SDP'20 – Team 18

[1]

AttendancePlus+

J. Palmer, CSE, J. Thornton, CSE, J. Eisenbies, CSE, and C. Lafountain, CSE

***Abstract*—Teachers waste precious time taking attendance for every class they teach throughout the day. Instead of performing roll calls each and every day (possibly multiple times a day), our system, AttendancePlus+, hopes to change that routine.**

The idea of AttendancePlus+ is to be able to know where any student is in a school building at any given time. The choice of tracking technology to do this is RFID. Two antennas are placed outside of a doorway facing polar oppositely of one another and students are given RFID tags such that when they pass through these doorways, they are immediately accounted for being in the room they walked in. The RFID reader takes in this tag data and is sent to our server where the data is processed and accounted for, giving an immediate update on student's whereabouts or where they are going.

Deliverables of the project include: accurate reading RFID tags (low false positive reads), writing to RFID tags, secure transmission and handling of tag data, full system integration, and an easy-to-use & secure GUI.

I. Introduction

Teachers spend a lot of precious time every day taking attendance for each of their classes. This time, although only a couple minutes per roll call, adds up over the school year and could be better spent teaching. The focus of this system relates to attendance of students, but it can also be used by administrative faculty as well. Since the system keeps track of student's locations within the school, it could also be used by administration to locate a given student in real time. That is, we know the *absolute* location of a student and not just their expected location at a given time.

A. Significance

A fast and effective roll call system gets class started on the right track. Effective roll calls should take only one to two minutes at maximum. Time spent during roll call reduces the learning time for students; therefore, efficient use of time is essential. When too much time is taken to perform a roll call, students become bored and further problems may arise [1].

Let's assume, for example, that a teacher in a middle school has eight different classes throughout their day and that a school year is 180 days. If roll call for every class takes approximately two minutes, then:

- ~16 minutes spent for roll call per day
- ~80 minutes spent for roll call per week
- ~2880 minutes (~48 hours) spent for roll call per academic year

This means that, per academic year, a typical middle school teacher will spend, on average, 48 hours on roll call alone. This is a lot of time that could be better spent teaching.

Now let's assume that it takes roughly thirty seconds from the start of class to take attendance using the idea of an automatic attendance system. That is, a teacher just has to press a button to get an instant roll call of who is in the classroom at that time. Then:

- ~4 minutes spent for roll call per day
- ~20 minutes spent for roll call per week
- ~720 minutes (~12 hours) spent for roll call per academic year

This means that, with an automatic attendance system, the time spent for roll call is four times more time efficient than a manual roll call, giving teachers an extra 36 hours to be teaching per academic year.

B. Context and Existing Products

The idea of an automated attendance system is not new. Outside of the context of a school, automated tracking of assets has been around for many years. Amazon, for example, does this for all of their warehouses and even sells this service to other companies wishing to do the same for their warehouses [3].

In the context of a school building, these systems have also been designed and/or implemented for this case. One example is described in a white paper titled, "*RFID based Attendance System with Applications."* In this application, this system makes use of low frequency (LF, 30 – 500 kHz) near-field RFID using only one antenna near a doorway. The use of low frequency in this system means that for a tag to be read by the RFID reader, the user must place the tag directly in front of the antenna at close range (0 – 3 inches) [2].

The difference of this implementation compared to the AttendancePlus+ implementation lies in the fact that the former uses low frequency RFID technology which requires the tag holder to actively hold their tag in front of the antenna in order for the tag to be read. Our system makes use of ultrahigh frequency (UHF) near-field RFID, that is, tags are able to be read from up to seven meters [7]. Because of the long range, an antenna is able to pick up a tag in close proximity and the tag holder is *not* required to actively hold up their tag to be read. They can simply walk through and be detected.

RFID technology isn't the only type of technology that is able to implement an automated tracking system. Another type of system described in an IEEE conference publication, titled, "*Automatic attendance management system using face*

[1]

detection," uses facial recognition to track the whereabouts of individuals. The idea behind this system uses a fixed camera within a classroom that uses facial recognition technology to determine whether or not a student is inside that room [4].

The largest difference between RFID and facial recognition for automated tracking lies in the fact that facial recognition is much more invasive than RFID. For a facial recognition system, the "tag" that identifies you is your own facial patterns. Facial recognition does have its advantages, such as the impossibility of false negatives during detection, but it also comes with a slew of societal and political concerns, especially when dealing with younger children in a school.

C. Societal Impacts

There are several societal impacts when it comes to systems designed to track and collect data about people. This is accentuated when dealing with systems that track and collect data about children inside of a school.

On the positive side, as discussed previously, any system that tracks student's whereabouts in real time gives the ability for teachers to perform instant roll calls, increasing the time spent teaching and less performing a roll call. It also gives administration the ability to locate a student in real time as opposed to knowing where a student should be at a given time.

There are also certain negative societal impacts that come with this type of system. In an article on Wired titled, "*Tracking School Children With RFID Tags? It's All About the Benjamins,"* criticisms of this type of technology related to the idea of tracking children being similar to tracking cattle on a farm. Other concerns related to the lack of understanding of the technology, especially parents who have little to no technology background, and how this data is being used in the system. For example, some parents were concerned with the long-term health of their child from having high frequency waves around them at all times during the school day [5].

