Software Design Document

Victoria State Accident

Ben White - s2850345

Liam Preston - s5301986

Thomas Chapman - s5251549

Table of Contents

[1.0 System Vision 3](#_Toc46748622)

[1.1 Problem Background 3](#_Toc46748623)

[1.2 System Overview 3](#_Toc46748624)

[1.3 Potential Benefits 3](#_Toc46748625)

[2.0 Requirements 4](#_Toc46748626)

[2.1 User Requirements 4](#_Toc46748627)

[2.2 Software Requirements 4](#_Toc46748628)

[2.3 Use Cases 4](#_Toc46748629)

[3.0 System Components and Software Design 5](#_Toc46748630)

[3.1 System Components 5](#_Toc46748631)

[3.2 Software Design 5](#_Toc46748632)

[4.0 User Interface Design 6](#_Toc46748633)

# System Vision

## Problem Background

## System Overview

## Potential Benefits

# Requirements

## User Requirements

In this section you detail how a user is supposed to interact with or use your program. What do they ***need*** to be able to do? This should all be from the end users perspective. Can be a combination of narrative text and listing of needs.

**Assignment note: You have not been given a client/user, so you can make one up. Who do you think would be using your software?**

## Software Requirements

In this section you detail what the requirements for the software are. What functionality will it provide? This is usually a formal listing, with requirements often using the word ‘Shall’. IE:

R1.1 The program shall accept multiple file names as arguments from the command line.

R1.2 Each file name can be a simple file name or include the full path of the file with one or more levels.

etc …

Can be primarily functional requirements, though you may include other types if you think of them.

## Use Cases & Use Case Diagrams

In this section you provide some use cases showing how people may use your software.

# Software Design and System Components

## Software Design

A block diagram/flowchart of how your software might work

A diagram of a company

Description automatically generated

## System Components

### Functions

Preliminary list of all functions in the software. For each function in the list the following information is provided:

* a brief description of what it does (1 or 2 sentences);
* a list of the input parameters, and their data types, and what they are used for;
* a list of any side effects caused by the function (ie change global or member variables, changes data passed by reference from calling function etc)
* a description of the function’s return value

Load Data Function

**Description:**

The Load Data Function is responsible for reading and retrieving data from an external source, such as a file or a database, and making it available for use within the software.

**Input Parameters:**

1. `source` (string):

- Data Type: String

- Description: Specifies the source of the data, which could be a file path, database connection string, URL, or other relevant source identifier.

2. `options` (object, optional):

- Data Type: Object

- Description: An optional object containing settings or configuration parameters for loading the data. This may include options like file format, data format conversion, or authentication credentials for database access.

**Side Effects:**

- The Load Data Function may have the following side effects, depending on its implementation:

- It may read data from the specified source and load it into memory or a data structure for further processing.

- It may establish or close connections to external data sources, which could impact system resources.

- It may log data loading activities or errors for auditing or debugging purposes.

**Return Value:**

- The return value of the Load Data Function is typically the loaded data or a reference to the data structure containing the loaded data. The data type of the return value depends on the nature of the data and the implementation. It could be a list, array, data frame, database cursor, or other appropriate data structure.

The primary purpose of the Load Data Function is to acquire data from external sources and prepare it for subsequent use within the software, enabling data-driven applications and analysis.

R1.1 The system shall provide a user-friendly interface for easy navigation and interaction.

**Description**: The navigation bar provides a menu interface for users to navigate through different sections or pages of a website or application.

**Input Parameters**:

1. `menuItems` (array of objects): An array of menu items, where each object represents a menu item with properties like `label` (string) for the display text and `link` (string) for the URL.

**Side Effects**:

- None. The navigation bar component typically does not change global variables or modify data passed by reference.

**Return Value**:

- The navigation bar component is typically a user interface element and does not have a specific return value. It renders the menu items on the screen for user interaction.

R1.2 The system shall enable users to select specific date ranges for data retrieval.

**Description**: The Date Range Function is used to calculate or manipulate date ranges within a dataset.

**Input Parameters:**

1. `dates` (list or array of date objects):

- Data Type: List or Array of Date Objects

- Description: This parameter represents a collection of date values for which you want to calculate a date range or perform date-related operations.

2. `operation` (string):

- Data Type: String

- Description: Specifies the type of operation to be performed on the date range. It can be "calculate\_range," "filter\_dates," "normalize," or other applicable operations.

3. `parameters` (object):

- Data Type: Object

- Description: An object containing operation-specific parameters. The content of this object may vary based on the chosen operation. For instance, for "filter\_dates," it might contain filtering conditions, such as a start date and end date.

**Side Effects:**

- The function may have side effects depending on the operation and implementation:

- If the operation modifies the input `dates` in place, it will change the data passed by reference from the calling function.

- If the operation affects global variables or external data sources, document it here.

**Return Value:**

- The return value of the Date Range Function depends on the chosen operation:

- For "calculate\_range," it typically returns a tuple or object containing the calculated date range (start date and end date).

