Live Bolt Smart Lock and Security System

Linda Anyosa, Mohamed Faizel,

Andrew O'Reilly, Joel Shaw

Dept. of Electrical and Computer Engineering, University of Central Florida, Orlando, Florida, 32816-2450

Abstract — **Our objective is to create a safe and affordable smart home security system that features wireless communication between its components and companion mobile application. Many homeowners aspire to have an all-in-one security system that is within their budget and secure. With a growing and competitive market of security systems, there is a lack of secure solutions at a reasonable price point. What makes our project different than other security systems is how it incorporates home monitoring and access via our mobile application and wireless hardware components. As a group we wanted to challenge ourselves with a growing problem.**

***Index Terms* — Building services, Identification of persons, Image sensors, Infrared surveillance, Internet of Things, Mobile application development, Motion detection, Security**

I. Introduction

We decided to name our Smart Lock System, Live Bolt. One of our main goals we'd sought to accomplish in creating Live Bolt would be to ideally have a fully functioning and secure system to showcase. We also sought to create a mobile application with an aesthetically pleasing user interface and functionality to promote user usage. As a team, we know that installation of any home system requires handy tools, so we have designed both the security and home monitoring box to have installation ease within 3D printed enclosed boxes to secure our physical electronics.

To share the data between the hardware and software of the system, we decided to opt for a wireless connection between our two essential boxes. Our two lockboxes, one controlling the deadbolt attached to the door lock and the other for monitoring activity outside the system work together to provide a comprehensive and fluid experience to the user. Additionally, the companion mobile application gives users much customization and control when interfacing with the system. As a package we wanted to design the Live Bolt system in such a way that it would be easy to set up without having to be bogged down with wires and outlets. We discuss in detail more about our software and hardware components later in our paper.

In terms of design constraints, we had to be realistic about taking them into account. These included the necessary wireless range access for when unlocking via the application, whether weather conditions could significantly affect system performance, would our system be easy to understand and visual appealing to our users, and how easy would it be for the batteries to be replaced. Ethical challenges from the construction of the system would also have to align with the path of preventing security breach. Security is important and crucial to a system in which users are trusting to safeguard their homes. Users are also trusting us with information personal to their person. We have taken these and other factors into account when designing our system. As such external data has been stored for citational and referenced purposes only to keep the consumers privacy secure.

The main motivation for creating this project was from our individual experience in package theft. 49 million Americans alone have experienced Package theft in the last year [1]. As we were working on our project this semester, we had an incident of package theft where part of our hardware was stolen. These details along with our research led us to think Why not monitor/ report any front door activity through a security system? From this question, we came with the idea of building a security system to monitor and allow front door access to delivery workers with a temporary one-pin code to reliably deliver the package within a home. Although this idea granted entrance, there must be an automated lock to open and close the door for a home. Therefore, we decided to expand upon this project to include variety of access control for the consumer. It logically makes sense to not only have a locking mechanism for delivery worker but to include forms of entrances for the consumers own usage. Additionally, we wanted to introduce a software application to harness the control the user had over the entire system. Additionally, this motivation of creating an application would challenge and utilize the team's software skills in building a reliable security system that could potentially be used in industry. We sought to challenge the current security market by creating a new set up with an app and hardware system interface. In doing so, we would keep the system as user-friendly as possible while keeping it as secure as possible. We have achieved these goals by creating a lightweight security system that can be placed in desired monitoring areas (e.g., home, shed, garage), distributing power to electronic hardware, and establishing a connection between the lock and phone application. While doing so we have also been able to keep the cost of hardware low while maintaining quality to make this product accessible to low-income individuals.

How we plan to target our objectives has been by researching existing home security and locking systems in the market. In access control we considered scope upon versatility about how a homeowner would like to open their front door. In security measures, we'd researched forms of home monitoring and reaction to keeping a property secured. While these two would create most of the hardware for the project, we need to make it available through a mobile application for the customer convenience. Objectives to execute that action continue to be to adhere to commonly held design standards to usability, have a secure software and hardware connection, and include multiple methods for accessibility. This has also factored into our choice of buying electrical/mechanical parts. For the requirement of including a designed PCB, we have tried to make this compact as possible to achieve our objective of a compact system. From the team's knowledge and capabilities, we will ensure through testing that we will deliver a fully functioning product that our customers will love.

