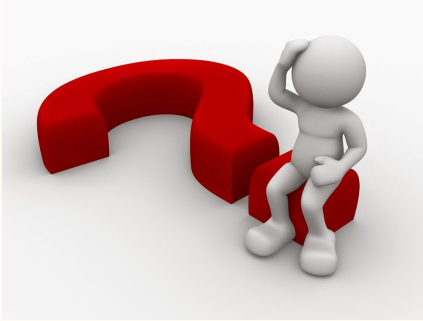
3D Printer Management System

— Group 9

By: Michael Puma, Michael Dolan, David Coco-Bassey, Liam Kennedy, Gazi Israpilov

Problem Statement

"A **need** exists for the CEED employees that **optimizes** the CEED building through **automation** using **Ross Dashboard** and a **microcontroller**; that is under the $100 budget and is easy to operate and maintain"

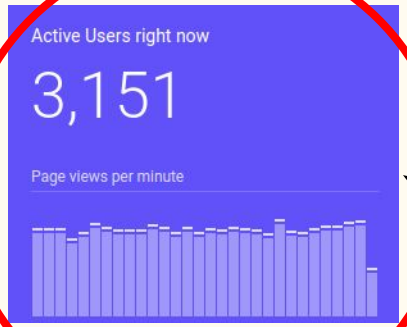


Design Criteria

#	Interpreted Needs
1	Must have a degree of autonomous function
2	Must use RossDash Board
3	Must be easy to understand and use
4	Must be within budget
5	Must be inexpensive and simple to replicate
6	Must be easy to modify (i,e. useful if changes are made to the CEED facility)

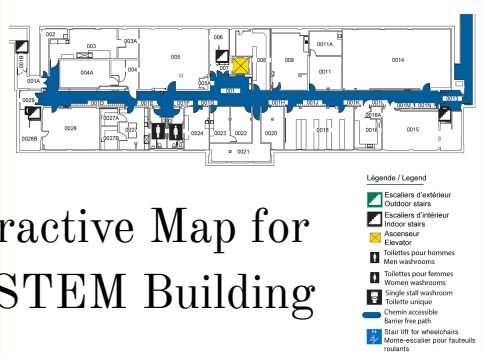
Priority	Design Criteria (Functional/Non-Functional)
1	Assists/replaces CEED worker
2	Understandable by user
3	Useable by user
4	Assembled easily
5	Easy to upgrade or modify

Brainstorming Concepts & Ideas

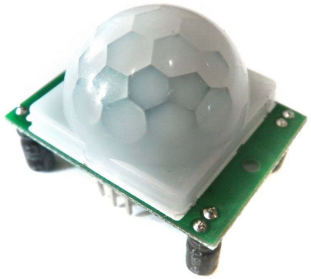


Printers in use

SD Card management system for 3D Printers



Interactive Map for the STEM Building



Student Counter for popular rooms

VR Station Timer and Queue

Ideate

Using a raspberry Pi create a system that allows the easy signing out and return of SD cards corresponding to 3D Printers in the makerspace employing the use of QR codes on the SD cards.

Needs

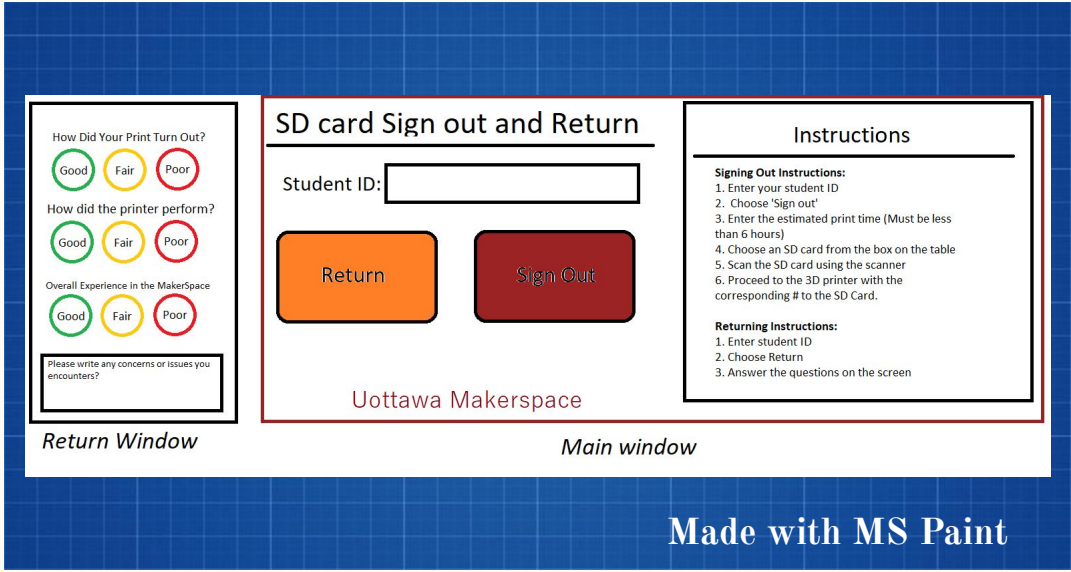
- Simple
- Fast
- Error free
- Expandable

