Rochester Institute of Technology

Final Project Report

CMPE-496

Computer Engineering Design Project I

Automated Cat Feeder

By:

Matt Smith, Chris Ranc, and Shabab Siddiq

Description:

A smart solution to automate and regulate the feeding of common household pets

Proposal Submission Date : 5/06/2018

Table of Contents

I Overview

Needs Statement

Animals are a welcome addition to many homes; unfortunately, their needs can sometimes conflict with modern lives. It can often be stressful to keep track of *how* your animal is eating, when it is eating, and if it is hungry while you are away. Often times, cats or dogs left to their own devices can cause havoc. There is a need for a system that would take guesswork out of taking care of one's animal while assisting in reducing the overall stress involved in pet care.

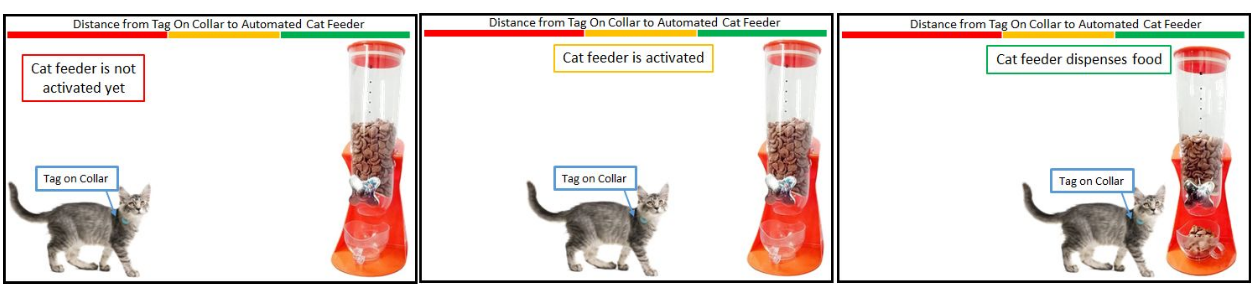
Objective Statement

The objective of this project is to design and prototype a device to automate and regulate the feeding of household pets. The device should permit owners to configure feeders to feed their pets at appropriate times and with healthy portions. The device should also allow the multi-pet owner to designate feeders to each pet to prevent them from eating one another's food.

Description

The system will consist of a master feeder and several supplementary feeders that will communicate using a localized wireless network broadcasted by the master feeder. The user will be able to connect to this network and configure each feeder via a smartphone app. The user will also be able to assign a feeder to a particular pet, the portion size to be dispensed, and the feeding time to dispense the pet's food when that pet is within the feeder's range. The feeders will be able to store information about the pet's dietary history, thus permitting the user some basic analytics to see if the pet is eating enough. In order to achieve this objective, sensors and servos will be used. There will be a passive tag on the collar of the cat that will be read by a sensor on the automated cat feeder when the cat is nearby for activation. Once the cat approaches the cat feeder, a servo or geared motor will be activated in order to dispense the appropriate amount of food to feed the cat.

Marketing Picture/Diagram



II Requirements and Specifications

Marketing Requirements

1. The user should be able to wirelessly connect to the Automated Cat Feeder system.
2. The user should be able to add and remove pets from the feeder system.
3. The user should be able to assign a feeder to a specific pet.
4. The user should be able to modify a pet's feeding time and food amounts.
5. The user should be able to monitor a pet's overall dietary habits.
6. The overall ability to feed one to several animals independently.
7. The capacity for a multi-day food supply.
8. The feeders should only dispense dry food.
9. The device should be capable of long term reliability for continued operation.
10. The device should have a level of dispensing precision to work with different animals.
11. The app should be able to track and analyze eating behavior.
12. The device can easily be set up wirelessly.
13. Multiple devices should be easily set up for scalability.

Engineering Specifications

Table 1a: Requirements Table

Marketing Requirements	Engineering Requirements	Justification
4,5	A. Software to keep track of multiple animals and differentiate behavior based on animals being served	The user should be able to reasonably have the ability to modify a pet's feeding schedule if the pet's eating habits are to change at any time. The user should be able to keep track of the pet's meals. This includes notifications of when the pet has been fed as well as a history of the pet's meal intake. This is important for monitoring the pet's overall dietary habits. Having software to keep track of the pet behavior will be useful in order to achieve this need
2	B. Database to store data for mobile app that presents in human readable format	The user should have the ability to modify the database in the event that the user has a new pet to use with the feeder, or if the user wants to no longer associate an existing pet with the feeder.

Table 1b: Requirements Table

Marketing Requirements	Engineering Requirements	Justification
1	C. Wireless operation (aside from power) to prevent curious animals from chewing on wires	Using a mobile application for the wireless user interface will make it easier for the user to remotely interact with the Automated Cat Feeder as long as both the feeder and the mobile application are connected to a wireless network.
3	D. Ability to wirelessly communicate to supplementary devices that act as simple feeders while the master unit manages smart features	The user should reasonably be able to change a pet's assigned feeder in the event the user wants to change the feeder units, whether it is the master feeder or its connected supplementary unit.
1,12	E. Supplementary and master feeders respectively need a controller with wireless peripherals or peripherals to integrate with a wireless module	Needed to permit network communication between Master and Supplementary feeders.
7	F. High capacity food container at least 0.5 gallons.	For an automatic feeder, it is important for the device to be able to hold multiple days worth of food; otherwise the automatic feeder would not be useful.
10	G. Precise dispensing control. Dispensing depends on food size, ranging from ~8mm to ~18mm	This device is intended to cater to small and large pets alike, therefore food must be able to be dispensed in small quantities and large quantities. Precision is primarily important for dispensing small quantities.
9	H. Long-term reliability for automatic feeding for at least 1 week	It is important that a device trusted with feeding a pet should not jam or crash, preventing food dispensing. For the health of the animal and peace of mind of the owner.
6	I. Dispense food within 5 seconds of detecting a pet near the feeder	Don't want pet to walk away before getting food

III Concept Selection

Existing Systems

Current automated pet feeders on the market meet various parts of our design objectives but are quite costly. The Petwant SmartFeeder by GemTune permits scheduled feeding and system configuration via a smartphone app. The Automatic Pet Feeder by Wireless Whiskers utilizes RFID to release food for the pet when close by and it is at the appropriate feed time.

The Petnet Smartfeeder is an automated pet feeder that focuses on delivering smarter meal portions for the pet, and it comes with a smartphone application as the main user interface. The smartphone application can modify a pet's feeding schedules and monitor a pet's overall dietary habits, but it can monitor only one pet and does not support multiple pets.

