# Small Equipment Checkout System

Design Document

Team Number - sddec20-21
Client & Advisor - Lee Harker
Samuel Sklar - Circuit Board Lead
Kailin Zheng - Hardware Design Lead
Seth Jones - Integration Engineer
Shubham Chauhan - Interface Lead
Michael Momot - Server/Database Engineer
Thomas Smith - Scrum Master/Architect/Developer
Team Email - sddec20-21@iastate.edu
Team Website - http://sddec20-21.sd.ece.iastate.edu/team.html

# Executive Summary

## Development Standards & Practices Used

We participate in using an Agile focus in managing our deadlines and requirement coverage for the completion of the product. In using this method, we designated our client as the product owner, and one of our team members as our scrum master. Regular stand-up meetings are practiced to ensure proper communication is maintained. Effort points are used to gauge complexity of particular tasks that members need to complete as well as estimate the velocity of the team to better estimate deadline completion.

Peer reviewing will be practiced, to reinforce a high quality of code is maintained by the team. Quality control will be utilized by incorporating Git into our development process. Commit messages will be written by members using guidelines encouraged by the Git community.

## Summary of Requirements

Please refer to section 1.4 for a summary of the requirements required for successful completion of the product.

## Applicable Courses from Iowa State University Curriculum

Classes useful for software: CPRE 181, COMS 227, COMS 228, COMS 321,
COMS 363, ENG 311, CPRE 388, COMS 309, COMS 319, SE 329, COMS 487

Classes useful for hardware: CPRE 281, CPRE 288, CPRE 381, EE 201, EE 230, CPRE 489

## New Skills/Knowledge acquired that was not taught in courses

Skills learned include learning to interface with a third party API to access a remote database. We haven't dealt with that kind of challenge in our classes, though our classes have equipped us with the knowledge to work through this challenge. Linking a Raspberry Pi to a web-app and server backend is another skill set we have acquired.

Designing and integrating a circuit board with a One-Wire communication system is a very unique situation that none of us had ever been in. New skill sets have been acquired by teammates such as circuit design, soldering, and embedded testing.

# Table of Contents

## List of Figures

-

# 1 Introduction

## 1.1 Acknowledgement

We would like to thank Iowa State University for allowing us the opportunity to work on this product. We would also like to thank our client and advisor Lee Harker, along with the Electronic Technology Group(whom we will be referring to as ETG for the remainder of this document) for providing us with the resources and guidance needed to achieve the best possible product for our client. A moment should be taken to appreciate and pay respect to the documents from past teams who have worked on this product. Their past work has helped us realize what approaches work and don't work in terms of design.

## 1.2 Problem and Project Statement

Software, Computer, and Electrical Engineering students are often expected to use specialized equipment provided by the college to complete assignments, projects, or to simply enhance their learning experience. Currently students are expected to check out this equipment by way of interacting directly with the ETG's office and borrowing said equipment. While effective, this method can be cumbersome for the ETG staff and has certain limitations that can be hard to overcome.

Records of what students have checked out are currently kept manually, which can lead to mistakes and is generally a hassle if a large number of students check out a large number of things at once, which often happens during busy times of the academic year. In addition, staff must prepare the equipment for checkout as well as process a request. As a result students' time is also wasted as they must wait for ETG staff to complete the steps mentioned above.

The goal of our product is to alleviate these shortcomings by providing an automated system that enables students to check out equipment without having to interact with ETG staff, and allows ETG staff to manage checked out equipment without having to manually service students' requests. This will allow ETG staff to provide better services in other areas while providing students with a streamlined, smooth experience when it comes to equipment checkout.

## 1.3 Operational Environment

The finished product will be installed on the wall outside the ETG's office in Coover Hall. This will require the product to be hanging from a wall. The expected temperature the product will be expected to operate in is room temperature, as it will be held within the hall. Due to the public space, each locker on the product is expected to prevent theft or other malicious actions against the items it stores. Part of this prevention will be a feature to detect if a door to a locker is left open by a user. If so, the product should notify the administration after a set amount of time that is yet to be determined.

The touch screen kiosk users will use to interact with the system will be attached to the product and will feature quick feedback to a user's interactions.

## 1.4 Requirements

Functional Requirements - Users:

- Select an available equipment item to checkout
- Return checked out equipment
- LED light to show contents of locker
- Ability to view checked out equipment
- Ability to view available equipment
- Report broken or missing items
- Report broken parts of system
- Ability to choose checkout duration
- Reminders for students in the form of an email

Function Requirements - Admins:

- Login/Logout functionality
- View available equipment
- View users(students) who have checked out equipment
- Modify privileges of users
- Receive status reports
- Ability to add new lockers to system
- Add new users
- Set checkout limits in specific items

Non-functional Requirements:

- Allow code to be maintained by ETG post senior design
- Provide complete and useful documentation to support future maintenance of system
- Take steps to protect personal info of users of system
- Take steps to protect system from malicious attacks or accidental harm

## 1.5 Intended Users and Uses

The product will accommodate two types of users. Normal users, which will consist of ISU students who have authorization to use the product. These users will be able to check out available items from the product by simply selecting the item they wish to check out and swiping their card or entering their ISU ID into the kiosk. This will prompt the product to open the appropriate locker and allow said user to take the item they requested.

Upon return of an item that was checked out the student will simply repeat the process of swiping a ISU card or entering a ISU ID, followed by the user placing the item the locker opened up by the system. Note that the locker selected by the system upon return of an item may not be the same locker that originally held that item.

The other type of user would be an Admin. Administrators will be able to access the system remotely by way of a web interface. This would allow them to see who has what items checked out, and if any subcomponents of the system need maintenance. Administrators would also have the ability to interact with the system though the touch screen interface if needed.

## 1.6 Assumptions and Limitations

Limitations:

- Product is to be mounted to wall, thus all wiring must be encapsulated in a way that supports this
- Main control board must fit within one of the storage lockers as required by client
- Product is to be left in a public space, and thus must have countermeasures to being tampered with.

Assumptions:

- Any SE, CPRE, or EE students and faculty are able to use the product.
- ISU card will be used to track who checks in/out items
- Product should be able to scale to have any amount of lockers
- Product should be available for use 24/7, with the exception of scheduled or emergency maintenance
- Normal Users will interact with product via a touch screen kiosk and card swipe
- Touch Kiosk is to be mounted/attached to product which will be hanging on wall
- Users will be able to manually enter their ISU ID in the event they do not have the ISU Card with them at the time of check out/in of an item
- Administrators will be able to access, modify, and update the system via a web interface

## 1.7 Expected End Product and Deliverables

End of First Semester - Prototype

We plan to use the prior teams designs to assist us in creating a “prototype” of the product. In reality, this prototype will be what the prior team had for a final result. The intention for this is that based on client feedback, we can implement and improve upon the features that were missing or not satisfactory in the past. Some key features that were missing that we know of are listed below.

