March 13th, 2022

Dr. Mike Hegedus
Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, BC, V5A 1S6

RE: ENSC 405W Design Specification

Dear Dr. Hegedus and Dr. Rawicz,

The document accompanying this letter outlines the design specifications for NaviBot, an autonomous robot to perform in-house deliveries. We hope to create an intuitive design that will allow workers to effectively perform their basic delivery duties with ease and with little to no need for training.

The purpose of this document is to outline the design specifications in relation to the requirements of the NaviBot, along with alternative designs, test plans, and user interface design. The design specifications will be listed for the proof-of-concept, engineering prototype, and production stages. Regarding Section 5, "Drive Mechanism", it was realized that design requirements for this system were overlooked in the previous document. As a result, the listed design specifications will not reference any previous requirements specifications.

NaviBot is DAM^3 Technologies' combined efforts towards developing a solution to the increased logistical workload faced by office and warehouse workers. Our project will contain applicable engineering principles our team has obtained during our time at SFU. DAM^3 Technologies aims to deliver a proof-of-concept prototype during the ENSC 405W time frame, with development continuing in ENSC 440.

If you require any further clarification or have any questions pertaining to this document, please contact DAM^3 Technologies' CCO Martin Yang at yangtaey@sfu.ca.

Sincerely,

Davis Hogg
Chief Executive Officer
DAM^3 Technologies

DAM^3 Technologies

Design Specification

Company 1

Davis Hogg, CEO: 301335409
Alek Srdic, CHO: 301342141
Mark Lavin, CIO: 301381318
Michael Kim, COO: 301340718
Martin Yang, CCO/CFO: 301347120

March 13, 2022

Abstract

NaviBot is an autonomous robot that will allow workers to perform in-house deliveries without the need for manual transportation. Using sensors and simultaneous localization and mapping algorithms, NaviBot will be able to navigate between locations whilst avoiding obstacles. This document outlines the design specifications and their related requirement specifications to be met at all stages of development: proof-of-concept, engineering prototype, and production. While still listing production requirements, this document will mainly focus on requirements for the proof-of-concept (POC) and engineering prototype (EP).

Contents

List of Figures

List of Tables

Glossary

Term	Description
Absolute Positioning	Positioning system used in relation to a fixed point, i.e. a specific point in calibrated SLAM data
DC	Direct Current
EP	Engineering Prototype
GPIO	General Purpose Input/Output
LiDAR	Light Detection and Ranging
MDDS30	Cytron Technologies SmartDriveDuo-30 MDDS30 motor controller
MDF	Medium Density Fiberboard
NaviBot	Product name
OS	Operating System
PCB	Printed Circuit Board
POC	Proof Of Concept
ROS	Robotic Operating System
RPi4B	Raspberry Pi 4 Model B microcontroller
SLAM	Simultaneous Localization and Mapping
UI	User Interface

1 Introduction

NaviBot, the smart delivery robot, aims to improve efficiency with respect to package delivery from the receiving bay to other departments in office and engineering spaces. To achieve this, a shipping and receiving handler will load small boxes and packages onto NaviBot and send them to different rooms, offices, or cubicles with the use of NaviBot's touch screen interface.

1.1 Background

As companies grow, so do their logistics needs, and with those needs come a growth in package intake. The overhead needed to expand a company's logistical capabilities can require an extensive amount of reworking or additions to existing infrastructure. NaviBot allows companies to address their increased logistical load with minimal modification to their logistics infrastructure.

1.2 Scope

The purpose of this document is to outline the design requirements of NaviBot in relation to the requirement specifications. These requirements will describe the goals for the proof-of-concept, prototype, and production models and are split into the multiple systems of the device- Microcontroller, Chassis Design, Drive Mechanism, Navigation, User Interface, and Power System.

1.3 Design Classification

Classification	Description
A	Proof of Concept
B	Engineering Prototype
C	Production Version

Table 1: Stage Classification

The Proof of Concept requirements specify the requirements that must be met by the end of ENSC 405W. The Engineering Prototype requirements specify the requirements that must be met by the prototype by the end of ENSC 440. The Production Version requirements specify the requirements that must be met by the device once it is in production.

Design codes will be generated as follows:

Des (Section).(Subsection).(Design Number) (Development Stage)

2 System Overview

NaviBot is an autonomous robot intended to aid in package transfers in offices, warehouses, and other locations requiring internal shipping. Being controlled by a shipping and receiving handler, it will drive itself to a destination, wait for the receiving party to acquire their package, then return to the home location.

NaviBot's body and frame consist of a square cargo hauling platform with its touch screen interface and LiDAR sensor mounted on an arm (herein referred to as "the control arm") spanning up and over the platform (Fig. 1). The LiDAR is located directly above the centre shipping platform, with its location chosen as it is unobstructed by itself and to act as a centralized reference point for the SLAM algorithms covered in Section 6.2 - SLAM.

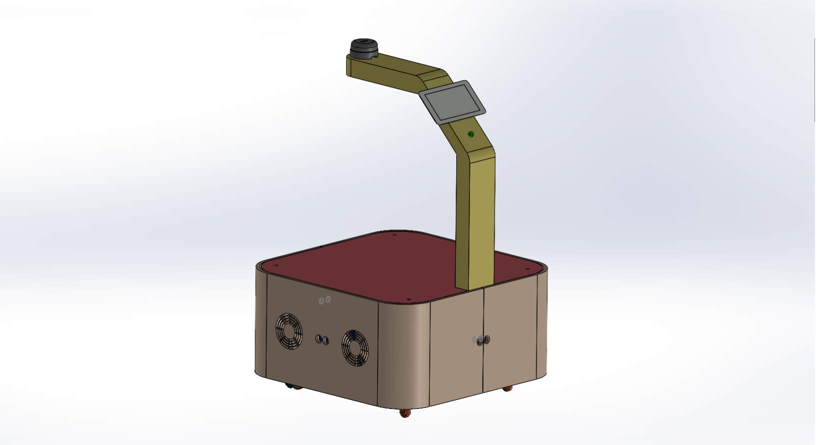


Figure 1: NaviBot Complete CAD Model

Considering NaviBot is an autonomous robot, it has several system components that must work in unison to achieve a safe and effective package delivery. This is managed through the use of a Raspberry Pi 4B (RPi4B) with a complete system diagram per Fig. 2 and 3. NaviBot makes use of two sensors to navigate its surroundings: LiDAR and ultrasonic. Both the LiDAR and ultrasonic sensors, which are mounted externally, intake external data and send it to the RPi4B.

In order to interface with the user, NaviBot makes use of a touchscreen interface with its main purpose being to obtain destination inputs. It also employs a speaker to warn people of its state, such as something blocking its path. The RPi4B is also responsible for movement of NaviBot, as seen in right-hand side of Fig. 2 and 3, as it interacts with the motor controllers to send forward/backward and speed signals to each of the 2 motors in the system.

The entire NaviBot system is powered through a 24V battery system located underneath the shipping platform, however not all components require or can even handle 24V. For this, a 5V regulator is used. Fig. 2 describes which components will receive 24V and which will receive 5V. For POC, as seen in Fig. 3, a 12V 120W power supply and 12V lead-acid battery will be utilized in place of a full 24V battery system.

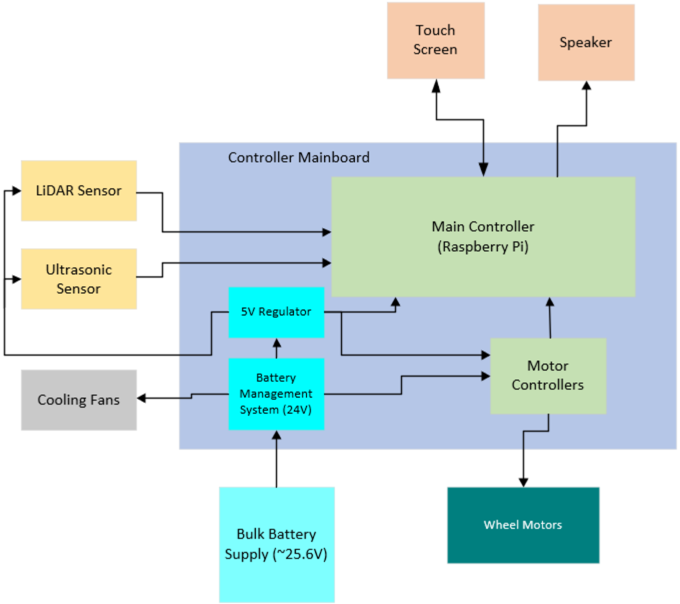


Figure 2: Engineering Prototype System Block Diagram

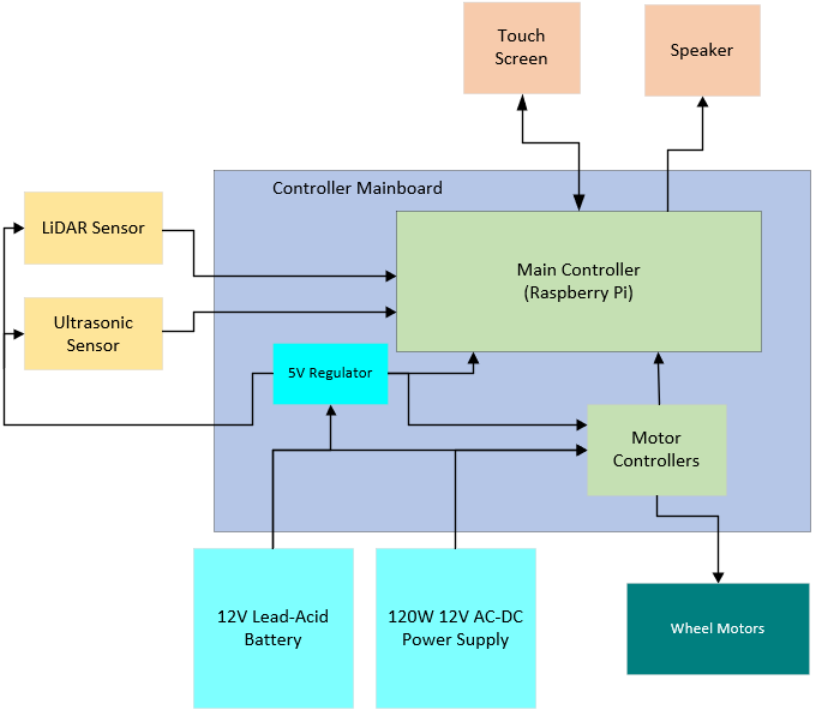


Figure 3: Proof-Of-Concept System Block Diagram

2.1 Design Challenges

The goal of NaviBot is to achieve automated driving whilst carrying a heavy load. As such, the complexity of the project is high, and with complexity comes challenges.

