CS261 Group 29 Report

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

January 2025

1 Preface

Dorset Software tasked our team with creating a traffic junction simulator. And we fucking did it.

Dorset Software has tasked our team with the creation of a traffic junction simulator which will be used for the modelling of traffic junctions on various parameters. The system provides data on how these configurations affect the traffic flow and the overall efficiency of the junction in comparison to other configurations.

This document explains the development process of the system and provide an explanation to the key changes and decisions made throughout the implementation of the prototype. Furthermore, it will give an insight into the algorithms and formulas used to calculate the traffic flow and efficiency of the junctions.

2 System Overview

2.1 Purpose

Our system is a traffic junction simulator intended to be used by traffic engineers, urban planners and district governments to model and analyse the efficiency of different traffic junction configurations. The system includes a frontend interface for users to input their desired configurations, a simulation page to visualise the traffic flow and a results page to display the metrics of the simulation.

2.2 Developer Tools Used

The sytem was developed using Python and a few of its libraries. This allowed for easier transition between the frontend and backend of the system. The fronted was developed using PyQT5 for the GUI and MatPlotLib for aid in the Visualisation of the results.

Our team also utilised GitHub for version control and better collaboration between team members.

2.3 User Interaction

The system is designed to be user-friendly and intuitive. When the application first opens, the user is greeted with the Home page where it displays the overview of the application and have the option to either exit to application or go to the "Input Parameters" page. page where they can input the desired parameters for the simulation.

2.4 System Architecture and Interaction

3 Modifications

4 Algorithms and Formulas

4.1 Semaphores

4.2 Traffic Flow

4.3 Calculating Metrics

4.4 Overall Score

5 Frontend

5.1 User Interface Design

The previous interface design combined the inputs, simulation, and results onto a single screen where you could tab between inputs and parameters. While this worked, a few issues arose when implementing it, such as, cluttered visuals, reduced usability, and a limited possibility of expansion. To address these newly found issues, instead of tabbing between the inputs and parameters pages, we made it so the inputs and parameters, the simulation, and results were given their own unique tab to function on. As a results of implementing the project this way, we found key advantages present in this layout compared to the previous one.

5.1.1 Improved Organisation

By separating the inputs, simulation, and results into distinct tabs, users can focus on one aspect of the interface at a time. This tabbed approach also enhances the workflow efficiency of the user and gives the user a natural progression to follow between data entry to simulation execution and results analysis.

5.1.2 Reduced Clutter

The previous design displayed all elements of our user interface on one screen, making it visually overwhelming and difficult to navigate. The new design split of the interface, allowing the user to interact with each section independently, reducing the visual overload and improving the overall ease of use.

5.1.3 Increased Space for Additional Features

The old layout restricted the ability of add new functionalities due to space constraints as all three segments were on one page. By utilising the tab-based system, we have created for room for additional features such as adding graphs to the results page.

