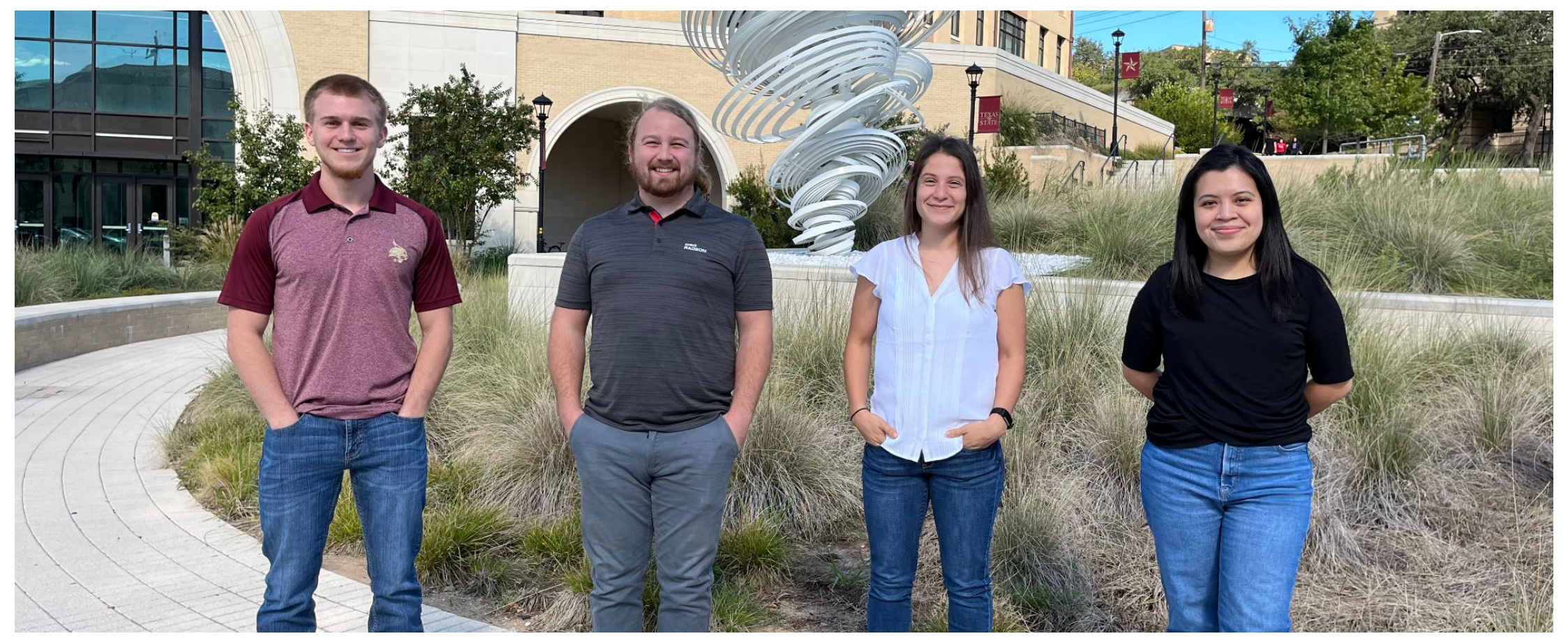
TEXAS STATE UNIVERSITY®

E2.04 - Automated Pet Feeder

PM: Nicolas Briggs
Rosemary Perez
Victoria Ticheli
Barton Maupin



Nicolas Briggs   Barton Maupin   Victoria Ticheli   Rosemary Perez

Requirements

Must be able to:

- Distinguish between at least 2 pets and open appropriately
- Support at least 2 pets approaching at once
- Alert user if food level drops below minimum level
- Be user-programmable but does not have to be user-interactive
- Run on backup battery with minimum life of 12 hours

Motivation

- To allow busy people to keep up with a feeding schedule for their pets
- To help prevent pet obesity by portioning food
- To allow for feeding more than 2 pets unlike many feeders on the market

