25x8X?

Revision date: 31st, Mar 2017

Prototype Lab Flow Project

Projects website: https://mndk0076.github.io/

Declaration of Joint Authorship

The work defined in this report is a joint collaboration project by Sukhdeep Sehra, Gurpreet Jhita, Matheus Almeida, and Kenneth Mendoza. The work specified is solely of our own creation and is expressed in our own words. We have clearly referenced all work with proper APA citatations with respect to the authors of the information. All work in this project is equally divided. Kenneth Mendoza was in charge of the development of the database, Sukhdeep Sehra was in charge of the design and construction of the hardware, Matheus Almeida worked on the overall layout and skeleton code of the mobile application and Gurpreet Jhita was in charge of designing and coding the admin website. Everyone has collaborated in all aspects of the deveplopent.

Approved Proposal

Proposal for the Prototype Lab Flow Project

Prepared by Kenneth Mendoza, Sukhdeep Sehra, Matheus Rondinelli and Gurpreet Jhita *Computer Engineering Technology Students*
https://mndk0076.github.io/

Executive Summary

As students in the Computer Engineering Technology program, we will be integrating the knowledge and skills we 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 sensors and actuators for LCD monitor and a scanner. The database will store all of the items from the parts crib for availability of the parts, and all of the users and admins info from the mobile app. The mobile device functionality will include a mobile app for users to request items, a LCD monitor that display the users requested items, a scanner to scan the user mobile app when returning the items, a mobile app for the users to send request items and will be further detailed in the mobile application proposal. We will be collaborating with Humber College Technology Department. In the winter semester we plan to form a group with the following students; Kenneth Mendoza, Sukhdeep Sehra, Matheus Rondinelli and Gurpreet Jhita, who are also building similar hardware this term and working on the mobile application. 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.

Background

Currently, we students have no way of booking time slots or requesting items from the prototype lab through a convenient application. Students have to go personally to the prototype lab and book the time or hope the item required is available. As a team, we feel like this is an inconvenient experience for the students and feel like an automated system would help to improve the current system by making it more efficient and more appealing to the student. Students would be able to book times slots and request items through the application and they would also be able to cancel requested time slots or items, freeing the space or item for another student and without having to go through the lab personally. That would also improve the time management of the staff, since they will not have to manage the times by themselves while having to work and help other students present in the prototype lab.

We have searched for prior art via Humber's IEEE subscription selecting "My Subscribed Content" and have found and read which provides insight into similar efforts.

The first article discusses how Gabor used QR decomposition for image encoding.(Lau, Papanikolopoulos, & Boley, 1993)

The second article discusses how support system for QR codes based educational processes.(Lezhebokov, Kravchenko, & Bova, 2014)

The third article discusses how VLSI implementation of hybrid QR code generation system. (Ramya & Sheela, 2014)

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 and Gantt charts from CENG 216 Intro to Software Engineering,
- 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.

Methodology

This proposal is assigned in the first week of class and is due at the beginning of class in the second week of the fall semester. My coursework will focus on the first two of the 3 phases of this project:
Phase 1 Hardware build.
Phase 2 System integration.
Phase 3 Demonstration to future employers.

Phase 1 Hardware build The hardware build will be completed in the fall term. It will fit within the CENG Project maximum dimensions of 12 13/16” x 6” x 2 7/8” (32.5cm x 15.25cm x 7.25cm) which represents the space below the tray in the parts kit. The highest AC voltage that will be used is 16Vrms from a wall adaptor from which +/- 15V or as high as 45 VDC can be obtained. Maximum power consumption will be 20 Watts.

Phase 2 System integration The system integration will be completed in the fall term.

Phase 3 Demonstration to future employers This project will showcase the knowledge and skills that I have learned to potential employers.

The tables below provide rough effort and non-labour estimates respectively for each phase. A Gantt chart will be added by week 3 to provide more project schedule details and a more complete budget will be added by week 4. It is important to start tasks as soon as possible to be able to meet deadlines.

Labour Estimates	**Hrs**	**Notes**
Phase 1		
Writing proposal.	9	Tech identification quiz.
Creating project schedule. Initial project team meeting.	9	Proposal due.
Creating budget. Status Meeting.	9	Project Schedule due.
Acquiring components and writing progress report.	9	Budget due.
Mechanical assembly and writing progress report. Status Meeting.	9	Progress Report due (components acquired milestone).
PCB fabrication.	9	Progress Report due (Mechanical Assembly milestone).
Interface wiring, Placard design, Status Meeting.	9	PCB Due (power up milestone).
Preparing for demonstration.	9	Placard due.

Writing progress report and demonstrating project.	9	Progress Report due (Demonstrations at Open House Saturday, November 7, 2015 from 10 a.m. - 2 p.m.).
Editing build video.	9	Peer grading of demonstrations due.
Incorporation of feedback from demonstration and writing progress report. Status Meeting.	9	30 second build video due.
Practice presentations	9	Progress Report due.
1st round of Presentations, Collaborators present.	9	Presentation PowerPoint file due.
2nd round of Presentations	9	Build instructions up due.
Project videos, Status Meeting.	9	30 second script due.
Phase 1 Total	**135**	
Phase 2		
Meet with collaborators	9	Status Meeting
Initial integration.	9	Progress Report
Meet with collaborators	9	Status Meeting
Testing.	9	Progress Report
Meet with collaborators	9	Status Meeting
Meet with collaborators	9	Status Meeting
Incorporation of feedback.	9	Progress Report
Meet with collaborators	9	Status Meeting
Testing.	9	Progress Report
Meet with collaborators	9	Status Meeting
Prepare for demonstration.	9	Progress Report
Complete presentation.	9	Demonstration at Open House Saturday, April 9, 2016 10 a.m. to 2 p.m.
Complete final report. 1st round of Presentations.	9	Presentation PowerPoint file due.
Write video script. 2nd round of Presentations, delivery of project.	9	Final written report including final budget and record of expenditures, covering both this semester and the previous semester.
Project videos.	9	Video script due
Phase 2 Total	**135**	
Phase 3		
Interviews	TBD	
Phase 3 Total	**TBD**	

