Software Design Document

<Data Analysis and Visualisation tool>

Harmol Kaur

Jigme Yeshey

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)

5.0 Conclusion……………………………………………………………………………………………………………………………………7

# System Vision

## Problem Background

The Victoria State Accident Dataset Contains a wealth of information related to accidents that occurred over a five-year period from 2015 to 2020. This dataset provides a comprehensive record of road crash statistics, enabling in-dept analysis of road safety incidents. Even though, a large amount of data is available, effectively extracting insights and trends from this dataset poses significant challenges. Without a dedicated tool, the process of visualising the data and gaining actionable insights becomes even more toiling. This can hinder the ability of researchers or the policymakers to make informed decisions based on accurate accident analysis. Moreover, manually sifting through vast amounts of data to analyse accident patterns, identify contributing factors and predict accident outcomes is a time consuming and complex task. Thus, to address these challenges, our project aims to develop a user-friendly efficient data analysis and visualization tool specifically tailored to the Victoria State Accident dataset.

## System Overview

The envisioned data analysis and visualization tool for the Victoria State Accident Dataset will offer an intuitive and user-friendly interface that empowers users to interact seamlessly with the complex accident data. The system will integrate advanced data processing and visualization techniques to provide users with clear understanding of accident trends and contributing factors. The system will allow the users to select specific time frames to analyse accident data with the desired periods. The system will display detailed information about the accidents including accident date, time, type, and all other information related to the accidents. Users will be able to visualise the average number of accidents per hour throughout the day, aiding in identifying high-risk hours. The system will enable users to search for accidents bases on accident types containing specific keywords. Users can analyse the impact of alcohol and potential correlations. The system will provide one additional insight by recommending the users the safety measures that can be taken based on the accident frequency in the particular location, enhancing the tool’s analytical capabilities.

## Potential Benefits

The envisioned data analysis and visualization tool for the Vitoria State Accident Dataset offer numerous potential benefits across various domains. By providing a user-friendly interface that enables users to interact with the dataset in a meaningful way, the software aims to facilitate a deeper understanding of accidents trends and contributing factors. Some potential benefits of the software include:

* The tools will enable users to quickly explore the dataset, filter relevant information and visualize accident data using charts and graphs.
* Through advanced data analysis techniques, the tool can uncover hidden patterns, correlations, and trends within the dataset, offering users valuable insights into road safety.
* Policymakers, researchers, and analysts can make evidence-based decisions for road safety improvements, resource allocation and accident prevention strategies.
* The software can potentially offer predictive capabilities, allowing users to anticipate accident hotspots, risk factors and patterns that might lead to accidents re-openings.
* By automating data analysis tasks and offering visualizations, the tools reduce the time and effort required for manual data interpretation.
* Ultimately, the software’s insights and recommendations could contribute to more effective road safety initiatives, leading to a reduction in accidents and then associated impacts.
* The tool will help acknowledge how much is the impact of drunk driving on the number of accidents happening in average.

# Requirements

## User Requirements

In this section, we describe how individual interact with the software from an end-user perspective. It sheds light on various tasks users should be able to perform and the capabilities they require from the software. For our project, we will envision a hypothetical user profile to guide the user requirements definition.

### 2.1.1 User interaction

#### User Profile:

Name: Anna Johnson

Role: Traffic safety Officer

Background: Holds a degree in urban planning and has been working with the local transportation department. Anna’s responsibilities include analysing accident data to identify road safety issues and propose improvement strategies.

User Requirements for the Data Analysis and Visualization Tool:

1. Dashboard Overview:

Upon logging in, Anna needs to be presented with a dashboard that provides an overview of recent accident trends and key statistics.

1. Custom time selection:

The user should be able to specify a time frame for analysis, such as a particular month or year.

1. Filtering parameters:

She should be able to filter accident data based on factors such as, time of the day, road types, vehicle types, and accident severity. Anna should be able to enter a keyword related to accident types (for example: “collision”, “pedestrian” to retrieve accidents of interest.)

1. View Accident Details:

She should be able to see the detailed information about the accident including date, time, accident type, location, the collision type, the type of vehicle and other data involved in the accident.

The information should be presented in an organised and easily accessible manner.

1. Analyse hourly trends:

The user needs to be able to generate a chart displaying the average number of accidents in each hour of the day within the selected period. She wants to view a time series plot that shows how accident frequencies have changes over the selected period.

