DISPOSABLE NFC BRACELETS AND READER

ECE 445 FINAL REPORT - SPRING 2023

Project # 42

Brennan Eng, Edson Alpizar, Ege Gunal

Professor: Olga Mironenko

TA: Zicheng Ma

Contents

1 Introduction

1.1 Problem

Waterparks have an issue with optimizing their security, sales, and customer experience. The first example of a problem is customers who manage to sneak past staff and get in for free. There is no easy way to discern between a paying customer and someone who has just snuck in with a fake or old paper armband and even then there is always human error in identifying them. Another problem is if a consumer is currently in the water and becomes hungry they must travel back to their locker/chair to get their card/cash, go to the food stands to purchase the food, go back to their locker/chair to safely re-stash their belongings after eating, and only then can they finally go back to the water. Another niche example of a problem is if a parent loses their child within the park; usually there is no log of where to narrow down the search for the kid or any way to determine their last known location beyond observation.

1.2 Solution

Our solution involves constructing a system that uses cheap, disposable, and reprogrammable NFC bracelets that can be scanned with an NFC chip reader. The solution has two purposes: user-sided actions (to improve user customer experience), and business-sided actions (to optimize profits and security). The purpose of the NFC bracelets for the user is to integrate a seamless low-latency experience for the consumer which takes care of access, payment, and other services if requested. On the other hand, the purpose of the NFC bracelet for the business is that invaluable data that is received from each interaction of the user. Depending on the use case, the NFC chip reader will then access a custom built consumer database that will carry out authenticated user-sided actions. The applications from a business perspective go so far as to use big data to determine localized traffic and popularity of events in a quantitative manner. In addition to the reader and the bracelets, the goal is to create a display of utility by attaching the ultrasonic sensor system onto a door frame ensuring that the count of users entering the site in the database ends up matching that of the ultrasonic sensor.

1.3 Visual Aid

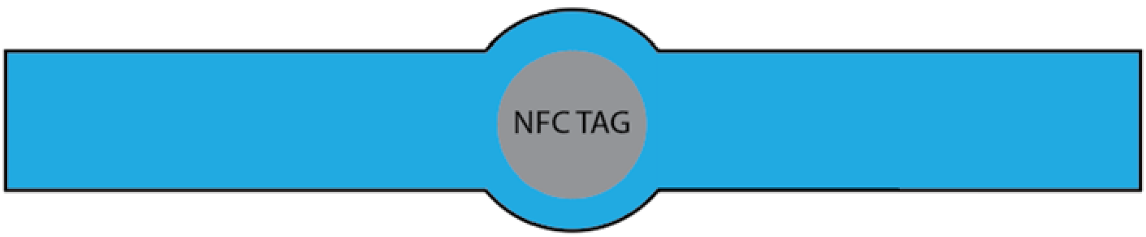

Figure 1: Silicon band with area to insert NFC tag

In most establishments the usage of NFC tags is usually in the shape of a keychain or card. For our project we will be using reprogrammable NFC tags that are inserted to Silicon bands that users will wear when interacting with different locations at an establishment. The NFC tags will be programmable by a server system that will be controlled by the workers of the establishment. The information written to the tags will vary from establishment to establishment. In our project we will use a Raspberry Pi to hold a local server that will also contain the NFC writer.

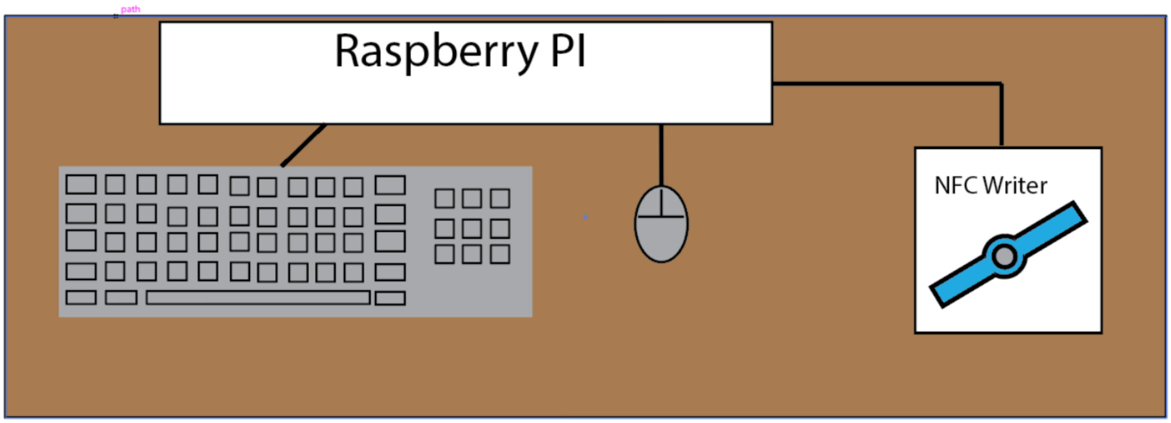

Figure 2: Example setup of server subsystem in the process of writing to NFC Band