One of these challenges has to do with the thermals of the system. Given the carry capacity is 25kg, this can cause some strain on the motors, and under continuous use at 24V, this will cause some heat buildup. Furthermore, under continuous use, the motor controller and regulator will also generate considerable heat. For the sake of ENSC 405W and 440, so as to not increase the complexity and scope of the project, the thermals will be monitored manually and NaviBot will be stopped to cool down if deemed too hot. The motor controller does have a fan mounting bracket that could be used if necessary.

Other challenges include troubles with the drive mechanism. The current design involves a drive belt reduction system to generate more torque at the wheels. Tensioning these belts to eliminate slippage in the belts or wheels could prove to be challenging and non-trivial. To mitigate this risk, the drive belts are toothed.

Challenges may arise with the odometry and localization in the path planning and navigation of NaviBot. The current plan is to use the LiDAR to fulfill these functions. However, achieving an accurate odometry reading while using the same sensor for SLAM may prove to be difficult. Estimating the current position of the robot in reference to the mapping with only the LiDAR can quickly accumulate positioning error if not attended to.

3 Microcontroller

The microcontroller component is responsible for 3 key functional features of NaviBot. The robot requires to be autonomous relying on the navigation implementation to successfully drive a package to its target location. The user interface displayed on a touchscreen permits the user to assign a task to the robot. Finally the outputs are sent to the driving control from navigation and sensors to complete the delivery. All three parts are controlled by the microcontroller and implemented together using ROS to control device state, localization, and orientation.

3.1 Electrical

The RPi4B requires a power adapter capable of outputting 5V and 3A through USB-C. Once powered, the RPi4B can supply power and connect to the LiDAR through a USB-A port, and the ultrasonic sensor from one of the designated 5V output header pins. Power and ground pins are GPIO pins that are not available for general purpose, as well as pins 27 and 28 (GPIO 0 and GPIO 1 respectively). Any required signals can be sent or received via any of the other available GPIO pins seen in Fig. 5. These available pins are 3.3V pins meaning that any devices requiring voltages different from 3.3V will need to be adjusted.

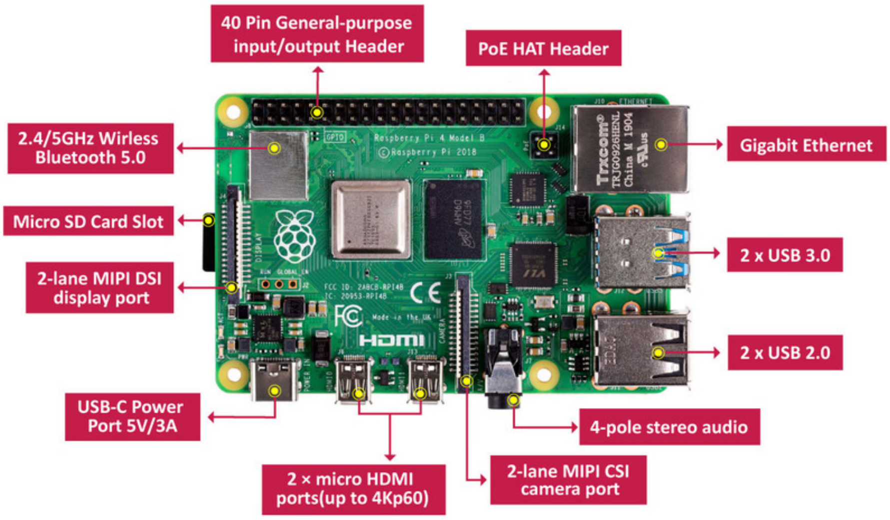


Figure 4: Top Down View of Raspberry Pi 4 Model B [9]

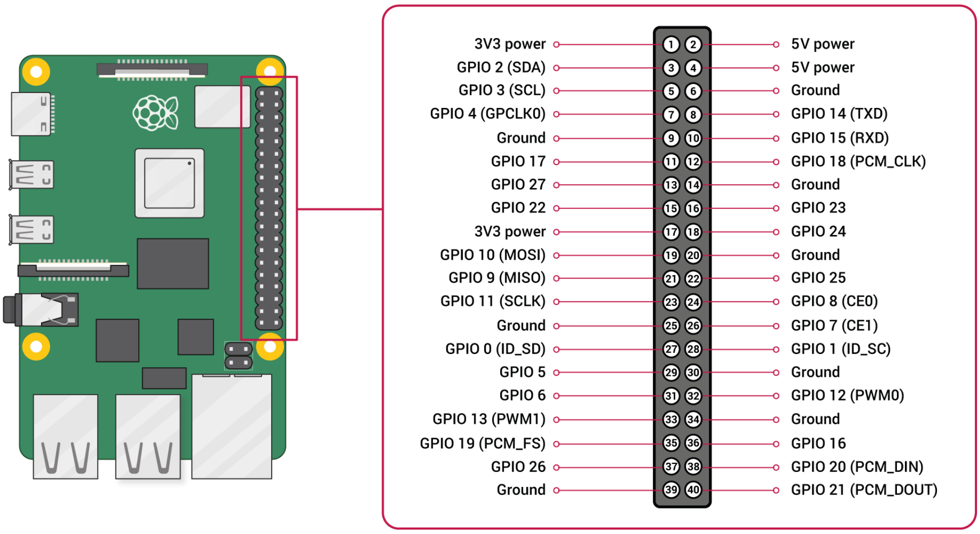


Figure 5: Raspberry Pi 4 Model B Pinout Diagram [10]

As for peripheral accessories, such as the motor controller, speaker, and touchscreen, they will be connected to the RPi4B using 4 GPIO pins, 3.5mm audio jack, and DSI display port respectively. The audio and display will work upon connection whereas the motor controller will need adjusted input values from the GPIO pins to control the speed of the motors.o

Design ID	Requirement ID	Description
Des 3.1.1 A	Req 3.1.1 A, Req 7.2.3 C	The device will run on a Raspberry Pi 4B
Des 3.1.2 A	Req 5.1 A	The motorcontroller voltage outputs will be adjusted to the correct value
Des 3.1.3 A	Req 3.1.2 A	The device will communicate with the LiDAR and Ultrasonic sensor
Des 3.1.4 A	Req 5.2 A	The device will send control outputs to the motor controller
Des 3.1.5 B	Req 3.1.6 B	The device will alert users by sound through the speaker

Table 2: Microcontroller Electrical Design Requirements

3.2 Software

The software integration of all components is connected to a central Raspberry Pi 4 using Linux to operate the ROS library- an open-source tool widely used and supported for robot implementations [1]. The user requests are received from the UI and translated into instructions for the navigation to determine the best course of action. By evaluating the map stored in the memory of the microcontroller and live data feed from the LiDAR and Ultrasonic sensor for local obstacles. The ROS determines the localization and orientation of the robot and outputs the instructions needed to drive the robot to its destination. The software requirements are shown in Table 3.

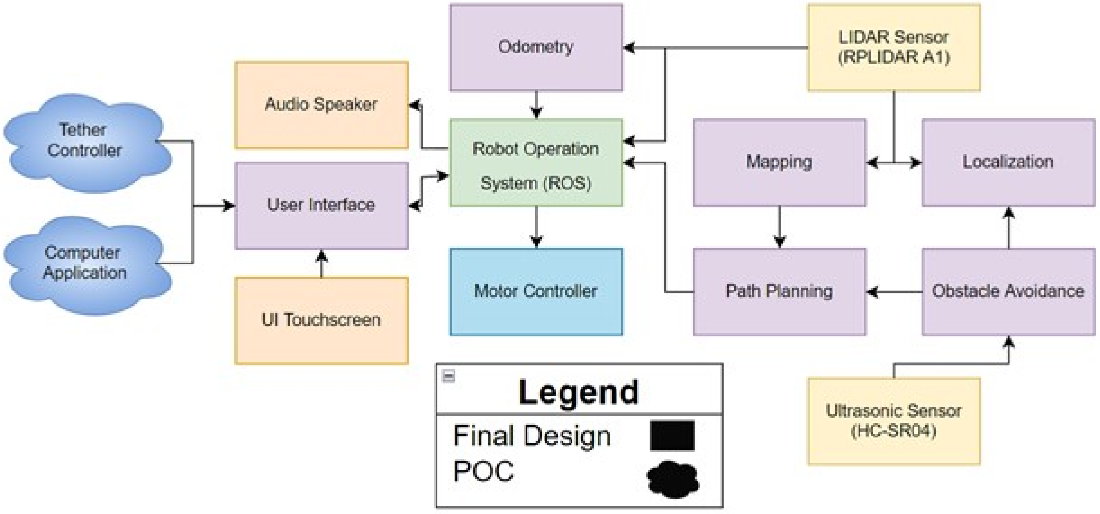


Figure 6: Diagram of the software implementation

Design ID	Requirement ID	Description
Des 3.2.1 A	Req 4.1 A	The device will run using Linux OS
Des 3.2.2 A	Req 4.2 B, Req 4.3 B, Req 4.4 B	The device will run using ROS library
Des 3.2.3 B	Req 4.8 B	The device will detect when in motion
Des 3.2.4 C	Req 4.7 C	The device will indicate when charging is required

Table 3: Microcontroller Software Design Requirements

4 Chassis Design

The chassis or frame design is responsible for housing all necessary components of the robot, will maintaining structure integrity during regular operation, including hauling cargo.

