Software Design Document

<Project Name>

Student Names

Table of Contents

[1.0 System Vision 3](#_Toc46748622)

[1.1 Problem Background 3](#_Toc46748623)

[1.2 System Overview 3](#_Toc46748624)

[1.3 Potential Benefits 3](#_Toc46748625)

[2.0 Requirements 4](#_Toc46748626)

[2.1 User Requirements 4](#_Toc46748627)

[2.2 Software Requirements 4](#_Toc46748628)

[2.3 Use Cases 4](#_Toc46748629)

[3.0 System Components and Software Design 5](#_Toc46748630)

[3.1 System Components 5](#_Toc46748631)

[3.2 Software Design 5](#_Toc46748632)

[4.0 User Interface Design 6](#_Toc46748633)

# System Vision

## Problem Background

This project's main aim is to develop user-friendly software for analysing and presenting data related to traffic penalties in New South Wales (NSW) from 2011 to 2017. The software's purpose is to improve users' comprehension of penalty data by producing reports that cover from distinct time gap, constructing graphs to illustrate various offense codes, showcasing involving radar and camera offenses, reviewing cases linked to mobile phone usage (including trends and categories of offenses), and incorporating an additional analytical function.

## System Overview

In order to effectively organise, analyse, and visualise NSW Traffic Penalty Data from 2011 to 2017, the suggested programme tries to address the issue. The programme promises to improve decision-making procedures, increase road safety, and optimise resource allocation for law enforcement organisations by providing detailed insights into penalty cases. Users will be able to engage with historical penalty case data using the application's user-friendly interface, investigate trends in the distribution of offences, spot high-frequency offences, and examine mobile phone usage habits. By utilising cutting-edge data analysis techniques, the programme will provide users with useful information that they can use to take appropriate action. This will encourage accountability, transparency, and well-informed decision-making in traffic law enforcement.

## Potential Benefits

The project could offer several potential benefits:

* The application enhances efficiency by providing easy access to historical penalty case information, at the same time, eliminating manual searches.
* Visualised offense distribution charts demonstrate penalty cases’ distribution, which helps facilitate the process of determining and managing high-frequency offenses.
* The application helps pinpoint areas and time of regular offenses by utilising radar or camera data.
* The application contributes to road safety by spotting patterns and trends in traffic violations, which results in more effective awareness campaigns and educational initiatives.
* The application assists with targeting offence codes and locations with higher rate of penalty cases. This supports law enforcement organisations make the best use of their resources.
* The application emphasises transparency, accountability, and accessibility of penalty case data for traffic offences and law enforcement.

# Requirements

## User Requirements

**R1. View Penalty Case Information:**

1. Users should be able to pick a specific time range (start and end dates) to see penalty case details.
2. The website must display a list of all penalty cases during the chosen time, showing case number, offense code, description, fine, offense date and time, location, and capture method (radar/camera).

**R2. Offense Code Distribution Chart:**

1. Users need to choose a time-period and generate a chart that shows how cases are distributed across different offense codes.
2. The chart can be a bar graph or a pie chart, clearly showing the percentage of cases for each offense code.

**R3. Radar/Camera Captured Cases:**

1. Users should be able to select a timeframe and retrieve a list of cases caught by radar or camera, based on the offense description.
2. Users should see the same details as in the penalty case info, with a note indicating if radar or camera caught the case.
3. Users should be able to access videos of cases caught by radar or camera.

**R4. Mobile Phone Analysis:**

1. Users can analyse cases related to mobile phone usage over time.
2. Trends in mobile phone usage cases will be visually presented over selected periods.
3. Details related to mobile phone usage offenses will be available, including offense codes and descriptions.

**R5. Flexibility in User Interaction:**

1. The website should ensure that users can easily modify their choices and settings at any time. This flexibility is crucial for enhancing analysis and exploring different aspects of the information.

## Software Requirements

**R1. Data Import and Integration:**

1. The application shall allow users to import NSW Traffic Penalty Data from 2011 to 2017 in various formats (CSV, Excel, etc.).
2. The imported data shall be integrated into a structured database for efficient storage and retrieval.

