4Dog GuideBot: Vision Companion

Cole Smith
Emmett Osborne

Montana State University
CSCI 482/483

Abstract

Guide dogs have long been invaluable aids for the blind and visually impaired, ensuring safe navigation through various environments. Despite their benefits, guide dogs also necessitate continuous care, including feeding and attention, and can sometimes be distracting in certain settings. Addressing these challenges, we propose an innovative alternative: an automated robotic guidance vehicle tailored for visually impaired students at Montana State University designed to operate outdoors. This robotic solution emulates the functions of a guide dog, employing advanced features such as location tracking, obstacle detection, and an intuitive speech to location feature. The integrated software manages robot movement, processes LiDAR data for real-time obstacle detection, interprets positional information for advanced path planning which are interpreted via voice commands on the bot for hands free use.

Introduction

Nearly 20 million Americans (8% of the U.S. population) have visual impairments, with a projection of significant increase over the coming decades [Georgetown University, 2019]. According to the World Health Organization, who defines visual impairment as any eye condition that affects the visual system and its functions, this number is close to 217 million globally. This means that visual impairment can range from reduced visibility all the way to complete blindness, and it exists on a massive scale. The more severe cases of visual impairment can significantly reduce the independence of those with the disease, leading them to rely on other people or service animals to navigate their surroundings, if that is even possible at all.

Guide dogs are one of the common solutions for helping those with visual impairment to regain some independence, however guide dogs come with numerous downsides that make them a subpar solution. To start, guide dogs are an enormously expensive investment, costing up to $50,000 *annually* to train and care for a guide dog throughout its working lifetime [Guide Dogs 101, 2021]. Furthermore, almost “half of the dogs bred to become guide dogs for blind people fail before the end of their training, the main cause being the presence of fear”, meaning that nearly 50% of the efforts towards training guide dogs are wasted entirely [Menuge et al, 2021]. In addition to the base cost and failure rates, there are several constraints that can make guide dogs even more unfavorable. Some considerations include: physical health of the owner (being able to maintain and keep pace with the dog), home environment, keeping distinction of pet vs. worker, and animal access restrictions in certain areas [Guide dogs are not for all visually impaired people, 2021]. Essentially, guide dogs are animals. They require food, cleanup, training, attention, attention to health, and good human/animal compatibility. They are both expensive and difficult to maintain, making them either unavailable or a poor option for many people.

The inability to either obtain or properly utilize a guide dog has led us to consider a different option: robotics. The idea is to make a pseudo-companion that functions similarly to a guide dog in providing independence to those with visual impairments without some of the downsides. A robotic system would be far cheaper, available to anyone, customizable, and wouldn’t require the same degree of care that an animal would. While this system wouldn’t offer the same companionship that a good human-dog match could, its primary goal is to offer navigational independence with far less headache. This idea has led us to the 4Dog Guidebot: a robot system that can help those with visual impairment by safely guiding them to desired destinations.

Qualifications

Emmett Osborne

1433 Tempest Court
Bozeman, MT 59718
+1 (307) 578-6390
EmmettOsborne.01@gmail.com

EXPERIENCE

Advanced Electronic Designs — *Engineering Intern*

May 2021 - Now

Collaborated in a dynamic team of over 20 professionals, spearheading the design, construction, and prototyping of custom electronics and FPGA systems. Contributed to the development and refinement of manufacturing processes.

Demonstrated extensive expertise in through-hole and surface mount component installation and rework, underpinned by a strong proficiency in quality control, imaging, and inventory management of components and products.

Led diverse software projects, including the redevelopment of firmware for lighting drones using C++, creation of OCR-based automation software in Python, and development of custom PCB and firmware for LiDAR sensors and door controllers using C/C++. Additionally, designed and implemented a bespoke data visualization tool utilizing Python.

Cody High School — *Technology Specialist*

April 2019 - August 2019; June 2020 - August 2020

Experienced in performing computer maintenance, software development and deployment, systems architecture, and networking.

PROJECTS

Lidar Door Sensor / Controller - *Custom PCB / Firmware*

January 2023 - November 2023

Collaborated in the design of a custom PCB for a LiDAR sensor, and developed the firmware to automate sliding glass doors for AED.

Tripedal Robot - *Custom Robotics System*

August 2022 - Now

Designing and building a custom tripedal robotics system to demonstrate the functionality of a unique robotic gait.

EDUCATION

Montana State University — B.S. *Computer Science & CompE Minor*

August 2020 - December 2024

SKILLS

Programming Languages:

Proficient in C, C++, C#, Java, Python, JavaScript, HTML, CSS

Experienced with VHDL and Assembly Language

Design and Modeling:

3D Modeling and Printing

Proficient in Altium Designer

Skilled in Adobe Creative Suite

Certifications and Licenses:

Technician Class Operator, Amateur Radio License

REFERENCES

Available upon request.

ACTIVITIES

Personal Robotics & Electronics Projects
Tripedal Robot / Gauss Rifle
Electric Scooter / 3D Models

Montana Experimental - Rocketry (Past)
Lead Member

Bridger Solar Team (Past)
Member

CHS Robotics Team (Past)
Leader/Co-Founder/Coach

Cole Smith, Chemist. Programmer. Problem Solver

ca7smith@gmail.com • 406 600 4680

Information

I am a passionate computer science student with a background in chemistry and microbiology. Received straight A's on all software development coursework. Seeking to use my skills and talents to solve problems that arise in software design and development.

EDUCATION & CREDENTIALS

Bachelor of Science in Computer Science, minor in Microbiology
Montana State University, Bozeman, Montana *Fall 2022 - Spr. 2023*

Bachelor of Science in Chemistry
Montana State University, Bozeman, Montana *Fall 2016 - Fall 2020*

PROGRAMMING EXPERIENCE

Student, *Montana State University, Bozeman, MT* *Aug 2022 - Present*

- Experienced problem solver in Python interpreting, long term data, with visual presentation.
- Experienced object-oriented programmer with sensitive data sets to protect client information and access information rapidly, and efficiently.
- Implements high level languages such as C to parse data systematically and analyze predicted models.
- Web design knowledge for implementing client interest in particular products they want to look professional.

LABORATORY EXPERIENCE

Michael Mocks Lab, *Montana State University, Bozeman, MT* *Aug 2020 - Jan 2021*

- Researching Chromium, Cr, catalytic reduction of nitrogenous ligands to help understand the bonding energy, and dissociation of key proteins in our soils that make ammonia
- Developed scientific methods to assess potential different charged Cr metal complexes for the reductive elimination of N2 into ammonia. We assessed Cr[PCPtBu2][I][N] complexes by density functional theory, to calculate orbital symmetry and energies of the above metal complex to predict our model's behavior under selected solvents.
- Rigorous comparison of bonding characteristics to the catalytic reactivity of Cr metal centers helped us develop solvents that favored binding of N2 ligands, we were interested in.
- Computational models were applied to our research to help facilitate the experimental outcomes before we investigated resources. Most of our energy calculations were done computationally on a very similar Cr metal complex.

HONORS & ACTIVITIES

Michael Mocks Lab, *Montana State University* *Aug 2020 - Jan 2021*

Dean's list, *Montana State University* *Fall/Spring 2019, Fall 2020, Spring 2022*

SKILLS

Languages: Python, C, C++, Java, Go, Fortran,

Technical: Software engineering, data structures, advanced algorithms, computational biology, security, embedded systems robotics, machine learning,

Design: Excel for electron spectra processing, html, css

Alternative: Automotive repair, engine electrical system diagnostics, welding, phlebotomy, computational chemistry, leadership, communication,

Background

Other tools, projects and research to tackle this issue :

- WeWALK [1]
  - Smart cane developed to enhance mobility for visually impaired individuals. Integrates smartphones and uses ultrasonic sensors to detect obstacles and provide haptic feedback to the user
- Project Vizzy [2]
  - Developed by IBM, a small robot designed to assist the visually impaired. Uses cameras and sensors to navigate the environment, providing audio descriptions of the surroundings and helping users identify and avoid obstacles.
- GuideBot [3]
  - Small robot developed at University of California, Berkeley. Uses depth sensors and cameras for navigation
- Binghamton University [4]
  - Small robotic seeing-eye dog with leash-tugging interface to reinforce learning. Only required 10 hours of training.
- AI Smart Suitcase [5]
  - Large wheels to go from indoor to outdoor spaces and a motor to overcome obstacles. LiDAR to detect obstacles and real time kinematic satellite positioning system for outdoor use. Large and robust but very durable.
- Lysa [6]
  - 2D map generation, app, vibrating handle that alerts users of obstacles and a camera for obstacle detection.
  -

Our work is different because :

- Our 4Dog assistance guide robot will be different from the previous robotic projects in that we hope to deploy a guidance robot that operates at optimal or normal speed. Previously deployed robots have a long latency period when encountering obstacles that ultimately leave the user standing still for long periods of time. By utilizing the a* algorithm to retrieve the shortest path our robot will dynamically walk along the corresponding path and dynamically detect obstacles from lidar information alone. Previous applications have also been robust and not easily transportable. We hope to utilize a sensible attachment that users can effectively hold on to while walking at a reasonable speed.This allows for flexibility and independence during use which is crucial for user feedback to improve our current goal.

