SMARTFEEDR

Team Members: Leon Feng, Kristina Fout, Da Eun Lee, Sriharsha Maddala,

Mallika Gupte, Ginna Khang

University of California, Santa Cruz

Baskin School of Engineering

Department of Computer Science & Engineering

June 13, 2024

Professor David Harrison

CSE 123 Team 4

Executive Summary

Project Focus:

The project's focus is to develop an advanced automatic pet feeder for working pet owners, addressing the need for remote pet care management during extended absences.

Target Audience:

The target audience is working professionals or everyday pet owners with dogs or cats facing prolonged periods away from home, seeking a convenient and reliable pet care solution.

Design Objectives:

- User-Friendly Web App: Create an intuitive web app accessible from any location, ensuring a seamless user experience. Functionality:
    - Enable the web app to read and display food/water levels, allowing users to monitor their pet's needs remotely.
    - Control Features:
        - Incorporate controls for turning the camera, speakers, and chimes on/off, providing users with flexible options for interacting with their pets.
- Pet Feeder Mechanism: Dispenses the correct amount of food and water into the pet bowl as set by the user.
    - Functionality:
        - Accurate dispensing of the correct amount of food and water into the pet bowl.
        - Implement a system to consistently track and update the levels of food and water in the containers.
        - Integrate a filtration system to ensure clean drinking water for pets throughout.

Ethics Statement

Recognition of Need: We recognize the importance of ensuring the well-being of pets, including their proper nutrition and feeding routines.

Ethical Problem Definition: Our goal is to design an automatic pet feeder that enhances convenience for pet owners while prioritizing the health and safety of their pets.

Alternative methods to achieve this goal:

1. Owners travel from home to work to feed their pets and spend time with them:
    a. Benefits: Establish a close bond and dedicate extra time to pet companionship.
    b. Costs: Less convenient, time efficient, and may interfere with the owner's other duties like their work.

2. Hiring a pet sitter to care for the pet:
    a. Benefits: The animal gets the opportunity for human companionship through the services of the hired pet sitter.
    b. Costs: The owner spends more money instead of a one-time purchase or fee for the automatic pet feeder.

3. Bring their pets to work with them:
    a. Benefits: The owner can take care of their pet while doing work without much travel.
    b. Costs: The company that the owner works at might not allow pets at the office.

Table of Contents

1 Introduction

1.1 Need Statement

For working pet owners, especially those returning after long hours or days away, there's a crucial need for an automatic pet feeder for remote pet care management. Our solution aims to provide convenient support for ensuring pets' well-being during extended periods of absence.

Our client targets:

- People in the workforce who have dogs or cats.
- People who come back home from a day of work to a few days of work.
- People who have to go to work have to leave their pets in the house for hours at a time, they need to care for them from a long distance and our pet feeder helps manage that.

1.2 Goal Statement

Our goal is to create an advanced automatic pet feeder specifically designed for working professionals with dogs or cats. By offering a convenient solution, we aim to simplify pet care and enable pet owners to effortlessly maintain a healthy work-life balance.

1.3 Personas

- Grace, who lives alone, has a cat who only eats dry food. To get to work on time, she has to leave the house by 7 am and commute for an hour. Sometimes, she has to do overtime. She is very productive and open-minded. She loves her cat and wishes there was an easier way for her to check in on her cat more. With a clear understanding of her cat's preferences, she seeks a solution that enhances her ability to care for her pet amid her demanding routine.

- Bob, a 43-year-old man, recently inherited his late mother's dog. Although he hasn't had much interest in animals, he feels a sense of obligation to care for his mom's pet. With limited experience in pet rearing, aside from raising a goldfish, he desires a solution that won't disrupt his established routine. As a business owner who frequently travels short distances for work and enjoys going to bars after work, Bob is searching for a pet care solution that aligns seamlessly with his existing lifestyle.

- Chase, a 19-year-old full-time college student working at a restaurant, is naturally anxious and tends to overthink. Concerned about the well-being of his first pet as an adult, a dog, he worries about providing sufficient care while managing school and work commitments. Chase is actively seeking a solution to interact with and check in on his dog, ensuring that the pet receives adequate attention and care during his busy schedule.

- Serena is a 30-year-old working woman who owns a well-trained medium-sized dog. Due to frequent business trips to other states during the weekends, she faces concerns about feeding her pet in her absence. While she often takes her dog along on these trips, it becomes a hassle. Serena is seeking a cost-effective solution to ensure her pet is well-fed when she is away.

- Sabrina is a 33-year-old female with 2 kids and an office job at Barnes and Noble. She likes coffee and is very insecure about her appearance. She doesn't like pets but her husband kind of pushed it on her and she ended up having to take care of the pet anyways. Sabrina is looking for ways to manage the pet efficiently, minimizing her direct involvement unless necessary.

1.4 Research of Existing Designs and Products

After identifying a problem we want to solve, we did research on what solutions are already available for pet owners out there on the market. The automatic pet feeders came in various designs and functionalities, but they generally shared common features. There were different designs on how the food was dispensed. Some were gravity-fed, where food drops down when the bowl empties and some were controlled by a motorized mechanism that dispenses a specific portion of food at set times. The most common power source methods are batteries or to plug into a power outlet. Some of the automatic pet feeders came with extra features such as portion control to prevent overeating and even webcam integration so pet owners can monitor their pets while they're away. All of them were made from plastic. One thing we also noticed was that many of the automatic pet feeders didn’t handle water. We wanted to make our automatic pet feeder something which had multiple features, since many of the pet feeders on the market only had a select few. This is why we added the dispense food and water functionality, as well as the webcam to take pictures.

1.5 Sustainability Statement

Our automatic pet feeder design prioritizes sustainability throughout its life cycle. We evaluate the environmental impact from manufacturing to disposal, focusing on reducing energy consumption, water usage, and material waste. By using durable, eco-friendly materials and implementing energy-efficient manufacturing processes, we aimed to minimize our ecological footprint. Furthermore, for instance, we used recyclable materials for our prototype such as cardboard and 3D printing material of PLA for the casing of our pet feeder food/water containers. Additionally, We ensured safe, recyclable disposal methods and adhere to ethical and regulatory standards, promoting the welfare of both pets and the environment. Our commitment to transparency and documentation ensures that all data collected on environmental impacts is available for review and continuous improvement.

2 Detail of Design

2.1 Aesthetic Prototype

2.1.1 CAD Images

Our final design is illustrated in Figure 1. It features a lid that fits into the bottom bowl, securing all the components inside the pet feeder using a slide fit mechanism. The design includes two compartments: top and bottom.

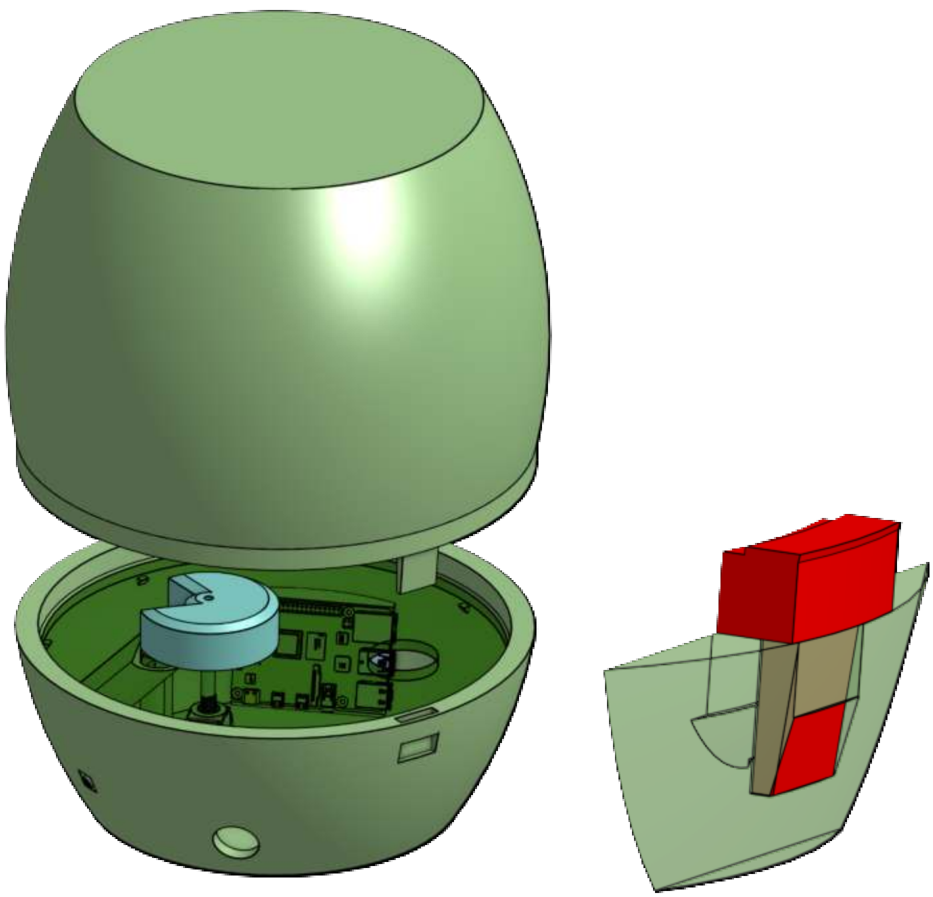


Figure 1: Overall design.

The top compartment will have a disk (Figure 3) that holds the food and water containers up, as shown in Figure 2. The food container is designed in such a way that the fan (Figure 4) can fit perfectly inside. This ensures that food doesn't just fall into the crevices during dispensing.

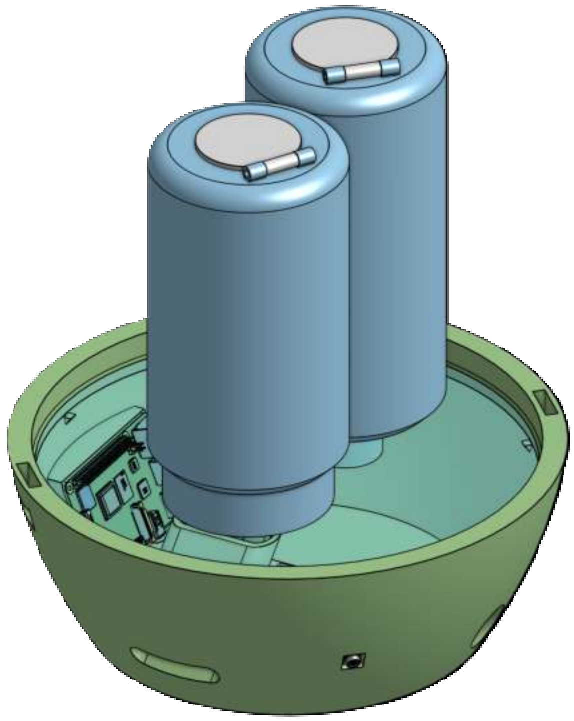


Figure 2: How the food and water containers are being held up.

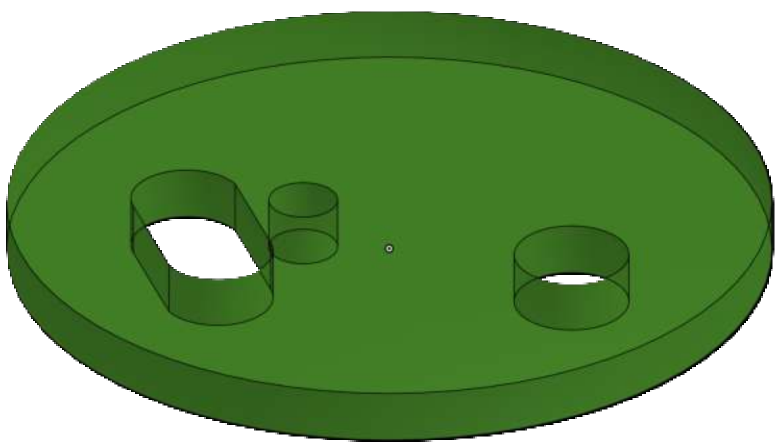


Figure 3: Disk that props the food/water containers up.

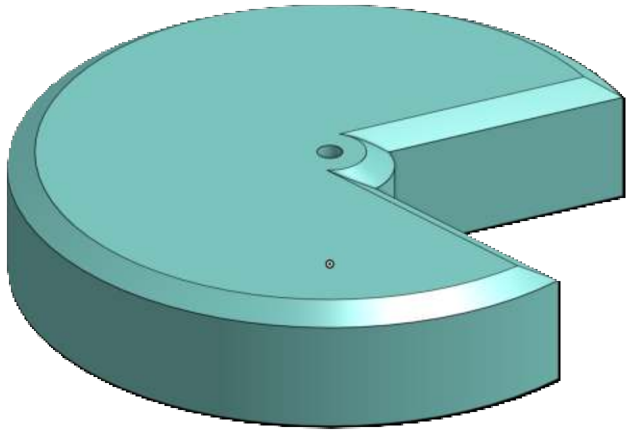


Figure 4: Fan.

Using the fan, we created a shaft (Figure 6) to attach it to the motor, enabling the fan to rotate and dispense food, as shown in Figure 7. The fan will sit on top of the disk, covering the hole through which the food would fall, depicted in Figure 5.

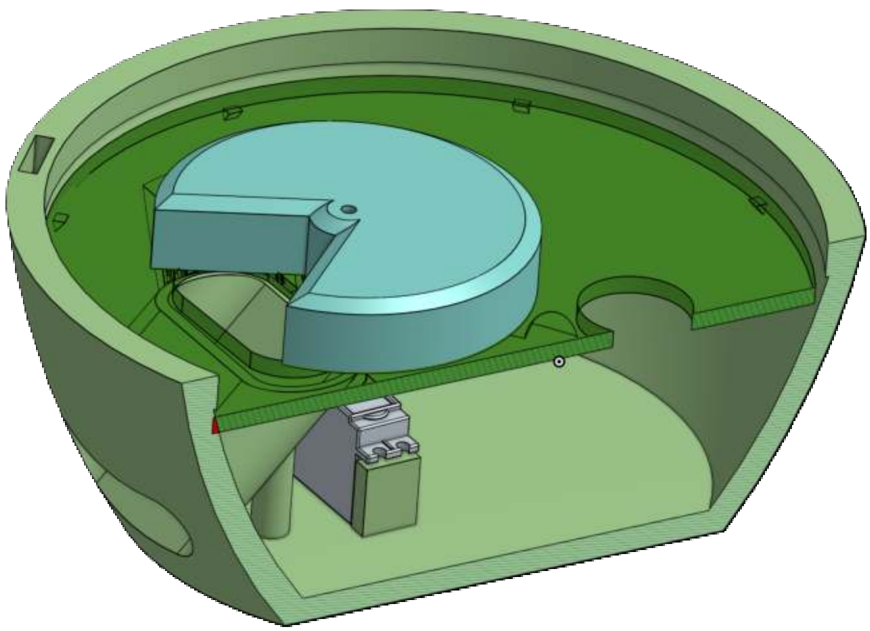


Figure 5: Fan placement on disk.

However, during testing, we observed that the shaft tends to break due to its thinness and the weight of the pet food on top of the fan. Additionally, the shaft frequently pops off the motor because we were unable to replicate the motor's teeth on our shaft.

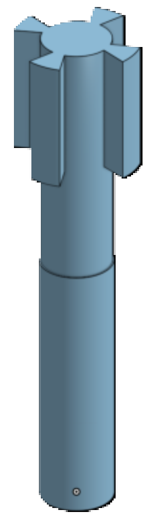


Figure 6: First shaft prototype.

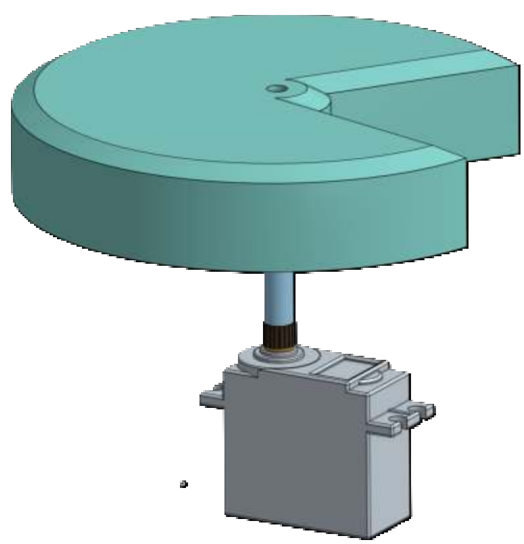


Figure 7: Fan-shaft-motor attachment.

To change up our design a little bit so that food is dispensed more efficiently and so that the shaft doesn't break, we created a thicker shaft (Figure 8) that sits on top of the adapter (Figure 9) that comes with the motor kit.

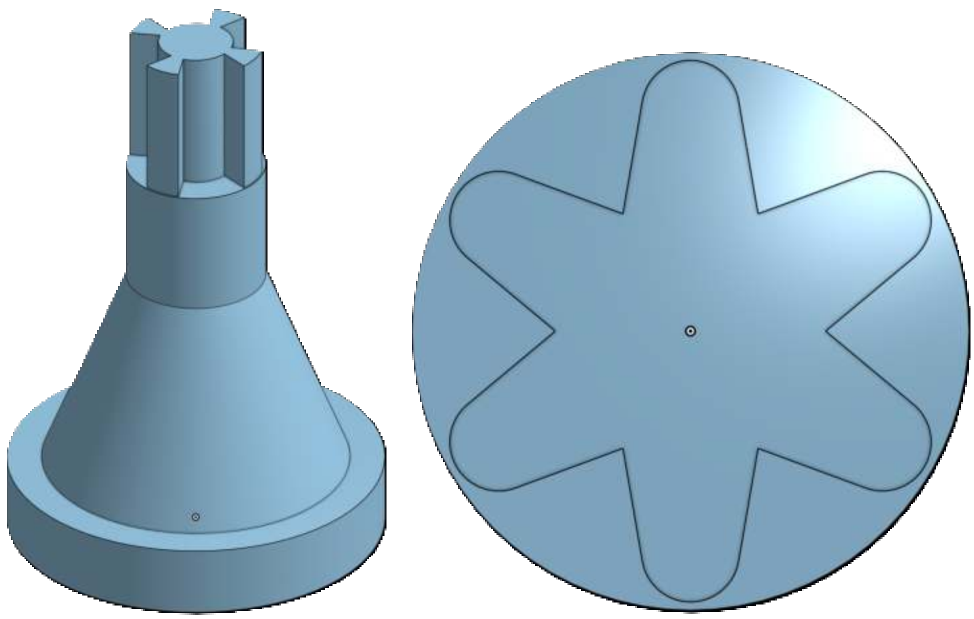


Figure 8: Second designed shaft.

