Software Design Document   
Victoria Accident Data Visualisation Project

Student Names

* Max Woollons s5349356
* Kail Terepai s5307474
* Reiss Tooze s5046939

**Table of Contents**

[1.0 System Vision 3](#_gjdgxs)

[1.1 Problem Background 3](#_30j0zll)

[1.2 System Overview 3](#_1fob9te)

[1.3 Potential Benefits 3](#_3znysh7)

[2.0 Requirements 4](#_2et92p0)

[2.1 User Requirements 4](#_tyjcwt)

2.2 Software Requirements 4

[2.3 Use Cases & Use Case Diagrams 5](#_1t3h5sf)

[3.0 Software Design and System Components 8](#_4d34og8)

3.1 Software Design 8

[3.2 System Components 9](#_46ve2j1vi9zi)

3.1.1 Functions 9

3.1.2 Data Structures / Data Sources 11

3.1.3 Detailed Design 12

[4.0 User Interface Design 16](#_2s8eyo1)

4.1 Structural Design 16

4.2 Visual Design 17

# System Vision

## Problem Background

A visualisation tool needs to be developed to best communicate data from the Victoria State Accident Dataset provided by Vicroads (Australia). This data contains information from 74,908 road-based accidents that happened on Victorian roads between 2013 and 2019. The dataset contains information about the date, time, location, type of accident, a short description of what happened, type of vehicle, number of vehicles, alcohol involvement, if it was a hit and run, speed limits on the road and many other relevant details. A visualisation tool needs to be created to better communicate insights from the data to relevant stakeholders.

## System Overview

The system itself will be built using Python and utilise Python libraries: pandas, matplotlib, seaborn, and tkinter. The application will read a dataset, and then show various visualisation options to a user using a GUI. Relevant data selection options will be displayed from which the user will then be able to make their selection from which the relevant data will be visualised for user interpretation.

Users will be able to select visualisation options which include:

* Display the information of all accidents that happened in a user-selected period.
* For a user-selected period, produce a chart to show the number of accidents in each hour of the day (on average).
* For a user-selected period, retrieve all accidents caused by an accident type that contains a keyword (user entered), e.g., collision, pedestrian.
* Allow the user to analyse the impact of alcohol in accidents – i.e., trends over time, accident types involving alcohol, etc.
* For a user-selected year, display the information of all accidents that occurred on a Victorian public holiday within the selected year.

## Potential Benefits

This tool will allow for the effective interpretation and filtering of relevant data from the Victoria State Accident Dataset. This dataset is large, not very end-user friendly and does not effectively show relevant data that could be useful for the user. Potential benefits for end-users include:

* Effective filtering and visualisation of accidents across date ranges.
* Effective filtering and visualisation of accidents across each hour of the day.
* Effective keyword searching for relevant accident keywords to produce visualisation outcomes.
* Effective visualisation of alcohol-related accidents.
* Effective visualisation of accidents on public holidays based on year.

# Requirements

## User Requirements

Users will be able to execute the software, input and or select relevant query parameters, then based on selection/input will be shown a visualisation of the data from the Victoria State Accident Dataset.

A user may be a relevant public servant, politician or member of the public with an interest in the data. Users will need to be able to access the program through an executable file. The user will then be presented with options for the data visualisation. These options will provide the user with the ability to:

* Specify a time period in the software and be shown data relating to the specified time period.
* Specify a time period in the software and be shown the average amount of accidents across a day at hourly intervals.
* Enter keywords relating to the accident and for the software to search and show data relating to the entered keyword(s).
* Clearly see the impact that alcohol has in Victorian accidents through visualisation.
* Select a year and view relevant data relating to accidents on public holidays.

With each requirement, users should have the ability to export the generated data for external use.

## Software Requirements

The software has a few core requirements for it to be effective for use. It needs to read in a csv data source, filter the data and then show the data in the form of charts to the users. Further requirements are below.

* R1 The program shall accept a csv document for data interpretation.
* R1.1 The program shall filter data from the CSV document including:
  + Date ranges
  + Times of day
  + Keywords
  + Alcohol involvement
  + Accidents involving public holidays
* R2.2 The program shall create charts based on data.
  + These charts are based on mostly filtered data and include but are not limited to bar charts, pie charts, and plots which are to be shown in a GUI.
* R2.3 The program shall allow the exporting of generated charts and other visualisations.
  + These exported files will be images.

## 

## Use Cases & Use Case Diagrams

Use Cases:

Table 1 was created to show all the use cases for the Victoria Accident Data Project. The table shows the use cases, users and description of the use case.

|  |  |
| --- | --- |
| **Use Cases with Descriptions: Victoria Accident Data Visualisation Project** | |
| Use Case | Description |
| View visualisation relating to crashes over a user-specified time period. | User opens the application (if not already open), locates the ‘Visualise crash data over time’ option, enters any date range they wish to view, and then selects the visualise button. Users are then shown the visualisation. |
| View visualisation relating to the average number of accidents in each hour of the day over a user-specified time period. | User opens the application (if not already open), locates the ‘Average number of accidents over the day’ option, enters any date range if they wish and then selects the visualise button. Users are then shown the visualisation. |
| View visualisation of crash data to a specified keyword/s. | User opens the application (if not already open), locates the ‘Crash data based on keyword/s’ option, enters keywords in the field,then selects the Visualise button. Users are then shown the visualisation. |
| View visualisation relating to the impact of alcohol in the accidents. | User opens the application (if not already open) and then selects the Visualise button on the ‘Alcohol’s impact on crashes’ option. Users are then shown a visualisation comparing crashes involving alcohol vs. not involving alcohol. |
| View visualisation relating to crashes on public holidays over a period of time. | User opens the application (if not already open), locates the ‘Public Holiday insights’ option and selects the visualise button. Users are then shown the visualisation. |

Table 1: Use case descriptions

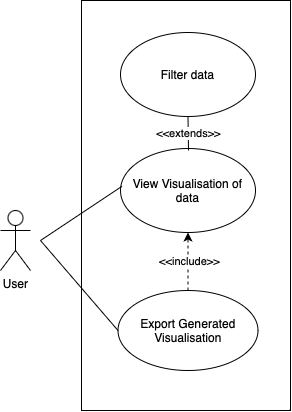
Use Case Diagrams:

Figure 1 shows the use case diagram for use case one & two. Although both use cases operate similarly, they produce different data for users’ interpretation.

