Team 21 - Sprint2 Planning Document
Transforming Drainage Project

Alexis Williams, Brandon Kent, Jacob Conley, Jordan Hagedorn

Sprint Overview

For the second sprint, we intend to focus on user stories that are a continuation of the first user stories. Over the course of the first sprint, there were a couple issues that were brought up concerning the implementation that was used. One of those issues concerns the size of files that are transferred to the client; this needs to be addressed. On the back end, we will work on improving the documentation of the code, making a way to compress large files to minimize network usage, and making algorithms to have data and maps downloadable. On the front end, we will work on making cosmetic changes and adding functionality to current charts. Also, we will work on being able to receive compressed data files, uncompressing them and successfully parsing them, as well as successfully receiving downloadable data files and pdfs.

Scrum Master: Jordan Hagedorn

Meeting Schedule: Tuesday/Thursday 6pm-8pm

Risks/Challenges: While we have made great progress in our understanding of JavaScript, Node.js, and HTML, we are still fairly new to these languages. This will continue to be a challenge for us as we work further into the project. In addition, despite us having a much better grasp on the previous team's code, there are still areas that we have not yet interacted with that we suspect will be challenging to work with.

Current Sprint Detail:

User Story #1
As a user, I would like to be able to download the results of the mass data calculation and comparison year data and graphs

#	Task Description	Estimated Time	Owner
1.	Add code to make mass data calculation results downloadable	3 hours	Jacob
2.	Add code to make comparison year data downloadable	2 hours	Jacob
3.	Add option to website to download mass data calculation	2 hours	Jacob
4.	Add option to website to download comparison year data	2 hour	Jacob
5.	Add code to make resulting maps downloadable	3 hours	Jacob
6.	Add option to website to download maps	2 hour	Jordan
7.	Unit test - assert that the data received is not corrupted and still parse-able	1 hour	Jacob

Acceptance Criteria

- Given that the code for making the data downloadable is complete, when the code is run, it will return the data in a downloadable form
- Given that the code for making the maps downloadable is complete, when the code is run, it will return the maps in a downloadable .pdf form
- Given that the code for the front end is complete, when the user sees the results, they should also see options to download the results

User Story #2
As a project owner, I would like to be able to easily and efficiently update the files that contain mapping data.

#	Task Description	Estimated Time	Owner
1.	Implement an alert to notify that the	2 hours	Jordan

	mapping data being used is not the newest from the database		
2.	Write a script for automated or manual updating of mapping data	3 hours	Jordan
3.	Write an error checker to ensure the integrity of the mapping data file structures	2 hours	Alexis
4.	Implement a loading bar and real-time feedback for the administrators to check progress of the update script	3 hours	Jordan
5.	Unit tests - run scripts on small datasets and assert correctness	1 hour	Jordan

Acceptance Criteria

- Given that a notification is implemented, when a project owner or server administrator updates the MySql database, they will be notified that the maps are utilizing old data

- Given that an update script is created, a project owner can easily update all of the map files that depend on pre-calculated data

- Given that a progress monitor is implemented, a project owner can easily see how far the file update has gone, and whether or not there was a crash

User Story #3

As a project owner, I would like all the code and data structures to be properly documented and optimized for reusability.

#	Task Description	Estimated Time	Owner
1.	Add comments to our algorithms explaining their implementation and purpose	2 hours	Alexis
2.	Add comments to previous group's code explaining its purpose and improving overall readability	3 hours	Alexis
3.	Make a word document to explain in more detail the overall designs of each main segment of code, the main data structures, and their interactions with each other	4 hours	Alexis

4.	“Unit Test” - check to make sure that an extra document is present in the github	0.5 hour	Alexis

Acceptance Criteria

- Given that the code is properly commented, when another user looks at the code they should be able to follow the code easier

- Given that the word document is made, when another user looks at the document, they should be able to gain a better understanding of how segments of the code work and interact

- Given that the word document is made, when the owner navigates to the documents section of the github, they will see a new document explaining the code in more detail

User Story #4

As a project owner, I want to be able to reduce the amount of data that must be transferred to the user.

