Software Design Document

Victorian State Accident  
Data Analysis  
and Visualisation Tool

Cooper Mitchell s5185611, Thomas Jensz s5099828

Table of Contents

[1.0 System Vision 3](#_Toc49282815)

[1.1 Problem Background 3](#_Toc49282816)

[1.2 System Overview 3](#_Toc49282817)

[1.3 Potential Benefits 3](#_Toc49282818)

[2.0 Requirements 4](#_Toc49282819)

[2.1 User Requirements 4](#_Toc49282820)

[2.2 Software Requirements 4](#_Toc49282821)

[2.3 Use Cases 4](#_Toc49282822)

[3.0 Software Design and System Components 6](#_Toc49282823)

[3.1 Software Design 6](#_Toc49282824)

[3.2 System Components 6](#_Toc49282825)

[3.2.1 Functions 6](#_Toc49282826)

[3.2.2 Data Structures 6](#_Toc49282827)

[3.2.3 Detailed Design 6](#_Toc49282828)

[4.0 User Interface Design 7](#_Toc49282829)

[4.1 Structural Design 7](#_Toc49282830)

[4.2 Visual Design 7](#_Toc49282831)

# System Vision

## Problem Background

The Victorian State records all road crash statistics from the past 5 years into a dataset. This dataset contains extremely useful information including insurance related interviews for data scientists, road safety research, fatal and injury crash analysis and much more. This data needs to be easily accessible, searchable and user friendly in order to gain the most efficient analysis of these statistics. It is important for researchers and the government to be able to select certain periods of time, crash types, alcohol impact and other trends from the data.

It is recommended that a new software should be developed in order to satisfy these requirements and negate the issues that are presented when manually searching a dataset.

## System Overview

The new system should be capable of searching for each individual attribute of the dataset e.g. time, location, conditions, crash type, road user type, object hit, alcohol present, description etc. The user must be able to select a time period of records. It must be able to distinguish relationships and trends between each crash. The most optimal means for the system to be deployed on will be an application on desktop computers or tablets. The data will need strong visualization and graphical aspects upon presentation to the user.

## Potential Benefits

With the implementation of the new Victorian State crash record system, there is potential to make improvements to the road safety of Victorian roads. The trends and statistics may be able to point researchers towards the root causes of crashes in the state. Knowing these causes, the government will be able to change legislation or infrastructure accordingly. Ultimately the soul benefit of the system would be to reduce the number of injuries and fatalities on the road.

# Requirements

## User Requirements

The user of the software will be able to interact with it via their input, either mouse, keyboard or touchscreen. The user interface will present all instructions in a simplistic manner and the user will only have to input their desired selections from the data. Their selections will include period of time, keywords from DCA\_CODE (e.g. animal, rear end), alcohol impact or relationship between two or more attributes.

The user must only understand what each detail of a crash means, the information given includes an accident number, type of accident and a DCA\_CODE (description). It is expected that the user already has knowledge of this dataset.

In order to restrict access to certain users the software will be protected by account login to verify user integrity based on the assumption that they have government permission.

## Software Requirements

**R1.1** The program shall accept new data as soon as it is entered into the database.

**R1.2** New data entered into the database shall update any existing accident analysis and reports.

**R1.3** The program shall provide searching capabilities for user-selected periods and user-selected keywords.

**R1.4** The program can generate graphical statistics to present the data from the dataset.

**R2.1** The program will contain security measures, specifically a user verification to gain access.

## Use Cases

**Case 1:** A user wishes to view all crash data from the past 5 years and manually scroll through it.

**Case 2:** A user wishes to view crash data from the past year. The user inputs this specified date. The system will present records of all crash data from exactly one year ago to today.

**Case 3:** A user enters a keyword, and the system returns all crash data containing the relevant keyword within any specified attribute.

**Case 4:** A user enters a keyword and specific dates, the system will then return all crash data containing the relevant keyword and within the specified dates.

# Software Design and System Components

## Software Design

## System Components

### Functions

**Login function:**

The login function takes two input fields (username and password) and checks them against stored keys of the elevated users who have permission to access the software. The input parameters are any text and numbers less than 50 characters long for each field. These inputs are string data types. The variable updated by this function is the current user variable. The function returns a Boolean true or false value which depicts if the user can access the rest of the program.

**Display raw dataset function:**

The display raw dataset function will take the database of crash records and simply print it. There are no input parameters. No variables are changed as the data is only being read. The function returns the raw dataset.

