Disposable NFC Bracelets and Reader

Electrical & Computer Engineering

Team 42
Brennan Eng
Edson Alpizar
Ege Gunal

May 3rd, 2023

Team Members

Brennan Eng
- Computer Engineer
- UI/Server Integration
- UI/UX Development

Ege Gunal
- Electrical Engineer
- PCB/Hardware Design
- Soldering

Edson Alpizar
- Computer Engineer
- Hardware Integration
- Server Development

The Problem

- Waterparks use disposable paper wristbands for entry
  - Easily replicated
  - Loss of revenue
- Customers sneak in extra people
  - Pay for one ticket and have people reuse the same one
- Lost people are hard to find in large premises (like children getting lost)
  - Lots of manpower to locate lost people
  - Whole park is on the lookout

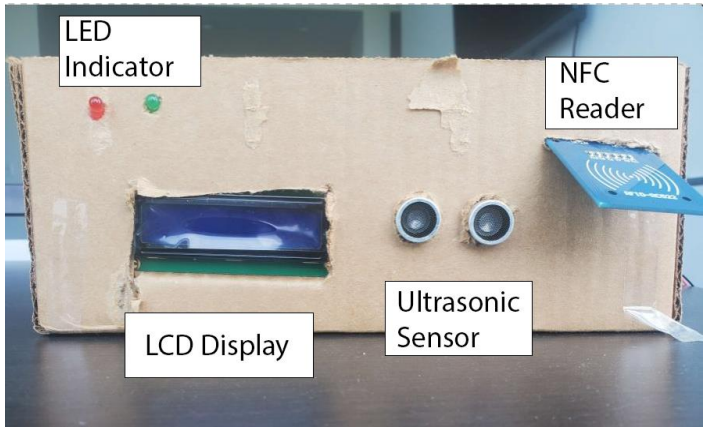
Our Solution

Disposable NFC bracelets with reader enabled security gate

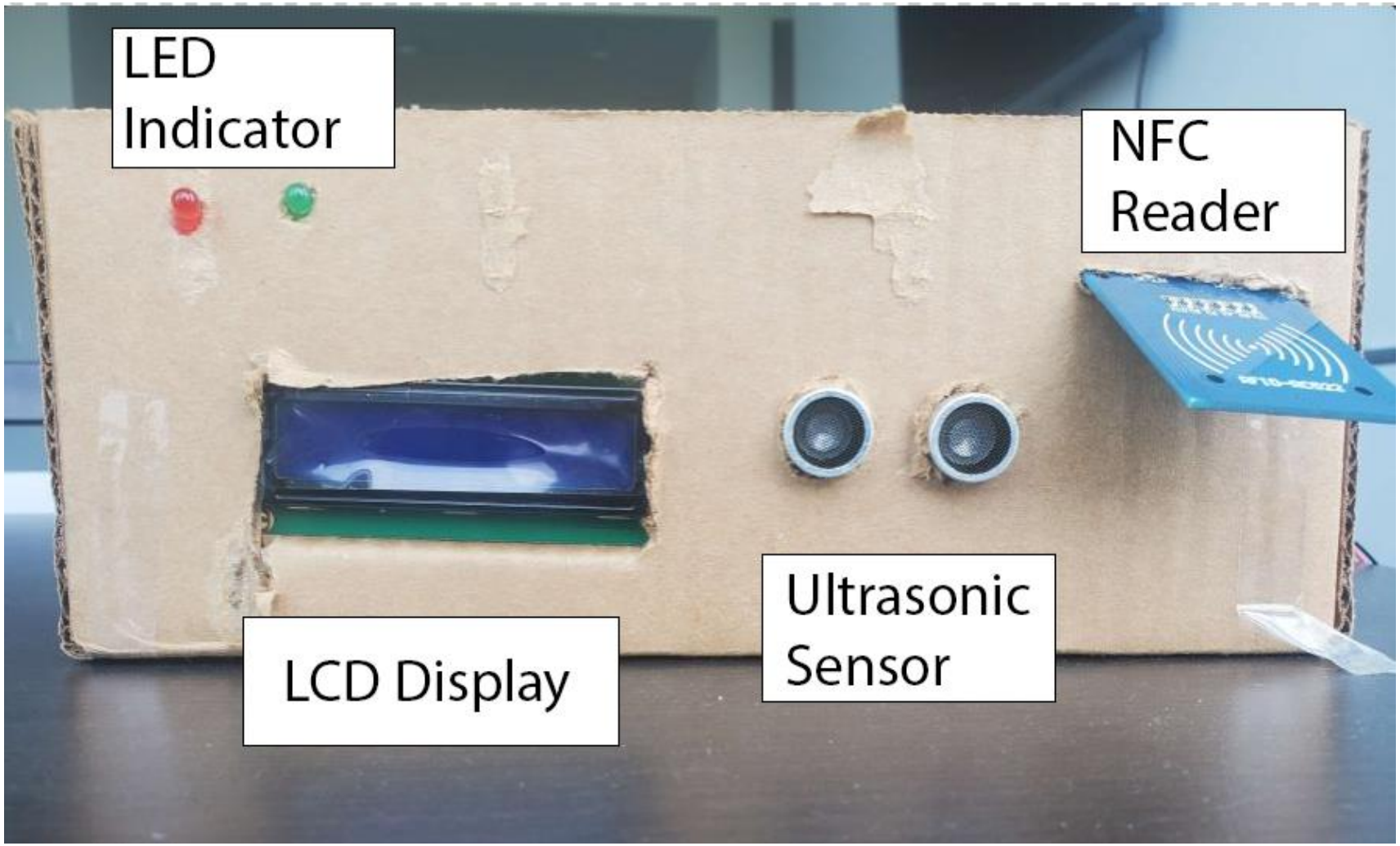
- Unique ID (UID) for each paying customer recorded into central server
  - Replicated bracelets no longer valid to enter
  - Reduces human error
- NFC reader gate security with ultrasonic sensor
  - Flags if more than one person goes through gate per valid scan
  - Physical confirmation for gate security and safety
- Server database that houses all customer interactions within park
  - Lost person: can pinpoint the specific location last entered
  - Reused bracelet: can determine if user is already in park and it attempting to enter without exiting

