

# Sprint 1 Planning Document

## RESERVOIR PLANNING TOOL

By The Transforming Drainage Project 

### Sprint Overview

Over the course of this sprint we plan on getting the core functionality of the website taken care of. The front-end tasks include: adding the Google Maps service, displaying mock graphs, adding GeoJson layering to the map, creating form data inputs, and sending an id to the server based on the selected location. Our backend team will be working initializing our database, creating a general routing system on the backend, and writing the Transforming Drainage Project's algorithm on the backend using inputs from form data.

### Meetings

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

### Risks and Challenges

As this is our first sprint, the biggest challenge is beginning to technically actualize our project stakeholders' vision for this application. Included in this are challenges such as: ensuring the geo-overlay indexing format can be formalized with our stakeholders, appropriately setting up a project of this size, and effectively managing our time to give all our classes enough attention. To meet these challenges, we intend to formalize the geo-overlay indexing method at our next stakeholders meeting. To set up this project appropriately, we will have to rely on each other's expertise to setup and connect our separate components while checking each other's work. Project setup is also our biggest risk as the decisions we make will have a ripple effect on the rest of the project's difficulty. Finally, effective time management will be crucial to our success and this will need to be something that every member of this team actively tracks.

## Sprint Detail

### User Story #1

As a user, I would like to be able to open the website.

Task	Owner	Time
Add express support to backend.	Clayton	2 hours
Handle page routing.	Clayton	2 hours
Set up basic page layout.	Vritant	4 hours
<b>Acceptance Criteria</b>		
Given that the required dependencies are installed, the server should start up without errors.		
Given that the index route is requested, the home page is served to the browser.		
Given that the home page is loaded in the browser, there should be a basic (not necessarily styled) page with a map, form, and a chart.		

### User Story #2

As a project administrator, I would like to initialize the database.

Task	Owner	Time
Parse daily data files into tables.	Jonah	4 hours
Dynamically create, delete, or update tables based on present data files.	Jonah	6 hours
<b>Acceptance Criteria</b>		
Given that the database is configured and model data from the TDP is accessible, the database initialization script can be run.		
Given that the script has been run, database tables will contain all of the daily model data for any given location.		
Given that one of the files has been changed, the script can be run again and the database will be updated accordingly.		

### User Story #3

As a user, I would like to be able to use Transforming Drainage Project's data.

Task	Owner	Time
Provide interface for querying data based on map id.	Jonah	5 hours
Provide interface for querying partial data based on missing input from user uploaded data files.	Jonah	5 hours
<b>Acceptance Criteria</b>		
Given that the database is initialized, we can pass a location ID and get all of the daily data associated with it.		
Given that the database is initialized, we can pass a location ID and get just the daily evapotranspiration data associated with it.		
Given that a segment of the daily evapotranspiration data is out of our available range, the system will gracefully select what we are able to provide and return it.		

### User Story #4

As a user, I would like to have my own data interlaced with Transforming Drainage Project's data.

Task	Owner	Time
Parse and verify uploaded data	Jonah	5 hours
Supplement missing user data with Transforming Drainage Project's own data.	Jonah	5 hours
<b>Acceptance Criteria</b>		
Given a CSV file in the format a user would upload, this system parses the daily data into Javascript in the format that the algorithm system would accept.		
Given a CSV file with missing data, the system queries the data and fills in any gaps.		
Given a CSV file with incorrect types, the system will reject it and report an error.		

### User Story #5

As a user, I would like to select a location on a map.

Task	Owner	Time
Adding the Google Maps API to home page.	Drew	5 hours
Add GeoJson as a map overlay.	Drew	7 hours
Add location lookup by address.	Drew	5 hours
Select location and receive correct id associated with the location.	Drew	6 hours
<b>Acceptance Criteria</b>		
Given that the home page is loaded, there is a google maps interface with an overlay of regions to select.		
Given that the map is loaded, you can select a region by clicking on the map and confirm your location.		
Given that the home page is loaded, you can alternatively enter an address to select your location. It will then appear on the map.		

### User Story #6

As a user, I would like to graphical representations of outputted data.

Task	Owner	Time
Add Google Graphs to the home page.	Vritant	8 hours
Add mock data to graphs.	Vritant	2 hours
Display both bypass flow and storage deficit on graph.	Vritant	4 hours
Query the server with id from map for data	Vritant	3 hours
Prepare calculated region data for graphing.	Vritant	7 hours
<b>Acceptance Criteria</b>		
Given that the home page is loaded, there will be a graph in the results section with mock data.		

Given that the graph is loaded, there are properly labelled and explained axes for pond volume (X axis), bypass flow (Y axis), and storage deficit (Y axis).

Given that the graph is loaded, you can hover over data points to see exact values at each possible pond volume.

### User Story #7

As a user, I would like the website to calculate data for graphing using my input

Task	Owner	Time
Write the Transforming Drainage Project algorithm for use on the server.	Clayton	10 hours
Accept and parse form data from client.	Clayton	6 hours
Accept and parse map data from client.	Drew	7 hours
Pass form data input as input to the Transforming Drainage Project's algorithm.	Clayton	5 hours
Test algorithm using mock data.	Clayton	5 hours
Create form data inputs.	Vritant	2 hours
<b>Acceptance Criteria</b>		
Given mock data, the algorithm will output daily data for each simulated pond volume.		
Given mock data, the algorithm will output yearly mean data for each simulated pond volume.		
Given form data input from the front end, the algorithm will change its outputs given		

## 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 select a location on a map~~
- ☒ ~~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 user, I would like instructions on how to use the website, and how to upload my data
- ☐ 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