We will be giving each NFC tag in our project a unique Identification Number. This ID will be stored in the local server and be used for the purpose of opening a door. The user with the NFC band will interact with a NFC reader system that will do two things: it will either unlock the door the reader system is attached to or will not.

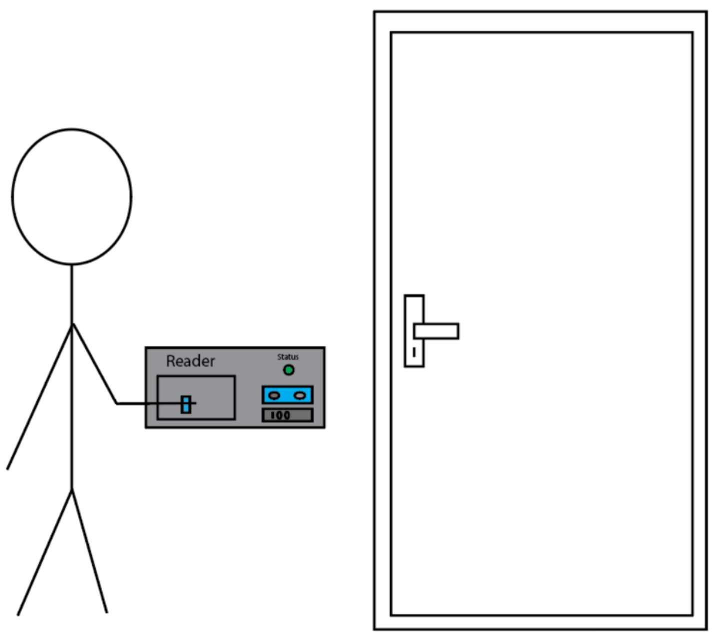

Figure 3: A different implementation of our NFC door opening mechanism, where it is used to open an office door.

1.4 High Level Requirements

We will consider our project to be successful if:

1. Our NFC bands need to check if a user has access with a millisecond-level latency.
2. Our NFC bands need to detect when someone with/without access attempts to pass through checkpoints via sensors, and store this information in a central database that gets cleared at regular intervals, chosen by the employer.
3. Our entry checking mechanism needs to count the number of people who enter, and alert when someone follows another person without tapping their NFC band. This will be done via the sensor subsystem.
4. Our NFC bands need to be linked to a central database linked to user accounts, to which food stalls can send the billing information.

2 Design

2.1 Physical Design

The most important part of our project is the NFC reader system. In our project design we will demonstrate our reader system by attaching it to a door opening mechanism that will also contain a sensor subsystem that will be used to keep track of the number of people that enter said door. Our NFC reader will contain a status LED that will be used to signify whether or not the user will be allowed to enter. If the user scans their tag and the reader recognizes the user the LED will turn green allowing them to enter. When the user enters through the door the ultrasonic sensor will keep track of the people that enter. This sensor will verify that only one person has entered. We will also have a LED display that will show the number of people that have entered said location. If there is a disparity in the number of people that enter per scan then the person that last scanned will be liable for the people that entered without permission.