High Level Requirements

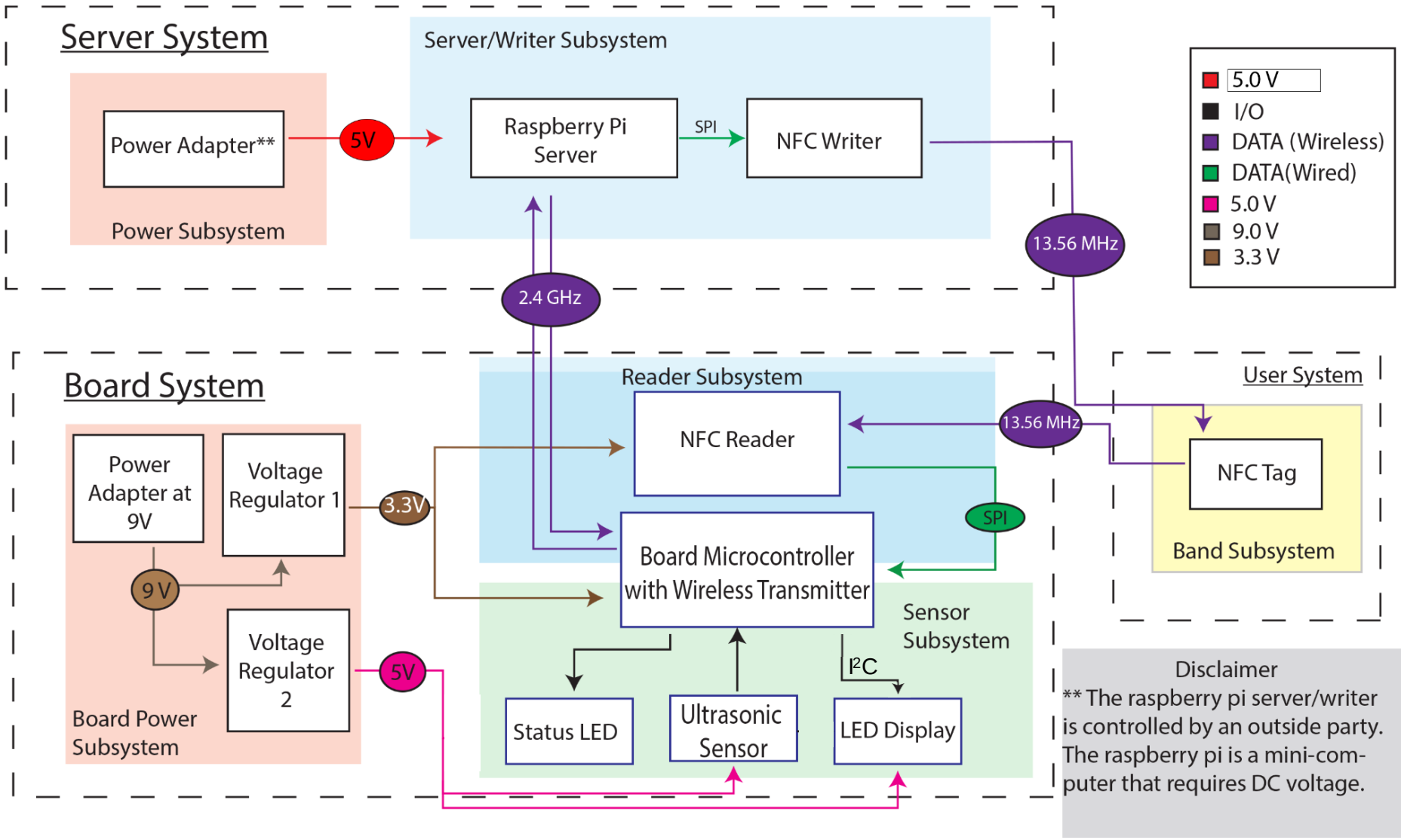
- Our NFC reader need to check if a user has access with a millisecond-level latency.
- Our NFC reader need to detect when someone with/without access attempts to pass through checkpoints via LEDs, and store this information in a central database that gets cleared on a semi-regular basis.
- Our entry checking mechanism needs to accurately count people who enter, and alert when someone follows another person without tapping their NFC band, this will be done via ultrasonic sensors.
- Our NFC bands need to be linked to a central database linked to user accounts.



