Brilliant Utility for Study Capacity App

(BUSCA)

ECE4011 Senior Design Project

Team Name: BUSCA

Project Faculty Advisor: Linda Milor

Team Members

Aldo Rogliero - Computer Engineering - arogliero3@gatech.edu
Nyle Malik - Computer Engineering - nmalik33@gatech.edu
John Lee - Computer Engineering - johnedwardlee@gatech.edu
JingXuan Wang - Computer Engineering - jwang892@gatech.edu
Salomon Nabine - Computer Engineering - snabine3@gatech.edu

Submitted

January 21, 2020

Table of Contents

Executive Summary

Study space on campus is a limited resource that most students struggle with on a daily basis. Student testimonials have shown that searching for available spaces often wastes valuable study time and usually results in the student being unsuccessful in his or her search. Brilliant Utility for Study Capacity App (BUSCA) solves this problem by implementing a service that allows students to know ahead of time what areas have empty tables on campus. The project involves making an Internet of Things (IoT) low-power embedded device with proximity sensors that is capable of detecting if a table is in use. Each device will then send this data over Wi-Fi to a cloud server. The data will then be accessible through a mobile app, which will display a map showing the available tables by building and floor. With this data, students will then be able to determine which buildings, floors, and tables are available. While university and college campuses are the primary market, BUSCA can also be modified to fit other limited seating scenarios, such as casual dining or entertainment venues. The expected outcome of the design is a functional prototype that successfully detects the availability of a table and updates the mobile app accordingly. The cost of this prototype is expected to be $79.58. The total amount needed to develop the project to completion is $133,925.22.

Brilliant Utility for Study Capacity App

1. Introduction

The Brilliant Utility for Study Capacity App (BUSCA) is study-space availability service. The service is comprised of small, discrete, low-power embedded devices at each study space, an app for displaying study space availability, and a cloud server. The team is requesting $79.58 to develop a working prototype.

1.1 Objective

The objective of BUSCA is to provide students with an easy and convenient way to find available study spaces on campus. The app will show users a color-coded map of the study space they're interested in visiting. As seen in Figure 1, the module containing the sensors and Microcontroller Unit (MCU) will be fitted in a discrete and noninvasive way under tables. The module will connect via Wi-Fi to the cloud server and send status data periodically. The server will handle the data collected from all of the modules by mapping the status of each corresponding module to its location. The BUSCA app on users' phones will connect with the cloud server and display this data in real-time with a user-friendly, color-coded format.

