GNG1103

Technical Report

Prototype III and Customer Feedback

Submitted by

Project Group 19

Senthura Yogarajan, 300077498

Dayu Liu, 300063931

Tom Lin, 300076426

Kian Mozafarian, 300138481

Alexander Steeves, 300085657

Date: November 24, 20199

University of Ottawa

Abstract

This project is being executed because we want to solve problems regarding the automation of STEM. We have empathized with our clients by becoming users in their space and listening to their needs. We then defined a problem statement in which we highlighted the main problems. We have ideated to assist in brainstorming possible designs and solutions. We created an initial prototype in which it deemed successful based on our criteria to evaluate it. Our second prototype was successful as well. Our final prototype incorporated elements from both the first and second prototype and is successful in its applications.

Table of Contents

List of Tables

List of Figures

1 Introduction

Group 19 seeks to create more automacy within Maker Space. Currently the process of making a 3D print takes an excessive amount of time and resources. A major issue is the loss of SD cards through users misplacing and losing them, which costs Makerspace money to re-purchase more SD cards. We seek to make a single hub using Ross Video Dashboard and OctoPrint to remove SD cards entirely and reduce confusions in using 3D printers and to minimize the wasted time. In this deliverable we will discuss the creation of our final prototype and the obstacles we have encountered whilst creating it.

1.1 Problem Statement

We want to design a system and user interface that improves the current problems with automating the use of 3D printers while eliminating the troubles with SD cards to enhance the user and staff experience with 3D printing, organizing, time management, and ease of use.

2 Design for Prototype III

The prototype consists of using one raspberry pi, four 3D printers, one computer, one power source, four printer cables, python source code, and multiple instances of OctoPrint. Each instance will connect to the same raspberry pi. The g-code is then uploaded from OctoPrint to the desired printer from the web browser.

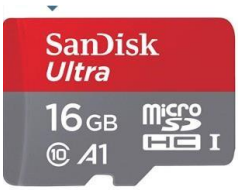
Our group is making a UI that indicates all key features and information regarding a print job. The UI will display the various states of the printer such as Operational, Printing, Cancelling, Printer is OFF, etc, while color coordinating the UI for each state. The UI will also display key details such as the time left for the print to be completed, the time spent printing, the percentage of progress made on the print job, the file name, and the filament left. Not only that, at the bottom of the UI, the time left until Makerspace closes is also displayed to indicate to the user if their print will be completed by the time Makerspace closes. The data for each print job is acquired using HTTP GET Requests which is made on a separate python code. The GET Requests save the response files locally on the computer. Dashboard reads and parses through these files while the requests are being made. Thus, allowing the UI to update with live values for time, percentage, states, etc. simultaneously as the GET Requests are being made. Another feature on the UI includes a button that will change color to indicate if there is a completed print or cancelled print left on the print bed. The button indicates to the user that the bed isn't free. When the completed print or the cancelled print is removed from the bed, the button is clicked and the state changes to indicate to the user that the print bed is free and that that printer can now be used. We have also benchmarked a filament plugin that already exists. This allowed us to utilize the sensor in the print head and

nozzle to get very accurate results. We were then able to make a separate GET Request to the Filament Manager Plugin so Dashboard can read the filament values as well.

We analyzed UI colors this deliverable. After researching into different types of UI color pallets such as monochromatic and analogous, we found a color pallet that fits our needs. We changed the colors on our UI to make it easier for the user to navigate the UI. Also, we've got rid of labels that don't serve a purpose in our UI so the user's vision isn't cluttered with unnecessary details.

3 Bill of Materials

Table 1 Bill of Materials

Part Name	Picture of Template	Description/ Specification	Quantity	Supplier	Cost (CAD)
Raspberry Pi		N/A	1	https://www.amazon.ca/RS-ComponentsRaspberry-Model-Motherboard/dp/B07BFH96M3/ref=sr_1_4?crid=J5VNE19VQ5WU&keywords=raspberry+pi+3b%2B&qid=1571017689&sprefix=raspberry+%2Caps%2C182&sr=8-4	$55.97
Micro SD		16 GB	1	https://www.amazon.ca/Sandisk-UltraMicro-UHS-IAdapter/dp/B073K14CVB/ref=sr_1_3?keywords=SanDisk+16GB&qid=1571260591&sr=8-3	$9.30
Printer Cable		6ft/1.8m	1	https://www.amazon.ca/gp/product/B00NH11KIK/ref=ox_sc_mini_detail?ie=UTF8&psc=1&smid=A3DWYIK6Y9EEQB	$4.43
Printer Cable		6ft/1.8m	3	https://www.amazon.ca/gp/product/B00NH11KIK/ref=ox_sc_mini_detail?ie=UTF8&psc=1&smid=A3DWYIK6Y9EEQB	$4.60

