

# Memento Bowl

Jacob Dallman  
Jake Statz  
Yovany Lopez Hernandez

14 May 2023

# TABLE OF CONTENTS

<b>1.0 Overview</b>	<b>6</b>
1.1 Executive Summary	6
1.2 Team Contacts and Protocols	6
1.3 Gap Analysis	7
1.4 Proposed Timeline	7
1.5 References and File Links	8
1.6 Revision Table	8
<b>2.0 Impacts and Risks</b>	<b>9</b>
2.1 Design Impact Statement	9
2.2 Risks	10
2.3 References and File Links	11
2.4 Revision Table	11
<b>3.0 Top-Level Architecture</b>	<b>12</b>
3.1 Block Diagram	12
3.2 Block Descriptions	13
3.3 Interface Definitions	14
3.4 References and File Links	15
3.5 Revision Table	15
<b>4.0 Block Validations</b>	<b>16</b>
4.1 Microcontroller	16
4.1.1 Description	16
4.1.2 Design	16
4.1.3 General Validation	17
4.1.4 Interface Validation	18
4.1.5. Verification Process	20
4.1.6 References and File Links	21
4.1.7 Revision Table	21
4.2 PCB	21
4.2.1 Description	21
4.2.2 Design	21
4.2.3 General Validation	22
4.2.4 Interface Validation	22
4.2.5 Verification Process	22
4.2.6 References and File Links	23
4.2.7 Revision Table	23
4.3 Weight Sensors	23
4.3.1 Description	23
4.3.2 Design	24
4.3.3 General Validation	27

4.3.4	Interface Validation.....	27
4.3.5	Verification Process.....	29
4.3.6	References and File Links.....	30
4.3.7	Revision Table.....	30
4.4	WiFi.....	31
4.4.1	Description.....	31
4.4.2	Design.....	31
4.4.3	General Validation.....	31
4.4.4	Interface Validation.....	32
4.4.5	Verification Process.....	33
4.4.6	References and File Links.....	33
4.4.7	Revision Table.....	33
4.5	App.....	34
4.5.1	Description.....	34
4.5.2	Design.....	34
4.5.3	General Validation.....	34
4.5.4	Interface Validation.....	35
4.5.5	Verification Process.....	36
4.5.6	References and File Links.....	36
4.5.7	Revision Table.....	36
4.6	Enclosure.....	37
4.6.1	Description.....	37
4.6.2	Design.....	37
4.6.3	General Validation.....	38
4.6.4	Interface Validation.....	38
4.6.5	Verification Process.....	39
4.6.6	References and File Links.....	40
4.6.7	Revision Table.....	40
<b>5.0</b>	<b>System Verification Evidence.....</b>	<b>41</b>
5.1	Universal Constraints.....	41
5.1.1	The system may not include a breadboard.....	41
5.1.2	The final system must contain a student designed PCB.....	43
5.1.3	All connections to PCBs must use connectors.....	43
5.1.4	All power supplies in the system must be at least 65% efficient.....	44
5.1.5	The system may be no more than 50% built from purchased modules.....	44
5.2	Requirements.....	45
5.2.1	Enclosure.....	45
5.2.1.1	Project Partner Requirement.....	45
5.2.1.2	Engineering Requirement.....	45
5.2.1.3	Verification Process.....	45
5.2.1.4	Testing Evidence.....	45

5.2.2 Item Accuracy.....	46
5.2.2.1 Project Partner Requirement.....	46
5.2.2.2 Engineering Requirement.....	46
5.2.2.3 Verification Process.....	46
5.2.2.4 Testing Evidence.....	47
5.2.3 LED Strip.....	47
5.2.3.1 Project Partner Requirement.....	47
5.2.3.2 Engineering Requirement.....	47
5.2.3.3 Verification Process.....	47
5.2.3.4 Testing Evidence.....	47
5.2.4 Market Research.....	47
5.2.4.1 Project Partner Requirement.....	47
5.2.4.2 Engineering Requirement.....	47
5.2.4.3 Verification Process.....	47
5.2.4.4 Testing Evidence.....	48
5.2.5 Power Supply.....	48
5.2.5.1 Project Partner Requirement.....	48
5.2.5.2 Engineering Requirement.....	48
5.2.5.3 Verification Process.....	48
5.2.5.4 Testing Evidence.....	48
5.2.6 Programmable.....	49
5.2.6.1 Project Partner Requirement.....	49
5.2.6.2 Engineering Requirement.....	49
5.2.6.3 Verification Process.....	49
5.2.6.4 Testing Evidence.....	49
5.2.7 User Interaction.....	49
5.2.7.1 Project Partner Requirement.....	49
5.2.7.2 Engineering Requirement.....	49
5.2.7.3 Verification Process.....	49
5.2.7.4 Testing Evidence.....	50
5.2.8 Wireless Notification.....	50
5.2.8.1 Project Partner Requirement.....	50
5.2.8.2 Engineering Requirement.....	50
5.2.8.3 Verification Process.....	50
5.2.8.4 Testing Evidence.....	50
5.3 References and File Links.....	50
5.4 Revision Table.....	51
<b>6.0 Project Closing.....</b>	<b>52</b>
6.1 Future Recommendations.....	52
6.1.1 Technical Recommendations.....	52
6.1.2 Global Impact Recommendations.....	53

6.1.3 Teamwork Recommendations.....	53
6.2 Project Artifact Summary with Links.....	54
6.3 Presentation Materials.....	54

## 1.0 Overview

### 1.1 Executive Summary

The purpose of this project is to remind people of their personal belongings before leaving the house. The system will be built as a bowl to hold certain items such as phones, wallets, headphones, etc. Multiple items can be placed on the system and can let the user know what exactly is forgotten by memorizing the weight of the item. Communication between the device and user will be done using wifi and an app for the user to download on their phones.

### 1.2 Team Contacts and Protocols

Team Contacts:

Name	Contact	Project Role
Jacob Dallman	dallmanj@oregonstate.edu	Power Supply, Wireless Integration, Web App
Jake Statz	statzj@oregonstate.edu	Enclosure, Weight Sensors, Code
Yovany Hernandez	lopezhey@oregonstate.edu	Microcontroller, PCB, Market Research

Protocols:

Topic	Protocol	Standard
Collaboration and Communication	Team will meet once a week to check in and discuss contributions.	Each team member will maintain a reasonable degree of communication. Expected to meet on time in correct location or notify the group if unable to make it to the proposed meeting.
Task Management	Team will keep track and assign tasks using Microsoft Planner.	During team meetings, the team will review tasks to do and assign tasks for the week. When a task is completed, individuals will mark it as completed.
Conflicts	Team will discuss together on how to approach the conflict	Whether an individual or group is encountering an obstacle, a meeting will take place to discuss how to solve the problem.

### 1.3 Gap Analysis

The doorway effect is a phenomenon associated with human forgetfulness: upon a change in environment, typically two rooms separated by a doorway, one may spontaneously forget the motivation behind the transition in rooms. Acting as a mental refresh, the doorway effect exists within and relies on very short-term memory, most likely as a result of numerous small changes or events that occur during that small timeframe. Consequences of the doorway effect usually bear no significant impact on one's livelihood, however it is an inconvenience nonetheless, and one that this system aims to prevent. The Memento Bowl is marketed as a product that replaces a household trinket bowl with the addition of a pressure sensor and built-in notification system. This system will notify the user after a set distance regarding any items left behind in the bowl in order to avoid running into problems with forgotten belongings.

Our intended market for the Memento Bowl is relatively broad; no one subset of individuals is especially prone or resistant to forgetting their everyday items. Current surveys the team has conducted show no prevalent trend in demographics: over half of the collected responses show at least semi-frequent occurrences of forgetfulness. The most common things left at home include wallets, keys, jewelry, and headphones, all of which are pocket-sized, relatively low in weight, and collectively the most frequent items left in a catch-all near the front door.

### 1.4 Proposed Timeline

## PROJECT TIMELINE

PROJECT TITLE	Product Competition	GROUP	Group 31
PROJECT MEMBERS	Jake Statz, Jaxon Young, Yovany Lopez Hernandez, Jacob Dallman	DATE	6/12/23

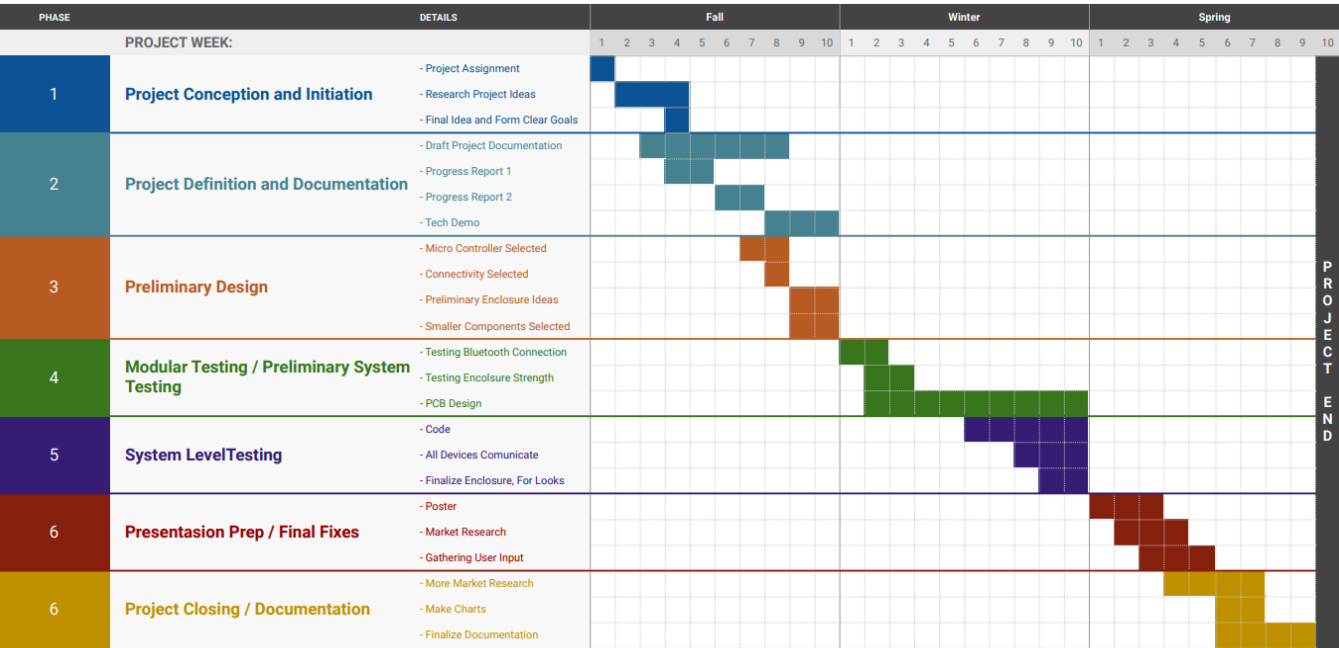


Figure 1

## 1.5 References and File Links

[1] Instructables, “The Don’t-Leave-Home-Without-It Machine,” *Instructables*, Nov. 19, 2014. [Online]. Available: <https://www.instructables.com/The-Dont-Leave-Home-Without-It-Machine/>. [Accessed: Oct. 21, 2022]

[2] T. Stafford, “Why does walking through doorways make us forget?,” *Bbc.com*, 2019. [Online]. Available: <https://www.bbc.com/future/article/20160307-why-does-walking-through-doorways-make-us-forget>. [Accessed: Oct. 21, 2022]

[3] “Unlocking the mysteries of the ‘doorway effect,’” *Scimex*, Mar. 10, 2021. [Online]. Available: <https://www.scimex.org/newsfeed/unlocking-the-mysteries-of-the-doorway-effect>. [Accessed: Oct. 21, 2022]

## 1.6 Revision Table

04/28/2023	Team - Updated roles and project timeline
11/18/2022	Jacob Dallman - updated gap analysis
10/21/2022	Yovany Lopez - Updated executive summary/gap analysis to match project
10/13/2022	Yovany Lopez - Formatted title page, included external references in IEEE format, and included an additional protocol
10/12/2022	Jaxon Young - Initial document creation



## **2.0 Impacts and Risks**

### **2.1 Design Impact Statement**

Engineers are responsible for every aspect of a product's design, intended or not. While the scope of the project may not be fully capable of addressing larger problems, it is useful to identify any shortcomings or more substantial issues that may arise with the creation of this product, regardless of its assistance or hindrance to the design process. The following paragraphs are a few examples of the impacts the project may have.

Although e-waste can be seen as an environmental impact, the build-up of e-waste also impacts public health. Countries that recycle e-waste are overfilled with technological products that affect their surroundings, such as their water supplies as an example [2]. A great safety feature that can be implemented is being able to repair the system fairly easy for anyone to do to reduce the e-waste build up that is affecting many already. A second option is to give resources to those that purchase the product to properly recycle the system.

The system has the potential to help those who have memory problems due to an illness or injury. This can be beneficial to these groups of people to help them with their everyday lives [3]. However, it can be a problem to those who want to purchase the system but cannot afford it. For the system to be accessible to the majority of people, the developers would need to construct the system with parts that are less expensive.

Our product's two most significant environmental impacts are the plastic enclosure and the electronic components housed inside. The enclosure was 3D printed using PLA. Unfortunately, there is a lot of conflicting information when it comes to the biodegradability of PLA, ranging from 20 years to a couple of hundred [7]. Therefore, we have selected three different plastics to make our bowl out of. The first is wood-filled PLA that biodegrades in about 20 years. The other two are both created with corn oil making them far more environmentally friendly than a full synthetic PLA, lasting only 100 years compared to 1000 [8]. Our electronics can all be recycled or reused and create little to no environmental impact besides manufacturing the parts.

