**Parts Crib Database System**

By: Ifeoluwa Adese, Mohand Ferawana & Tosin Ajayi

Discipline: Computer Engineering Technology  
Date: 26th February, 2018

# Declaration of Joint Authorship

We, ***Ifeoluwa Adese, Mohand Ferawana and Tosin Ajayi*** confirm that this work submitted for assessment is our own and is expressed in our own words. Any uses made within it of the works of any other author, in any form (ideas, equations, figures, texts, tables, programs), are properly acknowledged at the point of use. A list of the references used is included. In this project, Ifeoluwa Adese is responsible for the Software (the mobile app and the web application), Mohand Ferawana is responsible for the hardware and database management, and Tosin Ajayi is responsible for the database management.

# Approved Proposal

## Executive Summary

As a student in the Computer Engineering Technology program, I will be integrating the knowledge and skills I have learned from our program into this Internet of Things themed capstone project. This proposal requests the approval to build the hardware portion that will connect to a database as well as to a mobile device application. The internet connected hardware will include a custom PCB with the following sensors and actuators USB Barcode Scanner, Magnetic Strip, and PC2 Barcode Scanner. The database will store Information on the various available parts crib tools and equipment in a categorized manner. It will also store student information like names, student IDs, email and tools they have in possession. The mobile device functionality will include User Sign Up (i.e. student or admin), Ability to Search and Select Equipment Available at the Parts Crib, Display Record of Equipment already booked by the student, view all student holding a particular item, Edit and Approve Student Equipment Cart, Update Student Equipment Returns, Edit Item information, Display and Update Inventory etc. and will be further detailed in the mobile application proposal. I will be collaborating with the following company/department Humber College Applied Technology Parts Crib. In the winter semester I plan to form a group with the following students, who are also building similar hardware this term and working on the mobile application with me Mohand Ferawana and Tosin Ajayi. The hardware will be completed in CENG 317 Hardware Production Techniques independently and the application will be completed in CENG 319 Software Project. These will be integrated together in the subsequent term in CENG 355 Computer Systems Project as a member of a 2 or 3 student group.

## Background

The problem solved by this project is the frequent loss of materials to unidentified students and the cost of man power used in the Parts Crib Services as well as cost of papers used on a daily basis as a result of students having to manually list and sign out the items needed before borrowing them. A bit of background about this topic is simply the collection of the necessary student data like student names and IDs, email addresses etc. and associating them with randomly generated and registered barcode pins for easy student identification. These barcode pins will be the main information required to figure out which student has what equipment in possession. It simply implements a database consisting of different tables holding valuable information on the various available tools and equipment in a categorized manner, as well as administrative user and student user information. This will surely help improve accountability for materials in the Parts Crib, in order to keep good record of them.

Existing products on the market include [1]. I have searched for prior art via Humber’s IEEE subscription selecting “My Subscribed Content” [2] and have found and read [3] which provides insight into similar efforts.

In the Computer Engineering Technology program, we have learned about the following topics from the respective relevant courses:

* Java Docs from CENG 212 Programming Techniques in Java,
* Construction of circuits from CENG 215 Digital and Interfacing Systems,
* Rapid application development & Gantt charts from CENG216 Intro to Software Eng.
* Micro computing from CENG 252 Embedded Systems,
* SQL from CENG 254 Database with Java.
* Web access of databases from CENG 256 Internet Scripting; and,
* Wireless protocols such as 802.11 from TECH152 Telecom Networks.

This knowledge and skill set will enable me to build the subsystems and integrate them together as my capstone project.

## Concluding Remarks

This proposal presents a plan for providing an IoT solution for From an Administrative perspective, this project will be highly beneficial to the Parts Crib Employees in the sense that, it will heavily improve accountability for tools and equipment in the crib, just like a library keeping record of the books coming in and going out, it will also speed up the process of borrowing out equipment to students during peak hours for lab sessions. From the user perspective, it can also help provide students with information on the kind of tools and equipment available at the parts crib before their lab sessions, as well as keep a digital record of the parts crib tools they have in possession. This is an opportunity to integrate the knowledge and skills developed in our program to create a collaborative IoT capstone project demonstrating my ability to learn how to support projects such as the initiative described by [3]. I request approval of this project.

# Abstract

The project discussed in this report is a rental service system designed to improve the current rental process at the Applied Technology’s Parts Crib department, in the areas of time-consumption and resource management. The system simply enables students easily rent out the required materials for their upcoming lab sessions. It is an online system consisting of a mobile and web application as well as a remote database for fetching the necessary user or item information. The overall goal is to speed up the lending procedure at the part crib during peak lab hours, keep a monitored record of students with pending returns as well as an easy update of inventory record for all items.