Citations :

- **[1]** *Smart cane for the visually impaired*. WeWALK Smart Cane. (n.d.). https://wewalk.io/en/
- **[2]** Moreno, P., Nunes, R., Figueiredo, R., Ferreira, R., Bernardino, A., Santos-Victor, J., Beira, R., Vargas, L., Aragão, D., & Aragão, M. (1970, January 1). *Vizzy: A humanoid on Wheels for Assistive Robotics*. SpringerLink. https://link.springer.com/chapter/10.1007/978-3-319-27146-0_2
- **[3]** Ying, C. (2021, April 13). *UC Berkeley researchers create robotic guide dog for visually impaired people*. The Daily Californian. https://www.dailycal.org/2021/04/12/uc-berkeley-researchers-create-robotic-guide-dog-for-visually-impaired-people
- **[4]** Malayil, J. (2023, October 31). *Engineers create a robotic eye-seeing dog to aid the visually impaired*. Interesting Engineering. https://interestingengineering.com/innovation/engineers-create-a-robotic-eye-seeing-dog-to-aid-the-visually-impaired
- **[5]** *AI suitcase tested outdoors to guide visually impaired people: The Asahi Shimbun: Breaking News, Japan news and analysis*. The Asahi Shimbun. (2023, October 31). https://www.asahi.com/ajw/articles/15025247
- **[6]** Jones, F. (n.d.). *A robot guide dog for the blind*. Revista Pesquisa Fapesp. https://revistapesquisa.fapesp.br/en/a-robot-guide-dog-for-the-blind/

Work Schedule

First task : Develop system requirements and performance requirements
Assigned : Everyone @ 10-07-23

Second task : Develop the Use Cases for each feature
Assigned : Everyone @ 10-14-23

Third task : Develop the Use Case Diagrams corresponding to those Use Cases
Assigned : Split up to whom developed what requirement @ 10-21-23

Fourth task : Complete the Class Diagram & UML Diagram
Assigned : Everyone @ 10-28-23

Fifth task : Complete the Architectural Design Document
Assigned : Cole @ 10-28-23

Sixth task : Complete the Development Standards Document
Assigned : Cole @ 11-01-23

Eight task : Complete Data Collection and Policies form and Consent form
Assigned : Cole @ 11-16-23

Ninth task : Build and Spec the robot
Assigned : Everyone @ 01-01-24

Tenth task : Implement the hardware on the raspberry pi
Assigned : Everyone @ 01-15-24

Eleventh task : Implement the software for the hardware
Assigned : Everyone @ 02-01-24

Twelfth task : Implement the dynamic grid from UWB sensors
Assigned : Emmett @ 02-15-24

Thirteenth task : Implement the path planning algorithm
Assigned : Emmett & Cole @ 02-29-24

Fourteenth task : Testing the robots movement on dynamic grid sizes
Assigned : Everyone @ 03-15-24

Life Cycle Approach

Concept Development

- Was used to identify the specific needs and challenges of visually impaired users.
- Define the goals and functionality of the robot and the problems we wish to solve.
- Research the existing solutions and technologies to stand out and improve the design and user experience.

Requirements Analysis

- Define the technical and functional requirements of what the robot needs to do.
- Define the performance requirements such as weight, size, power of the servos, the necessary wheel tread, and power consumption requirements it must adhere to.
- Define the requirements of how the user will interact with the robot and hold on during travel.

Design

- Design the hardware for the bot to achieve path planning and obstacle detection with a rechargeable battery option.
- Develop the communication sensors for the bot to establish a dynamic gridspace in which the robot achieves path planning and obstacle avoidance.
- Design an architectural structure for the robot to achieve advanced functionality while detecting both static and dynamic obstacles, and how this information could be relayed to the user in the most practical and feasible manner.
- Design the physical appearance, and size of the robot considering walking speed and outdoor use cases.

Prototyping

- Specified the required hardware mounts and brackets to reliably assemble the robot together, considering durability for outdoor applications.
- Iteratively refine the design based on feedback from the design stage with the accuracy of the sensors during navigation.
- Design the required brackets needed to assemble the hardware and components to meet the design and sensor requirements.

Hardware Development

- Develop the final version of the robot's hardware mounts, integrating sensor mounts, servo motor brackets, UWB sensors with their own power supply, imu sensor placement, voltage regulator to refine the servo output on a durable chassis to achieve functionality requirements.

Software Development

- Develop the software that controls the robot, including navigation algorithms, and obstacle avoidance.
- Implement speech to text capability for user destination interaction.

Integration

- Integrate the software with the hardware components and ensure seamless communication between all hardware and power source.
- Conduct testing to identify and resolve any integration issues or sensor failures.
- Organize the sensors into classes that independently thread into the master class that controls the communication of the robot during navigation.
- Refine the organization of the code to increase readability and reusability while advancing the sensor data to independently communicate to the master class.

Testing

- Perform extensive testing, including both simulated and real-world scenarios of encountering obstacles, encountering walls, and navigating around different scenarios of static and dynamic moving obstacles.
- Test the robots spatial awareness of current position within the dynamic gridspace to reduce the error of detecting false achievable paths that aren't available for navigation.
- Test the robots gridspace accuracy to assess the overall accuracy of the robots final position relative to the users requested destination.

Deployment

- Monitor and update the software as needed to address emerging issues or improve functionality to increase responsiveness and awareness within the known gridspace.

Proposal Statement

Robot Functional Requirements

1. The 4Dog shall have self locomotion.
2. The 4Dog shall navigate around obstructions on flat surfaces to ensure user safety..
3. The 4Dog shall be rechargeable.
4. The 4Dog shall work in 0°F temperatures and be functional outside in variable weather.
5. The 4Dog shall be able to recognize voice commands for destinations and navigate to those locations.
6. The 4Dog shall take the shortest path from the user's start location to the specified destination.

Robot Non-Functional Requirements

7. The 4Dog robot shall be powerful enough to operate at a normal walking speed.
8. The 4Dog robot shall have an easily locatable handle in which the user can grab and hold on to while navigating to their destination.
9. The 4Dog robot shall be lightweight and small enough not to obstruct pedestrians or harm the user if the two collide in any manner.

Feature

F-1. The 4Dog shall have self locomotion.

Acceptance Criteria

GIVEN that 4Dog needs to physically guide a human

WHEN a human specifies a location they want to travel

THEN 4Dog safely guides them to that location.

User Stories

US 1-1. As a user of 4Dog I want to navigate safely from my location to my desired destination so that I can safely maneuver around.

US 1-2. As a user of 4Dog I want to be able to walk at a leisurely pace so that I don't trip over my own feet.

Feature

F-2. The 4Dog shall navigate around obstructions on flat surfaces to ensure user safety.

Acceptance Criteria

GIVEN I am a user of 4Dog and I wish to avoid walls, or other static obstructions

WHEN 4Dog is navigating along the designated path

THEN 4Dog should avoid static obstacles and stay on flat surfaces to maximize usability.

User Stories

US 2-1. As a user of 4Dog I expect to be able to navigate around static obstructions so that I don't collide into anything.

US 2-2. As a user of 4Dog I want to stop before colliding with any dynamic obstacle in my path and then to continue when the path is clear so that I don't have to walk around moving obstacles in my path.

Feature

F-3. The 4Dog shall be rechargeable.

Acceptance Criteria

GIVEN that 4Dog needs to be recharged

WHEN a human wants to have full battery

THEN the 4Dog can be recharged.

User Stories

US 3-1. As a user of 4Dog I want to be able to recharge the robot so I can recharge my system efficiently when I choose.

Feature

F-4. The 4Dog shall work in 0°F temperatures and be functional outside in variable weather.

Acceptance Criteria

GIVEN I am a user of 4Dog in an environment with low temperatures or variable weather

WHEN 4Dog is guiding me to my destination, along the desired path,

THEN 4Dog should operate normally without any noticeable degradation in performance.

User Stories

US 4-1. As a user of 4Dog I should expect that 4Dog will work in an environment with the temperature at 0°F or with light moisture to ensure that I can safely get to my destination regardless of extraneous outside weather conditions.

Feature

F-5. The 4Dog shall be able to recognize voice commands for destinations and navigate to those destinations.

Acceptance Criteria

GIVEN I am a user of 4Dog and want to easily and verbally communicate my destination

WHEN 4Dog is waiting for the users request

THEN 4Dog is able to accurately navigate to the destination request via voice command, given that the destination fits within the gridspace of the robots capabilities.

User Stories

US 5-1. As a user of 4Dog I expect to be able to give my destination verbally to the robot and to be able to accurately navigate to that destination, given that destination is within the gridspace.

Feature

F-6. The 4Dog shall take the shortest path from the user's start location to the specified destination.

Acceptance Criteria

GIVEN I am a user that has just prompted the robot to navigate me from my current location to my desired destination

WHEN 4Dog acquires the current location

THEN 4Dog will properly select the optimal path to my requested destination while navigating obstacles on that path.

User Stories

US 6-1. As a user of 4Dog I want to navigate to my desired destination by taking the shortest path while staying on accessible terrain so that I can safely arrive.

Feature

F-7. The 4Dog robot shall be powerful enough to operate at a normal walking speed.

Acceptance Criteria

GIVEN I am a user of 4Dog

WHEN the user is being lead to their requested destination

THEN 4Dog will lead the user at a comfortable walking speed, which is not too fast or too slow.

User Stories

US 7-1. As a user of 4Dog I want to comfortably walk from my current location to my specified destination so that I don’t have to run, and can safely walk outside.

US 7-2. As a user of 4Dog I want to comfortably walk from my current location to my specified destination so that I don’t have to pause in between steps, and can efficiently maneuver to that destination.

Feature

F-8. The 4Dog robot shall have an easily locatable handle in which the user can grab and hold on to while navigating to their destination.

Acceptance Criteria

GIVEN I am a visually impaired user of 4Dog

WHEN I’m being led to my destination

THEN 4Dog should have an easily accessible handle to find and hold onto.

User Stories

US 8-1. As a visually impaired user of 4Dog I want to be able to comfortably hold onto the robot's handle while navigating me to my current destination so that I arrive safely.

Feature

F-9. The 4Dog robot shall be lightweight and small enough not to obstruct pedestrians or harm the user if the two collide in any manner.

Acceptance Criteria

GIVEN I am a user of 4Dog

WHEN I want to use the robot to navigate me to my desired destination

THEN 4Dog shouldn't be wider than an average person, or weigh more than 10 pounds.

User Stories

US 9-1. As a user of 4Dog I want the robot to be lightweight enough so that if it collides with any obstacle or pedestrian that nobody will get hurt.

US 9-2. As a user of 4Dog I want to be able to walk behind the robot and have it be small enough not to obstruct other pedestrians while on the sidewalk.

Architectural Design Document (ADD) for Assistive Navigation Robot 4Dog.

1. **Introduction**

   1.1. Purpose

The purpose of this document is to provide a comprehensive architectural design for a small four-wheeled robot designed to assist visually impaired individuals in navigating safely outdoors. The robot incorporates two servo motors at the front, a LiDAR sensor at the front, a microphone, an IMU sensor, a raspberry pi b 4, a voltage regulator, and UWB sensors that send current spatial information to the robot. The software on the raspberry pi will recognize natural language to set specific destinations and respond via open ai responses in friendly and informative personalities.

   1.2. Scope

The scope of this project includes the hardware and software components necessary for the robot to safely navigate outdoor environments, or flat indoor environments, while detecting obstacles, or avoiding blockades. The robot shall interact with the user and process requests by voice and respond with a recognition personality that increases user experience.