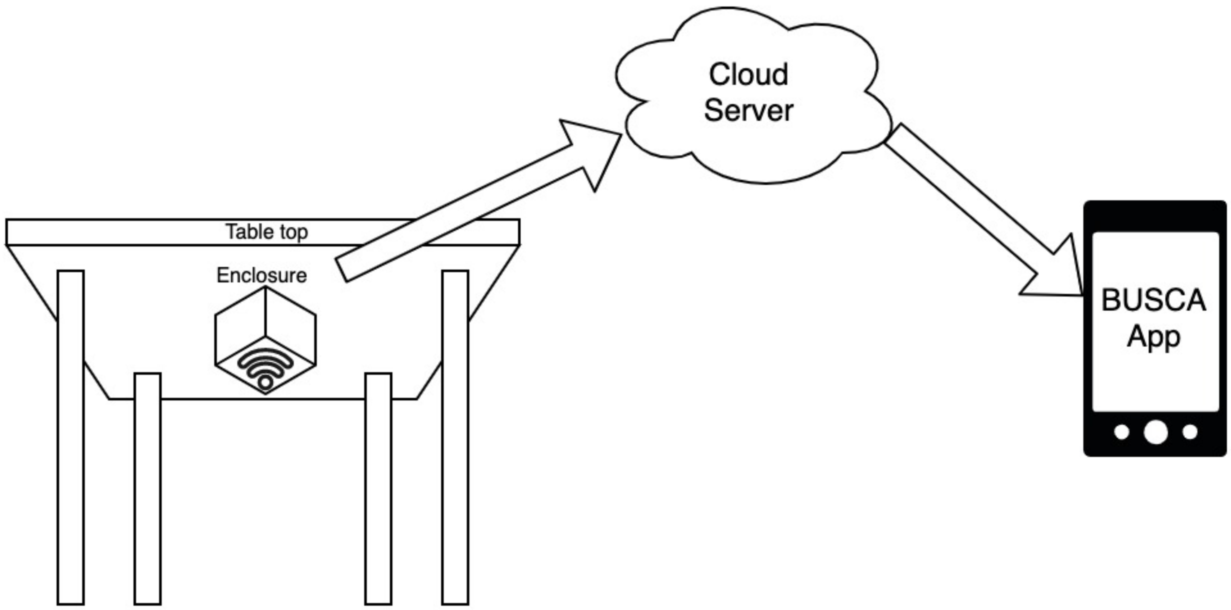

Figure 1. BUSCA components diagram.

1.2 Motivation

The motivation for this project is driven by students' frustration in searching for study spaces in popular buildings on campus. A small survey was conducted on ten random students and 100% indicated having experienced frustration when searching for study spaces. BUSCA aims to solve this by providing students with study space availability information before they spend time searching.

1.4 Background

This project involves four components: a microcontroller unit to handle processing, proximity sensors to obtain data from the environment, network connectivity to handle data transmission, and a cloud server to bridge the gap between devices and end users.

1.3.1 Microcontroller Unit

An MCU is an Integrated Circuit (IC) that contains all the components necessary for computing on a single chip. This includes one or more Central Processing Unit (CPU) cores, memory, and in some cases, a Wi-Fi module [1]. MCUs usually have a 32-bit ARM architecture, which for embedded applications offers the optimal tradeoff between power consumption and performance. The module will run on battery power; therefore, it is necessary to optimize low-power performance to reduce maintenance.

1.3.2 Proximity Sensor

A proximity sensor is able to detect the presence of an object within a range of distances. There are several types of proximity sensors in the market such as Light Detection and Ranging (LiDAR), Passive Infra-Red (PIR), and ambient-light proximity. This project will make use of ambient-light proximity sensors to detect the presence of people in a study space. The sensor detects a proximity signal which, after a burst of current pulses, is received from the infra-red photodiode [2].

1.3.3 Network Connectivity

Network connectivity between the module and the cloud server is essential in the context of this project. This involves having a reliable connection to send each module's status data to the server. The module will connect to the internet through Wi-Fi and send its data reliably using Transmission Control Protocol (TCP). This protocol establishes a connection and transmits data reliably after doing a three-way handshake with the server [3].

1.3.4 Cloud Server

A cloud server performs the function of any web server (hosting websites, storing user data, responding to user requests, etc.) [4] with the only difference being that it is hosted by a cloud service provider (i.e. Amazon Web Services or Microsoft Azure). The advantages of using a cloud server are scalability, cost efficiency, and ease of implementation. Cloud servers are scalable given the tremendous amounts of resources available that can be dynamically allocated as needed. Using a cloud server is cost-efficient because the billing system is usage-based. This means that cloud services users are only billed for the number of requests handled. Cloud service providers handle all of the maintenance and security for their clients, making implementation convenient.

