Software Design Document
New South Wales Traffic and Penalty Analysis Tool

Daehoon Seong

Waraphorn Srisomboonkamol

Justin Bilao

Table of Contents

1.0 System Vision

1.1. Problem Background

In the state of New South Wales (NSW), traffic violations and offenses have become a persistent challenge. As the region continues to grow, traffic authorities, policymakers, and law enforcement agencies now have access to thousands of raw traffic data for analysis. While traditional analysis methods offer some insights, they lack the depth and customization capabilities needed to understand offense trends, especially when considering the financial implications of penalties or violations. The complexity of current data analysis tools introduces a barrier to entry for initiating data analysis on traffic data. This project will be focusing on creating an intuitive and user-friendly data analysis tool for traffic data.

1.2. System Overview/Capabilities

Data Filtering and Selection

- Enable users to select specific time periods for reporting traffic offenses.
- Offer an intuitive text-based widget for easy period selection.
- Allow users to filter data by specific offense codes.

Data Display and Reporting

- Display a comprehensive table of traffic offenses, including but not limited to offense code, description, and penalty amount, for the user-defined period.
- Allow re-entering and modifying filters for data display.

Data Visualization

- Generate pie-charts to display the distribution of cases by offense codes for selected periods.
- Display line-graphs to show trends over time for mobile-related offenses.
- Offer hover-over descriptions for offense codes in all graphical representations.

Advanced Filtering for Special Cases

- Enable users to filter and display data on cases captured by specific types of radars or cameras.
- Enable advanced filtering options based on the description of the offense.

Mobile Usage Offense Analysis

- Allow users to select and re-select mobile-related offense codes for specialized data analysis.

- Provide a line-graph representation showing the number of cases for each selected mobile-related offense code over time.

Financial Analysis

- Provide tools to assess the total fees collected per offense code for a user-defined period.
- Display the top 10 offense codes based on total collected fees in a descending order using a bar graph or table.

Data Exporting/Reporting

- Offer options for exporting visualizations, tables, or raw data for external analysis.

1.3. Potential Benefits

Public Safety Information

- **Enhanced Public Awareness**: The system can serve as an educational tool for the general public, making them more aware of high-risk offenses and potentially reducing the occurrence of such offenses.
- **Targeted Safety Initiatives**: Authorities can use the data to prioritize safety initiatives, focusing on specific offenses or hotspots identified through the analysis.

Cost-Effectiveness

- **Resource Optimization**: By identifying trends and hotspots, law enforcement agencies can allocate resources more effectively, potentially reducing costs.
- **Administrative Efficiency**: Automation and advanced filters reduce the need for manual data sorting and analysis, freeing up administrative staff for other tasks.

Policy and Governance

- **Data-Driven Policymaking**: Policymakers can utilize the detailed analysis for more data-driven decision-making, leading to more effective and targeted policies.
- **Transparency and Accountability**: The system can also be used to generate public reports, thereby improving transparency and public trust.

User Experience

- **Intuitive Interface**: The system's user-friendly interface and visualizations can make complex data understandable for non-experts, broadening the range of personnel who can use the system effectively.

Financial Analysis

- **Revenue Tracking**: The financial analysis tools can provide insights into the effectiveness of current penalty systems and may offer suggestions for adjustments to optimize revenue without compromising fairness or public safety.

2.0 Requirements

2.1. User Requirements

User Interaction Overview

In the role of a data analyst for the New South Wales Traffic Authority, the user—let's name him Danny—opens the data analysis tool. John will be seeing a main menu that displays available tools for data analysis on traffic data.

Effortless Navigation

Requirement: The dashboard and user interface shall be designed for intuitive navigation.

How Users Interact: John should be able to effortlessly locate key features such as "Period Selection," "Offense Filters," "Financial Analysis," "Mobile Offense Trends," and "Data Visualization" within the dashboard. Icons and labels shall be used to signify the purpose and utility of each section clearly.

Dynamic Data Filtering

Requirement: The system shall provide dynamic data filtering options.

How Users Interact: John should be able to easily input a specific time frame, offense codes, and types of capture methods (radar or camera), among other filters. The system shall allow for re-entry or modification of these filters for refined data analysis.

Informative Data Presentation

Requirement: The system shall display data in tables and charts, enriched with informative tooltips.

How Users Interact: Upon selecting his filters, John can view detailed tables and charts. Hovering over specific data points or codes should display additional information, such as offense descriptions and penalty amounts, for enhanced understanding. He can also export the new tables into spreadsheet or PDF format.

2.2. Software Requirements

Functional Requirements

1. The program shall allow users to select a specific time-period for reporting penalty cases.
    1.1. The system shall provide an intuitive text-based widget for period selection
    1.2. The system shall provide a filter for the offence code
    1.3. The system shall provide a table to display the list information of offences based on the offence code and period
    1.4. The system shall allow for exporting the displayed charts/tables

1.5. The system shall display the description of the offence code when the user is hovering the mouse pointer on the offence code
1.6. The system shall display other information of the offence code when the user mouse clicks on the offence code

2. The program shall enable users to generate charts showing the distribution of cases based on offense codes for a selected period.

    2.1. The information shall include but not be limited to the offense code, offense description, and penalty amount.
    2.2. The system shall display a pie-chart displaying the distribution of the number of cases for each offence
    2.3. The system shall display the description of the offence code when the user is hovering the mouse pointer on the offence code
    2.4. The system shall display other information of the offence code when the user mouse clicks on the offence code
    2.5. The system shall allow for exporting the displayed charts/tables
    2.6. The system shall display the description of the offence code when the user is hovering the mouse pointer on the offence code
    2.7. The system shall display other information of the offence code when the user mouse clicks on the offence code

3. The program shall provide a filter to retrieve all cases captured by radar or camera based on offense descriptions within a user-selected time frame.
    3.1. The filter shall require inputting a user-selected timeframe
    3.2. The filter shall include options to select specific types of cameras or radars.
    3.3. The system shall have the option to filter by offence description
    3.4. The system shall display a table based on the pre-selected filters
    3.5. The system shall allow re-entering filters for display
    3.6. The system shall allow for exporting the displayed charts/tables
    3.7. The system shall display the description of the offence code when the user is hovering the mouse pointer on the offence code
    3.8. The system shall display other information of the offence code when the user mouse clicks on the offence code

4. Analysing the cases caused by mobile phone usage.
    4.1 The system shall include an option to display a chart of Mobile Usage Offence Over Time
        4.1.1. The system shall require inputting the type of Mobile Usage Offence to display
    4.2 The system shall include an option to display a table the Top 10 Mobile Usage Offences Per Year
        4.2.1. The system shall require inputting the year to display
    4.3 The system shall allow re-selecting offence types and year for display

