Smart Snack Dispenser (SSD)

ECE 445: Senior Design Laboratory

Design Document

Team No.23

Eric Nieto Gonzalez || Elinor Simmons || Adam Kramer

(enieto5@illinois.edu)||(elinors2@illinois.edu)||(adamlk2@illinois.edu)

TA: Surya Vasanth

Professor: Yang Zhao

March 6th, 2025

Table of Contents

1. Introduction

1.1 Problem

Difficulty in controlling snack portions is a common issue that can lead to overeating and unhealthy eating habits. Mindless snacking, especially during activities like working, studying, or watching TV, often results in consuming more than intended. Research indicates that individuals make over 200 food-related decisions daily, many unconsciously, contributing to unintentional overeating. Currently, the market lacks devices specifically designed to address this issue, leaving individuals to rely on willpower or manual portioning methods, such as separating snacks into smaller bags. While some automated portioning equipment exists, they are primarily intended for industrial food processing rather than personal use.

Without a structured approach, people often struggle to regulate their intake, leading to issues such as weight gain, unhealthy eating patterns, and difficulty in maintaining a balanced diet. Implementing mindful eating practices has been shown to improve eating behaviors, dietary intake, and body weight. However, without practical tools to assist in portion control, individuals may find it challenging to consistently apply these practices.

1.2 Solution

The smart snack dispenser provides a convenient and user-friendly way to regulate snack portions, reducing the reliance on willpower alone. By allowing users to pre-set portion sizes, the device ensures controlled snack distribution, helping to prevent mindless overeating. Designed as a home appliance that plugs into a wall outlet, the dispenser seamlessly integrates into daily routines, making portion control effortless.

With a curated selection of three snacks—M&M's, Skittles, and peanuts—the machine caters to a balance of sweet, salty, and protein-rich options, appealing to various taste preferences while supporting mindful eating habits. This structured approach is particularly beneficial for individuals monitoring their calorie intake, maintaining a balanced diet, or practicing portion control as part of a healthier lifestyle.

1.3 Visual Aid

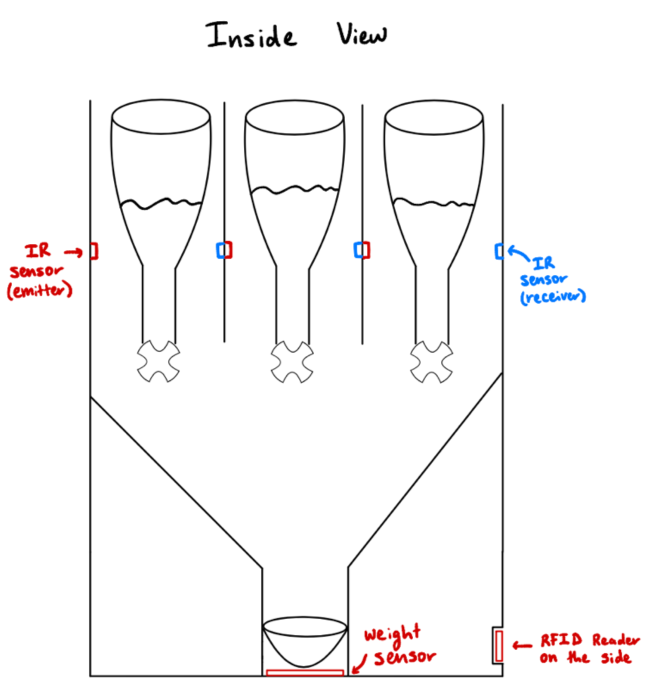


Figure 1: Visual Aid Internal View

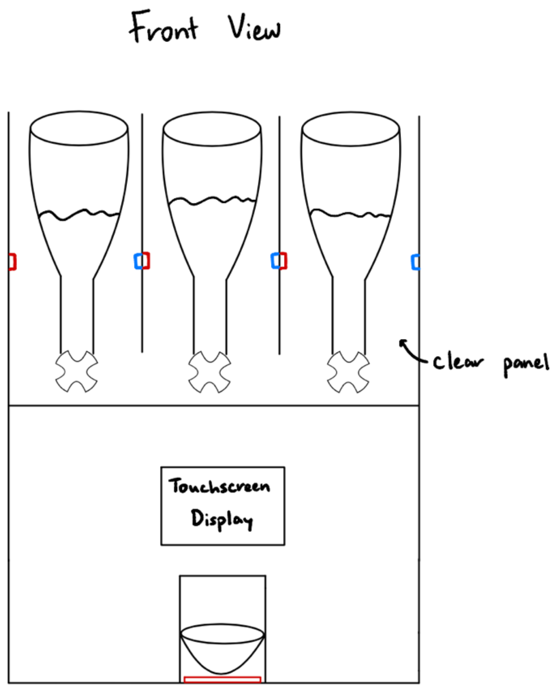


Figure 2: Visual Aid Front View

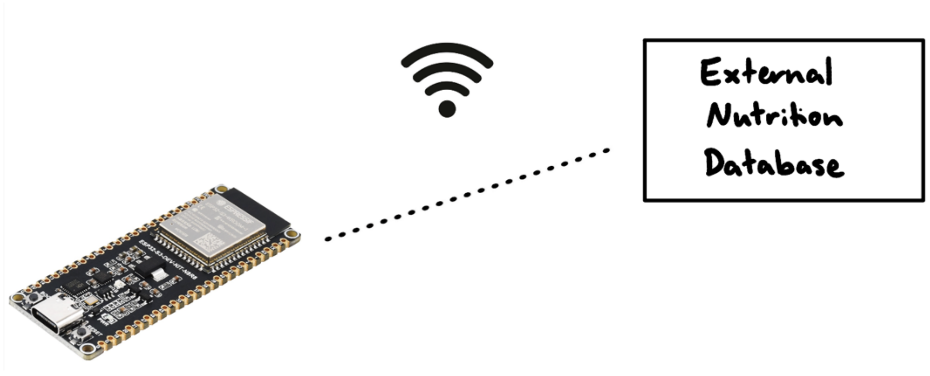


Figure 3: Wi-Fi Communication to Nutrition Database

1.4 High Level Requirements

For this project to be considered successful, specific parameters and measurable goals must be met. These goals define the necessary performance benchmarks and ensure objective evaluation of the outcomes:

- Accuracy: The machine must dispense the correct portion within a 15% tolerance or less. The machine must also correctly keep track of the nutrients that are in each portion.
- Speed: The machine must dispense the snack within 30 seconds or less after the snack has been selected. It also must register user input immediately as it is selected on the touchscreen.
- Usability: The UI must be smooth and organized. It should be easy for the user to dispense a snack or find their personalized data.

