Software Design Document

NSW Traffic Penalty Analyser

Joshua Thomas - s5220078  
Roger Harley - s5221666

Table of Contents

[1.0 System Vision 3](#_Toc113367649)

[1.1 Problem Background 3](#_Toc113367650)

[1.2 System Overview 3](#_Toc113367651)

[1.3 Potential Benefits 3](#_Toc113367652)

[2.0 Requirements 4](#_Toc113367653)

[2.1 User Requirements 4](#_Toc113367654)

[2.2 Software Requirements 4](#_Toc113367655)

[2.3 Use Cases & Use Case Diagrams 5](#_Toc113367656)

[3.0 Software Design and System Components 6](#_Toc113367657)

[3.1 Software Design 6](#_Toc113367658)

[3.2 System Components 6](#_Toc113367659)

[3.2.1 Functions 6](#_Toc113367660)

[3.2.2 Data Structures / Data Sources 6](#_Toc113367661)

[3.2.3 Detailed Design 6](#_Toc113367662)

[4.0 User Interface Design 7](#_Toc113367663)

[4.1 Structural Design 7](#_Toc113367664)

[4.2 Visual Design 7](#_Toc113367665)

# System Vision

## Problem Background

Revenue NSW data reveals there were “3.2 million fines worth $907,376,871 issued between December 2020 and November 2021” (Noble, 2022). This means that over 3.2 million units of data were required to be sorted, analysed and stored in order to be useful to anyone in need. However, even with this desperate need for data analysis and organization there is currently no solution in place.

This is simply not feasible for any human to complete without automation whatsoever and leaves valuable data useless for any tasks requiring it.

## System Overview

The designed solution will provide users with a GUI and the ability to sort collected data by a variety of different categories. For example, a user would have the ability to sort data by date and time and only show results collected within a specified date range. Another example would include users being able to sort data via offence type, and only display penalty data collected from specific penalties, as specified previously.

Another feature of the proposed design would show graphical analysis of specified data sets. This analysis will be able to provide users with percentages, ‘hot spots’ for accidents and an overall statistical analysis of the gathered data.

## Potential Benefits

The potential upside for this project could be very large. Providing law enforcement personnel with the ability for intelligent analysis would enable the use of the software to potentially prevent many incidents by identifying trends and patterns occurring within the data. Such data would be able to identify trends such as locations that are in need of observation in order to allow users of the software to prevent future accidents or issues. For example, if the software was able to show that a particular street or intersection had an unusually large number of penalties occurring near/in it, a user could observe the street and fix any issues that may be causing the issues. Such fixes may include activities such as: filling potholes, fixing traffic lights, etc.

# Requirements

## User Requirements

The user is to interact with the developed project through an installed piece of software residing within their company computers. The user must be able to able to interact with the software via a developed GUI. The software would open into a main menu, providing the user with many different options and selections in order to sort/analyse and visualise data. Once the selection is made, there will be a button provided to finalize the provided options and open a separate window in order to show the desired output, whether that be straight data, or a developed and predetermined visualisation option.

## Software Requirements

R1.1 - The program shall open into a ‘main menu’ showcasing all available options presented to the user.

R1.2 – The program shall provide multiple descriptive text boxes with a brief description of what each user selection does/ performs to the data.

R1.3 – The program shall provide users filtering options such as date/time, penalty details, etc.

R1.4 – The program shall provide users with a finalization option that takes the selected options and completes various operations on the collected data in order to output the desired results.

R2.1 – The program shall have the functionality to read and fetch available traffic penalty data downloaded as a .csv file from the provided source, attached below.  
<https://www.kaggle.com/datasets/llihan/australia-nsw-traffic-penalty-data-20112017>

R2.2 – The program shall create a localized formatted database using the python PANADS library.

R2.3 – The program shall interact with said database by utilizing mySQL, with correct formatting used so that each selected option generates a personalized query command

R2.4 – The program shall limit the use of allowable user input, and where provided, the program will insure the text is thoroughly ‘sanitized’ in order to harden the program against attacks such as SQL injection.

