Software Design Document

Data Analysis on Australia NSW traffic penalty

Group Number: 43

Student Name: Bozhao Li

Student ID: s2187237

Student Name: Jennifer-wei Lin

Student ID: s5291152

Student Name: Sakibul Islam

Student ID: s5228873

Course Name: Software Technologies

Course code: 7810ICT

Date: 02/09/2022

Assignment: Part A

Table of Contents

1 System Vision

1.1 Problem Background

Road safety is an important aspect of governance, in NSW around 300 people have lost their lives on roads and over 10,000 have been seriously injured during a 12-month period (NSW Government, 2022). The NSW Government had provided a New South Wales (NSW) traffic fines dataset from January 2012 and November 2017. Datasets contain a wealth of information, presented through an excel sheet. However, this presentation is not specific to user needs and can overwhelm with poor presentation and unnecessary information. Therefore, an easy-to-use software should be developed to effectively transform this data, to quickly respond to user query inputs and to graphically show data analysis results.

1.2 System Overview

The system allows users to input a time period, reporting penalty case information. Users are able to further define the search by method of recording by either radar or camera based on the description. Users will additionally be able to produce graphical analysis by entering desired offence code. The system also provides analysis of cases caused by mobile usage spanning the entire database time period of 2011-2017. In order to achieve the above requirements, the following capabilities should be implemented:

- A software loads data from the original dataset.
- A software allows the user to upload an up-to-date dataset.
- A software allows the user to input search options, such as start date, end date, offence codes, offence description, and offence types etc.
- A software is able to analyse the data based on search options entered by the user.
- A software displays the data analysis results in table, graph, or chart format.
- A software allows the user compare results between two different time period.
- A software allows the user to download the graph or chart for further uses.

1.3 Potential Benefits

The project allows for users to quickly and effectively search out case information based on time and other parameters. The users could have a better and deep understanding on key issues listed above. Additionally, graphical and other analysis is computed by the interface to picture possible trends that could lead to new insights or be potential avenues of further research.

2 Requirements

2.1 User Requirements

The primary user of the software would be NSW government and specifically road and transport authorities. The user would want to know the various penalties committed over a specific time-period and location. They would also want to know how the offence was being caught. Total penalty count and total face value can be also shown in the proposed software. The specific user requirements from the perspective of end users are listed below:

- The user should be able to find all offences by selecting start and end dates.
- The user should be able to search for particular offences from the dataset.
- The user would be able to combine two or more offence codes and view the result.
- The user would be able to view both the graph/chart and the actual data.
- The user would be able to change tabs/options and search for different outputs.
- The user would be able to query for data using “Keywords”.
- The user would be able to compare data between two different period.
- The user would be able to view how offences were caught either by camera or by radar.
- The user would be able to know the offences related to mobile phone usage.

2.2 Software Requirements

To achieve desired outcomes, the software needs to perform certain tasks. From the NSW penalty dataset, the user would like to find different information using different search criteria. In order to accomplish its objective, the software shall have these functionalities listed below:

2.2.1 Functional Requirements

2.2.1.1 Search Function

- The program shall search/query the NSW Traffic Penalty data to produce relevant output.
- The program shall accept different offence codes as criteria and show the findings.
- The program shall take start date and end date as input and produce all the offences committed during that period.
- The program shall show how many offences were caught either by camera or by radar.
- By taking offence type as input in search option, the program shall show all the offences within that category.
- The program shall notify when no match is found.

2.2.1.2 Display Function

- After finding the query, the program shall display the output in graph/ chart form.
- The program shall display the relevant actual data in a separate section of the program interface.
- The program shall enable user to add two or more offence types to display the output.

2.2.1.3 Compare Function

- The program shall take two different time period and compare the offence counts in different period.
- Data shall be compared based on offence codes, radar/camera detection and mobile phone usage penalties criteria.

2.2.1.4 Export Function

- The program shall provide the option of downloading the diagrams from successful querying.
- The program shall let user save the data after the search.

2.2.1.5 Upload Function

- The program would let the user upload a different dataset, given all the fields are similar to NSW Traffic Penalty Dataset.
- The program shall show error message if the data is not uploaded successfully or the data format is different.

2.2.2 Non-Functional Requirements

Non-functional requirements will support the functional requirements of NSW Traffic Penalty data analysis system. Table below illustrates the non-functional requirements for the system.

Table 1. Non-functional requirements

Usability	• System UI and user experience is intuitive • System distinguishes clearly between different offence codes • Title of the system is visible • Colour schemes of the system is coherent and ambient • Graph/ chart can be viewed easily • Supports same structured new datasets
Reliability	• System should handle all data without crashes • Original dataset should remain unchanged after queries
Performance	• System should have a response rate of no more than 2s • Should be able to handle combined query

2.3 Use Cases & Use Case Diagrams

It is very useful to understand what the use cases will be for a particular subsystem. In this section, use cases for NSW Traffic Penalty data analysis system has been depicted. These uses cases include from searching the database to downloading the graph from the proposed software. The tables below illustrate use cases.

Table 2. Use Case 1: QUERY THE DATA

Use Case Name	**Query the data**	
Scenario	The user wants to search for penalty offences from dataset	
Triggering event	When the search is submitted using any of the criteria of the software	
Brief Description	The user specifies which penalty they are looking for and then enters dates for the search period	
Pre-Conditions	• The program successfully loaded the database • Appropriate search options were selected	
Post-Conditions	• Data are cleaned for processing • Results displayed in the program	
Flow of Activity	**Actors**	**System**
	1. User selects the dates for query 2. Submitting the search	

	3. User enters extra search conditions 4. User presses the drill down button	2.1 Start date and end date is collected 2.2 The records are fetched based on the time period for the search 2.3 The query results are displayed in the table area 4.1 Extra conditions are collected 4.2 The records are fetched based on the additional conditions 4.3 The query results are displayed using an appropriate diagram in the diagram area
Exceptional Conditions	• Search details are not valid	

Table 3. Use Case 2: VIEW THE GRAPH AND DATA

Use Case Name	**View the graph and data**	
Scenario	The user wants to view the relevant graph/chart and data in more detail from the searched dataset	
Triggering event	Search is valid and search returns relevant data	
Brief Description	The system shows the user graph/chart and the actual data they were looking for	
Pre-Conditions	• Search criteria accepted by the program • Relevant dates are within search parameter	
Post-Conditions	• Graph/chart is displayed • Actual data is shown	
Flow of Activity	**Actors**	**System**
	1. User presses the show detail button 2. User views the graph/ chart	1.1 A diagram with more details is drawn based on the previous search conditions 1.2 A window is popped up to show the diagram
Exceptional Conditions	• Search details are not valid • Relevant chart could not be displayed	

Table 4. Use Case 3: COMPARE THE DATA

Use Case Name	**Compare the data**
Scenario	The user wants to compare data between two different time periods based on offence codes, camera/radar, and mobile phone usage offence.
Triggering event	Drill down button is pressed by selecting date and search criteria