2. **System Overview**

   2.1. Hardware Components

      2.1.1. Robot Hardware :

         2.1.1.1. Two servo motors for precise movement of the robot and enough power to navigate variable terrain.

         2.1.1.2. Lightweight And durable aluminum chassis that allows motors to function optimally with minimal battery consumption.

         2.1.1.3. IMU to track orientation and increase positional awareness.

         2.1.1.4. LiDAR sensor for obstacle detection and avoidance.

         2.1.1.5. Microcontroller for controlling robot's motors and sensors.

         2.1.1.6. Durable wheels with lightweight tread easily attached to the robots servo motor output shafts.

         2.1.1.7. AV microphone to listen for user requests

         2.1.1.8. UWB sensor grid for positional information.

      2.1.2. Robot Software

         2.1.2.1. Servo motor control developed in Python for motor and sensor management and communication between sensors.

         2.1.2.2. Navigation algorithm for obstacle avoidance and path planning in Python using the heuristic a* algorithm.

         2.1.2.3. Obstacle avoidance detection using averaged LiDAR sensor data to continuously monitor for obstacles while navigating.

   2.2. Integration of Components and Software

      2.2.1. Robot Hardware:

2.2.1.1. 3D print wheel hub mounts that allow for disassembly and repair while integrating a voltage regulator to ensure the software can accurately control the front servo drive motors.

2.2.1.2. 3D print sensor mounts and brackets to ensure accurate data transmission to the robot and allowing for optimal coordination.

2.2.1.3. Equipping each ESP32 with its own power source to power the UWB sensors where each one is mobile and allows the user to have a dynamic and scalable gridspace to navigate around in.

2.2.2. Robot Software :

2.2.2.1. Architectural design of the hardware where each sensor has its own class and corresponding methods to return sensor data for the desired functionality.

2.2.2.2. Modular design to manage each sensor with independent threads that accurately and synchronously communicate to the robot during navigation.

2.2.2.3. Architectural design for the UWB sensors to establish a gridspace by turning on each of the UWB sensors. After this initialization the gridspace can be updated with the robots current position and the established destination. This class then communicates to the master class while the robot moves to ensure obstacles are detected.

3. **System Functionality**

3.1. User Interaction

3.1.1. Microphone

3.1.1.1. Users can set destinations using natural language commands.

3.1.1.2. The application provides real-time inquiries about the user's needs.

3.1.2. Adjustable handle

3.1.2.1. Users can adjust the handle to the robot for desired heights to increase feasibility and practicality.

3.2. Navigation

3.2.1. Obstacle Avoidance

3.2.1.1. LiDAR sensor detects obstacles, with the navigation algorithm that plans alternative routes if the obstruction is static.

3.2.2. Path Planning

3.2.2.1. The navigation algorithm calculates the optimal path to the user-defined destination, while avoiding static obstructions.

4. **Communication Flow**

4.1. From sensors to Pi

4.1.1. Pi processes commands and controls the servo motors

4.1.2. IMU measure angle and provides feedback about positional data

4.1.3. Lidar data to detect any obstructing obstacles in the way of the path to the robot's motor controls.

4.1.4. UWB sensors send positional and gridspace data to the robot that is continuously updated.

4.1.5. Microphone that sends the user's requested destination to the gridspace class that gets the grid of the destination for acquisition from the master class.

5. **Deployment**

5.1. The robot will be deployed for users outdoors where visually impaired individuals require assistance, or inside for demonstrating purposes.

6. **Conclusion.**

6.1. The design of this robot is to provide a reliable and user friendly assistive solution for visually impaired individuals that want to navigate around safely. The integration of our sensors and selected hardware to provide the user with reliable, accurate and real time navigation, is to ensure we safely provide guidance from the users location to their destination. We aim to prioritize safety and user experience by handling the users desired destination locally on the bot and by controlling the navigation locally with rigorous testing before deployment. By continuous testing and receiving feedback from any user, optimizing the system is essential for functionality and better deployment.

Development Standards Document (DSD) for Assistive Navigation Robot

1. **Introduction**
   1.1. The purpose of this document is to establish development standards for the creation of a small four-wheeled robot designed to assist visually impaired individuals in navigating outdoor environments. Adhering to these standards will ensure consistency, maintainability, and the production of a high-quality, reliable system.
2. **Coding Standards**
   2.1. Language and Platform
      2.1.1. Use Python for embedded system programming on the robot for readability and accuracy of the sensors communicating through the raspberry pi.
   2.2. Code Organization
      2.2.1. Use a modular structure to enhance maintainability and control.
      2.2.2. Use Git for our code versioning control, with GitHub as the repository service.
   2.3. Naming Conventions
      2.3.1. Use consistent naming convention throughout relative to the class name and sensor inputs.
      2.3.2. Used the under_score syntax for variables and function names to prioritize attention to detail. This syntax also increases comprehension of advanced features by separating the logic in a detailed format.
      2.3.3. Use descriptive variables and method naming throughout the codebase.
   2.4. Comments
      2.4.1. Use clear comments for all functions and sections of code to increase readability, and maintainability while testing.
      2.4.2. Document the purpose of classes, functions, and major code blocks to increase the functionality of the robot during development and software integration.
   2.5. Error Handling
      2.5.1. Implement robust error handling mechanisms to handle unexpected scenarios.
      2.5.2. Implement robust error messages and checks for all the sensors in each class and master class to control the integrity of the system. This helped to decrease debugging time and to implement more advanced features overall by understanding the limitations of communication pathways.

3. **Hardware Standards**
    3.1. Servo Motors and Voltage Regulator
        3.1.1. To ensure reliability at low temperatures we selected 2000 series 5-turn dual mode servo motors from goBILDA.
        3.1.2. Implemented proper calibration procedures outlined by maestro servo motor controller center to ensure reliability.
        3.1.3. Properly configured the continuous operation mode to ensure our servo motors didn't engage the potentiometer during rotation and stop unwarranted.
        3.1.4. Implemented proper voltage regulation to the motors to ensure accurate power supplied and motor performance.
    3.2. LiDAR Sensor
        3.2.1. Properly managed the placement of the sensor to ensure integrity of the sensor's temperature and range of distance to accurately detect an obstacle that obstructs the robots path.
    3.3. IMU sensor
        3.3.1. Properly followed mounting requirements to accurately acquire the required angle data to accurately track the robots position during navigation.
    3.4. ESP32 UWB Sensors
        3.4.1. Properly calibrated the UWB sensors to be fast and accurate to adequately and dynamically communicate positional information to the robots tag during navigation. Consistent data communication protocol to create an accurate dynamic gridspace to realistically implement real world scenarios.
4. **Software Standards**
    4.1. Control class
        4.1.1. Maestro control software was used to establish the correct inputs to the servo motors to ensure the motors were optimized and not out of its designed range of outputs.
        4.1.2. Maestro control software was used to set the input to the correct values on the motors and to allow us to accurately initialize each motor which allows them to operate simultaneously.
    4.2. Tag class
        4.2.1. Receive incoming UWB data from serial bus. We return those distances for use outside of the class.
        4.2.2. Has the ability to return the data in several useful ways, including averaged values and anchor positions if necessary.

- 4.3. Gridspace class
  - 4.3.1. Utilize the incoming data from our UWB sensors to establish a grid that can be dynamically adjusted as 4Dog navigates through the grid.
  - 4.3.2. Implemented the appropriate a* algorithm to plan the shortest path of the gridspace inside the gridspace class to effectively manage the previous and current locations of the robot such that the robot could accurately re-path if obstructed for long periods of time and still navigate to the desired destination.

5. **Testing Standards**
   - 5.1. Unit Testing
     - 5.1.1. Conducted thorough testing inside each class before integration and after.
     - 5.1.2. Continuous testing and error handling were done to double check the robots location relative to the UWB tag data.
     - 5.1.3. Tested each sensor data with tape measures or rulers to ensure the sensor's integrity during deployment once mounted and before.
   - 5.2. Integration Testing
     - 5.2.1. Tested the integration of hardware and software components to ensure seamless operation. We continuously monitored the output of the gridspace during navigation to ensure proper functionality.
     - 5.2.2. Simulated real-world scenarios for comprehensive testing by having roommates walk in front of the robot and stop to refine the control.
   - 5.3. User Acceptance Testing
     - 5.3.1. Simulated a visually impaired user navigating around a wall of obstacles and the practicality of our handle during navigation to test the control and accuracy of the robots functionality.

6. **Documentation Standards**
   - 6.1. Code Documentation
     - 6.1.1. We generated comprehensive documentation for all code, including, inline comments and README files.
     - 6.1.2. In the code we provide clear comments describing the classes, methods, and variables used in all programs.
     - 6.1.3. In the Code we provide instructions on how the dynamic gridspace works and how the UWB sensor initialization works to ensure transparency and readability.

6.2. Hardware Documentation

6.2.1. During development we utilized the I2C communication protocols and respective pinouts on the raspberry pi to establish a connection with the LiDAR sensor and the IMU sensor. Originally intended to implement two LiDAR sensors but were unable to separate them on the same I2C bus, or establish a separate bus due to hardware limitations. Instead we focused on just a single middle mounted LiDAR sensor to accurately detect obstacles 50 cm from the robots current location. This half meter implementation was relative to the robots speed and the grid cells to increase the accuracy of the robots current location during locomotion.

6.2.2. During the wiring of the imu sensor we utilized the same I2C bus already utilized by the LiDAR sensor for simplicity.

6.2.3. During the wiring of the servo motors to the robots power supply we added a voltage regulator in line with the motor power switch to increase the accuracy of the motors input and to ensure the motors didn’t back power the Maestro servo controller and cause internal issues.

6.2.4. Wiring of the ESP32 and the UWB sensors incorporated a single switch with a power supply and a ground that allowed for the grid to be deployable and reduced battery maintenance.

7. **Security Standards**

7.1. Data Security

7.1.1. By handling everything locally on the raspberry pi we limited the amount of security vulnerabilities. We don’t locally save any user input other than the requested destination however we password protect the pi.

8. **Conclusion**

8.1. These development standards have been established to ensure a consistent and detailed approach to the development of our visually assistive robot. Adhering to these standards provides documentation, and guidelines that facilitate the project's goals and development. We aim to provide collaboration, maintainability, and increase success of this project by providing detailed documentation. Updates to these standards will be made during the production and development for future and all use of this robot.

