

Data analysis and visualisation

2810ICT – Software Technologies



Griffith university

trimester 2, 2022

Andrew Kinnaird (s2843939) | Shakya Fernando (s5169761)

Table of Contents

[**1.0** **Introduction** 1](#_Toc113207996)

[**1.1 Background Information** 2](#_Toc113207997)

[**1.2 Scope** 2](#_Toc113207998)

[**1.3 Document Contents** 2](#_Toc113207999)

[**2.0 Work Breakdown Structure** 3](#_Toc113208000)

[**3.0 Activity Definition and Estimation** 4](#_Toc113208001)

[**4.0 Gantt Chart** 7](#_Toc113208002)

[**5.0 Software Design Document** 8](#_Toc113208003)

[**5.1 System Vision Document** 8](#_Toc113208004)

[**5.1.1 Problem background** 8](#_Toc113208005)

[**5.1.2 System Overview** 8](#_Toc113208006)

[**5.1.3 Potential Benefits** 8](#_Toc113208007)

[**5.2 Software Requirements** 9](#_Toc113208008)

[**5.2.1 User Requirements** 9](#_Toc113208009)

[**5.2.2 Software Requirements** 9](#_Toc113208010)

[**5.2.3 Use Cases** 9](#_Toc113208011)

[**5.3 Software Design and System Components** 11](#_Toc113208012)

[**5.3.1 Software Design** 11](#_Toc113208013)

[**5.3.2 System Components** 12](#_Toc113208014)

[**5.4 User Interface Design** 13](#_Toc113208015)

[**5.4.1 Structural Design** 13](#_Toc113208016)

[**5.4.2 Visual Design** 13](#_Toc113208017)

# **Introduction**

## **1.1 Background Information**

The assigned group-based project requires the completion of simple data analysis and visualisation through a graphical user interface. The subject of this task is a hand-picked dataset; Australian NSW Traffic Penalty data 2012 – 2017. The project consists of two phases: design and implementation.

The design element of this task consists of producing a project plan, which includes: a Project Overview, Work-Breakdown Structure, Activity Definition and Estimation, and a Gantt chart. The project plan will be used to define our goals and objectives by creating an overview and a work-breakdown structure, which will subsequently define our activities. A Gantt chart will be utilised for the scheduling and time estimation of tasks. A Software Design Document will be developed containing a Systems Vision Statement, a formal list of Software Requirements, Use Cases, System Components, Software Design, and an early Interface Design.

The implementation process will be determined based on the design created in the first phase. Before implementation, a testing plan will be formulated to thoroughly test various components of the system. Upon completion of testing a report will be made to process the results alongside a user manual that will explain how to use the software and show it features. To finish the implementation phase, an executive summary will be utilized to present the data over a 12-month period.

## **1.2 Scope**

The project objective is to design and implement a system that will enable users to have access to a wide range of information regarding NSW Traffic Penalty Data. The system will assist in the visualisation of data and be able to perform simple data analysis tasks. The goal is to explore and understand the insights of the overall severity of fines. This can be further expanded upon by looking at the type of fine (speeding, parking, etc.), the dates when these offences occur and the fine amount for each category of offence.

Generally, we will be able to answer questions such as:

