

# OCDX Research Project

Version 1  
17 October, 2016

## **Group 10:**

Kate Watkins  
Andy Hine  
Dewi Kharismawati  
Alex Wilhelm  
Assia Hardiman  
Wei Xian Low

GitHub: [https://github.com/Ahine/SWE\\_Group10](https://github.com/Ahine/SWE_Group10)

Drive: <https://drive.google.com/file/d/0Bzc5cB9bdfSUdXJXa3NvNEFBSDA/view?usp=sharing>

# Table of Contents

● Requirements Analysis .....	2
○ User Requirements .....	2
○ System Requirements .....	2
○ Use Cases .....	3
● System Design .....	11
○ Screen Views .....	11
○ Class List .....	14
○ Table List .....	14
● Change Log. ....	20
● Glossary .....	21

# Requirements Analysis

## Expected Users

We begin by identifying four expected users for this system:

- Students - Read only, not affiliated with the research
- Researchers - Collecting data
- Data Scientists - Analyzing data/make predictions
- System Admins - Maintain the system

## User Requirements

A typical user expects the following from this system:

- To be able to browse through a collection of research data that was compiled and formatted as per the OCDX specification.
- As a researcher, to be able to upload their own collections of data formatted as per the OCDX specification.
- As a researcher or a student, to be able to download collections of data to their local machines in order to further their own research, or for educational value.
- A Student should be able to view the data in a visually appealing format.
- Researches should be able modify data sets.
- Researchers and Data Scientists can make edits to manifests

## System Requirements

- Database:
  - For this project there is a need for a database in which we will use MySQL or SQL
  - Database security is a must, through hash functions and encryption we will protect the data that is uploaded to our system.
- Servers:
  - This product will need to have servers for hosting, and possibly in multiple locations.
  - Backup servers to account for potential failure. Losing all of this data and cross-research could be detrimental.
- Web Service:
  - In this project a web service will be needed, and for this we will use Microsoft Azure, or Amazon Web Service.
- Internet Connection:
  - For a user to use this product, they will need to be connected to the internet (wireless or ethernet).

- Deployment:
  - For building and deployment we will use Microsoft Azure or Amazon Web Service.
- Operating Systems:
  - We we deploy for MAC OS, and Windows. Maybe be able to go mobile?

## Use Cases

For each of the use cases detailed below, one or more of the expected users will take part in an activity relative to the OCDX process.

### **Browse Manifest**

When the user searches by keyword, relevant manifests and corresponding SNC files are displayed with brief descriptions on what the data and files hold. This use case will activate whenever a user chooses to search or browse the manifest available in the website.

#### Functional Requirements and Non-Functional Constraints

- "Date Uploaded" will be differentiating factors for multiple sets of SNC files for the same data set.
- A search entry provided by the user if the user decides to search for a manifest.
- A search on Manifest returns a collection of appropriate results, in user's choice of list or table view.
- Search results can be filtered or sorted alphabetically or by length, keyword etc.

#### Non-Functional Constraints

- Minimum of one manifest available for browsing.
- Search by Manifest returns a maximum of ten results per page.

#### Technical Requirements

- Providing user a list of manifest available.

#### Primary Actors

- Researchers
- Students
- Data Scientists
- System Admins

#### Pre Conditions

- User has provided a search entry or clicked the button to show and browse all manifest.

#### Main Success Scenario

- A list of manifest is listed on the website for the user and the corresponding datasets and manifest pertaining to the manifest shown.

#### Failed End Condition

- No list of manifest is available for the user to browse.

#### Trigger

- User clicks on browse manifest or provided a search entry to the search bar and hit search.

#### Dependent Use Cases

- Search Manifest

### **Upload Data Set**

A researcher with a data set will come to our site to contribute their data set. They will also have the option to upload any SNC files (scripts, notebooks, and config files).

#### Functional Requirements

- Form to upload/link to dataset
- Form to upload scripts (if provided by the user)
- Docker config file (if provided) [could be generated with use case "Generate Config Settings"]
- There is going to be levels of authorizations; admins, researchers/data scientist, and students.
- For admins, there should be special administration functionality, such as who can upload/link data sets, security, maintenance, and monitoring of what is being uploaded, ect.
- An error message needs to be displayed if not uploaded correctly.
- A success message needs to be displayed if uploaded correctly.

#### Non-Functional Constraints

- There needs to be a maximum file size so that a user cannot upload a big file.
- Uploading data set needs to have acceptable performance(good run time), so that it doesn't take seven years to upload a set.
- Placing to upload/link to data set needs to be usable.
- Data sets that are uploaded need to have data integrity.
- Uploads need to have some form security so that data sets do not get corrupted during the upload process.
- There needs to be an error message if upload time runs out.