The main idea behind having two separate platforms built to perform the same function is that the web application is designed only for administrative users i.e. the parts crib employees simply because most operation will be carried out on a desktop which is easier. However, both administrative users & registered students will be able to use the mobile application but predictably, it will be mainly used by students. This way, students are provided with an easy on-the-go access to their accounts, in order to keep them updated on available items and also make personal account updates from anywhere, at any time.

Table of Contents

[Declaration of Joint Authorship iii](#_Toc449009667)

[Approved Proposal v](#_Toc449009668)

[Executive Summary v](#_Toc449009669)

[Background v](#_Toc449009670)

[Concluding remarks vi](#_Toc449009671)

[Abstract vii](#_Toc449009672)

[1. Introduction 1](#_Toc449009674)

[2. Project Description 3](#_Toc449009675)

[2.1 Problem 3](#_Toc449009676)

[2.2 Rationale Behind Project 3](#_Toc449009677)

[2.3 Project Scope 3](#_Toc449009678)

[2.4 Software Requirement Specifications 4](#_Toc449009679)

[2.4.1 Database 4](#_Toc449009680)

[2.4.2 Mobile Application 5](#_Toc449009681)

[2.4.3 Web Application 5](#_Toc449009682)

[2.4.4 System Communication 6](#_Toc449009683)

[2.4.5 Hardware 6](#_Toc449009684)

[2.4.6 Operating Environment 7](#_Toc449009682)

[2.4.7 Design and Implementation 7](#_Toc449009683)

[2.4.8 User Documentation 7](#_Toc449009682)

[2.4.9 Assumptions and Dependencies 8](#_Toc449009683)

[2.5 Build Instructions 8](#_Toc449009685)

[2.5.1 Bill of Materials 8](#_Toc449009686)

[2.5.2 Time Commitment 9](#_Toc449009687)

[2.5.3 Mechanical Assembly](#_Toc449009688) 9

[2.5.4 System Design](#_Toc449009688) 10

[2.5.5 PCB and Soldering 10](#_Toc449009689)

[2.5.6 Power Up 12](#_Toc449009690)

[2.5.7 Unit Testing 13](#_Toc449009691)

[2.5.8 Production Testing 14](#_Toc449009692)

[2.6 Basic User Features 15](#_Toc449009693)

[2.7 Administrative Features 17](#_Toc449009699)

[2.8 Interface Design 18](#_Toc449009699)

[3. Progress Reports 20](#_Toc449009711)

[3.1 Report 1 20](#_Toc449009712)

[3.2 Report 2 20](#_Toc449009713)

[3.3 Report 3 22](#_Toc449009714)

[4. Conclusions 27](#_Toc449009718)

[5. Recommendations 28](#_Toc449009719)

[6. Technical References 30](#_Toc449009720)

[7. Appendicies 32](#_Toc449009721)

[7.1 Hardware Data Extraction Scripts 32](#_Toc449009722)

[7.2 Database Communication Program 38](#_Toc449009723)

[7.3 Database Input and Retrieval Scripts for Web and Mobile Applications 40](#_Toc449009724)

# Introduction

The Applied Technology Parts Crib department is presently operating on an exchange system, whereby students who need to borrow certain lab related materials are expected to provide some sort of college or government issued piece of identification in exchange for what they need. This is done with the expectation that the borrowed item will definitely be returned back in exchange for their personal possessions. However, this form of exchange system has happened to be not-so-efficient in recent times, especially in most cases when a college issued ID is provided by the student.

The major problems identified with the present system is the time and resources required. Usually during college’s open hours, there are specific times of the day when the parts crib experiences a traffic i.e. high number of requests by students needing to borrow specific materials for lab sessions. Now, there could be several lab sessions starting or ending during these “Peak hours” and there is usually only one parts crib employee available to attend to these large number of students. This creates room for possible errors by the employee and leaves the part crib highly vulnerable to loss of materials. In terms of resources used, we intend to save paper. The normal procedure requires that before any item can be given out, students, in maximum of twos, write out their item requests on a piece of paper and hand them over to the parts crib employee along with an ID card. This exactly, is responsible for the time wasted and the high traffic at the parts crib during lab hours, as well as a significant amount of money spent on paper by the college.

So, our proposed solution was to develop an online rental service system which includes a mobile and web application to digitize this exchange process, by providing students with the ability to easily and remotely prepare their item requests before arriving at the parts crib. After which in one click of a button, employees can approve these requests in a less time-consuming manner. The main objective to be achieved here, is to improve accountability for tools and equipment owned by the Parts Crib department. This platform will not only help identify areas of losses and students yet to return borrowed items upon due time, but also improve the accuracy of the inventory records.

