ECE 445
SENIOR DESIGN LABORATORY
DESIGN DOCUMENT

Secure Food Delivery Dropbox

Team No. 64
Rohan Samudrala
(rohans11@illinois.edu)
Dhruva Dammanna
(dhruvad2@illinois.edu)
Taniah Napier
(tnapier2@illinois.edu)

TA: Chi Zhang
Professor: Michael Oelze

March 2, 2025

Table of Contents

1. Introduction

1.1 Problem

In 2024, there were approximately 311.1 million users of the online food and grocery delivery market in the United States alone, and this number is projected to continuously rise.[1] One prevailing problem in the food delivery industry is theft. Some services try to combat this problem by requiring delivery drivers to take pictures of the food once it's dropped and sending it to the user to confirm delivery. However, there is no safeguard preventing passersby or even the delivery drivers themselves from taking the food after taking a picture. Another annoyance with this process is that users might order ahead to have food waiting for them upon arrival, although the convenience of having food waiting for you is offset by the fact that the food will get cold.

1.2 Solution

To address this issue, we propose a secure food drop system/box that remains locked and can only be opened by the person who placed the order. The overall idea of the box is to have a box that is perpetually locked but allows the user to give access to open the box to the delivery driver through entering a keycode. Once the driver opens the box and puts the food inside the box, the box will be locked again. Then, the user will be the only person able to open the box through fingerprint. Additionally, we propose integrating a heating system inside the box that will maintain the food's warmth and monitor the temperature inside the box.

To start the process, weight must be temporarily added inside the box so that a randomly generated code can be created and sent to the user's email. The user must then text this code to their driver so that they can open it and place food inside. Once the food is placed inside, a weight sensor will trigger a solenoid lock to move a deadbolt and automatically lock the box as well as generate a new code, locking the driver out. The microcontroller will use its wifi capability to send the new code to the user through a program to be ready for the next order. The box will also keep the food at a high enough temperature to continually stay warm. When they arrive, the user will use a calibrated fingerprint on the fingerprint sensor so they can open it at any time securely.

1.3 Visual Aid

Figure 1: Visual Aid

1.4 High level requirements

- Authentication: Authentication should initiate unlocking the box with correct inputs and keep the box locked with incorrect inputs. Specifically, authentication using the keypad should behave as described at 100% of the time, and authentication using the fingerprint scanner should behave as described at least 80% of the time.
- Box Mechanism: The weight sensor in the box should detect the presence of an object of at least 50 grams and initiate locking the box within approximately 20 seconds at least 95% of the time.
- Code generation: A master keypad code should be generated 1 time when the user inputs a master code in the website and temporary keypad codes should be generated when food is placed on the weight sensor. The temporary code should be sent to the user within 1 minute ± 20 seconds of generation.

