Software Design Document

<Victorian Crash Sight>

Jack Brighton - S5184901

Nicholas Webster - S5178735

**Table of Contents**

[1.0 System Vision 3](#_heading=h.gjdgxs)

[1.1 Problem Background 3](#_heading=h.30j0zll)

[1.2 System Overview 3](#_heading=h.1fob9te)

[1.3 Potential Benefits 3](#_heading=h.3znysh7)

[2.0 Requirements 4](#_heading=h.2et92p0)

[2.1 User Requirements \*\* more here (functional/non-functional requirements) 4](#_heading=h.tyjcwt)

[2.2 Software Requirements 4](#_heading=)

[2.3 Use Cases & Use Case Diagrams 5](#_heading=h.1t3h5sf)

[2.3.1 Use Case 1: Senator looking to Improve Local Road Safety. 5](#_heading=h.o6who6mqwc7k)

[2.3.2 Use Case 2: Motorcyclist Travelling Across Victoria 6](#_heading=h.upxsok6hwi6z)

[3.0 Software Design and System Components 8](#_heading=h.4d34og8)

[3.1 Software Design 8](#_heading=)

[3.2 System Components 8](#_heading=)

[3.2.1 Functions 8](#_heading=)

[3.2.2 Data Structures / Data Sources 9](#_heading=)

[3.2.3 Detailed Design 9](#_heading=)

[4.0 User Interface Design 10](#_heading=h.2s8eyo1)

[4.1 Structural Design and Visual Design 10](#_heading=)

[5.0 References: 12](#_heading=h.jq5tj7y0gsy5)

# System Vision

## Problem Background

In the context of road safety in Victoria, the reduction of road-related fatalities over the past decade is encouraging (~10%). However, the improvements are not seen between motorists and vulnerable road users like pedestrians, cyclists and motorcyclists, as they face much higher risks. These groups make up a disproportionate amount of low-speed injuries, particularly within densely populated areas such as major cities. This continuing vulnerability highlights the need for a focused approach to improving road safety strategies. The introduction of a tool which can provide a comprehensive understanding of accident patterns, will aid legislators by providing actionable information to identify then improve safety for these parties within at-risk areas.

## System Overview

The proposed system is an interactive visualisation tool designed to provide a user-friendly platform for analysing and understanding road accident data in Victoria. The proof-of-concept build will be a self-contained system, using provided datasets for comprehensive data analysis and visualisation. The core system is an intuitive user interface with sorting features such as filtering, interactive maps, and graphs to present data by the user’s specification. This system will be specifically designed to utilise road accident data and will not be operable when using other datasets.

When the user opens the application they will have options to filter the dataset to their requirements. They will have options to filter by selected time period, hour, localisation, and alcohol influence. This data will be searched for in the dataset and presented to the user via multiple tools such as graphs or tabulated data. This data can be sorted in ascending orders depending on the users preference.

The system will be designed specifically for computer usage. The system will not be compatible with mobile devices, but will work across operating systems (windows and iOS). The basic hardware requirements to run this program will be minimal. The system should have a i5 processor, which is the most common kind of processor in modern computers. A visual display and a keyboard will be required to search and analyse the data.

## Potential Benefits

There are multiple benefits to this new system. It offers a user-friendly platform to analyse road accident data in Victoria. By providing comprehensive insights into accident patterns through intuitive graphs, maps, and other filters, policymakers can focus on vulnerable road users and extract relevant data to better road safety for these people. Its information leads to evidence-based decision-making, identification of high-risk areas, efficient resource allocation and long-term trend analysis. The benefits of this system can be a better informed public and safer roads for those who use them.

# Requirements

## User Requirements

The system is being designed as a proof-of-concept for legislators and manufacturers. These clients would not necessarily be interested in a large amount of the data but will need specifics when it comes to the context clues given in the data, they are more interested in the most pertinent information. For these clients there are a set of details that they will need, which are listed below. Depending on the client, some of these requirements are not as pertinent and are reflected in the use cases in 2.3.1 & 2.3.2.

* Accident date
* Accident time
* Alcohol Influence
* Accident Type
* Severity
* Speed Zones
* Location
* Involved Party Keywords (Pedestrian, Bicycle, etc)

The filtered data here will be best used to give the interested stakeholders context clues around what happened. As an assistance tool it has great value in data analytics alongside user stories. These should be intuitive to gain access to and present itself in a visually appealing and easy to understand way. If successful the system should be able to utilise this information for better legislative practice.