- Hardware System: New Circuit to control the lock in the locker that can be controlled through the website or touch screen interface; Sensor to check if door is left open and if items are returned back or not. If left open or not properly returned, the alarm will be activated.
- Software System: In this version of the product, the software will include a website, database, and 1-wire system to operate the product. This software will be integrated with the Printed Circuit Board present within each locker of the product, allowing control and moderation of the lockers via signals sent from our website or interactive touch screen terminal included with the product.

End of Second Semester - Final Product

The focus of the second semester will be to add the required features that were absent in the prior teams iteration of the product.

- Source Code: Source code used in the product will be made available to ETG to allow maintenance and expansion of product.
- Documentation: Proper and thorough documentation will be provided to ETG on completion of the product to assist in understanding and maintenance of the product in the future.
- Operational Product: All promised functionalities and requirements will be delivered in the form of an operational product that satisfies the customer's expectations. This will include but not limited to a website which can control the product remotely, magnetic locks and LED lights for each locker that are controlled via software, and a Printed Circuit Board for each locker that is connected to the server to enable software control of the locker.

# 2. Specifications and Analysis

## 2.1 Proposed Approach

The basic approach to the problem that we are addressing is based on the efforts of previous teams who had the same or similar undertaking. The overarching idea is to use a Dallas One-Wire System in combination with a RaspBerry Pi and a touch screen. This gives the system the functionality of allowing students to checkout equipment from the lockers. The One-Wire allows the system to have a centralized communication system without the need of lots of wires coming from each of the lockers. The touch screen is the primary means by which the user will interact with the system. There will also be a web-interface for administrative purposes. This will allow the admin user to maintain and facilitate the system remotely. This could include modifying user privileges, adding/removing users, viewing available equipment, etc.

The following will restate the requirements, but pointing to which piece of our approach will help to fulfil the requirement.

Functional Requirements - Users:

- Select an available equipment item to checkout → Screen
- Return checked out equipment → Screen
- LED light to show contents of locker → One-Wire System/RasPi
- Ability to view checked out equipment → Screen
- Ability to view available equipment → Screen
- Report broken or missing items → Screen
- Report broken parts of system → Screen
- Ability to choose checkout duration → Screen
- Reminders for students in the form of an email → Web Application

Function Requirements - Admins:

- Login/Logout functionality → Web Application
- View available equipment → Web Application/Screen
- View users(students) who have checked out equipment → Web Application
- Modify privileges of users → Web Application
- Receive status reports → Web Application
- Ability to add new lockers to system → Web App/One-Wire
- Add new users → Web Application
- Set checkout limits in specific items → Web Application

Non-functional Requirements:

- Allow code to be maintained by ETG post senior design → Repository

- Provide complete and useful documentation to support future maintenance of system → Team
- Take steps to protect personal info of users of system → Web Application

We need to acquire all components of the project. We already have access to the touch screen, so we will research and acquire the One-Wire system first as it plays a vital role in the project. We will then need to test the system with circuits. This will include building circuits on a breadboard. The finished circuit will have to detect signals from the One-Wire system so that it can enable the LED, open/close the door and detect if the door is shut or not. Once the circuit has been tested thoroughly enough we will implement the circuit design on a circuit board and then order multiple in order to begin using them with the lockers. We will then establish communication between the One-Wire system and the Raspberry Pi. After communication is achieved, we will be able to merge the hardware and software together and development on the backend software with regards to the One-Wire system will be test-able. This will allow us to make sure the overall functionality of the project for the regular user is achieved on both the hardware and software pieces. Once this has been confirmed, we will be able to begin further testing of the frontend software with the touch screen and admin web-interface. We will also be utilizing some software from a prior team that allows an RFID scanner to be used to scan ISU student IDs to log on to the system. We will be re-implementing and testing that piece as well. At this point, most of the pieces of the project will have been tested and brought together to meet the basic requirements. After all of this has been completed we will spend the remainder of our time improving existing pieces to be better solutions to the overall problem that this project addresses.

## 2.2 Design Analysis

Hardware-

As for hardware, we have been looking to past projects as our primary resource for determining which hardware to use. Once finding the pieces of hardware, such as the Dallas One-Wire system, we then have done further research to ensure that it would be the best option for our project. We have also looked into which Raspberry Pi we would like to use. Our next steps are to attain all necessary parts that we do not already have. We are expecting to see fast progress as we are building off of the success of prior teams who used similar hardware with a similar design.

Software-

For software we are also going to be strongly building off of the success of prior teams. Some of the past projects have included very successful software pieces. So far, we have been reviewing those past projects and evaluating which pieces will result in the most success for our project. This digging has been fruitful, mostly due to the documentation that past teams have done.

Overall, some of the strengths of our proposed approach is that we have seen that similar approaches have resulted in success of some kind. We are not going into this project ignorant and blind. Prior teams have completed a substantial amount of research on all components of this project, which we have already seen to be a vital resource that we must use to be ultimately successful. All members of our team have software experience, which will give us an advantage in software development by default. Although there exist many strengths related to our design approach, there are also weaknesses which exist. Due to the fact that we are reusing large pieces of the software side of this project, it may take a considerable time to get the software to a desirable state as well as identify legacy bugs that are created by code we did not develop. This may or may not include fixing errors due to outdated method calls and other time related errors. This kind of development can be hard and slow. Another weakness in our overall approach is that we have no EE team members. This will make hardware development harder by default.

