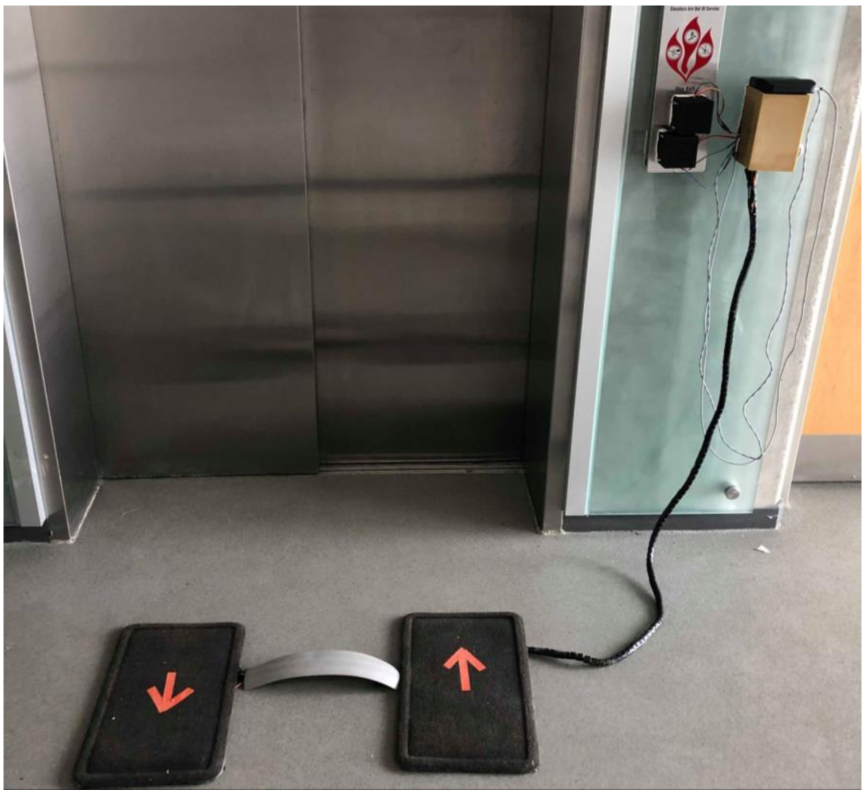
SensiTouch

A universal pressure sensitive mat providing independence and accessibility



Alisa Douglas, Luka Mircetic, Nardine Abd Elmaseh

Team 38

April 9, 2019

Table of Contents

I. Academic Integrity Statement

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Submitted by [Nardine Abd Elmaseh, 400172907]

Signature

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Submitted by [Luka Mircetic, 400196057]

Signature

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Submitted by [Alisa Douglas, 400177625]

Signature

II. Executive Summary

SensiTouch is a unique solution to many accessibility issues for quadriplegics and other individuals with disabilities. It is long-lasting, efficient, and can be implemented across all buildings. It is a device designed to give access and control to users whose disabilities prevent them from living their daily lives independently.

Quadriplegia is the paralysis of all four limbs. The client behind the influence of this device is a quadriplegic who experienced a spinal cord stroke and has had no movement from the neck down since its occurrence. The client's daily tasks and routines described many challenges; one main issue was the inability to open doors and elevators independently despite accessibility buttons present. In accordance with many other quadriplegics and disabled individuals, independent tasks are difficult to perform due to their limited range of motion and inconveniently placed accessibility buttons.

SensiTouch explicitly resolves problems of independently accessing elevators and doors. It is a pressure sensitive mat with multiple cushioned force sensors. It is connected to a mechanical component that is triggered by a sense of pressure. The purpose of the device is to open elevators with applied pressure by stepping or rolling on the mat. It is a reusable device that can be implemented onto many types of elevators by placing it on the ground and attaching the cased mechanical component onto the buttons. This external component is added on as an additional layer and is created to physically push the elevator button. It is triggered when there is a pressure signal received or the external button on the component is pressed. The device is frequently used and continues to process pressure and open the elevator doors accordingly. SensiTouch can operate in both external and internal environments.

SensiTouch functions to create an easier and more accessible way for individuals with disabilities. It prevents extra strain on the person to individually reach for buttons and limits the need of a caregiver to help access out of reach buttons. The device creates a solution for quadriplegics and many disabled individuals to independently travel in and around buildings.

In the future, SensiTouch anticipates stronger materials and a larger number of sensors to increase the accuracy of pressure sensing. Individuals with disabilities can foresee SensiTouch as an efficient and useful device because of the independence and accessibility it provides.

III. Background and Underlying Condition

Ryan is a third-year Political Science student and is interested in improving accessibility for individuals with disabilities. At the age of 17, he experienced a spinal cord stroke while at the gym. It caused him to lose movement from the neck down and he has learned to adapt to his disability by using various technologies. He has a very limited range of motion and often needs a friend to help in performing daily tasks such as drinking water, eating, and opening doors and elevators. Ryan uses a power wheelchair and other assistive technologies to travel between locations.

Ryan expresses many challenges in adapting to his disability; one main challenge he experiences is getting in and out of rooms and buildings despite the presence of an automatic wheelchair button. He has limited arm movement and therefore, is unable to reach the buttons if they are not properly positioned. When he often tries to extend his body, it heavily affects his movement and as a result, he experiences irritable muscle spasms. Moreover, Ryan notices that it is often difficult to generate enough force to push on buttons independently due to his limited motor abilities. Ryan depicts that others in the university are kind and understanding of his disability and often assist him. Therefore, he relies on other individuals to help in opening doors and is fearful when no one is present due to his inability to independently perform the tasks. In accordance with his fear, he regularly has a friend by his side to assist in daily tasks, moving around locations, and obtaining notes in school. In addition to the struggle of pushing elevator and door buttons, Ryan is regularly at the university campus and recognizes the variety of doors and elevators. He expresses that he does not have enough time to enter due to some doors and elevators being too quick or too slow.

An article reviewing elevators and the disabled discusses the hazards individuals in wheelchairs struggle with, in working and accessing buildings in situations where emergencies occur such as a fire [6]. The issues discussed in the article are summarized from 1989, and the author depicts that although accessibility regulations have improved over the years, they continue to be a present issue. Although individuals assist wheelchair users constantly, as observed with Ryan’s experiences around the university, it is difficult to comprehend their struggles in performing tasks a non-disabled individual finds very easy. To seek solutions to their challenges, Ryan and many other disabled individuals continue to implement many assistive technologies to their lifestyle. In an article examining assistive technology with quadriplegics, independence is desirable for the individual so tasks are more bearable for caregivers, and therefore, improve the morale of the disabled individual [5]. Therefore, any technology designed to improve one's ability to perform simple tasks independently can enhance their self-esteem. Another study examining the impact of assistive devices in quadriplegics displayed that simple devices were often advantageous due to their easy use and simple control [4]. In accordance with Ryan, he expresses that he has adapted to his disability, however, in using assistive technology, he is interested in designs that are easy to use and allow him to be independent.

Similar to other individuals who are physically disabled, Ryan hopes to inform and bring greater light to some of the many challenges of accessibility he and many others conquer daily. He anticipates solutions to improve his sense of independence and therefore, enhance methods of physically moving from several locations.

IV. Need Statement

Throughout the duration of this project, we anticipate designing a device that allows Ryan to access and control elevator and door buttons more easily since he has fine and gross motor deficits that prevent his performance in

V. Specifications

A. Product Identification

- Our proposed product, SensiTouch is a mat that is pressure-sensitive purposely to signal doors and elevators to open. This is to make buildings more accessible for disabled users. They do not have to angle their chair or rely on others to assist in opening doors or pressing buttons, but can rather step on the mat, which will trigger the button to be pressed.
- The product can be broken down into two components, one for doors and another for elevators. Regarding elevators, the product is two separate parts to distinguish for the up and down button, while the door has one single mat to open it.
- Our device is an addition to elevators and doors. It requires physical force from either one's step or an external device such as a wheelchair. It is placed in front of elevators and doors and functions when it senses pressure.
- The mat is wired to a mechanical component that when pressure is sensed, will turn a motor to push down on the button. In accordance with doors, the motor part pushes on the accessibility button previously integrated with the door system. Moreover, the motor pushes the elevator's respective buttons based on the specific mat that senses pressure (up or down)
- A key feature is that individuals can continue to press the elevator or door button without the use of the mat. Our device's mechanical component is housed in a box, which is used as an additional layer on top of the elevator button itself (figure 10.c). There is a button protruding out of the box to allow other users to press.
- Our device can be easily implemented in society. Our device addresses Ryan's issues, along with many other individuals with disabilities. It is not specific to one individual but is targeted towards the general disabled community. It is an addition to the elevators and doors and is a cost-efficient solution that can be executed at every location.
- The mat's appearance is similar to other regular mats, it is a carpeted mat with hidden sensors underneath, and each mat has an arrow to indicate where the user should apply pressure to call the respective button (up or down).