4.4 The system shall display selected options of charts/tables
4.5 The system shall allow for exporting the displayed charts/tables
4.6 The system shall display the description of the offence code when the user is hovering the mouse pointer on the offence code
4.7 The system shall display other information of the offence code when the user mouse clicks on the offence code

5. The program shall offer financial analysis tools to assess the total fees collected per offense code based on a selected period.
    5.1 The filter shall require inputting a user-selected timeframe
    5.2 The system shall include an option to display a table of locations with highest total amount of penalties in descending order.
    5.3 The system shall include an option to display a table of offences with highest total amount of penalties in descending order

2.3. Use Cases & Use Case Diagrams

The Fully Developed Use Case represents a technique for documenting and explaining a use case. This approach has gained extensive adoption for comprehensively understanding user needs and is commonly employed by software developers. It has the specific scenario involved in the use case, along with a brief description of the action's context. It is also included pre-conditions and post-conditions in order to address the conditions and critical information that need to be valid before and after the use case begins and ends. In addition, the sequence of actions links the actor and the system where the actor carries out the activity while the system responds appropriately to the actor's request. Finally, the except condition identifies any potential errors that may arise during the activities.

- See the next page -

Use Case Table #1

Use case name:	Choose reporting period.	
Scenario:	User selects a specific time period for reporting penalty cases.	
Brief description:	User wants to access the penalty cases report by selecting a specific time period. The option to choose a reporting period enables users to filter the offense codes, and the table will display information related to these offense codes.	
Actors:	User (Officer)	
Preconditions:	Users have successfully logged into the system. Users have appropriate permissions to access penalty cases reporting functionality.	
Postconditions:	User has selected a specific time period for reporting penalty cases. System must display an intuitive text-based widget for period selection. System must provide a filter for selecting the offense code. System must generate and display a table listing information about offenses, based on the selected offense code and the chosen period.	
Flow of activities:	Actor	System
	1. Users initiate to generate a report of penalty cases by clicking 'Overall Report'. 2.Users select a specific time period. 3.Users select the offense code filter option to refine the report. 4.Users submit the selected time period and offense codes by press 'submit'	1.System presenting the user with options for generating reports. 2.System provides an intuitive text-based widget allowing the user to input or select a time period. 3.System provides a filter mechanism allowing the user to select specific offense codes. 4.System validates the inputs and processes requests. 4.1 System retrieves relevant data based on selected time period and offense codes. 4.2 System generates a table of information of offenses. 4.3 System displays the table to the user as per selected filtering criteria.
Exception conditions:	3. Invalid time period input. 4. Input validation error. 5. No data found and technical error.	

Use Case Table #2

Use case name:	Generate charts.	
Scenario:	User generates an offense code distribution chart for a user-selected period.	
Brief description:	Users can generate offense code distribution charts for selected periods, and the pie-chart will display the number of cases for each offense codes.	
Actors:	User (Officer)	
Preconditions:	User has logged into the system with valid credentials. User has access rights to view case distribution charts and related data. Data related to offense codes, description, and penalty amount is available in the system for the selected period.	
Postconditions:	User has successfully generated charts that illustrate the distribution of cases based on offense codes for the selected period. System must effectively display charts including offense codes, description, and penalty amounts. A pie chart must display and represent the distribution of the number of cases for each offense codes. The description of the offense code must display when user hovers the mouse pointer over on the chart. The additional information of specific offense code must display when user clicks on an offense code on the chart.	
Flow of activities:	Actor	System
	1.Users initiate to generate charts of distribution cases based on offense codes by clicking 'Chart Distribution'. 2.Users select period to analyze the case distribution. 3.Users request the system to generate charts with specific information by click 'submit' 4.Users hover the mouse pointer over offense code on the chart. 5.Users click on the offense code on the chart.	1.System presenting options for generating charts. 2.System acknowledges selected time frame. 3.System processes request and prepares to display the charts. 3.1 System retrieves data related to cases, offense codes, description, and penalty amounts for selected period. 3.2 System organizes the data for generating the charts. 3.3 System generates charts and presents to the user. 4.System displays the description of the offense code. 5.System presents additional information of the specific offense code.
Exception conditions:	1. Insufficient permissions. 4. Data unavailability and slow performance.	

Use Case Table #3

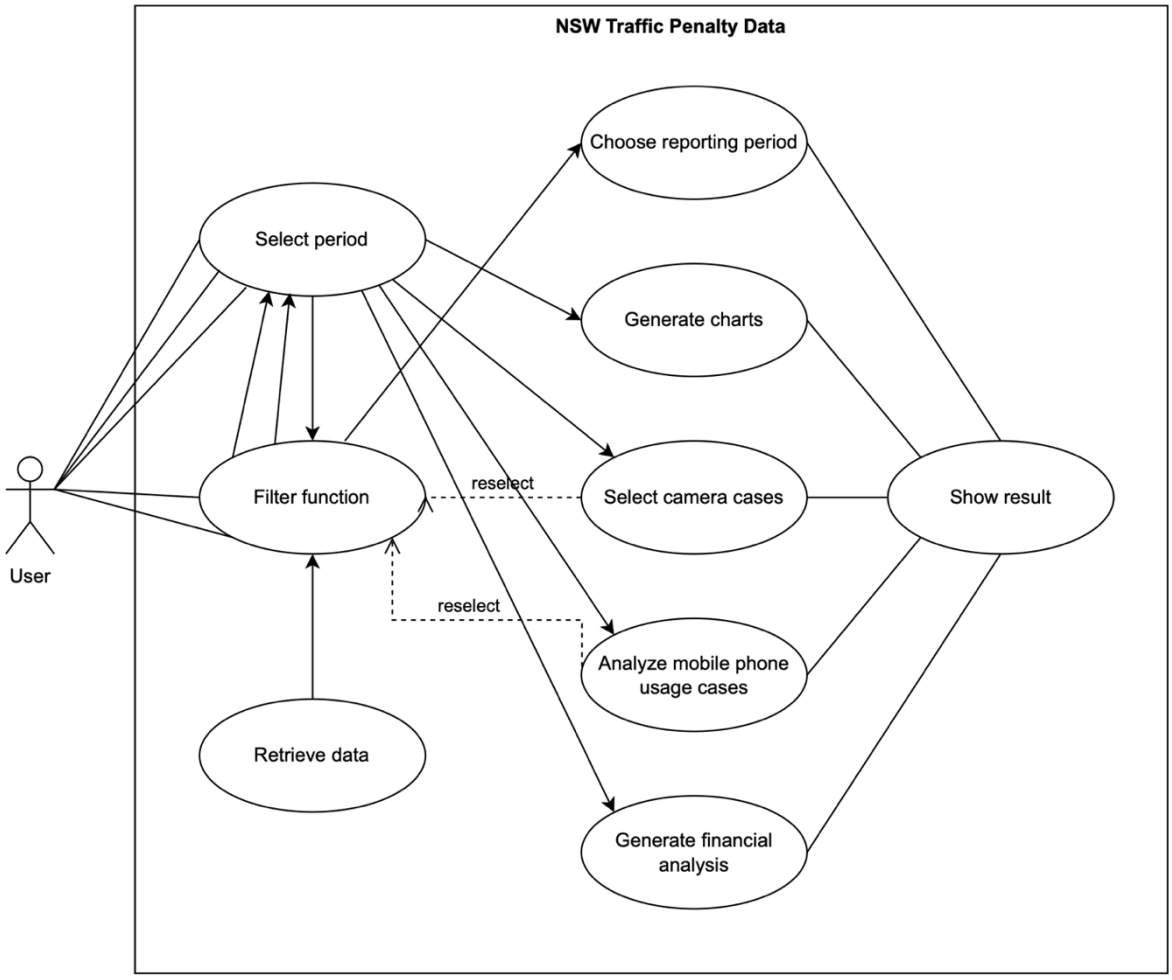
Use case name:	Select camera cases.	
Scenario:	User filters radar or camera cases by offense description and time frame.	
Brief description:	Users can utilize a filter to retrieve cases captured by radar or camera within a user-selected time frame, the table will display data based on pre-selected filters and allow users to re-enter filters if needed.	
Actors:	User (Officer)	
Preconditions:	User has logged into the system with valid credentials. User has access permissions to view and retrieve case data. Data related to cases captured by radar or camera, offense descriptions, and time frames is available in the system.	
Postconditions:	User has successfully retrieved cases captured by radar or camera based on offense descriptions within the selected time frame. System must offer options to select specific types of cases captured. A table must display, and successfully present case information based on offense descriptions. System must allow user to re-enter and adjust the filter for further display.	
Flow of activities:	Actor	System
	1.Users initiate to retrieve cases captured by radar or camera by click 'Cases Captured'. 2.Users select time frame for case retrieval. 3.Users select specific types of cameras or radars to see the results. 4.Users re-select the filter.	1.System presenting options for case retrieval and filtering. 2.System acknowledges the selected time frame for further processing. 3.System displays options to select specific types of cameras or radars. 3.1 System acknowledges the selected filtering options. 3.2 System processes the selected options and offense description. 3.3 System retrieves cases captured by camera or radar. 3.4 System gathers relevant case data for display. 3.5 System displays a table of information of cases captured by camera or radar and present to user. 4.System generates new table of information selected by user.
Exception conditions:	3. Invalid time period input. 5. System unable to generate current information due to technical problems	

