

Florida International University
School of Computing and Information Sciences

Software Engineering Focus

Final Deliverable

Project Title: Biscayne Bay Quality of Underwater Data
Analytics (BBQUDA)

Team Members:

Senior Project: Ariel Omar Arias, Orson Meyreles, Leonardo Pantaleon

Capstone I: Catherine Angelini, Adrian Perez, John Quitto-Graham

Product Owner(s): Gregory Reis

Mentor(s): Gregory Reis

Instructor: Masoud Sadjadi

The MIT License (MIT)

Copyright (c) 2016 *Florida International University*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Abstract

The BBQUDA web app is a service that allows users who are conducting research on the environmental conditions in the Biscayne Bay area to upload their mission data directly and view all of the corresponding data points along with the trajectory of the mission. The tool also provides additional resources such as GPS mapping needed to run the missions and useful information that researchers may require.

Table of Contents

Introduction

Current System	5
Purpose of New System	6

User Stories

Implemented User Stories	7
Suggested User Stories	8

Project Plan

Hardware and Software Resources	9
Sprints Plan	10
<i>Sprint 1</i>	10
<i>Sprint 2</i>	10
<i>Sprint 3</i>	11
<i>Sprint 4</i>	11
<i>Sprint 5</i>	12
<i>Sprint 6</i>	12
<i>Sprint 7</i>	12

System Design

Architectural Patterns	13
System and Subsystem Decomposition	13
Deployment Diagram	14
Design Patterns	14

System Validation 14

Robotics Ecosystem

EcoMapper	15
VectorMap	15
QGIS	16
ROS	17
Field Work	17
Data Collection	18

Appendix

Appendix A - UML Diagrams	20
Appendix B - User Interface Design	28
Appendix C - Sprint Review Reports	38
Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents	40

Introduction

Current System

There was a basic system created by the Product Owner Gregory Reis that we took some design cues from. Aside from this, the project was completely built from scratch.

Purpose of New System

The purpose of the BBQUDA web app is to provide a site for researchers to be able to plan and view data from missions conducted on the health of underwater ecosystems. The web app allows for a place to show all of the data in a easy-to-read manner that allows the user to visually see the trajectory of the robot and the data points at each given point along the path. The ability for users to have personal logins that they can then upload and save multiple test missions allows them to easily switch between analyzing various missions. The web app also allows users to select new points on a global map for a new mission and export these points, making it much easier to create new mission plans in the various third-party software solutions. We also provide resources in our dashboard page, should the user need to contact anyone to help with the project.

User Stories

Implemented User Stories

Story ID	Story Title
001	As a developer, I want to choose a tech stack by fully understanding the requirements and finding what would work best to fulfill them.
002	As a developer, I want to set up a Github repository so I can version control along the development process and have effective collaboration among the team
003	As a developer, I want to familiarize myself and set up all dependencies needed for Django
004	As a developer I want to plan the frontend and have an initial Hello World landing page for the user to see
005	As a developer I want to set up the SQLite backend
006	As a developer I want to be able to change the file format of the data logs from .dvl to .csv
007	As a developer I want to be able to delimit the csv files in an automated process in order to have them appropriately structured for easy display
008	As a developer I want to develop database models to store mission data and user data
009	As a developer I want to have a basic frontend that allows a .csv file to be uploaded properly formatted
010	As a developer I want to have a basic frontend that allows a log file to be uploaded, converted to .csv, and properly formatted
011	As a user, I want to be able to sign up and login into my own account
012	As a user I want to be able to save my missions on the site
013	As a user I want to see a map that has readings of the ecomapper at various points along the mission
014	As a user I want to land on a homepage when I first come to the site so I can navigate to create a user
015	As a developer I want to take out the ecomapper robot to test missions in real-world environments and gather real data to test our web app with
016	As a developer I want to speak to professionals to see how to improve the current product

017	As a developer I want to add functionality to allow for password resets on user accounts
018	As a developer I want to begin looking into the setup process for the Clearpath Heron robot
019	As a user I want a dashboard that has contact information for various parties involved with the robot should I need any help
020	As a user I want the ability to place points on a map and download the coordinates of the path I have created
021	As a developer I want to find an appropriate hosting service to deploy the web app

Suggested User Stories

S-01	As a developer, I want the web app to be fully deployed and accessible anywhere
S-02	As a user, I want different formats to download new coordinates to make sure any outside solution is usable

Project Plan

This section describes the planning that went into the realization of this project. This project incorporated the agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

Hardware and Software Resources

Hardware:

Development devices:

	Desktop PC	Macbook Pro (Mid 2015)	Desktop PC
Processor	3.2 GHz AMD Ryzen 1600	2.2 GHz Quad-Core Intel Core i7	3.4 GHz AMD Ryzen 2600
Capacity	500GB SSD	250 GB Flash Storage	500GB SSD
Memory	16 GB RAM	16 GB 1600 MHz DDR3	16 GB RAM

Autonomous Vehicles:

- EcoMapper Iver3 EP Open System UUV
- Clearpath Heron

Software:

- Windows 10
- Visual Studio Code
- Django
- Python
- Javascript
- HTML5
- CSS
- jQuery
- SQLite
- Github
- Vectormap

Sprints Plan

Sprint 1

Sprint Goals:

- To familiarize ourselves with the requirements set by the Product Owner and lay the foundation for the tech stack we are going to be using for the project.

Sprint User Stories:

- User Story 001
 - Choose a tech stack
- User Story 002
 - Set up Github repository
- User Story 003
 - Install and familiarize ourselves with Django
- User Story 004
 - Set up basic front end
- User Story 005
 - Setup database
- User Story 006
 - Successfully convert data log format to csv

Sprint 2

Sprint Goals:

- To create the basics of the web app and make sure that the front and backends are communicating properly.

Sprint User Stories:

- User Story 005
 - Setup database
- User Story 006
 - Successfully convert data log format to csv
- User Story 007
 - Properly delimit csv files
- User Story 008
 - Properly store uploaded files
- User Story 009

- Properly upload csv file

Sprint 3

Sprint Goals:

- Finalize file upload and have data visualization. First in-person test missions also began here.

Sprint User Stories:

- User Story 010
 - Properly upload, convert and store log file
- User Story 011
 - Have user sign up/login feature
- User Story 012
 - Have user-uploaded missions be saved
- User Story 013
 - Show interactive map of ecomapper trajectory
- User Story 014
 - Landing page for users to sign up/login
- User Story 015
 - Take out the EcoMapper for real-world tests

Sprint 4

Sprint Goals:

- Get feedback on current implementation and improve upon based on that. Continue real-world testing.

Sprint User Stories:

- User Story 015
 - Take out the EcoMapper for real-world tests
- User Story 016
 - Get feedback on current product
- User Story 017
 - Have user be able to update their password
- User Story 018
 - Explore setup for the Clearpath Heron

Sprint 5

Sprint Goals:

- Continue getting feedback and implement a dashboard page with helpful links. Continue real-world testing.

Sprint User Stories:

- User Story 015
 - Take out the EcoMapper for real-world tests
- User Story 016
 - Get feedback on current product
- User Story 019
 - Set up dashboard page

Sprint 6

Sprint Goals:

- Add user-generated coordinates feature and host the web app.

Sprint User Stories:

- User Story 015
 - Take out the EcoMapper for real-world tests
- User Story 020
 - User can map their own path and download the coordinates
- User Story 021
 - Host web application

Sprint 7

Sprint Goals:

- Complete final deliverable documentation
- Record and prepare all desired videos
- Complete the final presentation
- Create posters

System Design

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

Architectural Patterns

For our architecture, we had a choice of various services that we could build our architecture on. Our decision came down to our need for users and a unique database design. Also choosing something that all members were familiar with or had knowledge of would allow the development process to be greatly expedited. Using the type of MVC architecture allowed us to handle all types of interactions our users could potentially have with the site.

System and Subsystem Decomposition

We chose to build on Django 3.1.2 since it is built on Python and comes with support for many features which we deemed critical. Django's template engine would allow for ease of creating and displaying website pages to the frontend. Django's built in user and authentication system was also a great time saver in this project since they were all features we would have to manually implement otherwise. Using Django as our backend and media storage would allow for quick expansion and ease of manipulation with CSV files.

In order to have dynamic maps, we are using Leaflet and jQuery in order to pull the proper information from the files stored in the database. Changes were pushed to Github in order to have different developers work on various components at the same time and have appropriate version control.

Deployment Diagram



Design Patterns

The model, view, controller architecture is commonly used for applications because it divides program logic into 3 interconnected parts. Django does a good job at providing boilerplate for many of the model-view-controller functionality as well as making it simple for expansion. Our Model component consists of two separate data handling techniques, mainly a database to store user information and a Django media folder to hold user uploaded CSV mission files. Our views are handled through Django's template engine and Bootstrap.

System Validation

Because of our desire to create the most streamlined interface for our web application, the testing of it was easy enough to complete manually. The site is only usable for users who have registered and are logged in, so we only have to worry about one type of user. Test cases include trying to upload files of the wrong type in the upload sections of the site. We also attempted to have users attempt to register with login info that was already in use and making sure it would not work. The functionality of every feature was also verified, such as the proper display of the maps in various pages on the site.

Robotics Ecosystem

This project's ultimate goal is to assist the robotics research community in being able to conduct successful underwater missions. That being said, it is important that we take into consideration the actual usage of a robot when designing and building the BBQUDA application. Our team is unique in that this application serves to not just assist others, but to also assist our very own water quality endeavors. To accurately model our task and provide working data for the team to use, we conducted several missions with the EcoMapper underwater robot and utilized the development stack surrounding it. Likewise, we began work on ways of giving users and ourselves more ease of access for creating missions end-to-end. To do this we also made use of the Robot Operating System, ROS, and the Heron surface water vehicle.