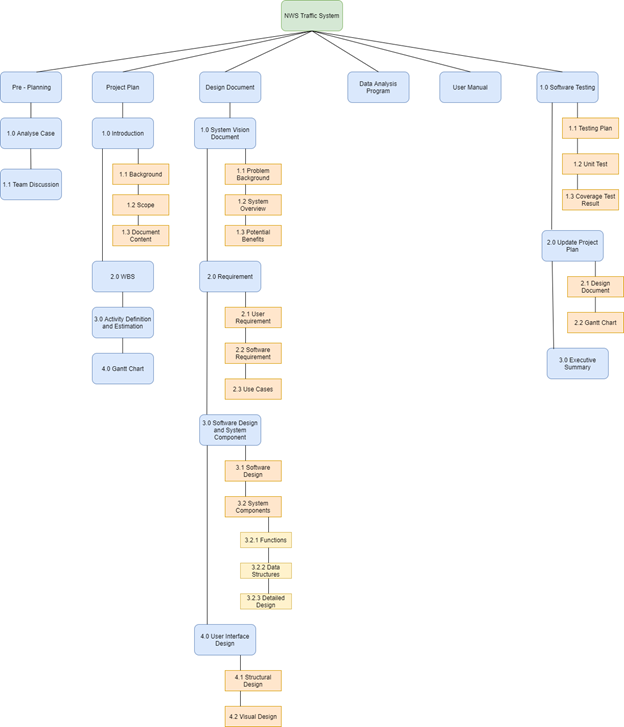
* What type of offence are more likely to occur?
* Where are the most offences being committed (location code)?
* Is the fined amount increasing over the years?
* Are the number/rate of fines increasing? If so, what measures can the NSW government take to prevent or reduce these offences?

Additionally, the system will be able to utilise all attributes and data to output a reasonable answer and provide a visual representation of that information (where possible) with given constraints. The system should provide an insight into the Australian NSW Traffic Penalty data to determine if more action must be taken to maintain road safety.

## **1.3 Document Contents**

The document contents of this project plan will feature a system to help analyse and visualise data for fines issued in NSW Australia by exploring the selected data set. The Australian NSW Traffic Penalty data features a various type of data which will be explored and presented in a user-friendly interface. The project plan will include documents such as A Work Breakdown Structure, Activity Definition and Estimation, A System Vision Document, and A Gnatt Chart. Additionally, it will include user and software requirements, use cases, software design and user interface design. The system will design then implemented and tested for certain requirement and criteria to access its functionality and reliability.

# **2.0 Work Breakdown Structure**



# **3.0 Activity Definition and Estimation**

|  |  |
| --- | --- |
| Activities | Duration |
| **Pre - Planning** | 1 Day |
| 1.0 - **Case Analysis:** We analyse the case and determine which dataset to use for this assigned task. | 1 day  5 August 2021 |
| 1.1 - **Team Discussion:** Used to determine when each team member is free for consecutive meetings each week. | 1 day  5 August 2021 |
| **Project Plan** | 4 days |
| 2.1 - **Background:** Provide background information for the task at hand. | 1 day  6 August 2021 |
| 2.2 - **Scope:** Determine the scope of the software being created - what will be included and what will not. | 1 day  6 August 2021 |
| 2.3 - **Document Content:**  Convey what will be covered in the supplied documentation. | 1 day  6 August 2021 |
| 2.4 - **WBS:** Determine each activity needed to be undertaken to complete the project. | 1 day  7 August 2021 |
| 2.5 - **Activity Definition and Estimation:** Defining each activity found in the Work Breakdown Structure, as well as determining a time estimation for the task. | 1 day  7 August 2021 |
| 2.6 - **Gantt Chart:** Used to determine the management of time regarding the completion of project activities. | 2 days  8 - 9 August 2021 |
| **Design Document** | 18 days |
| 3.1 - **Problem Background:** Describe the problem / reasoning as to why the system is being created. | 3 days  10 - 13 August 2021 |
| 3.2 - **System Overview:** List of capabilities that the system must be able to perform upon completion. | 3 days  10 - 13 August 2021 |
| 3.3 - **Potential Benefits:** List of potential benefits the system will bring upon completion. | 3 days  10 - 13 August 2021 |
| 3.4 - **User Requirement:** Details of what a user needs to be able to do to use the completed system. | 2 days  14 -  17 August 2021 |
| **Build Design** |  |
| 4.1 - **Software Requirement:** Details of what the software needs to be capable of and the functionality that it will provide. | 2 days  18 - 20 August 2021 |
| 4.2 - **Use Cases:** Details of how users will interact with the system. | 4 days  18 - 22 August 2021 |
| 4.3 - **Software Design:** Diagram of how the system will work. | 1 day  23 August 2021 |
| 4.3.1 - **Functions:** List of all the functions that make up the system, detailing return values, inputs, side effects etc. | 2 days  24 - 25 August 2021 |
| 4.3.2 - **Data Structures:** List of all the data structures in the system, detailing type, description members etc. | 1 day  25 August 2021 |
| 4.3.3 - **Detailed Design:** writing pseudo-code that runs on the algorithms of data structures | 4 days  26 - 30 August 2021 |
| 5.0 - **User Interface Design:** Description of the tools and insights used in the creation of the user interface design. | 2 days  31 August - 2 September 2021 |
| 5.1 - **Structural Design:** Navigational and informational structure of the system, presented with a diagram and discussion. | 5 days  31 August - 5 September 2021 |
| 5.2 - **Visual Design:** All the visual elements of the system, presented with wireframes, icons, layout etc. Supported with discussion. | 5 days  31 August - 5 September 2021 |