2. Project Description, Customer Requirements, and Goals

The team will design a prototype a battery-powered module containing an MCU, proximity sensors, and Wi-Fi capability. The team will also develop an app for iOS that will connect with the cloud server and display a color-coded map of available study spaces. Figure 2 outlines the tradeoffs in achieving these requirements. The module will be non-invasive, discrete and will not require any input from the people occupying the space other than their presence. It will successfully determine the presence of people at the study space at least 90% of the time, under different scenarios varying the number of people at the space, people walking around the space without occupying it, and different chair positions. The module will be battery powered, and the battery will need to be recharged at most once every month. The mobile app will display the status of each study space updated every ten minutes. Each module needs to be able to connect to the internet in order to send its status data to the cloud server. It will do this through the MCU's built-in Wi-Fi module to connect wirelessly to the campus Wi-Fi network. The module will establish a TCP session with the server and send its data reliably. The features of the system include:

- Reliable data transfer from modules to cloud server
- At least 90% accuracy in table status
- Color-coded map

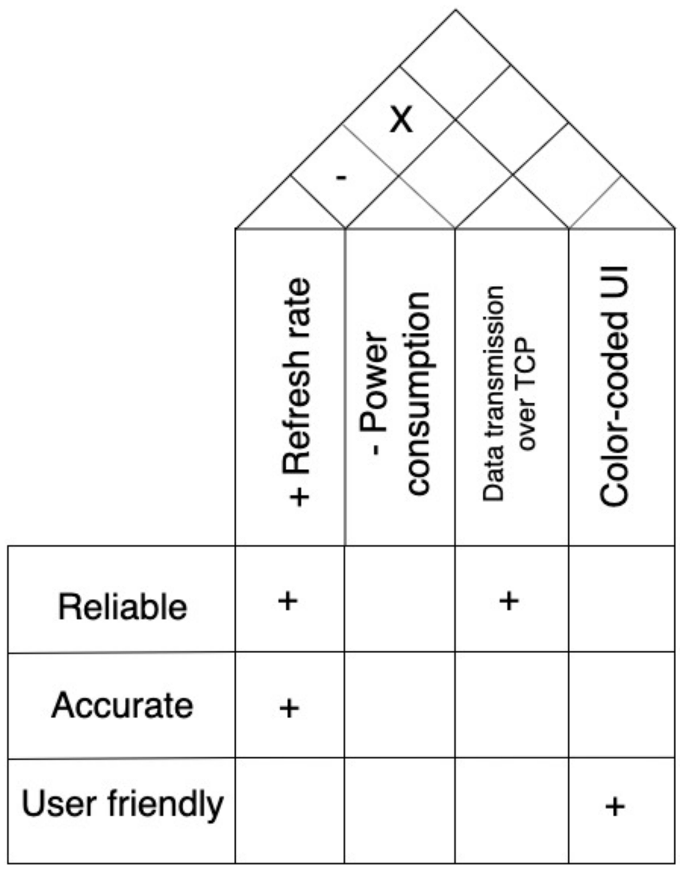

Figure 2. Quality Function Deployment (QFD) diagram.

3. Technical Specifications

The two major hardware components of the system are the MCU and the proximity sensors. Table 1 displays the performance specifications of the MCU, and Table 2 displays the specifications for the proximity sensor. Table 3 outlines the battery specifications that will be used to power these components. As shown in Table 1, the hardware specifications of the MCU need to be adequate to support sensor reading and transmission through the network. The sensor will be connected to the MCU through the Inter-Integrated Circuit (I^2C) interface. Multiple sensors can be used in concurrency to get a wider detection angle. It is important to note that power considerations are vital to the system. Thus, maximizing battery life is a core consideration through all components.

Table 1. Microcontroller Specifications

Feature	**Specification**
Embedded Architecture	Dual-Core ARM Processor
Clock Speed	> 10 MHz
RAM	128 KB
I/O Interface	I^2C
Standby Current Draw	< 10 µA
Active Current Draw	< 200 µA
Network Capabilities	Wi-Fi Network Processor
Network Latency (Round trip time) *	< 10 s
Wi-Fi TX Power	18 dBm
Wi-Fi RX Sensitivity	-95 dBm
Maximum Dimensions	15 mm x 15 mm x 2 mm

* Time to establish a TCP connection with Cloud server, send data, and teardown

Table 2. Sensor Specifications

Feature	**Specification**

Input Voltage	3.3 V
Current Draw	< 100 µA
Response Time	< 5 µs
Minimum Detection Range	20 cm
Minimum Detection Angle	40°
Accuracy within 50 cm	90%

