Kitchen Dry Ingredient Tracker

ECE 445: Senior Design
Team 43
20th March 2024

Anju Jain (anjuyj2)
Nynika Badam (nbadam2)
Sanjana Kumar (spkumar4)

TA: Vishal Dayalan
Professor: Arne Fliflet

Contents

1 Introduction...3
1A. Problem...3
1B. Solution...3
1C. Visual Aid...4
1D. High-level requirements...4
2 Design...5
2A. Block Diagram...5
2B. Physical Design...6
2C. Subsystem Overview & Requirements...7
2C.I User Subsystem...7
2C.II Sensing Subsystem...10
2C.III Control Subsystem...13
2C.IV Power Subsystem...14
2C.V Hardware Interface Subsystem...15
2D. Tolerance Analysis...16
3 Cost Analysis and Schedule...19
3A Cost Analysis...19
3B. Schedule...20
4 Ethics and Safety...22
5 Citations...23

1 Introduction

1A. Problem

In today's world, there is no end-to-end maintenance system for kitchen ingredients. It's hard to keep track of ingredients in our kitchen, their availability, and gauge when a necessary grocery run is required. Even though online grocery shopping has gotten very popular, consumers still frequently go to physical grocery stores. The primary grocery shopper in U.S. households made an average of 1.6 shopping trips per week in 2022 [14]. This implies that many consumers usually forget to buy ingredients they need and are forced to go back to the store. The Kitchen Dry Ingredient Tracker is designed to optimize these grocery trips by cutting down on the number of grocery runs a consumer makes. Home cooks can improve their shopping experience through the Kitchen Dry Ingredient Tracker.

1B. Solution

The system is designed to track and communicate with users about their spice necessities. The user can tailor each individual spice to different lower weight threshold measurements. The system uses RFIDs to allow users to place spice containers in any slot while recognizing the spice and its lower weight threshold measurement. The system maintains a dynamic digital grocery list by adding/removing spices based on their individual weights. To allow for a visual representation that a spice has run below the user specified weight limit, the LED light at the container's spot turns red. If a user is outside and close to a grocery store (0.1 miles away), a mobile app notification will be sent to the user's phone to notify them about ingredients on the grocery list and encourage them to stop at the grocery store. An example notification is "You are 0.1 miles away from Target. Time to grab salt and pepper"

1C. Visual Aid

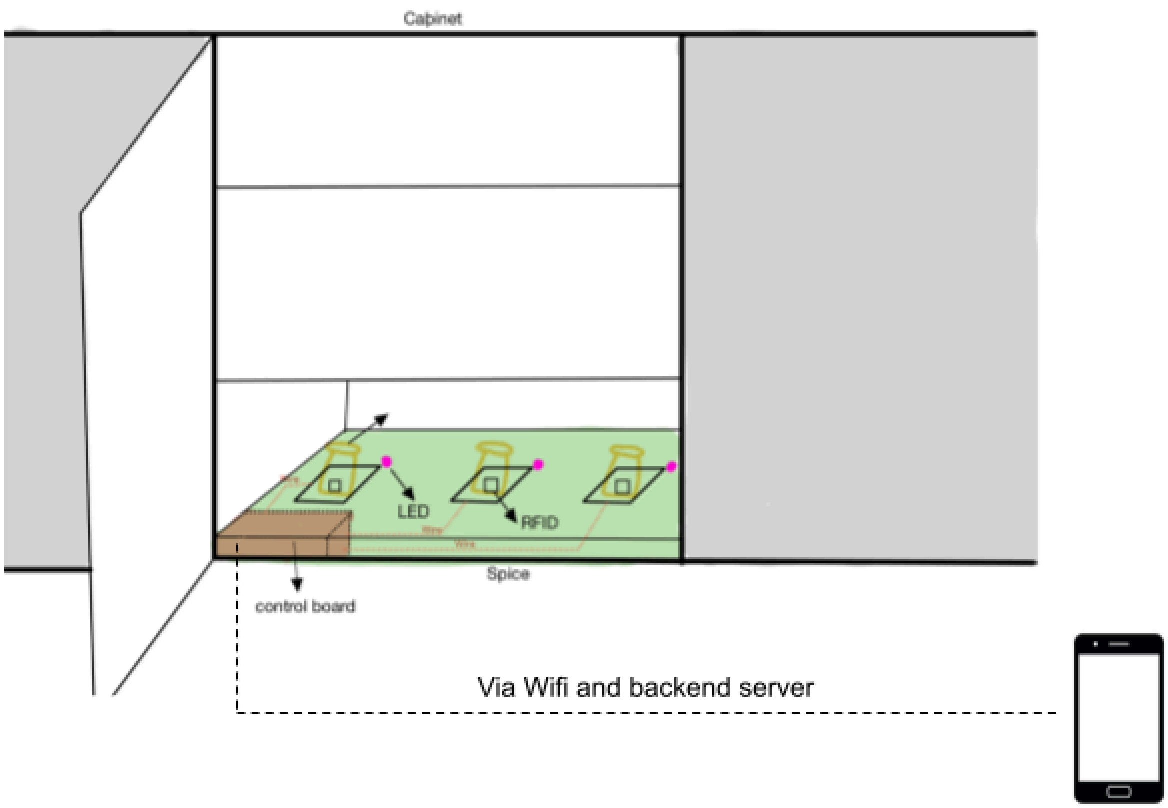

Figure 1: Visual Aid

Our system is capable of tracking numerous spices but for the constraints of the semester, it is designed to track three spices. The product tracks three ingredients, since adding more spice slots is redundant in mechanism. There are also additional constraints for manufacturing this product with more than 3 slots in a timely manner via the mechanical shop. The spices tracked for the demonstration are salt, garlic powder and chili powder.

1D. High-level requirements

1. The MCU will pull weight data captured from the load cells every 15 minutes for the 3 spices tracked. The maximum weight for the spices is 500 grams.
2. The app will have an “Ingredient Dashboard” and “Grocery List” interface. The Ingredient Dashboard allows users to specify lower weight thresholds for each spice within the range of 0 - 500 grams and recorded in whole grams. The Grocery List interface maintains a list that adds/removes spices based on whether their recorded weight is below its lower weight threshold.

3. Users can place spice containers in any of the 3 spots. The container's RFID will keep track of the container's position, allowing for the right comparison between the spice's weight and its lower threshold weight.