II. System components

A. MICROCONTROLLER

a. ESP

For communication between the security module located outside and the lock module located inside, each module has an ESP-12F microprocessor chip. The ESP-12F functions essentially the same as any Arduino chip like the Atmega328, except it also has Wi-Fi capabilities. There is a breakout board for development that is called the ESP8266 NodeMCU. For the final product, the ESP-12F chip will be desoldered from the ESP8266 and soldered to the Live Bolt PCBs. It has an 80MHz clock frequency and 9 GPIO pins. It lacks any Bluetooth capabilities and operates on the HL20 20MHz spectrum of the 2.4GHz Wi-Fi band. The block diagram in Figure 2 shows how the ESPs function with the rest of the system. The security ESP take signals from the app or keypad and sends a signal to the lock ESP whenever the proper criteria is met to turn the servo to either lock or unlock the door.

b. RASPBERRY PI

The Raspberry Pi will be used to host the security system, which includes the security camera and PIR sensor. It will also allow communication between the ESP chips and the mobile application. The Live Bolt system uses a Raspberry Pi 4B+. Whenever the PIR sensor is triggered, it will turn a GPIO pins voltage on the Raspberry Pi to HIGH. When the Raspberry Pi detect this, it will trigger Python code to start recording from the camera module. The other feature of the Raspberry Pi is being the bridge between the security ESP-12F module and the application. The Raspberry Pi and ESP-12F are connected via Wi-Fi, and the Raspberry Pi is also connected to the application. With this setup, when the user presses the unlock button via the application, the application will send a signal to the Raspberry Pi to tell the ESP-12F that the unlock criteria have been met.

B. MOTION SENSOR SYSTEM

As an addition to Live Bolt, the system is made for home monitoring; therefore, for the system to detect front door activity, our design would need to correspond to a motion sensor system. We've selected the passive infrared (PIR) sensor that will be placed into our security system for proximity detection. The PIR sensor chosen was the Stemedu HC-SR50. It is known for being energy efficient, high quality, and provide long service life. From the adjustment settings in the package, we've determined to use 3 meters from the sensor, 5 second delay, and including the 100-degree coverage. This choice will allow us to detect motion at the front door and relay a captured image using the raspberry pi camera. We will be using the pins from the pi to have that communication with the PIR sensor and send to the application.

C. KEYPAD

Since we're using a 4x4 Arduino keypad, collecting the reading of the number and/or letter is a simple connection based on the division of rows to columns. When no buttons are pressed, the columns are read as HIGH, and rows are read in LOW. When a button is pressed then the column is pulled to LOW from the current. It'd know the column in place so the roles would switch then the rows on HIGH. From the rows on HIGH and columns on LOW, the ESP-12F chip can find the location of the button when the column pins retrieve back to HIGH. In our design, we will use this to configure onto the ESP-12F chip and successfully determine whether the code has been inserted correctly. From this observation we're connecting this keypad to our ESP-12F security chip to talk to the ESP12-F lock chip to open and close the deadbolt lock.

D. RASPBERRY PI CAMERA

Another way we want to maximize the use of the Raspberry Pi is by including a Mini Camera Module. It's a low-cost add-on that is compatible with the models we're using. This camera provides an angle view of 54 x 41 degrees with the capability of processing videos of 1080p30, 720p60, and 640x48-p60/90. The camera connects to the BCM2835/BCM2836 processor on the Pi via the CSI bus which carries a high bandwidth link of pixel data to the processor. In our project, the raspberry pi camera will be enabled once it detects front door activity and passes on a captured image to our mobile application. The camera will be a part of the security box and provide home monitoring accessibility to the user.

E. SERVO MOTOR

After researching similar smart lock projects, we chose this a high torque servo motor to turn the deadbolt knob. The servo motor of choice provides a rotation of 120 degrees (60 in both directions) with the flexibility of up to 170 degrees. It is light weight of 62.14 g with the size dimensions of 40.7x19.7x42.9 mm. Aside from the dimensions of the motor, the packaging came with plastic horns. We took advantage of those horn to fit the shape of the 3D printed knob which can be elaborated in the topic of design.

The servo motor will be attached to our lock box PCB and perform that task to turn the knob when recognizing access entry. Our PCB will include an ESP-12F chip to ping the servo motor to turn based on the user's choice to access entry. How it will be powered is through a 12V battery pack but will truly need 5V to operate. PCB design choice is further elaborated in our PCB design section.