Data Collection and Usage Policies

Overview

This document outlines the data collection and usage policies for the 4Dog assistive navigation robot. The goal is to ensure transparency, privacy, and responsible use of data collected during the project.

Data Collection

1. **Purpose:**
   The collected data is used to improve the performance and functionality of our small robot in assisting visually impaired users with real-time navigation. This includes navigation algorithms, enhancing obstacle avoidance mechanisms, and improving the overall user experience.
2. **Types of Data Collected:**
   The following types of data may be collected during the project:
   1. Location Data to facilitate navigation and provide real-time assistance.
   2. Usage Data on how the robot is used and to enhance its features and functionality.
   3. Error and Diagnostic Data related to system errors for debugging and improvement purposes.
3. **Data Collection Methods:**
   Data will be collected through sensors embedded in the robot.
4. **Data Security:**
   All collected data will be stored securely, and access will be restricted to only authorized project personnel. Measures will be implemented to prevent alteration of the data.

Data Usage

1. **Purpose:**
   Collected data will be used exclusively for research and development purposes related to small robot navigation assistance projects. It will not be used for any other commercial or non-project related activities.
2. **Research and Development:**
   Data will be used to enhance the robot's navigation capabilities, optimize algorithms, and improve the overall functionality based on user interactions and experiences.
3. **Anonymity:**
   Personal identifiers will be removed or anonymized whenever possible to protect the privacy of participants.
4. **Third-Party Access:**
   Collected data will not be shared with third pirates for commercial purposes.

Consent Document for Data Storage

Project Title : 4Dog Robot Navigation Assistance Project

Researchers : Cole Smith, Emmett Osborne, Austen Harrell

Date : TBD

Introduction
I, the undersigned participant, hereby provide consent for the collection, storage, and usage of data as described in the Data Collection and Usage Policies document for the 4Dog Robot Navigation Assistance Project.

Data Collection
I acknowledge that the collected data will be used exclusively for research and development purposes to enhance the capabilities and functionality of the small robot.

Anonymity
I am aware that personal identifiers will be removed or anonymized to protect my privacy.

Data Security
I understand that all collected data will be stored securely, and access will be restricted to authorized project personnel only.

Data Sharing
I acknowledge that collected data will not be shared with third parties for commercial purposes and will be used solely for the Robot Navigation Assistance Project.

Withdrawal of Consent
I understand that I have the right to withdraw my consent at any time, and my participation in the project will not be affected if I choose to do so.

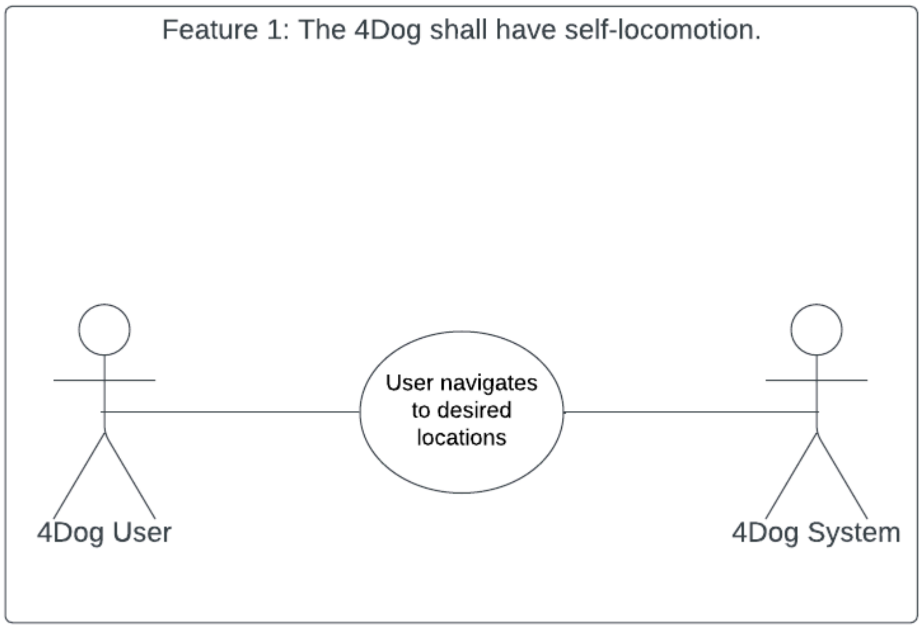
Contact Information
If I have questions or concerns regarding the data collection and usage policies, I can contact Cole Smith at [406 600 4680]

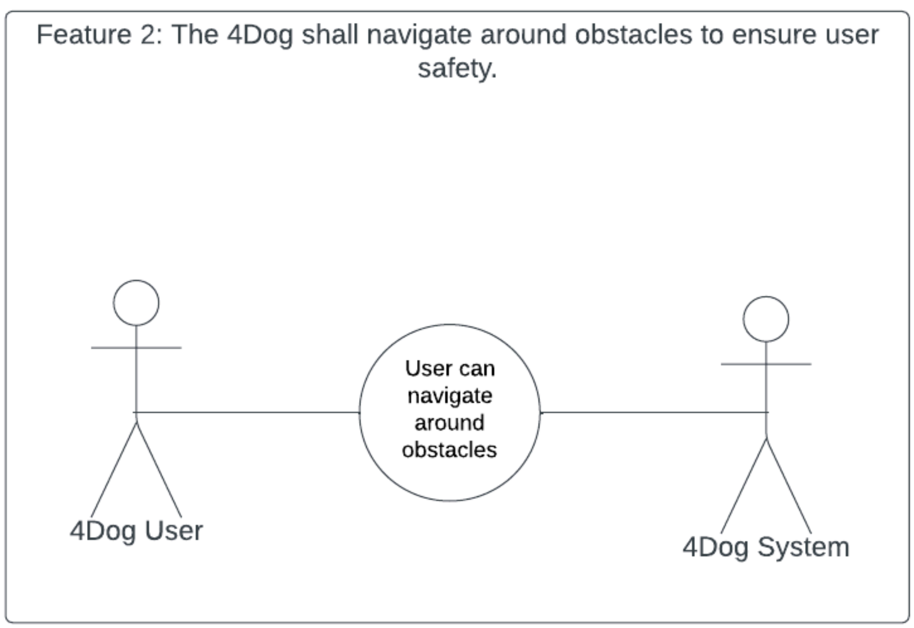
Participants Name (printed) : ____

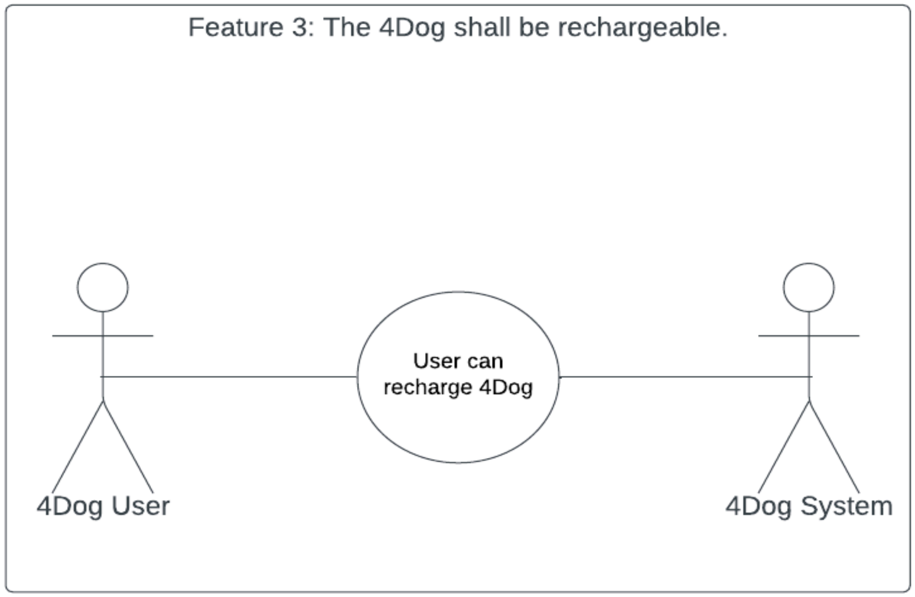
Participants Signature : ________________________________**Date :** _______________

