SeatDetect

By

Owen Brown (owenmb2)

Yue Li (yuel7)

Huey Nguyen (hueyn2)

Design Document for ECE 445, Senior Design, Spring 2021

TA: Haoqing Zhu

March 4, 2021

Project #18

Contents

Contents ii
1 Introduction 1
1.1 Objective 1
1.2 Background 1
1.3 Visual Aid 2
1.4 High-Level Requirements List 3
2 Design 4
2.1 Block Diagram 4
2.2 Functional Overview 4
2.2.1 Control Module 4
2.2.2 Sensing Module 7
2.2.3 Power Module 13
2.2.4 Software Module 14
2.3 Tolerance Analysis 19
3 Cost and Schedule 23
3.1 Cost Analysis 23
3.1.1 Labor 23
3.1.2 Parts 24
3.1.3 Total 25
3.2 Schedule 25
4 Ethics and Safety 26
References 27

1 Introduction

1.1 Objective

Before COVID-19, there was always a shortage of tables and seats at the Grainger Engineering Library, especially during weeknights. The cubicles were almost always full and if they were not, they were occupied by miscellaneous items. During those times, it would be extremely helpful to know which table in the library is open instead of walking in circles and waiting for a table to open. When there are absolutely no seats available, students spend an excessive amount of time coming to Grainger just to realize there is not an available cubicle to take. Our team seeks a way to alleviate this issue and promote a more accessible library environment.

To solve this problem, the team will design and implement a device for each table that measures the temperature of the surrounding area which can tell the user whether the cubicle is occupied in advance. The occupancy status should transmit through Wi-Fi to the end-user, and the status should be updated in a timely manner after each change to ensure real-time accessibility. Therefore, the user should be able to view these changes, physically locate the desired location, and occupy the free seat. This allows students, the primary users, to save lots of time before entering Grainger, making the library more easily accessible using digital solutions.

In addition to the main problem, the solution should also solve additional issues regarding library policies. For instance, the library staff can enforce more policies such as a maximum amount of inactivity time within each cubicle. This eliminates the issue of students holding spots for others, or themselves, which is especially an issue during crowded times. The team strives to allow all library patrons an equal amount of accessibility in Grainger library and seeking to cultivate a fair environment for all.

1.2 Background

While there are no known existing solutions to this specific problem, solutions to similar problems are already present today. One of these similar problems is finding a parking spot in a parking garage. This problem is solved using infrared technology [1], AI [2], and using various other sensor types. The idea of using a type of sensor to detect an object can be adapted to fit the problem statement, which is to detect humans. Similar to how the parking spot communicates to the driver whether the spot is occupied, the solution must also find a means of real-time communication to inform users of status updates.

From here, it is easy to draw the parallel between finding a parking spot and finding a seat in the library. Since these technologies are utilized in real-world applications, it proves that this problem can be solved in an affordable way, especially since the technology needed to detect a person in a booth is arguably cheaper than detecting a car in a parking spot. In efforts of increasing accessibility and functionality, the project includes an additional web user interface as the primary method in identifying availability spaces.

Due to the scale of the team's proposed project, only the designing and testing of the prototype for one unit will be carried out. However, if the project is scaled up to cover all of Grainger Engineering Library, it would require a much greater number of sensors and power supply. The throughput of the data being transferred would also be much higher. These are all factors to consider, especially when planning to increase the scalability to truly solve the main problem statement.

1.3 Visual Aid

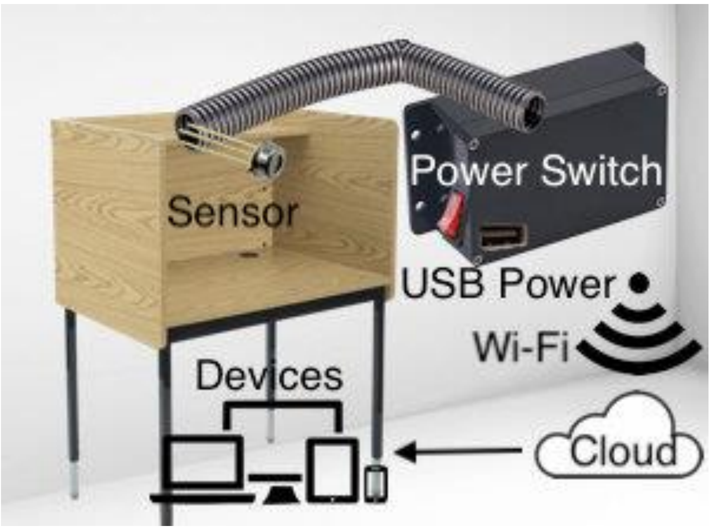

Figure 1. SeatDetect Physical Design Diagram

Figure 1 represents the physical design diagram for SeatDetect. Its main components include the electronics box, booth, and sensor where the end users is detected, and the database system.

The physical design considers the PCB dimensions as well as the dimensions of the chosen sensor. Power comes from the USB port that is attached to the board and sticks out of the physical design. This allows for the USB mini-B port to allow powering the device and interacting with the microcontroller. Beyond the physical device, the device interacts with the database which relays data to a web user interface over Wi-Fi.

1.4 High-Level Requirements List

- An accuracy of occupancy status over 95% on repeated tests of the same booth is the benchmark. This is measured by running multiple occupancy sensing tests on the same sensor unit and counting how many successful readings there are out of total attempts.
- The occupancy status should change from unavailable to available within 15 mins after the seat is no longer occupied. This is to account for brief periods of inactivity such as bathroom breaks. However, the status should change from available to unavailable immediately after the seat becomes occupied.
- The transfer of user data for occupancy status updates on the mobile/web app should be within 30 seconds of a status change. This is set to ensure the system of the device and web/phone app of all users can be serviced and accessible.

