Desk Reservation System

Team 37: Pragya Aneja, Ninad Godbole, Varad Khandelwal

ECE 445 Project Proposal - Spring 2020

TA: Shaoyu Meng

Table of Contents

1. Introduction

1.1 Objective

As a student, libraries are a fundamental part of college life. It is therefore vital that the student is able to find a place to sit to be able to study effectively. However, due to overcrowding it can become very difficult to find a spot, and even more so if one has a group study session. Additionally, as different students have vastly different timetables, coming early into the library to save a spot is also hard. Furthermore, it can regularly happen that even though there are some desks available, it takes a lot of time to scan the entire library to find it. For example, in Grainger Engineering Library there are 5 floors and to find a desk to sit at can take a lot of time which then reduces the student's time to study. This problem only exacerbates during exam time, where finding alone, distraction free time becomes increasingly more of a necessity. As can be seen, there are a variety of issues that are involved when it comes to a student being able to reliably find a spot to study in a place like the Grainger Engineering Library.

Hence, our main objective and goal for this project is to reduce the inefficiency involved in a student trying to find a spot to study. Our solution consists of designing a desk reservation system where one can reserve a desk in advance at a specific time for upto a few hours. This system will increase space efficiency for a library, allow the student to more conveniently find a place to study and allow the student to make more prudent use of his/her time.

1.2 Background

Today, reservation systems are used in a plethora of settings and contexts and have been for a while. One example of this is in the hospitality industry, i.e. booking rooms for a hotel. By allowing customers to use an online reservation system to choose the type of room and duration of stay, the customers have a more streamlined experience [1]. Another example of this is setting up an appointment to get a haircut. Scheduling the haircut beforehand increases efficiency by making it so that there isn't overcrowding and large wait times in the hair salon [2]. The last example worth discussing is that of reserving study rooms for libraries for universities, which is probably the most related to our project. Similar to the first example regarding the hospitality industry, giving the students the option to reserve study rooms improves their experience.

All these examples are inspiration for the desk reservation system project, which incorporate aspects from the examples mentioned above like improved efficiency through reduced overcrowding and waiting times as well as a better overall experience for the student. The concept of reserving a specific desk in a library isn't that popular most probably because of the cost required due to each desk requiring some sort of communication subsystem. Therefore, our project is trying to look into creating this system while keeping the cost as low as possible.

1.3 Comparing the Previous and Current Projects

When we consider the previous project and our version of the project, there are three aspects that differentiate the two which include the cost, inefficiency, and the design choice of the communication subsystem.

For the previous project the cost was extremely high. From a preliminary look at the parts used, the cost ranged from 40-50$ for each desk. Assuming around 1000 desks in the library, scaling the system for this amount of would cost a range of 40000-50000$. Expanding this to all the libraries in UIUC would balloon the cost up to the hundreds of thousands of dollars. Our approach improves upon this greatly by using only a bluetooth module and battery as parts for each desk which costs around 10$. A led display, microcontroller and Wi-Fi module are only needed for every 64 desks, which cost around 50$. This means the cost for 1000 desks approximates to 10900$. This is a great improvement in the form of a 80% reduction in the cost of the original.

Efficiency and convenience of use for the student was another facet that we improved upon. An example of this is that the old project enforced the student to leave the desk after an hour which was done through the use of sensors. However, this is inefficient with regards to space because if a given desk isn't actually reserved, the student should not just be forced to leave. We therefore decided to alert the student using a status LED that lights up red when the desk is currently being reserved for someone else. This way any student can use any desk as long as they want without reserving it until the status LED.

One of the biggest design changes we made from the previous project is the way we handle communication. In the previous project, there is use of an LCD screen and Wi-Fi module to show the status of the reservation which is essentially a one to one communication between the application and desk. In our approach, the communication occurs through Wi-Fi to a Master unit to a Piconet network. The Piconet consists of master-slave units of bluetooth and works in a one way direction, where the master unit floods a message to all the slave I units it is connected to and these slave I units further flood the message down to all the slave II units.