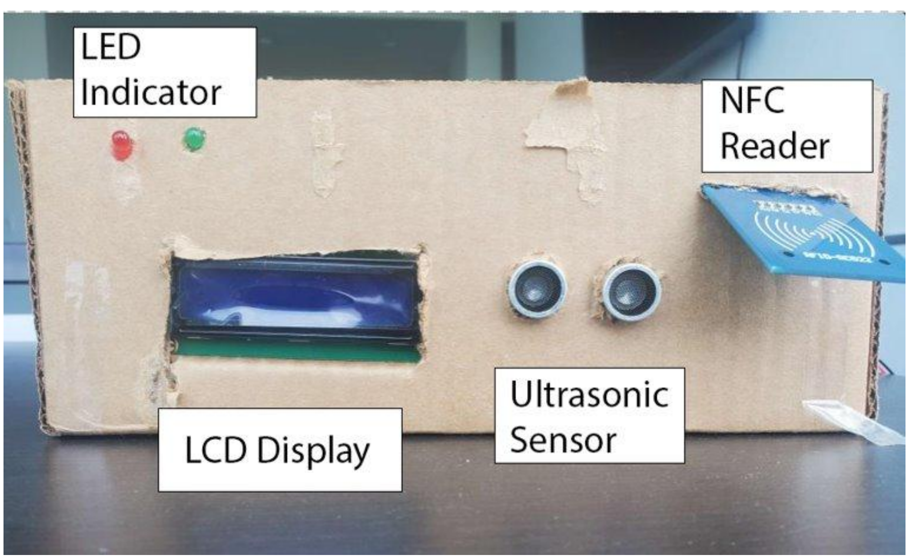

Figure 4: LED Display will display the amount of people that were detected.

The NFC reader and the local server will communicate wirelessly to determine if the user who scans is registered or not.

