Introduction to Engineering Design with Professional Development 1

Final Report for

Seeing-Eye Robot

Team: Undecided

Section 1

Instructors: Christine Allard, Karthik Paneerselvam

Version 1.0

August, 2024

Prepared by

Leighton Hesch (2026 / Computer and Systems Engineering)

Matthew McCall (2026 / Computer Systems Engineering)

Gabriel Sabin (2026 / Computer Systems Engineering)

Alan Lee (2026 / Computer Systems Engineering)

Ben Stanley (2026 / Mechanical Engineering)

Victor Garcia-Gallet (2026 / Environmental Engineering & Hydrogeology)

Executive Summary

This project aims to answer the navigational needs of the visually impaired by offering an affordable alternative to guide dogs. The motivation behind this is that guide dogs are very expensive to take care of, with average total prices over the dog's lifespan ranging from $40,000 to $60,000. The project is not trying to directly replace guide dogs, as they are capable of actions too advanced for a robot, but more so provide a wider range of visually impaired people with the opportunity to travel places they otherwise would not be able to.

There were multiple designs that were initially considered for the project. They all had positives and negatives for the user, but it was ultimately decided that creating a miniature robot to guide the user would be the most beneficial. After, the project would have to be broken down into multiple parts, that would be ultimately put together. These included the sensing system, the pathfinding system, drivetrain control, drivetrain electronics, the leash, and the chassis. All these had their own selection matrix that decided on what would be best for the system. However, some of these systems would face some difficulties due to the chosen method from the selection matrix, changing the implementation of the subsystem. However, at the end, every subsystem was able to integrate with one another.

The resulting prototype did many things successfully, but also had many flaws. The robot was able to read mesh data and examine its surroundings quite well, but it did not use the user's phone in realtime. It was also very capable of pathfinding and appropriately responding to obstacles by shifting its motors. However, turning was not implemented so the user would have to sidestep when changing directions. The leash was comfortable to hold and offered a reasonable range of adjustability, but the connection was not assembled very well and was not as robust as the team would have liked it to be. Additionally, while the chassis was durable and portable, it offered no protection for the electronics. The final design would improve on all of these issues by offering proper usage of the user's phone, turning, a more robust leash, and a more protective and versatile chassis. It would also include additionally sensory information, like haptic feedback in the leash and audio cues to warn of conditions the robot is not able to traverse.

Table of Contents

1 Introduction

While society can attempt to make accommodations for disabled people, there will always be a clear bias in making the world comfortable for able-bodied individuals. Often it can fall on the disabled to find a way to navigate a world that takes certain things for granted. However, assistance options are extremely niche and therefore expensive. Thus the team wanted to see if alternative designs for aiding products were possible to make cheaper.

The team's target was the seeing-eye dog. Between their training and constant care, a blind person is faced with significant burdens of both time and money. The most obvious issue is that a dog is a living being, which means it has production and user end constraints. However, this is also its biggest benefit, as it's able to adapt to their owner and the owner's needs better than any product the team could build.

This narrowed down the goal of the design. The attempt was made to design a guide that could replace many functionality of a dog, though with the constraint that it was never meant to replace a dog. This limit on scope allowed for the goal of an inexpensive solution to be easily realized.

The product would be a robot that sensed obstacles around it, and, after the user had set a destination, would create and follow a safe path to guide the user along. The user would interface with the robot by leash, similar to that of a seeing-eye dog's leash, with a controller on the end, to allow user choice in how to proceed. In turn, the robot would communicate to the user through its movement, sound cues, and haptics in the controller.

This document will discuss how this project was decided, and the specific design choices made in building the prototype. Each subsystem will be discussed, and how they each contributed to the user experience, and the team's goal of providing an alternative solution to a seeing-eye dog.

2 Project Objectives & Scope

Table 2.1 - List of objectives with corresponding scopes/boundaries

Project Objectives	In Scope	Out of Scope
Develop a functional Seeing Eye Robot	Create a prototype with the main functionalities of a Seeing Eye Robot	Improving the production by mass producing the Seeing Eye Robot
Implement obstacle recognition and avoidance	Install a LIDAR camera to the robot and develop pathfinding	Making the robot autonomous with low reliance on other technology
Ensure capability to navigate different terrain and weather conditions	Develop a robot capable of traveling on different indoor floor materials	Improving the robot to navigate terrains outdoors and handle different weather conditions
Designing a simple control system that can easily be learned by the user	A handheld device with buttons to allow for simple user control	Include haptics and other forms of communication for the user
Create a power source that allows the robot to operate for long periods of time	Connecting alkaline batteries to produce enough power for robot	Use lithium or rechargeable batteries to allow for longevity

2.1 Mission Statement

To create an alternative product that alleviates the burden of a seeing-eye dog's cost and care, while creating a market for robot aides for the blind.

2.2 Customer Requirements

The team was able to come up with some basic, sensible requirements on their own. However, as none of the team is visually impaired, there were some requirements that were not realized very clearly. To achieve this clarity, a professional in this field was interviewed: Jarnail Chudge. He worked for Microsoft's Soundscape project, so he is very familiar with designing solutions for the problems that the visually impaired face. His most useful advice was on providing the user freedom: the focus of the robot should be less to take the user from point A to point B, but to provide information about the environment so that the user can effectively travel from point A to point B.

Table 2.2 - List of customer requirements with an importance ranking of 1 to 5 (least to most)

Customer Requirements	Importance
Can find a clear route between two locations	5
Be aware of surroundings both moving and stationary	5
Will be able to walk for as long as a dog	3
Can find alternative routes	4
Product weight is kept to a minimum	3
Designed with handles or other architecture to facilitate being picked up	2
Locomotion that won't get stuck and work on a variety of surfaces	4
Circuitry and inner workings are concealed from the user	4
Pieces should be tightly connected to withstand blunt force	3
Wheels are controlled by speed control software that matches the user	2
Leash is made of rigid material, allowing the robot's movement to be communicated to the user	5
Leash is retractable/detachable	4

2.3 Technical Specifications

Table 2.3 - The technical specifications that correspond with each customer requirement