Ideally, cost is minimized through eliminating waste and redundancy; while beneficial in the end, much of the budget is then expended on design. Custom PCBs in small quantities carry a much higher per-board cost relative to simpler bulk boards. Experimentation with the enclosure design requires 750g of filament and roughly 90 hours to print. Expansion beyond a web app into iOS requires licensing and more advanced software. Comparatively, IoT devices (e.g. sensors) have been declining in average price as manufacturing gets optimized down to a fraction of initial costs, including the pressure sensors our device is reliant upon [5]. The filament for printing the enclosure can also be sold cheaper in bulk once a design is finalized. However, the lower costs cannot be realized until the system is deemed satisfactory.

ICs are becoming increasingly sought after with rapid growth in tech. E-commerce created a similar insatiable demand for importing/international shipping, and with that comes an increase in low-cost consumer electronics [6]. This can be an issue not only because of an explicit rise in price per chip or increase in delivery times but also implicit reasons tied to the cost of labor or less savory business practices that are taken advantage of to cater to those willing to pay. As such, it is essential we monitor how our funds get allocated in order to design our device while balancing economic and ethical considerations effectively.

## 2.2 Risks

Risk ID	Risk Description	Risk Category	Risk Probability	Risk Impact	Performance Indicator	Action Plan
R1	Potential Fire Starter / board sparks	Technical	M	H	Fire / releasing of secret smoke	Components to not receive max inputs to lower chances of a fire hazard
R2	High Cost	Cost	H	M	Strained budget, compromises on other components	Reassess scope of product, search for cheaper alternatives
R3	Lack of clarity	Technical	L	M	Lack of interest in the product as a result of having to 'learn' the device	Minimize the number of user interfaces
R4	Issues with meeting development goals	Timeline	M	H	Pushing out deadlines, less than expected functionality	Set achievable goals and deadlines w/ group to be on track
R5	E-waste	Cost	M	H	Un recyclable Trash	Avoid bulk buying unnecessary parts
R6	Low Part Supply/Shipping Delays	Logistics	L	H	Unable to assemble the device	Order parts sooner, spend extra for expedited shipping, or alter design to allow for alternatives

## 2.3 References and File Links

- [1] "The Human and Environmental Effects of E-Waste," *PRB*, 2022. [Online]. Available: <https://www.prb.org/resources/the-human-and-environmental-effects-of-e-waste/#:~:text=For%20example%2C%20primary%20and%20secondary,to%20pulmonary%20and%20cardiovascular%20disease>. [Accessed: Nov. 05, 2022]
- [2] "NASD - Reduce Fires with Electrical Safety," *Nasdonline.org*, 2022. [Online]. Available: <https://nasdonline.org/1249/d001053/reduce-fires-with-electrical-safety.html>. [Accessed: Nov. 05, 2022]
- [3] J. Shaw, "The biggest risk & barriers to market entry: Market entry barriers," Kadence, 10-Sep-2021. [Online]. Available: <https://kadence.com/en-us/biggest-risks-and-barriers-to-market-entry/>. [Accessed: 04-Nov-2022]
- [4] <https://sites.psu.edu/edsgn453/3-cultural-issues-that-affect-design/>
- [5] M. Leonard, "Declining price of IOT sensors means greater use in manufacturing," *Supply Chain Dive*, 14-Oct-2019. [Online]. Available: <https://www.supplychaindive.com/news/declining-price-iot-sensors-manufacturing/564980/>. [Accessed: 04-Nov-2022].
- [6] C. Baraniuk, "Why is there a chip shortage?," *BBC News*, 26-Aug-2021. [Online]. Available: <https://www.bbc.com/news/business-58230388>. [Accessed: 04-Nov-2022].
- [7] "3D printlife woodplay wood fill pla 3d printer filament," 3D Printlife WoodPLAy Wood Filled PLA 3D Printer Filament. [Online]. Available: <https://www.3dprintlife.com/wood-play>. [Accessed: 04-Nov-2022].
- [8] D. G. Studio, "How long does PLA take to decompose?," Gianeco. [Online]. Available: <https://www.gianeco.com/en/faq-detail/1/10/how-long-does-pla-take-to-decompose#:~:text=In%20the%20wild%2C%20it%20takes,plastics%20and%20above%20all%20microplastics>. [Accessed: 04-Nov-2022].

## 2.4 Revision Table

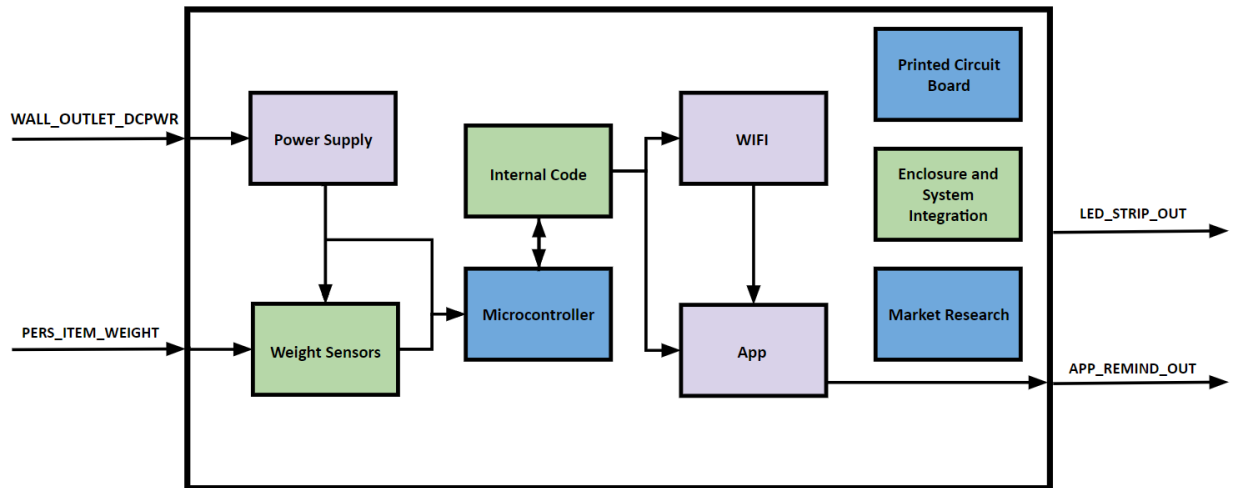
04/28/2023	Team - Included section 2.1 of the Design Impact Assessment and updated 2.3.
11/04/2022	Jacob Dallman - Provided IEEE citations, updated risk table.
11/04/2022	Yovany Lopez - Started 2.0 Impacts and Risks subject.

### 3.0 Top-Level Architecture

#### 3.1 Block Diagram



Figure 2: Black Box Diagram



Jacob Dallman - Purple  
Jake Statz - Green  
Yovany Lopez - Blue

Figure 3: Top Block Diagram

### 3.2 Block Descriptions

Name	Description
<b>Enclosure</b> <b>Champion: Jake Statz</b>	<p>The enclosure will be made out of PLA plastic. We will make a couple of versions, and two of them that I have picked up are a wood-filled one and a marbled stone one. There will be a clear band around their perimeters, so we can include an LED strip.</p>
<b>Weight Sensors</b> <b>Champion: Jake Statz</b>	<p>This block allows us to accurately weigh each item that is placed in the system. This block contains five main components, four load cells, and one analog-to-digital converter board. Load cells typically consist of a strain gauge, which is a resistive material that deforms slightly when a load is applied, and a Wheatstone bridge circuit, which converts the changes in resistance caused by the strain gauge into an electrical signal proportional to the weight being measured. This means the more flex that is applied to the load cells, the more the resistance increases, and in turn, the voltage output increases. The load cells will be mounted to the bottom of the enclosure in a square This analog signal is then amplified and provided to an analog-to-digital converter (ADC), the HX711, for conversion to a digital representation of the weight. The digital representation can then be read by our microcontroller the ESP32</p>
<b>Internal Code</b> <b>Champion: Jake Statz</b>	<p>Programs the microcontroller and Wifi module to run the entire system. Manages the weight sensors and produces notifications.</p>
<b>Power Supply</b> <b>Champion: Jacob Dallman</b>	<p>The power supply will regulate voltage from a wall source and supply 3.3V DC to the microcontroller.</p>
<b>App</b> <b>Champion: Jacob Dallman</b>	<p>The application is responsible for relaying the information received from the weight sensors and microcontroller to the user. The app itself is a web server hosted by the ESP32 and utilizes the respective device's Bluetooth/Wi-Fi capabilities to communicate between the system and the user. When the system detects a significant enough change, the app will notify the user.</p>
<b>WiFi</b> <b>Champion: Jacob Dallman</b>	<p>WiFi is used to communicate between the product and the user. Data, i.e. net change in weight detected by the weight sensors or user information, will be sent via WiFi to a web server hosted on the system's microcontroller.</p>
<b>Microcontroller</b> <b>Champion: Yovany Lopez Hernandez</b>	<p>The ESP32-WROOM-32D microcontroller is used in our system to direct the input data to the app and other components used in the system to let the user know that their personal items have been forgotten when leaving their homes. Such capabilities can be done by using the microcontroller's integrated WIFI and Bluetooth. Digital and analog input/output pins from the microcontroller will be used to connect components of the system to make the final</p>

	product be user-friendly.
<b>PCB</b> <b>Champion: Yovany Lopez Hernandez</b>	A PCB will be created for the system to have a functional microcontroller and the necessary components for the system to work as intended. Other connectors will be installed on the PCB for the LED's, weight sensors, SD card module, and power supply. The shape of the PCB will be circular to be stored inside the enclosure of the system.
<b>Market Research</b> <b>Champion: Yovany Lopez Hernandez</b>	The system will have market research that informs the development of our product from beginning to end with an anonymous survey containing 100+ entries where the survey responses will take into account the draft and final design.

### 3.3 Interface Definitions

<b>Name</b>	<b>Properties</b>
<b>otsd_enclsr_envin</b>	<ul style="list-style-type: none"> <li>● <b>Other:</b> Holds 20lbs</li> <li>● <b>Other:</b> Volume &gt; 800cm<sup>3</sup></li> <li>● <b>Other:</b> Survives 3ft drop</li> </ul>
<b>pwr_sply_wght_snsrs_dcpwr</b>	<ul style="list-style-type: none"> <li>● <b>Inominal:</b> 3.1mA</li> <li>● <b>Ipeak:</b> 3.3mA</li> <li>● <b>Vmax:</b> 3.4V</li> <li>● <b>Vmin:</b> 3V</li> <li>● <b>Vnominal:</b> 3.3V</li> </ul>
<b>pwr_sply_mrcntrllr_dcpwr</b>	<ul style="list-style-type: none"> <li>● <b>Inominal:</b> 50 mA</li> <li>● <b>Ipeak:</b> 1100 mA</li> <li>● <b>Vmax:</b> 3.6 V</li> <li>● <b>Vmin:</b> 3.0 V</li> <li>● <b>Vnominal:</b> 3.3 V</li> </ul>
<b>app_otsd_rf</b>	<ul style="list-style-type: none"> <li>● <b>Messages:</b> Push notification of missing item</li> <li>● <b>Other:</b> App displays active connection</li> <li>● <b>Other:</b> App displays user information</li> </ul>
<b>wght_snsrs_mrcntrllr_dsig</b>	<ul style="list-style-type: none"> <li>● <b>Max Frequency:</b> 10 Samples per Second</li> <li>● <b>Vmax:</b> 3.5V</li> <li>● <b>Vmin:</b> 2.6V</li> </ul>

	<ul style="list-style-type: none"> <li>● <b>Vnominal:</b> 3.3V</li> </ul>
<b>wf_app_data</b>	<ul style="list-style-type: none"> <li>● <b>Messages:</b> App receives user information</li> <li>● <b>Messages:</b> Send wireless notification to device</li> <li>● <b>Other:</b> Connection distance exceeds 5 feet</li> </ul>
<b>intrnl_cd_wf_data</b>	<ul style="list-style-type: none"> <li>● <b>Messages:</b> Transmits data from device to microcontroller</li> <li>● <b>Messages:</b> Transmits data from microcontroller to device</li> <li>● <b>Other:</b> Connection distance exceeds 5 feet</li> </ul>
<b>pcb_enclsr_dcpwr</b>	<ul style="list-style-type: none"> <li>● <b>Inominal:</b> 17.5mA</li> <li>● <b>Ipeak:</b> 40mA</li> <li>● <b>Vmax:</b> 3V</li> <li>● <b>Vmin:</b> 2.4V</li> </ul>

### 3.4 References and File Links

[1] Espressif, “ESP32 Microcontroller”, ESP32-WROOM-32 Datasheet, Aug. 2016 [Revised Feb. 2023].

[2] “loadsensor.pdf,” *Spark Fun*. [Online]. Available: <https://www.sparkfun.com/datasheets/Sensors/loadsensor.pdf>. [Accessed: 11-Feb-2023].

[3] “LMS1585A 5A/LMS1587 5A and 3A Low Dropout Fast Response Regulators Check for Samples: LMS1585A , LMS1587 1FEATURES,” 2003 [Online]. Available: [https://www.ti.com/lit/ds/symlink/lms1587.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-nul-l-ww&ts=1675059325230&ref\\_url=https%253A%252F%252Fwww.ti.com%252Fgeneral%252Fdocs%252Fsuppproductinfo.tsp%253Fdistld%253D10%2526gotoUrl%253Dhttps%253A%252F%252Fwww.ti.com%252Flit%252Fgpn%252Fims1587](https://www.ti.com/lit/ds/symlink/lms1587.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-nul-l-ww&ts=1675059325230&ref_url=https%253A%252F%252Fwww.ti.com%252Fgeneral%252Fdocs%252Fsuppproductinfo.tsp%253Fdistld%253D10%2526gotoUrl%253Dhttps%253A%252F%252Fwww.ti.com%252Flit%252Fgpn%252Fims1587)

### 3.5 Revision Table

05/14/2023	Jacob Dallman - Updated references list.
04/15/2023	Yovany Lopez - Updated PCB block
03/06/2023	Yovany Lopez - Started section 3 and included all information to be reviewed and edited by the team.