F. 3D PRINTS

To house all the hardware components, the team has decided to construct a housing compartment from 3D printed filament. The boxes will be split between the locking mechanism and security electronics. To hold value in our introduction, we chose this method to hold to our goal of making this product lightweight and portable.

When designing a 3D print, things that need to be accounted for is printing size and filament choice. With those in mind, we design our boxes to be within the constraint of a typical 150x150x150 mm printer and chose PETG for the filament. PETG was chosen for its durability in weather conditions considering our security box will be placed outdoors.

III. SYSTEM CONCEPT

To determine how we'd construct Live Bolt, we highlight import aspects to what we wanted our system to include, therefore, we built a house of quality for to ensure the best engineering and marketing quality of our product. Using a house of quality matrix allows us to devise a strategy on how to rate the importance of our product, in a customer standpoint, for quality and user preference.

Before beginning to identify the importance for each requirement, creating a legend for the correlation, relationship, and improvement of direction for our system is indicated below the table. In our table we highlighted what was stated in our introduction such as portability, security protection, sensors, battery life, etc., in engineering requirements. Now this must align with customer requirements within the quality or product and user preferences. This would market to customers with the understanding that our team puts their usability/ flexibility first. With the weight column ranging from 1-10, we identify cost and user interface as top priority to make it accessible and user-friendly as possible to be purchased by a larger audience.

Engineering and customer requirements go together when finding the direction of improvement needed to be done in our system. The house of quality outlined in Figure 1 shows ways we can improve our design while considering a consumer desire.

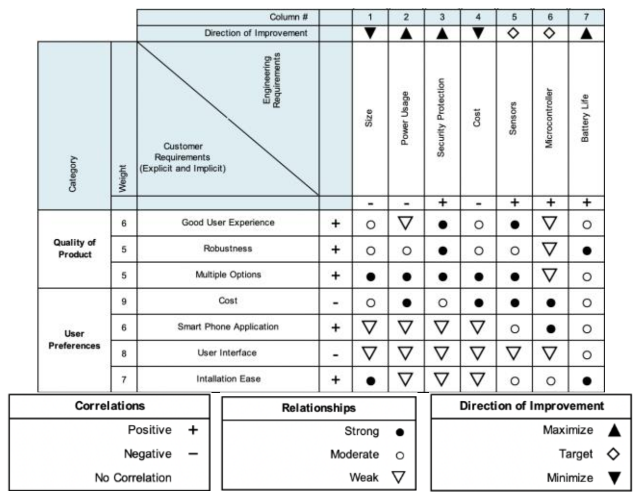


Figure 1: House of Quality

A. SYSTEM HARDWARE CONCEPT

The hardware of the system has gone through multiple adjustments to solidify for the build for Live Bolt. We finalized our hardware interaction within both the security and lock in the hardware diagram below. Within the security box, we have multiple electronics in place to provide home monitoring while configuring this data to the application. Following is the lock box which houses the Lock ESP-12F servo motor and battery pack. The arrows pointing through the diagram in Figure 2 represents their relationship within the system and whether their representation in the graph is an input or output.

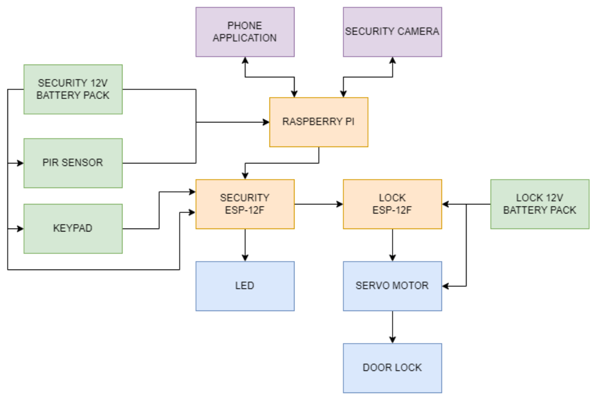


Figure 2: Block Diagram

IV. HARDWARE DETAIL

A. SERVO MOTOR CASE/ SERVO HOLDER

To turn the deadbolt knob, we needed to provide a housing for the servo motor to hold it in place. Holding it in place would allow the motor to work effectively to turn the knob. This design can be seen in Figure 3.