## Software Requirements

The purpose of this project is to allow a client / user to interact with the given data set and analyse the pertinent data. For our project to be effective, we need to meet a certain set of functional requirements, these functionalities are listed below.

R1.1 The program will read from a CSV file of a defined name within the program files.

R1.2 The program will store the data in a dataframe which can be created & filtered efficiently via the pandas python package.

R1.3 The program will filter the dataframe to the users specification - most commonly to the below list.

* Selected time period
* Alcohol Influence
* Accident Keywords
* Location
* Average Accident Hour
* Severity

R1.4 The program will filter the dataframe to any combination of the before mentioned list.

R1.5 The program will be completely unit and user tested, ensuring that the same data is returned on multiple re-queries.

R1.6 The program will load within a reasonable time of under 5 seconds from user query.

R2.1 The program will have a user interface which is user-friendly and intuitive. This will have an area which presents tabulated data, and will generate graphical representations on user request.

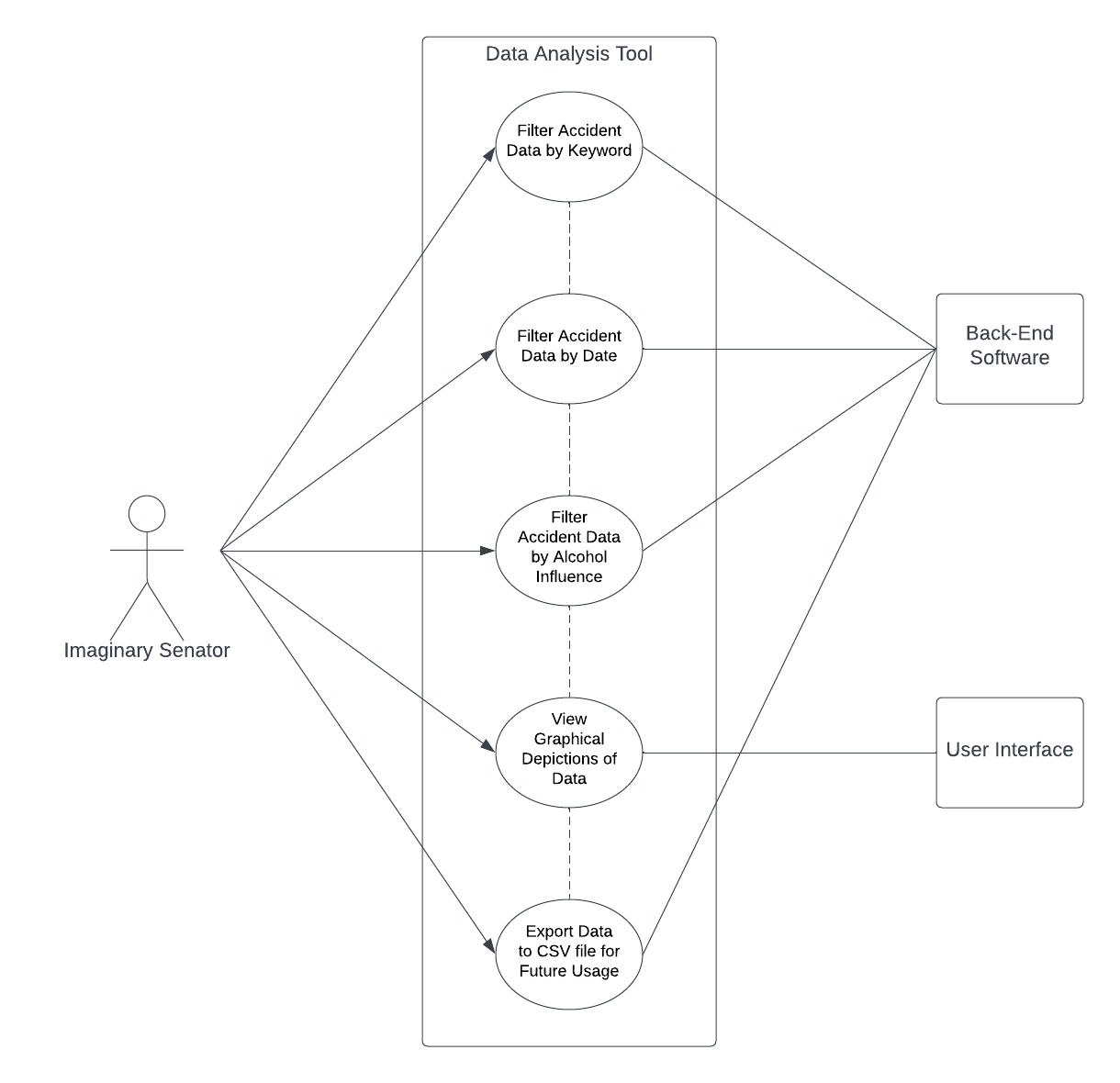
R3.1 The program on user request will produce a map view of victoria showing a heatmap of accidents within that location. The deeper the colour in a region the more accidents present.

