keep the controller of the raspberry pi always cool down.

The final product consists of a raspberry pi, an IR transceiver module, an USB microphone, a fan and Power cable. These elements integrated together make our product able to recognize trained words to control any type of electronic device through IR signaling.

Figure 11 and 12 display our final product.

Final Product Features

1. Customize and train any type of word or sound to control electronic devices with your voice.
2. User can create up to 15 trained words to control a device
3. Command words can be in any language or with any type of accent.
4. No need of an internet connection to operate the product.
5. Excellent performance in quiet and noisy environments.
6. It can be adapted to any type of user.
7. Controls up to 4 different electronic devices.
8. Fully customizable.
9. Transmits and receives IR signal up to 8 meters of distance.
10. Receives signal from any type of remote controller
11. Easy to install and use.
12. It can be fully expandable for user entertainment or learning.

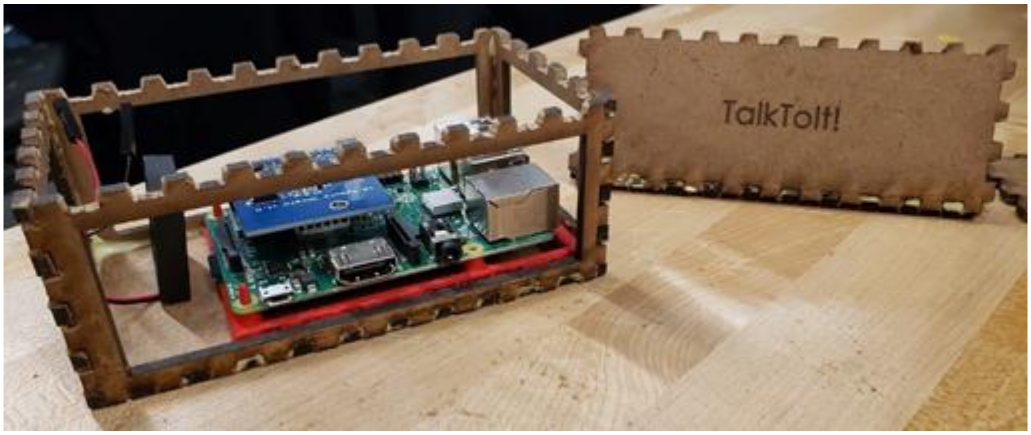


Figure 17:Final Product(a)

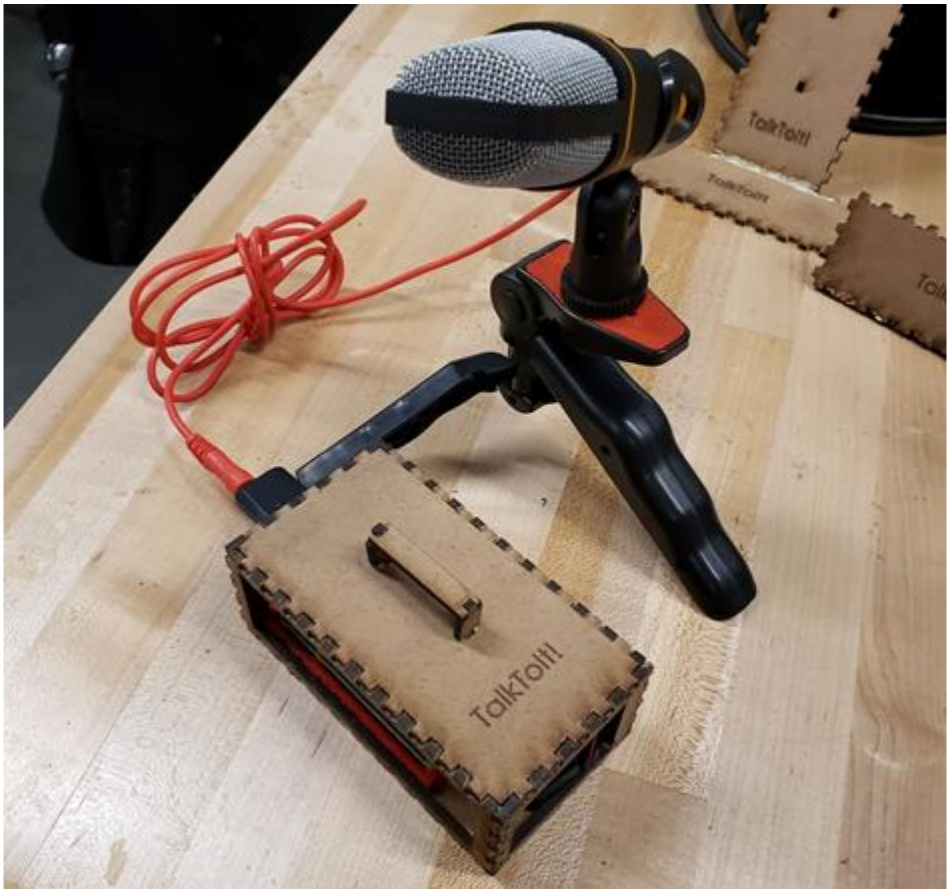


Figure 18:Final Product(b)

Table 4 displays the final testing result for our product.