These constituencies played a large part in how we wanted to design our system. We wanted to be able to accurately track students throughout the school while also keeping privacy invasion to a minimum. That is, we wanted private information about a student and their whereabouts to be only displayed on a need-to-know basis and only viewable by authorized individuals. The core goal of our system stayed the same, however the way we process and store private information was heavily influenced by privacy concerns.

D. Requirements Analysis and Specifications

Requirements for our system include: the ability to autonomously detect, identify, and locate students in any given room, ability for administration or faculty to locate a student in real time, private data on students is secured and inaccessible without proper authorization, and non-intrusive, low maintenance integration with existing technologies in the school.

To autonomously detect, identify, and locate a student in any given area of a school, the system must be able to have a significantly high rate of detection and a very low rate of false positives. That is, a high degree of reliability (> 95% accuracy of detection rate). For administration or faculty to be able to locate a student in real time, the design must come with an easy-to-use GUI such that any authorized administrator is able to easily find the location of that student. Providing privacy to private information about students requires that the system provide a crypto suite that dictates how information is gathered, stored, and used in our system. Lastly, to have our system be non-intrusive with low maintenance for integration into the school, our detection system must come with as few parts as possible and easily implemented without the need of extra mounts or stands. *See Table 1 below.*

Requirement	**Specification**	**Value / Solution**
Autonomous detection, identification, and location	High degree of reliability	> 95%
Locate student in real time	GUI for display of private information	Easy-to-use GUI
Protect private information	Secure data gathering	Registration process
	Secure stored data	Non-readable data stored (encryption & hashing)
	Secure display of data	Require authentication from user
Easy integration in school	Minimal number of parts	Two antennas & 1 RFID reader
	Parts can be easily installed	Cables can be hooked onto existing framework

Table 1: Requirements and Specifications

II. Design

A. Overview

The idea behind creating an automated attendance system is creating a system where a student is able to be detected and identified just by walking into a room. If we are able to detect

and identify every individual that walks into a room, then the system will be able to know who and how many people are in that given room. If this can be done accurately and in real time, then the core functionality of the system is complete and other applications related to this can be implemented.

The technology to do this varies depending on the application. For our application, we wanted the technology of our choice to meet to following criteria: (1) long detection range, (2) have the ability to identify the person using the technology (e.g. card or tag stores information to identify you), (3) be cost effective for implementation into a school building, and (4) not require effort by the holder to be detected. For example, technologies we considered to do this were contactless cards, near field communication (NFC), beacons, and RFID. Refer to the Appendix (Section A) for more details on contactless cards, NFC, and beacons.

After doing research into all of these technology possibilities, we found that UHF RFID technology would be the best fit for our application as it had met all four of our criteria. Ultrahigh frequency (or UHF) gives the ability for a tag to be detected from up to several meters away, which did not require effort from the tag holder for their tag to be detected. RFID tags are also cheap and could also store information that could relate back to the identity of the tag holder (e.g. data on the tag contains a unique student ID) [6].

Meeting the specifications listed in Table 1 imposed many tradeoffs to the overall design, as there are multiple solutions to building a system that satisfy those requirements. For example, achieving a high degree of reliability (in regards to our application; being able to pick up any tag, in any orientation, quickly and accurately) gave us two options: 1.) low-power (no battery) tags that don't have a strong signal but easy to wear and cheap, or 2.) high-power (has battery) tags that have a very strong signal and can't be easily hidden but expensive and bulky.

It should also be mentioned that for the "minimal number of parts" and "parts can easily be installed" specifications also played a role in the types of tags we wanted. With option 1.), we have many low-powered tags that don't have a strong return signal. That is, the tag cannot generate enough energy given from the RFID reader's signal to be able to propagate its signal back to the reader. This would mean that we would need to have the design of the physical system be around the doorway of a room or section. With option 2.), we would have few high-powered tags that would be able to be detected from a large range with a high degree of accuracy (that is, it would be difficult to block incoming and returning signals). This would also affect our overall design by only requiring 1 antenna to be mounted in the middle of a room, as we can simply rely on detecting who's inside of the room rather than taking the chance to detect tags coming through the doorway.

Our overall system design contains three main components: sensing system & PCB components, security system, and the GUI. *See the block diagram in Figure 1 below.*

