Design Document

Brief Overview

To enable our visually impaired client to operate a wheelchair independently, we propose an adaptable attachment mounted to the back of the wheelchair, which collects obstacle proximity data with a LiDAR sensor and guides the user with haptic feedback. Additionally, we provide a second layer of security for the client through the employment of sonar sensors attached to the lower chassis of the wheelchair.

The LiDAR scanner collects omnidirectional proximity data by spinning at ~5.5Hz and utilizing laser triangulation to gather over 8000 distance readings per second. Though it is precise and capable of detecting surfaces up to 12 meters away, it only collects points in a single, two dimensional plane of space. Thus, shorter obstacles which lie below the sensor's plane cannot be detected. To address this limitation, we design a tilting mechanism for the scanner (Figures 3-6), allowing it to assess multiple planes at different degree tilt intervals. This allows for shorter obstacles to be detected, increasing safety for the user.

As an additional source of protection, sonar sensors will be employed near the bottom of the wheelchair chassis to detect drops. Although sonar sensors have a low range and are less precise than LiDAR, using them to detect drops is feasible due to close proximity to the ground and the larger size of the "object," given the comparative size of the floor compared to a traditional stationary object. The use of sonar allows us to not only detect additional obstacles, but also deliver our product at a lower cost, due to sonar's affordability over other sensors.

Note: When referencing designs, design 1 indicates the camera-based model, design 2 indicates the sonar based model, and design 3 indicates the LiDAR model.

Design #3 - Attachment and Tilting Mechanism for LiDAR sensor

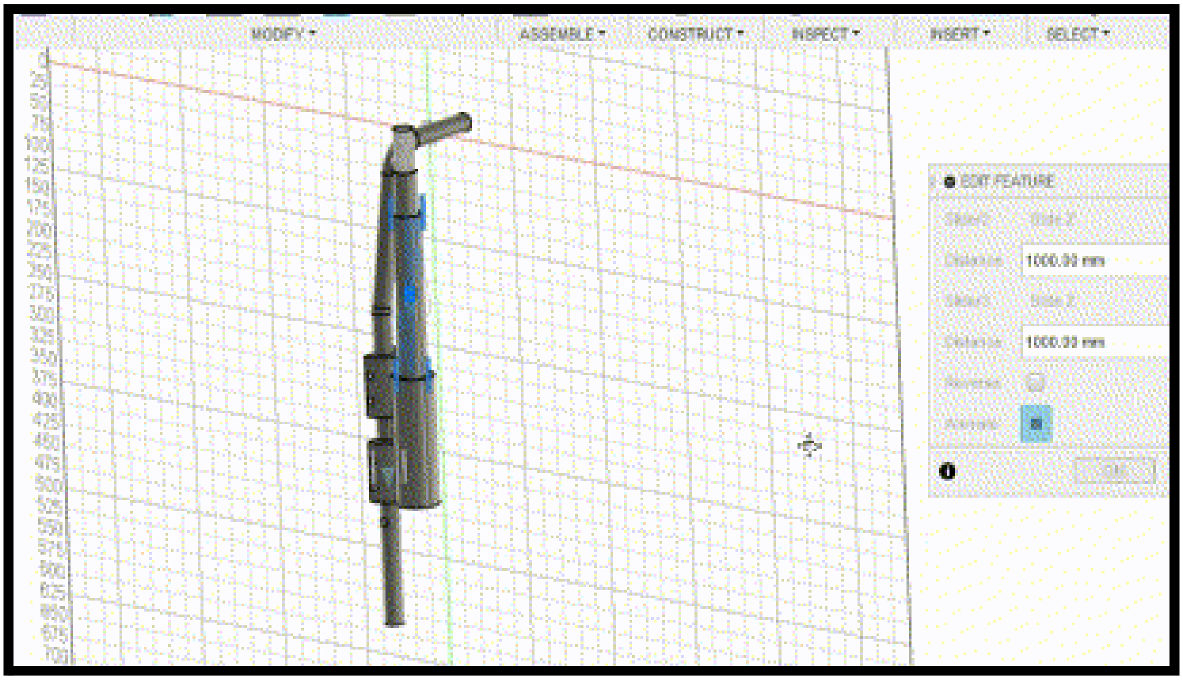

Figure 1

Method for Attaching LiDAR Sensor to the wheelchair to scan the environment above the user. Snaps over the bar connecting to the side of the wheel chairs seat (bolt tightened), has a tripod-like extension, threaded to tighten LiDAR attatchment. Future extensions include adapting the base of this attachment to a preexisting extendable pole, and investigating the dimensional accuracy of this design with a wheelchair.

