Personal Wardrobe Assistant
Group 17

Efren Cintron ------------- Electrical Engineering

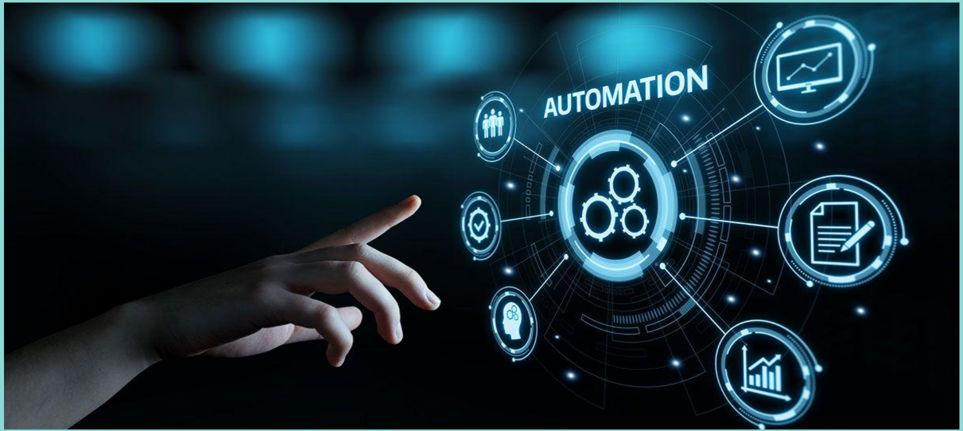
Juan Herrera ------------- Computer Engineering

Maximilian Carroll -------- Electrical Engineering

O’Neal Thomas ----------- Computer Engineering

Motivations

- Wanted to expand the automation process
- Focus on an aspect of daily life that really has not been innovated on by modern technology.
- Create a project where everyone could work on something they enjoyed
- Making life easier,



Goals & Objectives

- Learn & apply concepts of radio frequency technology, power supplies, and microcontroller communications, and developing a full stack application.
- Build a system that will track a person's clothing inventory with precision.
- Help average people save time dealing with articles of clothing. (Making the process of choosing clothes simple)
- Completing our undergraduate careers as electrical & computer engineering students

Specifications

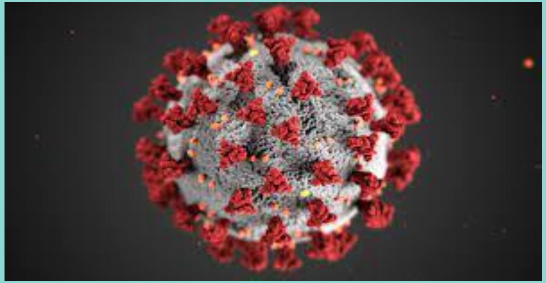
- Passive low frequency scanner range of 5 cm-10cm
- Power source 1, 12 volt battery pack 1600mAH
- Power source 2, a Lithium ion battery 3.7 volts 500mAH
- Connects to WiFi
- Implement a MERN stack (MongoDB, Express, React, NodeJS)
- Adafruit huzzah total current drawn 48mA
- Reader, and LCD screen total current drawn 86 mA

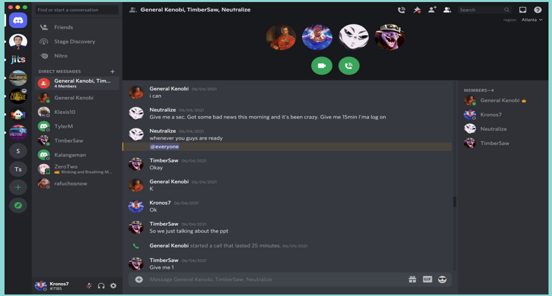
Demonstratables

- 5cm range on the antenna
- Scanning in wardrobe items
- Scanning out wardrobe items

Challenges

- COVID-19
- Limited budget
- Equipment to perform testing on Hardware
- Virtual meetings
- PCB fulfillment in a short period of time





General Kenobi, TimberSaw, Neutralize

General Kenobi 06/04/2021
i can

Neutralize 06/04/2021
Give me a sec. Got some bad news this morning and it's been crazy. Give me 15min I'ma log on

Neutralize 06/04/2021
whenever you guys are ready
@everyone

TimberSaw 06/04/2021
Okay

General Kenobi 06/04/2021
K

Kronos7 06/04/2021
Ok

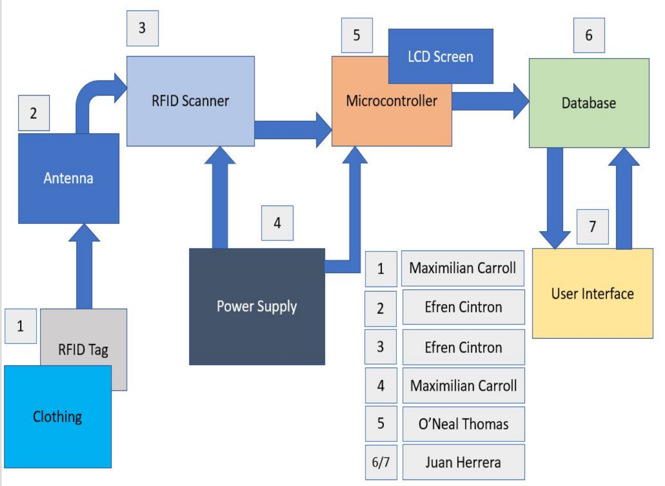
TimberSaw 06/04/2021
So we just talking about the ppt

General Kenobi started a call that lasted 25 minutes. 06/04/2021

TimberSaw 06/04/2021
Give me 1

Distribution of Workload

- **O’Neal Thomas** - Team Leader - Microcontroller / WiFi Module research & part selection/ programming / PCB design
- **Maximilian Carroll** - Team member - Power Supply Design / Regulation / PCB design
- **Efren Cintron** - Team member - RFID research and part selection / Antenna design and tuning / PCB soldering
- **Juan Herrera** - Team member - Full stack web application development.



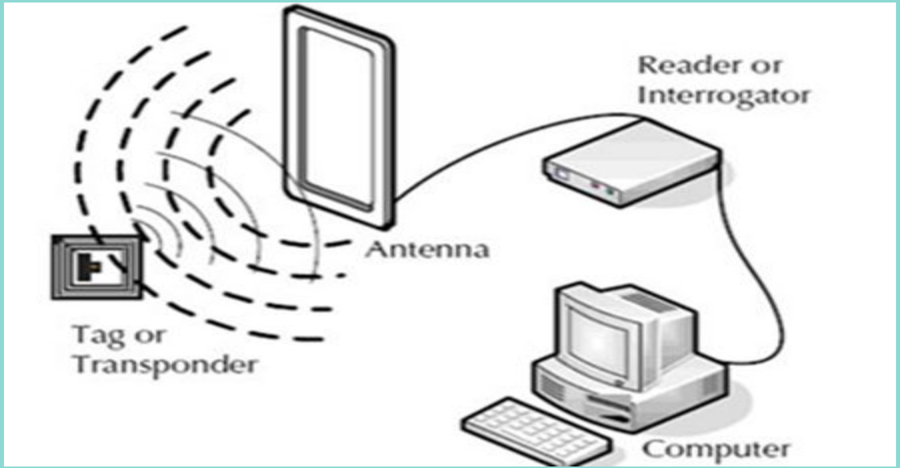
1	Maximilian Carroll
2	Efren Cintron
3	Efren Cintron
4	Maximilian Carroll
5	O’Neal Thomas
6/7	Juan Herrera

AIDC comparison

RFID	Barcode
· Can read from a greater distance and depending on the tag are more accurate · Able to scan multiple tags simultaneously · Collision errors are common when scanning more than one tag at a time · Are able to store large amounts of information · More expensive and complex than barcodes. · Easily jammed by things like wifi or when there is a similar frequency being transmitted	· Smaller and lighter than RFID tags · Less expensive than RFID tags · Almost equal in performance when it comes to accuracy · Not able to store as much information · Cannot be replaced once once a value has been assigned it is permanent and cannot be rewritten.