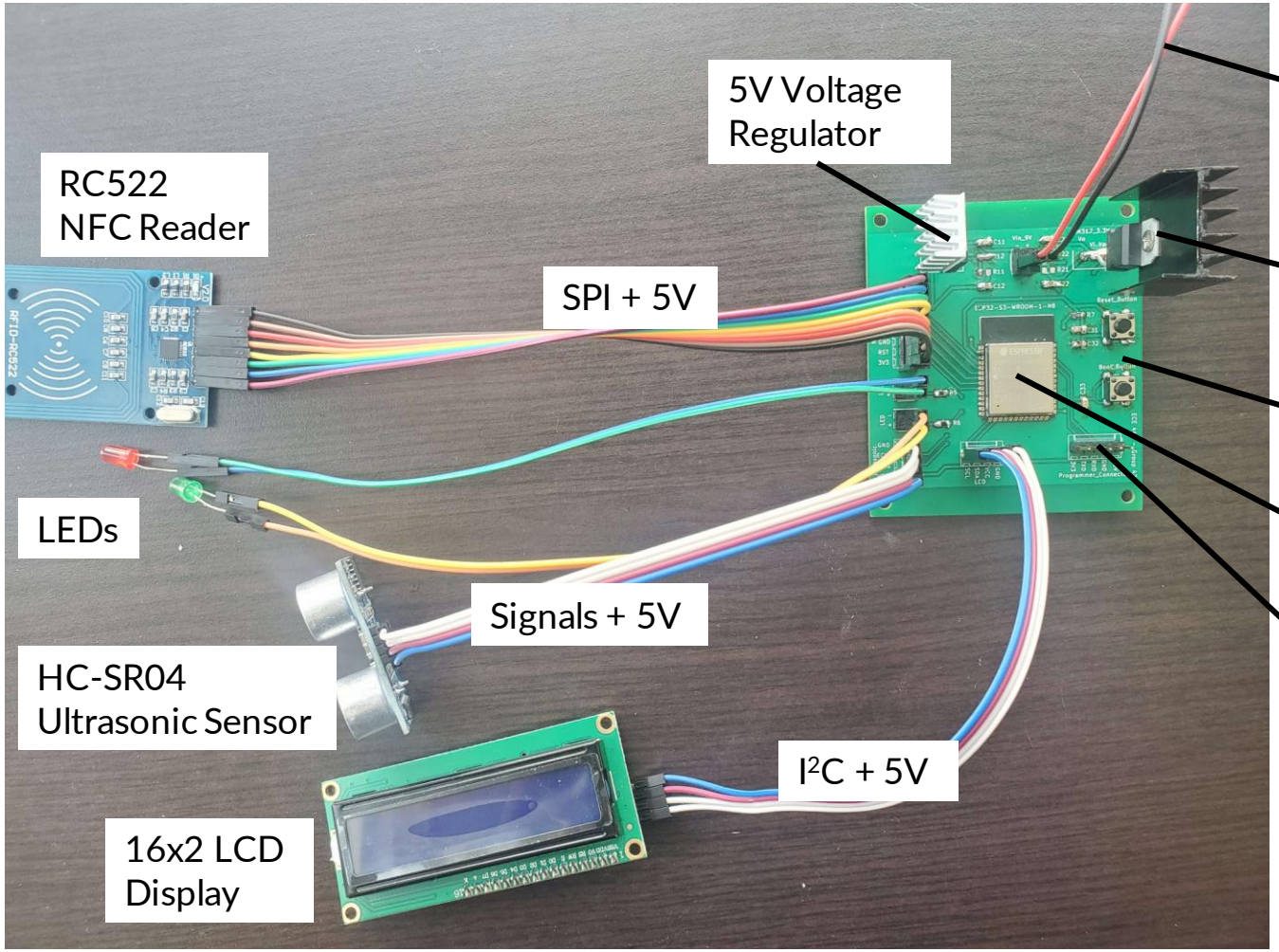
Product Image



Block Level Diagram



Product PCB



9V Main Power Line

3.3V Voltage Regulator + Heat Sink

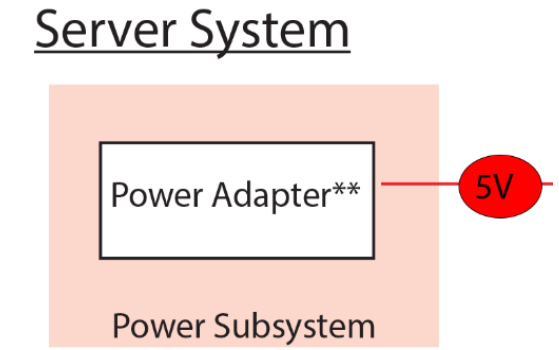
Programming Logic Buttons

ESP32-S3-WROOM-1 Microcontroller

Programmer Pins (disconnected on everyday use) UART

The Components

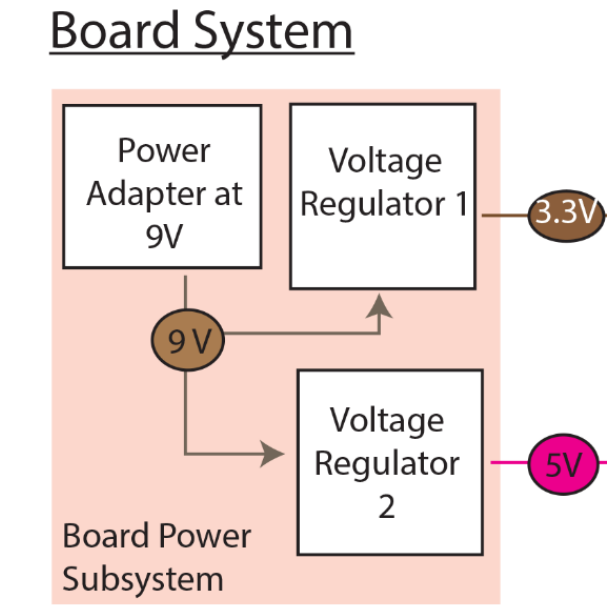
Power Subsystems



Server Power Subsystem

- Raspberry Pi power adapter

Part	Driving Voltage	Max Current Draw
ESP32	3.3V	355mA
NFC	3.3V	30mA
Ultrasonic	5V	15mA
LCD	5V	22.5mA