# 2. Project Description

## 2.1 Problem

Apparently, the process of students signing out lab materials from the parts crib is highly time-consuming and requires some unnecessary resources e.g. use of too much paper. The inability to easily maintain an accurate inventory record and keep track of students with pending lab materials also seems to be a thing of concern.

## 2.2 Rationale behind Project

The proposed solution for facilitating an easier parts crib operation is to develop an online system to digitally carry out the item sign-off and return process as well as the inventory updates. The system includes a mobile application, a web application and two ID readers i.e. a USB bar-code scanner and a magnetic strip card reader.

## 2.3 Project Scope

There are two main softwares specified in this document; a mobile application and a web application. The web application is designed for both administrative users and registered student users but will be mainly used by administrators i.e. the parts crib employees, while the mobile application is also designed for both administrative users & registered student users but will be mainly used by students. The goal was to provide students with an easy on-the-go access to their accounts, in order to keep them updated on available tools and equipment and also make personal account updates from anywhere, at any time.

## 2.4 Software Requirement Specifications

### 2.4.1 Database

Both mobile and web applications are dependent on a common MySQL database, which was set up remotely with Host-gator; an online web hosting service. The database management tool provided was PHPMYADMIN. It has its own graphical user interface for creating and modifying databases and tables as well as a command line console for running queries as desired by the user. The role of the database in the overall system is to store basic user information like username, full name, email address etc and also inventory records such as the item name, serial number, total or available quantity etc. It also plays a major role in holding records of signed off items under specific student accounts.

### 2.4.2 Mobile Application

Our mobile application was developed specifically for only Android devices. It is compatible with tablets and phones with versions starting from android version 1.0 up to the latest version 8.1. The application is highly dependent on internet access which is required for almost every user activity. This is because it extracts most of its information from the online remote server. As stated above that the mobile application is designed for both administrative users & registered student users but will be mainly used by students. The reason is because there are certain functionalities that can be easily implemented into the web application but not into the mobile application. So administrative users are better off operating on the website than on the mobile application. The mobile application will allow students prepare their sign-off requests, which can be seen by an admin either on the web app or on the mobile app for approval or for modifications based on item returns.

### 2.4.3 Web Application

The web application basically provides a platform where administrative users can carry out specific tasks and daily operations like user registration, inventory updates, item sign-off and return procedures etc. Just like the mobile application, the web application is also highly dependent on internet access for its server requests and data exchange. It was built using HTML and CSS for the front-end design and Javascript for the behavioral aspect of the user interface, while other back end server-side languages were used like PHP and MySQL for database communications. From the overall project design, the hardware only fits into the system when used on the web application for student authentication and item scanning.

### 2.4.4 System Communication

From the overall project design, the mobile and web applications are both set up to interact with the same online database. However, the mobile application will function independently to exchange information with the server, while the web application hosted on the same remote server will sometimes, although not necessarily, require the hardware for scanning student IDs, which is somewhat faster.

### 2.4.5 Hardware

Utilization of the hardware only requires a text field on an application running on a device with HID keyboard recognition. The hardware simply decodes raw the data from scanned IDs, performs parity-checks and prints out the decoded information. This happens to work even better on a web application. Testing the hardware on Google’s search engine proved that the hardware does not only scan data but the submit function is also automatically triggered, which means the feedback is received almost immediately.

### 2.4.6 Operating Environment

The mobile application will be compatible with only android devices (tablets and phones) with versions starting from 1.0 up to the latest version 8.1. The web application will also be compatible with all desktops and personal computers using any browser on any kind of operating system. The hardware will also be compatible with any computer, provided there is port that suits the device’s connection cable. For example, the USB barcode scanner will have to be plugged into a USB port on the computer running the web application to function. A stable internet connection will also be required for both mobile and web application to connect and interact with the online database.

### 2.4.7. Design and Implementation Constraints

In terms of design and implementation constraints, the only major limitations developers might encounter will be the implementation of a non-relational database into the system design. There are specific PHP/SQL scripts that carry out the main exchange of data in a specific format between the mobile/web application and the database. So, implementing a non-relational/no-SQL database might cause certain complications and force developers to undergo an intensive system redesign. Other less threating limitations include the fact that the mobile application is only available for android users and only supports two languages i.e. English and French.