* View visualisation relating to crashes over a user-specified period of time.
* View visualisation relating to the average number of accidents in each hour of the day over a user-specified period of time.

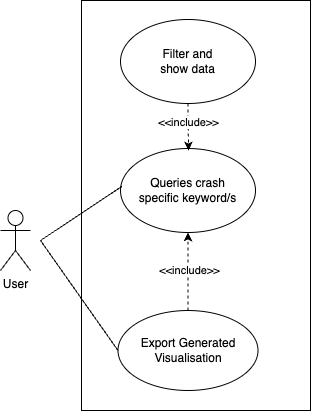


Figure 1 - Use case one and two diagram

Figure 2 shows the use case diagram for use case three ‘View visualisation of crash data to a specified keyword/s.’

Figure 2 - Use case three diagram

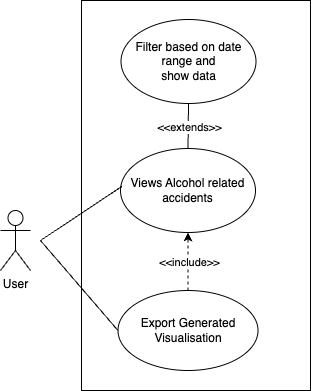
Figure 3 shows use case four diagram ‘View visualisation relating to the impact of alcohol in the accidents.’

Figure 3 - Use Case 4 diagram

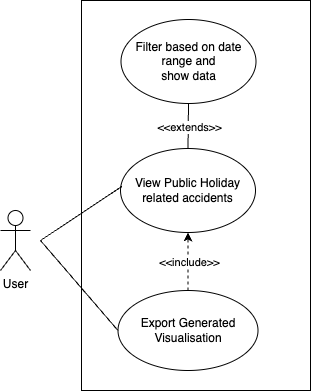


Figure 4 shows use case five diagram ‘View visualisation relating to crashes on public holidays over a period of time.’

# 

Figure 4 - Use Case 5 Diagram

# Software Design and System Components

## Software Design

Figure 5 - Flowchart of Software

## 3.2 System Components

### Functions

Below is a preliminary list of the functions in the software. Each function has a description of its functionality in the software.

* Load and filter data
  + This function loads the data within the CSV file and allows it for use in the program. The function will also clean and filter the data that it loads.
  + Start and end date parameters will be used as inputs for the functions. Strings will be accepted.
  + The function will return the loaded and filtered data for use in visualisations.
* Word and data filter function.
  + This function takes the original data and converts it into a usable data structure for matplotlib or seaborn.
  + This function will accept a string of words.
  + The function will filter the data and return search results matching the string input.
* show\_visualisation1
  + This function will be executed when a user clicks on the first visualisation option. The visualisation shown will relate to crashes over a user-specified period of time. Data to be visualised includes:
    - Total number of crashes, month by month (bar chart)
    - Most common crash type (pie chart)
    - Deadliest days to drive (bar chart)
    - Most common daylight condition to crash. (pie chart)
  + The function will have two input parameters, a start and an end date. These input parameters will be used to filter the data being presented.
  + The function will show a new GUI window with dashboard-style visualisations of the data.
* show\_visualisation2
  + This function will be executed when the user clicks the second visualisation option. The visualisation shown will be relating to the number of accidents in each hour of the day over a user-specified period of time. This data will be visualised using a bar chart.
  + This function will also have two input parameters, a start and an end date. These input parameters will be used to filter the data presented.
  + The function will show the chart in a new GUI window.
* show\_visualisation3
  + This function will be executed when the user clicks the third visualisation option. The visualisation will be relating to crash data which includes a specified keyword/s. Data to be visualised includes:
    - Number of crashes with keyword/s (bar chart)
    - Speed Zones where crashed occurred (bar chart)
    - Number of fatalities and injuries. (bar chart)
    - Number of alcohol to non-alcohol related crashes (pie chart)
  + This function will have one input parameter, a string which contains the words to search the data for.
  + The function will show a new GUI window with dashboard style visualisations of the data.
* show\_visualisation4
  + This function will be executed when the user clicks the fourth visualisation option. The visualisation will be relating to the impact of alcohol in the accidents. Data to be visualised includes:
    - Accidents relating to alcohol vs accidents non-alcohol related (pie chart)
    - Top LGA’s with the most alcohol related crashes (Bar Chart)
    - Average number of fatalities in accidents alcohol vs non-alcohol (pie chart)
    - Pedestrians struck alcohol vs not alcohol (plot).
  + This function has no input parameters.
  + The function will show a new GUI window with dashboard style visualisations of the data.
* show\_visualisation5
  + This function will be executed when the user clicks the fifth visualisation option. The visualisation shown will be relating to crashes on public holidays over a period of time. Data to be visualised includes:
    - Number of Accidents on each public holiday (bar)
    - Deadliest public holidays (bar)
    - Public holidays with the most alcohol related crashes (bar)
    - Most common accident types (pie chart)
  + This function has no input parameters.
  + The function will show a new GUI window with dashboard style visualisations of the data.
* Export Function
  + This function will be used to allow users to export images of the visualisations throughout the software. When users select the export button, this function is activated and allow users to select where the visualisation is to be saved. This function will be activated when the export button is selected.

### Data Structures / Data Sources

The data source used for this project is the Victoria State Accident dataset. This dataset is available in a csv format and will be imported by the software for use. The pandas library will be used to read in the data source converting it into a class called a data frame. This dataframe can then be filtered and used for visualising the data as needed for each function that requires it.