1.4 High-Level Requirements

1. Within a span of 15 seconds, the student must be able to reserve a particular desk from a desk map corresponding to the chosen time slot and receive confirmation of the reservation with a unique 6 digit reservation ID. The student can also cancel this reservation until 15 minutes of the reservation time.
2. The student must be able to input the 6 digit reservation ID into the master unit via keypad, which should then map to the corresponding desk and turn on the status LED on that desk red within 5 seconds.

3. After 15 minutes, if the student does not confirm the reservation, the central system must add the UIN to a database which it keeps track of. The student must be penalized after the third non-confirmation by removing his/her right to reserve a desk for a period of a month.

2. Design

2.1 System Overview (Functional overview)

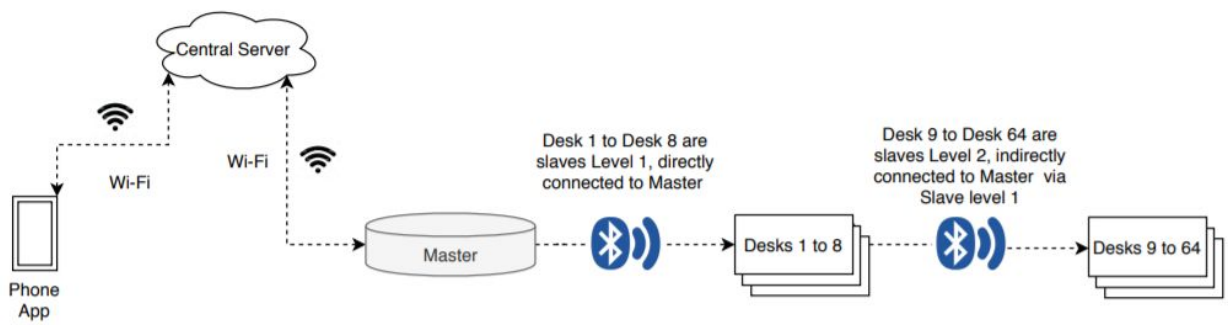

Figure 1: Top-level System Design

The desk reservation system consists of three components: a phone app as the user interface, a master unit which controls reservations for 64 desks, and the desk itself. Whenever a user wants to reserve a desk, he/she must open the phone app and input the timings for which a desk should be reserved, in addition to his/her UIN. This information would then be communicated via WiFi to a central server, which would then return an accurate seat map of the library with desks available for reservation marked green. The user could now select any desk marked green, and upon receival of this information at the central server, the server would generate a unique reservation ID mapped to the user's UIN. This reservation ID along with the desk ID and the timings of the reservation would be broadcasted via WiFi to all the master units in the library and also sent to the user for later use. The master unit must be aware of the ID's of all 64 of its slave desks, and thus would only accept the broadcasted message if the desk ID in the message belongs to one of its slave desks. Thus, the user will have the reservation ID associated with this reservation stored in the app, and the master unit in charge of the desk will have the reservation ID, timings and desk ID of the reserved desk.

At the time of the reservation, the user must arrive at the library, walk upto the master unit associated with his/her reserved desk, and feed the reservation ID received earlier into the master unit's numeric keypad. The master unit would send the initiation information to the central server, so that if the reservation was initiated more than 15 minutes after the start time of the reservation, a penalty point would be added to the UIN associated with this reservation. This would allow us to discourage false bookings, as we would not allow any user with over 3 penalties to conduct any reservations for 2 weeks. The master unit would then display the desk ID associated with this reservation to the user via an LED screen, and communicate the desk ID to all the slave-I desks via Bluetooth. If the desk ID in the message matches the ID of one of these level-1 slave desks, a status LED placed on the corner of the desk would light up red. If not, each level-1 slave would pass on the message to all eight of its level-2 slave desks, and if the desk ID in the message matches any desks' ID, the status LED on the corner of that desk would turn red. The LED turning red would act as an indicator for anyone sitting on the reserved desk to leave, as the reservation has now been initiated. Then the user would walk upto the desk and use it for the allotted time.