### 2.4.8 User Documentation

A descriptive video tutorial, a wire-framed documentation and a well detailed FAQ, for both web and mobile applications will be provided to answer possible questions or any predicted issues that may arise from users. All three user documentations will be provided mainly in the help section of the web application and also made accessible from the mobile application by simply redirecting users upon selection.

### 2.4.9 Assumptions and Dependencies

Both mobile and web applications are highly dependent on a bunch server-side scripts written in PHP and SQL. These scripts are hosted on an online remote server purchased from HostGator; a company that provides servers for online web hosting as well as database management and web design tools like MySQL, PhpMyAdmin, File Manager etc. These scripts are mainly responsible for the exchange of data between the mobile/web application and the provided database. Without these scripts, absolutely nothing can function. In a scenario where by the implemented database is to be replaced, then these scripts will also have to be altered in some areas to function properly.

## 2.5 Build Instructions

### 2.5.1 Bill of Materials

* [USB Barcode Scanner - $87.2](http://www.amazon.ca/Arduino-A000066-Uno-R3-Microcontroller/dp/B008GRTSV6/ref=sr_1_1?ie=UTF8&qid=1449726852&sr=8-1&keywords=arduino+uno)5
* [Magnetic Strip Card Reader - $28.69](http://www.amazon.ca/Arduino-A000066-Uno-R3-Microcontroller/dp/B008GRTSV6/ref=sr_1_1?ie=UTF8&qid=1449726852&sr=8-1&keywords=arduino+uno)
* [Raspberry Pi 3 starter kit - $74.95](http://www.amazon.ca/gp/product/B00MV6TAJI?psc=1&redirect=true&ref_=oh_aui_detailpage_o05_s00)

These costs may differ based on currency, time, and supplier.

### 2.5.2 Time Commitment

The estimated duration of time it could take one to reproduce this whole system from scratch could be as short as a month, provided there are no issues encountered and everything works perfectly well. In our development process it took about a week to conceptually design the database structure and decide the kind of information stored and how the relationships between items and users could be set up and just a day for the implementation and testing because it wasn't really a big structured database. The mobile application took about a month, while the web application which is in progress is estimated to take about 2 months, due to recent technical challenges encountered that are slowing down the development process.

### 2.5.3 Mechanical Assembly

The nature of this system happens to be more software inclined and majority of the work done is mostly software development i.e. the mobile and web apps, even the database too. So, in terms of mechanical assembly, there was absolutely no need for building or setting up any complex hardware or writing code to alter the functionality of the hardware. Our mechanical assembly is as simple as connecting the USB barcode scanner and the Magnetic stripe card reader into a USB port, and they’re ready to go.

### 2.5.4 System Design

This section gives a more detailed description of how the whole system was designed and set up and from the very beginning, including the key roles played by each group member in the development process. The system includes four major parts listed and explained below:

1. **Hardware Parts**: The system uses two different hardware materials for two different purposes. The first material is the USB barcode scanner, which is by default, designed to scan specific barcode IDs and print out the data into any text field. We simply used this device for scanning registered barcode IDs assigned to individual crib items, during the item check out process. In terms of powering up this device, it just needs to be plugged into a USB port on the device running the web application, and it’s ready to be used.

The second material is the Magnetic Stripe Card Reader. This device functions similarly to the USB barcode scanner. However, it requires physical contact with the card holding the information. The card has to be swiped in order to get the data extracted and printed into a text field. We used this in the user authentication process, for extracting data from student cards and processing the raw data to get the exact student ID number. The magnetic stripe also needs to be plugged in to the device running the web application in order to be powered on.

Mohand Ferawana is solely responsible for handling the extraction and processing of data from the Magnetic stripe and the USB barcode scanner, using server-side PHP scripting. He is also responsible for designing the casing for the USB barcode scanner module. The barcode scanner package arrived as a raw uncovered part, which was very prone to destruction. The casing was designed using CorelDraw X6, and laser cut at the prototype lab with the help of the Lab assistants i.e. Vlad and Kelly. A photo of the newly covered USB barcode scanner is shown below. (Photo not Included Yet).

1. **Remote Server and Online Database**: The remote server is provided by HostGator; an online domain provider and web hosting company. One of the most important tools provided by the company for our web development process is File Manager. It is simply an online directory that used to host source code files and templates related to the hosted website. So, we used this tool in storing our development files for the web application as well as the scripts required for handling exchange of data between the database and the mobile or web application.

