Critical Design Review

P.U.M.P.E.D

Precision Weighing, User-Friendly Heating, Mobile App Integration, Printed Labels, Expiration Tracking, and Data Management

Group 16

Zachary Celentano

Jason Devito

Jamieson Foley

Isaiah Robinson

Sponsors

Victoria Rodriguez

Matthew Aberman

Advisors

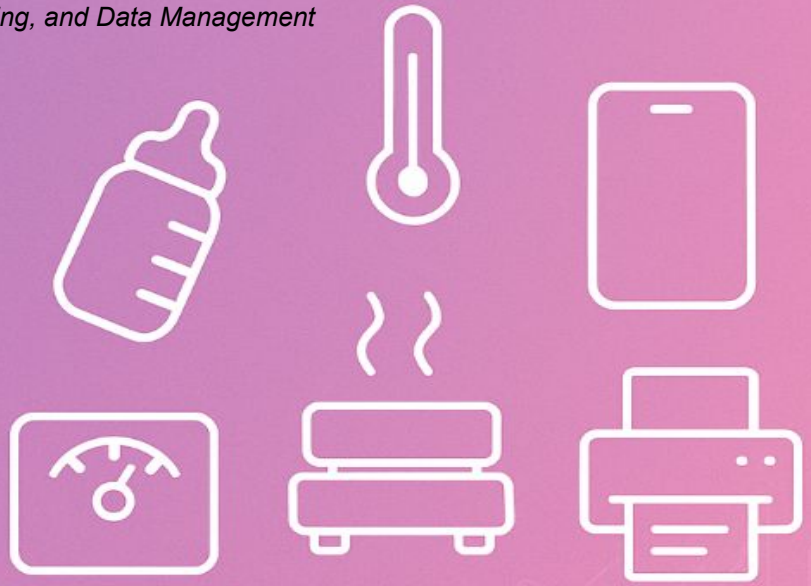
Dr. Suboh Suboh

Dr. Wei Sun

Dr. Hadi Kamali

Mentor

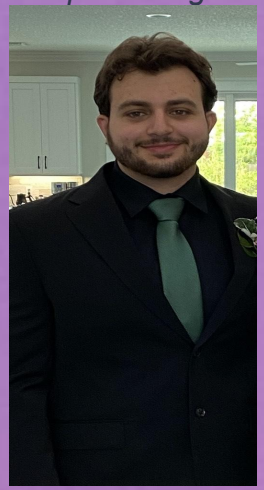
Dr. Chung Yong Chan



MEET THE TEAM

Zachary Celentano
Electrical Engineering

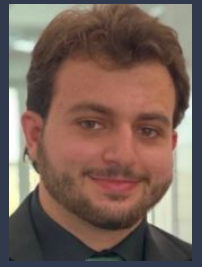
Jason Devito
Computer Engineering



Isaiah Robinson
Computer Engineering

Jamieson Foley
Electrical Engineering

Problem and Motivation



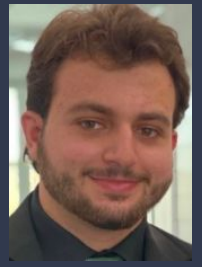
Parents currently must weigh each milk bag, write down weight, date, and time by hand, then heat it in a separate hot-water bath—steps that are slow, prone to mistakes, and unsafe if the temperature is not monitored carefully. No single product exists to combine weighing, labeling, warming, and tracking, leaving caregivers to juggle multiple tasks and worry about wasted or expired milk.

- Manual recording is time-consuming and often leads to mislabeling
- Inaccurate dates cause milk to expire unnoticed or be discarded unnecessarily
- Heating in a hot-water bath lacks precise temperature control, risking safety

The current solutions really do not **cut** it. We need to give our mothers an **easier** time!

- Existing tools only handle one step (weighing, labeling, or storage) rather than the whole process
- Our device aims to **centralize** these operations

Goals



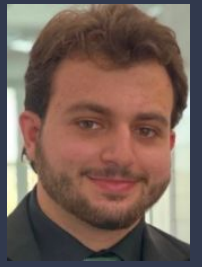
Basic Goals:

- Build a product that allows parents to easily weigh, label, and store milk bags.
- Build an application that allows an accessible interface with the data of bagged milk.
- Build a system that ensures milk is stored and used at safe temperatures, helping prevent spoilage or degradation of nutrients.

Stretch Goals:

- Allow allergen tracking within application.
- Allow for user data to be stored within a cloud storage database.
- Provide feedback to parents within app based on usage.
- Maintaining tracked data for a period of up to a year.

Objectives



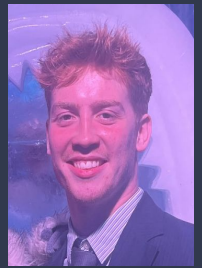
Basic Objectives:

- Implement a load cell and ADC for accurate weight measurement. Convert the measured weight to fluid ounces for parents ease of use.
- Integrate a label printer to produce labels containing all relevant information, including fluid ounces, date, time, and temperature.
- Show real-time weight, date, and time on the LCD prior to label printing.
- Measures the temperature of bagged milk using thermal coupling.
- Implement a heating element to allow parents to more freely manage milk temperature.
- Implement a custom power supply to power the device and all peripherals.
- Any surface that makes contact with the milk bags will be able to be taken apart for sanitation purposes.
- Develop an iOS app to interface with a database of information regarding stored milk.
- Provide features for viewing, editing, and managing milk inventory.
- Send alerts for impending expiration dates and low general storage.

Stretch Objectives:

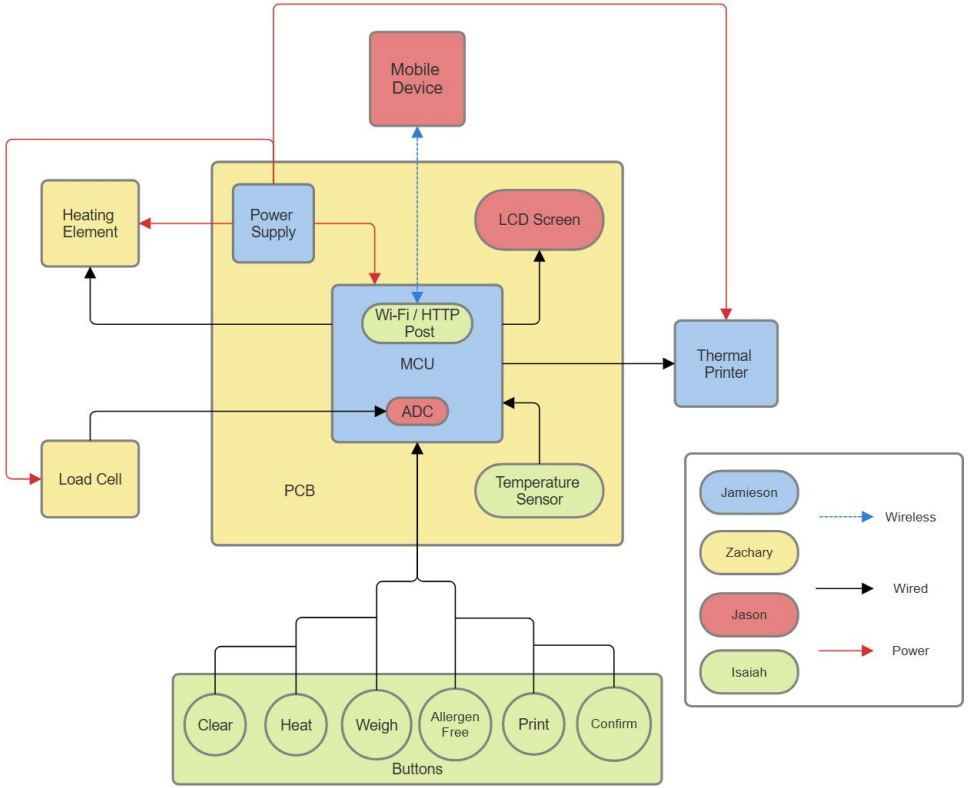
- Build a custom thermal printer and its subsequent PCB to minimize form factor.
- Design the power supply to be able to work with batteries so the product can be portable.

Engineering Specifications



Parameter	Value	Description
Weight Measurement Accuracy	± 1 Grams	Accurate weight measurements.
Temperature Measurement Accuracy	± 1 °F	Accurate temperature measurements.
Label Size	2” x 1”	Large enough to display all data.
LCD Display Time	< 1 Second	Fast response time.
Power Consumption	< 80 Watts	Overall power consumption device.
Bluetooth Connectivity Response Time	< 5 Seconds	Efficient interface connectivity.
Local Data Storage Capacity	> 20 entries	Enough data storage capabilities to give parents enough information.
Heating Temperature	90-100 °F / 35-38 °C	Must heat milk to the proper temperature of consumption.
Cost Estimate	< $800	Needs to be cost effective to manufacture.

Hardware Block Diagram



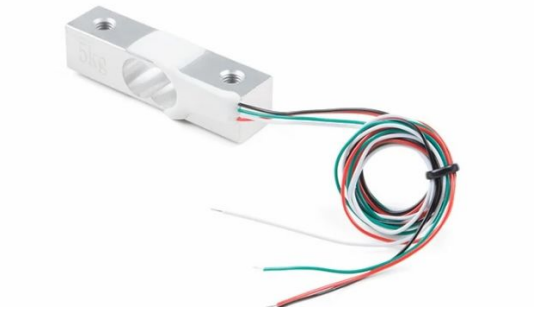
Hardware Comparison: Weight Measurement

- Provides the highest accuracy, with the added benefit of not needing to be re-calibrated over time.
- Although the prices can vary greatly depending on the size of the load cell, we only need a small and cheap one.
- Relatively simple to implement.

Method	Cost	Implementation	Accuracy
Load Cell	Moderate (Anywhere between $ & $$$)	Moderate (Requires an amplifier IC)	High (Does not lose accuracy over time)
Piezoelectric Force Sensors	High (Between $$ & $$$)	Difficult (Requires Charge amplifier)	Moderate (Loses accuracy with time, best for alternating forces)
Capacitive Force Sensors	Low (Between $ & $$)	Moderate (Requires Capacitive Sensing IC)	Moderate (Sensitive detection, best for low force detection)

Hardware Selection: Load Cell

- Includes an amplifier IC with the purchase.
- Has the lowest cost.
- Manufacturer offers the schematic design of the amplifier IC necessary for operation.