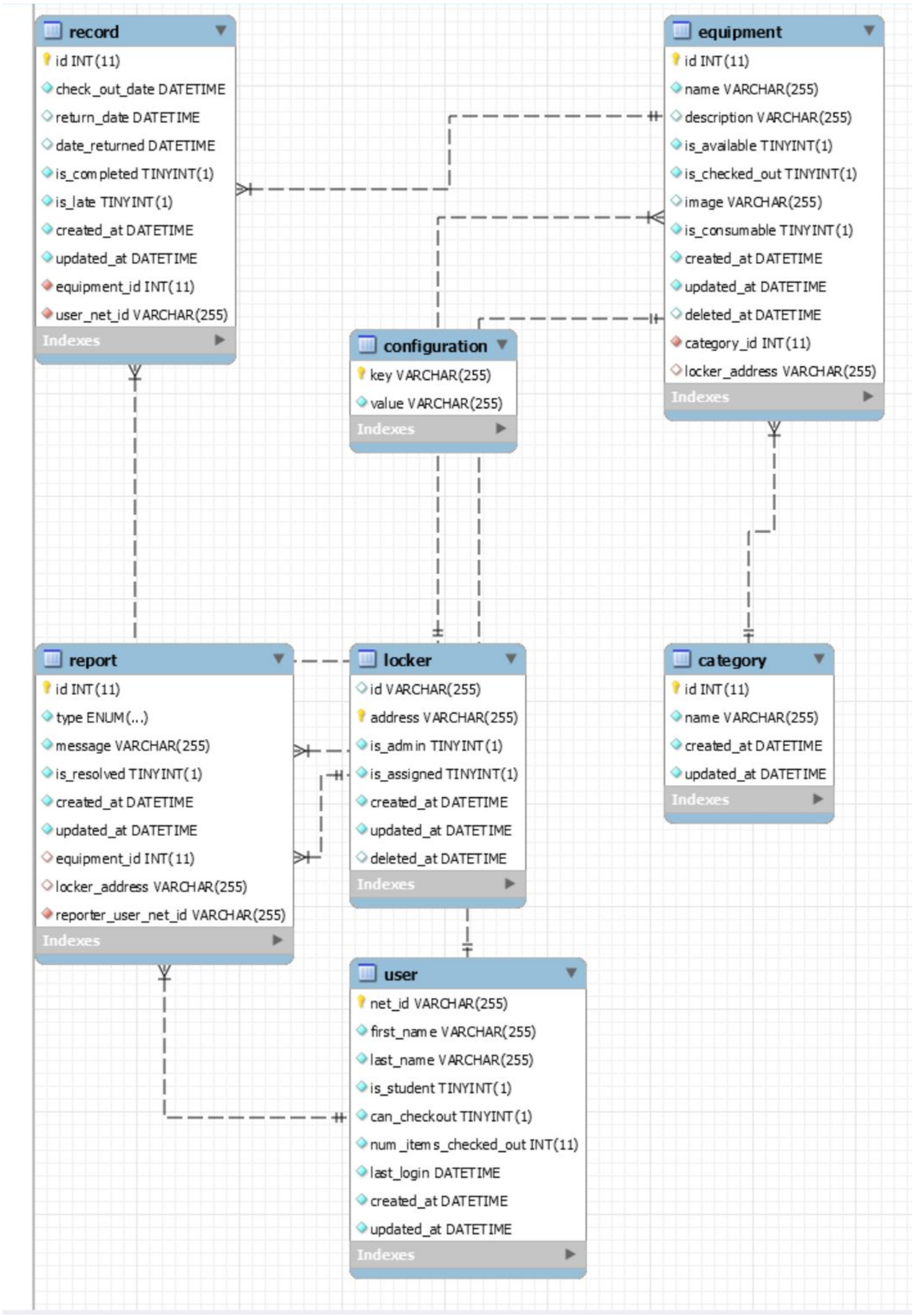


*Figure 1: Database Schema*

This is the initial database design. The initial design is based off of the use cases we have in mind, and the design of previous groups. After testing, additional fields or tables may be necessary if we find that we want to store more information to allow for additional functionality.

## 2.3 Development Process

We decided to use agile development for several reasons. One of the most critical aspects of a successful agile team is its ability to accurately estimate work. Our team is in a special position because we have multiple previous groups' work to give us a better idea of how long certain milestones will take to hit. With the previous teams' documentation, we hope to accurately predict how long tasks will take, and stick as close to our expected timeline as possible. We have elected a "Scrum Master" to coordinate meetings and make sure everyone is on the same page and communicating well. Another pillar of agile is the daily standup meeting. Our team will be holding standup only 2-3 times a week, with the purpose of updating the rest of the team with what you've been working on, and to identify blockers sooner rather than later. We do not believe a daily meeting will be helpful, because most of the team will not work on the project every day. In agile, work is broken down into stories and distributed to the team. Our team will be using Trello to keep track of stories and who is working on what. In conclusion, we believe Agile development will be the best development process for us because of our ability to accurately estimate work, meet regularly, and communicate effectively.

## 2.4 Conceptual Sketch

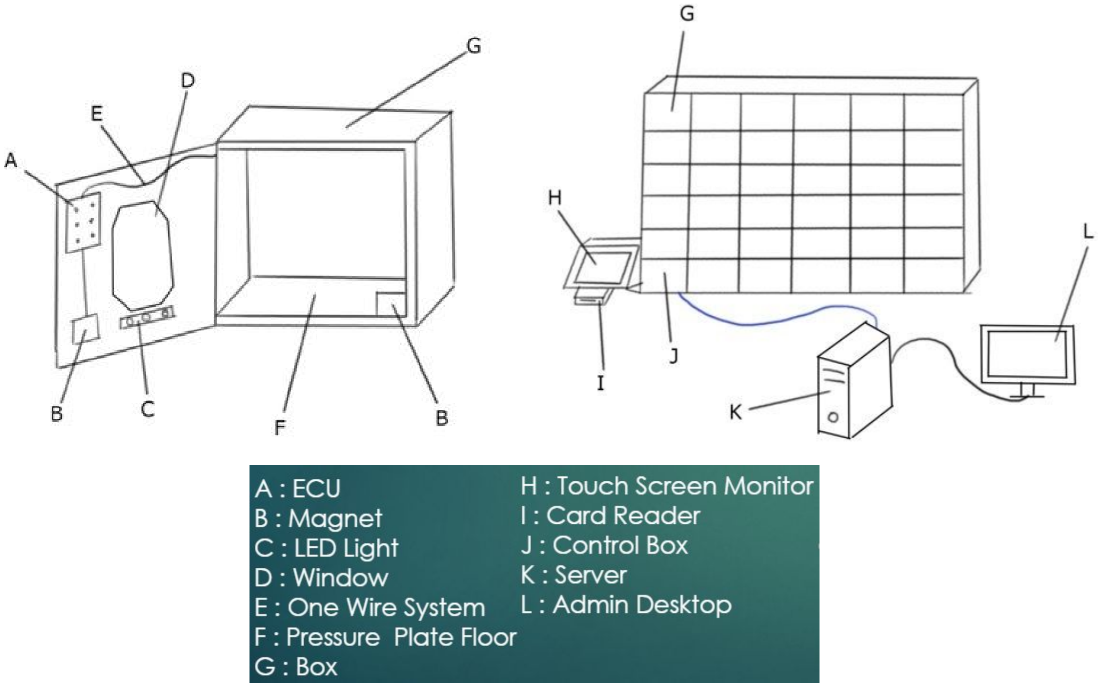


*Figure 2: Conceptual Sketch*

- Metal Locker System
  Cutouts to see the equipments stored inside the locker
  LED lights
- One-Wire system
  Connects from the control hub to all the 35 different lockers
- Display for the front end user
  Monitor screen
- Hardware Keyboard
- Raspberry Pi
- Website
  React
  Java