Here we will discuss the use of the following in our work with the robots intended to be used in conjunction with BBQUDA:

- EcoMapper
- VectorMap
- QGIS
- ROS
- Field Work
- Data Collection

EcoMapper

The YSI EcoMapper AUV is our underwater vehicle of choice when it comes to data collection in the Biscayne Bay. The AUV comes equipped with several sensors such as YSI conductivity and temperature sensors, YSI depth sensor, blue-green algae, pH/ORP, and many more which were able to provide us with useful data in our missions. We chose this particular AUV for several reasons but primarily it is because of its ease of availability, prominence in the underwater robotics research community, and the support we have from the PO's prior work. Normally, to operate the EcoMapper, one would need to use VectorMap, its proprietary software for mission planning. Much of the information learned from using the EcoMapper was reflected in the design choices for this project.

VectorMap

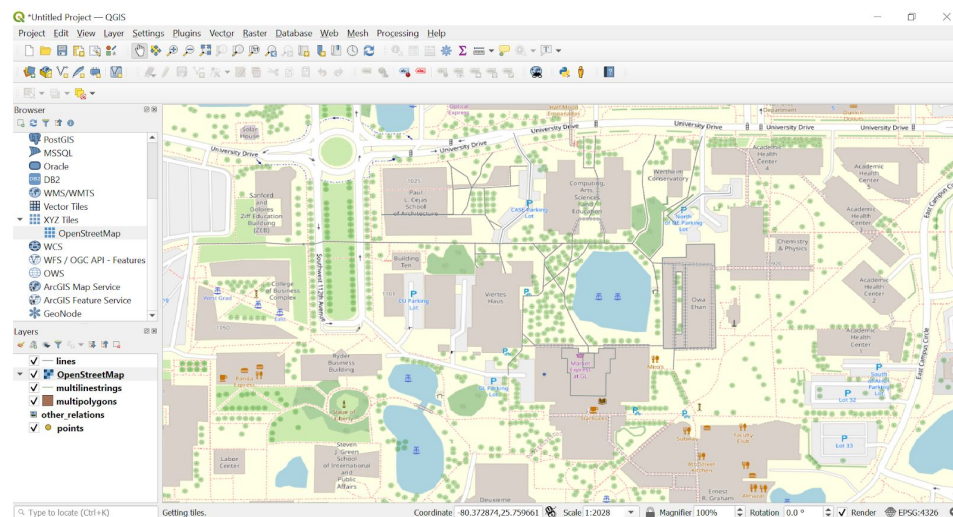
VectorMap is a proprietary software used by the EcoMapper to plan missions for data collection and water traversal. The software enables a user to generate waypoints along geo-coordinates for the robot to follow and perform desired tasks. This can include operations such as diving, sonar mapping, and active/passive data collection to name a few. This software, however, suffers from some key flaws.

Mainly, the software is simply in-accessible and clunky to use. Waypoint editing and geo-mapping prove to be a challenge through not only the user-interface but also in the actual conversion from waypoint movement to real life execution. Also not to mention that the system is a complete blackbox in nature. So if a researcher desires something more precise or autonomous, it is near impossible to do so.

For this semester, we chose to work with VectorMap as a means to produce intermediary results. We do, however, expect that our work with ROS will allow us to create an easier, more user friendly, and precise solution to not just the EcoMapper, but other water robotic solutions. Still, though VectorMap was hard for the team to use, we were able to conduct many successful missions with the program and provide important data to the team, the local government, and FIU.

QGIS

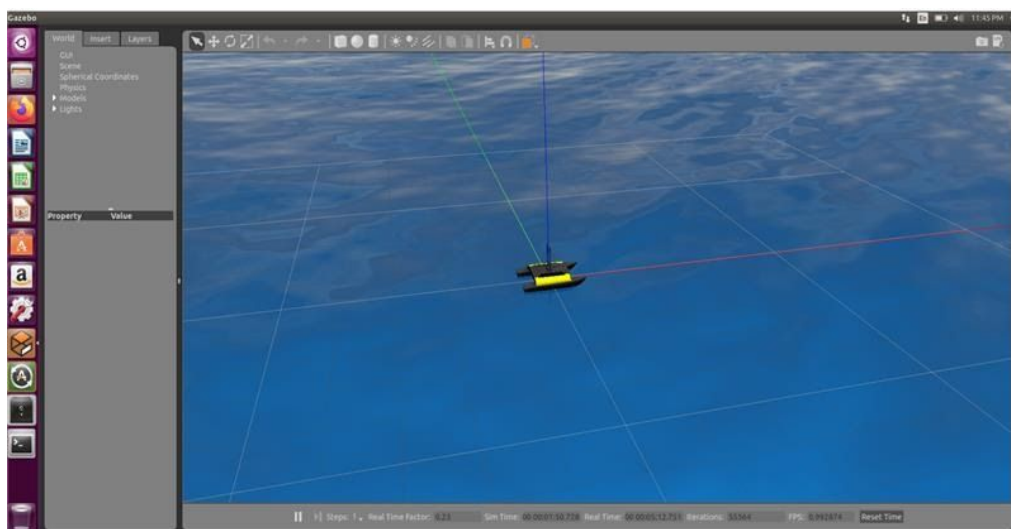
QGIS is a free geographic information system (GIS) software that allowed us to create and export a Tagged Image File Format (TIFF) file needed for VectorMap. In order to create the missions in VectorMap, a TIFF file was required of the area the EcoMapper will cover with coordinates of each part of the map embedded into it. A jpeg or png image would not be enough, as the missions we had to create for the EcoMapper work through the use of waypoints that have latitude and longitude coordinates. Our missions covered several areas such as FIU's Modesto A. Maidique (MMC) lake, a dockside location in FIU's Biscayne Bay campus, Little River Pocket Park, etc. Hence, for every location, we would have to export a map from OpenStreetMap as an OSM file and import this file into QGIS as a vector layer to be able to export it as a TIFF file that can be used for VectorMap. As seen below, this is the overall map of the FIU's MMC lake in QGIS, which was created when we had to conduct a preliminary mission there.



ROS

Robot Operating System (ROS) is a robotics middleware that allows developers access to sets of useful libraries and packages that simplifies the development of robot behavior. As we faced a lot of challenges using VectorMap, a better option we encountered was to set up a ROS development environment for the Heron vehicle. While there is no existing support for the Ecomapper with ROS, we still are aiming to be able to collect water quality data, which the Heron will allow us to do.

We set up our environment by using an Ubuntu 16.04.7 LTS (Xenial Xerus) virtual environment and installing Gazebo, the Heron simulator, ROS Kinetic, and the uuv simulator package, which can be found in https://github.com/uuvsimulator/uuv_simulator. This is a simulation of the Heron vehicle in Gazebo, which can be controlled using ROS scripts.



Field Work

In order to collect preliminary data for the BBQUDA application as well as for the research team, we conducted several live missions with the YSI Ecomapper. Most of these missions were conducted in the Biscayne Bay as per our research and client goals. The data collected here was integral to development and testing of what the BBQUDA application should aim to help. These missions collected a series of data points ranging from temperature to pH levels.

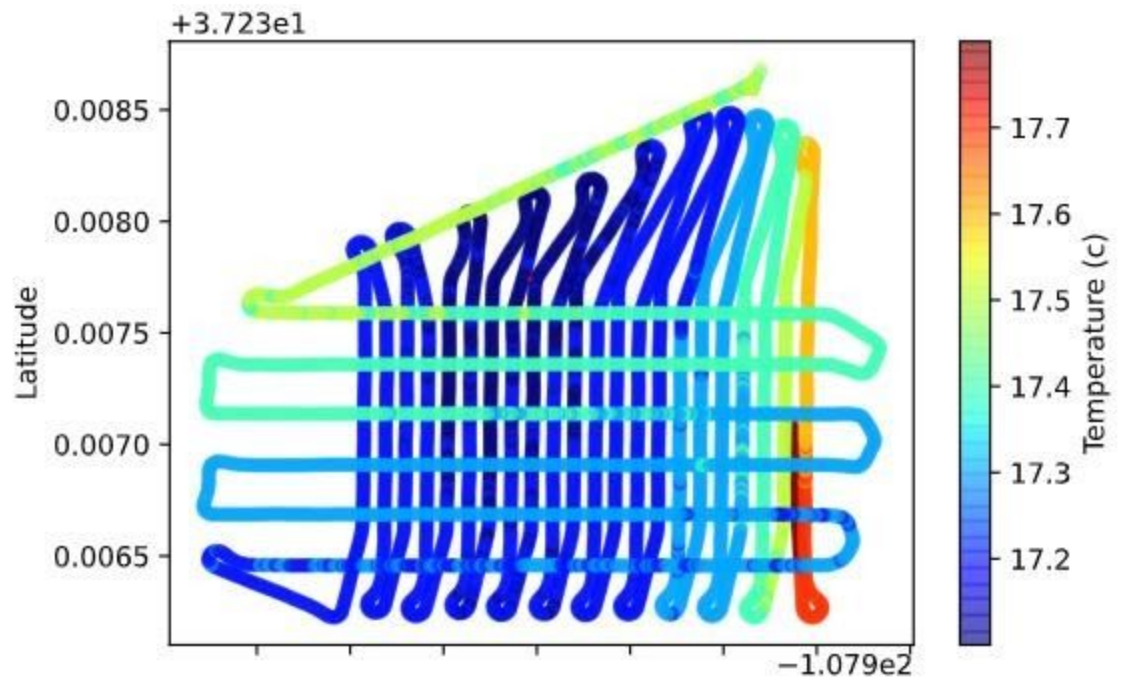
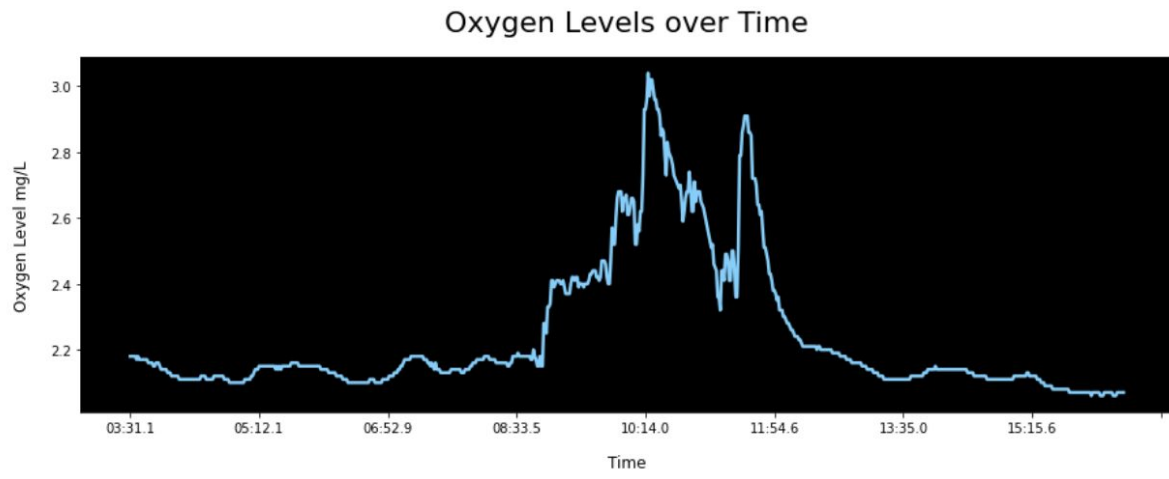
The field work also allowed us to identify flaws that exist with some of the current research methods when using underwater autonomous vehicles. We were also able to identify flaws in VectorMap and determine ways we could improve the BBQUDA application. This was one of the reasons why BBQUDA has implemented a waypoint mechanism to allow researchers to map out their research mission with a robot.