2. Design

2.1 Block Diagram

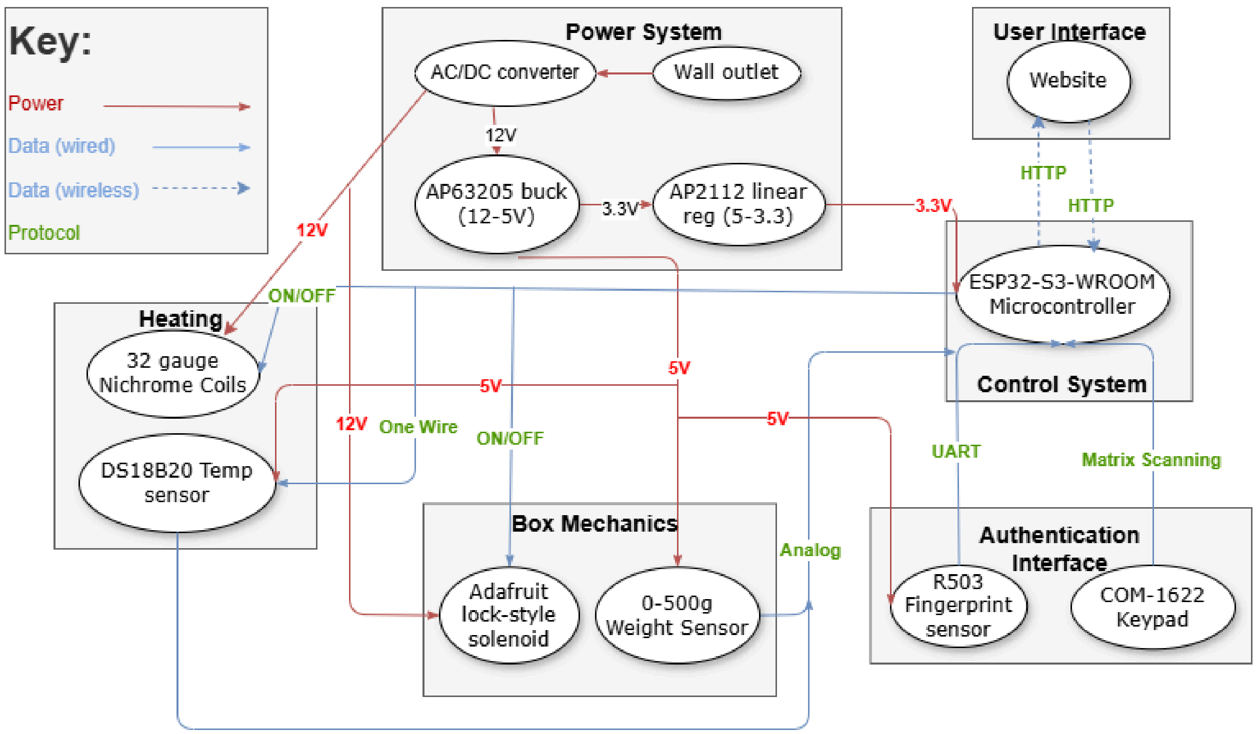


Figure 2: Block Diagram for Secure Delivery Dropbox

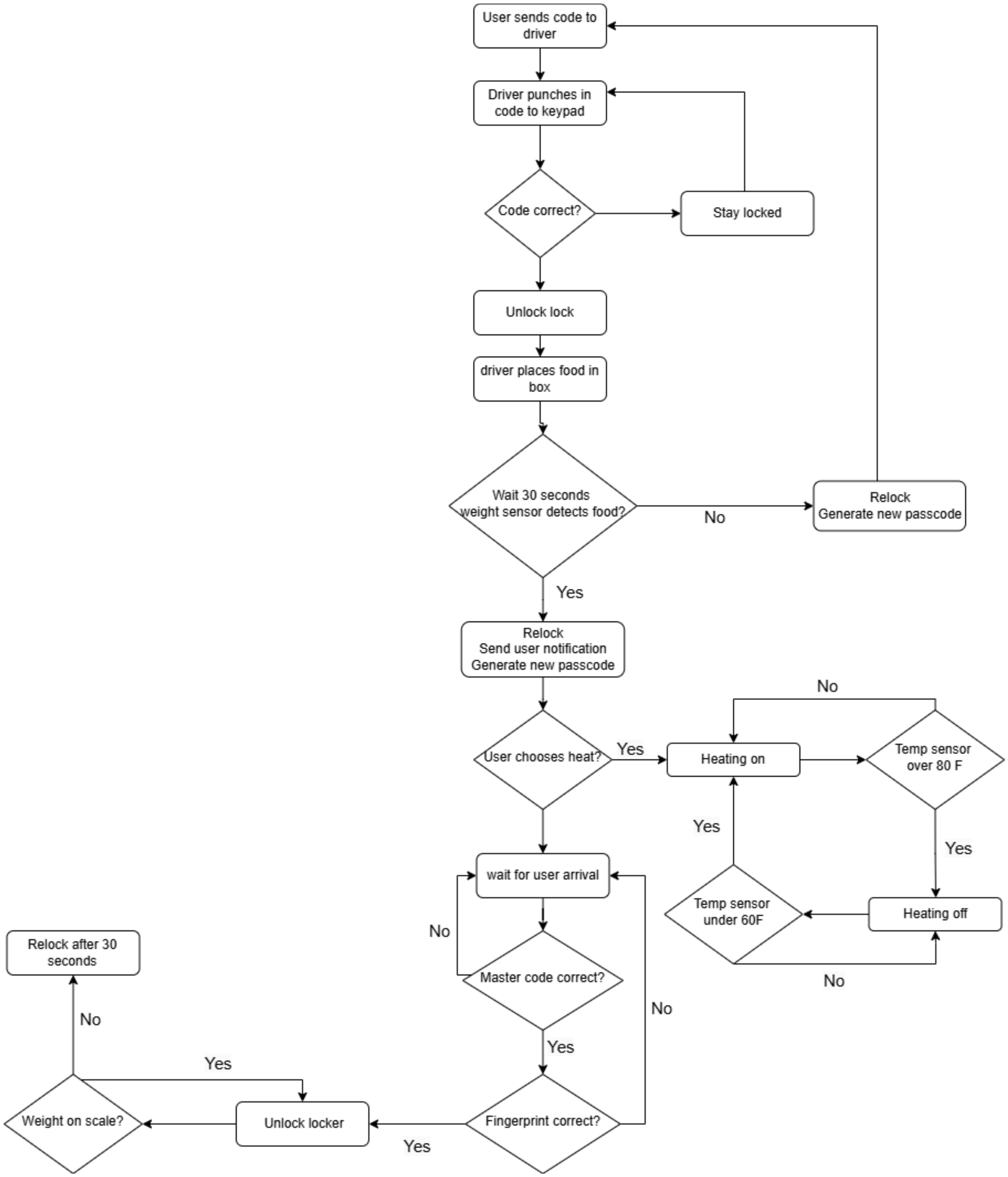


Figure 3: Top level flowchart of the overall box function

2.2 Physical Design

Our lockbox will be constructed out of a modified locker (15"D x 15"W x 24"H) held in the machine shop from previous projects. We will attach our keypad and fingerprint sensor on the outside through holes drilled in the locker. The data and power wires will travel on the outside into a box created on the outside to house the PCB, attached to the side of the locker (shown on the door in Figure 4.). This is because the heating elements inside the box may harm the electronics running it. The heating element will consist of nichrome coils at the top of the box, far away from the weight sensor load cell at the bottom. A temperature sensor in the box will send a temperature reading to our ESP.

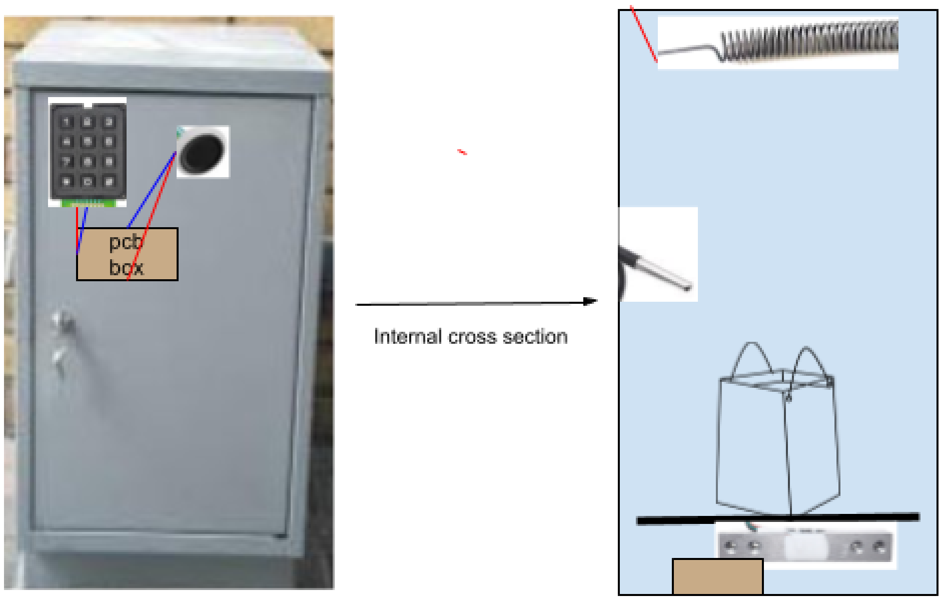


Figure 4: Diagram of box with desired placement of box components

2.3 Subsystem Overview

2.3.1. Power

The power subsystem provides power for different components of the box to function. This subsystem will use a power supply adapter to convert AC wall outlet power to 12 Volts 5 Amps of DC power. We decided to use outlet power since we know that the heating subsystem will require more power than a battery can provide. The approximate amount of current the whole board will draw is around 2.6A, this is well below what our AC to DC converter can supply. This DC power supply will then go through voltage regulators to achieve the desired output voltage to power the components in other subsystems. The control subsystem will receive 3.3 Volts, the heating subsystem will receive 12 Volts, the fingerprint scanner from the Authentication subsystem will receive 5 Volts, the lock and the weight sensor in the box mechanisms subsystem will receive 12 Volts and 5 Volts respectively. We step down the 12 Volts to supply 5 volts by using an AP63205WU voltage regulator coupled with a 4.7µH inductor and 4 capacitors (22µF, 22µF, 10µF, 100nF), as specified by the datasheet [2]. We also step down the 5 Volts from the AP63205WU to 3.3 Volts using a AP2112K-3.3 voltage regulator coupled with 2 capacitors (1µF, 1µF), as specified by the datasheet [3]. There is also another voltage regulator that is not in the power subsystem and is in the control subsystem. It is the AMS1117-3.3 and it is used when programming the ESP 32 via usb-c. It regulates the voltage a computer provides from 5 Volts to 3.3 Volts and uses 2 capacitors to work(22µF, 1µF). This was in accordance with the datasheet[4]. These voltage step downs are done in order to be able to use every component in our design with only 1 12 Volt 5 A supply.

Lastly, there is a screw terminal with space for 2 wires which will be used to connect the power and ground of the 12 Volt 5 A supply.

