PROJECT REPORT OF EXPLORATORY PROJECT (EP)

ON

Traffic Light Control System

submitted in partial fulfilment of the requirements for the award of degree of

BACHELOR'S OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Serena Balyan (2210992281)

Jinny Kapur (2210990462)

Khushi (2210991796)

Supervised By:

Dr. Gifty Gupta

Assistant Professor

Department of Computer Science & Engineering

Chitkara University, Punjab

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CHITKARA UNIVERSITY INSTITUTE OF ENGINEERING AND TECHNOLOGY

CHITKARA UNIVERSITY, PUNJAB, INDIA

CONTENTS

ACKNOWLEDGEMENT

It is our pleasure to be indebted to various individuals who directly or indirectly contributed to the development of this project and who influenced our thinking, learning and effort throughout the course of this study.

this Exploratory Project as a part of the curriculum and for offering us an academic environment that continuously supports learning and innovation.

We are extremely thankful to **Dr. Gifty Gupta**, Assistant Professor, Chitkara University, Punjab, for his constant support, cooperation, motivation and valuable guidance throughout the duration of the project. Her encouragement, expert suggestions and constructive feedback played a crucial role in helping us successfully complete this work.

We also extend our sincere appreciation to all the trainers of **BridgeLabz** for their continuous support, valuable insights and assistance during the execution of our project, which helped us improve the quality of work and deepen our understanding.

Lastly, we would like to express our heartfelt thanks to the almighty and our parents for their moral support, encouragement and blessings. We are also grateful to our friends for sharing their suggestions and companionship throughout this journey, which helped us refine our work and stay motivated.

Serena Balyan	Jinny Kapur	Khushi Dawar
2210992281	2210990462	2210991796

DECLARATION

I hereby certify that the work being presented in this project report entitled **"Traffic Light Control System"**, submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering (Computer Science and Engineering)** at **Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India**, is an authentic record of our own work carried out under the supervision of Dr. Gifty Gupta, The matter presented in this report has not been submitted to any other institute/university for the award of any degree.

Place: Rajpura

Date: _____

Signatures of Students :

Serena Balyan (2210992281)	Jinny Kapur (2210990462)	Khushi Dawar (2210991796)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge and belief.

Dr. Gifty Gupta

3. ABSTRACT

Rapid urbanization and the increasing number of vehicles on roads have made efficient traffic management a critical challenge. Traditional traffic signal systems often operate on fixed timing mechanisms, which may not adapt well to varying traffic conditions or emergency situations. This project focuses on the development of a **Traffic Signal Control System Simulation** using **Java and Swing**, aiming to demonstrate a flexible, interactive, and realistic traffic light control mechanism.

The proposed system simulates the operation of a standard three-light traffic signal—**Red, Yellow, and Green**—with configurable timing logic. The system dynamically adjusts the duration of the green signal based on **traffic density levels** (Low, Medium, and High), thereby reflecting real-world traffic flow variations. This adaptive behaviour improves traffic efficiency and reduces unnecessary waiting times at intersections. The simulation also incorporates an **emergency mode**, during which the signal switches to a blinking yellow light, representing situations such as emergency vehicle passage or signal malfunction.

The application is designed using **object-oriented programming principles** and follows a structured separation of concerns similar to the **Model–View–Controller (MVC)** architecture. The model manages traffic logic and signal states, the view handles graphical representation, and the controller processes user interactions through an intuitive control panel. A dedicated animation thread ensures smooth signal transitions and accurate countdown timers without interrupting the graphical user interface.

By utilizing **Java Swing components**, **multithreading**, and **event-driven programming**, this project provides a realistic and interactive simulation environment. The system serves as an effective educational tool for understanding traffic signal operations and software design concepts. Furthermore, it can be extended in the future to include features such as sensor-based input, real-time data integration, or AI-driven traffic optimization, making it a strong foundation for advanced intelligent transportation systems.

4. INTRODUCTION

Traffic congestion has become a major concern in modern cities due to rapid population growth, urbanization, and the continuous increase in the number of vehicles. Inefficient traffic signal control not only leads to longer waiting times and fuel wastage but also increases air pollution and the risk of road accidents. Conventional traffic light systems generally operate on fixed time intervals, which do not adapt to real-time traffic conditions or unexpected situations such as emergency vehicle movement. As a result, there is a growing need for smarter and more flexible traffic control mechanisms.