## 4.0 Block Validations

### 4.1 Microcontroller

#### 4.1.1 Description

A microcontroller is used in the system called the Memento Bowl. The Memento Bowl is a product that can remind the user of their personal belongings before leaving the house. The system will be built as a bowl to hold certain items such as phones, wallets, headphones, etc. Multiple items can be placed on the system and can let the user know what exactly is forgotten by memorizing the weight of the item. Weight of each item will be saved and retrieved using an SD card module. Communication between the device and user will be done using Wifi and an app to download on devices.

The ESP32-WROOM-32D microcontroller is used in our system to direct the input data to the app and other components used in the system to let the user know that their personal items have been forgotten when leaving their homes. Such capabilities can be done by using the microcontroller's integrated WIFI and Bluetooth. Digital and analog input/output pins from the microcontroller will be used to connect components of the system to make the final product be user-friendly.

#### 4.1.2 Design

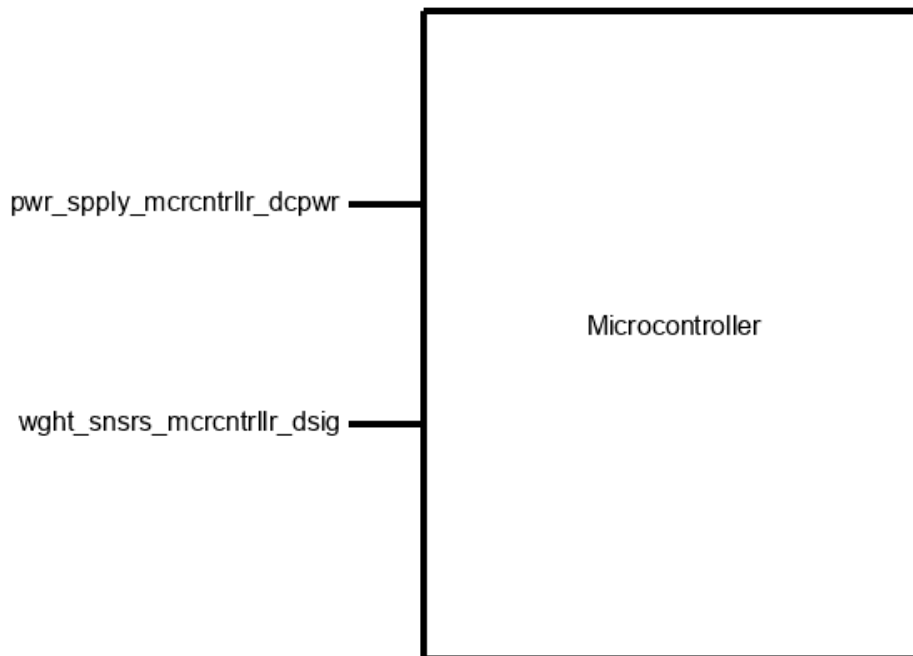


Figure 4



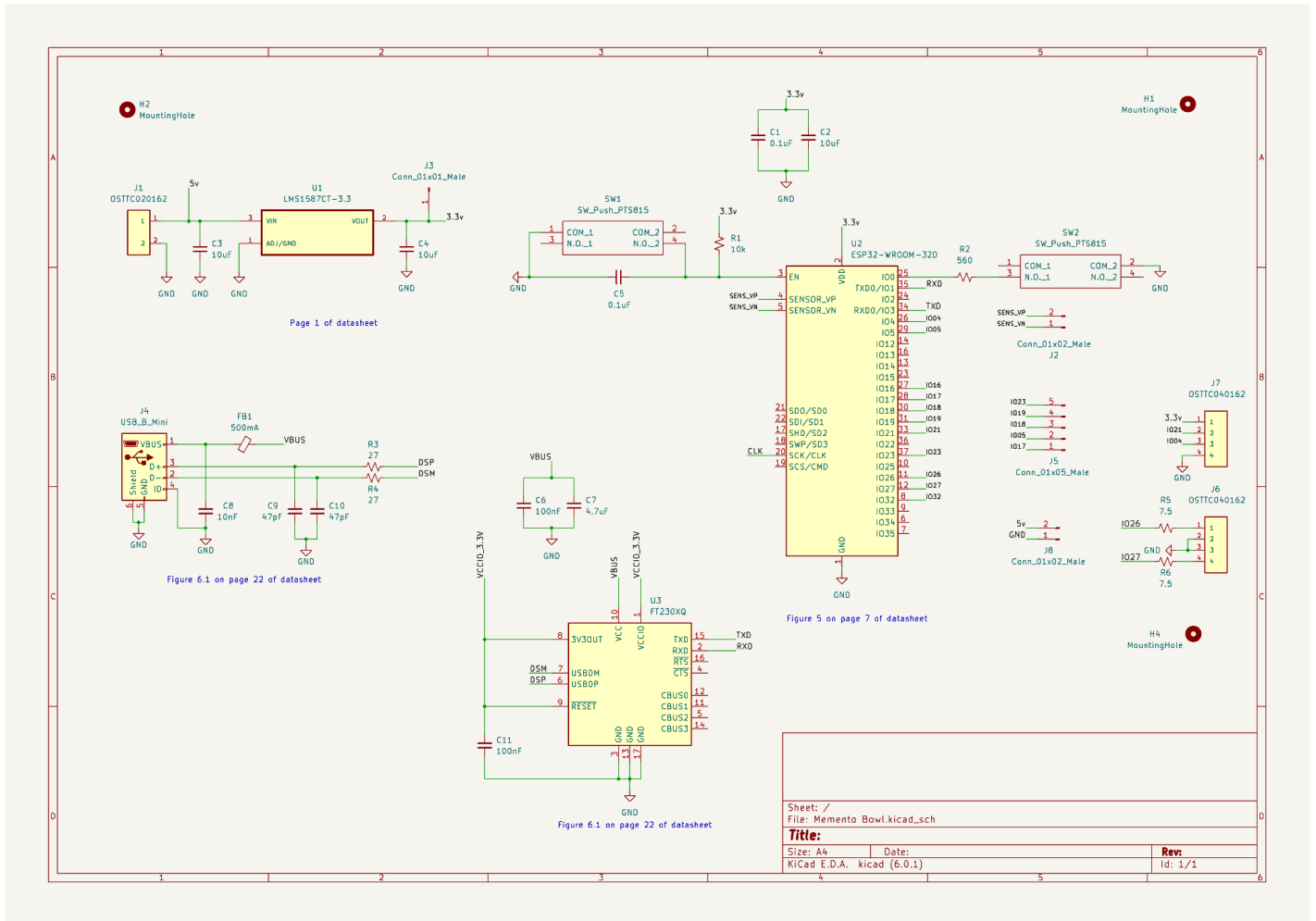


Figure 5

### 4.1.3 General Validation

To direct the weight of the items that are being placed on the weight sensors to the other components used in the system, such as LEDs, connection to a device with a developed app, and an SD card module a microcontroller is needed in the system to do all these capabilities where the ESP32-WROOM-32D microcontroller meets the systems needs to function efficiently.

Another reason behind choosing the ESP32 microcontroller is that two out of three members in the group have experience designing the ESP32 microcontroller. Our group's knowledge makes it less difficult to design and program the microcontroller by using Arduino code which is a language all members of the group know. The price for an ESP32 microcontroller is reasonable with the number of capabilities it can deliver as well.

The integrated Bluetooth from the ESP32 microcontroller will be used to communicate with the app to inform the user that their personal belongings have been forgotten when leaving the house. Most of the components used in the system, including the microcontroller need a power supply of 3.3v which reduces the amount of different power needed for all components. The ESP32 input/output pins are being used to

connect other components the system needs to function as a product that can remind users of forgotten items. Multiple pins are used in the microcontroller, such as the digital pins to receive the weight sensor data it outputs and transfer to the SD card module. The variety of pins the microcontroller has makes it convenient for the SD card module, where it requires a clock with master and slave pins, all of which the microcontroller has. The good amount of input and output pins available allowed for multiple components to connect to the microcontroller with a couple of input-to-output pins to spare. For clarification as well, the weight sensors contain an analog-to-digital converter where the microcontroller will take in the data of the weight sensors as digital input. Overall, the microcontroller will be programmed using Arduino code to take in the weight of the item placed by the user and be transferred to other components to function as a product that can remind users of their personal belongings.

#### 4.1.4 Interface Validation

##### Pwr\_sply\_mrcntrlr\_dcpwr: Input

Interface Property	Why is this interface of this value?	Why do you know that your design details for this block above meet or exceed each property
<p><b>Voltage Nominal: 3.3v</b></p>	<p>The input voltage was chosen based on the voltage range provided by the ESP32 datasheet</p>	<p>For the ESP32-WROOM-32D package:</p> <ul style="list-style-type: none"> <li>● Power supply voltage of 3.0v to 3.6v is needed to power the microcontroller.</li> <li>● Datasheet recommends a typical power supply voltage of 3.3v is efficient for the microcontroller to function</li> </ul>
<p><b>Voltage Max: 3.6v</b></p>	<p>The maximum voltage input the microcontroller can receive before risk of damage</p>	<p>For the ESP32-WROOM-32D package:</p> <ul style="list-style-type: none"> <li>● Power supply voltage of 3.0v to 3.6v is needed to power the microcontroller.</li> <li>● Datasheet recommends a typical power supply voltage of 3.3v is efficient for the microcontroller to function</li> </ul>

<p><b>Voltage Min: 3.0v</b></p>	<p>The minimum input the microcontroller can receive to function which is provided by the ESP32 datasheet</p>	<p>For the ESP32-WROOM-32D package:</p> <ul style="list-style-type: none"> <li>• Power supply voltage of 3.0v to 3.6v is needed to power the microcontroller.</li> <li>• Datasheet recommends a typical power supply voltage of 3.3v is efficient for the microcontroller to function</li> </ul>
<p><b>Current Nominal: 0.5mA</b></p>	<p>The input current for the ESP32 microcontroller was chosen based on the nominal current provided by the component datasheet</p>	<p>For the ESP32-WROOM-32D package:</p> <ul style="list-style-type: none"> <li>• Power supply current recommends a nominal current of 0.5ma for the microcontroller to function</li> </ul>
<p><b>Current Peak: 1100mA</b></p>	<p>The maximum input current the microcontroller can receive before risk of damage</p>	<p>For the ESP32-WROOM-32D package:</p> <ul style="list-style-type: none"> <li>• Power supply current recommends a nominal current of 0.5ma for the microcontroller to function</li> <li>• Datasheet provides the current peak the microcontroller can receive as input power supply before risk of damage</li> </ul>

**Wght\_snsrs\_mrcntrlr\_usrin: Input**

Interface Property	Why is this interface of this value?	Why do you know that your design details for this block above meet or exceed each property
--------------------	--------------------------------------	--

<p><b>Weight of item: 5 pounds or less</b></p>	<p>A weight of 5 pounds or less is expected to be placed on the system as it is the current max weight that can be placed on the weight sensors used for the microcontroller to receive as an accurate input</p>	<p>For the Load Cell Sensor packaging:</p> <ul style="list-style-type: none"> <li>● A max weight of 5 pounds can be placed</li> <li>● The microcontroller can receive an accurate input if the weight of the item is below this value</li> <li>● Weight sensor will output a digital signal that the microcontroller will receive as an input</li> </ul>
--	--	--

#### 4.1.5. Verification Process

The following steps will verify the block meets the interface properties and works as intended for the system to work efficiently. A few of the interface properties will not be able to be performed that rely on other blocks that are not completed yet. Although other tests will be performed that can validate the block does meet the interface properties.

- DC voltage used to power the microcontroller.
  1. Connect two wires to a power supply and use the screw in terminals of the PCB labeled J1 to connect the other ends of the cables.
  2. Turn on the power of the power supply and measure the voltage entering the input power pin of the microcontroller.
  3. The J3 pin of the PCB is a pin designated to measure the voltage and current of the power entering the PCB. The pin is placed after the voltage regulator from 5v to 3.3v and coupling capacitors of the microcontroller. Measure and compare to the intended voltage of 3.3V and current of 0.5mA.
- Weight sensor data input to the microcontroller.
  1. Connect the weight sensor module pins using wires and connect the other ends of the wire on the screw in terminals labeled J6 on the PCB. The screw-in terminals are connected to GND and two digitals to output pins of the microcontroller.
  2. Connect the PCB using the micro-USB B port to a laptop to display the serial monitor of the Arduino software program
  3. Supply power to the PCB using a power supply and test the weight sensors to input data to the microcontroller.
  4. The weight of the data is transferred to the input pins of the microcontroller. Display the weight of the item in the serial monitor of the Arduino software.

#### 4.1.6 References and File Links

[1] “ESP32WROOM32D & ESP32WROOM32U Datasheet” [Online]. Available: [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d\\_esp32-wroom-32u\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf)

[2] “Amazon.com: 4pcs 50kg Half-bridge strain gauge Load Cell Body Scale Weighing Sensor Amplifier+ 1pcs HX711 AD Weight Module Geekstory : Industrial & Scientific,” *Amazon.com*, 2023. [Online]. Available: [https://www.amazon.com/dp/B079FTXR7Y?psc=1&ref=ppx\\_yo2ov\\_dt\\_b\\_product\\_details](https://www.amazon.com/dp/B079FTXR7Y?psc=1&ref=ppx_yo2ov_dt_b_product_details) [Accessed: Jan. 21, 2023]

#### 4.1.7 Revision Table

04/02/2023	Yovany Lopez - Updated verification process
03/12/2023	Yovany Lopez - Completed section 4 of the Microcontroller block
03/06/2023	Yovany Lopez - Started section 4 and included first block validation.