1. Comparative analysis:

Anna needs the ability to compare accident data based on different factors, such as urban, vs rural or licensed or unlicensed.

1. Alcohol Impact Analysis:

Amma needs to be able to analyse the impact of alcohol consumption in accidents by exploring trends and accident types involving alcohol.

1. Interactive maps:

She needs to interact with maps that display accident locations, and hotspots in the chosen region and time frame. She should be able to zoom in and click on markers to view accident details.

1. Safety Recommendations:

Amma needs to get safety recommendations and preventative measures for different scenarios based on the analysis.

1. Customization and Export:

She should be able to customise the visualizations and reports by selecting specific filters and parameters. She also needs the ability to export the analysis results.

#### User interaction: -

1. Anna logs into the tools and is greeted with the dashboard overview.
2. She selects the state and time frame for analysis.
3. She applies filters based on her research interests, such as location and road types.
4. She is presented with the list of all the accidents that occurred within the time frame and the location that she selected for analysis.
5. She then selects “comparative analysis” for alcohol impact analysis and interacts with the map to explore accident locations.
6. She switches to the time series plot to analyse trends.
7. Anna compares accident data across different factors and views predicted accident occurrences.
8. She reviews safety recommendations provided by the tool.
9. Anna views the maps and make an analysis report and exports her report.

## Software Requirements

**R1. Data Selection and Filtering:**

* R1.1: The tool shall allow users to select a specific location for analysis.
* R1.2: Users shall be able to specify the time period for the analysis.
* R1.3: User shall be able to apply filters for weather conditions, road types, vehicle types, and accident severity.

**R2. Interactive Mapping:**

* R2.1: The tool shall provide interactive maps displaying accident locations, hotspots, clusters with the chosen location and time frame.
* R2.2: Users shall be able to zoom in and out of the maps for detailed analysis.
* R2.3: Users shall have the option to click on accident markers to view detailed information about each accident.

**R3. Time series visualization:**

* R3.1: The tools shall be able to produce the charts based on the average number of accidents that happen over the selected period.
* R3.2: Users should be able to adjust the time granularity for the time series plot.

**R4. Comparative analysis:**

* R4.1: The tool shall enable users to compare accident data across different factors, such as, urban vs rural areas, time of the day, weekday, or weekend, licensed or unlicensed.
* R4.2: Users will be able to analyse the affect of alcohol time on rate of accidents on daily basis.

**R5. Safety Recommendations:**

* R5.1: The tool shall be able to generate safety recommendations for improving road safety based on the analysis results.

**R6. Customisation and export:**

* R6.1: The users shall be able to customise the visualisations by adjusting filters and parameters within each section.
* R6.2: The tool shall provide options to export visualisations and analysis results in various formats, such as PDF and CSV.

**R7. User friendly interface:**

* R7.1: The tool’s interface shall be intuitive and easy to use and navigate providing a seamless user experience.

## Use Cases & Use Case Diagrams

**Use Case1: Display Accident Information:**

|  |  |
| --- | --- |
| **Use Case ID** | 1 |
| **Use case Name** | Display Accident Information |
| **Actors** | Traffic Safety Analyst |
| **Description** | This use case outlines how the user can view detailed information about all accidents that occurred within a user-selected period. |
| **Flow of events:** | 1. The Traffic Safety Analyst accesses the Accident Analysis Tool through a secure login. 2. The system validates the user's credentials, granting authorized access to the tool's features. 3. The user specifies the geographical location (e.g., city, region, intersection) and time period for focusing the analysis. 4. After setting the analysis parameters, the user navigates to the dedicated "Accident Information" section. 5. The user's selection triggers the system to query the accident database based on the defined parameters. 6. The system compiles a comprehensive list of accidents that occurred within the specified time period and at the chosen location. The list includes key accident attributes such as date, time, location, severity, and involved parties. 7. User can click on an accident entry to view more detailed information:  * For each entry in the displayed accident list, the user has the option to click on a specific accident to access more detailed information. * The detailed view provides a comprehensive overview of the accident, including: involved vehicles and parties, description of the accident scenario, and contributing factors (e.g., weather conditions, road conditions) |

