Software Design Document

**Victoria State Accident Data Analysis**

**S5254936 – Bebin Roy - System Vision (10 marks), User Interface Design (15 marks)**

**S5182075 – Atticus Burgess - Requirements (10 marks), I CAN HELP WITH ANYTHING JUST ASK :)**

**S5299401 – Ansh - Software Components and Software Design (15 marks)**

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

## In the initial stages of our interface design, we employed a combination of modern design tools and methodologies to ensure a user-centric approach. Our primary objective was to create an interface that is both intuitive and efficient, catering to the diverse needs of our users.

Tools used:  
Draw.io: This digital design toolkit allowed us to create a blueprint of our interface, laying out the basic structure and elements.   
  
Figma: Leveraging Figma's collaborative features, our team was able to work simultaneously on design prototypes, ensuring real-time feedback and iterations. Its vector-based tools also facilitated the creation of high-fidelity mock-ups.   
  
Adobe XD: For interactive prototyping, we turned to Adobe XD. This enabled us to simulate the user experience, testing the flow and transitions between different sections of our application.

## Structural Design

.

The structure of this product is designed to provide users with a seamless experience, allowing them to easily navigate through the data, apply necessary filters, visualize the data in tables or graphs, and export the results. The choices made in this design prioritize user convenience and efficiency, ensuring that users can quickly access and analyse the data they need.

## Visual Design

A screenshot of a computer

Description automatically generated

**Visual Design Justification:**

Our visual design prioritizes clarity, efficiency, and user-friendliness. The layout was crafted to ensure that users can quickly access and interact with the most crucial features.

* **Date and Time Selector at the Top**: Placing the date and time selector at the top is to ensure the users are able to filter to apply when analysing data. Its prominent position ensures that users can quickly set their desired time frame simply.
* **Alcohol Influence Filter**: Given the significance of understanding the impact of alcohol in accidents, a straightforward 'yes' or 'no' toggle was integrated. This choice simplifies the filtering process, allowing users to instantly view accidents influenced by alcohol.
* **Casualty Dropdown and Keyword Description**: These features are essential for users who want a more filtered analysis. By providing a dropdown for the number of casualties and a keyword description, users can tailor their data view to specific scenarios or types of accidents. This may include locations and maybe vehicle descriptions.
* **Apply and Reset Buttons**: These buttons are placed near the filters for immediate action. The 'apply' button ensures users can confirm their choices, while the 'reset' button offers a quick way to start afresh, enhancing user experience by offering flexibility.
* **Graph and Table View on the Right**: Visual data representation is vital for intuitive understanding. By placing the graph and table view on the right, users can immediately see the results of their filters, making data analysis more seamless.
* **Export Data Button**: Recognizing the need for users to utilize the data outside the platform, an export button is conveniently placed. This ensures that users can easily take their insights and share or further analyse them in different environments.

In conclusion, every element of our wireframe is designed with the user's needs in mind. The layout and visual elements are not just aesthetic choices but are rooted in the objective of making data analysis as intuitive and efficient as possible for our users.