**Update raw dataset function:**

The update raw dataset function takes new data and inputs it into the crash record database. The new data is input via a spreadsheet of new records. The user inputs the new data which must fit the parameters of the database, each input will be a string data type. The variables changed are the arrays which contain the crash records.

**Search dataset function:**

The search dataset function will take the users search parameters and check them against the database to print records which correspond with the users searches. There are 4 variables the system can take, date from, date to, keyword 1 and keyword 2. Date from and date to are date data types and the keywords are strings. The function returns a new dataset containing the search results and is temporarily created and printed.

**Generate visual statistics and graphs function:**

**Compare search results to analyse impact:**

### Data Structures

**Accident** – A list that stores an individual record of an accident and its associated attributes, such as ABS\_CODE, ACCIDENT\_DATE and ALCOHOLTIME.   
Used by: DisplayDataset, UpdateDataset, SearchDataset, GenerateGraph, CompareResults

**AccidentSet** – A list that stores all accidents.  
Used by: DisplayDataset, UpdateDataset, SearchDataset, GenerateGraph, CompareResults

**User** – A list that stores an individual record of an authorised user and all of their associated attributes, such as Name and Password.  
Used by: Login

**UserSet** – A list of all authorised users.  
Used by: Login

**Results** – A list that stores accidents that match the query criteria.  
Used by: SearchDataset, GenerateGraph, CompareResults

**Results2** – A list that stores another set of query results, to allow comparison to the first set.  
Used by: SearchDataset, GenerateGraph, CompareResults

**LinearGraphData** – A list that stores x and y axis information based on query results  
Used by: GenerateGraph, CompareResults

**ProportionalGraphData** – A list that stores percentages based on query results  
Used by: GenerateGraph, CompareResults

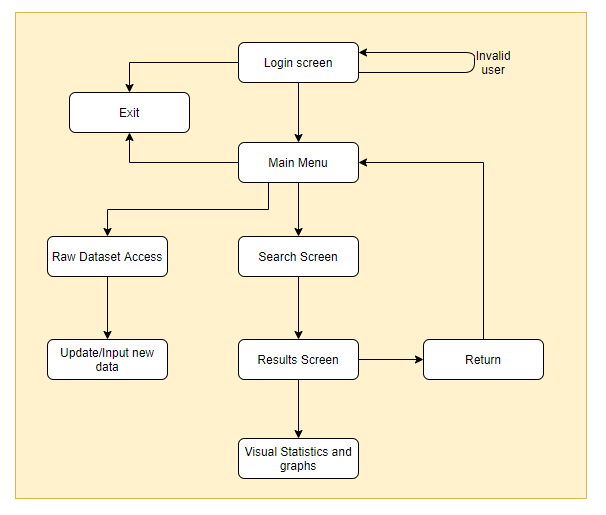
### Detailed Design

# User Interface Design

For the initial design of the accident software, careful collaborative planning was used to arrive at the first user interface design. This design encompasses each of the objectives and requirements discovered in parts 1, 2 and 3. The tool used in this design stage was Draw.io for the diagrams. The structural design will be presented, meaning navigation through the software, each interface that the user can visit and how to get there. This visual design will also be shown, meaning prototypes of the actual graphical design of the interfaces. Including the layout and all visual elements. Justification for each of these design decisions will be presented.

## Structural Design

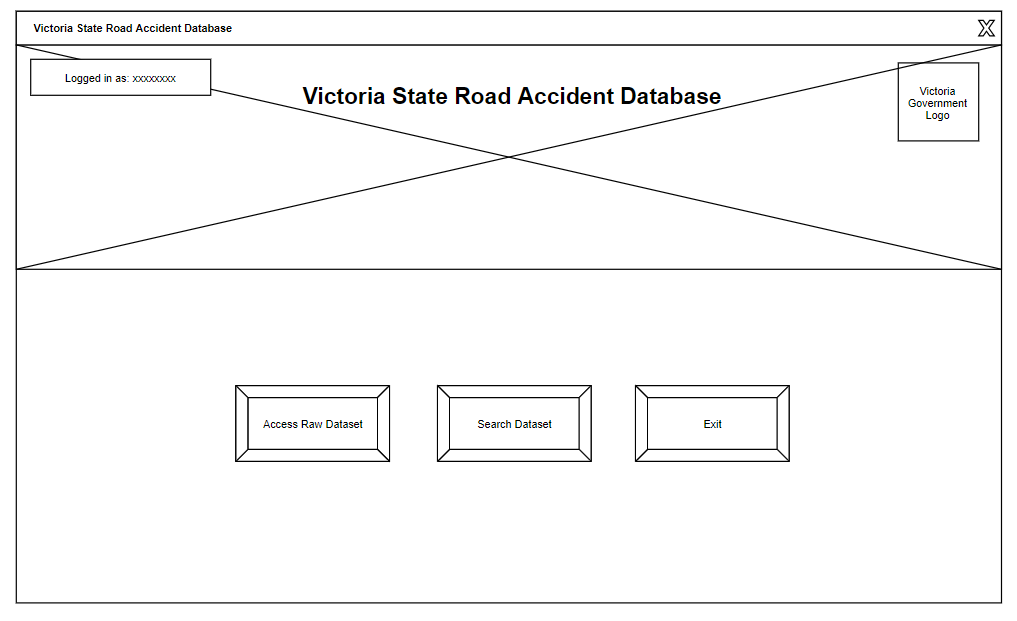
The structural design of the accident software can be seen in the navigational structure diagram below.

