

# Project E-Clerk

## A Small Equipment Checkout System

### Project Plan

Team Number: SDMAY18-01

Client: Lee Harker

Advisers: Lee Harker

Anne Steenson — Project Manager, Software Architect, Full Stack Software Developer

DeQuan Burnside — Software Reporter, Backend Developer

Rex Borseth — Client Liaison, Embedded Systems Developer

Nicholas Parsons — Secretary, Computer Hardware Architect

George Zachariades — Treasurer, Systems Administrator

Sandra Sebastian — Hardware Reporter, Hardware Engineer

Team Email: [sdmay18-01@iastate.edu](mailto:sdmay18-01@iastate.edu)

Team Website: <https://sdmay18-01.sd.ece.iastate.edu>

Revised: December 6th 2017 / Version 3.0

# Table of Contents

Table of Contents	1
List of Figures	3
List of Tables	3
List of Symbols	3
List of Definitions	3
<b>1 Introduction</b>	<b>5</b>
1.1 Acknowledgement	5
1.2 Project statement	5
1.3 Operating Environment	6
1.4 Intended Users	7
1.5 Assumptions and Limitations	8
1.6 Expected End Product and Other Deliverables	8
<b>2 Proposed Approach and Statement of Work</b>	<b>9</b>
2.1 Functional requirements	9
2.2 Constraints considerations	11
2.2.1 Non-Functional Requirements	11
2.2.3 Standards	11
2.3 Technology considerations	12
2.4 Safety considerations	14
2.5 Previous Work and Literature	15
2.6 Possible Risks and Risk Management	16
2.7 Project Proposed Milestones and Evaluation Criteria	18
2.8 Project tracking procedures	19
2.9 Objective of the task	20
2.10 Task approach	20
2.11 Expected Results and Validation	
The end goal is a fully functioning, autonomous, checkout system. The user experience should be able to walk up to a monitor with keyboard and mouse, checkout or return equipment through the website, and the correct locker box(es) should open for the user to retrieve equipment from or to return equipment to.	24
<b>3 Estimated Resources and Project Timeline</b>	<b>25</b>
3.1 Personnel Effort Requirements	25
3.2 Other Resource Requirements	27
3.3 Financial Requirements	27
3.4 Project Timeline	31
4.1 Conclusion	32

4.2 References	33
4.3 Appendices	35

## List of Figures

- Figure 1: Macro-level System Diagram
- Figure 2: Software Stack
- Figure 3: MySQL Database Schema
- Figure 4: Semester 1 Gantt Chart
- Figure 5: Semester 2 Gantt Chart
- Figure 6: Model-View-Controller

## List of Tables

- Table 1: Major Tasks
- Table 2: E-Clerk Hardware Equipment Costs

## List of Symbols

## List of Definitions

- **Admin:** A nickname for administrators
- **C.R.U.D.:** An acronym for Create Read Update Delete. This acronym is typically used to describe software functionality.
- **Code Pair:** A code pair is when two developers work on the same piece of code together. Developers often do this when training someone.
- **E-Clerk:** The name for the project. A short, memorable name for the small equipment checkout system. See the Introduction section for more details regarding the system and its use.
- **ETG:** An acronym for Electronics and Technology Group. See more at reference #22
- **Identity Provider:** An identity provider is a system entity that creates, maintains, and manages identity information while providing authentication services to applications.
- **ISU:** An acronym for Iowa State University
- **Iowa State:** An abbreviation for Iowa State University
- **MVC:** An acronym for Model-View-Controller. MVC is a popular software architectural pattern for implementing user interfaces on components. See more at reference #21
- **1-Wire® Device:** 1-Wire is a device communications bus system that provides low-speed data, signaling, and power over a single conductor. It is designed by the Dallas Semiconductor Corp and trademarked by Maxim.
- **OWFS:** An acronym for 1-Wire Filesystem. See more at reference #23.

- **Responsive Website:** A website that changes user interface layout based on the device's screen size.
- **REST API:** A Representational State Transfer Application Programming Interface is a common design pattern used to define how clients will communicate with a server. See more at reference #18.
- **Shibboleth:** Shibboleth is a single sign-on system for computer networks and the internet. It allows people to sign in using just one identity to various systems run by federations of different organizations or institutions.
- **SSO:** An acronym for single sign-on.
- **24/7:** A shorthand name for twenty-four hours, seven days a week

# 1 Introduction

## 1.1 Acknowledgement

The Project E-Clerk team would like to kindly thank Iowa State University and the College of Electrical and Computer Engineering for promoting student professional experience and sanctioning this cross-disciplinary project. As students, the team appreciates the university for prioritizing outstanding issues with the current small equipment checkout system from the Electronics and Technology Group (ETG). Project E-Clerk's team would also like to thank ETG for their mentorship in developing the team's professional skills, for allocating human and financial resources, and for sharing their workspace with a handful of engineering students.

## 1.2 Project statement

Due to immersive course work, Electrical and Computer Engineering students will utilize various electronic devices throughout their studies at Iowa State University (ISU). Most of this equipment must be checked out from the university's Electronics and Technology Group (ETG). However, due to limited resources, ETG currently does not have an automated process for facilitating equipment checkout.

Present checkouts require faculty to spend copious amounts of time and tedious attention maintaining inventory and manually holding students accountable for equipment returns and damaged goods. Not only is the process inefficient, but the process also limits students to ETG's business hours. Students find it difficult to carve out time from their busy class schedules to visit ETG during their checkout operating hours.

Research has shown lower student to teacher ratios improve teaching quality and learners' understanding. With that, teachers' and faculty members' time is proven to be more effective when spent with students. Equipment checkout wastes faculty resources on mundane secretarial tasks that would be better utilized working with students' academic needs. In order to not only improve the use of faculty resources but also enhance students' academic experience, the team will build a virtual and physical solution to regulate small equipment checkout from ETG.

Project E-Clerk will enable students to checkout equipment without being confined to the ETG's hours and also address current checkout process inefficiencies from an administrative standpoint. E-Clerk will involve both a hardware and software component.

In terms of software, E-Clerk will have two types of end-users: Administrators and Students. For both groups, E-Clerk will allow users to authenticate with their ISU Shibboleth (a.k.a. NetId) credentials. Right now, there are various systems at ISU that allow students and faculty to authenticate by entering their NetId and password. E-Clerk will replicate and extend those existing software procedures for retrieving user information.

Although, software is hardly effective without a user interface. Therefore, the team will build a responsive website for both types of end users to interact with. This website will be available to students on a Raspberry Pi attached to a set of lockers. Furthermore, the website will also be available to administrators through any web client connected to Iowa State University's private network.

E-Clerk will enable administrators to manage checkout records, equipment inventory, student privileges and effortlessly maintain a secure system. Students will be able to self-facilitate equipment checkout, view privilege status, and be notified via email of pending return dates.

In terms of hardware, the team will implement a replicable hardware system that securely stores equipment and regulates the checkout process. Using metal lockers, provided by ETG, E-Clerk manage locker opening based on checkout requests and also alert users of system misuse. Each locker box will be cut and framed with a transparent window and be interactive with led lights to permit users to confirm they are checking out the correct equipment when they go to select an item.

### 1.3 Operating Environment

The expected environment for the small equipment checkout system would be located outside the Electronic and Technology Group (ETG) in Coover Hall. This means that the system will be kept in a clean and room temperature location. Knowing this, there are not many environmental obstacles. The possibility of this system over heating is unlikely will be taken care of in testing.

Project E-Clerk's software component will run on an Iowa State University virtual machine maintained by ETG. The locker, keyboard, mouse, and monitor will be exposed to both types of users (student and administrators). Administrators and students will be frequently using the system so normal wear and tear will occur.

Some malicious students may have an incentive to damage the lockers. To address misuse, the software will detect and report communication errors between the OWFS server and the 1-Wire® Device System. Issues with the keyboard, mouse, and monitor can be either reported verbally to ETG staff members or students can file an issue report on the website.

## 1.4 Intended Users

Project E-Clerk's mission is to modernize students' and administrators' equipment checkout experience. Therefore, E-Clerk will have two types of users: students and administrators. The students will be enrolled at Iowa State University (ISU). Since they are enrolled at ISU, students will have authentication credentials to ISU's Shibboleth Identity Provider. Administrators will likely be individuals who are employed at ISU under the ETG department with few rare exceptions for student staff. Therefore, administrators will also have authenticated access to E-Clerk's website.

