Senior Design

UNIVERSITY OF
CENTRAL FLORIDA

Increasing Immersion and Interactivity in Themed Entertainment Environments Through the Usage of RFID Technology

Group 15

Darren Yong - Electrical Engineering

Kalindi Desai - Computer Engineering

Rebecca Casimir - Computer Engineering

Trey Sandefur - Electrical Engineering

Table of Contents

Section Name	Page Number
List of Figures	**v**
List of Tables	**viii**
1.0 Executive Summary	**1**
2.0 Project Description	**3**
2.1 Motivation	3
2.2 Project Background	3
2.3 Project Goals and Objectives	3
2.4 Project Narrative	4
2.5 Project Requirements and Specifications	5
2.5.1 Updates to Requirements and Specifications	7
2.6 Project Constraints	8
2.7 House of Quality	9
Legend	10
2.8 Project Prototype Illustration	11
2.9 Project Block Diagrams	11
2.9.1 RFID Device Design Block Diagram	12
2.9.2 Pokéball Design Block Diagram	13
2.9.3 Project Software Block Diagram	14
3.0 Research Related to Project Definition	**15**
3.1 Existing Products	15
3.1.1 Disney MagicBand	15
3.1.2 Universal Interactive Wands	17
3.2 Existing Applications	19
RFID	19
3.3 Software Tools	20
3.3.1 Communication	20
3.3.1.1 Discord	20
3.3.1.2 Google Drive	21
3.3.2 Developmental	21
3.3.2.1 Github	21
3.3.2.2 Fusion 360	22
3.3.2.3 Arduino IDE	22
3.3.2.4 Code Composer Studio	22

4.0 Standards and Constraints **23**
4.1 Related Standards 23
4.1.1 Quality Management System Standard 24
4.1.2 Software Testing Standard 26
4.1.3 Communication Standards 29
4.1.4 Documentation Standards 32
4.2 Realistic Design Constraints 35
4.2.1 Functional Constraints 35
4.2.2 Safety Constraints 36
4.2.3 Quality Constraints 36
4.2.4 Manufacturing Constraints 36
4.2.5 Economic and Time Constraints 37
4.2.6 Aesthetic Constraints 37

5.0 Hardware Research **38**
5.1 Communication Networks 38
5.1.1 Wired LAN 38
5.1.2 Wireless LAN 38
5.1.3 Infrared Communication 39
5.1.4 Bluetooth Communication 39
5.1.5 Radio Frequency Identification 40
5.1.5.1 Passive Tags 41
5.1.5.2 Active Tags 42
5.2 Microcontrollers 43
5.2.1 Microcontroller Selection 43
5.2.2 RFID Transmitter Module 52
5.2.3 RFID Reading Module 53
5.2.4 Printed Circuit Board 55
5.2.5 Battery 56
5.3 Pokéball Shell Material 56
5.3.1 Staying True to the World of Pokémon 56
5.3.2 Brief History on the Pokéball 57
5.3.2 Material Research 58

6.0 Software Research **62**
6.1 Programming Languages 62
6.2 RFID Programming 64
6.3 Software Flow 65
6.3.1 Updates to Software Flow 67

6.4 Software Security 67
6.4.1 Minimum Password Length 68
6.4.2 2 Factor Authentication (2FA) 68
6.4.3 Tiered Admin 70
6.4.4 Location Based Login 70
6.4.5 Error Codes 72
6.5 Frontend UI 74
6.5.1 MERN 76
6.5.2 MongoDB 77
6.5.3 Express 78
6.5.4 React JS 79
7.0 Prototype Construction and Coding 82
7.1 Project Hardware Assembly 82
7.1.1 Assembly Using MSP430 82
7.1.2 Assembly Using Arduino UNO 83
7.2 Printed Circuit Board Vendor 86
7.3 Final Coding Plan 86
7.4 Final Pokéball Shell Design 90
7.4.1 Pokéball Design Utilizing Fusion 360 92
7.4.2 Final 3D Printing Process 97
8.0 Project Prototype Testing Plan 100
8.1 Basic Hardware Testing 100
8.2 Basic Software Testing 104
8.2.1 Initial Graphical User Interface Development 111
8.3 Pokéball Shell Testing 118
9.0 Administrative Content 121
9.1 Budget Estimate 121
9.2 Project Milestones 122
Appendix 124
References 124

List of Figures

Figure 1: House of Quality Specifications	8
Figure 2: Pokéball Interacting with RFID Reading Device	9
Figure 3: RFID Device Design Block Diagram	10
Figure 4: Pokéball Design Block Diagram	11
Figure 5: Software Design Block Diagram	12
Figure 6: MagicBand vs Smartphone Connection	14
Figure 7: Variety of Disney's MagicBands	14
Figure 8: An Interactive Wand from Universal	15
Figure 9: Chart of Top 10 Media Franchises	16
Figure 10: A Medallion at Universal	17
Figure 11: Schematic Representation of the Elements ISO/IEC/IEEE 90003	22
Figure 12: PDCA Cycle from ISO/IEC/IEEE 90003	23
Figure 13: Verification and Validation Methods from ISO/IEC/IEEE 29119-1	25
Figure 14: Test Model in Perspective From ISO/IEC/IEEE 29119-1	26
Figure 15: Passive RFID Key and Tag	40
Figure 16: Active RFID Transponder	41
Figure 17: FRAM Schematic	42
Figure 18: MSP430FR6989 Visual	44
Figure 19: Arduino UNO Visual	46
Figure 20: Atmega328P Visual	47
Figure 21: EM18 RFID Reading Module Pin Layout	51
Figure 22: RC522 RFID Module	52
Figure 23: Arduino with RFID Shield	53
Figure 24: Pokéball Reference (Provided by The Pokémon Group)	55
Figure 25: RFID Software Flow	62
Figure 26: Backend Software Flowchart	64
Figure 27: Updated Software Flowchart	
Figure 27: Authentication Flowchart	66
Figure 28: Sample Portal Page	72
Figure 28: MERN Flowchart	74
Figure 29: SQL vs MongoDB Venn Diagram	75
Figure 30: Express Flowchart	76
Figure 31: React JS Flowchart	78
Figure 32: EM18 RFID Reading Module Pin Layout	79

Figure 33: RC522 Module to Arduino UNO Schematic 81
Figure 34: Schematic of Assembled Breadboard 82
Figure 35: Starter Pokémon from Generation 1 85
Figure 36: Starter Pokémon Evolution 85
Figure 37: Basic Pokémon Battle 86
Figure 38: Awardable Gym Badges 86
Figure 39: Articles 1 & 2 of the Assembly 87
Figure 40: Articles 3, 4, & 5 of the Assembly 88
Figure 41: Article 6 of the Assembly 88
Figure 42: Initial Diameter Sketch 90
Figure 43: Sketch Models of Pokéball 91
Figure 44: Front Facing Extrude Action 92
Figure 45: Completed Articles 1 & 3 93
Figure 46: Final 3D Model Prior to Colorization 94
Figure 47: Final Pokéball Model Post Colorization 96
Figure 48: Hardware Prototype 97
Figure 49: Key Card and Key Tag Chain for RFID Test 98
Figure 50: Breadboard Setup with 1602 LCD 99
Figure 51: Schematic of Assembled Breadboard 100
Figure 52: RFID Reader Full Prototype 100
Figure 53: RFID Reader After Scanning Key 101
Figure 54: LoRa RFM95W Transceiver 109
Figure 55: ATMEGA328P-AU Microcontroller 109
Figure 56: FT232RL FTDI Module 110
Figure 57: Final Base Station 110
Figure 58: Example Code Used to Test RC522 111
Figure 59: Serial Monitor Showing Key UID 112
Figure 60: Arduino IDE Code for Full Prototype 113
Figure 61: Arduino IDE Code for LED Toggle 115
Figure 62: Graphical Interface for Toggling LED 116
Figure 63: Toggling the LED Using Processing 116
Figure 64: Start Screen for Prototype 116
Figure 65: Start Screen Processing Code 117
Figure 66: Character Name Selection 118
Figure 67: Character Name Selection Input 119
Figure 68: Pokémon Selection Interface 120

Figure 69: More Info Popup for Bulbasaur 121
Figure 70: More Info Popup for Charmander 121
Figure 71: More Info Popup for Squirtle 121
Figure 72: Processing Code for Pokémon Selection 122
Figure 73: Processing Code for Pokémon Selection Buttons 123
Figure 74: Pokéball Drop Test 125

List of Tables

Table 1: Engineering Specifications 6
Table 2: Updated Engineering Specifications
Table 2: House of Quality Legend 9
Table 3: FCC Restricted Bands of Operation for Intentional Radiators 28
Table 4: Field Strength Emissions from Intentional Radiators 30
Table 5: USITT RF Bible Matrix Example 31
Table 6: Example Punch List 32
Table 7: RFID Comparison Table 38
Table 8: FRAM vs FLASH Comparisons 43
Table 9: MSP430FR6989 Data Sheet Specifications 44
Table 10: Arduino UNO Board Specifications 46
Table 11: Atmega328P Data Sheet Research 48
Table 12: MCU Comparisons 49
Table 13: Differences Between Passive and Active RFID Tags 50
Table 14: Differences Between Through Hole and Surface Mount 51
Table 15: 3D Printed Filament Comparisons 57
Table 16: 3D Printed Filament Pros and Cons Comparison 58
Table 17: Programming Language Comparisons 61
Table 18: Tiered Admin Chart 67
Table 19: Browser Compatibility 69
Table 20: Error Code Descriptions 70
Table 21: Error Code Descriptions 73
Table 22: Error Code Descriptions 71
Table 23: EM18 Wiring Description 80
Table 24: Wiring For RC522 to Arduino 81
Table 25: 1602 LCD Connection Guide 83
Table 26: General Final Coding Plan Checklist 84
Table 27: Budget Estimate 127
Table 28: Final Project Budget 128
Table 29: Senior Design 1 Project Milestones 129
Table 30: Senior Design 2 Project Milestones 130

1.0 Executive Summary

There is something fascinating about being able to interact with your favorite virtual characters either from a television show or video game. This is where our project brings the user and their favorite Pokémon closer than ever before. This project is designed to mimic the interactions experienced in the Pokémon shows and games in the physical world. The objective of this project is to give a user a Pokéball and allow it to interact with the environment around them. The Pokéball utilizes Radio Frequency Identification (RFID) to identify what is around them.

Pokémon has long been a staple in pop culture. From television to video games, published works to vast product lines, but not yet has the world of Pokémon breached into physical interactive experiences. This project begs the question: what would the world of Pokémon look like as a themed land inside one of the major theme parks around the world? As the world of video games finally breaches into the themed entertainment industry, and immersive experience market, it is destined that in the future a project similar to this could become a reality. This project presents and asks for the reader to immerse themselves into the world of Pokémon, and ponder what it is they would desire to see in a real environment they could experience. In this world, you are the trainer. You are the one who is in control of your own Pokémon destiny. Whether you're 5 or 85, this world is designed for you to be the very best like no one ever was. Whether you've taken in all there is to take in from Pokémon, or you're just getting started, this idea is for you.

The theorized land portrays various attractions, interactive show-action-elements, and experiences that will be new and exciting, but yet familiar and nostalgic to those who have grown alongside this world. It all begins with the narrative, what is the story we're trying to tell here? It begins with you, our guest. As a guest in this world, you begin your journey at the Pokécenter to purchase your first Pokéball. From here, you are taken through a never-before-seen experience that allows for you to make the most crucial choice a new trainer could make. What is your staring Pokémon going to be? This decision will then carry on with you as you experience this world. After making the thrilling choice, you are free to explore the world set-forth in front of you, and are encouraged to explore as you so choose to. The major 'e-ticket' experience in this land is the Pokémon gym battle experience. Where guests are thrown into a real Pokémon battle to duel for gym badges, a sign of respect and honor within the world of Pokémon. In this battle, your Pokéball controls all. Your starter and you progress with one another as you climb the ranks of Pokémon trainer. To the guests, this is all thrilling and very real. A highly immersive narrative experience that throws the guests right into the center. To those with background knowledge of the project, you know the secrets and methods used by this team to complete this project.

The magic lies within radio frequency identification (RFID). Within our Pokéball's lies a custom designed PCB that holds within a memory element and RFID tag, allowing for the land to read and write information to within the Pokéball. Beginning at the Pokécenter, the Pokéball is established with a guest ID, a Pokéball ID, and Pokémon ID that is held within the ball. These three guiding pieces of information drive all experiences. Through communication with various readers/writers throughout the land, the guests are able to utilize their Pokéball to evolve their Pokémon, and live alongside them. Through databases held within the backend of this system, the continually reading and writing cycle allows for constant updating. This system also allows for high-volume traffic, as this land should be thought of and designed to hold an immense number of guests. The project displayed here is a passion project and letter of love to the themed entertainment industry, and how the world of Pokémon can achieve great things within it.

With Orlando being the home and heart of themed entertainment, and the close ties that UCF holds with it, this project is a showcase of local industry and knowledge gained while at the university. Though this is a prototype presentation of what a fully conceived themed land could operate around, major engineering problems were faced along the way. Deciding materials that would not interfere and attenuate our communication networks, software development to create a video game similar system from the group-up, and custom fabrication of a PCB that can fit within the Pokéball that fits within the hand of most guests. As the project advanced, the further complexity of questions and roadblocks we found. Considering major factors such as safety, effectiveness, sustainability, and reliability were all guiding lights throughout the completion of this summer design sprint.

2.0 Project Description

The following sections will briefly describe our project, the Pokéball. It will present the motivation and background of the Pokéball. This will also begin to describe the necessary technical aspects of the Pokéball. Our goals and specifications will determine the Pokéball's required configurations. The following section also has requirements that specify both the needs from the engineering aspect but also the marketing aspect. It will also present the separation of necessary components of this project between the group members. This section is an overview of what our project's intentions are.

2.1 Motivation

The world of Pokémon has long been a staple in the entertainment industry. Being evaluated at a rough estimate of $92.1 billion, it is the most valuable franchise in existence. Their products and stories range from card trading to graphic novels, movies, video games, television, and more. Not yet have they breached into the world of themed entertainment, and physical immersive experiences. This project serves as a proof of concept for how the world of Pokémon could theoretically breach into this industry. The background of this project lies in a fictitious themed land corresponding to the world of Pokémon, and serving as a celebration for all this property has to offer. This themed land would feature various attractions that would include a high level of immersion and interactivity from the guests that enjoy them. How they immerse the guests, and request the interaction of the guests is where this project plays in.

2.2 Project Background

The core idea is to create an interactive physical Pokéball that allows for certain low level control from guests throughout this fictitious land, and on the interior of the varying attractions. The varying attractions could consist of a gym battle based experience, where guests get to utilize their own acquired Pokémon on the 'inside' of these Pokéballs. Then proceed to battle with computer controlled Pokémon on the back-end of this attraction to earn "gym badges". The following is a breakdown of how a guest can acquire their first Pokéball, and utilize it throughout this fictitious land.

2.3 Project Goals and Objectives

The guests would enter a merchandise location called the 'Pokécenter', where they can purchase their first Pokéball. On the completed purchase, the operator of the sale would initialize the Pokéball with key information that would be necessary to interact throughout the themed land. The initialization would be completed through RFID means, and would write to an embedded memory element located on a PCB on the interior of the Pokéball. The information held on the Pokéball would consist of a unique guest ID (string of characters designated by the company operating the themed land), Pokémon ID (each Pokémon would be given a set ID designated by the company operating the themed land), and ID equated to the Pokéball itself. These three pieces of data would be initialized on a database privately controlled by the company operating the themed land, along with initializing it to a set memory element embedded on the Pokéball as previously stated.

Once this initialization is complete, and both database and Pokéball are stored with the proper data, the guest is then escorted through the process of gaining their first Pokémon. This would update the database to notify the attractions of what Pokéball ID is being utilized, what guest ID is equated to it, and what Pokémon ID is embedded in this specific Pokéball. The Pokéballs would be limited to one Pokémon per unit. After this, the guest is free to utilize the Pokéball at the differing experiences the land offers.

When entering the gym battle attraction; the guest would follow through a set queue, and enter the specific gym room they are instructed to by the operator. Once inside of the set gym room, an RFID reader would send a power and message signal to the Pokéball to retrieve certain data. The pieces of data that have previously been stored on the Pokéball would be used to initialize a certain battle scenario based on set parameters by our team. Some parameters could include the type of Pokémon that can be used to battle against the Pokémon the guest has selected to battle with. Once the battle has concluded, the guest can be rewarded a certain progression level to evolve their Pokémon of use during the battle. This progression system allows for the land to always be evolving alongside the guests, and the Pokéballs to be the key figure in the guests' interaction with this land.

2.4 Project Narrative

The only other product that features a similar level of interactivity would be the Disney MagicBand system, and the Universal Studios Wizarding World Wand system. The levels of interactivity of these products are utilized in different means, and through different subsystems than what group 15 is portraying. The audience is a broad range of guests from numerous generations. Those who enjoy the world of Pokémon in any facet would find joy and excitement in this advancement of immersion and interactivity to the themed entertainment industry.

This project is meant to act as a proof of concept for how portable handheld RFID based products can be used to increase immersion and interactivity in the themed entertainment industry. This will be a challenging design to accommodate for aspects such as: weight, reliability, environmental consciousness in choice of material, longevity, and ease of use. This goes to show that RFID can also be used in more than one property. The use of property in this project is for academic purposes only, and will not be used in any commercial sense.

Design aspects in weight and reliability come majorly into play with the casing in which our PCB and RFID tag will be embedded in. Systems were needed to be in place to assure that mishaps such as dropping the handheld RFID device won't break, and still operate. Longevity will be crucial from a themed entertainment perspective. Guests who utilize and purchase these items will ideally be traveling from around the world to interact and immerse themselves in the environment in which they can interact with. Ease of use is needed on both front and back ends of this project. Guests of all ages will be interacting on the front end, and operators of the themed environments will be initializing systems while potentially troubleshooting on the back end. These aspects were in mind while approaching the design and build of this project.

2.5 Project Requirements and Specifications

When beginning the Pokéball's design, we analyzed the various aspects to ensure a functional, practical product to be produced. This in depth look into what is required from our design will in turn assist us in other aspects of the production. Determining a clear set of requirements and specifications that the Pokéball must abide by will in turn assist us with an easier process of decision making when it comes to the hardware and software specifications. In order to remain on track, our project has a list of requirements and specifications that must be met. These requirements and specifications were decided first to dictate our project's hardware and software. The table lays out all of the essential specifications, descriptions, and the requirements for each specification. How these requirements and specifications are implemented will be easily seen further in the report. These requirements and specifications will dictate the choices we make on the parts and methods we use to develop the Pokéball. In the table, there are three highlighted requirements and specifications. These three requirements will be demonstrated at the year end project presentation. They are the three requirements we have chosen because its quantitative data will be easily demonstrable at the presentation. The three requirements we have chosen are the activation time, maximum Pokémon connection, and progression. The activation time will show that under 30 seconds a new Pokéball will be able to be activated for a new user. The maximum Pokémon connection will show that each individual Pokéball unit will only allow for a singular Pokémon to be connected to the device. It will reject any attempt to connect a new Pokémon. The progression will highlight one of our features for the Pokéball. It will show that under 3 minutes the Pokéball can battle another Pokéball unit and determine a winner.

Requirements and Specifications		
Active Time	The Pokéball shall be able to run for a specified amount of time	$\geq 12\ hours$
Wirelessly Transmit Data	The Pokéball shall be able to transmit data wirelessly within a specified amount of distance	$\geq 25\ feet$
Weight	The Pokéball shall be lightweight and within the specified weight parameters	$\geq 3\ pounds$
Dimension	The Pokéball shall be portable and within the specified size parameters	$\geq 5\ inches\ in\ diameter$
Activation Time	The Pokéball shall be able to activate within a specified amount of time	$\geq 30\ seconds$
Maximum Pokémon Connection	The Pokéball shall only be able to connect to a specified number Pokémon	$= 1\ Pokémon$
Progression	The Pokéball shall be able to determine a winner between two units within the specified amount of time	$\geq 3\ minutes$
Unit Connection	The Pokéball shall only be able to connect to a specified number of units	$= 1\ Pokéball$
Accuracy	The Pokéball shall be able to access a specified amount of data from the Pokéball database	$\approx 98\%$
Unique Identifiers	The Pokéball shall be able to be identified with a specified number of identifiers	$= 3\ unique\ identifiers$
Durability	The Pokéball shall be able to withstand dropping from a specified amount of distance above the ground	$\geq 5\ feet$

Table 1: Engineering Specifications

2.5.1 Updates to Requirements and Specifications

When implementing the design for the Pokeball, we were able to abide by a majority of the requirements and specifications we set during the design portion. There were some that we adjusted to follow a more accurate representation of the product. Initially we required our activation time to be under thirty seconds. Once we began implementing the design, we discovered that there were more aspects to activation than initially assumed. The final requirement of activation required the user to register and login to the website and activate their new Pokeball. Accounting for various factors, we set the new activation time to be under three minutes. This change allows the user and program to activate and update within an appropriate amount of time. Another requirement we adjusted was the Pokemon connection. The Pokemon connection requirement was specified to have only one Pokemon to be connected. This specification remained consistent through our implementation but we added a time requirement. Under five minutes, the Pokeball shall be able to detect a Pokemon and then store that Pokemon and as it is stored, it will not allow another to store. We specified five minutes because within that time the program will be able to accurately detect and properly store the Pokemon. The final specification that was altered was the battle sequence requirement. Initially we specified that the battle sequence shall not take any longer than three minutes. As we were creating the program and tested it, we saw it took a user, on average, about one minute to complete the battle. Within these two minutes, the user will be able to complete the battle and then check the leaderboard to see it update the scores. These updated requirements and specifications used for the demonstration of the project can be seen in the table below.

Requirements and Specifications		
Activation Time	The User shall be able to register, login, and activate a new Pokeball within a specified amount of time	$< 3\ minutes$
Pokemon Connection	The Pokeball shall be able to detect and capture a single Pokemon within specified amount of time	$< 5\ minutes$
Battle Sequence	The User shall be able to connect their Pokeball to the Battle Sequence in the program and the program will be able to update the scores within a specified amount of time	$< 2\ minutes$

Table 2: Updated Engineering Specifications

2.6 Project Constraints

There are many constraints for this project. The major constraints have been highlighted below. These constraints will begin us on the path to determine the design and more precise constraints for the Pokéball later. An in depth analysis of all constraints will be presented later in the report. This is to show the main, preliminary concerns of the design and what has major influence on the design and development of the Pokéball.

- Design
    - The Pokéball's design must be relatively light, portable and reliable. It must be lightweight to ensure portability and be fairly durable as it is susceptible to being dropped a few times due to our intended audience. Also as there will be a PCB and RFID tag embedded inside the Pokéball, it must be able to withstand the possible droppings as well as protect the hardware and continue to be functioning.
- Price
    - As the Pokéball is not sponsored, price is one of our major constraints. We have also chosen to reduce the cost of the Pokéball due to its intended use and audience. Overall the Pokéball is low cost and does not require many parts to design or development. As it will be described further in the report, our chosen tools and parts are free or low cost. Any cost of this project will be self funded and split between the group equally.
- Time
    - The project's time frame is spread amongst two semesters. It will require a lot of individual and group time to ensure the Pokéball's success. The class has set specified deadlines for each aspect of the project and the group has set its own specified deadlines for the Pokéball to ensure it will be complete on time. The deadlines prioritize the various requirements when necessary and assist in focussing on the important aspects of the project first. The group will focus on and prioritize the Pokéball's design and development in our individual schedules.

2.7 House of Quality

The house of quality is displayed in Figure 1 below. This is an essential tool for demonstrating requirements for our Pokéball. It easily demonstrates the combination of the engineering requirements as well as the marketing requirements for the project. The figure below will show that cost is an important requirement for both marketing and engineering requirements. All of the other requirements are negatively correlated with cost. The engineering requirements of the Pokéball are efficiency, weight, quality, dimensions, and cost. The marketing requirements are durability, usability, portability, cost, and long lasting. The house of quality demonstrates the correlations, relationships, and direction of improvements for all the requirements mentioned above. Table 2 will serve as a legend to demonstrate the positive, negative, strong, and weak aspects of each requirements connection. This legend will assist in understanding the house of quality further.

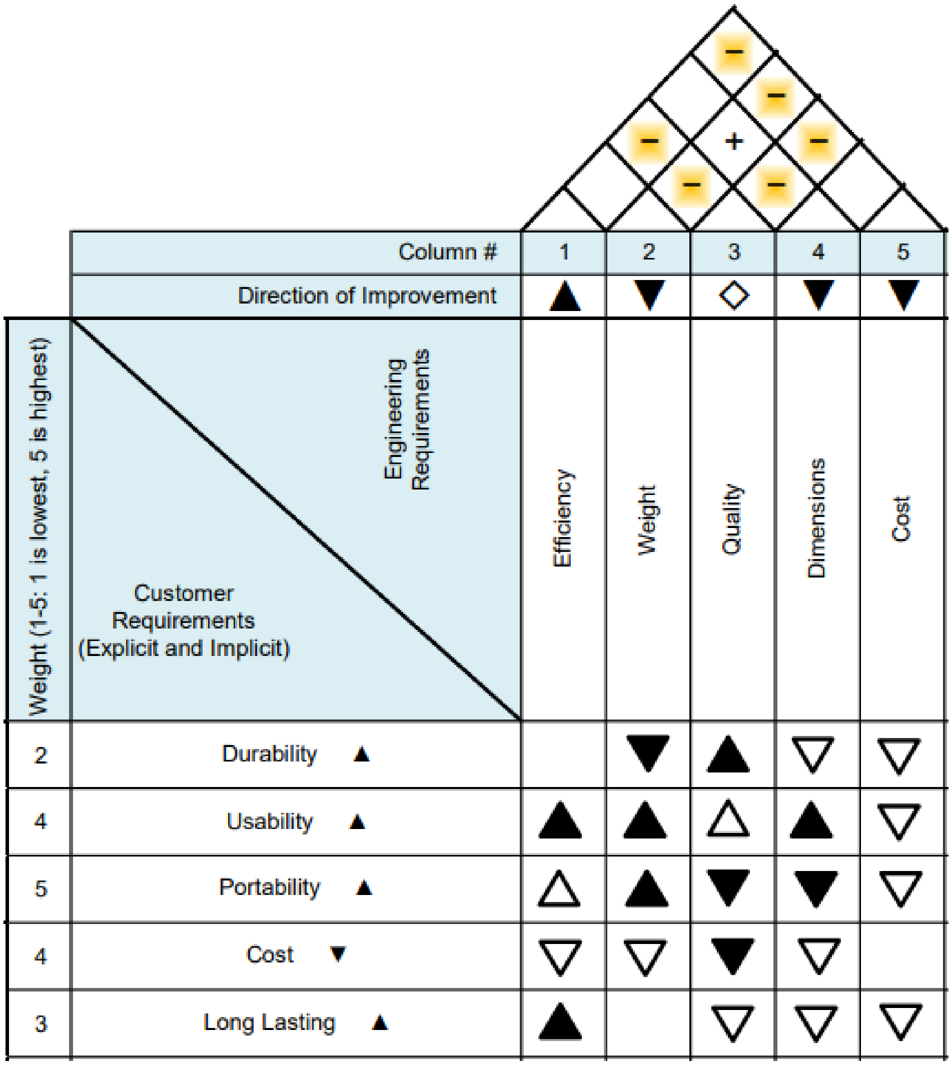

Figure 1: House of Quality Specifications

Legend

Correlations	
Positive	+
Negative	−
No Correlation	

Relationships	
Strong Positive	▲
Positive	△
Negative	▽
Strong Negative	▼

Direction of Improvement	
Maximize	▲
Target	◇
Minimize	▼

Table 2: House of Quality Legend

2.8 Project Prototype Illustration

Figure 2: Pokéball Interacting with RFID Reading Device

The figure above illustrates the project prototype. It provides a simple illustration of the connection between the RFID tag with the Pokéball. The figure established that our project's goal is to maintain a connection between the Pokéball and the RFID reader.

2.9 Project Block Diagrams

The legend below describes the distribution of labor for each group member by assigning each a specific color. It also has a section dedicated to determining the block status. This legend will be used to identify and differentiate between the various block diagrams within this section.

Legend

Assigned Members
(Electrical Engineers)
Trey and Darren

(Computer Engineers)
Kalindi and Rebecca

Block Status
To be acquired (TBA)
Acquired (A)
Research (R)
Design (D)
Prototype (P)
Completed (C)

2.9.1 RFID Device Design Block Diagram

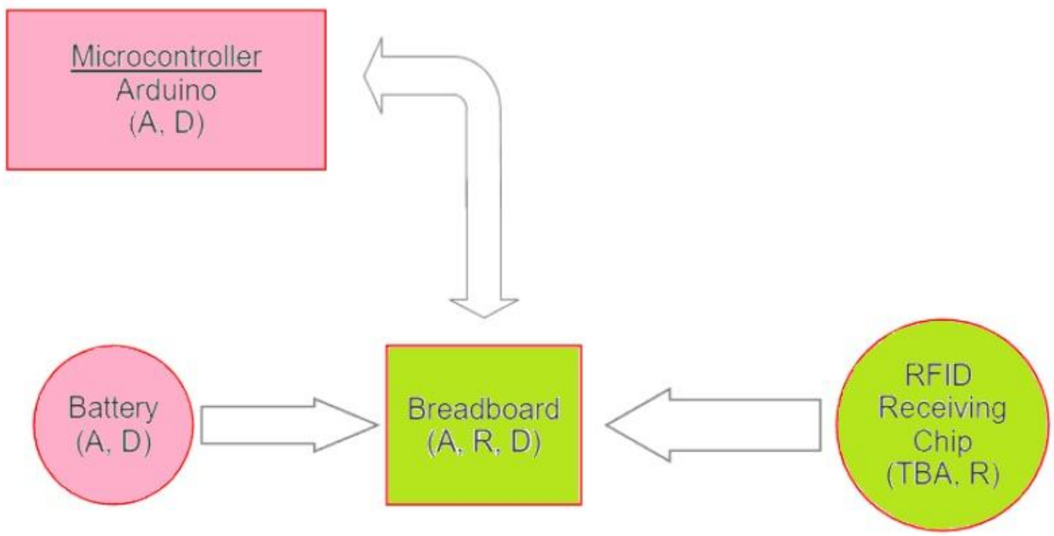

Figure 3: RFID Device Design

The block diagram above illustrates the connection of the RFID Device Design. It also presents a visual of who will be working on each aspect and the block status of each aspect. The legend will assist in showing that the electrical engineers of the group will be in charge of this section.

2.9.2 Pokéball Design Block Diagram

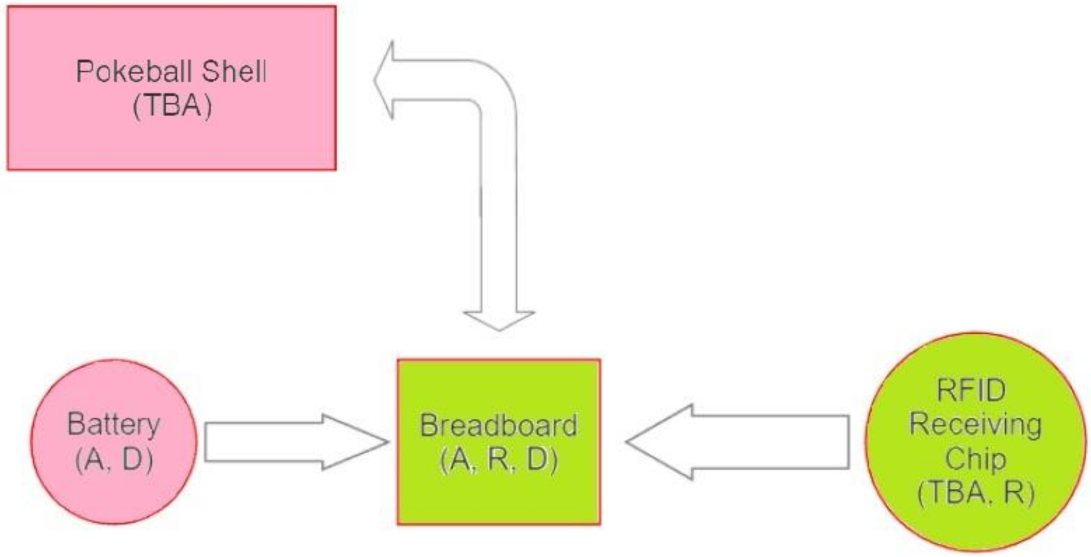

Figure 4: Pokéball Design

The block diagram above illustrates the design of the Pokéball. It shows the two way connection between the Pokéball shell and the breadboard. The diagram also shows the relationship between the breadboard and the battery and the RFID receiving chip. In addition to the diagram demonstrating the relationship in the Pokéball design, it also shows the status of each part in the blocks that can be deciphered using the legend. The diagram also shows the electrical engineers being responsible for this aspect of the Pokéball and who specifically will be in charge of each part.

Sections 2.9.1 and 2.9.2 represent the physical hardware components required for communication between the guest and system. The Pokéball will have a RFID chip implemented so that it is able to connect with the system. In order to make the RFID work it requires power, a battery in this case so that it is portable, and a breadboard to connect the battery and RFID chip together.

The RFID device works similarly in that it requires a breadboard to connect the RFID chip to a power source. But it also requires a microcontroller so that it is able to interact with the Pokéball. It needs to be able to read the Pokéballs ID and determine what action is performed next. The actions the microcontroller can provide are battle gym access, store registered guests, and communicate with multiple Pokéballs as well.