Customer Requirement	Technical Specification	Target Values
Can find a clear route between two locations	Pathfinding Software	90%-95% certainty of shortest path
Be aware of surroundings both moving and stationary	Identification	95%-99% certainty of correct identity
Will be able to walk for as long as a dog	Battery Life	1 hour
Can find alternative routes	Continuous Renavigation	< 25 ms to find the path
Can detect objects around it	Sensor Range	20 ft range
Product weight is kept to a minimum	Weight	<20 lbs
Designed with handles or other architecture to facilitate being picked up	Handles	2 handles
Locomotion that won't get stuck and work on a variety of surfaces	Wheels	4 omnidirectional wheels
Circuitry and inner workings are concealed from the user	Amount of Circuitry Enclosed	85%-95% of parts concealed
Pieces should be tightly connected to withstand blunt force	Force Withstandable	130 N
Wheels are controlled by speed control software that matches the user	Variance in User Speed	5%-10% average difference
Leash is made of rigid material, allowing the robot's movement to be communicated to the user	Modulus of Rigidity	28 GPa
Leash is retractable/detachable	Telescoping	3 sections

3 Assessment of Relevant Existing Technologies

Table 3.1 - Competitive Benchmarking

Competitive Product	Title / Description	Relation to this project
Guide Dog	A dog that has been specifically trained to guide the visually impaired through unpredictable environments.	This project aims to provide a less costly alternative to guide dogs, as they can be quite costly to take care of.
Glidance	An autonomous robot that guides its user from one location to another, similar to a guide dog. Its goals include affordability and ease of	The two projects have the same design goal of providing an affordable alternative to guide dogs. The main difference is the camera: the Glidance has its camera built into the handle whereas this project would use the user's smartphone.

Table 3.2 - Patent Research for Related Technologies

Patent Number	Title / Description	Relation to this project
CN103330636A	Electronic Guide Dog	This is a pending patent for the system of an electronic guide dog. It uses some similar parts to the project and provides a similar purpose.

4 Professional and Societal Considerations

Table 4.1 - Engineering Solutions Impact

Area of Impact	Impact	Description of Impact
Public Health and Safety	Y	Due to the smaller size and stature, the robot could be a tripping hazard in an emergency scenario. It is possible due to the locked angle of the camera, that the robot misses an obstacle and fails to warn the user of said obstacle
Global	Y	May struggle with navigating different parts of the world as it is designed for an American city.
Cultural	Y	The public may not be ready or willing to rely on semi-autonomous robots for help Some religions do not allow for the use of electronic or machines on days of rest
Societal	Y	Should greatly improve the quality of life for the visually impaired community. If designed solely around the user’s interest the robot could be disruptive to other people.
Environmental	Y	Relies on disposable batteries and electronics that are hard to dispose of. Modular electronics allow for more sustainable repairs: faulty parts may be swapped out, resulting in less waste overall.
Economic	Y	If a cheaper alternative is introduced, the seeing-eye dog market may be negatively impacted. By undercutting (reducing demand for)

		the seeing eye dog, reducing the access to higher quality sight assistance out there.

5 System Concept Development and Selection

Once the project was decided to focus on helping the visually impaired, one of the ideas that was initially considered was for a wearable glasses device with an embedded camera and incorporated headphones. The device would act as the eyes of the user, but translate that into audio messages that would guide them (see Sketch X).

Another idea was that of a handheld pen, allowing the user to point in the direction they wish to go, with the pen determining distance to the destination. Depending on the distance the pen would then vibrate or give audio cues to whether it was feasible to move that direction. (see Sketch X)

These two ideas were ultimately discarded as they were either not useful or too hard to implement (see Table 5.1). The next ideas were generated when the team refocused on what product was being replaced. This was when a robot guide was introduced, with two design paths to follow, wheeled or walking. Though walking would more accurately mimic the experience of using a guide dog, the cost and design effort was deemed too great, and a wheeled option was chosen (see Table 5.1)

Table 5.1

	Handheld Pen	Smart Glasses	Wheeled Robot	Walking Robot
Cost	+1	-1	0	-1
Ease of Implementation	-1	-1	+1	-1
Ease of Use	-1	+1	0	0
Reliability	0	0	+1	-1
Size	-1	-1	+1	+1
Total	-2	-2	+3	-2

Once it was decided to use wheels, the step was to determine the drive-train. A variety of different methods were considered (Table 5.2) but ultimately the decision was made to go forward with Omni wheels as the group felt that they provided the most utility while having the fewest drawbacks, as mobility and the ability to work on a variety of surfaces were highly valued.

Table 5.2

Option	Description	Pros	Cons
Tank Treads	Continuous tracks for movement across various surfaces	Works on various surfaces	Slow, Mechanically complex, Slips on smooth surfaces
Forklift Drive	Wheels with motors that control angle and direction	Potential for high maneuverability	Very Complex, requires 8 motors
Standard Car Design	Traditional 4-wheel car configuration	Tried, tested, and reliable	Wide turning radius, unsuitable for tight spaces like an office or classroom
2 wheels with Central roller	Two powered wheels and a central roller	Easy to execute, tight turning radius	Unstable, prone to getting stuck, only works on smooth surfaces
Omni Wheels	Wheels that allow for movement in any direction without rotation	Translation in any direction, very high maneuverability	Half power (4 motors, can only use 2 at a time) carries lower weight

To bring the subsystems together, a material for the chassis was needed. While metal and PLA have their pros and cons, wood is the best compromise between all the considered factors.

Table 5.3

	Metal	PLA (3D Print)	Wood
Assembly	-1	0	+1
Strength	+1	-1	+1
Versatility	-1	+1	0
Total	-1	0	+2

The decisions for the components of the Leash can be decided between two categories: Utilities, what functionality does it add for the user, and Feasibility, is it within the team's ability to implement. As seen in Table 5.4, a Wii nunchuck was not only the most feasible, but also the most useful, since its both comfortable for the hand and allows the user to command the robot. Other features such as a 360 degree joint and telescoping shaft were chosen as a balance between feasibility and utility, especially in the face of other options.

Table 5.4

	Static Joint	Rotating Joint	360 Joint	Rod Handle	Custom Handle	Nunchuck Handle	Folding Shaft	Telescoping Shaft
Feasibility	+1	-1	+1	+1	-1	+1	+1	0
Utility	-1	+1	0	0	+1	+1	-1	+1
Total	0	0	+1	+1	0	+2	0	+1

Table 5.5

	Dijkstra's Algorithm	Bellman Ford	A* Algorithm
Implementation	0	0	0
Speed	0	0	+1
Robustness	-1	-1	+1
Total	-1	-1	+2

This selection matrix was used to determine the best algorithm for the project. All algorithms had similar implementations, but where they varied was their speed and robustness. A* algorithm was the best for both these categories, which led it to be used in the project.

