DESK RESERVATION SYSTEM

By

Siddhant Jain

Akeem Kennedy

Mark Syrek

Final Report for ECE 445, Senior Design, Fall 2019

TA: Mengze Sha

11 December 2019

Project No. 20

Abstract

As the world strives to use the internet in more convenient and innovative ways, the distance between technology and human is decreasing drastically. Today, we have at our fingertips the ability to bring food to our door, record university lectures, and soon enough have cars drive us. Technology has permeated many fields, but others have remained stagnant. As students, we see in our university there are opportunities for technology to make lives easier and smoother. The Desk Reservation System, detailed in this final report, innovates on these public spaces allowing users to monitor the occupancy of a desk and to reserve a desk on their mobile phone. This eliminates the time wasted in searching for a desk during peak hours. The Desk Reservation System comprises of a physical module that sits on each desk, which can be purchased by libraries or any other shared space communities.

Contents

1. Introduction

1.1 Problem Statement

We live in a world with constant distractions, hurried interactions, and stressful deadlines; sometimes we need personal space and time to think. What if there was a solution to reserve a seat in a library to facilitate this quiet time? Quiet time is fundamental to today's society as it gives us a chance to meditate, but also seats at our favorite library, Grainger, are often unavailable because of the first-come first-serve system. The frenetic rush for seats often leads to tension between students and often misplaces the energy on finding a seat rather than the subjects they came to study.

1.2 Objective

The goal of our product is to accomplish the following type of workflow: The user reserves a particular seat in Grainger Library through their phone. The LCD on the desk then says the seat is reserved. The user arrives at the desk in a timeframe of 15 minutes and taps their RFID button on the compact device to signify check-in. The user uses the desk for the duration of their reservation; there is a proximity sensor verifying that the desk is being occupied. When it is time for the user to leave the desk, the user taps the RFID button on the compact device to signify check-out. The proximity sensor will confirm if the user left the desk. We will build three desk modules as a foundation to simulate a desk reservation system.

Our Desk Reservation System will be similar to the reservation systems that already exist in other industries. An example is the food industry. Currently, restaurants allow their customers to reserve tables in advance through their respective mobile application [1]. This helped restaurants eliminate the problem of overcrowding and it reduced wait times. Another use of reservation system is in the airline industry. An example is the Computer Reservation System, which kept track of seats on airplanes and returned the availability of them [2].

1.3 High Level Requirements

1. The desk module knows the status of each desk and if a reservation is made.
2. The desk module will verify if a user is present and alert the user on the LCD if they exceed their reservation time.
3. We intend to build three desk modules; however, the system is modular and can support multiple units.

2 Design

2.1 System Overview

The Desk Reservation System is comprised of several modules. The Desk Module System communicates with the database and sends sensor data indicating the current status of the module. The database determines what to display on the LCD screen and sends it to the Desk Module System. Users interface with the desk module through the mobile application. The mobile application communicates with the database to acquire the availability of each desk and allows users to make reservations. Figure 1 shows how all the components of our desk reservation system are connected.

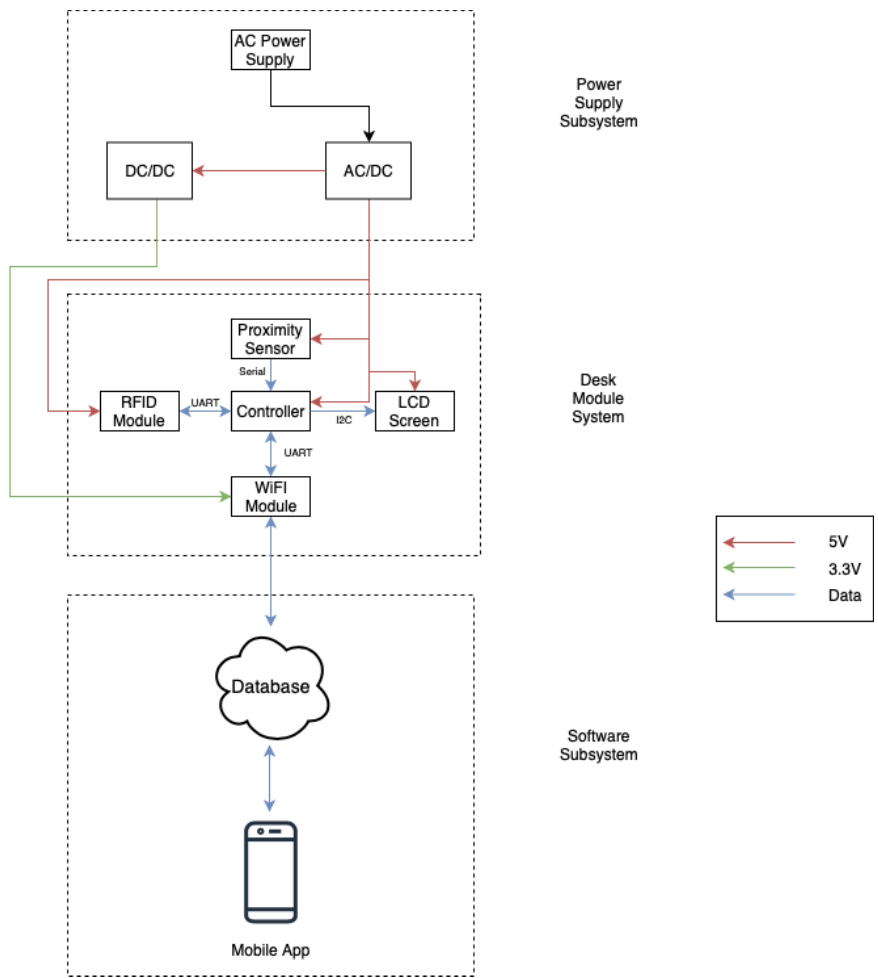


Figure 1: Desk Reservation System Block Diagram

2.2 Power Subsystem

Each desk module has a power subsystem which is responsible for powering its own module. All components onboard the desk module require specific DC voltages, and the components below meet these requirements. This subsystem is comprised of two components: AC/DC converter and a DC/DC converter.

2.2.1 AC/DC Converter

The first is an AC/DC converter, which converts the 120 V AC to 5 V DC. This also will connect to our PCB through a barrel jack, which is a safe and reliable way to connect power to the board. The component chosen for our project is limited to outputting 10 Watts and a max current of 2 Amps.

2.2.2 DC Voltage Regulator

The second component is a 3.3 V linear regulator, which is used to step down the voltage for the Wi-Fi module. The benefit of a linear regulator is the small footprint, low noise, and low cost.

2.3 Desk Module Subsystem

This subsystem is the hub of hardware communication between the sensors, display, microcontroller, and Wi-Fi module. These components will all be placed on a printed circuit boards which, will eliminate shorting wires and ensure solid connections.

2.3.1 LCD

The display, NHD-0216K3Z-FS(RGB)-FBW-V3, chosen for the desk reservation system has the ability to display two by sixteen characters with an RGB backlight screen. Also, the LCD screen is able to communicate with a microcontroller through I2C and SPI communication protocols. We decided to use the I2C protocol because less wires are required for communication. Furthermore, the Arduino has the Wire.h library which supports I2C natively [3]. There are four status modes for the Desk Module: Reserved, Open, Violation and OFF, and their color relation can be seen below in Table 1.

Table 1 Status Messages and Color of LCD Screen

Color	**Text**
Green	Open
Blue	Reserved
Red	Violation
Off	No Text

2.3.2 Proximity Sensor

We chose to use an ultrasonic sensor, HC-SR04, to determine if the desk is occupied by a user or not. The sensor outputs a series of pulses and waits for them to return as an echo; the timing diagram of Figure 2 displays how this operation works. This device was chosen because of the simple documentation, low cost, and quick measurements.

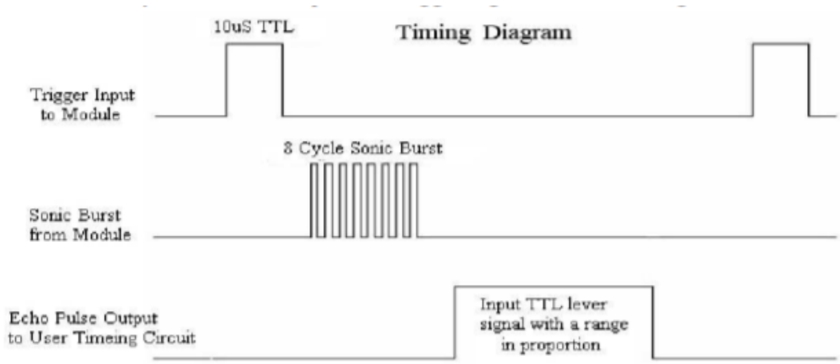


Figure 2: Timing Diagram [4]

2.3.3 RFID Module

The RFID module, ID-12LA, allows users to check into their reservation since a unique RFID button is required per user. This module has a built-in antenna, which fits in our modular design. Although the documentation said the range of the RFID module is 12 cm, in testing it was discovered to be 4 cm. This was not a limiting factor to our design because the enclosure has cavities in it; this allows the user to get as near to the RFID module as needed.

