Personal Wardrobe Assistant (PWA)

O'Neal Thomas, Maximilian Carroll, Juan Herrera, Efren Cintron

College of Engineering and Computer Science, University of Central Florida, Orlando, FL, 32826

***Abstract*— This paper presents the design process utilized to realize electronic circuitry, radio frequency, microcontroller, and web application technology used to create a system that will organize and maintain a household wardrobe. The design process involved the selection and testing of several electronic components which will be covered in this paper.**

I. Introduction

In the modern world the idea of automation has spread far and wide to virtually every aspect of a person's life. The process of automation has been applied to many different ideas to great effect, Like Henry Ford with the car manufacturing process or the McDonalds brothers with fast food. Some people claim it to be an inevitability that machines will soon do every task that the average person finds too trivial. What about the most basic of human activities though? Simple menial tasks that waste a person's time, that could be used more productively. There are many individuals, across the planet, that find organization to be mandatory. It obviously allows them to be more efficient and save time. Like everything else a person does it is prone to human error or interruptions from outside sources. The idea of Automated organization is very simple in concept but what would be the best way to test this theory.

There are hundreds of people who want to keep orderly. For demonstration purposes, what better way to test this idea than on clothing. Keeping track of clothes becomes more cumbersome the more one thinks about it. A person has lost their favorite shirt even though they are certain it was in their closet, but it cannot be found. Ensuring that people remember to wash the clothes they need for a specific situation or realizing that this brand of attire has not been worn in months. So why not donate it or dispose of it?

The Personal Wardrobe Assistant is a device coupled with a web application & database that will keep track of a person's wardrobe. Gone are the days of missing a shirt that you thought was in your closet. Get updates on your favorite colors to wear and when it might be time to get rid of some of the items you don't wear.

A. Distribution of Workload

Within this project the workload is distributed based on each individual's strengths and interests. Efren and Maximilian being electrical engineers were put in charge over the more strictly electrical and hardware based tasks such as delivering the proper power to the circuit and tuning the antenna for the specific frequency. Juan and O'Neal being computer engineers were put in charge over the more software based tasks such as sending and receiving data from the reader and the web application.

B. Flow of the system

In order to keep track of clothing articles we will be utilizing RFID technology in order to create a digital signature for each item. Each item's RFID information will be stored in a remote database and can be interfaced through a web application.

C. Block diagram

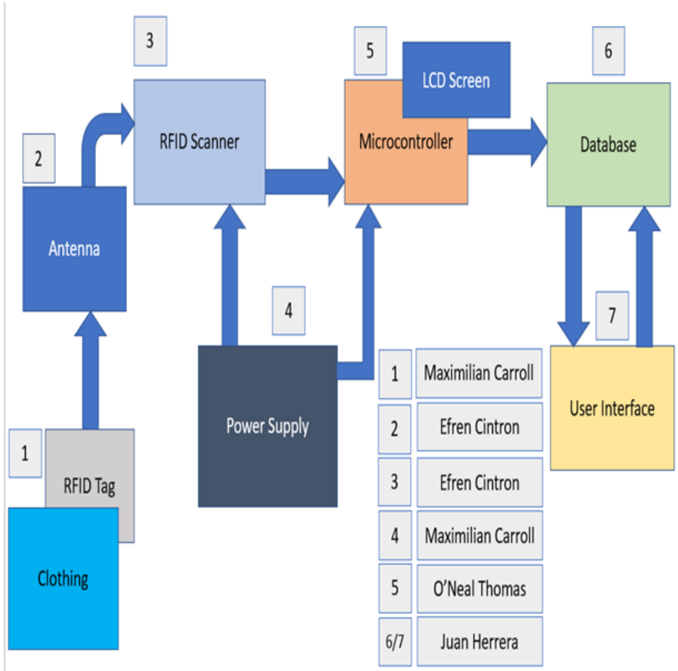

Figure 1: Block Diagram

II. Design Requirements

A. RFID Antenna / Reader

The RFID antenna is one of the most important components of our project. This antenna takes the RFID information from the tag and sends it to the RFID Reader so it can read/add the type of clothing the user is going to wear. The RFID tags that we are using are Low frequency, the antenna is going to be suitable to scan at a frequency of 125KHz.