Brief Description	The system shows the user comparison of offence counts in two different time periods according to selected option.	
Pre-Conditions	• One of the 3 options for search is selected • Relevant dates are within search parameter	
Post-Conditions	• Graph/ chart is displayed • Comparison of data is shown on the graph	
Flow of Activity	**Actors**	**System**
	1. User selects either the offence codes, camera/radar or mobile phone usage option 2. Start and end dates are selected	
	3. User presses the drill down button	3.1 Conditions are collected 3.2 Data are processed for visual presentation 3.3 The graphs/charts for comparison between two time periods are shown
	4. After submitting the search, user views the graph/chart	
Exceptional Conditions	• Start and end dates are not within range	

Table 5. Use Case 4: DOWNLOAD THE GRAPH/CHART

Use Case Name	**Download the Graph/chart**	
Scenario	The user wants to download the graph/chart after successful querying	
Triggering event	Download button is pressed after viewing the graph/chart	
Brief Description	The system allows the user to download the graph/chart from the program.	
Pre-Conditions	• Valid search is done • The system displays relevant graph/chart	
Post-Conditions	• Graph/chart is successfully downloaded to local storage	
Flow of Activity	**Actors**	**System**
	1. Download button is pressed	1. A window is popped up to allow user to choose a directory and a file name.
	2. User choose a directory and file name	2. A diagram will be downloaded to the directory.
Exceptional Conditions	• Search did not return any graph/ chart	

Table 6. Use Case 5: UPLOAD A NEW DATASET

Use Case Name	Upload a new dataset	
Scenario	The user wants to upload a new dataset set for data analysis	
Triggering event	Upload is selected from the system menu	
Brief Description	The system allows the user to upload a new dataset given the data fields are structured similar to NSW Traffic Penalty Dataset for data analysis.	
Pre-Conditions	• Dataset is structured similar to NSW Traffic Penalty Dataset • All the required fields are present in the new dataset for analysis	
Post-Conditions	• New dataset allows the user to perform data analysis	
Flow of Activity	**Actors**	**System**
	1. User selects the menu	
	2. Upload is selected	2. A window is popped up to allow user to choose a directory and a csv file.
	3. User browse and selects the new dataset	3.1 System load data from the new dataset and check the format 3.2 The data is loaded successfully and ready for analysis
Exceptional Conditions	• New dataset structure is not similar to NSW Traffic Penalty Dataset • Required fields are not found for data analysis	

The use case diagram for NSW Traffic Penalty Data Analysis System is shown below.

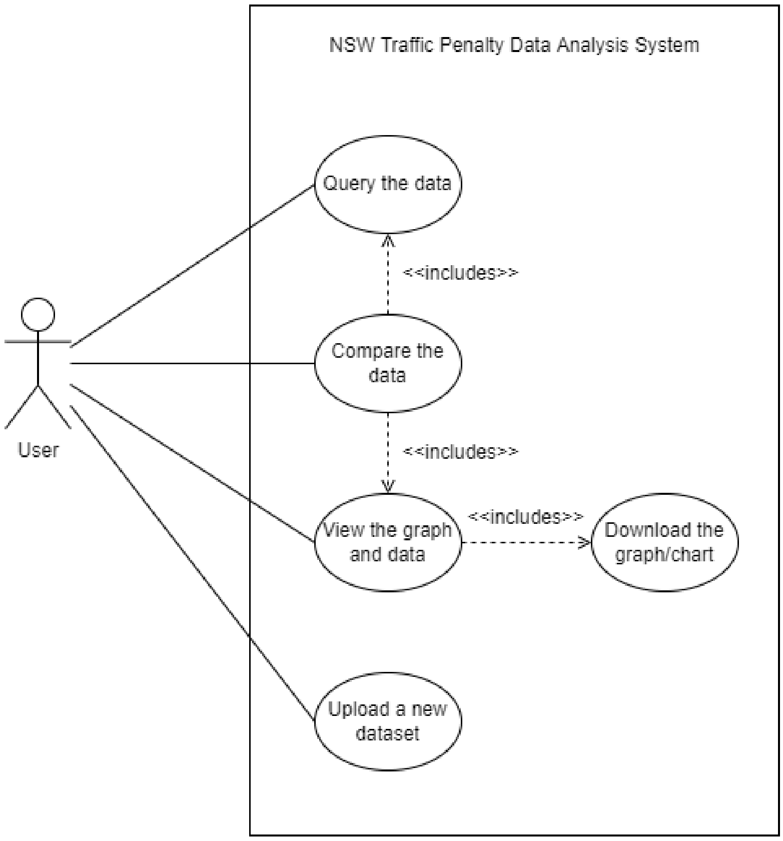


Figure 1: Use case diagram for NSW Traffic Penalty Data Analysis System

3 Software Design and System Components

Based on the above requirements analysis, a software solution has been identified in order to support to further implementation. Detailed software design and software components, which include software block diagram, main functions, data structures, and a brief description of key algorithms, have been provided in this section.

3.1 Software Design

In order to meet customer requirements, the NSW Traffic Penalty Data Analysis System has been designed as a standalone system. The system can be easily installed and work independently without an Internet connection. It can be seen from the Figure 2 that the system configuration file is loaded to set software environment when starting the program. Followed by reading data from a dataset, the records are checked by the program, and invalid data would be removed before the data analysis. When the system is ready to use, the end users can input their business requirements and get data analysis results. Visual views, such as diagrams, tables, and photos, would be used to interact with the end users.

