Software Design Document

Accident Analysis Software

Nicolas Donaldson – S5256284

Table of Contents

[1.0 System Vision 3](#_Toc46748622)

[1.1 Problem Background 3](#_Toc46748623)

[1.2 System Overview 3](#_Toc46748624)

[1.3 Potential Benefits 3](#_Toc46748625)

[2.0 Requirements 4](#_Toc46748626)

[2.1 User Requirements 4](#_Toc46748627)

[2.2 Software Requirements 4](#_Toc46748628)

[2.3 Use Cases 4](#_Toc46748629)

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

[3.1 System Components 5](#_Toc46748631)

[3.2 Software Design 5](#_Toc46748632)

[4.0 User Interface Design 6](#_Toc46748633)

# System Vision

## Problem Background

A dataset containing data on Victoria State Accidents presents opportunities to discover underlying causes and other statistics surrounding road accidents that occur in Victoria. Software is required to perform an analysis on the data and present the data to the user in a way that is accurate, interpretable, and relevant. An increase in data has required the Department of Transport for Victoria to seek software that can draw conclusions and present findings, so that the Department of Transport may overcome the challenges associated with data analysis and human error. As data becomes too large, it becomes impossible to analyse the data without errors, misunderstandings, and large amounts of human resources. This is the problem the software intends to solve.

## System Overview

The system intends to provide a user-friendly interface that correctly and promptly returns search queries. The system will include functions that:

* For a user-selected period, displays the information of all accidents that happened in the period.
* For a user-selected period, produces a chart to show the number of accidents in each hour of the day (on average).
* For a user-selected period, retrieves all accidents caused by an accident type that contains a keyword (user entered), e.g., collision, pedestrian.
* Allows the user to analyse the impact of alcohol in accidents – ie: trends over time, accident types involving alcohol, etc.
* One other ‘insight’ or analysis tool

## Potential Benefits

Implementing software in the context of data analysis, alleviates the pressure on humans to avoid producing errors in the analysis. Without software, the Department of Transport may spend countless human and financial resources on producing results that software can produce within seconds. By streamlining long tasks into functions that can be performed by software, the Department of Transport can benefit with increased employee productivity and streamlined processes. Software can discover patterns and statistics that are otherwise impossible for humans to observe from a dataset. With this knowledge, the Department of Transport can benefit greatly with the software’s ability to guide the data analyst to solutions not recognised from just the human’s perspective on the data.

# Requirements

## User Requirements

As the client is the Department of Transport for Victoria, it is expected that it is employees of the Department that are the end users of the program. The end user must first begin with inserting the dataset. Once entered, the user may toggle between variations of the user interface. It is only required that the user enters certain parameters surrounding the nature of the user’s question. Once decided, and entered, the results of the user’s query shall be viewable via the output.

In a sequential list of needs, the above would be translated to:

1. Open program
2. Insert Dataset (if first time using application)
3. Insert Query Parameters
4. View Results

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

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

## Software Requirements

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

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

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

etc …

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

## Use Cases & Use Case Diagrams

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

# Software Design and System Components

## Software Design

A block diagram/flowchart of how your software might work

## System Components

### Functions

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

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

### Data Structures / Data Sources

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

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

### Detailed Design

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

# User Interface Design

This is your initial interface design. Describe the tools you used for this design stage and any key findings that informed your design. This introduction is descriptive and should explain what you have completed for the actual design work you will present in the sub-sections below.

## Structural Design

Structural design refers to the navigational and information structure of your product – the structure that supports the interface layout. How will you structure your product? How will you group your information? How will you navigate through your product? Why? This can take the form of a diagram showing structure and hierarchy, supported by a discussion and justification of your choices. Why have you made these design choices? Describe and outline the structure of your interface and of your information.

## Visual Design

Detail your visual design: Layout, visual elements, icons, graphics, style, colour, fonts general screen designs. This can be sketches, wireframes, mockups etc, supported by a discussion, explanation, and justification of your choices.