**R2. Visualization and Analysis:**

1. The software shall provide visualizations such as charts and graphs to represent the distribution of penalty cases over time, by offense type, and geographical location.
2. Users shall be able to filter and explore data to identify trends, peak periods, and patterns related to specific offenses.
3. Users shall be able to compare the selected offence code other offense types.

**R3. User-Friendly Interface:**

1. The user interface shall be intuitive, user-friendly, and responsive to ensure a seamless experience across different devices.
2. Users shall be able to navigate through the application, access relevant information, and interact with visualizations effortlessly.

**R4. Security and Data Privacy:**

1. The software shall implement appropriate security measures to protect user data and ensure data privacy.
2. User authentication and authorization mechanisms shall be employed to control access to sensitive information.

**R5. Error Handling and Data Validation:**

1. The software should have mechanisms to handle errors, such as incorrect date inputs or missing data.

**R6. Reporting and Exporting:**

1. Users shall have the ability to generate and export reports summarizing penalty case statistics and analysis results.
2. The software shall support exporting visualizations and data in commonly used formats (PDF, CSV, etc.).

**R7. Integration with Version Control:**

1. The application shall integrate with version control systems, such as Git and GitHub, to track changes, manage collaboration, and ensure code integrity.

**R8. Scalability and Performance:**

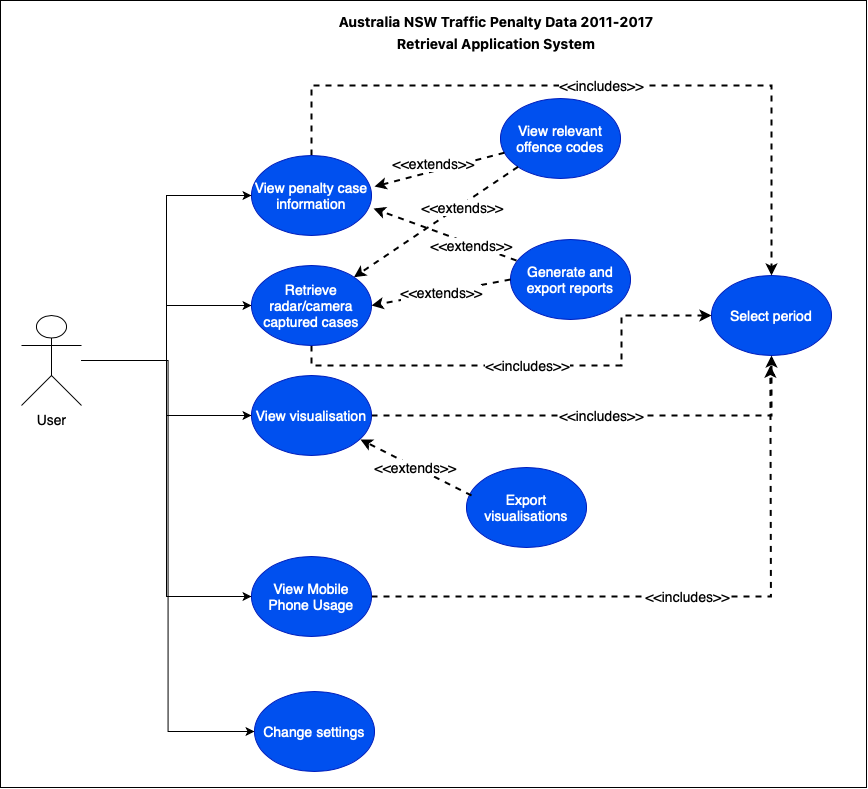
1. The software shall be designed to handle a significant amount of penalty case data efficiently.
2. The software shall be designed to handle a significant amount of penalty case data efficiently.

**R9. User Training and Support:**

The software shall provide user guides, tooltips, and contextual help to assist users in navigating and utilizing the application effectively.

These software requirements serve as the foundation for developing a robust and user-centric application that meets the project's goals and addresses the needs of its intended users.