3D Printed Case + Stand		ABS/PLA	1	https://www.thingiverse.com/thing:3719217	$0
Wall plug		3 Amp/5Volt	1	N/A	$13.99
TOTAL COST(GST=3.59)					**$101.08**

4 Project Plan and Execution

The prototype will be tested in MakerSpace. This prototype is a physical and comprehensive prototype. Our prototype implements all attributes and components (e.g. raspberry pi, OctoPrint, 3D printer) and combines them to create one whole system. Our prototype is also a physical design allowing us to analyze unmodelled behavior (e.g. printer not extruding filament when print is started).

This prototype will analyze the accuracy of the Filament Manager Plugin. This will be tested by cancelling prints midway and letting prints complete to check the decrease in the total filament.

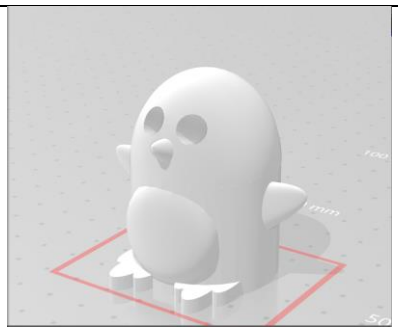
This prototype will also look at comparing Ultimaker Cura and Slic3r software to see which one produces the better g-code output in order to maximize print quality. This will be tested by exporting the same STL file through Cura and Slic3r and printing them using OctoPrint. The final printed products will then be compared to analyze print quality and determine which slicing software produces better results.

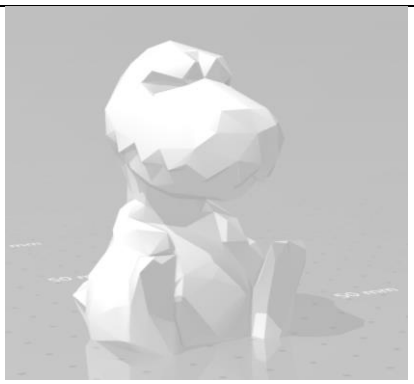
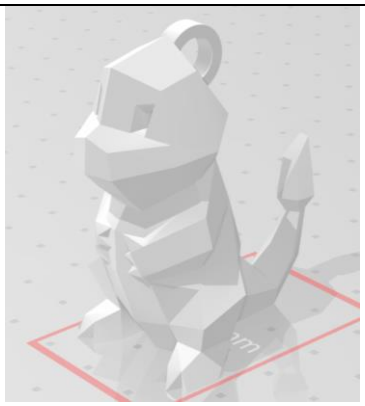
5 Prototyping and Testing and Customer Feedback

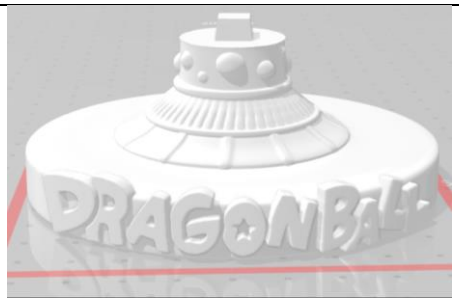
Some test cases were carried out where multiple prints were executed with different conditions and each test case has been labeled with its corresponding condition(s).

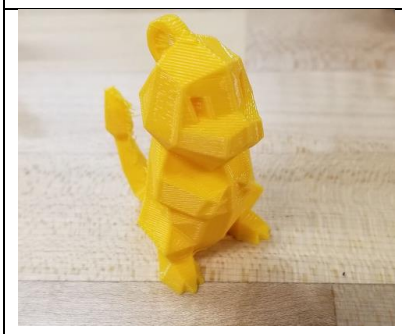
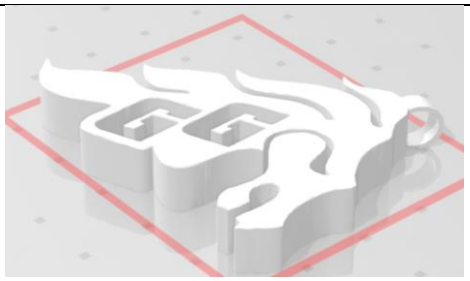
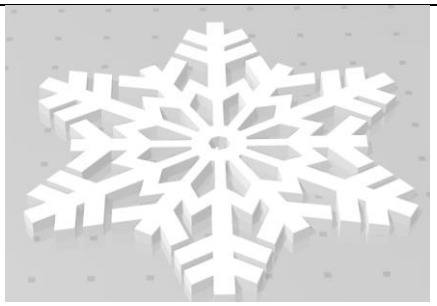
Table 2: FMEA