For the RFID Antenna, it requires a voltage of no more than 5V peak-to-peak in order to work properly. It also needs a range from 5 to 10cm to be able to scan in a relative short distance. This antenna was built by hand and required to be precise in order to get those values mentioned above.

The ID-3LA RFID reader encodes the information of the RFID tag that has been scanned by the antenna to send it to the microcontroller. The specifications of this reader have a limited range for an antenna up to 30cm. For our design purposes, we kept a range no more than 10cm which we encountered to be better for user experience.

Figure 2: ID-3LA RFID Reader

RFID Reader	ID-3LA
Frequency	125KHz
Card Format	EM4001 or compatible
Read Range ID3	Up to 30cm
Encoding	Manchester 64-bit, modulus 64
Power Requirement	+2.8 VDC thru +5 VDC
RF I/O Output Current	+/- 200mA Peak to peak
Inductance	1.337mH

Table 1 []

B. *Power*

The power supply circuit is a very basic design that is the backbone of the project. A DC to DC conversion circuit that ensures all components of the project receive the required voltages.

Component	Required input voltage
ID-3LA reader	2.8V-5V
LGDehome IIC/I2C/TWI LCD	5V
Microcontroller	3.3V
mxuteukBlack Electronic Buzzer	min: 800-900mV

Table 2

The power supply circuit consists of 4 components, not including the DC voltage sources. A 1N4001 diode, two electrolytic capacitors, with values of 1000uF and 10uF. Finally it will also include a L7805 voltage regulator.

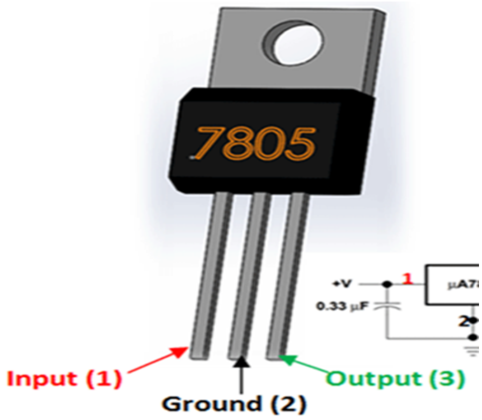

Figure 3: L7805 Voltage regulator

When all these components are connected together and transform the 12V input into a 5V output. The voltage regulator is responsible for stepping down the voltage so that the electronics in the circuits receive the specified power. The capacitors are used to smooth the DC signal so that the output voltage will be a more smooth and efficient wave form. This helps ensure that the electronic components work more efficiently. Finally the diode is used as a fail safe. Specifically its job is over voltage protection by ensuring that current only flows in one direction and if a spike is detected it sends the potentially damaging voltage to ground ensuring that it will not damage any of the electronic components.

C. Interacting with the database

The RFID reader is designed to send bytes over a serial bus in the format of UART (Universal Asynchronous Receiver-Transmitter) ; these bytes need to be processed and delivered to the database which will be remote. This will allow the device to be portable. For these requirements to be met we will need a component that will be able to communicate using UART and can also establish a WiFi connection to allow for the transfer of the information to the database. In order to communicate with the database via WiFi the component will have to be able to send POST requests. This will be the most optimal way to communicate with the database. After listing these requirements for interacting with the database and RFID data. In the next part we discuss selecting the most optimal component to get this job done.

III. Parts selection

A. *Selecting a Power Source*

The system has two DC power sources. One twelve volt battery pack that consists of eight double AA batteries. This battery pack is responsible for powering the LCd screen, buzzer and RFID reader. A separate lithium ion battery with a voltage of 3.7 volt and a current of 500 mAh is independently responsible for powering the microcontroller for the final project. The microcontroller is the brains of the operation and used to interpret the RFID after it is scanned by the antenna. The main reason for this was a matter of reliability and troubleshooting.

All the other components of the project are powered by the 12V battery pack but have little to no software requirements. To ensure that the microcontroller could be worked on separately it made sense to use an independent power source for it. The battery pack itself also contains an on/off switch. This will be directly exposed by the final case and will be directly accessible to the user. This is simply to ensure that the DC batteries are not being rained continuously and that the project will be able to operate for longer periods of time without having to replace the batteries in the pack.