2 Design

2A. Block Diagram

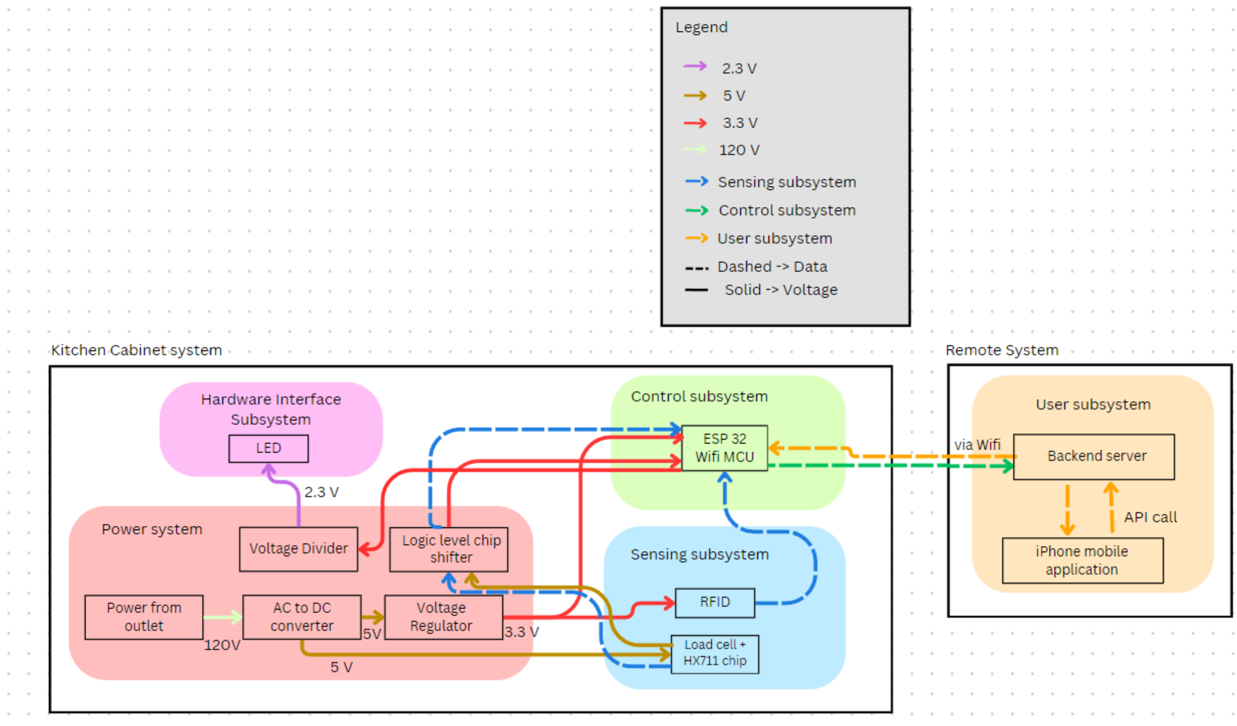

Figure 2: Block Diagram

The system is made up of a power subsystem, control subsystem, sensing subsystem, hardware interface subsystem, and user interface subsystem. The power subsystem powers the device. A wall power adapter converts 120V AC to 5V DC for the RFIDs and load cells. A voltage regulator will be used to step down the 5V to the 3.3V needed by MCU. The system uses resistors to step down the voltage from 3.3V to 2.3V for the LEDs. The sensing subsystem includes the RFIDs and load cells. The container's RFID will keep track of the container's position, thus allowing for the right comparison between the spice's weight and its lower threshold weight. The load cells in conjunction with the HX711 chip measures the weights of the spices. The control subsystem is made up of the ESP32 MCU. The Wifi protocol of the ESP32 MCU will be used to communicate with the Firebase database to store data. This data is transferred and shown within the app. The user interface subsystem allows the user to interact with the system. The app allows the user to input the spices names and their lower thresholds, see ingredient weights, maintain a grocery list, and receive notifications when they are near a grocery store. The hardware interface subsystem consists of LEDs. These LEDs turn red when the spice weight runs below its lower weight threshold.

2B. Physical Design

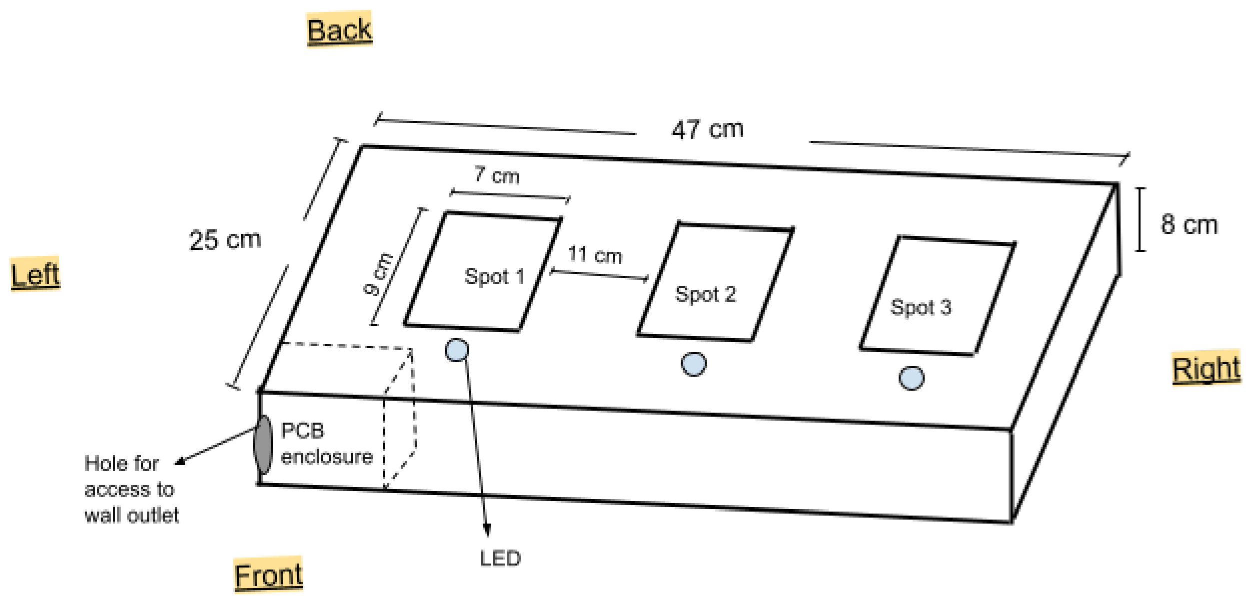

Figure 3: Overall structure (from above)