2.2 Block Diagram

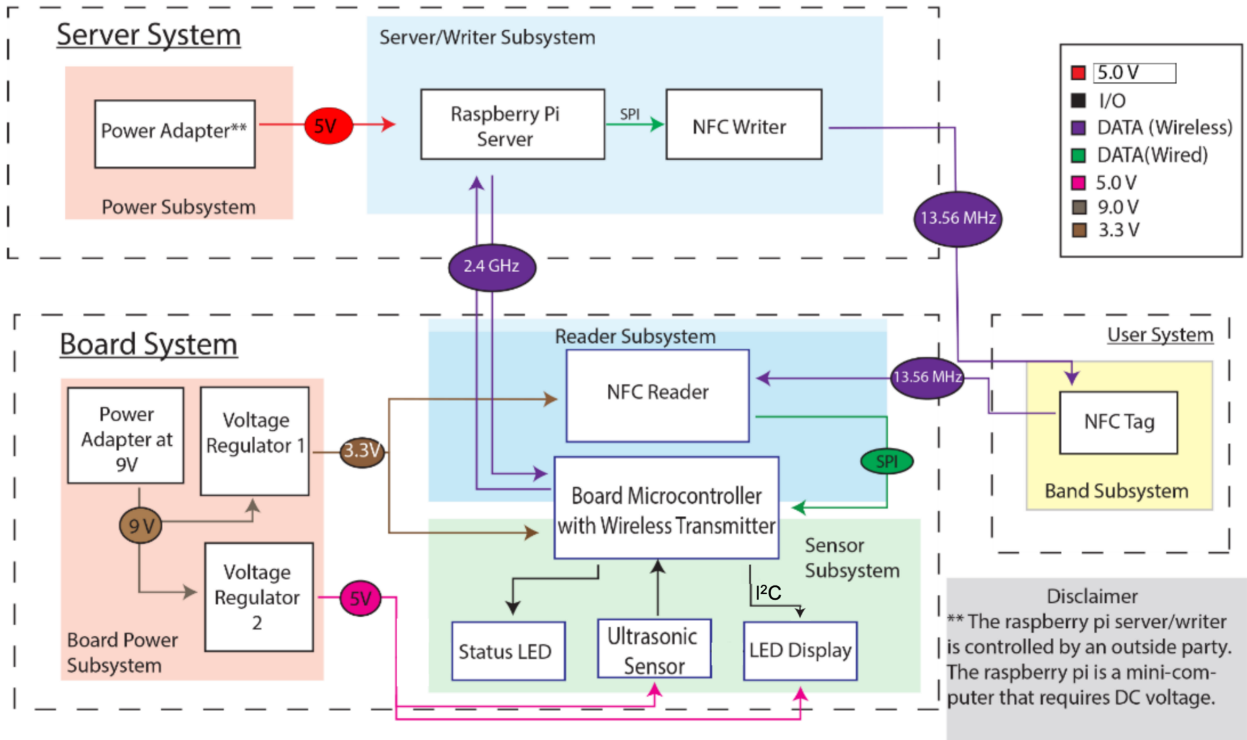

Figure 5: Block Diagram

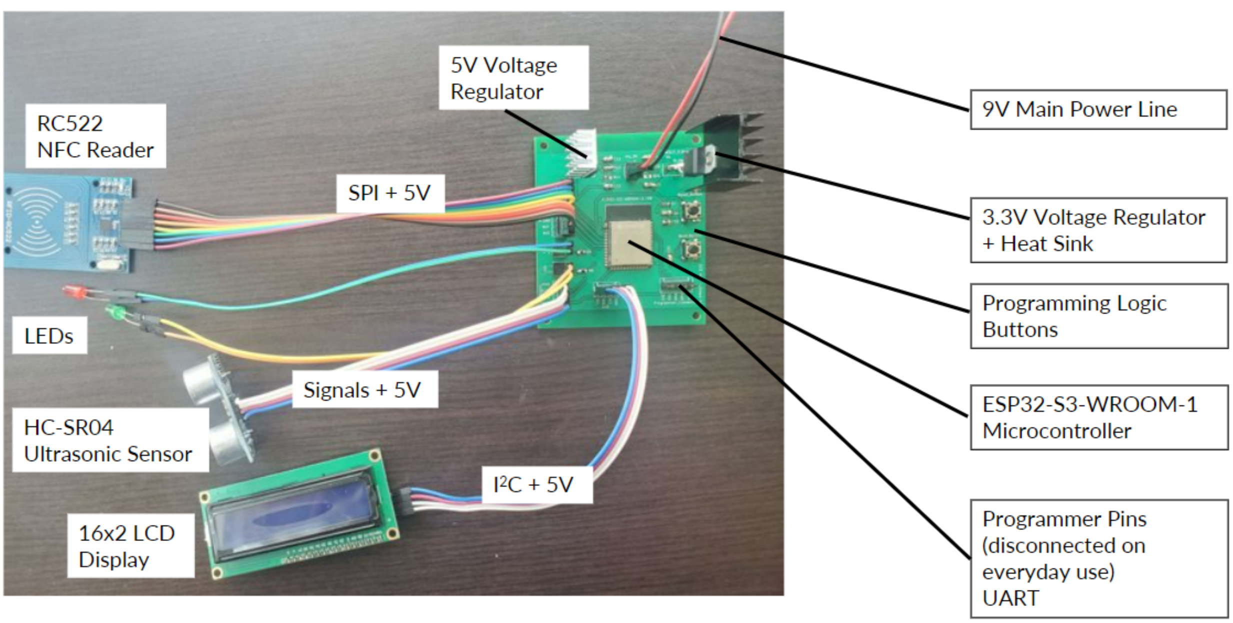

Figure 6: Annotated Product PCB

2.3 Subsystem Overview and Requirements

2.3.1 Server Power Subsystem

The server power subsystem is composed of a Type C power supply. This subsystem is responsible for powering up the Raspberry Pi that will hold our local server.

Requirements	**Verification**
1. Turn on the Raspberry Pi	1. Standard Raspberry Pi Power Adapter

Table 1: Server Power Subsystem - Requirements

2.3.2 Board Power Subsystem

The board power subsystem is responsible for powering up both the Reader and Sensor Subsystems. This will be accomplished by using an adapter connected to a standard wall outlet. The components in sensor subsystems will require 5 Volts to be functional, and the microcontroller and NFC reader require 3.3 Volts. In order to properly power up our components we will have to use voltage regulators on the 9 Volts input to acquire 5 Volts and 3.3 Volts.

Requirements	**Verification**
1. Convert the 9 Volts from adapter to 5 Volts for our ultrasonic sensor and LEDs, 3.3 Volts for NFC reader and microcontroller 2. Voltage regulators should not reach a temperature that's above their temperature rating.	1. Connect the laboratory power supply to the main power line. 2. Turn on the power supply at 3.3V and check for overheating on the 3.3V regulator. Also measure voltage drop between input and output pins of the regulator. 3. Change the power supply to 5V and repeat the above step for both voltage regulators.

	4. Change the power supply to 9V and repeat the above step.

Table 2: Board Power Subsystems Requirements

2.3.3 User System

The user system is a NFC tag attached to a silicon band. The tag will contain user and identification information. This system will interact with both the reader and writer subsystems. The NFC tag will be programmed via the writer subsystem where the raspberry pi will be used to write into the tag. In order to access a particular area the user will have to have their tag be read by the reader in order to allow access to a particular location.

Requirements	Verification
1. The NFC tag should be able to contain the information we program it to have.	1. Reader/Writer from writer subsystem will verify the intended information got written on the NFC tag. 2. When the NFC tag is read the status LED on the reader system should light up green if the user is registered in the database. 3. If the tag is not registered the status LED will light up red.