R3.1 - Once all selections have been chosen and the finalizing button pressed, the program shall open a new window displaying the user with the requested data

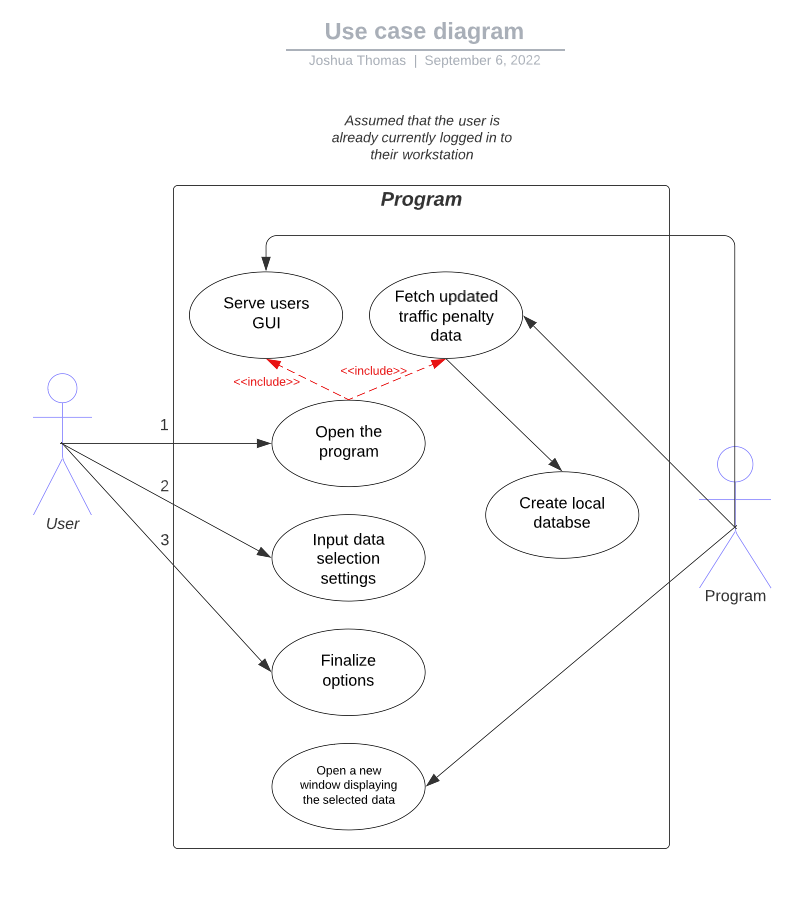
R3.2 – Upon the finalization, the program may (depending on user settings) generate a selected data visualisation model. Such models include, Box chart, line graph, scatterplot, etc. This will be generated using the python library ‘matplotlib’.

R3.3 – The program shall provide users with the ability to save the generated chart to their machines locally as a .png file.

R3.4 – Once the generated data is saved / discarded, the program shall remain operational and allow users to re-input settings and repeat the process over and over again.

