**SOFTWARE DESIGN DOCUMENT**

**Victoria State Accident Database**

**Submitted by:**

**GROUP 33**

Augustine Kim | s5125270

Kai Chun Yu | s5309764

Ann Tapiador | s5314834

Table of Contents

[1.0 System Vision 3](#_Toc144641063)

[1.1 Problem Background 3](#_Toc144641064)

[1.2 System Overview 3](#_Toc144641065)

[1.3 Potential Benefits 3](#_Toc144641066)

[2.0 Requirements 4](#_Toc144641067)

[2.1 User Requirements 4](#_Toc144641068)

[2.2 Software Requirements 4](#_Toc144641069)

[2.3 Use Cases & Use Case Diagrams 5](#_Toc144641070)

[3.0 Software Design and System Components 7](#_Toc144641071)

[3.1 Software Design 7](#_Toc144641072)

[3.2 System Components 8](#_Toc144641073)

[3.2.1 Functions 8](#_Toc144641074)

[3.2.2 Data Structures / Data Sources 10](#_Toc144641075)

[3.2.3 Detailed Design 15](#_Toc144641076)

[4.0 User Interface Design 15](#_Toc144641077)

[4.1 Structural Design 15](#_Toc144641078)

[4.2 Visual Design 17](#_Toc144641079)

[REFERENCES 19](#_Toc144641080)

# 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

This section details how the user interacts with the system:

* A user guide that describes how to use the system.
* A selection of a period (eg. Start date and end date) for viewing specific data in the period.
* A selection of accident categories for viewing specific information in selected accident types.
* A selection of alcohol-related cases to identify the influences of alcohol in accidents.
* All information on accidents that match the selected period and category.
* A diagram (line chart) that can show the average number of accidents in each hour of the day based on the selected period, category, and relationship of alcohol.
* Some diagrams (bar chart, pie chart) that show the number of accidents based on selected categories and, the relationship of alcohol.

Once the user accesses the user interface dashboard of the proposed system, the user will be able to view a user guide on how to use the system. Then, the user can use the filter to select a specific period and category to retrieve the accident data and diagram based on their needs. In addition, a filter for selecting alcohol-related cases will be available for users to identify the influences of alcohol. After selection, the user can click the search button to retrieve accident data and analyse diagrams based on the selected requirement.

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

Below are details of the software requirements:

* 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 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 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.

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

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 accident information from a selected period. |
| **Flow of**  **Events** | 1. The user will access to the system user interface. 2. A system user guide will tell the user to select a period for search. 3. The user will select the start date and end date. 4. The system will display a table with all relevant data between the period on the screen. |
| **Alternate Flow** | None |

|  |  |
| --- | --- |
| **Use Case ID** | **2** |
| **Use Case Name** | View average number of accidents in each hour chart. |
| **Actors** | User |
| **Description** | The user will be able to view a line chart that show the average number of accidents in each hour of the day. |
| **Flow of**  **Events** | 1. The user will access to the system user interface. 2. A system use guide will tell the user to select a period for search. 3. The user will select the start date and end date. 4. The system will display a line chart on the screen. |
| **Alternate Flow** | None |

|  |  |
| --- | --- |
| **Use Case ID** | **3** |
| **Use Case Name** | Search by accident category. |
| **Actors** | User |
| **Description** | The user will be able to search by accident category and view the accident information based on the selected period and accident category. |
| **Flow of**  **Events** | 1. The user will access to the system user interface. 2. A system use guide will tell the user to select a period and accident category for search. 3. The user will select the start date and end date. 4. The user will type/select the accident category. 5. The system will display a table that contains relevant record on the screen. |
| **Alternate Flow** | None |

|  |  |
| --- | --- |
| **Use Case ID** | **4** |
| **Use Case Name** | View alcohol analyses. |
| **Actors** | User |
| **Description** | The user will be able to use an alcohol related filter to sort the output diagrams and table. |
| **Flow of**  **Events** | 1. The user will access to the system user interface. 2. A system use guide will tell the user to select a period, accident category, and alcohol-related filter for search. 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 on alcohol-related filter. 6. The system will display the table and diagrams on the screen. |
| **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 chart that show the number of harm case to each type of road user. |
| **Flow of**  **Events** | 1. The user will access to the system user interface. 2. A system use guide will tell the user to select a period, accident category for search. 3. The user will select the start date and end date. 4. The user will select the road users on the accident category filter. 5. The system will display a bar chart on the screen. |
| **Alternate Flow** | None |

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

A diagram of a problem

Description automatically generated with medium confidence

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

| **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. |

* 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.

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

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 |

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

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.

