**SOFTWARE DESIGN DOCUMENT**

**Victoria State Accident Dashboard**

**Submitted by:**

**GROUP 33**

Augustine Kim | s5125270

Kai Chun Yu | s5309764

Ann Tapiador | s5314834

Table of Contents

[1.0 System Vision 3](#_Toc144667149)

[1.1 Problem Background 3](#_Toc144667150)

[1.2 System Overview 3](#_Toc144667151)

[1.3 Potential Benefits 3](#_Toc144667152)

[2.0 Requirements 4](#_Toc144667153)

[2.1 User Requirements 4](#_Toc144667154)

[2.2 Software Requirements 4](#_Toc144667155)

[2.3 Use Cases & Use Case Diagrams 5](#_Toc144667156)

[3.0 Software Design and System Components 8](#_Toc144667157)

[3.1 Software Design 8](#_Toc144667158)

[3.2 System Components 9](#_Toc144667159)

[3.2.1 Functions 9](#_Toc144667160)

[3.2.2 Data Structures / Data Sources 10](#_Toc144667161)

[3.2.3 Detailed Design 14](#_Toc144667162)

[4.0 User Interface Design 15](#_Toc144667163)

[4.1 Structural Design 15](#_Toc144667164)

[4.2 Visual Design 17](#_Toc144667165)

[REFERENCES 19](#_Toc144667166)

# 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).

Data on road accidents between July 1, 2013 to March 21, 2019 within the state of Victoria is available. The record includes other details such as time, location, severity, road geometry, and type of road users to name a few. 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 and organised in tables and charts in a way that will aid in data analysis. The system should be able flexible enough to show status and trends based on user-selected parameters (e.g., by period or by type of accident etc.). This will allow the Victorian 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:

1. Information of all accidents based on a user-selected period.
2. A chart showing the average number of accidents in each hour of the day based on a user-selected period.
3. 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.
4. A chart that shows the impact of alcohol in accidents such as trends over time and accident types involving alcohol.
5. A chart showing the number of accidents among road users, i.e., bicyclists, passengers, drivers, pedestrians, pillions, and motorists, based on user-selected period.

## Potential Benefits

The following are the primary benefits of the system:

* Automated processing and visualisation of data for analysis
* Tool to measure the performance of the government and its policies on improving road safety.
* Serve as basis for policy improvements.
* Aid in lowering incidences in accident-prone areas.
* Lower social and economic costs brought about by road crashes.

# Requirements

## User Requirements

The users of the system are primarily employees of the Victorian government (e.g. Transport Accident Commission, Victoria Police, Vic Roads etc.). The user interface will be in a form of a dashboard, which will display the user manual to serve as a guide on how to navigate through and utilise the system. The user can then select a specific period (start date and end date), category (e.g. accident type), and the accidents’ relation to alcohol (alcohol-related, not alcohol-related, both). After selecting the filters, the user can click the search button to retrieve the data and display in a line graph, a bar graph, a pie chart, and a table, that the user can use for analysis. When the user clicks the Victorian government logo, it takes them back to the initial view which displays the user manual.

* Access a user manual that describes how to use the system.
* Select a period range (e.g., start date and end date) for viewing a specific time frame.
* Select categories for viewing specific information such as accident types.
* Select alcohol-related cases to identify the impact of alcohol in accidents.
* View a table which displays all information on accidents that match the selected period, category, or keyword.
* View a graph (line graph) that can show the average number of accidents in each hour of the day based on the selected period, category, and whether the accident is alcohol-related or not.
* View graphs (bar graph and pie chart) that show the total number of accidents based on selected period, category and, whether the accident is alcohol-related, not alcohol-related, or both.

## Software Requirements

Below are details of the software requirements. These include functional requirements, i.e., those that are part of the primary requirements noted on the System Overview (see system vision), as well as non-functional requirements, which pertains to system characteristics and performance goals.