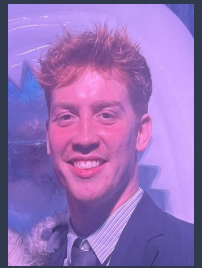
Product	Cost	Implementation
Phidgets CZL635 (5kg)	$7 per unit + $30 - $95 for proprietary IC	Easy (Can implement amplifier IC into PCB design)
SparkFun SEN-14729 (5kg)	$13.12 per unit	Moderate (Requires amplifier design and implementation)
TAL220B (5kg)	$11.95 per unit	Moderate (Requires amplifier design and implementation, which is provided by the seller)

Hardware Comparison: Label Printer

- Requires the least amount of maintenance.
- Has the lowest cost.
- Greatest disadvantage is finding one that will easily be implemented into our device.

Label Printer	Cost	Implementation	Maintenance
Thermal printer	Low $50-$100	Easy	Low (fewest moving parts, no toner or ink)
Laser Printer	High $100-$500	Moderate	Moderate (toner needs to be replaced often. Regular cleaning needs to take place)
Inkjet Printer	Moderate $50-$300	Easy	Moderate (regular cleaning of the printhead to avoid clogging)

Hardware Selection: Thermal Printer

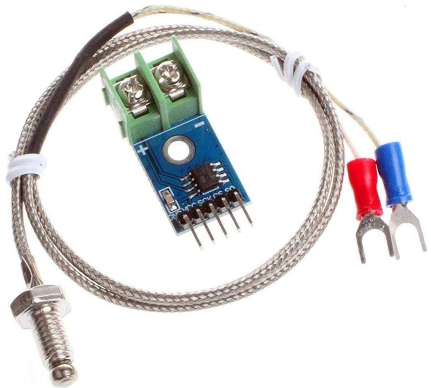


- Our primary concern with the thermal printer is the cost. Being that they all work relatively the same
- We chose the ASPIRON due to its lowered cost and moderate integration through USB or serial port

Thermal Printer	Cost	Integration	Maintenance
MUNBYN IMP001 58mm Thermal Printer (Serial/UART & USB)	Moderate $40-$80	Moderate as it uses UART or USB connection	Moderate maintenance as needed depending on use of paper, also dependent on jams.
PRIMUZ Micro Thermal Printer	Moderate $45-$65	Moderate, can be connected through USB or through serial port	Ventilation may be a concern, as overheating could possibly occur.
ASPIRON 58mm Embedded Thermal Printer Module (UART & TTL Interface)	Moderate $45-$70	Moderate; connected through USB or through serial port	Similar ventilation problems are possible, also may need additional protection from dust buildup.

Hardware Selection: Temperature Sensor

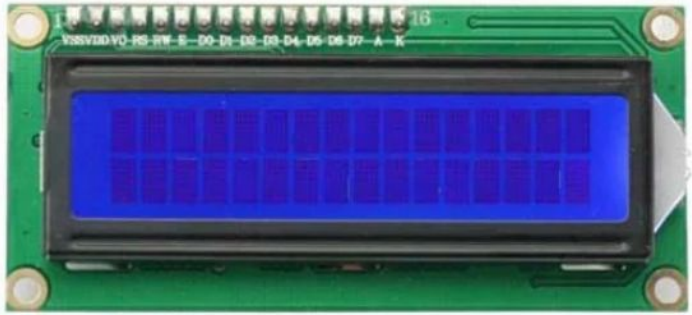
- Moderately priced but makes up for it in efficiency
- Simple and easy integration
- Low maintenance due to minimal upkeep
- Cleaning of this sensor will need to occur so that the sensor does not degrade from outside elements



Temperature Detector	Cost	Integration	Maintenance
Type K Thermocouple with MAC6675 Amplifier	Moderate $4-$13	Easy to integrate. Uses SPI making it easy to connect to most microcontrollers	Low, durable with minimal upkeep. Cleaning will need to occur to prevent degradation of the sensor.
PT100(RTD-Resistance Temperature Detector)	High $10-$100+ Varies significantly with accuracy and material	Moderate to Complex. Requires an excitation current, specialized amplifier and signal processing	Moderate to High as it is more fragile and needs careful wiring. Needs to be calibrated overtime
NTC Thermistor (Negative Temperature Coefficient Sensor - 10KΩ or 100KΩ)	Low $0.50-$5	Easy to Moderate. Requires voltage divider, but is more sensitive to noise	Moderate as it is prone to aging and drift. This also needs to be recalibrated over time

Hardware Selection: Display

- Has lowest cost and is the easiest to integrate into our system.
- Requires low maintenance as it is durable with simple graphics.
- Has the option of being upscaled to a 20x4 character LCD with little to no changes to be integrated into our system.



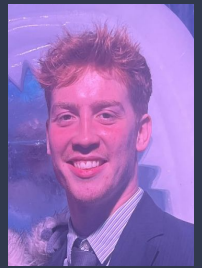
LCD Display	Cost	Integration	Maintenance
16x2 Character LCD (HD44780 - I2C or Parallel Interface)	Low $3-$10	Easy integration with simple character display and uses I2C	Low as it is durable with no complex graphics. Has minimal issues
128x64 Monochrome LCD (ST7920 - SPI/I2C/Parallel Interface)	Moderate $6-$15	Moderate integration and requires SPI/I2C/Parallel interface. Also needs a graphics library to support simple graphics and texts	Low as it is reliable, but backlight or contrast may degrade overtime
2.8 inch TFT LCD (ILI9341 - SPI Interface, Full Color)	High $10-$30	Complex integration, requiring graphics library, more memory and processing power needed	Moderate as it is more fragile. Color display may degrade and the touchscreen may wear out

Hardware Comparison: Heating

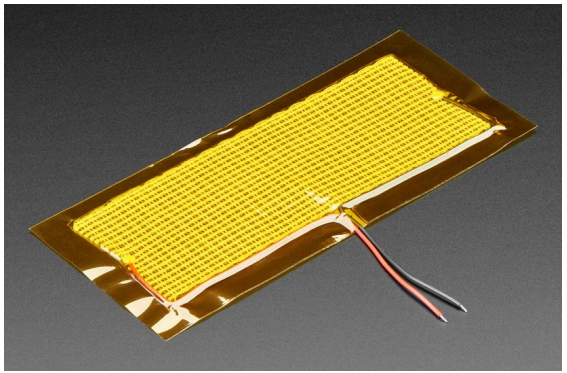
- Conduction has the highest proficiency in heating
- Although the operational cost is higher, its efficiency and ability to heat an item directly is why we chose it.
- If our conduction is efficient enough we will experience convection as we are heating a stainless steel cup filled with water

Method	Efficiency	Implementation
Conduction	High (Element makes direct contact with subject, and direct heat transfer is better for small area heating)	Easy as the heating element is directly in contact with the water which would transfer the heat efficiently
Convection	Moderate (heats large spaces well, requires consistent fluid motion)	Moderate (implementation of forced convection would require moving parts to move fluid)
Induction	High (requires significant space for coils generating magnetic current, energy efficient)	Difficult (requires complex power management due to AC requirement for operation)

Hardware Selection: Heating Element



- The second lowest cost option and is a simple resistor that heats up
- Low maintenance as there is no clean up due to moving parts
- We wanted the heating option to not be a burden on the user due to cleaning and maintenance



Heating Pad	Cost	Integration	Maintenance
Electric Heating Pad 14cm x 5cm by Adafruit	Low $6	Easy to integrate as it is thin and flexible and can be attached to a surface by an adhesive. Requires MOSFET or Relay for control	Low maintenance as there are no moving parts. The adhesive will weaken over time
Silicone Heating Pad 15cm x 5cm by Meccanixity	Medium $10	Easy to integrate as it is thin and flexible and can be attached to a surface through an adhesive. Requires MOSFET or Relay for control	Low maintenance as there are no moving parts. Very durable choice. Adhesive may degrade overtime
Cartridge Heater by MakerTechStore	Low $4	Moderate to integrate as it requires a hole or slot to be inserted into a heating block or reservoir of water. Requires MOSFET or Relay for control	Medium maintenance as it can accumulate limescale. Requires regular cleaning.

Hardware Selection: Microcontroller

- The ESP32 is the ideal choice as it provides the most features at an affordable cost
- It enables both bluetooth support as well as Wifi support, which is ideal for our MCU of choice

Name	WiFi	Bluetooth	RAM	Clock Speed	Cost
ESP32	Yes	Yes	520 KB	240 MHz	Low ($ per unit)
STM32	No	No	512 KB - 1MB	84-216 MHz	High ($$ per unit)
nRF52840	No	Yes	1 MB flash, 256 KB RAM	64 MHz	Medium ($-$$ per unit)

We plan to use buttons for the user interaction as they are able to be used in multiple different ways depending on each function of our system

Hardware Comparison: Power Supply

- Ultimately, the primary determining factor in the power design for our system is cost effectiveness.
- Utilizing grid power for our use case minimizes the necessity for additional infrastructure, like chargers for batteries, or to have stationary solar panel installations.
- Furthermore, as we are going to need to heat water, a larger current draw is necessary for efficient heating, which grid power can readily provide.

Power Supply	Cost	Integration	Maintenance
Grid Power	$5-$50, depending on voltage and current requirements	Requires a power jack and voltage regulation circuit	Low, as it relies on existing power infrastructure
Battery Storage	$20-200, depending on capacity, chemistry, lifespan and energy needs	Requires charging circuit, battery management system and protective enclosures	Medium, because of charging cycles and the battery degrading over time
Solar with Battery Storage	$50-$500, depending on solar panels, charge controllers, a battery storage system, and power output	Requires solar panels, charge controllers, and power management circuitry	High, because of environmental exposure the solar panels and batteries degrade overtime

Hardware Selection: Power Supply

- Since we are using Grid Power infrastructure, we need a way to turn the 120 Volt outlet output to a usable power.
- To do this, we are choosing a signal transformer.
- The signal transformer will drop the voltage of the output, which will input to a rectifier circuit that outputs approximately 17 volts
- This option is more expensive, but will afford us modularity in our design if power needs change.

