Secure and Remote 3D Printing User/Developer Manual

Tiffanie Petersen - tpetersen2018@my.fit.edu

Isaiah Thomas - ithomas2018@my.fit.edu

Carl Mann - cmann2013@my.fit.edu

Nick Contrell - ncontrell2019@my.fit.edu

Sponsored by
Dr. Siddhartha Bhattacharyya - sbhattacharyya@fit.edu

Table Of Contents

Table Of Contents 2

User Guide 3

1.1 Getting started 3

1.2 File Options 5

1.3 Security 5

Developer Guide 5

2.1 Installation 6

2.2 Dependencies 6

2.3 Setup 7

2.4 Local development 7

2.5 Pushing changes 8

2.6 Project structure 8

2.7 Diagrams 9

Deployment Guide 12

3.1 Printer 12

3.2 Raspberry pi 12

3.3 Repository 12

3.4 OctoPrint 12

3.5 Docker compose 12

GreatFET and Facedancer 14

4.1 About 14

1.2 How to use 14

Website Functionality Guide 15

5.1 Accounts 15

5.2 File Upload 15

1. User Guide

1.1 Getting started

Our project was created with two specific end-users in mind; students and faculty. The first goal was to provide a functioning web application with which normal users can upload files, view their jobs' statuses in a queue, and log in or log out. For administrative users, we wanted to ensure the ability to display the status of printing requests, accept/decline printing requests, and allow for profile bans.

Students:

A basic demonstration of creating an account may be found here. All accounts are required to register with a valid Florida Tech email address and are validated by administrators before printing jobs are accepted.

User Accessible Pages	**Description**
Home page	Main page of the website which displays general information as well as directs users to other pages.
User Registration	A webpage that will allow a user to request an account on the website.
Admin approval of registrant	Upon registration with valid credentials send off user information for admin approval.
Saving user information	To finalize account creation user information is saved in the database.
Florida Tech styling	Modeling user interface off of Florida Tech's general styling
Mobile friendly interface	Webpages are readable and clean on browsers in mobile devices
User Acknowledgement	The web application will display a greeting message based on the signed in user's information.
Navbar	A navigation bar displaying links to various

	pages on the site. Used for traversal of the website.
Login page	Page for user authentication. Functional for all user types
Logout	Option provided along navbar after successful sign-in that allows user to logout
Submit button	Button which sends user input using post request.

Faculty:

Features	**Description**
Home page	Main page of the website which displays general information as well as directs users to other pages.
User Registration	A webpage that will allow a user to request an account on the website.
Admin approval of registrant	Upon registration with valid credentials send off user information for admin approval.
Saving user information	To finalize account creation user information is saved in the database.
Florida Tech styling	Modeling user interface off of Florida Tech's general styling
Mobile friendly interface	Webpages are readable and clean on browsers in mobile devices
User Acknowledgement	The web application will display a greeting message based on the signed in user's information.
Navbar	A navigation bar displaying links to various pages on the site. Used for traversal of the website.
Login page	Page for user authentication. Functional for all user types

Logout	Option provided along navbar after successful sign-in that allows user to logout
Submit button	Button which sends user input using post request.

1.2 File Options

Students:

Faculty:

1.3 Security

The scope of this project was to create remote printing capability for the printer via a web application and ensure the security of the 3D printer from cyber threats. Some key security measures implemented to protect remote users and the client in this project include HTTPS, CSRF tokens, uploaded file screening, authorized user whitelist, administrative control of prints, and non-exposed local handling of files between the printer and the site. While our application is tailored to the specific infrastructure in the lab, we have designed a platform that may be deployed and scaled up with relative ease should there be a demand for printing services.

The purpose of this project was to provide students and faculty outside of the printer lab an ability to request their 3-D model be printed while ensuring the protection of the lab's Ender-3 printer and its network. This was accomplished by deploying a secure web application that allows authenticated users to upload GCODE files that are placed in a print queue by administrators. Throughout development, our team maintained regular communication with clients and faculty in order to keep fulfill expectations and stay on track with their vision. When it comes to the importance of our client's work we are looking to mitigate situations in which injection or exfiltration may occur.