- For "filter\_dates," it returns a new list or array containing the filtered dates.

- For "normalize," it returns a new list or array with normalized dates.

- Describe the specific return value format and content for each operation the function can perform.

R1.3 The system shall display accident details for user-selected periods.

**Description**:

The "user-selected periods" function allows users to specify and work with specific time intervals or periods within a dataset.

**Input Parameters:**

1. `data` (list or array of time-series data):

- Data Type: List or Array

- Description: This parameter represents a dataset containing time-series data, where users want to select specific periods.

2. `selected\_periods` (list of tuples or objects):

- Data Type: List

- Description: A list of tuples or objects specifying user-selected periods. Each tuple or object should define the start and end dates or timestamps for a period.

3. `operation` (string):

- Data Type: String

- Description: Specifies the operation to be performed on the user-selected periods. It can be "aggregate\_data," "analyze\_periods," or other relevant operations.

4. `parameters` (object):

- Data Type: Object

- Description: An object containing operation-specific parameters. The content of this object may vary based on the chosen operation. For instance, for "aggregate\_data," it might contain aggregation functions like "mean," "sum," etc.

**Side Effects:**

- The function may have side effects depending on the operation and implementation:

- If the operation modifies the input `data` in place, it will change the data passed by reference from the calling function.

- If the operation affects global variables or external data sources, document it here.

**Return Value:**

- The return value of the function depends on the chosen operation:

- For "aggregate\_data," it returns aggregated results within the user-selected periods, such as mean values, sums, etc.

- For "analyze\_periods," it returns analysis results, such as statistical summaries or insights based on the selected periods.

- Describe the specific return value format and content for each operation the function can perform.

R1.4 The system shall produce an hourly chart showcasing the average number of accidents for chosen periods.

**Description:**

This function generates an hourly chart displaying the average number of accidents that occurred during user-selected periods.

**Input Parameters:**

1. `accident\_data` (list or array of accident records):

- Data Type: List or Array

- Description: Represents a dataset containing accident records, with each record including information about the date, time, and other relevant details.

2. `chosen\_periods` (list of tuples or objects):

- Data Type: List

- Description: A list of tuples or objects specifying user-selected periods. Each tuple or object should define the start and end timestamps for a period during which accidents should be considered.

3. `operation` (string):

- Data Type: String

- Description: Specifies the operation to be performed, which is to create an hourly chart showing the average number of accidents during the chosen periods.

4. `parameters` (object):

- Data Type: Object

- Description: An object containing parameters specific to the operation. For example, it might include settings for chart appearance or aggregation functions.

**Side Effects:**

- The function may not have significant side effects:

- It primarily processes input data to generate the chart and does not typically change global variables or modify the input data passed by reference.

**Return Value:**

- The function generates an hourly chart showcasing the average number of accidents for the chosen periods. As such, it typically does not have a specific return value, as the result is typically a visual chart or graph. The chart can be displayed directly or saved to a file for further use or visualization.

The function's primary purpose is to visualize and present the average accident data in an hourly chart based on user-selected periods.

R1.5 The system shall allow users to input keywords and filter accident types accordingly.

filters accident types based on input keywords:

**Description:**

This function filters accident types based on user-input keywords, allowing users to narrow down the dataset to specific accident types of interest.

**Input Parameters:**

1. `accident\_data` (list or array of accident records):

- Data Type: List or Array

- Description: Represents a dataset containing accident records, each record containing details about the accident, including its type.

2. `keywords` (list of strings):

- Data Type: List of Strings

- Description: A list of keywords or search terms provided by the user. The function will filter accident types that match these keywords.

3. `operation` (string):

- Data Type: String

- Description: Specifies the operation to be performed, which is to filter accident types based on input keywords.

4. `parameters` (object):

- Data Type: Object

- Description: An object containing parameters specific to the operation, such as search options or filtering criteria.

**Side Effects:**

- The function typically does not have significant side effects:

- It processes the input data to filter accident types based on keywords but does not typically change global variables or modify the input data passed by reference.

**Return Value:**

- The function returns a filtered list or array of accident records that match the input keywords. The filtered dataset contains accident records with accident types that are relevant to the user's search criteria.

The primary purpose of this function is to provide users with a filtered dataset of accident records, making it easier for them to focus on specific accident types based on their input keywords or search terms.

R1.6 The system shall offer dedicated analysis tools focusing on alcohol-related accidents.

**Description:**

These analysis tools are designed to provide insights and statistics specifically related to alcohol-related accidents, allowing for a more detailed examination of this subset of accidents within a larger dataset.

**Input Parameters:**

1. `accident\_data` (list or array of accident records):

- Data Type: List or Array

- Description: Represents a dataset containing accident records, each record including details about the accident, including information about whether alcohol was a contributing factor.

2. `operation` (string):

- Data Type: String

- Description: Specifies the operation to be performed by the analysis tools, such as "calculate\_statistics," "generate\_charts," or other relevant operations specific to alcohol-related accidents.