The Feed and Go Smart pet feeder is a Wi-Fi enabled automated pet feeder that allows a user to remotely feed a pet, and it also comes with a smartphone application that can modify a pet's feeding schedule and food amount. It is also supported as a web, Android, and iOS application.

Moving away from existing general feeders, more macro components must also be considered, specifically the mechanism being used to dispense the food from the container. This mechanism must have fine control of dispensing food and also be mechanically sound enough to avoid jamming and binding along the path of the food. Several dispensing mechanisms were found in practical scenarios. The first mechanism is the paddle based dispenser seen in cereal dispensers. These are simple but often bind and do not have very fine control. The second mechanism is the sliding door dispenser. This is a simple open and close sliding door that would rarely experience any binding but would be very difficult to control portions. The final mechanism is the feed screw, or auger. This design uses a helical screw to push food along a pipe. Small amounts of food fall between each screw blade and are pushed by the rotating motion. This allows for precise feeding as well as a design that mitigates binding on the edges.

Considered and Chosen Concepts

1. Master and Supplementary feeder systems
   The master feeder needs to be advanced enough to manage multiple supplementary feeders with its own localized network, store feeding configuration data, and control its peripherals. The brands of embedded computer that can suffice for this task are Raspberry Pi, Banana Pi, and Beaglebone. These brands all have variants that can run an operating system to easily manage many tasks. They can run an embedded linux operating system thus allowing for the use of a database for storing information. Each variant also has enough peripherals for controlling the sensors and motors of the feeder and have built in wireless for broadcasting or receiving. There are enough differences, however, to utilize a decision matrix to compare among them.

Table 2: Decision Matrix for Master Feeder Controller

Criteria	Weight	Raspberry Pi 3 B (-1,0,1)	Banana Pi Zero (-1,0,1)	Raspberry Pi Zero W (-1,0,1)	Beaglebone Black (-1,0,1)
Price	3	0	0	1	-1
Adequate Peripherals	4	1	1	1	1
Wireless Built-in	3	1	1	1	1
Development Resources	2	1	0	1	1
	Score	9	7	12	6

Of the devices considered, the Raspberry Pi Zero W variant was the best choice. The Raspberry Pi 3 B is much faster and has more cores than the Zero W. It is 3.5 times the cost of the zero at $35, so it is not cost effective. The Beaglebone was not as popular as the Raspberry pi and almost twice the price of the Pi 3 B at $68.75. The competitor to the Raspberry Pi brand, Banana Pi, has its own Zero variant with wireless and a faster quad-core processor. The board, however, does not appear available on any online shops to purchase or see its price, which prevents this board from being considered.

The supplementary feeder simply requires some kind of microcontroller to operate concurrently with controlling the detection of pets, dispensing of food when needed, and communicating with the master feeder to transmit and receive data. These tasks can be performed by most microcontrollers provided they have proper connections to appropriate modules or sensors to perform those tasks.

Table 3: Decision Matrix for Supplementary Feeder Controller

Criteria	Weight	KL64 (-1,0,1)	Teensy (-1,0,1)	ESP8266 (-1,0,1)
Price	3	0	1	1
Adequate Peripherals	4	1	1	1
Wireless Built-in	3	0	0	1
Development Resources	2	1	1	1
	Score	6	9	12

The ESP8266 wireless module by espressif is normally configured to run firmware called nodemcu allowing versatile and cost effective controllers like the KL64 and Arduino Teensy to connect to wireless networks. However the ESP8266 can also operate using custom firmware, allowing access to its other peripherals such as its UART and PWM related pins. This permits it to control the sensors and motors operating the feeder while at the same time utilizing its internal wireless communication hardware to communicate with the master feeder's network. The network communication is made easier due to the availability of micropython and the Software Development Kit (SDK) provided for the ESP8266. This provides easy to use options to control its hardware and wireless communication. The controller has been widely used by hobbyists in projects ranging from driving remote control cars to remotely reading RFID sensors and transmitting data. This gives an indication that the device can handle at least a moderate workload which the supplementary feeder will require. Stress testing the device for optimal scheduling will be necessary to ensure optimal operation of the wireless communication and utilizing peripheral devices.

2. User Interface for Interaction with the Automated Cat Feeder
The proposed platforms of the user interface are a web application, an Android application, and an iOS application. Table 4 represents the constructed decision matrix to ultimately decide the platform to implement the user interface.

Table 4: Decision Matrix for the User Interface

Criteria	**Weight/ Importance**	**Web Application (0, +1, +2)**	**Android Application (0, +1, +2)**	**iOS Application (0, +1, +2)**
Familiarity and Ease of Development Environment	5	+1	+2	0
Availability of Testing Devices	4	+2	+2	0
Overall Portability	2	+1	+2	+2
Feasibility to Meet Customer Needs and Engineering Requirements	3	+1	+1	+1
	Score	18	**25**	7

As seen in Table 4, an Android application appears to be the best choice for the platform to implement the user interface.

3. Dispensing Mechanism
The proposed mechanisms seen in the existing systems were compared to each other on their merits with regard to these criteria: precision, its ability to control the size of the portion as close to the desired amount as possible; reliability, its ability to work as intended over extended periods of time; and durability, its ability not to break during operation. These criteria are important to the main functionality of the feeder and were the deciding factors in the chosen dispensing mechanism.

Table 5: Decision Matrix for the Dispensing Mechanism.

Criteria	**Weight/ Importance**	**Paddle (-1, 0, 1)**	**Sliding Door (-1, 0, 1)**	**Feed Screw (-1, 0, 1)**
Precision	2	0	-1	+1
Reliability	3	0	+1	+1
Durability	3	+1	+1	+1
	Score	3	4	**8**

Rationale for choice(s)

1. Master and supplementary feeder system was chosen to reduce the cost of feeders needed for owners of multiple pets. This is accomplished by having a localized wireless network for the lower cost supplementary feeders to communicate to the master feeder for configurations and storing data.
2. The Android application was chosen to provide an easy user interface for customers to configure their feeders and see data visualizations of their pets eating habits.
3. For the dispensing method, the feed screw was determined to be the optimal choice in terms of precision and overall reliability.

IV Design

Overall System

The overall system will consist of a master feeder, operated by a Raspberry Pi, and supplementary feeders, that are each controlled by a ESP8266, if the owner has more than one pet. Each feeder will control a motor for food dispensing, RFID sensor for pet detection, and a load cell for food measurement. The master feeder will be broadcasting a localized wireless network for the supplementary feeders to receive pet configurations and transmit feeding statistics back to the master. The master stores feeder configurations and feeding statistics for all feeders on its network in a MongoDB database. A high level diagram of the of the overall system interaction can be seen in **Figure 1** on the following page.

