

Progress Report Final - Group 63

27-03-2019

Ross Alexandra - V00867152

Ethan Kwan - V00862439

Josh McIntosh - V00875715

Austin Smith - V00867501

Risk Analysis

No.	Risk Description	Mitigations
1.	Incomplete and unpolished UI	Will submit a functional version of the UI without beautification.
2.	Incomplete unit test cases	Will write unit tests to complete code coverage, with other tests marked as TODO. If this still isn't enough time, reduce code coverage goals to approximately 60-70% and prioritize essential functions.
3.	Failing to initialize children docker containers from within the master container on Heroku	Using threads within the main script might get around the necessity of having to initialize more containers. It is believed this should scale relatively well, but would require an analysis to determine the feasibility of this solution.
4.	Storing aggregated result from algorithm	Our current solution for data retrieval is from a public read-only AWS data store. If we want to return data or results to a similar store we will need to create our own data store on our Heroku server to facilitate the cloud storage side. Due to size limitations on our account, we also may store the files locally. While this isn't representative of a real system, it is within our budget of 0\$.
5.	Proper data analysis algorithm	Creating a data analysis algorithm is difficult without the proper knowledge. Rather than have a realistic algorithm, we will instead try to use a simpler image modification to act as a representative for the real algorithms.
6.	Pulling user data from our own server	To have full control over the data, we want the ability to create and manage our own data set. While we have STAC data, we would need some sort of flask app to handle requests for the data. We can do without this, but it would be more representative.

Issues

These are a list of issues that underwent some development or are had a pull request some time in the last week.

No.	Open Date	Description	Resolution
1	Mar 1	Create home page	<ul style="list-style-type: none">• A home page is created which is the landing page after a login.• The homepage links to a separate page which allows the user to change their password. Once their password is changed, they are redirected to this homepage.
2	Mar 1	Link "create job" page to homepage	<ul style="list-style-type: none">• A link is added to the homepage allowing the user to navigate to the Create Job page.
3	Mar 1	Create Jobs Page	<ul style="list-style-type: none">• A Jobs page has been created• The Jobs page has a list of the currently running jobs, with an indication saying that they are running• The Jobs page has a list of the finished jobs, with an indication saying that they are finished• The Jobs page displays only for the user that made those jobs.
4	Mar 1	Create Job Details Page	<ul style="list-style-type: none">• For each job stored in the database, a page exists which can be navigated to by the job owner.• This job will have the following information on the job: Completed/Running, Dockerfile, and Dataset link.
5	Mar 1	Jobs table is updated when a job is created	<ul style="list-style-type: none">• When a job is created by a user, a new row is created in the table.• A user should be linked to each job (new value in the row.)
6	Mar 8	Fetch user data from data repository	<ul style="list-style-type: none">• The system can pull data from a data-store that has been navigated to.
7	Mar 8	Return computed data to user	<ul style="list-style-type: none">• A user's data-store is updated with their processed data once all of their containers have completed, and the data aggregated.

8	Mar 8	Aggregation algorithm for results	<ul style="list-style-type: none"> Once all containers have finished their run, all the processed data must be combined for later use.
9	Mar 8	Return data to master container	<ul style="list-style-type: none"> Once a user container has completed its run, the processed data is returned to the master container for later use.
10	Mar 17	Partition input data	<ul style="list-style-type: none"> 2-4 even(ish) partitions are created each partition is a crawlable static/active STAC catalog All partitions should be reference only. ie no data is cloned
11	Mar 17	Initialize children docker containers	<ul style="list-style-type: none"> Docker containers are successfully created with test algorithm Partitioned data is used properly a input in algorithm Containers all have proper config/libraries etc.
12	Mar 17	Create Prototype Data Algorithm	<ul style="list-style-type: none"> function successfully executes if data is produced, it is temporarily stored before aggregation

Meeting Minutes

Meeting Date: March 30, 2018

Venue: Online

Duration: 1h 17m

Agenda:

- Work Division
- Status of current front-end systems
- Status of current back-end systems
- Redefining Scope
- Demo overview

Highlights:

- Scope will likely need to be reduced
- Front-end is largely complete
- Back-end is the current section that needs to most work (where the scope reduction might happen)

Action Points	Discussion
<p>Work Division: how are we planning on subdividing the remaining issues?</p> <p>Proposed by Ross.</p>	<p>Assigned role of Product Owner/DevOps to Ross. This entails a workload more focused on team/project management and DevOps than feature development.</p> <p>Assigned Josh to front end development.</p> <p>Assigned Ethan and Austin to focus on backend development.</p>
<p>Status of current front-end: what is done and what is left to do?</p> <p>Proposed by Josh.</p>	<p>Done: all the pages are done or in development</p> <p>Todo: add linking between pages and signal job is created. Assigned to Josh.</p>
<p>Status of current back-end: what is done and what is left to do:</p> <p>Proposed by Ethan.</p>	<p>Done: Have sample data in STAC format</p> <p>Implement the prototyped data fetching Assigned to Austin .</p> <p>Create and implement placeholder processing algorithm Assigned to Ethan.</p>
<p>Redefining Scope: Where can we reduce complexity to meet the deadline:</p> <p>Proposed by Ethan</p>	<p>Examine the feasibility of initializing Docker within Docker to create children.. Will be determined if there is a need to cut closer to the deadline.</p> <p>Assigned to Ross</p>
<p>Demo Overview: What do we want to do for the demo in lab?</p> <p>Proposed by Austin.</p>	<p>Demo will likely be carried out by Austin and Ross (exams studying abiding.) Will likely walk through the web app and then run one instance of the algorithm. Likely will need to add print statements or built in pause to demonstrate the partitioning algorithm and docker container creation.</p>