3. `parameters` (object):

- Data Type: Object

- Description: An object containing parameters specific to the operation. For instance, for "calculate\_statistics," it might include settings for statistical analysis, and for "generate\_charts," it might include charting options.

**Side Effects:**

- The analysis tools should generally not have significant side effects:

- They analyze the input data to generate statistics or visualizations specific to alcohol-related accidents but do not typically change global variables or modify the input data passed by reference.

**Return Value:**

- The return value of the analysis tools depends on the chosen operation:

- For "calculate\_statistics," it returns statistical summaries or metrics related to alcohol-related accidents, such as the average severity, frequency, etc.

- For "generate\_charts," it may return visual charts or graphs showcasing trends or patterns related to alcohol-related accidents.

- The specific format and content of the return value vary based on the analysis tools' operation.

These analysis tools are intended to provide users with focused insights and information related to alcohol-related accidents, aiding in decision-making and safety efforts.

R1.7 The software shall provide an "insight tool" to automatically generate notable patterns or trends from the dataset.

**Description:**

The Insight Tool is designed to automatically analyze a dataset and identify significant patterns or trends, providing users with valuable insights without requiring manual analysis.

**Input Parameters:**

1. `data` (list or dataset):

- Data Type: List, Dataset, or appropriate data structure

- Description: Represents the dataset to be analyzed for patterns and trends.

2. `operation` (string):

- Data Type: String

- Description: Specifies the operation to be performed by the Insight Tool, such as "automated\_pattern\_detection," "trend\_identification," or other relevant operations for automatic data analysis.

3. `parameters` (object):

- Data Type: Object

- Description: An object containing parameters specific to the operation. These parameters may include settings for the type of patterns or trends to be detected, as well as thresholds for significance.

**Side Effects:**

- The Insight Tool should generally not have significant side effects:

- It analyzes the input data to automatically generate insights but does not typically change global variables or modify the input data passed by reference.

**Return Value:**

- The return value of the Insight Tool depends on the chosen operation:

- For "automated\_pattern\_detection," it may return a structured report or data structure containing detected patterns and their significance.

- For "trend\_identification," it may return a report or data structure outlining identified trends and their characteristics.

- The format and content of the return value are specific to the operation and are designed to provide actionable insights to the user.

The Insight Tool aims to automate the process of discovering important patterns or trends within a dataset, making it a valuable tool for data-driven decision-making and discovery.

R1.8 The system shall ensure data privacy and prevent unauthorized access.

**Description:**

This function or system is responsible for safeguarding sensitive data and preventing unauthorized access to ensure the privacy and security of the information.

**Input Parameters:**

1. `data` (various data types):

- Data Type: Dependent on the data being protected (e.g., database records, files, user accounts, etc.)

- Description: Represents the sensitive data that needs protection.

2. `authorization\_credentials` (object or authentication token):

- Data Type: Object, Token, or appropriate authentication mechanism

- Description: Authentication credentials or tokens provided by authorized users or systems to gain access to the protected data.

3. `operation` (string):

- Data Type: String

- Description: Specifies the operation to be performed, such as "access\_data," "update\_data," or "authenticate\_user."

4. `parameters` (object):

- Data Type: Object

- Description: An object containing parameters specific to the operation, including access control settings, data encryption keys, or user permissions.

**Side Effects:**

- Depending on the implementation, the function or system may have side effects, which can include:

- Modifying global variables related to access control.

- Logging access attempts and security-related events.

- Blocking access to unauthorized users or systems.

- Encrypting or decrypting data as needed.

- Modifying data in accordance with authorized operations.

**Return Value:**

- The return value varies depending on the operation and can include:

- For "access\_data," it returns the requested data if the user or system is authorized, or it returns an access denied message.

- For "update\_data," it may return a success message or an error message based on the outcome of the update operation.

- For "authenticate\_user," it typically returns an authentication token or user session if authentication is successful, or it returns an authentication failure message.

This function or system plays a critical role in maintaining data privacy and security by ensuring that sensitive information is accessible only to authorized users or systems while preventing unauthorized access and data breaches.

### Data Structures / Data Sources

List of all data structures in the software (eg linked lists, trees, arrays etc) or eternal data sources. For each data structure in the list the following information is provided:

* Type of structure (tree, list etc),
* Description of where and how it is used
* List of data members, and what each one is for do
* List of functions that use it

External data source provided by <https://www.kaggle.com/datasets/gaurav896/victoria-state-accident-dataset?resource=download>

**Arrays**

* **Type**: Linear structure
* **Description**: Used to store a collection of elements of the same data type in contiguous memory locations.
* **Data Members:** Elements, Length (number of elements)
* **Functions**: Indexing, Insertion, Deletion, Searching, Sorting, Iteration

Linked Lists

* **Type**: Linear structure
* **Description**: Used to store elements in a sequence, where each element points to the next one.
* **Data Members**: Head (pointer to the first element), Tail (pointer to the last element)
* **Functions:** Insertion, Deletion, Searching, Traversal

**Graphs**

* **Type:** Non-linear structure
* **Description:** Used for modeling relationships between objects, like social networks, network topology, etc.
* **Data Members**: Nodes, Edges, Adjacency List/Matrix
* **Functions:** Traversal (DFS, BFS), Shortest Path Algorithms, Connectivity Checks

Files

* **Type**: External Data Source
* **Description**: Used for reading and writing data to non-volatile storage.
* **Data** **Members**: File Handles, Buffers
* **Functions**: Read, Write, Open, Close, Seek

Trees (Binary Trees, AVL Trees, etc.)