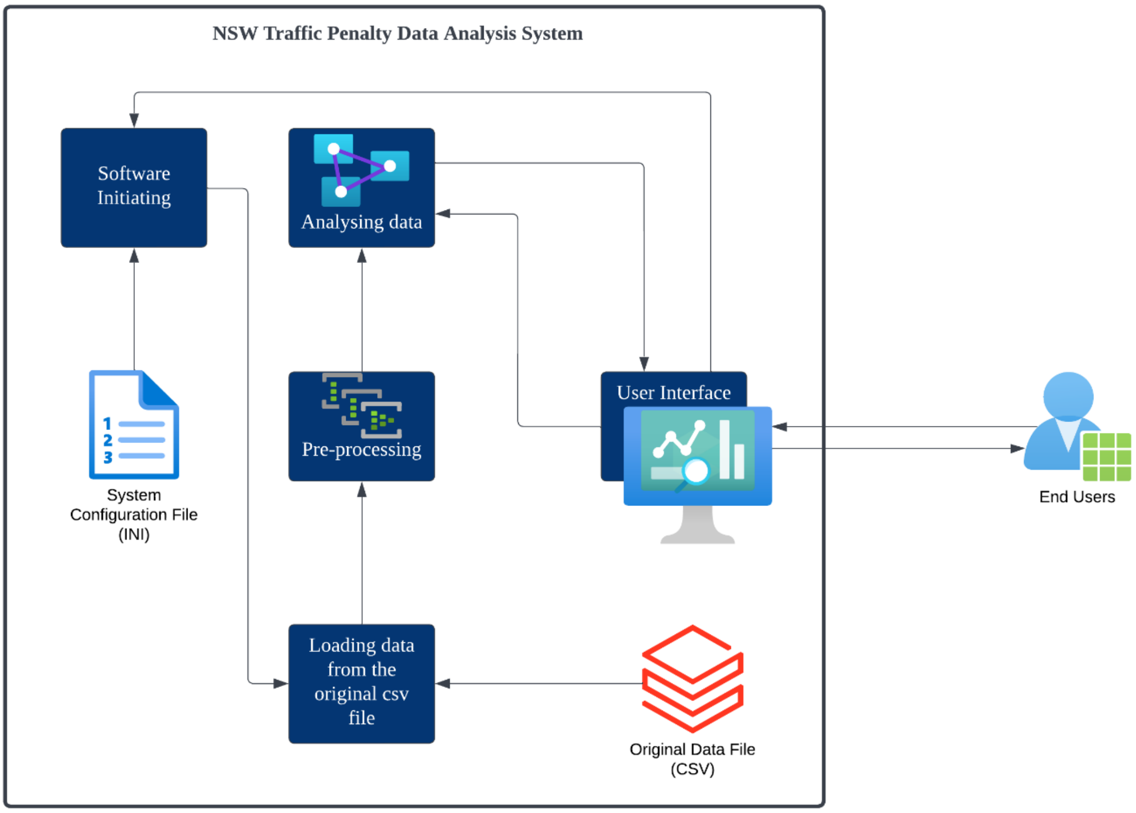

NSW Traffic Penalty Data Analysis System
Software Initiating
Analysing data
System Configuration File (INI)
Pre-processing
User Interface
End Users
Loading data from the original csv file
Original Data File (CSV)

Figure 2. NSW Traffic Penalty Data Analysis System Block Diagram

Even though the system is a standalone system, three-layer architecture called Model-View-Controller (MVC) has been used as a design pattern to facilitate software designs. As this business requirements are not complex, two-layer architecture can also be applied. However, MVC seems to be the best solution when the project team took further functions into account.

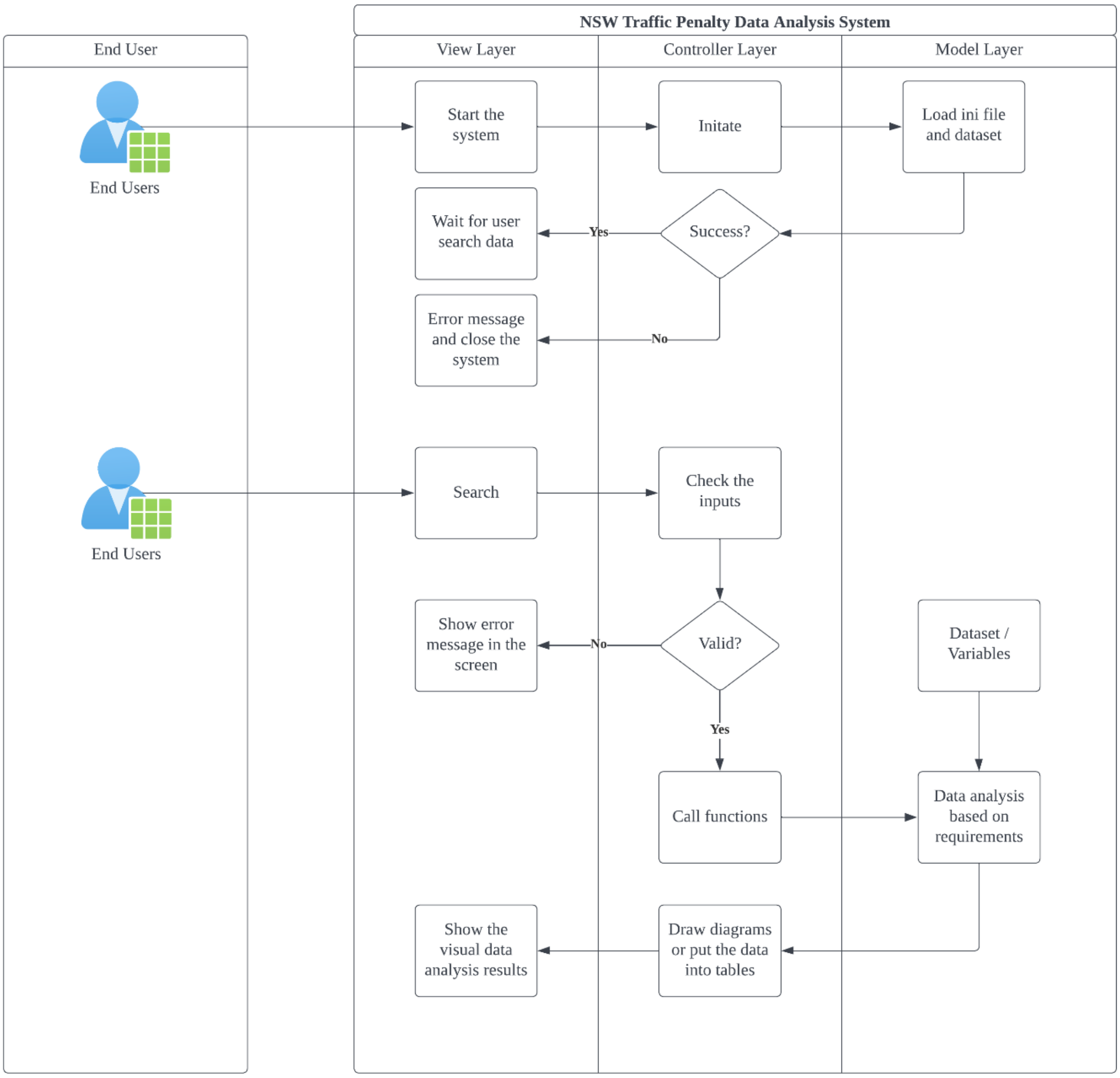

NSW Traffic Penalty Data Analysis System
End User
View Layer
Controller Layer
Model Layer
End Users
Start the system
Initate
Load ini file and dataset
Wait for user search data
Yes
Success?
Error message and close the system
No
End Users
Search
Check the inputs
Show error message in the screen
No
Valid?
Dataset / Variables
Yes
Call functions
Data analysis based on requirements
Show the visual data analysis results
Draw diagrams or put the data into tables

Figure 3. NSW Traffic Penalty Data Analysis System Workflow Diagram

3.2 System Components

3.2.1 Functions

In order to meet the requirements, the NSW Traffic Penalty Data Analysis System would contain the following functions.

Table 7. System Function List

Function No	Function Name	Function Type	
b001	initiate	base	3.2.1.1
b002	load_data	base	3.2.1.2
c001	check_and_clean_data	core	3.2.1.3
c002	format_data	core	3.2.1.4
c003	fetch_data_by_date	core	3.2.1.5
c004	aggregate_data_by_offence_code	core	3.2.1.6

c005	fetch_data_by_offence_description	core	3.2.1.7
c006	fetch_data_by_offence_code	core	3.2.1.8
c007	aggregate_data_by_offence_month	core	3.2.1.9
c008	sort_data	core	3.2.1.10
ui001	show_scatter_diagram	ui	3.2.1.11
ui002	show_bar_diagram	ui	3.2.1.12
ui003	show_line_diagram	ui	3.2.1.13
ui004	set_annotation	ui	3.2.1.14
ui005	annotation_on_move	ui	3.2.1.15

3.2.1.1 initiate

Function Name:	initiate	Function No:	b001
Function Description			
The initiate function will load software configuration file, including meta information (software name, description, and version) and default configuration (dataset file path and name).			
Input Parameters	Name	Data Type	Description
	None		
Side Effects			
None			
Return	Name	Data Type	Description
	iniDict	dict<str, str>	all settings in the configuration file