2 Design

2.1 Block Diagram

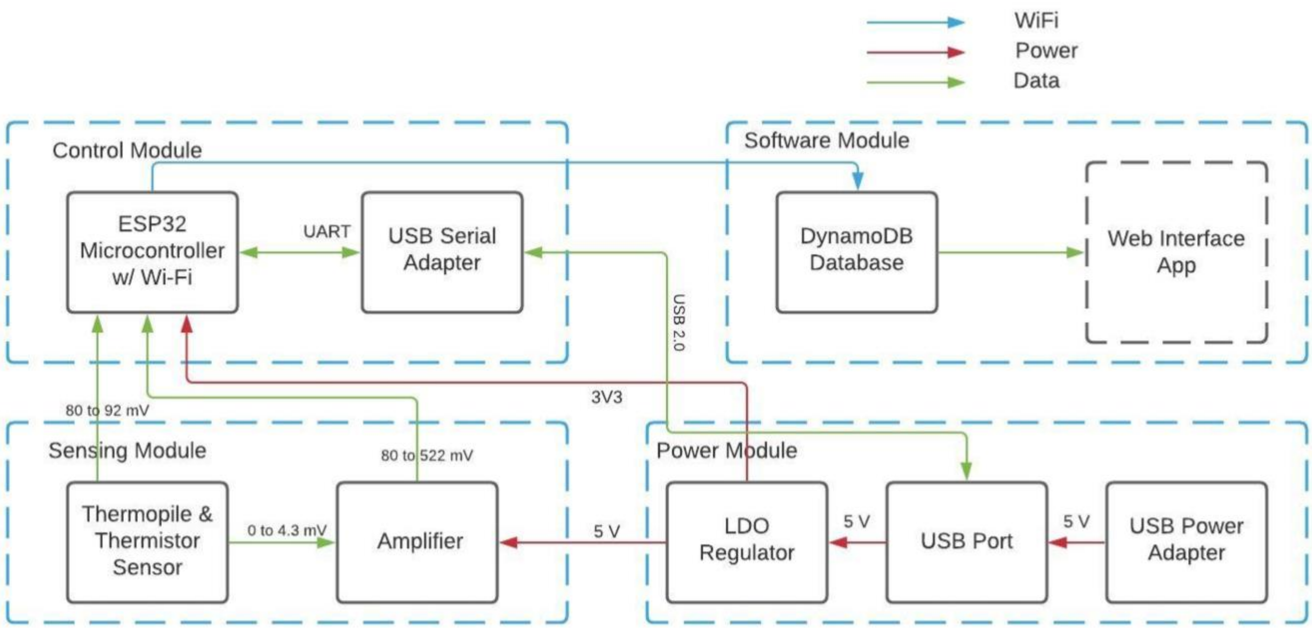

Figure 2. Block Diagram

Figure 2 represents the block diagram for SeatDetect. SeatDetect consists of four main modules: a power supply, a software module, a control unit, and a sensing module. The power system ensures that the system can be powered continuously all day and night with the proper 3.3V. The control unit contains a microcontroller with integrated Wi-Fi, which handles the data received from the sensing module. Lastly, the software module displays the status data to the end-users, the students.

2.2 Functional Overview

2.2.1 Control Module

The control module interfaces with every single module. It also relays preprocessed data from the sensing module to the software module and includes dynamic memory storage included within the ESP32 microcontroller. The data is processed from the sensing module by converting the voltages it receives every 30 seconds to readable temperatures. These temperatures are then interpreted using a detection of human presence threshold of $\sim 35.5\ ^\circ C$. The calculations for the temperature conversions are discussed further later in section 2.2.2.

2.2.1.1 ESP32 Microcontroller w/ Wi-Fi

The ESP32 microcontroller receives the sensor signals, processes the data into whether the seat is occupied or not, and sends the data to the database over Wi-Fi. Using Wi-Fi communication allows for faster data transfer and scalability between multiple sensor data being processed through the microcontroller, which is then relayed to a hosted DynamoDB database.

2.2.1.2 USB Serial Adapter

The USB serial adapter is necessary to enable the programming of the ESP32. It takes in the upload of code from a computer by converting the USB data to serial data that the ESP32 can then interpret.

Control Module	
Requirements	**Verifications**
1. The control unit must be able to reliably transfer up to 10kB of data within 1 second to ensure 6.8kB can be sent out to convey occupancy status and cubicle ID.	1. a. Connect the control unit wirelessly to a computer as a wireless serial device. Add "Illinois_Net_Guest" under WIFI_SSID[] and login credentials into WIFI_PASSWORD[] in Arduino code to connect to a Wi-Fi network. b. Generate 10kB of random bytes using Python. c. Program the control unit to transmit the data over the serial link and record the time in the control unit it took to transfer the data.
2. When the USB port is plugged into a computer, the device must show up as working.	2. Plug the device into at least 2 windows computers, and a UNIX-based computer and verify it is listed as a serial device without additional software.
3. The control unit must be able to persist the data with the correct SeatID into the backend DynamoDB table of the software module within 10 seconds of receiving the signal.	3. a. Repeatedly occupy and leave one seat containing the sensor and b. Confirm that the record in the database has been updated.

