Software Design Document

NSW Traffic Penalty Analysis Software

Student Names:

* Zoe Waters - s5131193
* Martin Puncken - s5056926
* Elliott Horne - s5222361

**Table of Contents**

[System Vision](#_gjdgxs) **3**

[Problem Background](#_30j0zll) 3

[System Overview](#_1fob9te) 3

[Potential Benefits](#_3znysh7) 3

[Requirements](#_2et92p0) **4**

[User Requirements](#_tyjcwt) 4

[Software Requirements](#_fl4wqyv1im61) 4

[Use Cases & Use Case Diagrams](#_1t3h5sf) 4

[Software Design and System Components](#_4d34og8) **6**

[Software Design](#_iy1kql9hzvgx) 6

[System Components](#_khq1c5md40ol) 6

[Functions](#_p256e4y7xc2a) 6

[Data Structures / Data Sources](#_dpzxv3kyuhaa) 7

[Detailed Design](#_ehjhc73wt6n6) 9

[User Interface Design](#_2s8eyo1) **10**

[Structural Design](#_9uqiw9jzdbyi) 10

[Visual Design](#_el68is4th68r) 10

# **System Vision**

## **Problem Background**

The NSW government has asked for our group to create a program in order to help file and filter several entries that have been collected and compiled in a huge data set. This data has become quite extensive and extremely hard to navigate and filter through for certain criteria. The data focuses around traffic penalties that the NSW police have charged the public over the period of 2011 to 2017, with various information inserted into the data set from violation codes to the nature of the offence.

## **System Overview**

The system doesn’t need to be extensive or revolutionise but needs to be able to provide all possible outcomes that the data can provide. The system doesn’t need to be flashy or aesthetically pleasing but still needs to provide a clear and precise UI in order to keep a simple and clear process for any user to follow. The program just has to sort and display the data that is asked by the user and that is a nice simple basis to start on in order to build off of in the future if required.

## **Potential Benefits**

The potential benefits that this software could provide for the government is a more stable and easily access of information for them. This would also open up the possibility to apply this system to any other data dense problems they may face, with only the parameters of the filtering system limiting them to what they need. Other smaller benefits would be the ability to easily train any user about the system being a simple “click this for xyz information”. This would be beneficial in the long run as it provides an already complete system that can constantly have information entered or removed for archival if required.

# **Requirements**

## **User Requirements**

The user of the program would be required to know what they are looking for when it comes to certain criteria, such as violation codes or the legislation that the offence violates. They would need access to all aspects of the data from the codes to the date and to if certain variables are involved such as speeding cameras, witnesses etc. The user should be able to type or click on certain criteria and then press search in order to bring up a list of the sorted data, displaying the data that correlates with the criteria provided.

## **Software Requirements**

The software would require:

R1.1 The program shall be able to sort and distinguish data by types and content.

R1.2 The program shall be easy to follow/interpret

R1.3 The program shall be able to display and organise the data.

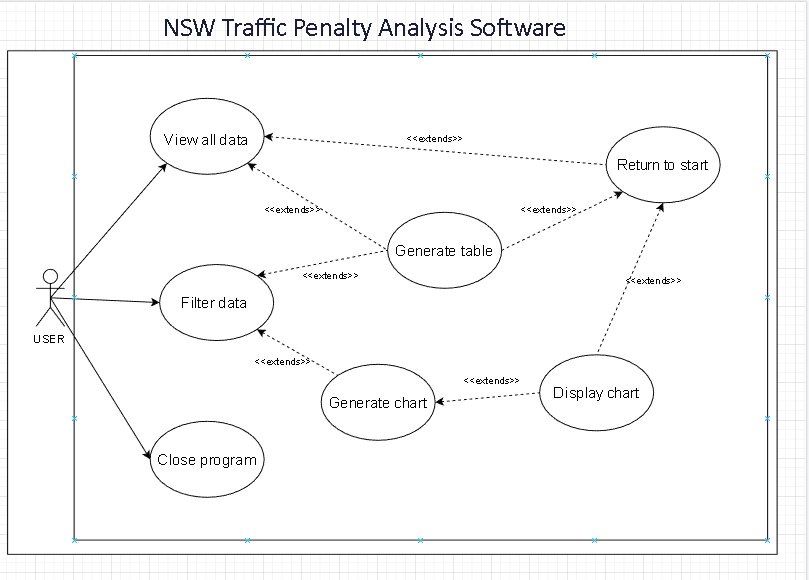
R1.4 The program shall be able to generate a graph.