Top of Form

**User Case 2: Time based Accident Frequency Chart:**

|  |  |
| --- | --- |
| **Use Case ID** | 2 |
| **Use case Name** | Time based Accident Frequency Chart |
| **Actors** | Road safety officer |
| **Description** | This use case involves the analysis of accident trends over a specific time period using the Victoria State Accident dataset. The road safety officer utilizes the system to gain insights into accident occurrences throughout the day, helping to identify patterns and trends that can inform safety measures and policies. |
| **Flow of events:** | 1. The Road Safety Officer logs into the system using authorized credentials. 2. The officer selects a specific location for analysis and sets a desired time period using a date-picker. 3. From the dashboard menu, the officer chooses the "Time Trends" option to start the time-based accident frequency analysis. 4. The system processes the input and generates a time series plot showing accident frequencies over the selected time period. This provides a visual view of trends in accident occurrences. 5. Additionally, the system calculates and presents a chart that displays the average number of accidents for each hour of the day. This chart helps in understanding accident distribution throughout the day and identifying peak accident hours. 6. The officer can adjust the time granularity, choosing to view data on hourly, daily, or weekly basis for more detailed analysis. 7. The officer interprets the charts to identify patterns and trends in accident occurrences, including peak hours and recurring patterns. These insights inform decisions on road safety measures and policies. |

**User Case 3: Alcohol Impact analysis:**

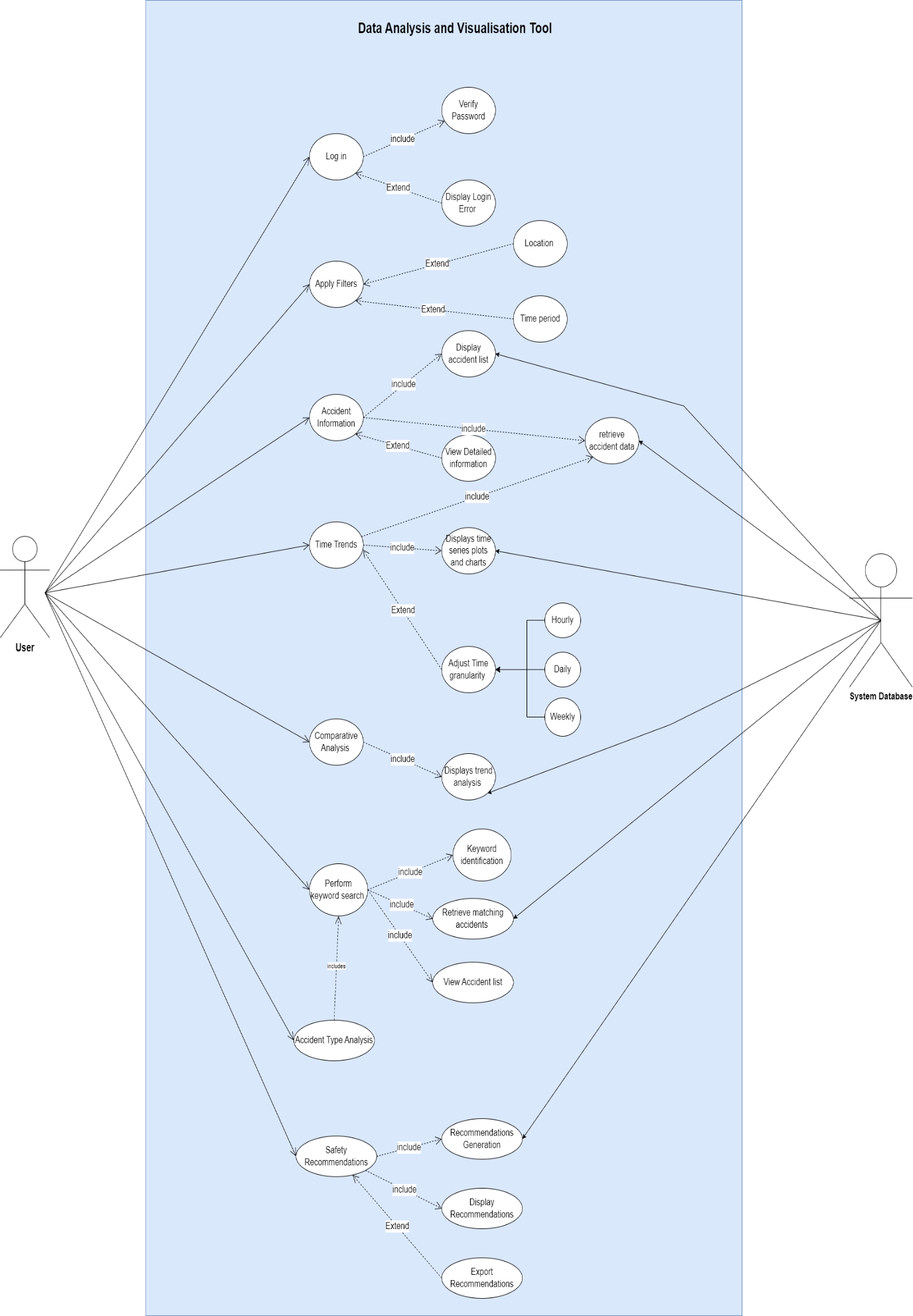
|  |  |
| --- | --- |
| **Use Case ID** | 3 |
| **Use case Name** | Alcohol Impact analysis |
| **Actors** | Road safety Manager |
| **Description** | The Road Safety Manager compares road accidents involving drunk driving with those not involving alcohol to identify patterns and correlations. |
| **Flow of events:** | 1. The Road Safety Manager logs into the system. 2. The manager selects the desired location and time frame for the analysis. 3. Filters are applied to focus on incidents related to alcohol. 4. The "Comparative Analysis" option is clicked. 5. The system presents a comparison of accident data, including:   - Trend analysis of alcohol-related accidents over time.  - Visualizations of accident types frequently involving alcohol.   1. The Road Safety Manager interprets the visualizations to identify correlations and patterns related to alcohol-related accidents. |