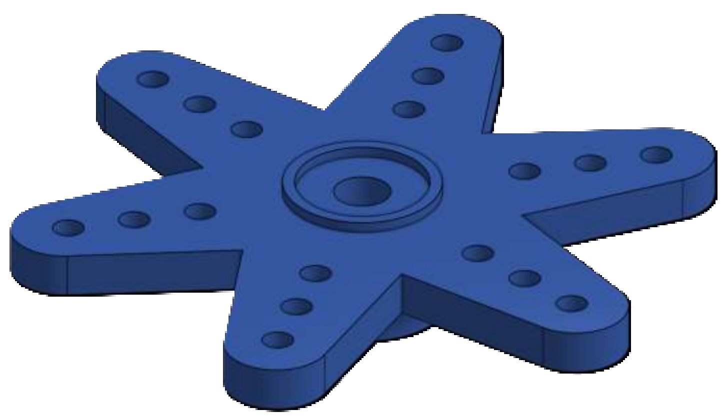


Figure 9: Adapter that sits perfectly on the gear of the motor.

To reduce the stress from the pet food on the fan, we added another layer that sits inside the food container, holding the food up, as depicted in Figure 10. The food will only fall down onto the dispensing hole when the fan rotates.

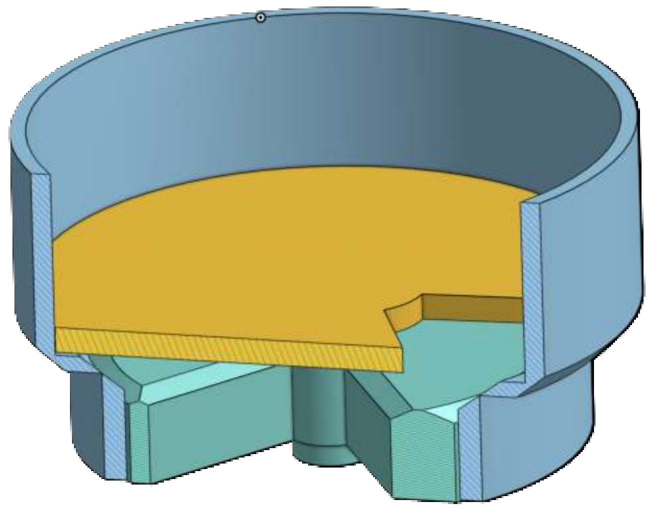


Figure 10: Inner view of the food container.

The bottom compartment houses all the hardware components, such as the Raspberry Pi 4, camera, motor, and wires to power and connect everything, as shown in Figure 11. The design includes designated areas for each main component. For example, the bowl has a cutout that allows the camera to peek through, as shown in FIgure 12. There are two tabs fitting and securing the motor with screws, as depicted in Figure 13. Additionally, we created a platform extending from the bowl to securely mount the Raspberry Pi 4 with screws, as illustrated in Figure 14.

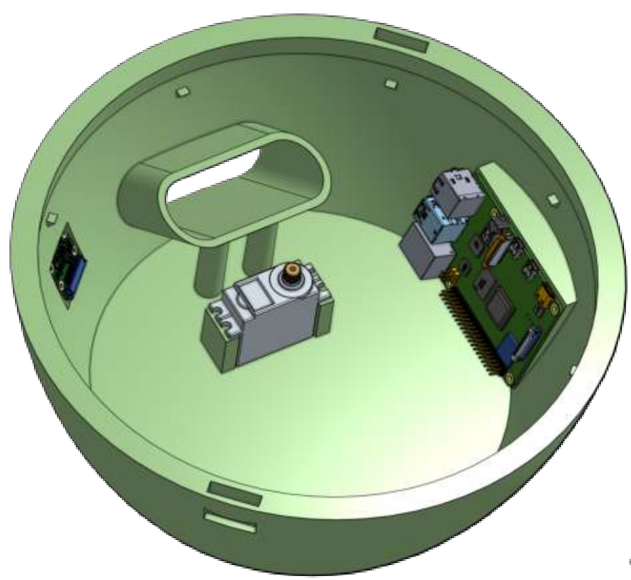


Figure 11: Overview of the bottom compartment.

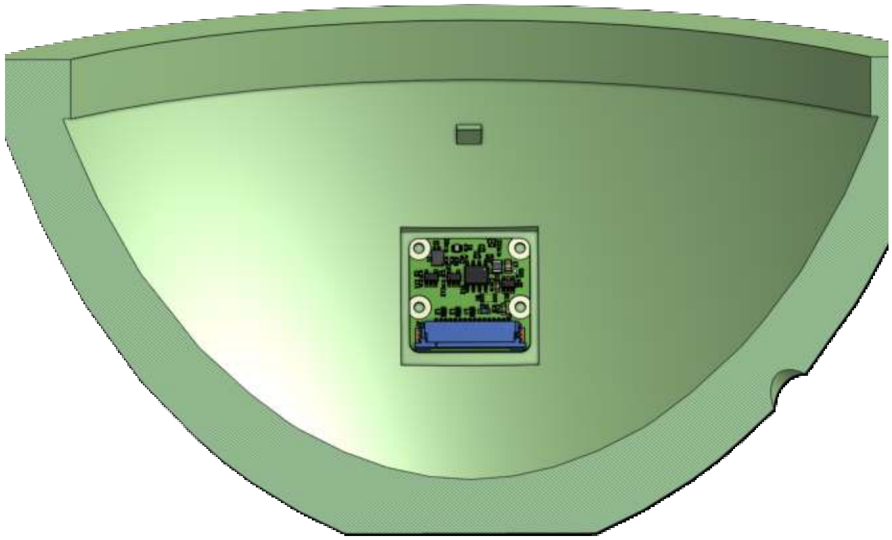


Figure 12: Camera placeholder.

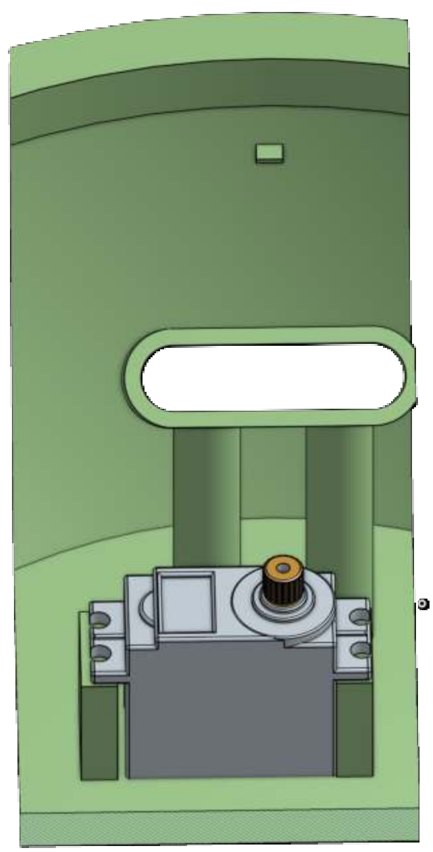


Figure 13: Tabs for motor placement.

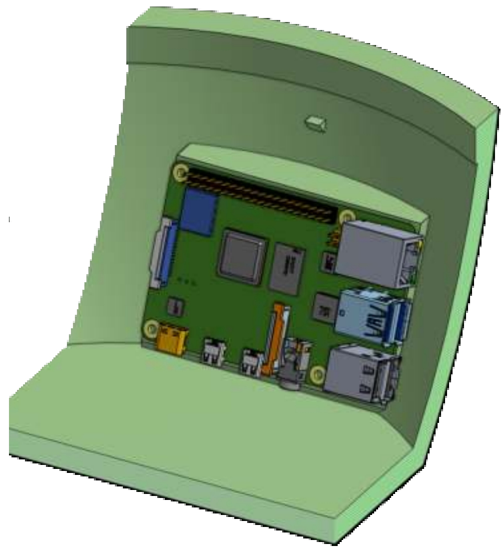


Figure 14: Platform for Raspberry Pi 4.

We designed our food and water containers from scratch to seamlessly integrate with other components of our pet feeder. As depicted in Figure 15, both containers feature lids that can be lifted up and down. The key distinction is that the mouth of the food container is significantly larger than that of the water container to accommodate the fan design shown in Figure 10.

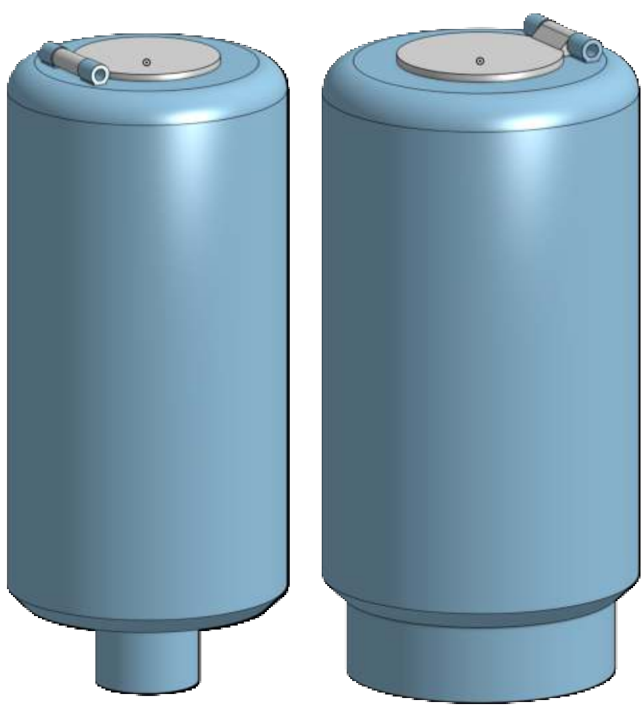
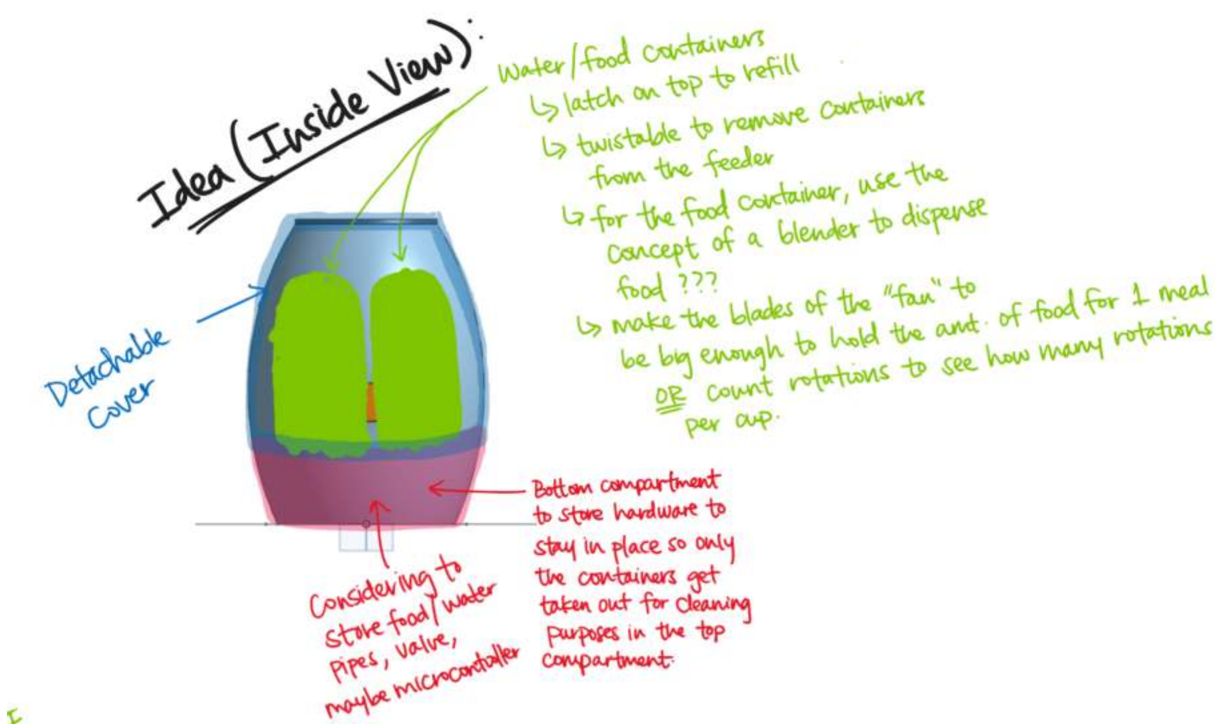


Figure 15: Food and Water containers.

2.1.2 Physical Prototype

Our aesthetic prototype is essentially a compact version of our functional prototype, designed to be more visually appealing. While we were unable to 3D print everything to fully represent our final product, Figure 16 provides an example of how it would appear when assembled.

Figure 16: 3D print of our aesthetic prototype.



2.2 Design for Manufacture, Assembly, Maintenance

2.2.1 Manufacture

Food Bowl/Water Bowl:

- Material: Stainless Steel bowls (final product)
- Manufacturability: Standard pet bowl but with a hole on the side, purchased through a store

Power Supply & Cable:

- Material: Raspberry Pi
- Manufacturability: Acquired from BELS

Water Dispenser: Faucet Waterer

- Material: Steel
- Manufacturability: Purchased from Amazon https://www.amazon.com/dp/B01AYD3R7A?psc=1&ref=ppx_yo2ov_dt_b_product_details

Food Dispenser: Fan Rotation System

- Material: 3D Printed Fan, MG 996R Motor
- Manufacturability: Cadded Design and then Printed through 3D Printer, and motor borrowed from Bels

Food Level Sensor:

- Material: UltraSonic Sensor
- Manufacturability: Sensor provided by CSE 121 Course

Water Level Sensor:

- Material: Liquid Level Switch Sensor
- Manufacturability: Purchased from Digikey

Food Container:

- Material: 3D Printed Container (Prototype)
- Manufacturability: Cadded Design and then Printed through 3D Printer

Water Container:

- Material: 3D Printed Container (Prototype)
- Manufacturability: Cadded Design and then Printed through 3D Printer

User Interface:

- Website and Mobile App which the pet owner can so the pet owner can control the camera and other functions for interaction
- User-friendly design with a simple display

2.2.2 Assembly

The assembly of the food and water bowls for the automatic pet feeder involves using stainless steel bowls. The choice of stainless steel ensures that the bowls are robust enough to withstand frequent use and are resistant to corrosion. The outer casing in our final product would be plastic, to ensure durability and a potential While the containers for the food and water within the feeder are ultimately intended to be made of stainless steel for their robustness and ease of cleaning, the current prototype utilizes 3D printed containers.

While the containers for the food and water within the feeder are ultimately intended to be made of stainless steel for their robustness and ease of cleaning, the current prototype utilizes 3D printed containers. These prototypes allow for rapid testing and modification of the design at a low cost. In the final design, the outer casing of the automatic pet feeder is constructed from high-quality plastic, which is both lightweight and durable. This choice of material ensures that if the feeder is accidentally knocked over by the pet, it won't shatter or break, thus maintaining its structural integrity and continuing to function as intended. This resilience is crucial in a household with active pets that might bump into the feeder during play or feeding times.

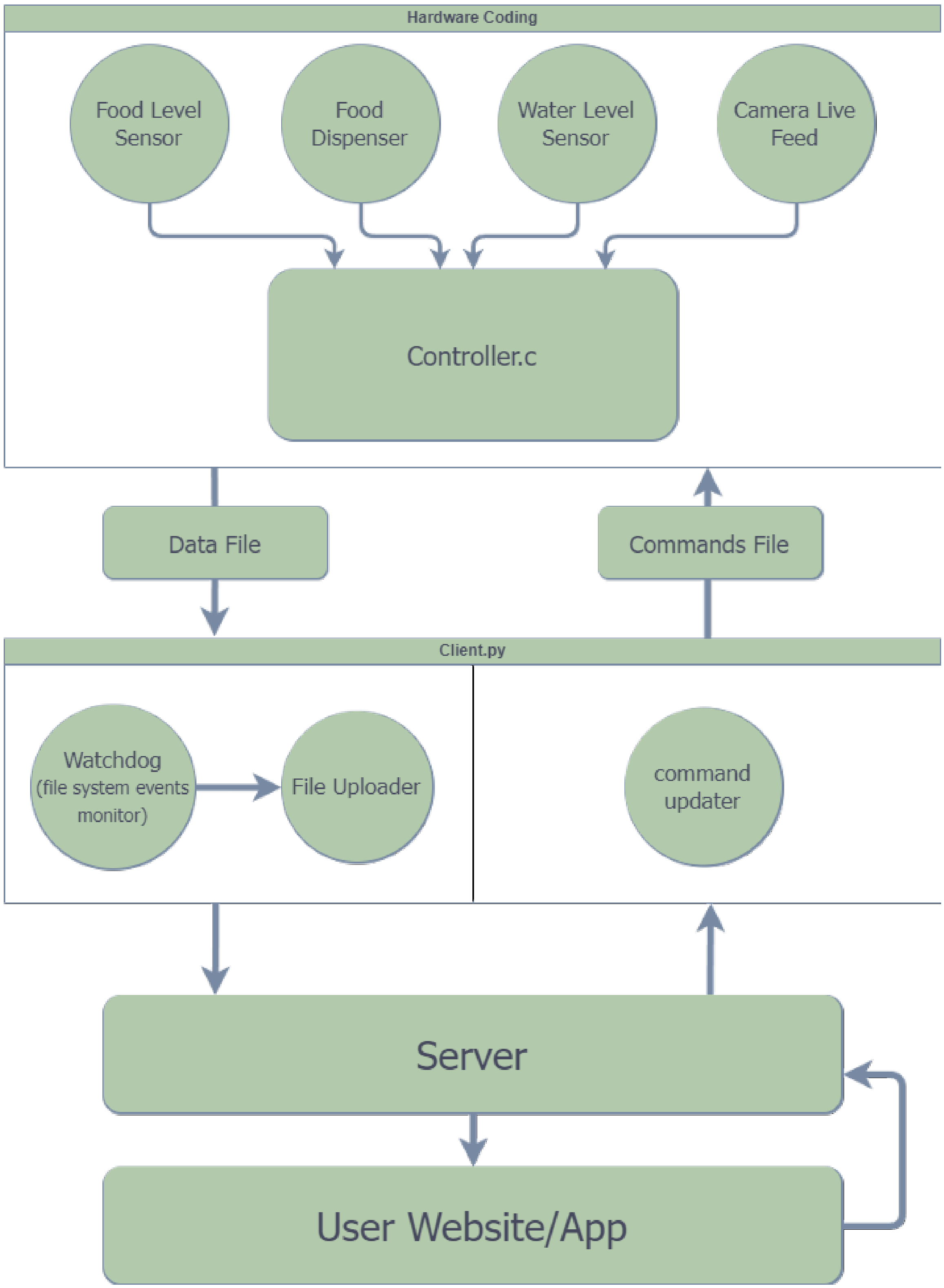
Additionally, the design incorporates a snap-lock system on the side of the feeder. This feature is a significant enhancement in preventing accidental spills. The snap-lock ensures that the feeder remains securely closed, even if the pet attempts to paw or push against it. This system not only keeps the food contained but also maintains the cleanliness of the feeding area by preventing food from spilling out and attracting pests or creating messes.