Material Estimates	**Cost**	**Notes**
Phase 1		
A microcomputer composed of a quad-core Windows 10 IoT core compatible Broadcom BCM2836 SoC with a 900MHz Application ARM Cortex-A7 32 bit RISC v7-A processor core stacked under 1GB of 450MHz SDRAM, 10/100 Mbit/s Ethernet, GPIO, UART, I^{2}C bus, SPI bus, 8 GB of Secure Digital storage, a power supply, and a USB Wi-Fi adaptor.	>$80.00	An example of a retailer:
Peripherals with cables		
Sensors		
Actuators		
Hardware, etc.		
Phase 1 Total	**>$200.00**	
Phase 2		
Materials to improve functionality, fit, and finish of project.		
Phase 2 Total	**TBD**	

Phase 3	
Off campus colocation	<$100.00
Shipping	*TBD*
Tax	*TBD*
Duty	*TBD*
Phase 3 Total	**TBD**

Concluding Remarks

This proposal presents a plan for providing an IoT solution for *Future Humber PrototypeLab Flow*. 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. I request approval of this project.

Abstract

The purpose of the prototype lab flow project is to automate a system that is outdated and by doing so, make it more efficient and appealing to students.The Prototype lab at the momment is using ancient technology involving paper and pen, with our project students will be able to do the requests and check availability all through their mobile devices, without having to show up in the prototype lab and not be able to use its services. Our hardware is going to run a camera to scan a QR code generated by our mobile application. Each student will have a QR code generated to him/her that is going to be scanned when entering the lab. The system then will store the information on the database allowing the administrator to have more control over who and when they have been in the lab.

Table of Contents

List of Illustration/Diagrams

1. Introduction

The objective of our project is to develop a better method for requesting time slots in the Prototype Lab. Currently the Prototype lab does not have a very good method for requesting time slots and booking out equipment, they currently use a clipboard method. Using the skills we learned throughout the last 5 semesters, we're going to make the experience for the students a more pleasant and efficient one, by being able to book time in the Prototype Lab off a mobile application and also add more functionality which also makes the experience better. The types of additional functionality will include being able to book, cancel and reschedule time slots, being able to also request tools and parts that are only available in the Prototype Lab as rental equipment. Students will use the web application to book time slots in the Prototype Lab. Our application will also work in conjunction with a QR code scanning device. As an addition to making the experience better for the students, we are also working on a webpage for the administrator, allowing them to manage time slots and collect specific data. This Technical report will illustrate the work involved in bringing our vision to life.

2. Project Description

2.1 Product Introduction

2.1.1 Purpose

The main goal of our project is to simplify the process of requesting items and booking time slots at the Humber Prototype Lab. Up until now, students have no control on whether there are any items available (or any time slots to fill up) at the lab. And the lab administrators have no way of booking time slots and sometimes have to deal with high traffic of students, without any real control. We feel like the system is outdated and needs to be updated for the fast-paced world of today. Therefore, we are developing the Lab Flow project, which is an mobile application that aims to speed up this process by allowing students to book time slots through their mobile devices as well as providing more control to the lab administrators, effectively increasing the time efficiency and organization of the lab.

2.1.2 Scope

In this project, we will create a mobile application which allows the students to register their accounts, check availability of items and time slots at the Prototype Lab, and request them. Our database will store each user's information like their name, student number, program of study etc. Furthermore, it will also include all the time slots and all the items requested by students. Lab monitors will have a special account where they will be able to check the booked times and manage them, if necessary. They will also have additional statistics presented to them in charts and graphs showing them what days and times most of the time slots are getting filled up the most and which items are getting requested the most.

2.1.3 Product Overview

We will be using a Raspberry Pi 3, a digital camera and a monitor. These three components will make up the majority of our hardware that will be used to scan student QR codes. Our product will consist mostly of a software application and webpage synced to a database.

2.2 Software Requirements Specification

2.2.1 Database

Our main goal with the database is to make it store the user's information, mainly user's student number so that it can be assigned to the corresponding item or time. It will also have all items available in the prototype lab, such as raspberry pi and ardoino so that students may be able to request them. The main function of the database will be to store the time slots that users will be able to reserve in order to use the prototype lab. Once the time is reserved by a student, its student number will get assigned to that time slot and will not be able to be requested by any other student. In order to cancel the time reservation the student will have to do so 24 hours prior to the time reserved. By accessing the student information stored in the database, we will also be able to limit access to the prototype lab as requested by Kelly. The access limit will be done as requested by the prototype lab, and will be done by group such as computer engineering, electrical, design and we will be able to restrict a day or a certain amount of time to that specific group. We will be using MySQL to create the database since we are already familiar with it from our database course in semester 4 and to host the database after doing some research we decided that hostinger would be the best available option. We will be using their free plan which has a limit of 250 visitors a day and the bandwidth is about 100gb, but we came to the conclusion that it should be enough for now just to implement the system and test it and if needed in the future the plan can be changed to a paid one that will offer a higher visitor limit and more bandwidth. Kenneth will be in charge of the database.

2.2.2 Web Interface

Our web application is going to allow the student to register and/or log in to his account. The application will allow students to book a time through the web page and also check history of items or booked time slots in the past. The admin will have access to the student's information and to the info graphs just as in the app. Development of the website will rely heavily on how well we are doing with the application. The more progress we can achieve on it the more time we will have to develop the website and add more functionalities to it. Gurpreet will be in charge of the web page. Meanwhile, Sukhdeep will be working the hardware.