- The mat is not directly embedded in the building but is reusable and can be moved accordingly. It can withstand multiple pressures without causing an error in the system due to its durability.
- The mat is lightweight and can simply be picked up with minimal effort. However, along with its mechanical components, the mat is roughly two kilograms. The mechanical parts and the casing for the wires, raspberry pi, and the remaining computing parts, increase the overall weight of the product but are light enough to be fastened to the elevator with a material such as sticky tac. In the future, a stronger material or component such as glue or screws can be implemented to securely affix the casings.
- Considering there are two mats in the elevator component, it allows a clear indication of each mat's function. There is a space in between both mats to allow individuals to stand and call the elevators using the button if they do not wish to use the mat.
- In accordance with the mat's versatile uses, it does no harm to its users and the space around it. It is easily assimilated into society in accordance with its reusability. They are compactly sized and do not occupy a large area, and therefore has a very low footprint.
- However, the mechanical casings increase the extrusion of the elevator button and take up a bigger area (Figure 10.c). Moreover, our device has a large number of wire extensions to accommodate the mat's placement on the ground. Therefore, our wires are carefully cased but do take up space in connecting the mechanical components with the mat itself.
- Our device uses a variety of different system processing components and so requires multiple goods such as sensors and wires to produce. These resources and components add to its manufacturing and its ecological footprint.

B. Usage

- SensiTouch acts as an external component; it has built-in sensors that attach to the Raspberry Pi as well as the motors that will press the elevator buttons.
- The motors are contained inside a box that will be attached on top of the original elevator or door buttons as an additional layer. It does not require access to the elevator's circuit

and therefore, can easily be removed and implemented onto any elevator doors as well as other automatic doors

- For the door buttons to be triggered, the mat must sense pressure either from someone standing on it for a specific amount of time or from a wheel of a wheelchair. Once the pressure has been sensed, a signal is sent to the motor and causes it to turn and press the door button
- The user needs to stand on the mat for a specific amount of time to trigger the button as a way to differentiate between someone purposefully trying to press the button and someone accidentally walking over the mat. The taken average in the processing system accounts for such accidents and allows for extra time for disabled individuals to pass through the elevator door.
- If the user does not want to use the mat, there is an additional button on the box of the motor that can be pressed. When the additional button on the box is pressed, it will trigger the motor inside the box to turn and therefore press the original door button behind the box.
- The two methods for pressing the button makes SensiTouch versatile. It can be used by the general public without any inconvenience and by wheelchair users. Moreover, other people can also use it if their hands are full such as laborers, and maintenance and construction workers.
- SensiTouch allows wheelchair users to access the elevator independently without straining to reach the button and reposition themselves to get into the elevator quickly. They can roll onto the mat which will already be in front of the elevator and easily go into the elevator without repositioning themselves.
- The mat has a significant amount of cushion to protect the sensors. This feature makes it easy for wheelchair users to roll over the mat.
- The mat and the program should run infinitely and can be used as often as needed as long as all components of the device are working. However, if the mat is not used, it can remain in front of the doors.
- SensiTouch can be operated in front of any automatic doors or elevators, whether it is indoors or outdoors

- Already existing commercial products can be used to protect the carpeted mat from the outside conditions such as sprays that can make the carpet waterproof.
- The cables will also be covered to protect from animals forming chewing on them or any possible wire damage

C. Standards and Safety

- Some possible factors that can cause the device to fail can be due to damage to the sensors from improper use, the code does not work or motor malfunctions or is damaged
- The probability of failure mostly depends on sensors and how users interact with them; if people use the mat improperly by either stomping on the mat or jumping on them, then it can damage the sensors and cause failure.
- Moreover, if the sensors are not placed properly on the mat it can also cause the sensors to not detect pressure will also prevent the device from functioning properly
- Other factors such as malfunctioning of the motor will be likely caused by the strength of the motor and how secure it is placed
- If these failures do occur, SensiTouch will not be able to function and therefore cannot press the door button, instead, the user will have to manually press the door button
- The severity of the malfunction is very low since it will not cause any physical harm and it is not hazardous to anyone’s health. However, it impacts the sole purpose and function of the device.
- Steps that can mitigate risks include proper use by the user, i.e. standing on the mat or rolling on to the mat rather than jumping or stomping. Also, having a durable material that will protect the sensors and wires can prevent damage.
- In the future, more sensors can be added to the mat so if there is a malfunction with a sensor, the remaining can continue detecting pressure, and behave accordingly.
- An important aspect to mitigate risks is by decreasing fatigue. This can be done by decreasing the size of the mat, perform maintenance checks, surface hardening and choosing appropriate material such as titanium which has high ability to withstand fatigue [2].

D. Manufacturing

- In creating our prototype, we began by developing the mat with the sensors. We connected all six sensors by soldering their corresponding wires to the circuit board (PCB) (Figure 15.b). The sensors were connected to two ADC converters which were connected to a Multiplexer breakout module (Figure 15.a). The sensors functioned in correspondence with a code (Figure 13). Additionally, there were two buttons attached to the PCB which control the motors through the aforementioned code.
- The code took each sensor reading (reading is taken every 0.5s) and appended each value to an existing data list. The average of the most recent values of this list was then taken. To clarify, each sensor's average reading is being computed independently from the others. If the average value was above the set threshold, the code functioned to allow a certain Servo motor to turn a certain angle.
- We continued to use Autodesk Inventor to sketch and create an attaching piece for the motor to physically press the elevator and door buttons, and casings to protect the wires and processing parts such as the raspberry pi (Figure 11).
- The next step involved extending our sensors using wires and placing them accordingly on the mat. When choosing positions for the sensor, we allocated them evenly to achieve accurate readings from different areas in the mat. Due to limited resources, we used three sensors for each mat. In expanding our device in the future, we plan to implement multiple sensors or use larger sensors to sense accurate pressures.
- We secured each sensor using electrical tape, on a separate cardboard piece, the same size as the mat. To ensure the sensors read pressure, we placed thin discs on the mat in the same locations to where the sensors are placed. This was done to ensure the sensors are receiving enough pressure when the mat is stepped on.
- We proceeded to hot glue the mat to the cardboard once all the wires and sensors were positioned correctly. The cardboard and mat were secured together to have the mat as one complete piece
- We installed the button and motor accordingly in their casings and encased all cords using a cord cover. The reason that there is two separate cases for the motor is so that it

can fit onto any elevator button interface, since not every elevator has the same spacing between buttons

- We used sticky tac to adhere each component to the elevator (Figure 10.c-d). In future implementations, it will probably be secured to the wall with screws.
- Our complete, high fidelity prototype was produced using a variety of accessible resources from the design studio, and other purchased equipment. The motors, raspberry pi, wires, PCB, 3D-printed parts, cardboard, tape, sticky tac, etc., were obtained from the design studio, while the cord covers, and the mat were purchased separately.
- Due to limited resources and timing, the mat's material is currently carpet. It is comfortable and easy to step and roll on. This material can be tedious to clean; therefore, we plan to introduce rubber-style mats, similar to car mats, that are weatherproof and more durable.

E. Life Cycle

- The device should have regular checks to ensure it is functioning as it is supposed to
- The mat can be separated into two sections which are the actual mat itself and the layer that contains the sensors
- When the mat is removed it can be washed regularly which can involve washing it manually and vacuuming if needed
- The mat can easily be replaced as long as it is the same size as the original mat. Also since no installation is needed since a new mat can simply be placed on top of the sensors.
- The wires and sensors can also be replaced but it will require installation to connect the sensors properly as well as securing it to the mat in proper positions
- The client can replace the mat by calling maintenance since it is used by the general public as well
- If SensiTouch malfunctions and can no longer be replaced, then a new mat can be purchased and installed
- The boxes for the motors and the Raspberry Pi can be kept and reused for each mat unless they are damaged or broken, then new boxes can be purchased

- The wires can be recycled and rubber mats can be reused for each new device or used for a different purpose if it is not damaged, i.e. car mats, door mats, etc.
- In the future, we plan to extend our device to a wireless connection, so it is easier to maintain and clean.

