Software Design Document Part B

< NSW Traffic Penalty Data Analysis and Visualisation Tool >

Zak Cobham-Davis, William Crane, Christopher Linnett

Table of Contents

[1.0 System Vision 3](#_Toc113206237)

[1.1 Problem Background 3](#_Toc113206238)

[1.2 System Overview 3](#_Toc113206239)

[1.3 Potential Benefits 3](#_Toc113206240)

[2.0 Requirements 4](#_Toc113206241)

[2.1 User Requirements 4](#_Toc113206242)

[2.2 Software Requirements 4](#_Toc113206243)

[2.3 Use Cases & Use Case Diagrams 5](#_Toc113206244)

[3.0 Software Design and System Components 8](#_Toc113206245)

[3.1 Software Design 8](#_Toc113206246)

[3.2 System Components 9](#_Toc113206247)

[3.2.1 Functions 9](#_Toc113206248)

[3.2.2 Data Structures / Data Sources 10](#_Toc113206249)

[3.2.3 Detailed Design 12](#_Toc113206250)

[4.0 User Interface Design 15](#_Toc113206251)

[4.1 Structural Design 15](#_Toc113206252)

[4.2 Visual Design 16](#_Toc113206253)

# System Vision

## Problem Background

The NSW traffic penalty dataset contained information on all traffic penalties in the state of New South Wales from 2011 to 2017. This dataset included the month and year of the offence, details on the offence, the related legislation, the value of the penalty, and more. Data is contained within a CSV file organised by the date of the offence but is challenging to analyse. It is vital for researchers and the government that this dataset be easily accessible, searchable, and user-friendly to analyse the data best. It is suggested that software should be developed to satisfy these requirements and aid in analysing this dense dataset.

## System Overview

The developed software system allows a user to search for and organise by the individual attributes in the dataset, i.e., date of penalty, offence code/description, and the face value of penalty. Users can select a particular period that a correction had been issued and be able to view relationships and trends for penalties in the period chosen. The system can be deployed on Windows and Unix systems. The data also has a graphical representation of trends and distribution.

## Potential Benefits

Implementing this system of viewing the NSW traffic penalty data aids in policing traffic violations. It can be used in research on improving the safety of roads in New South Wales. Through data analysis, researchers may find trends that can aid in finding the root cause of traffic violations. Through the gained knowledge analysing this data, the government can change traffic legislation, transport infrastructure, and police presence more accurately where needed. This system's potential benefit would be reducing dangerous road activities and increasing knowledge that can assist in this.

# Requirements

(\* The Requirements have Remained the Same between Part A and Part B)

## User Requirements

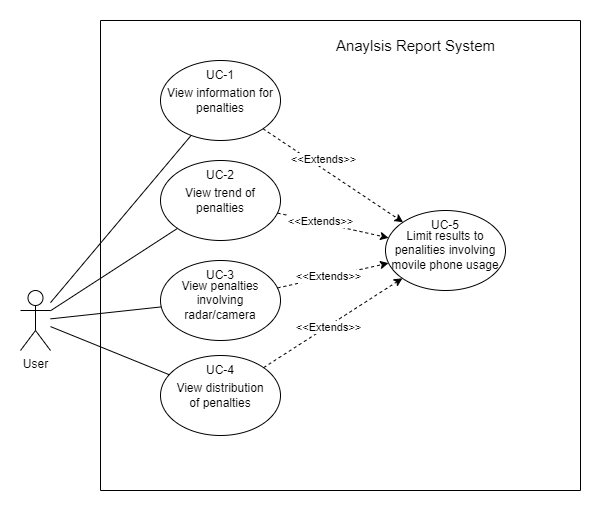
* As a user, I can limit results for all features to only offences involving mobile phone usage.
* As a user, I can view all information for penalties in a selected date range.
* As a user, I can view the distribution of all penalties by offence code in a selected date range.
* As a user, I can view the distribution in either pie- or bar-chart format.
* As a user, I can view all penalties where the offence was captured by radar or camera in a selected date range.
* As a user, I can view the trend in penalties of a specified offence code over time displayed in a bar-graph format.