Requirements	Verification
1. The buck converter must take a 12V input and output a 5V output	1a. Use the screw-in terminals on the 12V AC/DC converter to attach the V+ and GND to the PCB. 1b. Measure the voltage using a multimeter at the V+ and GND of the AC/DC screw terminal and verify that it is 12 Volts. 1c. Measure the voltage across Capacitor 9 by putting the multimeter probes on both sides of the capacitor, the voltage should read as 5 Volts
2. The linear regulator must take a 5V input and output a 3.3V output when being powered by the barrel jack	2a. Use the screw-in terminals on the 12V AC/DC converter to attach the V+ and GND to the PCB.

connector	2b. Measure the voltage using a multimeter at the V+ and GND of the AC/DC screw terminal and verify that it is 12 Volts. 2c. Measure the voltage across Capacitor 7 by putting the multimeter probes on both sides of the capacitor, the voltage should read as 3.3 Volts
3. The linear regulator must take a 5V input and output a 3.3V output when being programmed via USB-C	3a. Connect the PCB to a computer via the USB-C port 3b. Measure the voltage across Capacitor 1_polar by putting the multimeter probes on both sides of the capacitor, the voltage should read as 3.3 Volts

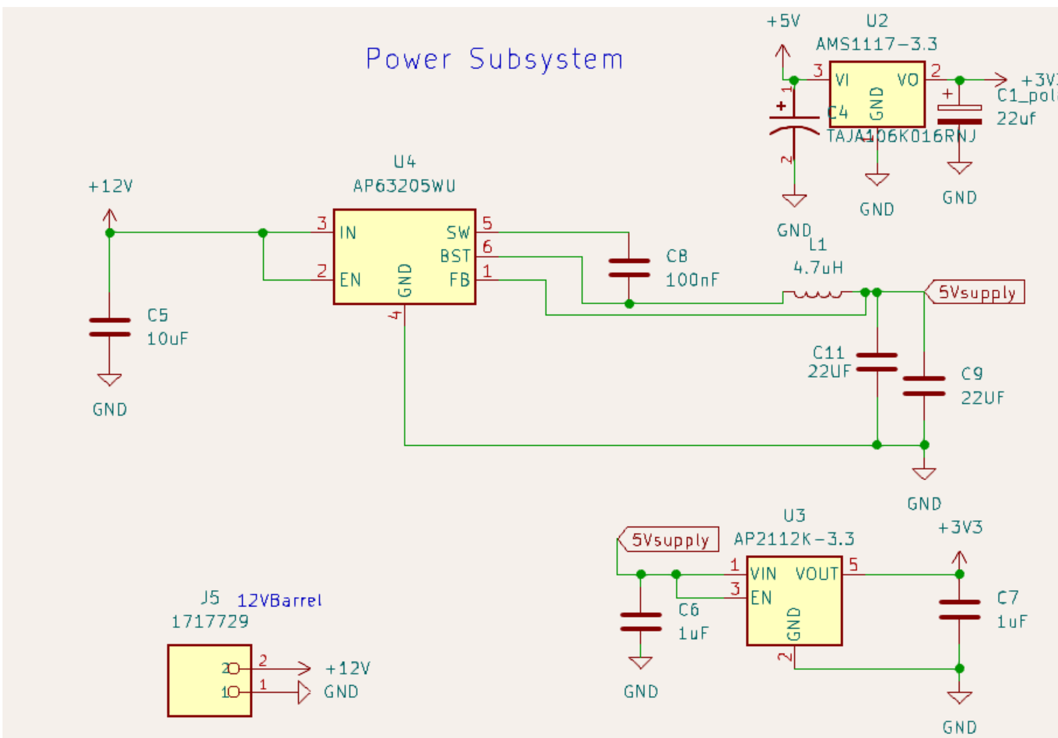


Figure 5: Power subsystem schematic

2.3.2 Control

The control subsystem will make sure that all communication between subsystems are carried out such that the box will be able to lock, unlock, manage heating, and send codes to the user. We chose to use the microcontroller ESP-32-S3-WROOM-1[5] because of its Wi-fi capabilities, which we will use for communicating with the user interface subsystem. The control subsystem will receive 3.3 volts from the power subsystem to operate and 3.3 volts from a usb-c and voltage regulator to program it.

The microcontroller will send the user, via the user interface subsystem, the temperature reading inside the box as a notification when the food is placed in the box, along with the new randomly generated keycode, while the user will send the microcontroller 2 inputs, the master keycode that the user will set as well a whether the heating should be turned on or off. We chose to allow the user to choose whether heating should be turned on or off because users might not want to heat up some foods, such as drinks. The microcontroller will receive data from both the fingerprint sensor and keypad, and use this data to determine whether to initiate unlocking the in box mechanics subsystem, as outlined in the high level requirements. The microcontroller will also receive a weight reading from the box mechanics subsystem, and use this information to determine if there is food present inside the box. The microcontroller will then take this input, the input of whether to enable heating, and a temperature reading inside the box from the heating subsystem, and send a signal of whether to turn off or on the Nichrome heating inside the box, as described in the high level requirements. If the user does not want heating to be turned on, no matter the other inputs, heating will not be initialized. If the user does want heating and there is food inside the box, the management of the heating will be as described in the high level requirements.

Requirements	Verification
1. EN button sends 3.3V to the EN pin in the ESP 32 when pressed	1a. Press and hold the Enable button (S1) on the PCB 1b. Put the positive end of the multimeter at the EN pin of the ESP 32 and the negative end of the voltmeter at the ground. It should read as 3.3V.
2. BOOT button sends 3.3V to IO0 pin in the ESP 32 when pressed	2a. Press and hold the Boot button (S2) on the PCB 1b. Put the positive end of the multimeter at the IO0pin of the ESP 32 and the negative end of the voltmeter at the ground. It should read as 3.3V.
3. USB-C is properly connected to PCB and can program ESP 32	3a. Plug in the USB-C into a computer 3b. Import a test program using the Arduino software (test programs are available for free example) 3c. While the program is being imported check the D+ and D- pins of the USB-C using oscilloscope probes to see if there is any waveform showing up. If there is it is working
4. New temporary code is generated	4a. Place food on the load cell

within 1 minute ± 20 seconds after food is placed on the weight scale	4b. New code should be sent within 1 minute 20 seconds on the website
5. ESP32 correctly recognizes all input signals from subsystems	5a. Connect each used input pin to a high voltage of 3.3V 5b. Check that ESP verifies input on COM6 output terminal

