SELF-CLEANING LOCKER

By

Chilo Llamas (cllamas2)

Immanuel Fernando (ijf2)

Nithin Nathan (nnatha3)

Final Report for ECE 445, Senior Design, Spring 2012

TA: Yichi Zhang

4 May 2021

Project No. 65

Abstract

We designed and built a locker that disinfects itself when the locker door is closed, and no objects are detected inside the locker. The locker disinfects itself using a sprayer that sprays Hydrogen Peroxide, and then uses fans attached to the sides of the locker to dry the interior. An LCD display attached to the front of the locker allows users to see if the locker is clean, in the process of cleaning, not clean, or has a low disinfectant supply.

Additionally, we designed an Android application that can visually show the relative amount of disinfectant supply left in the locker.

This project was successful, as it was able to provide the functionality as stated above.

Contents

1 Introduction

1.1 Objective

Sanitation means everything, especially in a pandemic. We must make sure that ourselves and our belongings are clean in order to help mitigate the spread of COVID. Especially with more companies, buildings, restaurants, and gyms opening up, it's becoming easier for the disease to spread. Particularly with gyms, people are constantly sharing lockers with others, which can easily spread germs due to the storage of their sweaty belongings and clothes.

To combat the spread of COVID, and germs in general, we are proposing a self-cleaning locker. When the locker detects that nothing is inside, it will automatically disinfect the inside of the locker using disinfectant sprays. An LED display on the outside of the locker door will display the status of whether or not it is cleaned, and if there is still disinfectant in the locker.

In addition, we will be building an app that will monitor the status of the locker. For gym owners, the app will allow them to keep track of all their self-cleaning lockers in their locker rooms and make sure that every locker is properly maintained for the safety of their employees and individuals using the gym's lockers.

1.2 Background

We are trying to solve the issue of germs spreading between uses at gym lockers. The gym locker is the main area to hold someone's belongings while using the gym, and everyone throws their belongings in there. However, no one knows who has used that particular locker before them, and if that person has been in contact with other people who have had COVID. Since the locker room is a shared space, an individual really has no choice where else to put their belongings, and whether or not the locker they choose is COVID, or germ free in general.

The gym is one of the easiest places where bacteria can spread. Many different parts of the building and locker room have a multitude of germs. For example, the gym faucet handle has 545,312 CFU (colony forming units), which has eight times as many bacteria than a school cafeteria water fountain spigot. Gym benches have 8,241 CFU, which has six times more bacteria than an animal cage [1]. Contact with these objects can easily lead to the spread of germs. In addition, research shows COVID can last up to two days on fabric, and even up to nine days on certain surfaces [2], [3].

This is where our project comes in. The main problem we are trying to solve is allowing users to have that ease of mind by not worrying about who has used the locker before them, and whether or not it's clean to put their belongings in by eliminating bacteria that could be spread from surface to clothing, and ultimately an individual.

1.3 Physical Design

Our initial project sketches can be found in Appendix A. Figures 1, 2, and 3 shown below are pictures of our final project design as shown in demonstration.

Figure 1: Front Exterior of Locker

Figure 2: Interior of Locker

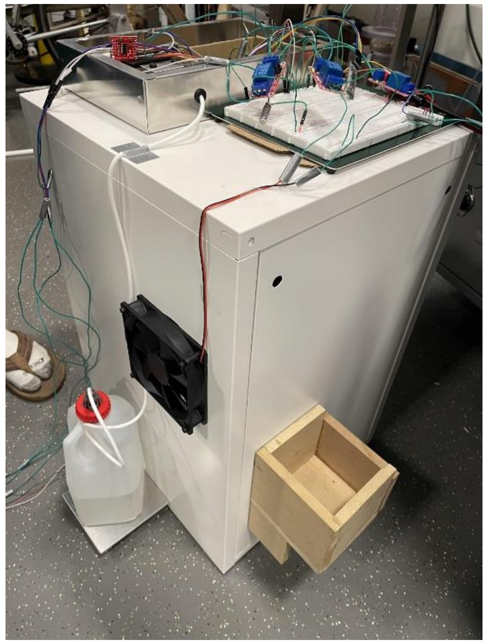

Figure 3: Back Exterior of Locker

1.4 High Level Requirements

- Weight sensors correctly detect when items weighing at least 250 grams are on top of them with an error range of roughly 5%.
- Spray correctly cleans the inside of the locker when it is empty and closed, covering at around 90% +/- 5% of the interior surface area.
- Project correctly detects different disinfectant supply levels with an error range of around 5% at each level.

2 Design

The three main subsystem units we will be using for our project will consist of a control unit, disinfectant unit, and monitoring unit. Our final schematic and PCB layout can be found in Appendix B and C. Additionally, a picture of our physical PCB can be found in Appendix D. We will go into each unit into further detail.

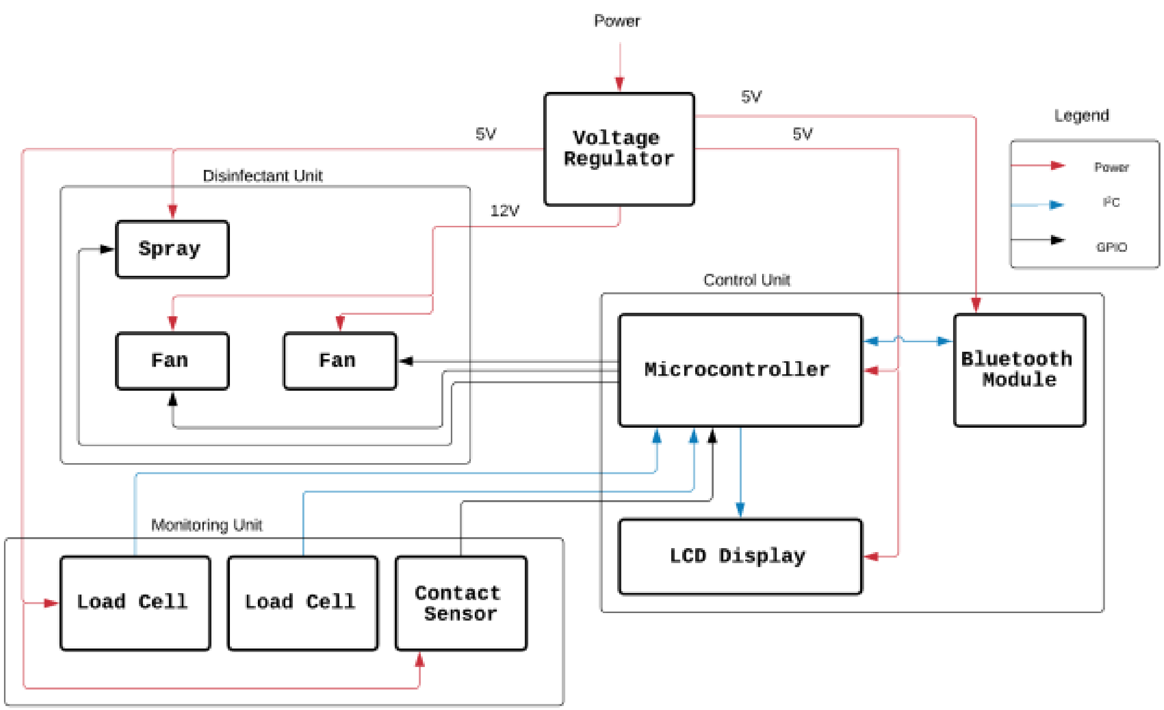

Figure 4: Project Block Diagram

2.1 Design Alternatives

During the design process of the Self-Cleaning locker, there were a few changes to our original design that we had to make:

1. Choosing an automatic sprayer over a manual sprayer: We decided on choosing an automatic sprayer due to conversations with the ECE Machine Shop. Having a manual sprayer had more potential go to awry, as we would need to implement a servo motor to push down on the sprayer. However, if the sprayer got moved, or if our motor broke, then our disinfectant subsystem would not work anymore. As the machine shop said, less moving parts allow for a less potential for error. Additionally, it would have been harder to aim our sprayer to cover as much interior surface area.
2. The Android application initially was to update in real time, with our exterior load cell continuously feeding information to our application. However, throughout the debugging process, we were not able to get this working, as the Bluetooth connection could not continuously get the values from the microcontroller. We ended up implementing a "Refresh" button that reestablishes the Bluetooth connection every time that the button is pressed, and then sends the value of the disinfectant supply. This ended up working for us, but we would still like to be able to implement a continuous stream of data to our Android application in the future.

