

Software Design Document

VicAccidentStats App

Taylor Edgerton

Davinder Grewal

Kiarna Broomhead

Table of Contents

| | | |
|-------|--|----|
| 1.0 | System Vision | 3 |
| 1.1 | Problem Background..... | 3 |
| 1.2 | System Overview..... | 3 |
| 1.3 | Potential Benefits..... | 3 |
| 2.0 | Requirements..... | 3 |
| 2.1 | User Requirements | 3 |
| 2.1.1 | Non-Functional Requirements..... | 3 |
| 2.1.2 | Functional Requirements..... | 4 |
| 2.2 | Software Requirements | 5 |
| 2.3 | Use Cases & Use Case Diagrams | 6 |
| 3.0 | Software Design and System Components..... | 8 |
| 3.1.1 | Functions..... | 8 |
| 3.1.2 | Data Structures / Data Sources..... | 9 |
| 3.1.3 | Detailed Design | 9 |
| 4.0 | User Interface Design..... | 10 |
| 4.1 | Structural Design..... | 11 |
| 4.2 | Visual Design | 11 |

1.0 System Vision

1.1 Problem Background

The Victorian Government regularly creates educational campaigns for drivers on Victorian Roads. A program is required to demonstrate to these drivers' organised data and visual representations of the data to aid in road use education and how and when accidents occur.

1.2 System Overview

Graphical User Interface allowing users to select information and will provide organised and visual representation of Vehicle accident statistics. A simple user interface will allow many demographics of users to use the system. The system will utilise an already available CSV file of crash statistics. Python programming language will be used which is optimal for data analysis and data visualisation, and can be used on many operating systems, and has many data libraries to choose from.

1.3 Potential Benefits

This proposed system can be rolled out to many workstations, laptop and tablets for use within Victorian main roads education offices. Graphical user interface will ensure user experience is simple and caters for a large demographic of people and ranging tech savviness. By implementing a graphical user interface, users do not require experience using python command lines. By utilising SQLite, a database is not required, only a CSV file.

2.0 Requirements

2.1 User Requirements

It has been determined that the app will be targeted towards a large demographic of users for visual education of road crash statistics. User's age and tech savviness is determined to range, although most who use this app will have moderate tech savviness and be familiar with modern day apps and programs.

2.1.1 Non-Functional Requirements

User's need to access the program from a computer and use a keyboard and mouse for input. User's will then select from a list of headings which statistic they wish to view, and below each heading will contain instructions for selecting data constraints. The output or visualised data will then be shown in a window below the heading.

| Non-functional Requirements | |
|-----------------------------|---|
| Requirement | Description |
| Usability | <ul style="list-style-type: none">• Conventional icons and symbols used• Mobile version of the web user interface for remote use on a device• Conventional web symbols and icons to be used• Error messages will let a user know if an input was not valid |

| | |
|-----------------|--|
| Reliability | <ul style="list-style-type: none"> • Program will be backed up using a 3-2-1 backup policy |
| Performance | <ul style="list-style-type: none"> • Program and database can be installed on multiple workstations • System can be used on a computer workstation or laptop if required. |
| Security | <ul style="list-style-type: none"> • VPN must be used if devices connect to public networks • HTTPS protocol to be used for the user interface webpage • All devices using the system to have antivirus and malware software installed and configured • Devices accessing the system to be using up to date operating systems. • Office network to have a Firewall configured. • No user personal information will be entered or stored in the program |
| +Supportability | <ul style="list-style-type: none"> • Any code written will contain commenting • Training documentation to be provided • Support included in yearly expected costs to include helpdesk support and maintenance. |

2.1.2 Functional Requirements

Functional requirements below are the requirements for the program to perform the tasks required and are listed below.

| Functional Requirements | | |
|-------------------------|---|---|
| System | Function | Description |
| VicAccidentStats App | View accident details for a date period | Users can specify a date period and view details on all accidents that occurred during that period. |

| | | |
|--|---|--|
| | View accidents from each hour | Users can specify constraints for producing a chart of the average number of accidents in each hour of the day of a selected period will require a user to input a date range. |
| | View accidents caused by a keyword | To display all accidents caused by an accident keyword users will enter text input of a keyword. |
| | View trends of accidents due to alcohol | For a user to display trends of accidents due to alcohol will require a user will click a button for the program to display that information. |
| | View accidents involving motorcycles according to road geometry | A user will select a road geometry in order to view an analysis of trends of accidents involving motorcycles and the selected road geometry. |

- User requires the information of all accidents that happened during specific period.
- User also requires a chart which shows the average of the number of accidents happened on hourly basis during each day.
- By using keyword such as collision, overturn, struck, pedestrian that user seek the information regarding all the accidents caused by accident type for the specific period.
- User wants to analyse all the accidents happened due to the consumption of alcohol by just entering the time or accident types into the system.
- Users also seek the information about accidents happened due hit and run-on yearly basis.