Table 1. Control Module Subsystem Requirements and Verifications

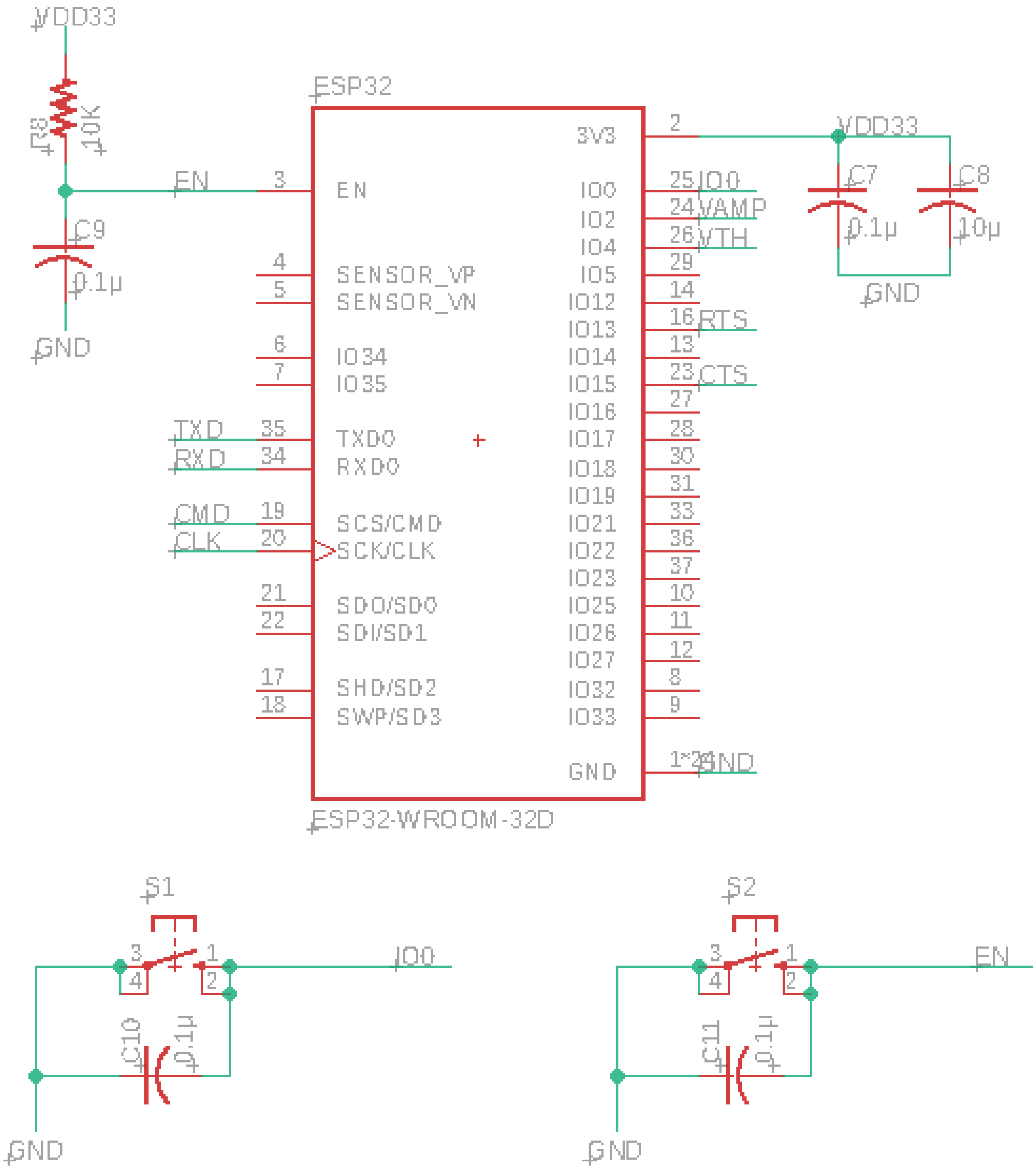

Figure 3. ESP32 Circuit Schematic

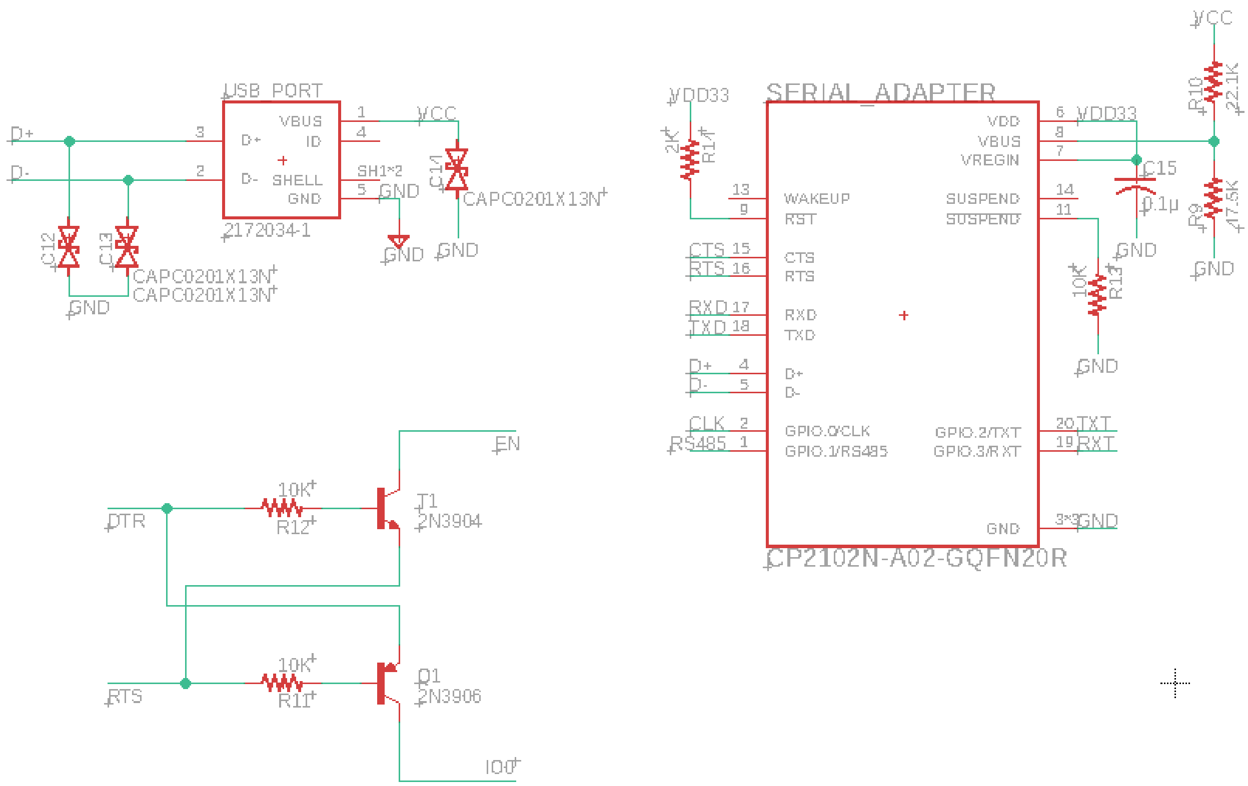

Figure 4. Serial Adapter Circuit Schematic

2.2.2 Sensing Module

The sensing module senses if the user is in the booth using a thermopile sensor and some residual circuitry. It relays passive signals from the thermopile to the control module where it is then processed to sensible and understandable data.

2.2.2.1 Thermopile Sensor

ZTP-135SR is a thermopile sensor that detects human presence by thermal energy correspondence.

2.2.2.2 Amplifier

The amplifier needs to amplify the signal from the sensor as the signal voltages are much too small to be read by the ESP32.

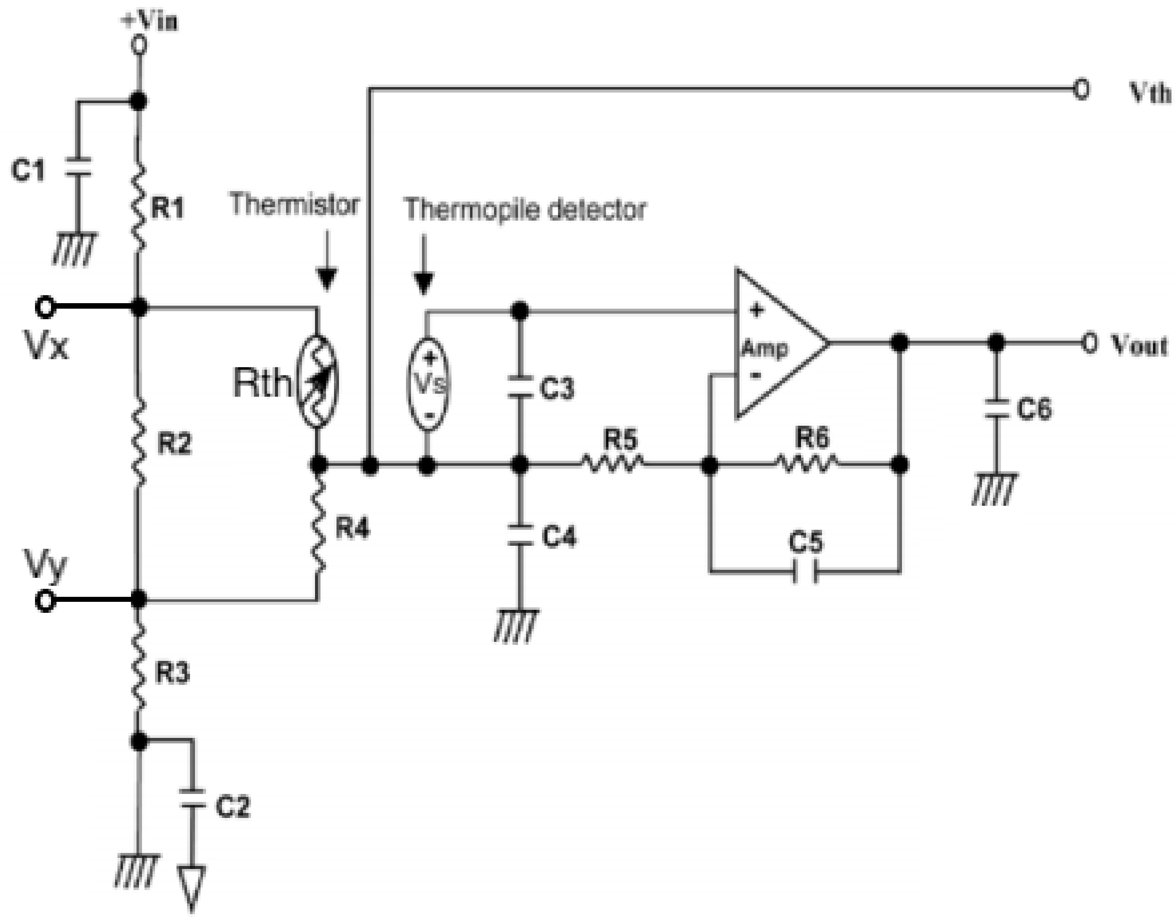

Figure 5. Sensing Module Circuit

Figure 5 represents an example interface circuit for the thermopile [3].

$$V_{out} = V_s\left(1 + \frac{R_6}{R_5}\right) + V_{th} \tag{1}$$

Equation (1) is the equation for V_{out}, which is derived from simple ideal op-amp analysis. As seen in Figure 5, V_s represents the voltage generated from thermal energy by the thermopile while V_{th} represents the voltage of a node that depends on the value of R_{th}, the resistance of the thermistor that depends on ambient temperature.

$$V_s = S(T_O^B - T_A^B) \tag{2}$$

$$S = \text{Sensitivity coefficient (mV)}$$

$$T_O = \text{Object temperature (K)}$$

$$T_A = \text{Ambient temperature (K)}$$

$$B = \text{Coefficient } (\sim 4)$$

Equation (2) is the equation for the thermopile voltage V_s [3]. This equation is not useful for calculating T_O, unless S and T_A are known values. Measuring T_O is the main purpose of this circuit because T_O represents the temperature of the person sitting at the booth, which is needed to detect the presence of a user.

