**Project Plan**

**NSW Traffic Penalty Data**

**Daehoon Seong**

**Waraphorn Srisomboonkamol**

**Justin Bilao**

Table of Contents

[1. Summary of changes 3](#_Toc147568125)

[1.1 Changes of Design 3](#_Toc147568126)

[1.1.1 Home page 3](#_Toc147568127)

[1.1.2 Requirement #1 3](#_Toc147568128)

[1.1.3 Requirement #2 3](#_Toc147568129)

[1.1.4 Requirement #3 4](#_Toc147568130)

[1.1.5 Requirement #4 4](#_Toc147568131)

[1.1.6 Requirement #5 5](#_Toc147568132)

[1.2 Changes of Project Schedule (WBS) 5](#_Toc147568133)

[1.2.1 Delete “Plan Database Schema” 5](#_Toc147568134)

[1.2.2 Replace “User Acceptance Test” to “Coverage Test” 5](#_Toc147568135)

[2. Introduction 5](#_Toc147568136)

[2.1 Background 5](#_Toc147568137)

[2.2 Scope 6](#_Toc147568138)

[2.3 Document contents 6](#_Toc147568139)

[3. Work Breakdown Structure 6](#_Toc147568140)

[4. Activity Definition & Estimation 8](#_Toc147568141)

[4.1 Project Initiation 8](#_Toc147568142)

[4.1.1 Define Objectives 8](#_Toc147568143)

[4.1.2 Define Scope of Project 8](#_Toc147568144)

[4.1.3 Define Constraints of Project 8](#_Toc147568145)

[4.1.4 Define Team Roles and Responsibilities 8](#_Toc147568146)

[4.2 Requirement Analysis 9](#_Toc147568147)

[4.2.1 Gather Requirements 9](#_Toc147568148)

[4.2.2 Analyse Requirements 9](#_Toc147568149)

[4.2.3 Define Functional Requirements 9](#_Toc147568150)

[4.2.4 Define Non-functional Requirements. 9](#_Toc147568151)

[4.3 Design and Planning 10](#_Toc147568152)

[4.3.1 Design Architecture 10](#_Toc147568153)

[4.3.2 Design User Interface 10](#_Toc147568154)

[4.3.3 Plan Database Schema 10](#_Toc147568155)

[4.4 Implementation 10](#_Toc147568156)

[4.4.1 Develop Basic Framework 10](#_Toc147568157)

[4.4.2 Implement User-Selected Period Reporting 11](#_Toc147568158)

[4.4.3 Implement Chart Distribution 11](#_Toc147568159)

[4.4.4 Implement Radar/Camera Case Retrieval 11](#_Toc147568160)

[4.4.5 Implement Mobile Phone Usage Analysis 11](#_Toc147568161)

[4.4.6 Implement Penalty Amount Analysis 11](#_Toc147568162)

[4.5 Testing and Validation 12](#_Toc147568163)

[4.5.1 Develop Test Cases 12](#_Toc147568164)

[4.5.2 Perform Unit Testing 12](#_Toc147568165)

[4.5.3 Perform User Acceptance Testing 12](#_Toc147568166)

[4.6 Deployment & Project Wrap-up 12](#_Toc147568167)

[4.6.1 Prepare Deployment Environment 12](#_Toc147568168)

[4.6.2 Create User Documentation. 12](#_Toc147568169)

[4.6.3 Host All Resources on GitHub 13](#_Toc147568170)

[5. Gantt Chat 14](#_Toc147568171)

# 1. Summary of changes

Here is a summary of the changes made in comparison to the original design and schedule that was set up at the start of the project. Up until the development stage, the project was progressing smoothly. However, during development, we faced some challenges such as pxPython visualization with matplotlib, data retrieval and cleaning, and misunderstanding of the requirements. These issues caused a delay in progress and required some design changes. Despite this setback, we were able to speed up the testing and validation process as we were able to set up the test cases much faster than expected. As a result, we were able to successfully release the app on GitHub.

## 1.1 Changes of Design

### 1.1.1 Home page

- The detailed description of the application tool is updated in the design.

|  |  |
| --- | --- |
| **Before** | **After** |
|  |  |

### 1.1.2 Requirement #1

- The functionality of downloading the information displayed in the application is deleted.

|  |  |
| --- | --- |
| **Before** | **After** |
|  |  |

### 1.1.3 Requirement #2

- The legend of the pie chart is added because hovering functionality in wzPython causes unexpected errors during the operation of the application. Furthermore, the direct view of description gives users more readable and understandable.

- The functionality of downloading the information displayed in the application is deleted.

|  |  |
| --- | --- |
| **Before** | **After** |
|  |  |

### 1.1.4 Requirement #3

- The window components are arranged vertically to show more table columns, making it easier to check data.