Conceptualize and Design

Different Ways of creating

1) Students having full control of signing out and returning SD card with interactive terminals set up in the makerspace

2) CEED Staff member operates, with Ross storing and tracking the use

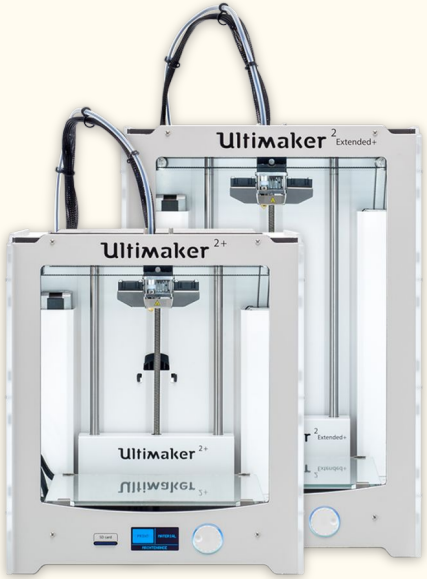


Client Meeting

- Liked the idea of QR Codes for scanning
- CEED staff member controlled instead of students
- Record data on printers
- Collect feedback on prints and the printer

Finalized Idea

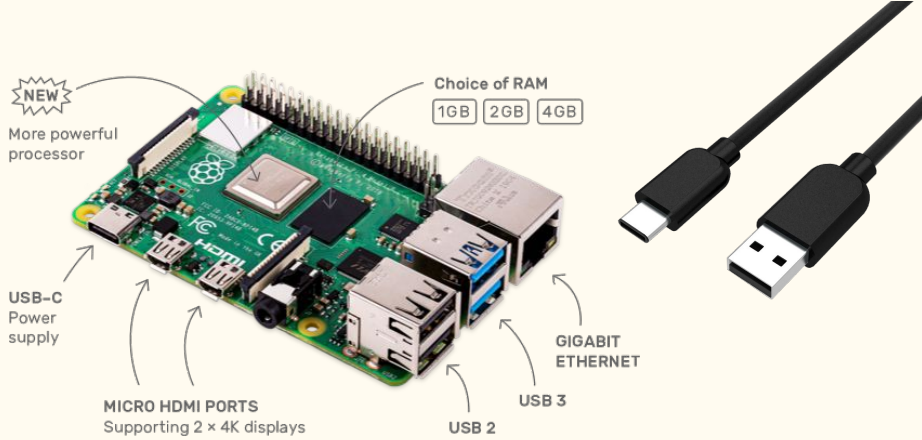
- Raspberry Pi based QR scanner that tracks the usage of printers through their SD cards
- Records data on printer usage & print times
- Collects user feedback through emails tied to makerspace ID
- Ross Dashboard GUI for CEED members sign out/return and to track print times
- Linked with makerspace website to show active printers

