Assignment 3

Map Awareness

Course: Mobile App Development, WiSe 2025/26
Group name: Map Awareness
Pratik Kumar: general UI
Marvin Dorsch: Routing
Gabriel Glaser: Location & Warnings and UI & UX improvements

Routes

The Routes screen presents two options to choose from, the first is to plan a route, and the second option is to switch to the already saved routes. Calculating a new route is kept simple it only requires a starting location and a destination. The route is being calculated by an API called "Graphhopper", which first converts the input of the starting and ending points to longitude and latitude coordinates using the "Geocoding" services. Then using the "Routing" services it calculates the Route. The API returns a collection of Polyline points and street names. With the street names, we can determine which "Autobahn" is on the route. With the names of the "Autobahn" we can use the "Autobahn API" to get a list of all the roadworks on the Autobahn. Using the Polyline coordinates and the coordinates of the roadworks from the API we can filter out the roadworks which are not on the route. Once the route is calculated it can be expanded to the see a list of widgets which represent the collection of roadworks on the route.

The Routes Screen also shows a quick summary of the Weather conditions and Weather warnings.

Adding a Gemini API key grants the option to generate an AI summary on the Weather and Routes conditions.

The last button on the screen is a redirection to the Map screen.

The application also filters external data to display infrastructure such as electric vehicle charging stations and parking areas along the active route. Additionally, users can save, load, and delete specific routes using persistent local storage.

Warnings and Locations

The Warning screen shows all Weather warnings from the "Deutscher Wetter Dienst" API in a specified Location or using the phones GPS data on your current location in a radius from 1 to 100km.

We also provides the option to get an Summary from Gemini, once the user provides an API key.

The weather warnings are color coded from red as storm warnings and similar dangerous warnings to orange for less dangerous warnings.

In addition to weather data, the application aggregates Civil Defense warnings from NINA via ARS lookup. It displays environmental metrics including current temperature, wind speed, precipitation, air quality (AQI, PM2.5, PM10), and river discharge data for flood risk. Users can filter warnings based on severity levels and event categories. The application also includes functionality to save, load, and delete specific locations for quick access.

Map

If the user selected a route, the map will show the route and the roadworks along it. Otherwise it shows the radius from the Warning screen.

Users can interact with the map by tapping to select start and destination locations. The interface includes navigation controls for zooming and re-centering. Markers are rendered for roadworks, charging stations, parking areas, and warnings. Selecting a marker opens a bottom sheet displaying comprehensive details about the specific item.

Application Configuration

The application supports theme management, allowing users to switch between Dark mode, Light mode, and System default settings. It includes an input field for managing the Google Gemini API key and displays privacy policy information.

Technical Capabilities

The application uses Riverpod for state management. Data handling involves parallel execution for API requests and implements HTTP caching strategies to ensure resilience. The user interface provides feedback through toast notifications and haptic feedback.

Challenges

The first problem was calculating the route and getting the coordinates from the starting and end points. But thanks to Graphhopper calculating the route was resolved quickly, and using the Geocoding service it also cleared the problem of getting the coordinates.

We also ran into some issues in GitLab, due large changes and no clean project structure at the start.