## Use Cases & Use Case Diagrams

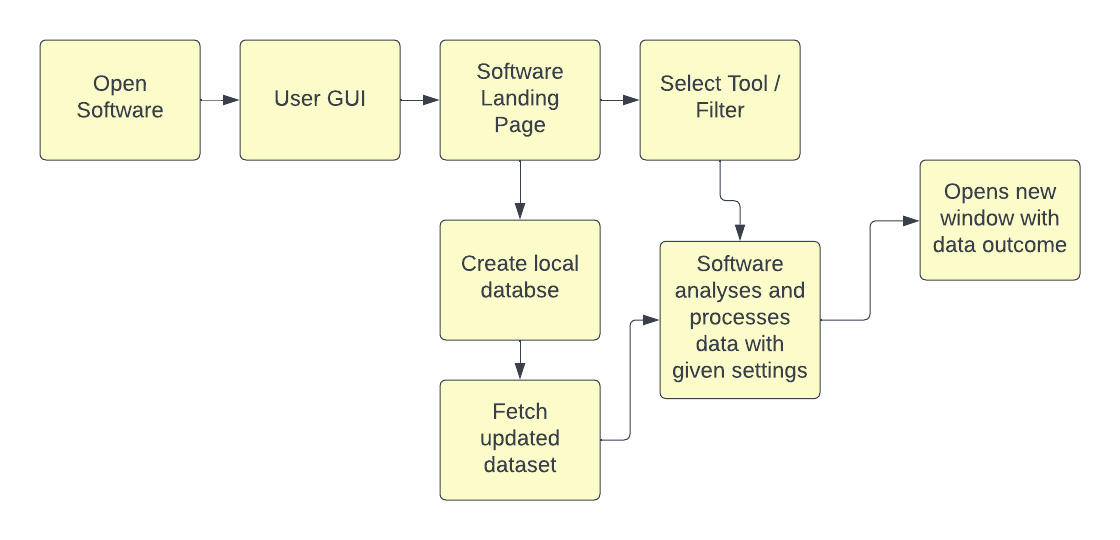
**Use case example:**   
A user wishes to view a line graph of the totally accumulated fines between the dates 01-01-2020 and 01-01-2022. The user must select (from available options) the following: Type: Fines, Between dates checkbox, and input the dates from 01-01-2020 to 01-01-2022, furthermore the user must select the ‘display data’ checkbox and indicate the type of display as a line-graph. This particular search could be used to analyse whether currently implemented traffic safety measures are effective in reducing the number of accumulated penalties.

A user wishes to view a histogram of all traffic penalty data occurring at a particular location code “7296”. The user must select; Location code; 7296, the user must also select the ‘display data’ checkbox and indicate the type of display as a histogram. This particular search could be used to analyse whether a particular location is higher or lower risk of traffic penalties occurring, of which a further analysis would be able to indicate why this issue is present and possible solutions to mitigate the problem.   
**Use case diagram:**

# Software Design and System Components

## Software Design

**Block Diagram**



## System Components

### Functions

|  |  |  |
| --- | --- | --- |
| **Function** | **Description** | **Return Value** |
| loadData | Loads the data (the dataset) from the database. | Returns ‘true’ or ‘false’. |
| saveData | Saves the data into the database. | Returns ‘true’ or ‘false’. |
| loadDisplay | Loads up the display (GUI) | Returns ‘true; or ‘false’. |

### Data Structures / Data Sources

Array: Used to store data, in this instance it will be used with functions such as ‘loadData’. The data from the database will be temporarily be stored in the array.

Tree: A tree will be used to connect data items of different hierarchies. It will be used throughout the programs code and for all functions.

### Detailed Design

Some algorithms will be implemented into the tool to accurately analyse and estimate the given datasets.

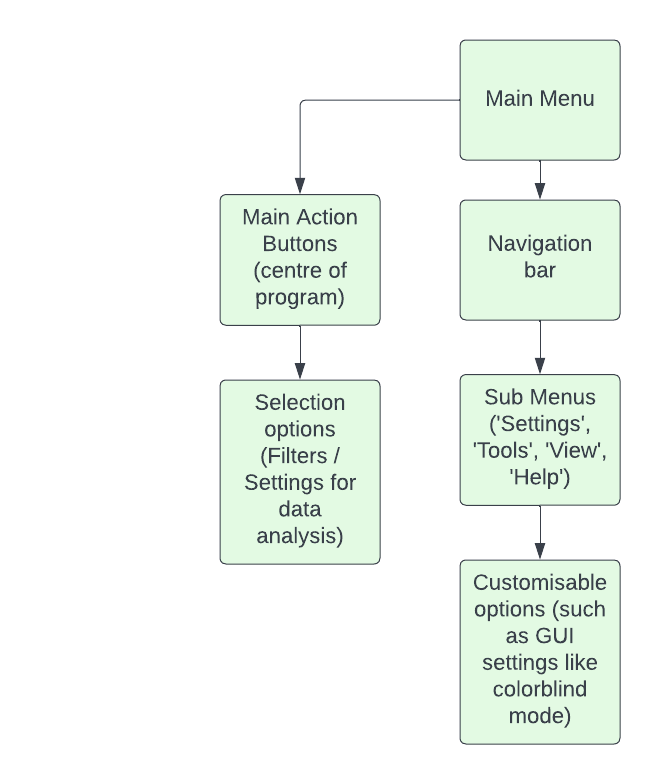
# User Interface Design

The tools that were used to create the designs are Lucidchart, a visual drawing tool and paint. No key findings were found that informed the design. The designs include a structural design that shows the navigational and information aspects of the software. This includes a hierarchy diagram and screen mock-ups.