Figure 1: Production Cost

Materials	**Quantity**	**Total Cost**
Mat	2	$11
Wires	Approx. 150	$10
Circuit Board (PCB)	1	$8
Raspberry pi	1	$10
Force sensitive resistor	6	$36
Longrunner LKY62 motor	2	$10
Button	2	$10
Cable sleeve	1	$7
Floor cable cover	1	$5
3D printed parts	⅕ of a roll	$5
Total		$110

- The total production cost mostly consists of the prices of materials within the design studio and some out-pocket-expenses. Some materials such as the mats, cable sleeve, and the floor cable cover are not from the studio
- Both mats cost around $11, $7 for the cable sleeve and $25 for a 6ft. roll of floor cable cover, but only a small section was used which costs around $5

- The rest of the materials which are mostly 3D printed parts, sensors, motors, and computing equipment are all from the design studio. The device requires a lot of jumper wires in order to extend them in length as well as having multiple sensors to connect to. In total, approximately 150 wires were used which costs around $10 per packet of 120 wires
- The circuit board (PCB) costs around $10 and the Raspberry Pi costs around $10 [1]. The force sensitive resistors cost $36 in total for all 6 sensors which were distributed evenly between the two mats
- The two Longrunner LKY62 motors are high-torque motors that cost $10 total [3]
- Approximately a ⅕ of a filament roll was used for all the 3D printed parts which cost around $5 considering one full roll is $25
- The cost of all these materials combined comes to a total of $110.

VI. References

[1]“ELECTRONICS-SALON 1PCS Double-Side Prototype PCB, Universal Board, 77.4x72mm.,” *CZH*. [Online]. Available: https://czh-labs.com/-p0312.html?gclid=EAIaIQobChMIxLuz_4jC4QIVwoWzCh1lugizEAkYBSABEgIXkvD_BwE. [Accessed: 09-Apr-2019].

[2] J. Homan, “Which material should be selected for fatigue critical structures,” *Fatec Engineering*, 11-Jul-2018. [Online]. Available: https://www.fatec-engineering.com/2018/07/11/material-selection-for-fatigue/. [Accessed: 01-Apr-2019].

[3]“Longruner 10Pcs MG90S Metal Geared Micro Servo Motor 9G for Helicopter Airplane Boat Controls Mini Servo 450 LKY61 (MG996R Metal Gear),” *Amazon*. [Online]. Available: https://www.amazon.com/Longruner-MG996R-Digital-Helicopter-LKY62/dp/B01MTW06IU. [Accessed: 09-Apr-2019].

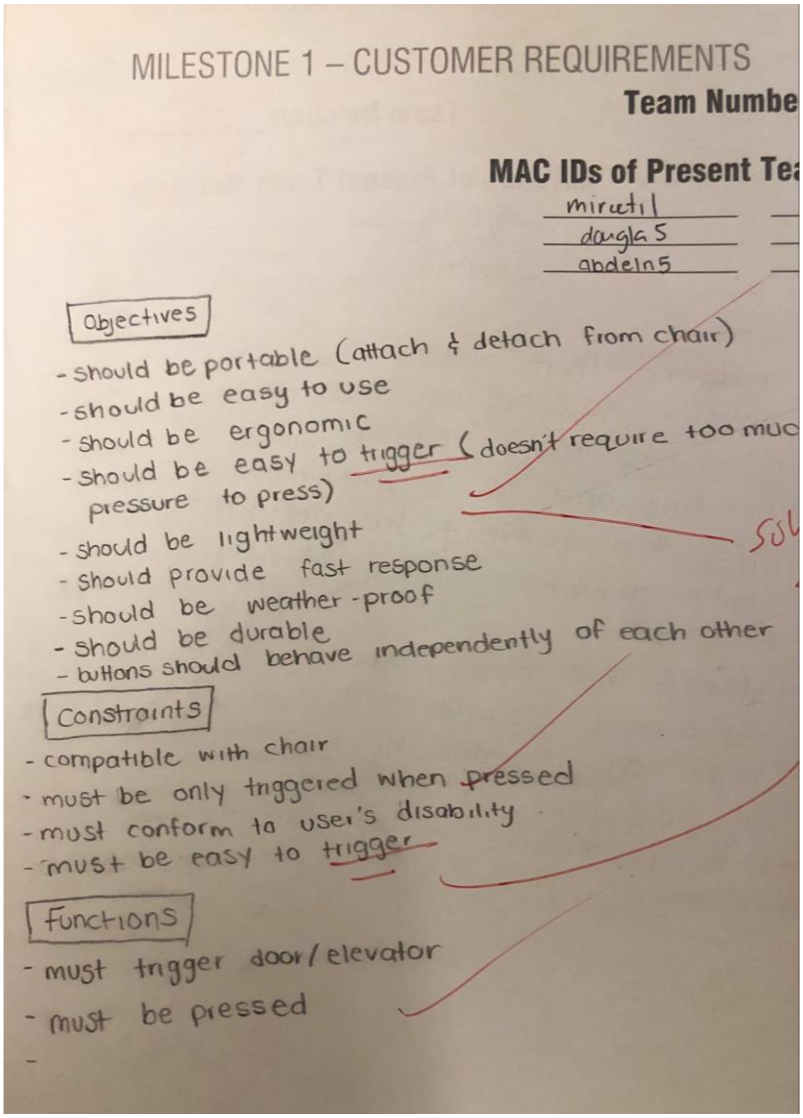
[4] P. Bell and J. Hinojosa, “Perception of the Impact of Assistive Devices on Daily Life of Three Individuals with Quadriplegia,” *Assistive Technology*, vol. 7, no. 2, pp. 87–94, 1995.

[5]R. G. S. Platts and M. H. Fraser, “Assistive technology in the rehabilitation of patients with high spinal cord lesions,” *Paraplegia 31*, pp. 280–287, 1993.

[6] R. W. Bukowski, “Protected Elevators and the Disabled,” *CiteSeerX*, 31-Aug-2005. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.531.3479. [Accessed: 06-Apr-2019].

VII. Appendices

Figure 2: Customer Requirements



MILESTONE 1 – CUSTOMER REQUIREMENTS

Team Numbe

MAC IDs of Present Tea

miruti1

dougla5

abdeln5

Objectives

- should be portable (attach & detach from chair)
- should be easy to use
- should be ergonomic
- should be easy to trigger (doesn't require too muc pressure to press)
- should be lightweight
- should provide fast response
- should be weather-proof
- should be durable
- buttons should behave independently of each other

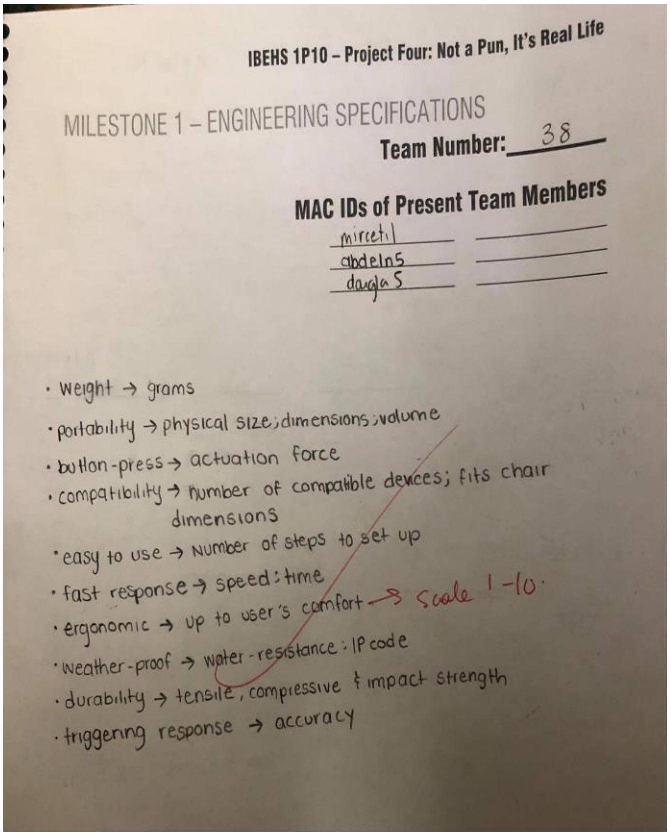
Constraints

- compatible with chair
- must be only triggered when pressed
- must conform to user's disability
- must be easy to trigger

