ECE 2799

Electrical and Computer Engineering Design

Homework 6: Final Report

Professor Stander & Michalson

Authored by

William Tyrrell, Adam Spencer, Alexander Galvin

5/3/23

Table of Contents

3.3.7 Convert ADC Measurements to Text 28

1. Introduction

1.1 Problem Statement

We were tasked to design a product that measures some physical characteristics of its environment, processes those measurements, and takes some action based on the result. We were restrained to a budget of $100 for components to build and prototype our system. Given these requirements and constraints, we decided to design a trash bag weighing system. Our design was intended to be integrated inside a trash can and measure the weight of the bag as it is filled. Our system was intended to display the weight of the bag and indicate if the bag is in danger of ripping. We decided to call this product the Weigh Bin.

The goal of our market research was to determine the interest level for our product and if similar products existed. We researched trash cans at different price points to see if there were products with similar functionality to ours. We also looked into the trash bag industry because we needed to find information on which bags are most purchased in the US. This allowed us to determine which size bag we should focus on based on the market. Additional research was needed to determine customer requirements and interests so we sent out a survey to gauge product preferences. Based on the customer requirements derived from the survey we were able to set our product requirements and specifications. From these specifications, we conducted a multitude of tests and came out with a functional prototype.

Our product was marketed as a quality-of-life improvement. While it may not be a necessity, it helps to make peoples' lives easier. It can be frustrating when you take a trash bag out of the can and the bag is ripped. In most cases, you probably could not tell until you got to that point. Our product aims to prevent this problem from ever happening and it removes a simple frustration from everyday life. Our product may also be helpful to disabled or elderly people. It may be difficult to lift a heavy trash bag for these groups of people, and our product allows the user to see the weight of the bag, that way they can judge when to take the bag out before it gets to the point of being too heavy. For these reasons, we believe our product has market appeal. We conducted many value analyses in relation to our competitors, possible components, additional features, and return on investment. This report goes over the entirety of the process undergone to design the first Weigh Bin prototype.

1.2 Market Research

The desired demographic that this product was intended to appeal to is the general public, as almost everyone uses trash cans, as shown in Figure 1. In the ever-growing, technologically advancing, world that we live in, a majority of people are happy to see improvements to any of the things that they interact with on a day-to-day basis. We felt as if something like this would be popular with the masses, simply because it improves upon something that everyone uses all the time, making their lives slightly more technologically intertwined.

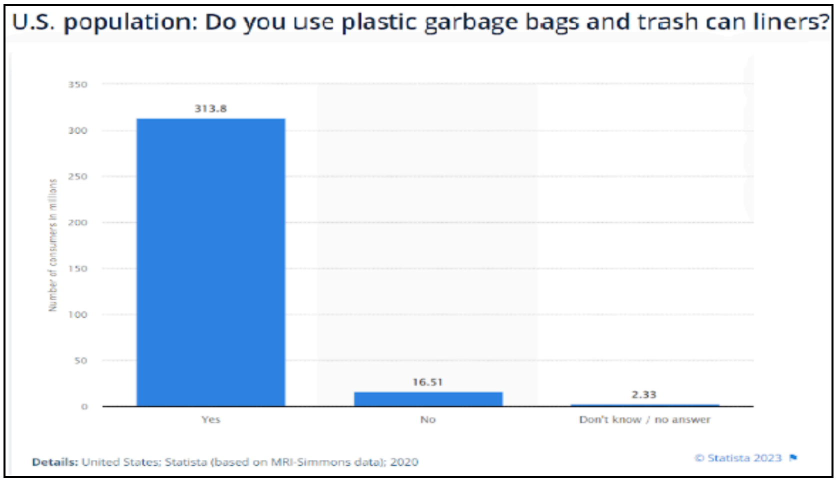


Figure 1: Use of plastic garbage bags/can liners in the U.S.

In addition to the Weigh Bin being a pleasant quality-of-life improvement for many people, it also has the potential to help specific demographics of people that may struggle with the task of dealing with a full bag of trash. Take the elderly, for example; many older people tend to have a harder time lifting heavy objects, as strength decreases with age. Having the capability to always know the weight of a trash bag before attempting to take it out could be of major assistance to people that struggle with problems such as these. In addition to the elderly, disabled people or those who have recently undergone surgery could also use this product for the same reasons stated before. Imagine someone who recently had surgery on their arm and their doctor told them not to lift anything heavier than 15 pounds for 2 months; that person would be able to view the current weight of their trash bag, preventing any possibility for additional injury. Another example is that the city of Worcestor will not accept trash bags weighing over 30 pounds; our product could be used so people don't need to weigh their bags outside of the can. This invention is by no means revolutionary, but is something that no one has ever thought to do. This could help a multitude of people - either simply by making their day easier or preventing them from injuring themselves.

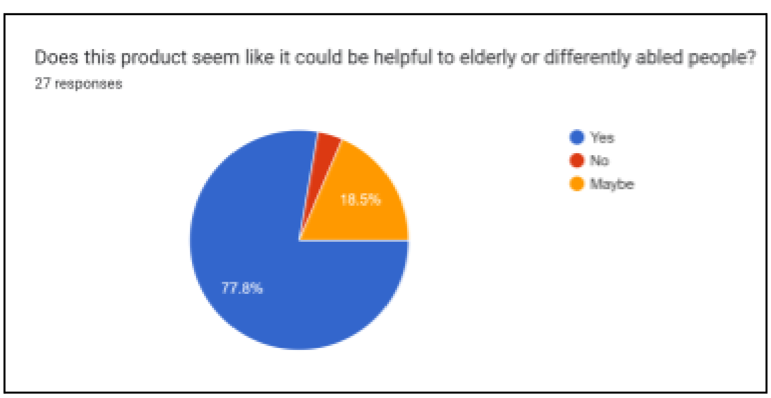


Figure 2: Percent responses for elderly/disabled assistance

As can be seen here, an overwhelming majority of people thought that this would be helpful to the demographics previously stated, and almost everyone else thought it had the potential to do so, with only one person saying that they think it wouldn't be helpful at all. This will be discussed in more detail later in the "Recommendations" section.

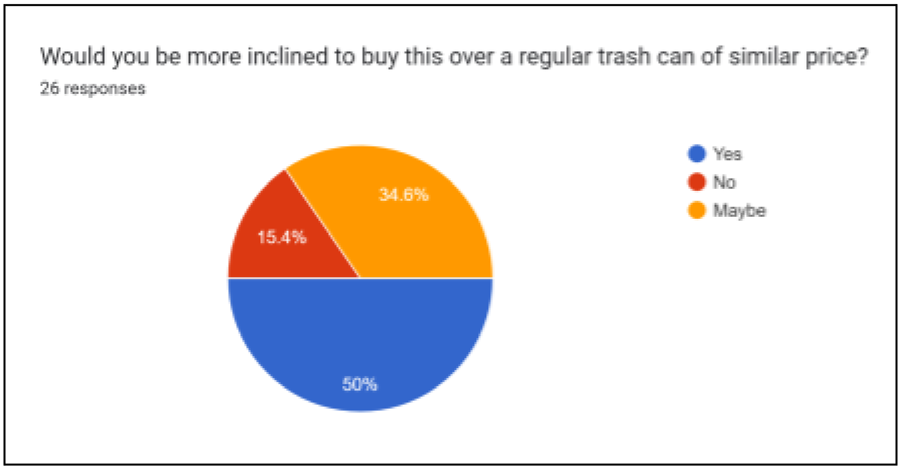


Figure 3: Percent responses for smart trash can vs. regular

After surveying a group of people about our product, 22/26 of the participants stated that they would at least consider purchasing this, in place of a regular type of trash can. This proved that the market for this product exists, and many people would be interested in purchasing it if it were to show up in a store next to a regular trash can.

On top of the previous findings, we asked the participants of the survey how much money, in addition to the price of a regular trash can, they would be willing to spend on the included features.

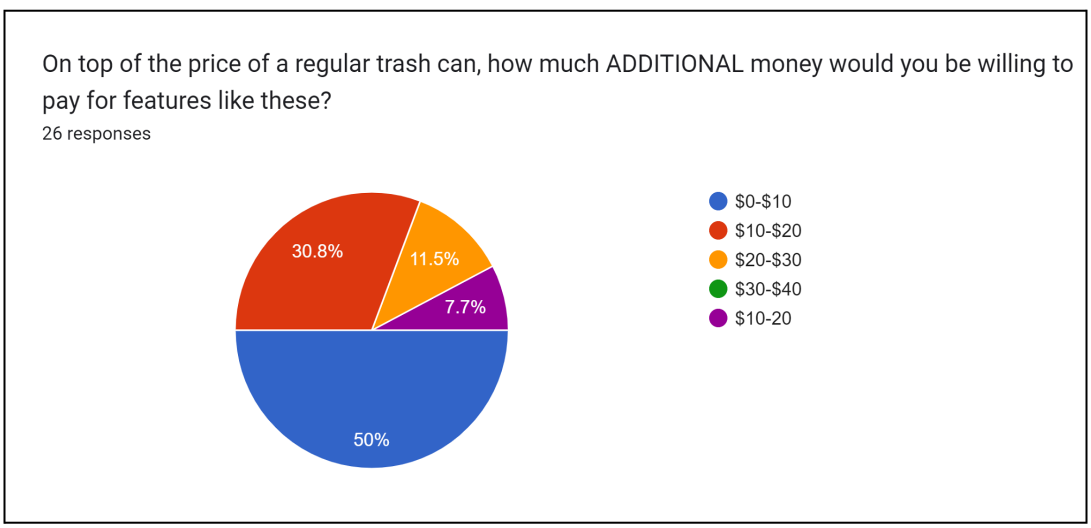


Figure 4: Percent of responses for additional price

As shown in Figure 4, no participant that responded was willing to pay any more than $30 for the features that this product offers. While this may be possible if it were ever to be a consumer product, we determined that it was highly unlikely that our initial prototype will cost us less than $30. This determination was correct as the Weigh Bin cost approximately $68 to make our first prototype. The costs necessary for testing and implementing different ideas and designs were more than thirty dollars for the first attempt, but if this were to be manufactured with custom parts and an efficient process, an additional price tag of $10-$20 seems very reasonable for sale. Also, note that the purple and red sections of the graph were the same price range; this was due to the fact that we allowed for an "other" option for the respondents to put their own price ranges and a few people put one of the ranges that was already listed.

Relative to other trash cans on the market, a $10-$20 increase is not extreme, especially considering the additional features. After browsing a few popular internet shopping websites, we collected some data regarding the average prices of trash cans, metal, and plastic, based on the highest-rated 13-gallon bins. The results that were found are shown below.

Website	**Material**	**Number of Products Viewed**	**Approximate Average Price**
Amazon	Metal	14	$88
	Plastic	10	$45
Walmart	Metal	15	$80
	Plastic	13	$22
Target	Metal	15	$113
	Plastic	9	$33
Totals	Metal	44	$94
	Plastic	32	$32

Figure 5: Average prices of plastic and metal trash cans from various websites

There were several products that we found to be part of the smart trash can market. These include the Simplehuman 15.3 Gallon Smart Trash Can, the iTouchless 13 Gallon Sensor Trash Can, and the Townew T1 Smart Trash Can. The Simplehuman 15.3 Gallon Trash Can was the only one of its size on the market that was voice-activated, other models were foot-activated via a pedal [4]. The iTouchless 13 Gallon Sensor Trash Can was motion activated and was valued at approximately $80 [1]. The Townew T1 Smart Trash Can was approximately $50 on their website but Amazon has listings from $70 to $400. This trash can also only held 4.4-gallon custom bags. This product also featured a bin that can self-seal trash bags [4]. Other trash cans of similar size do not contain automated features. Through market research, it was found that no other household trash can currently on the market had automated features similar to the ones we are including in our design.

Some smart trash can designs were featured as being part of a system such as the design proposed in this paper on an "Electronic System Comprehensive Design Experiment-Intelligent Trash Can Design" [6]. In this paper, a system was proposed where the bin could detect overflow, but this bin would be part of a system and most likely be used in a commercial setting to improve the efficiency of waste management services [6]. It was a similar case in the "Smart Trash Can System with Ultrasonic Sensor and Flame Detector Using Arduino" in which several sensors including a level sensor and moisture sensor, but this proposed design would be part of a system and therefore out of the household waste bin market [5]. These proposed trash can systems were able to detect overflow and were part of systems, not individual trash cans. This paper titled, "Design of Intelligent Trash Can Based on STC89C54 Microcontroller Unit", proposed a trash can that could automatically open its lid depending on the distance of a person, had the temperature and ultrasonic sensors inside the can to detect and alert the height and temperature of the trash if needed, and was solar powered [3]. These products had several automated features and sensors but none of them were on the market nor did they indicate in their design the weight of the trash and whether the bag was in danger of ripping.

Based on market and academic research we concluded that there were no designs of waste bins that contain features that would measure the weight of the trash inside of the bag and indicate whether or not that bag was going to rip. Because of this, it was evident that our product was the only one in the market with similar features to what we planned to offer.

2. Product Requirements

2.1 Customer Requirements

To better understand the needs of customers a survey was conducted in which the participants were prompted to answer questions and rate specific features. The survey was sent to the classmates and friends of the authors, thus identifying the questioned demographic as mostly college-age students within the Electrical and Computer Engineering major. A total of 26 responses were counted. Useful features and feedback were delivered and carefully examined. The questions posed in the survey included the following: preference between the proposed smart trash can versus the currently available garbage cans if both products were of a similar price, the frustration experienced over the ripping of a trash bag, what size trash bag was most commonly used, willingness to spend additional money, whether the product seemed helpful to certain demographics of people, and suggestions for additional features. Participants were also asked to rate these five features in order of importance: the ability of the garbage can to measure the weight of the trash inside of it, the indication and display of the capacity of the garbage can, the ability to warn the user if the bag is in danger of ripping, having access to a roll of bags on the side, and the accessibility of the smart garbage can.

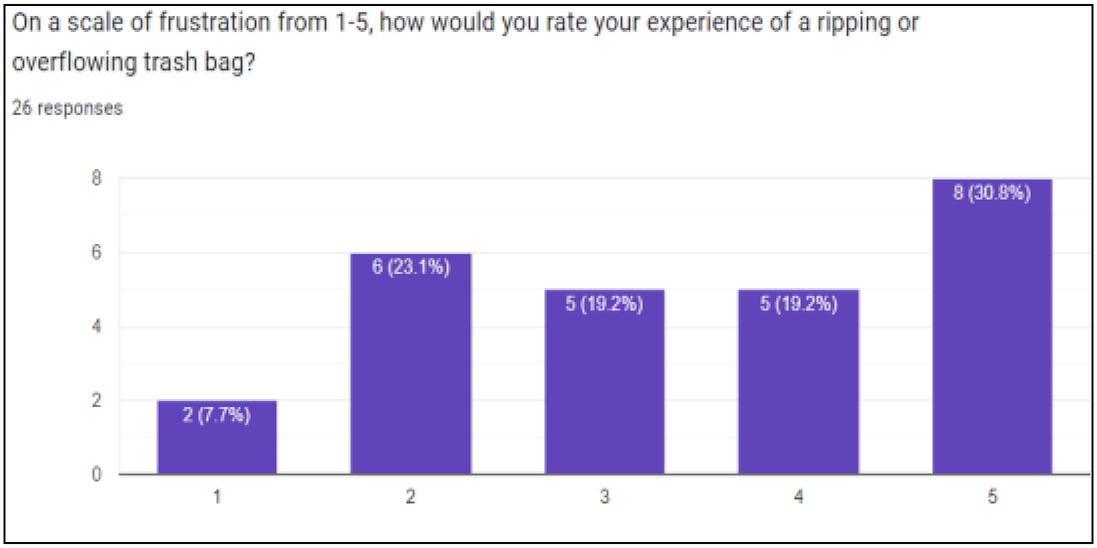


Figure 6: Percent responses for frustration regarding bag complications

Because participants indicated a high level of frustration if the bag were to rip and that it was the most important feature overall, it was reasonable to assume that it would be the most important feature of the Weigh Bin. Figure 6 indicates participants' ranking of frustration at the idea of the trash bag ripping.

