Software Design Document

<Accident Data User Interface (Placeholder)>

Lambert Niyuhire, Cody Perkins, Mitchell Reid

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

A simple data analysis and visualization tool for a dataset. You will need to design and implement this tool, and it must provide a graphical user interface that will handle the analysis and visualization of the data.

## Problem Background

What’s the dataset?

Victora state accident dataset, crash statistics from 2015-2020 for Victoria. A way to analyse and represent this data in an easy to use and fast manner. An application that allows users to manipulate the dataset compare different parts of the dataset. Use keywords to compare and contrast different data points. Graph relevant data points and trends over time and display correlations between different factors.

What sort of data input/ouput is needed?

Text input and mouse clicks are examples of input that the software will receive. The text input will constitute users searching keywords, the mouse clicks users selecting periods of time. The software will output tabulated accident data and be able to formulate bar graphs for specific data points.

What problem does this solve?

The application allows users to automate certain data processing techniques. Saving the user time by generating results that allow users to check if specific keywords are tied to different types of crash data (e.g., determining how many accidents involved pedestrians). This frees the user from having to manually cross-check the dataset which would be immensely time-consuming and cumbersome for a dataset of this size. The software also provides the user with aggregated ready-to-use data representations in the form of bar graphs. These graphs can be taken directly from the interface and pasted into presentations. Supplying users with a way to tailor-make their own graphs to visually display whatever part of the dataset they are interested in.

Who’s going to use it and why?

The target audience for this software is individuals or entities with a vested interest in Victorian accident data. This is not limited to the Victorian Police, private insurance companies, government committees dedicated to road safety (e.g., Victorian Parliamentary Safety Committee), The Department of Transport and Planning, and non-profit organisations (e.g., Road Safety Promotion Australia). These entities will all use the software to review and draw conclusions on vehicular accidents in the state of Victoria. A non-profit may use this data to enact a target road safety plan. An insurance company might use this data to increase or decrease what they charge in excess for age groups.

## System Overview

What will the system do?

The proposed system will allow end users to manipulate the Victora state accident dataset. Displaying information in an accessible and easy to use manner. The key features of the system are as follows:

* Allow users to display all accidents that happened in a specific time-period.
* Produce a bar graph that displays the average number of accidents in each hour of the day for a selected period.
* Search a selected period by a keyword (e.g. collision) and retrieve all accidents caused by this accident type.
* Enable the user to examine, the impact of alcohol on accidents (e.g. most common accident type associated with alcohol).
* Allow users to graph the accident fatalities over the 5-year period to investigate data trends.

For a user-selected period, display the information of all accidents that happened in the period.

For a user-selected period, produce a chart to show the number of accidents in each hour of the day (on average).

For a user-selected period, retrieve all accidents caused by an accident type that contains a keyword (user entered), e.g. collision, pedestrian.

Allow the user to analyze the impact of alcohol in accidents – ie: trends over time, accident types involving alcohol, etc.

Features, functions

Graphic user interface (GUI) for

What can’t it do?

## Potential Benefits

How will this create a benefit?

# Requirements

## User Requirements

In this section you detail how a user is supposed to interact with or use your program. What do they ***need*** to be able to do? This should all be from the end users perspective. Can be a combination of narrative text and listing of needs.

**Assignment note: You have not been given a client/user, so you can make one up. Who do you think would be using your software?**

Insurance companies

## Software Requirements

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.