2.2.3 Maintenance

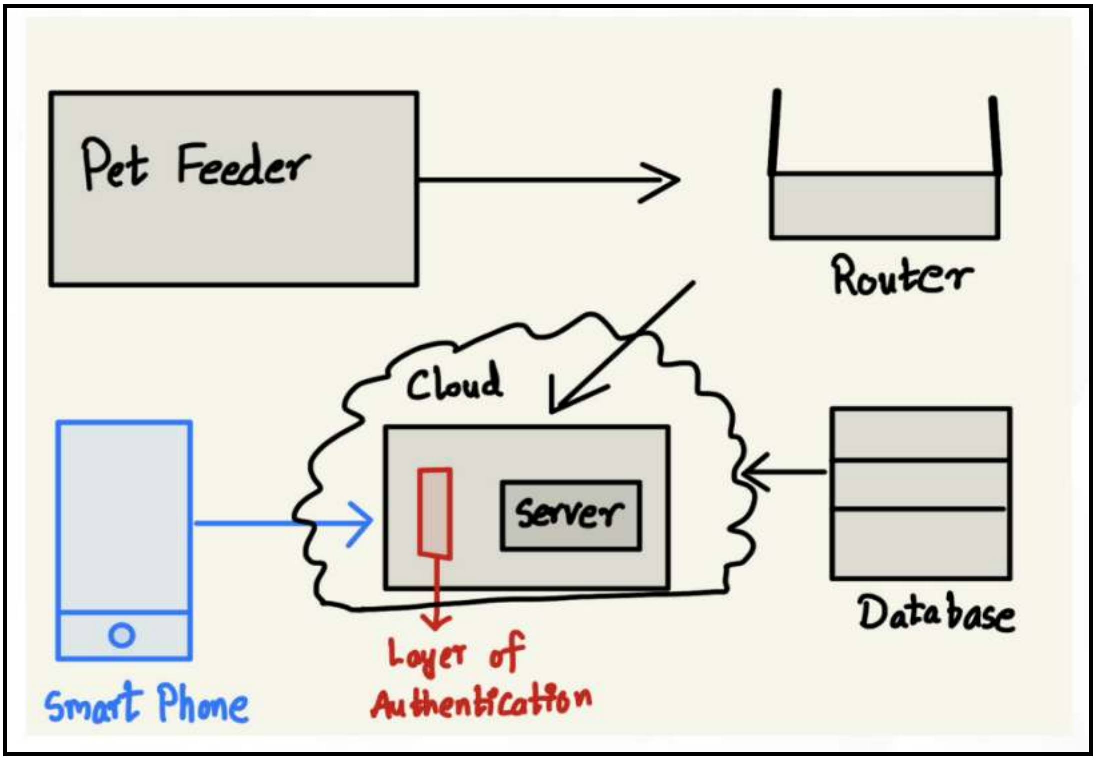
Maintaining an automatic pet feeder is crucial to ensure it functions properly and provides a reliable source of food for pets when their owners are away. Regular maintenance should start with the basics: checking and refilling the food compartment, ensuring it is free from any clogs or debris that could impede the flow of food. It's essential to clean the feeding bowl, the dispenser mechanism, and the water faucet periodically, using warm soapy water to prevent the buildup of food particles and bacteria that could potentially harm the pet. Additionally, the internal mechanisms and the dispensing chute should be inspected for any signs of wear and tear or mechanical issues, such as stuck motors or misaligned gears, which could disrupt the feeding schedule. Adhering to these maintenance steps will ensure the automatic pet feeder remains a reliable, efficient, and hygienic way to manage pet feeding schedules.

2.3 Block Diagram

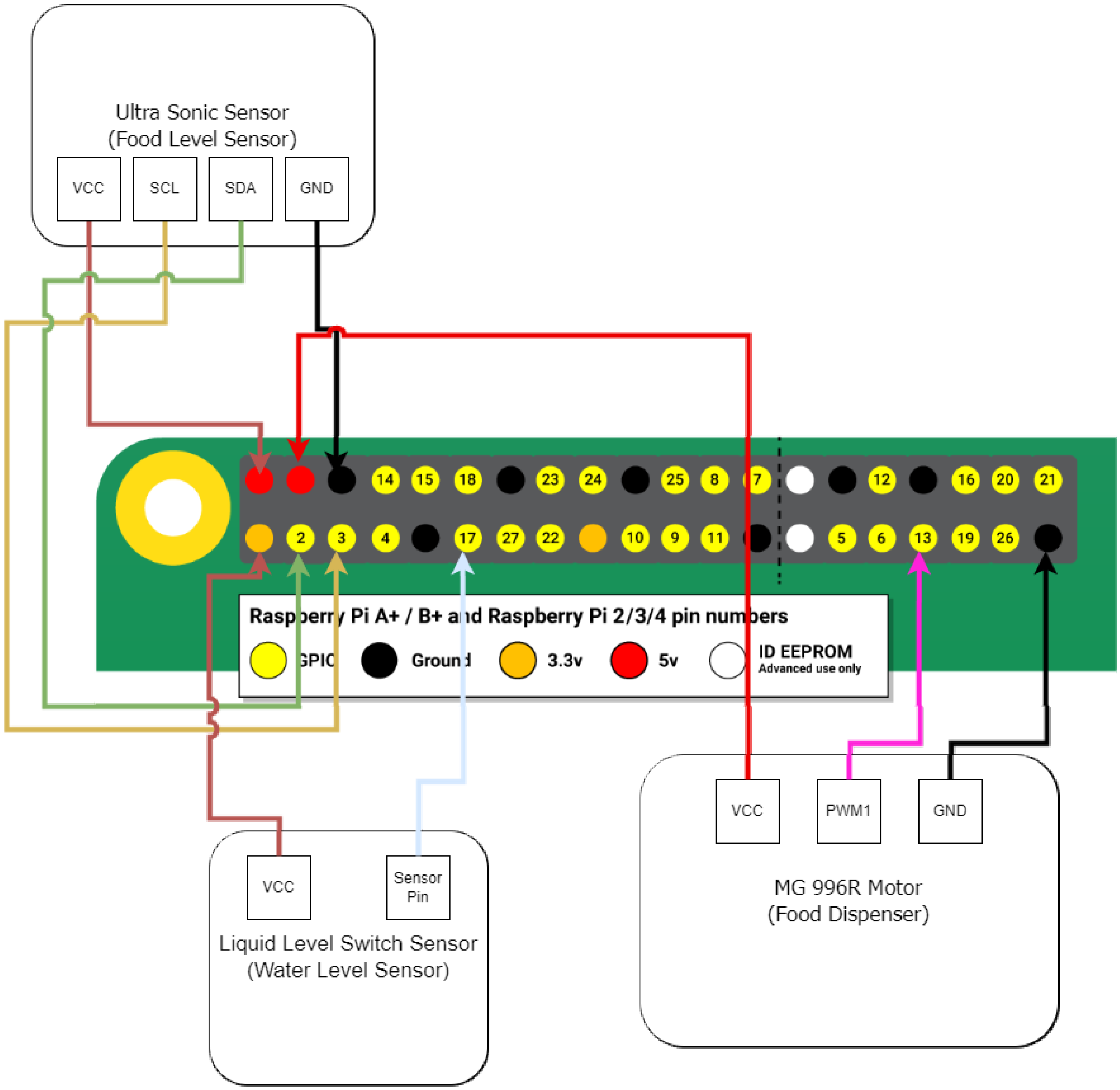
a. Hardware Block Diagram



b. Software Block Diagram

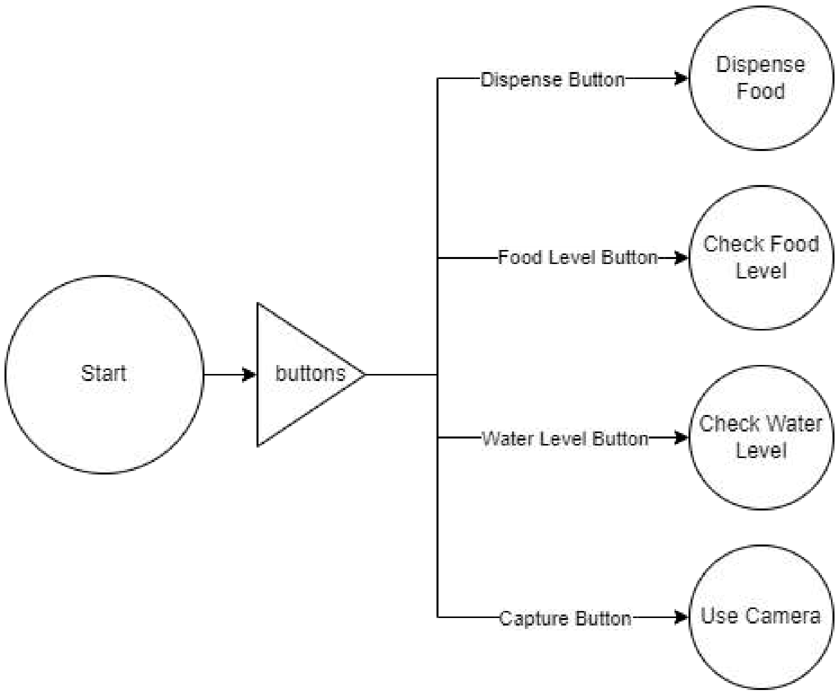


2.4 Wiring Diagram

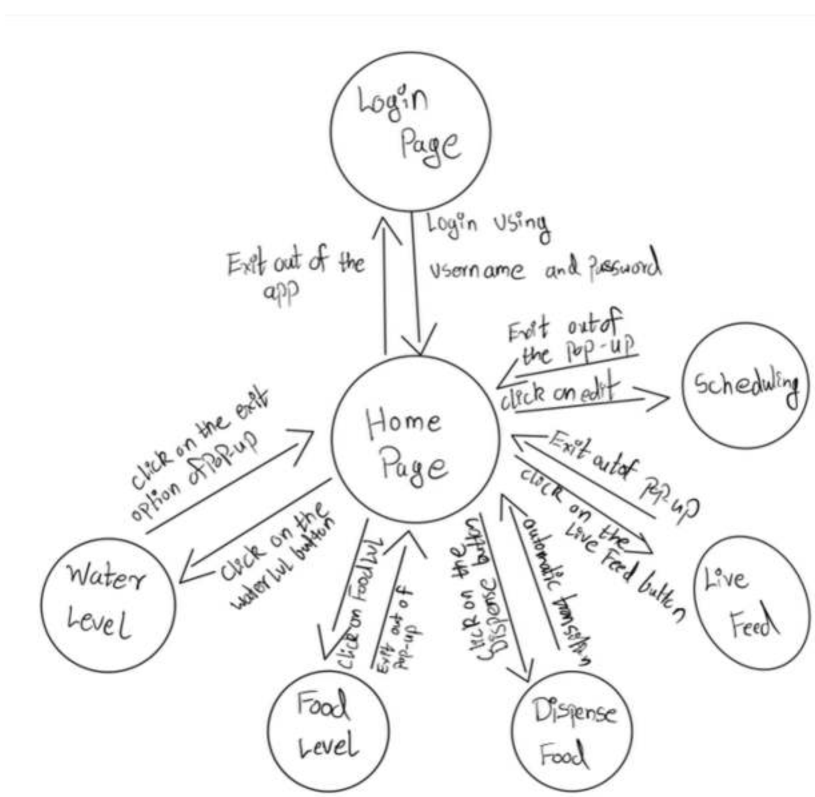


2.5 State Transition Diagram

Hardware State Diagram:



Software State Diagram:



2.6 Technology

2.6.1 Software

Backend: Python, Javascript
Purpose: Python was chosen for its simplicity and robust libraries to create the client-server architecture. The client-server system handles communication between various components, relays commands, and transfers data efficiently. Javascript was also used to connect to the API endpoints in the Python server.

Implementation:

- Client Hardware and Software Server Communication: Python's socket programming capabilities were utilized to establish and maintain a reliable communication channel between the client and server.
- Data Handling: The Python Flask server connected to endpoints of the database management system, Google Firebase, where data was dynamically stored.
- Python Backend Flask Server: The Python server used Flask to create a web server that interacted with a Firebase database and managed file uploads and user sessions. It had Firebase integration, file management, authentication and authorization, API endpoints, error handling, and cross-origin resource sharing (CORS).

Frontend: HTML, CSS

Purpose: There were interactive UI Elements for checking food and water levels, dispensing food, and accessing live feed. There were also UI Elements for a user login page for the application.

Implementation:

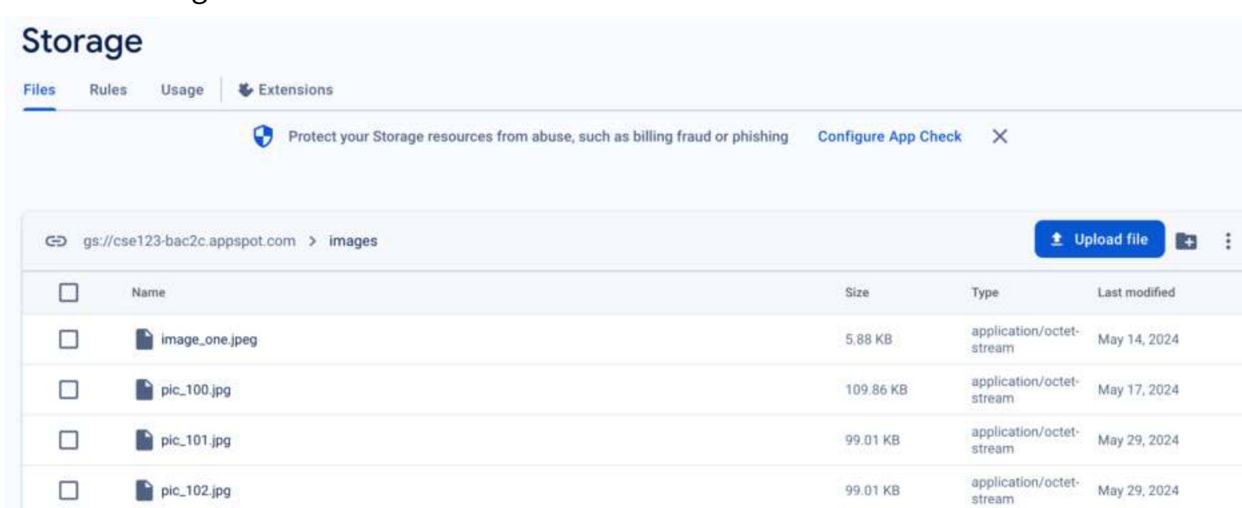
- Pet Feeder Dashboard: Included JavaScript functions connected to API endpoints that allowed the user to fetch real-time data on food and water levels and send commands to the server to perform actions like dispensing food and capturing an image. This interaction was done through AJAX calls, ensuring the page does not reload upon submission. CSS was used for styling.

- Login Page: A user interface for logging into CSE123's Pet Feeder web application. It was set up to post data to the /login endpoint when submitted and handled it as an AJAX request. When the form was submitted, the JavaScript fetched the username and password, sent them as JSON to the server using a POST request, and expected a response containing an API key if the credentials were correct. If the login was successful and an API key was returned, the script stored the API key in the browser's session storage and redirected the user to the Pet Feeder dashboard.

Google Firebase Realtime Database and Firestore
Purpose: Realtime and Firestore were used to store the data on the server.