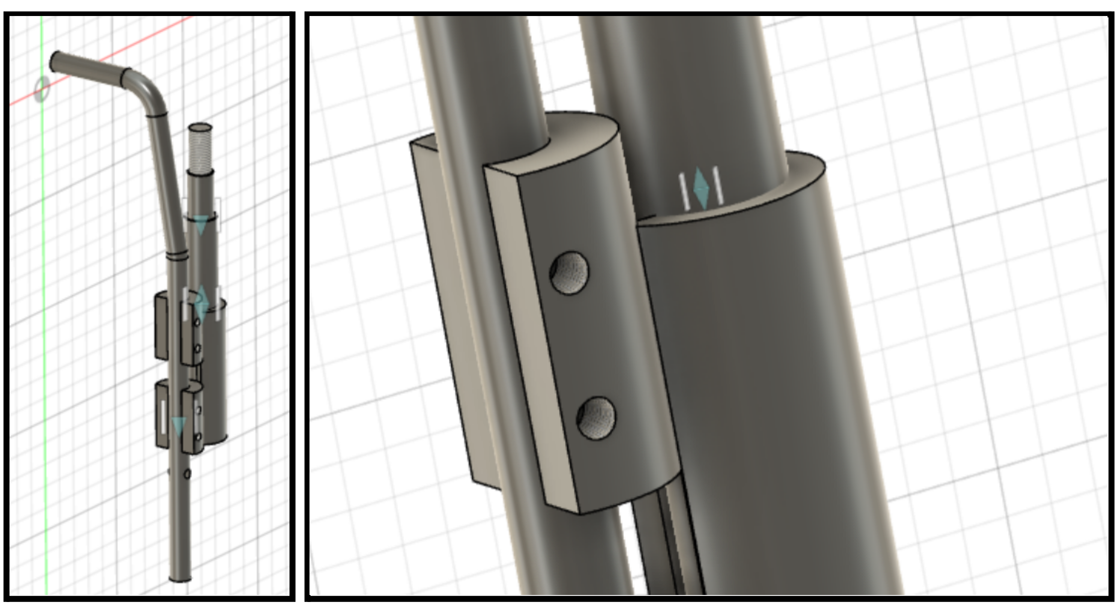

Figure 2

Close up on the attachment sensor.