Use Case Table #4

Use case name:	Analyze mobile phone usage cases.	
Scenario:	System analyzes cases related to mobile phone usage based on user selections.	
Brief description:	Users can examine cases resulting from mobile phone usage, the line graph will display the number of cases for each selected offense code.	
Actors:	User (Officer)	
Preconditions:	User has logged into the system with valid credentials. Data related to cases caused by mobile phone usage is available in the system for analysis purposes. A functional mechanism to gather and organize data for analysis must be ready to use.	
Postconditions:	Users have successfully gained insights into trends related to cases caused by mobile usage. System must allow the user to re-select offense codes for further analysis. A line graph must display, representing the number of cases for each selected offense code over the specified period of time.	
Flow of activities:	Actor	System
	1.Users initiate to analyze cases resulting from mobile phone usage by click 'Cases Analyse' 2.Users input offense description then click 'submit' 3.Users select the offense code by drop down and select period of time. 4.Users re-select the different offense codes for analysis.	1.System presenting options for analysis. 2.System acknowledges the selected offense code for further processing. 3.System processes the selected time frame and prepares to generate a line graph. 3.1 System retrieves data related to cases, offense codes, and time periods for analysis. 3.2 System organizes data to generate line graph. 3.3 System displays the line graph, plotting the number of the top 10 of the selected offense case over the specified period of time and present to user. 4.System generates new line graph referring to information selected by user.
Exception conditions:	3.System unable to generate current information due to technical problems	

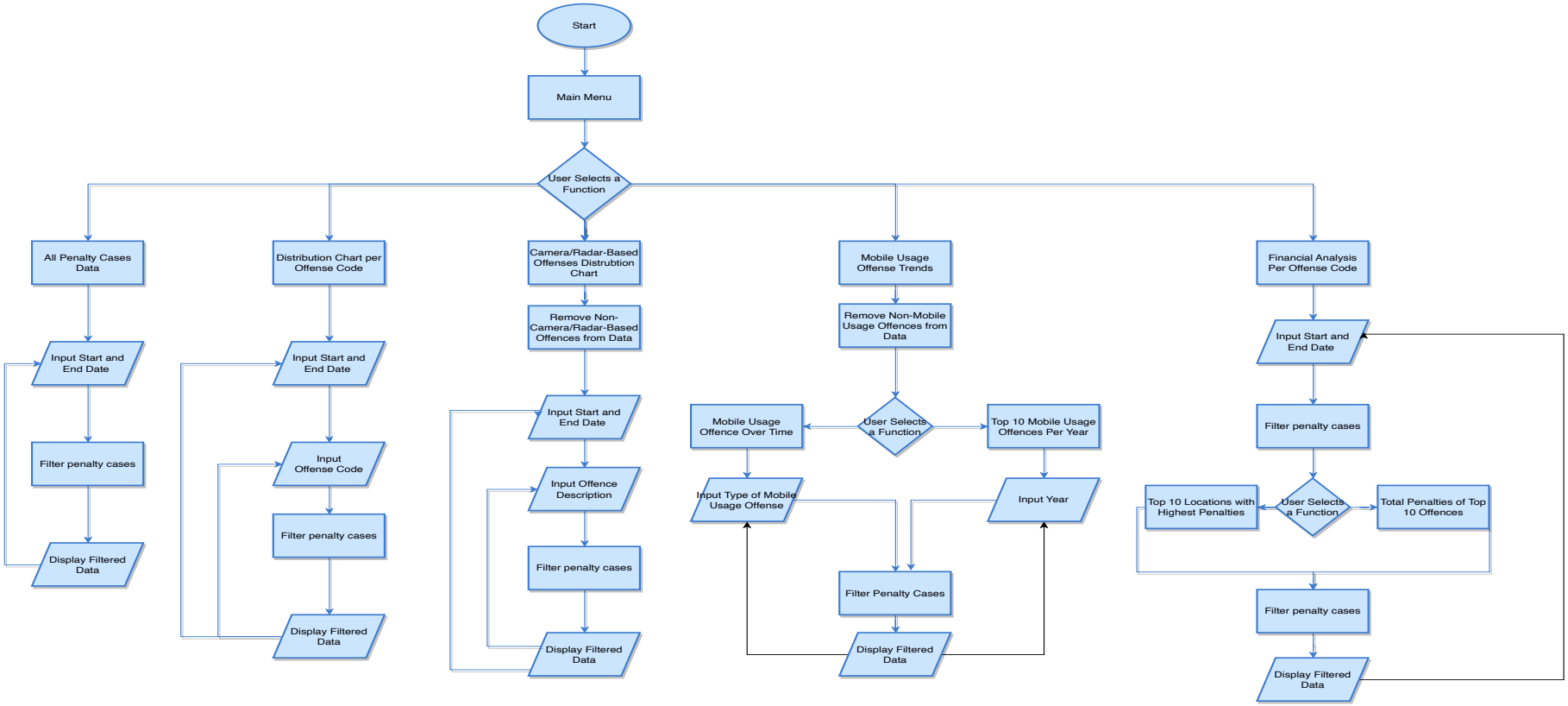
Use case name:	Generate financial analysis.	
Scenario:	System generates a financial analysis of total fees by offense code and selected period based on user selections.	
Brief description:	User can inquire about the financial analysis of total fees collected per offense code; the bar graph will display information.	
Actors:	User (Officer)	
Preconditions:	User has logged into the system with valid credentials. Data including the offense codes, total fees collected, and other related information is available in the system for analysis. System must be ready to perform financial analysis.	
Postconditions:	Users have successfully initiated the financial analysis tools. System must display the corresponding offense code's description and additional information when the user hovers the mouse pointer and clicks on an offense code over the table.	
Flow of activities:	Actor	System
	1.Users initiate to perform financial analysis on total fees collected by click 'Financial Analysis' 2.Users select time period to analyze and click 'submit'	1.System presenting options for financial analysis. 2.System acknowledges the selected time period for further processing. 2.1 System processes the selected time frame and prepares to generate a bar graph. 2.2 System retrieves data related to offense codes, and time periods for analysis. 2.3 System organizes data to generate a table. 2.4 System displays the table, presenting corresponding total fees over the specified period of time and present to user.
Exception conditions:	2.System unable to generate current information due to technical problems. 3.Interaction issues with graph.	