The database management tool provided by HostGator is PHPMyAdmin. Before setting up the database, the team did a lot of paper work during the conceptual design process. We agreed on the kind of information to be stored as well as their relationships before beginning the set-up stage. Our database consists of 4 tables listed and described below:

* **User Information Table**: This table holds data about users e.g. Username, first name, last name, password, email, user type etc. It is mainly used during the user login and registration process and also during user profile update.
* **Item Information Table**: This table holds data about items e.g. item name, serial number, total quantity, available quantity, rented quantity etc. It is mainly used during the item check-out and return process and also during general inventory or specific item info update.
* **Cart Information Table**: This table holds data on the relationship between items and users in rows and columns. For instance, when crib items are added to a student cart, specific indexes in this table are updated where the item row intersects with the user column.
* **Rental Information Table**: This table also holds data the relationship between items and users in rows and columns. However, it is only updated when a student cart items are approved by an admin user. In this part of the project, Ifeoluwa David is responsible for the database design, while Tosin Ajayi is responsible for the set up and implementation using specifications provided by David.

1. **Mobile and Web Application**: The mobile application was developed using Android Studio. The main development language used was Java. Its an app compatible with all Android device versions from 1.0 to 8.0, which can be accessed on the Google Play Store. The web application, which is also compatible with any browser i.e. Google Chrome, Mozilla Firefox, Microsoft Edge or Internet Explorer, Safari and so on, was developed using HostGator’s File Manager Online Code Editor tool. The main development languages used were HTML, CSS, PHP and JavaScript. It can be accessed online at [Partscribdatabase.tech](http://partscribdatabase.tech/)

The development process for both mobile and web applications were broken into three main parts for each group member to take responsibility. These parts include: the user authentication process (User Login, Registration and Profile Settings) handled by Tosin Ajayi, the item check-out and return process handled by Ifeoluwa David, and finally the item info and inventory update process handled by Mohand Ferawana. These 3 parts are the most important functionalities that sum up the overall purpose of the system. Both mobile and web apps are also designed to interact with the same online database.

### 2.5.5 PCB and Soldering

During the week of the PCB and Soldering deliverable, a schematic was provided which was expected to be edited and 3D printed. A step by step procedure was also given, as well as some materials for the first soldering project. Upon completion, some testings were done to check for short and open circuits in the finished product, before mounting the board on the Pi. Afterwards, we remotely accessed some repository files via the Pi’s command line. These files were test programs that had to be modified before execution using various sensors (i.e. temperature reading, light etc.) Upon successful execution, observation of the behaviour of the program and its effects on the mounted PCB board’s LEDs, we could then conclude that the raspberry Pi was perfectly functional. The main goal of this phase was to help the team familiarize with the raspberry Pi’s OS interface and its command line environment.

### 2.5.6 Power Up

The main required hardware materials in this system are the USB bar code scanner and the Magnetic strip card reader. The power up process for these devices are as simple as plugging them in to USB ports on the device hosting the web application. In this case, the raspberry pi is the "model" hosting device with its own operating system, on which we can use its browser to access the web application developed specifically for this project. On the software side, the web app can be accessed online using the designated domain name partscribdatabase.tech. And finally, the mobile application can be accessed by downloading it online off the Google Play app store. The database is always functional, provided the server subscription is renewed on a monthly basis.

### 2.5.7 Unit Testing

Each subsystem was individually tested before integration. At the time the database was set up online, we tested it separately by using SQL queries and PHP scripts for connection at first, then inserting, deleting and modifying information in the database, and it was successful, so it was clear that the database worked perfectly fine without any issues. Further unit testing was done on the mobile application and the web application in the development process to ensure that every module was independently functional before syncing the various modules together and setting up a connection between both applications and the server. The various software modules tested in both web and mobile applications include the user authentication and profile settings module, the item sign-off and returns module as well as the inventory update aspect.

### 2.5.8 Production Testing

Here, we set up each subsystem to function together i.e. the apps, the hardware and the database. And we can confirm that the production testing was a successful procedure because the changes made in the database from the web application, reflected on the mobile application and vice versa. During the production testing, the first step was to populate the database with fake user information and mock crib items. Then we tested the user authentication process by using the fake user IDs that were manually entered into the database. Other phases of the production testing were editing user profile settings, signing off and returning mock crib items, as well as inventory update activities. These testings were done on both mobile and web applications.

## 2.6 Basic User Features

The basic user features are the specific functionalities student have access to on the mobile application. They include viewing list of available items, selecting and adding items to cart as well as editing their carts. Other basic functions students have access to include edit profile settings and change user password.

## 2.7 Administrative Features

The administrative users have access to a variety of functions on both the mobile application and the web application. This provides them with a lot more information compared to basic student users. These functions include checking-out items and performing item returns for students, user registration and removals, update item information, view student cart and current student possessions, view daily reports on pending student returns and so on. They also have access to other basic user functionalities like edit profile settings and change user password.