2.2 Software Requirements

Software requirements below outline the minimum requirements for software to perform the tasks of the functional requirements.

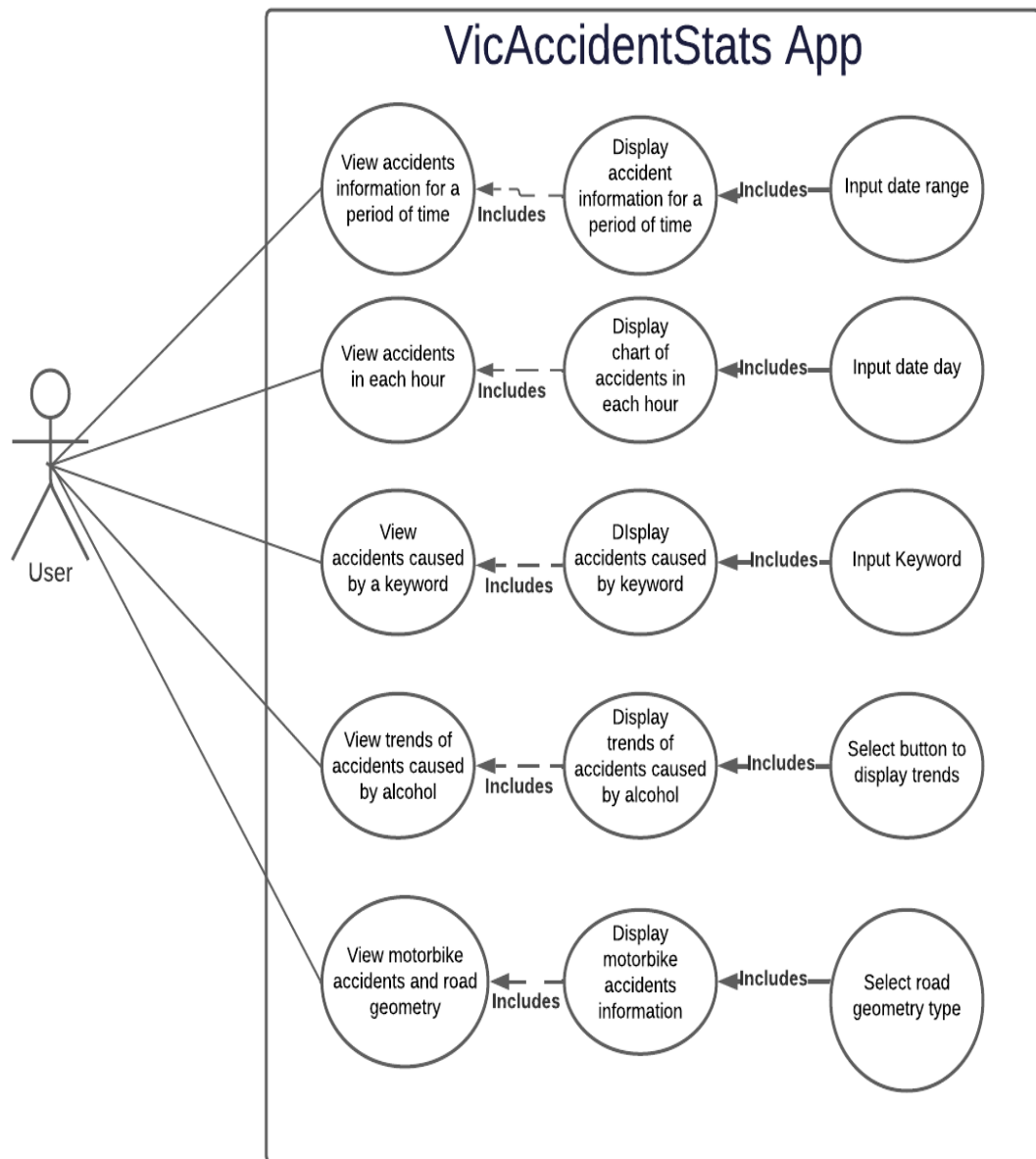
| Software Requirements | |
|-----------------------|---|
| ID | Description |
| R1 | The program shall accept a database from a CSV file to perform functions on the data and create an output |
| R2 | The program will be programmed in Python |

| | |
|----|---|
| R3 | The data used will be a CSV file that will be inserted into the program by a programmer |
| R4 | The program will apply the CSV file as an SQLite database. |
| R5 | Each input will apply an SQL query and the output will be based on that Query |
| R6 | The program will use SQL queries to access data from the database |
| R7 | The program will use libraries to create data visualisation elements |
| R8 | The program will use a python GUI library wxPython |