Product	Cost	Implementation	Maintenance
Bel Signal Transformer with Rectifier Circuit	Moderate ($30-$50)	Hard (We will have to self build everything to our design specification)	High (with a lot of individual parts, there is a possibility of having to fix individual parts if problems arise)
LRS-100-24 (SMPS Adapter)	Low ($17)	Easy to Moderate (Comes pre-built)	Moderate (cleaning may need to occur to prevent dust build up)
Alitove Power Supply(AC/DC Adapter)	Low ($23)	Easy to Moderate (Comes pre-built)	Moderate (cleaning may need to occur to prevent dust build up)

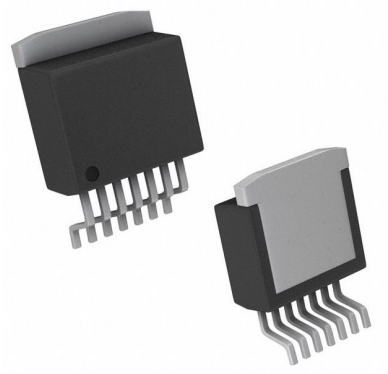
Hardware Comparison: Linear v. Switching Regulators

Type	Design Complexity	Integration	Cost
Linear	Low	Easy	Low
Switching	Moderate	Moderate	Moderate

- With regards to regulators, we need regulators that can handle significant current from our supply and potentially output larger amperage.
- Switching regulators, while generally more expensive and complex in design, are the more viable choice because they can handle more current.

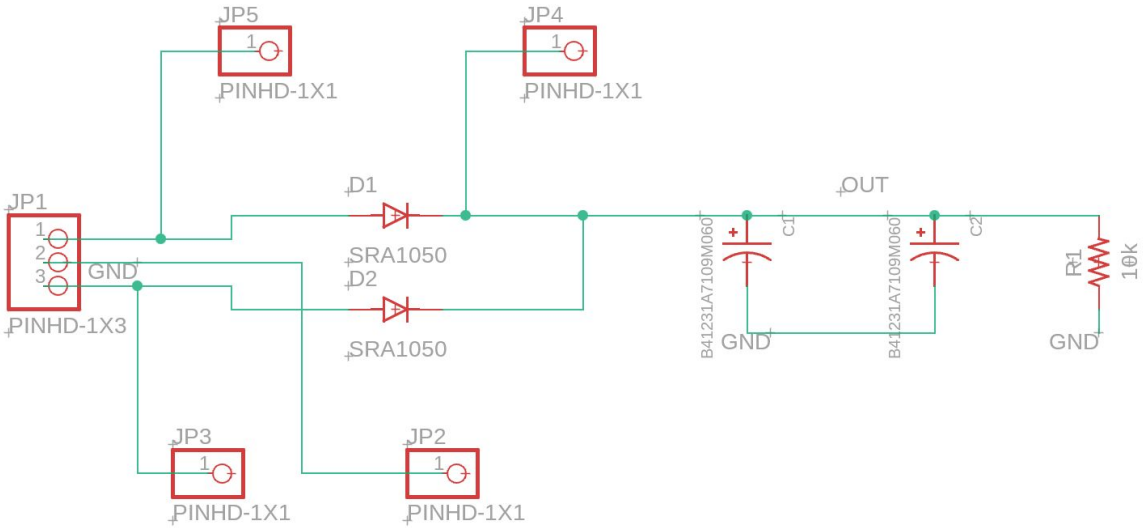
Hardware Selection: Switching Regulators

- At an affordable price we selected the LM2679 as it will perform with efficiency
- It also has a low integration level at an affordable cost



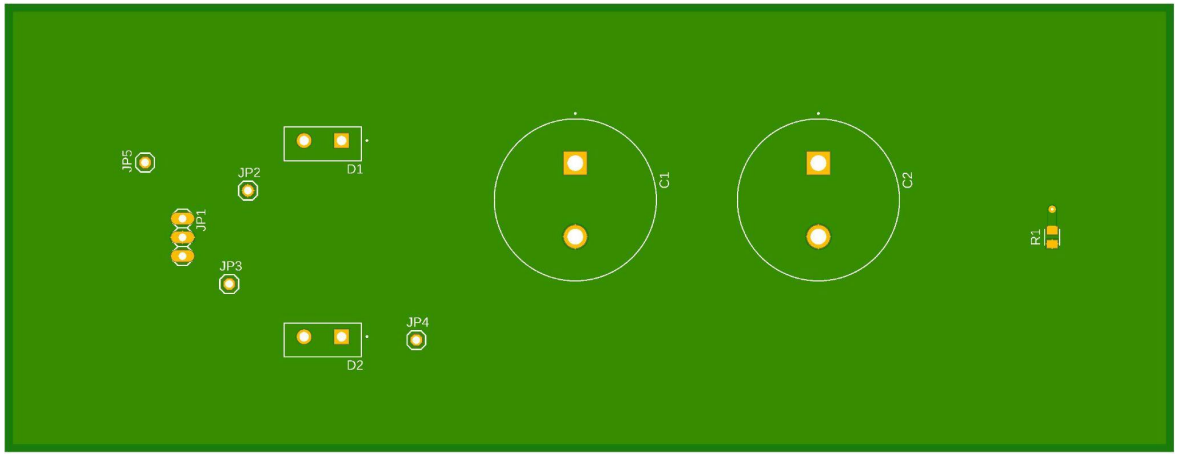
Regulator	Integration Difficulty	Cost
LM2679	Low	Medium
MP2322GQH-Z	Moderate	Low
LTC3355EUF#TRPBF	High	Medium

Hardware Diagram: Rectifier

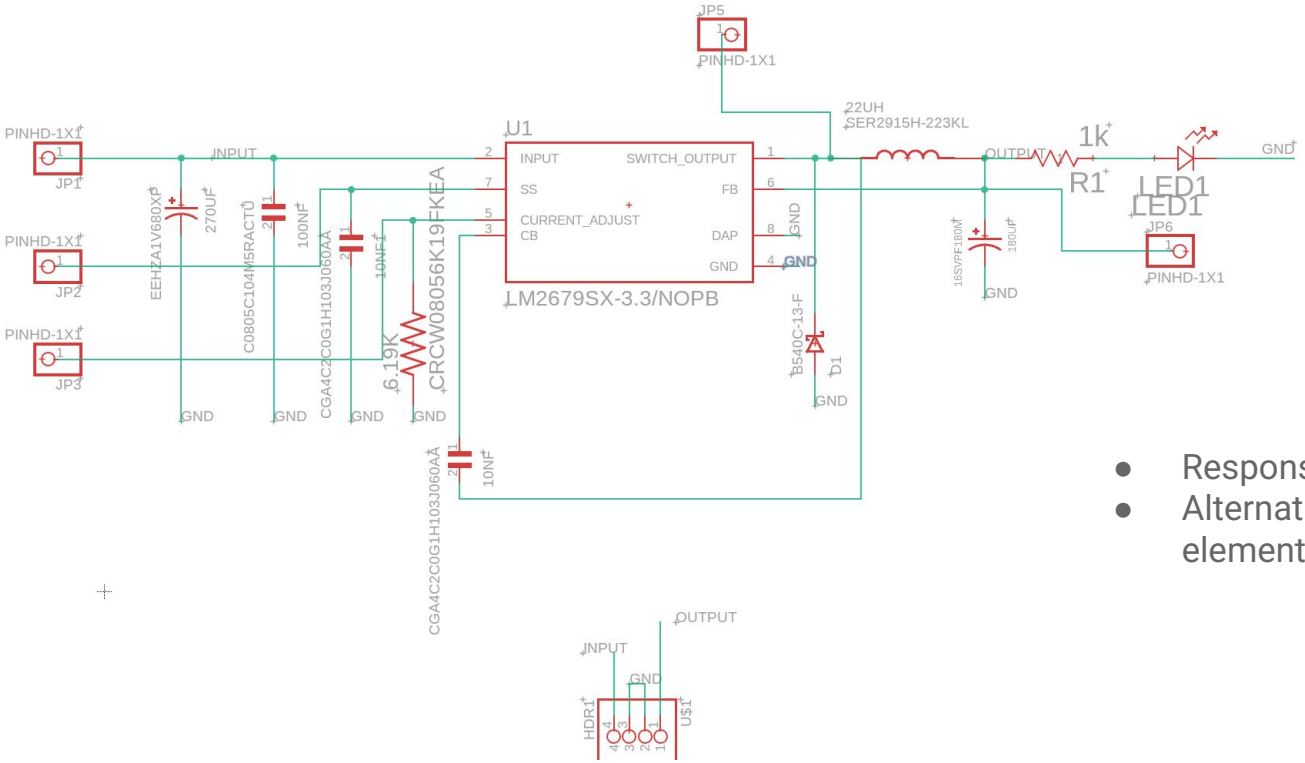


- The noteworthy components in this schematic are the schottky diodes and capacitors.
- We are utilizing a full wave, center tapped rectification circuit.
- The two outputs of the schottky diodes will give us a rectified output and the capacitors will be significantly large as to “smooth” the output response.