Table 3: User System Requirements

2.3.4 Server/Writer Subsystem

The server and writer subsystems are combined parts in our project. This is the case because the Raspberry Pi will act as the local server. The server will contain a database in SQL that will use a unique identification number for every user in the database. With that unique identification number the host of the server will be able to access any necessary information that is contained in the user NFC bands. The server will be powered by the server power subsystem. In our project design the writer subsystem is connected directly onto the Raspberry Pi, since the raspberry pi is a computer and can be used to write the information we want onto the NFC tags. The writer subsystem will then write the user information onto the SQL server. The server subsystem will also interact with the reader subsystem whenever a user enters a restricted area. The server will figure out if the NFC tag that was read by the reader subsystem has access to a particular location and return a true or false signal that will be indicated by the status LED on the reader subsystem.

Requirements	**Verification**
1. Store user information in a database. 2. Receive entry logs (communicate wirelessly) from reader subsystems and store them in a central database.	1. Be able to print the contents of the database and see the user input information. 2. We can print the contents of a database for a particular user and see when they scan their NFC band.

Table 4: Server/Writer Subsystem Requirements

The options the user has to manipulate the data on the server can be seen in the flowchart:

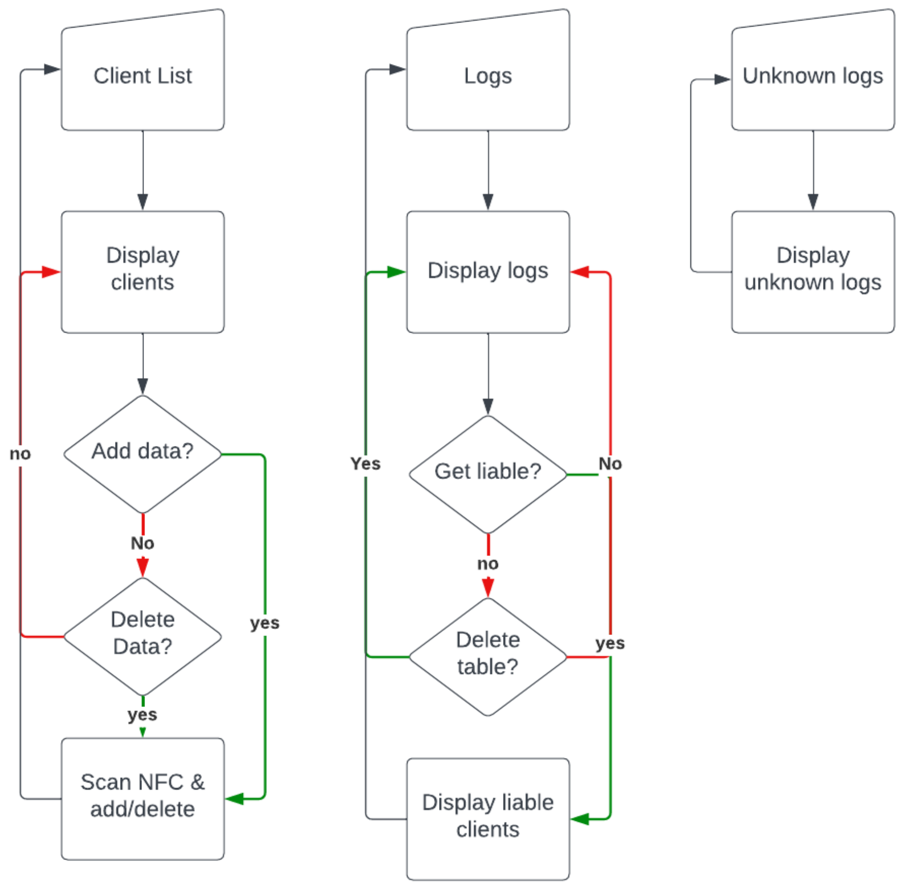

Figure 7: Server Flowchart

Each table in the SQL has various options it can execute as shown above. All of the options that a particular table has can be seen on the bottom right of the GUI window. In the example shown below the user is viewing the client list table and has various options they can execute, the options include the option to add and delete data as well as deleting the entire client list.