Actual Product

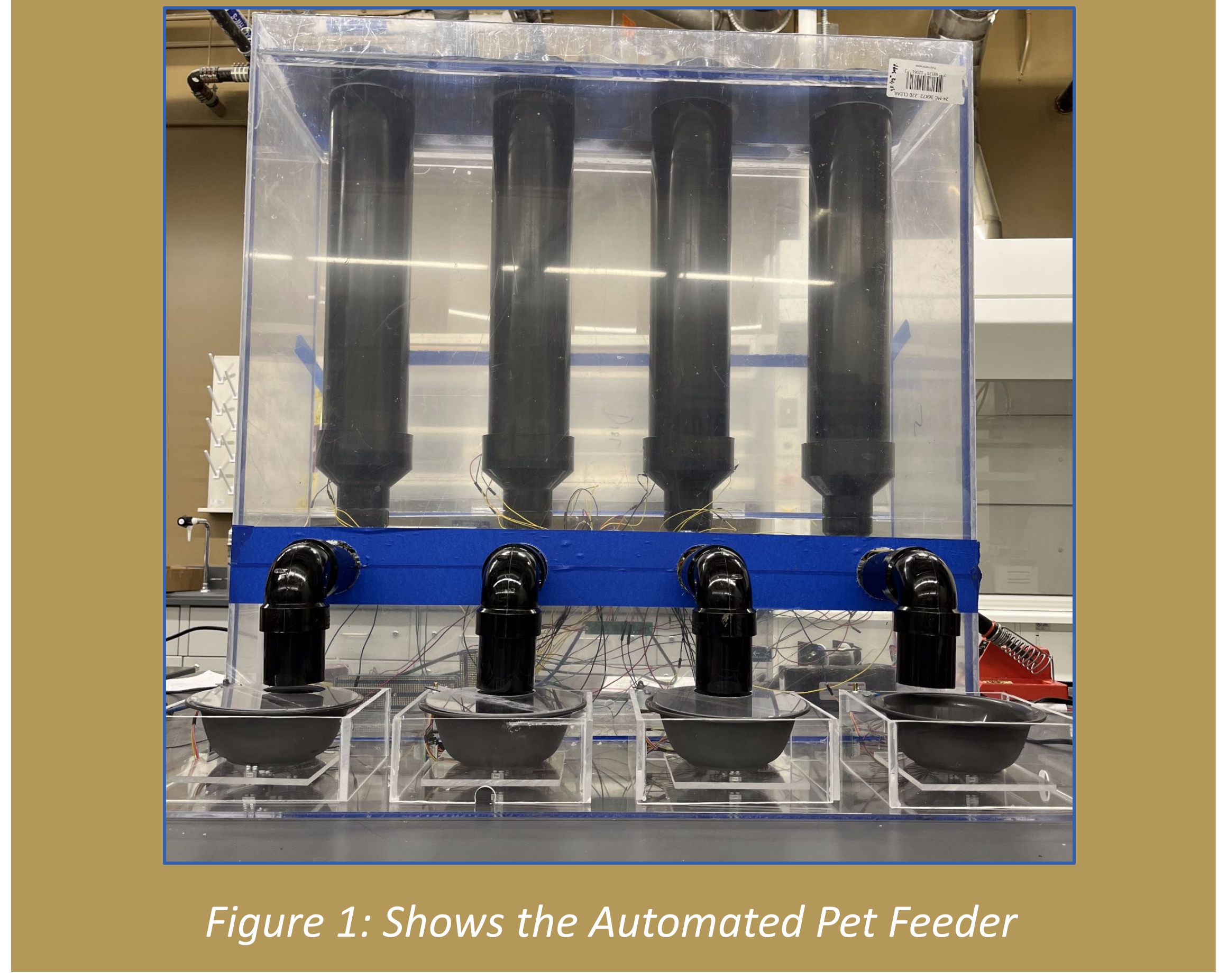


Figure 1: Shows the Automated Pet Feeder

Design

Our project is an indoor automated pet feeder that will feed common household pets their food with specified controlled portions. The feeder will differentiate multiple pets in a single household. It will also record the time of each feeding along with current malfunctions and notify the user through SMS texts and emails.