Use Case Diagram


NSW Traffic Penalty Data
Choose reporting period
Select period
Generate charts
Filter function
reselect
Select camera cases
Show result
User
reselect
Analyze mobile phone usage cases
Retrieve data
Generate financial analysis

3.0 Software Design and System Components

3.1. Software Design



3.2. System Components

3.2.1. Functions

This system will take the functional programming paradigm suitable for building a pipeline of data transformation and visualisation tasks. The main focus of this paradigm is to write pure functions that take inputs and return outputs without altering any external state. Accordingly, 12 functions are designed to be pure and stateless. They can be part of a pipeline where the output of one function serves as the input for another function.

1. **getRawData( )**

Description:	This function is for retrieving raw data from CSV file and storing them in different variables mapped to the column's name.
Input:	a) Source location b) Optional arguments to specify particular columns, time ranges, etc
Output:	A data structure containing the raw data.
Side Effects:	None. Purely functional.
Example	raw_data = getRawData('data_source.csv')

2. **getAllPenaltyCases( )**

Description:	This function serves the functional requirement # 1. It extracts data related to all penalty cases. The input data comprises all raw data obtained through the output of the getRawData function, and only specific data will be filtered through this function. Additionally, the data can be selected based on a designated time period, which will be determined as a flag input from outside the function.
Input:	a) The raw data structure from the output of getRawData() b) Optional parameters to specify what constitutes a penalty cases c) Range of period to filter by
Output:	A new data structure containing only the rows that correspond to penalty cases.
Side Effects:	None. Purely functional.
Example	penalty_cases = getAllPenaltyCases(raw_data, penalty_column='Penalty_Amount', isSelectedPeriod = true or false)

3. **getCasesForOffenceCode( )**

Description:	This function serves the functional requirement #2. It is designed to extract data pertaining to all cases under each offence code from the input data. The time, offence code, and description will be carefully chosen and saved into a new list. Furthermore, the data can be filtered based on a specified time period, which will be determined by a flag input from an external source.

Input: a) raw data from getRawData()
b) Offence code to filter by
c) Range of period to filter by

Output: A new data structure containing only cases with the specified offence in the period selected by user.

Side Effects: None. Purely functional.

Example offence_cases = getCasesForOffenceCode(raw_data, '81223', range(start, end))

4. **getCasesByCameraRadar( )**

Description: This function serve the functional requirement #3. This program retrieves data from radar or camera-based cases based on their descriptions. It scans through the list of data, checks for any instances of radar or camera, and stores that data in a separate variable. The final output is then returned to the external function.

Input: a) raw data from getRawData()
b) Camera or radar identifier or criteria
c) Range of period to filter by

Output: New data structure containing only cases caught by the specified camera or radar offence in the period selected by user.

Side Effects: None. Purely functional.

Example camera_radar_cases = getCasesByCameraRadar(raw_data, 'RADAR_001', range(start, end))

5. **getCasesMobilePhone( )**

Description: This function serves the functional requirement #4. This function is for filtering the raw data to get all the cases involving mobile phone usage.

Input: a) raw data from getRawData()
b) Range of period to filter by

Output: New DataFrame or list containing only cases involving mobile phones

Side Effects: None. Purely functional.

Example: mobile_phone_cases = getCasesMobilePhone(raw_data, range(start, end))

6. **getFinancialAnalysis( )**

Description: This function serves the functional requirement #5. This function is for performing financial analysis on the data, like summing up penalties, calculating averages, etc.

Input: a) raw data from getRawData()
b) Range of period to filter by

Output: DataFrame containing financial metrics

Side Effects: None. Purely functional.

Example: financial_metrics = getFinancialAnalysis(raw_data, range(start, end))

7. **filterPeriodTime**

Description: This function is for filtering the raw data to only include cases from a specified time period. This function will be used within the major analytic function 1 to 5.

Input: a) raw data filtered in the main feature function 1 to 5.
b) Start time

c) End time

Output: New data structure containing only cases within the specified time period.

Side Effects: None. Purely functional.

Example: filtered_cases = filterPeriodTime(raw_data, '2021-01-01', '2021-12-31')

8. filterKeyword

Description: This function is for filtering the raw data to only include cases that contain a specified keyword in any of the text fields.

Inputs: a) raw data
b) keyword

Outputs: New data structure containing only cases that match the keyword

Side Effects: None. Purely functional.

Example: keyword_cases = filterKeyword(raw_data, 'camera')

9. showLineChart

Description: This function is for generating a line chart based on the provided data.

Inputs: Data to be visualized (DataFrame, list, etc.)

Outputs: A graphical representation (could be returned as a plot object or displayed directly)

Side Effects: Visual output

Example: showLineChart(financial_metrics)

10. showBarChart

Description: This function is for generating a bar chart based on the provided data.