## 2.8 Interface Design

The user interface design will vary on both mobile and web platforms. On the mobile application, the administrative home screen will be a sectional menu holding the most important admin functions, while the student home screen is a direct access to list of available tools and equipment as well as buttons that redirect the user to view cart and view personal possession. A navigation drawer will also be present at the top right-hand corner of every screen on the mobile application for both administrative and student users. This navigation drawer will contain other frequently needed feature options common between both user types for easy access. These options include; return to home, edit profile, change password, logout, view student cart, view student possession etc.

On the web application, both user types will be operating mainly on simple web page with a leftsided navigation drawer covering about one-fourth of the screen. Again, the idea with the navigation drawer is simply to provide the user with easy access to a list of frequently needed feature options. The remaining three fourths of the screen will be the main screen for operational activities by the user. A user interface specification describing style design and layouts for each functionality screen can be provided upon request.

The hardware only requires one interface, which is a text field; any text field. The text field has to be on an application running on a device with HID keyboard recognition. The major function of the hardware when an ID is scanned is to decode raw the data, perform a parity-check and then print out the decoded information. This process happens to work even better on a web application. When the hardware i.e. the barcode scanner, was tested on Google’s search engine, the data was not only scanned but the search function was also automatically triggered. Same with our web application, when a valid ID is scanned, the enter function is also automatically triggered without the user having to click a button, so the server feedback is received almost immediately. Nothing else asides a text field on a device with HID keyboard recognition, is required to utilize the hardware.

The whole system is highly dependent on transmission of data between the mobile/web application and the remote server. On the mobile side, the data sent to the server is usually encoded in raw data, but after the message is interpreted, feedback is received in JSON format. The data sharing method is usually carried out on a separate thread from the main thread in order to keep the data flow asynchronous. On the web application side, the user interface and data transmission scripting are all done on the server.

The only communication functions required by this product on the mobile application side is the HTTP network protocol. The httpurlconnection API available on android platforms enabled remote connection to the server from any mobile device. On the web application side, no protocols were required because connecting to the database directly was much easier using server-hosted PHP scripts. However, electronic forms were required on the web application in communicating with users on the kind of information needed.

# 3. Progress Reports

## 3.1 Report 1

|  |  |
| --- | --- |
| From: Ifeoluwa David[ifeoluwaadese@ymail.com](mailto:ifeoluwaadese@ymail.com) | Monday, Mar 5, 2018 at 6:42 AM |
| To: Kristian Medri [Kristian.Medri@humber.ca](mailto:Kristian.Medri@humber.ca), Austin Tian [Austin.tian@humber.ca](mailto:Austin.tian@humber.ca)  Cc:Mohand Ferawana [m.ferawana@gmail.com](mailto:m.ferawana@gmail.com), Tosin Ajayi [ajayi.oluwatosin5@gmail.com](mailto:ajayi.oluwatosin5@gmail.com) | |
| |  | | --- | | Dear Austin,  The team, Austin and Vlad had a discussion last week concerning the parts crib’s preference on which of our 2 different identification readers will be more convenient for the student authentication process. We tried to decide between Mohand’s magnetic strip card reader (Swipe) and David’s USB Barcode reader (Scan). We later came to the conclusion that both readers could actually be utilized for specific purposes as described below:   * USB Barcode Scanner/Reader: For scanning individual items into the web app. * Magnetic Strip Card Reader: For student authentication.   **Mohand Ferawana**: Mohand worked on the Magnetic strip card reader. This is implemented in the first stage of the main operations procedure for User authentication. Basically, he created a page on the web application where administrative users can swipe student cards from a text field and auto trigger a server request for valid and existing user information before proceeding to the second stage of the main operations procedure which is David’s item scanning/selection functionality. He made this possible by extracting the data needed from the student card in order to query the database through the web application. After which the returned information is presented in the web page for user verification. He successfully completed this functionality on the 4th of March and no major problems have been encountered so far. This week, during the group meeting, the team will decide the next step to focus on.  ***[NOTE] “Main Operations” is a 3-step procedure we’re implementing into our web application in order to complete the item check-out process. These three steps include User Authentication, Item scanning/selection and Quantity Selection (Summary) and Approval.***  **Tosin Ajayi**: Tosin’s role in the project is to set up the user login, registration and profile settings functionality. This simply allows administrative users to login into the system, register students, make changes to their personal profile as well as carry out inventory update activities. He successfully completed the login and registration functionality; however, he’s currently working on implementing an input validation process for the user registration and profile update aspect. This will ensure that the data entered is valid enough to be entered into the database. Based off of Austin’s email and the agreement with the parts crib, the group has also decided that upon completion of this stage, Mohand will be taking over the registration process, in order to implement the student card authentication feature using the magnetic stripe card reader. Simply because every card’s data needs to be extracted and registered at first, before it can be considered valid for user authentication during the item check-out process.  **Ifeoluwa David**: Personally, the focus has been on facilitating the selection of items in the system using the USB barcode scanner. Instead of programmatically extracting a list of items from the server for user selection through a long list of categorically arranged checkboxes and what not, I decided with my team to keep the web application’s functionalities strictly administrative. I successfully worked on the front and back end of the second stage of the main operations procedure which simply involves the scanning of registered barcodes on an item, and the corresponding item name for that specific barcode will be returned from the server into a cumulative list box on the web page. At the moment, I am currently at a stage where I’m trying to ensure this list box of scanned items is modifiable by implementing a way to delete specific indexes in order to make the item selection process a bit more flexible. | | |

