CS 4320 Final Project: Requirements Analysis & Software Design

OCDX Engine

GitHub Repository

Deployment Environment

Team 4 Members:

- Holt Skinner *Project Manager*
- Justin Hofer
- Ali Raza
- Bo Zhang
- Bradley Rogers
- Pramod Pudotha

Table of Contents

Requirements Analysis

User Descriptions

Rogers/Pudotha

- **Data Scientist**
  - Uploads and downloads data
  - Conducts sophisticated and systematic analyses of data
  - Extracts insights from large data sets
- **System Administrator**
  - Review the uploaded data
  - Manage users

Use Cases

Skinner/Hofer

- A Data Scientist can Browse by keyword for Manifest
- A Data Scientist can Search on Manifest
- A Data Scientist can Contribute to Existing Dataset
- A Data Scientist can Download Info
- A Data Scientist can Generate or Upload Manifest
- A Data Scientist can Save
- A Data Scientist can Upload Data Set

Functional Requirements

Pudotha/Skinner

- The system will take an inputted manifest and place it into storage.
- The system will retrieve a manifest from storage and present it to a Data Scientist.
- The user interface can accept and serve search, upload, and update requests to and from Data Scientists.
- The system will process search, upload, and update requests, both into and out of Database Layer.

Non-Functional Requirements

Hofer/Raza

User Non-functional

- Data Scientists will be able to perform efficient searches based on keywords. Users should not wait more than about two seconds for a query to resolve.
- Data Scientists will be able to upload and update manifests.
- Data Scientists will be able to upload and download data files.

System Requirements

Raza/Pudotha/Hofer

- **Space Requirements**
  - The system must have enough physical space to handle extremely large data-sets.
  - The system may need to be a distributed system using clusters for efficiency.
- **Reliability Requirements**
  - If the system crashes, the data and any manipulations/analytics must be preserved.
- **Privacy Requirements**
  - The data access must be limited to authorized users only.
  - The system must enable secure data transfer over the internet
- **Web Server**
  - Web Server must provide reliable service for the appropriate amount of traffic that will be sent and received from the system.
- **Database Backend**
  - Database must be able to convert data into easily storable format, and return in original format.
- **Storage Array**
  - System must create intermediate backups and update logging to revert to earlier states if needed

User Requirements

Zhang/Rogers

- Data scientists will be able to upload Manifest.
- Manifest can be reviewed by other data scientists.
- Data scientists can edit or delete their Manifest.
- Data scientists can search for Manifest they wish to view or test.

- Data scientists will be able to include special comments or suggestions on manifest
- Data scientists should be able to notify the changes or suggestions that improves manifest to other users
- System admins can ban an illegal data scientist
- System admins can delete an illegal manifest

Software Design

Sketches

Skinner/Rogers

- User can Browse by keyword for Manifest
- User can Contribute to Existing Dataset
- User can Download Info
- User can Search on Manifest

Browse Screen

Dataset Information

- User can Generate or Upload Manifest
- User can Save
- User can Upload Data Set

Upload Screen

Data

Zhang/Hofer

- **Members**
    - ID
    - name
- **OCDX_manifest**
    - author

    - date
    - size
    - category
    - version
    - modify_request

- **Users**

    - ID
    - name
    - email
    - phone

- **Login_attempts**

    - userID
    - date
    - attempts

Database Structure

Database Diagram

Deployment Enviornment

Raza

- Hosting Platform: Amazon Web Services
- Operating System: Ubuntu 16.04.1
- Web Server: Apache HTTP Server 2.4
- Database: MongoDB

Testing

Zhang/Hofer

1. Build user acceptance test scenarios for documented requirements on separate Wiki page, linked to all sprints.
2. Build unit test scenarios
3. Describe regression testing and your regression testing plan.
4. Describe how your team will perform integration testing. What needs to be integrated? When?
5. Describe which tests are for verification and which tests are for validation

User Acceptance Test (UAT) Scenarios

- The data scientists can search for manifests with keywords, the app returns the results or shows the reasons why no result is found. The data scientists can upload or download manifests, if it fails the reasons are also reported.
- The system administrator can ban a data scientist when improper behaviors are frequently observed. The system admin will also leave the reason for this action, and the comment will be sent to the user. The system admin can also make change to a manifest when it is necessary, the reason is also given to the author.

Unit Test Scenarios

Login

- If the user exists, the action continues.
- If the user does not exist, further action is denied and the error message is shown.

Verify File

- If the file size and type is an acceptable format, the action is allowed.
- If the file size is too big or the type is illegal, the action is denied.

Search for Manifest

- The system finds the record from database by matching the keywords.
- The system returns an error message if no record is found.

Upload

- The file is stored in the database.

Download

- The required file is pulled from database and presented to the user.

Edit/Delete Manifest

- The manifest is changed and database is updated.

Regression Testing

- Regression testing is a type of software testing that verifies that software previously developed and tested still performs correctly even after it was changed or interfaced with other software.
- In this system, a set of unit tests are prepared to cover all the functions of the software. The tests are run after every update or bug fixing.

Integration Testing

- At the end of each sprint, we will perform integration testing

Data Scientist Uploads Manifest

- If the user has not signed in, the action fails and the message "login first" shows on screen.
- If the user signed in, but the file is not acceptable (file is too big or type is illegal), the action fails and the reminding message shows.
- If the user signed in, and the file is valid, the action succeeds and the success message shows.

Data Scientists Review Manifest

- If the manifest is still valid, it is extracted from database and shows to the user.
- If the manifest is no longer valid, the action fails and the message "manifest does not exist" shows.

Data Scientists Search for Manifest

- If the keyword matches any records in the database, they are shown to the user.
- If the keyword cannot match any record in the database, the error message is shown to the user.

Data Scientists Notify Changes to Other Users

- If the user to be notified still exists, a notification is sent to the user, and it shows notification success.
- If the user no longer exist, the message shows user not exist.

System Admin Bans an Illegal Data Scientist

- The data scientist account is transferred to the "banned" group. A notification is sent to the user as well as the reason for the ban.

System Admin Deletes an Illegal Manifest

- The manifest is deleted (moved to "trash" group), and a notification and reason are sent to its author.

Summary:

- The unit testing, integration testing and regression testing are for verification, the user acceptance test is for validation.

Change Log

- Version 1.0: Pre-Implementation Design *10-16-2016*
- Version 1.1: Sprint 1 *10-31-2016*