CAT FOOD DROPPER

ECE 445 Design Document

Hailey Harin Cho

Lexie Kolb

Michael Park

Group 1

TA: Dean Biskup

September 29th, 2021

Table of Contents

1. Introduction

1.1 Problem

Many cats have the innate ability to portion out their meals throughout the day. However, there are other cats who are either lacking this ability or too stressed by competition with other cats to properly portion their meals and as a result force down their meals too quickly [1][10]. Cats who eat too fast regurgitate their meals as their brains interpret their stomach expanding too quickly as a problem; this causes dehydration and malnutrition [10]. This problem can be solved by having the cat owner manually portion out meals and feed the cat slowly throughout the day, sometimes up to 7 times a day [2]. Ultimately this is neither convenient nor practical for owners.

While there are several automatic cat food dispensers on the market today, none of them actively controls a cat's eating speed. Instead, they simply dispense at a pre-arranged time regardless of whether food is present in the bowl, or if the amount dispensed enables the cat to eat too much too quickly.

1.2 Solution

We decided to make a food dispenser that "spoon-feeds" the cat its portions throughout the day. First, the owner will set a max rate at which the cat is allowed to eat (e.g. 1g/min). Then, using a timer, the product will demarcate a one minute dispensing period wherein a scale sensor underneath the bowl will weigh the amount of food left, and will replenish the difference between the max amount per period and the amount left in the bowl. This ensures that the highest possible eating rate in each period can not exceed the max rate as there physically isn't enough food in the bowl within one period for that to be possible. Depending on the effectiveness, the owner may choose to raise or lower the max rate after consulting with a veterinary professional.

Additionally, we will use an RFID transceiver that can distinguish individual cats via an RFID receiver affixed to the cats collars. This feature will allow the owner to buy one bowl to serve multiple cats and prevent cats from stealing food from one another to overeat.

Finally, the owner can use the user software application to configure each cat's eating habits. We will provide vet-recommendations for serving size per portion based on gender and age, or the amount can be set manually by an informed user. Additionally, data such as amount eaten and eating rate will be persistently stored so that owners can talk about their cats' eating habits during monthly visits to the vet.

1.3 Visual Aid

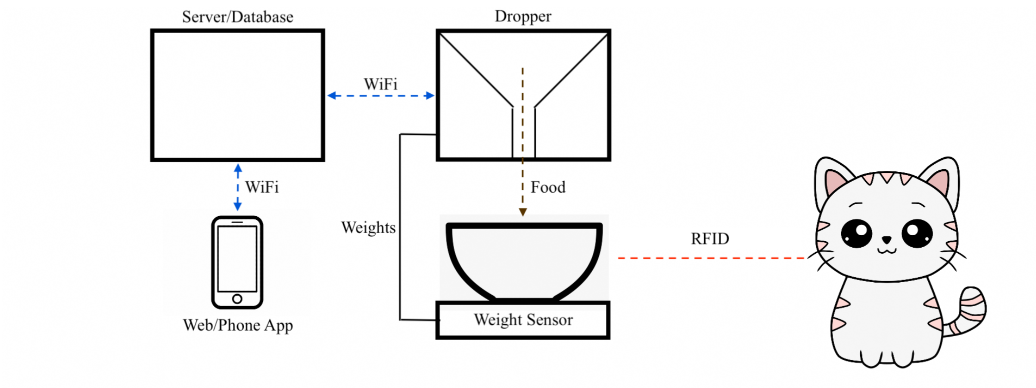


Figure 1. Cat Food Dropper Visual Aid

1.4 High-level Requirements

- ☐ The system must enforce a max eating speed (in g/min) set by the user by adjusting the amount of food dropped within every one minute dispensing period.
- ☐ The dropper must distinguish between multiple cats by using an RFID transceiver to detect RFID receiver strips adhered to the cats' collars. Then, it must customize the system's max eating rate, portion sizes, and portion times to the respective cat.
- ☐ The software component must visualize data about the cat's eating habits (e.g. for every portion, the eating rate and amount consumed) and must configure the bowl (e.g. max rate, number of portions, number of calories per day) to values set by an informed user.