# 3. Statement of Work

## 3.1 Previous Work and Literature

There have been numerous preceding teams for our project that we are able to reference and build upon. Their design docs and design choices are being used throughout our decision process. One example of an advantage we've taken from previous work is the ability to communicate with A-Star. A-Star is the student tracking system that allows us to get a student's name from their ID card, along with current standing and other useful info. A previous team contacted A-Star and were granted access to their API, which is still in service today. In essence, we did not have to do any extra work in being able to poll the School's database. A shortcoming of previous work was their hardware designs. They are a good reference point, but are taken with a grain of salt, as there are problems present in their designs.

## 3.2 Technology Considerations

We will consider new and existing technologies to implement this project. A lot of what we will need to do is somewhat dictated by the previous teams, as it will be extremely difficult to completely switch over to new tech regarding some of the decisions that these previous teams have made. Given this, we are somewhat limited in what we can use. The front-end will likely need to be completed in React. We already have an incomplete copy of the software from the previous team that we will use for the basis of our work. The backend and the REST APIs will be written in Java. This is probably set in stone as well, especially given our collective confidence with the language from our experience in CPRE/SE classes.

As we continue building, if we discover that a new technology will be easier to implement than getting what the previous team used to work, we will pivot to whatever will be the smoothest to implement. This will be tracked through our Trello board, and we will regularly discuss changes in implementations during our meetings.

## 3.3 Task Decomposition

On the software side, a back-end and front-end needs to be created, which will be divided into distinct tasks. Each of these tasks will be broken up into multiple, smaller subtasks. The front-end will need two use cases, one for admins to maintain the system and another for common users. The back-end will primarily be one single task to handle all incoming requests. For the hardware side, since the shell of the project is done, the major task will be to design and implement the weight system for the detection of items inside.

1. Software
    a. Backend

        i. Handle incoming requests
        ii. Be able to communicate with lockers
    b. Frontend
        i. Admin use cases
        ii. Student use cases
2. Hardware
    a. Design and Implement weight system
    b. Fix previous team's board design

The sections shown above are a very high level decomposition of our tasks for the product. Further task decomposition is done within our Trello board, which contains dozens of tasks that are used to assign jobs to individual team members.

## 3.4 Possible Risks and Risk Management

The major risk involved with this project is the lack of EE knowledge discussed in other sections. A lot of this project hinges on our ability to implement new hardware features on-top of what other groups have accomplished in the past. This will require new circuitry, additional software, and other yet to be seen items to accomplish. Because of this, we will need to learn new EE concepts that we may not have been exposed to before to complete some of the wiring.

To manage this major risk, we will likely move team members from software to the EE portion of the project depending on where there exist major needs. The software side can be completed by anyone in the team, but the EE side will need research by everyone. Members moved to the EE portion of this product will focus on researching and educating themselves in the required areas needed to complete the requirements set by our customer.

## 3.5 Project Proposed Milestones and Evaluation Criteria

The small equipment checkout system project has two major milestones. The first milestone is greatly influenced by the effects of the COVID-19 epidemic. This epidemic largely impedes our ability to work on the physical hardware and lockers for at least the first semester. With this in mind, our first milestone will be to implement the software side of the project using the existing software pieces from past projects. The interaction between the software and the user for this first milestone will be through the GUI via a web application. This milestone will be completed by the end of the first semester. Evaluation criteria will be determined by the client, but will likely involve ensuring that essential parts are working and both an administrator and student user can interact with the unit via the web-app.

Barring the continuation of the effects of the COVID-19 epidemic, the second milestone will be implementing the hardware side of the project and replacing the web-app with the kiosk/locker system along with adding in the features prior teams failed to incorporate into their own design. This will be completed by the end of the second semester. After this, the project should be fully ready to install by ETG, so thorough testing will be used to ensure that all features are working, as well as testing some of the edge cases.

## 3.6 Project Tracking Procedures

We are going to use multiple sources to keep track of our progress throughout this semester and next semester. Primarily, we plan to use Trello to visualize our tasks to completion. We will also use Google Drive to share project information, both within the team and for our client for review. In our meetings, we plan to go over status and review our current standing in terms of meeting our deadlines. Git will be used for version control, and will also provide a reliable history of feature updates for both our own and the future maintainer's convenience.