## Software Requirements

|  |  |
| --- | --- |
| Identifier | Requirement |
| REQ1 | The main UI of the system shall display a dropdown to select the required feature, a checkbox to limit the results to offences involving mobile phone usage, date inputs for start and end date, a textbox for text entry where applicable, and a submit button. |
| REQ2 | The system shall only accept numerical input from the user where applicable. |
| REQ3 | The system's main screen shall display the program's name and a list of group members. |
| REQ4 | The submit button on the system's main screen shall be disabled until the user inputs for the feature are valid. |
| REQ5 | Upon clicking/pressing the submit button on the system's main screen, a results screen shall display. |
| REQ6 | The results screens shall be independent from the system’s main screen |
| REQ7 | Upon clicking the exit button on the results screen, the results window should close leaving the main screen only |
| REQ8 | Results screens which display a list of information on penalties, shall display an interactive scrollable panel where the data is displayed. |
| REQ9 | Results screens that display charts shall include buttons for interacting with the appearance of the chart. |
| REQ10 | The User should be able to open a new results window, even if one is already open. |
| REQ11 | Results screens that display a trend over time for penalties shall display a bar chart. |
| REQ12 | The start and end date inputs will be filled with the dates of the earliest and latest offences recorded respectively by default. |

## Use Cases & Use Case Diagrams

(\* The Use Cases and Use Case Diagrams have Remained the Same between Part A and Part B)



|  |  |
| --- | --- |
| Element | Description |
| ID | UC-1 |
| Name | View information for penalties |
| Brief Description | The report for displaying information for offences in a given time-period is viewed |
| Flow of Events | The user chooses this report to view, inputs a valid start and end date, then submits the inputs to view the report |
| Special Requirements | The report can only be viewed when valid inputs are given |
| Preconditions | * The program is open * The program is on the main UI * The report is selected to be displayed * Valid inputs are given |
| Postconditions | * The data fetched and displayed for the report is cleared from memory |
| Extension Points | Before submitting the report, the user can select to limit the results to offences involving mobile phone usage. |

|  |  |
| --- | --- |
| Element | Description |
| ID | UC-2 |
| Name | View trend of penalties |
| Brief Description | The report for displaying trend over time for offences in a given time-period for a given offence code is viewed |
| Flow of Events | The user chooses this report to view, inputs a valid start date, end date, and offence code, then submits the inputs to view the report |
| Special Requirements | The report can only be viewed when valid inputs are given. Inputting no offence code is a valid input and will display the trend for every offence in the time-period |
| Preconditions | * The program is open * The program is on the main UI * The report is selected to be displayed * Valid inputs are given |
| Postconditions | * The data fetched and displayed for the report is cleared from memory |
| Extension Points | Before submitting the report, the user can select to limit the results to offences involving mobile phone usage. |

|  |  |
| --- | --- |
| Element | Description |
| ID | UC-3 |
| Name | View penalties involving radar/camera |
| Brief Description | The report for displaying information for offences involving radar/camera detection in a given time-period is viewed |
| Flow of Events | The user chooses this report to view, inputs a valid start and end date, then submits the inputs to view the report |
| Special Requirements | The report can only be viewed when valid inputs are given |
| Preconditions | * The program is open * The program is on the main UI * The report is selected to be displayed * Valid inputs are given |
| Postconditions | * The data fetched and displayed for the report is cleared from memory |
| Extension Points | Before submitting the report, the user can select to limit the results to offences involving mobile phone usage. |