Currently located in Coover, ETG is open Monday through Thursday from 8 AM to 8 PM and on Friday from 8 AM to 5 PM. While ETG's operating hours seem long compared to normal business hours, one should also consider that most students take around 15 credits per semester. Students spend most of their time on weekdays attending classes. From market research, E-Clerk's team has found that a majority of potential users need to checkout equipment late at night or on weekends. Moreover, ETG's operating hours are not optimal for student's availability. In order to best serve student's, Project E-Clerk will be available 24 hours a day, 7 days a week.

ETG staff (synonymous with E-Clerk's administrators) currently support requests for wiki creations, website spaces, operating system rebuilds, virtual machine instantiations, equipment checkout, and approving administrator privileges. Between helping students utilize ETG's various services and maintaining equipment operations, staff members have very little free time to micromanage overdue equipment checkouts, keep track of equipment malfunctions, or service student requests for trivial items such as a screwdriver or Raspberry Pi. Project E-Clerk's mission will enable students to help themselves, and reduce overhead for administrators.



## 1.5 Assumptions and Limitations

### Assumptions:

- Every ISU student and administrator will have authentication credentials to Iowa State University's Identity Provider, Shibboleth.
- Project E-Clerk's website will not be accessible outside of Iowa State University's intranet.
- After E-Clerk's team has graduated, Iowa State University's Electronics and Technology Group will be in charge of maintaining and extending Project E-Clerk.
- ETG expects Project E-Clerk's website to utilize Iowa State University's universal web theme.
- Students will use a keyboard, mouse, and monitor adjacent to the locker unit system to interact with E-Clerk.
- Administrators will be able to access E-Clerk through the physical display system attached to the locker and through their own computers connected to ISU's intranet.
- The system will be powered on and active for all 24 hours of the day for all 7 days of the week unless the system is offline for maintenance or updates.
- The end user can read a screen displayed in the English language.
- The system will be kept in a single location in Coover Hall.

### Limitations:

- The physical system will only have up to 12 volts to power both the Raspberry Pi and the 1-Wire® System.
- The cost to produce each locker will be ~ \$12.06
- People not affiliated with Iowa State University will not be able to use the system.
- The website will not be available outside of Iowa State University's intranet.
- The project will utilize a Dallas 1-wire system (advisor recommendation).
- The project will use a raspberry pi device.
- The system must use the metal mailbox unit provided by the ETG.

## 1.6 Expected End Product and Other Deliverables

The final product delivery will be split into three major milestones - a prototype, a minimum viable product and a final product. The prototype will be delivered at the end of the first semester. The Minimal Viable Product (MVP) will be delivered before Spring Break. The final product will be delivered at the end of second semester. A software manual and hardware manual will be delivered with the final product.

- Prototype - December 6th, 2017
  - The prototype will serve as a proof of concept and will consist of a website hosted on a linux web server. Users can create a checkout record on the website. The website will persist records to a MySQL database and send a signal to the 1-Wire® System to open up the locker. The prototype will demonstrate connections between frontend software, backend software, the MySQL database and the 1-Wire® System.
- Minimal Viable Product (MVP) - March 9th, 2018
  - The MVP will be a functional system without the bells and whistles (minimum viable product). The team will deliver functional software and hardware with the given mailbox unit creating an inventory checkout system.
- E-Clerk - April 27, 2018
  - E-Clerk is a small equipment checkout system with 3 lockers that are controlled by a software component paired with a website as a user interface. The product must be maintained by the ETG past the point of the final product delivery.
    - Design documents - This package will include hardware and software block diagrams.
    - Code - Both frontend and backend code will be made available to ETG on ISU's GitLab.
    - Software Manual - This manual will serve as a guide to E-Clerk's software implementation. This manual will describe design patterns, testing methods, third-party libraries, and instructions for maintenance and extension.
    - Hardware Manual - This manual will serve as a guide to E-Clerk's hardware implementation. This manual will contain a list of hardware components and instructions for building individual locker boxes.

## 2 Proposed Approach and Statement of Work

### 2.1 Functional requirements

The system's functional requirements are split into two user groups - students and admins. The functional requirements below cover a single student's typical use of the system. The expected use case is to login to the system, select an equipment item to check out or check in, to retrieve or return an item, then to logout of the system. Based on this, the functional requirements for students are:

- Log in and out
- View available equipment
- Select an equipment item to checkout
- Determine checkout duration
- View currently checked out equipment (personal)
- Return equipment, close checkout record
- Report system misuse or broken items
- Receive email reminders

The admin use cases are much more broad because admins will need control over all of the aspects of the system. A general admin use case would be to login to the system, perform an action that modifies the state of a locker, a user, or an item within the system, and then logout of the system. The system will also be able to send email status reports of lockers, users, or items. Warnings of improper use of the system will also be reported in this way. Based on this, the requirements for an admin are:

- Log in and out
- C.R.U.D. available equipment
- Determine max checkout durations for items
- C.R.U.D. student users, records, and user privileges
- Receive email status reports
- Create, update, remove lockers
- Add new Administrators

## 2.2 Constraints considerations

### 2.2.1 Non-Functional Requirements

Scalability - E-Clerk must be scalable in case the ETG decides to expand the checkout-system. This will be accomplished by documenting the process and designing a modular system.

Availability - E-Clerk needs to be available 24/7.

Reliability / Recoverability - E-Clerk must be able to recover from single locker, power, or software outage because ETG will not be able to service or troubleshoot outside of business hours on business days. E-Clerk must have built in recovery functions that still server or notify users in an erroneous state.

Maintainability - E-Clerk needs to be maintainable past the final product delivery. The ETG must have the resources and information to keep the system functional and behaving as designed. This will be accomplished with detailed documentation and well designed hardware / software solutions.

Security - The checkout unit must be secure yet accessible as the system will be active during non-business hours. This requirement can be fulfilled with considerate software design.

Data Integrity - The records of which equipment has been checked out and who has checked it out should only be modified by administrators. This is achievable by preventing direct access to the data by non-administrators.

Usability - E-Clerk will be accessible by any student or administrator. E-Clerk will be a useful because students will be able to checkout small equipment at their convenience and administrators can spend little time managing checkouts. The students will use a keyboard, mouse, and monitor hooked up to a Raspberry Pi to easily sign in to E-Clerk. E-Clerk will also accommodate for disabled users by making website disability accessible.

### 2.2.3 Standards

There are a few standards that the project implementation will be complying to. Most of these standards are pre-built into the servers such as Http with Node.js server and MySQL connect with Sequelize. They are somewhat trivial. The project implementation

will be conforming to the 1-Wire® protocols for 1-Wire® devices and to the OWFS (1-Wire Filesystem) standards. The project will also be utilizing OWFS to communicate with the world outside of the physical 1-Wire® system. OWFS creates a virtual file system where each folder and file relates information about the 1-Wire® devices connected. The file system can be effectively browsed by making a connection to a OWFS server that runs alongside the filesystem itself. This server can be polled to get information about the files in the virtual filesystem. The server will also be used to send command directly to the devices on the 1-Wire® system.

## 2.3 Technology considerations

Before choosing what method to implement for the project, it was important to think about why it would be the best solution. After deliberation and research, the decision was to use the Raspberry Pi as the connection between the lockers because it already has an I2C protocol built into it. This avoids many difficulties when establishing the connections between the PI and the locker boxes.

Mr. Harker provided a Raspberry Pi Touch Screen. However, Shibboleth requires users to type in their username and password. Students will also need to report details about issues on the website. Unfortunately, the provided touch screen is 7” and is, therefore, not user-friendly for typing.

There is a larger, 10” Raspberry Pi Touch Screen available, but it costs around \$108.00 (Reference #39). The E-Clerk team has found \$27.00 computer monitors, \$7 keyboards, and \$4 mouses. Consequently, the team decided to go with a computer monitor, keyboard, and mouse rather than a larger Raspberry Pi Touch Screen in order to save money.

Mr. Harker did not specify what type of software should facilitate E-Clerk’s functionality. Because E-Clerk’s administrators want to be able to access the software from other devices, besides the raspberry pi connected to the lockers, the team decided E-Clerk’s software should be available on a website rather than a mobile application or desktop application. A website can serve across all platforms and is, therefore, not confined to the raspberry pi.

The proposed product involves using the Raspberry Pi to connect to E-Clerk’s website. After a lot of testing, the team found a browser that would meet loading time expectations: Chromium. The product will utilize an internal VM within Iowa State’s network to host a web server for the web application. This ensures that in case the

Raspberry Pi malfunctions, the system can support a backup that is configured and everything can be up and running in little time. This also means that the system is vulnerable in two instances rather than just one, but security precautions will be ensure system safety.