2.3 Use Cases & Use Case Diagrams

Use cases below outline the use case, actor and a brief description and the use case diagram outlines the overall use of the program.

| Use Case | Actor | Brief use case description |
|---|-------|---|
| View accident details for a date period | User | User inputs a date range and views a table of information for all the accidents that occurred during that date range. |
| View accidents in each hour | User | User inputs a date range to view a chart of accidents for each hour of the day |
| View accidents caused by a keyword | User | User inputs an accident cause keyword to view accident data of that cause |
| Display trends of accidents due to alcohol | User | User selects an option to display trends of accidents due to alcohol |
| Display road geometry and motorbike accidents | User | User selects a type of Road Geometry to view motorbike accident statistics for that selection |



3.0 Software Design and System Components

3.1.1 Functions

Modules and libraries:

Openpyxl is a python tool for reading and writing Excel 2010 xlsx/xlsm/xltx/xltn files. This will be used to access the excel CSV file for data manipulation.

wxPython is a cross-platform framework for creating graphical user interfaces for desktop computers. Pandas and SQLite is a module and database type that can be managed and created within python. Pandas will read a CSV file and convert into a database. Sqlite will allow queries to be performed using SQL.

Function and pseudocode included in the programming.

Function used to retrieve the data from the database for accidents by a date range.

Pseudocode:

```
Def GetAccidentsByRange()
```

```
cur = con.connect()
```

```
cur.execute("Select all from accidents where date range equal to user input")
```

Function used to retrieve the data from the database for accidents each hour of a day. This data will then be used to create a visual representation of the data.

Pseudocode:

```
Def getAccidentsByHour()
```

```
cur = con.connect()
```

```
cur.execute("Select all from accidents where day equal to user input")
```

Function used to retrieve the data from the database for accidents caused by a user entered keyword. If a keyword is not valid, a popup window will inform the user.

Pseudocode:

```
Def getAccidentsByKeyword()
```

```
cur = con.connect()
```

```
Try:
```

```
cur.execute("Select all from accidents where day equal to user input")
```

```
Except:
```



```
wxMessageDialog("Keyword not valid")
```

Function used to retrieve the data from the database for accidents due to alcohol

Pseudocode:

```
getAlcoholTrends()
```

```
    cur = con.connect()
```

```
    cur.execute("Select all from accidents where reason equals alcohol")
```

Function used to retrieve the data from the database for accidents due to motorbike from a selected road geometry

Pseudocode:

```
getMotorbikeAccidents()
```

```
    cur = con.connect()
```

```
    cur.execute("Select all from accidents where vehicle equals motorbike")
```

3.1.2 Data Structures / Data Sources

Tuples - will be used within the wxpython library classes for text and button locations. By including in the function pos = (x-axis, y-axis) the element will be positioned.

wxControls – Buttons and radio boxes and text and date pickers are used for user input and are held within wxPanels.

WxMessageDialog – a wxpython gui element that creates a “pop up” window alerting the user with an error message. This will be used if a parameter or keyword typed is not valid.

wxPanel – a wxpython gui element used to contain wxControls.

wxWindow – a wxpython gui element used to create the window for the program.

Arrays (Lists) - will be used for wxRadioBox function parameter of “Choices = List()” This will allow wxpython wxRadiobox to display an option for each item within the list.

3.1.3 Detailed Design

Pseudocode for page navigation:

```
    Def Page1()
```

```
        Hide() page2 and page3 elements
```

```
        Show() page1 elements
```

```
    Def Page2()
```

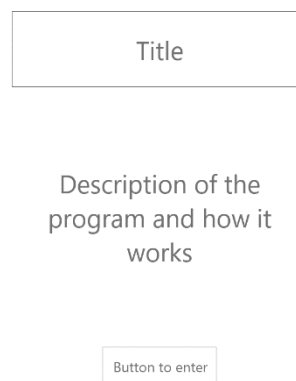
```
Hide() Page1 and page3 elements
Show() page2 elements
Def Page3()
  Hide() Page1 and page2 elements
  Show() page3 elements
```

4.0 User Interface Design

It was decided that the program would follow a web app design where a user would navigate through views using a main menu with buttons. Each data analysis page will contain a description of the data that will be shown. This was thought to be the most effective and simple way for users to understand and navigate the app with minimal tech savviness. The title would be located at the top of each page for consistency and a menu page with menu option buttons would allow users to navigate to different pages.