Firebase Database:

Firebase Storage:

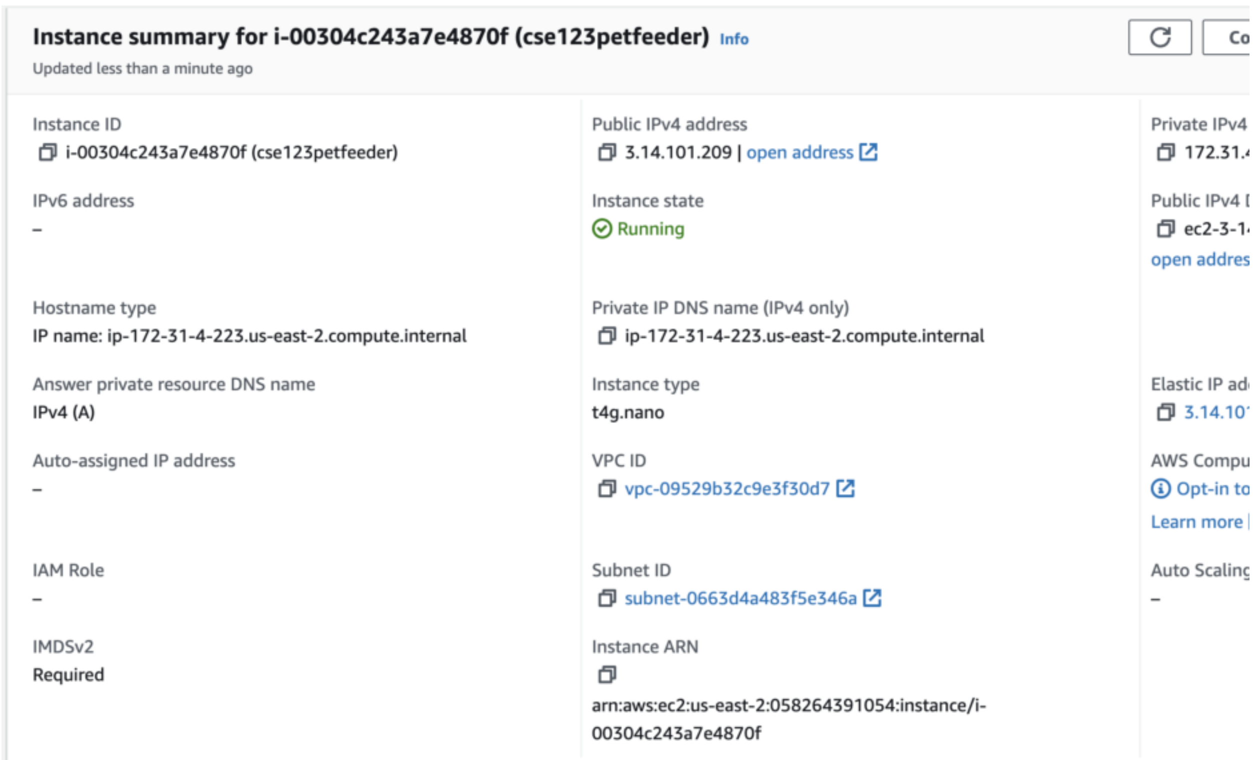


Implementation:

- Database Operations: Firebase Realtime Database was used to store user login and registration information, which included email, password, and a unique API key.

Furthermore, it stored the food and water level readings from the hardware client side that were sent to the server. Moreover, Firebase Storage was used to store images from our camera.

Amazon Elastic Compute Cloud (AWS EC2 Instance):



Purpose: For cloud deployment, Amazon AWS EC2 instance was utilized and implemented for a reverse proxy with Nginx. Additionally, HTTPS certifications were purchased to ensure secure connections when accessing our domain: https://cse123petfeeder.com.

Implementation:

- Documentation for Cloud Deployment with AWS EC2 and Nginx: Use the permission key issued by AWS to SSH into the EC2 instance. Securely copy the server-side files to the EC2 instance. Configure the nginx.conf file in /etc/nginx to set up a reverse proxy with Nginx and Flask.
- Nginx Configuration: Restart the Nginx service to apply the new configuration. Verify that the server is accessible via the configured domain and that the reverse proxy is correctly routing traffic to the Flask application.

C Programming Language

Purpose: C was employed to program the hardware components directly due to its low-level access to system resources and efficient execution.

Implementation:

- Hardware Interaction: The control and monitoring of the MG 996R Motor, UltraSonic Sensor, Liquid Level Switch Sensor were managed through C, allowing for precise and real-time hardware interfacing. 2.6.2 Hardware

The hardware setup for our project included the following components, all connected to the Raspberry Pi 4.

Raspberry Pi 4:

- Role: The central hub of our hardware setup.
- Functionality: Equipped with multiple GPIO pins, USB ports, and a powerful processor, the Raspberry Pi 4 managed all connected sensors and actuators. It facilitated data collection, processing, and communication with the server.

MG 996R Motor:

- Functionality: The servo motor was utilized to control the fan responsible for dispensing food from the container to the dispensing hole. Being a servo motor, it allowed precise control over the degree of rotation, ensuring the fan rotated to an angle that facilitated food falling into the dispensing hole.
- Integration: Controlled via the GPIO pins of the Raspberry Pi 4 (specifically VCC was connected to 5V pin, the ground was connected to the GND pin, and the control pin was connected to GPIO pin 13.), programmed using C for accurate and reliable movement tasks.

UltraSonic Sensor:

- Purpose: The UltraSonic Sensor was utilized to measure the food level by calculating the distance from the sensor to the bottom of the food container. Mounted at the top of the container, the sensor could detect changes in food level as the distance decreased.
- Integration: Controlled via the GPIO pins of the Raspberry Pi 4 (Specifically the VCC was connected to 5V pin, the ground was connected to the GND pin, the SDA was connected to GPIO pin 2, and the SCL was connected to GPIO pin 3.), programmed using C to provide real-time data on food level proximity.

Liquid Level Switch Sensor:

- Purpose: This Liquid Level Switch Sensor was used to determine the water level inside the water container.
- Integration: Controlled via the GPIO pins of the Raspberry Pi 4 (Specifically the VCC was connected to 3.3V pin and the sensor was connected to GPIO pin 17.), programmed using C to ensure action when called upon.

Raspberry Pi Camera:

- Purpose: The camera was used for visual data capture of the pet when the pet is near the pet feeder.
- Integration: Connected to the Raspberry Pi 4, it enabled image processing tasks, essential for surveillance and monitoring of pets.

2.6.3 Database

Firebase:

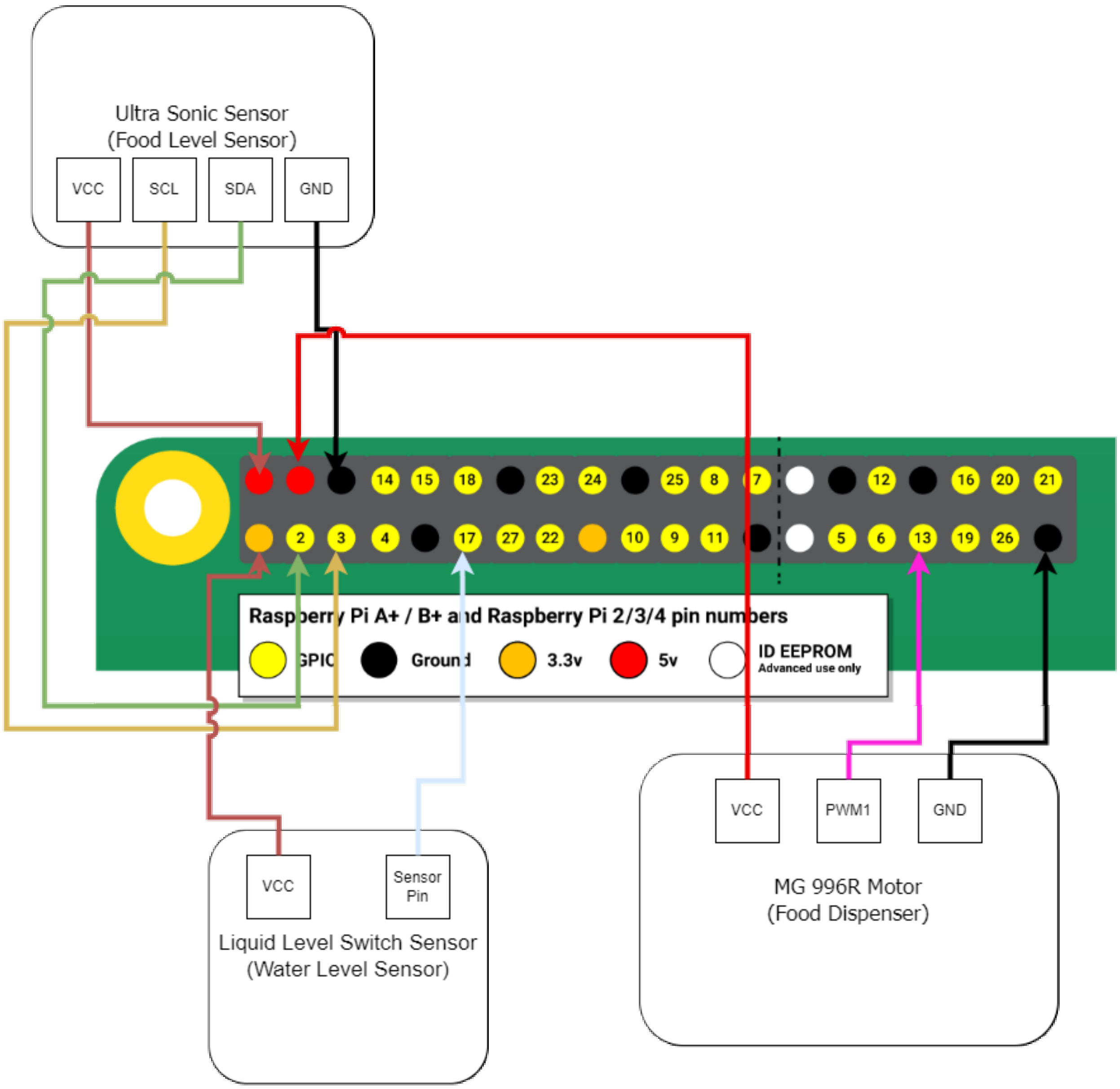
- Role: Firebase was chosen as our database solution to store and manage data
- Functionality: Firebase provided real-time database and backend as a service, allowing seamless integration with our Python-based server. It supported real-time data synchronization and ensured the reliability and scalability of our system. The data collected from the sensors and commands were stored in Firebase, enabling efficient data retrieval and analysis.

The seamless integration of software and hardware was crucial to our system's success. Python, Javascript, and C programming languages were effectively utilized to manage communication, hardware control, and data storage, respectively. The Raspberry Pi 4 served as a central hub, connecting various sensors and actuators, while Firebase ensured reliable data management. This cohesive integration allowed our system to perform complex tasks efficiently and reliably.

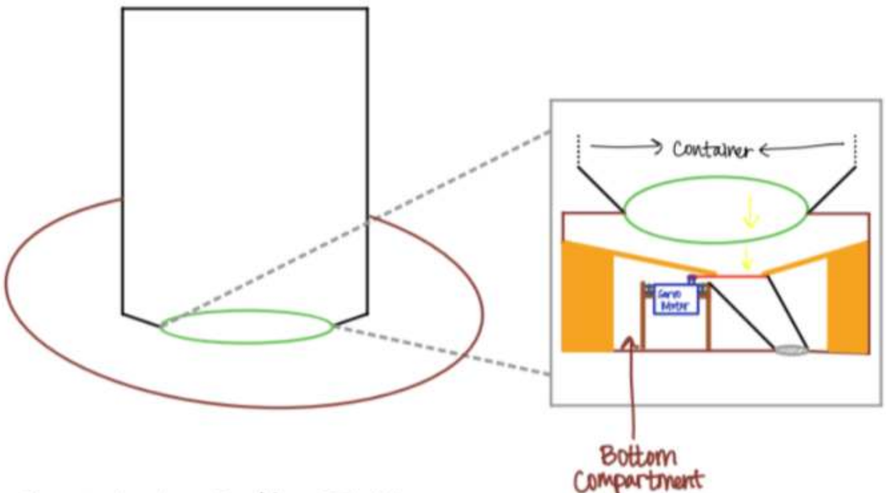
2.8 Modeling

Hardware:

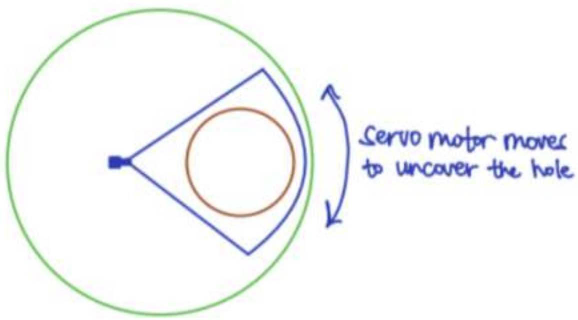
Based on our wiring diagram reposted in the figure shown below:



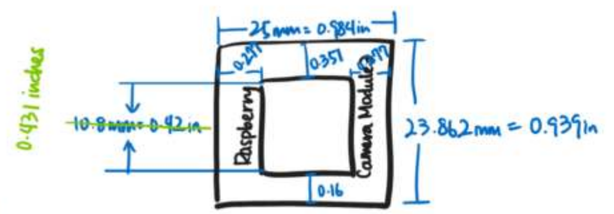
Each component on the Raspberry Pi is powered by individual pins designated for supplying power. Specifically, the ultrasonic sensor operates with a 5V power supply. The liquid level switch sensor requires a 3.3V power supply, while the MG996R motor, needing 5V, is powered by a different pin than the one used for the ultrasonic sensor's 5V supply.

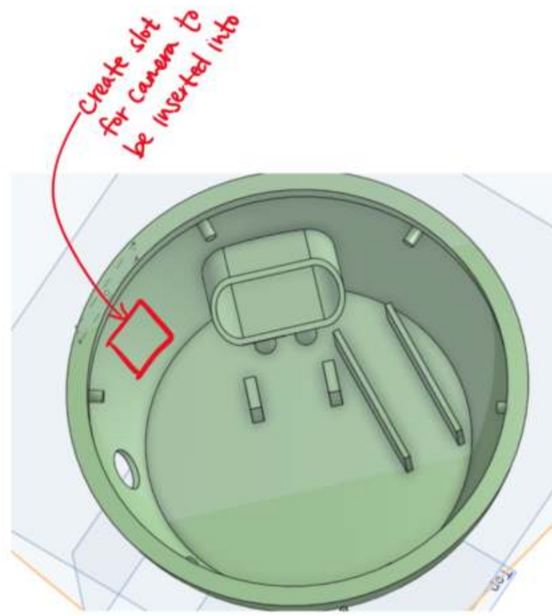


Servo Motor Section Top View

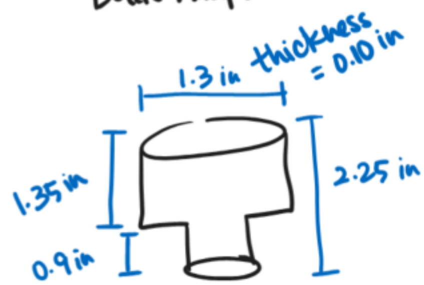


Raspberry Pi Camera

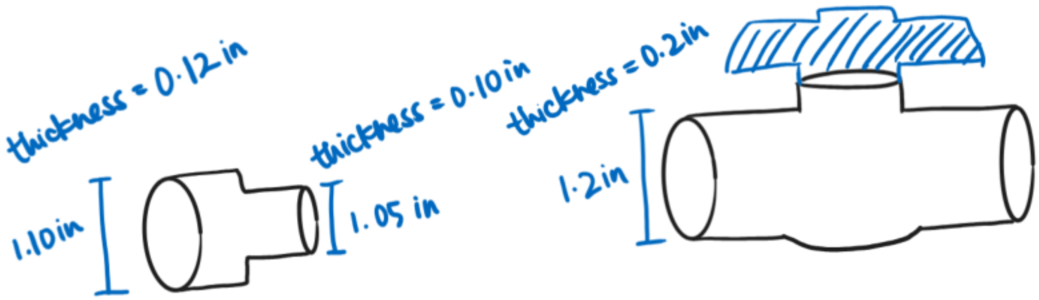




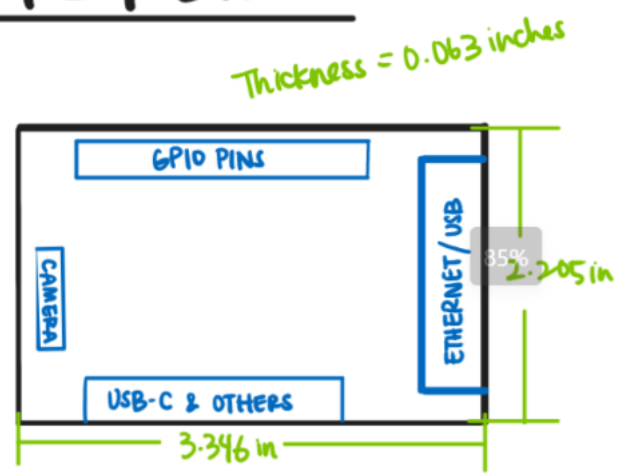
Bottle Adapter — Diameter of Jug = 1.5inches



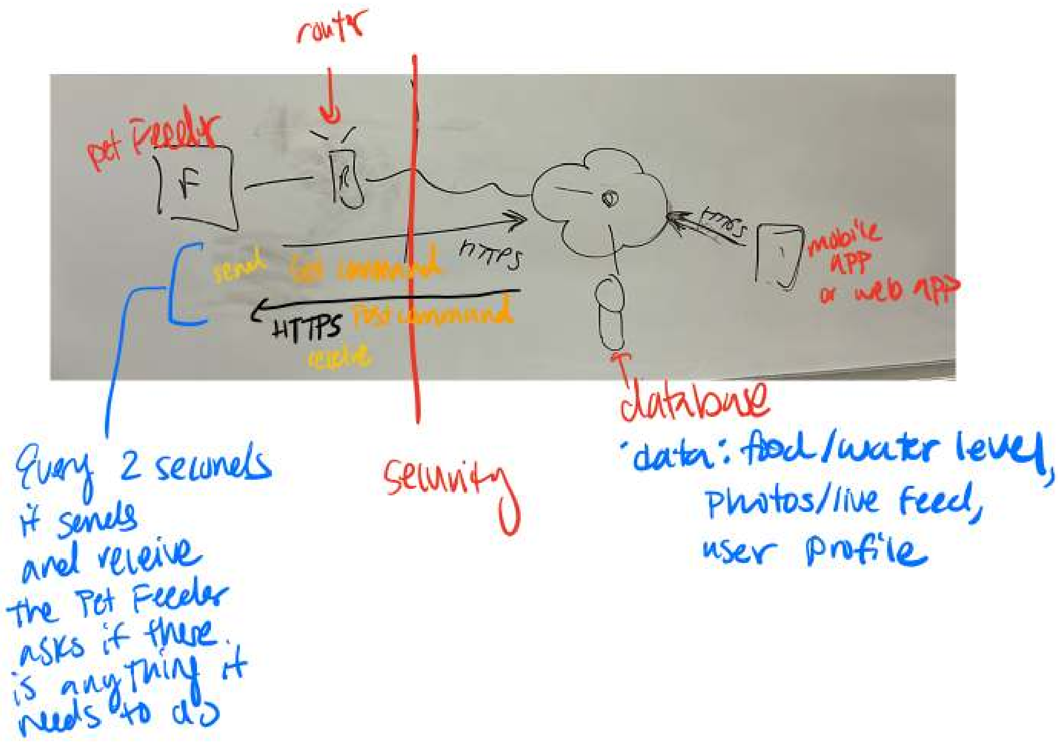
$$\begin{array}{r} 1.5 \\ -\ 0.05 \\ \hline 1.45 \end{array}$$