	Total Water Column (m)	Temperature (c)	pH	ODO mg/L	Salinity (ppt)	Turbid+ NTU	BGA-PC cells/mL
count	320	320	320	320	320	320	320
mean	4.0960	27.6028	7.8249	2.0867	2.8400	4.1328	-1,904.3250
std	0.7802	0.0166	0.0231	0.0683	0.3310	1.5204	1,371.9033
min	2.1800	27.6000	7.7600	1.9800	0.4900	0	-5751
25%	3.5700	27.6000	7.8100	2.0400	2.6600	3.4000	-2,748.2500
50%	4.5900	27.6000	7.8200	2.0900	2.9100	3.9000	-1898
75%	4.7300	27.6000	7.8400	2.1100	3.0500	4.5000	-1019
max	4.9700	27.7000	7.8900	2.6800	3.2200	13.7000	2034



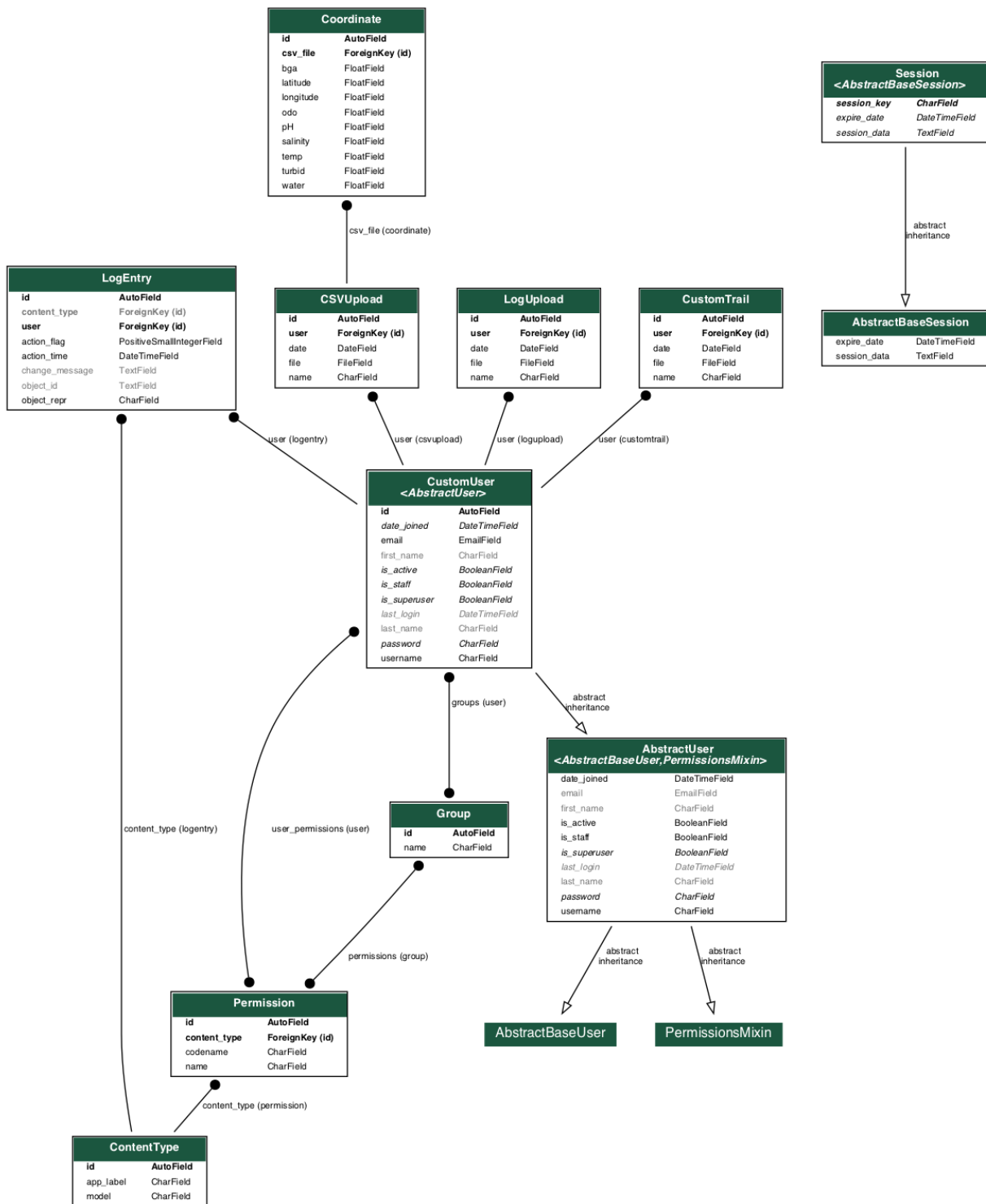
Data Collection

During each deployment of the EcoMapper, various types of data would be collected of quantitative water quality metrics such as salinity, pH levels and temperature for each second of the entire mission, as seen in the figure below. All of this data would then be saved as a csv file that could be visualized to see the oxygen levels over time, the change in pH, etc. While our focus this semester for Capstone I was on the data collection and understanding VectorMap and the EcoMapper, we were still able to generate scripts using Python and data visualization libraries that allowed us to apply statistical measures to this data and will help us for next semester, which is when we want to create more efficient and precise missions. These preliminary scripts will be improved upon and actually integrated into the BBQUDA system in the following iteration.