2.2.3 Hardware

Our hardware will be a scanning device used to check-in to the prototype lab. The database will be linked to our scanning device which will allow the student to check-in by just scanning their student id. If the student arrives too early the system will not allow the student to check-in until the time slot is available. If the student does not register for a time slot through the application the device will not allow the student to check-in and will request the student to register into the next available time slot. We will be using a high definition camera to scan the QRs faster and more precisely. Sukhdeep also developed a case for the hardware where the camera will be fixed and there will also be a fixed part to indicate where the student have to display the card and therefore not allowing the student to display it too far from the camera and always displaying at the same distance, making the scan much easier and more reliable.

2.2.4 Application

Our application will allow the user to register and by doing so adding himself to our database. Once registered, the student will be able to access the other classes and request time slots and items from the prototype lab. He will also be able to see all the time slots available and the ones already taken, he will also be able to see if certain time period or day is reserved to certain group. Once an item or time is requested by student, it will be displayed on his account info. There will be two types of account in our application. The first will be the student account and the second will be an admin account. The admin will have access to all the classes in the app without any restriction and will be able to see what students have requested, when and the student's information. One of the classes will be the statistics, where the admin will be able to see graphs with all the relevant information such as what are the times that are booked the most, by what group of students and the items that are requested the most. The admin will also be able to cancel a time reservation or item request from a student if necessary and update the student's info page with the reasons why time or item reservation had to be cancelled or denied. Most of the classes will have access to the database, since it needs its information to display it back to the user and it will support English and French. Matheus and Kenneth will be in charge of the application.

The UI is going to display a log in page as soon as it is opened. User will be able to then create his own account. Upon login, user will have a menu with the following options: My account where user will be able to find account info, such as requested items, schedule appointments in the lab, account information such as user name and etc. Request item, where he will be able to see and search for all the items and check if they are available, if so, he will be able to request them. Scan QR Codes, where the user will be able to scan a QR code present on the item and it is going to display extra information. About us, where the user will have a little bit of information about the development team of this application. Our main activity and focus of the application, Schedule, where the user will be able to select a date and time to book a time slot in the prototype lab and Log out, where user will Log out of his account.

3. Build Instructions

3.1 Hardware Production

3.1.1 System Diagram

The webcam will be used to scan ID barcodes and item QR codes to register the parts with students. The codes will then be processed by the raspberry pi. Once everything is processed the Pi will upload the info to the database which can then be accessed through a mobile app for a better visual experience.

Figure 1: systemdiagram

Image 1: System Diagram *Image 1: System Diagram shows how the different components work*

3.1.2 Build Budget

Before we start building something, we need to plan a budget. A budget will be especially helpful in managing the financial costs related to our build.

Here's a breakdown of the item needed to build this project.

1. A Raspberry Pi
2. USB Webcam
3. A MicroUSB AC adapter
4. A Case for Raspberry Pi
5. HDMI Cable
6. A MicroSD Card (8GB or higher)
7. Keyboard/Mouse Combo (USB or bluetooth)
8. A monitor

The list of PCB components needed to build the PCB

- 2x20 Header Female (Qty: 1) $3.73
- 100nF Capacitor (Qty: 4) $5.80
- 10uF Capacitor (Qty: 1) $1.45
- 1kΩ Resistor (Qty: 1) $0.15
- 3.9kΩ Resistor (Qty: 4) $2.40
- 4.7kΩ Resistor (Qty: 3) $1.35
- 3mm LED (Qty: 1) $0.61
- N-Channel Logic Level MOSFET (Qty: 1) $0.99
- Real Time clock Battery (Qty: 1) $0.48
- I2C Real Time Clock (Qty: 1) $11.95
- 8-bit I2C ADC&DAC with sensors (Qty: 1) $8.50
- 1x4 Header Female (Qty: 2) $1.70
- 1x5 Header Female (Qty: 2) $3.40

Let's start with the core of the project, a Raspberry Pi. The Raspberry Pi alone costs around $60-70 online if you're lucky enough find one. We highly recommend opting in for the Raspberry Pi starter kit instead as it comes with all the necessary items like a microSD card, an AC adapter and a case (as well as other stuff like a microSD reader). It's available for about $20-30 extra.

A case is highly recommended as the raspberry pi is relatively fragile microcomputer. A microSD card is required to run the Raspbian OS on the pi. And of course, the AC adapter can be substituted with any microUSB phone/tablet charger. Or you can follow this guide to construct your own case.

Moving on, a webcam will be required to scan QR codes. Any USB webcam will do fine. So no need a spend the extra bucks for a high definition webcam. An average webcam will cost between $20 and $35. Or even cheaper if you find a used/refurbished one.

Finally, there is cost associated with the fabricating of a PCB (Printed Circuit board) as well it's components. The Prototype Lab at Humber College charges about $40 to print a PCB. The components we have on my PCB cost about $42.52. However, we do a have a bunch of extra components like sensors and headers that you don't need. Therefore, we have them crossed out on the list. Moreover, this brings the cost down from $42.52 to $16.49. The actual cost might still be higher since not all components are sold in the quantity of single units. As Humber students, all of this is covered as part of our tuition fees.

Altogether we spent about $100 on the Raspberry Pi starter kit from Amazon. If you buy an older Raspberry Pi, you have to spend extra on a USB wifi receiver or use the ethernet instead. We used my older webcam which we bought for about $35-$40 at the time. And we didn't have to pay extra for the PCB since we're Humber students as well as the PCB components. We're not including a monitor, and a keyboard and a mouse since almost everybody owns at least one of each. If not then you can buy any monitor with an HDMI and any USB or Bluetooth keyboard/mouse combo.

