SMART BIKE RACK SYSTEM

Senior Design II
Summer 2018
Group 13

Amanda Chanthalangsy (EE)
Vanessa Garcia De Quevedo (CpE)
Joel Gonzalez (CpE)
Trung Luu (CpE)

Table of Contents

Table of Figures

1 Executive Summary

The material contained in this paper documents the motivation, specifications, research, design, implementation and testing of EzRack. EzRack is a smart bike rack storage system. EzRack was inspired by existing bike rack solutions such as Juice Bike and BCycle. This project is designed to provide security, storage space, and convenience to students that operate or store their bike on the University of Central Florida's campus.

With technology continuously improving, this project aims to integrate those innovative technologies to improve the experience of the user. This smart bike rack station is accompanied by an Android application that will allow students to view the availability of slots on the rack and also to reserve the slot. Students understand that time can be very valuable and should not be wasted trying to secure a bicycle. In addition, students won't need to bring their own bike lock. EzRack has a built-in lock mechanism to provide security for the bike and a peace-of-mind for the user.

EzRack will make an impact by being environmentally friendly with its off-grid solar power system. The components connected to the bike rack station is completely powered by solar energy. Not only does this reduce our environmental footprint, it allows for easy installation in various locations. EzRack hopes to motivate students to bike around campus rather than to drive a motor vehicle. At the University of Central Florida, finding a parking spot may be very difficult. There are no parking spot reservations or indicators currently available. EzRack will provide students with a smarter alternative. Although this project is limited due to certain constraints, future implementations of this design may explore bike rental opportunities or personal locker attachments. The bike rack station can also be expanded to hold a larger number of bike slots. Further innovations are plentiful.

The purpose of this project is to apply our knowledge and experience from electrical/computer engineering to enhance fellow students' daily commute and help alleviate the stress of searching for parking on campus. This project contains motivation from the knowledge obtained at the University of Central Florida. All aspects of this project are critical to the development of our engineering experience. From the complex power system to the embedded hardware and the mobile application, each component is thoroughly researched and designed.

2 Project Description

The following section will cover the motivation that drives this project as well as the inspiration behind it. It will also set forth the goals and objectives that are accomplished. The specification and requirements will also be defined to help guide the development process. Project constraints will also be identified to set realistic expectations and an understanding of the true potential that lies in the design.

2.1 Project Motivation

Finding parking at the University of Central Florida may be a difficult task. Students may look towards alternatives that may eliminate the task completely. Having a bike-friendly campus may encourage those students that live near the campus to give up their automobiles in favor of biking to school. This would potentially help alleviate the University of Central Florida's parking problem. Not only would it improve the parking situation on campus, it would also be a better alternative for the environment. With more students biking to campus, it would reduce the traffic as well as parking congestion. Also, less traffic means less emissions from vehicles that are circulating the campus trying to find an open parking spot. There is no need to waste fuel driving up and down parking garages all around campus during rush hours. Students could simply bike to campus with the comfort of knowing they'll be able to park it where they need to.

The University of Central Florida has bicycle lanes going around the campus grounds as well as low speed limits that help ensure bicyclist safety. In addition to the speed limits, the school has four bike-share locations that provide students with bicycles at no charge for a 24-hour period [1]. There is also a bike shop that provides no cost repairs to students provided that they supply the parts. There are plenty of useful resources that are available to students at low to no cost at all in regards to having, maintaining, and operating a bicycle around campus. What the University of Central Florida lacks, however, are convenient bike racks. Not only are bike racks located inconveniently, they are often overcrowded. At times, some bikes may be locked to a rack for an extended duration without any necessary purpose other than storage. This takes up a slot that could be used by a student that is currently in need of simply locking their bike for a short duration of time. This may cause frustration and force the student to look for a nearby open bike rack. Not only will locating a different bike rack take time and effort, it could ultimately cause that student to be late or unknowingly secure their bike somewhere that is not allowed. Our project aims to solve these recurring issues.

2.2 Goals & Objective

The goal of this project is to develop a bike rack that improves the safety and convenience of students who use bikes as transportation on campus. Our bike rack offers students the ability to see the availability of the slot on the bike rack beforehand and reserve the slot. This will ensure the student that they are able to park their bike when they need to and where they need to. Students will also have the ability to see the nearest bike rack and check if the current rack is fully occupied. The bike rack offers a smart lock for students using a mobile application or a keypad for entering a physical pin. There is no need to provide a personal lock or have to carry around a key.

The bike rack station is completely solar powered off grid. This will allow each station to be located almost anywhere the sun can reach. Each slot in the bike rack station has an individual lock that secures either the frame or the wheel of the bike to that respective station. The individual slots are able to detect whether or not it is in use by the cycle that the lock has been through.

Main Goals:

- Powered off grid to promote renewable energy
- Station kiosk with solar panel on top
- Battery adjacent to kiosk station
- Mobile application for ease of use
- Online reservations
- Locating available racks and open slots
- User defined passcodes
- Intuitive locking system for hassle-free storage
- Physical keypad for manual pin entry option
- Sensing slot availability from lock activation
- Secure electronic lock

The initial inspiration for EzRack came from Smack, a smart bike rack developed by UT electrical and computer engineering seniors. Smack allows users to reserve a space for their bike and set a pin to unlock a specific slot in the rack. Inspiration for the use of solar panels and finding rack locations came from Bikeep.

2.3 Project Constraints

For the duration of completing our project, the three major constraints that were faced were physical, economical and time related. With any project, there are environmental, social, political, ethical, health, safety, manufacturability, and sustainability constraints. We will examine all of the realistic constraints and how they may have effect on our overall design.

2.3.1 Manufacturability and Sustainability Constraints

Certain components of our design may face manufacturability constraints and restrictions. Due to the physical aspects of the bike rack, and the fact that the team is comprised solely of electrical and computer engineering students, the design and construction of the rack is limited to easy to assemble and premade components. The locking mechanism may have to be purchased rather than designed and developed if needed. This may also be the case for the actual frame of the bike rack itself. Many components of the bike rack station are large and industrious and cannot simply be manufactured for this project. As for sustainability constraints, our bike rack station must be designed with consideration of how it will withstand the outside elements. The user kiosk must be waterproof or water resistant. The solar panel must be able to behave in extreme temperatures. The battery must also be able to last in extreme conditions. The housing of the bike rack itself must be fully water resistant so that moisture, dirt, and debris is not able to interfere with the electrical components such as the locking mechanism and the user kiosk.

2.3.2 Economic and Time Constraints

EzRack is entirely self-funded and the budget is determined by the amount that can be contributed by team members. When selecting components, how well they fit into the budget is one of the primary considerations, especially since a large portion of the budget had to be reserved for the solar panel. The strictly limited budget must also factor in lost or damaged parts.

The largest constraint however, may be time. All team-mates either have a full time job and/or a full course load potentially taking time away from the project. Additionally, the second half of the class will occur during the summer which is a shorter semester. Summer semester is cut short by four weeks which will leave us with less time to complete the project. It may also hinder our ability to add more advanced functionality to the smart bike rack.

2.3.3 Ethical, Health, and Safety Constraints

Certain constraints regarding ethics, health, and safety may affect our project. The safety of our bike rack station design must be considered. The locking mechanism should not be a risk to the user that is operating it. All electrical components must be secured and enclosed to ensure safe operation as well as properly grounded. Bike riders should be advised to always wear a helmet and be cautious while biking around campus. This advice may be communicated through the app or located at the kiosk to promote safety.

2.4 Specifications and Requirements

Our project has two main parts: the physical bike rack and the mobile application accompanying it. The physical bike rack houses the power system, embedded system, and locking mechanism. The mobile application is used to communicate wirelessly to the embedded system and hardware on the physical bike rack. These are the requirement specifications for the bike rack station.

Since our project is just a prototype for an envisioned, larger scale bike rack station, we will build a minimum design that incorporates components that may easily be expanded. Figure 1 illustrates the overall design concept of our bike rack station.

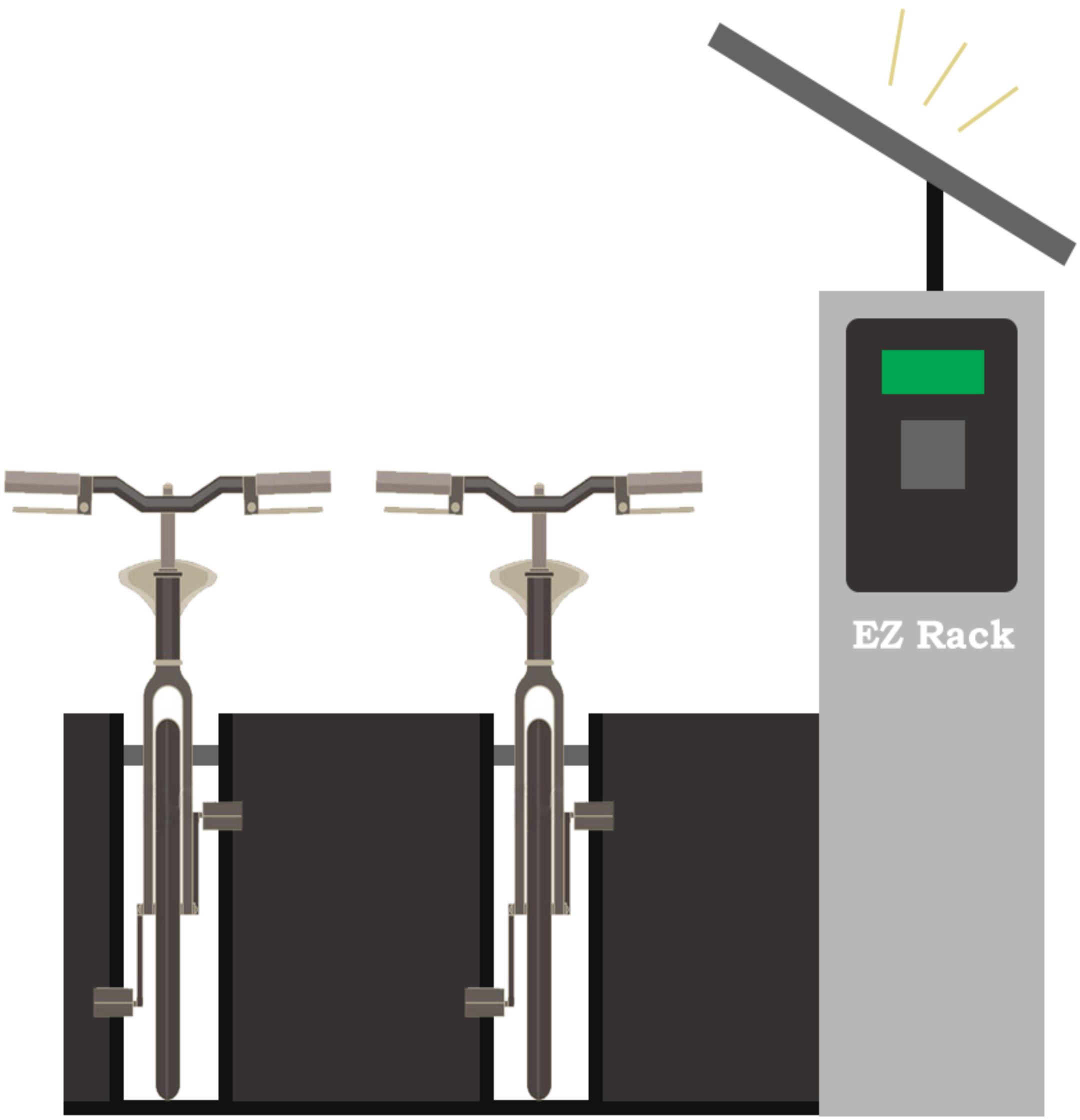

Figure 1 EzRack Station Design

2.4.1 Physical Specifications

- The bike rack contains two slots capable of holding two bicycles.
- Each slot has a minimum of 3” clearance
- The distance between each slot is a minimum of 8”
- The lock bolt path is a minimum of 3”
- The height of the lock is no taller than 28”
- The diameter of the lock bolt does not exceed 2”
- Each slot of the rack has a 12V linear actuator
- The lock completes operation in 5 seconds or less
- The kiosk contains the LCD and keypad
- The kiosk is at least 36” tall
- The solar panel is mounted at 45 degrees facing south
- The solar panel does not exceed 15 lbs
- The solar panel is mounted at least 50” from the ground
- A 12 V battery is located under the kiosk housing
- The power system delivers 12V DC and 5V DC
- The solar panel provides 50 W at 12V
- The battery lasts at least 2 days without a recharge
- The solar charge controller charges the battery at 12V
- The solar charge controller has 2 outputs (12V, 5V)
- The solar charge controller contains a buck converter to convert 12V output to 5V output
- A motor driver enables 2 lock mechanisms with 5V input
- Lock mechanisms are powered by 12V
- The operating voltage of the microcontroller is 5V
- The operating voltage of the wireless module is 3.3V
- All electrical components are properly grounded
- The power system has appropriate heat sinks

2.4.2 Software Specifications

- A mobile application is developed for:
    - Finding bike rack station locations
    - Checking each station for slot availability
    - Making rack slot reservations
    - User define unique passcode
    - Locking/Unlocking the slots on the rack
- Open slots are updated in real time
- Unlocking will commence within seconds of receiving unlock code

2.4.3 Engineering Marketing Requirements

In this section, all of the requirements from the system's features in regards to functionality and quality are highlighted below. In the table 1 below, engineering requirements are correlated with marketing requirements. Justification is provided.

Engineering & Marketing Requirements		
Requirements		Justification
Marketing	Engineering	
3	1. Must have physical lock to secure bicycles.	Theft prevention (Functionality, Safety)
2, 4	2. Must have a physical keypad for entering pins manually.	In the event where mobile application is not accessible. (Functionality)
1, 2, 4	3. Must have solar panel to provide enough power to the system.	Rack can sufficiently provide its own power. (Energy)
2	4. Mobile application must have the ability to check for slot availability.	This feature offers convenience (Software Functionality)
2	5. Rack mobile application must be able to offer slot reservation for *x* amount of time.	This feature offers convenience (Software Functionality)
2, 4	6. Rack mobile application must allow user to define five digits pass codes.	User defined passcode is utilize to lock and unlock EzRack lock mechanism. (Software Functionality)
2	7. Rack mobile application must be able to lock and unlock rack slot.	This feature offers convenience (Software Functionality)
Marketing Requirements 1. The rack is cost effective. 2. The rack must offer convenience to bike rider. 3. The rack must offer security to bicycle. 4. The rack must offer fail safe.		

Table 1 Engineering & Marketing Requirements

2.5 Project Milestones

There are many aspects in the project that require an abundant amount of time to achieve. It is vital that group members work efficiently towards completing this project. This project has been completed within two semesters. The project development has spanned from spring semester and finished in the summer.

At the end of the summer semester, a fully functional prototype of our project is complete. The project has gone through extensive hardware and software integration testing. The prototype is ready for presentation and each group member has complete knowledge of the design. The table 2 shown below explicitly lists the deadlines that have been met in order to ensure that our project is developed in an efficient time.

Projected Milestones		
Goal	**Projected Start Date**	**Date of Completion**
Design Concept (Research Paper)	January 2018	April 23, 2018
Permission Requests	January 2018	April 23, 2018
Obtain components/materials	January 2018	May 2018
Software/app development	April 2018	July 2, 2018
Hardware assembly	May 2018	July 9, 2018
Testing/debugging software	May 2018	July 16, 2018
Re-obtain components/materials as necessary	May 2018	July 2018
Assemble housing	June 2018	July 2018
Final Abstract Paper	July 2018	July 2018
Develop Website	July 2018	July 2018
Finalize Project	July 2018	July 23, 2018

Table 2 Project Milestone Dates

3 Research Related to Project

The following section discusses what bicycle-sharing is about and the types of systems that are currently being used. It will also examine products that are available today and existing technologies that are currently being used.

3.1 Existing Products

The origin of bike-share began in Europe in 1965. Bicycle-sharing system, public bicycle system, or bike-share scheme is a service that offer bicycles as a mode of transportation that is shared to more than one individual. The use of bicycle can be for a small fee or even free to encourage the use of the system.

Bicycle-sharing system can be offered by local community groups or non-profit organization such as IIT Bombay (Indian Institute of Technology Bombay). Government agencies can also implement this system in a public-private partnership like in London or by private companies like most of cities in China. This mode of transportation provide a convenience way to travel a short-distance in busy urban area. This can help with solving several concerning issues in such busy area. Such area can benefit from the issue of pollution, noise, and traffic congestion.

There are many types of bike sharing system developed over the years. First, unregulated system is a type of program that release bicycles into an area for anyone to use. These bicycle remain unlocked to be used by anyone but must remain in within a certain boundary such as a school campus or a city. Deposit system requires a small deposit from a user at a locked terminal to release the bike for use and can only be retrieved when the bike is returned. Membership type offer subscription to the service. Dockless system are bikes consist of advance technology that allow the share-bicycle to be tracked by users via mobile application GPS. Therefore, eliminating the need of stations. Long-term check out also known as bike library system that allow bicycle to be lent out for a small fee, refundable deposit or free of charge.

As opposed to systems mentioned above, EzRack does lean out or provide bicycles for users. EzRack can be thought of as BYOB Smart Rack (Bring Your Own Bicycle). EzRack simply offer users the convenience through security and availability by providing a secure bike rack to individuals who prefer to use their own bicycle. This will help the responsible agency by reducing cost in terms of not providing bicycles, maintenance of bicycles while providing high quality service.

3.1.1 Juice Orlando

Juice is a bike-share system located downtown Orlando that offer flexible bike rental transportation through their bike rental. Juice system offers bike rentals at 30 locations throughout Orlando downtown area. Their system consist smart locks that uses 4-digit PIN code on the keypad for unlocking the bike. Juice also have a mobile app that accompanied their system. The Mobile app let user find the nearest bike hub, reserve a bike, and a map for user's convenience. Most of the technology is on the bike itself. The bike have a device at the rear that has LCD display, pin pad, gps tracking, and also a "U" shape lock mechanism to lock onto bike racks. In Figure 2, Juice Bike Orlando Florida is shown.

Figure 2 Juice Bike Orlando Florida
(Pending permission from Juice Bike)

3.1.2 BCycle

BCycle is another public bicycle sharing company that have forty-seven systems operating cities all over United States. BCycle system allow users to purchase annual memberships or a day or week pass. The intended use is for short trip. Depending on local system in certain city, user can operate the bicycle up to twenty to thirty minutes without additional fee. Members with annual membership are issued membership card that contain RFID chip that is uniquely associated with their account. Members can utilize bike at any B-Station and can be returned to any B-Station. BCycle also has a Mobile application for Android and iOS that allow users to checkout bikes, locate bike & stations, get directions, purchase memberships and track trips. B-Stations varies because there are different models. All stations have bike docks, bikes, and central hub but not all stations has solar panel. In figure 3, it shows Aventura BCycle.

Figure 3 Aventura BCycle
(Pending permission from Aventura BCycle)

3.2 Identification of Need

In order to verify and authenticate users of EzRack, our system must keep a record users information and their data for processing. User is about to register for an account by utilizing our mobile application. A database server is a must for this task. This database server is intermediary contact for the microcontroller located on the rack and the mobile application. There are a few ways of implementing this process, build an actual database server, use an on-device embedded database system, host a local server on a laptop, or utilize cloud solution.

First, creating our own database server is a great learning experience. But with the time and budget constraints this can be a daunting task. Our group would need to purchase hardware and software components to make this happen. In addition, we must rent space for this server to be placed and pay the electric bill. The amount of cost and for a one day showcase for this project does not make sense for this method.

Second, on-device embedded database system is another option in which EzRack can use. The database is on the microcontroller itself. This system can process and manipulate data collected by the system. But this requires all data to be stored on EzRack, overtime this is not suitable for growing amount of users that will take up limited storage on the system.

Third, for demonstration purpose, hosting a database server by using a spare laptop with a connection of a wire to the microcontroller looks attractive. This method with an addition of an android mobile emulator on the same laptop would simplify connectivity even further.

Our fourth option is to utilize cloud base solution. This will eliminate the need for acquiring additional hardware to support as a database. The cloud solution will act as our intermediary contact between the mobile application and the microcontroller on the bike rack. It will act as our database and process its information to be delivered to the rack due to limited computational power and storage on the microcontroller. To use EzRack, users must register for an account by utilizing the mobile app. With cloud solution, user can do this at any time since the server is always online. It will also deliver services to and from the microcontroller to the user via mobile application user interface. Parse cloud-based backend services for mobile application by Amazon known as AWS (Amazon Web Services) is considered. It supports android, iOS, and windows mobile application. This will offer us flexibility on spending base on our use for this project. Another cloud solution that is free is Firebase by google will also be considered, our last consideration is ThinkSpeak. ThinkSpeak is an IoT analytic platform.

3.3 Wireless Communication

Since EzRack is an off-grid system that has mobile application functionality, we need to have a way to transmit data to and from the mobile app. This wireless communication will make EzRack more versatile and can be placed anywhere on campus. Due to this fact, hardwiring to the University is no longer an option and we must look for a wireless hardware option that is compatible with EzRack. There are a few popular options available for M2M communication (machine to machine) for EzRack: ZigBee, Wi-Fi, 2G/3G, Bluetooth. The goal is to have a wireless communication that can transmit and receive data between EzRack and the mobile application at end user. Since ZigBee and Bluetooth are rated for short range communication that only range up to 100 meters [TL9], this is not ideal for EzRack due to it limited range. While 2G/3G wireless communication on cellular bands look attractive but it does not follow our low-cost project goal. With the product line of ESP8266 that is compatible to EzRack embedded system that is cost effective, Wi-Fi communication is used on EzRack for wireless communication.

3.3.1 Wi-Fi

Wi-Fi abbreviated from “Wireless Fidelity” is a networking device that uses radio technology from IEEE 802.11. The device has a low range antenna that can connect to any AP (access point) that is broadcasting its SSID (service set Identifier, or “network name”). Wi-Fi networks operate in the unlicensed 2.4 GHz (802.11b/g/n) and 5 GHz (802.11a/n) radio bands, with an 11 Mbit/s (802.11a/n) or 54 Mbit/s (802.11a or g) data rate [TL4]. Refer **to Table 3 Wi**-Fi Technology shows the overview of 802.11x protocol. The Wi-Fi network supports many types of devices and applications such as printer, mobile phone, laptop, data collector, Desktop PC, PDA, Desk Phone just to name a few. Here EzRack falls into these types of applications. Therefore we is using an ESP8266 Serial Esp-01 Wi-Fi module for wireless communication. The Esp-01 module features 802.11b/g/n which will connect to most Wi-Fi access point devices [TL5].

Wi-Fi Technology				
802.11 Protocol	**Release**	**Freq. (GHz)**	**Type throughput (Mbit/s)**	**Max net bitrate (Mbit/s)**
-	1997	2.4	0.9	2
A	1999	5	23	54
B	1999	2.4	4.3	11
G	2003	2.4	19	54
N	2009	2.4/5	74	600

Table 3 Wi-Fi Technology

Update: Due to issues with the connectivity, the Wi-Fi module was changed to the ESP8266 12E model.

3.4 Charge controller

Since EzRack is an off-grid system that utilizes solar panels to charge the battery which resulted in the need for a charge controller sometimes known as battery regulator. A charger controller controls the flow of the power coming out of the solar and regulate the rate at which electric current is added to or drawn from electric battery. A typical charge controller offers protection against overcharging which result in better battery performance. It may also prevent a battery from completely drained. As a result this can prolong the battery lifespan. To design and implement a charger controller for system many factors must be taken into account. Such factors include temperature, sunlight hours, solar panel ratings, desired output of the system (12V, 24V) just to name a few.

Currently there are 3 types of solar charger controllers:
1. Simple 1 or 2 stage controls
2. PWM (pulse width modulated)
3. Maximum power point tracking (MPPT)

The two main type of charger controller: PWM (Pulse Width Modulation) charge controller and MPPT (Maximum Power Point Tracking) charge controller. Though both of which are charge controllers.

3.4.1 Simple 1 or 2 Controls

This type of charge controller uses a shunt transistor or relays to control the voltage in 1 or 2 steps. The controller shorts the solar panel when at certain voltage. This is an older concept which have a reputation of being reliable. This type of regulator only use the instantaneous battery voltage to determine the battery state of charge [TL6]. In single stage regulator, the charging procedure is regulated by the cut-in and cut-off setting. When the battery is charged up to its cut-off voltage, the regulator stops flow of current from the solar panel to the battery until the battery voltage falls into cut-in level. In two stage or dual stage regulator, there is a second pair of cut-in and cut-off setting known as “Float Mode” that help maintains the battery voltage at lower level once the battery is charged up to a high voltage (stage one). Float voltage allow the battery to be maintained indefinitely therefore the system is reliable [TL6].

3.4.2 PWM (Pulse Width Modulated)

This type of charge controller control the current flow by analyzing the battery’s condition and recharging requirements. The PWM does this by checking the battery to determining how long and how fast the pulse should come. It then self-adjust to the appropriate pulse to charge the battery. Pulse may be long and continuous for a drained battery. As the battery become charge the pulses become shorter or trickled off.

A PWM controller is not a DC to DC transformer. The PWM controller is a switch which connects solar panel to the battery. When the switch is closed, the panel and the battery is at nearly the same voltage [TL3]. Figure 4 below shows the relationship of current and voltage curve.

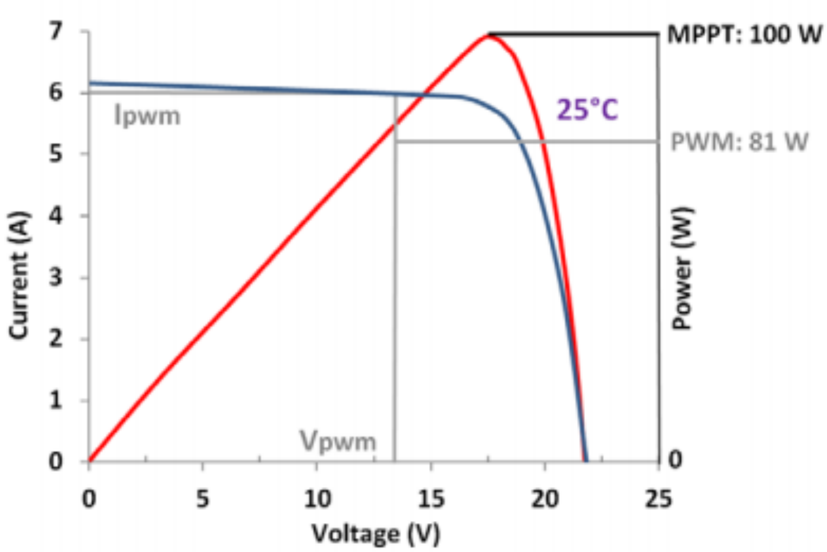

Figure 4 PWM Charge Controller
(Reprinted with permission from VictronEngergy.com)

To further explain Figure 4, it is illustrating a 12V PV system that is powered by a 100 Watt Crystalline PV panel with the specification below in table 4.

Maximum Power (Pmax)	**100W**
Open Circuit Voltage (Voc)	**21.60V**
Short Circuit Current	**6.32A**
Max Power Voltage	**17.4V**
Max Power Current	**5.75A**
Max System Voltage	**1000VDC**

Table 4 Specification from figure 4

Blue line is the current that initially starts at the “shorted circuit current” which is 6.32 Amp the starts to decrease as the voltage level increases.

Red line is power that starts at 0 Watt. It increases linearly as the voltage and the current value increases then decrease exponentially after reaching maximum power.

Grey line indicate Voltage at Pulse Width Modulation and Current at Pulse Width Modulation.

When the battery is at full discharged level, a voltage charge request by the PWM charge controller. Here the PV panels will supply 12 Volts with the addition of 0.5 Volts (voltage drop in the cabling and charge controller). The PWM charge controller will then increase charge voltage slowly until full charge level is reached. The maximum wattage provided by PWM charge controller is 12.5V x 6A =74W. Notice that the maximum power is 25% lower than the maximum power rating of the PV panel.

This type of charge controller has several benefits:
1. PWM charge controller increase charging efficiency, allow for rapid recharge and help maintain healthy battery life.
2. Automatically adjust its charging to older batteries.
3. Helps avoid with gassing and heating to the battery.
4. Increases charge acceptance.
5. PWM charge controller hold battery cells in better balance through equalization, which evens out the acceptance of charge to avoid capacity deterioration.
6. Provide better battery capacity maintenance due to increase in numbers of charge and discharge cycles.
7. Recover lost of capacity by deterring sulfate deposit formation in lead-acid batteries.
8. PWM self regulate with drops in voltage or temperature.

3.4.3 Maximum Power Point Tracking (MPPT):

This type of charge controller extract the maximum available power from PV module under certain condition. MPPT does this by analyzing the PV module and compares it with the battery voltage then fixes what is the best power that PV module can produce to charge the battery and convert it to the best voltage to get maximum current into the battery. MPPT is a DC to DC converter which operates by taking DC input from PV module, changing it to AC and converting it back to a different DC voltage and current to exactly match the PV module to the battery. It uses Boost converter for DC input voltage that is less than the DC output voltage. For DC input that is greater than DC output voltage, Buck converter is used.

MPPT charge controller offers several benefits:
1. Potential increase in charging efficiency up to 30%.
2. Offer the potential ability to have array with higher input voltage than battery bank.
3. Sizes up to 80 amps.
4. Offer great flexibility for system growth

The MPPT charge controller utilizes a DC to DC transformer which is tasked to transform the power from higher voltage to lower voltage [TL3]. When DC to DC converting, the DC voltage is converted into a high frequency AC voltage which then ultimately converted back to DC voltage. The figure 5 below, is showing the graph of the current vs voltage and power vs voltage in MPPT charge controller.

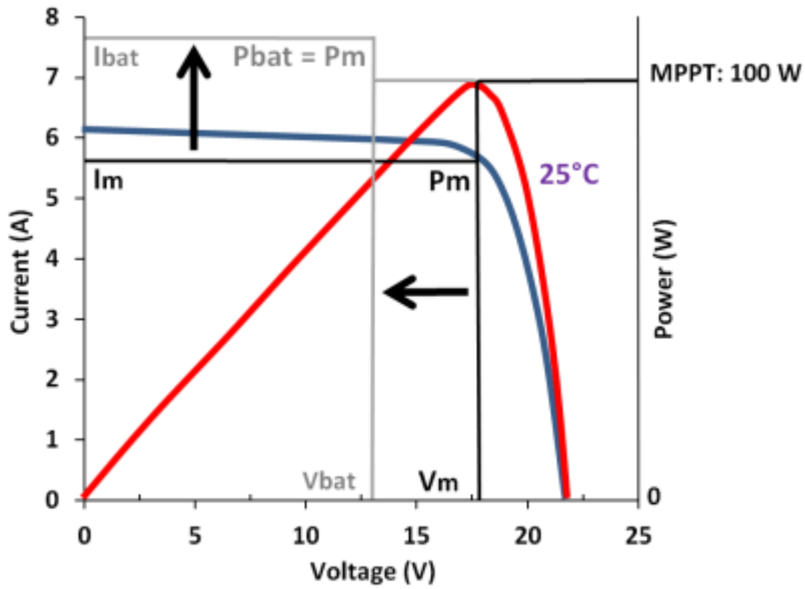

Figure 5 MPPT controller
Graphical representation of the DC to DC transformation
Pm = Vm x Im = 18V x 5.6A = 100 W, and
Pbat = Vbat x Ibat = 13V x 7.7A = 100 W
(Reprinted with permission from VictronEngergy.com)

Based on Figure 5, the MPPT charge controller does not waste voltages like PWM charge controller does. MPPT charge controller has better power efficiency due to the fact that any extra voltage is converted into current for use. That can be seen in the red line.

EzRack is an off-grid system at which the loads depends on the solar panel and the efficiency of the charge controller to maximize the utilization of the batter that has to offer. This is why our group decide to use MPPT charge controller to power our smart bike rack. With MPPT charge controller, we can expect battery to perform at it maximum power performance and to utilize the battery for a long time. Another factor in deciding MPPT charge controller over PWM charge controller is the fact that MPPT charge controller can be design and implement with microprocessor which is used to help with regulating the voltage and current more actively. Though the industry states that PWM is a cheaper option but our group decide on MPPT charge controller because it is more attractive to our project.

3.4.3.1 Perturb and Observe

Maximum Power Point Tracking uses two popular algorithm. The first algorithm is Perturb and Observe and the second algorithm is Incremental Conductance (IC). In perturb and observe algorithm, a slight perturbation is introduce. This causes the power of the solar module changes. If the power increases due to the perturbation then the perturbation is continued in that direction. When the peak power is reached, the power at the next instant decreases and hence after that the perturbation reserves. When the steady state is reached the algorithm oscillates around the peak point. This can be seen in figure 6 below.