PCB Rectifier

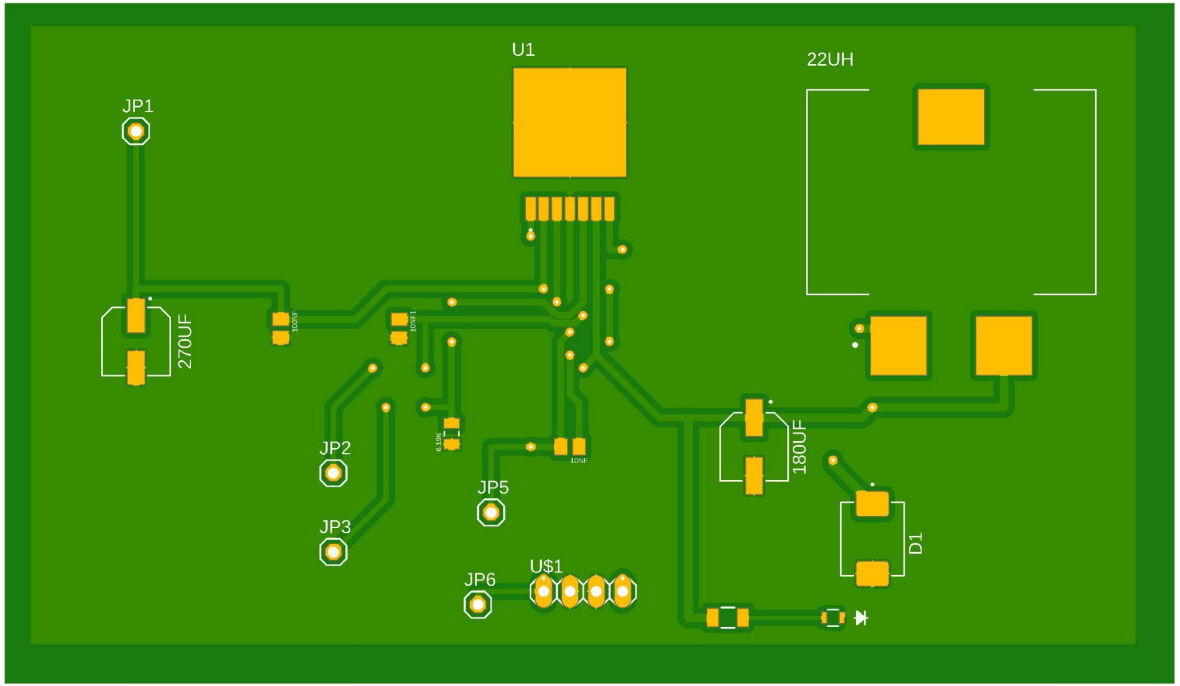


Hardware Diagram: 12 Volt Regulator

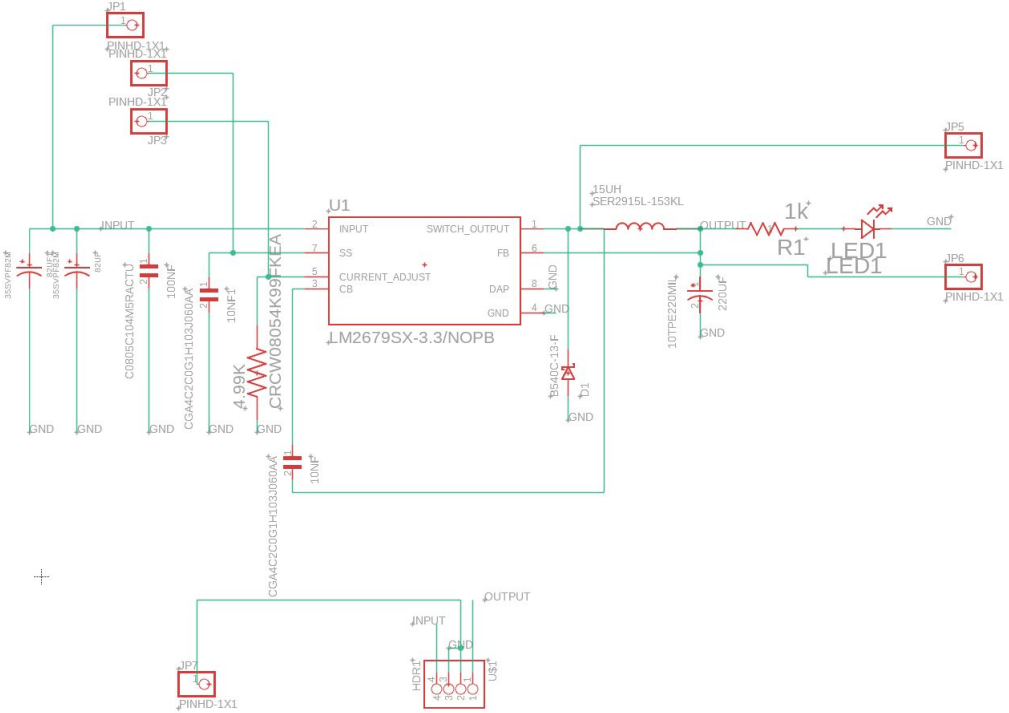


- Responsible for power to printer.
- Alternative power for heating elements.

PCB Regulators: 12 Volt

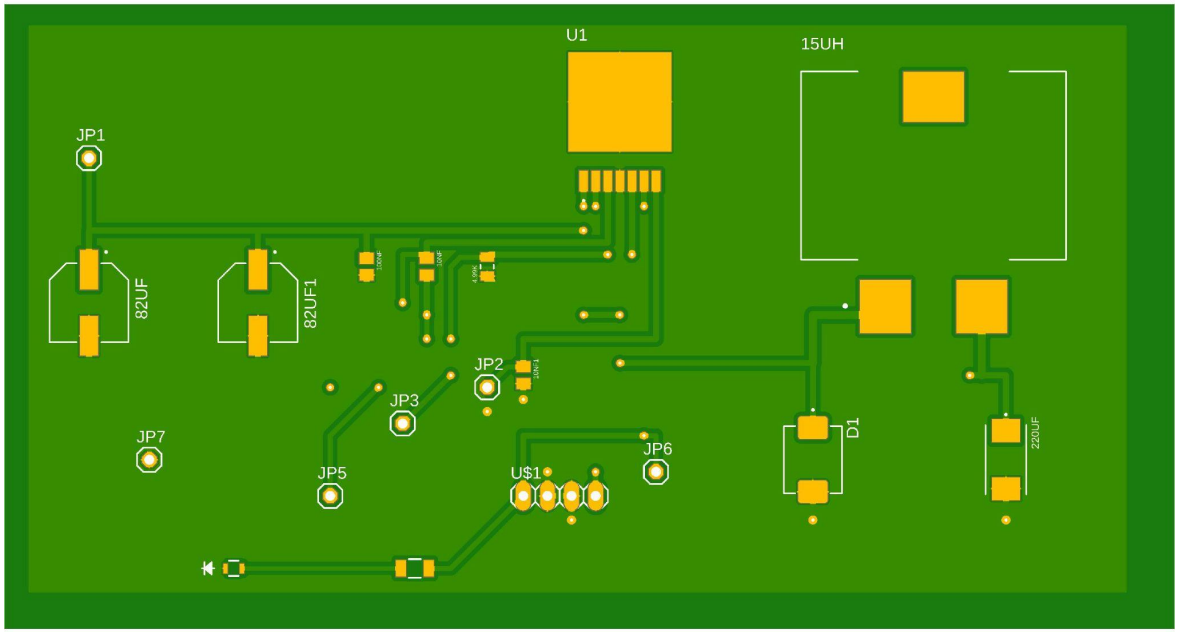


Hardware Diagram: 5 Volt Regulator

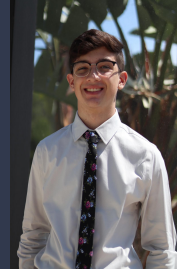


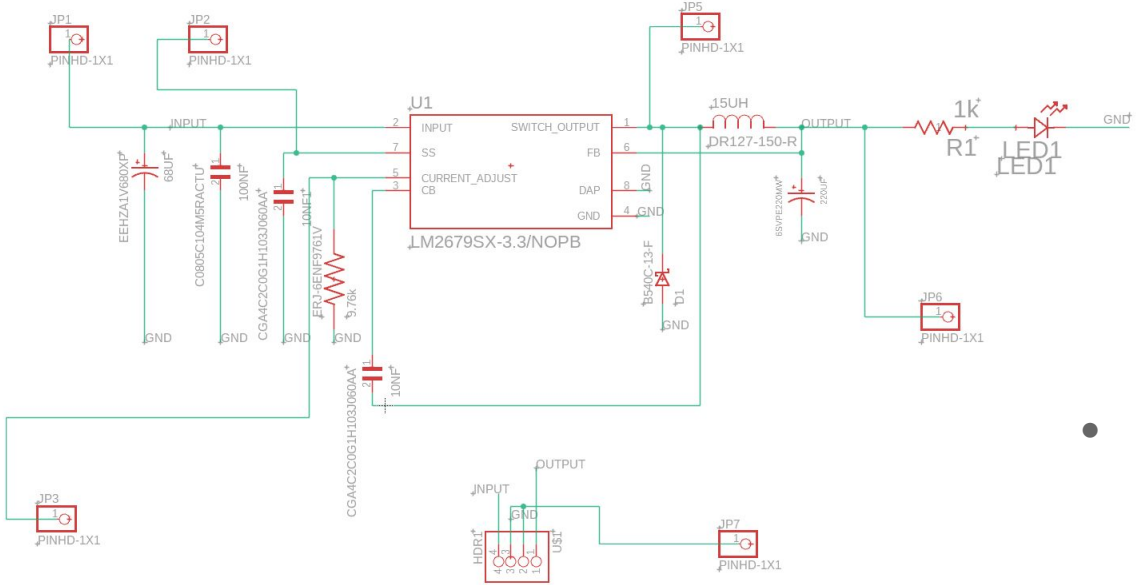
- Power supply for LCD and heating elements.
- Most current draw of regulators.