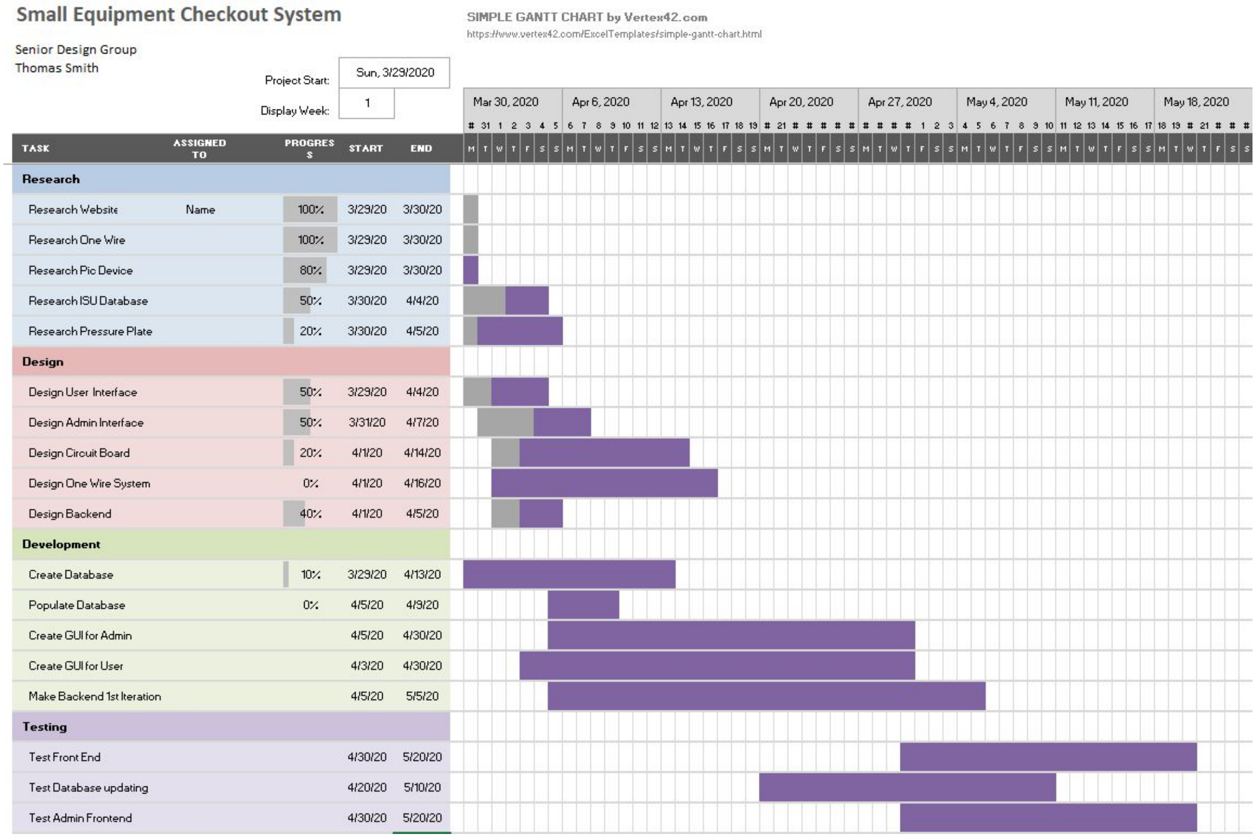
## 3.7 Expected Results and Validation

At a high-level, our client will be the one to confirm that we meet the functional requirements that we were presented with. We will need to ensure that the client is satisfied with what we have implemented, and also that another senior design team will not need to work on this project after us.

# 4. Project Timeline, Estimated Resources, and Challenges

## 4.1 Project Timeline

The following time-line has been created in respect to the first semester. Once the second semester starts, we will reassess the timeline based off of the status of the COV19 epidemic



Small Equipment Checkout System

Senior Design Group
Thomas Smith

Project Start: Sun, 3/29/2020

Display Week: 1

SIMPLE GANTT CHART by Vertex42.com
https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html

| TASK | ASSIGNED TO | PROGRESS | START | END |
|---|---|---|---|---|
| **Research** | | | | |
| Research Website | Name | 100% | 3/29/20 | 3/30/20 |
| Research One Wire | | 100% | 3/29/20 | 3/30/20 |
| Research Pic Device | | 80% | 3/29/20 | 3/30/20 |
| Research ISU Database | | 50% | 3/30/20 | 4/4/20 |
| Research Pressure Plate | | 20% | 3/30/20 | 4/5/20 |
| **Design** | | | | |
| Design User Interface | | 50% | 3/29/20 | 4/4/20 |
| Design Admin Interface | | 50% | 3/31/20 | 4/7/20 |
| Design Circuit Board | | 20% | 4/1/20 | 4/14/20 |
| Design One Wire System | | 0% | 4/1/20 | 4/16/20 |
| Design Backend | | 40% | 4/1/20 | 4/5/20 |
| **Development** | | | | |
| Create Database | | 10% | 3/29/20 | 4/13/20 |
| Populate Database | | 0% | 4/5/20 | 4/9/20 |
| Create GUI for Admin | | | 4/5/20 | 4/30/20 |
| Create GUI for User | | | 4/3/20 | 4/30/20 |
| Make Backend 1st Iteration | | | 4/5/20 | 5/5/20 |
| **Testing** | | | | |
| Test Front End | | | 4/30/20 | 5/20/20 |
| Test Database updating | | | 4/20/20 | 5/10/20 |
| Test Admin Frontend | | | 4/30/20 | 5/20/20 |

*Figure 3: Project Timeline*

## 4.2 Feasibility Assessment

For circuit design, our team's biggest challenge is that we do not have EE students, and the EE classes for Computer Engineering we have taken may not give us enough knowledge to build the desired circuits in a reasonable time. We will ask an EE professor or do more research to figure out this issue. We believe that we can figure the issue out with responsible time management and resource allocation.

## 4.3 Personnel Effort Requirements

| **Task** | **Effort Level** | **Description** |
|---|---|---|
| Setup locker units database | Low | Need to request a database from ETG and implement tables based on project requirement. |
| Design the door detecting circuit | High | Using the sensor and magnet to detect the position of the locker's door, and using a buzzer to alarm users. |
| Design the lock circuit | High | Use a 1-wire chip to assign an unique address for one locker, and voltage supply of lock controlled by 1-wire system. |
| Setup server | Medium | Configure the server based on the instruction of the server website corresponding to the 1-wire device. |
| Setup Backend | Medium | Node.js is a Backend system we will use. |
| Implement Frontend | High | React.js is a Frontend system we will use. |
| Combine circuits and component then test | Low | Connect the lock and door detecting circuit and test it. |

*Table 1: Requirements*

## 4.4 Other Resource Requirements

The project needs the campus server to access the website. Permission to use Attendance Tracker has been granted, and is successfully implemented into the backend of our application. Training has been complete and permission has been granted to use power tools from ETG.

## 4.5 Financial Requirements

| Item | Cost ($) |
|---|---|
| Kiosk | 0.00 (Provided by ISU) |
| Slave Devices: | 3.90 * 34 |
| 12v Electric Lock | 4.00 * 34 |
| Raspberry Pi 3 | 41.99 |
| Box Modification | Undetermined exactly what modifications need to be made, but likely minimal and able to use metal pieces already in the shop |
| 3 Track Magnetic Card Reader | $15.94 |
| One Wire | Roughly $5 per circuit board depending on how many ordered |
| Server (Linux VM) | $0 (Provided by ISU) |

*Table 2: Costs*

# 5. Testing and Implementation

Integrity Testing - Our application depends on the integrity of the data within our database. User input that could potentially compromise the integrity of the database will have to be tested. Input validation will have to be good enough to ensure the user cannot put in any characters that will cause problems in the database. SQL injection is a common attack on databases, so we will test to ensure that we are not vulnerable to malicious user input attacks. It is also possible that a user uses a stolen card to check out a device. In this scenario, we must be able to recover from this and ensure that the student's reputation is absolved. We need to test our system to ensure that an admin can adjust the database as they see fit, and can recover from such a spoofing attack.