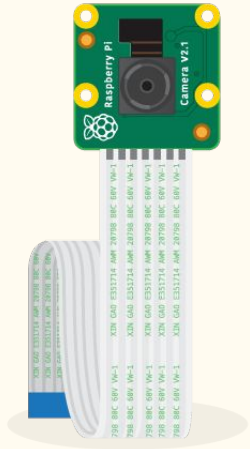


Costs and Materials

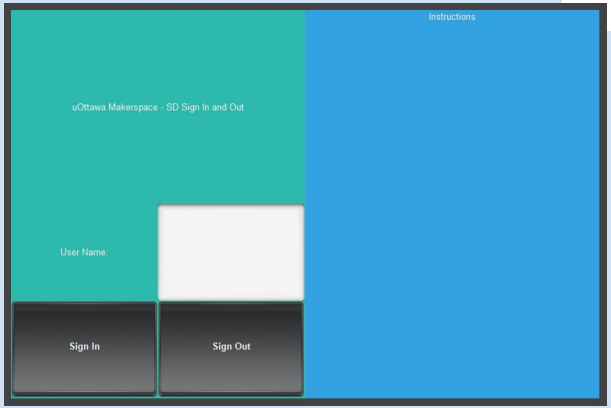
Material	Cost
Raspberry Pi 0, Case, Cable Bundle	34.98
*Raspberry Pi 4	46.95
*Raspberry Pi 4 Case	6.95
Camera Module	32.98
*USB-C cable	4.30
SD Card 16gb	4.99
Final total*	**96.17** + tax
Original total	**72.17** + tax

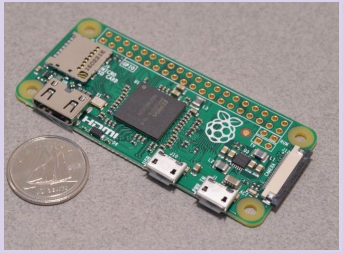
*Added later due to failure of first prototype





Prototype I





Dashboard User Interface	
Pros	**Cons**
- Colors are inviting - Easy to understand	- Designed for a student, not a staff member - Format & box shapes are awkward - Lacks functionality

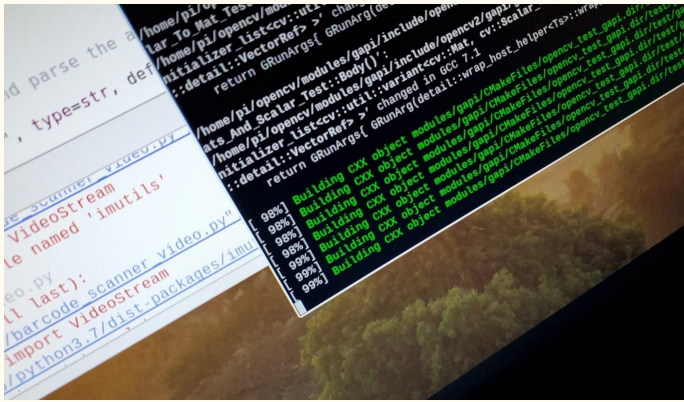
Raspberry Pi 0 W	
Pros	**Cons**
- Small - Inexpensive	- Too slow: Not powerful enough to run the software for more than 2 minutes before crashing - Overheating issues

Prototype 1, What's Next?

1. Optimize the code, to reduce stress on the hardware
2. Improve cooling, and test again

Results:

- Despite efforts, the Pi continued to overheat
- The program was to power hungry for the Pi 0 W, another solution must be found