## Use Cases & Use Case Diagrams



**Figure 1:** Application Use Case Diagram

|  |  |
| --- | --- |
| **Use Case ID** | 1 |
| **Use Case Name** | View penalty case information |
| **Actor** | User |
| **Description** | A user selects a period of time to view a chart that shows how cases are distributed across different offense codes |
| **Flow of Event** | 1. The user will launch the application 2. The user will select a period of time 3. The user selects a penalty case among the penalty cases listed by the system 4. The system displays the penalty case information |

|  |  |
| --- | --- |
| **Use Case ID** | 2 |
| **Use Case Name** | Retrieve radar/camera captured cases |
| **Actor** | User |
| **Description** | A user selects a period of time and choose a penalty case which was captured by radar/camera to see the recording |
| **Flow of Event** | 1. The user will launch the application 2. The user will select a period of time 3. The user selects a penalty case which was captured by radar/camera 4. The system displays the recording |

|  |  |
| --- | --- |
| **Use Case ID** | 3 |
| **Use Case Name** | View visualisations |
| **Actor** | User |
| **Description** | A user selects a period of time and choose a penalty case to view its information |
| **Flow of Event** | 1. The user will launch the application 2. The user will select a period of time 3. The user will select to view offence code distribution chart 4. The system will generate a chart that shows how cases are distributed across different offense codes for the selected period |

|  |  |
| --- | --- |
| **Use Case ID** | 4 |
| **Use Case Name** | View relevant offence codes |
| **Actor** | User |
| **Description** | A user selects a relevant offence code to after view the selected offence case |
| **Flow of Event** | 1. The user will select a penalty case 2. The system will display the penalty case, and suggest relevant offence codes 3. The user will choose to view a relevant offence code 4. The system will show a list of cases of the chosen offence code within the selected period |

|  |  |
| --- | --- |
| **Use Case ID** | 5 |
| **Use Case Name** | Generate and export reports |
| **Actor** | User |
| **Description** | A user generates and exports a report for the chosen offence case |
| **Flow of Event** | 1. The user will select and view the offence case 2. The user will choose to generate a report for the chosen case 3. The system will export the report for the case |

|  |  |
| --- | --- |
| **Use Case ID** | 6 |
| **Use Case Name** | Change settings |
| **Actor** | User |
| **Description** | A user changes settings for the application |
| **Flow of Event** | 1. The user will select settings tab 2. The user will change the settings to their preferences 3. The system will update the settings according to the user changes |

# Software Design and System Components

## Software Design

A diagram of a process

Description automatically generated

**Figure 2**: The flowchart of how the application might work

## System Components

### Functions

1. load\_data:

Description: Loads the NSW traffic penalty data from a given source (e.g., CSV file).

Input Parameters: File path (string).

Side Effects: Reads data from the file and stores it in memory (data structures).

Return Value: Loaded data in a suitable data structure (e.g., DataFrame).

1. filter\_cases\_by\_period:

Description: Filters penalty cases based on the user-selected period.

Input Parameters: Start date, end date (both in datetime format), data (loaded penalty data).

Side Effects: None.

Return Value: List of penalty cases within the selected period.

1. view\_penalty\_case\_information:

Description: Displays a list of penalty case information based on user-selected period.

Input Parameters: Start date, end date (both in datetime format), data (loaded penalty data).

Side Effects: None.

Return Value: None.

1. generate\_distribution\_chart:

Description: Generates a chart showing the distribution of cases for each offense code within a selected period.

Input Parameters: Start date, end date (both in datetime format), data (loaded penalty data).

Side Effects: None.

Return Value: None.

1. retrieve\_radar\_camera\_cases:

Description: Retrieves cases captured by radar or camera based on offense description within a user-selected period.

Input Parameters: Start date, end date (both in datetime format), data (loaded penalty data).

Side Effects: None.

Return Value: List of cases captured by radar or camera.

1. analyze\_mobile\_phone\_usage:

Description: Analyzes cases related to mobile phone usage over a user-selected period.

Input Parameters: Start date, end date (both in datetime format), data (loaded penalty data).

Side Effects: None.

Return Value: Analysis results, trends, offense codes, and descriptions related to mobile phone usage.