2. Design

2.1 Block Diagram

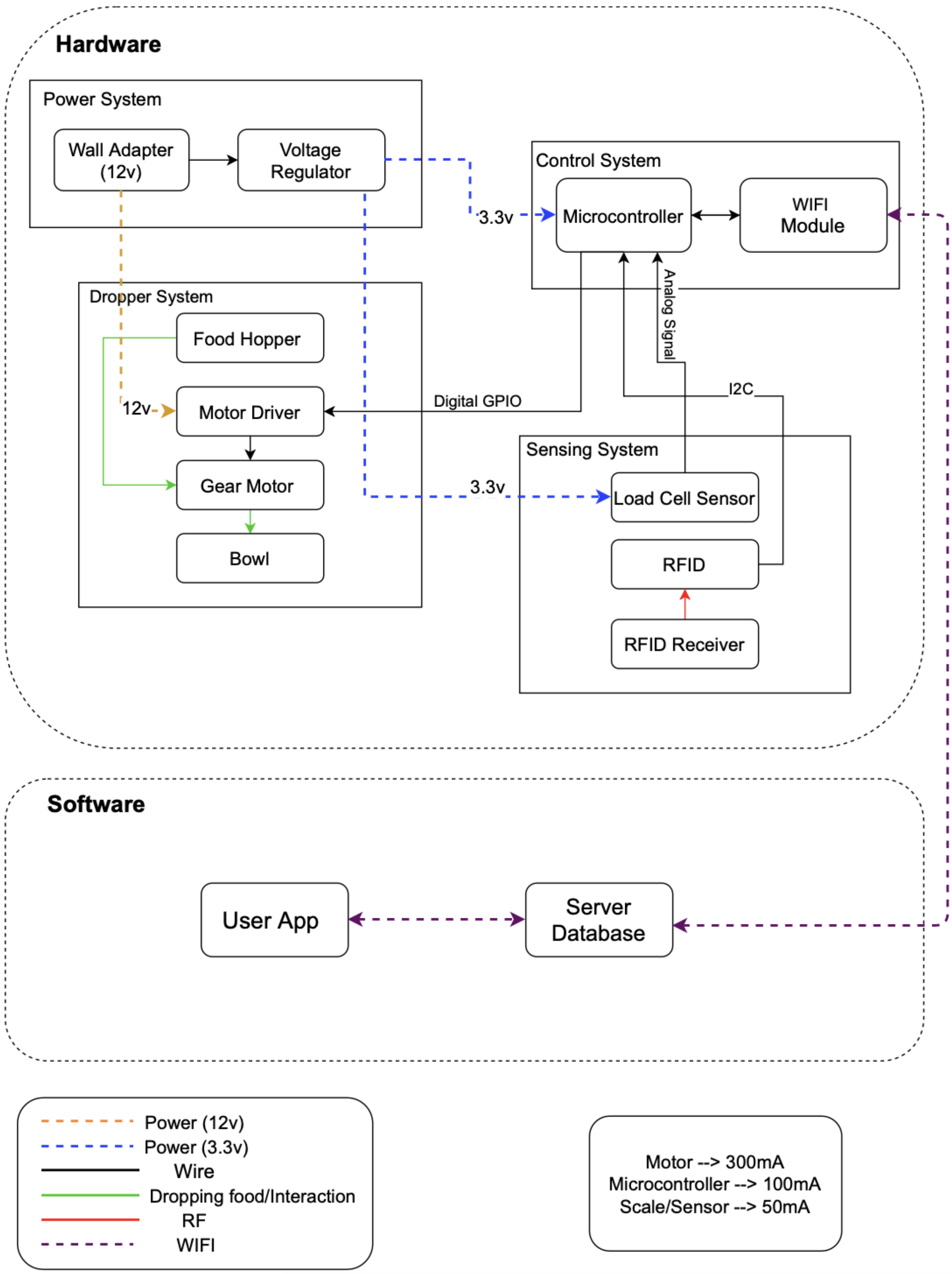


Figure 2. Cat Food Dropper Block Diagram

The system is made of six components: four hardware and two software. All systems are powered by the power subsystem which will ensure that correct voltages and currents are supplied to each subsystem. The control subsystem will receive weights from the sensing subsystem in order to adjust the speed of the dispensing motor in the dropper subsystem. This interaction will enforce the max rate as described in high level requirement one. In addition, the sensing subsystem will distinguish individual cats using RFID, and will notify the control subsystem so that high level requirement two can be met. Finally, the control subsystem and web app will communicate to the server in order to exchange data and cat profiles as described in high level requirement three.

2.2 Physical Design

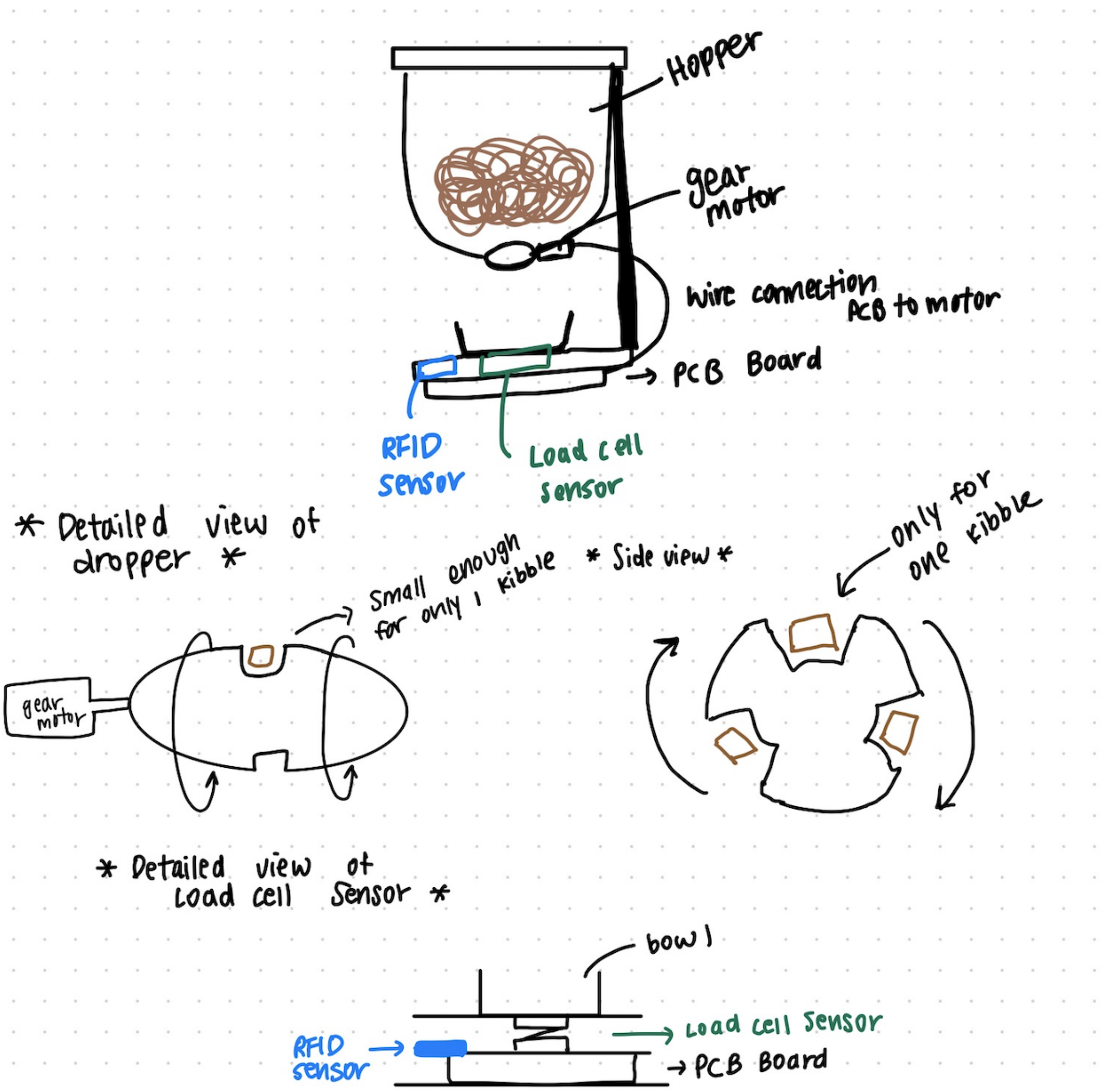


Figure 3. Cat Food Dropper Physical Diagram

The physical design of the cat food dropper will vaguely resemble a coffee machine. There will be a box foundation which will house the PCB, power subsystem, RFID sensor and weight sensor. This bottom will likely be weighted, or extended in order to provide a stable base. Next to the base will be a food hopper that can dispense to the bowl sitting on the base via a motor controlled mechanism. The dispensing mechanism is a rotating cylinder with several small divots large enough to hold one 6mm x 7mm x 4mm kibble piece. The mechanism will be rounded in order to push excess pieces away as it turns and the motor will be strong enough to crush pieces that would otherwise jam the system.

2.3 Subsystem Requirements

2.3.1 Power Subsystem

The Power Subsystem plugs into a standard wall outlet and converts the voltage to the adequate DC voltage required by the other subsystems. The Power Subsystem must be capable of outputting sufficient current such that all devices can be powered consistently.