**User case 4: Generate Safety Recommendations:**

|  |  |
| --- | --- |
| **Use Case ID** | 4 |
| **Use case Name** | Generate Safety Recommendations |
| **Actors** | Transportation Planner |
| **Description** | The user generates safety recommendations based on analysis results. |
| **Flow of events:** | 1. The user logs in to the analysis tool. 2. With the system interface open, the planner specifies the location, time period for targeted analysis. 3. User selects the relevant filters like, road type, speed limit, to narrow down the analysis. 4. System process analysis data and generates safety recommendations. 5. The planner navigated through the recommendations, viewing detailed explanations and supporting data. 6. The planner then exports the recommendations for integration into official safety reports or action plans. |

**User Case 5: Search by Accident Type keyword:**

|  |  |
| --- | --- |
| **Use Case ID** | 5 |
| **Use case Name** | Search by Accident Type keyword |
| **Actors** | Researchers and analysts |
| **Description** | This use case illustrates how Researchers and Analysts can retrieve accidents caused by a specific accident type containing a user-entered keyword. This capability assists in targeted analysis and understanding of specific accident scenarios. |
| **Flow of events:** | 1. The user logs in to the analysis tool. 2. With the system interface open, the planner specifies the location, time period for targeted analysis. 3. User enters the relevant filters in the search bar, such as, “collision”, “pedestrian”, “daylight”, “urban”, “public transport”, etc. 4. The user initiates the search by navigating to the dedicated “Accident Type Analysis” section within the tool. 5. The system provides a specialised environment for advanced accident type analysis. 6. The system scans through the accident type descriptions to identify keywords and patterns. 7. The system retrieves and displays a comprehensive list of accidents that match the entered keywords in their accident type descriptions. 8. The user views the listed accidents that includes the attributes such as date, time, location, severity, and more. |



# Software Design and System Components

## Software Design

The following is the flowchart of how the software might work: -

A diagram of a diagram

Description automatically generated

## System Components

### Functions

There are several function that will be used in the software. In this section, a brief description of the functions including their input parameters, list of side effects and the return value of the function has been mentioned. The following is the list of all the functions in the software: -

1. loadData Function: -

* Description: The loadData function is a critical component of the tool responsible for importing accident data from the provided dataset. It initializes and populates the tool's internal data structures with the accident data required for analysis.
* Input Parameters:
* File Path (string): The File Path parameter specifies the location of the dataset file to be loaded. It serves as the source from which accident data is extracted.
* Side effects: This function reads and processes the dataset, converting it into a format suitable for analysis. It may trigger changes in the tool's memory as it stores the loaded data for subsequent functions.
* Return Value: The loadData function does not return any specific value; instead, it initializes the tool's data storage.