- The Lidar checkbox was added to filter data based on camera type.

|  |  |
| --- | --- |
| **Before** | **After** |
|  |  |

### 1.1.5 Requirement #4

- The requirement #4 should contain the information associated with “mobile” or “phone” so that the offence description is necessary to filter the information before plotting the chart. Finally, the input box is deleted.

|  |  |
| --- | --- |
| **Before** | **After** |
|  | A screen shot of a graph  Description automatically generated |

### 1.1.6 Requirement #5

- The penalty value is categorised into four within the selected timeframe. The data is displayed into a table.

|  |  |
| --- | --- |
| **Before** | **After** |
| A screenshot of a computer screen  Description automatically generated | A screenshot of a report  Description automatically generated |

## 1.2 Changes of Project Schedule (WBS)

### 1.2.1 Delete “Plan Database Schema”

The final application retrieves datasets directly from a .csv file, eliminating the need for planning a database schema.

### 1.2.2 Replace “User Acceptance Test” to “Coverage Test”

This course guide outlines the process of conducting a coverage test instead of using the user acceptance test.

# 2. Introduction

## 2.1 Background

The NSW Traffic Penalty Data offers a comprehensive compilation of traffic violations from 2011 to 2017. This dataset provides insights into various details such as the offence codes, offence descriptions, penalty amounts, etc. By analysing this dataset, authorities can discern patterns, peak offence periods, and specific regions with higher incidences, guiding better resource allocation and targeted awareness campaigns.

## 2.2 Scope

This project aims to develop a data analysis and visualisation tool tailored for the NSW Traffic Penalty Data. The objectives and deliverables for the tool are as follows:

1. User-Defined Period Analysis:
   * Allow users to view penalty cases for a selected period.
2. Offence Distribution Visualization:
   * Generate charts displaying the distribution of cases based on offence codes for a chosen time frame.
3. Camera and Radar Captured Cases Retrieval:
   * Retrieve cases captured by radar or camera based on offence descriptions.
4. Mobile Phone Usage Analysis:
   * Offer an analysis of causes related to mobile phone usage, showcasing trends over time, offence codes, and other relevant data points.
5. Financial Analysis:
   * Analyse the yearly revenue from penalties.
   * The tool will highlight which offences typically incur the highest penalties.

By the end of this project, stakeholders will have an interactive platform that provides insights on traffic penalties and sheds light on the financial aspect of the penalties, helping in better decision-making and policy adjustments.

## 2.3 Document contents

* Work Breakdown Structure with definition of each activity
* Gantt Chart

# 3. Work Breakdown Structure

This project will be executed for 12 weeks, which aligns with our trimester period, and each week has 5 working (or studying) days for students. This means that it will take the project team 60 working days to complete this project from the commencement to the deployment of the application. Hence, each activity will count daily, and all project members are considered to attend all activities in this work break structure. Furthermore, the work processes are divided into six steps, from planning to deployment. The first two weeks will focus on planning and requirement analysis, while the following nine weeks will be dedicated to actual development. Finally, the deployment and handover process will take one week.

| Task | WBS  No | Task Name | Duration | Remarks |
| --- | --- | --- | --- | --- |
| 1 | 1 | Project Initiation | 1 week |  |
| 2 | 1.1 | Define Project Objective | 1 day |  |
| 3 | 1.2 | Define Scope of Project | 1 day |  |
| 4 | 1.3 | Define Constraints of Project | 1 day |  |
| 5 | 1.4 | Define Team Roles and Responsibilities | 2 days |  |
| 6 | 2 | Requirement Analysis | 1 week |  |
| 7 | 2.1 | Gather Requirements | 1 day |  |
| 8 | 2.2 | Analyse Dataset of NSW Traffic Penalty Data | 1 day |  |
| 9 | 2.3 | Document Functional Requirement | 1 day |  |
| 10 | 2.4 | Document Non-functional Requirement | 2 days |  |
| 11 | 3 | Design and Planning | 2 weeks |  |
| 12 | 3.1 | Design Architecture | 5 days |  |
| 13 | 3.2 | Design User Interface | 5 days |  |
| ~~14~~ | ~~3.3~~ | ~~Plan Database Schema~~ | ~~5 days~~ | Delete in A2 |
| 15 | 4 | Implementation | 5 weeks |  |
| 16 | 4.1 | Develop Basic Framework | 2 days |  |
| 17 | 4.2 | Implement User-Selected Period Reporting | 3 days |  |
| 18 | 4.3 | Implement Chart Distribution | 5 days |  |
| 19 | 4.4 | Implement Radar/Camera Case Retrieval | 5 days |  |
| 20 | 4.5 | Implement Mobile Phone Usage Analysis | 5 days |  |
| 21 | 4.6 | Implement Penalty Amount Analysis | 5 days |  |
| 22 | 5. | Testing and Validation | 2 weeks |  |
| 23 | 5.1 | Develop Test Cases | 2 days |  |
| 24 | 5.2 | Perform Unit Testing | 7 days |  |
| 25 | 5.3 | ~~Perform User Acceptance Testing~~  Perform Coverage Testing | 1 day | Change |
| 26 | 6. | Deployment & Project Wrap-up | 1 weeks |  |
| 27 | 6.1 | Prepare Deployment Environment | 1 day |  |
| 28 | 6.2 | Create User Documentation | 3 days |  |
| 39 | 6.3 | Host All Resources on GitHub | 1 day |  |