Inputs: Data to be visualized (DataFrame, list, etc.)

Outputs: A graphical representation (could be returned as a plot object or displayed directly)

Side Effects: Visual output

Example: showBarChart(financial_metrics)

11. showPieChart

Description: This function is for generating a Pie chart based on the provided data.

Inputs: Data to be visualized (DataFrame, list, etc.)

Outputs: A graphical representation (could be returned as a plot object or displayed directly)

Side Effects: Visual output

Example: showPieChart(financial_metrics)

12. **showTable**

Description: This function is for generating a table based on the provided data.

Inputs: Data to be visualized (DataFrame, list, etc.)

Outputs: A table representation

Side Effects: Visual output

Example: showTable(raw_data)

3.2.2. Data Structures / Data SourcesData Structures / Data Sources

A total of seven potential data structures and external data sources are involved in this application.

1. **Raw Data Source (External)**

Type of structure: External file (.csv file)

Description: It serves as the primary source of all data for the application.

List of data members: None

List of functions that use it: getRawData

2. **Dataframe for Raw Data**

Type of structure: Dataframe (from Pandas Library)

Description: It holds the raw data derived from the external source for futher processing and analysis

List of data members: Columns in the raw data includes OFFENCE_FINYEAR, OFFENCE_MONTH, OFFENCE_CODE, OFFENCE_DESC, LEGISLATION, SELECTION_CLAUSE, FACE_VALUE, CAMERA_IN, CAMERA_TYPE, LOCATION_CODE, LOCATION_DETAILS, SCHOOL_ZONE_IND, SPPED_BAND, SPEED_IND, POINT_TO_POINT_IND, RED_LIGHT_CAMERA_IND, SPEED_CAMERA_IND, SEATBELT_IND, MOBILE_PHONE_IND, PARKING_IND, CINS_IND, FOOD_IND,

BICYCLE_TOY_ETC_IND, TOTAL_UNMBER, TOTAL_VALUE.

List of functions that use it: getAllPenaltyCases, getCasesForOffenceCode, getCasesByCameraRadar, getCasesMobilePhone, getFinancialAnalysis, filterPeriodTime, filterKeyword

3. **Filtered Dataframes**

Type of structure: Dataframe (from Pandas Library)

Description: These are the modified or filtered versions of the raw DataFrame, created by applying specific filters or analyses.

List of data members: Similar to the Raw Data DataFrame but only includes the filtered rows.

List of functions that use it: showLineChart, showBarChart, showPieChart, showTable

4. **Financial Metrics**

Type of structure: Dictionary (Python)

Description: It holds the financial metrics calculated from the raw or filtered DataFrames.

List of data members: It could include metrics like “Total_Penalties”, “Average_Penalty”, “Highest_Penalty”, etc

List of functions that use it: showLineChart, showBarChart, showPieChart, showTable

5. **Time Period**

Type of structure: Tuple or Dictionary (Python)

Description: It holds the start and end times for filtering data.

List of data members: “Start_date”, “End_date”

List of functions that use it: filterPeriodTime

6. **Keywords List**

Type of structure: List (Python)

Description: It holds the keywords for filtering data based on textual content.

List of data members: Keywords like speeding, mobile phone, camera radar, etc.

List of functions that use it: filterKeyword

7. Plot Objects (optional)

Type of structure: Objects (e.g., Matplotlib or Plotly objects)

Description: These objects hold the plot configurations and data for visualizations.

List of data members: None

List of functions that use it: showLineChart, showBarChart, showPieChart, showTable

3.3. Detailed Design

The application codes are divided into Functions, Class Definition for Frame, and Main file. In the functions file, you will find all the functions that have been defined above. These functions will be imported by external files as needed. Secondly, the class definition file is specifically designed for the Frame class of the wzPython library. This is separated from the main file to enhance the readability and usability of the codes. Lastly, the main function contain all codes related to the execution of the application. (Note: This code is just pseudocode using the context of Python programming languages.)

1. Functions File

File name: function.py

```
import pandas as pd
import matplotlib.pyplot as plt

def getRawData(source_location, optional_args=None):
    # Read data from CSV into DataFrame
    raw_data = pd.read_csv(source_location, **optional_args)
    return raw_data

def getAllPenaltyCases(raw_data, penalty_column='Penalty_Amount', isSelectedPeriod=False,
time_range=None):
    # Perform operation to get all penalty cases
    if isSelectedPeriod:
        raw_data = filterPeriodTime(raw_data, time_range)
    penalty_cases = raw_data[raw_data[penalty_column] > 0]
    return penalty_cases

def getCasesForOffenceCode(raw_data, offence_code, time_range=None):
    # Filter the DataFrame based on the given offence_code
    raw_data = filterPeriodTime(raw_data, time_range)
    offence_cases = raw_data[raw_data['OFFENCE_CODE'] == offence_code]
    return offence_cases

def getCasesByCameraRadar(raw_data, camera_radar_id, time_range=None):
    # Filter the DataFrame to get cases caught by camera radar
    raw_data = filterPeriodTime(raw_data, time_range)
    camera_radar_cases = raw_data[raw_data['CAMERA_TYPE'] == camera_radar_id]
    return camera_radar_cases

def getCasesMobilePhone(raw_data, time_range=None):
    # Filter the DataFrame to get cases involving mobile phones
    raw_data = filterPeriodTime(raw_data, time_range)
    mobile_phone_cases = raw_data[raw_data['MOBILE_PHONE_IND'] == True]
    return mobile_phone_cases

def getFinancialAnalysis(raw_data, time_range=None):
    # Perform financial analysis on the raw_data
```

```
    raw_data = filterPeriodTime(raw_data, time_range)
    total_penalty = raw_data['Penalty_Amount'].sum()
    average_penalty = raw_data['Penalty_Amount'].mean()
    highest_penalty = raw_data['Penalty_Amount'].max()
    financial_metrics = {'Total_Penalties': total_penalty,
    'Average_Penalty': average_penalty,'Highest_Penalty':
    highest_penalty}
    return financial_metrics

def filterPeriodTime(raw_data, time_range):
    start_date, end_date = time_range
    filtered_data = raw_data[(raw_data['OFFENCE_FINYEAR'] >= start_date)
    & (raw_data['OFFENCE_FINYEAR'] <= end_date)]
    return filtered_data

def filterKeyword(raw_data, keyword):
    # Assuming 'OFFENCE_DESC' is a column that can contain the keyword
    keyword_cases = raw_data[raw_data['OFFENCE_DESC']
    .str.contains(keyword, case=False)]
    return keyword_cases

def showLineChart(data):
    # Code to plot line chart using matplotlib or other plotting libraries
    plt.plot(data)
    plt.show()

def showBarChart(data):
    # Code to plot bar chart
    plt.bar(data.keys(), data.values())
    plt.show()

def showPieChart(data):
    # Code to plot pie chart
    plt.pie(data.values(), labels=data.keys())
    plt.show()

def showTable(data):
    # Code to display table (can use Pandas DataFrame display feature or custom table in GUI)
    print(data)
```

