COMP.SE.110 [2022-2023] – Group 6

Software Design Documentation

Introduction

The purpose of this document is to describe how our group has planned to implement the group assignment application. The document will go through requirements, the structure and most important components and interfaces of the software as well as the reasonings for our decisions. A prototype of the application will be submitted alongside the document.

Table of contents

1. Requirements

- Visualization window
    - Timeline (week, day, hour view), history 24h & forecast 12h
    - Coordinates
    - Traffic data type
        - Maintenance
        - road condition forecast of 2, 4, 6 or 12 hours
        - Messages
        - Select items of interest
            - Visibility
            - Friction
            - Precipitation
            - Winter slipperiness
            - Overall road condition
    - Weather conditions
        - must be able to see at least the following and combine them in some way
            - Temperature
            - Obs. Wind
            - Obs. Cloudiness
            - Pred. Wind
            - Pred. Temperature
        - must be able to request calculations and visualization on average daily temperature at certain location in certain month
        - must be able to request calculations and visualization on maximum and minimum daily temperature at certain location in certain month
    - Adjust the parameters of the visualization
        - Timeline
        - Location
        - Selected weather information
    - Road information integration
        - Must be able to combine weather information with:
            - Road maintenance data
            - Road condition forecast information
        - Data queries done with coordinates (minimum of 5 predefined locations OR give free coordinates)
        - User must give a timeline (in a user-friendly environment)
    - Ability to save data sets and produce visualizations, capability to compare current and saved data sets
    - Ability to save preferences for producing visualizations and fetching those preferences will produce a visualization using the most recent data with the given parameters
    - The design must be such that further data sources, e.g., Statistics Finland or additional data from existing sources could be easily added
- Bonus
    - Ability to save visualisations as images
    - Ability to choose from several plotting options
    - Addition of a third data source (with meaning)

2. User Interface

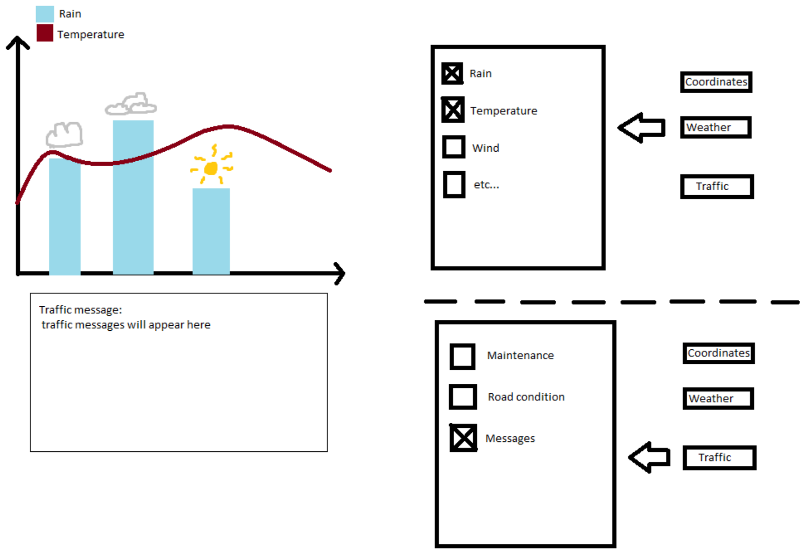

Figure 1: Sketch of data visualization implementation

The user interface consists of the visualization graph on the left side and different menus on the right side of the interface. The menus are hidden before the user clicks on them. From the menus user can modify what kind of data is displayed on the graph.

Reasonings

We went with a streamlined approach for the application. We tried to approach the requirements with the mindset of trying to fulfil at least the wanted base requirements. We did our platform decisions based on past experiences, such as choosing Java and JavaFX, as we felt that we could perform in the most optimal way with these tools when dealing with graphical user interfaces. The goal was and is to deliver the requirements in an optimal, timely manner. Our design choices were also inspired by our earlier studies, going for a one main viewport, where we then can display all the needed data in intuitive and compact manner, while avoiding overly complex menu and submenu structures. Everything is clearly labelled, and the user has control over what he/she wants to visualize.

