Small Equipment Checkout System

Design Document

Team Number - sddec20-21
Client & Advisor - Lee Harker
Team Email - sddec20-21@iastate.edu
Team Website - http://sddec20-21.sd.ece.iastate.edu/team.html
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 - Meeting Scribe/Architect/Developer

Table of Contents

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. In addition, staff must prepare the equipment for checkout as well as process a request. In addition, student's 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 of 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 box 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 box 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 terminal users will use to interact with the system will be attached to the shelf of boxes and will have 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 box
- 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 boxes 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

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 by swiping their card or entering their ISU ID into the kiosk. This will prompt the product to open the appropriate box and allow said user to take the item they requested.

Upon return of an item that was checked out the student would simply repeat the process of swiping a ISU card or entering a ISU ID, followed by the user placing the item the box opened up by the system. Note that the box selected by the system may not be the same box that originally held the 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 boxes 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 boxes
- 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
- Touch Kiosk is to be mounted/attached to product which will be hanging on wall
- Users will be able to use the ISU ID in the even 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, the prior iteration of the product did so much have things wrong with it, as it was missing key features.

- Hardware System: New Circuit to control the lock in the box and can be controlled through the website; Sensor to check door lock and items return back or not, if is not, 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 box of the product, allowing control and moderation of the box via signals sent from our website or interactive touch screen terminal included with the product.

End of Second Semester - Final Product

The focus if 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 box that are controlled via software, and a Printed CIrcuit Board for each box that is connected to the server to enable software control of the box.

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 mailboxes. The One-Wire allows the system to have a centralized communication system without the need of lots of wires coming from each of the mailboxes. 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. 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 box → 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 boxes 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 → Screen
- 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 mailboxes. 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. Our next steps are to begin development for the frontend and backend pieces of the software as the hardware team continues in their initial steps.

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 working condition. This could include fixing errors due to out of date 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.

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 previous group's work to give us a better idea of how long certain milestones will take to hit. With the previous team's 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, however, 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

**** Cited**

- 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 extensively throughout. Those documents are referenced at the end of this document. Further literature used is TBD as we are not completely sure what additional items we will be adding to the project.

3.2 Technology Considerations

We will consider all 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 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 a near-working copy with most features from the previous team. The backend and the REST APIs will be written in Java. This is probably set in stone as well, especially given 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 most definitely pivot to whatever will be the smoothest to implement. This will be tracked through our Trello board, and we will constantly discuss changes in implementations during our meetings.

3.3 Task Decomposition

Given that we are still in the research phase, it is impossible to create a comprehensive list of tasks and deadlines.

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 add team members 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.

3.5 Project Proposed Milestones and Evaluation Criteria

Currently, we have two major milestones. The first will be to get the existing project working properly and all functions working. This will be completed by the end of the first semester. Evaluation criteria will be determined by the client, but will likely involve ensuring that all parts are working and interact with the unit as both an administrator and as a student. The major thing that will be evaluated will probably be the happy case for the use of the unit.

The second milestone will be the bettering of the existing project and addition of new features. This will be completed by the end of the second semester. After this, the project should be fully ready to install by ETG, so rigorous 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. 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 to our client for review. In our meetings, we plan to go over and review the many aspects of this project.

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 team will not need to work on this project after us.

4. Project Timeline, Estimated Resources, and Challenges

4.1 Project Timeline

In first semester:

- Research (2020.1.14 - 2020.2.11)
- Project Plan (2020.1.14 - 2020. 2.18)
- Design circuits (2020.2.25 - 2020.3.17)

- Design Software (2020.2.25 - 2020.3.31)
- Order components (2020.3.24 - 2020.3.31)
- Test components (2020.4.7 - 2020.4.28)

In Second semester:

- Assembly (TBD)
- Test whole System/Bug fix (TBD)
- Improvement (TBD)
- Finish final version of project (TBD)

4.2 Feasibility Assessment

For circuit design, our team biggest challenges is that we do not have EE students, and some EE classes for Computer Engineering may not give us enough knowledge to build the perfect circuits, and we should asked EE professor or do more research to figure out those issue, and we believe that we can figure that out with great time manage and excellent learning ability.

4.3 Personnel Effort Requirements

Task	Effort Level	Description
Setup locker units database	Low	Need 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 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 may use.
Implement Frontend	High	React.js is a Frontend system we may use.
Combine circuits and component then test	Low	Connect the lock and door detecting circuit and test it.

4.4 Other Resource Requirements

The project needs the campus server to access the website, and we need permission to use Attendance Track and also need to use power tools from ETG.