Developer Guide

2.1 Installation

Setting up SSH for GitHub:

- Git clone git@github.com:IsaiahST2020/SeniorDesignProject.git
- Navigate to ~/.ssh
- ssh-keygen -t rsa -b 4096 -C "SeniorDesign NAME"
- Then copy the generated id_rsa.pub to Github
- Finally, enter the following command in your cloned repository
    - git remote set-url origin git@github.com:IsaiahST2020/SeniorDesignProject.git

Setting up pipenv:

- To have this environment run as smoothly as possible, you can run the following commands to get a working setup for production or development. To get all the dependencies independent of the system, you will need to install pipenv

  debian/ubuntu:

  ```
  sudo apt install pipenv
  ```

  every other system:

  ```
  $ pip install pipenv
  ```

2.2 Dependencies

Installing dependencies with pipenv:

- Once that is done, navigate to the workspace and run the following commands

  ```
  $ pipenv install
  ```

Installing dependencies and development libraries:

- This includes things like pylint and autopep8 to help keep formatting and documentation consistent.

```
pipenv install --dev
```

2.3 Setup

Running the application:

- If you want to run the server using the virtual environment provided by pipenv you can follow these steps on a fresh install

```
pipenv uninstall --all
pipenv install --dev
pipenv shell
sudo python3 -m pip install -r requirements.txt
pip freeze
mkdir data
cd data
touch db.sqlite3
cd ..
sudo python3 manage.py makemigrations
sudo python3 manage.py migrate
sudo python3 manage.py createsuperuser
sudo python3 manage.py runserver
```

- And for post-setup development

```
pipenv shell
exit (to leave shell)
```

2.4 Local development

The website runs on the Django framework. One great resource we found for picking up how to use this framework and its key features was the video here. This video covers how Django operates. Some of the topics covered include models, views, databases, and security measures. Most of the local development for this project was done using sublime on kali Linux virtual machines. On the Github repository, you may see both sublime and visual studio configuration files to ensure your editor uses spaces rather than tabs.

2.5 Pushing changes

In order for changes related to models and various other important structures in the Django framework to be realized a user must migrate them with two simple commands.

- Python3 manage.py makemigrations
- Python3 manage.py migrate

Most other changes take effect immediately upon saving. An example of changes which take effect during runtime include html, views, and admin files.