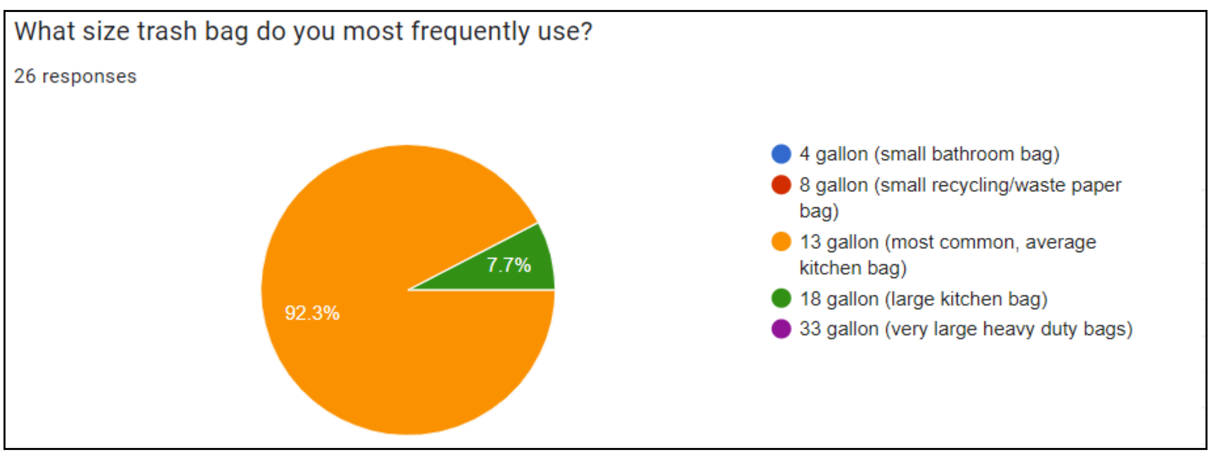


Figure 7: Percent responses for trash bag size

The overwhelming majority of survey participants indicated that they use 13-gallon garbage bags most often, as shown in Figure 7. We concluded that it was reasonable to say that the product will most likely cater to 13-gallon bags. This will be expanded upon in the “Recommendations” section later in the paper.

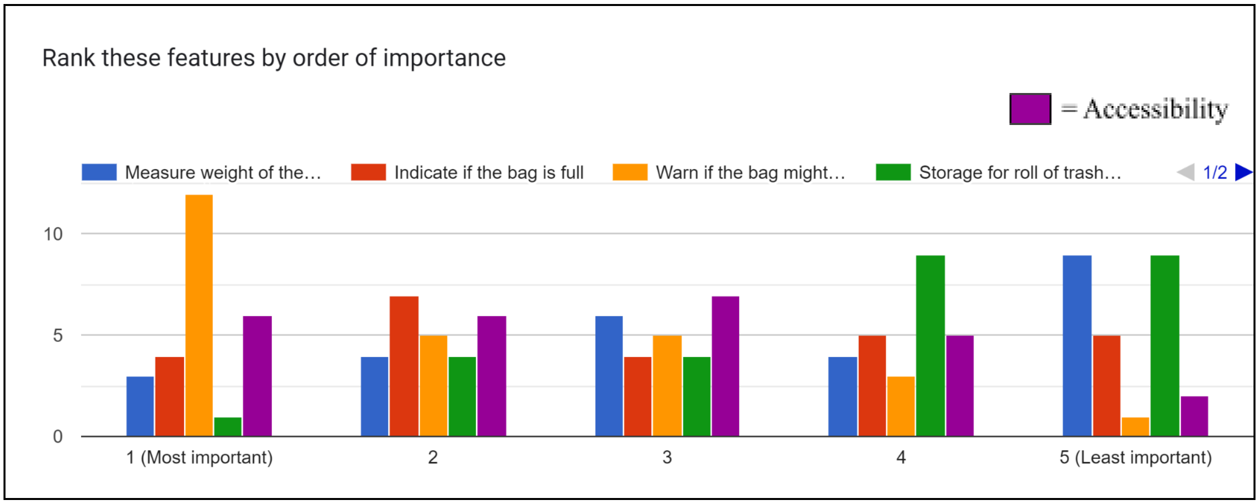


Figure 8: Responses for ranking the importance of features

Based on the five core features that we planned on including, we asked the respondents of the survey to rank each of the features' importance. The overwhelming majority of people stated that warning if the bag would rip was the most important feature to them while measuring the weight of the bag and storage for additional trash bags were less important. The accessibility and capacity indication seemed to be somewhat important to most respondents. While we agreed with the responses to the survey, our method for determining whether or not the bag will rip was dependent upon the weight of the bag, making that a key component of the design.

When customers were asked to give feedback on the survey and list additional features of particular interest, responses included: a trash compactor, making the bin battery-operated, an indication that the bag reached capacity via a "jingle", the bin being nicely scented, looking fashionable, the ability for it to take itself to the curb, and a way to handle sharp objects.

Based on the responses, time constraints, and overall scope of the project it was determined that some features could not be added. Adding a trash compactor or making the trash able to take itself to the curb are both extremely complex additions, which did not necessarily relate to the intention of our design. A trash compactor negated the purpose of the product and would ultimately make it more likely that the bag would rip. Automating the movement of the trash can was determined to be far beyond our current constraints. On the other hand, most of the other recommendations appeared to be reasonable and some were added. The purpose of the Weigh Bin was to indicate when the bag was full and when it was in danger of ripping. To fulfill both of those requirements the weight of the trash inside the bag was calculated, in this way, the user was shown the weight of the bag and was given recommendations by the trash can for when to take it out. Thus it came as a surprise that most participants did not find this feature particularly useful. The weight and capacity of the bag go hand in hand. The weight indicates when it's time to take out the trash and therefore implies fullness.

2.2 Product Requirements

The product requirements for the Weigh Bin were determined via a customer survey and market research on similar products. In this way, the product specifications are derived from customer requirements. It should be noted that the points in Figure 9 were derived from our initial design, which has been changed; The motion-sensing portion no longer applies.

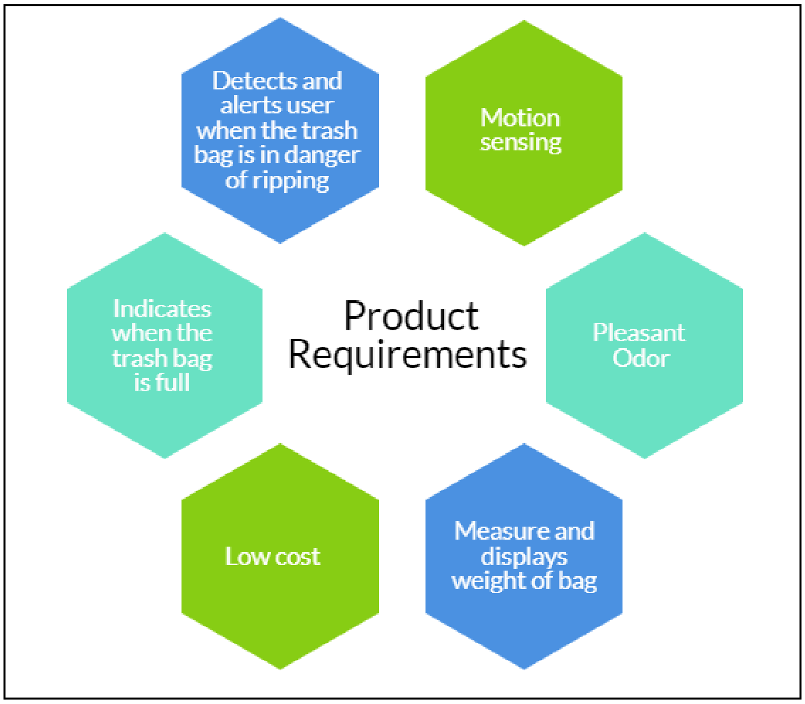


Figure 9: Product requirements

The primary feature of the Weigh Bin was the ability for it to detect when the trash bag was about to rip. This was the most important feature because it addressed the problem of the bag ripping and causing a mess as indicated by those who participated in the survey. The ability of the trash can to detect and alert when the bag is going to rip serves to benefit the quality of life of the customer as this feature can prevent a mess.

Along with the ability to detect when the bag is going to rip, the ability for the trash can to detect when the bag was full serves to better the lives of the customers and remove any uncertainty they may have regarding how much space is in the trash can. This feature was also intended to help them to determine when they should take it out.

Designing a product that displays the weight of the trash bag serves to prevent injury in elderly or disabled individuals. Because the elderly or disabled may struggle to lift trash bags exceeding a certain threshold they must know how much the trash weighs. Once they see the weight of the trash, this information indicates to them whether they continue to fill the waste bin or dispose of the trash inside. According to the survey, participants indicated that the product would be of use to the elderly and disabled. Our research and testing suggested that the average 13-gallon trash bag holds up to 15 pounds without ripping, with some of the more expensive, name-brand bags being able to withstand weights up to 50 pounds [7]. This is important to us because, when considering parts to include in our design, we had to find a pressure sensor that could detect forces up to those limits, to ensure that the readings were accurate and that nothing would break. We did our own additional research regarding the strength of some trash bags, due to the fact that it was difficult to find any scientifically backed information on trash bags' durability.

A popular feature among the smart trash cans found was the ability of the products to detect and react to the presence of a user. Some products, such as the Simplehuman waste bin, were voice-activated and the trash lid opened when prompted by the user [2]. Other products, such as the iTouchless were motion sensing while something like the Townew T1 Smart Trash Can detected individuals with an infrared sensor [1][4]. For the accessibility aspect of the product, constrained by the scope of the design period, the product does not detect the motion of a person nor does it open the lid automatically.

Many participants in the survey indicated that something should be done about the odor of the trash can. The iTouchless trash can, a competitor, contains an odor filter. It was in the best interest of the customer that the trash can mask the scent of its contents. In this way, the lid was sealed tight and emitted a pleasing fragrance after the lid closed to mask the scent of its contents.

An important feature of the product was that it was of high quality yet low cost. According to Amazon, the iTouchless 13-gallon trash can was about $90, a 13.2-gallon Simplehuman trash can was valued at $140, and a Townew T1 Smart Trash Can costs $120 for a 4.1-gallon can and $400 for a 4.4-gallon can. It was determined that it would be ideal if the Weigh Bin was within the budget provided and competitive with other products on the market. The cost of the garbage can, post-manufacturing, was intended to range between $50 and $100. Based on the survey, customers indicated that they would pay up to $20 extra for the proposed product over a similar product because of its additional features.

The final product requirement was that the garbage can was pleasing to the eye. From a marketing and logical perspective it would stand to reason that a garbage bin should conceal wires and circuit boards as well as blend in with the general environment. Figure 10 shows a basic design of the product.

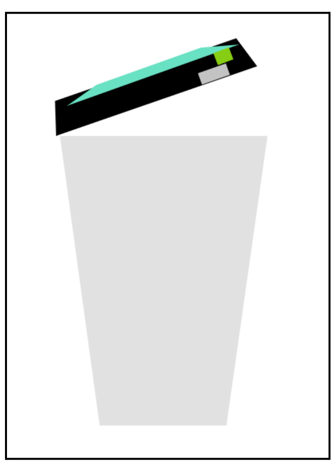


Figure 10: Concept image of design

2.3 Product Specifications

Based on product requirements that were developed using our market research and customer survey results, we developed our first idea of which components should be included in the design. The product specifications for our trash can were determined and our first idea of components to include in the design are the following:

- Pressure/weight sensor (kept)
- Motion sensor (removed)
- Lid Mechanism (changed)
- Scent spray dispenser (changed)
- Display Screen (kept)
- ON/OFF switch (changed)
- Buttons and/or scroll wheel (removed)
- Microcontroller (kept)

Keep in mind, everything discussed here was our original ideas for the design, the changes and final decisions will be discussed in later sections. The first major part of our product requirements is designing a system that measures the weight of the bag and displays this information. To do this, we decided to use pressure/weight sensors to detect the weight of the bag. Not fully understanding how these sensors would need to be mounted, we thought they would be located at the top and bottom of the bin to give different readings related to the forces on the bag. The location of the sensors will change later, but the sensors themselves were crucial to our product and needed for our design to function.

For our product to display the weight of the bag and determine if it is in danger of ripping, we would need to process the information we obtained from our pressure/weight sensors. We needed some sort of microcontroller to do this. This microcontroller took information from the sensors and used it to implement our product requirement of determining if the bag is in danger of ripping.

The information from this microcontroller needed to be displayed to the user, so to implement this we used a display. The display was attached to the can and showed the current weight of the bag and if the bag is in danger of ripping. This also implemented our product requirement of indicating to the user when the bag is full.

The next part of our product requirements was designing a more accessible trash can. Based on our market research, our first idea to improve accessibility was a trash can that opens using a touch-free gesture. To achieve this, we planned to implement a motion sensor to detect when the user is gesturing to open the can. This would actuate a mechanism that would open the lid of the can. This would allow us to implement our product requirement of accessibility features. As will be discussed later, this feature was not included in the final prototype design, due to the desire to reduce the cost of the product, but it was replaced with an alternative that also provides accessibility.

The final product requirement we needed to implement is a scent spray mechanism. We intend to design an apparatus that will spray a nice scent in the can each time the user chooses to open it. To achieve this we planned to use the motion sensor that would detect when the lid was opened and the spray was sent into the can. This would allow us to implement odor control in our design as specified in our product requirements. This design element was also altered in favor of a more simple and user-friendly change, which will be discussed later on.