3.2.1.2 load_data

Function Name:	load_data	Function No:	b002
Function Description			
The load_data function will load all data from the dataset csv file set in the configuration file.			
Input Parameters	Name	Data Type	Description
	filePath	str	dataset file path
	fileName	str	dataset file name
Side Effects			
None			
Return	Name	Data Type	Description
	dataframe	DataFrame	all records read from the dataset file

3.2.1.3 check_and_clean_data

Function Name:	check_and_clean_data	Function No:	c001
Function Description			
The check_and_clean_data function will do data completeness check, data value limits check, data validation check, and clean missing, duplicated and invalid data.			
Input Parameters	Name	Data Type	Description
	dataframe	DataFrame	all records read from the dataset file
Side Effects			
None			
Return	Name	Data Type	Description

	dataframe	DataFrame	checked and cleaned records for further use

3.2.1.4 format_data

Function Name:	format_data	Function No:	c002
Function Description			
The format_data function will format data for further analysis use. In order to improve data analysis efficiency, unnecessary columns would be dropped.			
Input Parameters	Name	Data Type	Description
	dataframe	DataFrame	checked and cleaned records for further use
Side Effects			
As the input dataframe may have huge changes, such as removing many columns, programmers should copy checked and cleaned records before calling this function.			
Return	Name	Data Type	Description
	dataframe	DataFrame	formatted records

3.2.1.5 fetch_data_by_date

Function Name:	fetch_data_by_date	Function No:	c003
Function Description			
The fetch_data_by_date function will fetch records between the start date and the end date selected by the user.			
Input Parameters	Name	Data Type	Description
	dataframe	DataFrame	records
	startDate	Date	selected start date
	endDate	Date	selected end date
Side Effects			
None			
Return	Name	Data Type	Description
	dataframe	DataFrame	records between the start date and the end date

3.2.1.6 aggregate_data_by_offence_code

Function Name:	aggregate_data_by_offence_code	Function No:	c004
Function Description			
The aggregate_data_by_offence_code function will aggregate total numbers of cases group by the offence code.			
Input Parameters	Name	Data Type	Description
	dataframe	DataFrame	records
Side Effects			
None			
Return	Name	Data Type	Description
	dataframe	DataFrame	aggregated records group by offence code

3.2.1.7 fetch_data_by_offence_description

Function Name:	fetch_data_by_offence_description	Function No:	c005
Function Description			
The fetch_data_by_offence_description function will fetch records by offence description which contains key words entered by the user. This function supports a flexible key word input, and the user can give a list of key words.			
Input Parameters	Name	Data Type	Description
	dataframe	DataFrame	records
	listKey	list<str>	a list of selected key word(s)
Side Effects			
None			
Return	Name	Data Type	Description
	dataframe	DataFrame	records which offence description contains any key words entered by the user

3.2.1.8 fetch_data_by_offence_code

Function Name:	fetch_data_by_offence_code	Function No:	c007
Function Description			
The fetch_data_by_offence_code function will fetch records by the offence code selected by the user. This function not only supports the cases data analysis caused by mobile phone, but also provides a flexible input for further new function supports.			
Input Parameters	Name	Data Type	Description
	dataframe	DataFrame	records
	listCode	list<int>	a list of selected offence code(s)
Side Effects			
None			
Return	Name	Data Type	Description
	dataframe	DataFrame	records which offence code is the same as the codes selected by the user

3.2.1.9 aggregate_data_by_offence_month

Function Name:	aggregate_data_by_offence_month	Function No:	c008
Function Description			
The aggregate_data_by_offence_month function will aggregate the total numbers of cases grouped by the offence month.			
Input Parameters	Name	Data Type	Description
	dataframe	DataFrame	records
Side Effects			
None			
Return	Name	Data Type	Description

	dataframe	DataFrame	aggregated records grouped by offence month

3.2.1.10 sort_data

Function Name:	sort_data	Function No:	c008
Function Description			
The sort_data function will sort the records by columns. The user can select how many rows will be shown.			
Input Parameters	Name	Data Type	Description
	dataframe listColumn isAsc total	DataFrame list<str> bool int	records a list of column names asc or desc how many rows will be returned
Side Effects			
None			
Return	Name	Data Type	Description
	dataframe	DataFrame	sorted records

3.2.1.11 show_scatter_diagram

Function Name:	show_scatter_diagram	Function No:	ui001
Function Description			
The show_scatter_diagram function will draw a scatter diagram based on the parameters.			
Input Parameters	Name	Data Type	Description
	self x y	 array-like array-like	window parent data positions data positions
Side Effects			
None			
Return	Name	Data Type	Description
	None		

3.2.1.12 show_bar_diagram

Function Name:	show_bar_diagram	Function No:	ui002
Function Description			
The show_bar_diagram function will draw a bar chart diagram based on the parameters.			
Input Parameters	Name	Data Type	Description
	self x y	 array-like array-like	window parent coordinates of the bars height(s) of the bars
Side Effects			
None			
Return	Name	Data Type	Description
	None		

3.2.1.13 show_line_diagram

Function Name:	show_line_diagram	Function No:	ui003
Function Description			
The show_line_diagram function will draw a line chart diagram based on the parameters.			
Input Parameters	Name	Data Type	Description
	self		window parent
	x	float	point (x, y) to annotate
	y	array-like	point (x, y) to annotate
Side Effects			
None			
Return	Name	Data Type	Description
	listAnnotation	list	list of annotation

3.2.1.14 set_annotation

Function Name:	set_annotation	Function No:	ui004
Function Description			
The set_annotation function will set an annotation to diagrams based on the parameters.			
Input Parameters	Name	Data Type	Description
	self		window parent
	x	array-like	Horizontal coordinates of the data points
	y	array-like	Vertical coordinates of the data points
Side Effects			
None			
Return	Name	Data Type	Description
	None		

3.2.1.15 annotation_on_move

Function Name:	annotation_on_move	Function No:	ui005
Function Description			
The annotation_on_move function will show an annotation in the diagrams when the mouse is moved on the annotation points.			
Input Parameters	Name	Data Type	Description
	event	EVENT object	event object
Side Effects			
None			
Return	Name	Data Type	Description
	None		

3.2.2 Data Structures / Data Sources

In this software, the following main data structures would be used to meet the business requirements.