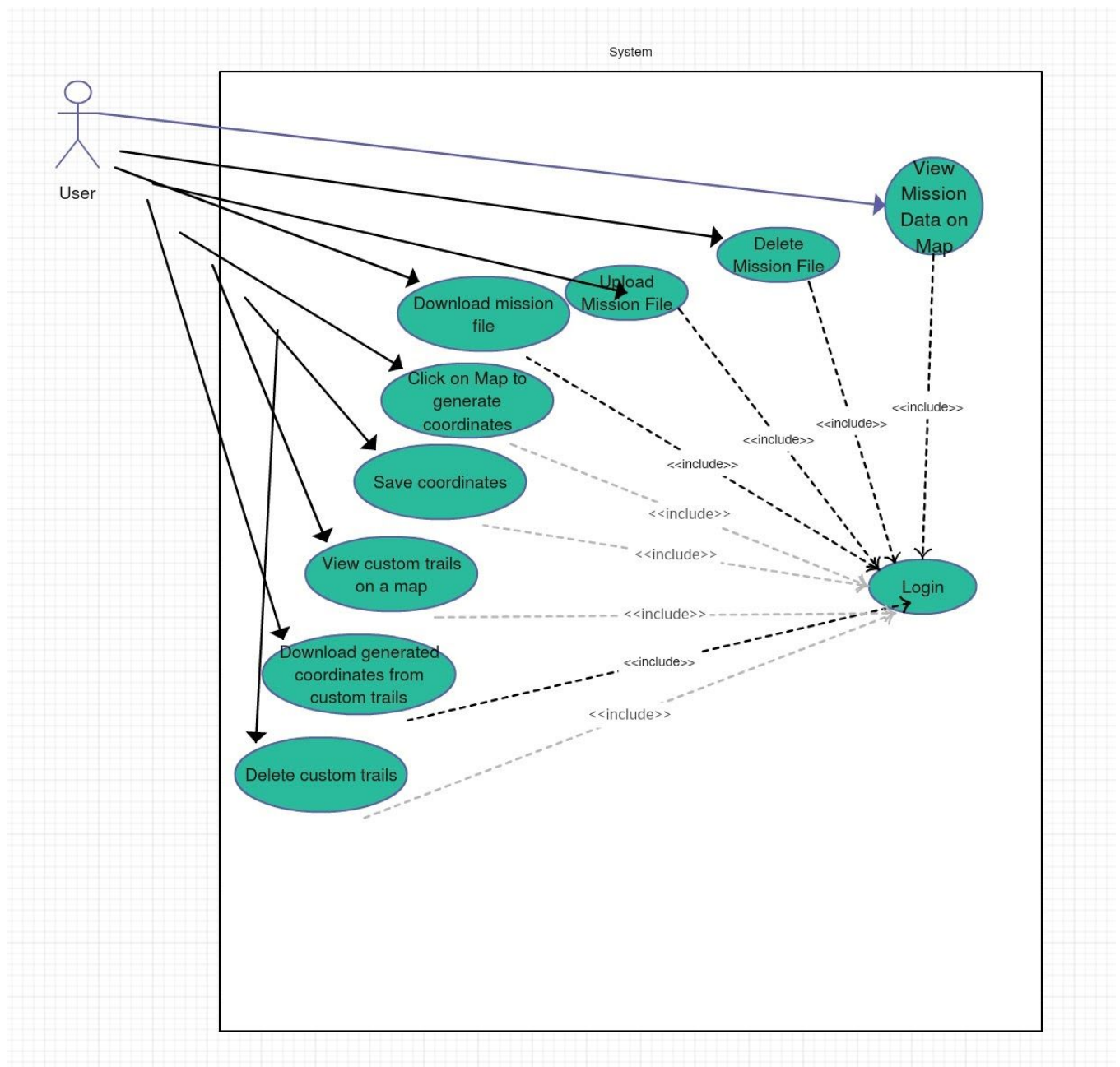


Appendix

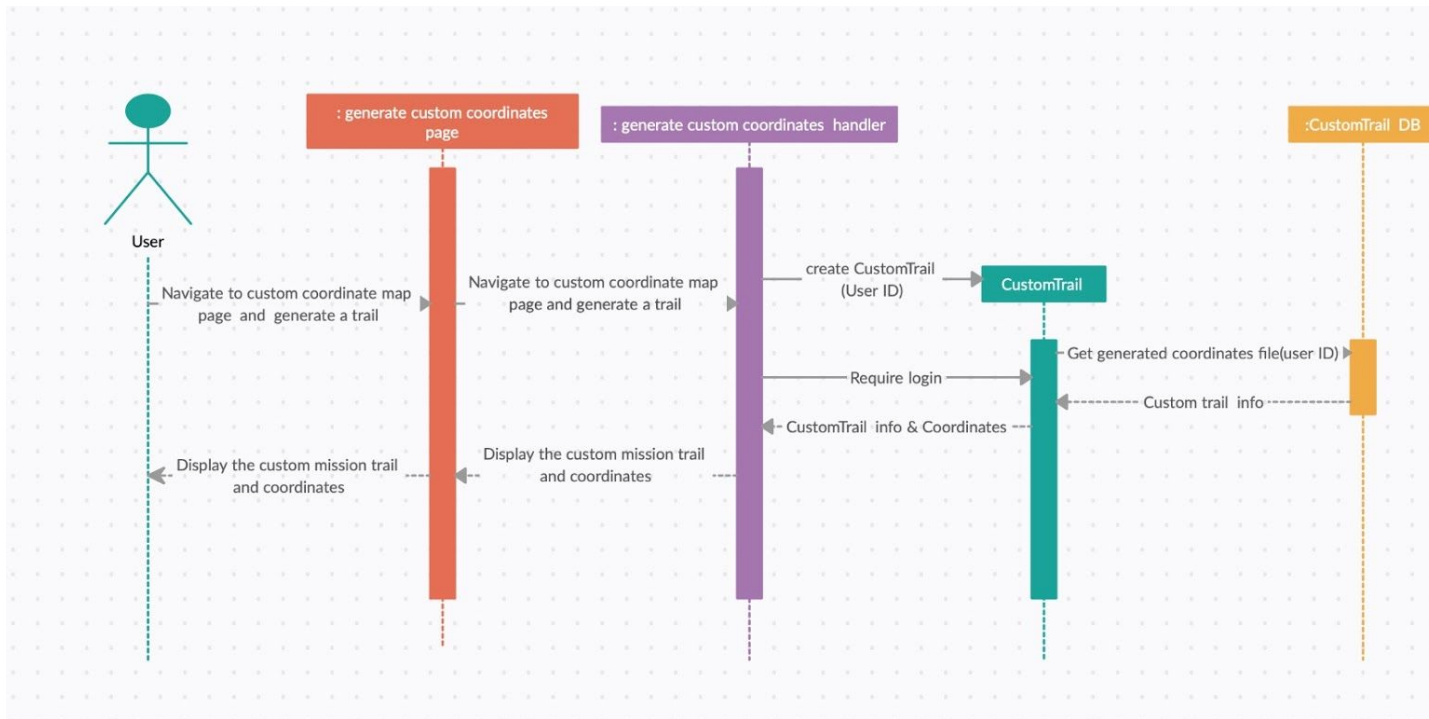
Appendix A - UML Diagrams



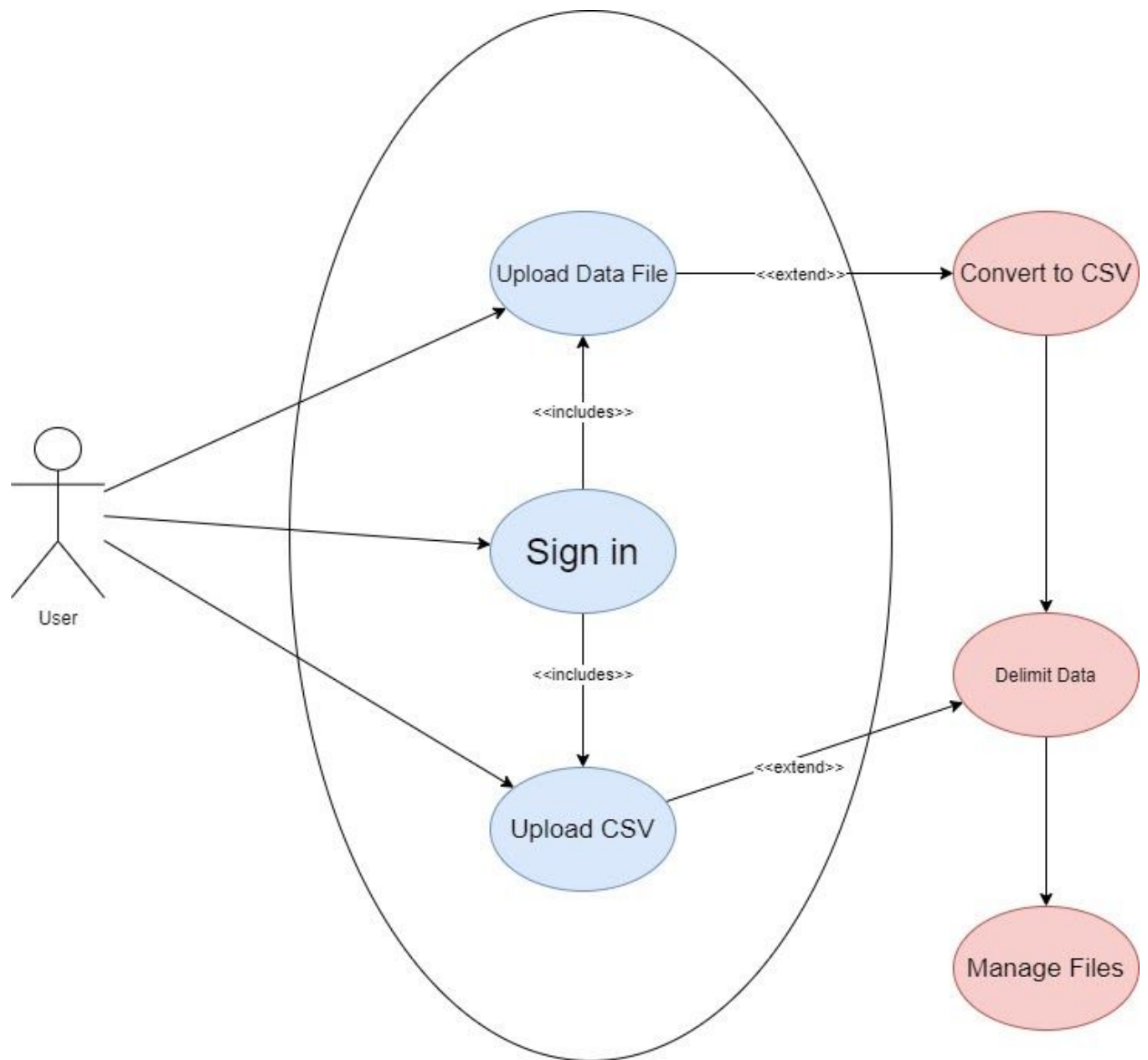
-Full Class Diagram (Ariel Omar Arias, Orson Meyreles, Leonardo Pantaleon)



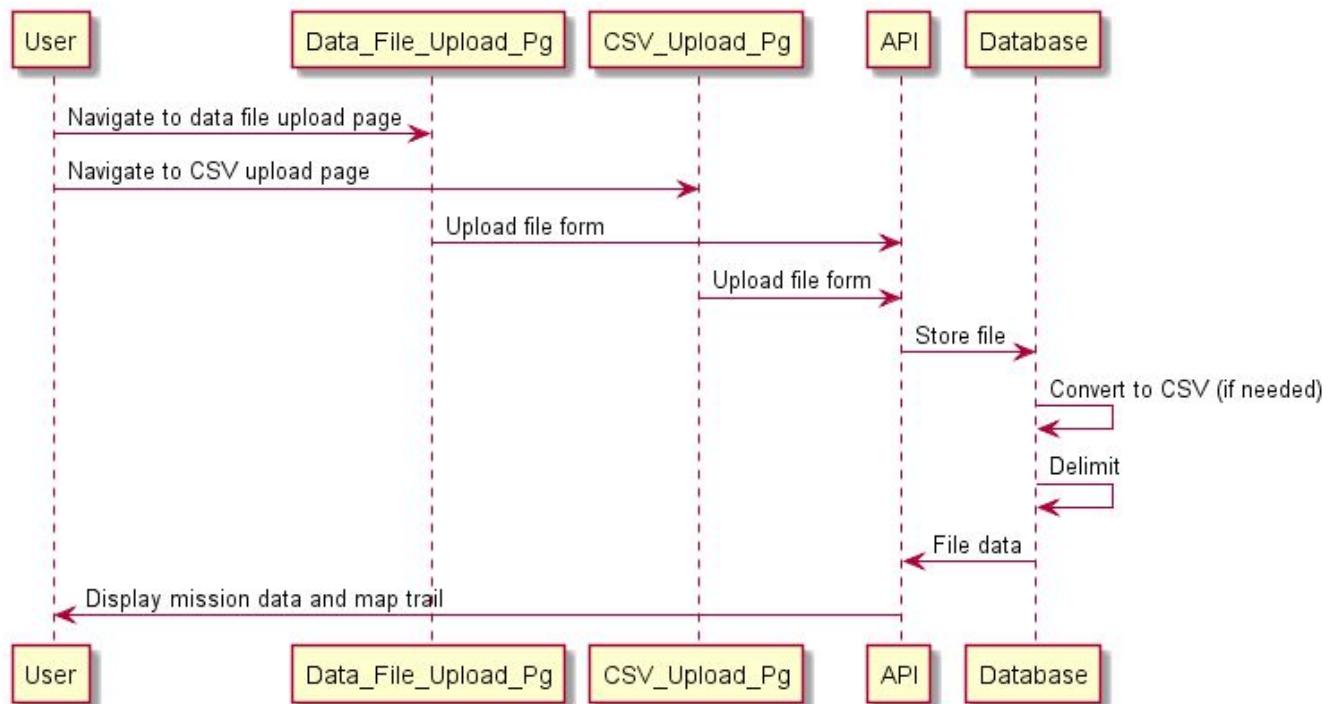
-System Use Case Diagram (Ariel Omar Arias)