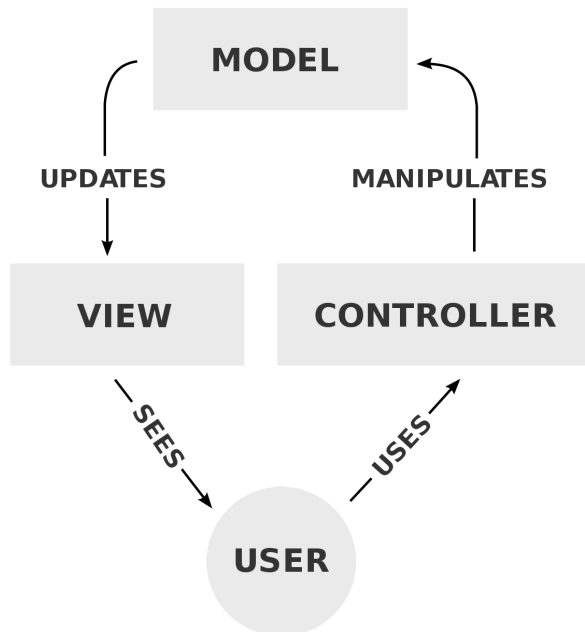


Figure 6: Model-View-Controller (Reference #24)

The MVC pattern (Reference #21), as seen in Figure 6: is popular for implementing user interfaces. MVC supports rapid and parallel development because developers can be working on either the model, view or controller at the same time. MVC allows the applications to switch views depending on the user. E-Clerk will use this functionality when serving either students or administrators. E-Clerk also works synergically with asynchronous development because both requests from the user to the controller and responses from the model to the view are usually in the form of HTTP.

However, MVC applications can be very complex to design and develop. There are a lot of moving parts and therefore MVC may not be optimal for small applications. In E-Clerk's instance, the team argues that MVC is the proper software architecture pattern because MVC allows future developers to easily extend the product's functionality and allows future developers to interchange layers (i.e. the model, the view, or the controller) without disrupting legacy functionality.

JavaScript is the chosen language for a various reasons. Annie has built industrial JavaScript applications through a previous internship. She can enforce industry standards

in E-Clerk's development. JavaScript is environment agnostic and, therefore, code can be reused on both the client and server side. Node.JS is not a framework, unlike Java Spring, Larval, or Django. Node.JS is an environment which means developers have more freedom in the implementation design.

Although, JavaScript is a non-blocking language which means that JavaScript code runs asynchronous functions. This can pose design issues when connecting a Node.JS server with two other servers (a MySQL server and OWFS server) to perform business logic. In order to handle various race conditions when pooling the two other servers, E-Clerk uses a JavaScript Promise (Reference #17) based methodology. This non-blocking/asynchronous feature is not present in other languages such as Java or Python. There this feature can be seen as a weakness.

Overall, this project has been given to us with a good design already in mind from the client, Lee Harker. The new modifications proposed seem right for completing the project on time and with high-quality.

## 2.4 Safety considerations

Project E-Clerk's team will use their individual personal computers to develop the software. The software will utilize ETG's virtual machine services and will, therefore, not have to physically construct a server rack. Overall, E-Clerk's software implementation poses little to none human-safety risks. Although cyber security attacks are very popular in the world. Project E-Clerk's team will take all necessary precautions to secure the software component.

Project E-Clerk will use a MySQL database to record student emails associated with virtual checkout records. Since ISU student emails are available to the public through the Iowa State University Student Directory (<https://www.info.iastate.edu/>), student's email are not considered confidential information. Although, the team does not wish for malicious students to alter checkout information stored in the database. All access to the database will be restricted to the IP addresses of the Virtual Machine (VM) and the Raspberry Pi only. The password will be encrypted in the configuration file in the server-side code.

Since Project E-Clerk will be connected to the ISU intranet, the team will ensure, only authenticated users will be able to access E-Clerk's website. The website will be hosted on an Iowa State University virtual machine. The virtual machine's passwords will be passed onto ETG and stored securely.

For the hardware team, much of what needs to be done are new skills that are based on using resources to guide the process. The chips that were ordered this semester were much too small to be placed on a breadboard alone. Given a breakout board, the team needed to solder chips and pins to prototype on a breadboard. Soldering can be a big safety concern considering the tool used reaches high enough temperatures to melt solder to metal.

One other concern on the hardware side would be the modifications the locker box will go through before installations can occur. Each of the units will be cut shaping a window to see inside. Glass will also be cut to fill the now cut out part of the unit doors. The entire process of how this will be done will be dangerous to attain the final look of the E-Clerk unit.

## 2.5 Previous Work and Literature

The software system will resemble the ETG's current in-house equipment checkout web application. The current system allows for administrative staff to browse inventory and manually checkout equipment to students. The application was built in PHP and AJAX which differs from the team's senior design project approach. The ETG's current system does not have a hardware component. It relies on the work of staff to fetch the items from storage and physically hand the item over to the student upon request. The current system requires much work for the ETG staff and it can only be of use to students during set hours on business days. The current in-house system only allows students to request equipment, not return equipment.

E-Clerk will allow students to fulfill their requests and retrieve the equipment themselves. The E-Clerk project will build upon the ideas of ETG's current checkout application, but it will implement a hardware component via the locker unit, and will remain active at all hours of the day.

The E-Clerk System will be similar to a couple self-automated locker systems. One system would be the Amazon Hub (Reference #40). This is a locker system that allows customers to retrieve their items at a convenient time and location of their choosing. The hub can be installed at any given location and used by anyone. The process goes, once an order is placed, a delivery service will bring the item to the location chosen by the customer. Before a customer can confirm package delivery spot, the unit is checked for space and whether it is in service at the moment.



The one who delivers the package will check in an item and select the size of the package. A locker compatible to fit the package will open and once shut again, the customer will receive a notification via email/text message that their package will be ready for pick-up with a unique code. All customer information is stored for future use and repercussions if a problem occurs. Customers are then sent a customer service survey about the system.

If a customer is not satisfied with an item, the system handles returns. The customer will check an item back into a locker after giving a brief reasoning for the return. The delivery service will be notified to come for pick up.

Issues with the Amazon Hub has been empty boxes being returned, barcode scanner and/or code input does not work, miscalculation on available space, multiple lockers opening for one code, and shipments going to the wrong unit location. Amazon offers customer support but can not fix any problems immediately from a remote location. Issues with orders coming into unit and people returning items resulting in no available space depending on the process orders.

Although, E-Clerk is similar to the Amazon Hub, E-Clerk will only have a designated amount of equipment to the system resulting in always available storage. In both systems, issues with lost, stolen, or broken items will occur. E-Clerk will have electric locks, LED lights, and a small window to see inside the units. The lights will be used for visibility purposes when users verify that the correct item is there. Like Amazon, E-Clerk's team needs to check for item verification constantly. By adding many prompts to students, the team believes most issues can be avoided.

## 2.6 Possible Risks and Risk Management

Project E-Clerk's website will be develop with an array of JavaScript libraries. Only the Software Architect, Annie Steenson, has experience with this software stack. Not only will the other team members have to learn how to use this software stack, but they will also have to learn JavaScript.

Currently, the main language Iowa State University teaches its students is Java, an Object Oriented Programming Language. JavaScript will have a very steep learning curve because it is a Functional Programming Language, very different compared to Java. JavaScript's steep learning curve will slow down E-Clerk's software production.

Because Annie is the only team member with JavaScript domain knowledge, the other team members will have to put in extra effort to learn so they can effectively contribute to

code production. Annie currently mentors students for Udacity and has experience teaching people about JavaScript. Although, it is not always feasible for Annie to constantly code pair. The steep learning curve will challenge other team members' ability to use online resources and be self-sufficient. Making other team members responsible for the majority of their own learning can be risky because others may not manage their time wisely or be motivated to self-teach.

The plan is to authenticate website users using Iowa State University's Identity Provider, Shibboleth. None of the team members have integrated Shibboleth into an application prior to this project. Shibboleth integration will also involve adding an Apache web server layer on top of the JavaScript stack. The team systems administrator, George Zachariades, has never implemented Apache with Node.JS and Shibboleth authentication before. Therefore, a learning curve will be involved when building out the authentication/authorization functionality.

The team may run into many unexpected barriers when integrating Shibboleth simply because none of the members have worked with such a system. The team will use other ISU examples as reference and reach out to Shibboleth administrators if necessary.

Concurrency issues may pose a threat to the security of the system. E-Clerk has to make sure that multiple students cannot access the system simultaneously. Students should not be able to open more than one browser tab or browser window and gain access the system with more than one session. Concurrency testing will prove to be a significant aspect of the system's security.

