

# OCDX File Share System

---

Jared Welch, Geoffery Husser, Elizabeth Bryan, Zach Dolan, Andrew Stoll

Version: 1.0

10/18/16

---

# Final Project Requirements Analysis

---

Github Repository for the project: <https://github.com/jaredwelch1/softwareEngFinalProj.git>  
(<https://github.com/jaredwelch1/softwareEngFinalProj.git>)

## Change log

This will be where we log changes to the requirements as we make them

## Users:

- Researchers
- Students
- Data Scientists
- System Administrators

## Activities:

- Browse Manifest

The user should be able to view all of the manifests listed. This should be a web page feature, drawing from a backend storage device, and displaying the relevant information pulled in. There should be descriptions associated with them, displayed. The interface of this page should be by clicking on corresponding file (button, link, design decided later), and by searching through the files by keyword, user, title, or category.

- Contribute To Database for an existing dataset:

The user should be able to click an option for a specific dataset to be able to contribute to it. The method of contribution, should be a link to the data to store, place to upload scripts, docker config file OR link to generate a file for the configuration.

- Download a Dataset or SNC files:

Displayed data sets and SNC files to the user should have an option available to download said data and files to their local machine.

- Generate or Upload manifest:

A manifest should be generated for a user based on a provided form and must be completed, or a complete manifest can be uploaded directly.

- Save:

Same functionality as contribution to the existing dataset, just offering that option through the interface of Jupyter before they exit the instance.

- Upload New Dataset and Optionally SNC files
- Delete System admin and uploader should be able to delete/remove manifests from system

## **Data and Constraints:**

- Browsing and Searching of manifests:
  - Categories for manifest
  - Tags (?) to enable better searches
  - Title and captions for descriptions of manifest
- User Data:
  - Unique data per each user
  - Username
  - Password
  - Permission Level
  - list of uploads
- Manifests (Uploading, Downloading, Updating):
  - Time of upload, who uploaded
  - Time of any updates, who updated, and what was changed
  - Record of downloads
  - SNC files
  - Dataset associated with that manifest

## **System requirements and Constraints:**

- needs to be a user and browser friendly app/webpage
  - must be available 24 hours
  - User interface must be easy to understand and accessible
  - must have enough stability to support expected volume of users
- server to host the web files/scripts/account information
  - back up server in case of outage
  - must have enough capacity to handle expected volume of manifests

- must be easily upgradeable in case requirements exceed projections
- Database to save json and other pertinent data
  - Must have quick accesses to database and quick lookups to database information
  - Indexing by category, title, keyword, and user (That's a little more design, but something to keep in mind)
- Security for data, both user and manifest data
  - HTTPS webpage security
  - Hashed passwords
  - Backend services to hide database access from outside view

## Design First Pass Ideas

- Tables and data:

User
Username Password Permission Level IDs for manifests by this user

Manifest Search Data
Title contributors SNC search tags related categories

**NOTE: this is a first pass, will need to think more and gather more information in this. For example, it might not be useful to have search tags in the database itself, maybe that is a different type of data**

Manifest Data
Owner User Accesses time of upload time of last change version dataset (json? not sure) SNC files (if present)

## **Pages we have designed and have images of in our repo:**

- Browwing manifests page
- editing manifests page
- viewing individual manifest page
- Log in page (first page you see)
- User account page