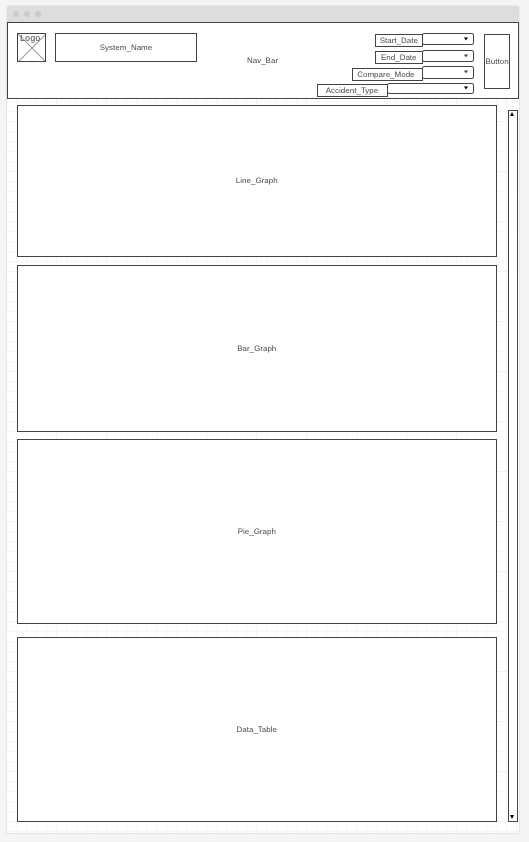
This is your initial interface design. Describe the tools you used for this design stage and any key findings that informed your design. This introduction is descriptive and should explain what you have completed for the actual design work you will present in the sub-sections below.

## Structural Design

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.



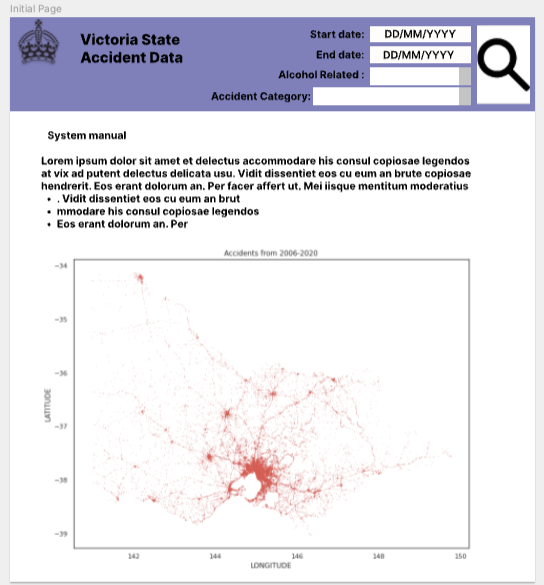
The header was designed for instant recognition, housing the system name and logo. Dropdowns are below the header to facilitate condition-based searching. Graphs and tables follow in a logical flow to present data from a general overview to specific details. The vertical scroll option was added to ensure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

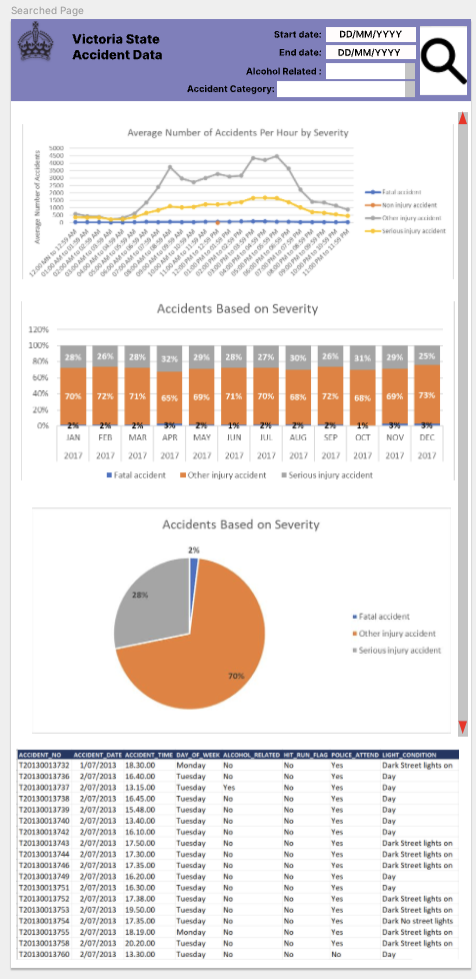
Structural design refers to the navigational and information structure of your product – the structure that supports the interface layout. How will you structure your product? How will you group your information? How will you navigate through your product? Why? This can take the form of a diagram showing 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

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.





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>