(*Short print = Time < 20 min, Long print = Time > 20 min)

3D Print	3D Model	Test case description
		A short print with appropriate quality. No special cases done. Complete print with a successive test run. The print was sliced with slic3r.
		A short print with initial good print quality. The Pi was disconnected and reconnected to the printer to see how the printer would react and the printer behaved as if it was restarted, meaning it assumed no file existed to read and print. This STL file was sliced using CURA

		A finished short print of a test case. No abnormalities in this test case. The test case came out as a poor quality. A poor test result. This test case was sliced with CURA
		A long print with a somewhat good quality outcome. Majority of the print executed properly and made pristine quality print until right before the end. The head of the print became a poor outcome. This test case was sliced with CURA
		A long print with a decent print quality initially but towards the end the print started becoming a poor result of for this test case. No variables no abnormalities tested in this case. This test case was sliced with CURA

		A short print with a poor print quality but the design printed properly; not many overlaps/over extruding. Slic3r was used to slice this print case. There was no support material enabled however compared with the other prints sliced with CURA, this Vader can be considered a decent test case that has been printed so far. The print quality increased as time passed.
		This print was a long print. STL file was sliced with slic3r. In this test case, no interference was introduced that could prevent a successful print. The end quality of the print is observed to be a successful test case.
		A long print with a decent quality. The STL file was sliced using slic3r. No support material was used for the test case, and considering so, it can be said this print is a successive print case

		comparatively to using supporting material
		A short print with pristine quality. The STL file was sliced with slic3r. There were no external factors incorporated from allowing the print to finish. Overall the quality of the print made this test case one of the best successive test cases.
		A long print with good quality test result. The STL file was sliced with slic3r. A print without any interference. Printed while multiple print cases were being executed at the same time. A successful test case
		A long print with a fair quality test result. The STL file was sliced via slic3r. The print took place parallel, in terms of time, to other test cases being printed. This test case is labeled

		as a successive test case.

Overall notes after testing:

- The slicing software that provided consistent high-quality prints was Slic3r. When slicing prints with Cura, the successful print percentage dropped nearly 50%.
- If the raspberry pi is to disconnect by powering off during a print, the print will not continue.
- If the raspberry pi was to lose internet connection, the print will continue, and the status can be updated once a connection is reestablished.
- If the raspberry pi is still connected during a print but the printer is shut off, the print will not continue if the printer is turned on again
- The current system is capable of printing 4 prints at once with a 100% success rate for all printers. The limitations for the current system include the budget and computational power of the raspberry pi.
- If an SD card is to start the print with the raspberry pi connected, the print will still work however Octorpint will not be able to read all the values such as print time left. However, some data such as the print time left will be displayed on the printer.
- If the print cable were to be disconnected and then reconnected the printer will need to restart the entire print.

After testing our product with two CEED members we received valuable feedback. They both liked the product and thought that there could be use within Maker Space. Their primary interest lied within Octoprints return values, allowing us to get print times, errors and filament

usage. They expressed that it would be nice to see all the printers unified onto one program so that you could see their stats, even if the SD cards were still used. The Ceed staff preferred the current way we chose to implement our design where users would walk into the space and would be able to see the UI on a large screen while being able to upload gcode wirelessly. A key feature preferred by a lot of our user's includes setting up a slicing profile which would allow users to slice their files and upload to OctoPrint from OctoPrint's web interface. After discussing our UI with the professor, he preferred that there was a system that could be integrated into our design which gave unique access credentials which would allow for individuals to access printers remotely from anywhere without having to be in Makerspace. However, this was beyond our knowledge and experience of coding.

6 Analysis:

After testing several prints using both Ultimaker Cura and Slic3r to slice the g-code, we found that the prints we got from Slic3r are of better quality than Cura. This can be seen in Figure 5Table 2, as the print on the left made by Slic3r is much cleaner and the print nozzle did not get stuck in the print, which happened in the head of the print on the right. One of the issues with Cura was that near the end of the print, the print head was getting stuck inside the print dramatically reducing the quality of the prints. This is not an issue we have faced thus far with Slic3r. Prints sliced by Slic3r have had more consistency and better quality than those sliced by Cura, so we have decided to use Slic3r as the 3D printing software on the design day.