RPI4 Board

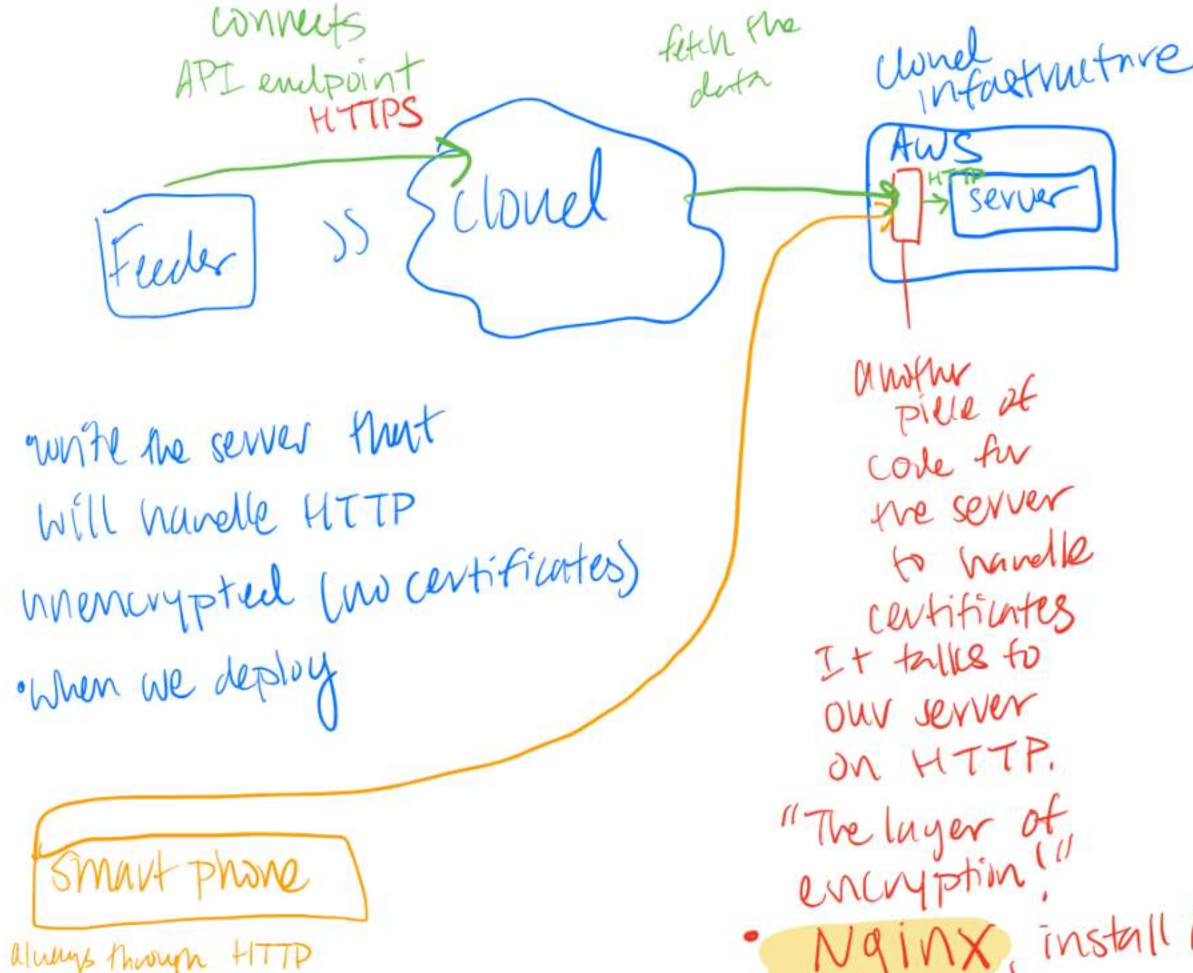


Software:



Other cases:

·Multiple pet feeders,
unique_id match to
IP address to authenticate
pet feeder → user
- Token, PI

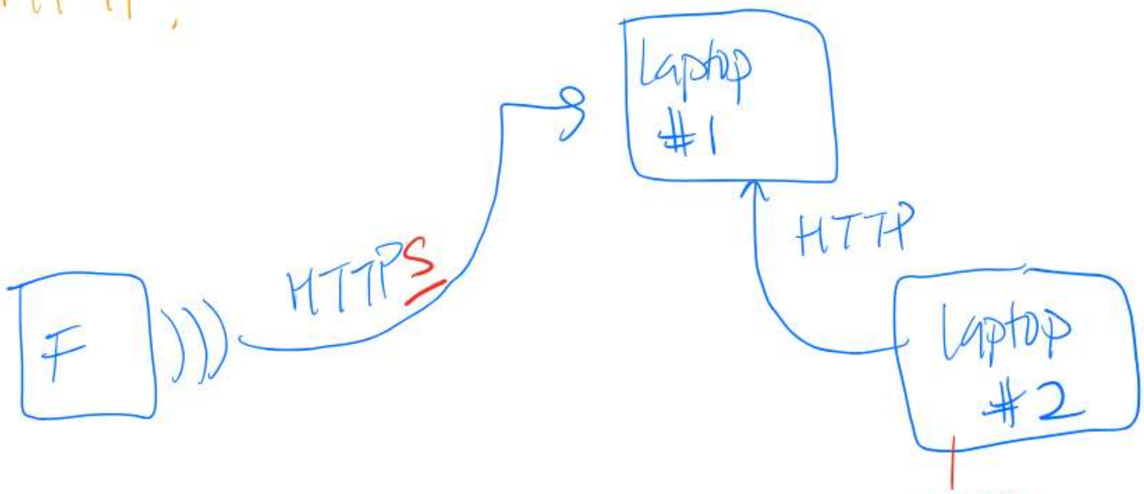


write the server that
will handle HTTP
unencrypted (no certificates)
• When we deploy

another
piece of
code for
the server
to handle
certificates
It talks to
our server
on HTTP.
"The layer of
encryption!"
• Nginx, install it
in AWS instance,
who to talk to in server,
put its in the outside
world. We need this
for deploying

Next steps:

- Take out certificate stuff and run HTTP!



person browser can see the data to see if it works before deploying into the cloud

- we can run all of this on laptop.
- Raspian, raspberry pi, (c or python?)

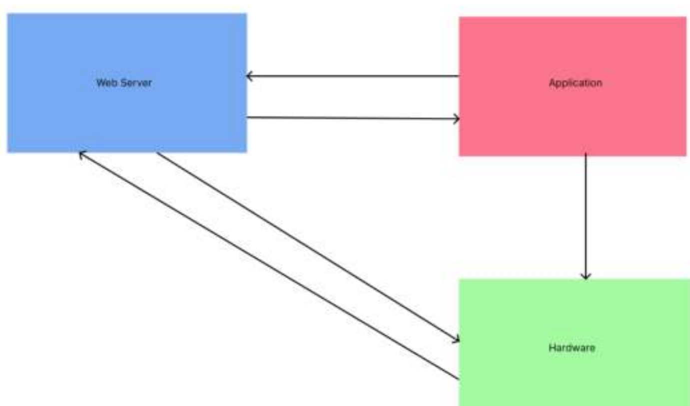
• Laptop : VM, server, Browser

→ then deploy the server to the Pi/feeder

* Pi needs to talk to server, someone needs to work on reading data from sensor. Pi needs to tell the valve to turn on.

Application:
- Frontend UI provides a UI interface to interact with the application
  - Allows users to register/login, manage pets, set feeding schedules, and trigger immediate feeding actions
- Frontend sends requests to the web server to add/edit/delete pets and set feeding schedules
- Webserver updates the database accordingly

Web Server:
- Responsible for handling the HTTP requests from the fronted
- Communicates with the database and micro controller

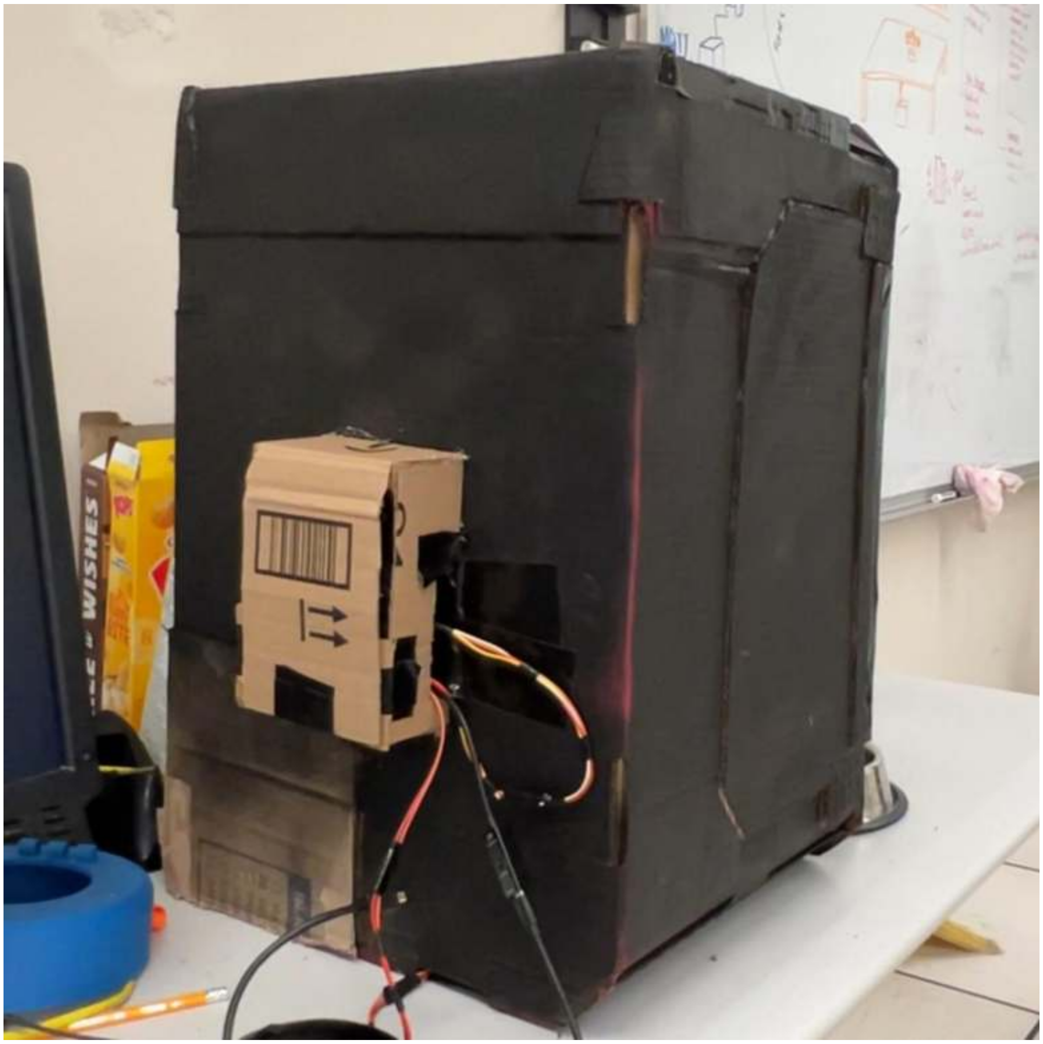


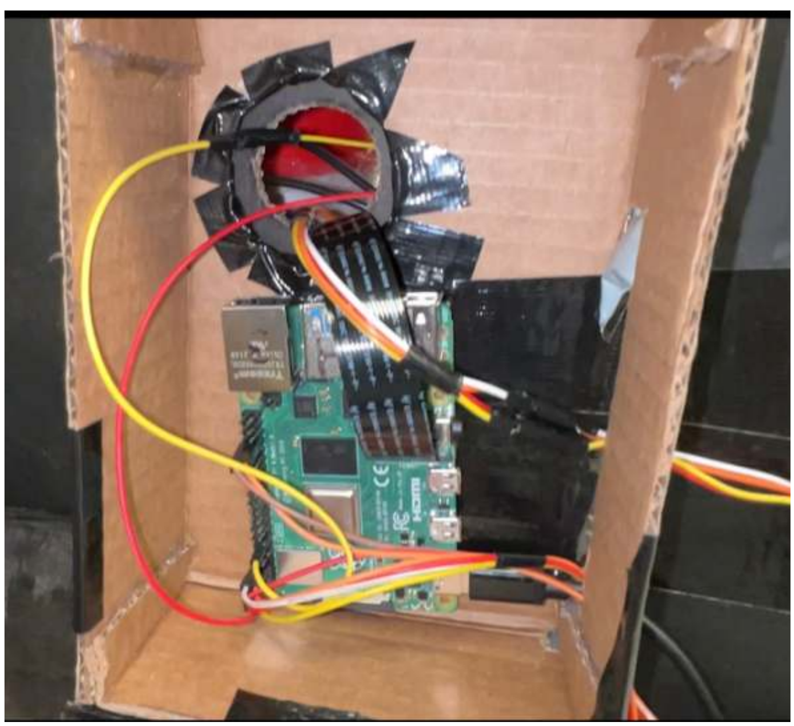
Hardware:
- Controls the automatic pet feeder hardware
- Communicates with the web server over Wi-Fi or Hotspot to receive commands for dispensing the food and water. Also by using the camera

3 Evaluation

3.1 Functional Prototype

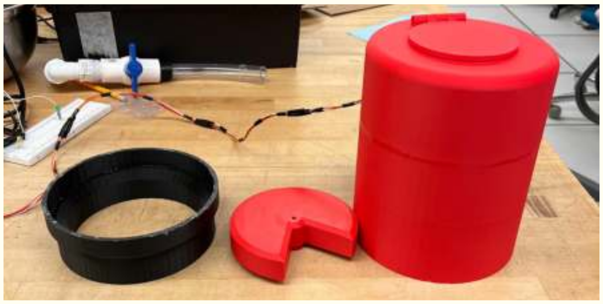
3.1.1 Photograph of Functional Prototype (Front, Back, Raspberry Pi)

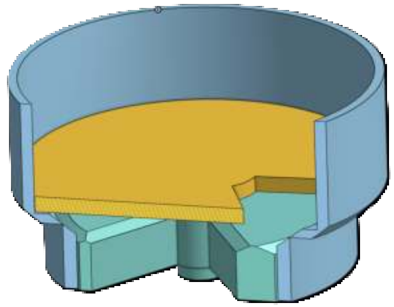




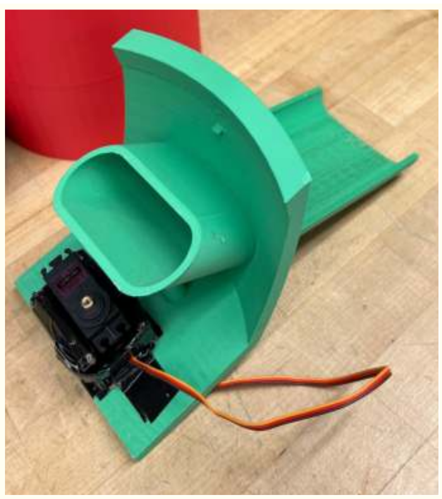
3.1.2 Details

The first photo showcases the front side of our functional prototype, highlighting two 3D-printed cylindrical containers.





The container on the left is designated for food. Inside the food container, an ultrasonic level sensor is mounted at the top. Additionally, a 3D-printed plate is positioned to reduce the load on the food wheel beneath it. This food wheel, shaped like Pac-Man, is connected to a robust extension stick, which is driven by an m996r motor to dispense food. There is also a green slide connected to the exit mouth for smooth food dispensing.



The container on the right is for water. Inside the water container, a liquid level switch sensor is installed. A tube connected to the water container features a lick faucet for easy dispensing.