4.1 General Dimensions

In Fig. 7 below, the general exterior dimensions of Navibot are shown. Dimensions regarding the main cargo carrying platform were chosen to maximize available cargo space while conforming to minimum building code dimensions for doorways [13]. Additionally, the control arm seen over top of the robot needed to be positioned to allow for the LiDAR to adequately capture its surroundings without being too large or significantly impeding cargo loading/unloading.

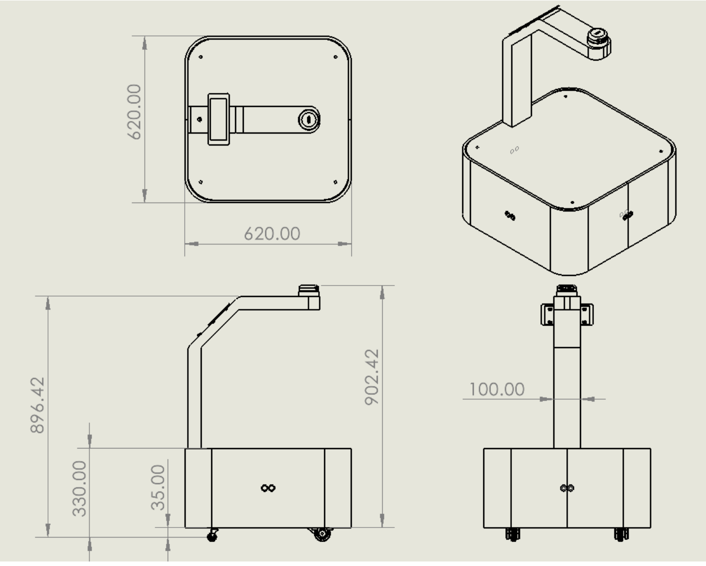


Figure 7: Model Dimensions (mm)

The robot chassis also needed to be large enough to adequately house all required components for the robot's operation. Seen below in Fig. 8 is a proposed internal layout of the robot's components.

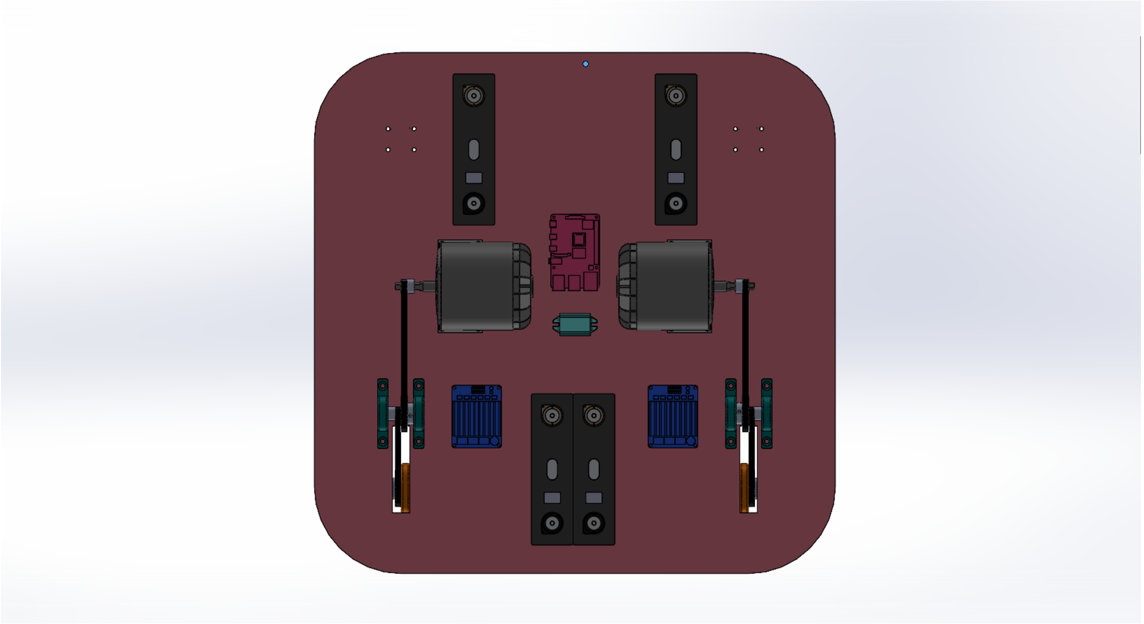


Figure 8: Proposed Internal Component Layout

Design ID	Requirement ID	Description
Des 4.1.2 A	Req 6.3 A	Device will use a straight control arm for positioning of the LiDAR
Des 4.1.3 B	Req 6.3 A	Device will use curved control arm for positioning of the LiDAR and Pi touchscreen

Table 4: Chassis Design Requirements

4.2 Materials

Materials utilized for Navibot's chassis will differ throughout the development process. POC will use MDF as the main baseboard seen in Fig. 8 above due to being easily accessible, machinable, and recyclable. This will be acceptable given POC will not be required to haul cargo. For EP and subsequent revisions, materials such as extruded aluminum or pressed steel sheet will be explored due to its improved mechanical load bearing characteristics as well as being recyclable. Use of petroleum-based plastics will be minimized wherever possible.

The design requirements for the chassis can be seen in Table 5 below.

Design ID	Requirement ID	Description
Des 4.2.1 A	Req 6.1 A, Req 6.2 A, Req 6.4 A, Req 6.5 A,	Device will use a sheet of MDF for the main baseboard of the robot
Des 4.2.2 B	Req 6.6 B, Req 6.7 B	Device will use extruded aluminum and/or sheet steel for chassis materials
Des 4.2.3 B	Req 6.6 B	Device will carry

Table 5: Chassis Design Requirements

5 Drive Mechanism

5.1 Mechanical Design

Navibot utilizes a belt-drive mechanism that is responsible for transferring motor rotation to wheel rotation. Seen below in Fig. 9 and 10 is a CAD model of the respective mechanism.

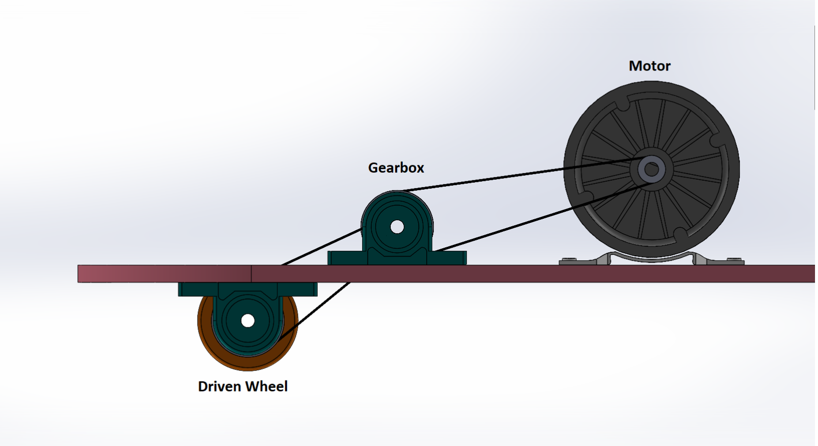


Figure 9: Side View CAD Model of Belt-Drive System

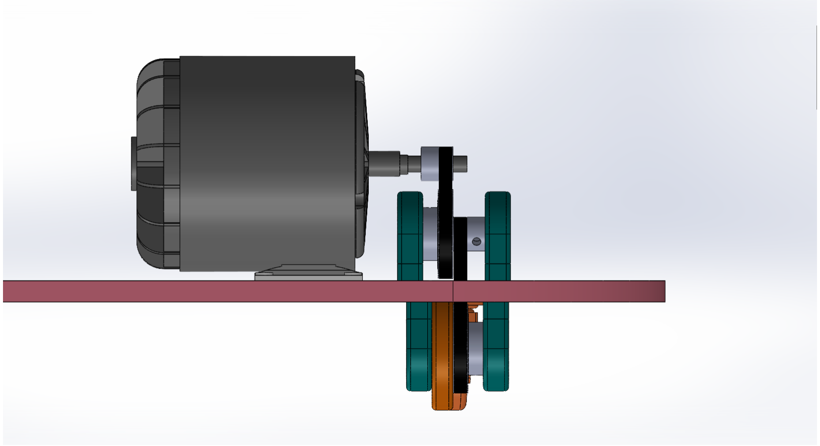


Figure 10: Front View CAD Model of Belt-Drive System

This design allows for a wide range of gear ratios to be utilized depending on motor output and load capacities. The driven wheel responsible for the movement of the robot has fixed mounts with no degree of freedom. The motor and gearbox have one degree of freedom which allow them to be slid back and forth in the plane of the belts to allow for installation, removal, and tensioning of the belts. Toothed belts are used to help avoid belt slip and ensure proper torque transmission during all reasonable load capacities.

5.1.1 Motor Characteristics

For both POC and EP versions, two brushed 24V DC motors (AmpFlow P40-250) will be utilized, as seen below in Fig. 11 with characteristics described in Fig. 12.