2.9.3 Project Software Block Diagram

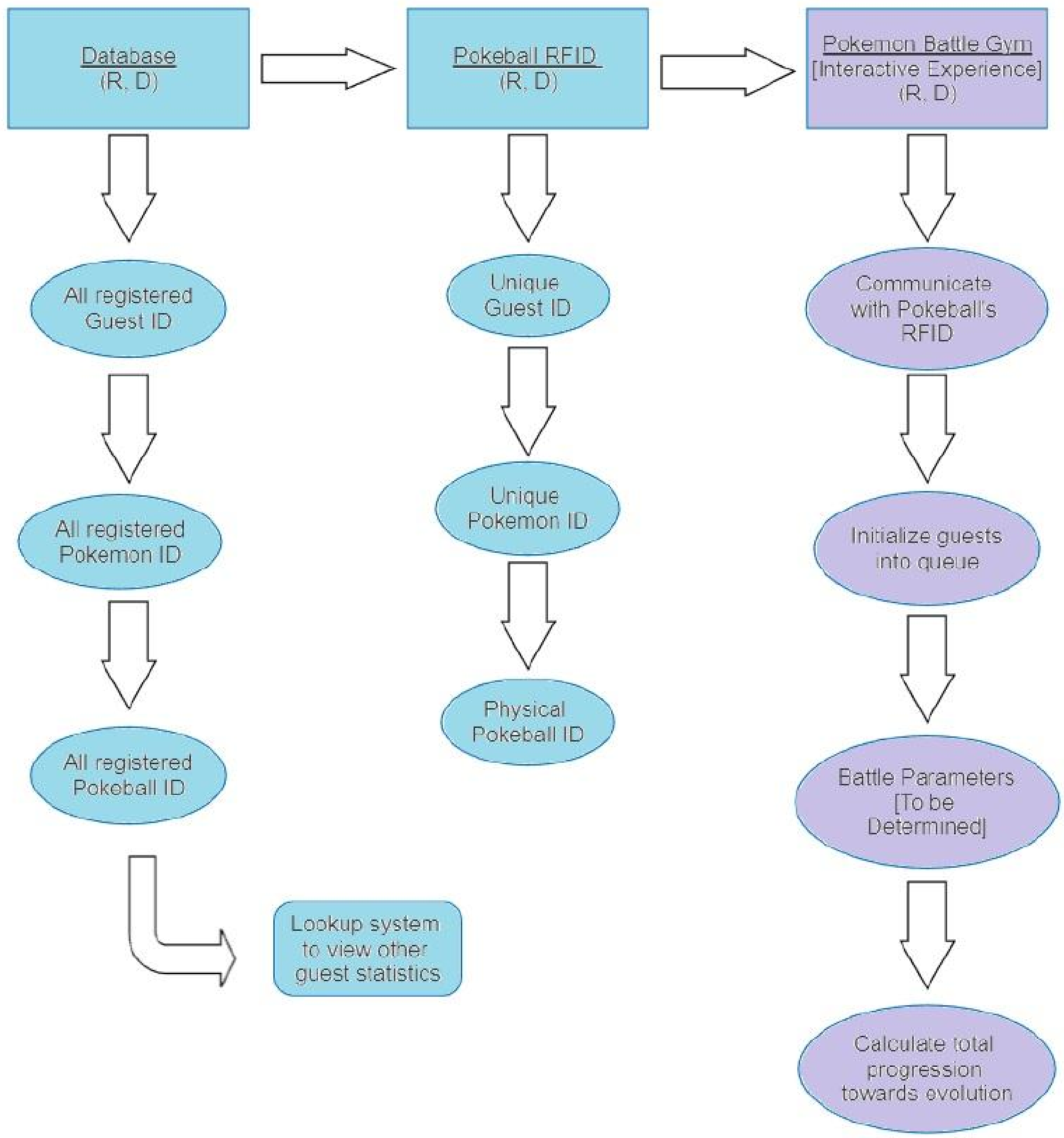

Figure 5: Software Design

The project software block diagram illustrates how each physical component interacts with each other. In this case, the database needs to store every guest who signs up for a Pokéball. When registered, they will receive a unique guest ID along with a specific Pokémon that they decided upon. A Pokéball will be given to the guest so that their specific Pokémon ID is registered together with the Pokéball.

2.10 Updates in Senior Design 2

The following subsections provide in-depth reports on how the project progressed, and grew through the finality of Senior Design 2. The updates include changes to our overall project description, motivation, goals, and objectives. How those items influenced the narrative and overall guest experience we wished to create. Once the narrative and experience were refined based on our goals and objectives, we moved into shifting our overall block diagrams as new issues and requirements made themselves apparent.

2.10.1 Description and Motivation Changes

One major change to the overall description of the project is found in its name: *Project Apricorn*. The namesake of this project derives from the fictional Apricorn fruit found in the world of Pokémon. In the original franchise, the Apricorn fruit was hollowed out and turned into the first ever Pokéball. The group decided upon this namesake being the perfect fit for our project. This was the overall only update to the motivation and description that were changed in the runtime of Senior Design 2.

2.10.2 Updated Goals and Objectives

As previously stated, as the description and motivation were updated, our goals and objectives all required updating. From the above *Table 1: Engineering Specifications* showcases what our intentions were heading into the design process in Senior Design 1. As the project progressed, and more hands-on work began in Senior Design 2, the group decided to stick with these requirements. These specifications were found to be accurate and narrowed down, yet robust enough to allow our project to be completed properly.

2.10.3 Updated Narrative and Experience

As the project progressed, it became apparently more difficult to describe to those unfamiliar with themed entertainment terminology what it was exactly that we searched to complete through this project. This convinced us as a group to graphically showcase what it exactly was we were striving for in Project Apricorn. This was completed through creating a map of the proposed themed land that Project Apricorn lives within. The experience and narrative remained otherwise stagnant, but the addition of a visual cue provided the necessary means to adequately describe Apricorn's goals.

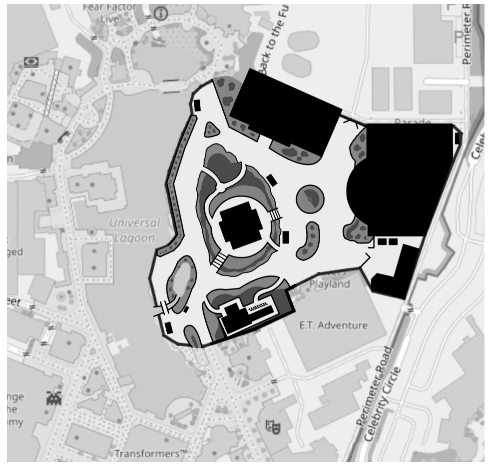

Figure 6. Project Apricorn themed land diagram

2.10.5 Updated Project Block Diagrams

The project as it was completed consisted of creating a new block diagram, the final overall block diagram. The overall project can be broken down into two major diagrams. The two diagrams are the overall, as well as the hardware specific diagram that showcases how the Pokéball hardware works as well. These two diagrams were the provided final diagrams.

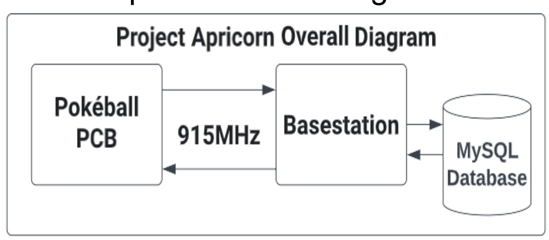

Figure 7. Apricorn Overall Block Diagram

The overall diagram consists of four major elements. The PCB, Base Station, Database, and User Experience. The PCB diagram below:

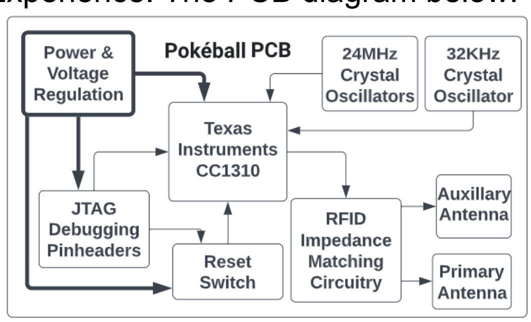

Figure 8. Apricorn PCB Block Diagram

3.0 Research Related to Project Definition

The Pokéball is the first of its kind. Although there are different products in the market that resemble various aspects of our project, none fully compares to our project. In the following sections of our report, our research will fully describe the existing projects currently in the market that are similar to our project. This section will also report on the necessary software tools for this project. It will define the tools we determined essential to this project and the reasoning behind that decision. This section serves to provide the research pertaining to the non-technical portion of the project definition.

3.1 Existing Products

The Pokéball fills the void in the market for Pokémon enthusiasts. Although there are various types of Pokéballs in the market, they are all meant solely for display and have no real function associated. Our Pokéball is intended to be connected with the experience of a Pokémon World. There are other products in the market currently that resemble this function but none that match the format we are trying to achieve as well as the Pokémon aspect. The two products that are most closely related to what we are trying to achieve would be Disney's MagicBands and Universal's Interactive Wands. This project analyzed how our Pokéball is different from the other two products in terms of intention and design.

3.1.1 Disney MagicBand

Disney's MagicBand intention is to significantly reduce the amount of items individuals need to carry. Prior to the MagicBand creation, park goers would have to carry an ID, payment card, hotel key, fastpass tickets, park tickets, and various other things. It utilized RFID chips in the same way we intend to. Although MagicBand achieved its goal of minimizing the necessary items to carry, it has become less desirable due to the growth of integrating the same cards into one's smartphone. In figure 6 below, the comparison between the MagicBand and smartphone connection can be seen. In this day and age, everyone is guaranteed to have their smartphone with them especially when going to a theme park. It is easier to simply carry a smartphone that has the capabilities to hold all of the items the MagicBand can hold virtually. Also, in order to pay with the MagicBand, you must first tap and then enter a passcode whereas you must simply tap with a smartphone.

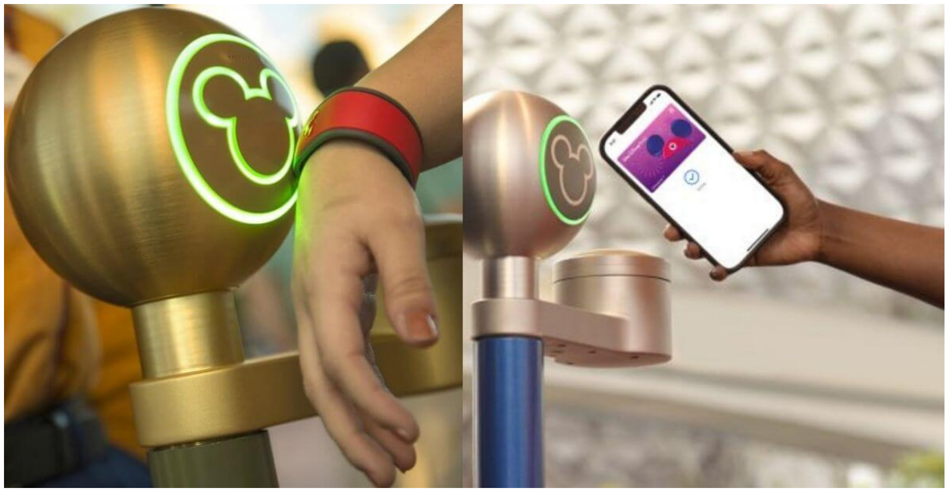

Figure 9: MagicBand vs Smartphone Connection

The MagicBands do not necessarily have a core connection to the Disney characters the way the Pokéball does. Figure 7 below shows the way MagicBands integrate various of their characters into the product and shows how irrelevant the design of the MagicBand are. Any variation of designs can be used for the bands to curate towards their current product whereas the Pokéball is a constant design with a strong familiarity to those who enjoy the Pokémon franchise. The Pokéball has a direct connection to the Pokémon world where it is recognizable by many. Disney's MagicBand and our Pokéball do not have the same intention for use. Our Pokéball is intended to be an integral part of our Pokémon world not an add-on as the MagicBands are. Although the MagicBands utilizes the same technology our Pokéball is based around, the MagicBand is intended for different uses and therefore cannot be fully comparable to our Pokéball product.

Figure 10: Variety of Disney's MagicBands

3.1.2 Universal Interactive Wands

Universal's Interactive Wands are representative of the wands from the Harry Potter series. The interactive wands have a strong connection to the Harry Potter series. Each of the interactive wands comes with a map of the replica Diagon Alley and Hogsmeade created in Universal, two locations in the Harry Potter series. The map and a photo of one of the wands is shown in figure 8 below. The variety of wands offered are replicas of the wands described in the books and shown in the films. There are also a set of interactive wands known as unclaimed wands that have a unique design not related to a character. This direct connection to the franchise connects the user with the product resulting in higher enjoyment.

Figure 11: An Interactive Wand from Universal

The variety of interactive wands and integration of the locations that are within the Harry Potter franchise excites Harry Potter fans as they feel closer to the series. The Pokémon franchise is substantially larger than that of Harry Potter's in terms of gross revenue and global following. In figure 9 below, we can see that, as of January 2021, the Pokémon franchise is the most valuable franchise and the Harry Potter franchise is tenth valuable. Even in mid to late 2022, this fact still holds true, the Pokémon franchise continues to be the most valuable. According to this data, a similar if not better reaction should be expected for our Pokéball along with the associated Pokémon world.

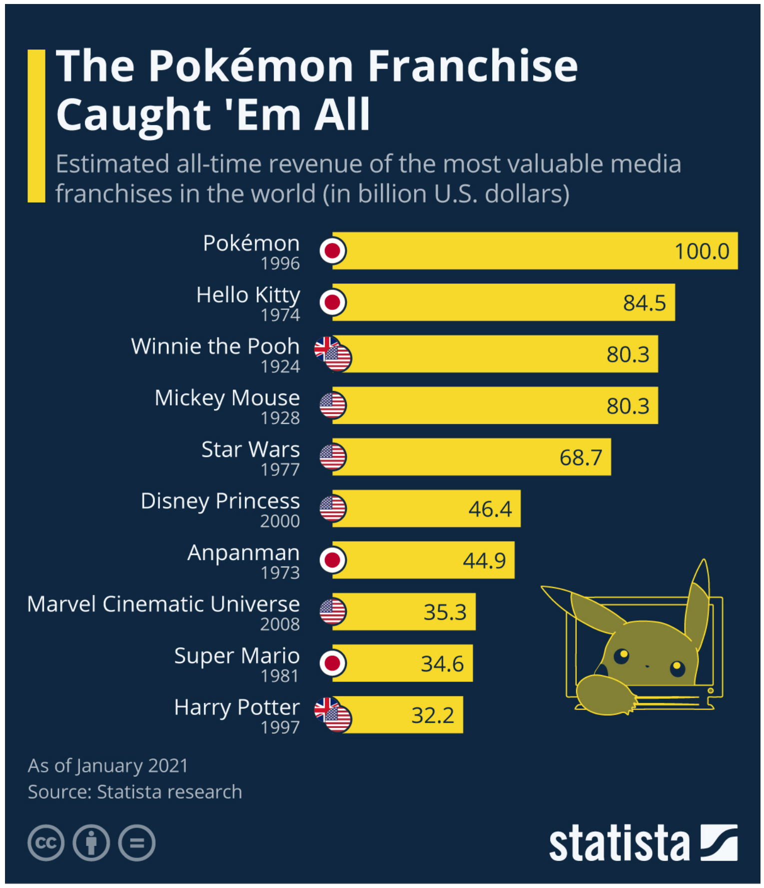

Figure 12: Chart of Top 10 Media Franchises

In order to use the interactive wands, park goers must locate the various medallions on the map and correspond it to the location at the park. These medallions are on the ground that have an arrow pointed towards the location of where the park goer should cast the spell. In the center of the medallion, there is an image of the motion that the park goers must perform for their spell to work. The image of the motion is surrounded by the words that the park goer should say to cast the spell. A photo of the medallion can be seen below in figure 10. The spell names are directly correlated to the Harry Potter series and are meant to replicate in the window where the spell is being cast. The window is equipped with infrared cameras designed to replicate the spell motion and cleverly hidden in the window to not diminish the experience. Due to the cameras' positions, the range of motion is limited so the park goer must be relatively close to the window and perform the spell in a relatively small motion. As the camera is picking up the reflection from the tip of the wand, there is no reason to speak the spell but it does make the park goers feel further integrated in the series.

Figure 13: A Medallion at Universal

In comparison with our Pokéball project, Universal's interactive wands have a similar aesthetic appeal. It allows the users to feel connected to the franchise they so passionately follow. The success of the interactive wands shows that our idea for Pokéballs should be just as successful considering the substantially larger following. The interactive wands however do not use the same system we plan to implement. We intend to utilize the RFID chips similar to that of Disney's MagicBands. We also intend to implement an additional feature of interacting between two Pokéballs that neither the MagicBands nor the interactive wands have.

3.2 Existing Applications

RFID

RFID is used in a variety of applications in all types of fields. It is a cost effective way to automate certain tasks that otherwise would have taken lots of hours and also not as accurate.

- Retail - RFID is a cost effective way to track inventory. Items can be logged into the system to determine quantity and how long products have been in the store.
- Toll Roads - RFID is used to automatically determine what account the vehicle belongs to so they can be charged.
- Asset Management - RFID is used in hotels so guests can have individual programmable cards that only have access to the room they paid for.
- Passports - some passwords have embedded chips that store biometric information.

3.3 Software Tools

Our project requires various software tools to assist in designing and implementing the Pokéball. Our group consists of four members located in various locations. It is vital to the success of the Pokéball to have reliable and predetermined software tools to ensure we do not waste any time or issues relating to our chosen software. Among the software tools, the most important are related to communication and development. These types of tools will assist in the design and production of the Pokéball. All of the tools we have decided on utilizing are free to use. None of the tools were previously purchased by any member. This was also an important aspect when deciding the software we wanted to utilize as it contributes to keeping our Pokéball affordable. None of the software tools described below were needed to be included in our budget. Although there may be other tools individual members may use to implement the Pokéball, these are the centralized tools that will be the main tools for the Pokéball.

3.3.1 Communication

Communication is essential to any team working on a project. As there are a total of four members in our group working from different locations, communication is very important to the success of our project. In order to accomplish the tasks to work towards our Pokéball, we must use a dedicated communication tool. There are many choices to consider for communication. We must determine which tool is best suited for our group. There are many aspects to consider which will be described below. Ultimately we decided to use Discord and Google Drive.

3.3.1.1 Discord

For general communication, we choose Discord to be our main source of contact. Discord allows for private servers and the creator of the server must invite members to join said server. Another benefit to using Discord is the cost. It is free and allows us to communicate with no financial commitment. Unlike other options, Discord is available on all platforms and allows for easy communication through phone, laptop, and web. All of the team members have Discord installed originally and did not have to allot additional storage space to a new app. As we all already owned the app, we were familiar with Discord’s environment. Discord also has a voice chat capability to allow us to easily conduct team meetings within the app. When we begin developing the Pokéball, if any individual runs into an issue that requires assistance from another team member, Discord also has screen share capabilities. It allows for all of the members to view the screen of another’s without being together in person. Due to all of Discord’s capabilities, Discord was the best platform to conduct our communication.

3.3.1.2 Google Drive

Google Drive is the most efficient platform to share files. It is very simple to create a shared folder in which all necessary files pertaining to our project can exist. As this is a two semester project, we implemented a shared folder that can be accessed by all members at any time during the duration of this project. The shared folder allows us to store our necessary files for the project and reduces our concern of an individual possibly losing a file. Within Google Drive, the Google Docs is an additional online word processing software. Google Docs allows us to seamlessly write our reports together online. This way we do not have to share documents and merge information, but rather it is already all in one location to begin with. Google Drive has various associated software that could assist us in our project. It is very simple to use the embedded table and drawing tools to create the variety of tables and figures needed for our report. Google Drive is all accessible on all platforms and is easier to use and understand. As all of us are already familiar with Google Drive, this was the most logical software to utilize.

3.3.2 Developmental

After deciding the design of the Pokéball based on the requirements, our next step is to do the necessary planning for the Pokéball and the programming associated. The planning portion of this project is the beginning of implementing the Pokéball. There are various tools that can assist in the programming and designing of the Pokéball. We have chosen two tools that will primarily assist with this as it is the most familiar and easiest to use amongst the other options. Our choice for programming is Github and designing is Fusion 360.

3.3.2.1 Github

Github is an online service for software development. It also has a variety of features embedded in the program to assist in the developing the code. Since our group has four people, this program will help us to code together. Similar to the reasons for choosing Google Drive, Github's main benefit is the possibility of use by various users in various locations. Since coding for the Pokéball does not necessarily need to be done together, this assists in sharing the code without constant transfer. This eliminates the risk of misnamed variables, misunderstanding explanation of code, and other risks involved with transfers. Github will allow each of us to contribute to developing the Pokéball simultaneously. Some of the features included with Github are bug tracking and task management. This will assist in resolving the bugs within Github. Also, it will be helpful in assigning tasks within Github to individuals to ensure we accomplish our tasks in a timely manner and easily ask for assistance when needed. Although this will not be the final destination to execute our code, it will be a great place to plan our code and ensure it is ready for the final product.

3.3.2.2 Fusion 360

Fusion 360 was developed by Autodesk for the intentions of computer-aided design, computer-aided manufacturing, and printed circuit board design. It is a paid subscription available on Windows and Mac. Fusion 360 also currently has a campaign allowing students to utilize the software for free allowing us to use the software without adding into our budget estimate. This is very important to us as the program is approximately $396 and would be necessary for our project. This software assists in developing our Pokéball components. In further sections, it will be described in depth how exactly we utilized this software for the Pokéball. As a brief overview, we designed the exact specifications of the physical Pokéball in order to then properly have a 3D printed model. It will also be useful in designing the PCB for the Pokéball. Fusion 360 is familiar amongst some of the members of the group allowing for the software to be very useful for developing the Pokéball. The group will not need to waste valuable time familiarizing themselves with a different software. Fusion 360 is extremely essential to developing the Pokéball.

3.3.2.3 Arduino IDE

The Arduino IDE is a text editor for writing code that is used to program a variety of microprocessors and controllers. These microcontrollers are programmed using C or C++ programming languages. It is a free downloadable software online that will not add to our budget estimate like our other software tools. This tool is a specific necessary program to one of our chosen components, the Arduino UNO board. We further describe the use of Arduino IDE and its use with our chosen microcontroller in one of the coming sections. The Arduino IDE will allow us to program the Arduino UNO board for the Pokéball. Similar to Fusion 360, this is a vital program that we must utilize for our design. The Arduino IDE is also familiar amongst some of our members allowing for easy implementation with the Pokéball.

3.3.2.4 Code Composer Studio

Code Composer Studio is an IDE developed by Texas Instruments to develop applications for Texas Instruments processors. The MSP430 microcontroller is configured using this program. This is a necessary component to the Pokéball. We further describe the use of this program in a future section of the report. The use of this software is dependent on the microcontroller we decide to use for the final Pokéball. As Code Composer Studio is used to design the MSP430 microcontroller, Arduino IDE is used to design the Arduino UNO board. So whichever microcontroller benefits the Pokéball more will be utilized more, it will be defined in further sections.

3.4 Updated Research and Tools

All softwares was utilized to some extent throughout both Senior Design 1 and Senior Design 2. In addition, the software KiCad was utilized to create the schematic and PCB of the handheld assembly. The Texas Instruments SmartRF Studio software was also utilized when completing the testing for the RF communications between the handheld assembly and Base Station. To create the 3D-Print files of our Pokéball, the software Ultimaker Cura was utilized. These were necessary to construct “slices” of each layer of filament that was needed to make the print into a reality.

4.0 Standards and Constraints

This section examines the related standards and realistic constraints for the Pokéball. Standards are established requirements determined by professionals in the industry which are applied to repeated use of rules. The standards defined below are determined by professional associations that we have determined to be relevant to our project. Constraints have been decided by us with the Pokéball's design in mind. We have examined various categories of constraints that are applicable to our design. These constraints include functional, safety, quality, manufacturing, economic and time, and aesthetic constraints. These standards and constraints assist our Pokéball to follow standards decided by professionals and have realistic views regarding the constraints.

4.1 Related Standards

The International Organization of Standardization (ISO) is an international standards development organization. They are a well established organization with international recognition. The ISO has 167 members, each representing their countries. They develop and publish standards regarding all fields except electrical and electronic engineering. Their standards are reviewed every five years to ensure the relevance of the standard in current time. After five years, the standard can be updated or withdrawn. The ISO is an organization that develops their standards through a rigorous process and regular review system.

The International Electrotechnical Commision (IEC) is an international standards development organization. IEC covers the areas of standardizing fields that the ISO does not. Their standards are all in relation to electrical, electronic, and related technologies. The organization consists of 89 participating countries. Members of the IEC are professionals in the industry with a combination of those working in the public and private sector. They are a highly credible organization in regards to standards development.

The Institute of Electrical and Electronics Engineers (IEEE) is a professional association that is dedicated to the betterment of humanity through technology. They have established a lot of credibility amongst the industry. The IEEE Standards Association is a unit within the IEEE. Its purpose is to develop standards that are recognized globally in a variety of industries. They establish these standards through a sense of balance, openness, fair procedures and consensus.

All of the organizations above collaborate closely together to create standards benefiting the electrical and electronic industry. The following standards have been developed by the organizations described above. They are the standards we believe are most related to our project.

4.1.1 Quality Management System Standard

ISO/IEC/IEEE 90003 is a software engineering guideline for the application of ISO 9001:2015 to computer software developed by the International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), and the Institute of Electrical Engineers (IEEE). The standard defines the applications of the quality management standard defined by the ISO to computer softwares. It explains the process of developing a product of quality and all its vital steps involved. In addition to the development components of a product, this standard involves the necessary business, support, and operational aspects that contribute equally to the product. Below is a figure that is a schematic representation of the standard procedure of any process. At every step in a process, there are possible controls and checkpoints to monitor and measure performance. The figure is a broad overview of the standard's intention without its application to computer softwares.

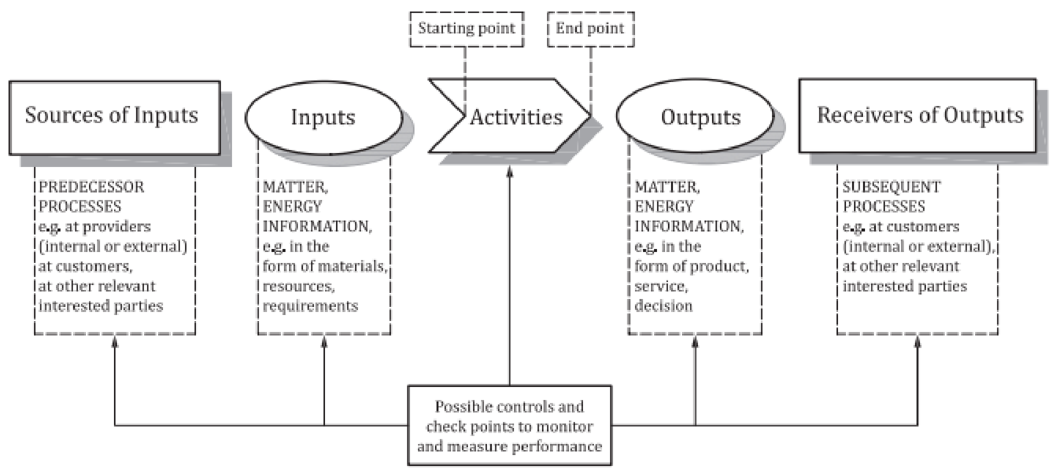

Figure 14: Schematic representation of the elements of a single process from ISO/IEC/IEEE 90003

This schematic design shown above represents the various steps of a single process. It shows the preliminary step as recognizing the relevant interested parties. This is a vital stage in the process as it will assist in deciding the customer requirements and begin the business aspect of the process. This will determine where the customer focus is and assist in establishing the quality policy. At the input stage, the objectives and risks can be analyzed. This is also where the planning of the product is accomplished. The activities refers to the development of the product and the processes associated with that. Outputs are the finished product after the testing phase is complete. Receivers of outputs refers to the performance evaluation and improvement stage of the process. This is where the sources of inputs process research is tested. The standard defines this process as a continual improvement. Customer satisfaction and management reviews are analyzed here.

The following figure is a representation of the Plan-Do-Check-Act (PDCA) cycle as defined by the ISO. It presents the relationships between all of the stages of a product's life cycle within a quality management system. The figure helps to visualize the importance of customer requirements and the results of utilizing the quality management system. This system is utilized by every product during development but the standard sets a verified system of procedures. Each action in the cycle is closely related to one another. Also noticeably, it is a cycle that promotes continual improvement as the standard explicitly describes it as a vital part of the process.

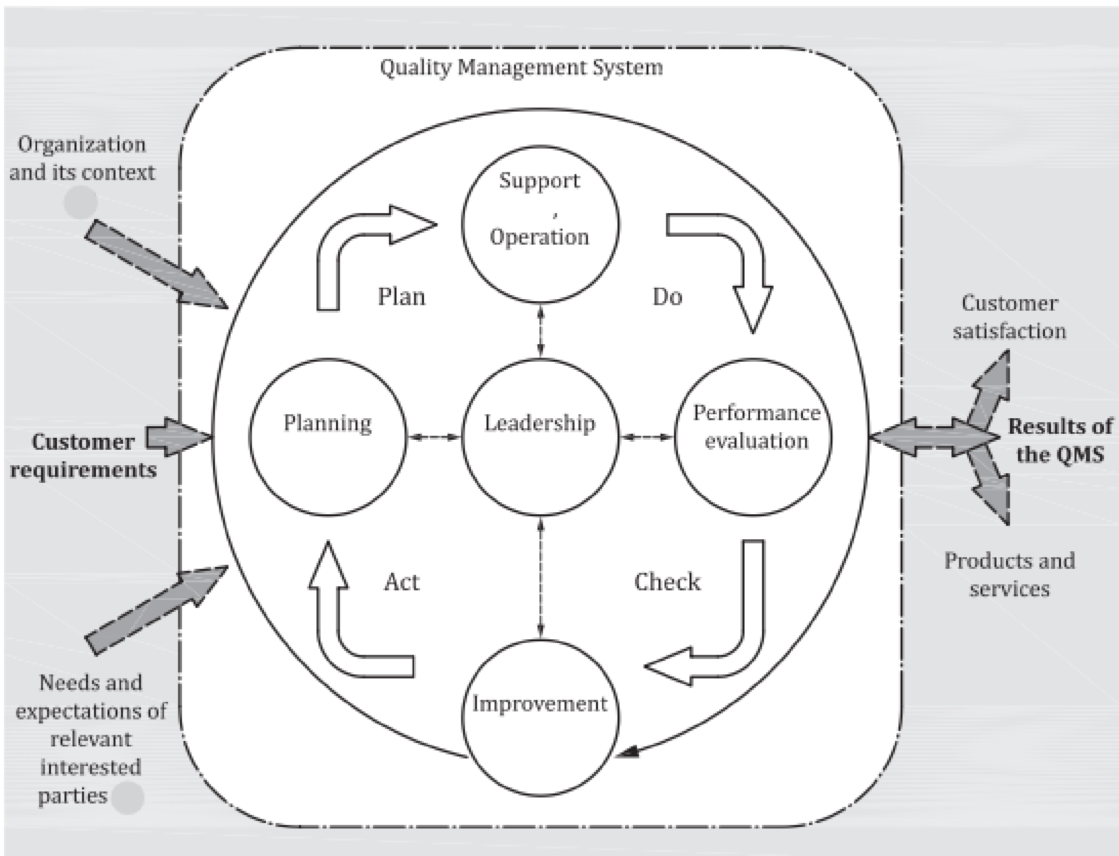

Figure 15: PDCA cycle from ISO/IEC/IEEE 90003

The core aspect of a quality management system is a properly designated leadership system. By having a well defined core leadership, all other aspects of the system are influenced. There is a two way relationship between the other integral steps to the design and production of a product. The standard defines there is a lot of benefit in defining a customer focus and actively utilizing that when designing and developing the product. The separation of roles and responsibilities should be assigned to those who have the most experience in the specified area.

In association with the planning portion of the project, it is important to address the risks involved with developing the product. When software is involved, there are certain safety and security issues that must be addressed. A plan should be created that defines the actions that will address these risks and the effectiveness of the actions.

Support is necessary for keeping on track for all aspects of the project. By determining designated people that are integral for effective implementation of the quality management system, the software will be better executed. There is a lot of benefit in separating individuals based on their experience and assigning them the tasks that are best suited for their skills. This will show measurable success in project design and development.

The customer should be heavily involved in or their desires should be immensely recognized throughout the design and development of any project. The customer focus should be determined early in the project design but then their need for online help, user manuals, complaints, etc should be taken into account. After development, the product should be analyzed again to see if it met all of the customer's requirements previously determined. Once the product begins to be used by the customer, the quality management system deems that the customers' perceptions and satisfaction towards the product should be analyzed and recorded for future reference. This will assist in evaluating the product's success and future improvements that could be made to the product. The quality management system shows that the product will undergo continual improvement reviews to ensure the product will be exceeding the customer's expectation.

4.1.2 Software Testing Standard

ISO/IEC/IEEE 29119-1 is a software and systems engineering global standard published in collaboration by the International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), and the Institute of Electrical Engineers (IEEE). The standard defines the purpose, process, and strategies for testing. The following figure presents the standards proposed methods of testing to verify and validate the "test item". The term "test item" is defined by the standard as "a work product that is an object of testing" (ISO/IEC/IEEE 29119-1 1). The test item could be a form of hardware, software, or related documentation. The figure will show that the executable test items can undergo dynamic testing while non-executables, i.e. related documentation, can undergo static testing. Dynamic testing involves the test item's software. It can only occur when executable code can be run. Testing requires one to execute code and run test cases for the software. The intention of testing is to run exhaustive tests that will likely present issues in the software. This method of testing is utilized to detect any possible defects in the software before the software is finalized.