1. displayDashboard Function: -

* Description: displayDashboard is the entry point for users, presenting them with an informative dashboard upon login. This function offers a comprehensive view of recent accident trends and key statistics.
* Input Parameters: None
* Side effects: It interacts with the user interface to render the dashboard, displaying visualizations and statistics derived from the loaded accident data. It provides users with an initial overview of the dataset.
* Return Value: None

1. setAnalysisParameters Function: -

* Description: setAnalysisParameters empowers users to define the scope of their analysis. Users can specify location, time frame, and apply filters to narrow down the dataset for focused examination.
* Input Parameters:
* Location (string): The Location parameter allows users to specify a geographical location, such as a city or region, to restrict the analysis to a specific area.
* Time Period (start\_date, end\_date): The Time Period parameters define the temporal scope by specifying a start and end date for analysis.
* Filters (dictionary): The Filters parameter is a dictionary that includes user-defined filters for factors like alcohol impact, road types, regional analysis, and accident severity.
* Side effects: This function updates the tool's analysis parameters based on user input, ensuring that subsequent analyses are tailored to the user's requirements.
* Return Value: None

1. displayAccidentList Function: -

* Description: displayAccidentList presents users with a list of accidents that meet the specified criteria based on location, time frame, and applied filters.
* Input Parameters: None.
* Side Effects: It retrieves and displays a list of accidents that align with the user's selected parameters, providing users with a detailed view of accidents that match their criteria.
* Return Value: The function returns a list of accident data that meets the selected parameters, including details such as date, time, location, severity, and involved parties.

1. generateTimeSeriesChart Function:

* Description: generateTimeSeriesChart creates a time series chart that visualizes accident frequencies over the selected time period.
* Input Parameters: None.
* Side Effects: This function computes and displays a time series chart, enabling users to identify trends and patterns in accident occurrences over time. Users can gain insights into temporal accident distribution.
* Return Value: The function returns data or a visualization representing accident frequencies over the selected time period. Users can customize the time granularity for more detailed analysis.

1. compareAccidentData Function:

* Description: compareAccidentData facilitates the comparison of accident data across different factors, such as urban vs. rural areas, time of day, weekdays vs. weekends, and licensed vs. unlicensed driving.
* Input Parameters:
* Comparison Factors (list): Users specify a list of factors they wish to compare accidents on, allowing for insights into how various factors influence accident occurrences.
* Side Effects: The function performs comparative analysis based on the selected factors and presents users with visualizations or insights into the differences in accident patterns.
* Return Value: The function returns data or visualizations that enable users to compare accident data across the specified factors.

1. customizeAndExport Function:

* Description: customizeAndExport empowers users to customize visualizations and export analysis results in their preferred format, such as PDF or CSV.
* Input Parameters:
* Customization Parameters (dictionary): Users can adjust filters and parameters within different sections to customize visualizations.
* Export Format (string): Users specify the format in which they want to export the analysis results, facilitating integration into reports or action plans.
* Side Effects: This function enables users to tailor visualizations to their specific needs and export them for external use. It ensures flexibility and ease of data sharing.
* Return Value: The function may return the exported data or visualizations in the format requested by the user.

1. generateSafetyRecommendations Function:

* Description: generateSafetyRecommendations generates safety recommendations based on the analysis results, helping users identify measures to improve road safety.
* Input Parameters: None.
* Side Effects: The function analyzes the data and derives actionable safety recommendations that users can consider for implementation. It contributes to informed decision-making regarding road safety.
* Return Value: The function returns safety recommendations tailored to the analysis results.

### Data Structures / Data Sources

Within this tool, a range of data structures and data sources have to be designed and integrated to fulfill these vital functions. In this section, we will delve into the key data structures and data sources that power our software, elucidating their roles, structures, and interactions.

1. **Data structure 1: Accident data**

* Type of structure: list or array
* Description: The Accident data structure represents individual accident records in our dataset. It is instrumental in storing and manipulating information about each accident including time, data, location, severity, vehicle type, speed zone, and many more.
* Data members:
* Date (string): Stores the date of the accident.
* Time (string): Holds the time of the accident.
* Location (string): Represents the accident's location.
* Severity (string): Indicates the accident's severity.
* Vehicle Type (string): Specifies the type of vehicle involved.
* Other relevant attributes as per dataset.
* Functions that Use It:
* loadData: Populates instances of this class with accident records.
* Functions related to displaying, analyzing, and exporting accident data utilize instances of this class.
* Other functions that use it are generateSafetyRecommendations, customizeAndExport, compareAccidentData, generateTimeSeriesChart, and displayAccidentList Functions.