## Structural Design

The software tool will be structured in a simplistic manner. Customisable options will be grouped up in the navigation bar at the top of the program. These include dropdown menus for ‘Settings’, ‘Tools’, ‘View’ and ‘Help’. The main actions will take place in the centre of the program screen. These actions include the starting of analysing the dataset, choosing data analysis settings, finalizing and beginning the process and a loading action.

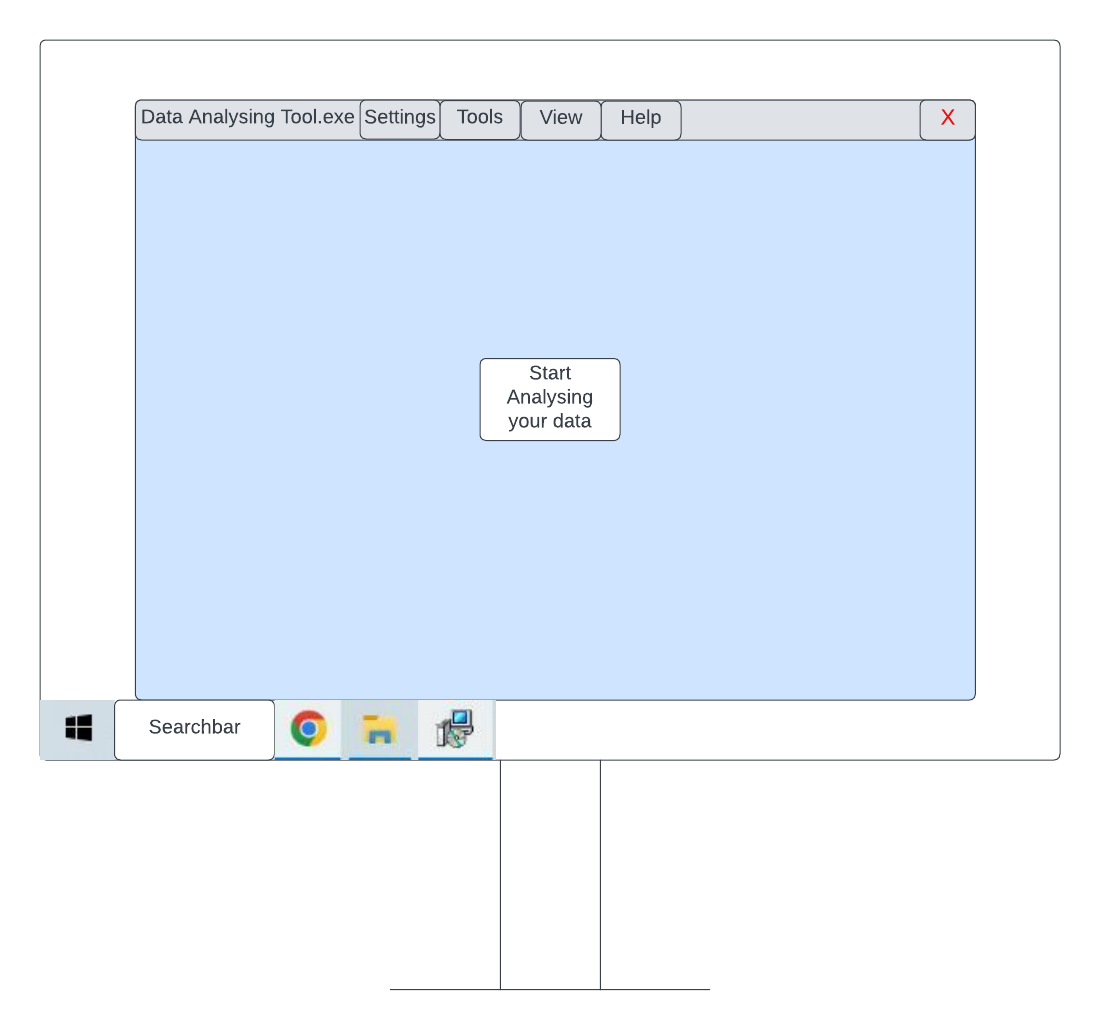
Below is a brief visualisation of the software's hierarchy.