Target specification	Expected Result	Actual Result

Word Distinction	15 words out of 15 words	15 words out of 15 words
Word recognition	100%	100%
IR Transmission distance	8m	8m
IR Receiver commands	Receiving all commands from remote control	Receiving all commands from remote control
IR Transmitting	100%	100%
Number of Electronic devices that can be controlled	4 Devices	4 Devices
Voice recognition with quiet environment	100%	100%
Voice recognition with noisy environment	100%	85%

Table 4: Test results for final product.

.

Business Model Canvas

Since the Voice Remote project began with gathering customer needs in a meeting with the client and then were used to build the system specifications of our voice remote prototype. It makes sense that the type of business model suited for commercializing the voice remote is an Empathic design. This is because with the customer needs, we are now working towards on designing a voice remote the can perform only its important functionality; thus, the MVP is only required. Afterwards, we will be constantly improving our voice remote design in a iterative manner.

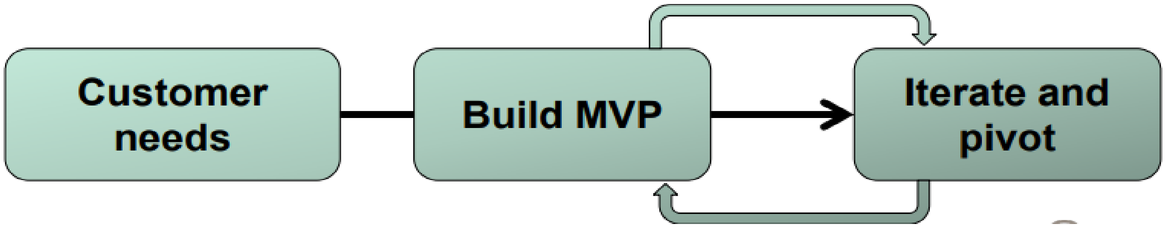


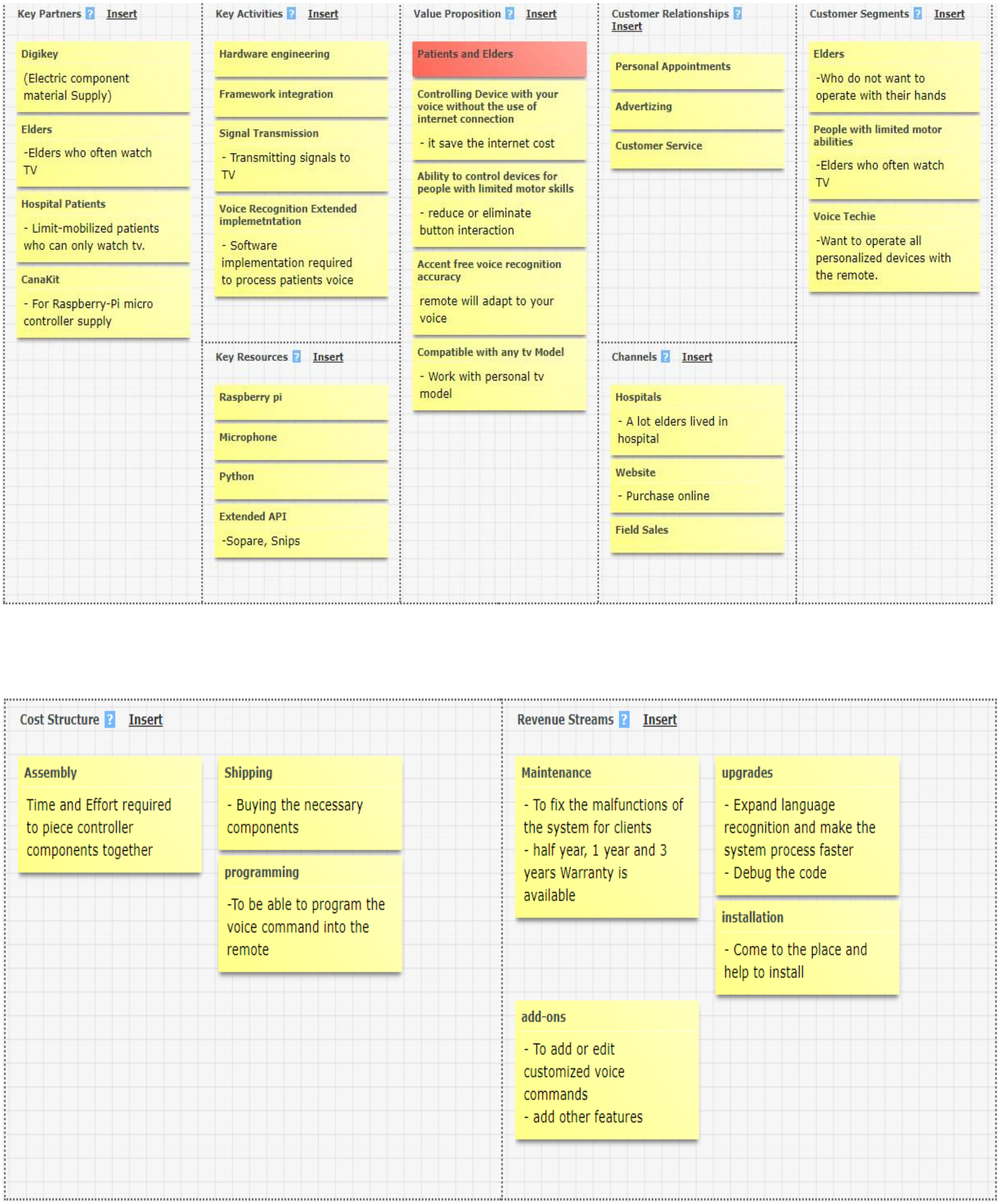
Figure 19:Business Model

Business Model Canvas

The following assumptions were made in terms of what our future business will require in order to be successful and be able to create value to society.

- The products will be sold in face to face by appointment. The feasibility is that customers can come to see how the product works and examine the products so they can be sure if the product is worth to purchase.
- The channel of hospital is a great way to do business because there are a lot elders lived in hospital or nursing home so they can get voice remote from hospital which is more convenient to get our products.
- Online sale is also a good way for clients who love to do online shopping. Usually it can be delivered to homes.
- Maintenance and upgrades are two main revenue streams for business. Voice control remote is machine so it might get malfunctions and bugs in the future. Offering warranty is a good way for make profits for us. And after we keep study about the products, we will also come up with some new idea about the voice remote like making the machine accept different language or enhance its ability to recognize voice which will be benefit to the business.
- Hardware engineering is the main part for our products. Unlike others, our product is not rely on Wi-Fi so it is important for us to make a good hardware so it is powerful enough to handle codes.

Business model Canvas



Key Partners ? Insert

Digikey

(Electric component material Supply)

Elders

-Elders who often watch TV

Hospital Patients

- Limit-mobilized patients who can only watch tv.

CanaKit

- For Raspberry-Pi micro controller supply

Key Activities ? Insert

Hardware engineering

Framework integration

Signal Transmission

- Transmitting signals to TV

Voice Recognition Extended implemetntation

- Software implementation required to process patients voice

Key Resources ? Insert

Raspberry pi

Microphone

Python

Extended API

-Sopare, Snips

Value Proposition ? Insert

Patients and Elders

Controlling Device with your voice without the use of internet connection

- it save the internet cost

Ability to control devices for people with limited motor skills

- reduce or eliminate button interaction

