Software Design Document

UI implementation for Accident Data set

Manish Saily s5236772

Edward Winston s5133176

Table of Contents

[1.0 System Vision 3](#_Toc144638053)

[1.1 Problem Background 3](#_Toc144638054)

[1.2 System Overview 3](#_Toc144638055)

[1.3 Potential Benefits 3](#_Toc144638056)

[2.0 Requirements 4](#_Toc144638057)

[2.1 User Requirements 4](#_Toc144638058)

[2.2 Software Requirements 4](#_Toc144638059)

[2.3 Use Cases & Use Case Diagrams 5](#_Toc144638060)

[3.0 Software Design and System Components 9](#_Toc144638061)

[3.1 Software Design 9](#_Toc144638062)

[3.2 System Components 10](#_Toc144638063)

[3.2.1 Functions 10](#_Toc144638064)

[3.2.2 Data Structures / Data Sources 12](#_Toc144638065)

[3.2.3 Detailed Design 14](#_Toc144638066)

[4.0 User Interface Design 17](#_Toc144638067)

[4.1 Structural Design 17](#_Toc144638068)

[4.2 Visual Design 20](#_Toc144638069)

[5.0 References 21](#_Toc144638070)

# System Vision

## Problem Background

Data analysis is a time-consuming activity and researchers don’t always have the convenience to spend time on data analysis. To circumvent this issue the user interface will be created to help researchers analyse the data. This UI will be tailor made for car accident reports with. Governments and/or insurance companies will be using the UI meaning that they will supply the data, and we will be providing a visualisation of the data. The UI is a data visualisation tool. This means that the users will mainly be interacting with the program to visualise a set of data for ease of analysis. A data set was supplied which contained road crash statistics for a five-year period between 2015 to 2020 within the State of Victoria in Australia.

## System Overview

The software created in the project is a user interface which will allows for analysis of data and a visualisation. The user interface has been created with several functions. These include search by accident type, select time period, sort by speed zones, average number of accidents per hour and the effect of alcohol based on accident type. The tool will be coded in python with the use of the Pandas library and streamlit Builder. The data set that will be used is the Victoria State Accident Dataset and is in the form of a .csv file. The data will be visualised in the form of tables and charts depending on the analysis method that the user selected.

## Potential Benefits

There are several benefits for the creation of this user interface. This UI will allow the researchers to analysis data in a timely manner allowing for research to be produced quickly. The UI will be simple enough that researchers could use the UI for display the information for others. Produce charts that will help researchers display information. The chart will be produced by the UI for average number of accidents per hour. This will allow insurance researchers to provide data to insurance companies that will allow them to create specialised insurance plans based on the time their clients drive. The UI will allow the users to analysis the data resulting in better understanding for the relationship between alcohol and accidents. This UI will also aid in understanding the effects of speed zones on accidents. The UI’s ability to search by key words will allow researchers to access data on accidents by searching for their type.

# Requirements

## User Requirements

Assumptions: governments and/or insurance companies will be using the UI meaning that they will have the data already available, and we will be providing a visualisation of the data. The UI is a data visualisation tool. This means that the users will mainly be interacting with the program to visualise a set of data for ease of analysis.

The users should be able to select a period of time and display information from that time period. The information that should be displayed includes the following:

* the information on all the accidents that happened within that period of time.
* The UI should produce a chart to show the average number of accidents per hour, for each hour of the day.
* Within the period of time, it should be able to retrieve information on accidents by searching for key words such as pedestrian (user enters key words)
* The UI should allow the users to analysis the impact of alcohol in accidents by showing trends over time or accident types involving alcohol.
* The user should be able to use the UI and be able to see the number of accidents per speed zones in order to see the effect these zones have on total accidents
* Interface needs to allow the user to copy the data out of the tables and charts.

The users should have a menu to select the time period themselves. This will be in the form of a drop-down menu.

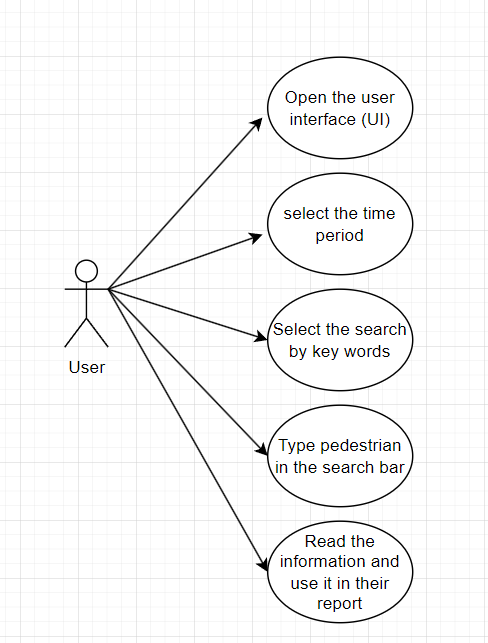
## Software Requirements

The software requirements are all the functional requirements that are needed by the software. There are many functionalities the software will need to meet the user requirements. Below are all the software requirements.