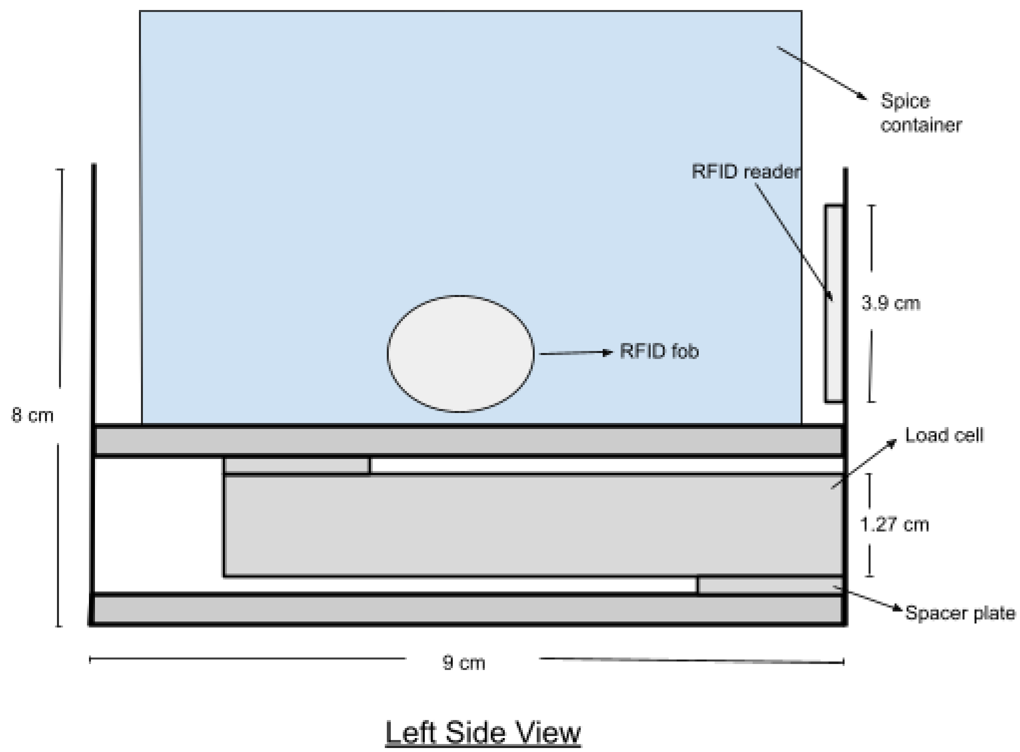

Figure 4: Individual Ingredient Spot (sideview)

Figure 3 and 4 depicts the physical drawing of the system. The slots for each container to be placed in the system is 9 x 7 centimeters. This accounts for the container's dimensions and space for the RFID readers. The RFID readers have a reading range of 3 to 5 cm [11]. The RFID stickers work about 4 inches (10.16 cm) from the reader [2]. Each spot is spaced out by 11 cm in order to stay within the reading range of its reader while avoiding the range of other RFID readers. The overall base that the spice container slots will be built into is 13 cm x 47 cm x 8 cm to account for the dimensions just described. The control board will be placed in the left front corner so that it's close enough to the sensors that need to be wired.

2C. Subsystem Overview & Requirements

2C.I User Subsystem

The subsystem contains the mobile application which allows the user to interact with the device. On a mobile app, the users will be able to track weights of each kitchen ingredient, view their digital grocery list, and get notifications to grocery stores 0.1 miles away. The "Add New Ingredient" screen, as shown in Figure 6, allows the user to provide the name of spices and the lower weight thresholds for each ingredient. The subsystem will communicate with the microcontroller to gather information about ingredient weights from load cells and compare them to the user's specified lower weight thresholds. The dynamic "Grocery List" synchronizes with the "Ingredient Dashboard" and only displays ingredients that are below the lower weight threshold. The subsystem gathers information from the RFIDs to determine what ingredient is in a particular slot. This information is used to compare against the weight sent by the load cell to the spice's lower weight limit.

If a user is outside and close to a grocery store (0.1 miles away), a mobile app notification will be sent to the user's phone to notify them about ingredients on the grocery list and encourage them to stop at the grocery store. An example notification is "You are 0.1 miles away from Target. Time to grab salt and pepper"

Google's Firebase Realtime Database is used to sync the mobile application and MCU. It is a cloud hosted database that allows for devices to receive data on wifi and mobile data [7]. The ESP32 MCU will connect to a user's home wifi network with its built-in wifi protocol. It will then use HTTPS requests to communicate with the Firebase database and update information. The database will communicate with the mobile application using the Firebase JavaScript API.

Every 15 minutes the MCU will pull weight data from the load cells and send the data to Firebase. The 15 minute data frequency rate is sufficient assuming it takes at least 15 minutes to get ready and arrive at the grocery store of the user's choice. The average user takes 40 minutes to shop at a grocery store [5], therefore, the user will have an updated grocery list by the time they arrive/begin shopping at the grocery store.

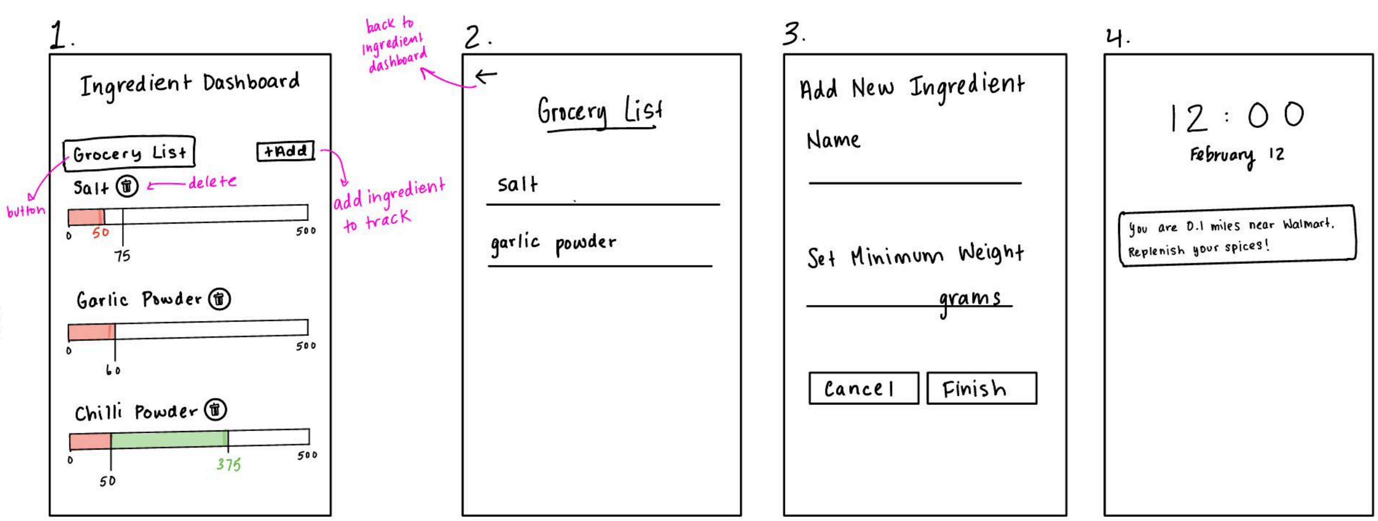

Figure 6: Mock Mobile Application Screen Wireframes

When a user opens the app, the first screen they will see is the "Ingredient Dashboard" (screen 1 in Figure 6). They will then have the ability to add/delete ingredients from the "Ingredient

Dashboard" and view the "Grocery List." If they choose to view the "Grocery List," they will be shown the current ingredients that are below their lower weight thresholds (screen 2 in Figure 6). The user can add an ingredient in the "Ingredient Dashboard" using the "+Add" button. The button directs the user to the "Add New Ingredient" screen. The screen will prompt the user to enter a spice name and lower weight threshold. The lower weight value entered must be between 0 - 500 grams and in whole grams.