B. Antenna

As mentioned in the design requirements section, this antenna was built by hand. In order to build it, it requires the following: PVC tubes and enameled copper wires. For the trial of this project, I was able to build a total of 9 antennas. These antennas have different configurations and measurements that were tested to get the best range possible.

Figure 4: Nine Handmade antennas

To end up with 9 different antennas, we consider to use 2, 3 and 4 inch diameter rounded PVC tubes combined with 28, 30 and 32 AWG enameled copper wire cable. To configure the number of loops needed to the antenna to get the best range possible, we used an online calculator[2] that approximates the number of loops to wind on the PVC tube. The input needed to get the number of loops are the inductance value of 1.337 mH, the outside diameter of the PVC tube in mm, and the thickness of the enamel copper wire.

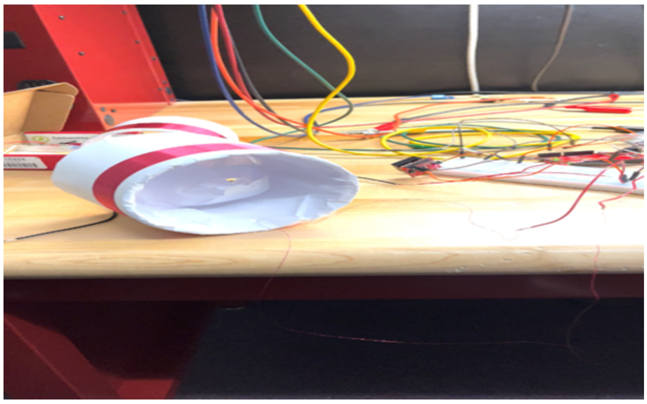

Figure 5: Testing the antenna and the RFID reader

When testing the antennas with the ID-3LA RFID Reader, connected to the oscilloscope we get a sine wave output. This means that the antenna is going to be able to scan any RFID tag compatible with a low-frequency tag. Each antenna was tested and improved their scanning range. The best way to know if the antenna needs to add/subtract loops, the output sine wave will either expand or compress.

When we talk about expanding/compressing the waveform is when you get the RFID tag closer to the antenna and the waveform gets either larger or smaller. When the RFID tag gets closer to the antenna and the waveform expands it means that it doesn't have enough inductance and needs to add more cable to the coil. On the other hand, if the waveform compresses it means that it has too much inductance and the copper cable needs to unwind some cable from the coil.

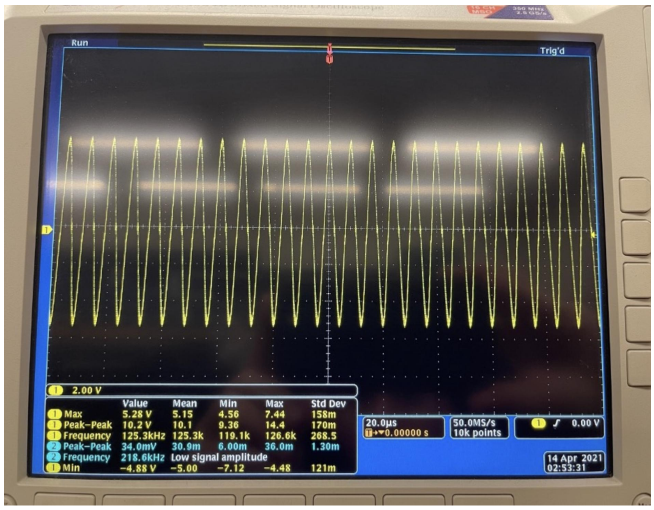

Figure 6: Testing the antennas on the oscilloscope

The results obtained with the antennas were shocking. 7 out of the 9 antennas were scanning the RFID tags. The two antennas that were not working are the ones that the waveform expands, meaning that they need to wind more copper wire. Based on the results, the antennas that get the highest range are the 2 inch and the 3 inch from the 30 AWG copper wire.Now we are going to run a second trial test making sure that this time, all the antennas can read

RFID tags at their optimum state. For the two antennas that were not working, it will have a total of 120 turns on the coil, an increase of 20% to start tuning them up and make them scan RFID tags.

Based on the results obtained, we can determine that all the antennas can scan tags at this point.We selected the 30 AWG of 2 inch out of the other 8 antennas. The 30 AWG of 2 inch gives us a range of 8cm. The smaller the coil, the better to increase the field strength and coupling.

