Abbottabad University Of Ssience And Technalogy

SOFTWARE REQUIREMENTS SPECIFICATION

(SRS DOCUMENT)

For

Traffic Control Simulation

Version 1.0

By

[Waleed Babar 14830]

Supervisor

[Sir Jamal Abdul Ahad]

Table of Contents

1. Introduction

1.1 Purpose

1.2 Document Conventions

1.3 Project Scope

2. Overall Description

2.1 Product Perspective

2.2 User Classes and Characteristics

2.3 Operating Environment

2.4 Design and Implementation Constraints

2.5 Assumptions and Dependencies

3. System Features

4. External Interface Requirements

4.1 User Interfaces

4.2 Hardware Interfaces

5. Quality Attributes

1. Introduction

1.1 Purpose

This document outlines the functional and non-functional requirements for the Traffic Control Simulation project. The simulation is designed to visualize traffic management and optimization through a Tkinter interface, incorporating algorithms like Dijkstra’s and animations via Matplotlib.

1.2 Document Conventions

Boldface: Highlights key terms or functions.

Italics: Indicates explanatory notes or additional details.

Code: Code elements are written in a monospaced font.

1.3 Project Scope

The system simulates traffic flow for a three-lane intersection with cars and traffic light controls. It uses Dijkstra's algorithm to optimize traffic based on lane congestion and handles emergencies by prioritizing emergency lanes.

2. Overall Description

2.1 Product Perspective

The project is an independent desktop application that uses Python libraries like Tkinter, Matplotlib, and NetworkX for traffic visualization and management.

2.2 User Classes and Characteristics

Traffic Engineers: Analyze traffic patterns.

Students/Researchers: Understand traffic optimization techniques.

2.3 Operating Environment

Hardware: Any system with at least 4GB RAM and Python installed.

Software: Python 3.8+ with necessary libraries.

2.4 Design and Implementation Constraints

Uses Tkinter for UI and Matplotlib for visualization.

Real-time data input is not supported; simulation is based on random data generation.

2.5 Assumptions and Dependencies

Assumes proper Python environment setup.

Relies on random data generation for car and traffic weights.

1. System Features

3.1 Traffic Light Control

The system manages traffic lights dynamically, prioritizing green lights for emergency or optimal lanes.

3.2 Car Positioning and Movement

Cars are simulated across lanes with randomized paths, represented visually.

3.3 Emergency Lane Prioritization

Users can input an emergency lane to prioritize traffic flow.

4. External Interface Requirements

4.1 User Interfaces

Input: Tkinter-based GUI for emergency lane entry.

Output: Animated traffic graph with visual cues for traffic lights and car positions.

4.2 Hardware Interfaces

Requires a standard keyboard and mouse for interaction.

5. Quality Attributes

5.1 Performance

The system ensures smooth animation updates with an interval of 1 second.

5.2 Usability

The interface is user-friendly, requiring minimal input to start the simulation.

5.3 Security

Local execution ensures data safety, with no external dependencies.

5.4 Maintainability

Code is modular, allowing easy updates or feature additions.

This document ensures clear understanding and provides guidelines for implementing the Traffic Control Simulation project efficiently.