AIDC System

- Two major automatic identification Data technologies were considered
  - Barcode system
  - RFID system
- Three major components to an RFID system
  - Scanning antenna
  - RFID tag
  - RFID reader
- Passive RFID tag with a frequency of 125KHz
  - Store larger amounts of information
  - Easily replaceable when damaged

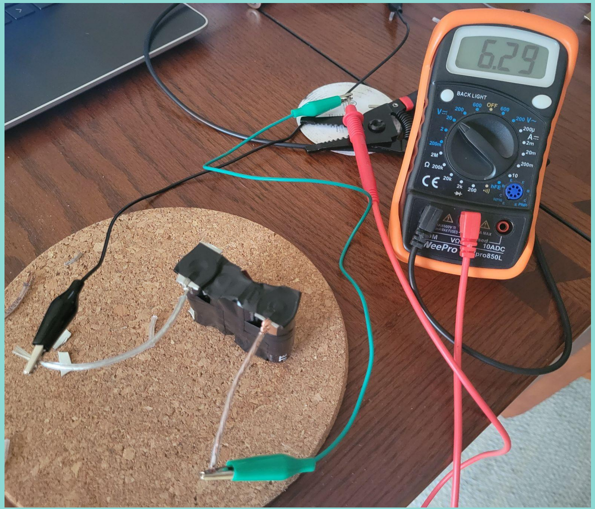


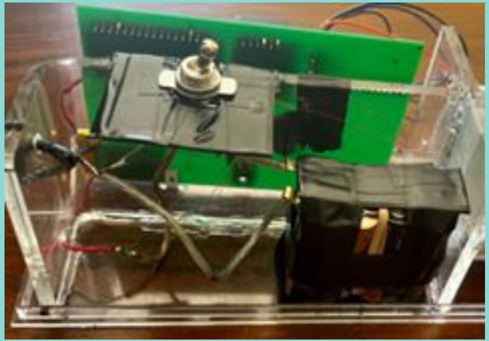
AC versus DC

AC source	DC source
· Single source of power that allows for greater freedom of design. · Requires more components to convert from AC to DC, more expensive and complex. ·Consistent source of power that doesn't need to be replaced. ·Location dependent, because it requires a consistent source of power.	·Offers greater mobility for the final project which will allow the user to choose the location of the project. ·Simpler and cheaper to design because it only consists of less components. ·Easier to troubleshoot because of relative simplicity. ·Once fully drained will not work and sources will need to be replaced or recharged.

Voltage supply

- Solely a DC power source
  - Greater mobility for the project
  - Electronic components require DC power
  - Easy to replace
- Batteries
  - 8 AA batteries connected in series produces 12 volts
  - Responsible for power reader, LCD and antenna
  - Lithium ion battery 3.7 volts
  - Sley responsible for power microcontroller





DC Regulator 5 volt

- Several options to achieve desired output voltage

  Step down regulator

  Zener regulator

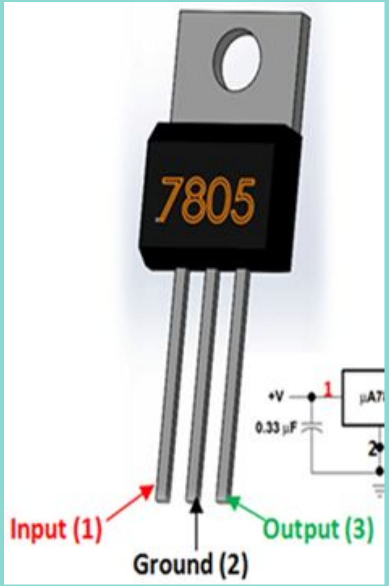
  Voltage divider

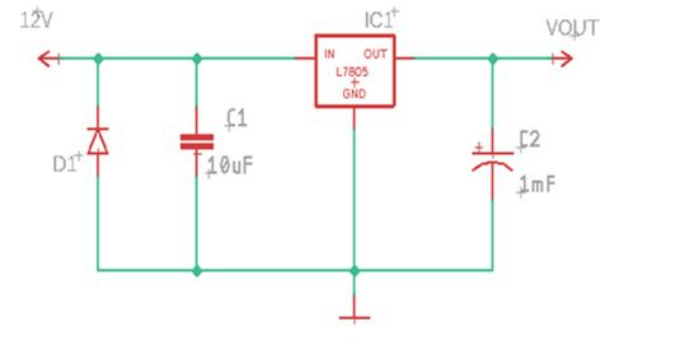
- Capacitors and diodes

  1N4001 Diode to regulate current and protect against overvoltage

  Two smoothing capacitors to improve the voltage output

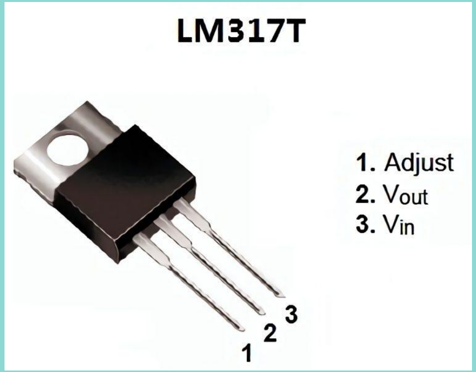
- L7805 was recommended in the datasheet for the RFID reader