6 Subsystem Analysis and Design

The design is composed of six subsystems, each with equal weight in determining the success of the project and in fulfilling the users requirements. The subsystems (Figure 6.1) include: Object Detection, Path Finding, Drivetrain control, Drivetrain electronics, Chassis, and Leash. The overall design of each subsystem was developed by the team, but each team member was given their own subsystem to fabricate and work out specific design features relating to implementation and integration. Figure 6.2 shows the hierarchy of the subsystems and the features within.

Figure 6.1 - Subsystem Diagram

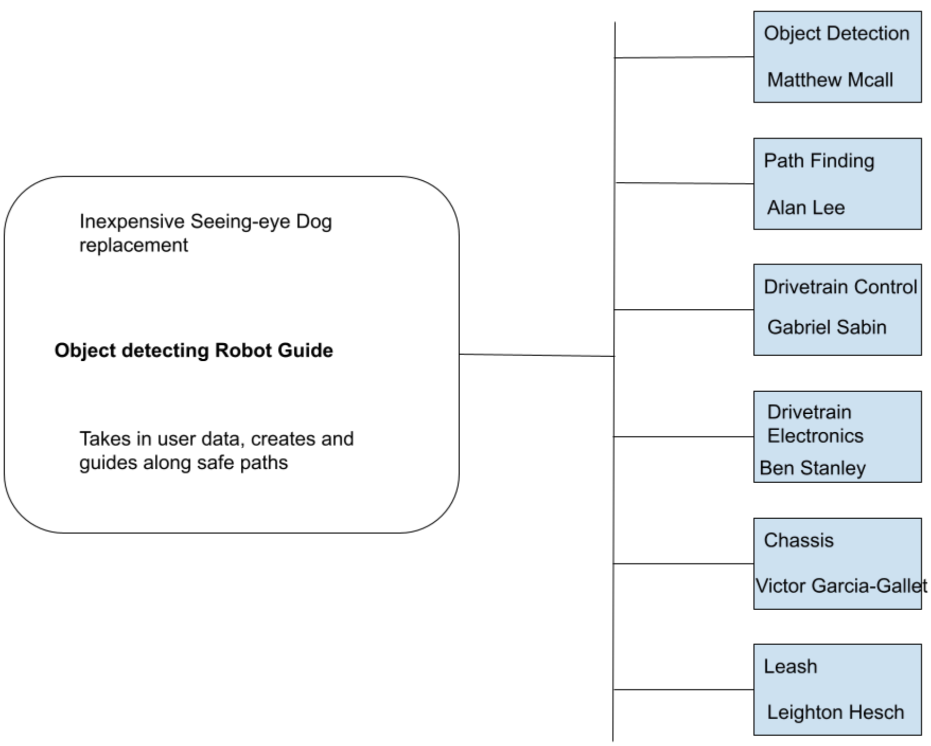
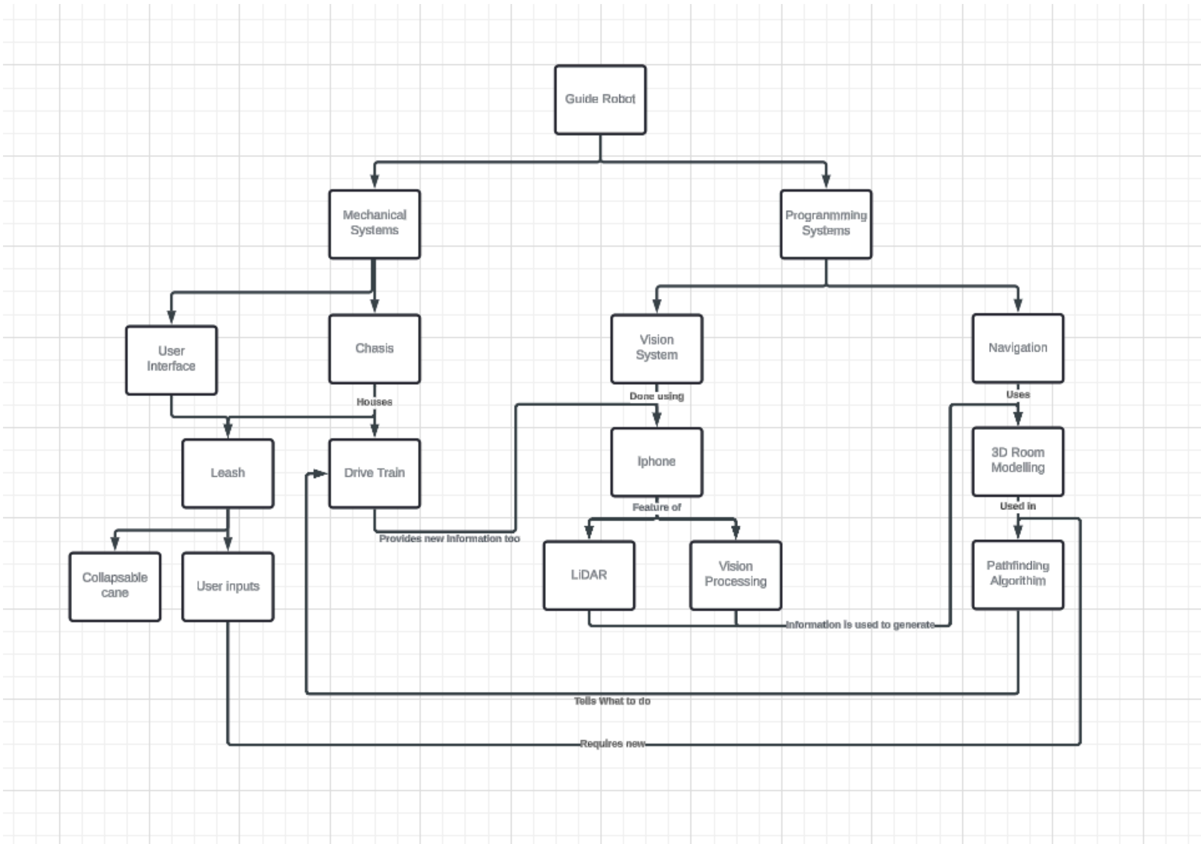