| **Requirements** | **Details** |
| --- | --- |
| **Functional** | * The program shall have two filters for the user to select a start day and an end date. * The program shall have a filter that accepts typing for the user to select the accident category. * The program shall have a filter for users to select alcohol-related accidents. * The program shall contain a search button for the user to click and execute the search action. * The program shall remind the user start date and end date must be filled in order to commence search. * After performing a search, the program shall display all accident information that matches the selected filters. * After performing a search, the program shall contain a line chart to show the average number of accidents in each hour of the day based on the selected period. * After performing a search, the program shall contain a bar chart and pie chart to show the accident number based on the selection of category and alcohol-related filter. * Once a selection has been made on the accident category filter, the program shall only display the information that is related to the selected category. * Once a selection has been made on the alcohol-related filter, the program shall only display the information that based on the selection. |
| **Usability** | * The program shall show a Victoria State logo for users to recognise the authenticity of the website on the homepage. * The program shall contain a system use guide to describe how to use the function. * The system shall be able to access from anytime and anywhere using a URL. |
| **Reliability** | * Since this will be web-based, a refresh button can be used in case of system lags. |
| **Performance** | * The system shall be able to perform the search result and display the output (data and diagrams) within 15 seconds. |
| **Security** | * The user will only be able to view the data but will not be allowed to make modification on the database. We can use https to secure the site. |
| **Implementation** | * The program shall use python as the main programming language |
| **Physical requirements** | * The system shall be able to access from any operation systems including MacOS, Windows, Linux, etc. |

## Use Cases & Use Case Diagrams

Below are the 5 use cases based on the identified functional requirements:

|  |  |
| --- | --- |
| **Use Case ID** | **1** |
| **Use Case Name** | View all information of accident. |
| **Actors** | User |
| **Description** | The user will be able to view all relevant accident information that happened on the selected period. |
| **Flow of**  **Events** | 1. The user will access the system through a dashboard. 2. A user manual will guide the user on how to filter information. 3. The user will select the start date and end date. 4. The user clicks the search button. 5. The system will display a table with all relevant data from the selected period. |
| **Alternate Flow** | None |

|  |  |
| --- | --- |
| **Use Case ID** | **2** |
| **Use Case Name** | View average number of accidents in each hour of the day. |
| **Actors** | User |
| **Description** | The user will be able to view a line graph that shows the average number of accidents in each hour of the day. |
| **Flow of**  **Events** | 1. The user will access to the system through a dashboard. 2. A user manual will guide the user on how to filter information. 3. The user will select the start date and end date. 4. The user will click the search button. 5. The system will display the data on a line graph. |
| **Alternate Flow** | None |

|  |  |
| --- | --- |
| **Use Case ID** | **3** |
| **Use Case Name** | Search based on keyword. |
| **Actors** | User |
| **Description** | The user will be able to view a table containing relevant information on all accidents with an accident type keyword entered by the user (e.g., collision, pedestrian), based on a user-selected period. |
| **Flow of**  **Events** | 1. The user will access to the system through a dashboard. 2. A user manual will guide the user on how to filter information. 3. The user will select the start date and end date. 4. The user will type a keyword in the accident type category. 5. The system will display data on a table. |
| **Alternate Flow** | None |

|  |  |
| --- | --- |
| **Use Case ID** | **4** |
| **Use Case Name** | View alcohol-related against non alcohol-related data. |
| **Actors** | User |
| **Description** | The user will be able to view a bar graph and a pie chart that will display the impact of alcohol, using a filter with options on whether only alcohol related data are displayed, or non alcohol-related data, or both. |
| **Flow of**  **Events** | 1. The user will access to the system through a dashboard. 2. A user manual will guide the user on how to filter information. 3. The user will select the start date and end date. 4. The user will type/select the accident category. 5. The user will select Yes, No, or both on the alcohol-related filter. 6. The system will display results on at least a bar and a pie chart. |
| **Alternate Flow** | None |

|  |  |
| --- | --- |
| **Use Case ID** | **5** |
| **Use Case Name** | View road user chart. |
| **Actors** | User |
| **Description** | The user will be able to view a bar graph and a pie chart that will show the number of accidents by road user. |
| **Flow of**  **Events** | 1. The user will access to the system through a dashboard. 2. A user manual will guide the user on how to filter information. 3. The user will select the start date and end date. 4. The user will select the accident category “road users” on the filter. 5. The system will display results on at least a bar and a pie chart. |
| **Alternate Flow** | None |

Below is the use case diagram that describes the functions noted on the use cases above:

A diagram of a crash data software

Description automatically generated

# Software Design and System Components

## Software Design

The flow chart below illustrates how the software interacts with the user.