## Use Cases & Use Case Diagrams

The primary stakeholders interested in our proof-of-concept project are legislators and manufacturers, but a secondary potential user could be vulnerable commuters. Legislators and manufacturers are policy makers in a sense but with vastly different levels of lawful influence. Conveniently, both of their interactions will be similar, being interested in slightly different information so having slightly different use cases. The vulnerable commuters would be interested in location and severity information, so they know what areas to avoid while commuting.

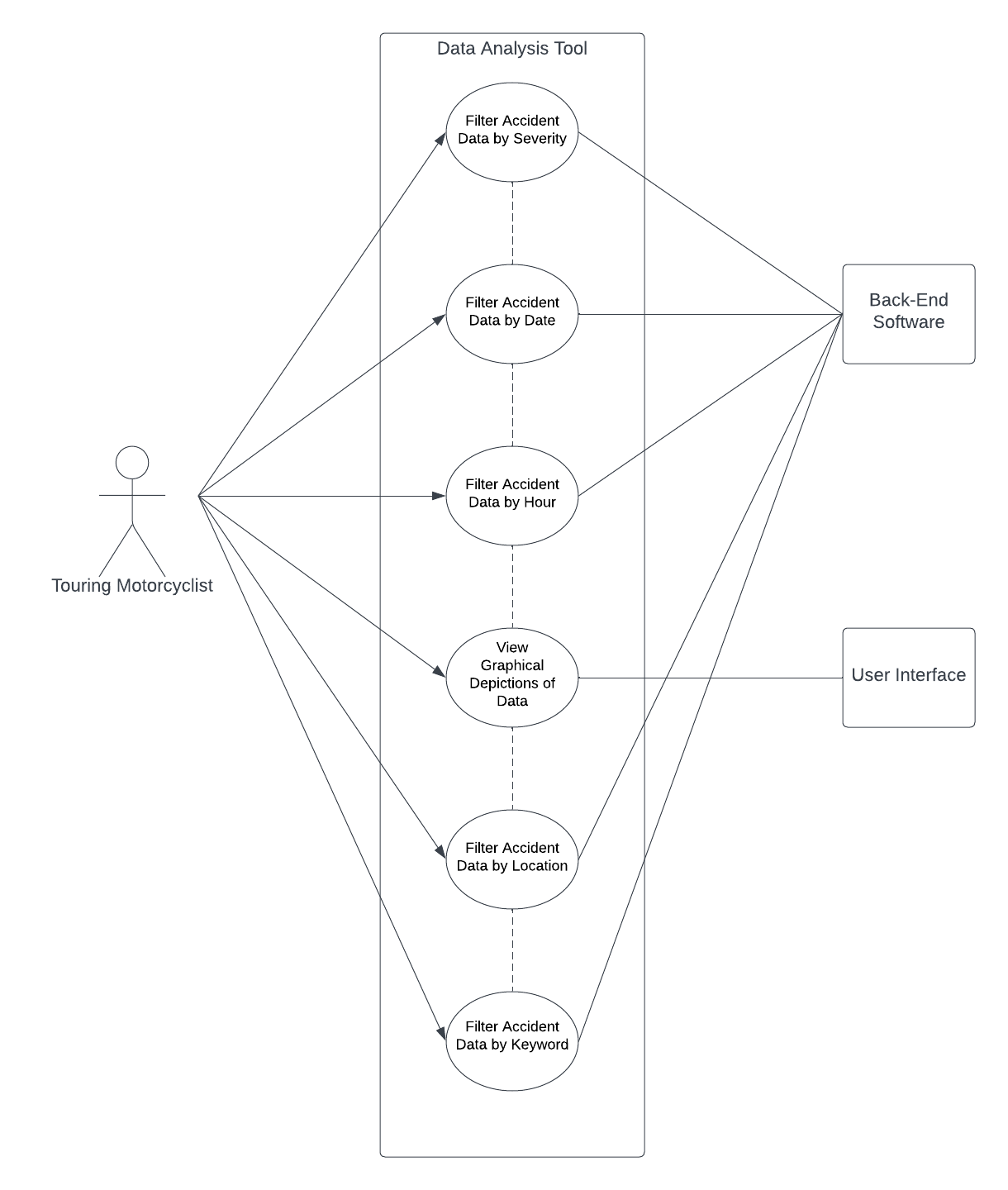
### 2.3.1 Use Case 1: Senator looking to Improve Local Road Safety.