This project, **Traffic Signal Control System**, is developed to address these challenges at a conceptual and educational level. The system simulates the working of a real-world traffic signal using **Java Swing**, providing a visual and interactive representation of traffic light behaviour. The simulation includes three standard signals—red, yellow, and green—and models their transitions based on predefined timing logic. Unlike basic fixed-timer systems, this project introduces **traffic density–based control**, allowing the green signal duration to vary according to low, medium, or high traffic conditions.

Another important aspect of this system is the inclusion of an **emergency mode**, which reflects real-life scenarios such as ambulance or fire brigade movement, traffic signal malfunction, or special traffic control requirements. In this mode, the system switches to a blinking yellow signal, alerting road users to proceed with caution. This feature enhances the realism of the simulation and highlights the importance of emergency handling in traffic management systems.

The project is implemented using **object-oriented design principles**, ensuring modularity, readability, and ease of future enhancement. The system architecture separates traffic logic, graphical rendering, and user controls, making the application well-structured and maintainable. Multithreading is employed to manage signal animation and countdown timers efficiently without affecting the responsiveness of the graphical user interface.

Overall, this project serves as a practical demonstration of how software-based simulations can be used to study and understand traffic signal control systems. It is particularly useful for students and beginners to gain hands-on experience with **Java GUI programming**, **multithreading**, and **real-time system simulation**, while also laying the groundwork for more advanced intelligent traffic management solutions in the future.

4.1 Background

Traffic signal control systems have traditionally relied on preset timers that operate in a cyclic manner regardless of actual traffic conditions. While such systems are simple to implement, they lack flexibility and often result in inefficient traffic flow, especially during peak and off-peak hours. In recent years, advancements in computing and sensor technologies have enabled the development of adaptive and intelligent traffic control systems that adjust signal timing based on real-time data.

In academic and learning environments, traffic simulations play a vital role in understanding how signal control algorithms work without the need for expensive physical infrastructure. Software simulations allow developers and students to model traffic behaviour, test different

control strategies, and visualize outcomes in a controlled environment. This project draws inspiration from real-world traffic systems and simplifies their functionality into a simulation that is easy to understand, modify, and extend.

4.2 Problem Statement

Conventional traffic signal systems operate on fixed time intervals and lack mechanisms to handle varying traffic densities and emergency situations effectively. This often results in unnecessary delays, poor traffic flow, and increased frustration among road users. Additionally, many educational traffic models fail to provide interactive control, real-time visualization, or support for emergency handling.

Problem Statement:
To design and implement a traffic signal control system simulation that dynamically adjusts signal timing based on traffic density and supports an emergency mode, while providing a clear visual representation and interactive user controls.

4.3 Objectives

The primary objectives of this project are as follows:

- To design a graphical traffic signal simulation using Java Swing.
- To implement a real-time traffic light control mechanism with red, yellow, and green signals.
- To introduce traffic density–based timing for improved adaptability and realism.
- To incorporate an emergency override mode that simulates real-life emergency traffic scenarios.
- To apply object-oriented programming principles for modular and maintainable code.
- To utilize multithreading for smooth animation and accurate countdown timing.
- To provide an interactive control panel for user input and system monitoring.
- To create an educational and extendable model that can serve as a foundation for advanced traffic management systems.

5.METHODOLOGY

The methodology of this project follows a systematic and structured approach to ensure that the traffic signal control system simulation is reliable, efficient, and easy to understand. The development process includes requirement analysis, system design, implementation planning, and testing. Each phase contributes to building a robust simulation that accurately models real-world traffic signal behaviour while maintaining software quality standards.

5.1 System Requirement Analysis

System requirement analysis identifies the functional and non-functional requirements necessary for the successful operation of the traffic signal control system.

Functional Requirements:

- The system shall simulate three traffic signal lights: **Red, Yellow, and Green**.
- The system shall allow users to **start, stop, and reset** the simulation.
- The system shall dynamically adjust the green light duration based on **traffic density levels** (Low, Medium, High).
- The system shall support an **emergency mode**, where the signal switches to a blinking yellow light.
- The system shall display a **countdown timer** for the active signal.
- The system shall provide a graphical user interface for user interaction.

Non-Functional Requirements:

- The system should be **responsive** and free from UI freezing.
- The application should be **platform-independent**, running on any system with Java installed.
- The system should be **modular and maintainable**, allowing future enhancements.
- The simulation should provide **real-time visual feedback**.

5.2 System Design

The system is designed using an **object-oriented approach** with a clear separation of responsibilities among components. The architecture closely resembles the **Model–View–Controller (MVC)** design pattern.