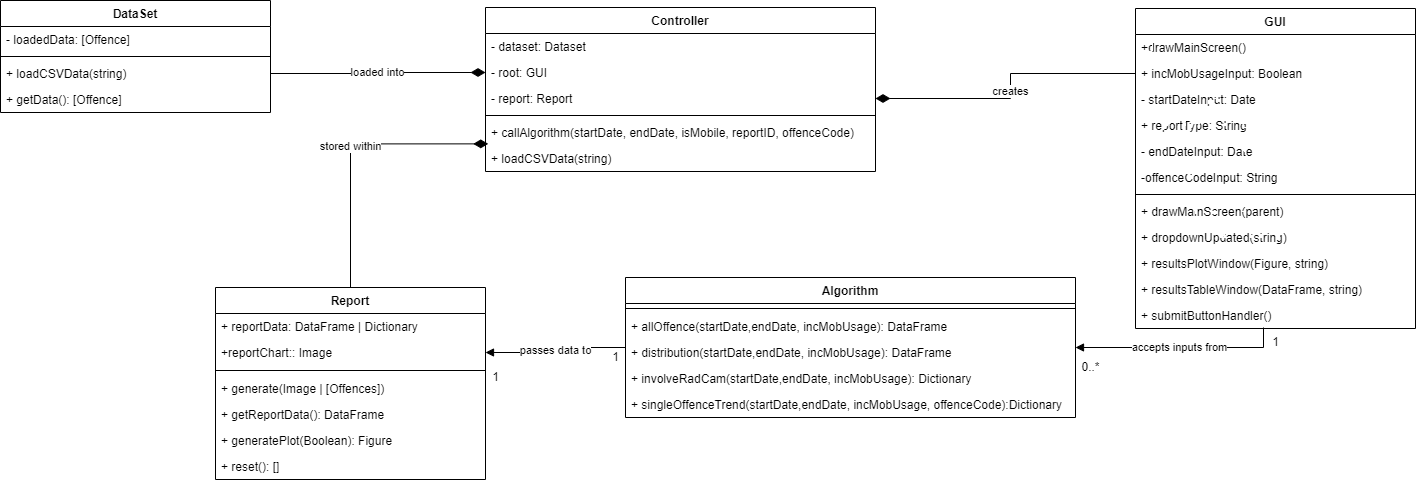
|  |  |
| --- | --- |
| Element | Description |
| ID | UC-4 |
| Name | View distribution of penalties |
| Brief Description | The report for displaying the distribution of offences by offence code in a given time-period is viewed |
| Flow of Events | The user chooses this report to view, inputs a valid start and end date, then submits the inputs to view the report |
| Special Requirements | The report can only be viewed when valid inputs are given |
| Preconditions | * The program is open * The program is on the main UI * The report is selected to be displayed * Valid inputs are given |
| Postconditions | * The data fetched and displayed for the report is cleared from memory |
| Extension Points | Before submitting the report, the user can select to limit the results to offences involving mobile phone usage. |

|  |  |
| --- | --- |
| Element | Description |
| ID | UC-5 |
| Name | Limit results to penalties involving mobile phone usage |
| Brief Description | The user can select to limit any generated reports to include only offences involving mobile phone usage |
| Flow of Events | Before submitting a report, the user can select to limit the results to include only offences involving mobile phone usage |
| Special Requirements | - |
| Preconditions | * The program is open * The program is on the main UI |
| Postconditions | * Any reports generated will be limited to offences involving mobile phone usage |
| Extension Points | After choosing to limit the results, any report generated will only fetch and display offences involving mobile phone usage |

# Software Design and System Components

## Software Design

This class diagram highlights the classes that make up this software. It shows the data stored inside each class and the methods to be used for interacting with the data for the goal result. This class diagram includes changes from the initial diagram that were made due to new information being attained from starting the project. This included learning about how to use the libraries and what methods will be necessary to interact with those. Also in Part A, while all the data types were included, we neglected to add a controller which will manipulate them. As we have written our code using an Agile approach, it has been flexible and open to changes where necessary.