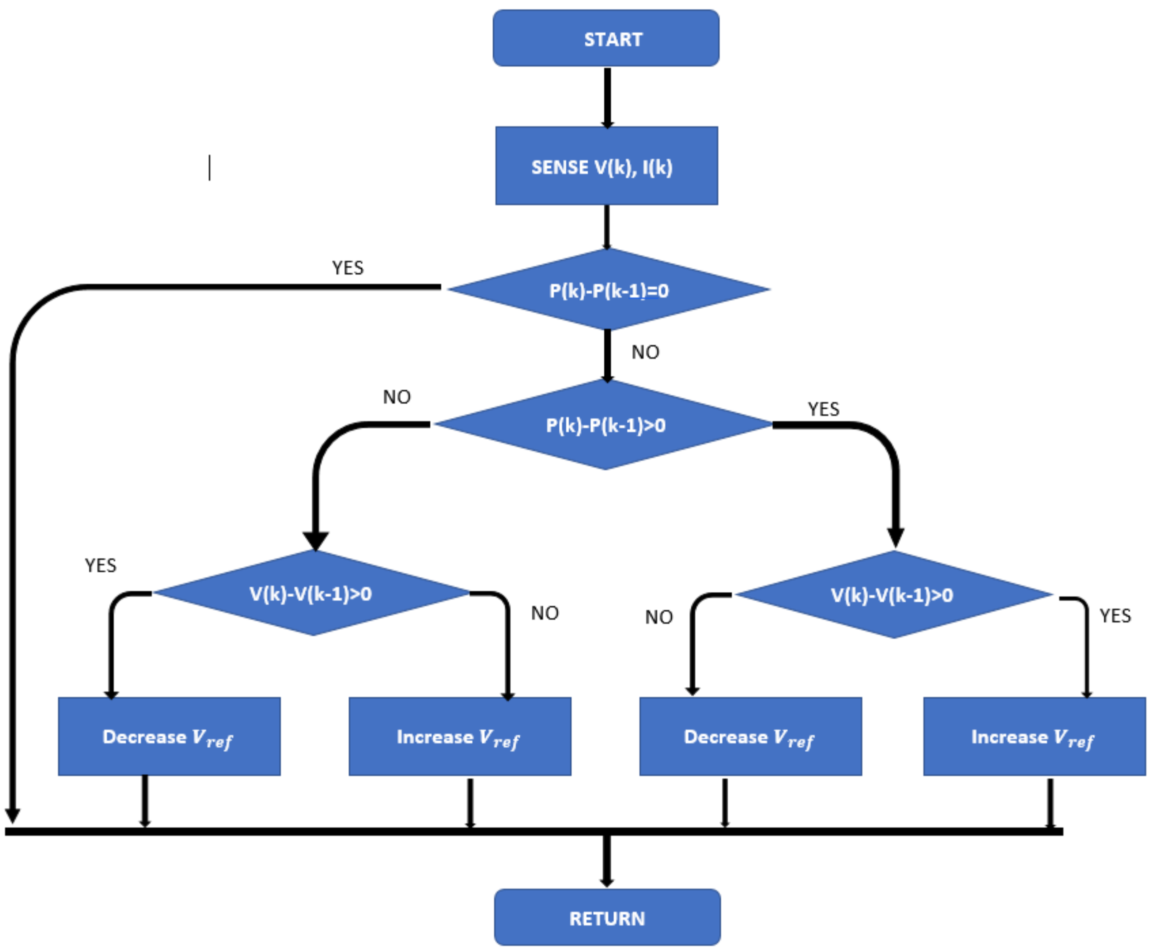

Figure 6 Perturbation and Observe

3.4.3.2 Incremental Conductance

The second popular algorithm is Incremental Conductance (IC) which can be seen in figure x incremental conductance. This algorithm will tract for the maximum power point. After tracking for the maximum point, the MPPT charge controller will set the output for the V_m (maximum voltage) and draw the maximum current $I_m$from the solar panels. Then, the ICs in MPPT charge controller determine the difference in power between the battery and the input power coming from the panels.

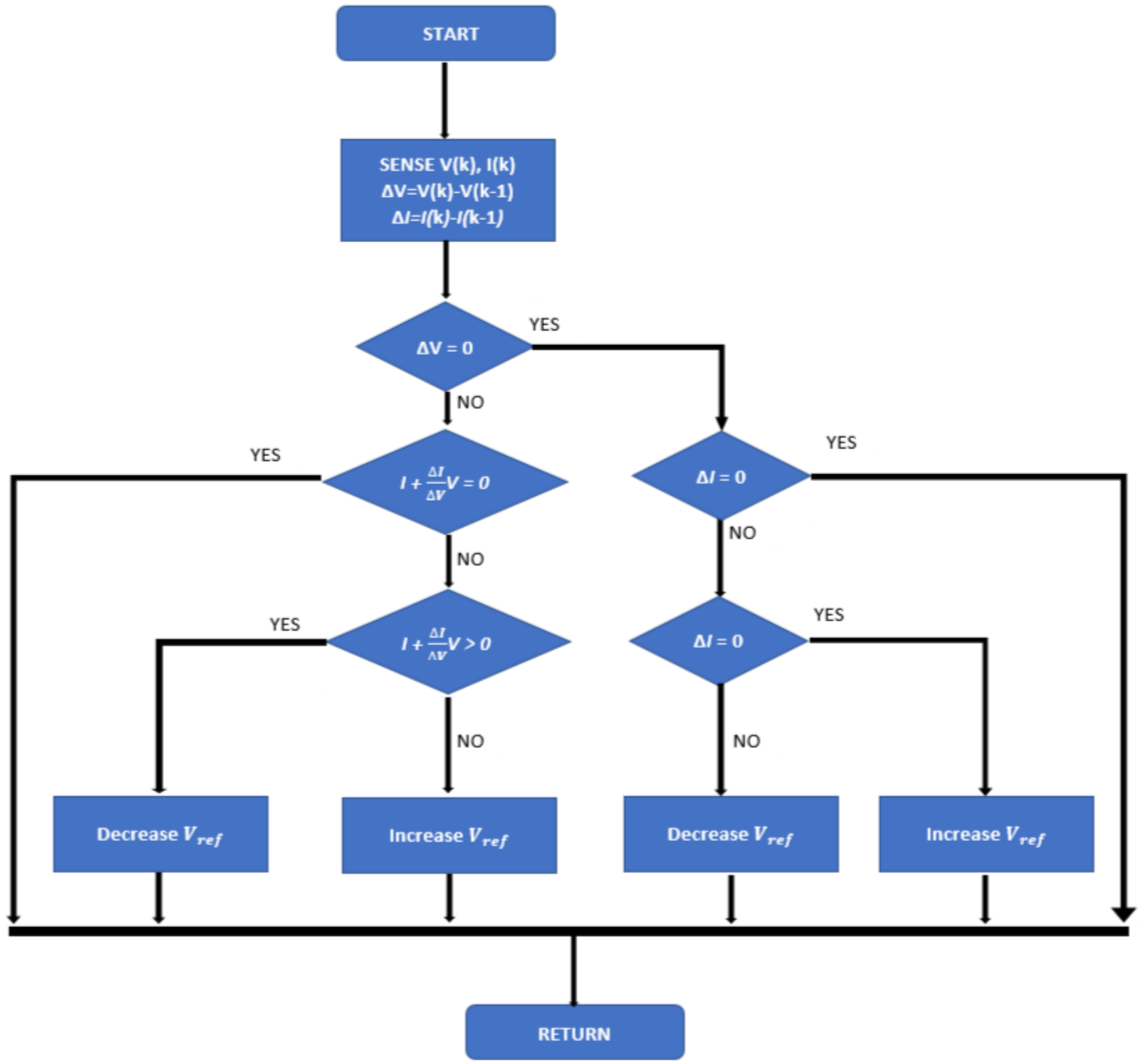

Figure 7 Incremental Conductance

3.5 Solar Panels

Solar panels are photovoltaic systems composed of cells that harvest energy from the sun to produce solar power. A typical photovoltaic system contains many cells to generate and store a usable amount of power. Each module may only be able to produce a limited amount of power. Therefore, a complete photovoltaic energy system will have multiple modules installed. A solar panel cannot simply be hooked up directly to an output, however. These systems include not only the photovoltaic solar panel, but must also be equipped with inverters, battery storage, wiring, and even a monitoring system. A solar energy system can be designed in many ways. There are many options in the technology nowadays that offer many different uses and efficiencies. A solar cell efficiency can be affected by factors such as incident light, the material used to build the panel, and the complete structure. For this project, we is building and operating the prototype model under ideal conditions for use of any solar panel. We are in an ideal location for solar power and need not worry about the performance rating of the panel itself.

3.5.1 Crystalline Silicon Photovoltaics

The Figure 8 below shows the structure of monocrystalline cells and polycrystalline cells.

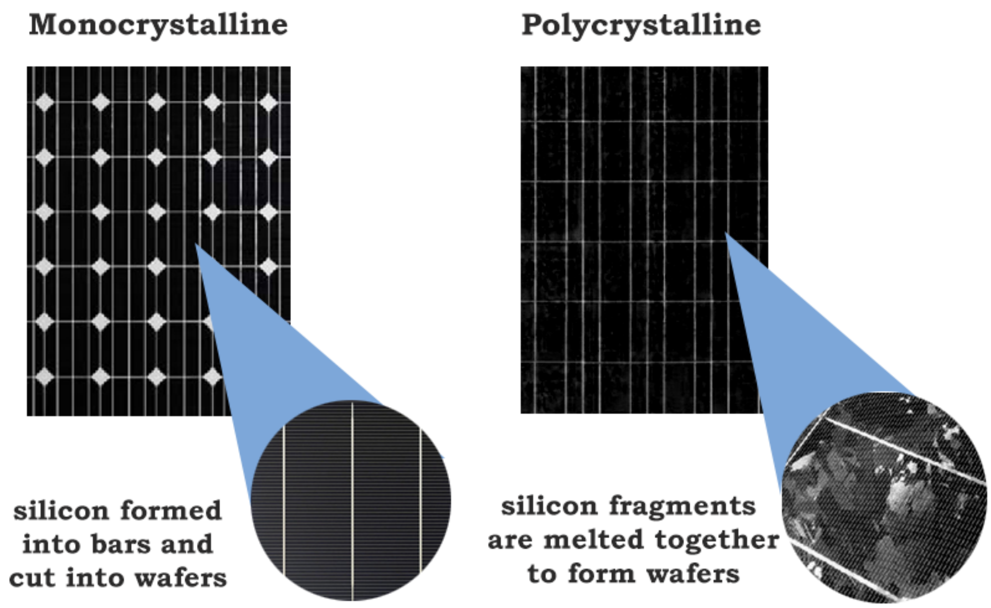

Figure 8 Silicon formation of photovoltaic cells

Current, the most used photovoltaic cells are crystalline silicon photovoltaics. About 90% of the total production in the world is towards this technology. These are the most efficient technologies and can be considered more reliable. Solar systems can be costly. In the long run, solar energy may pay for itself. However, the technology is still a large upfront cost to build such a system. Monocrystalline cells are the more costly types of cells that are available. Although they are able to yield the highest efficiency, they are not as commonly used due to the high cost. However, a fewer amount of cells can be installed due to such high efficiency. Other solar cells that are more common are polycrystalline. The efficiency of polycrystalline cells are less than the monocrystalline cells due to the grain boundaries that are present. This presence reduces the lifetime for the material.

Overall there are many benefits of crystalline silicon photovoltaic cells. These cells are mature, they have been researched and developed to ensure reliability and efficiency. They also have proven to maintain a long lifespan. The overall lifetimes can reach 25 years or more. Silicon is also an abundant material that can be found in the Earth's crust. There is ongoing research and development going on in this field to continuously innovate the technology and reduce costs.

3.5.2 Thin-Film Photovoltaics

Another type of solar cell that we can consider are thin-film photovoltaics. These types of photovoltaics are the thinnest and the least expensive to produce. There is a trade-off with thin-film compared to the other traditional photovoltaic panels. The trade-off is that they are less efficient. They way that thin-film photovoltaics are made is by depositing thin layers of photovoltaic material that is even thinner than silicon wafers. Due to the material and production of thin-film, they can even be flexible. Despite the limited efficiency, they are ultimately still competitive against the other types due to the structural flexibility and the lower cost. The table 5 below compares the characteristics of different types of photovoltaics.

Characteristics of Photovoltaic Cells			
Solar Cell Type	**Structure**	**Efficiency (%)**	**Lifetime (years)**
Monocrystalline	Single crystal of silicon	13-20	25-30
Polycrystalline	Multiple crystals of silicon	14-16	20-25
Thin Film	Amorphous silicon	6-12	15-20

Table 5 Comparison table of photovoltaic cells

3.6 Batteries

In the following section, we will analyze advantages, disadvantages, and limitations of different types of batteries. The following types of batteries is considered:

- Nickel Cadmium (NiCd)
- Nickel-Metal Hydride (NiMH)
- Lithium Ion (Li-ion)
- Lithium Ion Polymer (Li-ion polymer)
- Lead Acid
- Alkaline

For our off-grid system, the most efficient battery storage is needed for our photovoltaic system. The bike rack station does not have many size constraints for the battery storage as it may be implemented into the design in many ways. However, due to the bike rack station being located in an outdoor environment, we do need to be wary of temperatures. A battery that can operate under extreme conditions and remain durable in any temperature will work best in this case.

Also due to our budget limitations, the most desirable battery type for this project may not actually be the most ideal. The figure 9 below illustrates a lead acid battery connected to a photovoltaic panel.

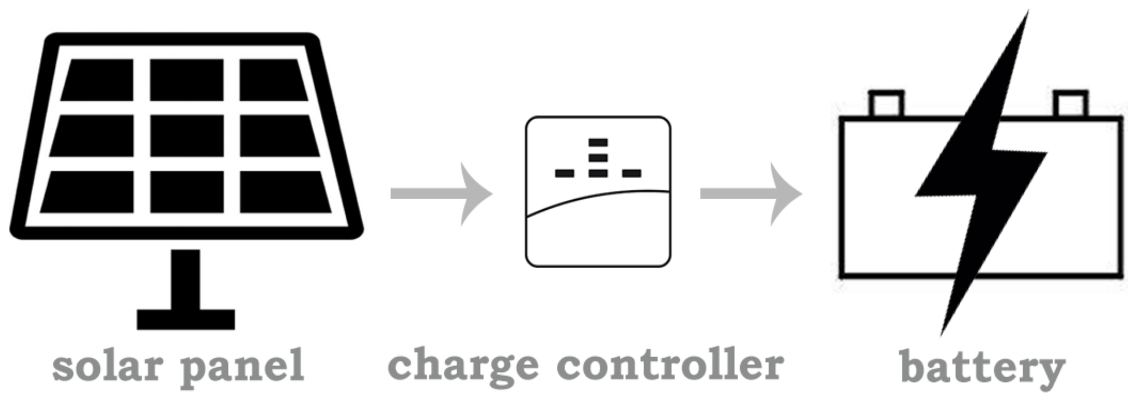

Figure 9 Power Flow of Solar Panel to Battery

3.6.1 Nickel Cadmium

Nickel Cadmium batteries are used for their long life, high discharge rate, and economical price. Some common application are in two-way radios, biomedical equipment, video cameras and power tools. The NiCd battery actually performs well under rigorous conditions. It is one of the most rugged batteries. It is ideally used for brief periods only occasionally and should not remain on a charger for an extended amount of time. They hold a long shelf life regardless of its state of charge.

Although Nickel Cadmium batteries are well understood and maturely developed, they contain toxic metals and are generally environmentally unfriendly. Also, the energy density is very low. The use of this type of battery for our project is undesirable due to this fact. The following table 6 depicts the technical aspects of a NiCd battery.

Nickel Cadmium Specifications	
Cycle Life	1000
Charge Time	1-2 hr
Self-discharge per month	20%
Toxicity	Very High

Table 6 Nickel Cadmium Specs

3.6.2 Nickel-Metal Hydride (NiMH)

Compared to the previous battery type, NiCd, Nickel-Metal Hydride batteries have a higher energy density. The higher energy density has a tradeoff for a reduced cycle life. NiMH can offer up to 40 percent more energy density than NiCd. Another difference from the NiCd battery is its use of environmentally friendly metals. There are no toxic metals present in NiMH. Nickel-Metal Hydride batteries may be used in mobile devices or laptop computers. With any type of battery, there is some limitations. NiMH requires high maintenance. A regular full discharge is required to prevent crystalline formation. This also limits its service life. Deep discharge cycles can reduce performance and shelf life. The following table 7 lists NiMH specifications.

Nickel-Metal Hydride Specifications	
Cycle Life	300-500
Charge Time	2-4 hr
Self-discharge per month	30%
Toxicity	Low

Table 7 Nickel-Metal Hydride Specs

3.6.3 Lithium Ion (Li-ion)

The Lithium Ion battery is one of the most attractive battery systems as it boasts its most promising battery chemistry. Lithium is one of the lightest metals and can provide a high energy density. However, lithium metal is unstable. Therefore, Lithium Ion is used instead. It has gained notability for being the fastest growing battery system. It is mostly used for its high energy density and lightweight property. This type of battery also requires a low amount of maintenance. Common uses include laptops and cellular devices. The following figure 10 shows how the production of Li-ion batteries is increasing.

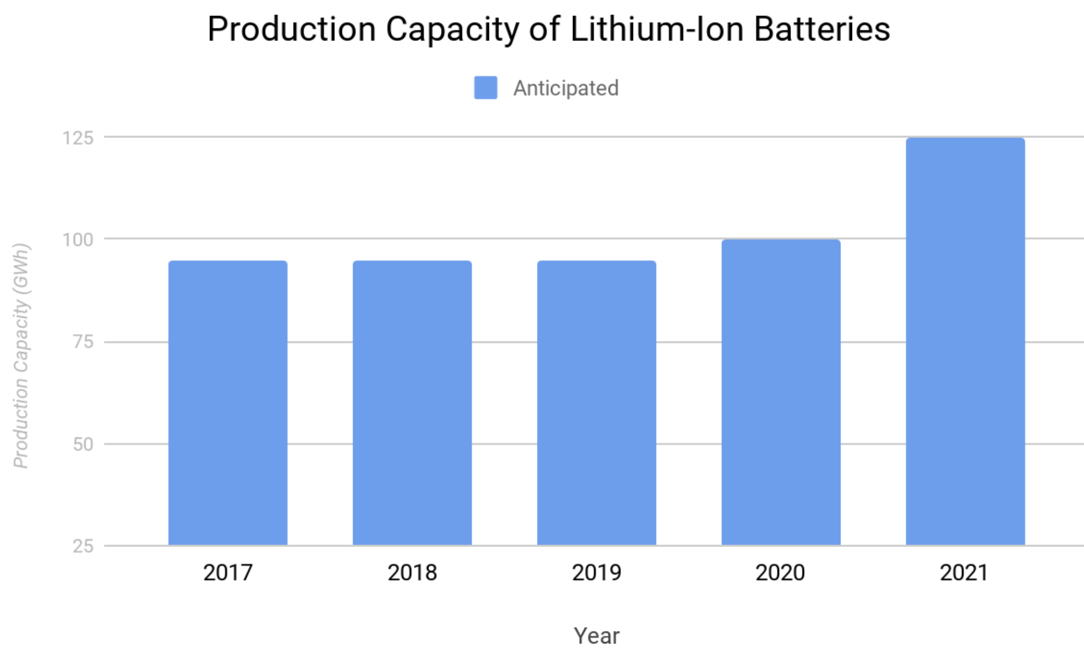

Figure 10 Production Capacity of Lithium Ion Batteries

Although it may have many advantages, there are disadvantages. To ensure safe operation, a protection circuit is needed since the technology is still fragile. Extreme temperatures are also a risk so must be limited. Deterioration is also a factor. Whether or not the battery is used does not affect its deterioration. This characteristic is not exclusive to the Li-ion. Age affects many types of battery. One way of lessening this effect is to store the battery in cool temperatures.

3.6.4 Lithium Ion Polymer (Li-ion polymer)

Lithium Ion Polymer batteries are comparable to Li-ion batteries since they use the same lithium-ion technology. However, instead of using polymer electrolyte, this battery uses a liquid one. They offer all the same attributes as Li-ion in thinner, simpler package. The main application for this type is use in mobile phones.

Due to the differences from the Li-ion battery, its capacity is smaller. There are also delays in its manufacturing and technical difficulties. The main advantage that is poses over the Li-ion is its form. There are no cost advantages. The wafer-thin form and low profile is highly demanded in the mobile device industry. These batteries can be as thin as a credit card and can be produced economically. They are also more resistant to overcharge, thus improving safety. The following Table 8 compares Li-ion polymer to Li-ion.

Lithium-ion Vs Lithium-Polymer		
Characteristic	**Lithium-Ion**	**Lithium-Polymer**
Energy Density	Higher	Lower
Aging	Loss of Charging Capacity over time	Less loss of charging capacity over time
Explosiveness	Likely when overcharged	Safe from explosion
Weight	Heavier	Lightweight

Table 8 Lithium-Ion vs. Lithium-Polymer

3.6.5 Lead Acid

Lead Acid batteries are the most economical batteries for use in large power applications. The lead acid was actually the very first rechargeable battery that was available for commercial use. The flooded lead acid battery is still used in automobiles, machines, and large uninterruptible power supply systems. There are two types of designations that are driven towards different applications. The first designation are small sealed lead acid batteries (SLA). The second designation are large valve regulated lead acid batteries (VRLA). Lead acid batteries are the preferred choice for hospital equipment, wheelchairs, emergency lights, and UPS systems. They are useful if weight is not a concern.

The Lead Acid can be on float charge for a prolonged amount of time without incurring any damages. It has one of the best charge retentions compared to other rechargeable batteries. Both SLA and VRLA have a lower over-voltage potential. This is to prohibit the batteries from reaching the potential to generate gas during charge. This is unlike to flooded lead acid battery. These batteries will never be charged to full capacity due to this characteristic. The Lead Acid battery family, compared to the other rechargeable batteries in modern day technology, has the lowest energy density. For devices that are handheld and require a compact size, it is unsuitable for use. Low temperatures also cause these types of batteries to have a poor performance. The figure 11 below shows the internal resistance vs. temperature of a lead acid battery.

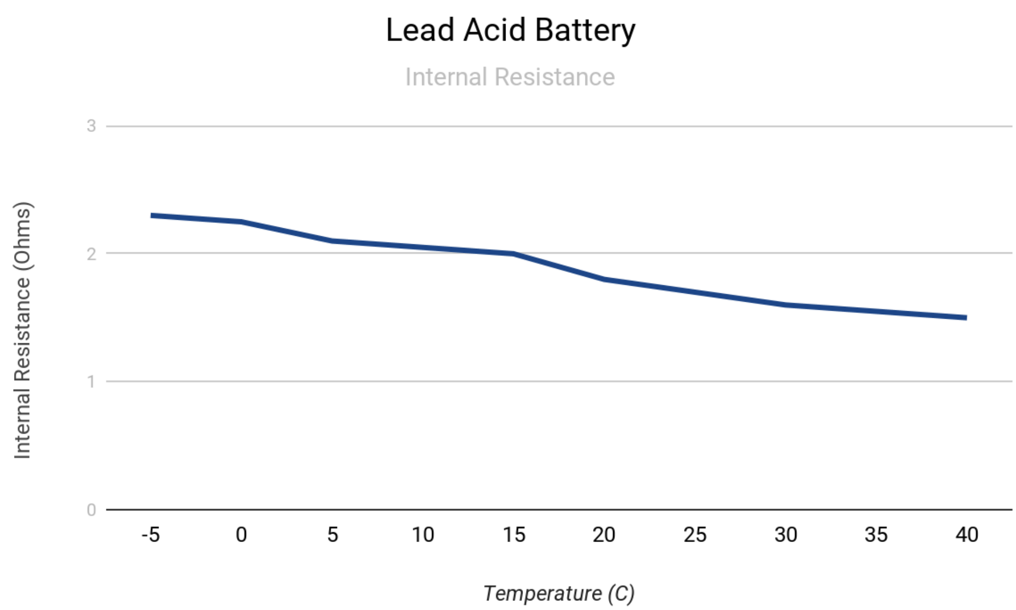

Figure 11 Lead Acid Battery Internal Resistance vs. Temperature

3.7 Locking Mechanism

In order to achieve the most efficient locking mechanism design for our project, many options for locking mechanisms need to be researched. There are many potential ways to implement an electronic lock into the housing of our bike rack. Mechanical aspects are not as important as the electrical aspects due to the nature and the motivation behind our project. In this case, we further analyze the features that can be implemented electronically to ensure adequate security for the locking mechanism.

3.7.1 Electronic locks

Electromagnetic locks are the simplest electronic locks. The lock is active when the magnet is powered and can be opened when the magnet is disengaged. A magnet is mounted to a large door frame and the corresponding armature is installed onto the door. This design, however, is not suitable for our bike rack as there will not be an attached fixture that will remain shut and be opened manually. Next, we can consider electronic strikes. This type of electronic lock is also known as the electric latch release. These locks can replace a standard strike that is mounted on a door frame. The way that these locks work is by releasing the latch on command even if the lock on the door is in the locked position. This allows the user to open the door without having to operate the door mechanically. This is also unnecessary for our design. Electronic deadbolts are another option. These locks can replace mechanical deadbolts. They offer a locking action that is electronically driven instead of mechanically driven. Most of these locks are a bit expensive for the budget of our project and the length of the bolts themselves are very short for our purposes. In this project, we must design and build our own locking mechanism using mechanical aspects driven by electrical components. Since the existing locks on the market do not fulfill our specific needs, we must look further into the design of our lock.

3.7.2 Linear Actuator

A linear actuator is capable of creating motion in a straight path. They are contrast to traditional motors that rotate. An electro-mechanical actuator is powered by an electric motor. In a simple design, the electric motor simply drives a bolt in a linear direction. The rotation of the motor causes the bolt to move in the direction of the rotation. These electro-mechanical linear actuators are capable of moving a large load over just a small distance. They can be designed in specific ways to achieve certain specifications. The speed of the linear actuator can be controlled. It can be controlled by variations of the voltage that is sent to the motor that either increases or decreases the motor speed which in turn increases or decreases the speed of the linear actuator. The figure 12 shows the design of the electric linear actuator.

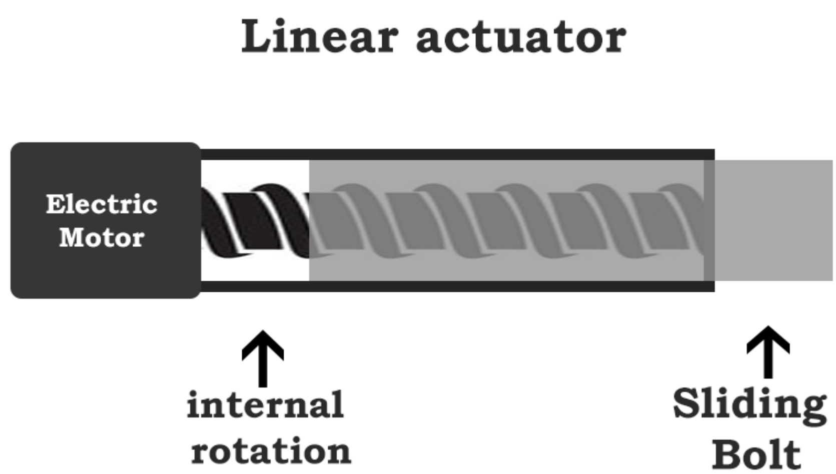

Figure 12 Linear Actuator Concept Design

3.7.3 Wireless Locks

Wireless locks are emerging as “connected” home systems are being introduced. Wireless locks, also known as “smart” locks, are electrically powered, and connected through wireless networks or Bluetooth. These locks can be accessed with a mobile device, voice commands, or even other “smart” devices that may be connected to it. These locks are currently available on the market. However, they may be costly. They can be installed in place of traditional locks. Most of these locks are designed for traditional doors and deadbolts. This would not be an appealing option for our project due to the inability to customize the locking mechanism itself.

4 Related Standards

The following section will discuss industry standards for the use of components that are contained in this project. It will cover the conversion standard on applications of DC/AC components. Battery storage and wireless standards will also be laid out for reference. Standards allow for a mutual understanding and help facilitate implementations in other designs and projects.

4.1 Android Standards

The development of the mobile application for our project is Android-based. There are Android standards and guidelines that define how an Android application should perform, be presented, and used. Standards for the Android-based applications emphasize the importance of being user-friendly and beneficial. Android applications should have a standard visual design as well as user interaction. The application must be designed and thoroughly tested for high quality performance.

4.2 DC/AC Conversion

A DC/AC conversion is needed to power a device by regulating the voltage current. AC is the form of electricity that is standard for utility power that can be directly connected to. Direct current to alternating current can be done using a power inverter. It takes the direct current, changes it to an alternating one, and switches it back to direct in order to power for example a device that requires a lower voltage than the one provided through an outlet. Outlet voltages are usually standardized so an adapter is needed to be able to draw the appropriate voltage from it.

A power inverter needs a stable DC power source to supply a current capable of powering the desired output system. Input voltage for DC has to conform to the system and the battery supply. A common standard for input voltage is around 24V DC to 48V DC for simple home systems. In this case, for our project requirements, the input voltage demands for a source such as a photovoltaic solar panel is from 200V DC to 400V DC. More specifically, a solar inverter can be used with a photovoltaic system in order to adapt to more specific functions needed for maximum efficiency. All inverters have a certification for safety and should be certified by a testing lab.

The standard for AC power also varies from country to country. The standard AC power in the United States is 120V 60Hz. This standard is for most electronic devices and appliances found in residential homes. Outlet types also vary with the standards. The outlet types that are used for the United States standards are commonly type A and type B outlets.

4.3 Solar Panels

One of the main objectives of this project is to be eco-friendly. Solar panels are a source of renewable energy. Solar power is clean energy that is directly from sunlight. Solar panels can be implemented for off-grid, stand-alone systems for energy. In order for solar systems to be reliable, safe, and most efficient, they must be built in compliance with multiple standards and codes. The International Electro technical Commission (IEC) Technical Committee 82 sets International Standards pertaining to photovoltaic energy systems. “Photovoltaic energy systems” refer to the entire process of solar power including the input of light to the solar cell to the electrical system interface [2].

Standards that are set by the IEC include:

- Terms and symbols
- Salt mist corrosion testing
- Design qualification
- Type approval of crystalline silicon
- Type approval of thin-film modules
- Characteristics of stand-alone systems

The IEC Technical Committee 82 is working towards including more standards in the future such as:

- System commissioning
- System maintenance
- System disposal
- Characterization/Measurement of new thin-film technologies (CdTe, CIS, CuInSe2, etc)
- New storage systems
- Special site condition applications (tropical zones, northern latitudes, marine areas, etc)

The IEC isn’t the only one that sets standards for solar systems. Standards are also set by the National Electrical Code (NEC) and the International Residential Code (IRC). Standards may vary from region to region. However, photovoltaic panels have come a long way since they were developed. These devices continue to be standardized for applications.

4.4 Battery Storage

Battery storage is vital in any power system. The Institute of Electrical and Electronics Engineers (IEEE) Standards Association creates standards for lead-acid batteries used in photovoltaic systems. There are risks associated with improper use or poor storage. Installations of large batteries must be taken seriously. Access to these batteries need to be limited and allowed only to certified technicians.

A constant current constant voltage (CC/CV) charge method is used by lead acid batteries. For stationary batteries, there is an average 12-16 hour charge time with large batteries requiring an average of 36-48 hours. There are other battery systems that are capable of charging much faster than lead acid batteries. There are three-stages in the charging process for lead-acid batteries. These stages are: constant-current charge, topping charge, and float charge.

The battery performance will decrease due to many different factors. Elevated heat, charging cycles, and age are all factors that affect a battery's performance. It is important to note that lead acid batteries that have been operated may begin to sulfate. Lead acid batteries must always be stored in a charged state. With an adequate charge state and a desirable operating temperature, the lead acid battery provides an efficiently high charge.

Many batteries come with a manufacturer's warranty. This will generally only cover the prorated price of the malfunctioned battery. Purchase of a new battery may ultimately be more reasonable than replacing an old battery under a prorated warranty. Risks may arise from the operation of a bad battery. In this case, the manufacturer may be willing to replace the battery at no charge in order to reduce those risks.

4.5 Wireless Standards

For the wireless components that are to be used in this project, the IEEE has defined standards. The IEEE 802.11 contains sets of standards for methods of transmission for wireless networking. Versions 802.11a, 802.11b, 802.11g, 802.11n and 802.11ac are versions that are used for wireless connectivity commonly found in home and office environments. The 802.11b standard is an extension from the direct-sequence spread spectrum (DSSS). Its modulation technique uses complementary code keying.

The IEEE 802.11 standard originally supported a slow throughput of 2Mbps. Versions 802.11a and 802.11b were amendments to the 802.11 standard that were developed at the same time. The standard IEEE 802.11b, however, was more successful due to its affordability. IEEE 802.11b defines specifications for wireless-networking for up to 11Mbps. It uses the radio frequency band of 2.4 GHz. This is the same frequency band as the 802.11 standard.

4.6 Solar Charge Controllers

Off-grid solar systems require a solar charge controller. These charge controllers have influence on the efficiency of not only the batteries but also of the overall power system. Certification standards that certify the durability and safety of solar charge controllers are the certifications IEC 62509 and IEC 62093. The IEC is the International Electro technical Commission. These leading standards contain the minimum requirements for solar charge controllers that are set for safety, protection, and correct functionality [3].

5 Hardware Design

The following section will outline the hardware design components and thoroughly describe the aspects of each system. The design details of the power system, embedded system, lock system, and schematics are included. Due to certain changes and updates in research and development, the content contained in the following sections may have been altered or taken out of the project scope and explained at the end of the respective sections.

5.1 Charge Controller Design

Based on the research of charge controllers, our group is interested in MPPT (Maximum Power Point Tracking) charge controller. The diagram which outline the MPPT charge controller can be seen in figure 13 the charge controller design is referenced from Abid Jamal from www.electronicslovers.com.