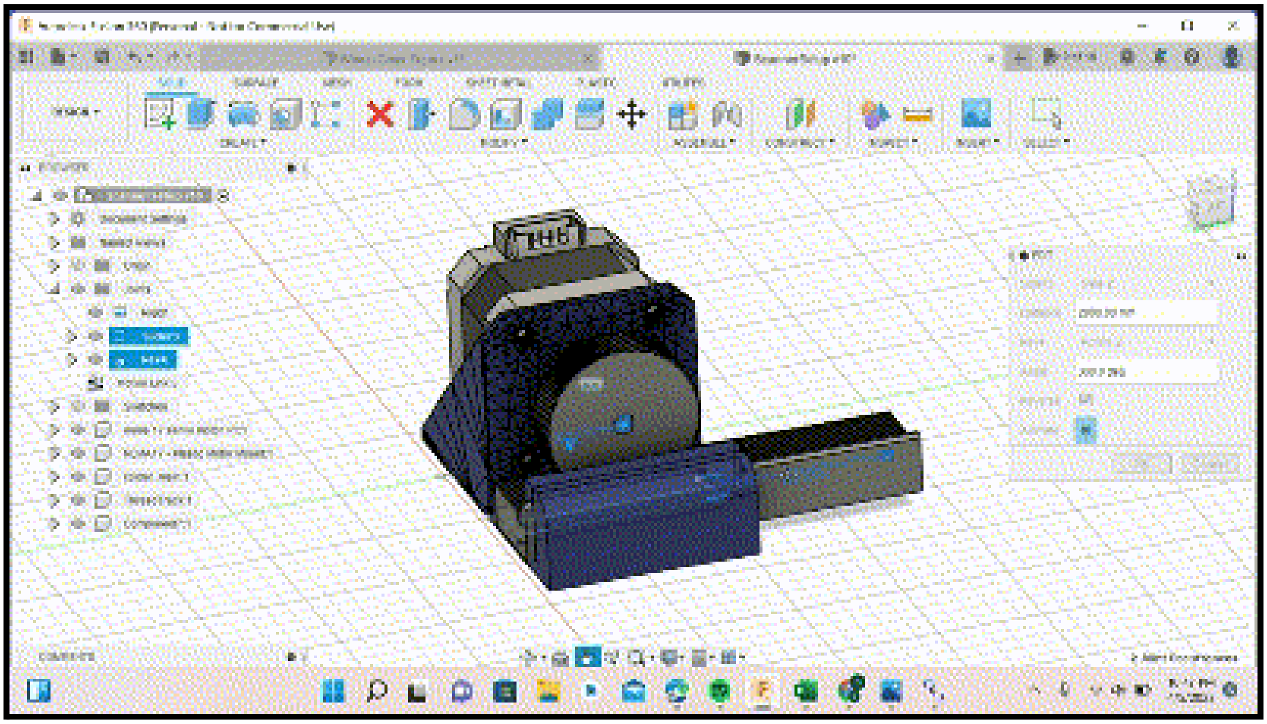

Figure 3

Tilt Assembly Part 1

Controlled Motor Driven "Linear Actuator" with a Mounted Nema-17 Stepper Motor

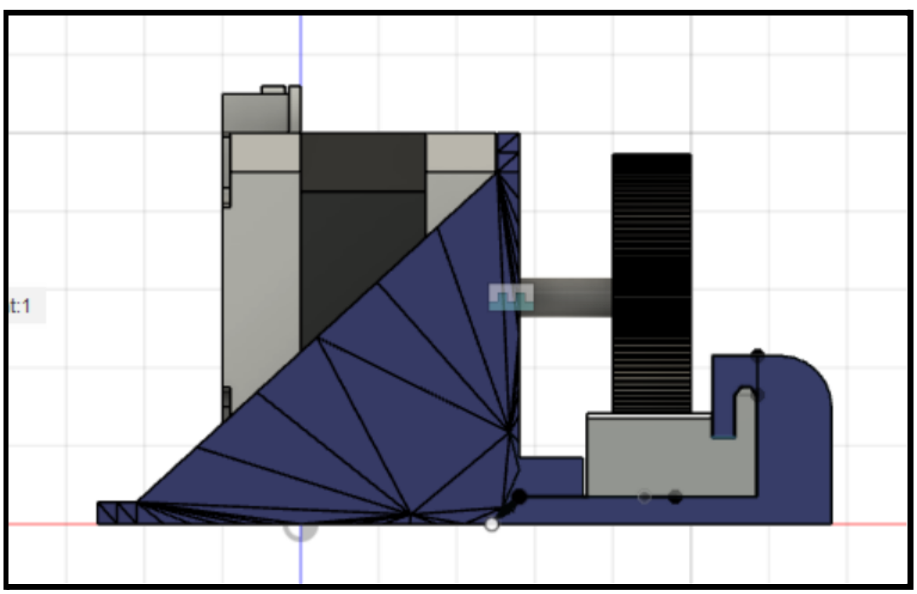

Figure 4

Side view of Tilt Assembly Part 1

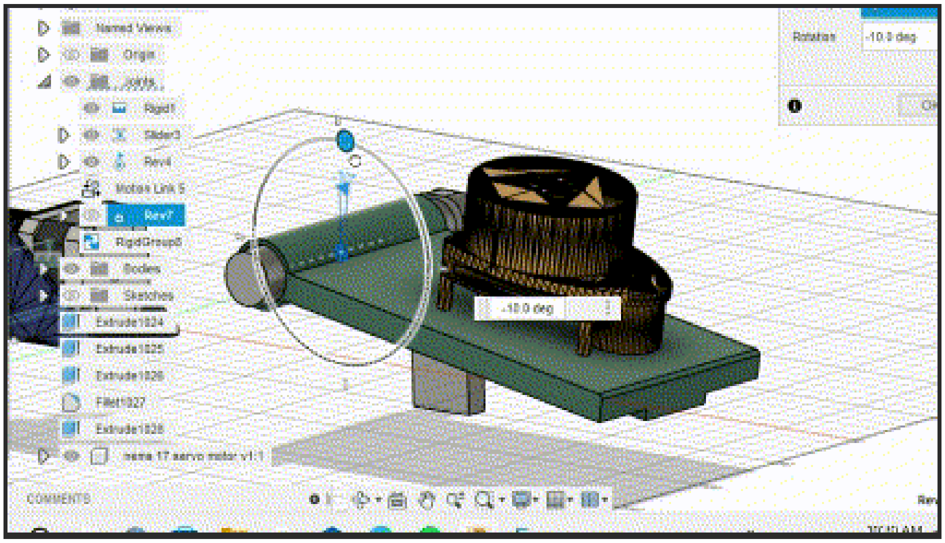

Figure 5

Tilt Assembly Part 2

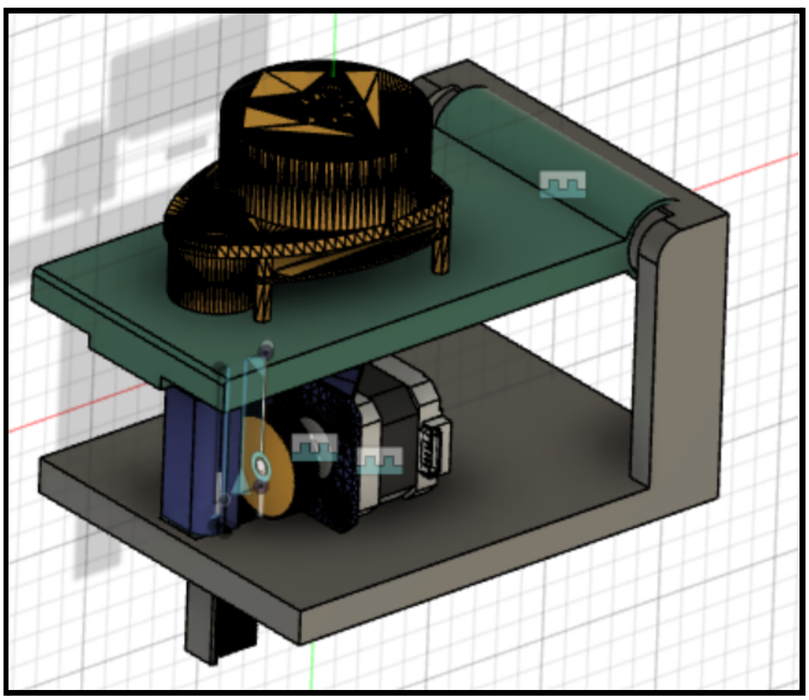

Figure 6

Exposed Side view of Tilt Assembly Part 2

Future extensions include adding the surrounding casing, introducing slots for the attachable Lidar sensor on the lever's top, and adding a threaded slot beneath the motor and the platform to attach to the design that is depicted further above.

Design #3 - Stepper Motor Code

This code was successful and allowed us to move the stepper motor. The stepper motor was able to rotate at a specified rate and is precise, though we are unsure if the motor will be strong enough to turn the sensor on its own and will adjust during further testing.

```
stepper_test §
// Define pin connections & motor's steps per revolution
const int dirPin = 2;
const int stepPin = 3;
const int stepsPerRevolution = 200;

void setup()
{
  // Declare pins as Outputs
  pinMode(stepPin, OUTPUT);
  pinMode(dirPin, OUTPUT);
}
void loop()
{
  // Set motor direction counterclockwise
  digitalWrite(dirPin, HIGH);

  // Spin motor slowly
  for(int x = 0; x < stepsPerRevolution; x++)
  {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(2500/5);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(2500/5);
  }
  //delay(1000); // Wait a second

  // Set motor direction counterclockwise
  //digitalWrite(dirPin, LOW);

  // Spin motor quickly
//  for(int x = 0; x < stepsPerRevolution; x++)
//  {
//    digitalWrite(stepPin, HIGH);
//    delayMicroseconds(1000);
//    digitalWrite(stepPin, LOW);
//    delayMicroseconds(1000);
//  }
//  //delay(1000); // Wait a second
}
```

Figure 7

C++ Code for Stepper Motor to rotate at 5Hz

Future extensions include determining optimal rotational frequency, as well as setting up a system for delayed rotation to allow for a full 360º scan before shifting planes.