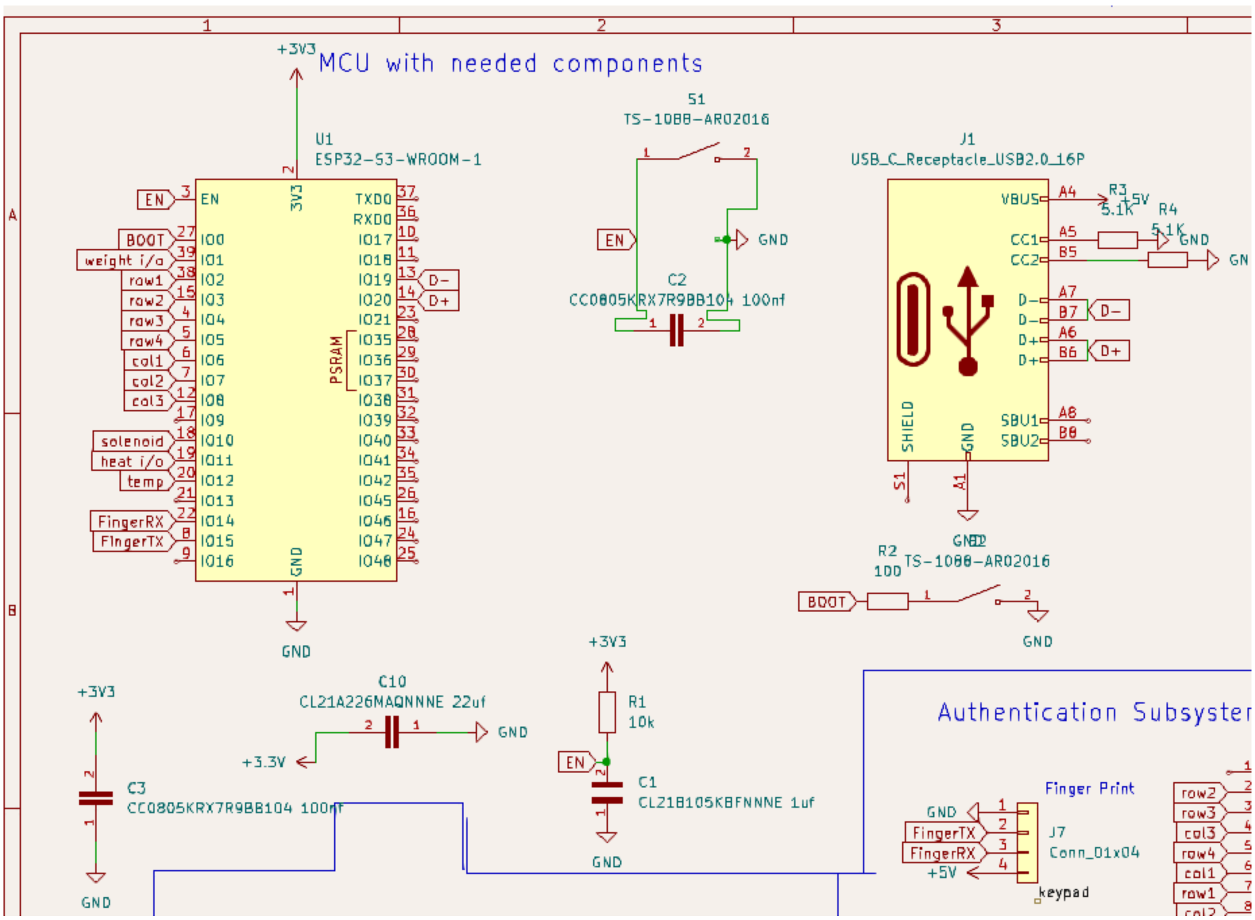


Figure 6: Control subsystem schematic

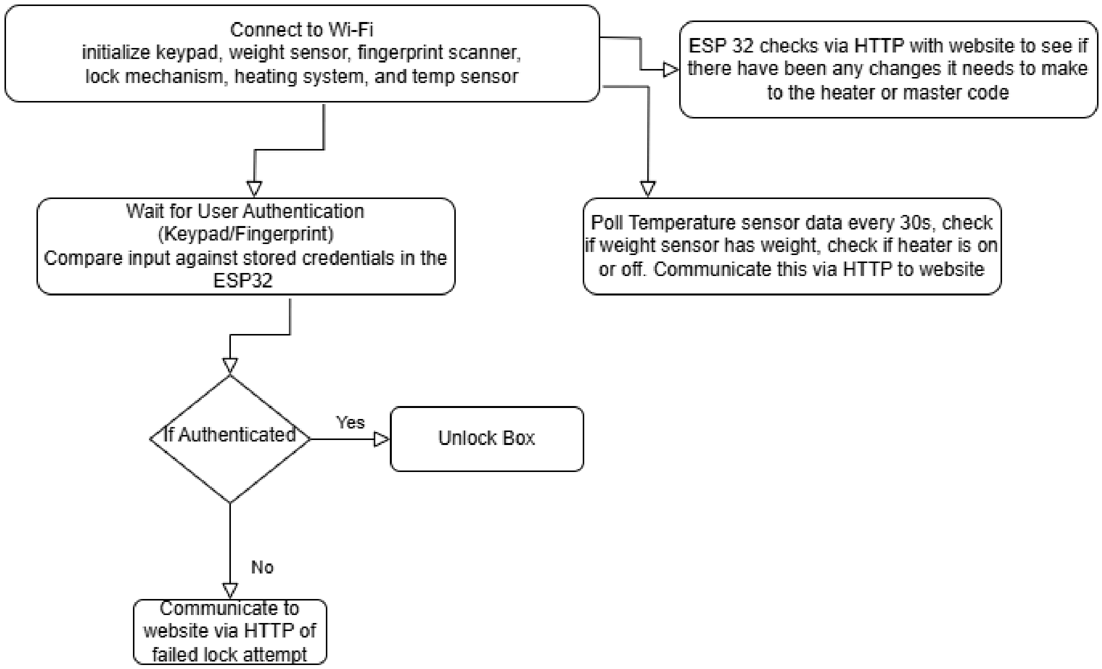


Figure 7: Flowchart of control subsystem responsible

2.3.3 User Interface

The User Interface will be a website that the user will use to interact with the PCB. It is essential for the secure box to be able to function seamlessly for the user. The user can set their master keycode for the keypad on the website and this code is sent to the microcontroller in the control subsystem to use in connection with the authentication subsystem. The user will also have a button on the website that will tell the microcontroller to turn on or off the heating inside the box. The microcontroller will tell the user via the website what the temperature reading inside the box is. The microcontroller will display on the website when food is inside the box. The microcontroller will also send a notification containing a new randomly generated code when food is detected inside the box through the website. All communication between the user interface and the microcontroller occurs via HTTP. The user interface will be programmed using C++ and will communicate with the ESP 32 using the HTTP protocol. Both the website and ESP 32 will perform POST and GET requests to communicate with each other.

Requirements	Verification
1. Heater turns on within 30 seconds of pressing the heater enable button on the website	1a. Press the heater enable button on the website 1b. Wait up to 30 seconds for the heater to turn on 1c. Check the website to see if the

	temperature inside the box is increasing
2. User receives notification of food arrival and new random keycode	2a. Send input on weight sensor by putting food on scale 2b. Check the website to see if a notification arrives and new random keycodes get generated within 1 minute 20 seconds.
3. User can create a new master key code	3a. Enter a new 4 digit master keycode in the master key code box on the website and press enter 3b. Wait 1 minute 20 seconds 3c. Enter the new master keycode in the keypad and enter your fingerprint to unlock the box

2.3.4 Authentication

The Authentication subsystem consists of a basic 3x4 matrix keypad that is wired as described in the datasheet[6] and a ZFM-20 Series fingerprint scanner[7]. The keypad will receive its 3.3 volts from the microcontroller and the fingerprint scanner will receive 5 volts from the power subsystem to operate. The keypad component will allow the user to enter a specific 4 digit keypad code and the driver to enter the randomly generated 4 digit keypad code that will initiate unlocking the box when correctly inputted, however when incorrect keycodes are imputed into the keypad, the box will remain locked, as specified by the high level requirements. The fingerprint scanner component will allow only the user to use their fingerprint to unlock the box, but the box will remain locked when other people's fingerprints are used, as specified by the high level requirements. We want a correct keycode to be able to unlock the box when food is not inside the box, but once food is inside the box both the correct master code and a fingerprint are required. This subsystem will give its data to the microcontroller to communicate with the lock appropriately.