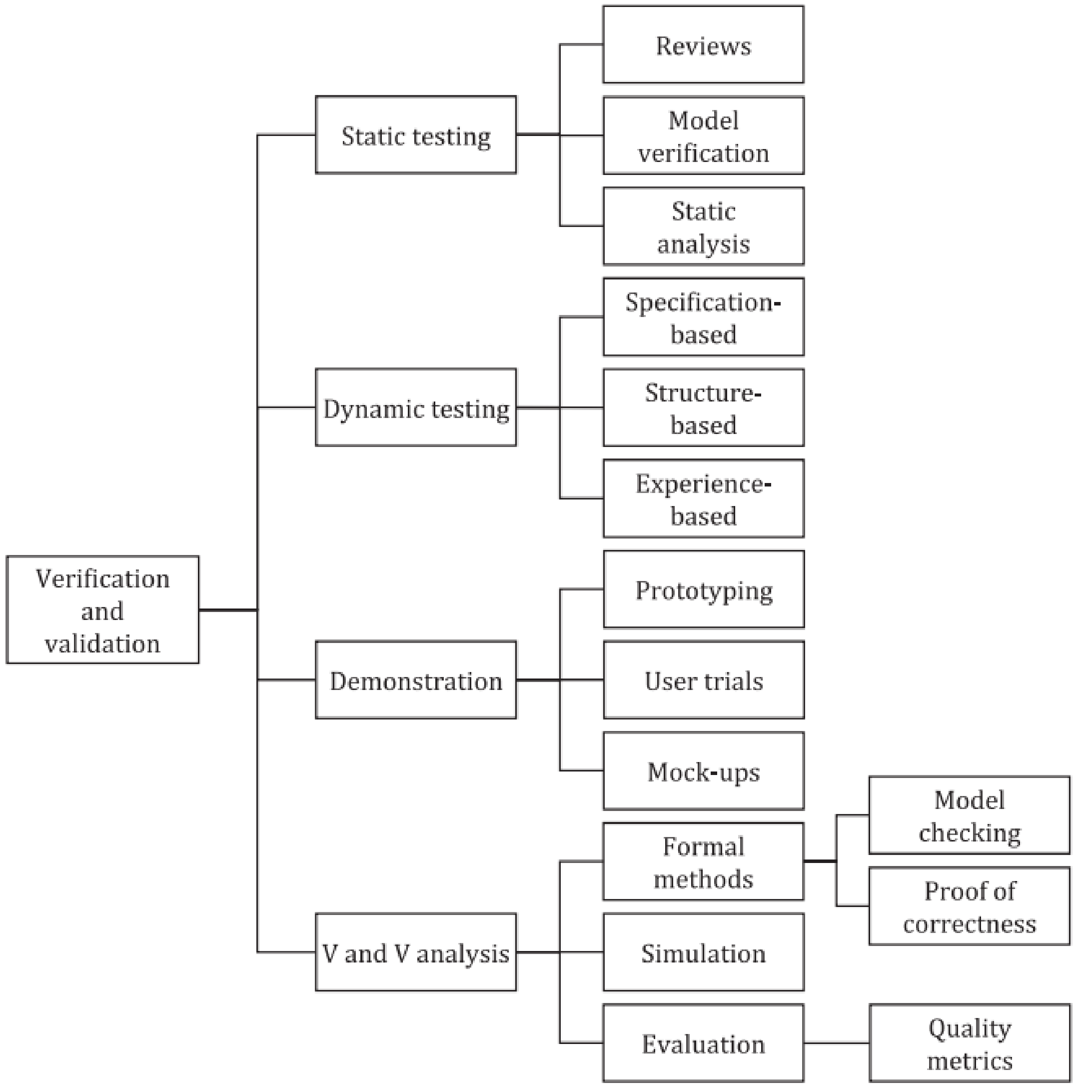

Figure 16: Verification and validation methods from ISO/IEC/IEEE 29119-1

The methods used for testing, shown in the figure above, are executed for more reasons than simply detecting bugs. As described by ISO/IEC/IEEE 29119-1, a single test item can benefit all individuals involved:

"— developers can use the information to remove defects, increase the code quality or learn to create better code in the future;
— testers can use the information to create better test cases;
— managers can use the information to decide when to stop testing;
— users eventually benefit from a higher product quality"

(ISO/IEC/IEEE 29119-1 2). Testing can benefit all of the purposes mentioned above and should be identified at the beginning of testing to identify true purpose. By identifying the true intentions of testing, this will assist the testing team to fully understand the purpose and execute testing properly and with intent.

The standard also briefly defines the risks of software testing. Two major risk categories are product and project risks. Product risks are related to the test item and the potential harm to users and others related if the product was not to work as intended. Project risks are related to how the project is developed and potential of not having the required skills to accomplish the project resulting in a higher possibility of the product being delivered late or over budget. The potential risks must be examined prior to testing to implement the strategy known as risk-based testing.

Another major relevant contribution of the standard, ISO/IEC/IEEE 29119-1, is the description of test design and execution. There are many strategies to testing and various test models that can be created for chosen strategies. The figure below presents the relationships between major test objects related to test design according to ISO/IEC/IEEE 29119-1. The figure shows the test model to be the key part of the relationships.

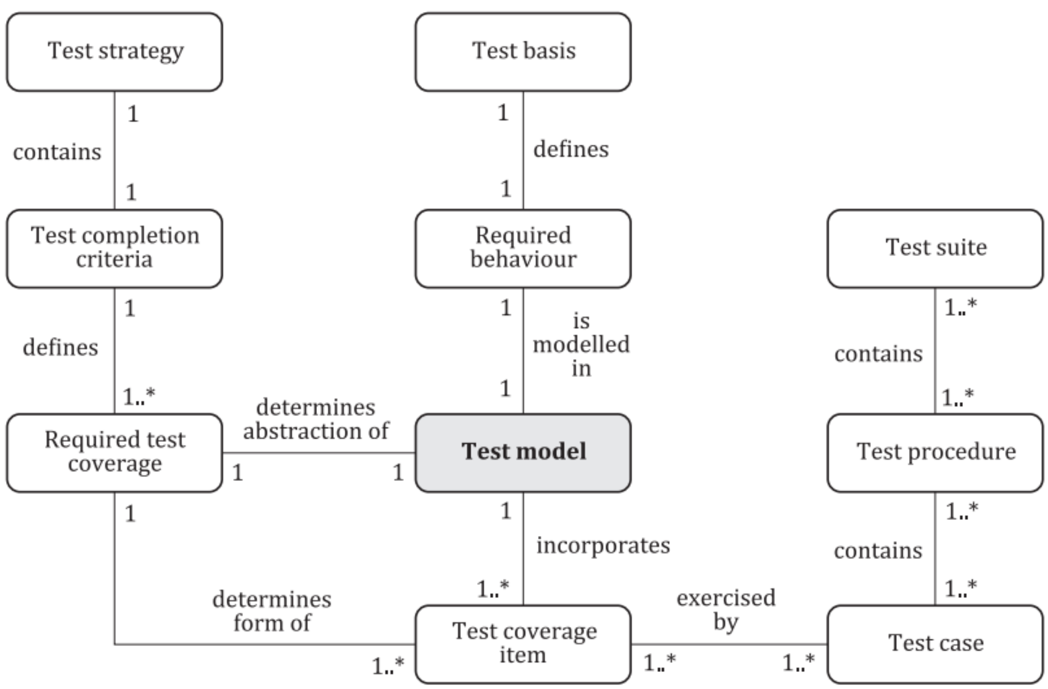

Figure 17: Test model in perspective from ISO/IEC/IEEE 29119-1

When deciding the appropriate test model, the figure above is important to keep in mind. In relation to the test model, the figure shows the behavior of the relationships to the required test objects. The test strategy contains the test completion criteria that defines the required test coverages. The test basis defines the required behavior which is modeled in the test model. Test suites contain a series of test procedures and test cases that are exercised by test coverage items that are incorporated by the test model. All of these required aspects of the test model are described in the figure above. The use of the test model assists testers to execute exhaustive testing to ensure the best performance from the test item in its final product.

4.1.3 Communication Standards

The Federal Communications Commision (FCC) sets the precedent in the United States for all equipment authorization that utilizes radio frequency communication based networks. This duty is also split with the National Telecommunications and Information Administration (NTIA). The FCC controls and operates usages for non-government affairs, while the NTIA controls government affairs within radio frequency operation. Equipment authorization is completed by the FCC for radio frequency emitting devices that operate from 9KHz-3000GHz. The FCC separates and categorizes different radio frequency emitting items in the following fashions:

- Incidental Radiators
    - Part 15, Subpart A
- Unintentional Radiators
    - Part 15, Subparts B and G
- Intentional Radiators
    - Part 15, Subparts C through F and H
- Industrial, Scientific, and Medical Equipment
    - Part 18
- Equipment Operating in Licensed Radio Services
    - Varying sections and locations on requirements and standards

With this project being grounded through RFID communication with a reader/writer and the tag embedded within our assembly, we are required to understand and follow the requirements and constraints set forth by the FCC. Defined in Title 47, Chapter 1, Subchapter A, Part 15, Section 3. Under definition 47 CFR 15.3(o) labeled ‘*Intentional Radiator’*: A device that intentionally generates and emits radio frequency energy by radiation or induction. With the definition in alignment with the core of this project, the defined sections above under Part 15 of Title 47 set forth by the FCC are the guidelines and standards that are necessary to follow throughout the duration of this project and our final presentation to prepare for exiting the University of Central Florida.

Under §15.205 of Title 47 is the Restricted bands of operation section. This is set forth by the FCC for civilian usage on the items that title suggests. The table of restricted frequencies is as follows:

MHz	MHz	MHz	GHz
0.090-0.110	16.42-16.423	399.9-410	4.5-5.15
0.495-0.505	16.69475-16.69525	608-614	5.35-5.46
2.1735-2.1905	16.80425-16.80475	960-1240	7.25-7.75
4.125-4.128	25.5-25.67	1300-1427	8.025-8.5
4.17725-4.17775	37.5-38.25	1435-1626.5	9.0-9.2
4.20725-4.20775	73-74.6	1645.5-1646.5	9.3-9.5
6.215-6.218	74.8-75.2	1660-1710	10.6-12.7
6.26775-6.26825	108-121.94	1718.8-1722.2	13.25-13.4
6.31175-6.31225	123-138	2200-2300	14.47-14.5
8.291-8.294	149.9-150.05	2310-2390	15.35-16.2
8.362-8.366	156.52475-156.52525	2483.5-2500	17.7-21.4
8.37625-8.38675	156.7-156.9	2690-2900	22.01-23.12
8.41425-8.41475	162.0125-167.17	3260-3267	23.6-24.0
12.29-12.293	167.72-173.2	3332-3339	31.2-31.8
12.51975-12.52025	240-285	3345.8-3358	36.43-36.5
12.57675-12.57725	322-335.4	3600-4400	+38.6GHz
13.36-13.41			

Table 3: FCC Restricted bands of operation for intentional radiators

Considering the above table and the design standards in mind, the operating band for commercial UHF RFID devices lies within the allotted frequency band of column 3 rows b & c in the above Table 3. The following standard set forth by the FCC is the standard set by §15.203 of Title 47, Chapter 1, Subchapter A, Part 15, Subpart C; titled *'Antenna Requirement'* informs us of the requirements of the antenna as the means of our communications to proceed within. The subsection definition states that an intentional radiator shall be designed with the notion in mind that another antenna from a differing manufacturer/modifier/producer cannot be furnished as a means of communication with our device. This notifies us that our antenna must remain fixated and permanently installed by the manufacturer/modifier (group 15 members). This does allow for the replacement and repair under the same definition. Standard 'plug & play' type connection jacks are not allowed under this definition, and informs the group that our designs for the RFID reader/writer and receiving tag must be properly handled throughout the duration of this project until the completion of our final presentation at the finality of Senior Design 2 in the Fall 2022 semester.

FCC standard under Title 47, Chapter 1, Subchapter A, Part 15, Subpart C titled §15.243 *'Operation in the band 890-940 MHz'* describes in 4 paragraphs some of the major standards and procedures when operating within the UHF RFID frequency range that this project is designed to operate within. Under 47 CFR 15.243(b), the field strength of any RF radiated emissions within the frequency band of 890-940 MHz cannot breach values of 500μV/m at a range of 30 meters. Within 47 CFR 15.243(a), the frequency band utilized here is not specified for messages or voice communications. Only measuring the characteristics of a material, as this project is utilizing, is viable when utilizing 890-940 MHz. Paragraph (c) refers to a further section in §15.209 describing where the field strength emissions limits are designated under certain frequency bands. The fourth and final paragraph of section 15.243 describes that the device shall be self-contained with no accessible controls for modification that can allow for adjustments that would not align with the corresponding standards presented in this section and set forth by the FCC.

§15.245 titled *'Operation within the bands 902-928 MHz, 2435-2465 MHz, 5785-5815 MHz, 10500-10550 MHz, and 24075-24175 MHz.'* sets forth the fundamentals for field strength and harmonics through a tabular representation as well as subsection paragraphs with defining attributes that continue to define the standard that is being presented here. Paragraph (a) describes the intentional radiators definition further and non-necessary to the scope of this project. Paragraph (b) introduces the table regarding field strength emissions from the defined device we are creating for this project. The following table is to be considered within the scope of this project and to be taken under major consideration through the finality of this project. Paragraph 47 CFR 15.245(b)(1) describes items that are exterior to the scope of this project and non-fundamental in the considerations for this project. 47 CFR 15.245(b)(2) details that the limits defined in the table for this section are specified at measurements of 3 meters.

Fundamental frequency (MHz)	Field strength of fundamental (millivolts/meter)	Field strength of harmonics (millivolts/meter)
902-928	500	1.6
2435-2465	500	1.6
5785-5815	500	1.6
10500-10550	2500	25.0
24075-24175	2500	25.0

Table 4: Field strength emissions from intentional radiators operated within certain frequency bands

From the table, we can see the top row discussing 902-928 MHz provides the baseline for our project and what we're hoping to accomplish with this project. These major constraints complete the standards that are required under the communication section of this project.

4.1.4 Documentation Standards

The United States Institute for Theatre Technology is the leading force in the entertainment, theater, themed entertainment, and performing arts industries when it relates to the establishment of standards. The standards set forth by the United States Institute for Theatre Technology (USITT) are guiding lights for the operational procedures and standards relating to varying items within the themed entertainment industry. These standards relate to items such as radio frequency communication labeling, identifying, and quantifying within a matrix to be shared amongst the team. Others such as schedules for all equipment utilized, in a similar fashion both a BOM and panel schedules found within low-voltage panels in all powered structures. These standards set forth by the USITT are to be followed to ensure that the 'show' element of our presentation, and the quality of immersion that the guest(s) experiencing this device will not falter in any way.

The first major standard to consider that is set forth by the USITT is the *'RF Bible'* standard. This standard relates to the *'USITT/TSDCA Sound Documentation*

Recommended Practice' in which all items relating to sound, communication, and encompassing items are included thereof within. The RF Bible standard relates dominantly to utilization within theater performances that utilize microphone based radio frequency communication. This standard still greatly applies to the project currently in-work here. The major application that will be utilized for our project is that of a matrix in which all frequencies, IDs, and required information for our Pokéball's. The example matrix format is presented below:

Device #	Transmission Frequency	Pokéball ID#	User ID#	Pokémon ID#	Reader /Writer Serial No.	RFID Tag Serial No.	Notes
1							
2							
3							
4							

Table 5: USITT RF Bible Matrix Example

Other standards involved in the documentation standards relate to personal experience developed through group members internship experience within the Walt Disney Imagineering (WDI) company. As a project within WDI is completed, as well as other firms, punchlists and "living-breathing" to-do lists evolve as the project evolves. This is the final completion step prior to turning over to the operational team within the governing park body that the experience was constructed in. Once the final punch lists are completed, this marks the final segment of a scene, zone, land, experience, attraction, and all other encompassing regions that are located within the punch list. This being a project grounded within themed entertainment, standards set forth by the pillars of the industry and professional experience gathered during stints with them should be followed with proper execution. These punch lists are broken down by the segmented 'scenes' that combine together to form a complete narrative that is told once experienced from the beginning scene to the final scene. With this not being a legitimate attraction to properly tie the label 'scene' to, it is paramount that this procedure is utilized and adjusted to properly fit within this project's scope. These documents are held internally within the group members, and dissemination of the lists is to not be shared externally to the group unless specifically requested by a professor/faculty member. This breakdown allows for proper communication and completion to occur at the arrival of an error or 'punch item' to make addition of. The lists for this project will include the following milestones:

- Senior Design 1 Report Punch List
- Senior Design 2 Punch List
    - Broken down by major milestone once in progress during senior design 2 in the Fall 2022 semester
- Senior Design 2 Presentation
    - Proper breakdown will occur during the formal stages in the latter portions of Senior Design 2

With these three major milestones to breakdown and digest, the following example showcases what a punch list that evolves continuously shall be represented as:

SD1 Punch Items			
Section	**Item**	**Fix**	**Notes**
Administrative			
Chapter 1.0			
Chapter 2.0			
Chapter 3.0			
Chapter 4.0			
Chapter 5.0			
Chapter 6.0			
Chapter 7.0			
Chapter 8.0			
Chapter 9.0			

Table 6: Example Punch List

This table can then be expanded to feature numerous items within one chapter, and fully communicate with the group non-verbally as needed to adjust and correct the errors within the project report and final construction in Senior Design 2. This format allows for all members to add, subtract, and modify items with an ease of use template that features all required items for proper communication as necessary.

4.2 Realistic Design Constraints

Every project must have realistic design constraints to successfully implement the design. The constraints have a firm understanding of the project's details and realistically demonstrate the restrictions. The Pokéball's design and production will be examined and each of the constraints below will reflect the design and production. Each of the constraints below are broad categories that will be examined in detail in relation to the Pokéball's design and production in order to be realistic.

4.2.1 Functional Constraints

Functional constraints will provide realistic design constraints for the project. A requirement of the Pokéball relies on its portability component. This restricts the potential of the Pokéball to be too heavy as well as too large in surface area. In order to abide by this constraint on the overall geometry and weight, the project set the appropriate requirements.

In addition to the constraints revolving around portability, there are also concerns for durability. As the audience for this project in ideal circumstances is broad and expected to be operated by various individuals, the Pokéball should be able to withstand a fair amount of wear and tear. The concept of the Pokéball revolves around the thought of a Pokéworld theme park, comparable to the relation of Harry Potter wands and Universal Studios, where individuals can interact with the Pokéball at the theme park. With this assumption, the park-goers would potentially drop the Pokéball while riding the attractions or accidentally drop it in a highly populated area. This concept that the Pokéball was based around requires it to be durable. Although it does not necessarily need to be the strongest material used, the Pokéball must have enough reinforcements to protect the hardware inside it. This constraint requires the materials to be relatively heavy-weight. The materials on the outer layer of the Pokéball must be stable enough and also affordable.

Individually these constraints are important and have a lot of influence on the design and production of the Pokéball. The functional constraints of this project also have a great effect on the other constraints that will be clarified deeper later. The functional constraints are very important especially to the initial design of the Pokéball as it clarifies the importance of the parameters it must follow.

4.2.2 Safety Constraints

In the hands of younger children, the Pokéball can be considered dangerous. Although the Pokéball is based on a franchise revolving around games and toys, that should not result in the Pokéball to be considered as a “game”. Due to the functional constraints, the Pokéball will be light but still have the capability to hurt an individual using it improperly. This constraint is not necessarily of relevance to the design or production but should be noted as it is a part of the business and marketing requirements.

The Pokéball should also have warnings regarding high heat and long water exposure. If exposed to high temperatures for a long period of time, the Pokéball may overheat and malfunction. Also one of the main concerns for the initial Pokéball design does not revolve around its capabilities to withstand water contact. For this reason, the excessive heat and water constraints are important to note for safety. This will prevent individuals from having their Pokéball malfunction. These circumstances should be preventable as the intended use for the Pokéball does not revolve around these constraints.

4.2.3 Quality Constraints

As previously mentioned in the functional constraints, the intentions of the Pokéball have great importance on the design constraints. To restate, the intentions are for use as a Pokémon related theme park. The majority of park-goers typically intend to go to a theme park for a full day’s worth. For these concerns, the Pokéball requires reliability. The Pokéball at the minimum must last the majority of a day on low power mode. With the assumption that the Pokéball will not be in constant use for the entirety of the day, it should be able to function at various points of the day at a minimum. Due to this constraint, we must be conscious about the design and the lasting impacts. Considering our intentions for this project, this is a very important constraint revolving around the design of the Pokéball.

4.2.4 Manufacturing Constraints

Manufacturability constraints require the components of the Pokéball to be limited to a simple build and easy to acquire materials. Due to the intention of the Pokéball, it should be easily manufactured as it will be mass produced for the majority of the park-goers. The manufacturing constraints are directly related to the economic constraints that will be detailed below. The Pokéball should be relatively simple to assemble for a higher turnover rate. The production of the Pokéball is what the manufacturing constraints directly affects. In order for our project to be successful in the long run and be a lasting product, the manufacturing constraint is important.

4.2.5 Economic and Time Constraints

Economic constraints limit the potential parts and have a direct impact on the other constraints. As our project is funded by four students, we do not have access to certain desirable components. The budget for the Pokéball is ideally under $200. Considering the required components, this budget is overestimated as some components are already in possession of some members and available for use for the Pokéball. Also for the desired requirements of the Pokéball, some of the components are potentially inexpensive. The cost aspect of the Pokéball is better if maintained as low cost due to the manufacturing constraints. Ideally if used for its intention, the low cost nature of the product will assist the manufacturing constraints to be a nonissue. The economic constraints are correlated to the functional and quality constraints. These requirements expect for the Pokéball to be of higher quality. The economic constraints are important to many of the other constraints and important to the engineering requirements as well as the marketing requirements.

In addition to the economic constraints, the time constraints are as important and also connected to the other constraints. The research portion of this project is to be completed by early August 2022. This should allow for the team to decide on the proper parts to purchase and determine the exact specifications and constraints to follow. After mid August 2022, the Pokéball design should be finalized and begin the production and testing phase. The Pokéball should be fully built by late October 2022. The initial design flaws should be solved by that point. Then the testing portion can begin to prepare for the product presentation in late November-early December. Considering the time constraints, it is important to realistically determine deadlines for the project to ensure a better product.

4.2.6 Aesthetic Constraints

For our project, the aesthetic constraints are very important. The Pokémon franchise was established in 1996 and is well-defined and has a proper following. As our project is a component of the Pokémon industry, it must be reflected as well. If the Pokéball is not properly portrayed with its defining characteristics, it will not be well received by the public. The Pokéball has a distinguishable design and our product must reflect said design. It is a sphere where the top half is red and bottom half is white. The two halves meet at a black line going horizontally across with a white button in the center. When properly designed, this will reinforce our project's intentions. Pokémon is known by everyone, not only individuals involved with the franchise. It is very important to implement this design correctly. This constraint is also associated with functional constraint. The Pokéball is portable in all of the Pokémon franchise's works therefore to mimic this our Pokéball must also be equally as portable. This constraint requires the Pokéball to have a specific design and requirements to meet rules of portability.

4.3 Updated Standards and Constraints

The project shifting as Senior Design 2 progressed caused the groups defined standards and constraints to arise new or modify. As the project continued, original standards that were outlined above all stayed in play throughout the project. The above standards and constraints continued to guide us as a group, and provided much needed structure to the overall project. The major changes to standards and constraints were in construction and time. Deliveries of components, delays due to hurricanes, and supply-chain shortages littered the project. These were expected in various ways, so components and selected materials were acquired prior to major issues occurring due to such.

In construction, the new major standard was the utilization of soldering standards. The standards were provided within the WebCourse for Senior Design 2, and guided the group on how to properly construct our boards in the Senior Design Laboratory. These standards that were provided allowed us to properly use the equipment without any harm occurring to the project hardware, or group.

5.0 Hardware Research

The following sections included in this chapter discuss the research committed to numerous aspects of this project. When discussing how a project of this type can be completed, many considerations must be made. How the handheld device will communicate with the overall system, what materials will be used for the shell of the Pokéball, what microcontroller will act as a network hub, and other considerations are included.The research discussed in this following section has been gathered from scientific articles, textbook research, and higher education literature.

5.1 Communication Networks

The communication network is crucial in two aspects for this project. When communicating from the Pokéball to the Pokémon gym battle experience, and the major network of the themed land itself that initializes and updates information in both itself and the Pokéball. When deciding what type of networks to utilize for both aspects, we investigated various network types. Some of the key features we require are: robust, reliable, private, and responsive. All of these requirements are discussed in the following subsections.

5.1.1 Wired LAN

A local area network (LAN) had to be researched for the ‘back-of-house’ system that is integrated on the larger scale of the entire land. When researching different LAN networks, certain aspects of differing LAN connections stuck out amongst others. Wired LAN provides precisely what is needed on the backside of this project on the basis of reliability, privacy, and responsiveness. Wired LAN is the ‘most common’ connection for LAN usage. This connection ties to a central server, which is utilized for communication between differing experiences to update the set parameters necessary for guests to utilize them properly. This network type also provides access to only those physically connected through an ethernet connection. Other network types, such as the peer to peer (P2P) LAN were found to be less powerful and didn’t provide a centralized server that is necessary for the backend control needed.

5.1.2 Wireless LAN

Wireless LAN has the potential to be utilized in numerous avenues for this theorized themed land. This network configuration allows for radiofrequency as the basis for data communication, but is also open to numerous security threats if not configured properly. This network configuration is also highly unreliable with factors such as location, number of users accessing it, and others. Wireless LAN is utilized in nearly every public location, and theoretically will be utilized in many ways on both front and backend applications for guest use. The practical utilizations for wireless LAN in this project is unfortunately exterior to the scope of this project.

5.1.3 Infrared Communication

Infrared (IR) communications are based on infrared radiation that utilizes wavelengths neighboring those of visible light on the electromagnetic spectrum. Wavelengths of IR communications are larger than those of visible light, but smaller than those utilized in radio wave communications. The range of wavelengths that infrared occupies is typically defined from 700nm - 1mm in length, while the frequency range is roughly 300 GHz - 400 THz. Interior to the wavelength range, IR can be separated into subcategories of near, mid, and far communications. Near IR is typically designated to 700nm - 3μm, mid IR designated 3 - 50μm, and far IR designated from 50μm - 1mm.

When designing the interior of the Pokéball, communications that can penetrate walls & general rigid geometry is crucial for the success of this project. This is in-part to the communication system being embedded on the interior of the Pokéball, but also due to the nature of interacting with attractions and other users utilizing the interactive Pokéball system. Unfortunately, IR communications do not process these necessary requirements. IR communications require line-of-sight (LOS) clearance to communicate with the intended receiver, and also fail to penetrate through surfaces that are necessary to see this project come to fruition. If infrared communication was to be utilized, the design of the Pokéball vessel would be drastically changed to no longer match the world of Pokémon, as well the project would require complete overhaul. The physics and engineering specifications that describe further as to why IR communications are limited in use are exterior to the scope included in this project.

5.1.4 Bluetooth Communication

Bluetooth communication methods were the next progression in the research for how the Pokéball vessel can communicate with both other users, and the interactive experiences that are being pitched in this project. Bluetooth communication utilizes short-range radio wave communications, situated in the ISM band of 2.4GHz designated by the Federal Communications Commission (FCC). The true range that Bluetooth is designated to operate in is 2,400 - 2483.5 MHz, and typically reaches a range of 30 meters in commercial usage. While extensive design could take place to increase the range to accommodate for utilization throughout an entire experience or themed land; the time allotment to design, test, and implement would extend further than the deadline of this project. Other factors also made Bluetooth incompatible with this project, such as the security risks that occur when utilizing this communication system on a large scale. Considering this item can not only communicate with an internal database, but also with other users leaves room for potential cybersecurity issues if a cyberattack were to take place. While this project is intended to store no personal information of the guests in its realization, room for security issues can still arise with Bluetooth systems in place. Utilizing Bluetooth as a proof of concept in this project would suffice, but for actual deployment other communication means would be necessary. Therefore, we have decided to forgo the utilization of Bluetooth, and rely on other communication means as the basis of this project.

5.1.5 Radio Frequency Identification

Continuing the research of how our Pokéball system will communicate with the overall experiences of our themed land and the neighboring users; we further navigated through the electromagnetic spectrum to reach Radio Frequency Identification (RFID) systems. RFID systems include two major subsystems: readers/writers & tags. Unlike IR communications, RFID does not require LOS for communication, nor is it as limited in penetrating rigid geometry. There are major positives to the deployment of RFID communication means in a project such as this. Some of the positives include the lacking requirement of LOS communications, as well as the capability of robust control of reading/writing terminals to initialize and identify numerous differing tags at one time. Just as with IR communications, RFID systems consist of numerous frequency ranges. Low frequency (LF) ranges from 30 - 500KHz, and can operate within ~1 meter of the reader/writer. High frequency (HF) operates from 3 - 30MHz, and within ~1 meter of the reader/writer, similar to LF. Ultra-high frequency (UHF) occupies the frequency range of 300 - 960 MHz, and can be read from much farther distances of +8 meters. While RFID provides numerous upsides for a project of this caliber, there also comes concerns that could arise.

Types of Radio Frequency Identification			
	Low-Frequency	High-Frequency	Ultra-high Frequency
Frequency Range	30 - 500 KHz	3 - 30 MHz	300 MHz - 3 GHz
Common Frequency	125 KHz	13.56 MHz	960 MHz
Relative Price	$	$	$$
Read Range	~ 1 M	~ 1 Meter	< 8 Meters
Read Compatibility	Usually single read	Good	Excellent multiple read

Table 7: RFID Comparison Table

In the previous section discussing Bluetooth, the concerns of cyberattacks was brought up. This concern also arises with the implementation of RFID. There are concerns that potential attacks could occur to corrupt, transform, and overall distort the data that is being transmitted by the RFID tags, readers, and writers. These concerns are hyperspecific in context though, as this would require guests in range of the devices to own and operate personal RFID readers and writers. This risk can also be taken under control by two main measures: the prohibition of personal RFID readers and writers interior to the themed experience, and not utilizing any personal information in the transfer of data during the experiences themselves. Enacting these two measures can reduce the risk of potential cybersecurity issues to a negligible percentage value.

Continued research into RFID implementation shows differing subcategories further than frequency band range. Passive and active methodology in RFID communications also inform major design decisions for this project. The discussion of passivity and activity are featured in the following subsections.

5.1.5.1 Passive Tags

There are two major types of tags that can be embedded in our vessel, passive and active tags. Passive tags are constructed and designed to feature no internal power system. The power required to transmit data from the antenna comes in the form of the electromagnetic waves sent by the reader/writer and induces a magnetic field in our passive tag. Under Ampere's law this induces a current in the tag system to then return information requested by the RFID reader. There are both positives and negatives to the deployment of this system. Some negatives include the lack of range due to no internal power system, and lack of scalability memory sizes due to the same reasons. Positive aspects of this tag type are more financially based, as the tags are far cheaper to manufacture and last far longer due to not relying on other parts for operation. This option deployed in the proper size and scale can be tolerated for a project of this sort, but these types of tags would require more output power and reading locations across the differing experiences. Financial analysis could be completed to compare different system deployments, but that is vastly outside the scope of this project. This item also comes completely designed and does not allow for the requirements of Senior Design to be properly met when utilizing passive tags. The entire objective of this project, while fully realizing an operational prototype of the RFID device we are completing here, is to design a custom PCB and implement it with proper standards and engineering work applied. Utilizing purchased passive tags would not allow for this to be completed and would shift the focus of the project into other realms that would not be featured within the scope that was pitched to Dr. Lei Wei during the beginning phases of this project.

Figure 18: Passive RFID Key and Tag

5.1.5.2 Active Tags

Included in the active tag category lies two other subsets; beacons and transponders. Though similar in design and completion, the realization of these two subsets of active tags provide differing results. Active tags differ from passive as previously stated, due to an onboard power system. Thus allowing for greater storage capabilities, and faster sending & receiving times at the request of a reader. This system does cause for higher cost per unit, and an overall larger system, but does allow for a robust level of control and far greater transmitting ranges due to that additional power system. When concerning the idea of active tags, safety and security must be in the conversation. Active tags do introduce the concept of tracking the Pokéball's movement when in range of a reader. When away from a reader set to the designated frequency specified for the Pokéball tags, the idea of tracking becomes a nonissue. When in the range of readers surrounding the theorized land, the understanding of where guests go, at what time, and in what volume provides necessary data to continually improve the efficiency & effectiveness of this land. This data can be compiled for numerous utilizations, but the detailed description is outside the scope of this project.

Stated above are the two major subsets of active tag systems; beacons and transponders. Beacons are active tags that continually operate at a certain clock cycle frequency, usually every 3-5 seconds transmitting outward. These systems do not operate at the control of a reader antenna, but operate on that cycle regardless of whether the transmission is heard or not. These systems can be utilized in numerous industrial industries, but if the system is battery operated (as this project will be) you run the risk of wasting energy even when at a far range from the experiences that best utilize the Pokéball. Transponders are similar in fashion to beacons, but with much higher efficiency than that of its counterpart. Transponders will sit in an idle state, wasting far less energy than that of a beacon, until the reader antenna requests information from the specific tag. These systems are remarkable at wasting little to no battery life when far from the reading range radius.

Figure 19: Active RFID Transponder

5.2 Microcontrollers

The purpose of the microcontroller in this project is to be the base of our RFID transmitting, writing, and receiving. This can be realized by utilizing a specified microcontroller unit (MCU) with the correct number of general purpose input/output (GPIO) pins connected to a purchased RFID reader/writer. With varying MCU manufacturers, extensive research was completed to ensure the proper unit was selected; along with the proper RFID module that is compatible with our selected MCU. The following sections provide a breakdown and in-depth look into the decisions surrounding the MCU & RFID units.

5.2.1 Microcontroller Selection

The search for a base microcontroller began with looking into the product line that is provided by Texas Instruments of Dallas, Texas. The primary reason of searching into the products of Texas Instruments (TI), was due to the following:

- Supply chain availability
- Expansive product line with varying MCUs
- Experience gained in relation to the manufacturer through varying coursework
- General ease of use

These features among numerous others guided us to begin the search here. The main MCU that TI has available to purchase is the MSP430 microcontroller, a 16-bit general use analog microcontroller that was first introduced into the electronics market in 1992. This MCU encompasses over 2000 variations, under four major categories:

- General purpose
- Sigma-Delta Analog-to-Digital Conversion
- Capacitive Touch Sensing
- Ultrasonic Sensing

Of the four major options, general purpose is where the vast majority of their products land, and where this project will also land under. Two primary families of MCUs live under the MSP430 general purpose umbrella that fit into our project potential:

- MSP430FRXXXX
- MSP430G2XXX

The major differences of the MSP430FR and MSP430G series of MCUs lie in their price point, communication systems, memory size, speed, and overall capabilities. Though both are named MSP430's and operate under 16-bit systems, both vary in operation.

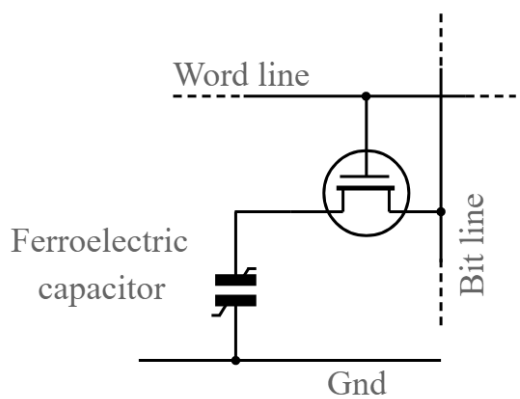

Figure 20: FRAM Schematic

The MSP430G series is a staple of the 'value line' greater-value MCUs that TI has to offer, as where the MSP430FR series is included in the more robust FRAM series of MCUs. FRAM, Ferroelectric Random Access Memory is a specialized category of non-volatile flash memory that has a much faster reading/writing capability than traditional RAM. This type of memory can accommodate for many other features that normal flash memory cannot account for.

Comparisons	FRAM	FLASH
Non-Volatile	Yes	Yes
Write Durability	1x10^15	1x10^6
Write Speed (per cell)	50ns	0.05ms
# of Processes to Write	1	2
Operating Voltage	~1.5V	10-14V
Safe to Solder With?	Yes	Yes

Table 8: FRAM vs FLASH Comparisons

The advantages in lifetime, speed, and capabilities of the FRAM based MSP430 MCUs versus those of the traditional flash memory based MCUs guided us to then proceed with further research into the MSP430FR MCUs. The MSP430FR series of MCUs features over 130+ distinct microcontroller units on the TI catalog. The scope of this project does not allow for extensive search into this number of differing units, so experience with specific FR series units was needed to guide the research completed in this stage of the project. Through differing courses and personal development, experience was gained with the MSP430FR6989 microcontroller unit that TI offers. This distinct product includes numerous features that allow for endless general purpose projects, and with those features we are able to complete this project as defined by our set scope.