2.3.4 Wi-Fi Module

The Wi-Fi module, ESP8266-01, communicates between the hardware and the database. We chose this device because of its low cost, small size, and FirebaseArduino [5]. FirebaseArduino allows us to send data from the hardware to the database easily. This module interfaces well with our microprocessor, meaning that each device could run separate processes at the same time. This simplifies the testing because fewer conditional requirements are required to determine success. This module uniquely requires 3.3 V DC while demanding the most current as it operates at a peak of 215 mA and an average of 80 mA [5-6]. This did not affect the IO pins; as a result, we used the pins at 5 V logic. Although this is inconsequential, we learned that for best practices a logic shifter (voltage divider) is required between the Wi-Fi module and the microprocessor.

2.3.5 Microcontroller

The microcontroller, ATMEGA328-PU, is the brain of our system, which handles the communication between the Wi-Fi module, RFID module, LCD, and proximity sensor. The microcontroller has analog and digital inputs and outputs. In addition, the microcontroller supports serial communication protocols (UART, I2C) [7]. The advantage of using this microcontroller is the compatibility with Arduino libraries. The ATMEGA328-PU is a synchronic device which means it must operate on a clock cycle. We used an oscillator to function as the clock. Furthermore, it simplifies the development cycle as the module can be programmed on the Arduino UNO board and then transferred to the socket on the PCB.

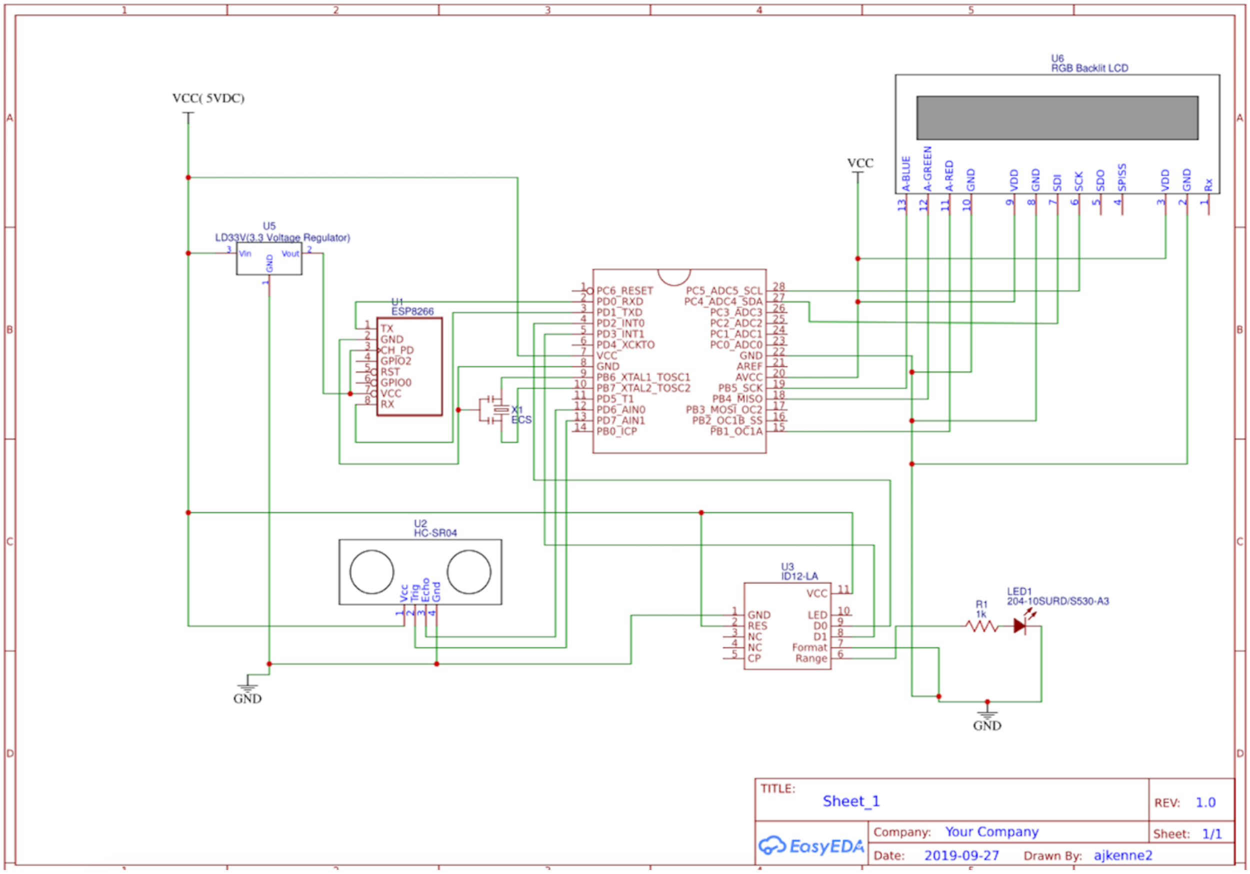


Figure 3: Desk Module Schematic

Figure 3 above shows the desk modules circuit schematic and how each of the components are connected to each other. The microprocessor (center) handles the UART and I2C protocol coming from the Wi-Fi module (top left) and the LCD (top right). The LCD is connected to four different voltages because it has a reference voltage of 5 V, but the red, blue, and green backlight need specific voltage to operate. Red, blue, and green require 2.0 V, 3.0 V, and 3.0 V respectively, which is accomplished through three voltage dividers. The RFID module (bottom right) only requires a power connection and GPIO pin to receive the data from scanned RFID button. The ultrasonic sensor (bottom left) needs two GPIO pins to measure distances. Below in Table 2, we display the voltage and maximum current for each of the components in the desk module system.

Table 2 Voltage and Current Characteristics of Desk Module Components

Components	Voltages [V]	Maximum Current [mA]
LCD Display – (NHD-0216K3Z-FS(RGB)-FBW-V3)	5.0 (Logic), 2.0(Red), 3.0(Green), 3.0 (Blue)	20
Proximity Sensor – (HC-SR04)	5	15
RFID Sensor – (ID-12LA)	5	15
Wi-Fi Module – (ESP8266-01)	3.3	215
Microcontroller – (ATMEGA328-PU)	5	15

2.4 Microcontroller Code

The desk module functionality is implemented by source code running on the microcontroller and Wi-Fi module. The microcontroller has two main functions running on it:

- dataToESP
  - The purpose of this function is to send data from the ultrasonic sensor and the RFID module to the Wi-Fi module through serial communication. Since the ultrasonic sensor polls rapidly, we decided to only send updated proximity data when someone approaches or leaves the desk area. Furthermore, small changes of movement are not significant enough to trigger a new change.
- dataFromESP
  - This function takes data from the Wi-Fi module and displays the updated desk module status on the LCD. The updated desk module status is determined in the backend.

2.5 Wi-Fi Module Code

There are two functions running on the Wi-Fi module:

- dataToFirebase
  - This function sends the data received from the microprocessor to the database. This data consists of the scanned RFID string and the updated proximity status from the desk module.
- dataFromFirebase
  - This function retrieves information from the database. It has an event listener on the specified desk module LCD path. If the stream is updated, the new change is sent to the microprocessor through serial communication.

3 Database and Cloud Functions

3.1 Software Architecture

The microcontroller reads data from the RFID module and ultrasonic sensor and sends it to the Wi-Fi module. The data is then sent from the Wi-Fi module to the database, where the cloud functions associated with the database processes the incoming data and updates the desk status. If a change of desk status occurs, then the event listener on the Wi-Fi module is notified. This update propagates through the system eventually reaching the LCD on the specific desk module. The mobile application is the user interface of the desk module. It communicates with the database to allow users to reserve desks. The software communication between different components of the desk module system can be visualized below in Figure 4.

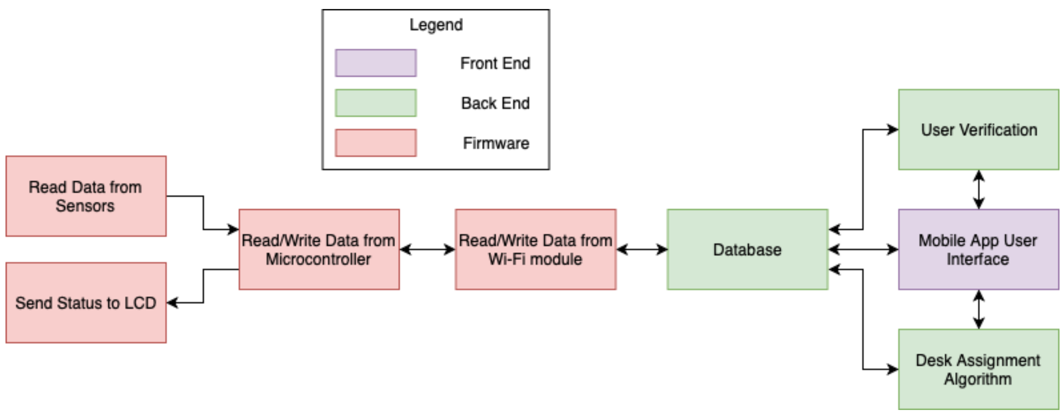


Figure 4: Software Flowchart of Desk Reservation System

3.2 Database Schema

We picked Firebase to be our choice for the database because there was an open source library, FirebaseArduino [5], available that allowed the Wi-Fi module to communicate with Firebase. Firebase operate in a NoSQL fashion which means the schema is dictionary based. Our database schema can be visualized below in Figure 5. We simplified our database to keep track of three types of information: Reservations, Locations, and User's data. This database is scalable because when we add support for more locations, we just insert a new key for that particular location in the Location dictionary.