#### Technical Requirements

- Back end version control for the scripts
- Disk space for storage of data (possibly large data sets)
- Working, public web server

### Actors

- Researchers
- System Admin
- Students
- Data scientists

### Pre Conditions

- One primary actor wants to upload and share data.

### Main Success Scenario

- Datasets and SNC files have been successfully contributed to the site. A successful outcome would be when a data set is uploaded completely and correctly.

### Failed End Condition

- A researcher with a data set will come to our site to contribute their data set. They will also have the option to upload any SNC files (scripts, notebooks, and config files). A non successful outcome would be if during the data set upload, something got corrupted, or failed to upload during the upload process.

### Triggers

- User clicks "Submit Data Set Button on the OCDX.IO website."

## **Contribute to Existing Dataset**

A user wishes to add SNC files for a dataset already uploaded to the server. The user will provide a link to the dataset that will be found on the server using its SHA1 code.

### Functional Requirements

- Place to upload JSON file
- Docker config files (if provided) or link to "Generate Config Settings".
- Files are scanned prior to upload to avoid file corruption or viruses.

### Non-functional Requirements

- Ability to upload up to 3 JSON files at one time.
- Database security ensured by detecting and removing potential SQL injections.
- Limit on file size for large dataset uploads.

### Technical Requirements

- Back end version control for scripts
- Working, public web server.

### Primary Actors

- Researchers
- Students
- Data scientists
- System admins

#### Pre Conditions

- User wished to upload SNC files.
- A request was sent from the save

#### Main Success Scenario

- Dataset has been found on server and SNC files have been successfully uploaded to site and connected to the dataset's manifest.

#### Failed End Condition

- User is unable to upload SNC files or dataset not found on server.

#### Trigger

- User clicks "Contribute to Existing Database"
- User saves

#### Dependent Use Case

- Save

### **Download Info**

When a user has found a dataset or SNC files they are interested in, they can opt to download it onto their local machine. They can choose to download just the dataset or the SNC files with the dataset.

#### Functional Requirements and Non-Functional Constraints

- A button on the page that the user clicks to prompt the system to start the download process.
- A form or pop-up where the user can select whether they want to download only the dataset or the SNC files with the dataset.
- Some type of form/progress bar that keeps the user updated on the status of their download.

#### Non-Functional Constraints

- If the download is proceeding exceptionally slower than expected, the user should be prompted with an option to retry or cancel the download.

#### Technical Requirements

- Copying files onto user's local machine.

#### Primary Actors

- Researchers
- Students
- Data Scientists
- System Admins

#### Pre Conditions

- A user has browsed for and selected a dataset.

#### Main Success Scenario

- A copy of the files is on the user's local machine.

#### Failed End Condition

- The download fails and the user does not have a copy of the files.
  - The user should be notified when this occurs.

#### Trigger

- User clicks "Download" on manifest's browsing page.

#### Dependent Use Cases

- Search Manifest
- Browse Manifest

### **Generate Upload Manifest**

After a user has uploaded the dataset and any SNC (Scripts, Notes & Configuration) files, they will have one of two options. If a complete OCDX Manifest is already available to the user, the file can be uploaded directly. Otherwise, the user will manually fill out a manifest form with the necessary specifications. Once they do this, they will also have the ability to search for the manifest on the web app. Information of the manifest should be described in detail, from basic author information to tags. In both cases, the manifest must be complete before the dataset and SNC files are saved.

#### Functional Requirements

- For this function to work, the users must have already uploaded their datasets and SNC files into the website. From there, the requirements are as follows:
- Buttons to "Generate Manifest" and "Upload Manifest"
- Include input fields for all manifest specifications for "Generate Manifest"
- Dataset and any SNC files has been contributed.

#### Non-functional Constraints

- Maximum file size for uploaded Manifest.
- Only one Manifest is generated or uploaded per dataset.

#### Technical Requirements

- Database for storing manifests and populating entries in the table to catalog events.



### Primary Actors

- Researchers
- Students
- Data scientists
- System admins

### Pre Conditions

- A primary actor has shared data and any SNC files.

### Main Success Scenario

- Manifest, data, and any SNC files are uploaded to server and are saved and searchable.

### Failed End Condition

- User does not have a complete manifest.

### Trigger

- User clicks "Create Manifest" or "Upload Manifest" button on site after submitting dataset.