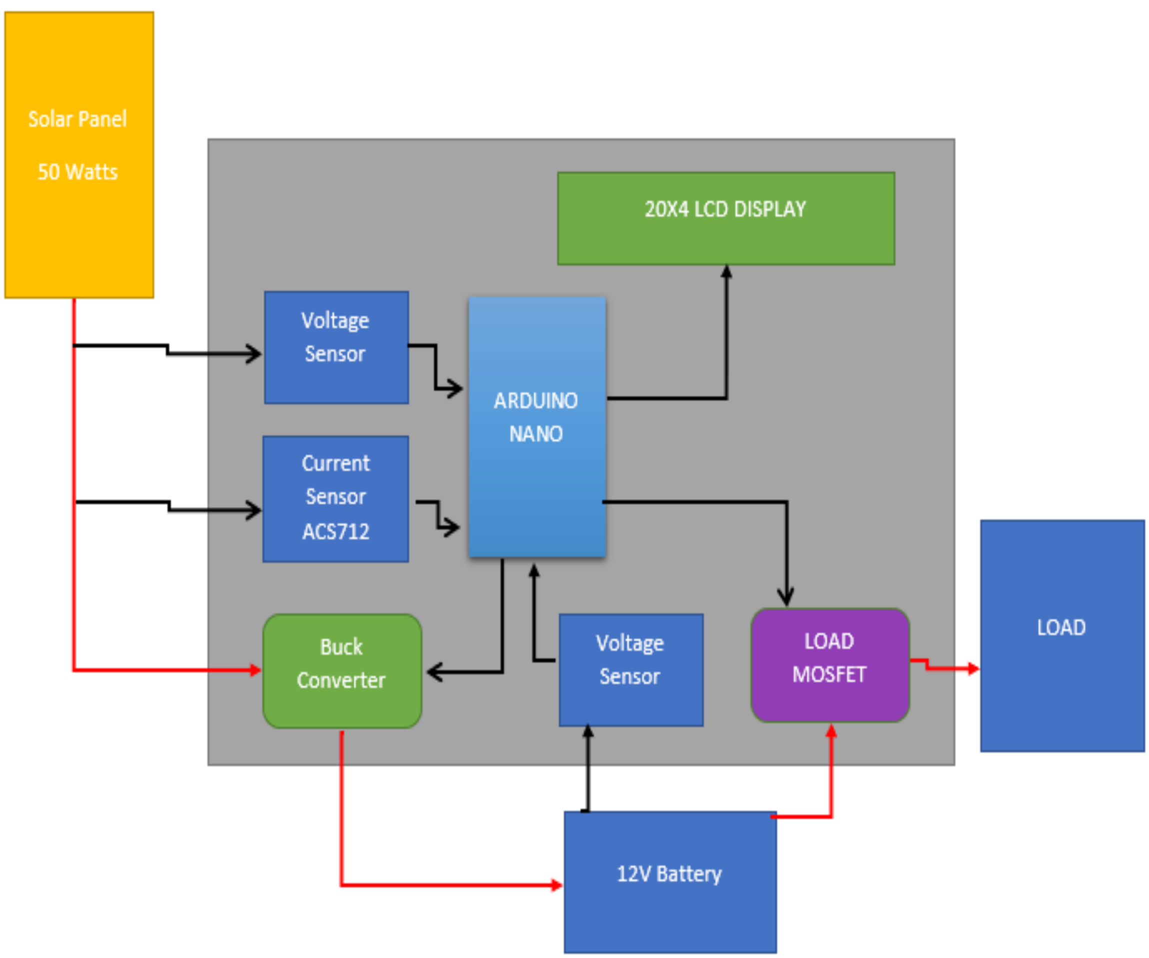

Figure 13 MPPT charge controller Diagram
(Red line indicate power, Black line indicate control)

5.1.1 Voltage Sensor

The charger controller will have voltage and current sensor. The voltage sensors is used to sense the voltage of the solar and the battery. This is done by using two voltage divider circuit. Consist of two resistors R1 at 100k and R2 at 20k for sensing the solar panel. Similarly, R3 at 100k and R4 at 20k is used for the battery. The output of R1 and R2 is connected to the Arduino analog pin A0 the output of R3 and R4 is connected to Arduino analog pin A1. This can been seen in figure 14.

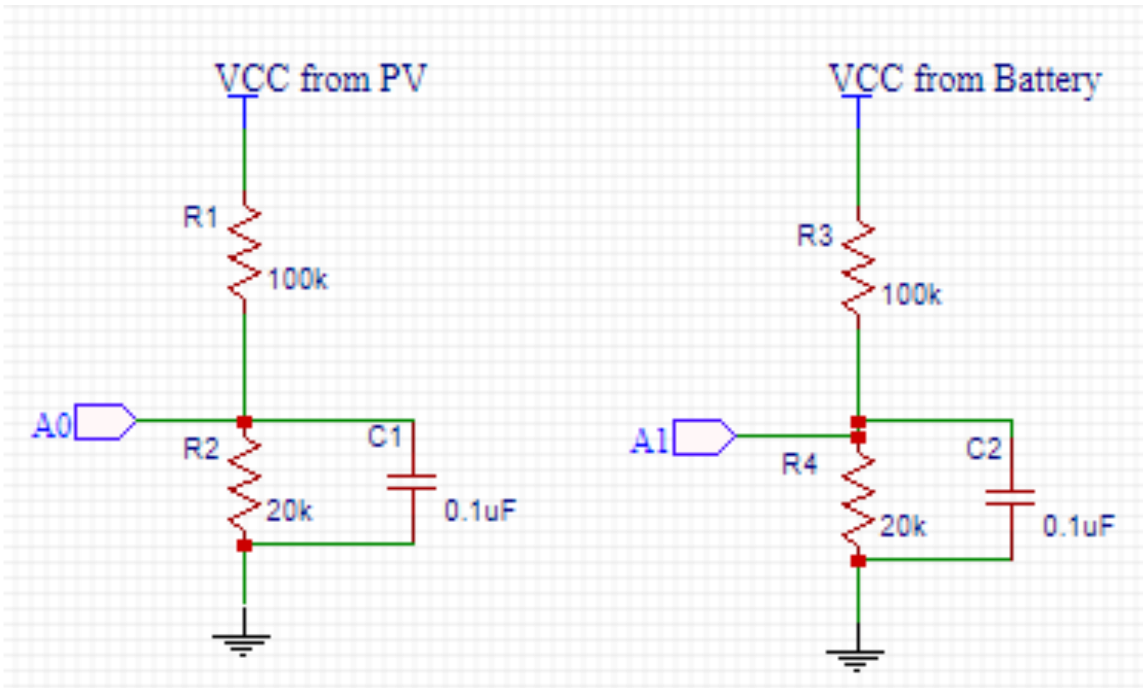

Figure 14 Voltage Sensor
(Voltage divider used to sense voltage coming from Solar panel and battery)

5.1.2 Buck Converter

For power supply efficiency and heat consideration, we is utilizing a DC-to-DC Buck Converter LM2595s instead of linear regulators to help regulate the voltage. Linear regulators are inefficient and produce a lot of heat because it consume a lot of power. Therefore we decide to use buck converter in our charge controller and it will act as our step down to provide to the load. Which is purchased to use in our charge controller. Figure 15 shows the buck converter.

Figure 15 LM2595s Step down Buck Converter.

5.1.3 Current Sensor

The current sensor is used to measure the load current and later use to calculate load power and energy. We is using ACS712-20A shown in figure 16.

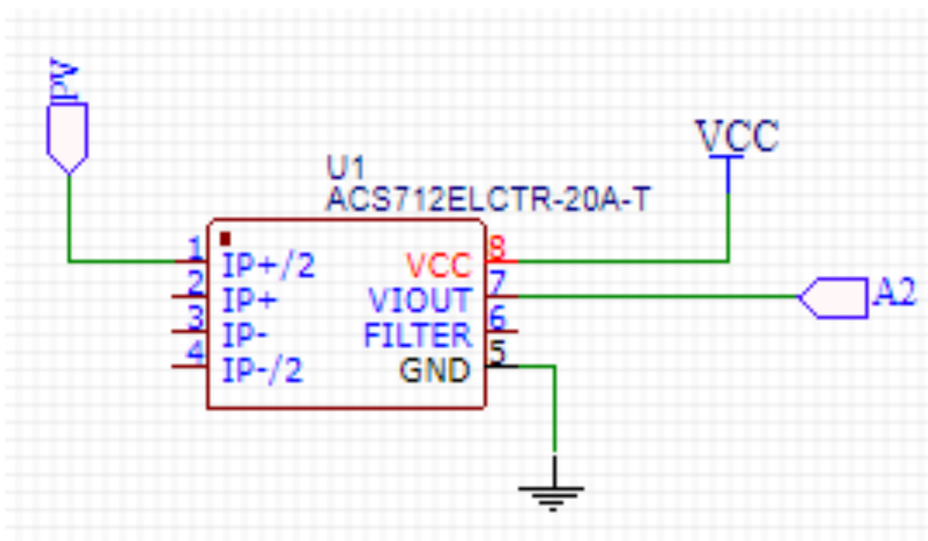

Figure 16 ACS72-20A
(Current Sensor sensing current from PV panel passing data to Arduino Nano via pin A2)

5.1.4 MOSFET and MOSFET Driver

The charge controller will utilize a MOSFET driver IR2104 also known as a Half-Bridge Driver. It is a high voltage, high speed power MOSFET and IGBT driver with dependent high and low side referenced output channel. Which is compatible for our design since it is rated at 3.3V, 5V and 15V logic compatible. We will use this driver to driver our MOSFET IRFZ44N. The MOSFET IRFZ44N features ultra-low on-resistance, dynamic dv/dt rating, 175 degrees Celsius rating and fast switching. These MOSFETs will help the charge controller with reverse power flow in both battery and solar panel. This is a critical feature that the charge controller must have to protect EzRack power system.

5.1.5 MPPT Charge Controller Logic

In order to maximize the efficiency and reliability of the charge controller, we will utilize MPPT charge controller logic. The main component in controlling the logic is the Arduino Nano microcontroller. The Arduino Nano is uploaded will provided codes to handle a MPPT “on, bulk, float of state” algorithm for tracking. It is called once in the main loop to detect which charge state the charge controller should be in.

The off-state occurs when there is no more power being generated by the solar panels. Therefore the MOSFETs are turned off in this state so that the power from the battery doesn’t leak back into the solar panel.

In bulk-state, hence the name bulk, this is where the battery will receive most of its charging and where Peak Power Tracking algorithm is run. At this state the batter will receive the maximum of current that the solar panel is generating.

In the float-state, the battery exit charging when the voltage reaches the battery max voltage. Here the state will try to keep the voltage at maximum voltage. When battery is no longer at high voltage, logic will reverse into bulk-state.

In on-state, the system detect that the solar panel wattage is too low for bulk charging state but enough low enough to go to off-state so pmw value is set the system to remain on until solar panel wattage rise.

5.1.6 Charge Controller Specifications

Based on our research regarding power to operate our system, we have decided that the specifications is the following.

The electrical specifications:
1. Rated Voltage = 12V
2. Maximum input Current = 5A
3. Load current support up to 10A
4. Input voltage = Solar panel 12 to 24V
5. Power of solar panel = 50 Watts

The specification of the charge controller are:
1. Based on Maximum power point tracking (MPPT) algorithm.
2. LED indication for the state of charge.
3. 20x4 character LCD display for displaying voltages, current, power ect.
4. Overvoltage and Lightning protection.

5. Reverse power flow protection.
6. Short circuit and overload protection.

The charge controller will result:
1. Rated Voltage = 12V
2. Maximum current = 10A
3. Maximum load current = 10A
4. Open circuit voltage = 8-11V for 6V system / 15-25V for 12V system.

The seven main function of our controller will includes:
1. Prevent battery overcharge.
2. Prevent battery over discharge.
3. Provide load control functions - automatically connect and disconnect and electrical load at a specified time.
4. Monitor power and energy - monitor load power and energy and display it.
5. Protect from abnormal condition- such as lightning.
6. Indicating and Displaying: display parameters
7. Print various parameters in serial monitor

The parts and tools in which we will use are located below in table 9 and table 10.

Parts and Tools for Charge Controller		
Name	**Part No.**	**Amount**
Arduino Nano		1
MOSFETs	IFFZ44N	4
MOSFET driver	IR2104	1
3.3V Linear regulator	1117	1
Power diode for 10A and 2A	IN5402	2
Buck Converter	LM7805	1
Current Sensor	ACS712-5A	1
TVS diode	P6KE38CA	2
Transistors	2N3904 or 2N2222	1

Table 9 Charge Controller Parts

Parts and Tools for Charge Controller (Continued)		
Name	**Part No.**	**Amount**
Diodes	IN4148 UF4007	2 1
Resistors	200 330 1k 10k 20k 100k 470k	3 3 1 2 2 2 1
Capacitors	0.1uF 10uF 100uF 220uF	4 3 1 1
Inductor	33uH-5A	1
20x4 I2C LCD		1
LEDS (Red, Yellow, Green) Fuses (5A) Fuse Holder		1 each 2 2

Table 10 Charge Controller Parts Continued

The schematic design of the in our project in figure 17 is based on the reference design from Abid Jamal at electrionicslovers.com. In the reference design, the Arduino Nano is chosen to be the control center for the charge controller. The following will list the appropriate pins that is connecting to the Arduino Nano board and demonstrated in Figure 18:

Pin Reset Use to connect with push button for reset
Pin A0 Voltage output of the solar voltage sense
Pin A1 Current output of the solar current sense
Pin A2 Voltage output of the Battery voltage sense
Pin A4-A5 Used to operate the LCD display
Pin D5 Push button
Pin D6 Regulator
Pin D8-D9 Used to operate Half-Bridge Driver IR2104
Pin D11-D13 Operate LED

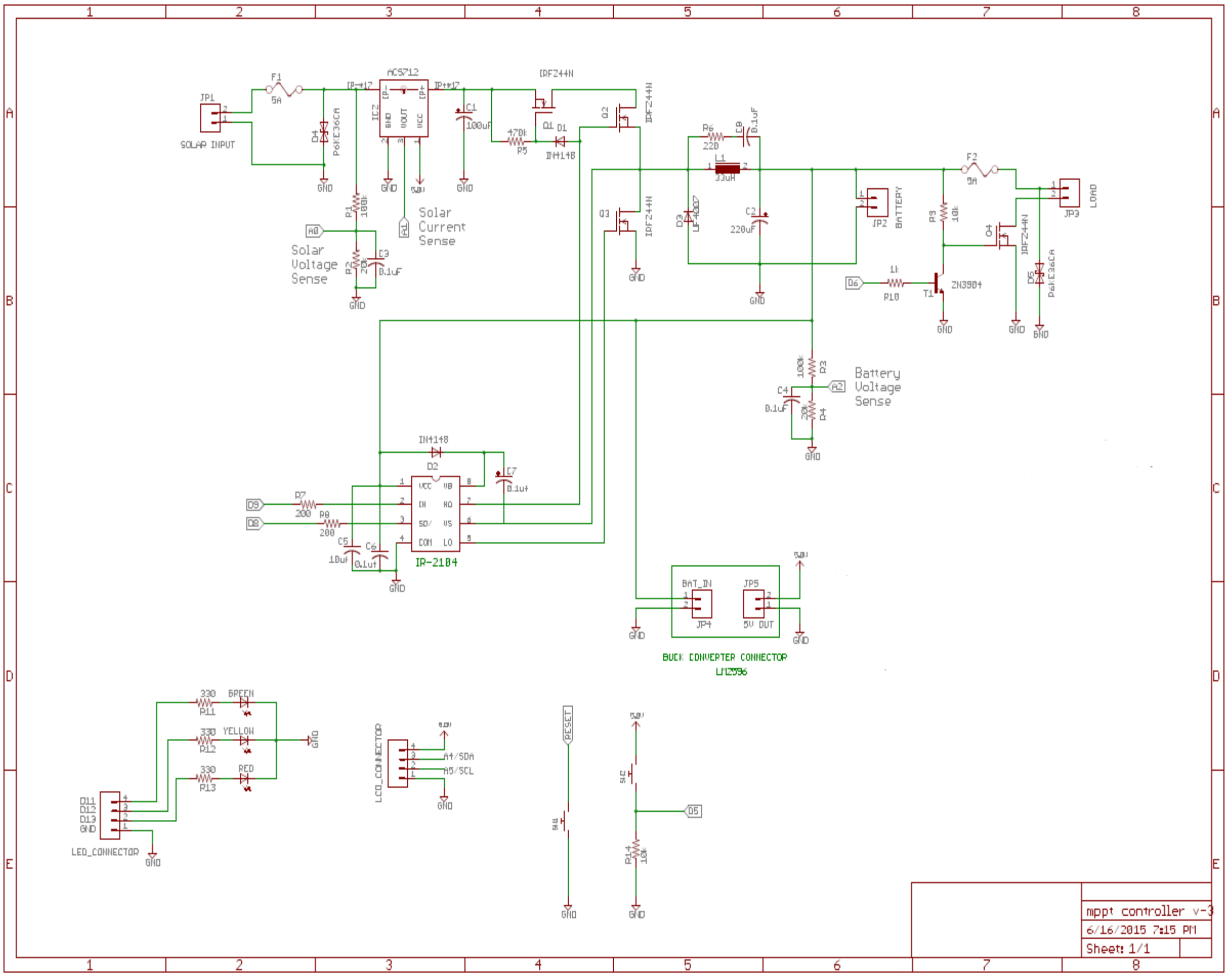

Figure 17 MPPT Charge Controller

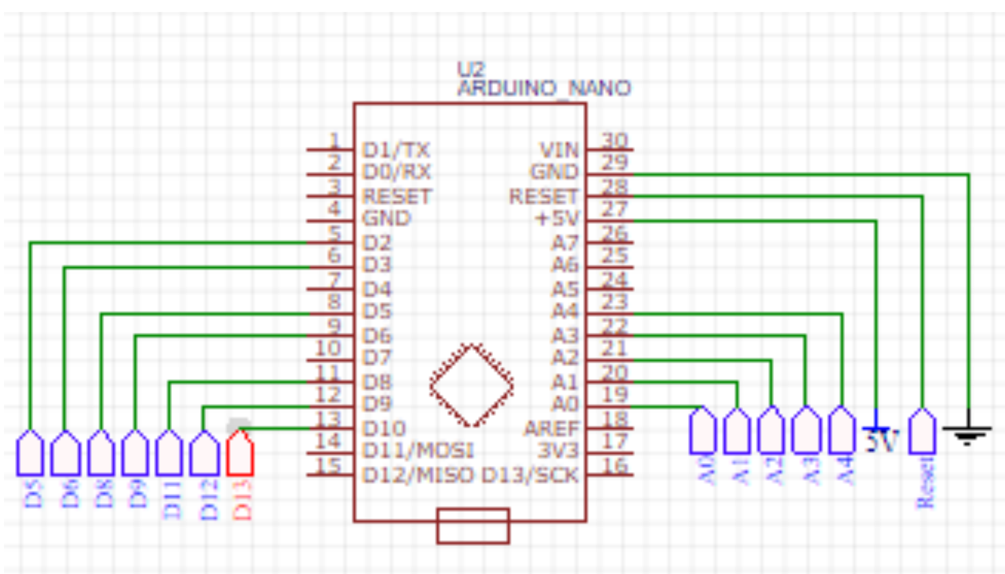

Figure 18 Connections to Arduino Nano controller

5.1.7 Charge Controller Design Alteration

After building and testing our initial MPPT charge controller, we have decided to utilize a PWM charge control design instead. This is due to the fact that our power system requires a moderate voltage and power consumption. The design of the PWM is more feasible in our system and has been optimized for our purposes. The same Arduino Nano is used to control the charging algorithm. The charging contains two stages which are bulk and float stage. Bulk charge is set to reach 14.4V and the float stage tapers down the voltage to a constant 13.6V. The figure 19 below shows the updated schematic of the PWM charge controller and figure 20 shows the final EAGLE PCB design.

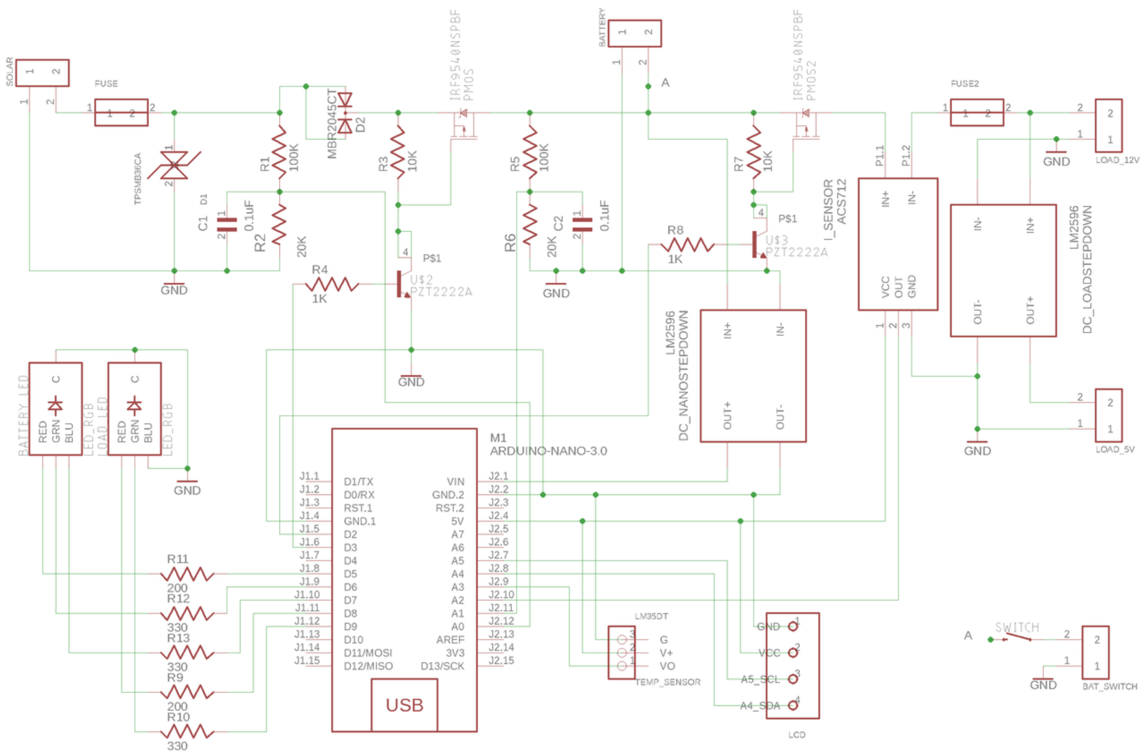

Figure 19 Charge Controller Schematic

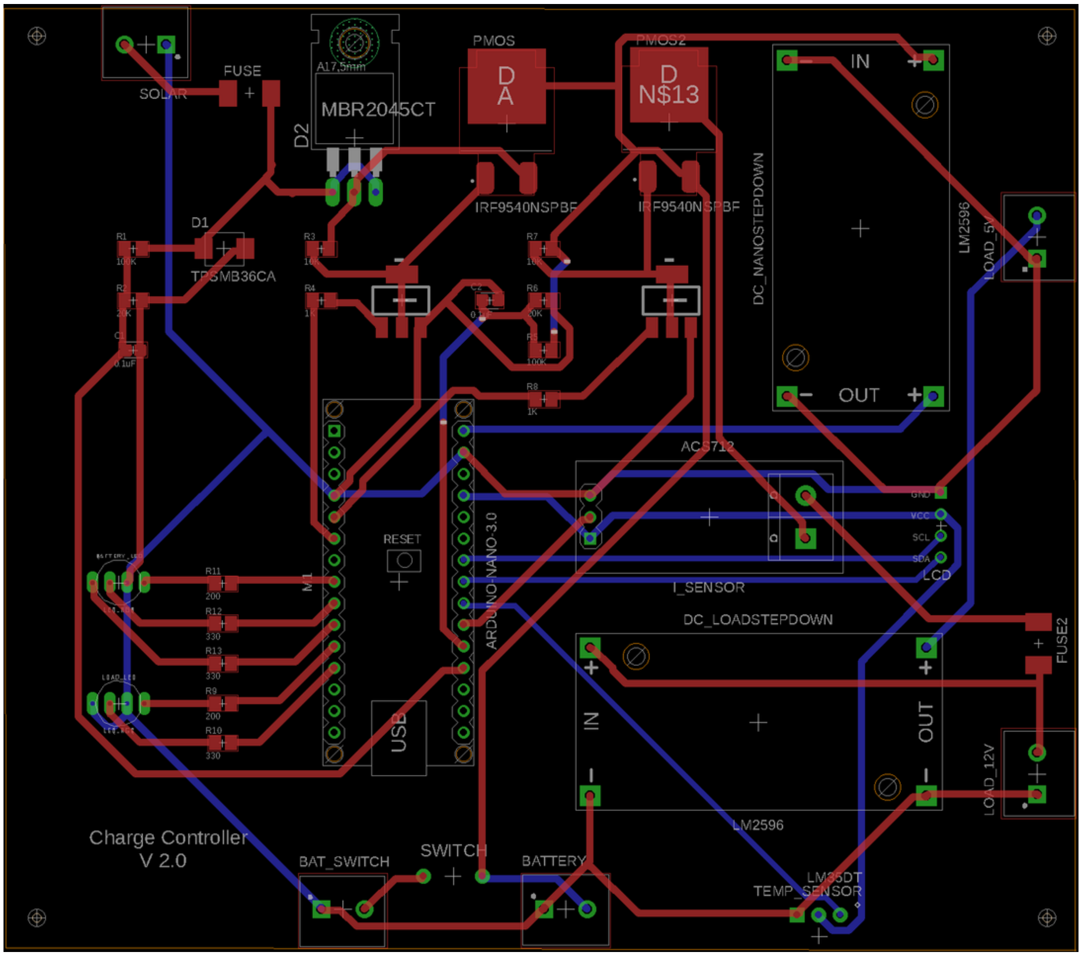

Figure 20 Charge Controller PCB

5.2 Solar Panel

In selecting which solar panel to implement into our power system, there are certain factors that weigh more than other factors. Since our project is aimed to be an off-grid system, we need to ensure that the solar panel is capable of harvesting and generating enough solar power during sunlight hours to power our bike station through the night time or times of minimal sunlight. With the designated budget, we are limited to our choices of solar panels. After researching the market for the most cost efficient solar panel that will fulfill the needs of this project, we determined that a 50W solar panel is sufficient.

The solar panel chosen is a 50W 12V Monocrystalline Solar Panel manufactured by Renogy. The table 11 contains the specifications for the monocrystalline solar panel.

50W 12V Monocrystalline Solar Panel	
Manufacturer	Renogy
Quantity	1
Cost	$88.00
Maximum Power at STC (Pmax)	50 W
Open-Circuit Voltage(Voc)	22.7 V
Short-Circuit Current (Isc)	2.84 A
Optimum Operating Voltage	18.5 V
Optimum Operating Current	2.70 A
Max System Voltage	600 VDC (UL)
Max Series Fuse Rating	15 A
Fire Rating	Class C
Weight	9.9 lbs
Dimensions	24.8 x 21.3 x 1.2 in

Table 11 Specifications of Solar Panel

5.3 Battery

For the selection of the battery to be used in our project we must consider the following factors:

- Capacity & Power
- Efficiency
- Performance

5.3.1 Capacity & Power

Since our project is using an off-grid power system, we will need a battery that has a large capacity. This is to ensure that the system is still operational in low sunlight conditions or night time. For this purpose, lead acid batteries seem to be the most ideal due to its size and capacity. Lithium Ion batteries may also be a candidate for its high energy density but since the budget it limited, stacking lithium ion batteries for a larger capacity may be too expensive.

5.3.2 Efficiency

Depth of discharge is another measure to note. Preferably, a battery with a higher depth of discharge should be used. This ensures that the system is able to use most of the battery's capacity. A solar system may not be fully charged at all times and may deplete more during hours when it is not charging. Round-trip efficiency of a battery is a percentage reference of the amount of energy that can be used compared to the amount of energy that it took to store it. To achieve the best economic value from the battery, a higher round-trip efficiency is desired. The following figure 21 illustrates round-trip efficiency.

Figure 21 Round-trip Efficiency Example

5.3.3 Performance

For the purpose of this project, the battery cycle is not expected to drain on a daily basis. Therefore, deterioration due to recharge can be ignored as a concern. However, a different factor that may affect a battery's lifespan and performance is its sensitivity to temperature. The bike rack station is located outdoors. The housing for the battery will not have insulation to help regulate temperatures. Therefore, a battery that can withstand varying temperatures is the best choice. Since the bike rack is located where sun exposure is high, the battery must be able to withstand high temperatures without deteriorating quickly or malfunctioning.

With many different battery manufacturing companies, it may be tempting to opt for the cheapest solution. However, many companies may be too small or too new to the industry. To guarantee a reliable system, we must choose a battery from a reliable source. Different manufacturers will also have different standards and warranties.

5.3.4 Battery Selection

With all of the above factors in mind, we can conclude that the best battery for our project is a lead acid battery. Lead acid batteries have been used in off grid systems for a while and are thoroughly tested. They are the least expensive battery option that still have a large capacity for energy storage. Although they do not have the longest life span or the highest DoD, they is suitable for our purposes.

The battery we will use in this project is a 12V 12 Amp Rechargeable Sealed Lead Acid battery that is manufactured by ExpertPower. The table 12 below lists the specifications for this battery.

12V Sealed Lead Acid Battery	
Manufacturer	ExpertPower
Quantity	1
Cost	$30
Weight	3.45 lbs
Voltage	12V
Current Rating	12 A
Terminal	F2 Type
Dimensions	3.9 x 3.86 x 5.94 in

Table 12 Lead Acid Battery Specifications

5.3.4.1 Alternative Considerations

Lithium ion batteries are also a suitable candidate. They are more compact and have a higher DoD and even a longer lifespan. However, the price tag on that technology is too high. The table 13 below compares the specifications of different battery types.

Battery Specifications					
	Battery Type				
Specification	**NiCd**	**NiMH**	**Li-Ion**	**Li-Ion Polymer**	**Lead Acid**
Gravimetric Energy Density (Wh/kg)	45-80	60-120	110-160	100-130	30-50
Internal Resistance (mOhm)	100-200 (6V Pack)	200-300 (6V Pack)	150-250 (7.2V Pack)	200-300 (7.2V Pack)	<100 (12V Pack)
Cycle Life	1500	300-500	500-1000	300-500	200-300
Fast Charge Time	1 hr	2-4hr	2-4hr	2-4hr	8-16h
Overcharge Tolerance	moderate	low	very low	low	high
Self-discharge / month	20%	30%	10%	~10%	5%
Cell Voltage	1.25V	1.25V	3.6V	3.6V	2V
Operating Temperature (degrees C)	-40 to 60	-20 to 60	-20 to 60	0 to 60	-20 to 60
Maintenance requirement	30 - 60 days	60 - 90 days	none	none	3 - 6 months
Typical Cost	$50 7.2V	$60 7.2V	$100 7.2V	$100 7.2V	$25 6V
Commercial Use Began In	1950	1990	1991	1999	1970

Table 13 Comparison of specifications of different battery types

5.4 Locking Mechanism

The following section will describe the design of the locking mechanism of the bike rack. The material will briefly cover the mechanical design as the main focus is on the electrical components and the software communication. The locking mechanism design is a crucial aspect for our project.

5.4.1 Bicycle Slot

An important feature of this bike rack is its ability to secure any type of bike to the rack. Bicycles range in size with varying aspects changing from the frame to the rim to the thickness of the tire. To ensure that this bicycle rack is universal for use with any type of bicycle, the locking mechanism will lock the rim to the frame. This is achieved by a locking mechanism that is constructed with a lock actuator, a motor driver, and a lever system. A metal rod is sent through the frame of the bicycle rack to secure the rim of the bike in the slot. To ensure correct placement of the wheel in the slot to allow the locking rod to pass through, there is an indicator of the path of the locking mechanism. This requires small action from the bike owner to place the bike and ensure that the rim allows clearance for the lock. The height and diameter of where the locking mechanism is placed shall allow most wheel sizes and spoke variations to be held securely in place. Average bike wheels can vary from 26 inches to 28 inches in diameter. The clearance in between the two posts of the slots is a minimum of 6 inches to allow any wheel width to fit the slot. In Figure 22, the graphic depicts how average sized bicycle wheels can be placed in the slot with the lock mechanism indicator placed in between the spoke of the wheel.

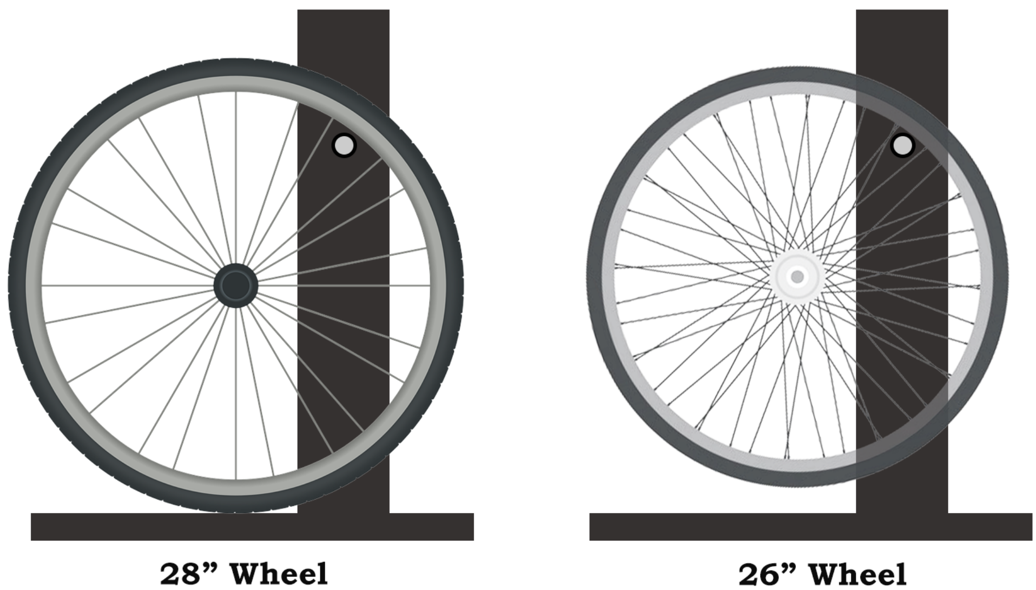

Figure 22 Side view of lock operation with average wheel sizes

5.4.2 Locking Mechanism Design

For the design of the locking mechanism, a lever type action will provide a secure lock with low power consumption. A door lock actuator with a push and pull type action is attached to a long rod. This long rod is attached to the metal rod that is used to secure the wheel to the rack slot. Since the group is primarily made up of all Computer Engineering majors and Electrical Engineering majors, the mechanical design of the lock itself is as secure as possible with a focus on the electrical aspects. This type of locking mechanism was chosen for being able to satisfy the locking requirement with the least amount of mechanical design and low power consumption. Since the bike rack is using solar power off-grid, this type of lock is most efficient. The lock actuator will push the lever rod to unlock the slot. For the slot to be locked, the lock actuator will pull the lever to push the metal rod through the slot. This action is depicted below in Figure 23.