Figure 11: AmpFlow P40-250 DC Motor

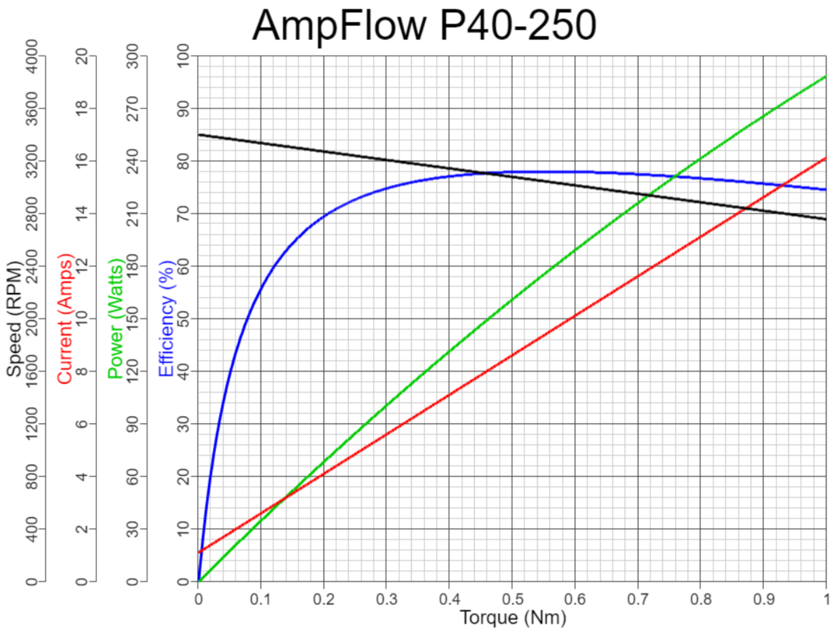


Figure 12: AmpFlow P40-250 DC Motor characteristic plots

For POC the motor will be driven at 12V and as such the motor's characteristics will be about half of what is seen in the above plot. Future work regarding drive-train gearing will target nominal operation at peak efficiencies.

5.1.2 Torque and Speed Calculations

Seen below are calculations for the POC expected peak torque and speed capabilities with the motors driven at 12V and the currently utilized drive-train ratios:

$$Peak\ Motor\ Torque\ with\ 12V = \boldsymbol{0.5N \cdot m}$$

$$Peak\ Motor\ RPM\ with\ 12V = \mathbf{1700}\ \boldsymbol{RPM}$$

$$Driven\ Wheel\ Circumference\ = \mathbf{0.1995}\ \boldsymbol{m}$$

$$Gear\ Ratio = \frac{DrivenToothCount}{DriveToothCount} \cdot NumberOfInstances = \frac{60}{20} \cdot 2 = \mathbf{6}$$

$$Expected\ Peak\ Wheel\ Torque = MotorTorque \cdot GearRatio = 0.5 \cdot 6 = \boldsymbol{3N \cdot m}$$

$$Expected\ Peak\ Wheel\ RPM = \frac{MotorRPM}{GearRatio} = \frac{1700}{6} = \mathbf{283.3}\ \boldsymbol{RPM}$$

$$Expected\ Peak\ Robot\ Speed = \frac{WheelRPM \cdot WheelCircumference}{60\ Seconds} = \frac{283.3 \cdot 0.1995}{60} = \mathbf{0.9419}\ \boldsymbol{m/s}$$

Driving the motors at 12V yields a peak torque of $\sim$ 3 Nm and a top speed of $\sim 0.9419 m/s$. Future revisions, with motors driven at 24V, will yield about double the 12V driven characteristics, with peak torque at $\sim$ 6 Nm and a top speed of $\sim 1.88 m/s$.

5.1.3 Estimated Torque Requirement Calculations

For POC, the robot will be moving without any payload. Operation of the motors at 12V, along with the drive-train gearing, will provide the torque needed to move the weight of the POC.

For EP and subsequent revisions, the robot will be expected to support payloads of up to 25 Kg (55 lbs). As such, overcoming rolling resistance becomes non-trivial. The below calculations explore if the above calculated torque output at 24V is enough to move the robot and additional 25 Kg payload:

$$Estimated\ Coefficient\ of\ Rolling\ Resistance = \mathbf{0.1}\ [9]$$

$$Radius\ of\ Drive\ Wheel = \mathbf{0.03175}\ \boldsymbol{m}$$

$$Estimated\ Mass\ of\ Unloaded\ Robot = \mathbf{20}\ \boldsymbol{Kg}$$

$$Mass\ of\ Supported\ Payload = \mathbf{25}\ \boldsymbol{Kg}$$

$$Additional\ Margin = \mathbf{50\%}$$

$$Required\ Force = (MassOfRobot{+}MassOfPayload){\cdot}g{\cdot}ResistanceCoefficient = (20{\cdot}25){\cdot}9.8{\cdot}0.1 = \boldsymbol{44.1N}$$

$$Required\ Torque = RequiredForce \cdot RadiusOfWheel = 44.1 \cdot 0.03175 = \mathbf{1.4}\ \boldsymbol{N \cdot m}$$

$$Additional\ Margin = 1.4 \cdot 1.5 = \mathbf{2.1}\ \boldsymbol{N \cdot m}$$

The above calculated torque requirement is the minimum torque needed to overcome rolling resistance. EP and subsequent revisions, with motors driven at 24V, will be capable of delivering 3 times the necessary torque to overcome rolling resistance of a fully loaded robot.

Design ID	Requirement ID	Description
Des 5.1.1 A	N/A	Device will use belt drive system for movement of driven wheels.
Des 5.1.2 A	N/A	Device will allow for movement of gearbox and motor for the purpose of belt installation, removal, and tensioning.
Des 5.1.3 A	N/A	Device use a 6:1 mechanical gear ratio.

Table 6: Motor and Drive System Specification

5.2 Motor Control

This section outlines the motor control that NaviBot has to drive the wheels for the POC and EP development stages. An ROS node running on the RPi4B will be in charge of the wheel driving functions of NaviBot, sending PWN signals to the MDDS30 controller to drive the motors forward and backward at a desired speed. NaviBot will accept The table below outlines the software design requirements to drive the motors.

For for the POC and EP versions of NaviBot, the Cytron SmartDriveDuo-30 MDDS30 (herein referred to as "MDDS30") will be installed to control the two 24V DC motors, and can be seen below in Fig. 13. Specifications for the MDDS30 can be seen in table 7.

Figure 13: MDDS30 Motor Controller

Description	Specification
Motor Supply Voltage	7V-35V
Max Continuous Motor Current	30A
Peak Motor Current (1sec)	80A
Logic High	1.3V-5V
Logic Low	0V-0.7V

Table 7: MDDS30 Motor Controller Specifications [2]

The MDDS30 has multiple modes, ranging from radio control, microcontroller, analog/PWM, single or dual motors, and more. For the purposes of NaviBot, it is using an RPi4B to control 2 motors independently through PWM signals. According to the user manual, this requires a DIP switch setting of 10110100, as seen in Fig. 14 below [2].

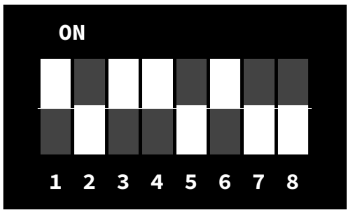


Figure 14: MDDS30 DIP Switch Setting

The MDDS30 has 2 pairs of signal wires that are connected to the RPi4B's GPIO pins. It will receive a software-PWM signal from the RPi4B to modulate the speed of the motors.

In the proof of concept design, the MDDS30 will be supplied power by a 120W 12V AC-to-DC mains converter. This will result in significantly less power at the wheels but will confirm all components work together. For the engineering prototype design, the mains converter and lead-acid battery will be replaced with a full 24V battery and battery management system to ensure maximal performance from motors.

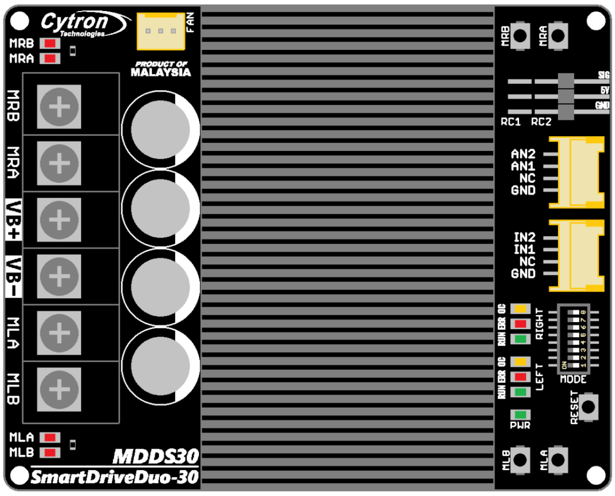


Figure 15: MDDS30 Motor Controller Main Board

Seen below in Table 8 are the design specifications related to motor control.

Design ID	Requirement ID	Description
Des 5.2.1 A	N/A	Device will use MDDS30 Motor Controller to drive the motors with 12V.
Des 5.2.2 B	N/A	Device will use MDDS30 Motor Controller to drive the motors with 24V.
Des 5.2.3 A	N/A	Device will use an ROS node to interface with MDDS30 Motor Controller from RPi4

Table 8: Motor Control Specifications

6 Navigation

6.1 Path Planning

The path planning for NaviBot relies on a few external inputs- maps, ultrasonic sensors, LiDAR sensors for odometry and localization, and goal locations. All of these inputs are sent to a preexisting ROS package called navigation. The block diagram detailing this setup can be seen below in Fig. 16.