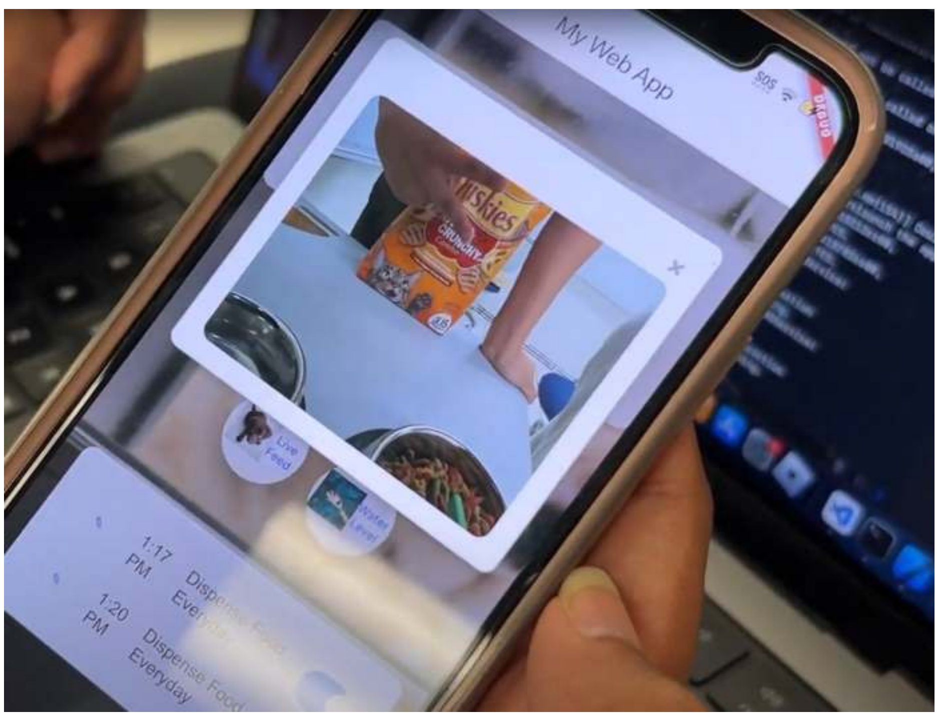
Liquid Level Switch Sensor:

3.2 Testing/Simulation

Full bowl:

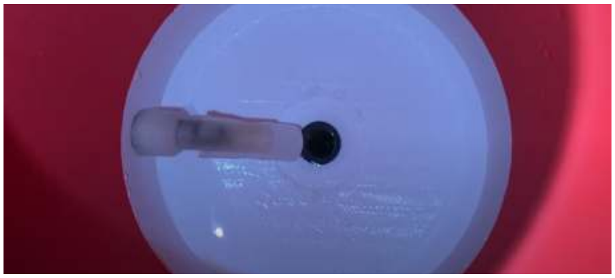
Small bowl:

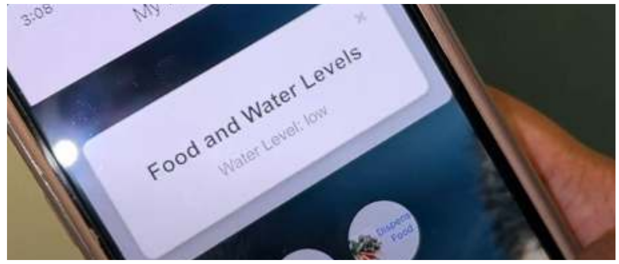
Image captured through camera feed:



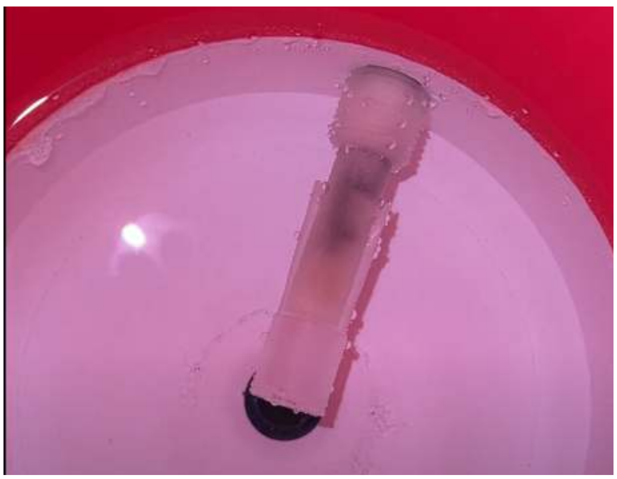
Water Level:

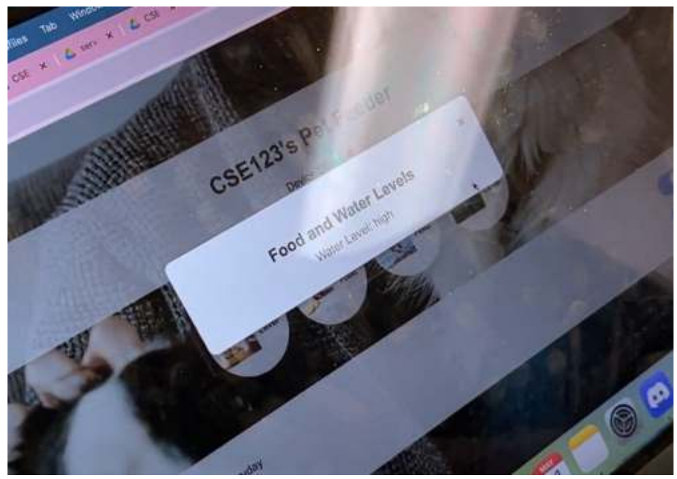
Low:





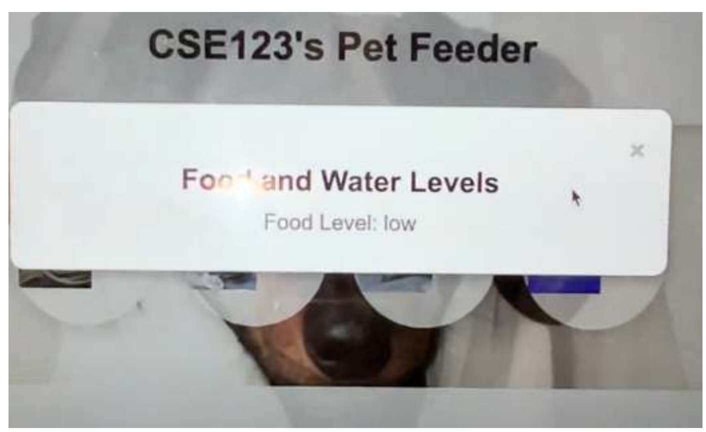
High:





Food Level:

Low:



High:

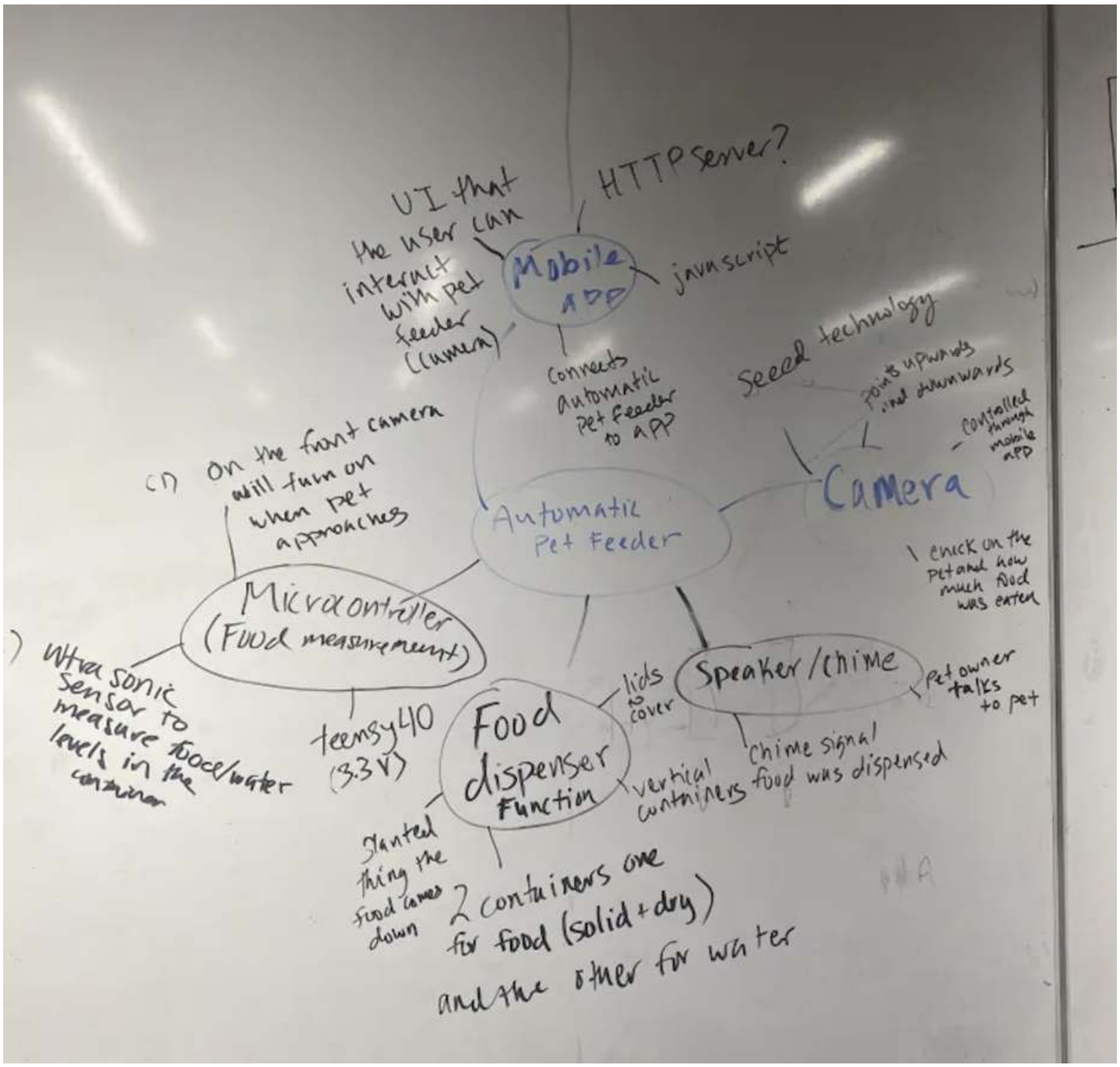
The conducted tests aimed to evaluate the prototype's functionality, reliability, and user interface across various parameters, including food and water dispensing accuracy, camera feed clarity, and app communication efficiency. In the Food Dispensing Accuracy Test, the system consistently delivered precise amounts of food within acceptable error margins, demonstrating its reliability in meeting pets' nutritional needs accurately. Similarly, the Camera Feed Quality Test revealed satisfactory performance in providing high clarity images, although latency issues were observed, highlighting the need for optimization to improve real-time monitoring capabilities. Additionally, the Website/Mobile App Usability Test indicated a moderate level of app responsiveness, with variability in user satisfaction scores suggesting areas for improvement in usability and functionality. Furthermore, the water and food level testing indicated reliable sensor performance in maintaining accurate levels of water and food, contributing to the overall reliability of the system.

Moving forward, it is recommended to prioritize optimization efforts to address latency issues in the camera feed, enhance user satisfaction through iterative improvements in the mobile app interface, and implement continuous testing and feedback cycles to drive ongoing optimization and ensure the product remains competitive in the market landscape. Providing comprehensive user education materials and responsive customer support channels will further assist users in maximizing the product's potential and navigating any challenges encountered during usage. By staying abreast of emerging trends and technological advancements, the prototype can remain relevant and adaptable to meet evolving user needs and preferences effectively.

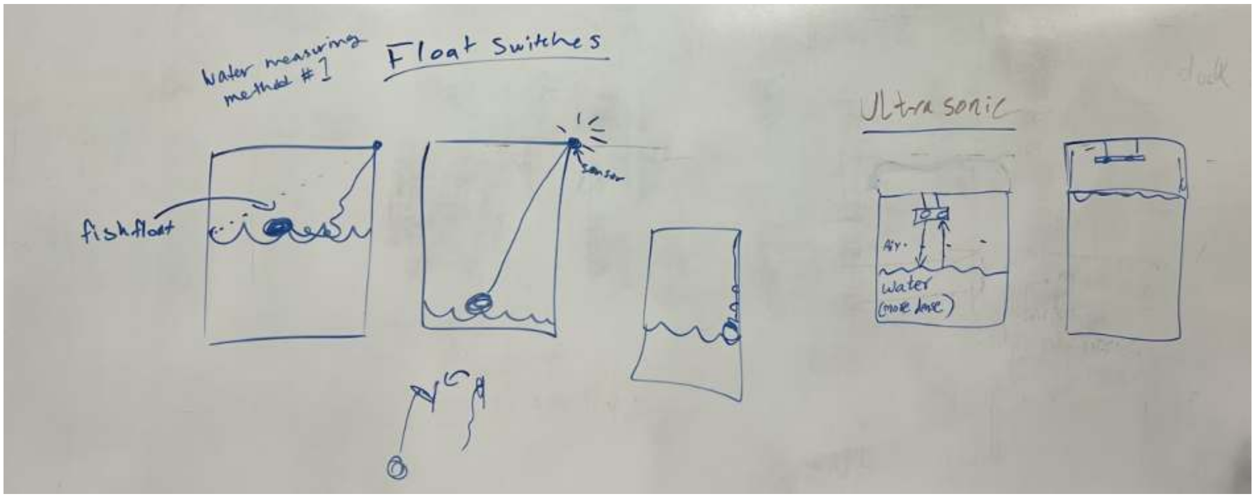
Appendices

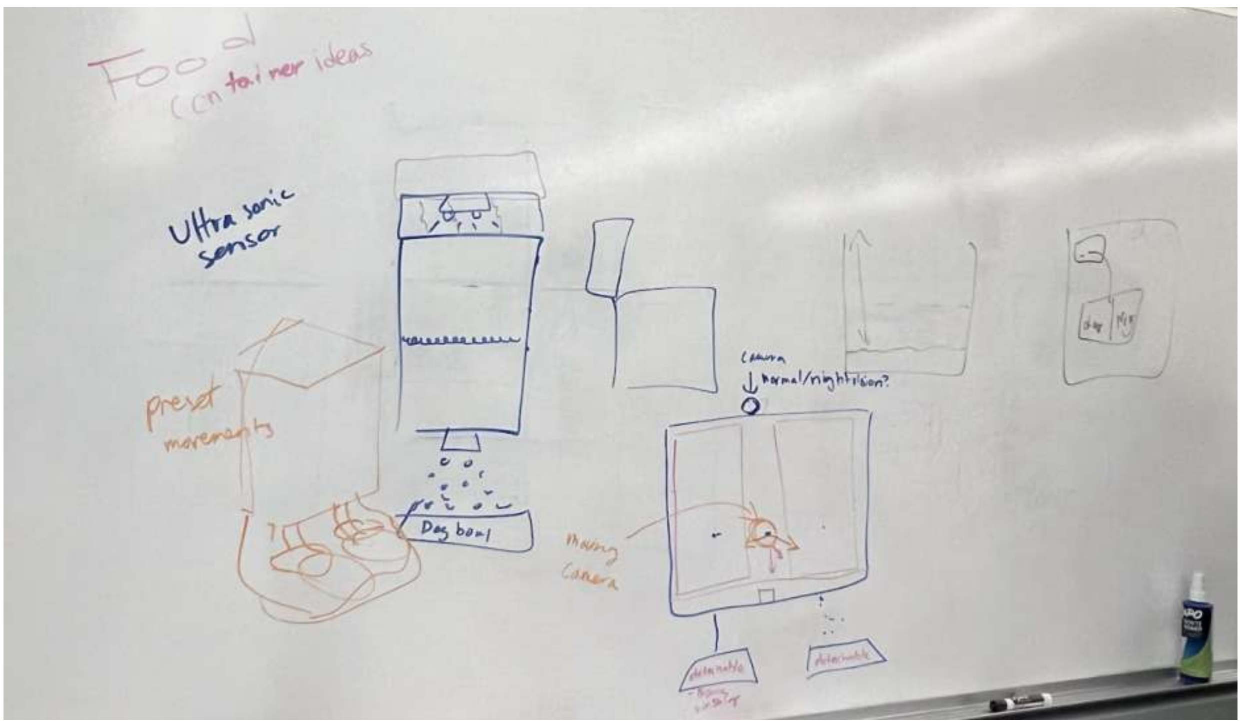
Appendix A: Problem Formulation

1. Conceptualizations & Brainstorming
   a. Mind Map



b. Sensor Methods





c. 6-3-5 Method: Each person write down three ideas within 5 minutes

Dawn #1 Idea: For the water level measuring stuff, we can possibly use a waterproof water float ball with a string attached to it and a lever. If the lever gets pulled then it indicates the water container needs a refill.

Dawn #2 Idea: For food level we can use a weight measurer at the bottom of the food container. ~~to the~~

Dawn #3 Idea: Add cameras to view the status of the animal during feeding time.

Kristina #1: We have a camera that points towards the pet to see how they are doing and eating their food. Then the camera points downwards to see the food bowl contents.

Kristina #2: Camera can have night vision capabilities and can be controlled through a mobile app.

Kristina #3: The automatic pet feeder is controlled through the mobile app.