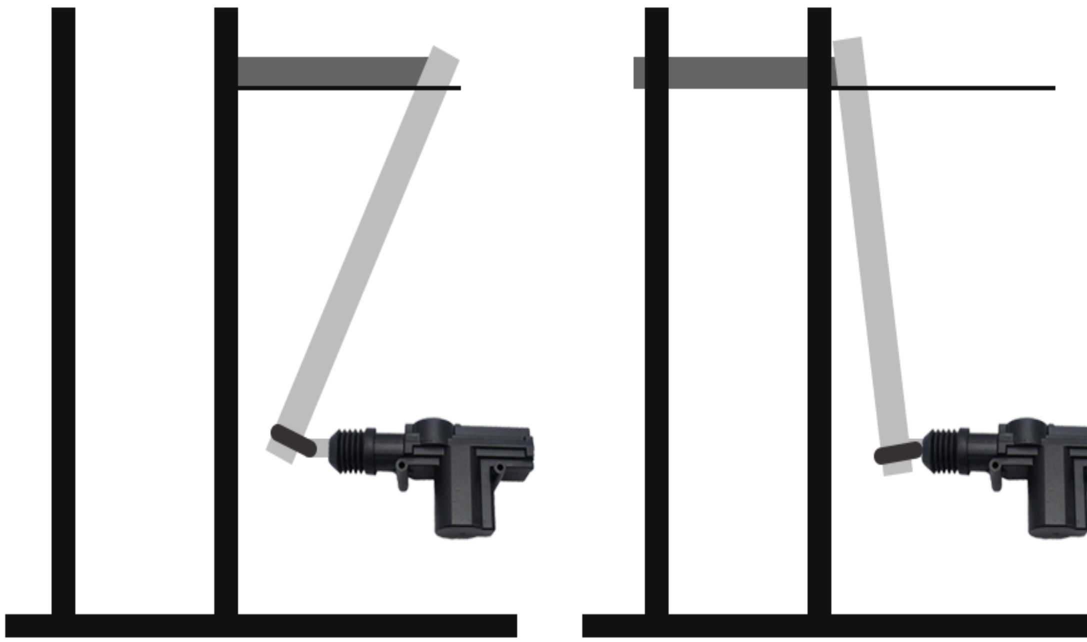

Figure 23 Locking mechanism design

5.4.3 Lock Motor Driver

A motor driver is designed using a L293D driver. This is a common driver that is used in DC motor drivers. The L293D device is a quadruple high-current half-H driver that can drive up to 600 mA at a voltage range of 4.5 V to 36 V. An InstallGear 12V universal motor actuator is connected to a motor driver. For this motor driver a L293D driver chip is used to control two actuators since our prototype bike rack will house two bike slots on the bike rack. The input voltage enabling the L293D driver itself is 5V and will come from the main PCB microcontroller while the actuators are powered by 12V that will come directly from the charge controller. This motor driver is located on a board that is placed near the actuators on the bike rack. The schematic of the motor driver is shown below in Figure 24.

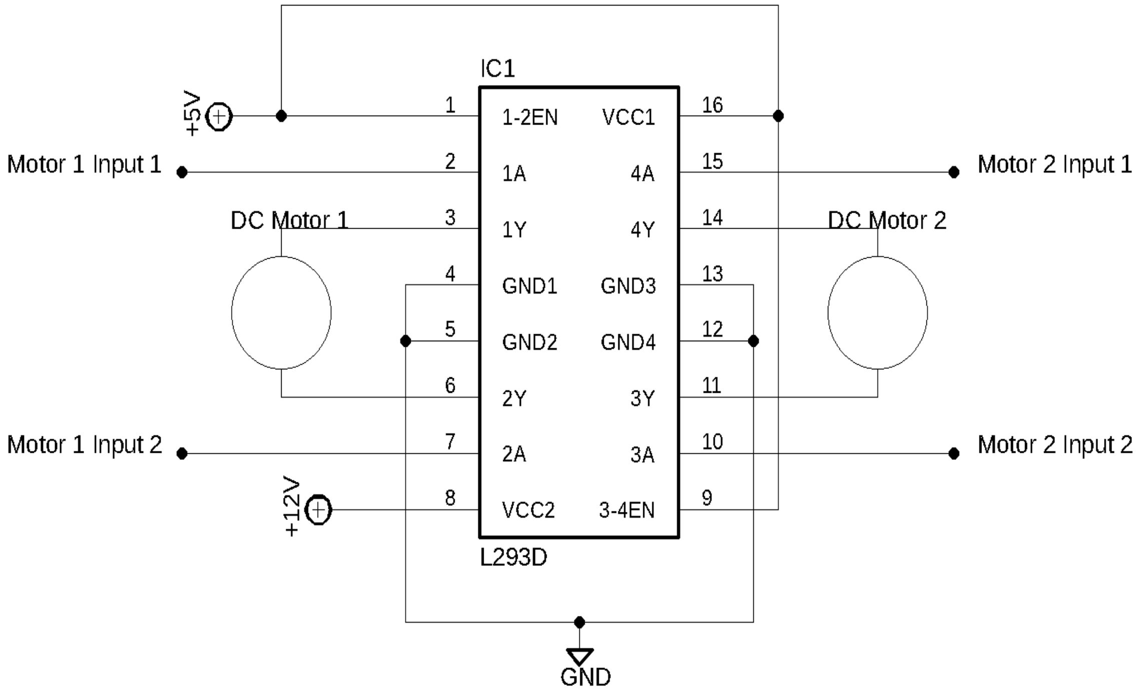

Figure 24 Schematic for motor driver

Update: After developing this design for the lock motor driver, the choice was made to change the IC to the L298N driver. This was done to improve the performance of the locking mechanism. Testing proved to us that the L293D was unable to handle the current draw for an extended period and therefore the change was made to utilize a full-bridge driver over the half-bridge.

5.4.4 Motor Driver Communication

The motor driver will communicate to the main microcontroller on the PCB through 4 wires. Two inputs are needed to operate each motor, therefore two inputs will control one motor while another set of two inputs control the second motor. The microcontroller allows communication between the software and the embedded hardware. Each passcode generated in the application for the bike rack station is used to access the locking mechanism. The unique code is sent through the microcontroller that will communicate with the driver, which will identify which lock to operate and which operating position the actuator shall be in. The Figure 25 below shows the connections between the motor driver to the actuators and the microcontroller with the power sources connected respectively.

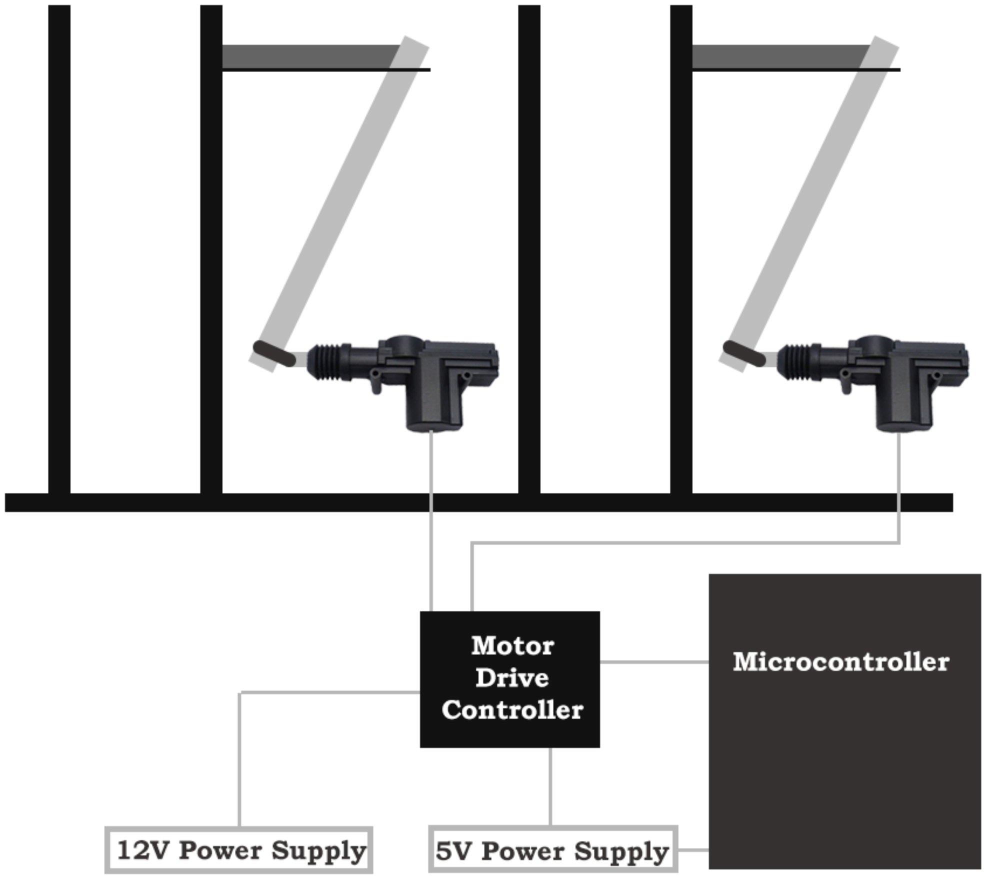

Figure 25 Locking Mechanism control connections

5.5 Embedded System Overview

The following section will provide an overview of the embedded system. The embedded system is at the heart of EzRack and is what makes EzRack stand out from its competition. It is through the embedded system that EzRack's more advanced features, such as reserving a slot in advance through the mobile app, are made possible. Great care was taken in selecting the appropriate microcontroller and peripherals.

5.5.1 Central System

Selecting the appropriate microcontroller is of the utmost importance since it plays a central role in the workings of the project. When searching for a microcontroller, there are many options to choose from. Many companies manufacture microcontrollers and each have many models to choose from. Texas Instruments, Intel Corporation, and Atmel are just a few of the companies that produce excellent microcontrollers. When selecting the microcontroller that best fit our needs, several things were put under consideration. The architecture, features/specifications, price, reliability and ease of use.

The two main architectures in use today are Harvard Architecture and Von Neumann Architecture. Without going into much detail, Von Neumann Architecture uses a single data and address bus to communicate from the CPU to RAM and ROM. This makes Von Neumann architecture very slow. In comparison, Harvard Architecture uses two separate bus resulting in faster instruction execution times. Whether the microcontroller uses RISC (Reduced Instruction Set Computer) or CISC (Complex Instruction Set Computer) based architecture must also be taken into consideration. The RISC architecture performs all arithmetic, logical, conditional and Boolean instructions in one or two cycles while CISC based architecture takes four or more cycles. For the purposes of this project, a microcontroller with Harvard Architecture and a RISC instruction set are preferred.

Initially, the microcontrollers considered were Texas Instruments' msp430 family of microcontrollers, ATmega328p, ATmega2560, Atmega32U4, and ATmega1168. Atmel's AVR microcontrollers where considered since they are used in Arduino's development kits while TI's msp430 family of microcontrollers where considered because of their ability to save power with their different low power modes. Since the msp430 family employs the Von Neumann architecture, they were taken out of consideration.

Due to our time constraints, it's vital for the microcontroller chosen to be easy to work with and have plentiful documentation. Due to Arduino's current popularity, there is a dearth of information available. Arduino has a large community surrounding it with plentiful libraries, online tutorials, and open forums thus making it an ideal choice. Table 14 lists a few of the microcontrollers used in Arduino development kits along with their specifications. Out of the four microcontrollers listed in the table, the two that stand out are the ATmega328P and the ATmega32U4. Their price, CPU speed, and memory are ideal for the project. The ATmega328P is cheaper and more reliable than the ATmega32U and so was chosen.

Processor	**Average Price**	**Operating Voltage**	**CPU Speed**	**IO Pins**	**Memory**
ATmega328p	$2.20	5V	16 MHz	23	32K
ATmega2560	$12.35	5V	16 MHz	86	256K
Atmega32U4	$4.20	5V	16 MHz	26	32K
Atmega168	$3.68	2.7V	8 MHz	23	16K

Table 14 Comparison of MCUs

5.5.2 Embedded System Overview Block Diagram

The embedded system is comprised of six major components, the microcontroller, LCD, Keypad, RTC Module, Lock, and the Wi-Fi Module. Figure 26 demonstrates the relationship between the components. All the components comprising the embedded system is discussed in this section except for the lock.

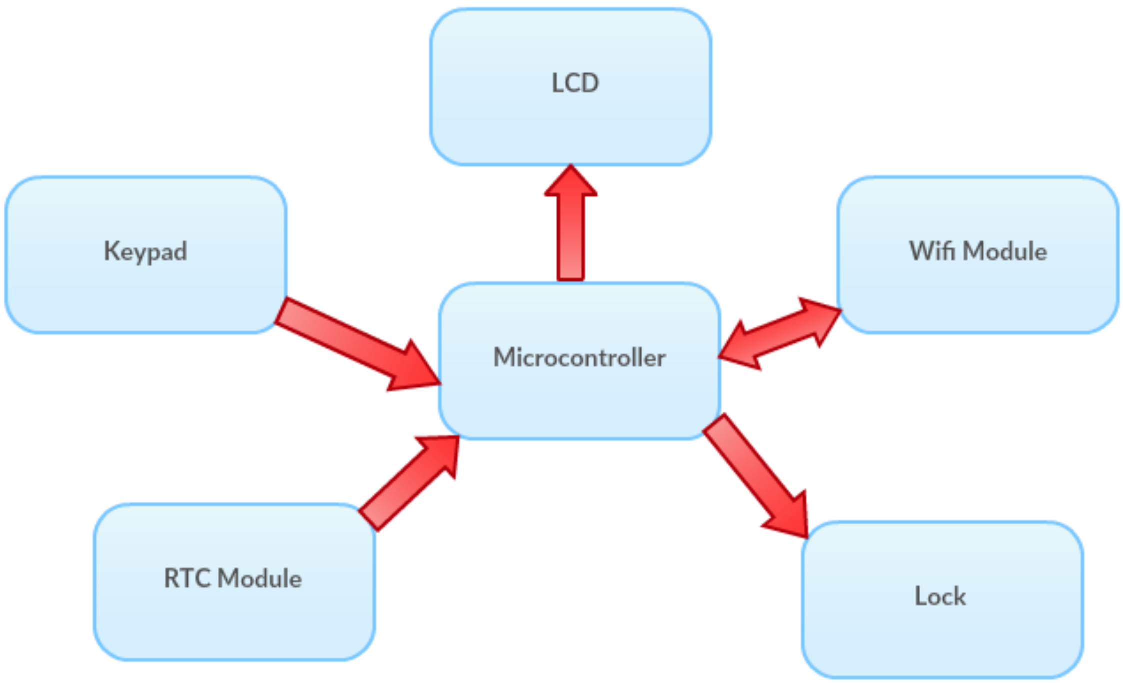

Figure 26 Embedded System Overview Block Diagram

Update: Due to the time constraints, the RTC module was removed from the embedded system.

5.5.3 Wi-Fi Module

The passwords for the bike rack are generated by the mobile application and stored in an online database. To access the password for a slot, the micro-controller needs access to the internet. The three main ways of connection to the internet are through the Arduino Ethernet Shield, The Arduino Wi-Fi shield and the ESP8266. The Arduino Ethernet Shield allows you to connect to the internet by plugging an Ethernet cable from the wall or router into MCU. Due to the outdoor nature of the bike rack this is not the most optimal solution. Both the Arduino Wi-Fi Shield and ESP8266 allow you to connect wirelessly. The major difference between the two is the price with the ESP8266 being significantly cheaper. For this project we chose the ESP8266. Figure 27 shows the chosen module.

Figure 27 Wi-Fi Module ESP8266

Update: The final design utilized the ESP8266 12E.

5.5.4 Real Time Clock Module

EzRack's distinguishing feature is its ability to allow users to reserve a slot in advance. Once reserved, the slot is rendered unusable until it is either claimed with the password provided to the reservee or a certain amount of time has passed without the reservee using the password to unlock the slot, in which time it iscome available for reservation through the app or use through the kiosk. To provide this feature, it is necessary to keep accurate time. The DS3231 is low-cost, accurate, and can keep track of hours, minutes, and seconds along with day, month and year. It also compensates for leap-years and months that are shorter than 31 days. The DS3231 runs off a separate battery and will keep track of time even if the microcontroller is disconnected from its power source. In addition to its features, the RTC module is relatively simple to set up and there are plenty of online tutorials on how to program it.

Update: Due to the time constraints, the RTC module was not implemented in the final design. Time is kept using a 16MHz dip quartz crystal oscillator.

5.5.5 Keypad

Bike slots is locked and unlocked by entering a 5 digit passcode into the keypad. Figure 28 shows the 3x4 membrane keypad that is used to input the necessary numbers.

Figure 28 4x4 Membrane Keypad

5.5.6 LCD

The embedded system will use the 2x16 LCD display pictured in Figure 29 to communicate with the user. The LCD pictured in Figure 29 and the keypad pictured in Figure 28 is combined into a single compact unit attached to EzRack and thus a smaller display is preferred over a larger one.

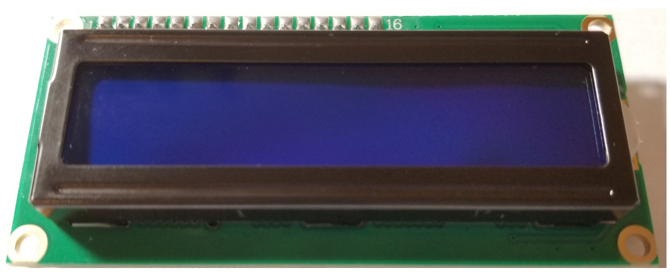
Figure 29 2x16 LCD

5.6 Embedded System Design

The embedded system is at the heart of EZRack and is one of the essential systems in transforming EZRack from an ordinary bike rack into a smart one. Not only is it responsible for locking and unlocking the bike slots, it must also be capable of communicating with a mobile application and accepting reservation requests. In order to ensure customer satisfaction, the embedded system must reliably lock/unlock the bike slots and be counted on to uphold reservations. To ensure smooth operation, the system is rigorously tested throughout each stage of its development. Each component of the embedded system is designed and tested separately. Only when the design of each component has been verified as working properly will they be assembled and tested jointly. Larger circuits are harder to debug than smaller ones and this approach ensures efficient debugging and eliminates excess time spent tracking down errors in the design. Development and testing is done using an Arduino like board and the Arduino

IDE. The surge in popularity of Arduino has led to a massive online community with plentiful documentation, libraries, forums and tutorials which is ideal considering the fact time is in short supply in the semester EZRack gets assembled.

5.6.1 Essential Parts

The embedded system is comprised of six major components, the microcontroller, LCD screen, keypad, Wi-Fi module, rtc module and the locking mechanism. Table 15 lists the major components in the embedded system along with their model and price. The microcontroller chosen for use in the embedded system is the AVR ATmega328P but development and testing is done using the UNO R3 Controller Board developed by Elegoo Inc. The UNO R3 Controller Board is a replica of the Arduino UNO R3 development board and is compatible with the same components as Arduino. In addition, the Elegoo's development board can be programed using Arduino IDE and is compatible with all its existing libraries.

Part Name	Model	Price
Microcontroller	AVR ATMega328P	$4.30
Keypad	4 x 4 Matrix Membrane Keypad	$7.56
LCD Screen	LCD1602 Module	$5.99
WIFI Module	ESP8266	$6.95
Voltage Regulator	LD1117	$5.70
Real Time Clock Module	DS3231	$6.99

Table 15 Embedded System Part List

Once the embedded system has been thoroughly tested on the Arduino board, and all components are functioning properly on the development board, the ATmega328P is removed from the development board using an IC removal tool and placed in the custom designed embedded system circuit.

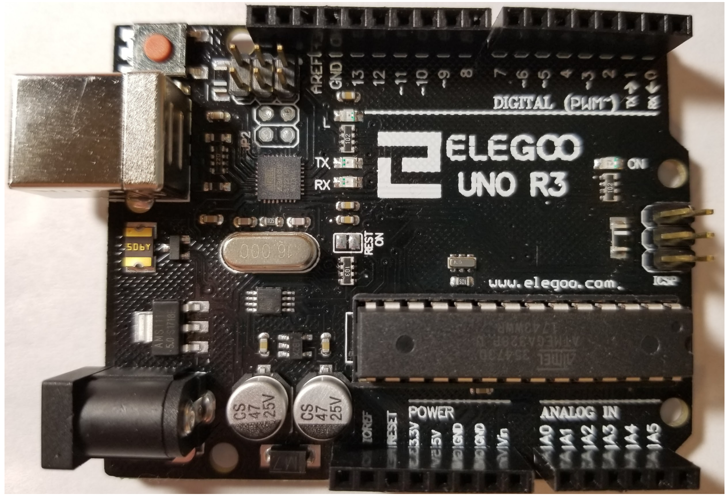

Figure 30 UNO R3 Development Board

Update: In the final design the ATMega328P has been changed to the ATMega328P-PU and the ESP8266 has been changed to the ESP8266 12E.

5.6.2 LCD Display Design

The LCD display used in the design of the embedded system is the LCD1602 module with 3.3V backlight. To improve the readability of the screen, the contrast between the text displayed on the LCD and its background can be adjusted by connecting a potentiometer to pin three of the module. The screen is two lines by sixteen characters with a capacity of displaying a total of thirty-two characters. The LCD module can be interfaced with a microcontroller in either eight bit or four bit mode. In eight bit mode, the module uses eight data lines, D0 to D7 and is faster than four bit mode. Four bit mode on the other hand uses four data lines, D4 to D7 instead of eight and thus saves pins on the microcontroller. Though slower, it doesn’t make much of a difference on the LCD display and is what is commonly used. Due to the limited number of available pins on the ATmega328P conversving pins is necessary in order to assure space for all the necessary modules. Like most text based LCD displays based on the Hitachi HD44780 or compatible chipsets, the LCD1602 module can make use of the Liquid Crystal Library. The Liquid Crystal Library makes it easy to display text to the screen by calling a host of premade functions.

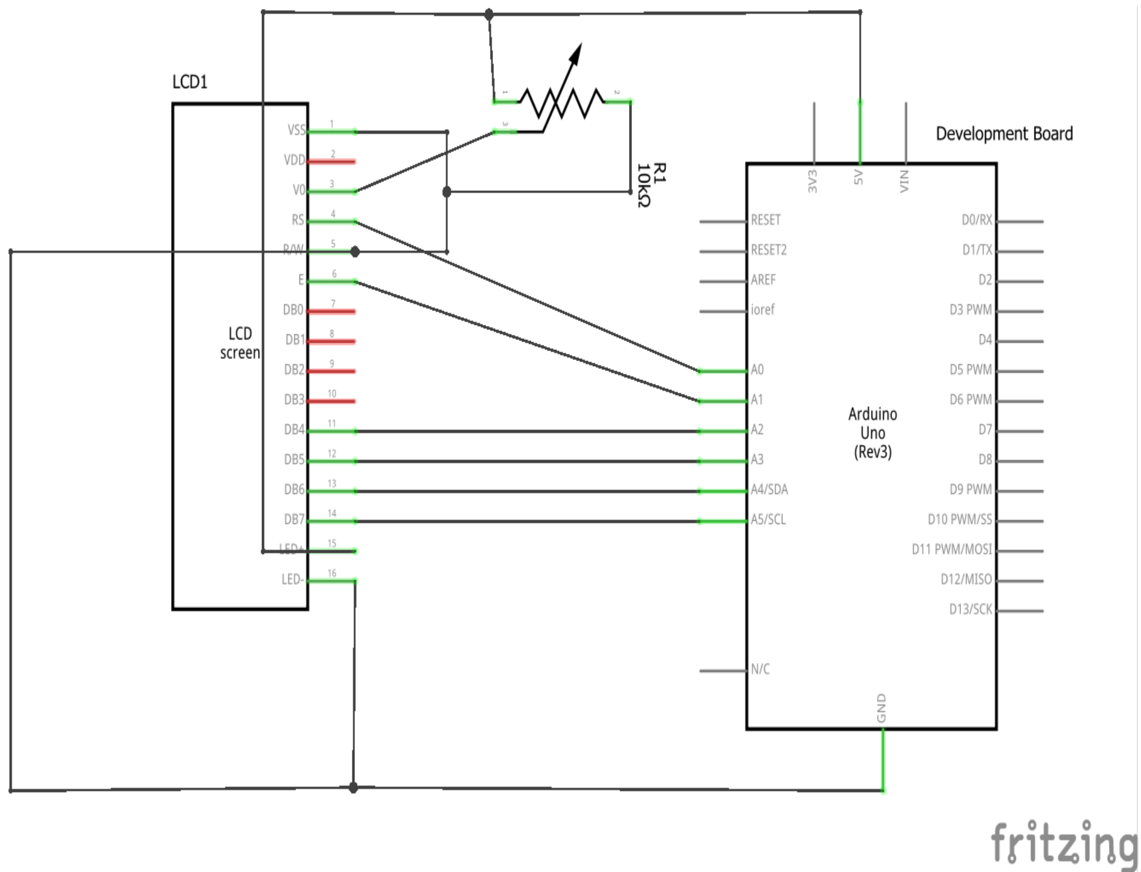

Figure 31 LCD/Arduino Schematic

5.6.3 Keypad Design

The keypad used in the design shown is a simple 4x4 membrane keypad. It has 8 connections as shown in Figure 32. The keypad is compatible with Arduino's Matrix Keypad Library which makes reading inputs easy by calling predefined functions. The library is non-blocking which means that keys can be pressed indefinitely and the microcontroller will continue to process the rest of the code.

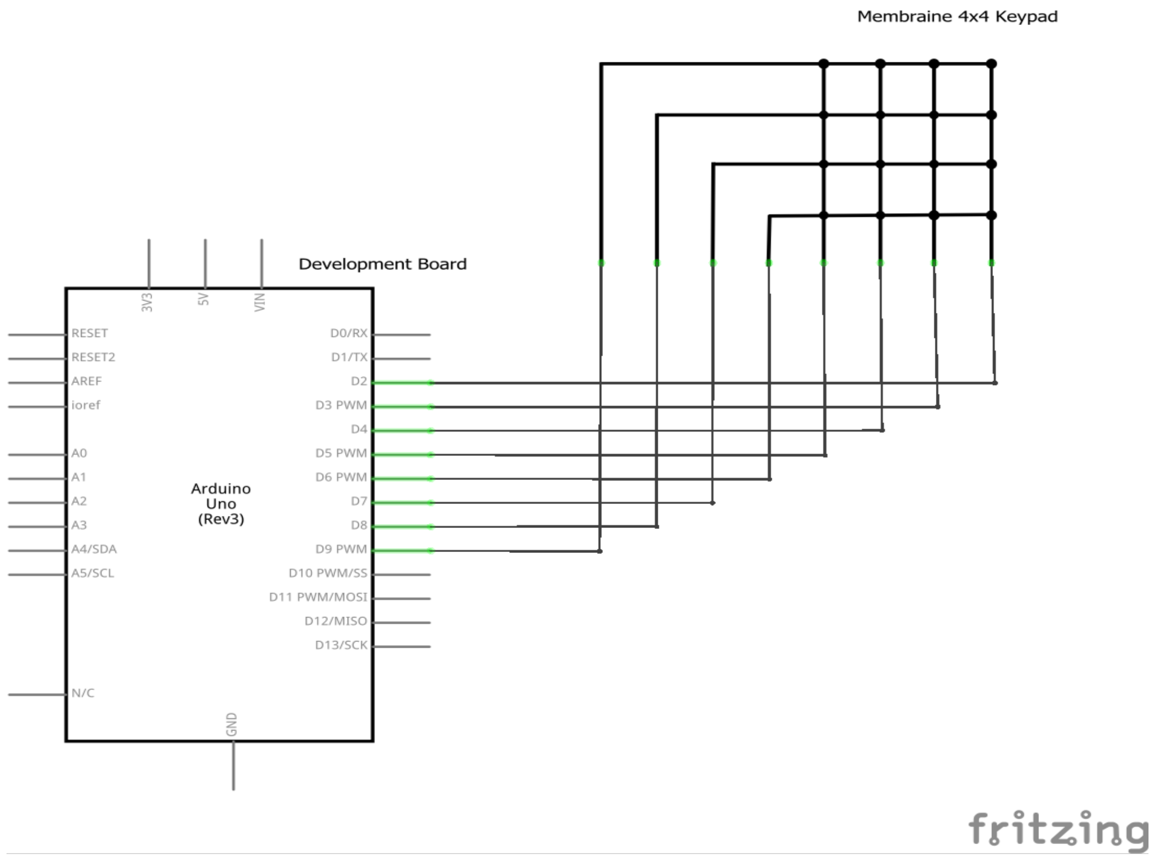

Figure 32 Keyboard/Arduino Schematic

Update: During development, it was discovered that the motor driver required two extra pins on the microcontroller. To accommodate this, the keypad was instead connected to the Wi-Fi module.

5.6.4 Wi-Fi Module Design

EZRack's main feature is the ability to reserve a bike slot in advance through the use of a mobile application. The reservation feature is dependent on the embedded system's ability to connect to the internet and maintain a constant and reliable connection. As previously discussed, the embedded system will rely on the ESP8266 Wi-Fi module to connect to the internet. Figure 31 demonstrates the connections between the Wi-Fi module and the development board. It is necessary to connect the RX pin on the Wi-Fi module to the TX pin on the development board. Due to the fact that the RX pin on the Wi-Fi module requires 3.3V while the TX pin on the board delivers 5V, they cannot be connected directly. Something must be placed between them to lower the voltage. There are three options for lowering the voltage output from the TX pin on the board into something the Wi-Fi module can accept-resistors, voltage regulator, or a level shifter. Of the three, a voltage regulator is better suited for providing a constant voltage to the circuit. Voltage regulators fall into one of two categories, linear and switching. Linear regulators are easy to use, simple, cheap, and are low noise.

Though they are normally inefficient, they are ideal for low powered devices and in situations where the difference between the input and output are very small such as in the case of the design shown in Figure 33. Shifting regulators, while much more efficient are more costly and noisy. The use of a shifting regulator is not necessary in this case. The embedded system will employ the LD1117 voltage regulator to lower the voltage.

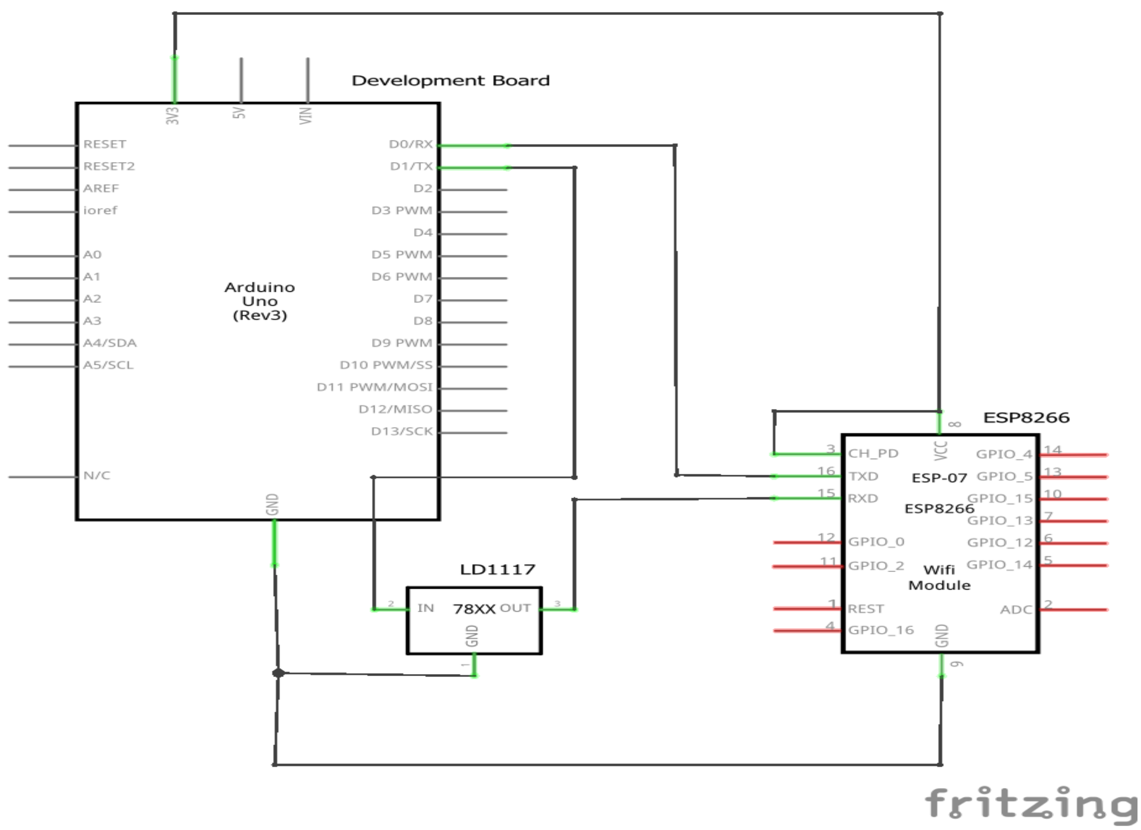

Figure 33 Wi-Fi Module/Arduino Schematic

5.6.5 RTC Module Design

The embedded system needs a way to keep track of time which is provided by the real time clock module. Figure 34 shows the connections between the real time clock module and the development board. Users may request a reservation for any open bike slot but the slot will not be on hold indefinitely. Users is given a window of time in which to claim their reservation. Reservations not claimed in that time frame is lost and become available again to other users to reserve through the mobile application or May claimed for use in person through the keypad at the EZRack kiosk. Having reservations expire ensures that bike slots are not rendered unusable due to open reservations that have not been claimed. Without this feature it is possible to eventually end up with an empty bike rack without a single available slot.

