Small Equipment Checkout System

Design Document

Sdmay19 - 13
Client: Leland Harker
Adviser: Leland Harker
Team Members/Roles:
Yimin Wang - Chief Manager / Hardware team
Fengnan Yang - Meeting Manager / Hardware team
Jiaxin Li - Treasurer / Hardware team
Cai Ning Wang - Software Reporter / Software team
Bei Zhao - Secretary / Software team
Team Email: sdmay19-13@iastate.edu
Team Website: http://sdmay19-13.sd.ece.iastate.edu/

Revised: 10/12/2018
Version:1

Table of Contents

List of Figures

List of Definitions

● C.R.U.D.: An acronym for Create Read Update Delete. This acronym is typically used to describe software functionality.

● ECpE: Electrical and Computer Engineering department of Iowa State University

● OWFS: An acronym for 1-Wire Filesystem.

● ISU: An acronym for Iowa State University

● 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.

1 Introduction

1.1 Acknowledgement

This design team, sdmay19-13, would like to thank Iowa State University(ISU), and college of Electrical and Computer Engineering(ECpE) for providing the excellent opportunity for students to form teams and to work on professional design process. Especially, we want to thank Electronic Technology Group(ETG), and our client and advisor, Lee Harker, for all the guidance and technical support. We are also appreciate for all the contribution from the team, sdmay18-01, who has worked on this project before. Their design shows us many possibilities for this project.

1.2 Problem and Project Statement

Electrical and Computer Engineering are subjects rely heavily on experimental experience, and students will utilize various components and equipment throughout their learning process. Currently, the most common way for students to checkout equipment is borrow directly from Electronic and Technology Group (ETG)'s part shop. However, this checkout procedure has its own limitations in working time and efficiency.

The current checkout system requires ETG faculty spending time in part shop to process students' requirement, preparing demanding equipment, and recording manually. This is not just inefficient in time, but also a waste in human resource. Also according to our research, a large percentage of students likely to do the experiments after school or during weekends. In this situation, the ETG's limited working time, which is weekdays from 7am to 5 pm, becomes a hinder for students to get necessary resources.

Overall, the final goal of this project is designing a feasible and reliable equipment checkout system that can solve the problems listed above by simplifying faculty's maintain process, and providing more availability for students.

1.3 Operational Environment

The accomplished design will be placed outside the ETG's part shop in Coover Hall. The whole system is supposed to work under room temperature for all times. The system can be divided into two parts, one is the shelf with boxes that used for storing equipment, the other one is a touch screen for user's interface. The shelf is made by metal and will be fixed to the wall, so its stability is the most important concern and will be tested carefully. The touch screen should have quick and accurate feedbacks to user's operations. Since the system will be place in public environment, the locker in each box should be able to protect the box from steals and slight damages, and the hardware system should be able to detect any functional error of lockers, then the system can send notification to administrators through e-mail system.

1.4 Intended Users and Uses

There are two kinds of intended users: students and administrators.

In order to have needed equipment in time, plenty of students have to schedule their experiences during weekdays, and this may cause the shortage in both laboratory space and equipment. But Small Equipment check out system will allow students to borrow equipments by all day and all hours. As the result, students will have more flexible studying time by using this system.

This system will only requires administrators to do the maintain at very specific time, so administrators will be released from fixed daily office hour. Then, they can focus more on technical support for ECpE department and improving our lab experience.

1.5 Assumptions and Limitations

Assumptions:

All ECPE students and staffs can use the system.

User will use a touch screen to interact with the system.

User can swipe ISU card or type in ISU ID to use the system.

The number of boxes is not constant, it can extends as many as the number of boxes.

The system can be used all the time except it is in maintenance.

For each system, one box will be used to store main control board.

Limitations:

The box should be larger than the size of PCB.

The cost to build a locker will be about $12.06.

ISU students and staffs who are in other departments cannot use the system.

1.6 Expected End Product and Deliverables

1. Prototype - end of first semester

At the end of the first semester, we will have a prototype that software and hardware systems can work well separately.

> a) Hardware system We will design new PCBs to control lockers. The new design should have a appropriate size to be placed on the door for each box. And it should also be able to control the LED inside the box.The PCB will be powered by Raspberry Pi and need to pass the correct voltage for locker and 1-Wire chip. A sensor should be included in box to check the door is closed after using, or the alarm will be activated. All PCBs in boxes are supposed to connect Raspberry Pi through one bus line.
>
> b) Software System The software include website, database, One wire file system, and identification system. For the prototype, the website should be able to react with user's operation, and the identification system can identify users by using A-track system and card reader. The software system also should have a relatively well-developed database to save all the information of boxes

2. Final product - end of second semester At the end of second semester, we should have a perfect and functional system for ECpE student to use in real operation environment.

> a) Integrated System For the final product, the software and hardware system should work together perfectly. The lockers and LEDs inside boxes are able to controlled by software system. And the software system should have function for sending notifications to administrators after receive reports from users for any missing and damages of equipment, and any error of system.

b) Document The final delivery should include the detailed design documents. By following the documents the ETG should be able to maintain and connect more unit easily.