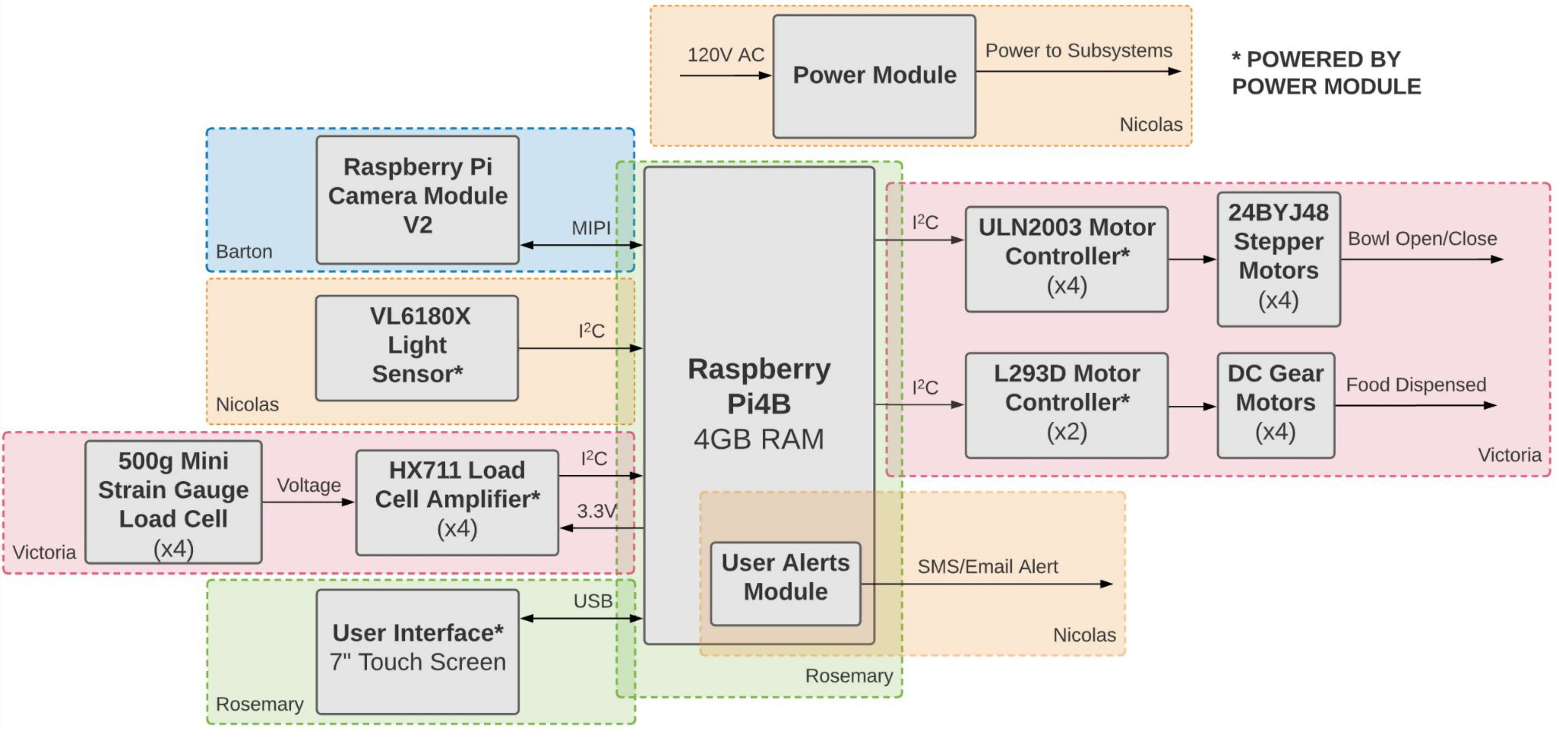


Figure 2: System's High Level Block Diagram

Testing

Requirement	Measured Results	Outcome
Correctly distinguish between four different types of animals	Not able to test	FAIL
Dispense the user specified portion for a verified pet	Specified Portion = 170.0 g Measured Portion = 176.3 g 96.29% match	PASS, within 90% of specified portion
The backup battery must last a minimum of 12 hours	With 8 feeding cycles, the battery lasted 12 hours and 40 minutes	PASS
If pet is done eating, or if another pet approaches the feeder, the lid shall close.	Not able to test	FAIL
Malfunctions and food level replenishment alerts will be sent via SMS/text message to the user's phone.	Testing at Ingram Hall, each SMS alert was properly sent out within 7 seconds	PASS
System must be user-programmable but it does not have to be user-interactive	User is able to make change to the pet information saved on file which is seen by the script	PASS

