CS261 Group 29 Requirement Analysis

Ani Bitri, Krister Hughes, Thomas Phuong, Eshan Sharif, Josh Turner, Antoni Zyla

January 2025

1 Introduction

Dorset Software has been contracted to create a system used for the modelling of traffic junctions based on various parameters. The system will provide data about how each of these configurations impacts the performance of a given junction and allow for the comparison of various sets of parameters to help determine the best configuration for a given junction. As a group, we have been given the task of developing this system, along with managing the project and creating documentation.

2 System Architecture

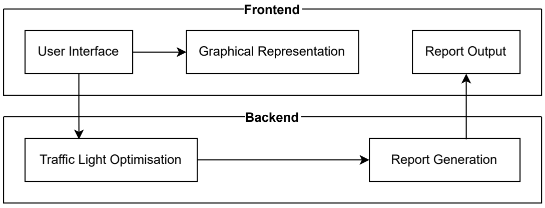

Figure 1: System Architecture Diagram

A high-level overview of how we intend the system to work is as follows: the user will enter the traffic flow rates for the simulation and select the configurable parameters for each junction configuration via the user interface, leading to the graphical representation displayed updating to reflect the user input. After pressing a button to start the simulation, the user input will be sent to the backend, where each junction configuration will be simulated and have its traffic light sequencing optimised based on the parameters restricting the order and the best values of the junction efficiency metrics. After all junction configurations have been processed, the data will be sent back to the frontend to generate a report of the efficiency of each junction configuration.

3 System Requirements Specification

3.1 Functional Requirements

- **Requirement 1 (Must)**
  - **C**: The user must be able to input the rate of traffic flow from each direction to each other direction via the User Interface (UI).
  - **D**: The system must accept traffic input from text fields, parse the input to determine if each traffic flow is a valid integer, and prevent the simulation from running if any are not.
  - **Verification**: Run the simulation with two valid sets of traffic configurations to check that the simulation both runs and responds to changes in traffic flow. Then attempt to run the simulation with an invalid set of traffic flow rates, which should not be possible.
  - **Traceability**: Input Parameters section in the specification
- **Requirement 2 (Must)**

  - **C**: The user will be able to adjust the number of lanes (between 1 and 5) and toggle left lanes, right lanes, bus lanes, and pedestrian crossings. They will also be able to configure the timings and frequency of pedestrians crossing.
  - **D**: For a given configuration the system will determine if it is a valid configuration and prevent the simulation from running if it is not.
  - **Verification**: run a simulation with default parameters (2 lanes per road entering the junction, equal priority on all lights, and all other settings set to off/No) as a reference, and then run a simulation per setting with that setting adjusted, showing the results of each simulation is different from the reference.
  - **Traceability**: Configurable Parameters section in the specification

- **Requirement 3 (Must)**
  - **C**: The user will be able to assign a priority value (From 0 which means no priority, to 4 which is the highest) to traffic entering the junction from certain directions.
  - **D**: If any priorities are invalid, then prevent the simulation from running. Otherwise, ensure that the simulation takes into account traffic priority.
  - **Verification**: run a simulation with no priorities, then compare it to another with the same layout but with some priorities. Wait times for cars coming from priorities directions should be reduced in the second simulation.
  - **Traceability**: Configurable Parameters section in the specification

- **Requirement 4 (Should)**
  - **C**: The user will be presented with a graphical representation of the junction based on the parameters they have entered.
  - **D**: The system must take the current configuration settings of the junction and generate a representation of the junction which mirrors those settings.
  - **Verification**: Junctions can be generated with all lane counts from 1-5, then with a left and right lane, a pedestrian crossing, and a bus lane. We will also check that lanes display correctly in each orientation.
  - **Traceability**: Output section in the specification - 'a simple graphical representation is a "nice-to-have"'

- **Requirement 5 (Must)**
  - **C**: The user must be able to create multiple junction configurations (as well as remove configurations), with the system simulating all of them when the run simulation button is pressed.
  - **D**: The system must allow for the user to create and remove multiple junction configurations, and when the simulation is run, all junction configurations must be simulated and their required results generated.
  - **Verification**: Create more than one junction configuration, press the run simulation button, and then verify that results are shown for all the created junction configurations. Remove a configuration, press the run simulation button again, and verify that the results are the same except the removed junction's results are no longer present.
  - **Traceability**: Output section and Configurable Parameters section in the specification.