The Code took 4 Days to compile, due to the nature of the Pi

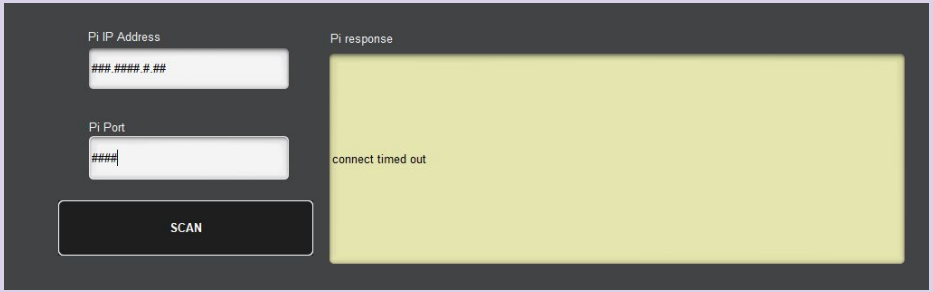
Prototype II

SD Card Port for upgrading software

Camera

Ethernet Port for more stable internet connection

Prototype II



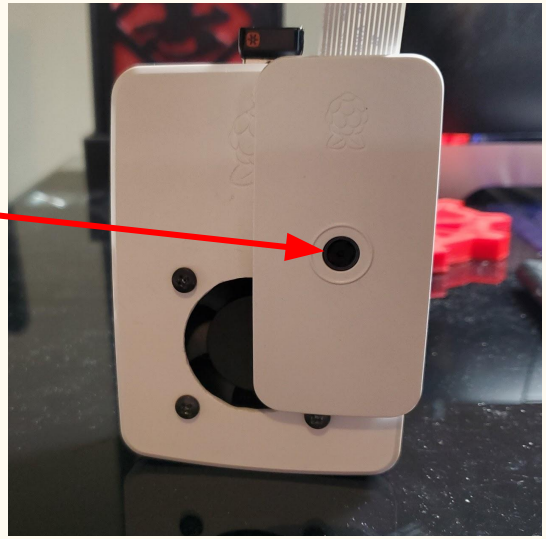
Dashboard User Interface	
Pros	**Cons**
- Redesigned for staff members - Easy to understand - Functional	- Not very inviting

Raspberry Pi 4	
Pros	**Cons**
- Powerful - Runs software without issue	- Larger size - More expensive - Overheating issues when running for long periods of time

Final Product

Final Build

Camera



Added fan for active cooling

By providing active cooling the Pi runs both the server and the scanning software simultaneously while staying cool