MSP430FR6989	Data Sheet	Research
Specifications	**Data**	**Interpretation**
Architecture	16-Bit RISC	Fast operating with low power consumption
Minimum Operating Vcc	1.8V	Informs battery choice
Maximum Operating Vcc	3.6V	Informs design decisions
Minimum Operating Temperature	-40C	Meets required spec for this project with ease
Maximum Operating Temperature	+85C	Meets required spec for this project with ease
Max Clock Frequency	16-MHz	Required for use with the

		RFID module
FRAM Memory	128KB	Informs design choices
SRAM Memory	2KB	Informs design choices
GPIO Pins Available	63 / 83 Pins	Depends on the package of PN or PZ
Active Power Use	101.25 µA/MHz	Informs design choices for other modules tied to MCU
Clock Systems Available	DCO HFXT LFXT	Informs design decisions on backend / software interpretation of data
ADC	12-bit SAR	Useful conversion knowledge to be used in design.
UART	2	Informs design choices.
I2C	2	Informs design choices.
SPI	4	Informs design choices.
Data Bus Width	16-Bit	Informs design choices.
Price	$12.80 per unit	Major consideration.

Table 9 MSP430FR6989 Data Sheet Specifications

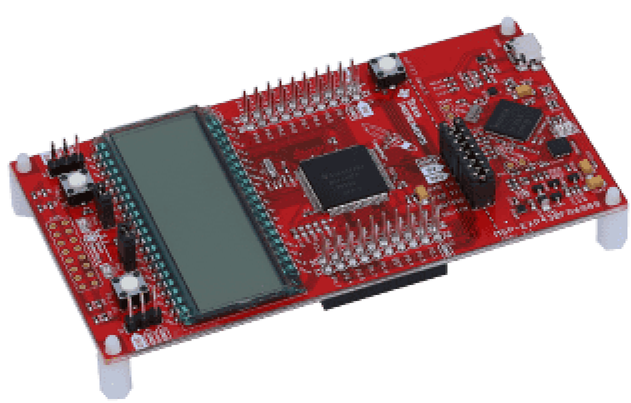

Figure 21: MSP430FR6989 Visual

Through discussion and design collaboration within the group, we found this MCU to meet all specifications defined by the scope of this project. Although, to ensure the due diligence and proper research defined by an engineering project was met, we continued to explore other MCU options that were provided by different vendors and manufacturers. This led to searching into other widely known MCUs such as the Arduino UNO.

The Arduino UNO is a highly popular MCU designed and manufactured by Arduino. The central processing unit (CPU) that is utilized in the UNO MCU is the ATmega328P designed by Atmel. The Uno comes with its own integrated development environment (IDE) that allows for the MCU to be programmed. The current version of the Uno that has been researched is the Rev3 model that is available for purchase directly from the Arduino catalog. Similar to the search beginning with the catalog of TI, the search into the catalog of Arduino was informed from key factors such as:

- Supply chain availability
- Current availability through group members personal inventory
- Applicable experience gained through personal & research projects
- General ease of use & a lower learning curve than other MCUs

Arduino as a manufacturer features an expansive catalog of boards with differing MCUs to choose from. Some of the more popular boards (from quantity sold) are the:

- Arduino UNO
- Arduino Nano
- Arduino Due
- Arduino Mega

As mentioned previously with the research discussion on TI MCUs and boards, the scope of this project does not allow for the extensive research into every Arduino board that is manufactured. Understanding and refining the scope of this project allowed for the search into the UNO to represent the catalog of Arduino.

There are many aspects to the UNO and the Atmega328P that allow for the scope that we did define to be completed. Just like the MSP430, the Atmega328P features an advanced RISC architecture, but rather 8-bit than 16-bit control. The board features of the UNO are also accounted for in the decision for this project.

Arduino UNO Board Specifications	
Operating Voltage	5V
Input Voltage Minimum	6V

Input Voltage Maximum	20V
Input Voltage Recommended Range (Recommended by Arduino)	7-12V
Digital I/O Pins	14
Flash Memory	31.5KB (From Atmega328P)
SRAM	2KB (From Atmega328P)
Clock Speed	16 MHz
Power Jack	Yes
USB	Yes
Price	$27.60

Table 10: Arduino UNO Board Specifications

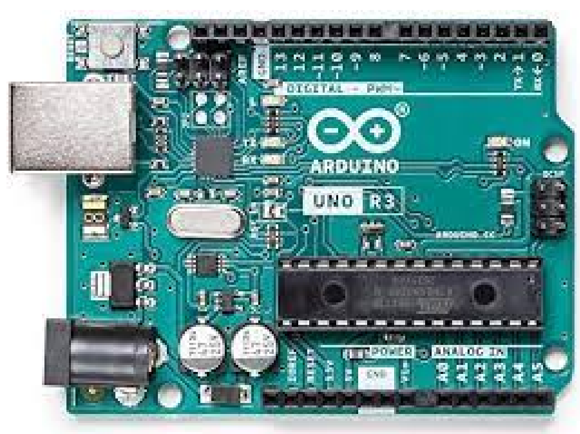

Figure 22: Arduino UNO Visual

Along with these specifications, the UNO board features other specifications that were kept in mind as research continued. Some of these factors included the layout of the board, the I/O pins, and how those pins can be configured. The board of the UNO includes both a +3V3 and +5V pins, along with 6 other pins that are configurable with pulse width modulation outputs (PWM). The PWM output pins can be configured separately to obtain different duty cycles on each of the pins if desired. Although this feature proves useful for many other general purpose projects, the PWM pins provide no expansive use in the scope of this project.

The Atmega328P microcontroller as stated is an 8-bit advanced RISC architecture, but created with CMOS. Just as the RISC architecture of the MSP430 of TI, most instructions on this MCU are executed in a single clock cycle. With sacrificing 16-bit architecture for 8-bit architecture, the main difference would be speed of the processes and arithmetic involved in the processes. Even though our data-bus is only 8-bits in width, the Atmega328P does include registers that feature true 16-bit design for timers. The TCNT1, OCR1A/B, and ICR1 registers are all designed and implemented into the Atmega328P as 16-bit registers that can be accessed via the 8-bit data bus through the CPU. Also featured is 32 x 8 general purpose registers in the MCU.

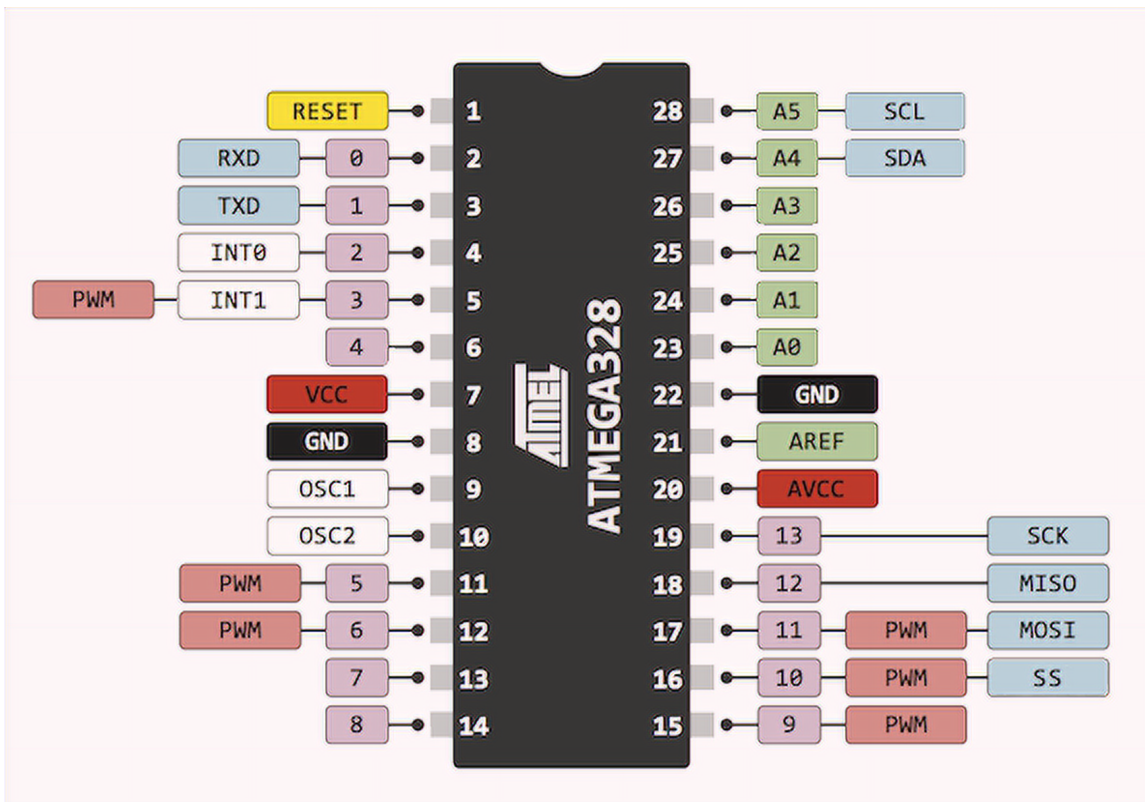

Figure 23: Atmega328P Visual

Atmega328P Data Sheet Research		
Specifications	**Data**	**Interpretations**
Architecture	8-bit RISC	Half the bit value of MSP430, held in consideration with processing power

Minimum Operating Vcc	1.8V	Informs battery choice
Maximum Operating Vcc	5.5V	Informs design decisions
Minimum Operating Temperature	-40C	Meets required spec for this project with ease
Maximum Operating Temperature	+85C	Meets required spec for this project with ease
Maximum Clock Frequency	20MHz	Higher than MSP430, held in consideration for design decision
Flash Memory	32KB	Much less than the MSP430, and not FRAM
SRAM	2KB	Matches that of MSP430
GPIO Pins Available	23	Informs design decisions
Active Power Use	0.2mA/MHz	Higher than that of MSP430, but not a make-or-break number in design process
Power-down Power Use	0.1μA/MHz	Informs design decisions
Clock Systems Available	Low Power Crystal Osc. Full Swing Crystal Osc. Low Frequency Crystal Osc. 128kHZ RC Osc. External Clock	Numerous sources for the clocks that can be designated to various elements in the MCU
ADC	10-bit	Informs design decisions
UART	1	Informs design decisions
I2C	1	Informs design decisions
SPI	2	Informs design decisions
Data Bus Width	8-bit	Informs design decisions
Price	$3.11 per unit	Major consideration

Table 7: Atmega328P Data Sheet Research

Arduino features a full SCH, PCB, and 3D view of the UNO board on their online catalog. When exploring the schematic view, certain elements of the existing engineering efforts put into the board were both apparent and appreciated. The Atmega328P features an analog VCC pin named AVCC that separates the analog from the digital to reduce noise in the overall MCU. The AVCC pin ties into a +5V source that then ties directly into a 2nd order low-pass filter with a 10μH inductor and 100nF capacitor. The normally established VCC pin ties directly to a 100nF capacitor through the +5V source to act as a bypass capacitor to element noise in the system. Being able to analyze both AVCC and VCC pins as examples of the noise management engineering of the MCU compels an argument to show the quality of the MCU that is being researched during this phase of the project.

When arriving at the selection point for our MCU, considerations that have been stated previously and others were kept fully in mind. The comparisons for the two units presented and those considerations are featured here:

MCU Comparisons		
Comparison	**MSP430**	**Atmega328P**
Clock Frequency	16 MHz	**20 MHz**
Op. Voltage Range	1.8V - 3.6V	**1.8V - 5.5V**
Memory Size	**128 KB**	32KB
Power Usage	**101.25 μA/MHz**	0.2 mA/MHz
Price	$12.80	**$3.11**

Table 8: MCU Comparisons

The final decision was made to be the Atmega328P and the components thereof found in the Arduino UNO as the basis for our MCU utilized throughout the finality of this project. The price, clock frequency, and range of voltage operation compared to that of the MSP430 stood out as the specifications we required for this project. Other factors, such as the open-source nature of the Arduino IDE and variety of usage found within the board and MCU, led us further into confidence with the decision of the Arduino UNO utilizing the Atmega328P. With the current situation as of Summer 2022 and the supply chain of most electronic goods and the global economy, group availability of both MCUs that were researched was found to be an extreme gift and is felt with immense gratitude amongst all members. Having the lead time of 0 days on a major component defined by the scope of this project provides for the immediate advancement of the design, and realization of this project to not be further constrained into the future of both Senior Design 1 & 2

5.2.2 RFID Transmitter Module

In our application there needs to be a transmitter RFID module, which is the Pokéball in our case, and a receiving RFID module, which is going to be the reading device. For the transmitter, there are two options as discussed in Section 5.2.5.1 and 5.2.5.2 to be considered: active or passive tags. Both have their own benefits, but the passive tag may be considered the better option.

Since passive tags do not require power as they are powered by the energy transmitted by the RFID reader, this type of tag will be better for the Pokéball since they need to be able to fit in the guests' hand. Active tags are larger and require a battery, but allows the use for active tracking. The table below differentiates two types of RFID modules.

Depending on what microcontroller the project will utilize, it will either use a Low Frequency RFID or a High Frequency RFID which will be using 125 kHz or 13.56 MHz respectively.

	Passive	**Active**
Frequency	860-960 MHz	433 Mhz - 2.45 GHz
Range	5 Meters	100 Meters
Tag Size	Size of smartphone	Smaller than credit card
Power	Internal battery (2-5 years)	External source
Cost	Around $25-$50	Around $1

Table 13: Differences Between Passive and Active RFID Tags

During the testing phase of our project, a solderless implementation will be used to determine how the final PCB should be designed. Once the PCB is designed, a decision whether the RFID should be through hole or surface mounted is required to be made. The through hole's benefits are it is more cost effective and is able to be self soldered on. However, the benefits of the surface mount RFID is that it is able to be implemented on a much smaller board size but requires a perfect design. The table below compares some of the differences between the two.

Through Hole	Surface Mount
• Hole is required for component lead insertion • 2 sided boards are rare • Component lead spacing is typically 0.100” or greater • Manual assembly using soldering • Larger PCB footprint • Rework is relatively simple	• Components mounted directly to surface, so no holes • 2 sided boards are common • Component lead spacing around 0.0157” • Automated assembly • Minimal PCB footprint • Rework is more involved

Table 10: Differences Between Through Hole and Surface Mount

5.2.3 RFID Reading Module

There are currently many different solutions to implementing a RFID reading device. In our specific application, a MSP430 microcontroller is one possible solution to read in the passive RFID tags. The EM18 RFID reader is an example module that this project can utilize for reading tags. When the EM18 is connected to a microcontroller with a LCD display, it will print the corresponding ID on the display.

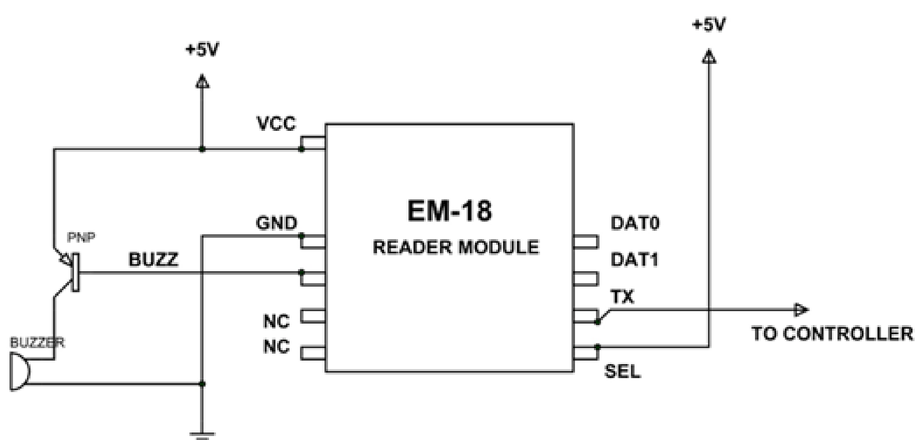

Figure 24: EM18 Reading Module Simple Circuit Diagram

The figure above illustrates the pin layout and how it can be connected to the MSP430 microcontroller. The EM18 features two types of communication between the controller depending on whether the SEL pin is set to HIGH or LOW. When the SEL pin is set to HIGH then the form of communication is RS232 and if the SEL pin is selected as LOW then the form of communication is the WIEGAND. The popular form of communication for this module is the RS232. The EM18 operates on a RF transmit frequency of 125 kHz and has an integrated antenna that has a range of 12 cm. The serial communication parameters are 9600 bps, 8 data bits, 1 stop bit. It is powered by a 5V power supply.

Another solution in implementing the RFID reading module can be used on an Arduino microcontroller. The Mifare RC522 reader module is an available sensor that is compatible with the Arduino. In this application, it is different from the EM18 on the MSP430 as the RC522 is powered by 3.3V. This module operates with 13.56 MHz frequency. The module supports SPI, I2C, and UART communication but out of these types, SPI is most often used since it is the fastest with a maximum data rate of 10 Mbps.

Either of these two solutions can be an efficient way to read a RFID tag because they both utilize low operating power at around 30 mA of current and under 10 mA when in the power down mode.

Figure 25: RC522 RFID Module

5.2.4 Printed Circuit Board

The printed circuit board is vital in our project because it is what connects every hardware component together. There are many types of PCBs as they come in single-sided, double-sided, multilayer, rigid or flex types. For ease of design and assembly, it may be wise to stick with a single sided board as it is easier to manufacture, design and repair if needed. As mentioned in section 5.2.2, there are two types of mounting methods: through-hole technology and surface mount technology.

With through-hole technology, the lead components are installed into pre-drilled holes on the circuit board which are then soldered to pads on the opposite sides. Surface mount technology requires precise placement of electrical components on the surface of the circuit. This is where a PCB software design comes into action.

The use of EAGLE by Autodesk may be the best solution to design a PCB because we have had prior experience designing a PCB from previous coursework. Another benefit of using this software is that it allows free usage with limitations. Utilizing this software will allow us to decide how to configure the PCB, but it may seem like a through hole designed board is the better solution.

An easier approach in place of a PCB could be to take advantage of Arduino Shields. Arduino Shields are pieces of hardware that sit on top of the Arduino providing specific purposes. With a Proto shield, it is an Arduino shield that allows for custom circuit building. We can take our existing breadboard layout and solder them to the surface of the Proto shield.

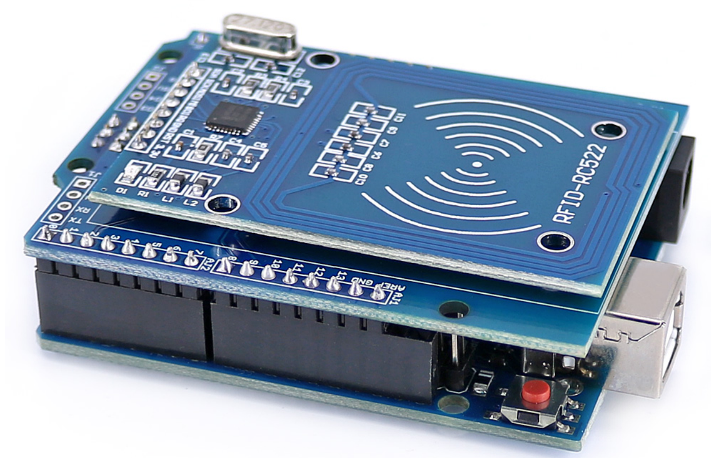

Figure 26: Arduino with RFID Shield

5.2.5 Battery

The microcontroller will require a battery to operate wirelessly from a computer. For the MSP430 microcontroller, to power the board two AA batteries can be used with a step up booster. The step up booster can then be connected to the MSP430's external power pins. The Arduino is able to be powered by a 9V battery because it has an operating range of 6 to 20V. To make it work, a 9V battery will be connected to a 9V battery snap connector to barrel plug converter which will connect to the Arduino.

5.3 Pokéball Shell Material

This following section describes and walks through the research and development on the material selection for the hard shell casing that the Pokéball will be constructed of. The PCB containing the RFID receiver will be encased in the interior of the Pokéball once the final showcase installation is made for Senior Design 2 in the Fall 2022 semester. The following discussion details how the world of Pokémon and the story we are telling is the driving force of this project, and how that integral story informs all of the design decisions that are made along the course of this semester and next. A crucial element to the project's success is the fidelity in which we present the world of Pokémon and how immersed a user can be once inside of this fictitious land that is presented here. The core idea that RFID items can be controlled in the palm of the guests hand adds a layer of care that was highly regarded during the design of the Pokéball.

5.3.1 Staying True to the World of Pokémon

Beginning with the design of the Pokéball itself, the source material was treated as a guiding light for how this design can not only function as intended, but stay true to the world of Pokémon at the same time. When discussing the project during the proposal phase, the stretch goal of creating more than one Pokéball was mentioned and fleshed out. This allows for more room to explore the world of Pokémon itself, and the differing Pokéballs that are offered inside of this universe. Although enticing to those who are fans of the Pokémon franchise, staying grounded to the scope of this project was necessary. A decision to create one type of 3D printed shell, but spend more time on that design to ensure thematic requirements were met was agreed upon. Rather than design 2-4 differing shells and produce a product that is less than 100% up to standards. Once the decision to create a hard shell that is reproducible was made, the study of the world of Pokémon and the Pokéball itself was required.

5.3.2 Brief History on the Pokéball

Defined by the Pokémon Group and a verified online encyclopedia named "Bulbapedia", the Pokéball is a critical item to a user's quest to catch and store Pokémon. The term is used as both a blanket term for all varieties of Pokéball, but also the proper name to the basic starting item to begin the trainer's journey. This basic item is the one that is defined by the scope of our project.

Figure 27: Pokéball Reference (Provided by The Pokémon Group)

In the base Pokémon world, there are 27 variants to the base Pokéball. The core design is an upper red shell and lower white shell combined together with a black internal case that features a white button at the center. The white button is a momentary push button that allows for the Pokéball to then attempt to either capture or release the Pokémon in question. In the printed and onscreen media for Pokémon, the entire Pokéball glows as the momentary switch is triggered and transforms the Pokémon into or out of an energy state that allows for the containment of the creature. Unfortunately due to the current laws of physics, creating an orb that can transfer items into and out of the 4th state of matter is currently outside of the scope of this project. We hope as a group to revisit this feature in the future, and hope that innovation has provided for safe usage in a commercial environment. Until the change in natural law, this aspect of the Pokéball must be overlooked and proceed to delve into the color choices and why the shapes are presented in the form they are.

The Pokéball base item that is designed and produced for this project has seen various forms throughout the various media that Pokémon has been presented in. The early printed literature for the world of Pokémon presented the base Pokéball in a similar fashion to that of the figure above, but featuring slight variations that inform differing design decisions. On-screen media such as the numerous television series and feature films based in the world of Pokémon show a near identical Pokéball to that of the figure above. The beginnings of Pokémon were found in the video game industry, and feature a base Pokéball that has been presented identical to the one featured in the figure above. With the television and video game produced Pokémon media being the most recognizable (as well as the origin), the design for the item itself was then proceeded with grounding in the representation found in that media source.

With the custom PCB having to fit into the interior of this Pokéball, a design decision was made to differ from the world of Pokémon that is presented in the television media. In that form, the Pokéball itself can open up and the internal mechanisms that tightly cling to the interior walls can be viewed in various episodes. The hinge mechanism that allows for the upper red half of the shell to open away from the lower white half of the shell are never shown. In this media form, it appears as if no hinge is actually visible. This causes for a few differentiations from the world of Pokémon that are required to see the realization of this project in a safe and reliable manner. Those major differenations are:

- The shell WILL NOT open on a miniscule hinge function as presented in the television series'
- The shell WILL open in a different manner to allow for servicing as seen fit throughout the operational and troubleshooting phase of this project
- NO visible internals will be designed in high fidelity as to not impede on the scale restraint for the custom PCB that will fit interior to this item

With these major considerations kept throughout the design phase of this shell, the next milestone in the Pokéball shell research and design was to decide how the shell was to be realized to meet requirements defined by our scope.

5.3.2 Material Research

When beginning material research, one guaranteed factor was the necessity for this material to be manufactured through 3D printing means. The required shapes and configurations of the shell are not available through off-the-shell products, so 3D printing is the default course of action to construct the assembly. When beginning to research the material choice for our 3D printed assembly, major factors played into our comparative approach to understanding what the best filament/material of choice was for us. Some of those considerations were:

- Strength: to not break under immense amount of pressure while being held in the hand of the guest
- Flexibility: to ensure the item stays rigid at all times
- Durability: to ensure dropping the assembly, jostling in a backpack, and overall shell can hold up under testing procedures
- Warping/Shrinkage during printing process
- Availability
- Price

- Conductivity
- Safety
    - Non-toxic
    - Sustainable
    - Non-slip

These major items helped in informing our decision on the proper material choice. With many differing choices on 3D printable filaments, a table describing the breakdown and pros/cons of the materials follows with a segment on the final decision promptly following:

Filament	Strength	Flexibility	Durability	Warping	Lead Time	Cost per oz	Safety
PLA	Med.	Low	Med.	Min.	1 Day	$0.54/ ounce	Varies
ABS	High	Med.	High	Considerable	1 Day	$0.65/ ounce	Hazardous fumes
PETG	High	Med.	High	Min.	1 Day	$0.62/ ounce	Varies, slick when wet
TPE	Med.	V. High.	V. High	Min.	1 Day	$0.82/ ounce	Near rubber, highly safe
TPU	Med.	V. High.	V. High	Min.	1 Day	$0.82/ ounce	Near rubber, highly safe
TPC	Med.	V. High.	V. High	Min.	1 Day	$0.82/ ounce	Near rubber, highly safe
PA	V. High	High	High	Considerable	2 Days	$0.64/ ounce	Varies
PC	V. High	Med.	V. High	Considerable	2 Days	$0.99/ ounce	Highly safe

Table 15: 3D Printed Filament Comparisons

Filament	Pros	Cons
PLA	Biodeg.	Brittle.
ABS	High strength	Hazardous
PETG	Durable	Slick when wet and scratches easily
TPE	N/A	Difficulty in printing methods and slow print times
TPU	N/A	Difficulty in printing methods and slow print times
TPC	N/A	Difficulty in printing methods and slow print times
PA	High strength and durability	Requires high temp to print
PC	High strength especially impacts	Requires high printing temperature

Table 16: 3D Printed Filament Pros and Cons Comparison

When deciding from the above table on the final 3D printed filament to print our Pokéball, another major consideration is conductivity within the materials itself. With selecting from the above list, considering the molecular structure and possibilities of conducting within low-voltage circuits can hinder the communication through our transmitter and receiver. PLA has been known for being a conductive material due to the carbon structure that the material stems from. This as well occurs within ABS once paired with a conductive filament itself. With the room for error within PLA and ABS materials, it is fair to remove those from consideration. It is proper to keep the recurring notion in our minds that the PCB inserted into the interior of the Pokéball will be covered in all axes by this 3D printed hard shell assembly, and must be non-conductive as possible to ensure the proper communications can occur with our prototypes. PETG is another material choice that provides great positives, but the negatives for proper deployment provide reasoning to be wary of this choice. Ease of denting, scratching, and overall integrity issues can provide us with information on how this material can potentially be harmful for our prototype. TPE, TPU, and TPC are all highly flexible materials and do not provide the rigidity to the body that we are in search of with this project. This same issue occurs with PA filament as well. Thus providing us with the info that PC filament shall be the choice filament for this project. Though the price point at a per ounce basis is the highest choice, the quality and qualities included in the intrinsic composite informed us that this decision was the proper choice to 3D print all articles of the Pokéball assembly.

5.4 Hardware Updates

Most of the major changes that occurred through Senior Design 2 was the refinement of choices made to the hardware systems of Project Apricorn. As previously stated within the overall block diagram section of chapter 2, Project Apricorn consists of two major elements in hardware.

- Pokéball
- Base Station

These two elements consisted of a proto-board for the Base Station, and a designed RF PCB from JLCPCB embedded within the Pokéball. The Pokéball PCB schematic and board layout was designed within KiCad, sourced through DigiKey, JLCPCB, and Mouser Electronics for the required components. The chosen controller for the PCB within the Pokéball was the Texas Instruments CC1310 Sub-1GHz MicroController. The specific packaging was determined to be the 48-pin package with 128KB of embedded Flash memory.

The PCB itself comprised 7 major sections. These sections attributed to an overall assembly that was designed with ease-of-use by guests, as well as ease-of-access for back-of-house (BOH) maintenance to occur in the chance that our assembly errored. The major sections of the PCB embedded within the Pokéball were the RF impedance matching circuit, balun, and antennas. The hardware of the Pokéball PCB was programmed via pin-headers attached to the appropriate pins on our CC1310 microcontroller unit. The battery unit utilized within the hardware of the CC1310 was a lithium CR2032 coin-cell battery at 3.0 Volts. This was then boosted and regulated at a specified 3.3 Volts by a Texas Instruments TPS61291DRV voltage boost regulator.

The housing was still designed utilizing Fusion360, printed via 3D-printing in Polycarbonate filament. The finalized housing was then sanded, primed, and painted as necessary. The overall assembly changed minimally, with only instances of major change occurring on the mounting of the PCB to the 3D-printed Pokéball via the use of 6 mounting holes via M4 sized screws.

6.0 Software Research

The following section in our report we showcase the outline of our intentions for the software aspect involving the Pokéball. It will validate the reasoning behind the choices we made regarding the software. This section will clearly demonstrate the difference of possible programming languages that could be utilized for the Pokéball. After demonstrating a clear difference between the potential programming languages, we were able to determine the best suited language to be used for our project. The following sections will describe the best way to implement the integration of the RFID tag/card and the related database programming. RFID programming is an integral part of the Pokéball. A proper understanding and description of the RFID will also assist in understanding our intended software flow. The software flow will demonstrate our intended layout of the program and how we plan to configure the Pokéball for our project. As this is a program with an intended public audience, we have also decided to implement security features. We have highlighted the various security features we have decided to implement and describe the requirements for said security features. Some additional security features will include a tier-based admin system and location-based login. These are preemptive features based around our project's intention of the Pokéball being a part of a larger Pokémon theme park. In addition to those preemptive measures, we further described predetermined possible error codes. The final section of the software research will describe our intentions for the frontend user interface and the methods related to it. It will describe the requirements we expect to uphold to ensure a good user experience.

6.1 Programming Languages

Programming languages that would be best suited for this project at this specific scale are C and C#. Equipment like microcontrollers, smaller grade PCBs and RFIDs do not require high level languages for the technology to function.

Why not any other languages? Embedded projects and applications typically need a quick runtime. This project would require simple tasks from the actual code like retrieving, storing, and displaying data. Object Oriented (OO) languages such as Java could potentially slow down the device. It can also lead to other internal issues because OO programming requires larger memory space than our hardware can possibly handle.

For our front-end facing UI this will require languages software that include (but not limited to) HTML5, CSS, and Javascript (JS). Ideally, it would be best to have a UI that is not too layered/complex which is why unless it is absolutely necessary languages like python will not be included in the mix. Incorporating Javascript will also give the space to incorporate some security measures which is a good practice for any developer to use.

Language	Best Case Scenario
C	Embedded system drivers Machine-level hardware
Javascript	Web page interaction, Creating web/mobile applications
HTML	Web app display
CSS	Stylinging and Layout of web application pages
PHP	Interaction with databases
React JS	Solely Web application User Interface with non-static elements
React Native	Solely Mobile User Interface
Python	Backend, Javascript, and HTML are all built in together
Swift	Mobile application
Angular	Single Page application
Vue	Similar to React, but also best for a single page application
Flutter	Mobile Application that desires a flexible UI
Jquery	Web application that requires light Javascript

Table 17: Programming Language Comparisons

6.2 RFID Programming

For a successful implementation of an RFID tag/card, there must be a successful programming of the RFID reader which is projected to be in or on the Pokéball shell. Each tag needs to be assigned an ID which will essentially be the ID of the player. When using the tag to access any system, the algorithm must be able to find the existing tag ID within the system and then move on to “if” the ID exists, “then” assign this Pokéball. Another useful practice would be to also implement a program that is able to read foreign/stolen RFIDs and freeze the ID with the same “if, then” algorithm. The figure below displays a general software flow of the RFID.

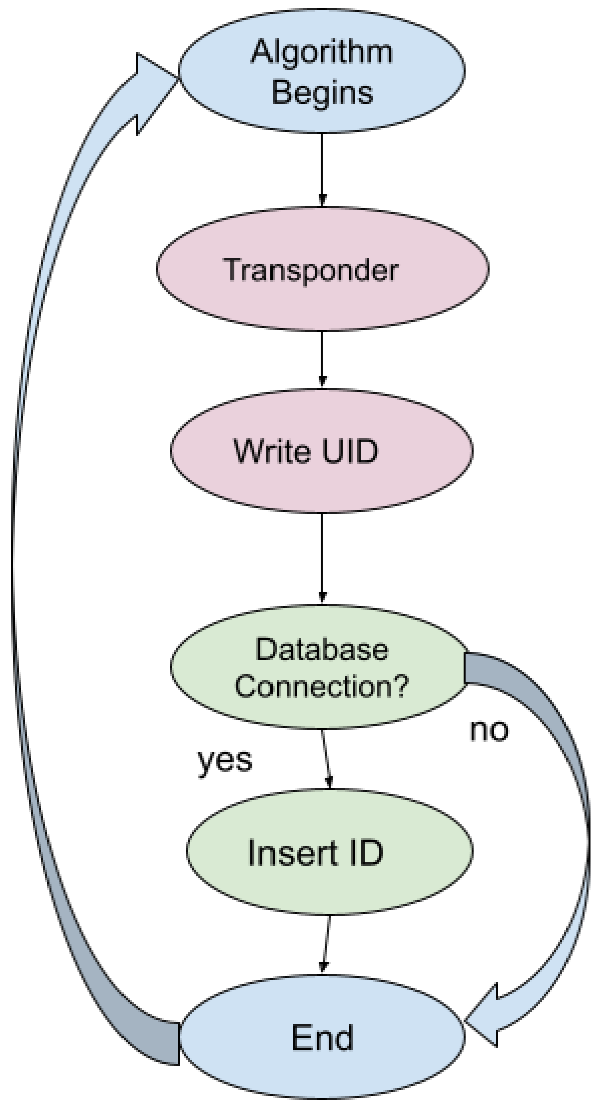

Figure 28: RFID software flow

6.3 Software Flow

For configuring the device for the necessary aspect of this project, we must first establish a database in which the pertinent information and identifiers will exist. The figure shown below is a flowchart regarding the necessary steps regarding configuring the Pokéball and Pokémon. This step in the software will require the assistance of SQL programming to properly integrate the database.

