Software Design Document

Data Analysis Tool: NSW Traffic Penalties

Brianne Byer

Wonwoo Choi

Marco Querzola

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

## System Overview

## Potential Benefits

# Requirements

## User Requirements

This program is designed for the user to be able to search through a set of data relating to traffic penalties in NSW between 2011 and 2018. The user will first be able to select a period they want to search from ranging from the whole data set to a day or week window. Once the period is selected the user can search for certain key words such as mobile phone usage or drink driving offences. During a selected time period the user will be able to produce a chart which shows a distribution of number of cases in each offence code, this will help to get an idea of what is the most common breach of NSW traffic rules. This program will be a very useful analysis tool and would help those working in analytical roles within the NSW government especially those involved in the Transport for NSW department.

## Software Requirements

R1.1 The program shall ask for input for a date range between 2011-2018, this can be anywhere from years to days.

R1.2 The program will then provide a search box where user can filter offences by type of offence or various keywords.

R1.3 The program shall provide a list of offences relevant to type/keyword. User can click on each offence to get more information.

R1.4 The program shall provide a chart showing the distribution of number offences in each code.

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

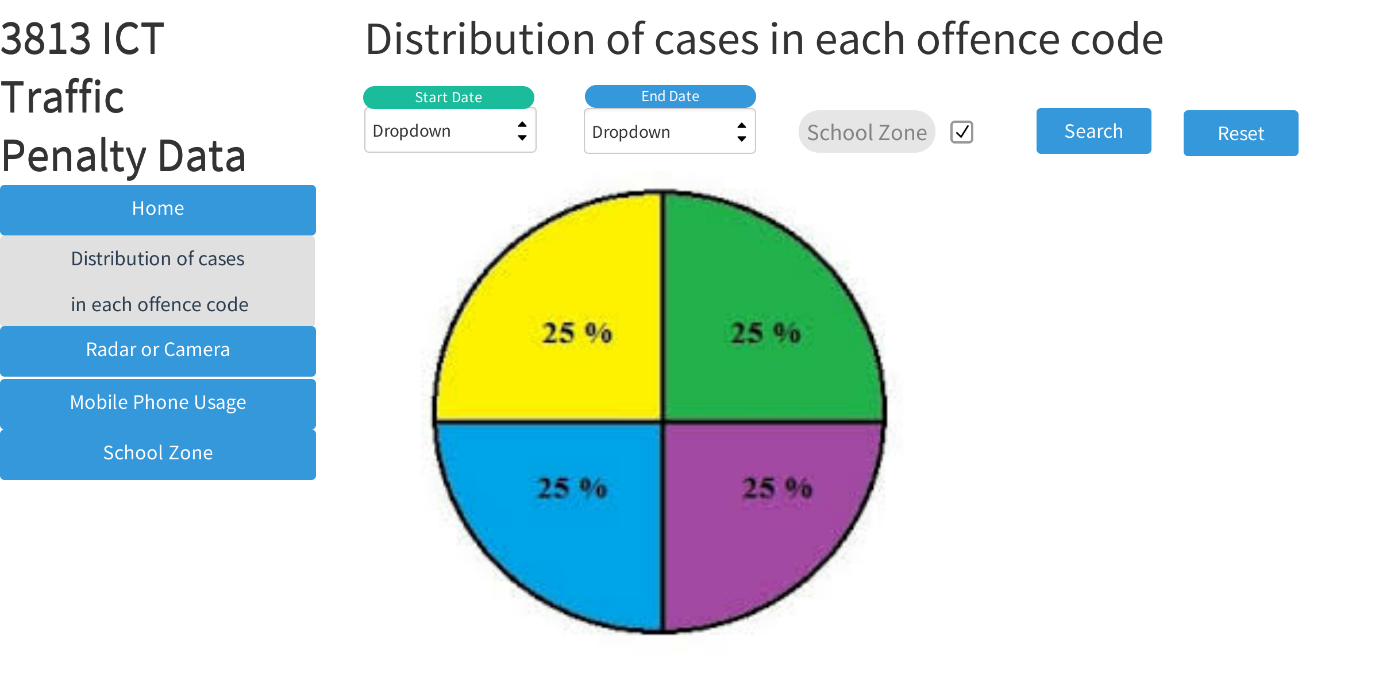
Detail your visual design: Layout, visual elements, icons, graphics, style, colour, fonts general screen designs. This can be sketches, wireframes, mockups etc, supported by a discussion, explanation, and justification of your choices.