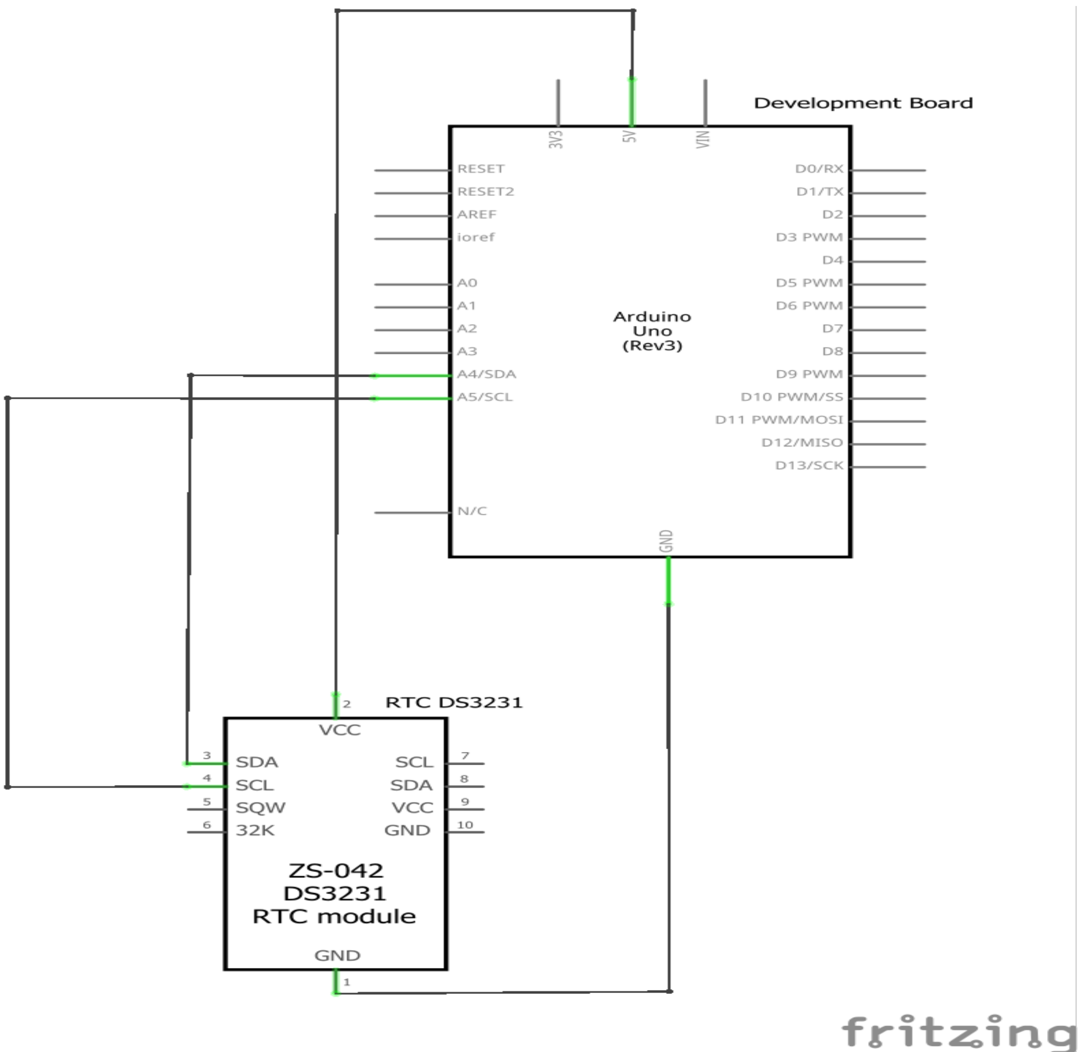

Figure 34 RTC Module/Arduino Schematic

The real time clock module used in the embedded system is the DS3231. The DS3231 was chosen for its accuracy in keeping time and for its ability to keep track of time even after being disconnected from its power source. When a user reserves a slot and is told they have X amount of time to claim their slot, their reservation should be held for that amount of time and not expire before the time has run out. In designing EZRack, our goal is to create a positive and stress free user experience. Accuracy is important since reservations expiring before the allotted time has passed will negatively impact the positive user experience we are aiming for. Once assembled, programmed and mounted onto EZRack, the embedded system will not be removed or reprogrammed except for cases of malfunction or damage. As is the case with all electronics, there may be times when they are disconnected from their power source for a length of time. With other time keeping methods, loss of power results in them having to be reprogrammed. Due to the outdoor nature of EZRack and the fact that any reprogramming of the systems requires the circuit to be taken apart, it is necessary to protect against cases where reprogramming of the system is necessary. The DS3231 incorporates a battery input and monitors the voltage to detect power failures and automatically switch to its backup supply when necessary.

5.6.6 Combined LCD and Keypad Design

Once both the LCD display and the keypad have been successfully tested individually, they is jointly connected to the development board as shown in Figure 35.

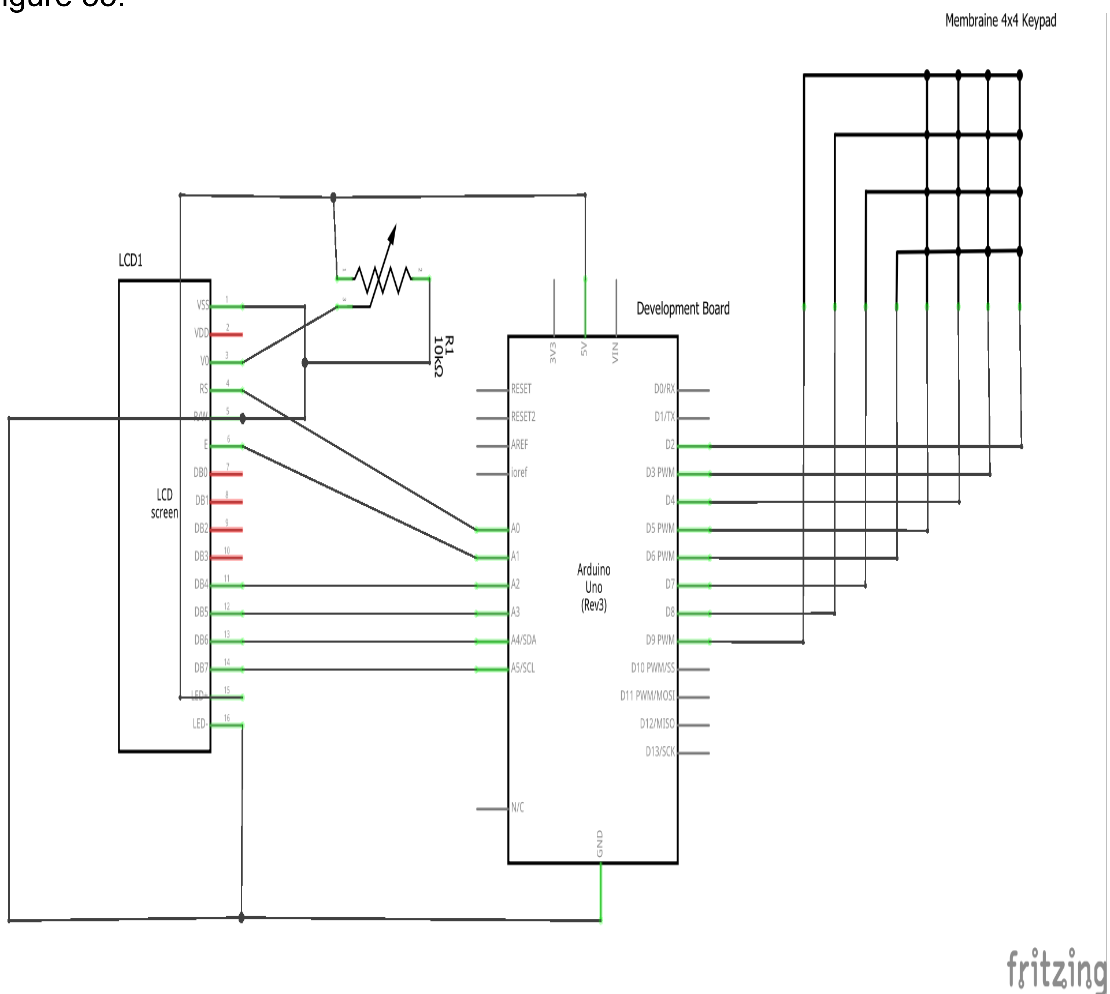

Figure 35 LCD/Keypad Schematic

5.6.7 Combined LCD, Keypad and Wi-Fi Module Design

Once the LCD display and the keypad have been found to function well together, the Wi-Fi Module is introduced to the circuit as shown in Figure 36.

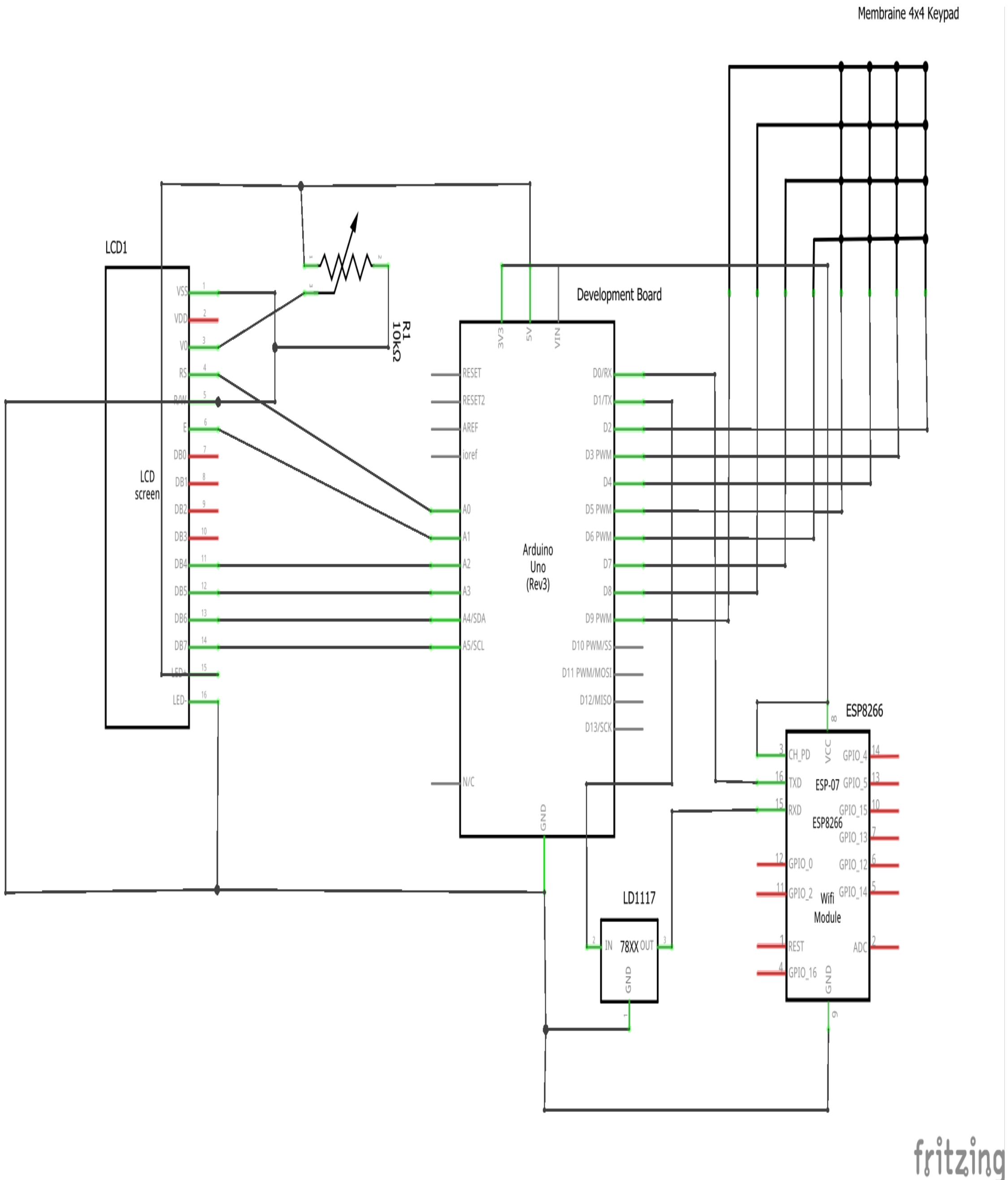

Figure 36 LCD/Keypad/Wi-Fi Module Schematic

5.6.8 Complete Design with Real Time Clock Module

Once all other modules have been added to the circuit and been tested to make sure they continue to function properly, the real time clock module is added to the design as pictured in Figure 37.

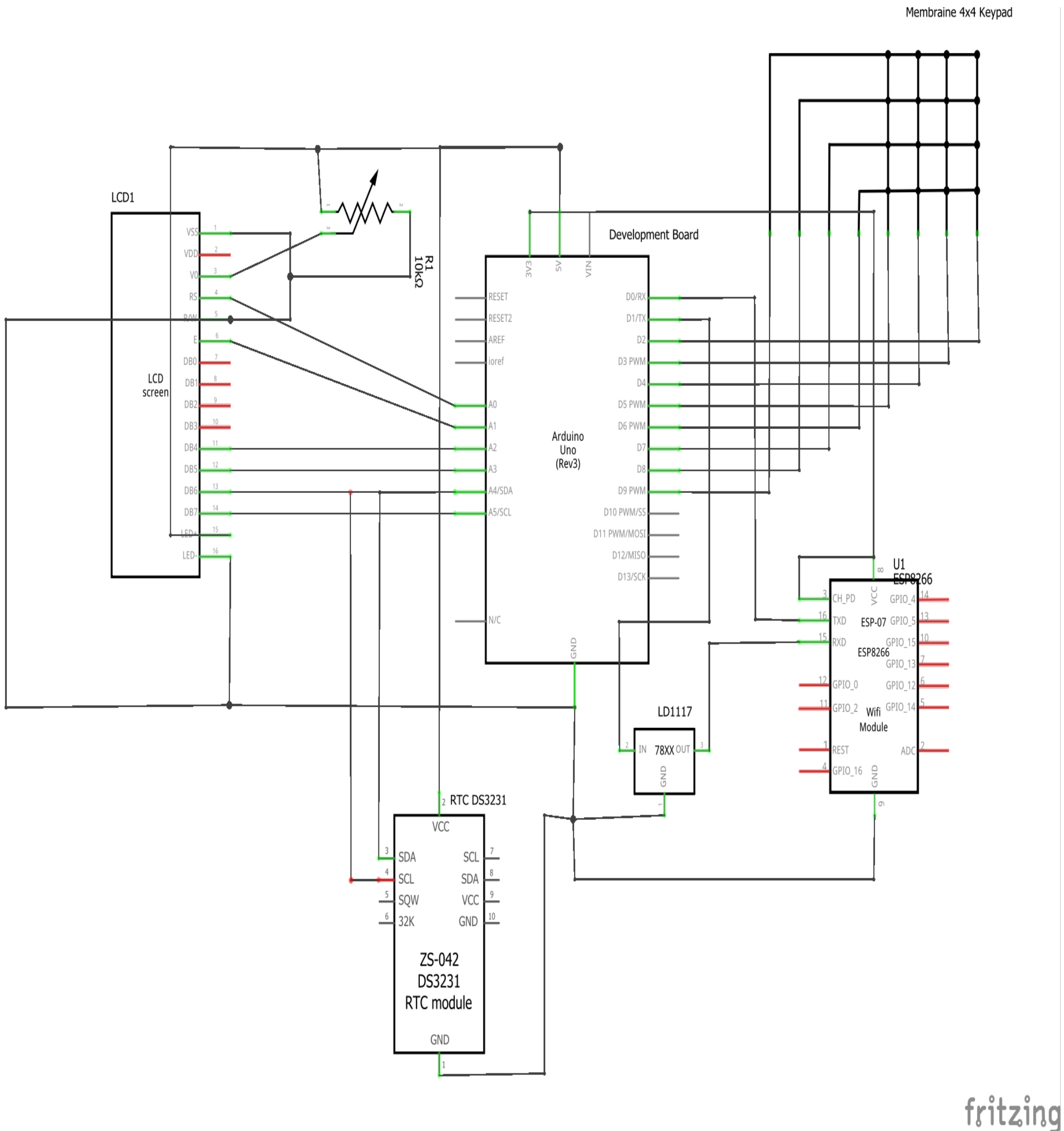

Figure 37 LCD/Keypad/Wi-Fi Module/RTC Schematic

5.6.9 Final Design with Locking Mechanism

After all the components have been added to the embedded system and properly tested, the locking mechanism is added to the embedded system design as shown in Figure 38. Figure 39 shows the final PCB design developed in EAGLE.

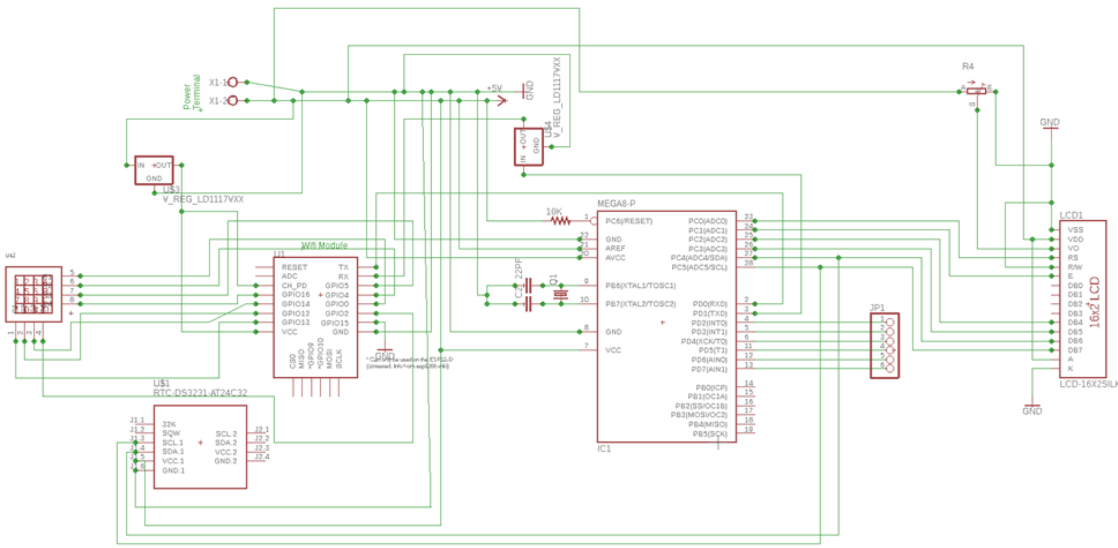

Figure 38 Final Design with Lock Mechanism

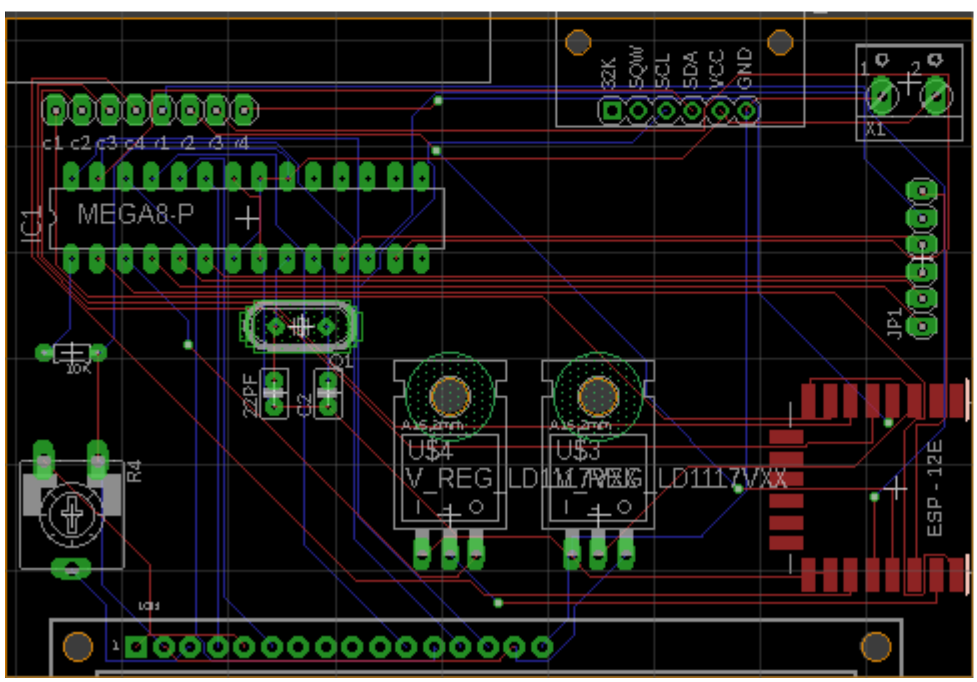

Figure 39 Embedded PCB

6 Software Design

This section will describe the software portion of this project. The software portion contains aspects such as the application interfaces, databases, and user interfaces.

6.1 Android Application Design

Due to the active lifestyle of individuals who utilize bicycles it is important that the EzRack design makes use of today's mobile platform. Users is able to use a free mobile application to make to make use of the functionality incorporated in the EzRack. This mobile application is available through the Google play store at no cost to the user. This application will allow users to take advantage of all of the features that the EzRack has to offer. The design of this mobile application is aimed to prove a very simple interface for the users. Ideally, the user is able to utilize any of the features with just a few clicks. Most of the computation is done in the back end the application even though the user will only interact with the front end. Having all of the computation done in the Android application eliminates much of the overhead that would otherwise go to the mobile application.

As far as the user interface is concerned, the structure of the application is fairly simple. The top activity is the main menu. This is where the user makes their initial decision. The user is able to view a detailed explanation of how to navigate the entire application. The user will also not have to log on to an account in order to use the EzRack. Removing a login option greatly reduces the amount of time it takes for the user to fully utilize the features of the application. This decision also makes the application friendlier to new users, since it removes the inconvenience of having to register and verify an account. The interface itself features options that are represented with big buttons in order to be clearly seen even when the user's focus is not directly aimed at their device.

As far as the android application backend is concerned, the algorithm is fairly simple. Based on what the user decides to do, the application will update the database accordingly. Regardless of what the user ultimately selects, the database will receive a six-digit integer from the application. The last five digits of that integer is the key that will is used to control the status of each slot. The first digit is the control signal that will tell the device what the user wants to do. When a bike slot reservation is requested and that request is granted, the application will then update the database with the necessary fields. When the user arrives at the EzRack and is ready to secure their bike in the slot assigned to them the app will send the same key that they have entered previously to the database. It is not part of the mobile application's algorithm to determine how the EzRack will handle each situation of the bike reservation process. The information is presented in that way, however, to make that mobile application as usable as possible. The system embedded to the device is ultimately responsible for the task of determining what action the user is taking based on the control variable that is part of the six-digit integer that is receives from the database.

Android Studio was used to create the EzRack Mobile application. Most of the source code on the entire application consists of either Java Files or xml files. Although the EzRack source code features C++ support, the application is primarily written in Java. Developing a Java application on Android Studio that is compatible with Android smartphones makes the development and testing a simpler process. Adding the additional C++ functionality was a choice was done in case it may be needed for further updates in the future. The entire application consists of nine different screens. A screen may feature various activities that modify the contents of it and change the layout. Each screen, however, will have its own behavior and may interact with other activities in other screen. Figure 40 displays a flowchart of all of the screens and how each screen interacts with another.

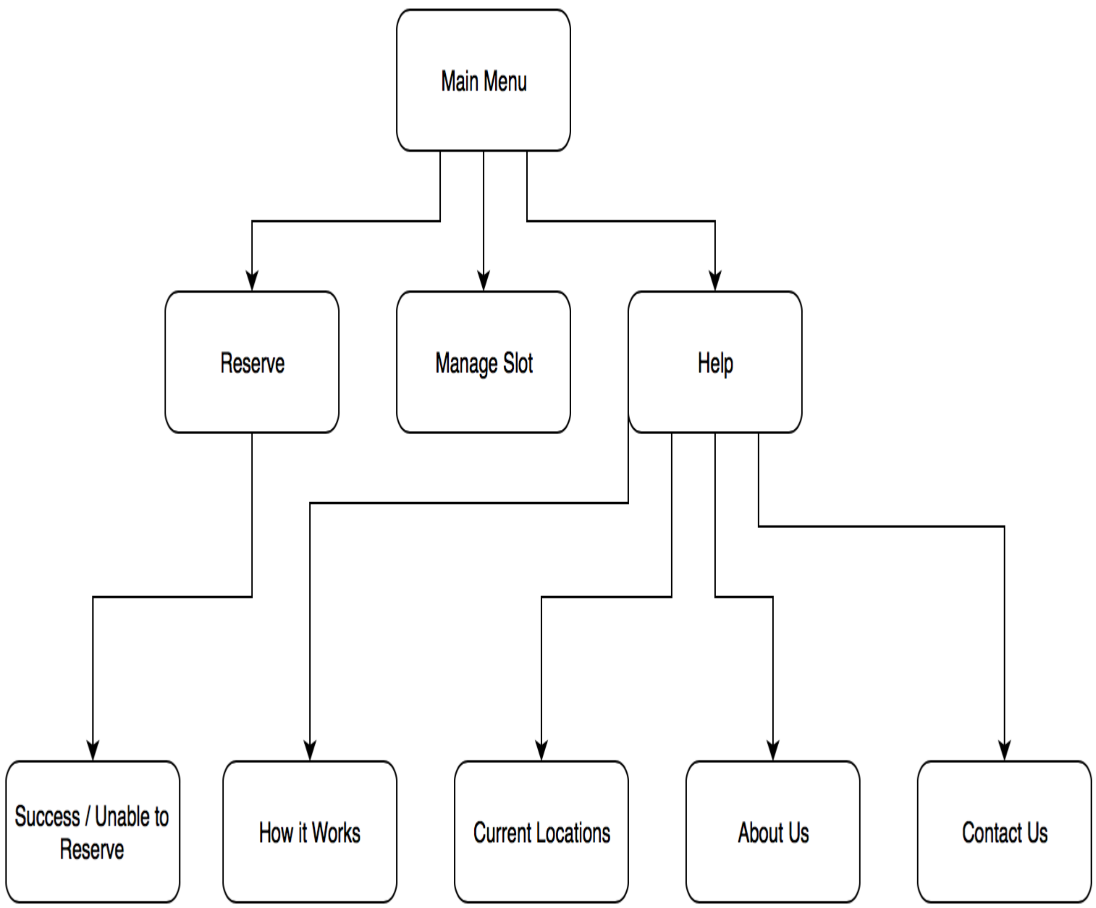

Figure 40 EzRack Flowchart

At any given point of the application the user can hit the Android back button to return to the previous screen. If the user selects the Android back button on the main menu screen, their will exit out of the application. When a user progresses through each screen, information will typically be passed into the following screen. Whenever the user selects the back button, none of the information is passed from the deepest screen. The mobile application does not store any user information at all. It also does not carry information from one session to another. Once the user closes out the application they should not expect to have any of their keys saved. This makes the application a lot more user friendly since there is no need to keep the application up and running while they are using any of the features available on the EzRack. Because of the shallow depth of the entire user interface, the user can simply close the application while not directly in use and quickly open it up and continue the process very quickly. Since the EzRack's main motivation revolves around convenience, it is important the mobile application is as simple and convenient as possible.

6.1.1 Main Menu

Once the user opens the application they is given three main options. Those options are:

1. **Check Availability and Reserve:** The user will select this option if they would like to reserve a slot on the EzRack. The user will able to set their own key to lock and unlock the slot but will not be given an option as to which slot they would like to use which is determined by the EzRack based on what's available.

2. **Manage Slot:** The user will select option when they have arrived at the EzRack or they would like to lock or unlock their bike from the corresponding slot.

3. **Help:** Selecting this option will bring the user to another menu screen that will let the user access various resources.

Each of the options on the main menu is displayed as large buttons so that they are easily visible when the user is on the go. Figure 41 shows the layout of the main menu screen as it appears when the user opens the application.

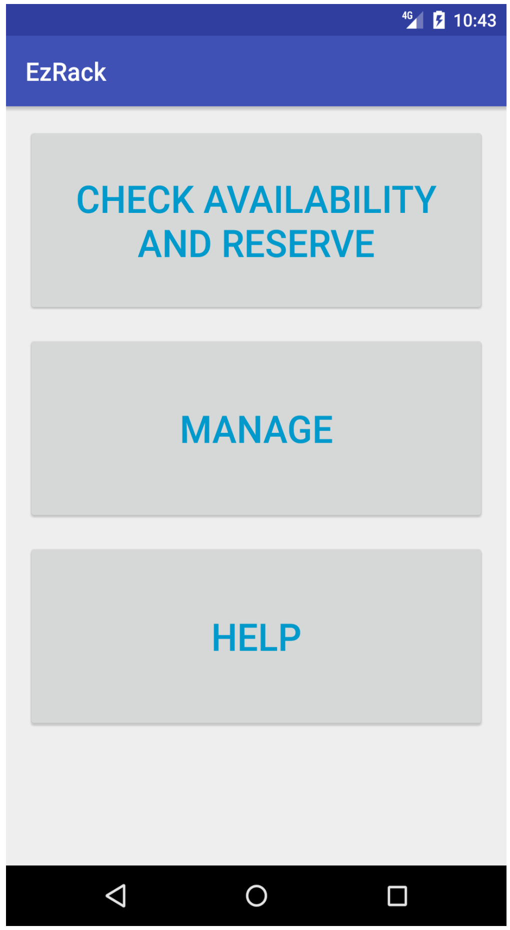

Figure 41 Main Menu Screen

The user is able to return to this screen from any other by using the Android back button that is incorporated into the phone's operating system.

6.1.2 Check Availability and Reserve

When the user selects this option from the main menu they is taken to a screen with some options on it. This screen will only ask the user for two pieces of information: location and desired key. The default location will appear first on the location selection box. When the user clicks on the modifiable location box it will expand to show all of the EzRack locations. When the user makes their selection, it will appear on the box. The user's selection will not remain in this box if they decide to exit from the screen and return to it. The location selection action is demonstrated in Figure 42 below.

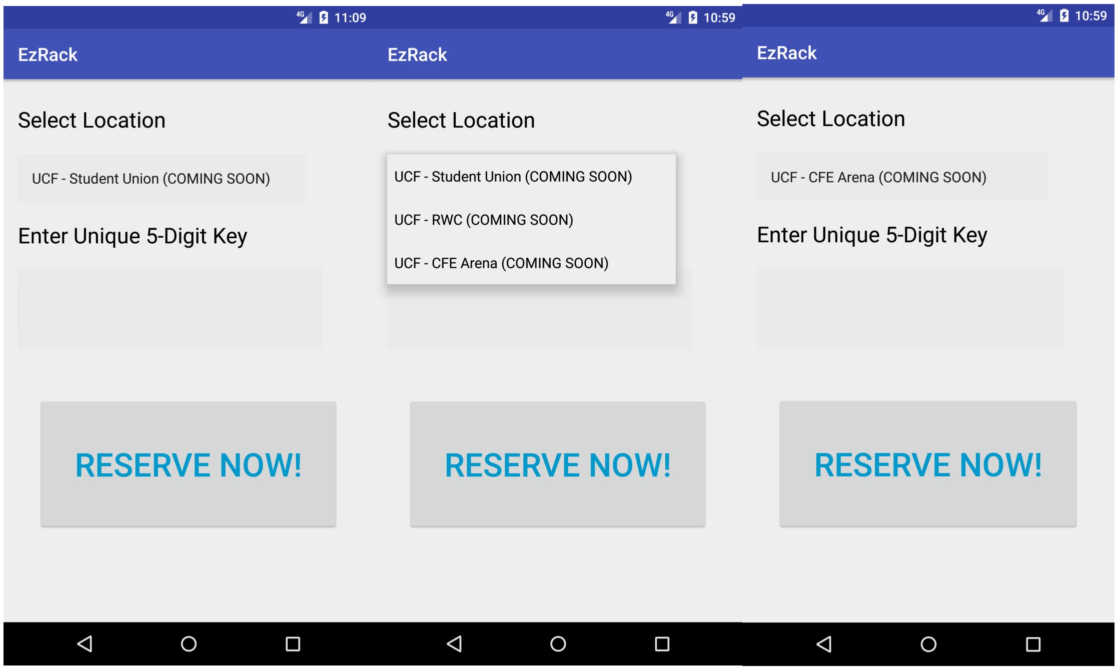

Figure 42 Selecting a Location

The user is able to select a location even if there are no slots available at the time. In this situation, the user is notified that the reservation will not be able to compete until a slot at the desired EzRack is available.

Once the user has selected a location they may enter the key they would like to use to lock and unlock this slot. It is the user's responsibility to create a key that is secure and that no one else would be able to easily guess. Once the user has populated the "Enter Unique 5-digit Key" entry, they will click on the "Reserve Now!" button to finalize their reservation. If the request has been successful it will notify the user in the next screen. If the attempt was unsuccessful, they user is notified why their reservation could not be completed. There are various reasons why a reservation may not be completed. Such reasons include unavailability of slots and invalid key.

6.1.3 Reservation Request

When a reservation is request is made the user is taken into a new screen that will tell them if the request was successful or not. Whenever the user makes a successful request it will there is some important information displayed on the screen. It is advised that the user screenshot the information or refrain from restarting the application until they fully understand the process of reserving a slot. The user is told which slot they have reserved since it is assigned by the application given the information it received from the availability. At this point, the bike rack will reserve a slot and hold for a maximum of ten minutes. If the user does not arrive and unlock their slot within the allotted time period they the slot is opened for use by other users, and the key associated with it is cleared. The user who has lost the reserved slot may still use the EzRack by either reserving a new slot or using any of the available slots when they arrive. Figures 43. Display the outcome of both events once the user makes a request to reserve a slot.

