

Secure & Remote 3D Printing

Nick Contrell, Carl Mann, Tiffanie Petersen & Isaiah Thomas

Faculty Advisor(s):Dr. Siddhartha Bhattacharyya, Dept. of Computer Engineering and Sciences, Florida Institute of Technology



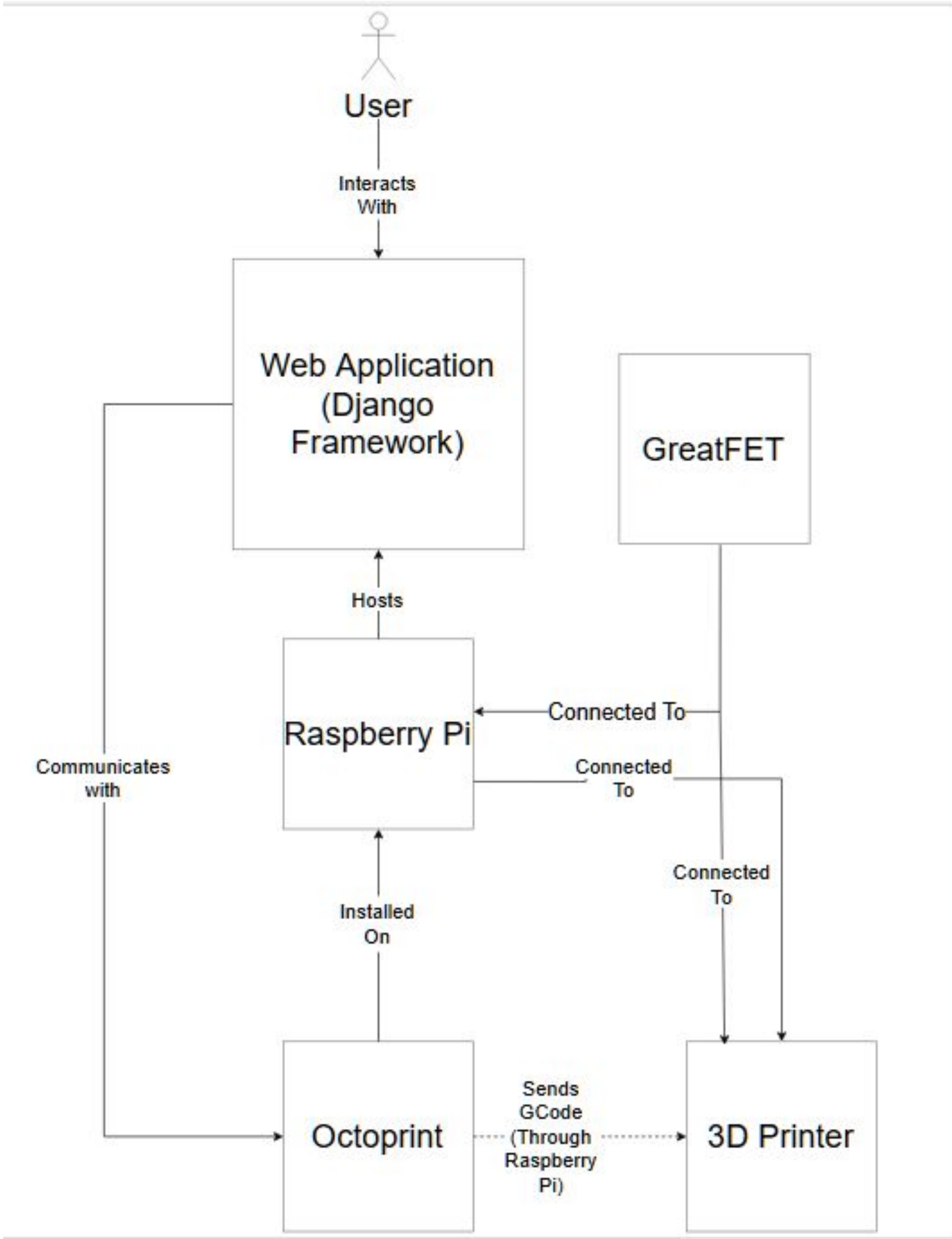
Motivation

- Currently there are exploits for the 3D printer which causes the printer to stray from the original design to create injected modifications
- 3D printers require hands on activity which many users would like to mitigate
- Adding a remote way to monitor the printer would allow administrators to multitask

Goal

- Develop a web application to remotely print an uploaded 3D model
- Have a secure line of communication form user to web server to printer
- Allow administrators to control a queue of print requests and provide them with the tools necessary to moderate which files should be printed