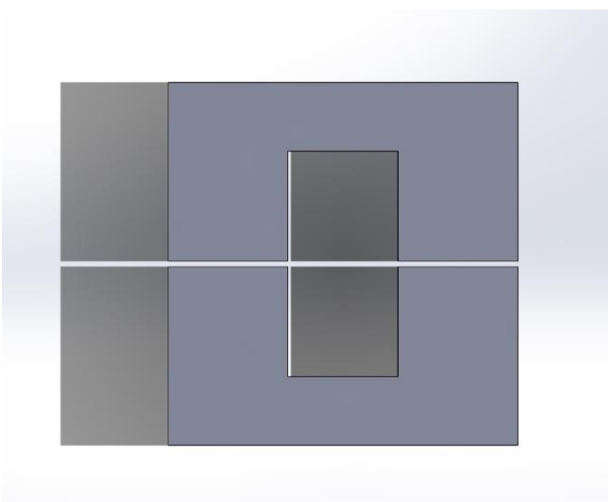


Figure 3: Servo Holder Model

In addition to our servo motor case, we have a knob design to enclose on the deadbolt knob. We chose a design that mimics the standard shape of a knob to provide the effectiveness of opening and closing the door. The knob enclosure can be seen in Figure 4. These two parts have been modeled using SolidWorks and are 3D printed. These parts will be integrated into our lock box.

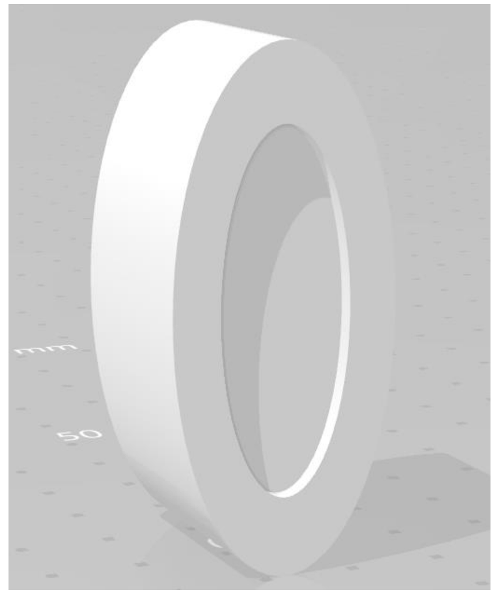


Figure 4: Knob Holder Model

B. SECUIRTY BOX DESIGN

Behind the security box, for Live Bolt, it was designed to accommodate the electronics to encompass home monitoring, home entrance, and wireless updates to the application. After collecting part dimensions and using the CAD software, SolidWorks, we were able to begin the design phase.

Outside of our security box we have a PIR sensor, keypad, LED, and raspberry pi camera. A consumer of Live Bolt will be able to see these features through cut out portions of the 3D print. Within the security box, it will house our PCB, 12V battery pack, and raspberry pi. The total sizing of the box will be approximately 150x82 mm. The complete model can be seen below in Figure 5.

The security box has an additional feature where a user can access the battery. The box will include a feature that will be able to open and close securely with a back plate. This back plate will also include drill holes to mount onto a wall/ door. This gives the user the opportunity to place their security box in their desired location.

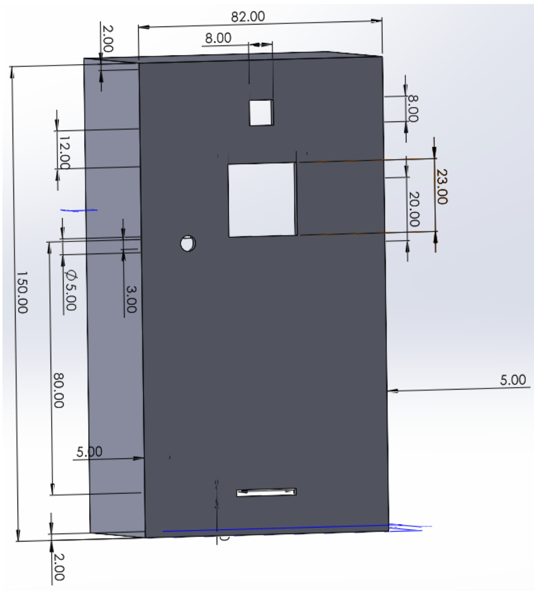


Figure 5: Security Box Model

C. LOCK BOX DESIGN

Our lock box design comprised of three main objectives: housing the electronics, accessing the battery, and holding the servo motor. In this 3D print, we've sized the box to fit all electronics; include a back sliding panel to access the battery; and a 3D print to hold the servo motor in place. Accounting for all these feature, total size of the box came to be 145x80x65 mm.