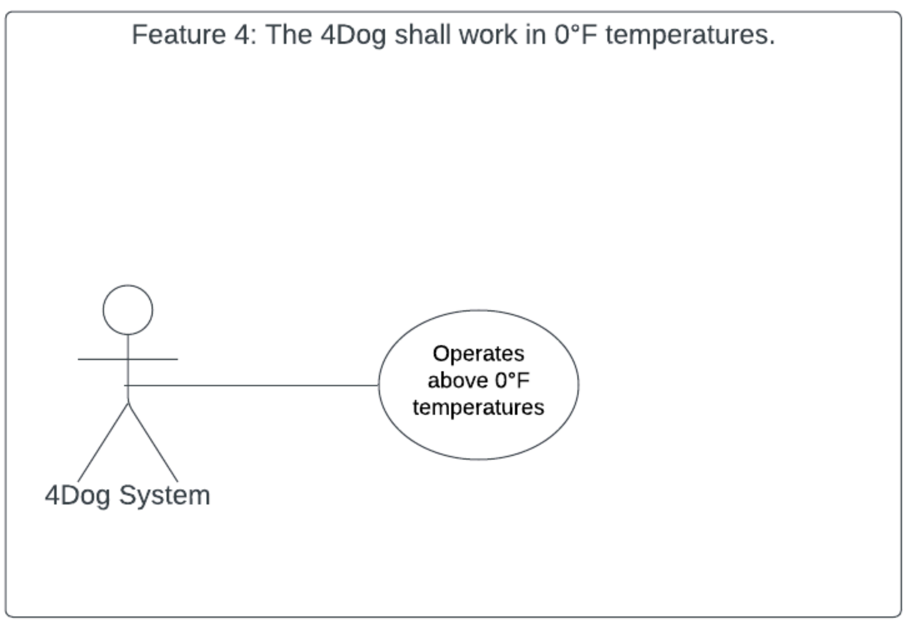
Methodology

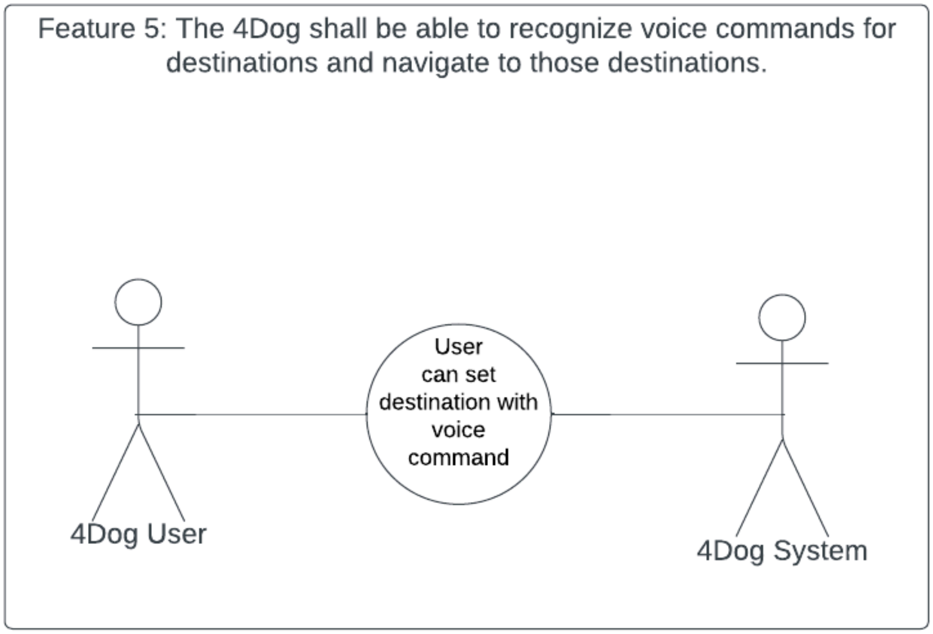
Use Case Diagrams:

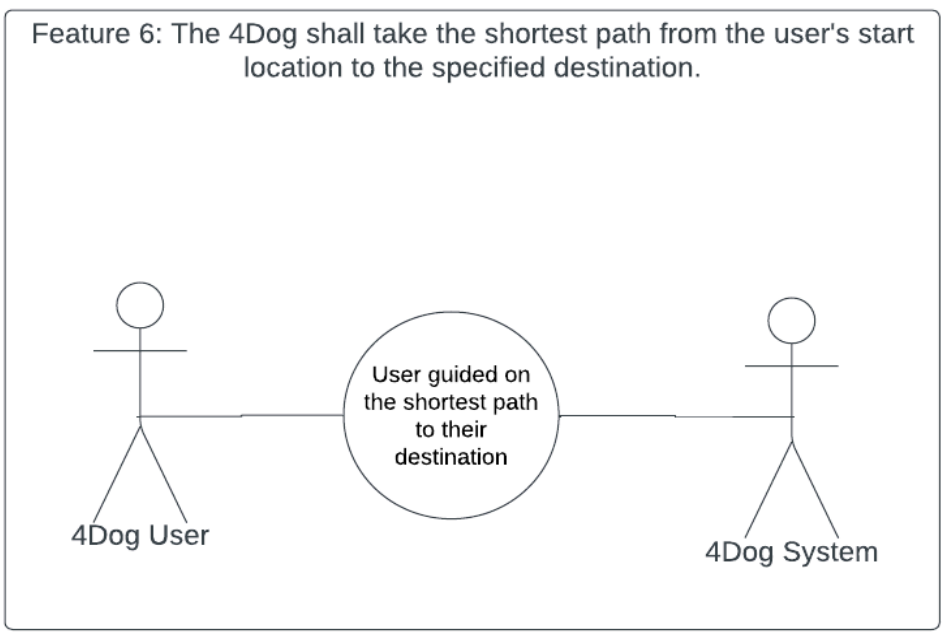


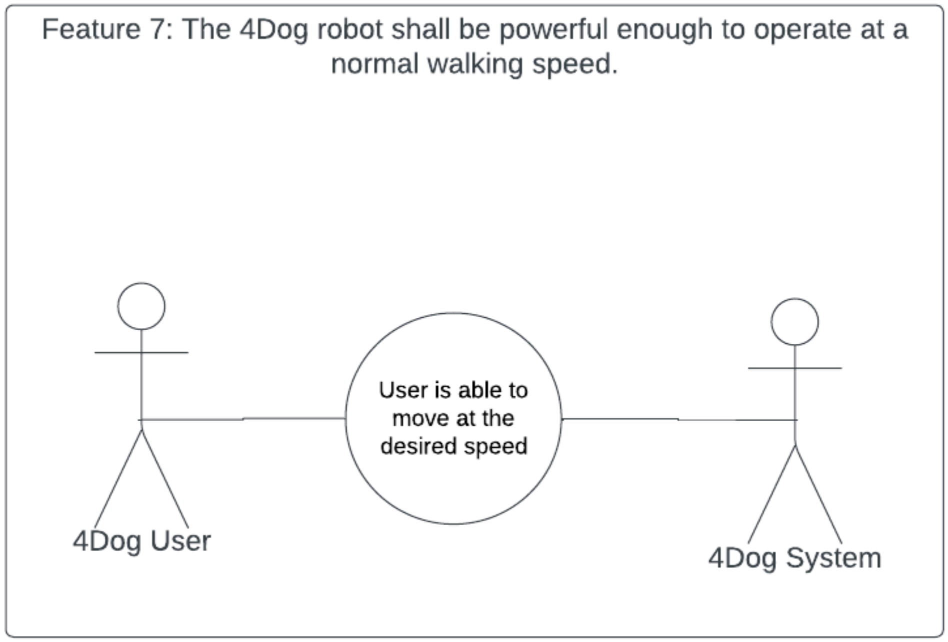


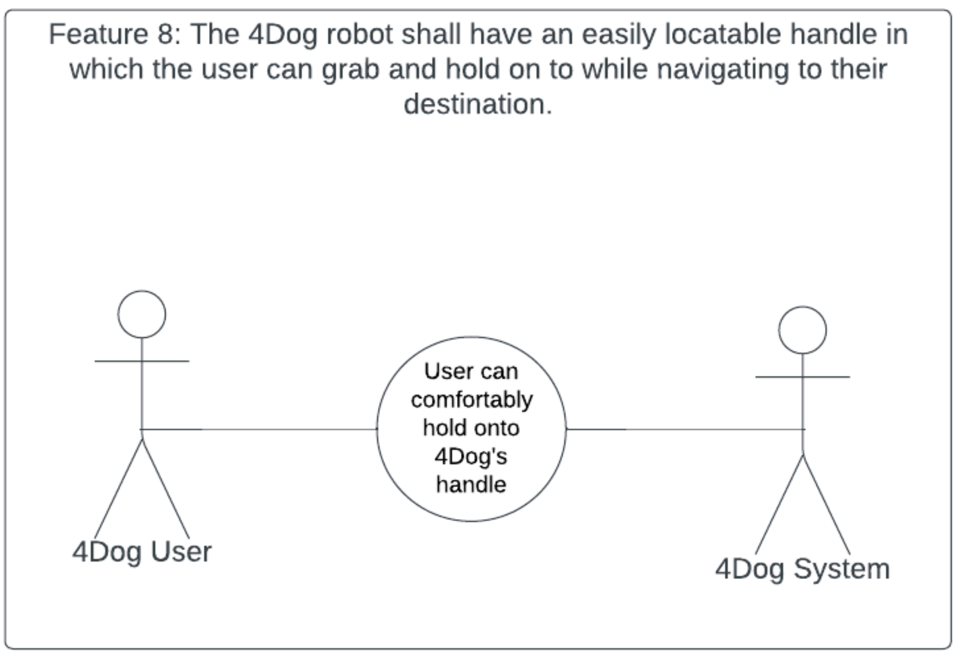

Feature 3: The 4Dog shall be rechargeable.
User can recharge 4Dog
4Dog User
4Dog System

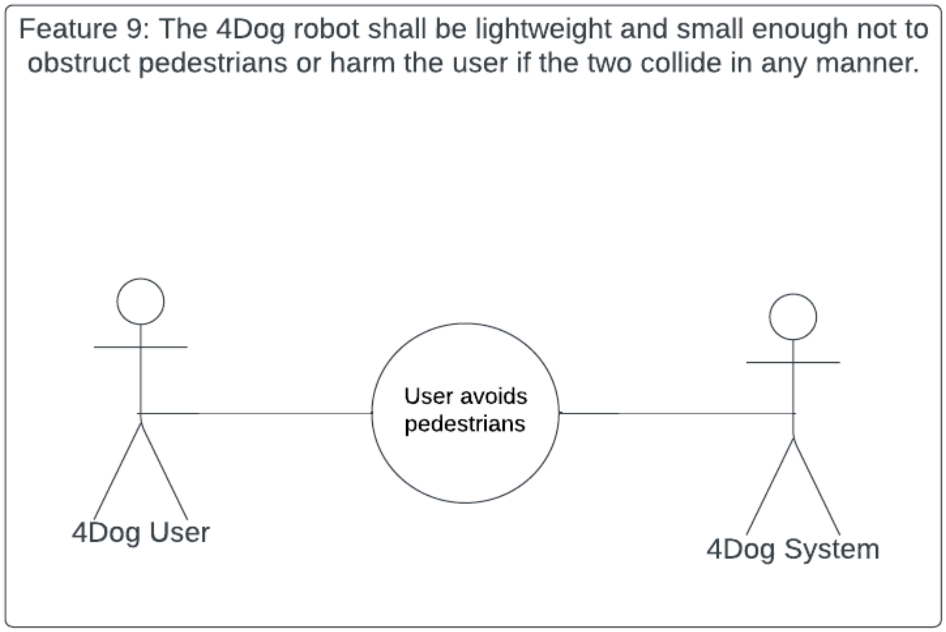

Feature 4: The 4Dog shall work in 0°F temperatures.
Operates above 0°F temperatures
4Dog System


Feature 5: The 4Dog shall be able to recognize voice commands for destinations and navigate to those destinations.
User can set destination with voice command
4Dog User
4Dog System