- **Model:** Handles traffic logic, signal states, traffic density, and emergency conditions.
- **View:** Responsible for visually rendering the traffic signal using graphical components.
- **Controller:** Manages user interactions such as button clicks and density selection.

The design includes multiple interacting classes to ensure scalability and readability. Multithreading is used to handle signal transitions independently of the graphical user interface, ensuring smooth animation and accurate timing.

5.3 Implementation Strategy

The implementation is carried out using **Java** and **Swing GUI components**. A step-by-step strategy is followed:

1. Define core entities such as traffic lights, density levels, and signal timing.
2. Implement the traffic logic within a dedicated model class.
3. Design the graphical interface using Swing panels and custom drawing techniques.
4. Implement an animation thread to control signal transitions and countdown updates.
5. Integrate user controls such as start, stop, emergency, and traffic density selection.
6. Ensure synchronization between the model and the user interface.

This modular approach ensures flexibility, code reuse, and easier debugging.

5.4 Testing and Validation

Testing and validation are essential to verify that the system behaves as expected under different scenarios.

Testing Techniques Used:

- **Unit Testing:** Individual components such as signal transitions and density-based timing were tested independently.
- **Integration Testing:** Interaction between model, view, and controller components was verified.
- **Functional Testing:** All user controls (start, stop, emergency, density selection) were tested for correct behaviour.
- **Boundary Testing:** Signal timing values and emergency toggling were tested at extreme conditions.

Validation:

- The system was validated by comparing simulated behaviour with real-world traffic signal logic.
- Emergency mode behaviour was checked to ensure proper blinking and alert indication.
- Timer accuracy and UI responsiveness were validated through repeated execution.

Successful testing confirms that the system meets all defined requirements and provides a reliable traffic signal simulation.

6. TOOLS AND TECHNOLOGY

The successful development of the **Traffic Signal Control System Simulation** relies on a combination of programming languages, development tools, and software libraries. Each tool and technology was carefully selected to ensure efficiency, portability, and ease of implementation.

6.1 Programming Language

- **Java**
  Java is used for its object-oriented features, platform independence, and built-in support for multithreading and GUI development.

6.2 Framework

- **Java Swing**
  Swing is used to design the graphical user interface and visually simulate traffic signals using custom drawing techniques.

6.3 Development Environment

- **Visual Studio Code (VS Code)**
  VS Code is used as the primary IDE due to its lightweight design, Java extension support, integrated terminal, and debugging features.

6.4 Libraries and APIs

- **Java AWT** for graphics, colors, and fonts
- **Java Multithreading** for real-time signal animation

6.5 System Requirements

- **Software:** JDK 8 or above, VS Code, any Java-supported OS
- **Hardware:** Minimum 4 GB RAM

7. IMPLEMENTATION

- The system is implemented using Java with Swing and AWT libraries for graphical user interface development.
- The main application initializes a window that integrates the traffic signal display and the control panel.
- Traffic signal logic is managed through a model that controls red, yellow, and green light states.
- Traffic density selection dynamically adjusts the green signal duration.
- An emergency mode is implemented to activate a blinking yellow signal.
- A separate animation thread controls signal transitions and countdown timers.
- Custom drawing techniques are used to visually render traffic lights.
- User actions such as start, stop, density selection, and emergency activation are handled using event listeners.
- The system ensures smooth real-time operation without affecting UI responsiveness.

8.RESULTS

- The system successfully simulates a **three-state traffic signal** (Red, Yellow, Green) with smooth visual transitions.
- Signal timing operates accurately with a **real-time countdown display**.
- Traffic density selection (**Low, Medium, High**) dynamically adjusts the green light duration as expected.
- The **emergency mode** functions correctly by switching the signal to a blinking yellow state.
- User controls such as **Start**, **Stop**, and **Emergency** respond instantly without UI lag.
- The graphical interface remains **responsive and stable** due to multithreading.
- The simulation closely reflects **real-world traffic signal behavior** in a controlled environment.
- The system runs successfully on **Visual Studio Code** with Java, without runtime errors.

9. CONCLUSION AND FUTURE SCOPE

Conclusion

- The Traffic Signal Control System Simulation successfully demonstrates the functioning of a real-world traffic signal using Java and Swing.
- Traffic density–based signal timing improves flexibility and efficiency compared to traditional fixed-time systems.
- The emergency mode effectively simulates real-life scenarios such as emergency vehicle movement.
- The project applies object-oriented programming and multithreading concepts, ensuring smooth and responsive system performance.
- The user-friendly graphical interface makes the system suitable for learning and demonstration purposes.