The database will hold important information about the Pokéballs, including but not limited to the unique guest ID, unique Pokéball ID, unique Pokémon ID and the overall Pokéball score.

The program will first distinguish between three options the user could potentially make: new Pokéball configuration, Pokémon configuration, or updating scores. New Pokéball configuration is only meant for initial activation by the supervisor. This will start the connection with the RFID and continue to record statistics. It will also store the guest ID and the Pokéball ID during this initial step. The secondary option is for when the user “catches” a Pokémon. This will connect the chosen Pokémon ID with Pokéball ID. After the Pokémon is connected, it will not allow for another Pokémon to be connected. Only one Pokémon can be connected to the Pokéball at a time. The last option is only to occur after the first two have been configured. The Pokéball will store statistics of the Pokéballs and the Pokémons. As the user interacts with different activities, it will collect points. The database will store the individual score of the Pokémon and the overall score of the Pokéball. This information will be stored in the database and will be accessible to determine any pertinent information relating to an individual Pokéball or Pokémon.

Figure 26 depicts a flowchart of how the processes of database connection will be established and what data will be stored in the database. It also shows the three options for the Pokéball to take with every action. The flowchart shows what steps will be taken for each option. It accurately depicts the overview flow of what was described above. This is what we utilized when designing the database for the Pokéball. In further sections we determined what will be the best option to use as our database is meant to hold a significant amount of data with the possibility of constant updates.

There were other possible approaches to the flow of the software. We have chosen to proceed with this flow because it is the most straightforward in our opinion and allows for the least amount of mistakes. This allows for the simplest method of inputting relevant data into the database. The database will be accessible to view all of the data by the admins.

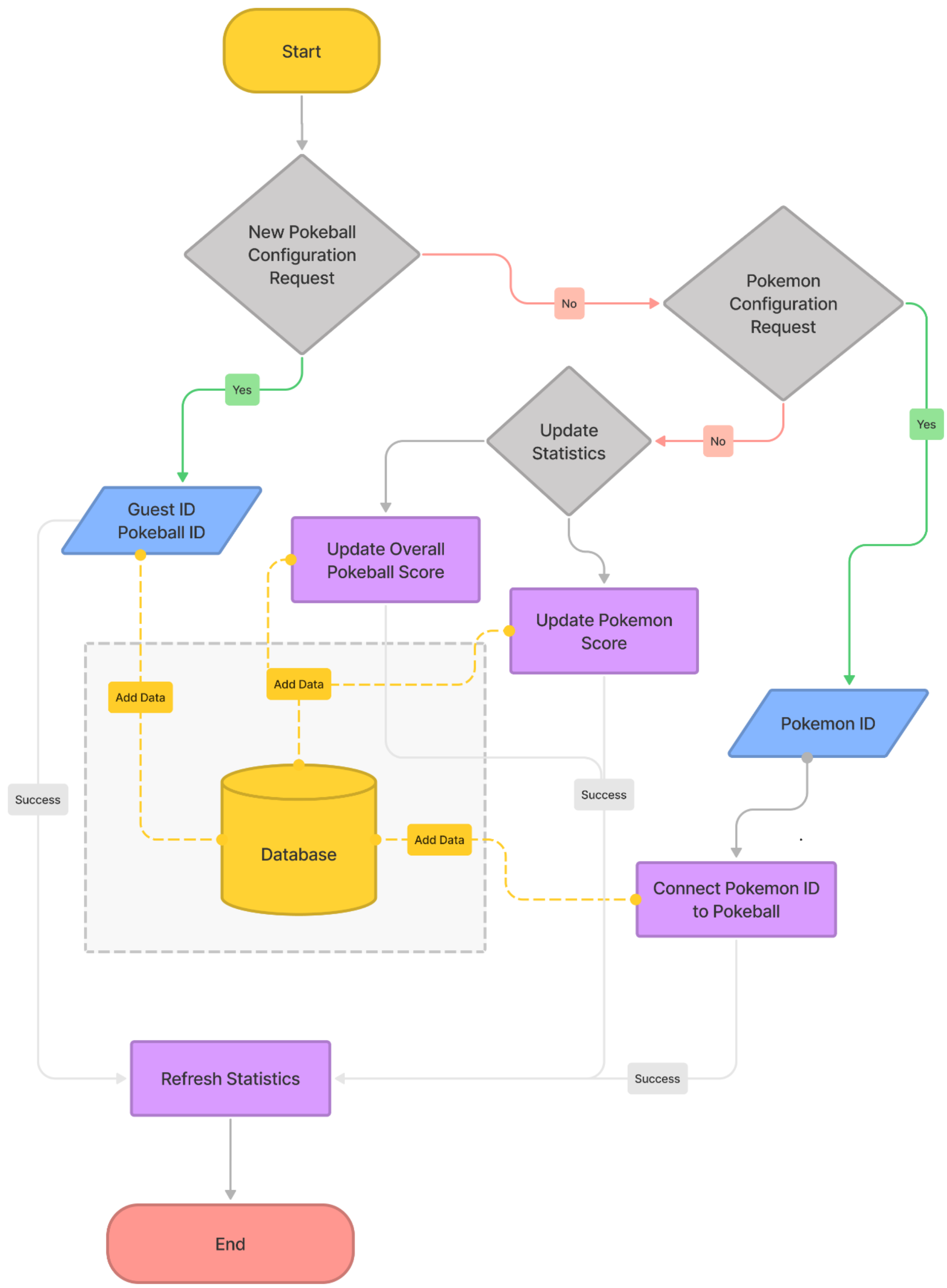

Figure 29: Backend Software Flowchart

6.3.1 Updates to Software Flow

The project followed a similar flow to the one we presented previously but the figure below shows a more accurate depiction of the software flow that was implemented. This presents a clearer flow of actions that a user can take to use the program for the Pokeball. As shown in the figure below, depending on the type of user, a new user must register then the user, regardless of new or existing, must login to access the various features of the program. Once the user has logged in, they will have access to the Pokedex, where they are free to view different Pokemons and unique attributes about each one. It also shows the primary keys that are necessary for the database and where they connect within the program. The symbol of the Pokeball depicts where the hardware connects to the software and from there the user can begin the battle sequence. After the user completes the battle, the program will automatically update the leaderboard and statistics. The user can then check the leaderboard after the battle is completed.

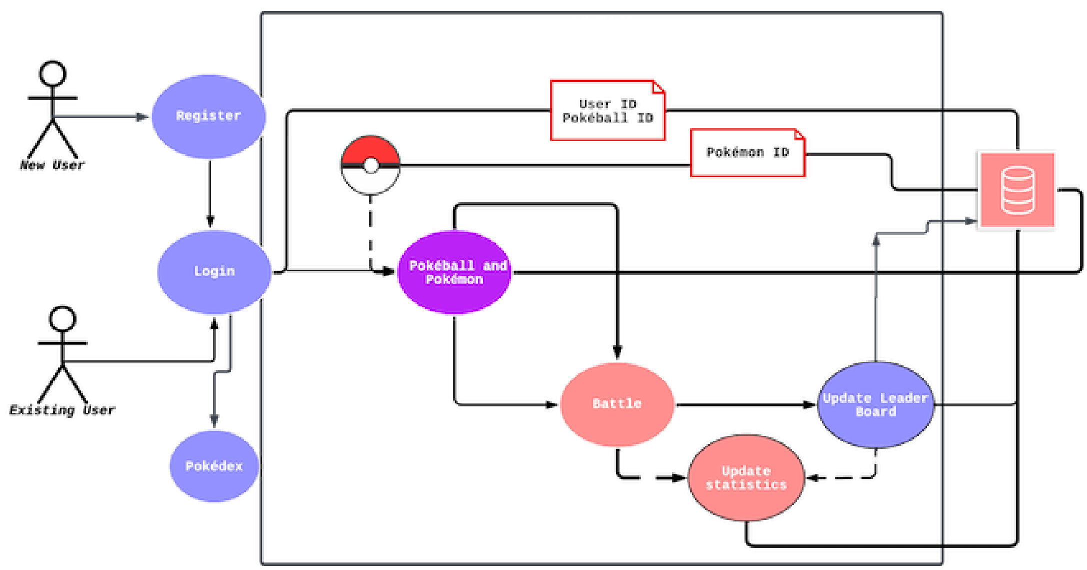

Figure 30: Updated Software Flowchart

While initially our plan was to enable a wide range of javascript code, some of this implementation was not possible. Most of this is due to the framework compatibility. It would have needed to be programmed in Angular.js or React.js in order to function accordingly. Also due to time constraints, we were not able to program as many security features as we would like. This would include features such as:

3d Pokémon Page	Required multiple dependencies. Not Compatible.
Video Landing Page	Required multiple dependencies. Not Compatible.
Custom Audio	Required multiple dependencies. Not Compatible.

Table 17.1: Removed Software Features Chart

As seen from Figure 30. The main components remained. In addition to the main components, the Pokédex was also one of the newest features. This is meant to replace the 3D landing page, while also displaying multiple individual Pokémons along with their characteristics. The Pokédex works by displaying a multitude of Pokémons with a combined amount of API'S instead of calling one API endpoint at a time. This saves time and storage capacity.

6.4 Software Security

From a security perspective, there needs to be a few components placed into our program to not only ensure that the PokéBall software is safe, but also that the user/admin information is safe as well. A couple of software requirements that can be implemented into the design could be (but not limited to): 2 Factor authentication, minimum password length, restricted and tiered admin, and location based login. In order to have full functioning and seamless security measures the front-end UI will have to be coded in Javascript (JS). This should have no effect on the Pokéballs itself but only make the user/admin experience easier.

6.4.1 Minimum Password Length

Minimum password lengths are quick and simple to implement. This security measure lowers the probability of attackers guessing passwords and having access to restricted information. The code itself should not go beyond the HTML layer. In HTML, there are certain parameters that can be placed on input form. For an example if there was a need to have a 16 digit passcode (in which in 8 digit will be fine) then the input tag would look to something like:

```
<input type = “password”
minlength = “16” required>
```

The required tag will trigger a warning message to the user informing that they should be using a 16 digit password. There is also a technique to implement password requirements in JavaScript, but for this project it will not be necessary since we want to keep a balance between programming languages and not make anything more advanced than it needs to be. We also don't need to stop at just the password lengths, but the type of characters that are being put into place. A lot of common requirements can be but not limited to: a mix of numbers and letters, special characters, and lower/capital case letters. Specific password requirements are also a new attribute that can easily be done in HTML, the syntax would just have to look like:

```
<input type = “password”
minlength =“16”
required passwordrequirments = “required: upper; required: lower; required>
```

6.4.2 2 Factor Authentication (2FA)

2 factor authentication is a common practice when it comes down to securing information. While it is not bulletproof it does add an extra layer of security. Because players will not really have access to UI themselves, the 2FA will mostly be directed towards admin. While also considering that the whole design will most likely be programmed in a low-level language, the best case scenario of having 2FA will be generating a random pin that the admin will have to input before accessing any sort of data. To implement this, we had a general login data that is pulled from the database (normal login) and in between the layer of the logged in page and actual UI, there will be a place to input a 4-6 digit pin. Ideally the 2FA portion will most likely have to be coded in Javascript. The key will be to keep any front-end facing code away from the hardware itself, that way it will not mess up the functionality of the RFID and Pokéball. Javascript is also able to implement email verification, so in order to add a seamless transition between pin and input, the code itself can generate a random pin that will be sent to a work email in which the admin can access. The header to achieve any kind of email validation in Javascript would be “email.js”. This header contains the file that holds all the necessary components needed to. Email.js is mostly used for email validation which is necessary if we are going to incorporate any kind of email structure. With this being said, the code itself must go through an algorithm that checks if the email is within the right syntax and also if it even exists.

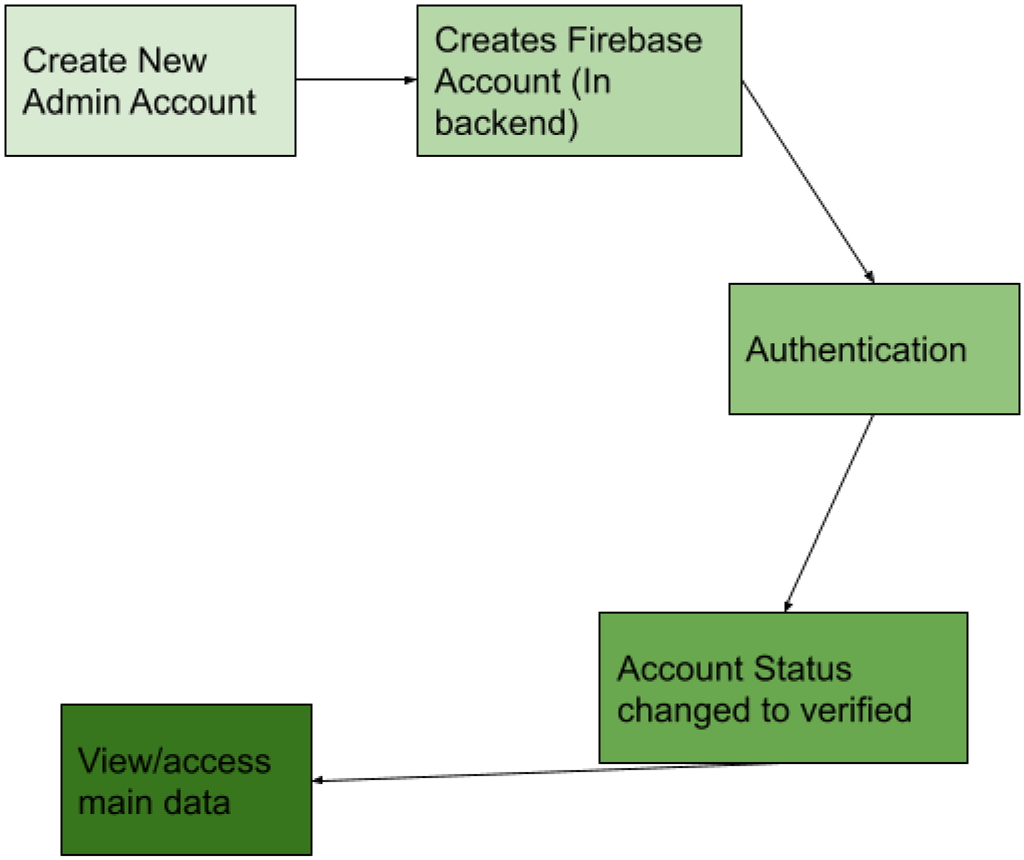

Figure 31: Authentication Flowchart

Dependencies for email verification should be installed into the packages as well. Firebase is a free tool accessible to developers that will take out any complexities to send an email authentication with JS. In the code itself a function will be created that will call Firebase to create a new profile that is provided from the user input. When Firebase has successfully created an account, it will send a link to the email requesting that it authenticates the email. This will change the status of the admin's account and they will be able to access whatever they have permissions to access. While it sounds as if the software flow is becoming complex, these are all simple components that will not weigh down the program, just added security measures.

6.4.3 Tiered Admin

In most entertainment and technology industries, admin are typically designated specific roles to limit misuses of the technology. Our system Ideally should contain a minimum of three roles labed: gray, red and gold. These roles will be established and designated into our database. When an admin with certain permissions access the frontend facing UI, any changes that they are to make will either be inaccessible or unsavable. An example of tiered admin will be displayed below.

	Gray	Red	Gold
View Scoreboards	✓	✓	✓
Update Scores	✓	✓	✓
Edit Code			✓
Block users		✓	✓
Delete History			✓
Create User	✓	✓	✓
Delete User		✓	✓
View User Information	✓	✓	✓
View Game Status	✓	✓	✓

Table 18: Tiered Admin Chart

6.4.4 Location Based Login

Location based login will be a feature used on the UI to ensure that no one who is not currently in the park or near the Pokémon attraction will have access to any on-going battles and players. Even the admin will not be able to access the software if they are not on the grounds of the attraction. A possible way to set up this security measure would be to grab the geolocation at the time of user login. The way that geolocation works in JS is similar to email.js in the sense where it takes the longitude and the latitude of an ip address and it returns the location based on those variables. A simple technique would be to define a range of longitude and latitude pairs and compare the logged in geolocation to anything that is in those pairs. If it falls outside of those ranges, then the program will not allow the user to successfully login. These methods also can be used to find time zones' so in the case that an attacker from another country or even a state like texas, then the UI will still be inaccessible if the comparison does not match. It may also be more accurate because it can be an added measure to confirm if the geolocation is in fact what it pulled up to be. A concern may be usage of a VPN, but in a hypothetical scenario that this Pokémon is integrated in a park like Universal, then the cyber security resources that can be provided will be much more advanced that hardcoded security measures.

There is also another technique to enable geolocation detection in javascript without the use of longitude and latitude variables. HTML5 / JS has a header called geolocation. When you call the attribute within the code, it should be able to pull the user ip address location. Below would be the pseudocode to (in theory) execute this action.

```
Function getlocation ()
 {
nav.geolocation.getCurrentPossition(x, y);
}
```

After the program has received the geolocation or at least attempts to, the program should then run through a few switch cases that hold error functions/codes to ensure that the programming is not pulling user location for no reason and letting them through. Error codes will range from what is shown in 6.4.5.

Geolocation functions and implementations should be supported by most commonly used and compatible with most browsers. But the amount of support varies from browser to browser. In the table below, most of the mobile browsers have a lower range of support than those of desktop web browsers that are widely used. It will most likely bring up the question if mobile web access should be blocked/restricted in order to ensure less attacks/breaches. These browsers would include but are not limited to:

Browser	Support Rank
Chrome	High
Edge	High
Firefox	Medium
Internet Explorer	High
Opera	High
Safari	Medium
Chrome Android	Medium

Safari IOS	Low
Samsung Internet	Low
Opera Android	Medium

Table 19: Browser Compatibility

6.4.5 Error Codes

In addition to ensuring that there will be added security measures, we needed to understand what error codes will be in place so that both admin and user understand why they may run across issues. These error codes are also supported by Firebase which will be used to authenticate emails. Now what normally happens when a request is made (example: entering the domain of a website to access content) a status code HTTP 200 is returned. Some do not know this because it is typically not visible to the user. But When a request is made, and something has gone wrong on either end point, a status code of HTTP 400-500 is returned back to the user. This is why one may receive the common “Error 404 Not Found”. These error codes will be implemented in either security component, whether passwords, email validation, and geolocation data. Ideally these codes should not be so easily displayed to a general client with basic permissions. The reason for this is so that any attacker or foreign client will not be able to easily manipulate the software. Remember that the UI is intended to be simple and of course any advanced hacker or developer may find loopholes, but as long as there are key features put in place, then it should deter any spam users from abusing the software.

Error	Meaning
ALREADY_EXISTS	Request has an existing duplicate
INTERNAL	Firebase server has returned an error (check firebase)
INVALID_ARGUMENT	Invalid parameter, or input
NOT_FOUND	Request has attempted to edit or update a page or component that does not exist
PERMISSION_DENIED	User does not have access to make the request

UNAUTHENTICATED	Request does not have the appropriate credentials to move forward
TIMEOUT	Request has exceeding time to appropriately move forward
UNKNOWN_ERROR	Unexpected and unknown error
HTTP Request Error Codes	
404 Bad Request	Request could not be understood by server
401 unauthorized	Un-authentic or incorrect client information has been imputed and requires authorized credentials (useful for admin roles)
403 Forbidden	Also means that client does not have the permissions to access/view content
404 not found	Common. Servers cannot find what was requested.

Table 20: Error Code Descriptions

In the same breath, there are many http success codes that have an option to be displayed from a UI perspective. Many choose not to show them, but for the purpose of this prototype, it will be useful to display successful HTTP requests and responses that are happening. Below shows a table of successful codes. Keep in mind that success codes do not necessarily mean that it has done exactly what the client has requested it to do, but can also mean that the request has been able to pull elements and content from the document. This could be anything from headers to bits of information about the document.

Code	Meaning
200 OK	Request succeeded
205 Reset Content	Reset Document/Page

202 Accepted	Request Received but still processing
206 PARTIAL CONTENT	Request has received partial
204 No Content	Successful request and no content required to be displayed
202 Created	Successful request and new content has/will be created
207 MULTI STATUS	Multiple actions have happened/ or they are happening
208 ALREADY REPORTED	Same resource has already been mentioned

Table 20: Error Code Descriptions

6.5 Frontend UI

In the above sections we have spoken a bit about how the Admin UI may operate. Here we are going to discuss exactly what that may entail from a front-end perspective. Ideally the goal would be to have a “portal” where workers/admin can access to not only keep track of scoreboards but also where information can be updated. In addition to admin access, the portal will contain a simple front facing element that can be displayed to players (the landing page). The landing page cannot be manipulated by players simply viewed.

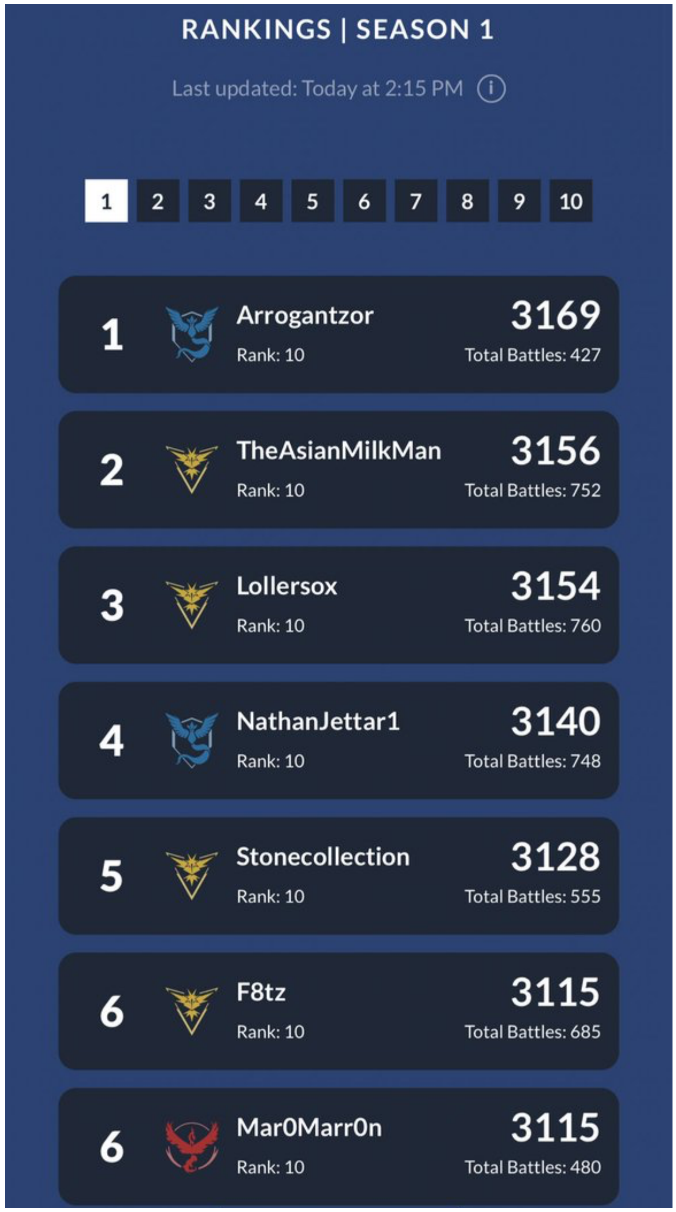

Figure 32: Sample Portal Page

The portal will be on the internet. In a real life scenario, this portal would be an extension of a theme park's entertainment web application or mobile application . For now, our portal will simply be around a 2-3 page web application.

There are many approaches one can take when it comes down to building a web application. Top stacks would include:

Framework	Definition
MEAN	MongoDB, Express, Angular, Node JavaScript
LAMP	Linux, Apache, MySQL, PHP or Python
MEVN	MongoDB, Express, Vue, Node JavaScript
SERVERLESS	Built on cloud, AWS most ideal
ROR	Ruby on Rails - Uses HTML, CSS, JavaScript, XML and JSON
Flutter	Mainly Dart
Django	MySql, Django, Python, JavaScript

Table 21: Error Code Descriptions

The best method however that I see fit would be to create a MERN stack in which could be coded in pure JavaScript. Not only is it easier to create simple applications, but also a framework that is familiar amongst the computer engineering class. Other stacks might involve an unnecessary amount of 3rd party applications to achieve our goal so a MERN may have to work.

6.5.1 MERN

What is a MERN stack/application? It is a set of tools placed together to create a dynamic program. The basis of the word MERN represents: MongoDB, Express, Node, and React (JS). This framework is indeed considered a full stack.All of these components can come together to create a user friendly application. The bulk of MERN stacks are based on javascript which can allow for a wide range of components to be implemented in the UI. Not only is it effective but it is also more widely used in today's technology. This framework makes web applications flow more smoothly and it is also capable of being light in program (depending on the application). It is also very cost effective which can allow for a low-risk solution for theme-parks and companies. Specifically in our case, it will not affect the project budget substantially. But with this being cost effective, developing a MERN stack will also allow for the application to not be spread out in too many environments. Especially from a developers perspective (when working as a group) it is essential for the application to be accessible and readily available to be edited. Visual Studio code offers a great environment for at least 3 of the letters in MERN to exist. That means that MongoDB will simply have to be an area that is created and manipulated elsewhere, which is fine because most database spaces are.

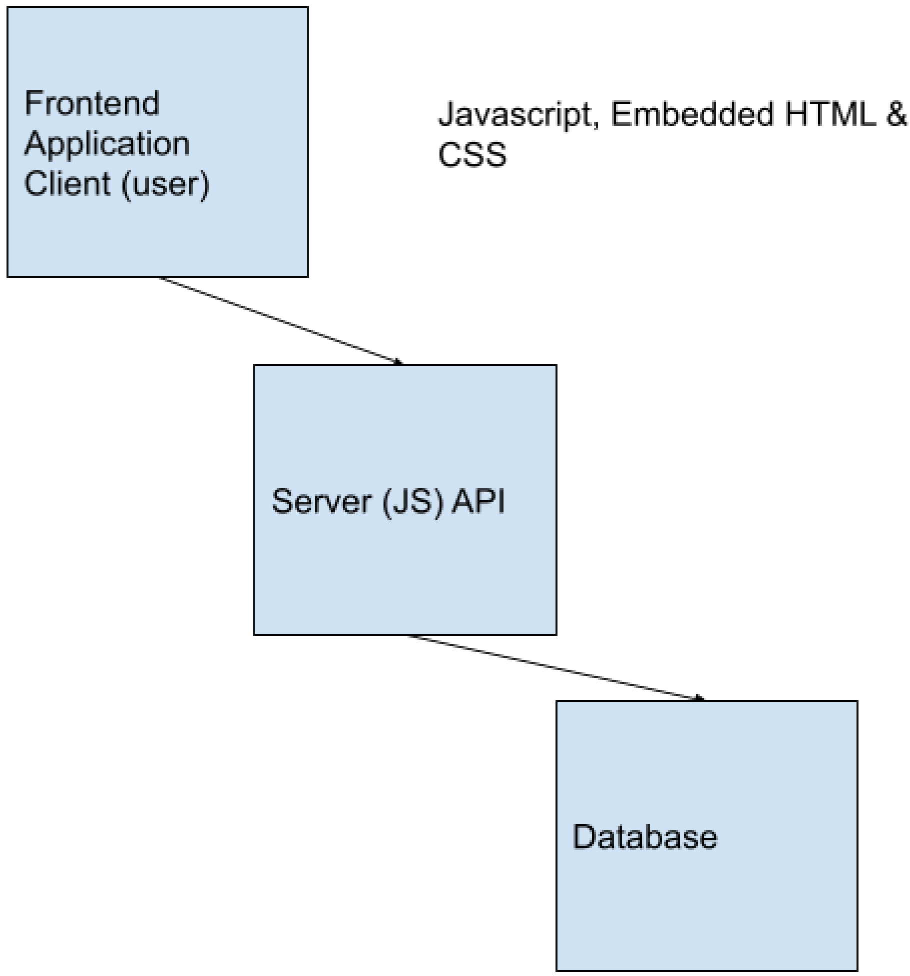

Figure 33: MERN Flowchart

6.5.2 MongoDB

MongoDB is a type of database that is considered NoSQL. Earlier we have discussed implementing a database with my SQL, but now we may come to notice that MongoDB is not only easier to implement but also is able to hold a large volume of data. This may not be necessary now, but in a theme park where 100's of guests visit daily, it will be important to ensure that there is enough space allocated for the max capacity of users.

While MongoDB is still in the same family as SQL, MongoDB is considered relational. MongoDB is intended to be faster and more energy efficient. MongoDB is also capable of holding a large amount of data. Not that we in actuality are in need of a large database, but considering that this is meant for a theme park, we should make room for any potential users/players. This is not meant to say that one is better than the other because they are both widely used databases amongst the energy. One just has to evaluate the attributes that they consist of to make the appropriate decision. Below you will see a comparison between the two.

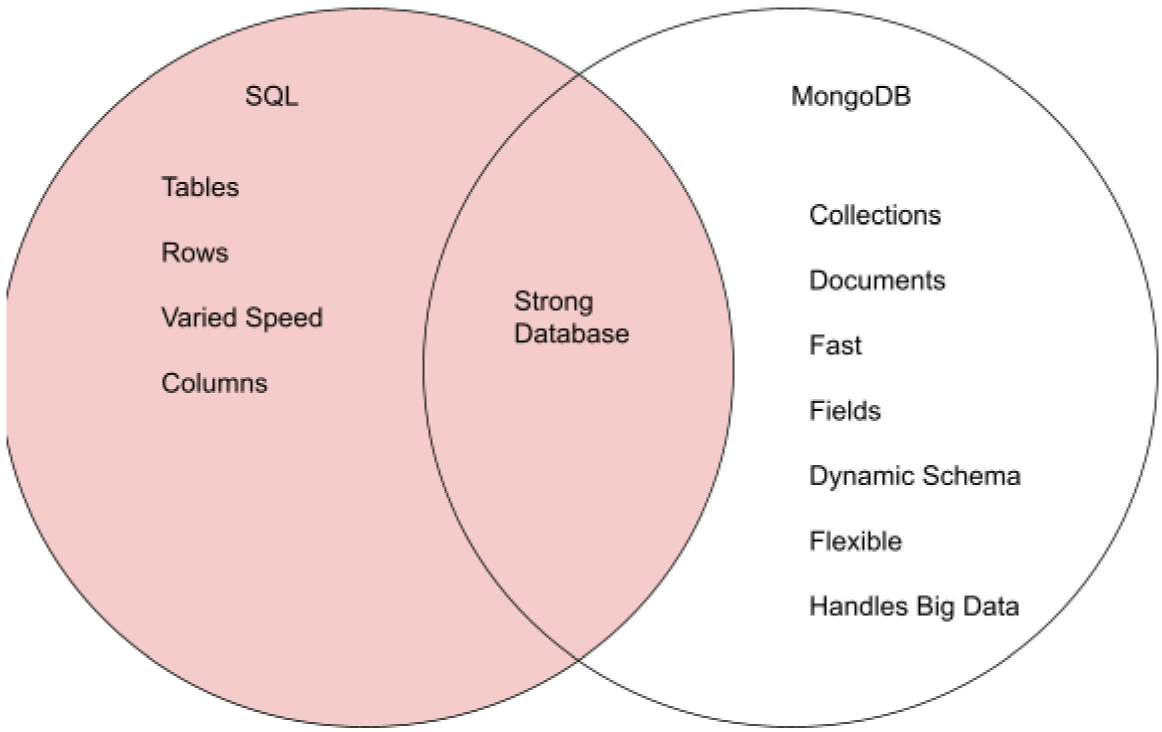

Figure 34: SQL vs MongoDB Venn Diagram

6.5.3 Express

The best way to describe Express in MERN will be that it is middleware that takes in requests. It is considered the backend of an application and typically will be seen as the server if you were to build it on something like VS code. This portion of the stack is also responsible for sending and returning the HTTP requests. A lot of popular apps such as Twitter, Uber, and IMDB have transitioned to use Express, so from an enterprise perspective it seems to be a reliable tool.

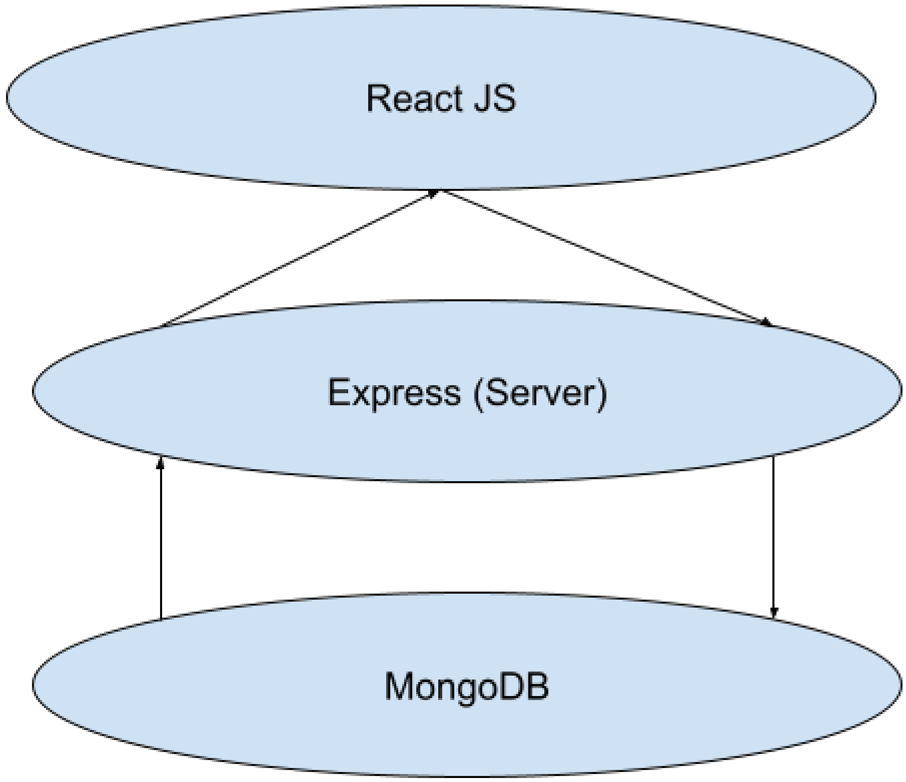

Figure 35: Express Flowchart

6.5.4 React JS

React allows one to build dynamic applications on the frontend. It is largely responsible for how it looks and not so much how it operates on the middle and backend layers. React Js allows for minimal code, whereas other stacks and frameworks will require at least 2 pages of code to create one page web application. As said before, a dynamic application is not the end goal of this project; however, we would like to have space to integrate it to something more complex if needed.