Board Power Subsystem

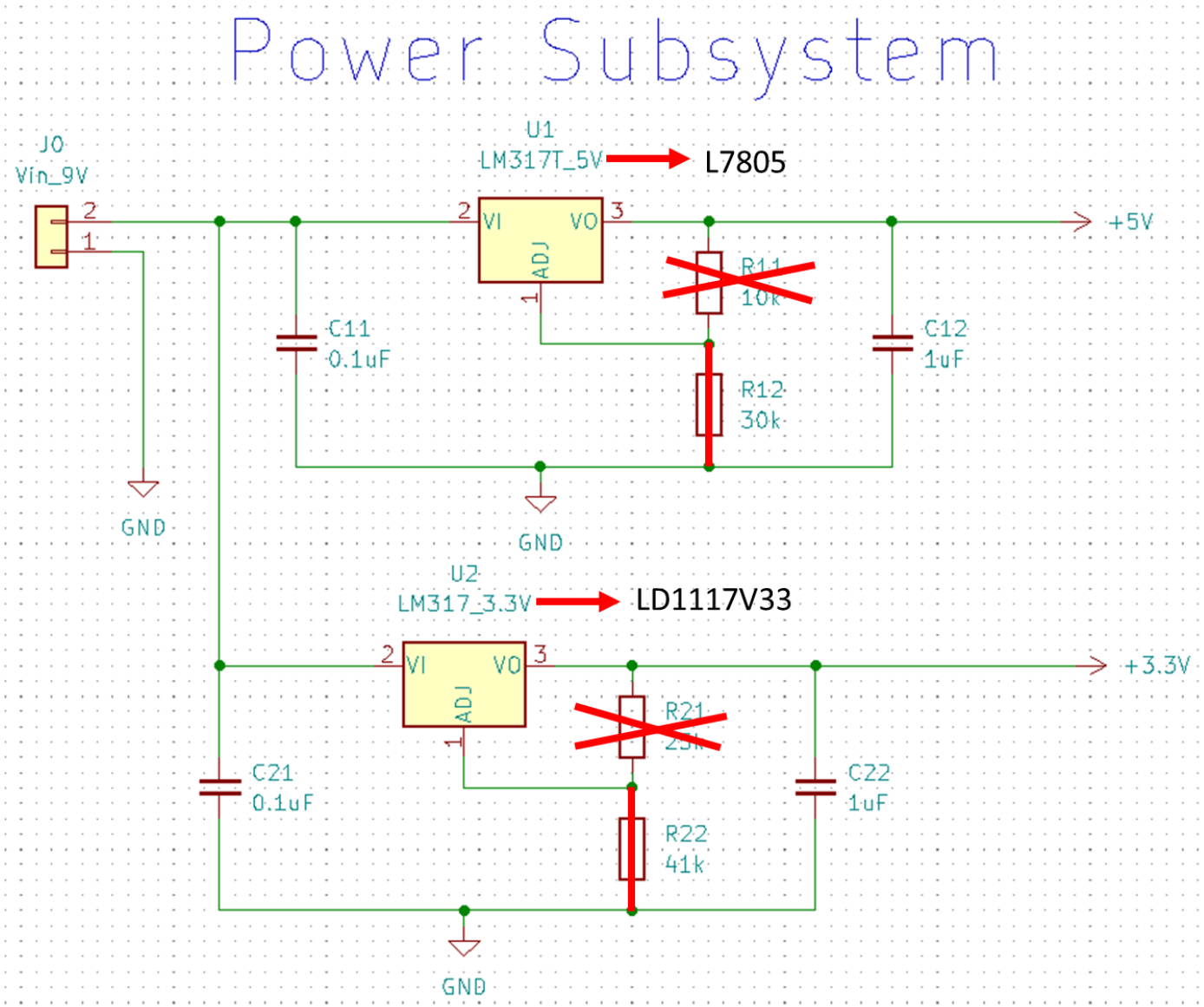
- 2 Voltage Regulators with voltage outputs of 3.3V and 5V
- 3.3V powers the NFC Reader and microcontroller (who powers LEDs)
- 5V powers the ultrasonic sensor and the LCD display

Regulator	Pdrop [W]	Max T [C]
5V	0.15	32.5
3.3V	2.1945	134.725

L7805 (5V) - R(th-JA)=50 C°/W

LD1117V33 (3.3V)
R(th-JA)= 50 C°/W
R(th-JC)= 3 C°/W
=> heat sink! (junction-case-sink-ambient)

Voltage Regulator Schematics



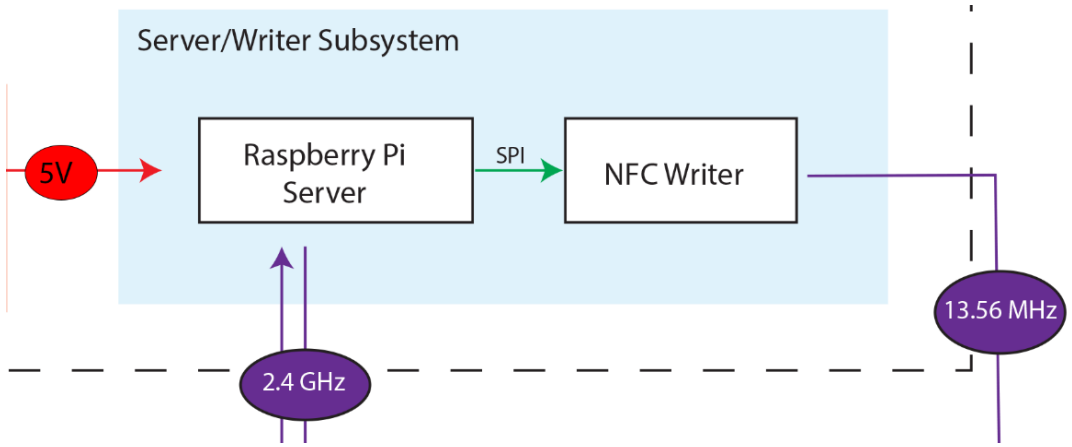
LM317T linear voltage regulator

- Lower R(th-JA) 39 [C°/W]
- No need for heat sinks
- Custom output voltage
- Initial design didn't fit predicted values

LD1115V3 fixed voltage regulator
L7805 fixed voltage regulator

- Higher R(th-JA) 50 [C°/W]
- 3.3V regulator would overheat without heat sink
- Fixed output voltage
- Different pinout, had to get creative with soldering