Future Scope:

- Integration of real-time traffic sensors for automatic density detection.
- Expansion to support multiple intersections and coordinated signal control.
- Implementation of AI-based traffic optimization techniques.
- Inclusion of pedestrian crossing signals.
- Development of a web or mobile-based version for wider accessibility.
- Use of IoT and real-time traffic data for intelligent traffic management.

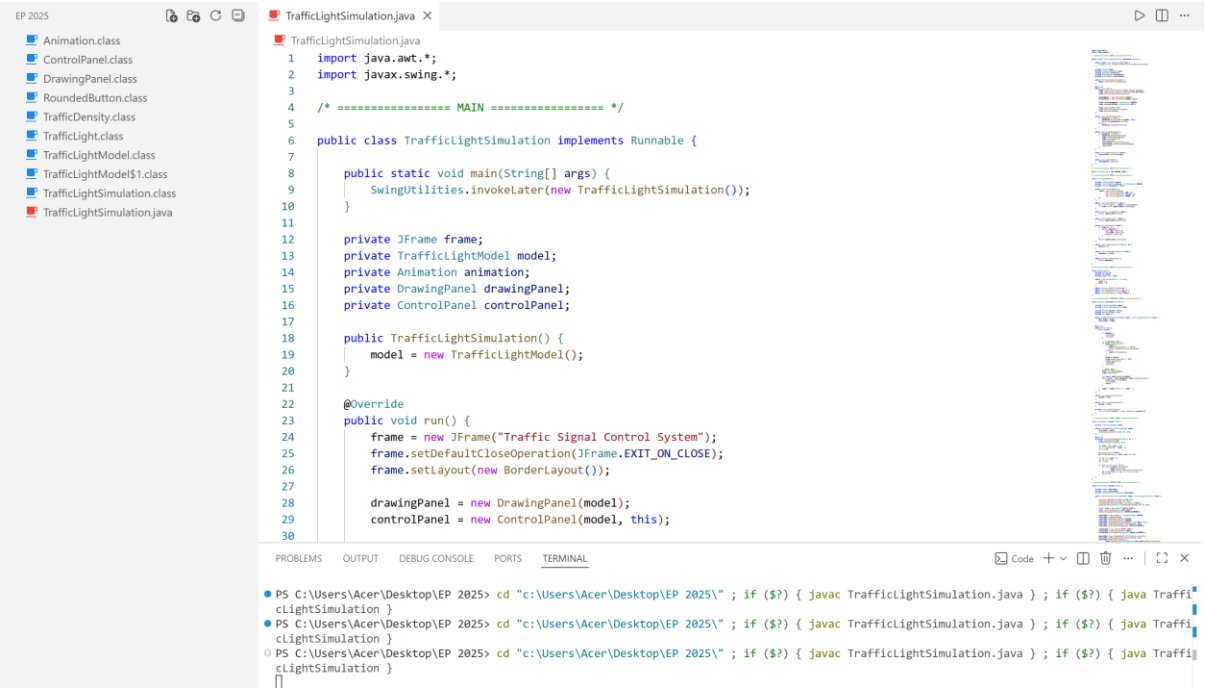
10. REFERENCES

- Oracle Corporation, Java Platform, Standard Edition Documentation.
  Available at: https://docs.oracle.com/javase/
- Oracle Corporation, Java Swing Tutorial.
  Available at: https://docs.oracle.com/javase/tutorial/uiswing/
- Oracle Corporation, Java AWT Documentation.
  Available at: https://docs.oracle.com/javase/
- Schildt, H., Java: The Complete Reference, McGraw-Hill Education.
- Deitel, P. and Deitel, H., Java How to Program, Pearson Education.
- Pressman, R. S., Software Engineering: A Practitioner's Approach, McGraw-Hill.
- Visual Studio Code Documentation, Java Development in VS Code.
  Available at: https://code.visualstudio.com/docs/java/java-tutorial