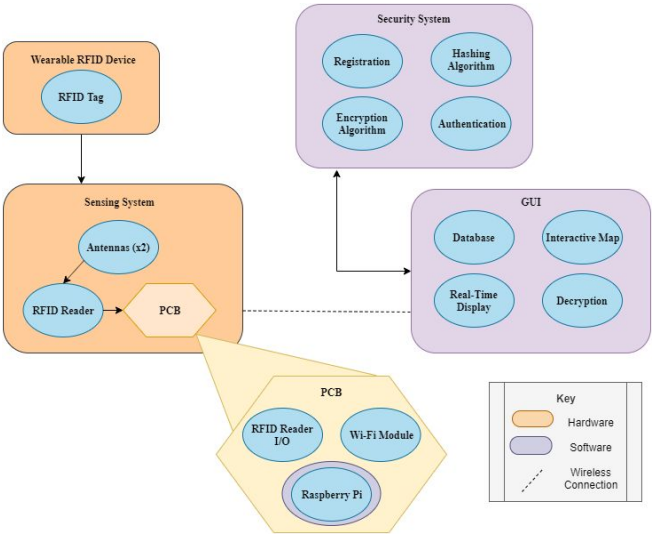

Figure 1: Block Diagram of AttendancePlus

The first significant component (sensing system & PCB sub-components) contains the RFID tag, antennas (x2), an RFID reader, and the PCB that handles reader communication between the Raspberry Pi (RPi) and the RFID reader [7][8][9][10]. Any compatible RFID tag will be detected by the antennas and any information stored on the tag is sent to the RFID reader. The reader then sends this information via TCP to the RPi. Once the data has been sent to the RPi, the RPi then sends this data via WiFi to a local server that hosts our GUI application. Due to the fact that our system uses UHF near-field RFID technology and the small number of components included, this satisfies our requirement for autonomous detection, identification, and location as well as ease of integration into a school. *See section B for more information on these components.*

The second significant component contains the security system of our application. The security system only relates to the server's GUI as the server contains all of the private data that we need to protect. The security system provides a registration system, symmetric encryption / decryption algorithm, an MD5 hashing algorithm, and authentication for registered tags. This system handles the securing of all private data such as how data is processed, how data is stored and secured, and how data will be displayed on the GUI. This provides our overall system with protection on private information. *See section C for more information on the security system.*

The last significant component contains the server's GUI. Because the GUI is hosted on our server, it has access to the database that contains up-to-date information relating to the tags and students including names, tag labels, and locations, all stored as either encryptions or hashes depending on the data type. All relevant data is also displayed in real time on the GUI (the GUI also handles when information should be decrypted, e.g. an authorized user provides credentials to see private information, the GUI communicates with the security system to ask for authorization to decrypt and display private data). Lastly, an interactive map is to be implemented on the GUI as well that will show all students in a given room. This GUI must meet our system's specification that the GUI should be easy to use such that any authorized user is able to find information about any student(s) quickly and accurately. This meets our final requirement for locating students in real time. *See section D for more information on the GUI.*

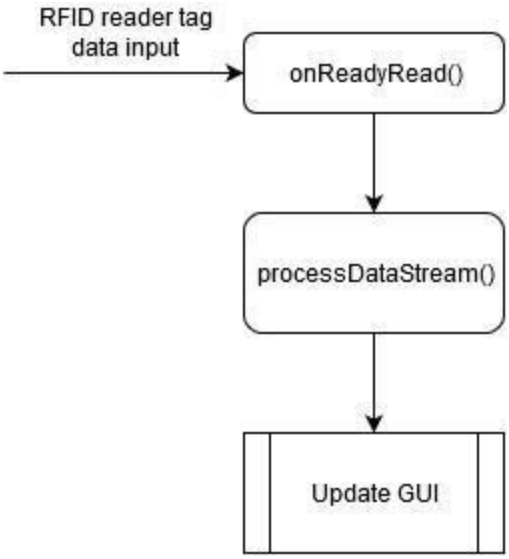

Figure 2: Software Flow Diagram

Before diving into Figure 1 and its components, a high-level understanding of the software flow of this system should also be understood. Figure 2 above shows a high-level view of the software flow for AttendancePlus+. The RFID reader is constantly outputting tag data and sending it via. TCP to the server that the GUI is hosted on. All data sent from the RFID reader comes into the `onReadyRead()` function that does some minor preprocessing to the data based on the format the data was sent (e.g. sometimes data will come in batches of tag data reads). Once the raw data has been formatted correctly, each tag data set is then sent to the `processDataStream()` function that does the main bulk of data processing. This function will break up each individual data piece from the set of data sent, save relevant data, process that data and previous data stored, and then updates the GUI with that processed information. See Section D – Block 3 – GUI for more information.

B. Block 1 – Sensing System & PCB Sub-Components

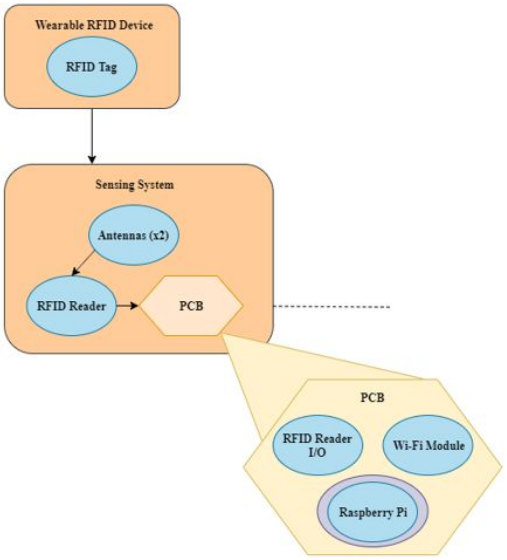

Figure 3: Sensing System & PCB Block

Block 1 handles the reading of tag data and transmission of this data as described in section A above (*also see Figure 3)*. The RFID tag is energized from the signal sent from one of the two antennas and sends back another signal containing the information stored on the tag. This information is then sent from the antenna back to the RFID reader that then sends this tag data to the RPi via a TCP connection set up on the RPi. The RPi then transmits this data via. WiFi to the server's GUI that then processes and stores relevant data in its database.