### Dependent Use Cases

- Upload Data Set

## **Save**

Users will be able to save their work, either created, edited, updated, or manipulated SNC (Scripts, Notes & Configuration) files, before closing the program. If they choose to save it, the user will be directed to "Contribute to Existing Dataset" where they will be able to choose where to save in the database. If they click Don't Save, none of their work will be saved.

### Functional Requirements

- Button to "Save": save by clicking the save button in the program
  - The user will be directed to "Contribute to Existing Dataset" page
- Popup to "Save" if user tries to exit without saving their changes
  - If user decides to click the "Save" button on the popup, they directed to the Contribute to Existing Dataset like the regular save
  - If user choose "Don't Save" exit the program normally
  - If user click cancel, it canceling the popup and the exit action

### Non-functional Constraints

- If the popup does not get any response for 10 minutes, the program will cancel the exit action.

### Technical Requirements

- Redirected to “Contribute to Existing Dataset” to save the changes that made by the users

### Actors

- Researchers
- Students
- Data Scientists
- System Admins

### Pre Conditions

- User has opened data or SNC files in the program
- User has made any changes to original SNC files or has created new SNC files

### Main Success Scenario

- User is able to save their additions and is directed to create a new project linked to an existing dataset.

### Failed End Condition

- The user is not able to save their additions or is not directed to "Contribute to Existing Dataset" when wanting to save.

### Trigger

- Click “Save” button
- Exit the program without saving

### Dependent Use Cases

- Search Manifest
- Browse Manifest

## **Search on Manifest**

A user will choose to search the site's database for datasets and corresponding SNC files using keywords. The manifest will be searched for these keywords.

### Functional Requirements

- Input text field for keywords
- Users have input a search entry or keyword into the search input text field.
- Users have input valid search entry keyword.
- Search keywords on creator, date, content etc.

### Non-Functional Constraints

- Search by one keyword at a time.
- Search results display ten per page.

#### Technical Requirements

- Program to search and select manifest for keywords.

#### Primary Actors

- Researchers
- Students
- Data scientists
- System admins

#### Pre Conditions

- A user wants to search the site for data.
- User is logged in to the web application.

#### Main Success Scenario

- Manifests on the server have been searched for the keyword(s) and SNC files and datasets connected to the manifests with the keywords are displayed to the user.

#### Failed End Condition

- No manifests found with given keywords.

#### Trigger

- User clicks "Search Manifests"