## **Use Cases & Use Case Diagrams**

* + 1. **Use Case**

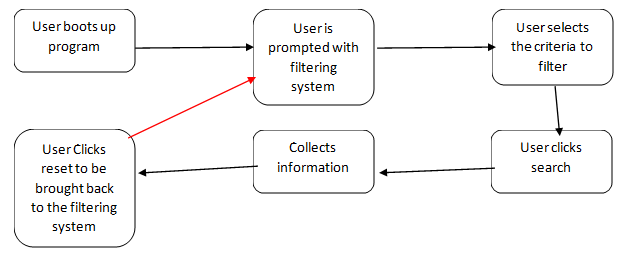
|  |  |
| --- | --- |
| **Use case name:** | Filter Information |
| **Scenario:** | Searching for certain information |
| **Triggering event:** | User requires to see offences with cameras involved |
| **Brief Description:** | User uses the filtering system and checks the box that has cameras marked in the data set, then displaying all the offences with cameras involved |
| **Actors** | User |
| **Related Use cases** | Generate chart |
| **Stakeholders** | User |
| **Preconditions** | Run application  Application on filtering page |
| **Postconditions** | User selects “camera involved” option  User selects “search” |
| **Flow of Activities (User)** | 1. User selects “camera involved” option 2. User selects “search” |
| **Flow of Activities (System)** | 1. System runs the data through its code 2. System removes the data that doesn't adhere to the constraints 3. System displays data that does adhere to the constraints |
| **Exception conditions:** | Other conditions are selected |

* + 1. **Use Case Diagram**



# **Software Design and System Components**

## **Software Design**



## **System Components**

### **Functions**

The functions of the software need to be able to take inputs from the user and match it with the dataset table that it’s provided. This is relatively a simple function but has a lot of different filters possible from numerical codes to descriptions of crimes.

**Function 1 - Ask user to select a time period**

* Description: This function asks the users to input a specific period of time so that future functions can display information from the SQLite database.
* Input Parameters: None
* Side effects: Two variables are created that will be passed to functions later on in the code
* Description of return value: The return value is that the user specified period is now stored within in two variables

**Function 2 - Display All Info**

* Description: Displays all the information records within the database between the selected time period the users has selected.
* Parameters (data types and what they’re used for):
  + startDate (Date) - The starting date of the user selected time period
  + endDate (Date) - The ending date of the user selected time period
  + NSW Traffic database - a database containing the NSW Traffic penalty data stored in a SQLite database
* Side effects: Array created to store relevant records
* Description of return value: All relevant records are displayed

**Function 3 - Generate Chart**

* Description: This function creates a chart displaying the distribution of case in each offence code for the user selected time period
* Parameters:
  + startDate (Date) - The starting date of the user selected time period
  + endDate (Date) - The ending date of the user selected time period
  + NSW Traffic database - a database containing the NSW Traffic penalty database stored in a SQLite database
* Side effects: Array created to store relevant records
* Description of return value: Graph with offence code distribution created

**Function 4 - Display all radar related cases**

* Description: Displays all information for radar and camera based offences during the user selected time period
* Parameters:
  + startDate (Date) - The starting date of the user selected time period
  + endDate (Date) - The ending date of the user selected time period
  + NSW Traffic database - a database containing the NSW Traffic penalty database stored in a SQLite database
* Side effects: Array created to store relevant records
* Description of return value: All radar related offences are displayed