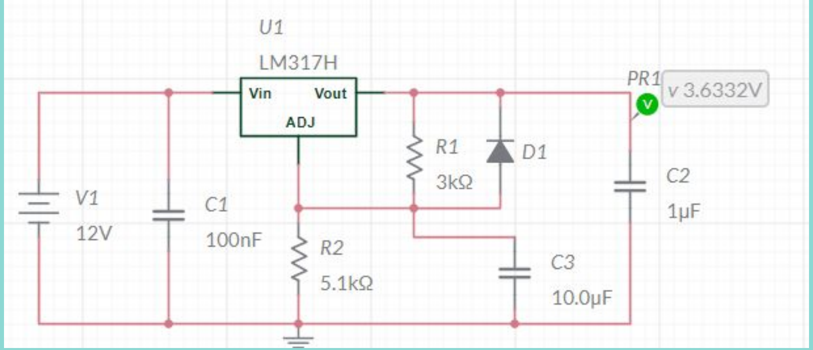


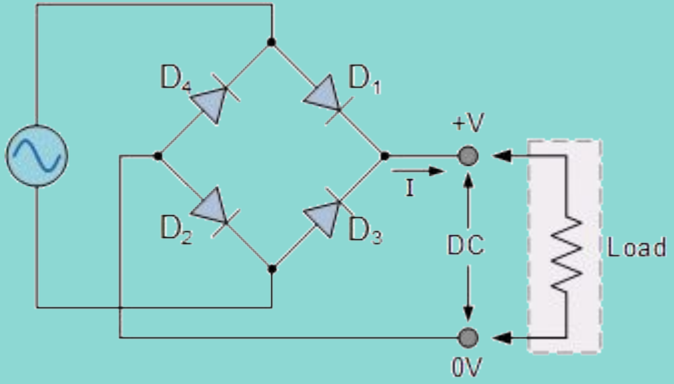


3.3v-3.6v regulator (Stretch goal)

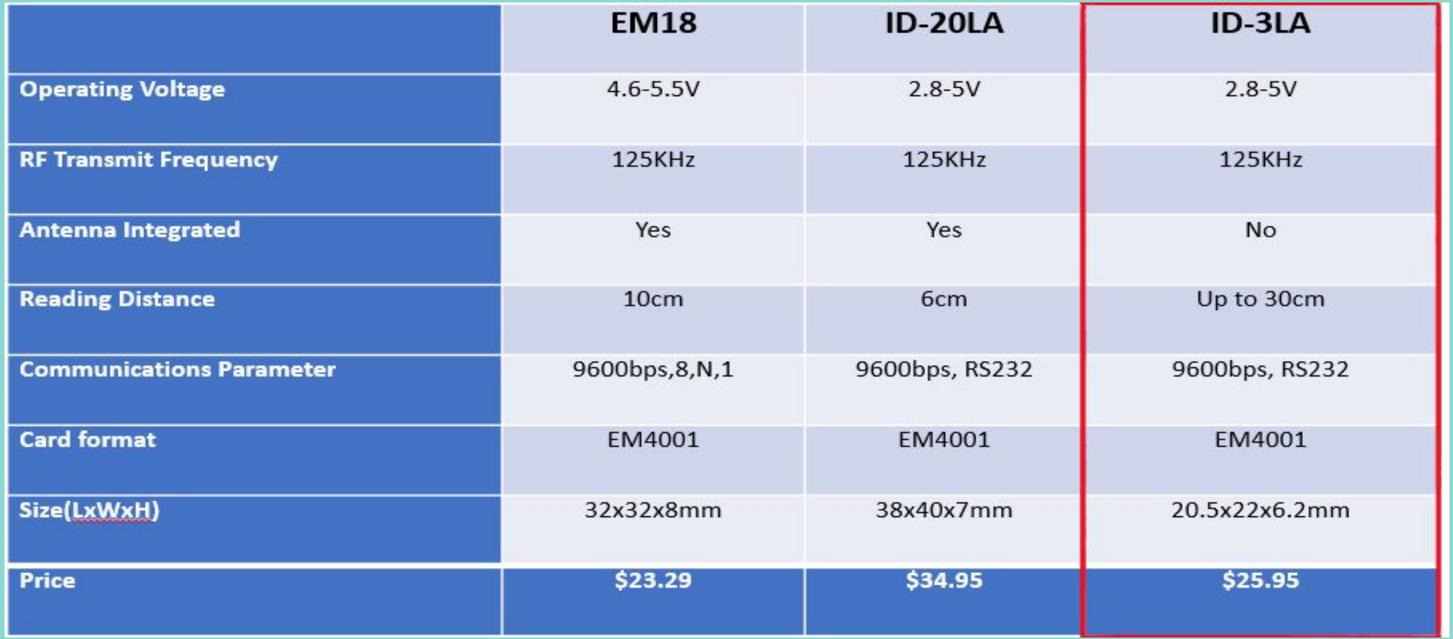
- One of the major stretch goals is to try and create a second version of the project but have it be powered by an AC power source that can be directly connected to the wall.
- This will get rid of the need to replace batteries and once connected users don't have to keep track of battery life anymore.
- Only need to power the wifi module and the RFID reader. For both components of the project 3.3 to 3.6 volts is sufficient to turn on and power the project.







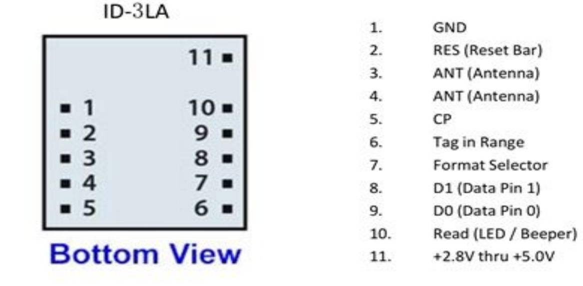
RFID Reader



	EM18	ID-20LA	ID-3LA
Operating Voltage	4.6-5.5V	2.8-5V	2.8-5V
RF Transmit Frequency	125KHz	125KHz	125KHz
Antenna Integrated	Yes	Yes	No
Reading Distance	10cm	6cm	Up to 30cm
Communications Parameter	9600bps,8,N,1	9600bps, RS232	9600bps, RS232
Card format	EM4001	EM4001	EM4001
Size(LxWxH)	32x32x8mm	38x40x7mm	20.5x22x6.2mm
Price	**$23.29**	**$34.95**	**$25.95**

RFID Reader ID-3LA

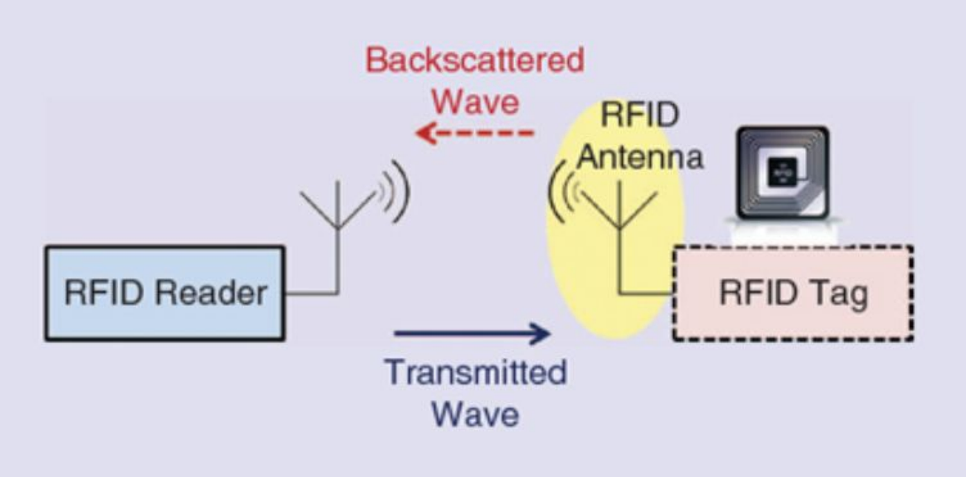
- Acts like a transceiver
- Arduino/ I2C compatible
- Energy efficient



1.	GND
2.	RES (Reset Bar)
3.	ANT (Antenna)
4.	ANT (Antenna)
5.	CP
6.	Tag in Range
7.	Format Selector
8.	D1 (Data Pin 1)
9.	D0 (Data Pin 0)
10.	Read (LED / Beeper)
11.	+2.8V thru +5.0V

RFID Antenna

The RFID Antenna performs two important functions :
They transmit power to the RFID tags and receive data back from the activated tags.