The technology required for this block are as follows: at least 1 RFID tag, 2 UHF circularly polarized antennas, 1 RFID reader, and a custom PCB that contains a RPi, WiFi module, and TCP I/O connection to communicate to and from the RFID reader [11]. Relevant courses to build this block relate significantly to the Intensive Software Engineering course (I/O of data, serial/TCP connections between a host and server application) and Computer Systems Lab I & II (connection of peripherals to an embedded design, ways embedded systems communicate between other peripherals).

Testing for this block was broken up into four separate parts:

1. RFID reader detection of RFID tags
2. RFID reader TCP communication between reader and RPi to send tag data to
3. RPi WiFi communication between RPi and server's GUI
4. Full communication from RFID reader to server's GUI

First, our goal was to make sure the hardware was set up appropriately such that an active RFID reader is able to pick

up a tag. Next, we wanted the information received by the reader to be sent to the RPi via a TCP connection. Once we could see proper data transmission from the reader to the RPi, we then needed to get communication set up from the RPi to the server's GUI via a WiFi connection. Once each individual line of communication was operating appropriately, we then tested how well the full integration of each part sending data from the initial read of the tag down to how the data appeared coming into our server. Refer to Appendix (Section B – Block 1 Testing) for more information on the experiments.

C. Block 2 – Security System

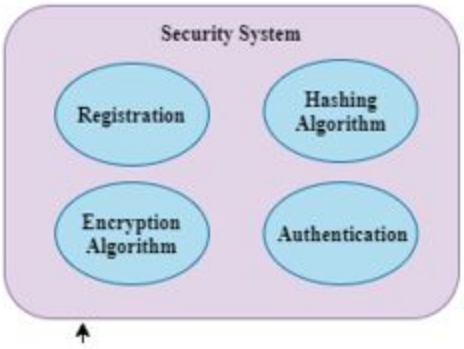

Figure 4: Security System

Block 2 handles the security implemented on the server's GUI, *see Figure 4 above.* Our security system implements a registration system that allows us to filter and verify registered tags within a school. Registration is set up such that an authorized user types in the name of the student and label of the tag being assigned to the student. Once the tag has been registered, an encryption of the tag holder's name and a hash of the tag label is stored into the server's database. The hash of the tag label is used for authentication of the tag and the tag holder. When data is read in, the tag label is also part of that data. When we take a hash of this tag label, we compare it to the other hashes of registered tags stored in our database. If a match is found, then we know that this tag has been registered and further processing for the rest of the data is continued.

The technology required for this block is strictly software based that relates directly to our Trustworthy Computing course. Ideas such as symmetric encryption, hashing, when and how to use these security services were required to be able to build this block (which was adopted and adapted from this course).

Testing for this block was also broken up into two separate parts:

1. Ensure registered tag information is being stored appropriately in our database (e.g. only registered tags should appear on the GUI when being read in)
2. Ensure encryption and decryption of the tag holder's names are displayed appropriately given the users level of authorization

Our first goal in testing with this block was to make sure that the registration system was performing as we expected it to. That is, we wanted to make sure that only the encrypted name and hashed tag label was being stored in our database correctly and no human-readable private information was being stored. The second goal of this block was to ensure that the decryption of the tag holder's name was appropriately executed if the user of the GUI provided accurate credentials into the system. Refer to Appendix (Section B – Block 2 Testing) for more information on the experiments.

D. Block 3 - GUI

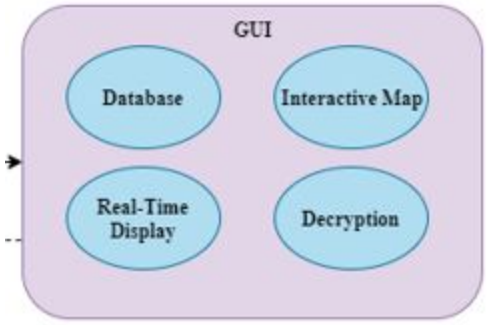

Figure 5: GUI

The final block of our system is the GUI which handles data processing, database management, real-time display of tag holder information (in both list form and on an interactive map), and request for decryption of tag holder's names. *See Figure 1 and Figure 5 above.*

Like our security system block, the technology required to build this block is strictly software based. Relevant courses required to design this block were Data Structures & Algorithms (database & data storage), Trustworthy Computing (decryption techniques), and programming experience from all other programming-related courses (designing and building the GUI).