Table 3. Battery Specifications

Feature	**Specification**
Nominal Voltage	3.3-5.0 V
Nominal Capacity	> 1 Ah
Maximum Diameter	14.5 mm
Maximum Length	50.5 mm
Expected life	4 months

4. Design Approach and Details

There are three main components to the design system: the data collection through sensors and MCUs, data transfer and processing on the server, and the mobile app. The sensors will capture data when people are detected around the area and transmit it to the server. The server will process the data and the mobile app will display space availability.

4.1 Design Concept Ideation, Constraints, Alternatives, and Tradeoffs

4.1.1 Data Collection Through Sensors

The sensors should be able to capture data relevant to people's presence in its vicinity. They must also have a connection to the local MCU via I^2C so multiple sensors can be handled by a single MCU. The sensors must be able to run at low power with a quick start-up time in order to prolong battery life.

4.1.2 Microcontroller Unit

The MCU should be able to connect to the server via Wi-Fi and send the data collected by the sensors. It should also have an I^2C interface for data retrieval. An important constraint on the MCU is power consumption. Since they are running on battery, the target is for the system to run for four months (approximately the length of a semester) before maintenance is needed. An alternative to having MCUs directly connecting to the server is to have an edge-device that manages a cluster of smaller, Bluetooth-capable modules in the same general area and sending the collective data from the cluster to the cloud server. Although having an edge-device locally allows processing of data before being sent to the server, its cost is much higher than the original approach and thus is not economically preferable at the current scale.

4.1.2.1 Internet Connectivity

The three main ways of connecting the MCUs to the server are Bluetooth, wired Ethernet cables or Wi-Fi through IEEE 802.11 a/b/g standards. The wireless approach is selected since it prevents having multiple wires laying on the ground and Bluetooth is not used because of its relatively short connection range. Running and processing all the data on the server provides two benefits: it provides a way for us to check and debug the sensors' conditions by analyzing the data, and it leaves more options for data processing since there are no limits on power, current or storage.

4.1.2.2 Power

There are two ways to power the MCUs using batteries: the first one is AA batteries using an off-the-shelf battery enclosure inside of the module. Alternatively, a comparable lithium-ion rechargeable battery may be used. For maintenance purposes, the AA battery enclosure approach is preferable given it's an industry standard.

4.1.3 Cloud Server

Amazon Web Services (AWS) IoT is used as the cloud service provider for this project because of its diverse functionality, programmability and accessibility. MCUs can be programmed with Amazon FreeRTOS -- an operating system that provides native integration with the services and protocols of AWS. The server will use Message Queuing Telemetry Transport (MQTT), which runs on top of TCP, to communicate with the MCUs and transfer information.

4.1.4 Mobile App

To take the social and sustainable impact into consideration, the design ideally supports both Android and iOS users. The Android Software Development Kit (SDK) is used for Android app development and Xcode is used for iOS app development in this project because they are open source and easy to use. Considering the time and labor constraints, only an iOS app will be developed.

4.1.5 Hardware/Software Interactions

The software computing part of the design is performed by analyzing and visualizing data on the server and the mobile app. The hardware component of the design focuses on data collection. The raw data will be captured by each module consisting of one MCU and several sensors, transferred through a Wi-Fi connection to the server.

4.2 Preliminary Concept Selection and Justification

4.2.1 Promising Concepts

There are two promising approaches that will be evaluated. The first is connecting the MCUs directly to the server and the second is connecting the MCUs to an edge-device and then to the server. The two approaches are evaluated by their complexity, sustainability, flexibility, relative accuracy and cost.