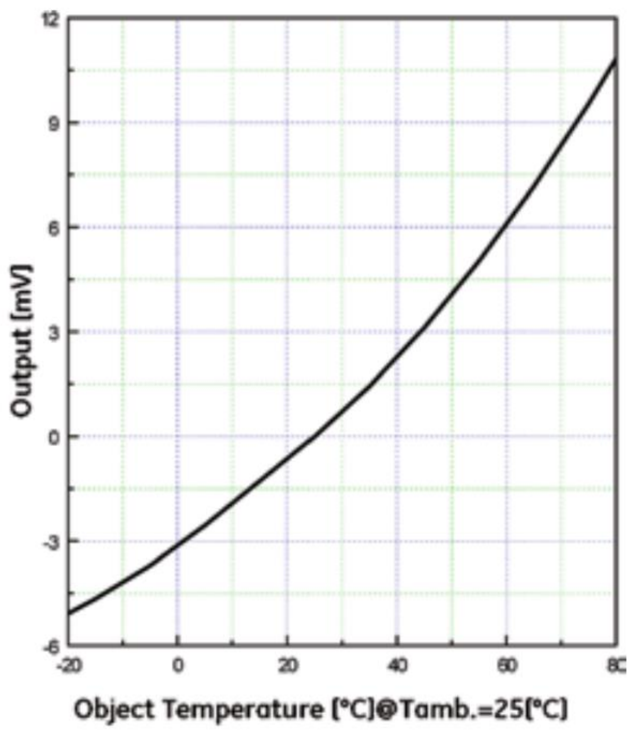

Figure 6. Thermopile Voltage Plot

Figure 6 [5] represents the plot of V_s vs. T_O when $T_A = 25\ °C$. While this plot is not useful by itself since it only provides values of V_s for a fixed T_A, it is useful for approximating S using a rearrangement of equation (2).

$$S = \frac{V_s}{T_O^B - T_A^B} \tag{3}$$

To approximate $\boldsymbol{S}$ using equation (3), the variables that most closely resemble the actual application is used in the calculation. The application takes place in the Grainger Engineering Library, which is most likely using $\boldsymbol{T_A \approx 22\ °C}$ corresponding to room temperature and $\boldsymbol{T_O \approx 37\ °C}$ corresponding to body temperature. Figure 6 can be used to find that $\boldsymbol{V_s \approx 2.65\ mV}$ at $\boldsymbol{T_O = 37\ °C}$. Then, equation (3) can be used to estimate $\boldsymbol{S = 1.96 \times 10^{-12}}$.

If $T_A = 22\ °C$ always held true, no further calculations would be necessary as all of the variables would be solved for, but it is better to account for the ambient temperature as well, as the room temperature of Grainger is not always going to be the same. Since V_{th} is a function of R_{th} which is a function of T_A, R_{th} needs to be calculated next, as V_{th} is obtained by simply probing the node seen in Figure 5.

Since V_{th} is a function of R_{th} which is a function of T_A, R_{th} needs to be calculated next, as V_{th} is obtained by simply probing the node seen in Figure 5. R_{th} can be solved for by doing analysis on the surrounding resistor divider circuit.

$$R_{eq} = \frac{R_2(R_4 + R_{th})}{R_2 + R_4 + R_{th}} \tag{4}$$

$$V_x = \frac{V_{in}(R_{eq} + R_3)}{R_1 + R_3 + R_{eq}} \tag{5}$$

$$V_y = \frac{V_{in}R_3}{R_1 + R_3 + R_{eq}} \tag{6}$$

$$V_{th} = \frac{R_4(V_x - V_y)}{R_4 + R_{th}} + V_y \tag{7}$$

Equations (4), (5), (6), and (7) are created using Figure 5, where R_{eq} is the equivalent resistance between nodes V_x and V_y. These four equations can then be used to solve for R_{th} in equation (8).

$$R_{th} = \frac{(V_{in} - V_{th})[R_2(R_3 + R_4) + R_3R_4] - V_{th}R_1(R_2 + R_4)}{V_{th}(R_1 + R_2 + R_3) - V_{in}R_3} \tag{8}$$

The reason a resistor divider is used in this circuit is to linearize V_{th} with temperature because R_{th} has an exponential relationship with temperature as seen in equation (9) [4].

$$R_{th} = R_2 = \frac{R_1}{e^{\beta(\frac{1}{T_1} - \frac{1}{T_2})}} \tag{9}$$

Equation (9) represents the method to approximate R_{th} at temperature $T_A = T_2\ (K)$ using a known temperature $T_1\ (K)$ and the resistance R_1 at T_1. The equation also uses the variable beta, which is found in the thermopile's datasheet [5] to be $\beta = 3960\ K$. This equation can be rearranged and used to solve for T_A.

$$T_A = T_2 = \frac{\beta T_1}{\beta - \ln\left(\frac{R_1}{R_{th}}\right) T_1} \tag{10}$$

T (°C)	R_{th} (kΩ)
-20	948
-15	716
-10	545
-5	419
0	324
5	253
10	198
15	157
20	125
25	100
30	81
35	65
40	53
45	46
50	36
55	30
65	20
70	17
75	14
80	12
85	10
90	9
95	7
100	6

Table 2. R_{th} vs. T

Using Table 2 [5], any T_1 near T_A can be used in equation (10).

Finally, now that all the variables are accounted for or solvable, equation (1) can be rearranged into equation (11) and used to find T_O (K), the object temperature, otherwise known as the temperature of the person sitting at the booth.

$$T_O = \sqrt[B]{\frac{V_{out} - V_{th}}{S\left(1 + \frac{R_6}{R_5}\right)} + T_A^B} \quad (11)$$

Sensing Module	
Requirements	**Verifications**
1. The amplifier must amplify the voltage from the sensor by a factor of 100 ± 5% (gain of 100).	1. Measure the voltage before (V_{in}) and after (V_{out}) the op-amp while the complete circuit is under operation. V_{out}/V_{in} must be equal to 100.
2. V_{th} must be between 80 and 90 mV.	2. Put the system under the highest and lowest room temperatures possible while monitoring V_{th} to see if it is within range.
3. V_{out} must be between 80 and 520 mV.	3. Put the system under the highest and lowest room temperatures possible while monitoring V_{out} to see if it is within range. Stand in front of sensor and away from sensor while monitoring V_{out} to see if it is within range.
4. V_S must be between 0 and 4.3 mV.	4. Put the system under the highest and lowest room temperatures possible while monitoring V_s to see if it is within range. Stand in front of sensor and away from sensor while monitoring V_s to see if it is within range.