2.2 Power Supply

Our power supply consisted of a 12V DC Wall Power Supply adapter and a voltage step down power converter. We chose the wall power supply due to its ease of use in allowing a user to be able to just plug in the locker into a wall outlet, and have the locker be able to be powered just off that. Additionally, we wanted 12V in order to supply more voltage to our sprayer, along with our fans, to help make the sanitation process more efficient.

In order to supply 5V to necessary components, we used a voltage step down converter. This was used to supply 5V to our microcontroller and Bluetooth transceiver, as their corresponding data sheets specified to use 5V for those particular components. Table 1 shows the effectiveness of our voltage step down converter.

Table 1: Voltage Step Down Converter Values

Input Voltage	Desired Voltage	Output Voltage
12V	5V	4.994V

2.3 Control Unit

For our control unit, we used a ATMEGA328P-PU microcontroller, HC-05 Bluetooth Transceiver, and LCD display. For all three components, there was a wide variety of resources online, such as documentation and example resources, making it easy for our team to understand how the components worked together as well as within the larger scope of our project. Additionally, all three components were cheap, so they fit within our project's budgeted cost.

First, we chose the ATMEGA328P-PU microcontroller because our project only needed a simple microcontroller since we did not really need a lot of compute resources. Additionally, our microcontroller can communicate via I2C and serial communications, so it has support for many sensors.

Second, we chose the HC-05 Bluetooth Transceiver due to the fact that it was cheap (in terms of cost) and communicates via serial (TX/RX). Regarding our HC-05 Bluetooth Transceiver, we needed to implement a voltage divider circuit within our PCB due to the transmit pin having a 5V output, but the receive pin only supporting 3.3V. Equation 1 shows the math to get 5V to 3V.

$$Vout = Vin * \frac{R1}{(R1 + R2)} \quad \text{(Eq. 1)}$$

Replacing R1 with 2000 Ohms, and R2 with 1000 Ohms, and using 5V as our input voltage, we get an output voltage value of 3.3V. Table 2 shows the specified voltage values that we got from our multimeter.

Table 2: Voltage Step Down Converter Values

Input Voltage	Transmit Pin	Receive Pin
5V	4.99V	3.35V

Third, we chose using an LCD display due to the fact that it is also easy to use and communicates via I2C. For our project, we only needed a small amount of space to convey the information that we needed to print, so 16x2 size was ample enough for us to use in our case. Additionally, the LCD display can display a wide range of characters, so it was helpful for us to communicate messages to users of the locker.

Finally, we used relays in our project because we did not want to direct power through our microcontroller in case it could burn our microcontroller out. We only wanted to use our microcontroller to send signals to switch our fans and motor on and off, thus we needed to use relays to appropriately perform those actions. The relays gave us much flexibility in terms of whether we wanted to use high or low signals to signify either on or off. Even though there are several positives to using relays, the biggest negative to using relays are that they are more expensive than transistor switches. Even with this, our team believed that relays in conjunction with the ATMEGA328P-PU microcontroller, HC-05 Bluetooth

Transceiver, and LCD display would be all that we needed to have the perfect control unit for our project.

2.3 Disinfectant Unit

The disinfectant unit is responsible for sanitizing the interior of our locker when no items are inside of it, along with the locker door being closed. This unit is consisted of an automatic sprayer, along with two fans. The sprayer was disassembled when ordered, and instead of using the battery pack that was provided with the sprayer, we soldered wires to the motor wires of the sprayer to provide 12V from our own power source. The fans are used to help dry the interior after disinfectant is sprayed on the surfaces of the inside through the circulation of air.

We chose using an automatic sprayer rather than a sprayer with a nozzle for several reasons as described in section 2.1. Overall, the automatic sprayer provided less room for error, as a manual sprayer with a servo motor had more moving parts that were prone to breaking. Additionally, we knew we could implement a relay to easily to turn the automatic sprayer on when we wanted to through our microcontroller programming.

Our input will be information from our microcontroller to let this subsystem know when to turn on, along with 12V for our fans and sprayer. Our output for this subsystem will be our sprayers and fans turning on when desired. The output voltage and current values for the sprayer and fans are shown in Tables 3 and 4 as below.

Table 3: Sprayer Voltage and Current Values

Input Voltage	Output Voltage	Current
12V	12.22V	0.11A

Table 4: Fans Voltage and Current Values

Input Voltage	Output Voltage	Current
12V	12.33V	0.13A

We implemented relays in our design in order to control with our microcontroller when we wanted our fans and sprayer to turn on. The relays were in conjunction with our microcontroller to have the fans and sprayer function as intended within the logic of our design.

However, we came quite short of our high-level requirement target of 90% with a 5% error range. Our locker dimensions were 15” by 15” by 24”. To calculate the interior surface area, we used Equation 2 as shown:

$$Interior\ Surface\ Area = 4 * (Base * Height) + (Base * Base) \qquad \text{(Eq. 2)}$$

We excluded the top of the locker in our calculations, as we felt that the top of the locker was not as important to cover compared to the sides and bottoms. The final interior surface area that we got for our locker was 1665 in^2. By measuring the area that got wet in the interior by hand, we were only able to cover roughly 318.75 in^2. This is only 19.1% area coverage, much lower than our initial high-level requirement.

2.4 Monitoring Unit

The monitoring subsystem is used to check whether there are items inside the locker and if the locker door is closed in order for the microcontroller to know when to start the sanitation sequence. Additionally, this subsystem is used to monitor the disinfectant supply level of the locker in order for it to communicate with our Android application. The disinfectant supply level should be communicated to an Android application that allows users of the application to see the relative level of the disinfectant supply (intervals of 25% from 100% capacity).

To monitor if the locker is ready to sanitize, the locker makes use of a load cell and load cell amplifier for the interior of the locker, along with a contact sensor to detect whether the locker door is closed to initiate the sprayer. Our initial high-level requirement for our locker required for the interior load cell to be able to detect at least 250 grams. During our testing, we were able to surpass that initial target by about 31%, as an iPhone 12 with a thin case on it weighed about 172 grams. Additionally, we wanted the error range of the load cell and load cell amplifier’s accuracy to be less than 5%. The readings for the interior load cell as shown below in Table 5, along with a graphical view of our readings shown in Figure 5.