#	Task Description	Estimated Time	Owner
1.	Compress data json files - find the most efficient compression method for our use	2 hour	Alexis
2.	Send compressed files instead of full files json files to the client	2 hours	Alexis
3.	Uncompress and parse the json file when it is received	6 hours	Jacob
4.	Unit Test - Ensure compression is compressing each file to the determined percentage. Ensure the file is exactly the same as the pre-compressed file	1 hour	Jordan

Acceptance Criteria

- Given a suitable compression method is selected, the files are a certain percentage smaller than their uncompressed version.

- Given a compressed json file the client cleanly un-compresses the file and parses the data to the geojson maps.

- Given that the compression is successful, the total amount of data passed through the network is significantly less.

User Story #5
As a user, I would like to have a responsive and intuitive experience while using the application

#	Task Description	Estimated Time	Owner
1.	Make maps and their sizes more visually appealing and give positioning options	4 hours	Brandon
2.	Enhance the flow for users interacting with map	2 hours	Brandon
3.	Enhance the flow for users inputting and viewing data results	2 hours	Jacob
4.	Add color switch option for colorblind users	1 hour	Jacob
5.	Statistically analyze data to be able to generate a more revealing and sophisticated visualization	3 hours	Brandon
6.	Unit test - Ensure that components are marked as visible at the correct times	1 hour	Jacob

Acceptance Criteria

- Given that the maps are adjusted to be visually pleasing, a user can view them in a preferred layout such that it can be easily viewed while in a smaller window
- Given that the overall GUI execution flow is enhanced, the user's experience will be much less confusing while navigating the application
- Given that colors can be adjusted, users with visual imparities can better interpret the graphs

User Story #6
As a user, I would like to view the map data in a graph.

#	Task Description	Estimated Time	Owner
1.	Create algorithms for computing data such as max/min, median, etc	4 hours	Alexis
2.	Add button to display map below the graph.	2 hours	Alexis

3.	Parse data into the graph from a json or other format	2 hours	Jordan
4.	Implement options to show and hide values on the graph	3 hours	Jordan
5.	Unit test - check algorithm correctness, assert that correct graph is displayed	1 hour	Brandon

Acceptance Criteria

- Given that algorithms are implemented, a user has access the map's data in graph form
- Given that new graph styles are added, a user will have more options for seeing data
- Given that options to show and hide data are implemented, a user can view a less cluttered graph

User Story #7

As a stakeholder, I would like to improve the functionality of the maps on the regional map tab.

#	Task Description	Estimated Time	Owner
1.	Calculate data for two new results on the regional results map	2 hours	Brandon
2.	Parse data and change cell colors for the two new results	2 hours	Brandon
3.	Display the graphs in the regional results tab	3 hours	Brandon
4.	Change results based on button selection	2 hours	Brandon
5.	Add year selection to regional results map	1 hour	Brandon
6.	Create data files for each year, or one file that contains all years	4 hours	Jordan
7.	Parse data for individual years in map	3 hours	Jordan
8.	Unit Test-Are the graphs displayed in metrics, Are the results correct with the metric units, can the user input in metric.	1 hour	Jordan

Acceptance Criteria

- Given the graph is chosen, the data is displayed quickly in the graph for the user.
- Given the graph is selected, the data displayed is determined by the selection buttons.
- Given the graphs are able to display results, the user should be able to analyze the graphs with their data.

Backlog:

As the project owner, I would like to:

1. Have user uploaded data categorized/named properly
2. Be able to analyze user uploaded data
3. Have a way to determine uploaded data quality

As a user, I would like to:

1. Have more options available when downloaded computed data
2. Have an option to save my personal data to the website database for later use/for other people to use
3. Have recent and historical climate data incorporated into the algorithms
4. Be able to specify crop root zone depth instead of only using the constant values currently provided
5. Be able to see the cost and yields impacts of irrigation planning
6. Have an option to view and use other people's data from the website
7. Be able to assess the quality of other people's data