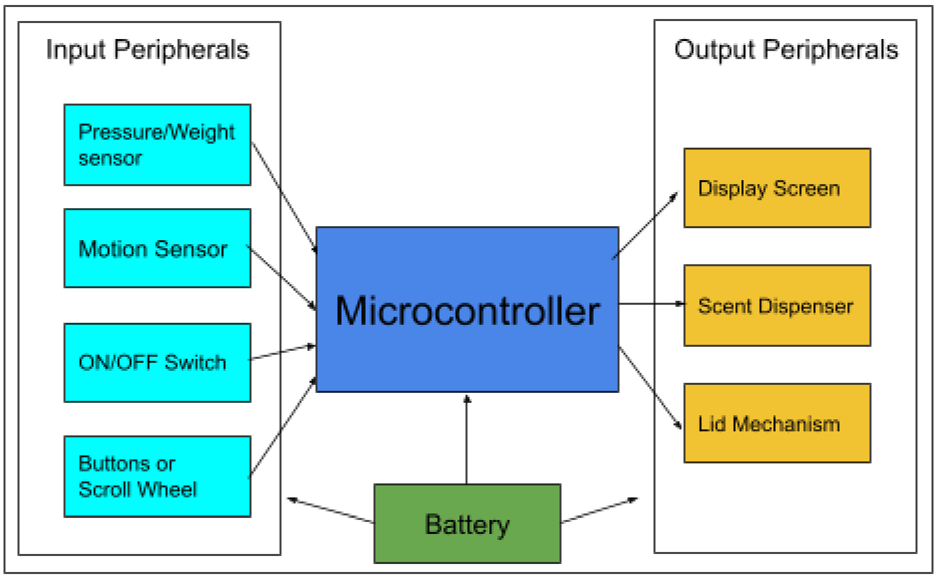


Figure 11: Original block diagram for the design

The figure above demonstrates our original idea of the basic flow of our design. The following few paragraphs depict what our original intentions for the design were, not what it actually ended up being. For the input peripherals, we begin with the ON/OFF switch. This would toggle our system ON and OFF so that the user can choose when it is functioning. The next input peripheral is the motion sensor; this would be used to detect when the user is gesturing for the can to open. The information from this sensor was fed to the microcontroller. Then we have pressure/weight sensors, which would measure the weight and stretch of the bag. The information from these sensors is then sent to the microcontroller. Finally, we have the buttons and/or scroll wheel. We would use these so that the user can interface with the output display. The inputs from these buttons/scroll wheels are also fed into the microcontroller.

The next major part of our system is the microcontroller and the battery. The battery would be connected to all components requiring power in our system. The microcontroller connects the input peripherals to our output peripherals. The microcontroller would take information from the motion sensor and then send this information to the lid mechanism. It took information from the pressure and weight sensors, calculated the weight of the bag, determined if the bag is full/in danger of ripping, and then sent these results to the output display. The microcontroller also takes user inputs from the buttons/scroll wheel and updates the display/bag calculations to reflect the user's inputs.

The final part of our system is the output peripherals. The most important of these is the display screen. This screen is intended to display to the user the weight of the bag if the bag is full/in danger of ripping. It would also display to the user the maximum weight they are setting until the system warns that the bag is full. The next output peripheral is the lid mechanism. We intended to design a mechanism that when the user triggers a motion sensor, the lid of the can will automatically open and remain open for a set amount of time. Finally, we intended to include a scent dispenser that sprays a nice scent into the can each time the lid is opened. Many of these features were altered or removed and they will all be discussed in section 3, “Design Approach”.

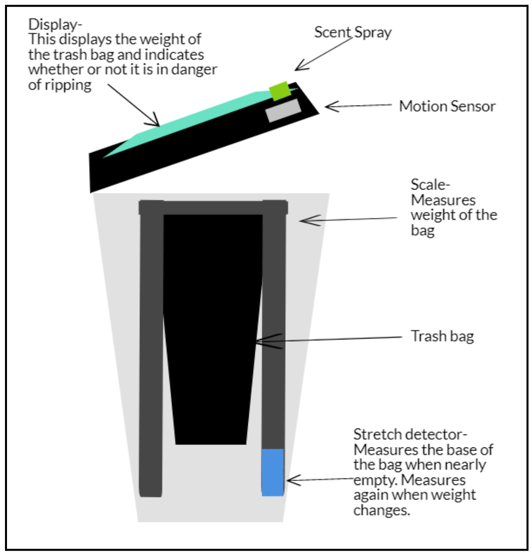


Figure 12: Original diagram of prototype design with specifications

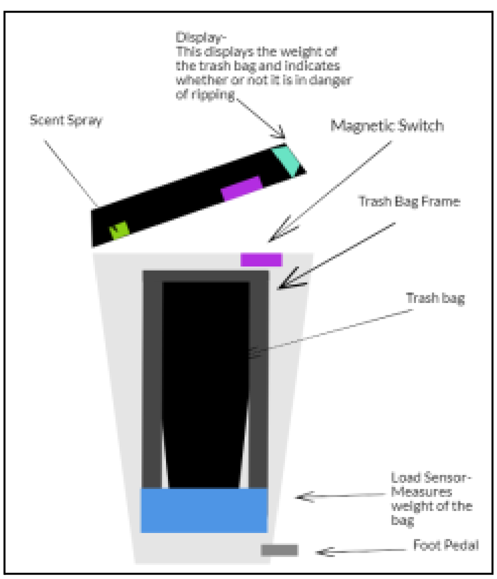


Figure 13: Second Artist’s Concept

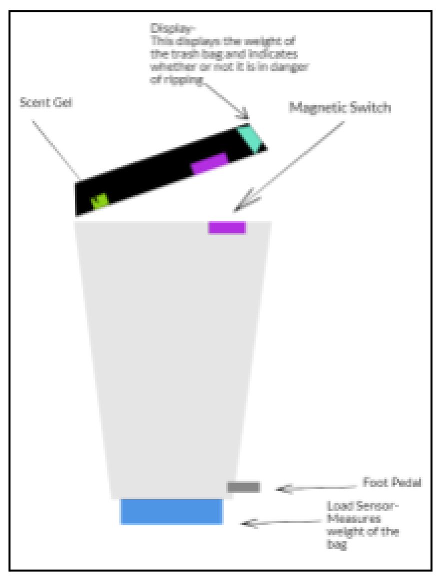


Figure 14: Final Artist’s Concept

The original design placed the load cell system inside of the trash can with the trash bag holder and trash bag being weighed as shown in Figure 13. The system was intended to be zeroed with the trash bag holder and the trash bag. Due to time constraints, we determined that the scale will go outside of the system and instead weigh the entire trash can; then it would zero in on its weight. The scented spray was also changed to a scented gel. The revised design is shown in Figure 14. It should be noted that we still intend for the weighing system to be inside the trash can as depicted in the original design, however, given the time constraints of this project we determined that having the bin on the scale is what is most feasible for the prototype.

2.4 Competitive Value Analysis

Customers require certain essential criteria when investing in a product. The quality of the product, convenient features it may have, and cost all factor into the decision-making process. The Weigh Bin functions as a quality-of-life product designed to add convenience to the lives of the customer. The Weigh Bin retains the basic functions of a typical trash can, such as having a foot pedal, with a few additional features. It was designed to measure the weight inside of the waste bin, display that said weight, and give recommendations for when to take out the trash based on the weight inside of the can.

In terms of quality, the Weigh Bin provides all the necessary functionalities of a regular trash can. It contains trash, has a foot pedal to open the lid of the trash can, and it requires the use of bags. It was also tailored to the most commonly used bag size of 13 gallons. When prompted in a survey we conducted the majority of participants indicated that the trash bags they most often used were 13-gallon trash bags. Customers would simply not buy it if it does not meet basic requirements. No one wanted to relearn how to use a trash can and as product designers, we did not expect them to.

In regards to overall convenience, participants in our survey indicated that the ability of the trash can to detect and warn when the bag was in danger of ripping was the most important feature of the product. This is where our trash can differs from other products on the market. Our smart waste bin has the capability to determine when the best time is to take out the trash, depending on the weight of the trash can. All of this information is easily visible on a display mounted on the lid of the system. This feature removes uncertainty and adds convenience to the life of the customer. This feature also serves to reduce the risk of injury for individuals who are among the elderly or disabled population because it allows them to decide whether the bag is too heavy or not. This was the most important feature because of the points listed previously. These additional features were the ones that separated the Weigh Bin from that of our competitors as it adds a level of convenience not seen in the smart trash can market thus far.

The cost was also considered when designing the Weigh Bin. The product was intended to be a reasonable price because the average trash can user does not want to allocate money to an overpriced trash can. Our team aimed to design a product capable of being competitive in the market. A smart trash can anywhere from $50 to $100 was an advisable goal to ensure affordability. This price range would also make our product competitive against others in the market.

The criteria mentioned above proved to be essential when considering the design of our product. The customer criteria were weighted on a scale of 1 to 5, where 5 is of the greatest importance and 1 is of the least. The results of the requirements are shown in the table below:

Requirement	**Weight**
Affordability	5
Weight Display	3
Visual Appeal	2
Accessibility	3
Additional Features	3
Size	3

Figure 15: Customer smart trash can criteria and weights

The table above depicts the factors considered when determining the customer's essential criteria, the most important factor in terms of weight for the Weigh Bin being affordability. It was determined that if the trash can was not affordable, then it has little to no value in the market. While the weight display was not considered to be a critical factor, it does provide a level of convenience to the customer. Our product is the only one that considered this feature. Visual appeal is the least critical factor because smart trash cans are fairly similar in terms of appearance, and as long as the trash can is not an eye sore it will suffice.

Accessibility is a moderately important factor that adds convenience to the customer. However, if a trash can is not exceptionally accessible, such as if the customer has to remove the lid and throw away the trash, it still serves its overall purpose. Additional features provide a level of convenience to the customer that is desirable, but these features are not entirely necessary. The last factor considered was size. The size was also a decently important factor because the customer expects the bag to have a certain capacity. At the same time, a trash bag is fairly adaptable in terms of how well it integrates with a trash can.

There were various smart trash can designs currently available in the market. We looked into the smart trash can market and determined that there were no competitors who designed and sold household smart trash cans that could measure and display the weight of the trash. We concluded that smart trash cans from Simplehuman, iTouchless, Townew T1, and EKO Mirage were our primary competitors in our market.

Simplehuman

The Simplehuman smart trash can is among the more costly of our competitors. According to Amazon, the cost of this product is $117.99. It is visually appealing and has a foot pedal for easy accessibility. However, it does not contain any additional features and it is primarily designed for 12-gallon trash bags. Both the pricing and additional features of our product are superior to that of the Simplehuman smart trash can. Other models are voice-activated but those models are designed for larger trash bags and are more expensive. Because of these factors, our product is competitive.

iTouchless

The iTouchless smart trash can comes at $84, making it quite affordable compared to the rest of the smart waste bins taken into consideration as competitors. This product is not capable of measuring the weight of the contents inside of it, nor is it capable of displaying this information. It does not stand out aesthetically, along with the rest of our competitors, so it is easily integrated into a kitchen. In terms of accessibility, this device is automatic. At the click of a button, the lid opens. An additional feature is that this device has an odor filter. The trash can is designed for the standard 13-gallon bag size making it appeal to the general market. This product may be competitive monetarily, but it does not display the weight of the trash can, the customer has to touch the trash can for it to open, and, while it does filter odor, it doesn't provide pleasant scents from it.

Townew T1 Smart Trash Can

The Townew T1 Smart Trash Can is by far the most expensive smart trash can identified as a competitor, costing $399.99. It does not feature a weight display, however, it does have a notable additional feature that our product nor our other competitors have. This device can seal bags with heat. Despite such a feature, this product does come with several drawbacks. The bags are self-sealing but they are custom bags specific to the product that the customer will have to buy more of when they inevitably run out. The trash can is also constrained to custom bags that only have a capacity of 4.4 gallons. Even though the device seals its own bags, opens via a motion sensor, and is visually appealing, it is far more expensive and impractical than any other competitor.

EKO Mirage

The EKO Mirage smart trash can comes at a decent price, relative to this market, of $99. It does not contain a weight display nor does it have any other notable additional features. This product does have a built-in motion sensor for added convenience and it is tailored to 13-gallon-sized bags for general consumer appeal. In terms of functionality it does the job but ultimately falls short when it comes to additional features and therefore does not stand out against the competition.

Our team conducted a value analysis of how our product compared to that of our competition. Based on customer requirements and product specifications our smart trash can asserts itself as the best product on the market. When compared to the iTouchless and EKO Mirage it comes out on top in terms of affordability and additional features. It may not necessarily be as visually appealing, but it does have a weight display for added convenience.

	Criteria							
Model #	**Affordability**	**Weight Display**	**Visual Appeal**	**Accessibility**	**Additional Features**	**Size**		
Weight	**5**	**3**	**2**	**3**	**3**	**3**	**Unweighted Score**	**Weighted Score**
Weigh Bin	3	5	3	3	4	5	23	72
Simplehuman	3	1	4	3	1	4	16	50
iTouchless	5	1	4	5	1	5	21	69
Townew	1	1	4	5	5	1	17	49
EKO Mirage	4	1	4	5	1	5	20	64

Figure 16: Competitor Value Analysis

In regards to the criteria of our value analysis, our biggest drawbacks are visual appeal and the user's accessibility. Our competitors profit from visually appealing products. Some of them also boast motion-sensing opening sequences. It is our belief that in terms of accessibility, a foot pedal is enough and keeps the cost down. Because we do not know what our product will look like when it enters the market it is difficult to gather its visual appeal, in this way, we decided to place it in the middle. This could lead to a source of error in our value analysis. Another possible source of error is that because the product has not been made yet it is difficult to determine its cost. For this reason, we also put it in the middle category based on our expected cost. The cost ended up being ranked 5 as it was more affordable if components were bought in bulk. This is discussed later on.

3. Design Approach

The final construction was intended originally to contain the following parts: SEN-10245 load cells, SparkFun Pro Micro microcontroller, Arducam 1602 LCD, and an alkaline 9V battery. However, upon further research, we opted to swap out the majority of these parts for a cheaper and more compatible option. Originally, we intended to use buttons that would connect to our microcontroller in order to interface with the system, but instead, we opted for a magnetic reed switch. The magnetic reed switch is a mechanism that is used to detect the lid of the Weigh Bin opening and closing. The magnetic reed switch was the perfect solution, which also allowed for everything to happen autonomously with no need for user interaction. The SparkFun Pro Micro microcontroller was what we initially intended to use as our microcontroller, however, we decided on the Arduino Nano Every. The original LCD we intended to use was the Arducam

1602 LCD. Instead, we choose to use GeeekPi I^2C 1602 LCD Display, which came with an I^2C interface module. This LCD was chosen because we realized when we switched from the SparkFun Pro Micro to the Arduino Nano Every microcontroller, that the Arducam 1602 LCD was incompatible with the Arduino Nano Every. The Arducam display was a 5V SPI-compatible LCD and the newly selected microcontroller did not come with a built-in SPI interface, which would make communication much more difficult. Needing a simple solution, we changed the display to one that could easily interact with the microcontroller that we settled on. After further research into the load cell system, we conclude that the best ADC was the HX711. Our initial intention, regarding which load cells to use, was to purchase and implement the SEN-10245; upon further consideration, we opted, instead, to utilize a DIYmalls 4-piece Load Cell. The Energizer 9-volt battery was chosen over the Duracell 9-volt Alkaline battery mainly due to the simplicity of our order and shipping time.

Buying all these parts from Amazon should result in a total price of around $42, which initially saved quite a lot compared to Digikey due to free shipping and bulk deals. We had to buy other things for testing later on which added to the price. After thoroughly considering customer and product requirements, as well as how realistic certain approaches would be due to time constraints, we decided that these were the components of the highest importance that can accomplish our goals while remaining well within our budget and time restrictions. These components allowed us to gather information about the weight of the trash bag, show it to the user, and let them know if there is any danger of ripping. All of this, in tandem with the foot pedal and scent gel, satisfied the needs of the customer, as well as the specifications of the project.

