Test Results Report

TraViz - Matthew Varona, Gopika Krishnan, Arshiya Khattak

Once the TraViz web app was completed, the team found a minimal amount of bugs through testing (only 2 minor ones).

We attribute the lack of bugs to the team following SWE best practices during the construction phase. Despite the fact that different group members focused on different aspects of the program, putting these parts together went smoothly because of design principles. For example, the data visualization engine for the website followed polymorphism, because the various possible modes of transport were incorporated into a single flexible Trip class that could dynamically change its behavior. This removed the need to hard-code different outcomes when the user inputs different transportation modes, and thus reduced the probability of bugs stemming from edge cases.

These were the bugs encountered during testing:

1. **Incorrect Map Visualization**

Description: After submitting a trip, the visualization would display the new trip, but the locations on the map would be incorrect (e.g. a dot for Berlin would appear in the middle of Africa).

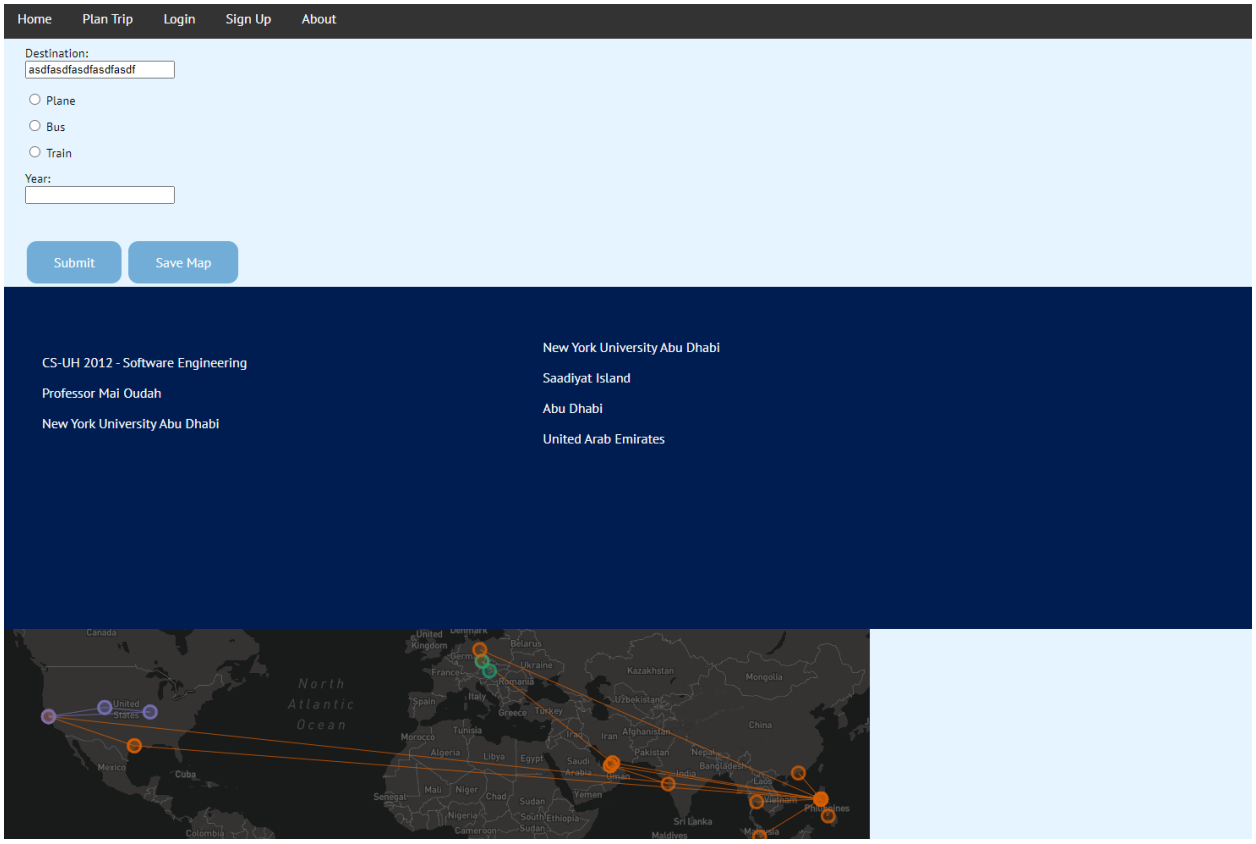
Test Case: The bug was discovered during Test Case 2.1 (DataVisualization-Valid), violating Step 4 (“Map is produced according to the input of the user and indicates them correctly”).

Error Analysis: To uncover the root of the problem, we attempted to eliminate possible sources of error. First, we noticed that pre-loaded trips (loaded from a .csv file for demo purposes) would be displayed correctly. Second, we checked the projection algorithm that converts latitude and longitude to x and y coordinates on the screen, and we verified that it computes as intended.

Finally, we checked the API query that is made to the LocationIQ API. The engine was correctly sending the location name to the API, and the API was indeed returning latitude and longitude. However, in the visualization code the latitude was being used as longitude (and vice-versa) due to typos in the code. Thus, the fix was simply to use the correct variables for the visualization.

2. **Misplaced Visualization**

Description: When opening the “Plan Trip” page, the map visualization would sometimes appear below the footer of the page, depending on the browser. It would usually work on Chrome, but not on Safari or Firefox. (See image below)



Test Case: The bug was discovered during Test Case 1 (WebsiteResponsiveness), violating Step 3 (“Page loaded properly and is presented as desired”).

Error Analysis: We deduced that this was an issue with p5.js, which is the Javascript library used to generate the visualization. After researching about p5.js, we realized that the visualization is created as a JavaScript Canvas object, which can be placed inside HTML elements. As such, we set the parent of the Canvas object to an HTML div, and applied proper CSS styling such that the visualization would appear in the correct position above the website footer.