## Main Page

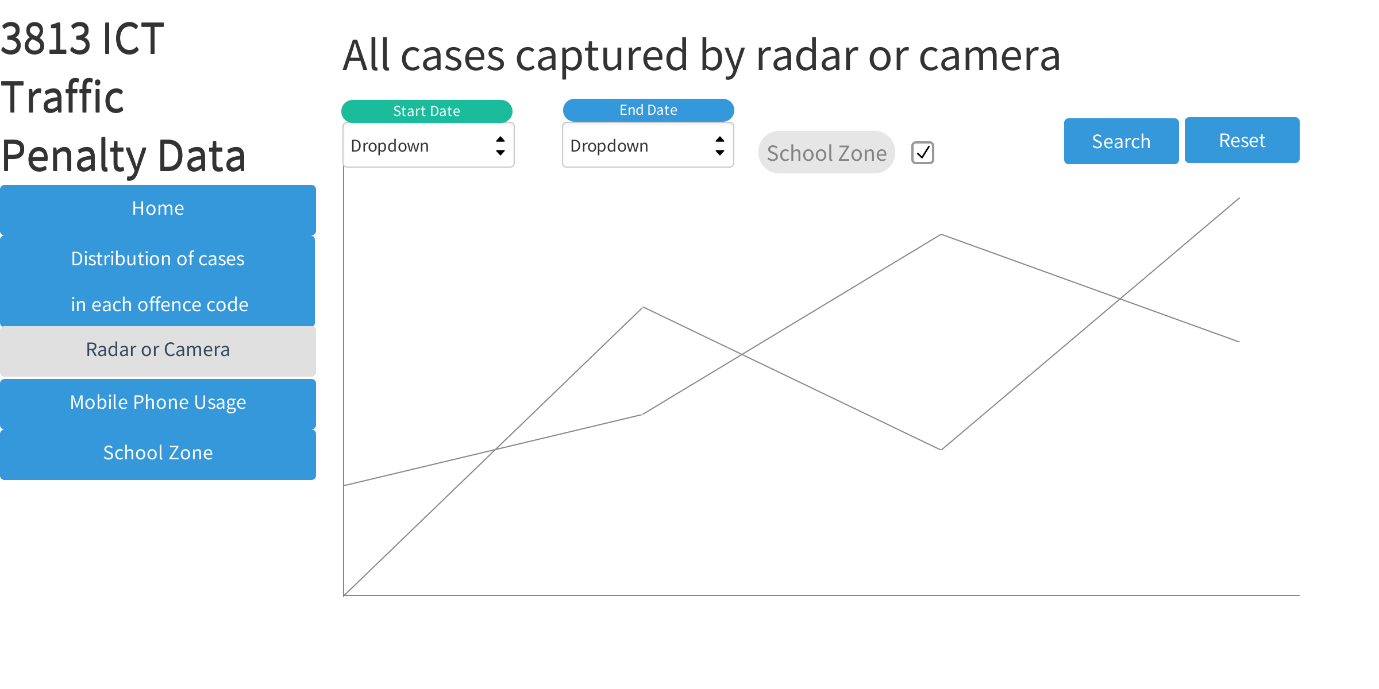
테이블이(가) 표시된 사진

자동 생성된 설명

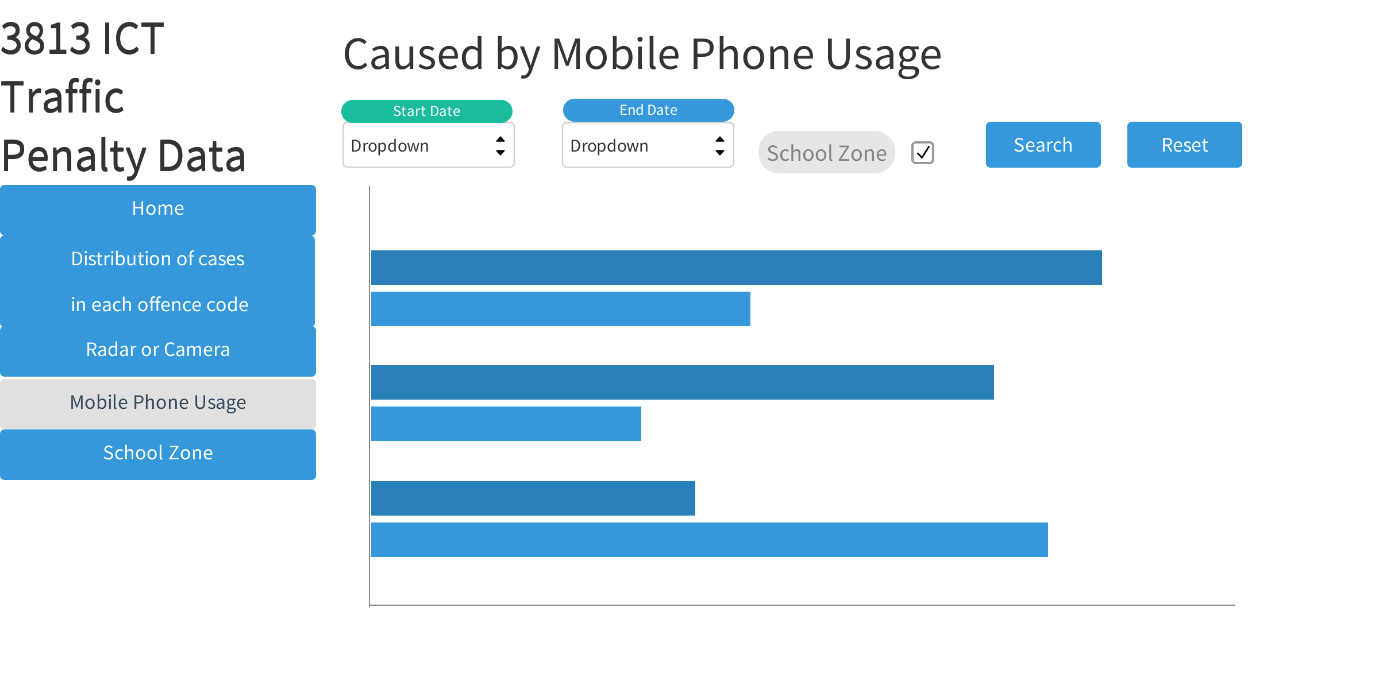
## Distribution of Cases in each offence code



## All cases captured by radar or camera



## Cases caused by mobile phone usage



## Penalty caused in School Zone