# **4.0 Gantt Chart**

Chart, bar chart

Description automatically generated

# **5.0 Software Design Document**

## **5.1 System Vision Document**

### **5.1.1 Problem background**

The NSW Government has an issue analysing and visualising the traffic penalty dataset. This leads to inconsistent funding allocations as well as other issues. Given the large amount of data collected, it is important that additional software be developed to allow for the NSW government to track common traffic penalties committed. The task assigned is to aid the NSW government in the designing and developing of a data analysis and visualization tool that can help to analyse the statistics of the NSW traffic penalties dataset. The software will include a graphical user interface which will allow the NSW government to pinpoint various trends and anomalies within the data to help make the roads safer.

### **5.1.2 System Overview**

 Upon completion, the system should be able to fulfill the following:

-        For a user-selected period, report the information of all penalty cases.

-        For a user-selected period, produce a chart to show the distribution of cases in each offence code

-        For a user-selected period, retrieve all cases captured by radar or camera based on offence description

-        Analysing the cases caused by mobile phone usage - i.e.: trend over time, offence code, and so on.

-        For a user-selected period, rank the offences based on the total money value spent on fines.

### **5.1.3 Potential Benefits**

It is estimated that the incorporation of this system will bear the following benefits:

-        Improve road safety for all NSW road users as the government will be able to create more relevant traffic penalties.

-        Faster implementation of road safety penalties as the data are displayed in an easy-to-understand format.

-        Increasing or decreasing penalty costs as it will be easier to determine how often a penalty occurs, and how important it is to decrease the occurrence rate.

-        Decrease the frequency of road incidents by identifying roads where tighter surveillance needs to be applied.

## **5.2 Software Requirements**

### **5.2.1 User Requirements**

The user is supposed to interact with the program through a graphical user interface. The graphical user interface will be easily usable by the end user. To do this, several factors are required. The user must be able to achieve the intended purpose through using the software. In the case of software that is being designed and developed, that is to extract specific data out of a dataset and then display the data in an easily presentable way. The user needs to be able to extract specific data out of the dataset to be easily presentable.

### **5.2.2 Software Requirements**

R1.1 The program shall accept dates presented DD-MM-YYYY as arguments for start and end dates.

R1.2 In the case improper format is entered or date is out of range, the program will provide respective errors.

R2.1 Users can input their license number or CRN to get information based on the case diagram NSW traffic system.

R3.1 Users can input their offence code to track their fine cost.

R4.1 The program shall recognise radio options and only allow one to be selected at a time.

R5.1 The program buttons shall route to the correct page.