4.2.2 Initial Evaluation and Critical Path

The main design limitation at the initial evaluation stage is the sensors' range, accuracy and cost. Without making sure that the sensors will provide valid and accurate data, the processing and visualization part of the design cannot proceed. Although some types of sensors seems to provide a promising range and accuracy, they will be eliminated from the design if their relative cost is economically unacceptable. Therefore, at an early stage of the design, the team will buy and test different sensors and collect data via a serial port to confirm accuracy. After comparing the specifications of different sensors, the team will settle on one sensor and proceed with the design.

4.2.3 Contingency Plan and Potential Risks

If the proposed design approach doesn't work, a wired connection via Ethernet will be considered. Although the complexity of the design will be increased, it will at least provide a stable connection for data to be transferred to the server. Potential risks of using wireless data transfer is that data packets might be intercepted or corrupted during transport. Therefore, security and checksum method may be needed to gather reliable data from the MCUs.

4.3 Engineering Analyses and Experiment

4.3.1 Prototype and Testing

Testing of the sensors will be performed first in the lab with data collected through a serial communication with the MCU. The prototype will consist of a single MCU and several proximity sensors. Testing of the prototype will take place in Clough Undergraduate Learning Commons (CULC) due to its popularity as a study space. Then the data will be transferred and visualized on the server.

4.3.2 Analyses and Experiments to be Performed

During experiments, data will be collected by testing the combination of the MCU and different sensors. Sensor range, accuracy, and power consumption will be analyzed, and an assessment will be performed on each combination. Obstacles and interferences with the sensors will be added to mimic the real study space environment.

4.4 Codes and Standards

1. IEEE 802.11b will be used to connect the MCUs to the server [5]. It features:
    - A maximum raw data rate of 11 Mbit/s and uses the same Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA).
    - Throughput that an application can achieve is 5.9 Mbit/s using TCP and 7.1 Mbit/s using UDP.
2. Inter-Integrated Circuit (I^2C) is a serial protocol shared between MCU and sensors [6]. It features:
    - Transfer rates up to 100 Kbits/s and 7-bit addressing.
    - Clock stretching using SCL and arbitration with SDA.

5. Project Demonstration

The demonstration of the project will take place in the CULC. It will be divided into two main sections: Device and Application demonstration.

5.1 Device Demonstration

The following device features will demonstrate the functionality and the time response of the device. Since power efficiency is an important consideration, this demonstration will also showcase the implementation for a maximized battery life.

- No status change with pass-byers
- Status change of the device upon leaving the table
- No status change upon temporary table departure and return
- Status change with a different number of people sitting at the table (one or more)

5.2 Application Demonstration

The following steps will demonstrate how the mobile application will be used by users to locate the available study spaces. These simulations will also demonstrate how quickly the app updates the status of the tables. In order to decrease user frustration and incorrect information, the mobile application fetches the most recent information for an accurate representation of table vacancy.

5.2.1 Demonstration Steps

- Open the mobile application and select the CULC building
- Use the app to select a particular space in the building
- Occupy the selected space and observe how the app updates within 5 minutes
- Simulate leaving the table and observe how the status changes in the app to “available”
- Simulate a “bathroom break” scenario where the table is abandoned for less than 5 minutes and observe the app status remains “occupied”

6. Schedule, Tasks, and Milestones:

The first few tasks are on-schedule and work has begun. The bulk of the major milestones and scheduled deadlines are during the Spring 2020 semester. Appendix A groups the different future tasks with the task leads and the date assigned. Appendix B is a Gantt Chart including the timeline and milestones. The critical path is highlighted in red.

7. Marketing and Cost Analysis

7.1 Marketing Analysis

The primary market for BUSCA is university and college campuses. Crowded study spaces can reduce the efficiency of students' studying and searching for a non-crowded area can be time consuming [7]. This makes BUSCA a great addition to any campus' study spaces that are frequently crowded. BUSCA could also market to increasingly popular counter-serve, fast-casual restaurants [8]. Like the crowded study space issue, customers would be discouraged by selecting a counter-serve restaurant only to arrive to no seating being available. BUSCA has no direct competitors in either market.