For the hardware team, many courses required for the Computer Engineering curriculum teaches basic circuit components and their functionality. The team lacks experience with the concept of integrating new and previously used components. For the hardware team, most of the information is needed to be researched in order to create the design of the system.

Project E-Clerk's hardware system will utilize a Dallas 1-Wire® System. None of the team members have used the 1-Wire® System beforehand. Therefore, the hardware team has needed to take time out to read documentation and experiment with prototypes to become acquainted with the technology.

The hardware team also struggles with the question of powering the system along with ensuring a plan that the system will be reliable for long term. Collectively, the team has never been given a project that required this type of obstacle to overcome. Through

classes, the team has been able to rely on given power supplies and no need for extended time effects on the circuits. The hardware team will need to use academic and human resources to get a better understanding of what to expect.

## 2.7 Project Proposed Milestones and Evaluation Criteria

- Prototype:
  - The prototype will test E-Clerk's implementation front to back. The website will be hosted on an ISU virtual machine. A user should be able to access the website from a computer connected to the ISU intranet. On the website, the user should be able to click a button to signal they want to checkout an item. The frontend will then send an HTTP request to the backend. The backend will then send a TCP request to the OWFS server turning the lock off.
- Minimal Viable Product:
  - The MVP will include a barebone (minimal CSS) website where students can create checkout records and retrieve their item from a locker box. Authentication should be working in the MVP version of E-Clerk. Students should also be able to log back into the website and return the item to a locker box.
- Alpha Testing
  - During Alpha Testing, E-Clerk's team and ETG will manually test and trial the Minimal Viable Product. The team will then take any feedback and bug reports and use that to further improve the system.
- Product Ready for Staging
  - The staging product must have a pristine user interface and fully functional student and administrative features. The staging product must have 3 functional locker boxes. The staging product should be viewable from an administrator's desktop.
- Beta Testing
  - During Beta Testing, the team will ask a group of students and ETG faculty members to trial the staged product. Testers will then provide feedback to E-Clerk's team.
- Completed Final Product (Production Ready) Package (E-Clerk System, Software Manual, Hardware Manual)
  - This product will be only slightly different from the staging product. Lessons learned from Beta testing will be implemented in the

production-level system. This product must have the correct security features such as SSL, form validation, software logging, error handling, and backup scripts. The final product will also include software and hardware manuals.

## 2.8 Project tracking procedures

The team will present weekly status reports from every member with detailed descriptions of what each individual worked on during that seven day period. All team members will have assigned tasks that come with a delivery date and time. Deadlines are not always met in industry and in academia. In most cases, issues arise and new factors must be taken into consideration. Therefore, task deadlines will be tentative, but task completion is still a priority. Team members will hold one another accountable for their contributions to ensure that one person is not completing a large majority of the workload.

### **Milestones**

Prototype - Established connection from software and hardware components. This is critical to the functionality of the end product.

Checkpoint: First Semester Demo

MVP - The MVP product will extend the prototype to a version much closer to the final product. The focus will be placed on checking out and returning items.

Checkpoint: 40% unit test code coverage

Alpha Testing - The team and client will test the system and its use. This will ensure that we are building the right product for the ETG and for students.

Checkpoint: Every team member must test the system and provide qualitative feedback. The client Lee Harker must also test the system and provide qualitative feedback.

Staging - The product will be staged for beta testing. This includes visual design improvements, error or failure handling, and product setup for outside use.

Checkpoint: 60% unit test code coverage

Beta Testing - The product will be beta tested by other engineering students and ETG staff members. The feedback provided will drive final product adjustments or additions.

Checkpoint: Five students and two staff members have tested the system and provided qualitative feedback to the team.

Final Product - This final product will include documentation, code, and system demonstration to ETG staff and students.

Checkpoint: 95% unit test code coverage and successful project demonstration.

### **Progress Tracking Measures**

Weekly meetings with the client - The team will conduct weekly status meetings with the client / adviser. At these meetings, individual team members will come prepared with clarification questions, assumptions, or challenges to present to the client / adviser. The adviser will do their best to provide new insight for the team to overcome any current issues. These meetings provide a foundation for client communication.

Internal Peer Review - At the end of every month (until May), individual team members will spend 10 minutes reviewing their individual progress with the project manager, Annie Steenson, to discuss room for improvement. This will provide a foundation for accountability and personal improvement amongst individual team members which will translate to team improvement and project progress.

## **2.9 Objective of the task**

The objective of the project is to design a product that will enable students to checkout small equipment from the ETG. The product will be a mailbox available to them 24/7. The team goal is to deliver a fully functioning ETG small equipment checkout system by the end of the spring 2018 semester. This includes at least 3 working mailboxes, a working kiosk, and a database that will be able to pull a students info from Iowa State University's network to create an account for students when needed. Also included is the ability for an administrator to make any necessary changes to the checkout system.

## **2.10 Task approach**

At the beginning of the project the client laid out his plan for the project. This involved using a large mail locker box which is to hold all of the equipment and the an idea of using the dallas 1-Wire® system to control the lockers. To control the 1-Wire® system the client gave us a Raspberry Pi to drive the 1-Wire® system. One of the team members is experienced in web development so a web app will be the main piece of software. To store information a standard MySQL database will be used. Once all of the components were decided on the macro-level architecture shown in figure <insert number> below was created.

Figure 1 shows a macro-level overview of the E-Clerk's architecture.

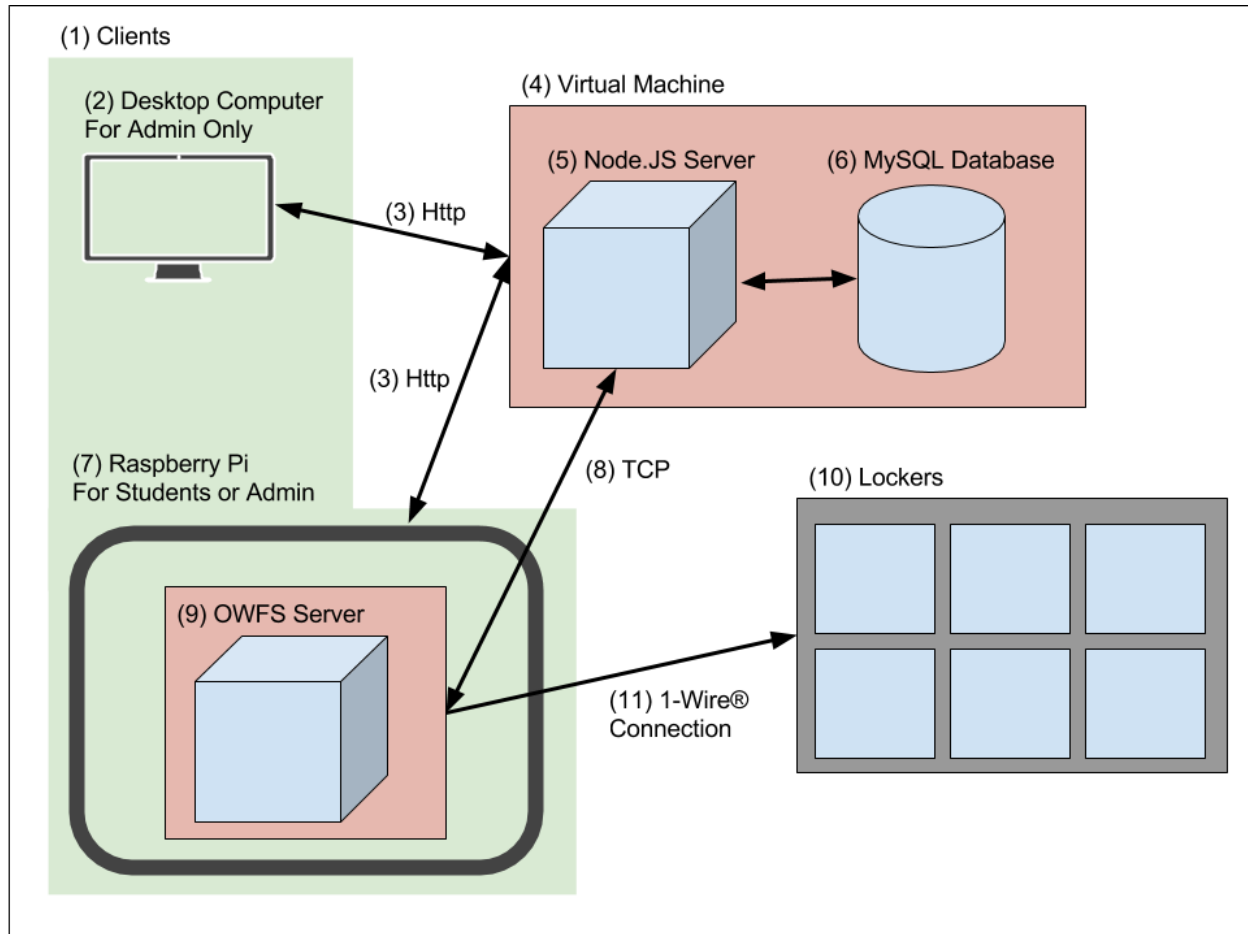


Figure 1: Macro-level System Diagram

This solution is broken into 3 main components: Clients(1) Virtual Machine (4) and Lockers (10).