* The user interface needs to be able to access the data set.
* The user interface needs to be able to read the data using pandas .read\_csv().
* The user interface needs to be able to filter the data based on the year.
* The user interface needs to have a drop-down menu for the user to select the time period.
* The interface needs to be able to read user inputs for keywords.
* The interface needs to be able to produce charts for the users.
* The interface needs to be able to produce relevant tables.
* The interface needs to be able to calculate the percentage of alcohol involvement per type of accident.
* The interface needs to allow the users to return to the starting page.
* The interface needs to be able to produce a table to show the data the user searched for using the search engine.
* The program will use streamlit to create the user interface.
* Streamlit will create the UI through a local host meaning that installation of streamlit through miniconda is required.
* Streamlit will also be used to produce all the relevant charts and table for the user.

## Use Cases & Use Case Diagrams

Use case 1

User is a researcher who is working for the government. He will be using the UI to find out the impact pedestrian have on accidents. The researcher will be using this data in his report. To achieve the user will use the user interface (UI) that we created.

Use case 2:

A researcher is hired by a car insurance company to research the relationship between driving times and accidents. If the link between these two can be proven, then the insurance company can charge people depending on when they usually drive.

A diagram of a person with text

Description automatically generated

Use case 3:

The user is a government researcher who is researching the link between speed zones and accidents. The researcher will provide a report with this information to the policy makers to help inform the creation of new roads and the speeds that will be allocated to these roads.