- **Requirement 6 (Must)**
  - **C**: After running the simulation the user must be provided with the average wait time, maximum wait time, maximum queue length, and an overall score for each road entering the junction. This must be done for each simulated configuration.
  - **D**: When running the simulation, the average wait time, maximum wait time, maximum queue length must be tracked throughout. These should then be used to generate an overall score and all four should be output to the user. This should be done for each road entering the junction in each simulation.

  - **Verification**: run the simulation once with two different configurations or twice with a single configuration and different traffic flows, then verify that the expected metrics have been generated and that they align with each configuration.
  - **Traceability**: Output section in the specification

- **Requirement 7 (Must)**
  - **C**: After running the simulation, the user must be presented the efficiency metrics and score for each junction configuration side by side in a table or graph format.
  - **D**: The system must display the junction efficiency metrics and score of each junction side by side in a table or graph format.
  - **Verification**: Simulate two different junction configurations with the same input traffic flows and verify that the results are displayed in a side by side comparison.
  - **Traceability**: Output section in the specification.

- **Requirement 8 (Should)**
  - **C**: The user must be able to select whether cars drive on the left-hand side of the road or right-hand side, affecting the graphical representation of the junction configuration and the results of the simulation (junction efficiency metrics).
  - **D**: The system must take into account the setting determining which side of the road cars drive on, affecting the calculations of the simulation and how the junction configuration is displayed on the UI.
  - **Verification**: switching cars from driving on the left-hand side to the right-hand side and observing a change in the graphical representation, running the simulation, then switching cars back to driving on the left-hand side and running the simulation again, observing a change in the graphical representation and a difference in the junction efficiency metrics
  - **Traceability**: Constraints and Assumptions section in the specification, included as a unique selling point of our software

- **Requirement 9 (Should)**
  - **D**: The system must take into account the distance between the entrance and exit points of each junction configuration. So, for example, a left turn from the leftmost lane of a road has a shorter distance than a right turn from the leftmost lane of the road (assuming driving on the left-hand side of the road)
  - **Verification**: run the simulation on a specific example where at least one of the metrics should be affected when the distance is taken into account (e.g. one road has 100vph exiting right, preventing another road's 100vph from exiting to its own right and another road's 100vph from exiting ahead such that these two do not interfere with each other)
  - **Traceability**: Constraints and Assumptions section in the specification, included as a unique selling point of our software

- **Requirement 10 (Could)**
  - **C**: When the simulation is run, the user should be able to select a certain junction entrance and display the amount of traffic going into other exits via a Sankey diagram.
  - **D**: When the simulation is run, the user should be able to click on a junction entrance and see the amount of traffic flowing into the other exits from the selected entrance, where the volume flowing into an exit is proportional to the traffic flow through it.
  - **Verification**: Select a junction entrance and check whether the volume flowing is proportional to the associated traffic flows. Repeat for the other exits.
  - **Traceability**: while not directly specified, this requirement could be traced back to the Output section, where the customer wants to be able to "identify the most effective configuration of the traffic junction". By adding this, the user could see how the traffic flows into other exits via a graphical interface that will change depending on the user's inputs.

- **Requirement 11 (Must)**

- **C**: The user must be provided with user-friendly error messages whenever there is an invalid input or system failure. They must also be provided technical details so they or a colleague can debug any issues if desired.
- **D**: The system must provide user-friendly error messages for invalid inputs or system failures. These messages should clearly explain the issue in non-technical terms for the user, while also logging technical details for debugging purposes.
- **Verification**: Attempt to run the simulation with invalid traffic flow rate values, then verify that an error message is produced . Then simulate a system failure and confirm that a technical error message is logged for debugging, while the user sees a generic failure message.
- **Traceability**: Input section and User Experience / User Interface subsection of Considerations section in the specification.

3.2 Non-functional Requirements

4 Project Philosophy

4.1 Team Roles

Our team is comprised of six members with the following roles:

- Backend: Krister and Antoni
- Frontend: Eshan and Thomas
- Both: Josh and Ani

In addition to this, Ani will be largely responsible for planning the video presentation.

As a team, we have been meeting once a week on Wednesdays and will continue to do so until the end of the project, and will meet up at additional points through the term if we decide it to be necessary (e.g. to work through the design of a particular part of the project). When recording the Dragon's Den video presentation, we will allocate some more time than a regular meeting, as well as make sure that the person allocated to editing the recording is able to fully focus on the video to make it as good as possible.

Together, we have decided to forgo a project manager and opt for regular meetings with a shared understanding of the goals. If any team member has any concerns about the group structure or work distribution, then this can be brought up at the whole group meeting to everyone, since everyone is an equal on the team.