2. Class definition file

```
File name: my_frame.py
import wx
import wx.grid
from functions import getRawData, getAllPenaltyCases, getFinancialAnalysis,
getCasesForOffenceCode, getCasesByCameraRadar, getCasesMobilePhone

class MyFrame(wx.Frame):
    def __init__(self, parent, title):
        super(MyFrame, self).__init__(parent, title=title, size=(500,
        600))

        # Initialize Panel
        self.panel = wx.Panel(self)
```

```
        # Initialize Buttons
        self.load_button = wx.Button(self.panel, label="Load Data",
        pos=(20, 20))
        # Continue to write the codes about buttons

        # Bind Events
        self.load_button.Bind(wx.EVT_BUTTON, self.loadData)
        # Continue to write the codes about events

        # Data Grid
        self.grid = wx.grid.Grid(self.panel, pos=(150, 20), size=(300,
        400))

    # Event Handlers (same as in the original code)
    # ...
```

3.3.1. Data Structures / Data Sources

```
File name: main.py
import wx
import wx.grid
from my_frame import MyFrame

if __name__ == "__main__":
    app = wx.App(False)
    frame = MyFrame(None, "New South Wales Traffic and Penalty Analysis Tool")
    frame.Show(True)
    app.MainLoop()
```

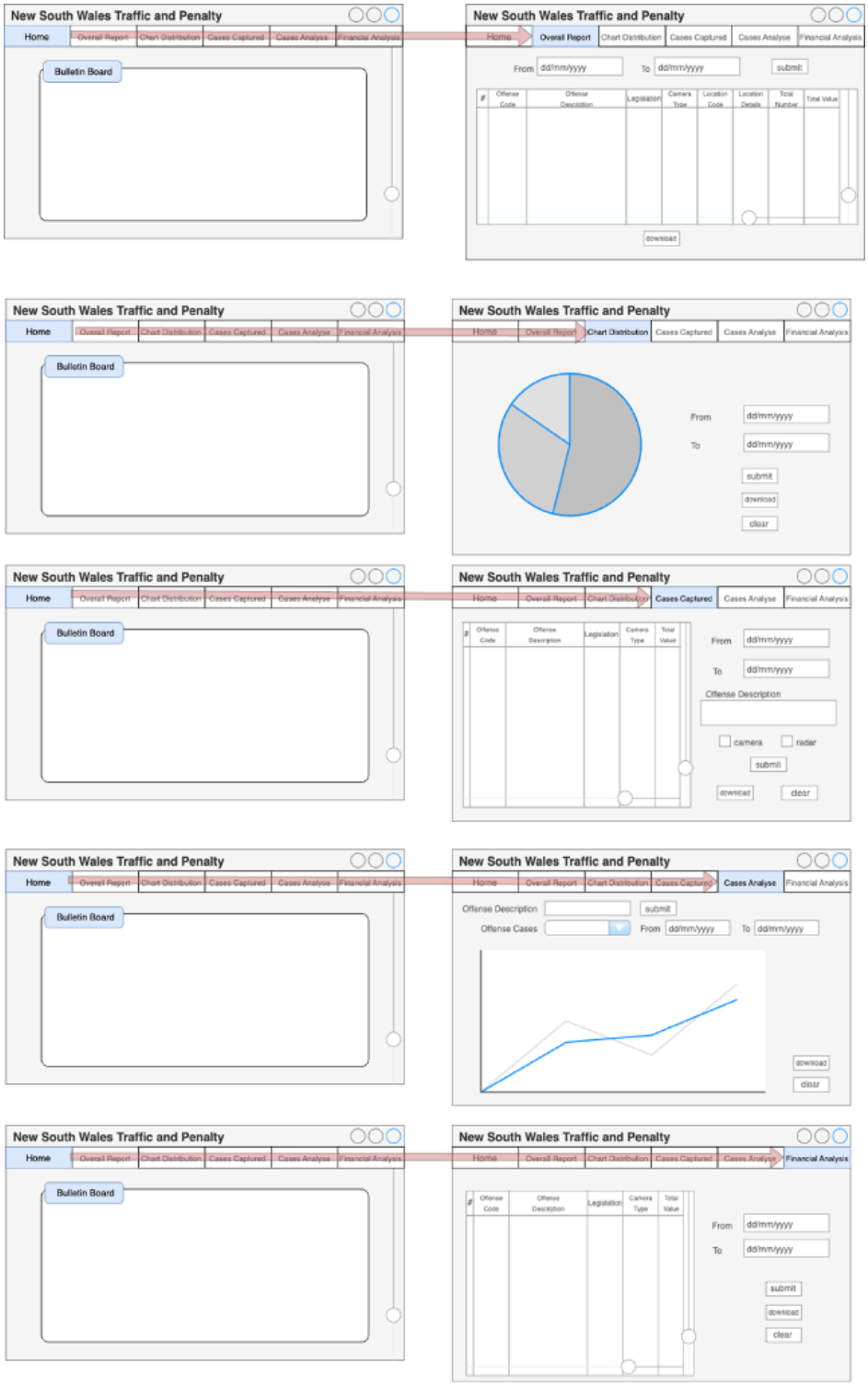
4.0 User Interface Design

The New South Wales Traffic Penalty Data System is a dynamic platform designed to provide users with an enhanced experience in exploring and analyzing traffic penalty cases. The design aims to streamline the journey of users as they engage with multiple features and data representations offered by the application.

Draw.io has been used to create the visual design of the application. The main reason of using Draw.io is that it offers vary levels of design expertise, also supports a wide range of diagram types, including bar charts, pie charts, and bar graph, which have been chosen to create the mockup of application. Moreover, Draw.io is very useful when working with a group, as it allows multiple users to collaborate on the same design real time.

4.1. Structural Design

The interface of New South Wales Traffic Penalty Data System is designed to provide a user-friendly experience for interacting with case data and visualizations. The data is organized into clear sections to facilitate easy navigation and understanding.