Figure 6.2 - Hierarchy of Subsystems



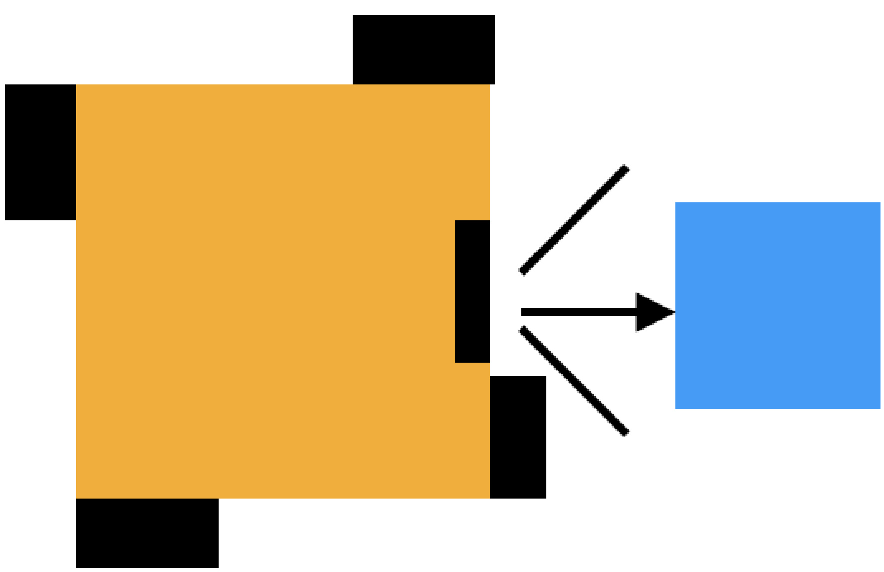
6.1 Subsystem 1: Object Detection

The first subsystem of the project is object detection. This subsystem is concerned with mapping the room, identifying walkable areas, and identifying walkable routes. The original plan called for a combination of computer vision and “time-of-flight” based sensors. Essentially, onboard sensors, such as LiDAR would emit electromagnetic waves to probe the surroundings. Specifically, LiDAR fires a laser at a known angle and measures the time it takes for the laser to bounce back. Given the speed of light, the sensor can estimate how far the laser traveled before it bounced back based on the time it took to do so. The robot would then be able to reconstruct a three-dimensional representation of the environment. This three-dimensional representation is called a “mesh” and consists of a list of vertices in three-dimensional space. The vertices are ordered in such a way that they form an array of triangles. Multiple triangles share adjacent edges to form surfaces. These surfaces would represent real world objects. This process is known as “mesh reconstruction”. On the other hand, the computer vision approach involves the use of digital cameras. The camera would capture an image and then divide the image into multiple subsections. The robot would then compare the subsections against an extensive database of known objects. Unfortunately, due to time constraints, the computer vision approach was dropped.

The initial attempt aimed for complete inclusion of the necessary sensors and computing resources onboard the robot itself. However, this would increase the cost of the project, and required additional time to design and test. As a result, other existing solutions were explored that were on the market and were commonplace, such as the iPhone. Modern iPhones, specifically the “Pro” series iPhones, include a LiDAR sensor. Furthermore, Apple provides a suite of tools and software called “ARKit” that accelerates the development of technologies that use LiDAR on the iPhone. As such, our second attempt was to build an iOS application using SwiftUI. While mesh reconstruction was achievable, it was difficult to integrate with the pathfinding subsystem and to facilitate wireless communication with the drivetrain electronics. To alleviate these issues it was decided that there would be a switch to the Unity Engine. Unity still leverages Apple’s ARKit to perform mesh reconstruction, but also includes tools to perform navigation and wireless communication.

For the final product, revisiting the initial concept of completely integrated onboard computing and sensors would be helpful. This would provide consistency and would not require the user to have a specific model of iPhone or have to set up their personal device for use with the robot. The robot can also be tailored to customer needs by positioning sensors to better detect low overhangs and cover areas normally covered by the peripheral vision of a sighted person.

Figure 6.3 - LIDAR Object Detection



6.2 Subsystem 2: Pathfinding Programming

The second subsystem of the project is the pathfinding around obstacles within the robot's path. There were many different pathfinding algorithms to choose from, each with their positives and negatives. However, from the research done about pathfinding, a few algorithms stood out, which were: Dijkstra's Algorithm, Bellman Ford, and A* algorithm. These were some of the more commonly used pathfinding algorithms, so finding different implementations for them was quite simple.

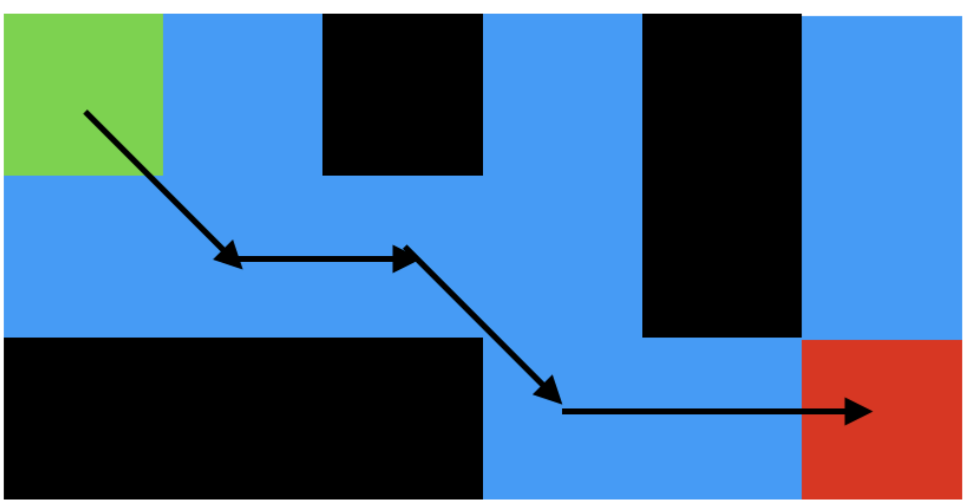
Initially, the pathfinding algorithm was designed in SwiftUI with the idea of integrating the sensing subsystem's information with the pathfinding's system. However, this wasn't the right approach as it allowed communication between the sensing and pathfinding, but not with any other subsystem. This led to a switch of software, from SwiftUI to Unity, to allow for better compatibility between subsystems.

From the three pathfinding algorithms, it was eventually narrowed down to one algorithm, the A* algorithm, due to its numerous benefits over the others. Another reason this algorithm was chosen was due to Unity having a built in A* pathfinding algorithm that could be used. The code was initially implemented with two dimensions in mind as the mesh of a room was intended to be a top down view, but this had to be changed due to the sensing subsystem creating the mesh within three dimensions. All the information generated from the pathfinding algorithm is sent to the motor control to spin the wheels the direction needed to avoid the obstacle.

Looking at Figure 6.4, the path generated uses diagonals, as diagonals reduce the amount of time to reach the end destination. Instead of going, horizontally then vertically, going diagonal saves time in creating the path. It was implemented this way due to the user having to follow the robot, and if the robot takes too long to determine a path or update, the user could accidentally go into a wall.