Table 8. System Data Structure Lists

Data Type	
dictionary (key, value)	3.2.2.1
list	3.2.2.2
pandas.DataFrame	3.2.2.3
NumPy.array	3.2.2.4

3.2.2.1 dictionary (key, value)

<table>
<tr><td>Data Type:</td><td>dictionary (key, value)</td></tr>
<tr><td colspan="2">Data Type Description</td></tr>
<tr><td colspan="2">The dictionary (key, value) data type will be used to store system configuration settings, such as software name, description, version, dataset file path and file name.</td></tr>
<tr><td colspan="2">List of functions</td></tr>
<tr><td colspan="2">load_data</td></tr>
</table>

3.2.2.2 list

<table>
<tr><td>Data Type:</td><td>list</td></tr>
<tr><td colspan="2">Data Type Description</td></tr>
<tr><td colspan="2">The list data type will be used to store inputs or outputs, such as search offence keywords, columns, codes for inputs, data results and diagram annotations for outputs.</td></tr>
<tr><td colspan="2">List of functions</td></tr>
<tr><td colspan="2">fetch_data_by_offence_description()
fetch_data_by_offence_code()
sort_data()
show_annotation()
show_wordCloud()</td></tr>
</table>

3.2.2.3 pandas.DataFrame

<table>
<tr><td>Data Type:</td><td>pandas.DataFrame</td></tr>
<tr><td colspan="2">Data Type Description</td></tr>
<tr><td colspan="2">The pandas.DataFrame data type will be used to store both records read from the dataset and data analysis results.</td></tr>
<tr><td colspan="2">List of functions</td></tr>
<tr><td colspan="2">load_data()
check_and_clean_data()
format_data()
fetch_data_by_date()
aggregate_data_by_offence_code()
fetch_data_by_offence_description()
fetch_data_by_offence_code()
aggregate_data_by_offence_month()
sort_data()
show_wordCloud()</td></tr>
</table>

3.2.2.4 NumPy.array

<table>
<tr><td>Data Type:</td><td>NumPy.array</td></tr>
<tr><td colspan="2">Data Type Description</td></tr>
</table>

The NumPy.array data type will be used to store data analysis results for drawing diagrams.
List of functions
show_scatter_diagram () show_bar_diagram () show_line_diagram () set_annotation()

3.2.3 Detailed Design

In this section, two pseudocodes have been offered. One is for checking and cleaning data, and another is for data analysis of radar or camera.

3.2.3.1 Pseudocode for checking and cleaning data

```
Begin

    Input DataFrame

    Check whether key columns are missing

    IF missing key columns:

        drop these rows

    Check whether duplicated rows exist

    IF duplicated rows found:

        drop these rows

    Check whether invalid values exist

    IF invalid values found:

        drop these rows

    Return cleaned DataFrame

END
```

3.2.3.2 Pseudocode for data analysis of radar or camera

```
Begin

    Input user search conditions

    Get cleaned DataFrame

    Call fetch_data_by_date()

    Call format_data()

    Call fetch_data_by_offence_description()
```

```
        Call aggregate_data_group_by_offence_code()
        Swap DataFrame columns <==> rows
        Return DataFrame
    END
```

4 User Interface Design

The structure and hierarchy diagram, created with Microsoft Office SmartArt, showcases the flow of the interface from least specified user inputs to most from left to right. The wireframe was created with PyQt5 Designer which effectively shows the general screen designs of the interface including layout, visual elements, icons, graphics, style, colour, and fonts etc. In this section, structural design and visual design will be shown.

4.1 Structural Design

The interface structure and hierarchy (see Figure 4.) has been created in response to the degree of specificity in the required features. The most immediate user inputs from the main panel are the two dates. Once the search button is hit, users are presented with text results in the first table and a graphical presentation in the second which is exportable. Users will be able to switch between four tabs positioned between the text and graphical outputs to identify their area of focus. Located within each are further specifications uniquely chosen to refine information within each area. Finally, users can compare between tabs.

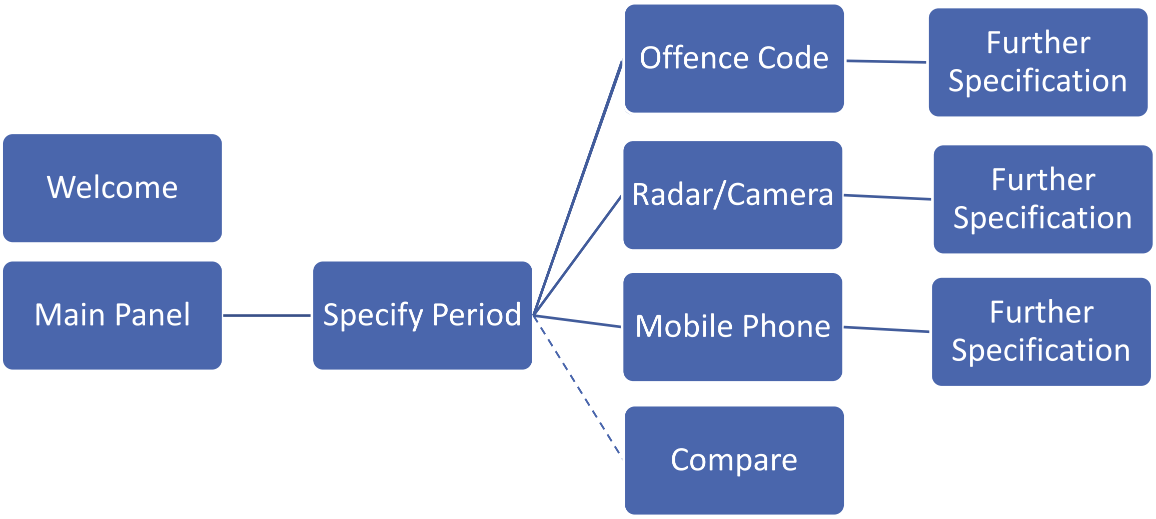
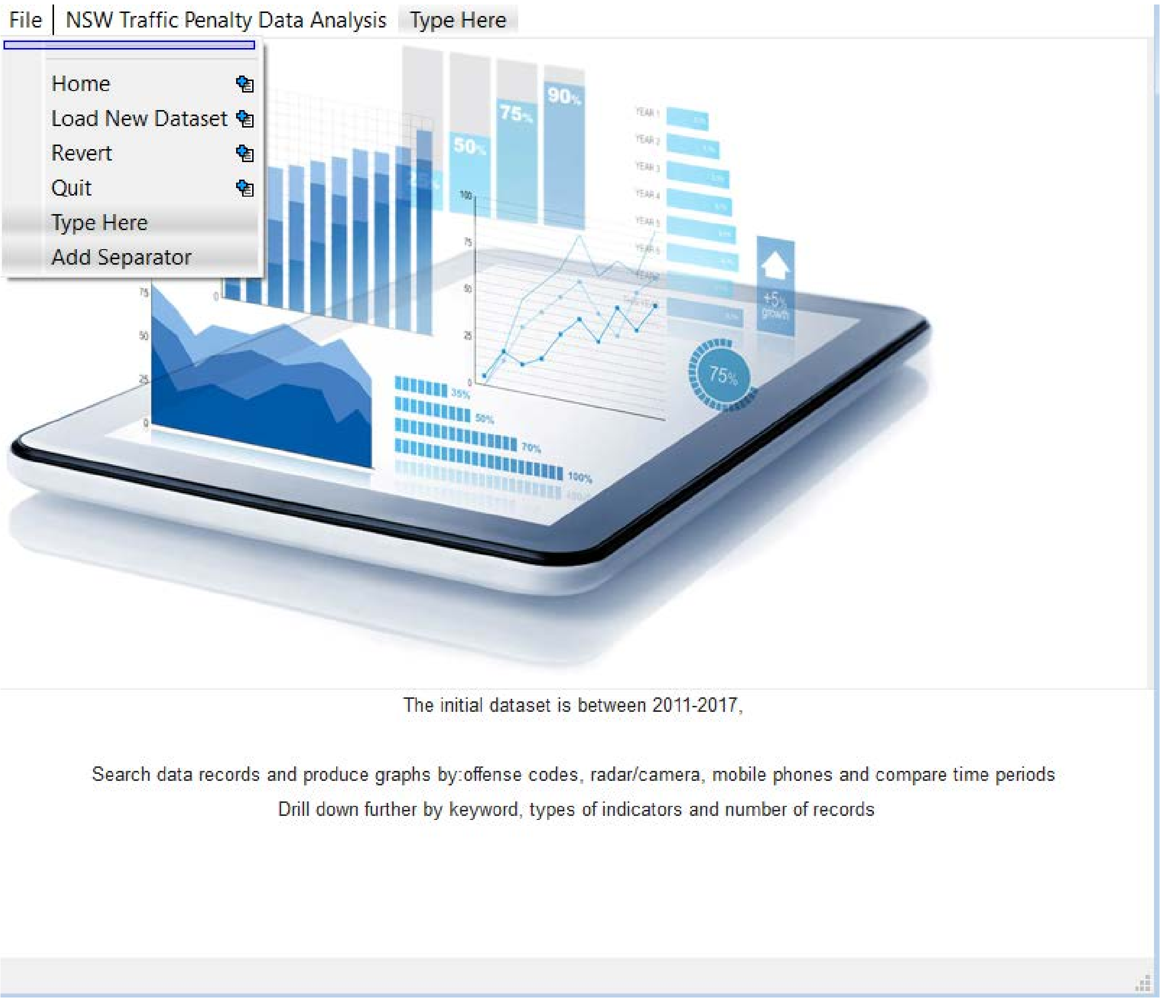