React JS is interesting in a way that it does take a bit of effort to run and even sometimes, if it is not connected to a fast system (say you went the cheap route), it can even be slow launching it on the first run. This may bring up concerns, because in a theme park scenario, or even in a senior presentation, the last thing that anyone wants is to have a crashing app.

React also does not normally look like regular code, it is multiple codes (Java script, HTML, and CSS placed into one which may cause some confusion if one has never done it before. On some HTML pages you may have seen some methods where they use a <script> tag, but React is different in the sense that it actually uses functions, very similar to c, that will be later called in the HTML portion. Below is a Hello world example.

```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<h1>Hello, world!</h1>);
```

Or even:

```
function helloWorld(props) {
  return <h1>Hello, World </h1>;
}

ReactDOM.render(
  element,
  document.getElementById('root')
);
```

Furthermore, React JS can have its flaws when solely used. React JS is a high level development language, meaning that it will consistently require changing and updating the software to function. Unfortunately, by the time we are ready to program our design, there will probably already be significant updates to the software that were needed to be implemented.This is also not considering that React (at a minimum) has very large libraries that could be useful if one is willing to learn all of them, but time will come to be a constraint. In an ideal world a simple HTML page with underlying JS would suffice, but our project consists of elements that are not static.

A fear will also be the runtime between the Frontend facing project and our physical design. It is important that it is able to have a seamless transition and even from our own experimentation React can increasingly become inconsistent and unreliable as more components are added. This is why it is important to have a program that is small and does not contain too many moving elements.

But it is not all a disadvantage, React JS is notorious for having components that can be reused and recycled. If the design is taken a step further, similar components can be used to create a front-end facing application. Now of course this would impact that project cost substantially and even increase the amount of working hours but it is a possibility for any future sponsorships. In the figure below the functionality of React Js is displayed. JSX will mostly determine what the application will look like. This means that JSX is responsible for the embedded html which makes it easier to write code without the need to separate the javascript from the visible components. This is still Javascript, but it is in a different syntax. Redux assists with binding the react with the use of libraries and APIs. This typically comes with the React packages and a lot of developers may not even find the need to see, alter or change it. Redux will also be responsible for applications being able to run in different environments. Very similar to C and any other programming languages, Redux sets the basis and rules for testing and running.

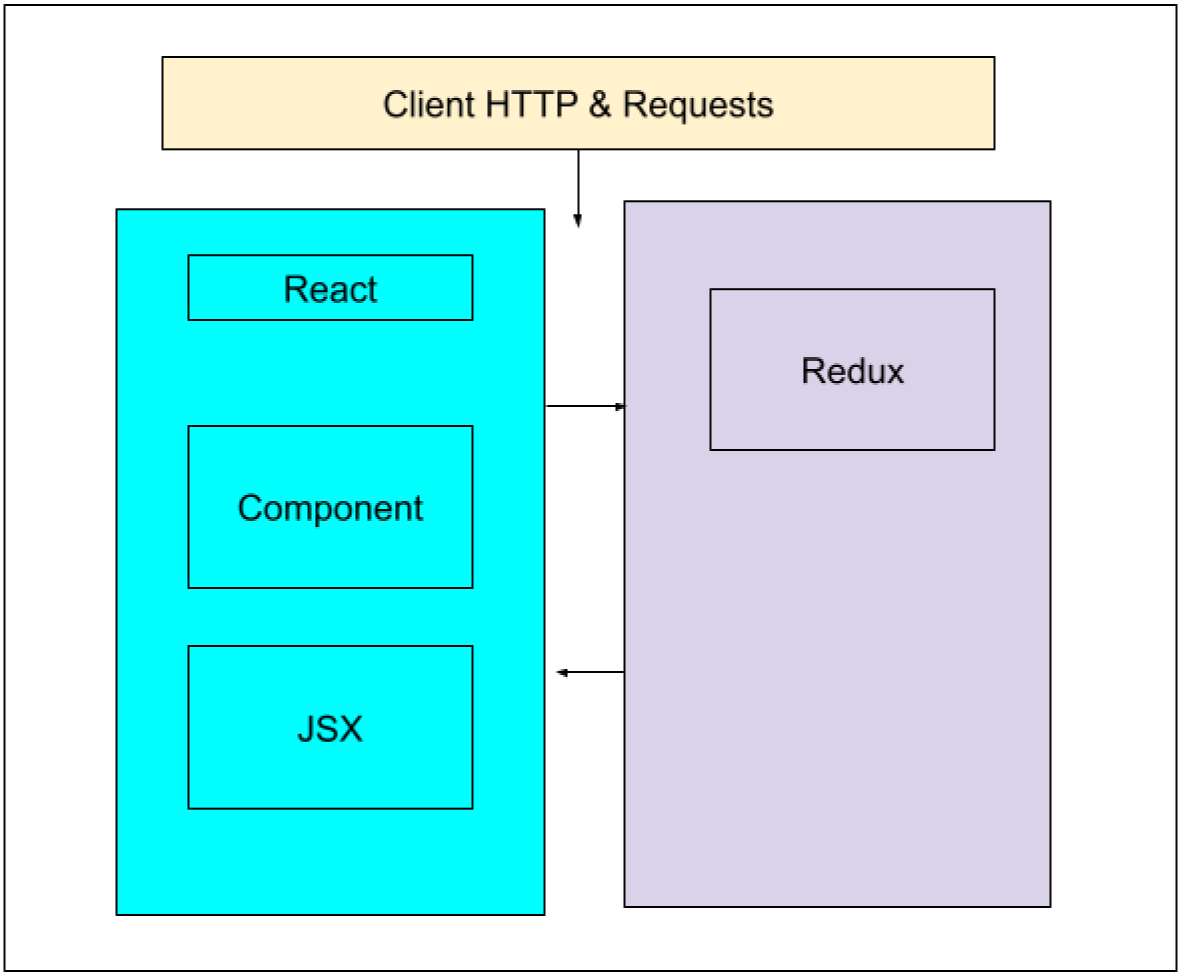

Figure 36: React JS Flowchart

7.0 Prototype Construction and Coding

This section will demonstrate the design phases that will be used as a stepping stone to the final prototype build. It will show how the microcontrollers will be connected to the RFID reading modules and how the Pokéball is designed using Autodesk Fusion 360 software and eventually be 3D printed after the initial STL file is created.

7.1 Project Hardware Assembly

The objective of the project hardware assembly is to include two different ways to implement the RFID reader as it depends on which microcontroller is suited the best.

7.1.1 Assembly Using MSP430

The MSP430 will be utilizing the EM18 RFID reading module.

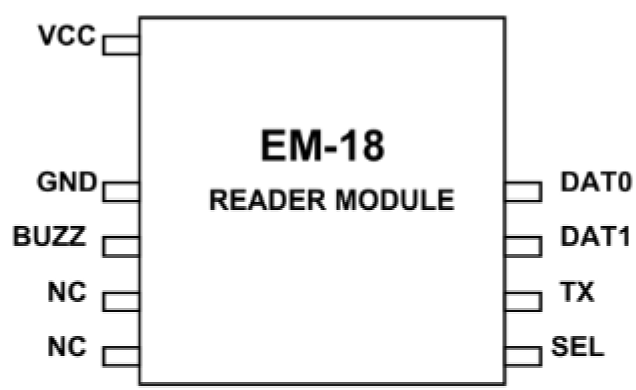

Figure 37: EM18 RFID Reading Module Pin Layout

The chart below describes the pin layout of the EM18 and how it would be configured to be connected to a MSP430. Setting the SEL pin to either 1 or 2 will allow the communication to be set between RS232 (serial communication) or WIEGAND.

Pin	Description
VCC	Connected to power source (5V)
GND	Connected to ground
BUZZ	Connected to buzzer
NC	No connection
NC	No connection
DAT0	WIEGAND interface HIGH pin
DAT1	WIEGAND interface LOW pin
TX	Data is outputted of RS232
SEL	SEL = 1 then using RS232 SEL = 2 then using WIEGAND

Table 23: EM18 Wiring Description

7.1.2 Assembly Using Arduino UNO

The Arduino UNO will be utilizing the RC522 RFID reading module. The assembly of the RFID is based on the schematic below. The schematic below features pin connections in schematic form as well as following in tabular format a descriptive version acknowledging where the pins are connected. The RC522 RF module connected to the Arduino UNO is featured first. This schematic and diagram features the home reader/writer that is acting as our home base to implement code into. The second schematic features the fully constructed, tested, and verified breadboard testing that was completed during the Senior Design 1 process. This allowed for the group to test both the breadboard, RF units as needed, and the software developed during this phase of the project. Understanding recurring errors and developing further advancements prior to Senior Design 2 has assisted in clearing numerous roadblocks and design faults along the process. This step of the project will only assist in the further constraints and construction roadblocks that will undoubtedly appear during the Fall 2022 semester.

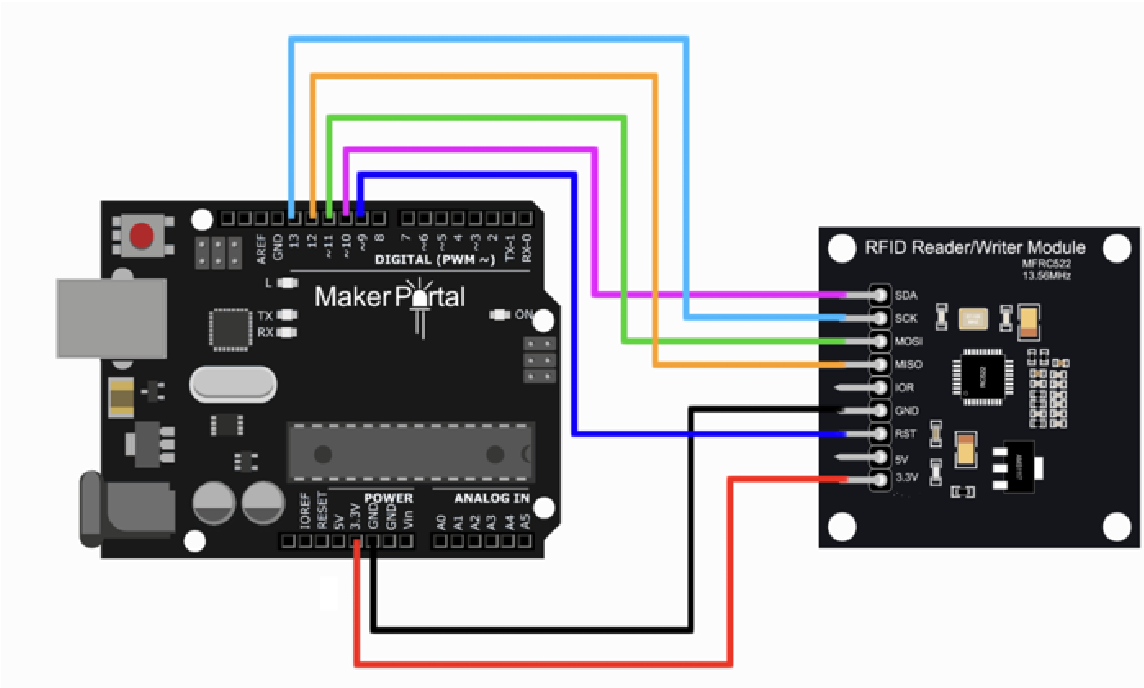

Figure 38: RC522 Module to Arduino UNO Schematic

Pin	Wiring to Arduino Uno
SDA	Digital 10
SCK	Digital 13
MOSI	Digital 11
MISO	Digital 12
IRQ	Unconnected
GND	GND
RST	Digital 9
3.3V	3.3V

Table 24: Wiring For RC522 to Arduino

After the RC522 RFID module is connected to the Arduino, the battery can be connected using a barrel jack connector. Then the USB can be connected to a PC where coding and the display will appear.

After testing that the RFID reader works, an LCD screen will be added as a testing parameter. A potentiometer is required with the use of the LCD screen because it will be used to adjust the brightness of the display so the text appearing will be clear. The figure below will be used as a reference when the breadboard is assembled.

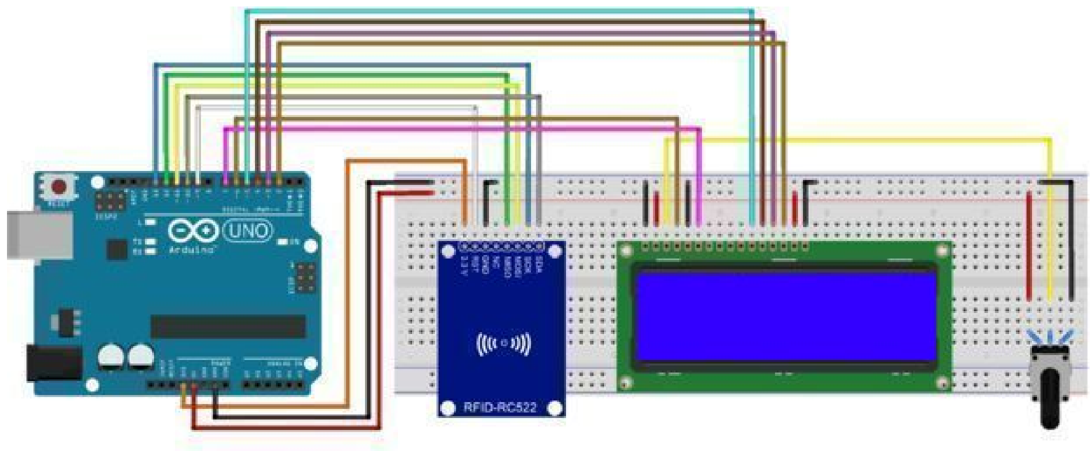

Figure 39: Schematic of Assembled Breadboard

The table below will be referenced when the breadboard assembly occurs. It will be needed to determine how to connect the 1602 LCD to the Arduino Uno.

Pin	**1602 LCD Pin**	**Connection**
01	GND	GND
02	VCC	5V
03	Vo	Potentiometer
04	Rs	Arduino Pin Digital 6
05	R/W	GND
06	EN	Arduino Pin Digital 7
07	D0	No connection
08	D1	No connection
09	D2	No connection
10	D3	No connection

11	D4	Arduino Pin Digital 5
12	D5	Arduino Pin Digital 4
13	D6	Arduino Pin Digital 3
14	D7	Arduino Pin Digital 2
15	A	5V
16	K	GND

Table 25: 1602 LCD Connection Guide

7.2 Printed Circuit Board Vendor

Printed circuit boards may be needed to be researched in the future as the design has not been created yet. Pricing of the board will be a deciding factor on how the board will be created.

However, certain components that definitely must be considered can be discussed so far. The PCB should have an integrated RFID module connected to it. In order to connect a LCD screen to the reading device, a display connection may be required whether it is an USB or HDMI connection. The final design has not been decided yet so it is unclear whether our group will create a custom PCB or develop an Arduino Shield setup as discussed in Section 5.2.4. The Arduino Shield will be an overall larger design, but it reduces the complexity.

7.3 Final Coding Plan

Depending on which microcontroller is used for this project, either the Code Composer Studio IDE for the MSP430 or the Arduino IDE for the Arduino microcontroller may be a suitable development environment.

Since the Arduino will probably be used as the choice of microcontroller, the Arduino IDE is the programming interface that will be used. The start of the plan will begin by programming the Arduino to allow the reading of a RFID key or card. Once the Arduino is able to read the key, a UID will be able to read back to the PC. The UID in this case is very important because it will be used as the registering/identification of a users' Pokéball.

In the testing phase of the coding plan, the UID will not be stored yet. However, the final coding plan will consist of reading and storing the UID to identify which

Pokéball has been read. An integrated display will be used to display a welcome message acknowledging that the Pokéball and user has been detected.

New parameters will be able to be created since the users' UID has been registered. This may include the starter Pokémon, traits and/or stats, along with the name of the user. The registering of a new Pokéball should only occur when the Pokéball's UID has never been registered before on the device.

After the Pokéball has been registered, then the next action may occur whether it may be a Pokémon battle or something else. The table below will consist of the steps required to make the coding process work.

Coding Checklist
Read the RFID key
If the key read is an existing user, allow the user to access their data or continue to the action parameter.
If the key read is not an existing user, then store the RFID UID as new Pokéball
If a new Pokéball is detected, then ask the user for choice of starter Pokémon and store choice as parameter.
Action Parameters ● Battle ● Evolution ● Gym Battle (Fight to win badge)

Table 26: General Final Coding Plan Checklist

Once a new user is given a Pokéball with RFID technology, then they shall proceed to scan their Pokéball against the reading device. The user will be prompted, hopefully by a graphical user interface, for a unique name they will like. After name selection, the user will be given the choice of three starter Pokémon from Generation 1 that will be shown below. The Pokémon shown have their names shown above them.

Figure 40: Starter Pokémon from Generation 1

After Pokémon selection, the user will be able to fight other Pokémon to increase their “experience.” The experience will be a numerical number that represents the necessary amount to “level up” to the next level. The advancement in levels will increase the statistics of their Pokémon. After reaching a certain level, their Pokémon will be eligible to evolve.The image below shows the evolution version of each starter Pokémon.

Figure 41: Starter Pokémon Evolution

The Pokémon will be able to continue their journey to fight in gym locations. These locations will host a gym "leader" who has several Pokémon where the user will have to face against. The user's Pokémon will face off in a series of battles to defeat the gym leader. After successfully winning against the gym leader, the user will receive a badge proving their win against the gym leader. A basic Pokémon battle is shown below.

Figure 42: Basic Pokémon Battle

The gym badges are shown below. After successfully collecting all of the gym badges, the user will have completed the game.

Figure 43: Awardable Gym Badges

7.4 Final Pokéball Shell Design

The completed hardshell design for the Pokéball was engineered and designed utilizing the Autodesk Fusion 360 platform. This is a platform that is known and familiar with the team. This platform provided the necessary tools and three dimensional modeling workflow to design and render the final Pokéball hardshell article for 3D-printing. This platform assists with ease in exporting to 3D printed format files through the '.stl' function embedded within the workflow for this software. A license to utilize the Autodesk applications for this project was acquired through Autodesk providing free software for University affiliated accounts and active university email addresses. This comes with the understanding that utilization of a Student account does not allow for the use of this software in a 'for-profit' sense.

The Pokéball hard shell consists of seven (7) major articles. These major articles contributed to be major factors in the overall construction of the total article of the Pokéball itself. The Pokéball in totality includes the seven (7) inclusive articles, alongside one (1) interior custom designed PCB that provides the electronic immersive layer to interact with the microcontroller and software system that has been designed, created, and tested by group 15. The seven (7) articles feature:

1. Upper Exterior Shell
2. Upper Interior Shell

Figure 44: Articles 1 & 2 of the Assembly

3. Lower Exterior Shell
4. Lower Interior Shell
5. Locking Mechanism

Figure 45: Articles 3, 4, & 5 of the Assembly

6. Frontward Facing Push Button

Figure 46: Article 6 of the Assembly

7. Tamper Proof Seal

The tamper proof seal and locking mechanism combine to form one sub article of the full scale article that is inclusive of all items that contribute to the full Pokéball item. The exterior shells are seperated in the design for a few reasons. The upper exterior shell and lower exterior shell are different colors from one another and from the interior shell halves. The interior shell halves are identical in design other than the locking mechanism that holds both halves together. This minor difference is a mirror of one another, other than it being an indention vs protrusion. The interior shell halves are of the same color, with a hex code of #222224 (black). The exterior shell halves (both upper and lower) are of two color varieties for the base Pokéball item. These colors are #ee1515 (red) and #f0f0f0 (white). The decision on separating these items based on color allows for the full color implementation to be achieved without cutting corners in the final deployment of the item prior to the Senior Design 2 showcase.

The tamper proof seal (separate article featured in a sub article of the locking mechanism) is a rubber gasket that blocks guests from tampering and dispersing design & engineering secrets. The penetration of this seal would acknowledge to maintenance that a potential warranty could be voided in the full deployment of this system in the proposed themed land (this is a design decision made with the full knowledge that this is to be presented as a showcase during Senior Design 2 and this item will not be fully implemented. It is shown in this project that extensive research and design decisions were made with a full engineering perspective in mind).

Article 6 is a front facing pushbutton of the same color variety as the lower exterior shell (#f0f0f0). This is an off-the-shell item that can be locked into place with a nut locking system that will be implemented into the lower half of the Pokéball. This push button will provide a physical means for the guest to 'activate' and 'release' their Pokémon when utilizing the experiences that are featured in the pitched land.

7.4.1 Pokéball Design Utilizing Fusion 360

The design inside of the Fusion 360 workspace began with a blank workbook. Following this step was the selection of the sketch tool and the backplane to begin the design. The prior research into Pokéball standards set by The Pokémon Company© story and heritage provided that the diameter of the outer shell article be no more than 3.15in in diameter. This is 5% larger than a regulation baseball utilized in Major League Baseball to provide a sense of scale.

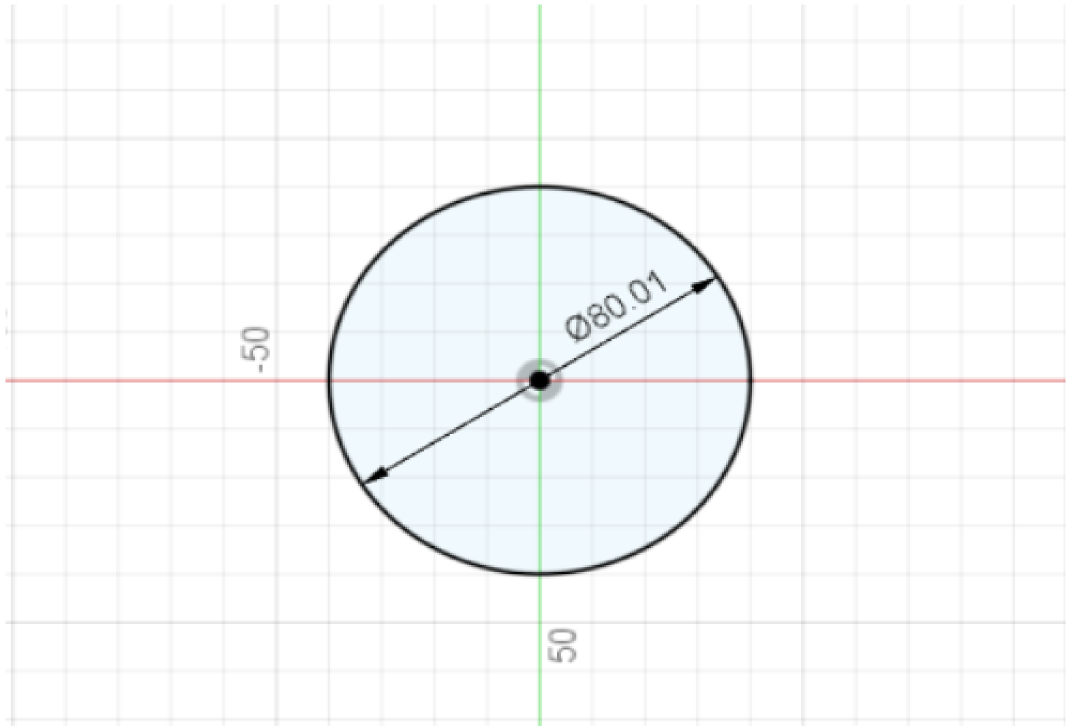

Figure 47: Initial Diameter Sketch

With a diameter of 3.15 in, we get a total volume measured from the outer portion of the interior shell (excluding the exterior shell halves) of the Pokéball to be 16.366 cubic inches, or 268.19 cubic centimeters. With the deployment in the United States of America as a basis for the land, empirical units and measurements were utilized in the design as well as metric system measurements.

Following the first creation of the diameter sketch, the object was trimmed to feature only half of the total area of the 2D circle, and then revolved 360° to create a 3D object that was 3.15 inches in diameter. The sketches of the interior shell halves and button decals were then created through the sketch tool. The design featured three (3) different sketch layers in the Fusion 360 workspace. The three sketches were:

- Upper and lower exterior shell articles
- Upper and lower interior shell articles
- Push button and ornament details revolving around the push button. Along with the rectangular cross section that separates the upper and lower exterior shell articles.

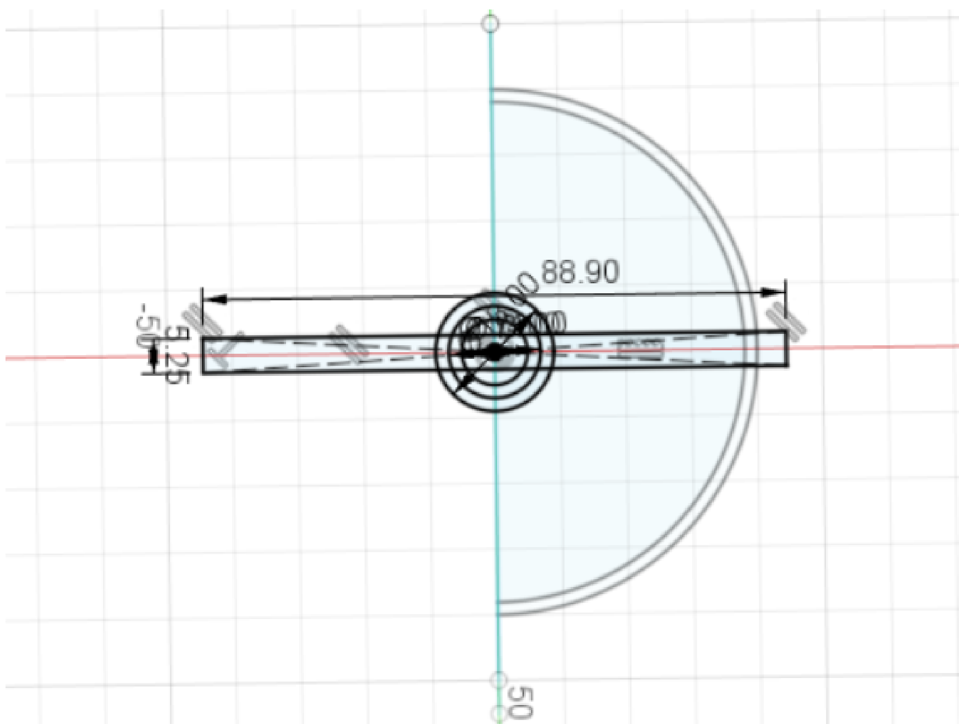

Figure 48: Sketch Models Of Pokéball

The thickness of the interior shell was decided to be 0.15 inches thick, which allows for 3.00 inches in diameter of clearance for the interior PCB, locking mechanism, and tamper proof seal. The outer shell was designed to be 0.15 inches thick to match the thickness of the interior shell width as well. Providing 0.3 inches or 7.62 millimeters of shell thickness in totality.

The proceeding design step was to create the volume of the total assembly, and extrude the rectangular & circular sections that will separate our two exterior shell halves and push button region. This motion was only taken from the central point in our assembly to extrude past the front facing section of the shell. The rectangular section would only then extrude through the back facing portion of the assembly, as the pushbutton and ornament details surrounding are only featured on the front facing portion of the Pokéball itself.

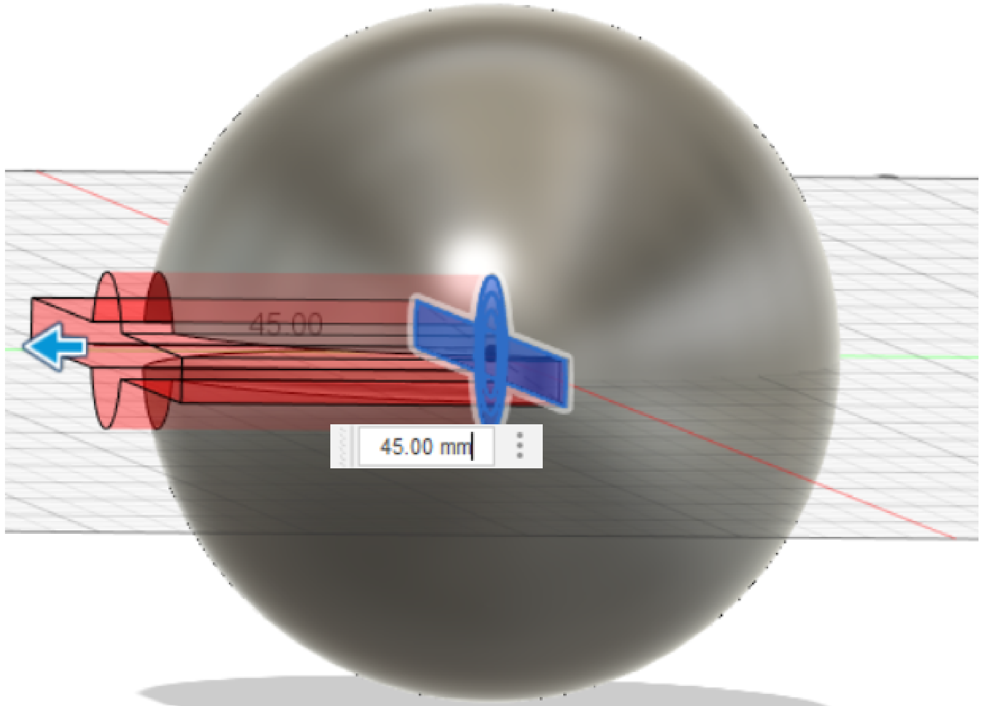

Figure 49: Front Facing Extrude Action

The two extrude actions described above allowed for the upper and lower shell halves to be free and portions of their own sketch. This completed the design portion of articles one (1) and three (3). Articles two (2), four (4), five (5), and six (6) were still to be progressed. The push button design was featured as a part of this design to understand exactly what size of off-the-shell item was necessary to purchase, as well as how the alignment could be configured with the full assembly. This was an oversight in the first few iterations of the design process, and during iteration three (3) of the design, the push button was featured as a plain circular sketch with cylindrical extrusion to allow for understanding on how the completed assembly should resemble. This also allows for proper renders of the final 3D model was utilized throughout this project. This does not however provide a 100% accurate model of the final render, and will then however potentially be adjusted as needed during the Senior Design 2 process. With this knowledge going into Senior Design 2, we are more prepared in this aspect to combat any issues therein. With availability of specific articles that can match the off-the-shelf push button article, it is not viable to 100% complete the necessary steps in the process of modeling. This was counteracted as much as possible with measuring online models of push-buttons and deciding what values were close to the necessary sizes.

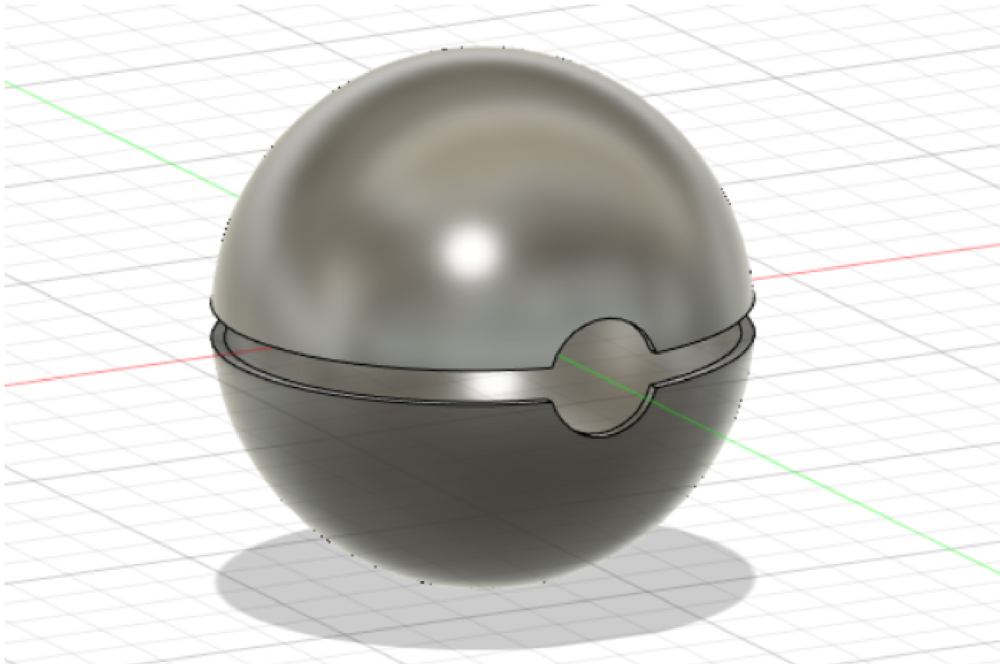

Figure 50: Completed Articles 1 & 3

Another design decision that was made along the varying iterations that occurred this summer were in the methods of first creating the sketches. Through creating the separate sketches, this allowed for the above Figure. 26 to show only the exterior upper and lower shell halves through the usage of visibility tools built into the Fusion 360 workspace. This function allowed for the proper extrude function for the ornament details and push button placeholder geometry to take place. The disk that surrounds the installed push button is 3.5mm thick and extrudes 1.5mm from the edge of the interior shell. The exact protrusion of the push button from the disk itself is unknown to be exact, and can be modified in the printing process if necessary. With all articles mentioned above completed (excluding articles that are only installed and not modeled on the interior of the shell) this completed the final iteration of 3D modeling to prepare for the final assembly and testing plans. The assembly modeled in the Fusion 360 workspace itself is minus the colors that will be utilized in the final presentation assembly. The process for which the colors will be added is mentioned in the following section on the final 3D printing process.