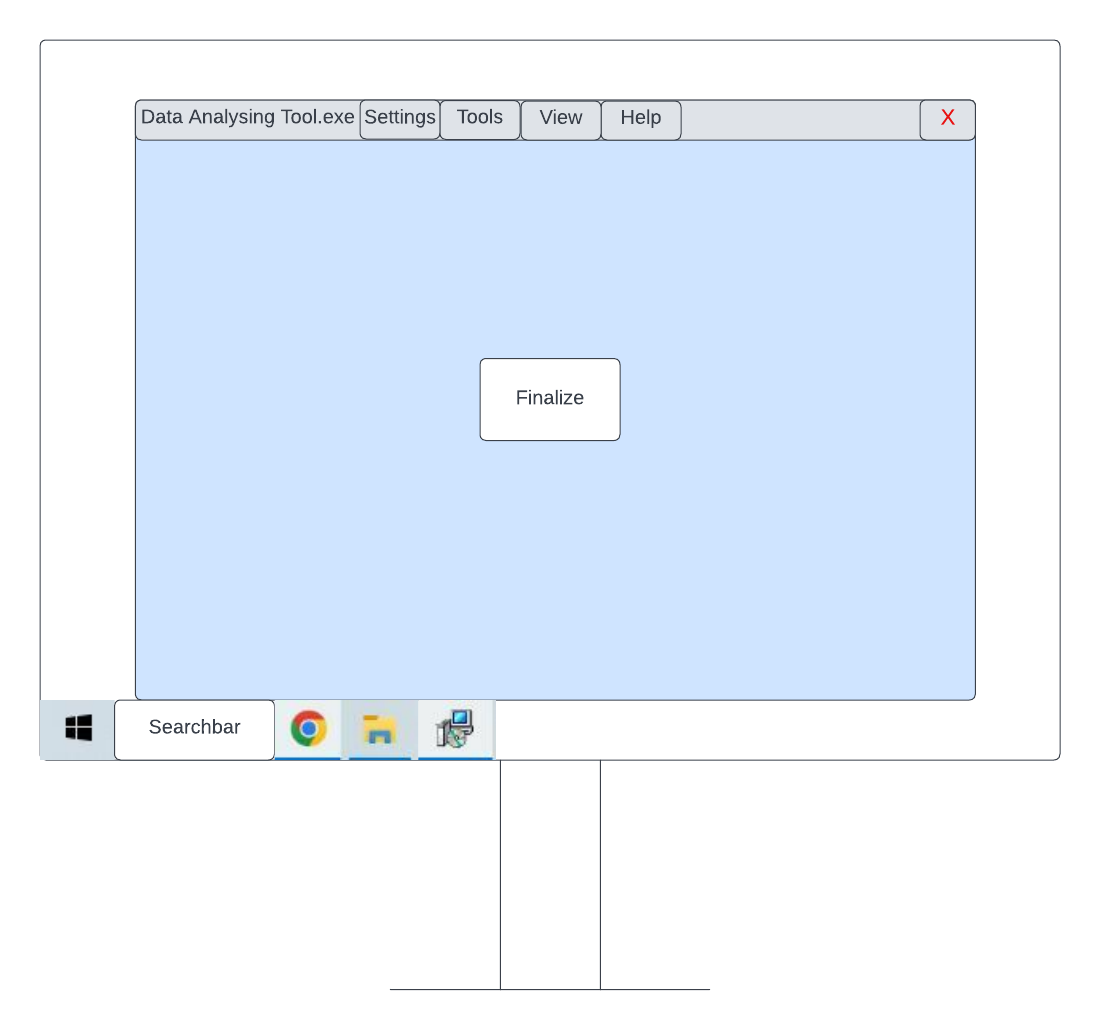
## Visual Design

The layout includes the programs name with an icon in the top left of the navigation bar. Right of the program name, is a settings dropdown menu, tools dropdown menu, view dropdown menu and help dropdown menu. To access these menus, it is required to either hover your mouse or click the button to drop the menu down. The background of blue and navigation bar of grey are the only colours used. This is to keep the design as simplistic as possible. The font selected for all texts in the software is ‘Liberation Sans’. This font was selected as it is clear and easy to read, creating an ideal user experience.