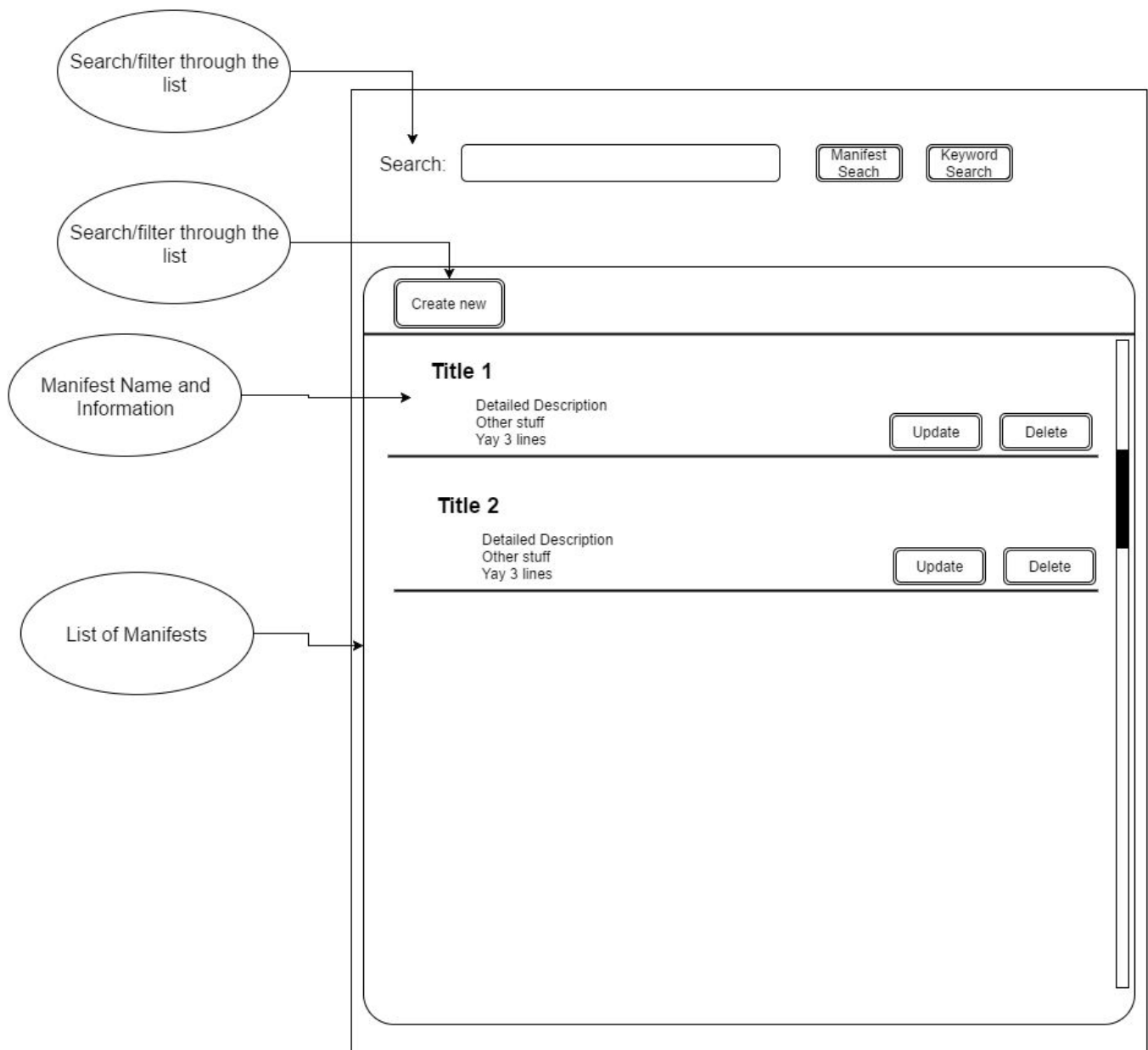
#### Dependent Use Cases

- Upload Data Set
- Generate Upload Manifest
- Contribute to Existing Dataset

# System Design

## Screen Views

### Browse Manifest



## Upload/Create Manifest

Upload a file and it will auto-generate details

**File Picker**  Choose File

**Details**

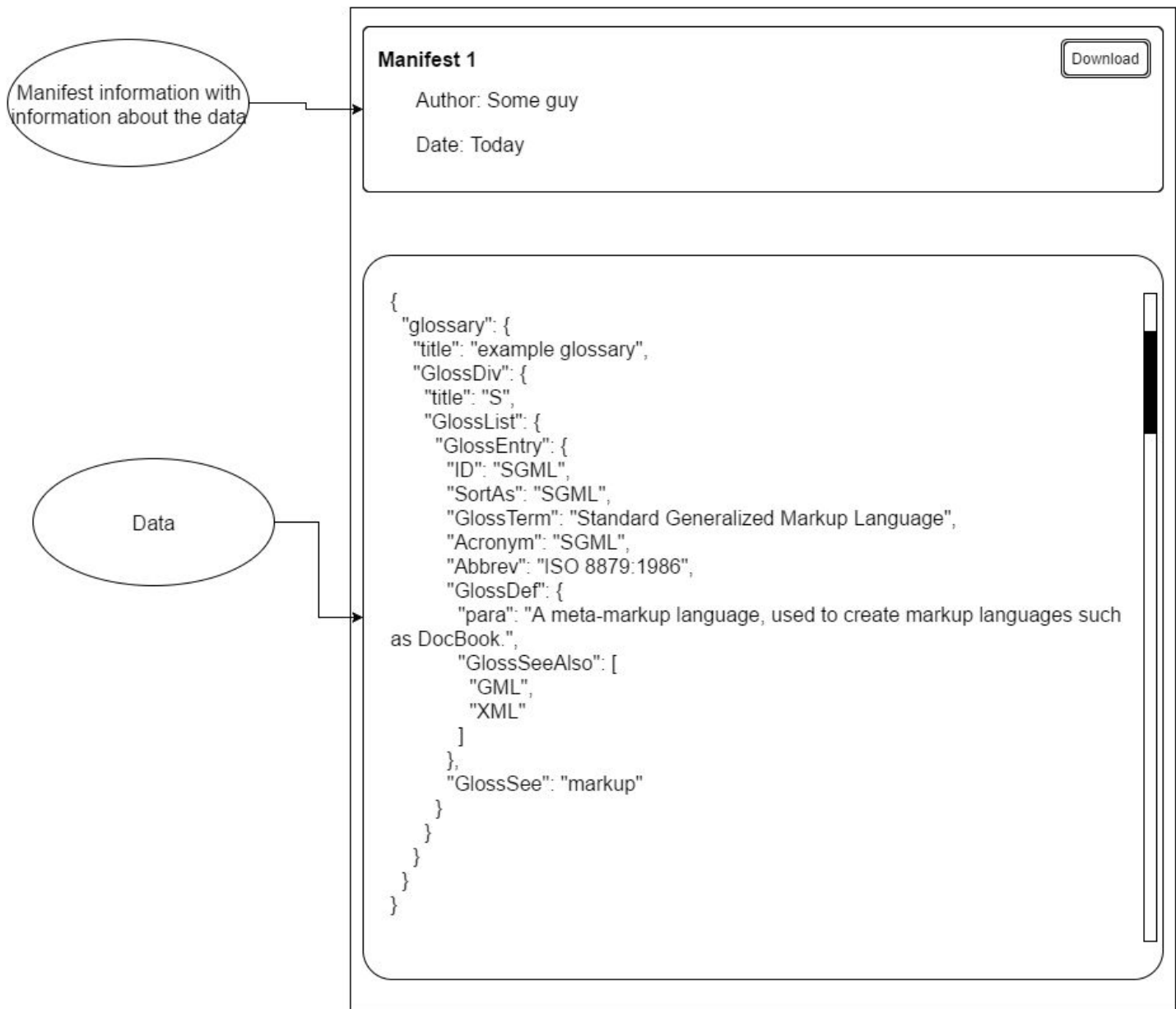
**Manifest Name:** Manifest 1  
Author: Some Guy  
Date: Today