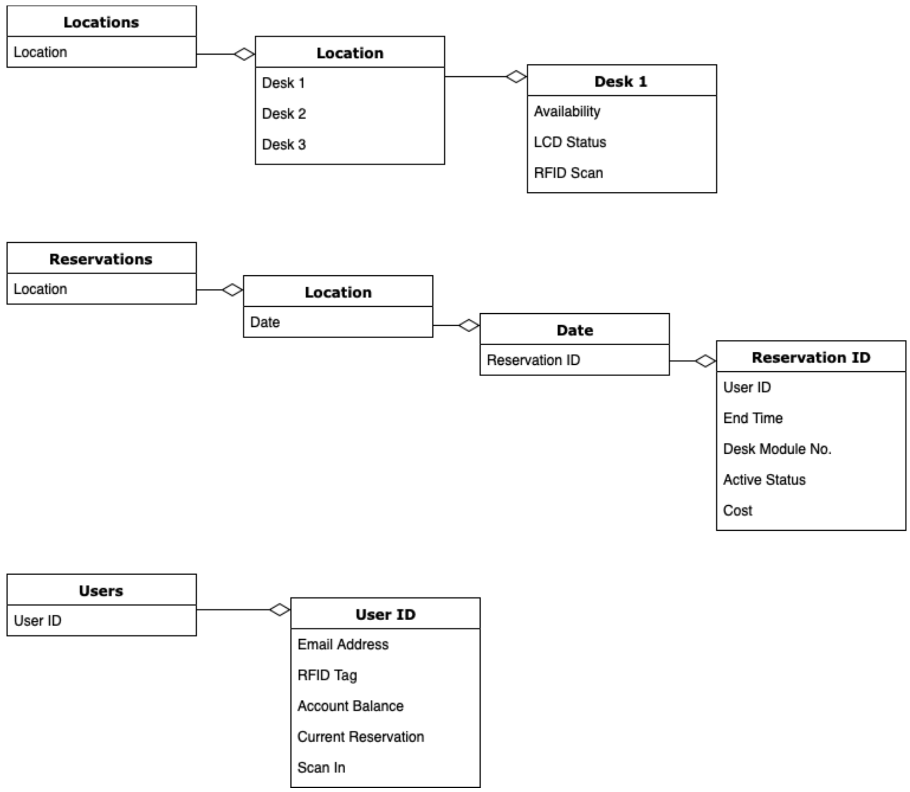


Figure 5: Firebase Database Schema

3.3 Desk Assignment Algorithm

The Desk Assignment Algorithm is an API endpoint hosted on Google Cloud Functions. It allowed our users in the mobile application to make reservations. To make a reservation, the client needs to send a POST Request with the following parameters: start time, end time, location, and desk number. Without these values, a valid reservation will not be created. Furthermore, the function calculated the cost of the reservation and charged the user at the creation of the reservation.

3.4 User Verification

We used Google Cloud Functions associated with our Firebase Database in order to conduct user verification and enforcement. The figure below explains the different potential scenarios that our Desk Reservation System can handle. It requires the desk module to send the updated proximity data and the scanned RFID string to the database where the Google Cloud Functions can process this new data. Our Google Cloud Functions used the ultrasonic sensor to determine if a spot was open or closed and the scanned RFID string to determine if a user had arrived. One of the Google Cloud Functions when triggered by a scanned RFID string watches for data that is written to the Reservation dictionary. The second Google Cloud Function ran every minute and used the proximity data to determine if a user was in violation and the scanned RFID string to determine if a user arrived for their reservation.

Our system can handle the following five scenarios which can be seen in Figure 6:

1. User makes a reservation and checks into their reserved spot.
2. User checks into the desk module with the incorrect RFID button.
3. User arrives into a reserved desk without a reservation.
4. User does not make reservation and checks into an open spot.
5. User makes a reservation and never arrives.

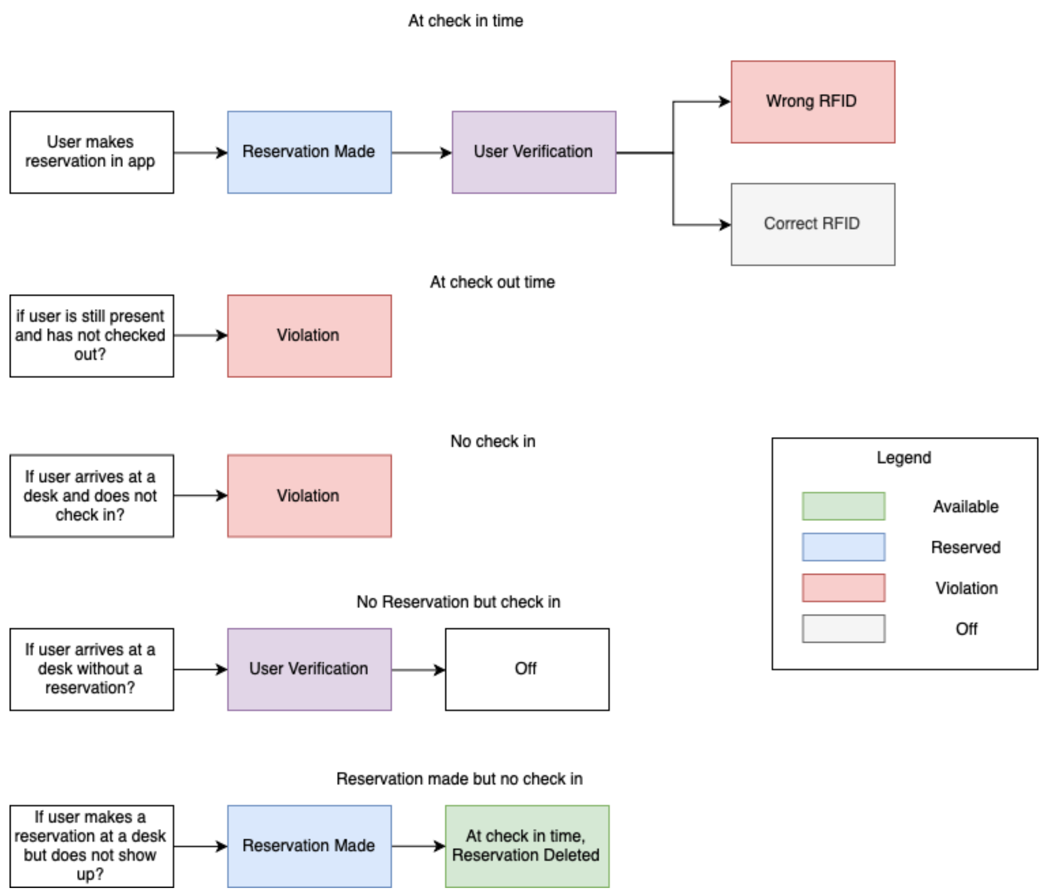


Figure 6: Desk Modules' Scenarios

4 Mobile Application

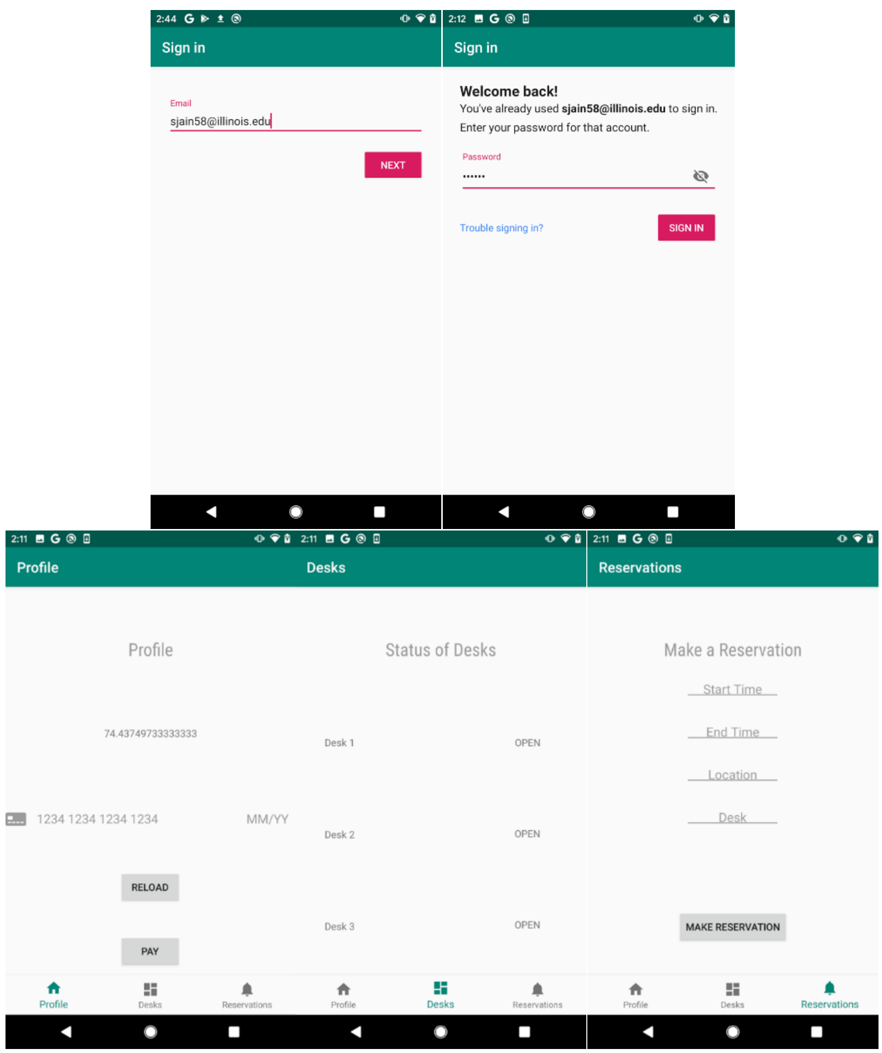
4.1 Mobile Application Overview

We built a mobile application for our project; this is the user interface for our Desk Reservation System. We developed our mobile application on Android because we wanted to target the largest market share of users [8]. Also, Android applications are the most compatible with our choice of database, Firebase. We wanted the interface for the desk module and the Desk Reservation System to be intuitive. The mobile application has three tabs to easily navigate between the different functionality.