Unit Testing - We will be unit testing for all the use cases of our system to ensure all aspects are functional. The most important use cases we need to test are the ability to check out and return an item. We do not plan on having any kind of auto-test for the GUI, instead, we will manually test GUI functionality. Backend testing will also be done both through the GUI and through test scripts. Test scripts will consist of simple commands sent to the backend, and we will verify if the backend did what it was supposed to.

User Study - Due to having to work remotely, it is impossible for us to observe real user's using the physical system. However, we are able to allow for students to use the website. In its current state, the website is only able to run locally on our computers. We believe that allowing users to use the website on our local computers with our supervision is the best possible way to study users. This is not ideal, because the locker system is the biggest unknown for our system, and we are unable to test it with real users. By next semester, we will hopefully be able to test the system in full with real users.

## 5.1 Interface Specifications

There are several areas of our project that will need to interface with each other. The major points of communication are detailed in the graph below. It will be important for us to thoroughly test backend/hardware communications because that is the area where we will most likely have the most risk. We need to consider all possible user cases, to make sure the UI meets all requirements and is also reliable. A challenge we are expecting for interface testing is that both pieces of software need to be completed in order to test the communication. We expect integration testing of the different systems to be done towards the end of the project when most of the functionality is complete.

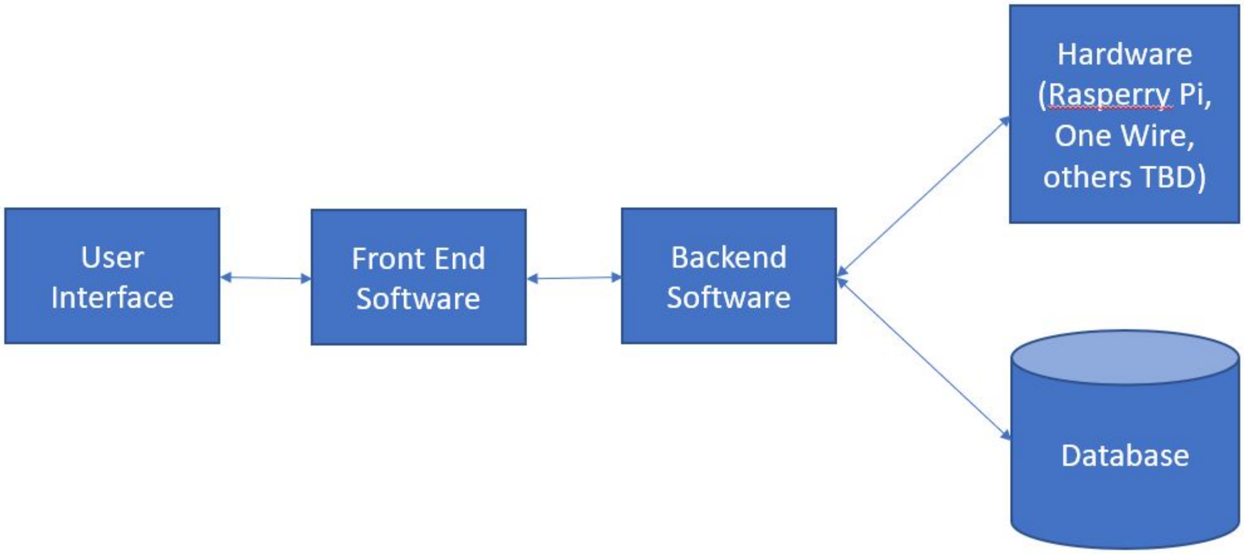


*Figure 4: System Interface*

## 5.2 Hardware and Software

Software: GUI testing will need to be done to ensure the touch screen display will not crash while the user is using it. Testing will also need to be done for the administrator user interface, which will be a web application. Querying the database and updating the frontend is a good opportunity for the use of automated test scripts. Test scripts to simulate message flow between front/backend will be needed to test communication with other pieces of software that has yet to be complete. Integration testing with the hardware will need to be done when we can verify the individual components are working as expected in a simulated environment.

Hardware: The main hardware components to test are the master board and the boards that will be going into each of the lockers. There are cases where a test script will not be sufficient in testing the hardware. We will have to physically test the locking mechanisms, LED lights, and the touch screen ourselves. In general, individual components/functionalities will be tested before they get tested as a whole. Embedded testing will also be practiced.

## 5.3 Functional Testing

The team plans on doing requirements-driven testing. Our tests shall be designed to test for specific requirements. This way, we know we are not wasting time developing and testing for things that are not essential to the project. Requirements such as swiping a card to log in, an LED turning on, or a locker opening will all need to be tested manually. For the project to be considered complete, it will need to pass a series of manual or automated tests for each individual requirement.

## 5.4 Non-Functional Testing

SQL is known to be vulnerable to certain types of attacks, so we shall test the database to ensure information security. We will also need to test for data validation to ensure the user cannot compromise the integrity of the database. We also want to user test the frontend to ensure it not only meets functional requirements, but also that it is intuitive and aesthetically pleasing.

## 5.5 Process

For the software side, we will have unit tests to test specific functionality. These will be written as we develop. The idea is to have a set of regression tests so that when changes are made, we can run the tests to ensure that the changes did not break existing functionality. If the test does fail, then we know right away what changes caused something to break. Hardware testing is going to be less defined, as we do not expect the hardware to change as often as the software will.

