Modularized Electronic Locker

Jack Davis

Jake Pu

Josh Nolan

Final Report for ECE 445, Senior Design, Spring 2021

TA: Ali Kourani

05 May 2021

Project No. 61

Abstract

The Modularized Electronic Locker is a secure package delivery system intended to serve a wide audience unlike other solutions on the market. With doorstep delivery being a rapidly growing industry, this project aims to securely protect delivery contents from the mounting issue of package theft. The modular system contains one control module and up to 127 additional lockers. The control module houses the primary interactive components such as the touchpad and keypad which are used to input delivery and pickup codes which are synchronized to a cloud database. The system is designed such that multiple users are dynamically assigned a locker upon each delivery and each individual can pick up all of their packages once their pickup code is entered to the control module. The following report details the motivation, design, and verification of the system. Further details outside of this paper can be found in our GitHub repository where the source code and YouTube demo are located [8].

Table of Content

1.Introduction

Package theft is a common problem both in the US and abroad. According to a recent survey, 43% of American consumers claim they have had a package stolen in the past year[1]. As ecommerce becomes more prevalent, porch piracy will only become more prevalent. In 2020, consumers spent $862.12 billion online. This represents a 44% increase from the previous year[2]. While online shopping due to the pandemic certainly contributed to this number, our reliance on online merchants is only expected to increase over time. The objective of our project is to create a system of modular electronic lockers that can be used to protect deliveries to homes, apartments, etc. These lockers will provide a safe haven for a wide range of deliveries (Amazon, UPS, Uber Eats, etc.) as well as provide package delivery notification through a network connection. While similar solutions like Amazon Hub do exist, these systems are large, expensive, and not suitable for individual homeowners or small apartment buildings. The modular design of our electronic lockers will be able to provide affordable delivery protection for a one person household or an entire apartment building.

Students living off campus without a packaging station are affected by stolen packages all the time [3]. As a result of privacy concerns and inconsistent deployment, public cameras in Champaign and around the world cannot always be relied upon. Therefore, it can be very difficult for victims to gather evidence for a police report. Most of the time, the value of stolen items is small and they are usually compensated by the sellers (Amazon and Apple are very understanding). However, not all deliveries are insured [4] and many people are suffering from stolen food deliveries during the COVID-19 crisis [5]. The smart lockers in the market do not address expandability which makes it difficult for locker owners to increase locker capacity. The design of this locker will accommodate many different use cases with its modular design to keep packages safe.

2.Design

2.1. Introduction

Our design consists of three distinct modules: a control unit, cloud server, and locker module. In the early phases of the project power was given its own module, however it's simplicity did not warrant one. The control unit contains a touchscreen LED display and keypad for user input, a camera, ROM storage, a PIR motion sensor, and a Raspberry Pi. This unit is responsible for taking in user input, controlling the locker modules, and storing user data. Locker modules ensure package security and provide modularity to support numerous applications. Lastly, the cloud server module is responsible for receiving photo data to the cloud as well as send pickup/dropoff information when a data request is sent from the control module.

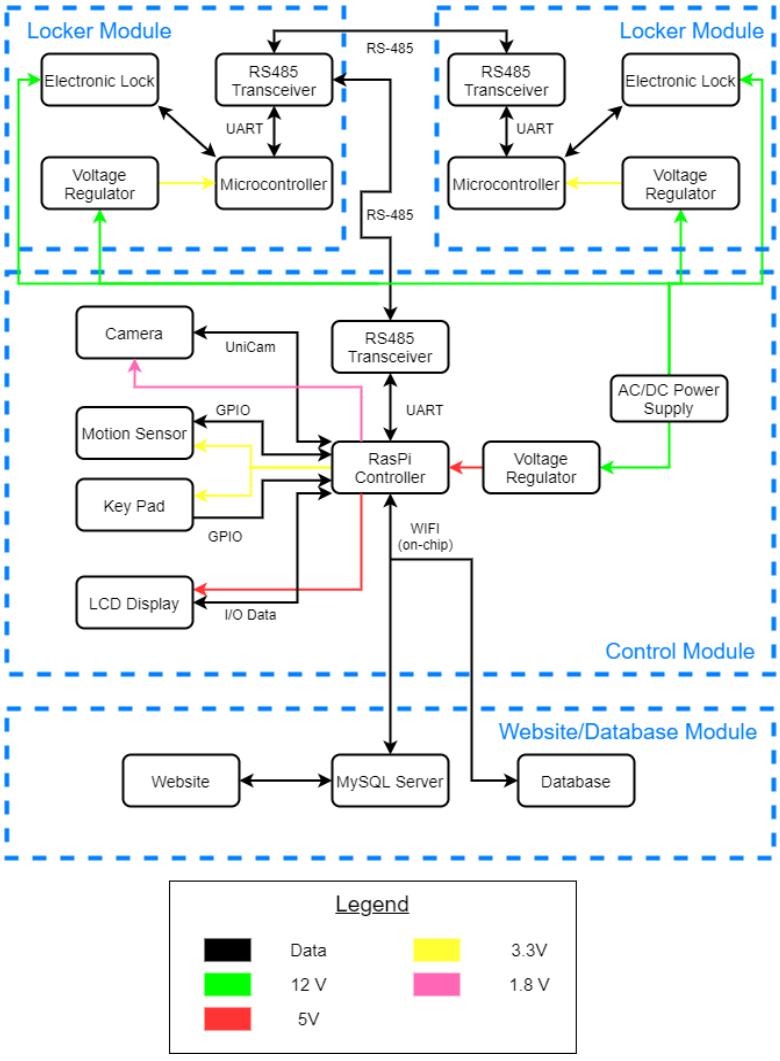

Figure 1. Block Diagram

2.2.Control Module

2.2.1.Power

The power delivery system in our system was simple but robust, consisting of a single off-the-shelf 12 V AC/DC power supply connected to an off-the-shelf 12 V to 5 V voltage regulator capable of consistent 6 A drivery. The AC/DC supply is capable of steady 12 V and 3 A delivery. This exceeds the approximated 24W of power in the system, and leaves enough room for the Raspberry Pi to render graphics and drive the LCD display. While previous locker projects elected to provide battery power to their projects, we decided that as a locker that it should have access to wall power for the majority of its operating time. Furthermore, when cut off from power the contents of the system remain secure until power is restored.