**Function 5 - Analyse mobile phone offences**

* Description: This function analyses the offences that relate to mobile phones and display information such as trend over time, offence code etc
* Parameters:
  + NSW Traffic database - a database containing the NSW Traffic penalty database stored in a SQLite database
* Side effects: Array created to store relevant records
* Description of return value: Displays an analysis of mobile phone offences

**Function 6 - Search database for keyword**

* Description: This function searches the OFFENSE\_DESC for a specific keyword and displays the relevant records.
* Parameters:
  + startDate (Date) - The starting date of the user selected time period
  + endDate (Date) - The ending date of the user selected time period
  + NSW Traffic database - a database containing the NSW Traffic penalty database stored in a SQLite database
* Side effects: Array created to store relevant records
* Description of return value: Displays all offences with the keyword in in the OFFENSE\_DESC column

### **Data Structures / Data Sources**

**Data Type 1 - NSW Penalty Data**

* Type of Structure: SQLite Database
* Description: This database is used with any function that needs to display information from the NSW Traffic dataset to the user.
* Data members:

|  |  |  |
| --- | --- | --- |
| **Data Member Name** | **Data type** | **Description** |
| OFFENCE\_FINYEAR | String | The financial year in which the offence occurred |
| OFFENCE\_MONTH | Date | The first day of the month of the penalty notice |
| OFFENCE\_CODE | Int | The unique identifier that refers to the offense |
| OFFENCE\_DESC | String | A brief description of the offence |
| LEGISLATION | String | The legislation that creates the offence |
| SECTION\_CLAUSE | String | The clause in the legislation that is relevant to the offence |
| FACE\_VALUE | Number | The dollar amount of the fine attached to the offence |
| CAMERA\_IND | String | An identifier for offences that are “Camera detected“ related |
| CAMERA\_IND | String | The type of camera used to detect the offence |
| LOCATION\_CODE | String | A unique identifier for the location of a camera |
| LOCATION\_DETAILS | String | Location details relating to the offence such as street, suburb and the direction of the camera |
| SCHOOL*ZONE*IND | String | An indicator for offences that are “School zone“ related |
| SPEED\_BAND | String | The speed range |
| SPEED\_IND | String | An indicator for offences that are “Speeding“ related |
| POINT*TO*POINT\_IND | String | An indicator for offences that are “Point to Point“ related |
| RED*LIGHT*CAMERA\_IND | String | An indicator for offences that are “Red light Camera“ related |
| SPEED*CAMERA*IND | String | An indicator for offences that are “Speed Camera“ related |
| SEATBELT\_IND | String | An indicator for offenses that are “Seatbelt“ related |
| MOBILE*PHONE*IND | String | An indicator for offences that are “Mobile phone“ related |
| PARKING\_IND | String | An indicator for offences that are “Parking“ related |
| CINS\_IND | String | An indicator for offences that are “Criminal Infringement Notice Scheme“ related |
| FOOD\_IND | String | An indicator for offences that are “Food Safety“ related |
| BICYCLE*TOY*ETC\_IND | String | An indicator for offences that are “Bicycle, Wheeled toy and other Non-motor vehicle“ related |
| TOTAL\_NUMBER | Int | The number of penalty notices issued for the offence |
| TOTAL\_VALUE | Number | The face value for all offence penalties issued |

* Functions:
  + **Display All Info**
  + **Generate chart**
  + **Display all radar related cases**
  + **Analyse mobile phone offences**
  + **Search database for keyword**

### **Detailed Design**

**Display All info:**

Get user selected start date

Store Start date in date type variable

Get user selected end date

Store end date in date type variable

Store dataset in array type variable

loop dataset array

Store all elements from dataset array that are between start date variable and end date variable in an output array type variable

end loop

Return output array variable

**Generate Chart:**

Get user selected start date

Store Start date in date type variable

Get user selected end date

Store end date in date type variable

Store dataset in array type variable

loop dataset array

Count each offence codes for records that are between start date variable and end date and store result in int variable