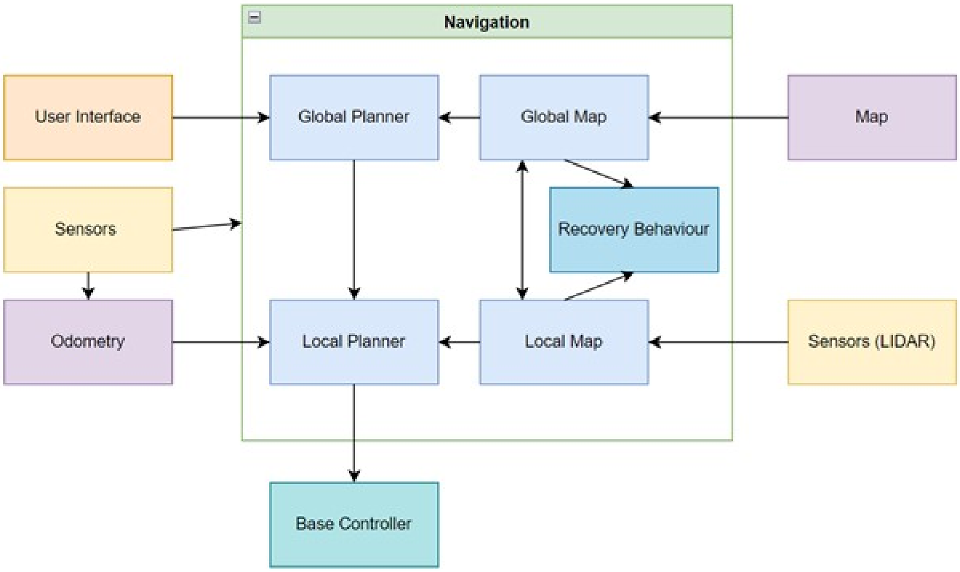


Figure 16: NaviBot Navigation Setup [3]

The main hardware requirements of ROS navigation are [4]:

1. A differential drive robot
2. A planar laser mounted on the robot
3. A roughly square robot

All 3 of these items are fulfilled by NaviBot's design, as it has 2 independent drive wheels, has a centralized LiDAR mounted, and is in a rounded square shape. Meeting these 3 requirements allows for an easy implementation of this package.

As previously stated, the odometry data will be collected with the preexisting LiDAR sensor location at the top center of NaviBot. This could also be completed with wheel encoders, but to reduce project complexity and cost, the LiDAR was ultimately selected.

NaviBot will make use of ultrasonic sensors for collision prevention, which affects the path planning algorithms. For the POC, the sensors will act as a pause on the robot's movement, only resuming when the detected obstacle is cleared. For the EP, NaviBot will use the sensors to stop, wait, and go around the obstacles if they are not cleared within a certain time limit.

The initial stage of implementing the path planning requires the robot to travel through rooms to learn its environment. The POC will connect to a keyboard via bluetooth to manually drive the robot through different locations. In the final design this feature will be done using the touchscreen. Table 9 displays the design requirements needed to implement the path planning.

Design ID	Requirement ID	Description
Des 6.1.1 A	Req 4.6 A	Software will accept manual control via keyboard
Des 6.1.2 A	Req 3.2.2 A	ROS path planning node will traverse robot between 2 points in an clear room
Des 6.1.3 B	Req 3.2.5 B	ROS path planning node will guide robot through doorways
Des 6.1.4 A	Req 3.2.3 A	Software will utilize ultrasonic sensors for collision prevention
Des 6.1.5 C	Req 3.2.4 C	ROS path planning node will guide robot around obstacles

Table 9: Path Planning Design Requirements

6.2 SLAM

SLAM is a method applied on autonomous vehicles that builds and localizes the robot on the map to carry out tasks such as path planning and object avoidance. NaviBot will use a combination of the RPLIDAR A1 and HC-SR04 sensors to implement this practice. During the initializing stage of NaviBot, it will be navigated through the work setting to generate a localized mapping of the workspace area. Next the localization of the robot with respect to the generated map will be calibrated using the LiDAR to predict its position based on local obstacles.

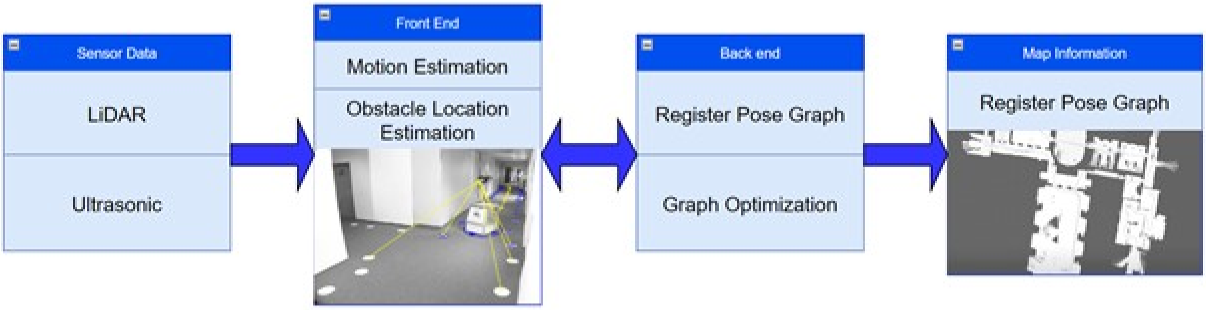


Figure 17: SLAM Processing Flow [5]

The RPLIDAR A1 sensor provides precise data for fast moving devices, low cost in processing and memory requirements, and generates simple 2D mapping [5]. It consists of a pulsed laser connected to a rotating motor to collect data 360 degrees without needing to physically turn the robot to achieve the same results. This device is supported by the ROS library, providing all necessary commands required to carry out the SLAM processing flow. The HC-SR04 ultrasonic sensor will be used to detect live local obstacles to prevent collision. The specifications for both devices are shown below.

Description	Specification
Measuring Range	0.15m–12m
Samples per Rotation	8K
Rotational Speed	5.5 Hz
Angular Resolution	1°
Dimensions	96.8 x 70.3 x 55mm
System Voltage	5V
System Current	100mA
Temperature Range	0℃-40℃
Angular Range	360°
Range Resolution	1% of the range (12m) 2% of the range (12m-16m)
Accuracy	1% of the range (3 m) 2% of the range (3-5 m) 2.5% of the range (5-25m)

Table 10: RPLIDAR A1 Specification [6]

Description	Specification
Measuring Range	2cm–4m
Sampling Frequency	40 Hz
Angular Resolution	15°
Dimensions	45 x 20 x 15mm
System Voltage	DC 5V
System Current	15mA
Trigger Input Signal	10 µS TTL pulse

Table 11: HC-SR04 Ultrasonic Specification [7]

Design ID	Requirement ID	Description
Des 6.2.1 A	Req 4.3 B, Req 4.4 B	The device will generate and save a local map

Table 12: SLAM Design Requirements

6.3 Electrical

The chosen HC-SR04 ultrasonic sensor is designed for Arduino which has 5V GPIO pins. The trigger pin accepts voltages of 3.3V; however, the echo pin (signal output) must be bucked to a voltage of 3.3V through the use of a voltage divider to avoid any potential damage to the RPi4B.

$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

Solving this equation yields a resistance ratio of 1:2 (R1:R2), and R1 and R2 are chosen appropriately as seen in Fig. 19.

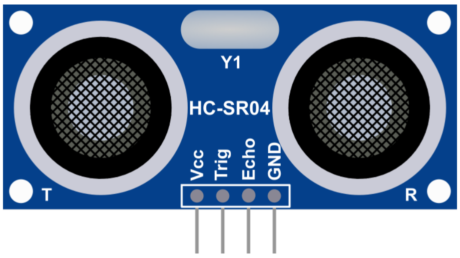


Figure 18: HC-SR04 Sensor Diagram [11]

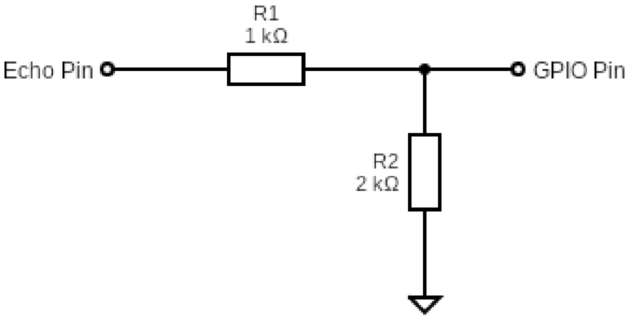


Figure 19: Ultrasonic Sensor Output Voltage Divider

Design ID	Requirement ID	Description
Des 6.3.1 A	Req 3.2.3 A	The sensor will provide close proximity data

Table 13: Electrical Design Requirements for Navigation

7 User Interface

The final design will require the user to interact with NaviBot using the Raspberry Pi 7" Touch Display located on the arm [8]. The UI display will consist of a menu for the destination selection and when it is travelling, it will show the status screen of current task. The mapping process and path planning will occur in the background to allow for a simple user experience. The touchscreen is fully supported by the RPi4B allowing for easy integration.

Design ID	Requirement ID	Description
Des 7.1 B	Req 3.2.8 B	The menu will be displayed on the touchscreen
Des 7.2 B	N/A	The device will allow the user to initialize and edit destinations in the menu
Des 7.3 B	Req 4.5 B	The device will wait on the user confirmation when the item is taken

Table 14: Microcontroller Software Design Requirements

8 Power System

During the proof of concept stage of NaviBot, the device will be powered using an AC-to-DC, 12V, 120W power supply to confirm all the intended functions of the robot can be achieved. A 5V voltage regulator supplies the correct voltage to the sensors. There is an additional 12V lead-acid battery connected in parallel to the motor controller's power input acting as a current sink in situations where back EMF arises from motor slowdown.

Moreover, during the prototype stage, the device will have an on-board 24V battery that will supply all components with power, bucked down to 5V for the sensors. The engineering prototype will have its 12V lead-acid battery removed as the battery management system will handle back EMF and charging. Throughout normal operation, when NaviBot reaches a threshold battery level, its software will proceed to autonomously seek a charging pad which comprises of spring-loaded charging terminals that the robot can drive over to charge itself.

The design requirements for power systems are listed below.

Design ID	Requirement ID	Description
Des 8.1 A	Req 5.1 A	Device will receive sufficient power
Des 8.2 A	Req 5.3 A	Device will utilize mains-to-DC converter
Des 8.3 B	Req 5.4 B	Device will operate using a removable, rechargeable battery source
Des 8.4 B	Req 5.5 B	Device will have a charging station

Table 15: Power System Design Requirements

9 Conclusion

DAM^3 Technologies strives to bring innovation to the logistical management field by providing a support system to reduce the workload on employees. The objective is to implement a reliable and simple to use autonomous robot for companies seeking to increase productivity. NaviBot's autonomous driving will require a mapping of the environment to be produced in order to safely traverse to its destination. Additionally, NaviBot needs to fulfill the requested delivery without impeding or burdening the employee and their work responsibilities. This device aims to provide an affordable solution to reduce unnecessary downtime by the demanding delivery market.