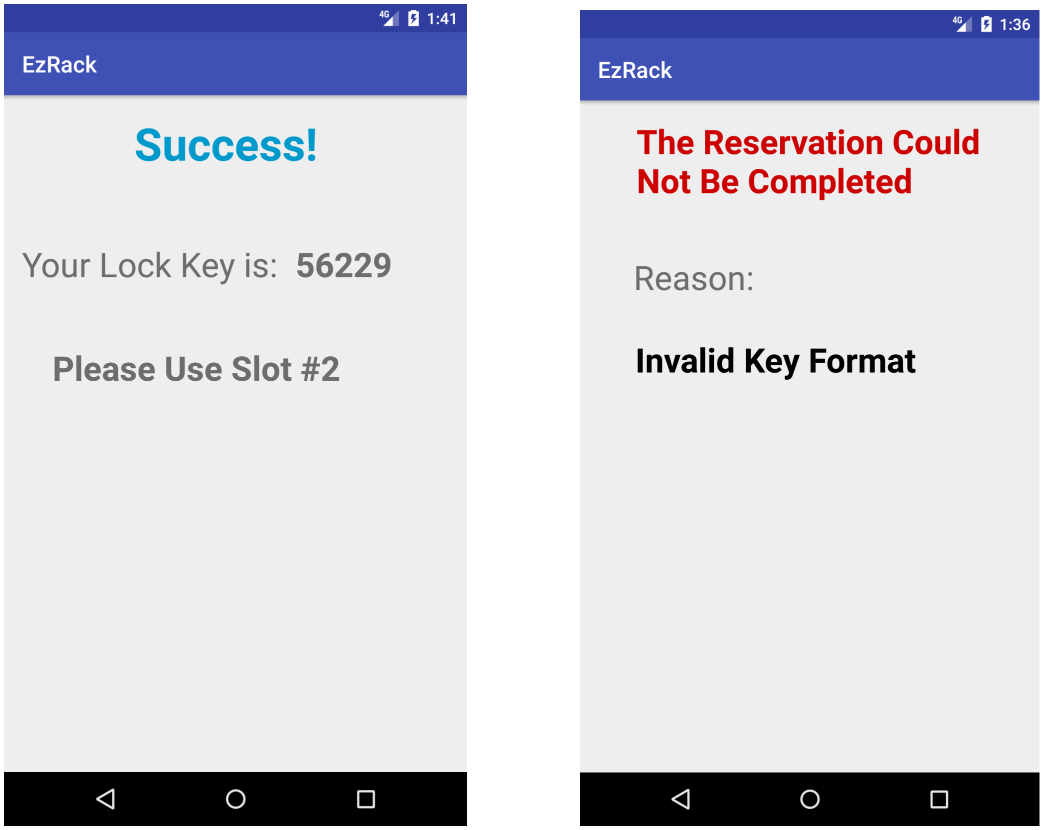

Figure 43 Reservation Outcome

If the user closes the application after the have reserved a spot it will not affect their reservation. The mobile application will not need to be running during the period of time that the user is traveling to the EzRack. The user will have to be responsible in remembering or documenting the lock key that they have entered to reserve the slot. The key that the user uses to request a reservation can also be entered into the physical keypad built into the EzRack. This allows to user to be able to use the EzRack in the event that their phone's battery has run out of power or if their phone is not readily available. The user will still need to use the same key to manage the slot even when they are no longer using their phone to communicate with the EzRack.

If the request could not be completed, the application will not reserve a spot for the user. There are a few reasons why a request would fail to complete, and most can be overcome by simply trying to reserve a slot again. The following errors could prevent a user from reserving a slot:

1. **All Slots Are Occupied:** If a user attempts to reserve a slot and all of the slots in that EzRack location are being used, they is given this error. The

user may choose to reserve a slot in another EzRack location if they wish or they may try again later.

2. **Invalid Key Format:** If the user enters a key that contains any character that is not a numeric digit, they is given this error. To complete the reservation, the user should re-enter a key that is in the correct format. This error can also be a result of not entering enough digits

3. **System Error:** This error is a result of a situation that the user no control of. This error can be the result of a bug in the software. Most of the time the user can overcome this error by simply restarting the application. If the error persists, they user should contact EzRack directly using the email provided in the “Contact Us” screen, which is accessible from the “Help” button on the main menu. This error only effects the reservation feature and should not affect the managing of the slots. The user will also be able to use the EzRack using the physical keypad and LCD on the system itself.

4. **Location Not Available/Closed:** If a location has not been officially opened yet the user may receive this error. The implementing party will also be able close a location if needed by contacting EzRack directly using the email provided in the “Contact Us” screen, which is accessible from the “Help” button on the main menu. This will prevent any reservations from being made for that particular location.

It is always advised that the user should try again in the case of an error. The user can simply use the back button on the Android operating system to return to the previous page. The user can always restart the application, since the interface in not very deep, this will still be a very quick option. The “Reservation not Completes” page will not automatically bring the user back to the previous screen after any period of time.

The timeout feature for reservations was not implemented due to timing constraints. Timing out a reservation after a certain amount of time has passed was dependent on having a real time clock as part of the embedded system. The RTC module was not implemented as part of the final embedded system design.

6.1.4 Manage Slot

The second option on the main menu is what the user will select when they would like to lock and unlock their bike slot using their phone while at the EzRack. There user will only be asked to enter the key that have created to use the EzRack. Figures 44 display the Managing Slot screen and how the “Enter Your Lock Key” is populated.

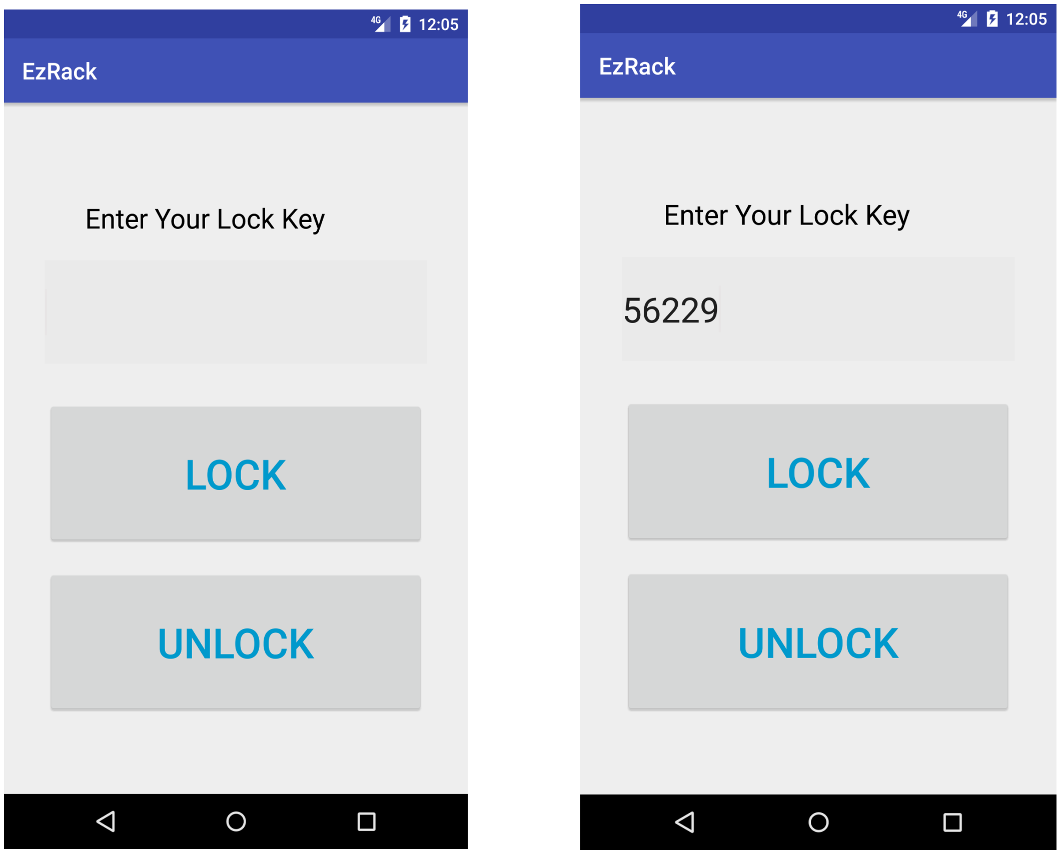

Figure 44 Managing Slot

Once the user has entered their key they may choose to either lock or unlock the bike slot. The user is notified on this screen whether or not their request was successful. If the user makes a request to lock a slot that is already locked, the application will notify the user of this situation. Likewise, if the user makes a request to unlock a slot that is already unlocked, this information is displayed at the bottom. Any time a user enters a key that does not belong to any particular slot, a “Key not found” error will appear at the bottom of the screen. Information related to each error and their causes can be found in the “How It Works” option in the “Help” screen. Any further questions on the process after reviewing “How It Works” can be sent to the email provided by the “Contact Us” option, which is also located in the “Help” menu. When a successful lock/unlock request is made the user is notified of the success and the EzRack ishave accordingly.

Like the reservation process, the user is notified whether or not their action was successful. Unlike the reservation process, however, the messages is displayed directly on the current screen. This will eliminate the need to open a new screen and have the user navigate back to the previous screen. This action was necessary for the slot reservations since there are more instructions that must be displayed to any new users. Whenever a request is successful, a green message will appear notifying the user that the slot has either been locked or unlocked.

Figure 45 shows the new appearance of the "Managing Lock" screen when the user successfully performs the any of the actions.

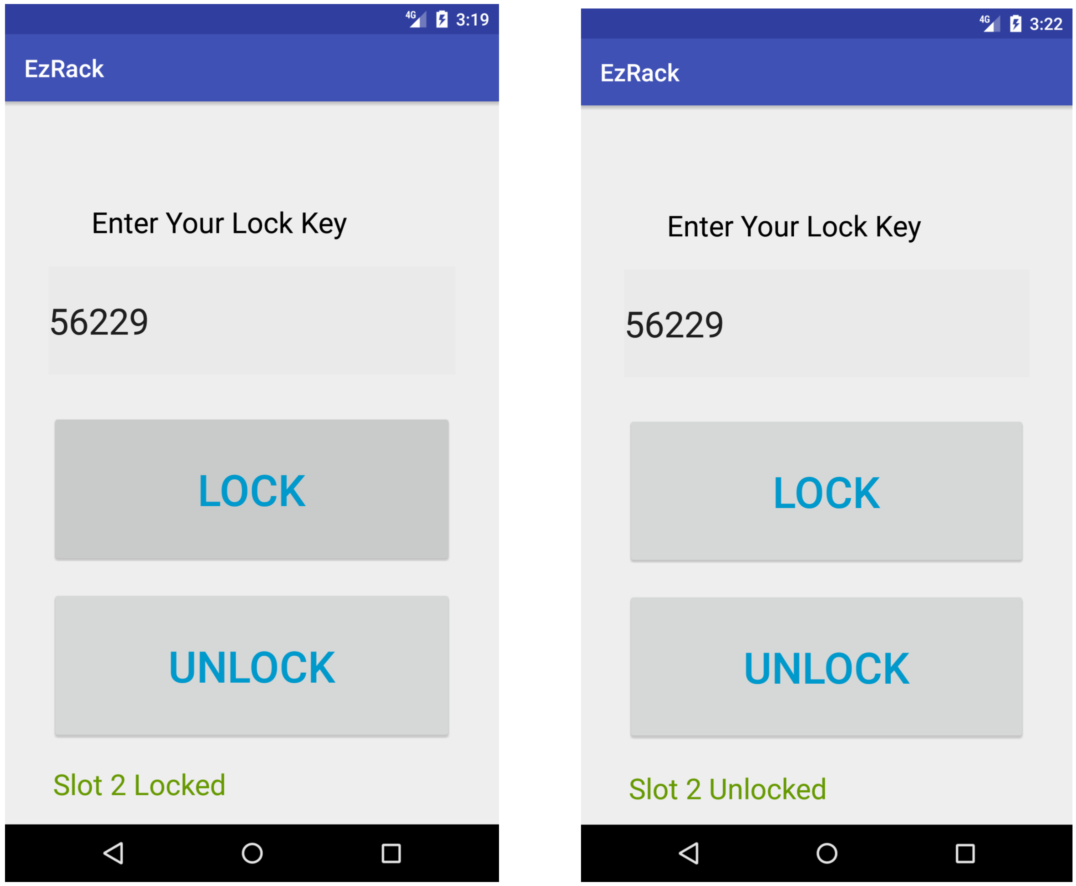

Figure 45 Successful Lock and Unlock

As far as unlocking is concerned, as long as the user is unlocking a slot that is currently being reserved by them, or a slot that is being occupied by their bike, they is is given the green message. Likewise, for locking, as long as the user is locking a slot that they have reserved and is in the process of securing their bike, they will receive the green locked message. This message will continue to appear on the page until the user decides to either take a new action, or close out the application. The message will not clear after a set time has passed. This will allow the user to know what stage they are in while using EzRack to sure their bike. Once the user exits the application, or returns to the main menu using the back button on the Android operating system, the screen will appear in its normal state when they return. This screen is used to lock and unlock slots only, and should not be used to try to make a reservation.

Similar to the success messages prompted for each successful action, any unsuccessful attempts will result an error message. If the user enters a key that is not associated with any of the slots in any of the EzRack bike racks, they user will receive a red message that states "Key Not Found". Similarly, if the user enters a key that is incomplete, or a key that isn't in the proper format, they is prompted with an “Invalid Key Format” error. Figure 46 shows how the “Managing Slot” screen changes as a result of either of the errors.

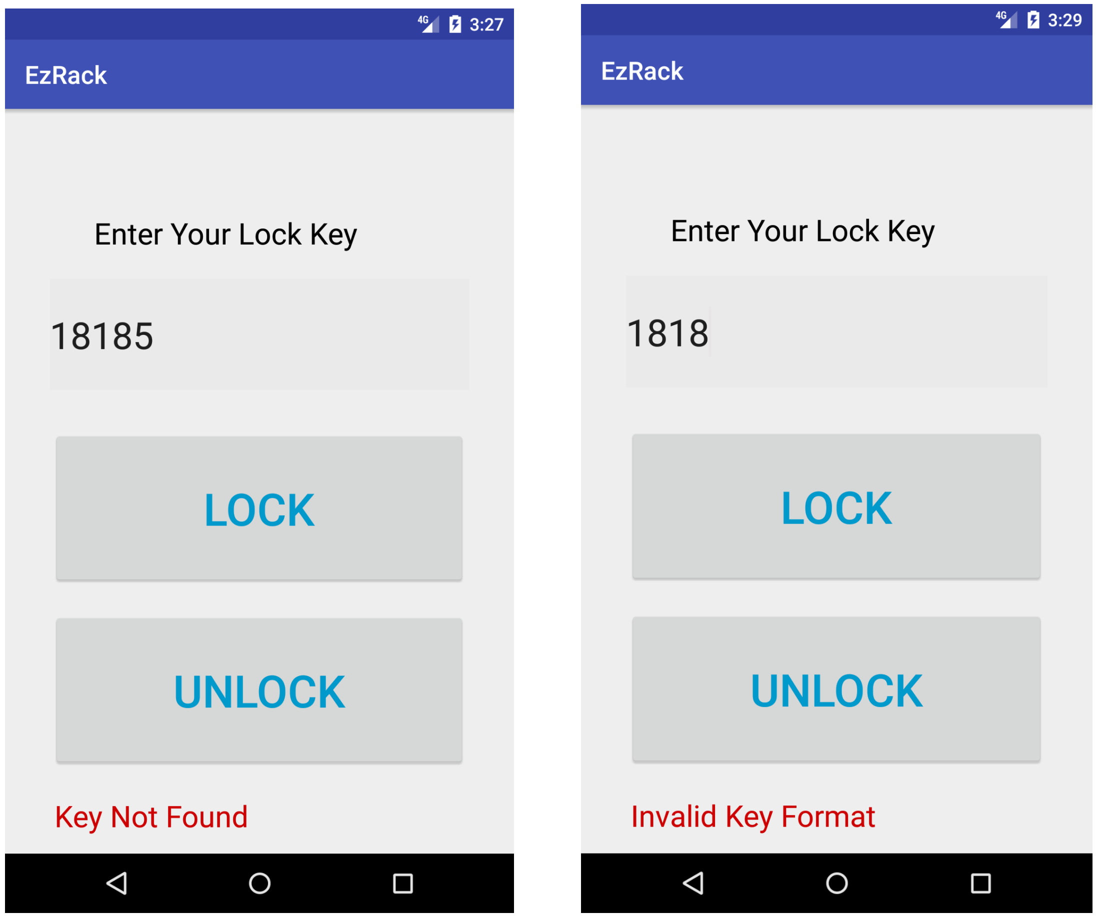

Figure 46 Locking and Unlocking errors

If the key is not found on the database, the user may try to enter the key again on the same screen. The user will not be required to exit the screen and return to it in order to proceed. An “Invalid Key Format” error is typically overcome by simply correcting the key that was originally entered in the box towards the top of the screen. just like the appearances of the success messages, the error messages will remain on the screen until the user enters a new key in the box, or the user enters a key, or exits out of the “Managing slots screen”. If the user is attempted to perform an action that has already been done, such as locking a bike slot that has already been locked by them previously, they is prompted with a message that will state this. The user can even visually verify that the action has already been completed. Figure 47 shows how the appearance of the “Managing Slots” screen changes when the user is performing an action that has already been done.

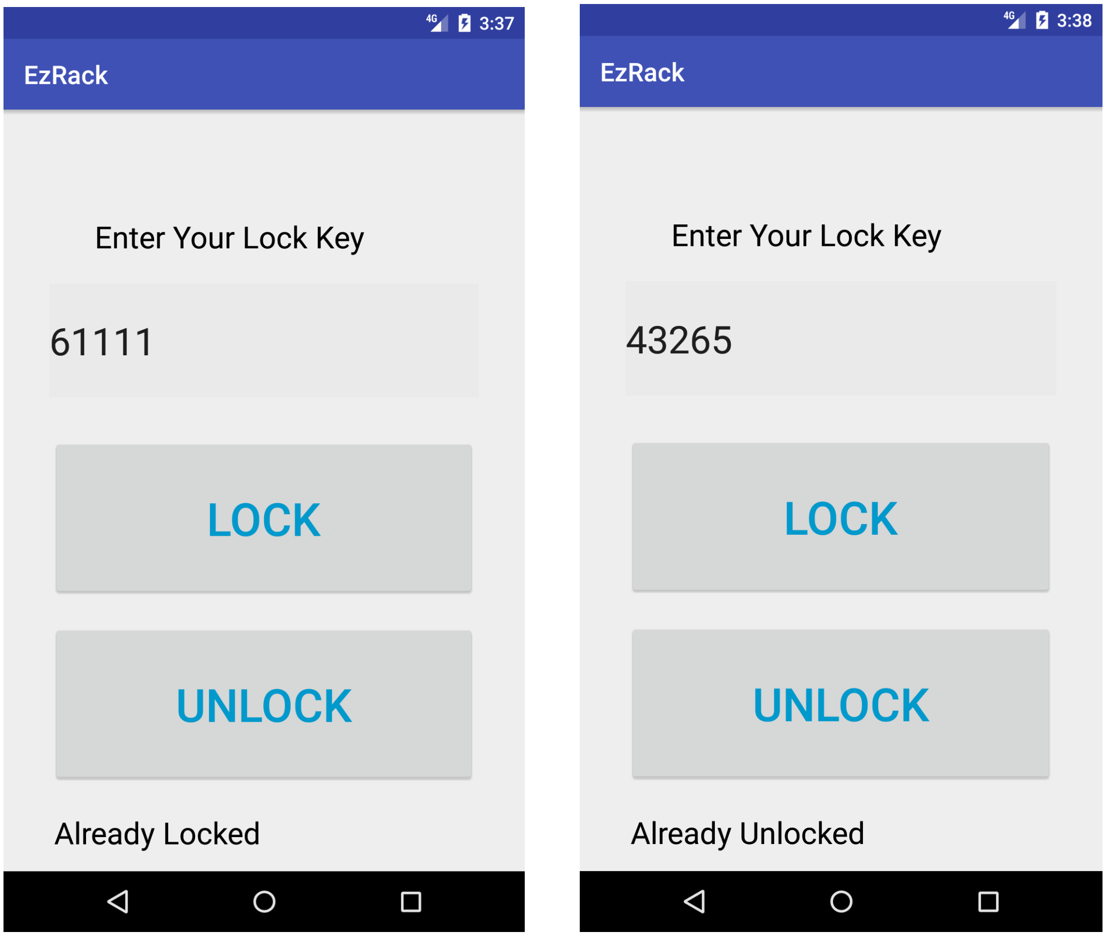

Figure 47 Redundant Operations

This action will not affect whether the bike is actually locked in place and is used for the information purposes only.

It is important to understand the user will only be able to unlock the slot twice and lock the slot once from this screen. Once the user arrives at the EzRack, they will place their bike in the slot that was assigned to them. The reserved slot will then be unlocked once the user selects the unlock option on this page. After verifying that the bike has been placed in the correct slot in the proper way they may use the “Lock” button on this screen to secure the bike. The final operation that the user is allowed to perform is an additional “Unlock”, which will then free the slot and clear the key that the user had been using to occupy the space. At this point, any other user may be able to use that slot, on a first come first serve basis. This process is important in preventing an individual from indefinitely holding a slot that is unusable by others.

6.1.5 Help

The last option the user can select from the main menu is the “Help” option. Clicking on this will take the user to a new screen with further options. This screen will direct the user to any other resources available in the EzRack mobile application that isn’t directly reserving a slot or managing that slot that they have reserved. As the software continuously gets updated, user should expect to see all new and relevant information added the resources on this screen. The following options can be selected from the “Help” screen:

1. **How It Works:** Selecting this option will reveal a plethora of information related to all of the functionalities of the mobile application. When a user is having issues with anything related to the EzRack, it is recommended that they first select this option to receive any information that may help them. This selection is also broken up into multiple options but all of the sections will remain on the same screen. The user may scroll down to view all of the relevant information

2. **Current Locations:** Selecting this option will open a new screen that will display a list of all of the EzRack locations. It each item on the list also features a brief description on its exact location. Like the previous option, this screen will have a scroll bar in the event that the number of EzRack locations grow past a size that is not representable be a single screen

3. **About Us:** Selecting this option will open a new screen that will display general information about the EzRack and its purpose. This will also show information about the creation of the EzRack and its anticipated future.

4. **Contact Us:** Selecting this option will open a new screen with contact information that the user may use for any EzRack issues that have not yet been resolved.

Like the main menu, each button on this screen is large and easily selectable by the user. There is also a limited amount of selection on this screen which will allow the user to find what’s relevant to their needs as quickly as possible. Information on all of the “Help” selections will constantly be updated so it is important that the users check up on this option to keep up to date with any relevant information.

6.1.6 How It Works

This option that can be selected from the "Help" screen is the user's best bet when it comes to finding any useful information directly related to utilizing all of the features present on the EzRack. The users should expect to find the following pieces of information on this page:

1.) **Brief Tutorial:** The user is able to view a brief tutorial on how to use the EzRack mobile application to reserve and manage a bike slot. This step by step guide will simplify the process in order to keep the length of the page as short as possible. This tutorial will also cover the basics of using the EzRack without reserving a spot in advance.

2.) **EzRack Walkthrough:** This section will simply cover the basics of how the EzRack operates both on the physical side and the software side. The goal is to be as transparent as possible in order to help the user gain a firm understanding of the system that they are using to secure their bike.

3.) **Troubleshooting Tips:** This is where the user can view a number of solutions to various potential issues. All of the common potential issues is listed

4.) **EzRack User Etiquette:** The user is able to view a list of common tips to make their biking experience more secure and enjoyable. The purpose of this section is to remind the users of common mistakes and how to avoid them. All of the information is presented in a humorous tone.

As mentioned earlier, if a user is seeking information and cannot find it on the screen, they are encouraged to contact EzRack directly using the email provided in the "Contact Us" page. All of the information accessible by this screen is dynamic. There may be changes to any of the sections mentioned above so it is advised that the user check back to this section for any updates.

6.1.7 Current Locations

As the EzRack develops, the amount of locations that are implementing it should increase. Selecting this option from the "Help" screen will display all of the locations of the EzRack. Included with the list of locations is a brief set of details for each one. Including those brief details is important in order to ensure that the users will easily find it. This is a dynamic list, and it is updated the moment any changes I the number of EzRack present becomes detected. Some locations on this list may not be operating per the request of the respected implementing parties. These locations will still appear here, however, if a user ties to select a closed location they will receive a notification informing them of the situation. Most of the time, any closed or not operating EzRack will have some kind of indicator next to the name. This indicator should prevent anyone from trying to reserve a slot in a location that is not available.

6.1.8 About Us

If the user is interested, they may find more information on the motivation and creation of the EzRack by selecting this option from the previous screen. This will display tons of information on the events leading up to the creation of the project. This will also feature a timeline of events and a brief summary of where the EzRack will expect to be in the future. Hopefully, the EzRack will motivate other similar projects and continue the wave of creativity and innovation. Expected updates to the mobile application is shown here with all of the information related to the releases. Any future projects or improvements is discussed in this screen

6.1.9 Contact Us

Selecting this option form the “Help” menu will reveal some more resources that may be very useful to the user. This is where the user is suggested to go in the situation that they have exhausted all the other resources on the application including the “How It Works” page. At the top of the page the user will find an email address that is associated to the EzRack. Any questions and concerns should be sent directly to the address so that the EzRack team can work to find a solution and improve the overall experience. It is important that the users express any concerns that thy might have in order for the EzRack to continue to grow and develop as it was intended.

6.2 Database Connectivity

This section will outline the connectivity between firebase realtime database system and our EzRack mobile application and the microcontroller on the rack. FireBase realtime database by google is a NoSQL cloud database. Data are stored as JSON and will synchronized in real time for every client that is connected every time the data changes. Firebase realtime database uses JSON tree as its data structure. When adding data in JSON tree, it then becomes a node in the tree with an associated key. Though Firebase realtime database allow up to 32 levels deep of nesting, the best practice is to avoid nesting data. Once the data is fetched, the retrieved data contains all of the child’s node. Another reason to keep the data as flat as possible is permission and security issues. Once a user has read or write access to a node, the user have access to all of data under that node.

To integrate Firebase realtime database to EzRack mobile application, there is only five steps need to be taken.

Implementation path:

1. Integrate the Firebase Realtime Database SDKs
2. Create Realtime Database References
3. Set Data and Listen for Changes
4. Enable Offline Persistence
5. Secure your data - Utilize security rules to secure data

Firebase will also be linked with the microcontroller through ESP8266 Wi-Fi module so that the information from the mobile application can communicate with the microcontroller through the database. This is shown in Figure 48 below. This is one of our methods to control the microcontroller. Arduino IDE is used in conjunction with android studios due to the nature of this project. Arduino IDE will need to include some library regarding Firebase for Arduino.

Steps:

1. Link database information to Arduino IDE
   a. All database secrets can be found on Firebase console
   b. Download all firebase libraries and include into IDE
   c. Include ESP8266 library
2. Preferable to have mobile application user interface and backend database structure established.
   a. Use Firebase assistance to connect to real time database

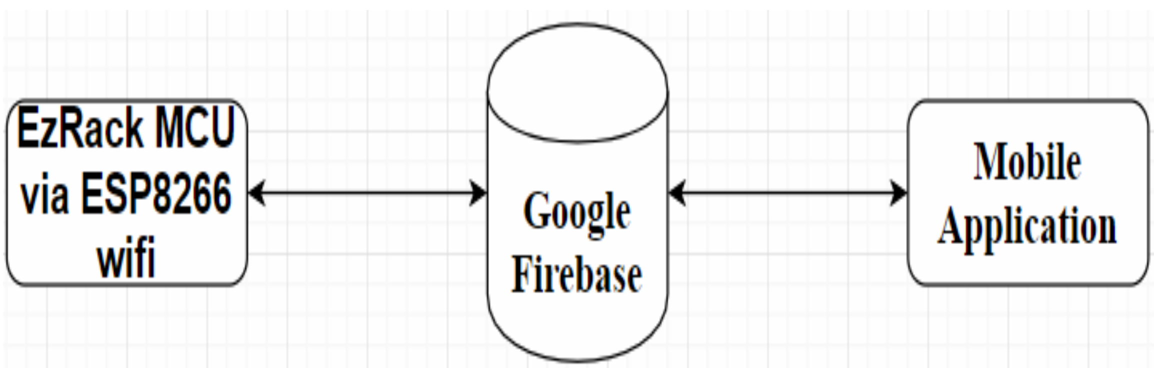

Figure 48 Connectivity of system

6.3 Embedded Software Design

At the heart of the EzRack is an ATmega328P microcontroller with software embedded onto it. This embedded system is developed using the C programming language. The embedded software will provide a means of communication between the microcontroller and the various components of the EzRack. These components include the numeric keypad, ESP8266 Wi-Fi Module, Real Time Clock Module, Locking Mechanism and the LCD display. The embedded software will also be tasked with keeping track of what slots are being occupied at each stage, which slots are reserved or unavailable and with locking and unlocking the slots..

The system designed to keep track which slots are reserved and currently in use is quite simple. An integer array is used to determine whether or not a space is occupied, where the index of the array represents the slot number and the value stored at each index is a unique six-digit integer. There will also be three separate arrays that represent stages of the slot, these stages include:

1) Reservation stage: This stage begins when the mobile application generates a unique key that is displayed to the user. This key will also be received by the EzRack system and placed into the master array in the index that corresponds to the slot that is being reserved. At this point, no one else is able to reserve, lock, or unlock this particular slot. The key is also placed into a reservation array that is used to tell the system that the slot is being held and not actually occupied by a bike. Elements in this array is cleared from their index in the case that the allotted time for a reservation has passed. When an element from this array is cleared due to timeout, the value stored in the master array is also cleared. Once the system gets an input from either the web application or the keypad, and it detects that this input in indeed part of the both the master array and the reservation array, the value is cleared from the reservation array and placed into the arrival array.

2) Arrival stage: At this stage, the user has arrived at the EzRack and it is ready for them to place the bike in the reserved slot. The unique key that the user was given is stored in the arrival array. To ensure that that the slot doesn’t stay unusable in the event that a user is in this stage but decides not to use the slot, the values stored in this array are cleared after a set amount of time has passed. Once the user places their bike in the specified slot and signals the locking action, the key stored at the corresponding index in this array is moved to the occupied array.

3) Occupied stage: This stage represents the time a bike is being held in the slot for the user. This stage has no time limit associated with it. Values inside of the occupied array will remain in the master array until the user enters the unique code on the keypad or mobile application. Once the code is entered, and the program detects that the value is indeed stored in the array, the unique code is cleared. The corresponding slot then becomes available, and the unique code will no longer be usable.

When a slot is left unoccupied, the value in the corresponding index is -1. The mobile application requests to check which index of the array (if any) contains -1 in order to determine slot availability.

Slot/Array Index	Unique Code
1	881239
2	-1
3	-1
4	171788
5	783219
6	323331
7	-1
8	912839

Table 16 Master Array

Timers is used to determine whether or not the allotted times have been surpassed. These timers is coded as part of the microcontroller and not be implemented externally.

The concept of stages was used in the final embedded system software design although the separate arrays were removed. The embedded system constantly polls the database to check for any changes in the state of the slots. To first number of the six-digit number stored in the passcode signifies the state the slot is in – 1 for reservation, 2 for unlocked reserved slot, 3 for a locked slot, and 0 for an empty slot.

6.4 Kiosk User Interface

The software embedded into the EzRack will receive input from only two sources. One of the inputs is a seven-digit code from the dedicated web application. That seven-digit code consists of the unique key and the opcode. The second input is received through a keypad that is attached to the bike rack itself.

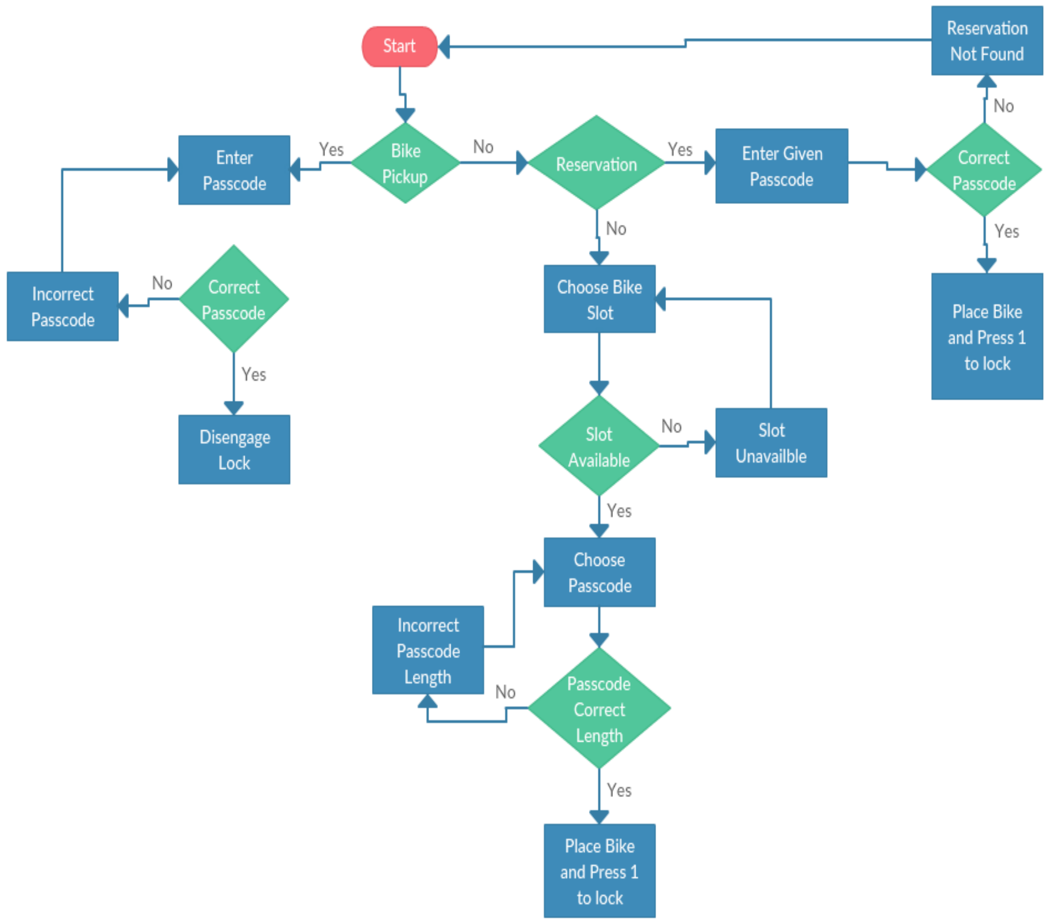

Figure 49 Kiosk User Interface

The attached keypad is mainly used for bikers who would like to utilize the EzRack without using the mobile application. Figure 49 demonstrates how the user interface on the kiosk will function. The keypad is accompanied by an LCD display that is placed right above it. The LCD is used to display simple instructions to the user. The interface in the bike rack can also be used to unlock a slot even if the user used the mobile application to reserve and lock a slot initially. When a user walks up to the keypad, it will ask the user if they are picking up a bike as shown in Figure 50. If the user is picking up a bike, it will then ask for a passcode and verify its validity.