2. Design

2.1 Block Diagram

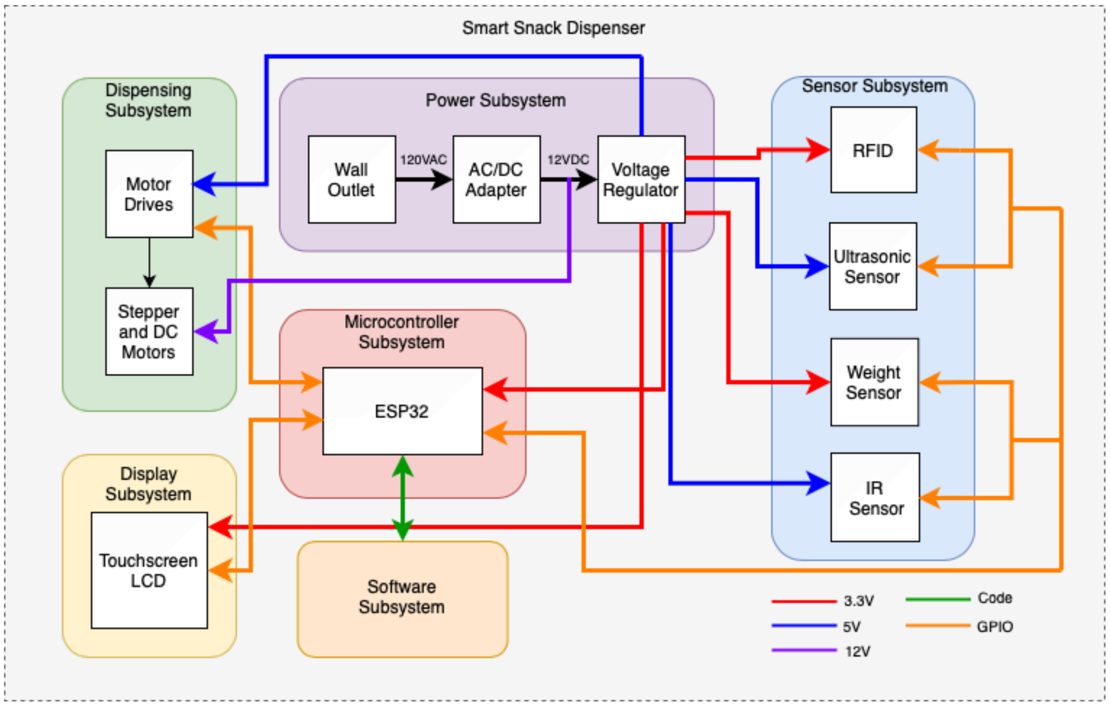


Figure 4: Block Diagram

The block diagram includes five subsystems. This includes the dispensing, power, sensor, software, display, and microcontroller subsystems. The dispensing subsystem consists of our motors and dispensing mechanisms to output the snacks. The power subsystem includes our AC/DC wall adapter and voltage regulators to power the other subsystems. The sensor subsystem includes sensors to RFID to allow multiple users, the ultrasonic sensor to check for bowl placement, the weight sensor to weigh desired portions, and the IR sensor to notify when the snack containers need to be refilled. The software subsystem contains all our internal features that will be implemented through code and programmed onto the microcontroller. The touchscreen display subsystem is used for our user interface. The microcontroller ties all our other subsystems together and allows them to communicate through the controllers GPIO pins.

2.2 Physical Design

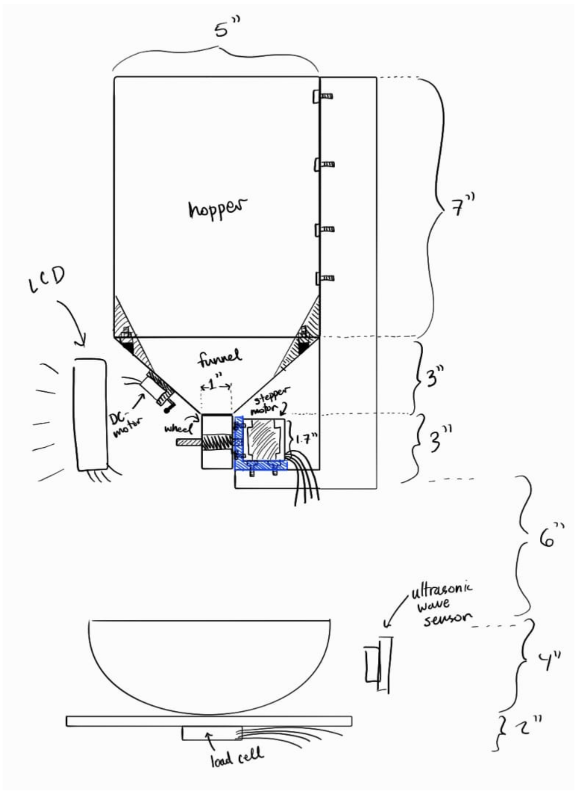
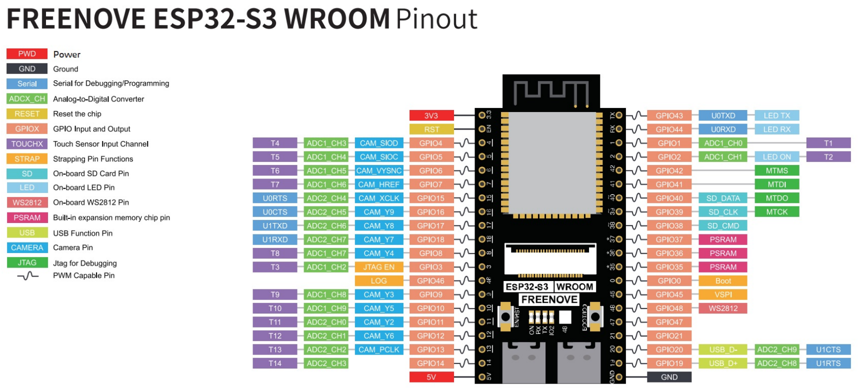


Figure 5: Internal and Side View of Machine

The dispenser will have overall dimensions of 12 [in.] x 25 [in.] x 16 [in.] (L x H x W), housing all internal components efficiently. Each snack storage compartment will measure 5 [in.] x 10 [in.] x 5 [in.]. Additionally, the bowl input will have dimensions of 6 [in.] x 6[in.], ensuring a suitable space for snack collection.