Testing done for this block was very similar to any type of testing when building a GUI. Key parts of the GUI that we tested thoroughly were:

1. Parsing and formatting tag data correctly
2. Tag data displayed correctly and in real time given the level of authorization of the user
3. Decryption of tag holder's name displayed appropriately
4. Database was kept up-to-date and refreshed if GUI was restarted

Our first goal of designing the GUI was to ensure that the server parsed and formatted tag data correctly. Tag data from

the reader to the server came in the example form below:

01740AD\t12:34:44.922 PM \t-70\t10\t1

Where each field is separated by "\t". The fields follow this order: tag label, timestamp (when the tag was read), RSSI value, read count of that tag during the time it was read, and the antenna number the tag was read by. The goal here was parsing the data as well as creating a "TagData" object type that stored this information inside of it which would then be stored into our database.

Our second goal and third goal were making sure that the information about the read-in tags were displaying appropriately in real time based on the level of authorization of the user. That is, if a user did not provide credentials that would allow them to see the private information of the students, then what would be displayed on the GUI would be the encryption of the name of the tag holder and the word "RESTRICTED" for the location of the tag. When appropriate credentials were provided to the GUI, then the user would be able to see the decryption of the tag holder's name as well as the actual location of the tag.

Lastly, we wanted to make sure that upon restart of the GUI, the registry of tags would stay up-to-date. This was done by writing the encryptions of the tag holder's names and the un-hashed tag labels of registered tags into a text file. Upon startup of the GUI, the first action the GUI performs is reinitializing the registered tags into the database. Refer to Appendix (Section B – Block 3 Testing) for more information on the experiments.

III. The Product

A. *Overview*

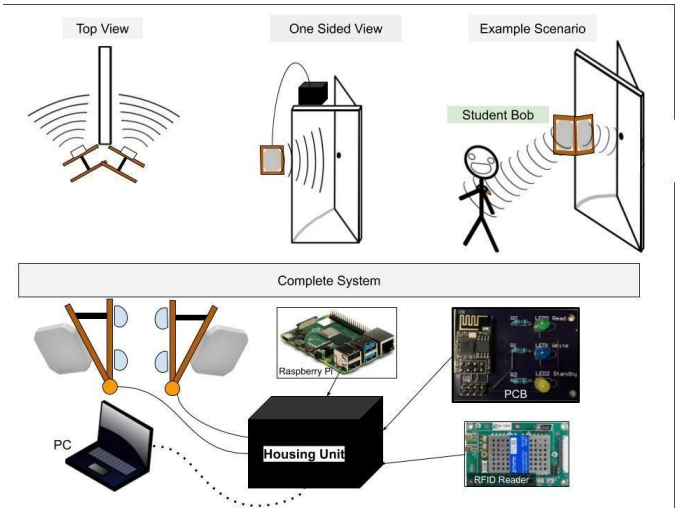

Figure 6: AttendancePlus+ Product Sketch

The overall product (shown in Figure 6 above) resulted in a two-antenna system placed at the doorway or entrance of a given room. The antennas are positioned in the middle of the doorway. The antennas connected to a black box containing the Raspberry Pi, the WiFi module, and the RFID reader. Any PC remotely connected to the Raspberry Pi (hence the dotted line from the PC to the Housing Unit in Figure 6 / dotted line from the Sensing System to the GUI in the block diagram in Figure 1) then had access to the database where their distributed application (the GUI) would then run and gather this data. Note that the Sensing System in Figure 1 is equivalent to the antenna and housing unit (block box) in Figure 6.

B. *Electronic Hardware Component*

Our printed circuit board had two main functionalities. The first was to provide a WiFi communication system that connects the Raspberry Pi, which collects the data from the RFID reader, to the server that provides the GUI. The PCB traces are laid out from each pin on the ESP8266 WiFi module to header pins that the raspberry pi can connect to. The other function that the PCB provides are LED indications lights. There is one for when the RFID reader is in standby, reading, and writing mode which allow the user to see exactly what the hardware is doing. The PCB was designed using Altium and fabricated by the company OSH Park. Each component was then hand soldered onto the board. For testing, before soldering the first thing to do was to check for continuity with each trace. This was done by using a multimeter. Then after having everything soldered was to make sure that the WiFi module and LEDs were functioning with the Raspberry Pi properly. This was done by starting the RFID Reader up and having it in each mode - standby, read, and write - to see if the corresponding LEDs would light up. We then also tested by sending data through the WiFi module to the server.

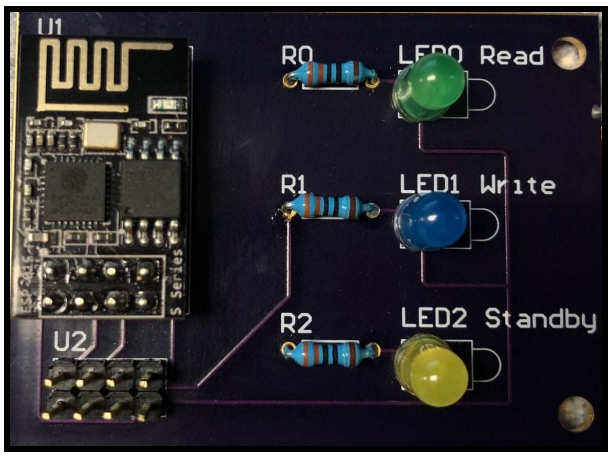

Figure 7: Image of Original Printed Circuit Board

C. *Functionality*

In reference to the block diagram in Figure 1, by the time of CDR, every element in the diagram was functioning and working appropriately. There had been a lot of testing done previously when the team was working on integrating the custom PCB into the design. Testing the integrated PCB also required us to essentially be testing the entire system, as the system could not function appropriately without the WiFi module on the PCB working.