2.6 Project structure

2.7 Diagrams

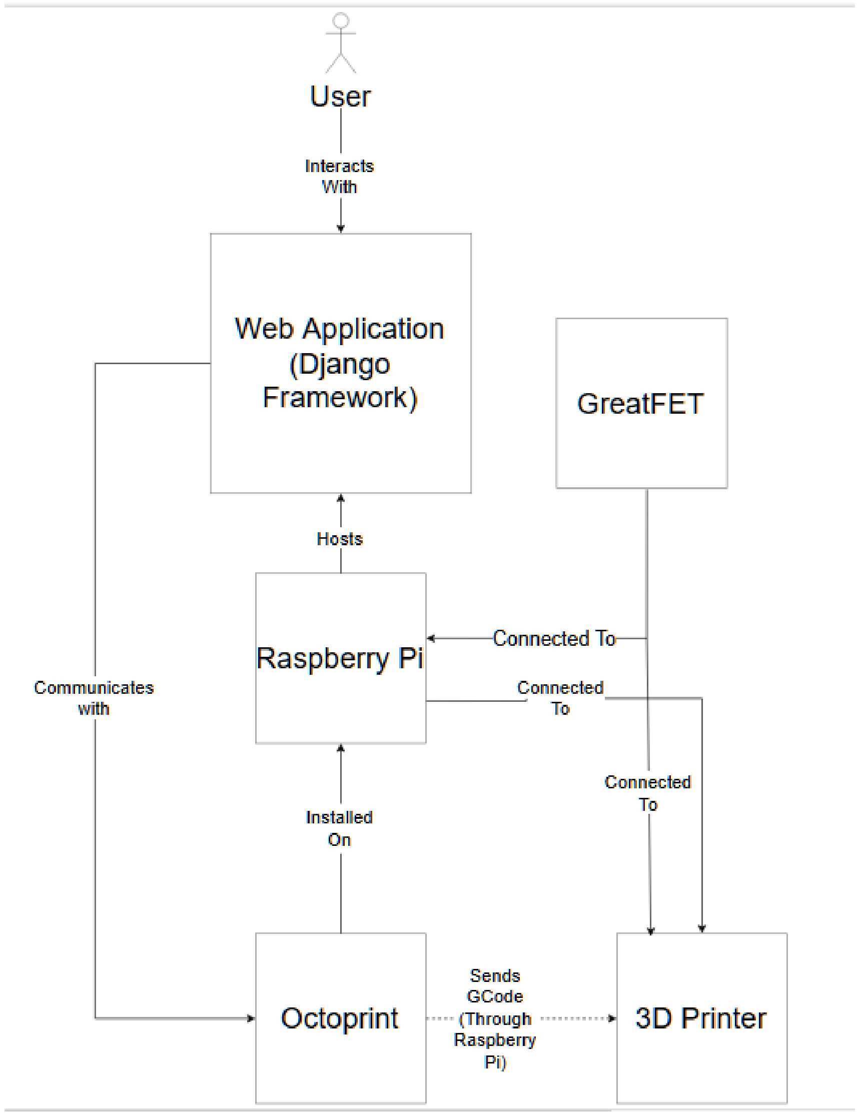


Figure 2.7.1: Framework diagram

The framework design shows how the software and hardware interact with each other. The main part of this project takes place with the Raspberry Pi because it stores all of the data from the web application, runs the docker image, interfaces with octoprint, and connects to the 3D printer and the GreatFET.