## 3.2 Report 2

|  |  |
| --- | --- |
| From: Mohand Ferawana [m.ferawana@gmail.com](mailto:m.ferawana@gmail.com) | Monday, Mar 19, 2018 at 8:52 AM |
| To: Kristian Medri [Kristian.Medri@humber.ca](mailto:Kristian.Medri@humber.ca), Austin Tian [Austin.tian@humber.ca](mailto:Austin.tian@humber.ca)  Cc: Ife David[ifeoluwaadese@ymail.com](mailto:ifeoluwaadese@ymail.com), Tosin Ajayi [ajayi.oluwatosin5@gmail.com](mailto:ajayi.oluwatosin5@gmail.com)  **Ifeoluwa David**: Last week, David simply focused on fixing the bug found on the second stage of the main operations. The second stage of the main operations is the part where admins have the ability to scan the barcodes off crib items into the system and actually view the items in a list box. That feature worked properly, but there were certain issues experienced while trying to implement another feature that allows a user delete an item from the list. It was easy to implement, but some things were responsible for not letting it work properly as expected. So, it took about the whole week to find a solution. David has also been working collaboratively with Tosin on the last stage of the Main operations process which is the item quantity and check-out summary.  **Mohand Ferawana**: Mohand worked on implementing the functionality that allows a user swipe the college issued ID card in the registration part. The observation was that if the data from the card was required in the item sign off process for identity verification, then it must have to be registered first in the user registration process to be considered valid or invalid. So, Mohand had to find a means of extracting and processing the data from the student card before having it registered in the database. This way, when a student tries to sign-off an item, the user identity can be verified first, by swiping the student card before proceeding. Also, recently, Mohand also designed a laser-cut plastic casing for the USB barcode scanner hardware module, which will protect the device from easy destruction.  **Tosin Ajayi**: So far, in the development process, Tosin worked on moving the user registration functionality into the admin menu. At first, a hyperlink was included on the login page, which allows anybody to easily gain access to the user registration page and create an admin or student account for themselves, which was not the initial agreement. So, the plan was to move that feature into the admin menu, so only administrators can have access to that feature to register new users, be it student or admin. Tosin also further worked on the input validation process for the user registration and profile update over the past week, and that works perfectly now. Tosin is presently working collaboratively with David on the last stage of the main operations process which is the item quantity and check-out summary. | |

## 3.3 Report 3

|  |  |
| --- | --- |
| From: Tosin Ajayi [ajayi.oluwatosin5@gmail.com](mailto:ajayi.oluwatosin5@gmail.com) | Monday, Apr 2, 2018 at 10:00 AM |
| To: Kristian Medri [Kristian.Medri@humber.ca](mailto:Kristian.Medri@humber.ca), Austin Tian [Austin.tian@humber.ca](mailto:Austin.tian@humber.ca)  Cc: Ife David[ifeoluwaadese@ymail.com](mailto:ifeoluwaadese@ymail.com), Mohand Ferawana [m.ferawana@gmail.com](mailto:m.ferawana@gmail.com) | |

**Ifeoluwa David**: This week, David focused on the front-end development for the item-returns functionality and is currently working on the back-end functionality. The item returns feature basically allows administrative users to view pre-selected items in student carts and approve them to become checked-out items. It also allows admin users view a student’s current item possessions and return specific items based on provided quantity. Upon completion of this feature, David will be working on completing the update inventory functionality.