3.1 System Architecture

This section of the report details the design block diagram. The block diagram illustrates the functionality of the system. It depicts several inputs which are processed and subsequently responded to. This section also outlines the purpose of each individual module of the block diagram and discusses hardware and software partitioning where applicable.

3.2 Block Diagram

The original block diagram of the Weigh Bin is depicted in Appendix A, and the final one is in Appendix B. As described previously, some of the elements of the design changed due to many factors including cost and ease of use. The final block diagram outlines the various modules of the system and where these modules are in relation to one another. There are four main parts to the system illustrated in differentiated colors. The block in green indicates the power source, red indicates the microcontroller, blue indicates the sensors, and orange indicates the output. The software elements are gray with a magenta outline.

3.3 Module Purpose

This section of the report discusses each individual block outlined in the block diagram. It will address its purpose, utility to the project, and satisfaction with respect to customer requirements**.** This section of the report also provides a detailed explanation of the functionality of each module of the Weigh Bin system. Each section will provide a justification of each module's functions, supporting calculations, derivations, and experiments where necessary. It will also include a discussion of the module inputs, operation, schematics, outputs, testing, and system integration..

3.3.1 Energizer 9 Volt Battery

The battery, differentiated via the green outline, is essential to the functionality of the system. Its purpose is to power the microcontroller so that it can carry out its tasks. It is an Energizer 9-volt battery and is more than capable of supplying the microcontroller with enough power to have the system operating efficiently. Further discussion of battery life suggested that it will last a very long time, which will decrease how often customers will have to replace the battery.

The Energizer 9 Volt Battery is utilized in the system to provide direct current input into the Arduino Nano Every microcontroller. As depicted in Appendix E, the battery is in parallel with a 0.1 μF capacitor while connected to the V_{in} pin and ground of the microcontroller. This allows for the reduction of noise so as to provide a smoother, more stable voltage supply that can optimize the performance of the circuit. The capacitor will also hold and release voltage when necessary.

The voltage of the battery is 9V and the current capacity is 550mAh. It was determined by the team that the trash can be used for a very high estimate of 12.5 minutes per day, based on the number of times it was opened and how long the display remained on. The number of times it was estimated to be opened was about 12 times per day per person on the high end. For a family of four people, it would add up to a very high maximum of about 50 times the trash can was opened. These figures were derived by the team by keeping count of how many times each person opened their own trash cans. The 12.5 minutes calculation is derived from the number of times the trash can was opened multiplied by 15 seconds for the amount of time the current was drawn. The current draw was calculated to be ~0.55 mAh/day which equates to about 1000.8 days of use or about 2.74 years of use before one battery should die. We neglected to consider the lifetime of the battery because we were not able to find consistent results that could be put well into calculations. Although this calculation does not take battery degradation into consideration, we feel as if this number is still accurate to what a customer could expect due to the very high estimate of how often the trash can is opened.

Calculations:

Time in active state: 12.5 mins per day = 0.208 hrs per day (50 times opened, system active for 15 seconds)

Time in Idle state: 23.7919 hrs/day

Item	Active (mA)	Idle (mA)
CPU Clock	0.021	0.0007
I^2C	0.041	N/A
LCD	1.1	N/A
Load Cells	1.4	N/A
Total	**2.562**	**0.0007**

Figure 17: Current Draw from System

Active:
2562µA * 0.208hrs = **532.896µAhrs/day**

Idle:
0.7µA * 23.7916hrs = **16.65µAhrs/day**

Total:
532.896µAhrs/day + 16.65µAhrs/day = **549.55µAhrs/day**

$\frac{550\ mAhrs\ 9V\ Battery}{549.55\mu Ahrs/day}$ **= 1000.8185 days of usage**

This battery life was deemed to be more than satisfactory. The goal in terms of battery life was originally approximately one month. ~2.7 years is beneficial to both the team in terms of marketability and customer satisfaction, as they do not have to constantly replace the battery.

The battery was tested to determine how efficient it is initially and how the efficiency changes over time. It is important that the battery is attached to the microcontroller properly so as to not blow out the board and cost the team valuable time and capital.

3.3.2 Arduino Nano Every

The Arduino Nano Every module is the microcontroller of the system. This controller was used to process the input data from our sensors and switches and output relevant information to our display. We used the Arduino IDE to program the board so that we can take the data from our load cells and convert it to a digital readout of the weight of the trash bag. The board handles much of the power distribution of the system. This particular board is a relatively small form factor compared to many similar boards, and it consumes much less power making it optimal for our application.

When choosing the proper microcontroller we considered many factors including cost, form factor, power consumption, and I/O pins. We found that this board met our requirements the best based on our value analysis. The general specifications of the board are that it has 48 KB of flash, 6 KB of SRAM, 41 general-purpose I/O pins, I^2C compatibility, a 16 MHz base system clock, and a 32,768 Hz ultra Low-Power oscillator. The board also has many sleep functionalities including idle, standby, and power-down with limited wake functionality. We considered all of these features when choosing this board.

Looking at the necessary performance of the board for our application we did not have very high demands. We did not plan on having many complex algorithms and we believed that our board specs were more than exceeding the requirements we have for our performance. This board also has similar specs to other boards that we could have chosen but it was better in other categories.

One of those categories would be the form factor. The board is only 45mmx18mm in size and weighs under 5 grams. This makes it one of Arduino's smallest boards while still having the performance of some larger ones. Since we want our design to take up as little space as possible on the trashcan, this is a major benefit of this design choice.

The I/O pins were also a vital consideration for our board. Most displays come with many required connections unless you can use an interface protocol that removes most of these connections. Our board is I^2C compatible and only requires the use of 4 pins on the board compared to the potential 16 I/O pins we might have needed. This allowed us to have much more space to connect our ADC, power source, and input switch. With regards to input power, most Arduino boards require at least 6V to function normally. Since we wanted to use a battery for our application we needed a board that could be run using standard battery voltages and capacities making it easier for the user to change it out when necessary. This particular board has an input V_{in} pin that is connected to a power regulator. The regulator steps the voltage down to the normal 5V that the board functions at. Since we are using a 9V battery this worked, as this was enough voltage to keep the board powered. This board also has 5V and 3V3 output pins which we used to drive peripherals. Our displays run at 5V so we can use this pin for that device. We ran our ADC off of the 3V3 pin as this is better suited for its operating range and will consume less power, due to a lower voltage. Internal pull-up resistors were considered for the external active-low RESET pins on the board as well as a 35kΩ pull-up resistor for our switch. The internal pull-up resistors can be anywhere in the range of 20kΩ to 50kΩ with a typical value of 35kΩ.

The board can be run at a variety of voltages but we plan to set V_{dd} for the board to 5V. This means that the input high voltage for the board is 4.2V and the input low voltage is 1.5V. We have done the calculations for our switch to confirm that it meets these voltage thresholds for high and low input values. Each pin on the board also has a max sink/source current of 200mA at

25°C which is very close to the range we plan to operate at. None of our peripherals are near this range so we should not be in danger of damaging our I/O pins.

Perhaps the most important consideration for our design was the power consumption for the board. This board has a 32,768 Hz ultra low-power oscillator which was used for our system clock. The board also has many power modes which we can take advantage of. When we want to display the information we can put the board in an active state which using this clock consumes only 21 μA. When we are not displaying information the board only consumes 0.7 μA. With these numbers, we can extend the life of our battery seen in the calculations in section 3.1. The lower power usage and longer battery lifespan improve our project and are a big portion of why we chose this board.

The board does not have soldered headers but it comes included with through-hole attachments that can slot into a breadboard. This way we can access the pins just as if they had soldered headers. It is a tradeoff of this board but we believe it is worth the sacrifice given the small form factor and power consumption of the board.

The Arduino board handles much of the power and connectivity of the design. Power from the 9V battery is connected to the V_{in} pin on the board and also to GND. The microcontroller also takes samples in from the HX711 ADC through pin D16, connects the clock through pin D13, and takes power in from the 3V3 output on the board. The board also connects the magnetic reed switch through pin D4 as well as the 5V output pin on the board. The final connection comes with the I^2C module on the LCD display. Using pins D18 and D19 which are specified to use the I^2C protocol as well as the 5V output pin on the board.

Using the Arduino IDE software and the built-in micro-USB port on the board, we will be able to program the board to provide the functionality we desire. The Arduino IDE has a built-in debugger and simulation tools that will aid us in testing the functionality of our device. Using the LCD allowed us to test the functionality of our software as this is the output device that determines if our measurements have any real meaning to our system.

3.3.3 Magnetic Reed Switch

The magnetic reed switch is a mechanism that was used to detect the lid of the Weigh Bin opening and closing. It contains a magnetic switch that opens or closes - depending on the configuration - whenever the two parts of the switch are moved away from each other. This doesn't directly satisfy any customer requirements but indirectly helps us to satisfy them in another manner, as discussed below.

The magnetic reed switch lets the rest of the system know when to turn off. Depending on the way that it is connected, it can either function normally open (NO) or normally closed (NC). Based on our application and the fact that it is battery operated, we originally determined that the NO option would be more suitable, as it consumes less power than the NC, which would drain constantly. But as it turned out, the software was much easier to implement when the switch was in its NC orientation. Also, after testing the leads of the switch with a multimeter, there is only current passing through it (0.15mA) when the lid is open and there is no current when it is closed.

When the lid is closed, a custom interrupt is sent to the board, telling it that it needs to enter active mode and begin computing. This is a highly important part of our design, as the system as a whole would consume much more power if there was no way to enter and exit low-power modes as needed.

Being a product from Amazon, there was not much valuable information to find regarding the specifications of the part itself. From the little data that we could find, the part can handle up to 100V and 500mA, but it didn't state minimum or typical voltage and current ratings anywhere. Although, from searching through examples and other applications online, we found that using power from a microcontroller was plenty for the switch to work as intended and to provide us with a way to wake and sleep our system as needed. We supplied the switch with the 5V supply from the board and used an internal pull resistor to register low and high at the pin.

Additionally, the trigger range for the sensor was said to be anywhere between 5 and 20mm, so it was important to consider the way that the switch is mounted so it can receive correct signals when it is in the open or closed position. As long as the idle, closed-lid position of the sensor is from 5-20mm, it should work as intended when opened.

In order to test the functionality of the switch, we tested it outside of the system before attempting to integrate it to ensure that it works with the power that we are supplying it, as well as the distances that it advertised as being acceptable ranges of operation. We also tested the functionality of the custom interrupts to ensure that the logic high and low values that we supplied through the switch are correct and allowed the board to wake from a low power state. This was done with simple software tests and debugging, prior to attempting to integrate it into the whole system.

3.3.4 Board Power Mode Swap

Using the magnetic reed switch, the Arduino Nano Every would receive a custom interrupt through software, which would be used to wake the system from a state of low power. Every time the lid opens, the power state of the microcontroller would switch to active for a set amount of time (15 seconds), immediately returning to its low power mode when the timer is up. This optimization allows for much longer battery life, as the board will only be in active mode for a short time after the system is interacted with, rather than being awake all the time.

This part of the software is the reason that the magnetic reed switch was considered in the first place; we needed a way for the system to swap between power states, but we had no physical way to acknowledge when the lid was open or closed, making it near impossible to swap between power modes. We considered having buttons that were pressed or some sort of contact sensor that could tell when the lid opened, but those came with complications such as wear over time or possible faulty connections. The magnetic reed switch, on the other hand, can be used a near infinite amount of times due to the nature of the way that it connects, making the wear negligible. This allows for a simple and long-lasting connection that can serve as a way for the board to switch between power modes reliably.

Having a sensor that allows the system to know when the lid is open is a very powerful tool when it comes to saving battery life. After all of the parts of the system are set up and configured in the code, the first thing that happens in the forever loop is that the board enters one of its low-power modes. This is only changed when the magnetic reed switch sends an interrupt to a digital I/O pin, allowing the board to enter active mode and begin calculating. When in active mode, it will read from the load cells and use the ADC to convert the analog readings to digital signals for the microcontroller to read, and those signals will then be converted to text that will be sent through the I^2C module into the LCD screen. After these operations have been conducted for 15 seconds, the board will re-enter low-power mode, consuming only mere microamps of current, and saving a lot of power, allowing one battery to last exponentially longer than it would have without this feature. Refer to the calculations in the section about the 9V battery for exact values for current in active and low power modes.

As explained previously, the power mode of the microcontroller will be swapped using a custom interrupt from the magnetic reed switch. This will be set up and configured through one of the Arduino's digital I/O ports and used to wake the system from low-power mode for it to be able to make calculations and send data. We tested both the low-power mode and the switch separately, in order to make sure that both of them work on their own and combined the functionality once we knew that was true.

3.3.5 HX711 Amplifier

The HX711 amplifier is an analog-to-digital converter. Its purpose is to convert analog signals, received from load cells, into digital signals for the microcontroller to process. Once the signals are converted, they can be processed by the Arduino Nano Every microcontroller so that it may utilize this information to carry out its subsequent actions. It provides high resolution and low noise, making it optimal for high-precision applications. It is an important module because it allows for the interpretation of information gathered from the load cells, which can be used to derive the weight of the waste inside. This is the primary feature that differentiates the Weigh Bin from the competition.

The HX711 Amplifier is an analog-to-digital signal converter utilized in the system as a means to convert the analog signal obtained from the load cells into a digital signal that can be interpreted by the microcontroller. The E+ (Excitation Positive) port provides a voltage supply of 3.3V. This value is based on the Arduino Nano Every microcontroller board. The E- (Excitation Negative) pin provides a ground reference for the load cell configuration. The A+ (Signal Positive) pin receives the analog signal output from the load cell configuration. The A- (Signal Negative) pin acts as a ground reference for the analog signal output.

According to HX711 data sheets, the minimum voltage for the component is 2.7V while the maximum is 5.5V. The typical voltage ranges between these two values. The voltage that will be used is 3.3V based on the Arduino Nano Every. The typical analog current is 1400 μA while operating normally and 0.3 μA while operating when the power is down. The typical digital current is 100 μA while operating normally and 0.2 μA while operating when the power is down. The minimum output data coding is 800000 while the maximum is 7FFFFF according to the hex code values derived from analog readings.

The HX711 amplifier successfully integrates with the load cell system and the Arduino Nano Every microcontroller. It also accurately and effectively converts the analog signals from the load cell into digital signals for the microcontroller. This module is critical because without it the load cells and their subsequent readings are useless. This is detrimental in terms of meeting deadlines for the team, as well as having a useless product, if at all, for the customer.

3.3.6 DIYmalls 4pcs Load Cell

The DIYmalls 4pcs Load Cell is a set of four load cells that would be used together for weight measurement applications. The purpose of the load cells is to convert the force generated by an object's weight into electrical resistance changes that can be analyzed and processed by an analog-to-digital converter, such as the HX711. These load cells are essential to the project because of the Weigh Bin's ability to measure weight and therefore provide an assessment of the trash bag's potential to rip.

The DIYmalls 4pcs Load Cells function as a means to measure the weight of the trash inside of the trash can. In the schematic shown in Appendix D, the load cells are depicted as a Wheatstone bridge. They each act as variable resistors and when a force or weight is applied to the resistors the value of each resistance changes. Each load cell has a maximum weight capacity of 50 kg, which for the purposes of our project was more than sufficient. As these changes in resistance are very small, the analog signal from the variable resistor is sent to the HX711 to be amplified and converted to a digital signal to be read by the microcontroller.