{  
 "glossary": {  
 "title": "example glossary",  
 "GlossDiv": {  
 "title": "S",  
 "GlossList": {  
 "GlossEntry": {  
 "ID": "SGML",  
 "SortAs": "SGML",  
 "GlossTerm": "Standard Generalized Markup Language",  
 "Acronym": "SGML",  
 "Abbrev": "ISO 8879:1986",  
 "GlossDef": {  
 "para": "A meta-markup language, used to create markup languages  
such  
as DocBook.",  
 "GlossSeeAlso": [  
 "GML",  
 "XML"  
 ],  
 },  
 "GlossSee": "markup"  
 }  
 }  
 }  
 }  
}

This field will auto generate, but still be editable

The user will enter this data or have it auto-loaded by the file picker

# View Manifest



## Class List

- Users
  - Users()
    - This function will have a list of users. This list of users will be grabbed from a database. This function can also grab data and details about the user, such as if the user is a researcher, student, data scientist, or system admin.
  - userLevel()
    - There could another function under class user which handles different levels of administration privileges.
  - login()
    - There will also be a function that will check user login information, and handle the situation where a user needs to register an account.
- Manifest upload
  - uploadManifest()
    - There will be a function just for manifest upload, which will include an upload button that will be an onClick event. This function will grab manifest data.
  - displayDataUpload()
    - After fetching data, there is going to be another function that displays the data that was just uploaded. Also, this is where the functionality of displaying and error or success message.
- Manifest download
  - downloadManifest()
    - This function will use the database, and download new data to the database. There will be a button that will be onClick event.
  - displayDataDownload()
    - After downloading data, there is going to be another function that displays the data just downloaded. Also, this is where the functionality of displaying and error or success message.
- Manifest search
  - search()
    - This function will use the database. There will be a text field that allows a user to search through the site's database for datasets and corresponding SNC files using keywords. The manifest will be searched for these keywords. The function will take these keywords or phrases and respond with relatable/significant data.
  - displaySearch()
    - This function simply just displays searched keywords/phrases.

- Contribute to existing
  - contribToExisting()
    - A user wishes to add SNC files for a dataset already uploaded to the server. The user will provide a link to the dataset that will be found on the server using its SHA1 code. This code will require talking to the database of files. This function will also talk to the users part of the database. This is because, certain users will only have certain rights to edit certain material. For example, researchers can upload and edit their own work, but can't edit another researcher's work. Researches may have the option to request edits on other researchers work. Students would not have any editing powers, and system admins will have all editing powers.