Figure 3: Overall test results of the system

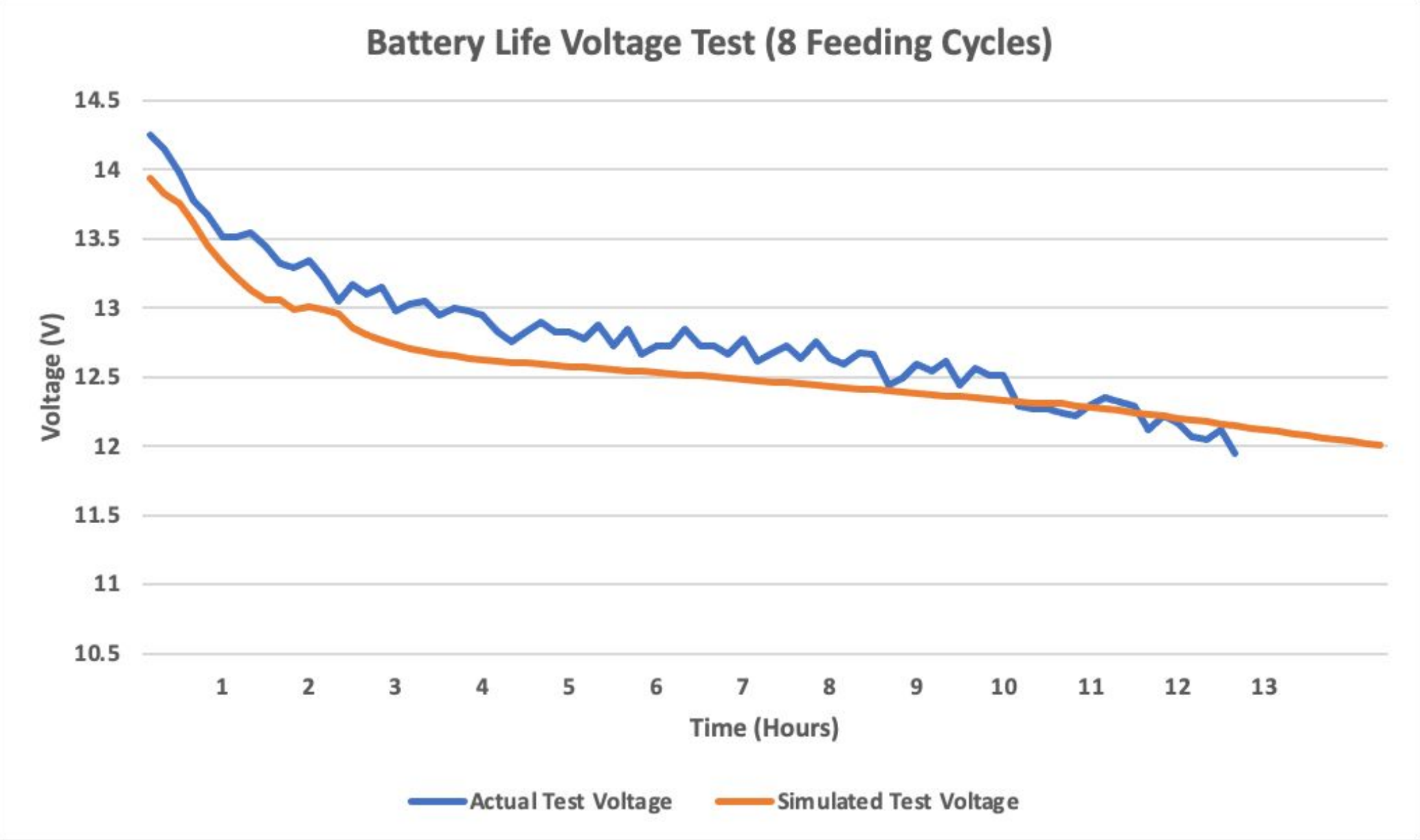


Figure 4: System's battery performance

WEIGHT SENSING SUBSYSTEM TEST

REQUIREMENT	PROCEDURE	OVERALL RESULT
Measure 256 grams with a 90% match, ± 0.1 grams	Use kitchen scale to verify weight of 256 grams, then place this on the weight sensor after calibration. Perform test 3 times. Expected result should measure 256 grams.	PASS - average of 270.5 grams, a match of 94.3%
Measure amount of food a pet has eaten	With 128 grams of food in bowl, measure weight. Take out 56 grams of food to simulate pet eating. Measure new weight of bowl. Perform test 3 times. Expected result should measure 56 grams.	PASS - average of 52.2 grams, a match of 93.2%