The values for the variable resistors are 1000±20Ω for the input resistance and 1000±20Ω for the output resistance. Since we could not find a datasheet for this specific product, this specification was derived from the SEN-10245's datasheet, which is a similar load cell that was found on Digikey. The minimum voltage was listed at 2.7V, while the maximum voltage was listed at 5.5V. The current draw when operating normally was listed as less than 1.5mA and when operating at power down mode current draw was less than 1μA. Because we are using the Arduino Nano Every as the system's microcontroller the voltage that is going to be used for the load cell system is 3.3V.

The DIYmalls 4pcs Load Cell were able to handle at least 50 kg each and 3.3 V. It is expected that users would generally not have over 30 lbs of trash per bag. The load cells are also able to function as they are expected to with a favorable degree of accuracy, given their price. They are able to be properly put in a wheatstone bridge formation and integrated with the HX711 amplifier.

3.3.7 Convert ADC Measurements to Text

The software would also be used to convert the measurements from the load cells into something that the I^2C module and LCD can understand. This is a necessary step because without software, interfacing an analog sensor with a display would be extremely difficult. This is a major part of the required functionality, as weighing the trash bag and doing nothing with the information would completely defeat the purpose of the design and not satisfy the customer requirement of knowing the weight of the bag and when it will rip.

This is another critical part of our design that can only be achieved through software. Once the load cells and ADC work together to create a digital signal that is sent to the microcontroller, some computing must be done in order to change the code readings into characters that the LCD can display. The readings from the ADC must be converted from grams to pounds, which are then changed into characters for the screen and transmitted through I^2C protocol, and sent in a manner that the 16x2 LCD can understand. This satisfies the customer requirement of knowing the weight of the bag and when it is in danger of ripping; if we didn’t have a display, there would be no way of informing the user of these things.

This module would be tested by creating some sample text to ensure that the display worked properly before trying to send readings from the load cells and ADC to it. This is an important step in being able to pinpoint errors in code to debug and solve issues. If each module is not tested individually before trying to combine them, it makes finding errors much more difficult, as there are many more potential sources of error. Once we tested the display to make sure that it works on its own, we incorporated other aspects of the design.

3.3.7 Convert ADC Measurements to Text

In this segment of software, the microcontroller takes the measurements from the load cells and ADC and converts them into characters for the LCD. Involved in this is mainly just some simple math and conversions; the code value that the ADC presents to the Arduino must be converted into a weight, which is relative to the zeroed weight in the beginning of the code, and converted into a format that can be sent through I^2C buses to the LCD.

We will test these algorithms with some known test inputs that we will put into the functions manually, rather than taking it from the ADC, to make sure that the LCD works and the functions convert the numbers properly. This is a necessary step to ensure that our math is correct before trying to connect the sensors and ADC so we can identify problems in code much easier.

3.3.8 GeeekPi I^2C Module

The GeeekPI I^2C module is an interface device that comes with our selected display. This module, which came soldered to the pins on our LCD, uses the I^2C communication bus to send data from the microcontroller to the display. This module allows us to use fewer I/O pins than if we were to not use the I^2C protocol, thus saving board space, and clutter from the wiring, and making the whole system less complex. Using this module we are able to take the output data from our microcontroller and convert it to a digital display on our LCD. This allows our customers to see the measured weight of the bag.

Given that our board has I^2C compatibility, we decided to find an LCD display that also had I^2C compatibility. I^2C cuts the number of connections from 16 to 4 with this module. These connections are GND, 5V VCC, SDA, and SCL. The LCD we are using functions at 5V and our chosen Arduino board has a 5V output voltage pin, making the two compatible. The GND pin is tied to our common ground. In the I^2C protocol, there are two lines, SDA (serial data) which allows the master and slave to send and receive data, and SCL (serial clock), which carries the clock signal from the protocol. These lines will be connected to the corresponding pins on the board, D18, and D19. The protocol is configured in software and there is a provided library in the datasheet that would help us interface with the display. This particular module comes with our LCD so by nature it is compatible.

In order to test the functionality of this module we verified that the expected readout from the microcontroller is displayed properly on the display. We also have a library for the I^2C protocol that we needed to verify in the Arduino IDE. We were able to easily test that this module is working.

3.3.9 GeeekPi I^2C 1602 LCD Display

The GeeekPi I^2C 1602 LCD Display module is our display of choice for this project. This module is connected to an included I^2C module so it can communicate with the board. Using the output data from our board, we would be able to display, to our customers, the measured weight of the bag and if the bag is in danger of ripping. We would be able to output our measurements as displayed text on the screen for our users to see. This display meets our form factor requirements and would be mounted on the front of our trash can so it is easily seen by the user. It also has a bright backlight, which allows users to see the display even in the dark.

The GeeekPi I^2C 1602 LCD would be used to display the processed data from our microcontroller. It would output in text format the measured weight of the bag and also if the bag is in danger of ripping. The display itself runs at 5V and we have a 5V output pin on our microcontroller to power it. The display typically consumes 1.1mA of power when active, which was not a problem given our power calculations in section 3.1. Input high voltage is anywhere above 2.2V up to 5V and input low voltage is below 0.6V to -0.3V. Output high voltage on the display is anywhere above 2.4V up to 5V and output low voltage is anywhere below 0.4V up to 0V. The display also has an adjustable backlight controlled by a potentiometer located on the back of the I^2C module. The display can either be configured to use 16 I/O pins or 4 if you use the included I^2C module. As detailed in section 3.8, we used this module to interface with the display. We can display 16 characters on 2 lines on the display, which is plenty for our application. The display is 80x36x1.6mm in size, which was small, but worked for prototyping purposes. We mounted the display to the front of the trash can, so given these dimensions, it should not be too heavy so as to be a burden, but also not too small so that we cannot see the display. Given these specifications, we have faith that this display will function well given our application and other components.

To verify that the display was working we would test different readouts first in the software and then make sure the corresponding readout also showed on the display. We verified the I^2C protocol and then were able to test our display, showing the expected text from the microcontroller.

3.4 Overall Weigh Bin Design

In all, the main design of the Weigh Bin has stayed fairly consistent since the original idea. Other than some minor parts changes, the only design elements that were altered were the placement of the load cells, the method of opening the lid, and the method of producing scent.

We opted to change the location of the load cells from inside the trash can on a wire frame to under it and screwed it into a board. This was due mainly to the way that we had to attach them and the additional price of a frame to put inside the existing bin. Changing this element of the design allowed for much easier installation and testing, as the wooden plank with load cells attached is a separate unit from the trash can.

We decided that the motorized lid, which can be seen as a feature on many of our competitors' designs, was unnecessary and not feasible for our design. While, for some people, the automatic lid is preferred, the overall consensus was that a foot pedal was sufficient, and actually preferred by most. On top of the fact that the customers prefer the foot pedal, it was also much less expensive and easier to implement into our design compared to a motor, which would require additional modules in our design, as well as higher power consumption, which takes away from a major advertising point of long battery life.

Rather than continuing the idea of a scent spray that would actuate when the lid is interacted with, we opted for a small gel patch that sticks to the inside of the lid, providing a pleasant citrus scent for about a month. This still satisfies the customer requirement of having a nice-smelling trash can, while simplifying the design for our group and making it more affordable and easier for the user to replace. If we were to manufacture this product to sell it, we would either produce these gel pieces on our own or try to partner with the company that produces them to save money and have a strong point of advertising.

Finally, the material of the trash can was a large deciding factor that majorly impacted the price of the overall prototype. Originally, we debated between a metal or plastic trash can to implement our design into, and after a lot of consideration, a plastic one was chosen. This is mainly due to the price gap in buying a metal versus plastic trash can. This decision saved our group a lot in our budget, and if we were to manufacture these in the future, having a less expensive product would help us to better compete with the competition.

4. Product Results

4.1 Product Functionality

After much testing and prototyping we were able to successfully meet our product requirements within our final working prototype. Our product needed to have trash can functionality, weigh the trash placed in the bin, display that weight to the user, warn if the bag might be in danger of ripping, and also smell nice. Using the components mentioned earlier in the report, we developed a prototype that meets these requirements. The product, by default, is in low power mode and system peripherals are off. Once the user actuates the lid and allows it close using the foot pedal or by hand, a signal from our magnetic reed switch is sent to the microcontroller waking it from its sleep mode. The controller turns the display on and begins sampling the weight of the trash. The display turns on and displays "Calculating…" for three seconds and then displays the measured weight in grams from the load cells, which is then converted to pounds. The weight is displayed for another twelve seconds and then the system transitions back into low power mode. There are three different warning messages that appear on the second line of the display depending on the weight applied to the load cells. There is also a scent filter on the lid of our can to freshen the scent of it.

Our final prototype meets all the functionality requirements that we set out to achieve. When opening and closing the lid we use the magnetic reed switch to power the system on and off. The system is also completely operated off of a 9V battery. The system displays the weight of the trash to the user and outputs a corresponding message based on the weight range set form our research. The accuracy of our measurements is about ±3 pounds, which is acceptable for our application. We frequently get better results than this but, as it stands, three pounds is our worst case accuracy. While our load cells might not be accurate, they are precise. They consistently output very similar values when the weight does not change. This is a good start but we would like to improve the accuracy in future prototypes.

4.2 System Testing and Results

In order to test each component of the system and ensure that they all work properly on their own, brief software tests were written and run on each part. This was done to remove confusion later on by ensuring that the hardware was all working as intended. By proving this with basic code, we could remove one variable later on if we ran into problems; we would know that the hardware would not be what is causing the problem.

The first element of the system that was tested was the microcontroller, as almost nothing else can be tested without proving that the Arduino works first. We ran a simple "blink" test that comes with the Arduino IDE in order to prove functionality of the board. Once this was working, we were able to move on to testing each individual component of the design - the reed switch, load cells/ADC, LCD, and battery.

The LCD was one the easiest external parts of the system to test. Simply by plugging it in and using the library, everything worked as intended right out of the box. We were able to use the screen to display all the things we needed with virtually no complications.

Our first test for the reed switch was simply printing text to the terminal based on what state the switch was in. We used a simple loop that ran continuously, printing either “The lid is open” or “The lid is closed”. This was triggered by reading the pin that was connected to the switch and using an “if” statement that checked if the pin was low or high. We then implemented software that only printed when the state of the lid changed; it recognized lid-opening and lid-closing events. This style of recognition was ultimately used in the final code, as it allowed us to recognize the times that the lid was closing to be able to actuate the system. After we understood the mechanics of the reed switch, we used it to control the power state of the board, being its main function. Everytime the switch recognizes a lid-closing event, it triggers an interrupt, waking the board from low power mode and initiating the calculation and display sequence.

The load cells and ADC were by far the most complicated part of the system to test. We tested them first by using calibration code from the HX711 amplifier library. By running this code and testing different calibration weights and factors, we were able to find one that suited our weight range very well. The calibration factor that we settled on allows for reliable measurements within the range of about 0 to 30 pounds. While the measurements from the load cells are very precise - meaning they stay close relative to each other - they are not very accurate, as mentioned previously. There needed to be a lot of tweaking in order to find a calibration factor that was accurate within the intended range of weights. While this may seem like it was a simple task, this was by far the most time-consuming part of the software process. It took a long series of trial and error to be able to satisfy our requirements, mainly due to the inconsistency of the load cells.

Once basic tests were performed using each individual part, we began combining multiple elements and testing them together. First, we used the HX711 library calibration code and printed it to the LCD, as well as the serial monitor. This proved the functionality of those components together and allowed us to see the refresh rate of the display compared to printing directly on the computer. After this was proven to work, we combined the functionality with the reed switch and altered the code from all of the libraries so that it would function for our purposes.

Once all elements were tested and the basic software functionality was implemented using the micro-USB port for power, we tested the microcontroller and all of its peripherals being powered by the 9V battery. After operating for a few minutes and feeling the circuit to see if anything heated up, it was clear that it was operating safely. From this point on, all of our testing was done on battery power.

Testing all of the circuit components individually was very helpful for debugging and finding out what problems were as they arose. While it did take additional time to test each component by itself and integrate them slowly together, it definitely saved us time in the end, as debugging was much faster than it would have been otherwise.

5. Cost Analysis

5.1 Initial Investment

After conducting the cost analysis for the Weigh Bin, we determined, after salaries, office space, and other expenses that our fixed cost is $363,400 per year. As shown in the table below, the fixed cost for the first year, assuming we intend to continue to develop the prototype and form a company, is $363,400. This figure was derived from the costs in the table above fixed costs. Each of the 4 workers would receive a yearly salary of $70,000. The rest of the costs would be on a per month basis and make up the rest of our fixed cost. We estimate that to develop further prototypes, test the prototypes and eventually get the Weigh Bin on the market would take about 6 months given what we accomplished with 7 weeks of construction. This first 6 months would cost $182,000, half the yearly fixed cost. We have come to the conclusion that an initial investment of $182,000 would cover the costs for the first 6 months and would give us time to develop and get the Weigh Bin on the market.

Salaries: 3 Engineers	$210,000
Salaries: 1 Assembly Worker	$70,000
Office Space Per Month	$1,200
Business Trips Per Month	$3,000
Equipment Cost Per Month	$2,500
Marketing Per Month	$250
Fixed Cost	$363,400

Figure 18: Fixed Costs

5.2 Return on Investment (ROI)

PCB	$1.00
Display	$6.00
Load Cells	$10.00
Scent Gel	$2.00
Plastic Garbage Can	$10.00
Total	$29

Figure 19: Prototype Bill of Materials (Bulk Buy)

Assembly Cost	$2.00
Test Cost	$2.00
Packaging Cost	$3.55 for 990 units
Bulk Shipping to Amazon	$4 + $0.50/lbs *8
Unit Cost	$44.55
Selling Price	$78.33

Figure 20: Variable Costs (Bulk Buy)

As shown above, Figures 19 and 20 show the bill of materials and variable costs in terms of buying components in bulk. The cost of components in bulk was found on Digi-key and research on the cost per square inch of a custom PCB. The variable costs were determined from Amazon shipping costs per pound, test cost research, assembly cost research, and packaging cost research. From this, the unit cost was derived. The selling price was calculated by multiplying the unit cost by 1.3 for Amazon's fee, and again by 1.3525 in order for us to profit. This also allows us to keep the Weigh Bin cost competitive with our competition.