Specifications:

- 125 KHz
- Reading range from 5 to 10cm
- Compact size

RFID Antenna Material Selection

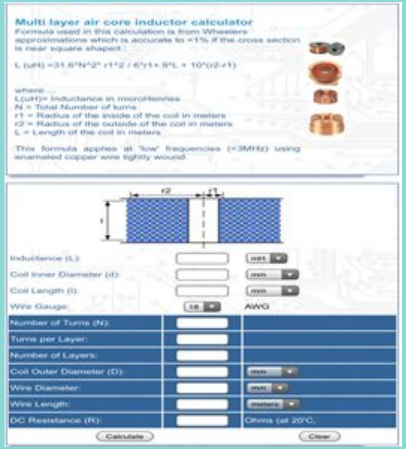
- PVC pipes(2,3, and 4 inch diameter)
- 28, 30, 32 Gauge Copper magnet wire
- 3 PVC pipes diameter with three different copper wires = 9 antennas

Homemade RFID Antennas

- Use of online calculator to calculate how many turns does each pipe needs

- Winding copper wire to the pipe

- Burning insulation of the two ends of the wire to solder bigger wire to fit into the breadboard



Multi layer air core inductor calculator

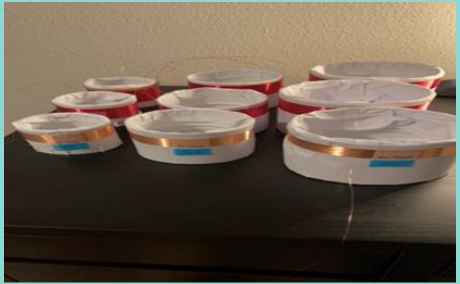
Formula used in this calculation is from Wheelers approximations which is accurate to <1% if the cross section is near square shaped.

L (uH) =31.6*N^2* r1^2 / 6*r1+ 9*L + 10*(r2-r1)

where ....
L(uH)= Inductance in microHenries
N = Total Number of turns
r1 = Radius of the inside of the coil in meters
r2 = Radius of the outside of the coil in meters
L = Length of the coil in meters

This formula applies at 'low' frequencies (<3MHz) using enameled copper wire tightly wound.

Inductance (L):		mH
Coil Inner Diameter (d):		mm
Coil Length (l):		mm
Wire Gauge:	18	AWG
Number of Turns (N):		
Turns per Layer:		
Number of Layers:		
Coil Outer Diameter (D):		mm
Wire Diameter:		mm
Wire Length:		meters
DC Resistance (R):		Ohms (at 20°C.

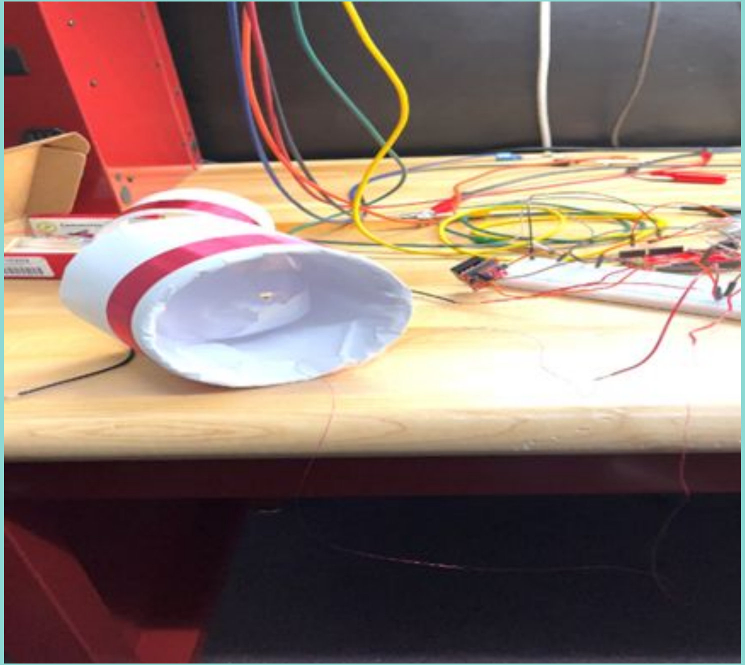


Testing RFID Antennas

- Use of microcontroller MSP430FR6989 as a power source connected to the breadboard

- RFID reader and SparkFun RFID starter kit to test each antenna read range

- Buzzer makes sound when tag is detected on range



RFID Antenna Range Results

AWG	Diameter Size (inches)	Number of Turns	Voltage Peak-to-Peak (V)	Range (cm)	Waveform Compress/Expands
28	2	100	3.8	N/A	Expands
28	3	90	6.28	1	Compress
28	4	80	9.6	5	Compress
30	2	100	2.84	N/A	Expands
30	3	90	6.8	7	Compress
30	4	80	15	13	Compress
32	2	100	10.1	0.5	Compress
32	3	90	10.2	2.5	Compress
32	4	80	10.2	5.5	Compress

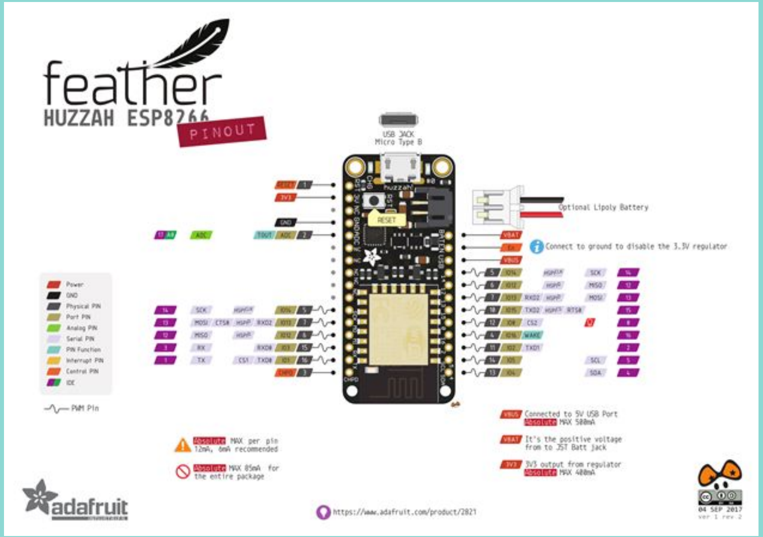
RFID Antennas with no tuning

AWG	Diameter Size (inches)	Number of Turns	Voltage Peak-to-Peak (V)	Range (cm)
28	2	112	7.6	5
28	3	84	9.6	7
28	4	75	10.5	8
30	2	114	8.9	8
30	3	86	8.6	13
30	4	76	17	9
32	2	94	11.5	7
32	3	87	13.6	11
32	4	74	14	13