# 4. Activity Definition & Estimation

## 4.1 Project Initiation

Each subject of the sub-activities will just take 1 day, and project member must conclude the meeting agenda within the schedule.

### 4.1.1 Define Objectives

|  |  |
| --- | --- |
| **- Description:** | In this task, the project's overall objective must be defined because the main objective would provide insights and information to implement the project. All members need to focus on the assignment's goal to develop a simple data analysis and visualisation  tool for the NSW Traffic Penalty Data. |
| **- Time Estimation:** | This activity will take 1 day to finish. |

### 4.1.2 Define Scope of Project

|  |  |
| --- | --- |
| **- Description:** | The project's scope for developing programs has a critical impact on the planning and implementation, testing, and deployment stages. All members must understand the objectives defined in the previous WBS, and they need to define the scope of the project based on it. This project requires “a simple analysis and visualisation tool,” The lecturer already proposed the development environment and programming languages such as Miniconda and Python. Accordingly, all members investigate how to develop this tool under the development environment and try to research frameworks and  libraries utilised during the project. |
| **- Time Estimation:** | This activity will take 1 day to finish. |

### 4.1.3 Define Constraints of Project

|  |  |
| --- | --- |
| **- Description:** | This task involves identifying the limitations or constraints ahead of our implementation plan. By defining these constraints, we can start to make a list of risks to be mitigated during the project. For example, limitations of working hours or lack of programming skills could be constraints. |
| **- Time Estimation:** | This activity will take 1 day to finish. |

### 4.1.4 Define Team Roles and Responsibilities

|  |  |
| --- | --- |
| **- Description:** | In this task, all team members discuss their roles and responsibilities to plan and develop the analysis tools. This task will specify who  will do a specific task defined in the WBS. |
| **- Time Estimation:** | This activity will take 2 days to finish. |

## 4.2 Requirement Analysis

In this section, the project team members need to have a meeting to analyse the project's requirements. It will take one week to finalise the application's requirements including functional and non-functional requirements.

### 4.2.1 Gather Requirements

|  |  |
| --- | --- |
| **- Description:** | In this task, all team members will collect detailed information on what the analysis tool must do to meet the project objectives. For our project, it is necessary to understand the requirements for analysing and visualising NSW Traffic Penalty Data. More specifically, the following list will be required to review:   1. Reporting penalty cases 2. Producing charts for offence codes, 3. Retrieving cases by radar or camera 4. Analysing trends for mobile phone usage and 5. Analysing penalty amounts |
| **- Time Estimation:** | This activity will take 1 day to finish. |

### 4.2.2 Analyse Requirements

|  |  |
| --- | --- |
| **- Description:** | This task involves understanding how to report information for user-selected periods, the types of charts required for offence code distribution, methods for retrieving cases, and specific analyses for the dataset. |
| **- Time Estimation:** | This activity will take 1 day to finish. |

### 4.2.3 Define Functional Requirements

|  |  |
| --- | --- |
| **- Description:** | Functional requirements describe the specific activities or functionalities the software system must perform (Satzinger et al., 2016). For this project, the functionalities we will create are to extract the file data and visualise it in the application window. All members will discuss the details of them. |
| **- Time Estimation:** | This activity will take 1 day to finish. |

### 4.2.4 Define Non-functional Requirements.

|  |  |
| --- | --- |
| **- Description:** | Non-functional requirements describe the system's features like usability, reliability, performance, and security. For this project, we only consider usability and performance considering the features we submit to get the grade for our assignment. Other features like  security and reliability are not needed. |
| **- Time Estimation:** | This activity will take 2 days to finish. |