A Node.JS server (5) will be hosted on an ISU virtual machine (4). The client machines (2 or 7) will connect to a Node.JS server (5) which will serve a web page based on the user type. There are two types of users Admins (2 or 7) and Students (7). Admins will be able to access admin privileges to modify the state of components in the lockers, change student account statuses, and other necessary administrative actions. Students will use a Raspberry Pi (7) to browse E-Clerk's website. After a login, students will be able to checkout and checkin equipment. The Raspberry Pi (7) will also run OWFS (9) which will use the 1-Wire® system (11) to control the Lockers (10).

The Virtual Machine (4) will run the Node.JS (5) server and the MySQL Database (6). The Node.JS server be accessed by the clients through an http connection (3). The Node.JS server will also pull information from the MySQL Database (6) and will control the lockers through a TCP connection (8) to the OWFS server (9) on the Raspberry Pi (7). The Lockers (10) will be controlled through OWFS which will manage all signals to controlling the lockers.



Figure 2: Software Stack

As seen in Figure 2, many packages and frameworks will be used to create the website. Node.JS runs the server and will be running a React framework. The React framework is used to make the pages dynamic so interaction within a page is able to be done without reloading the page or loading new pages. Sequelize will be used to connect to and retrieve information from the MySQL database. Shibboleth will be used to authenticate users that

are using the system. Jest will be used to test the pages and code. GitLab will be used for version control and for code management.

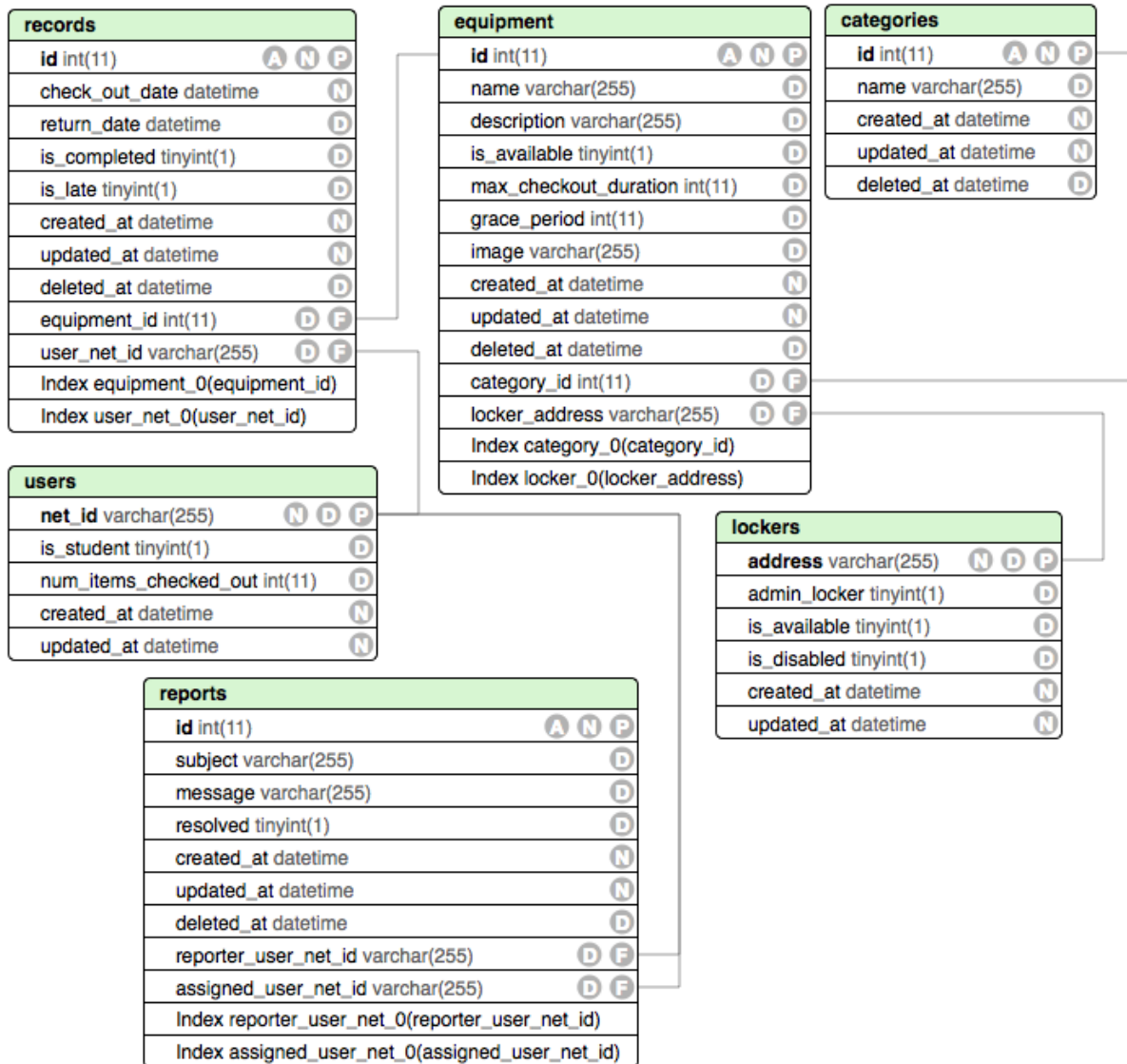


Figure 3: MySQL Database Schema

As seen in Figure 3, the database has 4 major tables and 2 minor tables. The major tables are the users, reports, records and equipment which has 2 minor tables lockers and categories. Each of the major tables contains vastly varied information that is key to the system working. The users field contains all the information of a user that is registered into the system and is a field in the records and reports tables. Reports will be generated when a user performs an action that necessitates a report to be generated and has two fields that are users. A record will be created for each item that is checked out from the



system and is used to keep track of who has which equipment checked out. Each record has an equipment and a user associated with it. Equipment will hold all the information on a specific piece of equipment. The equipment will have a locker and a category associated with it. The locker holds all the information of the locker and the category will help us sort equipment into relevant fields.

For the hardware a 1-Wire® system will be implemented. OWFS will be used to control the 1-Wire® system from the Node.JS. For the lockers LED's will be used to light the boxes and an electromagnetic lock will be used to secure the door.

## 2.11 Expected Results and Validation

The end goal is a fully functioning, autonomous, checkout system. The user experience should be able to walk up to a monitor with keyboard and mouse, checkout or return equipment through the website, and the correct locker box(es) should open for the user to retrieve equipment from or to return equipment to.

The team has allocated a good portion of time to test the system. This will include unit testing starting with the prototype. Once the hardware team begins adding additional locks to the circuit, the 1-Wire® master device will need to identify individual locks to engage. The logic behind this will then be implemented.

Testing will be done by repeating simulation tests and seeing how the system would react. By doing this, any type of errors or problems will appear and fixes can be made. The testing will continue for as many test cases the team thinks of before confirming the system is fully functioning.

The fully functioning product will need to be able to unlock/lock all mailboxes, turn on/off LEDs on all the mailboxes, without any errors.

The software tests will be categorized as either frontend, backend or integration tests., it will be broken down into two major components.

In the frontend, the team will be doing visual inspection of elements in the User Interface and functional testing to ensure a quality cross-platform, cross-browser user experience. In the backend, there is the MySQL database, Node.JS webserver, and OWFS (1-Wire Filesystem) server. The web servers will control the user interface and control the user experience. The backend tests will be conducted using Jest. Jest will allow developers to unit test all sections of the backend code, measure code coverage and mock API calls.

OWFS interacts with the hardware directly and runs a TCP server for connecting to a virtual file system that represents the devices connected to the system. The OWFS server will be connected to JavaScript code through an NPM module titled 'owfs.js'. All testing that needs to be done for OWFS will be completed on the JavaScript side with the other backend tests.

The locker boxes and associated electronics are much simpler to test. The team will start by testing the circuit on breadboard. Once a working prototype is finished, the next step is to expand it to multiple boxes. This way functionality of the circuits and the behavior can be tested. Stress testing the circuits is unnecessary as only one box can be activated at a time.