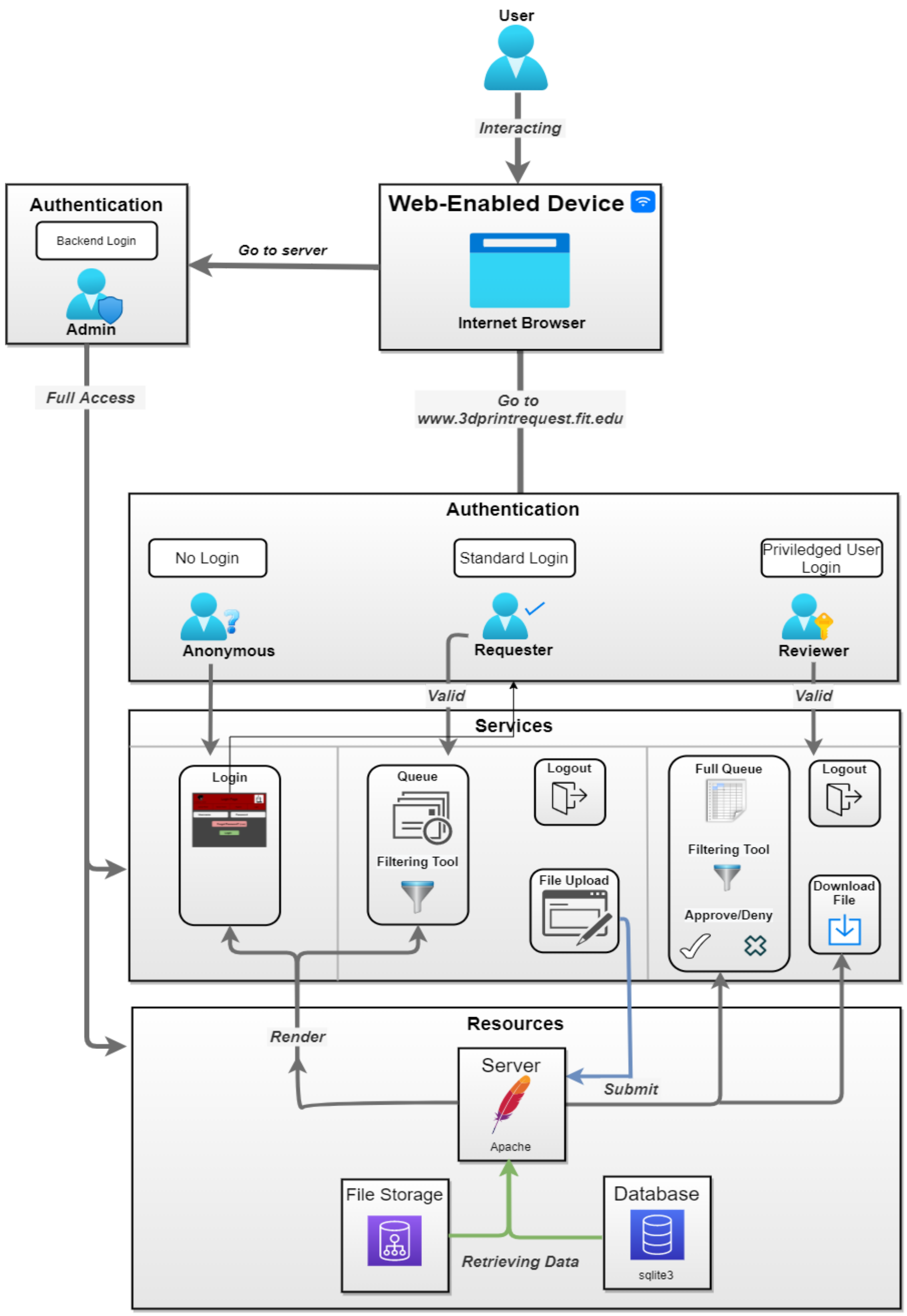


Figure 2.7.2: Web services diagram

Figure 2.7.2 shows how a user will interact with the web server. An unauthorized user will have limited access to the web application to keep the traffic to a minimum while also ensuring that sensitive information cannot be accessed by the public. The next type of user will be authorized or logged in, which will allow the user to see more webpages and gain access to uploading files which an administrator can print. The Privileged logged in user (or administrator) will be able to see a full queue of all the projects waiting to be approved and printed as well as deleting users, downloading files from the databases and more.