### 4.2 PCB

#### 4.2.1 Description

The PCB is the fundamental component of the system that houses the microcontroller and other essential components to make the system work as intended. The PCB serves as the foundation of the entire system, providing a compact and organized layout for all the required components. The design of the PCB will contain connectors for various components, including LED's, weight sensors, SD card module, and power supply. The PCB will also contain several other components, such as resistors, capacitors, UART chip, and a micro-USB B connector that allows the microcontroller to be programmed. Overall, the PCB serves as the backbone of the system that is required for the system to work as intended.

#### 4.2.2 Design

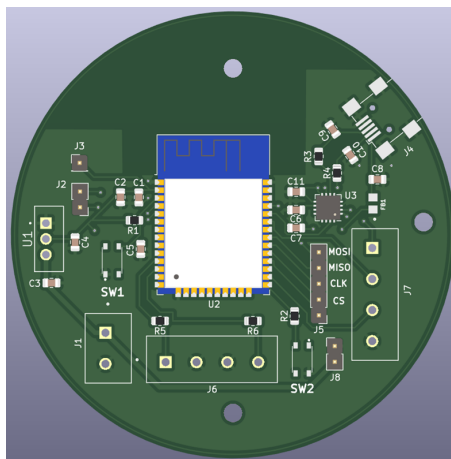


Figure 6

### **4.2.3 General Validation**

The custom PCB created for the project is designed to meet the specific needs of the system. The structure of the PCB was chosen to be circular to fit inside the enclosure of the system due to the enclosure being a bowl shape. An important component that the PCB contains is the microcontroller, which is programmed to control the various components of the system. The microcontroller is responsible for managing its inputs and outputs. The microcontroller is also connected to various components such as the LED's, weight sensors, and SD card module, which are necessary for the system to function correctly. The PCB layout ensures that all these components are connected efficiently.

The custom PCB also includes a power supply circuit that regulates the voltage and current required by the system. This circuit ensures that the system operates within the required specifications and prevents damage to the components.

### **4.2.4 Interface Validation**

The custom PCB does not contain an interface validation as it is a printed circuit board used to connect components together.

### **4.2.5 Verification Process**

The verification process of the PCB requires that the other blocks of the system function correctly with the PCB. Connectors are used to connect the blocks of the system to the PCB and code to program the microcontroller will be required to test the system overall. The following steps will be followed to show that the blocks of the system functions correctly with the PCB.

Weight Sensor Test:

1. Connect the weight sensor module wires to the PCB connectors labeled J7. The first terminal of the top is the 3.3v and the last terminal is the GND terminal. The middle terminals of the connector are the digital connectors.
2. Power the PCB using the power cord and barrel jack being connected to the power connector of the PCB labeled J1.
3. Upload the code to the microcontroller using the Arduino software.
4. Calibrate weight sensors.
5. Display the weight of the items the weight sensors record.

LED Test

1. Connect the LED wires to the PCB connectors labeled J6. The connectors of the first and last terminals are the digital connection to the PCB. The middle terminals of the connector are connected to ground.
2. Power the PCB using the power cord and barrel jack being connected to the power connector of the PCB labeled J1.
3. Upload the code to the microcontroller using the Arduino software.
4. Display the LEDs turning one once the code is uploaded.

## App Test

1. Turn on the microcontroller by supplying power to the PCB using the J1 connector and power cord.
2. Upload code for the microcontroller to communicate with the app downloaded from a device. The code will activate the wifi integration of the microcontroller to be on.
3. Display that the microcontroller and app are connected. Test app by leaving at a certain distance with the device away from the microcontroller to show that app works as intended.

### 4.2.6 References and File Links

[1] “ESP32WROOM32D & ESP32WROOM32U Datasheet” [Online]. Available: [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d\\_esp32-wroom-32u\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf)

[2] “FT230X USB TO BASIC UART IC Future Technology Devices International Ltd. FT230X (USB to BASIC UART IC)” [Online]. Available: [https://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS\\_FT230X.pdf](https://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT230X.pdf)

[3] “LMS1585A 5A/LMS1587 5A and 3A Low Dropout Fast Response Regulators Check for Samples: LMS1585A , LMS1587 1FEATURES,” 2003 [Online]. Available: [https://www.ti.com/lit/ds/symlink/lms1587.pdf?HQS=dis-dk-null-digikeymode-ds-f-pf-null-ww&ts=1675059325230&ref\\_url=https%253A%252F%252Fwww.ti.com%252Fgeneral%252Fdocs%252Fsuppproductinfo.tsp%253FdistId%253D10%2526gotoUrl%253Dhttps%253A%252F%252Fwww.ti.com%252Flit%252Fgpn%252Flms1587](https://www.ti.com/lit/ds/symlink/lms1587.pdf?HQS=dis-dk-null-digikeymode-ds-f-pf-null-ww&ts=1675059325230&ref_url=https%253A%252F%252Fwww.ti.com%252Fgeneral%252Fdocs%252Fsuppproductinfo.tsp%253FdistId%253D10%2526gotoUrl%253Dhttps%253A%252F%252Fwww.ti.com%252Flit%252Fgpn%252Flms1587)

### 4.2.7 Revision Table

04/02/2023	Yovany Lopez - Updated verification process
03/12/2023	Yovany Lopez - Started section 4.2 of the PCB block and is up to date.

## 4.3 Weight Sensors

### 4.3.1 Description

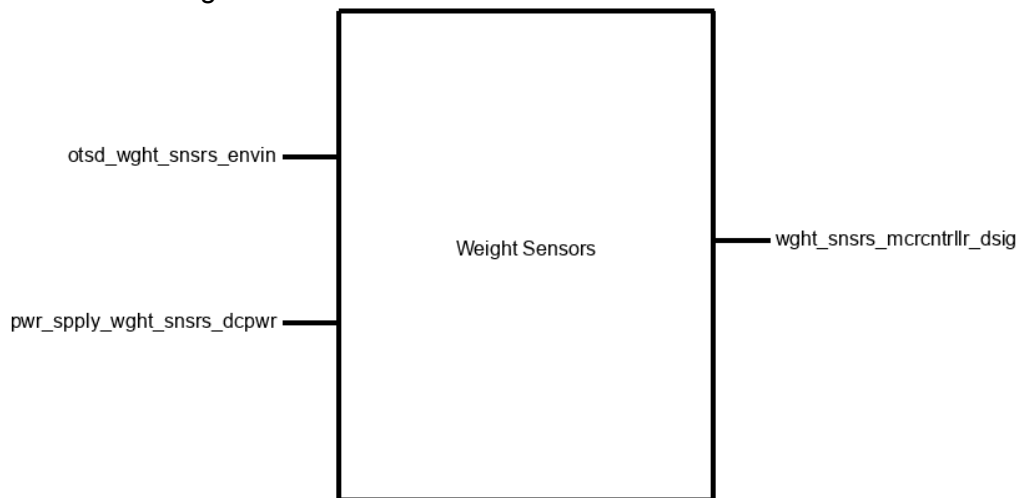
This block allows us to weigh each item in the system accurately. This block contains five main components, four load cells, and one analog-to-digital converter board. Load cells typically consist of a strain gauge, a resistive material that deforms slightly when a load is applied, and a Wheatstone bridge circuit, which converts the changes in resistance caused by the strain gauge into an electrical signal proportional to the weight being measured. This means the more flex applied to the load cells, the more the resistance increases, and the voltage output increases. The load cells will be mounted to the bottom of the enclosure in a square. This analog signal is then amplified and provided to an analog-to-digital converter (ADC), the HX711, for conversion to a digital representation of the weight. The digital representation can then be read by our microcontroller, the ESP32.

### 4.3.2 Design

We will be using the supplied amplifier board instead of recreating the circuit on our main PCB. This will allow for easier repairability and to be more cost-effective. The wiring for the block is relatively simple. The white and black wires are connected between two of the sensors, respectively. Then the red wires are connected to the data lanes of the amp. The amp is connected to the microcontroller with two digital pins, 3.3V, and GND. A wiring diagram is provided in Figure 9 below.

We have three total inputs and outputs. The inputs into the block are DC Power from our power supply and a force applied from outside the system from any objects placed into the system. The output of the block is a digital signal that will then be interpreted by our microcontroller. To demonstrate the use of the weight sensors, I will be using an Arduino and code similar to the following pseudocode:

```
Begin serial communication
Begin reading scale data pins 3 & 13
Set the scale using the calibration number
Tare the scale
Loop
  Get weight
  Print weight
```



**Figure 7**



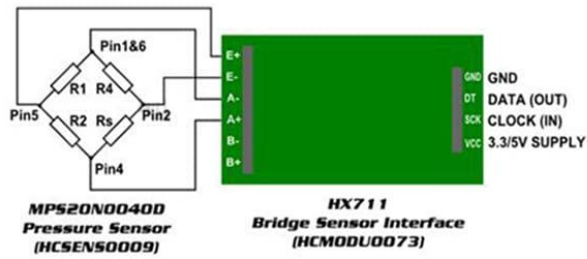
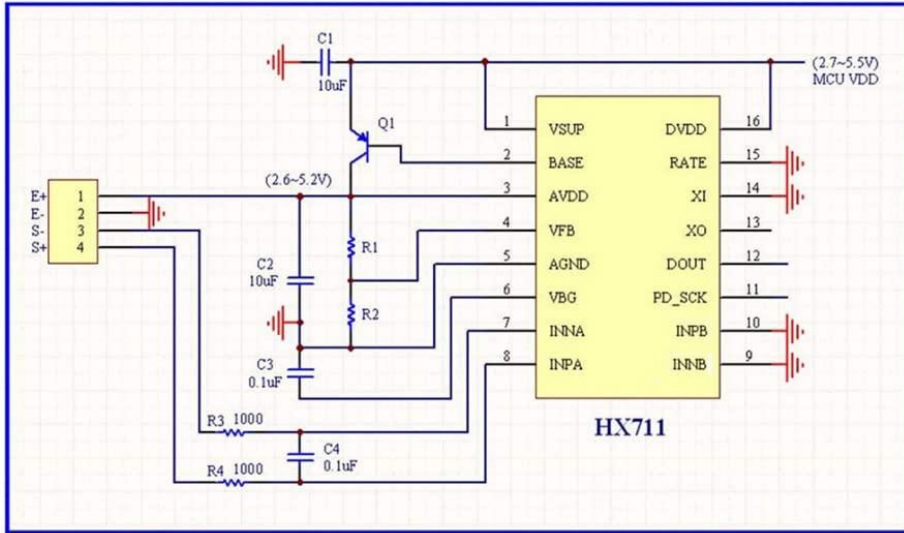


Figure 8

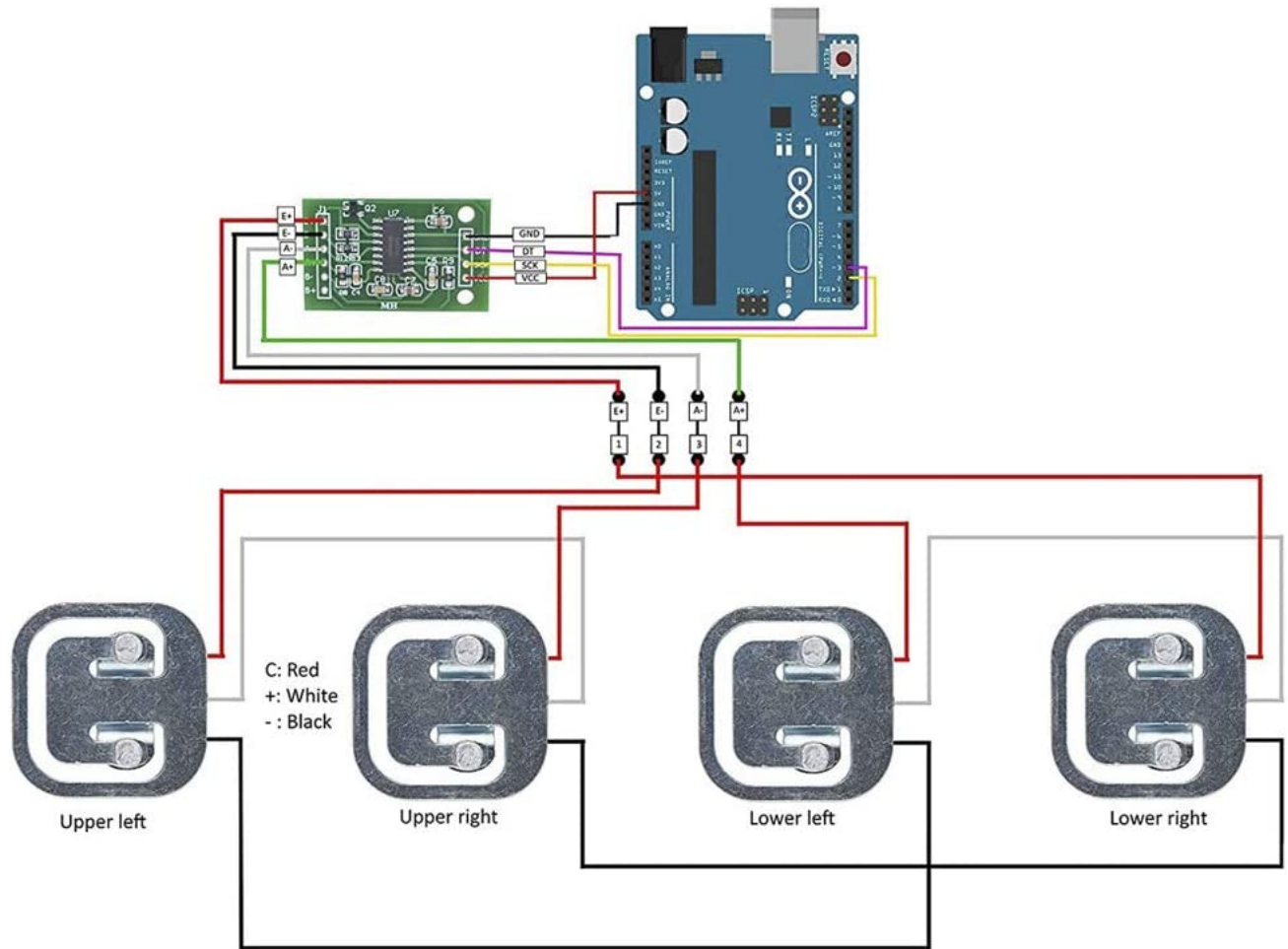


Figure 9

### 4.3.3 General Validation

The generic 50kg load cells are definitely not the best solution for weighing objects accurately, but for our application and constraints, they will work well for our system. The first reason is the size of the sensors. They are quite small, and although they don't have any built-in way to mount to the bottom of our system, the small size makes it easy to include a foundation on the inside of our chassis for the sensors. The second reason is measurement accuracy and precision. There are other force sensors on the market, like force-sensitive resistors, but these resistors are very hard to mount, and after a period of time, they become less sensitive to forces. The third and most important is the availability of the parts. There are a lot of other load cells on the market, but they are oftentimes a lot more expensive and are a lot more complicated because they have been customized for a specific customer or application in mind.