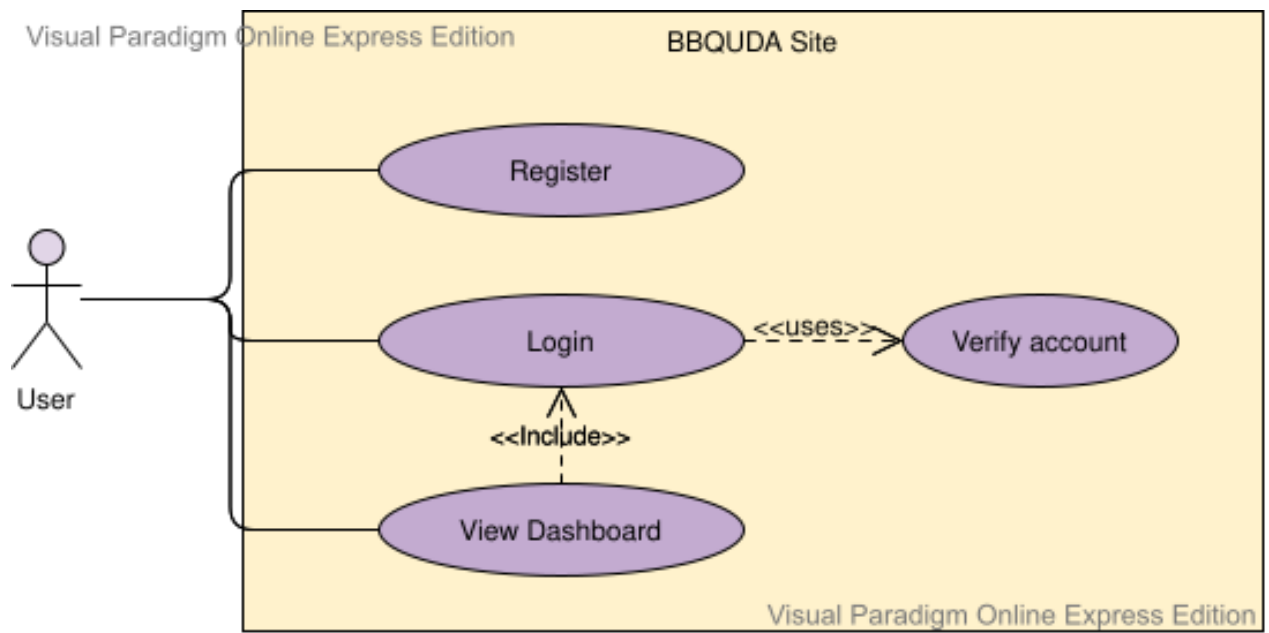
-Map Coordinate Generation Sequence Diagram (Ariel Omar Arias)



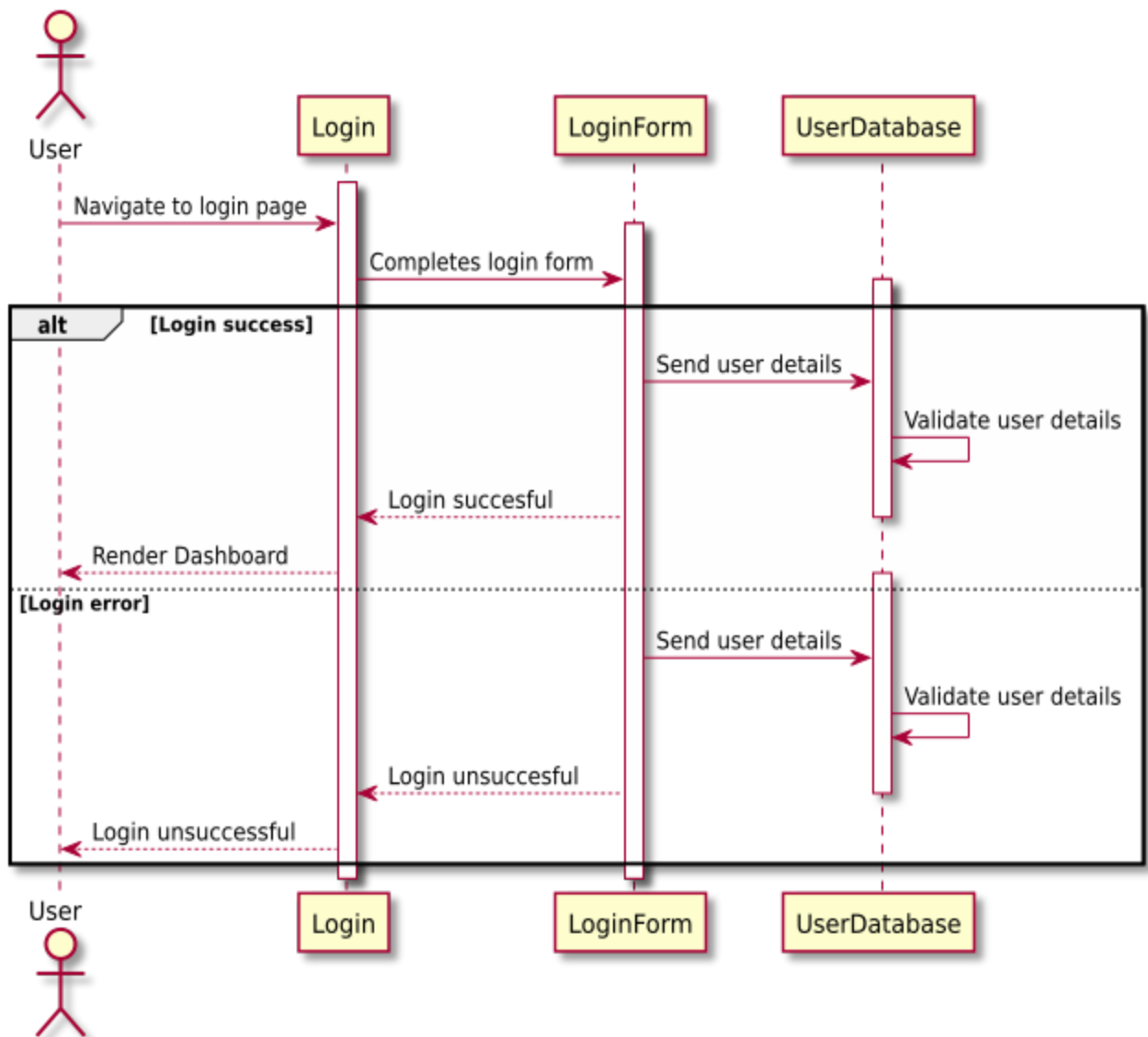
-Mission Data Upload Use Case Diagram (Orson Meyreles)



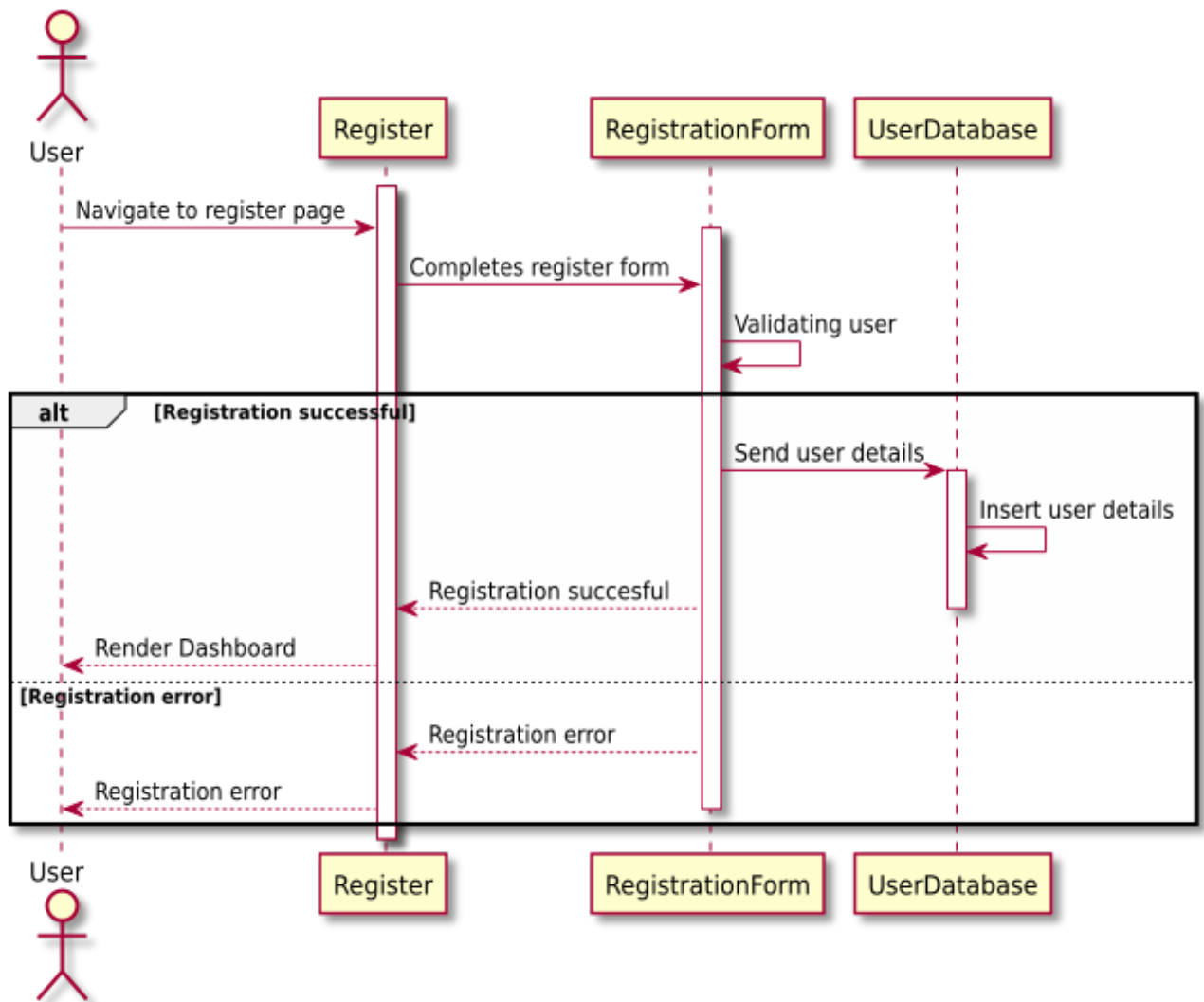
-Mission Data Upload Use Case Diagram (Orson Meyreles)