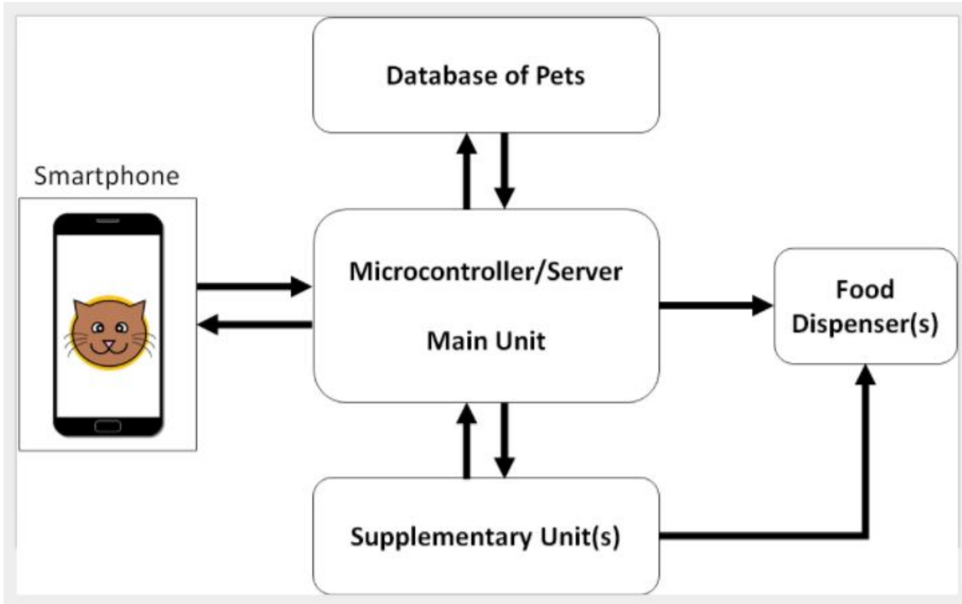


Figure 1: High Level System Diagram

Subsystems

Controller Subsystems

Autonomous functionality of the feeders will be provided by the Raspberry Pi Zero W for managing the master feeder and the ESP8266 for managing the supplementary feeder. When fully developed, the supplementary feeders managed by the ESP8266 will be able request the information pertaining to the pet it monitors when it is time to dispense food, as well as return information regarding whether or not the pet came and got its food. The master feeder also communicates with the user upon request when changing configurations for the pet's feeding times via the app. It will do this through an app while connected to the the unit's localized network or over the user's local network, provided the feeder is connected. GPIO pinouts for the controllers can be seen in **Figure 2** below and **Figure 3** on the following page.

Figure 2: Raspberry Pi Zero W GPIO Pinout

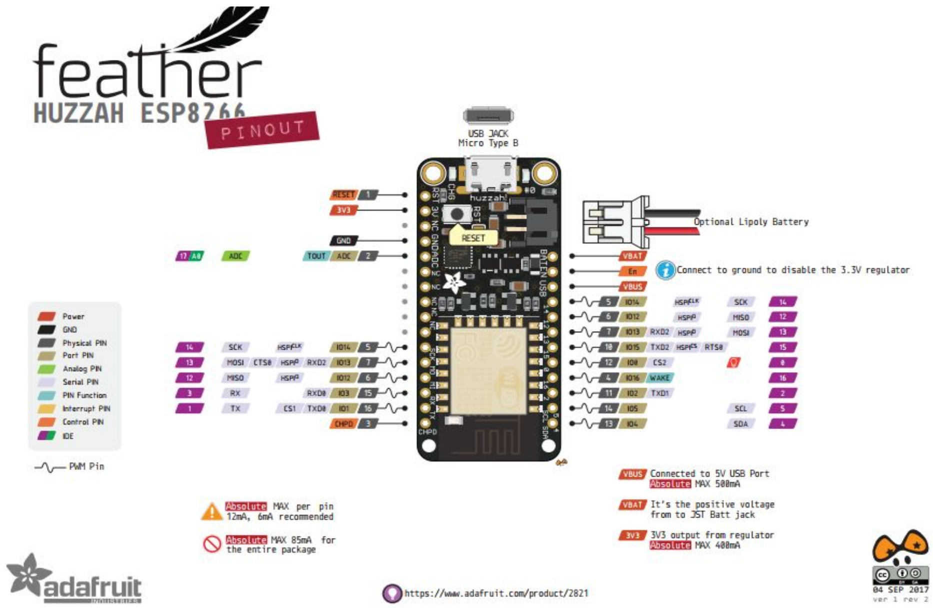


Figure 3: Feather (ESP8266) GPIO Pin Out

<u>Dispensing Mechanism</u>

The most important aspect of the design being considered is the ability to portion food. Therefore, the best choice for the dispensing mechanism is the feed screw design. With this choice the remainder of the dispenser can be designed around it. For this design, only dry food will be accepted. The overall design will be a vertical, gravity fed container to act as a hopper to feed into the feed screw. The feed screw will be held in a well fitting tube with a large enough tolerance to allow dust past the edges. The tube under the hopper will be cut to allow food to enter, and the end will be open to allow for the food to fall. This mitigates the possible binding seen in **Figure 4** below. The feed screw will be driven by a stepper motor to allow for precise control of the feed screw rotation. This will be constructed on an elevated base to allow for food to fall into the bowl.