Another good solution that we could use is a strain gauge. A Strain gauge is an aluminum bar that, when flexed, will increase or decrease in resistance. The biggest problem with strain gauges is the amount of room that you need to mount them. With our current shape of a round bowl, it would be impossible to mount a strain gauge in the current enclosure.

### 4.3.4 Interface Validation

#### wght\_snsrs\_mrcntrllr\_dsig

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Rate: 10 Samples per Second	The rate is determined based on if the rate pin on the HX711 is shorted to ground. When the pin is not shorted, the rate is increased to 80 SPS. The increased rate would increase the accuracy of data, but this requires more power and processing power.	For the HX711 in a SEN-13879 package: <ul style="list-style-type: none"> <li>• Clock Frequency: 11.0592MHz</li> <li>• 10SPS when shorted to ground</li> <li>• 80SPS when set to high</li> </ul>
Vnominal: 3.3V	This voltage was selected because of the voltage limit of the ESP32. The operating voltage of the ESP32 is 3.3V, so anything too far over this would damage the microcontroller. The digital signal is dependent on the DC input voltage that is applied to VCC.	For the HX711 in a SEN-13879 package: <ul style="list-style-type: none"> <li>• Regulator Supply: 2.7V ~ 5.5V</li> <li>• Analog Supply: 2.6V ~ 5.5V</li> <li>• Digital Supply: 2.6V ~ 5.5V</li> </ul>
Vmax: 5.5V	This voltage was selected because the HX711	For the HX711 in a SEN-13879 package:

	analog-to-digital converter, is set to a maximum of 5.5V. This voltage will never be reached because The ESP32 is unable to reach an output voltage of over 3.3V. The digital signal is dependent on the DC input voltage that is applied to VCC.	<ul style="list-style-type: none"> <li>● Regulator Supply: 2.7V ~ 5.5V</li> <li>● Analog Supply: 2.6V ~ 5.5V</li> <li>● Digital Supply: 2.6V ~ 5.5V</li> </ul>
Vmin: 2.6V	This voltage was selected because of the voltage limit of the ESP32. If the voltage drops below 2.6V the HX711 would not have enough power to convert data, so It is impossible that the digital signal would be under 2.6V. The digital signal is dependent on the DC input voltage that is applied to VCC.	For the HX711 in a SEN-13879 package: <ul style="list-style-type: none"> <li>● Regulator Supply: 2.7V ~ 5.5V</li> <li>● Analog Supply: 2.6V ~ 5.5V</li> <li>● Digital Supply: 2.6V ~ 5.5V</li> </ul>

[1] [HX711 Datasheet](#), Accessed 2/11/2023

**pwr\_spply\_wght\_snsrs\_dcpwr**

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Inominal: 3.1mA	This value was measured during testing. At 3.3V, the current will be well under the peak of 3.3mA. Depending on the temperature and the humidity of the air, this value could change because the resistances of the metal load cells will change.	For the HX711 in a SEN-13879 package: <ul style="list-style-type: none"> <li>● Current consumption including on-chip analog power supply regulator: normal operation &lt; 1.5mA for the amp, power down &lt; 1uA</li> </ul>
Ipeak: 3.3mA	This current peak was selected because this is the most the HX711 and surrounding circuits will consume.	For the HX711 in a SEN-13879 package: <ul style="list-style-type: none"> <li>● Current consumption including on-chip analog power supply regulator: normal operation &lt; 1.5mA</li> </ul>
Vmax: 5.5V	This voltage was selected because the HX711 analog-to-digital converter is set to a maximum of 5.5V. This voltage will never be reached because The ESP32 is unable to	For the HX711 in a SEN-13879 package: <ul style="list-style-type: none"> <li>● Regulator Supply: 2.7V ~ 5.5V</li> <li>● Analog Supply: 2.6V ~ 5.5V</li> <li>● Digital Supply: 2.6V ~ 5.5V</li> </ul>

	reach an output voltage of over 3.3V. The digital signal is dependent on the DC input voltage that is applied to VCC	
Vmin: 2.6V	This voltage was selected because of the voltage limit of the ESP32. If the voltage drops below 2.6V the HX711 would not have enough power to convert data, so It is impossible that the digital signal would be under 2.6V. The digital signal is dependent on the DC input voltage that is applied to VCC.	For the HX711 in a SEN-13879 package: <ul style="list-style-type: none"> <li>● Regulator Supply: 2.7V ~ 5.5V</li> <li>● Analog Supply: 2.6V ~ 5.5V</li> <li>● Digital Supply: 2.6V ~ 5.5V</li> </ul>
Vnominal: 3.3V	This voltage was selected because of the voltage limit of the ESP32. The operating voltage of the ESP32 is 3.3V, so anything too far over this would damage the microcontroller. The digital signal is dependent on the DC input voltage that is applied to VCC.	For the HX711 in a SEN-13879 package: <ul style="list-style-type: none"> <li>● Regulator Supply: 2.7V ~ 5.5V</li> <li>● Analog Supply: 2.6V ~ 5.5V</li> <li>● Digital Supply: 2.6V ~ 5.5V</li> </ul>

[1] [HX711 Datasheet](#), Accessed 2/11/2023

#### 4.3.5 Verification Process

All Interfaces will have to undergo the following steps to be verified:

1. Connect the positive power supply lead to VCC on the amplifier board.
2. Connect the negative power supply lead to GND on the amplifier board.
3. Connect the Dsig of the amplifier to Arduino nano pin D3.
4. Connect the SCLK pin of the amplifier to the Arduino pin D13.
5. Connect the lower right load cell to pin E+ on the amplifier board
6. Connect the upper left load cell to pin E- on the amplifier board
7. Connect the upper right load cell to pin A+ on the amplifier board
8. Connect the lower left load cell to pin A- on the amplifier board
9. Plug the Arduino into a laptop using a micro USB cable.
10. Run the calibration program on Arduino IDE.
11. Input the calibration offset into the test program.
12. Run the test program and open the serial monitor.

#### **wght\_snsrs\_mcrctrlr\_dsigt**

1. The serial monitor will be outputting 10 data points a second proving the rate of 10 samples a second. (The option to cut the rate pin to increase the rate to 80 samples a second should also be talked about).
2. Using the voltage readout on the power supply and a multimeter across Dsig and GND, the min, max, and nominal, voltage can be proven, and the fact the data in the serial monitor will show that the chip has not been fried.

#### **pwr\_sply\_wght\_snsrs\_dcpwr**

1. Using the voltage readout on the power supply and a multimeter across VCC and GND, the min, max, and nominal, voltage can be proven, and the fact the data in the serial monitor will show that the chip has not been fried.
2. The readout on the power supply should also say the current for the selected voltage, cycle through Vmin and Vmax to show Inominal and Ipeak.

#### 4.3.6 References and File Links

[1] Mouser, "24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales," *Mouser*. [https://www.mouser.com/datasheet/2/813/hx711\\_english-1022875.pdf](https://www.mouser.com/datasheet/2/813/hx711_english-1022875.pdf) (accessed Jan. 20, 2023).

[2] Geekstory, "4pcs 50kg Half-bridge strain gauge Load Cell Body Scale Weighing Sensor Amplifier+ 1pcs HX711 AD Weight Module Geekstory," *Amazon*. [https://www.amazon.com/dp/B079FTXR7Y?psc=1&ref=ppx\\_yo2ov\\_dt\\_b\\_product\\_details](https://www.amazon.com/dp/B079FTXR7Y?psc=1&ref=ppx_yo2ov_dt_b_product_details) (accessed Jan. 20, 2023).

[3] "loadsensor.pdf," *Spark Fun*. [Online]. Available: <https://www.sparkfun.com/datasheets/Sensors/loadsensor.pdf>. [Accessed: 11-Feb-2023].

- [HX711 Datasheet](#)
- [SEN-10245 Datasheet](#)

#### 4.3.7 Revision Table

2/10/2023	Jake Statz - General validation complete
2/9/2023	Jake Statz - Interface plan and validation updated
2/8/2023	Jake Statz - Description and design updated
1/20/2023	Jake Statz - First Draft

## 4.4 WiFi

### 4.4.1 Description

WiFi is used to communicate between the product and the user. Data, i.e. net change in weight detected by the weight sensors, will be sent via WiFi to an app on an Android smart device. The ESP32 is capable of communication over WiFi as well as Bluetooth, with integrated capability for both, and will thus be responsible for managing communication between the system and user. WiFi will be responsible for updating data on the force sensors (including what items will have been determined to be missing) and prompting the app for any form of notification for the user, however, the computation will be left to the ESP32 itself and the other internal code loaded onto the microcontroller. In addition, WiFi will be able to manage the transmission of data that is stored onto the microcontroller for user reference and have it registered in the app.

### 4.4.2 Design

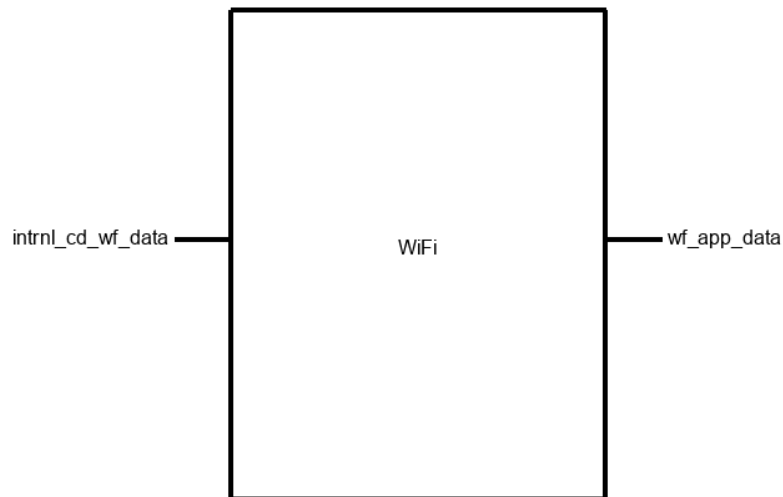


Figure 10

### 4.4.3 General Validation

The wireless component of internal code that will be hosted on our microcontroller has been compartmentalized into its own block, rather than being included with any of the computational code. The reasoning behind the design choice is mainly in order to modularize all of our task-specific processes. Larger blocks within our system, e.g. the PCB and the internal code have a number of subsections that vary both in complexity and in overarching function. Despite technically being code hosted on the microcontroller, with protocols designed around the microcontroller being able to communicate with other devices, its sole purpose is to enable communication between multiple devices and nothing else. With this in mind, we separated Bluetooth from the other code and established its base parameters of having data in from the microcontroller, and data out from the microcontroller to be received by a smart device.

WiFi can also be used to update the information stored on the microcontroller. In this case, information from the force sensors would be loaded from the microcontroller into either system memory or a micro SD card to hold reference points for any logged items in the system. This information can then be sent to the device, providing a log or

array of the registered items stored in a JSON file. This may also open the door for the ability to alter the stored information at either end (microcontroller or smart device), however, this relies more upon the app's structure and internal code that will be running alongside the web server.

#### 4.4.4 Interface Validation

##### wf\_app\_data : Output

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Messages: Wireless notification is sent to the device.	One of the Engineering Requirements is that the user must receive a wireless notification from the system on their device.	For the ESP32-WROOM-32D: <ul style="list-style-type: none"> <li>WiFi connection range: 20 meters (typical).</li> </ul>
Messages: App receives transmitted information.	The user must be able to view their stored data accurately; a wireless connection is required to do this from their device.	For the ESP32-WROOM-32D: <ul style="list-style-type: none"> <li>WiFi connection range: 20 meters (typical).</li> </ul>
Other: Connection distance exceeds 5 feet.	Connection distance is based on proximity to the device within a household. Wireless communication should be possible from a position that is not within reaching distance, after accounting for any obstacles between devices.	For the ESP32-WROOM-32D: <ul style="list-style-type: none"> <li>WiFi connection range: 20 meters (typical).</li> </ul>

##### intrnl\_cd\_wf\_data : Input

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Messages: Transmits data from microcontroller to device.	Communication between the microcontroller and user device is bidirectional. The ESP32 must be able to relay data to the user.	For the ESP32-WROOM-32D: <ul style="list-style-type: none"> <li>WiFi connection range: 20 meters (typical).</li> </ul>



Messages: Transmits data from device to microcontroller.	Communication between the microcontroller and user device is bidirectional. Changes to the data from the user must be received and processed by the ESP32.	For the ESP32-WROOM-32D: <ul style="list-style-type: none"> <li>WiFi connection range: 20 meters (typical).</li> </ul>
Other: Connection distance exceeds 5 feet.	Connection distance is based on proximity to the device within a household. Wireless communication should be possible from a position that is not within reaching distance, after accounting for any obstacles between devices.	For the ESP32-WROOM-32D: <ul style="list-style-type: none"> <li>WiFi connection range: 20 meters (typical).</li> </ul>

#### 4.4.5. Verification Process

Verification can be done through establishing a connection between two devices and being able to prove connectivity as well as the ability to transmit data. Additionally, since the WiFi functionality is built into the ESP32 directly, testing can be done without the final product itself, and can instead be proven through the use of a separate ESP32 dev module. This does not require change in code, nor does it require additional hardware (outside of the separate microcontroller). The code must be written and flashed to the ESP32 prior to testing, which will include:

- A message written to the console that confirms that WiFi has properly been initialized.
- A message written to the console that confirms the ESP32 has successfully been paired with the device.
- A method of sending data from the ESP32 to the phone or vice versa; this can include a photoresistor, an LED, printing a single number or character, etc. This is to verify that the connection is stable. For this instance, the method will be using a Terminal app that already exists for Arduino devices, downloaded onto the smart device being used to pair with the ESP32.
- Repeated testing at further distances to gauge the effective radius of the ESP32.