RFID Antennas with tuning

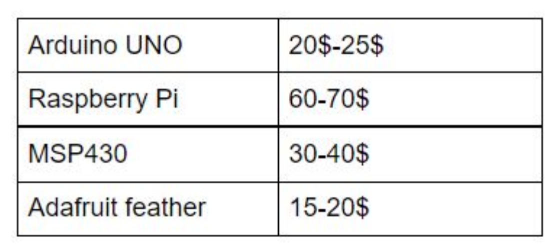
Selection of Microcontroller

- Adafruit Feather Huzzah
- Small compact size
- I2C + UART communications
- Arduino IDE compatible (Easily Programmable)
- Integrated WiFi Module ESP8266


feather
HUZZAH ESP8266
PINOUT
USB JACK Micro Type B
RESET
Optional Lipoly Battery
Connect to ground to disable the 3.3V regulator
Power
GND
Physical PIN
Port PIN
Analog PIN
Serial PIN
PIN Function
Interrupt PIN
Control PIN
IDE
PWM Pin
Absolute MAX per pin 12mA, 6mA recommended
Absolute MAX 85mA for the entire package
VBUS Connected to 5V USB Port Absolute MAX 500mA
VBAT It's the positive voltage from to JST Batt Jack
3V3 3V3 output from regulator Absolute MAX 400mA
adafruit
https://www.adafruit.com/product/2821
04 SEP 2017
ver 1 rev 2

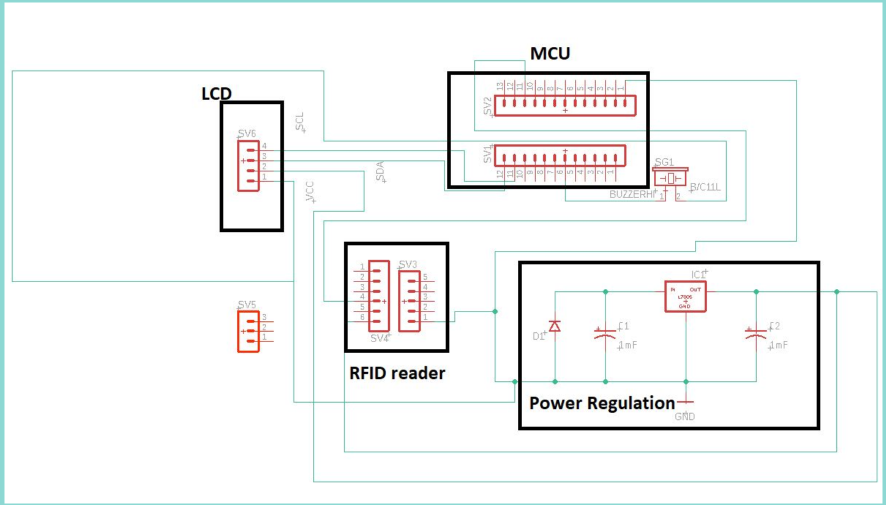
Other considerations:

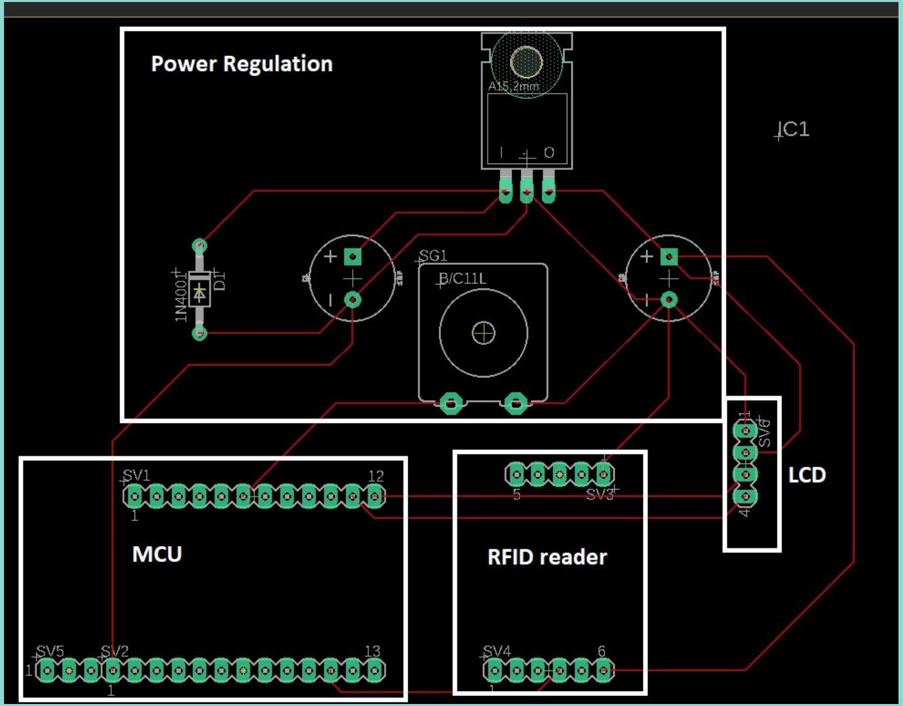
MSP430, Arduino UNO, Raspberry Pi



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

PCB version 1



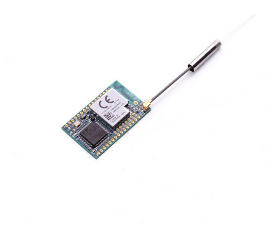


Using a stand alone WiFi Module (Stretch goal)

- Our second design of the Personal Wardrobe Assistant will break away from the use of the entire Adafruit microcontroller and use a standalone WiFi Module. Using this design will save space on the PCB and require less power in order to operate.
- There are literally dozens of WiFi Modules to select from on the market, but selecting the best option for our application became easy after some online research.
- We chose the ESP-01s for our WiFi application due to programmability and availability.