There are 63 data members that will be created from the data source of which 26 are key to the software’s use. Table 2 shows key data members relating to the:

|  |  |  |  |
| --- | --- | --- | --- |
| **Key Data Members** | | | |
| **Name** | **Description** | **Name** | **Description** |
| ACCIDENT\_DATE | Date of accident | TOTAL\_PERSONS | Total number of people involved in the accident. |
| ACCIDENT\_TIME | Time of accident | INJ\_OR\_FATAL | Number of people injured or deceased. |
| ALCOHOLTIME | If alcohol was involved in the incident | FATALITY | Number of people who are deceased due to the accident. |
| ACCIDENT\_TYPE | Type of accident | SERIOUSINJURY | Number of people who are seriously injured. |
| DAY\_OF\_WEEK | Day of the week the accident occurred | OTHERINJURY | Number of people who are injured in another way. |
| DCA\_CODE | Definition for classifying accident | NONINJURED | Number of people who were not injured during the crash. |
| LIGHT\_CONDITION | Light Condition | MALES | Number of males involved in the crash. |
| ROAD\_GEOMETRY | Type of road at crash | FEMALES | Number of females involved in the crash. |
| SEVERITY | Severity of crash | PEDESTRIAN | Number of pedestrians involved in the crash. |
| SPEED\_ZONE | Speed zone where crash occurred | REGION\_NAME\_ALL | Name of the region in Victoria the crash occurred. |
| NODE\_TYPE | Intersection vs non-intersection | RMA\_ALL | Type of road the accident was on |
| LGA\_NAME | Local Government Area Name | STAT\_DIV\_NAME | Metro/country definition of the accidents location. |
| REGION\_NAME | Name of the region |  |  |

Table 2 - Key data members

Functions that will utilise the data source directly include:

* Load and filter data
* show\_visualisation1
* show\_visualisation2
* show\_visualisation3
* show\_visualisation4
* show\_visualisation5
* Word and data filter function

The export function will be reliant on the data source/structure being used however won’t use the data directly.

### Detailed Design

Shown below is the pseudocode for each of the functions required.

**Load and filter data**

Load in the CSV document to the pandas DataFrame.  
 If dates are specified  
 Load in the CSV document to the pandas DataFrame.  
 Clean and filter the data with date range  
 If no date   
 Load in the CSV document to the pandas DataFrame.  
 Clean the data.  
   
 Return the filtered data for use.

**Word and data filter function**

Load in the CSV document to the pandas DataFrame.

Filter data where keywords are present.

Return filtered data for use.

**show\_visualisation1**

Use load and filter data function and pass the dates to the function to be filtered.

Create a 2x2 grid in the GUI for each chart.   
   
Create chart 1 in the first section of the grid. This shows a bar chart of the number of accidents on a month-by-month basis. The X axis shows the days of the week, Y the quantity of accidents over the date period.

Create chart 2 in the second section of the grid. This shows a bar chart for the most common crash types with the crash types on the x axis and the number of crashes shown on the y axis.

Create chart 3 in the third section of the grid. This shows fatalities in a bar chart with day of the week on the x axis and fatalities on the y axis.

Create chart 4 in the fourth section of the grid. This shows the most common daylight conditions where crashes occur in a pie chart.

Show visualisation.

**show\_visualisation2**

Use load and filter data function and pass the dates to the function to be filtered.  
   
Create a chart. This shows a bar chart for average number of accidents by hour. The X axis shows the hour, Y the average number of  
accidents over the date period.

Show visualisation.

**show\_visualisation3**

Use the word and data filter function to search the spreadsheet for entries with keywords.

Create a 2x2 grid in the GUI for each chart.   
   
Create chart 1 in the first section of the grid. This shows a bar chart with the number of crashes with keyword/s. The X axis shows the month, Y the average number of  
accidents over the date period.

Create chart 2 in the second section of the grid. This shows a bar chart relating to Speed Zones where crashed occurred. The x axis shows the speed zone and the number of crashes shown on the y axis.

Create chart 3 in the third section of the grid. This shows the number of fatalities and injuries in a bar chart with month on the x axis and fatalities on the y axis.

Create chart 4 in the fourth section of the grid. This chart shows the number of alcohol to non-alcohol related crashes in a pie chart.

Show visualisation.

**show\_visualisation4**

Use load and filter data function.

Create a 2x2 grid in the GUI for each chart.   
   
Create chart 1 in the first section of the grid. This shows a pie chart number of accidents relating to alcohol vs non-alcohol related crashes.

Create chart 2 in the second section of the grid. This shows a bar chart relating to number of alcohol related crashes in each LGA. The x axis shows the LGA and the number of crashes shown on the y axis.

Create chart 3 in the third section of the grid. This shows a pie chart with the average number of fatalities in alcohol vs non-alcohol related crashes.

Create chart 4 in the fourth section of the grid. This shows a plot with the average number of pedestrians struck in alcohol related incidents vs non-alcohol related incidents. The x axis shows the month and y shows the average number of people struck.

Show visualisation.

**show\_visualisation5**

Use load and filter data function.

Filter data based on public holiday dates in victoria.

Create a 2x2 grid in the GUI for each chart.

Create chart 1 in the first section of the grid. This shows a bar chart showing the number of accidents on each public holiday. The x axis shows the date (Date + month) and the y axis showing the number of crashes.

Create chart 2 in the second section of the grid. This shows a bar chart showing date on the x axis (day + month) and number of fatalities on the y axis.

Create chart 3 in the third section of the grid. This shows a bar chart showing date on the x axis (day + month) and number of alcohol related crashes on the y axis.

Create chart 4 in the fourth section of the grid. This shows a pie chart showing the most common accident types over public holidays.