Requirement	Verification
1. Our dropper must plug into the wall; The Wall adapter must take as input 110-240 VAC. 2. The voltage regulator must output 3.2V ± 0.1V to power the load cell sensor and the microcontroller. 3. Voltage regulator must be able to provide up to 300mA of current.	1A. Use a multimeter to check that the wall adapter's barrel jack is supplying 12V (tolerance dealt by adapter's spec). 2A. Use an oscilloscope to check that the voltage regulator output is 3.2V ± 0.1V 3A. Use a multimeter to check the current output is 300mA ± 50mA

2.3.2 Dropper Subsystem

The Dropper Subsystem uses a wheel turned by a motor to dispense food into the bowl. The motor speed is modulated by a PWM signal generated by the microcontroller of the Control Subsystem.

Requirement	Verification
1. Dispenser is able to turn without being jammed by pieces of food. 2. Dispense food with 0.222g ± 0.222g granularity; the weight of one kibble for an average sized cat is approximately 0.222g. The dispensing mechanism's divots should only fit one kibble piece per divot. However, it is possible that the divot passes by without a piece of food fitting in. Likewise, it is possible for a fraction of a kibble piece to be crushed while the mechanism turns for an added amount of up to and approaching 0.222g in the divot. See tolerance analysis for more details. 3. Hopper's volume must be at least 0.5 cup which is the amount of food needed for an average cat consuming 250 calories [3].	1A. With food in the hopper, power the dispensing system for 5 trials simulating 5 portions throughout the day. In each trial, dispense 20g of food continuously. Dispenser should be able crush any jams and continue dispensing. 2A. With food in the hopper, power the dispensing system for 5 trials. In each trial, empty 10 divots. For each divot emptied, measure the weight and ensure the amount dispensed falls within 0.222g of the target 0.222g. 3A. Put 0.5 cup of cat food into the hopper to see if the hopper has enough space. Hopper top should be able to sit flush with the rest of the container.

2.3.3 Sensing Subsystem

The Sensing Subsystem is composed of all the sensors in the system: the scale and the cat identification blocks. The scale is used to weigh the amount of food that has been dispensed by the Dropper Subsystem and the amount currently in the bowl. The scale reports the weight (in grams and milligrams) to the microcontroller in the Control Subsystem. The cat identifier uses an RFID transceiver to distinguish individual cats. This sensor reports which cat is using the bowl to the microcontroller in the Control Subsystem.

Requirement	Verification
1. The scale must accurately report the weight of the food in the bowl with 0.222g accuracy. 2. RFID correctly identifies which cat is using the bowl within a proximity of 18cm.	1A. Take measurements of the same kibble amounts on a kitchen scale. Ensure they are equal in decimal values as far as they go out. 1B. Using a calibrated control scale, and our load cell sensor. Drop kibbles in sets of one, five, ten, and twenty and measure their weight on the scale in g. Ensure that the measured weights match the control weights within 0.222g of accuracy. 2A. Bring the tags one at a time to the RFID reader with fur covering part of it and 17.25 cm height above the ground and 5.14 cm in horizontal distance. Verify that the sensor is reading the correct tag. Repeat this with all tags.

2.3.4 Control Subsystem

The Control Subsystem receives the dispensed food's weight (in grams) from the Sensing System's scale. Additionally, it uses a timer to count a dispensing period. Every elapsed period, the Control Subsystem will compare the current weight of the food to a "max eating rate" (in g/min). This max rate is either manually set by the user, or provided to the user as a recommendation based on their cat's weight, breed, gender, and age. After the comparison, the dispenser will drop the difference between the max rate and the amount currently in the bowl. This ensures that in the next period, the cat's consumption rate is still less than or equal to the max rate.

The Control Subsystem also learns which cat is using the bowl from the Sensing Subsystem's cat identifier. This information is used to retrieve the cat's "profile" from the remote server/database. The profile will persistently store information related to the cat such as max rate, number of portions per day, calories per gram of food, and max calories per day. The following example explains how these quantities are used.

For example, suppose the dispensing period is 1 minute and the user chooses a max rate of 2.75 g/min. If the cat eats 1.75g within the first minute, we would drop 1.75g at the start of the next period so that the max possible rate in the next period is at most 2.75g/s. Likewise, if the cat didn't eat at all, we would dispense 0g of food in the next period because there is already 2.75g in the bowl. Finally, this behavior would continue until we hit the max amount of calories per portion. This is computed using this boolean:

$$(grams\ dispensed) \times (calories\ per\ gram) \leq \frac{max\ calories\ per\ day}{number\ of\ portions\ per\ day}$$

Finally, the Control Subsystem will send statistics (such as average eating speed per portion, and amount eaten) to the server where it will be stored persistently. Additionally, it will request updated cat profile information from the server to be used in the calculations above.

Requirement	Verification
1. The control subsystem must compute the amount of food to be dispensed in the next time step.	1A. Test the situation where the cat eats some of the food. Set the rate to 2.75g/min. In the first minute, remove 1.75g. Ensure the control subsystem correctly replenishes those 1.75g at

2. The control subsystem must control the motor via a PWM signal in order to dispense food. 3. The control subsystem must communicate with the server via WiFi. 4. A maximum of three, distinct cat profiles can be maintained.	the start of the next period. 1B. Test the situation where the cat eats all of the food. Set the rate to 2.75g/min. In the first minute remove all 2.75g. Ensure that the subsystem is programmed to replace those 2.75g at the start of the next period. 1C. Test the situation where the cat eats none of the food. Set the rate to 2.75g/min. Do not remove any of the food in the next minute. Ensure that the subsystem does not dispense any additional food at the start of the next dispensing period. 2A. To ensure there is a PWM signal, an oscilloscope is needed. Test the signal between the motor and subsystem on the board by attaching an oscilloscope to each side. Verify on the screen the waveform is a PWM signal. 3A. Ensure both the control subsystem and server are on the same wifi. Input data into the app: set the rate wanted. As the food dispenses ensure the correct rate of food is dispensed via a timer and scale; the timer will give minutes and the scale will give g for g/min.

	4A. Requires functioning server and web app or preloaded cat profiles. Register three different cats to three different RFID receivers. For each receiver, perform test 1A except with the respective profile's max rate. Ensure that the system is able to customize the experience for each unique tag.

2.3.5 Server

The database will store cat profile information in persistent memory. The cat profile includes biographical info, the cat's max rate, number of portions per day, number of calories per day, calories per gram of food in use, and statistics collected about the cat's eating patterns (consumption speed and meal times). In order to read or write cat profile information, a client must interact through the server; only the server is capable of reading or writing data to the database. The server will communicate using TCP/IP via WiFi with two clients: the Control Subsystem's microcontroller and the User App. For more information about what data is exchanged between clients, see their respective sections.