Functions

- must trigger door/elevator
- must be pressed
-

Figure 3: Engineering Specifications



IBEHS 1P10 – Project Four: Not a Pun, It's Real Life

MILESTONE 1 – ENGINEERING SPECIFICATIONS

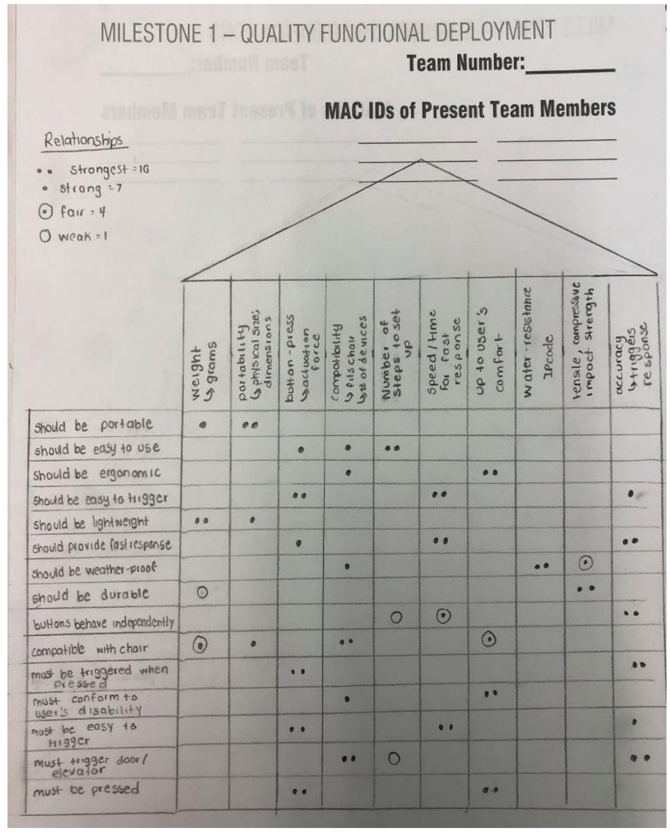
Team Number: 38

MAC IDs of Present Team Members

mircetil
abdelns
dauglas

- weight → grams
- portability → physical size; dimensions; volume
- button-press → actuation force
- compatibility → number of compatible devices; fits chair dimensions
- easy to use → Number of steps to set up
- fast response → speed: time
- ergonomic → up to user's comfort → Scale 1-10.
- weather-proof → water-resistance : IP code
- durability → tensile, compressive & impact strength
- triggering response → accuracy

Figure 4: Quality Functional Deployment Chart (QFD)



MILESTONE 1 – QUALITY FUNCTIONAL DEPLOYMENT

Team Number:

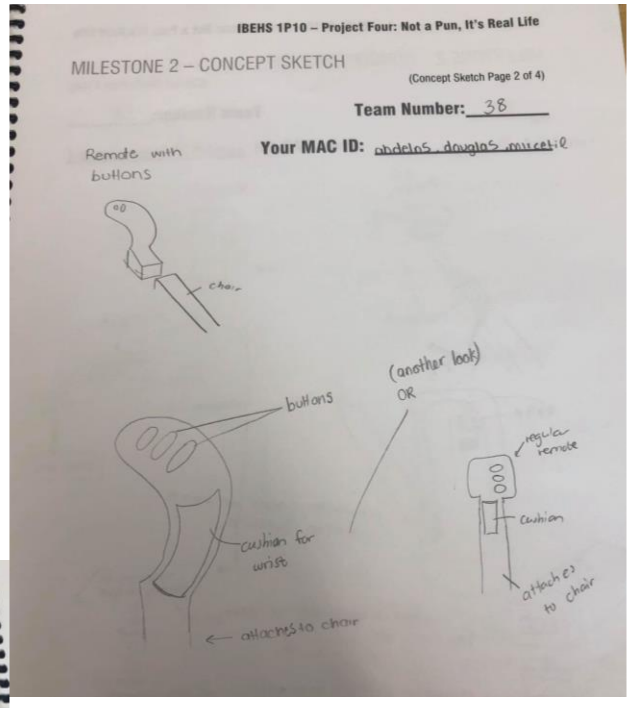
MAC IDs of Present Team Members

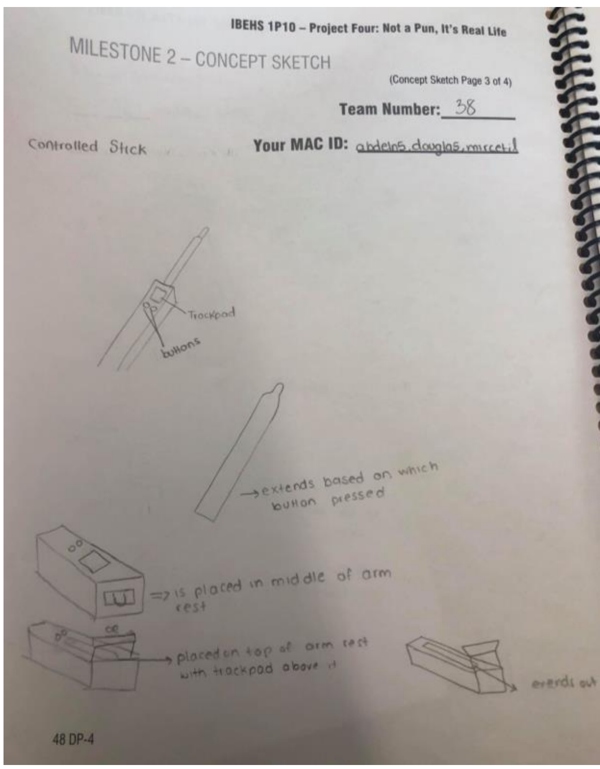
Relationships
•• Strongest = 10
• strong = 7
⊙ fair = 4
○ weak = 1

	weight ↳ grams	portability ↳ physical size; dimensions	button-press ↳ actuation force	compatibility ↳ fits chair ↳ # of devices	Number of steps to set up	Speed / time for fast response	up to user's comfort	water-resistance IPcode	tensile, compressive impact strength	accuracy ↳ triggers response
should be portable	•	••								
should be easy to use			•	•	••					
should be ergonomic				•			••			
should be easy to trigger			••			••				•
should be lightweight	••	•								
should provide fast response			•			••				••
should be weather-proof				•				••	⊙	
should be durable	○								••	
buttons behave independently					○	⊙				••
compatible with chair	⊙	•		••			⊙			
must be triggered when pressed			••							••
must conform to user's disability				•			••			
must be easy to trigger			••			••				•
must trigger door/ elevator				••	○					••
must be pressed			••				••			

Figure 5: Sketches of Proposed Solutions

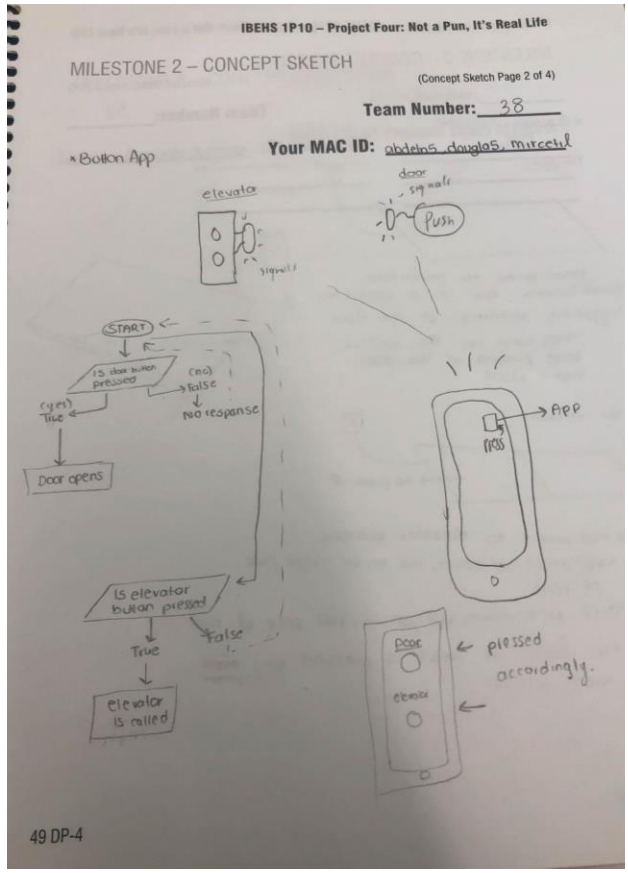
a) Concept Sketch 1: Attachable remote with accessible buttons