# User Interface Design

The user interface design is shown in the next few sections. This includes the structural design of the software which shows how the user interface is structured. This section further details the flow of the software and the journey each user can expect when launching the software. Also included is the visual design. This outlines the visual elements of the software, their locations and features. Each screen has a mock-up design created in Figma showing the outlay and visual structure of the software.

## Structural Design

When the software is launched, a GUI will appear showing the options for visualisation. A title will show at the top of the GUI informing users of the software project they have open. There will be five options for the user to choose from which are based around the five use cases for the software. Each option (use case) on the main/home GUI will be grouped together with the relevant inputs shown next to each option. Users will be able to input their own parameters in some of the options (use cases), and input fields will be shown next to the show visualisation button to indicate that they are linked. When the user is satisfied with the inputs available, they can activate the visualisation by clicking the relevant visualise button which then will open a new GUI window showing the data requested. On the new window, users will have the option to export the data for external use. This structural design allows for all user requirements to be met in an intuitive way and allows for exporting if the user has the need to do so.

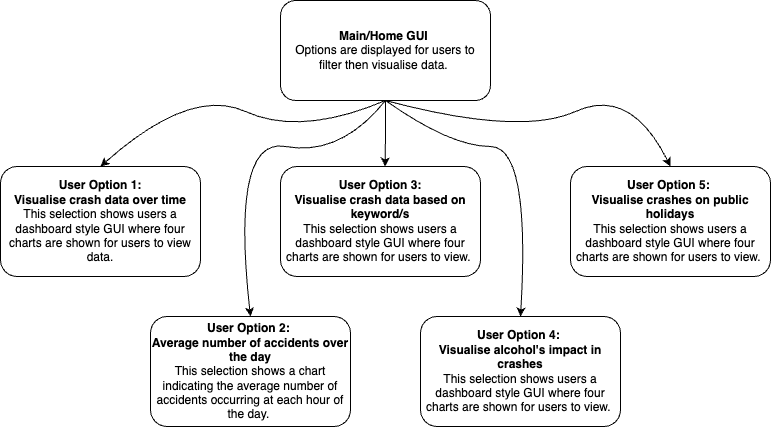


Figure 6 - Diagram showing the hierarchy of software.

The structural design of the software groups key information together. Each user option launches a new window to separate it from the other visualisation options. This is to allow users to better understand the data shown.

## Visual Design

This section of the report shows the visual design of the software. Mock-ups with layouts and visual elements are shown in the figures below.

The GUIs will include titles, text, input fields, buttons, and charts/visualisations to encompass the software as a whole. The core colours (excluding charts) will be black and white for best clarity and contrast. Helvetica is the standard font for the software as it is a sans-serif style font which is easy for users to read and provides a clean look. Each visualisation option will be presented either as four charts or one singular chart.

**Main/Home GUI**

The main/home GUI will show users five sections relating to visualisation options each option will need to have its own title and required inputs and buttons to activate the visualisation. Shown in figure 7 is a mock-up of what the software should look like. The mock-up includes the key features:

|  |  |
| --- | --- |
| **User Option 1:**  Name: Visualise crash data over time.  Inputs: Start Date, End Date Button name: Visualise | **User Option 2:**  Name: Average number of accidents over the day Inputs: Start Date, End Date Button name: Visualise |
| **User Option 3:**  Name: Crash data based on keyword/s Inputs: Keywords Button name: Visualise | **User Option 4:**  Name: Alcohol’s impact on crashes Inputs: None Button name: Visualise |
| **User Option 5:**  Name: Visualise crashes on public holidays Inputs: None Button name: Visualise |  |