4.2 Mobile Application Frontend Overview

When the user opens the mobile application, the user is greeted with a login screen, which requires an email and password. Once they login, users will be able to navigate the mobile application through the bottom tabbed layout which can be seen in Figure 7 along with the other screens made for the mobile application.

Figure 7 Android Mobile Application Screens



The first screen after logging in is the profile screen which displays the user's account balance and allows users to make payments on their balance. The next tab linked on the bottom of the app is the desks screen which displays the available desks for a location. This informs the users of the available desks. The third tab is the reservation screen which prompts the user to input the parameters required for a reservation: start time, end time, location, and desk number.

4.3 Mobile Application Backend Overview

The mobile application is connected to Firebase and on the profile screen there is an event listener on the balance of the user's account. We integrated Stripe, an online payment processor, which allow users to either reload $25.00 or fully pay their account balance. Stripe allows us to securely process a credit card without ever storing the details of the card. Next, on the available desks screen there are event listeners watching for new changes to the desk status. When a desks' availability changes, in real time the app will update with the new status. When a user makes a reservation, a POST Request is fired to the Make Reservation, Google Cloud Function. This will immediately change the status of the particular desk in Firebase. The Reservation dictionary in the database will have the start time, end time, location, and desk number.

5 Verification

To ensure that each component was functioning properly, we tested each component rigorously. This ensured that when the components were integrated, no issue would arise. This worked to our advantage since our final project does not have any errors.

5.1 Hardware Verification

To verify that every component worked properly each component on the desk was connected to a breadboard and an Arduino and tested individually. After each component passed their test, we created a PCB of our schematic seen in Appendix B. Our PCB worked correctly and did not need any modifications. Hardware verification of each component can be found in our Requirements and Verification table in Appendix A.

5.2 Database Verification

The communication between the desk module, the database, and its associated cloud functions were tested individually and together as a whole system. One of the most important verifications was testing the serial communication between the microprocessor and Wi-Fi module. We had to make sure the updated proximity data, the scanned RFID string, and the new LCD status were sent properly between the different parts through serial communication. We started small with just text before moving to longer strings with delimiters that noted the specific data. Next, we added the database to this verification chain. Resulting in testing the communication from the microprocessor to the Wi-Fi module to the database and the reverse of this as well. After integrating the database in these workflows, we added the Wi-Fi module which would send the scanned RFID string and the updated proximity data to the database. At the same time, it would use event listeners to watch for a new LCD status from the database. We verified that the Google Cloud Functions worked by testing out the various scenarios listed in Figure 5 because those were the cases, we designed the system to handle.

5.3 Mobile Application Verification

The mobile application was verified by testing the different functionalities of the application. Each screen was built individually and was tested that the required functionality worked before being combined into one mobile application. The profile page and view available desks screens were tested to see if a manual change in the database for account balance and desks' status were reflected in the mobile application. The reservations screen was tested by running the Cloud Function, Make Reservation, and checking if the reservation was created in the database. Stripe integration was tested by entering valid and invalid credit cards and checking if clicking reload, added $25.00 to the existing balance, and clicking pay, would make the balance $0.00. We tested the system with all three of our desk modules operating and running different tests on them at the same time. Our application was able to show different availability statuses, accurate balances, and make reservations to all three of the desk modules.

6. Costs

6.1 Labor

Our cost of labor was created by estimating our hourly rate to be $40/hour, working 12 hours a week, for 16 weeks, among three people. This results in a total cost of labor to be $57,600.00, and our individual labor cost per person can be seen in Table 3.

Table 3 Labor Cost

Name	Hourly Rate	Hours (16 hours for 12 weeks)	Cost * 2.5
Akeem Kennedy	$40.00	192 hours	$19,200.00
Mark Syrek	$40.00	192 hours	$19,200.00
Siddhant Jain	$40.00	192 hours	$19,200.00

6.2 Parts

Total cost for one module is $81.24 while the total cost for all three modules would be $243.72. Each individual part, their part number, manufacturer can be seen below in Table 4, this can be a reference for us to view the cost of our module at the component level.

Table 4 Parts List

Item	Part Number	Manufacturer	Quantity	Price	Cost
Wall Adapter Power Supply	TOL-12889	NLPOWER-CN	3	$5.95	$17.85
Voltage Regulator 3.3V	COM-00526 ROHS	STMicroelectronics	3	$1.95	$5.85
DC Barrel Power Jack/Connector (SMD)	PRT-12748 ROHS	Adam Technologies Inc.	3	$1.50	$4.50
Wi-Fi Module - ESP8266	WRL-13678 ROHS	Ai-Thinker	3	$6.95	$20.85
16x2 Character LCD-RGB Backlight 5V	763-0216K3ZFSRGBFBWV	NewHaven Display	3	$21.19	$63.57
Ultrasonic Distance Sensor HC-SR04	SEN-15569	Elec Freaks	3	$3.95	$11.85
RFID Button (125kHz)	SEN-09417	Sparkfun Electronics	3	$3.95	$11.85
RFID Reader ID-12LA (125kHz)	SEN-11827 ROHS	ID-innovations	3	$29.95	$89.85

Table 4 Parts List (cont.)

Microcontroller	ATMEGA328-PU	Microchip Technologies Inc.	3	$1.90	$5.70
PCB	PCBWay	PCBWay	3	$0.00	$0.00
Assorted Capacitors and Resistors	Digikey	Digikey	3	$3.00	$9.00
Resonator	COM-09420 ROHS	ECS Inc. International	3	$0.95	$2.85

7. Conclusion

7.1 Accomplishments

We accomplished each of the high-level goals set in the design review at the beginning of the semester. The Desk Reservation System knows the status of each desk and if a reservation is made, verifies if a user is present and alerts the user on the LCD if they exceed their reservation time. We also proved that our system can support multiple units by creating three modules. The Desk Reservation System is completely functional and solves the problem we identified.

7.2 Challenges

While working on our project, we encountered several issues. The first issue we encountered was with the LCD. The predefined library for the LCD would not display text correctly. We unit tested our parts and narrowed the issue to the I2C protocol and realized that the clock (SCLK), which is needed for the transmission of data was not set properly. After fixing this issue, the LCD functioned accurately. The hardware components that we had issues with was the RFID module and ultrasonic sensor. We noticed when we ran the RFID button over the module too quickly, the module did not pick up the string. We attempted to fix this problem by removing the material in between the RFID module and the RFID button and raising the RFID module in the desk module so it is more accessible. For the ultrasonic sensor we initially had worries of the accuracy of the device due its cheap cost and design. However, we discovered that the accuracy was fine, and the greater issue was how often the ultrasonic sensor would take and send a measurement. The Ultrasonic sensor would flood the serial communication with measurements disrupting the scanned RFID string that was also being passed through serial communication to the Wi-Fi module. Another important challenge that we faced was establishing the handshake between the Wi-Fi module and the microprocessor. This was difficult because we could not see the serial monitor of the Wi-Fi module. We could only see what was being sent and received by the microprocessor and not what was being received or sent by the Wi-Fi module. The third challenge was getting data from Wi-Fi module to firebase. The library that we used [5] was not documented well and unstable. It was only after researching different forums, we were able to understand how to use it. Lastly, using the Wi-Fi module was a struggle, the documentation did not explain that there was a programming and running mode which required different pin locations. It was not intuitive to know that for the programming mode that RST and GPIO needed to be low, while for operation that RST needed to be off and GPIO on high.

7.3 Ethical Considerations

7.3.1 Physical Safety

It is our obligation to uphold our standards of the IEEE Code of Ethics. We ensured that we followed Rule #1 and prioritized “the safety, health, and welfare of the public, to strive to comply with ethical design [9]” for our project. Our system will be placed indoors so we do not have to account for weather. The desk module will be powered from the wall outlet and designed to prevent accidental electrocution. This enclosure will also be IP20 (hole greater than 12.5 mm) to protect against large objects from coming into contact with the electronics [10]. This rating has to be lower than initially planned in the design document because we needed to expose the RFID module due to a range issue, along with preventing our desk module from being a Faraday cage. In addition, we will protect the desk module from abuse and prevent damage from drops.