Requirements	**Verification**
App has two main interfaces: an "Ingredient Dashboard" and a dynamic grocery list.	The application opens with the "Ingredient Dashboard". The grocery list can be navigated from the "Ingredient Dashboard" with a button and vice versa.
The "Ingredient Dashboard" can track up to 3 ingredients.	The "Ingredient Dashboard" on the mobile application displays three spices with individual progress bars indicating lower weight threshold and current weight progress. The "+ADD" button will become unresponsive after 3 spices have been added to the dashboard.
The app will have an "Ingredient Dashboard" interface that allows users to specify individualized lower weight thresholds for each spice. These weight thresholds are within the range of 0 - 500 grams and recorded in whole grams.	Add a spice to the "Ingredient Dashboard" using the "+ADD" button. The app will show the "Add New Ingredient" screen to enter the spice name and lower weight threshold. The lower weight threshold section of the screen will turn red until a valid whole value number is entered in the range from 0 to 500 grams. Once valid input is entered and the user clicks "Finish," a progress bar on the original "Ingredient Dashboard" will appear.
Each ingredient on the dashboard visualizes the recorded weight of the spice through a progress bar. The progress bar ranges from 0 - 500 grams with a marker indicating the user specified lower weight threshold. When the recorded spice weight falls below the lower weight threshold, the filled portion of the progress bar turns red. Otherwise it remains green.	A progress bar only shows up if there are spices added to the "Ingredient Dashboard". The spice's weight will be shown as a progress bar with a marker indicating its inputted lower weight threshold. Place a spice container that is above the lower weight threshold in a spot on the device. Withdraw the spice from the container until it is below its threshold. The filled portion of the progress bar should turn red after 15 minutes.

Ingredient dashboard receives updates on spices weight every 15 minutes. The "Ingredient Dashboard" data is stored in an online database.	Have a container already in a spot on the device and make sure it has already been added to the "Ingredient Dashboard". Remove or add spice into the container. After 15 minutes the progress bar in the "Ingredient Dashboard" should reflect the change in weight.
Dynamic grocery list synchronizes with "Ingredient Dashboard" and only ingredients that are below the lower weight threshold are displayed.	Have a container already in a spot on the device and make sure it has already been added to the "Ingredient Dashboard". Remove spice from the container. After 15 minutes the spice should show up in the "Grocery List" screen. Have a container already in a spot on the device and make sure it has already been added to the "Ingredient Dashboard". This spice needs to already be below its lower weight threshold and the "Grocery List" screen should include that spice. Add the spice to the container so that its weight goes above its threshold. After 15 minutes the spice should disappear from the "Grocery List" screen.

2C.II Sensing Subsystem

The sensing subsystem is made up of the RFIDs and load cells. The sensing subsystem is connected to the control and power subsystems via wires. The system uses RFIDs operating at 13.56MHz [13] to allow users to place spice containers in any slot while recognizing the spice and its lower weight threshold. The spices will be held in plastic airtight containers that weigh 481.94 grams and can hold up to 498.952 grams of spice. The load cells are used to measure the weight of the spices and send the information to the MCU. HX711 chips will be used in conjunction with the load cells to turn the voltage measured across the load cell into 24 bit 2's complement values.

Requirements	Verification
Load cells distinguish weights with a resolve of 0.5g (refer to tolerance analysis)	If 0.5g of the spice is taken out of the container, the spice's progress bar fill in the "Ingredient Dashboard" is reduced by 0.1% ((0.5/100) *100).

	Weigh a container and record its weight. Then weigh the container with its spice and record that weight. Subtract the container weight from the overall weight to get the weight of the spice. Place that same container and spice in a spot on the device, and make sure it has already been added to the "Ingredient Dashboard". After 15 minutes the calculated spice should be reflected in the progress bar in the "Ingredient Dashboard". Next, weigh out 0.5 grams of the spice and add it to the container. After 15 minutes the calculated spice plus the 0.5 grams weight should be reflected in the progress bar in the "Ingredient Dashboard".
Load cells measure the maximum weight of the spices to 500 grams each.	The load cell must be able to read 1kg. If the weight goes above the maximum threshold, the load cell will send invalid data. The mobile application will give a notification saying that the amount of the spice is over capacity. Weigh a container and record its weight. Then weigh the container with its spice and record that weight. Subtract the container weight from the overall weight to get the weight of the spice. The weight of the spice needs to be 500 grams. Repeat steps 2 and 3 until 500 grams is reached. Place that same container and spice in a spot on the device, and make sure it has already been added to the "Ingredient Dashboard". After 15 minutes the 500 grams should be reflected with a completely filled and green progress bar in the "Ingredient Dashboard" for the spice.
Only the 3 valid RFID tags will be read and recognized by the device.	To ensure only valid tags are read, a fourth tag will be used as an edge case. Weigh the container with its spice and record that weight. Place that same container and spice in a spot on the device, and make sure it has already been added to the "Ingredient Dashboard". After 15 minutes the weight of that spice should be reflected in the progress

	bar in the "Ingredient Dashboard". Remove that container and replace it with one of the same weight, but with an invalid tag. After 15 minutes, the system should recognize the invalid tag and the application sends a notification saying that the invalid device is placed onto the system
The 3 containers with RFIDs can be moved between the 3 spots. The spot's readers allow for the right comparison between the spice's weight and its specific lower threshold weight.	Weigh the container with its spice and record that weight. Place that same container and spice in a spot on the device, and make sure it has already been added to the "Ingredient Dashboard". After 15 minutes the weight of that spice should be reflected in the progress bar in the "Ingredient Dashboard". Remove that container and remove a teaspoon of that spice. Place that container with its new weight on a different spot. After 15 minutes the new weight of that spice should be reflected in the progress bar in the "Ingredient Dashboard". Repeat that process for the last spot and for the rest of the containers and their RFID.