C. Selecting a "stack" for web app

The MERN stack was ultimately the choice that we picked for the application.

1. MongoDB - document database
2. Express(.js) - Node.js web framework
3. React(.js) - a client-side JavaScript framework
4. Node(.js) - the premier JavaScript web server

The choice came down to the differences between React(JXS), Angular, and Vue. Angular being the most "complete" out of the three comes with this the condition of learning TypeScript. React and Vue were really the only option, both are very powerful common languages that are similar. We ended up choosing React mainly for the numerous amounts of documentation and assistance you can find online. NoSQL database was chosen due to its flexibility and dynamic properties. NoSQL allows you to make quick changes to the database without the hassle of a SQL database. Additionally, I am very familiar with Mongoose and it was a no-brainer going with MongoDB. Counter arguments are that the system is not easily scalable. It is a valid point but we do not see having a large number of users where scalability is a concern. Node and Express were easy choices since I have experience with back-end development and not only are those two great if not the best at what they do but I also have experience programming with them. Express does so much work behind the scenes that it would not be advisable not to use it.

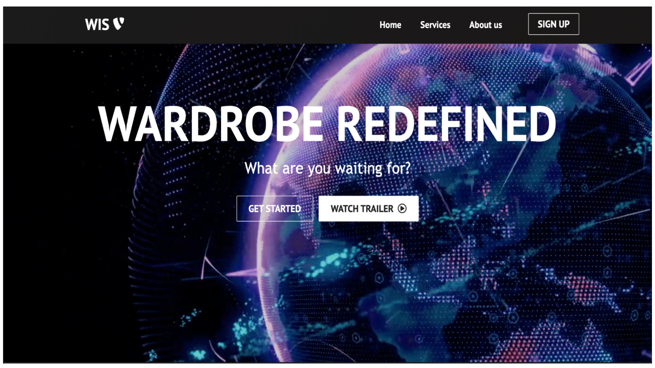

Figure 7: Main page of web application

The front end is mainly composed of React, React Hooks, and Bootstrap. The front end is interacting with our API using axios. Axios allows us to send request and interact with the server/back-end.

D. Microcontroller / LCD / Buzzer

After doing research online and comparing similar systems. We've concluded that the most flexible and cost effective way to communicate with the database while simultaneously communicating with the rfid reader is to use a microcontroller unit. Microcontroller units come with many of the individual pieces that we could not program or manipulate on our own. The best microcontroller options for us were the Arduino UNO, MSP430, Raspberry Pi, or Adafruit feather. Each microcontroller has its own strengths and weaknesses but only one reigned supreme for our project application. The Adafruit feather Huzzah was chosen to be the microcontroller for our project for two important reasons. The first reason being that the Adafruit feather Huzzah is exponentially less pricey than its opponents in this selection. The average price for one of these on average is 15 to 20 dollars.

Arduino UNO	20-25
Raspberry Pi	60-70$
MSP430	30-40$
Adafruit feather	15-20$

Table 3 MCU price comparisons

This price range was the most suitable for our budget. Secondly the Adafruit Feather Huzzah comes equipped with an ESP8266 WiFi module. This reason was the main selling point for the Feather. With the WiFi Unit being integrated onto the board, this allowed our project to have WiFi accessibility without having to buy an extra component. We are able to program the ESP8266 WiFi module directly from the board.

We also decided to integrate an LCD screen and a buzzer into our system. Having an LCD screen and a buzzer will give visual and auditory responses for the user. For selecting an LCD this decision was quite simple. We chose a 16x2 LCD screen that allowed us to use I2C in order to communicate with the MCU. A 16x2 LCD screen provided our project with enough room to write prompts but not too big that it would be a hindrance to our system. As for the buzzer we decided to go with an active buzzer instead of a passive one. Active buzzers are very clean and easy to use. All we simply need to do is send a voltage to the positive lead and tie the negative lead to ground. We were also able to use the LCD screen and buzzer as troubleshooting mechanisms for our system to tell if the connections were properly made.