PCB Regulators - 5 Volt



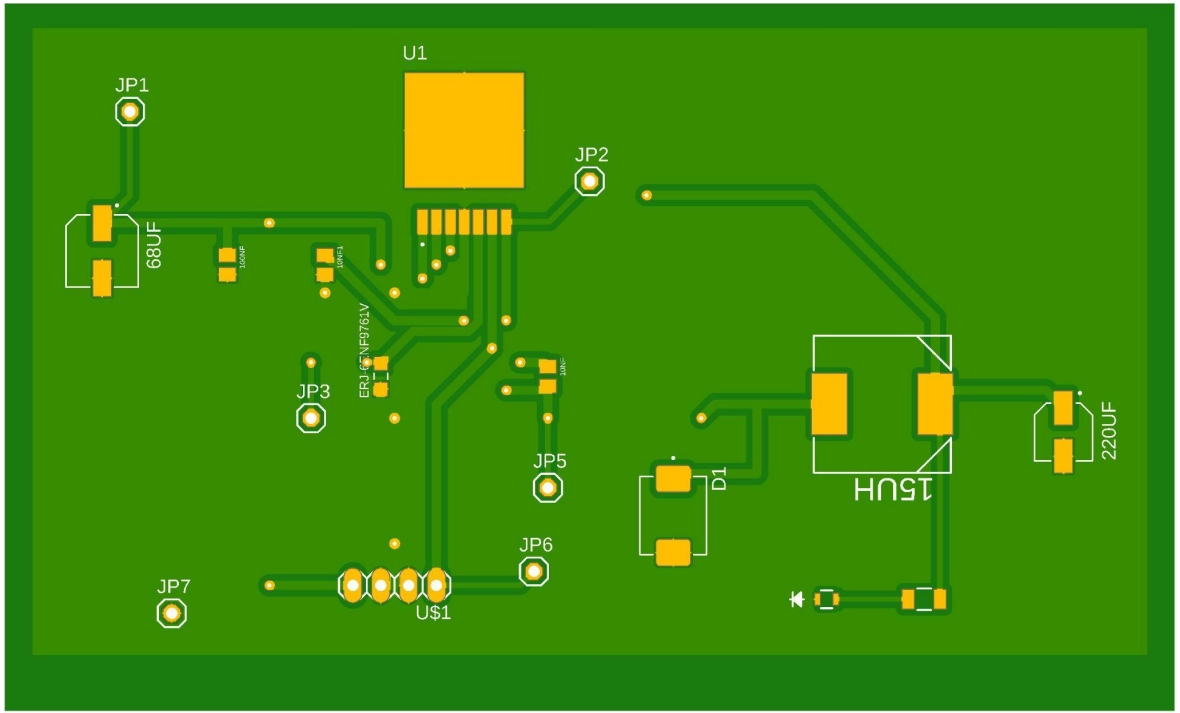
Hardware Diagram: 3.3 Volt Regulator



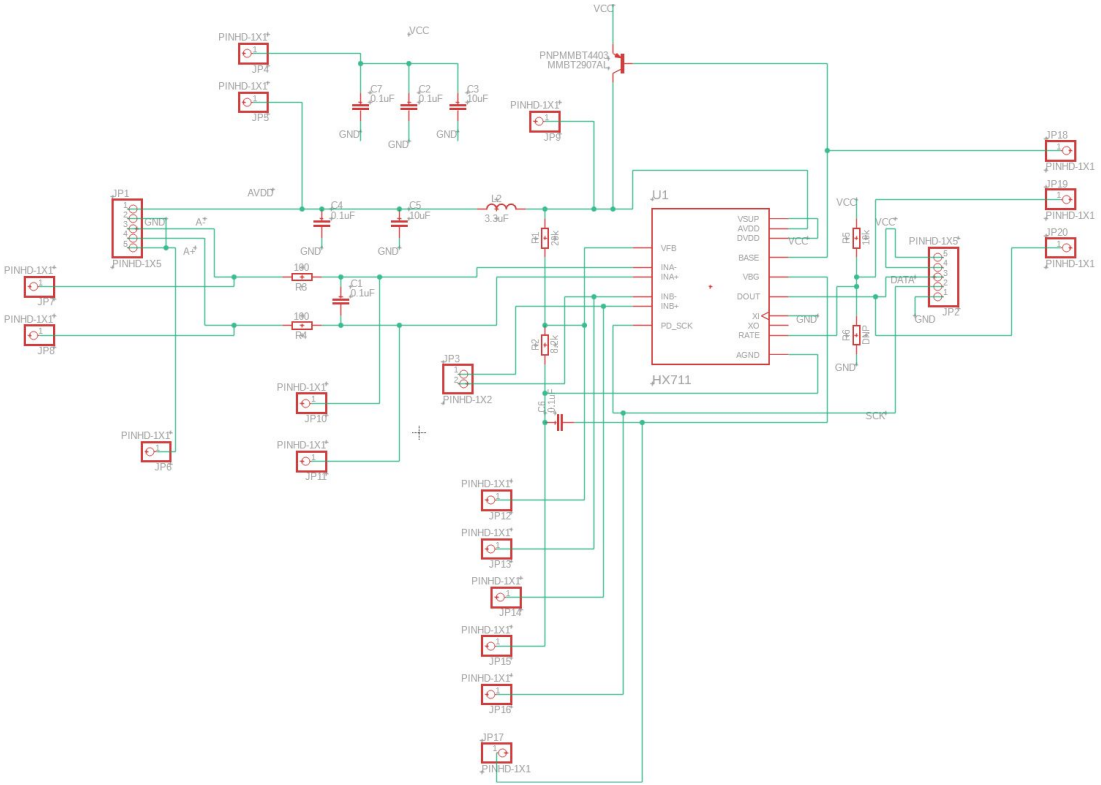


- Provides power to MCU, thermocouple, load cell amplifier, and buttons.

PCB Regulators - 3.3 Volt

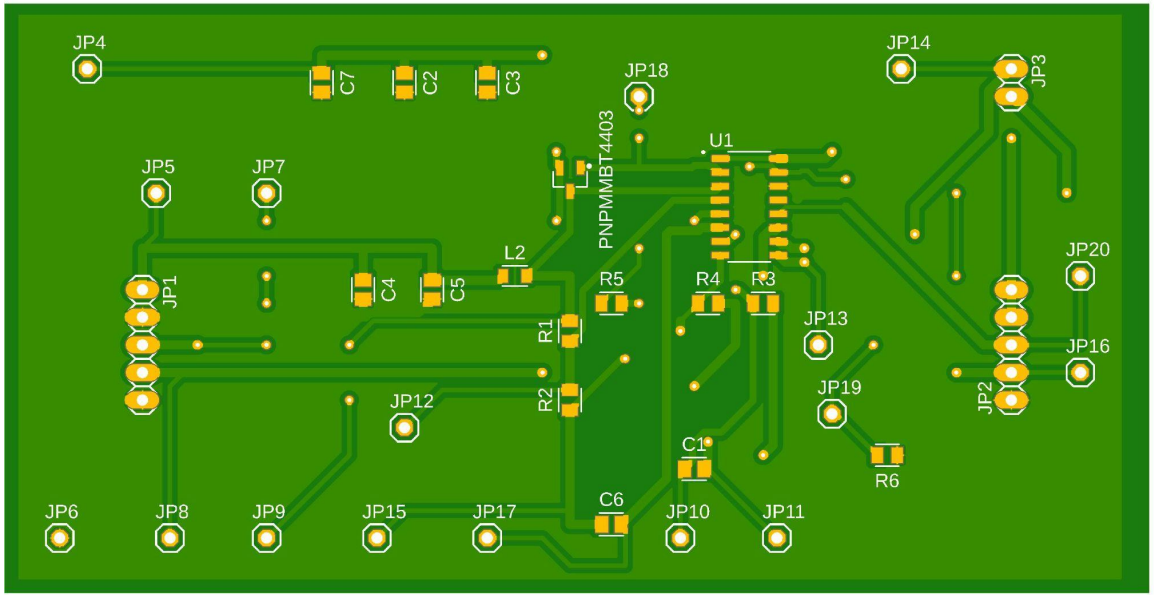


Hardware Diagram: Amplifier

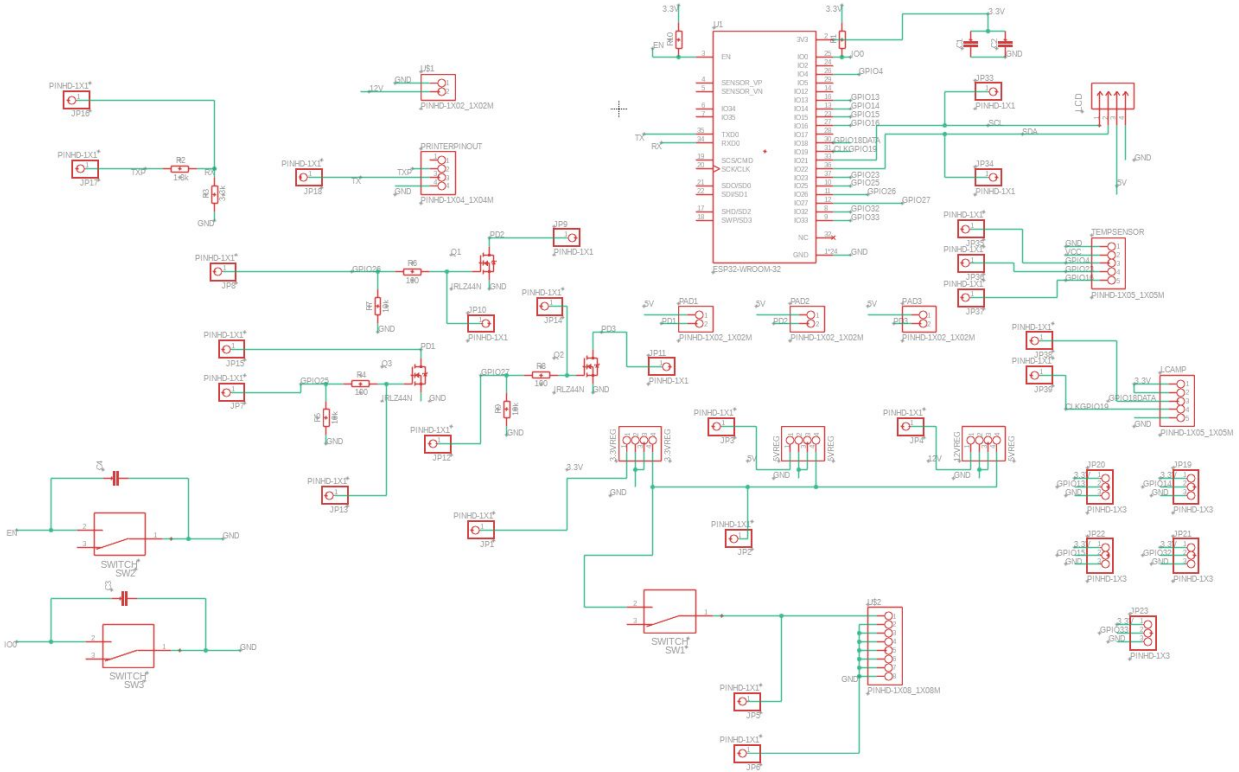


- Amplifier for load cell.
- Necessary for usable readings.