FREENOVE ESP32-S3 WROOM Pinout
PWD Power
GND Ground
Serial Serial for Debugging/Programming
ADCX_CH Analog-to-Digital Converter
RESET Reset the chip
GPIOX GPIO Input and Output
TOUCHX Touch Sensor Input Channel
STRAP Strapping Pin Functions
SD On-board SD Card Pin
LED On-board LED Pin
WS2812 On-board WS2812 Pin
PSRAM Built-in expansion memory chip pin
USB USB Function Pin
CAMERA Camera Pin
JTAG Jtag for Debugging
PWM Capable Pin
ESP32-S3 WROOM
FREENOVE

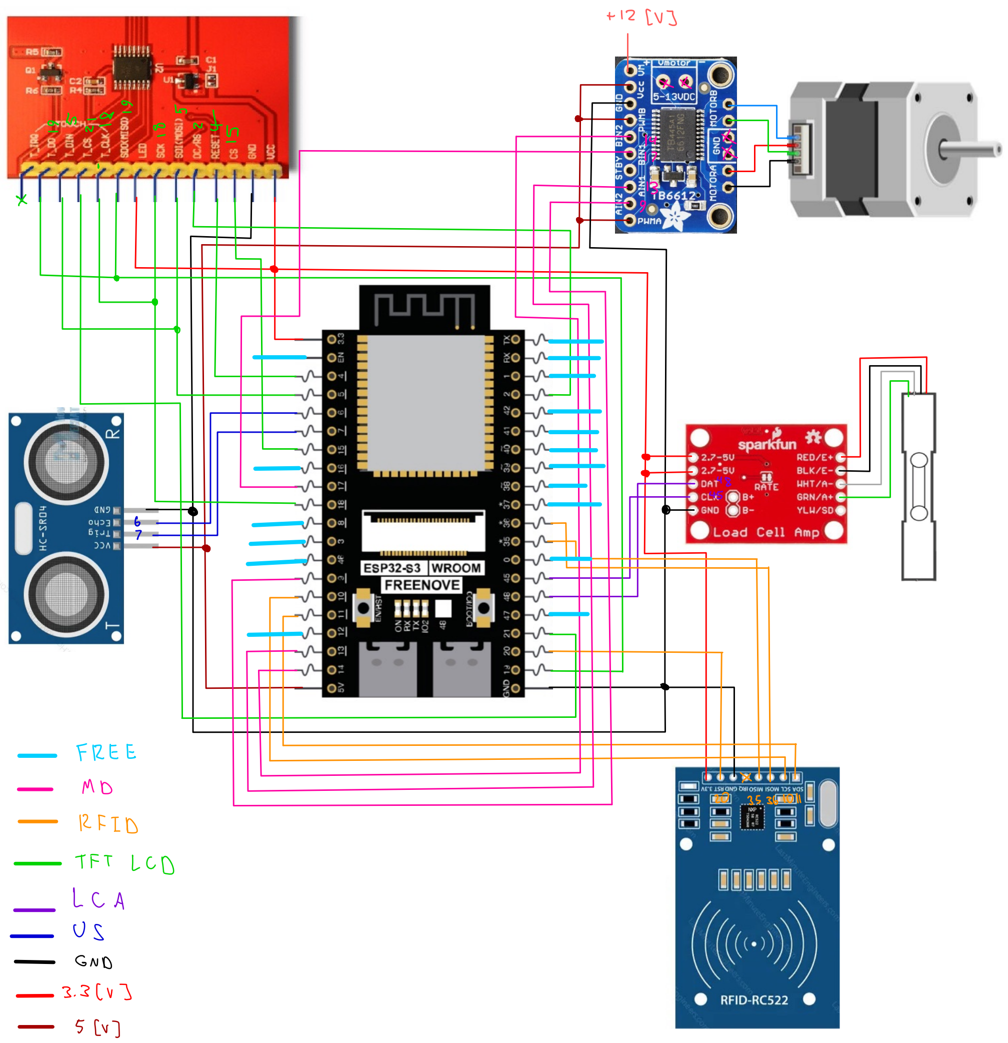


Figure 6: Current Pin Layout for Prototye

The current pin layout includes power connections (12V, 5V, 3.3V), ground (GND), and GPIO pins for interfacing with a TFT LCD and other peripherals. This creates a working embedded system design that can be programmed to have a GUI and work as one unit.

2.3 Subsystem Overview and Requirements

2.3.1 Dispensing Subsystem

The dispensing subsystem ensures snacks are delivered accurately and efficiently, preventing jams or over-dispensing. It will be powered by three stepper motors, each controlling a snack hopper. These motors are driven by motor drivers, which are controlled by the microcontroller. Each hopper will also have a DC motor to provide vibration, aiding the movement of snacks into the cutouts of a rotating dispensing wheel. The wheel's cutouts are spaced 90 degrees apart to maintain precision during dispensing. As the wheel rotates, the snacks fall into the cutouts and are deposited into the weighing area. If there were more than four cutouts, the risk of jamming or accidental spillover could increase, with snacks potentially spilling into the bowl when not intended.

The user will be able to choose if they want to dispense one, two, or three types of snacks. The minimum and maximum amount the user can dispense is 15 grams and 100 grams for each snack, respectively. After the portion is chosen for each snack, the machine will dispense them one at a time to ensure accurate weighing and nutrition tracking. The snacks will be stored in separate hoppers which each feed into a small opening that their respective wheels can grab from. The DC and stepper motors will be powered through the 12V input that is provided from the wall, and the motor driver will operate at 5V to control the stepper and dc motors' motion.

Requirements	**Verification**
• Stepper motor must dispense snacks within 30 seconds or less and spin in the correct direction.	• Record the time of a dispense with a stopwatch and see if the motor is spinning in the correct direction. • Accuracy will be checked through our code and comparing real results with the datasheet provided.
• Wheels can produce the ideal number of pieces for each snack on average. • Wheels minimize jamming	• Create and test wheel designs to ensure the right amount is being

	grabbed per spin by running multiple dispenses. • Test the dispensing to see how often the dispenser will jam.
• DC motor must be able to spin a small weight fast enough to provide the necessary vibration to avoid snack jamming.	• Run a dispense and check if the containers are vibrating.

Table 1: Dispensing Subsystem R&V

2.3.2 Microcontroller Subsystem

The ESP32 microcontroller serves as the central hub for processing inputs from the sensor subsystem and controlling the outputs of the snack dispenser. It is responsible for receiving data from various sensors and making real-time decisions based on this information. For example, when an RFID scan is detected, the ESP32 verifies the user's profile and retrieves their personalized snack settings to be shown in the LCD. The microcontroller also connects the ultrasonic sensor with the LCD and dispensing subsystem. If the ultrasonic sensor confirms that a bowl is present, the microcontroller signals the motor subsystem to dispense the selected snack, and if there is no bowl present, the microcontroller signals the LCD to display a notification. The weight sensor ensures accurate portion control by sending data to the ESP32, which adjusts the motor speed accordingly. Additionally, the IR sensor will alert the microcontroller to send a low stock notification to the display. The ESP32 also holds the code that manages the software subsystem. The EP32 will also communicate to an external database that hold snack nutrition information over Wi-Fi.

Requirements	Verification
• All code will be stored within the ESP32 to ensure full independence.	• Buy a chip that has enough local storage for the necessary programming.
• ESP32 must relay communication between subsystems when signals are sent through the GPIOs.	• Run dispenses and verify that the subsystems are communicating properly.