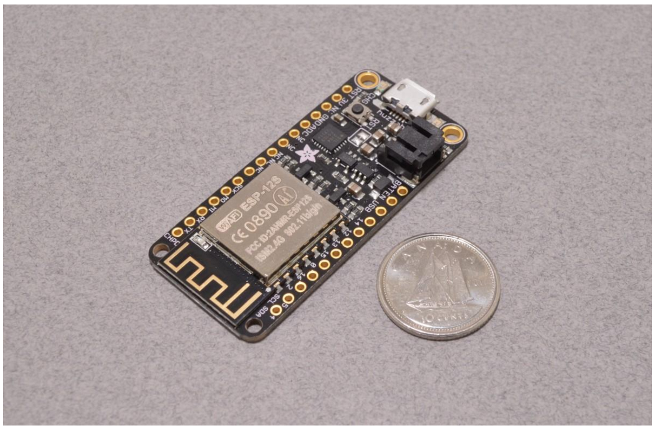

Figure 8: Adafruit Feather Huzzah size compared to a coin

IV. PCB Board design / case design

A. Putting it all together

Once all of the components were ordered and tested individually the next step is to create a printed circuit board in order to place each component on to interact with each other. We used Eagle to create our PCB prototype. Our group is relatively brand new to PCB board design, so our first iteration of the PCB was a complete disaster. We hit the drawing board and redesigned the PCB. Upon receiving the second iteration of the PCB although we made some important changes, we still were not able to even test our PCB due to sizing issues. So, we went back to the drawing board once again for our third iteration. After making the necessary sizing changes we ordered the PCB once again. With this third iteration we were actually able to run tests on our system and in the next section we will explain our testing process.

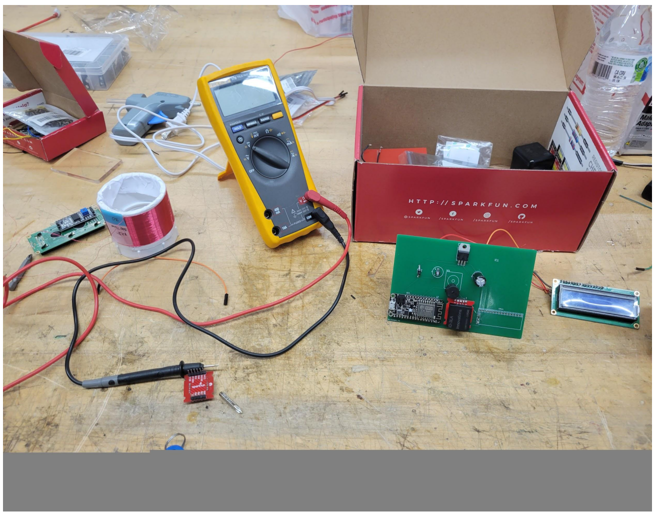

Figure 9: Testing the PCB board with the antenna and LCD Display

B. Testing the system

Although our system is fairly simple, testing was a very strenuous process. We ran into several issues and some of them took seconds to solve and others have taken the duration of our semester.

1. Power: Testing the power components was straightforward. using a breadboard all the components were brought put together and then tested before confirming them for the final design. Using common tools like a multimeter were then used to test individual outputs.

2. Antenna / Reader : Testing the RFID reader didn't have any complications. Just need to connect the antennas to the pin of the RFID reader to make it work. As for the antenna we had some issues regarding the read range which was adjusted to meet the range requirements so it could be user friendly.

3. MCU Programming: The process of programming the microcontroller was very simple to set up. The MCU was able to use Arduino IDE which made programming also simple because we already understand C. So the MCU's job is broken down into 3 basic steps. 1.) Receive the data
   2.) Process the data
   3.) Send the data
   In the first step we are utilizing UART communication with the reader and the MCU. The only thing there is to do is read what is on the serial bus at the event a tag is scanned. The second step was a little bit tricky due to the byte read from the serial bus would cause an overflow for the variable to store them. So I deployed a very simple hashing method of summing the digits read from the serial bus. There are obvious flaws to doing this but for our small scale application. This worked phenomenally. The last step is sending data. For this we had to package the information into a json package and send it across WiFi using a POST request which will ping the remote database. After doing this we were able to successfully receive and send tag information.