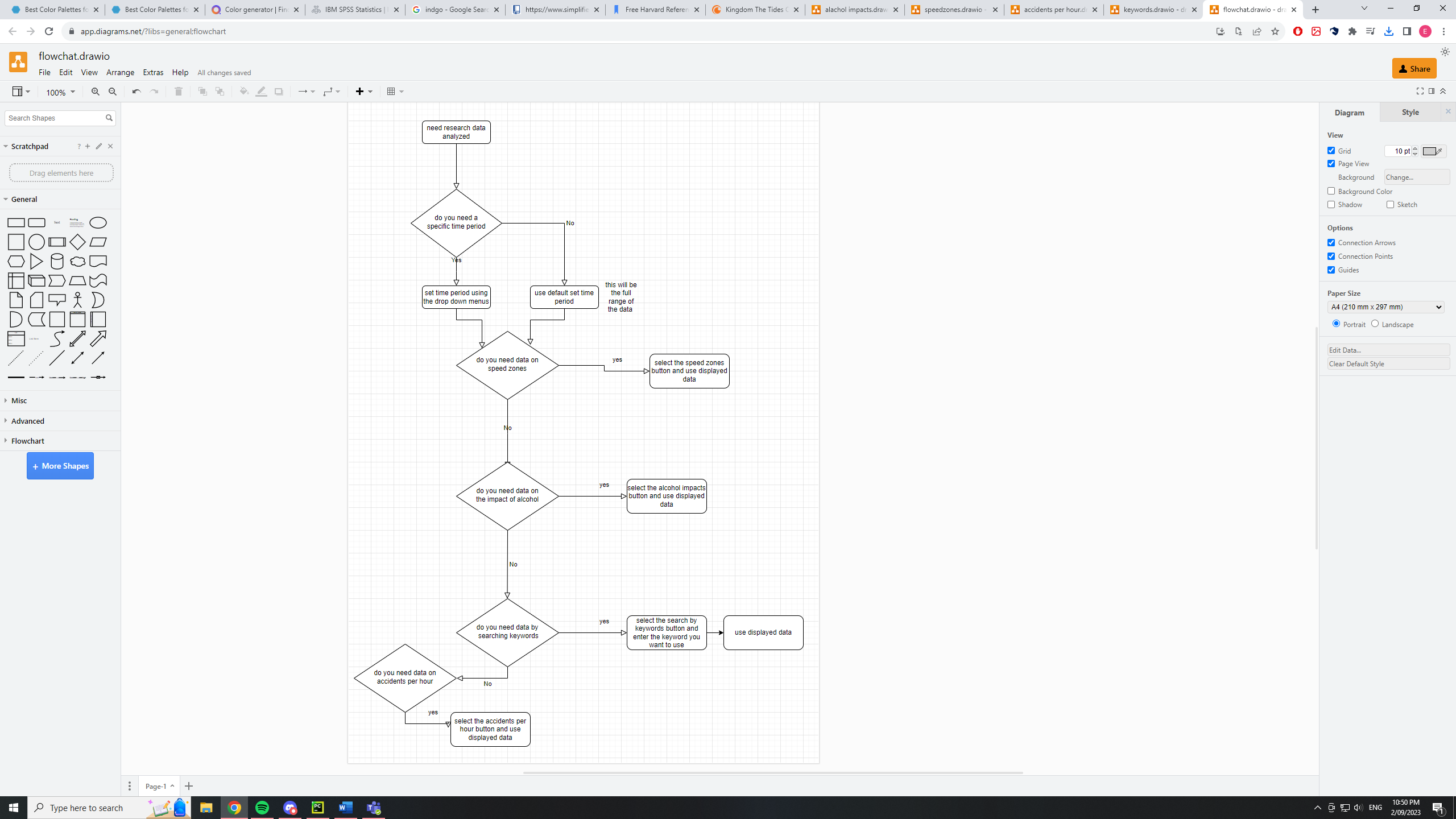
A diagram of a person with text

Description automatically generated

# Software Design and System Components

## Software Design

When the user opens the interface, they will be greeted with the home screen. On this screen there will be two drop down menus that will allow the user to select the time periods that they want analysed. Below this there will be four buttons. When the user opens the home page, they will fist select the time period that they want then they will collect one of the four buttons. Once they click on of these buttons it will move them to a new page corresponding to what they selected. This new page will show the data analysed with their inputs taken into consideration.



## System Components

### Functions

Below is a list of the functions that will be used in the software:

* import pandas and create variables for .csv file path and data reading.
  + import pandas as pd
  + csv\_file = ‘Crash Statistics Visctoria’ | data = pd.read\_csv(csv\_file)
    - the data variable will change depending on the csv\_file.
    - The return value of the function will be the correct .csv file is read as per the users choice.
* search engine with user input keyword
  + chosen\_year = string(input(“Enter the year: “))
    - this will change the (filtered\_data == chosen\_year) variable to the user input.
    - the return value will be a chart showing all accidents from the chosen filtered data.
* data filter function. The software requires at least two filter options one for year and one for keywords.
  + Data\_filter = data[data[‘ACCIDENT\_DATE’].dt.year == chosen\_year]
    - The data\_filter will change depending on whether the function requires a key word or year.
    - The return value will be the relevant filtered data that can be turned into a chart using streamlit.
* accidents per hour and sum to get averages using for loop.
  + chosen\_year[‘hour’] = chosen\_year[‘ACCIDENT\_TIME’].dt.hour
  + Count = chosen\_year.groupby(‘hour’)[‘ACCIDENT\_TIME’].count()
  + For hour, count in count.items():

Print(f’Hour {hour:02} average accidents = {count:.2f}’)

* + - Chosen\_year will store all hours form user selected year and count will count the accidents for each hour.
    - The acc\_per\_hour extracts all the hours stores them in ‘hour’. Count than groups the data by hour and counts the accidents for each hour. The for loop then iterates through the data stored in count and prints the average accidents and the hour. The output is displayed with zero-padding.
    - The return value is HOUR 01: Average Accidents = 174 …
* create charts using streamlit.
  + St.dataframe(filtered\_data[‘ACCIDENT\_TIME’])
    - Using streamlit we can create a data frame and specify the amount of data we want to use (filtered\_data) and the columns we want to use from the file.
    - The output will be a char that displays all the ACCIDENT\_TIME from the filtered data which can be the year.
* Selected column function where the charts will display only the relevant file columns.
  + Selected\_columns = [‘OBJECTID’, ACCIDENT\_DATE’, ACCIDENT\_TIME’]
    - The selected columns as the names of all the cell heading in the .csv file.
    - When we use st.dataframe() we can use selected\_columns as the use\_countainer\_width to only display the specified columns.

### Data Structures / Data Sources

The way the data is displayed will vary based upon which option the user selects between. There are examples below of how the data will be displayed for each option.