For future implementations of this subsystem, the main component that can be altered is the speed of the pathfinding algorithm. The path has to be constantly updated in order for the user to avoid obstacles and adjust to environmental factors. This ensures the safety of the user and allows for multiple paths to be created before the robot moves, allowing for error checking.

Figure 6.4 - Pathfinding around obstacles



6.3 Subsystem 3: Drivetrain Control

The third subsystem was the drivetrain programming, which took the information given by the pathfinding subsystem. In the early stages of the design process for this subsystem, the main concern was what would be needed to allow it to integrate with the others and work effectively. After analyzing the pros and cons of different types of microcontrollers and ultimately decided that a wireless board would make communication with pathfinding much easier.

The original wireless microcontroller that was planned to be used was the Raspberry Pi Zero 2W due to its flexibility and ease of use. However, when testing it out,it was discovered that it had trouble connecting to the RPI WiFI due to the security measures. The solution to this was to use another board called the ESP32 (Figure 6.5). A big advantage the ESP32 has over the Raspberry Pi is that it creates its own wireless network, allowing seamless integration. Another advantage of the ESP32 is its flexibility due to being compatible with Arduino IDE. Ths specification allows customers to reprogram whatever they want since Arduino is so common and very simple to pick up.

Programming the motors ended up going very smoothly and was straightforward. Three pins were required to control one pair of motors, making for a total of six pins initialized. Two pins control the direction of the wheels by switching between high or low and the remaining pin controls the pulse width modulation output, allowing for change in speed. This specification was put in place in order to address customer's needs for changing the walking pace. Movement commands in the program are as follows: forward, backward, right, diagonal right, and diagonal left. The pathfinding subsystem communicates with this subsystem in order to determine which command to use at what time, leading the user around obstacles and to their destination.

Figure 6.5 - ESP32-C6 Microcontroller

6.4 Subsystem 4: Drivetrain Electronics

The fourth subsystem, the physical component of the drivetrain, is responsible for translating the code of substem 3 into the physical movement of the robot. The first thing that was needed was to determine what wheels and motors were suitable. The next step was to determine the average walking speed of someone who is visually impaired, which was determined to be approximately 0.7 meters per second, and an estimated weight of 7 pounds. By using the formulas:

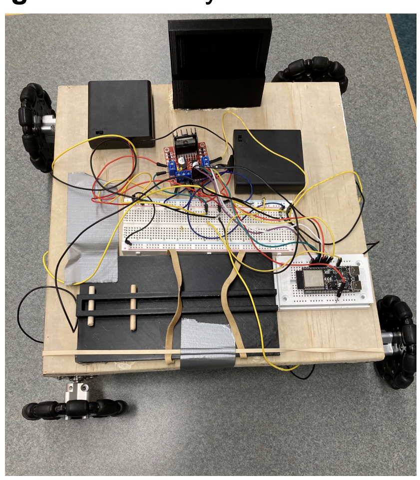
$$Vehicle\ speed\ = [2*\pi*(D/2)]*[RPM/60]$$

$$Torque\ *\ RPM\ =\ Mass\ *\ acceleration\ *\ Vehicle\ speed\ /\ (2\ *\ \pi)$$

Where D is the diameter of the wheel, and RPM is the speed of the motor

it was possible to find wheels and corresponding motors that would not only meet the standards set, but also ensure that the wheel had the right bore size for the size of the axel so that they could be attached by using a mounting hub, a machined piece of metal which can screw onto the wheel and fit onto the D-shaped axel, transferring the force of the motor into the wheel. A motor mount was also needed, essentially a machined piece of metal that could screw onto both the chassis and motors securing them. Once the parts had arrived, and assembled, the next step was to wire the motors to the ESP32, however, it could only output 5V while the motors required 12V to run properly. To accommodate for this, it was decided to use a 12V external power supply with two dual h-bridges, which can not only provide the necessary voltage to the motors, but also allow us to power the ESP32 without connecting it to a laptop. Unfortunately one of the H-bridges had burned out when too much current ran through it. A Replacement would not arrive in time so the decision was made to run two motors parallel off one H-bridge. While this allowed the robot to move, it reduced the total voltage per motor, weakening them. This also tethered the motors in each principal direction (forwards/backwards and right/left) to each other, eliminating the robot’s ability to rotate due to lack of single wheel control.. In spite of this setback, the drive train was able to achieve its goal of translating the directions from the drive-train control to the real-world movement of the robot The final iteration of the drive train can be seen in figure 6.6.

Figure 6.6 - Fully Wired Drivetrain



6.5 Subsystem 5: Leash

The fifth subsystem, called the Leash, is practically the user interface of the product. Thus it was imperative that the human was taken into consideration in the entire process. The team devised scenarios in which a certain functionality for the Leash was needed.

The first thought was how the Leash would communicate the starting and stopping of the robot. A traditional dog leash will fall slack, and the user will walk into the robot. As such, the overarching design constraint in the Leash was that it had to be rigid, causing any movement to be communicated instantly to the user.

However, the user should be able to stop the robot whenever they please, which requires the user to communicate with the robot. This would have to be done in the handle, as it was where the user's hand would rest, allowing for quick access. Initially a four button control scheme was devised; one button for each direction. However, since the requirements called for more control to the user, a joystick with extra buttons was deemed more appropriate. The joystick would allow the user to determine direction with a greater degree of freedom, while the buttons could give commands such as stop. These commands would be communicated through a wire that runs through the Leash, leading to the next design constraint that it had to be hollow.

The next scenario was when either the product was not in use, or would be required to be picked up. In this scenario the leash would have to be collapsible, to not be burdensome upon the user. Since the leash also had to be hollow, this led to telescoping as the method to accomplish this, which had the added effect of being adjustable in length, accommodating users of different heights.

Also, considering the robot would need to turn, the leash would need to be attached in a way that communicated the change in direction, without detaching from the user. The only real choice was a 360 degree joint, as the angle of the leash would communicate direction, but the angle would only go so far, eventually pulling the user along. The extra warning would give the user time to react before this force, allowing guidance without unnecessary jerking.

Most importantly, the user may end up using this for a long period of time, so it must be comfortable to hold. This would require shaping the handle to a user's hand. There are already a number of controllers on the market that contain the previously mentioned joystick and button scheme that shape to a human hand, so the design was modeled off those.

From top to bottom the final design of the Leash was a 360 degree joint connected to the chassis at one end and a telescoping rod at the other. This rod has a wire running through it that connects to a handle designed for a hand to hold, containing a joystick and buttons that allow the user to communicate their desires with the product.