7.2 Cost Analysis

The total development cost of the BUSCA prototype is approximately $79.58. Table 4 shows the component costs for this prototype. The prototype will be developed using a development kit version of the MCU.

Table 4. Equipment Costs

Product Description	Quantity	Unit Price ($)	Total Price ($)
Texas Instruments CC3220SF-LAUNCHXL MCU	1	$49.99 [9]	$49.99
SI1153-AB00-GMR Proximity Sensor	4	$2.14 [10]	$8.56
5-pack SAFT LS14250 1/2AA 3.6v Li-SOCl2 Lithium Batteries	1	$20.99 [11]	$20.99
Amazon Web Services (Usage Based Service)		~$0.0072/Month [12]	~$0.04 (5 Months)
Total Cost			$79.58

Table 5 outlines the development costs of the labor and parts required for project completion. Labor is assumed to be $33 per hour [13]. The Labor Hours column of Table 5 accounts for the number of engineers working on the project.

Table 5. Development Costs (Labor + Part)

Project Component	Labor Hours	Labor Cost ($)
Project Planning and Documentation		
Check the locations where the device will be tested	50	$1,650.00
Order sensors/MCU/batteries	10	$330.00
Test the sensors/MCU/Batteries	50	$1,650.00
Website Development	80	$2,640.00
Final Project (Review of Proposal, Presentation)	80	$2,640.00
Device Implementation (Hardware)		
Work on the device case (design)	60	$1,980.00
Assemble the case	4	$132.00
Mount the device (Sensor, MCU, Battery)	24	$792.00
Device Implementation (Software)		
Create the framework for processing sensor data		
Digitalize the analog input	42	$1,386.00
Filter the input data	130	$4,290.00
Demo with a LED	4	$132.00

Create a framework for updating the table status		
Server Creation and Management	180	$5,940.00
Send the data to the server	52	$1,716.00
Update the tables location status	8	$264.00
Phone App		
Create the building list	120	$3,960.00
Create the building space GUI	180	$5,940.00
Combine the list and the GUI	80	$2,640.00
Update the tables status (Occupied, Available)	30	$990.00
Display table status	10	$330.00
Group Meetings	225	$7,425.00
TOTAL LABOR	1419	$46,827.00
TOTAL PART COST		$79.58
Total Cost (Labor + Part)		$46,906.58

The fringe benefits will be 30% of the total labor cost, and the overhead will be 120% of the total labor, parts, and fringe benefits. Table 6 shows the cost breakdown of the total development costs for the BUSCA project. The total is $133,925.22.

Table 6. Total Development Costs

Parts	**$79.58**
Labor	**$46,827.00**
Fringe Benefits (% of Labor)	$14,048.10
Subtotal	$60,875.10
Overhead (% of Parts, Labor, and Fringe)	$73,050.12
Total	**$133,925.22**

Production will consist of 10,000 units total with about 1,000 units being used for each client. Over the next five years, BUSCA expects to serve at least ten clients. A production version of the MCU will be used instead of the development kit. Assembly and quality testing will be outsourced at a rate of $20 per unit. Advertising will account for 10% of the final selling price as a sales expense. Each device will have a setup charge of $35 and a monthly operation fee of $12. Each client must sign an operation contract at a minimum of one year. This is to ensure profitability for BUSCA and to lower the initial setup cost for the client. The selling price is modeled in Table 7 as the price for a client for one year of operation. The expected revenue per unit is $179 with a profit of $34.77 per unit or 19.42%. Over the five-year production cycle assuming the worst case where each of the ten clients only fulfils the minimum one-year contract, the expected revenue is $1,790,000 with a profit of $340,770. This estimation is drastically improved if a client continues service past the one-year minimum contract length.