Server/Writer Subsystem



Raspberry Pi

PN532 NFC HAT

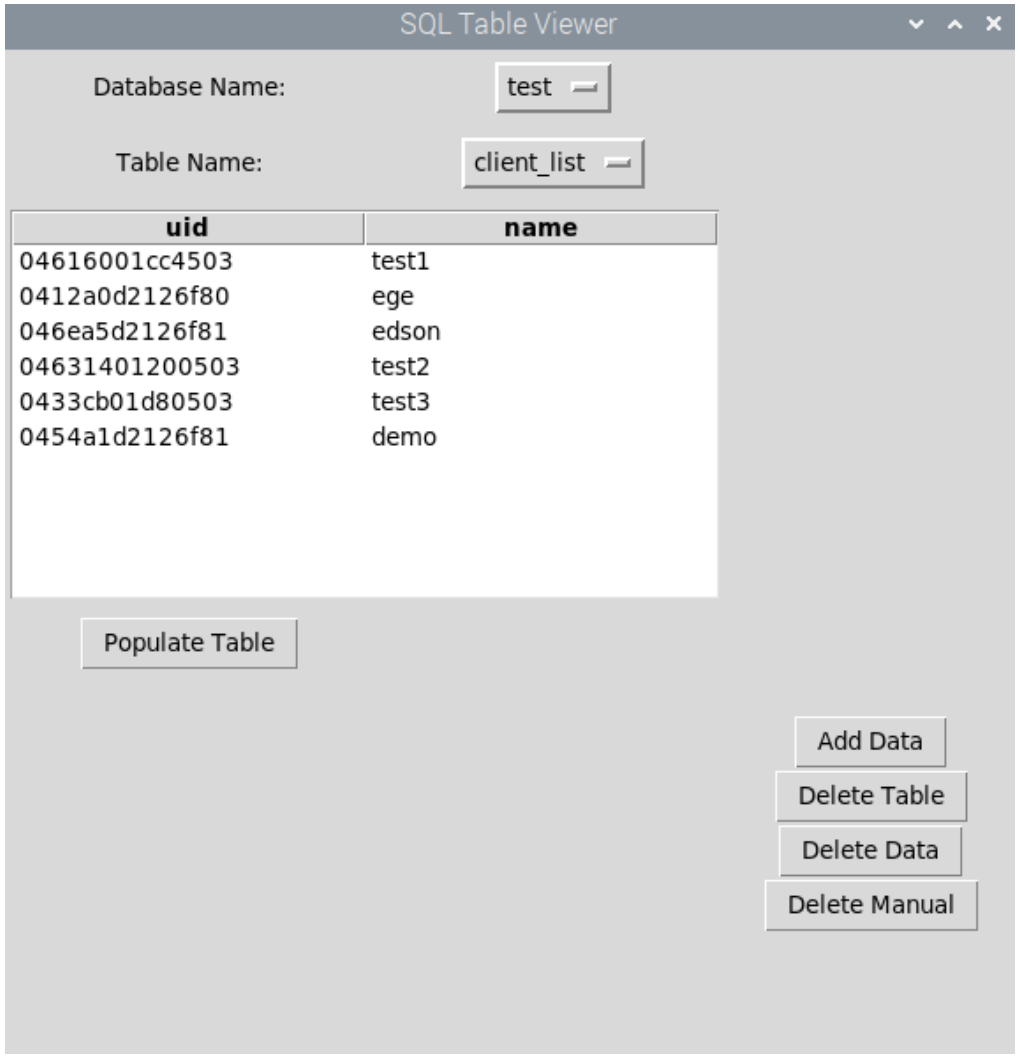
SQL Server (Raspberry Pi):

- Client_list
- Logs (Valid interactions)
- Unknown (Invalid interactions)
- Shares server information with PCB using 2.4 GHz wireless connection

NFC Writer:

- Records 13.56 MHz Wristband information
- Communicates with the Raspberry Pi using the PN532 NFC HAT attachment