Table 5: Interior Load Cell Weight Values & Percent Difference

iPhone 12 with Case	Median Load Cell Reading	Percent Difference
172 grams	174.66 grams	1.54%

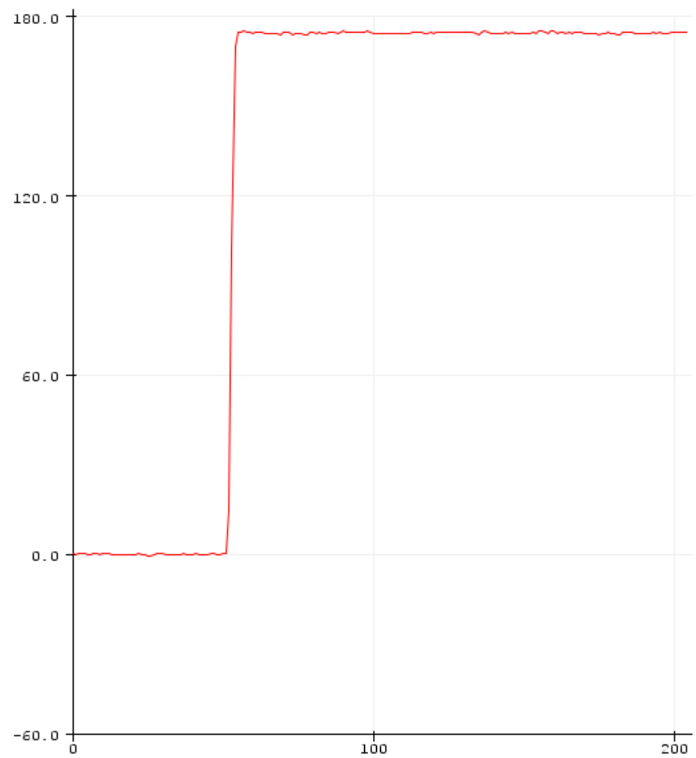

Figure 5: IDE Interior Load Cell Readings (Grams vs. Time (ms))

For the interior load cell, we were able to stay within our target error range of 5%, as our percent difference was only 1.54%.

For our contact sensor, through initial testing, we found out that it acts like a switch. If the two parts of the sensor are together, the output voltage read 4.95V. If they were separate, 0V was read. This was implemented within our microcontroller programming in that if voltage were read from our contact sensor, it would read it as “on”. This would be integrated into our microcontroller logic as a whole to make sure when to initialize the spraying mechanism.

For our exterior load cell that detects our disinfectant supply level, we went through the same process in testing our interior load cell. Taking our 1.89L carton that held our disinfectant solution, we filled it up to full capacity and checked to see if our load cell readings were accurate. Our readings are shown in Table 6, along with a graphical view of our readings in Figure 6.

Table 6: Exterior Load Cell Weight Values & Percent Difference

Full Disinfectant Supply	Median Load Cell Reading	Percent Difference
1890 grams	1970.43 grams	4.25%

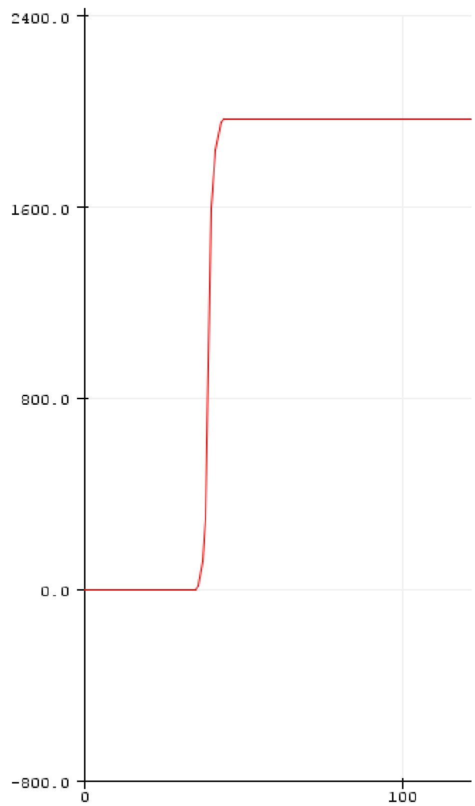

Figure 6: IDE Exterior Load Cell Readings (Grams vs. Time (ms))

Once again this stayed within our error range of 5% with a percent difference of 4.25%. This percent difference was much larger, and we think with the increasing weight of objects placed on the load cell, accuracy starts to diminish. This worked out fine for our initial project, but if this were to be implemented in a real-world scenario, we would have to fine tune our calibration to make sure the load cell would stay within a 5% error difference if the disinfectant supply tank were larger and heavier.

2.4.1 Software

The software that we chose to make our application was Android studio. This was due to how not everyone in our group had a Mac computer to be able to create an application in iOS, but everyone could download Android Studio regardless of the operating system their computer ran on.

Additionally, there was a lot of resources and documentation online for Android Studio, as it is the official IDE for Android development. They have a lot of tools to help make the application development process easier, such as showing the application layout in real time based on the XML code that is written, along with support for virtual device emulation.

The Android application was written in Java, and its purpose was to show a battery icon based on how much capacity was left in the lockers disinfectant supply. The disinfectant supply level was updated as follows:

- 100% - 76% disinfectant capacity: Show full battery image (green and 4 bars)
- 75% - 51% disinfectant capacity: Show ¾ battery image (yellow and 3 bars)
- 50% - 26% disinfectant capacity: Show ½ battery image (orange and 2 bars)
- 25% - 0% disinfectant capacity: Show ¼ battery image (red and 1 bar)

Additionally, the flow chart to our Android application is shown in Figure 7 below:

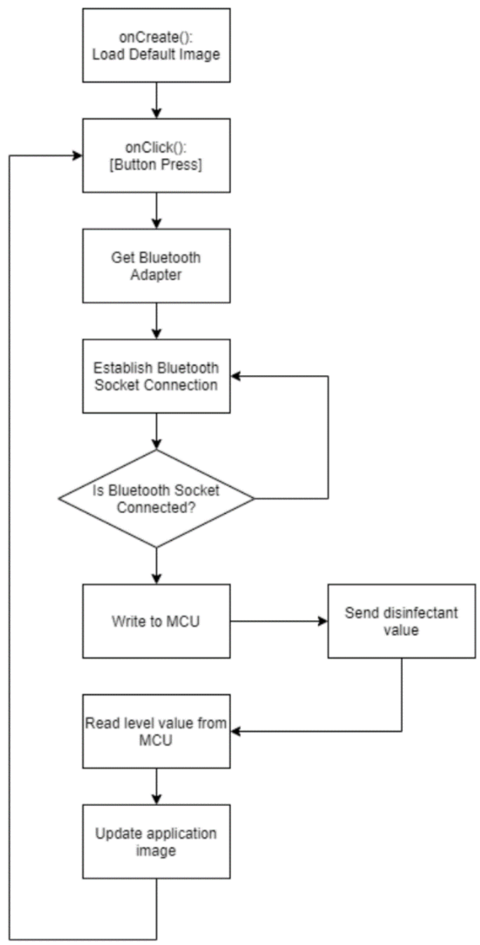

Figure 7: Android Application Flowchart

Screenshots of our application can be found in Appendix E, along with a link to the GitHub repository that has the source files of the application.