Accent free voice recognition accuracy

remote will adapt to your voice

Compatible with any tv Model

- Work with personal tv model

Customer Relationships ? Insert

Personal Appointments

Advertizing

Customer Service

Channels ? Insert

Hospitals

- A lot elders lived in hospital

Website

- Purchase online

Field Sales

Customer Segments ? Insert

Elders

-Who do not want to operate with their hands

People with limited motor abilities

-Elders who often watch TV

Voice Techie

-Want to operate all personalized devices with the remote.

Cost Structure ? Insert

Assembly

Time and Effort required to piece controller components together

Shipping

- Buying the necessary components

programming

-To be able to program the voice command into the remote

Revenue Streams ? Insert

Maintenance

- To fix the malfunctions of the system for clients
- half year, 1 year and 3 years Warranty is available

upgrades

- Expand language recognition and make the system process faster
- Debug the code

installation

- Come to the place and help to install

add-ons

- To add or edit customized voice commands
- add other features

Figure 20:Business Model Canvas

Economic Analysis

Considering that a possible business can be developed from our product. We must look into many economic aspects that will come into play when the business will be operating. We devised a cost list which covers every type of cost present on our potential business. Using these costs and the BOM, we created an income statement of the expected profit our business could potentially make. The core assumptions used into making these costs possible are stated.

Cost List

The following table outlines different types of cost that are involved in our potential voice remote business TalkToIt!. Some costs are considered as if the company is already in large scale. If our company plans to sell 400 voice remotes in the first year.

Cost Type	Cost Name	Cost
Variable	**3D material filament price per kg ($20)**	**x*$20 (X represent the amount of material filament bought)**
Direct	**Prototyping** (total price that it took to design one prototype alone) Devised from only one raspberry pi controller, one RF transceiver shield and microphone	**$80**
	High-Volume Manufacturing materials: (expected cost to make a high volume amount of voice remote controllers 400 units for the first year)	**$15000**
	Equipment- Expense 3D printers required to print casing (price $400)	**$400*20 printers=$8000**

Indirect	**Electricity :**For the first year In a small place to begin the business). Can be semi Variable.	**$10000**
	Product Marketing Expense. Required to advertise our product to other possible customers	**$2000**
	Depreciation Expense, Value of 3d printers goes down.	**$1000**
Fixed	**Salaries- Labor (**Assuming we are only hiring 5 workers to design the remotes)	**$20000**
	Company Facility Rent. (Renting a small space) to begin the business	**$20000**

Table 5:Cost List

Income Statement

After 3 years, our business will be selling about 5000 voice remotes for $130 per unit.
It will also be taken into consideration that 250 kg of filament is bought. The income statement is taking into assumption that electricity cost will double and that the production of voice remote has grown exponentially. The salary is taken with minimum wage of $12.50.The depreciation takes into consideration the cost lost through the use of the 3D printers over time.