Requirement	Verification
1. Server must respond to requests to read cat profiles from clients. 2. Server must write requests from clients to the database. 3. Server must handle at least two web app clients concurrently communicating at a time.	1A. Populate test file with dummy cat profile data. Send a valid read request. Ensure the server can parse the request and return the correct dummy cat profile. 1B. Repeat 1A except with an invalid read request. Ensure server does not crash. 2A. Send server valid write request ("e.g. Write "Hello World" > testfile.txt). Ensure

	that requested output is stored in testfile. 2B. Repeat 2A except with an invalid write request. Ensure server does not crash. 3A. With the server running, execute two instances of the web app. Ensure the server does not crash. 3B. With the server running, execute two instances of the web app. For each web app client input new, distinct data into the same fields of the cat profile fields of the web app. Submit the requests simultaneously. Check that no data races occurred by ensuring that only one of the profile's data was saved and not a mix of two.

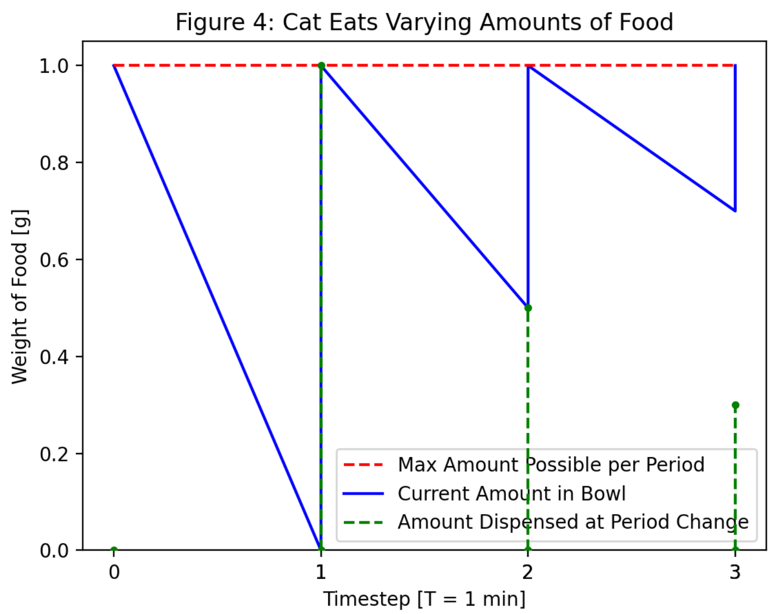
2.3.6 Web App

The User App communicates with the Server/Database via WiFi. It sends user updates of the cat profile to the server where it will be updated in the Database's persistent storage. Updates include the cat's max rate, number of portions per day, number of calories per day, and calories per gram of food in use. Additionally, the dropper will request the statistics collected by the bowl from the Server/Database so that it can be visualized in the app for the user. Examples of statistics include eating rate by the portion, and amount eaten.

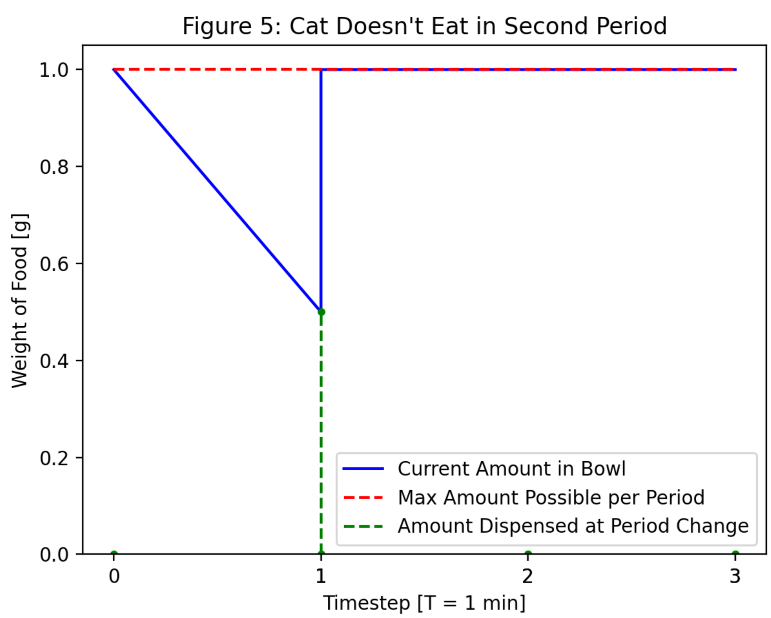
Requirement	Verification
1. Configure the system's max rate, number of portions per day, and max weight of food per portion.	1A. Connect the device to the server. Set the max rate to 1g/min using the Web App. Bring a test RFID chip to the device in order to

2. Visualize data collected by the bowl. 3. Communicate with the server via WiFi.	activate the dispenser; do not remove any food yet. Ensure that only 1g is dropped in the first minute, and no food is dropped afterwards. Remove 500mg of food. Ensure that 500mg of food is dropped in the next minute. 1B. After test 1A, change the max rate to 20g/min. Repeat the process of 1A, this time ensuring the max rate changed from 1g/min to 2g/min. 2A. Load test values for amount eaten and eating rate per portion onto client. Ensure visualization function creates a readable graph and that the data is consistent with the test values. 3A. Populate a test file on the server with content, “Hello World.” Initiate a read request to the server. Ensure returned content is identical to the test file using *diff*. 3B. Populate a test file on the web app client with content, “Hello World.” Initiate a write request to the server. Ensure file written to server content is identical to the test file using *diff*.

2.3.7 Supporting Figures

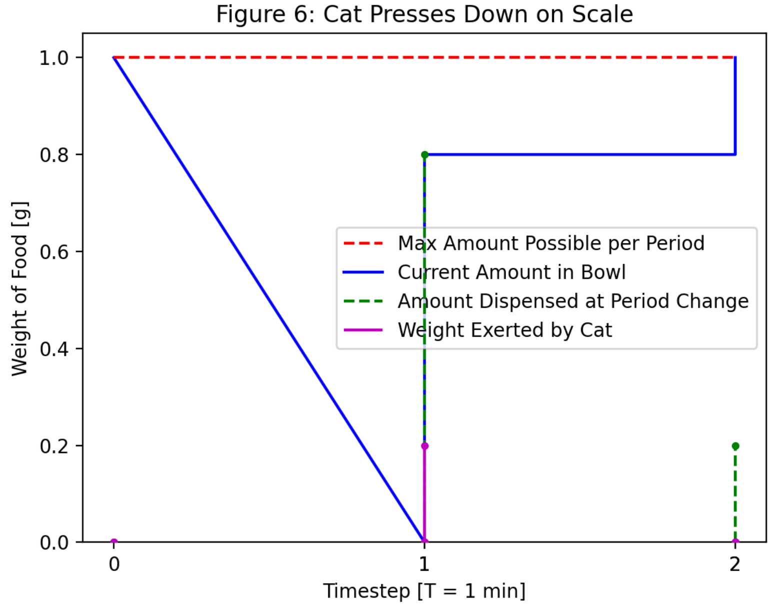


In this graph, 1g is in the bowl at time T=0 min. In the first period, the cat eats 1g of food. Then, the dispenser replenishes it back to 1g at T=1 min. During the next period, the cat eats 0.5g of food, so the dispenser replenishes 500mg at the end of the period, T=2 min. Finally, during the third period, the bowl only has 700mg of food left, so the dispenser replenishes 300mg at T=3 min.



