Projektprotokoll

Kastrati, Steinbrugger

Technical steps and decisions

We started off using the demo-project provided in Moodle to create our backend.
We kept the structure of persistence-layer, service-layer and presentation-layer. The persistence is implemented by Jakarta and PostgreSQL, the service-layer uses Spring and the presentation-layer is built in React.
The communication between persistence and service-layer is handled by Data transfer objects, the communication between service-layer and presentation-layer is handled by a REST API that uses http-requests (as seen in Postman collection).
Separating the layers allows changes to be made without affecting the rest of the project a lot. We focused (also due to a shortage in time) mainly on the backend structure, rather than on the UI, which explains the Spartan style.

The communication with openrouteservice to retreive addresses as well as directions is managed by MapApiImpl, where the requests and calculations happen. Interacting with openrouteservice for addresses, directions, distances and time estimations was managable, but we did not succeed in retrieving the map tiles, so we had to focus on other functionalities due to our poor time management.

Unit Tests

The unit tests check test the functionalities of the service-layer because this is the most critical part of the program.
We assumed that there are not as many possible mistakes in the other layers and the communication channels between them.

Time spent

We forgot to track the time so our estimation is that we spent about 100 hours on the project.

Link to GitHub

https://github.com/sinkas123/Tourplanner/tree/main