BZ 214 Visual Programming Project Report

Group No: 28 **28/05/2025**

Number: 1030510731	**Name Surname: Özge Nur Kök**
Doing requirements analysis, designing SimulationView screen, writing project report and simulation controller	
Number: *1030510730*	**Name Surname: Fadimana Nisa Öztürk**
Doing requirements analysis, preparing Use-case diagram and use case descriptions, orginizing project structure and packages, writing model classes and WelcomeView controller, InputView controller and SimulationView controller, writing project report	
Number: *1030510740*	**Name Surname: Zehra Nur Erciyaz**
Designing WelcomeView and Inputview screen, designing and readme.md document	

Abstract

This project is a traffic control system designed to adjust green light durations at an intersection based on the vehicle density on each road. The main purpose is to minimize traffic congestion by dynamically changing the green light times. The project was developed using the Java programming language, with JavaFX utilized for the graphical interface. The UI was developed using JavaFX, and we followed MVC and OOP principles throughout the project. During the development process, Object-Oriented Programming (OOP) principles were followed, and the project architecture was structured according to the Model-View-Controller (MVC) design pattern.

Software Design

This project is developed based on the Model-View-Controller (MVC) architectur3e and follows Object-Oriented Programming (OOP) principles.

-Model Layer: The model layer contains the core data structures and business logic of the system.

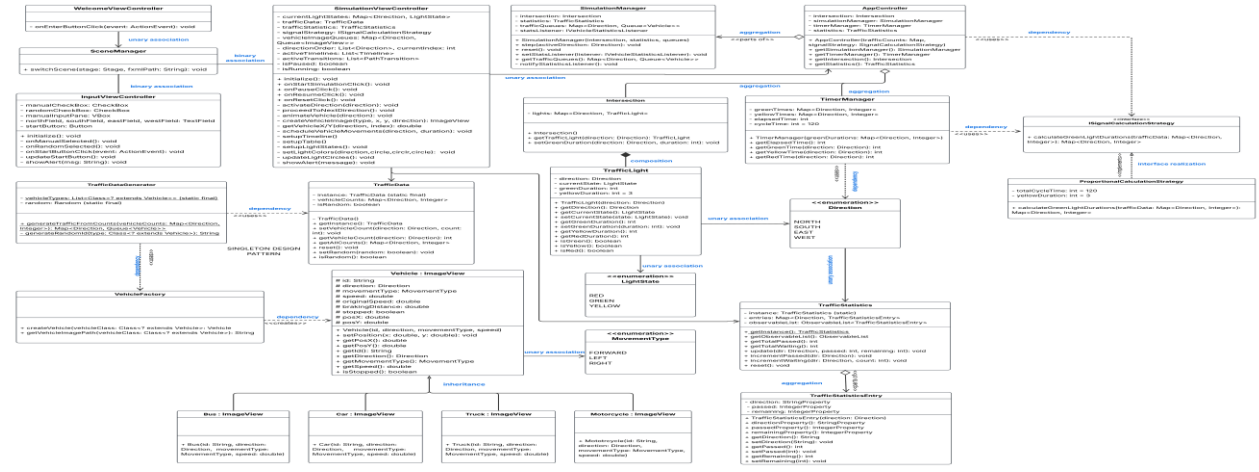

Image 1 : Traffic Light Control System UML Diagram (You can examine by using zooming in)

Classes are designed according to the encapsulation principle; all variables are defined as private and accessed through getter and setter methods. Inheritance and polymorphism principles are applied via the Vehicle class and its subclasses.

- View Layer :The graphical user interfaces are designed using JavaFX technology. Three main pages have separate FXML files:

WelcomeView.fxml which is the welcome screen displayed at application startup, InputView.fxml where the user selects between random or manual input and enters vehicle numbers, and SimulationView.fxml which displays the simulation visually.

-Controller Layer: Controller classes manage interactions between the user and the system. There is a separate controller class for each page: WelcomeController manages transitions on the welcome screen, InputController validates user input and transfers data to the model, SimulationController starts, stops, and updates the simulation.

Diagrams

-Class Diagram: Classes and their relationships are visualized with a class diagram which clearly shows the system architecture and interactions. The Factory Design Pattern is used to make vehicle creation flexible and to reduce code duplication.

-Use Case Diagram: Use case diagrams depict user-system interactions and main scenarios such as starting the simulation manually or randomly, entering vehicle counts, resetting, and stopping the simulation.

Interface Design

The JavaFX-based user interface is designed to be functional and simple. The welcome screen includes an image and a button directing to the input page. The input page includes checkboxes and text fields to easily get necessary information from the user. The simulation page visualizes the model using PNG images. A control panel with buttons and a table for tracking vehicle counts is added to allow easy management of the simulation. Lucid website was used to draw the Use-case and UML diagrams used in the project, and Canva and Unsplash websites were used for images , color palettes and icons.

Conclusion

As future development possibilities of the project, vehicle density can be increased and vehicle type diversity can be expanded. Different vehicle movement roues can be defied to diversify vehicle strategies. Improvements can be made using Strategy design pattern. Potential areas for future enhancement include the implementation of more dynamic vehicle movement algorithms, enrichment of graphical elements also can be added. Throughout the project, key competencies such as collaborative teamwork, effective time management, and foundational software architecture design were significantly strengthened.