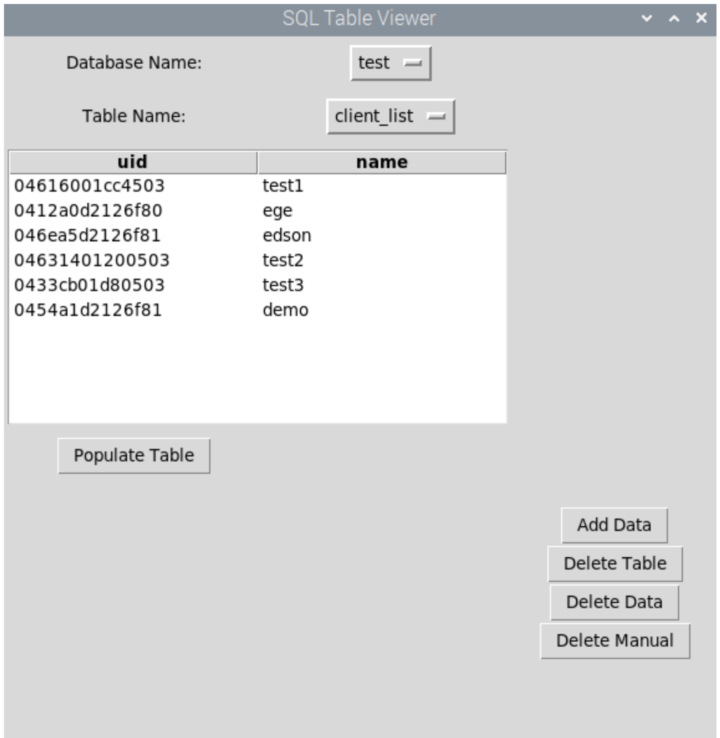

Figure 8: Client List SQL Table

2.3.5 Reader Subsystem

The reader subsystem will contain the wireless transceiver, NFC reader and microcontroller. This subsystem will be powered up by the board power system since all the components require five Volts instead of nine. The reader subsystem will interact with the user system whenever a NFC tag is read. The reader will isolate the unique identification number and send it to the server system to see if the UID is a valid entry. If the user entry is valid then that UID will be stored onto on chip memory to reduce latency whenever the user scans their tag in the same location. The reader will also log information and send it back to the server subsystem. The reader will also communicate with the sensor subsystem. The sensor system will send the number of people the ultrasonic sensor detects to the microcontroller via SPI and then send that information back to the server via php [3].

Requirements	Verification
1. Receive NFC UID from user bands and share it with the board microcontroller via SPI. 2. Share UID information between the local server and the reader wirelessly. 3. Reduce latency when the user scans for the first time and when they scan in the same spot multiple times.	1. We can print the UID of the NFC that is scanned onto the serial monitor using the Arduino IDE to verify that the NFC reader works. 2. We can print the UID that is shared between the reader and the server onto the raspberry pi and the Arduino IDE and verify that they match. 3. We can print out the time it takes to search for a UID in the database vs. the time it takes to search for a UID in the microcontroller cache.

Table 5: Reader Subsystem Requirements

In order to communicate information between the ESP32 microcontroller and raspberry pi server we had to use HTTP requests to send the data wirelessly. We wrote PHP scripts that would get all the variables and values necessary to make the adequate request. The image below shows a proper HTTP request URL where we will add an entry to our logs table in the SQL database[2].

```
https://localhost/php_scripts/add.php?uid=123794719823&location=door_1&counter=1&time=23:32:55
```

Figure 9: Example log request from door_1, for 1 person, and UID of their band and current time.

2.3.6 Sensor Subsystem

The sensor subsystem will contain a status LED, an LED display, and an ultrasonic sensor. This part of the subsystem will also be powered by the board power subsystem. The sensor will use an LED indicator that lights up green if the user who scans is eligible to enter and lights up red if the user is not eligible. That information will be communicated between the reader and server subsystem. This will be possible because the sensor and reader subsystem share a microcontroller. When a person walks through the ultrasonic sensor when the status LED is light green it will count how many people enter and send that information to the microcontroller so it can be logged onto the server. If two people were detected entering when only a single user scanned then the owner of an establishment will know who is liable.

Requirements	**Verification**
1. Display whether someone has access via LEDs, as determined by the controller. 2. Display the number of people who have entered on the LED display. 3. Log the number of people that the ultrasonic sensor detects at the time of scan onto the server.	1. Have two NFC tags with UIDs. One of the tags will be registered onto the database and the other will not. The one that is registered will make the green LED light up. The unregistered tag will make the red LED light up. 2. Have multiple people walk by the sensor and after a 5 second period it should display the correct amount of people that walked by. 3. Write some code that will print onto the raspberry pi the time and the number of people that entered at the time of scan.