In this graph, the bowl starts out at 1g of food. During the first period, the cat eats 500 mg, so the dispenser drops 500mg at T=1 min. However, the cat does not eat any of the food during the second period. Thus, at T=2 min, the dispenser does not drop any additional food. The weight of food in the bowl would remain at 1g until the cat continues to eat it.

Figure 6: Cat Presses Down on Scale



Finally, in this example, the bowl starts with 1g of food which the cat eats completely within the first period. As the next period begins at T=1 min, the cat is applying 200mg of force on the scale. As a result, the dispenser drops only 800mg of food, believing there is 200mg of food already in the bowl. This scenario is acceptable by our standards for two reasons: the cat will neither be underfed nor overfed. Observe that even though the bowl is not dispensing the maximum amount of food at T=1 min, the cat will not end up being underfed. This is because the dispenser will stop feeding the cat once the portion's allotted amount has been dispensed as opposed to having a maximum number of dispenses. While there will have to be more dispensing periods to drop the full amount, this does not lead to less food being dropped overall. Furthermore, since weights applied by the cat are transient, subsequent periods where the cat is no longer applying weight to the scale will still dispense the full amount. Finally, transient weights only allow the amount of food dropped to be less than the maximum; therefore, transient weights do not pose the problem of overfeeding the cat.