Using SolidWorks as our main CADing software, we were able to mockup the design of the box as shown in Figure 6. The circular cut in the box is what will close on the deadbolt lock.

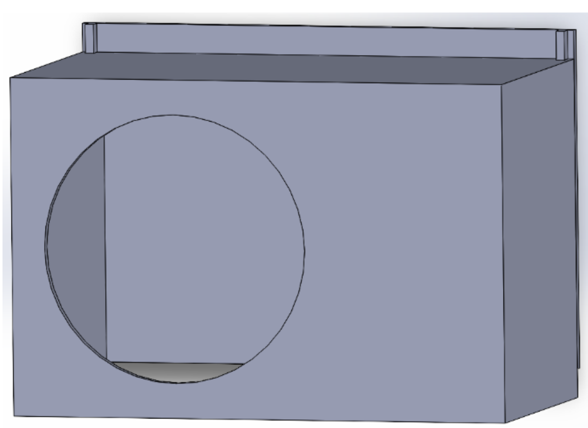


Figure 6: Lock Box Model

V. POWER DETAIL

Our system is complex in the sense that each component will require a voltage input to keep the system running. Since our design requires microcontrollers, sensors, Wi-Fi module and a servo, it's essential to make sure that each respective component will function properly and run given the power source provided. The main components that need direct power from our battery packs are the ESP-12F modules, which run off 3.3V, and the Raspberry Pi and Servo motor, which both need 5V to function. Both modules will be powered by a 12V AA battery pack. The reasoning for 12V is because when testing with a 6V pack, there was not enough consistent power to power the servo motor, which caused jittery movements. The 12V packs will then be stepped down to 5V through a LM7805 [2] regulator, and then stepped down once more to 3.3V using an LM11733 [3] regulator. Both regulators have bypass capacitors at their inputs and outputs.

VI. SOFTWARE DETAIL

When designing our user experience, our team's philosophy was focused on readability and consistency. Our aim was to create an app that felt like a natural extension of the capabilities of the physical Live Bolt. We initially considered creating a web application instead of a published mobile app in order to save time on development. However, there are many issues to consider when creating a web app meant to be accessed by phones. Since many phones have unique screen sizes, the native web app would have to adjust to every screen size in order to provide consistency. This would be difficult to achieve without using a CSS library that automatically scales UI elements, such as Bootstrap. Even so, our team determined that using a native web app was not worth the tradeoff in customizability. There are many smart lock companion apps on both the Google Play Store and the App Store, so we desired to create an experience that would be immediately recognizable and unique for the consumer.

A. MOBILE APP FRAMEWORK

We spent time looking at a few popular application stacks when it came to designing our application. These included the LAMP, MERN, and MEAN stacks. After considering our options we decided to go with the MERN stack. The MERN stack comprises of a MongoDB database, a Node web server, an Express web framework, and a React client-side. Since most of our team has a measure of familiarity with developing a MERN stack in previous classes, namely in Processes of Object-Oriented Development, there was not as much learning required to create our application stack. Our team researched several frontend frameworks and libraries, both cross-platform and iOS/Android only, before settling on React Native. React Native is a cross-platform JavaScript UI library currently maintained by Facebook. We decided to use React Native as the foundation for our frontend due to its portable nature and its superior performance to other UI frameworks such as Flutter and Xamarin. Additionally, React Native has a variety of developer tools and modules for every aspect of mobile development and testing.

B. REMOTE DATABASE/BACKEND

The backend components of our MERN stack mobile application included server-side technologies such as Node and Express, which some of our members already had experience working with in previous coursework. Node is a very popular JavaScript backend runtime environment, which enables our team to access a large amount of existing documentation. Express is a minimal and lightweight Node framework that provides many of the features that our mobile application will be using. Additionally, both Express and Node sport fast response times when communicating with other technologies. This backend application is being hosted on Heroku's platform as a service (PaaS) that allows us to run and operate our backend completely on the cloud. Using the currently available free tier, Heroku is able to connect to our backend repository and host it on their servers, as opposed to one of our team members needing to continuously host the application on their own machine. In order to store any user data that our backend application creates or interacts with, the Heroku server is regularly connected to another remote service, a database hosted by MongoDB. MongoDB is a fast and scalable distributed database which makes it an ideal choice for our smart lock. Since MongoDB is a document-based no-relational database system, it does not use SQL syntax to manage its collections, making it much easier for our team members to use and maintain. Internally, our database's structure is very simple with three separate collections: user information, user activity, and lock PIN codes. Each of these collections have smaller tables for specific uses, ensuring that our application is able to efficiently access and retrieve information. For example, the user information table has a separate document for each account, which contains fields such as username, password, and email address. Figure 7 displays an entity-relationship diagram (ERD) of the user information table.