The master unit would also check for change of status in any of slave desks' reservations by tallying the current time with the timings of the reservation every 15 minutes. This would allow it to track if any reservations that were active earlier have now ended. It would send the updated status of all reservations to its slave desks to ensure that the status LED for any desk that had turned red at the initiation of a reservation turns off within 15 minutes of that reservation ending.

2.2 Block Diagram

2.2.1 Master Unit

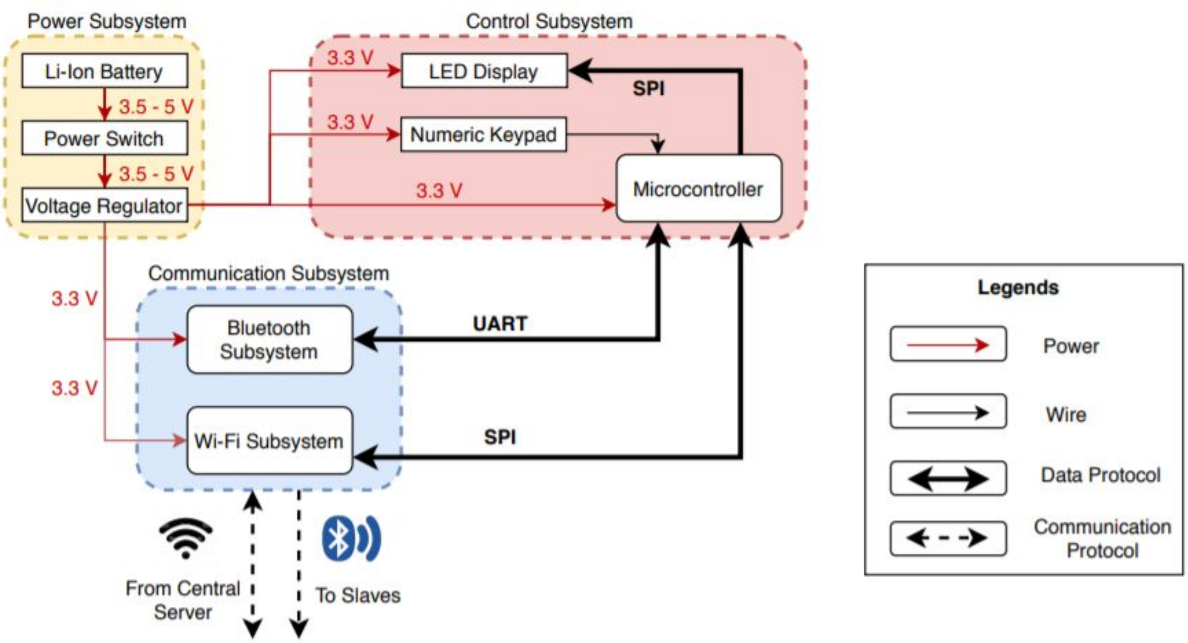

Figure 2: Master Unit Block Diagram

2.2.2 Slave Units - Level 1

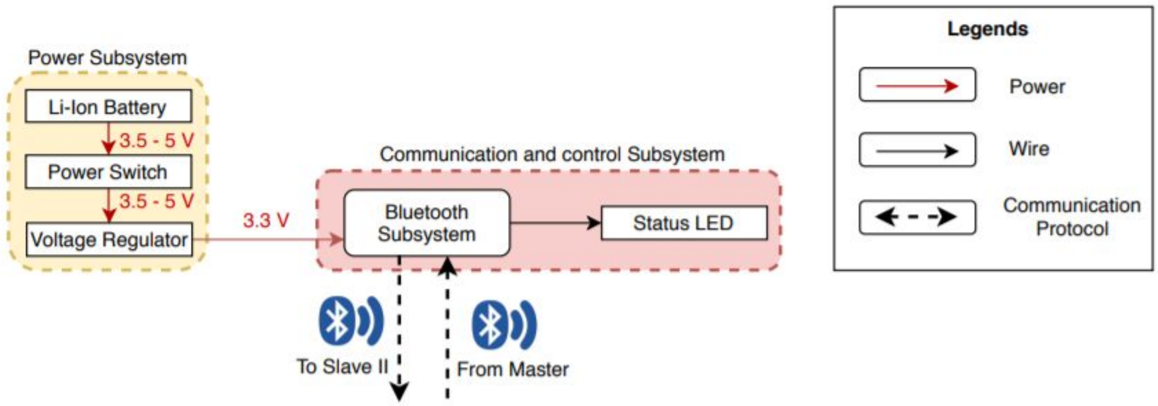

Figure 2: Slave Unit Level 1 Block Diagram

2.2.3 Slave Units - Level 2

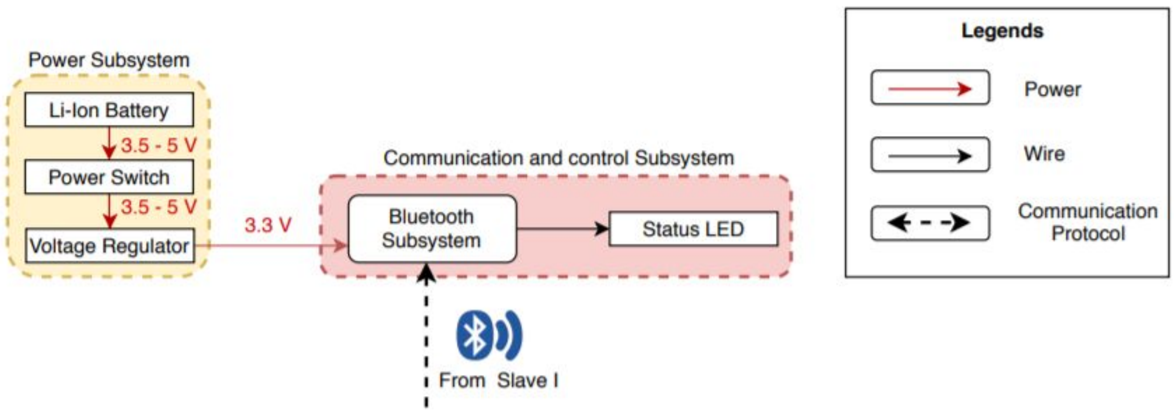

Figure 3: Slave Unit Level 2 Block Diagram

2.3 Physical Diagram

2.3.1 Master Unit & Slave Units I, II

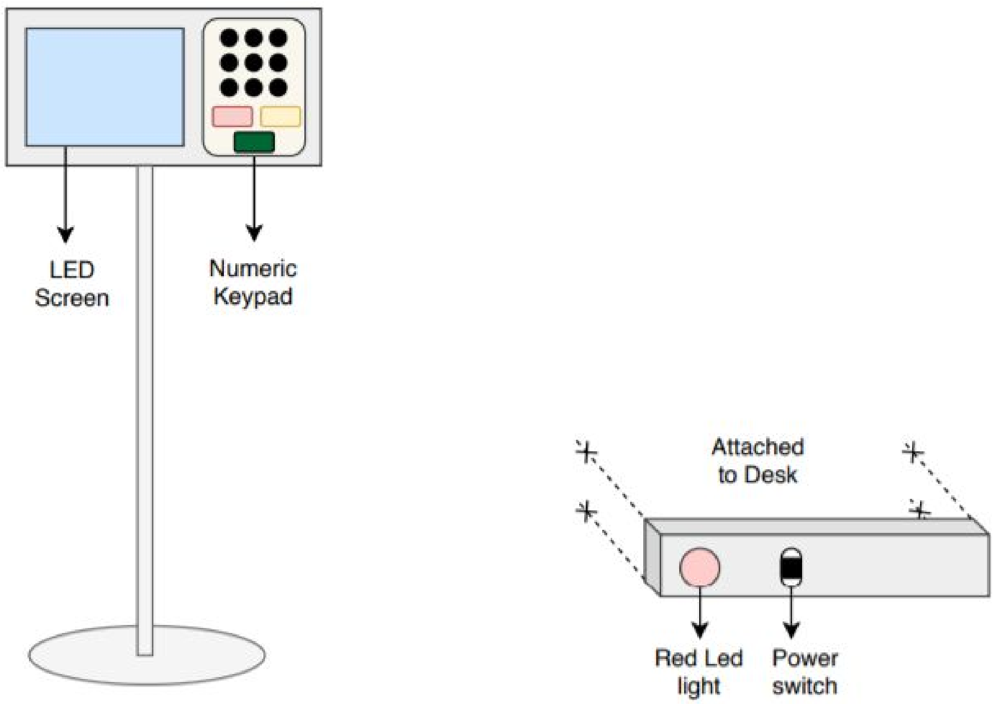

Figure 4: Physical Diagram of Master Unit (seen on left) and Slave Unit (seen on right)

2.4 Master Unit Subsystems

2.4.1 Power Subsystem

Li-ion battery

The lithium-ion battery must be able to keep the circuit continuously powered when switched on for 2-3 days. Additionally, the voltage regulator would drop some voltage in converting higher voltages to 3.3 V which is needed by all the components (Microcontroller, LED display, Wi-Fi and Bluetooth system), so the power supply should be able to provide 3.5-5V. Additionally the battery must be rechargeable.