2.2.2.Security

Security in the control module consists of two primary components: an infrared motion sensor and a 720p camera. The motion sensor outputs a digital voltage signal; when motion is detected it outputs a high signal and is held for a controllable period of time until the motion stops. Further details on the output sensor can be found in 3.1.2. While the output is held to the high position, the Raspberry Pi signals to the camera to capture a photo once every 5 seconds. For the storage of the photo, it is first stored on the local storage up to 64 GB. For further security, the photo is backed up on the cloud server.

2.2.3.GUI/Raspberry Pi Software

The touchscreen GUI is the interface through which users and delivery services interact with the lockers. It was created using the python GUI library tkinter. The GUI consists of five pages (home page, pickup page, dropoff page, admin sign in page, and admin control page). Each page instructs the control unit of the locker to perform different functions. All five of the GUI pages can be viewed in Figure #2.

The home page of the GUI is simply made of three buttons that allow you to navigate to the proper page depending on what actions you intend to perform. Whenever someone navigates to the home page, any codes that previously existed in the textboxes of other pages are immediately deleted to protect the pickup and dropoff codes of users.

The dropoff page of the GUI instructs the user to enter the dropoff code that they were given into the text box using the keypad. Once the “Enter Code” button is pressed, the

control module first checks if the dropoff code entered matches any existing dropoff code in the MySQL user table. If it does not, a message pops up informing the user that the code was incorrect. If the code is correct, the software next checks if there are currently any lockers available. In the case that there are not, a message is displayed informing the user that there are not currently any available lockers. When a dropoff code is both valid and there are one or more empty lockers, the email of the user associated with that dropoff code is pulled from the MySQL user table, that email is placed into the occupying email column of the associated locker in the MySQL backup table, the locker's status is changed occupied locally, and a message is sent on the bus to open the specified locker. An email is also sent to the customer informing them that their package has been dropped off.

The functionality of the pickup and dropoff pages are almost identical; however, when a pickup is performed the specified locker is changed to vacant locally and the user's email is deleted from the MySQL backup data table. If the user has more than one locker to pick packages up from, the two lockers will open up sequentially on a time delay of six seconds.

The admin sign in page allows users with admin access to sign in and be granted access to the admin control page. After an admin code has been entered and the "Enter Code" button has been pressed, software first queries the MySQL users database to see if the given code matches any admin codes for the locker. If the code is valid, the user is given access to the administration control page. Just as in the dropoff and pickup pages, whenever the "Enter Code" button is pressed on the admin control page, the code in the textbox is immediately deleted to prevent code theft.

The admin control page allows users with admin privileges to set up new lockers and to open all existing lockers for maintenance or in the event of a system failure. By clicking "Start Setup Mode", the lockers system deletes all local locker memory as well as that in the MySQL backup database. Lockers can then be set up by pressing a button on their associated PCB which will cause the address of that locker's microcontroller to be sent on the bus. After the buttons of each locker have been pressed, clicking "End Setup Mode" will create empty locker objects for each address that was received. The data in the MySQL backup table will also be updated to represent these changes. If the "Open All Lockers" button on the GUI is pressed, all known lockers will open sequentially on a time delay of six seconds.

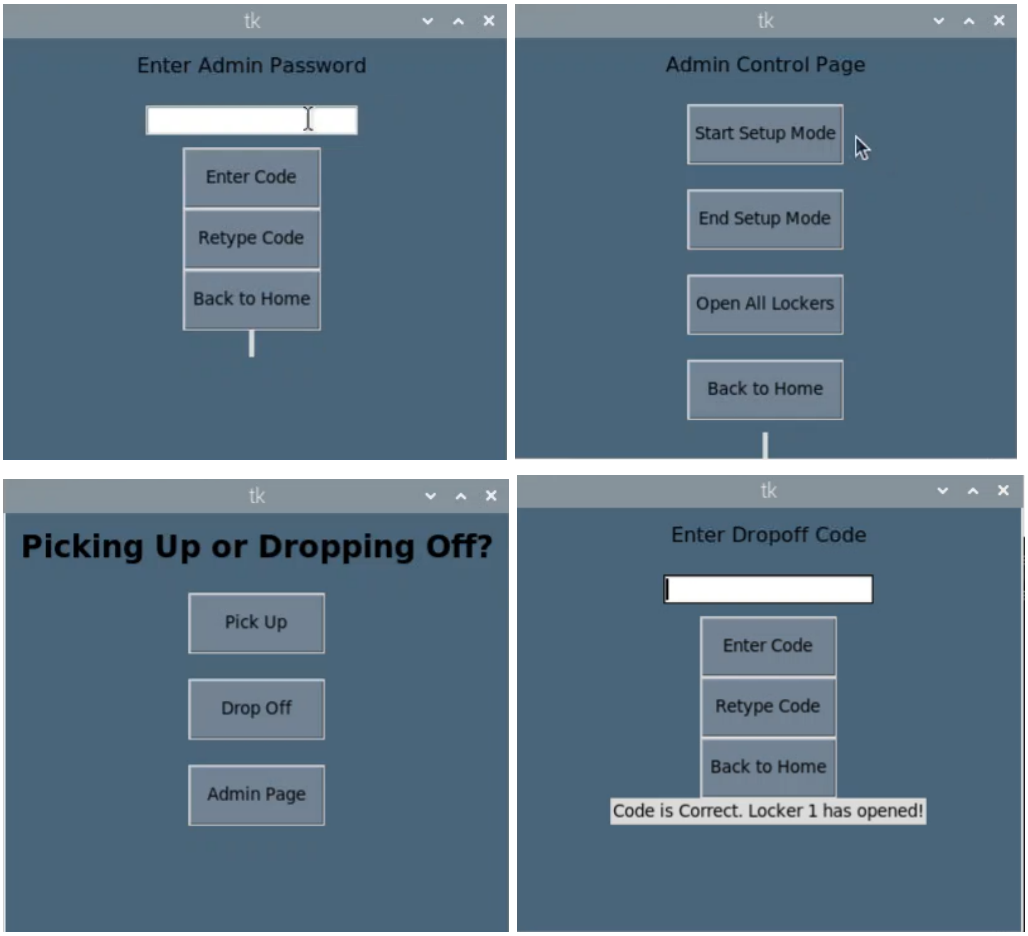

Figure 2. Screen captures from the GUI pages.