Table 6: Sensor Subsystem Requirements

The flowchart for how the ultrasonic plus reader system works is as follows:

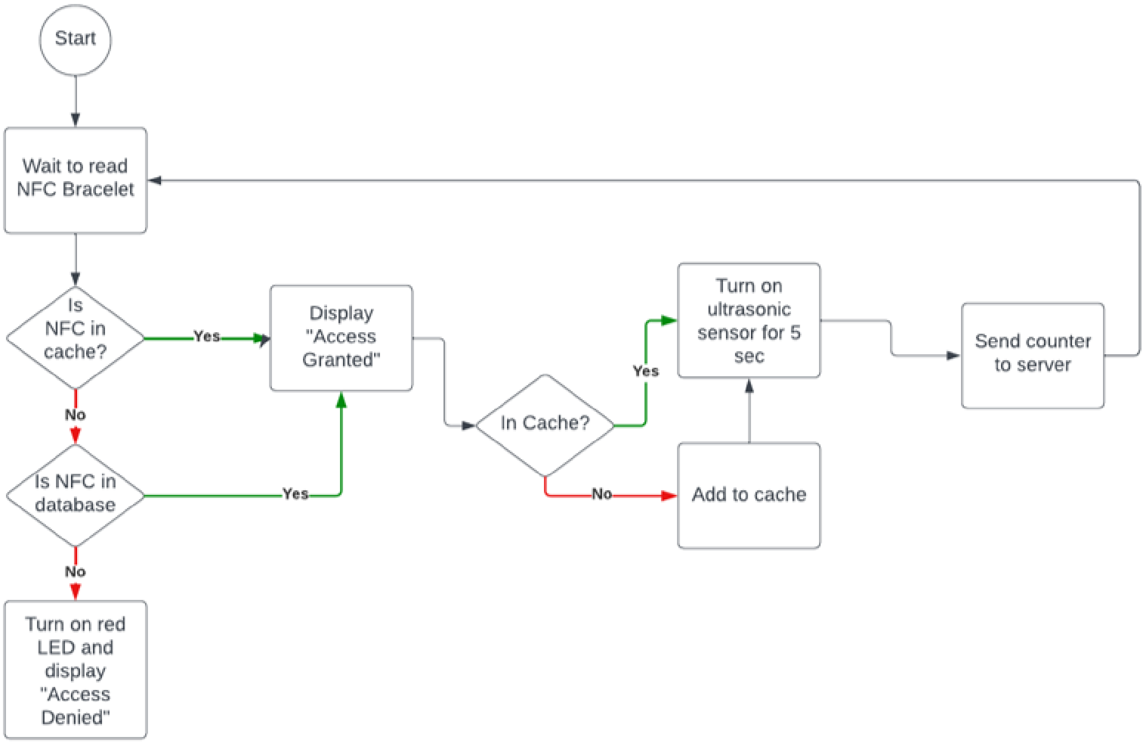

Figure 10: Board system flowchart

The algorithm that was used to allow the ultrasonic sensor to detect people walking through a door required an understanding of how these types of sensors work. Our code took inspiration from a tutorial guide[1] but the actual code that could count people was written by us. We decided to use a double flag algorithm that would determine if the counter would increment or not. The following code shows how this flag system works.

```
if (check == true && detect == false) {
    //detecting a person for the first time
    detect = true;
    start_counter = counter;
}

if (check == false && detect == true) {
    // person is done walking
    final_counter = counter;
    delta_counter = final_counter-start_counter;
    Serial.print("Delta Counter: ");
    Serial.println(delta_counter);
    if (delta_counter < 400) {
      detect = true;
    }
    if (delta_counter > 400) {
      detect = false;
      person_counter += 1;
    }
}
```

Figure 11: Ultrasonic sensor code

The two flags we used were a “check” and “detect” flag. When the sensor was initially turned on the detect flag would be set to false. The check flag would be determined by a function that would determine if the distance the sensor would read is less than or equal to the width of the hallway that sensor would be placed in. If the distance is smaller than the width of the highway then the check flag would become true and remain as such until the person walking by completely passed by. When the check flag first returns true and only if the detect flag was false, the detect flag will become true to signify that a person is walking by. Once the person finishes walking by the check flag will turn false and if the detect flag is still true then this will tell the microcontroller that a person has finished walking by and thus increase the counter that will contain the number of people that walked by during a scan.