Requirements	Verification
1. Keypad recognizes a correct code	1a. Program a master code onto the ESP connected to keypad using the website 1b. Enter the password and check that the box does not unlock
2. Keypad declines all incorrect codes	2a. Program a password onto the ESP connected to the keypad. 2b. Enter the password and check that the code doesn't recognize it and create a false positive.

3. Fingerprint sensor recognizes correct fingerprint with 80% accuracy	3a. Register master fingerprint by following documentation on ESP 3b. Check correct fingerprint 20 times and look for system verification at least 16/20 attempts
4. Fingerprint sensor recognizes incorrect fingerprint with 100% accuracy	4a. Register master fingerprint by following documentation on ESP 4b. Check incorrect fingerprint 20 times and look for system verification on all 20 attempts
5. Check if dual factor authentication unlocking works once there is food on the scale	5a. Put a weight on the load cell 5b. Let the box automatically lock in 30 seconds 5c. Unlock the box using fingerprint and keypad

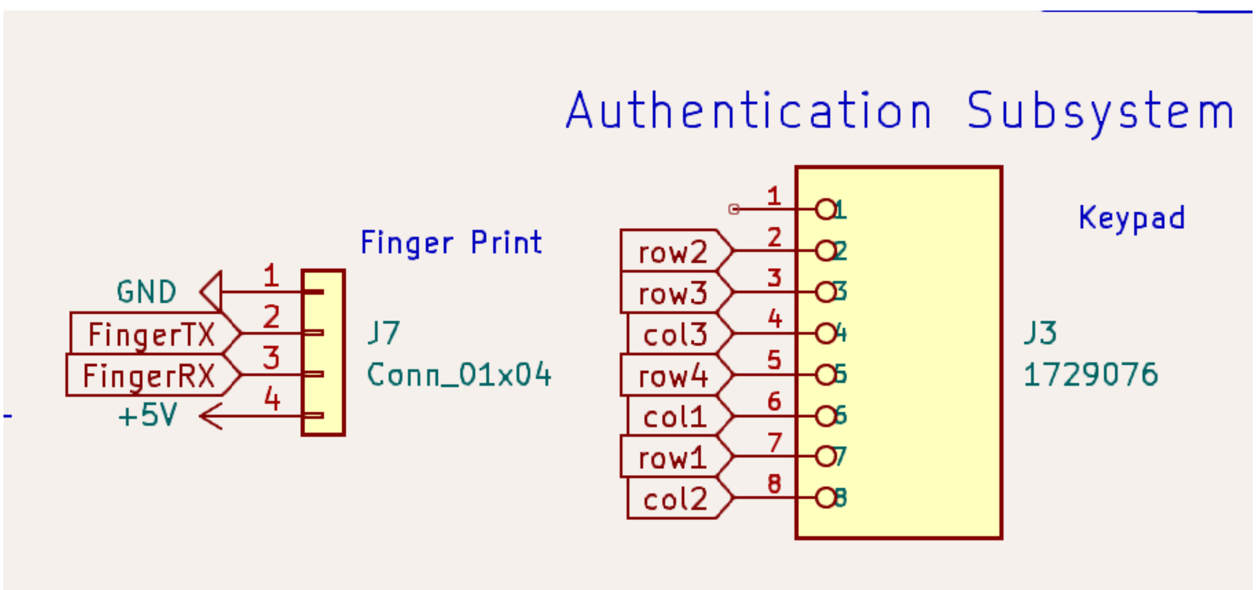


Figure 8: Authentication subsystem schematic

2.3.5 Box Mechanics

The box mechanics subsystem consists of the MP001161 electromagnetic solenoid lock [8] and the SKU 314990000 weight sensor[9], used to lock the box as specified in the high level requirements and detecting the presence of food as described by the high level requirements. The lock and weight will receive 12 volts and 5 volts, respectively, from the power subsystem.
We chose to power the lock specifically with 12 volts from the wall outlet because solenoid locks

act like electromagnets and a lot of current (about 500MmA) will rush into the lock to charge up the electromagnet, so we will need more than a 9 volt battery can provide.

The lock will receive a signal from the control system that will indicate if it should be in the locked or unlocked position. Based on the high level requirements the lock should be in the locked position by default but when food is not inside the box, a correct keycode entry or fingerprint entry should cause the lock to move to the unlocked position. When food is detected inside the box, both a correct master keycode entry and fingerprint entry are required for the lock to move to the unlocked position. When unlocked the lock will remain in the unlocked position for 30 seconds before returning to the locked position. We estimate this is enough time to place food inside the box or retrieve food from the box and close the door. The keycodes and fingerprint entries will be sent to the control system that will convert these entries into signals for the lock. Detecting whether there is food inside the box will be handled by the weight sensor. The weight sensor will send a weight reading signal to the control subsystem, which will use this input to carry out other functions of the box, as described in the control subsystem section. We decided that the presence of food will be indicated by an object of at least 50 grams being placed on the weight sensor. We chose a smaller weight sensor (0-500 grams) because most food orders won't exceed 500 grams, and we decided to make 50 grams our threshold for food since it's about the weight of a single aerated food item, such as a donut.

The weight sensor circuit will feature an INA125 amplifier [10] to amplify the voltage signals coming from the weight sensor. Since the weight sensor will produce small voltage signals that are proportional to the weight applied to it, since our sensor is for smaller weights, we need the amplifier or else the signal will not be strong enough for the microcontroller to read.

Requirements	Verification
1. Solenoid lock activates and deactivates when prompted by ESP	1a. Connect ESP I/O pin to solenoid lock data input 1b. Send a signal through the program checking both activation and deactivation prompts
2. Weight sensor receives and sends signal to ESP	2a. Connect weight sensor i/o pin to ESP 2b. Place weight of 250g on sensor (avg weight) 2c. Check that sensor output is correct on the website

