Software Design Document

<Project Name>

Student Names

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

Improving road safety helps alleviate the corresponding social and economic costs brought about by road trauma. Victoria carries the 2nd biggest cost burden across Australian states, valued at $6 billion per year, which not just affects the crash victims, but also their families, other road users, the Commonwealth, and the Victoria State Government. This includes direct costs such as hospitalisation, medication, rehabilitation, and property damage, as well as other potential costs due to loss in productivity of patients, those who allot time to provide informal care, and the surrounding community (Steinhauser & Lancsar, 2022). Raw data that shows details of each road crash between the 3rd quarter of 2013 to 1st quarter of 2019 within Victoria is already available. The goal of the proposed system is to make the said data be more meaningful and useful by providing a user interface where data can be aggregated or broken down into more detailed information, to show status and trends by user-selected parameters (e.g., period, area, type of accident etc.). This will allow the government to measure if the polices that were set to improve road safety are effective, based on actual performance versus identified goals or metrics (e.g., lowering the number of alcohol-related accidents by a set percentage versus previous year).

## System Overview

The system will be capable of performing simple data analytics tasks, with the following output that will be visualised on a dashboard:

* Information of all accidents based on a user-selected period.
* A chart showing the average number of accidents in each hour of the day based on a user-selected period.
* Retrieve all accidents caused by an accident type that contains a keyword entered by the user (e.g., collision, pedestrian), based on a user-selected period.
* A chart that shows the impact of alcohol in accidents with the following filters:
  + Period Covered: Number of accidents by year, by month across a 5-year period, by day across a 5-year period, day of the week, hour of the day.
  + Type of accident
* A chart showing accidents that caused harm to each type of road user i.e., pedestrian, cyclists, motorists, older people etc., based on user-selected period and area.

## Potential Benefits

The primary benefit of the system is its potential to lower the social and economic costs brought about by road crashes, by serving as a tool to measure the performance of polices and departments in improving road safety, which in turn will serve as basis for policy improvements.

# 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?**

## 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.

# REFERENCES

Steinhauser, R., & Lancsar, E. (2022, September). Social Cost of Road Crashes: *Report for the Bureau of Infrastructure and Transport Research Economics* (Final Report). The Australian National University. <https://www.bitre.gov.au/sites/default/files/documents/social-cost-of-road-crashes.pdf>