Wireframes were created using Adobe XD and a screen flow diagram also created to demonstrate the structural design of the program.

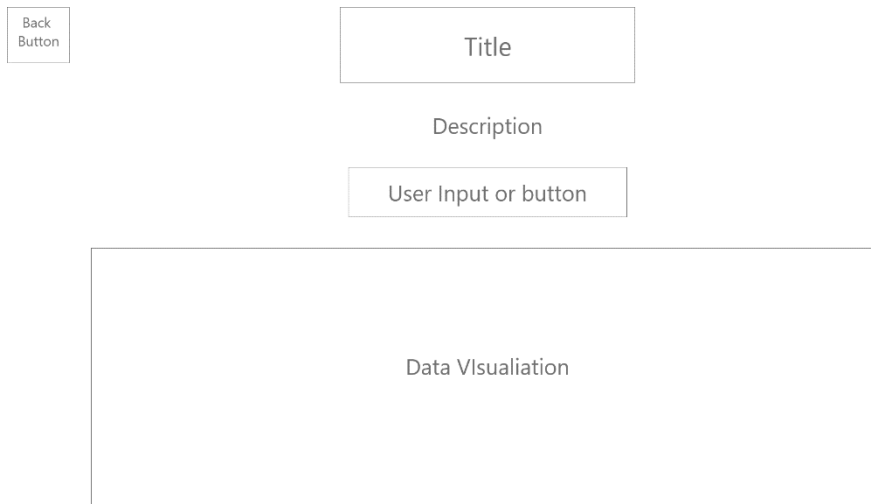
Wireframe 1



Wireframe 2



Wireframe 3



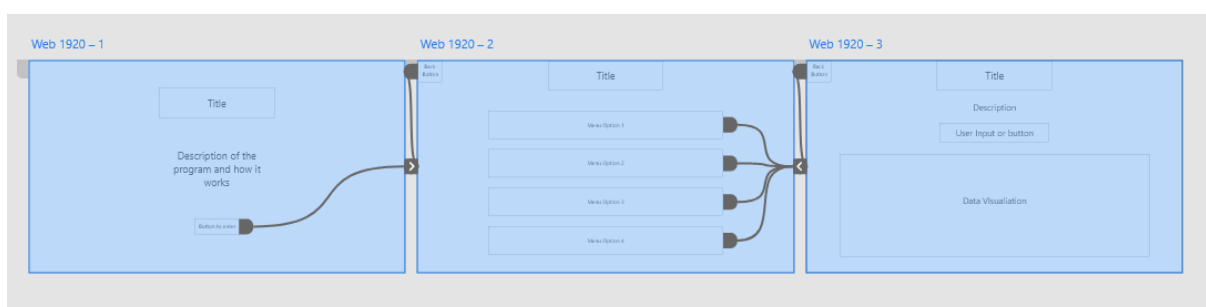
4.1 Structural Design

To ensure a program that was simple in navigation and familiar to users it was decided that a web app/mobile app style navigation design would ensure users are familiar with this style of navigation and would follow program navigation conventions. By utilising a “Back Button” at the top left of the frame, users would easily identify button locations based on these conventions.

As per Wireframe 3 example, a title and description of the page function will explain the purpose of the page clearly for a user. The button or user input field located below would associate that button with the title and description, and the data or data visualisation below would ensure that clearly that data being presented is relevant to the description and title.

The menu page containing menu options vertically is found directly after a title page with a description of the program, this would ensure each page is uncluttered and contains only relevant information required for the user, breaking it down will ensure less tech savvy users will understand their location within the program.

The below screen flow diagram shows how the title page, menu and back buttons allows the user to navigate the program.



4.2 Visual Design

Google’s Material Icons will be used as these icons are the most used and will be clear and modern to most demographic of users. Icons to be used will include the “Back” Icon.

To ensure the buttons stand out, button outlines would be used to contrast the button from the page background. The buttons would also be coloured, so the text is clear.

Text font will be Open Sans because it is used for long-term usage and readability because of its strong letterform legibility properties. Open Sans is easy on the eyes and suitable for everyday use.

Colour Blue is one of the best colours in UI design, and one of the most frequent. Just look at your smartphone app icons, and you'll see that a lot of them are blue: Facebook, Twitter, Shazam, Safari, etc.