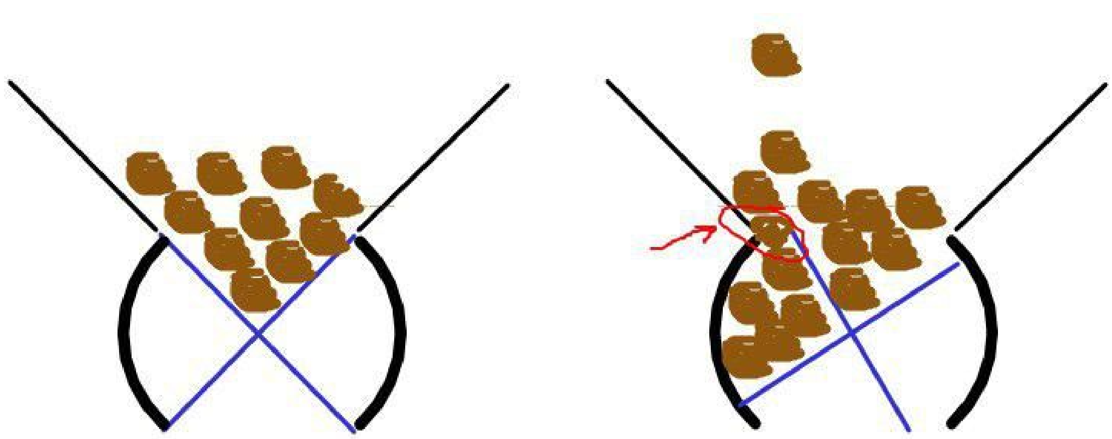


Figure 4: Illustration of the Binding

The first focus of this design will be in the feed screw itself. A feed screw is a common design that is used in many applications including material extractions. It is very similar to a drill or auger in most ways except that the feed screw remains stationary while augers and drills move. Since feed screws are a relatively common design, it is easy to find many examples online. For the purpose of a prototype and the lack of considerable modeling experience, open source designs will be used as a basis for creating a working prototype. One particular design found provided the same service as this project, albeit far more simple and static in configuration. This design used a 3D printed auger within a PVC tube and fed by a plastic container. This 3D auger design can be modified to fit the scale of this design and can be made to fit the motor of our choice. A 3D rendering of this design can be seen in **Figure 5** below.

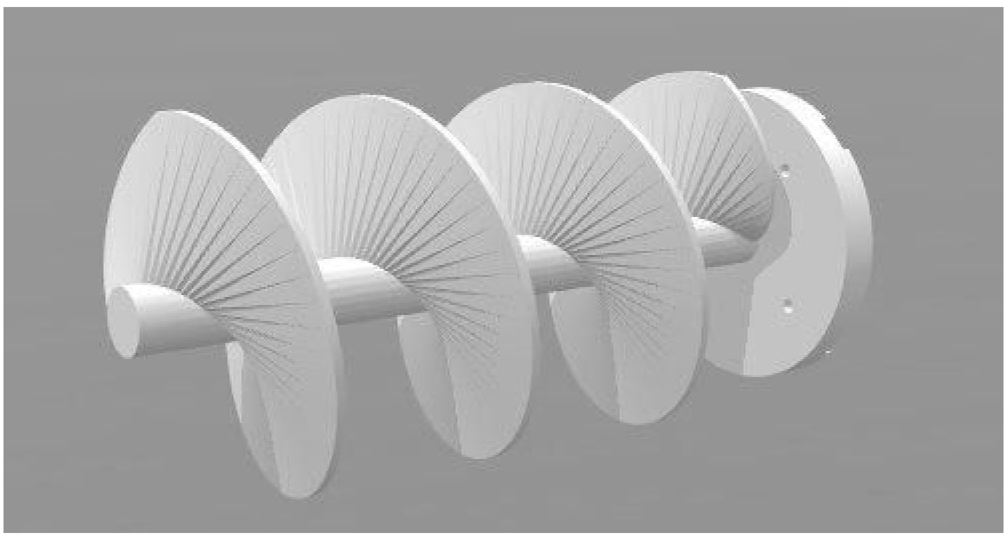


Figure 5: 3D Model of the Feed Screw

This feed screw/auger design will be used. Material must be selected for the outer tube. The original application called for a 1.5’’ female PVC T coupling. Since the inner diameter of this fitting is the outer diameter of standard PVC, the design will likely need to be modified to fit a standard sized piece of pipe. It would be beneficial to use a standard size pipe with a smooth interior to reduce and mitigate any possible binding that could be caused by a rough surface. PVC is also commonly used in water and food applications, making it a safe choice to use with food. It will still be important to have clean cuts and wash after working on it to ensure nothing could get into the animal's food.

In order to drive this feed screw, an electric motor with both feedback and continuous rotation will be needed. A simple DC motor does not have the ability to know its rotational position and would not be a good choice for driving something where you need fine control. A servo can be aware of position, but a continuous servo loses that ability and is just a geared motor. A stepper motor can have fine grain control over movement with the ability to rotate continuously in a single direction. The torque requirements of the stepper motor are currently unknown; therefore, specific decisions on motors and belt/pulley configurations are currently unknown as well. What is known is that additional electronics will be needed to drive a stepper motor using a microcontroller.

Finally, the hopper/food container can be made from durable acrylic panels. These panels can be cut to construct a proper taper at the bottom of the hopper to ensure smooth feeding. The acrylic can be constructed via screws and pre-drilled holes, combined with a glue to seal it. This will result in a strong and safe container for an animal's food. A cover can be constructed in the same fashion to prevent foreign objects from falling in.

With proper clearances, this design will be able to avoid most binding while being powerful enough to push the food. The feeder screw allows for fine control due to how it releases small amounts of food per revolution. With a smooth outer tube and a correct blade angle, binding should be rare. This setup allows for food to be filled high in the hopper, supporting a multi-day food supply.

Circuits and Assemblies

Figure 6 represents the overall diagram of the Automated Cat Feeder, including the sensors and motors that will be used for the food dispensing.

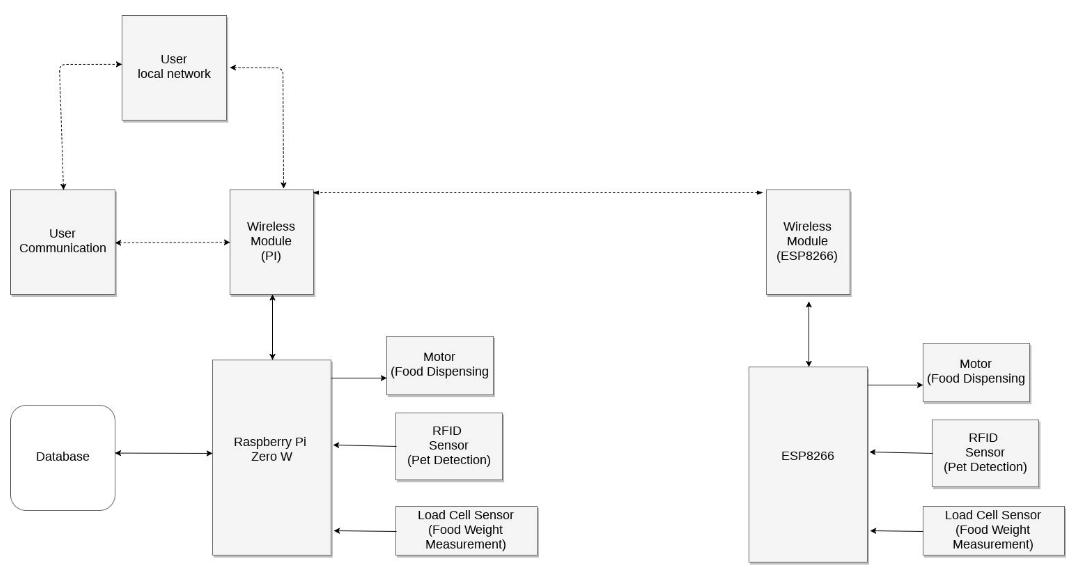


Figure 6: Overall System Diagram

User Interface and Controls

An Android application serves as the user interface. The user should be able to wirelessly connect to the Automated Cat Feeder, add and remove pets from the feeder's database, assign feeders to pets, modify a pet's feeding schedule and food amounts, and monitor a pet's overall dietary habits. **Figure 7** represents the layouts that that user will see when using the application.