2.5 Tolerance Analysis

One important tolerance we want to maintain is being able to detect objects of at least 250 grams. Our strain gauge load cell will be used to detect if there are objects inside the locker to know whether to initiate cleaning. However, readings from the load cell are very small. Our load cell has a Wheatstone Bridge circuit built into it, with gauge sensors replacing the resistors shown in the figure below:

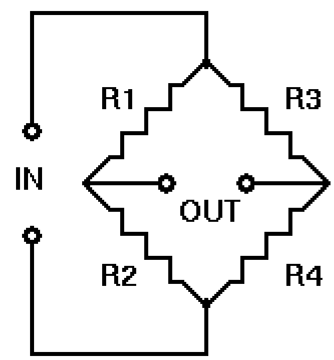

Figure 8: Wheatstone Bridge Circuit

These strain gauges commonly have a base resistance of either 120Ω, 350Ω, or 1000Ω (we will use 120Ω as a placeholder value, as the strain gauge load cells we ordered do not tell us) [7]. If R1 / R2 = R3 / R4, then our voltage output will be 0. However, when there is strain on one of these gauges, it will increase its resistance value based on the amount of strain. This percent change in resistance is given by:

$$\%\ Change\ in\ Resistance\ =\ [(Gauge\ Factor)\ \cdot\ Strain]\ \cdot\ 100 \qquad \text{(Eq. 3)}$$

Based on the Wheatstone Bridge, the output Voltage will be calculated using:

$$V_{out}\ =\ \left[\left(\frac{R3}{R3+R4}\right)-\left(\frac{R2}{R1+R2}\right)\right]\cdot V_{in} \qquad \text{(Eq. 4)}$$

We want to be able to detect at minimum 250 grams. We calculate the amount of strain by first finding the amount of stress that at 250-gram object would create. We use the following stress equation along with our locker dimensions to find this value:

$$Stress\ =\frac{Force}{Area} \qquad \text{(Eq. 5)}$$

$$Stress\ =\frac{\left(0.250kg\ \cdot\ 9.81\ {}^{m}/_{s^2}\right)}{(0.35m\ \cdot\ 0.31m)}\ =\ 22.6\ Pa$$

To find strain, we can use the following equation to find the relationship between stress and strain:

$$Stress = (Elastic\ Modulus) \cdot Strain \quad \text{(Eq. 6)}$$

The elastic modulus is the proportionality constant based on the material of the strained object. Our load cell is aluminum, so the modulus will have a value of 7.0 * 10^10 Pa [8]. Plugging these numbers into our equation 3, we get a strain value of 3.23 * 10^-10 **ε**. Plugging this strain value into equation 1, with a Gauge Factor of 2 (a common gauge factor for metallic strain gauges), we end up with a 6.46 * (10^-8) % change in resistance.

This change in resistance is very small, and plugging that into equation 2 for R1, with a voltage input of 5.5V, we would only get a voltage output of 0.0888 µV. While this value is what is expected out of a load cell, it is still a very small value to read.

To get more accurate measurements, especially with our load cells that will be reading change of weight on top of it (for our disinfectant supply monitoring system), we are using the HX711 Load Cell Amplifier. What this load cell amplifier does is allows our load cell output to be amplified with a gain of either 128 or 64, with a full-scale differential input voltage of +/-20mV or +/-40mV, based on what gain we choose [9]. Using this load cell amplifier, this will enable us to detect our base weight of 250 grams, along with being able to accurately tell between different weight values on our load cell to correctly implement our disinfectant supply monitoring system.

3 Design Verification

Our full Requirements and Verification table can be found in Appendix F.

3.1 Power Supply

1. To verify that our voltage step down converters sufficiently dropped 12V to 5V, we used a multimeter to check the output voltage value of our converter. This value was 4.994V.

3.2 Control Unit

1. We verified that our microcontroller should correctly communicate with our LCD display to display the status of the locker. When the sprayer and fans are running, the LCD display shows “Locker Cleaning.” When the locker has an object inside, the LCD displays “Locker Not Clean.” When both the sprayer and fans are finished running, the LCD shows “Locker Clean.” Finally, when exterior load cell detects that the disinfectant supply is less than 25% capacity, the LCD display shows “Low Disinfectant.” This was verified during our demonstration.
2. We verified that our microcontroller was able to send data to our HC-05 Bluetooth transceiver. This was verified due to the Bluetooth app being able to read the value sent by the microcontroller and showing it on the applications display.
3. We verified that our HC-05 Bluetooth Transceiver was able to communicate information regarding our disinfectant level at 100%, 75%, 50%, and 25% capacities, with an error range of less than 5%. This was verified with our Android application correctly updating the image based on the value sent, as shown in demonstration.
4. We verified that our HC-05 Bluetooth Transceiver was able to communicate with a range of 20 feet. This was verified by walking away from the transceiver and measuring how far the connection held.
5. We verified that the LCD display was visible from one meter away. This was verified by standing a meter away and being able to read the text shown.
6. We verified that LCD display changed its text based on the status of the locker. This was explained earlier and verified during demonstration.

3.3 Disinfecting Unit

1. We were not able to meet our high-level requirement of having our sprayer be able to cover 90% of the interior surface area of the locker, with an error range of +/- 5%. This was due to how our sprayer nozzle was not able to output a mist like we originally intended. The sprayer outputted a cone of disinfectant and did not cover as much area as we had hoped. Additionally, we originally ordered a locker that was 3 inches smaller in depth and width than the locker we demonstrated in, but the original locker was not able to come in due to it going out of stock. The smaller depth and width would have made it a little easier to come closer to our benchmark of 90%. This was verified by hand measuring the interior of the locker that was wet. We were only able to cover about 19.1% of the interior surface area (excluding the top).

2. We were able to verify that the fans were being powered with 12V with an error range of 5%. We verified this by using a multimeter, which ended up reading a voltage value of 12.33V.
3. We were able to verify that our fans dried the interior of our locker. We verified this using our touch and paper towers to see if the interior was dry. However, this took a very long time, as we had to keep the fans running for approximately 50 minutes due to the fact that our fans were powered at 12V. The reason why it took this long is because the sprayer that was used sprayed the disinfectant solution as a cone spray rather than a fine mist. Due to this, there was more disinfectant that was sprayed into the interior of the locker than originally anticipated. Thus, more time was needed to completely dry out the interior of the locker.

3.4 Monitoring Unit

1. We were able to verify that our interior load cell was able to detect at least 250 grams. This was verified by using a scale to weigh an object, in our case an iPhone 12 with a thin case, placing it on top of our interior load cell, and checking our output of our microcontroller program if it detected it. Our iPhone 12 with a case weighed around 172 grams, and the disinfectant process did not occur, as seen in demonstration. Additionally, our readings from the Arduino IDE were accurate in detecting around 172 grams.
2. We were able to verify that the load cell should output different values based on different weight values. Using the Arduino IDE, after calibrating our load cell, we were able to get different weight readings based on the weight of the object that was placed on the load cell. As seen in Tables 5 and 6, the accuracy of our load cells and load cell amplifiers were within a 5% error range for actual values.
3. We were able to verify that the contact sensor was able to correctly output when the two parts were together and separate. This was verified by creating a test circuit that had an LED light up when the sensors were together and turned off when they were separated.
4. We were able to verify that the Android application was able to read different disinfectant supply levels on top of the exterior load cell. This was tested by passing in the value of the load cell to the HC-05 module, and then printing that value on our Android application screen and checking if they were equal.
5. The Android application was able to report information regarding the status of the locker. Depending on where the disinfectant level value fell under (in terms of percentages), it would update the Android application image as follows:
    a. 100% - 76% disinfectant capacity: Show full battery image (green and 4 bars)
    b. 75% - 51% disinfectant capacity: Show ¾ battery image (yellow and 3 bars)
    c. 50% - 26% disinfectant capacity: Show ½ battery image (orange and 2 bars)
    d. 25% - 0% disinfectant capacity: Show ¼ battery image (red and 1 bar)

    This was verified during demonstration and testing the application.