Table 2: Microcontroller Subsystem R&V

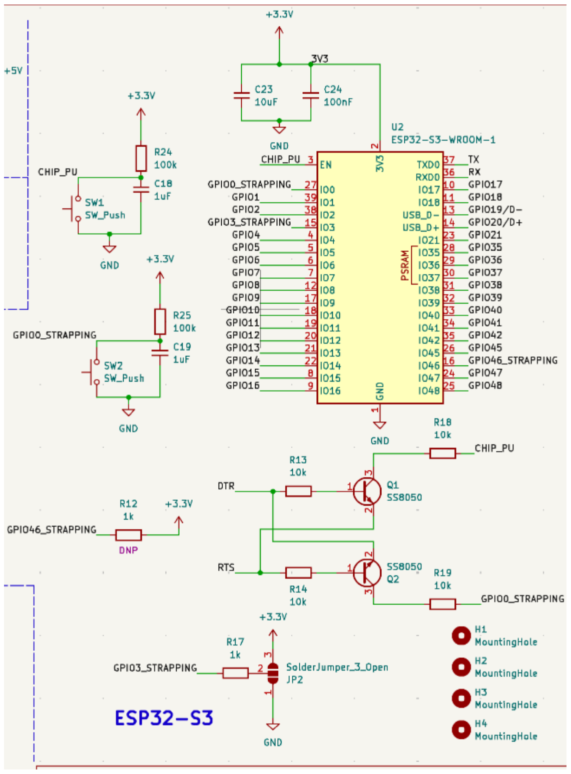


Figure 7: Schematic of Microcontroller Subsystem

2.3.3 Sensor Subsystem

The sensor subsystem features four different sensors. The first sensor is the RFID (RC522), which allows each user to have their own profile. The RFID is responsible for verifying and storing the tag ID and information. The next sensor is the ultrasonic sensor (HC-SR04), which is used to make sure a bowl is present before dispensing. The ultrasonic sensor will be able to read up to the length of the bowl input slot, when a bowl is not present. When a bowl is placed, this will shorten the readable distance, notifying the ultrasonic sensor that a bowl was placed. The next sensor is the weight sensor which is used to measure our desired portion. This includes an HX711 chip and a 500g load cell (SEN-14728). The load cell is used to measure the weight of our snacks through the stress applied to the cell and the HX711 is used to convert this information into readable data. The last sensor is an IR sensor to monitor stock level. This sensor includes an emitter to send out an IR beam of light and a receiver that is sensitive to that light. When snacks are present, this receiver should not be able to sense the beam, but once snacks are below this level, the receiver should be able to sense a signal. The ultrasonic and IR sensors will operate at 5V, while the other sensors will operate at 3.3V. All the sensors will relay information through the GPIO pins on our microcontroller.

Requirements	**Verification**
• The RFID must read the tag and display the correct information associated with that tag.	• Check that the correct user is displayed when the tag is scanned. • The data collected will be compared with the datasheet of the RFID alongside the tags where the correct information can be read.
• Ultrasonic sensor must recognize when that distance is shorted due to the placement of a bowl.	• Check the code output terminal to see if the sensor is reading the correct distance when a bowl is not present. • Place a bowl and check the output terminal to see if it reads a shorter distance.
• Weight sensor must measure the correct amount of weight within 15% and be able to tare weight of the bowl.	• Compare the weight sensor readings with a commercial food scale. • Measure the weight of the bowl and then check to see if that weight is subtracted from the weight sensor reading at the end of dispensing.
• The receiver must be able to sense the IR beam from across the length of the container. • The receiver must not sense the IR beam when snacks are present, so a false notification is not sent.	• Place the IR sensors the correct distance away from each other and check the code terminal to see verify that there is a reading. • Place an object between the two sensors and check that there is no reading from the receiver.

Table 3: Sensor Subsystem R&V

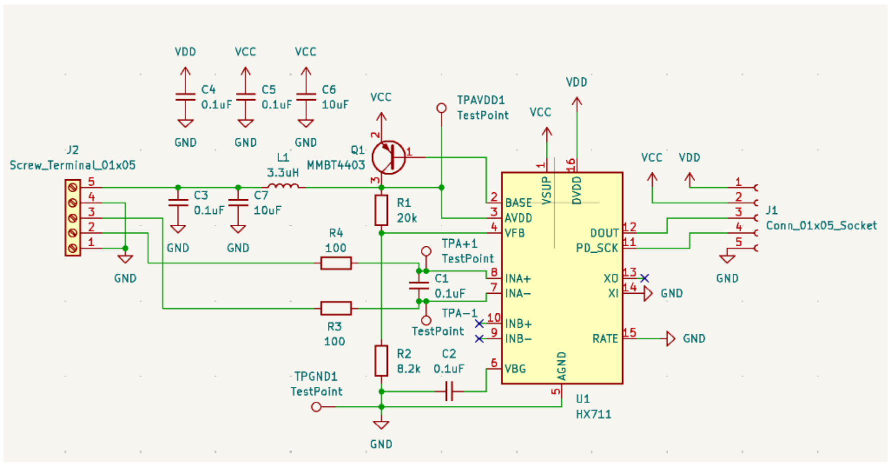


Figure 8: Weight Sensor (HX711) Schematic

2.3.4 Touchscreen LCD Subsystem

The touchscreen LCD serves as the primary interface for users to interact with the smart snack dispenser, providing real-time updates and controls. It is directly linked to the ESP32 microcontroller, which processes inputs from various sensors and delivers relevant information to the display. When a user scans their RFID tag, the LCD retrieves and displays the user's personal settings and nutrition tracking when necessary. If the user tries to dispense without a bowl present, a notification should appear on the display to place a bowl. As snacks are dispensed, the weight sensor provides live portion measurements, which are displayed on the screen to confirm accuracy. Once the snack level goes below the IR sensor, a notification will be shown on the LCD to alert the user to refill soon. Additionally, the LCD should display a notification if the user attempts to dispense a snack when their calorie limit has been reached for the day. By integrating all these functionalities, the touchscreen LCD enhances usability, making the snack dispenser both intuitive and efficient for users managing their dietary habits. We will be using the MSP4022 display, and it will be powered with 3.3V. The LCD will communicate to the rest of the sensors through our microcontroller's GPIO pins.

Requirements	Verification
• The LCD correctly shows values collected from various sensors.	• Verify that the values placed on the screen are correct and match the measured values. • Ensure that the value placement is within reason and is readable.

• The display should be capable of user touch to be able to select a variety of options.	• Write the necessary code that is needed to make the touch screen function. • Verify that touch location is within reason and does not register at a different part of the screen.
• The LCD should display the correct notifications when necessary.	• Perform scenarios where notifications are necessary, such as not placing a bowl or the containers need refilled and ensure that these notifications are displayed on the screen.