Design



Features

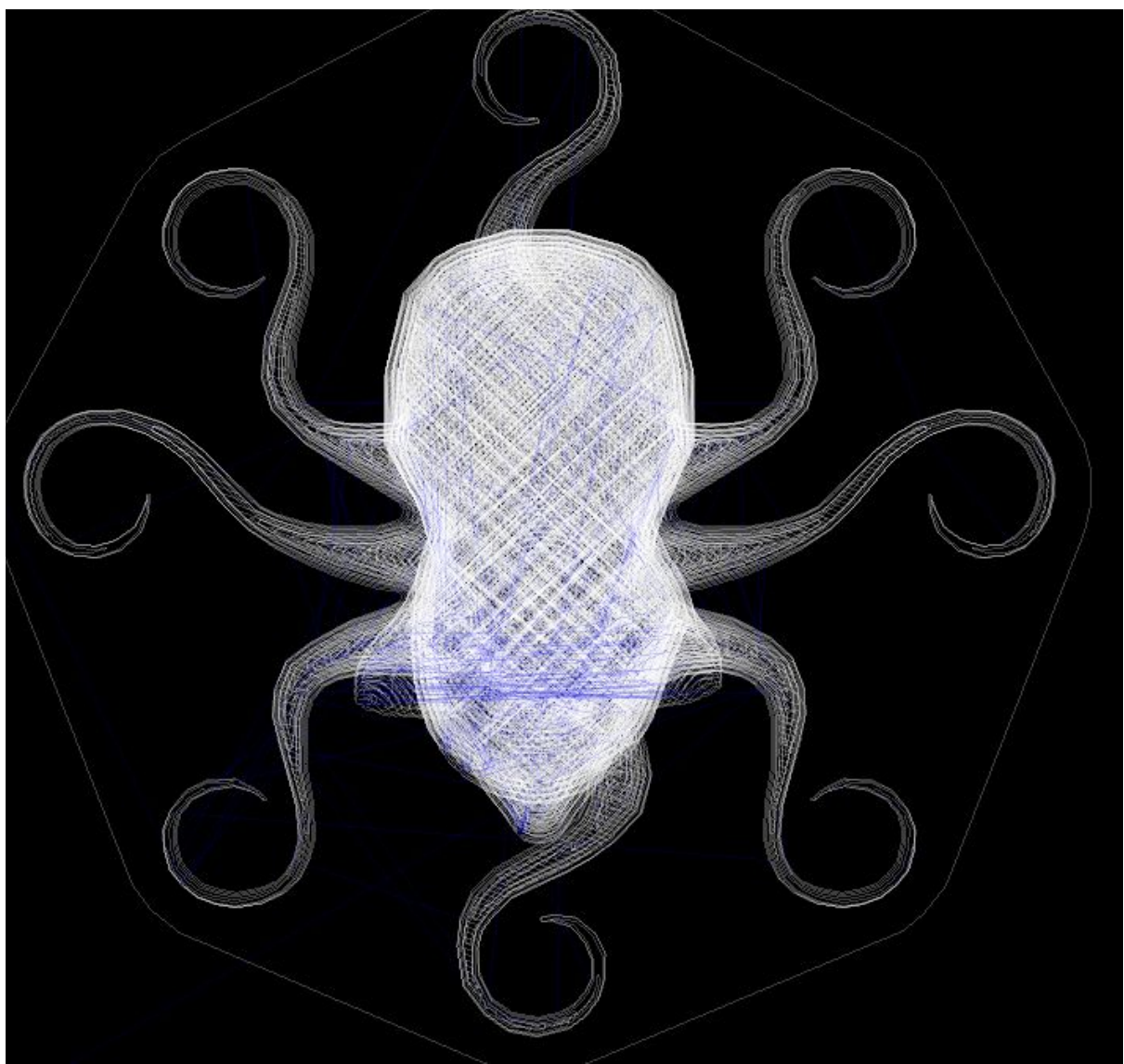
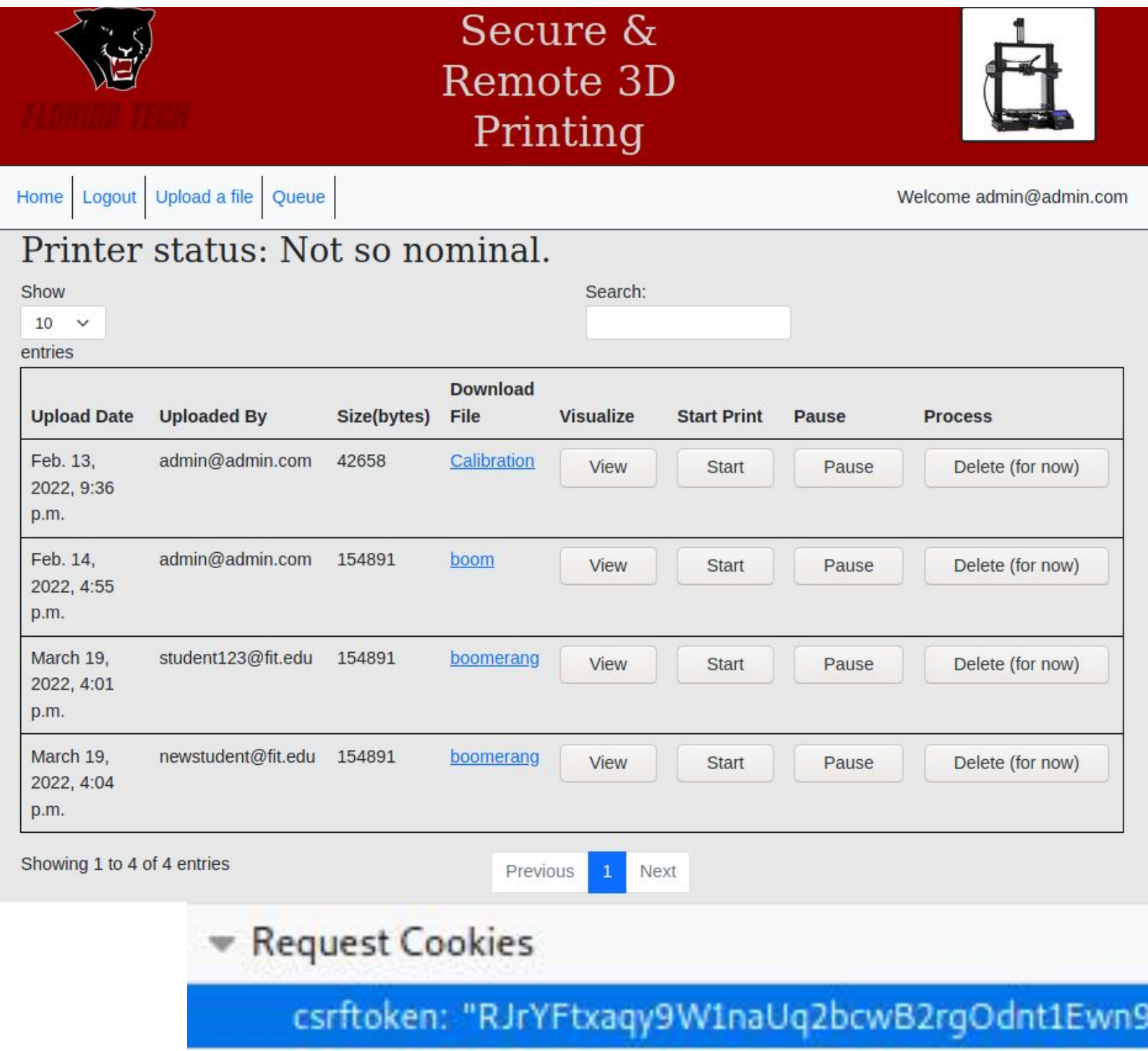
- Website features
 - Users may create an account, login, logout, and upload files to be screened by an administrator
 - Queue view for administrators to monitor, view, and print models
 - Interfaces with Octoprint via REST API calls to pull relevant data and control printer operations

