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

Road accidents have been a persistent issue in Victoria, Australia, with numerous fatalities and injuries reported over the years. The State of Victoria has accumulated data from 2015 to 2020, detailing various aspects of these accidents. However, the raw data, while extensive, doesn't provide immediate insights or patterns that can be used for preventive measures or policymaking. There's a need for a tool that can analyse, visualise, and provide insights from this data to various stakeholders, including accident commissions, insurance companies, and the public.

## System Overview

The proposed system will be a data analysis and visualization tool tailored for the road crash statistics dataset of Victoria. This tool will offer a user-friendly graphical interface, allowing users to filter and understand specific aspects of the data, such as crash types, locations, conditions, and more. The software will not only address the predefined analysis tasks but will also introduce an additional unique insight feature, derived from the dataset, to provide a comprehensive understanding of the road safety situation in Victoria. 

## Potential Benefits

**Informed Decision Making:** Policymakers can use the insights from the tool to make informed decisions regarding road safety measures, infrastructure development, and public awareness campaigns.

**Insurance Insights:** Insurance companies can gain a better understanding of high-risk areas, types of accidents, and other relevant data points to adjust their policies and premiums accordingly.

**Public Awareness:** The general public can access and understand the data, leading to increased awareness about high-risk areas, times, and other factors. This can potentially lead to safer driving habits.

**Resource Allocation:** By identifying high-risk areas and times, resources such as ambulances, police, and emergency services can be allocated more efficiently.

**Continuous Improvement:** As the tool will be continuously updated with new data, it will evolve and provide more accurate and timely insights, leading to a dynamic approach to road safety in Victoria.

# 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.