Design #2 - Object Detection using the Sonar Sensor

We also experimented with a Sonar sensor. This sensor was successful in delivering precise and interpretable data at a high sample rate—40 Hz maximum. The sonar sensor is accurate at up to four meters and may be used to detect immediate obstacles in the wheelchair's path. However, it is only capable of collecting points in one dimension. Obtaining a full field of vision around the wheelchair would be unfeasible with this design, as it would require a number of sensors that exceeds the number of ports on our processor. Furthermore, as shown in Figure 9, the sensor occasionally collects larger-than-expected noise, which may confound a rudimentary obstacle detection algorithm. It is, however, possible to identify these anomalies and exclude them from the data.

sketch_apr05a §

```
/*
 * HC-SR04 example sketch
 *
 * https://create.arduino.cc/projecthub/Isaac100/getting-started-with-the-hc-sr04-ultrasonic-sensor-036380
 *
 * by Isaac100
 */

const int trigPin = 3;
const int echoPin = 2;

float duration, distance;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = (duration*.0343)/2;
  Serial.print("Distance: ");
  Serial.println(distance);
  delay(100);
}
```

Figure 8

Code for sonar sensor

The sensor is capable of ranges from 2 cm - 4.5 m. Future extensions include accessing the raw data of one-dimensional distances to perform the required drop-off checks

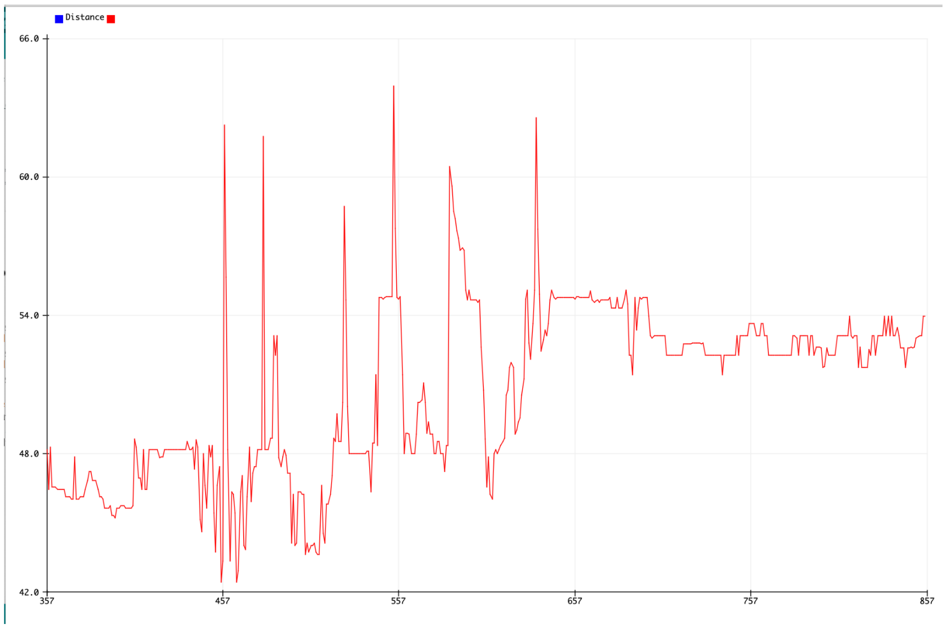

Figure 9

Graphed data from the Sonar Sensor

This depicts distance data between 42 and 64 cm away from the sensor.

```
10:53:42.750 -> Distance: 22.04
10:53:42.890 -> Distance: 20.55
10:53:42.965 -> Distance: 20.13
10:53:43.076 -> Distance: 20.56
10:53:43.186 -> Distance: 21.15
10:53:43.261 -> Distance: 23.19
10:53:43.371 -> Distance: 22.31
10:53:43.481 -> Distance: 27.25
10:53:43.592 -> Distance: 27.71
10:53:43.703 -> Distance: 28.50
10:53:43.816 -> Distance: 26.27
10:53:43.890 -> Distance: 26.67
10:53:44.001 -> Distance: 24.92
10:53:44.110 -> Distance: 23.96
10:53:44.223 -> Distance: 24.94
10:53:44.335 -> Distance: 28.86
10:53:44.404 -> Distance: 28.25
10:53:44.514 -> Distance: 25.11
10:53:44.626 -> Distance: 35.86
10:53:44.739 -> Distance: 34.71
10:53:44.850 -> Distance: 28.25
10:53:44.959 -> Distance: 28.50
```

```
10:53:45.035 -> Distance: 25.71
10:53:45.147 -> Distance: 28.78
10:53:45.260 -> Distance: 27.25
10:53:45.373 -> Distance: 48.48
10:53:45.448 -> Distance: 35.64
10:53:45.559 -> Distance: 28.61
10:53:45.673 -> Distance: 28.61
10:53:45.784 -> Distance: 35.64
10:53:45.893 -> Distance: 43.37
10:53:46.002 -> Distance: 28.88
10:53:46.079 -> Distance: 26.27
10:53:46.189 -> Distance: 44.68
10:53:46.298 -> Distance: 22.33
10:53:46.409 -> Distance: 19.17
10:53:46.520 -> Distance: 17.03
10:53:46.631 -> Distance: 14.15
10:53:46.702 -> Distance: 9.98
10:53:46.813 -> Distance: 9.19
10:53:46.920 -> Distance: 6.16
10:53:47.032 -> Distance: 4.49
10:53:47.143 -> Distance: 3.21
10:53:47.215 -> Distance: 3.89
```

