Design Document

Architecture

The elevation-based navigation system will follow an MVC(S) architecture.

In terms of EleNa, we do not store or modify any data. Instead, we call GoogleMaps APIs (DirectionsAPI and ElevationAPI) with given source and destination inputs. The Service component is responsible for defining and calling the related GoogleMaps services.

The View component is responsible for rendering the user interface for displaying the map, inputting start and end locations, and visualizing the calculated routes.

The Controller component is responsible for handling user interactions, such as receiving input from the user for start and end locations, invoking the appropriate services for route calculation and elevation data management, implementing and invoking different shortest path algorithms and updating the View accordingly.

Tech Stack

The following tech stack is used for the elevation-based navigation system:

1. Front-end: ReactJS
2. Back-end: Java
3. API: Google Maps APIs

System Overview

EleNa application consists of two main parts: Frontend design with ReactJS and backend implementation in Java.

GoogleMapsService Component: getRoutes() method which calls the DirectionsApi of the GoogleMaps services is implemented by this component. For each user source-destination input, the service is invoked and returns all routes (alternatives is set to True) to the controller object.

GoogleMapsController Component: The application's home page allows users to input source, destination and x values. After appropriate input validation operations, the steps below are taken:

- GoogleMaps DirectionAPI is called for source-destination inputs and the returned results stored in the variable called “routes”. *(get_routes method implemented in the controller)*
- For each unique step in any of these routes, two new vertices are created (start and end location). Edge weight between these vertices are calculated based on the distance in meters.
- A graph is generated from this vertex/edge set using the JGraphT library. This graph is used as input for Dijkstra and BellmanFord implementations.
- For each route found with Dijkstra and BellmanFord as well as DirectionsAPI, minimum and maximum elevation along the route is calculated and displayed to the user. *(get_elevation method implemented in the controller)*

The front-end has 3 pages in total.
The first page takes input from the user. The start location, end location and the x value are the inputs. “Search possible routes” button renders the user to page 2.

The second page shows all the possible routes on the left side of the page in a map. On the right side, information about the routes such as distance, time, minimum elevation and the maximum elevation of the route will be displayed. When the user clicks on any route information, the third page will be rendered.

The third page gives the user a chance to look at all the reviews and also add a review of his own. The user can enter the review and click the “add review” button to enter his/her review.