4. Backend API : The Backend API is tested both locally and on the cloud using Postman. Postman allows us to send custom headers and Forms/Json objects to any desired endpoint that needs to be tested. This allows us to quickly test endpoints and see how the queries and data are interacting with the database. We can also see what the server's response is, such as seeing what the callback function is outputting. This can either be the docs returned by the query or an error. The ability to test the endpoint without having to depend on front end components is massive. It allows it to build the front-end and back-end independently. It also forces the front end to conform to the backend and the backend was tested out and built before the front end. There were many issues that arose from the testing. Storing an image in the database was challenging. MongoDB typically only allows items up to 4mb of data. You either have to use GridFS

or hope that when you convert the image to a string using react-64 it ends up smaller than the 4mb. This was skipped altogether and instead I used Muller. Muller allows us to take in that file and store it in a folder in the server/client. We then store the path of the file into the database. This stops us from having to pull that file when we try to re-render the page and it is just a more efficient way of dealing with images. We also had to change the header in the request as I used form-data instead of a JSON object to transfer the date. Muller also assists in dealing with the form as express deals with JSON objects. Mongoose Identifiers were also a problem, you can select a specific index in the array of objects from the query using the '$' operator. This is simple enough when you do not have nested arrays. When you do have nested arrays it becomes an issue mainly because you not only need to select the correct document, mongoose typically returns the entire user document when using a query, then you need to be able to iterate through that object and all its nested components to update them. The Front end had way more issues than the backend. This was my first time developing a front end and I was not aware of the difficulty that arose from it. Testing was mainly done using the react-development-server, chrome development tool, and a lot of trial and error. The front end went through 3 different iterations and a lot of the issues that arose were a lack of understanding of react in general. React Hooks, Probs, and how a react-app is structured made it difficult to build a front-end without any experience. Features such as reactivity, responsiveness, re-rendering, state variables, and cookies all created issues when developing the front end. The handling of an image for example, dynamically rendering the elements without having to refresh the state. Dealing with the consequences of using asynchronous features since we had to wait for the database to give us a response. There was a lot of trial and error when dealing with the application, the only thing that did not give me an issue was hitting the api endpoints with axios.

C. *Creating the case*

All the necessary hardware components of the project will be stored in a "Pioneer Plastics Clear Acrylic Display Case for 1:32 Scale Cars, 8" x 3.75" x 3.875."

Figure 10: Acrylic Display Case

This box was chosen because of its size and material. The case is large enough to hold not just the PCB and battery pack but also the antenna as well. It will be modified in such a way that the LCD screen, power switch and antenna are directly exposed to the user of the project. In addition to these three there will be two minor alterations that will display the LCD screen. This will allow the user to directly check to see when to scan an item and when to wait. Also one side of the case will be cut open in order for the antenna to be directly protruding from the case. Additionally the base of the case will be divided into three separate sections. One for the battery pack, antenna and finally with all necessary components in order to function. All necessary hardware components will be contained within the case and once functioning the user will not have to worry about it unless the batteries need to be replaced.

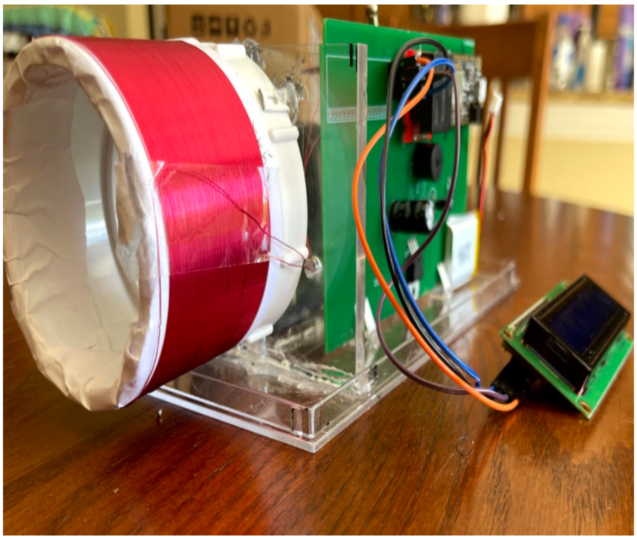

Figure 11 & 12: The Personal Wardrobe Assistant

V. Stretch goals / improvements

A. *[Alternate power system]*

The Final project being presented is solely DC and included a basic DC regulation system. Even though this is more convinetinet the users will have to replace the batteries and have to worry about power consumption. If more time and monetary resources were available a second version would be designed. This one would have an AC adapter and would be required to be plugged into a wall socket. Despite monetary constraints this was initially not implemented in the project because one of the goals was to have the final case be portable and have as few restrictions as possible when placing it somewhere. Certain individuals might not care for this feature and would be completely content with having the Wardrobe assistant be very restricted in movement for the tradeoff of not having to worry about replacing the power supply.