Table 7. Selling Price and Profit Per Unit Per Year

TI CC3220SF MCU	$7.48 [14]
SI1153-AB00-GMR Sensor	$8.56
LS14250 Li-SOCl2 Battery	$4.20
Additional Wires and Case	$5
AWS for 1 Year of Normal Operation	$0.09
Assembly Labor	$10
Testing Labor	$10
Total Labor	$20
Fringe Benefits (% of Labor)	$6
Subtotal	$51.33
Overhead (% of Parts, Labor, and Fringe)	$61.60
Subtotal (Input Costs)	$112.93
Sales Expense	$17.90
Amortized Development Costs	$13.40
Subtotal (All Costs)	$144.23
Setup Fee	$35

Operation Fees (1 Year of Operation)	$12
Profit (1 Year of Operation)	$34.77
Selling Price	**$179.00**

8. Current Status

The project is at the 40% mark of the Project Planning and Documentation phase. Research has been conducted in order to identify the locations for prototype testing. The sensors, microcontroller, and batteries are ready to be ordered. Upon arrival, testing the various possible sensors with the MCU can begin. When completed, the project will move forward to the Device Implementation (Hardware & Software) phases.

9. Leadership Roles

Team leaders for major tasks are assigned as follows:

App Development & Server Side IoT

- Nyle Malik
- Aldo Rogliero

Embedded Hardware/Software & Power

- John Lee
- JingXuan Wang
- Salomon Nabine

Webmaster

- Aldo Rogliero
- Nyle Malik

Expo Coordinator

- John Lee

Documentation

- JingXuan Wang
- Salomon Nabine

10. References

[1] A. Fant, P. Torta, L. Dörrer and L. Sant, "A system containing an ambient light and a proximity sensor with intrinsic ambient light rejection," *2012 Proceedings of the ESSCIRC,* 17-21 September 2012.

[2] A. Singh, Microcontroller and Embedded System, New Age International, 2008.

[3] P. Fox, "Khan Academy," [Online]. Available: https://www.khanacademy.org/computing/ap-computer-science-principles/the-internet/tcp-fault-tolerant-transmission-protocol/a/transmission-control-protocol-tcp. [Accessed 16 November 2019].

[4] T. G. Peter Mell, "The NIST Definition of Cloud Computing," September 2011. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf. [Accessed 16 November 2019].

[5] "IEEE 802.11TM WIRELESS LOCAL AREA NETWORKS," *IEEE 802.11, The Working Group Setting the Standards for Wireless LANs*. [Online]. Available: http://www.ieee802.org/11/. [Accessed: 16-Nov-2019].

[6] "I2C Bus Specification," *I2C Info – I2C Bus, Interface and Protocol*. [Online]. Available: https://i2c.info/i2c-bus-specification. [Accessed: 16-Nov-2019].

[7] A. Raychawdhuri, Raychawdhuri, A. Raychawdhuri, and A. Raychawdhuri, "Lack of study spaces at UCLA imposes unnecessary stress on students," *Daily Bruin*. [Online]. Available: https://dailybruin.com/2019/02/28/lack-of-study-spaces-at-ucla-imposes-unnecessary-stress-on-students/. [Accessed: 18-Nov-2019].

[8] T. Chef, "Fast Casual vs. Fast Fine Dining Trends," *Restaurant Insider*, 11-Feb-2019. [Online]. Available: https://upserve.com/restaurant-insider/fast-casual-fast-fine-dining. [Accessed: 18-Nov-2019].

[9] "CC3220SF-LAUNCHXL," *CC3220SF-LAUNCHXL | SimpleLink™ Wi-Fi® CC3220SF wireless microcontroller LaunchPad™ development kit | TI store*. [Online]. Available: https://www.ti.com/store/ti/en/p/product/?p=CC3220SF-LAUNCHXL. [Accessed: 18-Nov-2019].

[10] "SI1153-AB00-GMR Silicon Labs: Mouser," *Mouser Electronics*. [Online]. Available: https://www.mouser.com/ProductDetail/Silicon-Labs/SI1153-AB00-GMR?qs=sGAEpiMZZMs3uAJYYmvlKwH/cpsSORxEcPk1oScwvp2TPI9RLOujTw==. [Accessed: 18-Nov-2019].