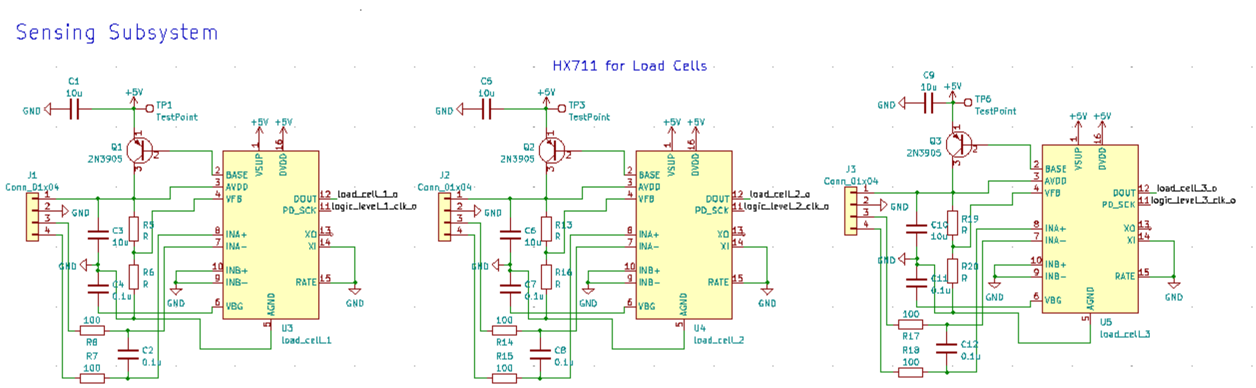

Figure 7: Sensing Subsystem Schematic – HX711 for Load Cells

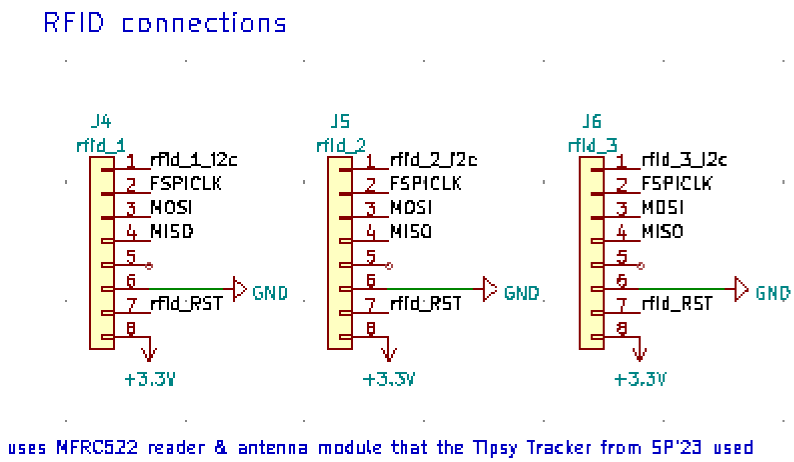

Figure 8: Sensing Subsystem Schematic – RFID Connections

2C.III Control Subsystem

The control system is made up of the ESP32 MCU. The control subsystem is connected to the power and sensing subsystems using wires. The subsystem connects the user interface subsystem to the sensing subsystem and the hardware interface subsystem. The MCU will use its built-in Wifi protocol to send data from the load cells and RFIDs to the app. It also sends data received from the app to the LEDs. The MCU pulls weight data captured by the load cells every 15 minutes.

Requirements	Verification
The output weight from the HX711 chip gets translated through the logic level chip shifter from 5V to 3.3V. This 24 bit 2's complement output value [3] will be converted to decimal values.	Weigh a container and record its weight. Then weigh the container with its spice and record that weight. Subtract the container weight from the overall weight to get the weight of the spice. Place that same container and spice in a spot on the device, and make sure it has already been added to the "Ingredient Dashboard". After 15 minutes the same calculated spice weight should be reflected in the progress bar in the "Ingredient Dashboard". Take an oscilloscope and place one probe on Vin of the logic level chip shifter and the second probe to ground. The oscilloscope should measure 5V. Move the probe from Vin to Vout of the logic level chip shifter and the oscilloscope should show 3.3V.

The data pulled from the HX711 chip to the MCU is transferred to the Firebase database every 15 minutes.	Have a container already in a spot on the device and make sure it has already been added to the "Ingredient Dashboard". Remove or add spice into the container. After 15 minutes the progress bar in the "Ingredient Dashboard" should reflect the change in weight.