A final sketch of the leash, with all the features decided upon, is seen in Sketch 6.7.

Figure 6.7 - Leash Subsystem Sketch

6.6 Subsystem 6: Chassis

The final subsystem is the chassis, which is the robot's body. The previous subsystems work very well in conjunction, and there must be something to unify them. This provides the product an important sense of wholeness in the eyes of the user. It is also responsible for holding the other subsystems together during movement, allowing the robot to be fully autonomous.

The main design goals of the chassis included portability, strength, and protection. Portability would be achieved with a small size and a light weight. This is especially important in cases where the user would need to carry the robot, such as up stairs or over terrain the robot cannot traverse. Strength would be achieved by selecting an appropriate material and assembling the chassis in a way that does not compromise the robot's stability. Protection is in regards to the electronics: the chassis must be able to protect the electronics from any dangerous external influence. This is especially important for the user, as not only could they be stranded if the robot fails, but they would need to spend money on repairing it.

The most important decision for meeting these goals was which material to use. Metal and plastic were both considered, but both had drawbacks that were too significant for them to be used in the prototype. When considering their advantages and disadvantages, they are essentially opposites of each other. Metal is very strong and durable, but it is very difficult to work with. Not only did the chassis need a specific shape and size, but it needed holes for the motors and leash. This would have been too difficult with the tools at the team's disposal. On the other hand, plastic offers much flexibility with assembly but is not very strong. 3D printing the chassis with PLA was an option, and was attempted by the team. However, the plastic was not only susceptible to bending and breakage, but turned out too small because of the 3D printer's constraints.

For the prototype, the chassis was made of wood, as shown in Figure 6.7. It offered the best compromise for strength and versatility: plywood is very difficult to break but easy to cut and drill holes into. The piece of wood used for the chassis was about a square foot in area and half an inch in size. This offered a desirable light weight and substantial size for the electronics to fit. Unfortunately, protection for the electronics was not made, so the robot is vulnerable to harmful external influence. In any case, the chassis worked well in the context of the prototype.

Figure 6.8 - The wooden slab used for the chassis

7 Results and Discussion

7.1 Results

The tests conducted on the prototype were largely of a binary nature in regards to functionality. All the features were tested, and most were able to complete the desired function. Though largely, the subsystems were designed with completing the task they were given in mind. The only exception was the Drivetrain Control, which had a desired speed of 1.5-2 ft/sec, or about the walking speed of the average person. This was tested by measuring the length of a room and timing how long it took to traverse the room. Since a 12 ft room was crossed in 8 seconds, the minimum desired speed was achieved..

All other subsystems also passed their tests. When an on signal was sent, the wheels moved in the correct directions, thus the Drivetrain electronics were working. The Object Detection software was able to identify objects. The Pathfinding software found its way around objects in a simulated environment. The Leash communicated movement to the user, and user inputs to the robot. And finally, the chassis was dropped from 3 ft, and held up.

Overall the product did what the team designed it to do, and the team tested for what was expected of the product. Perhaps there were unimagined tests that could have provided insight into improvements for the product.

7.2 Significant Technical Accomplishments

Since each product was developed in parallel, the greatest accomplishment was combining all the technology. The Drivetrain control had software to determine movement, but it was the Pathfinding software that needed to determine those commands. However, at that time, the Pathfinding software only found a path, and still needed to extract direction. At the same time, the Pathfinding software needed input from the handle to determine direction. These three separate softwares, with just enough code to show they worked independently, needed to be combined into a single process, yet the team overcame this hurdle.

The other technical accomplishment was working with wireless communications. Since the Pathfinding and Object Detection software were on one machine, while the Leash and Drivetrain control were on another, and these machines ran on separate languages, networking was necessary. While combining the code was difficult, this was impressive, as the most exciting moment in production was when the team could move the robot by entering commands on a separate device.

8 Conclusions

The purpose of the product is to replace the core functionality of a seeing-eye dog. It does this by scanning its surroundings to create a mesh of the environment. Then it takes user inputs to determine its destination, and, in combination with the mesh, finds the optimal path between it and the destination. Then it breaks this path down into a series of directional commands, and, applying these commands to the wheels, guides the user to their desired location.

These design features were made with the end user experience in mind. The path found is meant to keep the user safe. The drivetrain is meant to regulate speed in order to keep pace with the user, neither too fast nor too slow. The chassis is meant to be carried by the user should the product need to be moved. Finally, the leash has user control and feedback in its physical properties and digital abilities.

Moving forward, the team should have more time to refine pathfinding, object detection, and leash controls, so the robot may better yet work for the user. Drivetrain control could also be improved, reducing jerk and smoother directional changes. It should also be investigated how to deal with stairs, or other terrain that was considered outside the scope of this project.

9 Appendix A: Selection of Team Project

Ideas were generated through the team conversing in how to overcome visual impairment. They fell into two groups: assistive, where the device would act as one's eyes and merely advise the user, and proactive, where the device would navigate autonomously and the user follows it.

The only other product on the market that solves a similar problem is a seeing-eye dog. Therefore, the baseline for the generated designs is the seeing-eye dog.

The first assistive idea was the handheld pen, which would be held in front of the customer and direct them where to go based on the obstacles identified. Although inexpensive, a visually impaired person is unable to accurately aim the pen, due to their diminished sight.

The next assistive idea was the smart glasses. They are fairly easy to operate because the user doesn't need to do anything but wear them and listen to directions. The downsides come with the implementation, as it would be easy for them to get knocked off and the visually impaired customer will have trouble locating them, assuming they didn't break from the fall. It might also be difficult to make a sleek and convenient design due to it requiring obstructive parts such as sensors or a camera.

The first proactive design was the wheeled robot. This design ended up having the least downsides, becoming the chosen design. It operates in the role of a guide dog, finding a path for the user, and guiding them along. Unlike the assistive designs, the baseline of a seeing-eye dog gives the team a target, both to emulate and strive for. The wheels limit the movement capabilities of the robot, but provide a simpler movement system.