2.2.4.Keypad

The keypad is used in conjunction with the LCD display. While the touchscreen is theoretically capable of the user to input each of the digits into the screen, the mere 3.5” diagonal display size would make the task difficult. To supplement this, the keypad inputs to the Raspberry Pi. A script on the Pi continually reads the 8 pins on the keypad to detect two completed circuits corresponding to the row and column of the key pressed. When a keypress is detected the RasPi outputs the digit to the text field in the GUI.

2.2.5.Communication Bus

The modularity feature depends on the correct selection of communication bus protocol. The protocol must support multiple devices on the same bus while allowing bi-directional communication. The protocol must also have a transmission distance of at least 6 meters since this is the longest distance a signal has to travel in our product design. We chose UART over RS485 signal standard for our product. The RS485 can support up to 128 devices on the same bus and can transmit a signal for up to 4000 feet at 90Kbps. Since

our design involves a 2D bus network, we do not have impedance matching nor termination at the end of the network to minimize reflection of the signal. Therefore, in our implementation, we chose a baud rate of 115200Hz which makes our maximum expected signal travel distance (6m) much less than 1/10th of the wavelength (2602m) to minimize the effect of signal reflection while ensuring quick data transmission.

2.3.Locker Module

2.3.1.Lock Control

The solenoid locks are shaped like typical lock bolts and operate similarly as well. While unpowered the solenoid extends to the locked position and while powered with 12 V and 0.43 A the solenoid retracts to the unlocked position allowing the locker door to open. Initially we were under the assumption that the lock had a third pin for a digital control signal to assign the state of the lock. However, upon arrival we learned that it could only be controlled when powered or unpowered. To fix this, we added a simple n-type MOSFET to each locker in the circuit configuration shown in Figure #3 shown below. The control signal is sent from the MCU within each locker to a daughter board which contains the NMOS and resistor.

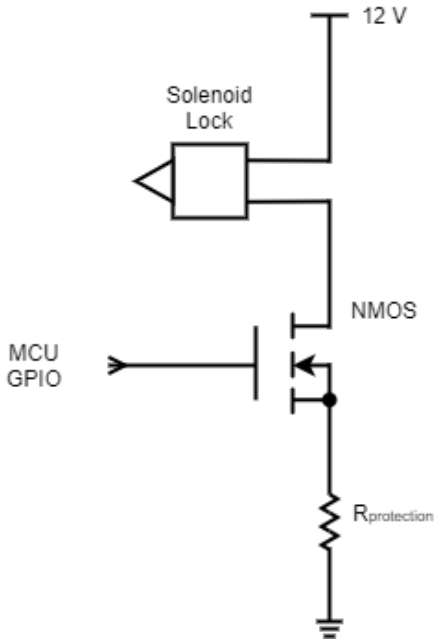

Figure 3.

2.3.2. Communication Bus

In Setup Mode, ESP32S2 in each module will send its own MAC address on the Communication Bus after a button press. They also parse the message on the bus, if there is a message that contains a matching address, the locker module will recognize it and execute instructions.

2.4. Website/Database Module

2.4.1. Website Front End

The front end of the website consists of two pages, a login/registration page and a profile page. The login/registration page allows new users to create an account given a name, email, and password in the “Registration” section at the top of the screen. Once a new user is registered, they can login to their account in the section labeled “Login” at the bottom of the screen. The layout of the login/registration page can be seen below in Figure #4. Whenever an incorrect or unregistered email address or password is entered into the Login section, an error message is displayed notifying the user that the account information entered is incorrect. Once a user is logged in, they can click the link at the bottom of the page to navigate to their profile page. The profile page is accessed through a protected link so navigation to this page is impossible unless a user is first authenticated.

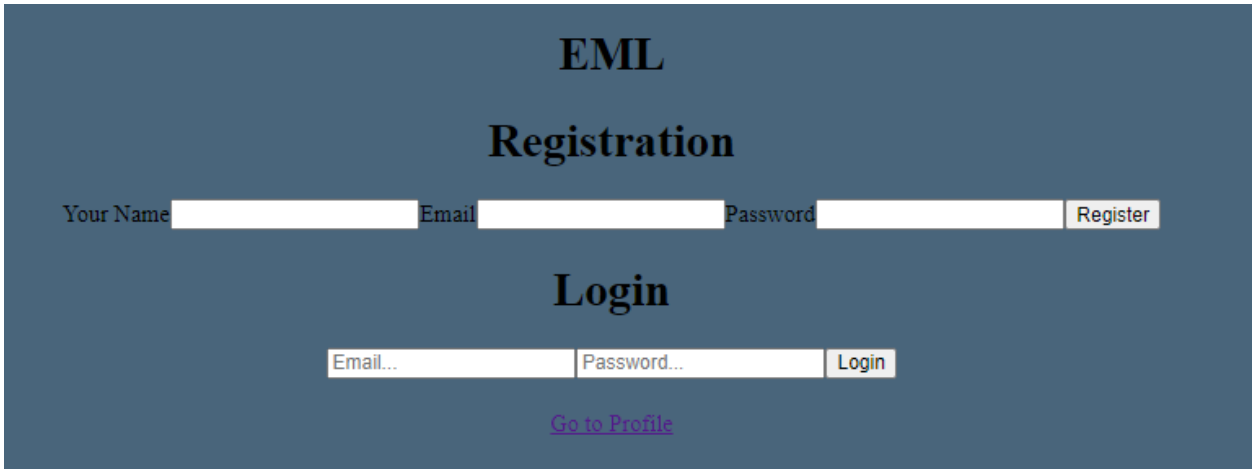

Figure 4. Login/Registration page of website

Once the profile page has been accessed a user has access to all of the information that they need to use the lockers. The top section of the screen displays current pickup and deposit codes for the locker. When a user first registers, six digit randomized codes are automatically assigned to them. They also have the option to get new pickup and deposit codes whenever they would like by pressing the “Get New Codes” button. All codes are guaranteed to be unique for security purposes. The next section of the profile page is the package status section. Here a user can see if they currently have any packages ready for pickup in the lockers. If a user is given administration privileges, they will also be able to

see an “Admin” section of the profile page. Here, an administrator can check their current admin code as well as generate a new randomized code. The layout of the profile page is show below in Figure #5.