Security Features

- By design
 - Website, octoprint, and file server are all hosted on a raspberry pi with docker
 - Containers prevent an attacker from listening to any internal communications
- Additional measures
 - Encrypted communication channels include HTTPS between the user and the website as well as a direct USB to serial connection between the pi and the printer
 - A Django CSRF Token is used by the server to provide a user with a unique connection specific value to be included in the HTTP requests
 - Extensive file checks prevent users from uploading malicious code and files are stored outside of the projects scope
 - Users are required to create an account in order to upload files

Evaluation

- Recommended features
 - preview 3D projects (implemented)
 - Improve upon the websites appearance
 - Provide users with more feedback relating to their requested prints
- Issues
 - Cannot establish a MiTM connection to the 3D printer reliably, limiting our ability to fuzz traffic



Security Test	Response
Invalid input for usernames, passwords, & file uploads.	Website denies user until the input follows the specifications.
Wireshark traffic captures to test encryption of data.	The data is encrypted and the scheme is hidden.
Gcode bounds checking.	The application does not allow a print to exceed the constraints of the 3D printer.
Sniffing on the network.	No sensitive data was leaked due to the containerized nature of the processes.

Conclusion

- Users can print remotely while knowing that their projects will come out as expected
- Administrators may view and approve files that have been scanned to ensure they are not malicious
- The web application tracks each of the user uploads and places them in a queue

Future Work

- Establish a connection to the printer to fuzz gcode input to ensure files cannot exploit unforeseen edge cases
- Add a contact page to reach out to an admin
- Setup an email server to notify users of print job progress