-User Login / Registration Use Case Diagram (Leonardo Pantaleon)



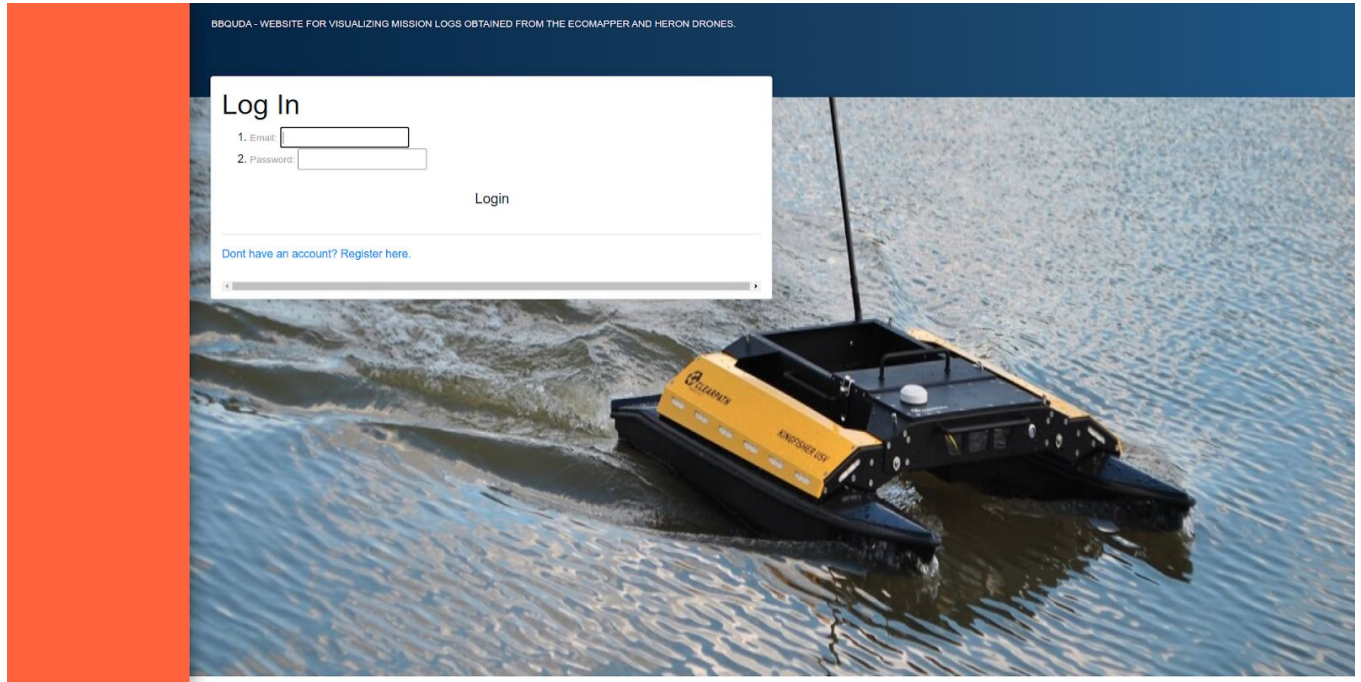
-User Login Sequence Diagram (Leonardo Pantaleon)



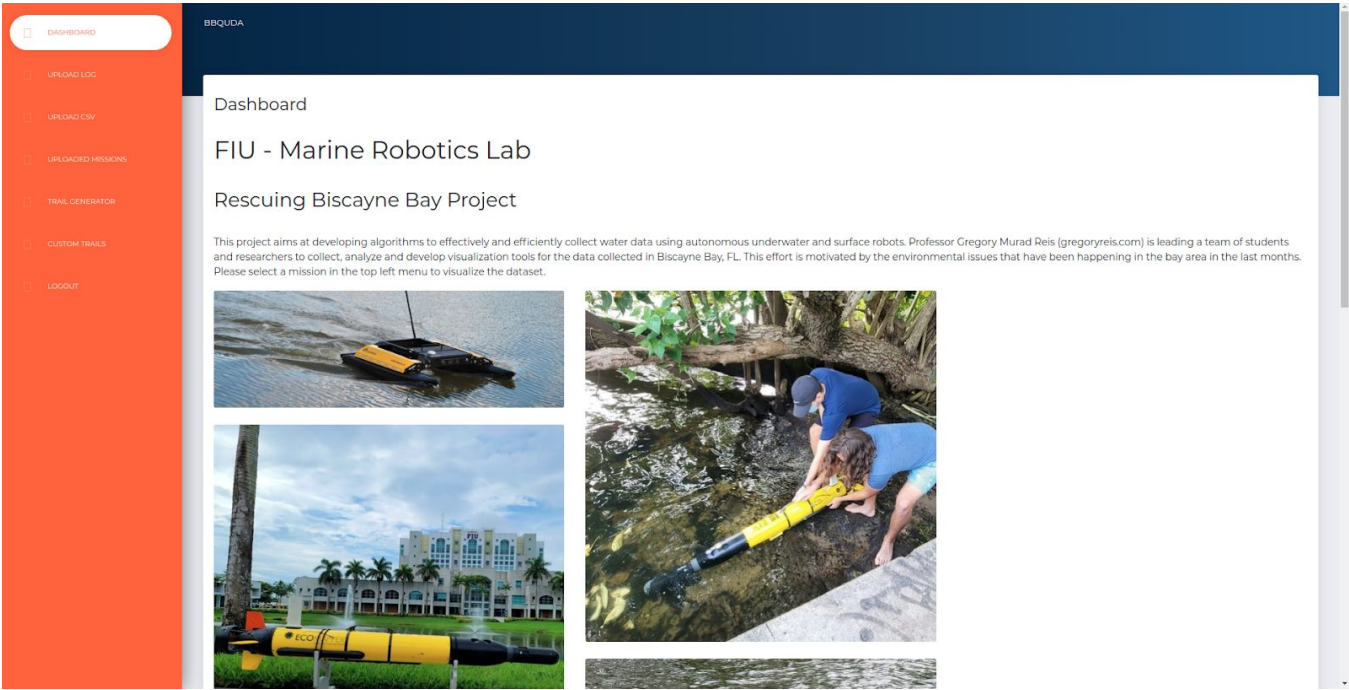
-User Registration Sequence Diagram (Leonardo Pantaleon)

Appendix B - User Interface Design

Login Page:



Dashboard:
Main page when you first login



Upload Log File:

Allows the user to upload a log file in order to be stored in the site. Opens a dialog box for the user to find the file on their system. Does not allow for a file type other than log format.

The screenshot shows the 'Upload Log File' interface within the BBQUDA application. On the left is an orange sidebar with a menu containing: DASHBOARD, **UPLOAD LOG** (highlighted), UPLOAD CSV, UPLOADED MISSIONS, TRAIL GENERATOR, CUSTOM TRAILS, and LOGOUT. The main content area has a dark blue header with 'BBQUDA' and a white modal titled 'Upload Log File'. Inside the modal, there is a file input field with a 'Choose File' button and the text 'No file chosen'. Below this is a grey bar with the text 'Upload File'. At the bottom of the modal is a horizontal scrollbar.