PCB Amplifier

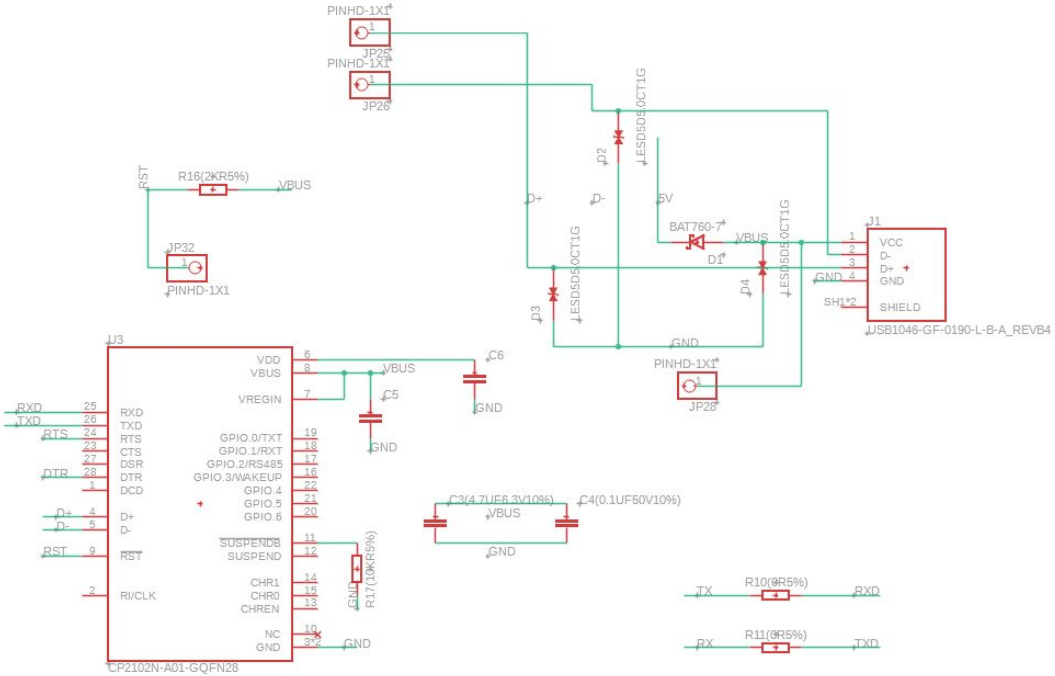


Hardware Diagram: MCU



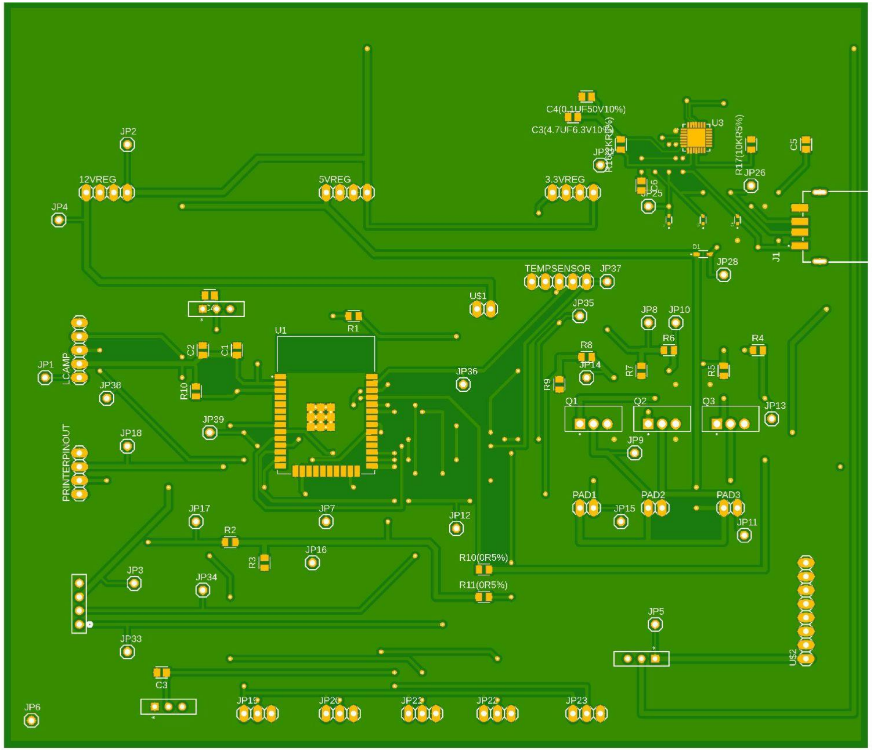
- First section of MCU schematic, which contains all connection points for peripherals.
- Communicates to printer through UART.
- Communicates to the temperature sensor via SPI
- Communicates to the load. cell amplifier via custom two-wire connection.
- Protection circuitry for heating elements.

Hardware Diagram: MCU Cont.



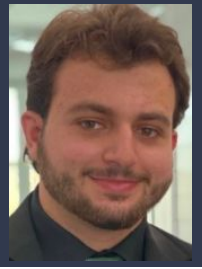
- Second section of MCU which contains infrastructure for programming the ESP32

PCB MCU

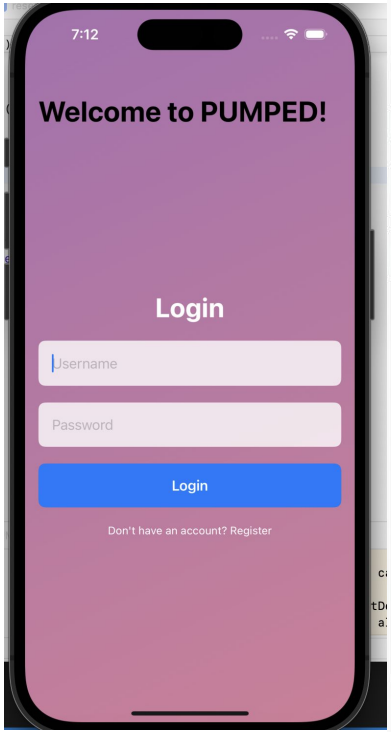


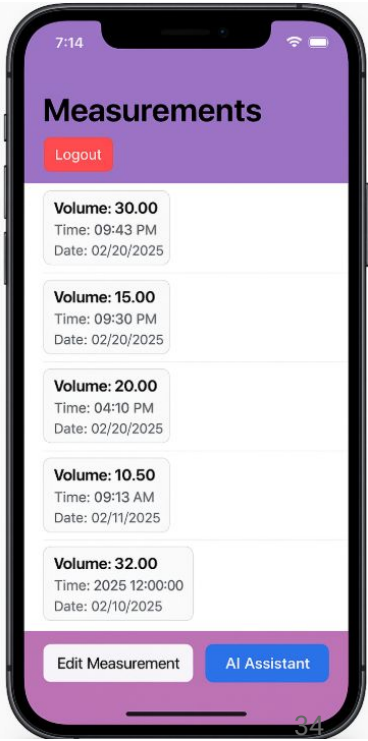
JP2
12VREG
5VREG
3.3VREG
C4(0.1UF50V10%)
C3(4.7UF6.3V10%)
U3
R17(10KR5%)
C5
JP26
JP4
J1
JP28
TEMPSENSOR
JP37
U$1
JP35
JP8
JP10
U1
R1
R6
R4
R8
JP14
R7
R5
C2
C1
JP1
LCAMP
JP36
JP38
R10
R9
Q1
Q2
Q3
JP13
JP39
JP9
JP18
PRINTERPINOUT
PAD1
PAD2
PAD3
JP15
JP11
JP17
JP7
JP12
R2
JP16
R3
R10(0R5%)
R11(0R5%)
JP3
JP34
JP5
U$2
JP33
C3
JP6
JP19
JP20
JP21
JP22
JP23

Mobile Application



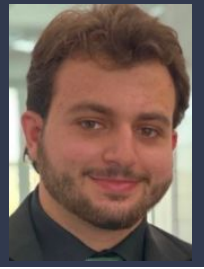
- **User Authentication**
  - Login and Signup
- **Supply Dashboard**
  - Fetch and measurements from backend
  - Graphical Interface with current supply
- **Manual Management**
  - Add, edit, and delete measurements for convenience
- **AI Powered Agent**
  - ChatGPT LLM tuned as “PUMPED” assistant
- **Notifications**
  - Push alerts when supply is low
- **User Interface**
  - The interface is intended to be simple, yet reactive and colorful
  - Custom logos will be implemented by the final demo





The images on the right are not the finalized design and are subjected to being changed

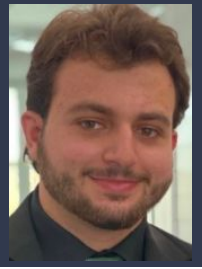
Software Technology Comparison



We evaluated multiple options across three software layers to find the best fit for performance, maintainability, and long-term support

- ***Mobile Application***
  - Swift - Native iOS performance & direct access to Apple SDKs; limited to the Apple ecosystem
  - Flutter/Dart - One codebase for both iPhone & Android; produces larger apps
  - React Native - Leverages web skills for mobile; can feel slower with heavy updates
- ***Backend***
  - Node.js(express) - JavaScript on server for quick data handling; not ideal for heavy calculations
  - Django(Python) - “All-in-one” framework with built-in admin & security; can be bulky for small changes
  - Ruby on Rails - Rapid app setup with conventions; uses more server resources
- ***Database***
  - PostgreSQL - Strong data safety and flexible fields; more work to scale across many servers
  - MySQL - Very common and reliable for straightforward data; limited flexibility for changing schemas
  - MongoDB - Very common and reliable for straightforward data; limited flexibility for changing schemas

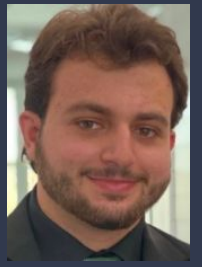
Software Technology Selection



Based on our evaluations, we selected the technologies that best balance performance, ease of development, and long-term support

- **Mobile Application**
  - Swift(XCode)
    - Fast, smooth experience on iPhone
    - Sponsors wanted an IPhone only app
    - Easier to get on appstore
- **Backend**
  - Node.js(Express)
    - Experience using JavaScript in past projects
    - Excellent at handling real time updates
    - Vast ecosystem of readymade packages and libraries
- **Database**
  - PostgreSQL
    - Secure, reliable storage with strict data safety
    - Great for rigid structures data like our project measurements
    - Strong community and industry adoption