4.5 Financial Requirements

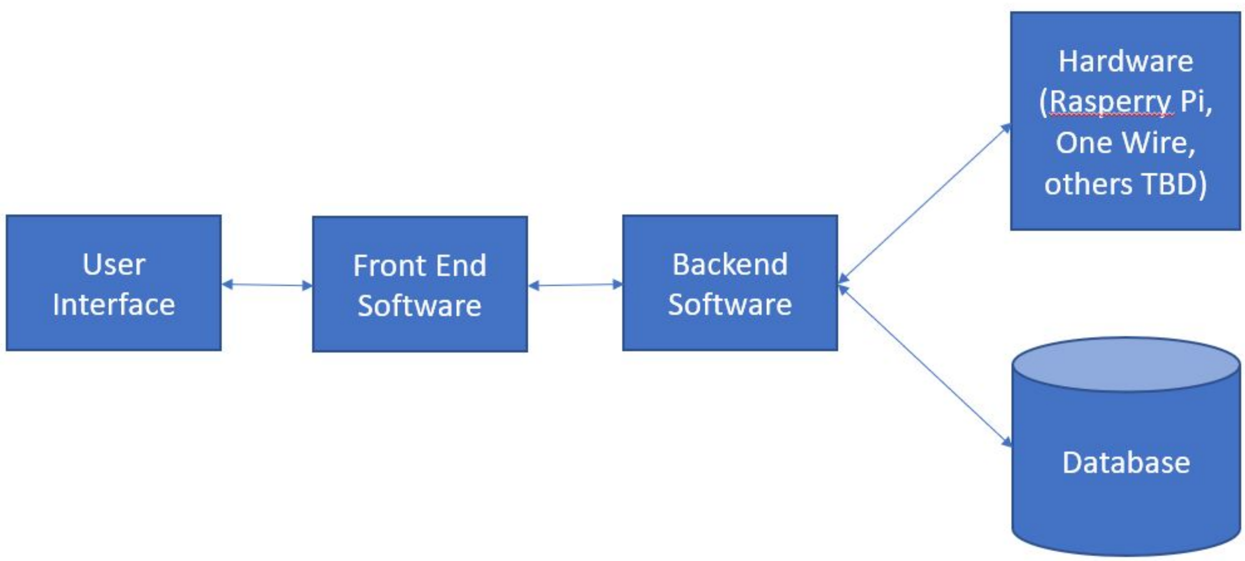
The feasibility cannot be described now because the project is under research and plan designing.

5. Testing and Implementation

Until we have written functional software, we will not be able to test our system. Assumptions are made based on the project plan, which could change as the project progresses.

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.



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. GUI testing will also need to be done for the administrator user interface, which will be a website. Querying the database and updating the frontend is a good opportunity for an automated test script. Test scripts to simulate message flow between front/backend will be needed to test communication with another piece of software that is not complete yet. 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 boxes. 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 touch screen ourselves. The individual components will also need to be tested. In general, individual components/functionalities will be tested before they get tested as a whole. Embedded testing will be done alongside this testing.

5.3 Functional Testing

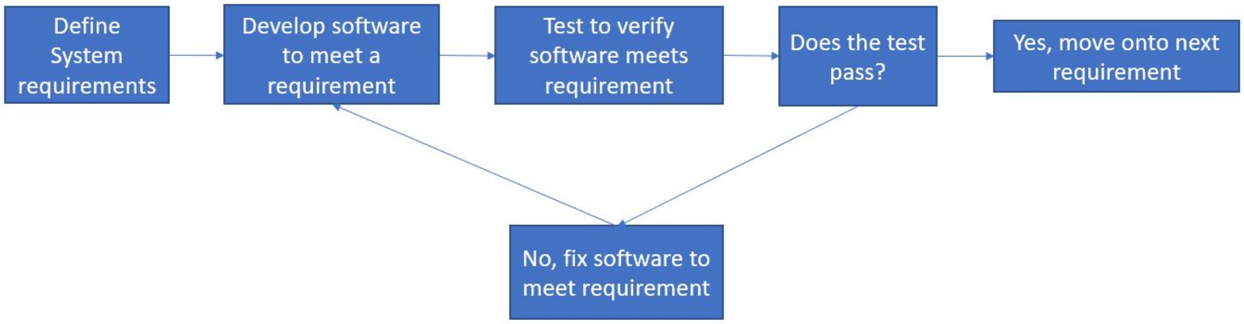
The team plans on doing requirements-driven testing. Our tests shall be designed to test for specific requirements. This way, we know we are not wasting time developing and testing for things that are not essential to the project. Requirements such as swiping a card to log in, an LED turning on, or a box opening will all need to be tested manually. For the project to be considered complete, it will need to pass the test 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. The website will also be tested on different operating systems and devices, to ensure our code is portable. 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 the hardware does not change as much as the software.



5.6 Results

TBD - We will have test results when the prototype is built.

6. Closing Material

6.1 Conclusion

We examined the past semesters projects in detail and analysed every single component and how they can be used in our design implementations. The past teams work can be used but there is a major problem from the past that we would need to resolve by ourselves. The One-Wire system is not functioning as per our teams research and the working success of the project. The circuit boards were already soldered and we have to get that major challenge fixed.

The goal for this semester would be to work on hardware implementation along with improvement in software handling. The mechanism box and the door handling is our next goal to accomplish along with a better UI.

6.2 References

** Image being used from past semesters team document

http://sdmay18-01.sd.ece.iastate.edu/documents/

Semester 1 Presentation