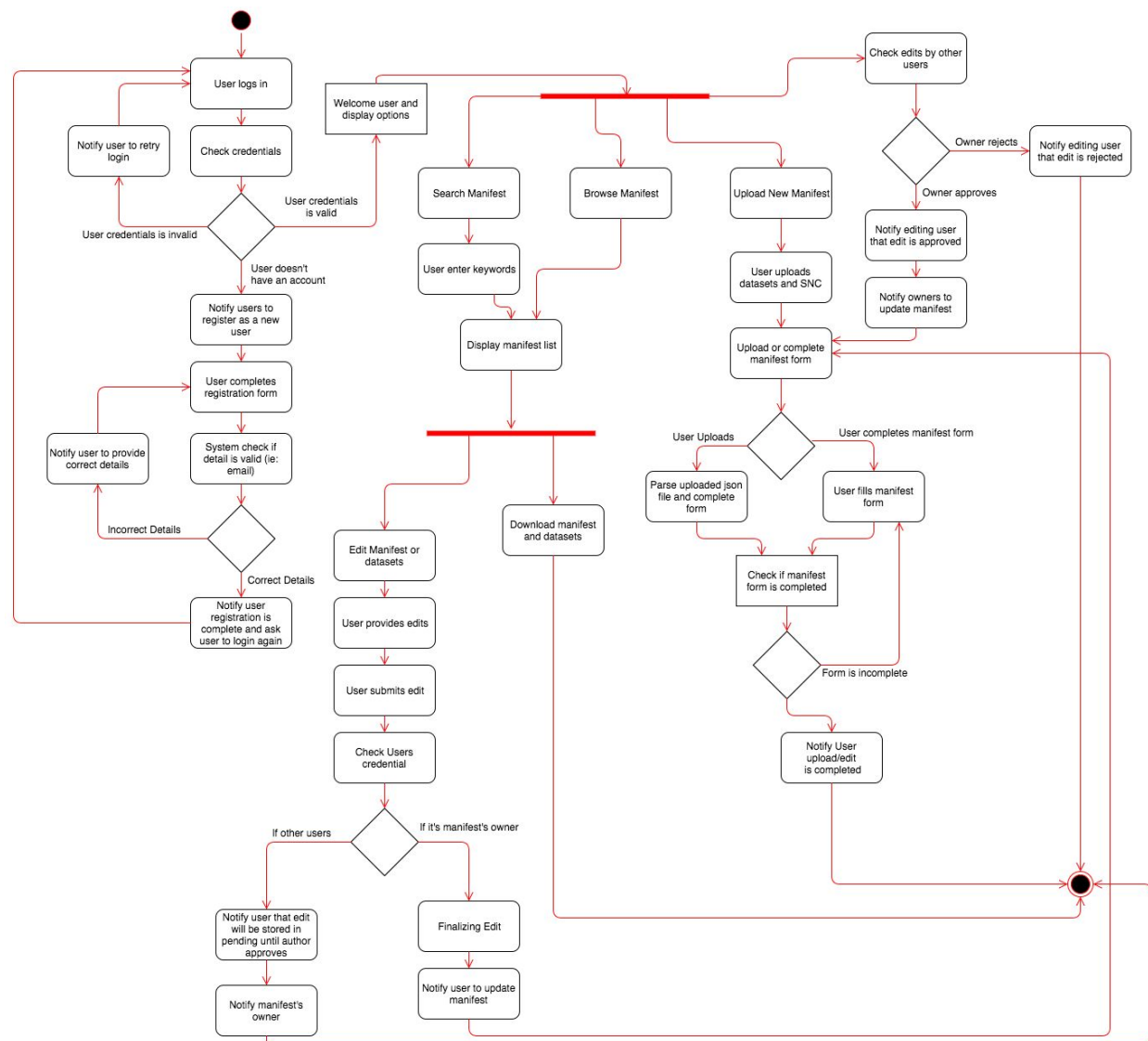
## Table List

- Manifest
  - abstract:string, title:string, arovenance:string, bibliogrpahicCiations:string
- Creator
  - name:string, email:string
- Date
  - datasetTimeInterval:time
  - dateRetrievedTimeInterval:time
  - dateCreated:datetime
- Distribution
  - Uri:string
  - Comment:string
- File
  - Name:string
  - Format:string
  - Date->fileTimeInterval:time
  - Date->dateRetrievedInterval:time
  - Date->dateCreated:datetime
  - Abstract:string
  - Size:int
  - Uri:string
  - Checksum:string
  - Permission:string
- Manifest Relations: Every manifest has one or more than one creator, one date, one distribution, and one or more than one file.