Sales Good		**5000*$130=$650000**
Goods Cost		**$80*5000=$400000**
Gross Profit		**$650000-400000=$250000**
Operating Expenses	**High Volume Manufacturing production**	**$10000*3 years = $30000**
	Product Marketing	**$10000**
	Electricity	**$20000**

	Material Filament required	**$20*250 kg=$5000**
	Facility Rent	**$20000**
	Salaries	**$120000**
	Equipment	**$8000**
	Depreciation	**$5000**
Total Expenses		**$618000**
Operating Income		**$650000-$618000= $32000**

Table 6:Income Statement

Core Assumptions

We assumed our product is demanded so the sales are not declined by the time passed because there are always clients with common health issues that the voice remote can compensate for. We also assumed that the business started in the small facility which is why there is not a high rent and electricity cost to pay for. As a result, electricity cost at this point is considered semi-variable until future relocation.

Due to the company starting at a small facility and taking into consideration that majority of the of the work is just assembling the microcontroller, not too many employees are required. The salaries is taken into the account that the start there are only about 5-10 employees with each having equal salaries.

The price of the final voice remote prototype is taken into consideration as $80 based on the cost of the raspberry microcontroller, which is about $40, transceiver shield ($20), and the microphone quality ($20).

For the Equipment costs it is also taken into consideration that the company will be using only 10 3D printers since not many 3d printers are required at this stage of business (1 year to 3rd year). The average price of one 3d printer comes to be approximately $400.

Break-Even Point

We are going to do a NVP analysis for a period of 5 years and a tax MARR rate of 10 percent. We will calculate our break analysis according to that period of time.

$$NVP = -23080 + \frac{32000}{(1+0.1)1} + \frac{32000}{(1+0.1)2} + \frac{32000}{(1+0.1)3} + \frac{32000}{(1+0.1)4} + \frac{32000}{(1+0.1)5} = \$\ 98225.17$$

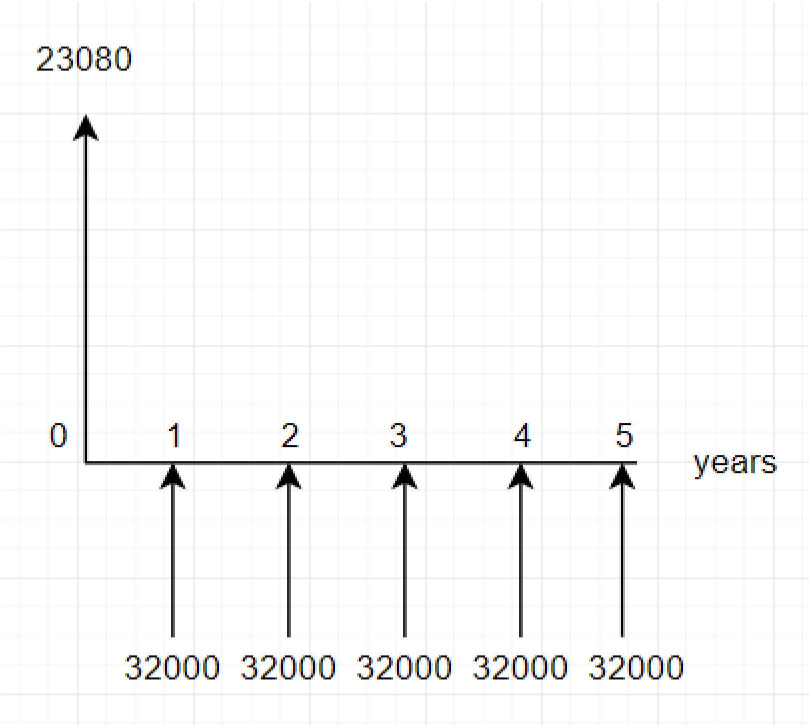


Figure 21:Cash flow diagram for TalkToit for 5 years

According to our calculations our company TalkToIt will break even at 1262 units. The chart below describes the chart obtained.