An alternative design considered was the walking robot. This design would use a leg mechanism such as Jansen's Linkage ("Jansen's Linkage") or the Klann Linkage ("Klann Linkage") which simulate a walking motion. Filling a similar role as the wheel robot, the walking robot would guide the user much like a seeing-eye dog. However, the implementation of legs was decided to be too difficult. Although the design would be more versatile and provide the most assistance to the user, the downsides led the wheeled variant to be chosen

	Porch Defender	WallE	Trash Grabber	Heavy Item Transport Platform	Seeing Eye Dog	Food delivery Improvement	Waste Transport	Mice Finder	Text to speech	Massage	Medicine Admin (needles)	Sustainability (hand crank or bike generator)
Cost	-1	-1	0	+1	-1	0	-1	-1	+1	-1	-1	-1
Feasibility	-1	0	+1	+1	+1	-1	-1	+1	+1	-1	0	+1
Complexity	-1	-1	-1	0	+1	+1	0	-1	+1	-1	0	+1
Weight	0	-1	+1	-1	-1	-1	-1	+1	+1	-1	0	
Size	+1	0	+1	-1	0	0	0	+1	0	-1	+1	
Number of Motors	0	-1	-1	0	0	0	0	+1	+1	-1	-1	
Other Parts needed	-1	+1	+1	+1	0	0	0	0	+1	0	-1	
Approvability	-1	+1	+1	0	+1	0	+1	0	+1	+1	-1	
Sustainability	0	+1	0	0	0	+1	+1	0	0	0	0	
Universal Design	0	-1	-1	+1	+1	0	+1	-1	+1	-1	0	
Design Innovation	+1	+1	0	0	0	0	0	+1	-1	-1	0	
Public Service	0	+1	+1	0	+1	0	+1	+1	+1	0	+1	
Sum	-3	+1	+4	+2	+3	0	+1	+3	+7	-7	-1	

10 Appendix B: Customer Requirements and Technical Specifications

Customer Needs	Customer Requirements
Provide similar functions to a guide dog	Can find a clear route between two locations Be aware of surroundings, both moving and stationary Will be able to walk for as long as a dog
Should be able to navigate around obstacles/won't lead user into things	Can find alternative routes, Is able to detect objects around it
Leash is comfortable to hold and at the right height	Leash is made of rigid material, allowing the robot's movement to be communicated to the user
Walks with the users, as a dog might	Wheels are controlled by speed control software that matches the user
Light enough to carry up and down stairs/If it runs out of battery or breaks, I want to be able to carry it	Product weight is kept to a minimum Designed with handles or other architecture to facilitate being picked up
Usable on different terrains, or wherever the user is likely to travel	Locomotion that won't get stuck and work on a variety of surfaces
Simple to interface with	Circuitry and inner workings are concealed
Leash isn't in the way	Leash is adjustable/detachable
Won't break the first time it's dropped	Pieces should be tightly connected to withstand blunt force

Customers	Visually impaired people with trouble navigating and reading unfamiliar environments to be navigated
Stakeholders	For-profit and nonprofit organizations that work with the visually impaired. The electronics manufacturers.

11 Appendix C: Gantt Chart

Team Undecided

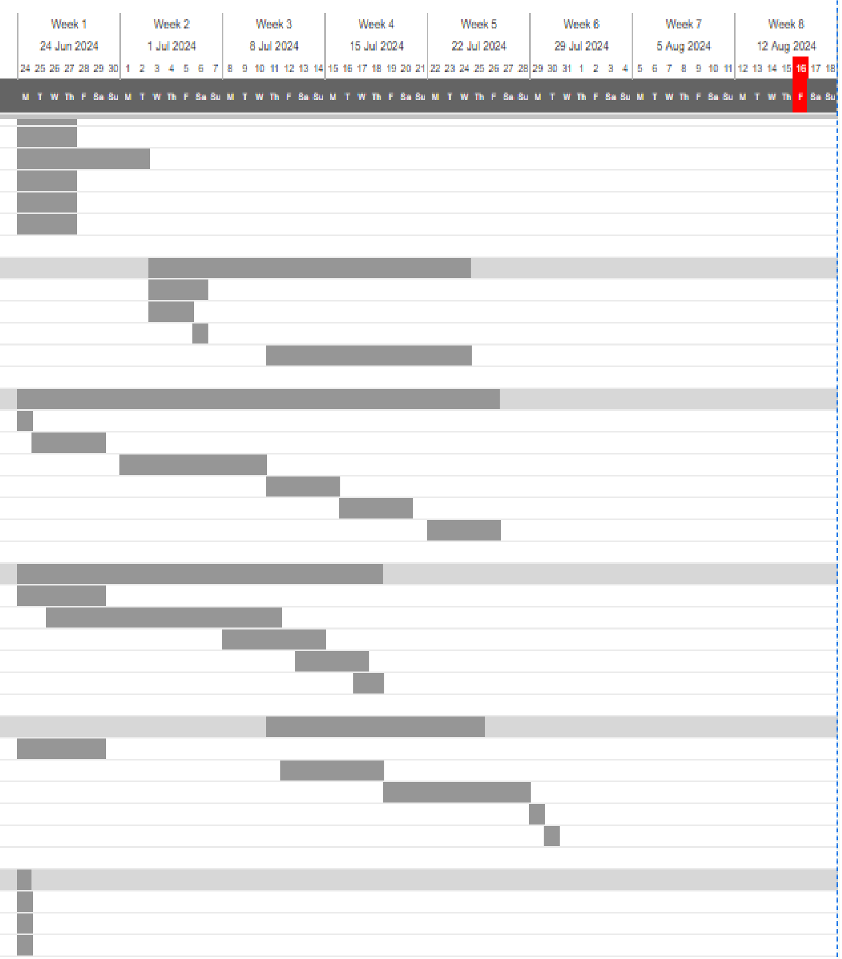
Go Pro

Gantt Chart Template © 2012-2020 by Vertex42.com. Licensed for private use only. Do not publish on the Internet.