1. **Analysis Parameters Data Structure**

* Type of Structure: Dictionary
* Description: This data structure is used to hold the parameters set by the user for analysis, including location, time period, and various filters.
* Data Members:
* Location (string): The geographical location for analysis.
* Time Period (start\_date, end\_date): The time frame for analysis.
* Filters (dictionary): User-defined filters for various factors like weather conditions, road types, vehicle types, and accident severity.
* Functions that Use It: setAnalysisParameters Function, loadData Function.

1. **Time Series Data Structure:**

* Type of Structure: List or Dictionary (depending on implementation)
* Description: This data structure is used to store accident frequency data over time for generating time series charts.
* Data Members:
* Time (string): The time intervals (e.g., hours) over the selected time period.
* Accident Frequency (int): The number of accidents during each time interval.
* Functions that Use It: generateTimeSeriesChart Function.

1. **Comparative Analysis Data Structure**

* Type of Structure: List or Dictionary (depending on implementation)
* Description: This data structure is used to organize and store data for comparative analysis across different factors, such as urban vs. rural areas, time of day, and other user-defined criteria.
* Data Members: Varies based on the factors being compared.
* Functions that Use It: compareAccidentData Function.

1. **Analysis Results Data (Structure/External Source):**

* Type: Data structure (e.g., list, array, or dictionary) or external data source (e.g., database)
* Description: This structure holds the results of various analyses, such as time series data, comparative analysis, and safety recommendations.
* Data Members: N/A (Data is stored based on specific analysis results)
* Functions that Use It: Functions like generateTimeSeriesChart, compareAccidentData, and

generateSafetyRecommendations populate and return results stored in this structure.

1. **Analysis Parameters (Class):**

* Type: Class
* Description: This class stores user-defined analysis parameters, including location, time frame, and filters.
* Data Members:
* Location (string): Stores the selected geographical location for analysis.
* Time Period (start\_date, end\_date): Defines the temporal scope for analysis.
* Filters (dictionary): Contains user-specified filters for analysis factors (e.g., weather conditions, road types).
* Functions that Use It: Functions like setAnalysisParameters utilize an instance of this class to manage user input for analysis customization.

### Detailed Design

In this section, we present pseudocode for the algorithms that drive our data analysis and visualisation processes. Thes algorithms are critical to generate meaningful insights from accident data.

**Algorithm 1: Data Loading and Preparation**

Description: This algorithm loads accident data from external sources, transforms it into the appropriate data structures, and prepares it for analysis.

Pseudocode:

Function loadData:

Input: Data source (e.g., CSV file)

Output: Accident Data (List of Accident Data objects)

Initialize empty list AccidentDataList

For each record in Data source:

Create a new instance of AccidentData

Populate AccidentData attributes with record values

Append AccidentData to AccidentDataList

Return AccidentDataList

**Algorithm 2: Time Series Chart Generation**

Description: This algorithm generates time series charts to visualize accident trends over selected time periods.

Pseudocode:

Function generateTimeSeriesChart:

Input: Accident Data (List of Accident Data objects), Time Period

Output: Time Series Chart

Initialize empty Time Series Chart

Group Accident Data by time intervals within the Time Period

For each time interval group:

Calculate the number of accidents in that interval

Add data point to the Time Series Chart

Return Time Series Chart

**Algorithm 3: Safety Recommendations Generation**

Description: This algorithm generates safety recommendations based on the analysis results, aiming to improve road safety.

Pseudocode:

Function generateSafetyRecommendations:

Input: Comparative Analysis Results, Analysis Parameters

Output: Safety Recommendations

Initialize empty Safety Recommendations

Analyze Comparative Analysis Results and identify areas for improvement

Formulate safety recommendations and preventative measures

Return Safety Recommendations

# User Interface Design

The user interface (UI) design is a pivotal phase in the development of our data analysis and visualization tool for the Victoria State Accident Dataset. This section provides an overview of the initial interface design, outlining the tools and methodologies employed during this crucial stage of development. The wireframing tools that are used to create low-fidelity representations of the UI are Draw.io and Figma. Wireframes served as the foundation upon which the user-friendly interface is built.

The following sub-sections will contain the following:-

1. Information Hierarchy: It is important to present the information in a structured and hierarchical manner. Thus, we have carefully determined which data elements should be prominently displayed on the dashboard and how to organize them for easy access.
2. Customization and Flexibility: There are customization options for the users to tailor the interface to their specific analysis needs such as customizable filters, search options, time period selection, and export options.
3. Data Visualization: The UI design incorporates charts, graphs, and interactive maps to visually represent accident data and trends.
4. Intuitive Navigation: There is a well designed a menu structure that guides users seamlessly through various analysis sections while ensuring a user-friendly experience.