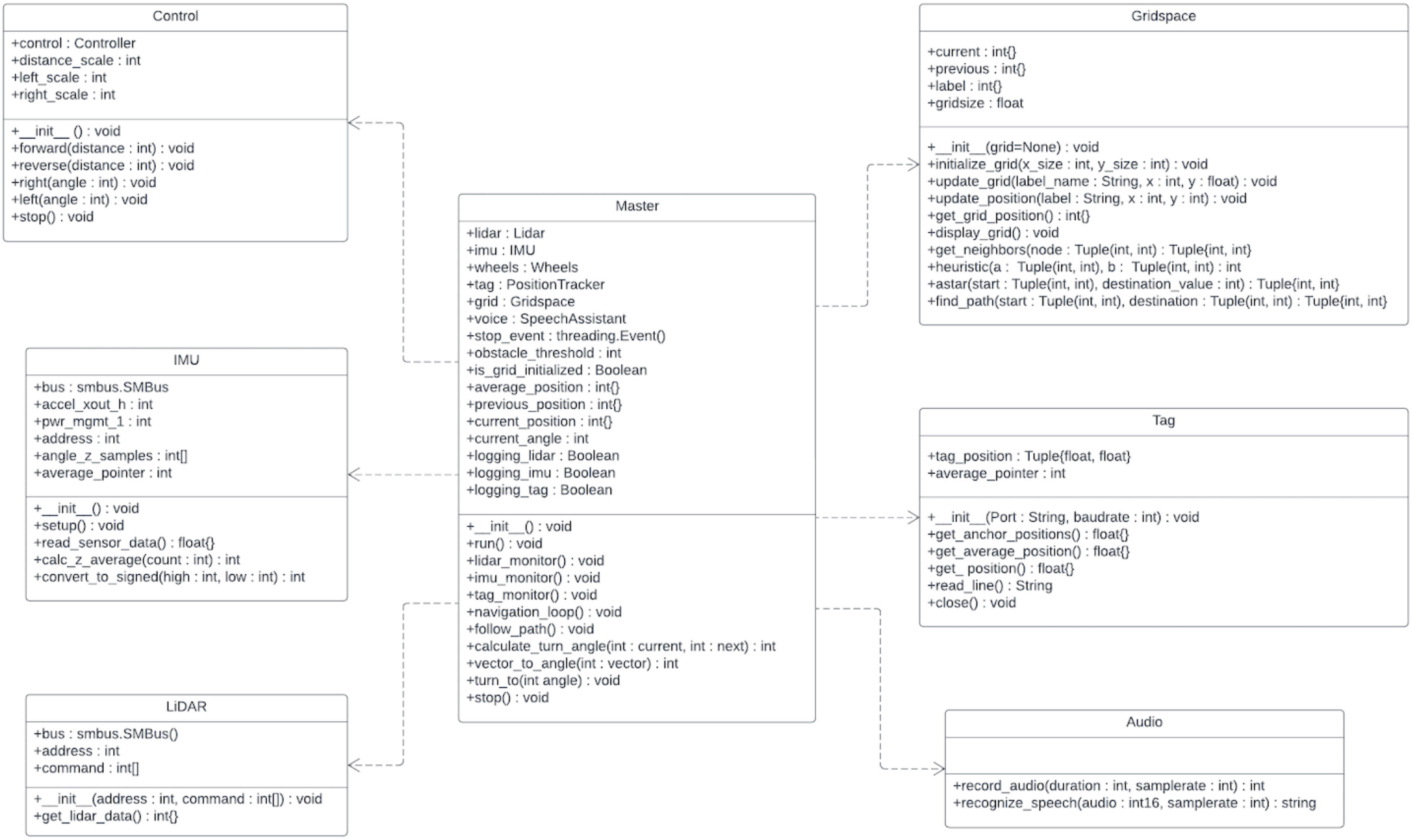

Feature 6: The 4Dog shall take the shortest path from the user's start location to the specified destination.
User guided on the shortest path to their destination
4Dog User
4Dog System


Feature 7: The 4Dog robot shall be powerful enough to operate at a normal walking speed.
User is able to move at the desired speed
4Dog User
4Dog System


Feature 8: The 4Dog robot shall have an easily locatable handle in which the user can grab and hold on to while navigating to their destination.
User can comfortably hold onto 4Dog's handle
4Dog User
4Dog System


Feature 9: The 4Dog robot shall be lightweight and small enough not to obstruct pedestrians or harm the user if the two collide in any manner.
User avoids pedestrians
4Dog User
4Dog System

Class Diagram:



Design Tradeoffs

We discussed the Cost and precision of high precision motors and encoders and which accurate control will give the appropriate feedback for our software system to handle. More expensive encoders and motors will provide more accurate positioning and control, but we are balancing the project cost and implementation of path planning algorithms to correct for any precision issues. We need a reliable, lightweight battery system to provide enough charge for the torque of our motor against the weight of our robot chassis and components. We also need enough batteries for extended use. We have prioritized the balance between a lightweight chassis and powerful enough batteries in order to increase user convenience and overall usability of the robot. When it comes to Bluetooth, we need to make sure our implementation is secure and that the range is appropriate for flexibility but that it doesn't drain the battery life. We decided how far our range should be to allow for feasibility and practicality to guide the user to and from locations. Next, we discussed the obstacle avoidance algorithm and how complex it needs to be to reduce latency during decision time, and path planning. Lastly we chose a reliable LIDAR sensor that is accurate enough to detect obstacles in variable weather and small enough to reduce power consumption. All together our focus was on the practicality and feasibility of our robot to be durable enough outside but capable of navigation paths for long durations of time reliably.

Expected Results & Observations

Robot Functional Results:

1. Self-Locomotion: The 4Dog will demonstrate the ability to move autonomously in a controlled environment.
    a. We were able to achieve self-locomotion by creating a gridspace and orienting the robot inside that gridspace. We utilized the modulus operator to calculate vector angles relative to our gridspace cardinal directions that controlled the turning orientation of the robot and allowed it to move freely in our controlled gridspace.
2. Obstacle Navigation: The 4Dog will successfully navigate around obstacles on flat surfaces in various test scenarios, ensuring no risk to user safety.
    a. We were able to achieve this result by implementing a LiDAR sensor on its own thread that always checked for obstacles within a 50 cm threshold. If this was ever the case we stopped the motors. We were also able to tell the robot about obstacles within the grid space and have the path planning algorithm give the shortest path around those obstacles. Once we had this established the robot was then autonomously controlled and continuously navigated around those obstacles to the users destination.
3. Rechargeable System: The 4Dog will show efficient battery usage and can be fully recharged, with documented recharge times and battery life under different usage conditions.
    a. We could have implemented this feature on the robot simply by sticking a longer usb cord, male to female to the battery pack out the bottom of the robot, and then having the user plug into the female end of the usb, however the battery pack and the voltage regulator that we supplied during development of the hardware lasts for extended periods of time. This expected result was not observed but is very practical for longer periods of use and could be easily implemented to suit the needs of the user
4. Visibility Lighting: The 4Dog will be equipped with lighting that effectively makes it visible to people in its vicinity, with tests confirming visibility under different lighting conditions.
    a. We did not implement any lights on the robot and opted to focus on the voice command to select for destinations within the gridspace since this provided functionality to the robot and not just performances. We did in fact have an LED strip of lights that was going to flash different colors during operation however the practicality of this feature took away from the optimality of the batter;y performance and added unnecessary cost to the development.
5. Smartphone Connectivity: The 4Dog will consistently connect to a user's smartphone via an app, with successful demonstrations of stable connectivity under various conditions.
    a. We were unable to see this expected result. When developing the app on a smartphone we investigated the .Net Maui platform for a windows emulator or an

Android emulator. However during integration of this software to control the hardware on the robot we realized that firewall restrictions on campus would not allow us to communicate to a Python Apache Flask server on the backend. Thus we decided to implement a microphone that would locally handle the users requests instead of adding additional costs to the robots projected timeline.

6. Low-Temperature Operation: The 4Dog will function effectively in environments with temperatures as low as 0°F, with performance metrics documented under these conditions.
    a. We were not able to see this result, but are fairly confident that the thermal protection of the robots software would adhere to this functional requirement. Thus we implemented a waterproof design and concept to protect the robot during variable weather. We deployed our robot outside in 40°F degree weather and found that our robot operated fairly consistent to the inside application.

App Functional Results:

1. Wireless Connectivity: The app will reliably connect wirelessly to the 4Dog robot, with tests showing stable connections over a range of distances.
    a. We did not see this expected result. We were unable to get the .Net Maui app to interface with the user and communicate to the raspberry pi because of a lack of time management and firewall restrictions that MSU has in place for communicating with an Apache Flask Server.
2. Non-visual Interface: The app will provide a fully functional nonvisual user interface, accessible and user-friendly for all target users.
    a. Without the app we were unable to implement the speech recognition software on the app however we did implement this on the robot locally. This actually works better since visually impaired users of the robot shouldn’t be required to hold down a large button to send their destination to the robot. We did implement both the speech recognition and text-to-speech through Chat GPT open ai calls and were successful however when playing text-to-speech response to the user via a bluetooth speaker the degradation in the Audio quality was poor enough that static crackling just wasn't of the appropriate quality to continue. We tested other speech recognition modules and also found the quality of the output to the speaker to be terrible and filled with static. Thus we did not complete a nonvisual user interface that
3. Battery and Failure Notifications: The app will accurately notify users of low battery and any failures in the 4Dog, with tests showing timely and clear notifications.
    a. We were unable to achieve this expected result but unsurprisingly. This functional feature just didn’t seem very important since the battery life of our battery pack exceeds 2 hours of running time. We instead focused on the accuracy of our

robots position within the grid which wasn't as straightforward as anticipated and thus didn't have the needed time to add this user feature.

4. Real-Time Navigation and Tracking: The app will effectively track the user's geographic position and provide real-time navigation assistance, demonstrating accuracy in various locations.
    a. Without the app we did not see this expected result either. We chose not to use phone GPS location tracking and optimal positional accuracy because we were uncomfortable with communication protocols from apps to raspberry pis on a restricted network. After one of our group members with negligible experience with apps left the project, our focus became the implementation of the robot and less focused on the user's experience. While this feature would have improved the overall functionality of the robot, we were unable to effectively grasp the creation and communication of an app to get positional data sent to the robot to use for route navigation.