6. The Android application was able to connect to the HC-05 Bluetooth Transceiver from 20 feet away. This was tested by walking away 20 feet and trying to connect the application to the transceiver.

4 Costs

Our fixed development costs are estimated to be $40/hour, for 12 hours a week for three people. We are using the length of a semester to determine the number of weeks to work on this (16).

$$3 \cdot \frac{\$40}{hour} \cdot \frac{12\ hours}{week} \cdot 16\ weeks \cdot 2.5\ =\ \$57{,}600 \qquad \text{(Eq. 7)}$$

Table 7: Parts Costs

Part	Cost (prototype)	Cost (bulk)
Battery Powered Sprayer (Amazon; Craftsman; B08KH81W9Q)	$7.56	$7.56
ATMEGA328P-PU Microcontroller (Digi-Key; Microchip Technology; ATMEGA328P-PU-ND)	$2.52	$2.09
HC-05 Wireless Bluetooth RF Transceiver (Amazon; HiLetgo; B071YJG8DR)	$7.99	$7.99
MC-38 Wired Door Sensor (Amazon; Gikfun; B0154PTDFI)	$6.98	$6.98
DC 12V Cooling Fan (Amazon; PANO-MOUNTS; B07D493BDX)	$13.99	$13.99
HX711 Load Cell Amplifier (SparkFun; SEN-13879 ROHS) x2	$19.90	$8.46
LCD Module (Amazon; KNACRO; B01ID8O574)	$6.22	$6.22
Strain Gauge Load Cell - 4 Wires - 20Kg (Adafruit; 4543) x3	$11.85	$3.16
Locker (Ikea; 204.765.20)	$40	$40
12V DC 2A Wall Power Supply Adapter (Amazon; XINKAITE; B07GRZB5Y9)	$9.99	$9.99
10pcs Mini360 3A DC Voltage Step Down Power Converter (Amazon; SongHe; B07T7L51ZW)	$6.88	$6.88
Capacitor Ceramic 22pF (SparkFun; COM-0857) x5	$1.25	$0.23
Oscillators (Digi-Key; 887-2015-ND) x3	$0.90	$0.14
Total	**$136.03**	**$113.69**

We will only be making one locker, and therefore our total prototype cost will be $136.03 for one unit, which includes the locker cost. Without the locker cost, the rest of the components have a total of $96.03.

The total development cost will be $57,736.03. The quoted machine shop labor is $56.12 an hour, with 12-16 hours spent on it approximately.

5 Schedule

Table 8: Project Schedule

Week	Chilo	Nithin	Immanuel
3/8/2021	Go to lab to start testing components and building subsystems	Go to lab to start testing components and building subsystems	Go to lab to start testing components and building subsystems
3/15/2021	Develop baseline Android application	Continue building out physical circuit schematic	Finalize PCB Design 1
3/22/2021	Test Bluetooth module with Android application	Testing load cell accuracy for overall project	Finalize PCB Design 2
3/29/2021	Connecting sanitation subsystem to monitoring unit on breadboard	Connecting sanitation subsystem to monitoring unit on breadboard	Connecting sanitation subsystem to monitoring unit on breadboard
4/5/2021	Program microcontroller	Program microcontroller	Finalize PCB Design 3
4/12/2021	Final testing	Final testing	Final testing
4/19/2021	Mock Demo & remove bugs	Mock Demo & remove bugs	Mock Demo & remove bugs
4/26/2021	Demonstration	Demonstration	Demonstration
5/3/2021	Presentation & Final Paper	Presentation & Final Paper	Presentation & Final Paper

6. Conclusion

6.1 Accomplishments

The Self-Cleaning Locker was able to correctly function as intended. If there were no objects detected on top of the interior load cell, and the locker door was closed, then the sprayer mechanism would initiate, and shortly after, the fans would run as well. Our LCD display also correctly displayed the state of the locker, and if disinfectant supply hit less than 25% capacity.

Additionally, our Android application works as intended as well. Based on whether our exterior load cell detects 100%, 75%, 50%, or 25% disinfectant supply capacity, our "Refresh" button allowed the app to update the battery image shown in the application correspondingly.

6.2 Uncertainties

Although we believed our project turned out to be a success, there were two main aspects of the project that we were uncertain about and could have made our project even better. The first is that the sprayer that we used in the locker was unable to cover the 90% of the interior surface area that we wanted. Instead, the sprayer covered roughly 19.1% of the interior surface area due to the positioning and angle of the sprayer. Additionally, the sprayer sprayed a heavy cone of water, opposed to the light mist that we desired initially. Because the aspects of the sprayer were different from the ones we wanted, our team was left with results that didn't match up with what we were looking for. The results of the sprayer before and after the disinfecting cycle were calculated and determined appropriately using a measuring tool and calculator.

The second has to do with power step-down converters that our team wanted to use for our project. While doing some unit testing with these components, they worked fine within our subsystems. However, when trying to assemble the entire project together, the step-down converters were not able to hold and instead burnt out one of our microcontrollers. When putting the entire circuit together, the step-down converter did not properly convert the 12V from the wall outlet plug to 5V and instead 12V was sent to the microcontroller. We verified that this was the conclusion after using a multimeter to check the results. For future work, we would use an adjustable spray that sprays a fine mist throughout the locker. Additionally, we would use a better-made power step-down converter to make sure that our 12V gets properly converted to 5V and to ensure that our microcontroller is not burned out.

6.3 Ethical considerations

We are responsible for keeping the public's safety, health, and welfare in mind when designing this project. This refers to the IEEE Code of Ethics, #1, stating "to hold paramount the safety, health, and welfare of the public" [10]. We must be sure that our locker does not pose any safety concerns for people when using our device. We will go into how we would implement the required safety restrictions in our project later in this section.

Another ethical concern that is out of our control would be the discrimination of use with our product. This refers to the IEEE Code of Ethics #7, stating "to treat all persons fairly and with respect, and to not engage in discrimination based on characteristics such as race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression" [10]. While we will never discriminate the use of our product based on the characteristics of a person such as race and gender, in a real-world application of our product where it is readily available in public gyms, certain establishments may discriminate against certain people on the use of the lockers. We do not have a solution for this, as it is out of our control as to how someone else may allow others to use our locker. The best thing we can do is to investigate whether a buyer has a good record of respecting an individual's race, gender, religion, etc., if we were to sell this product.

In terms of safety for our self-cleaning locker, we must make sure that our pressure sensor is accurate in detecting whether there are items, even small ones, in our locker. We must make sure that no one's belongings get wet on accident by our sensor not detecting that they are there. Another safety issue that can come up is the issue of wiring in our project. We want to make sure that our wires do not short and cause damage to anything. We will address this by covering our wires with electrical tape whenever possible, along with having circuitry in a separate encasing on the exterior of the locker.

Another health concern will be the issue of the toxicity of disinfectant used in our locker. There has been research on the effects of disinfectant spray inhalation and the effect that it has on people. There have been studies on asthmatic patients inhaling disinfectant spray (three doses of 0.6 mg of benzalkonium chloride in water), and 20% of those patients observed signs of Bronchoconstriction [11]. The amount of spray that is released and how much the fan dries it off to prevent inhalation of sanitation solution must be monitored carefully, along with looking at more research, to ensure that our locker does not pose any health concerns to users of the locker along with people in the surrounding area.