Table 3. Sensing Module Subsystem Requirements and Verifications

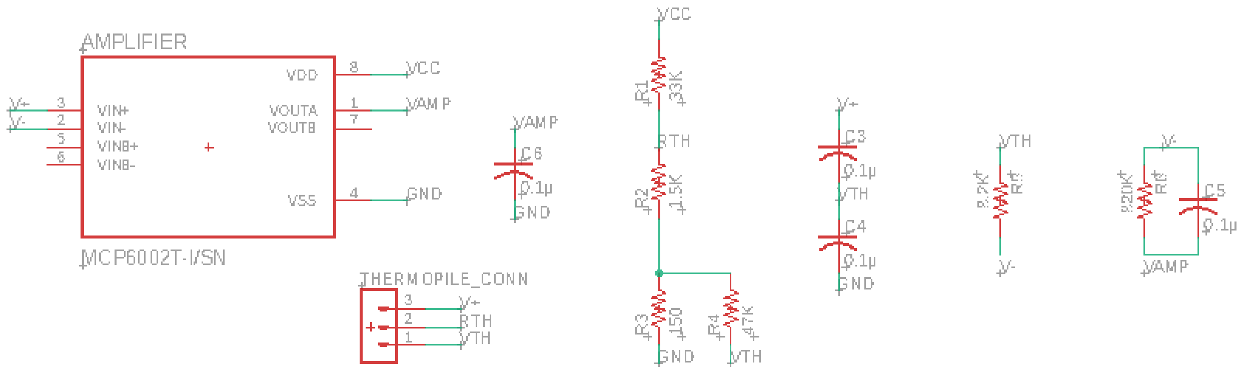

Figure 7. Sensor Circuit Schematic

2.2.3 Power Module

The power module is a wired USB power supply for increased scalability and consistency, which is required to keep the communication network up continually. In a network with multiple sensing modules, this power module is used to power the whole system. Because the addition of sensing modules does not add much additional power requirements from the control module, the addition of sensing modules does not require significantly more power from the power module. In the scenario where enough sensing modules are adding to warrant additional control modules due to the I/O limitations of the ESP32, there would be also be a need for more power modules, and therefore the whole system would be duplicated except the software module. The following requirement does not cover the ladder case.

2.2.3.1 USB Power Adapter

The power adapter converts the grid electricity into DC power that is used for the whole system.

2.2.3.2 USB Port

The mini–B female USB port receives power from the power adapter and allows for programming the ESP32 from a computer (at separate times). The port connects to the voltage regulator for power supply and a serial adapter for communication to the ESP32. When a computer is hooked up to the port, the power adapter wouldn't be, and vice versa.

2.2.3.3 LDO Regulator

The LDO regulator steps down the voltage from the power adapter for the ESP32.

Power Module	
Requirements	**Verifications**
1. The subsystem must be capable of outputting a regulated 3.3V ± 0.1V at 750mA.	1. Provide the subsystem with 4V from a bench power supply and connect the regulated output to a resistive load pulling 750mA, and at the same time measure the steady state output voltage using an oscilloscope.
2. Maintain thermal stability below 125°C	2. Use an IR thermometer to ensure the IC stays below 125°C

Table 4. Power Module Subsystem Requirements and Verifications

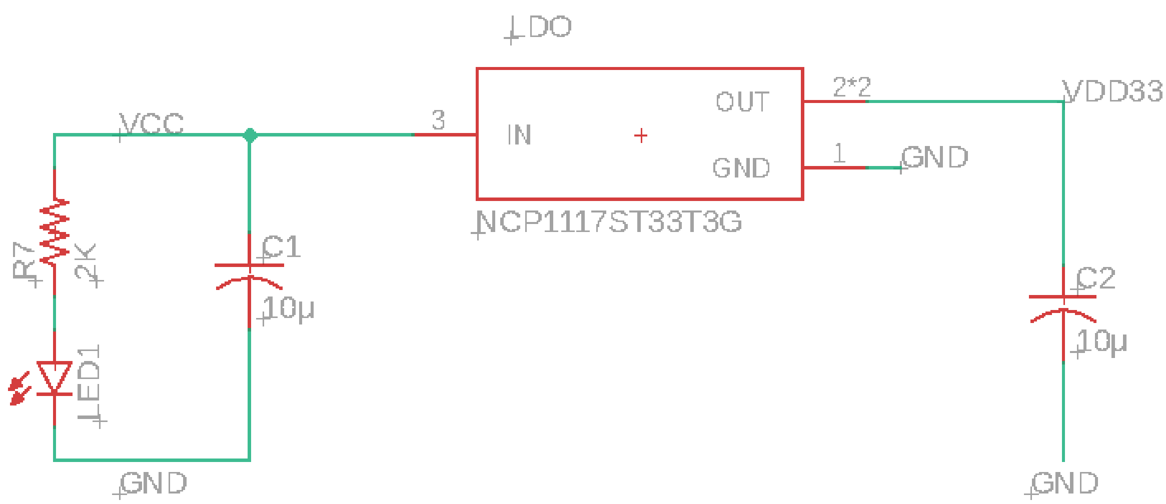

Figure 8. Power Circuit Schematic

2.2.4 Software Module

The software module consists of a web-based application that allows users to check availability and location of seats as well as a database that stores the information pertaining to each seat.

2.2.4.1 DynamoDB Database

The DynamoDB database stores the data and the ability to view the status of a seat (occupied by a person, personal items, inactivity, etc.). We are deciding to use DynamoDB due to its durable and multi-active database capabilities.

SeatID (Integer)	Status (Boolean)	lastUpdatedTime (DateTime)
1	FALSE	2021-02-13T17:09:42.411
2	TRUE	2021-02-13T17:09:42.411

Table 5. DynamoDB Table

The data transmitted by the sensors must be persisted into the DynamoDB database. This database should have the Seat ID as a primary key and it stores the perspective occupancy status and the time which it was last updated. As a table it looks like Table 5.

2.2.4.2 Web Interface App

The Web Interface App allows the user to view the map of Grainger Engineering Library (for now) and its corresponding tables on each floor. It is available to view on a standard web browser on a computer. The interface is interactive so that users could select which floor to view availability, and the corresponding status of each cubicle at the specified location. The user-friendly design and lay-out is intended to allow the user to find the seat relatively easy, as the web page displays the location and orientation of each cubicle in a clear and intuitive way which simplifies the view of Grainger Engineering Library.

2.2.4.2.1 Front End

The front end of this web application is constructed using Python Flask coupled with HTML and CSS. The front end of this application has 2 versions, one for the web browser, another for the mobile phone. The web browser version looks like Figure 9 and the mobile version looks like Figure 10.