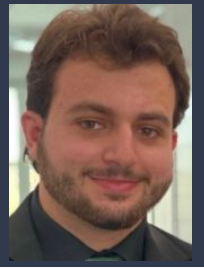
Backend API Routes



No code displayed—just a description of each endpoint's routes, inputs, and actions

- **User authentication**
  - *POST /auth/register*
    - Inputs: email, username, password
    - Action: Validates fields, checks for existing user, hashes password, creates new user
  - *POST /auth/login*
    - Inputs: username, password
    - Action: Validates credentials, compares password hash
- **Measurement management**
  - *POST /measurements*
    - Inputs: volume (in ounces)
    - Action: Captures current date/time, saves record to database
  - *GET /measurements*
    - Inputs: (none)
    - Action: Fetches all measurements ordered by newest first
  - *PUT(edit) /measurements/:id*
    - Inputs: URL param: id
    - Action: Updates an existing measurement
  - *DELETE /measurements/:id*
    - Inputs: URL param: id
    - Action: Removes the measurement record from the database
- **LLM Integration**
  - *POST /generate*
    - Inputs: userQuestion (plain-English requirement
    - Action: Sends prompt to ChatGPT-powered service, retrieves raw gpt answer

Data Handling

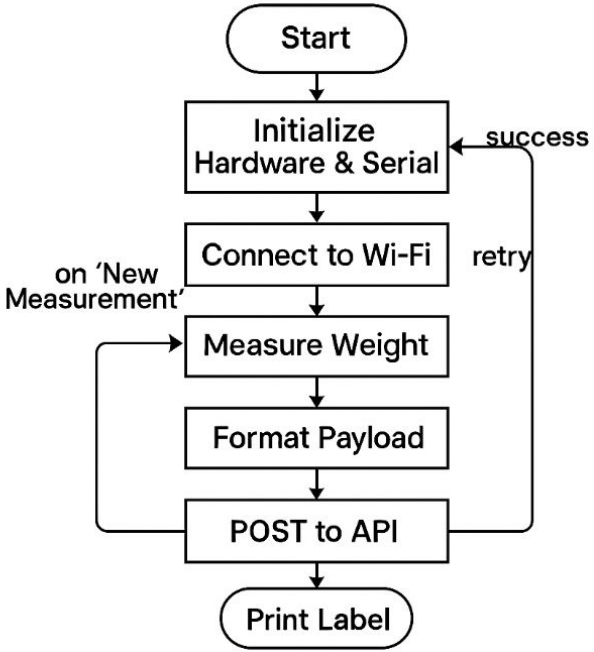


In order to for our firmware to deliver data that is displayed on the mobile application we used some of the ESP32's built in features

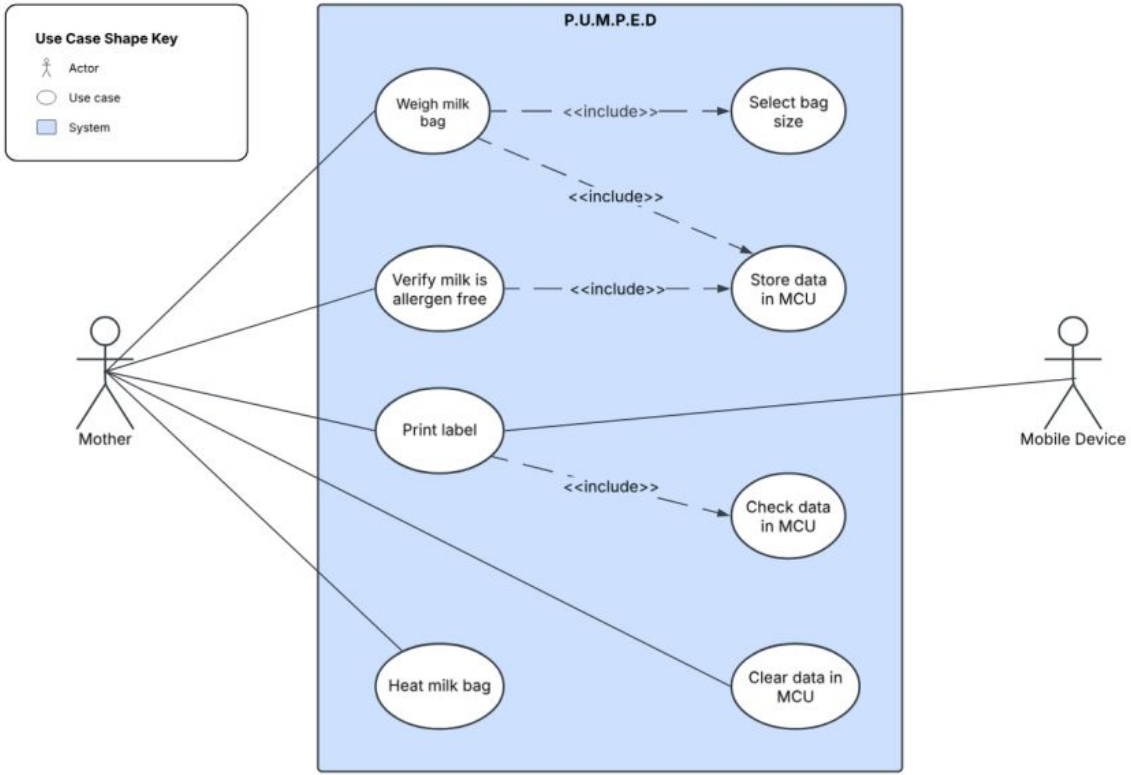
- Wifi
  - Connects to local network using WiFi.begin(ssid, password)
- HTTP Communication
  - HTTPClient for simple POST requests
  - JSON payload with volume, date, and time

Data Flow:

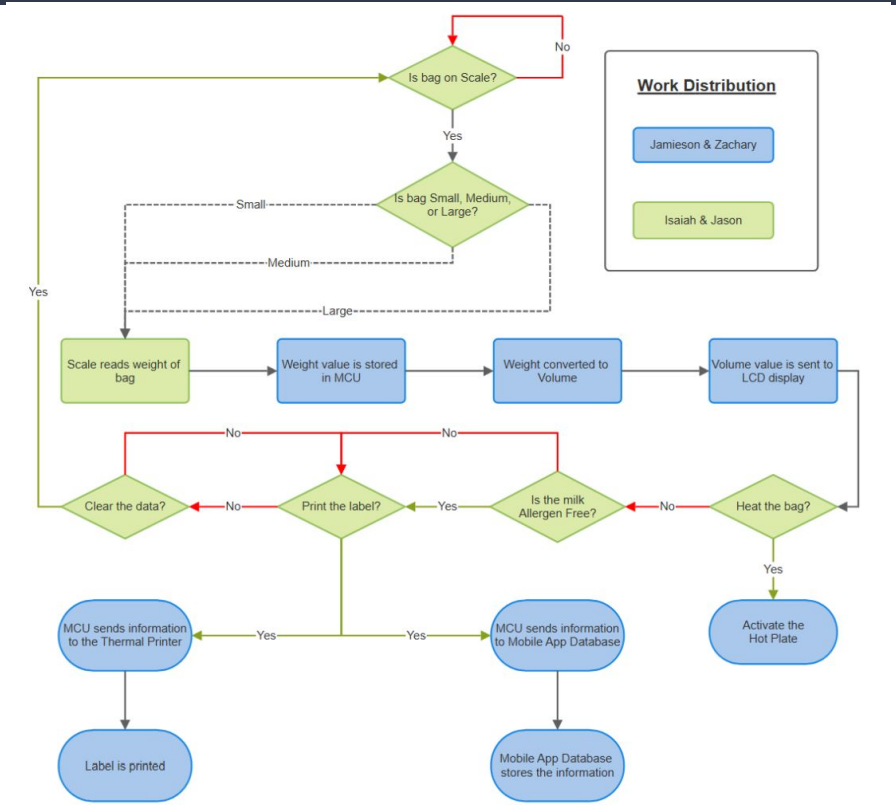
1. Take measurement from the load cell and store in a variable
2. Format variable and send to API server via HTTP POST /measurements
3. Print Label
4. Backend stores in Database
5. Database measurements are fetched and displayed on mobile app