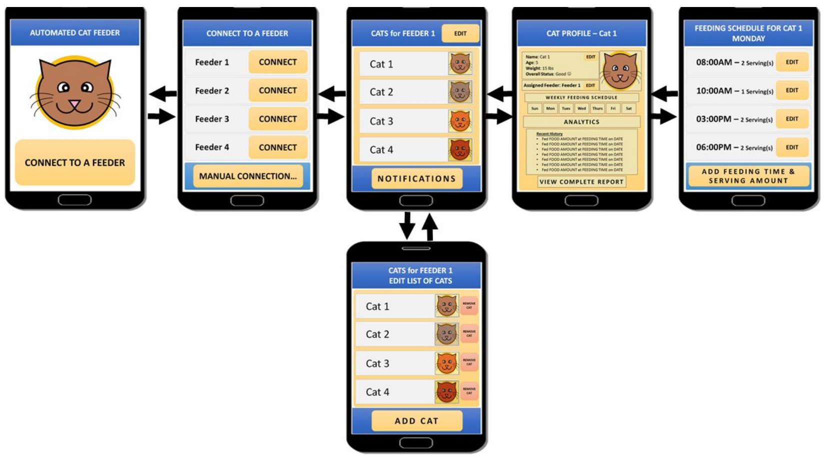


Figure 7: ACF App Layouts

Figure 8 represents the screen layouts that will be used in order to wirelessly connect to the Automated Cat Feeder.

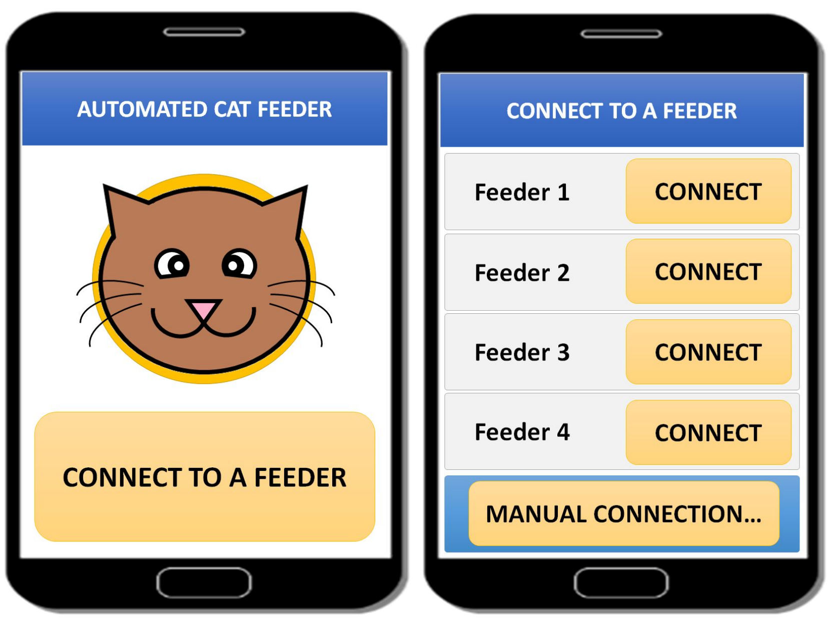


Figure 8: Wireless Connection Screens

Figure 9 represents the screen layouts that will be used in order to add and remove pets from the Automated Cat Feeder system.

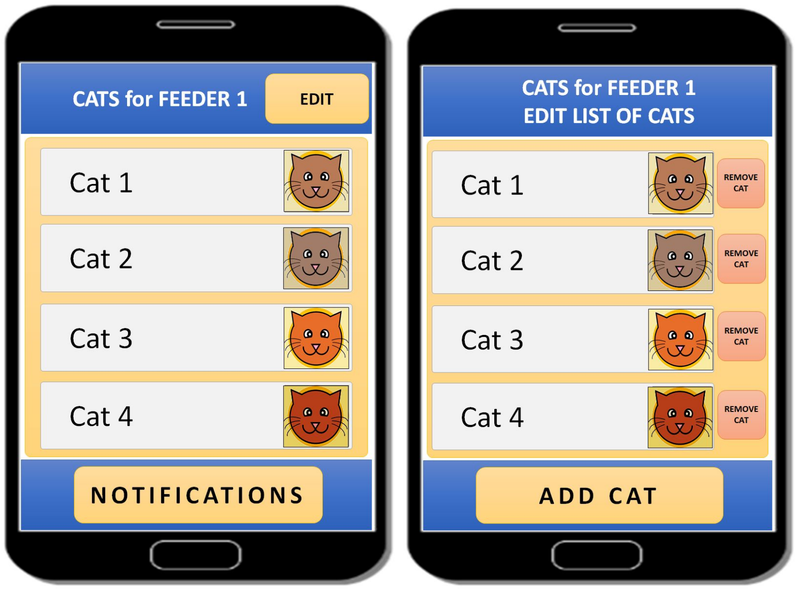


Figure 9: Pet List Screens

Figure 10 represents the screen layouts that will be used in order to modify a pet's feeding schedule and monitor the pet's overall dietary history. It also allows the user to assign or reassign a feeder to a pet.