This led to the team completing testing of the entire system as we were testing the integration of the custom PCB. The only part of the system that required significant troubleshooting was the actual orientation of the system. That is, testing how the antennas should be placed on the doorway such that we maximize the rate at which we successfully pick up a tag.

In regards to CDR specifically, our system functioned appropriately and completely given optimal antenna and tag placement. The work done after CDR was towards increasing reliability given suboptimal tag orientations (which we found to be the middle of the doorways).

D. *Performance*

AttendancePlus+ has met each of the goals outlined in the requirement section of this report, along with others. Reliability when used with an RFID tag on an individual's back is near perfect, meeting the goal of greater than ninety-five percent.

Trial Number:	1	2	3	4	5	6	7	8	9	10	Success Rate:
Two People: Hanging From Wrist	☑	☑	.5	☑	.5	☑	☑	☑	☑	☑	90.9%
Two People: Necklace Chest Level	.5	.5	☑	☑	X	☑	☑	.5	☑	☑	75%
Two People: Necklace on Back at Chest Height	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	100%

Figure 8: Analysis of Product Success

The above figure shows a round of testing that analyzes different use cases of the product in order to discover the optimal usage and orientation of the tgs and antenna. Our testing shows perfect use in ten cases in the final design orientation, which makes us quite confident in our design.

The final GUI design is user-friendly and does not require extensive knowledge of the underlying engineering, as can be seen in the figure.

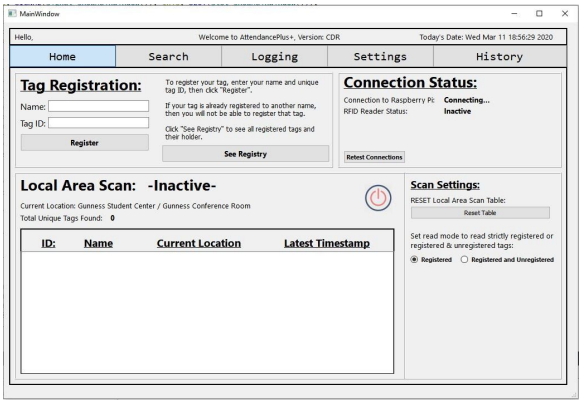

Figure 9: User Interface of Attendance Plus+

It also implements many features allowing for easy access to information by authorized individuals. A combination of registration, encryption, hashing, and user authentication ensure that data is secure and only accessible by authorized individuals. All of these components are easily visible and accessible on the GUI with different labeled sections and a clear view of real-time tag activity.

The system uses only two antennas and an RFID reader, which makes the design straightforward for users. It is also simple to install as it only requires the connection of the antenna cables and attachment of a mount. The mount uses military-grade suction cups that allows it to attach to multiple surfaces.

IV. Conclusion

AttendancePlus+ has created an automated attendance system that is a viable alternative to the current manual attendance-taking in schools. It is easy to see the documented benefits to our system, such as time savings for teachers.

The system has met the goals we set for it at the beginning of the project in terms of reliability, security, and ease of use. Our RFID system functions with near perfect accuracy and with the added benefit of an easy to use interface. The remaining time before FPR would have been used to further test the system and improve its appearance, though much of the functionality is finished.

Acknowledgment

We would like to thank Prof. Jeremy Gummeson for all of his valuable help in understanding the capabilities of RFID technology and helping us choose the appropriate parts to purchase for our system.

We also would like to thank Francis Caron for all of his hard work in making sure we had everything we needed to be successful with our project.

Finally, we would like to thank our advisor Prof. David

Irwin, who helped us see the big picture of our project and understand the importance of prioritization of ideas.

References

[1] Wisconsin Department of Public Instruction, *1-Active PE Minutes,* 2013, pp. 1 – 3.

[2] ElProCus - Electronic Projects for Engineering Students. (2019). *RFID Based Attendance System with Related Applications*. [online] Available at: https://www.elprocus.com/rfid-based-attendance-management-system/ [Accessed 16 Dec. 2019].

[3] T. S., "Amazon Inventory Management Software Integration," *Finale Inventory*, 27-May-2018. [Online]. Available: https://www.finaleinventory.com/amazon-inventory-management. [Accessed: 18-Dec-2019].

[4] Ieeexplore.ieee.org. (2019). *Automatic attendance management system using face detection - IEEE Conference Publication*. [online] Available at: https://ieeexplore.ieee.org/document/7916753 [Accessed 16 Dec. 2019].

[5] D. Kravets, "Tracking School Children With RFID Tags? It's All About the Benjamins," *Wired*, 03-Jun-2017. [Online]. Available: https://www.wired.com/2012/09/rfid-chip-student-monitoring/. [Accessed: 18-Dec-2019].