Albert #1 Design idea: Add pet bowl s1 s2ee

#2 and use solid regulator to output water

#3 Use raspi pi to control cameras + everything else

Ginna #1 ~~install a foo~~ For the food bowl we can have a design that looks like ~~a~~ this where each portion gets covered

#2 Integrate slow feeder into food bowl

#3 Install a feature that stores the bag of treats / food

Sriharsha #1 Use an ultrasonic sensor to measure the water levels

#2 Use a camera to look at the food remaining in the bowl

#3 Use a motorized cap to cover the pet food

Mallika #1 mobile app to have profile for their pet to control and track how much food is dispensed and date

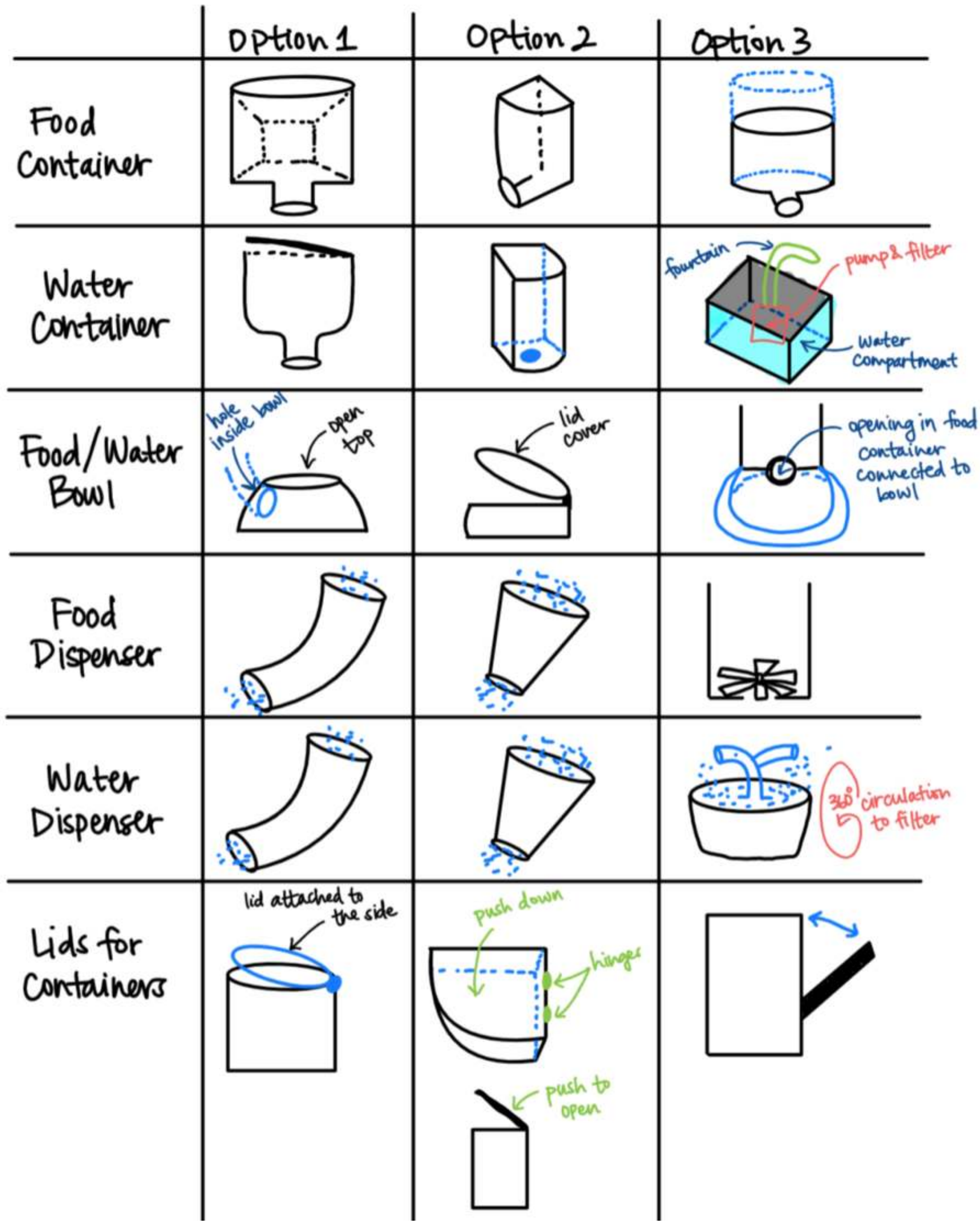
#2 retractable food bowls, so pet can only have food for certain amount of time

#3 A sound/signal that signals to the pet that the food was dispensed.

2. Decision Tables

Design	Cost	Wearbility/Robustness	Mobility	Reliability	Cleanability	Unweighted	Weighted
Design 1	5	4		2	4	15	53
Design 2	3	3		3	5	14	46
Design 3	2	3		5	3	13	45
Weight	5	3		4	2		
	Convenience	how easy it is to refill and clean the pet feeder					
error caused by an outside source	Wearability/Robustness	ability for product to remain stable and functional under variety of conditions such as the pet hitting/attacking the product					
error through time	Reliability	ability to consistently and accurately dispense the correct amount of food at scheduled times					
	Mobility	How easy it is to move around the device					
	Cleaning	How easy/hard it is to clean the inside of food/water containers/bowls (being able to take them out easily to clean them)					
	Cost	How much will this cost? 1 being little 5 being expensive					

3. Morphological Charts



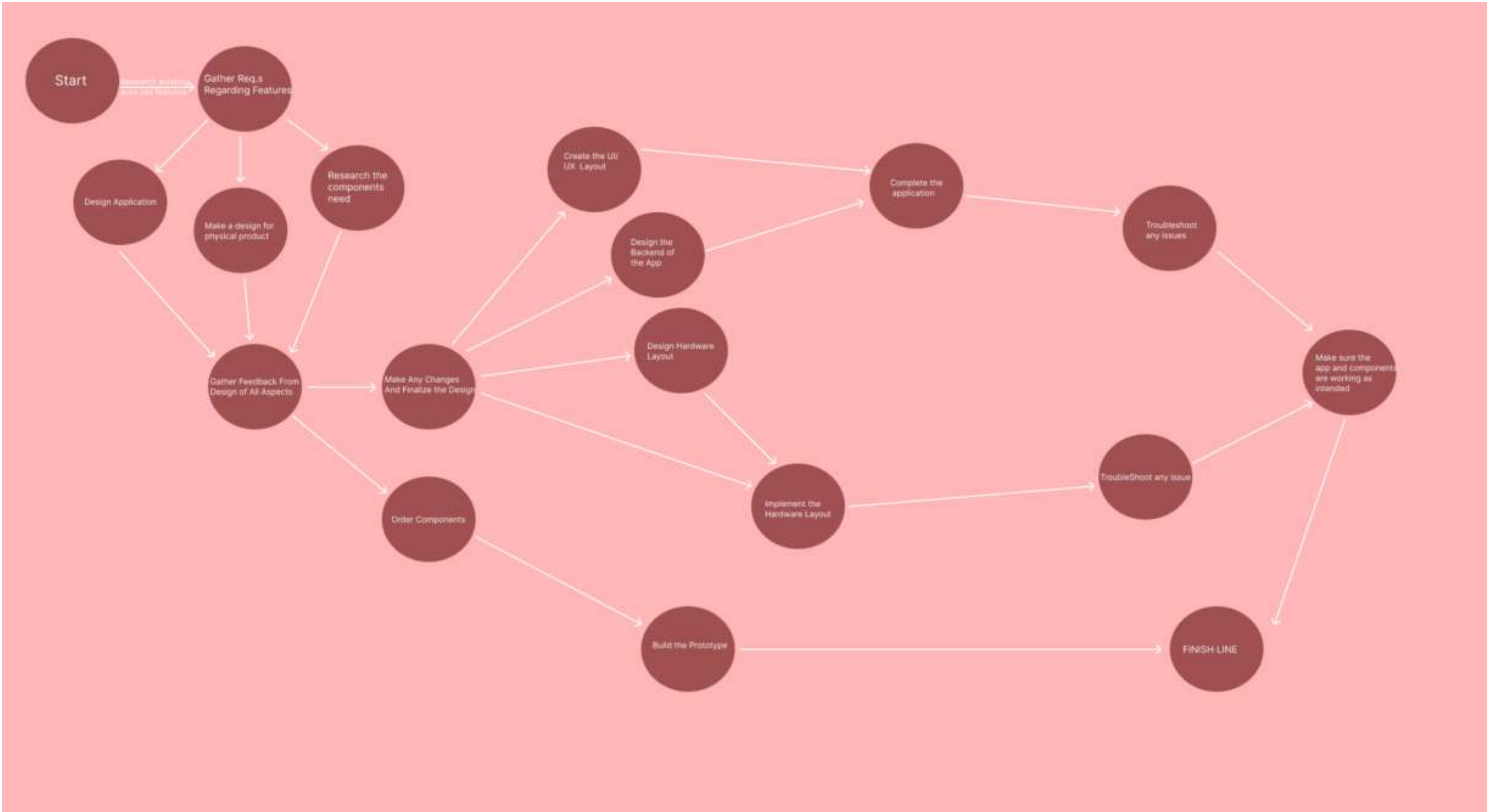
Appendix B: Planning

1. Gantt Chart

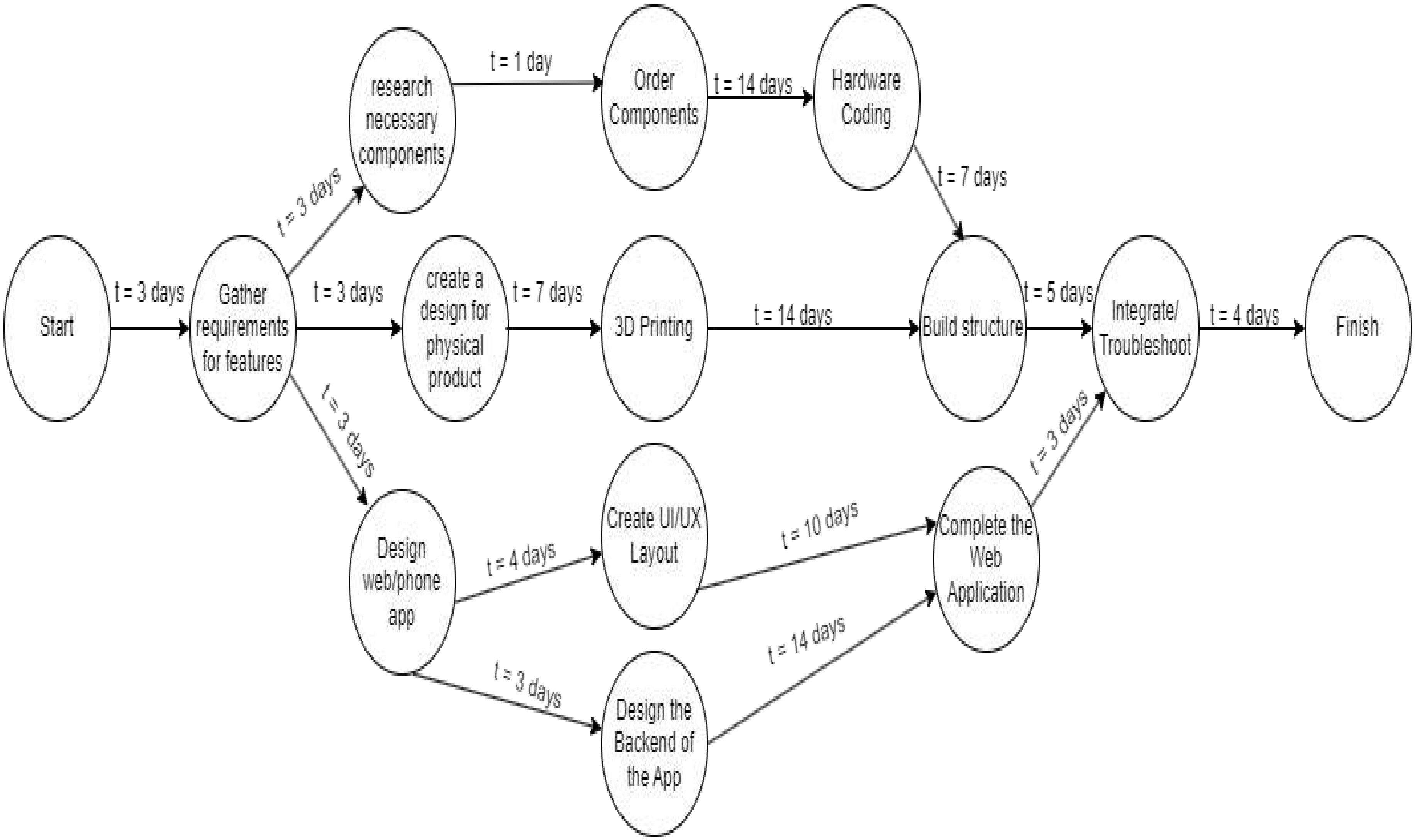
	A	B	C	D	E	G	H–L	M–Q	R–V	W–AA	AB–AF
1–2	TASK NO.	TASK TITLE	TASK OWNER	START DATE	DUE DATE	PROGRESS OF TASK	WEEK 6 (02/12 - 02/16) M T W T F	WEEK 7 (02/19 - 02/23) M T W T F	WEEK 8 (02/26 - 03/01) M T W T F	WEEK 9 (03/04 - 03/08) M T W T F	WEEK 10 (03/11 - 03/15) M T W T F
3	**1**	**Designing/Constructing**									
4	1.1	Finalize Design for Pet Feeder (Outside Cover)	Everyone	2/12/24	2/13/24	Completed					
5	1.2	Finalize Design for Water Dispenser	Leon, Ginna, Dawn, Albert	2/12/24	2/28/24	Completed					
6	1.3	Finalize Design for Food Dispenser	Leon, Ginna, Dawn, Albert	2/19/24	2/28/24	Completed					
7	1.4	Finalize Hardware Parts to use for Product (Create Bill of Materials)	Leon, Ginna, Dawn, Albert	2/19/24	2/28/24	Completed					
8	1.5	Food & Water Dispensing Prototype that tests whether our design choice for food/water works correctly	Leon, Ginna, Dawn, Albert	3/11/24	4/5/24	Completed					
9	1.6	Design a hardware MVP (Minimum Viable Product) which includes the hardware components using for final product	Leon, Ginna, Dawn, Albert	4/8/24	4/19/24	Completed					
10	**3**	**Hardware**									
11	3.1	Research Hardware Components	Leon, Ginna, Dawn, Albert	2/12/24	2/16/24	Completed					
12	3.2	Design Hardware Layout	Leon, Ginna, Dawn, Albert	2/26/24	3/1/24	Completed					
13	3.3	Purchase Components needed for Hardware	Leon, Ginna, Dawn, Albert	3/4/24	3/15/24	Completed					
14	3.4	Program Hardware Components (the chip & sensor)	Leon, Ginna, Dawn, Albert	4/8/24	5/3/24	Completed					
15	**2**	**Web App**									
16	2.1	Create a UI/UX Layout of the Web App	Sri, Mallika, Kris	2/12/24	3/1/24	Completed					
17	2.2	Discuss Backend & Frontend Frameworks	Sri, Mallika, Kris	2/19/24	3/1/24	Completed					
18	2.4	Backend Programming Part 1: Plan Code for HTTP Server & Communication with Hardware	Sri, Mallika, Kris	3/4/24	3/15/24	Completed					
19	2.3	Programming Part 1: Plan Code for Backend/Frontend	Sri, Mallika, Kris	3/4/24	3/15/24	Completed					
20	2.6	Backend Programming Part 2: Start Coding	Sri, Mallika, Kris	4/22/24	5/10/24	Completed					
21	2.5	Frontend Programming Part 3: Start Coding	Sri, Mallika, Kris	4/22/24	5/10/24	Completed					
22	**4**	**Integration of Pet Feeder & Web App**									
23	4.1	Discuss Integration of Feeder & App	Everyone	4/15/24	4/19/24	Completed					
24	4.2	Integrate Hardware & Web App	Everyone	4/22/24	5/3/24	Completed					
25	4.4	Troubleshoot (Design & Functional)	Everyone	5/6/24	5/24/24	Completed					
26	**5**	**Canvas Group Assignments**									
27	5.1	Draft Design Document	Everyone	2/26/24	3/8/24	Completed					
28	5.2	Interim Presentation	Everyone	3/4/24	3/15/24	Completed					

2. CPM & PERT Analysis

   a. Detailed Plan - Critical Path Analysis / CPM:
   https://www.figma.com/file/IXBO6dpw7tgYTe3TeWWaAN/Untitled?node-id=0%3A1&mode=dev



b. PERT Analysis:



3. Division of Labor

In addition to maintaining high efficiency and fostering good collaboration, our team is divided into two specialized groups: the software team and the hardware team.

The software team comprises Kristina, Sriharsha, and Mallika. Kristina serves as our cloud architect and full-stack developer, responsible for setting up the server in AWS. Sriharsha, and Mallika worked on Web and mobile app development and communication from frontend to the server .

The hardware team includes Da Eun, Ginna, and Leon. Da Eun is tasked with writing hardware code, as well as making and assembling parts of the product and designing the presentation slides. Ginna handles mechanical designs of the necessary components. Leon focuses on hardware coding and developing the communication protocols for the cloud servers.

And each of us is dedicated to testing and quality assurance, ensuring that our project meets the standard.

4. Collaboration

Our team was initially divided into two main groups: hardware and software. This division helped us focus on our specific areas of expertise and manage our tasks more efficiently. Within the hardware team, we did not use a typical code repository like GitHub. Instead, we utilized Google Drive to store and update the latest versions of our code. This approach allowed us to easily share and collaborate on documents and files, ensuring that everyone had access to the most current information and resources. The software team used Github to store and manage the code. The software team worked on a separate branch to complete separate tasks and everything was integrated and was deployed to the cloud in the end. We further subdivided responsibilities to streamline our workflow:

- One team member was dedicated to creating the CAD models and handling the 3D printing of parts.

- Another team member focused on writing the hardware communication code, which included interfacing with the server and programming the hardware components.

- The third member of the hardware team primarily concentrated on assembling the parts, while also programming hardware components.

- The fourth member of the team worked on the cloud deployment and backend development of the application, primarily configuring the nginx and ec2 instance.

- The fifth member of the team worked on the backend, connecting the firebase database to the app, creating new endpoints to handle data and also worked on the flutter mobile app.

- The sixth member of the team worked on the UI making the app look good and easier to navigate.

This clear division of labor allowed us to work concurrently on different aspects of the project, enhancing our overall productivity and ensuring that each component of our design was given adequate attention and expertise. By assigning tasks based on individual strengths and interests, we were able to collaborate effectively and achieve our project goals.

Appendix C: Test Plan & Results

1. Test Plan

Scope:

- Goal/Purpose: To evaluate the prototype's functionality, reliability, and user interface, focusing on food and water dispensing accuracy, camera feed clarity, and app communication efficiency, as well as the accuracy of water and food level sensors.
- Test Parameters:
    - Food and water dispensing accuracy (grams of food and milliliters of water dispensed vs. programmed amount)
    - Camera feed resolution and latency
    - App responsiveness and ease of use
    - Water level monitoring accuracy
    - Food level monitoring accuracy
- Expectations: The feeder should accurately dispense the correct amount of food and water within a 10% margin of error, provide a clear and stable live video feed with less than 2 seconds of latency, and offer a user-friendly app interface with no significant delays in commands. The water and food level monitoring sensors should accurately measure the respective levels.