2. Specifications and Analysis

2.1 Proposed Design

The main controlling system of our design is the Dallas One-Wire system. So, understanding how it works will be our first mission. We already have the Master and Slave devices of One-Wire system that built by previous design team, but the result is unsatisfactory. One problem of current design is that whatever the box is opened or not, the LED that used for users to indicate whether item exist in box is always on. And the power plan of locker also need to be improved. So, before we start modify other function or connect more units to system, we have to redesign the circuit to make sure our PCB can control the LED and lockers properly.

In order to make the wiring of whole system tidy and clear, we need to resize the PCB so it can fit in an plastic cover and fix on box’s door. It could be a good idea to change the package of capacitor on PCB to reduce the height, and remove the extra 3-pin input.

Raspberry Pi will be the microcontroller in our project. There are two kinds of applications under-construction right now.

The first one is the web application which has already implemented in the operating system of Raspberry Pi. However, since the web application developing tool React has version updated, when we compiled the source code of previous team, there are a lot of errors shown up. What’s more, the previous team has set up locking screen to the Raspberry Pi, so we cannot access the operating system of the Raspberry Pi currently. Therefore, the main two tasks in the web application side are refactoring the web application code to force it to be suitable for current React version and finding a way to break the locking screen mode set up by previous team.

The second one is the Java application. We will use swing API to make our GUI, it will have a homepage that shows all the boxes in grid view with a picture of the item insides. On the top left of the page, there will be a Manager login button. User will choose which item he wants and it will go to the item detail page. After user click on checkout, a window will pop out and ask user to check if the item is missing in the locker, if user choose item missing, it will go back to the home page. If user choose item is not missing, system will ask user to swipe the ISU card, if the system successfully get user information, it will open the locker and make a record, then back to homepage. If it's successful, it will tell user to swipe again. If user choose manager login on homepage, it will goes to a new page to ask user to enter password. After manager login to the system, it will allow manager to edit item information in each box or reinitialize the whole shelf to change the size of shelf.

Generally, the **functional requirements** that will be implemented by the proposed design above are:

Whole System:

- New PCB can control the LED and lockers.
- New PCB can fit in an plastic cover and fix on lockers' door.
- Alert users by sound when they forget to close the lockers' door

For Students:

- 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

For Administrators:

- 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

The **non-functional requirements** that will be implemented by the proposed design above are:

- The whole checkout system could work stably with properly implemented hardware and software.
- The whole checkout system could work safely with appropriate set up firewall.

2.2 Design Analysis

Hardware-

We did lots of research on the One-Wire system to see how it works. And the only One-Wire system we could use is the product from previous team. But their product does not work as we expected. Because the components from previous team has already soldered on the board, we have to order another slave device from ETG for prototype. We will start combining our hardware and software system together ones new devices arrive.

Besides that, we will keep working on hardware itself. The current task is redesigning the LED and lock parts on the circuit so they can work properly. We still need to reduce the size of the PCB. We could arrange all components more sequently and remove unnecessary parts on the circuit.

Software-

As the discussion in the Design Analysis section, currently for the software portion, there are two kinds of software application are under construction.

The first one is the web application that has already implemented the major functionality by previous team. But this application has two major issues need to figure out which are the errors came out while compile it for React version updated and not being able to access the operating system for the locked screen settled by previous team.

There are three approaches that could use to settle the two problems above. For the compiling problem, we would like to try to refactoring the source code to force it to follow the rules of the new version of React. If this is not working, we may implemented a new website by referring to current source code that we have. For not being able to access operating system problem, we would like to format the current SD card to set up a new operating system for our raspberry Pi to break the locking screen situation. The **strength** of the proposed solutions above is the problem will definitely be settled down at last. However, the **weakness** of the proposed solutions is that the progress of the software construction will be slower than previous team, since we only have two software programmers.

The second one is the desktop application. We decided to use Java as our second choice because it has the **strength** that can be easily transferred to android devices which is most popular platform in the world. However it still has a **weakness** that user have to set up Java environment before run a Java app.

3 Testing and Implementation

For the hardware section, we need to test each part detaily to make sure the hardware system can working well properly. This is a efficient way to allocate problems. First, the Raspberry Pi needs to process the command send by the user through touch screen. Second, after Raspberry Pi received the order, it can transmit a correct signal, such as unlock the door, lock the door, or turn the LED. Third, we need to make sure our one-wire system can transit the signal to the right locker box and "unlock" or "lock" the door. After make sure all individual part working correctly, we can put them all together and get the final test results.

For the software section, we need to generate a lot of test cases to make sure each application work appropriately. Sometimes some errors may be happened out of our expectation, we need to generate extra test cases to test that. Based on the basic functional requirements, we need to set up different kinds of users to test the website and make sure the website will show up correctly. What's more, we also need test the application in different systems to prevent the application cannot run when the ETG group wants to switch current monitors and computers.

3.1 Interface Specifications

There are two main parts for this project. First, the check in and out system should read the student ID and let the user choose either borrow or back the component to the locker. Then, the computer will send the signal to the raspberry Pi. Then, the raspberry Pi will process the signal and send the signal to the locker box by one-wire system.