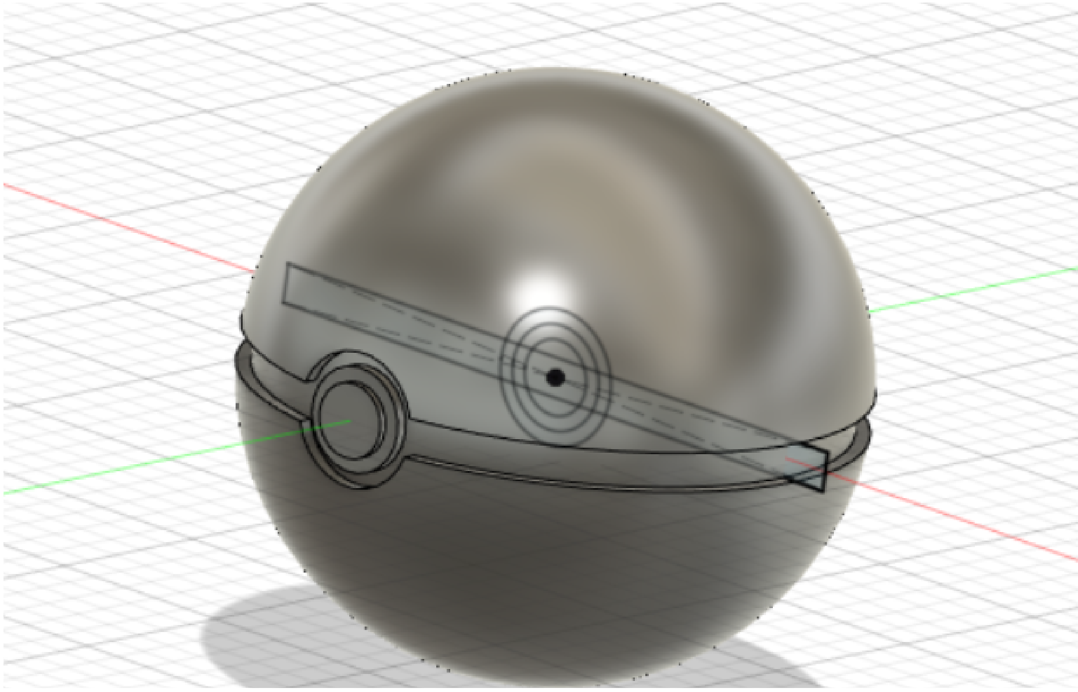

Figure 51: Final 3D Model Prior to Colorization

The above figure showcases the final assembly with sketches visible to provide visibility on more of the design process in its totality. The central cylinder on the front facing portion of the assembly is to be replaced with the off-the-shelf purchased push button and slotted into the ornament disk that surrounds it, and locked in the interior portion of the assembly through a locking nut. The final 3D model iteration for the process of Senior Design 1 was completed after three (3) full design iterations, and includes many adjustments to the design that are outside the scope of presentation here. Many errors and constraints may (and will) occur in Senior Design 2 that allows for this design to adjust and change. With those factors kept in mind going into the approach for the final iteration, the design is easily scalable both positive and negative. All items are separate components that can be adjusted to pivot along with the variable changes that Senior Design 2 will provide for group 15.

7.4.2 Final 3D Printing Process

The process for 3D printing this full assembly occurs first in the Fusion 360 workspace as previously utilized. The assembly is to be deconstructed into its major articles as described in section 6.4 above. The file is to then be exported into the extension of '.stl' to allow for the import into a privately owned 3D printer. The major shell contributions (articles 1-4) will be separated into four (4) distinct domes with the apex facing in the vertical z-axis direction. This will allow for the best quality 3D print to allow the diameter of the material to shrink overtime, rather than the opposite of attempting to print a bowl like structure. This configuration also allows for the supports to connect to the always interior facing portions of the shell articles. This will provide for faster and more efficient trimming and sanding time to the support structures that will be used in this print.

The printing material will be polycarbonate filament for this project, this material was selected for many reasons described in the chapter discussing project research above. Some of the major factors that informed the decision were:

- Highly durable
- Shatter resistant
- Heat resistant to 135℃
- Non-conductive (won't interfere majorly with RFID)
- Tensile strength of up to 9,800 PSI
- Ease of access (minimal lead time)

A further detailed look into the material itself and the testing procedure on its durability is featured below in section 7.3. To attach the exterior shell portions to their respective interior shell portions, two-part epoxy purchased off-the-shelf will be utilized. Most two-part epoxy mixtures feature an average PSI rating of 3,500, which is more than enough to hold the exterior shell articles to their interior shell articles respectively. Prior to bonding the exterior and interior shell articles together, the cleaning process and painting must occur.

Once the final articles are 3D printed, the support structures created will be trimmed off and sanded to remove any excess portions that deform our desired form. 80 grit sandpaper will be utilized to smooth the entire surface area of each printed article to ensure that the miniscule ridges that occur during the printing process are removed. This also prepares the painting process to provide those iconic colors that Pokéballs are known for. The priming process will begin after sanding has been completed. This will ensure that the articles are sealed and prepped to receive the paint. The paint utilized will be off-the-shelf spray paint of similar to exact coloration that is featured in the hex codes above section 6.4. Once sanded, primed, set, and painted, the final painting will include numerous sanding and repainting procedures to ensure even distribution along the articles. This will incur until satisfaction of the group and desired look is achieved. Once completed, the final assembly in its totality shall resemble the render of the final 3D model created in Fusion 360.

Figure 52: Final Pokéball Model Post Colorization

7.5 Senior Design 2 Construction and Coding Implementation

As Senior Design 2 progressed throughout the semester, the group had to decide on making various changes that would overall improve the outcome of our project. The major changes to the overall scope of the project occurred in the final assembly, as well as the implementation of the coding and software deployment for our project.

7.5.1 Final Assembly Construction

The final Pokéball assembly was still modeled in Fusion360, and constructed via the use of a PC polycarbonate 3D printed filament. The final assembly printed over the course of a 17h30m print. Once the print was complete, it was then removed from the support structure, sanded and filled with body-filling material, primed with 5 coats of spray-on primer, then painted to the correct colors to match the iconic assembly. The major change from Senior Design 1 to Senior Design 2 was the implementation of the PCB mounting on the interior of the assembly. The overall article count and descriptions remained the same otherwise. The assembly was bonded together utilizing the same mentioned JB-Weld two-part epoxy.

8.0 Project Prototype Testing Plan

The project prototype was created as an initial test before PCB design was created so we knew what parts will be used in the final product. To test the prototype, an Arduino Uno will be used. The final testing prototype will most likely consist of a breadboard design that uses the Arduino Uno, RC522 RFID module, and 1602 LCD to print messages to the user. After the test, a PCB design can be created that mocks the prototype. The Pokéball shell design will also be discussed in this section.

8.1 Basic Hardware Testing

This section will be explaining the basic hardware testing used as more components may be added to this prototype if required. The basic prototype setup consists of an Arduino Uno and an RC522 RFID module connected. The connections between the Uno and RFID module is referenced from Section 6.1.2, where the table lists where each pin is connected to. The image below shows the initial connections between the two parts.

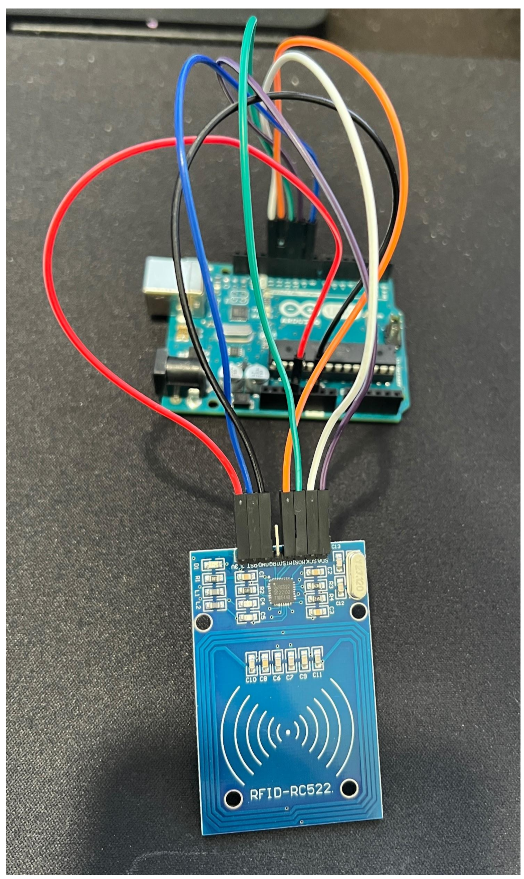

Figure 53: Hardware Prototype

For the testing purposes, the Arduino Uno will be connected to a PC using a USB connection for power and programming, but when the design is finished it will utilize a 9V battery so it can be used as a stand alone RFID reader. The system will also eventually showcase a display attached to the device so that the user is able to manipulate certain data parameters if necessary.

The device will be able to read two different types of tags which are going to be used for testing purposes. A key card and key tag chain are the two testing tags which both have an operating frequency of 13.56 MHz. The image below shows both the key card and key tag chain that are going to be used for testing. The key tag chain may be the better choice for the Pokéball because of its small form factor that can be integrated into it.

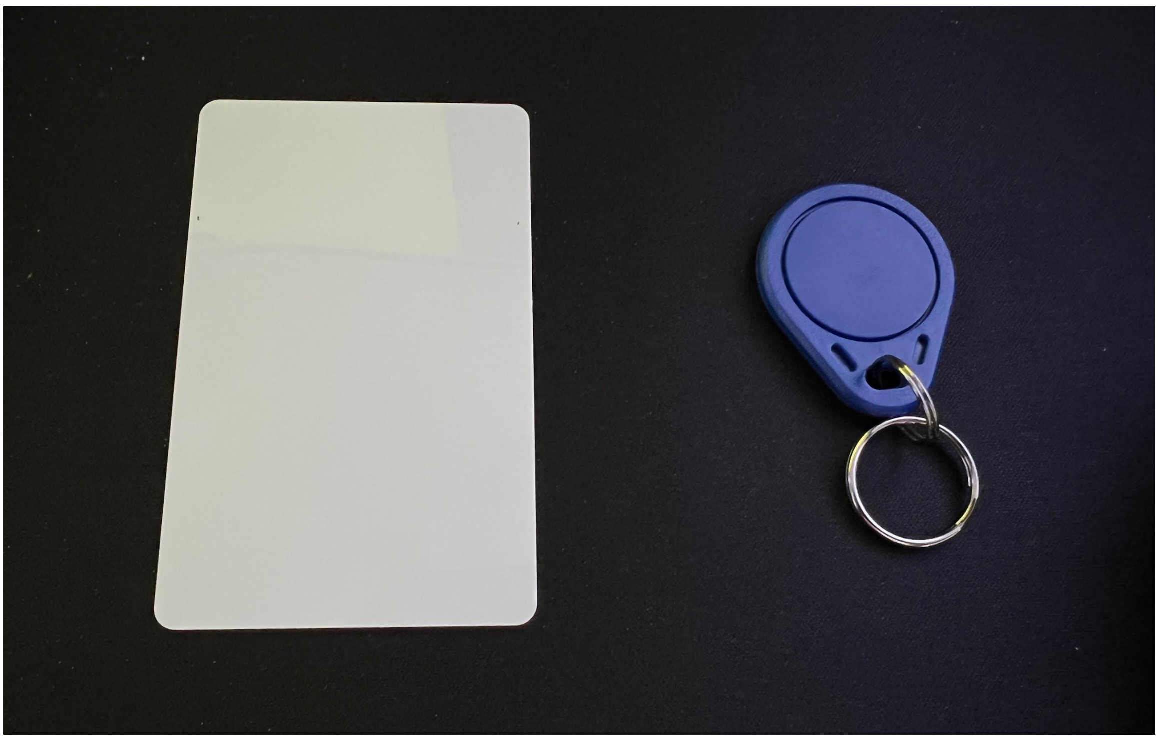

Figure 54: Key Card and Key Tag Chain for RFID Test

After confirming that the hardware above works to detect the key, I wanted to test adding an LCD screen to the setup so that it will be able to print the key UID directly to the LCD screen. To make this work a breadboard, 1602 LCD, and a 10K ohm potentiometer is required. The 10K potentiometer will be used as a dial to configure the brightness of the LCD screen. Figure # in Section 7.1.2 will be used as the reference schematic to assemble the breadboard. The figure below shows all components connected together on the breadboard.

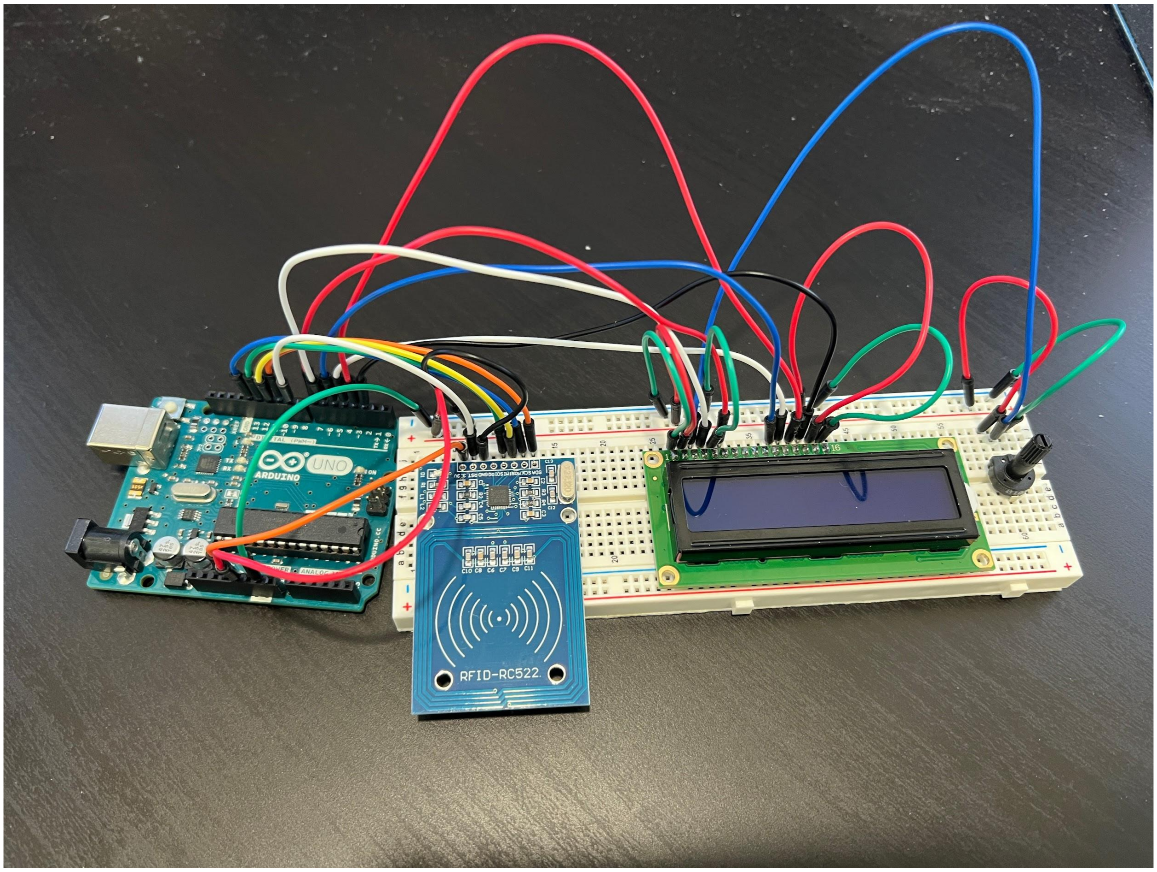

Figure 55: Breadboard Setup With 1602 LCD

After the board has been programmed, the two figures below will show what will appear on the screen. Before the user scans their key, which will be the Pokéball eventually, a welcome message will be displayed so in our case “Scan RFID Key”. When the key has been read by the RFID, the LCD will print a message containing the Pokéball’s UID. The UID is unique and the key cannot be rewritten, this helps prevent other users from accessing other data parameters.

The schematic shown in Figure 33 is what was used to make the breadboard connections as shown in Figure 47. Currently the Arduino Uno is still powered by a PC which will also allow data transfer between the PC and Arduino. The sketches can be uploaded to the Arduino using a USB B cable.

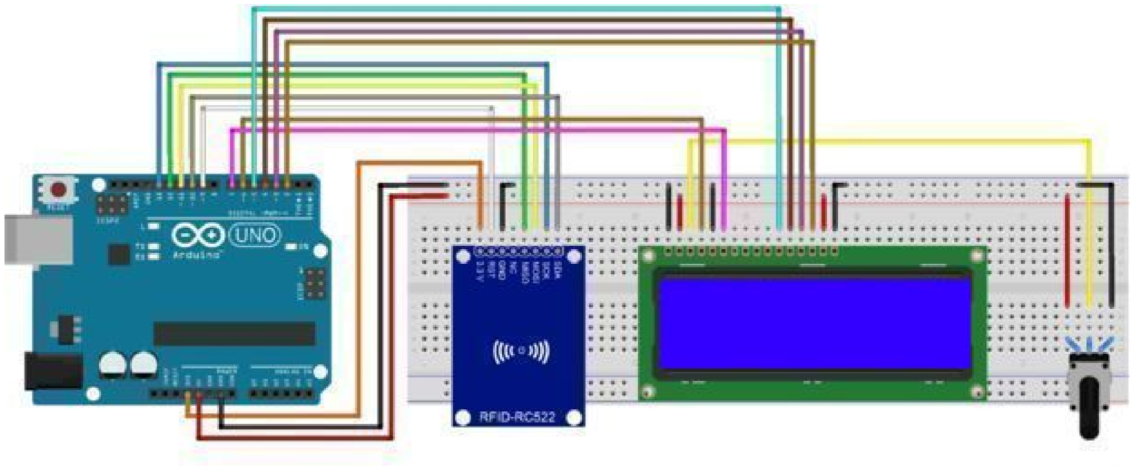

Figure 56: Schematic of Assembled Breadboard

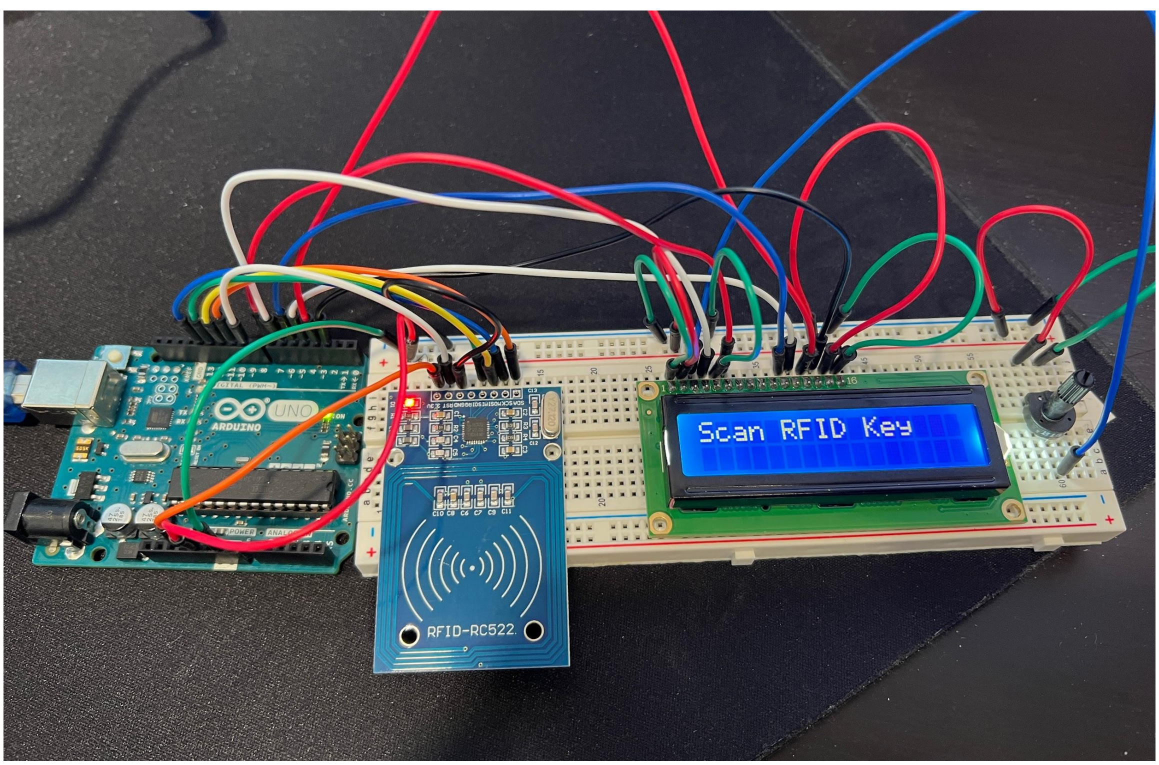

Figure 57: RFID Reader Full Prototype

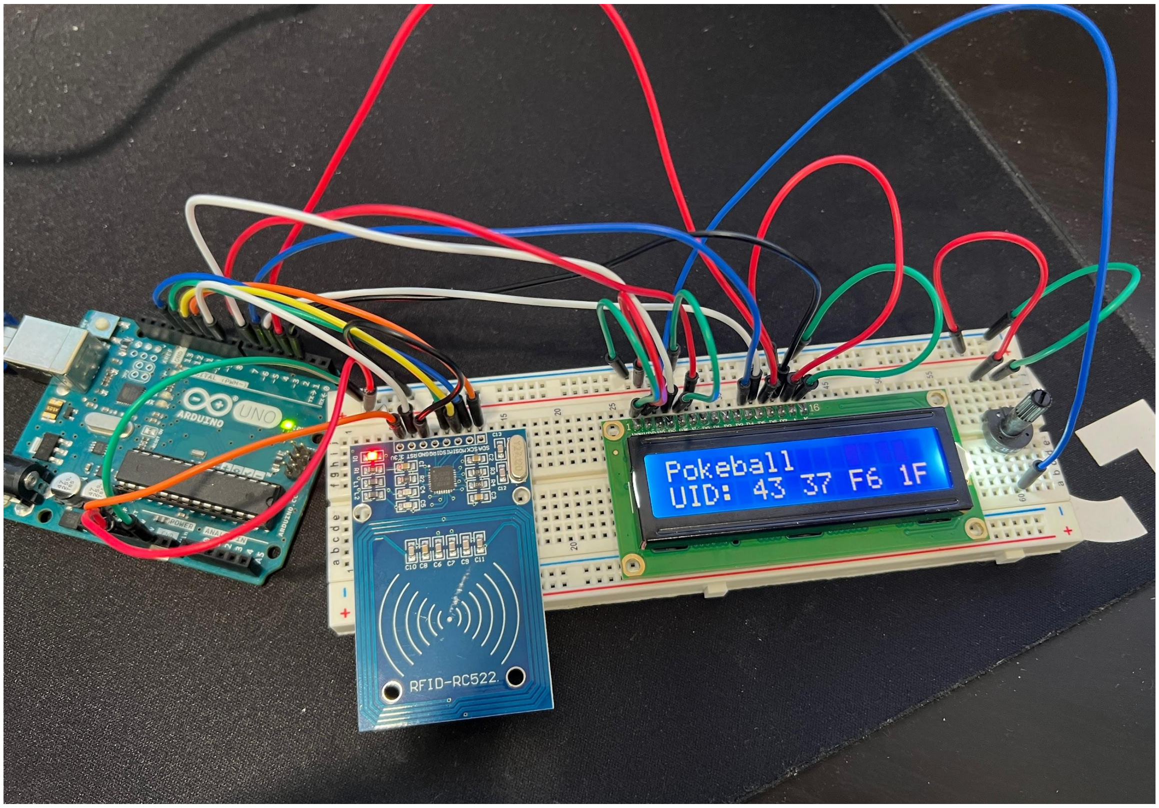

Figure 58: RFID Reader After Scanning Key

8.1.1 Final Base Station

After the completion of the demonstration video for Senior Design 2, the base station is built using three major components. But to recap, the base station was designed to be the radio frequency receiving device that will identify guests' Pokéball and send the information to the database.

The first major component of the base station is the radio frequency transceiver. The purpose of this transceiver is to receive the 915 Mhz signal that is sent from the PCB located in the Pokéball. The component of choice for this is the LoRa RFM95W, which is 915 Mhz radio frequency transceiver. The reason this transceiver is selected is because since this project does not rely on close proximity as users may change their location based on interactions in the real world, it is crucial to pick a transceiver that has the capability of detecting a device from a larger range.

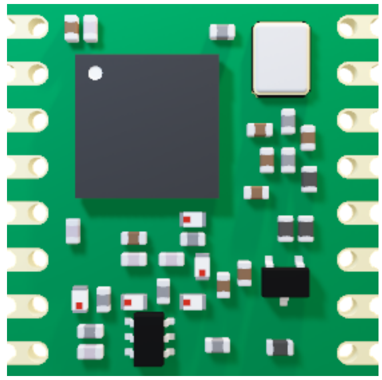

Figure 59: LoRa RFM95W Transceiver

The second major component is the microcontroller. The purpose of the microcontroller is to allow software to control the transceiver and process any of the data received and sent to the database. So the base station features an ATMEGA328P-AU. It is a very small chip which is perfect for the base station since one of its features is its small form factor. The microcontroller allows the use of the Arduino IDE which is how this microcontroller was programmed. However, in order to communicate with the microcontroller, the use of an FTDI chip is necessary which is discussed below.

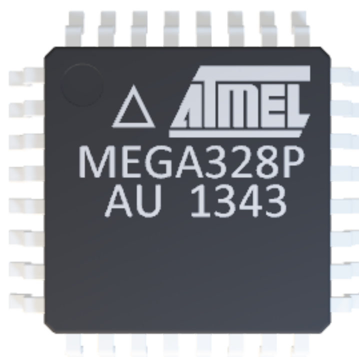

Figure 60: ATMEGA328P-AU Microcontroller

The last major component is the universal serial bus controller which is the FT232RL. In order for the microcontroller to be programmed, it needs to be connected to a computer to work with the Arduino IDE. So a FTDI chip is wired to the microcontroller so that it will allow communication over a USB cable.

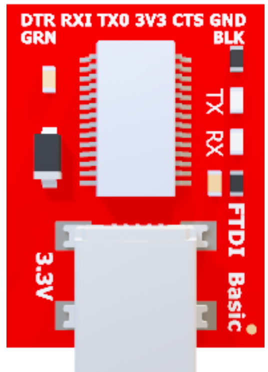

Figure 61: FT232RL FTDI Module

The three components discussed above for the base station were soldered together on a vector board. The use of a breakout board for the LoRa RFM95W and ATMEGA328P-AU allowed the use of correct spacing to allow the components to be soldered on the board.

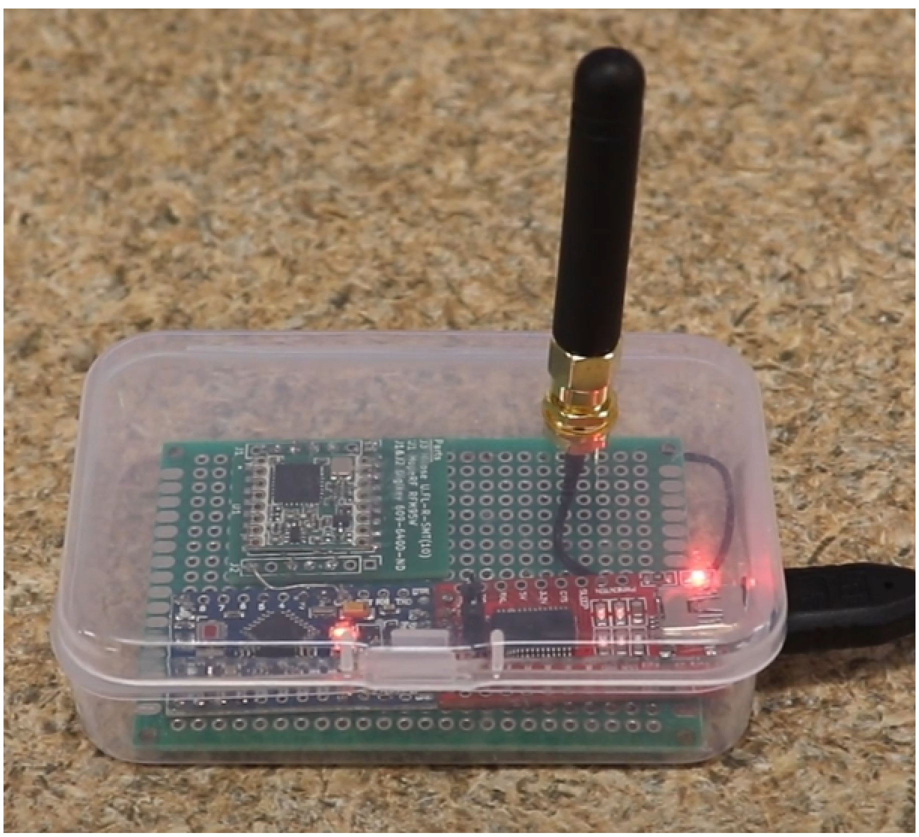

Figure 62: Final Base Station

8.1.2 Final PCB

The final PCB schematic and deployed model were validated and tested utilizing the Texas Instruments SmartRF Studio software. The final PCB was made via the use of hand soldering components and solder paste reflowed. Below showcases the final PCB schematic from KiCad 6.0

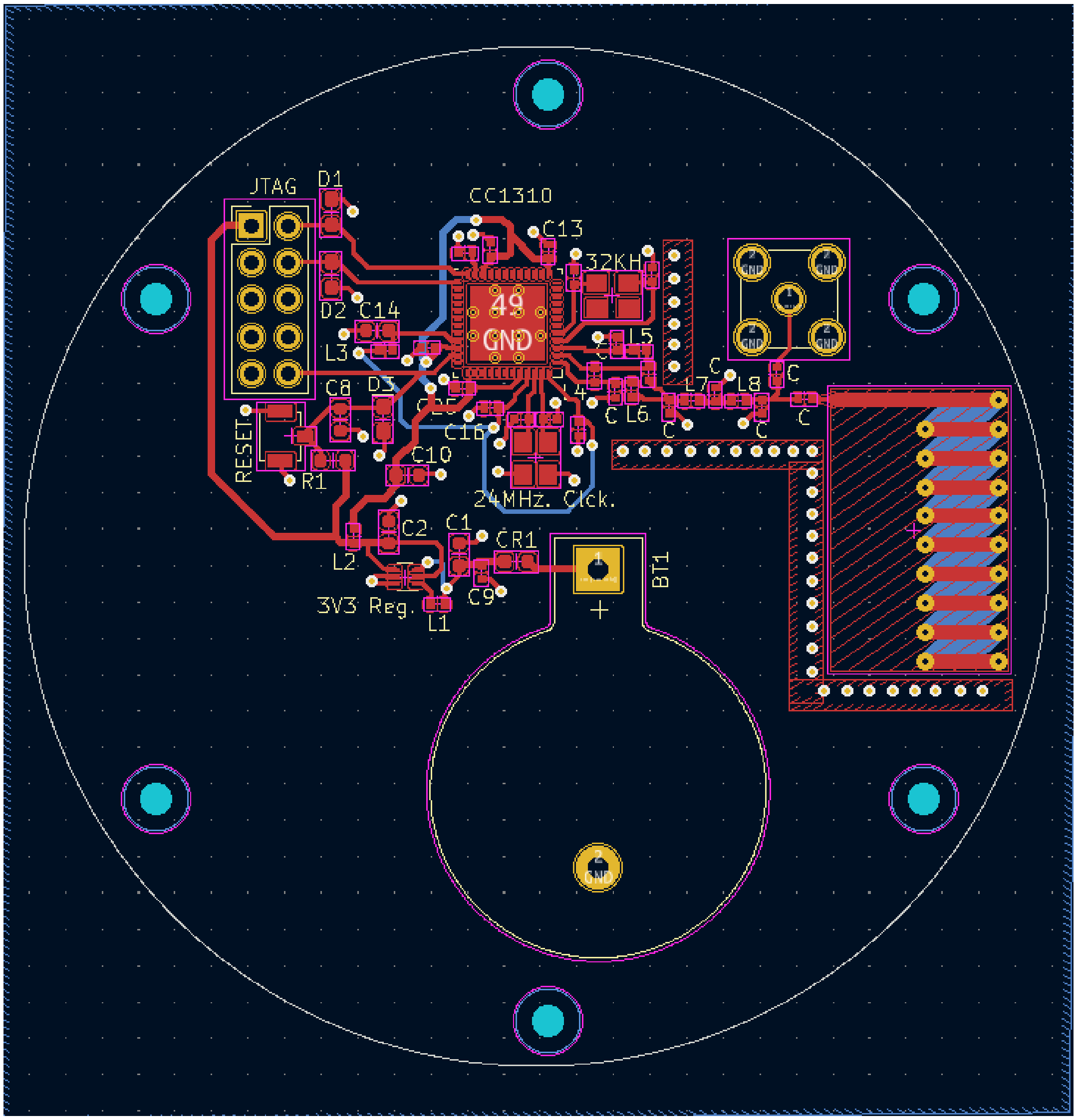

Figure 63. Pokéball PCB Layout

The PCB consists of the 7-major sections. The power and voltage regulation, two crystal oscillator circuits, reset switch, JTAG pin headers, microcontroller and bypass capacitors, and the RF Impedance Matching & Balun circuitry. These sections all relied on the CR2032 coin cell battery and voltage regulation done by the TPS69291DRV Boost Regulator from Texas Instruments. The major MCU is found in the CC1310 by Texas Instruments. The RF Impedance Matching network was design and manufactured to match the impedance of 44+j15Ω, specified in the CC1310 datasheet courtesy of Texas Instruments.

Once the final PCB was delivered via JLCPCB, components were delivered by DigiKey and Mouser Electronics, the board was soldered and constructed various times to provide the final board. The final board was completed to be accurate and reliable. The final construction combined together with the 3D-printed housing, and was able to mount properly while still maintaining appropriate transmission with the Base Station, overall proving the engineering requirements and specifications of Project Apricorn that were set in Senior Design 1.

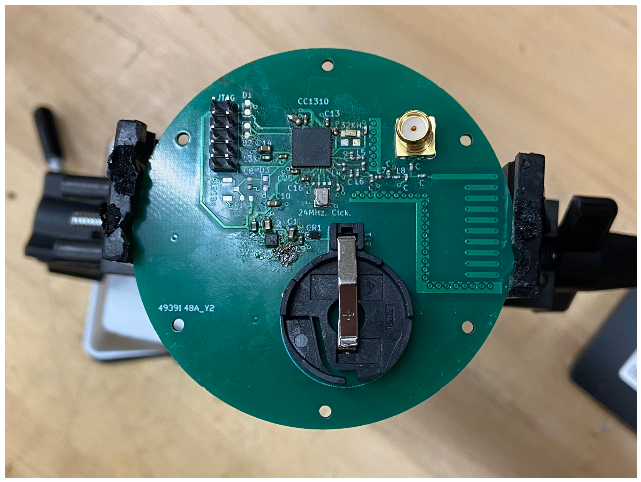

Figure 64. Final Pokéball PCB

8.2 Basic Software Testing

The basic software testing will be mainly for testing if the Arduino Uno with RC522 module is able to read and identify multiple tags. The Arduino IDE will be used as the programming interface used. This is a great way to ensure that the sketches, which are the uploadable programs that are created by us, work with both the Arduino and all hardware components attached to the breadboard.