## System Components

### Functions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Function | Description | Input Parameters | Side Effects | Returns: |
| Offence.get | A class method that will get a property's value from the instance. | propertyName: String. The string is the metadata name of the requested property type | none | The value that corresponds to the property |
| DataSet.loadCSVData | A method that loads the dataset from a file, then converts it to a usable format | File: String. The path to the file holding the desired data | loadedData property is assigned a list of offence objects | void |
| DataSet.getData | A method for accessing the processed dataset information | -none | -none | The dataset processed as a list of Offence objects |
| Algorithm.allOffence | Method filters data for user selected time-period and produces report with all offences | startDate: Date, endDate: Date, incMobUsage: Boolean, these are selected in the user GUI, |  | List of offence objects |
| Algorithm.distribution | Method produces chart showing distribution of cases over a user selected time-period | startDate: Date, endDate: Date, incMobUsage: Boolean, these are selected in the user GUI |  | Image of the generated chart |
| Algorithm.involveRadCam | Method filters data for user selected time-period and produces report for offenses captured by radar or camera | startDate: Date, endDate: Date, incMobUsage: Boolean, these are selected in the user GUI |  | List of offence objects |
| Algorithm.singleOffenceTrend | Method generates chart for user selected time-period for user selected offence code. It produces a temporal distribution of occurrences. | startDate: Date, endDate: Date, incMobUsage: Boolean, OffenceCode:String, interval: string, these are selected in the user GUI |  | Image of generated chart |
| GUI.dropDownUpdated | A function to assess whether to limit accessibility of GUI options based on the user's current report selection | dropDownSelection: String. A string describing the function that the user wishes to execute on the data | incMobUsageInput, startDateVisible, endDateVisible, offenceCodeVisible Boolean values are conditionally set | void |
| GUI.submitButtonHandler | A function to submit the user's selections to generate a report | -none |  | Function from Algorithm class, conditionally based on GUI user input properties |
| GUI.resultsPlotWindow | This function accepts a plot from the results and converts it to a tkinter widget, displaying it on a new sindow | Plot: Figure, a datatype from matplotlib  Title: string, will become the title of the results window | A new window object is created within the GUI class | void |
| GUI.resultsTableWindow | This function accepts a dataframe from the results and places it within a pandastable before attaching this to a newly created window | Results: DataFrame from pandas  Title: string, will become the title of the results window | A new window object is created within the GUI class |  |
| Report.generate | This method accepts the algorithm output, stores the results in a property, then displays to the screen in appropriate way. | reportPage: Image or List of Offences. Can receive either of these. | Passes input to reportData or reportChart conditionally based on type | void |
| Report.reset | Hides, then clears the current report and redirects back to the main screen for new selection | -none | reportData and reportChart are set to none | void |

### Data Structures / Data Sources

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Structure | Type of Structure | Description of where and how it's used | List of data members, and their corresponding purposes | List of functions that use it |
| penalty\_data\_set\_2.csv | Comma separated value document | This source holds the data in CSV format | Each member is one record of the dataset. The purpose is to give hold information in a form that is easy to be processed | DataSet.loadCSV |
| DataFrame | DataType from Pandas | This datatype holds the csv data and results in a format easier to work with. | - | DataSet.loadData: will create a new instance of this type passing in the csv data to the constructor  Report.generate: this function will create new instances of this data type to serve as the output for the user depending on the selection criteria of the user. |
| GUI | Class/Object | The GUI object contains values that are used to interface the user's interactions with the code. Different user actions, will change the values of the members within this data type | dropDownSelection: will hold a string value that will identify which of the system's algorithms to act on the data source. Will be selectable by a dropdown menu by the user.  incMobUsageInput: will hold a Boolean value that will indicate whether the results will be limited to offences including an indicator of mobile phone in the offence. This will be used as an input when running the algorithm function  startDateInput: a datepicker that will indicate the start date for the algorithm function.  endDateInput: same as above but for the end date.  offenceCodeInput, a text input that will allow an offence code to be entered. Will be passed into the singleOffenceTrend method if executed.  offenceCodeVisible: indicates whether the option will be open to the user as this is only used for one algorithm. If any other is currently selected this member will remain false. |  |
| DataSet | Class/Object | This is used to hold and access the input data for analysis | loadedData: a list containing only Offence Objects. This list will be the data source for all algorithms. | All Algorithm methods, an instance of this data structure will be used as input for the algorithm method to process and give the desired result. |
| Report | Class/Object | This is used to store the output of the program. This can be either an image or a list of offence objects. | reportData: a list of offence objects that correspond with the input parameters from the user  reportChart: an image depicting a chart that reflects the input parameters from the user | Other methods do not interact with this data structure. Only its own methods. Generate will accept the output from the algorithm and will conditionally add data to one of the members. Reset will clear that data and call a method within GUI to return to the main screen |
| Controller | Class/Object | This is used as the main controller for the application. Initialising this object will start the program | Its members are each of the other classes listed above. This object is used for storing an instance of the others and manipulating data between multiple other classes. | Program launch will call the init of this object. |