The design specifications stated in this document detail the system implementations required to produce a POC and EP. DAM^3 Technologies will proceed with development by the aforementioned implementation of the chassis design, drive mechanism, navigation system, UI, and power system.

10 References

[1] "Why ROS? It's the fastest way to build a robot!," in *ROS,* 2022. Available: `https://www.ros.org/blog/why-ros/`. [Accessed: 12- Mar- 2022].

[2] "SmartDriveDuo-30 MDDS30," *Cytron Technologies,* 1.11 rev. Cytron Technologies Sdn Bhd, July 2019.

[3] G. Granosik et al., "USING ROBOT OPERATING SYSTEM FOR AUTONOMOUS CONTROL OF ROBOTS IN EUROBOT, ERC AND ROBOTOUR COMPETITIONS", *ResearchGate*, 2016. [Online]. Available: `https://www.researchgate.net/figure/Standard-ROS-Navigation-Stack-7_fig2_312018119`. [Accessed: 12- Mar- 2022].

[4] "navigation," in *ROS,* 2022. Available: `http://wiki.ros.org/navigation`. [Accessed: 12- Mar- 2022].

[5] "What Is SLAM? 3 things you need to know", *MATLAB*, 2022. [Online]. Available: `https://www.mathworks.com/discovery/slam.html`. [Accessed: 12- Mar- 2022].

[6] "RPLIDAR A1", *SLAMTEC.* [Online]. Available: `https://www.slamtec.com/en/Lidar/A1Spec`. [Accessed: 12- Mar- 2022].

[7] "Ultrasonic Ranging Module HC - SR04", *ELEC* Freaks. [Online]. Available: `https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf`. [Accessed: 12- Mar- 2022].

[8] "Raspberry Pi Touch Display", Raspberry Pi. [Online]. Available: `https://www.raspberrypi.com/products/raspberry-pi-touch-display/`. [Accessed: 12- Mar- 2022].

[9] "Raspberry Pi 4 Model B 4Gb", *Robot Advance.* [Online]. Available: `https://www.robot-advance.com/EN/art-raspberry-pi-4-model-b-4go-2640.htm`. [Accessed: 12- Mar- 2022].

[10] "Raspberry Pi Documentation - Raspberry Pi OS", *Raspberry Pi.* [Online]. Available: `https://www.raspberrypi.com/documentation/computers/os.html`. [Accessed: 12- Mar- 2022].

[11] "HC-SR04 Ultrasonic Sensor", *Components101,* 2021. [Online]. Available: `https://components101.com/sensors/ultrasonic-sensor-working-pinout-datasheet`. [Accessed: 12- Mar- 2022].

[12] "Wheels and Castors Guide", *Blickle.* [Online]. Available: `https://www.stalus.si/PortalGenerator/document.aspx?ID=100&Action=2&UserID=0&SessionID=0&NavigationID=0`. [Accessed: 12- Mar- 2022].

[13] "Building Code requirements," *Ontario.ca,* 2019. [Online]. Available: `https://www.ontario.ca/document/build-or-buy-tiny-home/building-code-requirements`. [Accessed: 12- Mar- 2022].

A Design Alternatives

A.1 Microcontroller

A.1.1 Microcontroller Choice

Per Req 3.1.1 A (Robot will utilize a microcontroller), multiple hobbyist controllers come to mind, such as Raspberry Pi, or Nvidia Jetson. The Raspberry Pi 4B was ultimately chosen as it is more powerful than the Arduino, and more readily available for DAM^3 Technologies than the Nvidia Jetson. Initially a RPi3B was used, however the 1GB of RAM was determined to be insufficient, so a migration to a RPi4B was performed.

A.1.2 Raspberry Pi and Arduino

Should the RPi4B struggle to run all required functions, an Arduino may be employed to handle certain functions such as the motor controlling. This will require the Arduino to interface with the RPi4B, but will reduce the over RPi4B workload. It will also allow for easier coding of the motor controlling on the Arduino. The current design choice is to use the RPi4B for all computations to meet Req 3.1.1 A (Robot will utilize a microcontroller), which implies a requirement of using only 1 controller.

A.2 Drive Mechanism

It is advised to stay away from tension belt driven systems as they incur more power losses due to poor tensioning of the belt and gear tooth and belt skipping. A step-down gearing system could serve as the next mobility system as it does not suffer from poor tensioning losses; however, per Des 5.1 A (device will use a tension belt drive system), the tension belt system is chosen as parts available to purchase are more abundant than geared systems and any specific parts could be 3D printed to support the lack of available parts. The power transferred from motor to wheel appears small enough to justify the tension belt system.

A.3 Navigation

A.3.1 POC Manual Control

The POC for NaviBot requires manual control, per Req 4.6 A (Software will be hard coded to move the robot). To fulfill this, a wireless keyboard can be directly attached to the RPi4B, or a laptop can remote connect to it. For simplicity, a wireless keyboard was selected, however, if there are problems with it the laptop can be easily transitioned to.

A.3.2 Odometry

The most common and easily implementable odometry options for NaviBot are wheel encoders or LiDAR. While the wheel encoders may result in the most accurate solution, it also increases the scope of the project by adding more sensors. As such, the preexisting LiDAR was chosen to gather the odometry data for NaviBot. This also allows the design to remain consistent with the Requirements Specifications document, which frequently requests the use of LiDAR for positioning and mapping, such as in Req 3.1.5 B.

A.3.3 Collision Prevention

Some options for collision detection and prevention are camera vision, ultrasonic sensors, or LiDAR. Per Req 3.2.3 A (Robot will have collision prevention) and Req 3.1.2 A (Robot will interface with LiDAR and ultrasonic sensors), the choice narrows down to either ultrasonic or LiDAR. NaviBot's LiDAR is mounted on the top centre of the control arm, and thus creates a blind spot around the base of the robot. The LiDAR's optimal location is at the top centre, so the ultrasonic sensors were ultimately chosen to be mounted around the base of NaviBot to cover that blind spot and allow for smaller object detection such has boxes or plant pots.

B Test Plan

Test: Power Supply
Tested Requirements: Des 8.1 A, Des 8.2 A, Req 5.1 A, Req 5.3 A
Procedure: Connect the device to an appropriate power source.
Expected Outcome: Connect the device to an appropriate power source. LED indicator will turn on, indicating power is being supplied to the microcontroller.

Test: Ultrasound Sensor
Tested Requirements: Des 6.1.1 A, Req 3.2.3 A
Procedure: Obstruct the sensor.
Expected Outcome: LED indicator will turn on, indicating that an obstruction is detected.

Test: Touchscreen
Tested Requirements: Des 7.1.1 B, Des 7.1.3 B, Req 3.2.8 B, Req 4.5 B
Procedure: Select options on the touchscreen.
Expected Outcome: Touchscreen application is responsive and smooth.

Test: Charging Station
Tested Requirements: Des 8.3 B, Des 8.4 B, Req 5.4 B, Req 5.5 B
Procedure: Connect device to charging station.
Expected Outcome: LED indicator will turn on to indicate successful charging status.

Test: Mobility
Tested Requirements: Des 3.1.4 A, Des 5.1 A, Des 5.2 A, Des 5.3 A, Des 5.4 A, Des 5.5 A, Des 5.6 A, Des 5.7 B, Req 5.2 A
Procedure: Select a default test travel destination.
Expected Outcome: Robot navigates to, and successfully stops at destination.

Test: Undetected Obstruction
Tested Requirements: Des 3.1.3 A, Des 3.1.4 A, Des 3.2.3 B, Req 3.1.2 A, Req 5.2 A, Req 4.8 B
Procedure: Robot makes contact with an obstruction unseen by sensors.
Expected Outcome: Robot will determine if it has not moved locations when it currently should be moving and begins reversing.

Test: Load Capacity
Tested Requirements: Des 4.2.3 B
Procedure: Robot is mobile with a 5kg, 10kg, 20kg, 25kg load
Expected Outcome: Robot will be able to travel to its goal destination under each load.

C User Interface and Appearance Design

C.1 Introduction

NaviBot aims to aid logistics efforts all the while being easy to set up and use. DAM^3's philosophy is that the device should always add to the performance of an operation and not hinder. The interface thus aims to be as intuitive and fluid as possible allowing for minimizing the learning curve and maximizing performance on the floor.

C.1.1 Purpose

The purpose of this document is to outline the design of NaviBot's user interface (UI) design.

C.1.2 Scope

This document explores 7 sections of NaviBot's UI design as follows:

1. Graphical Representation
2. User Analysis
3. Technical Analysis
4. Engineering Standards
5. Safety and Sustainability
6. Empirical Usability Testing
7. Analytical Usability Testing

C.2 Graphical Representation

A full CAD representation of NaviBot can be seen in Fig. 20 to 22 below. Fig. 20 and 21 represent what the end-user will see and interact with during regular operation.

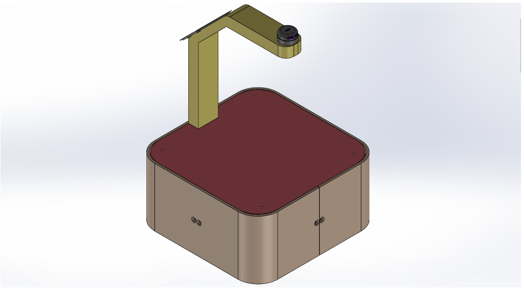


Figure 20: Complete CAD of NaviBot Visualization

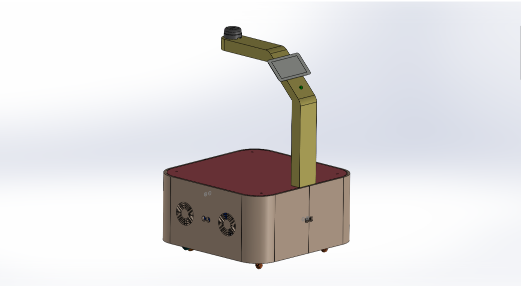


Figure 21: Visualization of Touchscreen Location

Fig. 22 below represents a potential layout of internal components, including the mechanical design for robot propulsion. Navibot's internals are designed to be both easy to service and accessible.