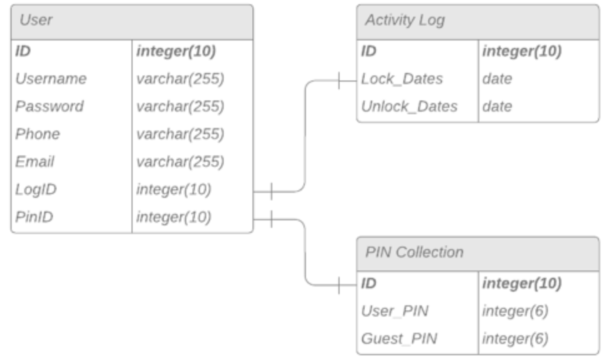


Figure 7: Entity Relationship Diagram

C. USER EXPERIENCE

At the beginning of development, our team used Figma to create rough designs for several of the pages for our app. Figma is a web-based graphical tool that can be used to create both app designs and prototypes. Using the free starter level functions, our team was able to come up with a variety of potential user interfaces. Additionally, Figma also has a companion app on iOS and Android that allows users to see how their designs look on phone screens. Once the initial color palette and design motif was decided, our team set about creating the app. On the frontend, a variety of JavaScript packages were used in order to regulate user input and provide a seamless user experience. The figure below shows the login screen for the companion app on an iPhone 13. As shown in Figure 8, the color scheme mostly consists of shades of blue and white for most of the text and buttons. Our team opted for a more simplistic aesthetic which would avoid potentially alienating or overwhelming users, making each screen clear in both style and purpose.

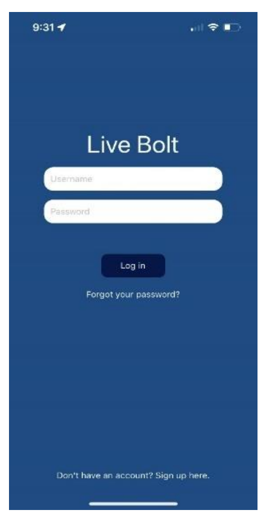


Figure 8: Login Screen

As the Live Bolt Smart Lock's hardware design places, a priority on security, so does the software design of its companion app. User information is fetched from the remote database using JWT standard to ensure that personal data is transmitted securely from client to server and vice versa. Additionally, passwords are hashed server-side to prevent leaking of user information and are only decrypted on the backend. Accounts are verified by successfully entering a 6-digit PIN contained in an email sent to the account's associated email address.

VII. PCB DESIGN

As previously mentioned, the system will include two separate modules. This means there are 2 separate PCBs required, one for the Security Module and one for the Lock Module. Both circuit boards are very similar. They both have pads for a 12V battery pack, a footprint for the ESP-12F, voltage regulator circuits and associated bypass capacitors, and finally a 6-pin header with associated 10K Ohm pull down resistors to attach a separate debugger if there is a need to reprogram anything after the ESP-12F has been soldered on.

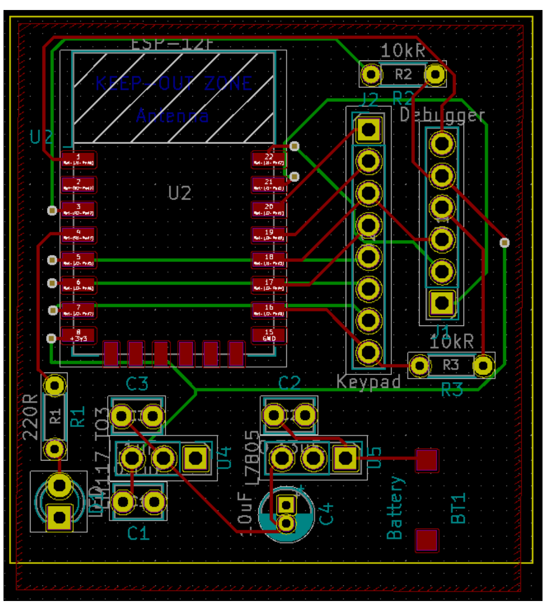


Figure 9: Security PCB

The main difference between both PCBs is that the Security PCB has an 8-pin header for the keypad and a spot for an LED, while the Lock PCB has a 3-pin header for the servo motor. The Security PCB can be seen in Figure 9 and the Lock PCB can be seen in Figure 10.