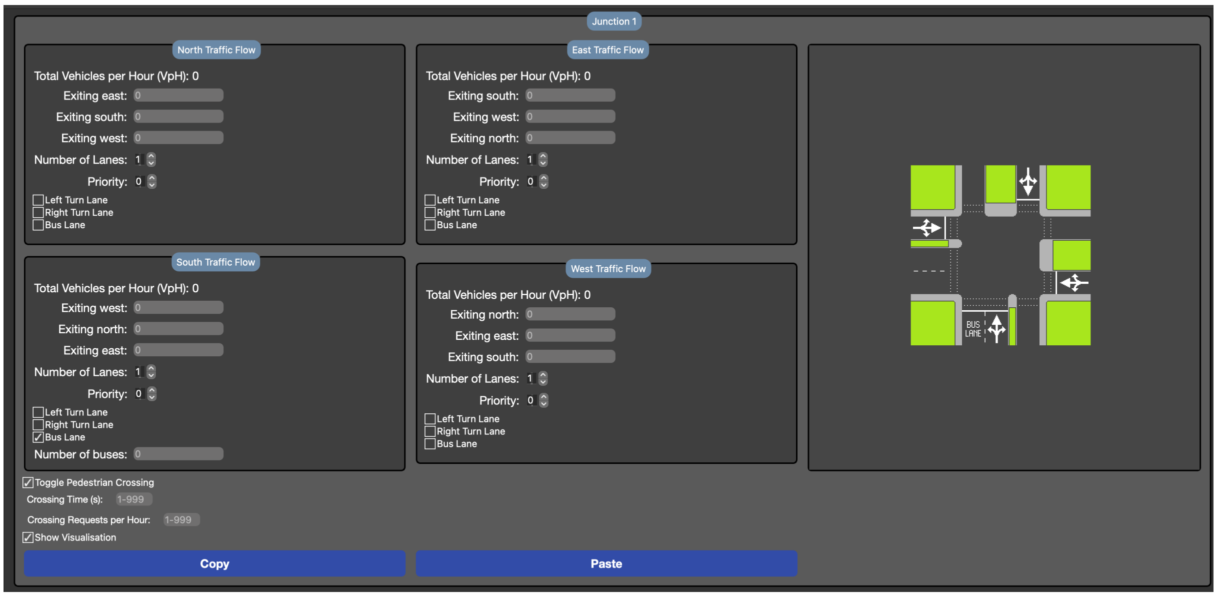
5.2 Home Page

The user is first greeted with the home page where some general information about the system is displayed. The user can then click the "Go to Input Parameters" button to go to the Inputs Page or click the "Exit" button to exit the application. The image below shows the home page of the system:

Figure 1: Home Page of the System

5.3 Input and Validation Page

The Input Page provides a structured interface for configuring traffic flow conditions. The page initially consists of a parent junction box, labelled "Junction 1", which contains child boxes for each direction of traffic flow. Each child box contains input fields for the user to input the desired parameters for the simulation. The image below shows the parent junction box and the child boxes for each direction of traffic flow:


Junction 1

North Traffic Flow
Total Vehicles per Hour (VpH): 0
Exiting east: 0
Exiting south: 0
Exiting west: 0
Number of Lanes: 1
Priority: 0
☐ Left Turn Lane
☐ Right Turn Lane
☐ Bus Lane

East Traffic Flow
Total Vehicles per Hour (VpH): 0
Exiting south: 0
Exiting west: 0
Exiting north: 0
Number of Lanes: 1
Priority: 0
☐ Left Turn Lane
☐ Right Turn Lane
☐ Bus Lane

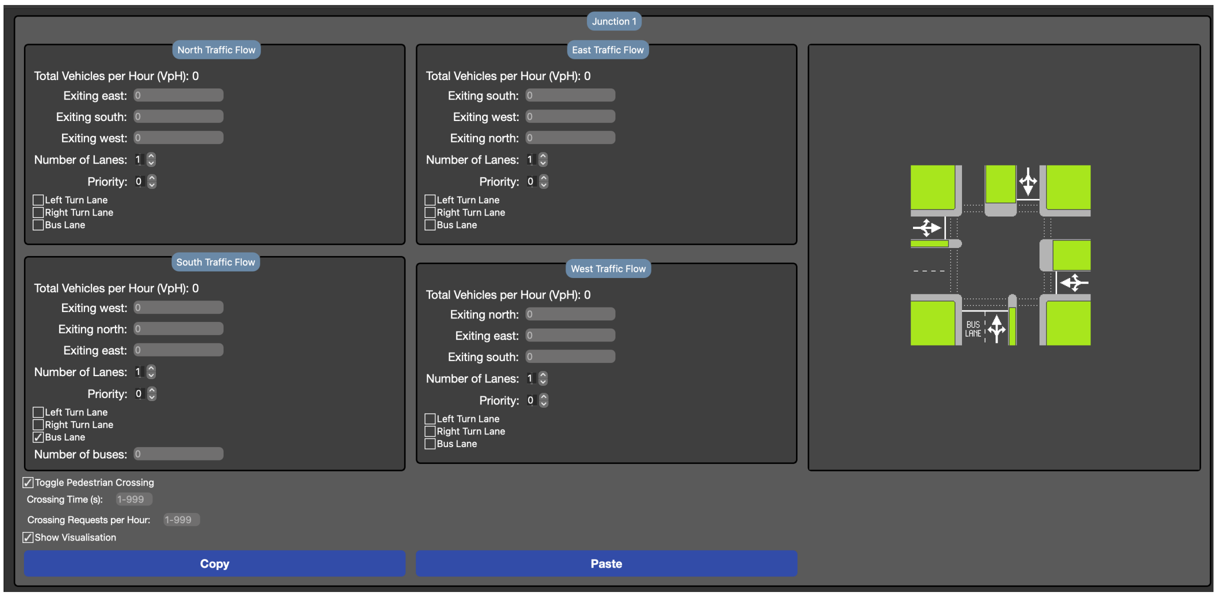
South Traffic Flow
Total Vehicles per Hour (VpH): 0
Exiting west: 0
Exiting north: 0
Exiting east: 0
Number of Lanes: 1
Priority: 0
☐ Left Turn Lane
☐ Right Turn Lane
☑ Bus Lane
Number of buses: 0