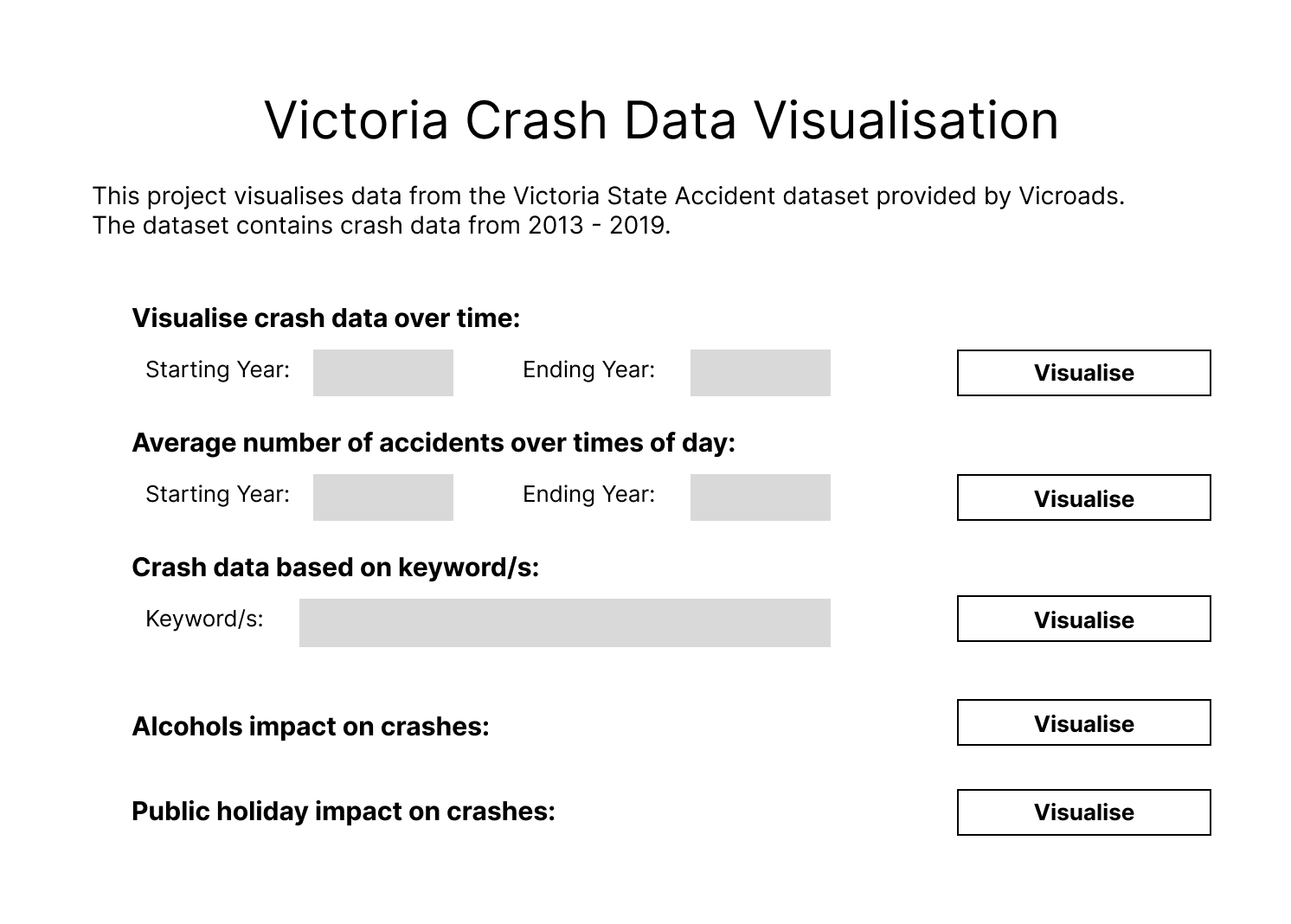


Figure 7 - Mockup of Main/Home GUI

Each user option opens a new GUI window and will show visualisations relating to the option selected. Each option launches the visualisation in a new window, so users can open multiple visualisations at once if they choose. Below outlines what each option would show in each window.

**Visualisation 1 -** **Visualise crash data over time window**:

Shown below is a mock-up of what the software should look like. The mock-up includes the key features:

* Bar chart of the number of accidents on a month-by-month basis.
* Bar chart for the most common crash types.
* Bar chart showing number of fatalities on day of the week.
* Pie chart showing most common daylight conditions where crashes occur.
* Export Button.

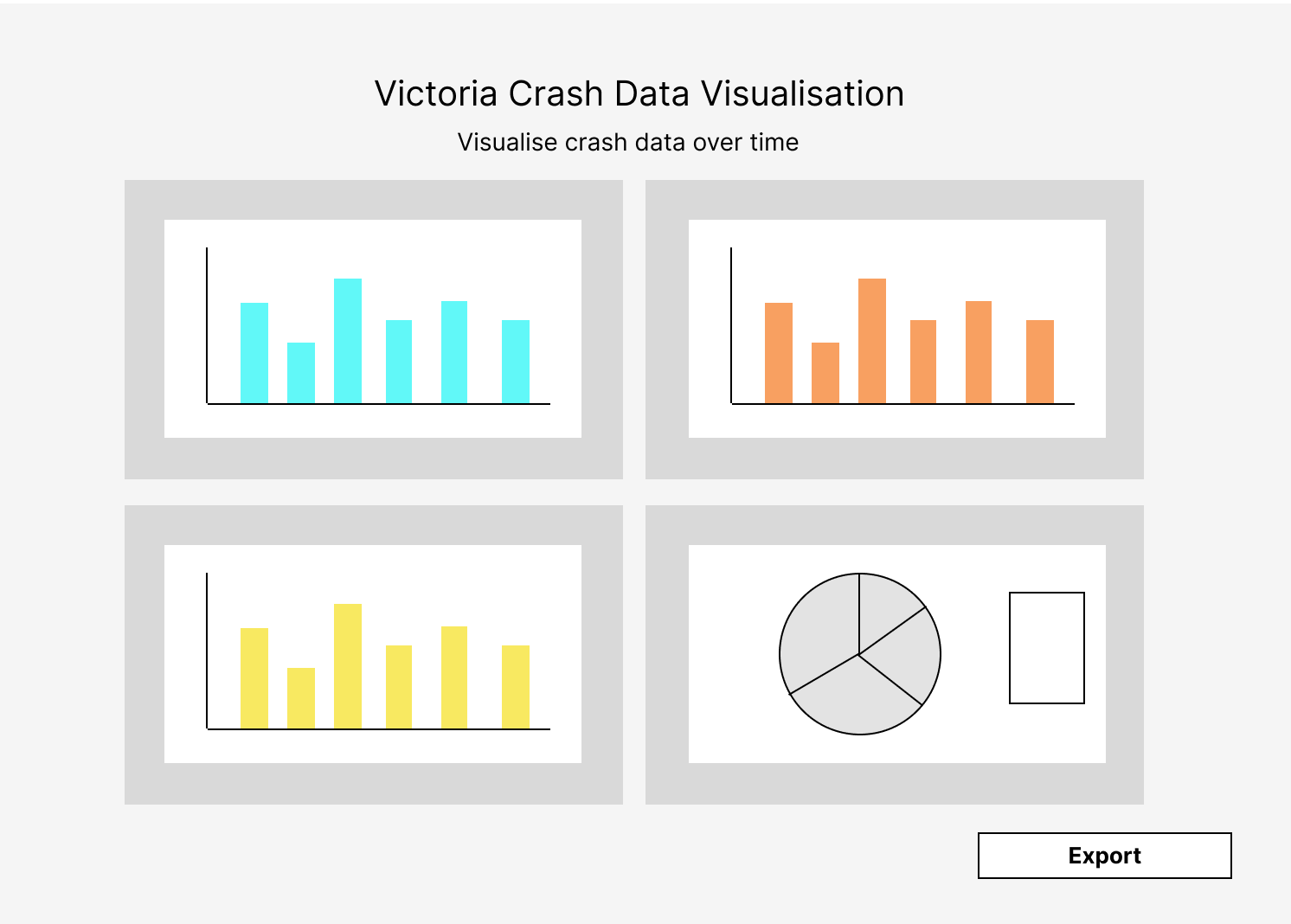


Figure 8 - Visualisation 1: Visualise crash data over time mock-up.