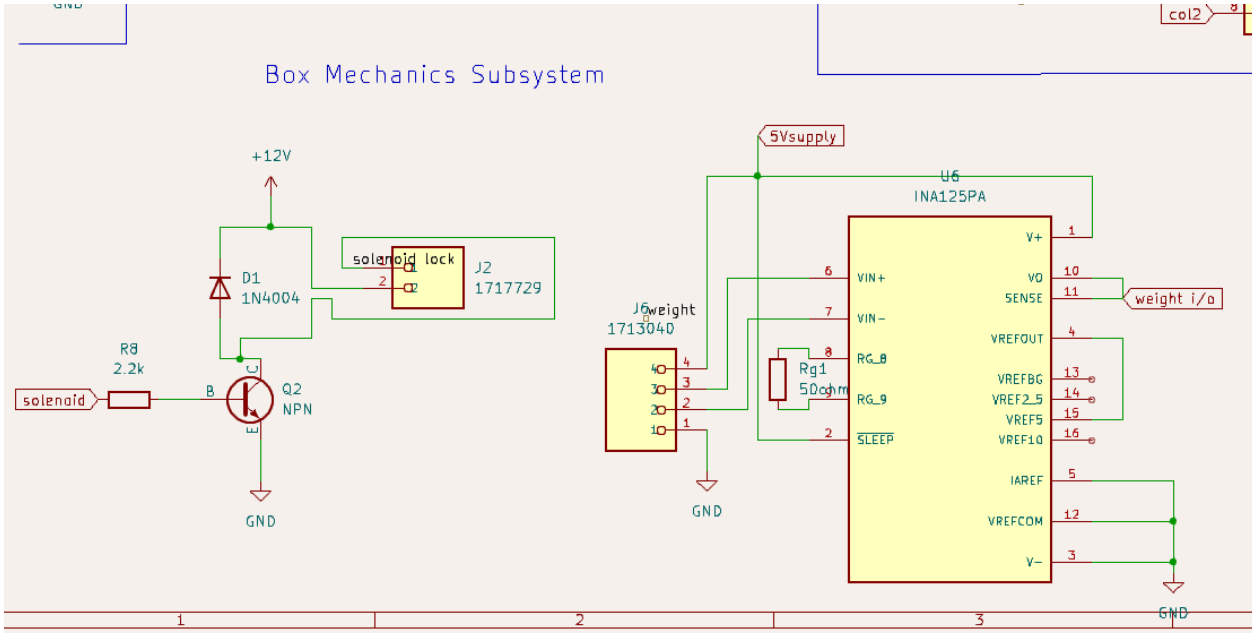


Figure 9: Boxes mechanics substem schematics

2.3.6 Heating

The heating system consists of a DS18B20 digital thermometer temperature sensor[11] and 32 AWG Nichrome wire. The heating will be initiated when the user decides to keep their food warm and food is correctly detected inside the box. The temperature sensor will receive 5 Volts from the power subsystem and the wire will receive 12 volts from the power subsystem. The temperature sensor will send a temperature reading to the control subsystem, the control subsystem will use this reading and signals from other subsystems to control whether the Nichrome wire is actively heating up. When the user specifies that they want heating on and food is inside the box, if the temperature sensor reading indicates that the environmental temperature is over 32.2 degrees Celsius (90 degrees Fahrenheit) the wire should stop heating up, but if the temperature reading falls below 15.5 degrees Celsius (60 degrees Fahrenheit) the wire should begin heating again. We will poll the temperature sensor every 30 seconds for its temperature reading. The length of wire we will use to reach the desired temperature range and the calculations done to determine it are described in the tolerance section.

Requirements	Verification
1. The heater will turn off within 30 seconds after the temperature inside the box is 78 degrees Fahrenheit or more	1a. Turn the heater on using the enable heater button on the website 1b. Wait for the temperature sensor reading on the website to be 80 degrees Fahrenheit or more. 1c. Once it is 80 degrees Fahrenheit or more,

	check the heater enable button to see if it is off and also check the heater inside the box to see if it is off.
2. The temperature displayed on the website is updated every 30 seconds	2a. Turn on the heater to change the temperature 2b. Wait for the temperature on the website to update 2c. As soon as it has updated start a timer for 30 seconds 2d. Once the 30 seconds are finished look to see if the temperature has updated inside the box

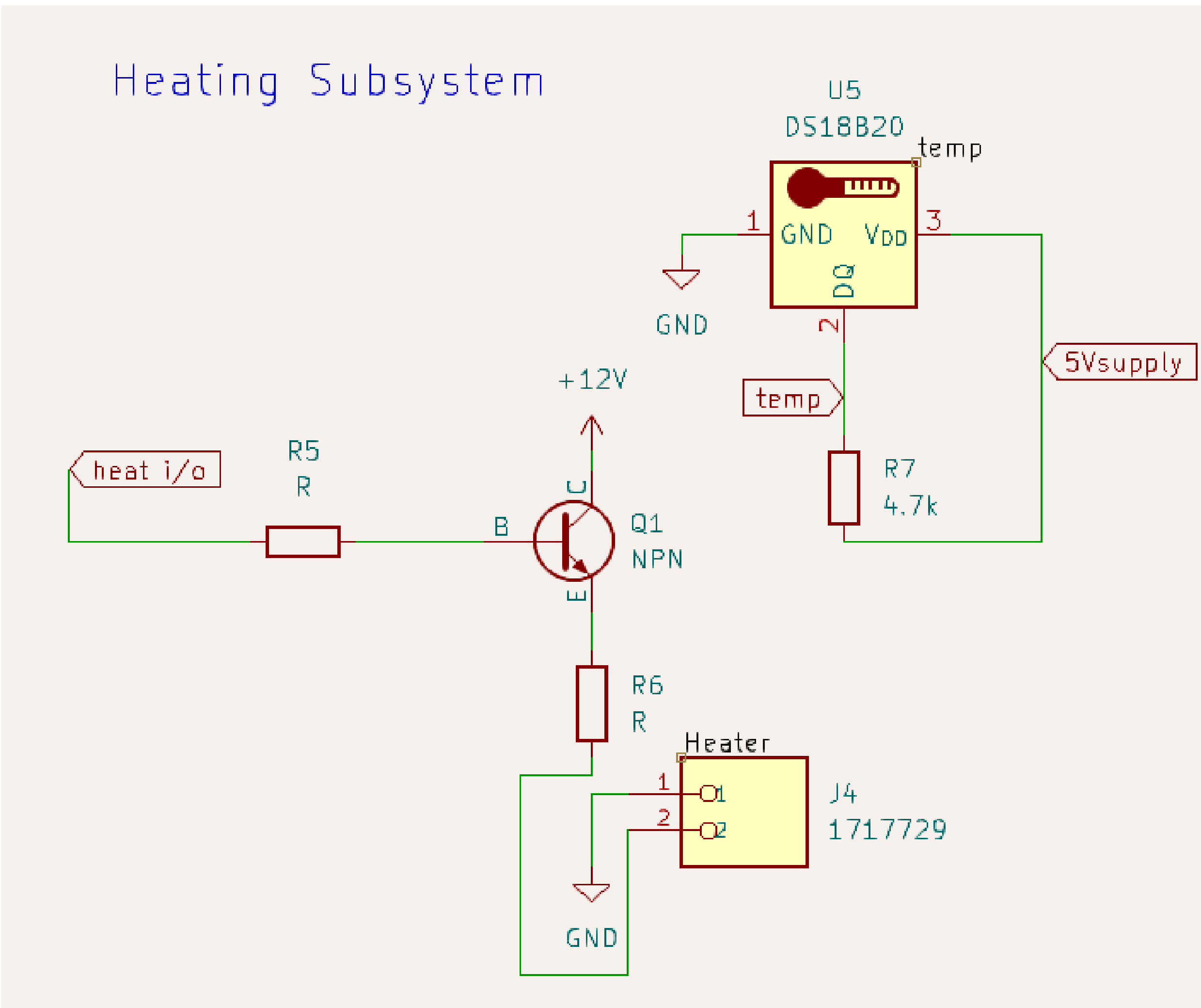
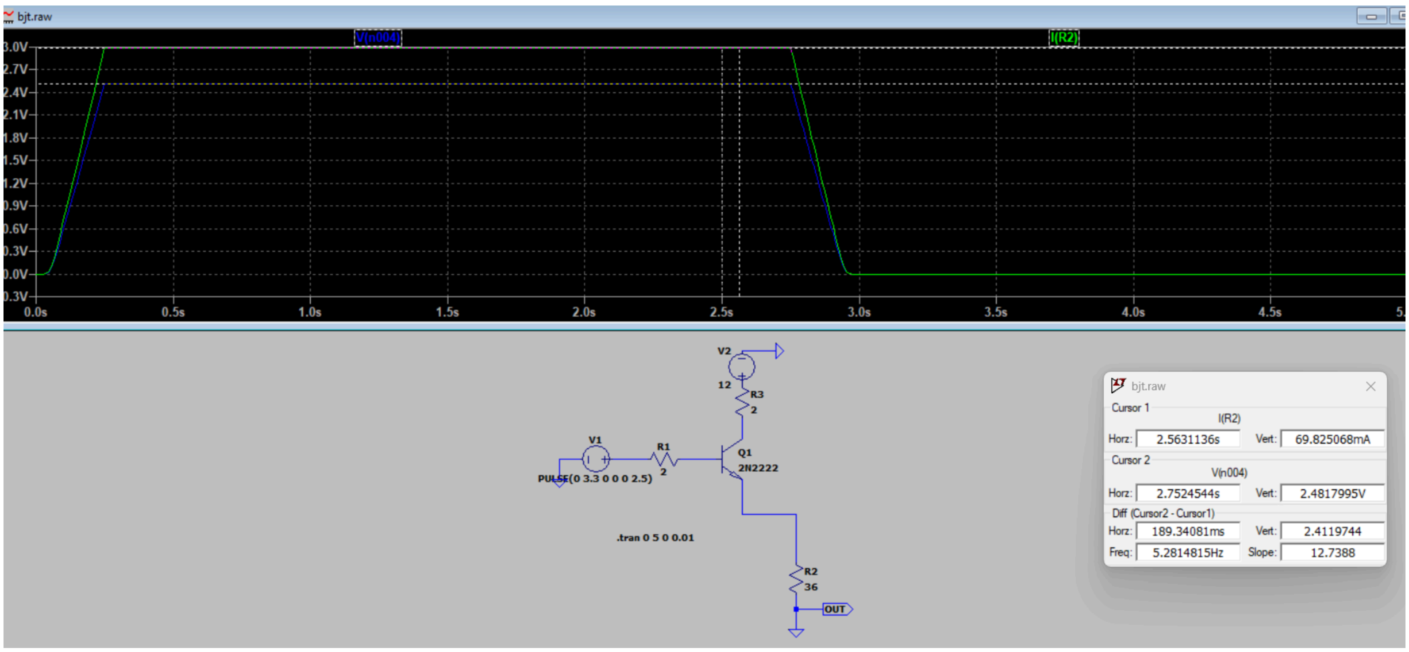