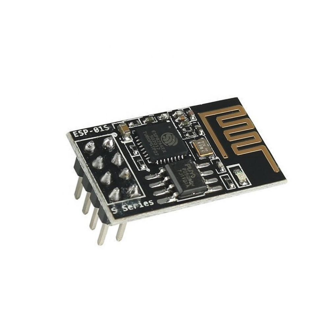
Air602 WiFi module

EMW3162

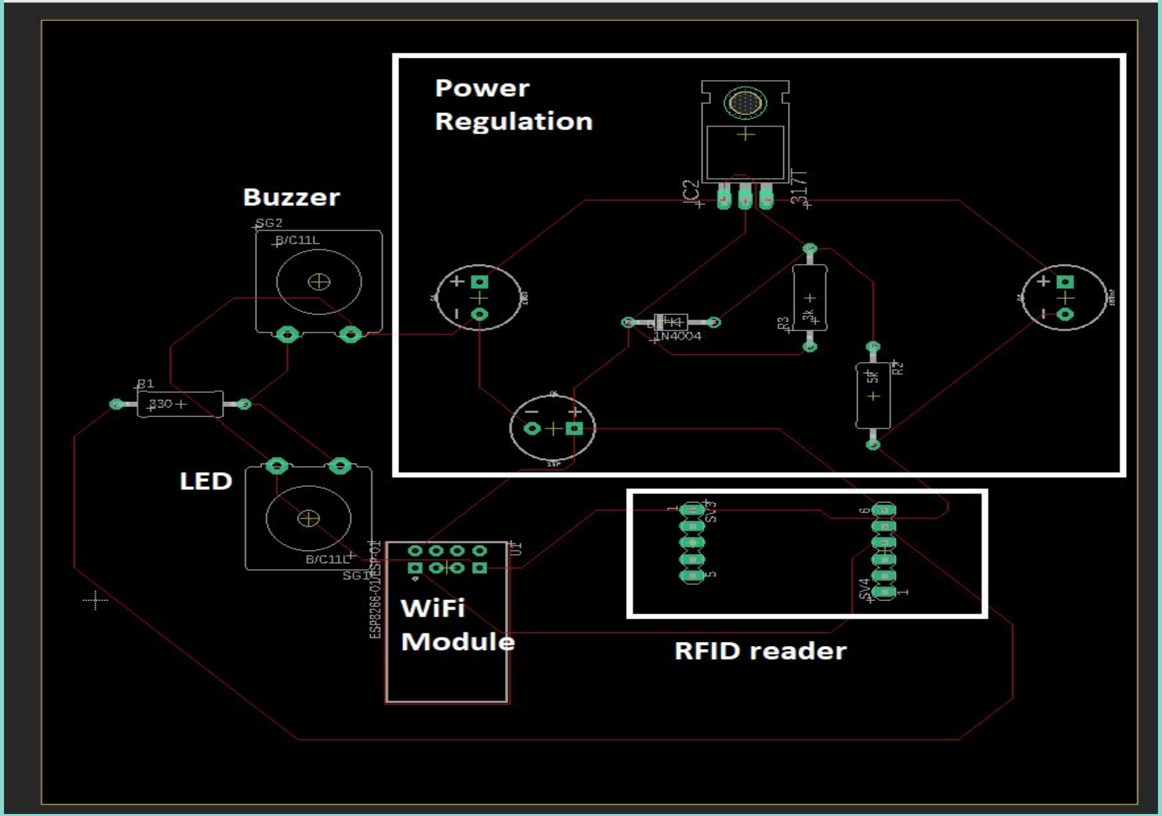


AMW007 & AMW037

ESP-01s

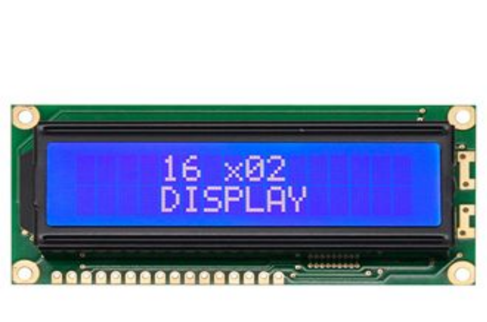


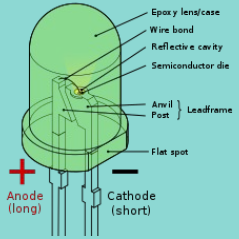
PCB version 2 (Stretch goal)



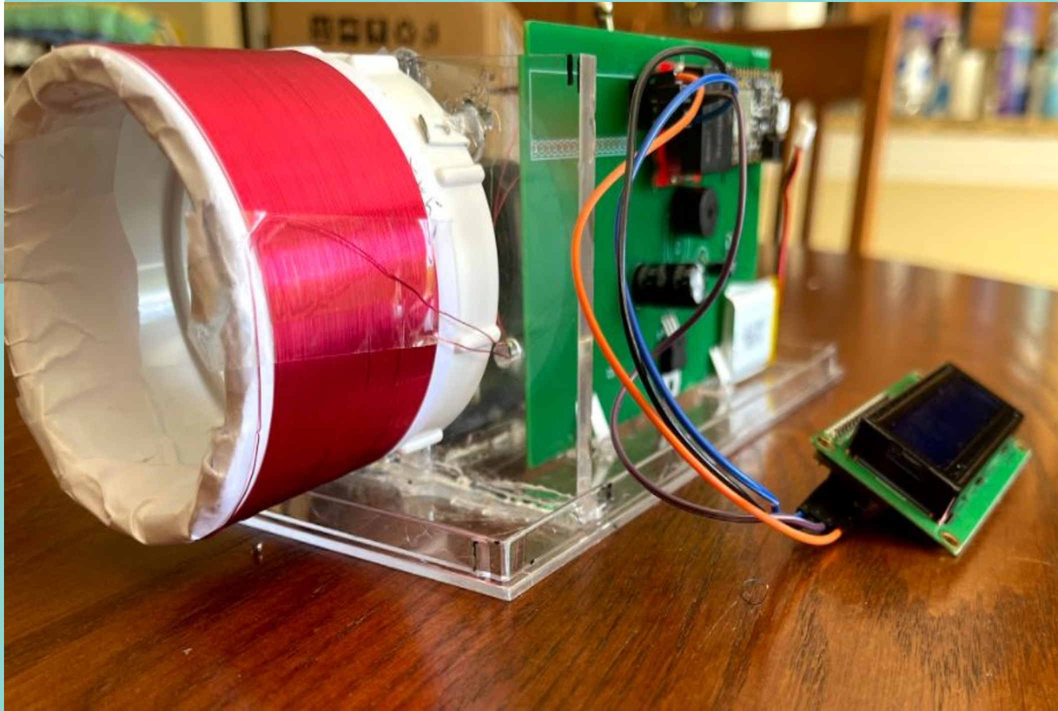
Integrating LCD Display / Buzzer / LED

- On our first design will include an LCD display and a buzzer for the user to have visual and audio queues for when a tag has been scanned and it in the system. We decided to go with a 16x2 LCD due to simplicity and space. There's no need for a large display ex: 20x4. We also decided to go with an active buzzer because we only need 1 beep per scan and an active buzzer only needs a logic signal in order to operate.
- Our second design towards our stretch goal will exclude the LCD display and include an LED. The ESP-01s does not have a secure I2C connection for our application thus we cannot integrate an LCD.



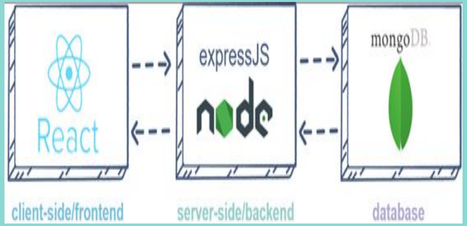


WARNER ROBINS ALC



MERN Stack

- Familiarity: I have recently been working with Node and MongodB.
- NoSQL databases are more flexible and dynamic which works perfectly with the fact that features can be added or deleted as we get closer to the deadline. Additionally, NoSQL databases are best used for ever-changing data sets.
- Our queries are very simple this utilizing MongodB's high performance.
- Angular was not an option due to its learning curve and typescript. It was between Vue.js and React.js. Ultimately went with react since there is more documentation online.
- I wanted to work with Fastify in this project but I have previously used Express and did not want to deviate.