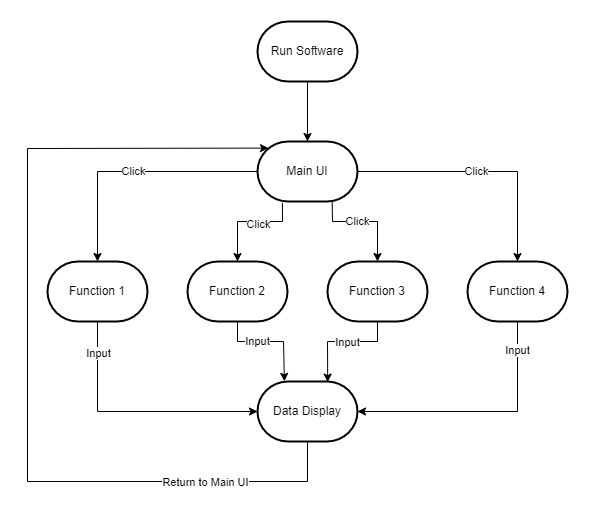
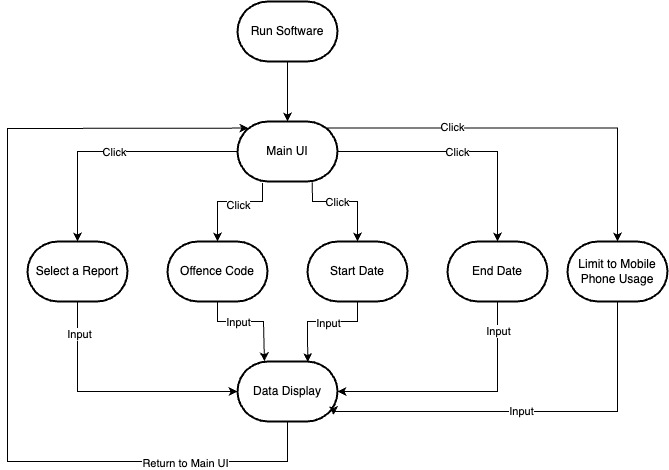
### Detailed Design

|  |  |
| --- | --- |
| Function | Pseudocode |
| Dataset.loadCSVData: | Load dataset into system memory.  Assign an empty list to the dataset's loaded data property.  For each record:  instance a new offence object containing a property for every column of data.  Push that object to the new list. |
| Algorithm.allOffence: | Create New List  For each Offence object in the loadedData property:  If the date property is not between the user's input start date and the user's input end date: Skip to next object.  If the mobileOnly checkbox is not checked: add offence to the new list and skip to next object  If the Mobile phone indicator property is false: skip to next object  Add this offence object to the new List  return the new list |
| Algorithm.distribution: | Create new dictionary object called reportData  For each offence object in the loadedData property:  If the date property is not between the user's input start date and the user's input end date, skip to the next object.  If the mobileOnly output box is checked AND offence property mobilePhone is false: skip to the next object.  If reportData does not have an entry for the offence code of the current offence: set a key as the offence code, and the value to 0  For the entry in reportData that is this offence's code, add 1, move to the next object  Create a bar chart using reportData's keys for the X-axis, and the corresponding values as the Y-axis.  Return this chart as an image |
| Algorithm.involveRadCam | Create New List  For each Offence object in the loadedData property:  If the date property is not between the user's input start date and the user's input end date: Skip to next object.  If the offence description does not include the text "Radar" or the text "Camera": Skip to the next object.  If the phone only checkbox is not checked: add offence to the new list and skip to next object.  If the offence's mobile phone indicator is false: skip to next object.  Add this offence object to the new list.  return the new list |
| Algorithm.singleOffenceTrend | Create new dictionary object called reportData  For each offence object in the loadedData property:  If the date property is not between the user's input start date and the user's input end date: skip to the next object.  If the offence code does not match the user input code: skip to the next object.  If the mobileOnly output box is checked AND offence property mobilePhone is false: skip to the next object.  If reportData does not have an entry for the offence month of the current offence: set a key equal to the offence month property, and the value to 0  For the entry in reportData that is has this offence month, add 1, move to the next object  Create a bar chart using reportData's keys for the X-axis, and the corresponding values as the Y-axis.  Return this chart as an image |