West Traffic Flow
Total Vehicles per Hour (VpH): 0
Exiting north: 0
Exiting east: 0
Exiting south: 0
Number of Lanes: 1
Priority: 0
☐ Left Turn Lane
☐ Right Turn Lane
☐ Bus Lane

☑ Toggle Pedestrian Crossing
Crossing Time (s): 1-999
Crossing Requests per Hour: 1-999
☑ Show Visualisation

Figure 2: Illustration of the Input Box

The parent box also contains two checkboxes for the user to select whether they want to add pedestrian crossing options and whether they want to see the updates on the junction on real time. A visual representation of the junction is displayed on the right side of the parent box, which updates in real time as the user inputs the parameters. The image below shows the visual representation:


Junction 1

North Traffic Flow
Total Vehicles per Hour (VpH): 0
Exiting east: 0
Exiting south: 0
Exiting west: 0
Number of Lanes: 1
Priority: 0
☐ Left Turn Lane
☐ Right Turn Lane
☐ Bus Lane

East Traffic Flow
Total Vehicles per Hour (VpH): 0
Exiting south: 0
Exiting west: 0
Exiting north: 0
Number of Lanes: 1
Priority: 0
☐ Left Turn Lane
☐ Right Turn Lane
☐ Bus Lane

South Traffic Flow
Total Vehicles per Hour (VpH): 0
Exiting west: 0
Exiting north: 0
Exiting east: 0
Number of Lanes: 1
Priority: 0
☐ Left Turn Lane
☐ Right Turn Lane
☑ Bus Lane
Number of buses: 0

West Traffic Flow
Total Vehicles per Hour (VpH): 0
Exiting north: 0
Exiting east: 0
Exiting south: 0
Number of Lanes: 1
Priority: 0
☐ Left Turn Lane
☐ Right Turn Lane
☐ Bus Lane

☑ Toggle Pedestrian Crossing
Crossing Time (s): 1-999
Crossing Requests per Hour: 1-999
☑ Show Visualisation

Figure 3: Illustration of the Input Box with additional options

This layout was achieved by using the PyQT library to create the boxes using QGroupBox widgets.