In addition maybe using a rechargeable battery pack would also have been possible. Designing a 12V lithium ion battery pack that can be recharged would offer the best of both worlds to the users. This wasn't considered till later though and because of the restrictions during COVID-19 was dismissed because of time and budgetary constraints.

B. *[Adjustable Antenna Range]*

An adjustable range of the antenna could be something that users would like. They can decide if they want to scan the clothing near the antenna or farther from it. Just by turning a wheel to the clockwise direction could get a max range. Also by turning counterclockwise to reduce the range. We know that may be somewhat useful to people.

Based on the constraints and the short time to build this project, the adjustable antenna range was not possible based on the complexity and the time that can be consumed working on it.

C. *[Expand on User Statistics & Integration with Retailers]*

At this time we hope to show user statistics, such as, what is the user favorite article of clothing, color, etc… . We hope to expand on this by using algorithms to determine what the user should pair their favorite articles of clothing based on their preferences and what they have available. This will require machine learning and is a very unlikely goal. Additionally we hope to integrate retailers in our application. Example, we take the most used article from the user and find a similar item in amazon so the user can purchase something they know they will like. Machine learning is not necessary but will definitely be optimal.

D. *[Using a stand alone WiFi Module]*

We have plans in the making to use a stand alone WiFi Module instead of integrating the entire Adafruit MCU. By doing this we will be using less power on our circuit board and also saving space. The MCU does a lot more things than we actually need and by using a stand alone WiFi Module we can use only what we need for our application. There are dozens upon dozens of WiFi modules on the market and after some online research and comparing similar applications we made a comfortable decision on using the ESP-01s WiFi Module. Currently our PCB is designed and ordered. We've fully tested each component and they're in working condition. Once we've obtained the PCB then we will realize this stretch goal.

VI. Conclusion

In conclusion, the personal wardrobe assistant was designed to bring a new edge of automation to the average household. We all deal with our wardrobe and sorting clothes, but with this device we can enjoy the extra time saved by having our wardrobe organized and tracked. Our group had a very exciting time selecting parts and building the system. Senior design has been an eye opening experience on how engineers communicate in order to get a job done. This realization will help us further as we move forward into our careers beyond our education.

References

[1] Device operational Characteristics
http://www.id-innovations.com/httpdocs/ID-3LA,ID-12LA,ID-20LA.pdf

[2] Multi-layer air Core inductor Calculator
http://www.circuits.dk/calculator_multi_layer_aircore.htm

[3] Adafruit Feather HUZZAH ESP8266

https://learn.adafruit.com/adafruit-feather-huzzah-esp8266/overview

[4] Active RFID vs. Passive RFID: What's the Difference?

https://www.atlasrfidstore.com/rfid-insider/active-rfid-vs-passive-rfid
[5] Power Supply basics
https://www.teamwavelength.com/power-supply-basics/

[6] React Documentation
https://reactjs.org/docs/hooks-state.html

[7] Mongoose Documentation
https://mongoosejs.com/docs/api.html

[8] Material-UI
https://material-ui.com/

Biographies

Maximilian Carroll, a senior student at the University of Central Florida studying Electrical engineering. After Graduation in the summer of 2021, he will start pursuing his masters degree at UCF. He hopes to receive his masters in electrical engineering with a focus on power systems and will begin in the fall semester of 2021.

Efren Cintron, a senior student at the University of Central Florida studying Electrical engineering on Satellites and Communications Systems. After Graduation in the summer of 2021, he will move out to Tennessee to start a new beginning on his engineering career.

Juan Herrera, a senior student at the University of Central Florida studying Computer Engineering. After Graduation in the summer of 2021, he will begin work with Lockheed Martin in Boulder, Colorado. He will be mainly working on Space Systems, and hopes to pursue a masters degree in late 2022.

O'Neal Thomas, a senior student at the University of Central Florida studying Computer Engineering. O'Neal has plans to move to Georgia following graduation at the university in order to begin his career as an electronics engineer working with the Air Force on radio communication systems.