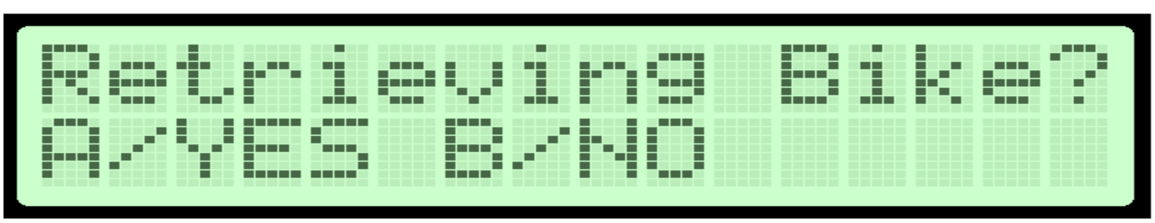

Figure 50 Initial Screen

If the user is not picking up a bike, they will then be asked if they have made a reservation for a slot as pictured in Figure 51. If a reservation has been made, the system will check the master array to determine if the reservation is valid. If the reservation is still active, the user is prompted by the LCD to place their bike into the slot and press 1 to secure it. However, if no reservation is found, the LCD will notify the user by displaying the message shown in Figure 52.

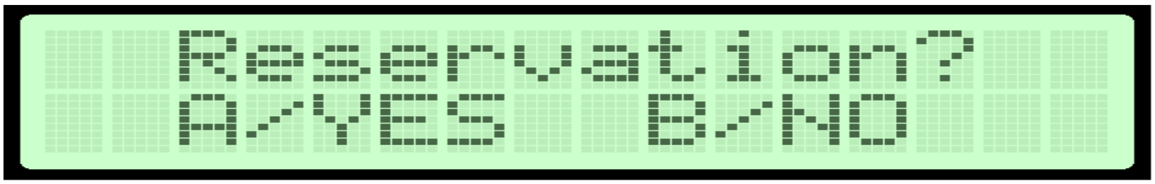

Figure 51 Asking If User Has Made a Reservation

If no reservation has been made, the user is prompted to enter the slot number that they wish to use. When they enter a slot number, the system will check the master array to determine whether the slot is in use or not. If available the user is prompted to place their bike into the slot and press 1 to secure it. If not, the user is notified by the LCD as shown in Figure 52 and is again prompted to enter a new bike slot. A slot may be unavailable due to one of two reasons. A slot may have been reserved through the mobile application or have been locked from an administrative account. For simplicity and to avoid confusion among users, the LCD will simply notify the user that the requested slot is currently unavailable. Once the user has selected an open slot and entered an appropriate passcode, they will then be reminded to remember or document the passcode entered to ensure successful retrieval of their bike upon their return. Once the user confirms, the bike will lock into place. The passcode received by the keypad is received by the microcontroller with an opcode of 3 appended to the front of the integer. If at any point the user starts the process of entering inputs to the keypad be stops at any stage after the initial screen, the LCD display will display the initial message after a specified amount of time has passed. If the user would like to return the initial LCD menu they may do so by entering the ‘#’ symbol.

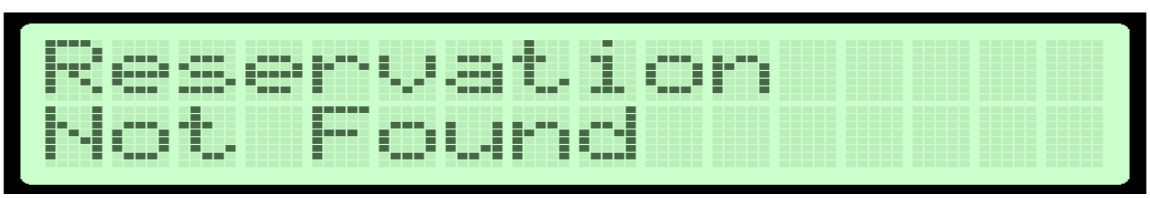

Figure 52 Reservation Not Found

EZRack aims to be a convenient and simple to use bike rack. While the mobile application provides instructions on how to use the kiosk, not all users is familiar with it and will need instruction on how to proceed on the keypad. With that taken into consideration, the user interface on the kiosk aims to be simple and easy to use in order to provide a good user experience. The goal is to avoid confusion and frustration when using EZRack in order to encourage repeat use of the bike rack. Using the LCD, the user interface will guide the users through the steps of securing and retrieving their bicycles by displaying various messages. Due to the use of a small LCD screen, it will not be possible to display all the instructions for the proper use of the keypad at the same time. In order to aid users in the use of the keypad, a sign is placed on the bike rack with detailed instructions on how to use EZRack and the keypad.

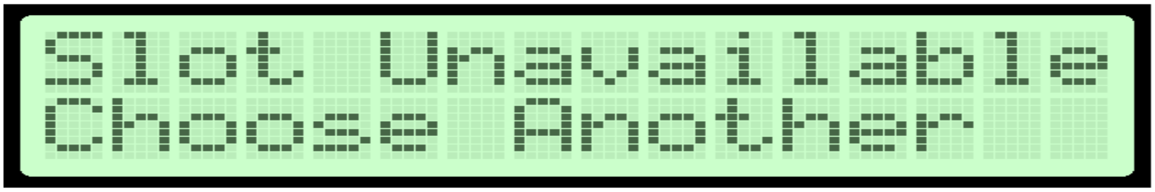

Figure 53 Slot is Unavailable

Update: The user interface on the kiosk has been changed completely. While not in use, the screen remains blank. To begin, the user will press any key followed by a five-digit passcode. In the case there is no slot with a reservation or bike with the entered passcode, the user will be assigned the first available slot and a new reservation will be made. If there is a reservation, or a bike with a passcode that is the same as what is entered, the corresponding slot will unlock. To lock a slot, the user will enter any key followed by their chosen passcode.

7 Project Implementation

The following section will describe the implementation of the hardware and software components. The design of each system is integrated together in a prototype construction and fully connected.

7.1 Solar Panel

First, to implement the power system, the solar panel is positioned correctly to ensure that there is proper incident lighting to generate adequate solar power. Generally, the solar panel must be positioned to face south to receive optimum sunlight. Since our bike rack station orientation is independent from the solar panel orientation, this can be achieved. The solar panel can be rotated on its mounted axis to face south.

7.2 Charge Controller

This charge controller is tested and assembled prior to connecting with the 50 W solar panel and the battery. Once connected to the 50 W panel and the battery, we can proceed to test the performance of the panel and the charge controller on a day with optimum sunlight available. We may account for varying conditions such as temperature and cloud coverage to understand that maximum charging may not be achievable. The 5V output of the charge controller will then be connected to the power supply of the main PCB.

7.2 User Kiosk

The user kiosk contains the main PCB, the keypad, and the LCD display. The PCB receives its power supply from the charge controller. The Wi-Fi module on the main PCB allows communication between the application and the kiosk. The keypad and LCD display are connected to the main PCB for communication. The keypad receives inputs from the user that correspond to the LCD instructions in order to operate the bike rack. This kiosk is water resistant to enclose all of the electrical components inside and maintain insulation.

7.3 Locking Mechanism

Implementation of the locking mechanism requires more mechanical assembly rather than electrical. A metal rod and PVC pipe are cut and hinged together to form the actual locking lever. This is attached to the linear actuator that will push or pull the lock. The locking mechanism is attached to the housing of bike rack in between the slots. The motor driver and actuators along with the locking lever is enclosed by the housing to ensure security and safety. The housing is insulated to prevent lock failure or overheating.

7.4 Mobile Application

The mobile application is accessed through a mobile device. The user will reserve a spot on the bike rack at a specific location. The mobile application will communicate to the user how many spots are available and remind the user of proper safety precautions for operating a bike.

8 Project Prototype Testing

This section will cover the Quality Assurance process of the development process. This section will act as our test plan, test cases and test scenarios. Due to the nature of testing, this section may be subject to change upon unexpected difficulties. When implementing, all features is tested. When implementing additional as the development process continues, new functionality is test with previous functionality. Most of our testing is manual white-box testing. Due to our time constraint, rapid application development (RAD) is used in our development process. Since RAD is based off of prototyping and iterative development, we is testing our system iteratively. Some aspects of quality of the software application testing may include the categories in Figure 54 below.

Figure 54 Quality Attributes

8.1 Primary Functions

In this section, the primary focus is Functional testing. This type of testing will verifies that each function of the software application operates in conformance with the requirement specification. This type of test is considered as black box testing, we are not concern with the source code. We only look at the main functions that will fulfill EzRack. Our primary objective is to check the functionalities of the software system.

Concentration of this test be focus in:

- Mainline functions - Testing the main functions of an application.
- Basic Usability - Basic usability testing. Check whether a user can freely navigate through screens without difficulties.
- Accessibility - Checks the accessibility for the system for the user.
- Error conditions - Usage of testing techniques to check for error conditions. It checks whether suitable error are displayed.

The process isgin with:

- Identifying test input (test data)
- Compute the expected outcomes with the selected test input values
- Execute test cases
- Comparison of actual and computed expected result

The following functionality must pass in order to meet our engineering requirement.

Test Objective: Ensure that the mechanical baseline functionality of the lock is working properly.
Test Conditions: Test lock mechanism. Mechanism must be able to engage and disengage
Procedure: 1. Supply power and signal to lock mechanism to lock and unlock.
Results:

Table 17 Testing

Test Objective: Ensure that physical keypad on bike rack works properly.
Test Conditions: Under certain circumstances, bike rack lock mechanism must be able to be lock or unlock.
Procedure: 1. Use the bike rack as a user. 2. Allow lock to engage. 3. Disengage lock by using physical keypad.
Results:

Table 18 Testing

Test Objective: Ensure that solar panel power system can provide enough power to power the bike rack for off grid use for at least 10 minutes.
Test Conditions: Under the condition where sunlight is out, solar panels must be able to charge battery.
Procedure: 1. Use multimeter to check voltage values and current values to match against calculated values
Results:

Table 19 Testing

Test Objective: Ensure that mobile application can check for a slot availability
Test Conditions: Check availability when rack is full, empty, and under reservation.
Procedure: 1. Use mobile application to check for slot availability when full, empty, and some occupied. 2. Use mobile application to check for slot availability when slot is under reservation
Results:

Table 20 Testing

Test Objective: Ensure that reservation function properly with timer.
Test Conditions: Reserve a slot on the rack and allow time out. Rack must be open for use after reservation time has ran out.
Procedure: 1. Reserve a slot on the bike rack using mobile application. 2. Allow reservation timer to run out then check functionality. 3. Reserve a slot before and after the timer runs out then check functionality.
Results:

Table 21 Testing

Test Objective: Mobile application must allow user to define their own passcode.
Test Conditions: Test forget pass pin.
Procedure: 1. Make reservation to bike rack 2. Use bike rack slot and allow lock to engage. 3. Test retrieve new passcode. 4. Use new passcode to unlock.
Results:

Table 22 Testing

Test Objective: Ensure that mobile application must be able to utilize the lock mechanism.
Test Conditions: Mobile application can lock and unlock a slot utilizing passcode. Lock initially must be engaged.
Procedure: 1. Make reservation for slot. 2. Use mobile application to unlock, lock, and then unlock.
Results:

Table 23 Testing

8.2 Mobile Application Test Plan

Due to the heavy presence of software in the creation and functionality of the EzRack, there is a huge need for software testing at every step. Each step in the design and development of the mobile application and its integration with the rest of the system will need to be thoroughly tested in order to ensure that the software is free of any major bugs. Although detecting and correcting every possible bug in the software may seem challenging and unlikely given the time constraints, every possible action should be taken to prevent any wrong functionality of the system. For this project, there is six main stages that make up the testing plan for the software. Table 24 shows a comprehensive list of all of the testing stages as they relate to the development of the entire system.

Stage	Testing Type	Project State
1	Development Testing	This is performed at the primary research phase of the project. Typically before any physical prototypes and started.
2	Individual Functionality	These tests are done once the design of the entire system has been finalized. The prototype is in the early stages of development
3	System Functionality	These tests will occur when a working prototype in in the later stages of development
4	User Acceptance Testing	The working prototype is complete at this point.
5	Partial Exhaustive Testing	These tests is conducted after the development of the EzRack has been completed. This will take place before any deliverables and demonstrations

Table 24 Application Test Plan

Each stage of the software testing will present the use of different technologies and testing methodologies. Since time is one of the biggest constraints associated with the development of the project, testing is done very quickly, allowing the presence of some contingency hours to provide insurance for any unforeseen events. This will prevent any software issues from stifling the development of the rest of the project stages that depend on them.

8.2.1 Development and Testing Environment

Before the start of any software development, it is important to select the right development and testing environment to use to build the application. The process of selecting an environment is not a simple once. At this current time there are many tools, languages, and technologies that make mobile application development possible. Each Tools and environment has its own distinct features and each language has its own purposes and uses. To create a mobile application there were a few languages to choose from including Java, Swift, Objective-C, C#, and C++. When choosing one of these languages it is also important research which development environment pairs with each one. For example, if Swift or Objective-C were chosen as the development language, all of the development would have to be done in XCode only, which is an Apple created environment. However, some other languages offer a bigger variety in terms of development environments. Java can be developed using Eclipse, Intellij, and Android Studio, to name a few. Ultimately, Android Studio was chosen to create the EzRack mobile application, and software testing was a big part of why it was chosen. Since Android Studio was specifically designed to create Android applications, its environment is optimized to quickly test emulate of the applications that are built and maintained with it.

With the amount of files and resources included in the application there is ample room for warnings and errors. If there is an error when building and compiling the program, the console will pinpoint the location of the error. Also, any errors during the development itself will appear as the code is being written. Although this feature is available across many of today's application, development environments, Android Studio quickly fixes these errors and provides helpful solutions when the error is not able to be fixed instantly.

Another handy tool that is included with the Android development environment is the mobile phone emulators. These emulators allow for efficient testing since each change in the application can be visually inspected on a mobile device. Android Studio includes many different phone screen options as well as a wide array of API's. Figure 55 displays the view of the primary emulator used for the development of the EzRack, which is a Pixel 2 phone.

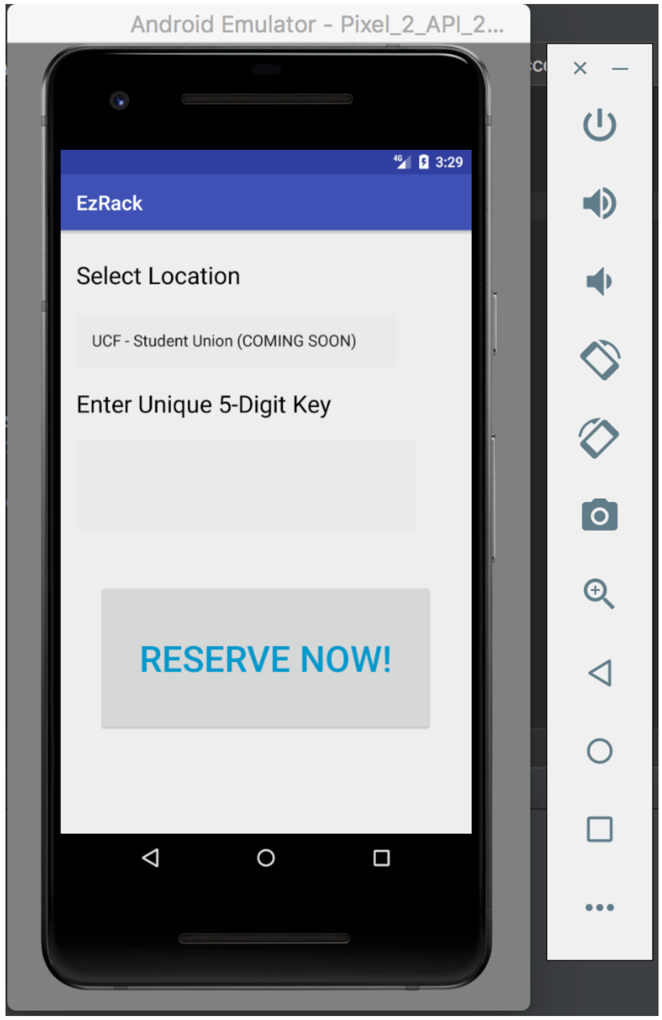

Figure 55 Pixel 2 Emulator

The Pixel 2 uses Android 22 API. It is important to choose the target operating system versions appropriately. If the target operating system version is high, the application is able to implement newer features. This can come at a cost however, since the application may not function properly on older versions of the Android operating systems. If the target operating system version is low, the application is able to run on the most amount of Android devices. This too come at a cost since it may lack some important vital features needed for the development of the mobile application. The solution here is to select an API that is high enough to implement all the necessary features, but not too high where it can be used on a wide array of Android devices. Of course, Android Studio allows for the application to be tested and different types of API's as partially shown in Figure 56 below.

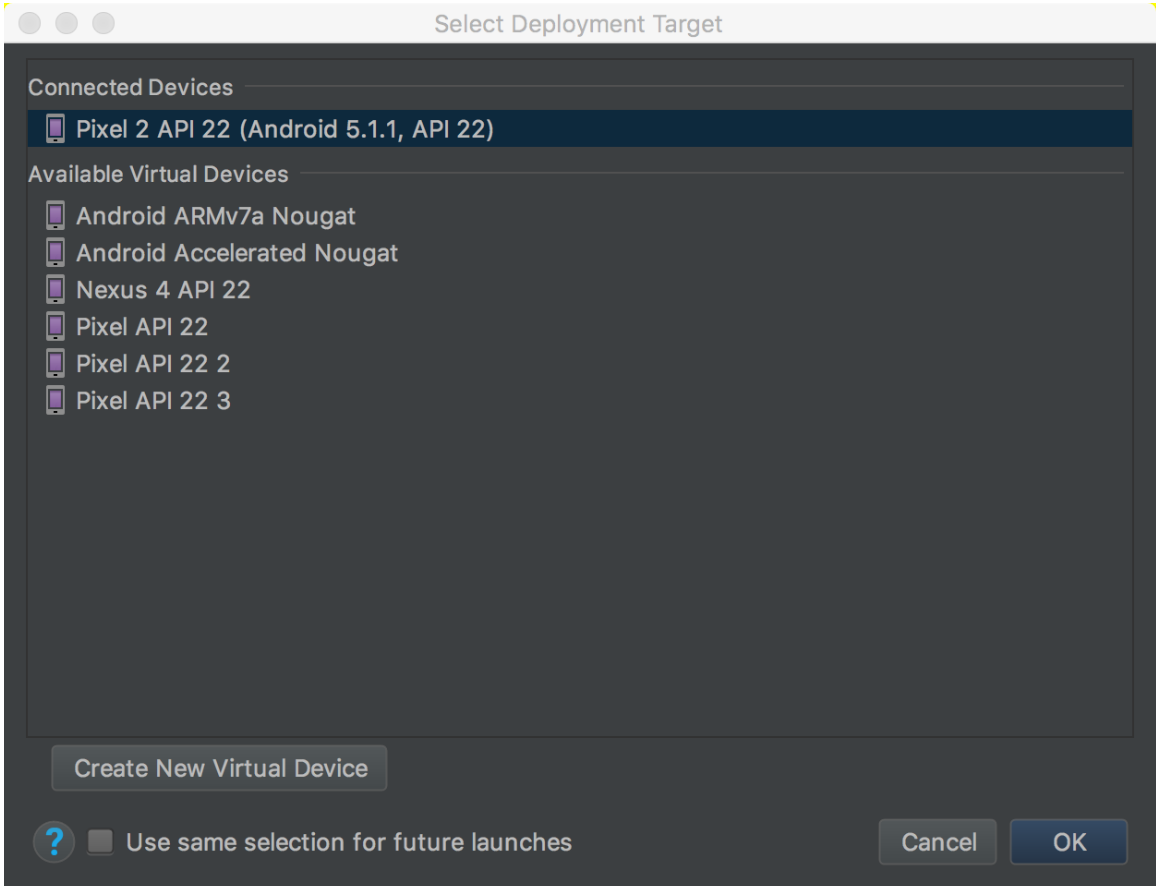

Figure 56 Creating Virtual Devices to Test API Functionality

Since the EzRack does not implement any complicated features and its user interface was made to be as simple and intuitive as possible, there is no need to target high API's. The mobile application will still be able to run on new phones that are released even after the EzRack was developed. This fact alone made choosing a target operating system version very simple.

Making sure that the mobile application could run all types of phone screen sizes is a lot more of a complex endeavor than selecting a target API. The XML code written with the application is what ultimately determines whether or not the application is compatible with a wide array of screen sizes. If the application is developed without taken the various sizes of the available displays into consideration, the application will not be functional at all. When designing the layouts, each layout must have a relation, position wise, with everything else on the same screen. Every button, box of text, and text entry must have a position representable by a ratio and not a hard coded position. To ensure that the application does indeed function properly and with all types of screen sizes, the varying device definitions on Android Studio must explored. Figure 57 shows a few of the hardware types that may be used to test the mobile application.

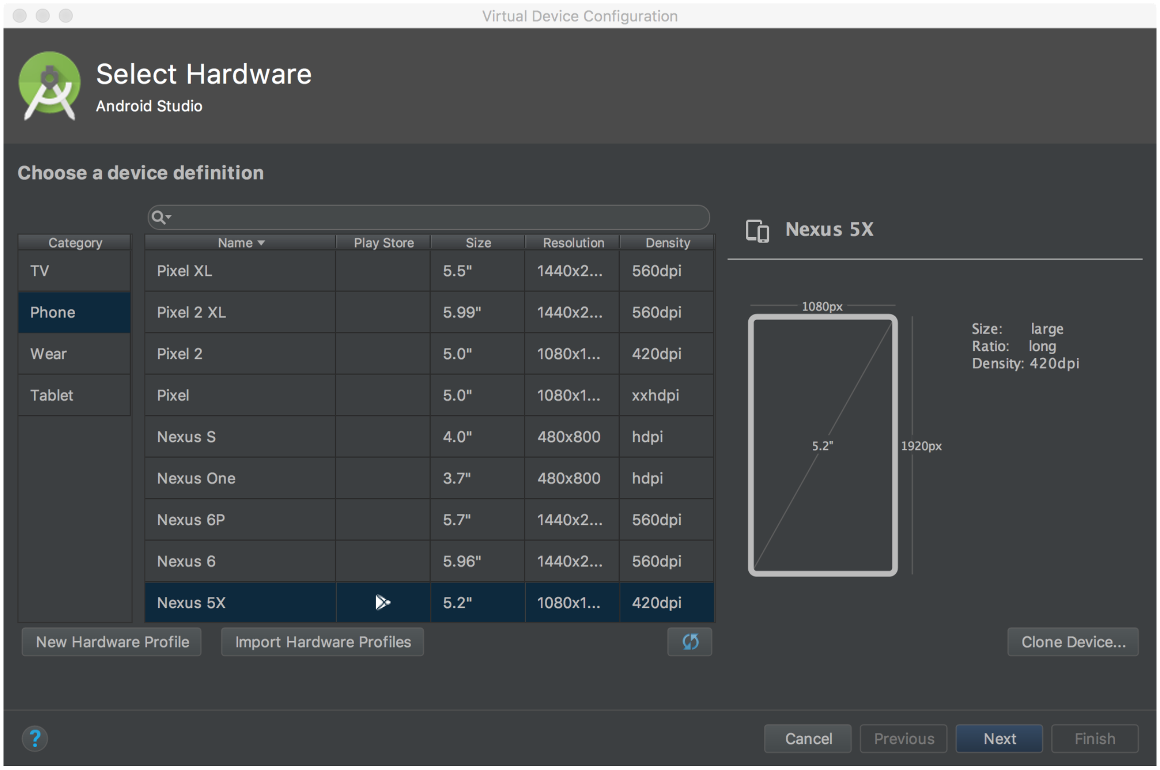

Figure 57 Hardware Selection

Any changes to the layout of the application is tested with at least 5 different hardware sizes. Changes to the actual functionality and algorithms, however, will not be tested against multiple sizes.

Certain variables, such as startup time, speed of activities, and behaviors that are not controlled by the actual applications will not be considered for the initial testing. It is assuming that the physical devices that is used for testing at the later stages is performing as they normally should.

8.2.2 Development Testing

The first stage of testing is conducted during the mobile application research and development phase. With every single change to the application, no matter how small, there is some testing done to ensure that the application has not been broken. This stage in the testing plan will occur before any of the physical components come together. This time period is especially crucial since each because of the nature of the hardware/software integration. A mobile application written in java and created using the Java based development tools is considered to be very portable. Applications can also be modified to fit a specific hardware configuration, even if that configuration is dynamic in nature. For these reasons it’s important to develop the mobile application as quickly as possible to allow time for the application to be modified with respect to those changes. In order to develop the application fast, it must be tested at each development step to keep the creation headed in the right direction. If consistent testing is not performed at each step of the development, there may be some lingering issues that will present themselves in later stages as bigger problems.

Development testing is done a little differently than the other types of testing included in the software testing plan. Most of the tests are done to examine the intents and activities of the Android Applications instead of individual specific functions and tasks. This means that the negative results of the tests may be corrected very quickly, or they may require an overhaul of the mobile applications primary functionality. If major changes are needed to correct an error caught in this testing stage, this information can be communicated to the other team's members who can then analyze whether or not they must make changes to their parts of the system. Also, unlike other parts of the application testing plan, the tests performed at this step are neither task driven nor process driven. The mobile application will not run, unless these tests are conducted which makes them vital to the development of the system. The processes and tasks in this stage are based on simple behaviors and correct code structure.

Although it is important to keep the function and behaviors of the entire system in mind when developing and testing the mobile application, it is important to isolate those factors t this current state. Trying to incorporate other parts of the system while performing essential coding duties and software tests allowed the primary focus to stay on the mobile application development, which in turn, shrinks the possibility of major bugs in the later stages. Since those later stages are indeed process and task driven, there is no room for fatal errors. Also, any major software updates will not include this stage of testing so it is important to keep that in mind during the primary development. For this reason, hard coded values should rarely ever be used. It is expected that the mobile application be created in such a way that the primary functionalities is dynamic as newer version emerge. Some of the tests conducted in this stage will identify any hard coded values that may lead bigger issues in the future. These tests are usually done in the later stages of the initial development.

Instead of using test cases to represent the types of tests that is conducted in this stage, general behavior tests is done instead. Table 25 shows the different actions and functionalities that are tested throughout the initial development.

Type	Tests	Purpose
Functional	Verify the intents correspond to the proper activities	Keeping the applications correct structure is essential
Functional	Enter text into the user test entry boxes	User input is a significant part of the applications functionality
Functional	Verify functionality of buttons	Every screen must be accessible in the app one way or another
Functional	Verify that the application is free from any “Not Responding” errors	These errors are may be signs of inconsistencies with the manifest or any other resources
Functional	Verify that the compilation is free from any errors or warnings	Errors will cause the application to crash and warnings may turn in to errors in the future
Behavioral	Check for incorrect algorithm results	Hard coded values may be a huge issue during the later stages of development
Behavioral	Verify Layout	Improperly build layouts may get in the ways of the applications correct behavior

Table 25 Development Test Scenarios

There is more tests conducted during the development aside from what is listed in the table above. However, some of those tests cannot be predicted and is conducted once the need for them is discovered.

8.2.3 Individual Functionality

At this stage, the design of the entire EzRack system is finalized and a working prototype isgin its development stage. The mobile application is completed for the most part. Before the application is tested with other components of the system, it must be individually tested to ensure that it is ready for the partial integrations. Doing this makes it much easier to isolate any problems that should arise in the later stages of the entire systems development. This stage is the shortest stage in the mobile application testing plan since a good amount of functional and behavioral testing was done in the previous development testing stage. None of the components that is physically attached to the rack while be tested at this stage.

Unlike the previous stage, the tests conducted during the individual software testing will mostly be task driven. This means that the process of the EzRack utilization will not be tested and each test is analyzed independently from one another. Process driven tests would not work for this stage since it is virtually impossible to demonstrate a process when a working prototype has not yet been finalized. Having these tests be task driven works in this case since none of the main users or contingents is conducting any of the tests. The tests done in this stage are simply used to finalized development and prepare for the first step of the hardware software integration. In addition, task driven tests will allow the test plan to be more flexible in the generation of any new test cases. Should any test cases be added, modified, or removes, it will not affect the test cases done in later stages of the testing plan. As the testing stages continue, the entire focus of each test will shift closer to that process driven outlook. The penultimate testing plan stage, User Acceptance Testing, is responsible for transforming the overall testing methodology from task driven to process driven. Before that stage it reached, all of the test cases is task driven.

Testing in this stage will also differ from the previous stage in the sense that most of the tasks included in the tests will focus on individual actions that would essentially be verified only once. Tests during the development stage were done several times throughout the course of the period. Once the results at this stage have been successfully verified, there is no need to verify them once again in the context of the application working individually. All of the tests will not be repeated during later stages since any form of redundancy should be avoided to fall in line with the timing constraints. This approach also isolated any issues that arrive in the future which will make the overall application easier to debug, even when the system has been completely built. It is also important that there should be no further software development aside from modifications and additions to small parts of the application. The mobile application is complete on its own during this stage which will lead to the hardware and software integration.

Initially, a list of test cases is created based on the information that is currently known about the entire system at the start of the development phase. Ideally, the test cases must provide some meaningful depth regarding how it relates to the overall behavior of the mobile application. As previously discussed, they will not have to have depth in the realm of the entire system as a whole. This meaningful depth provided in theses test cases ensures that no time is wasted on items that will not need to be tested, as well as items that have been tested previously. Table 26 shows the test cases included in the stage of the testing plan.

Test Number	Task	Importance
1	Verify that the Reserve Now button is not activated until the user enters a key	This will prevent the system form crashing if it can't handle certain inputs
2	Verify that the application notifies the user when a key is not in the proper format when reserving	This ensures that the application retained an interface that is user friendly
3	Verify that the lock button is not activated until the user enters a key	Verify that the application notifies the user when a key is not in the proper format when locking
4	Verify that the application notifies the user when a key is not in the proper format when locking	This ensures that the application retained an interface that is user friendly
5	Verify that the unlock button is not activated until the user enters a key	Verify that the application notifies the user when a key is not in the proper format when locking
6	Verify that the application notifies the user when a key is not in the proper format when unlocking	This ensures that the application retained an interface that is user friendly
7	Verify that all buttons are activated when correct keys are used.	This ensures that the application is unable. Without this the application will essentially be useless

Table 26 Isolated System Test Cases

The list of test cases in this stage are dynamic and it is expected to grow. The skeletal test cases, however, are needed as a bare minimum of how much software testing is done at this stage

8.2.4 System Functionality Testing

System functionality tests are used to verify that the integration of hardware and software is done correctly. The EzRack mobile application has no use if it does not work well with other components in the system. This stage is separated into two parts. The first part will test the application with the database and the database only. During this first part, the prototype is in its later stage of development and is not ready for the complete software integration. The incomplete prototype will not hinder any tests from being done as far as the mobile application communicating to the Firebase database. With that being said, all of the tests during the first stage will incorporate the Firebase database in one way or another. This doesn't include any individual tests done with the database only. For a test case to be considered for this stage there must be some interaction between the database and the mobile application itself. Table 27 contains the test cases for the first part of the system functionality stage.

Test Number	Task	Importance
1	Verify that keys are stores into the database when the user makes a reservation	This prevents a user making a false reservation
2	Verify that the mobile application accurately determines if a slot is indeed free	This prevents the mobile application from overriding an occupied slot in the database
3	Verify that the mobile application accurately determines that an EzRack location is full	This prevents the mobile application from overriding an occupied slot in the database
4	Verify that list of locations on the mobile application accurately corresponds to the location structure of the database	This prevents an reservation being made at the incorrect slot
5	Verify that the mobile application displays the correct slot number	This will prevent confusion as a result of incorrectly assigning a slot

Table 27 Database Functionality

The second part of the system functionality tests incorporates the entire system. This step takes place right after the working prototype has been completed. At this point, all of the testing for each of the individual components have been completed. Some of the components of the system have even been integrated. A good example of two parts of the system being integrated at this stage is the mobile application and the database, as discussed previously. This is the first major step is the completion of the entire project since it is the first opportunity to manifest the functional correctness of the system. These test cases will incorporate three or more parts of the entire system. Table 28 shows a list of tests cases that is used at the part of the application testing stage.

Task Number	Task	Importance
1	Verify that the mobile application can determines that a slot is indeed occupied	This prevents the mobile application from overriding an occupied slot in the database
2	Verify that the mobile application determines that a slot is free	This will essentially prevent a false negative situation
3	Verify that the mobile application can lock a slot for a reservation	Reservations are a crucial part of the importance of the mobile application
4	Verify that the mobile application is cannot manage a slot that has not been previous reserved	This prevents a user from errors in the physical system
5	Verify that the slot numbers are consistent through the entire system	This prevents an reservation being made at the incorrect slot
6	Verify that the locations of the EzRack are consistent throughout the entire system	This prevents an reservation being made at the incorrect slot
7	Verify that errors in key entry do not affect the state of the bike Rack	This ensures a secure system

Table 28 System Integration Testing

These tests will conclude the integration step on the development and the fully functional prototype is ready to be tested by potential users.

8.2.5 User Acceptance Testing

Generally Speaking, Acceptance Testing is the final stage in the software development lifecycle. At this stage, the system is tested with actual end users in actual real world scenarios. The importance of User Acceptance testing is known all throughout the software engineering community. All of the specifications laid out at the start of the project is considered when testing at this stage. Although some User Acceptance testing implementations take a more task drive approach, this project will essentially go in a different direction. It is important that the end users see the big picture when testing the EzRack. Every step in the process of reserving a slot and managing a slot is a part of a bigger process. This is why the EzRack will take a process driven approach when creating test scenarios for the User Acceptance Testing cases. Being process driven is a direction that will allow the testing to be fully integrated. The point of this approach is not to see if the user can make the system fail, but instead to help user to verify and understand each step of the process. This is why there are multiple stages in the testing plan. Those multiple stages are used to generate a system failure due to an incorrect application structure. This will help the EzRack development finalize its prototype and essentially turn it into a finished product.