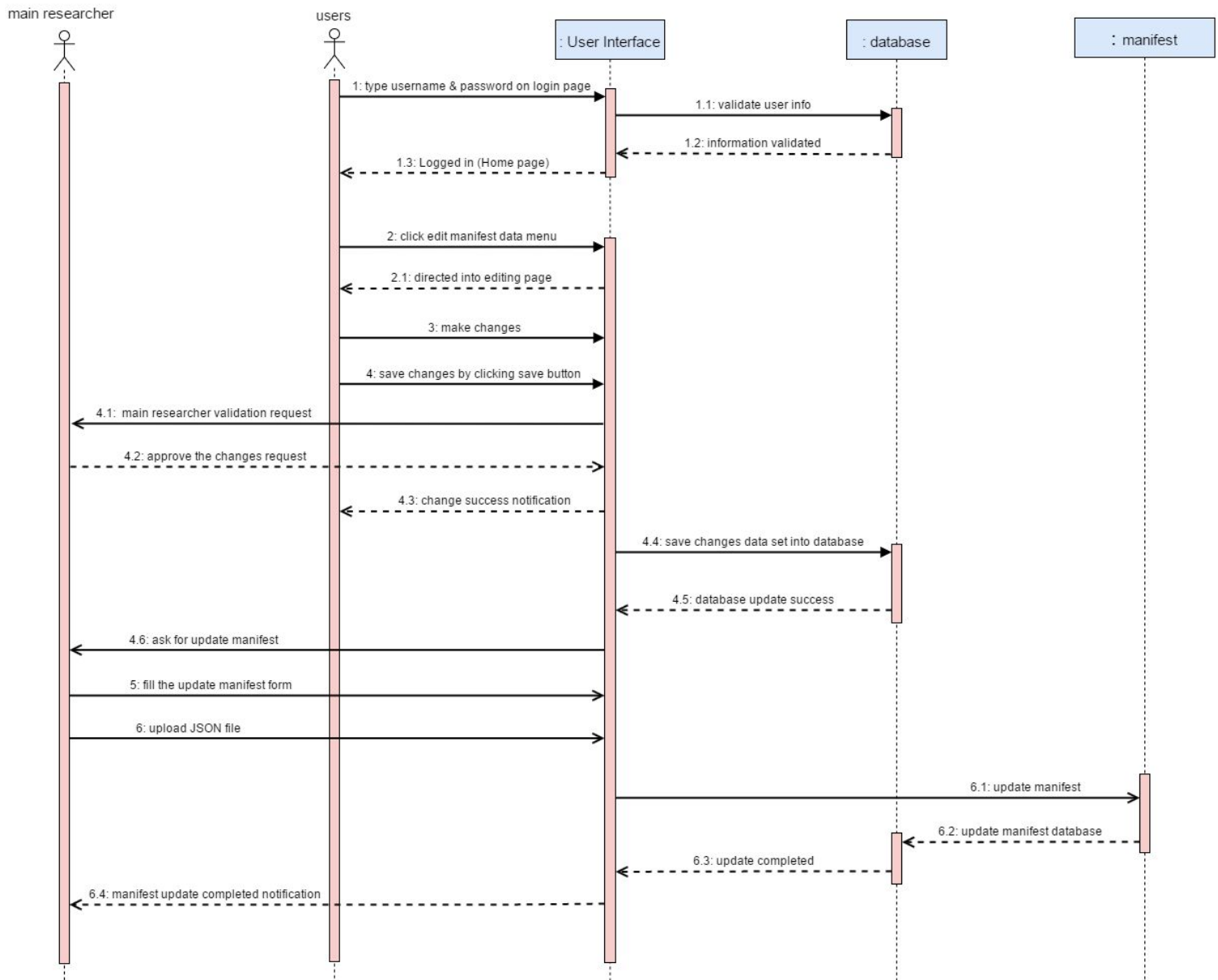


- User
  - userID:string, password:string, permissionLevel:int
- Researcher
  - Manifest:manifest
  - PermissionLevel:int = 2 (level two permission - edit and upload access)
  - Relation: Researcher is a User (one to one), researcher will have one or more than one manifests attributed to him/her.
- Student
  - permissionLevel:int = 1 (read and download access only)
  - Relation: Student is a User (one to one)
- SysAdmin
  - permissionLevel:int = 3 (full access to front and back-end)
  - Relation: SysAdmin is a User(one to one)