***Requirement 1*:** Battery should be able to provide a voltage output of 3.5-5 V.
Requirement 2: Battery should be able to store > 2500 mAh.

Requirement 3: Battery should be able to provide charge for 2-3 days and be rechargeable.

Power switch

We don't want the device to run indefinitely, and would like to switch it off if the library is on a break or not in use. Thus, we need a switch which disconnects the voltage regulator from the battery when switched off.

Requirement 1: A switch which disconnects the voltage regulator from battery when off and provides very less voltage (<= 0.2 Volts) drop when switched on.

Voltage regulator

As we are using Li-on batteries, the voltage across will degrade over time and the battery won't provide the voltage level needed for all circuits. So, we would need a voltage regulator which would provide a constant 3.3V to all the components.

***Requirement 1*:** Can output stable 3.2-3.3V from an input supply of 3.5-5V.
Requirement 2: Maintains thermal stability below 50°C.

2.4.2 Control Subsystem

Microcontroller

The microcontroller should be compatible with the bluetooth module, WiFi module, LED Display and numeric keypad. It should communicate with the bluetooth module via UART and the LED Display and WiFi module via SPI.

1. It should be able to receive reservation ID along with the desk ID and the timings of the reservation via SPI from the WiFi module which would receive this information from the broadcast by the central server.
2. It should have knowledge of the ID's of all 64 of its slave desks, and thus should only care about reservation messages that contain one of its slave desks' ID.
3. It should take the reservation ID input from the numeric keypad and should send the desk ID associated with it via SPI to the LED display.
4. It should send the desk ID whose reservation has just been initiated to all of its slave desks.
5. Every 15 minutes, it must tally the current time with the timings of all reservations it received from the central server. Then it should send a bluetooth message to all of its slave desks containing information of the desks whose reservation has ended, and those desks should be removed from its list of reservations.

***Requirement 1*:** The microcontroller must be able to communicate over both UART and SPI simultaneously and interact with different subsystems.

***Requirement 2*:** Must have memory storage of ~15kB for accommodating a reservation list for 64 desks assuming each desk must be reserved for a minimum of 30 minutes where each entry in the list is 4 bytes (64x48x4 = 12288 bytes).

LED Display

The LED must display the desk ID associated with the reservation ID input by the user. It receives this desk ID from the microcontroller via SPI.

***Requirement 1*:** The LED must clearly display the 2 digit desk ID.
***Requirement 2*:** Must be able to receive input from microcontroller via SPI.