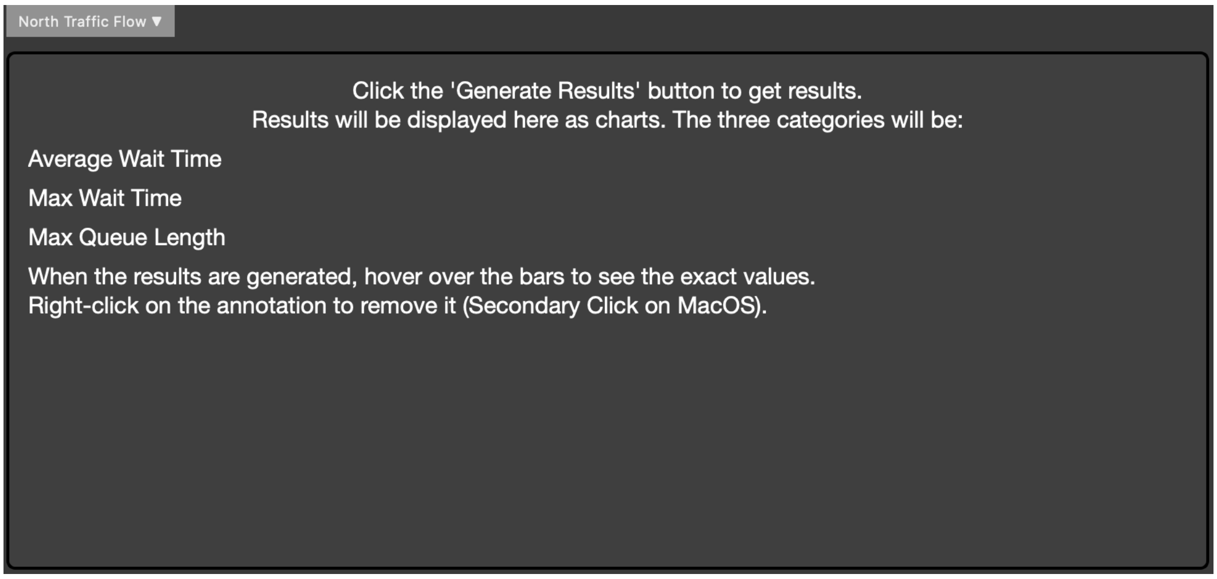
5.3.1 Visualisation

In the original design, we had the idea to have a Sankey diagram placed on top of the visualisation so that the user could view the flow of traffic into each exit where the size of the arrow would be the VpH going towards that exit. We decided against this as it would overcomplicate the visualisation by including this overlay with the addition of add buttons to choose which exit you would like to view.

Instead, we chose to have an adapting visualisation where the diagram of the junction would update whenever the user would add an exit or change a parameter, such as adding an extra lane or adding a pedestrian crossing.

5.4 Results Page

The Results Page initially displays a divided design, a 2x2 grid where each cell is planned to contain result information about the traffic flow for each direction, but initially, they display instructions on how the user can generate the results and how they they will be displayed. The image below shows the initial state of a box in the results page:



Click the 'Generate Results' button to get results.
Results will be displayed here as charts. The three categories will be:

Average Wait Time

Max Wait Time

Max Queue Length

When the results are generated, hover over the bars to see the exact values.
Right-click on the annotation to remove it (Secondary Click on MacOS).

Figure 4: Illustration of the initial state of a box in Results Page

Once the user has generated the results, the boxes will display the metrics of the simulation on bar charts, where for the results categories (Max Queue Length, Average Wait, and Max Wait) each bar represents a different junction configuration. While the cursor is hovering over a bar, the user will be able to see the exact value of the metric. As instructed in the initial state of each direction box, the user can remove the annotation by right clicking over the ot (Secondary Click on MacOS). The image below shows the results page with the results of the simulation:

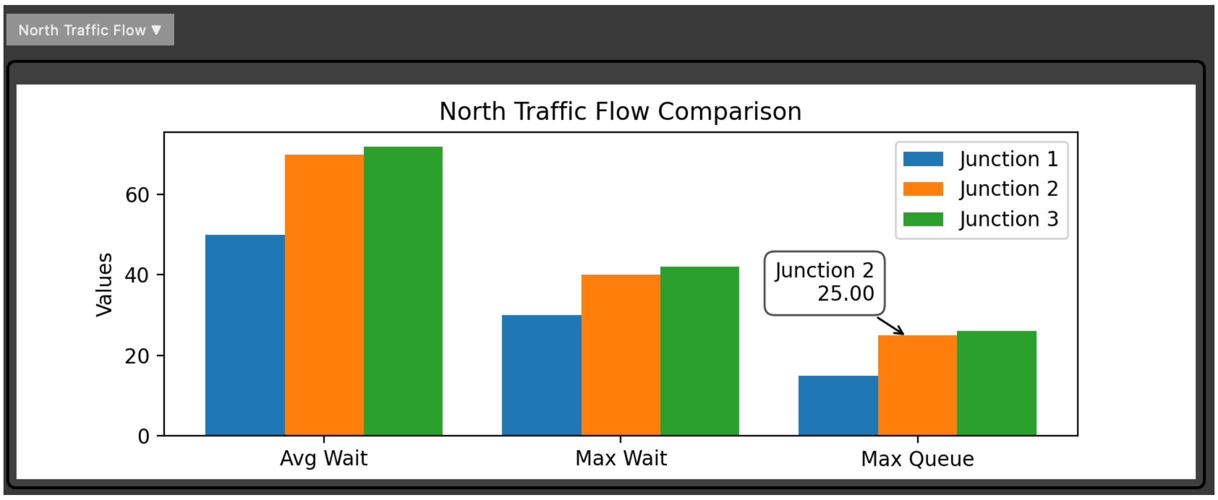
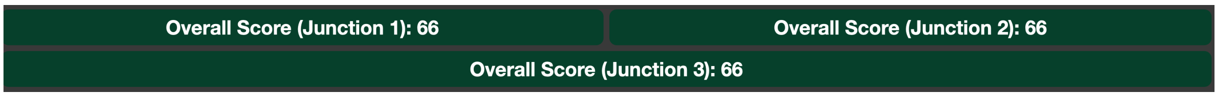


Figure 5: Illustration of the chart with the results of the simulation for 3 different junction configurations

Once the results are generated, the overall scores for each junction configuration are also shown above the box layout of the directions. Depending on the number of junction configuration, the the overall score labels will be adjusted accordingly to the layout of the page. The image below shows the results page with the overall scores displayed:



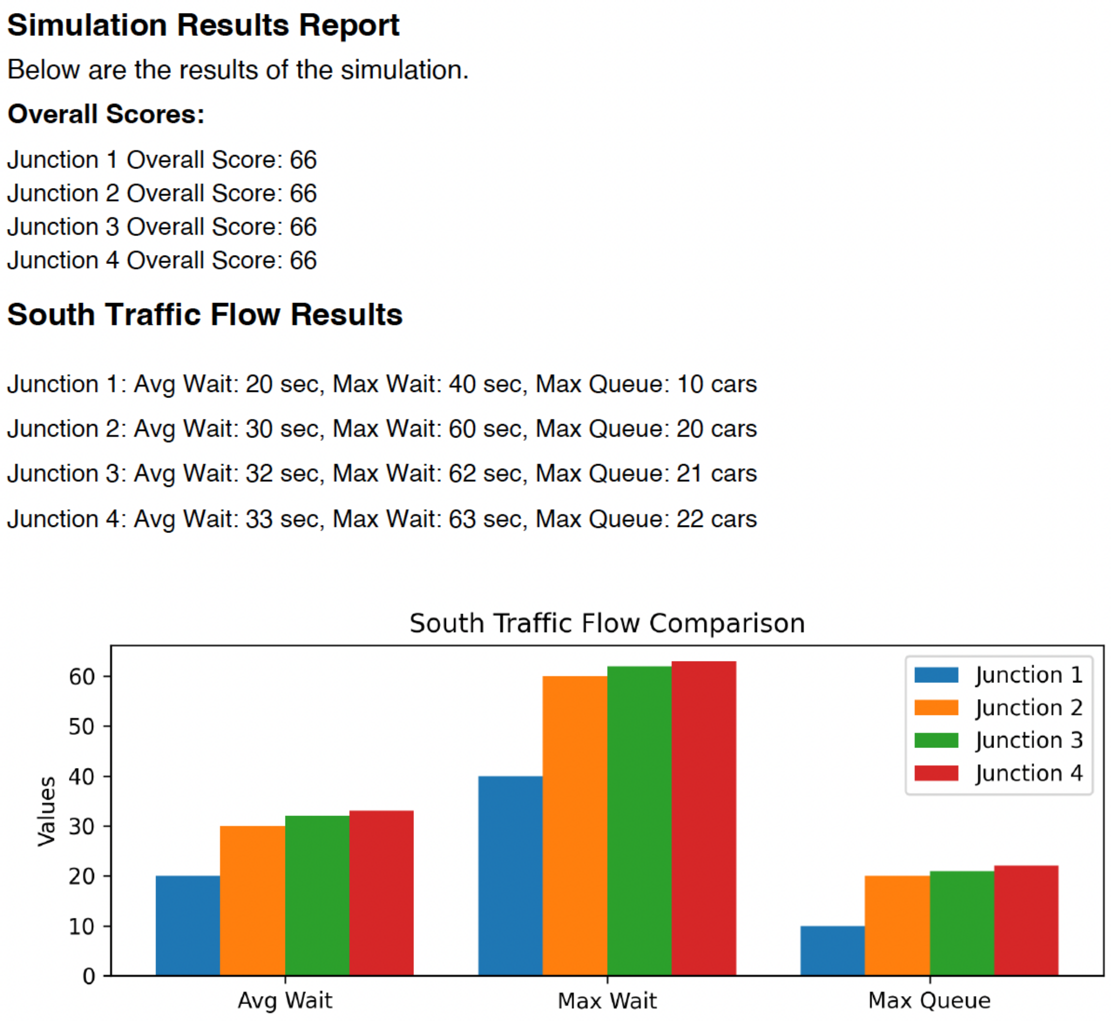
Overall Score (Junction 1): 66

Overall Score (Junction 2): 66

Overall Score (Junction 3): 66

Figure 6: Illustration of how the overall scores are displayed

Furthermore, with the generation of the results, the user will also be able to generate a report PDF of the results. The user can do this by clicking the "Generate Report" button at the bottom of the page. The PDF report will contain the charts and the metrics for each direction in a text format. The image below shows an example of a PDF report generated by the system:



Simulation Results Report

Below are the results of the simulation.

Overall Scores:

Junction 1 Overall Score: 66
Junction 2 Overall Score: 66
Junction 3 Overall Score: 66
Junction 4 Overall Score: 66

South Traffic Flow Results

Junction 1: Avg Wait: 20 sec, Max Wait: 40 sec, Max Queue: 10 cars

Junction 2: Avg Wait: 30 sec, Max Wait: 60 sec, Max Queue: 20 cars

Junction 3: Avg Wait: 32 sec, Max Wait: 62 sec, Max Queue: 21 cars

Junction 4: Avg Wait: 33 sec, Max Wait: 63 sec, Max Queue: 22 cars

Figure 7: Sample of a PDF report generated by the system

5.5 PDF generation

5.6 Error Handling

Within our application, there are many scenarios which require the user to input certain parameters or complete steps in a specific order. The way we handled these problems as a team were by checking and verifying the user's inputs by setting up certain constraints on the input boxes and by displaying error messages with the necessary steps to fix the issue on them.

5.6.1 Input Checking

On the Input and Parameter page we ensured that the user entered the correct inputs by having certain checks in place:

- When the user enters the VpH exiting each exit, we ensured that the input was an integer, greater or equal to 0, and had an upper bound of 999 VpH.
- When the user enters the number of lanes for each exit, the number had to be greater than 0, had an upper bound of 5, and was always an integer.
- For the Priority, we made sure that the user could only have a priority that was greater than or equal to 0, at most 4, and was an integer.
- If the user had chosen to add a pedestrian crossing, we made sure that the time it takes for people to cross and the number of crossing requests per hour were both greater than 0, less than 1000, and always an integer.