#### 4.4.6 References and File Links

[1] "ESP32WROOM32D & ESP32WROOM32U Datasheet" [Online]. Available: [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d\\_esp32-wroom-32u\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf)

#### 4.4.7 Revision Table

05/14/2023	Jacob Dallman - updated process, general validation
03/06/2023	Jacob Dallman - Creation of section, update to project document

## 4.5 App

### 4.5.1 Description

The application is responsible for relaying the information received from the weight sensors and microcontroller to the user. The app itself utilizes the respective device's Bluetooth/Wi-Fi capabilities to communicate between the system and the user. When the system detects a significant enough change, the app will notify the user.

### 4.5.2 Design

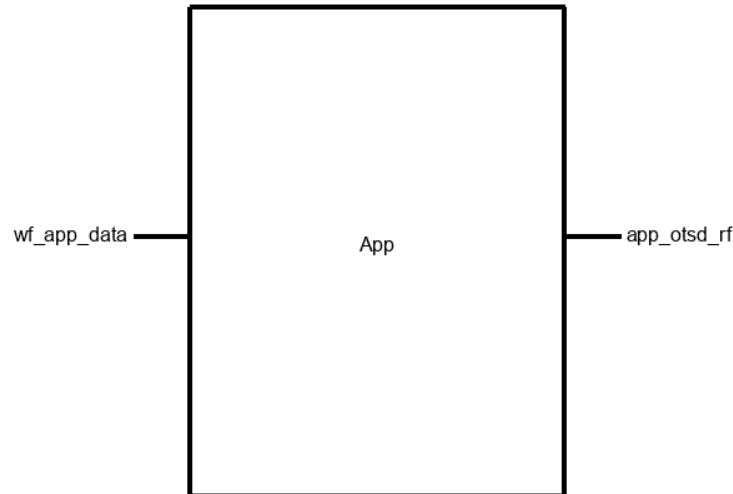


Figure 11

### 4.5.3 General Validation

The app has gone through a number of iterations in order to find the friendliest interface both for implementing functionality and for general usability. For our purposes, stylistic choices are not weighed as heavily, meaning CSS and UX come second to prove that the system works as intended. The current application and underlying system runs through accessing the local IP of the ESP32, which is running a web server. Hosting the server on the ESP32 directly allows for much easier access and manipulation of the stored data, alongside familiarity with the IDE and less obscurity amongst layers when working with more complex designs. Other ideas included using Visual Studio and Android Studio to create a standalone application running on the Android platform, providing better security and ease of use at the cost of running additional protocols to get the same functionality.

The use of a web server allows us to easily store values in a file and upload it in a table for organization of registered items and weights. Another benefit of using the ESP32 as a web application is that program memory can all be stored on the ESP32 via SPIFFS, which is able to partition a certain amount of memory to manage the server as well as all of the stored data. A microSD module was included on the PCB in the event extra memory was required, although this has given us a workaround. Because this system is designed to alert the user, this means the ESP32 is also responsible for processing the notification request. Using a web server removes the ability to send a push notification to a device directly; therefore, the most effective way to process an alert

is through an app that has push notification ability as well as the ability to monitor active connections on the same WiFi network.

#### 4.5.4 Interface Validation

##### wf\_app\_data : Input

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Messages: Wireless notification is sent to the device.	One of the Engineering Requirements is that the user must receive a wireless notification from the system on their device.	For the ESP32-WROOM-32D: <ul style="list-style-type: none"> <li>WiFi connection range: 20 meters (typical).</li> </ul>
Messages: App receives transmitted information.	The user must be able to accurately view their stored data; a wireless connection is required to do this from their device.	For the ESP32-WROOM-32D: <ul style="list-style-type: none"> <li>WiFi connection range: 20 meters (typical).</li> </ul>
Other: Connection distance exceeds 5 feet.	Connection distance is based on proximity to the device within a household. Wireless communication should be possible from a position that is not within reaching distance, after accounting for any obstacles between devices.	For the ESP32-WROOM-32D: <ul style="list-style-type: none"> <li>WiFi connection range: 20 meters (typical).</li> </ul>

##### app\_otsd\_rf : Output

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Messages: Push notification of missing item.	The system is intended to register and remember items placed in it by the user. If any of those items are left behind, the user is intended to be alerted.	From the Engineering Requirements: The system will send a notification to the user's mobile device identifying the item(s) left behind with 90% accuracy.

Other: App displays user information.	The application is responsible for providing an interface to store and/or modify user data. The user must be provided a method to accurately manage their information.	From the Engineering Requirements: The system will record and store the weights of at least 3 individual items registered by the user with 90% accuracy.
Other: App displays active connection.	A wireless connection must be established and active in order for the user to be able to properly access and edit their data.	The efficacy of Bluetooth is a binary measurement, i.e. whether or not the connection is present determines if data is transmitted. For the ESP32-WROOM-32D: <ul style="list-style-type: none"> <li>• Bluetooth LE connection range: 100 meters (maximum).</li> </ul>

#### 4.5.5. Verification Process

Using a web app requires both devices running on the same network. For demonstration purposes, the network credentials will be hard coded into the microcontroller that is hosting the application, although functionality may be provided for a login portal.

1. Connect the system to power.
2. Enter the local IP address of the device that is hosting the application on a device connected to the same network.
3. A successful connection will load the web page on the device.
4. Enter a name for the new item that will be registered in the system.
5. Place the item in the bowl and use the 'Save' button on the page to retrieve the weight.

#### 4.5.6 References and File Links

[1] "ESP32WROOM32D & ESP32WROOM32U Datasheet" [Online]. Available: [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d\\_esp32-wroom-32u\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf)

#### 4.5.7 Revision Table

05/14/2023	Jacob Dallman - Updated block parameters, verification process
03/06/2023	Jacob Dallman - Creation of section, update to project document

## 4.6 Enclosure

### 4.6.1 Description

The enclosure of our system consists of three main parts. The first part is the bottom of the bowl. The bottom of the bowl houses the weight sensors, PCB, and DC power connector. The second part is the LED ring. The LED ring is illuminated by two “noodles”. Depending on the material the rest of the enclosure is made out of, we can change the color of the LEDs to better match the aesthetic. The last part of the bowl is the top. The top of the bowl will hold all of the user's items and contain threaded inserts, so the top and bottom can be attached using 3mm screws.

### 4.6.2 Design

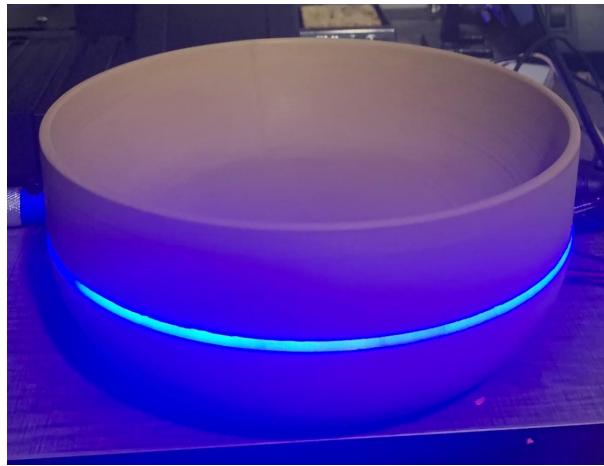


Figure 12

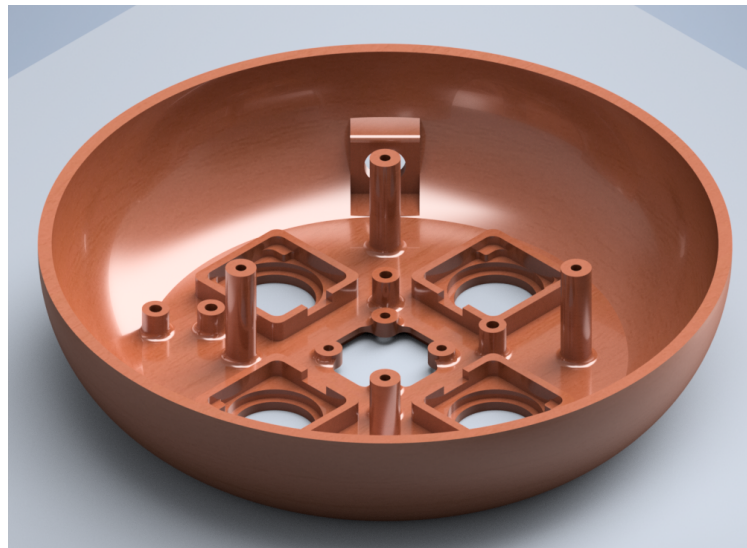


Figure 13

The bottom of the bowl is the most important part of the enclosure. Containing a total of 9 threaded inserts, five 3mm, screws, and 4 2.5mm screws. The hardest part of the bottom to design was the housing for the weight sensors and how they would be

mounted solidly enough. The caps of the weight sensors are super glued because they would not be enough room to allow screws.

#### 4.6.3 General Validation

The design of the bowl allows for all of the different components to be mounted using screws, ensuring that nothing is moving around inside of the bowl and getting unplugged. Since this is not a production model of the Momento Bowl, it is important to allow for fast disassembly, and using threaded inserts allows us to reduce wear on the plastic, so repeated disassembly without having to worry about longevity is possible.

#### 4.6.4 Interface Validation

##### otsd\_enclsr\_envin

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Holds 20lbs	We chose 20lbs as our target weight not because we ever expect a user to place 20lbs of weight into the bowl, but because there will be a repeated use of 1-5 pounds in the bowl at all times.	The enclosure was modeled with a strong base and thick walls that will be enough to support the 20lbs without cracking.
Volume > 800cm <sup>3</sup>	This is the interface for the volume of the bowl because the 3D printer that we used to print the bowl could not reasonably print a bowl with a longer diameter. The height of the bowl could have been increased, but not without sacrificing the aesthetics of the design.	Using basic geometry, you could calculate the volume of the bowl. What we did was measure how much fluid oz of water the bowl could hold.
Survives 3ft drop	3ft was selected for a drop test because 36 inches is slightly taller than a standard table height.	We know that the impact from a 3ft drop should not be able to split the two halves of the bowl apart.

##### pcb\_enclsr\_dcpwr

Interface Property	Why is this interface this value?	Why do you know that your design details for
--------------------	-----------------------------------	--

		<b>this block above meet or exceed each property?</b>
Inominal: 17.5mA	The LEDs require 17.5mA to power on and adding more current does not affect the brightness of the LEDs.	This information was provided in a video on the store page [1]
Ipeak: 40mA	The maximum current that is used by the LEDs is 40mA. This maximum is hit when the voltage is 3V.	This information was provided in a video on the store page [1]
Vmax: 3V	These LEDs are specced at 3V. It is possible to over volt them for more brightness, but they become hot and can burn out after a couple of days.	This information was provided in a video on the store page [1]
Vmin: 2.4V	The voltage of 2.4V was chosen because we would like to be able to fade the brightness of the LEDs and 2.4V is the lowest voltage that light is emitted.	After testing we found that the lowest voltage that the LEDs can receive with still being visible is 2.4V.

#### 4.6.5 Verification Process

##### Otsd\_enclsr\_envin

1. Place 30lb dumbbell into the bowl and read the value using the serial monitor
2. Fill the bowl with 800 milliliters of water
3. Measure the height of a table, making sure that it is over 3ft tall
4. Slide the bowl off of the table to simulate a real-life scenario of how the bowl could be dropped.

##### Pcb\_enclsr\_dcpwr

1. Power on the ESP32 to power the LEDs
2. Break the circuit and use a multimeter to measure the current draw at nominal.
3. Disconnect the LED from the ESP32 and attach it the power supply at 3V
4. Break the circuit and measure the current using a multimeter
5. Finally, turn the power supply down to 2.4V and make sure the LEDs are still visible

#### 4.6.6 References and File Links

[1] "Adafruit nOOds store page [Online]. Available: <https://www.adafruit.com/product/5508>

#### 4.6.7 Revision Table

3/11/2023	Jake Statz - Section Created
3/13/2023	Jake Statz - More information added to each section
3/09/2023	Jake Statz - Section Completed



## 5.0 System Verification Evidence

### 5.1 Universal Constraints

#### 5.1.1 The system may not include a breadboard

The Memento Bowl consists of a custom PCB that houses the microcontroller and other essential components used to make the system work as intended. Figure 13 displays an image of the PCB used in our system. Figure 14 displays the schematic of the PCB containing the names of the components being installed.