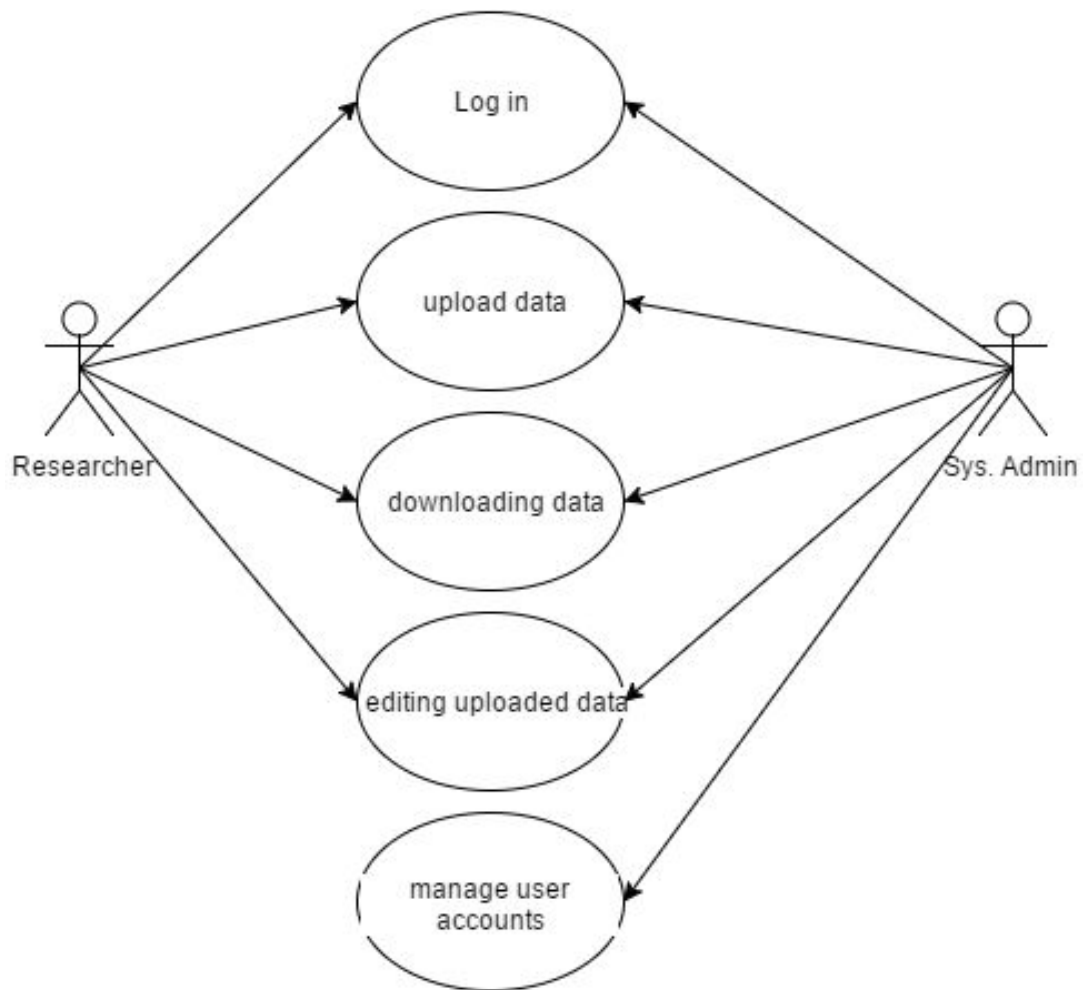
## General System Wide Activity Diagram



## Sequence Diagram: Edit a Dataset



## Use Case Diagram: Working with Datasets



# Change Log

10/18/2016:

Individual work compiled into Version I.

Contributions:

- Alex - First pass screens, Contribute to Existing Dataset use case analysis
- Andy - Download Info use case analysis, first pass table list
- Assia - Generate Upload Manifest use case analysis, use case diagram first pass
- Dewi - Save use case analysis, sequence diagram first pass
- Kate - Upload Dataset use case analysis, class list first pass
- Wei Xian - Browse on Manifest, Search on Manifest use case analysis, full activity diagram first pass

# Glossary

<b>OCDX</b>	: Open Community Data Exchange
<b>SNC</b>	: Scripts, Notes, Configurations
<b>DBMS</b>	: Database Management System
<b>AWS</b>	: Amazon Web Services (A web hosting services provider)
<b>Azure</b>	: Microsoft Azure (Web Hosting Services Provider)
<b>mySQL</b>	: Open Sources Relational Database Management System
<b>SQL</b>	: Structured Query Language
<b>Mac OS</b>	: Apple's Operating System
<b>Windows</b>	: Microsoft Operating System
<b>PHP</b>	: Server Side Scripting Language
<b>AJAX</b>	: Asynchronous Javascript and XML
<b>JSON</b>	: Javascript Object Notation
<b>UML</b>	: Unified Modelling Language
<b>Manifest</b>	: A type of inventory/catalog detailing the information related to the datasets and SNC uploaded by a user.
<b>Datasets</b>	: Information and resources pertaining to the research done by a researcher.
<b>ERD</b>	: Entity Relationship Diagram
<b>Use Cases</b>	: List of actions and event, detailing interaction between an actor and a system to achieve a goal.
<b>GitHub</b>	: Web based git repository hosting services
<b>git</b>	: Version Control Software
<b>HTML5</b>	: Hypertext Markup Language v5
<b>Database</b>	: A structured set of data held digitally.
<b>Wiki</b>	: A website that provides collaborative modification of it's content.