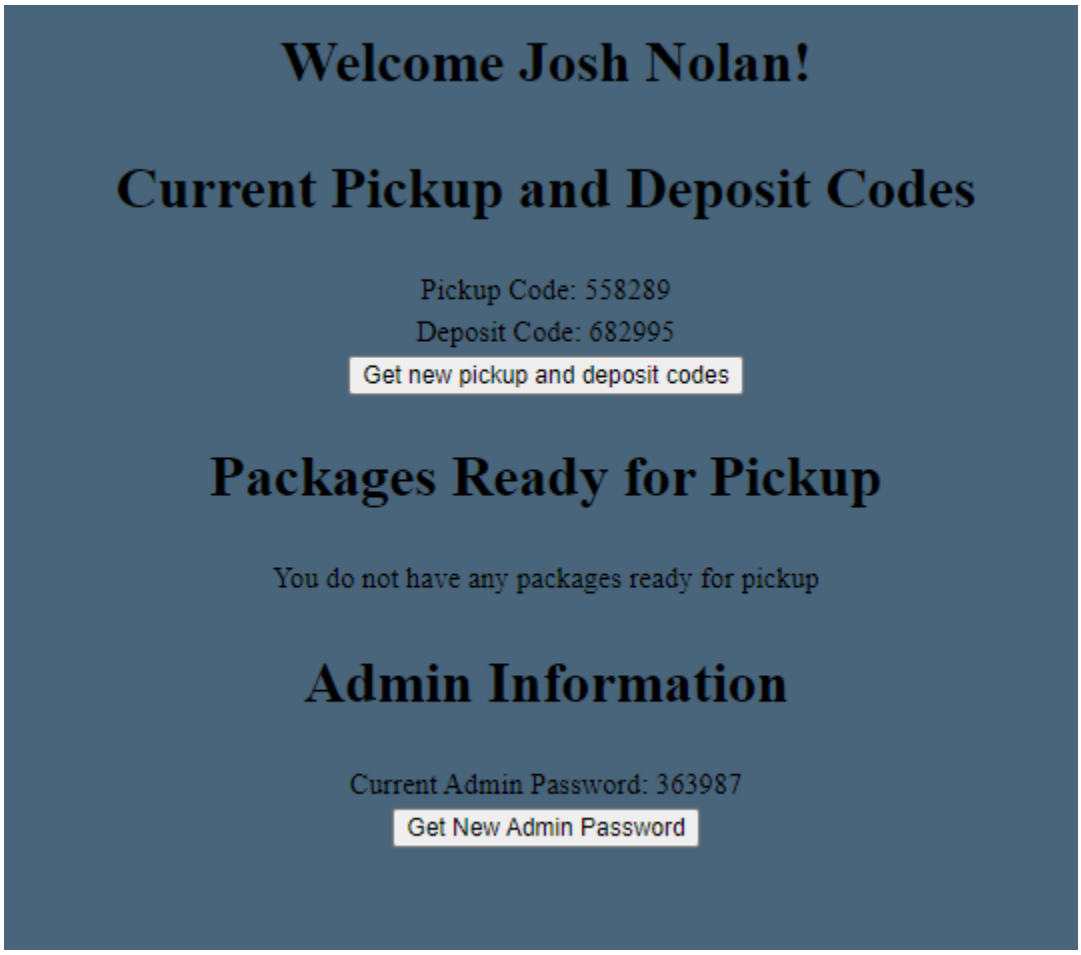

Figure 5. Profile page of website

2.4.2.Website Back End

The back end of the website consists of a simple server using the Express Web Application Framework and various queries to/from our MySQL database. These queries communicate with the tables in the database to provide the front end with information that it requires at various stages for the UI. Communication between the front and back end is facilitated through the http request handler library Axios. The relationship between the front end, back end, and mysql server is shown in Figure #6 below.

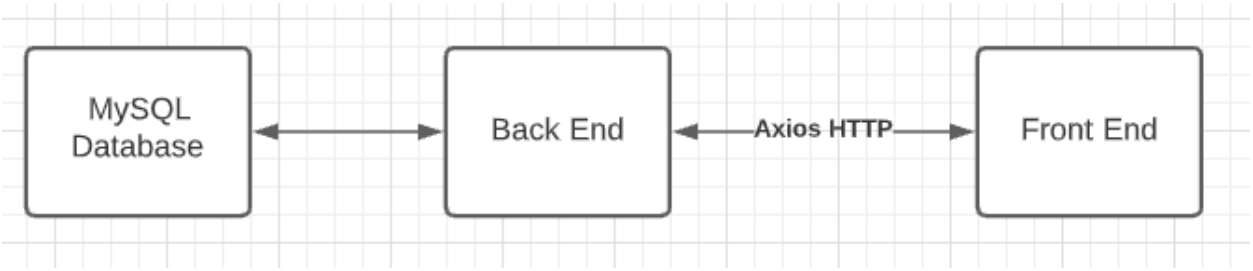

Figure 6. Website data flow

2.4.3. MySQL Database

The MySQL database used for this project contains two data tables. The user data table and the backup data table. MySQL is an ideal way to keep track of all this information because databases can be stored on almost any device with memory and a wifi connection. MySQL can also be easily read from/written to from many different coding languages including python and javascript which are the predominant programming languages used in this project.