Figure 4. Structure and Hierarchy Diagram

The welcome screen (see Figure 5.) will briefly introduce the software, highlight the parameters and functionality of the interface. The software loads a dataset file, either the original dataset file or a dataset uploaded by the user, and shows the time period of the dataset.

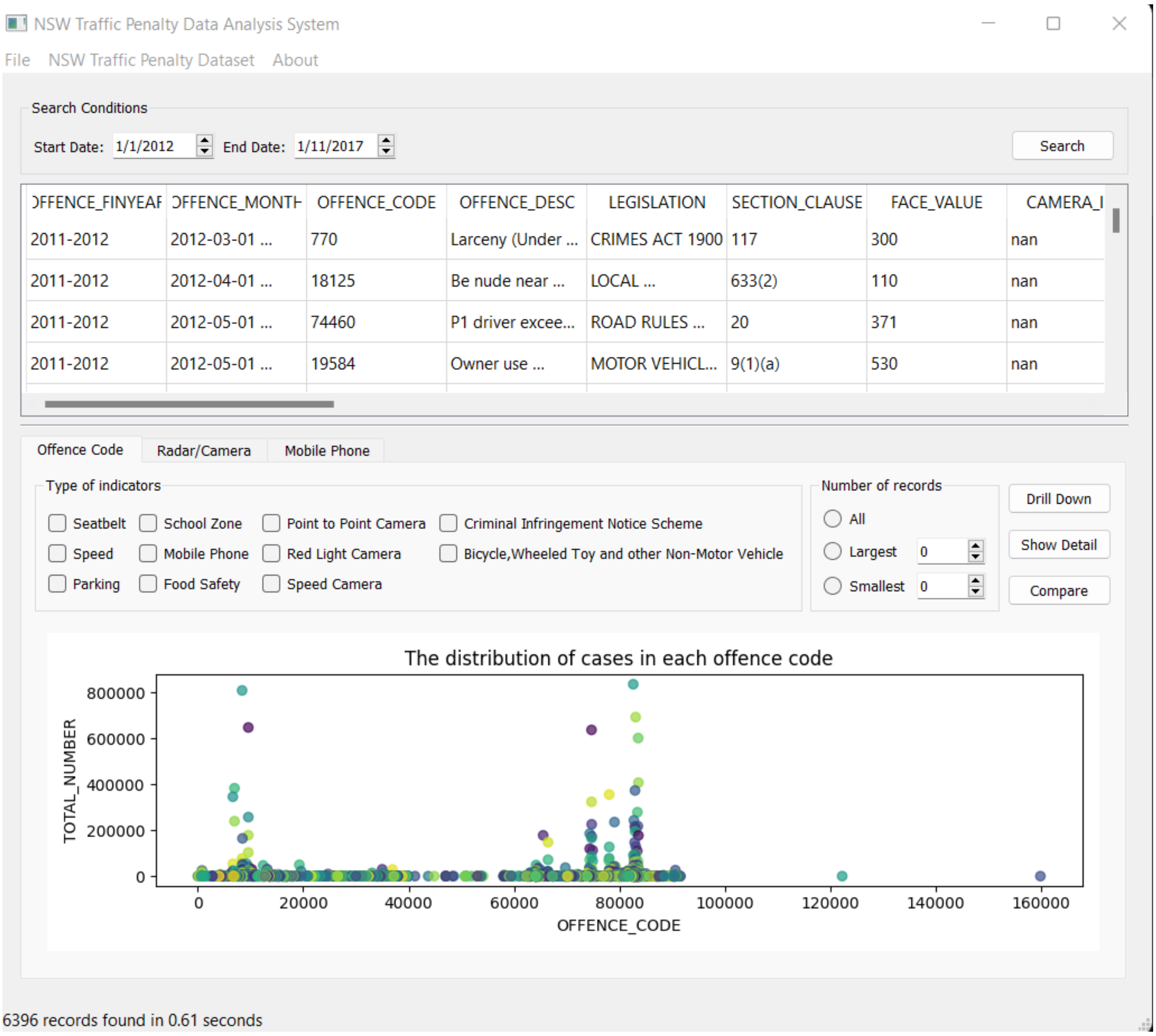


The initial dataset is between 2011-2017,

Search data records and produce graphs by:offense codes, radar/camera, mobile phones and compare time periods
Drill down further by keyword, types of indicators and number of records

Figure 5. The Wireframe of the Welcome Screen

Moving on from the Welcome screen to the right brings the user to the main program. If new dataset is loaded, the program will also check the header and change it accordingly from the default “NSW Traffic Penalty Data Analysis” (see Figure 6).