* **Type**: Hierarchical structure
* **Description**: Used for hierarchical data representation, like file systems, organization charts, etc.
* **Data Members**: Root (pointer to the root node), Parent, Left Child, Right Child
* **Functions**: Insertion, Deletion, Traversal (Inorder, Preorder, Postorder)

### Detailed Design

Pseudocode for all non-standard / non-trivial algorithms that operate on data structures

function binary\_search(array, target):

left = 0

right = length(array) - 1

while left <= right:

mid = (left + right) // 2

if array[mid] == target:

return mid

elif array[mid] < target:

left = mid + 1

else:

right = mid - 1

return -1

# User Interface Design

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.

In this design stage we have used a variety of tools to help come up with how we are looking to have an idea of the final design of the product. The main of course being the data set provided so we can get an understanding of what we are working with. to have an idea of how we want the page to look we had a look at Liam Preston’s assessment from a previous assessment which is the image displayed below in section 4.1. other tools to help map out what we are looking to do were tools such as draw.io, lucidchart and external personal work tools (e.g. hudl sports code) in which we used to get a layout and ideas on what we want the graphs to potentiall look like as well as providing us with a structural design oh what data and information each graph will contain.

A screen shot of a speedometer

Description automatically generated

(Hudl sports code)

The image above is an example of the type of graphs we were looking at. As you can see the brief design there are buttons on the side in which we will make interactive for the users so that they can select what they’re specifically wanting to look at. As this is the early stage of building the design we are still looking at options but are going to keep it similar so that it stands out to the users and not just a plane fixed graph. The idea is they can pick and select the data in which they are wanting rather than using all the data provided.

## Structural Design

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.

When coming up with the structural design we have learnt that this is a dataset in which people from a wide range of ages will be looking at. It can start from the young ages of 15 where it could be used for school assignments, up to the possible age of 80+. Baring this in mind we need to make sure that this design will be simple and easy to use for everyone. If we over complicate it the older population won’t understand how to use it but we want to make sure that this design speaks out to all users. Providing everyone with facts and accurate results to help educate everyone in safety.

In the image below shows a little example of a potential way in which we look to have the initial design.

A screenshot of a computer

Description automatically generated

(Liam Preston (2022) ‘Assignment Part B’ *1621ICT\_3221\_OL: 1621ICT WEB Technologies* Griffith University Unpublished essay/assignment)

Is simple but easy to navigate. At the top you have a clear navigation bar which highlights which page is being viewed so that that whoever is using will know what they’re looking at. It also shows how we would want it to stand out with clear description for everyone to read and what the site will be used for.

For our design we are looking to have 6 pages. These pages will be the; home/landing page, Accidents within time period, Accidents per hour, accident per external subject, alcohol related accidents and speed related accidents. These pages will be all for educational purposes. We want to help educate the public with the data.

Each page will use different parts of the data provided by: <https://www.kaggle.com/datasets/gaurav896/victoria-state-accident-dataset?resource=download> when grouping up the data we’ll aim to select the specific information from the data provided so we don’t over complicate the data for everyone.

Here is how we looking to group the data per page:

* Home/landing page
  + Just an overview of what the page is about
* Accidents within time period
  + Accident\_Date
  + Day\_of\_the\_week
  + Region\_Name
  + Total\_Persons
  + INJ\_OR\_Fatal
  + Fatalities
  + Old\_Driver
  + Young\_Driver
  + RMA
  + Unlicensed
* Accidents per hour
  + Accident\_Date
  + Day\_of\_the\_week
  + Region\_Name
  + Total\_Persons
  + INJ\_OR\_Fatal
  + Fatalities
  + Old\_Driver
  + Young\_Driver
  + RMA
  + Unlicensed
* accident per external object
  + Accident\_Date
  + Day\_of\_the\_week
  + Light\_Condition
  + Pedestrian
  + Pillion
  + Unknown
  + Region\_Name
  + Total\_Persons
  + INJ\_OR\_Fatal
  + Fatalities
  + Old\_Driver
  + Young\_Driver
  + RMA
  + Unlicensed
* alcohol related accidents
  + Accident\_Date
  + Day\_of\_the\_week
  + Light\_Condition
  + Alcohol\_Related
  + Region\_Name
  + Total\_Persons
  + INJ\_OR\_Fatal
  + Fatalities
  + Old\_Driver
  + Young\_Driver
  + RMA
  + Unlicensed
* speed related accidents
  + Accident\_Date
  + Day\_of\_the\_week
  + Light\_Condition
  + Speed\_Zone
  + Run\_Offroad
  + Region\_Name
  + Total\_Persons
  + INJ\_OR\_Fatal
  + Fatalities
  + Old\_Driver
  + Young\_Driver
  + RMA
  + Unlicensed

The data may seem similar as the but each page has roughly the same but then specific data in which the page is focusing on. This is due to the similarities in the pages. It is important that we keep the pages relating the same common data but then have its specific data from the data set in which we are focusing on. This will help make our site easier for the users to use and make sure they’re getting the right information.