Here's a table that shows all the items we bought and how much we paid for them. Note all prices are in CAD. We're also including the PCB fabricating cost to create a reference for you to compare total costs.

Item	Cost
Raspberry Pi 3 Starter Kit	$99.99
USB Webcam	$34.99
PCB Fabrication Cost	$40
Required components for PCB	$42.52
Raspberry Pi Case Construction	$33.89
Total	**$284.08**

Our final build cost totaled to about approximately $300 (That's including taxes and other expenses like shipping and handling).

3.1.3 Time Commitment

Time is a resource arguably as valuable as money. And this project demands plenty of it.

We had almost 15 weeks to work this project however with these detailed instructions we will be cutting that time significantly for anyone wanting to recreate this project. Majority of our time was spent in implementing the QR scanning functionality and blinking the L.E.D. for feedback. It was our first time soldering and we were able to solder the PCB in less than an hour.

If you already have acquired all the parts and got the PCB printed, we believe this project can be completed in a day. That's if you copy paste the all the code and it runs flawlessly without any errors.

Tasks	**Time Required**
Preparing Online Template	1 Weeks
Parts Ordered/Delivery	2 Weeks
Raspberry Pi Setup	1 day
PCB Soldering	2 hours
Testing PCB	30 mins
Research and Coding	5 hours in span of 4 days
Mechanical Assembly	15 minutes
Designing acrylic box	2 hours
Laser Cutting	15-30 minutes

Work Schedule: *Table2: Displays time required to for each task*

3.1.4 PCB Soldering

Time to start building the PCB! However, you build something, you must design it first. That's where these files come in handy. Go ahead and download these files.

Board File	Schematic File
HSHV4-student version.brd	HSHV4-student version.sch

Now that you have these files, you'll need eagle to open them. Eagle can be downloaded **-->** here. After installing Eagle, edit the name in the files to your own. After you finish editing, go ahead and follow the instructions **-->** here to generate the following files.

Top Copper (.GTL)	Bottom Copper (.GBL)	Top Soldermask (.GTS)	Bottom Soldermask (.GBS)	Top Silkscreen (.GTO)	Bottom Silkscreen (.GBO)	Drill File (.TXT)
HSHV4-student version.GTL	HSHV4-student version.GBL	HSHV4-student version.GTS	HSHV4-student version.GBS	HSHV4-student version.GTO	HSHV4-student version.GBO	HSHV4-student version.TXT

%20Log%20Files/Schematic.PNG" %20Log%20Files/Schematic.PNG.pdf" %20Log%20Files/Schematic.PNG.PDF" %20Log%20Files/Schematic.PNG.eps" %20Log%20Files/Schematic.PNG.EPS" %20Log%20Files/Schematic.PNG.mps" %20Log%20Files/Schematic.PNG.MPS" %20Log%20Files/Schematic.PNG.ps" %20Log%20Files/Schematic.PNG.PS" %20Log%20Files/Schematic.PNG.png" %20Log%20Files/Schematic.PNG.PNG" %20Log%20Files/Schematic.PNG.jpg" %20Log%20Files/Schematic.PNG.JPG" %20Log%20Files/Schematic.PNG.jpeg" %20Log%20Files/Schematic.PNG.JPEG" %20Log%20Files/Schematic.PNG.jp2" %20Log%20Files/Schematic.PNG.JP2" %20Log%20Files/Schematic.PNG.jpf" %20Log%20Files/Schematic.PNG.JPF" %20Log%20Files/Schematic.PNG.bmp" %20Log%20Files/Schematic.PNG.BMP" %20Log%20Files/Schematic.PNG.pict" %20Log%20Files/Schematic.PNG.PICT" %20Log%20Files/Schematic.PNG.psd" %20Log%20Files/Schematic.PNG.PSD" %20Log%20Files/Schematic.PNG.mac" %20Log%20Files/Schematic.PNG.MAC" %20Log%20Files/Schematic.PNG.TGA" %20Log%20Files/Schematic.PNG.tga" %20Log%20Files/Schematic.PNG.gif" %20Log%20Files/Schematic.PNG.GIF" %20Log%20Files/Schematic.PNG.tif" %20Log%20Files/Schematic.PNG.TIF" %20Log%20Files/Schematic.PNG.tiff" %20Log%20Files/Schematic.PNG.TIFF" %20Log%20Files/Schematic.PNG" %20Log%20Files/Schematic.PNG" %20Log%20Files/Schematic.bb" %20Log%20Files/Schematic.PNG"

Figure 2: Schematic

Image 2: Schematic for the PCB *Image 2: Technical Schematic of the PCB*

These files are used to instruct the machine with the coordinates and to engrave the PCB. It takes about 4-5 hours for the machine to print it. After it's done, wash with water *(preferably deionized water)* and wipe it a piece of cloth to avoid leaving water marks (I recommend using a micro-fiber cloth) and wait for it to dry completely. After that's done you can sand the PCB a bit just to make it shinier. If you do sand it you have the repeat the washing process again.

Now the PCB is ready to be soldered. However, safety comes first. Make sure to wear safety glasses and dress appropriately for a lab/workplace environment. Now Use the schematic as the guide and solder each of the components to their respective places and making sure that the excess solder is not causing a short. Now inspect each soldering carefully again use a voltmeter to test for continuity. If all goes well you are ready to move on to the next chapter.