To begin with the programming of the Arduino Uno with RC522, the MFRC522 library was imported. Using the example code provided by the MFRC522 library, the RFID can be tested to ensure the hardware was working properly. The serial monitor for the Arduino IDE will display the UID of the key tag that is scanned with the RC522.

```
#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN         9          // Configurable, see typical pin layout above
#define SS_PIN          10         // Configurable, see typical pin layout above

MFRC522 mfrc522(SS_PIN, RST_PIN);  // Create MFRC522 instance

void setup() {
  Serial.begin(9600);   // Initialize serial communications with the PC
  while (!Serial);    // Do nothing if no serial port is opened (added for Arduinos based on ATMEGA32U4)
  SPI.begin();      // Init SPI bus
  mfrc522.PCD_Init();   // Init MFRC522
  delay(4);       // Optional delay. Some board do need more time after init to be ready, see Readme
  mfrc522.PCD_DumpVersionToSerial();  // Show details of PCD - MFRC522 Card Reader details
  Serial.println(F("Scan PICC to see UID, SAK, type, and data blocks..."));
}

void loop() {
  // Reset the loop if no new card present on the sensor/reader. This saves the entire process when idle.
  if ( ! mfrc522.PICC_IsNewCardPresent()) {
    return;
  }

  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial()) {
    return;
  }

  // Dump debug info about the card; PICC_HaltA() is automatically called
  mfrc522.PICC_DumpToSerial(&(mfrc522.uid));
}
```

Figure 65: Example Code Used to Test RC522

After placing the tag over the reader, the Serial Monitor on the Arduino IDE is able to detect the tag and is able to give out the UID of the key tag. Multiple tags were tested to verify that the reader was compatible. The figure below shows what is printed out of the Serial Monitor. The UID that is printed out in the Serial Monitor can then be used with a database system so that the device will eventually know which tag has been registered before or requires new registration.

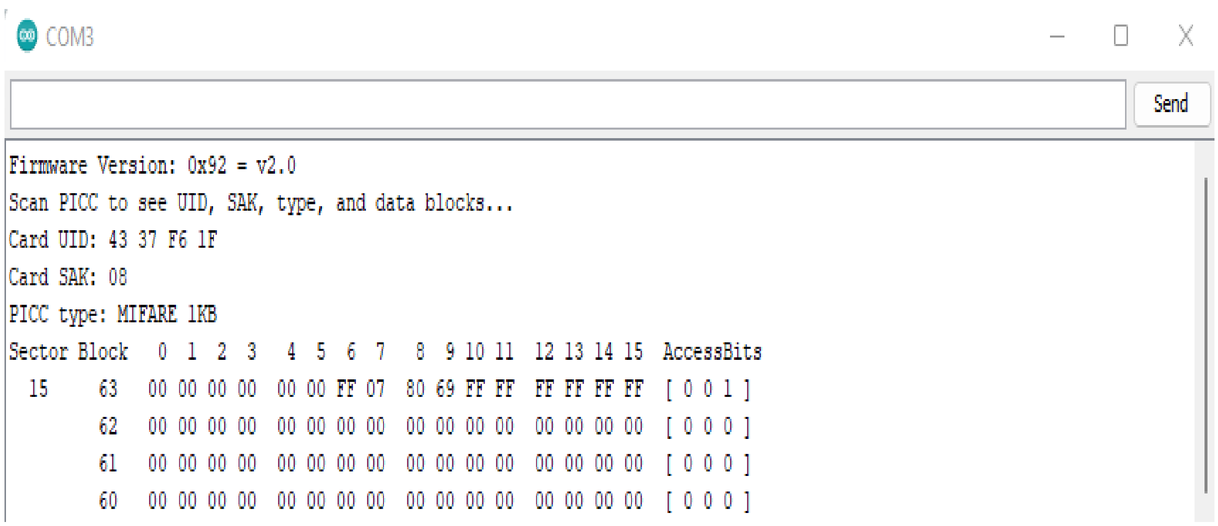

Figure 66: Serial Monitor Showing Key UID

The Card UID that is shown above will be very important for the application of the Pokéball. When designing the final code, the UID obtained from this process is going to be stored so that it will be accessible for use later. If the UID does not match any existing UIDs that are stored, then a prompt to allow the user to register their Pokéball should appear. Once the user has selected what Pokémon they wish to start with, it should be stored within their UID's parameters.

After the breadboard has been assembled with the RC522 and 1602 LCD, the Arduino needs to be programmed. The code used will be shown below. The code demonstrated will prompt the user to scan their rfid key, then once the tag has been tapped along the reader, it will identify their UID and print it on the LCD. In the later sections, a graphical user interface will be developed so that once the tag is scanned, the system will be able to determine whether that UID has been registered before. If the UID has not been registered, then the system will prompt the user to register their Pokéball and be stored within the system's database.

In the future, more libraries may be implemented to the sketches to allow easier use of functions. It will be important to keep the code concise as it will be built on further when the graphical interface has started to be developed. The breadboard design will also change to a more permanent PCB design so the code will be modified to change the pins to the correct new location.

```
#include <SPI.h>
#include <MFRC522.h>
#include <LiquidCrystal.h>
#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN);
LiquidCrystal lcd(6 , 7, 5, 4, 3, 2);

void setup()
{
  SPI.begin();
  mfrc522.PCD_Init();
  lcd.begin(16, 2);
  lcd.print("Scan RFID Key");
}

void loop()
{
  if ( ! mfrc522.PICC_IsNewCardPresent())
  {
    return;
  }
  if ( ! mfrc522.PICC_ReadCardSerial())
  {
    return;
  }
  lcd.clear();
  lcd.begin(16, 2);
  lcd.print("UID tag :");
  String content= "";
  byte letter;
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {
    lcd.setCursor(0, 1);
     lcd.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
     lcd.print(mfrc522.uid.uidByte[i], HEX);
     content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
     content.concat(String(mfrc522.uid.uidByte[i], HEX));
  }
lcd.clear();
lcd.begin(16, 2);
lcd.print("Pokeball");
 content.toUpperCase();
  if (content.substring(1) == "43 37 F6 1F")
  {
   lcd.setCursor(0,1);
    lcd.print("UID: ");
    lcd.print(content.substring(1));

    delay(10000);
   lcd.clear();
    setup();
  }
 else   {
  lcd.setCursor(0, 1);
    lcd.print(" Access denied");
    delay(3000);
    lcd.clear();
      setup();
      }
}
```

Figure 67: Arduino IDE Code for Full Prototype

A few Arduino IDE software parameters will be discussed below to allow readers and members of the group to reference as a guide.

Libraries

- #include <LiquidCrystal.h>
  - The LiquidCrystal library simplifies the process of controlling the LCD so no knowledge of low-level instructions are required.
- #include <SPI.h>

    - Allows communication with SPI devices with the Arduino as the controller device.
- #include <MFRC522.h>
    - Read/Write a RFID card or tag using the MIFARE interface.

General Functions

- Serial.begin(speed)
    - Sets the data rate in bits per second for serial data transmission.
- SPI.begin()
    - Initializes the SPI bus by setting SCK, MOSI, and SS to outputs, pulling SCK and MOSI low, and SS high.
- Serial.print("message")
    - Prints the contained message to the Serial Monitor located within the Arduino IDE

Object Parameters for LCD

- LiquidCrystal lcd(RS, EN, D4, D5, D6, D7)
    - Defines which Arduino's pins are connected to LCD pins. Any six Arduino pins can be used as long as they are specified where they are defined in the object.
- lcd.begin(row,column)
    - Set up the LCD's number of columns and rows.
- lcd.setCursor(column_index, row_index)
    - Moves the cursor to the desired position on the LCD.
- lcd.print("Message")
    - Prints the contained message to the LCD screen.
- lcd.clear()
    - Clears the LCD screen and positions the cursor to the upper left corner.

Object Parameters for RFID

- *name*.PICC_IsNewCardPresent()
    - Detects if a new tag is available
- *name*.PICC_ReadCardSerial()
    - Reads the UID of the tag detected

An attempt to create an Arduino GUI sketch for Pokémon starters will be described in this section. The Processing software will be used to create a simple GUI that will work after the Pokéball has been detected and scanned. Processing is an open source integrated development environment (IDE) similar to the Arduino IDE but is used by designers and artists.

Processing allows users to create visual and interactive programs. It can use Arduino serial data and send it to Processing using a serial library. Arduino and

Processing are able to do multi-byte communication between each other using a call-and-response (handshaking) method. This will allow the Arduino IDE to send the required information to Processing which will create the graphical interface which will then send the information back to the Arduino IDE.

To make the initial testing easier, a simple graphical interface to turn on the LED on and off will be the test. The Arduino IDE code is shown below.

```
void setup() {
  Serial.begin(9600);//enable serial communication
  pinMode(22, OUTPUT);//define arduino pin
}

void loop() {
  if (Serial.available() > 0) {
    //check serial value
    char led = Serial.read();

    if (led == '1') {
      //check serial read
      digitalWrite(22, HIGH);//LED on
    }
    else if (led == '0') {
      //check serial read
      digitalWrite(22, LOW);//LED off
    }
  }
}
```

Figure 68: Arduino IDE Code for LED Toggle

Since the Processing code is too long to present here only the GUI created can be shown.

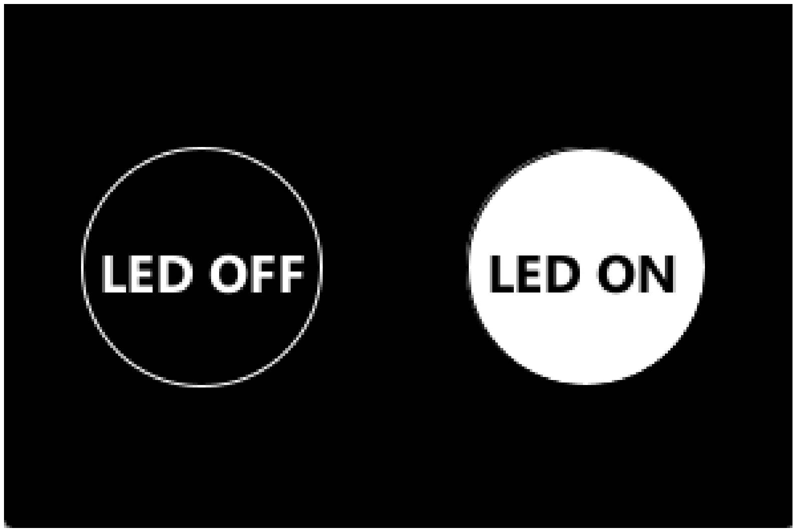

Figure 69: Graphical Interface for Toggling LED

The graphical buttons have been tested and do work. The figure below shows that the LED is off and then toggled to the lit position. An Arduino MEGA was used to test this as the Arduino UNO was still occupied as being the completed prototype.

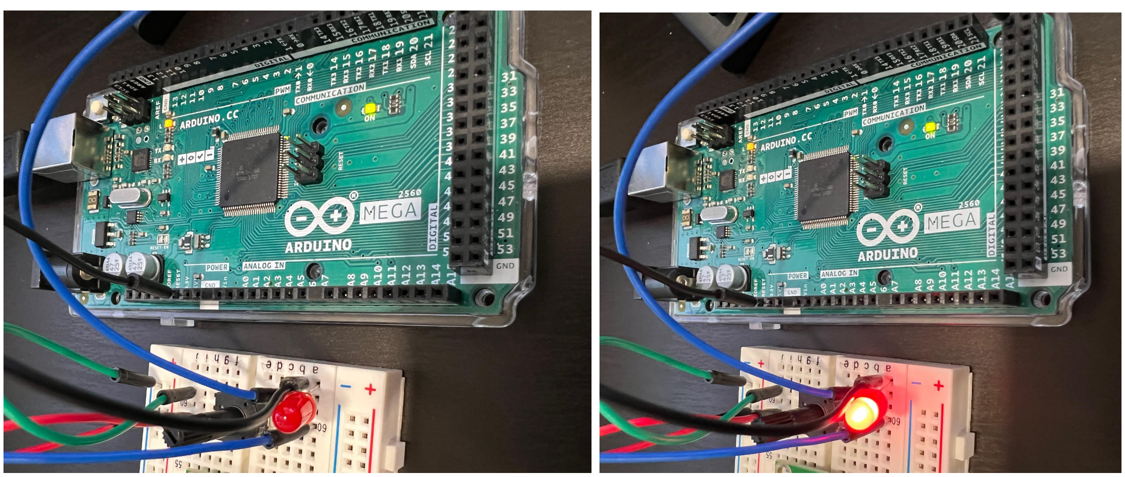

Figure 70: Toggling the LED Using Processing

8.2.1 Initial Graphical User Interface Development

As discussed in section 7.3, when the software begins a start page will display indicating the user to tap their Pokéball against the RFID reader. Once the reader has identified the Pokéball's UID, then the device will determine whether the UID has never been registered or if that user is a registered user. The two figures below show the start up screen and the Processing code that has been written before any user has scanned their Pokéball.

Figure 71: Start Screen for Prototype

This start screen was taken from the first generation Pokémon game, specifically the Red version. This will be the general page until any user scans their Pokéball to begin a new action whether it may be new user registration or continuing their Pokémon adventure.

```
import processing.serial.*;
PImage img;
PFont font;

Serial myPort;
static String val;
int sensorVal = 0;

 // Set position for asking for RFID
 int rfidx = 70;
 int rfidy = 250;


void setup() {
  size(500, 500); // Sets dimensions of window in pixels
  frameRate(30); // Frame rate
  img = loadImage("startscreen.png"); // Load the image to the window
}

void draw() {
 background(img); // Set the Pokemon start screen as background
 fill(0); // Make text black
 font = createFont("GillSansMT-BoldItalic-24", 24); // Similar to Pokemon font
 textFont(font);
 text("Please tap your Pokeball to begin!", rfidx, rfidy);
}
```

Figure 72: Start Screen Processing Code

Once a new user has scanned their Pokéball, a new user registration page will prompt the user asking for what they would like to name their virtual character.

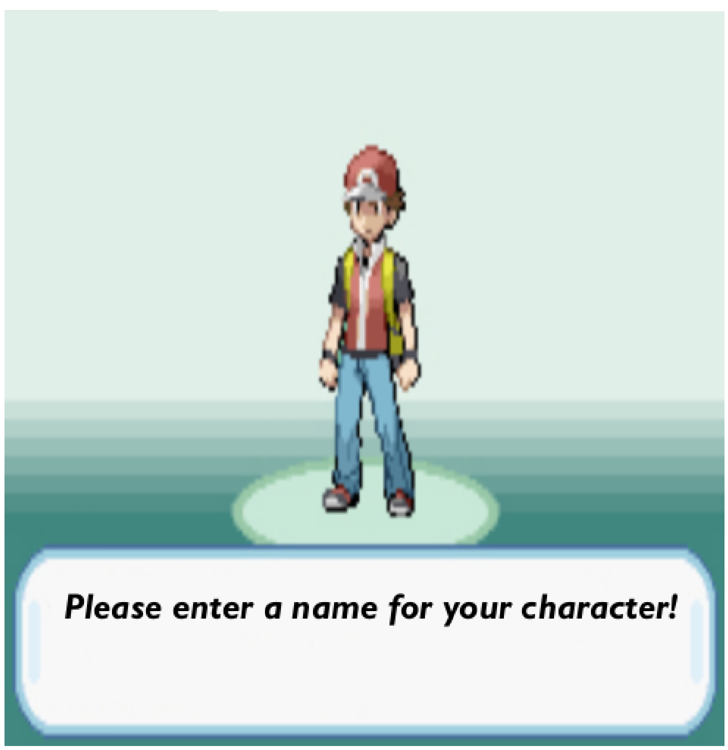

Figure 73: Character Name Selection

The user will have the chance now to type in their name for their virtual character. The graphical interface will have a text by text input so the user is able to see what they are typing before pressing the return key on the keyboard to continue.

Figure 74: Character Name Selection Input

After the user has typed the character name they desire, the next prompt will greet them with their selected name and thank them for registering with the system. After the character name selection, the starter Pokémon selection will begin. The figure below shows the graphical user interface showing the three possible starter Pokémon that are selectable: Bulbasaur, Charmander, and Squirtle.

Figure 75: Pokémon Selection Interface

After the successful completion of the character name selection, the start Pokémon selection page will display. On this page, the user will have the option of choosing between the three Pokémon shown above. If the user is unsure of which Pokémon to select, they can press the More Info button to display a popup of a general description of that Pokémon. There will be a close popup button at the top left corner of the popup that will allow the user to close the window so they can select a Pokémon. The three figures below will show what the popup window will look for all three Pokémon, Bulbasaur, Charmander, and Squirtle. It is important for the user to decide wisely on which starter Pokémon they select because once they select a Pokémon they will be unable to change it for the rest of the gameplay.

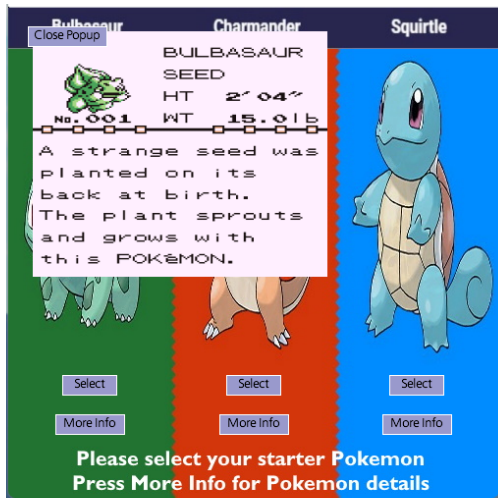

Figure 76: More Info Popup for Bulbasaur

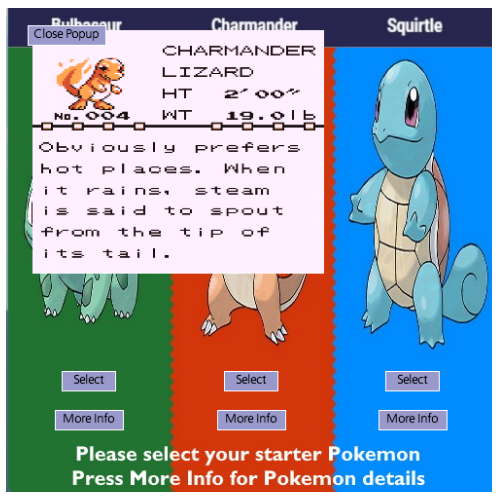

Figure 77: More Info Popup for Charmander

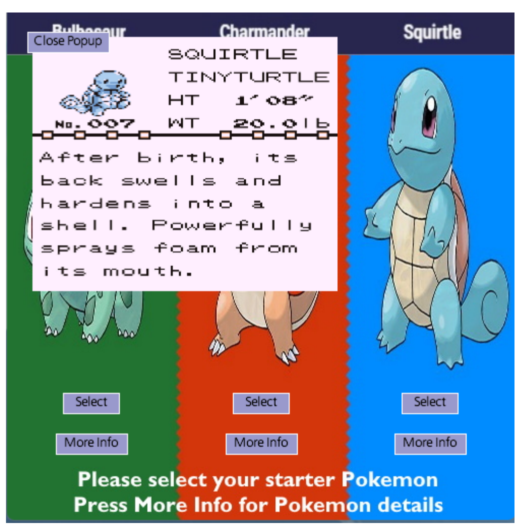

Figure 78: More Info Popup for Squirtle

The figure below shows the processing code used for the Pokémon selection window. In order to make the button implementation easier for this part, a new library was used for this. Interfascia is a graphical user interface library for the Processing graphics programming environment. It provides a toolkit of standard interface widgets like text field, buttons, checkboxes, sliders, etc. Interfascia automatically handles interactions within the collection of widgets and sends event messages to your project.

For our graphical interface, six buttons were required, three for Pokémon selection and another three to view details of the Pokémon. All the buttons were coded in using the new IFButton command. The buttons were able to have custom labels along with pixel positioning using x and y coordinates. The buttons are also able to custom size using the number of pixels for width and height. The figure below shows the Processing code for the Pokémon selection method.

```
void pokemonSelection() {
  img = loadImage("pokemonselection2.jpg"); // Pokemon selection screen
  background(img); // Set the Pokemon selection screen as background

  b1 = new IFButton("Select", 55, 375, 55, 20); // Button to select Bulbasaur
  b2 = new IFButton("Select", 222, 375, 55, 20); // Button to select Charmander
  b3 = new IFButton("Select", 388, 375, 55, 20); // Button to select Squirtle

  m1 = new IFButton("More Info", 48, 415, 70, 20); // More Info Button for Bulbasaur
  m2 = new IFButton("More Info", 215, 415, 70, 20); // More Info Button for Charmander
  m3 = new IFButton("More Info", 381, 415, 70, 20); // More Infor Button for Squirtle

  // When the associated button is pressed, goes to actionPerformed method
  b1.addActionListener(this);
  b2.addActionListener(this);
  b3.addActionListener(this);
  m1.addActionListener(this);
  m2.addActionListener(this);
  m3.addActionListener(this);

  // Add the associated buttons
  c.add(b1);
  c.add(b2);
  c.add(b3);
  c.add(m1);
  c.add(m2);
  c.add(m3);
}
```

Figure 79: Processing Code for Pokémon Selection

After the buttons have been integrated along with their required labels, sizing, and positioning, the buttons need to have an action associated when they are selected. Currently, the select Pokémon does not have an action attached as the database needs to be set up beforehand. However all the other buttons are currently operating correctly. The three more info buttons will display a popup showing a description of the Pokémon. The description is basically an image that is loaded to the graphical interface. When any of the three more info buttons are pressed, a close popup button will also appear with the popup window. This will allow the user to close the window when they are ready to select their starter Pokémon.

```
void actionPerformed (GUIEvent e) {
  if (e.getSource() == m1) {
    img = loadImage("bulbasaur.png");
    image(img, 25, 25);
    close = new IFButton("Close Popup", 20, 20, 80, 20);
    c.add(close);
    close.addActionListener(this);
  }
  else if (e.getSource() == m2) {
    img = loadImage("charmander.png");
    image(img, 25, 25);
    close = new IFButton("Close Popup", 20, 20, 80, 20);
    c.add(close);
    close.addActionListener(this);
  }
  else if (e.getSource() == m3) {
    img = loadImage("squirtle.png");
    image(img, 25, 25);
    close = new IFButton("Close Popup", 20, 20, 80, 20);
    c.add(close);
    close.addActionListener(this);
  }
  else if (e.getSource() == close) {
    c.remove(close);
    img = loadImage("pokemonselection2.jpg"); // Pokemon selection screen
    background(img); // Set the Pokemon selection screen as background
  }
  else {
    img = loadImage("pokemonselection2.jpg"); // Pokemon selection screen
    background(img); // Set the Pokemon selection screen as background
  }
}
```

Figure 80: Processing Code for Pokémon Selection Buttons

The figure above shows the Processing code for the Pokémon selection buttons. The more info buttons have been labeled as m1, m2, and m3, for Bulbasaur, Charmander, and Squirtle . An else if statement is used for when the popup window appears, then the close popup button will be removed and the popup window will disappear.

8.3 Pokéball Shell Testing

The testing for the 3D printed assembly of our Pokéball will occur in its totality at the beginning of the printing process during the first week of Senior Design 2. Described in section 7.4.2 on the final 3D printing process, the different articles that construct the final Pokéball assembly will be printed in their own pieces. This will allow us to separate and optimize the testing process for this final assembly. As each piece is completed through the printing stage, they must undergo measurement inspections to ensure that the correct arc length, width, and density are met during the process. This must incur due to the scenarios that 3D printers have an error percentage, and have potential to fail during the printing process. Once each printed article is validated, a mock assembly will occur in stages. The stages of the mock assembly will feature:

- Upper exterior and upper interior shell formed articles
- Lower exterior and lower interior shell formed articles
- Upper half shell and lower half shell formed articles
- Ornament detail and push button formed articles
- Full Pokéball assembly of all printed & purchased articles

A successful mock assembly for this prototype project will involve each article combination to feature a gap of equal or less than 0.1 mm. This strict gap regulation will ensure that the Pokéball still stays true to the roots it is bounded by through the world of Pokémon. Following the mock assembly, another mock assembly with a dummy weight will occur. This dummy weight will be in comparison to the weight that combines to feature all interior aspects of the assembly. This includes tamper proofing materials, PCB, and gasket. This will allow us to validate the initial claims made in the requirements of weight for the Pokéball assembly.

Following weight validation, the mock assembly with dummy weight will be bonded together through means similar to that of the final assembly to present. This then allows for more strenuous testing to occur. Drop testing and tensile testing will be the two major contributors to determine the physical strength of the assembly. The drop testing will occur starting at 6” above the ground plane of the testing zone, and increase 6” every successful test until the assembly can no longer withstand the impact force with the ground plane of our testing zone. The surface in which we tested the impact force was a concrete surface, this was the most strenuous material for the drop test to occur on that would resemble the surfaces that guests utilize for ingress and egress in the proposed land. A successful test will be described as the following:

- No major cracks, openings, crevices, gashes, and other deformations that compromise the integrity of the full assembly and the interior articles after incrementing 6” drops will be considered a successful test. Cosmetic damage such as scuffs, bumps, and paint scrapings that do not affect the operation and overall integrity of the full assembly will not constitute a failing test.

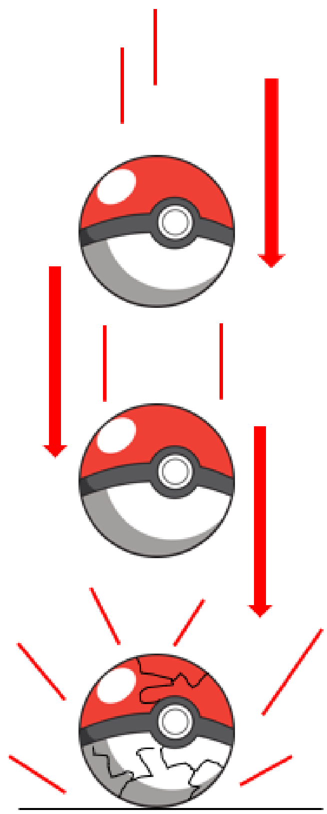

Figure 81: Pokéball Drop Test (Pokéball Image Provided by The Pokémon Group©)

A tensile strength test will validate the locking mechanism and gasket strength of the assembly itself. A point on the apex of the upper half shell and apex of the lower half shell will be connected, and a tension force will be applied. A gauge to measure how many Newtons (N) of force can be applied until failure of the joining assembly will be in use during the testing. The test will not have a successful Newton measurement, but this will inform the group of how much tensile strength can be applied to the assembly prior to failure. If this test informs us of a number that is unappealing to a majority of group members, adjustments to the assembly design will occur in the Fusion 360 workspace again.

The last major test that will have to be completed is a thermal test. With the core concept of this being deployed in a themed land, temperatures of 100°F and above can occur in numerous places around the United States. This number is not the one that is concerning, the test comes to mind with a guest who might have left their personal Pokéball in their motor vehicle in an accidental case. On a clear sky day with the same temperature stated above, the interior of a motor vehicle can reach temperatures of up to 170°F or 76.7°C. A steady heat source will be applied to the exterior of the full mock assembly to validate that our prototype can withstand the temperatures that can be reached in a motor vehicle. A passing test would consist of the following:

- No major compromises to the integrity, density, and strength to the full assembly
- Normal operation and function as intended not being impeded by potential deformations to the assembly itself
- Access to the interior portions of the assembly for routine maintenance are not impeded by potential deformations
- Opening and closing of the locking mechanism still operates as intended

Once the above tests are completed and validated by all members of the group, the assembly will then be validated as a proper design. This will then allow for the final construction of the Pokéball to occur, with all articles, functions, and operations being involved.

9.0 Administrative Content

The administrative content section of this report will list parts that are required for the project which will include the Pokéball design and the RFID reading device. It will include most components needed along with the estimated purchase price. The milestones section will list the dates along with what will be completed. This is to ensure that the group stays on track for completion at the end of Senior Design 2.

9.1 Budget Estimate

Since there is no budget given by the university and our group does not have a sponsor, it will be very important to test the project using available parts that the group has. One member of this group has access to an Arduino UNO and MSP430FR6989 which will be useful for testing purposes and will not be included in the final budget estimate. A screen monitor is required for this project due to the interactive nature of the gym battles.

Product Name	**Quantity**	**Estimated Cost**
Arduino UNO Micrcontroller	1	$0
MSP430 Microcontroller	1	$0
3D Printed Shell	1	$0
RC522 RFID Module	1	$6
EM18 RFID Module	1	$4
Battery	1	$5-$10
Breadboard	1	$0
PCB	1	$10-$50
LCD Display	1	$12-$50
Passive RFID Tags	1-3	$5-$10
Miscellaneous Wiring / Parts	TBD	$0-$10
Total Estimate		**$42-$140**

Table 27: Budget Estimate

9.1.1 Final Project Budget

The table below shows the final project budget after the conclusion of the demonstration video. The items were grouped up into a section of components and not fully itemized to make the table easier to read.

Item	Cost	Description
PCB	$93.18	5x PCBs and Components
Pokéball	$65.00	3D Materials and Hardware
Base Station	$36.78	Components and Casing
Domain	$2.99	Ordered via GoDaddy
25GB Server	$10.00	Purchased via Digital Ocean
Total	**$207.97**	

Table 22: Final Project Budget

Comparing the final project budget to the estimate proved that the project did cost more than anticipated. If this project were to be utilized in a theme park, the cost of the full Pokéball would need to be reduced significantly so that park goers would be able to affordably purchase a Pokéball.

As stated in Section 9.1, this project was fully self sponsored so all costs and expenses were out of pocket or through other means. The Pokéball, for example, was able to be printed from a project member's job facility which allowed the cost for printing the shell to be cheaper than if purchased elsewhere. The PCB was purchased through JLCPCB and the Pokéball required sanding, priming, and painting which is why there is a cost for it.

The base station was able to be designed and built using parts that were available for purchase online. For the software aspect of the budget, the domain to host the project, which showcased the user registration and battle sequence, was required. In addition to the domain, the server is required to store the database and any relevant information. This totaling this project to just shy of $208.

9.2 Project Milestones

The chart below lists the milestones that need to be completed along with the dates associated. Since our project consists of four members, it is determined that each member is writing around 30 pages each to meet the 120 page criteria.

Senior Design 1

Brainstorm and Project Selection	**Week 1 & 2**
Divide and Conquer	**Week 3**
Divide and Conquer V2	**Week 4**
Technology Research and Material Requests	**Week 4 - 9**
Rough Draft	**Week 9**
Revise and include missing Pieces	**Week 10**
Final Draft	**Week 11-12**

Table 29: Senior Design 1 Project Milestones

A test prototype has been developed during week 10 of Senior Design 1. The prototype consists of a breadboard that connects the Arduino Uno, 1602 LCD, potentiometer, and RC522 RFID module. It has been tested by uploading a script from the Arduino IDE and verified that the components used do work together successfully. This will help reduce the amount of testing required during the prototype phase of Senior Design 2.

An initial software prototype has been developed during week 11 and 12 of Senior Design 1. The software prototype for the Arduino IDE included the scanning and identifying of RFID UID and prints the UID to the LCD attached to the Arduino. A graphical user interface has also begun development to allow the user to choose their character name and their starter Pokémon.

The most important factor to consider for Senior Design 2 is the time. If our group is unable to get materials in time, it will not be possible to complete the design in time for presentations. It is also important to have the bulk of materials ordered before Senior Design 2 begins. This is because parts may still be in high demand/low stock, and to avoid any complications, we must be aware of our material availability now. The parts that we know can be acquired without issues would include small buttons, shells, and aesthetic components to make it look nicer. Aesthetic materials are typically not an issue due to the fact that they are readily available (even at local stores) or they are a simple part to build. The milestone chart below ensures that the prototype and project can be completed in a timely manner. Of course it is subject to change based on any obstacles that may come by, but the idea is to allow room for any error by giving an extra week, or subtracting/combining others. Ultimate goal would be to complete the prototype as early as possible to give room for small fixes and functionality.

Senior Design 2

Order Materials	**Week 1 & 2**
Build ⅓ of Prototype	**Week 3 - 5**
Test and adjust	**Week 5**
Build ⅔ of Prototype	**Week 6-8**
Test and adjust	**Week 8**
Build complete Prototype	**Week 9-13**
Test and adjust	**Week 13**
Add any additional elements & create framework of report	**Week 14**
Final Report at presentation	**Week 15-16**

Table 30: Senior Design 2 Project Milestones

Appendix

References

Buchholz, Katharina, and Felix Richter. "Infographic: The Pokémon Franchise Caught 'Em All." *Statista Infographics*, 24 Feb. 2021, https://www.statista.com/chart/24277/media-franchises-with-most-sales/.

Figure 9

"Disneyland Will Support the Use of IPhone and Apple Watch to Enter the Park." *Real Mi Central*, 14 Mar. 2021, https://www.realmicentral.com/2021/03/14/disneyland-will-support-the-use-of-iphone-and-apple-watch-to-enter-the-park/.

Figure 7

"ISO/IEC/IEEE International Standard - Software engineering -- Guidelines for the application of ISO 9001:2015 to computer software," in ISO/IEC/IEEE 90003:2018(E) , vol., no., pp.1-86, 30 Nov. 2018, doi: 10.1109/IEEESTD.2018.8559961.

Figures 11 and 12

"ISO/IEC/IEEE International Standard - Software and systems engineering --Software testing --Part 1:General concepts," in ISO/IEC/IEEE 29119-1:2022(E) , vol., no., pp.1-60, 27 Jan. 2022, doi: 10.1109/IEEESTD.2022.9698145.

Figures 13 and 14

MuggleNet. "Spring Has Sprung: Interactive 2021 Collector's Edition Wand Available for Purchase." *MuggleNet*, 24 July 2022, https://www.mugglenet.com/2021/04/spring-has-sprung-interactive-2021-collectors-edition-wand-available-for-purchase/.

Figure 8

Patterson, Jenny. "Magicbands vs Magicmobile - What You Need to Know." *Inside the Magic*, 30 Mar. 2021, https://insidethemagic.net/2021/03/magic-bands-magic-mobile-jp1/.

Figure 6

Rasor, Drew. "A Comprehensive Guide to the Interactive Wands of the Wizarding World of Harry Potter!" *AllEars.Net*, 16 Aug. 2020, https://allears.net/2020/08/21/a-comprehensive-guide-to-the-interactive-wands-of-the-wizarding-world-of-harry-potter/.

Figure 10

General Arduino Tutorials
https://arduinogetstarted.com/arduino-tutorials

Arduino LCD Reference
https://arduinogetstarted.com/tutorials/arduino-lcd

Arduino RC522 RFID Reference
https://arduinogetstarted.com/tutorials/arduino-rfid-nfc

USITT Sound Documentation Recommended Practice
USITT DOCUMENT PDF

CFR TITLE 47 PART 15
FCC RFID STANDARDS

TEXAS INSTRUMENTS MSP430FR6989 ONLINE DATA SHEET
DATA SHEET ONLINE VIEWER

ARDUINO UNO REV 3 ONLINE DATA SHEET
DATA SHEET ONLINE VIEWER

Copyright Usage