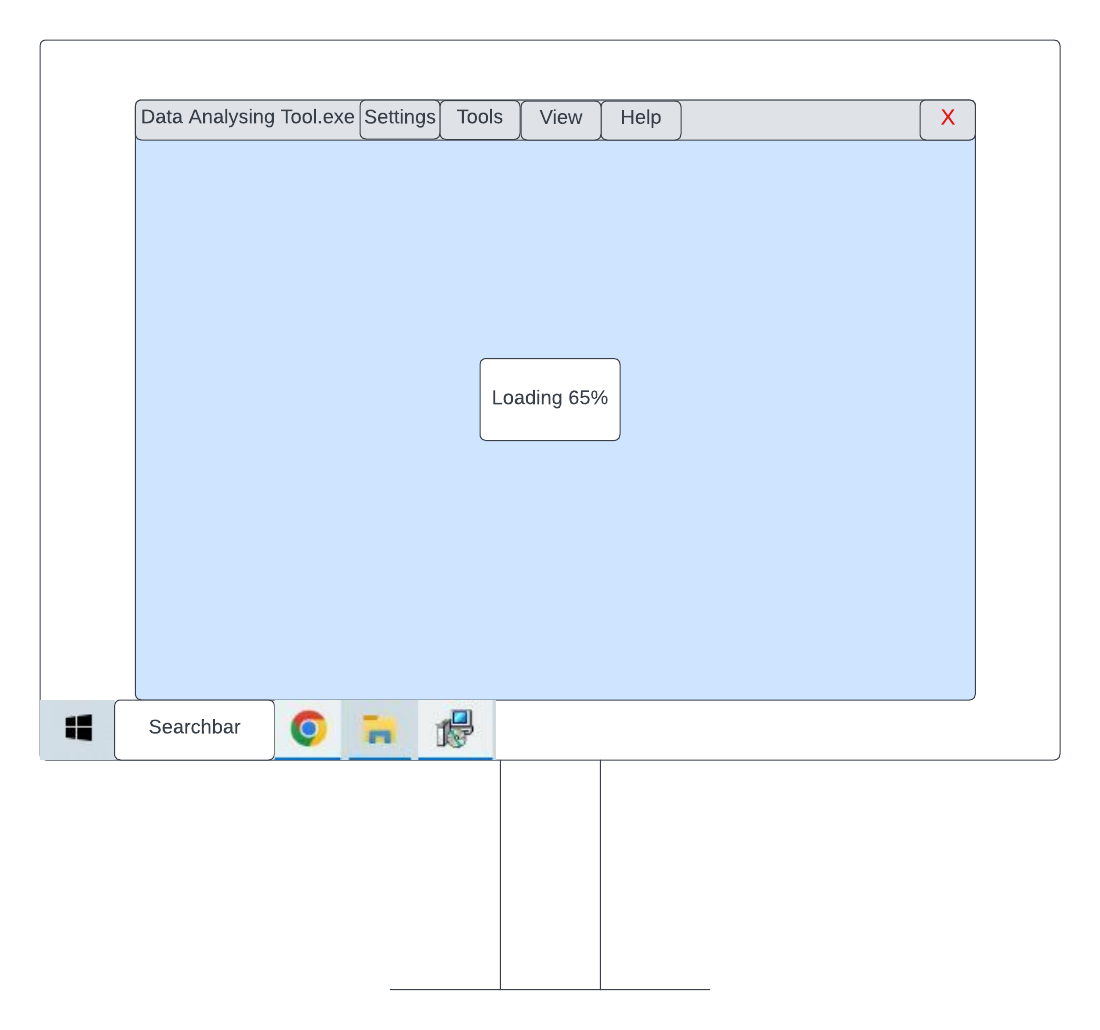
Below are some design mock-ups for some of the pages within the software.