## Visual Design

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.

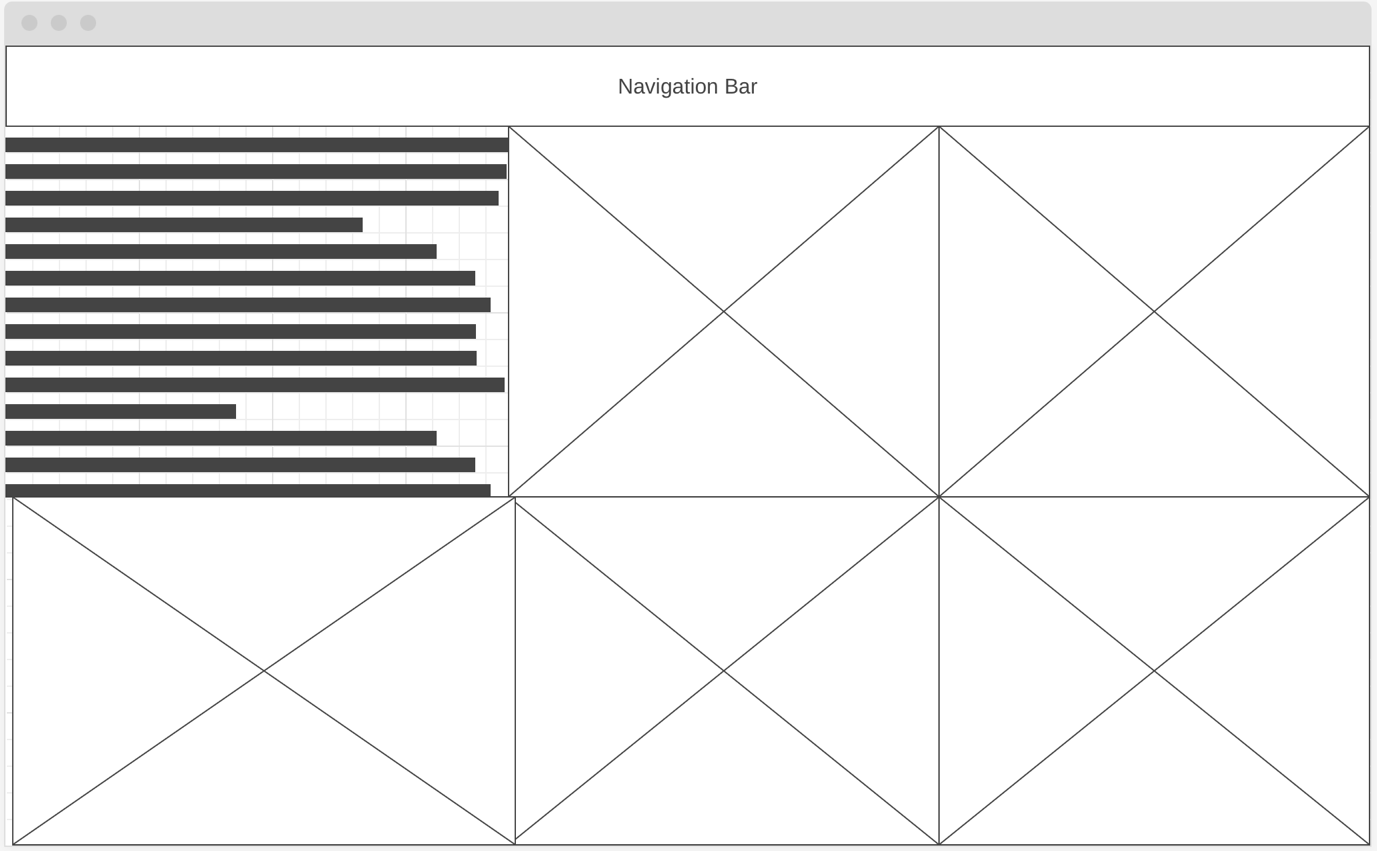
Home/Landing page

A screenshot of a computer

Description automatically generated

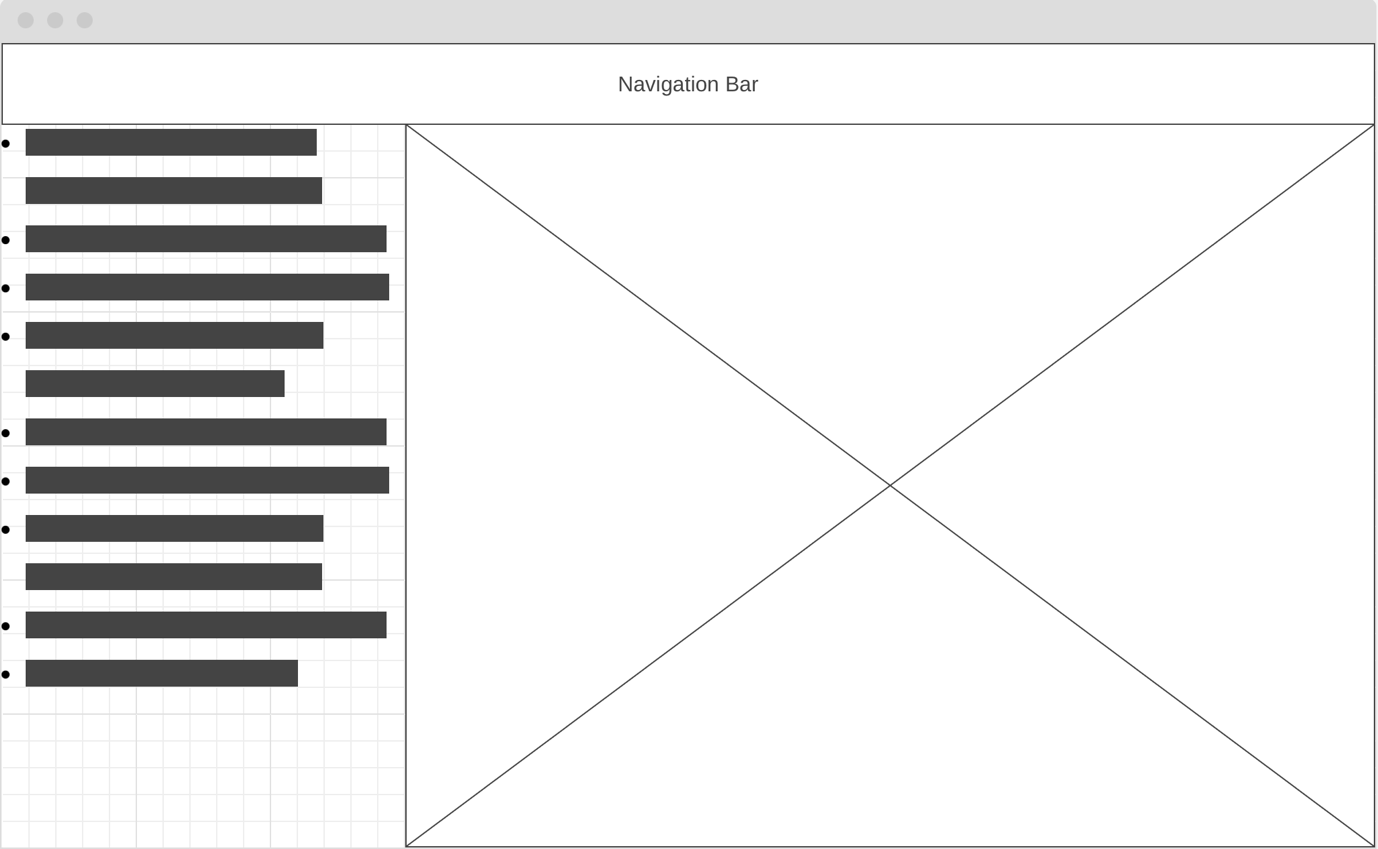
This page will be of simple design to help people of all ages navigate around with being able to achieve the data that they’re looking for. On this page it’ll state what the website will be used for and give a brief description and have an interactive navigation bar at the top where people can move from page to page. Colour wise for this page we want it to be summer tones to stand out. We want the page to pop and show the beautiful state of Victoria rather than dark colours. We want the navigation bar at the top to stand out so the users know the difference between the navigation bar and the background page.

Accidents within time period page



This page is made up of 5 graphs which will outline the number of accidents over each of the 5 individual years this data set is spread amongst. It will also have a section which has a brief explanations of what this page is and provide facts on the most common form of accidents and how we can help provide any information on how to prevent them. For the graphs we’re looking to make each year a different colour so the users know which one is for what year but the data within the graphs stay the same. And potential example of the graph can be seen in section 4.0 user interface design.