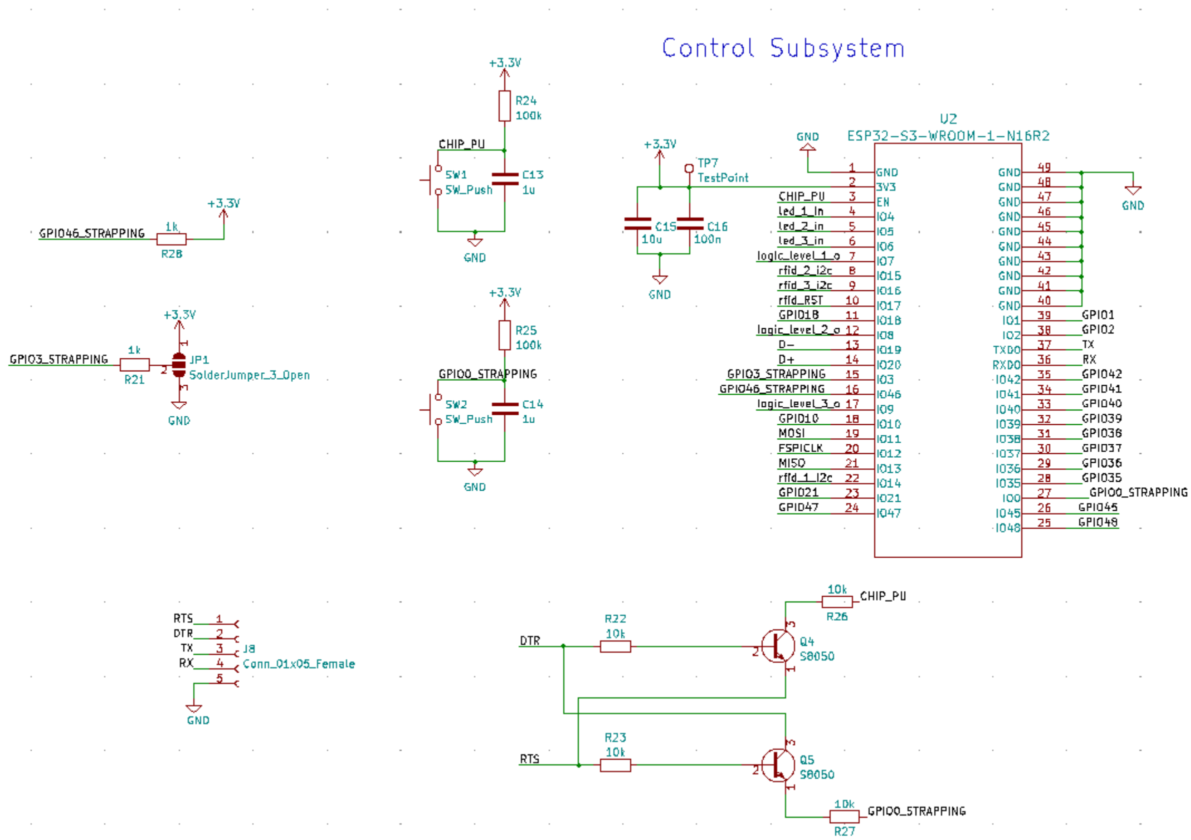

Figure 9: Control Subsystem Schematic

2C.IV Power Subsystem

A wall power adapter that converts 120V AC to 5V DC will be used for the load cells, HX711 chip and RFIDs. A voltage regulator will be used to step down the 5V to the 3.3V needed by the MCU. A voltage divider will be used to step down the 5V to the 2.3V needed by the LEDs. A logic level chip shifter will be used between the HX711 chip and the ESP32 to translate the 5V signals to 3.3V signals needed by the ESP32.

Requirements	Verification
The LM317 configured as a 5V regulator must be able to provide 3.3V from 5V at 620mA continuous current, with a voltage tolerance no greater than ±0.3V. The voltage tolerance is ±0.3V for the ESP32 as indicated on the datasheet [6].	When on the PCB, the voltage output waveform of the 5V regulator will be monitored with all peripherals ON using an oscilloscope. The nominal voltage must be 5V with a maximum ripple of 0.3V. The ripple must also be less than 0.3V as the relay is turned ON and OFF.

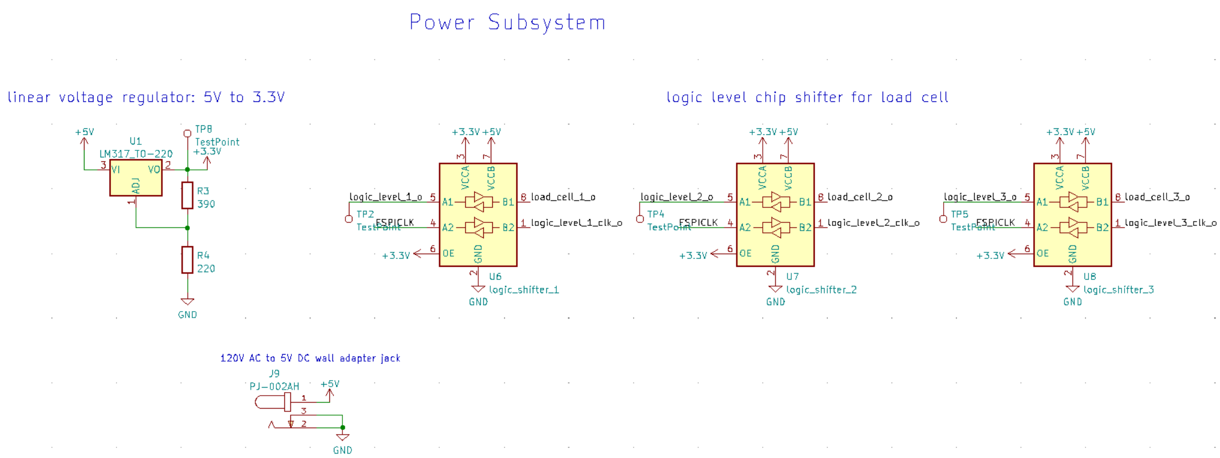

Figure 9: Power Subsystem Schematic

2C.V Hardware Interface Subsystem

The subsystem consists of an LED which requires 2.3V±0.3V at each spot [10]. Using a two resistor voltage divider, the 5V will be scaled down to 2.3V. The LEDs will provide a visual representation to see which spices have run low. The microcontroller will communicate with the LEDs via wires to display red when a spice is below the user specified lower threshold. The LEDs will be enclosed within a see-through covering to prevent spices and dirt from getting on it.

Requirements	Verification
The LEDs turn on and remain red until the load cell reads that the spice weight is above its lower weight threshold	Place a spice container that is above the lower weight threshold in a spot on the device. Withdraw the spice from the container until it is below its threshold. If the container is still in the spot, the red LED for that spot should turn on after 15 minutes. Repeat this for each spot.

The LEDs will remain off when the record weight is between 0 - 482 grams to account for the absence of a spice container as the weight of the spice containers is 481.92 grams.	Place a spice container that is below the lower threshold so that the red LED turns on for that spot. Removing the spice container should turn off the LED after 15 minutes. Repeat this for each spot.

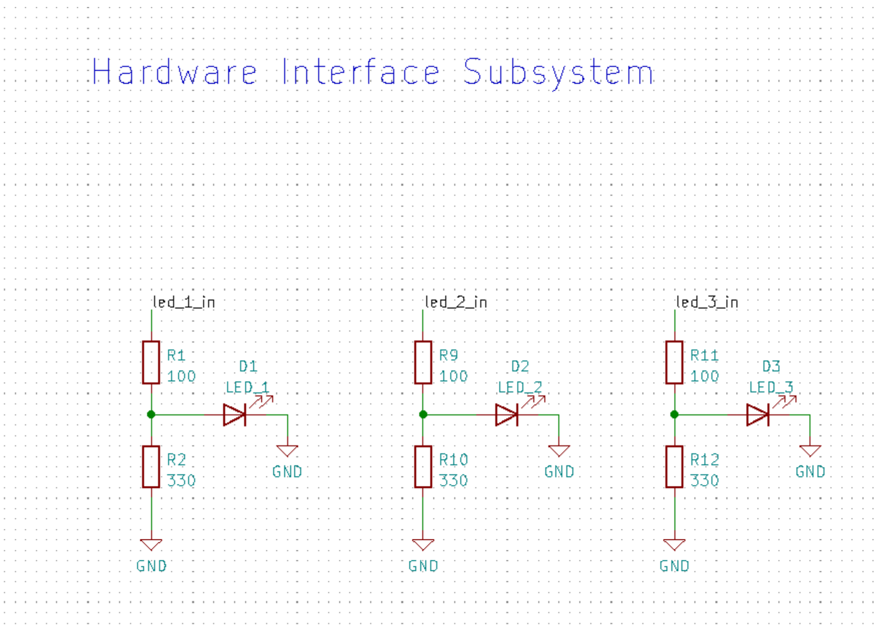

Figure 10: Hardware Interface Subsystem Schematic

2D. Tolerance Analysis

Load Cell Tolerance Analysis:

The system needs to measure spices and compare it against the user specified lower weight threshold. The digital grocery list interface adds/removes spices based on whether their recorded weight is below its lower weight threshold.