2.4.4. User Table

The purpose of the user table is to keep track of the user information of all registered locker users. This table keeps track of each user's name, email, website account password, pickup code, deposit code, admin status, admin password, and when any of this information was last updated. A picture the sample information in this database is provided below in Figure #7.

email	pwd	customer_name	pickup_code	deposit_code	last_updated	admin	admin_password
johnhd4@illinois.edu	12345	Jack Davis	557431	794796	2021-04-26 22:20:26	0	0
jtnolan2@illinois.edu	illini	Josh Nolan	558289	682995	2021-05-03 20:11:10	1	363987
ece445lockeremail1@gmail.com	JoshN445	Mark Cuban	397075	702016	2021-04-26 22:22:33	0	0
ece445lockeremail2@gmail.com	JoshN445	Selena Gomez	947330	417627	2021-04-26 22:22:51	0	0
ece445lockeremail3@gmail.com	JoshN445	Ron Howard	425938	265874	2021-04-26 22:23:38	0	0
ece445lockeremail5@gmail.com	JoshN445	Kevin Hart	444172	736284	2021-04-26 22:29:02	0	0

Figure 7. MySQL user data table example data

2.4.5. Backup Table

The backup table keeps track of the current status of the physical locker modules. For each locker, the table keeps track of the unique Raspberry Pi ID that the locker system is associated with, the address of the microcontroller of that locker, the number of that locker in the system, and the email address of the person who currently has a package in that locker. This table is very important to the design of our locker system because it allows the locker system to refresh to its previous state in the event that it loses power or the

software on the locker restarts for any other reason. A picture of example data from this table is provided below in Figure #8.

ID	Locker_Number	Address	occupied_email
100000000ca0824e	1	124:223:161:5:121:170	NULL
100000000ca0824e	2	124:223:161:15:137:118	NULL

Figure 8. MySQL backup data table example data

2.4.6.File Storage

The images captured by the camera of the control module are uploaded to the cloud server through Secure Copy Protocol. The server we used for this project has a capacity of 500 GB and can store approximately 142,000 pictures.

3.Design Verification

3.1.Control Module

3.1.1.Power

The power module has two parts - converting 12V to 3.3V and converting 12V to 5V. There were only two simple voltage checks on these power modules. The stable operation of the locker system proved the reliability of the power module.

3.1.2.Security

The check on security module is done by running script *security.py* on Raspberry Pi and checking local storage for captured images. When a person approached the locker system, the security module could capture at least two images. The test shown in figure 9 below shows the duration that the motion sensor will hold the signal after initially detecting motion. After the sensor returns to a low state there is a delay between when it can detect motion again. These parameters are controllable in hardware using two variable resistors. We have set them in the lowest time state that the unit allows to give the security system the most real-time data available. The time delays that the sensor output fall comfortably within the specifications outlined in the verification table for adequate photo capturing.

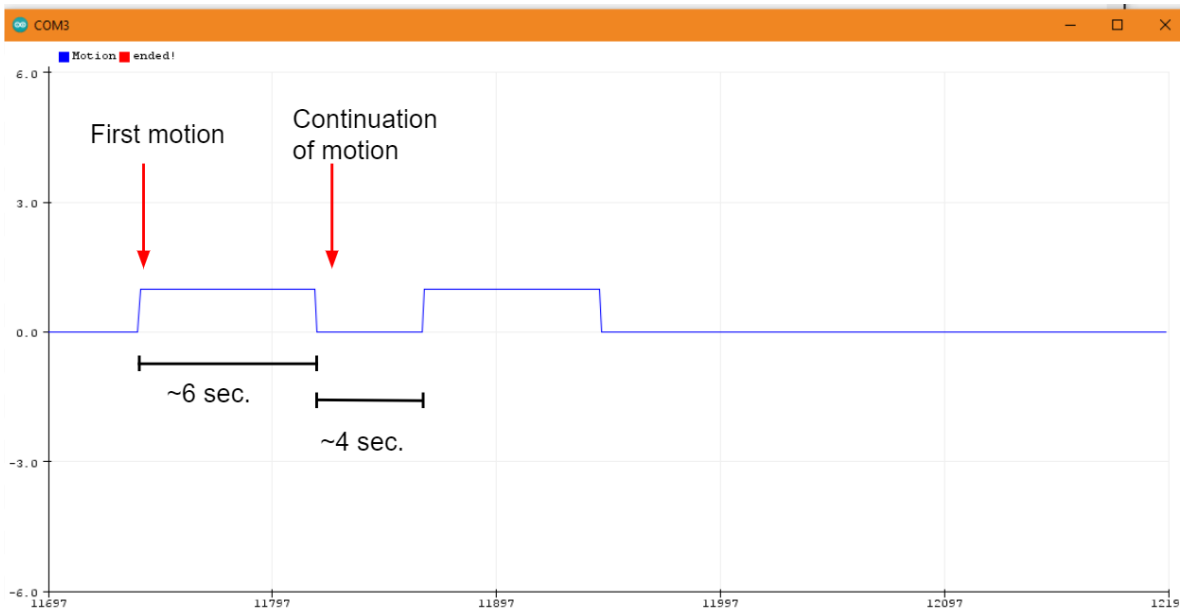

Figure 9.

3.1.3.GUI

To verify the correct operation of the GUI, we first retrieved a dropoff code from an existing user. Next we entered that code into the dropoff section of the GUI when the lockers were all vacant. The desired behavior was for one of the lockers to open and for the MySQL backup database to be updated with the email of the user who had a package dropped off.

When a valid code was entered into the dropoff section of the GUI, locker number one immediately popped open. The MySQL backup database was also updated with the proper email for locker number one. The before and after pictures of this database are shown below in figure 10 and 11.

ID	Locker_Number	Address	occupied_email
100000000ca0824e	1	124:223:161:5:121:170	NULL
100000000ca0824e	2	124:223:161:15:137:118	NULL

Figure 10. Initial State of MySQL backup data table

ID	Locker_Number	Address	occupied_email
100000000ca0824e	1	124:223:161:5:121:170	jtnolan2@illinois.edu
100000000ca0824e	2	124:223:161:15:137:118	NULL

Figure 11. MySQL backup data table after dropoff has been performed

3.1.4.Keypad

Throughout testing the keypad we verified that it worked independently of the system without issue. Shown in Figure #12 below is shown a simple test of the keypresses. In the project's mock demo we were able to demonstrate the functionality of the keypad without issue. However problems arose once that the full system was demonstrated. We did our best to isolate the contacts of the keypad from the metal front plate that it was mounted with electrical tape to to isolate undesired short circuits. However this did not seem to fix the issue. We suspect that the immense computing power that the Raspberry Pi was subjected to from the GUI or other threads may have caused enough delay in the sampling for the Pi to incorrectly read the pins.