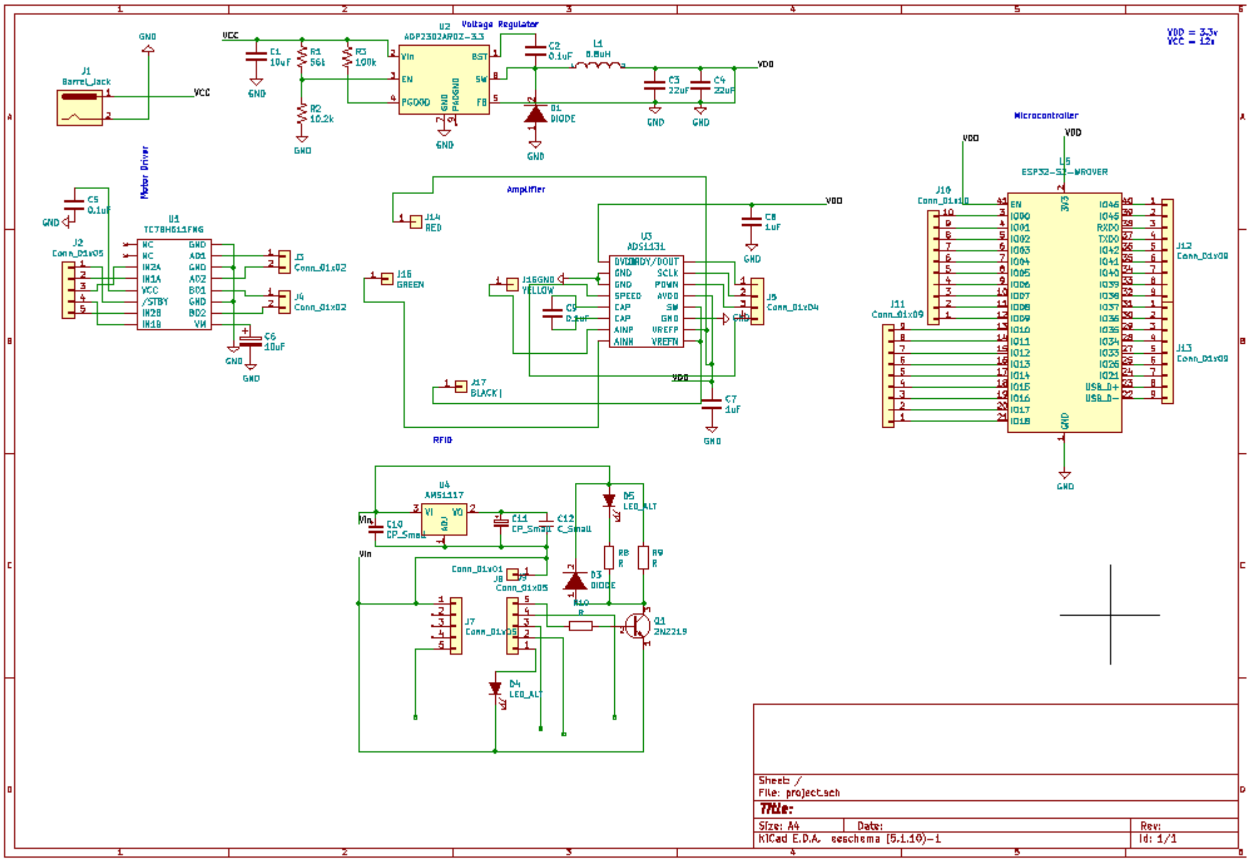


Figure 7. Cat Food Dropper Schematic

Figure 8: Microcontroller Firmware

Michael Park | September 30, 2021

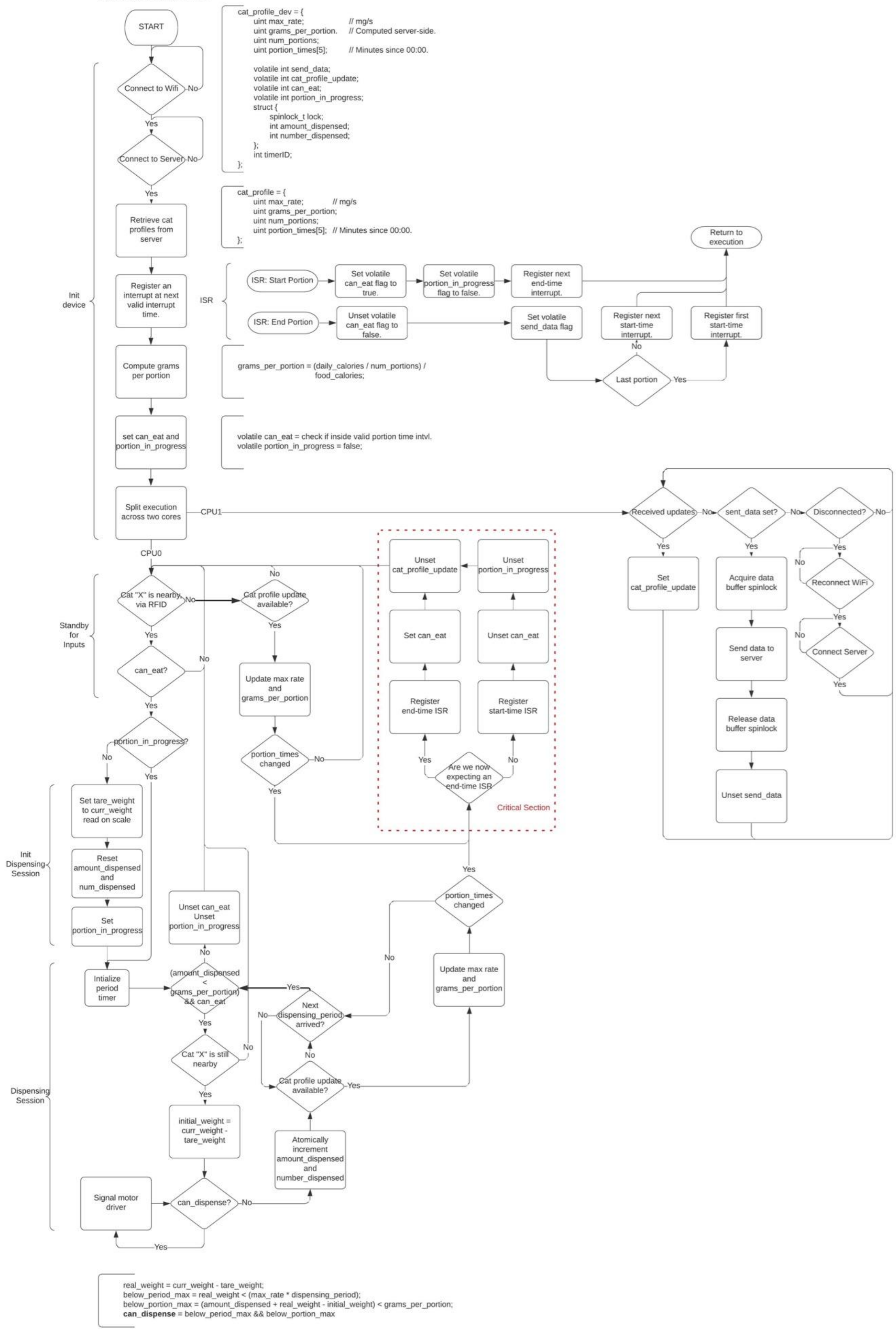


Figure 9: Server Software

Michael Park | September 30, 2021

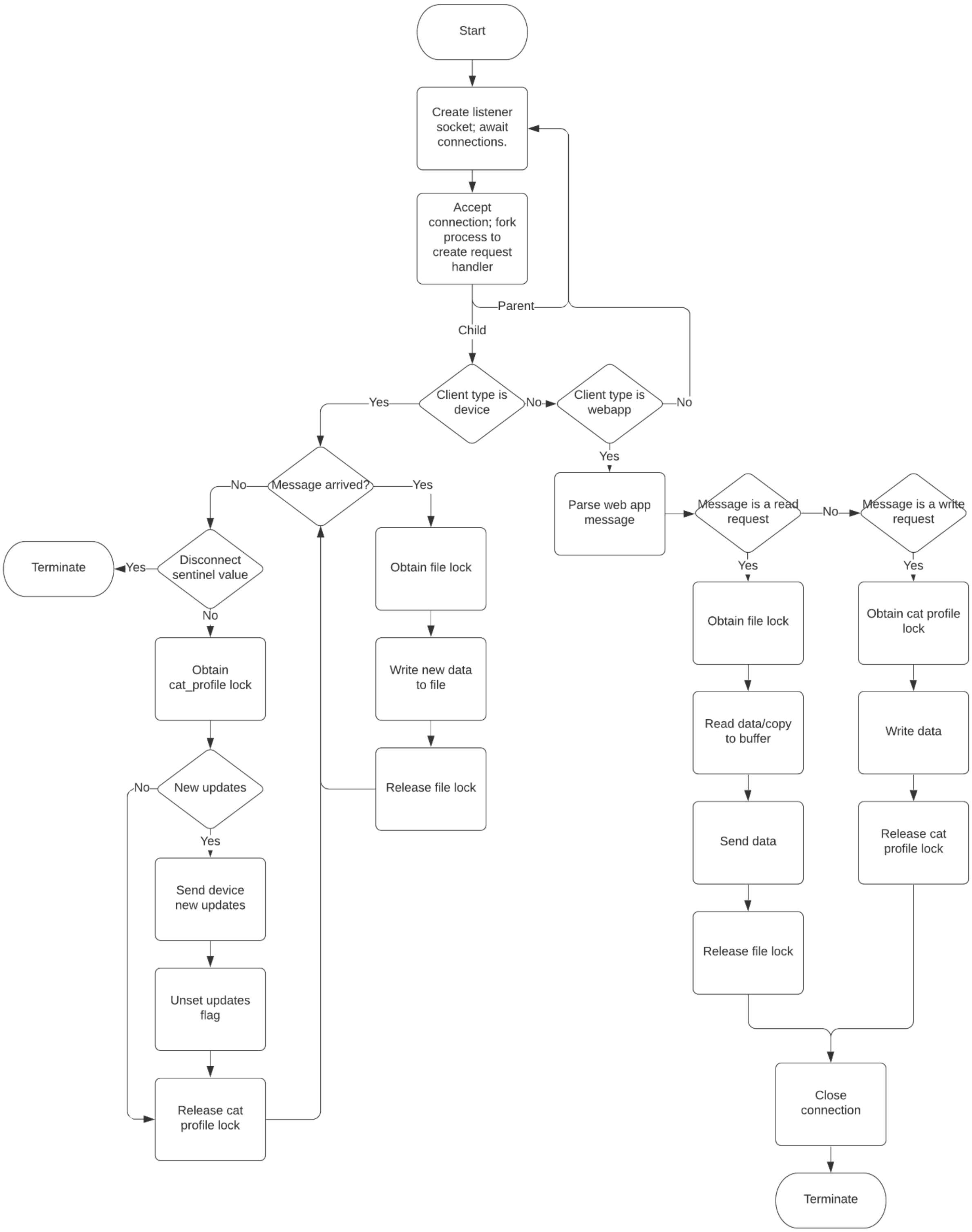
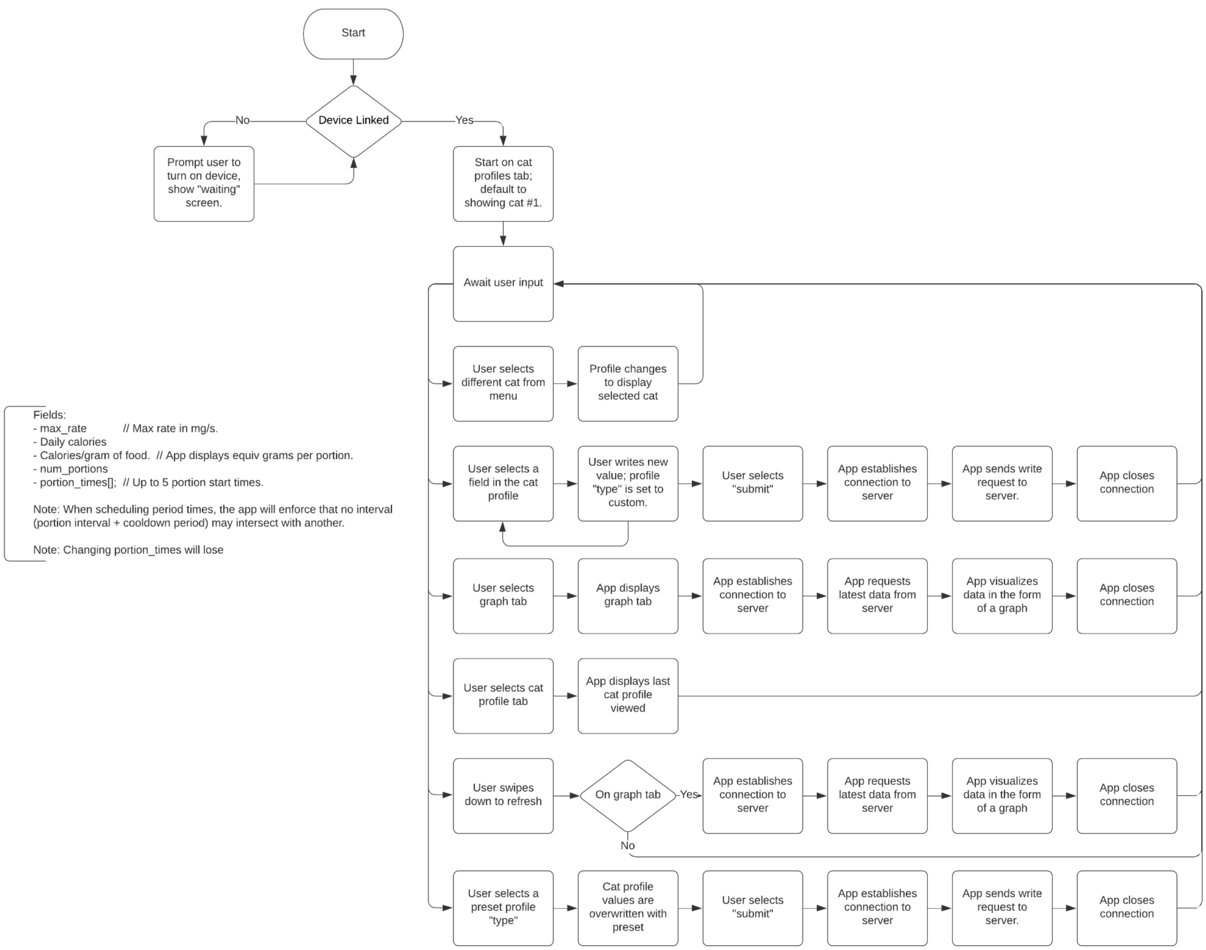