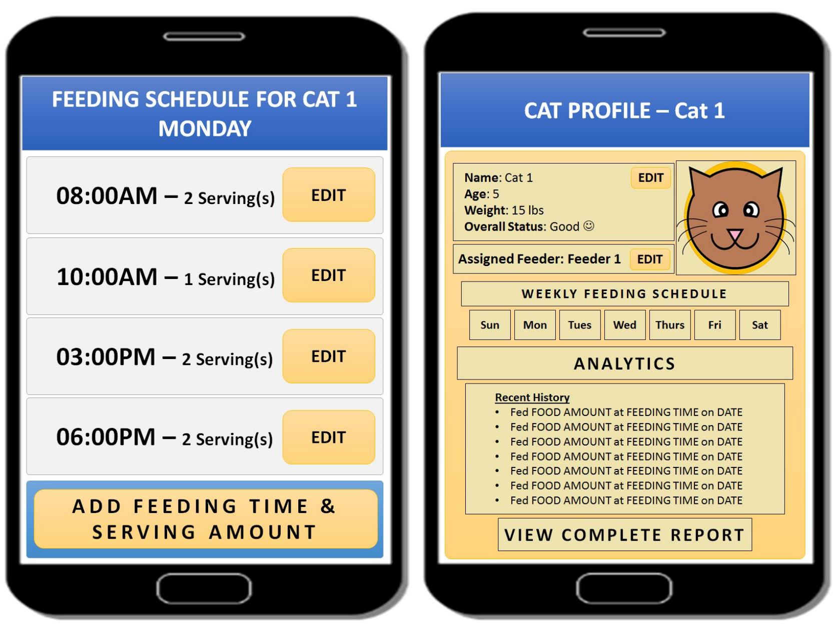


Figure 10: Pet Profile Screens

Engineering Standards

A primary standard to be used in the design is the HTTP protocol. This will be used to send data, commands, or requests between the feeders and the mobile app. By implementing this standard, each device can expect uniform formats and be able to operate on them without compatibility issues across software versions, as the standards are relatively old and not subject to extreme change. Most, if not all, Android devices manufactured in the last seven years are capable of using multi-band WiFi; however, there are more realistic concerns over versions of Android still supported by assorted distributors. This results in relatively modern devices. Therefore, compatibility is of little concern.

Multidisciplinary Aspects

Since the Automatic Cat Feeder requires physical construction and design, much of the mechanical hardware is outside the scope of the Computer Engineering discipline. There will be 3D design aspects such as the feeder screw and associated drive components that will need to be produced for the final product. Similar to the feed screw, many general components are available as open source models that can be printed. This includes connecting components used in the drive line such as grooved pulley to mate to a grooved belt. 3D printing could also be used for a angled trough design for feeding into the feed screw. The versatility of 3D printing and the open source modeling community means than many aspects of the physical design will have been mostly implemented and are available to be modified for our application.

Background

All team members have experience with embedded programming from taking the Assembly and Interface & Digital Electronics courses at RIT. These courses provided familiarity of programming embedded devices and how they can be interfaced with peripheral devices. Some of the group members have done personal embedded system projects and have taken real time systems courses to better understand the aspects of task scheduling for embedded devices.

Outside Contributors

Open Source tools are an incredible resource that allow designers and engineers to avoid reinventing the wheel in order to build the car on top. In this project, open source models will be used and the authors credited per the license applied to said design.
For mechanical aspects of the design, outside help will be accepted from a very close friend, Scott Eisele of the Mechanical engineering dept. with considerable ability in 3D modeling.

V Constraints and Considerations

Extensibility

This project can be extended for other applications. The Automated Cat Feeder can be used to feed various kinds of animals, or it could be repurposed to monitor people's feeding habits. Additionally, this project is been designed with scalability in mind from the beginning. It can easily accommodate many additional pets and/or feeders.

Manufacturability

The nature of software means that the designs are easily reproduced. Difficulty arises in the physical construction. Models for printing will need to be created with real constraints in mind. This means that certain round components will need to be split in half for printing. 3D printing also requires certain wall thicknesses and scaffolding to support certain pieces. These issues are very common and easily worked through.

Reliability

Due to the nature of a pet feeder, the device must be extremely reliable and should never fail to feed. This means that any code written will be tested for long term continuous runs. The dispensing unit will also need to run without binding or jamming. Utmost care will go into the reliability of the drive unit, as the unit may be awake and running but no food will be able to come out due to a jam.

Intellectual Property

In searching for patents of a similar design to the screw, one patent, US 6401657 B1, was found to be similar but not nearly identical in form or purpose. This device uses a very common auger feed delivery system but the bowl is integrated in the unit. It does not extend to the extended functionality of the design.

VI Bill of Materials

Table 6: Bill of Materials

Component Name	Cost ($)	Our Cost ($)	Notes
Android OS Device	200.00-500.00	0.00	Already Owned
Raspberry Pi Zero W	10.00	0.00	Already Owned
Micro SD Card	7.80	0.00	Already Owned
ESP8266 Dev Board	8.79	0.00	Already Owned
Stepper Motor	12.99(x2)	25.98	
Stepper Motor Driver VAA A3967	5.98(x2)	13.96	
.25” 2x4 Maple plywood	14.48	14.48	
Screws and hex nuts	11.82	11.82	
PVC Tee	1.22(x2)	2.44	
Metal Food Bowl	2.13(x2)	4.26	
3D Printed Feed Screw and gears	1.20 (x2)	2.40	RIT Construct 40g at 0.03/g for each set
MFRC522	7.98(x2)	15.96	
5kg Load Cell with HX711 Driver	12.99(x2)	25.98	
Dual output PSU	11.95(x2)	23.90	
	Total Cost	**141.18**	70.59 a body

VII Testing

Software Experiment Results Table

Test Case Name:		Master Feeder & App Communication			**Test ID#:**	MF-01
Description:		Checking if the smartphone app is able to make a connection with the Master Feeder server and make http requests to add, remove, or update data. This is to meet Engineering Requirements A, B, C, and D			**Type:**	☑ white box ☐ black box
Setup:		Configure Master Feeder to broadcast localized wireless network and to be connected to users local network.				
Step	**Action**	**Expected Result**	**Pass**	**Fail**	**N/A**	**Comments**
1	Connect to feeder via app on user's local network	Connection is established	x			The Master feeders dual connection operated well enough to permit the android app to find and communicate with the feeder
2	Connect to feeder via app on Master Feeders network	Connection is established	x			Worked as well as the user local network
3	Add Pet to Database	Pet is added	x			Server responds to requests for adding
4	Remove Pet from Database	Pet is removed	x			Server responds to requests for removing
5	Update Existing Pet in Database	Pet info is changed	x			Server responds to requests for updating
Overall Test Result: successful						