End loop

Generate chart using Offence Code Counts variables

**Display all radar related cases:**

Get user selected start date

Store Start date in date type variable

Get user selected end date

Store end date in date type variable

Store dataset in array type variable

loop dataset array

Store all elements from dataset array that are between start date variable and end date variable and are radar or camera based offence in an output array type variable

End loop

Return output array variable

**Analyse mobile phone offences:**

Get user selected start date

Store Start date in date type variable

Get user selected end date

Store end date in date type variable

Store dataset in array type variable

loop dataset array

Store all elements from database array that are mobile phone related in an output array type variable

End loop

Display analysis tools on output array variable

**Search records for keywords:**

Get user selected start date

Store Start date in date type variable

Get user selected end date

Store end date in date type variable

Store dataset in array type variable

loop dataset array

If Start date and End date are empty

Store all elements from database array that have keyword in OFFENCE\_DESC in an output array type variable

If Start date and End date are not empty

Store all elements from database array that have keyword in OFFENCE\_DESC and that are between start date variable and end date in an output array type variable

End loop

Return output array variable

# **User Interface Design**

The application is for analysing traffic data, so the audience of the application will most likely be professionals in government agencies such as law enforcement. With this in mind, the application’s design should also be professional in appearance. The application purpose is to filter the large amount of data into a more specific readable form so the design only includes information essential to the application’s function.

## **Structural Design**

The structure of the application's design will be a minimalist approach, keeping the number of the application’s pages to a minimum to reduce the user's need to navigate far from the main function screen. The title (Element 1) of the application will consist over each page so the user knows what application they are using. The query functions (Element 2) will be listed along the left side of the screen so the user can switch between them whenever they are on the main screen. The search details(Element 3) will be displayed on the right side of the screen and will consist of user inputs and instructions required for the query selected.

Changing the query(Element 2) will not change the page, just change the information in the search details (Element 3). This keeps all the important functions in one place so the user can always find what they are looking for.

Pressing the search button(Element 4) will take the user’s input in the search details(Element 2) and create a query to the dataset. A new page will then display the information(Element 5) returned from the query. A return button (Element 6) will be present in each result page in the upper left, so the user will know how to navigate back to the main page of the application.

In the upper right hand corner, there will be an exit button(Element 7) which will quit the application. This function will be available on all pages for user convenience.

## **Visual Design**

The layout of the main page groups sets of information together, the title of the page, the queries available to the user and the search details. This decision was to improve readability/usability as the information and function location is consistent in each location. The location of the return(Element 6) and close(Element 7) buttons are consistent with standards of websites and other page based applications, which will help the user's cognition in finding them and understanding their usage..

The visual elements were created to keep the application readability as well as draw attention to important information, an example of this is a highlighted query (Example 8). The highlighted query uses inverted colours to grab the users attention so they know which query is selected. The input (Element 9) instructs the user exactly what information is required to reduce input errors from the user.

The return(Element 6) and close(Element 7) buttons have icons that are of common usage. The exit button is just an ‘X’ which has been used as the icon for the exit function for decades as part of the Microsoft standard. The return button has an arrow that is pointing backwards which indicates to the user, that along with the word return, that the button will take them back to the previous screen.

The graphics, style, colour and font all have professionalism and simplistic look in mind. The font is the san-serif font Calibri, this font is very readable and being San-serif, it lacks any decorative strokes, which gives the font the professional look. The font is also default on Windows, which increases the compatibility of the application. The style is very squared and to the point, to promote the professional look. The colours will be white and black except in situations where the user's attention is required to be grabbed, such as in Element 10 and Element 11 where data is highlighted by colour.