# User Interface Design

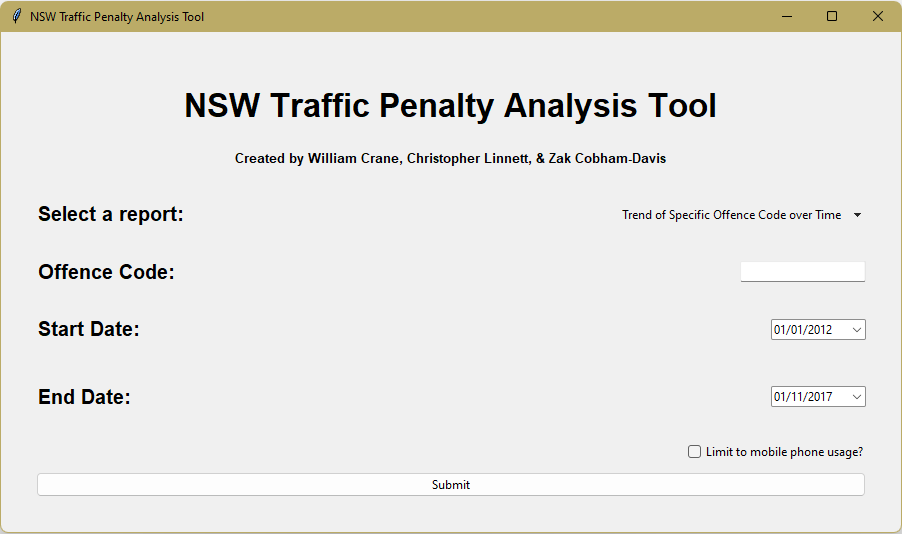
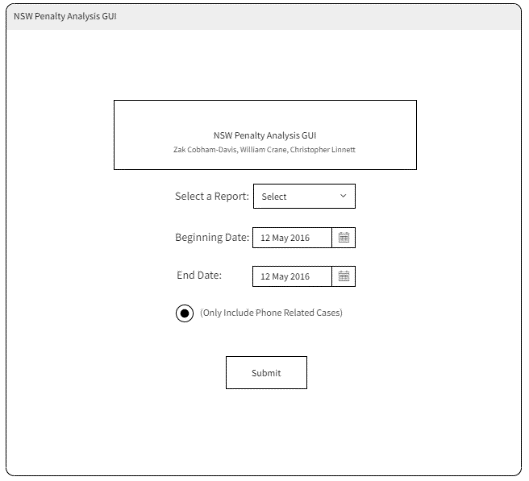
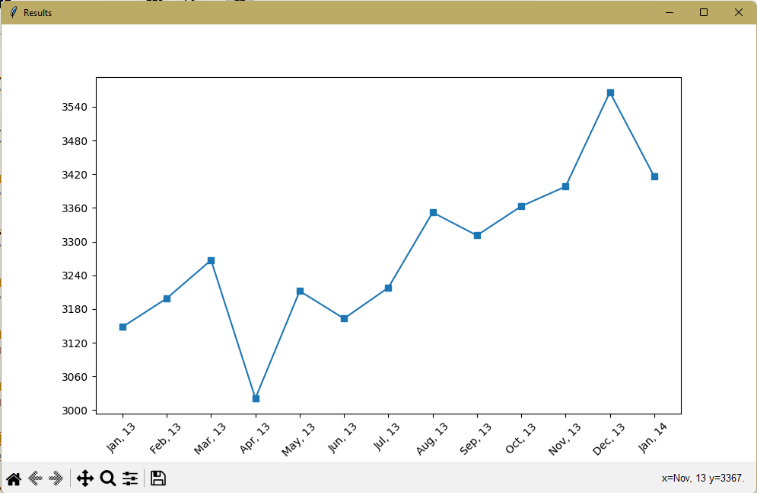
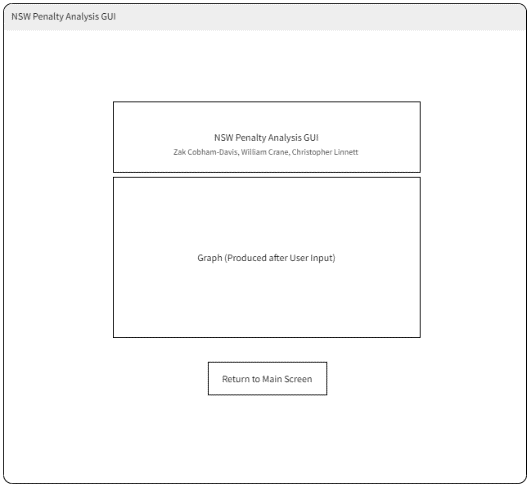
For the design of the NSW Traffic Penalty Program, a specific and detailed collaborative effort was used to produce the first design. This design allows for seamless user interaction amongst all four main functions. The utilised design tool was Wireframe. Cc, specifically for the diagrams. Below will depict and explain the program's structural design (4.1) and how it compares to the finalised GUI, specifically the navigation and layout of the functions and buttons. The visual design and prototype of the GUI will also be depicted (4.2); this will be compared with screenshots of the finalised GUI design and analyse any changes or development. Analysis and justification will also be presented to support the design choices.