1. analyze\_seatbelt\_usage:

Description: Provides insights related to cases where passengers are not wearing seatbelts properly.

Input Parameters: Start date, end date (both in datetime format), data (loaded penalty data).

Side Effects: None.

Return Value: Relevant offense codes, descriptions, and insights.

1. update\_settings:

Description: Allows the user to change their choices and settings for analysis.

Input Parameters: Current settings, user's new choices.

Side Effects: Updates the analysis settings.

Return Value: Updated settings.

1. error\_handling\_and\_validation:

Description: Handles errors and validates user inputs, providing appropriate feedback.

Input Parameters: User inputs.

Side Effects: Displays error messages or prompts for valid input.

Return Value: Boolean indicating input validity.

### Data Structures / Data Sources

**Data Structure: PenaltyCase Class**

* Type: Class
* Description: Represents a single penalty case with associated attributes.
* Data Members:
  + OFFENCE\_FINYEAR (str): The financial year of the penalty notice.
  + OFFENCE\_MONTH (Date): The first day of the month of the penalty notice.
  + OFFENCE\_CODE (int): A unique identifier for the offence.
  + OFFENCE\_DESC (str): The description of the offence.
  + ... (Other attributes as per your data)
* Functions That Use It: Most functions in’ **PenaltyDataContainer’** interact with lists of ‘**PenaltyCase’** objects.

**Data Structure: Pandas DataFrame**

* Type: Class
* Description: Manages penalty case data and operations.
* Data Members:
  + penalty\_cases (list): List of **PenaltyCase** objects.
* Functions That Use It: Most functions in this class interact with the **penalty\_cases** list.

**Data Structure: Pandas DataFrame**

* Type: Library Data Structure
* Description: A two-dimensional, size-mutable, and heterogeneous tabular data structure used for data manipulation and analysis.
* Usage: The penalty data can be loaded into Pandas DataFrames for efficient manipulation, filtering, and analysis.
* Functions That Use It: Several functions in **PenaltyDataContainer** perform data operations using Pandas DataFrames.

**Data Structure: Dictionary**

* Type: Built-in Data Structure
* Description: An unordered collection of data in a key-value pair format.
* Usage: Dictionaries can be used to store and retrieve analysis settings, insights, and other metadata.
* Functions That Use It:’ **update\_analysis\_settings** ‘function uses dictionaries to update settings.

**Data Structure: List**

* Type: Built-in Data Structure
* Description: An ordered collection of data.
* Usage: Lists are used to store collections of PenaltyCase objects and other data structures.
* Functions That Use It: Multiple functions use lists to store and retrieve penalty cases and analysis results.

**Data Source: CSV File**

* Type: External Data The Source
* Description: A comma-separated values file containing penalty data.
* Usage: The **load\_penalty\_data** function reads penalty data from CSV files and creates **PenaltyCase** objects.
* Functions That Use It: **load\_penalty\_data** function reads from CSV files.

### Detailed Design

**Pseudocode for loading penalty data from CSV files:**

Function load\_penalty\_data(file\_path: str) -> List[PenaltyCase]:

// Initialize an empty list penalty\_cases

Try:

Open the CSV file at 'file\_path' for reading

For each row in the CSV file:

Create a new PenaltyCase object

Parse the row and populate the attributes of the PenaltyCase object

Append the PenaltyCase object to penalty\_cases list

Catch any file reading errors and handle them gracefully

Close the CSV file

Return penalty\_cases

**Pseudocode for filtering penalty cases by period:**

Function filter\_cases\_by\_period(penalty\_cases: List[PenaltyCase], start\_date: str, end\_date: str) -> List[PenaltyCase]:

// Initialize an empty list filtered\_cases

For each penalty\_case in penalty\_cases:

If penalty\_case.OFFENCE\_MONTH is within the range [start\_date, end\_date]:

Append penalty\_case to filtered\_cases

Return filtered\_cases

**Pseudocode for generating distribution chart:**

Function generate\_distribution\_chart(penalty\_cases: List[PenaltyCase]):