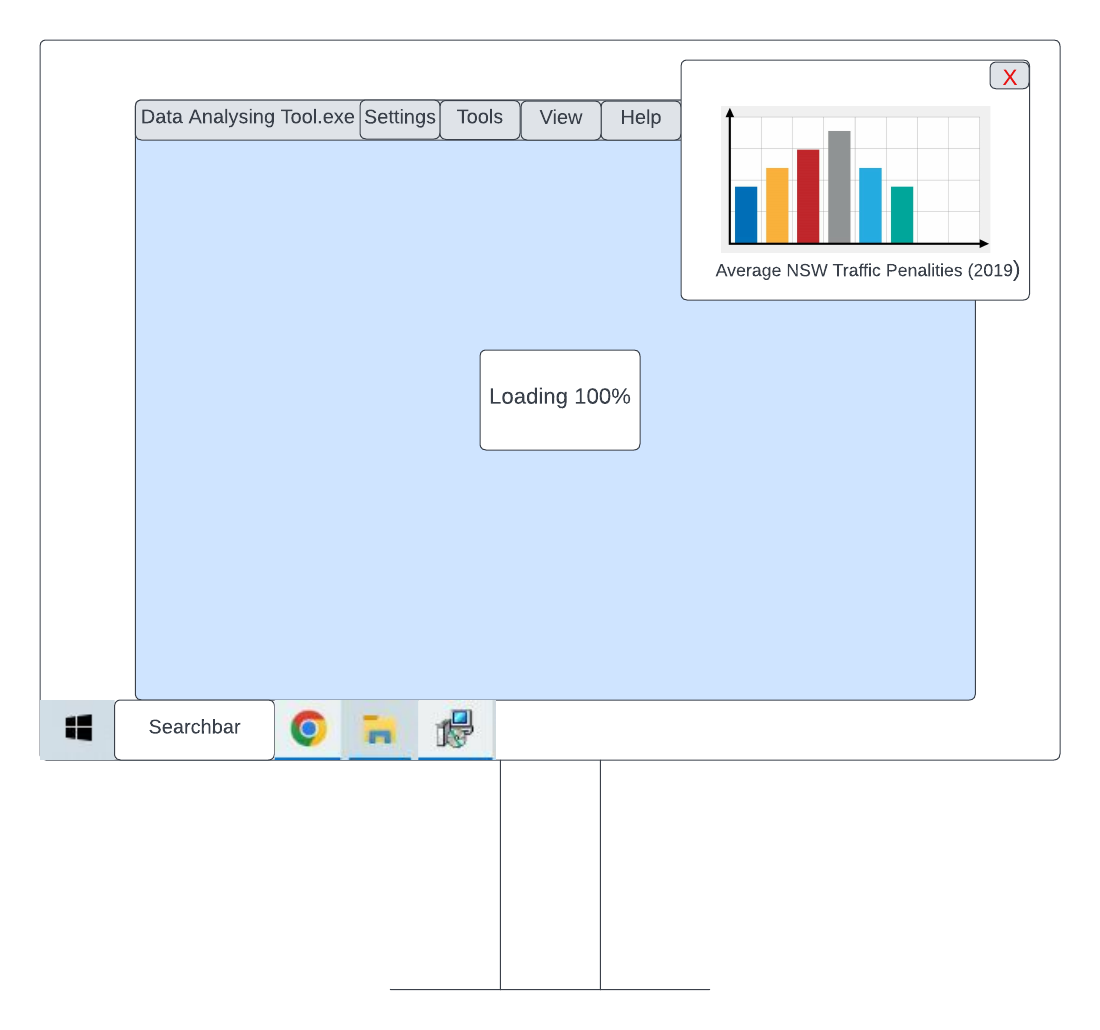
**Landing Page**



**Finalization Button**



**Loading Page**



**Statistical graph output**