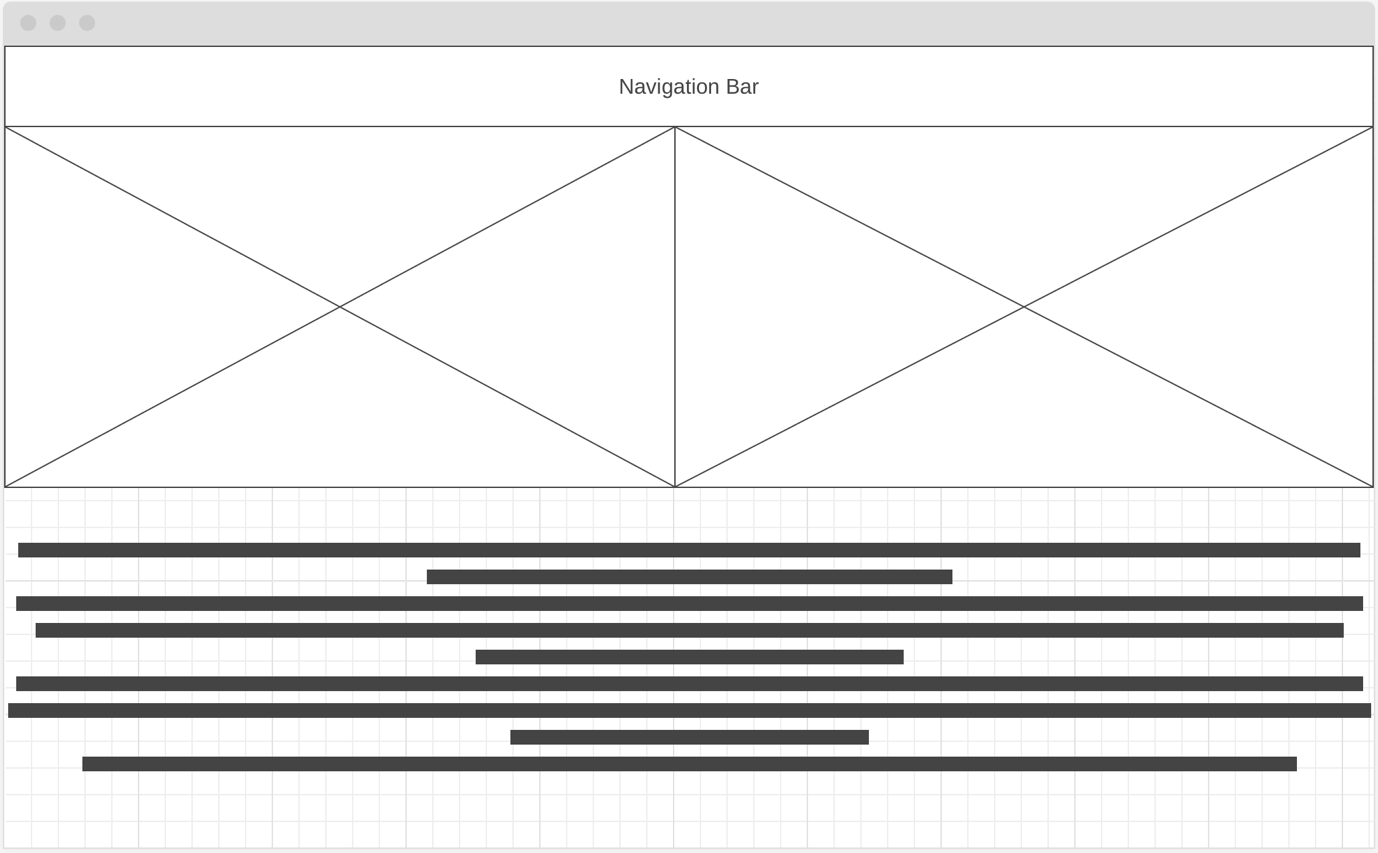
Chart showing number of accidents during each hour of the day



This page will be an interactive page where the user will be able to select an input certain hours of the day to see when most accidents commonly happen. We will also provide safety tips on how to limit the chances of you being involved in one of these accidents plus the what to do if you are in an accident and the step to help guide you through the situation.

Colours for this page we want to keep vibrant. The graph you can reference an idea from section 4.0 user interface design. The facts and information on the side we want to be bold so that it stands out. We are planning to keep a common background or similar colours across all pages so that we keep each page referencing each other. Colours will be summer and warm colours to provide a nice warm feeling so the users are comfortable and happy while looking at the page.

accidents caused by subject



This page is all about the accident caused by external subjects other than the vehicles. External subjects can be many factors such as pedestrians, dirt, fog, animals etc. This page’s layout is designed into two graphs and information about how you can help limit the amount of accidents in caused by external factors. 1 graph will display the most common different types of external factors causing these accidents as well as the other graph showing you around what time of year these accidents would normally occur.

This page is the same in terms of the summer/warm colour look. With the fact being bold for education purposes it’ll stand out in front of the of the colour tone. The two graphs on these pages will be difference but similar colours as they relate to each other. One will be the graph relating to the data on the flow chart and the other one will be related solely to the external objects in which have been causing the accidents so they can have a clear indicator compared to all the data.