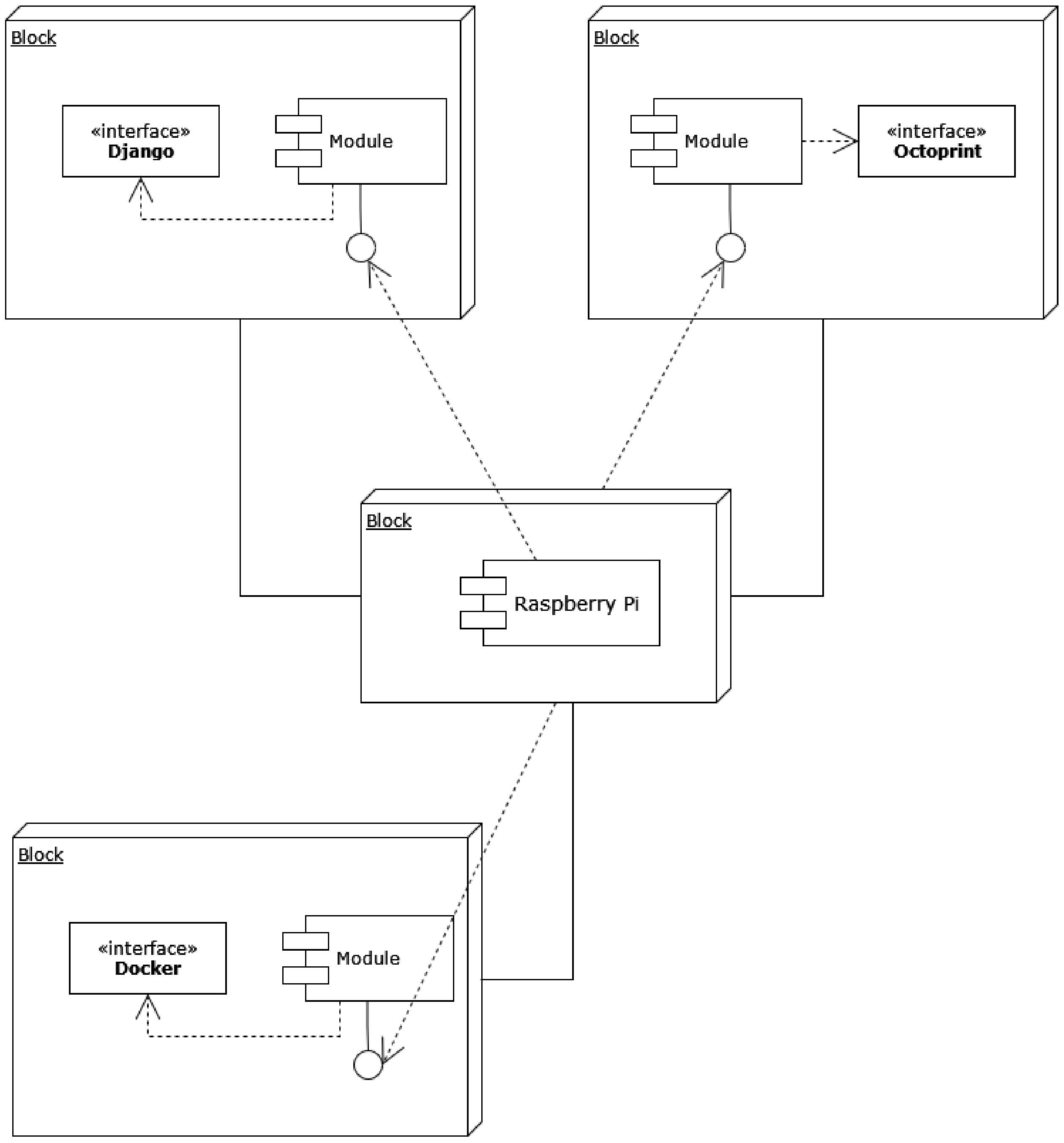


Figure 2.7.3: Software deployment diagram

The software that is needed for secure and remote 3D printing include Docker, Octoprint, and Django. These three software products will be downloaded onto a Raspberry Pi to interact with each other to provide the remote experience of 3D printing. Django will interface with Octoprint to ensure that each project fits the specifications of the 3D printer and ensures that the gcode file is not harmful to the printer.

2. Deployment Guide

3.1 Printer

An excellent guide for setting up an Ender3 may be found here. The guide covers more than enough information to use as a reference during the maintenance of the Ender3. We recommend that you find an experienced user of the Ender3 to help with calibration, troubleshooting, and testing. On the repository and the raspberry pi currently hosting the demonstration setup, there are calibration gcode files that may be used to ensure the printer is in working order.

3.2 Raspberry pi

A raspberry pi 4 is a great choice for hosting the website and OctoPrint as it is Wifi ready after setting up Raspbian. By containerizing each of the applications on the raspberry pi we effectively create a secure environment by design. Unless an attacker has gained access to the pi itself, API communications between the website and OctoPrint are secure by design.

3.3 Repository

The repository for this project is hosted on GitHub and may be found here. The repository was built for the purpose of making deployment onto a Raspberry pi with docker easy with a few simple commands. This is made possible by docker-compose as shown in section 3.5.