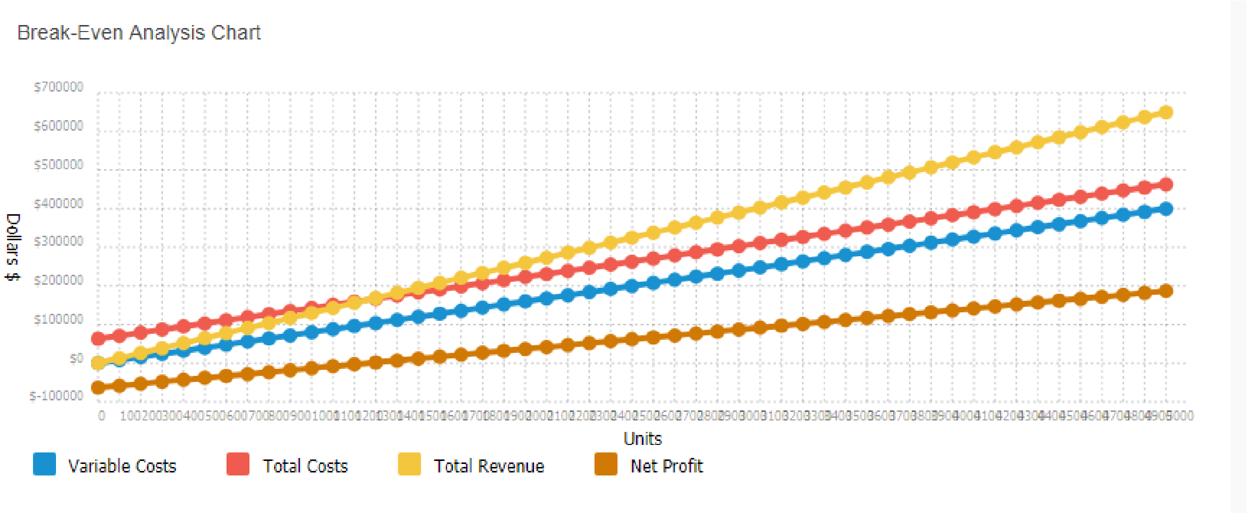


Figure 22:Break-even chart

User Manual

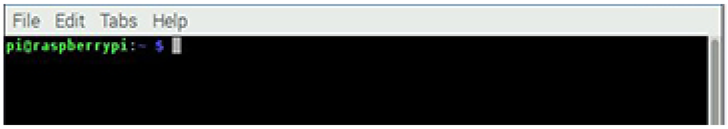
The developed product consists mainly in three systems that will need to be set up to fully customize the performance of this product to a specific user. Sopare is the software that controls the voice recognition system of the product, LIRC is the software that governs the reception and transmission of the IR signals and finally there must be an initial set up for any USB microphone that is used with the product.

USB Microphone setup

Any type of USB microphone will work; however, we will need to set the USB microphone as the default audio system of the raspberry pi.

Follow the following steps to set your USB microphone as the default device.

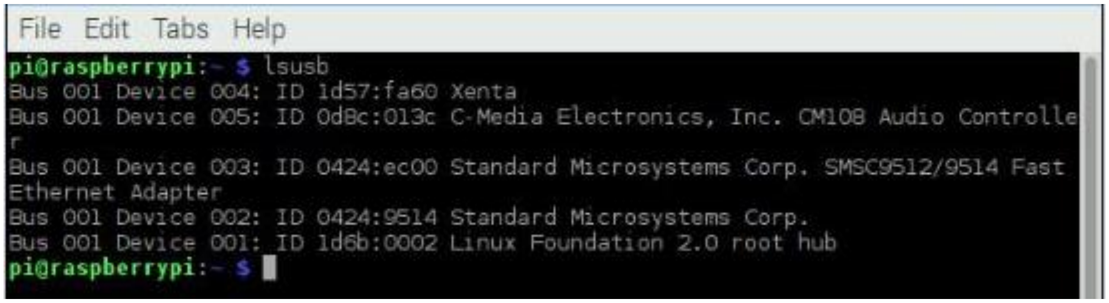
1. Connect the USB microphone in the Raspberry pi, through any USB port of the pi
2. Open the terminal in the Pi



```
pi@raspberrypi:~ $ 
```

3. Check the USB connection

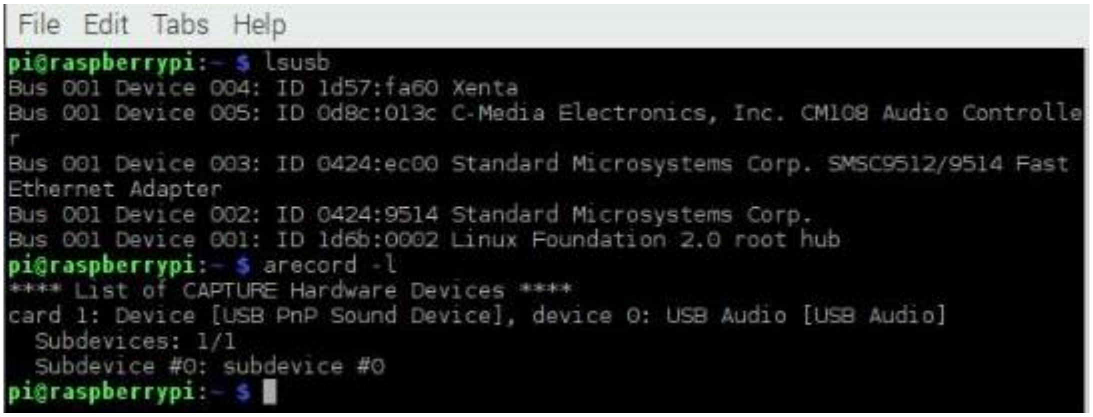
Type: **lsusb**



```
pi@raspberrypi:~ $ lsusb
Bus 001 Device 004: ID 1d57:fa60 Xenta
Bus 001 Device 005: ID 0d8c:013c C-Media Electronics, Inc. CM108 Audio Controlle
r
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast
Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
pi@raspberrypi:~ $
```

Make sure your device is listed: C-Media Electronics, Inc. CM108 Audio Controllers the USB microphone connected in the pi.