Table 4: Touchscreen LCD Subsystem R&V

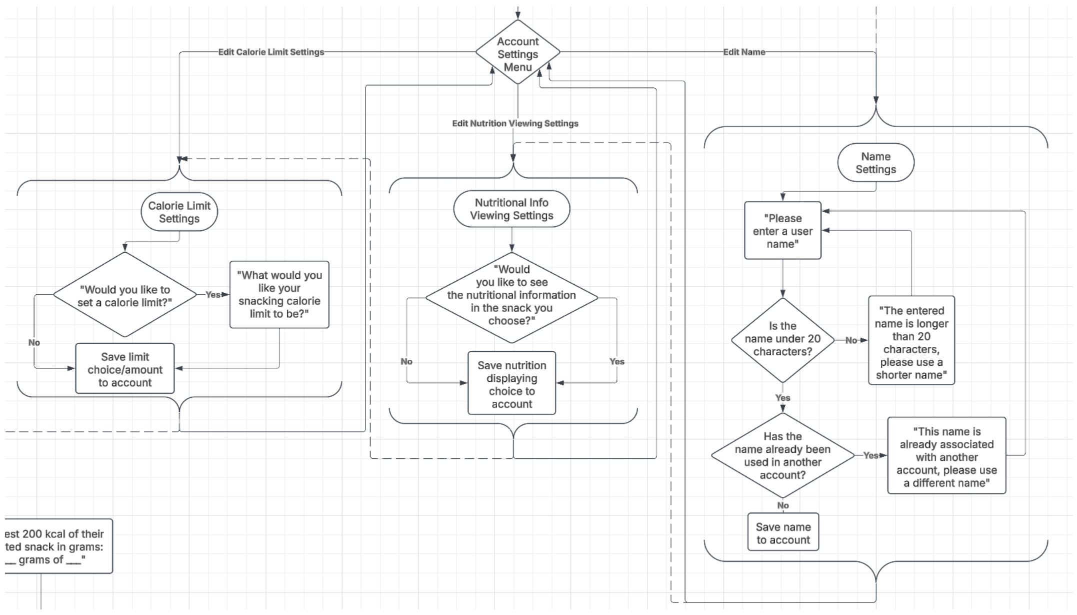
2.3.5 Software Subsystem

The software subsystem contains all our internal features. This includes a calorie limit, which prevents the user from dispensing any more after that limit is hit. If the user is close to their limit and tries to dispense a portion that will go over their limit, the machine will ask the user to select a smaller portion. We will also feature nutrition tracking. The machine will display the nutrition in each dispensed portion and keep track of the total daily nutrition. The user can choose whether they want each of these features activated or not. The user can also choose between two dispensing modes. One will allow the user to choose their desired snacks and portions. The other is a recommendation mode. This will allow the user to choose a snack, and if they have a calorie limit, the machine will divide their limit by three and suggest that amount in grams. If they don't, the machine will suggest 200kcal of their snack in grams. This feature is put in place to help with moderation. All these features will be coded into our microcontroller. The nutrition information for each snack will be held in an external database, which the microcontroller will connect to over Wi-Fi.

Requirements	Verification
• The machine should only display the features that the user has chosen.	• Run through the dispensing process and ensure that the correct features are enabled and implemented.
• The machine must keep track of and display the correct nutrition values.	• Compare the displayed nutrition per portion with our own values calculated by hand.

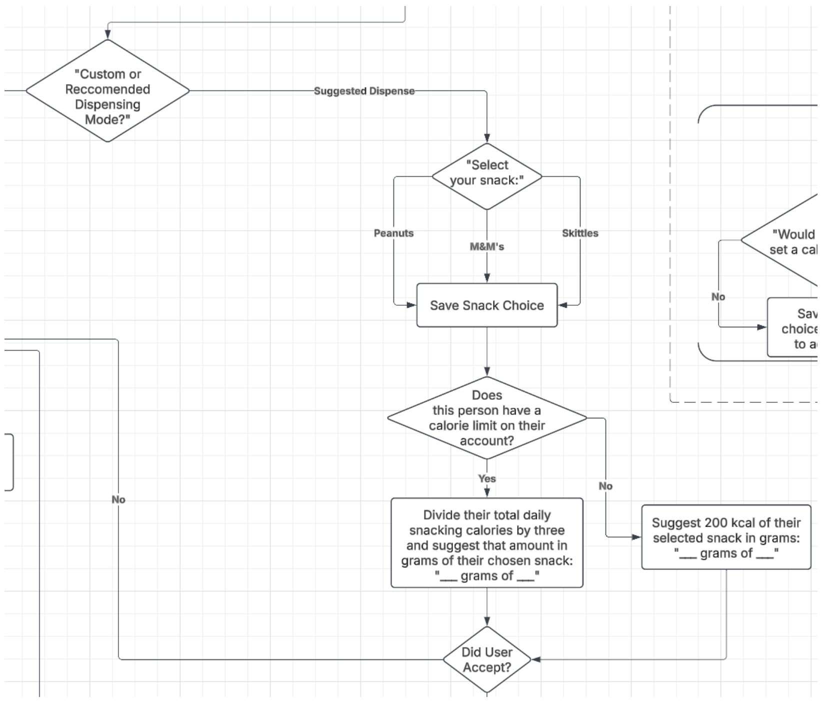
	• Compare the daily total with our own values calculated by hand.

Table 5: Software Subsystem R&V



Account Settings Menu
Edit Calorie Limit Settings
Edit Name
Edit Nutrition Viewing Settings
Calorie Limit Settings
"Would you like to set a calorie limit?"
Yes
"What would you like your snacking calorie limit to be?"
No
Save limit choice/amount to account
Nutritional Info Viewing Settings
"Would you like to see the nutritional information in the snack you choose?"
No
Yes
Save nutrition displaying choice to account
Name Settings
"Please enter a user name"
Is the name under 20 characters?
No
"The entered name is longer than 20 characters, please use a shorter name"
Yes
Has the name already been used in another account?
Yes
"This name is already associated with another account, please use a different name"
No
Save name to account
est 200 kcal of their
ted snack in grams:
__ grams of __"

Figure 9: Software Flowchart for Editing Account



"Custom or Reccomended Dispensing Mode?"
Suggested Dispense
"Select your snack:"
Peanuts
M&M's
Skittles
Save Snack Choice
Does this person have a calorie limit on their account?
Yes
No
Divide their total daily snacking calories by three and suggest that amount in grams of their chosen snack: "___ grams of ___"
Suggest 200 kcal of their selected snack in grams: "___ grams of ___"
No
Did User Accept?
"Would
set a cal
No
Sav
choice
to a