%20Log%20Files/PCB.png" %20Log%20Files/PCB.png.pdf" %20Log%20Files/PCB.png.PDF" %20Log%20Files/PCB.png.eps" %20Log%20Files/PCB.png.EPS" %20Log%20Files/PCB.png.mps" %20Log%20Files/PCB.png.MPS" %20Log%20Files/PCB.png.ps" %20Log%20Files/PCB.png.PS" %20Log%20Files/PCB.png.png" %20Log%20Files/PCB.png.PNG" %20Log%20Files/PCB.png.jpg" %20Log%20Files/PCB.png.JPG" %20Log%20Files/PCB.png.jpeg" %20Log%20Files/PCB.png.JPEG" %20Log%20Files/PCB.png.jp2" %20Log%20Files/PCB.png.JP2" %20Log%20Files/PCB.png.jpf" %20Log%20Files/PCB.png.JPF" %20Log%20Files/PCB.png.bmp" %20Log%20Files/PCB.png.BMP" %20Log%20Files/PCB.png.pict" %20Log%20Files/PCB.png.PICT" %20Log%20Files/PCB.png.psd" %20Log%20Files/PCB.png.PSD" %20Log%20Files/PCB.png.mac" %20Log%20Files/PCB.png.MAC" %20Log%20Files/PCB.png.TGA" %20Log%20Files/PCB.png.tga" %20Log%20Files/PCB.png.gif" %20Log%20Files/PCB.png.GIF" %20Log%20Files/PCB.png.tif" %20Log%20Files/PCB.png.TIF" %20Log%20Files/PCB.png.tiff" %20Log%20Files/PCB.png.TIFF" %20Log%20Files/PCB.png" %20Log%20Files/PCB.png" %20Log%20Files/PCB.bb" %20Log%20Files/PCB.png"

Figure 3: PCB

Image 3: Completed PCB

3.1.5 Mechanical Assembly & Power Up

By now, you have acquired all the parts you are ready to start doing mechanical assembly. It's extremely easy from here onwards since this guide a shortchut to the finishline.

You should now have the following: a Raspberry Pi, a USB Webcam, a MicroUSB power source and A MicroSD Card (8GB or higher). The starter kit's MicroSD card comes with pre-loaded Raspbian.iso so all you have to do is install it. If it does not then it can be downloaded from the Raspberry Pi website and copied on the MicroSD via a card reader (also included in the starter kit). Insert the flash card into Raspberry Pi.

Begin by connecting power, HDMI and a keyboard and mouse and the USB Webcam to the Raspberry Pi. Also connect your PCB to the Raspberry Pi. After it boots to the desktop, use **sudo apt-get update** on the command line terminal to update the Raspbian to the latest version. Then use **sudo apt-get dist-upgrade** to upgrade all the installed packages. These commands require internet (ethernet or wifi) to download the updates.

That's it for the assembly.

3.1.6 Unit Testing

Webcam functionality testing For testing, start by installing the fswebcam library by typing **sudo apt-get install fswebcam**. Then type fswebcam followed by a filename with .jpg extension. For example, **fswebcam image.jpg**. This will snap a picture and save it the current directory with the filename image.jpg. If this works, that means the USB Webcam functions properly.

PCB functionality testing To test the PCB, run the traffic2B.c (source code available **-->** here) with the following steps.

1. Type **sudo apt-get install wiringPi** on the terminal to the Pi Wiring Library
2. Compile the file using **gcc -Wall -o traffic2B traffic2B.c -lwiringPi** (this creates an execuatalbe object called traffic2B)
3. Run the file using **sudo ./traffic2B**

This should start blinking the L.E.D's in a traffic light pattern for an endless loop. Use **Ctrl+C** to exit the program. If this works, this means the PCB is ready to be used.

3.1.7 Scanning QR Codes

You can generate your own QR Codes ---> here, or any other website of your choosing. If you are using a service that allows you to set the pixel count of the QR code, we recommend setting it to the minimum as it's faster and can be handled by webcams with lower resolution.

Now we install the zbar library to scan QR codes. Follow these steps below:

1. Type **sudo apt-get install zbar-tools** on the terminal
2. Use **ls /dev/video*/** to find your video source
3. Test by scanning a QR code by typing **zbarcam /dev/video#** on the terminal. (Note: the pound sign "#" is not literal value, it's supposed to represent the number that step 2 returns returns).

Your webcam should now be working so go ahead and scan a QR code. You should now see the value of QR Code on the terminal saying preceded by the string "QR Code:". You can scan as many QR codes as you'd like. Use **Ctrl+C** to exit the program.

The QR Code scanner has been implemented.

Nope. You're still not done yet. You need to add the capability to blink the L.E.D as well as save the results into a file for external use (ex: storing information into a database). Keep reading to find out how you add these.

Shell Script	Python Blink	greenLight.c
scanqr.sh	alertLight.py	greenLight.c

Run the following code to Scan QR Codes to implement the QR code scanner The python code is pretty simple it turns the the L.E.D on for 1/8th of a second and then turns it off for the same duration. This cycle is repeated 11 times to show the L.E.D feedback.

To change the 1/8th second on/off duration simply change the **time.sleep(0.125)** value to any number you like in the blink function. Lower values results in short durations and conversely higher values result in longer durations.

```
def blink(pin):
        GPIO.output(pin,GPIO.HIGH)
        time.sleep(0.125)
        GPIO.output(pin,GPIO.LOW)
        time.sleep(0.125)
        return
```

Moreover, you can also change the number of cycles for how many times this led on/off will be repeated. Simply change the loop count by changing 11 to any value you desire.

```
for i in range(0,11):
```

When it's all done you test run this program by typing **python alert.py** on the terminal. The L.E.D. should display according to your modifications.

The C file *greenLight.c* is just there to initiate the green light for use during the scanning process. It just blinks the led in green for 1/10th of the second. I recommend leaving it as is.