Numeric Keypad

The numeric keypad must be able to take a 6-digit reservation ID as input from the user, and pass the information to the microcontroller.

***Requirement 1*:** The numeric keypad must be a 10-digit keypad.

2.4.3 Communication Subsystem

Bluetooth Module

Bluetooth Module in the master unit would need to be able to connect to 8 Slave level I, and so it should be able to maintain 8 simultaneous connections. It doesn't need to be very cost effective as it will only be used once per 64 desks and at the same it has to be moderately energy efficient as the master unit has a huge power supply. Additionally, it needs to be able to communicate over UART/SPI protocol to be able to receive information from the microcontroller.

***Requirement 1*:** It should be able to connect to 8 bluetooth modules simultaneously.
Requirement 2: Consumes less than 30 mAh power at peak consumption and operates at 3.3V.
Requirement 3: Should be able to communicate over UART or SPI protocol with a microcontroller.

Wi-Fi Module

Wi-fi in the master unit would need to perform basic data retrieval from the central server and be moderately energy efficient to ensure the master works over 2-3 days on the battery. It would also need to communicate with a microcontroller to send the information received from the central server. Additionally, after it restarts, it should be able to ping the central server with its updated IP address.

***Requirement 1*:** It should be able to receive Update requests from the central server, and be able to send POST requests with its IP address.

Requirement 2: Consumes less than 50 mAh power at peak consumption and consume less than 10 mAh during sleep mode.
Requirement 3: Should be able to communicate over SPI protocol with the microcontroller.

2.4 Slave I & II Unit Subsystems

2.4.1 Power Subsystem

Li-ion battery

The lithium-ion battery must be able to keep the circuit continuously powered when switched on for 6-7 days. Additionally, the voltage regulator would drop some voltage in converting higher voltages to 3.3 V which is needed by all the components (LED light and Bluetooth system), so the power supply should be able to provide 3.5-5V. Additionally the battery must be rechargeable. However, here the battery will be light weight compared to the Master unit as the battery only needs to power the Bluetooth submodule.

Requirement 1**:** Battery should be able to provide a voltage output of 3.5-5 V.
Requirement 2: Battery should be able to store > 1200 mAh.
Requirement 3: Battery should be able to provide charge for 5-6 days and be rechargeable.

Power switch

We don't want the device to run indefinitely, and would like to switch it off if the library is on a break or not in use. Thus, we need a switch which disconnects the voltage regulator from the battery when switched off.

Requirement 1: A switch which disconnects the voltage regulator from battery when off and provides very less voltage (<= 0.2 Volts) drop when switched on.

Voltage regulator

As we are using Li-on batteries, the voltage across will degrade over time and the battery won't provide the voltage level needed for all circuits. So, we would need a voltage regulator which would provide a constant 3.3Vs to all the components.

Requirement 1**:** Can output stable 3.2-3.3V from an input supply of 3.5-5V.
Requirement 2: Maintains thermal stability below 60°C.

2.4.2 Control and Communication Subsystem

Bluetooth Module

The bluetooth module used in the slave module, which will be at desk, will be the main component of our system and would need to perform several tasks and meet different requirements to make the project successful.

1. It should be really cheap as it would be used at every desk and hence a $1 increase in the cost would correspond to a $1000-1200 in the project cost for a library like grainger. For UIUC, a $1 increase in bluetooth module's cost would correspond to an increase of $40,000-50,000.
2. It needs to be very power efficient as we want each slave to work on battery for upto 5-6 days. The module should consume extremely less power in sleep mode, and minimal power in active mode.
3. It has to also connect to 8 devices simultaneously and maintain the connection while in operation.
4. It needs to be able to turn an LED light on/off without the help of a microcontroller and it should be able to identify if a particular bluetooth message corresponds to itself or should be flooded forward. This means it should be able to perform bare minimum control logic operations.