Administrative Details

- Date and Location of Testing: March 2024, at UCSC BSOE Lab
- Client/Organization: CSE123A Winter 2024 Students
- Conductors: the team

Design of Each Test

- Food Dispensing Accuracy Test:
    - Testing Method: Quantitative measurement
    - Significance: Ensures pets receive the correct amounts of food
    - Apparatus: Level Meter on the Container, Weighing Machine
    - Variables: Independent - programmed dispensing amount; Dependent - actual dispensed amount of food
    - Factors Considered: Quantity of food
- Camera Feed Quality Test:
    - Testing Method: Qualitative assessment and latency measurement
    - Significance: Ensures pet owners can clearly monitor their pets and water/food levels
    - Apparatus: Raspberry Pi and IOS application
    - Variables: Independent - camera resolution settings; Dependent - image clarity and feed latency
    - Factors Considered: resolution and latency
- Mobile App Usability Test:
    - Testing Method: User experience survey and response time measurement
    - Significance: Ensures ease of use and efficiency in managing both food and water features
    - Apparatus: An iPhone
    - Variables: Independent - user command input; Dependent - app response time and user satisfaction
    - Factors Considered: Response time, Connectivity issues
- Water Level Sensor Test:
    - Testing Method: Quantitative measurement
    - Significance: Ensures accurate measurement of water levels for timely refilling
    - Apparatus: Liquid level switch sensor
    - Variables: Independent - water level; Dependent - sensor output
    - Factors Considered: Sensory accuracy, Environmental factors
- Food Level Sensor Test:
    - Testing Method: Quantitative measurement
    - Significance: Ensures accurate measurement of food levels for timely refilling
    - Apparatus: Ultrasonic sensor
    - Variables: Independent - food level; Dependent - sensor output
    - Factors Considered: Sensor accuracy, Environmental factors

Sampling Procedure:

- Sampling Method: Random selection from prototype batch
- Number of Samples: 10 trials for each test to ensure statistical relevance

Detailed Procedure

1. Calibrate and prepare all necessary equipment prior to each test.
2. Perform a pre-test check to ensure the feeder and water dispenser's components are functioning correctly.
3. Execute each test according to a consistent, step-by-step procedure, maintaining controlled conditions.
4. Digitally record all quantitative data and collect user feedback through surveys.

Safety Precautions

- Adhere to protective gear requirements during all tests.
- Ensure safe setup of electronic and water components to prevent accidents.

Data Collection Method

- Utilize digital tools for recording quantitative data and capturing user feedback electronically for comprehensive analysis.

Observation of External Factors

- The product should be placed at room temperature on a leveled surface, away from any hazardous chemicals.
- The product should be connected to a reliable power source.

2. Results
    1. Food Dispensing Accuracy Test

       Objective: Ensure pets receive the correct amounts of food.

       Methodology: Quantitative measurement using a level meter on the container and a weighing machine.

       Test Parameters:

       - Independent Variable: Programmed dispensing amount
       - Dependent Variable: Actual dispensed amount of food
       - Factors Considered: Quantity of food

       Results Summary:

- Trial 1:
  - Small Bowl: Programmed: 226.8g food | Dispensed: 220.1g food | Error Margin: 2.95% food
  - Large Bowl: Programmed: 453.6g food | Dispensed: 430.2g food | Error Margin: 5.16% food
- Trial 2:
  - Small Bowl: Programmed: 226.8g food | Dispensed: 225.0g food | Error Margin: 0.79% food
  - Large Bowl: Programmed: 453.6g food | Dispensed: 450.0g food | Error Margin: 0.79% food
- Trial 3:
  - Small Bowl: Programmed: 226.8g food | Dispensed: 230.0g food | Error Margin: -1.41% food
  - Large Bowl: Programmed: 453.6g food | Dispensed: 440.0g food | Error Margin: 3.00% food
- Trial 4:
  - Small Bowl: Programmed: 226.8g food | Dispensed: 222.5g food | Error Margin: 1.90% food
  - Large Bowl: Programmed: 453.6g food | Dispensed: 435.0g food | Error Margin: 4.10% food
- Trial 5:
  - Small Bowl: Programmed: 226.8g food | Dispensed: 221.0g food | Error Margin: 2.56% food
  - Large Bowl: Programmed: 453.6g food | Dispensed: 425.0g food | Error Margin: 6.31% food
- Trial 6:
  - Small Bowl: Programmed: 226.8g food | Dispensed: 224.8g food | Error Margin: 0.88% food
  - Large Bowl: Programmed: 453.6g food | Dispensed: 448.5g food | Error Margin: 1.12% food
- Trial 7:
  - Small Bowl: Programmed: 226.8g food | Dispensed: 228.5g food | Error Margin: -0.75% food
  - Large Bowl: Programmed: 453.6g food | Dispensed: 445.2g food | Error Margin: 1.85% food
- Trial 8:

    - Small Bowl: Programmed: 226.8g food | Dispensed: 223.0g food | Error Margin: 1.68% food
    - Large Bowl: Programmed: 453.6g food | Dispensed: 440.8g food | Error Margin: 2.82% food
- Trial 9:
    - Small Bowl: Programmed: 226.8g food | Dispensed: 226.0g food | Error Margin: 0.35% food
    - Large Bowl: Programmed: 453.6g food | Dispensed: 452.0g food | Error Margin: 0.35% food
- Trial 10:
    - Small Bowl: Programmed: 226.8g food | Dispensed: 225.5g food | Error Margin: 0.57% food
    - Large Bowl: Programmed: 453.6g food | Dispensed: 449.8g food | Error Margin: 0.84% food

★ Average Error Margin: Food: Small Bowl: 0.952% | Large Bowl: 2.634%

Conclusion:

The food dispensing accuracy test aimed to ensure pets receive the correct amounts of food by quantitatively measuring the programmed and dispensed amounts using a level meter and weighing machine. Across ten trials for both small and large bowls, the average error margins were calculated to be 0.952% for the small bowl and 2.634% for the large bowl.

Overall, the results demonstrate a high level of accuracy in food dispensing, with the average error margins falling within acceptable ranges. The slight variations observed may be attributed to factors such as minor inconsistencies in the dispensing mechanism or variations in environmental conditions during testing. However, these deviations are minimal and do not significantly impact the overall effectiveness of the food dispensing system.

Based on these findings, it can be concluded that the food dispensing system reliably delivers the programmed amounts of food to pets, ensuring their nutritional needs are met accurately. Further optimization efforts could focus on reducing the error margins even further, although the current levels are already satisfactory for practical use.

These results validate the efficacy of the food dispensing system in maintaining precise portion control, contributing to the overall health and well-being of pets in their daily feeding routines.

2. Camera Feed Quality Test

Objective: To evaluate the clarity and latency of the camera feed for monitoring pets and water/food levels.

Methodology: Qualitative assessment and latency measurement using a Raspberry Pi camera and website or phone.

Test Parameters:

- Independent Variable: Camera resolution settings
- Dependent Variable: Image clarity and feed latency
- Factors Considered: Resolution and latency

Results Summary:

- Resolution Setting: 1080p | Clarity: High | Latency: 10-20 seconds (This latency remains consistent through multiple camera testings)
- ★ Average Latency: 10-20 seconds.

Conclusion:

The camera consistently provided high clarity images at a resolution of 1080p. However, the observed latency of 10 to 20 seconds may hinder real-time monitoring. Addressing latency concerns is crucial for improving the usability of the camera feed. Strategies such as optimizing network connections and adjusting software settings should be explored to ensure timely access to monitoring pets and maintaining water and food levels.

3. Website/Mobile App Usability Test:

Objective: To ensure ease of use and efficiency in managing both food and water features through the mobile app.

Methodology: User experience survey and response time measurement using a website or phone.

Test Parameters:

- Independent Variable: User command input through the website/mobile app.
- Dependent Variable: App response time and user satisfaction.
- Factors Considered: Response time, connectivity issues, and overall usability of the mobile app interface.

Results Summary:

- User 1: Response Time: 10 seconds | Satisfaction: 9/10
- User 2: Response Time: 11 seconds | Satisfaction: 8/10
- User 3: Response Time: 10 seconds | Satisfaction: 7/10
- User 4: Response Time: 19 seconds | Satisfaction: 4/10

★ Average Response Time: 12.5 seconds
★ Average Satisfaction: 7/10

Conclusion:

The average response time for user commands was approximately 12.5 seconds, indicating a moderate level of efficiency in the app's responsiveness. However, there was variability among user satisfaction scores, with an average satisfaction rating of 7 out of 10. While some users expressed high satisfaction with the app's performance, others reported lower levels of satisfaction, suggesting room for improvement in certain aspects of usability. Further investigation into the specific areas of dissatisfaction, such as user interface design or functionality, is recommended to enhance overall user experience and satisfaction with the app.

4. Water Level Accuracy Test

Objective: Ensure accurate measurement of water levels using a liquid level switch

sensor.

Test Parameters:

- Independent Variable: Actual water level
- Dependent Variable: Sensor reading (low/high)
- Factors Considered: Sensor calibration, water volume

Results Summary:

- Trial 1:
  - Actual Level: Low | Sensor Reading: Low
- Trial 2:
  - Actual Level: High | Sensor Reading: High
- Trial 3:
  - Actual Level: Low | Sensor Reading: Low
- Trial 4:
  - Actual Level: High | Sensor Reading: High
- Trial 5:
  - Actual Level: Low | Sensor Reading: Low
- Trial 6:
  - Actual Level: High | Sensor Reading: High
- Trial 7:
  - Actual Level: Low | Sensor Reading: Low
- Trial 8:
  - Actual Level: High | Sensor Reading: High
- Trial 9:
  - Actual Level: Low | Sensor Reading: Low
- Trial 10:
  - Actual Level: High | Sensor Reading: High

Conclusion:

The water level accuracy test aimed to ensure accurate measurement of water levels using a liquid level switch sensor. Across ten trials, the sensor consistently provided accurate readings of water levels (low or high), indicating its reliability. This simple measurement approach effectively ensures that the water level status is correctly detected, contributing to maintaining appropriate water levels for pets.

5. Food Level Accuracy Test

Objective: Ensure accurate measurement of food levels using an ultrasonic sensor.

Methodology: Simple measurement of food level status (low/high) using a level sensor.

Test Parameters:

- Independent Variable: Actual food level
- Dependent Variable: Sensor reading (low/high)
- Factors Considered: Sensor calibration, food volume

Results Summary:

- Trial 1:
  - Actual Level: Low | Sensor Reading: Low
- Trial 2:
  - Actual Level: High | Sensor Reading: High
- Trial 3:
  - Actual Level: Low | Sensor Reading: Low
- Trial 4:
  - Actual Level: High | Sensor Reading: High
- Trial 5:
  - Actual Level: Low | Sensor Reading: Low
- Trial 6:
  - Actual Level: High | Sensor Reading: High
- Trial 7:
  - Actual Level: Low | Sensor Reading: Low
- Trial 8:
  - Actual Level: High | Sensor Reading: High
- Trial 9:
  - Actual Level: Low | Sensor Reading: Low
- Trial 10:
  - Actual Level: High | Sensor Reading: High

Conclusion:

The food level accuracy test aimed to ensure accurate measurement of food levels using a level sensor. Across ten trials, the sensor consistently provided accurate readings of food levels (low or high), indicating its reliability. This simple measurement approach effectively ensures that the food level status is correctly detected, contributing to maintaining appropriate food levels for pets.

Observations and External Factors

The product was placed indoors to ensure consistent test conditions. No hazardous chemicals were present, and a reliable power source was used throughout testing.

Safety Precautions

All tests were conducted with regular clothes. Safe setup of electronic and water components was ensured to prevent accidents.

Final Analysis and Recommendations

Analysis:

The conducted tests encompassed rigorous evaluations of the prototype's functionality, reliability, and user interface across various aspects such as food and water dispensing accuracy, camera feed clarity, and app communication efficiency.

In the Food Dispensing Accuracy Test, the results consistently demonstrated a high level of accuracy, with average error margins falling within acceptable ranges for both small and large bowls. The system reliably delivered programmed amounts of food, ensuring pets receive precise nutrition.

Similarly, the Camera Feed Quality Test revealed satisfactory performance in providing high clarity images at a resolution of 1080p. However, the observed latency of 10 to 20 seconds may potentially hinder real-time monitoring, necessitating further optimization efforts to improve usability.

The Website/Mobile App Usability Test indicated a moderate level of efficiency in app responsiveness, with an average response time of approximately 12.5 seconds. However, there was variability in user satisfaction scores, suggesting potential areas for enhancement in certain aspects of usability and functionality.

The Water Level Accuracy Test confirmed that the water level sensor reliably provided accurate readings (low or high) across ten trials, ensuring the maintenance of appropriate water levels for pets. Similarly, the Food Level Accuracy Test showed that the food level sensor consistently provided accurate readings (low or high) across ten trials, effectively maintaining appropriate food levels for pets.

Recommendations:

To improve performance, it is crucial to address the camera feed's latency by optimizing network connections and adjusting software settings for better real-time monitoring. Enhancing user satisfaction can be achieved by addressing areas of dissatisfaction, such as user interface design or functionality, based on user feedback. Continuous improvement through regular testing, user feedback collection, and iteration cycles will ensure the product meets evolving user needs. Providing comprehensive user education materials, troubleshooting guides, and responsive customer support will assist users in navigating challenges. Regular calibration of water and food level sensors will maintain accuracy and reliability. Staying updated with emerging trends and technological advancements is vital for future-proofing the product and ensuring its market relevance. Implementing these recommendations will enhance the pet feeding system's accuracy, reliability, and user-friendliness, benefiting the overall health and well-being of pets.

Appendix D: Review

1. Da Eun Lee: Overall, team collaboration was excellent. Whenever someone needed help, others stepped in to lighten the load. Communication was frequent and effective, both in person and through Discord. I stayed on top of my tasks and learned a lot about managing my team and helping us all stay on track. I also gained significant knowledge in hardware coding, deepening my understanding of how hardware is controlled to perform specific tasks. Our integration of hardware and software components was successful, and it was fascinating to see them communicate seamlessly. I thoroughly enjoyed creating and designing the presentation slides. However, next time, we should plan further ahead to allow more time for higher-quality completion of tasks. Additionally, I would prefer to laser-cut a box for the food container instead of using a fruit box from Costco.
2. Sriharsha Maddala: Our Team collaboration was excellent throughout the course of this project. Whenever we had an issue with things not working as planned, all of the teammates responded really quickly and immediately made time to sit down and resolve the issue. I believe that this quick responsiveness among us is what has helped us in resolving all the issues that we had really fast and deliver the project on time. During the course of this project I have also learned a lot of technologies that I have never used in the past. The software team especially had difficulty with the cloud deployment initially and we had to do a lot of research to get to know what nginx was and learn how to use it. Another new technology that I learned how to use was Google

Firebase database which was challenging but was really fun at the same time. It was also really interesting to learn how data was transferred using the API endpoints from the database to the app and vice-versa. Overall I have personally learned a lot about new technologies that I have never worked with before. One thing that I think I would do differently if I could is to use a different and easier to use technology for cloud because our team had a lot of difficulty navigating through the Amazon EC2 instance for the app deployment.

3. Kristina Fout: Reflecting on our project journey, I am proud of the progress and achievements we have made as a team. Our collaboration and commitment to excellence allowed us to overcome numerous challenges, from slight technical mishaps to tight deadlines. I learned about effective communication, problem-solving, and the importance of thorough planning. I enjoyed coding the backend of the application in Python and Javascript, and then understanding how to connect these concepts through API Endpoints. Moreover, I am glad to have gained experience with cloud deployment with AWS and Nginx where I understand the importance of secure connections with HTTPS. Working on the automatic pet feeder project was a rewarding experience, not only in terms of technical skills but also in building strong team dynamics. I am grateful for the dedication and support of each team member, which was crucial in bringing our vision to life. What could have been done differently would be to allocate more time for adding additional software features such as having a light or dark mode, then getting the device status of the hardware to be fully functional.
4. Ginna Khang: Throughout the two quarters collaborating with my team, I observed that our communication remained consistently effective. Initially, during the winter quarter, our communication skills were somewhat challenging. However, after gathering the team and creating a Gantt chart, we successfully managed our timelines and completed our tasks punctually. My responsibility involved overseeing the mechanical design of our pet feeder, which I found enjoyable due to my prior experience with 3D modeling and printing. Looking back, one area for improvement could have been my initial request for additional equipment, such as a better 3D printer, at the beginning of the quarter. It would have been beneficial to have more accessible resources like additional 3D printers and essential lab equipment readily available in our workspace. It became inconvenient having to frequently visit BELS for basic materials such as duct tape, measuring tape, and super glue. Moreover, the lab only had two 3D printers that were often unreliable, leading us to opt for Slugworks where we could use more dependable equipment. However, because Slugworks only

opened during the hours of 12-9pm, it took forever to 3D print. Therefore, another thing I would like to have done better was complete the CAD design by week 3 and start 3D printing then.

5. Leon Feng: Our team collaboration was outstanding. Everyone supports each other when needed. We maintained frequent and effective communication all the time, ensuring that our project stayed on track. This collaborative spirit was a key factor in the successful integration of our hardware and software components. I am glad to have worked with such a talented and cooperative team, which not only enhanced the quality of our project but also improved my team collaboration skills. This project also marked my first experience with hardware programming, and it was a significant learning journey. I learned how to integrate the codes from different components into a unified system and how to establish connections between the hardware and server for data transmission. These skills were crucial for ensuring the seamless operation of our project. For any improvement, I would spend more time developing new hardware features, such as motion detection and live feed functionality for the camera. Better early planning could help achieve these advanced features. This underscored the importance of detailed upfront planning and effective time management.
6. Mallika Gupte: Something that went well was our team's exceptional collaboration. We coordinated on various components, each member contributing effectively and providing support wherever needed. This teamwork helped for a smooth progression and success of our project. Additionally, I learned new concepts such as HTTPS requests, API calls, and code testing. This helped deepen my understanding of web application development. The effective integration of front-end and back-end was also something that went well, which allowed our team to deliver a seamless user experience. We also implemented responsive design, ensuring that the web application performed well across different devices and screen sizes. In terms of what could have been done differently, there was room for earlier and more rigorous user testing. Engaging potential users during the earlier phases could have given us some valuable feedback that might have influenced design decisions or functionality adjustments.