Hardware Experiment Tables

Test Case Name:		Dispenser Operation	**Test ID#:**			HW-01
Description:		Verifying mechanical effectiveness of food dispenser. This is to meet Engineering Requirements F, G, and H.	**Type:**			☑ white box ☐ black box
Setup:		Put together dispenser assembly and acquire dry food for testing				
Step	**Action**	**Expected Result**	**Pass**	**Fail**	**N/A**	**Comments**
1	Can dispense food in small amounts	Food is dispensed in small amounts	x			Load sensor was capable of measuring food accurately within in a 10th of a gram for accurate dispensing
2	Can dispense food in large amounts	Food is dispensed in large amounts	x			Can load larger amounts into pet bowl
3	Can operate continuously for a period of time	Continuous operation for a defined period of time	x			Is capable of continuous operation provided it has a power source and food to dispense to a pet. Auto unjamming feature works well in case food gets stuck
Overall Test Result: successful						

Test Case Name:		RFID	**Test ID#:**			HW-02
Description:		Verifying RFID detection to meet Engineering Requirement I.	**Type:**			☑ white box ☐ black box
Setup:		Connect RFID module to relevant feeder i2c pins				
Step	**Action**	**Expected Result**	**Pass**	**Fail**	**N/A**	**Comments**
1	Can recognize the tag	Tag is recognized	x			RFID tag can be read and be made sense of efficiently by the device
2	Can read tag from adequate distance	Tag is read from an adequate distance	x			Provided the pet is hungry and close to its bowl the feeder will be able to detect and dispense food to the pet
Overall Test Result: successful						

Embedded Experiment Tables

Test Case Name:		Master Feeder			**Test ID#:**	MF-02
Description:		Verifying functionality of peripheral devices and communication of the main feeder unit. This is to meet Engineering Requirements B, C, G, and I.			**Type:**	☑ white box ☐ black box
Setup:		Connect peripheral devices to appropriate pins on master feeder Raspberry Pi				
Step	**Action**	**Expected Result**	**Pass**	**Fail**	**N/A**	**Comments**
1	Can make use of RFID data	Read and process RFID tag	x			Is able to parse the RFID number and cross reference it with pet profiles to determine if the pet can feed
2	Can drive stepper	Stepper motor is activated and driven	x			With use of the Easydriver motor controller the feeder was able to adequately control the driving speed and direction of the stepper. The draw of current from the stepper is a little higher than what the controller can dispense, causing minor overheating.
3	Creates ad-hoc network on boot	Ad-hoc network is created on startup	x			Network starts up successfully
4	Connects to home network on boot	Connects to home network on startup	x			It is able to connect to user network after booting up
5	Available for app on boot	Makes unit detectable for mobile application on startup	x			The network is started on bootup and is available to the user app
6	Can accept HTTP on boot	Enables HTTP messages send/receive functionality on startup	x			Servers http endpoints are available and can process app and supplementary feeder requests
Overall Test Result: successful						

Test Case Name:		Supplementary Feeder			**Test ID#:**	SF-01
Description:		Verifying functionality of peripheral devices and communication of the supplementary feeder unit. This is to meet Engineering Requirements C, G, and I.			**Type:**	☑ white box ☐ black box
Setup:		Connect peripheral devices to appropriate pins on supplementary feeder esp-12s board				
Step	**Action**	**Expected Result**	**Pass**	**Fail**	**N/A**	**Comments**
1	Can make use of RFID data	Detect and process RFID data	x			Supplementary feeder is able to operate its other devices while checking for RFID input and parse it
2	Can drive stepper	Stepper motor is activated and driven	x			See comment for master feeder
3	Can send or accept HTTP on boot	Enable HTTP message send/receive functionality on startup	x			It is able to communicate with the master feeder on its network to transmit RFID data and request permission to connect
Overall Test Result: successful						

VIII Risks

Master and Supplementary Feeder System Controllers

Controller selection involves finding controllers that are cost effective while also meeting the needs of the project. This means having adequate peripherals and available development resources. This is to ensure speedy development and usability in the project. This can be difficult to pinpoint due to the many options available to ensure cost mitigation to save resources for the more risky aspect of the project where our team lacks experience. This can be alleviated with thorough research of available controllers to ensure the most cost effective choices.

User Interface

The risks associated with developing the user interface mainly have to do with the overall unfamiliarity with Android application development and limited experience with networking and databases. A thorough amount of research and documentation available online will be used to alleviate these risks.

Dispensing Mechanism

As this area of the design is outside the standard Computer Engineering curriculum, none of the team members are particularly skilled in modeling or mechanical design. This means that a considerable amount of self learning and outside contributions from persons of distinguishable skill in the area will be necessary. However, the scope of this design is not too far outside the abilities of competent “maker” and is not an unreasonable task.

IX Management Plan

Milestone Schedule

Task Category	Deliverable	Scheduled Completion Date	Team Member	Modified Completion Date	Comments
Master Unit Controller	Raspberry Pi HTTP server	January 30th 2018	Matthew Smith	February 9th 2018	Completed
	Follower Unit Communication	February 15th 2018	Matthew Smith and Chris Ranc	February 19th 2018	Completed
	App Communication	February 10th 2018	Shabab Siddiq and Matthew Smith	February 21st 2018	Completed
	Stress Testing	April 5th 2018	Chris Ranc and Matthew Smith	April 19th 2018	Completed Pushed back due app development
Follower Unit Controller	Picoweb (HTTP) Server Implementation	January 31st 2018	ChrisRanc	February 17th 2018	Completed
	Btree Database Integration	February 15th 2018	Chris Ranc		Completed but redacted during testing for memory conservation
	Master Network Detection	February 15th 2018	Matthew Smith and Chris Ranc		Completed
	Peripheral Task Scheduler	April 5th 2018	Chris Ranc	April 9th 2018	Completed
	Stress Testing	April 5th 2018	Matthew Smith and Chris Ranc	April 19th 2018	Completed Pushed back due app development
Android Application	Basic App Functionality	January 30th 2018	Shabab Siddiq	February 21st 2018	Completed
	Basic layout	January 30th 2018	Shabab Siddiq	February 19th 2018	Completed
	Full Layout and Functionality	March 8th 2018	Shabab Siddiq	April 19th 2018	Completed
Hardware Design	Physical Body	March 30th 2018	Matthew Smith	April 15th 2018	Completed
	RFID	April 5th 2018	Matthew Smith and Chris Ranc		Completed