7.3.2 Data Security

Respecting our customer's privacy is the foundation of our product. We followed the standards set in the ACM code of Ethics. Customers are entrusting us with sensitive data. We abided by section 1.6 which states we "should only use personal information for legitimate ends and without violating the rights of individuals and groups [11]." Furthermore, this data is considered to be personally identifiable information. We followed IEEE Code of Ethics [9] Rule #8 as our service will not discriminate on race, religion, and gender. Our database is be encrypted, and we trust that the database service will safeguard our customers' data rigorously. After a certain period, we delete reservation data to follow the best industry practices. This follows ACM Code of Ethics Section 1.6: "prevent re-identification of anonymized data [11]." We have highlighted the issues above that we think are fundamental to our product. We also abided by the standards set by ACM, IEEE, and the University of Illinois at Urbana-Champaign.

7.4 Future Work

Our project can be used to create a desk reservation system at a library, but it can be used for data analytics as well. Since each reservation will be tracked in the database a library can determine the usage of certain areas of the library and peak hours. This information can be used to determine staffing for peak hours, or if certain floor plans are more effective than others. Also, given more time we could redesign the desk module in order to downsize the footprint. Another improvement we would make is designing a more elegant mobile application, along with developing a mobile application compatible with Apple devices so that we do not lose half of all potential users. The final issue is the limiting cost of the desk module. The components we chose were easy to use; however, they were not cheap. Since the desk module is modular, in the future we can replace parts with cheaper equivalents.

References

[1] R. Andrews, “The 9 Best Online Restaurant Reservation Systems Compared,” *The Simple Online Restaurant Reservation System*. [Online]. Available: https://restaurant.eatapp.co/blog/online-restaurant-reservation-systems. [Accessed: 18-Sep-2019].

[2] D, Wardell, (1991). *Airline Reservation Systems*. [online] Academia. Available at: https://www.academia.edu/487385/Airline_Reservation_Systems [Accessed 18 Sep. 2019].

[3] "Arduino - Wire", *Arduino.cc*, 2019. [Online]. Available: https://www.arduino.cc/en/Reference/Wire. [Accessed: 03- Oct- 2019].

[4] Cdn.sparkfun.com, 2019. [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf. [Accessed: 02- Oct- 2019].

[5] FirebaseExtended, “FirebaseExtended/firebase-arduino,” GitHub, 02-Oct-2019. [Online]. Available: https://github.com/FirebaseExtended/firebase-arduino. [Accessed: 02-Oct-2019].

[6] Nurdspace.nl. (2019). *ESP8266 - NURDspace*. [online] Available at: https://nurdspace.nl/ESP8266 [Accessed 16 Oct. 2019].

[7] Learn.sparkfun.com. (2019). *Serial Communication - learn.sparkfun.com*. [online] Available at: https://learn.sparkfun.com/tutorials/serial-communication/all [Accessed 17 Sep. 2019].

[8] "IDC - Smartphone Market Share - OS", *IDC: The premier global market intelligence company*, 2019. [Online]. Available: https://www.idc.com/promo/smartphone-market-share/os. [Accessed: 08- Nov- 2019].

[9] “IEEE Code of Ethics.” IEEE - Advancing Technology for Humanity, www.ieee.org/about/corporate/governance/p7-8.html.

[10] “Enclosurecompany.com.” (2019). [online] Available at: http://www.enclosurecompany.com/ip-ratings-explained.php [Accessed 15 Sep. 2019].

[11] "The Code affirms an obligation of computing professionals to use their skills for the benefit of society.", *Acm.org*, 2019. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: 17- Sep- 2019].

Appendix A: Requirements and Verification Table

Table 5 Requirements and Verification Table

Component	Requirement	Verification	Pass/Fail
AC-DC Converter	The AC-DC Converter should convert 120 V (+/- 5%) AC to 5 V (+/- 5%) DC voltage while under a resistive load (100Ω).	1. Measure output of 5 V converter with a 100 Ω load connected with an oscilloscope to verify 5 V (+/- 5%) DC voltage	Pass
DC-DC Converter	The DC-DC Converter can convert 5 V DC to 3.3 V (+/- 5%) with a 100 Ω load attached.	1. Use a voltage source to create a 5 V DC input 2. Connect a 100 Ω load to it and measure output voltage with an oscilloscope and verify it is 3.3 V (+/- 5%) DC	Pass
Microcontroller **(ATMEGA328P-** **PU)**	This microcontroller will need to send and receive data coming from other modules through UART with 1% failure rate.	1. Flash processor with Commands 2. Verify bits are flowing to the Wi-Fi module through the serial monitor 3. Have the Wi-Fi module return the bits 4. Compare the values to see if they match	Pass
Ceramic Resonator 16MHz	The resonator will oscillate at 16MHz (+/- 5%)	1. Turn on the multimeter and select the frequency function. 2. Measure across the resonator and verify that the frequency is around 16 MHz (+/- 5 %)	Pass
RFID Module	The RFID reader can scan the RFID button within range.	1. Scan the RFID reader with RFID button 2. Read the output of the RFID reader and display its value to the monitor 3. Repeat process to verify that the value remains the same when using the same button	Pass

Table 5 Requirements and Verification Table (cont.)

Proximity Sensor	Accurately detect whether an object is obstructing its view within the range of 0 to 1 meter (+/- 10 cm).	1. Place a box in front of the ultrasonic sensor to represent a person at the desk 2. Check to see if the ultrasonic sensor sends back a plausible measure 3. Use a measurement device (tape measure) to verify that the ultrasonic sensor's measurement is correct (+/- 10cm) 4. Change the orientation of the box and location and repeat	Pass
LCD	Accurately display the reservation status on the LCD using I2C protocol to update the LCD.	1. Use the I2C protocol to write data to display 2. Change background color	Pass
Wi-Fi	The Wi-Fi module can communicate with the backend with a 1% failure rate.	1. Blink(s) the LED on the Wi-Fi module then send the number of blinks to the database 2. Verify that the number of blinks on the Wi-Fi module is the number inputted into the database	Pass
Wi-Fi	The Wi-Fi module can communicate with the microcontroller after 1 second of bootup time.	1. Wait 5 seconds after bootup 2. Pass commands from the database through the Wi-Fi module to the LCD 3. Verify it changes the text/color on the LCD	Pass
Wi-Fi	The Wi-Fi module can receive data from the sensors on the desk module.	1. Print sensor data on the serial monitor then send the data to the database through the Wi-Fi Module. 2. Verify that the data on the serial monitor is the same as the data inputted to the database.	Pass
Software	Users can view open desks.	1. Verify opens desks presented in the mobile application are the same as the open desks in the database	Pass

Table 5 Requirements and Verification (cont.)

Software	Users can create reservations.	1. Query database to confirm new reservation was created	Pass
Software	Scan RFID button and see if it is in database.	1. Scan correct RFID and verify correct RFID is in database	Pass
Software	Database verifies that user has checked into the correct reservation.	1. LCD displays message indicating that user is checked in	Pass
Software	System will refuse an incorrect reservation.	1. Scan incorrect RFID button and verify that LCD does change.	Pass
Software	System alerts the user when they overstay their reservation.	1. LCD displays a status that indicates when the user is overstaying their reservation	Pass