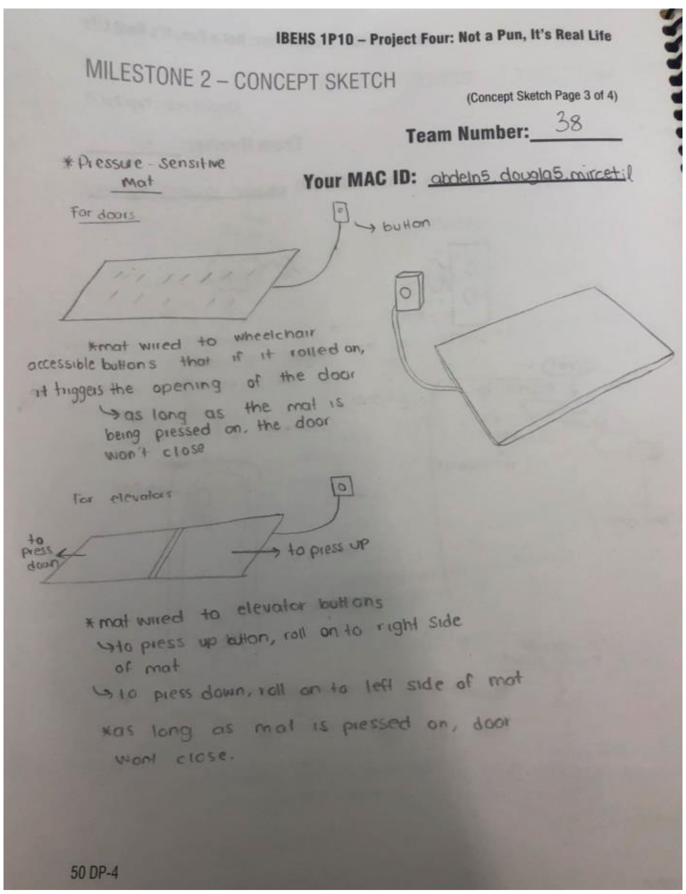




b) Concept Sketch 2: Controlled Stick with trackpad to adjust its movements

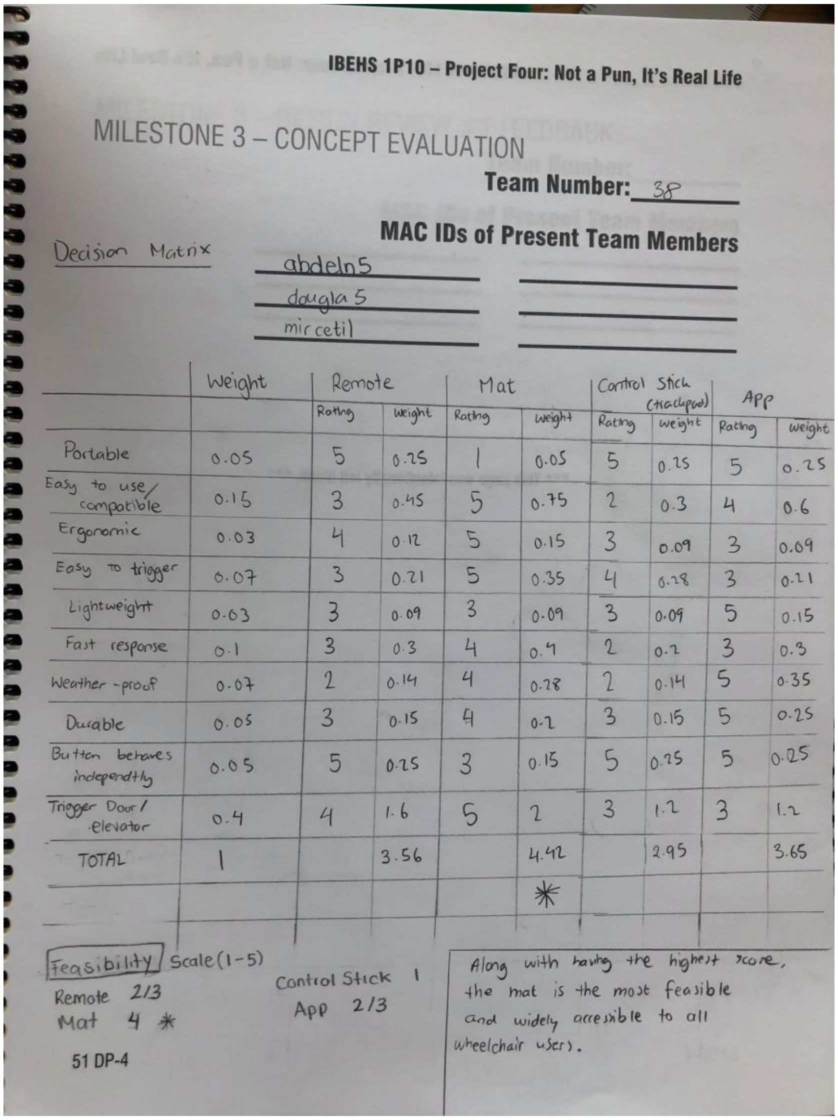
c) Concept Sketch 3: An app to access door and elevator buttons





d) Concept Sketch 4: Pressure Sensitive Mat to mechanically open doors and elevators

Figure 6: Concept Evaluation



IBEHS 1P10 – Project Four: Not a Pun, It's Real Life

MILESTONE 3 – CONCEPT EVALUATION

Team Number: 38

MAC IDs of Present Team Members

Decision Matrix

abdeln5
dougla5
mircetil

	Weight	Remote		Mat		Control Stick (trackpad)		App	
		Rating	weight	Rating	weight	Rating	weight	Rating	weight
Portable	0.05	5	0.25	1	0.05	5	0.25	5	0.25
Easy to use/ compatible	0.15	3	0.45	5	0.75	2	0.3	4	0.6
Ergonomic	0.03	4	0.12	5	0.15	3	0.09	3	0.09
Easy to trigger	0.07	3	0.21	5	0.35	4	0.28	3	0.21
Lightweight	0.03	3	0.09	3	0.09	3	0.09	5	0.15
Fast response	0.1	3	0.3	4	0.4	2	0.2	3	0.3
Weather-proof	0.07	2	0.14	4	0.28	2	0.14	5	0.35
Durable	0.05	3	0.15	4	0.2	3	0.15	5	0.25
Button behaves independtly	0.05	5	0.25	3	0.15	5	0.25	5	0.25
Trigger Door/ elevator	0.4	4	1.6	5	2	3	1.2	3	1.2
TOTAL	1		3.56		4.42		2.95		3.65
					*				

$\boxed{\text{Feasibility}}$ Scale (1-5)

Remote 2/3
Mat 4 *
Control Stick 1
App 2/3

Along with having the highest score, the mat is the most feasible and widely accessible to all wheelchair users.

51 DP-4

Figure 7: First Low-Fidelity Prototype

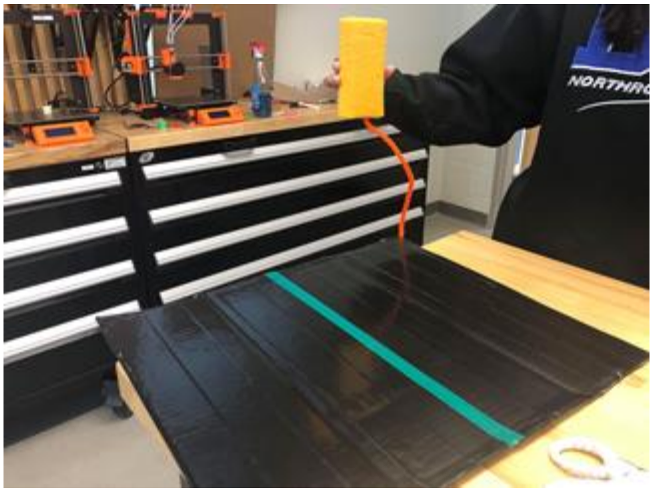
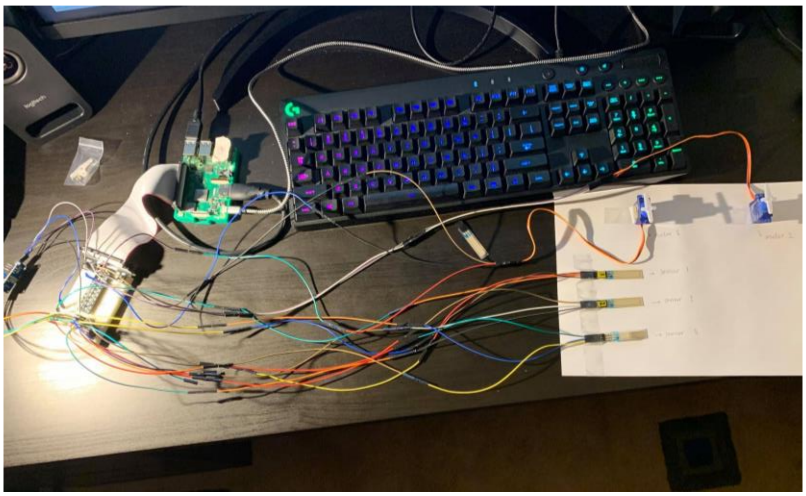


Figure 8: Second Prototype
Using three sensors to control motors



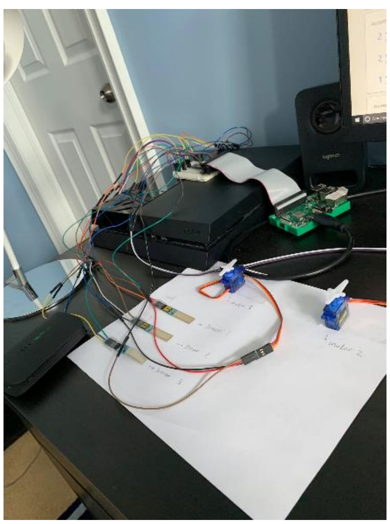
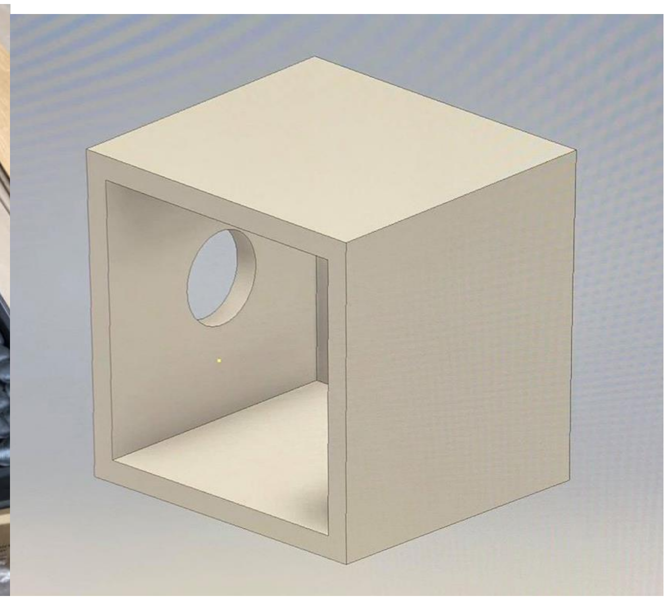