In the subsequent sub-sections, we will delve deeper into the structural design of the interface, providing detailed insights into the layout, components, and interactions that constitute our user-centric design. Each aspect has been carefully crafted to align with our users' needs and feedback, ultimately aiming to empower them to analyse accident data efficiently and make informed decisions for road safety.

## Structural Design

In this section, we will delve into the structural design of the user interface, using a hierarchy chart. we will delve into the structural design of the user interface. It will outline the key screens and components of the user interface, describe their functions and interactions, and provide insights into how they contribute to creating an effective and user centric software experience.

Our user interface comprises seven distinct screens, offering straightforward navigation for an intuitive user experience. The interface is organized hierarchically, with the "Search", "Analysis", and "Settings" options accessible directly from the home page, also referred to as the "Dashboard." Each of these screens fulfills unique tasks and functionalities.

**Hierarchical Navigation:**

A screenshot of a computer

Description automatically generated

Home Page / Dashboard: The central hub of the interface, accessible upon login, provides access to key functionalities.

Search option: This option takes the user to a different screen where the user can initiate searches and retrieve specific data sets.

Analysis: Users can explore a range of data analysis options through this, which offers a variety of tools and insights.

* **Exploring Data Analysis Options**

Upon selecting the "Data Analysis" option, users encounter a set of distinct options for further navigation. Each option leads to a unique screen containing relevant specific information, interactive maps, and tailored recommendations.

* **Automated Report Generation**

On the bottom right corner of each analysis screen, users will find a convenient "generate report" button. This feature streamlines the process of report creation, automatically generating reports based on the selected filters. Users can then easily export these reports to their preferred destination.

This structured interface design simplifies user interaction and enhances usability, ensuring that users can seamlessly access and analyze accident data while also facilitating the efficient generation of customized reports.Top of Form

## Visual Design

**Login/Signup Screen:**

* Layout: Traditional layout with input fields for username and password. Includes buttons for login, signup, and a "Forgot Password?" link.
* Visual Elements: Clean design with clear labels. Minimal use of icons, mainly for buttons.
* Icons/Graphics: Minimal use of icons, mainly for buttons.
* Style: Clean and minimalist style to maintain focus on functionality.

**Dashboard:**

* Layout: Organized layout with real-time accident charts and navigation options (home, analysis, search, settings).
* Visual Elements: Prominently features interactive charts. Navigation icons for home, analysis, search, and settings.
* Icons/Graphics: Charts and navigation icons.
* Style: Modern and data-centric, with the focus on presenting real-time accident data.

**Settings Dialog:**

* Layout: Pop-up dialog box with radio button options and a save button.
* Visual Elements: Radio buttons for user preferences.
* Icons/Graphics: Minimal use of icons.
* Style: Functional and straightforward for user preferences.

**Search Screen:**

* Layout: Separate screen with a search bar and text paragraphs.
* Visual Elements: A prominently placed search bar with clear labels.
* Icons/Graphics: Minimal use of icons.
* Style: Clean and organized for efficient searching.