A diagram of a process flow

Description automatically generated

## System Components

### Functions

| **FUNCTIONS** | **DETAILS** | |
| --- | --- | --- |
| **Load Data Function (Search)** | Description | This function will enable the system to load data from the original database which is the excel file. |
| Input parameters | Start date (date), end date (date), accident category (string), alcohol-relation (string). These parameters are used for identifying which data are going to load from the database to system. |
| Side effects | None |
| Return Value | Array that store all data sorted by the input parameters. |
| **Table display function** | Description | This display function will enable the system to show the result of the search in a table from for user to view the accident records. |
| Input parameters | The result of load data function (array). The parameter is used for inserting what data need to be displayed in the table. |
| Side effects | None |
| Return Value | This function should end up return a table that contain all data filtered by the search. |
| **Line chart display function** | Description | The function will display the number of accidents by selected categories. |
| Input parameters | Start date (date), end date (date). These parameters are used for inserting what period of data need to be displayed in the line chart. |
| Side effects | None |
| Return Value | This function should end up return a line chart that showing the average number of the accidents (Y Axis) by each hour in the day (X Axis). |
| **Bar chart display function** | Description | The function will display average number of accidents by selected category in a bar chart view by user. |
| Input parameters | start date (date), end date (date), accident category (string), alcohol-relation (string). These parameters are used for identifying which data are going to display in the bar chart. |
| Side effects | None |
| Return Value | This function should end up return a bar chart that showing the number of the accidents (Y Axis) by different categories (X Axis). |
| **Pie chart display function** | Description | The function will display a pie chart that present the number of accidents by selected categories in a percentage form. |
| Input parameters | start date (date), end date (date), accident category (string), alcohol-relation (string). These parameters are used for identifying which data are going to display in the pie chart. |
| Side effects | None |
| Return Value | This function should end up return a pie chart that showing the number of the accidents by different categories in percentage. |

### Data Structures / Data Sources

The data will be coming from an excel or csv file, arranged in a table contained within one worksheet. The raw data has 63 columns representing attributes while the rows represent records. However, columns that are relevant to the 5 functional requirements will be prioritised as shown below:

|  | **REQUIREMENT** | **INPUT** | **EXPECTED OUTPUT** |
| --- | --- | --- | --- |
| 1 | For a user-selected period, display the information of all accidents that happened in the period. | * A filter that will allow users to choose start date and end date using the attribute, ACCIDENT\_DATE | * A table showing all records within the specified period. |
| 2 | For a user-selected period, produce a chart to show the number of accidents in each hour of the day (on average). | * A filter that will allow users to choose start date and end date using the attribute, ACCIDENT\_DATE * Average count of accidents per hour using attributes ACCIDENT\_NO and ACCIDENT\_TIME. Since the raw data for ACCIDENT\_TIME is presented in a 24-hr format in minute intervals, operations should be applied to get the average per hour. | * A line graph showing the hours of the day on the x-axis and the average number of accidents for the given period on the y-axis |
| 3 | For a user-selected period, retrieve all accidents caused by an accident type that contains a keyword (user entered), e.g. collision, pedestrian. | * A filter that will allow users to choose start date and end date using attribute, ACCIDENT\_DATE * A key word from any of the data values under the attribute ACCIDENT\_TYPE that will be typed in a search bar, i.e.: * Collision with a fixed object * Collision with some other object * Collision with vehicle * Fall from or in moving vehicle * No collision and no object struck * Other accident * Struck animal * Struck Pedestrian * Vehicle overturned (no collision) | * A table showing all the records within the specified period that contains the keyword |
| 4 | Allow the user to analyse the impact of alcohol in accidents – i.e., trends over time, accident types involving alcohol, etc. | * A filter that will allow users to choose start date and end date using the attribute, ACCIDENT\_DATE * A filter that will allow users to show the number of accidents using the attribute ACCIDENT\_NO, that are ALCOHOL\_RELATED. * A filter that will allow the members to choose one of the following attributes to be included in the analysis: * DAY\_OF\_WEEK * HIT\_RUN\_FLAG * POLICE\_ATTEND * LIGHT\_CONDITION * SPEED\_ZONE * ROAD\_GEOMETRY * RMA * ACCIDENT\_TYPE * RUN\_OFF\_ROAD * SEVERITY * DEG\_URBAN\_NAME * STAT\_DIV\_NAME * REGION\_NAME * A filter that will allow the user to choose specific data values from the chosen attribute to be included in the analysis (see next table) | * A bar graph showing the chosen attribute’s values on the x-axis, and the number of alcohol-related accidents on the y-axis, based on the given period. * A pie chart showing the number of alcohol-related accidents across the chosen attribute’s values, together with corresponding percentages. |
| 5 | For a user-selected period, produce a chart to show the number of accidents by road user, i.e., bicyclist, passenger, driver, pedestrian, pillion, motorist, unknown. | * A filter that will allow users to choose start date and end date using the attribute, ACCIDENT\_DATE * Count of accidents using the attribute ACCIDENT\_NO, based on the following attributes (road users): * BICYCLIST * PASSENGER * DRIVER * PEDESTRIAN * PILLION * MOTORIST * UNKNOWN | * A bar graph showing the attributes on the x-axis, and the number of accidents on the y-axis, based on the given period. * A pie chart showing the number of accidents for each road user, together with corresponding percentages |