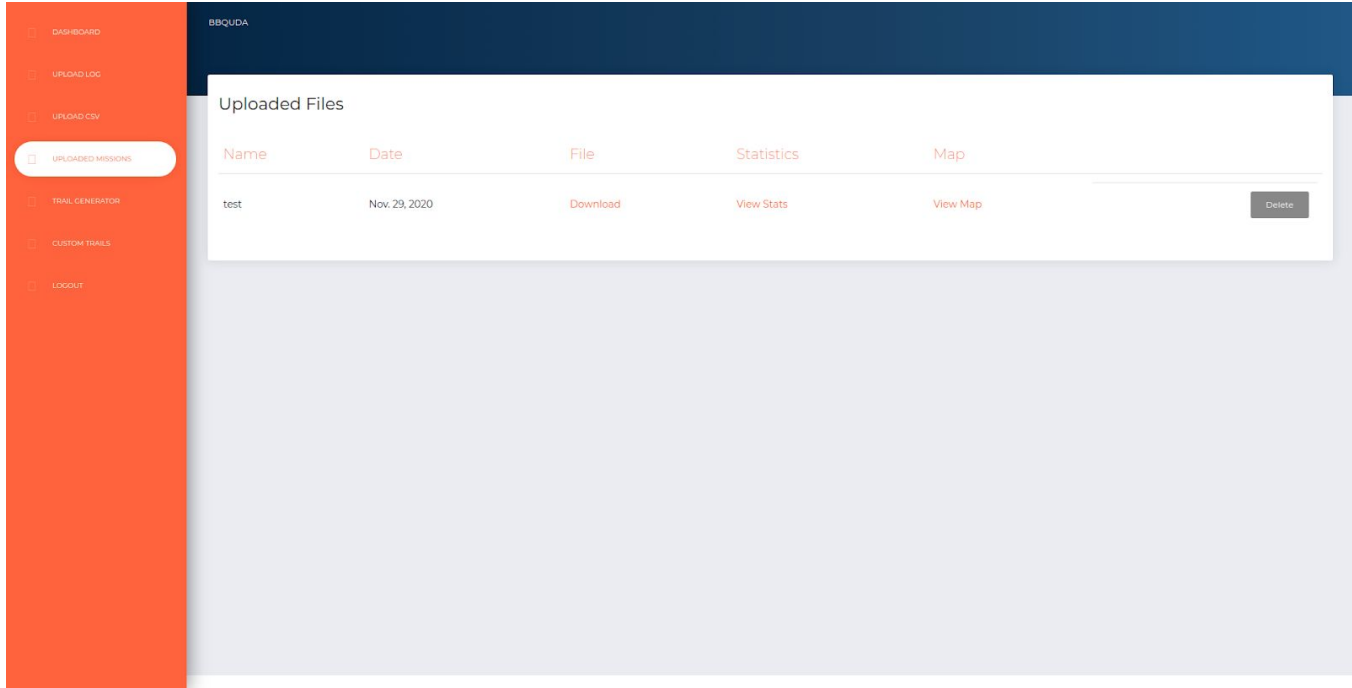
Upload CSV File:

Allows the user to upload a csv file in order to be stored in the site. Opens a dialog box for the user to find the file on their system. Does not allow for a file type other than a csv file.

The screenshot shows the 'Upload CSV File' interface within the BBQUDA application. On the left is an orange sidebar with a list of navigation items: DASHBOARD, UPLOAD LOG, UPLOAD CSV (highlighted with a white background and a file icon), UPLOADED MISSIONS, TRAIL GENERATOR, CUSTOM TRAILS, and LOGOUT. The main content area has a dark blue header with the text 'BBQUDA'. Below the header, the title 'Upload CSV File' is displayed. Under the title, there is a file selection area with a text input field and a 'Choose File' button. Below this, a message 'No file chosen' is visible. A large, dark gray button labeled 'Upload File' is positioned below the file selection area. At the bottom of the main content area, there is a horizontal scrollbar.

View Uploads:

User can view all of the missions that they have uploaded and choose whether to download them, view statistics, or view their path on a map



View Test Data:
User can view the descriptive statistics of the data the uploaded.

DASHBOARD

UPLOAD LOG

UPLOAD CSV

UPLOADED MISSIONS

LOGOUT

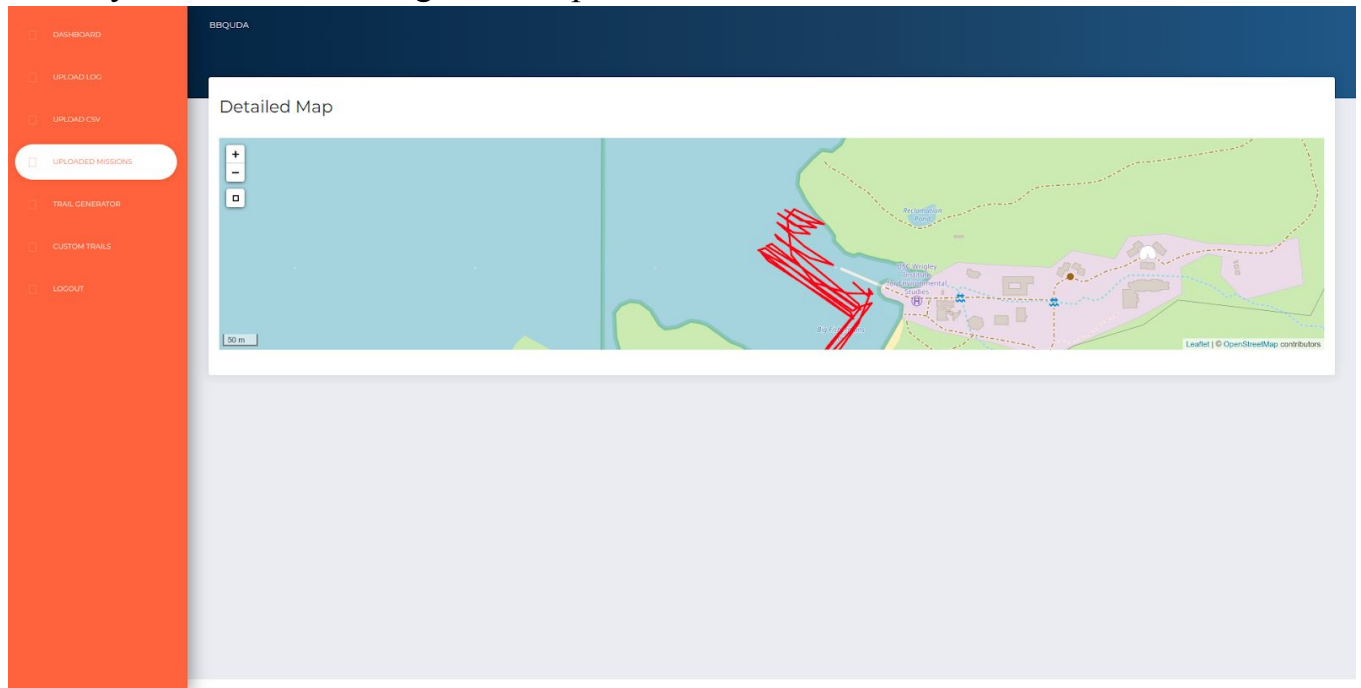
BBQUIDA

test - Descriptive Statistics

	Total Water Column (m)	Temperature (c)	pH	ODO mg/L	Salinity (ppt)	Turbid+ NTU	BGA-PC cells/mL
Count	2992	2992	2992	2992	2992	2992	2992
Mean	13.26	23.59	8.39	8.44	33.8	2.65	-428.45
STDev	7.87	0.72	0.03	0.35	0.8	26.8	10979.38
Min	0.31	22.7	8.16	8.12	5.99	-0.4	-5560
25%	5.4	23.0	8.37	8.19	33.83	0.9	-2588.0
50%	13.0	23.4	8.38	8.25	33.84	1.2	-1722.0
75%	20.25	24.0	8.4	8.64	33.85	1.6	-823.75
Max	49.43	26.3	8.45	9.66	34.01	1123.9	175773

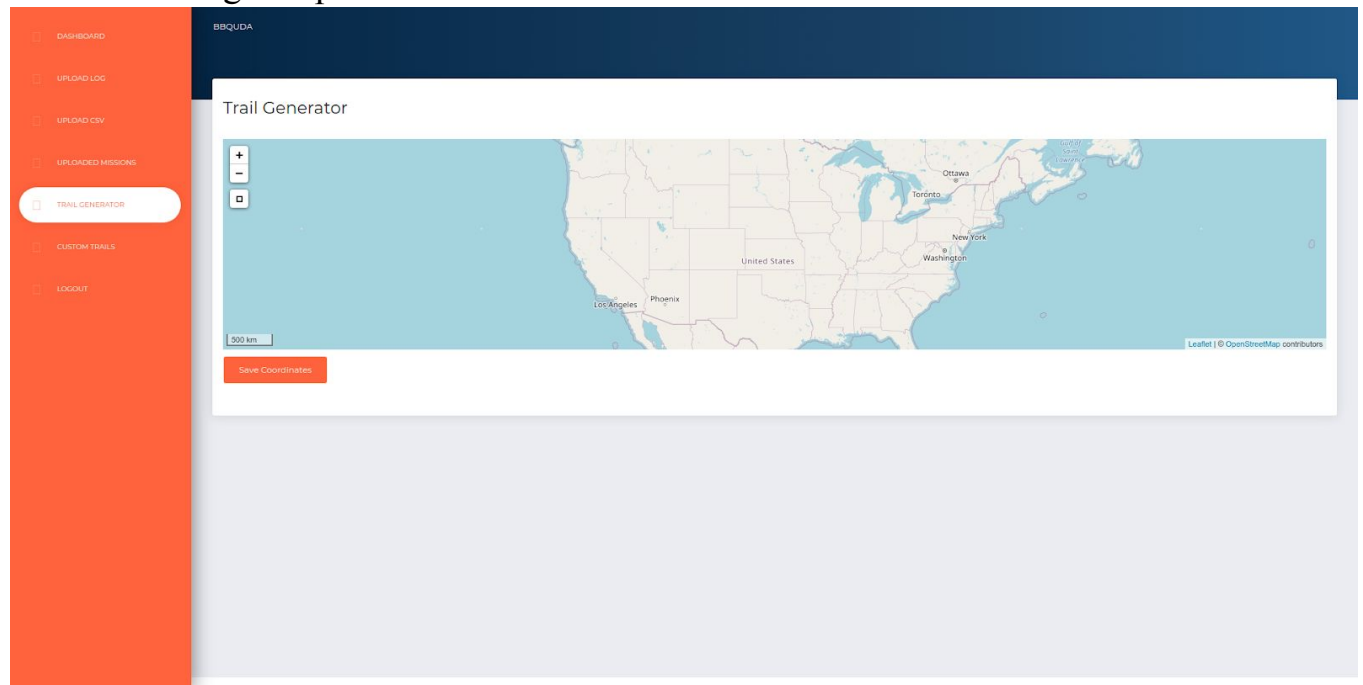
View Map:

Shows the user a map with the path that the robot took. Hovering over the trajectory allows you to see the findings at each point.



