List of team members:
Mikołaj Hościło
Filip Szymczak

VRP Solution Visualiser

The Project will be constructed in order to fulfil the following use cases:

- As a delivery company owner, I would like to see the optimal route for my fleet to take in order to minimize the distances and maximize efficiency.
- As a delivery company driver, I would like to know the best path to take while delivering packages.

Functional and Non-functional Requirements

Functional Requirements

The Application will be launched as a web service.
The User will be able to add any number of points in order to check the best path between the points. The points will be stored in a database. The resulting path will be displayed on a map. The calculated path will be saved in a database to be worked with later with standard CRUD operations. Any User can log into their account on the service, as well as create a new account to be used later, and save already created paths as well as view paths created previously.

Non-functional Requirements

Non-functional requirements will be listed in accordance with the FURPS+ classification:

- Functionality:
  - The complete functionality was described in the Functional Requirements section.
- Usability:
  - The app will possess a graphical interface (map) which will accurately represent the calculated path.
  - In addition to the graphical representation of the path, some additional information will be returned (i.e. length of the path).
  - Supplier can delete the delivered package from his map and mark the package as delivered on the database.
- Reliability:
  - Any unexpected interruptions will be resolved by the application itself.
  - In case of crash of the application on the user side, they may refresh the page normally
- Performance:
  - The application will be able to be run on any browser.
  - All data will be kept in databases by the server.
- Supportability:
  - In cases of malfunction, it will be possible to contact us.
- +:
  - No personal information will be collected.

- Adding the possibility to load several packages from JSON format to the database.

Optional features:

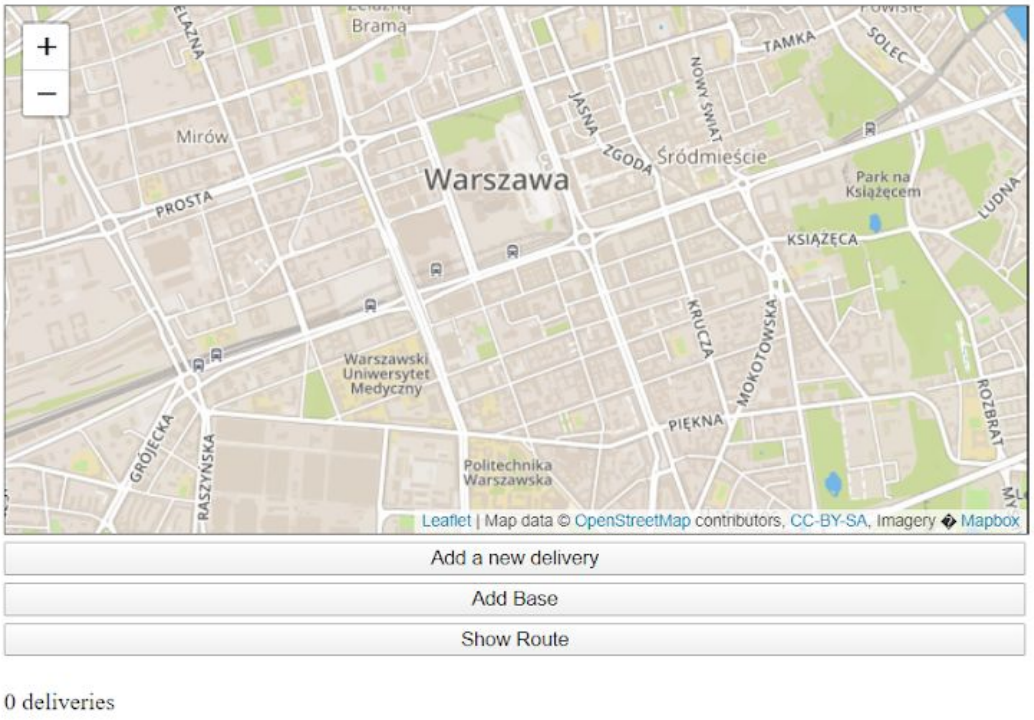
- Include creation of several routes for some amount of suppliers from the same company base.
- Include the view of just adding new packages to the base providing the package address, addressee and other info of the package and seperate that view from the view of supplier who can just state which packages were already delivered.