To further clarify, only 25 out of the 63 columns will be utilised, which should cover the functional requirements, as well, additional functions that may be useful to the analysis:

| **DATA MEMBER / ATTRIBUTES / COLUMN HEADERS** | **DATA VALUE** |
| --- | --- |
| ACCIDENT\_NO | This acts as the primary key of the table, thereby making its value unique for each record. The string is composed of 12 alphanumeric characters, which starts with a capital T, followed by the year of the accident, and a 7-digit number, e.g., T20130013732 |
| ACCIDENT\_DATE | Format of D/MM/YYYY starting from 1/07/2013 to 21/03/2019 |
| ACCIDENT\_TIME | The values are presented in minute intervals within a 24-hour period. It is a string with the format of hh.mm.ss, i.e., 00.00.00, 00.01.00, 00.002.00 and so on until 23.59.00 |
| DAY\_OF\_WEEK | String of each day of the week:  Monday  Tuesday  Wednesday  Thursday  Friday  Saturday  Sunday |
| ALCOHOL\_RELATED | String of either **No** or **Yes** |
| HIT\_RUN\_FLAG | String of either **No**, **Not known**, or **Yes** |
| POLICE\_ATTEND | String of either **No** or **Yes** |
| LIGHT\_CONDITION | String of either one of the following:  Dark No street lights  Dark Street lights off  Dark Street lights on  Dark Street lights unknown  Day  Dusk/Dawn  Unk. |
| SPEED\_ZONE | String of either one of the following:  30km/hr  40 km/hr  50 km/hr  60 km/hr  70 km/hr  75 km/hr  80 km/hr  90 km/hr  110 km/hr  100 km/hr  Camping grounds or off road  Not known  Other speed limit |
| ROAD\_GEOMETRY | String of either one of the following:  Cross intersection  Dead end  Multiple intersection  Not at intersection  Private property  Road closure  T intersection  Unknown  Y intersection |
| RMA | String of either one of the following:  Arterial Highway  Arterial Other  Freeway  Local Road  Non Arterial |
| ACCIDENT\_TYPE | String of either one of the following:  Collision with a fixed object  collision with some other object  Collision with vehicle  Fall from or in moving vehicle  No collision and no object struck  Other accident  Struck animal  Struck Pedestrian  Vehicle overturned (no collision) |
| SEVERITY | String of either one of the following:  Fatal accident  Non injury accident  Other injury accident  Serious injury accident |
| BICYCLIST | Integer showing the number of people classified as the said attribute, who were involved in the accident. |
| PASSENGER |
| DRIVER |
| PEDESTRIAN |
| PILLION |
| MOTORIST |
| UNKNOWN |
| LONGITUDE | Positive float data type that are represented in the x-axis, e.g., 144.9698 |
| LATITUDE | Negative float data type that are represented on the y-axis, e.g., -37.82202 |
| DEG\_URBAN\_NAME | String of either one of the following:  LARGE\_PROVINCIAL\_CITIES  MELB\_URBAN  MELBOURNE\_CBD  RURAL\_VICTORIA  SMALL\_CITIES  SMALL\_TOWNS  TOWNS |
| STAT\_DIV\_NAME | String of either **Country** or **Metro** |
| REGION\_NAME | String of either one of the following:  EASTERN REGION  METROPOLITAN NORTH WEST REGION  METROPOLITAN SOUTH EAST REGION  NORTH EASTERN REGION  NORTHERN REGION  SOUTH WESTERN REGION  WESTERN REGION |

### Detailed Design

Below is the pseudocode for the system using Python:

1. Import wx to for the graphical user interface (GUI) and widgets
2. Import pandas to be used for computation
3. Import matplotlib for plotting graphs (line, bar, pie)
4. Import wx.grid to be used for the table that will be included at the bottom of the output
5. Import the .csv file and convert to a pandas dataframe
6. Convert the values on the ACCIDENT\_TIME column to a time format
7. Create a dictionary to map the categories and corresponding values
8. Create a class for the wx frame and define attributes
   1. Widget containers
   2. Canvas where the charts will be placed
   3. Widgets for filters such as date pickers (wx.DatePickerCtrl), choice and text field widgets for the categories (wx.choice and wx.TextCtrl), widget that will allow users to choose alcohol-related, non alcohol-related or both (wx.RadioBox)
   4. Widget for the seach button (wx.Button)
   5. Scrollable window for the table (wx.ScrolledWindow)
9. Define the method for creating a line graph
   1. Assign hours of the day to the x-axis
   2. Compute for the average count of accidents per hour of the day and plot on the y-axis based on parameters selected on the filters
10. Define the method for creating the bar graph
    1. Assign category values on the x-axis
    2. Assign the count of accidents on the y-axis based on parameters selected on filters
11. Define the method for creating the table
    1. Show relevant column headers on table
    2. Show records based on parameters selected on filters
12. Display on GUI

# User Interface Design

For the interface design, wireframes were initially developed and later, Pixso was used for creating the final UI mock-up. User research underscored the importance of a user-friendly, easy-to-navigate interface, which guided our design decisions.

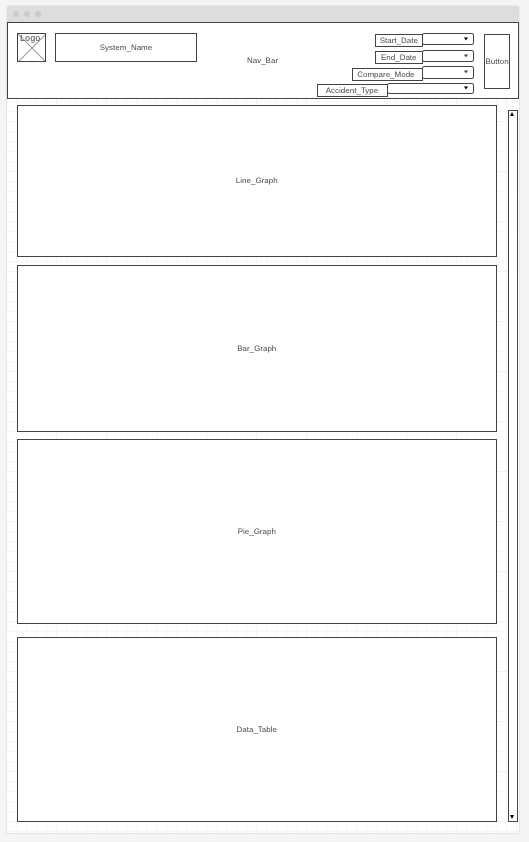
## Structural Design

The interface is organized into several key sections:

* **Header**: Contains the system name and logo, with date selectors aligned to the right.
* **Search Bar & Dropdowns**: Below the header, users can set specific conditions and select desired search results via dropdown menus.
* **First Graph**: Positioned further down to provide an initial data overview.
* **Second Graph**: Offers a more detailed data analysis.
* **Table**: Situated at the bottom for granular data representation.

A vertical scroll is implemented to accommodate smaller screens or situations where a vertical layout is more practical.

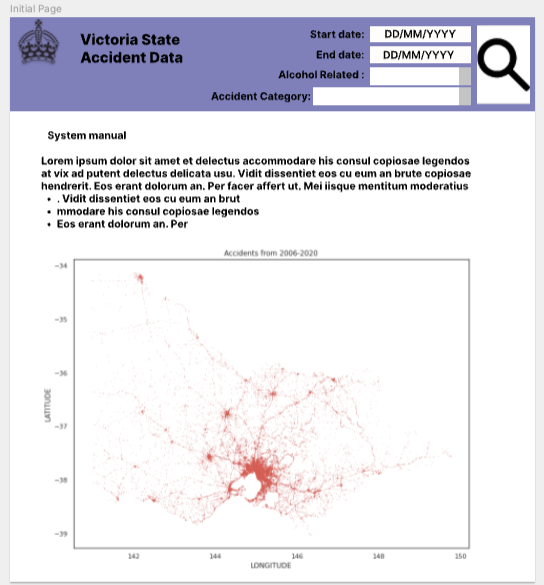
The header was designed for instant recognition, housing the system name and logo. Dropdowns are below the header to facilitate condition-based searching. Graphs and tables follow in a logical flow to present data from a general overview to specific details. The vertical scroll option was added to ensure user-friendliness regardless of screen size.