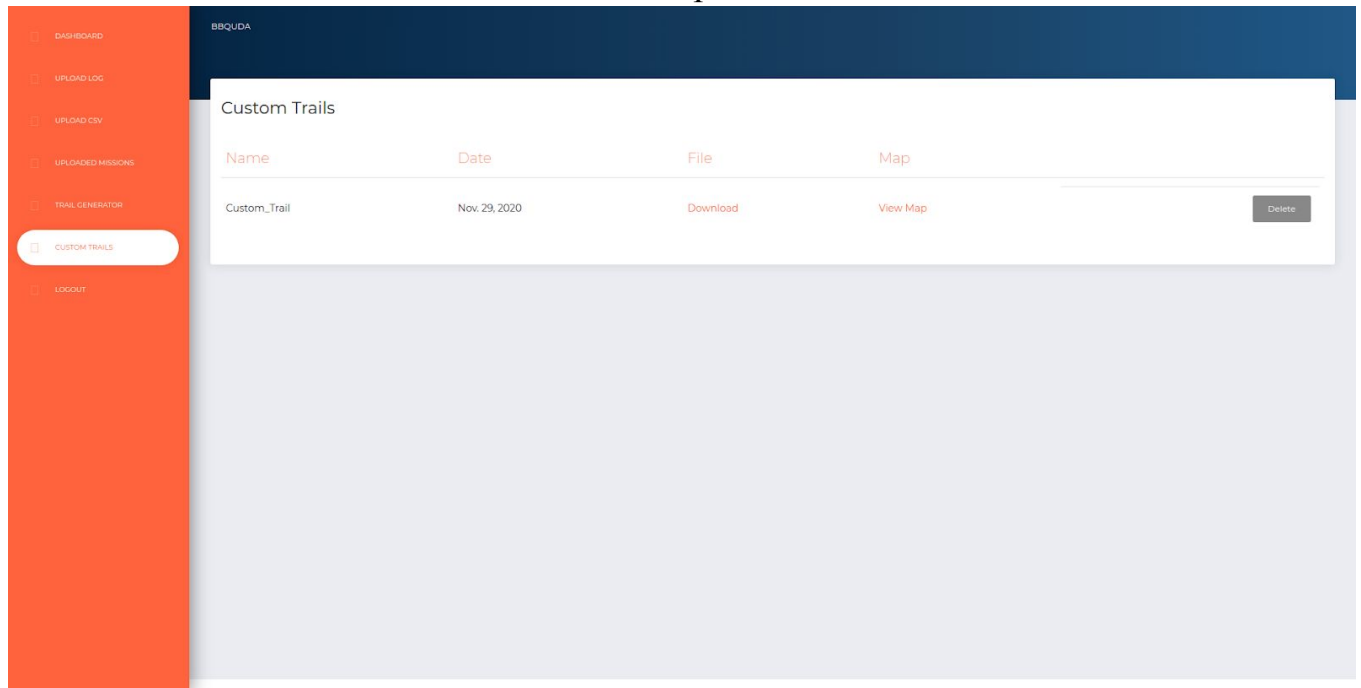
Trail Generator:

Allows the user to create their own trajectory anywhere in the world using the map. Will save after being completed.



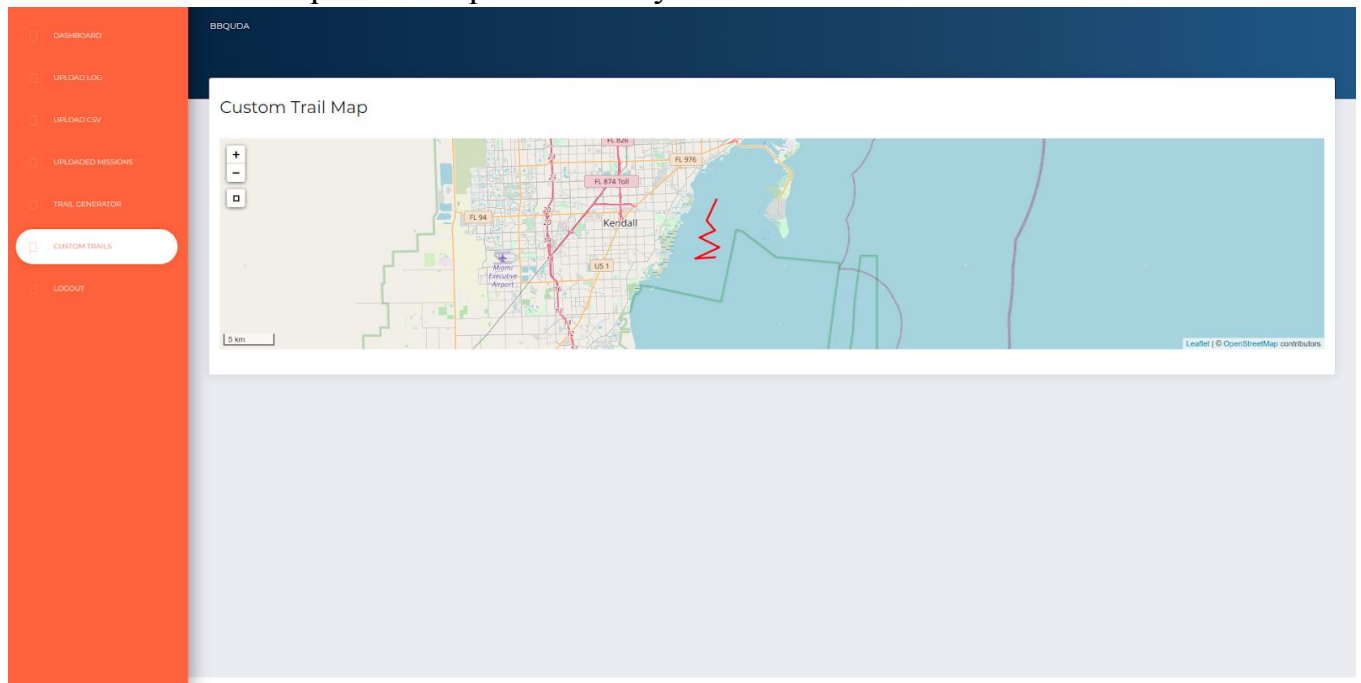
View Custom Trails:

User can view all of the custom trails that they have created and choose whether to download them as a csv or view them on a map



View Trail Map:

Shows the user a map with the path that they created.



Appendix C - Sprint Review Reports

Senior Project Team 3 Sprint 1 Review 9/13/20

- Ariel Omar Arias, Orson Meyreles, Leonardo Pantaleon, Gregory Reis
- List of Approved User Stories:
 - User Story 001 - Choose a tech stack
 - User Story 002 - Set up Github repository
 - User Story 003 - Install and familiarize ourselves with Django
 - User Story 004 - Set up basic front end
- List of Rejected User Stories:
 - User Story 005 - Setup database
 - Decided to focus more on frontend groundwork
 - User Story 006 - Convert data log format to csv
 - Needed un-converted file to test with

Senior Project Team 3 Sprint 2 Review 9/27/20

- Ariel Omar Arias, Orson Meyreles, Leonardo Pantaleon, Gregory Reis
- List of Approved User Stories:
 - User Story 005 - Setup Database
 - User Story 006 - Convert data log format to csv
 - User Story 007 - Properly delimit csv files
 - User Story 008 - Properly store uploaded files
 - User Story 009 - Properly upload csv file
- List of Rejected User Stories:
 - N/A

Senior Project Team 3 Sprint 3 Review 10/11/20

- Ariel Omar Arias, Orson Meyreles, Leonardo Pantaleon, Gregory Reis
- List of Approved User Stories:

- User Story 010 - Properly upload, convert and store log file
- User Story 011 -Implement user sign up/login feature
- User Story 012 - Have user-uploaded missions be saved
- User Story 013 - Show interactive map of ecomapper trajectory
- User Story 014 - Landing page for users to sign up/login
- User Story 015 - Take out EcoMapper for real-world tests
- List of Rejected User Stories:
 - N/A

Senior Project Team 3 Sprint 4 Review 10/25/20

- Ariel Omar Arias, Orson Meyreles, Leonardo Pantaleon, Gregory Reis
- List of Approved User Stories:
 - User Story 015 - Take out EcoMapper for real-world tests
 - User Story 016 - Get feedback on current product
 - User Story 017 - Have user be able to update their password
- List of Rejected User Stories:
 - User Story 018 - Explore setup for the Clearpath Heron
 - Many hiccups with this, such as setting up ROS on our machines

Senior Project Team 3 Sprint 5 Review 11/8/20

- Ariel Omar Arias, Orson Meyreles, Leonardo Pantaleon, Gregory Reis
- List of Approved User Stories:
 - User Story 015 - Take out EcoMapper for real-world tests
 - User Story 016 - Get feedback on current product
 - User Story 019 - Set up dashboard page
- List of Rejected User Stories:
 - N/A

Senior Project Team 3 Sprint 6 Review 11/22/20

- Ariel Omar Arias, Orson Meyreles, Leonardo Pantaleon, Gregory Reis
- List of Approved User Stories:
 - User Story 015 - Take out EcoMapper for real-world tests

- User Story 020 - User can map their own path and download coordinates
- List of Rejected User Stories:
 - User Story 021 - Host web application
 - Tried one hosting service but didn't work out. Will complete in next sprint

Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents

Installation, user guide and shortcomings can be found in the following videos, along with other parts of this deliverable.

- Installation Guide: <https://youtu.be/u7c9BVgNuB4>
- User Guide: <https://youtu.be/vu95TM3S19A>
- Shortcomings: <https://youtu.be/aVxX-63-ybI>