Start Date: 1/1/2012 End Date: 1/11/2017

)FFENCE_FINYEAF	)FFENCE_MONTH	OFFENCE_CODE	OFFENCE_DESC	LEGISLATION	SECTION_CLAUSE	FACE_VALUE	CAMERA_I
2011-2012	2012-03-01 ...	770	Larceny (Under ...	CRIMES ACT 1900	117	300	nan
2011-2012	2012-04-01 ...	18125	Be nude near ...	LOCAL ...	633(2)	110	nan
2011-2012	2012-05-01 ...	74460	P1 driver excee...	ROAD RULES ...	20	371	nan
2011-2012	2012-05-01 ...	19584	Owner use ...	MOTOR VEHICL...	9(1)(a)	530	nan

□ Seatbelt □ School Zone □ Point to Point Camera □ Criminal Infringement Notice Scheme
□ Speed □ Mobile Phone □ Red Light Camera □ Bicycle,Wheeled Toy and other Non-Motor Vehicle
□ Parking □ Food Safety □ Speed Camera

○ All
○ Largest 0
○ Smallest 0

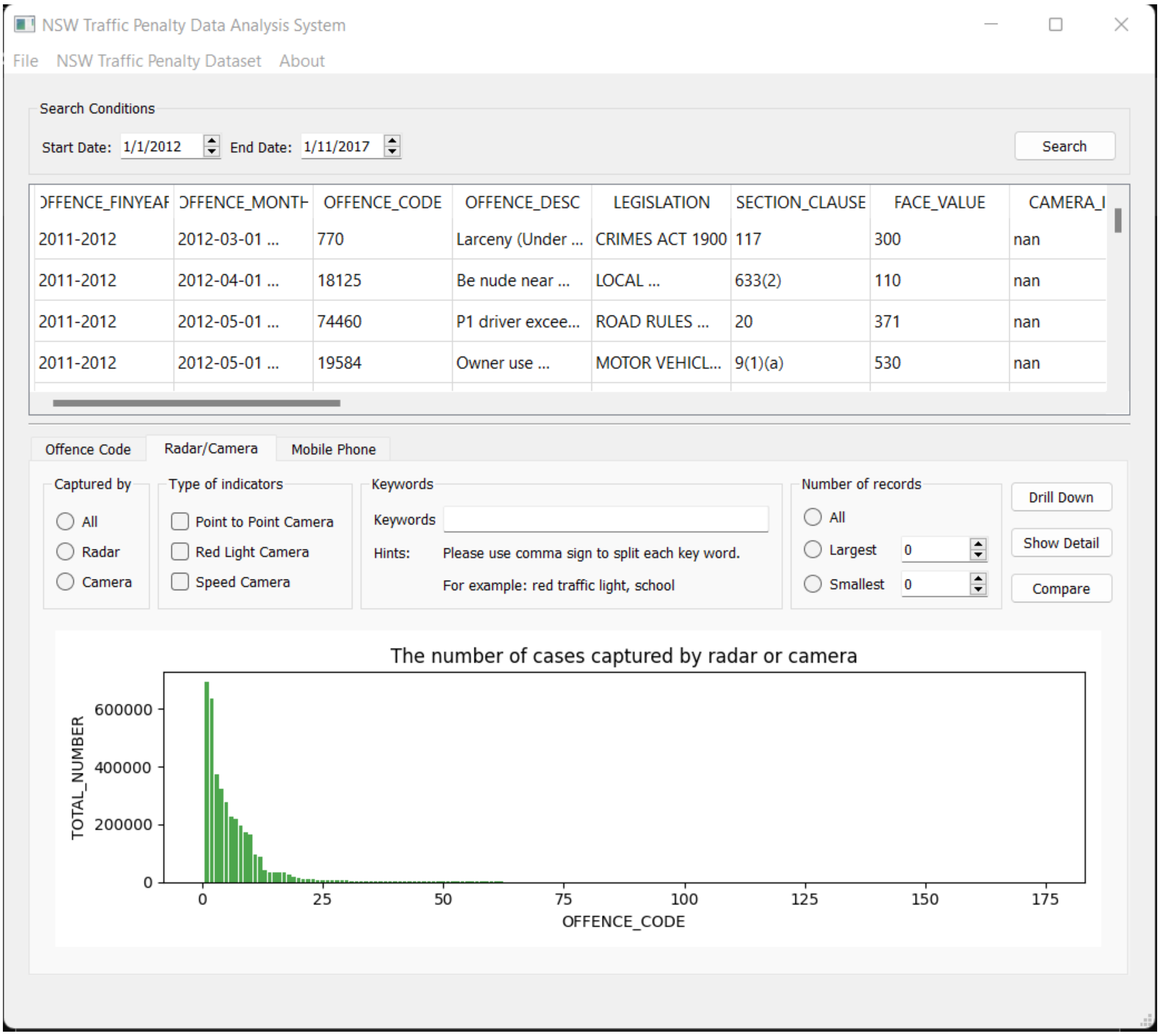
6396 records found in 0.61 seconds

Figure 6. The Wireframe of the Data Analysis Screen

The interface structure has been created in response to the degree of specificity in the required features. The most immediate user inputs are the two dates. Once the search button is hit, users are presented with text results in the first table and a graphical presentation in the second which is exportable through the Show Detail button. The Compare button also allows users to compare graphs from two different tabs.

The default tab is penalty case by offense code. In the graphical section of the interface, users will further be able to drill down by specifications including types of indicators and number of records.

The second tab (see the below Figure 7) services more specific user requirements of retrieving all cases captured by radar or camera based on offence description. Users will also be able to drill down by captured by, type of indicators, keywords and number of records.



NSW Traffic Penalty Data Analysis System

Search Conditions

Start Date: 1/1/2012 End Date: 1/11/2017

)FFENCE_FINYEAF	)FFENCE_MONTH	OFFENCE_CODE	OFFENCE_DESC	LEGISLATION	SECTION_CLAUSE	FACE_VALUE	CAMERA_I
2011-2012	2012-03-01 ...	770	Larceny (Under ...	CRIMES ACT 1900	117	300	nan
2011-2012	2012-04-01 ...	18125	Be nude near ...	LOCAL ...	633(2)	110	nan
2011-2012	2012-05-01 ...	74460	P1 driver excee...	ROAD RULES ...	20	371	nan
2011-2012	2012-05-01 ...	19584	Owner use ...	MOTOR VEHICL...	9(1)(a)	530	nan

Captured by

○ All
○ Radar
○ Camera

Type of indicators

□ Point to Point Camera
□ Red Light Camera
□ Speed Camera

Keywords

Keywords

Hints: Please use comma sign to split each key word.