An imaginary senator within Victoria wants to improve the road safety in their area. They need actionable information to bring to a hearing and to be the basis of their arguments. They want a strong grasp on the relevant information, and need visual tools to better aid their understanding. The breakdown of use-cases for this person are below. Notably there is some information that the senator would not be interested in such as hourly data or a full city breakdown. They are interested in general data for their location.



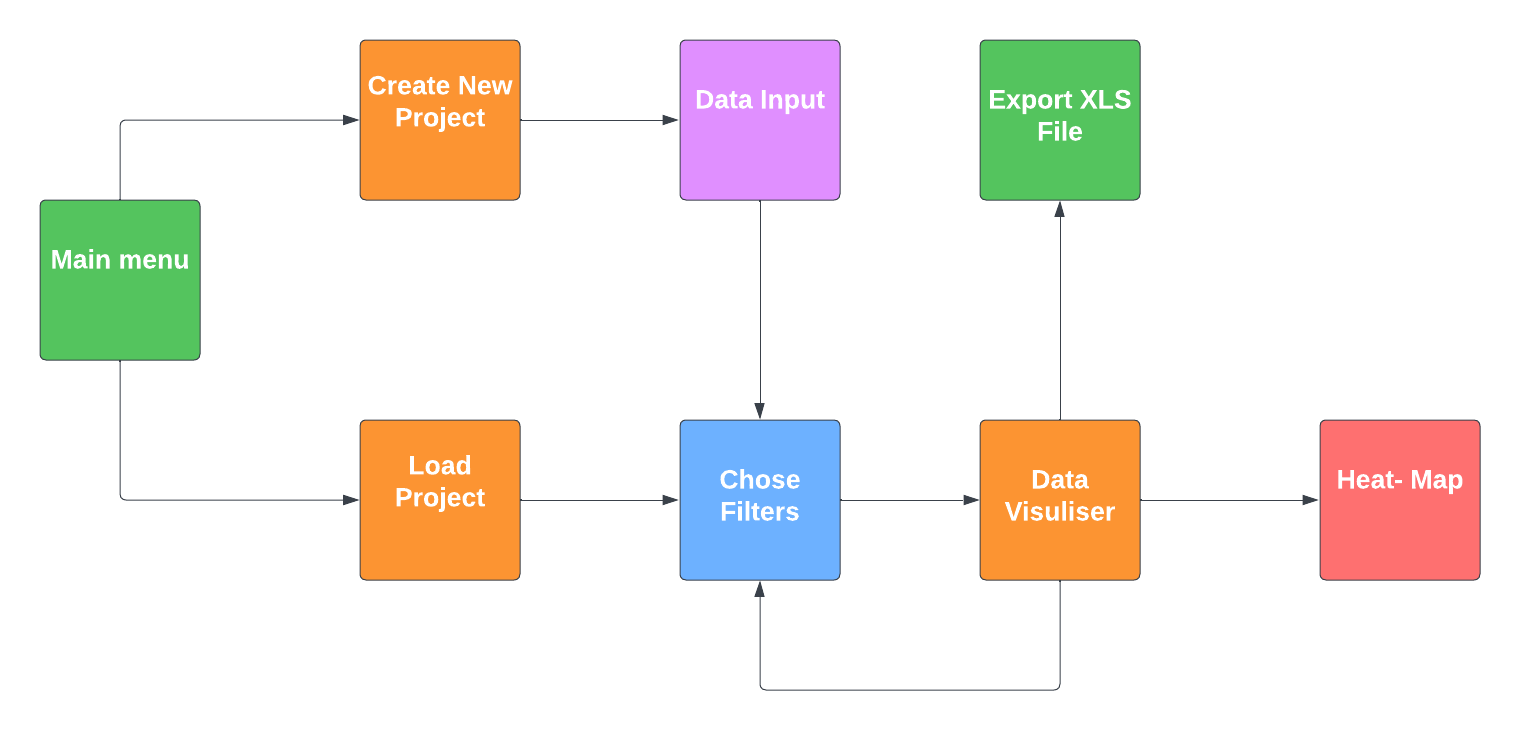
### 2.3.2 Use Case 2: Motorcyclist Travelling Across Victoria

A motorcyclist is on a tour across Victoria and intends to pass through much of its local regions. They are worried about their safety in general since they believe drivers are not vigilant the closer they are into the city. They want to understand what areas they should avoid. They’ll be interested in data such as accident severity, specifically motorcycle accidents, locations they’ll be passing through, hours they should avoid on the road, and they will need a graphical representation of the data to better help their understanding.