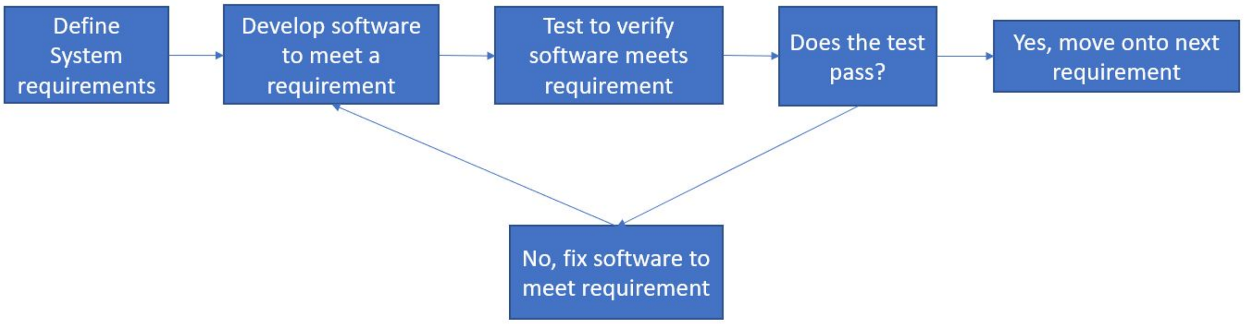

Define System requirements
Develop software to meet a requirement
Test to verify software meets requirement
Does the test pass?
Yes, move onto next requirement
No, fix software to meet requirement

*Figure 5: Testing Process*

## 5.6 Results

### Hardware Results

We have been able to meet our revised testing goals. The main goal for the hardware once we went remote, was to be able to connect to the backend, and show that we are able to send messages to specific lockers. We tested this functionality by setting up an Arduino with several lights representing different lockers. We were able to confirm the connection between our backend, database, and hardware is working correctly by this test.  This was one of the most difficult parts of the project, and completing it allows us to test other functionality once we are back on campus.  Once we go on campus again and are able to integrate with the locker setup, we will be able to test if our device fits within the locker, and how well the master device communicates through One-Wire to the many slave devices in the locker.

### Software Results

The Prototype has successfully implemented our post-Coronavirus goals, with some bugs at the moment. User authentication with A-Star student tracking is functional. Core functionality (Item checkout, locker assignment, viewing records) has been manually tested and verified. We have tested our server and database functionality with sample data, and showed that the database can successfully handle the calls we make to it.  Software testing has shown several bugs so far that need to be addressed before the final product can be released. The reporting system that customers use to report defective items is currently broken. The GUI exists for users to create and submit reports, however, admins cannot currently view the reports. Another major bug currently in the system is that an item is showing it was returned when it is really not. Although our

# 6. Closing Material

## 6.1 Conclusion

We examined the past semesters projects in detail and decided how they can be used in our design implementations. While the past teams work can be used, there is a major problem from the past that we would need to resolve by ourselves. The One-Wire system was not functioning in the prior team's version of the product as per our team's research. The circuit boards also did not fit the door panels, which our team will need to redesign and implement to meet our client's requirements.

The goal for this semester was to work on recreating the prior team's final product. Given the circumstances created by COV19, we were able to get the software and GUI portions of the product to the desired stages. Hardware on the other hand was not able to be where we wanted it, due to our lack of access to our prototype once the college shutdown. Due to this, hardware integration will be our first priority next semester.