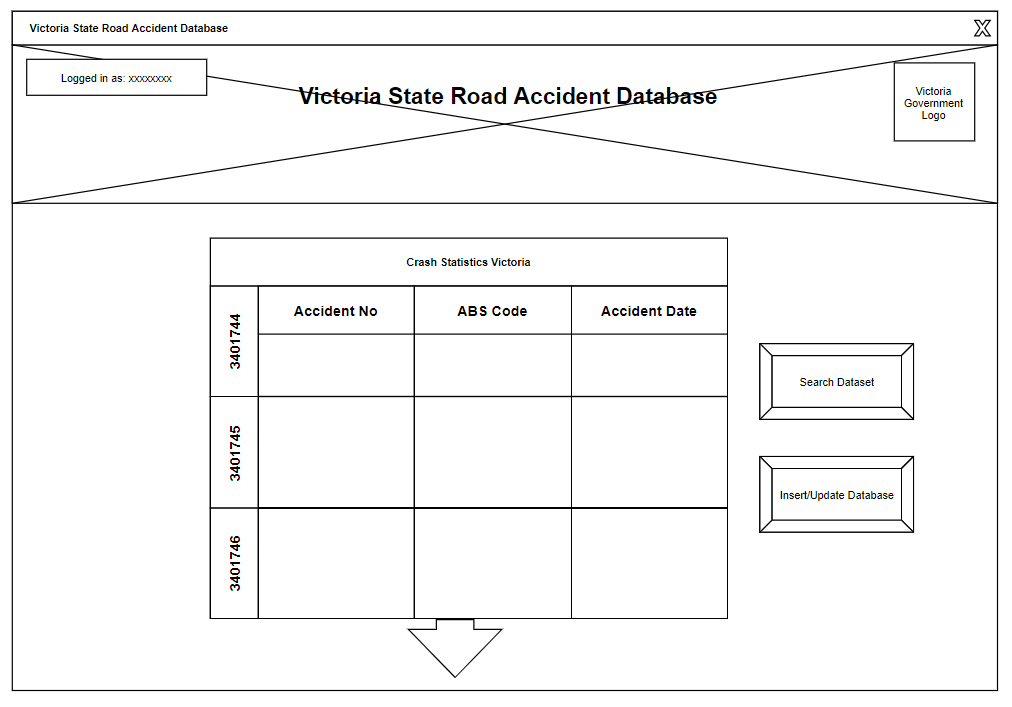


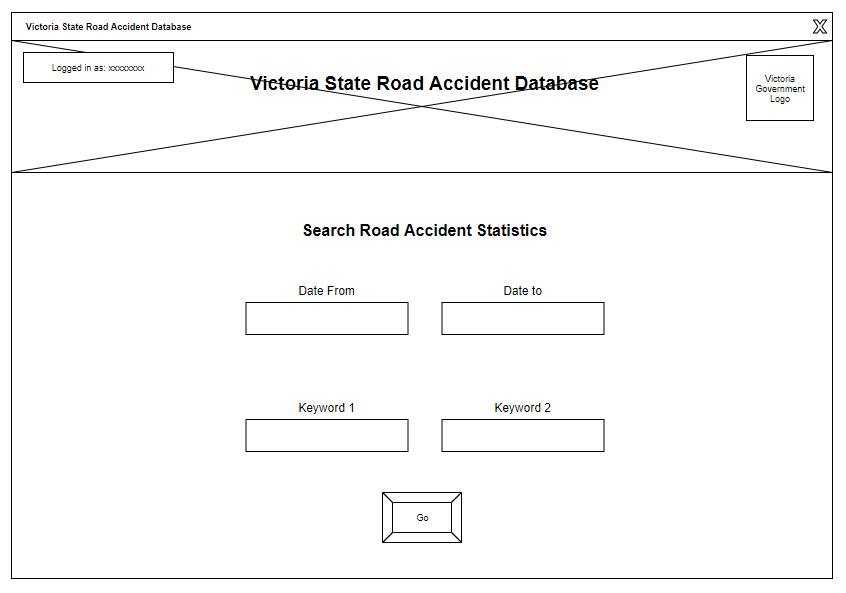
The diagram displays the structure in a hierarchal manner. Each element represents a page of the program and the arrows represent the navigation between the pages. Each page will have its own interface, displaying the relevant data or actions that can be taken from that page. This structure allows for the most simplistic approach for accessing the crash data while also offering further detail if it is desired. From the main menu, the user can immediately access all use cases. They may choose to access the raw dataset if they desired, or search for specific data. And going further from each of these options, can perform more elaborate functions such as inputting new data or viewing visual statistics. Return and Exit are functions instead of pages. This structure was chosen due to its straightforward nature. The user of this software is most likely looking for an efficient data retrieval method and not wasting time with fancy, unnecessary pages.