After our group did the benchmarking for prototype III, we found a plugin in OctoPrint that could monitor the filament used during the print. It does this by measuring the filament excreted from the nozzle with the assistance of a built-in sensor which provides accurate data in terms of the amount of filament. We have added this to our UI because of its usefulness and because it does not cost us any extra money. If the filament is to run out, the plugin will pause the print. The dashboard will also display that the filament is out and a new spool needs to be added.

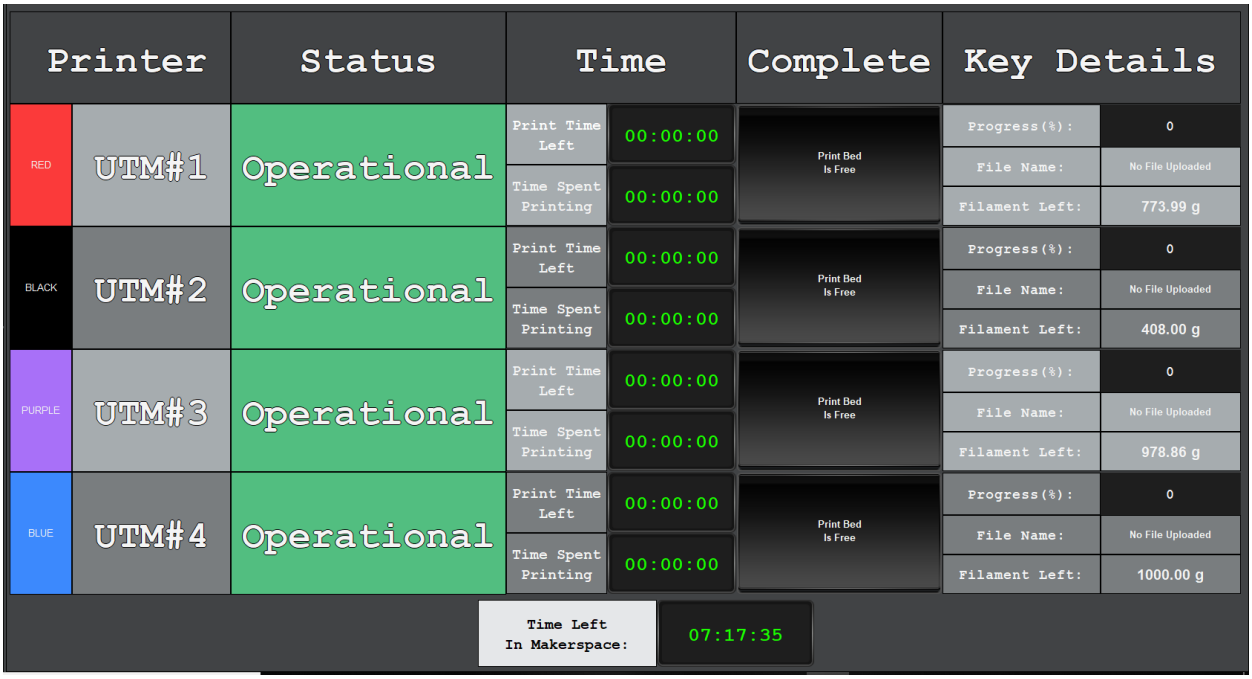


Figure 1: Different States of the UI

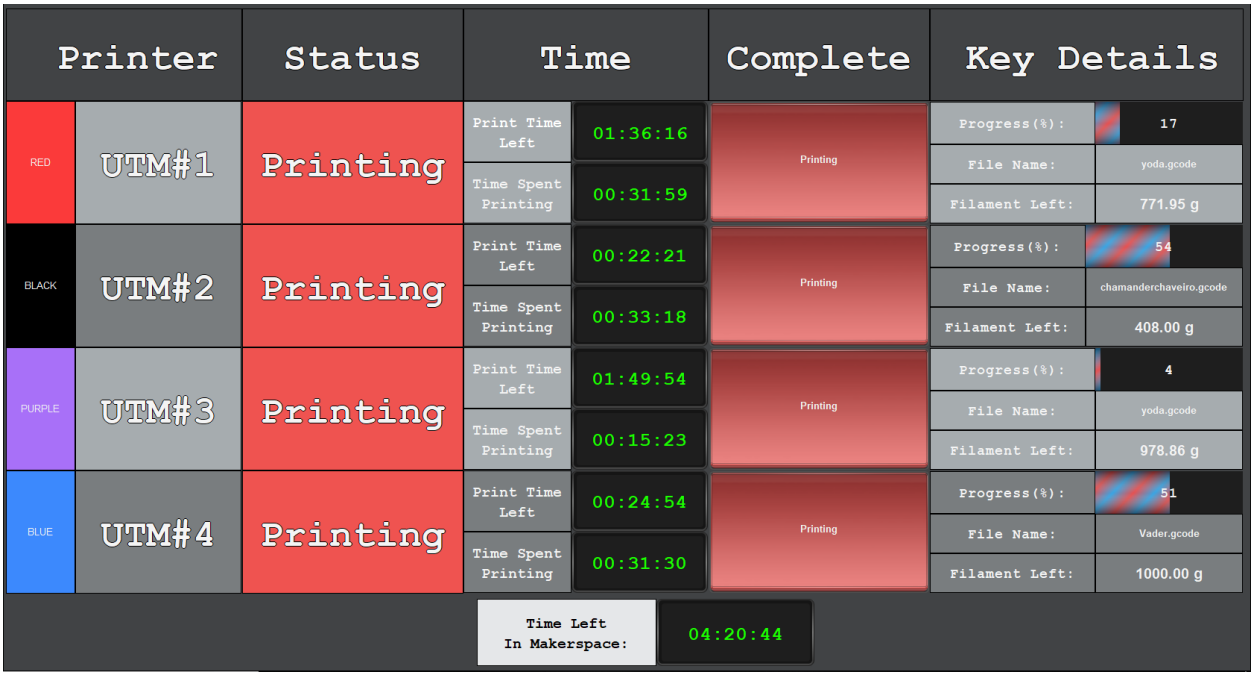


Figure 2: Different States of the UI

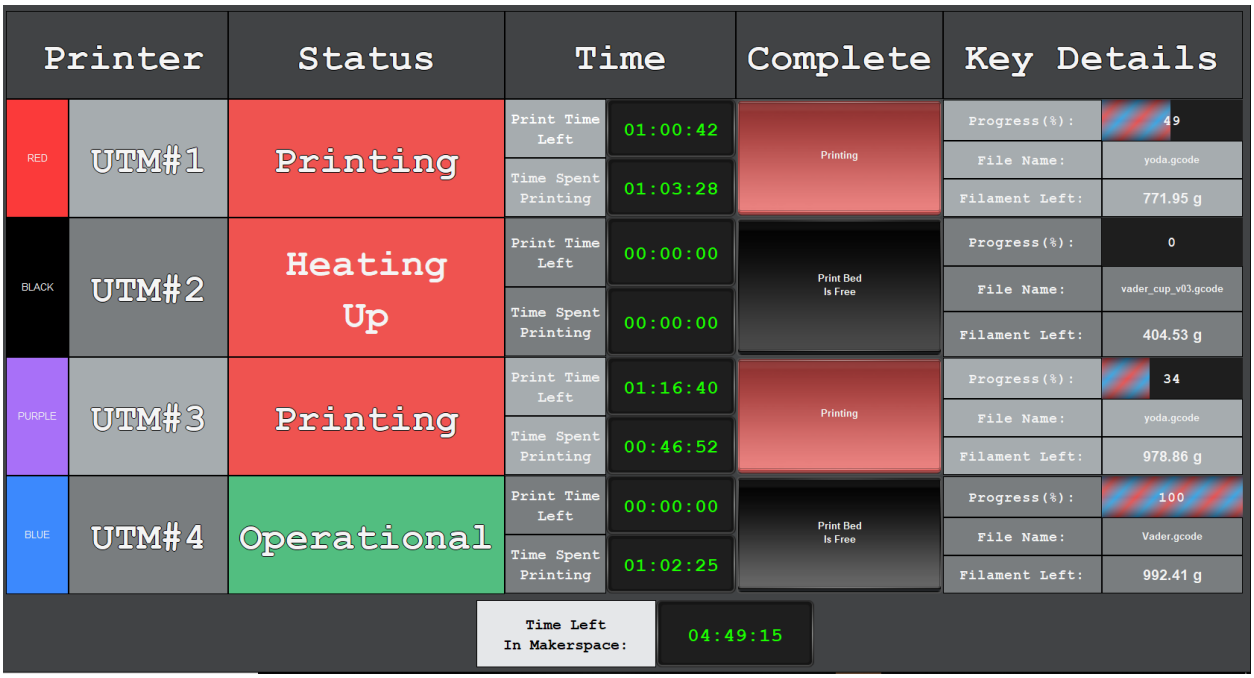


Figure 3: Different States of the UI

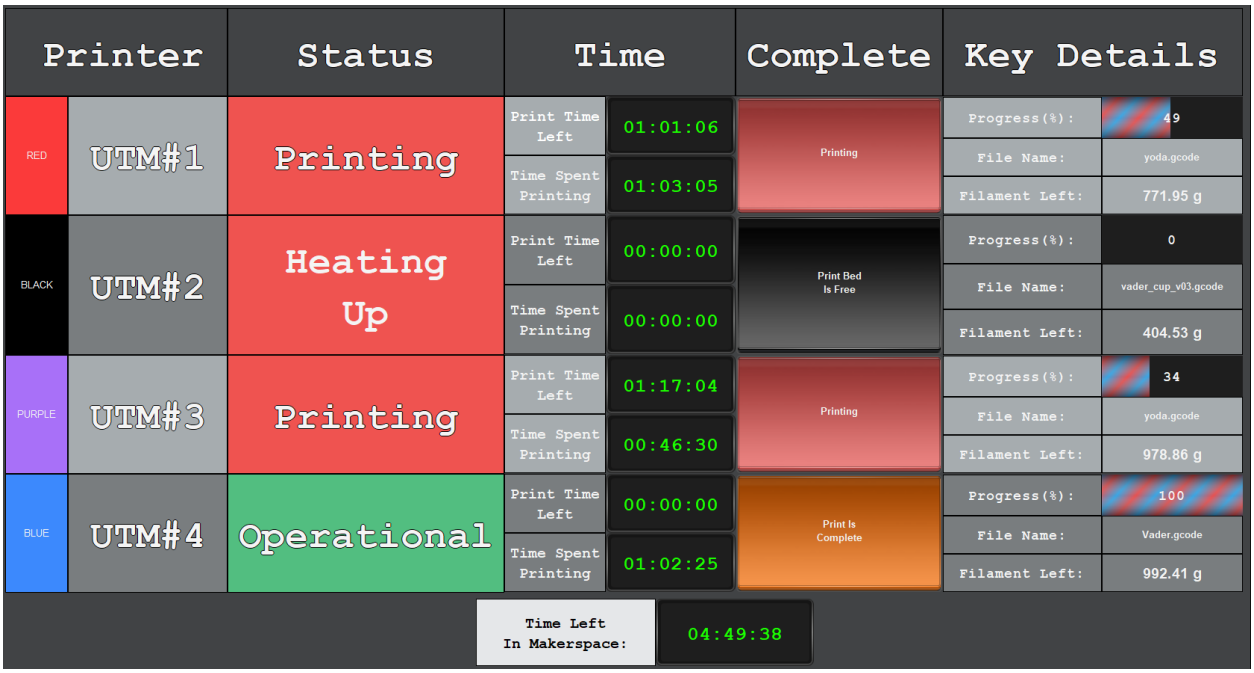


Figure 4: Different States of the UI