## Structural Design

The Structural Design of the NSW Traffic Penalty Program is displayed in a Navigation Diagram Figure, 4.1 displays the initial Structural Design (Part A) 4.2 displays the finalised functional design implemented in Part B.  
(Figure, 4.1)(Figure, 4.2)

The diagram contrasts above are stark in the function section, they have been named, and there is a fifth section. Each element represents a different option that has been actioned by the user (Some are Optional others are Mandatory), and the arrow indicates the action that produces the desired user result. Each action has its own button on the GUI, which displays the most relevant information and usable action to produce the desired result. This contrasts with the Part A design as each button was a function. To support functionality, this has been changed to making, 'select a report' a dropdown function. This updated structure has allowed for the most user-friendly approach and the most simplistic way to produce a Graph or Draw Data from the initial input. It doesn't overcomplicate the user experience as this GUI is required to be 'simplistic' by design. From the initial display, the user can access all reports from a dropdown, select the date range and optionally limit to mobile phone usage or offence code (Based on Report), to produce the required result. Once a user action has been performed on these functions, the user will be shown another view displaying specific data or a graph; on these displays, the user will also have the option to exit to revert to the main page, or simply move this window aside and repeat the process. This structure was chosen and implemented due to its user-friendly nature. The user is looking for a simple and efficient experience as described in the overall goal of the assessment, 'The overall goal for this assignment is to develop a simple data analysis and visualisation tool for a dataset.'.

## Visual Design

The prototype and the finalised GUI are compared below. Figures 4.2.1 and 4.2.3 display the initial conceptual designs for the GUI and graphs implemented in Part B. Figures 4.2.2 and 4.2.4 display the final GUI design. Both are functionally and visually similar contrasting in one button. (Figure, 4.2.1) (Figure, 4.2.2)  
(Figure, 4.2.3) (Figure, 4.2.4)

The initial design choices were made for their user-friendly nature and based upon this assessment's overall goal, which is creating a 'simple data analysis and visualisation tool'. For the finalised GUI Screenshot (4.2.2), any design difference from the Prototype Wireframe (4.2.1) has been functional. The main change is the addition of the offence code button that can be optionally selected for specific reports ('Trend of Specific Offence Code over Time'). No other complicated controls have been implemented that could cause a diminished and over-complicated user experience. The graphical wireframe display (4.2.3) was also developed with simple usability in mind, this display has been altered based on functional requirements (Figure 4.2.4) due to the Graphical Display package Tkinter and the functionality offered. However, it has arguably become more user friendly by implementing an intuitive 'Home Icon' to revert to the start screen and a Zoom option. Furthermore, this functional update has created ease of user access and produces a transparent and efficient result with the option to begin the process again without exiting the program, a key piece of design functionality (4.2.3) in the initial design. The displayed Wireframes were only an early expectation of the final GUI produced in Part B; as expected, the final design is more detailed and encourages a more user-friendly experience.