An additional health concern would be our locker being a potential fire hazard. Because we have electrical parts that could cause a spark when burning out or shorting, we must make sure that our disinfectant supply must not be flammable or combustible to prevent a fire in our locker, or even an explosion. The solution we decided on due to this issue is 3% Hydrogen Peroxide (H_20_2), and this solution is not combustible [12].

6.4 Future work

An improvement that could be added to our initial design would be making our final product more presentable. Our initial demonstration locker had a lot of wires hanging out, along with our circuitry and components being exposed. If this were to be mass produced for real world use, this should be much more presentable. Additionally, we need to improve our sprayer mechanism, as the initial high-level requirement of 90% interior coverage was not met by the time of demonstration. Some solutions regarding this could be implementing a different sprayer nozzle to allow the disinfectant to be sprayed as a mist, along with adding an additional sprayer nozzle to the interior of the locker. Additional fans could be added as well to speed up the drying process to allow users of the locker to be able to access it faster.

Some additional features that could be added to the locker include implementing a biometric lock. A biometric lock would allow for an increase in security due to having a fingerprint sensor and would allow a convenience for users to not have to worry about bringing their own lock. An additional hook weight sensor could be added as well, which would allow users to hang their belongings if they chose to (such as jackets).

If this locker were to be implemented in a real-world scenario, additional HC-05 Bluetooth Transceivers would have to be implemented. Each disinfectant supply tank would have to have its own Bluetooth Transceiver to communicate with the Android application. This could be implemented without too much trouble however, as each HC-05 transceiver has a unique MAC address corresponding to that particular device. Differentiating the disinfectant supplies / HC-05 transceivers would be able to be implemented by labeling each different HC-05 Bluetooth Transceiver a unique name that would allow a gym owner to tell which one is which.

References

[1] “Germs in the Locker Room,” FitRated. [Online]. Available:https://www.fitrated.com/resources/germs-in-the-locker-room/#:~:text=According%20to%20our%20test%20swabs,room%20faucets%20were%20even%20dirtier. [Accessed: 18-Feb-2021].

[2] J. Sensakovic, “How Long Does Coronavirus Last On Clothes?,” Hackensack Meridian Health, 27-Aug-2020. [Online]. Available: https://www.hackensackmeridianhealth.org/HealthU/2020/08/27/how-long-does-coronavirus-last-on-clothes/#:~:text=Research%20suggests%20that%20COVID%2D19,days%20for%20plastic%20and%20metal. [Accessed: 18-Feb-2021].

[3] J. Migala, “How is Coronavirus Spread? Here's What You Should (and Shouldn't) Worry About,” Health, 18-Aug-2020. [Online]. Available: https://www.health.com/condition/infectious-diseases/how-is-coronavirus-spread. [Accessed: 18-Feb-2021].

[4] “7-segment Display and Driving a 7-segment Display,” Basic Electronics Tutorials, 16-Feb-2018. [Online]. Available: https://www.electronics-tutorials.ws/blog/7-segment-display-tutorial.html. [Accessed: 19-Feb-2021].

[5] L. Ada, “Adafruit 16-Channel PWM/Servo HAT & Bonnet for Raspberry Pi,” Adafruit Learning System. [Online]. Available: https://learn.adafruit.com/adafruit-16-channel-pwm-servo-hat-for-raspberry-pi/powering-servos. [Accessed: 19-Feb-2021].

[6] N. *, “Weight Sensor : Different Types and Its Specifications,” ElProCus, 03-Feb-2020. [Online]. Available: https://www.elprocus.com/weight-sensor-working-and-its-specifications/. [Accessed: 19-Feb-2021].

[7] S. Al-Mutlaq, Getting Started with Load Cells. [Online]. Available: https://learn.sparkfun.com/tutorials/getting-started-with-load-cells/all. [Accessed: 04-Mar-2021].

[8] W. Moebs, S. J. Ling, and J. Sanny, “12.3 Stress, Strain, and Elastic Modulus - University Physics Volume 1,” OpenStax. [Online]. Available: https://openstax.org/books/university-physics-volume-1/pages/12-3-stress-strain-and-elastic-modulus#fs-id1163713135600. [Accessed: 04-Mar-2021].

[9] “24-Bit Analog-to-Digital Converter (ADC) for Weigh Scale,” SparkFun. [Online]. Available: https://www.sparkfun.com/products/13879. [Accessed: 04-Mar-2021].

[10]“IEEE Code of Ethics,” IEEE. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 18-Feb-2021].

[11]P. A. Clausen, M. Frederiksen, C. S. Sejbæk, J. B. Sørli, K. S. Hougaard, K. B. Frydendall, T. K. Carøe, E. M. Flachs, H. W. Meyer, V. Schlünssen, and P. Wolkoff, “Chemicals inhaled from spray cleaning and disinfection products and their respiratory effects. A comprehensive review,” International Journal of Hygiene and Environmental Health, 15-Aug-2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1438463920305381. [Accessed: 05-Mar-2021].

[12]Hazardous Substance Fact Sheet [PDF]. (2016, May). New Jersey Department of Health.