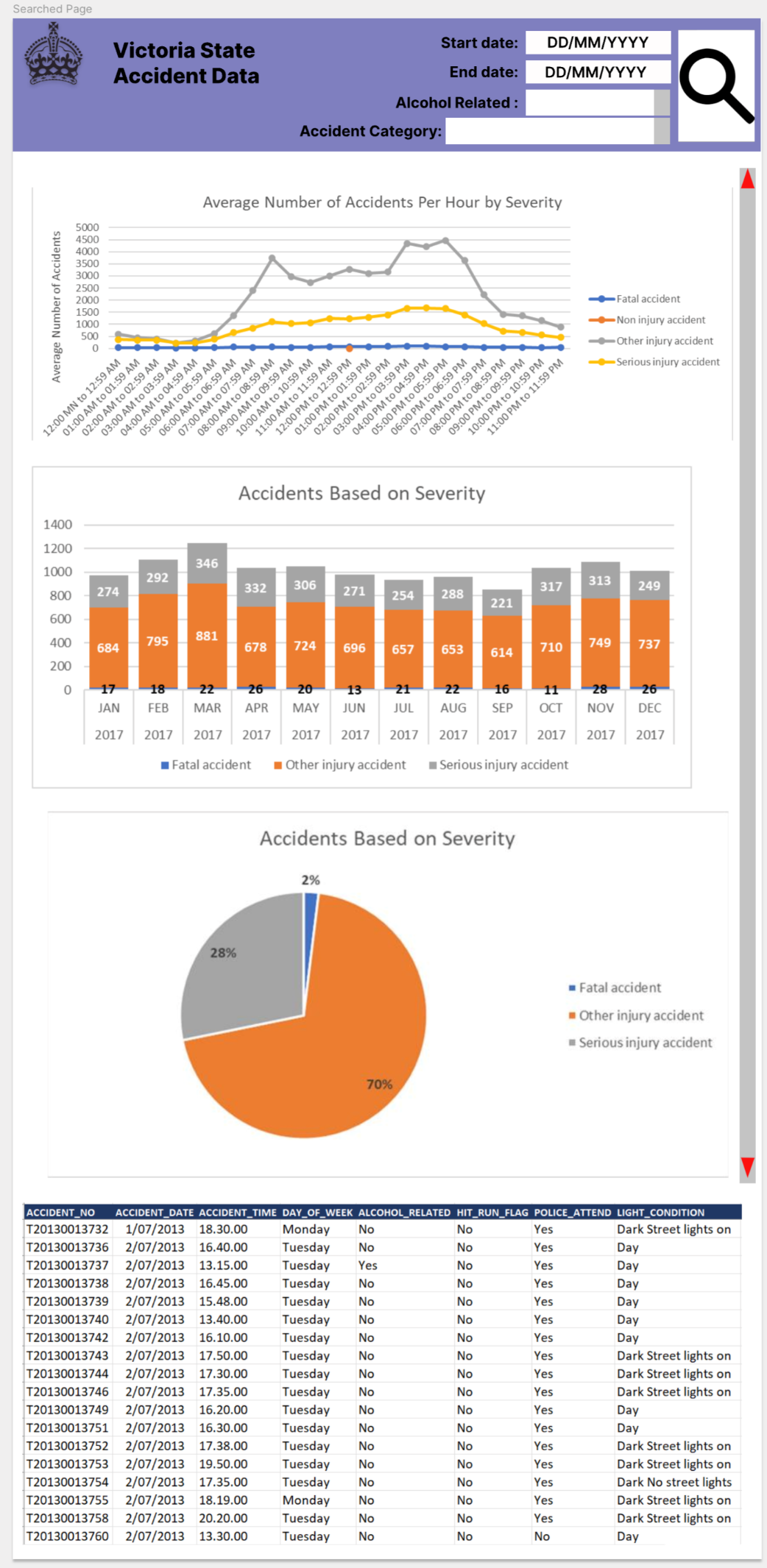


## Visual Design

The dynamic layout was chosen for adaptability, ensuring an optimal user experience irrespective of device dimensions. Minimalistic shadows add depth without causing distractions. The government's signature color in the header was incorporated to build a sense of trust and familiarity with the users.

* **Layout**: The layout is dynamic, able to extend vertically or adapt via a scrollbar, depending on the user’s screen size.
* **Visual Elements**: Minimalistic shadows are used to elevate elements.
* **Icons & Graphics**: Simple, easy-to-understand icons are employed.
* **Style**: The design maintains a clean and modern aesthetic.
* **Color**: The header utilizes the signature color of the relevant government body to evoke familiarity and trust. The rest of the design sticks to neutral shades.
* **Fonts**: Sans-serif fonts are chosen for readability and modernity.





# 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>