Gantt Chart

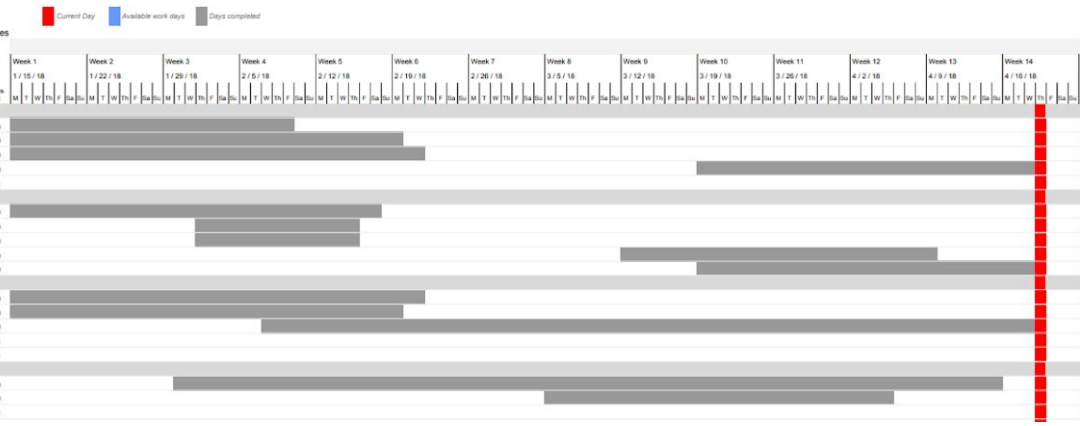
[ACF] Project Schedule

Project Start Date: 1/15/2018 (Monday)

Today's Date: [1] 4/19/2018 (Thursday)

Display Week: 1

	Task	Start	End	Cal Days	% Done	Work Days	Days Done	Days Left
1	**Master Unit Controller**							
1.1	Raspberry PI HTTP Server	Mon 1/15/18	Fri 2/09/18	26	100%	20	26	0
1.2	Follower Unit Communication	Mon 1/15/18	Mon 2/19/18	36	100%	26	36	0
1.3	App Communication	Mon 1/15/18	Wed 2/21/18	38	100%	28	38	0
1.4	Stress testing	Mon 3/19/18	Thu 4/19/18	32	100%	24	32	0
1.5				1	0%	0	0	1
2	**Follower Unit Controller**							
2.1	Picoweb (HTTP) Server Implementation	Mon 1/15/18	Sat 2/17/18	34	100%	25	34	0
2.2	Btree Database Integration	Thu 2/01/18	Thu 2/15/18	15	100%	11	15	0
2.3	Master Network Detection	Thu 2/01/18	Thu 2/15/18	15	100%	11	15	0
2.4	Peripheral Task Scheduling	Mon 3/12/18	Mon 4/09/18	29	100%	21	29	0
2.5	Stress testing	Mon 3/19/18	Thu 4/19/18	32	100%	24	32	0
3	**Android Application**							
3.1	Basic app functionality	Mon 1/15/18	Wed 2/21/18	38	100%	28	38	0
3.2	Basic app layout	Mon 1/15/18	Mon 2/19/18	36	100%	26	36	0
3.3	Full layout and functionality	Wed 2/07/18	Thu 4/19/18	72	100%	52	72	0
3.4				1	0%	0	0	1
3.5				1	0%	0	0	1
4	**Hardware Design**							
4.1	Physical body	Tue 1/30/18	Sun 4/15/18	76	100%	54	76	0
4.2	RFID	Mon 3/05/18	Thu 4/05/18	32	100%	24	32	0
4.3				1	0%	0	0	1



X Reflection

The first and most obvious change from the original design was the actual chassis construction. Early design discussions concluded on 3d printing a complete chassis for the feeder. As design continued and the chassis was put off, it was clear that a cohesive 3d printed chassis would be a lot more design work than initially intended, as well as many hours in the 3D printing lab working and reworking the design. A decision was made to move away from the 3D printed chassis and look into laser cut wood as an easier material to design and work with. This change had several benefits over the 3D printed chassis. The biggest change was that actual creation of the parts could be done over the course of one hour in the laser cutter for two feeders, rather than many days of back and forth to the 3d printing lab inspecting and starting over when prints were unsuccessful. Second, the actual design work was much more simple. Laser cut plywood construction mostly relies on properly positioned socket design with some basic fastener work to hold it all together. Each component was designed in a 2D sketch, given a 3rd, constant dimension, and test fit in a virtual assembly. Then the parts could be exported via PDF, inserted into a vector design program such as inkscape, laid out, and given to the laser cutter to be done all in one session. The same design for the 3D printing method would have required several disassembled parts of the chassis that could fit on the printer's relatively small printing deck, then put together for the main unit. Finally, we were unsure about the actual strengths component of the 3D prints. To make parts strong, a high density print would have been required, using up much more material and time. Plywood is a relatively constant material that can be inspected in its full form in the store and material properties are known before even going to the laser cutter, reducing the possibility for multiple revisions.

The second important aspect of the design that we would revisit if we could was the dispensing assembly. Overall, the dispenser assembly worked as intended but it had a couple issues that could have been easily revised, including jamming, stalling, and backfill from the wrong side of the dispenser. First, the off-the-shelf PVC unit would have been better off replaced by a unit of our own design. In the two units, we used two slightly different PVC T couplings. This made it so one feeder was significantly more reliable than the other. Secondly, due to the fit of the auger and the coupling, there was plenty of room for food to spill out on the wrong side of the feeder after several software unjamming attempts. This would mean after prolonged use, the feeder would need to be manually cleaned out or else jams would become far more severe. A purpose built design could have easily accommodated for that, having it fit the auger better in the rear as well as having a part in the front to suspend the front of the auger in a centered position rather than resting on the bottom of the coupling. Lastly, a self designed feeder tube could have accommodated an angled hopper connection and a wider area for food to spill into the auger, eliminating much of the jamming that was caused by pieces of food hitting the inner edge of the T coupling.

The motor selection could have been better as well. The stepper was selected before we decided to integrate the load sensor. As the most important aspect of choosing the stepper was the precision rotation, the load sensor made it so we treated it as a normal motor. A simple DC motor could have done the job more effectively with less load on the motor driver, as the H-Bridge supplies power to the stepper at all times to keep in place, also known as hold strength for steppers. This is nice for preventing cats from tampering with the device, but less so for power usage and longevity of the parts.

The Android application could have been improved as well. At the time of the layouts were made, more research needed to be made regarding how layouts are usually designed. The realization was made later into the development that the layouts needed to be redesigned for a more efficient for user-friendly user-interface. The application development could have been started much earlier to allow for more time for bug testing and implementing more features and finessing a much more effective user interface. With that being said, however, the application was successfully implemented, meeting all of the engineering requirements, and the user interface implementation was an overall success.