Appendix A: Initial Project Sketches

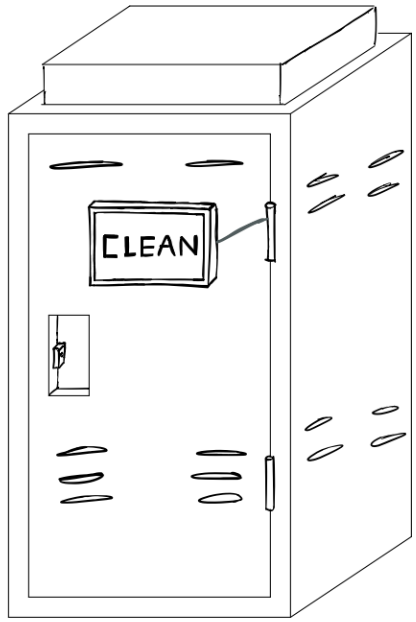

Figure 9: Initial Locker Exterior Sketch

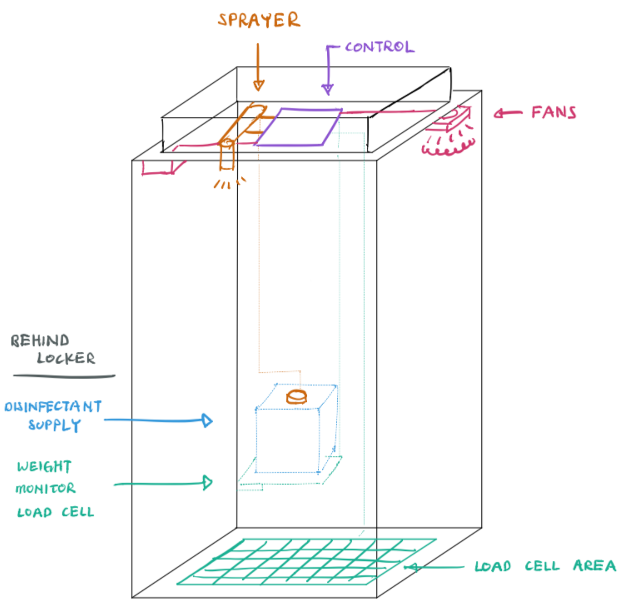

Figure 10: Initial Interior Locker Sketch

Appendix B: Final Schematic

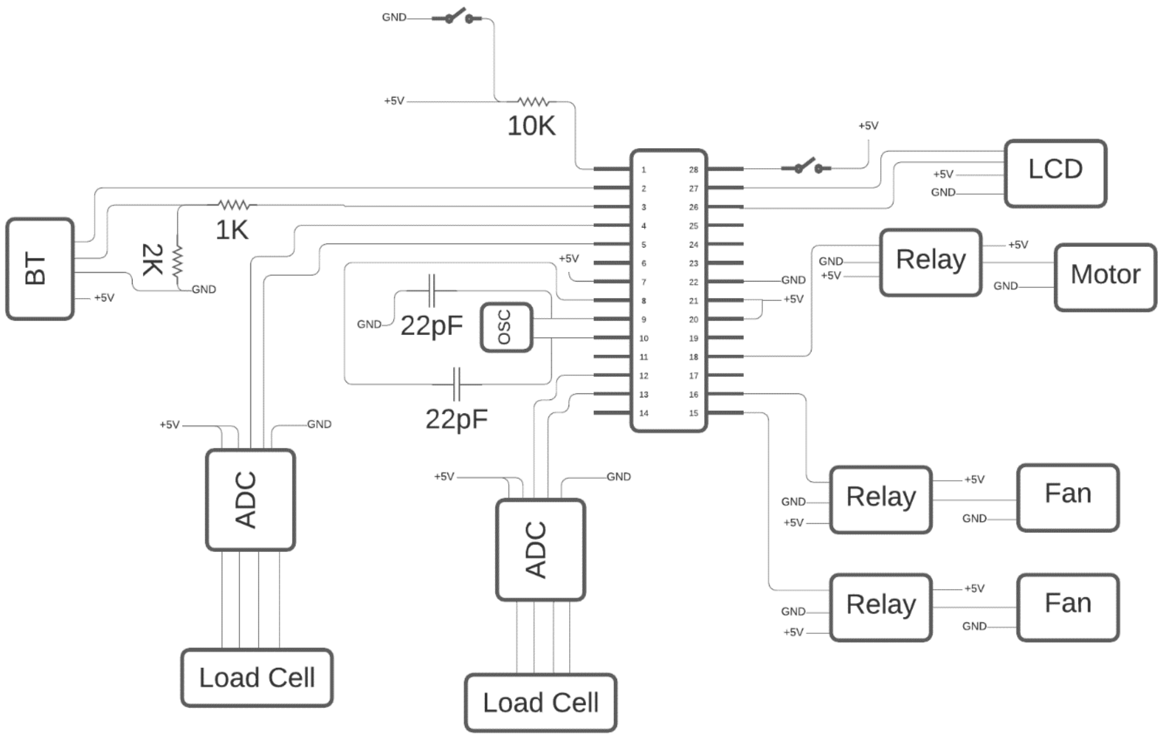

Figure 11: Schematic

Appendix C: Final PCB Layout

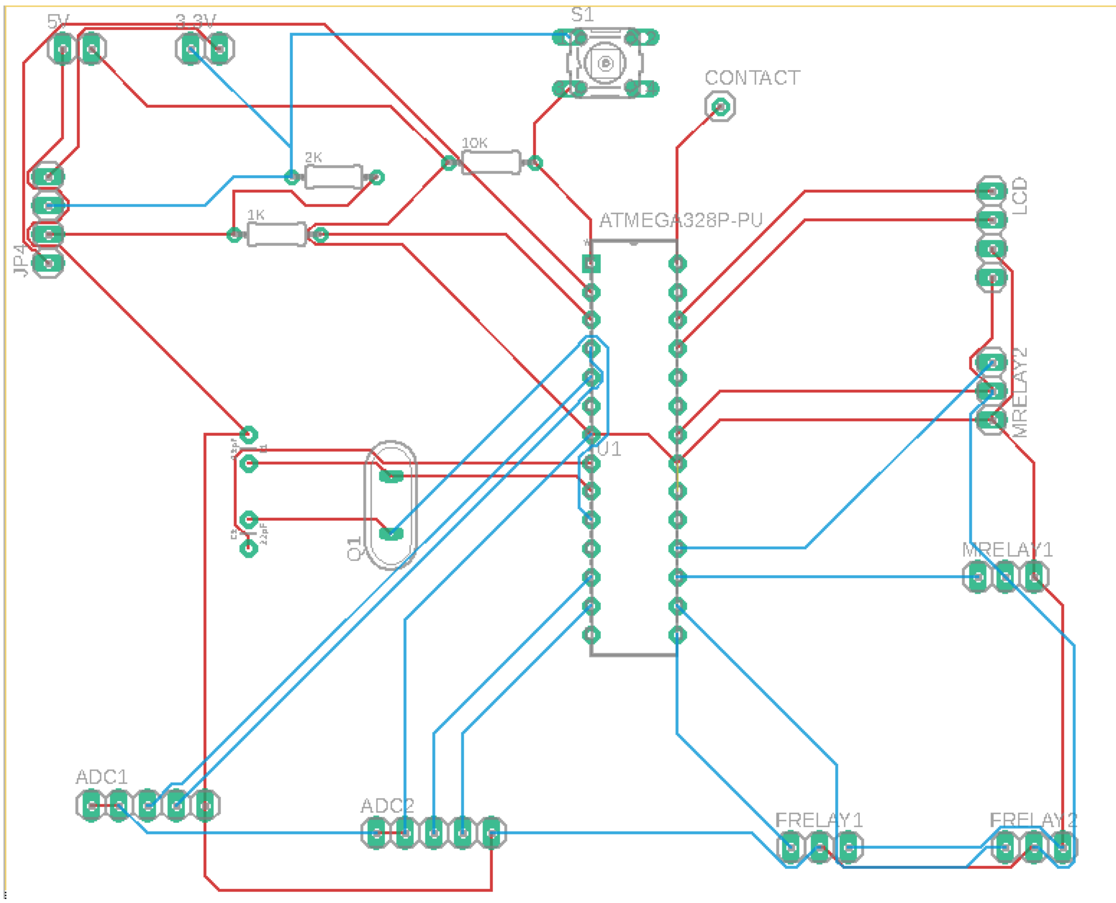

Figure 12: PCB Layout

Appendix D: Physical PCB

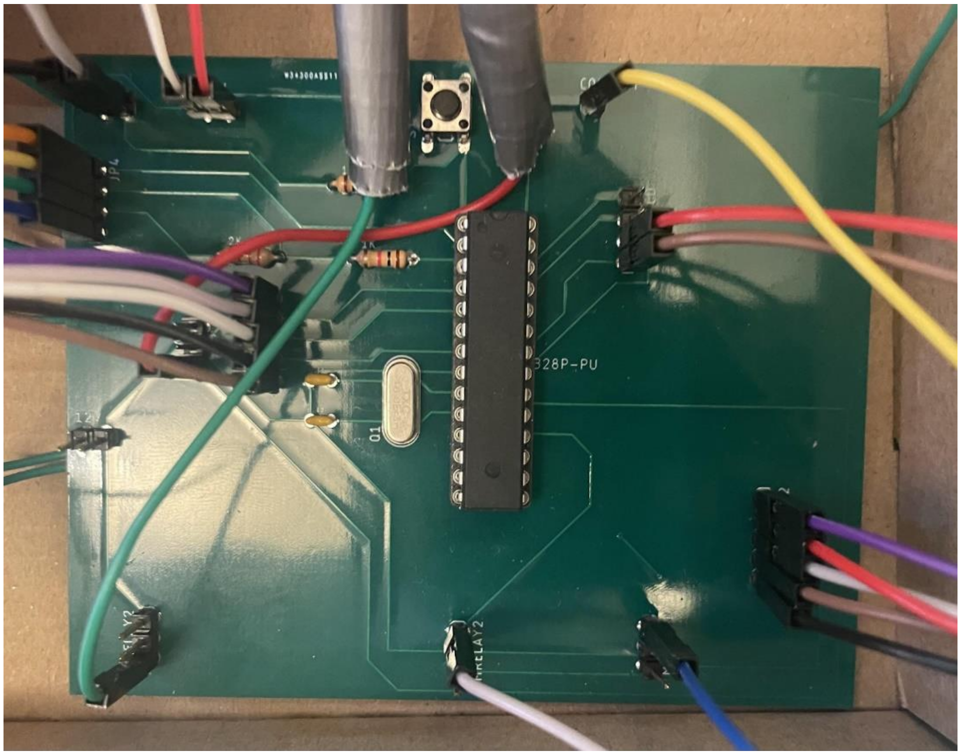

Figure 13: Picture of Physical PCB Board

Appendix E: Software Development

Project Repository Link: https://github.com/cllamas98/ECE445ApplicationCode

Application Screenshots:

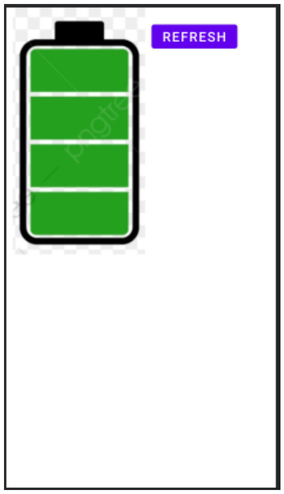

Figure 14: 100% Disinfectant Supply Android Screenshot

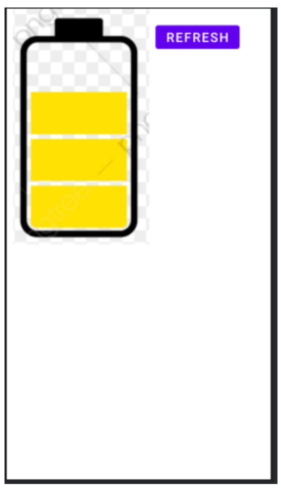

Figure 15: 75% Disinfectant Supply Android Screenshot

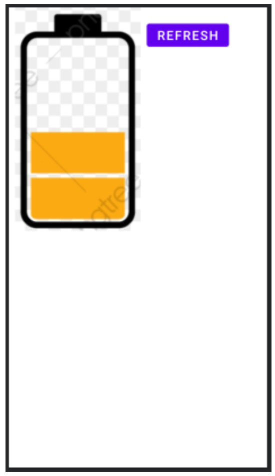

Figure 16: 50% Disinfectant Supply Android Screenshot

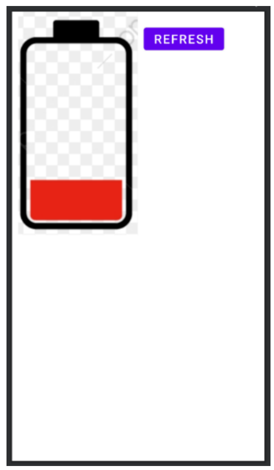

Figure 17: 25% Disinfectant Supply Android Screenshot

Appendix F: Requirement and Verification Table

Table 8: System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
Power System Requirements 1. Voltage step down converter should convert 12V to 5V +/- 5%.	Verification 1. Provide 12V from wall power supply and adjust step down converter to output 5V +/- 5% error, and check using multimeter.	Status 1. Y: 4.994V
Control Unit Requirements 1. Microcontroller should correctly communicate progress of sanitation cycle with LCD display. 2. Microcontroller should program the Bluetooth module to transmit data to the Android application. 3. Module needs to correctly communicate information about disinfectant supply level at 100%, 75%, 50%, 25%, and 0% (+/- 5% error range) with Android application. 4. Module must be able to receive and transmit information accurately from up to 20-30 feet away. 5. LCD should be visible from one meter away. 6. LCD should display “CLEANING” when locker is sanitizing. 7. LCD should display “CLEAN” when locker sanitation has finished. 8. LCD should display “LOW SUPPLY” when disinfectant supply hits 25% (+/- 5% error range).	Verification 1. Check that LCD correctly displays corresponding message. 2. Bluetooth module should change Android Studio global variable value. 3. Using Bluetooth terminal application and serial communication on Android phone along with code on Arduino, check that output on Bluetooth terminal shows values equal to output of weight sensor for disinfectant supply in the correct format. 4. Test that connection holds on physical Android phone 20-30 feet away from locker. 5. Check that LCD display is visible within one meter of standing. 6. Check that LCD correctly displays “CLEANING” using I2C communication while sprayer and fan components are on 7. Check that LCD correctly displays “CLEAN” using I2C communication when sprayer and fan components are done / off. 8. Calibrate microcontroller and load cell amplifier with	Status 1. Y 2. Y 3. Y 4. Y 5. Y 6. Y 7. Y 8. Y: Used 1890 grams and 472.5 grams due to disinfectant supply container volume.

	default weight (1000 grams). Place weight that is 250 grams (25% of 1000) and see if LCD correctly displays "LOW SUPPLY" using I2C communication.	
Disinfectant Unit Requirements 1. Have the spray be able to cover around 90% +/-5% of the interior surface area. 2. Fans should be powered with 12V +/- 5% coming from the outlet plug. 3. Fans should circulate air throughout the inside of the locker to ensure that the interior is dry.	Verification 1. Measure how much area spray covered of interior by hand and divide by total interior surface area. (Colored area in meters^2 / Total interior surface area in meters^2; [m*m / m*m]) 2. Connect multimeter and oscilloscope across the voltage regulator to measure if the potential difference across fan stays within 5% of +12V. 3. Check that humidity of locker interior is no more than 20% of environment humidity using humidity sensor	Status 1. N: Only hit roughly 19.1% interior coverage. 2. Y: 12.33V 3. Y: However, took roughly 50 minutes to dry interior.
Monitoring Unit Requirements 1. Check if load cell can detect at least 250 grams on top of it. 2. Load cell should output different weight values based on different weighted objects. 3. Have the sensor correctly output 0/1 based on whether the door is open / closed. (0 for open, 1 for closed). 4. Android application should be able to monitor locker disinfectant supply levels for 100%, 75%, 50%, 25%, and 0% levels (with around 5% error range). 5. Android application should connect to Bluetooth transceiver within 20-30 feet of locker.	Verification 1. Connect load cell to load cell amplifier. Have load cell amplifier output to LED. Place 250-gram object on load cell. See if LED lights up when object is placed. 2. Connect load cell to load cell amplifier. Place 250-gram object on load cell and check program if 250- grams is read correctly. Place 500-gram object on load cell and check program if 500- grams is read correctly. Place 750-gram object on load cell and check program if 750- grams is read correctly. 3. Connect output of sensor to breadboard and use a simple LED to check if sensor correctly outputs 0 if open and 1 if closed.	Status 1. Y: Detected iPhone 12 with case at 172 grams. 2. Y; Tested at 25%, 50%, and 75% of 1890 grams due to disinfectant supply container volume. 3. Y: Closed: 4.95V; Open: 0V 4. Y: Application updated images correspondingly 5. Y

	4. Using Bluetooth terminal application on Android phone along with code on Arduino, check if global variable on Android Studio changes based on disinfectant supply levels. 5. Test on a physical Android phone walking 20 to 30 feet away from locker and checking if connection holds.	