5.6.2 Error Messages

There are features in our application that require the user to complete steps in a certain order. If the pre-requirement is not met for a certain feature, an error message will be displayed:

- If the user chooses to remove an alternate configuration without creating one in the first place, an error message will be displayed that will tell the user too add an alternate configuration before they are able to remove one.
- In the scenario that the user decides to generate a report PDF without generating any results, an error message would display: telling the user to press the generate results button in order to generate the results of their configuration before pressing the report button again.
- As the user is setting up their junction configuration, if the user does not set a suitable number of lanes, when starting the simulation an error message will display. The error message will prompt the user to change the number of lanes on the exit that has an inconsistent number of lanes and incoming traffic. The error message should tell the user which junction configuration is causing an issue and which direction of traffic flow needs to be changed.

5.7 Styling

6 Backend

6.1 Data Classes

6.2 Simulation

6.3 Results

7 Testing

7.1 Unit Testing

7.2 User Testing

7.3 Error Handling Testing

- Precondition Testing
  - Ensured that error messages were displayed when prerequisites for actions were not met:
    - When attempting to remove an alternate configuration without having created one, an error message is displayed prompting the user to add one first.
    - When attempting to generate a report PDF without first generating results, an error message is displayed informing the user to generate results before generating a report.

        * when Attempting to start the simulation when the lanes aren't balanced with the traffic, the error message should always be raised.

- Sequence Testing
    - Validated that the correct error messages were triggered when users followed incorrect sequences of actions:
        * Attempting to start the simulation without setting a suitable number of lanes triggers an error message specifying the exit lane inconsistency.
        * Attempting to generate a report without generating results triggers an error message instructing the user to press the generate results button first.
        * Attempting to remove an alternate configuration before generating an alternate configuration should raise the error.
- Consistent Testing
    - Checked that error messages were consistently displayed for the same type of error across different scenarios:
        * If an exit lane configuration is inconsistent with incoming traffic, the error message always specifies which direction of traffic and which exit are causing the issue.
        * Made sure the error message when removing an alternate configuration only happened when there were 0 alternate configurations generated.
        * Made sure that the error message was always displayed if the user tried to generate a report before trying to generate any results

7.4 Input Checking

- Boundary Testing
    - Tested inputs on both the lower and upper bounds for the valid ranges:
        * For VpH, tested the range 0-999 to ensure values within this range are accepted.
        * For the number of lanes, tested the range 1-5 to ensure values within this range are accepted.
        * For priority levels, tested the range 0-4 to ensure values within this range are accepted.
        * For pedestrian crossing time and number of requests per hour, tested the range 1-999 to ensure values within this range are accepted.
- Erroneous Testing
    - Tested with non-integer inputs (e.g., strings like "abc" or decimals like 2.5) to ensure only integers are accepted.
    - Tested with invalid characters (e.g., special characters like "@#$") and blank spaces to ensure they are rejected.
    - Tested values just outside the allowed range (e.g., -1 for VpH, 0 for number of lanes, and 6 for priority) to ensure they are not accepted.

8 Product Evaluation

8.1 Deployment

8.2 User Feedback

8.3 Future Improvements

9 Process Evaluation

9.1 Team Coordination and Communication

As stated in the planning and design document, we met up once a week to discuss what needed to be worked on in the project. We discusses which each person had to do that week and whether and problems came up. While creating the application we met up again each week to work on the application together in the DCS labs where we would walk through what each person worked on so everyone on the team understood what they had written. Furthermore, everyone was active on the discord group we created to discuss any problems that arose.

Overall, our team coordination and communication was very good as everyone was able to keep up-to-date with the application, what was needed to be done, and any problems that came up.

9.2 Methodology Feasibility and Evaluation

9.3 Time Management

9.4 Key Strengths and Limitations

10 Conclusion