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

## 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.User should be able to select a specific period (start and end dates) for which they want to view penalty case information.

2. The software should display a list of all penalty cases within the selected period, including details like case number, offense code, offense description, fine amount, date and time of the offense, location, and mode of capture (radar/camera).

R2. Offense Code Distribution Chart:

1. User should be able to choose a period and generate a chart that shows the distribution of cases for each offense code within that period.

2. The chart could be a bar chart or a pie chart, clearly showing the percentage of cases for each offense code.

R3. Radar/Camera Captured Cases:

1. User should have the option to select a period and retrieve a list of all cases that were captured by radar or camera, based on the offense description.

2. The software should display the same details as in the penalty case information, provide with a note indicating whether the case was captured by radar or camera.

R4. Mobile Phone Usage Analysis:

1. User should be able to analyse cases related to mobile phone usage over time.

2. The software should provide trends (increases or decreases) in mobile phone usage cases over a selected period.

3. User should be able to view the relevant offense codes, descriptions, and any other insights related to mobile phone usage offenses.

R5. Additional Insight/Analysis Tool:

1. Users should have the ability to view the relevant offense codes, descriptions, and any other insights related to cases where passengers are not wearing their seatbelts properly.

R6. Flexibility in User Interaction:

1. The program should let user change their choices and settings anytime to make the analysis better and look at different parts of the information.

R7. Error Handling and Data Validation:

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

## 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.

**R3. Mobile Phone Usage Analysis:**

1. The application shall analyse penalty cases related to mobile phone usage and present insights into the trends and frequencies of such cases.
2. Users shall be able to compare mobile phone usage cases against other offense types.

**R4. 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.

**R5. 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.

**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.

In this section you detail what the requirements for the software are. What functionality will it provide? This is usually a formal listing, with requirements often using the word ‘Shall’. IE:

R1.1 The program shall accept multiple file names as arguments from the command line.

R1.2 Each file name can be a simple file name or include the full path of the file with one or more levels.

etc …

Can be primarily functional requirements, though you may include other types if you think of them.

## Use Cases & Use Case Diagrams

In this section you provide some use cases showing how people may use your software.

# Software Design and System Components

## Software Design

A block diagram/flowchart of how your software might work

## System Components

### Functions

Preliminary list of all functions in the software. For each function in the list the following information is provided:

* a brief description of what it does (1 or 2 sentences);
* a list of the input parameters, and their data types, and what they are used for;
* a list of any side effects caused by the function (ie change global or member variables, changes data passed by reference from calling function etc)
* a description of the function’s return value

### Data Structures / Data Sources

List of all data structures in the software (eg linked lists, trees, arrays etc) or eternal data sources. For each data structure in the list the following information is provided:

* Type of structure (tree, list etc),
* Description of where and how it is used
* List of data members, and what each one is for do
* List of functions that use it

### Detailed Design

Pseudocode for all non-standard / non-trivial algorithms that operate on data structures

# 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.

## Structural Design

Structural design refers to the navigational and information structure of your product – the structure that supports the interface layout. How will you structure your product? How will you group your information? How will you navigate through your product? Why? This can take the form of a diagram showing structure and hierarchy, supported by a discussion and justification of your choices. Why have you made these design choices? Describe and outline the structure of your interface and of your information.

## 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.