***Requirement 1*:** Bluetooth module should cost less than $5 and should operate for at least 6months without replacement.
Requirement 2: It needs to consume less than 1-2 mAh in sleep mode and less than 5 mAh during active mode. It should also operate on 3.3 V.
Requirement 3: Can maintain connections with 8 bluetooth modules.
Requirement 4: Have 1 I/O pin and a basic control flow to filter incoming bluetooth messages.

Status LED

The status LED on the corner of the reserved desk must light up when instructed by the bluetooth module, to indicate that the reservation for this desk has been initiated and anyone else using this desk must leave.

***Requirement*:** Must be red in colour in order to be clearly visible from 1 m distance.

2.5 Software

2.5.1 Phone Application

The phone application will be the primary means for the student to be able to manage his or her reservations. The main screen of the app will show the map of the desks for the student and will be highlighted red if they are reserved and green if they are free. If a desk is free, the student can tap on it and select the time frame that he/she wants it for with a limit of upto 4 hours. After the student reserves the desk, there will be a unique reservation id that will be prompted from the app, which the student will then have to go enter at the master unit within 15 minutes. The student can also use the app to cancel the reservation within the first 15 minutes if something else comes up. There will be another page on the app that the student can go to, to see how many times a reservation has been missed, since after the third violation, the student will be penalized.

***Requirement 1*:** Phone application should allow the student to reserve a desk for upto 4 hours at a specific time and provide the student with a unique 6 digit reservation id.
***Requirement 2*:** Phone application should allow the student to cancel the reservation within 15 minutes of having reserved it.
***Requirement 3*:** Phone application should show student's history of missed reservations.

2.5.2 Central Server

The central server acts as the main entity for controlling the desk reservation system. It performs a few critical tasks which allows the system to run and it is hosted at a fixed web address. This web address is used by the master unit and the phone application to send API requests.

1. It contains the desk map for the entire library and for each desk contains its future reservations schedule in a database. When the phone app sends a query for a reservation for a particular duration of time, the central server scans the database and based on the reservation schedule for each desk marks it as available or taken. Then it composes it into a message and sends it back to the phone app.
2. When the phone app sends a confirmation for reservation, the central server updates the database by adding this reservation to the corresponding desk entry and generates a random 6 digit code. This 6 digit code with the desk ID and time is composed as a message and is sent to the user and broadcasted to all the masters.
3. When a master unit restarts, it sends a post request containing its new IP address to the central server and the central server updates its directory of IP addresses of master units. The directory is used to flood reservation confirmations to the master unit.
4. Central server also maintains a penalized UIN list, to deny them reservation rights. It maintains this with a database where each UIN is a row ID, and the number of missed reservations is a column. It also maintains a timer for users who have been banned from reserving and when the timer becomes zero, resets their missed count to zero. After every semester, the missed count becomes zero for everyone.

***Requirement 1*:** Be able to receive POST and GET requests from phone app and master units.
***Requirement 2*:** Be able to maintain 4 separate databases.

Requirement 3**:** Be secure by not compromising student data and not completing API calls from non-master units.

2.6 Risk Analysis

The biggest obstacle and the trickiest part of the project involves the communication subsystem, particularly, bluetooth subsystem which involves formation of piconet with 64 slave units and 1 master unit. The master unit bluetooth submodule benefits from larger battery and lower cost constraints, however, the slave bluetooth submodule needs to be cheap and energy efficient. For it to be energy efficient, it needs to be able to go in and out of sleep mode while maintaining the bluetooth connection. The sleep mode needs to consume relatively no power otherwise the battery would be drained very quickly. In a library with 1000s of desks, if the battery needs to be replaced daily it makes the entire system unfeasible. Additionally, during sleep mode, it also needs to maintain the existing bluetooth connections with master and Slave II or Slave I (depending on where it is placed) as reestablishing bluetooth connection is a power consuming process.

Even if through smart implementation of bluetooth piconet we can reduce power consumption, another fundamental challenge comes from making the bluetooth module really cheap as at the same time it needs to be capable of performing basic I/O and logic operations. As there is no microcontroller, when the master sends out a broadcast to all slaves with a particular desk ID to activate a reservation the bluetooth module at Slave level I should be able to filter the broadcast and match its own desk ID with the one mentioned in the broadcast.