Figure 9: Third Prototype
Using a mat and creating CAD pieces



a) Mat b) Button and Motor casing

c) 3D-printed Casings d) Wire Cover

e) 3D-printed motor component: attaches to motor - to apply force on the button to push it

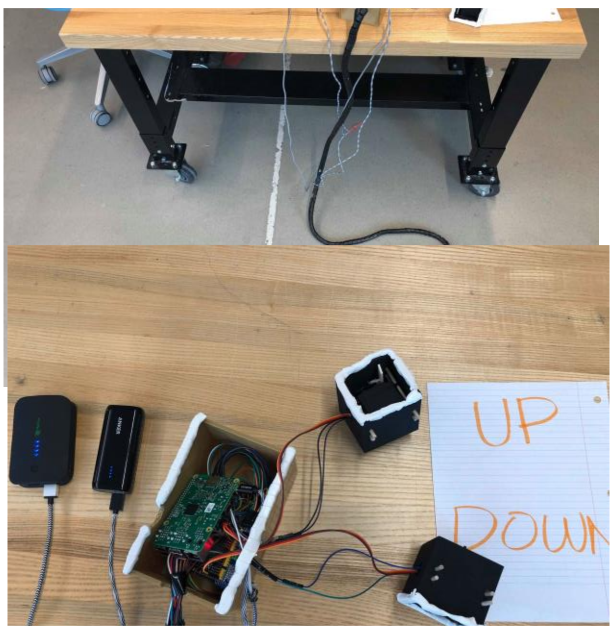
Figure 10: Final High-Fidelity Prototype

a) Front and back views of the mat

b) Cord cover

c) Casings attached to Elevator Buttons

d) Final prototype in its ideal setting



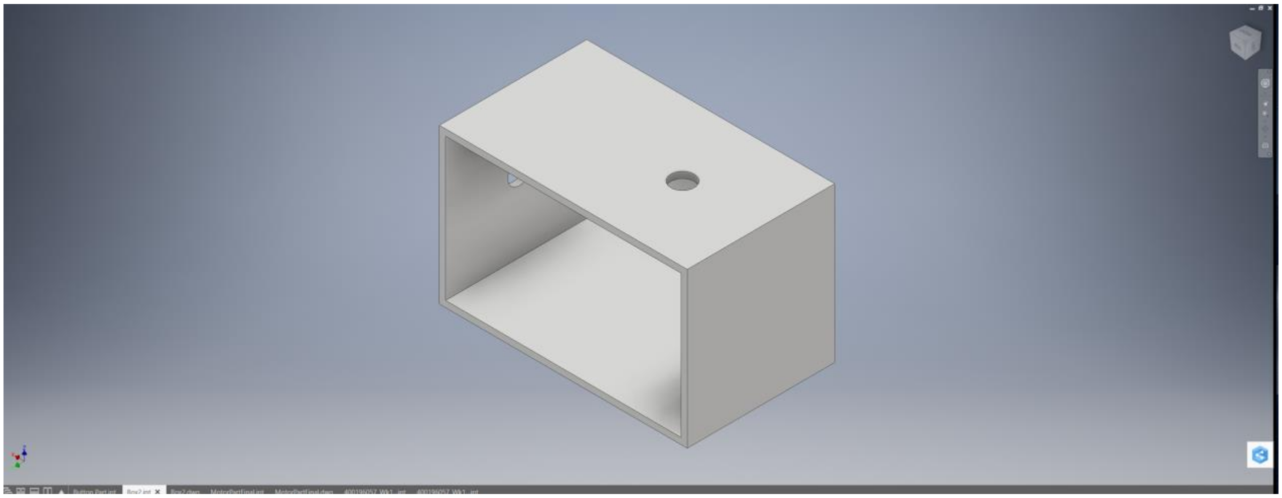
e) Each motor corresponds to one mat

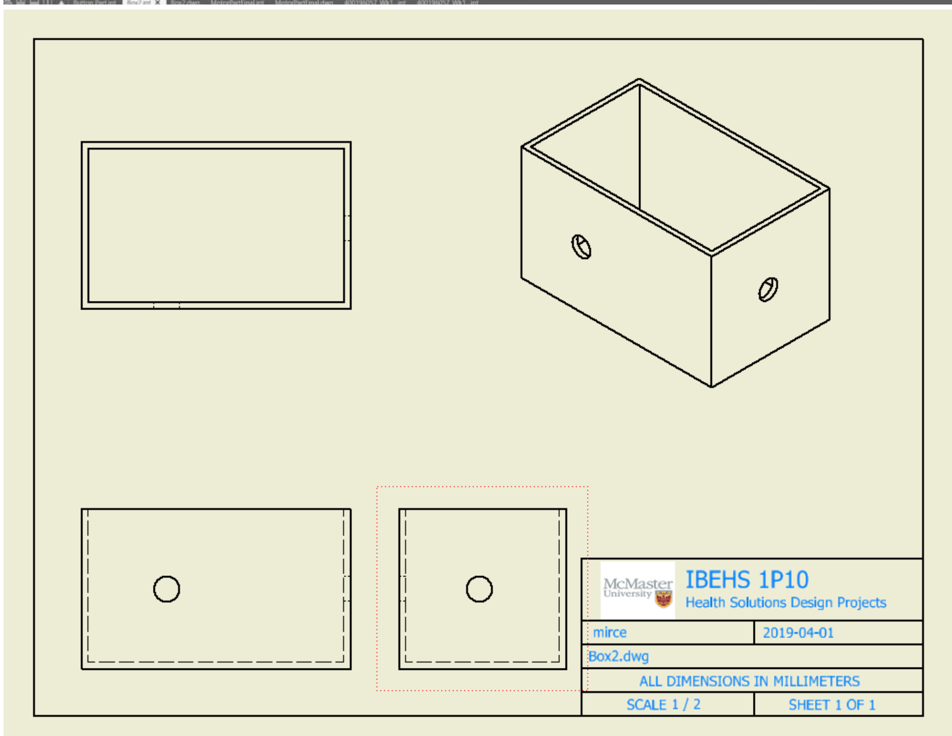
Figure 11: Summary of Design Changes

First Prototype	Second Prototype	Third Prototype	High-Fidelity Prototype
• One large and thin doormat • One line down the middle to distinguish up and down buttons • Can be wireless or mechanical	• Mechanical • Make the mat removable • Thick mat to cushion sensors and prevent fatigue • Smaller mat • Have two separate motors to press buttons	• Have two separate mats rather than one; leave space between them • Have casings around wires and motors to protect from damage and animals if the device is outside • Have a button outside the motor casing to allow other individuals to use the elevator without the mat	• Two mats with sensors underneath • Casings around wires and motors • Have a layer around the actual elevator button • Mechanical motors pushing elevator buttons • Buttons can be triggered by mat or physically pushing the protruded button • Increase number of sensors