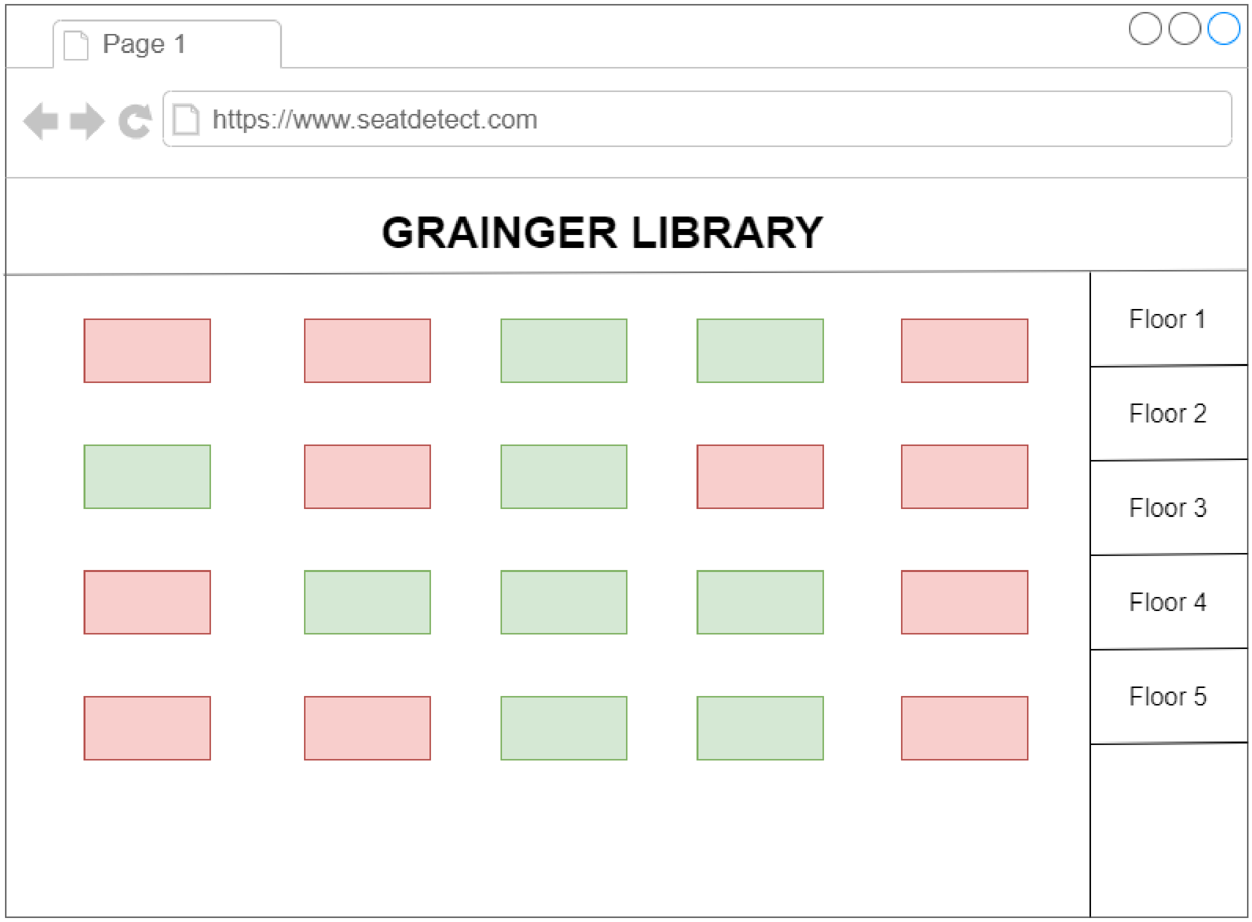

Figure 9. Browser version of the app

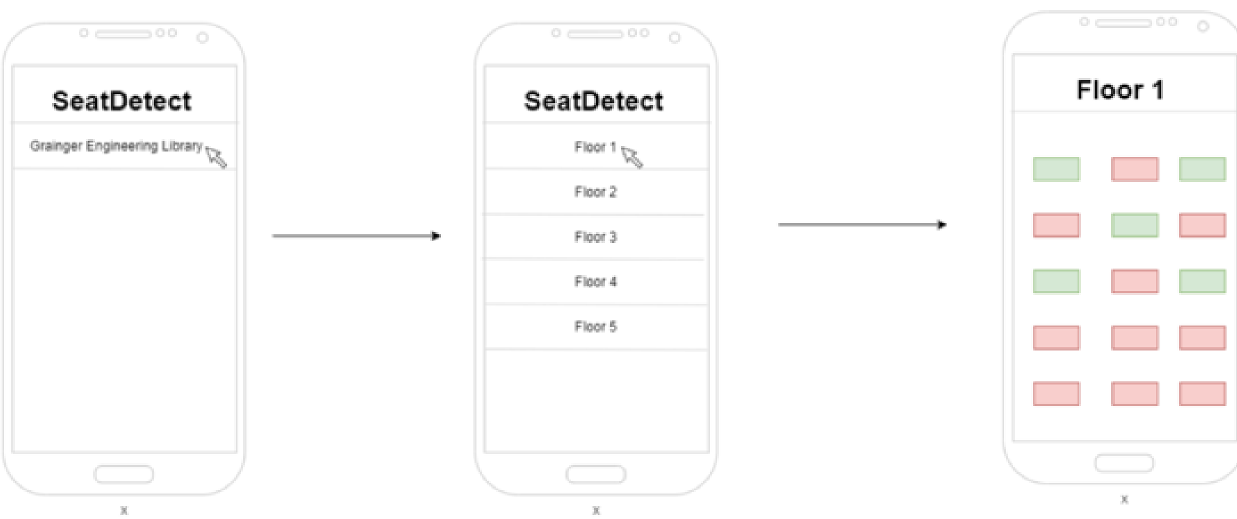

Figure 10. Mobile version of the app

2.2.4.2.2 Back end

The back end for this web application is constructed using Python with the package Boto3. The process scans through each seat and see if the status for that seat has been changed for not, if so, it is going to reflect the change to the end-user, if not, it moves on to the next seat. This process runs every 20 seconds. While for demonstration purposes the cost of reading a single record in the database is negligible, this is done to keep the cost of maintenance of the DynamoDB manageable in the event of the project being scaled up as constantly reading hundreds of records from the database can be expensive. The flow chart of the back-end application looks like Figure 11.