Figure 10: Heating subsystem schematic

2.4 Tolerance Analysis

One critical area of our project is the heating component. If the weight sensor is activated past the threshold, the temperature isn't too high, and our enable signal from the user is high, the ESP-32 needs to send a signal telling the heating to turn on. However, the data signal will be 3.3V like all signals from the ESP, nowhere near enough to power the heating element. To fix this we used a circuit with a BJT with the ESP signal at the base. This will only allow the 12V A current to travel from the source to drain through our heating coils when a 3.3V signal activates the BJT.



To control the heating element, we needed to find the right length that has the right resistance to reach 80 degrees F. Using an online calculator [12] we found that a resistance of 36 Ohms should fit our needs. We can then use an LT spice simulation to find the Voltage and current highs with our setup.

Figure 11: LT spice simulation used to determine the voltage and current highs for our heating component

We can then use this equation:

$$Vp = I \times (\text{Wire Length (inch)} / 12) \times (\text{Resistance (Ohms)}/12)$$

To calculate our wire length of about 11 inches (10.577) needed to reach the ideal temperature.

3. Cost and Schedule

3.1 Cost Analysis

3.3.1 Labor

According to [13], the average yearly salary of a graduate electrical engineer from UIUC is $87,769. From an average work week of 40 hours/week, this works out to almost $50 per hour. We expect to work an average of 15 hours per week for 10 weeks or a total of 150 hours per person on this project.

Labor cost per person = $50/hour * 2.5 * 150 hours = $18,750
Total Labor cost = $18,750 * 3 people = $56,250

3.3.2 Parts

Part	Manufacturer	Quantity	Cost/Unit	Link
ESP32-S3-WROOM-1-N4R2	Espressif Systems	1	$3.10	Link
UJC-H-G-SMT-P6-TR	Same Sky	1	$0.46	Link
AMS1117-3.3	Texas Instruments	1	$1.01	Link
AP63205WU-7	Diodes Incorporated	1	$1.38	Link
AP2112K-3.3TRG1	Diodes Incorporated	1	$0.52	Link
1717729	Phoenix Contact	3	$2.74	Link
1729076	Phoenix Contact	1	$3.91	Link
282837-4	TE Connectivity	1	$0.90	Link
ALITOVE DC 12V 5A Power Supply Adapter Converter Transformer AC 100-240V Input	ALITOVE	1	$11.99	Link
Tegg 1PC 3x4 Keypad MCU Board Matrix Array Switch Tactile Keypad 12 Button Phone-Style Matrix Keypad for Arduino Raspberry Pi	Tegg	1	$9.99	Link
Ultra-Slim Round Fingerprint Sensor and 6-pin Cable	Adafruit	1	$19.95	Link
TIP102	STMicroelectronics	2	$1.32	Link

314990000	Seeed Technology Co., Ltd	1	$17.00	Link
Waterproof 1-Wire DS18B20 Digital temperature sensor	Adafruit	1	$9.95	Link
uxcell 0.2mm 32AWG Heating Resistor Wire Wrapping Resistance Wires for Heating Elements 33ft	uxcell	1	$7.29	Link
ATOPLEE 2pcs Electromagnetic Solenoid Lock,DC 12V 0.8A Door Drawer Tongue Down Slim Design Assembly Magnetic Lock for Cabinet Door Drawer,55X42X27mm	ATOPLEE	1	$19.99	Link
INA125PA	Texas Instruments	1	$6.79	Link
GRM033C80J224KE90D	Murata Electronics	5	$0.08	Link
GRM21BR61C106KE15K	Murata Electronics	8	$0.13	Link
CL05B104KP5NNNC	Samsung Electro-Mechanics	3	$0.08	Link
ERJ-6GEY0R00V	Panasonic Electronic Components	3	$0.10	Link
RC0805FR-071KL	YAGEO	3	$0.10	Link
ERA-6AEB101V	Panasonic Electronic Components	7	$0.10	Link