Figure 10: Software Flowchart for Suggested Dispensing Mode

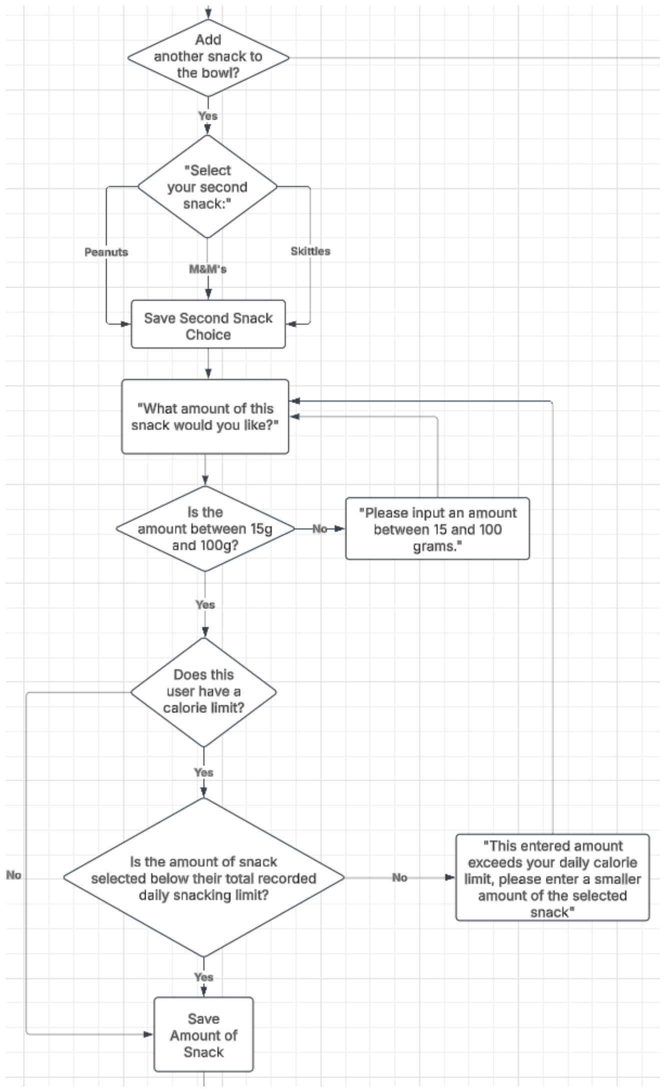

Add another snack to the bowl?
Yes
"Select your second snack:"
Peanuts
M&M's
Skittles
Save Second Snack Choice
"What amount of this snack would you like?"
Is the amount between 15g and 100g?
No
"Please input an amount between 15 and 100 grams."
Yes
Does this user have a calorie limit?
No
Yes
Is the amount of snack selected below their total recorded daily snacking limit?
No
"This entered amount exceeds your daily calorie limit, please enter a smaller amount of the selected snack"
Yes
Save Amount of Snack

Figure 11: Software Flowchart for Adding Another Snack to Be Dispensed

2.3.6 Power Subsystem

The power subsystem is required to power all our electrical components, such as the stepper motors, microcontroller, and sensors. This will require an AC/DC adapter that will convert the 120VAC from the wall outlet to 12VDC. We are choosing an adapter that is rated at 30W. We will have several voltage regulators to adjust the voltage to the needed level for each component. There will be a 12V to 5V voltage regulator and a 5V to 3.3V voltage regulator. The 12V to 5V regulator will utilize the LMR51430, which is a synchronous buck converter that is capable of outputting three amps of current. The 5V to 3.3V regulator will use a TSP62933O, which is also a synchronous buck converter capable of outputting three amps of current. Both voltage regulator schematics can be seen in Figures 12 and 13.

Requirements	Verification
• The wall adapter must be able to provide 12V±5% and a maximum output current of 3A.	• Check the output voltage with a multimeter and observe the output with the oscilloscope to measure the percent error. • Connect to the stepper motors to ensure they are provided the necessary voltage to operate. • Connect to the 12V to 5V regulator and measure the output power to ensure it is receiving the necessary amount of power to output 15W.
• The 12V to 5V voltage regulator must be able to provide 5V±3% and a maximum output current of 3A.	• Check output voltage and current with a multimeter before and after connecting the voltage regulator to the rest of the circuit. • Check the output voltage with the oscilloscope to measure the percent error.
• The 5V to 3.3V voltage regulator must be able to provide 3.3V±5% and a maximum output current of 3A.	• Connect the 5V to 3.3V regulator to the 12V to 5V regulator once it has been verified and check the output voltage and current with a multimeter. • Check the output voltage with the oscilloscope to measure the percent error. • Connect to the rest of the circuit and operate the machine to ensure all the other components are receiving their necessary power.

Table 6: Power Subsystem R&V

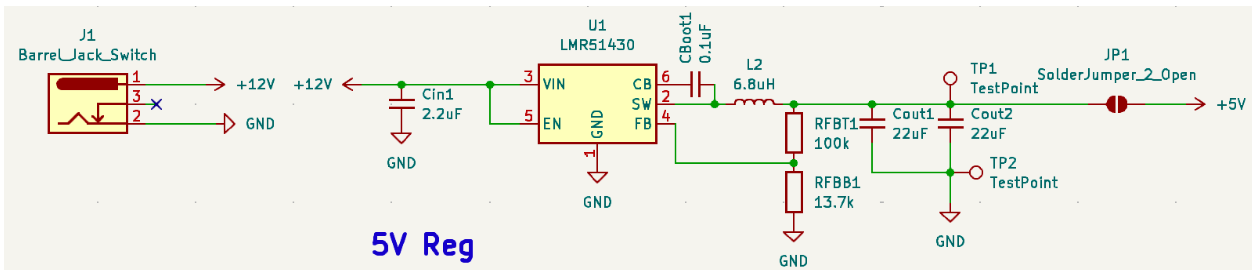


Figure 12: 12V to 5V Voltage Regulator Schematic

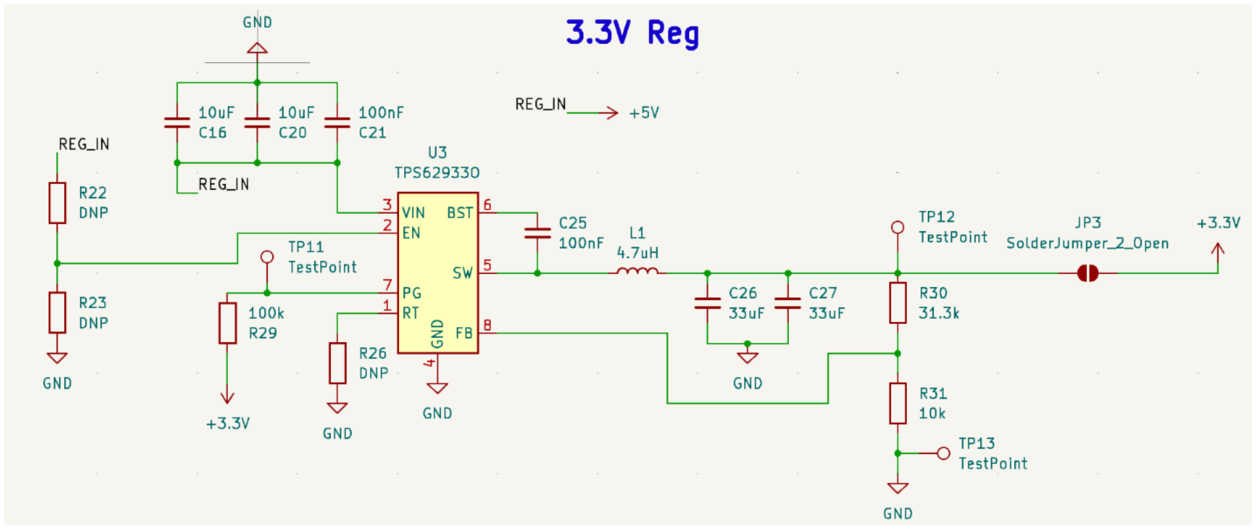


Figure 13: 5V to 3.3V Voltage Regulator Schematic

2.4 Tolerance Analysis

2.4.1 Weighing Accuracy

For our Smart Snack Dispenser, we primarily care about the error involved in dispensing an accurate weight of a snack to the user. This can be broken down into two primary concerns: the tolerance of the sensors themselves, and the tolerance of our system's ability to provide as close to the ideal number of pieces of snack in our dispensing wheel each time. The load cell that we've chosen has a 0.05% error, so for 500g that is a 0.25g error.