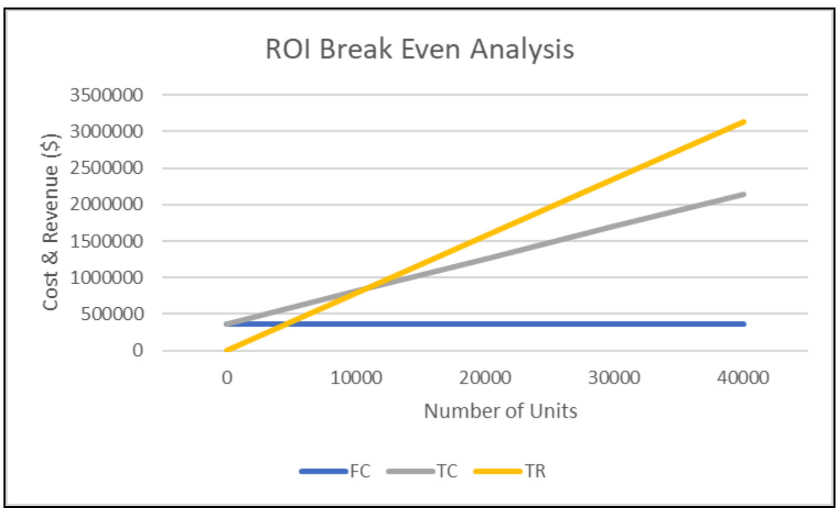


Figure 21: ROI Break Even Analysis

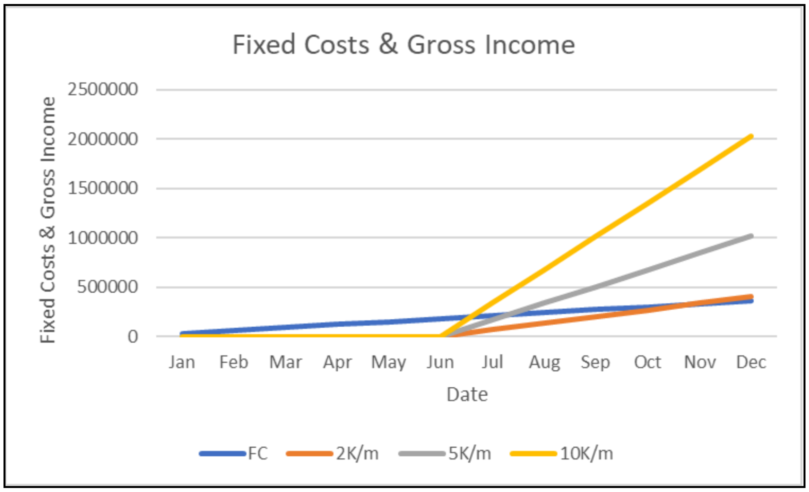


Figure 22: Fixed Cost and Gross Income

As shown in Figure 21 our break-even point occurs after the sale of 10,758 units given a unit price of $44.55 and a selling price of $78.33. As shown in Figure 22, if we sell 2000 units a month we turn a profit after 5 months of being on the market, selling 5000 units a month turns profits late-July or early August, and 10,000 units a month gets us in the green in just over a month. If we were to run a successful marketing campaign and if we obtained a large customer base, which is reasonable given that nearly everyone uses trash cans, we would expect to surpass our fixed costs in just a few months. Following the sale of 30,000 units, the return on investment comes out to be 16.9%. The equation below was used to calculate the ROI, where TC is the total cost, and TR is the total revenue. SP is the selling price, N is the number of units, and FC is the fixed cost.

$$ROI = \frac{TR-TC}{TC} = \frac{N*(SP-UC)-FC}{UC*N+FC}$$

6. Failure and Hazard Analysis

Failure and Hazard analysis have not been a large focal point for our project. Given the simplicity of our design, there are very few areas where failure may be an issue and there are also limited hazards we needed to consider. The obvious points of failure would be any of our electrical components: microcontroller, ADC, load cells, LCD, and the subsequent connections. While these are definitely considerations, each of these components should last well beyond our product's lifespan. It is reasonable to say that none of these components are likely to fail within five years of usage. This is only a general estimate gathered from our research and analysis of our component lifespans. Our product could likely run for much longer without failing but we believe our customers would most likely replace our product with a new one after five years. After that period of time, the can will become dirty and collect grime so it is reasonable to assume the user will want to replace it.

The only true failure point for our system would be the battery. It's not necessarily a failure point but once the battery capacity is depleted, the product will stop functioning. The user will be responsible for replacing it or it will not function. This will be clearly advertised to the user that when using our product you will have to replace the battery. But, given our 1000-day battery life, this is not a large inconvenience to the user as in the estimated five years the user might only have to replace it two or three times.

Another failure point in our project could be the hinge mechanism and the lid itself. We did not test durability or potential wear on the foot peddle and hinge mechanism. This was because the mechanical design was not important for our prototype. We bought a trash can and integrated our system into it to prove the concept. In a final design, we will hopefully design our own trash can with a hinge mechanism that we can test and design to meet our five-year estimated product lifespan.

Since we designed our product to function indoors we did not consider weather hazards or waterproofing our electronics. This could be a problem if the user spills something on the can but for the scope of our prototype, it was not a consideration. This would definitely be a topic we would need to explore in more detail for our final product. Spilling liquids on exposed electronics will most likely cause them to stop working and it could also present an electrical safety hazard if there are no fail-safes in place. This would be a major consideration for a final design but it is not something we addressed in our prototype.

7. Recommendations

As has been discussed in other sections of this paper, the model of the Weigh Bin that we have been working with throughout this term is a raw prototype with much room for improvement. There are a multitude of places in which we cut corners to be able to produce something that functions for our final presentations, but we have formulated plans for what would change in a future model that would allow for improved durability, functionality, appearance, and accessibility.

In a production model of the Weigh Bin, there are many changes to the design of the exterior that could be improved. Having exposed wires on the sides and top of the trash can is a hazard to the stability of the system, as it wouldn't take much force to rip one of them out of the pins in the breadboard or from the component itself. In the future, we would seek to either have these wires completely encased against the side of the trash can or even embedded in the plastic of the can itself. This would help so that the wires are not in any danger of being damaged, additionally having the benefit of improved visual design. Like most prototypes, the Weigh Bin is a rough approximation of what the final design would look like and having the wires cleaned up would make great improvements to the look of it.

Another improvement, which has already been mentioned, is relocating the load cells, microcontroller, and LCD units to more suitable places. As the load cells are right now, screwed into a piece of wood with the trash can sitting on top of it, the system isn't very stable - tipping or rocking occasionally when the foot pedal is used. If they were inside the trash can and only measured the weight of the bag, it would allow for a much more stable system that is more robust. We also would increase the quality of the load cells being used, as the current ones caused us a lot of trouble with accuracy and consistency. As for the microcontroller unit, it currently sits on top of the lid, which increases the weight of the lid, making it more difficult to open. Being able to store the microcontroller and battery in a place that is unseen, while still being accessible to change the battery, would be a great improvement for aesthetic and functional purposes. The LCD is in the same place as the microcontroller, sitting on top of the lid. This could be improved by having it be integrated into the lid or sit just below it. All of these improvements would make the design appear much nicer and improve how sturdy it is, making it more acceptable for sale.

After discussing our product with the Office of Accessibility Services (OAS), as well as our grandparents, we found a few things that could be changed to improve the accessibility of the Weigh Bin. The OAS stated: “This sounds like a great project. Some things that I can think of would be: An audio descriptor of the LCD screen (can it read aloud if someone is not able to see well/is blind?), how tall is the trash can height (i.e. 31 inches max for wheelchair accessibility).” Some advice that we received from our grandparents included: a buzzer, a bigger screen, a color changing screen (indicate warning level), and making the trash can taller to prevent bending over.

Overall, the model, as it is now, has a lot of room for improvement in relation to accessibility. The components that we purchased were all the least expensive options that we could find, making them not ideal for all situations. The LCD that we bought is white text on a blue background and is only a few inches long, making it difficult to see for someone with vision problems. This could be greatly improved with a larger, multicolored screen, which would allow more people to be able to see it clearly and even those who couldn’t read the small text would be able to see the color, indicating need for attention. Including a text-to-speech function or a buzzer would also be a very good idea to improve accessibility. This would allow the product to appeal to a wider group of people and increase the overall accessibility by a lot. Lastly, after receiving conflicting advice regarding the height of the trash can, we think it would be valuable to create multiple different styles that could be purchased based on need. A shorter and wider trash can would be better for those in a wheelchair and a taller one could help the elderly so they don’t need to bend over as much. This could also be incorporated by means of an adjustable-height trash can - having the top section be on sliding rails that allow it to compress or extend to fit anyone’s needs.

8. Conclusion

The Weigh Bin design approach has largely remained the same, with the exception of the placement of the load cells and the addition of the magnetic reed switch as opposed to the buttons. These modifications were crucial and quite necessary for the project to continue to run smoothly and for the team to meet its goals. The lesser modifications mainly involved switching out components for better or more optimal ones, which lowered our costs and power consumption, while raising functionality.

We believe our changes have enhanced our design and increased our chances of creating a functional prototype. While we realize that for a full product to be developed our design choices might have to slightly change, we believe the parts we have chosen provide a valid example of the intended design, which shows a working model of our product that demonstrates how it could be implemented if it were to become a product.

We also recognize we have some sources of error in our current design: quality of components, basic mechanical design, and available resources. While these items do hinder our results, they do not stop our prototype from displaying our intended functionality.

Over the past 7 weeks, we, team 7, were hard at work designing, building and documenting the Weigh Bin. We were even fortunate enough to be given the opportunity to present to CEO judges for valuable criticism on our product. This course gave each of us valuable insights and information on product market, design, and documentation, and through this, we improved our communication, time management, and teamwork. These skills will certainly come useful when we eventually do our MQPs. Overall this project provided invaluable experience, and even though, at times it was quite difficult, we managed to design a functional prototype that fulfills all requirements stated at the beginning of the class. From Adam, Alex, and Billy, we thank Professors Michalson and Stander as well as TAs Ankit and Rachel for all of your assistance and for the great experiences we all received from this course.

9. References

[1] Microchip Technology Inc., "ATmega4808/4809 Data Sheet," DS40002173C, Jan. 2019. [Online]. Available: https://ww1.microchip.com/downloads/aemDocuments/documents/MCU08/ProductDocuments/DataSheets/ATmega4808-09-DataSheet-DS40002173C.pdf. [Accessed: Apr. 11, 2023].

[2] SparkFun Electronics. (2014). Load Cell Amplifier HX711 Breakout Hookup Guide. [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/Sparkfun%20PDFs/Load_Cell_Amp_HX711_HookupGuide.pdf. [Accessed: Apr. 11, 2023].

[3] SparkFun Electronics. (2012). Load Cell - 50kg. [Online]. Available: https://www.sparkfun.com/datasheets/Sensors/loadsensor.pdf. [Accessed: Apr. 11, 2023].

[4] Sparkfun Electronics, "Getting Started with Load Cells," Digi-Key Electronics, 2017. [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/Sparkfun%20PDFs/Getting_Started_with_LoadCells.pdf. [Accessed: Apr. 11, 2023].

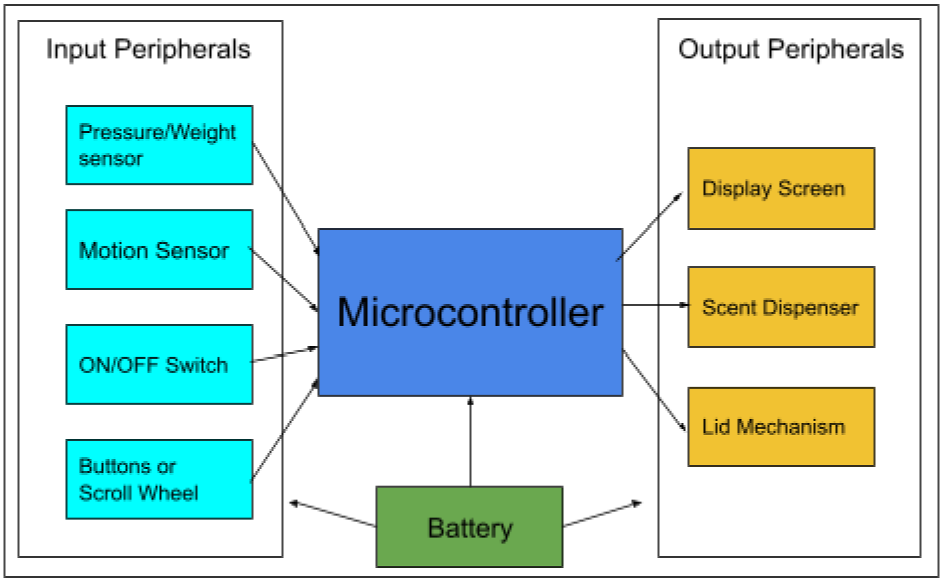
[5] Sparkfun Electronics, "HX711 Load Cell Amplifier," Sparkfun Electronics, 2015. [Online]. Available: https://cdn.sparkfun.com/assets/b/f/5/a/e/hx711F_EN.pdf. [Accessed: Apr. 11, 2023].

[6] Eone, "Eone 1602A1 LCD Module," OpenHacks, 2013. [Online]. Available: https://www.openhacks.com/uploadsproductos/eone-1602a1.pdf. [Accessed: Apr. 11, 2023].

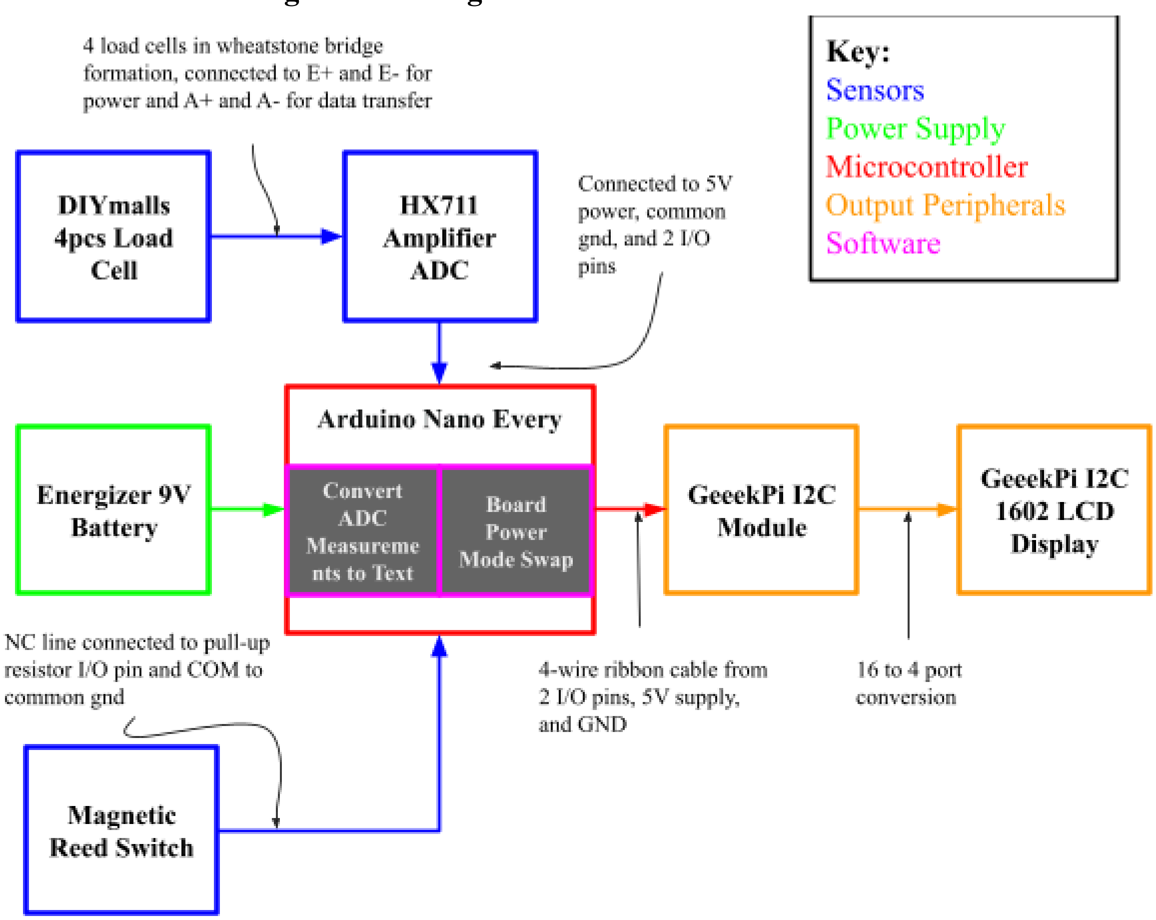
[7] "I2C 1602 LCD Datasheet," HandsOn Technology, 2012. [Online]. Available: http://www.handsontec.com/dataspecs/module/I2C_1602_LCD.pdf. [Accessed: Apr. 11, 2023].

