Final Project Requirements Analysis

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

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
  + 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 assosciated 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 outtage
* database to save json and other pertinent data
  + Must have quick accesses to database and quick lookups to databse 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