Figure 10

Raw distance data from sonar system

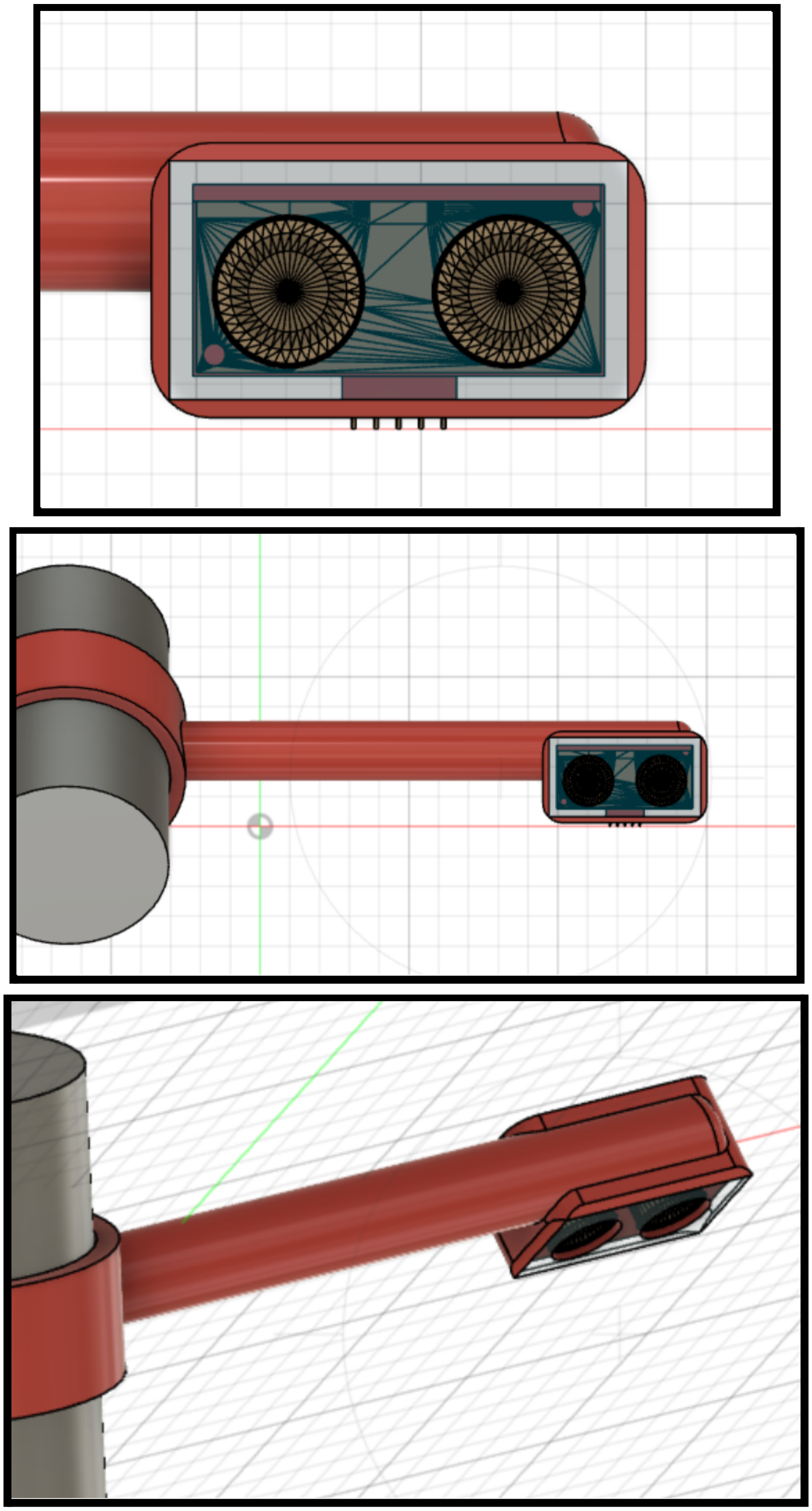

Figure 11

Attachment of the Hc-sr04 Ultrasonic Sensor in a casing snap mounted and extending half a foot beyond the back rail of the wheelchair