Architecture

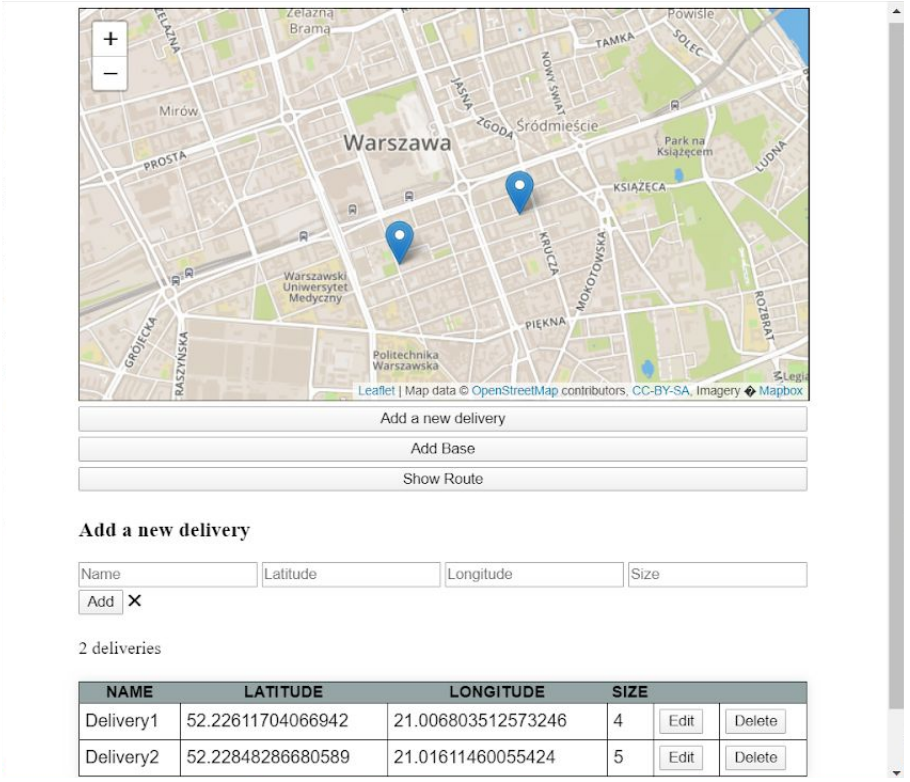
The backend of the application will use the Open Source Routing Machine in order to find the physical path between two points. This API will only be used to generate path, all other calculations will be done by the server.

All the data will be kept in databases, from the provided points, through user data, to the calculated paths.

The Application will look similar to the picture below:

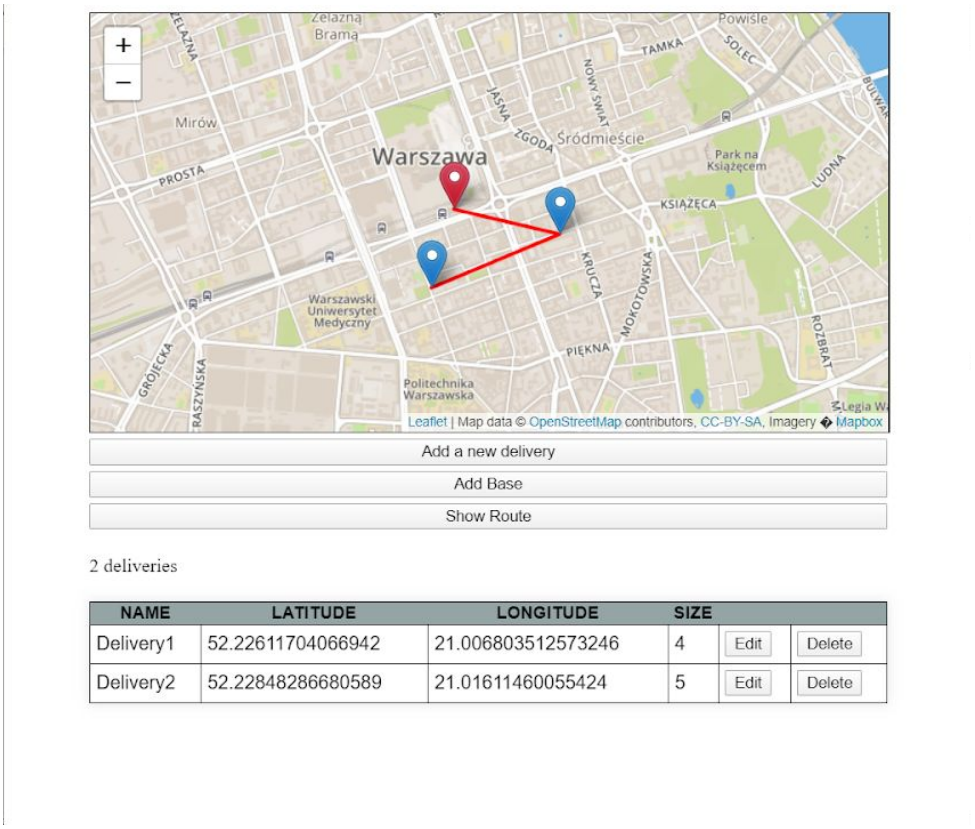


The map is interactable: a marker can be placed directly on it, which will be then stored in the database. The stored markers are visible on the map.



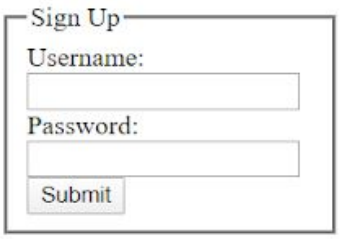
A marker can also be placed by writing in the coordinates. Optionally, it might be placed by specifying address by text.

A ‘base’ (red marker) is the delivery centre of interest. All packages will be delivered to and from the base.



As of right now, the path generated is random, although in the finished program the path will be optimised and properly routed through actual streets.

The area for signing in/up will look similar to this:



After signing up the user will be presented with some choices:

Additional utility and functionality will be added throughout the completion of the project.

The visual aspect of the application is subject to change.