## 4.3 Design and Planning

### 4.3.1 Design Architecture

|  |  |
| --- | --- |
| **- Description:** | The system architecture should be determined to show the interconnection that shows how the system works. This architecture will demonstrate the relationship between miniconda, python, IDE-Pycharm, and other libraries like wxPython. It will also give a clear understanding of how the development environment will be composed. The outcome will be presented as a diagram. |
| **- Time Estimation:** | This activity will take 2 days to finish. |

### 4.3.2 Design User Interface

|  |  |
| --- | --- |
| **- Description:** | Based on the functional requirement defined above activity, developers will design user interface. First, layout of the application will be designed. Second, developer will define which function or event will occur when user select some functions in the applications and design the graphical user interface accordingly. Lastly, basic wireframe that shows brief figure to apply it in practice in the system  development document. |
| **- Time Estimation:** | This activity will take 3 days to finish. |

## 4.4 Implementation

### 4.4.1 Develop Basic Framework

|  |  |
| --- | --- |
| **- Description:** | All developers will set up the development environment in this activity by installing Miniconda and Python first. Next, developers will draw the layout referencing the wireframe through the GUI tools. After that, developers will transform the graphic user interface into Python and set up the code. The result of this activity will demonstrate how the application works by retrieving data from the database and moving the layout when the button is clicked but without event handling. |
| **- Time Estimation:** | This activity will take 2 days to finish. |

### 4.4.2 Implement User-Selected Period Reporting

|  |  |
| --- | --- |
| **- Description:** | This activity will make a function about user-selected period in each layout. Each layout shows the different features from the dataset and the dataset should be filtered by using this functionality. |
| **- Time Estimation:** | This activity will take 3 days to finish. |

### 4.4.3 Implement Chart Distribution

|  |  |
| --- | --- |
| **- Description:** | This activity will make visualised charts in each layout. The dataset was already filtered in the user-selected period, and it should be visualised by using charting library. |
| **- Time Estimation:** | This activity will take 5 days to finish. |

### 4.4.4 Implement Radar/Camera Case Retrieval

|  |  |
| --- | --- |
| **- Description:** | This activity will make a function to filter the case of radar and cemara from the dataset, and it will be visualised by using the tools like WBS 3.4.3. |
| **- Time Estimation:** | This activity will take 5 days to finish. |

### 4.4.5 Implement Mobile Phone Usage Analysis

|  |  |
| --- | --- |
| **- Description:** | This activity will make a function to filter the case of mobile phone usage from the dataset, and it will be visualised by using the tools like WBS 3.4.3. |
| **- Time Estimation:** | This activity will take 5 days to finish. |

### 4.4.6 Implement Penalty Amount Analysis

|  |  |
| --- | --- |
| **- Description:** | This activity will make a function to filter the penalty amount analysis from the dataset, and it will be visualised by using the tools like WBS 3.4.3. |
| **- Time Estimation:** | This activity will take 5 days to finish. |

## 4.5 Testing and Validation

### 4.5.1 Develop Test Cases

|  |  |
| --- | --- |
| **- Description:** | In this activity, developers will create the test cases separately to see if the application will retrieve the correct dataset from the outside of the application. At least, 5 pecies of test cases should be prepared to apply it to each features of requirement. |
| **- Time Estimation:** | This activity will take 2 days to finish. |

### 4.5.2 Perform Unit Testing

|  |  |
| --- | --- |
| **- Description:** | With the test cases, developers will start to conduct testing, and see if the application retrieves the exact and correct data from the test cases. In addition, developers will check the visualisation quality which is mapped to each data retrieved. |
| **- Time Estimation:** | This activity will take 7 days to finish. |

### 4.5.3 Perform Coverage Testing

|  |  |
| --- | --- |
| **- Description:** | After completing the unit testing for each features, developers will conduct the coverage testing to check which parts of the code have been executed by the tests and which parts remain untested. |
| **- Time Estimation:** | This activity will take 1 day to finish. |

## 4.6 Deployment & Project Wrap-up

### 4.6.1 Prepare Deployment Environment

|  |  |
| --- | --- |
| **- Description:** | This activity is to check how the application is deployed and how the documentation will be produced. Based on the decision in this activity, developers will prepare the deployment environment to hand-over. |
| **- Time Estimation:** | This activity will take 1 day to finish. |

### 4.6.2 Create User Documentation.

|  |  |
| --- | --- |
| **- Description:** | This activity is for creating user manual which help use the application. This documents will guide to use the application. |
| **- Time Estimation:** | This activity will take 3 days to finish. |

### 4.6.3 Host All Resources on GitHub

|  |  |
| --- | --- |
| **- Description:** | Once all documents and application operations have been confirmed, the developers will proceed to upload all source code onto GitHub for archiving purposes. |
| **- Time Estimation:** | This activity will take 1 day to finish. |

# 5. Gantt Chat