10. Appendix

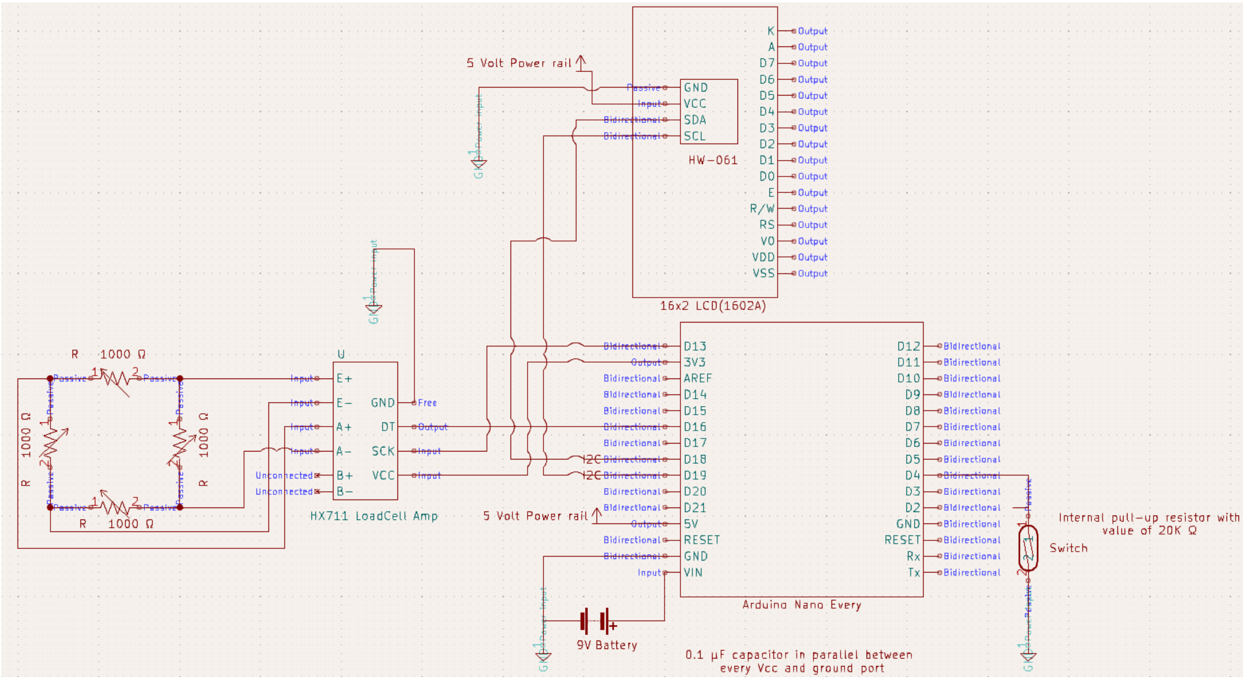
Appendix A: Original Block Diagram for Weigh Bin



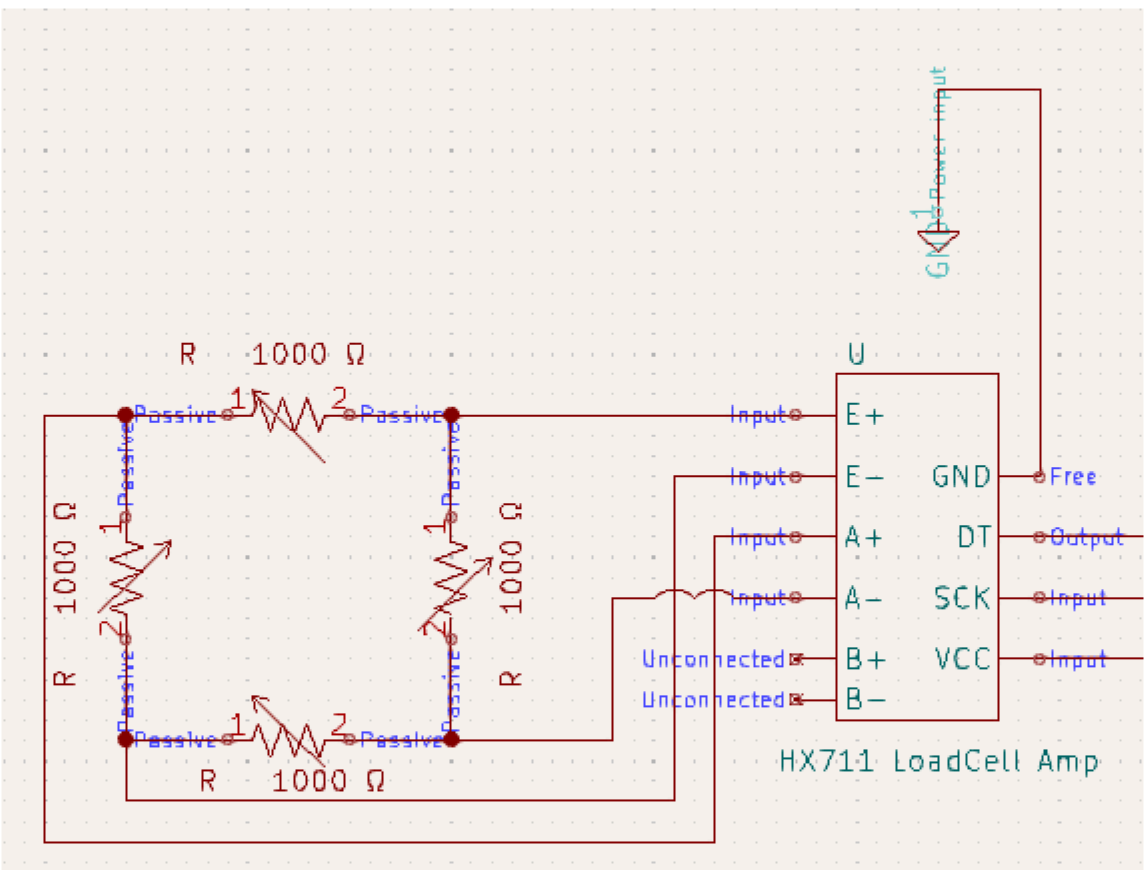
Appendix B: Final Block Diagram for Weigh Bin



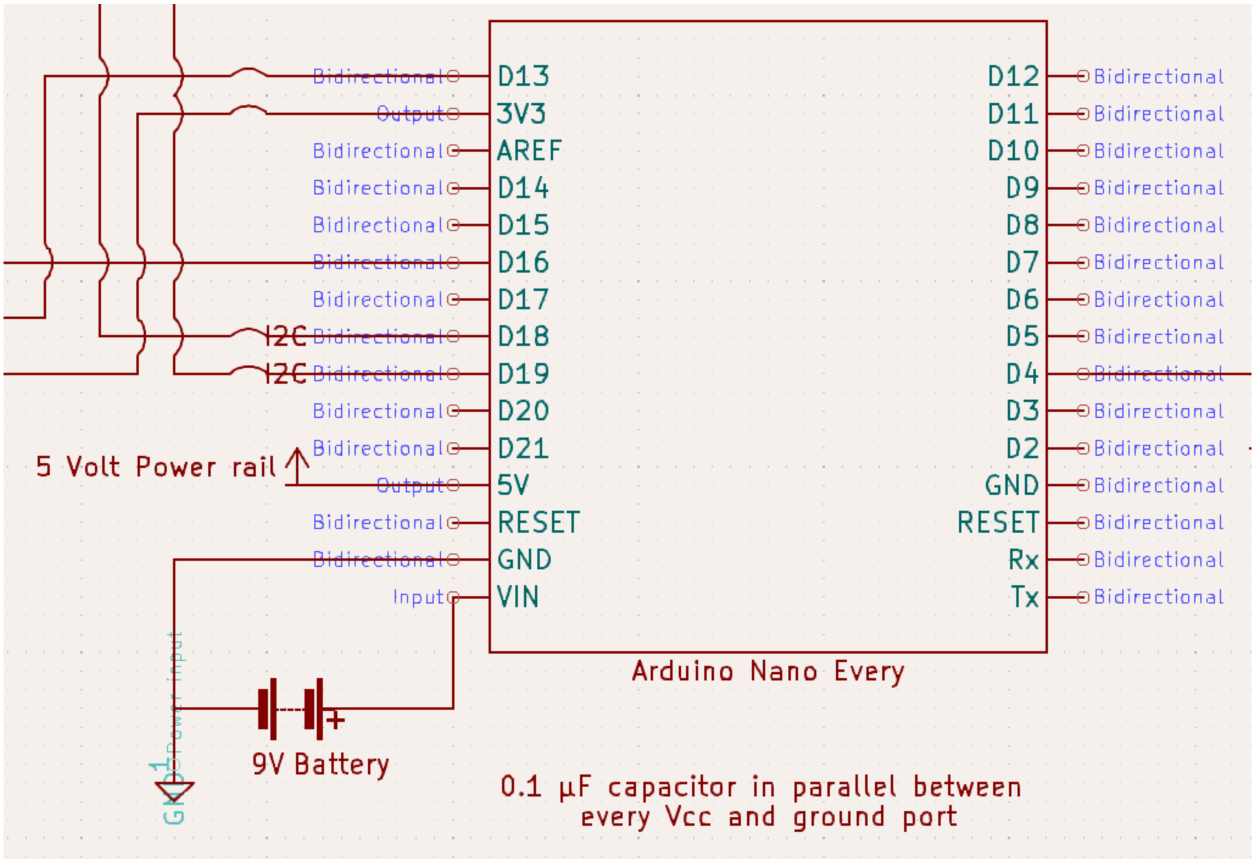
Appendix C: Full Hardware schematic for Weigh Bin



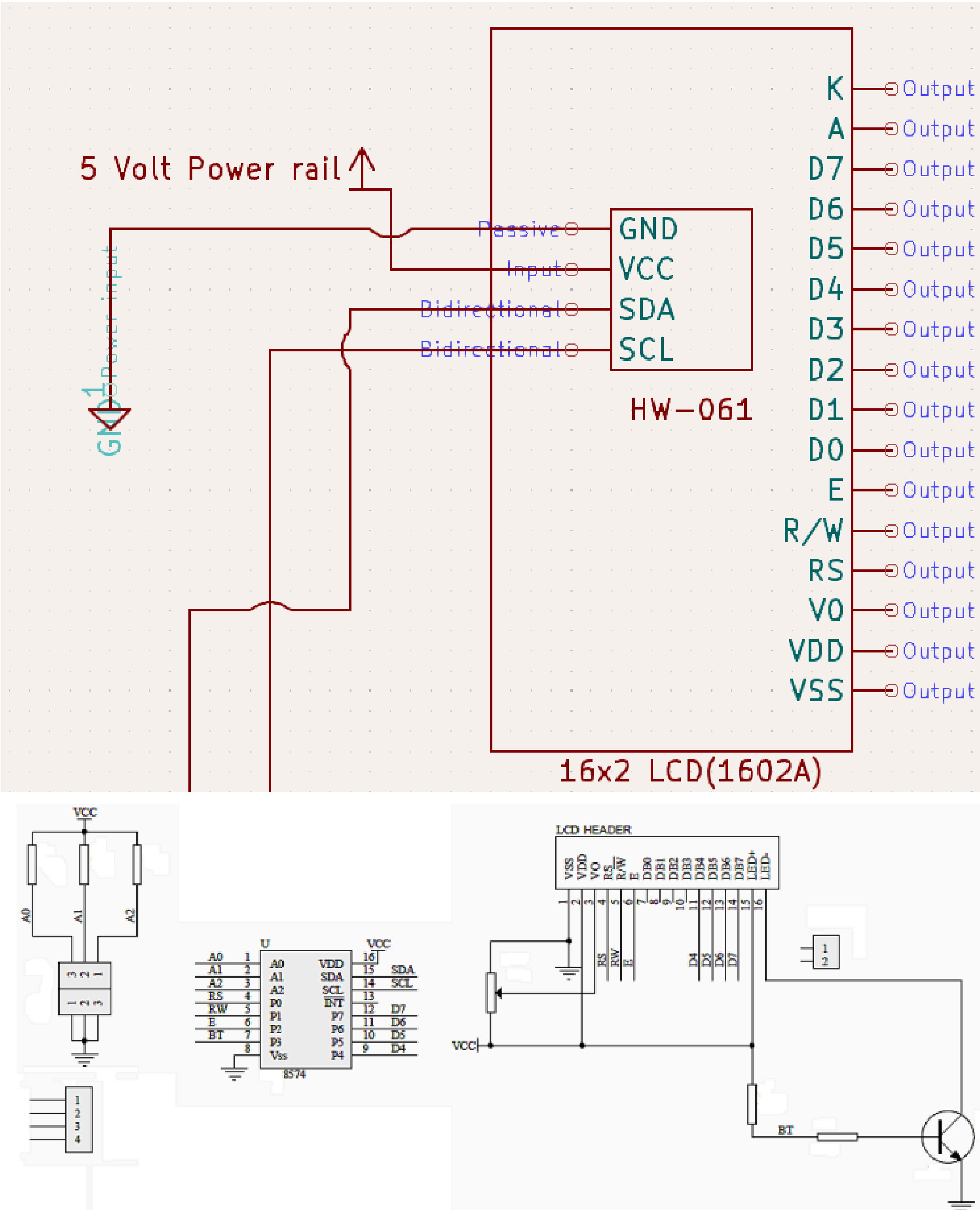
Appendix D: Hardware Schematic for Load Cells and HX711 ADC