The QR Codes

uOttawa
3D Printer 2

uOttawa
3D Printer 3

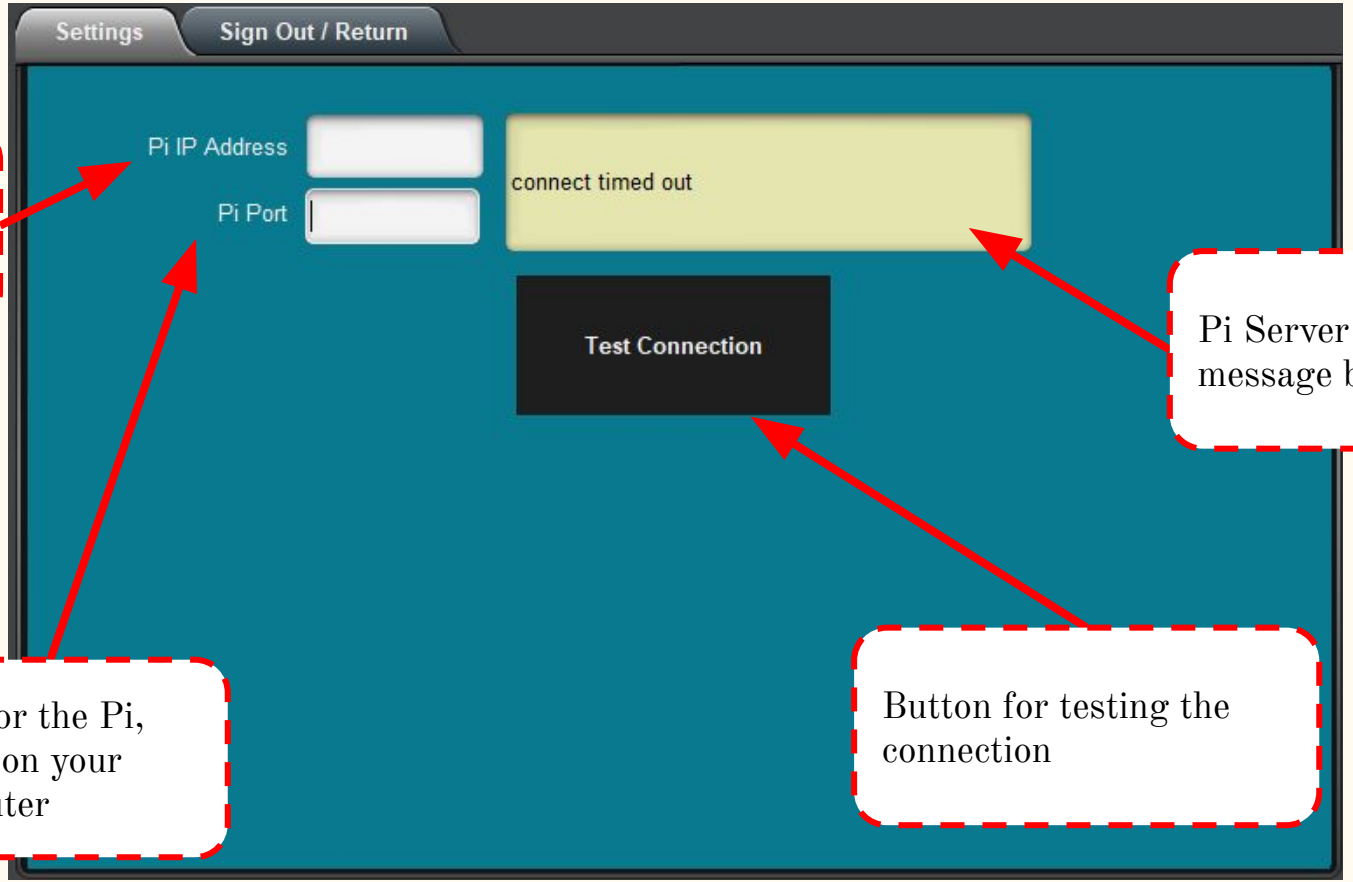
uOttawa
3D Printer 4

uOttawa
3D Printer 5

uOttawa