App Non-Functional Results:

1. Data Security: The app will meet or exceed industry standards for data security, ensuring the user's personal data is protected through robust security measures. This will be verified through security testing and compliance checks.
    a. This expected result was not observed. We did not find the time or implement the correct utilization of an app that could communicate to our robot on the backend. Despite our efforts to learn .Net Maui we were unsuccessful in creating this app. We alternatively tried to create a user interface just with HTML and CSS that was its own web server however this would not have met this security requirement and ultimately was disregarded. I found the challenges with protecting the users data to require far more time than we left available for this task and thus we fell short of implementation of this. In future work on this robot we hoped to establish a sound user interface and a circuit board that integrated all the sensors into the raspberry pi. This way we could industrialize the final product and not have to worry about the integrity of the sensors and wiring as much which would free up cost and time for implementing security checks on the user interface.

References

[1] Georgetown University. (2019, February 13). *Visual impairments*. Health Policy Institute. https://hpi.georgetown.edu/visual/#

[2] *Guide dogs 101*. Guiding Eyes for the Blind. (2023, August 31). https://www.guidingeyes.org/guide-dogs-101/

[3] *Guide dogs are not for all visually impaired people*. Outlook Enrichment. (2021, September 8). https://www.outlooken.org/resources/guide-dogs-are-not-for-all-visually-impaired-people/

[4] J. Pechiar, "Architecture and design considerations for an autonomous mobile robot," 2021 IEEE URUCON, Montevideo, Uruguay, 2021, pp. 343-346, doi: 10.1109/URUCON53396.2021.9647380.

[5] Lloyd, J., Budge, C., La Grow, S., & Stafford, K. (2016). An Investigation of the Complexities of Successful and Unsuccessful Guide Dog Matching and Partnerships. *Frontiers in veterinary science*, *3*, 114. https://doi.org/10.3389/fvets.2016.00114

[6] Menuge, F., Marcet-Rius, M., Jochem, M., François, O., Assali, C., Chabaud, C., Teruel, E., Guillemot, J., & Pageat, P. (2021). Early Evaluation of Fearfulness in Future Guide Dogs for Blind People. *Animals : an open access journal from MDPI*, *11*(2), 412. https://doi.org/10.3390/ani11020412

[7] Yadav, Pragya. (2023, October 12). *How to Build an ADA-compliant APP.* testRigor. How to Build an ADA-compliant App - testRigor AI-Based Automated Testing Tool

Appendix:

Original Proposal Statement

<u>Robot Functional Requirements</u>

1. The 4Dog shall have self locomotion.
2. The 4Dog shall navigate around obstacles on flat surfaces to ensure user safety.
3. The 4Dog shall be rechargeable.
4. The 4Dog shall include lighting for visibility to nearby people.
    a. This requirement was altered to only incorporate an LED strip around the robot in which nearby people would be able to detect the robot more easily. We initially were hoping to have a spot light that was dynamic for nighttime and powerful enough to see from across the room. However I think our updated LED strip will add light indications to the robot during different modes to add feasible and practical applications that our original idea did not incorporate.
5. The 4Dog shall connect to the user's smartphone via an app.
    a. This requirement was ultimately disbarred from the project for a number of reasons. The first being we realized that the robot could handle interpreting the desired location of the user without the app and a simple microphone and speaker that doesn't involve any security risks. We opted for this solution to help the user interact with the robot without having their phone present, which should increase the user's experience. The second reason we opted out of using an app, is we intended to program an android app in .Net Maui with the option to cross platform to ios devices if we required the specific license, however this added unnecessary attention away from the functional requirements that we were adhering to. We chose to focus on testing, and the integrity of our sensors while navigating a dynamic grid space rather than controlling the user's interaction with the app that ultimately just started the robots path and that's it. Our lack of requirements for the connection between the robot and the app in a secure and practical manner ultimately was our demise for this functional requirement.
6. The 4Dog shall work in 0°F temperatures

<u>App Functional Requirements</u>

7. The app shall wirelessly connect to 4Dog.
    a. This requirement was again disbarred from the project. The complexity of learning a new language while focusing our attention away from the functional requirements of the robot seemed to require more time than we planned for. It would have added a level of dimensionality to the users experience however one of our group members delayed production due to unfortunate circumstances and we decided it was best to proceed without an app to listen to the user. One large

issue that also influenced this decision was that MSU has firewall protection against using Apache Flask servers that listen on the backend for data being sent from apps. In order to work around this constraint, we found that we could establish a bluetooth connection from the users phone to the raspberry pi however we would need a hotspot in order to run the backend program which defeats the purpose of having a hands off interactive interface if it requires a pre-configured hotspot to be attached to the robot during use.

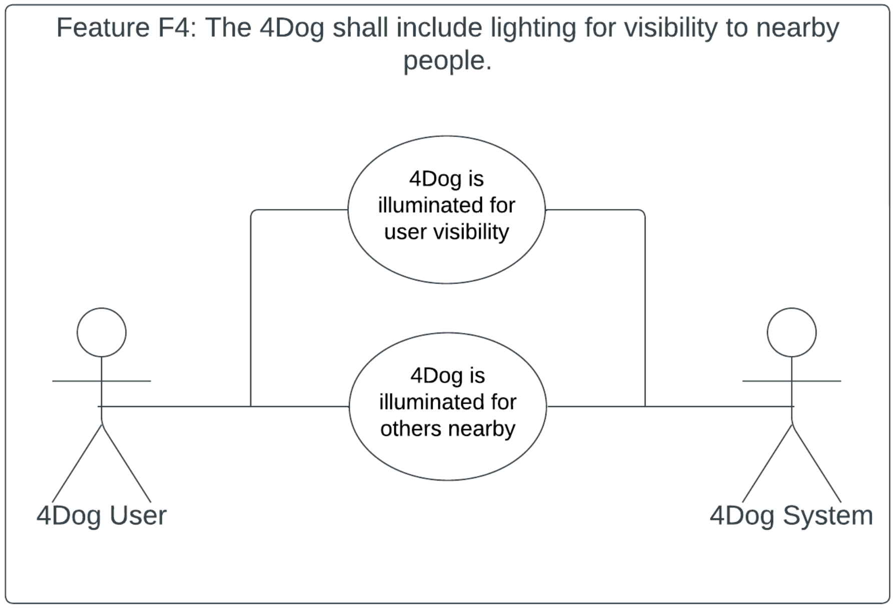
8. The app shall include a nonvisual user interface.
    a. This requirement is not exactly correct in which we don't provide an interface for the user outside of the microphone and speaker on the bot. While the raspberry pi can listen and respond to the users request with multiple personalities, it does not have an interactive interface as we had intended. We struggled with the debugging and communication protocol of .Net Maui to our raspberry pi and further found issues between the communication protocol and using MSU's wireless network. In order to implement this feature it was going to cause overages and add cost that would have delayed this current sprint and the sprints to follow.
9. The app shall notify the user if the battery is low.
    a. This requirement was not implemented since we only had one spare voltage regulator laying around. We decided to skip the installation of a voltage regulator to our battery in order to ensure reliability of our robot since we feared adding a voltage regulator to an exposed battery pack wasn't safe for long term user use.
10. The app shall notify the user of 4Dog failure.
    a. This requirement was not implemented since the app was never created, and mostly this requirement was not properly thought through. We intended to notify the user about the robot's failure state through the microphone on the phone, however it makes more sense for the user to have a free hand while holding onto the bot and to be notified about the robot's status via the bluetooth speaker, through a text to speech engine. This alleviates any safety concerns that were an oversight during the construction of this requirement.
11. The app shall track the user's geographic position in order to provide real time navigation.
    a. We were unable to implement and design a practical solution to effectively track the users geolocation in order to scale the robots practicality, and usability. We instead opted to use a dynamic grid space with uwb sensor data that can extend to ranges up to fifty square yards. With this dynamic gridspace we can track the position of the user and plan the navigation of the robot without having to process geo-satellite data and slow our navigation algorithm down.

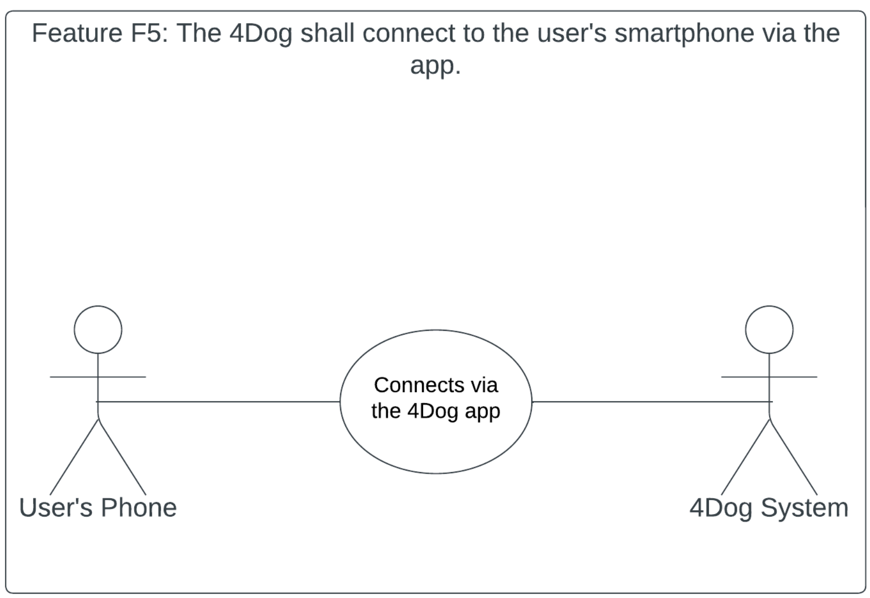
<u>App Non-Functional Requirements</u>

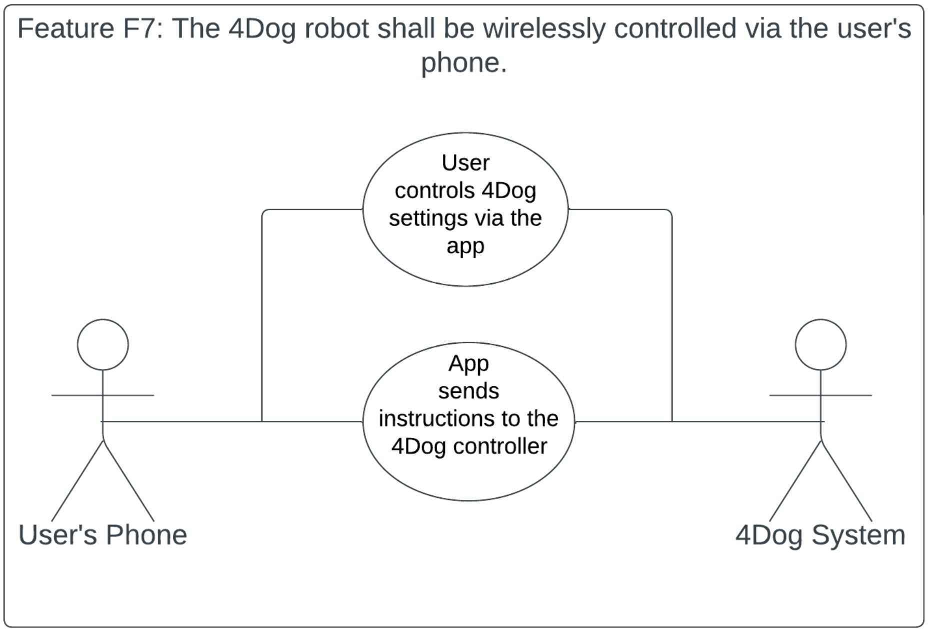
12. The app shall be secure and protect the user's personal data