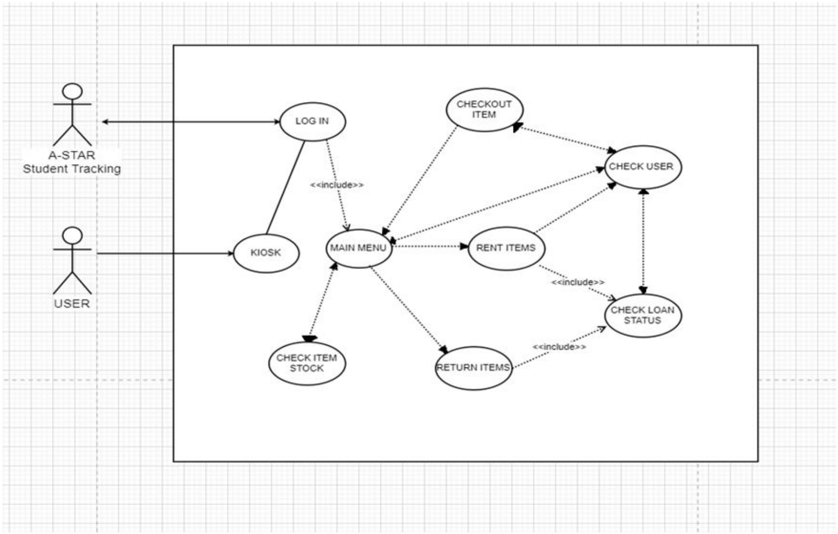
## 6.2 References

Prior Teams Website: https://sdmay18-01.sd.ece.iastate.edu/

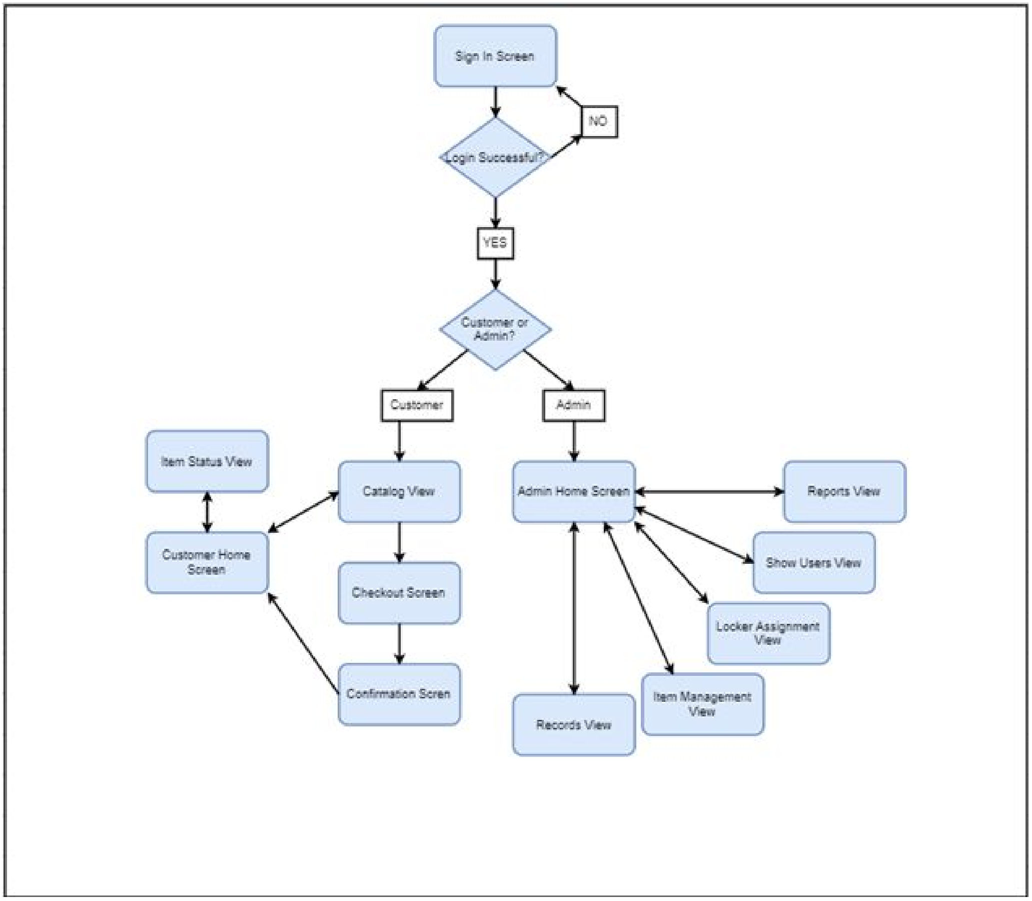
## 6.3 Appendices

ETG - Electronic Technology Group

Use-Case Diagram

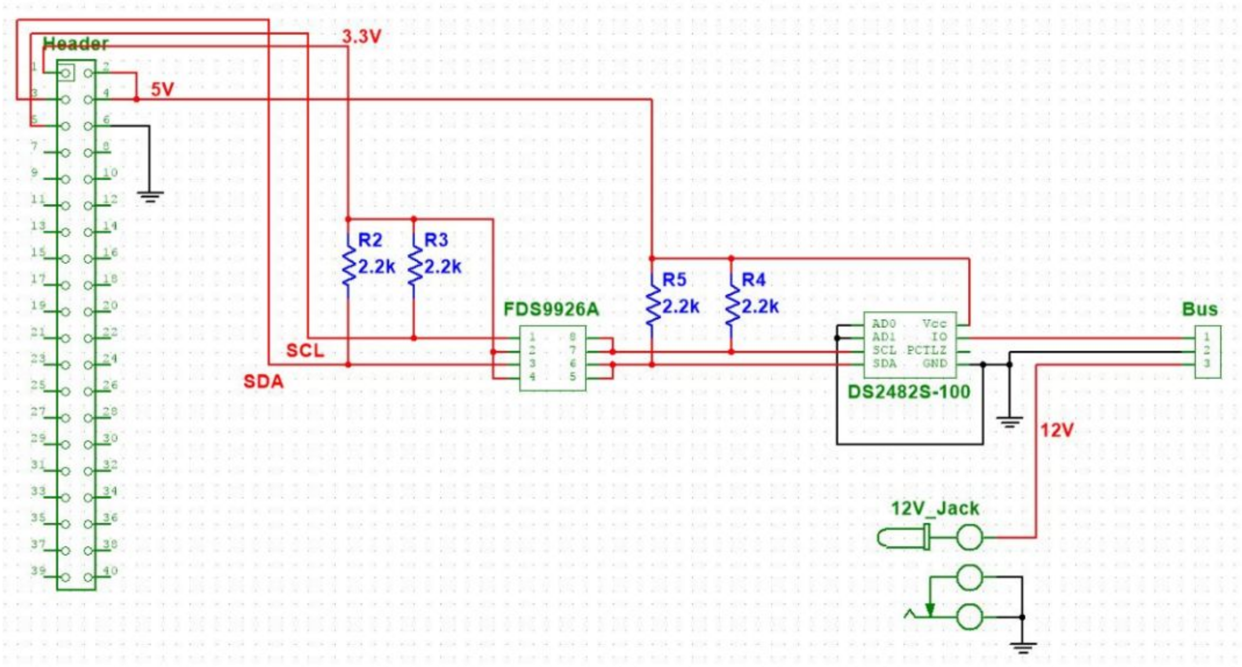


Activity Diagram

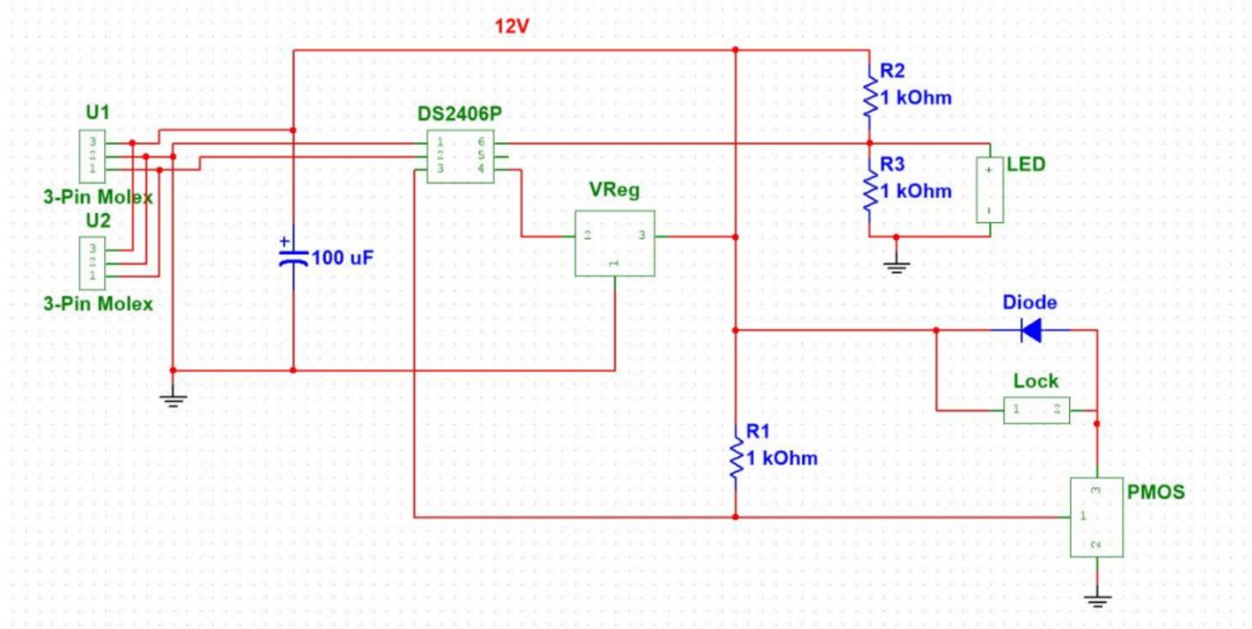


## Design Circuit Diagrams

### Master Device



### Slave Device


12V
R2
1 kOhm
U1
DS2406P
3-Pin Molex
U2
3-Pin Molex
R3
1 kOhm
LED
VReg
100 uF
Diode
Lock
R1
1 kOhm
PMOS