3D Printer 6

Dashboard

Settings Tab



Text box for the Pi's IP Address

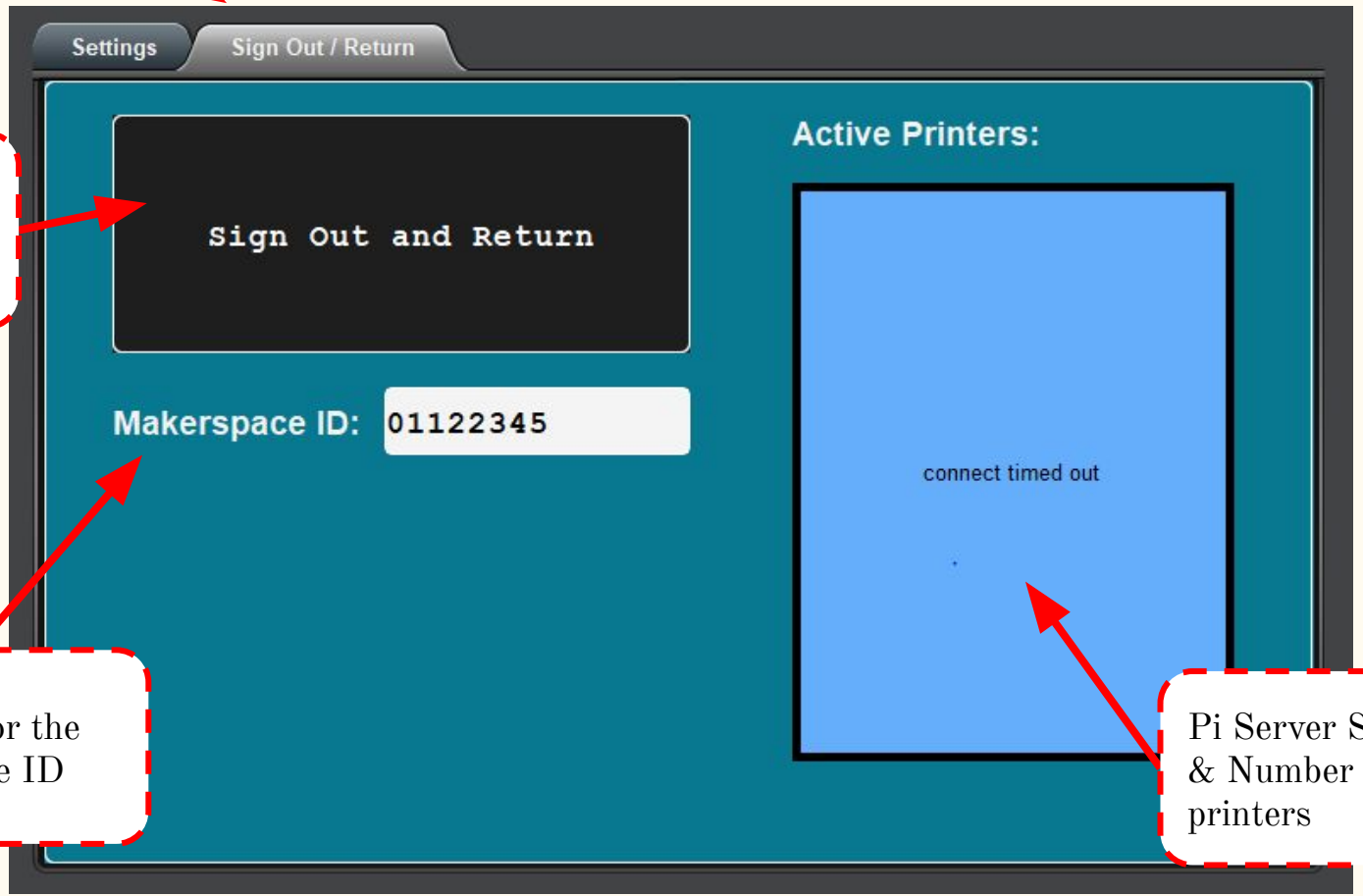
Pi Server Status message box (offline)