// Initialize a dictionary offence\_counts

For each penalty\_case in penalty\_cases:

If penalty\_case.OFFENCE\_CODE is not in offence\_counts:

Initialize offence\_counts[penalty\_case.OFFENCE\_CODE] to 1

Else:

Increment offence\_counts[penalty\_case.OFFENCE\_CODE] by 1

Plot a chart using offence\_counts data

Display the chart to the user

**Pseudocode for updating analysis setting:**

Function update\_analysis\_settings(settings\_dict: Dictionary, new\_settings: Dictionary):

For each key, value pair in new\_settings:

Update settings\_dict[key] with the new value

Return updated settings\_dict

# User Interface Design

This is your initial interface design. Describe the tools you used for this design stage and any key findings that informed your design. This introduction is descriptive and should explain what you have completed for the actual design work you will present in the sub-sections below.

Tools Used:

Wireframing and Prototyping Software: paint, Excel, Lucid Chart, Figma, or Sketch for wireframes and prototypes.

User Feedback: Continuous feedback and iterative design for improvements.

Responsive Design: Ensuring a consistent experience across devices.

Key Design Considerations:

Dashboard: Central hub for accessing functionalities.

Navigation: User-friendly menu for easy section access.

Data Visualization: Interactive charts and graphs for penalty data.

Search and Filters: Filters based on criteria like date and offense type.

Case Details: Detailed view for penalty case information.

Export and Reporting: Report generation and export.

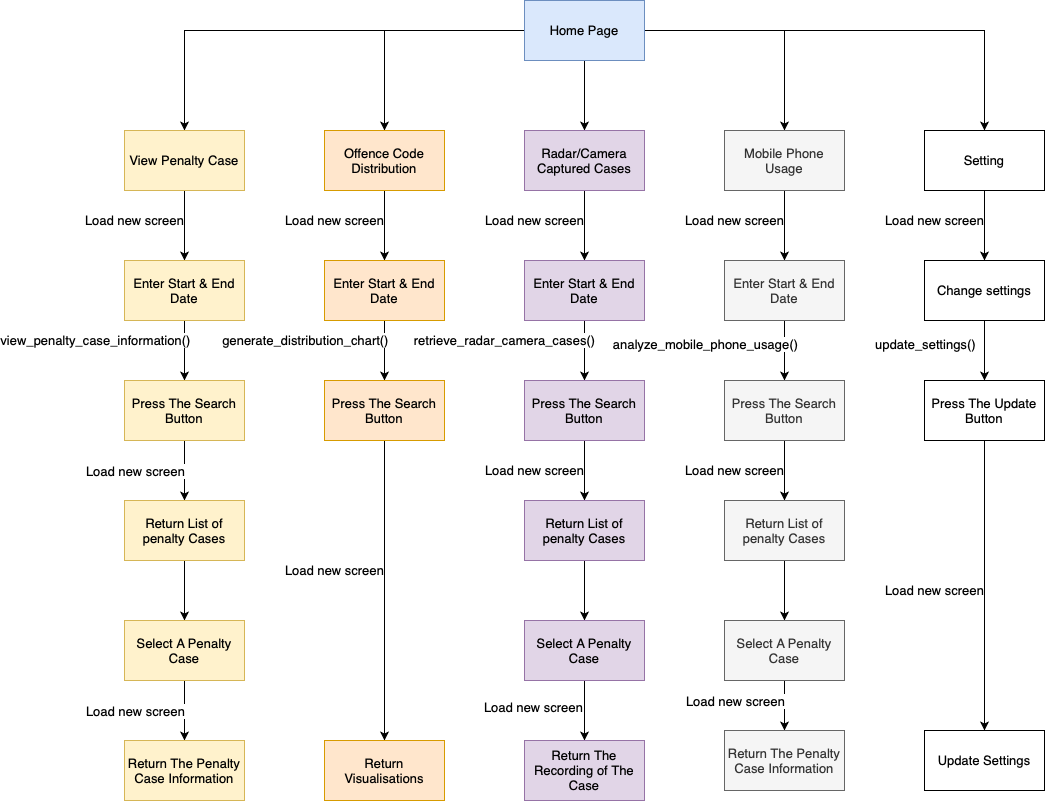
Settings: Customizable analysis preferences.