# Software Design and System Components

## Software Design



## System Components

### Functions

addDataPoint - this function adds a certain data point that is selected by the end user for the heatmap. The input arguments would be whether or not the data is included so a bool is suitable. And the actual data depends on what is selected so if a data is produced an integer would be used. But if a location is selected a floating number would be used to represent the longitude and latitude. The function would return the required data points that pandas would then extract and turn into usable data.

plot - this function handles all of the plotting of data points from location to time and date. this would use the pandas library to help with data extraction from the excel file. This function would be used to generate the heatmap of crashes in Victoria.

findPoints- this function would allow the end user to search through the data points assembled by pandas to find particular entries. Input parameters would be wide as they would need to cover different types of data entries but a search by their name. the function would return the matching data points.

averageDataHour — this function searches for the time of crash and tallies it with other crashes in that hour, then averages them out and returns a dictionary of times and average amount of crashes. the input parameters would be the select times of crashes selected as integers.

getAlcoholInvolved—this function searches for if the crash has alcohol involved, the input is the data frame that is initialised at the start of the program it returns a new filtered data frame of crashes in which alcohol is involved

### Data Structures / Data Sources

We are using data frames which are handled by pandas, these data frames consist of arrays and dictionaries. These are used to search through the excel data and allow us to create functions that do data analysis. Data frames are used in pandas to represent data in a tabular format, they will store and manipulate crash statistics. The data members of the data frame will be the columns which will represent different attributes in the crash data e.g. Location, Date, Number of cars involved. Another Member would be the rows which indicate the specific crash. Some functions that will be used are .head() and .tail() to display the top and bottom of each column.

Lists are also used in the program to store data structures, because they are flexible more so than data frames. You can .append() and .extend() them to add elements to a list as well as .pop() and .remove() to remove them from the list. Lists can be easily counted either by occurrences of the same data point with .index() or you can count the whole list with .count(). This will be useful to comprehend lists in this program.

Dictionaries will be stored to store values with a key value, this will be important for functions such as getDateHour which finds crashes per hour. The data members for this are the keys and values. Keys uniquely identify data points and the values are the data itself. .keys() will be used to find specific keys and .value() will be used to get a value from a key.

### Detailed Design

getAlcoholInvolved() - search columns for alcoholInvolvement -> returns dictionary of accident ID and the alcoholInvolvement