If the user selects search by key words, they will be greeted with a table of the data. An example of the table can be seen below. The table will be titled Input data (tear, year). This means if the user searches collisions with a fixed objected between 2016 and 2017 it will say: collision with a fixed object 2016-2017. Above this it will have a line which will show what was inputted; in this example it will read input=collision with fixed object.

Search by key words (2015-2016)

Inputted: collision with fixed object

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Accident type | OBJECTID | ACCIDENT\_NO | ABS\_CODE | ACCIDENT\_DATE | ACCIDENT\_TIME | Severity |
| collision with fixed object | 3401744 | T20130013732 | ABS to receive accident | 1/07/2013 | 18:30 | Serious injury accident |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

If the user selects the impacts of alcohol the software will generate a table. This table can be seen below. It will display the data within the time period that the user selects. It will be titled Impact of alcohol (2015-2017). If the user selects 2015 and 2017 as their time period. Percentage of alcohol involvement will have the user interface display the amount of that type of accident had alcohol involvement. For example, if there were 10 pedestrian accidents and 2 involved alcohol then the data will say 20%. For the severity they will be split into two categories; these being severity with alcohol involvement and severity without alcohol involvement. In the data set the severity shows as serious or other. Because of this the data set will display those two. To calculate these values the software will read the data and if it has alcohol involvement and was that accident type then it will read that data. In that data the software will tally each accident and assign them too serious or other. At the end the data will show this ratio as a percentage. So, if there were 10 accidents and 3 had a serious injury and 7 didn’t then it will show as 30% serous and 70% other. This will be done for both alcohol involvement and for no involvement. Of course, the user interface will only analyse the data within the time period that was stated at the start by the user.

For the impacts of alcohol

Impacts of alcohol (2015-2017)

|  |  |  |  |
| --- | --- | --- | --- |
| Accident types | % of alcohol involvement | Severity with alcohol involvement | Severity without alcohol involvement |
| Collison with fixed object | 20% of collison with fixed objects have alcohol involvement | 20% serious  80%other | 30% serious  70%other |
| Collision with vehicle |  |  |  |

If the user selects the speed zone option then they will be greeted with a graph. This graph will show the amount of accidents for each speed zone. On the x axis will be the speed zones and the y axis will be the number of accidents. It will be titled accidents per speed zone (year, year) where year, year is the user selected time period.

Graph example

If the user selects the accidents by hour then they will be meet with a bar graph. On this bar graph there will be the time of the day sorted by hour on the x axis. On the y axis the average amount of accidents for that hour will be displayed. This will be generated by using the total amount of accidents in that time period and dividing it by the amount of days that passed since then. An example of this graph can be seen below.

### Detailed Design

Import all the library’s

Import the data

Create a drop down menu

Read this as user inputs for time period

Filter the data set and only use data from that period from here on

Create the four buttons that the user can select.

If the user selects the alcohol impacts

Create a table for the data

Sort the data by accident type

Make a for loop for each of the accident types

In this loop if accident had alachol involvment then +1 to type\_involvement

#type\_involvement the type will be the accident type

Make another for loop for each of the accident types

If involve alcohol

If severity is serous +1 A\_serous\_type

Elseif severity is other +1 A\_other\_type

If not involve alachol

If severity is serous +1 N\_seroius\_type

Elseif severity is other +1 N\_other\_type

#Calculate the percentage of accidents are serious

A\_serous\_type/Type\_involvement = A\_serious\_%

A\_other\_type/type\_involvment = A\_other\_%

#now we need to figure it out for no alcohol involvement

N\_serous\_type/(Type\_involvement – total accident type) = N\_serious\_%

N\_serous\_type/(Type\_involvement – total accident type) = N\_other\_%

Print into table

Accident type

Type\_involvement/total number of accident for that type and have it as a %

A\_serous\_% #this will be the percentage of alcohol involved accidents with serous severity

A\_other\_% #this will be the percentage of alcohol involved accidents with other severity

N\_serous\_% #this will be the percentage of non-alcohol involved accidents with serous #severity

N\_other\_% #this will be the percentage of non-alcohol involved accidents with other #severity

If the user selects speed zones

Use loops for each speed zone

In these loops have it tally the total amount of accidents what occur in each speed zone

Create a chart

In this chart have each speed zone as the x bar

On the y axis show the total amount of accidents

Print chart

If the user selects accidents per hour

Use a for loop for the total accidents

If the time is 1pm then +1 to total\_1pm # and do this for each hour of the day.