11.APPENDIX

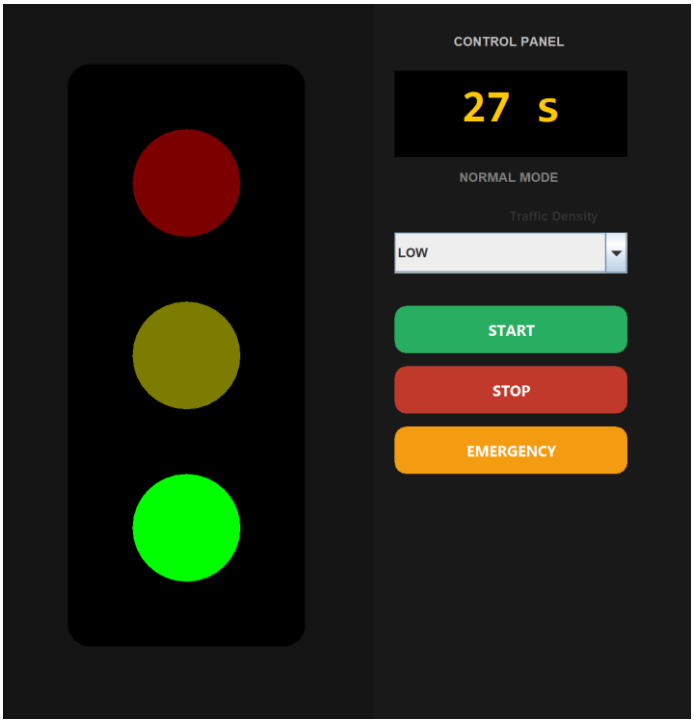
Appendix 11.1: Screenshot of Code Snippet

This screenshot shows key sections of the Java source code implementing traffic signal logic and animation control.



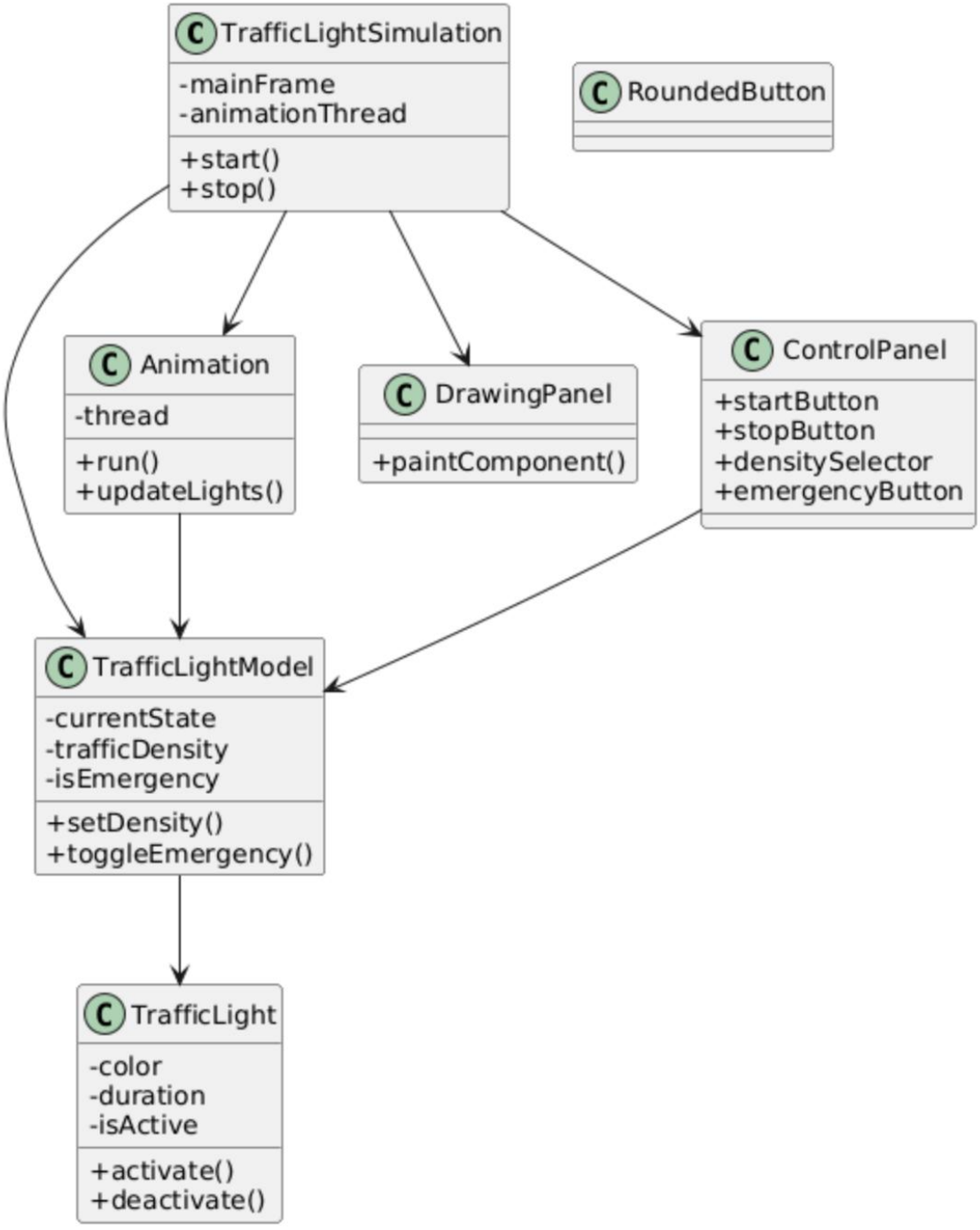
Appendix 11.2: Screenshot of User Interface