Figure 12. Simple test of the keypad using an Arduino Uno and the Keypad library written by Mark Stanley, Alexander Brevig [7]

3.1.5.Communication Bus

The communication is verified by checking the message transmitted on the RS485 bus. Raspberry Pi sends the MAC address of the target locker's ESP32S2 to unlock a locker module. If the unlock message can be detected and decoded correctly. It means that the sending device handled RE/DE of the RS485 transceiver successfully. In Figure 13, we can see that the RS485 signal is correct. After decoding, we have verified that message is the correct MAC address of the ESP32S2 of the target locker.

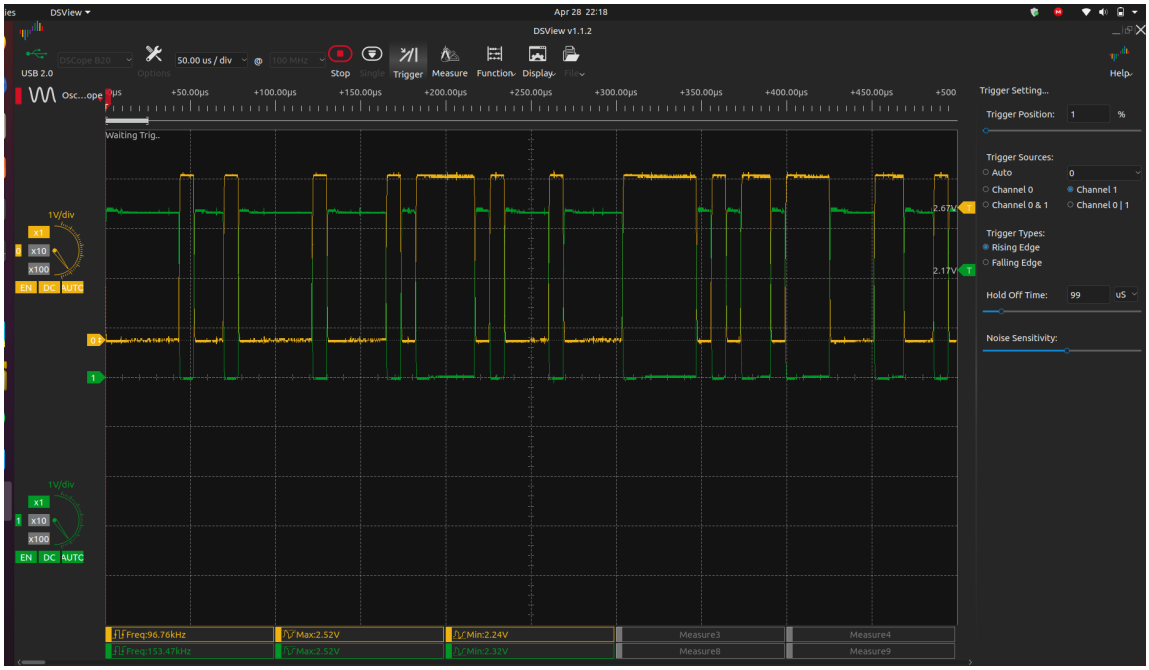

Figure 13. Oscilloscope Reading on RS485 Signal

3.2.Locker Module

3.2.1.Lock Control

There are two parts to check in lock control. The digital output of the ESP32S2 is verified by the multimeter. When ESP32S2 wanted to unlock the locker module, it output high level (3.3V) for a desired period (3 seconds) and fell back to low level. The functionality was checked by using the connection in Figure # and applying 3.3V to the gate to check whether the lock is actuated. The results of both tests were successful.

3.2.2.Communication Bus

The verification of the communication bus on ESP32S2 is done by the check in 3.1.5. Since the Raspberry Pi can only send the correct message after it receives the correct MAC addresses from all ESP32S2 chips in Setup Mode, the correct message in 3.1.5 proves that a correct ESP32S2 MAC address was sent over RS485 earlier in Setup Mode.

3.3.Website/Database Module

3.3.1.Website

In order to verify that the website interacted correctly with the MySQL database, we first created a new user to see if that user showed up in the MySQL user data table. Next we

logged into that user's account on the website and changed that user's pickup and deposit codes. We then verified that the MySQL user data table changed to show these new codes. When performed these actions on the website, the MySQL database updated in the correct fashion. Images of each of these steps are shown below in Figure numbers 14, 15, 16, and 17.

Figure 14. Registering a new user on the website and signing in to their profile

email	pwd	customer_name	pickup_code	deposit_code	last_updated	admin	admin_password
mthomas44@gmail.com	pwd1234	Matt Thomas	199388	161887	2021-05-04 00:16:44	0	0

Figure 15. New user's data in the MySQL user data table

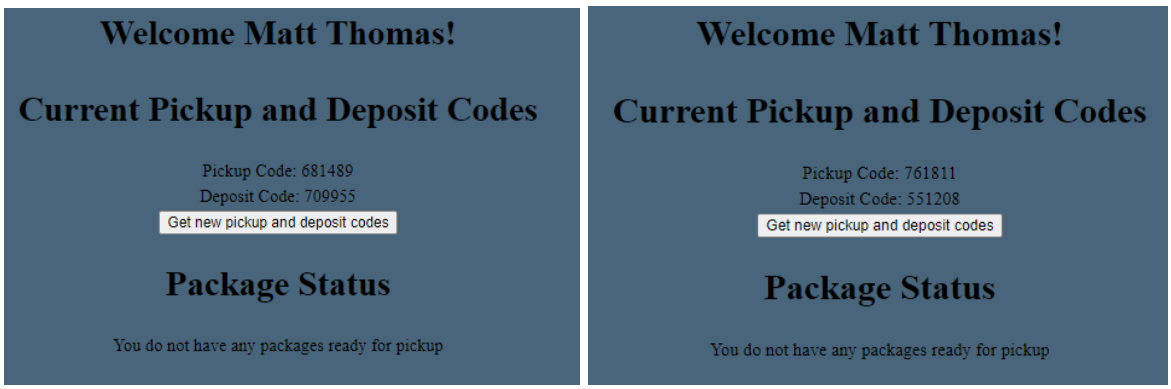

Figure 16. Initial pickup and deposit codes for new user followed by pickup and deposit codes for user after they have been changed.

email	pwd	customer_name	pickup_code	deposit_code	last_updated	admin	admin_password
mthomas44@gmail.com	pwd1234	Matt Thomas	761811	551208	2021-05-04 00:20:18	0	0

*Figure 17. Final state of the new user's pickup and deposit codes in the MySQL user data table. *Note that pickup and dropoff codes have changed to match those shown on the website.*

3.3.2.MySQL Database

The high level requirements for the MySQL database have been met by the tests in the website and GUI sections of verification. All data in both the user and backup data table can be read and written from both the Raspberry Pi and the back end of the Website.