Future extensions include determining optimal length and degree tilt for sensing drop-offs or objects, which depicted here is 40 degrees.

Designs 1, 2, and 3 - Wrist-based Vibration Communication

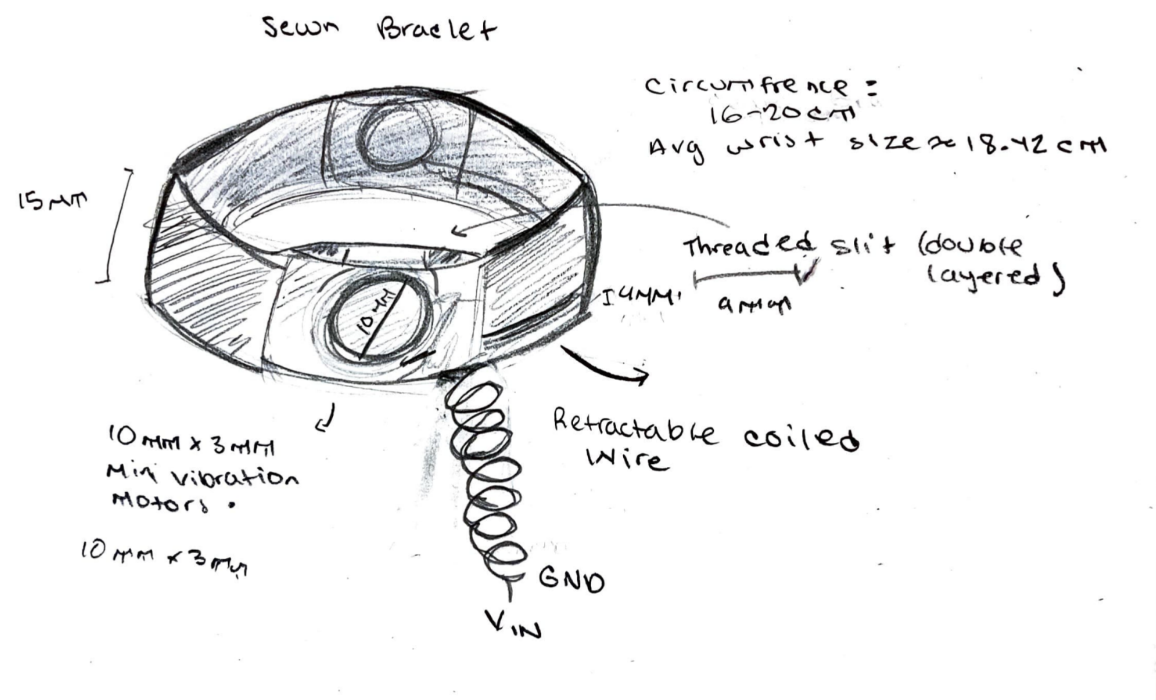

Figure 12
User communication system design
This illustrates the bracelet design, where information is relayed to the user using mini 3V DC motors to provide vibrations at the wrist.