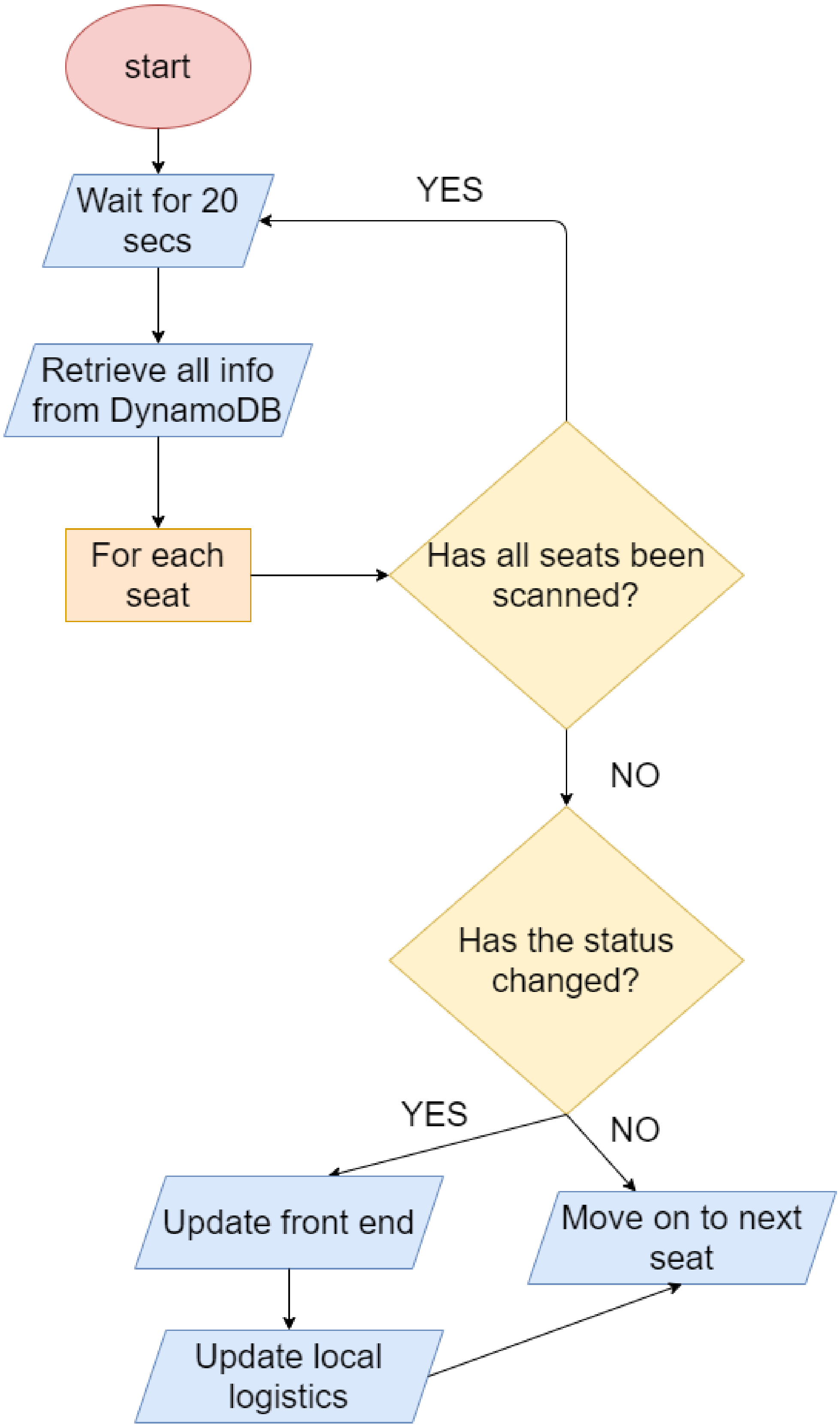

Figure 11. Application Flowchart

Software Module	
Requirements	**Verifications**
1. The software module must reflect the status changes detected by the control module and display such a change to the end user within 30 seconds.	1. Repeatedly occupy a singular seat and leave it and confirm if the status has been reflected to the end user each time.
2. The web interface app must be able to establish a connection with the DynamoDB database.	2. Manually change status values in DynamoDB to see if the status has been reflected to the end user each time.
3. The front end must contain comprehensive internal linking and a custom error page.	3. Click through all combinations of hyperlinks, navigate to invalid pages to ensure custom error page is functional.

Table 6. Software Module Subsystem Requirements and Verifications

2.3 Tolerance Analysis

One difficulty of this project deals with the accuracy of the thermopile sensor. To achieve a 95% accuracy of detection, the thermopile needs to be able to accurately differentiate between body temperature and room temperature 95% of the time, preferably 100%. To find out if this is a requirement that holds, the tolerances of the environment and of each need to be accounted for.

The actual room temperature, T_A, and body temperature, T_O, needs to considered first for the environmental tolerances. The worst-case temperatures would be a high room temperature and a low body temperature. For the purpose of calculations, $T_A = 26\ °C$ and $T_O = 35.5\ °C$ are the assumed worse cases, which is a temperature difference, T_D, of $9.5\ °C$. The room temperature should almost never reach that high assuming a working AC in Grainger and the body temperature is just above hypothermia levels. It is important to note that the worst case T_O does not consider the possible temperature that the sensor might measure if the clothes on the person are a different temperature as the person's body temperature. However, after doing further research, no analysis or studies have been done on measuring the clothes on a human with thermopiles or thermocouples, so this factor was not able to be included in the analysis and will have to be analyzed in testing. Knowing the *actual* T_D will never be less than 9.5 $°C$, the system can be set up to signal human presence when the calculated temperature difference, T_C, is $> 8\ °C$ (to give even more room for error). Therefore, two errors are possible to affect the success rate. Error 1 is if a person is not present but $T_C > 8\ °C$, and error 2 is if a human is present and $T_C < 8\ °C$.

Error 1 is not a calculation error but instead an environmental one. To make sure this error does not occur, it is important to note that the temperature of a vacant spot will always be equal to the room temperature. This assumption can be made because in the worst-case scenario that the temperature of the chair is being read as a human due to the transfer of body heat from the previous user, the sensor can simply be angled in a different fashion to completely avoid the chair and still sense a person. See Figure 12 for an example.

Figure 12. Alternative Sensor Placement

Error 2 is a calculation error. The resistors being used for this circuit have a tolerance of 1%. While this is very small and likely has no effect on the overall calculation anyways, it is useful to take a look regardless. Looking at the equation to find T_o, equation (11), it can be seen that T_o depends on two different resistors and T_A, which depends on R_{th} which depends on four more resistors as seen equation (8). From this system of equations, it can be found that variations in resistors R_1, R_2, R_3, R_4, and even V_{in}. Affect both T_O and T_A by almost the same relationship, with a negligible difference in relationship. Therefore, error in these variables will not cause error 2.

There are, however, variables in, equation (11) that affect T_O that do not affect T_A. These variables are R_5, R_6, and S.

Taking into account the worst-case scenarios for R_5 and R_6, which is an increase in R_5 by 1% and a decrease in R_6 by 1%, the effect on T_O is a decrease of only 0.5%, which is not even close to significant enough to cause error 2.

The only variable left to analyze is the sensitivity coefficient S, which is the most sensitive variable because it is assumed to be constant in order to make calculations possible, but in reality, it changes dynamically with many factors.

$$S = R_{th} N A \varepsilon \sigma F \tag{12}$$

$$\mathrm{N} = \text{Number of thermocouples}$$

$$\mathrm{A} = \text{Seebeck coefficent}$$

$$\varepsilon = \text{Net emissivity}$$

$$\sigma = \text{Stephan's constant}$$

$$\mathrm{F} = \text{Field of view (FOV)}$$

While a number of variables in equation (12) are dynamic, the most important one to keep track of is F, the field of view (FOV) of the sensor.

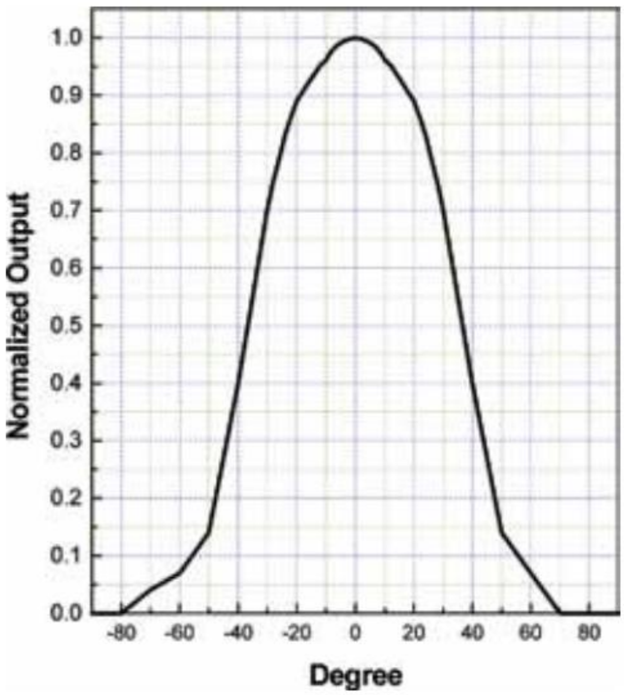

Figure 13. Sensor FOV Plot

Figure 13 [5] represents how V_s changes with changing FOV. To explain the plot, the output of $V_s = 0$ if the object of focus is 70 degrees out of the FOV. If the object is directly perpendicular to the sensor, $F = 100$.