Type: **arecord -l**



```
pi@raspberrypi:~ $ lsusb
Bus 001 Device 004: ID 1d57:fa60 Xenta
Bus 001 Device 005: ID 0d8c:013c C-Media Electronics, Inc. CM108 Audio Controlle
r
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast
Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
pi@raspberrypi:~ $ arecord -l
**** List of CAPTURE Hardware Devices ****
card 1: Device [USB PnP Sound Device], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
pi@raspberrypi:~ $
```

Make sure that in Card 1 the USB PNP sound device is listed and the Device 0 is USB Audio

4. Make USB Microphone the default Sound Device

Type: **sudo nano /usr/share/alsa/alsa.conf**

look for the following lines:

```
defaults.ctl.card 0
defaults.pcm.card 0
```

and change them to 1:

```
defaults.ctl.card 1
defaults.pcm.card 1
```

5. Make the default PCM (audio) output card #1 and the default control also card #1

Type: **sudo nano /etc/asound.conf**

and put the following in the file and save:

```
pcm.!default  {
 type hw card 1
}
ctl.!default {
 type hw card 1
}
```

The USB microphone has been now set as the default sound device, now the raspberry pi will always recognize the USB microphone as its default device.

Voice Recognition System

SOPARE stands for SOund PAttern REcognition and is a Python open source project developed on and for the Raspberry Pi. SOPARE allows us to execute offline and real time audio processing for any type of words that must be trained upfront.

SOPARE was developed by Bishoph organization, if you want to know how SOPARE software works and you want to get a more detailed explanation on configurations and especial settings for voice recognition please go to https://www.bishoph.org

SOPARE can learn sounds from training sessions and it will identify the same sound later on even under different circumstances. This means that you can train words in any languages or you can train any type of sounds like doorbells, knocks or whatever you want. The source code and more information are available on GitHub.

The current product has all the libraries and plugins installed and working, the only step the user must do is to train the desired words to control any device. To be able to do that please go to the “train words” section of this manual.

If you have a Raspberry pi and you would like to install SOPARE and start from scratch go to the “Start from scratch” section.

START FROM SCRATCH

SOPARE Installation

Make sure you are connected to the internet for the installation of SOPARE once you have installed it in the raspberry pi you will not need an internet connection

1. Open the terminal on the raspberry pi (make sure you are connected to the internet)
2. We will update the raspberry pie system so type:

   Type: **sudo apt-get update**
3. Let us get all the libraries needed, type:

   Type: **sudo apt-get install build-essential python-pyaudio python-numpy-python-scipy python-matplotlib**

4. Create a Directory to save SOPARE, we will call it Dev, so type:

```
mk dir Dev     // Making the directory "Dev"
cd Dev         // Going inside directory "Dev"
```

5. Once we are inside Dev directory we will get SOPARE from Github, type:

```
git clone https: //github.com/bishop/sopare.git
```

6. Once we have downloaded SOPARE, a Sopare folder will be created in our Dev directory, go inside the sopare folder and create two folders tokens and samples:

```
cd Sopare
mkdir tokens
mkdir samples
```

At this point SOPARE has been installed in your Raspberry pi and it is ready to be train words.

Training Words

SOPARE will create an array of tested and trained words that will be recognized, later these words will become commands that will activate the IR system to send signals to any electronic device to control it.

This part of the process is very important, and it can be complex, however the good use and a good setup of this words will make the voice recognition system very accurate and precise.

The user should decide what word he/she will train according to the device that he/she wants to control with voice commands, for instance we will train the voice recognition system with 5 words to control a TV and 4 words to control a light system.

Training words for the TV

Word	**Command**
Power	Turns TV ON and OFF
Mute	Mutes the TV
Volume Up	Turns volume up
Volume Down	Turns volume Down

Training words for Light System

Word	Command
Lights On	Turns lights On
Lights Off	Turns lights Off
Still	Makes lights to be constantly on
Shuffle	Makes light to go on and off

1. Open the terminal
2. Go to Sopare folder

Type: **cd dev**
cd Sopare

3. Let's check the audio tests (Sopare executes some audio tests to choose the best configuration of arguments, according to the environment).

Type: **python sopare.py -u**

Make sure there are not errors, if errors; make sure the microphone is being detected by the raspberry pi, check USB microphone setup section

4. Let's get the initial parameters

Type: **python test/test_audio.py**

This command will give you the values for the following arguments:

```
Your sopare/config.py recommendations:

SAMPLE_RATE = 48000

CHUNK = 512

THRESHOLD = 100
```

5. Let's load those values into the configuration file

Type: **nano config/default.ini**

You can now edit the configuration and change the file accordingly to the recommendations given in step 4.

6. Now we are ready to start testing words for recognition.

Let us train the word Power so

Type: **./sopare.py -v -t Power**

As soon as you hit enter wait until the screen shows:

```
INFO:sopare.recorder:start endless recording
```

When the screen displays the text above you should say the training word so at this point you should say “Power” and then the software will get your input and save it. You should see lots of lines rush over your monitor. This is good as SOPARE logs some debug information. If the lines are rushing before you said something SOPARE started the training because something triggered the THRESHOLD. In that case you should delete the trained file(s) and start the training again, maybe with a higher THRESHOLD. This will be explained later.