We are defining our worst-case scenario to be at the smallest amount we can dispense (15g) and to include the maximum error from the load cell. This is because errors are amplified at smaller amounts. For M&M's and Skittles, we would like the machine to ideally dispense two pieces at a time. For peanuts, we would like to dispense three pieces at a time. We aren't too worried about the machine dispensing less than the ideal number of pieces at a time because it can always turn more to dispense more. However, we are concerned about the machine dispensing more than the ideal number of pieces at a time. We will assume that the dispenser will be designed well enough so that only a maximum of one extra piece could fit. This is because multiple iterations will be tested to ensure we get the accuracy we desire.

For each snack, we will be testing a scenario where we are close to the desired 15 grams, but far away enough to need to dispense one more time. We will then see how far off we are from 15 grams if the machine dispenses three pieces for M&Ms and Skittles, and four pieces for peanuts. For example, in the case of the M&Ms, we are calculating how far off we are from 15 grams if we have already dispensed 13.65 grams and then dispense three more M&Ms. We also include our maximum load cell error in this calculation.

$$1\ M\&M \approx 0.91\ grams \quad 1\ Skittle \approx 1.04\ grams \quad 1\ Peanut\ \approx 0.8\ grams$$

$$M\&M's\!: \left|\frac{(13.65g +\ 3(0.91g) + 0.25g) - 15g}{15g}\right| * 100 =\ 10.86\%$$

$$Skittles\!: \left|\frac{(13.52g + 3(1.04g) + 0.25g) - 15g}{15g}\right| * 100 = 12.6\%$$

$$Peanuts\!: \left|\frac{(13.6g + 4(0.8g) + 0.25g) - 15g}{15g}\right| * 100 = 13.67\%$$

We can see that every snack fits into our desired tolerance of 15%. We believe this would only improve for larger amounts. This tolerance also accounts for the largest load cell error, which we believe would be unlikely for such a small amount.

2.4.2 Power Subsystem

We also would like to make sure that we have enough power being delivered to the entire system and that there isn't too much heat dissipation. We can first consider how many amps of current need to be provided. The estimated maximum current draw for each component can be seen in the table below.

Device	Maximum Current Draw
RFID (RC522)	30mA
Ultrasonic Sensor (HC-SR04)	20mA
Emitter (x3)	60mA
Receiver (x3)	30mA
Motor Driver (TB6612)	~10mA
Stepper Motor (XY42STH32-0354A)	0.7A
DC Motor (ROB-11696)	0.8A
Touchscreen Display (MSP0422)	~300mA
HX711 Load Cell Amplifier with 500g load cell	~10mA

ESP32-S3-WROOM-1 (Wi-Fi + all GPIOs)	~1A
	Total: ~2.96A

Table 7: Estimated Maximum Current Draw

This confirms that our 12V to 5V voltage regulator can provide the necessary current for the entire system. This is also an overshoot estimation because it isn't likely that all this current will be pulled continuously. We also know that our 5V to 3.3V voltage regulator can provide enough current for the ESP-32 because it can provide much more than 1A of current.

To analyze the heat dissipation of the 12V to 5V voltage regulator we can compare our output power to our input power. The maximum output power is 15W (5V*3A) and according to the datasheet, the regulator has around a 90% efficiency. This means that the input power should be about 16.67W. This gives us a heat dissipation of 1.67W and the LMR51430 can operate up to 150°C, so our regulator should operate well within the range.

We can analyze the heat dissipation of the AC/DC wall adapter using the same method. If the adapter needs to provide 16.67W and has around an 85% efficiency, the input power should be about 19.61W. This gives us a heat dissipation of 2.94W and the wall adapter can operate up to 40°C, so our wall adapter should be able to handle this dissipation well. We overshot our wall adapter power rating (36W) in case our efficiencies are lower than expected.

Finally, we can analyze the heat dissipation of the 5V to 3.3V voltage regulator. The necessary output power is around 3.3W. The datasheet says that the regulator has around a 85% efficiency, so the input power needs to be around 3.88W. This is only a heat dissipation of 0.58W and the TSP62933O can operate up to 150°C, so we should be well within operating range.

3. Cost & Schedule

3.1 Cost

Bill of materials:

Components	Description	Quantity	Total Cost [$]

ESP32 S3-WROOM-1	Microcontroller	1	6.13
HC-SR04	Ultrasonic sensor	1	4.50
RC522	RFID Tags & Chip	1	8.99
MSP4022	Touchscreen LCD	1	18.99
SparkFun HX711	Load Cell Amplifier	1	9.86
SEN-14728	500g Load Cell	1	12.50
Motor Driver	Motor Driver	1	6.26
XY42STH32-0354A	Stepper Motors	3	44.85
ROB-11696	DC Motors	3	7.50
L6R36-120	AC/DC Wall Adapter	1	12.19
LMR51430	12V to 5V Synchronous Buck Converter	2	2.58
TSP62933O	5V to 3.3V Synchronous Buck Converter	2	2.08
277-1276-ND	Terminal Block Connector	1	2.97
F1C1-201210-6R8M	6.8uH Inductor	2	0.98
SS8050-G	NPN Transistor	4	0.96
HX711 IC CHIP	Load Cell Amplifier IC	5	2.24
HPC 6028NF-4R7M	4.7uH Inductor	2	0.32
MMBT4403LT1G	PNP Transistor	2	0.26
Adafruit IR Break Beam Sensors	IR Sensor	3	2.95

Bill of labor:

Task	**Estimated Hours**	**Rate ($/hour)**	**Total Cost ($)**
PCB Designing	25 hours	$50	$1,250
Software/Coding	45 hours	$60	$2,700
Breadboard/Frame Construction	18 hours	$40	$720
Physical Debugging	28 hours	$50	$1,400
Software Debugging	35 hours	$55	$1,925
PCB Debugging	18 hours	$50	$900
Testing & Validation	25 hours	$55	$1,375

Documentation	18 hours	$40	$720
Machine Shop Services	9 hours	$70	$630

Total Price: $147.11 [Materials] + $10,620 [Labor] = $10,767.11

3.2 Schedule

Current schedule: Eric Nieto Gonzalez [E.N.G], Elinor Simmons [E.S.], Adam Kramer [A.K.]