Appendix B: PCB Schematic

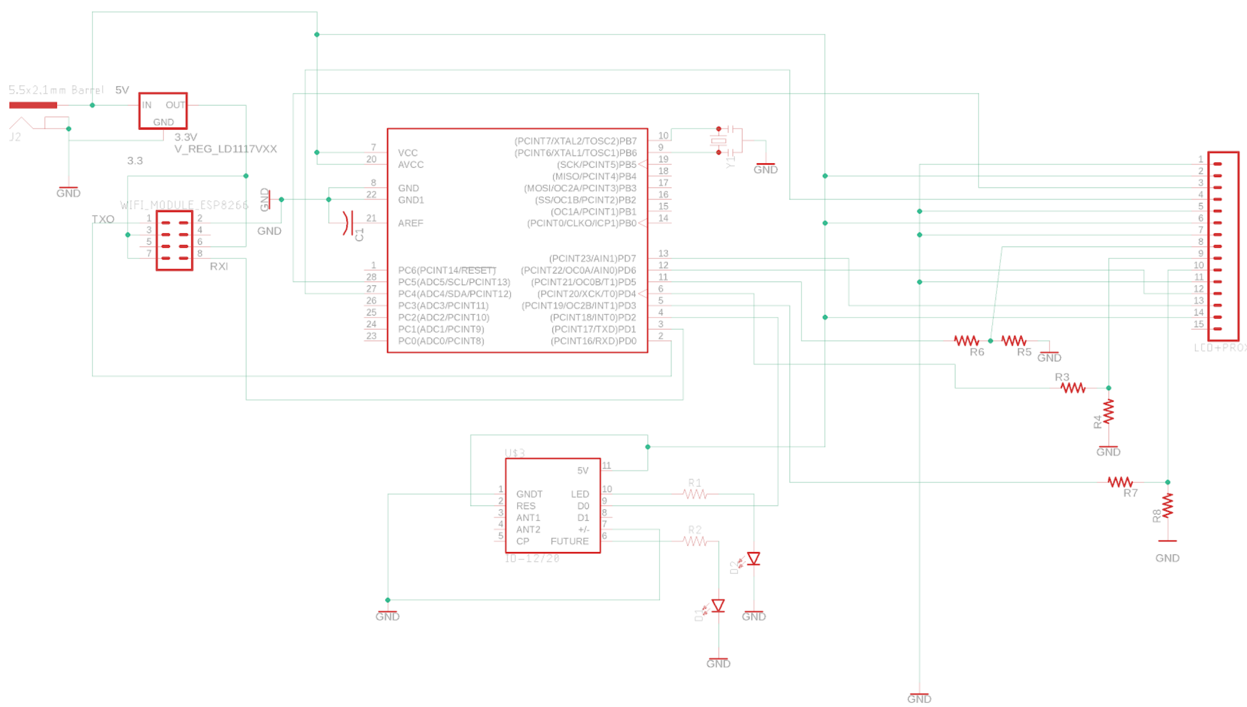


Figure 8: Main Board PCB Schematic

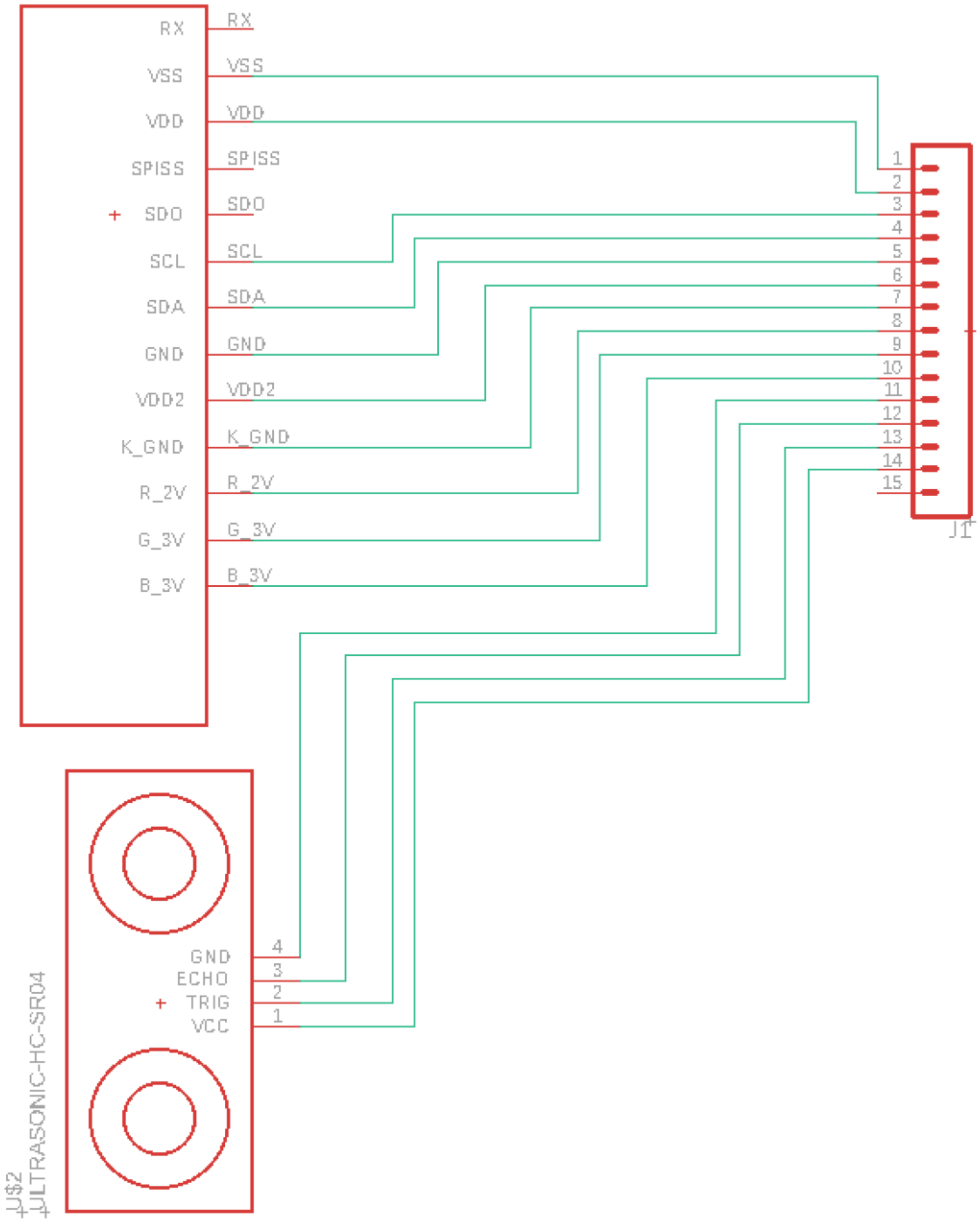


Figure 9: Front Panel PCB Schematic

Appendix C: PCB Layout

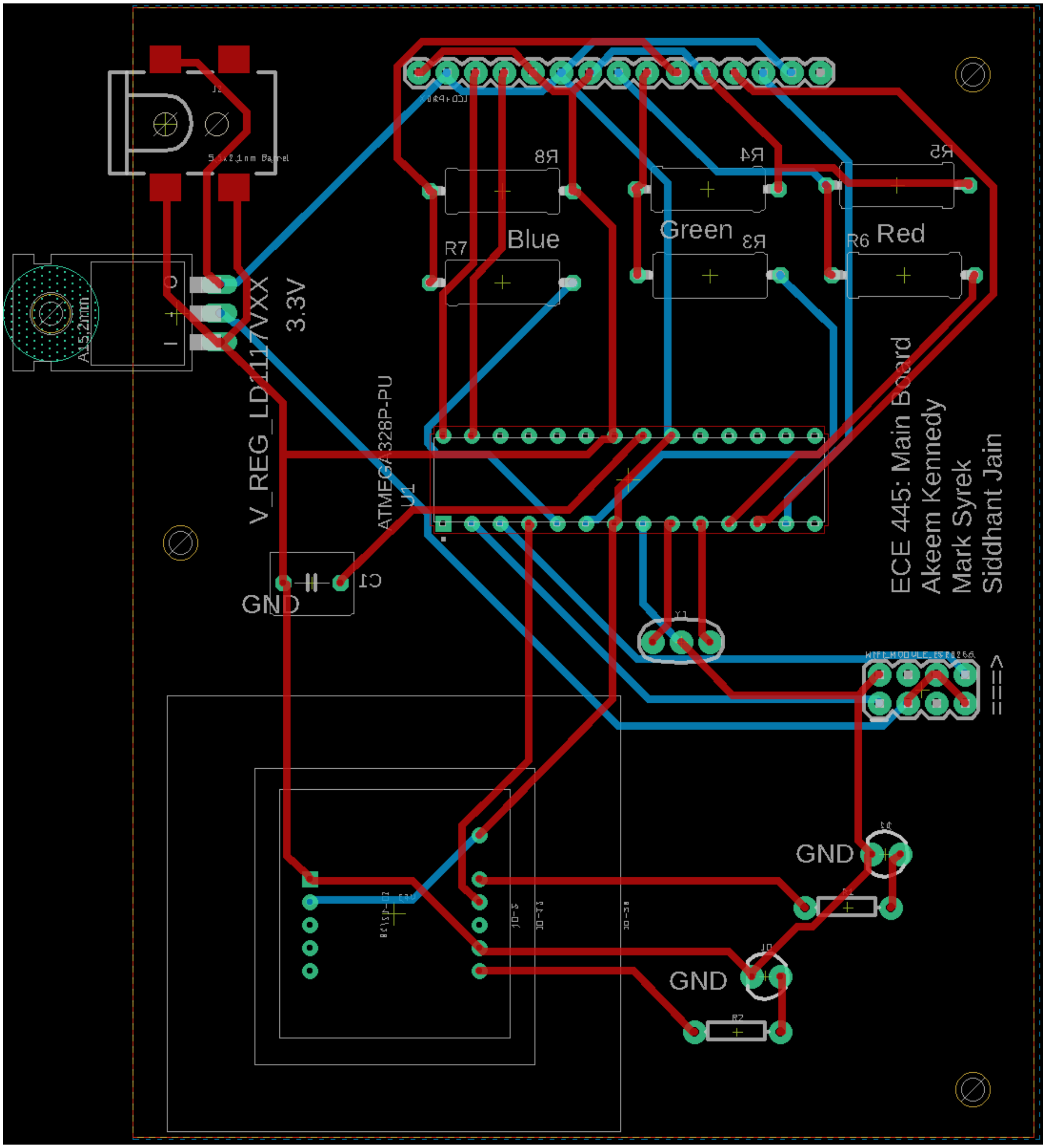


Figure 10: Main Board PCB Layout

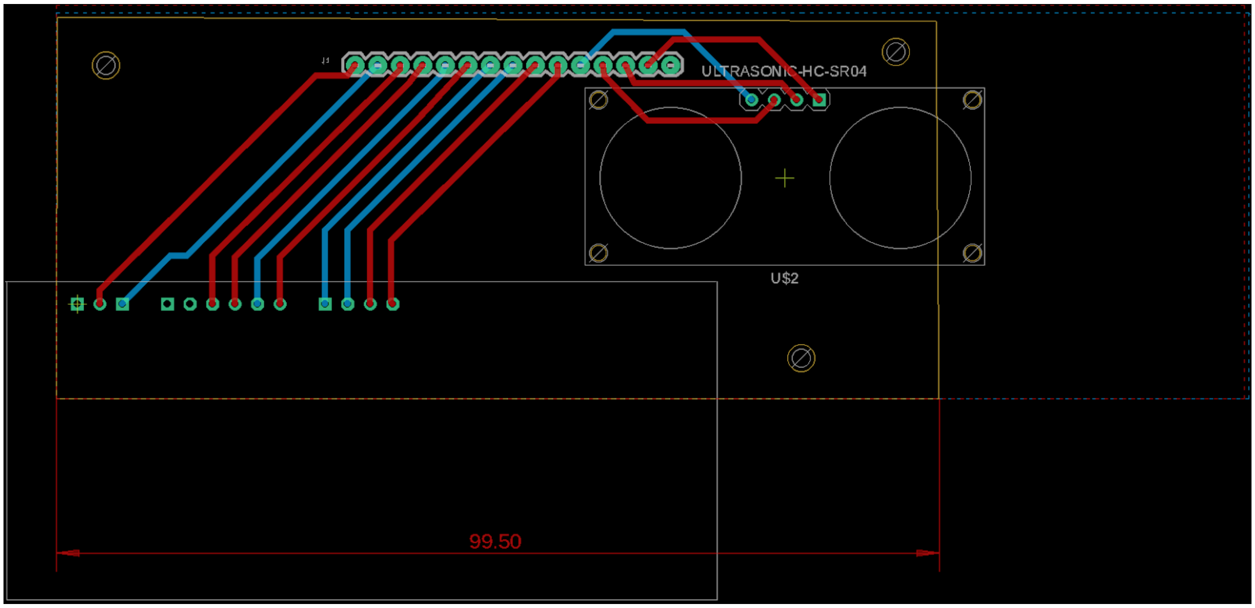


Figure 11: Front Panel PCB Layout

Appendix D: PCB Pictures

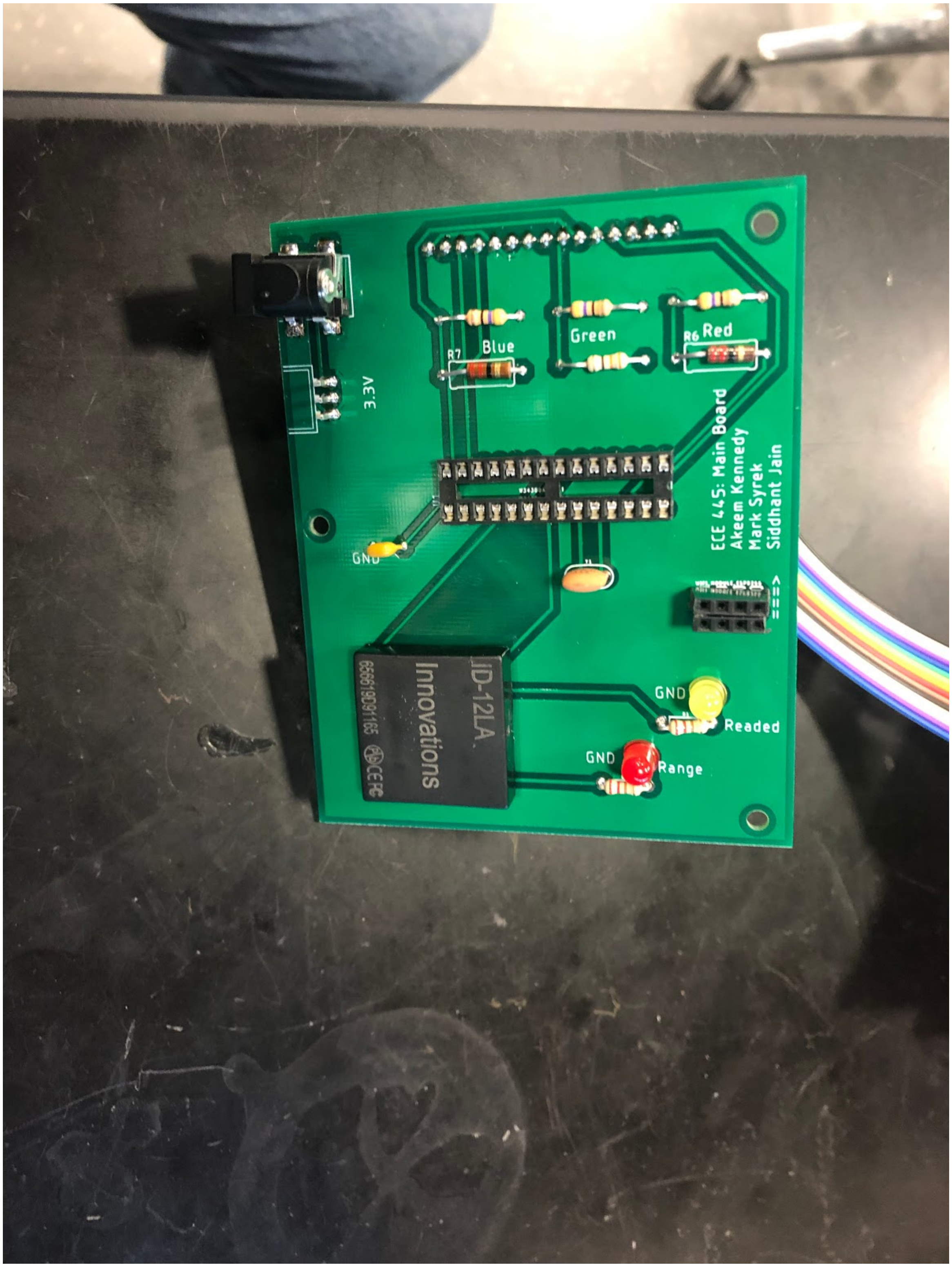


Figure 12: Main Board PCB

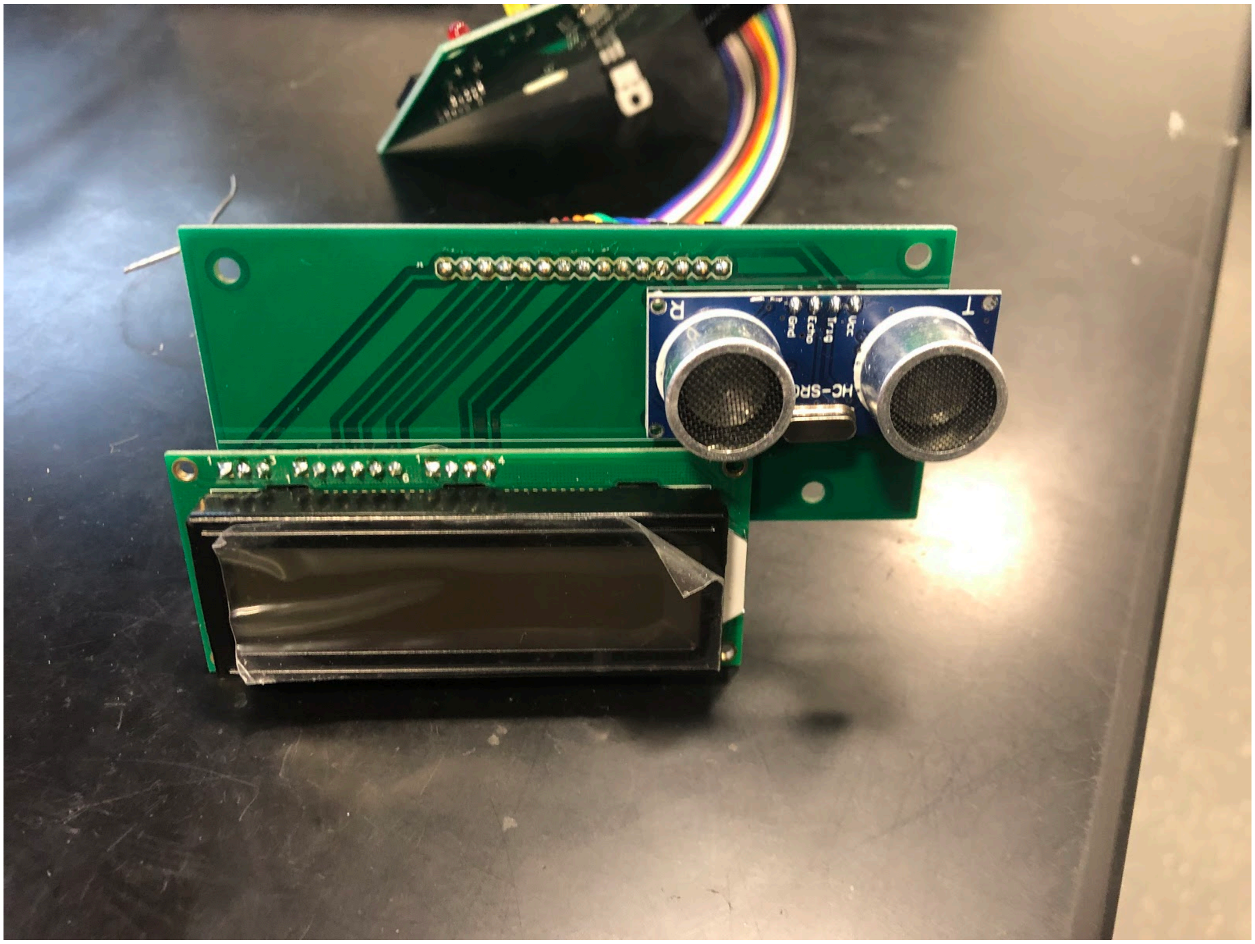


Figure 13: Front Panel PCB

Appendix E: Physical Design

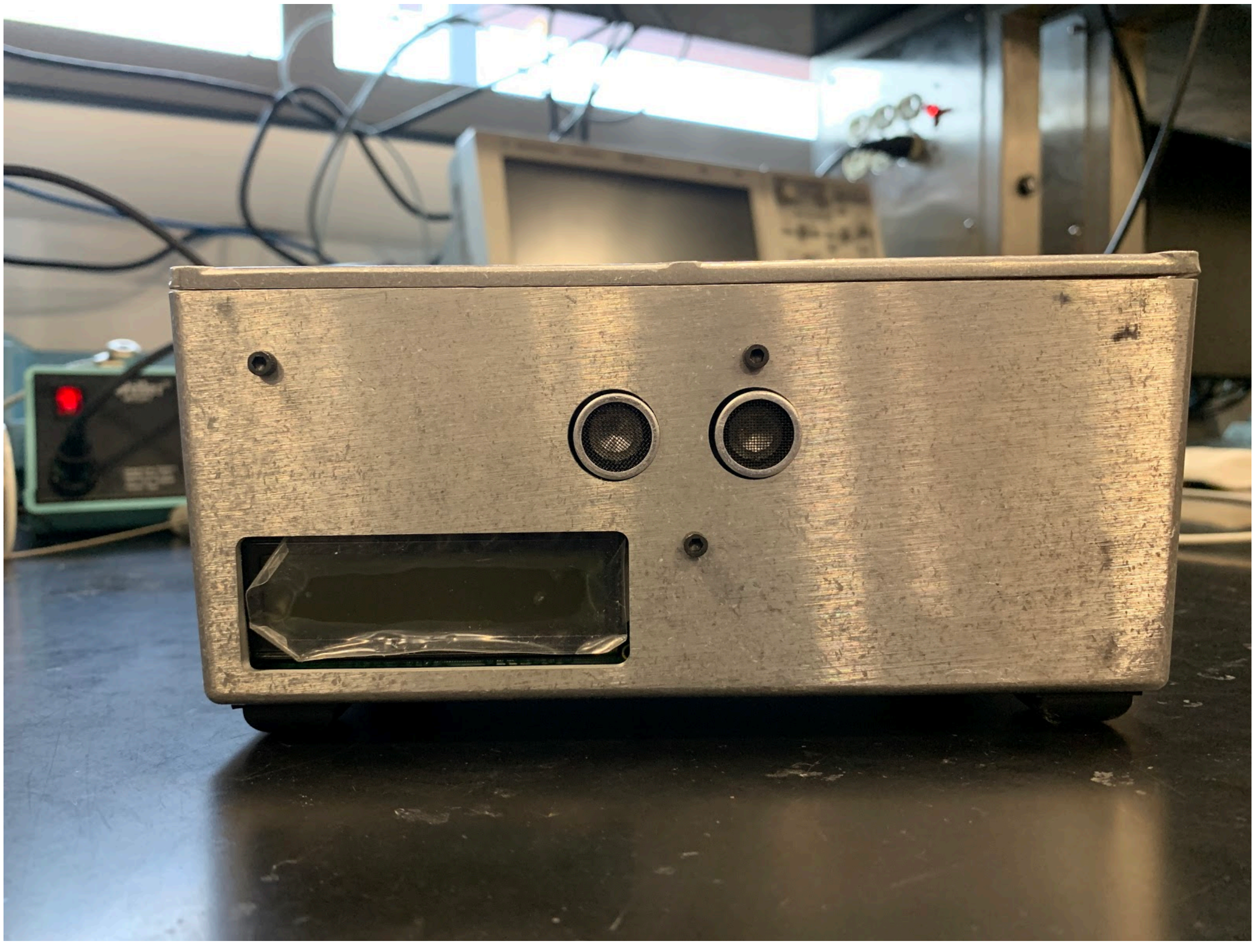


Figure 14: Desk Module Front View