3.2 Hardware and software

Hardware:

- D2482S-100+T&R One-wire chip to connect Raspberry Pi and locker.
- Raspberry Pi used as microcontroller.
- Electric lock To unlock and lock the individual doors.

Software:

We will use our Java based software for testing. It will provide a graphic user interface for user to make a checkout. It will give user the information of items in each box, and let user choose which box he wants. Then, it will ask user to input personal information and finish the checkout process. It will also send signal to the control board about which box is selected.

3.3 Functional Testing

- ECPE students and staffs can make a checkout.
- User can see what item is inside each box.
- Manager can edit item information.
- Manager can browse checkout records.

3.4 Non-Functional Testing

- Firewall can block hackers steal student information.
- A mechanical part can open and close the door.
- The website can work in different operating systems

3.5 Process

In general, we have to test our project in both hardware and software sides:

For hardware:

We need to test the led light in the new designed PCB work as we wish or not.

We need to test the scale of the plastic cover and lockers door to rescale the electronic components on our PCB.

We also need to test whether the alert function of our lockers work appropriately or not.

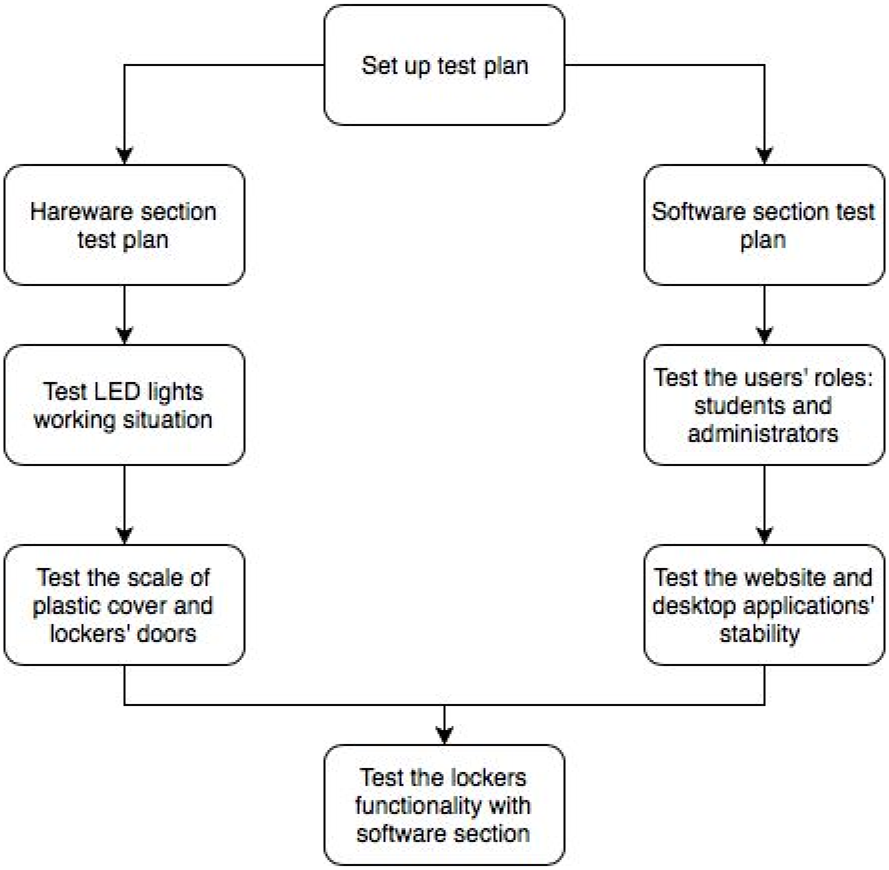
For Software:

We also need to test whether the alert function of our lockers work appropriately or not, since the alert functionality is a combined product of both software and hardware.

We need to test the stability of the website by testing in different browsers and operating systems.

We need to test different users' roles users/advisers required by the client/adviser.

The flowchart below is the current test plan of our team.


Set up test plan
Hareware section test plan
Software section test plan
Test LED lights working situation
Test the users' roles: students and administrators
Test the scale of plastic cover and lockers' doors
Test the website and desktop applications' stability
Test the lockers functionality with software section

3.6 Results

Since we have not finished the major implementation yet, we do not have testing result currently.

4 Closing Material

4.1 Conclusion

Due to the complication of this design document, we have fully analyzed the creation of previous design group. Based on our testing and research, we decided the main problem we need to solve in the following weeks is designing a new PCB that used to communicates with one-wire file system(OWFS) and controls LED and locker. Our two software team members plan to develop website application and desktop application separately, at the end of first semester, we will evaluate these two approaches and choose the better one. Our goal for the first semester is having a working prototype that can control one box through our software system. For the last semester, we will work on connect all 35 boxes to the system.

4.2 References

- 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-wireDS2482-100
- Datasheet: https://datasheets.maximintegrated.com/en/ds/DS2482-100.pdf

4.3 Appendices