Figure 5: Detailed test results of the Weight Subsystem

Camera Recognition

- Trained Convolutional Neural Network to target pets
- TensorFlow, Keras, and OpenCV used to build CNN

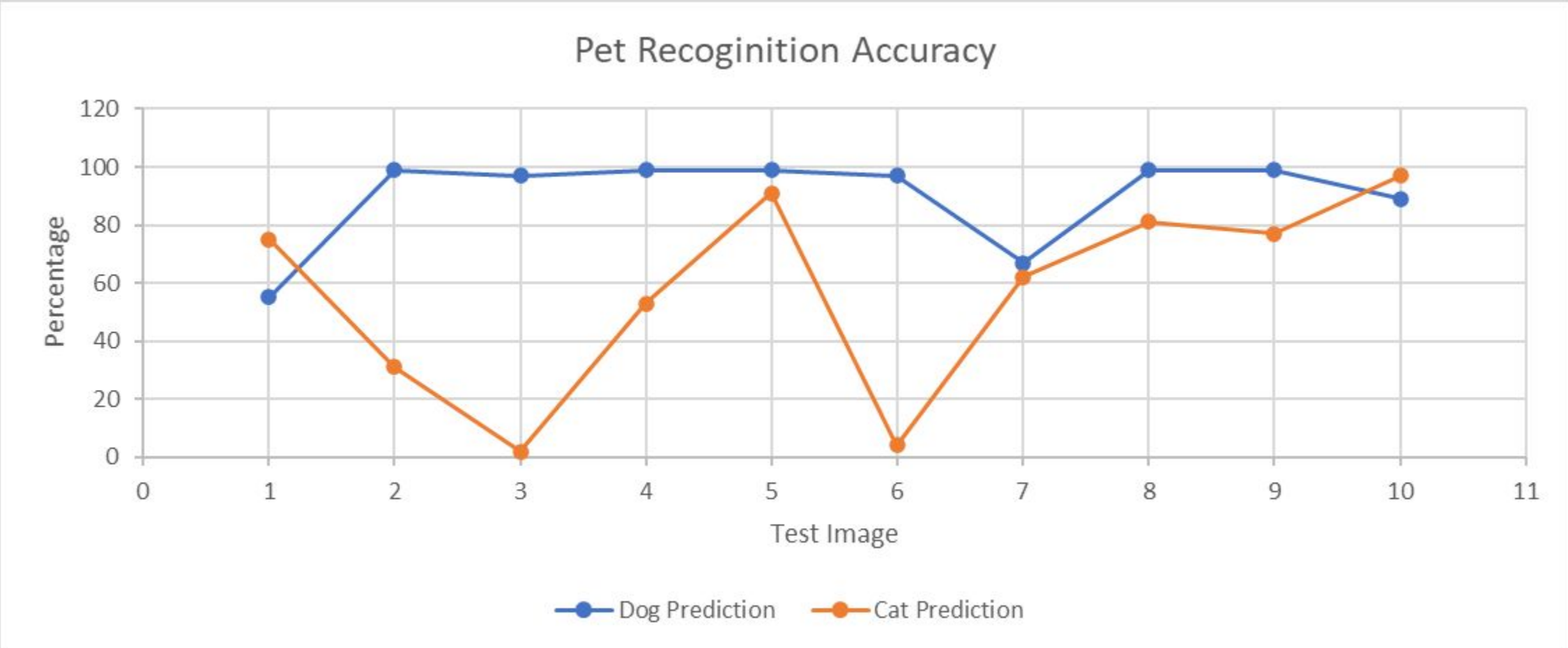


Figure 6: Test results from showing example pictures and accuracy

Overall Flow

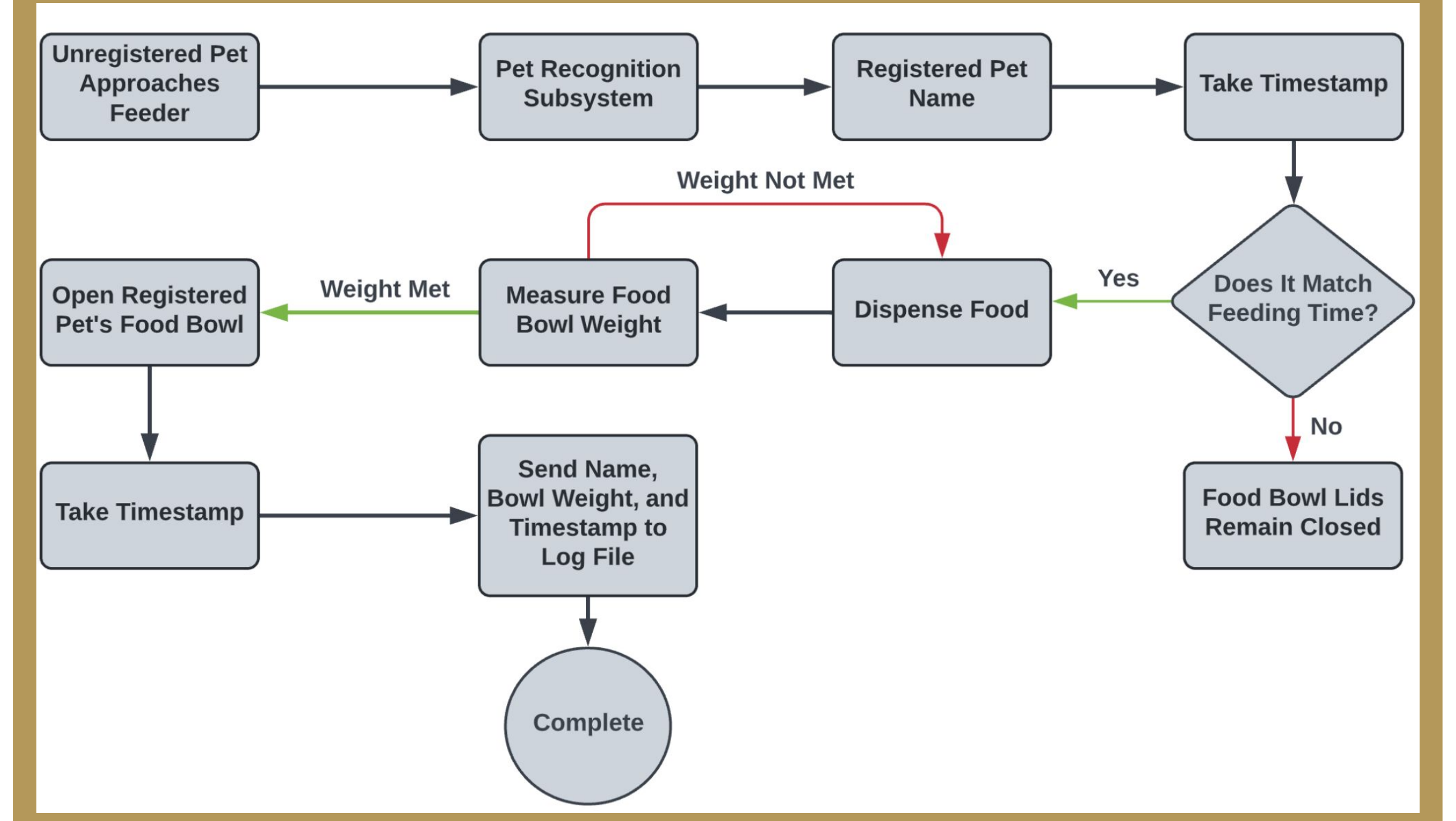

Unregistered Pet Approaches Feeder
Pet Recognition Subsystem
Registered Pet Name
Take Timestamp
Weight Not Met
Does It Match Feeding Time?
Yes
No
Dispense Food
Measure Food Bowl Weight
Weight Met
Open Registered Pet's Food Bowl
Food Bowl Lids Remain Closed
Take Timestamp
Send Name, Bowl Weight, and Timestamp to Log File
Complete