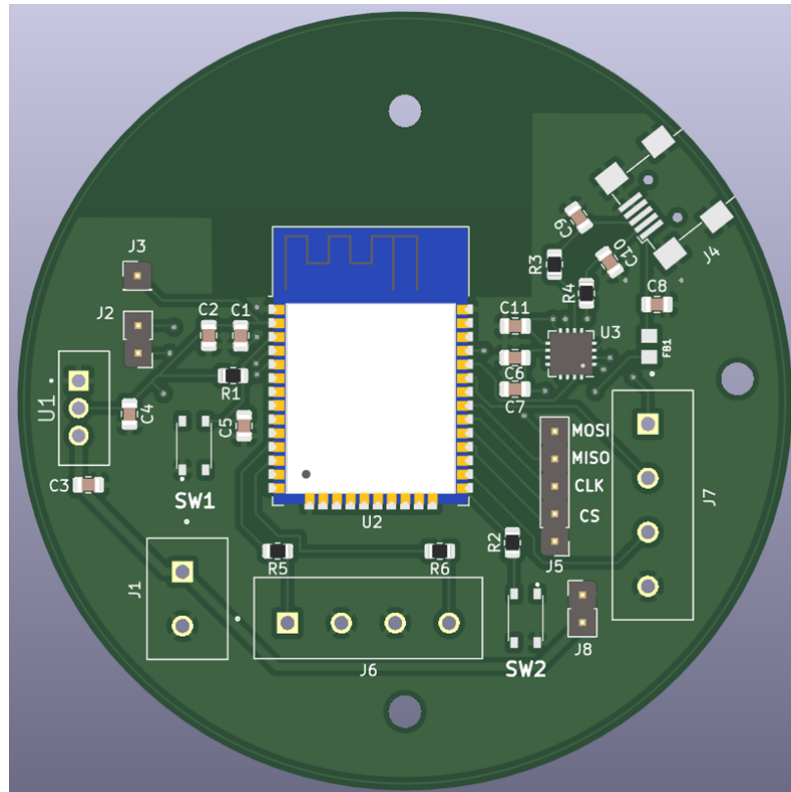


Figure 14

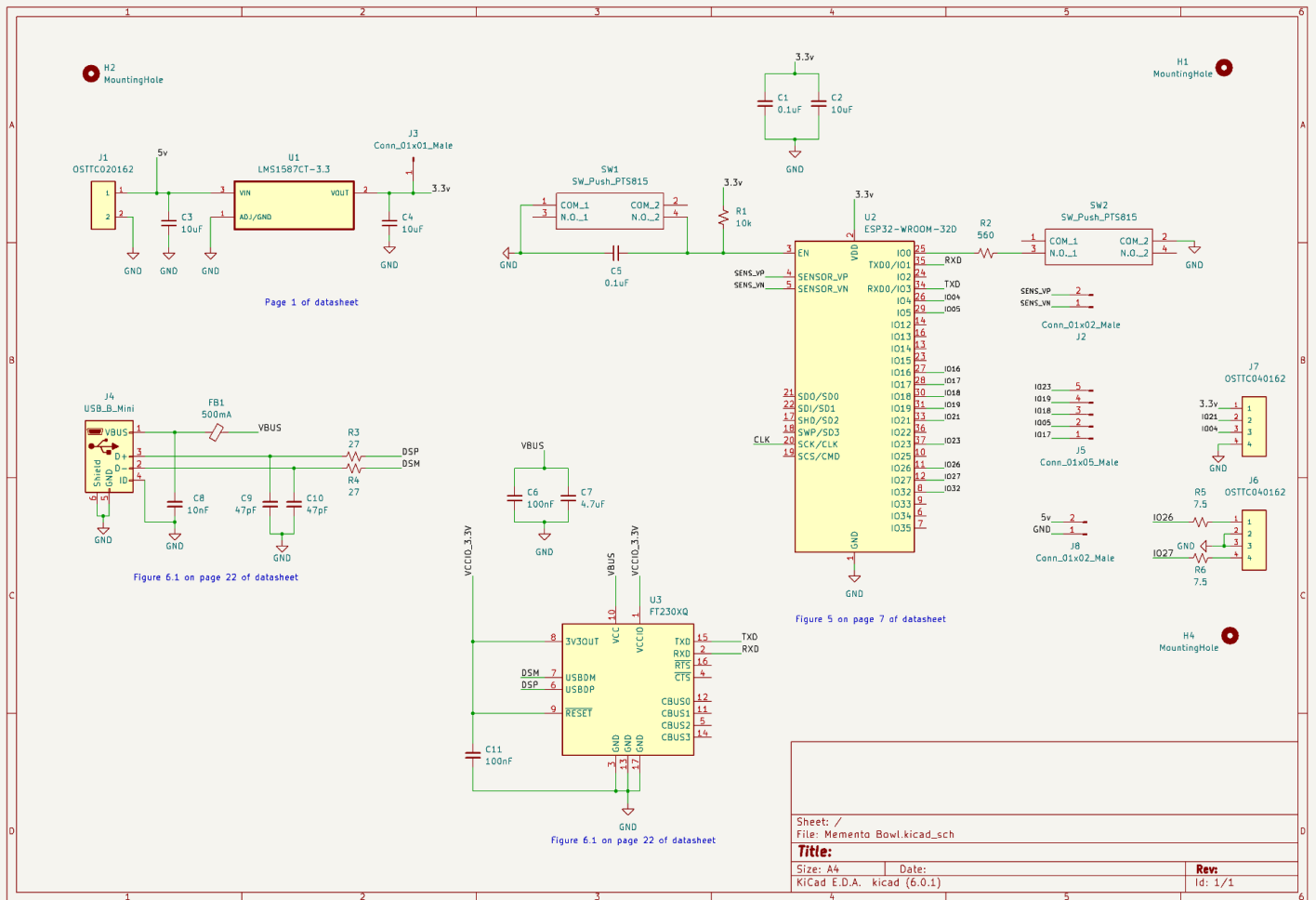


Figure 15



Figure 16

### 5.1.2 The final system must contain a student designed PCB.

As mentioned in section 5.1.1, a custom PCB was created for our system. This was possible by following the datasheet of the components. Figure 13 contains the image of the custom PCB created by the team. A total of 141 pads are being used in the PCB and the number of pads was determined by the Kicad software used to create the PCB.

### 5.1.3 All connections to PCBs must use connectors.

Figure 12 displays an image of the PCB. The PCB contains male header pins and screw-in terminals. J2, J3, J5, and J8 are the male header pins. J1, J6, and J7 are the screw-in terminals. All connections of the custom PCB are using only connectors. Figure 15 displays an image of the connectors used in the system.

#### **5.1.4 All power supplies in the system must be at least 65% efficient.**

The power supply in our system takes in a wall outlet and steps down 5V DC to 3.3V through a linear voltage regulator. Calculated efficiency of the power supply using the figures below yields an estimated 65.37%, which can be improved with a switching regulator.

$$\begin{aligned} V_{in} &= 5.028V, I_{in} = 123.1mA \\ V_{out} &= 3.287V, I_{out} = 123.1mA \end{aligned}$$

#### **5.1.5 The system may be no more than 50% built from purchased modules.**

The system is created by both purchased modules and custom-made parts from the team. The purchased modules include the weight sensors and analog-to-digital converter. Custom parts created from the team include custom PCB, enclosure, and App. Based on the figures provided below, the estimate for the percentage of the system built by purchased modules is **35.38%** at most.

- **Microcontroller - 100%**

Purchased Modules: ESP32

Designed Modules: N/A

- **Printed Circuit Board - 0%**

Purchased Modules: N/A

Designed Modules: Custom PCB (includes circuits, connectors, etc.)

- **Power Supply - 50%**

Purchased Modules: AC wall adapter

Designed Modules: 3.3V step-down circuit

- **WiFi - 50%**

Purchased Modules: ESP32

Designed Modules: Web server

- **App - 33%**

Purchased Modules: IFTTT app interface

Designed Modules: Notification protocol, web server

- **Weight Sensors - 100%**

Purchased Modules: Weight sensors

Designed Modules: N/A

- **Internal Code - 0%**

Purchased Modules: N/A

Designed Modules: Code

- **Enclosure - 0%**

Purchased Modules: N/A

Designed Modules: Enclosure base, LED shroud, bowl

## **5.2 Requirements**

### **5.2.1 Enclosure**

#### **5.2.1.1 Project Partner Requirement**

A minimum viable product.

#### **5.2.1.2 Engineering Requirement**

The system will function normally with a weight of 5 lbs of items.

#### **5.2.1.3 Verification Process**

- 1) Power the microcontroller by connecting the power cable to the PCB connector labeled J1. This powers the microcontroller to upload the code necessary to accurately read the weight of the item.
- 2) Upload the code to the microcontroller to see the weight of the items using the Arduino software serial monitor.
- 3) Place the item on the weight scale and read the recorded weight on the serial monitor.

#### **5.2.1.4 Testing Evidence**

Below are images of the recorded weights of two laptops. The recorded weight measured by the weight sensors of the system measures about 7.6 lbs, which is above the 5 lbs needed for our engineering requirement. Additional testing evidence can be found at this [link](#).

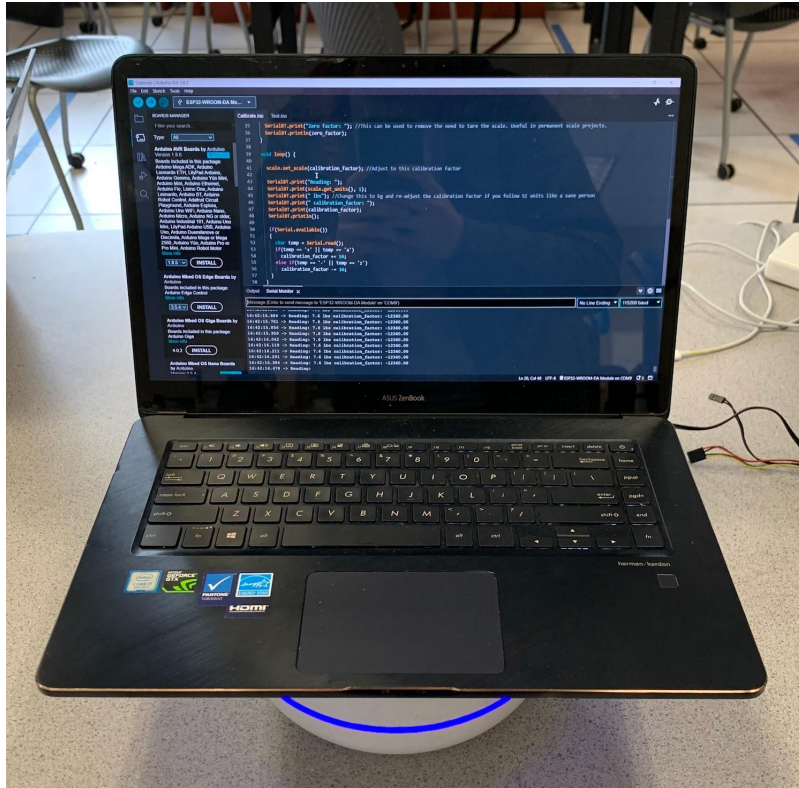


Figure 17

## 5.2.2 Item Accuracy

### 5.2.2.1 Project Partner Requirement

The items need to be detected.

### 5.2.2.2 Engineering Requirement

The system will indicate the correct contents of the bowl 90% of the time when an item is removed.

### 5.2.2.3 Verification Process

For this testing process, it will be assumed that all items are registered in the bowl prior to verification. Accuracy is dependent on the difference in item weights relative to each other and can vary based on the degree of accuracy desired.

- 1) Place two items (that are already registered in the system) into the bowl.
- 2) Wait 5-10 seconds to ensure that items have been detected by the bowl.
- 3) Remove one of the items from the bowl. Leave the other.
- 4) Disconnect the device from the WiFi network leaving the bowl powered on. Wait for the notification.
- 5) Verify that the notification corresponds to the item that is in the bowl.

#### **5.2.2.4 Testing Evidence**

Testing evidence can be found at this [link](#).

### **5.2.3 LED Strip**

#### **5.2.3.1 Project Partner Requirement**

A minimum viable product.

#### **5.2.3.2 Engineering Requirement**

The system will indicate when a new item has been recorded and 9 out of 10 users will state the indication was clear.

#### **5.2.3.3 Verification Process**

The LED strip is responsible for providing visual cues pertaining to the system, including power status and the recording of items. For this process, ten participants unrelated to the design process will be used to observe the system behavior.

- 1) Power on the system. Verify the LED strip turns on.
- 2) Place an item in the bowl.
- 3) Observe the behavior of the LED strip.

#### **5.2.3.4 Testing Evidence**

Testing evidence can be found at this [link](#).

### **5.2.4 Market Research**

#### **5.2.4.1 Project Partner Requirement**

Market research and how it informed your product development.

#### **5.2.4.2 Engineering Requirement**

The system will have market research that spans the development of our product from beginning to end, with an anonymous survey containing 100+ entries. The survey responses will be taken into account in the draft and final design.

#### **5.2.4.3 Verification Process**

A document containing a detailed description on the survey responses will be used as reference to summarize the market research block of the system.

- 1) Display on a laptop the anonymous survey and the number of responses collected.
- 2) Open the document for the market research block containing a detailed description of the data collected from the anonymous survey.
- 3) Summarize how questions collected from the survey influenced certain aspects of the design.

#### 5.2.4.4 Testing Evidence

Evidence:

[https://docs.google.com/document/d/1WbXrCvpl7DHzn0ivTB\\_sP0AOrMcSzBK-TcPh\\_8toHeo/edit?usp=sharing](https://docs.google.com/document/d/1WbXrCvpl7DHzn0ivTB_sP0AOrMcSzBK-TcPh_8toHeo/edit?usp=sharing)

#### 5.2.5 Power Supply

##### 5.2.5.1 Project Partner Requirement

Product will be supplied power through an outlet.

##### 5.2.5.2 Engineering Requirement

The system will be provided with power from a US wall outlet.

##### 5.2.5.3 Verification Process

- 1) Connect the power cable to the US wall outlet.
- 2) Connect the other end of the power cable to the barrel jack of the enclosure, powering the PCB stored inside the enclosure.
- 3) The LED ring of the enclosure will light up to show that the system is being powered through the US wall outlet.