Port for the Pi, found on your computer

Button for testing the connection

Dashboard

Sign Out/ Return Tab



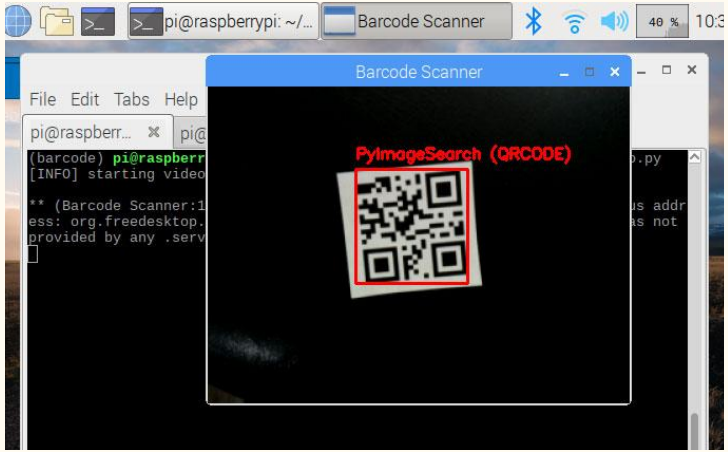
Sign Out & Return Button

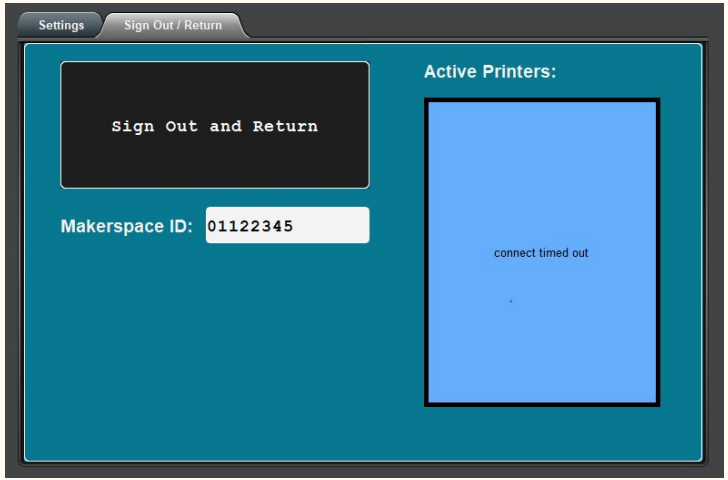
Text box for the makerspace ID

Pi Server Status (offline) & Number of active printers

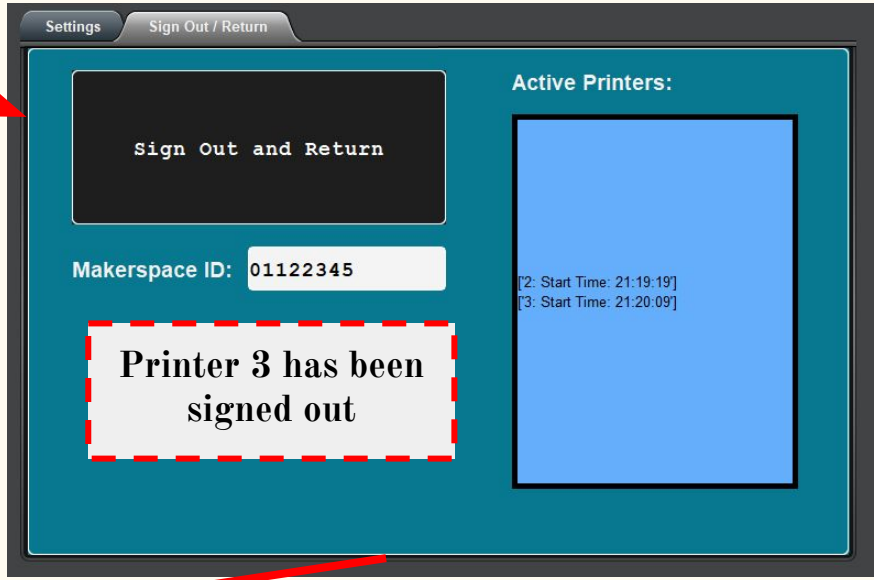
How it Works

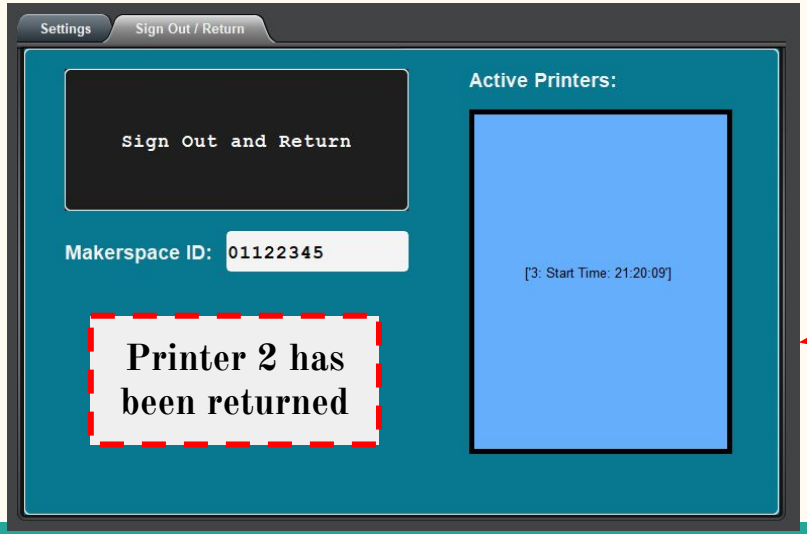
1. The Pi creates a server, that can be accessed from the dashboard software
2. On dashboard the Pi's IP Address and port is entered to initialize the connection
3. The QR code on the SD card is flashed in front of the camera on the Pi, where it is automatically scanned
4. On dashboard the signout/return button is pressed, and the 3D printer is automatically added or removed from the 'Active Printers' list. The start time is also recorded