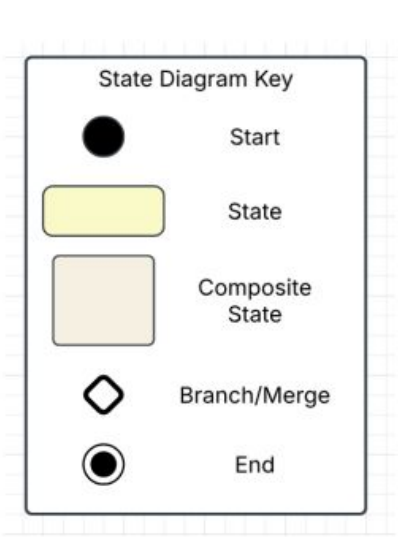
Software Use Case Diagram

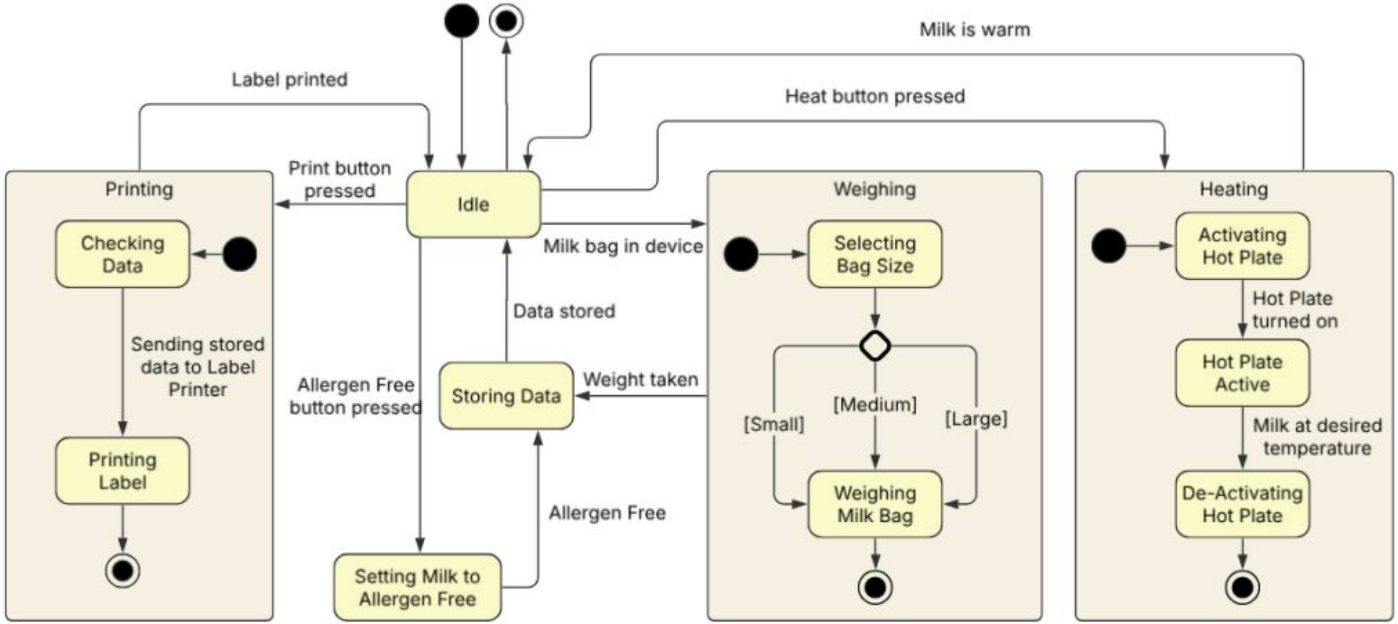


Software Flowchart

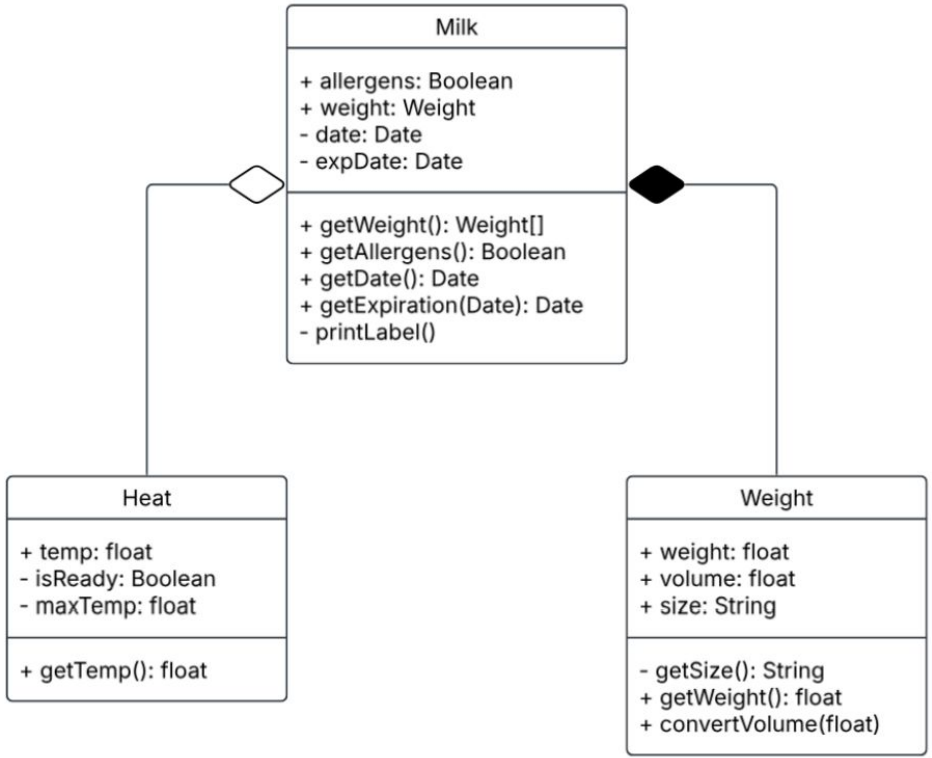


Software State Diagram





Software Class Diagram



Temperature Sensor Testing

- Originally used a waterproof DS18B20 digital temperature sensor
- Usable temperature range: -67°F to +257°F
- Tested temperature: ~75°F

Problems:

- DS18B20 sensor activated, but gave a reading of 0°C (32°F)

Solution:

- Switched to the Type-K Thermocouple temperature sensor
- Gave much more accurate temperature readings

Heat Pad Testing

- Dimensions: 14x5 cm
- Max Voltage: 12 V
- Supplied voltage: 10 V

Problems:

- Reached a temperature of only 76°C after about 10 minutes

Solution:

- Plan to use multiple heating pads wrapped in series around a metal cup to provide quicker and more uniform heat

Load Cell Testing

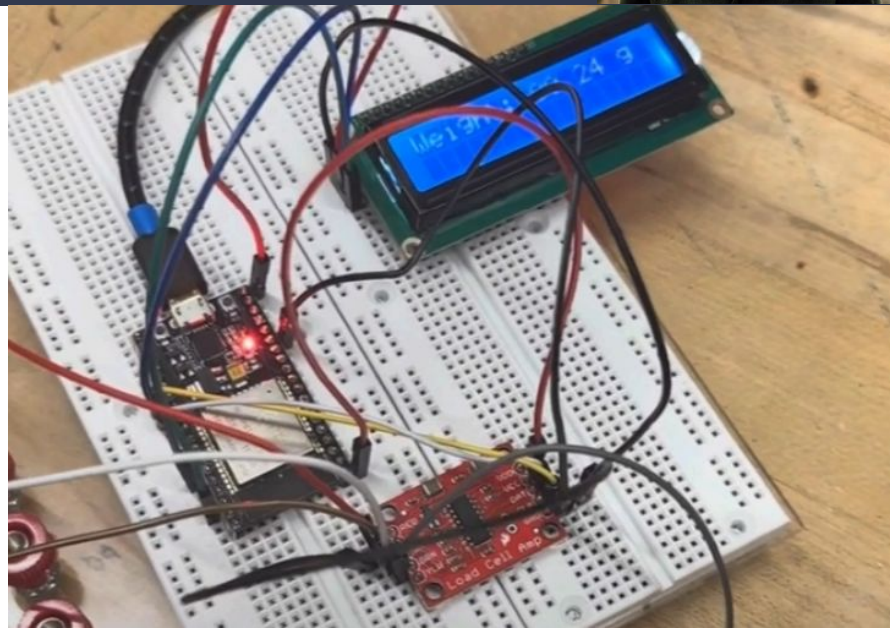
- Max weight: 5 kg
- Tested weight: 100 g

Problems:

- Originally gave a reading of about 90 g
- No longer gives accurate readings of the weight
- Doesn’t update the changes in the weight applied to the load cell

Solutions:

- Upscale to a 10 kg load cell
- Use a different 5 kg load cell



Challenges and Solutions

During our design process we ran into a couple of problems. Below are the main issues we ran into and some solutions we are currently working on.

Hardware:

- Heating efficiency is inconsistent and requires more voltage than expected
  - A solution to this is still to be found. We plan on trying to run the heating elements in series for maximum heating efficiency
- Weighing the bags of milk proved to be difficult.
  - Buying a new load cell as well as a bigger load cell
- Thermal printer integration has proved to be difficult to integrate
  - We included a level shifter to make the connection between the MCU and the Thermal Printer

Software:

- WiFi “brownout” while using hotspot.
  - A solution to this would to experiment using the ESP 32’s BLE since it does not need wifi, or trying to connect to UCFs wifi.

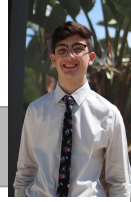
Budget

Component	Total Price
Peripheral Components	Load Cell + Display + Printer + Heating + Temperature Sensor + Power Supply $95.57
PCB Components	ESP32 + Resistors + Capacitors + Inductors + ICs + Diodes + Regulators ~$160 (not all parts are bought)
PCB Manufacturing	$137.03
Enclosure Manufacturing	~$50(last to be built and implemented)
Shipping	$95.69
Total	$538.29

Team Roles and Work Distribution

System/Task	Primary Engineer	Secondary Engineer
Administrative	Zachary Celentano	Jason Devito
Power Supply	Zachary Celentano	Jamieson Foley
Component prototyping & integration	Jamieson Foley/Isaiah Robinson	Zachary Celentano/Jason Devito
Thermodynamic Evaluation	Jamieson Foley	Zachary Celentano
Mobile App Development	Jason Devito	Isaiah Robinson
Backend Development	Jason Devito	Isaiah Robinson
Firmware Programming	Isaiah Robinson	Jason Devito
PCB Design	Zachary Celentano	Jamieson Foley
Enclosure Construction	Jamieson Foley/Zachary Celentano	Isaiah Robinson/Jason Devito

Progress



Component	Category	Status	Notes
ESP32-DevKit (WROOM 38 Pin)	Hardware	Integrated	Breadboard-mounted, USB-C powered
Load Cell and HX711 amplifier	Hardware	In Progress	Reads & displays weight, not yet fully calibrated. Needs more testing
Type-K ThermoCouple and MAX6675	Hardware	Integrated	Ambient & ice-bath tested
Resistive Heating Element	Hardware	In Progress	Reaches 76 °C in ~10 min on 10 V supply, requires more thorough testing
20x4 I2C LCD	Hardware	Integrated	Displays weight, temp; 0x27 address
Button Controls	Hardware	In Progress	Implemented and ready for components
PCB Hardware	Hardware	Ordered and on the way	Includes Regulators, Amplifiers, and rectifiers.
Mobile App - Login/Register	Software	Integrated	Active and using API endpoint
Mobile App - Dashboard	Software	Integrated	Active, retrieving data from database
Mobile app and device connectivity	Software	Integrated	Active, uses wifi and HTTP post requests

Plan For Completion

- Push Notifications via API method
- JWT integration for user login session
- Thermal Printer integration
- MOSFET Driver for controlling heating element
- Full firmware code with completed implementation
- Mounting PCB components
- Allergen Free option
- Enclosure Construction
- Further testing

Thank You! - Q & A