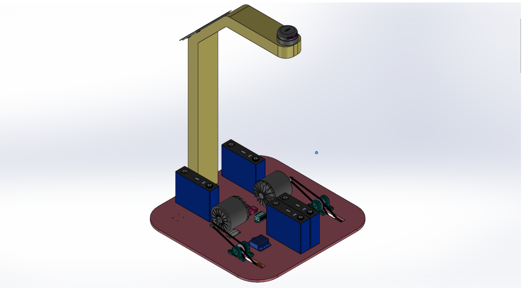


Figure 22: Visualization of Internal Layout

The primary mode of interaction between user and device will be through its mounted touch screen display. It will provide details of potential actions ready to be performed by the robot.

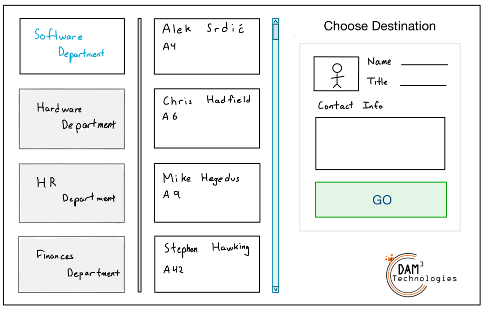
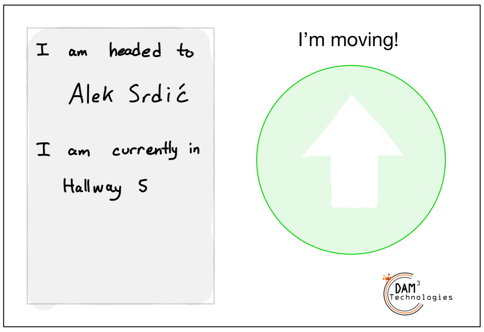


Figure 23: Room Selection Screen



I am headed to

Alek Srdić

I am currently in

Hallway 5

Figure 24: NaviBot Status Screen

After calibration and setup, the main screen will display the potential delivery destinations NaviBot may navigate to per Fig. 23. Each location may display additional information if desired such as room number or department name. After a command is received by NaviBot, the screen will display its current status to alert passerby of current actions as well as its destination as seen in Fig. 24.

C.3 User Analysis

The target users of NaviBot will be primarily for logistics or shipping and handling management staff. Usability of the autonomous robot will be similar to navigating through a menu on a smartphone application. This consists of scrolling through a list of saved locations and selecting a destination to send the robot with the package.

The initial setup requires the user to know how to follow the instructions on the robot and guide it to various locations to effectively train it. Once it is guided to a location of the users’ choice, they will have to save the location through the menu. Achieving the most effective driving automation will require the initial installer to direct the robot around the entirety of the environments domain to generate a detailed mapping.

The physical requirements to operate the device will be to reach the platform of robot roughly one foot off the ground. In the case of premature battery depletion, it will be required to push the robot back to a charging station.

C.4 Technical Analysis

NaviBot's design aims to be as non-obstructive and intuitive in its interaction with users through the implementation of the *Seven Elements of UI Interaction*: discoverability, feedback, conceptual models, affordances, signifiers, mappings, and constraints [1]. By designing NaviBot's UI with these considerations, it will remain accessible to the user, maintain clear and consistent visuals, contain well-organized menus, and impress upon the user that they are ultimately in control of the device through its menu.

C.4.1 Discoverability

In order to implement discoverability in NaviBot's design, all of its interfaces and buttons will be located on a main control arm rising up and over the center of the transport platform. The first thing a user will want to do with NaviBot is power it on, hence the power button will be in plain sight next to the touch screen and indicated with the IEC 60417-5009 [2] standby symbol, which is commonly used for computers. In the event of an error, a red emergency stop button will also be located in on the other side of the touch screen. In terms of NaviBot's touch interface, the user will be presented with a set of descriptive buttons, allowing them to access different menus to send the robot to different rooms or personnel. On initial startup, however, the user will be presented with the first-time setup screen for environment mapping to ensure that step is not missed.

C.4.2 Feedback

Feedback is used to let the user know that their action has been or is currently being processed by NaviBot. Given NaviBot's complexity, this can come in many forms such as status symbols, sound cues, or loading screens. The following will be used as visual feedback on NaviBot: a single colour LED regarding power status, a multicoloured charge status LED which will flash yellow while charging and stay solid green at full battery, a pop-up warning when the weight limit is exceeded, loading screens to inform the user that their request is being processed, and a status screen when NaviBot is on the move. In addition to visual feedback, NaviBot will make use of auditory feedback including sound bites after a touchscreen press and audio queues when it detects an obstacle has moved in front of it.

C.4.3 Conceptual Models

NaviBot is designed with the principle of conceptual models in mind. A user may want to operate NaviBot for the first time and will begin with an idea of how they think it should work. As such, all controls will be in a readily available control stack located above the center of the transport plate. In addition, touch screen menus will be as simple and intuitive as possible to eliminate any guesswork by the user.

C.4.4 Affordances

Affordances in the context of NaviBot are what a user is capable of doing with the robot which may or may not necessarily align with its intended use. NaviBot design will attempt to discourage users from any misuse whilst teaching them its intended use. One way to accomplish this will be to have readily available instruction menus on the robot. For example, the environment initialization setup guides the user step by step through the process, helping them learn about the robot along the way. Designing for the unintended use will be outlined in the Constraints section.

C.4.5 Signifiers

Signifiers on NaviBot are used to specifically indicate the purpose of a button or function, and can closely relate back to the feedback section. Each button will be labeled with a common symbol or label and any necessary LEDs. For example, the power button will be labeled with the IEC 60417-5009 stand-by symbol to indicate a power button, and will be illuminated with an LED when NaviBot is powered on. On the touch screen interface, buttons will be clearly identifiable and uniquely labeled. Weight capacities will also be labeled on the robot.

C.4.6 Mappings

Mapping design with NaviBot pertains mainly to the touch screen interface. To incorporate mapping in the UI design, only relevant information is displayed on the touch screen after a selection is made. If any buttons are next to each other, the labels for those buttons will clearly indicate which button is for which function. This is the case for both hardware and software buttons.

C.4.7 Constraints

Constraints are meant to prevent the user from misusing NaviBot. To ensure nobody sits on the platform, the control panel is located on an arm above the transport plate, making it more uncomfortable to sit on. Furthermore, a load sensor and will be installed to prevent NaviBot from burning out its motors if the weight is over capacity. A notification will also be displayed on the touch screen that the weight limit has been exceeded and audio feedback will be played. NaviBot requires an initial environment setup, so to ensure users perform this step, first time startup will open on the initialization screen and provide instructions and controls required. A quick setup guide will also be provided to ensure the first time use goes as intended.

C.5 Safety and Sustainability

Safety and sustainability is a top priority at DAM^3 Technologies. In order to create a long lasting impact on today's market while minimizing waste, certain measures will be taken. A more detailed description is explained in the following.

C.5.1 Safety

Given that NaviBot is an autonomously navigating robot, it needs to seamlessly work alongside individuals whilst avoiding collisions. Since NaviBot will be frequently interacted with, all circuitry should be enclosed and monitored to prevent any hazards such as overheating and electric shock. Furthermore, any and all sharp edges will be eliminated on the exterior, and mitigated on the interior with warning stickers indicating any remaining edges.

C.5.2 Sustainability

During development of NaviBot, sustainability will be achieved by allowing companies the "Right to Repair" through use of easily accessible and replaceable components to extend the employment of the robot. Along with easy maintenance, DAM^3 Technologies wholly believes in "Cradle-to-Cradle" sustainability, and as such, biodegradable and recyclable materials will be sourced for the product.

C.6 Engineering Standards

Table 16 shows the Engineering standards chosen for NaviBot. The left-hand column represents the Standard ID and Engineering organizations. The right side gives a description of the following standards.

IEEE 1872-2015	Standard Ontologies for Robotics and Automation [3]
IEEE 2755.2-2020	Recommended Practice for Implementation and Management Methodology for Software-Based Intelligent Process Automation [4]
IEC 62133:2012	Secondary cells and batteries containing alkaline or other non-acid electrolytes - Safety requirements for portable sealed secondary cells, and for batteries made from them, for use in portable applications [5]
ISO 9787:2013	Robots and robotic devices — Coordinate systems and motion nomenclatures [6]
ISO 18646-1:2016	Robotics — Performance criteria and related test methods for service robots — Part 1: Locomotion for wheeled robots [7]
ISO 13482	Safety requirements for service robots [8]
ISO 9241-810:2020	Ergonomics of human-system interaction — Part 810: Robotic, intelligent and autonomous systems [9]
ISO 11136:2014	Sensory analysis — Methodology — General guidance for conducting hedonic tests with consumers in a controlled area [10]
ISO 23482-2:2019	Robotics — Application of ISO 13482 — Part 2: Application guidelines [11]
S.C. 2010, c. 21	Canada Consumer Product Safety Act [12]

Table 16: Engineering Standards

C.7 Empirical Usability Testing

NaviBot's performance will be characterized through empirical usability testing. Usability performance will be based upon user feedback of the robot during its operation and include sections such as speed of delivery, ability to navigate, and carrying capability. Feedback will be obtained from key points during the robot's operation.