These charts were chosen as they best reflect the information each chart is trying to convey.

**Visualisation 2 - Average number of accidents over the day:**

Shown below is a mock-up of what the software should look like. The mock-up includes the key features:

* Bar chart showing average number of accidents across each hour of the day.
* Export button.

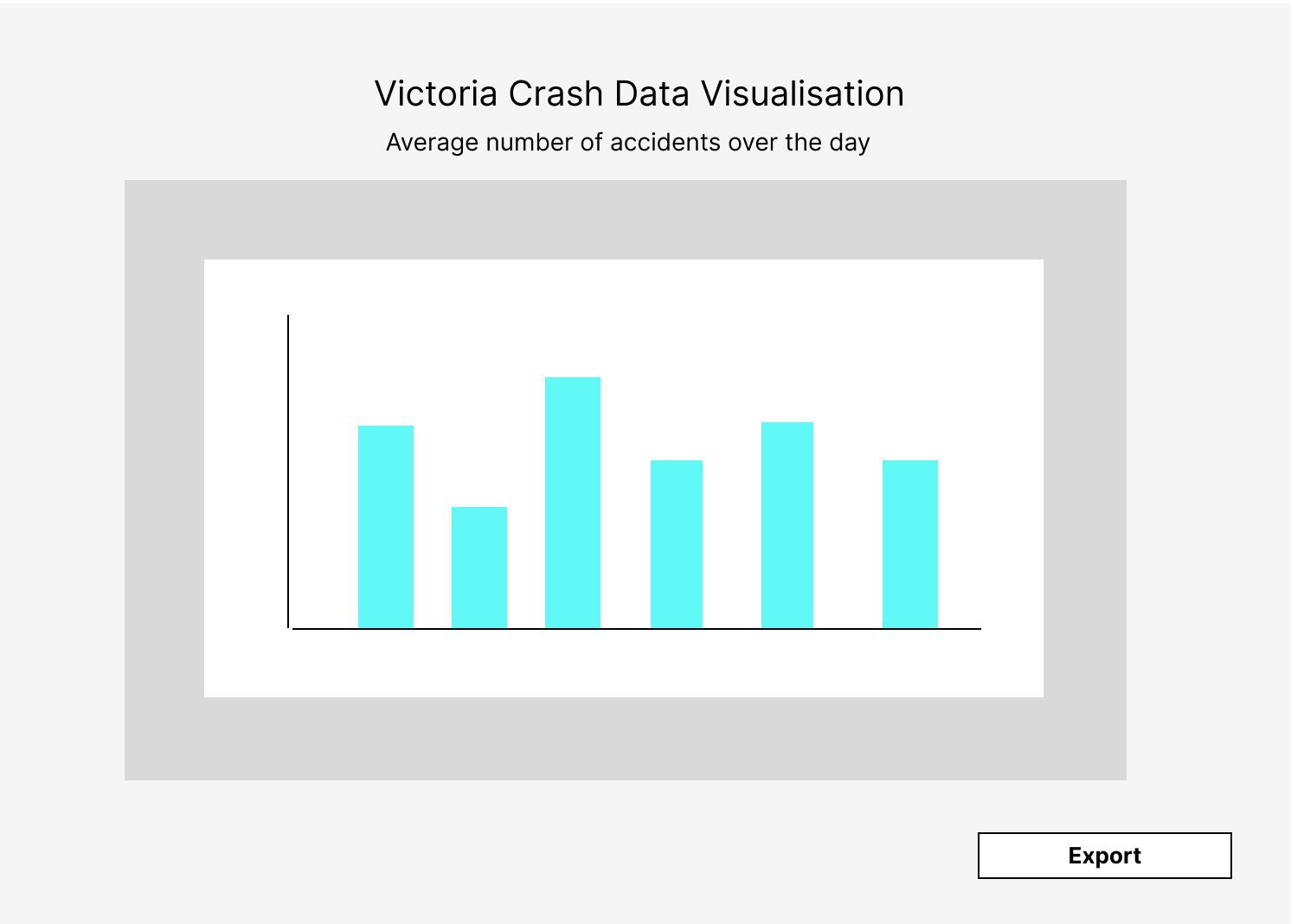


Figure 9 - Visualisation 2: Average number of accidents over the day mock-up.

A bar chart was chosen as this best reflects the data the visualisation needs to convey.

**Visualisation 3 -** **Crash data based on keyword/s:**

Shown below is a mock-up of what the software should look like. The mock-up includes the key features:

* Bar chart with number of crashes across months relating to keyword/s
* Bar chart showing number of crashes occurring in speed zones.
* Bar chart showing number of fatalities and injuries.
* Pie chart showing number of alcohol to non-alcohol related crashes.
* Export button.