The weight recorded by the load cell includes the spice container and the spice itself. The container used weighs 481.92 grams. The contents of the container range from 100g to 500g. For this system, a low spice level for a full 500g spice is around 200g. However, some users may want to wait until the spice reaches a lower weight, like 50g, to replenish, while others will replenish early at a higher weight, like 300g.

Therefore, we need a weight sensor that can measure up to 1 kg (~481.92 + 500).

However, a challenge comes with determining the tolerance of the increments of weight the threshold can be set at. Let's assume that the user is tracking cinnamon, which usually comes in 500g containers. The user may not care if the threshold is 200g or 202g. In this instance the user may be satisfied with 10g increments, for example 200g, 210g, 220g, etc. However, if saffron is the tracked spice, then the user may want increments of 1g.

Therefore, the system requires a load cell that can resolve 1g of ingredients, in the worst case. The system will be using a load cell which is rated for 1 kg with a resolve of 0.5g.

Each load cell has a sensor with 4 outgoing wires. These wires are connected to VDD, GND, INA^+, INA^- of the HX711 chip. The HX711 is a 24-Bit Analog-to-Digital Converter (ADC) for weight scales. The HX711 allows for two load cells (Channel A and Channel B). The HX711's "input multiplexer selects either Channel A or B differential input to the low-noise programmable gain amplifier (PGA). Channel A can be programmed with a gain of 128 or 64, corresponding to a full-scale differential input voltage of ±20mV or ±40mV respectively, when a 5V supply is connected to a VDD analog power supply pin. Channel B has a fixed gain of 32" [3].

This is a single point load cell and has 2 circular holes drilled in the middle. This design allows for flexibility and ensures accurate readings even if the weight is unevenly distributed. This design feature is essential "as it removes the effects of human error in placement" [15] of weights onto the load cell. The load cells have four mounting holes in total, two on each side for installation [10]. The HX711 requires 5V for the VDD. Therefore, the load cells will use 5V to maintain a consistent VDD across interacting components.

The output weight from the HX711 is represented as a 24 bit 2s complement value. The output will be sent to the MCU. Since each HX711 operates at 5V and the MCU operates at 3.3V, the design requires a logic level chip shifter to drop the voltage accordingly. The output signal received by the MCU is 0 - 3.3V, where 0V represents the 0 and 3.3V represents 1. The output is shifted out from pin DOUT. When 25~27 positive clock pulses are applied to the Serial clock input (PD_SCK) pin, data is shifted out from the DOUT output pin starting with the MSB bit first. Each PD_SCK pulse shifts out one output bit until all 24 bits are shifted out [3].

The load cell needs to be able to read up to 1kg with a resolve of 0.5g. One level will represent 0.5 grams. Since we have 1000g, the design requires 2000 levels (1000/0.5 = 2000). In order to accurately represent the weight, the chip needs to have at least 11 bits of digital logic ($\log_2 2000 = 11$). Therefore, the HX711 is the right choice as it is a 24-Bit Analog-to-Digital Converter and meets the system's needs.

Linear regulators Tolerance Analysis:

Part	**Worst Case Current Draw @ 3.3V**	**Comment**
ESP32-S3	500mA	Page 56 of the datasheet
LED	120mA	
Total	620 mA	

Table 1: Current Analysis

Variable	**Value**	**Comments**
$\max(T_j)$	125 C	Maximum operating junction temperature of LM317TO-200 package [12]
i_{out}	620 mA	Maximum current draw of components on 3.3V power
vin	5 V	Output of wall power output adapter
vout	3.3 V	Operating voltage of components
θ_{jc}	5 C/W	Junction-to-case thermal resistance of LM317TO-200 package found in datasheet [12]
θ_{ca}	32.9 C/W	The regulator datasheet conveniently specifies θ_{ja}= 37.9 C/W. [12]
T_a	30 C	Let's assume for a warm board

Table 2: Variable Values for Formula Estimating Junction Temperature

The formula below is used to estimate the junction temperature:

$$P_D = iout \times (vin - vout)$$
$$T_{ja} = P_D \times \theta_{ja}$$
$$\Rightarrow T_j - T_a = P_D(\theta_{jc} + \theta_{ca})$$
$$\Rightarrow T_j = P_D(\theta_{jc} + \theta_{ca}) + T_a$$
$$\Rightarrow T_j = i_{out}(vin - vout)(\theta_{jc} + \theta_{ca}) + T_a$$

Formula in regards with system values:

$$T_j = i_{out}(vin - vout)(\theta_{jc} + \theta_{ca}) + T_a = 620mA(5 - 3.3)(5 + 32.9) + 30 = 69.95\,C$$

$$T_j \approx 69.95C < max(T_j)$$

This process of estimating T_j is not meant to be very exact. Since the estimated temperature is 30 degrees below the maximum, the LM317TO-200 package is an appropriate linear regulator to use.

3 Cost Analysis and Schedule

3A Cost Analysis

The average salary for an electric engineer graduate in Illinois is $49/hr [17] and the average salary for a computer engineer graduate in Illinois is $56/hr [16]. Taking the average of these salaries, the salary for each member's labor cost is $52.5/hr. With 8 weeks left in the semester, each team member will work for approximately 20 hours per week for a total of 160 hours per person. The total labor cost for one member would be 52.5 * 2.5 * 160 = $21,000. Therefore, the total labor cost for the team is $21,000 * 3 = $63,000. Table 3 below shows an itemized list of all costs for this project.

The parts obtained by the Electronic Supply Store (ESS) are free and have been marked as $0 in Table 3. The total cost for parts ordered is $92.21. Thus, the total cost for this project is $63,000 + $92.21 = $63,092.21.