Operation 1: Power On and Initial Setup

1. Was the power button easy to locate and operate?
2. Were you able to enable the robot setup and mapping?
3. Did the robot accurately map the environment?
4. Were you able to easily assign delivery locations based on map data?

Operation 2: Location Selection and Loading of Cargo

1. Was the delivery location easy to find and select?
2. Was cargo able to be easily loaded with regard to ergonomics?
3. Was the robot's carrying capability adequate?

Operation 3: Delivering of Cargo

1. Does the robot deliver the cargo in a timely manner?
2. Was the robot able to properly navigate its environment?
3. Did the robot adequately notify the recipient of their delivery?
4. Did the robot return in a timely matter?

Overall Evaluation

1. Was the robot's battery life sufficient for your needs?
2. Did you find NaviBot to be a hindrance to your work?
3. Would you use NaviBot to aid in your day-to-day delivery responsibilities?
4. Is there any other feedback you would like to express?

C.8 Analytical Usability Testing

This section outlines the analytical testing procedures to be undertaken by the members of DAM^3 Technologies. This series of tests will determine whether the user interface of the Navibot performs expected behaviour and any detected design flaws will be brought to the attention of all team members.

Package Platform

1. The platform is flat with respect to the floor
2. Items within recommended package dimensions can be safely placed on the platform
3. An LED illuminates when the robot is on

Touchscreen Application

1. The screen displays the logo of the company
2. The screen displays a list of potential recipients/locations
3. The robot confirms the selected recipient with the user before departing

Speaker

1. The robot notifies the user when over the weight limit
2. The robot requests people to move if obstructing its path
3. Triggers an alarm if wheels no longer make contact with the ground

Charging Station

1. The power cord supplies power to the station
2. An LED turns on to indicate charging status

C.9 Conclusion

NaviBot's interface aims to allow the user full capability to do what they want and understand what the device is doing. It will support easy to read labels for ease of use, properly configured physical and software buttons, and functionality that does not increase risk of injury; as safety and sustainability are at the top of DAM^3's priorities. Analysis procedures were structured around its potential for maximum feedback to the developers. As a complete package, NaviBot's integration to offices and warehouses will be seamless.

C.10 References

[1] D. Norman, The Design of Everyday Things. Basic Books (AZ), 2013.

[2] “IEC 60417 — Graphical Symbols for Use on Equipment,” Online Browsing Platform (OBP), 14-Mar-2015. [Online]. Available: https://www.iso.org/obp/uiiec:grs:60417:5009. [Accessed: 02-Mar-2022].

[3] "IEEE Standard Ontologies for Robotics and Automation," in *IEEE Std 1872-2015*, vol., no., pp.1-60, 10 April 2015, doi: 10.1109/IEEESTD.2015.7084073 [Accessed: 01-Mar-2022].

[4] "IEEE Recommended Practice for Implementation and Management Methodology for Software-Based Intelligent Process Automation," in *IEEE Std 2755.2-2020*, vol., no., pp.1-59, 14 April 2021, doi: 10.1109/IEEESTD.2021.9404959 [Accessed: 01-Mar-2022].

[5] "IEEE Standard for Robot Map Data Representation for Navigation," in *1873-2015 IEEE Standard for Robot Map Data Representation for Navigation*, vol., no., pp.1-54, 26 Oct. 2015, doi: 10.1109/IEEESTD.2015.7300355 [Accessed: 01-Mar-2022].

[6] “Robots and robotic devices – Coordinate systems and motion nomenclatures,” 3rd ed. May 2013. [Online]. Available: https://www.iso.org/standard/59444.html [Accessed: 01-Mar-2022].

[7] “Robotics – Performance criteria and related test methods for service robots – Part 1: Locomotion for wheeled robots,” 1st ed. September 2016. [Online]. Available: https://www.iso.org/standard/63127.html [Accessed: 01-Mar-2022].

[8] “Robotics – Safety requirements for service robots,” 2nd ed. [Online]. Available: https://www.iso.org/standard/83498.html [Accessed: 01-Mar-2022].

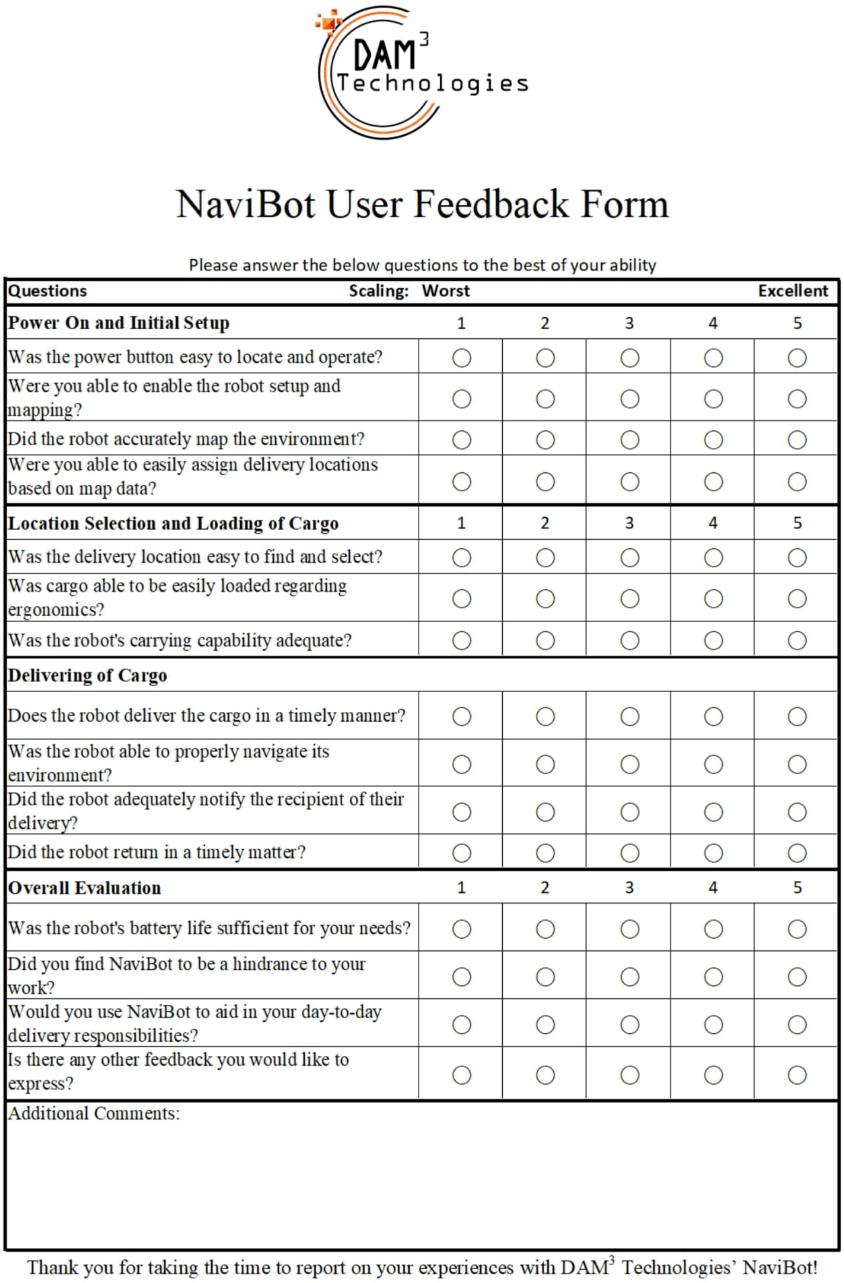
[9] “Ergonomics of human-system interaction – Part 810: Robotic, intelligent and autonomous systems,” 1st ed. September 2020. [Online]. Available: https://www.iso.org/standard/76577.html [Accessed: 01-Mar-2022].

[10] “Sensory analysis – Methodology – General guidance for conducting hedonic tests with consumers in a controlled area,” 1st ed. December 2014. [Online]. Available: https://www.iso.org/standard/50125.html [Accessed: 01-Mar-2022].

[11] “Robotics – Application of ISO 13482 – Part 2: Application guidelines,” 1st ed. March 2019. [Online]. Available: https://www.iso.org/standard/71627.html [Accessed: 01-Mar-2022].

[12] *Canada Consumer Product Safety Act*, C.R.C., c. 1370, Dec. 15, 2010. [Online]. Available: https://laws-lois.justice.gc.ca/eng/acts/C-1.68/page-1.html [Accessed: 01-Mar-2022].

C.11 Empirical Usability Feedback Form



DAM^3 Technologies

NaviBot User Feedback Form

Please answer the below questions to the best of your ability

Questions **Scaling:**	**Worst**				**Excellent**
Power On and Initial Setup	1	2	3	4	5
Was the power button easy to locate and operate?	○	○	○	○	○
Were you able to enable the robot setup and mapping?	○	○	○	○	○
Did the robot accurately map the environment?	○	○	○	○	○
Were you able to easily assign delivery locations based on map data?	○	○	○	○	○
Location Selection and Loading of Cargo	1	2	3	4	5
Was the delivery location easy to find and select?	○	○	○	○	○
Was cargo able to be easily loaded regarding ergonomics?	○	○	○	○	○
Was the robot's carrying capability adequate?	○	○	○	○	○
Delivering of Cargo					
Does the robot deliver the cargo in a timely manner?	○	○	○	○	○
Was the robot able to properly navigate its environment?	○	○	○	○	○
Did the robot adequately notify the recipient of their delivery?	○	○	○	○	○
Did the robot return in a timely matter?	○	○	○	○	○
Overall Evaluation	1	2	3	4	5
Was the robot's battery life sufficient for your needs?	○	○	○	○	○
Did you find NaviBot to be a hindrance to your work?	○	○	○	○	○
Would you use NaviBot to aid in your day-to-day delivery responsibilities?	○	○	○	○	○
Is there any other feedback you would like to express?	○	○	○	○	○
Additional Comments:					

Thank you for taking the time to report on your experiences with DAM^3 Technologies' NaviBot!