3. High level description

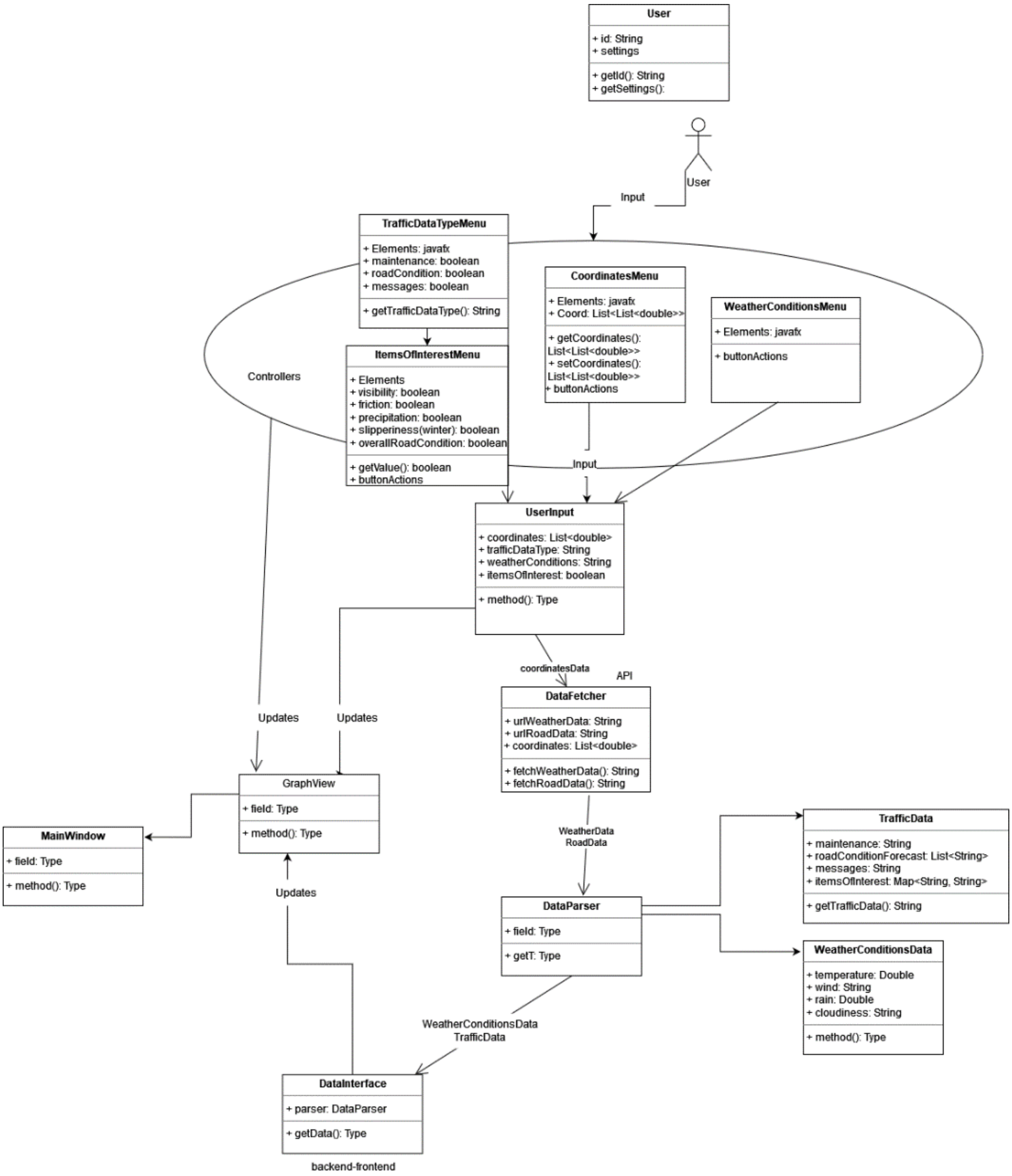

Figure 2. UML diagram of the application.

ClassDiagram.drawio - Google Drive

4. Boundries and interfaces (internal)

MainWindow

Is the main highest level-window, where our graphics are rendered in, this is the view that users will be seeing as the ‘app’.

GraphView

Is responsible for generating our graphs onto MainWindow from the gained data, gets UserInput boolean values to decide which things to display/render.

DataInterface

Is the end-receiver of all data, that will be then utilized in the visualization.

DataParser

Parses data when need be. Outputs (through accessors) two JSON trees of TrafficData and WeatherConditionsData and then relays both to DataInterface.

TrafficData

Fetched (and parsed) traffic data in JSON.

WeatherConditionsData

Fetched (and parsed) weather data in JSON.

DataFetcher

Fetches the data from the given APIs.

UserInput

Relays all the user-based data from menu interactions.

User

Stores all relevant data related to user.

Menus inside the app in general have the same interaction of filtering the data that user wishes for. The menus are categorised into two larger menus, where you can find all the fetched weather condition and traffic data items of interest. All of the data fetched is represented in menus as checkbox -esque items, where from you can pick and choose the ones you wish to see visualized on the viewport.

TrafficDataTypeMenu

Responsible for shown traffic data and its filtration, user chooses which data he/she wishes to see or not see with boolean variables.

ItemsOfInterestMenu

Submenu of TrafficDataTypeMenu, handles filtration with items of interest items, user chooses which items he/she wishes to see or not see with boolean variables.

CoordinatesMenu

Coordinates is the base data we want and need for fetching (specific) weather data for a certain area from the API. At base level the coordinates menu will be handling at least the pre-defined 5 locations of a list, the chosen from coordinates menu interaction are utilized in queries.

WeatherConditionsMenu

Responsible for shown weather conditions data and its filtration. User chooses which data he/she wishes to see or not see with the related boolean variables.

5. Timeline

Week 0 [5.9. - 9.9.]

- Group formation
- Requirements definition
- Start of Figma sketch prototype

Week 1 [12 - 16.9.]

- Figma sketch of the general UI

Week 2 [19. - 23.9.]

- Figma sketches of separate UI (requirement) scenes
- Start of documenting design choices

Week 3 [26. - 30.9.]

- Completed Figma sketch (our prototype)
- Completed design document

6. Prototype

The link to the prototype is also accessible from GitLab's Readme file.

https://www.figma.com/file/MgLkycQuzlvqfoukqI4FAX/First-draft?node-id=39%3A252

7. Group members

Arttu Lehtola – arttu.lehtola@tuni.fi

Vilma Lahti – vilma.s.lahti@tuni.fi

Mikko Moisio – mikko.moisio@tuni.fi

Aleksi Hasu – aleksi.hasu@tuni.fi