Another challenging aspect comes from the piconet itself. Implementation of piconet, particularly forwarding of messages from one level to another level can be really difficult as it has not been covered in any of the team’s classes or internships. Even if the piconet gets established, it needs to be robust, as when one slave restarts it should be able to reconnect itself to the previous bluetooth connections. For example, a slave in level-I would need to pair itself to the master first, and then to all the other 7 slaves in level-II. This has to be also done without the help of a microcontroller. In real life, bluetooth mesh networks are frequently used in IOT projects in industrial and home application. This means there is necessary documentation to understand the implementation, however, as we need to modify those implementations to our specific project, the existing documents might not provide us solutions directly. Overall, due to several constraints and requirements this makes the bluetooth subsystem the hardest part of our project.

3. Safety and Ethics

The IEEE code of ethics which are made up of 10 guidelines as well as the ACM code of ethics provided a good framework for us when we consider the ethical aspects with regards to our project. From this IEEE and ACM ethics framework, there are a couple of ethical issues that are worth discussing.

A few ethical issues for our project correspond to one of the ACM code of ethics which says "Be fair and take action not to discriminate" [3]. The ethical issues here correspond to some of the parts or subsystems which could potentially be faulty, leading some students to not be able to use the reservation system properly which would be unfair to them. One example of this is if the status LED on the desk does not light up red as it has stopped working. This could confuse the student as to the status of the desk and worsen the experience. Another example is if the numeric keypad stops working properly. This would become very problematic as this is what the student uses to confirm the reservation ID and so if it doesn't work the student won't get access to the desk. There are other parts that if they stop working will create an unfair experience for some students who went through the correct process of reserving a desk through this system. We will therefore strive to overcome these issues by making sure that the parts are high quality and tested thoroughly before placed in use.

Another ethical issue that should be brought up is students potentially taking advantage of the system which relates to one of the ACM code of ethics which says "Be honest and trustworthy" [3]. An example of this is if a student decides to reserve a desk without really having the intention of using the desk but just as a backup and then does not show up to the reservation, causing someone else to miss out. An additional ethical issue which is extremely important involves the data security of the students who are using the phone application as well as their UIN when reserving the desks, which corresponds to one of the ACM code of ethics which says "Respect privacy" [3]. We will therefore make it a priority to make sure all of our systems keep the data of the students secure.

Safety is the other aspect of this project that we have to be aware of which also relates to the first IEEE code of ethics which says "to hold paramount the safety, health, and welfare of the public" and one of the ACM code of ethics which says "Avoid harm" [3][4]. The biggest safety issue with regards to this project involves our usage of the lithium ion battery. As usage of these batteries increases, they begin to overheat and in a low likelihood scenario, can catch on fire putting a student in a lot of harm if he or she is near the battery [5]. We therefore have to be aware of the safety hazard of the lithium ion battery in our design.

4. References

[1] “How do hotel reservation systems increase efficiency and profits?,” SiteMinder, 27-Aug-2018. [Online]. Available: https://www.siteminder.com/r/hotel-distribution/hotel-direct-bookings/hotel-reservation-system-increase-profits/. [Accessed: 03-Apr-2020].

[2] Square, “How to Optimize Salon Scheduling,” Square. [Online]. Available: https://squareup.com/us/en/townsquare/effective-salon-scheduling-strategy. [Accessed: 04-Apr-2020].

[3] “The Code affirms an obligation of computing professionals to use their skills for the benefit of society.,” Code of Ethics. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: 03-Apr-2020].

[4] “IEEE Code of Ethics,” IEEE. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 14-Feb-2020].

[5] BatteryGuy, “Home,” The BatteryGuy.com Knowledge Base. [Online]. Available: https://batteryguy.com/kb/knowledge-base/safety-issues-with-lithium-batteries/. [Accessed: 04-Apr-2020].