**PROJECT PLAN**

**Victoria State Accident Dashboard**

**Submitted By:**

**GROUP 33**

Augustine Kim | s5125270

Kai Chun Yu | s5309764

Ann Tapiador | s5314834

Table of Contents

[1.0 Introduction 3](#_Toc144640883)

[1.1 Background 3](#_Toc144640884)

[1.2 Scope 3](#_Toc144640885)

[1.3 Document contents 4](#_Toc144640886)

[2.0 Work Breakdown Structure 5](#_Toc144640887)

[3.0 Activity Definition & Estimation 5](#_Toc144640888)

[4.0 Gantt Chart 7](#_Toc144640889)

# Introduction

## Background

The goal of this project is to be able to develop a simple data analysis and visualisation tool pertaining to road crashes in the state of Victoria, to aid the Victorian government in their overall goal of improving road safety among road users, and in turn, alleviate corresponding social and economic costs.

Currently, the raw data is stored as a table in Excel file. The tool should be able transform this data into meaningful information, where the output is a graphical user interface that will allow users select parameters (e.g. period, type of accident etc.), and produce results in tables and charts that will aid them in data analysis. This will allow the Victorian government to measure if the polices that were set to improve road safety are effective, based on actual performance versus identified goals or metrics (e.g., lowering the number of alcohol-related accidents by a set percentage versus previous year).

## Scope

The scope of the project will cover parts of Scope, Time, and Quality Management, which involves designing, building, and testing activities. Other areas, i.e., Cost, HR, Communication, Risk, and Procurement Management will not be included.

The end output is a software with a graphical user interface that must show the following:

1. Information of all accidents based on a user-selected period.
2. A chart showing the average number of accidents in each hour of the day based on a user-selected period.
3. Retrieve all accidents caused by an accident type that contains a keyword entered by the user (e.g., collision, pedestrian), based on a user-selected period.
4. A chart that shows the impact of alcohol in accidents such as trends over time and accident types involving alcohol.
5. A chart showing the number of accidents among road users, i.e., bicyclists, passengers, drivers, pedestrians, pillions, and motorists, based on user-selected period.

Project documentation is also included, as listed below:

1. Project Plan
2. Gantt Chart
3. Software Design Document
4. Git Log in .txt file
5. Python codes in .py files
6. User Manual
7. Software Testing Report
8. Executive Summary

## Document contents

The table below shows details of the required documentation for the project:

|  |  |  |
| --- | --- | --- |
|  | **Documents** | **Details** |
| **1** | **Project Plan** | Details the plan for the project containing the following:   1. Overview of the project  * Background * Scope * Document Contents  1. Phase-Based Work Breakdown Structure (WBS). 2. Activity Definition and Estimation showing detailed steps of the WBS and corresponding timelines 3. Project Gantt Chart showing the schedule with corresponding critical path |
| **2** | **Gantt Chart** | Project Gantt Chart in Excel file |
| **3** | **Software Design Documents** | Document to help the software development team understand the requirement:   1. System Vision Document  * Background * System Overview * Potential Benefits  1. Requirements  * User requirements * Software requirements * Use Cases with corresponding use case diagram  1. System Components and Software Design  * Software Design * System Components which includes functions, data structures / data sources, and pseudocodes for all non-standard / non-trivial algorithms that operate on the data structures. |
| **4** | **Git Log in .txt file** | Version control to help keep track of the changes made to the documents throughout the project. |
| **5** | **Python codes in .py file** | Codes used for the software |
| **6** | **User Manual** | Detailed description of how to use the software |
| **7** | **Software Testing Report** | Details of the tests conducted;   1. Unit tests 2. Coverage report 3. Requirements Acceptance testing |
| **8** | **Executive Summary** | A summary of the final output’s capabilities |

# Work Breakdown Structure

Below is a phase-based work breakdown structure (WBS), which gives an overview of the deliverables for the project. The top section is the final deliverable, followed by the phases of the project, i.e., project management, requirement analysis, software architecture and design, development, testing, and summary. Under each phase are the deliverables.

A screenshot of a computer screen

Description automatically generated

# Activity Definition & Estimation

The table below lists details of the work breakdown structure presented on the previous section, which includes the description, duration (in days), and start / end dates.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Duration** | **Period Covered** |
| 1.1 | Initiation | Write details of the project background, scope, and required documentation on the project Plan. Write a system vision on the software design document, which should include problem identification, system overview, and potential benefits. | 2 | Aug 13 to Aug 14 |
| 1.2 | Planning WBS and Activity Definition & Estimation | List the activities to be undertaken to complete the project, together with corresponding timelines. This should include a Gantt chart to visualise the schedule. | 1 | Aug 15 |
| 2.1 | User Requirement | Write details on how the user is supposed to interact with the program from the end-user perspective | 3 | Aug 16 to Aug 18 |
| 2.2 | Software Requirement | List functional requirements, as well as non-fucntional requirements (usability, reliability, performance, security, implementation, and physical requirements. | 3 | Aug 19 to Aug 21 |
| 2.3 | Use Cases & Diagram | Create use cases based on the identified functional requirement, and illustrate in a use case diagram. | 2 | Aug 22 to Aug 23 |
| 3.1 | UI Design | Plan the structural and visual design of the system | 3 | Aug 24 to Aug 26 |
| 3.2 | System Component | List functions and data structures / data sources. Write the pseudocodes. | 3 | Aug 27 to Aug 29 |
| 3.3 | Software Architecture | Illustrate how the user interacts with the system using a flowchart. | 4 | Aug 30 to Sep 02 |
| 3.4 | Server Architecture | Create a database design | 2 | Sep 03 to Sep 04 |
| 4.1 | Software Development | Create codes using Python | 12 | Sep 05 to Sep 16 |
| 4.2 | Server Development | Develop database using SQL | 5 | Sep 17 to Sep 21 |
| 4.3 | UI Development | Develop user interface using HTML and CSS | 4 | Sep 22 to Sep 25 |
| 4.4 | User Manual | Develop a user manual that will detail how to use the software. This can be accessed on the user interface. | 4 | Sep 26 to Sep 29 |
| 5.1 | Unit Testing | Test the codes | 2 | Sep 30 to Oct 01 |
| 5.2 | Requirement Testing | Test the codes against the functional requirements | 2 | Oct 02 to Oct 03 |
| 6.1 | Executive Summary | Create a summary showing | 3 | Oct 04 to Oct 06 |
| 6.2 | Executive Full Testing | Prepare presentation and demonstration of the system. | 4 | Oct 07 to Oct 08 |

# Gantt Chart

Below is the Gantt Chart as of September 3, 2023. The red arrows indicate the critical path for the project.