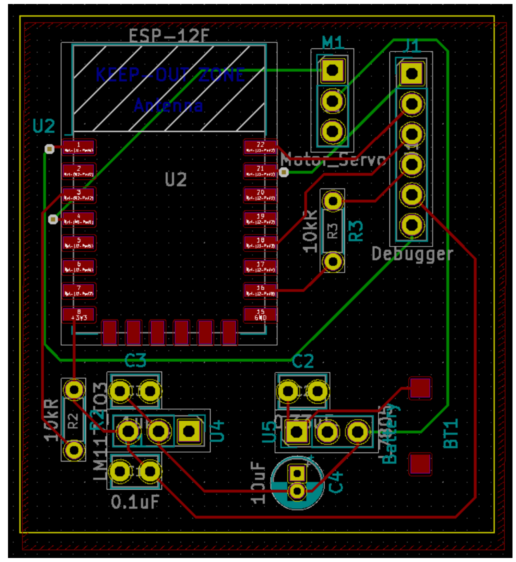


Figure 10: Lock PCB

VIII. TESTING

A. SOFTWARE TESTING

We considered a few testing frameworks and libraries for testing our mobile application. These included Jest, Mocha, Cypress, Jasmine, and Selenium. While developing the mobile application manual testing was done to determine the components functioned as intended. These included the frontend React App components, communication across the API, and our MongoDB database. For our finalized testing we have been using Jest in conjunction with Puppeteer. Both are JavaScript Testing Frameworks that serve our purposes well in testing the app from the front end all the way to the Database. Jest functions as a testing tool with simplicity in mind allowing for test to be created and run with ease. Additionally, having experience with Jest in prior coursework made the creation of the testing environment much easier to get started with when created. Puppeteer is a Node.js library that allows for an API control over the DevTools Protocol in Chrome and can allow for UI testing, automated form submission and keyboard input verification. This has been very useful when it comes to validation of the login, forgot password, and verify email API and components.

We also tested the Software connection to the Hardware like communication between the mobile app to turn the servo or sending the camera data from the Raspberry Pi to the mobile application. These tests we mostly manual to ensure that the correct functionality was occurring within an appropriate period for the User. Jest and Puppeteer will also be extended to these connections were appropriate to further ensure functionality for things we could miss in our manual testing.

B. Hardware Testing

Testing for Hardware was done through assembling the components on a breadboard. The breadboard circuit was the basis for the PCB schematic seen in the PCB section. Figure 11 shows an image of the completed breadboard prototype. For testing, only one battery pack was used to keep the wiring less complicated. From the picture, the ESP-12F chips are still on the ESP8266 development board. This was done because the included pin headers on the development board made it much easier to insert into the breadboard, as opposed to the surface mount pins on the ESP-12F. This breadboard prototype was very helpful in determining what was going well and what needed to be improved for the circuit. As previously mentioned in Section V, the servo was not acting as expected with a 6V power supply, and we needed to upgrade to a 12V battery pack. This issue was discovered during this phase of testing.

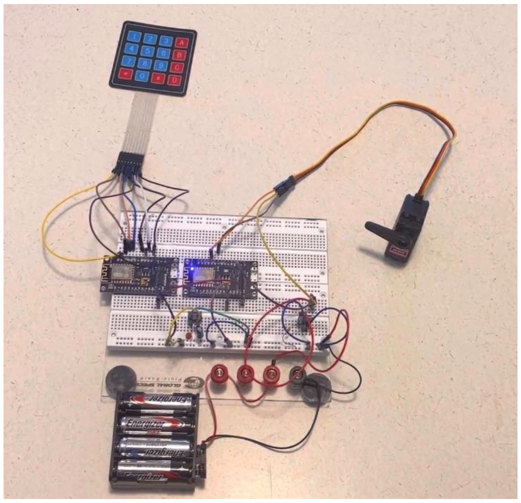


Figure 11: Breadboard Hardware Prototype

Another helpful part of the hardware testing phase was being able to use the equipment in the University of Central Florida Department of Electrical and Computer Engineering Senior Design Laboratory. There are digital multimeters, benchtop power supplies, soldering stations, and common electrical components like resistors, capacitors, and wires, that we were able to utilize while ironing out the design for the Live Bolt circuit [4].