User Support: Contextual help and tooltips.

Security: User authentication and authorization.

Error Handling: Clear guidance for incorrect inputs.

## Structural Design



**Figure 3**: The application Structure Diagram

The structure design of NSW Traffic Penalty application ensures that users can discover the app easily and effortlessly. The outline design in section 4.2 illustrates that the application enables users to retrieve data in accordance with their preferences. It satisfies the majority of the requirements in section 2.2 and visually represents most of the use cases displayed in section 2.3. The design goals are to define a logical group of information, navigation and provide a thorough hierarchy that satisfies the users’ UXUI desire.

Interface structure:

* Consistency: whether the users use IOS or Windows devices to access the application, its structure and layout are depicted in a consistent manner/ for better user experience, it is designed to be consistent and predictable.
* Headings: to draw attention to the page that users are currently in, each page headings will be modified in bold and always positioned in the top left corner. It is also filled with a colour that contrasts with the background colour, making the page easy to recognise.
* Fonts and colour scheme: Only one font is employed throughout the “NSW Traffic Penalty” application. We aim to provide a minimalist data retrieval application, thus, we just utilise black and white colours.
* Layouts: The layout of each page has some things in similar, such as the “Select Period” bar and the navigation panel on the right-hand side. However, there is a heading in the top corner and in bold that highlights the name of the page, and the contents displayed are different.

To organise and show data in a user-friendly and straightforward way, the web application must group information.

**Home Page:** The information presented on this page will be grouped as follows:

* The Home page will introduce the application and give a high-level overview of the functionalities that are offered.
* Information will be organised into parts or cards, each of which will stand for one of the application's primary functions: View Penalty Cases, Offence Code Distribution Chart, Radar/Camera Captured Cases, Mobile Phone Usage and Settings

**View Penalty Case Details**: The information presented on this page will be grouped as follows:

* The first component, the Date Range Selector, will let users choose a time frame.
* List of Penalty Cases: The list of penalty cases will be presented in tabular format, with columns for the case number, offence code, description, fine, date and time of the offence, and location.

**Offence Code Distribution Chart**: The data will be arranged in the following groups in this section:

* Time-period selector: Users can select the time period for analysis using the time-period selector.
* Display of Chart: A chart outlining how cases are distributed among the various offence codes will be prominently displayed.
* Legend: The offence codes and colours on the chart will be explained in a legend.

**Radar/Camera Captured Cases**: The following information will be categorised:

* Users can select a timeframe using the time-frame selector.
* Case List: Provides thorough details on the cases that have been chosen, including whether they were discovered by radar or camera.
* Video Player: If there are any videos related to the cases, they will be bundled with the relevant case information.

**Mobile Phone Usage:** Information will be grouped as follows:

* Time-Period Selector: Users can specify the time period for mobile phone usage analysis.
* Mobile Phone Offense Details: Information related to mobile phone usage offenses will be grouped together, including offense codes and descriptions.
* Trends Chart: The chart displaying trends in mobile phone usage cases over time will be the focal point.

Navigation strategy:

* The homepage is the first thing users see when they access the application.
* There will be a navigation panel in the right-hand side of each page, thus, the users can navigate among View Penalty Cases, Offence Code Distribution Chart, Radar/Camera Captured Cases, Mobile Phone Usage and Settings pages easily
* Except Home Page and Setting Page, other pages contain a search input, which includes a time-period selector that allows the users to input start and end dates of the period they want to search for penalty cases.
* After inputting the dates, the users can click the search button and view the results displayed.

## Visual Design

Detail your visual design: Layout, visual elements, icons, graphics, style, colour, fonts general screen designs. This can be sketches, wireframes, mockups etc, supported by a discussion, explanation, and justification of your choices.

A screenshot of a traffic penalty

Description automatically generatedA screenshot of a case

Description automatically generatedA graph of a number of blue bars

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generatedA screen shot of a phone usage

Description automatically generated