3.3.3.File Storage

The images can be checked over Secure Shell Protocol (Figure 18). The images can also be transferred to any laptop using Secure Copy Protocol.

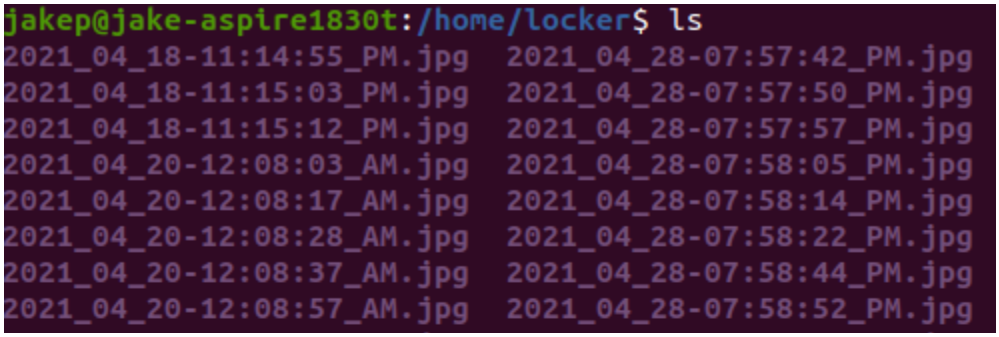

Figure 18. Photos stored on the database from a range of 10 days

4.Costs

4.1.Parts

Parts Costs					
Part	**Manufacturer**	**Retail Cost ($)**	**Bulk Purchase Cost ($)**	**Quantity**	**Actual Total Cost ($)**
DC Barrel Connector	CUI Devices	0.72	0.322	1	0.72
5V Regulator	Texas Instrument	2.09	0.936	1	2.09
ESP32-S2-WROVER-I	Espressif Systems	2.20	2.200	2	4.40
RS485 Transceiver	Maxim Integrated	5.20	5.040	3	15.60
3.3V Regulator	Diodes Incorporated	0.38	0.087	2	0.76
Tactile Switch	TE Connectivity	0.10	0.057	8	0.80
10k ohm Resistor	Bourns	0.14	0.025	4	0.56
1uF Capacitor	AVX	0.40	0.072	5	2.00
10uF Capacitor	TDK	0.73	0.206	4	2.92
22uF Capacitor	TDK	0.61	0.147	2	1.22
USB Type A Connector	Molex	1.37	0.607	1	1.37
Pin Header	Amphenol FCI	0.98	0.422	10	9.80
Terminal Block	TE Connectivity	1.62	0.553	12	13.63
USB Type B Connector	CUI Devices	0.45	0.283	2	0.90
NMOS	Alpha & Omega Semiconductor Inc.	0.50	0.105	2	1.00
Raspberry Pi 4 B 4GB	Raspberry Pi Foundation	54.99	54.990	1	0
PCB	JLCPCB	0.64	0.640	3	1.92
Total					59.69

Table 1. Parts Costs

4.2.Labor

We estimate the salary of an engineer to be $50/hr to work on this project for 10 hrs per week for 12 weeks. For 3 engineers and an overhead factor of 2.5, we estimate the development costs to be

$$3 \bullet \frac{\$50}{hr} \bullet \frac{10hr}{wk} \bullet 12wk \bullet 2.5 = \$45,000$$

Equation 1. Labor Costs

The quoted cost of the construction of the 3 boxes is $150 in the machine shop. The majority of our costs lie in the control module locker and as a commercial product each locker owner would only have to purchase one of these. Each locker module would be much cheaper to purchase than the control module for the entire system.

5.Conclusion

5.1.Accomplishments

We managed to create a completely modular system of lockers along with a graphical user interface and website for user interaction. While the demonstration of these lockers only included two locker units, we have shown that the technology that these lockers are built upon is able to support up to 127 lockers. Furthermore, we added security and administration features to the locker system that were not present in our original design to make the lockers more user-friendly and safe.

5.2.Uncertainties

Because we use an analog bus, there is always a possibility that the data on the bus becomes corrupted. We decided to use even parity in our Communication Bus along with error checking in our microcontroller code to safeguard against data corruption. Whenever the microcontroller detects an error on the bus, the message on the bus is ignored.

5.3.Ethical Considerations

This system of lockers has the potential to be exploited by criminals to exchange illicit goods and cash. In order to uphold section 1 part 4 of the IEEE Code of Ethics, the locker will take pictures of any person that walks within five feet of the control unit. These pictures can be turned over to the proper authorities in the event of any suspicious or illicit activity.

These lockers are also directly responsible for the protection of the property of others. In compliance with section 1 part 1 of the IEEE Code of Ethics, we will strive to ensure that

both the software and hardware components of our lockers protect the privacy of their users absolutely. One step that we can take to further ensure the privacy of our users is data encryption. While data encryption was not a part of this project, implementing the encryption of data both on the website and on the locker bus in the future would go a long way in ensuring the protection of our users' property and data.

5.4. Future Work

Future work to deploy this system in a commercial setting includes email verification, minimum password strength requirements during registration, and custom fabrication of larger connectors that would allow lockers to snap together easily in multiple configurations. As of now any email and password combination is accepted on the website to register a new user. While this does not change the operation of the lockers in any way, the security of users could potentially be compromised if their passwords are not strong enough. Furthermore, if the email that a user inputs during registration is not valid, the locker system would not be able to send them emails to notify the user of the status of their packages.

Additionally, the communication protocol could be switched from UART on RS485 to I2C differential for a more mature and robust multi-master multi-slave bidirectional communication.