Timeline	Task	Description	Team Member
Week 5	Integrate Prototype	Finish wiring the breadboard prototype to begin coding	E.N.G
Week 5	Submit 1st PCBs	Ensure we have PCBs to submit to test and modify when they arrive	E.S. & A.K.
Week 6	Order Parts	Order any components that we did not think of at the beginning	Everyone
Week 6	Test PCBs	Begin debugging and testing arrived PCBs to make modifications	E.S. & A.K.
Week 7	Prepare for Demo	Set up breadboard + prototype to present to our peers	E.N.G
Week 7	Submit 2nd PCBs	Ensure we have PCBs to submit to test when they arrive	E.S. & A.K.
Week 8	Refine GUI Software	Continue working on the graphical user interface code	E.N.G.
Week 8	Test PCBs + Prototype	Begin debugging and testing arrived PCBs to make modifications	Everyone
Week 9	PCB Improvements	Begin testing PCBs to see what improvements they need	E.S. & A.K.
Week 9	Refine Prototype	Everyone comes together to pitch in ideas on how to refine the product	Everyone
Week 10	Refine Prototype	Work on making it more functional and ensure all the code is well	E.N.G
Week 10	Submit 3rd PCBs	Ensure we have PCBs to submit to finalize when they arrive	E.S. & A.K.
Week 11	Order Broken Parts	Last opportunity to order any necessary parts to replace	Everyone
Week 11	Finalize GUI Software	Finish writing the code for the graphical user interface	E.N.G.

Week 12	Finalize PCBs + Prototype	Combine the prototype with the PCBs and test/debug	Everyone
Week 13	Mock Demo	Prepare for mock demo and ensure we are up to date with our work	Everyone
Week 14	Finalize Presentation	Prepare our overall presentation and ensure everyone is ready	Everyone
Week 15	Finalize Papers	Finish writing our finalized paper with all the details on our project	Everyone

4. Ethics & Safety

4.1 Ethics

The smart snack dispenser raises ethical concerns related to user privacy, data security, and autonomy. Since the system tracks individual snacking habits and nutritional data through RFID, it is essential to comply with ethical standards outlined by organizations such as the *IEEE Code of* Ethics and the *ACM Code of Ethics and Professional Conduct. IEEE's Principle 1* emphasizes the need to prioritize the well-being and privacy of individuals, ensuring that users' personal information remains secure and is not shared without explicit consent. Strong encryption, secure authentication, and strict access control measures must be implemented to prevent unauthorized access to user data.

Additionally, the dispenser's lockout system, which restricts snack intake based on preset calorie limits, introduces ethical concerns regarding user autonomy. In alignment with *ACM's General Ethical Principles*, particularly *Principle 1.4 (Respect Privacy)* and *Principle 1.6 (Respect User Autonomy)*, the system should promote healthier habits without imposing excessive restrictions. Users must retain the ability to override or customize their limits, ensuring that the feature is supportive rather than coercive.

Furthermore, *ACM's Principle 2.9 (Ensure Fairness and Non-Discrimination)* emphasizes the responsibility of technology designers to create systems that are fair and accessible to all users. The smart snack dispenser should be designed to prevent unintentional bias, ensuring that all individuals, regardless of their dietary restrictions, cultural preferences, or physical abilities, can use the system effectively. This includes offering a variety of snack options that cater to different nutritional needs, designing an intuitive interface that accommodates diverse users, and ensuring that the system does not unintentionally disadvantage any particular group. By integrating fairness into the design, the dispenser can serve as an inclusive and equitable tool for promoting healthier snacking habits.

4.2 Safety

Safety is a critical consideration in the design of the smart snack dispenser, particularly regarding electrical, mechanical, and food hygiene risks. The *IEEE Code of Ethics Principle 1* states that engineers must “hold paramount the safety, health, and welfare of the public,” which directly applies to preventing hazards associated with electrical and mechanical components.

- Electrical Safety: Since the dispenser plugs into a wall outlet, it must comply with industry standards for insulation, grounding, and circuit protection to prevent electrical shocks, short circuits, and overheating. Overcurrent protection and temperature monitoring should be included to mitigate fire hazards.
- Mechanical Safety: The motorized dispensing system should be designed to minimize risks of mechanical failure that could lead to snack jams or unintended dispensing. This is particularly important in preventing choking hazards, especially for small children. Safety mechanisms such as emergency stop features, or child-lock modes can help mitigate these risks.
- Food Safety: The design must ensure that snacks remain uncontaminated. In accordance with *ACM’s Principle 1.2 (Avoid Harm),* the dispenser should minimize food exposure to external contaminants. The use of food-grade materials, airtight containers, and easily cleanable surfaces will help maintain hygiene. Users should also receive clear maintenance and cleaning guidelines to prevent bacterial buildup or cross-contamination.
    - Users should be well-informed about the types of snacks used in the dispenser to avoid complications with allergies.

By adhering to IEEE and ACM ethical and safety standards, the smart snack dispenser can be designed as a secure, user-friendly, and responsible solution that prioritizes privacy, accessibility, and well-being.

5. References

[1] *Arduino.cc*, 2025. https://docs.arduino.cc/retired/getting-started-guides/TFT/, Accessed Feb. 13, 2025.

[2] Dejan, "Arduino Touch Screen Tutorial | TFT LCD," *How To Mechatronics*, Dec. 07, 2015. https://howtomechatronics.com/tutorials/arduino/arduino-tft-lcd-touch-screen-tutorial/, Accessed Feb. 11, 2025.

[3] Dejan, "DIY Vending Machine - Arduino based Mechatronics Project," *HowToMechatronics*, Nov. 13, 2017. https://howtomechatronics.com/projects/diy-vending-machine-arduino-based-mechatronics-project/, Accessed Feb. 13, 2025.

[4] Espressif, "ESP32 Series Datasheet Including," 2024. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf, Accessed Feb. 13, 2025.

[5] "IEEE Code of Ethics," IEEE, Sept. 19, 2024, https://www.ieee.org/about/corporate/governance/p7-8.html, Accessed Feb. 13 2024.

[6] Instructables, "Build Custom ESP32 Boards From Scratch! | the Complete Guide to Designing Your Own ESP32-S3 and C3 | Full Tut…," *Instructables*, Feb. 24, 2024. https://www.instructables.com/Build-Custom-ESP32-Boards-From-Scratch-the-Complet/, Accessed Feb. 11, 2025.

[7] Mayo Clinic Staff, "Counting calories: Get back to weight-loss basics," *Mayo Clinic*, 2018. https://www.mayoclinic.org/healthy-lifestyle/weight-loss/in-depth/calories/art-20048065, Accessed Feb. 13, 2025.

[8] "Smart Snacks Product Calculator," *Healthiergeneration.org*, 2016, https://foodplanner.healthiergeneration.org/calculator/, Accessed Feb. 13, 2025.

[9] B. Wansink and J. Sobal, "Mindless Eating," *Environment and Behavior*, vol. 39, no. 1, pp. 106–123, Jan. 2007, doi: https://doi.org/10.1177/0013916506295573.

[10] "Mindless to mindful eating," *Eufic.org*, 2024. https://www.eufic.org/en/healthy-living/article/mindless-to-mindful-eating?utm_source=chatgpt.com (accessed Mar. 05, 2025).