Back-End

- As stated before we will be using Node.

Main libraries being used:

Express: API development and Routes

Morgan: Logger

Helmet: Protects Headers

Cors: Cross-Origin Resource Sharing

Mongoose: ODM library for MongoDB

BCrypt: Hashing passwords

Multer: Allow us to take in files,store them on a folder, and keep the file path in a variable

Structure for Database

- The Main schema will hold basic information of the user, as well as initializing the email, security, and wardrobe array. For future reference ID numbers are automatically created once a record is inserted into the database.

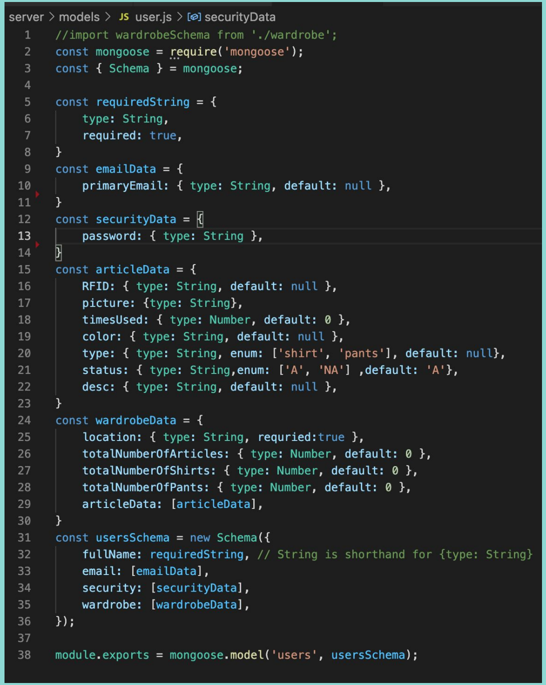
- The security schema will hold the password.. Passwords will be hashed (Bcript) once they reach the API.

- Email schema will hold the Primary Email. Primary Email will be used as a username.

- The main wardrobe schema will hold the number in relation to the user, location, and wardrobe data array are initialized here.

- Wardrobe data schema will hold the number of articles stored in the wardrobe, and have an array of articles.

- Article Data schema holds the RFID tag data, Picture Path, Type, Color, Status, and Times Used.



server > models > JS user.js > securityData

```javascript
//import wardrobeSchema from './wardrobe';
const mongoose = require('mongoose');
const { Schema } = mongoose;

const requiredString = {
    type: String,
    required: true,
}
const emailData = {
    primaryEmail: { type: String, default: null },
}
const securityData = {
    password: { type: String },
}
const articleData = {
    RFID: { type: String, default: null },
    picture: {type: String},
    timesUsed: { type: Number, default: 0 },
    color: { type: String, default: null },
    type: { type: String, enum: ['shirt', 'pants'], default: null},
    status: { type: String,enum: ['A', 'NA'] ,default: 'A'},
    desc: { type: String, default: null },
}
const wardrobeData = {
    location: { type: String, requried:true },
    totalNumberOfArticles: { type: Number, default: 0 },
    totalNumberOfShirts: { type: Number, default: 0 },
    totalNumberOfPants: { type: Number, default: 0 },
    articleData: [articleData],
}
const usersSchema = new Schema({
    fullName: requiredString, // String is shorthand for {type: String}
    email: [emailData],
    security: [securityData],
    wardrobe: [wardrobeData],
});

module.exports = mongoose.model('users', usersSchema);
```

Front end

- As stated before we will be using React. To create the functionally of the application we are going to be using JSX.

Main libraries being used:

Axios : Used to generate request to the back-end server

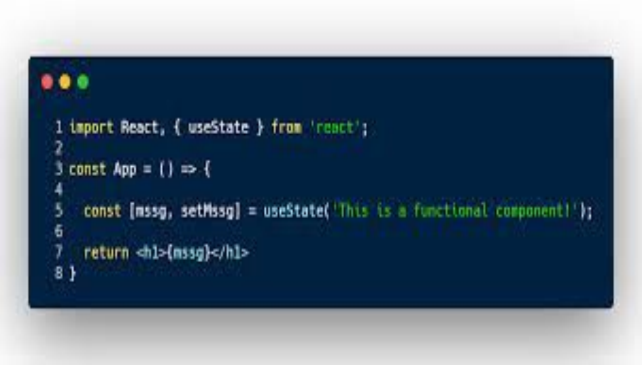
React-router-dom: Allows us to create routes, and link the different pages

React UseState, and UseEffect, and probs

React UseState allows us to state variables. Think of them as global variables that are keeped in between function calls, they are set to their default when the main function is rendered. It allows us to store variables such as from input values, form submissions, data being received from the api, modal status and other information that needs to be recorded and tracked.

It also allows to re render the page without having to refresh. This is accomplished by calling the /logs/userInfo database and set the state of variables needed to render out the page after waiting for the information coming from the back-end.

Lastly, the session is stored by keeping the sessionID (userId) stored locally.

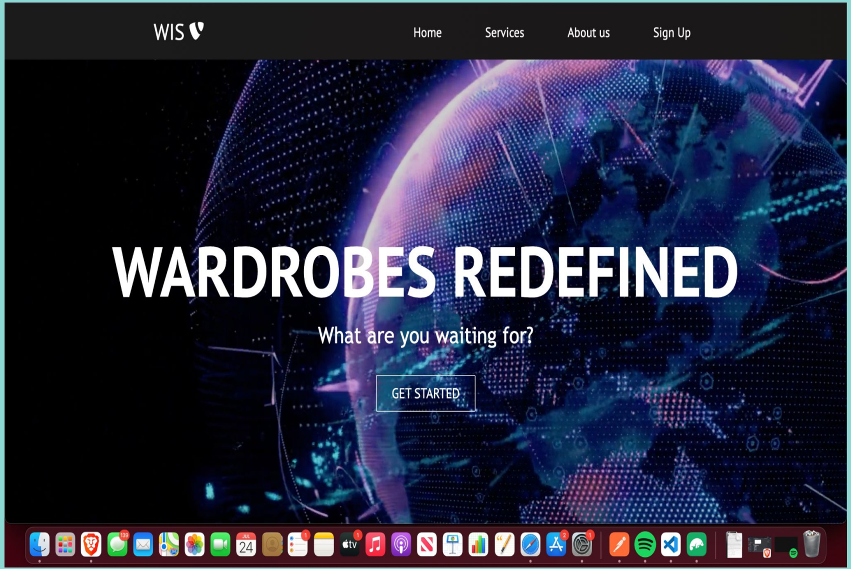


```
import React, { useState } from 'react';

const App = () => {

  const [mssg, setMssg] = useState('This is a functional component!');

  return <h1>{mssg}</h1>
}
```