### **5.2.3 Use Cases**

#### **Use Case Diagram**

#### Diagram Description automatically generated

#### Diagram Description automatically generatedDiagram Description automatically generated

#### Diagram Description automatically generated

Offence (Class):

* Year – string – setYear()
* Month – date – setMonth()
* Code - Int
* Type - String
  + School zone offence
  + Speeding offence
  + **Point to Point**
  + **Red light camera**
  + **Speed camera**
  + Seatbelt
  + Mobile phone
  + Parking
  + Criminal infringement notice (CIN)
  + Food Safety Offence
  + Bicycle/wheeled toy offence
* Legislation -String
* Description -String

Location (camera) (Class):

* Location code - Strong
* Location details – street, suburb, camera - String

Camera (Class):

* Camera type - String
* Camera indication (type offence):
  + **Point to Point**
  + **Red light camera**
  + **Speed camera**

Penalty (Class):

* Total number – int
* Face Value - number

## **5.3 Software Design and System Components**

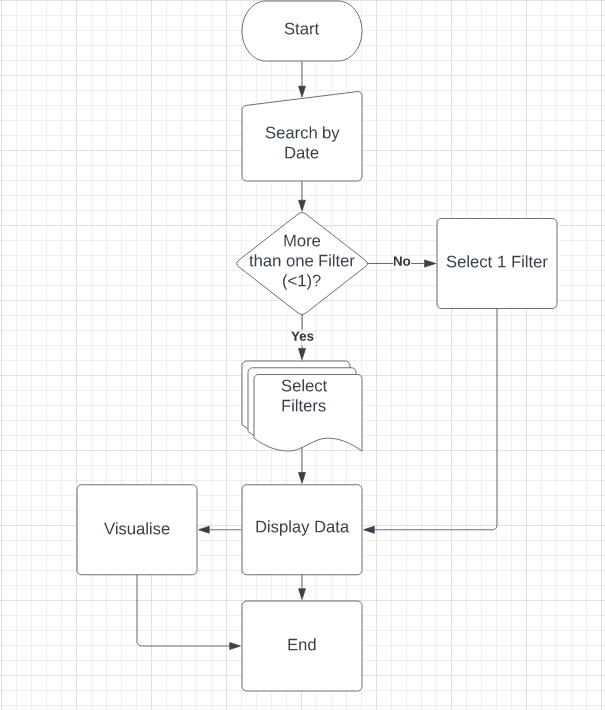
### **5.3.1 Software Design**

#### **Flow Chart**

**Diagram

Description automatically generated**

### 



### **5.3.2 System Components**

#### **Functions**

**Load Data (pandas.read\_csv*)***

* Read a comma-separated values (csv) file into a Data Frame.
* This function will take a csv input utilising newline character (\n) and tab character (\t) to preserve the format so python can read the data.
* This function will iterate or break files into chunks using additional parameters to help structure the data from the csv file.
* Returns a new DataFrame with data and labels into a table.

**Search Data**

* Read Integer input from user that meets requirements of Offence\_Code the unique identifier that returns all data in relation to that traffic penalty.
* Input parameter will only take an integer that meets the requirements of the Offence\_Code object. This input will be done through a search bar and will return all related data associated with the Offence\_Code.
* This function does not have any side effects and only returns data and does not edit or delete data.

**Update Data**

* Allows the database to handle in-place updates of opaque data type values. This allows management operators to update data values within the data structures.
* The function will take inputs based on what filter the user selects
* Data Types include String, Integer, Date, etc.
* Returns the updated information (if any) for the user to check the changes to the data.

**Time Period**

* This function allows a user to set a time period and produce data based on other filters.
* This function will have two inputs being a starting date and end date that will allow the data to be filtered between these dates. Alternative filters can be added to filter data based on other data such as Offence\_Code or Offence\_Description.
* This function has no side effects and will only display the data between the date ranges.
* This function will return all data between the given time period, unless the user specifies other filters which will then return data within the given parameters.

**Bind Function**

* This function binds an object to an event and provides it with a call back function.
* This function binds an object which in this case is a button to an event which will trigger the function to search or filter data.
* This function has no side effects and allows for the task to be completed before the function fires and produces the data.
* This function will return all data between the given time period, unless the user specifies other filters which will then return data within the given parameters.