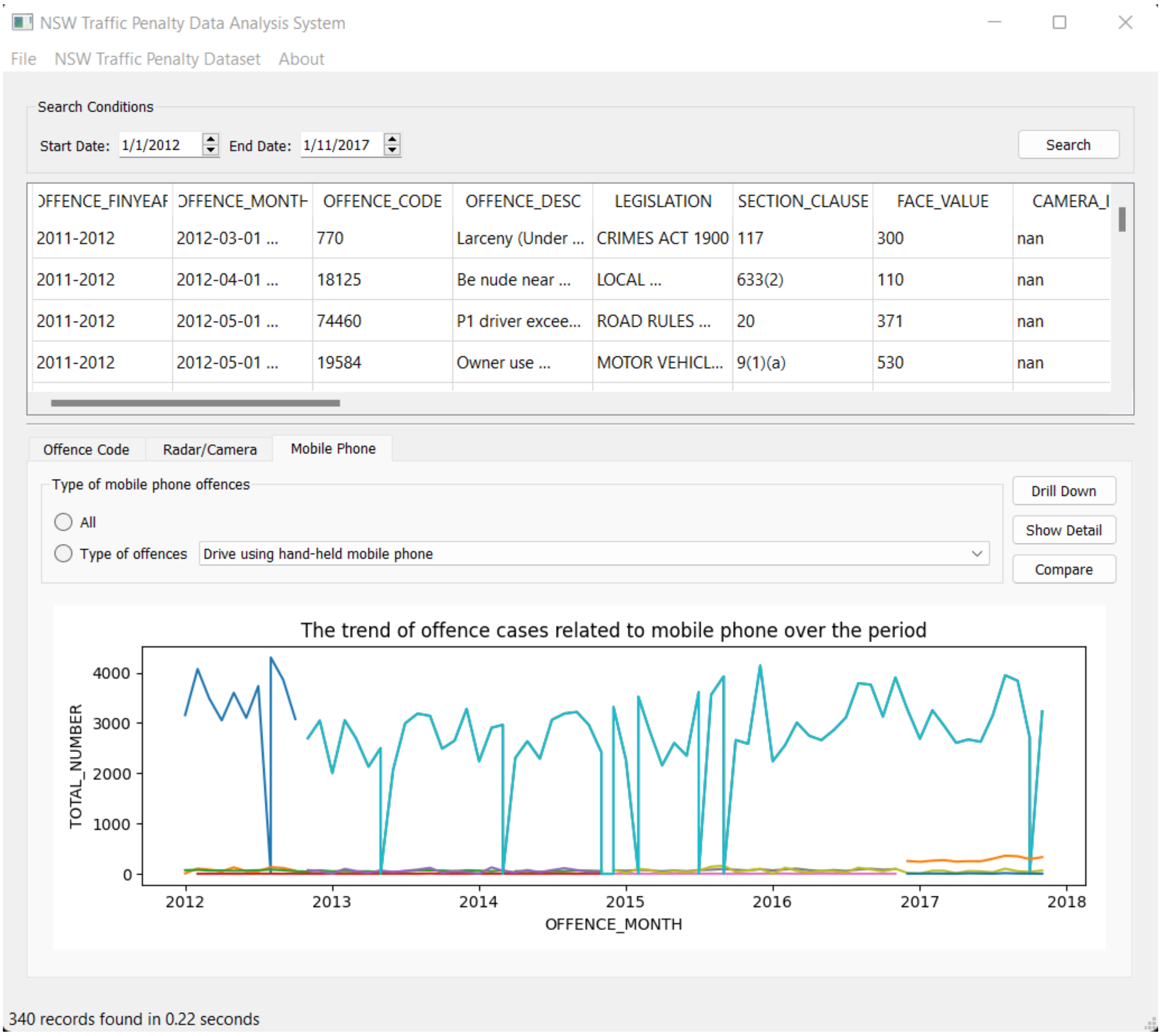
For example: red traffic light, school

Number of records

○ All
○ Largest 0
○ Smallest 0

Figure 7. The Wireframe of the Data Analysis Screen (Radar/Camera Tab)

The third tab (Figure 8) services the user requirement of analysing the cases caused by mobile phone usage. Users will be able to further refine their search by types of offenses and number of records.



NSW Traffic Penalty Data Analysis System

File NSW Traffic Penalty Dataset About

Search Conditions

Start Date: 1/1/2012 End Date: 1/11/2017 Search

)FFENCE_FINYEAF	)FFENCE_MONTH	OFFENCE_CODE	OFFENCE_DESC	LEGISLATION	SECTION_CLAUSE	FACE_VALUE	CAMERA_I
2011-2012	2012-03-01 ...	770	Larceny (Under ...	CRIMES ACT 1900	117	300	nan
2011-2012	2012-04-01 ...	18125	Be nude near ...	LOCAL ...	633(2)	110	nan
2011-2012	2012-05-01 ...	74460	P1 driver excee...	ROAD RULES ...	20	371	nan
2011-2012	2012-05-01 ...	19584	Owner use ...	MOTOR VEHICL...	9(1)(a)	530	nan

Offence Code | Radar/Camera | Mobile Phone

Type of mobile phone offences

○ All

○ Type of offences: Drive using hand-held mobile phone

Drill Down | Show Detail | Compare

340 records found in 0.22 seconds

Figure 8. The Wireframe of the Data Analysis Screen (Mobile Phone Tab)

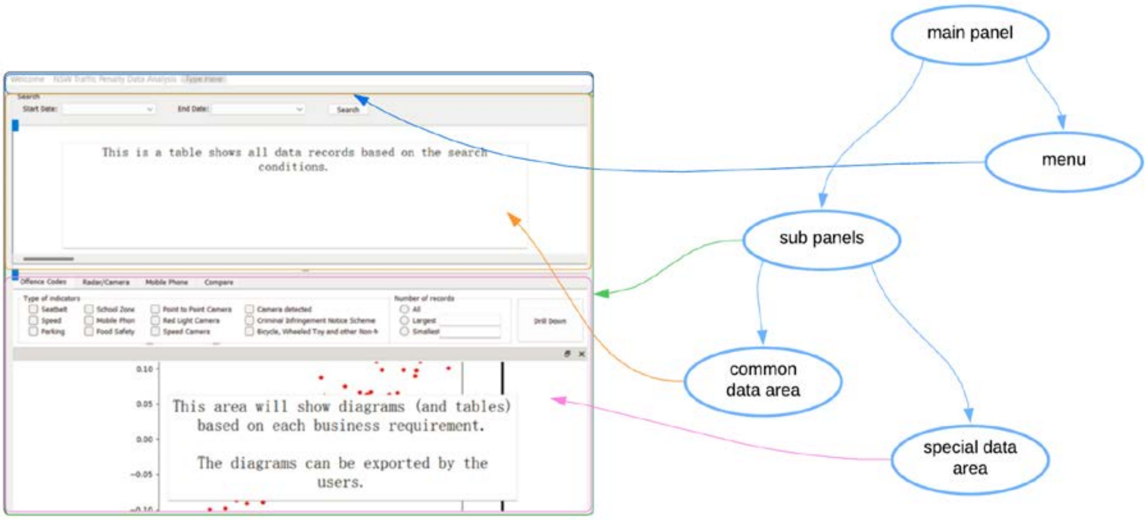
The hierarchy of options based on degree of specificity allows for an intuitive and natural flow that most effectively aligns to the proximity of user intention. Users that only require a general overview of information need not sift through confounding specific options and vice versa.

4.2 Visual Design

The screen has been designed to most effectively showcase text and graphical outputs created by user selected confines. The layout is split into two main sections, the top half for displaying text outputs and the bottom half for displaying graphical representations. The icons for user inputs sandwiches the text section. This consolidation allows for users to quickly and effectively create the confines of the search in an intuitive row. The default tab the user will see has the penalty cases by offense code. Users can easily switch to tabs with more specified information options by clicking to the right without needing to reinput date constraints.

The graphical section takes up a bigger section because it is a simple representation of information that will be the centre of users attention, they can check the graphical output against the finer detail in the above section. The kind of graph displayed differs by their suitability to the confines of user requirements in each tab.

All labels are in the font MS Shell to promote consistency in style. The choice for more neutral colours allows for clearer focus to be on information outputs and avoids clashes with the bold colours in graphical presentation.



This is a table shows all data records based on the search conditions.

This area will show diagrams (and tables) based on each business requirement.

The diagrams can be exported by the users.

Figure 10. The User Interface Layout

References

John, W. S., Robert, B. J., & Stephen, D. B. (2015). *Systems analysis and design in a changing world* (7th ed.). Cengage Learning.

NSW Government. 2022. *NSW Centre for Road Safety >Statistics*. Retrieved August 9, 2022 from https://roadsafety.transport.nsw.gov.au/statistics/index.html