Figure 7: High Level Software flow of the Host Scheduler

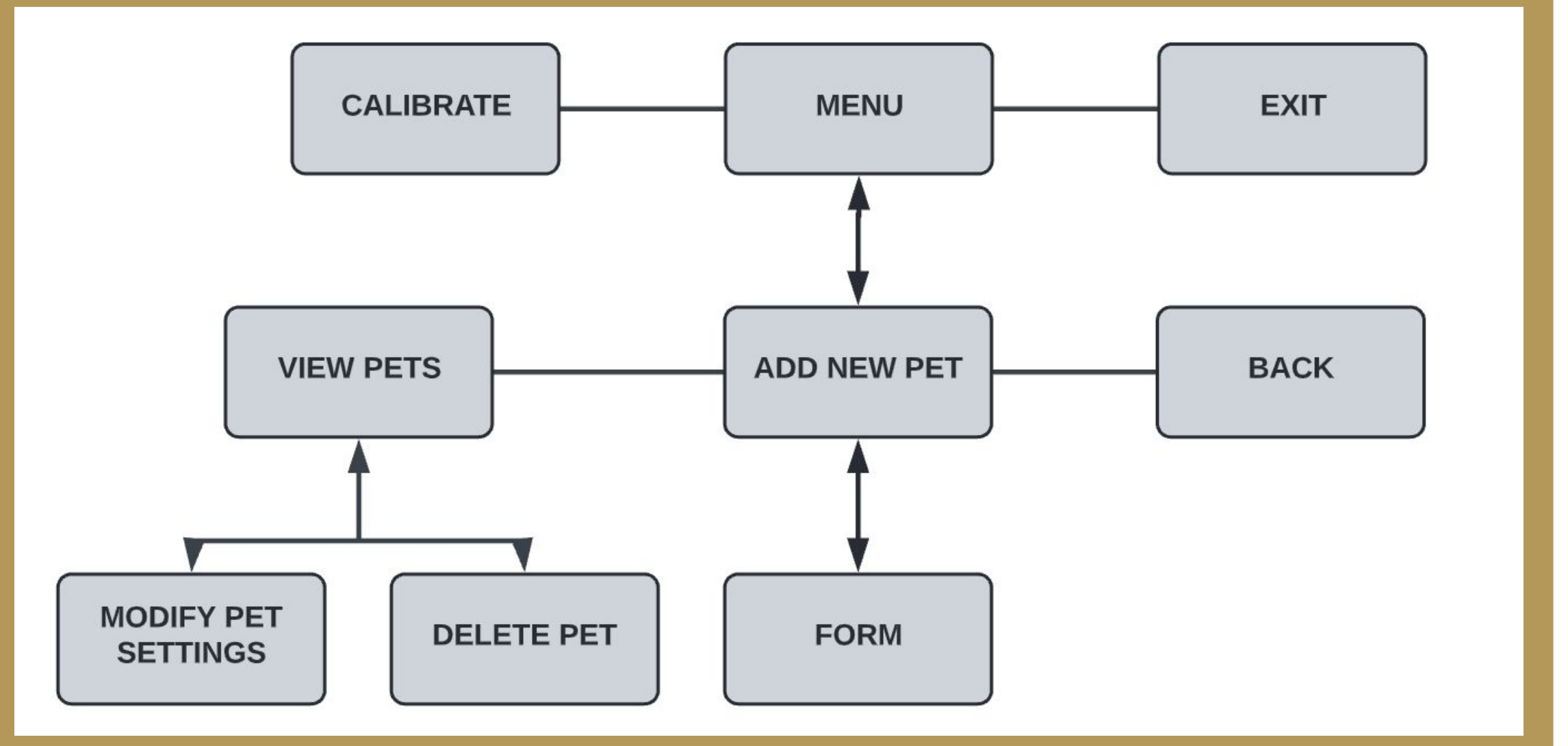


Figure 8: High Level Software flow of the User Interface

Acknowledgements

- Instructor: Lee Hinkle
- Faculty Advisor: Mark Welker
- Sponsor: Diane Welker