##### 5.2.5.4 Testing Evidence

Below is an image of the system with the LED ring lit up using the power cable connected to the US wall outlet. Additional testing evidence can be found at this [link](#).



Figure 18



## **5.2.6 Programmable**

### **5.2.6.1 Project Partner Requirement**

A minimum viable product.

### **5.2.6.2 Engineering Requirement**

The system will record and store the weights of at least 3 individual items registered by the user with 90% accuracy.

### **5.2.6.3 Verification Process**

Comparison of weights recorded by the system requires an additional source to provide a reference. The system will provide the weight measured by the bowl's sensors and stored into the system directly.

- 1) Power on the system and connect the bowl to WiFi.
- 2) Connect to the web app through any device on the same network. Cellular data is not required for this test.
- 3) Verify web connection through the server and enter a name for the item.
- 4) Place the item in the bowl and save. Repeat for a total of three items.
- 5) To verify the information has been stored, reboot the system by disconnecting and reconnecting the power adapter.
- 6) Refresh the web app on the user's device. All saved items should display the same values recorded in step 4.
- 7) Compare the values of each item to their respective measurements using a separate scale. This isn't mandatory, however, as everything in the bowl is working from the same reference point.

### **5.2.6.4 Testing Evidence**

Testing evidence can be found at this [link](#).

## **5.2.7 User Interaction**

### **5.2.7.1 Project Partner Requirement**

A minimum viable product.

### **5.2.7.2 Engineering Requirement**

The system will have the 9 out of 10 users calibrate the system for a specific object and receive notifications in under 5 minutes with only written instructions.

### **5.2.7.3 Verification Process**

A written instruction guide will be provided to ten participants uninvolved with the design of the system. This will contain a proper explanation of setup, standard usage procedures, and expected results. The guidelines of the test for each participant will be as follows.

- 1) Have the participant select an item they may carry on themselves throughout the day.
- 2) Provide the participant with the bowl and instructions. Set a timer with a proper explanation that it is solely for benchmarking the clarity of the instructions.
- 3) Once the participant receives a notification that an item has been left in the bowl, stop the timer.
- 4) Record time and take note of any difficulties the participant found.

#### **5.2.7.4 Testing Evidence**

Testing evidence can be found at this [link](#).

### **5.2.8 Wireless Notification**

#### **5.2.8.1 Project Partner Requirement**

User needs to be notified.

#### **5.2.8.2 Engineering Requirement**

The system will send a notification to the user's mobile device identifying the item(s) left behind with 90% accuracy.

#### **5.2.8.3 Verification Process**

Verifying the notification functionality of the system requires a device connected via WiFi and installation of the IFTTT app. Upon discovering that the device is no longer connected to the same WiFi network, a notification will be sent to the user's device with a reminder of the item left in the bowl.

- 1) Power on both the device and the bowl, and ensure that both are connected to the same WiFi network.
- 2) Place an item in the bowl. Ensure it has been detected.
- 3) Disconnect the device from WiFi. Leave the bowl powered on.
- 4) Wait for a notification to appear on the device.
- 5) Verify that the notification contains the appropriate item left in the bowl.

#### **5.2.8.4 Testing Evidence**

Testing evidence can be found at this [link](#).

### **5.3 References and File Links**

[1] "ECE44x: The Project Document Content Guide," *Google.com*, 2023. [Online]. Available: [https://docs.google.com/document/u/1/d/e/2PACX-1vSTGN-MsnjkEHcNywmy30H1PFOFKHb0VqDk4epHT8vFdkYmJU\\_G73ZmjhTbj4UtabMG7anF15-OSYx/pub](https://docs.google.com/document/u/1/d/e/2PACX-1vSTGN-MsnjkEHcNywmy30H1PFOFKHb0VqDk4epHT8vFdkYmJU_G73ZmjhTbj4UtabMG7anF15-OSYx/pub). [Accessed: Mar. 13, 2023]

[2] "ECE44x Universal Project Constraints," *Google.com*, 2023. [Online]. Available: <https://docs.google.com/document/u/1/d/e/2PACX-1vQhQWdXiNOFAbNN7tIT66qDIJLO0us8UJxr1OiAi-t9UL6gZzki0n06HkyOv6LxJPA6u4PyR73LzUn4/pub>. [Accessed: Mar. 13, 2023]

#### 5.4 Revision Table

05/14/2023	Team updated universal constraints
05/12/2023	Yovany Lopez - Updated Market Research document
05/10/2023	Team updated requirements of section 5
05/07/2023	Team updated section 5 of the document
03/12/2023	Team created section 5 of the document

## **6.0 Project Closing**

### **6.1 Future Recommendations**

#### **6.1.1 Technical Recommendations**

The system was able to function properly with our ideas and work that the team has implemented. The following are four technical recommendations that can be continued for a future team to take on.

Our first recommendation is to have the system be powered by batteries instead of a wall outlet [1]. The system is limited to locations in which the system can be placed in homes as it needs power through a wall outlet to function. By replacing the power of the system with batteries, the system can be placed anywhere.

The LEDs of the system turn on when the product is powered on by the wall outlet. A second feature of the LEDs blinks when the system has an item placed on the bowl to let the user know that the item's weight is being recorded. These features of the LEDs are basic and a good starting point for the system to have. The system could have the potential to have different colors to have different meanings of what the system is doing. An example can be having the LEDs light up blue when being powered off and red when recording the weight of an item [2].

The enclosure has a couple of design flaws. The first one is that the mounting system between the top and the bottom half should be more robust. Currently, there are four posts passing through the bottom of the bowl, allowing you to screw into the top half. The posts are very fragile and break easily. This is partly because the enclosure is 3D printed, and the layer lines create a weak point for the posts to shear off. This can be solved by beefing up the thickness of the posts or using a different method of manufacturing, such as injection molding or CNC milling. A lot of material is used to create one bowl. The methods mentioned above would solve this problem and allow for thinner walls and the ability to use a less dense material. [3]

Our final recommendation would be to turn the Web app that we have created into a full mobile application that can be easily downloaded. We also envision that there will be a version that is compatible with modern cars' touch display to allow you to receive notifications in your car, and not just your phone. [4]

[1] "PCB Mount Battery Holder 3 Cell AA Batteries - Arcade Parts and Repair," *Arcade Parts and Repair*, Jan. 25, 2023. [Online]. Available: <https://www.arcadepartsandrepair.com/store/components/battery-holders/pcb-mount-battery-holder-3-cell-aa-batteries/> [Accessed: Apr. 29, 2023]

[2] "Amazon.com: ALITOVE 3.2ft 60 Pixels WS2812B Individual Addressable RGB LED Strip Light," Amazon.com, 2023. [Online]. Available: [https://www.amazon.com/ALITOVE-Individual-Addressable-Programmable-Non-Waterproof/dp/B01MG49QKD/ref=sr\\_1\\_7?keywords=programmable%2Bled%2Bstrips&qid=1682735639&sr=8-7&th=1](https://www.amazon.com/ALITOVE-Individual-Addressable-Programmable-Non-Waterproof/dp/B01MG49QKD/ref=sr_1_7?keywords=programmable%2Bled%2Bstrips&qid=1682735639&sr=8-7&th=1). [Accessed: Apr. 29, 2023]

[3] Xometry, “*Manufacturing on Demand*”, Xometry. [Online]. Available: <https://www.xometry.com/quoting/home/> [Accessed: 28-Apr-2023].

[4] Android, “*Android for Cars overview*”, Android. [Online]. Available: <https://developer.android.com/training/cars> [Accessed: 28-Apr-2023].

### 6.1.2 Global Impact Recommendations

Future improvements of the system should also include changes that aim to lessen its footprint. One of the more direct alterations is potentially increasing the efficiency of the power supply much higher than the currently measured 65%. Switching the linear regulator for a switching regulator increases the power efficiency closer to 90% at the cost of additional components in the power supply circuit, most often seen in the form of a DC-DC buck converter [1].

Conceptually beyond any direct changes to the schematic is the ability to provide some level of repairability to the system, rather than encouraging the user to dispose of an entire product. E-waste is a growing concern with the expansion of technology in the consumer market and providing the user with proper documentation, product information, and resources encourages the user to consider maintenance over replacement [2]. In turn, landfills are spared unnecessary deposits of FR4 and plastic filament.

[1] Henry Zhang Download PDF, “140: Basic concepts of linear regulator and switching mode power supplies,” *Analog Devices*. [Online]. Available: <https://www.analog.com/en/app-notes/an-140.html>. [Accessed: 28-Apr-2023].

[2] J. Myers, “Right to repair: The first step towards tackling e-waste,” *Arm Blueprint*, 25-Nov-2021. [Online]. Available: <https://www.arm.com/blogs/blueprint/right-to-repair>. [Accessed: 28-Apr-2023].

### 6.1.3 Teamwork Recommendations

After the first term of our project, one of our members dropped the course without warning the rest of the group. While we were ultimately able to get the work divided, this would have been nice to know beforehand so we could properly scale the project to accommodate one less contributor. A solution to this problem could be having frequent check ups between members through a text message group chat to know as soon as possible [1]. An appropriate setting would be in a group meeting to let everyone in the team know in person. If a person is in the role of dropping a class, it is best to let the team know as soon as possible.

A team should have recurring meetings where every team member can meet on a specific day and time. It should be a top priority to figure out a day and time for the team to meet at the beginning of each term, as schedules can vary for each member. This can be beneficial overall as deadlines and goals made by the team can be met on time [2]. It is recommended to have meetings once every week in the fall term. A meeting once every week will be beneficial to the team as each member will understand one another and slowly build team relationships. For the next two terms, meetings can be done twice or three times a week as the workload can increase, and progress can be seen only by giving enough time for each member to work on their side of the project.

[1] Skills, "Perspectives on Communicating with Engineering Teams," 2023, <https://www.pluralsight.com/blog/teams/perspectives-on-communicating-with-engineers> [Accessed: 28-Apr-2023].


[2] L. Keel, "The Symphony Agency," The Symphony Agency, Aug. 19, 2020. [Online]. Available: <https://symphonyagency.com/how-to-find-a-meeting-time-that-works-for-everyone/> [Accessed: 28-Apr-2023].

## 6.2 Project Artifact Summary with Links

- [BOM](#)
- [PCB Schematic](#)
- [PCB Layout](#)
- [Market Research survey](#)
- [Market Research Document](#)
- [Poster](#)


## 6.3 Presentation Materials

COLLEGE OF ENGINEERING
Electrical Engineering and Computer Science
ECE 31



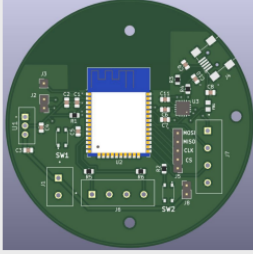
### Engineering Requirements

- **Enclosure** - The system will operate normally under 5 lbs of stress.
- **Item Accuracy** - 90% of registered items will be correctly recognized by the bowl.
- **LED Strip** - An LED strip will provide visual cues during operation.
- **Market Research** - Surveys pertaining to design and functionality of the bowl will be conducted to assess market demand. A report on the data collection will be drafted and the final product will contain evidence of market input.
- **Power Supply** - The system will be powered using a standard US wall outlet.
- **User Interaction** - Clear instructions will be provided to set up the system.
- **Programmable** - Users will be able to register at least 3 personal items through the web server.
- **Wireless Notification** - The system will notify the user of any belonging(s) left in the bowl.



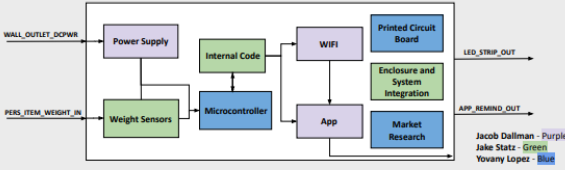
## Memento Bowl

Trinket Bowl With Weight Recognition and Wireless Notifications



### System Overview

- System is controlled by a custom microcontroller using an ESP32-WROOM-32D module.
- Power from a wall outlet is reduced to 3.3V to run the system.
- Weight sensors are stored in the enclosure base to record the weight of items placed in the bowl.
- Using the on-board SPIFFS, a partition of the flash memory, the weight of all of items is able to be saved even when power is cycled.
- LEDs are installed on the enclosure to display that the items have been recorded and stored in the system.



### Market Research

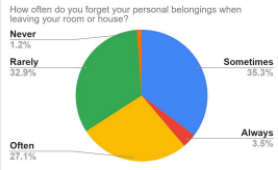
- Two surveys were conducted to determine what the general public wanted from the system. The first survey asked questions on what type of items will be used with the system to determine enclosure shape.
- The second survey asked specific questions about any potential features the system will have.
- Answers to the anonymous questions were used to influence the design and features of the system as a whole.
- The graph on the right breaks down survey responses about personal forgetfulness. Roughly two thirds of respondents report it as relatively common.

How often do you forget your personal belongings when leaving your room or house?

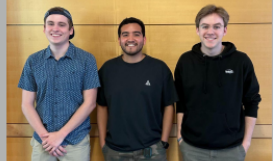
Never 1.2%

Rarely 32.9%

Often 27.1%



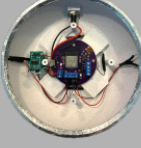
### Team Members




**Jake Statz** (Left) - Enclosure, Code, Weight Sensors  
**Yovany Lopez Hernandez** (Middle) - PCB, Market Research, Microcontroller  
**Jacob Dallman** (Right) - Web App, Power Supply, WiFi Integration

### What is the Memento Bowl?

The Memento Bowl was created as a solution to forgetting personal items at home. Tasked with building a minimum viable product, our team sought out to design a device with market potential.





Project Info

Figure 19

Project Showcase: [Memento Bowl](#)