[6] T. I. T. Guy and T. I. T. Guy, "RFID, NFC, Beacons, Sensors & Other 'Things,'" *The IoT Guy - A Pragmatic View of the Enterprise Internet of Things*, 26-Nov-2014. [Online]. Available: http://theiotguy.com/rfid-nfc-beacons-sensors-things/. [Accessed: 18-Dec-2019].

[7] JADAK LLC, "Datasheet for ThingMagic Micro carrier board." Available: rfid.atlasrfidstore.com/hubfs/1_Tech_Spec_Sheets/Thingmagic/ATLAS%20ThingMagic%20Micro%20Embedded%20RFID%20Reader%20Module%20JADAK.pdf [Accessed: 25-Jan-2020].

[8] VULCAN RFID, "RFID tags." Available: rfid.atlasrfidstore.com/hubfs/Tech_Spec_Sheets/Vulcan%20RFID/ATLAS%20Vulcan%20RFID%20Custom%20Key%20Fob%20Tag%20Version%202.pdf [Accessed: 25-Jan-2020].

[9] Laird Technologies, "Laird UHF RFID Antenna." Available: rfid.atlasrfidstore.com/hs-fs/hub/300870/file-1480469978-pdf/Tech_Spec_Sheets/Laird/ATLAS_Laird_S9025PLNF.pdf [Accessed: 25-Jan-2020].

[10] Raspberry Pi, Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. [Accessed: 25-Jan-2020].

[11] Sparkfun, "WiFi Module - ESP8266." Available: https://www.sparkfun.com/products/13678. [Accessed: 25-Jan-2020].

Appendix

A. Design Alternatives

The design alternatives we considered for our system were contactless cards, near field communication (NFC), and beacons. The criteria that needed to be met for our technology of choice was that: (1) it had to be somewhat long range, (2) have the ability to identify the person using the technology (e.g. card or tag stores information to identify you), (3) be cost effective for implementation into a school building, and (4) not require effort by the holder to be detected. If all four criteria are met, then the goal of our system could be reached.

Contactless cards and NFC provide similar use as you would for Apple Wallet, however it came with a lot of drawbacks that didn't fit the system's criteria. Although they are "contactless" devices, they are not long range and required the user to hold up the card near a reading device, failing to meet criteria (1) and (4) [6].

Beacons are low-energy Bluetooth devices that operate similar to RFID in the sense that they broadcast a unique identifier to a specific local area. Beacons seemed to be the perfect fit for our system as they met criteria (1), (2), and (4), but each beacon were very expensive, failing to meet criteria (3) [6].

B. Technical Standards

Some examples of standardized hardware and software that were used in our project include the following: WiFi, with the IEEE standard 802.11, which we with the ESP8266 WiFi module to communicate from the raspberry pi to our server, RFID with the IEEE standard 802.15, which we used passive far field RFID to read and write to tags. Also we had the operating frequency of the RFID reader stay within 902 - 928MHz abiding to the FCC regulation for RFID UHF in North America.

C. Testing Methods

Block 1 Testing:

Experiments that have been designed and executed so far are the correct reading of tags by the RFID reader, TCP communication between the reader and the RPi, and RPi WiFi communication between the RPi and the server's GUI. Full communication from the reader to the server's GUI has been in testing and not yet fully integrated yet. Further testing will be done to ensure this part is operating appropriately.

The purpose of these tests is to ensure that dataflow from the reader to the server's GUI is transmitted correctly and accurately so that the correct information is being received by the server in real time. The methods to ensure appropriate testing were straightforward for this block, that is, we just needed to confirm that the same data the reader receives is exactly the same data being transmitted between each connection.

Block 2 Testing:

All experiments / testing done with this block were executed as described in part II, section C above. The purpose for each of these tests were to ensure that our security system

operated and behaved as we expected to when reading in and displaying data. If these tests were successful, then we could say that it fulfilled our requirement on protecting private data.

For the first part of our testing (ensuring registered tag information was being stored appropriately), all we needed to look for was to see that the encryption and hash displayed on the GUI was also being stored into our database, which is a hash map. By simply printing out the contents of the database, we could see both the encryption of the tag holder's name and the hash of the tag label were stored appropriately for each registered tag.

Lastly, for this test, we made sure that when a registered tag was read in, that not only was that tag displaying appropriately onto the GUI, but also that when the name of the tag was decrypted, it was the same name registered under that tag label. When we tried to read in a tag that was not registered, we checked to make sure that the security system rejected the tag data after confirming that its tag label hash was not found in the database.

For our second and last test for this block, we wanted to ensure that encryption and decryption of the tag holder's name was done appropriately. That is, we checked to see if the decryption of the encrypted name always decrypted correctly for the given tag that it was registered to. This was simply done by observing the encryption and known name of the tag and seeing that the decrypted name appropriately related to the tag label's hash.

Block 3 Testing:

All experiments / testing done with this block were executed as described in part II, section D above. The purposes of each of these tests were to ensure correct functionality of the GUI (including appropriate displaying of information, storing and reinitialization of the registry upon restart).

