Frankfurt University of Applied Sciences

– Faculty of Computer Science and Engineering –

Real-Time Traffic Simulation with Java

Milestone 1 Documentation

Module: Object-Oriented Programming in Java
Winter Semester 2025/2026

Submitted on 27.11.2025 by

Salaheddine Mabchour

Ma Yiyuan

Gradwohl Laura

Mauricio de Souza Hilpert

Elias Heß

Source Code Repository: *https://github.com/33meow/RealTimeTrafficSimulation/tree/main*

Lecturer: Prof. Dr.-Eng. Ghadi Mahmoudi

1. Project Overview

Project Goal:

This project aims to create a a real-time traffic simulation platform with the aid of the mobility engine SUMO that is controlled via the TraaS Java API. By the end of the project the user should be able to use a GUI that visualises a basic road network and basic real-time user interaction with it, more specifically vehicles (visualisation on a 2D map and optional 3D views) and traffic lights.

A possible real-life use case for this project could be stress tests to plan new crossroads in bigger cities to avoid congestion. Another possible use case might be the growing demand for smarter cities that are build to make transportation accessible for a lot of people since real-time traffic simulation programs make it possible to plan and scale designs in a easy way.

Milestone Goal:

The goal for the first milestone is a working SUMO connection, a mock-up for our GUI and the documentation necessary for a software project as according to the requirements specified in the course.

Technical Setup & Usage:

To ensure the project runs smoothly on any machine without requiring path modification, we have structured the project using relative path and self-contained configuration folder.

Project Structure: The project follows a standard Eclipse Java project structure with a dedicated folder for SUMO configuration files:

Src/trafficsimulation: Contains all Java source code, separated into logical classes (Main, SimulationManager, SumoVehicle, MainFrame).

SumoConfic: a folder containing the simulation map and scenario files (osm.sumocfg, osm.net.xml.gz, osm.passenger.trips.xml).

Referenced Libraries: Contains the TraaS.jar library required for the TraCi interface

How to Run the Project

1. Import: Import the project folder into the Eclipse DIE as an existing Java Project.
2. Verify Build Path: Ensure that TraaS.jar is correctly referenced in Java Build Path (Classpath).
3. Run: Navigate to src/trafficsimulation/Main.java.
4. Execute: Right-click on Main.java and select Run As > Java Application.
5. Control: The application window will appear. Click the „Start"button to launch the SUMO GUI and begin the simulation.

Project Organisation and Challenges:

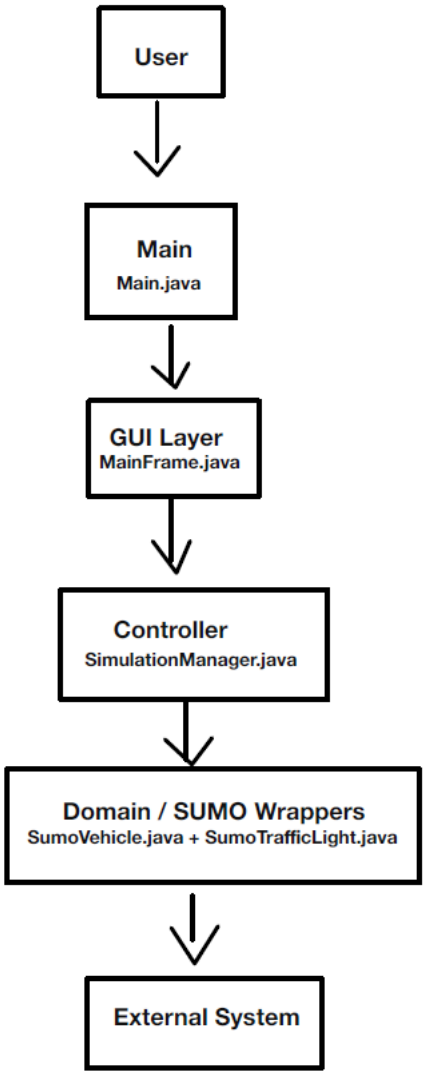
The personal goal of our first steps was to make sure that SUMO was running on every operating system that the group members are using and to create a git repository to share our work.
It was a challenge to connect SUMO with the IDE and we did a lot of that process at home while we shared our experiences during the time in the exercise groups and via messenger.
We wanted to have SUMO connected by the time of the second exercise, which we unfortunately did not accomplish, as we underestimated that step and miscalculated the time we would need. It

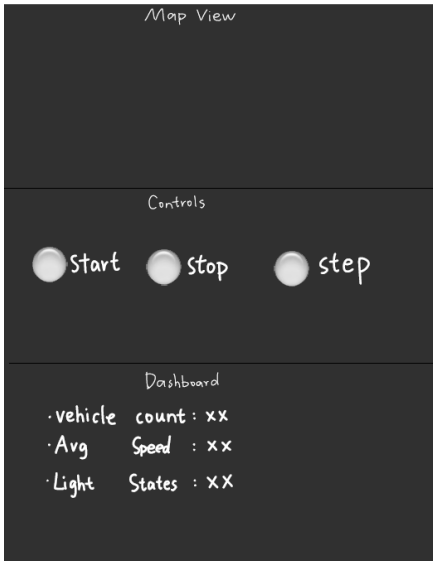
helped that we saw that others were struggling at that point as well.
It was also a factor that we all use different operating systems and therefore had to work out a way that everybody could use her or his preferred setup.
With regards to our roles in the project, Yiyuan was a good coordinator in the group and keept track what we had to do and how much time we have for the tasks. Salaheddine uploaded his code in the repository and helped those, that hadn't connected SUMO by now.
Laura did connect the group members and wrote the project overview and initiated more exchange online, since she was absent for some time due to illness.Elias connected all the parts together and set up an online exchancing room for even better comunication.

Next steps:

The next steps are another consultation as group before the presentation and then working on the next milestone goals and to have the core features running and to start with our code documentation.
We are also going to re-evaluate our roles in the team and optimise our ways of communication, to be a bit more independent from the time in the exercise.

2. System Architecture





Declaration of Authorship

I hereby declare that the submitted project is my own unaided work or the unaided work of our team. All direct or indirect sources used are acknowledged as references. I am aware that the project in digital form can be examined for the use of unauthorized aid and in order to determine whether the project as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future projects submitted. Further rights of reproduction and usage, however, are not granted here. This work was not previously presented to another examination board and has not been published.

Full Name:	Date:	Signature:
Mabchour Salaheddine	27.11.2025	*Mabchour Salaheddine*
Ma Yiyuan	27.11.2025	*Ma Yiyuan*
Gradwohl Laura	27.11.2025	*Gradwohl Laura*
Mauricio de Souza Hilpert	27.11.2025	*Mauricio de Souza Hilpert*
Elias Heß	27.11.2025	*Elias Heß*