Calculate the total amount of days that exist in the time period the user selected

Average1pm = Total\_1pm/total days

Create a chart

Use hours on the x axis

Use average 1pm and all the other times as the y axis

Print chart

If the user selects search by key words

Prompt the user to input a key word

Once user inputs key word use pandas to search the data set

#the key words will be the accident type

Retrieve OBJECTID, AACCIDENT\_NO, ABS\_CODE, ACCIDENT\_DATE ACCIDENT\_TIME, SEVERITY AND ACCIDENT TYPE

Create a table

Input accident type on the furthest left column

Then input OBJECTID, AACCIDENT\_NO, ABS\_CODE, ACCIDENT\_DATE ACCIDENT\_TIME, SEVERITY in that order

Print the user inputted keyword (year, year) #where year is the user selected period

Print inputted: user inputted keyword

Print the table

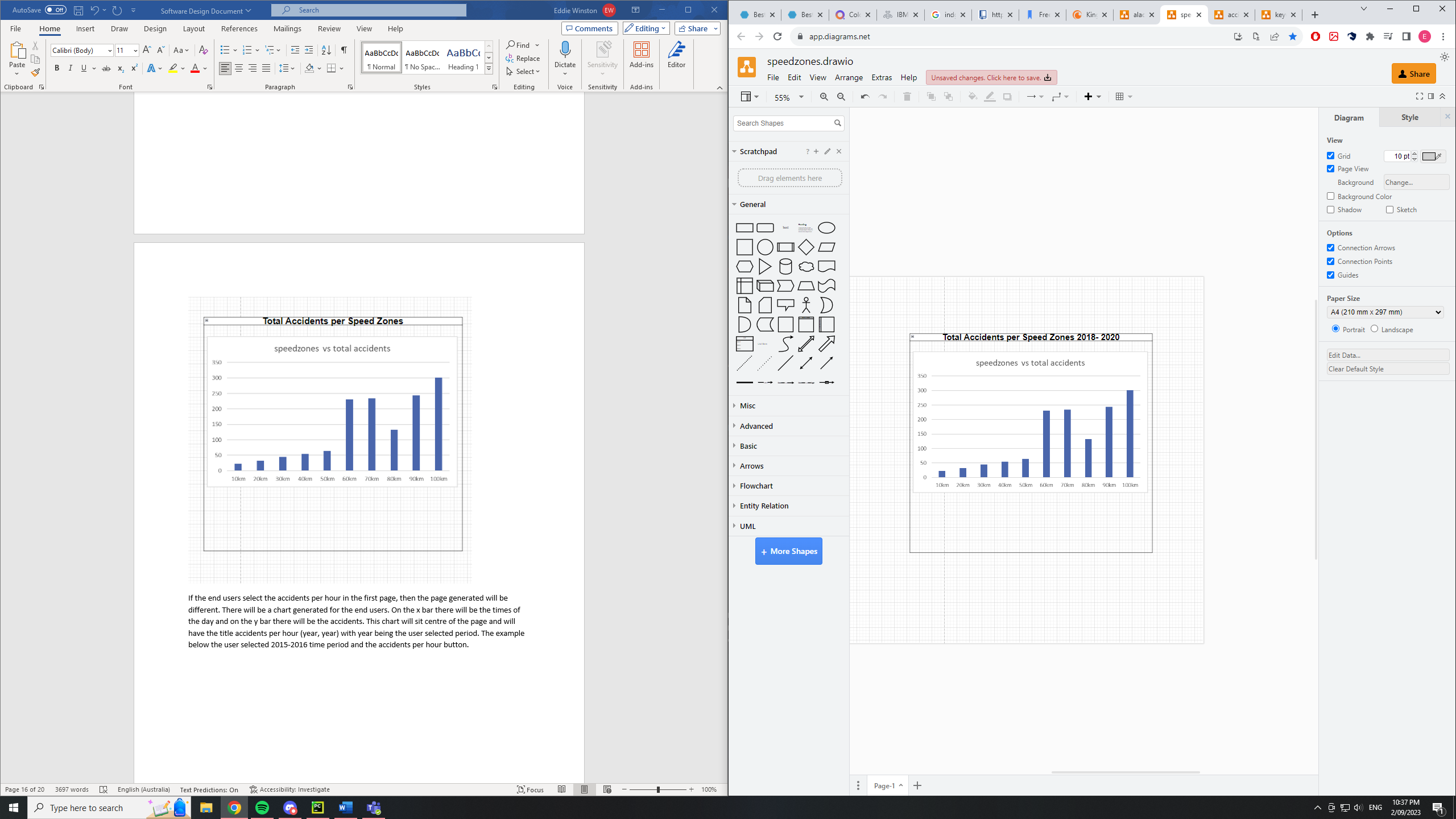
# User Interface Design

To create the initial interface design we used draw.io to convey ideas among the design team. These ideas were for the placement of buttons as well as the layout of the pages. Using draw.io wireframes were created which will allow us the visualise the end design. We have completed a wireframe for each screen that the user interface has. These include a home screen, alcohol impacts, speed zones, average accidents per hour and search by key words.