Front end

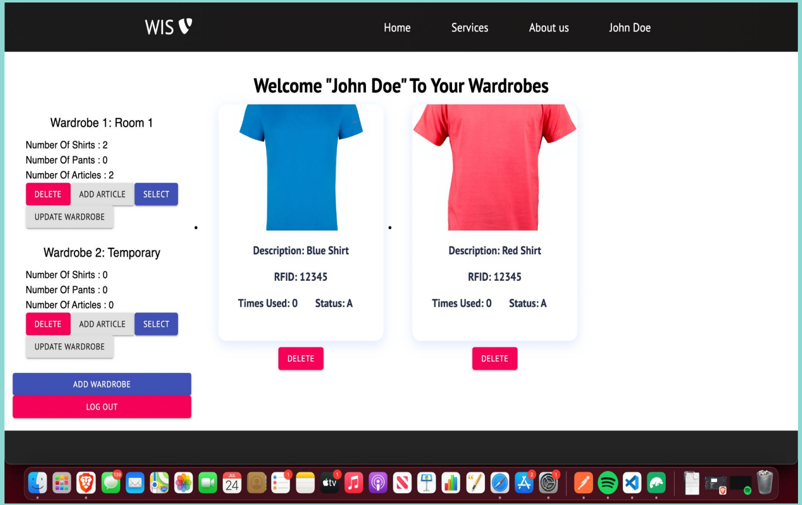
- Aesthetically wise some components have their own personalized CSS, BootStrap but mostly Material-UI.

Material-UI : is a react library that is similar to bootstrap. It has its own elements that mimic html elements. The Material-UI docs gives us all the information to develop a reactive and aesthetically pleasing application. Components such as typography, Box, Grid, TextField, Spacing, etc.. allows us to simply pick probs that the we want on the application and renders them without the use of any CSS.



WARDROBES REDEFINED

What are you waiting for?



Welcome "John Doe" To Your Wardrobes

Wardrobe 1: Room 1

Number Of Shirts : 2
Number Of Pants : 0
Number Of Articles : 2

Wardrobe 2: Temporary

Number Of Shirts : 0
Number Of Pants : 0
Number Of Articles : 0

Description: Blue Shirt

RFID: 12345

Times Used: 0 Status: A

Description: Red Shirt

RFID: 12345

Times Used: 0 Status: A

Budget

Virtual Wardrobe Assistant						
Number	Part Number	Description	Vendor	Price Per Unit	Amount	Total Estimated Price
1	MSP - 430	MSP-EXP430G2 LaunchPad	Texas Instruments	$23.40	1	$23.40
2		PCB Board	EAGLE	$5.00	15	$75
3		12 Volt battery pack(2)	Co rode	$4.00	1	$8.00
4		DNS and Server Hosting	Go Daddy	$49.99	1	$49.99
5	ID-3LA	ID-3LA RFID Reader	Electromaker	$25.99	1	$25.99
6	EM18	EM18 RFID reader	AliExpress	$5.99	1	$5.99
7		Pack of batteries (20)	Duracell	$0.75	1	$14.99
8		24 value 500pcs capacitors	OCR	$0.03	1	$14.99
9		RFID tags 125kHz	Yarong RFID	$7.98	1	$7.98

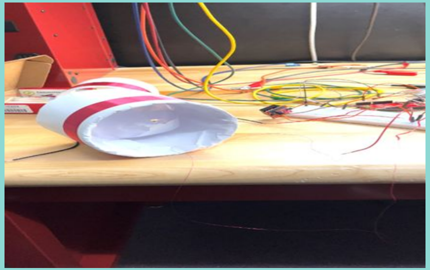
Budget Cont.

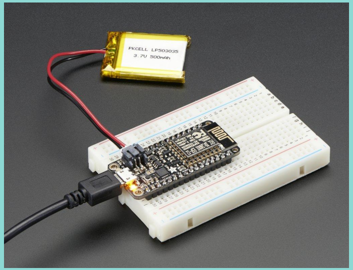
Virtual Wardrobe Assistant						
Number	Part Number	Description	Vendor	Price Per Unit	Amount	Total Estimated Price
10		15 Diodes (1N4001)	Fairchild semiconductor	$0.20	1	$2.98
11		Regulators (10 L7805)	C YUMU	$6.99	1	$6.99
12		28 AWG Enameled Copper	BNTECHGO	$8.48	1	$8.48
13		30 AWG Enameled Copper	BNTECHGO	$8.69	1	$8.69
14		32 AWG Enameled Copper	BNTECHGO	$8.69	1	$8.69
15		Adafruit feather HUZZAH	Adafruit	$18.00	1	$18.00
16		Active Buzzer 5V	Amazon	$1.00	5	$5.00
17		Plexiglass Case	Amazon	$30	1	$30.00
	Total					$315.16

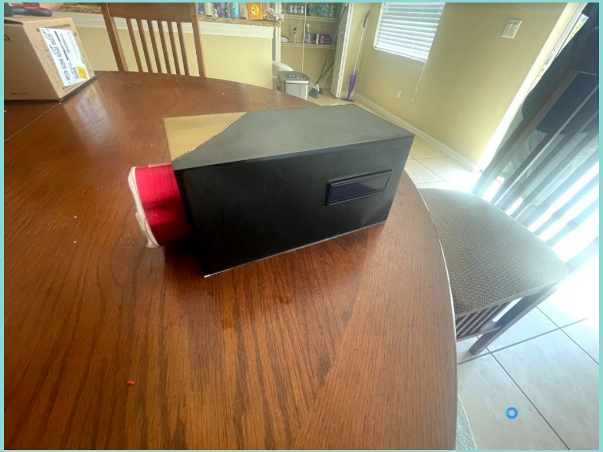
Completion

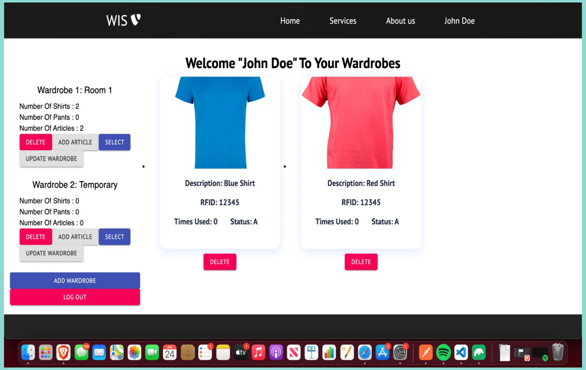
Part Selection	100%
DC Voltage Regulator	100%
RFID Antenna	100%
LCD Display Testing	100%
Microcontroller Testing	100%
Database	100%
PCB Board Design	100%
Wifi Transmission	100%
PCB Testing	100%
Casing	100%
Buzzer Install	100%

Conclusion









Welcome "John Doe" To Your Wardrobes

Wardrobe 1: Room 1

Number Of Shirts : 2
Number Of Pants : 0
Number Of Articles : 2

Wardrobe 2: Temporary

Number Of Shirts : 0
Number Of Pants : 0
Number Of Articles : 0

Description: Blue Shirt
RFID: 12345
Times Used: 0 Status: A

Description: Red Shirt
RFID: 12345
Times Used: 0 Status: A