Figure 10: WebApp Software

Michael Park | September 30, 2021



2.4 Tolerance Analysis

2.4.1 RFID

A key part of our project that is essential to our success is that the RFID works successfully. We plan to have the RFID resting between the bowl and where the cat is standing and the tag will sit on the cat's collar. The RFID is programmed to work at a distance up to 18cm separating the hardware and the tag. The cat will wear the tag around its neck which sits about ¾ of its height. The average cat has a height of 23 cm tall [8]. This would indicate that the average cat would have their collar and tag hanging around 17.25cm. The cat will stand a distance x from the bowl in which the RFID will pick it up. If the height is 17.25 and the total length the RFID and tag can be apart is 18 (hypotenuse side), this leaves the horizontal or distance from the bowl to be 5.14cm. A can walks on average 89.4cm/s [9]. This would mean if the cat was walking full speed in the last second it would easily reach that 5.14cm mark. There would be approximately 0.05 seconds between the time the RFID picks off the tag and the cat reaches the bowl. In theory if there was no delay the food could dispense it immediately. But since the gear must turn, there is a slight delay. We are working on a one second period, so at most the cat would have to wait about 0.95 seconds to get its first round of food. However the cat must also bend its head down once it reaches the bowl, which if we take to be about one second as well, the cat will reach its head into the bowl the same instance the food round of food comes out.

2.4.2 Dispensing Amount

The tolerance for the amount of food dispensed per divot is 0.222g ± 0.222g. The reason we provision for a 0.222g tolerance is to support two cases: the divot is empty, or the divot is overfilled.

The first case, the divot is empty, may happen probabilistically if the food doesn't naturally drop into the divot as the wheel is turning. While this leads to some inefficiency in that the wheel must turn for longer in order to dispense the missing kibble, it does not cause the cat to be underfed in the long run. This is because the portion remains active until the entire portion in grams has been dispensed, not for a certain number of motor rotations. Thus, even if a divot is empty when dispensed, the would-be dispensed kibble will *eventually* be dispensed.

The second case, the divot is overfilled, occurs when the wheel turns and crushes a kibble piece in order to unjam the mechanism. Since the divot will already be small enough that only one 0.222g piece of food can fit, only some fraction of the crushed piece will fit. Thus, we provision 0.222g of tolerance to allow up to one additional kibble in crushed form to be dispensed from one divot. Additionally, this means that we must apply a tolerance of +0.444g to the total amount dispensed for each portion. This is because in the worst case, the amount of the portion leftover to dispense is ~0g and the divot that will dispense holds 0.444g.

3. Cost and Schedule

3.1 Cost Analysis

3.1.1 Labor

The labor rate for our team was calculated from the 2019-2020 annual Illini Success Report [4]. Harin and Lexie are Electrical Engineering majors with an average salary of $76,129. Michael is a computer engineering major with an average salary of $99,145. There are a total of 9 weeks remaining in this semester for this class. We decided to work at least 10 hours a week. This will be a total of 90 hours. An electrical engineering major would get $3,294.00 for this project and a computer engineering major would get $4,290.30.