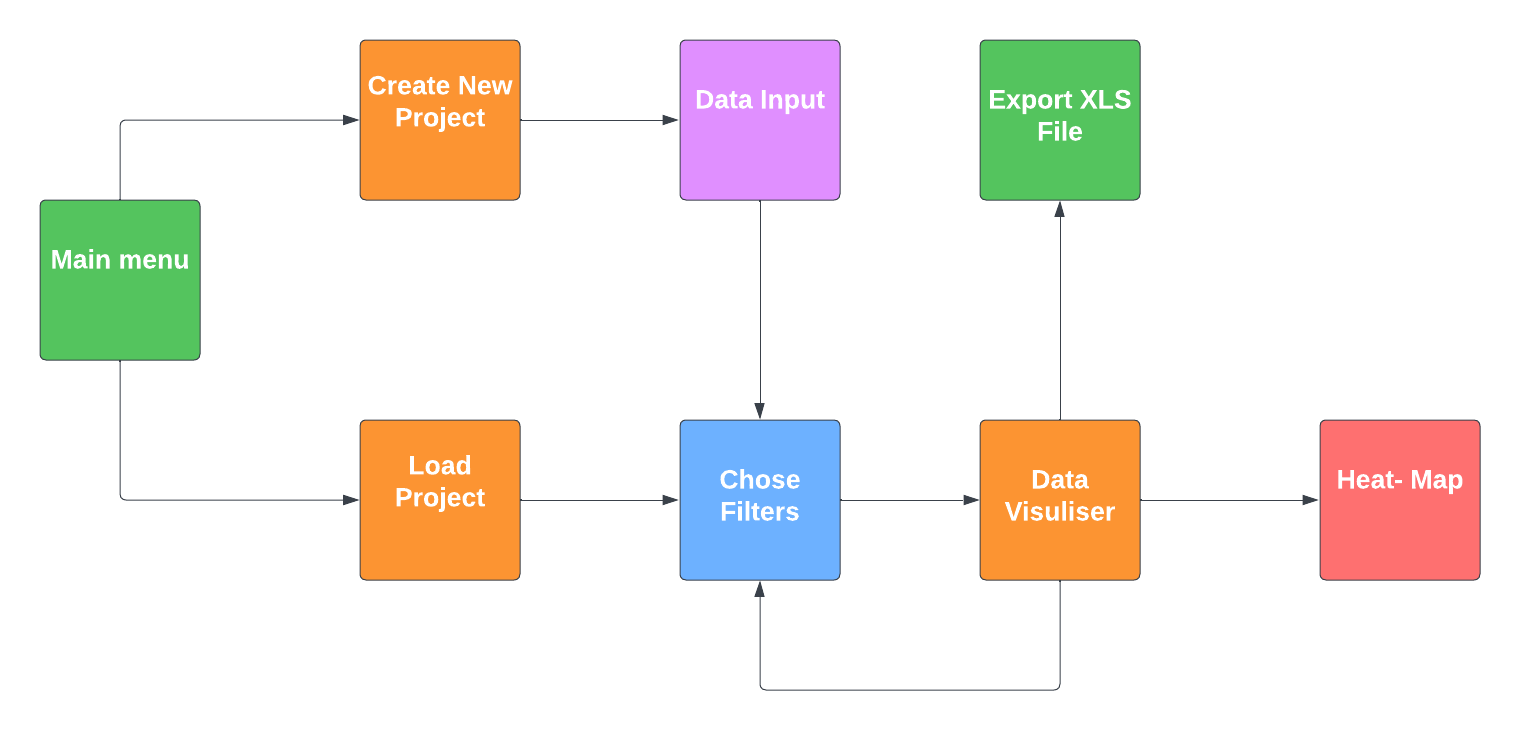
averageDataHour()- search columns for start and end time -> make dictionary of counts of crashes at certain time -> add crashes to value, and make sure its added to appropriate key -> then it calculate total crashes and then divides by the amount of time to get average crashes per hour between two times.

# User Interface Design

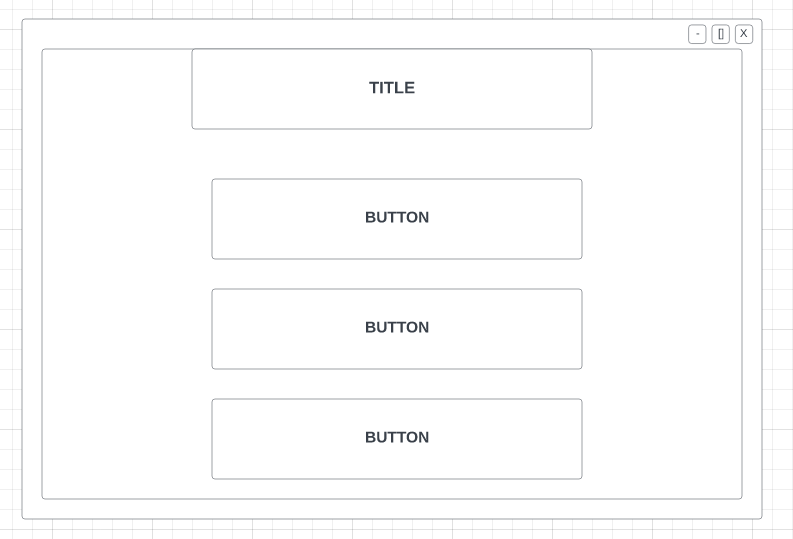
The tools used for the initial design phase of the program were python library wxPython, this was used to get a rough idea of the structure of how we would like the program to operate. Mockup Ui were brainstormed using figma and Lucidchart. we have designed a simplistic UI that focuses on readability and legibility of the data which allows the end user to have a streamlined experience and does not produce excess data points and information overwhelm.

## Structural Design and Visual Design

A sequential ordering of menus was the original concept of the team to approach this project, this was because the program we are designing is a tool used to convey data. So the steps to display the data should be simple and straightforward for the end user to follow. The original concept is as follows.