Server Example

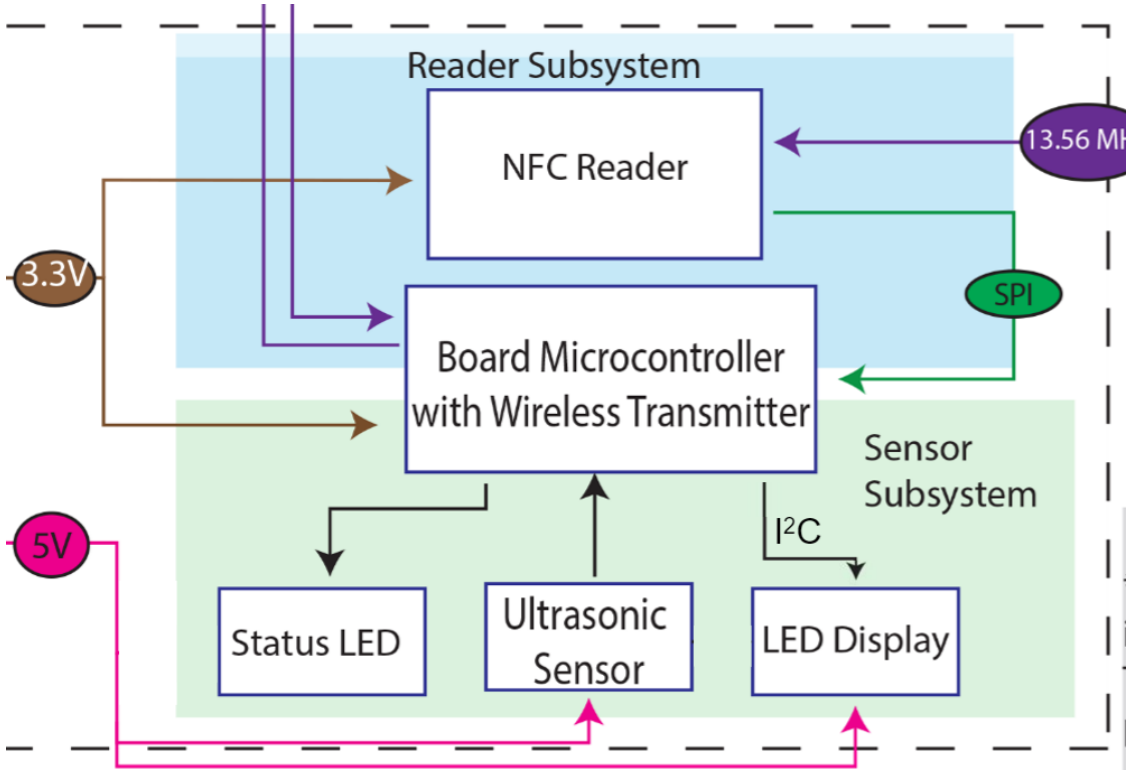


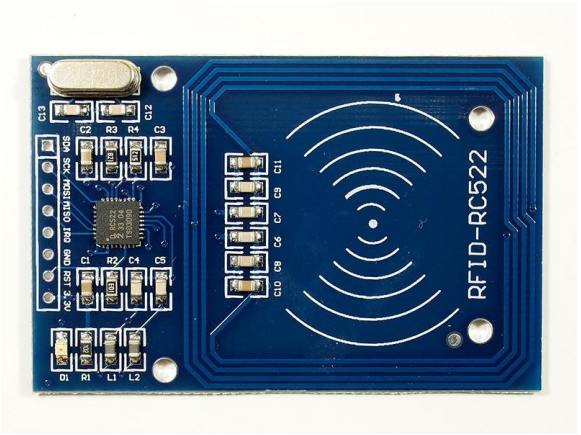
- A GUI was made to navigate the server with less difficulty.
- We can add and delete data as we see fit

```
+----------------+
| Tables_in_test |
+----------------+
| client_list    |
| logs           |
| unknown        |
+----------------+
```