For the first part of testing for this block, we wanted to ensure that we were parsing and formatting data correctly. The example form in which data entered our GUI is as below:

01740AD\t12:34:44.922 PM \t-70\t10\t1

Where each field is separated by "\t". Each field is broken up and stored into a "TagData" object. For testing, we simply had getters and setters for each of the fields of the TagData object and printed them out after a tag was read in to ensure that each field was being saved appropriately for each different TagData object. On top of that, we also made sure that information in each TagData object was being updated appropriately (e.g. location, timestamp). We confirmed that this was working correctly just by seeing how the data being displayed changed as location and timestamps were being updated (this was due in part to the fact that information displayed on the GUI pulled the data straight from the database itself , confirming that the database was updating data correctly).

For the second and third part of testing for this block, we wanted to ensure that not only was the GUI updating appropriately in real time (as covered in our first experiment), but also that the correct data was being shown based on the level of authorization of the user. That is, without appropriate credentials to view private information, the user should only see the encryption of the tag holder's name under the "Name:" field and the word "RESTRICTED" under the "Location:" field in the GUI. Testing for this was simple; observe how information is being displayed before and after authorization has been provided to the users.

For the fourth and last part of testing for this block, we wanted to ensure that the registry was kept up-to-date at all times, especially upon restart of the GUI. Because the database is created only when the GUI is running, we needed to ensure that all registered tags during previous sessions were still available in our database. This was done by writing to a text file the encryptions of all registered tag holder's names as well as all of the un-hashed tag labels during registration. This file, upon startup of the GUI, was read back in and reinitialized the registry as well as all TagData objects stored back into the database.

Testing Summary:

All testing described above have been successful expect for the testing for full data communication from the reader to the server's GUI as described in the Block 1 Testing section. Further testing for that part will be done during the Spring semester.

Conclusions drawn from all experiments of our system support that our requirements for our system are being met and that our system is fully functional aside from minor tweaks that need to be made.

D. Team Organization

Team 18 is organized as follows:

J. Palmer – Team manager, GUI integrator,

C. Lafountain – PCB lead, RFID reader application developer

J. Thornton – security & data transmission engineer, mount design

J. Eisenbies – lead data transmission engineer.

We believe that the team has been working very well with each other. The team highly values open communication as

well as setting intermediate deadlines for parts of the project to be completed. From these intermediate deadlines, we have been able to make continuous progress week after week by staying on track and knowing what needs to get done by when. When problems are arising in meeting deadlines, team members communicate with each other on what they're struggling on and help is offered by other team members when they can.

An example of leadership displayed by our team was when J. Palmer, the team manager, was unable to provide updates to our advisor, Prof. Irwin, due to the fact that he was out of town and traveling. C. Lafountain took initiative to take on that role while he was away and did so very well.

An example of when team members helped each other out was during the process of getting the reader to communicate with the RPi and the RPi to communicate with the server's GUI. J. Eisenbies and J. Thornton let the team know that they were struggling with this implementation and C. Lafountain drove from home in Easthampton to assist with the problem.

An example of when communication within the team broke down, which happens often due to the rigor of other coursework and job searching, was always remedied by calling for a meeting every once a week to talk about the project, any upcoming impedances (i.e. exams, other coursework due), and adapted ourselves to other's schedules to ensure a continuous flow of work being done to the project.

E. Beyond the Classroom

J. Palmer:

The skills that I have needed to develop has definitely been time management for not only myself, but also for others as well. Making sure the team was on track with deadlines and project completion while also handling managerial tasks, other coursework, and two jobs was very stressful for me during this semester. But I was able to make tradeoffs between needs and wants to be able to be successful this semester.

The faculty at UMass have been exceptionally helpful to me during this project. Especially from M5 when I was trying to figure out how a Raspberry Pi worked.

The work I have done so far, not only for this project but from the rest of my time as a student here in college, I have seen a lot of overlap between what I'm doing and what will be expected of me when I graduate and begin my career. I've also realized how valuable teamwork is to companies and not just my ability to perform my technical job.

J. Thornton:

In this project I have needed to use a majority of network connectivity skills, i.e. client-server capabilities, navigating firewalls.

These skills have helped me to troubleshoot Wi-Fi and client components, something that I see being useful in my post-grad job as I work to connect back and front-end parts of major software. I have also become more versed in front end design, which will directly help my future position where I will continue to be UI and frontend using languages like CSS, HTML, and QT.

J. Eisenbies:

I have made use of C, Java, and Python in attempting to communicate with the WiFi module. It also required some basic circuitry in connecting the necessary pins.

Data sheets for the different components helped some but most assistance came from online forums for programming advice.

While the project itself isn't really related to anything in my professional experience, the process has major similarities to projects in other areas. The technique of developing different parts of the project independently allows different parts to progress and makes it easier to keep track of your section of the project and then allowing them to be combined. Furthermore, the process of building a section slowly from basic functionality to completeness mimics the process I used during internships to solve a single problem at a time.

C. Lafountain:

Skills I've needed for success in this project include adapting to learn a new programming language and reading datasheets for embedded systems.

Professor Gummeson has been a great resource for learning about RFID and using the Mercury API has helped in implementing the code for the reader.

I do see this helping out if I choose to work with embedded systems and also working in an engineering team in general.