**Analysis Options Dropdown:**

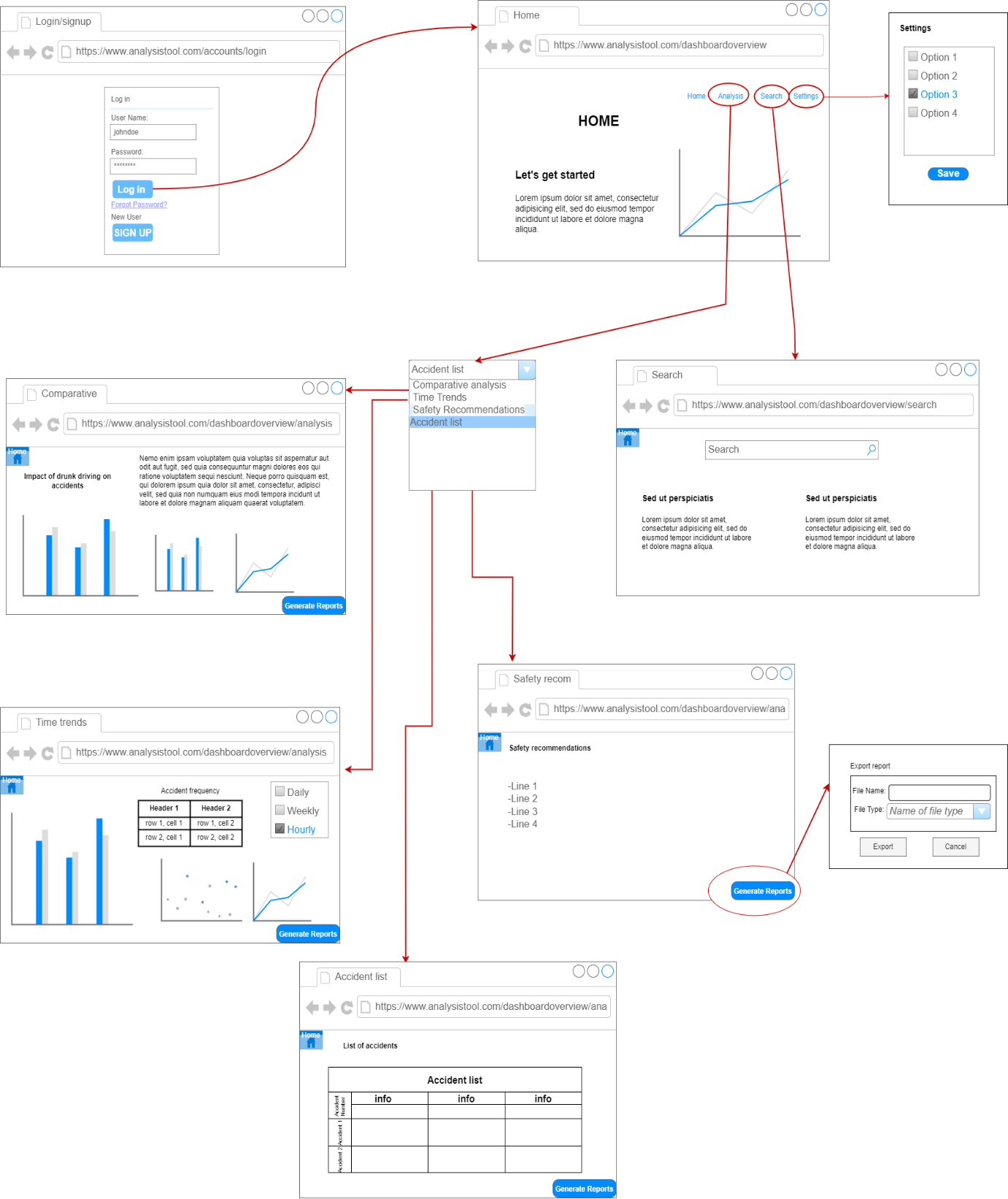
* Layout: Drop-down menu with four options: Comparative analysis, time trends, accident details, and safety recommendations.
* Visual Elements: Clear menu items with labels.
* Icons/Graphics: Icon or indicator for the drop-down menu.
* Style: Functional and concise for quick access to analysis options.

**Screens for Analysis Options:**

* Comparative Analysis: Clean layout with bar charts and line graphs.
* Time Trends: Table for accident frequency, time granularity selector, and time plots with line and bar graphs.
* Safety Recommendations: List of recommendations.
* Accident List: Table with comprehensive accident details.

**Generate Reports Button**:

* Layout: Located in the bottom right corner of each screen.
* Visual Elements: Clearly labeled button.
* Icons/Graphics: Minimalist export icon.
* Style: Unobtrusive and functional.



The user interface comprises five screens which each require specialized designs to provide an optimal user experience. Hence, five different wireframes have been meticulously drawn to accommodate the necessary elements and functionalities. Furthermore, these wireframes follow consistent and cohesive design choices with same colour scheme, styling, and fonts to maintain uniformity, seamlessness, and familiarity while interacting with the user interface.

The user interface will follow this colour palette where the monochromatic colours provide a minimalistic vibe and the light sky-blue colour highlights the navigation bar. The Montserrat font will be used as it provides a clear and modern appearance.

# Conclusion

In conclusion, our project endeavours to develop a user-friendly and efficient data analysis and visualization tool tailored to the Victoria State Accident Dataset. This tool will empower users to explore and understand road safety data comprehensively, enabling informed decision-making by researchers and policymakers. With a structured user interface, interactive features, and thoughtful design choices, our system aims to make the analysis process more accessible and effective.