**Mohand Ferawana**: Mohand simply worked on the item registration feature. He added this feature into the 3-tabbed registration page which includes: register student, register admin and register item. Right now, this feature is complete and individual crib items can be registered into the system database. Next job for Mohand is to work on implementing the item removal feature, allowing users entirely delete registered items from the databasAe.

**Tosin Ajayi**: Tosin worked on the change password feature, this allows users change their current password to a new password simply by providing and confirming the current password and also entering and confirming the newly intended password. This feature will be important for users in order to keep their account secure and maintain private access. After completing this part of the project, Tosin will be moving on to join Mohand on a removal functionality for admin and student users, allowing a single “top” admin user to delete another admin user or student user. This will be needed in order to maintain a small amount of user data in the database.

# 4. Conclusions

From the details given in this report, in the areas of problem identification, systems design, testing and implementation the following conclusions can be drawn:

* The main objective of the project is to speed up the item check-out process and also improve accountability for crib items.
* The overall system consists of 4 main parts i.e. the hardware devices, the online database, the web application and the mobile application.
* Both mobile and web applications interact with the same database which holds detailed information on personal user data, item record data as well as the “user-item” relationships.
* The hardware devices i.e. the barcode scanner and the magnetic stripe card reader, respectively help authenticate student identity and interpret specific item barcodes into complete item information.
* The web application is designed mainly for administrative use, while the mobile application is designed for both student users and administrative users.
* Each subsystem has been thoroughly tested individually before full integration.

# Recommendations

Although there are a few limitations involved in the successful implementation of this project, some recommendations can still come in handy as solutions for them. At the moment, the whole system runs off an online database hosted externally. The school will require a system that uses a database privately hosted by the college in order to ensure the user data protection law rules are followed. This is a mandatory policy required in every institution or company with a user-based system. The authorization of addition of new tables for the implementation of the project will take about a semester. Upon successful approval, item and user data can then be manually migrated from one database to another in probably CSV formats. However, this might cause for an intensive back-end redevelopment on both web and mobile applications.

# 6. Technical References

User Interface Design Basics. (2014, May 21). Retrieved from <https://www.usability.gov/what-and-why/user-interface-design.html>

Developing a Car Rental Website [Case Study]. (2016, December 27). Retrieved from <https://lunapps.com/blog/developing-car-rental-website/>

11 Web Application Security Best Practices. (2017, January 09). Retrieved from <https://www.keycdn.com/blog/web-application-security-best-practices/>

J. G. (2016, March). Principles of Mobile App Design: Introduction. Retrieved from <https://www.thinkwithgoogle.com/marketing-resources/experience-design/principles-of-mobile-app-design-introduction/>

# 7. Appendicies

## 7.1 Hardware Data Extraction Scripts

The web application, in some specific areas, are dependent on the data extraction carried out by some scripts in order to get the information required to retrieve valid data from the online database. On the web application, the USB barcode scanner is used mainly to scan item barcodes for the purpose of simply identifying the item or fetching all the necessary information needed by the administrator. The magnetic strip card reader is also used on the web application mainly for student authentication. When a student card is swiped, a certain script does the extraction of valid data (a student number) from a bunch of lightly encrypted information. This way, the extracted student ID is used in the validation of the student identity during an item check-out or return process. All the user has to do is just confirm if the returned identity information is correct or not.

## 7.2 Database Communication Program

Apart from the data extraction scripts, there are some other scripts hosted on the server written mainly for the purpose of exchanging data back and forth between the software applications and the online database. These scripts are written in PHP with some JSON formatting and SQL queries and wherever needed. Both mobile and web applications are highly dependent on internet access in order to connect to the server and operate properly as expected, this is because almost every major activity carried out by the user on the software applications are indirectly scripts being executed on the server.

## 7.3 Database Input and Retrieval Script for Web and Mobile Application

Apart from the data extraction scripts, there are some other scripts hosted on the server written mainly for the purpose of exchanging data back and forth between the software and the database. The development process for the web application was also done in such a way that the logical code is separated from the interface design code. However, the same logic scripts are utilized by the mobile application when communicating with the online database. This is done in order to ensure that the back-end functionalities in both applications operate in the exact same manner and also to avoid fixing possible bugs in multiple back end code files. The logical scripts used by both applications for interacting with the database are written in basic PHP code and some SQL queries without the use of any other frameworks or dependencies while the interface is written in simple HTML and Javascript for its front-end responsiveness.

Source codes for both mobile & web applications can be found in the GitHub links provided below:

* Web Application: <https://github.com/IfeoluwaDavid/PCD-Web-App-Project>
* Mobile Application: <https://github.com/IfeoluwaDavid/PCD-Mobile-App-Project>