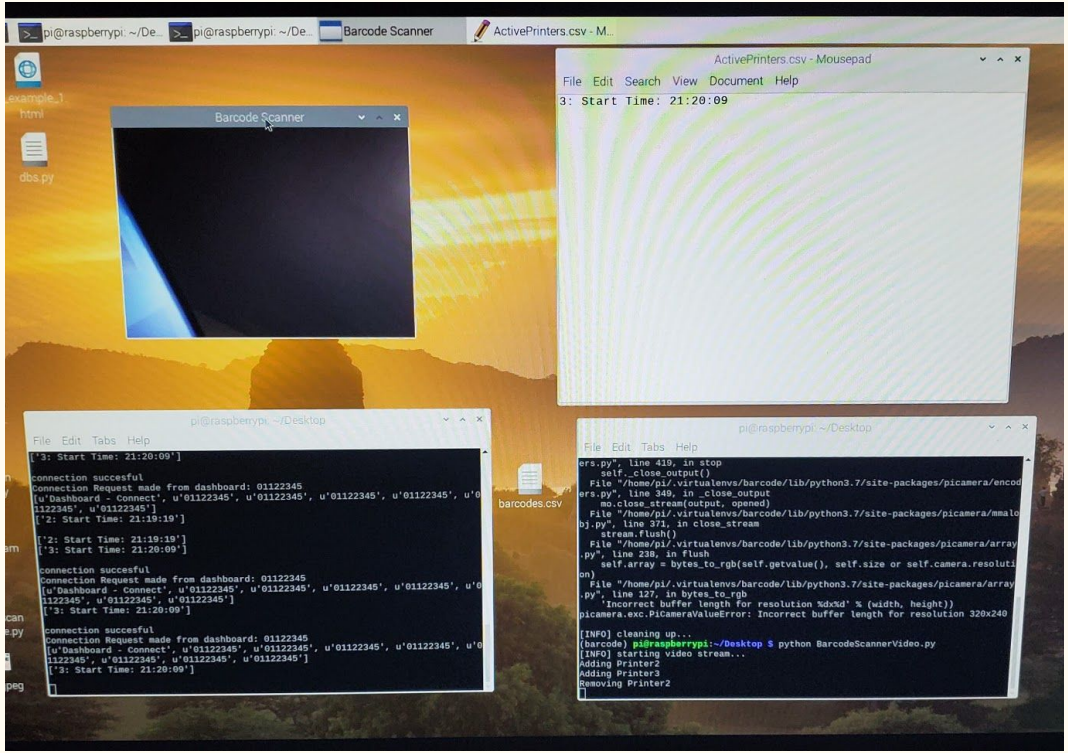


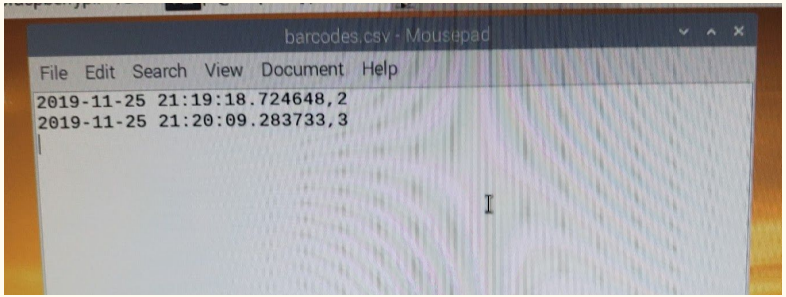
The User Experience





Raspberry Pi Desktop





The Pi logs all prints added and removed, as well as their start and finish times in a CSV file

1. QR Code scanning software (top & bottom left)
2. Active Printers CSV (Top right)
3. Server for dashboard (Bottom right)

Behind the Scenes

The Pi runs two separate codes simultaneously. One on the Pi itself the other on a “**virtual** environment”

The Virtual environment

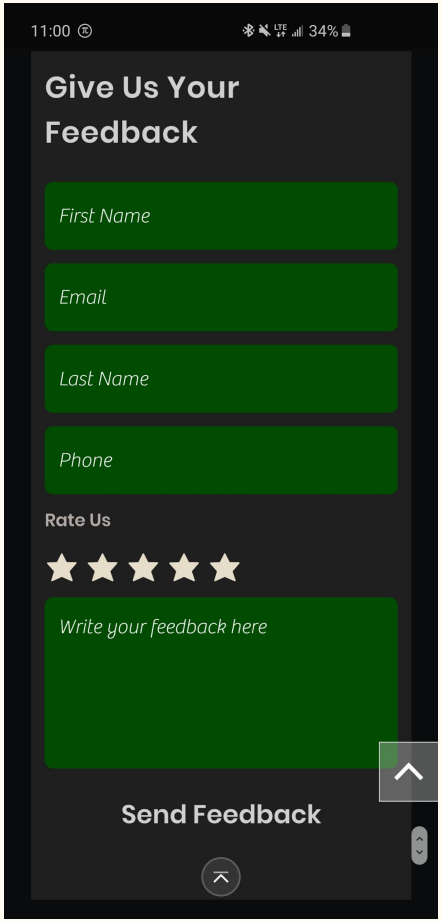
- Reads the QR codes by analyzing video from the camera in real time
- When found the code is scanned, data is collected, and added to a csv file saved to the desktop of the Pi

Regular

- Connect the Pi to dashboard by creating a server that runs on the Pi itself
- Reads the csv file when the button on dashboard is pressed and sends the data

Collecting User Feedback

- To gather feedback from the user, an email is sent with a link to our website
- The Pi automatically sends the email by using the universities database connected to their student ID
- Due to lack of access to the University servers, this is not included in our final project



Lessons Learned

1. Optimizing code is crucial when using low processing power devices such as a single core Raspberry Pi
2. Just because something will work on paper does not mean it will work in actuality, hence the failed Pi 0W
3. Start planning earlier; starting during the beginning weeks would have allowed this projects end result to be more polished

Questions?