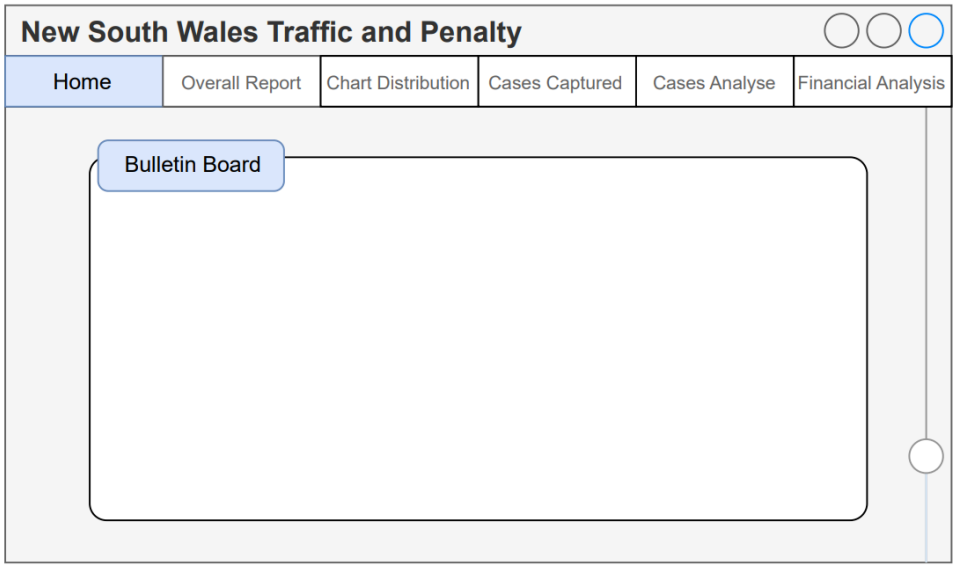


The navigation of the system is organized based on the primary features and activities that users will engage in by the following:

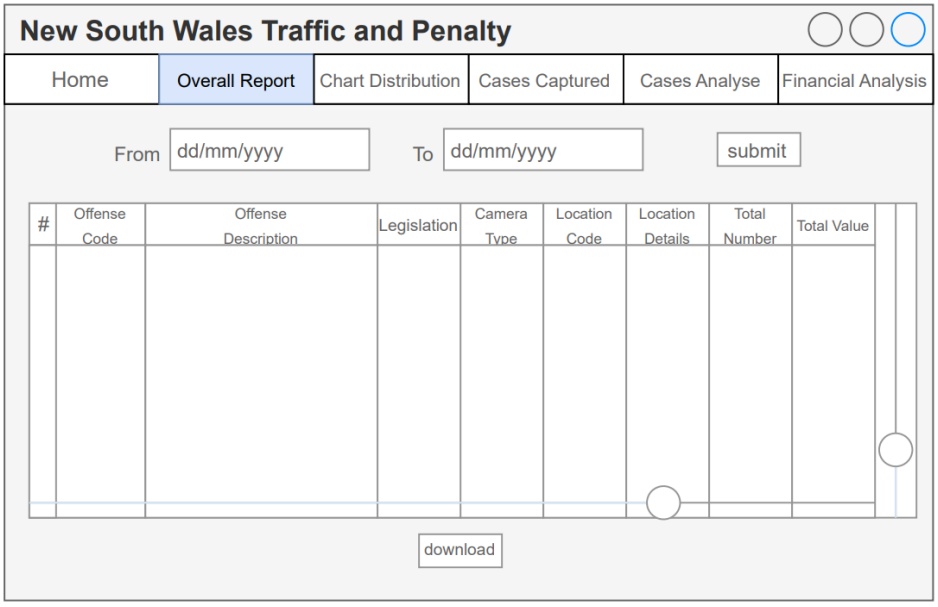
a. Dashboard: The main landing page displays navigation menu and bulletin board.
b. Case reporting: User can access the case reporting, also can input time periods and filtering options.
c. Visualization: The system offers options to generate charts and graphs based on the selected parameters.
d. Offense code details: User can hover over or click on offense codes to view descriptions and additional details respectively.

4.2. Visual Design

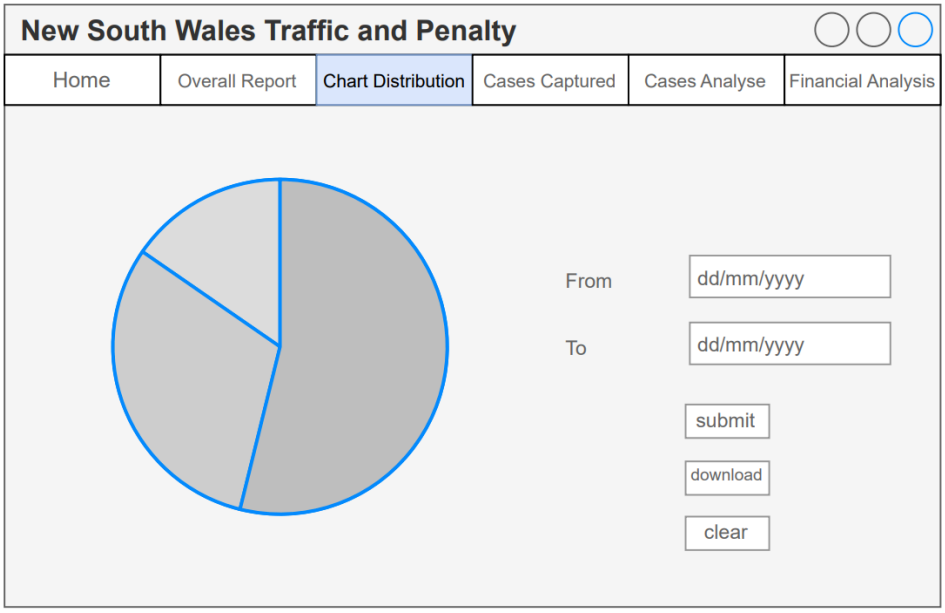
The visual design of the New South Wales Traffic Penalty Data System prioritizes user experience through a clean layout, user-friendly visual elements, a balanced colour palette, and easily readable fonts. This design is enhanced usability, ensuring that users can effectively interact with the system and the displayed information.



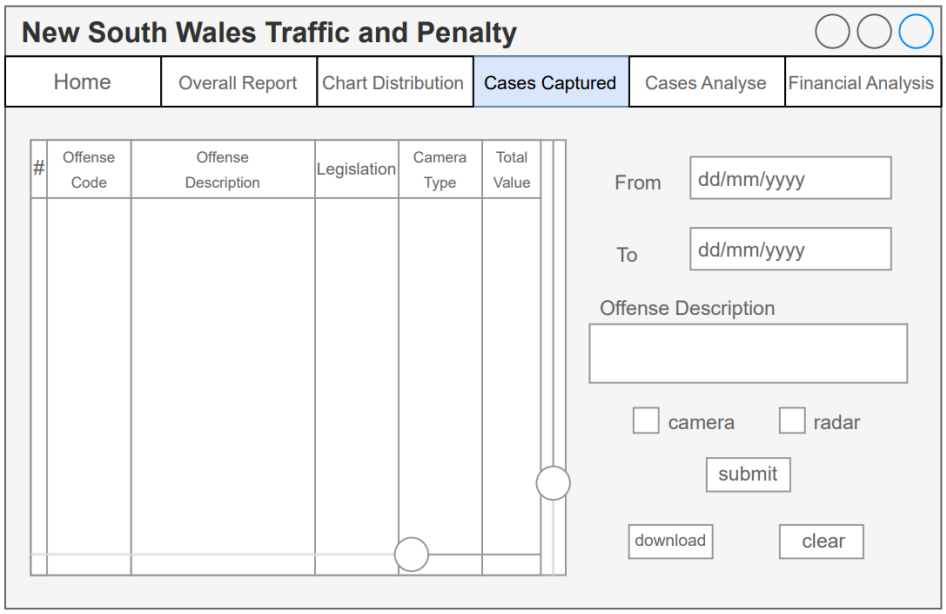
The above is a homepage of New South Wales Traffic Penalty Data System where the users can view updated news within the organization from the Bulletin Board. The navigation menu below the header provides access to sections that are the Overall Report, Chart Distribution, Cases Captured, Cases Analyse, and Finacial Analysis respectively.