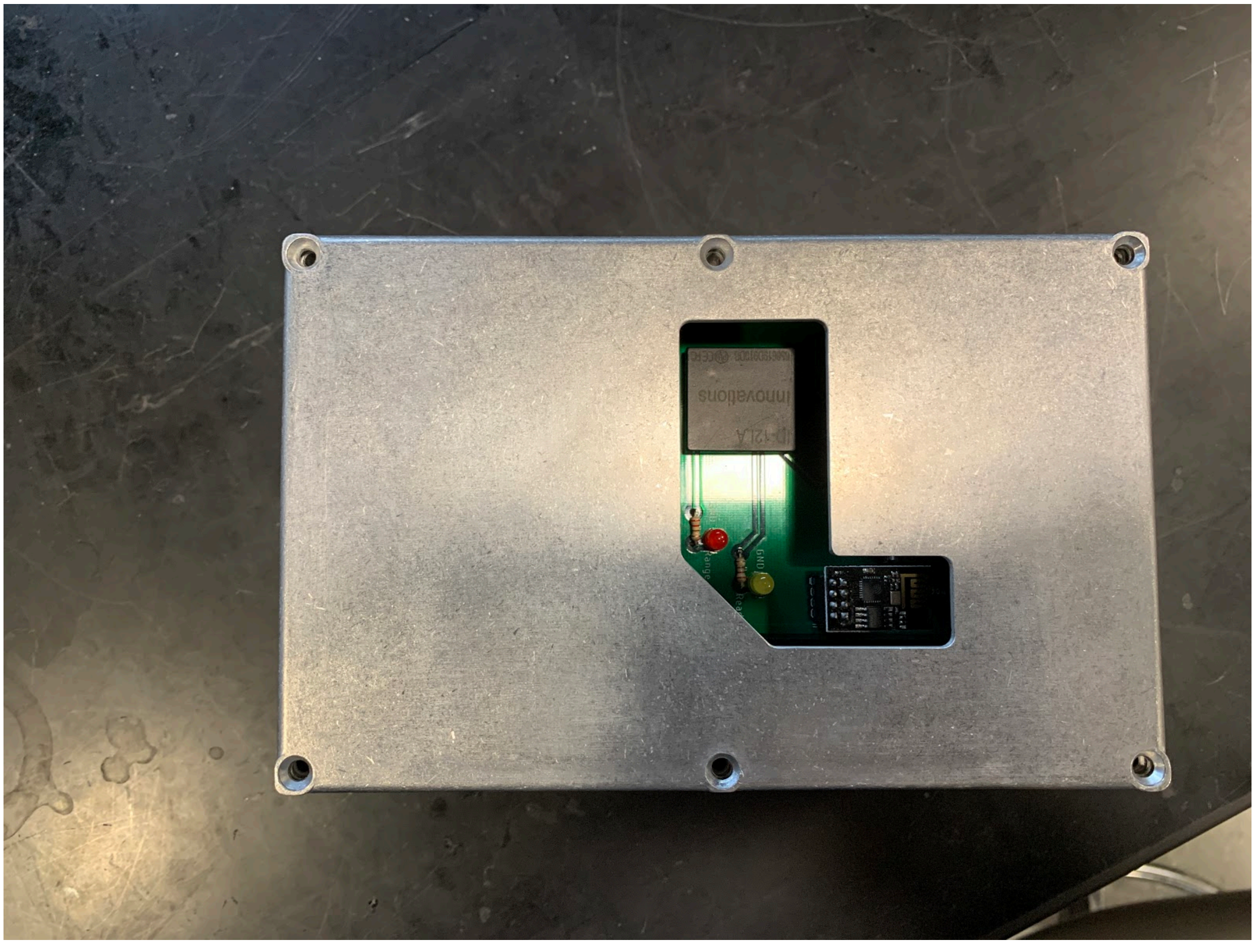


Figure 15: Desk Module Top View

Appendix F: Schedule

Table 6: Weekly Deadlines

Week	Akeem	Mark	Siddhant
09/29	Design Doc	Design Doc	Design Doc
10/06	Work On PCB	Acquire Parts/ test Power circuit	Serial Communication between Microcontroller and Wi-Fi module
10/13	Work On PCB	Read Data from RFID module and Ultrasonic Sensors and Write Data to LCD	Setup Database and have write and read functionality between Wi-Fi module and database
10/20	Work with Siddhant on communication between Wi-Fi module and Database	Acquire box for PCB to mount inside/	Develop the front end for the mobile app, finish sign-in process
10/27	Help Mark assemble PCB	Assemble Circuit onto PCB/Test Circuit	Develop the backend for the mobile app
11/03	Help Mark assemble PCB	Assemble Circuit onto PCB/Test circuit	Test End to End Software
11/10	Fix PCB issues/make sure all components work Assembly of Box	Fix PCB issues/make sure all components work Assembly of Box	Focus on firmware/mobile app issues
11/17	Fix any remaining issues surrounding software	Prepare for demo	Fix any remaining issues
11/24	Prepare for final report	Final Report	Final Report
12/01	Presentation and Final Report	Presentation and Final Report	Presentation and Final Report