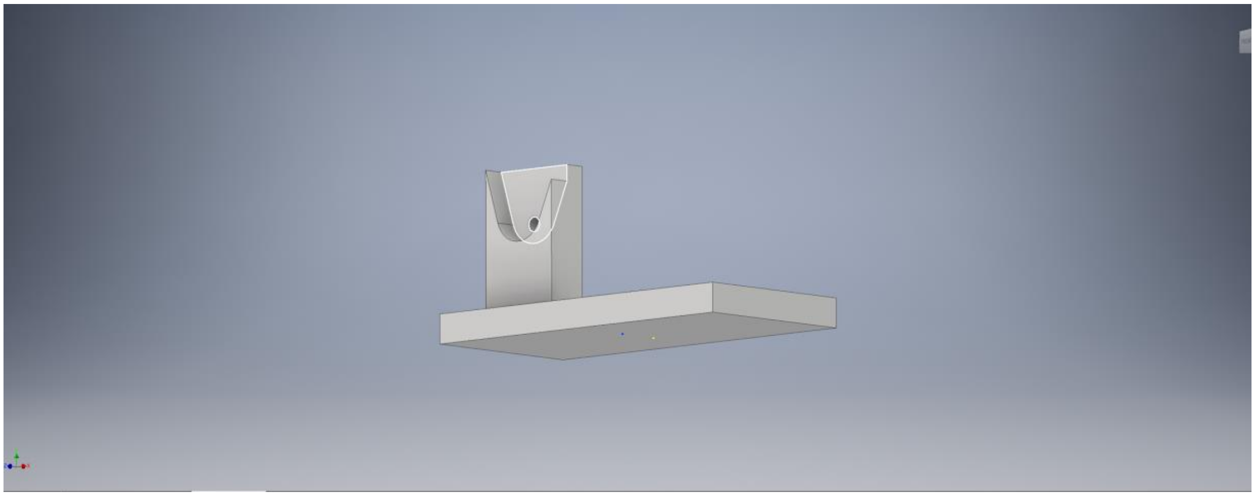
Figure 12: CAD designs

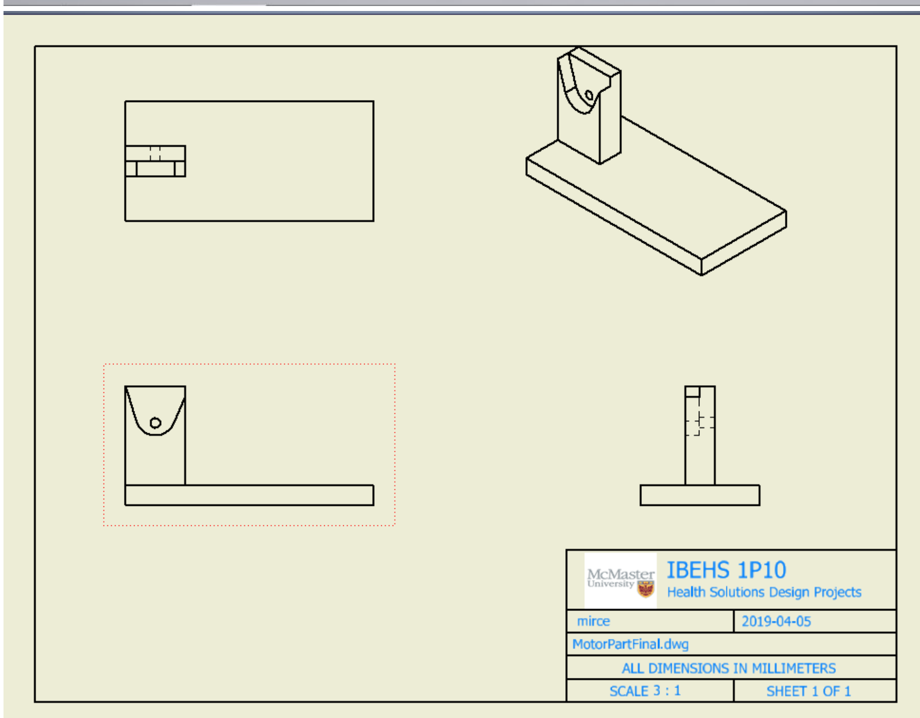
a) Raspberry Pi and PCB housing



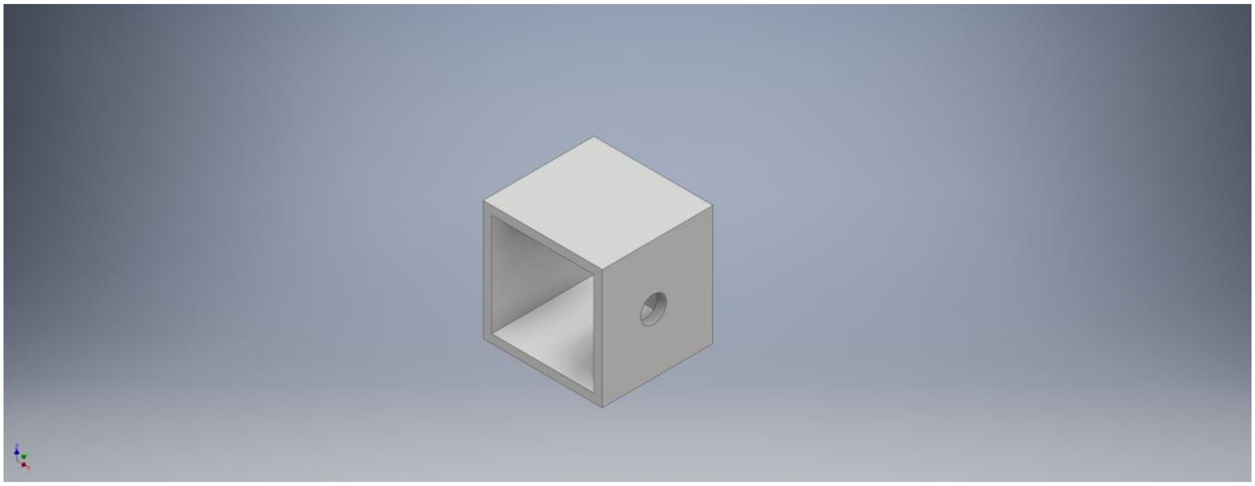


b) Motor Attachment





c) Button and motor casing



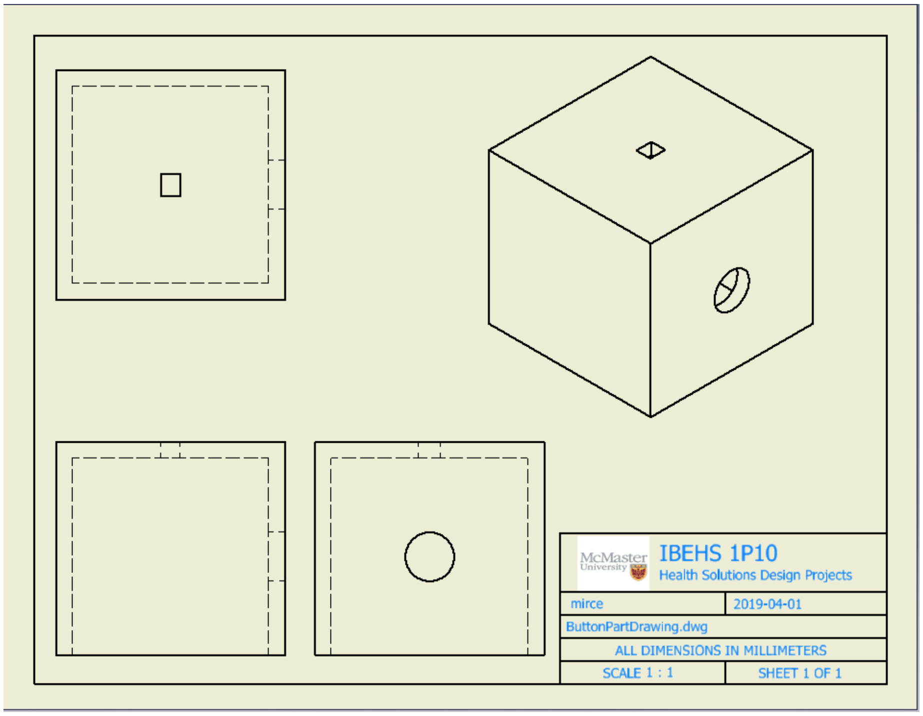


Figure 13: Force and Motor Code

Group38_Prototype_Code.py - C:/Users/mirce/OneDrive/Documents/Documents/Health Design and Solutions 1P10/Design Projects/DP-4/Coding Part/Final Code For Submission/Group38_Prototype_Code.py (3.7.0)

File Edit Format Run Options Window Help

```
#make sure to type sudo pip3 install adafruit-pca9685 into the pi terminal, however if you are using the pi provided it is already installed
#importing anything needed
import PCF8591_3 as ADC
import Adafruit_PCA9685
import RPi.GPIO as GPIO
import smbus
from pcf8591 import PCF8591Expanded
import time
GPIO.setwarnings(False)


#setting up the motor class
pwm = Adafruit_PCA9685.PCA9685()

GPIO.setmode(GPIO.BCM)

#setting up the buttons
GPIO.setup(12, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(19, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

#setting up the ADC converters (using two for this device on the pci expander board
ADC0 = PCF8591Expanded(0)
ADC1 = PCF8591Expanded(1)

#setting the servo rotation points for our device we only want 90 degree turns this is approximately  90
servo_min = 900
servo_max = 100

#setup is defined. Show that the code is working when not connected to a monitor -  so that a monitor is not needed to run it and it boots straight into the code
def setup():
        pwm.set_pwm(0,0,servo_max)
        time.sleep(1)
        pwm.set_pwm(15,0,servo_max)
        time.sleep(1)
        pwm.set_pwm(0,0,servo_min)
        pwm.set_pwm(15,0, servo_min)
        time.sleep(1)

#main loop function is defined where all the code will be run
def loop():
        status = 1
        count = 0

        #creating a list for each sound sensor(there are 6), the list data will be used to calculate the average over time
        data1 = [0,0,0,0,0,0,0]
        data2 = [0,0,0,0,0,0,0]
        data3 = [0,0,0,0,0,0,0]
```