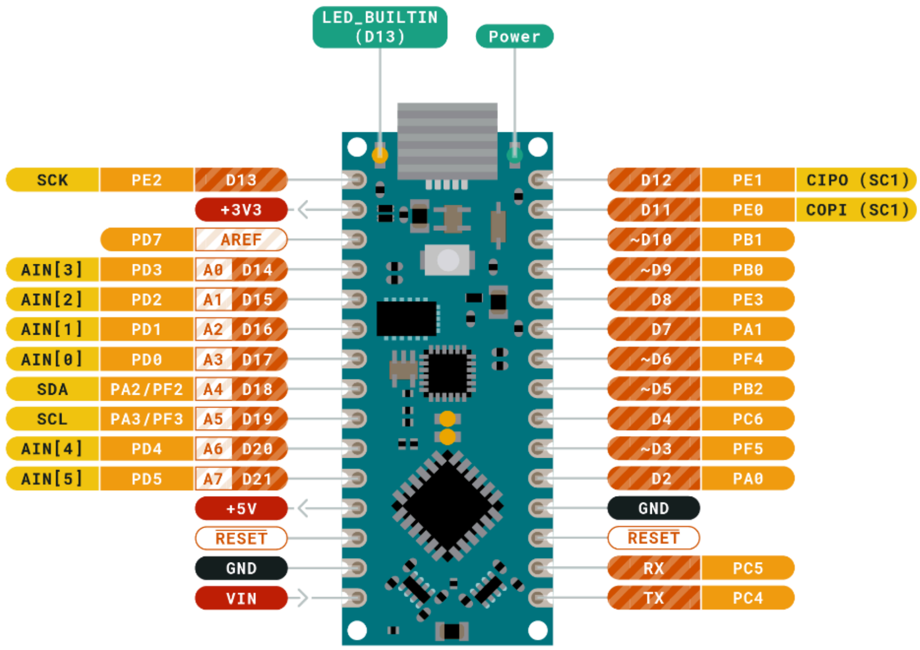
Appendix E: Hardware schematic for Arduino Nano Every and 9V Energizer Battery



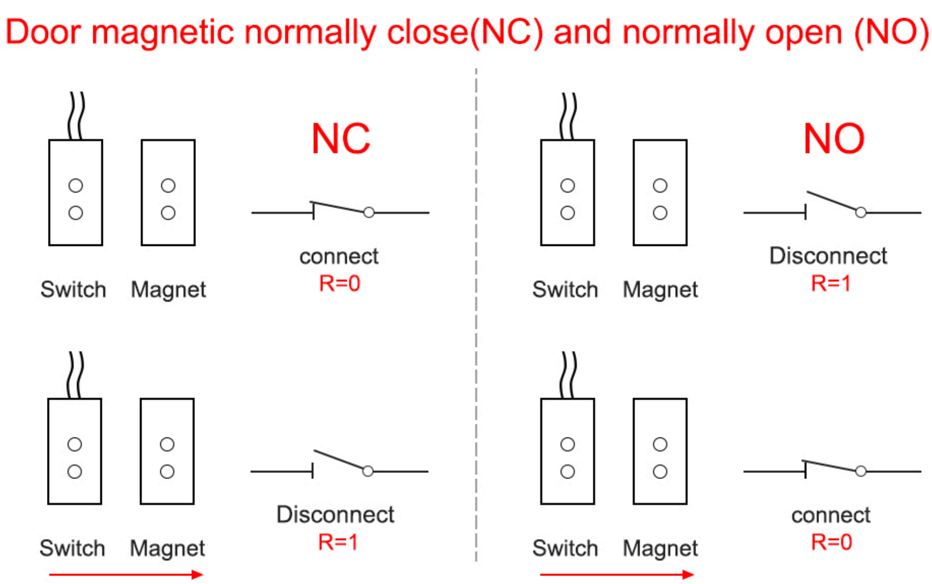
Appendix F: Hardware schematic for I²C module and LCD display



Appendix G: Pinout for Arduino Nano Every



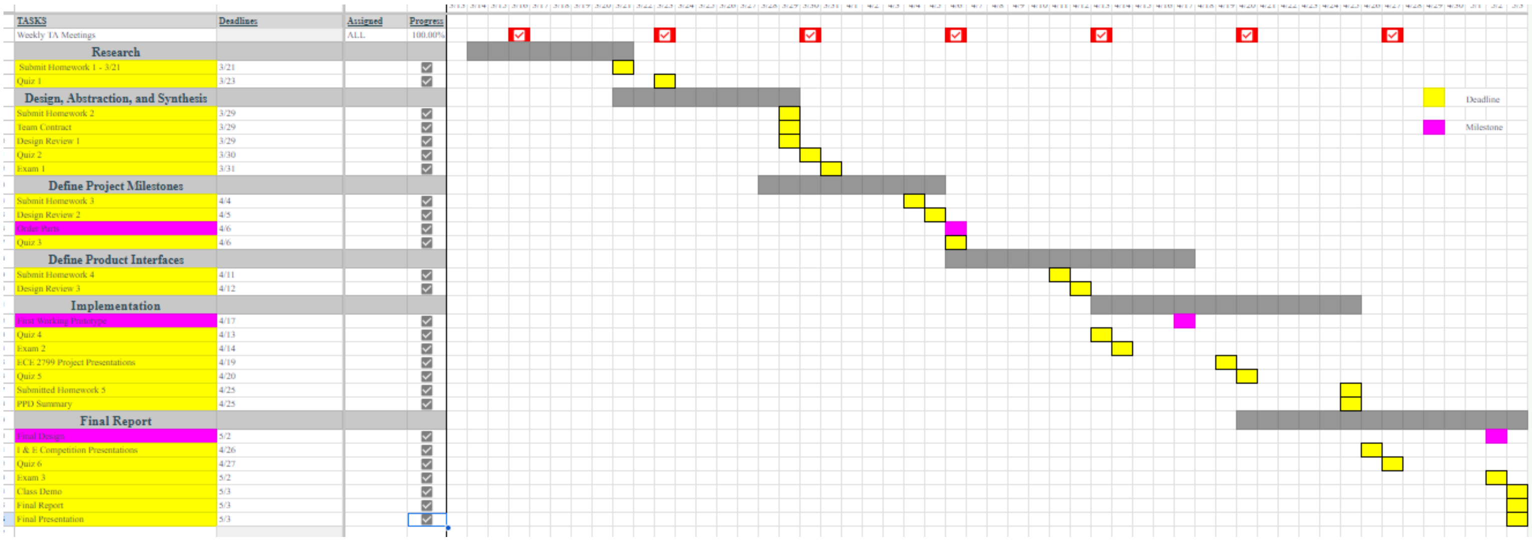
Appendix H: Magnetic Reed Switch Operation



Appendix I: Value Analysis and Power Calculations Spreadsheets

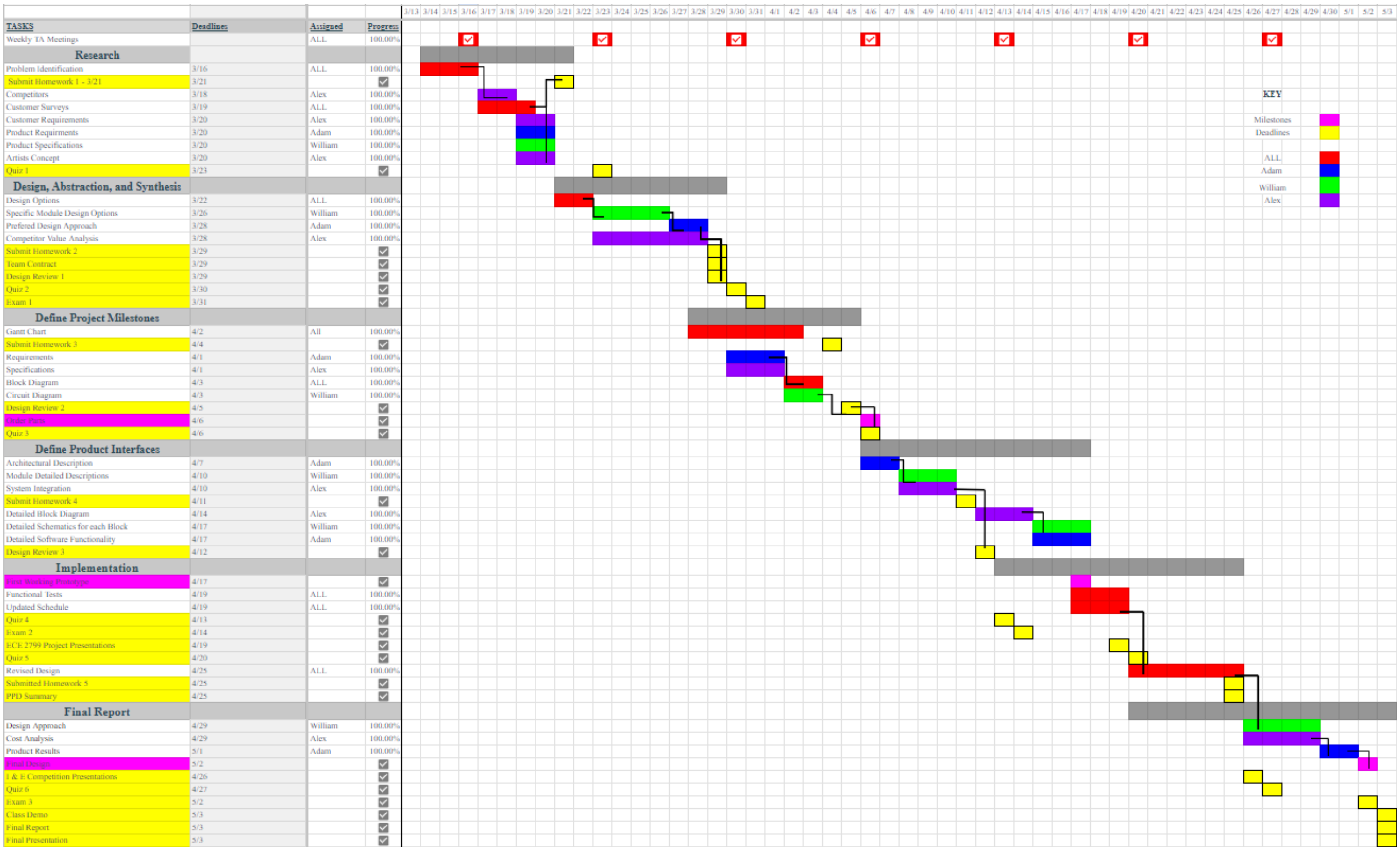
Value Analyses Power Calculations

Appendix J: Top-Level Gantt Chart



Top Level Gantt Chart

Appendix K: Full Gantt Chart



Full Gantt Chart

Appendix L: Final Code

```
#include <LiquidCrystal_I2C.h>
#include <avr/sleep.h>
#include <HX711_ADC.h>          // ADC library


const int magneticSwitchPin = 4;  // The pin to read the
magnetic switch
const int HX711_dout = 16; //mcu > HX711 dout pin
const int HX711_sck = 13; //mcu > HX711 sck pin

int sampleCount;
float total;

LiquidCrystal_I2C lcd(0x27,16,2);  // set the LCD address to
0x27 for a 16 chars and 2 line display
HX711_ADC LoadCell(HX711_dout, HX711_sck);

bool doorOpen = false;

void setup() {

  CLKCTRL.MCLKCTRLA = CLKCTRL_CLKSEL_OSCULP32K_gc; // sets CPU
clk to run on ultra low-power 32.768 kHz oscillator
  pinMode(magneticSwitchPin, INPUT_PULLUP); // Set the magnetic
switch pin as input and enable pull-up resistor
  attachInterrupt(digitalPinToInterrupt(magneticSwitchPin),
magneticSwitchISR, FALLING); // Attach interrupt to magnetic
switch pin
  lcd.init();                      // initialize the lcd

  LoadCell.begin();
  unsigned long stabilizingtime = 2000; // preciscion right
after power-up can be improved by adding a few seconds of
stabilizing time
```

```
  boolean _tare = true;
  LoadCell.start(stabilizingtime, _tare);

  float calFactor = -6;

// ***make sure nothing is on scale when powering up***
  LoadCell.tareNoDelay(); // Sets 0 weight
  LoadCell.setCalFactor(calFactor); // user set calibration
value
}

void loop() {
  if (doorOpen) { // If the door is closed: calculate for 3
seconds, get average weight, display for 12s

    sleep_disable(); // Disable sleep mode
    averageSamples();
    lcd.noBacklight();
    lcd.clear();
    lcd.noDisplay();
    doorOpen = false; // Reset the flag
  }

  // Enter deep sleep mode
  set_sleep_mode(SLEEP_MODE_PWR_DOWN); // Set the sleep mode to
power down
  sleep_enable();                      // Enable sleep mode
  sleep_cpu();                         // Enter sleep mode

}

void magneticSwitchISR() { // acknowledge lid opened
  doorOpen = true;
}
```

```
void averageSamples() {
  sampleCount = 0;
  total = 0;
  unsigned long startTime = millis();

  lcd.display();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(1,0); // beginning of text: column, row
  lcd.print("Calculating...");

  static boolean newDataReady = 0;
  float inputValue = 0;

  while (millis() - startTime < 3000){
    // read input from ADC and add to total

    if (LoadCell.update()) newDataReady = true; {
      inputValue = LoadCell.getData();
      newDataReady = 0;
    }

    total += inputValue;
    sampleCount++;
    delay(10);
  }

  float averageValue = total/sampleCount;

  float pounds = averageValue*0.00220462; // convert to pounds

  float readOut = 0;

  readOut = (pounds/1.3); // after some testing, the displayed
weight is about 1.3x the correct weight
  // this should offset it back to closer to the real number
```

```
  if (readOut < 0)
    readOut = 0; // prevent negative numbers from being
displayed

  if ((readOut >= 15) && (readOut < 20)) { // when weight is
between 15 and 20 pounds
    lcd.clear();
    lcd.setCursor(3,0); // beginning of text column, row
    lcd.print(readOut, 1);
    lcd.setCursor(9,0);
    lcd.print("lbs");

    lcd.setCursor(0,1);
    lcd.print("Caution,");
    delay(6000);


    lcd.clear();
    lcd.setCursor(3,0); // beginning of text column, row
    lcd.print(readOut, 1);
    lcd.setCursor(9,0);
    lcd.print("lbs");

    lcd.setCursor(0,1);
    lcd.print("Nearing capacity");
    delay(6000);
  }

  else if ((readOut >= 20) && (readOut < 25)) { // when weight
is between 20 and 25 pounds
    lcd.clear();
    lcd.setCursor(3,0); // beginning of text column, row
    lcd.print(readOut, 1);
    lcd.setCursor(9,0);
    lcd.print("lbs");
```

```
  lcd.setCursor(0,1);
  lcd.print("Warning!");
  delay(6000);


  lcd.clear();
  lcd.setCursor(3,0); // beginning of text column, row
  lcd.print(readOut, 1);
  lcd.setCursor(9,0);
  lcd.print("lbs");

  lcd.setCursor(0,1);
  lcd.print("May rip soon!");
  delay(6000);
}

else if (readOut >= 25) { // when weight is above 25 pounds
  lcd.clear();
  lcd.setCursor(3,0); // beginning of text column, row
  lcd.print(readOut, 1);
  lcd.setCursor(9,0);
  lcd.print("lbs");

  lcd.setCursor(0,1);
  lcd.print("Bag too heavy!");
  delay(6000);


  lcd.clear();
  lcd.setCursor(3,0); // beginning of text column, row
  lcd.print(readOut, 1);
  lcd.setCursor(9,0);
  lcd.print("lbs");

  lcd.setCursor(0,1);
```

```
    lcd.print("TAKE IT OUT!");
    delay(6000);
  }

  else { // default case, anything under 15 pounds
    lcd.clear();
    lcd.setCursor(3,0); // beginning of text column, row
    lcd.print(readOut, 1);
    lcd.setCursor(9,0);
    lcd.print("lbs");
    delay(12000);
  }

}
```