Aside from being process driven, this stage differs from the previous stage in the sense that the scenarios is specifically constructed. All of the test cases is in the form of specific actions that the user will make during the utilization of the EzRack. Since this part off the testing is solely focused on user experience and not system functionality and behavior, each test case is conducted in a specific order. This differs from the previous stages, where the tests could have been done in any order and still be equally affected. Also, it is not guaranteed that the test User Acceptance Tests are comprehensive. As long as the tests can display this is where the term “Process” comes in; his whole stage will paint a picture through a process. If this stage in the testing plan is task driven instead of process driven, the user may not give out accurate feedback on the effectiveness of the system as a whole. The user’s feedback at this stage will help shape the next step in the improvements to the system. If any dire last minute changes need to be done, it is likely that these needs is revealed at the User Acceptance Testing phase. The completion of this stage will ultimately determine where the immediate attention should be allocated.

The User Acceptance Tests is split up into three distinct parts. The first part will test the reservation feature only. The second part will test the slot management aspect. Finally, the third step will test the rest of the mobile application. These three steps will effectively have the user test the entire mobile application and its purpose. This also leaves the most important functions at the beginning, but also doesn’t leave out the lesser used parts of the mobile application. Ideally, the User Acceptance Tests should be done in iterative sprints which will make it much easier to determine progress. This progress is related to determining the functionality of each individual system. The EzRack system, however, is not very complex which makes it possible to test each state of the system with a single sprint. This will make it unnecessary to try to break up the testing into sprints since all of the tests is conducted together. Should the system become complex enough that it would be beneficial to test using iterative sprints, the nature of the User Acceptance Testing will allow it to do so. This makes the testing of this application scalable. Table 29 shows the first set of UAT test cases.

Test Number	Task	Expected Result
1	The user will enter a key that has too many or too little characters and then make a reservation request	Error message displayed
2	The user will enter a key that contains a hash symbol and then make a reservation request	Error message displayed
3	The user will select a location that is not available, enter a valid key, and then make a reservation request	Error message displayed
4	The user will select a location that is available but full, enter a valid key, and then make a reservation request	Error message displayed
5	The user will select a location that is available, enter a valid key, and then make a reservation request	The bike slot is reserved
6	The user will make a reservation and unlock their slot within the allotted time	Slot will unlock for user
7	The user will make a reservation but not unlock it to use	Slot is freed for other users

Table 29 Reservation UATs

The next set of tests is focused on managing the slots after the user has made a reservation. There is more test cases included in this set of tests since there are more different outcomes compared to the previous set of tests. Table 30 below shows the second set of UAT test cases.

Test Number	Task	Expected Result
1	The user will enter a key with too many or too little number and try to lock and unlock a slot	Error message displayed
2	The user will enter a key with a hash symbol or star symbol and try to lock and unlock slot	Error message displayed
3	The user will enter a key that is not associated with a slot and try to lock/unlock	Error message displayed
4	The user will try to lock/unlock without entering a key	No changes
5	The user will try to lock a slot that has is already locked with correct key	Notification displayed
6	The user will try to unlock a slot that is already unlocked by entering the right key	Notification displayed
7	The user will enter a key for a slot that has just been free, and then will try to lock/unlock a slot	Error message displayed
8	The user will enter a key for a slot that has not been reserved via mobile application, and try to lock/unlock a slot	Error message displayed

Table 30 Managing Slots UATs

Finally, the remaining mobile application functionality is tested with the third set of User Acceptance Testing cases. There is fewer tests for this part of the UATs since there is not much to test in the "Help" branch of the application. Nonetheless, it is important to test these functionalities in order to complete the process oriented nature of the UATs. Table 31 shows the final set of UAT test cases.

Test Number	Task	Expected Result
1	The user will select the "How it Works" option from the Help menu	The "How it Works" screen will appear and display all of the relevant information. There is a scroll bar present
2	The user will select the "About Us" option from the Help menu	The "About Us" screen will appear and display all of the relevant information. There is a scroll bar present
3	The user will select the "Contact Us" option from the Help menu	The "Contact Us" screen will appear and display all of the relevant information
4	The user will select the "Current Locations" option from the Help menu	The "Current Locations" screen will appear and display all of the relevant information. There is a scroll bar present

Table 31 Help UATs

After the Use Acceptance Testing is completed, the EzRack will essentially be ready for a complete handoff. During this time, there will still be software testing present, but it won't affect the original complete functionality. Any tests conducted after this point are solely for the purpose of implementing any new features or fixes through a software update in the future. With the completion of the User Acceptance Testing, the EzRack is ready for implementation.

8.2.6 Partial Exhaustive Tests

In a typical technical environment, the final phase of software testing is the User Acceptance Testing. This still holds for the EzRack, however, there will always be testing done even after the software has been completed and implemented. In order for the software to have updates that are meaningful and necessary, it is important that any small problematic items be detected. The EzRack employs a new distinct testing methodology to help track and fix issues before a user can detect them. The term Partial Exhaustive Tests implies that the system and application will go through tests that represent all sorts of unusual behavior. Since exhaustive testing is impossible to implement, this method of testing will try to be as exhaustive as possible while also being realistic. Table 32 shows examples of some of potential tests.

Task Type	**Test Description**
Functional	Enter the number “00000” into the text entry box to make a reservation. The application should allow to be entered as a key without registering the slot as empty. The same number should also be entered into the “Managing Slot” text box.
Functional	Enter a key to make a reservation and use the same key to make another reservation during the same time. The reservation should not complete and an error message is displayed to the user.
Functional	Try to open an external document with this application. The behavior of the application should not change
Behavioral	Leave the application running for a while to analyze its behavior. The application can theoretically run forever as long as no external events prevent it from doing so.
Behavioral	Analyze the time it takes for the database to update a slot and try to make a reservation in the slot before it does. The updated slot should prevent the reservation from being completed.

Table 32 Testing

The user will not be aware of any of these tests and these tests will not get in the way of the EzRack implementation. These tests are for future updates only.

8.3 GUI Test

User interface is the first line of interaction that customer faces. So it is important that the user interface have a good impression of our work. A good GUI provide the ease of use for our customer and that is a big factor to consider. In addition, it is also important to make our GUI as secure as possible to prevent security issues such as Cross-site Scripting. Once this happens our database can be compromised. Therefore leading to failure of product. The following are the basic GUI test cases can either make or break our product first impression.

These basic test are also required when there is an update the GUI look and feel, new features is added to the software, Defect fixing, and performance issue fix. This is to confirm that a recent program or code change has not adversely affected existing features. It ensures that old code still works with the new code changes are done.

There are three known techniques to regression testing. Depending on the situation:

1. Retest all is a method in which all tests in existing testing bucket should be evaluated. This requires a huge amount of time and resources.

2. Regression Test Selection this method select part of the test suite to run rather than re-executing the entire test suite.

3. Prioritization of Test Cases this method prioritize the test cases depending on business impact, critical & frequently used functionalities

Test Objective: Testing the size, position, width, height of the elements
Test Conditions: Verity all text boxes and labels are aligned properly
Procedure: Go through mobile/web application pages and check for alignment.
Results:

Table 33 Testing

Test Objective: Testing of the error messages that are getting displayed.
Test Conditions: Notification/Error messages must be aligned.
Procedure:
Results:

Table 34 Testing

Test Objective: Testing the different sections of the screen.
Test Conditions: Verify that alignment throughout the entire screen in proportionate.
Procedure: Check buttons, text fields, white space alignment for consistency and aligned. UI elements must be consistent throughout the entire screen.
Results:

Table 35 Testing

Test Objective: Testing of the font whether it is readable or not.
Test Conditions: Ensure that all text are aligned and wrapped in a manner that is aligned and consistent with the page.
Procedure: Check all pages for fonts are easy to read and proportionate. Ensure that text wraps in an appropriate manner.
Results:

Table 36 Testing

Test Objective: Testing the colors of the fonts.
Test Conditions: Verify that text color is correct.
Procedure: Check codes against emulator for proper text color output.
Results:

Table 37 Testing

Test Objective: Testing the colors of the error messages, warning messages.
Test Conditions: In a condition when there is an error.
Procedure: Verify that proper error generated if there is a mistake, red text for errors.
Results:

Table 38 Testing

Test Objective: Testing of the spelling.
Test Conditions: Ensure correct text is used instead of symbols.
Procedure: Go through all page and check text.
Results:

Table 39 Testing

Test Objective: The user must not get frustrated while using the system interface.
Test Conditions: Check for ease of use. GUI should be simple and easy to navigate.
Procedure: Check for consistencies functionality of UI. Check for use of common practice in UI. Check flow consistency of UI elements.
Results:

Table 40 Testing

Test Objective: Testing whether the interface is attractive or not.
Test Conditions: Visual representation of GUI can add value to product.
Procedure: check aesthetics consistencies throughout the application
Results:

Table 41 Testing

Test Objective: Testing of the disabled fields if any.
Test Conditions: Ensure fields are disabled when they are not in used or only allow data type of that field to be entered.
Procedure: Make sure unused fields are disabled and limit data type and number of character in fields.
Results:

Table 42 Testing

Test Objective: Testing of the headings whether it is properly aligned or not.
Test Conditions: Ensure that all headings are consistent and aligned.
Procedure: Go through all pages and check for alignment.
Results:

Table 43 Testing

8.4 Database Testing

Back end testing does not required the GUI, request can be passed directly through some browser or application with the parameters required for the function and get a response. In our case it is in JSON type. This can be verified by directly connecting the database and verifying using SQL queries. Currently our database design being designed. Once it is done we will employ several type of database testing such as the one listed below [TL1]:

- Structural Testing - deals with table and column testing, schema, stored procedures, etc.
- Functional Testing - check functionality of database from user point of view.
- Non-Functional Testing - involves with load-testing, risk testing, stress testing, minimum system requirements, and performance of database.

The main advantage of back end testing is that it helps discover many bugs in the early development process.

8.5 LCD Display Design Testing

Figure 58 illustrates the connection between the LCD module and the development board. To ensure the LCD is properly connected and operational, it is connected by itself to the development board and a simple program is created to display the words “Hello World!” to the screen as shown in Figure 58. Assuming the initial test is performed indoors, it will have to be repeated outdoors to make sure the text is still visible in bright daylight. Depending on the clarity of the text, the potentiometer may have to be adjusted in order to improve the visibility of the text.

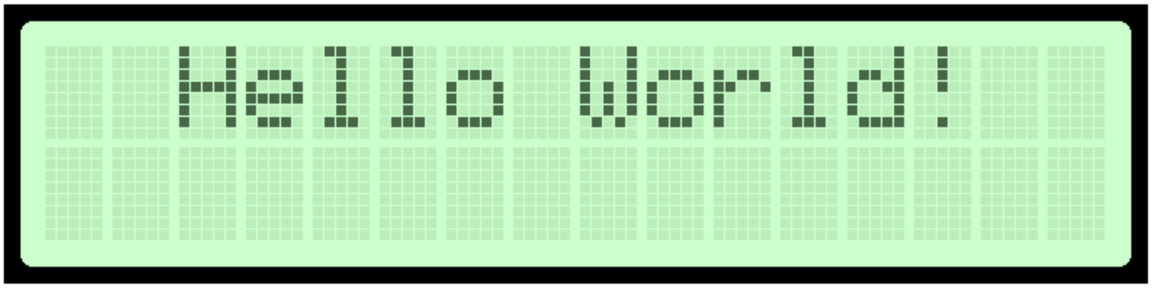

Figure 58 Hello World Program

8.6 Keypad Design Testing

For many users, the keypad is the sole means by which users will actively interact with EZRack. The keypad is used when users wish to lock their bikes and when they return to unlock them. Without a working keypad, EZRack is forced to rely solely on the mobile application to perform those tasks and will render users without an android phone with the installed application unable to use the bike rack. Table 44 lists all the tests the keypad is subjected to in order to ensure it works properly and fits all the needs of EZRack. Though all passcodes are comprised solely of a string of numbers, users will occasionally be asked to enter a letter or the pound key. At this time, the keypad is tested by itself to ensure any problems encountered are due to the module itself and not due to any other modules interfering with the keypad. At a later date, the keypad is tested with the other components jointly.

Test	Description
1	Read a number from the keypad
2	Read a string from the keypad
3	Press the # key
4	Press and hold a key

Table 44 Keypad Test Cases

The first three tests are comprised of reading in simple inputs through the keypad. A simple C program is written in the Android IDE to test the functionality of the keypad. The first test is to simply read in a number entered through the keypad and display it on the screen of the computer. The second test will do the same, except it will read a string of letters from the keypad and display it on the screen of the computer. Similarly, the third is comprised of pressing the pound key and displaying the result on the computer screen. The purpose of the first three tests is to make sure that input can be entered on the keypad and correctly read and used by the microcontroller. The Matrix Keypad Library used in reading inputs from the keypad is non-blocking and final test is ensuring that feature works properly. In the fourth test, two numbers is entered through the keypad, one at a time. The last number is held down indefinitely. Since the keypad is non-blocking, having the key held down should not prevent the microcontroller from proceeding to add the two inputs and displaying the sum onto the computer screen.

8.7 Wi-Fi Module Design Testing

Before moving on to the design and testing of the RTC module, the Wi-Fi module is connected by itself to the development board and thoroughly tested to make sure a stable and consistent connection can be made. A stable and reliable connection is necessary for the reservation feature of the mobile application to function properly. Reservations made on the mobile application must be upheld by the embedded system. Once reserved, the slot must be rendered unusable by all but the reservee for the specified amount of time. To ensure reservations made through the mobile application are available to users upon arrival, the embedded system must periodically check the website for any changes in the slot's reservation status and update its own reservation array. In order for the mobile application's reservation feature to be accurate, the embedded system must connect to the internet and update each slots status whenever a slot is claimed or becomes vacant. Having claimed slots appear on the mobile application as available for reservation or having reserved spots claimed by others besides the reservee will cause user frustration and will render the mobile application useless.

Table 45 lists the tests that is applied to the Wi-Fi module to verify it is functioning properly. These tests are designed to thoroughly test the Wi-Fi module and ascertain it can and will meet the project's needs. The first test is simply ascertaining the Wi-Fi module is capable of establishing and maintaining an internet connection. The objective of the second test is to make sure the Wi-Fi module is able to read in the necessary inputs provided by the mobile application and the microcontroller is able to make use of said inputs. The second test is comprised of two parts. The first part will make sure that once the internet connection has been made, the microcontroller is able to read in a number through the use of the Wi-Fi module and print it to the screen of the computer used for the programming and testing of the development board. Part two of the second test will read in a string from the website and print it to the screen of the computer. The third test's objective is to make sure the microcontroller is able to send the necessary data to the website for use of the mobile application. Like the last test, it is comprised of two parts. The first part is comprised of sending a number to the website. The second part involves sending a string to the website. The objective of the fourth and fifth tests is to test the reliability of the connection and the modules ability to consistently check the internet for changes in reservation status and to upload changes of slot availability on the bike rack. The embedded system must be counted on to be able to update its information frequently in order for the mobile application to provide reliable information to the user. Similarly to the second test, the fourth test retrieve a number and a string to the website and display it to the screen. Where it differs is that instead of it being a one-time thing, it will do it multiple times in the span of a couple of hours. The fifth test will do the same thing as the third test except it will repeat those actions several times in the span of a couple of hours. Tests four and five will determine if the Wi-Fi module is able to maintain a consistent and reliable connection to the internet.

Test	Description
1	Establish a connection
2	Read in input from a web page
3	Send output to a web page
4	Randomly read input from the web page over a period of a couple of hours
5	Randomly write to the webpage over a period of a couple of hours

Table 45 Wi-Fi Module Test Cases

8.8 RTC Module Design Testing

Before being incorporated into the overall design of the embedded system, the real time clock module is tested apart from the other components to ascertain any issues encountered relate solely to the real time clock module and not to any interference from the other components. By testing the components individually, the debugging process is simplified and the lengthy process of tracking down the source of the errors is shortened. The test performed on the real time clock module are listed in Table 46. Once the module is determined to be fully functional and operational, it is incorporated into the complete embedded system design.

Test	Description
1	Set time and leave running for a couple of hours
2	Disconnect from design board and reconnect

Table 46 RTC Module Testing

The goal of the first test is to determine how accurately the real time clock module keeps track of time. The first test performed on the real time clock module is simple. The real time clock module is programmed with the correct month, date, year, and time and left running for a couple of hours. After some time has passed, the time is checked against the time on a cell phone/computer/watch etc. to determine if the real time clock module has successfully and accurately kept time. The goal of the second test is to determine if the real time clock module will continue to keep accurate time after power failure. To simulate power failure, the real time clock module is disconnected from the development board which is active as its power source for a period of a few minutes to an hour. After some time has passed, it is reconnected to its power source in the form of the development board and the time is checked against the time of a cell phone/ computer / watch etc. to determine if the real time clock module has successfully kept time in the wake of a power failure.

8.9 Combined LCD and Keypad Design Testing

At this point, all the components have been individually tested and have been found to function as intended. The first step in creating the complete embedded system is to combine the LCD display and the keypad. Both functioned properly independently and now they must be tested to make sure they work well once connected at the same time. To insure they work well together several test cases have been developed as listed in Table 47.

Test	Description
1	Run all tests in the LCD Display Design Testing section
2	Run all tests in the Keypad Design Testing section
3	Write a program that uses the LCD display to prompt the user to enter two numbers on the keypad, adds them and displays the sum on the LCD

Table 47 Combined LCD and Keypad Design Test Cases

The first test's objective is to ensure the LCD continues to function properly after the keypad has been introduced to the circuit. To test this, all the tests performed in the LCD Display Design Testing section is performed again to ascertain there is no interference from the keypad and that there are no problems with the LCD display. Similarly, the goal of the second test is to ascertain the keypad continues to function properly after the introduction of the LCD into the circuit. To make sure the keypad continues to function as desired, all the tests performed in the Keypad Design Testing section is run again. Any deviation in the results of the first two tests will signal a problem in the design of the circuit which may lead in a change in the way the components are connected in the circuit. The goal of the third and final test is to make sure that both the LCD and keypad components are capable of functioning together. To test this, a simple program is written in the Arduino IDE. The program will display a prompt on the LCD asking the user to enter two numbers using the keypad. The user will enter first one number and then the other. As the numbers are being entered on the keypad, they is displayed on the LCD display. Once both numbers have been entered, the sum is displayed on the LCD.

8.10 Combined Module Design Testing

With the addition of the Wi-Fi module, the circuit must again be tested to ascertain all three components can operate jointly. The LCD and Keypad have already been determined to function properly together and any problems experienced at this stage are likely caused by the Wi-Fi module being introduced to the design. The complete test plan is listed in Table 48.

Test	Description
1	Run all tests in the LCD Display Design Testing section
2	Run all tests in the Keypad Design Testing section
3	Run all tests in the Wi-Fi Module Design Testing section
4	Run all the tests in the Combined LCD And Keypad Design Testing section
5	Write a program that prompts the user to enter a number on the keypad using the LCD. A second number is gotten from an online website and added to the first number. The sum is displayed on the LCD.

Table 48 LCD/Keypad/Wi-Fi Module Test Plan

The first three test focus on the individual functionality of the components. Before moving on, each component is tested to make sure that there is no interference causing any one module to malfunction. The first test makes sure the LCD continues to function as expected by running all the tests in the LCD Display Design Testing section anew. The second test ascertains the Keypad continues to run smoothly by re-running all the tests in the Keypad Design Testing section. The third test focuses on making sure that the Wi-Fi module continues to work without any issues with the introduction of the LCD display and the keypad. The LCD and Keypad were already determined to function properly together. Test four makes sure that is still the case after the Wi-Fi module has been added to the mix by performing the tests run in the Combined LCD and Keypad Design Testing section. Of the five tests, the last one is the only new one. Test five makes sure all three components are able to function jointly. A program is written using the Arduino IDE. The program will display a prompt on the LCD asking the user to enter a number on the keypad. The microcontroller will then retrieve another number from the online website through the use of the Wi-Fi module and add it to the number entered through the keypad by the user. The sum is displayed on the LCD display.

8.11 Complete Design Testing

With the addition of the real time clock module to the design, the circuit is now complete. All that is left is to test it to make sure the addition of the rtc does not cause any of the components to malfunction. A detailed test plan has been outlined in Table 49.

Test	Description
1	Run all tests in the LCD Display Design Testing section
2	Run all tests in the Keypad Design Testing section
3	Run all tests in the Wi-Fi Module Design Testing section
4	Run all tests in the RTC Module Design Testing Section
5	Run all the tests in the Combined LCD And Keypad Design Testing section
6	Run all the tests in the Combined LCD, Keypad And Wi-Fi Module Design Testing section
7	Display time on LCD display

Table 49 Complete Design Testing Plan

The first four tests in the testing plan ensure the individual components continue to operate smoothly. Tests five and six ensure all modules besides the real time clock continue to function properly together. The goal of test seven is to ensure the real time clock module is capable of functioning in a circuit along with the LCD display, keypad, and the Wi-Fi module. The seventh test is simple. It will display the current time on the LCD screen. Though long, the purpose of this step by step approach to designing and testing of the embedded system is to simplify the debugging process. By testing each component individually and gradually adding more modules, it is easier to isolate errors and determine their source. Time is limited when building the embedded system and it is much faster to track and fix errors on a smaller circuit compared to a larger one.

8.12 Embedded Software Testing

A detailed test plan for the embedded software is outlined in Table 50. Tests one through seven ensure the embedded system and the mobile application are able to effectively communicate with each other. Tests eight to thirteen make sure EZRack is fully functional without the mobile application.

Test	Description
1	Receive a number from the mobile application
2	Receive a string from the mobile application
3	Send a number to the mobile application
4	Send a string to the mobile application
5	Receive a reservation request from the mobile application for an empty slot
6	Receive a reservation request from the mobile application for an occupied slot
7	Send an update on slot availability to the mobile application
8	Have a user enter the correct password for a slot
9	Have a user enter an incorrect password for a slot
10	Have a user attempt to use a reserved slot
11	Have a user try to redeem an expired reservation
12	Have a user attempt to select a password that is not the correct length
13	Have a user attempt to select a password that has characters that are not numbers

Table 50 Embedded Software Design Test Plan

8.13 Locking Mechanism Testing

Table 51 below specifies the testing specifications for the locking mechanism.

Test	Description
1	Linear actuator responds to signal from MCU
2	Locking mechanism operates properly
3	Locking mechanism fully extends across bike slot
4	Locking mechanism supports bike wheel

Table 51 Lock Mechanism Testing

9 Handoff and Maintenance

It is important that the implementing party understands that the EzRack should be properly used and maintained. The entire system should be handled with care at all times and should not be misused or tampered with. Any actions resulting in misuse can result in a system that does not perform as it should and/or complete system failure. It should also be understood that the system is not one hundred percent effective in stopping theft. User should take all the necessary precautions to make their EzRack experience as safe and enjoyable as possible.

9.1 Placement and Modifications

Although the EzRack bike system is a fully assembled product, it is the responsibility of the implementing party to properly install it. This installation includes securely mounting the EzRack to the ground and placing the system in an area where it will receive direct sunlight whenever the sky is clear. The EzRack may still function without the use of direct sunlight but the battery will either need to be recharged manually or replaced. In order for the EzRack to work as intended, the use of solar energy is a necessity. As long as the EzRack is properly installed, the battery should not have to be replaced or recharged as long as the battery is functioning at its proper state.

The system will arrive mostly set-up and ready to use. The Wi-Fi module will already be configured to connect to your network. All that is needed is to connect the power block to a 5.3V power source. Once connected, wait at least three minutes for the Wi-Fi module to connect to your wireless network. Once connected, the system is ready to use. To prevent damage to the system, please ensure it doesn’t not receive more than 5.6V.

Some of the components that are included in the system is replaceable by the implementing party. The solar module can be replaced in the case that it is no longer functioning properly. The battery will also be replaceable as well as the lights that are used on top of the design. It is the responsibility of the implementing party to ensure that the parts used as replacements match the exact specifications as included in the original assembly. Other parts on the system will not be replaceable, these parts include: locking mechanisms, keypad with LCD, and any electronic components. Any other components that are used to power the EzRack will also not be replaceable. Replacing any unreplaceable parts may cause the system to not work as intended or not work at all. If there are any questions regarding what can be replaced or not the user may contact the EzRack directly using the email provided in the mobile application.

User friendly instructions is included with the EzRack system for both the keypad/LCD interface and the mobile application interface. The implementing party may choose to post any additional information on the EzRack such as contact information and specific requests/instructions. Although placing small signs on the system to assist the user in various ways is suggested, placing any other objects at and around the system is not recommended. Please note that there is also a quick guide that the user may view on the mobile application.

The slots on the EzRack system will not be compatible with any other locks or chains. In order to use a space to secure a bike, the user must lock their bike in place using either the mobile application or the LCD interface. The slots will only be compatible with normal and small size bicycles. Motorcycles, scooters, skateboards, and other riding devices will not be able to be secured with the EzRack system.

9.2 General Information

Due to the complex nature of the system, using a high powered washing method to clean the device is not recommended. To properly clean the system, it should be carefully scrubbed down and rinsed with a lower powered sprinkler system.

Although EzRack was developed with security as its top priority, it is important to follow any and all procedures in place to deter theft. If a user is notified of any potential theft, they should immediately report it to the proper authorities. The user should never share their EzRack passcode with anyone to eliminate the possibility of someone else uses the key to take a bike. Users should also be sure to use different keys every time they would like to lock or reserve a bike slot. If the user continues to use a key every time they have used on the EzRack before, there’s a greater chance of someone else getting a hold of it.

10 Troubleshooting

Throughout the development process, we will troubleshoot our design of project, hardware, and software. In this section will overview some of the standards, and list some of the possible scenarios and solutions.

1. Solar panel not charging battery.
   Solution: Clean solar panel. Make sure all cables are connected properly and to its appropriate components. Use multimeter to check for signals.

2. Battery not supplying power.
   Solution: Use multimeter to check voltage and current. Check readings against rated standard of the battery. Attempt a deep cycle charge.

3. DC-to-DC converter not working properly
   Solution: If converter is hot or smells burnt, use more heat sink on the regulator

4. Microcontroller not receiving data
   Solution: Check Wi-Fi module connections. Test connection and ensure that database secret keys are correct. Check against database.

5. Microcontroller not transmitting data
   Solution: check Wi-Fi module connections. Test connections and ensure that database secret keys are correct. Check against database.

6. Lock mechanism not working properly
   Solution: Ensure physical connection are working properly and that power supply is adequate. Check if signals can be transmitted.

7. Lock mechanism does not engage
   Solution: Test physical connection contact between components. Check for physical obstruction. Ensure that there is power.

8. Lock mechanism does not disengage.
   Solution: Test physical connection contact between components. Check for physical obstruction. Ensure power is being supply.

10. Memory fails
    Solution: This error usually caused by power or ground problems. A peripheral device (led or switch) may be disrupting the supply in (VDD), (MCLR), or the ground pin (VSS). Remove the offending external device or re-installing MPLAB IDE may solve this error [TL2].

11. Device not found or connected Target detected
    Solution: Target Device ID (0x0) does not match expected Device ID

(0x460). This problem is usually caused by the chip not receiving power. Make sure the programmer settings has power enabled (and correct voltage) [TL2].

12. Arduino board doesn't turn on

    Solution: If powering through USB, the jumper should be on the two pins closes to the USB plug. If powering with external power supply, the jumper should be on the two pins closes to the power plug.

11 Administrative

The following section covers administrative details such as budget and legalities.

11.1 Budget

The following section will outline the total costs associated with material costs and processing fees. Using the pricing table 52 shown below, our estimation for the total budget is laid out according to the item, its cost, the quantity needed, and the total cost. The budget is based on the upper bound costs of each component. This does not include reorders due to malfunction or incompatibilities. Funding is sourced directly from the group members. Table 53 shows the final budget and costs of the complete project.

Projected Component Costs			
Item	**Cost**	**Quantity**	**Total Cost**
50W Solar Panel	$100	1	$100
12 V Battery	$30	1	$30
Power Converters	$15	2-3	$45
Prototype PCB	$7	2-3	$21
Wooden Pallets	$10	3	$30
Building Materials (Hardware, etc.)	$20-50	Various	$50
Lock Actuators	$30	2	$60
ATMega328 Microcontroller	$5-15	1	$15
Web Hosting Service	$20	1	$20
Electrical components (Wires, Resistors, Capacitors, Etc)	Various	Various	$40
Electronic Device for App	$100	1	$100

Table 52 Projected Component Costs and Quantities

Final Component Costs			
Item	Cost	Quantity	Total Cost
50W Solar Panel	$132.03	1	$132.03
12 V Battery	$64.99	1	$64.99
UNO Project Starter Kit	$34.99	1	$34.99
ATMega328P w/ Bootlader 3pc	$13.98	1	$13.98
Keypad	$4.87	2	$9.75
ESP8266 12E Node MCU	$8.39	1	$8.39
ESP8266 12E	$5.89	1	$5.89
RTC	$5.99	1	$5.99
12V Lock Actuator (2pc)	$9.57	1	$9.57
ACS712 Current Sensor Module	$7.20	2	$14.40
Arduino NANO	$7.99	1	$7.99

Table 53 Final Component Costs and Quantities

11.2 Legalities

EzRack is a smart rack but it has similar features to bike sharing. EzRack offer the convenience of a smart bike rack to students but students must use their own personal bicycle. Here liability and damages must be taken into account to safeguard the institution. The following list a few suggestions that can help limit the liability towards the institution:

1. Waivers and user agreement must be signed before use. This pertain to institution not reliable for any theft and damages to bicycles.

2. Educate users on safe practices and operation of bike rack. This pertain to keep safe from electrical equipment and lock mechanism.

3. Maintaining system to safety standards.

Appendix A – Copyright Permissions

Acknowledge | Permission Request Accepted

Electronicslovers <abid@electronicslovers.com>
6:14 AM

To: Trung Luu

Dear Trung Luu,

Thank you for writing to us about Permission Request. Hopefully, you will Cite our official Website on your project Report once you Complete Your Project ✅ . We wishing you Good Luck for Your project ✅

Thank You,
Electronicslovers

On 26-Apr-2018, at 2:51 PM, Trung Luu <tluum89@gmail.com> wrote:

> Hello, my name is Trung Luu. I am a senior computer engineering student at University of Central Florida. My team and I are currently working on a smart bike rack with off-grid power system. I am browsing through the internet in researching for a charger controller and came across your design and we would like to have your permission to use it has our reference. The project is for education purpose only.
>
> Best regards,

RE: Permission Request

Johannes Boonstra <jboonstra@victronenergy.com>
1:29 PM

To: Trung Luu

Hello,

Yes, there is permission to use these.

Best regards
Met vriendelijke groet

Johannes Boonstra
Sales Manager
jboonstra@victronenergy.com
Tel: +31 651328860

victron energy BLUE POWER

Van: Trung Luu <tluum89@gmail.com>
Verzonden: dinsdag 24 april 2018 04:56
Aan: Johannes Boonstra <jboonstra@victronenergy.com>
Onderwerp: Permission Request

Hello,

My name is Trung Luu and I am a senior computer engineering student at University of Central Florida. In researching for solar charge controller for my group project regarding smart bike rack. I came across "White paper Which solar charge controller PWM or MPPT?".PDF which I have learn a lot from and would like to have permission to use figures, tables, and knowledge from this file to write my report. The pdf is attached for your convenience. Thank you.

Best regards,

Asking permission to use picture

TL Trung Luu <tluum89@gmail.com>
4:50 PM

To: info@juicebikeshare.com

To Whom It May Concern,

Hello, my name is Trung Luu. I am a senior computer engineering student at University of Central Florida. My team and I are currently working on a smart bike rack with off-grid power system. I am browsing through the internet in researching through existing similar existing product. I came across a picture of Juice bike and would like to use it in report. The project is for education purpose only. Thank you for your time.

Here is the picture I would like to use.

Best regards,

Requesting Permission to use picture

TL Trung Luu <tluum89@gmail.com>
10:31 PM

To: bikeshare@cityofaventura.com

To whom it may concern,

Hello, my name is Trung Luu. I am a senior computer engineering student at University of Central Florida. My team and I are currently working on a smart bike rack with off-grid power system. I am browsing through the internet in researching through existing similar existing product. I came across a picture of Aventura BCycle and would like to use it in report. I would like to have your permission to do so. The project is for education purpose only. Thank you for your time.

Here is the picture I would like to use.

Best regards,

Appendix B – References

[1] "SGA Bike N Gold." *SGA Bike N Gold » Student Union » UCF*, studentunion.ucf.edu/bike-share.

[2] "International." *IEC - About IEC > What We Do > Technology Sectors > Renewable Energies > Solar Power*, www.iec.ch/renewables/solar_power.htm.

[3] "Solar Charge Controller Certifications: IEC 62509 and IEC 62093." *Sinovoltaics - Your Solar Supply Network*, sinovoltaics.com/learning-center/certifications/iec-62509-and-iec-62093/.

[TL1] "What Is Backend Testing?" *Meet Guru99 - Free Training Tutorials & Video for IT Courses*, www.guru99.com/what-is-backend-testing.html.

[TL2] "Common PIC Problems." *Common PIC Problems | Microchip*, www.microchip.com/forums/m628745.aspx.

[TL3] *Arduino - Troubleshooting*, www.arduino.cc/en/Guide/Troubleshooting.

[TL4] Instructables. "ARDUINO SOLAR CHARGE CONTROLLER (Version 2.0)." *Instructables.com*, Instructables, 13 Oct. 2017, www.instructables.com/id/ARDUINO-SOLAR-CHARGE-CONTROLLER-Version-20/.

[TL6] "MPPT Algorithm." *MATLAB & Simulink*, www.mathworks.com/discovery/mppt-algorithm.html.

[TL7] *Comparison of MPPT Algorithms for DC-DC Converters Based Photovoltaic Systems - IEEE Conference Publication*, ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6533431.

[TL8] Frenzel, Lou. "12 Wireless Options for IoT/M2M: Diversity or Dilemma?" *Electronic Design*, 12 Apr. 2018, www.electronicdesign.com/iot/12-wireless-options-iotm2m-diversity-or-dilemma.

[TL9] "Solar Charge Controller Basics." *Northern Arizona Wind & Sun*, www.solar-electric.com/learning-center/batteries-and-charging/solar-charge-controller-basics.html.