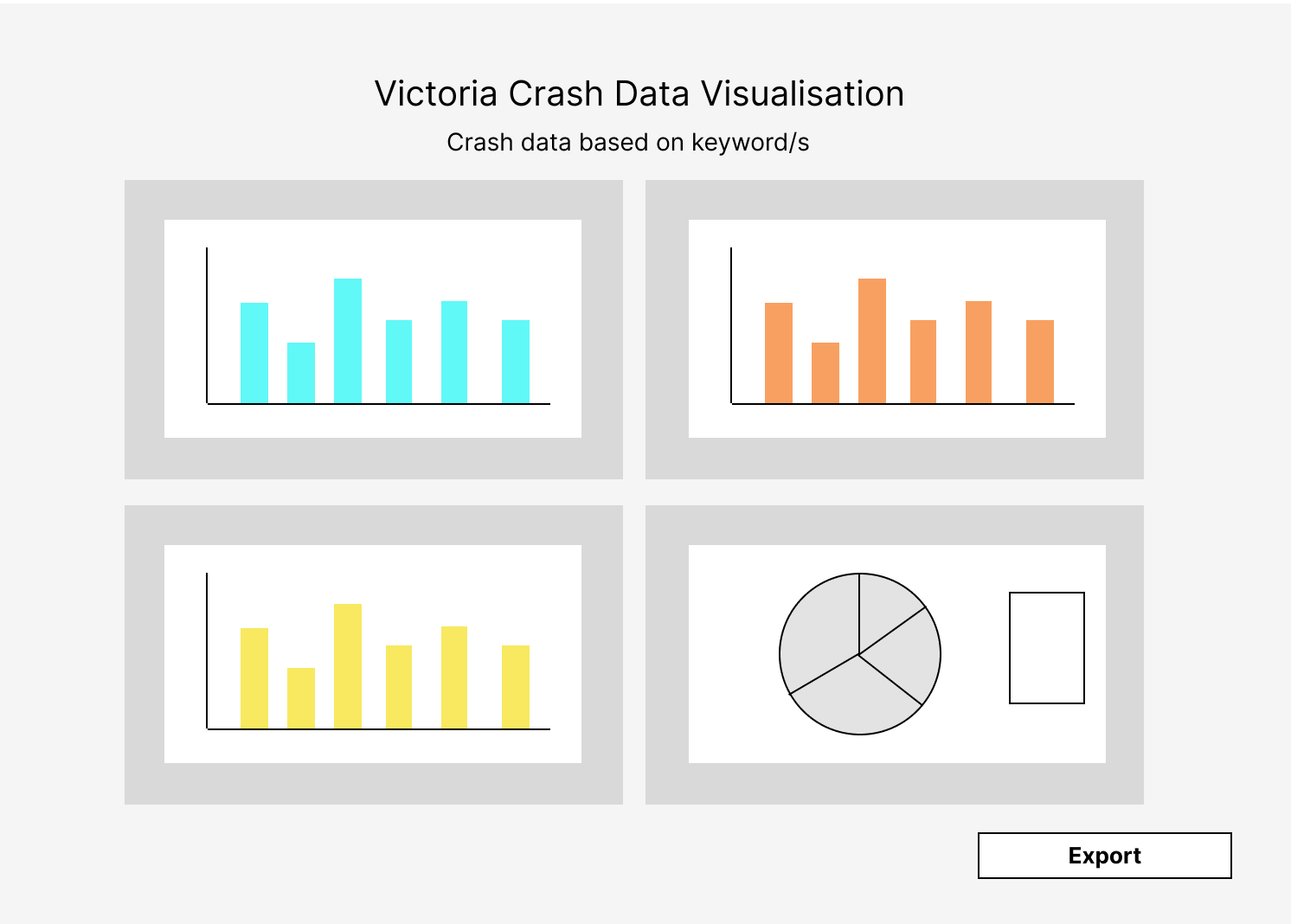


Figure 10 - Crash data based on keyword/s mock-up.

These charts were chosen as they best reflect the information each chart is trying to convey.

**Visualisation 4 -** **Alcohols impact on crashes:**

Shown below is a mock-up of what the software should look like. The mock-up includes the key features:

* Pie chart showing number of accidents relating to alcohol vs non-alcohol related.
* Bar chart showing crash numbers in each LGA.
* Pie chart showing average number of fatalities in alcohol vs non-alcohol related crashes.
* Plot of average number of pedestrians struck in alcohol related incidents vs non-alcohol related incidents.
* Export button.