Timeline

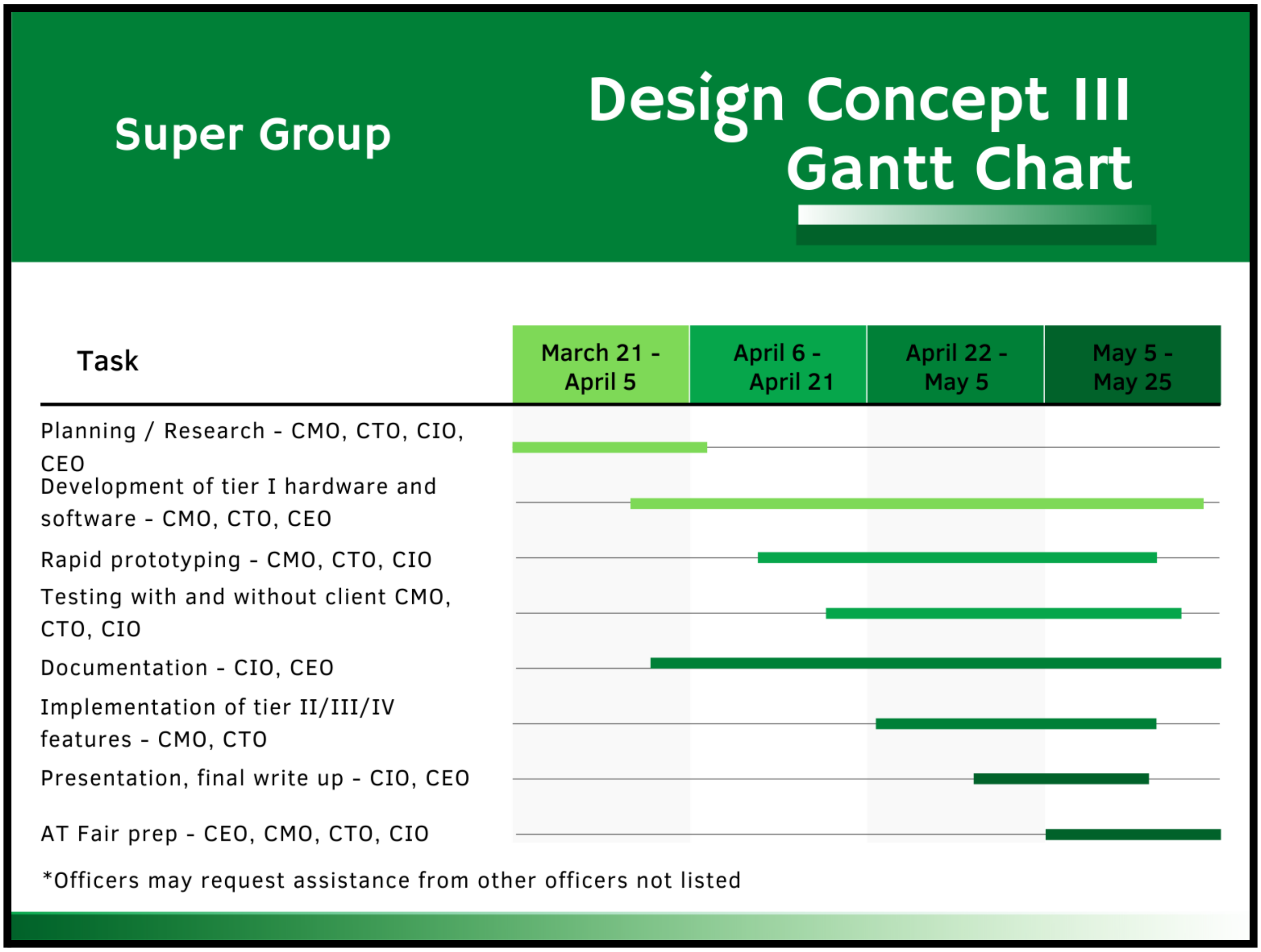

Figure 13

Gantt Chart detailing the project timeline

As we are moving forward with a modified version of design #3, we are not changing the timeline.

Materials List

Table 1 *Comprehensive materials list, detailing cost.*

Item	**Supplier**	**Catalog #**	**Quantity**	**Unit Cost**	**Total**
Slamtec RPLiDAR A1	Already in possession	-	1	-	-
HC-SR04 Ultrasonic Sensor	Already in possession	-	3	-	-
Raspberry Pi Model B+	Already in possession	-	1	-	-
Wires, Sauter, Electrical Tape, 3d printing filament	Already in Possession	-	1	-	-
Mini Vibration Motors DC 3V	Tatoko	-	1 pack (20 pcs)	$14.99	$14.99
Digital Servo Motor	Already in Possession	-	1 set(4-pack)	-	-
20800mAh Compact Rechargeable Battery for Raspberry Pi	PiShop.us	-	1	$24.95	$24.95
Buzzer Module	Tatoko	-	1 Pack (10pcs)	$10.69	$10.69
Arduino Uno	Already in Possession	-	5	-	-
Raspberry Pi 4 *Analyzing the necessity over the Model B+*	Soon to be in Possession (Borrowed)	-	1	-	-
					50.63