Software Design Document

*NSW Traffic Penalty Data Search*

Nikkelas Raines, Taiki Matehe, Kiarna Broomhead

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

The dataset that will be analysed is the NSW traffic penalty dataset which include reports from 2011 until 2017. The dataset holds a wide variety of information, including but not limited to, the financial year of the offence, the month of the offence, offence description, offence code, legislation the offence falls under and location.

Users will be able to input selected time periods, penalty codes, keywords, and phrases, if the penalty case was detected by cameras to receive and output of information within the given constraints. This output can be altered to produce information in the form of graphs, with additional information available such as the average value ($AUD) of the penalties.

The program helps solve the issues the user may have with displaying and organising the data to be able to gather useful information that can be used to help change areas of the organisation and see what is working and what needs to focus on more.

This program will be used by government officials such as police departments to keep track of data.

## System Overview

This system will be designed to provide specified data about the NSW traffic penalties to the user. It will be able to produce graphs for greater insight of the information and filter data specified to what is wanted by the user.

The system will include browsing functions which will let the user view all information. This browsing function will include features such as the ability to filter information -this can be by date, case codes, keywords, and phrases, or if the penalty case was detected by cameras-, as well as the ability to generate graphs for the information and calculate the average value ($AUD) of the penalties.

## Potential Benefits

The potential benefits this program proposes is the user has access to data analysis tasks without the need for extensive knowledge of a sorting program such as a database or excel formulas. With the data being displayed and filtered it can be used to in analysis on which area is doing well and which is not as well as the common penalties that are happening which can lead to more understand of what needs to be focused on.

# Requirements

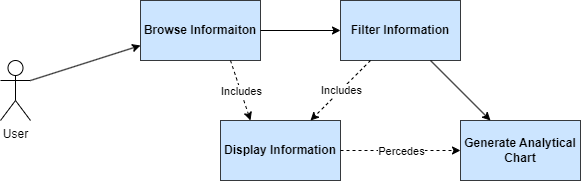
## User Requirements

As a user I can search through the data based on a start and end date.  
As a user I can search through the data based on a keyword.  
As a user I can remove data that is not recorded on camera.  
As a user I can search through the data based on a keyword within a start and end date.  
As a user I can toggle if the data is searched though the last data displayed or though the full dataset.  
As a user I can swap tabs between the data table and the graph tab.  
As a user I can display a bar graph of the case distribution in each offense code within a start and end date .  
As a user I can display a bar graph of the mobile phone penalty data within a start and end date.  
As a user I can display a line graph of the average cost of penalty cases within a start and end date.

## Software Requirements

* 1. The program shall be able to load csv files into the required format.
  2. The program shall accept a start and end date as a search parameter.
  3. The program shall accept the start and end date parameter while using other search parameters.
  4. The program shall accept a keyword as a search parameter.
  5. The program shall remove all data not recorded on camera if selected.
  6. The program shall have a toggle to search through the previously searched data or the full dataset.
  7. The program shall display a bar graph of the case distribution in each offense code.
  8. The program shall display a bar graph of the mobile phone penalty data.
  9. The program shall display a line graph of the average cost of penalty cases.

## Use Cases & Use Case Diagrams



# Software Design and System Components

## Software Design

A diagram of a flowchart

Description automatically generated

## System Components

### Functions

Function Name: Dataframe

Function Description: Converts data from csv to dataframe

Function Inputs: Data from NSW traffic penalty dataset

Function Side Effects: N/A

Function Return Value: Converted data

Function Name: Date Search

Function Description: Used in every other function to select data within a specified period, if left blank this will default to max. Can be searched alone

Function Inputs: Data, Search Dates (Must include start and end date)

Function Side Effects: N/A

Function Return Value: Adjusted Data List

Function Name: Keyword Search

Function Description: Filters data and provides the user with data that contains a specific keyword

Function Inputs: Data, Search Keyword, Search Column

Function Side Effects: N/A

Function Return Value: Adjusted Data List

Function Name: Bar graph of case distribution

Function Description: Produces a bar graph of the different case codes within the user selected period.

Function Inputs: Data from previous functions

Function Side Effects: N/A

Function Return Value: Bar graph showing the case distribution

Function Name: Line graph of average penalty case cost

Function Description: Generates a line graph of the average penalty case cost ($AUD) with data produced from the user selected parameters

Function Inputs: Data from previous functions (Requires at least 1 input from another function)

Function Side Effects: N/A

Function Return Value: Line graph showing the average penalty case cost in $AUD

Function Name: Refresh Toggle

Function Description: If on, refreshes the users search parameters so it includes all data. If off, keeps the users previous search parameters when searching again.