The Shell Script file *scanqr.sh* is where the bulk of the program lies. It starts off by compiling and executing *greenLight.c*. Then it defines a few variables for storing information. Every session of script execution produces a file called "scan" followed by a timestamp and the .txt extension. You can change the output filename by editing the variable $ScanResult.$“‘$bashNameofscanresultsfileScanResult =$ ”cwd/scan_$DATE.txt” “‘

The function *scan()* initiates zbarcam to scan QR codes and then kills the task after saving the output. It even asks the user to scan multiple entries. Since it’s using zbarcam you apply modifications by using flags like *--prescale* to set the resolution of the scanning resolution.

```
function scan() {
  zbarcam --raw --prescale=320x240 /dev/video0 > $tmp &
  ...
  # Kill tasks, free up space and call test.py to blink L.E.D.
  # Append scan results to file
  # Show scan results
  # Prompt to continue scanning
}
```

More information about the flags and their usages can be found --> here.

To run the final code follow these steps:

1. Make sure all three core files: *greenLight.c*, *alertLight.py* and *scanqr.sh* are in the same directory.
2. On the terminal, use **chmod** 777 followed by the filenames and grant all Read-Write-Execute permissions to the core files.
3. Type **./scanqr.sh** to execute the shell script.

You should now have the program working. The program scans QR code and then blinks the L.E.D. as well showing the scanned item on the screen. After scanning an item it prompts to scan another. If you choose yes, it scans another item and prompts to scan more items. If you choose no, then the program exits and shows the list of items scanned during the session. According to our code, every session generates a new file with a timestamp. You can use one file and reuse if you want. Furthermore, you can press **Ctrl+C** anytime to exit the program.

Congratulations! You’ve just implemented the full QR Code scanner program. We hope you were able to reproduce this project following these instructions.

3.1.8 Case Construction

This section is completely optional, if you don’t intend to package the webcam and raspberry pi together, feel free to omit this section.

We started with a few rough sketches of the overall outline of the case, then later decided on building a sliding top. It’s a fairly simple design, yet it is a lot more interesting than slotted laser-cut acrylic cases. The slot for the ports is chamfered inwards for convenience. The slider top cover holds the webcam, which will be pointing to a focal point where a QR code can be placed for quicker scanning.

Our design process consisted of brainstorming/outlining a rough sketch, followed by CAD model in AutoDesk Inventor, followed by the construction full scale foam model.

It is imperative that less extra space while measuring, marking, or cutting on the foam. This is because if you make a mistake (for example accidently chipping off an edge) you can sand it down and still have a perfectly usable piece.

Here is what our final case looked like. The cream/greyish ones are the AutoDesk renditions of it while the brown snapshots are the pictures of the full-scale foam model.

Image 4: A collection of pictures showcasing the CAD model (cream) and the real-life model (brown) of the case from various angles

Figure 4: Case

3.2 Application Development

Since the application is the backbone of our project, we put a lot of effort into making it as clean and user friendly as possible. First we all sat together and started brainstorming how we wanted the UI to look like with drawings and from there we started developing our application. By having a general idea of how we wanted the UI to look, we could start programming the buttons and apply functionality to the activities. We have in total 8 activities in our application and they are: AboutUsActivity where it gives information about the developing team of the Prototype lab flow. The MainActivity is the activity presented to the user as soon as the application is opened. The MySingleton activity where all the requests are queued. The ProfileActivity that shows the user their account info and also items and time slots booked. The RegisterAcitivity that handles user registration and allows him to create his account. The RequestItemActivity where the user will be displayed with the items information and amount and then be able to request it. The ScanQRActivity where the user will have to allow the prompt to give permission to the application to utilize his camera and then he will be able to scan QR codes. And our fine activity and one of the most important, the ScheduleActivity where the user will be able to request time slots and book them through it. Because we have been taught to program applications through android studio, and because it gives us more liberties and is easier to test on android phones, the application will only be available to android devices for now. In the beginning we were mainly testing it through the android studio emulators since we did not have an android device. But we have been utilizing Sukhdeep's phone lately which can give us a better way of testing it, since it is a real device and we can actually have a better notion of how it looks and how the usability of it works. So far we have been pleased if our application and as developing continues, we will improve it each time more and more. This iteration of the application is so far bug free but it has not yet been put out for mass testing where we expect users to find some bugs since it has not yet been widely tested by many different users. With that in mind, as bugs appear we plan to solve then out and release updates to the application with the fixes.

3.2.1 Application and Database testing

Test case:001 Test Scenario: Login test with good credentials

Purpose: Test will output the results of entering correct login credentials

Step Details: Start application and enter ”testuser” as the username, then enter ”123” for the password and press the login button.

Expected Results: Accepts the user credentials and logs in with the test account.

Actual Results: As expected

Test case:002 Test Scenario:Login test with bad credentials

Purpose: Test will output the results of entering incorrect login credentials

Preconditions: Clean installation of the application, Wi-Fi and data turned on.

Step Details: Start application and enter ”testcase002” for username and ”123” for password and press login.

Expected Results:The application will return a dialog box to the user, informing the user that the credentials are not valid and page data will flush

Actual Results: As expected

Test case:003 Test Scenario: Checking in the remember me box, so that it remembers user and password info

Purpose: This test will see if the checkbox remembers the user information when checked

Preconditions: Clean installation of the application, Wi-Fi and data turned on.

Step Details: Start application, enter the test credentials, check the remember me box and log in. Re-open application and check that the credentials were saved.

Expected Results:User credentials should be remembered already and ready to user to simply press log in without having to re-enter credentials

Actual Results: As expected

Test Case:004 Test Scenario: User does not check the "Remember me" box upon entering his credentials

Purpose: To test if without the checkbox "Remember me" checked, application will still save the credentials

Preconditions: Clean installation of the application, Wi-Fi and data turned on.

Step Details: Start application, enter test credentials and log in without checking the remember me box. Re-open application and credentials should not have been saved.