This screenshot displays the graphical user interface of the traffic signal control system, including the signal display and control panel.



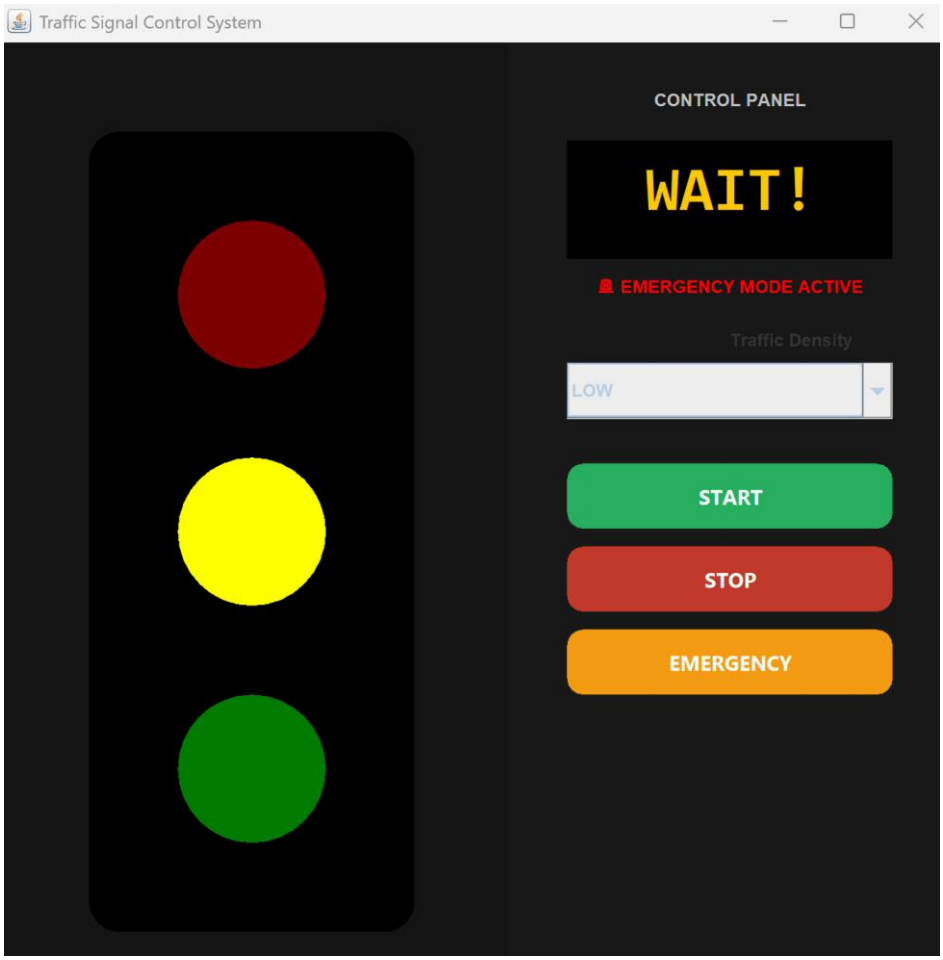
Appendix 11.3: Screenshot of UML Diagram

This screenshot presents the UML class diagram generated using PlantUML, illustrating the structural design of the system.



Appendix 11.4: Screenshot of Emergency Mode

This screenshot demonstrates the system behavior during emergency mode with a blinking yellow signal.



Appendix 11.5: Screenshot of Traffic Density Mode

This screenshot shows the effect of different traffic density selections on signal timing.