Ln: 1 Col: 140

Group38_Prototype_Code.py - C:/Users/mirce/OneDrive/Documents/Documents/Health Design and Solutions 1P10/Design Projects/DP-4/Coding Part/Final Code For Submission/Group38_Prototype_Code.py (3.7.0)

File Edit Format Run Options Window Help

```
        #creating a list for each sound sensor(there are 6), the list data will be used to calculate the average over time
        data1 = [0,0,0,0,0,0,0]
        data2 = [0,0,0,0,0,0,0]
        data3 = [0,0,0,0,0,0,0]
        data4= [0,0,0,0,0,0,0]
        data5= [0,0,0,0,0,0,0]
        data6= [0,0,0,0,0,0,0]

        #creating a variable which will be the sum of data in the list above
        sum_data1 = 0
        sum_data2 = 0
        sum_data3 = 0
        sum_data4 = 0
        sum_data5 = 0
        sum_data6 = 0

        #defining a variable for the average of data
        avg_data1 = 0
        avg_data2 = 0
        avg_data3 = 0
        avg_data4 = 0
        avg_data5 = 0
        avg_data6 = 0

        #while true infinite loop is created which will take sensor reading, find an average over time and then move the motor if it passes a certain value
        while True:
                #assigning variables to each force sensor - for reference, value, value2, value3 are all on the "up" mat and the other three are on the "down" mat
                value = float(ADC1.get_channel(1))
                value2 = float(ADC0.get_channel(3))
                value3 = float(ADC0.get_channel(1))
                value4 = float(ADC0.get_channel(0))
                value5 = float(ADC0.get_channel(2))
                value6 = float(ADC1.get_channel(0))

                #printing each of the values (this part of the code is more for troubleshooting but is not necessary for it to run properly)
                print ('Value1: ', value)
                print ('Value2: ', value2)
                print ('Value3: ', value3)
                print ('Value4: ', value4)
                print ('Value5: ', value5)
                print ('Value6: ', value6)
                print('******************************************')

                #creating variables for up and down buttons
                upbutton = GPIO.input(12)
                downbutton = GPIO.input(19)
```

Ln: 1 Col: 140

Group38_Prototype_Code.py - C:/Users/mirce/OneDrive/Documents/Documents/Health Design and Solutions 1P10/Design Projects/DP-4/Coding Part/Final Code For Submission/Group38_Prototype_Code.py (3.7.0)

File Edit Format Run Options Window Help

```python
        #creating variables for up and down buttons
        upbutton = GPIO.input(12)
        downbutton = GPIO.input(19)

        #another print statement for troubleshooting and seeing if button press is registering
        if upbutton == GPIO.HIGH:
            print ("UP button was pushed!")
        elif downbutton == GPIO.HIGH:
            print ("DOWN button was pushed")

        #if statement for each value is made so that if the value is greater than zero (aka the force sensor is being pressed then it will do the following:
        #the value will be appended to the list, the sum of the last six values in the list will be computed and then the average will be computed
        if value >= 0:
            data1.append(value)
            sum_data1 = data1[-1]+data1[-2]+data1[-3]+data1[-4]+data1[-5]+data1[-6]
            avg_data1 = sum_data1/6
        if value2 >= 0:
            data2.append(value2)
            sum_data2 = data2[-1]+data2[-2]+data2[-3]+data2[-4]+data2[-5]+data2[-6]
            avg_data2 = sum_data2/6
        if value3 >= 0:
            data3.append(value3)
            sum_data3 = data3[-1]+data3[-2]+data3[-3]+data3[-4]+data3[-5]+data3[-6]
            avg_data3 = sum_data3/6
        if value4 >= 0:
            data4.append(value4)
            sum_data4 = data4[-1]+data4[-2]+data4[-3]+data4[-4]+data4[-5]+data4[-6]
            avg_data4 = sum_data4/6
        if value5 >= 0:
            data5.append(value5)
            sum_data5 = data5[-1]+data5[-2]+data5[-3]+data5[-4]+data5[-5]+data5[-6]
            avg_data5 = sum_data5/6
        if value6 >= 0:
            data6.append(value6)
            sum_data6 = data6[-1]+data6[-2]+data6[-3]+data6[-4]+data6[-5]+data6[-6]
            avg_data6 = sum_data6/6


        #if the average values for sensor 1,2 are greater than 200 (a threshold we determined through trial and error)
        #or average for 3 is greater than 230(because this sensor was broken) or the up button is pushed, the up motor will turn 90 degrees
        if avg_data1 > 200 or avg_data2 > 200 or avg_data3 > 230 or upbutton == GPIO.HIGH:
            pwm.set_pwm(0,0,servo_max)
            time.sleep(0.5)


        #same for sensors 4,5,6 and the down button. They will move the other 'down' motor
```

Ln: 1 Col: 140

Group38_Prototype_Code.py - C:/Users/mirce/OneDrive/Documents/Documents/Health Design and Solutions 1P10/Design Projects/DP-4/Coding Part/Final Code For Submission/Group38_Prototype_Code.py (3.7.0)

File Edit Format Run Options Window Help

```python
            sum_data3 = data3[-1]+data3[-2]+data3[-3]+data3[-4]+data3[-5]+data3[-6]
            avg_data3 = sum_data3/6
        if value4 >= 0:
            data4.append(value4)
            sum_data4 = data4[-1]+data4[-2]+data4[-3]+data4[-4]+data4[-5]+data4[-6]
            avg_data4 = sum_data4/6
        if value5 >= 0:
            data5.append(value5)
            sum_data5 = data5[-1]+data5[-2]+data5[-3]+data5[-4]+data5[-5]+data5[-6]
            avg_data5 = sum_data5/6
        if value6 >= 0:
            data6.append(value6)
            sum_data6 = data6[-1]+data6[-2]+data6[-3]+data6[-4]+data6[-5]+data6[-6]
            avg_data6 = sum_data6/6


        #if the average values for sensor 1,2 are greater than 200 (a threshold we determined through trial and error)
        #or average for 3 is greater than 230(because this sensor was broken) or the up button is pushed, the up motor will turn 90 degrees
        if avg_data1 > 200 or avg_data2 > 200 or avg_data3 > 230 or upbutton == GPIO.HIGH:
            pwm.set_pwm(0,0,servo_max)
            time.sleep(0.5)


        #same for sensors 4,5,6 and the down button. They will move the other 'down' motor
```


```python
        elif avg_data4>200 or avg_data5 >= 200  or avg_data6 > 200 or downbutton == GPIO.HIGH:
            pwm.set_pwm(15,0,servo_max)
            time.sleep(0.5)


        #after being pressed or if not pressed. Both motors return to regular position (initial position)
        else:

            pwm.set_pwm(0,0,servo_min)
            pwm.set_pwm(15,0, servo_min)
            time.sleep(0.2)
        time.sleep(0.2)
#the setup and loop function are run, and keyboard interrupt is added
if __name__ == '__main__':
    try:
        setup()
        loop()
    except KeyboardInterrupt:
        p.stop()
        GPIO.cleanup()
        pass
```

Ln: 1 Col: 140

Figure 14: Project Schedule

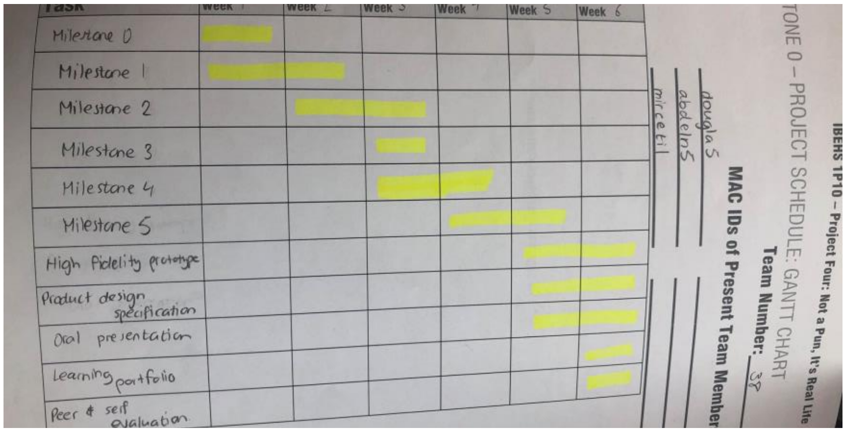


Figure 15: Computing Prototype Pictures

a) Before(PCB with the Multiplexer)   b) After (all soldering complete)