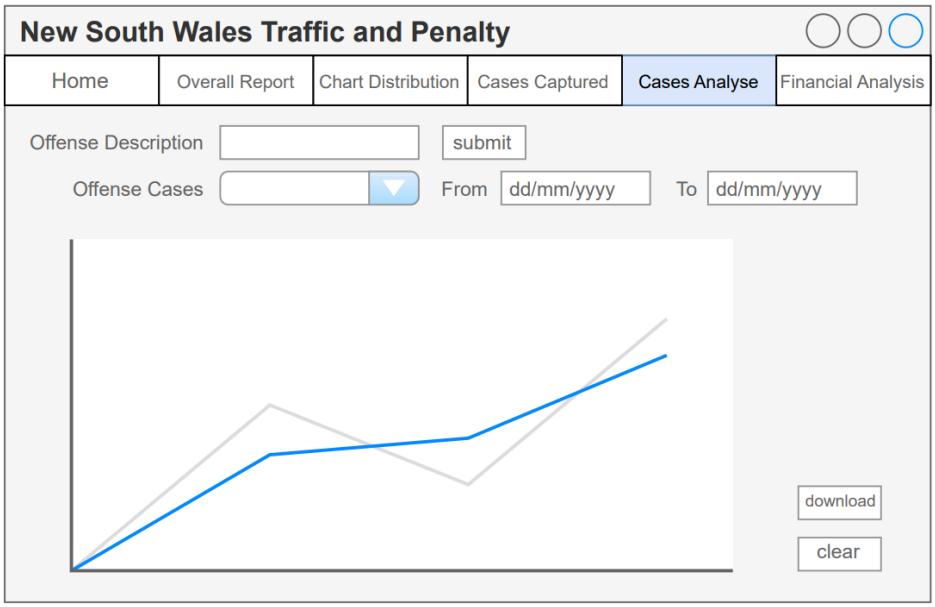
Above is the Overall Report page where users can view offense codes, offense descriptions, and other details. The system requires users to input a time period in the format dd/mm/yyyy and then click the submit button on the right side to retrieve data for that specific time.



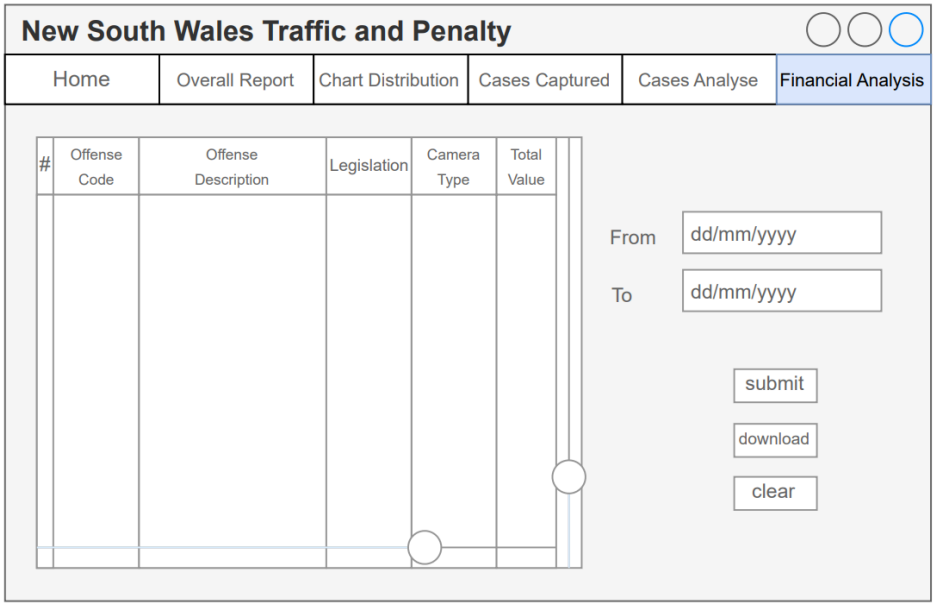
Above is the Chart Distribution page, where users can view a collection of offense cases represented in the form of a pie chart based on the selected time frame. The system requires users to input a time period in the format dd/mm/yyyy and then click the submit button below. Through this page, users can download the data and save it to their device as a PDF file. Also, the system allows users to re-select time periods using the clear button. Moreover, users can hover their mouse pointer over a specific offense code to view the offense description.



Structural DesignThe system requires users to input a time period in the format dd/mm/yyyy, including the offense description. After that, users are required to tick the checkbox and then click the submit button below. Through this page, users can also download the data and save it to their device as a PDF file. Moreover, the system allows users to re-select their pre-selected using the clear button.



Above is the Cases Analyse page, where users can view a line graph of offense cases based on the selected time frame, on this page, users are required to input an offense description and click the submit button on the right side to retrieve a set of offense cases. After that, users will need to select the offense cases from a dropdown and input a time period in the format dd/mm/yyyy, which the graph will automatically display the top 10 of selected offense cases. Also, users can download the data and save it to their device as a PDF file by clicking the download button. The system allows users to re-select from the first step using the clear button.



Above is the Financial Analysis page, where users can view a table of locations with the highest total amount of penalties in descending order based on the selected time frame. On this page, users are required to input put a time period in the format dd/mm/yyyy, and click submit button below. Through this page, users can download the data and save it to their device as a PDF file by clicking the download button. The system also allows users to re-select using the clear button.

As can be seen, the layout of the system is designed to promotes easy navigation and focused content consumption, also aligning with user expectations. With the visual elements, the system offers clear division between sections and easily recognizable icons contribute to users' understanding. Moreover, a minimalist style ensures that users can focus on the data without distractions where the chosen colour is sustains readability, especially for a long period of use. The body font is clean and legible to ensures comfortable reading by using Helvetica font. Overall, by a visually appealing design and user-friendly interface, the Penalty Case Reporting and Analysis System enables users to explore and understand traffic penalty cases efficiently.