## 3 Estimated Resources and Project Timeline

### 3.1 Personnel Effort Requirements

As seen below, Table 1 is a table of all the major tasks that need to be done in order to complete the project. Implementation and documentation will take up the most time. The total estimated time to complete this project is 820 hours. The team has currently put in 380 hours combined. Now that we have climbed over various learning curves and fine-tuned details, the rest of implementation should be smooth sailing

#### Major Tasks:

Task	Description	Estimated Time
Research 1-Wire®	Research needed to learn how to use the 1-Wire® connection with the Raspberry PI.	20 hours
Setup VM	Setup the Virtual Machine, install all necessary applications with configurations. (Node.js, Apache, Shibboleth)	40 hours
Design Circuit Diagram	Design circuit diagram on ModelSim to be used at each locker box.	40 hours
Wire/Cut mailboxes	Team will need to cut a rectangular opening on all mailboxes and wire them to be connected to	30 hours

	the 1-Wire® device and a power source.	
Research Software Frontend Architecture	Since the team is using React, research needed to learning how to properly design state management. Since the application is cross-platform, research is needed to learn how to properly design a responsive application.	20 hours
Research Software Backend Architecture	The team will need to research various REST API frameworks and object relational mapping libraries.	10 hours
Setup Software Development Environment	Developers will have to get acquainted with various development tools such as: WebStorm, SequelPro/HeidiSQL, Postman and Git.	10 hours
Implement Software Frontend Business Logic	This task will require developers to utilize various libraries and debug unwanted behavior.	90 hours
Integrate Software Frontend Styling	This task will require developers to apply responsive web design skills and debug ambiguous CSS layouts.	90 hours
Implement Software Backend Rest API	This task will require developers to integrate MySQL with a Node.JS server.	90 hours
Implement Hardware	Team members will need to construct and wire at least 3 locker boxes.	40 hours
Test Software	This task will require developers to consider edge cases, error handling, and user misuse in order to fully test the code base.	60 hours
Test Hardware	This task will require engineers to test edge cases and ensure safety precautions are taken, and that the system does not fail.	40 hours
Documenting Software	Team members need to document all software code, architecture, and design patterns used.	120 hours
Documenting Hardware	Team members need to document all hardware parts, instructions for construction, and protocols.	120 hours

Table 1: Major Tasks

## 3.2 Other Resource Requirements

There are a few other resources that are used on both the software and hardware sides that are required to complete these tasks. To develop the code an IDE is needed like Webstorm by JetBrains. Webstorm does not place many restrictions on how software that is developed through it is used. The software side will also require a physical machine to run the virtual machine that will be running node.js and the sql database. However, this could be a remote server and does not need to be an on site device. There is also a few different options for an OS on the Raspberry Pi. Noobs is what the Raspberry Pi ships with and will work fine.

There are a few required hardware resources. Because of the need to create custom circuits and PCB's a software to do this is required. Multisim has been used to create the custom circuits and ultiboard will be used to create the custom PCB's. Both of these programs are paid programs and Iowa State University's licences are being used. A soldering iron and solder will be used to attach all of the components to the custom PCB's.

## 3.3 Financial Requirements

For the hardware team, there is a financial responsibility to make the client aware of what the expected cost of the system will be. This is an ethical code that, as engineers, the team will follow in being truthful so the client will know if the funds will be available to complete the project asked of.

As a team, multiple components have been identified as a need for this project. The basics for each locker include: 12V electric locks, LED lights, 1-Wire® slave devices, MOSFET switch, various resistors, an operational amplifier, and lengthy wires to accommodate the locker location. As singular extra costs, the team will need to add the price of a raspberry pi, a computer monitor to display the website, a 1-Wire® Master Device, AC/DC transformer, HDMI cord, ethernet cord, micro-USB cable with wall outlet, keyboard, and mouse for navigation.

The resources used to find these approximate costs of each item came from all different companies. The process the team went about searching for items started with looking at the Iowa State University surplus website. From there, the focus was placed on looking for low cost items that will be sufficient for the system.

Some items' costs are estimations due to market variations. The hardware components' prices will vary depending on necessary wire lengths and vendors' required minimum order quantity. The total estimated cost for E-Clerk is ~ \$627.82. The costs of each are listed below in Table 2.

Reference Number	Item	Cost per Unit x Number of Units	Total Cost
1	12V Electric Lock	\$4.00 x 34	~ \$136.00
2	LED light	\$0.03 x 34	~ \$1.02
3	Compression Spring	\$0.75 x 34	~ \$25.60
4	1-Wire® Slave Device (One Output - DS2406+-ND)	\$3.90 x 34	~ \$132.60
5	MOSFET Switch	\$0.87 x 34	~ \$29.58
6	Various Resistors	~ \$2.07 x 1	~ \$2.07
7	Operational Amplifier (497-1580-6-ND)	\$0.01 x 34	~ \$.34
8	Lengthy Wires (1528-1161-ND)	~ \$3.95 x 1	~ \$3.95
9	Raspberry Pi 3 Model B	\$35.00 x 1	~ \$35.00
10	Computer Monitor	(\$ 27.00) x 1	~ \$27.00
11	1-Wire® Master Device (DS2482S-100+T&RCT-ND)	\$1.68 x 1	~ \$1.68
12	Logic Level Converter	\$.60 x 1	~ \$.60
13	HDMI Cord	(\$3.99) x 1	~ \$3.99
14	Ethernet Cord	(\$4.95) x 1	~ \$4.95
15	Micro-USB with a wall outlet	(\$4.89) x 1	~ \$4.89
16	Keyboard	\$6.98	~ \$6.98
17	Mouse	\$3.90	~ \$3.90
18	PCB	\$6.08 x 34	~\$206.72

Table 2: E-Clerk Hardware Equipment Costs

The project requires leveraging a large amount of third-party software. Much of this software is available for use free of charge through open source licenses. An open source license is one that allows for the full and free use of the software. All of the software

utilized is either open source or have a free version. The team is using a student license or an Iowa State University license for all software tools that is non-free license. The type of license and cost of each is listed below in Table 3: E-Clerk Software Equipment Costs.

Reference Number	Software	Type of License	Total Cost
18	Apache HTTP server	Open-source	\$0.00
19	Bable	MIT License (Open-source)	\$0.00
20	Bootstrap	MIT License (Open-source)	\$0.00
21	ES6	MIT License (Open-source)	\$0.00
22	ESDoc	MIT License (Open-source)	\$0.00
23	ESlint	MIT License (Open-source)	\$0.00
24	Express	MIT License (Open-source)	\$0.00
25	GitLab	\$40/year (using ISU License)	\$40.00
26	Jest	MIT License (Open-source)	\$0.00
27	Moment.js	Open source	\$0.00
28	MySQL	GPL (Open-source)	\$0.00
29	Node.js	Open-source	\$0.00
30	NPM	Open-source	\$0.00
31	OWFS	Open-source	\$0.00
32	React	MIT License (Open-source)	\$0.00
33	Redux	MIT License (Open-source)	\$0.00
34	Sass	Open-source	\$0.00
35	Sequelize	MIT License (Open-source)	\$0.00
36	Shibboleth	Open-source	\$0.00
37	StrongLoop	Open-source	\$0.00
38	Webpack	MIT License (Open-source)	\$0.00