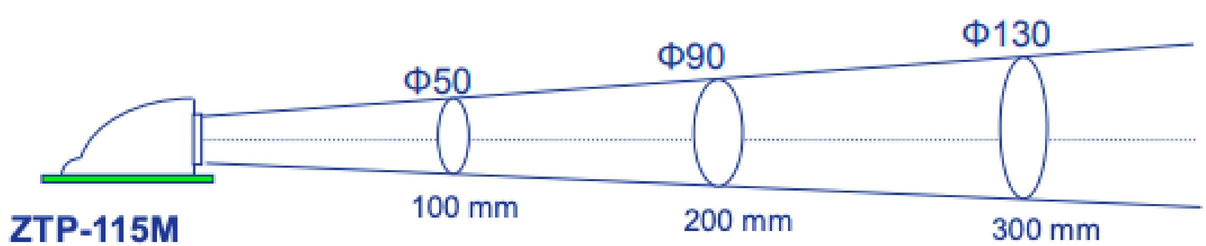

Figure 14. Sensor FOV Diagram

Figure 14 [3] represents the FOV for a related (in the same family) Amphenol sensor. While not the same sensor as the one that used in the system, this figure most likely represents a similar FOV of the ZTP-135SR. From this figure it is evident that the FOV increases linearly with increasing distance, so at a distance of about 600mm minimum, the FOV should be around $\phi 250$. This FOV is more than enough to accurately detect human presence, therefore $F = 100$. In the previous calculation of S that used Figure 6 for its estimation, $F = 85$ [5]. Therefore, S is actually underestimated. According to equation (11), S is directly related to T_O. Therefore, an underestimation of S actually results in an increase in T_C, which will definitely not cause error 2.

3 Cost and Schedule

3.1 Cost Analysis

3.1.1 Labor

Team Member	Dollars per Hour	Hours per Week	Weeks	Multiplier	Total
Owen Brown	$40	15	12	2.5	$18000
Yue Li	$40	15	12	2.5	$18000
Huey Nguyen	$40	15	12	2.5	$18000
Total	$120	45	36	2.5	$54000

Table 7. Labor Cost Breakdown

Table 7 represents the labor cost breakdown for each team member and in total. The hourly wage is a representation of the average starting salary for each of the respective majors.

3.1.2 Parts

Part Number	Description	Manufacturer	Quantity	Unit Cost	Ext. Cost
-	Box PCB	LeMotech	1	$7.19	$7.19
-	Box Sensor	YXQ	1	$11.25	$11.25
C1206C104K5RAC7800	Capacitor 0.1µ	KEMET	10	$0.076	$0.76
C1206C475K4PACTU	Capacitor 4.7µ	KEMET	1	$0.30	$1.54
C1206C106K9PACTU	Capacitor 10µ	KEMET	4	$0.27	$1.08
ESP32-WROOM-32D (4MB)	ESP32 Module	Espressif Systems	1	$3.80	$3.80
NCP1117LPST33T3G	LDO Regulator	ON Semiconductor	1	$0.49	$0.49
APTR3216SURCK	LED Red	Kingbright	1	$0.32	$0.32
102-1031-BL-00100	Mini-B Cable	CNC Tech	1	$2.78	$2.78
2172034-1	Mini-B Port	TE Connectivity	1	$0.98	$0.98
61300211121	Pin Header 2 Position	Würth Elektronik	1	$0.13	$0.13
61300311121	Pin Header 3 Position	Würth Elektronik	1	$0.12	$0.12
SWI5-5-N-I38	Power Adapter	CUI Inc.	1	$6.30	$6.30
CRGCQ1206F	Resistor	TE Connectivity	14	$0.10	$1.40
CP2104-F03-GM	Serial Adapter	Silicon Labs	1	$1.72	$1.72
RA11131121	Switch Rocker	E-Switch	1	$0.57	$0.57
B3F-1000	Tactile Switch	Omron	2	$0.28	$0.56
ZTP-135SR	Thermopile	Amphenol Advanced Sensors	1	$5.11	$5.11
2N3906-AP	Transistor PNP	Micro Commercial Co	1	$0.18	$0.18
2N3904-AP	Transistor NPN	Micro Commercial Co	1	$0.18	$0.18
VS5V0BN1HST15R	TVS Diode	Rohm Semiconductor	3	$0.33	$0.99
20675-2	Wire Sleeving	TE Connectivity	1	$1.24	$1.24

Table 8. Parts Cost Breakdown

Table 8 represents the parts cost breakdown for the SeatDetect system.

3.1.3 Total

Total Cost:

$54,000 (Labor) + $28.16 (Parts) = $54,028.16

3.2 Schedule

Week	Owen	Huey	Yue
3/8	Finalize PCB Schematic Finalize I/O, control unit, and power supply parts Finish ordering parts	Finalize PCB Schematic Finalize I/O, control unit, and power supply parts Finish ordering parts	Finalize PCB Schematic Start UI design for web interface and base for app Finish ordering parts
3/15	Finish PCB layout Start assembling PCB	Finish PCB layout Start assembling PCB	Work on building / designing web app Design software state diagrams
3/29	Work on control unit design Implement power supply with hardware	Work on control unit design Set up database and connect to web app	Finish web app Finish display and I/O testing Set up database and connect to web app
4/5	Finish power supply and control unit implementation Start testing and verification	Finish control unit implementation Assist finalizing and debugging Wi-Fi software	Combine, test, and verify ESP32 Wi-Fi and web app Start testing and verification
4/12	Full-system testing	Full-system testing	Full-system testing
4/19	Work out any bugs/issues and prepare for mock demo	Work out any bugs/issues and prepare for mock demo	Work out any bugs/issues and prepare for mock demo
4/26	Demo, system testing, and start final paper	Demo, system testing, and start final paper	Demo, system testing, and start final paper
5/3	Finish final paper	Finish final paper	Finish final paper

Table 9. Schedule

Table 9 represents the planned schedule for each team member for the remaining 8 weeks of the semester.

4 Ethics and Safety

Ethics and Safety is imperative to successfully carry out the project. During this difficult time of COVID-19, it is especially important that the team follows closely IEEE Code of Ethics #1 [7], that the team does not put other people’s health in harm's way while conducting this project. That means to closely follow the CDC guidelines as well as to build and test the project with as little face to face interactions as possible. When going to the lab is necessary, precautions such as wearing gloves, using hand sanitizer regularly needs to be taken extremely seriously.

In addition to paying attention to safety, it is also important that the team follows the guidelines of IEEE Code of Ethics #5, which suggests that the team needs “to seek, accept and offer honest criticism of technical work, to acknowledge and correct errors, to be honest and realistic” [7]. The team needs to take advantage and make the most out of the weekly TA meetings and be proactive when possible. Furthermore, the team is planning on following this guideline and conducting surveys of the prototype once it starts working to receive feedback to further improve the product. One ethics concern one would consider relates to the issue of privacy. It can be a source of concern because the team is collecting the seat occupancy data for the entire library and understandably people might feel uncomfortable dealing with their occupancy status being collected despite not collecting any personal information. Right now the team does not see a solution that can resolve this concern but will continue exploring options that can make the end users feel more comfortable.

References

[1] D. Roos, “How Parking Garages Track Open Spaces, and Why They Often Get It Wrong,” *HowStuffWorks*. [Online]. Available: https://electronics.howstuffworks.com/everyday-tech/how-parking-garages-track-open-spaces-why-they-often-get-it-wrong.htm

[2] *ParkingDetection*. [Online]. Available: https://www.parkingdetection.com/

[3] *Mouser*. [Online]. Available: https://www.mouser.com/pdfDocs/AAS_Infrared%20Sensor.pdf

[4] *Ametherm*. [Online]. Available: https://www.ametherm.com/thermistor/ntc-thermistor-beta

[5] “ZTP-135SR Datasheet.” Thermometrics.

[6] H. Haraki, “Demystifying Thermopile IR Temp Sensors,” *Fierce Electronics*. [Online]. Available: https://www.fierceelectronics.com/components/demystifying-thermopile-ir-temp-sensors

[7] ieee.org, "IEEE Code of Ethics", 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 08-Feb-2021].