All tables in our database

Reader/Sensor Subsystems

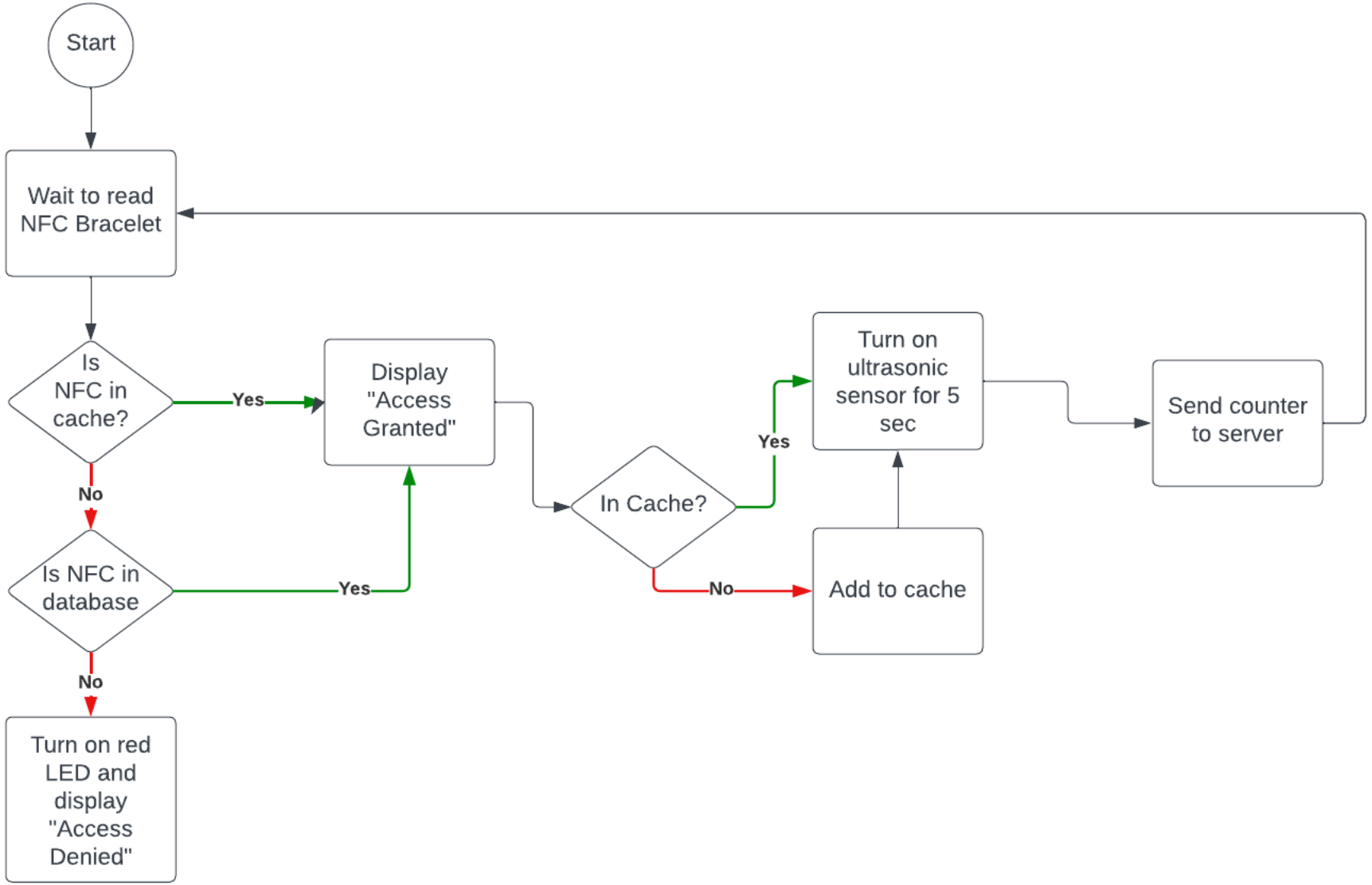




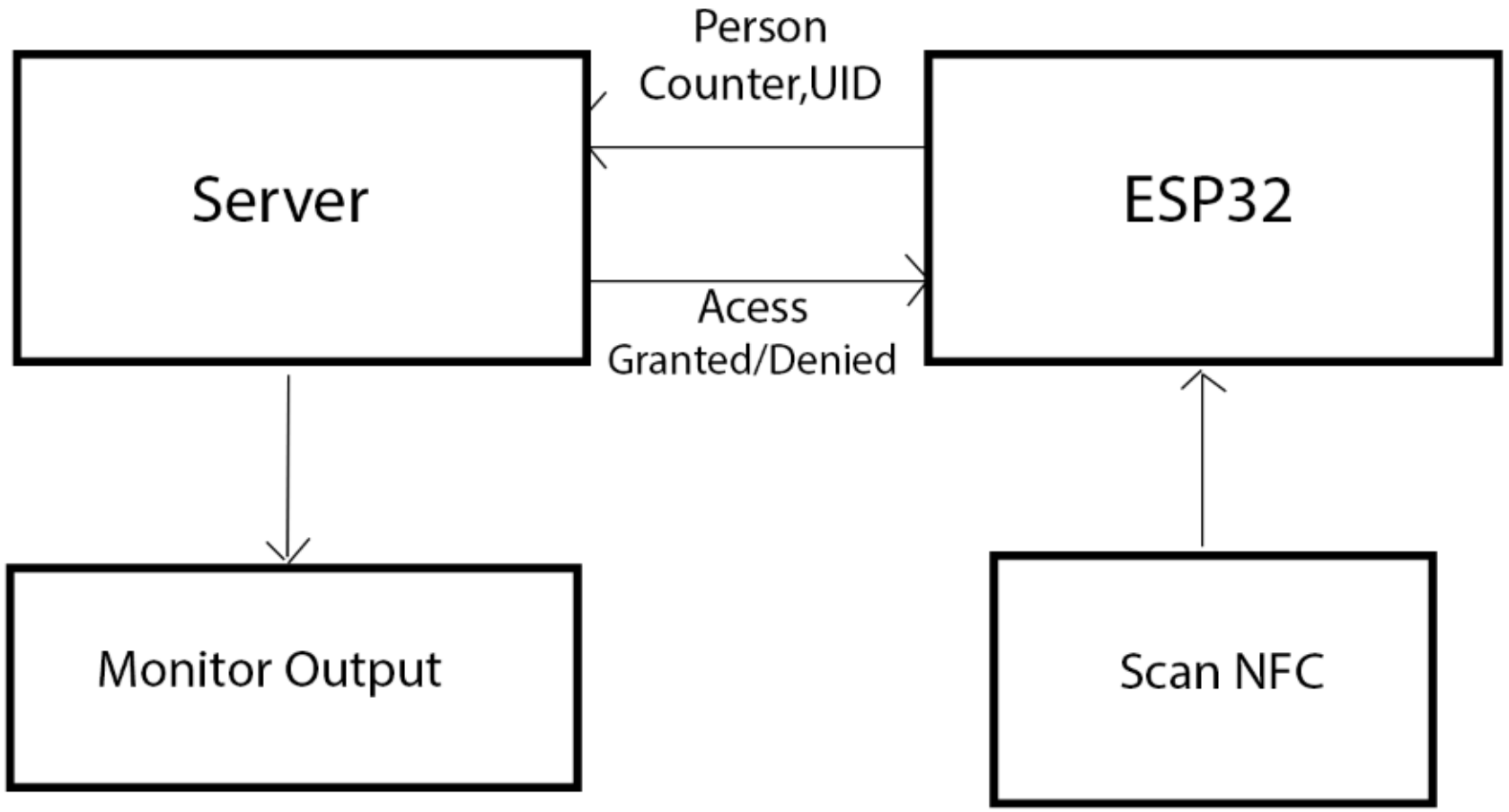
Reader/Sensor Subsystems

Requirements	Verification
Display whether someone has access via LEDs, as determined by the controller.	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.
Display the number of people who have entered on the LCD display.	Have multiple people walk by the sensor and after a 5 second period it should display the correct amount of people that walked by.
Log the number of people that the ultrasonic sensor detects at the time of scan onto the server.	Write some code that will print onto the raspberry pi the time and the number of people that entered at the time of scan.