Expected Results: Credentials should not be saved since checkbox was not checked

Actual Results: As expected

Test Case:005 Test Scenario: Scanning QR code with "deny" on camera permission for scanner

Purpose: To test if the application gives an error message since the camera does not have permissions to work with the app

Preconditions: Clean installation of the application, Wi-Fi and data turned on.

Step Details: Start Application, log in with test account and on the main menu, select scan QR code

Expected Results:Gives an error message back since it does not have permission to use the camera from the cell phone.

Actual Results: As expected

Test Case:006 Test Scenario: Scanning QR code with permission to use camera

Purpose: Test if the app is able to utilize the phone's camera

Preconditions: Clean installation of the application, Wi-Fi and data turned on.

Step Details: Start application, log in with test account and select Scan QR codes from the menu.

Expected Results: Opens the camera and prompts user to display QR code to the camera

Actual Results: As expected

Test Case:007 Test Scenario: User confirmation to exit the app

Purpose: To test if upon pressing the exit button, the app should prompt the user if he really wants to exit it.

Preconditions: Clean installation of the application, Wi-Fi and data turned on.

Step Details: Start application and click on the exit button and then click on "yes" on the prompt window

Expected Results: Should exit the application without any issues

Actual Results: As expected

Test Case:008 Test Scenario: Clicking on exit and on “NO” on the prompt window

Purpose: To check if the program cancels the exit once NO is clicked

Preconditions: Clean installation of the application, Wi-Fi and data turned on.

Step Details: Start application, click on the exit buttom and then click ”No” on the prompt window

Expected Results: Should cancel the exit process and not exit the program.

Actual Results: As expected

Test Case:009 Test Scenario: Scheduling date and time

Purpose: To check if the scheduling tab is working as intended.

Preconditions: Clean installation of the application, Wi-Fi and data turned on.

Step Details: Start application, log in with test account and click on ”Schedule”. Select date March 5, 2017 and select time 4pm. Click on the schedule button.

Expected Results: A date and a time were selected therefore app schedules it and it gets stored into the database.If a date or a time are not selected the application will not schedule it or send any of the information to the database.

Actual Results: As expected

Test Case:010 Test Scenario: Check functionality of request item

Purpose: To check if the request item tab and buttons are working as intended.

Preconditions: Clean installation of the application, Wi-Fi and data turned on.

Step Details: Start application, log in with test account. Once logged in click on request item and then select the item and the amount required and click on the ”resquest item” button.

Expected Results: Request Item tab should open allowing the user to select the amount and what item user wants, the item then is requested and assigned to that user in the database

Actual Results: As expected

Test Case:011 Test Scenario: User clicks on “My account” on menu

Purpose: To check if it is displaying proper user info

Preconditions: Clean installation of the application, Wi-Fi and data turned on.

Step Details: Start application, logs in with test account or an user created account and click on ”my account” on the menu.

Expected Results: Display student complete name, username and student number and also time slots or items requested.

Actual Results: As expected

Test Case:012 Test Scenario: User requests item and sets a date

Purpose: To check if the item and date info are being added to the right student in the database

Step Details: Start application, log in with test account or created account and click on ”Request Items”, request any items and any amount of them and then return to the main menu. Once in the main menu click on ”Schedule” and select a date and time.

Expected Results:User will be able to request the items and any amount of them required and then he will be able to request time slots and all the information should be stored properly in the database.

Actual Results: As Expected

Test Case:013 Test Scenario: User tries to scan a barcode with the QR scanner

Purpose: To test that it does not scan the bar code since it is set to scan only the QR

Preconditions: Clean installation of the application, Wi-Fi and data turned on.

Step Details: Start application, log in with test account or created account and on the main menu click on ”scan QR code”. Once the camera opens, try to scan a bar code instead of a QR code.

Expected Results: Application is going to open the camera and try to scan the QR code but it is not going to scan anything since the displayed code is a barcode

Actual Results: As expected

Test Case:014 Test Scenario: User tries to create account with user name that already exists

Purpose: Make sure user can not create an account with an account name that has already been used

Preconditions: Clean installation of the application, Wi-Fi and data turned on.

Step Details: Start application, click on ”Sign Up” and enter ”testuser” as username and enter any password.

Expected Results: App gives an error window informing user that user name is already taken

Actual Results: As expected

Test Case:015 Test Scenario: User leaves a blank field once creating or entering user name or password

Purpose: To test the prompt message telling the user to enter appropriate field

Preconditions: Clean installation of the application, Wi-Fi and data turned on.

Step Details: Start application and on log in leave either username or password blank

Expected Results: Application will prompt the user to enter the right information in the appropriate fields.

Actual Results: As expected

3.3 Administrator Website Development

3.4 Status Reports

TASK	DUE DATE
Writing Proposal	SEP 13, 2016
Project Schedule	SEP 20, 2016
Creating Budget	SEP 27, 2016
Acquiring Components & Progress Report	OCT 4, 2016
Mechanical Assembly & Progress Report	OCT 11, 2016
PCB Fabrication	OCT 18, 2016
Interface Wiring & Placard Design	OCT 18, 2016
Preparing for Demonstration	OCT 25, 2016
Demonstration & Progress Report	NOV 1, 2016
Build Video	NOV 8, 2016
Incorporating feedback & Progress Report	NOV 15, 2016
Practice Presentations	NOV 22, 2016
Individual Presentations	DEC 6, 2016
Project Video	DEC 13, 2016
Proposal resubmission	JAN 20, 2017
Requirements Specifications	JAN 27, 2017

Status Report Email I	FEB 3, 2017
Technical Report Structure and Mechanics	FEB 10, 2017
Status Report Email II	FEB 24, 2017
Build Instructions	FEB 3, 2017
Integretion Status Report Email	MAR 10, 2017
Technical Report Checklist	MAR 17, 2017
Status Report Email III	MAR 24, 2017
Technical Report	Mar 31, 2017

Status Update - Jan 20, 2017 So far, for our hardware we have a webcam connected to Raspberry Pi that can scan QR codes and save them into a file in CSV format. We still have to find a way to add the contents of the CSV file to our database. Our web server is hosted on Hostinger, where we are currently able to store registered accounts, list of items borrowed by each student, and students' preferred usage hours at the prototype lab.