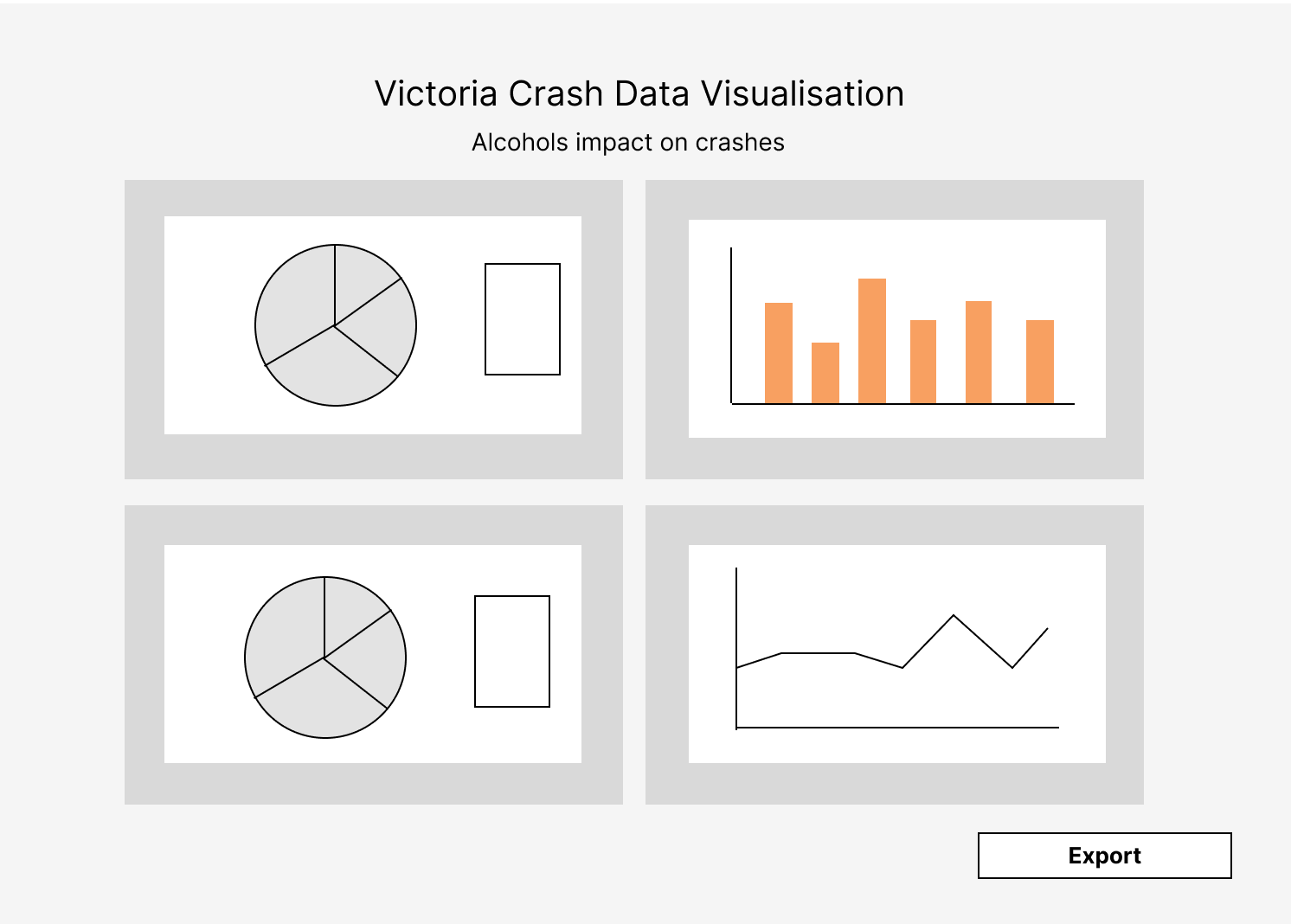
****

Figure 11 - Alcohol's impact on crashes mock-up.

These charts were chosen as they best reflect the information each chart is trying to convey.

**Visualisation 5 -** **Visualise crashes on public holidays:**

Shown below is a mock-up of what the software should look like. The mock-up includes the key features:

* Bar chart showing number of accidents on each public holiday.
* Bar chart showing fatalities on each public holiday.
* Bar chart showing alcohol related crashes.
* Pie chart showing most common accident types over all public holidays.
* Export button.

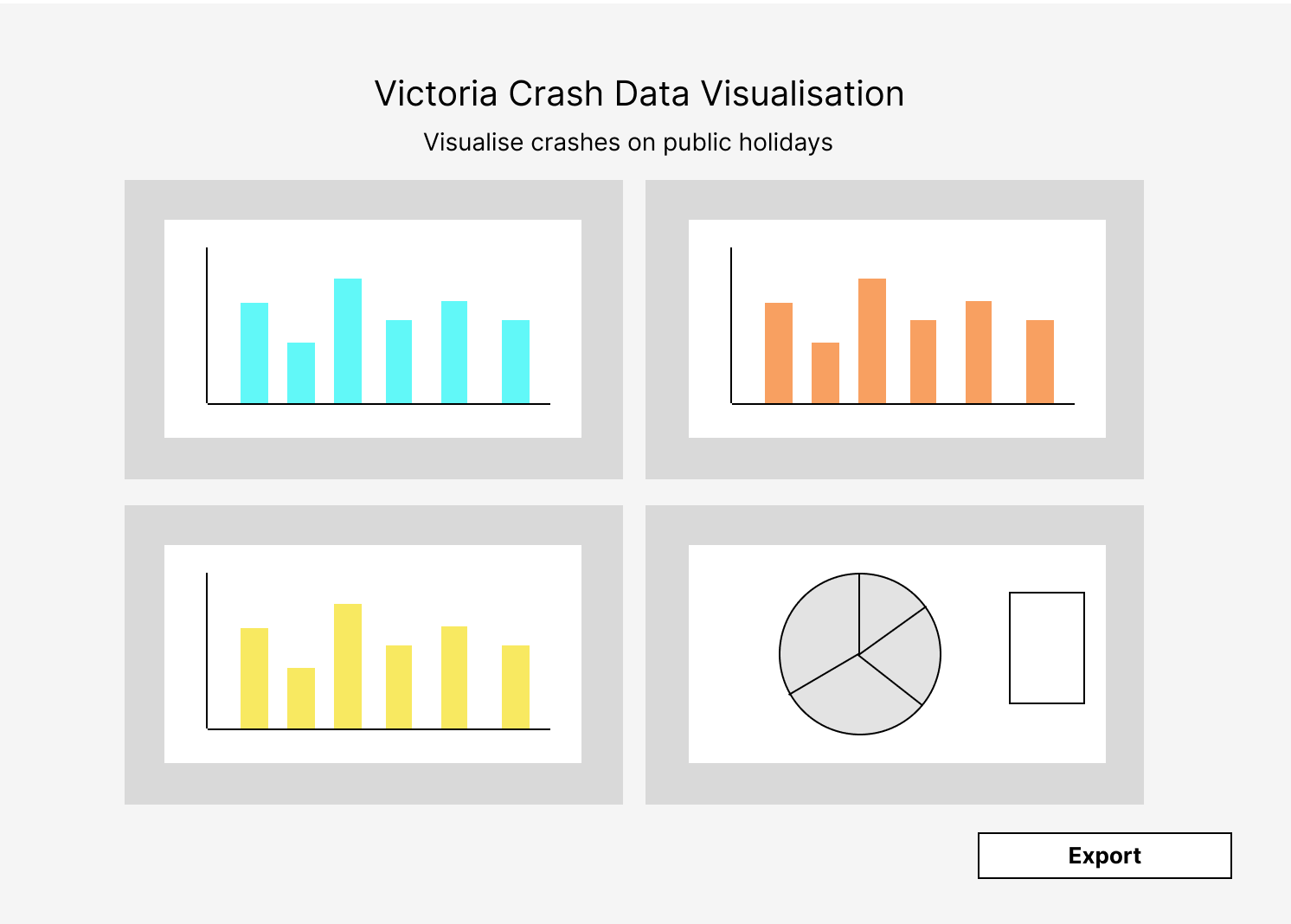


Figure 12 - Visualise crashes on public holidays mock-up.

These charts were chosen as they best reflect the information each chart is trying to convey.