Figure 5: Side to Side Comparison of Slic3r (Left) vs Cura (Right)

7 Conclusions and Recommendations for Future Work

The results of our final prototype have pleased our whole group. We were able to get 4 instances of Octoprint working at the same time and 4 printers printing at the same time too. Our Dashboard also includes all the features we wanted it to have such as filament, time left, etc. Overall the group worked well in completing tasks.

One major takeaway from our experimentation is to delegate more tasks to different group members. Often most of our group were working on a similar task such as creating the UI or testing prints. A better and more efficient use of time would be to delegate some members to the UI, some to do the test prints and some to the deliverables and creating of our presentations. This would result in less cramming for deadlines, less confusion and a more consistent workflow. Using Trello, helped us stay on track to complete prototype three and to distribute the workload in the team. We plan on sticking to the created schedule to finish the PowerPoint work.

Another problem that our group had was that we were not reasonable with the fidelity of our project at times. For example, we wanted to have the labels of the printers move into different columns at first but found out that it was too difficult and time consuming to do. We figured that changing the colors to display different states would be a much easier solution and would be easier for the users to navigate. We had also wanted to make OctoPrint update its password allowing only the user that signs out the current printer to be able to access the instance and control the printer. This idea however had low fidelity because the major constraints traced to our group’s sufficient lack of knowledge in coding and editing the open source code.