Sensor Flowchart

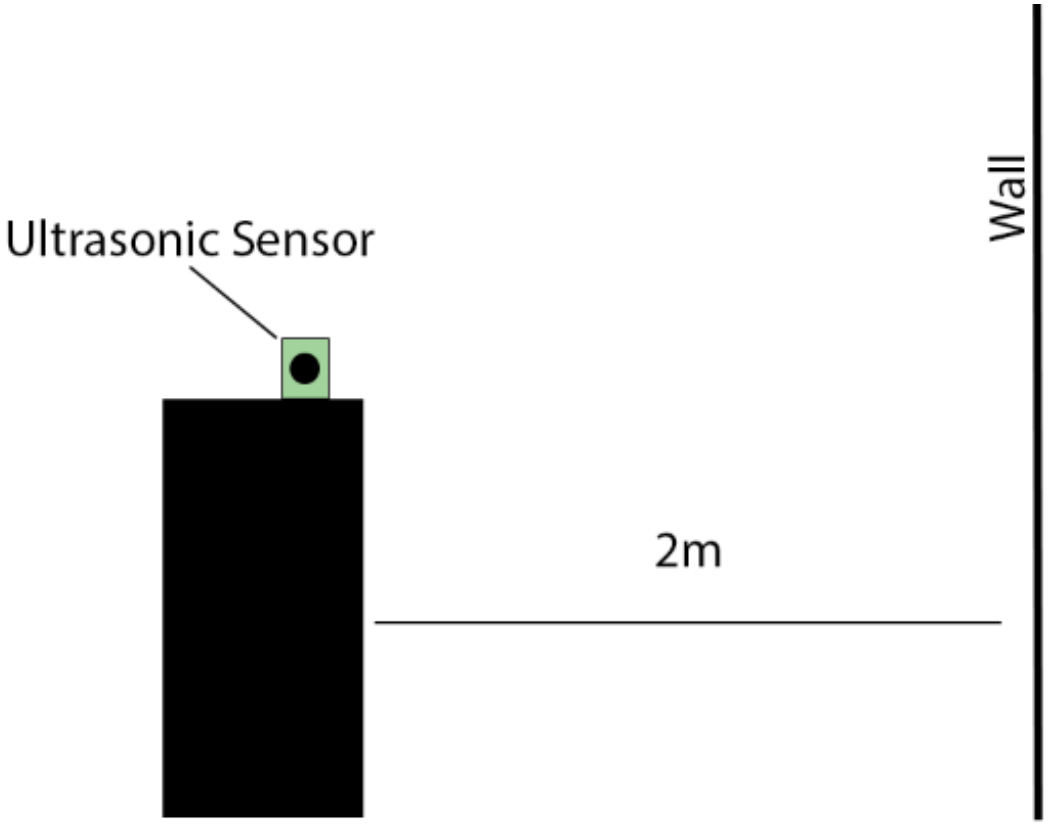


Data Flow

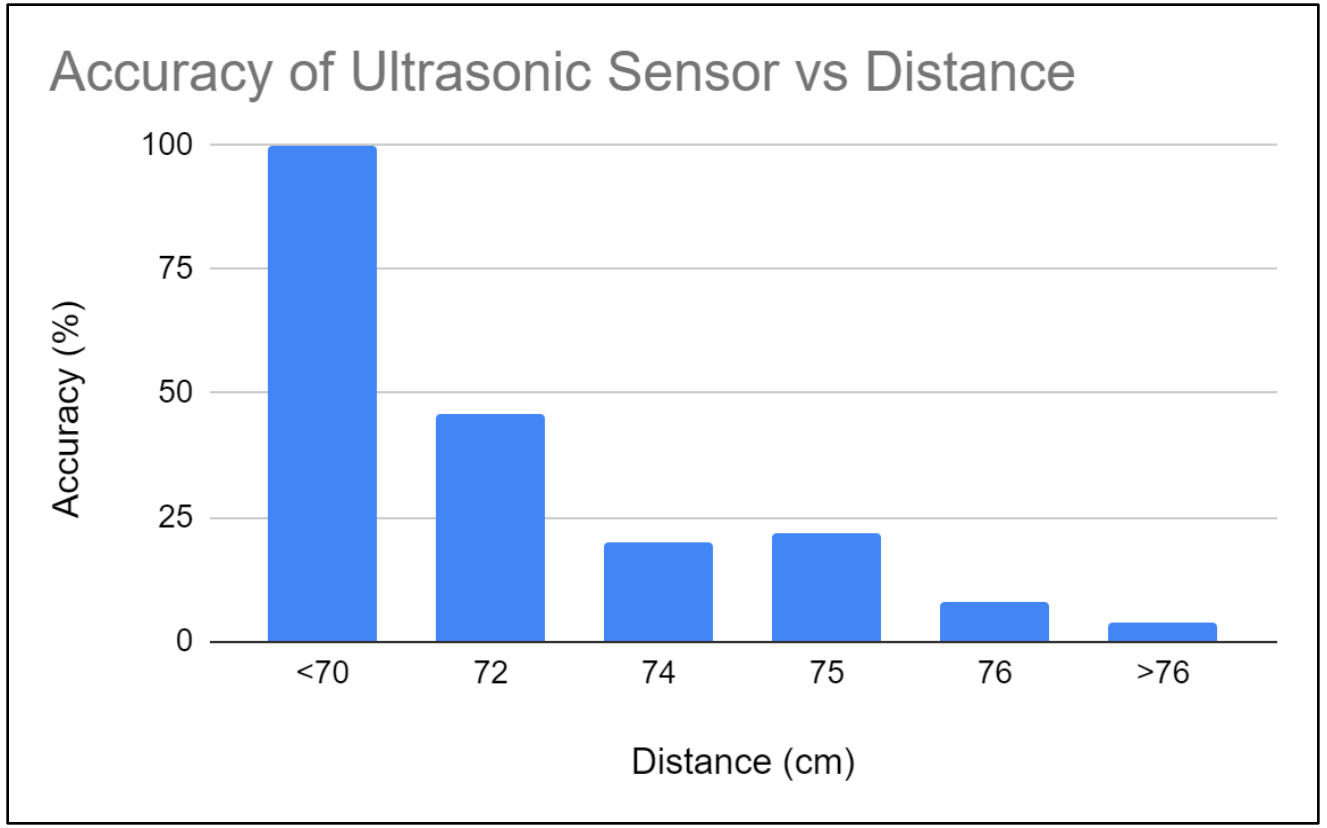

Person
Counter,UID
Server
ESP32
Acess
Granted/Denied
Monitor Output
Scan NFC

Results and Modifications

Calibration



After 50 Trials



Successes and Challenges

Successes

- Achieved all our high-level requirements.
- Got our project to work on a breadboard.
- Creating a functional algorithm that can detect when people are walking by.
- Learned new things about our respective parts:
  - PCB design, ESP32 interface, GUI design

Challenges

- Inability to test power subsystem on PCB before baking ESP32 chip
- Waiting for parts staggered progress
- Ultrasonic sensor has a shorter range for needed precision (70cm < 200cm)

Main Takeaways

- Leave more room for error in hardware
- Make sure power subsystem can be disconnected if other parts need to be baked/placed prior:
  - 0 Ohm resistors on main power line
- Communication and organization
- Work together in person

Recommendations for Future Work

- Use sensors that are more precise for counting people like IR laser/sensor
- Parallelize sensors that count people so they work regardless of NFC tag being tapped
- Create a faster search algorithm for determining whether a user exists in a database

Questions?

Thank you for listening