Lastly, in the current configuration, lockers need to be wired to one another through the back of each locker. While this design can support the lockers in multiple configurations, the exposed wires are a concern in any kind of commercial design. There are currently no connectors on the market that would serve our purposes properly, but custom fabrication of larger spring-loaded connectors that could be easily snapped together and held in place is possible if this product was distributed on a considerable scale.

6.References

[1] PracticalEcommerce, ‘Porch Piracy Is Growing’, 2021. [Online]. Available: https://www.practicalecommerce.com/porch-piracy-is-growing. [Accessed: 2-18-2021]

[2] Digital Commerce 360, US ecommerce grows 44% in 2020, 2021. [Online]. Available: https://www.digitalcommerce360.com/article/us-ecommerce-sales/. [Accessed: 2-18-2021]

[3] The Daily Pensnsylvanian, ‘More students have packages stolen as thieves resort to following delivery trucks’, 2021. [Online]. Available: https://www.thedp.com/article/2017/04/off-campus-package-theft. [Accessed: 2-18-2021]

[4] WUSA9, ‘VERIFY: Who is responsible if a thief steals your packages?’, 2019. [Online]. Available: https://www.wusa9.com/article/news/verify/who-is-responsible-if-your-package-is-stolen/65-52b8f478-8aee-4cf4-951b-848f263602f4. [Accessed: 2-18-2021]

[5] BusinessWire, ‘Buckle Temporarily Adds Food and Delivery Insurance Coverage to Member Policies at No Additional Cost’, 2020. [Online]. Available: https://www.businesswire.com/news/home/20200506005184/en/Buckle-Temporarily-Adds-Food-and-Delivery-Insurance-Coverage-to-Member-Policies-at-No-Additional-Cost. [Accessed: 2-18-2021]

[6] Ieee.org, "IEEE IEEE Code of Ethics", 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 2-29-2021].

[7] arduino.cc, “Keypad Library for Arduino”, 2018. [Online]. Available: https://playground.arduino.cc/Code/Keypad/. [Accessed 4-5-2021]

[8] github.com, “Modularized Smart Locker”, 2021. [Online]. Available: https://github.com/jakepu/Modularized-Locker.

7.Appendix B: RV Table

1.0 Power Supply

1.1.1.Voltage Regulation

Requirement	Verification
1. Raspberry Pi 4 can run steadily for at least 15 minutes with all of its peripherals on with a 12V to 5V converter. 2. ESP32 can run without any problem running off of the pcb-mounted 12 V to 3.3 V converter.	1. Raspberry Pi can steadily run the locker program for at least 15 minutes. 2. ESP32 runs all testing without failing.

Table 2.

1.2.Control Module

Requirement	Verification
1. Can retrieve the user data table from the server in under 5 minutes. 2. Can send control messages over the data bus to notify the according locker module to unlock its electric lock using RS485 signal standard. 3. Can match the user-input code with valid pickup and dropoff codes.	1. A. Dropoff package using provided dropoff code B. Create new pickup code and verify new code opens correct locker 2. A slave device on the databus can receive a correct message in RS485 signal standard from RPI4. 3. Can recognize input from the keypad and display it on screen. 4. Can react to the touch action

4. Can receive input from the touchscreen and display instructions to users.	from the touchscreen.

Table 3.

1.2.1.ROM Storage

Requirement	Verification
1. Can store at least twenty 720p images.	1. Display an image stored in the ROM.

Table 4.

1.2.2.Camera

Requirement	Verification
1. Must be able to take 720p photos whenever a locker is opened and transfer photo data to the microcontroller to be stored on a MicroSD card.	1. A. Take a picture when a person walks close. Check the MicroSD card for a photo. B. Take a picture when a locker module is opened. Check the MicroSD card for a photo.

Table 5.

1.2.3.LCD Display

Requirement	Verification
1. Will react to touch input from the user. 2. Able to display images with half a second refresh rate or lower.	1. Respond to touch and display different content. 2. A. Code a program that changes the LCD screen when it is touched. B. Time how long it takes for the screen to change.

Table 6.

1.2.4.Keypad

Requirement	Verification
1. Signals from all buttons can be transmitted without error.	1. Press the 12 keys on the pad to ensure and check whether the connector has the correct signal output.

Table 7.

1.2.5.PIR Sensor

Requirement	Verification
1. The sensor detects when someone walks within five feet of the front of the control module.	1. Walk in front of the sensor and check to see if the signal output is high.

Table 8.

1.3.Locker modules

1.3.1.Microcontroller

Requirement	Verification
1. Able to send unlock signals to its associated locker. 2. Able to communicate with the Control Unit over the databus in RS485 signal standard.	1. A. Code the slave microcontroller to send an unlock signal to its associated lock B. Verify that the lock opens 2. A. Send an unlock message over the data bus to the slave microcontroller B. Verify that the associated lock opens

Table 9.

1.4.Software

1.4.1.Cloud Storage/Database

Requirement	Verification
1. Store locker images for up to two weeks 2. Store user data table	1. A. Upload an image to the cloud through the API the RPI4 will be using. B. Check the storage/database for the new image. 2. A. Change a user pickup code on the locker website B. Display the user data table from cloud storage/database and verify that the pickup code has changed.

Table 10.

1.4.2.Web Page Frontend

Requirement	Verification
1. Let users log in with their deposit codes. 2. Let users change their deposit and pickup code once they log in. 3. Let the user check whether or not there are any packages stored in the locker for them.	1. Use the password to log in. 2. A. Change its deposit code and pickup code B. Log back in and check the newest deposit code and pickup code. C. Open the locker door using a deposit code D. Verify that there is notification that says there is a package ready for pickup.

Table 11.

1.4.3.User Interface

Requirement	Verification
1. Gives users and delivery services clear instructions for deposit and pickup of packages. 2. Secure state machine that does not have undefined behavior.	1. A person without prior experience with this locker system can finish deposit and pickof of a package following the instructions on the display. 2. Try all possible user behavior and the system should function normally.

Table 12.

1.4.4.Data Bus and Device Addressing

Requirement	Verification
1. Locker modules have a distinct address. 2. Locker modules can recognize a message directed to its address.	1. Ensure each locker module has a unique address. 2. Verify that the locker opens when it is addressed.

Table 13.