#### **Data structures / Data Sources**

**Trees**

* A tree data structure will be used to connect data items using references in a hierarchical manner. This non-linear structure will be used for searching and organisation of the penalty data.
* This data structure will contain data members such as Offence, Legal, Camera, Location, Speed, Indicator and Total. These members will fall under the parent Offence category and will make searching the database easier while also allowing ease of organisation of child members.
* Functions such as Search Data, Time Period and Update data will use this data structure and will allow the user to query and update the purposed data.

**Arrays**

* An Array data structure will be used for storing more than one data item that has a similar data type. The array will help with processing many values at once and faster. Sorting and searching these values will allow for analysis to be quicker and easier. The Arrays will hold the values of the objects and help reduce the overall size of the code.
* Data members include Offence\_Code, Face\_Value, Total\_Number, Total\_Value etc. This will help with analysis and creation of charts as we can easily and quickly analyse the collection of data in the Array.
* Functions such as Time Period and Bind function will use array to create graphs and other data visualisation as it uses the collection of data in array.

#### **Detailed Design**

**Load Data**

import pandas as pd

# Read the csv file

df = pd.read-csv(“file.name”)

**Search Data**

# define the search function

Def Search():

Key = datatype(list())

Value = datatype([key])

Cur.execute(‘Select From’ .format(key, value))

Result = cur.fetchall()

Return result

**Update Date**

A = {1, 3, 5}

B = {2 ,4 ,6}

# adds items of B to A and Updates A

A.update(b)

Print (A)

Output ( {1, 2, 3, 4, 5, 6} )

**Bind function**

Widget.bind(event, handler, add=None)

## **5.4 User Interface Design**

The visual design elements of the Traffic Penalty Data Analysis and Visualisation system was created in accordance with the idea of seamless and simple. The layout is designed in a simple top-down structure that contain buttons with clear labels that perform simple tasks. Some options allow the user to customize the search for better analysis and visualisations. This layout design doesn’t reinvent the wheel, it takes what users already know in terms of positioning of buttons, bars, and lists, making it simple for users to interact with.

### **5.4.1 Structural Design**

### **5.4.2 Visual Design**

Font sizing for texts will be at least 18p for headings and at least 14p for other text. This is so that the text remains readable but doesn’t take up too much room in the system. The font used will be Berlin SANS FB as it is a bold font which will complement the style of the system. The intended style for this system is that of a public service presence because this tool will mainly be used by the NSW government, but also available to others. In keeping with that style, the colours that will be used by this tool are Navy blue, and other similar colours, varying in brightness. These colours represent calm and stability which fits with the style, and text is readable on top of its lighter variants.

Table

Description automatically generated with medium confidenceMock up Design:

1. **Column Head**

* Display name of column heads

1. **Data**

* Display raw data

1. **Search by keyword:**

* Filter the data by only displaying words relevant to the typed word

1. **Start Date:**

* Enter/Select start date (minimum 2012-01-1)

1. **End Date:**

* Enter/Select end date (maximum 2017-11-01)

1. **Filter:**

* Displays a separate box with different filter options
* If no filters are selected or selected “All” option -> Displays all data for user selected period

1. **Types of filters:**

* Once a filter is selected the data will automatically display in accordance with the filter
* Filters are the names of the data columns:

Table

Description automatically generated

1. **Visualise Tool:**

* Displays a graph or plot based on the selected filter/s (only if applicable)

1. **Clear:**

* Clear all inputs and output and returns to start stage

The mock-up demonstrates general idea of the layout and visual design. Most of the screen space is used up by the data. This is the raw data retrieved from the dataset. However, this display will change according to the user inputs; this can be done by selecting Start and End dates and/or selecting one or more filters. The layout structure for the LHS, consisting of buttons and list, is straightforward and easy to follow as each step is design from a top to bottom structure.

The Filter button allows the user to select from multiple filters or just one. This function will group the data and display them on the RHS (1 and 2). The Visualise function will can be used after the above steps have been completed. This will produce a graph or plot based on the information given (if applicable). Lastly, the “Clear” function will reset the interface and data back to the original state.