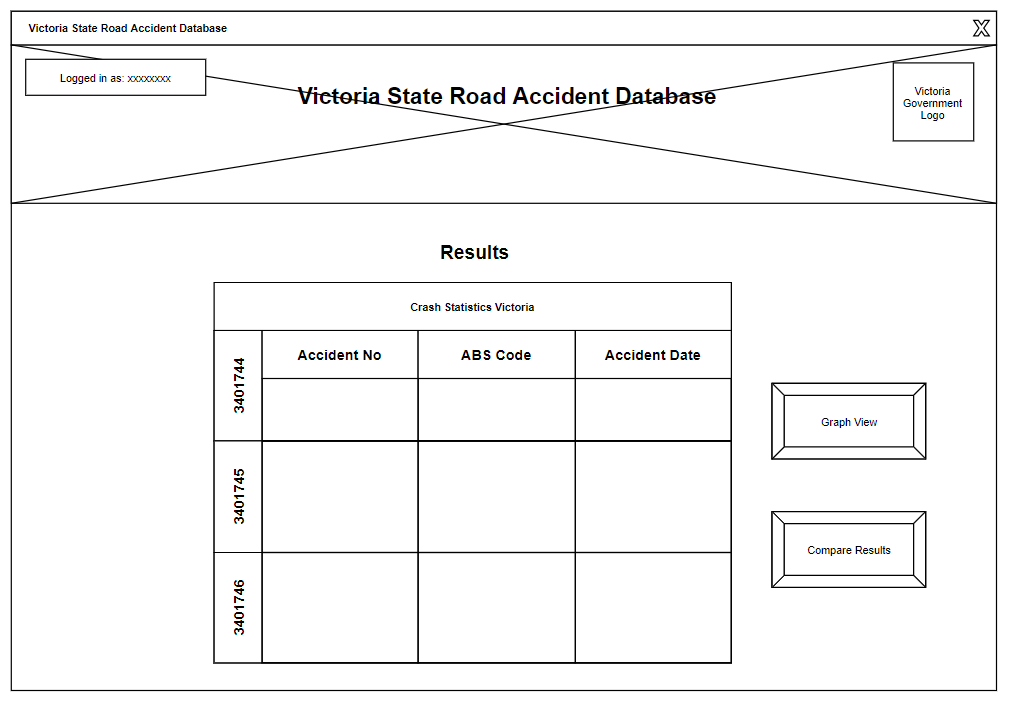
## Visual Design

The initial visual design of the system has been detailed in the diagrams below. These wireframes display a minimalistic presentation of how each page’s user interface will be presented. They focus on showing the layout and location of each visual element.

**Main Menu**

**Access Raw Dataset**

**Search Dataset**

**Search Results**

These design choices were made based on the simplicity of their use. From the main menu, large buttons instantly present the choices the user can make. There is no distracting information taking away from the purpose of the program, other than an image header underneath the title adding some style to the pages. The table of data is a rough representation of how it will be displayed. It will contain many more fields, records and detail in the final program.