Project Start Date: 6/24/2024 (Monday) Display Week: 1

Project Manager: Everyone

WBS	Task	Lead	Start	End	Days	% Done	Work Days
1.2	Drivetrain	Ben	Mon 6/24/24	Thu 6/27/24	4	100%	4
1.3	Leash	Leighton	Mon 6/24/24	Tue 7/02/24	8	100%	7
1.4	Sensing	Matthew	Mon 6/24/24	Thu 6/27/24	4	100%	4
1.5	Pathfinding	Alan	Mon 6/24/24	Thu 6/27/24	4	100%	4
1.6	Drivetrain Control	Gabe	Mon 6/24/24	Thu 6/27/24	4	100%	4
1.7	[Insert new rows above this one, then hide or delete this row]						
2	**Implement Leash**	Leighton	Wed 7/03/24	Wed 7/24/24	22	100%	16
2.1	Rod Aquisition		Wed 7/03/24	Sat 7/06/24	3	100%	3
2.2	Electronics Aquisition		Wed 7/03/24	Fri 7/05/24	2	100%	3
2.3	Combine Rod and Electronics		Sat 7/06/24	Sat 7/06/24	1	100%	0
2.4	Create and apply handle to Rod connection		Thu 7/11/24	Wed 7/24/24	13	100%	
2.5	[Insert new rows above this one, then hide or delete this row]						
3	**Implement Sensing**	Matthew McCall	Mon 6/24/24	Fri 7/26/24	33	100%	25
3.1	Create iOS Application		Mon 6/24/24	Mon 6/24/24	1	100%	1
3.2	Read Data from LiDAR		Tue 6/25/24	Sat 6/29/24	5	100%	4
3.3	Perform Mesh Detection		Mon 7/01/24	Wed 7/10/24	10	100%	8
3.4	Export Reconstructed Mesh to Pathfinding		Thu 7/11/24	Mon 7/15/24	5	100%	3
3.5	Perform Object Detection with Computer Vision		Tue 7/16/24	Sat 7/20/24	5	100%	4
3.6	Export Identified Objects to Pathfinding		Mon 7/22/24	Fri 7/26/24	5	100%	5
3.7	[Insert new rows above this one, then hide or delete this row]						
4	**Implement Pathfinding**	Alan	Mon 6/24/24	Thu 7/18/24	25	100%	19
4.1	Research Pathfining Algorithms		Mon 6/24/24	Sat 6/29/24	6	100%	5
4.2	Implement Pathfinding Algorithm		Wed 6/26/24	Thu 7/11/24	1	100%	12
4.3	Test Pathfinding Algorithm		Mon 7/08/24	Sun 7/14/24	1	100%	5
4.4	Improve Pathfinding Algorithm		Sat 7/13/24	Wed 7/17/24	1	100%	3
4.5	Combine with other subsystems		Wed 7/17/24	Thu 7/18/24	1	100%	2
4.6	[Insert new rows above this one, then hide or delete this row]						
5	**Implement Drivetrain Control**	Gabe	Thu 7/11/24	Thu 7/25/24	6	100%	11
5.1	Obtain Raspberry Pi		Mon 6/24/24	Sat 6/29/24	6	100%	5
5.2	Implement speed control		Fri 7/12/24	Thu 7/18/24	7	100%	5
5.3	Take pathfinding and make it drive		Fri 7/19/24	Sun 7/28/24	10	100%	6
5.4	[Task]		Mon 7/29/24	Mon 7/29/24	1	100%	1
5.5	[Task]		Tue 7/30/24	Tue 7/30/24	1	100%	1
5.6	[Insert new rows above this one, then hide or delete this row]						
6	**Implement Chasis**	Ben	Mon 6/24/24	Mon 6/24/24	1	100%	1
6.1	Make CAD		Mon 6/24/24	Mon 6/24/24	1	100%	1
6.2	3D Print		Mon 6/24/24	Mon 6/24/24	1	100%	1
6.3	Assemble		Mon 6/24/24	Mon 6/24/24	1	100%	1



12 Appendix D: Expense Report

The expenses listed in the table below all ended up being directly related to the robot in some physical manner. Even though this may be the case, in the final product, there would be a price to deliver the robot to costumer's. If additional funds were spent, the product could work more consistently and require less repairs. More money should have been allocated to the leash controller in order to provide more buttons and functions for the user. Another h-bridge could have also helped get the motors more power and allow for higher speeds.

Table 5 - Project Expenses

Item	Quantity	Unit Price	Subtotal
3604 Series Omni Wheel	4	$9.99	$39.96
Greartisan DC 12V 30RPM Gear Motor	4	$14.99	$59.96
Black 37mm DC Gear Motors Mounting Bracket	4	$8.99	$35.96
1310 Series Hyper Hub (6mm D-Bore)	4	$7.99	$31.96
WIi nunchuck & wires	1	$18	$18
Bates- Extension Pole, 1.4 to 3 Ft Pole, Telescoping Pole	1	$14.82	$14.82
WORKPRO Impact Universal Joint Set	3	$6.05	$18.05
Total	$255.72		

13 Appendix E: Statement of Work

Our goal is to be able to make an impact on the lives of the visually impaired with a cost effective alternative to seeing-eye dogs. We plan on achieving this by creating a robot that uses affordable, yet effective parts. This robot will be able to act autonomously and navigate around obstacles using a camera that scans its surroundings. It will also be able to cross the street, go across multiple terrain, and send feedback to the user. In order to send feedback to the user we will attach a leash that the customer can hold. This leash will also act as a controller so that the user can give commands to the robot as well.

14 Appendix F: Professional Development - Lessons Learned

Multiple lessons were learned throughout the engineering design process. One of which is defining the roles of group mates and setting expectations that should be met by all. This general consensus ensured accountability amongst group mates and that there wouldn't be any conflict amongst the group. If an assigned task wasn't completed, it was the groupmate's responsibility to complete it, while also finishing other assigned tasks. This strategy was quite helpful and would definitely be used in the future. A problem that occurred within the group setting was the idea to use democracy to determine votes. Due to having an even group, there were times where there were draws in the voting, leading to a stall in the engineering design process. Additionally, there were times where the group didn't respond in a quick fashion due to having outside matters to attend to. Something worth attempting in the future is communication before performing a task. At times, issues arise where something was unneeded due to miscommunication, so communicating a task before it is completed would save time and provide a greater experience for the group.

15 Appendix G Software / Technology Used

16.1 Collaboration Among Team Members

- Discord
- Text Messaging/Phone Calls
- Google Drive

16.2 Subsystem Design

- L298N Motor Driver
- ESP32-C6
- IPhone 14 Pro Max

16.3 Programming

- Unity ARFoundation
- Arduino IDE
- Swift
- Apple ARKit

16.4 Subsystem Testing/Simulation/Emulation

- Unity ARFoundation
- Arduino IDE in conjunction with the ESP32

17 Appendix I: User Manual

Setup: The user should first insert their phone into the holder, located at the front right of the robot, and start the app. The user will then take the leash, located at the back middle of the robot, and adjust the height to a comfortable level. This is done by unlocking one of the two latches, pulling the rod, then locking the latch, so the leash holds firm.

Traveling: Once the robot is set up, it may begin its guidance protocol. Take hold of the handle at the end of the leash farthest from the chassis. Hold onto the handle comfortably. The joystick is thumb operated, and should be moved in the desired direction. The robot will then move to the set destination, avoiding obstacles as it goes. Should there be a need to stop, simply press one of the buttons. To resume traveling, engage the joystick.

Reset: After completing the route, to return the robot to an idle state, reset the length of the leash by unlocking the latches and push down. Once the leash reaches its minimum length or when it stops retracting, lock the latches again, and the robot is reset.