SENG 371 Lab Documentation

Ross Alexandra - V00867152
Ethan Kwan - V00
Josh McIntosh - V00875715
Austin Smith - V00867501

Scrum Master - Ross Alexandra

User Interface

Week 1:

Design UI mockups using Adobe XD. Key components include:

- Create Account
- Login Page
- Home Page
    - Standard account management functionality
    - Create job link
    - View current job link
- Create Job
    - Upload files
    - Link datasets
- Running-Jobs List
- Job Details

Week 2:

- Develop skeleton web application using Django for backend
    - Should be navigable but with no production content
- Develop HTML templates using WIX for front-end. Specific pages this week include:
    - Create Account
    - Login
    - Home page

Week 3:

- Develop HTML templates using WIX for front-end. Specific pages this week include:
    - Create Job
    - Running-Jobs List
    - Job Details

Week 4:

- UI testing and validation

Business/Logic Layer

- When user clicks UI button to start job server should be informed
- Upon start job signal, server should run the “retrieve dataset” algorithm from Data Access Layer.
- Once data is retrieved, partition the data into n sets
- Spin up n containers
- Feed the data of each partition into the spun up containers
- Once the data has been processed in each container, retrieve that data.
- Run the “pushing output data” algorithm from Data Access Layer to send
- Send a signal back to the webapp that the job has finished.

Data Access Layer

Week 1:

- Create database skeleton using Django and PostgreSQL for client info/running jobs
- Develop prototype algorithm for retrieving dataset from cloud storage
    - Possibly will be executed using Azure functions

Week 2:

- Finalize database for to facilitate the production version of the UI pages/running jobs
- Develop prototype algorithm for pushing output data back to cloud
    - Possibly will be executed using Azure functions

Week 3:

- Develop production version of dataset retrieval and push to cloud algorithms

Week 4:

- Testing and verification of functionality via UI

Project Tools

For this project, there are two subsets of tools being used, development tools, and CI/CD tools.

Development tools

This project is implemented in Python using a Django backend. While developing, due to the range of experience and code styles of the people on this project, we have setup a few tools to manage the code. These tools are Black (a code formatter so that all the code follows a consistent style) and iSort (an import linter for Python.) If these tools are not run on the codebase before a push, then the build for that push will fail.

In order to reduce failing builds, another tool called “pre-commit” is used. Pre-commit will automatically run all these tools *before* the commit is made. This means that even if a developer forgets to run the tools, they will still be run.

In order to manage this system’s development, GitHub projects are being used along side tickets to create a rudimentary backlog and sprint setup.

CI/CD tools

This project uses Travis-CI as a build manager. Travis automatically builds each commit that is pushed to the GitHub repo as it is pushed. Further, this project is built on a Heroku server. Whenever a commit is pushed to master, after the build has completed, a docker image is pushed to DockerHub, and finally Heroku pulls that Docker image, and starts it.