If you see in your monito a line saying “Broken Pipe” that means that the word was not saved correctly and that you have to say the word again.

For each command word you must train it three times.

Let us train the word “Lights On”

Type: **./sopare.py -v -t LightsOn**

If you do not get any error or broken pipe message, it means that the word was saved correctly, remember that you must do it three times per each word.

7. Once you have trained all your words you have to compile Sopare and saved the trained words so:

Type: **./sopare.py -c**

8. Now let us test the trained words.

Type: **./sopare.py -l**

Say the words you trained, and the monitor should display the word as you say it.

For instance, if you say power then the monitor should display:

```
[‘Power’]
```

If you say Lights On the monitor should display:

[‘LightsOn’]

9. To delete files in the dictionary and restart the training please type:

```
rm dict/*.raw
./sopare.py -d “*”
```

All the trained files will be deleted so that you can start a new training

At this point the training words should be saved in a dictionary array in the Sopare folder. When training words in a noisy environment, the recommended configurations will not be enough for the voice recognition system to be fully accurate.

Quick Set up for Accuracy and Precision:

Marginal_value: It is the threshold to identify a word or a pattern
    0= Everything will be identified as the beginning of a word
    1=Current trained sample and sound must match 100%
    Recommended value (0.7 – 0.9)

MIN_CROSS_SIMILARITY: This threshold is to adjust the comparison of words or sounds
    0.6 = Multiple words commands (Lights Off, Volume Up)
    0.9= Single Words (On, Off, Still, Shuffle)

To improve the precision and accuracy of the trained words please read the documentation found in: https://www.bishoph.org/sopare-precision-and-accuracy/ this will help you to get the best configuration for the best accuracy and precision.

Reception and Transmitting of IR signals (LIRC)

To control any electronic device with voice commands, the product needs to get the IR codes from the device you want to control (TV, Light system etc) once our system gets the desired signal codes it can then send those codes to the device (TV, Light system etc) to control the device.

First our product needs to get the code signals from the remote controller of the device you want to control with your voice, once the system gets these codes a configuration file will be created for the remote and then the system will use this configuration file to send signals to the corresponding device.

The software that allows our system to do this is called LIRC and it is an open source software. The product as it is right now including all the LIRC libraries and dependencies, if you are using our product you will have to go to “Create new remote section” to configure the codes of the device you want to control. If you have a raspberry pi and want to start from scratch start from “LIRC installation” section.

LIRC Installation

1. Open the terminal
2. Install LIRC

Type: **sudo apt-get install lirc**

3. Assign the out/input pins of the GPIO of the raspberry pi
   Type: **sudo nano /etc/modules**
   And add:
   ```
   lirc_dev
   lirc_rpi gpio_in_pin=18 gpio_out_pin=17
   ```
4. Make the IR transceiver hardware be the default hardware for the raspberry pi

Type: **sudo nano /etc/lirc/hardware.conf**

Change it to:

```
lirc_ARGS=”--input”
LOAD_MODULES=true
Driver=”default”
Device=”/dev/lirc0”
Modules=”lirc_rpi”
```

5. Edit /boot/config.txt

   Type: **sudo nano /boot/config.txt**

   Add:
   ```
   dtoverlay=lirc_rpi, gpio_in_pin=18, gpio_out_pin=17
   ```

6. Create modeprobe.d
   Type: **sudo nano /etc/modprobe.d/ir-remote.conf**
   Add:
   ```
   Options lirc_rpi gpio_in_pin=18 gpio_out_pin=17
   ```
7. Finally **reboot pi**

We have configured the GPIO unit of the raspberry pi now you will be able to use the I/R transceiver, now you have to create a remote by getting the signal codes from the remote using the Receiver.

Getting IR Signals from remote controller

1. Testing the receiver

Type: **sudo modprobe lirc_rpi**

Press any button on your remote controller and the terminal should display bits every time you press a button, that means that the receiver is working.

2. Get the LIRC file with all the command names available for LIRC
   Type: **irrecord --list-namespace**
   You will get a text file with all the KEY commands that lyric will recognize save that file for future reference.
3. Create a new Remote configuration file
   Type: **irrecord -d /dev/lirc0 ~/lircd.conf**
   Please follow the instructions that will be appear on the terminal, you will be saving the Hexadecimal code of every button pressed on the remote controller. Follow the prompts displayed on the terminal.
   In this step you will name the remote that you are setting up. For demonstration purposes let us suppose you saved the remote as: SamsungRemote
   When being asked to save a command make sure you use the commands from the text file you saved in step 2, for instance if you want to save the hex code for the power button of your remote, you should save it as KEY_POWER. Please see all the key commands in the text file from step 2.
4. Coping your remote to LIRC

At this point you should have created a remote controller by following the prompts from step 3, now you have a configuration file with the hexadecimal codes for every button of your remote controller, now we need to add this folder into the LIRC configuration, to be able to use these codes and send them to the TV using the Transmitter.