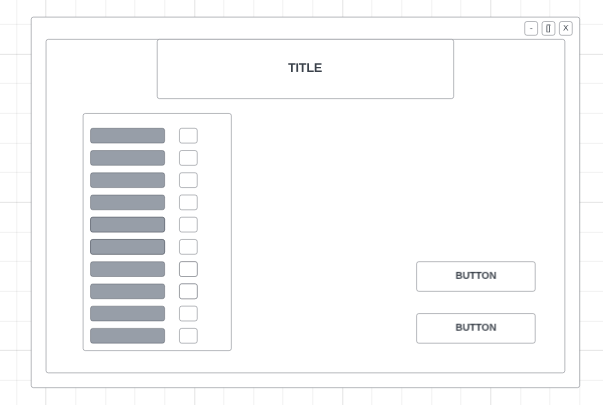
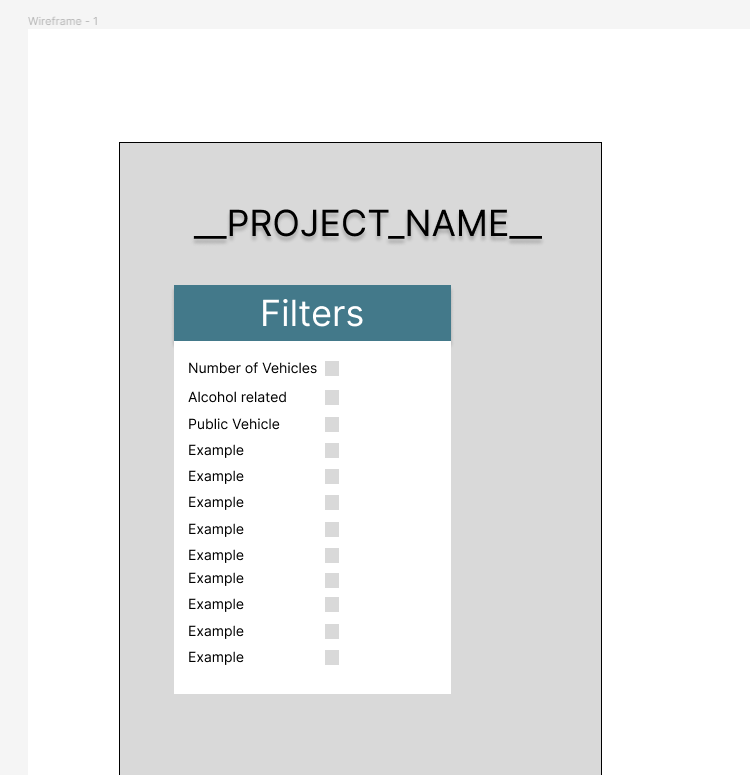
This structure is very simplistic and allows the team to focus on how the data is displayed, rather than complex UI functions. The user will navigate through the data presented to them in the data visualiser screen, this will include toggles for certain statistics involved in crashes such as public car involvement or if the person was harmed. Then the program will display what the end user has selected, this will include a heatmap of where those incidents with matching details occur.

Next we needed to design the look of the Interface this was done with Figma a web based wireframe designer. We kept the colours to simple blues and greys to allow for ease readability. A simple mockup of the main menu was designed as shown here.