Table 3: E-Clerk Software Equipment Costs

### 3.4 Project Timeline

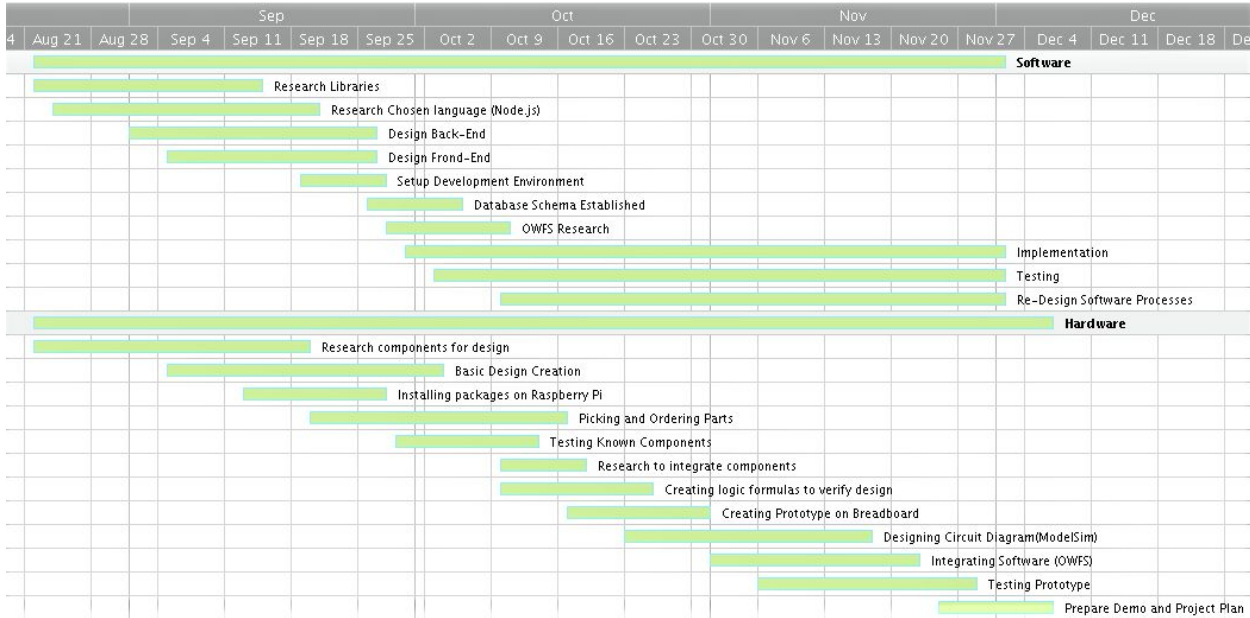


Figure 4: Semester 1 Gantt Chart

**Description of Figure 4: Semester 1 Gantt Chart**

Within the team’s first semester, most of the work was researching in correlation with the learning curve and the knowledge of each member. Figure 4 will show research blocks in between tasks from both sides of the team. Once the team decided on a goal to finish this semester, integrating and testing both the software and hardware with adjustments took majority of the time. The team ended on accomplishing initial prototyping in code and on the breadboard. During this time, each team needed to show an ability to learn outside of classwork and apply newly understood concepts to complete this semester’s work.



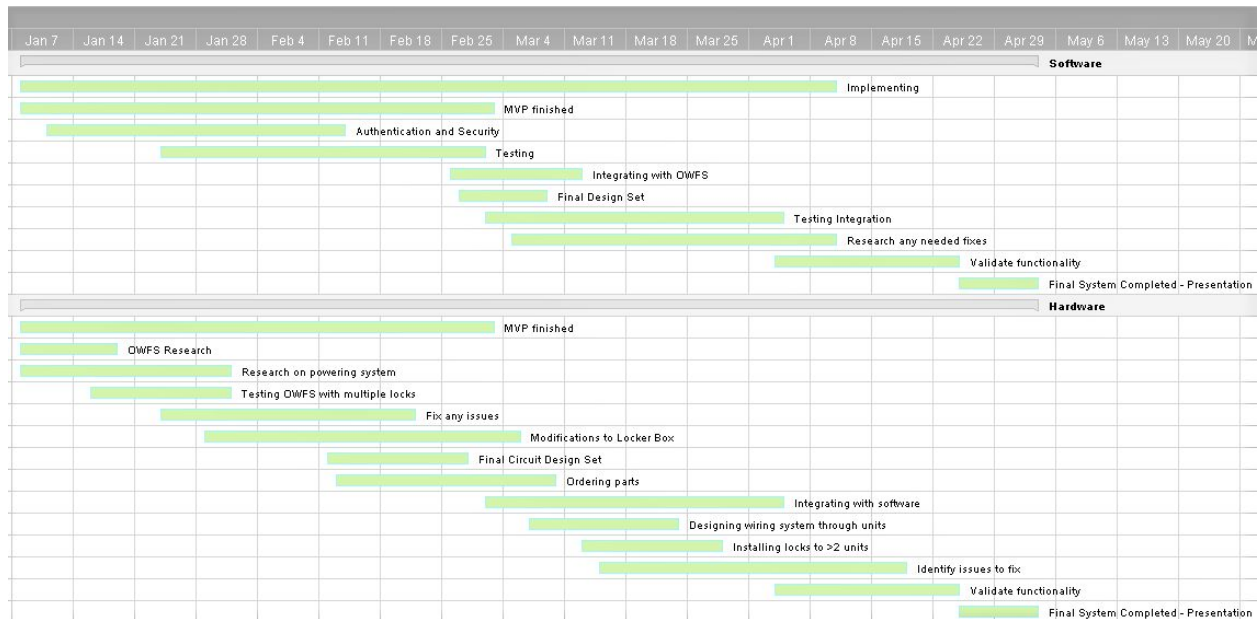


Figure 5: Semester 2 Gantt Chart

### **Description of Figure 5: Semester 2 Gantt Chart**

During the second semester, the expected behavior for the time left consists of solid progress in finalizing designs and running a sufficient number of tests. The implementation will be a running process since tests will show any mistakes needing to be fixed. As a team, roles taken for the project are being switched in order to achieve faster progress to begin testing earlier on and more frequent next semester. The hardware team will take on the responsibility of working with the OWFS package so the software team could have more assistance. All together, parts of the project are more easily identified to take less or more time concluding better planning possibilities. By the end of next semester, the team plans to have a working unit with installed locks and a system. The goal is to have three or more lockers ready but with structured planning, the team will accomplish more.

## 4 Closure Material

### 4.1 Conclusion

In conclusion, the team is making significant progress after stagnation in the early part of the semester. Splitting into two groups - software and hardware - has proven to work well. The software group will now focus on building the Minimal Viable Product web application. The hardware team will continue to implement the proposed hardware

solution. Specifically, they will work with the wiring and with implementing the designed layout of the hardware unit. When moving forward with the plan, new issues will arise and the project plan will need to be adjusted. This being said, team communication will become critical and client communication will need to improve. The team will gain more insight as the project progresses. The client will become more involved in design decisions. This project will take continuous planning and iteration until a quality end product is implemented. The team's mission is to build a useful and scalable solution for the ETG. Our progress shows that the project is moving in this direction. In the future, the team will continue to improve software and hardware processes, testing measures, and product documentation.

## 4.2 References

When building the product, a comprehensive list of resources used to develop this project will be compiled. This list will grow into the second semester.

### **References**

#### Pricing Reference Websites

1. <https://www.ebay.com/itm/DC-12V-mini-Electric-Solenoid-Lock-Assembly-for-Door-Cabinet-Drawer-/162651326271>
2. <https://www.ebay.com/i/121335651801?chn=ps&dispctrl=1>
3. <https://www.walmart.com/ip/ACORN-2415-002-001-Compression-Spring-PK-10/980856671?wmlspartner=wlp&selectedSellerId=1122&adid=2222222227074306075&wmlspartner=wmtlabs&wlo=&wl1=g&wl2=c&wl3=186817173327&wl4=pla-292213087658&wl5=9060053&wl6=&wl7=&wl8=&wl9=pla&wl10=101593696&wl11=online&wl12=980856671&wl13=&veh=sem>
4. <https://www.digikey.com/product-detail/en/maxim-integrated/DS2406/DS2406-ND/1197412>
5. [https://www.mouser.com/ProductDetail/Infineon/IRFZ44NPBF/?qs=9%252bKlkBgLFf24zghzPg2f9g%3D%3D&gclid=CjoKCQiAovnQBRDmARIsAELoM1mfikZ419z37sild3rb8tH6xXuulpp8BTC5\\_Tt9vM53CSmgZ4UucplAaiueALw\\_wcB](https://www.mouser.com/ProductDetail/Infineon/IRFZ44NPBF/?qs=9%252bKlkBgLFf24zghzPg2f9g%3D%3D&gclid=CjoKCQiAovnQBRDmARIsAELoM1mfikZ419z37sild3rb8tH6xXuulpp8BTC5_Tt9vM53CSmgZ4UucplAaiueALw_wcB)
6. <https://www.ebay.com/i/192078713889?chn=ps&dispctrl=1>
7. <https://www.digikey.com/product-detail/en/stmicroelectronics/LM324DT/497-1580-6-ND/1848272>
8. [https://www.digikey.com/product-detail/en/825/1528-1161-ND/5353621?WT.mc\\_id=IQ\\_7595\\_G\\_pla5353621&wt.srch=1&wt.medium=cpc&WT.srch=1&gclid=CjoKCQiAovnQBRDmARIsAELoM1nV5iQhMaiU7IXxwo6bvrlmkpvmQX8X\\_8VyluoO4-oEDOT-SDeHYwaApumEALw\\_wcB](https://www.digikey.com/product-detail/en/825/1528-1161-ND/5353621?WT.mc_id=IQ_7595_G_pla5353621&wt.srch=1&wt.medium=cpc&WT.srch=1&gclid=CjoKCQiAovnQBRDmARIsAELoM1nV5iQhMaiU7IXxwo6bvrlmkpvmQX8X_8VyluoO4-oEDOT-SDeHYwaApumEALw_wcB)