3 Costs

Labor: $40/hr * 10 hr/week * 3 group members = $1200 in labor costs

Name	Description	Manufacturer	Part #	Quantity (units)	Cost ($)
NFC Round Cards	Cheap and disposable NFC chips embedded into a plastic card	Yanzeo	B08Z7P7L3R	20	8
NFC Tag Wristbands	Cheap and programmable NFC wristbands meant to be used as a final product alongside reader and writer	YARONGTE	NTAG215CA RD-100	5	12
Reader Module Kit Mifare RC522 Reader Module	NFC Reader IC to be used with the PCB	SunFounder	MFRC522	1	8
NFC HAT for Raspberry Pi	Writer/Reader extension for RPI	Waveshare	PN532	1	28
Raspberry Pi 4	Used to transmit information between NFC tags and pcb	Raspberry Pi	Raspberry-PI-4	1	55
Ultrasonic Sensor	Used to physically verify that the number of entries matches up with the people entering	Sparkfun	HC-SR04	1	4
Microcontroller	Used for controlling the reader and sensor subsystems	Espressif Systems	ESP32-S3-WROOM-1-N8	1	4
USB to TTL Downloader	Used to program the microcontroller	HiLetgo	CP2102	1	8
9V Power Adapter	Used to power the board system	TB Tbuymax		1	10

5V fixed voltage regulator	Used to power ultrasonic sensor and LCD display	STMicroelectronics	L7805CV	1	1
3.3V fixed voltage regulator	Used to power microcontroller and NFC reader	STMicroelectronics	LD1117V33	1	1
Misc	PCB pin connectors, resistors, capacitors, jumper cables	multiple sellers	multiple parts		12
Total cost	Entire product, 1 server 1 board			131 +band	insert

Table 7: Project costs

Adding everything together, we get a total cost of $143 beside labor costs. With Labor costs it is $1343

4 Conclusion

4.1 Summary

The project was successful in completing all of our High Level Requirements. Our Raspberry Pi SQL server was established successfully and was capable of running and managing any data we required (High Level Requirement 4). On top of the Server, the capability of reading and writing to the NFC bracelets worked in tandem with the server so now it could enable the NFC reader to record the bracelet information without any manual insertion into the server. Furthermore, the NFC reader security gate was successful establishing a connection with the Raspberry Pi server so that information could be shared when a user scanned their bracelet at the security gate(High Level Requirement 2). The NFC reader security gate also held a local storage cache of recent valid UID scans for low-latency scans (High Level Requirement 1). The security gate also successfully worked by activating the ultrasonic sensor and green LED after a valid scan. The ultrasonic sensor was able to detect if multiple people pass through the gate for only one scan (High Level Requirement 3). While all of our High Level Requirements were completed, our PCB was unfortunately unsuccessful in being able to light up one of our LEDs. Because of this revelation, the project was forced to forgo the PCB and place the components onto a breadboard. We currently believe that the PCB failure was due to a badly soldered voltage regulator frying our on-board microcontroller, and given the time to manufacture another one we would succeed. Other than the PCB, the project was an overall success in completing everything we set out to do.

4.2 Ethics and Safety

One potential exposure in our project is the data aggregation of user interactions within the premise of a business. It is a concern because it presents the risk of exposure of customer privacy if not handled correctly (IEEE Code of Ethics 1.1). These factors may pose a threat to the privacy of others as well as the mishandling of financial data if payment systems are involved. While these devices pose no physical threat to a user or its customer, the database records and stores all interactions of consumers for the business to make more sound decisions based on the business intelligence it has gathered. That being said, while the data belongs to the business, the release of that data may pose a threat to those involved. To reduce the risk of data breaches, a comprehensive cybersecurity system is necessary as well as vetting of businesses who intend to use these products for malicious purposes (IEEE Code of Ethics 1.2). We will also need to ensure anonymity of long-term stored data and further investigate privacy concerns over data we store, such as California State Law that has a strict emphasis on data privacy (IEEE Code of Ethics 1.6).

5 References

[1] Dejan, “Ultrasonic Sensor HC-SR04 and Arduino Tutorial,” *HowToMechatronics*, Aug. 04, 2018. https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/

[2]“Esp32 HTTP GET and HTTP post with Arduino IDE,” *Random Nerd Tutorials*, 27-Oct-2022. [Online]. Available: https://randomnerdtutorials.com/esp32-http-get-post-arduino/. [Accessed: 03-May-2023].

[3]Santos, Rui. "How to Setup a Raspberry Pi Apache MySQL PHP (LAMP) Server." Random Nerd Tutorials, June 25, 2019, https://randomnerdtutorials.com/raspberry-pi-apache-mysql-php-lamp-server/. Accessed May 3, 2023.