IX. Conclusion

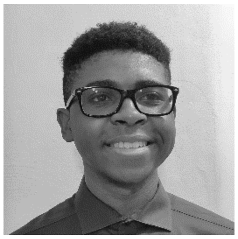
It's been a learning experience working on this project together as a group over the course of this past year. Over the course of the year, we have learned a lot about Software, Hardware, Internet of Things, and Teamwork. It was through a variety of different factors that ultimately led to our successful completion. Weekly meetings over discord and in person on campus helped us to stay on track over the course of both the Spring and Fall semesters of this year. Additionally, in Senior Design 2 we made sure to meet more frequently as the semester went on to ensure we were on track with our project. We also made opportunities for each other to spend time outside of Senior Design work to get to know each other better. We also met a few times in the summer to finalize anything we need to have done before starting the Fall semester. By assembling most of our parts before the beginning of the Fall semester we were able to get a head start on assembling our hardware components. This included building out the Website we to house our project, gathering of any extra parts, and discuss start of semester tasks. We also tried to assign everyone to a task that they would enjoy doing as well as was aligned wit their skillset. Doing so made it easier for us to stay engaged with what we were working on while making continued progress overall. Additionally, we made it easy for anyone who wanted to cross over into another area of the project to work with that area. It's been a satisfying journey working on our project and we are proud of our project.

Acknowledgement

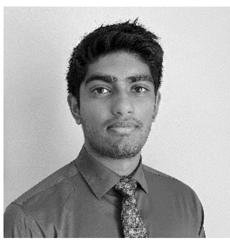
The authors wish to acknowledge the assistance and support of the UCF ECE Senior Design department, as well as the support and comments made by the team's review committee.

The Engineers

Linda Anyosa will be graduating with a bachelor's degree in Electrical Engineering focusing on Power and Renewable Energy. During her undergraduate degree, she interned with NASA, Northrop Grumman, and SpaceX. After graduation, Linda's career goal is to work in the space industry supporting launch missions in power engineering.



Joel Shaw will be graduating with a bachelor's degree in Computer Engineering on the Comprehensive Track. He has completed a few internships during his time at UCF ranging from Software Development to developing Digital Applications. He has also conducted research with the PREM center at UCF analyzing Data gathered from students in undergraduate courses. Some of his interests include Embedded Systems, Data Analysis, and Software Development. After graduation he will be working for Lockheed Martin as an embedded software engineer and in the future plans to pursue his master’s after getting more real-world experience.



Mohamed Faizel will be graduating with a bachelor’s degree in Computer Engineering on the Comprehensive Track. Some of his interests include Web and Mobile Development, Embedded Systems, and Firmware Development. After graduation, he hopes to pursue a career in either web development or systems engineering.

Andrew O’Reilly will be graduating with a bachelor's degree in Electrical Engineering on the Power and Renewable Energy Track. He has completed past internships as a Project Engineer Associate at Mitsubishi Power Americas and as a Reliability and Maintenance Engineer Intern with Amazon’s Solar Program. He has also held positions at the Florida Space Institute and the Center for Microgravity Research as a Research Assistant. After college, he hopes to pursue a career in the solar energy industry.

References

[1] "2022 Package Theft Annual Report," security.org, 22 July 2022. [Online]. Available: https://www.security.org/package-theft/annual-report/. [Accessed 10 November 2022].

[2] "7805," [Online]. Available: https://www.sparkfun.com/datasheets/Components/LM7805.pdf.

[3] "LM11733," [Online]. Available: https://www.ti.com/lit/ds/symlink/lm117.pdf.

[4] "Senior Design Laboratory," [Online]. Available: https://www.ece.ucf.edu/academic-laboratories/senior-design-laboratory/.

[5] K. Pattabiraman, "Circuit Basics," [Online]. Available: https://www.circuitbasics.com/how-to-set-up-a-keypad-on-an-arduino/. [Accessed 2022].