

# Sprint 2 Planning Document

## RESERVOIR PLANNING TOOL

By The Transforming Drainage Project



### **Sprint Overview**

During this sprint, we intend to begin piecing together our individual components that we created last sprint. After this sprint we will have a client that will display multiple graphs all of which should be accurate by the project owner's standards. Along with this we will be adding the CSV file upload feature during this sprint and integrate its contents into the Transforming Drainage Project algorithm. Lastly during this sprint we will be deploying our application on Amazon Web Services.

### **Meetings**

- **Tuesdays at 4:30-6:30**
- **Scrum master - Drew Atkinson**

### **Risks and Challenges**

Some of the biggest challenges that stand in our way for this sprint is learning to properly test the components we have worked on thus far, as well as continuing to link all of our components together to our platform. During this sprint we will be staying in close contact with our project owners so that we can be sure all criteria for the project are being met, and to avoid any confusion that may cause us to waste time. Another challenge our team faces is deploying our application on AWS. This is something we are not familiar with so there may be problems that arise during this process. Lastly, we will also have to maintain our strict management routines as we did in sprint 1 in order to test and deliver all selected user stories.

### **Sprint Detail**

### User Story #1

As a user, I would like to upload my own data to use

Task	Owner	Time
Create UI for uploading CSV files.	Jonah	2 hours
Post request with CSV	Jonah	5 hours
Direct CSV through parsing and into TDP Alg	Jonah	5 hours
Handle CSV inputs and substitute their contents with the database's.	Clayton	4 hours
<b>Acceptance Criteria</b>		
A user can make UI selections to be able to input their own CSV data.		
Add CSV to POST request and send to backend		
Once the CSV is on the backend, verify it, supplement it with TDP's data, and send it to TDP's Algorithm		

### User Story #2

As a user, I would like to download the results of the calculated data

Task	Owner	Time
Assemble the output of the algorithm into the CSV file.	Drew	10
Add input values if the user uploaded their own file	Drew	5
<b>Acceptance Criteria</b>		
Given the output of the algorithm, a CSV file is created with the daily output of the algorithm.		
Given this CSV file, a user can choose to download it along with the graphical output		
If the user uploaded their own input CSV, the output CSV will be added to it and passed back to the user.		

### User Story #3

As a user, I would like the project to have as few bugs as possible

Task	Owner	Time
Learn node testing framework	Jonah	1 hour
Learn python testing framework	Jonah	1 hour
Test DB setup, update, and remove	Jonah	8 hours
Test node DB exports	Jonah	4 hours
Test UserParse.js	Jonah	4 hours
Reassess potential bugs pointed out by project owners in TDP algorithm	Clayton	2 hours
<b>Acceptance Criteria</b>		
Functional and unit testing added for DB setup.		
Functional and unit testing added for node DB exports.		
Functional and unit testing added for UserParse		

### User Story #4

As a project administrator, I would like the database to be hosted on AWS

Task	Owner	Time
Create a MySQL server on an AWS instance	Drew	3 hours
Initialize Database with Daily Data	Drew	2 hours
<b>Acceptance Criteria</b>		
There is an instance of MySQL running on an AWS server.		
Given that the instance of MySQL is running, it is accessible with proper authentication.		
The database is initialized with our script and all of the daily data.		

### User Story #6

As a project owner, I would like the website to be deployed.

Task	Owner	Time
Set up a web server using Elastic Beanstalk	Drew	8 hours
Updates with Git will deploy to the server	Drew	2 hours
<b>Acceptance Criteria</b>		
A web server is running on AWS with the current version of our application.		
It is easy to deploy the web server when the Github remote is updated.		
If the Node server crashes, it will automatically restart and be constantly up.		

### User Story #5

As a user, I would like to be able to interact with graphs.

Task	Owner	Time
Organize TDP algorithm outputs for use by multiple graphs.	Clayton	10 hours
Improve JSON parsing on client side	Vritant	4 hours
Allow user to see multiple graphs	Vritant	10 hours
Write project owner's test cases, run them, and correct outputs as needed.	Clayton	7 hours
Work with project owners on test cases and expected outputs.	Clayton	3 hours
Learn testing framework and use it to automate new test cases.	Clayton	4 hours
<b>Acceptance Criteria</b>		
Multiple Graphs can be generated by the user		
Outputs of these graphs are scaled nicely and display accurate data.		

### User Story #6

As a user, I would like the website to be responsive and adaptive

Task	Owner	Time
Redesign UI	Vritant	6 hours
Change pixel bound containers to be responsive	Vritant	5 hours
Add animations to UI components	Vritant	5 hours
<b>Acceptance Criteria</b>		
Website UI looks good on tablets and desktops		
Animations work and help better user experience		

### User Story #7

As a user, I would like to switch amongst various data sets

Task	Owner	Time
Plan client server communication for each graph.	Vritant & Clayton	6 hours each
Implement server side data passing between each graph.	Clayton	4 hours
Allow user to make specific graphs	Vritant	4 hours
<b>Acceptance Criteria</b>		
User can add graphs using dropdown menus.		

### User Story #8

Improve upon User Story #5 (Selecting a location on a map) from Sprint 1

Task	Owner	Time
Sort polygons from GeoJSON on the map.	Drew	10 hours
<b>Acceptance Criteria</b>		
By searching for a location, the map returns an id without the user clicking.		

## User Story #9

Improving upon User Story #3 (using the Transforming Drainage Project's data)

Task	Owner	Time
Change units on all Transforming Data Project files. (Requested by project owners)	Jonah	4 hours
Write script to re-initialize database with new unit values	Jonah	6 hours
<b>Acceptance Criteria</b>		
The data returned by the database will be units that are consistent with the user's input.		

## Backlog

- ☐ As a user, I would like to open the website
- ☐ As a project administrator, I would like to initialize the database
- ☐ As a user, I would like to be able to use Transforming Drainage Project's data
- ☐ As a user, I would like to see graphical representations of data
- ☐ As a user, I would like my uploaded data to be calculated and shown on a graph
- ☐ As a user, I would like to have my own data interlaced with Transforming Drainage Project's data
- ☐ As a user, I would like to upload my own data to use
- ☐ As a user, I would like to download the results of the calculated data
- ☐ As a user, I would like the project to have as less bugs as possible
- ☐ As a project administrator, I would like the website to be secure
- ☐ As a user, I would like to interact with the graphs
- ☐ As a project administrator, I would like the server to be set up
- ☐ As a project owner, I would like the website to be deployed.
- ☐ As a user, I would like to switch amongst various data sets
- ☐ As a project administrator, I would like the database to be hosted on AWS
- ☐ As a user, I would like the website to have good user experience
- ☐ As a user, I would like the website to be responsive and adaptive
- ☐ As a user, I would like instructions on how to use the website