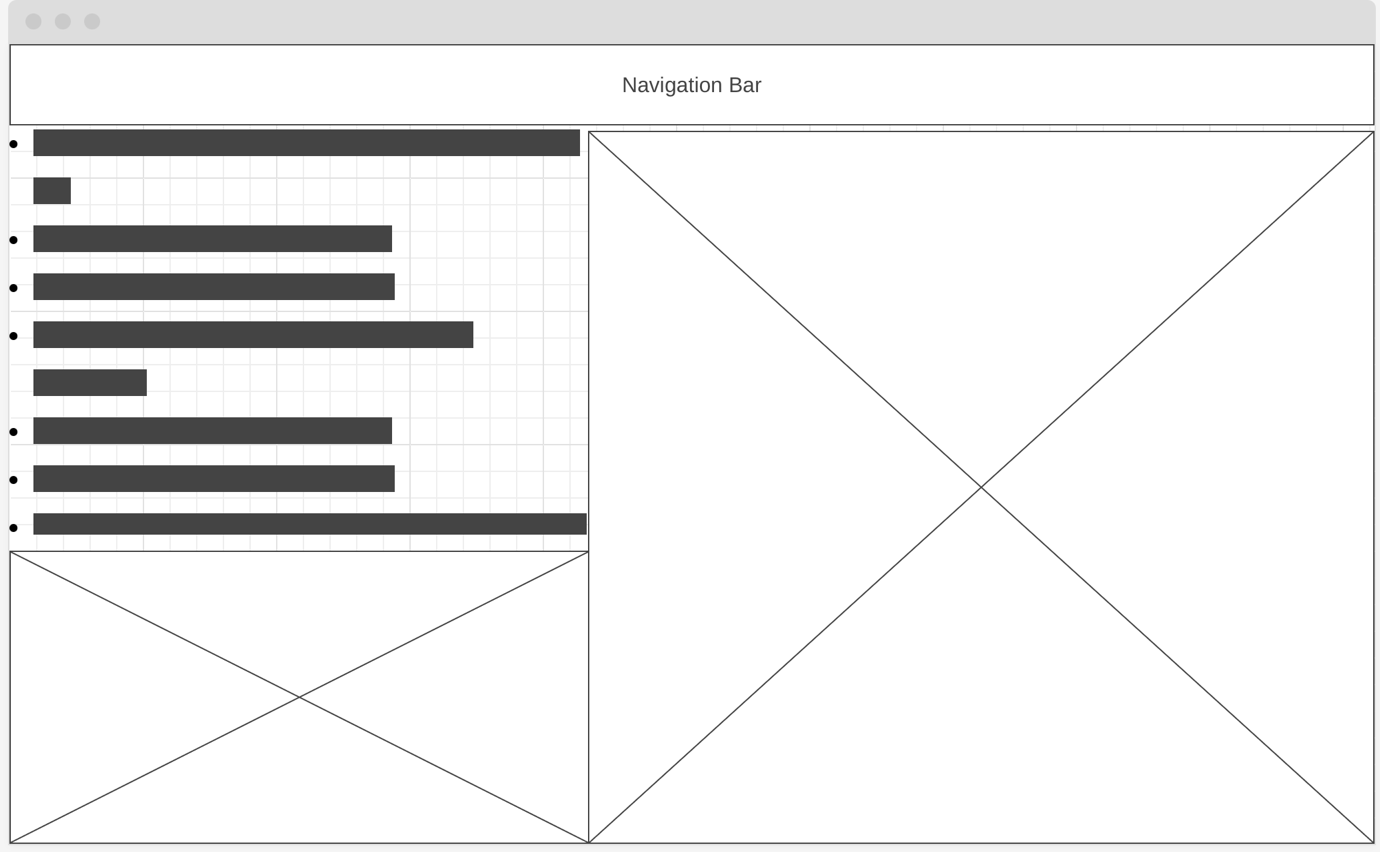
alcohol related accidents page



This page will have facts on surrounding the graph to help show the dangers of driving under the influence of alcohol. These facts are not a scare tactic but will be used for educational purposes on the negative effects of what alcohol can do to ones mind and how it can effect your capability to drive. the graph will also be interactive and allow the user to go through the years as well as be able to see the busiest times that these accidents usually occur.

Colours for this page will still be the same in terms of wanting it to be summer colours providing a warm feeling. The graphs you can reference an idea from section 4.0 user interface design. The facts and information on the bottom and left of the graph, we want to be bold so that it stands out. We want this page to be educational about the dangers of drink driving and the dangers it can cause not only to you but also everyone else on the road and off the road.

speed relating to severity of incident



This page will display the 2 graphs. 1 displaying the different types of injuries occurred and the other displaying the speeds of which each injuries occurred at. On this page it will also display facts on why there are speed limits put in place to help with limiting the causes and severity of the injuries at the events.

This page is the same in terms of the summer/warm colour look. The two graphs on these pages will be different but similar colours as they relate to each other. One will be the graph relating to the data on the big chart and the other one will be related solely to the speed and at what speed causes fatal or serious injuries. We want the facts to be educational on the dangers of speeding for the drivers and for the people on and off the road as your accident can cause people who aren’t eve driving serious injuries too.