**Name: Ifeoluwa David Adese**

**Student ID: N01063518**

**Course: Hardware Production Technology (CENG 317)**

**Project: Parts Crib Database + USB Barcode Scanner (Build Instructions).**

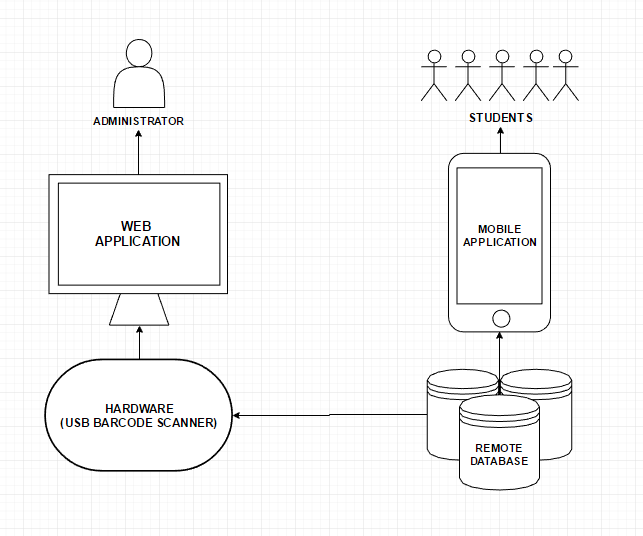
**Correct web template usage:** At the start of the semester, I set up about 2 repositories for this project, one for the software and the other for the hardware, as well as a website to keep track of our weekly progress on the hardware development aspect. The template for this website/blog was provided by the instructor. The web page files were embedded in the repository and updated via html scripts. After every web page update, I published and committed my local changes to the remote repository via the GitHub desktop app in order to view my changes live on the website/blog.

Webpage – [***ifeoluwadavid.github.io/PartsCribDatabase***](ifeoluwadavid.github.io/PartsCribDatabase)

**Introduction using a system diagram:** Although, this course specifically focuses on the hardware production aspect of this project,there are generally 3 major areas i.e. the hardware setup i.e. the USB barcode scanner & Raspberry Pi, the web application development (partscribdatabase.tech) and the mobile application development (Android). In terms of design, the simplest way I can explain is this:

* Web Application: Designed for Both Admins & Students but mainly used by Administrators.
* Mobile Application: Designed for both Admins & Students but mainly used by Students.
* Hardware Product (Barcode Scanner): Designed strictly for use by the Parts Crib Administrators.

Only the web application communicates with the hardware, although not compulsorily. It simply scans in registered barcode IDs into the web application’s text field and triggers an exchange of data between the server and the web application within a split second. However, the mobile application is developed to function on its own, independently. The diagram below illustrates how the project’s data flow works generally.



**Bill of Materials/Budget**: In terms of the budget for the required materials, I spent slightly over $160. The raspberry Pi 3 unit from Canakit (via Amazon) cost me exactly $74.95 and the USB barcode scanner module cost about $87 from Adafruit. Initially, from the start of the project proposal, this was already the proposed cost, so neither did I spend over nor below budget.

**Time Commitment:** Normally, this exact project or something much similar can be finished in about a week or less if full attention is paid to it. However, I wasn’t able to get things fully up and running within that time frame, because I also had to spend time on other project requirements like the software development side of things. I also felt the need to be in track with the weekly expectations and not try to run ahead of the class deliverables like I did on the software development side. I felt it was going be highly risky if anything happened to my hardware because of that.

Additionally,for the hardware section of this project, where all three group members were expected to work individually on their sensors and effectors. I realised a waterfall work flow would be the most ideal option for working individually, because I felt the need to work things out one after the other i.e. one task depending on the previous task. This way, I was able to measure my progress and what I’ve accomplished, on a weekly basis. However, I sort of deviated from the proposed project schedule after the school strike, which made things slightly disorganised, causing me to spend a lot of hours trying to cover as much work as I could on a daily basis. But after, getting familiar with the mechanical assembly, and building a portion of web application to respond to our hardware. I also figured out I didn’t need any kind of programmatical configuration (e.g. a python script) to get my USB barcode scanner to do what it was expected to, on the web application. At this point, things seemed a bit fast tracked and almost complete.

**PCB/Soldering:** During the week of the PCB/Soldering deliverable, our instructor provided us with a schematic which was expected to be edited and 3D printed. We were also given a detailed step by step procedure as well as some materials for our first soldering project. Upon completion, we tested for shorts and open circuits in our finished product before mounting the board on the Pi. We were then asked to remotely access some repository files via the Pi’s command line. These files were test programs that we had to edit, run and observe its effects (i.e. temperature reading, light etc.) on the mounted PCB board’s LED. I believe the main goal for this project was just to get us to familiarize with the raspberry Pi’s OS interface and the command line.

**Mechanical Assembly:** The setup procedure for my personal project i.e. the Parts Crib Database + USB Barcode Scanner, can actually be done in different ways depending on the resources available. Initially, I tried finding my Pi’s IP address (which was not yet static) for remote access control via VNC viewer, because that way, I could use minimal resources i.e. just the raspberry Pi, the USB barcode scanner and a PC/Laptop. However, this never really worked out for me, so I settled for set up using more materials like a VGA to HDMI cable, as well as an external mouse and keyboard. This helped me avoid the process of having to access my raspberry pi remotely, instead directly.

In my set up process, the VGA to HDMI cable connects the Pi (HDMI) to the PC (VGA), and the USB ports for both the external keyboard and mouse connects to 2 out of 4 USB ports on the Pi. The USB Barcode scanner also simply connects to one of the 2 other vacant USB ports on the Pi.

**Power Up**: After the initial setup procedure given above, the computer can then be powered on first, before the Pi’s power adapter is plugged in to power and load up the Pi. The home screen to the Pi’s operating system will be shown after switching configurations to the computer’s VGA settings. Usually, at this point, the Pi is already connected to the surrounding Wi-Fi, which I configured during the OS installation process. In the absence of Wi-Fi, a USB-Ethernet adapter can come in handy for network sharing on whichever computer the Pi is connected to, provided the ethernet sharing permission is enabled.

**Unit Testing:** For unit testing, I assume that was done during the week of the PCB Board/Soldering deliverable. Because it was simply a means of checking how well my raspberry Pi unit responded to external connections like the PCB board, before I proceeded to using the raspberry Pi with my any other project required materials. That way if I encountered any issues, I could be very sure it was not as a result of a faulty Raspberry Pi.

**Production Testing:** This was also a successful process and I can say that because after the complete mechanical assembly i.e. with the USB barcode scanner, the Pi unit, a PC/Laptop as well as the web application, everything functioned properly as expected. The barcode scanner was able to recognise registered barcode IDs and read them into the web application’s text field for exchange of data i.e. Barcode ID for complete user information.