Our software application, as of right now it consists of the following activities, a login page, a registration page, an account overview page, an item sign out activity, a time sign up activity, and a scan QR activity. Our next task is to add the list of days available for signing in as well as the list of available slots on each day. As well as a limit to number of times people can sign on for.

For financials, we've spent almost $200 on the project and it's projected a rise bit more as we will be designing a better case. We plan our progress through meetings we partake every week.

Independent Progress - Feb 17, 2017 We have so far begun implementing the different aspects illustrated in our technical report to our webpage and android application. So far our team has been very successful in finishing up all the tasks in time with our schedule. At the moment we have yet to implement some changes in our hardware but since our project is so heavily based on software we are not worried about the hardware and are sure it will be ready with the changes implemented in time with our schedule.

So far Kenneth has developed the database for our application, which is going to be implemented into our webpage and application. We have decided to use a better solution to our database server called Firebase by Google. So far Kenneth's progress has been on schedule.

Since our last update on the hardware, Sukhdeep has been looking into better ways of implementing the hardware into our webpage application but we have only come up with the design and function and have yet to implement these changes, since last year we had completed most of the hardware build, implementing these changes involves just adding simple code to the hardware but without a proper functioning webpage we decided to complete that after the webpage was fully developed.

Matheus has been working on the overall functionality of our Android application, since Kenneth is working on the database; Matheus is working on the flow of the Android app and testing the application in offline mode with dummy code to simulate a server connection, we are still also testing with our Hostinger we server but plan to change over to Firebase once it has been fully developed.

As for me, Gurpreet, I have designed a website login page that is able to connect to Firebase and allow you to login using an email and password. At the moment you can also upload a small message to the server as an example to show in today's class that we have the ability to upload data to the server.

As of now our team is on schedule, we have not faced any financial changes at the moment but may have to once we begin on changing the hardware casing but so far we are beginning to see our vision come together slowly and progressively. For the week following the reading week we should have Firebase connected to both the webpage and Android app with most of the basic functionality operating, things such as statistics and extra functionality will be added once the core functions are working flawlessly. We meet up together weekly to discuss our progress so far and also stay in touch via Whatsapp.

Integration Status Report - March 10, 2017 This past week we have been working in the integration of the database and the application using hostinger for our database. We have managed to integrate both but now we need to figure out how to make the time restrictions required for the prototype lab time slots, such as the time available and the group that the student belongs to, and use it on the application. This way the user, instead of being able to pick any day and any time, restrictions will be applied to the application allowing the user only to pick the time slots available for their group and only when the prototype lab is open. The website was done using firebase as the database, but we are now going to change it to hostinger as well so it can be integrated to the application. The hardware so far, is where we are having most difficulties. We are currently generating a file with the QR code and now we are working on a way of connecting the QR information stored on the file to our database, therefore once the student scans his QR code his info is going to be stored in the database for the corresponding time slot requested.

Troubleshooting Progress - March 24, 2017 In our troubleshooting report we are going to say a few of the major problems we were having with our different components and how we solved it. For our application and database, we were having issues at first at our registration option, where the student could register but the info wasn't getting stored properly in the database, therefore as soon as he signed out his account would be gone and he would have to create it all over again. And for testing we hard coded user accounts, items and time slots to the database to make sure it would all be displayed properly back to the user and if it would be applied also to the proper user in the database. For our hardware, at first when building the case, we were using too many layers of foam, which was not really being practical and it was not looking good either therefore we decided to use less foam to make it cleaner. Sukhdeep also tried in his first attempt at building the case, to keep it together with hot glue but it was melting the foam therefore he built it using wood glue. The case ended up adding $30 dollars to the budget to get the foam, glue and the other parts required to build it.

4. Conclusion

The prototype lab flow project is going to automate the school's prototype lab system, making it easier and more appealing to students and staff. With the application allowing students to book time slots and items available in the prototype lab, there will be a higher usage of the lab and therefore more opportunities to students to develop their ideas. The hardware assists on the automation by scanning the students QR code generated by the application and storing it to our database, where the student info will be displayed with the time that the student check in or out. By doing so, we hope to bring a system more in line with the program we are studying and also a system that is much more appealing to students and staff to utilize the prototype lab more often and therefore stimulate all the students ideas by imposing less barriers to them.

5. Recommendations

Having a higher resolution is not necessary,but it makes the scanning process the faster. We recommend purchasing this webcam.

6. Bibliography

Lau, P., Papanikolopoulos, N. P., & Boley, D. L. (1993). Gabor-qR decomposition for image encoding. *Electronics Letters*, *29*(25), 2182–2183. https://doi.org/10.1049/el:19931465

Lezhebokov, A. A., Kravchenko, Y. A., & Bova, V. V. (2014). Support system for qR-code-based educational processes. In *2014 iEEE 8th international conference on application of information and communication technologies (aICT)* (pp. 1–4). https://doi.org/10.1109/ICAICT.2014.7036011

Ramya, M., & Sheela, M. J. (2014). VLSI implementation of hybrid qR code generation system. In *2014 international conference on electronics and communication systems (iCECS)* (pp. 1–6). https://doi.org/10.1109/ECS.2014.6892829