Total Cost: $189.09

3.3.3 Grand Total

Grand total = *total labor cost* + *total parts cost*

$56,250 + $189.09 =

$56,439.09

3.2 Schedule

WEEK	TASKS	PERSON
3/3	Design Document	Everyone
	Continue buying parts	Rohan
	Start breadboard assembly	Rohan
3/10	Continue breadboard assembly	Everyone
	Assemble PCB and test	Taniah
	Work on and tests power subsystems	Work
	Breadboard demo - Wednesday	Everyone
	Redesign PCB	Dhruva
	Second PCB order - Thursday	Dhruva
	Talk with machine shop and give parts received - Friday	Rohan
3/17 (spring break)		
3/24	Assemble new PCB and test	Taniah
	Modify the schematic if needed	Everyone
	Work on and test authentication subsystem	Rohan
	Work on and test control subsystems	Dhruva
	Work on user interface	Taniah
	Redesign PCB	Dhruva
3/31	Third PCB order - Monday	Dhruva
	Work on and test box mechanics subsystem	Rohan
	Assemble new PCB and test	Taniah
	Individual progress reports due - Wednesday	Everyone
4/7	Fourth PCB order - Monday	Dhruva
	Work on and test Heating subsystem	Taniah
	Assemble new PCB and test	Rohan

4/14	Team Contract assignment - Friday	Everyone
	Finalize all components of the box	Everyone
	Prepare for mock demo	Everyone
4/21	Mock Demo with TA - Tuesday	Everyone
	Prepare for final demo	Everyone
4/28	Final Demo	Everyone
	Mock Presentation	Everyone
	Prepare for final presentation	Everyone
	Work on final paper	Everyone
5/5	Final Presentation	Everyone
	Final Paper due - Wednesday	Everyone

4. Ethics and Safety

Use of Open Source Projects:

The work that we will be doing in this project will use open source projects as needed. It is important to make sure that the open source projects that we use get credit. This is in accordance with the ACM code of ethics 1.5[14]. It is important to recognize those who have helped us make this project by giving them credit.

Privacy:

Because our project requires us to use fingerprints we will not use the fingerprints that users store for anything other than opening the secure box. The fingerprints will be directly stored on the microcontroller and will not be sent anywhere else. This is in accordance with the ACM code of ethics 1.6 [14]. It is to make sure that users know that their information is being stored securely.

Safety:

Voltage regulators will be used in this project to ensure that the voltage source and that the pcb do not overheat and /or short circuit. Using Lab safety equipment when appropriate also will be done to uphold maximum safety. The heating subsystem also must not overheat to dangerous levels above 80 degrees Fahrenheit. To make sure this does not happen we will perform calculations to make sure the voltage and current we are supplying the nichrome make it so it never reaches above 80 degrees Fahrenheit. Also rigorous safety testing of all parts of the project will be done before it is finished to ensure it works as intended. This is to ensure that IEEE I.1 [15] is followed and that no one is harmed because of an error this project has made.

Unlawful/Unethical Conduct:

We will not perform any unlawful/unethical activities to create this project such as but not limited to using code/hardware that is not available to use and bribery. IEEE I.4[15] says that bribery and unlawful conduct should not be done to uphold high standards of integrity.

4. References

[1]Statista, "U.S.: Users in the Online Food Delivery Market," *Statista*, 2024. https://www.statista.com/forecasts/891084/online-food-delivery-users-by-segment-in-united-states

[2]"AP63200/AP63201/AP63203/AP63205 3.8V TO 32V INPUT, 2A LOW IQ SYNCHRONOUS BUCK WITH ENHANCED EMI REDUCTION Description Pin Assignments," 2019. Available: https://www.diodes.com/assets/Datasheets/AP63200-AP63201-AP63203-AP63205.pdf

[3]"Pin Assignments (Top View) (Top View)," 2017. Available: https://www.diodes.com/assets/Datasheets/AP2112.pdf

[4]"et 1A LDO Voltage Regulator DESCRIPTION." Available: https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/5011/AMS1117.pdf

[5]"ESP32-S3-WROOM-1 ESP32-S3-WROOM-1U Datasheet 2.4 GHz Wi-Fi (802.11 b/g/n) and Bluetooth ® 5 (LE) module Built around ESP32-S3 series of SoCs, Xtensa ® dual-core 32-bit LX7 microprocessor Flash up to 16 MB, PSRAM up to 8 MB 36 GPIOs, rich set of peripherals On-board PCB antenna." Available: https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf

[6]"Matrix Keypad Created by Kattni Rembor." Available: https://cdn-learn.adafruit.com/downloads/pdf/matrix-keypad.pdf

[7]"Adafruit Optical Fingerprint Sensor," 2023. Accessed: Feb. 13, 2025. [Online]. Available: https://www.mouser.com/datasheet/2/737/adafruit_optical_fingerprint_sensor-3364191.pdf

[8]"Solenoid Lock Diagram Specifications Electrical Specifications." Accessed: Feb. 13, 2025. [Online]. Available: https://www.farnell.com/datasheets/2865757.pdf

[9]"Weight Sensor (Load Cell) 0-500g SKU 314990000." Accessed: Feb. 13, 2025. [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/Seeed%20Technology/314990000_Web.pdf

[10]"INA125 INSTRUMENTATION AMPLIFIER With Precision Voltage Reference." Available: https://www.ti.com/lit/ds/symlink/ina125.pdf

[11]“General Description Benefits and Features • Unique 1-Wire ® Interface Requires Only One Port Pin for Communication • Reduce Component Count with Integrated Temperature Sensor and EEPROM • Measures Temperatures from -55°C to +125°C (-67°F to +257°F) • ±0.5°C Accuracy from -10°C to +85°C • Programmable Resolution from 9 Bits to 12 Bits • No External Components Required • Parasitic Power Mode Requires Only 2 Pins for Operation (DQ and GND) • Simplifies Distributed Temperature-Sensing Applications with Multidrop Capability • Each Device Has a Unique 64-Bit Serial Code Stored in On-Board ROM • Flexible User-Definable Nonvolatile (NV) Alarm Settings with Alarm Search Command Identifies Devices with Temperatures Outside Programmed Limits Ordering Information appears at end of data sheet.” Available:
https://www.analog.com/media/en/technical-documentation/data-sheets/ds18b20.pdf

[12]“Nichrome Wire,” *Online Calculator*, May 23, 2023. https://calchub.xyz/nichrome-wire/ (accessed Mar. 05, 2025).

[13]Grainger, “Salary Averages,” *Illinois.edu*, 2021. http://ece.illinois.edu/admissions/why-ece/salary-averages (accessed Mar. 05, 2025).

[14]Association for Computing Machinery, “ACM code of ethics and professional conduct,” *Association for Computing Machinery*, 2018. https://www.acm.org/code-of-ethics

[15]IEEE, “IEEE Code of Ethics,” *ieee.org*, Jun. 2020. https://www.ieee.org/about/corporate/governance/p7-8.html