[11] “5 SAFT LS14250 LS 14250 1/2 AA 1/2AA 3.6v Li-SOCl2 Lithium ...” [Online]. Available: https://www.amazon.com/LS14250-Li-SOCl2-Lithium-Batteries-FRANCE/dp/B00KLEU8PW. [Accessed: 18-Nov-2019].

[12] D. Borycki, “Programming for the Internet of Things: using Windows 10 IoT core and Azure IoT Suite,” *Amazon*, 2017. [Online]. Available: https://aws.amazon.com/iot-core/pricing/?nc=sn&loc=4. [Accessed: 18-Nov-2019].

[13] Salary.com, “Hourly wage for Software Engineer I,” *Salary.com*. [Online]. Available: https://www.salary.com/research/salary/benchmark/software-engineer-i-hourly-wages. [Accessed: 18-Nov-2019].

[14] “CC3220SF ACTIVE This product has been released to the market and is available for purchase. For some products, newer alternatives may be available. SimpleLink Wi-Fi® and IoT, Single-Chip Wireless MCU Solution,” *CC3220SF SimpleLink Wi-Fi® and IoT, Single-Chip Wireless MCU Solution | TI.com*. [Online]. Available: http://www.ti.com/product/CC3220SF. [Accessed: 18-Nov-2019].

Appendix A - Project Timeline and Tasks Breakdown

Task Name	Task Leads	Start date	End date
Project Planning and Documentation	All (SN*)	1/6/2020	4/23/2020
Check the locations where the device will be tested	All (SN*)	1/6/2020	1/13/2020
Order sensors/MCU/batteries	All (AR*)	11/20/2019	12/1/2019
Test the various sensors/MCU/batteries	All (JW*)	1/6/2020	1/13/2020
Website Development	All (NM*)	4/5/2020	4/16/2020
Final Project (Review of Proposal, Presentation)	All (SN*)	4/5/2020	4/16/2020
Expo Preparation (Oral Presentation, Participate in Expo)	All (JL*)	4/16/2020	4/23/2020
Device Implementation (Hardware)	JL*, SN, JW	1/13/2020	1/31/2020
Work on the device case (design)	JL, SN*, JW	1/13/2020	1/25/2020
Assemble the case	SN, JW*	1/25/2020	1/27/2020
Mount the device (Sensor, MCU, Wi-Fi, Battery)	SN, JW*, JL	1/27/2020	1/31/2020
Device Implementation (Software)	All	1/31/2020	4/5/2020
Create framework for processing sensor data	JL, SN, JW*	1/31/2020	3/16/2020
Digitalize the analog input	SN*, JL, JW	1/31/2020	2/7/2020
Filter the input data	SN, JW*, JL	2/7/2020	3/16/2020
Demo with a LED	SN*, JW	3/15/2020	3/16/2020
Create framework for updating the table status	AR*, NM	1/13/2020	3/16/2020
Server Creation and Management	AR, NM*	1/13/2020	3/16/2020
Send the data to the server	AR, NM*	3/1/2020	3/16/2020
Update the tables location status	AR*, NM	3/15/2020	3/16/2020
Mobile App	All (NM*)	1/13/2020	4/5/2020
Create the building list	AR, JW*	1/13/2020	2/22/2020
Create the building Space GUI	NM, JL*, SN	1/13/2020	2/22/2020
Combine the list and the GUI	All (AR*)	2/22/2020	3/16/2020
Update the tables status (Occupied, available)	AR*, NM	3/16/2020	4/4/2020
Display tables status	All (NM*)	4/4/2020	4/5/2020

* Denotes team lead for that task

AR - Aldo Rogliero, NM - Nyle Malik, SN - Salomon Nabine, JW - JingXuan Wang, JL - John Lee

Appendix B – Project Gantt Chart