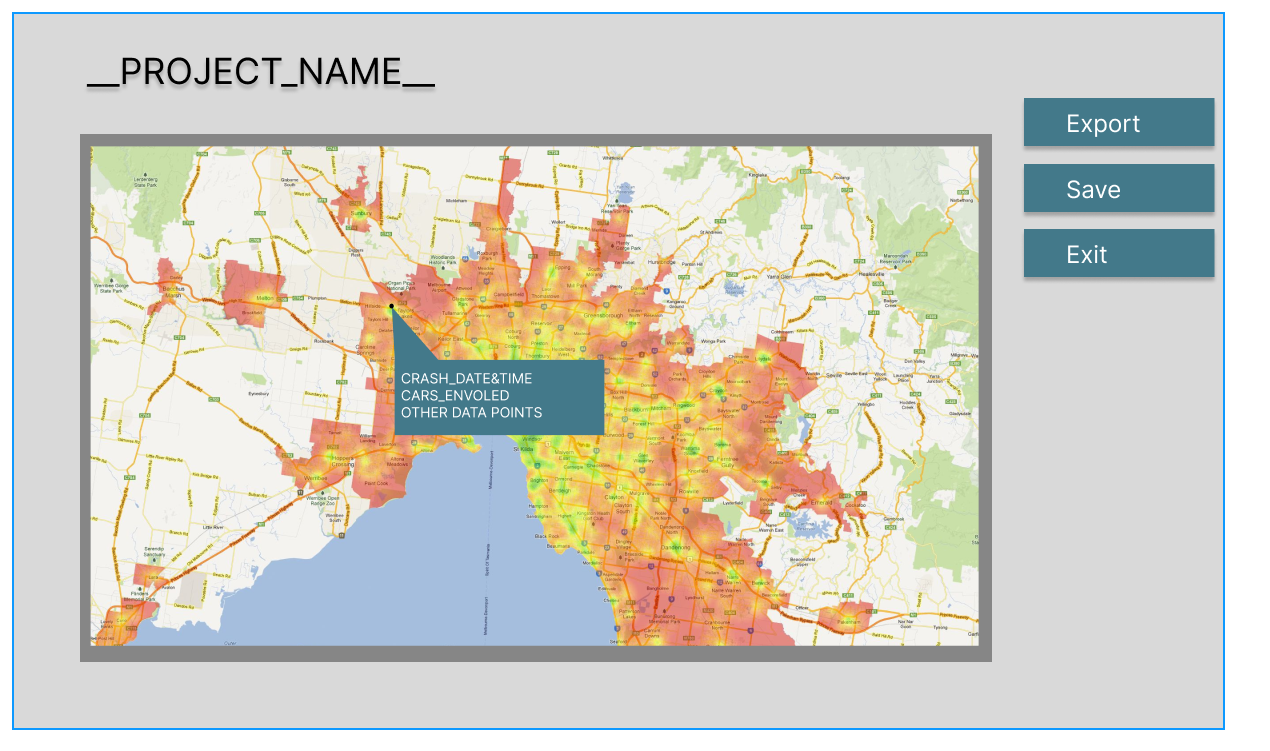
The main menu will not display as non-functional interactions for the end user at this point because this is the initialisation and setup stage. There will only be three options to choose from. New project will prompt the user at a later screen to select a XSL file to load the data from, and then a filter screen will be available to filter out any unwanted data points.Load project will allow the end user to quickly reopen a existing project which will allow for quick and easy viewing of data.Exit will allow the end user to leave the program. All of the fonts will be done in ‘Inter’ as it is very clear and legible at many scales. The buttons have been designed in a block list format to allow for drop down menus which will be used in the filters section, but the whole program uses this convention to standardise it.

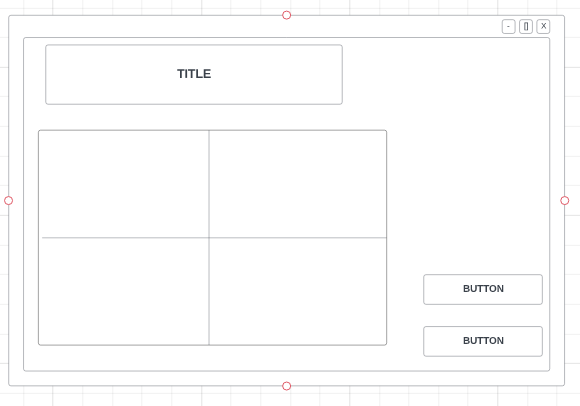
The User will be prompted if a new project is to be created with a prompt to name the project. this will appear at the top of the program to display to the end user what project they are looking at. This was chosen because it allows for multiple instances of different data sets to be easily identifiable and allows for a more swift workflow when working on large data analysis. Dropdown menus will appear on certain filters to allow the end user to select certain amounts or preset depending on the filter selected.



The next window is the data filter section that handles what data will be filtered out by pandas and what data will be kept. This is handled by selection boxes and dropdowns which gives the end user, this allows the data set to be exactly what the end user needs and doesn't allow excess bulk data to be stored which allows the data analytics to run faster.These choices were made to help the end user when they are working with bulky data sets that would need to be cut down.

This section has been given a lot of extra horizontal space to allow for the dropdowns and the other selection tool to be able to be seen by the end user. The same font and colour scheme has been kept to increase readability and legibility. It also helps make the program feel more complete and whole. Also the button position has been moved to allow the end user to find them more coherently to move onto the next part of the program the visualiser.





The data visualiser displays a heat map to the user of chosen data filters. The heat map could show to the end user the amount of alcohol related bus accidents in a local LGA by selecting those in the filter section. We have given the heatmap to most of the room because we feel like it is the centerpiece of the program that really shows off the data analytics which have been developed. You can mouse over particular data points on the maps since it is mapped longitudinally and latitudinally. This can intact a drop down which provides the end user with more information.

# References:

1. (Heatmap CDN2 2013) - <https://www.pinterest.com.au/pin/136585801171568935/visual-search/?x=16&y=16&w=532&h=295>