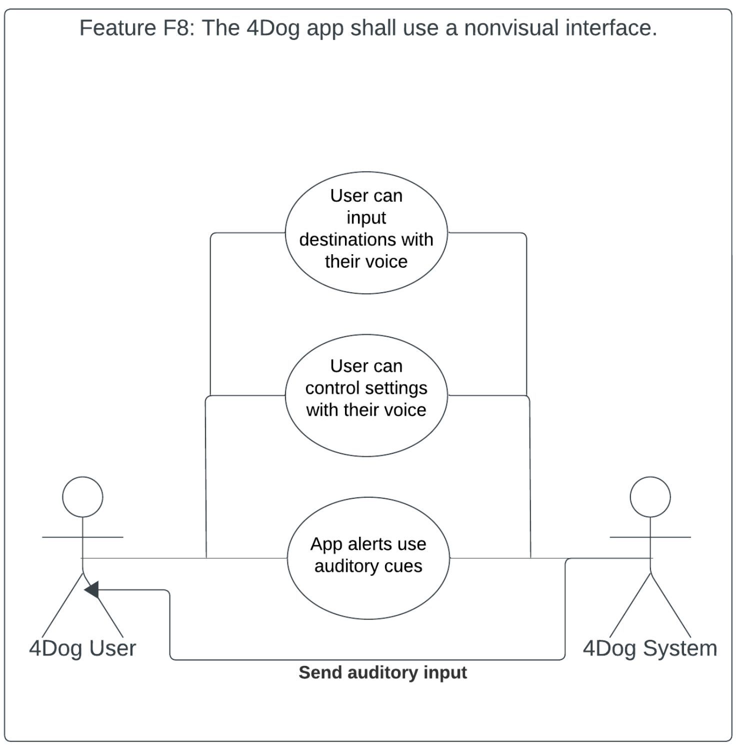
a. This would have been very challenging since the app would have required a bluetooth connection to avoid network latency issues when sending the data back and forth between the bot and the user's phone. Although we would have had very little packet loss sending the data this way, configuring a connection through bluetooth would have required a secure encryption of the users location and added more cost to this requirement than anticipated.

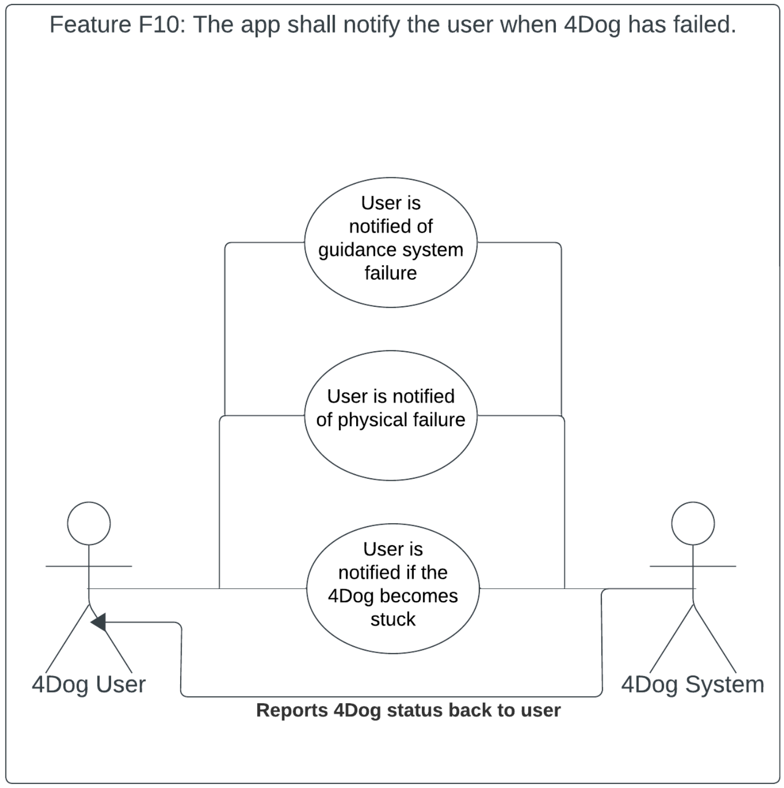
Original Use Cases

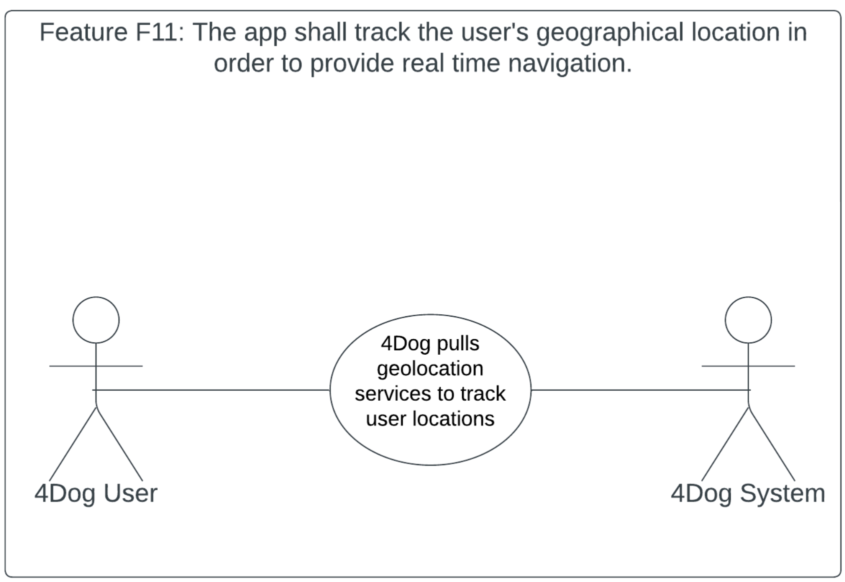


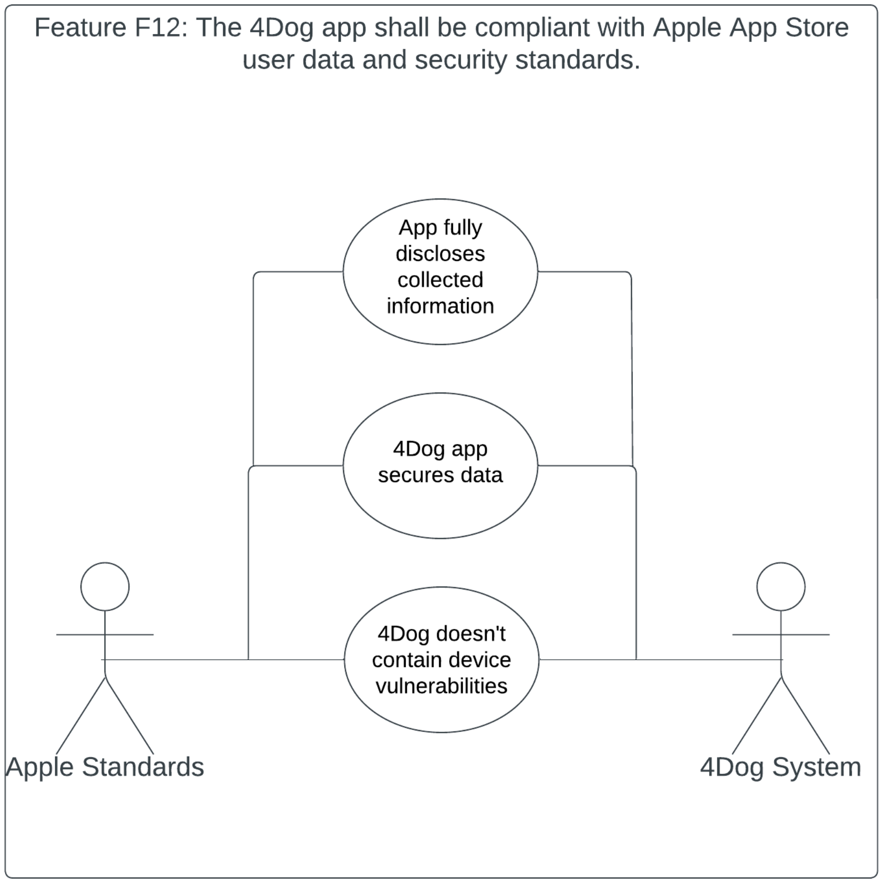



Feature F7: The 4Dog robot shall be wirelessly controlled via the user's phone.
User controls 4Dog settings via the app
App sends instructions to the 4Dog controller
User's Phone
4Dog System


Feature F8: The 4Dog app shall use a nonvisual interface.
User can input destinations with their voice
User can control settings with their voice
App alerts use auditory cues
4Dog User
4Dog System
Send auditory input


Feature F10: The app shall notify the user when 4Dog has failed.
User is notified of guidance system failure
User is notified of physical failure
User is notified if the 4Dog becomes stuck
4Dog User
4Dog System
Reports 4Dog status back to user


Feature F11: The app shall track the user's geographical location in order to provide real time navigation.
4Dog pulls geolocation services to track user locations
4Dog User
4Dog System


Feature F12: The 4Dog app shall be compliant with Apple App Store user data and security standards.
App fully discloses collected information
4Dog app secures data
4Dog doesn't contain device vulnerabilities
Apple Standards
4Dog System