Name	Michael Park	Harin Cho	Lexie Kolb
Rate	$47.67	$36.60	$36.60
Hours worked	90 hrs	90 hrs	90 hrs
Total Labor cost	$4,290.30	$3,294.00	$3,294.00

3.1.2 Parts

Part	Parts name	Price per Unit	Quantity	Price
Wall adapter	N/A	$8.29	1	$8.29
Bulk Voltage Regulator	ADP2302ARDZ	3.15	3	$9.45
Motor Driver	TC78H611FNG	1.63	3	$4.89
Gear motor	2183-4887-ND	34.95	2	$69.90
Amplifier	ADS1131ID	$6.85	3	$20.55
Load cell sensor	SEN-14727	$8.95	3	$26.85
RFID reader	SEN-11827	$29.95	1	$29.95
RFID Chip	SEN-09417	$3.95	1	$3.95

Microcontroller	ESP32-Wrover-E	$3.90	2	$7.80
ESP32-dev board	WRL-15663	$20.95	1	$20.95
MIcro USB to USB cable	CAB-13244	$1.95	1	$1.95
			Total Price	**$204.53**

3.2 Schedule

Week	Michael Park	Harin Cho	Lexie Kolb
9/27	Complete Design Document.	Complete Design Document. Work on getting PCB Board approved.	Complete Design Document. Work on getting PCB Board approved.
10/4	Complete Design Review. Fill out google form to acquire parts needed for the project.	Complete Design Review. Get PCB Board approved and get on first round order.	Complete Design Review. Get PCB Board approved and get on first round order.
10/11	Work on software code.	Complete PCB Board assembly with soldering and mounting.	Complete PCB Board assembly with soldering and mounting. Work with Michael on the front end coding.
10/18	Complete software code.	Test hardware systems, address issues and fix problems.	Work with Micahel on the front end coding. Test hardware systems, address issues, and fix problems.
10/25	Test software code. Perform full system testing with software.	Finish hardware system testing. Perform full system testing with software.	Finish hardware and software testing. Perform full system testing with software.
11/1	Complete Individual Progress Report. Continue testing and	Complete Individual Progress Report. Continue testing and	Complete Individual Progress Report. Continue testing and

	debugging the full system.	debugging the full system.	debugging the full system.
11/8	Create a Mock Demo and begin demonstration.	Create a Mock Demo and begin demonstration.	Create a Mock Demo and begin demonstration.
11/15	Mock Demo. Begin working on the final demonstration.	Mock Demo. Begin working on the final demonstration.	Mock Demo. Begin working on the final demonstration.
11/22	Fall Break	Fall Break	Fall Break
11/29	Final Demonstration. Work on presentation and Final Paper.	Final Demonstration. Work on presentation and Final Paper.	Final Demonstration. Work on presentation and Final Paper.
12/6	Final Presentation and Final Paper	Final Presentation and Final Paper	Final Presentation and Final Paper

4. Ethics and Safety

During the development of our project, we have to keep ethics at the forefront. IEEE code I1 states how we must “hold paramount the safety, health, and welfare of the public” [6]. This is applicable to our project as we are trying to create a cat food dispenser which is physically safe for a cat to use. We will avoid breaching this rule by ensuring all parts of the dispenser are cat-friendly and only cause the cat to slow down eating for its own good. Another ethical standard that could arise during our project is originality. As there are other cat feeders on the market, we have to make sure everything we produce is our own idea. To do this we will only create components that we deem we are competent to create. This goes along with the code 2.6 in ACM Code of Ethics; we vow to only design aspects of the dropper we know we can complete with our own knowledge and some additional research [5].

After our project is completed, the owner of a cat could accidentally or intentionally misuse our product. For example they could turn the speed so low the cat does not wait long enough for the food to dispense, and it ends up undereating. It could be the opposite situation as well in that the user could turn the speed all the way to the maximum and purposely cause the cat to eat too quickly and throw up. Although we do not have control of a person’s misuse in their own home, we could provide warnings of the dangers of doing this to the cat. We could give recommended levels that cats should eat as well as guidance on how to avoid this situation. We, as creators, would never use the product in an evil way as we want to “contribute to society” as stated in 1.1 of ACM Code of Ethics [6]. We will only set the rate to a setting the cat can handle and should eat at to maximize its health.

Specific safety concerns involved in our project are mainly involved when we are working in the lab. After reviewing the federal and state regulations, industry standards, and campus policy this is what we will ensure we do in each lab session. We will never work alone in the lab, will not bring food into the lab, will clear off the working space so it is free of hazards, will report any broken equipment, will not use two hands on a circuit when powered, and will clean up after ourselves [7]. The biggest potential safety concern for us would be accidentally touching the circuit when we should not, which could create a circuit through ourselves, or not reporting something if it goes wrong. If we are careful with both safety concerns, we should run smoothly.

There is nothing high risk that we will be working with. The highest voltage involved in our project is the input voltage of 12V. A procedure we have taken to mitigate this hazard of touching this voltage is our voltage regulator. The regulator takes this 12V and turns it into 3.3V. Additionally, this will be on the board and covered by our feeder so that neither a cat nor human could touch the hardware. This ensures all users are safe.

5. References

[1] Ávila, Lorena, and Mary Ellen Carselli. “How to Slow down a Cat Who Eats Too Fast & Throws up: Help Your Cat.” Cat Food Dispensers Reviews, 25 Feb. 2019, www.catfooddispensersreviews.com/how-to-slow-down-a-cat-who-eats-too-fast/.

[2] Feeding Times and Frequency for Your Cat, et al. “Feeding Times and Frequency for Your Cat.” vca_corporate, vcahospitals.com/know-your-pet/feeding-times-and-frequency-for-cats.

[3] Giorgio, K. M. (2020, August 24). *How much should I really feed my cat?* Daily Paws. Retrieved September 16, 2021, from https://www.dailypaws.com/cats-kittens/cat-nutrition/cat-food/how-much-feed-cat.

[4] *Box*, https://uofi.app.box.com/s/1t8xj69117lrsqm7753ujnrg8yyrtcwn.

[5] *The code affirms an obligation of computing professionals to use their skills for the benefit of society.* Code of Ethics. (n.d.). Retrieved September 13, 2021, from https://www.acm.org/code-of-ethics.

[6] *IEEE code of ethics*. IEEE. (n.d.). Retrieved September 13, 2021, from https://www.ieee.org/about/corporate/governance/p7-8.html.

[7] *ECE 445 Lab*. Lab :: ECE 445 - Senior Design Laboratory. Retrieved September 13, 2021, from https://courses.engr.illinois.edu/ece445/lab/index.asp.

[8] Turner, P., & Pam TurnerPam *How tall is the average cat?* CatWiki. Retrieved September 29, 2021, from https://www.catwiki.com/faqs/how-tall-is-the-average-cat/.

[9] Maris Munkevics, S. M. (2020, August 19). *Can a cat run faster than a human? Can a dog outrun a cat?* pet. Retrieved September 29, 2021 https://pet-happy.com/how-fast-can-a-domestic-cat-run/#:~:text=But%20how%20fast%20can%20a,close%20to%2045%20km%2Fh.

[10] Us, Iams. “Does Your Cat Have Tummy Troubles?” *IAMS US*, IAMS US, 17 Sept. 2021, https://www.iams.com/cat/cat-articles/does-your-cat-have-tummy-troubles.