Type: **sudo cp SamsungRemote /etc/lirc/**

Note that after the cp command you should type the name you gave to your remote in step 3.

5. Sending signals to TV

Let’s list the commands that LIRC knows for the remote you have created the configuration file for, in our case for SamsungRemote:

Type: **irsend LIST SamsungRemote**

It will display all the list of commands saved on LIRC for the Samsung TV

Let’s send a command to the TV, for instance let us send the KEY_POWER command once to the Samsung TV:

Type: **irsend SEND_ONCE SamsungRemote KEY_POWER**

The TV should turn on or off.

Let us send the KEY_VOLUMEUP command once:

Type: **irsend SEND_ONCE SamsungRemote KEY_VOLUMEUP**

The Tv should turn the volume up.

At this point you should be able to control your tv or electronic device from the terminal of your raspberry pi.
Now we must integrate the Voice recognition System with the IR Transceiver system to control your electronic device with your voice commands.

Voice Recognition and IR Integration

You should be able now to control your electronic device by typing commands on tour raspberry pi terminal and you also should have some training words saved in your system. We now must integrate the two modules so that you can control your electronic device with your voice.

1. Go to Sopare folder in the raspberry pi

Type: **cd Dev**
cd sopare

2. Open the Plugins folder that is in the Sopare folder

Type: **sudo nano /Dev/sopare/plugins**

This will open a python program that contains the array of all the trained words that you have saved before.
We need to modify this code to integrate the voice recognition system with the IR transceiver.

The file should display code like this:

```
Def run (readable_results, data, rawbuf):
        Print readable_results;
```

Readable_results is an array containing all the training words as you saved them in the training session. For example, if you saved the commands Power, Volume Up, Volume Down and Mute as power, volumeUp, volumeDown and mute respectively then the readable_results array will look something like this:

```
Readable_results=[‘power’,’volumeUp’,’volumeDown’,’mute’]
```

We need to send the IR signal as soon as the algorithm recognizes any of the words saved in readable_results array, as the user says the word the algorithm will recognize it and we need to select the correct command to transmit from the raspberry pi to the TV.

To achieve this, you need to change the displayed code to something like this:

```
import os

def run (readable_results, data, rawbuf):
        If(‘power’ in readable_results):
        os.system(“irsend SEND_ONCE SamsungRemote KEY_POWER”);
        If(‘volumeUp’ in readable_results):
        os.system(“irsend SEND_ONCE SamsungRemote KEY_VOLUMEUP”);
                If(‘volumeDown’ in readable_results):
                os.system(“irsend SEND_ONCE SamsungRemote
                KEY_VOLUMEDOWN”);
        If(‘mute’ in readable_results):
        os.system(“irsend SEND_ONCE SamsungRemote KEY_MUTE”);
```

Save the script

3. Run the product
   At this point the product is ready to use and it should be placed so that the transmitter is pointing to the electronic device you want to control.
   Make sure you are on the Sopare folder and type:

```
./sopare.py -l
```

Say the commands that you trained, and the Device should control the TV or the electronic device that you wanted to control.

The device will be in an infinite loop always listening for the command words that you trained it for.

Conclusion

During this project of voice remote control, we feel this project is interesting because this project brought us an experience with dealing with clients and customers. We made a great teamwork when we deal with each project deliverable. And we learnt a lot of skills during the project.

If we are going to continue this business in the future, we will need some modification and improvement on the product. To improve the product, we plan to add wake up command to the product so it can be activated by keywords with a signal light. We also decide to create bunch of environment settings. This setting can be used when facing different voice environments. To become customer beneficial, we will need to design a new user interface to make sure that customers can easy use it.

Bibliography

[1] Engineeringtoolbox.com. (2019). *Voice Level at Distance*. [online] Available at: https://www.engineeringtoolbox.com/voice-level-d_938.html [Accessed 13 Apr. 2019].

[2] 3ders.org. (2019). *3ders.org - price compare 3D printing materials*. [online] Available at: https://www.3ders.org/pricecompare/ [Accessed 13 Apr. 2019].

[3] Lirc.org. (2019). *LIRC - Linux Infrared Remote Control*. [online] Available at: http://lirc.org/?fbclid=IwAR1xNXTwL4v9vdcNo0rRhlRNmZkXRbq0vTafq3p1EYb0Jqyi89Mp0wwiqOE [Accessed 13 Apr. 2019].

[4] Bishoph.org. (2019). *SOPARE architecture and plugins | home of bishoph*. [online] Available at: https://www.bishoph.org/sopare-architecture-and-plugins/?fbclid=IwAR2MaGk9ikoGVUug4nhfGOU3wZtUOKLruUt2Oxyuo-U-wLzYZyo0Xy83jJo [Accessed 13 Apr. 2019].

[5] Salcedo, N., Guzman, J. and Yin, S. (2019). *GNG2101-C1A-Voice Remote | MakerRepo*. [online] Makerepo.com. Available at: https://makerepo.com/nsalc050/gng2101c1avoice-remote [Accessed 13 Apr. 2019].