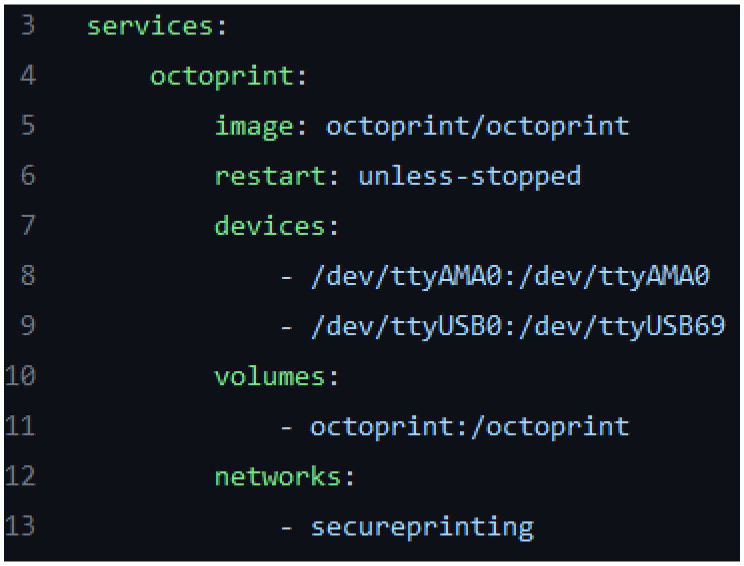
3.4 OctoPrint

Octoprint acts as a proxy between the website and the printer. During the end phase of website development, we shifted focus to implementing a channel for communication with a printer. To accomplish this we used Octoprint to handle the backend components of the application such as keeping track of the printing process. Octoprint is vital because it is capable of keeping track of the queue of projects running while also allowing administrators to collect and swap projects by using simple REST API calls.

OctoPrint Documentation

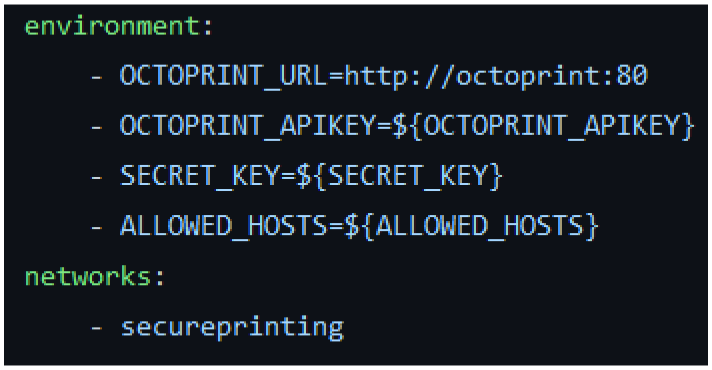
3.5 Docker compose

The file docker-compose.yml is crucial for the easy and complete deployment of both the website and OctoPrint. Figure 3.5.1 below shows a snippet of the file in which the USB devices of the pi are exposed to the container to be used by OctoPrint. This step is one of only 3 ways in which the containers have external access to pi's operation. The device "ttyAMA0" is the interface in which OctoPrint physically connects to the printer and exchanges data.



```
services:
    octoprint:
        image: octoprint/octoprint
        restart: unless-stopped
        devices:
            - /dev/ttyAMA0:/dev/ttyAMA0
            - /dev/ttyUSB0:/dev/ttyUSB69
        volumes:
            - octoprint:/octoprint
        networks:
            - secureprinting
```

Figure 3.5.1: Defining what hardware to expose to OctoPrint



```
environment:
    - OCTOPRINT_URL=http://octoprint:80
    - OCTOPRINT_APIKEY=${OCTOPRINT_APIKEY}
    - SECRET_KEY=${SECRET_KEY}
    - ALLOWED_HOSTS=${ALLOWED_HOSTS}
networks:
    - secureprinting
```

Figure 3.5.2: Important environment variables for the website to communicate effectively with OctoPrint

3. GreatFET and Facedancer

4.1 About

https://www.youtube.com/watch?v=h3VWvZ162QE
https://github.com/greatscottgadgets/Facedancer

1.2 How to use

Nobody knows.

4. Website Functionality Guide

5.1 Accounts

Common users have to make an account through the website's registration pages. They must provide: valid FIT email, password [...] Administrators can either have their account created as a superuser while the website is offline, or an existing administrator can use the Admin portal to create their account.

There is currently no verification process implemented for user account creation. Due to the website application being deployed solely within a restricted WLAN, creating an SMTP verification tokens was deemed unnecessary for project objectives.

Think we should couple website functionality guide with User guide? Up in section 1

5.2 File Upload

Authenticated users