#	Item	Count	Unit Price	Total Cost	Source
1	Load Cells	4	-	19.99	Amazon
2	RFID Stickers	3	2.95	8.85	Digikey
3	RFID readers	4	5.14	20.56	Digikey
4	Spice Container	3	-	21.99	Amazon
5	HX711	5	-	10.66	Walmart
6	Logic Level Shifter	4	0.63	2.53	Digikey
7	Power Barrel Connector	1	-	0.62	Digikey
8	Resistors 100 ohm	14	0.02	0.32	Digikey
9	Resistors 220 Ohm	6	0.10	0.60	Digikey
10	Resistors 390 Ohm	6	0.10	0.60	Digikey
11	Power Supply Adapter	1	-	5.49	Amazon

	ECE Electronic Supply Shop				
#	Item	Count	Unit Price	Total Cost	Source
1	Resistors 330 Ohm	8	-	-	ESS
2	Resistor 1k Ohm	7	-	-	ESS
3	Resistor 10k Ohm	9	-	-	ESS
4	Resistor 100k Ohm	7	-	-	ESS
5	Capacitors 0.1 uF	12	-	-	ESS
6	Capacitors 1 uF	7	-	-	ESS
7	Capacitors 10 uF	12	-	-	ESS
8	LM320T220	1	-	-	ESS
9	ESP32-S3 MCU	1	-	-	ESS
10	SS8050-G	2	-	-	ESS

Table 3: Cost Analysis

3B. Schedule

Week	Task	Person	
2/12 - 2/18	Finish design document	All	
	Finalize parts	All	
	Meet with Gregg to start building	All	
	Start board design (development board)	All	
	Start designing PCB	Anju	
	Wireframing app	Sanjana	Nynika
2/19 - 2/25	Continue designing PCB	Anju	
	Start initial app design	Sanjana	Nynika
	Order parts	All	
	Prepare for design review	All	

2/26 - 3/3	Continue app development	Sanjana	Nynika
	Finish first round of PCB design	Anju	
	Design review w/ professors and TA	All	
3/4 - 3/10	1st order PCB	All	
	Continue app development	Sanjana	Nynika
3/11 - 3/17	Spring Break		
3/18 - 3/24	Functional app DONE	Sanjana	Nynika
	Start soldering	All	
	Connect app to ESP32 via Wifi	All	
	Start load cell + RFID functionality	All	
	2nd order PCB	All	
3/25 - 3/31	3rd order PCB	All	
	Complete LED functionality	All	
	Complete load cell functionality	All	
4/8 - 4/14	Run thorough test to ensure app, LED, RFID, and load sensors all work properly	All	
	Run through practice demos for next	All	
4/15 - 4/21	Mock demo	All	
	Begin final paper	All	
4/22 - 4/28	Final demo	All	
4/29 - End	Final presentation	All	
	Final paper (due 5/1)	All	
	Lab notebook (due 5/2)	All	

Table 4: Schedule

4 Ethics and Safety

The project will comply with the IEEE and ACM Code of Ethics. The safety risks that this device poses are that it has the possibility of being a fire hazard. The electrical parts of the device including their connections will be enclosed within the enclosure. This means that other than the power port, any loose spices will not be able to reach the electrical components and cause a short circuit or electrocute anyone. The LEDs will be enclosed within a see through covering to prevent spices and dirt from getting on it. Thermal sensors can be added near components that have the possibility of overheating. Places that would benefit from a thermal sensor are the ESP32, linear voltage regulator, and the HX711 chips. If the thermal sensor reads a value higher than a preset temperature, the power for the entire system will shut off. Thus reduces the risk of a fire hazard.

This product does not pose a risk for cross contamination. The spices are placed within plastic airtight containers. Therefore if any containers fall down/move, the spices are safely contained within the container. If any spices fall onto the device, a user can use a wipe to remove the mess.

There are also the possibilities for potential ethical and privacy concerns surrounding the app accompanying the device. The app will require access to the user's location, however the app will not be using the user's information for malicious purposes. This aligns with the ACM Code of Ethics stating, "Computing professionals should only use personal information for legitimate ends and without violating the rights of individuals and groups [...] taking precautions to prevent re-identification of anonymized data or unauthorized data collection [...] and protecting it from unauthorized access and accidental disclosure" [1]. As this app will only be available on Apple devices, the user will have the option to choose their privacy setting in their Apple device settings. The team will follow all safety guidelines, standards, and regulations while completing our project. The team aims to create a device and app that makes kitchen dry ingredient organization easier, while prioritizing safety and the IEEE and ACM Code of Ethics.

5 Citations

[1] "ACM Code of Ethics." acm.org. Accessed Feb. 2, 2024. [Online.] Available: https://www.acm.org/code-of-ethics

[2] Adafruit. "13.56MHz RFID/NFC Sticker - NTAG203 Tag ." NTAG203 Tag datasheet.

[3] AVIA Semiconductor. "24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales." HX711 datasheet.

[4] B. Howard and P. Rowland. "Linear Regulator Design Guide For LDOs." Accessed: Feb. 20, 2024. [Online]. Available: https://www.ti.com/lit/an/slva118a/slva118a.pdf

[5] C. Tretheway. "The Average American Spends This Much Driving to the Grocery Store." Nasdaq. Accessed: Feb. 18, 2024. [Online]. Available: https://www.nasdaq.com/articles/the-average-american-spends-this-much-driving-to-the-grocery-store

[6] Espressif, "ESP32-S3-WROOM-1 ESP32-S3-WROOM-1U", ESP32-S3-WROOM-1 ESP32-S3-WROOM-1U datasheet, Spet. 2021 [Revised Nov. 2023]

[7] "Firebase Realtime Database". Firebase. [Online]. Available: https://firebase.google.com/docs/database

[8] "IEEE Code of Ethics." ieee.org. Accessed: Feb. 2, 2024. [Online.] Available: www.ieee.org/about/corporate/governance/p7-8.html

[9] Liteon. "Lite-On Electronics, Inc." LTL-4223 datasheet.

[10] "Load Cell Guide." Phidgets. Accessed: Feb. 18th, 2024. [Online]. Available: https://www.phidgets.com/docs/Load_Cell_Guide

[11] R. Jacobi and U. Denk, "Frequently asked questions for TRF7970A and TRF7964A devices." Tech. Report. Jan. 2018 [Revised Mar. 2019].

[12] Texas Instruments. "LM317 3-Terminal Adjustable Regulator." LM317 datasheet. Sept. 1997 [Revised Apr. 2020].

[13] Texas Instruments. "TRF7964A Multiprotocol Fully Integrated 13.56-MHz RFID Reader and Writer IC." TRF7964A datasheet. May 2012 [Revised Mar. 2020].

[14] T. Ozbun. "Grocery shopping: U.S. consumers' weekly trips per household 2006-2022." *statista.com*. Accessed: Feb. 14, 2024. [Online.] Available: https://www.statista.com/statistics/251728/weekly-number-of-us-grocery-shopping-tripsper-household/

[15] "What Is A Single-Point Load Cell, And How Does It Work?." Flintec. Accessed: Feb. 18th, 2024. [Online]. Available: https://www.flintec.com/weight-sensors/load-cells/how-does-a-single-point-load-cell

[16] "Computer Engineering Salary in Illinois." ZipRecruiter. Accessed: Feb.18th, 2024. [Online]. Available: https://www.ziprecruiter.com/Salaries/Computer-Engineering-Salary--in-Illinois

[17] "Electric Engineering Salary in Illinois." ZipRecruiter. Accessed: Feb.18th, 2024. [Online]. Available: https://www.ziprecruiter.com/Salaries/Graduate-Electrical-Engineer-Salary--in-Illinois