Function Inputs: Toggle on/off

Function Side Effects: N/A

Function Return Value: Adjusted Data List

Function Name: Detected by camera button

Function Description: Within the user selected parameters, displays cases which have been detected by a camera or radar. If no parameters are selected, filters from all existing data

Function Inputs: Button press

Function Side Effects: N/A

Function Return Value: Adjusted Data List

### Data Structures / Data Sources

Structure Type: Linked list-based stack

Structure Description: Each data search will save previous user selected parameters. Each new search with different parameters will provide an output from the previously updated data list.

List of Data Members: Top index, nodes (value, next index)

Functions Which Utilize This Structure: Search functions, Refresh function

Structure Type: Trie

Structure Description: Searches text-based data for keywords. Can be used for partial-match searches.

List of Data Members: keys, child node, reference to parent node, frequency count

Functions Which Utilize This Structure: Keyword search, Detected by camera button

Structure Type: B trees

Structure Description: Used to access and query date-based information. Will be used for indexing date columns.

List of Data Members: keys, references to children, order (determines the trees branching factor),

Functions Which Utilize This Structure: Date search

Structure Type: Array

Structure Description: Stores values which will be represented in graphs.

List of Data Members: Data type, size, indices

Functions Which Utilize This Structure: Bar graph of case distribution, line graph of average penalty case cost

### Detailed Design

**Delete all elements from a stack**

procedure deleteStack(stack):

while not isEmpty(stack):

pop(stack)

procedure pop(stack):

if isEmpty(stack):

return

else:

topNode = stack.top

stack.top = topNode.next

dispose(topNode)

function isEmpty(stack):

return stack.top is null

**Search for keywords in a trie**

procedure searchKeywordsInTrie(root, keywordList):

for each keyword in keywordList:

current = root

found = true

for character in keyword:

if current.children[character] is null:

found = false

break

current = current.children[character]

if found and current.isEndOfWord:

output(keyword)

else:

output("Keyword not found: " + keyword)

procedure output(result):

print(result)

**Search for date in b tree**

procedure searchDatesInBTree(node, startDate, endDate):

if node is null:

return

for each entry in node.entries:

if entry.date is within [startDate, endDate]:

output(entry.data)

if entry.date >= startDate:

searchDatesInBTree(entry.leftChild, startDate, endDate)

if entry.date <= endDate:

searchDatesInBTree(entry.rightChild, startDate, endDate)

procedure output(data):

print(data)

# User Interface Design

For the initial interface design, Microsoft Visio was used for the wireframes. Visio was chosen as it allows for high amounts of editability and customization with plenty of different shapes and features to be added into the wireframes.  
3 pages have been designed which uses all the functional requirements for the program mainly the dataset and graph pages with the other page (user manual) explaining how all the functions and buttons work on these 2 pages.

## Structural Design

A diagram of a flowchart

Description automatically generated

The program will be split into 3 different pages, the Dataset, Graph and User manual. The data was grouped based on how the data will be displayed and to allow the data to not be cluttered with different functions.  
The Dataset page will display a table containing the filtered data. This page will also include a date picker and a search bar for searching for key words. A reset toggle button and a button to display only camera recorded penalties will also be on the page.  
The Graph page has its own page to allow for large graphs to allow for easy readability even when there is a large amount of data.  
The User Manual page is on its own page to allow for easy understanding of the program without confusion of some functional aspects on this page.

## Visual Design

The general layout of the program consists of the name at the top to allow the user to quickly understand what the data is for. The navigation bar is at the top of the screen with the functional tabs on the left side of the screen while the right side consists of the user manual as it has no functional aspects. The current selected tab will be a slightly different colour than the other tabs to indicate which one is currently being used. The bottom of the screen has some consistent functions to allow for easy understanding.

A screenshot of a computer screen

Description automatically generated

The data table is displayed on most of the screen on within the data tab, this is done to allow for the most amount of data to be seen at once with a scroll bar on the right to allow viewing of other data not visible. The bottom left has a button that toggles if the data will be using the current displayed data or the base data. The green dot will change to red if the data is set to not reset when searching.  
The remove nonvideo data button removes all the data values that haven’t got a video recording of the penalty. The Start and end date is used to filter the data within a specific timeframe with the ability to be used alongside the keyword search.

A screenshot of a graph

Description automatically generated

The Graphs page will display 3 graphs, these will have the name of what they are displaying as well as more information in the X and Y Axis columns. A reload button is used to reload the graphs based on the start and end date at the bottom right.

A screenshot of a computer

Description automatically generated

The User manual page will have the buttons of the data page to allow for visual explanations of the functions these use, there will also be text explanations of how the program works.