## Structural Design

There were several use cases outlined in 2.3 which showcase how the end users will use the application. Each of the end users will select the time period for their research first. Because of this, the drop down menu for the time period will be placed at the top. This is because it’s the first thing that the users will see. After the users have selected the time period they will then need to select one of the four buttons placed below it. These buttons will have a short description of what they provide. These will include Accidents per speed zone, Accidents per hour, Search by keywords and Alcohol impacts. When the users select these buttons, it will move to the next page.

For the accidents per speed zone, it will display a bar graph showing the speed zones on the x axis and accidents on the y bar. Above this button there will be a text saying accidents per speed zone(year, year) in the year, year it will show the range that the end users selected in the previous page. In the example below the user selected the 2018 to 2020 time period and also the speed zones button.



If the end users select the accidents per hour in the first page, then the page generated will be different. There will be a chart generated for the end users. On the x bar there will be the times of the day and on the y bar there will be the accidents. This chart will sit centre of the page and will have the title accidents per hour (year, year) with year being the user selected period. The example below the user selected 2015-2016 time period and the accidents per hour button.

A screenshot of a computer

Description automatically generated

On the home page if the users select the search by keywords it will generate a new page. This page will include a search bar at the top of the page. This search bar will allow the user to search using keywords. When the users enter a key word then the page will display a table showcasing the information of the keyword. This information will be for the time period the users selected earlier. In the example below the user inputted collision with fixed object between 2015-2016

A screenshot of a computer

Description automatically generated

If the users selected the alcohol impacts, then the UI will generate this new page. This page will display statistics relating to the impact of alcohol in a table. This table will be in the centre of the page with the title Impact of Alcohol per Accident Type (year, year). In the example below the user selected the 2015 to 2018 time period and the impact of alcohol button.

A screenshot of a computer

Description automatically generated

## Visual Design

For the design of the user interface several factors need to be considered. This user interface is designed mainly with researchers in mind as the end users. Because of this aesthetics were less valuable as the functionality of the design. For the font of the overall design, we decided to use Calibri as this is the most universally used font for research papers. The titles for the data will be in larger text than the data as well as being boldened. This will make it easier to read and stand out on the page.

Colour choices

For the colour choice of the user interface, we have decided to go with a combination of black, grey, white, and blue. The buttons will be blue with white text inside of the text. The background for the user interface will be white. The text for titles and non-button text will be in black with white background. These colour combinations were selected to ensure that the users could easily read the text and navigate the interface. Research was done into statistical software and these colours were a common trend. For the version of blue that will be used it will be #0000a7 this is because if the user is colour blind for most forms, they will see this as indigo which will still look for this program (Simplifiedsciencepublishing. 2022). The tables generated will be black text and borders with a white background. For the graphs the background will be grey with black texted for the writing. The bars in the graph will be the same blue as the buttons with the same colour-blindness issue as the reasoning.

Layout

There is examples of the layout of the user interface in the 4.1 section. These wireframes illustrate how the interface will be laid out. On the starting page there is a drop down menu on the top. This is because this is the first thing that the user has to input. Below the drop down menu there are 4 buttons that the user can press. After the user has inputted the time, the user will select one of these buttons. They have been positioned like this so that the user will do things in that order. The button on this interface has labels briefly describing what they will do; this description will be know longer than four words. With this the users will know what each button’s purpose will be without having to read too much. This will allow them to use the application quicker. In this design we used drop down menus for the user to select the time periods. This is because it was the easiest way to guide the user into inputting a value that the program can work with.

# 5.0 References

www.simplifiedsciencepublishing.com. (2022). *Best Color Palettes for Scientific Figures and Data Visualizations*. [online] Available at: <https://www.simplifiedsciencepublishing.com/resources/best-color-palettes-for-scientific-figures-and-data-visualizations>.

‌