9. <http://www.newark.com/raspberry-pi/raspberrypi3-modb-1gb/sbc-raspberry-pi-3-mod-b-1gb-ram/dp/77Y6520?src=raspberrypi>
10. [https://www.discountelectronics.com/lcd-flat-panel-refurbished-monitors-cheap/?utm\\_medium=googleshopping&utm\\_source=bc&gclid=CjoKCQAgZTRBRDmARIsAJvVWAvDybmOPVhYFdARvj161qiFYZLkk7zO9Ec92nKuxlPcEOCV\\_kbW\\_JUaAg\\_WEALw\\_wcB](https://www.discountelectronics.com/lcd-flat-panel-refurbished-monitors-cheap/?utm_medium=googleshopping&utm_source=bc&gclid=CjoKCQAgZTRBRDmARIsAJvVWAvDybmOPVhYFdARvj161qiFYZLkk7zO9Ec92nKuxlPcEOCV_kbW_JUaAg_WEALw_wcB)
11. <https://www.digikey.com/product-detail/en/maxim-integrated/DS2482S-100-T-R/DS2482S-100-T-RCT-ND/3478855>
12. <https://www.digikey.com/products/en/power-supplies-board-mount/ac-dc-converters/923>
13. [https://www.bestbuy.com/site/dynex-3-4k-ultra-hd-hdmi-cable-black/7525166.p?s\\_kuld=7525166&cmp=RMX&extStoreId=812&ref=212&loc=1&ksid=f962bba2-efa7-4018-9288-e9a2af77fef6&ksprof\\_id=16&ksaffcode=pg252604&ksdevice=c&lsft=ref:212,loc:2&gclid=CjoKCQAgZTRBRDmARIsAJvVWAv3m2QO\\_he6ebUaCHbq25HS7sU6VqGN7y6CvKvc5LzGyJJRaA3XMYEaAhH1EALw\\_wcB](https://www.bestbuy.com/site/dynex-3-4k-ultra-hd-hdmi-cable-black/7525166.p?s_kuld=7525166&cmp=RMX&extStoreId=812&ref=212&loc=1&ksid=f962bba2-efa7-4018-9288-e9a2af77fef6&ksprof_id=16&ksaffcode=pg252604&ksdevice=c&lsft=ref:212,loc:2&gclid=CjoKCQAgZTRBRDmARIsAJvVWAv3m2QO_he6ebUaCHbq25HS7sU6VqGN7y6CvKvc5LzGyJJRaA3XMYEaAhH1EALw_wcB)
14. <https://www.walmart.com/ip/Cmple-50-FT-Black-RJ45-CAT5-CAT5E-ETHERNET-Patch-Cable/765865189?wmlspartner=wlp&selectedSellerId=4378&adid=22222222227070840942&wmlspartner=wmtlabs&wlo=&wl1=g&wl2=c&wl3=182331469598&wl4=pla-286916503655&wl5=9060053&wl6=&wl7=&wl8=&wl9=pla&wl10=117097532&wl11=online&wl12=765865189&wl13=&veh=sem>
15. [https://www.amazon.com/dp/B0711PVX6Z/ref=asc\\_df\\_B0711PVX6Z5291079/?tag=hyprod-20&creative=394997&creativeASIN=B0711PVX6Z&linkCode=dfo&hvadid=198093101467&hvpos=101&hvnetw=g&hvrnd=9647152028429065592&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9060053&hvtargid=pla-359221356227](https://www.amazon.com/dp/B0711PVX6Z/ref=asc_df_B0711PVX6Z5291079/?tag=hyprod-20&creative=394997&creativeASIN=B0711PVX6Z&linkCode=dfo&hvadid=198093101467&hvpos=101&hvnetw=g&hvrnd=9647152028429065592&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9060053&hvtargid=pla-359221356227)
16. <https://www.walmart.com/ip/Inland-PS-2-Basic-Keyboard-Black/28806157?wmlspartner=wlp&selectedSellerId=0&adid=22222222227019618537&wlo=&wl1=g&wl2=c&wl3=40879992512&wl4=pla-78810469832&wl5=9060053&wl6=&wl7=&wl8=&wl9=pla&wl10=8175035&wl11=online&wl12=28806157&wl13=&veh=sem>
17. [http://www.cnaweb.com/economy-usb-optical-computer-mouse-black.aspx?gclid=CjoKCQAgZTRBRDmARIsAJvVWAtR9B2mXYJhYOzHhNF2dUS09\\_1-fKngoaEoB\\_XhkL8utFakNvBnTuGUaAhY3EALw\\_wcB](http://www.cnaweb.com/economy-usb-optical-computer-mouse-black.aspx?gclid=CjoKCQAgZTRBRDmARIsAJvVWAtR9B2mXYJhYOzHhNF2dUS09_1-fKngoaEoB_XhkL8utFakNvBnTuGUaAhY3EALw_wcB)
18. <https://httpd.apache.org/docs/2.4/license.html>
19. <https://github.com/babel/babel/blob/master/LICENSE>
20. <https://github.com/twbs/bootstrap/blob/master/LICENSE>
21. <https://github.com/stefanpenner/es6-promise/blob/master/LICENSE>
22. <https://github.com/esdoc/esdoc/blob/master/LICENSE>
23. <https://github.com/eslint/eslint/blob/master/LICENSE>
24. <https://github.com/expressjs/express/blob/master/LICENSE>

25. <https://about.gitlab.com/products/>
26. <https://github.com/facebook/jest/blob/master/LICENSE>
27. <https://github.com/moment/moment/blob/develop/LICENSE>
28. <https://www.oracle.com/legal/terms.html>
29. <https://github.com/nodejs/node/blob/master/LICENSE>
30. <https://docs.npmjs.com/policies/license>
31. <http://owfs.org/index.php?page=license>
32. <https://github.com/facebook/react/blob/master/LICENSE>
33. <https://github.com/reactjs/redux/blob/master/LICENSE.md>
34. <http://sass-lang.com/documentation/file.MIT-LICENSE.html>
35. <https://github.com/sequelize/sequelize/blob/master/LICENSE>
36. <https://www.unicon.net/opensource/shibboleth>
37. <https://github.com/strongloop/loopback/blob/master/LICENSE>
38. <https://github.com/webpack/webpack/blob/master/LICENSE>
39. [https://www.newegg.com/Product/Product.aspx?Item=9SIA3525527044&ignorebb\\_r=i&nm\\_mc=KNC-GoogleMKP-PC&cm\\_mmc=KNC-GoogleMKP-PC- -pla- -Monitors+-+LCD+Flat+Panel- -9SIA3525527044&gclid=EA1aIQobChMI1oOC\\_Yzx1wIVCa tpCh2mCwYFEAQYASABEgLe D BwE&gclidsrc=aw.ds](https://www.newegg.com/Product/Product.aspx?Item=9SIA3525527044&ignorebb_r=i&nm_mc=KNC-GoogleMKP-PC&cm_mmc=KNC-GoogleMKP-PC- -pla- -Monitors+-+LCD+Flat+Panel- -9SIA3525527044&gclid=EA1aIQobChMI1oOC_Yzx1wIVCa tpCh2mCwYFEAQYASABEgLe D BwE&gclidsrc=aw.ds)

Amazon Hub Website

40. [https://thehub.amazon.com/https://www.amazon.com/b/?node=6442600011&ref=locker\\_short\\_url&pf\\_rd\\_m=ATVPDKIKXoDER&pf\\_rd\\_s=merchandise-search-1&pf\\_rd\\_r=Q4P4RMRYCGE3BDZQG4WW&pf\\_rd\\_t=101&pf\\_rd\\_p=c56e6fc2-4d64-4582-b2a4-4d147c48297e&pf\\_rd\\_i=6442600011](https://thehub.amazon.com/https://www.amazon.com/b/?node=6442600011&ref=locker_short_url&pf_rd_m=ATVPDKIKXoDER&pf_rd_s=merchandise-search-1&pf_rd_r=Q4P4RMRYCGE3BDZQG4WW&pf_rd_t=101&pf_rd_p=c56e6fc2-4d64-4582-b2a4-4d147c48297e&pf_rd_i=6442600011)

## 4.3 Appendices

Javascript Raspberry Pi IO Library:

<https://github.com/rakeshpai/pi-gpio>

Raspberry Pi and 1-Wire:

<https://www.packtpub.com/books/content/raspberry-pi-and-1-wire>

DS2482-100 Datasheet:

<https://datasheets.maximintegrated.com/en/ds/DS2482-100.pdf>