Project Plan

<Project Name>

Student Names

Table of Contents

[1.0 Introduction 3](#_Toc46748287)

[1.1 Problem Background 3](#_Toc46748288)

[1.2 Scope 3](#_Toc46748289)

[1.3 Document contents 3](#_Toc46748290)

[2.0 Work Breakdown Structure 4](#_Toc46748291)

[3.0 Activity Definition & Estimation 5](#_Toc46748292)

[4.0 Gantt Chart 6](#_Toc46748293)

# Introduction

## Background

*This project is about designing a software to analyse and visualize the NSW Traffic Penalty Data from 2011 to 2017. It aimed at providing a better comprehensive understanding of information about the penalty cases to the users. Users can obtain data such as the distribution of cases, number of cases captured by radar or camera, and cases caused by mobile phone usage. Trends of cases caused by mobile phone usage would be shown as well.*

## Scope

Part A is expected to be completed in approximately 4 weeks, followed by an additional 4 weeks for Part B. Therefore, the total time to complete both parts will be around 8 weeks. The purpose of the project is to create a comprehensive application for managing penalty cases that includes user interface design, backend system development, data integration, data analysis tool, testing and documentation. The process will employ agile approaches to provide data security and quality assurance while managing the user and software requirements, design, development, testing and deployment. The aim of the application is to deliver an effective and user-friendly solution for handling penalty cases that is supported by trustworthy analysis tools and procedures.

## Document contents

1. Project plan
2. Work Breakdown Structure
3. Gantt Chart
4. Software design document
5. Git\_log.txt

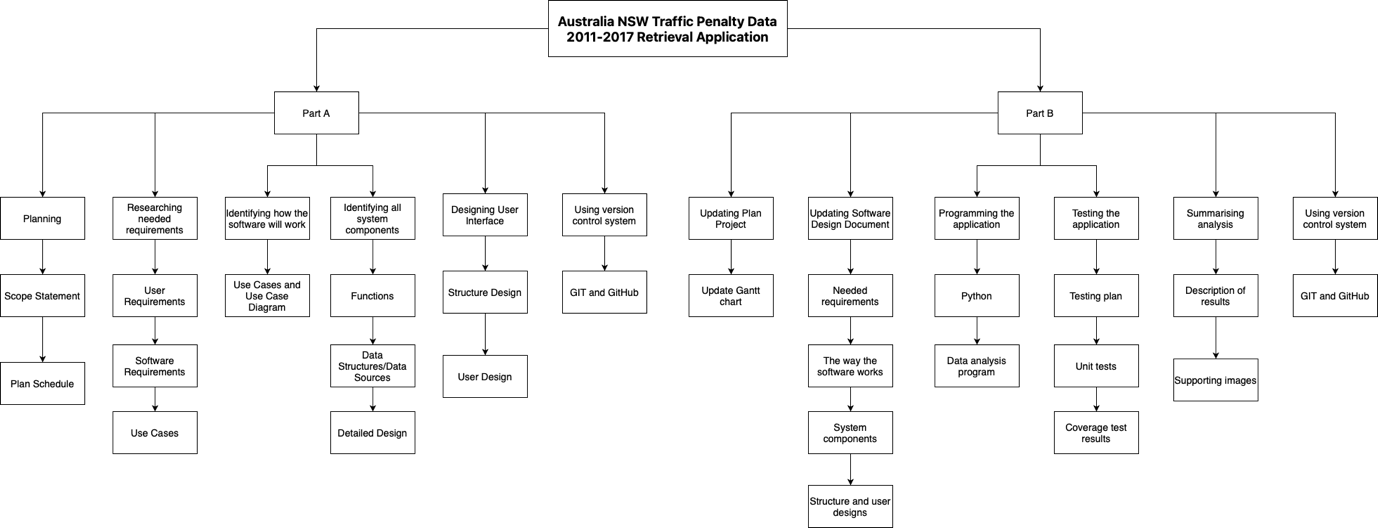
*Include some background information about the problem, the scope and what this document will contain.*

# Work Breakdown Structure

*This section should include the work breakdown structure for the whole project. The elements from the WBS should be used to generate your activity definition and those activities should then be scheduled in the Gantt Chart. Remember to consider ALL project activities – anything you do or will need to do should be included in the WBS*

*WBS’s are usually presented as some kind of hierarchical diagram/chart etc. The details what is involved each work unit should be provided in section 3:* ***Activity Definition***

*You do NOT need to do a WBS Dictionary for this project – the activity definition (whilst slightly different) will suffice. The WBS is focussed on SCOPE. The Activity definition is focussed on TIME.*



# Activity Definition & Estimation

*From your WBS, define the activities required for your project. You will revise this document and add more detail for part B as you discover more about the project.*

*Each activity should be clearly identified by a number and should match up to your Gantt chart. You should provide some estimations for the time you think each activity will take. This should make it easy to prepare your Gantt chart.*

*PartA:*

1.Planning

(a). Identify Scope of the project: Figure out exactly what the project will include and what time will this project take.

Estimated Time: About 2 days

(b). Generating WBS: Doing a flow chart to break down the project into smaller jobs that are easier to handle and see how they depend on each other.

Estimated Time: About 2 days

(c). Defining and Estimating Activities: Description for each task from WBS and guess how long it might take.

Estimated Time: Around 3days

(d). Create Gnatt Chart: Draw an excel that shows when each task will happen and what time they should be finished.

Estimated Time: About 3 days

(e). Allocate work: Give each team member specific tasks to do, considering what they're good at and when things need to be done.

Estimated Time: Around 3 days

2.Researching requirements

(a). Work out User requirements: Figure out what the users of the app really want and need it to do.

Estimated Time: About 3 days

(b). Work out software requirements: Identify what kind of software and technology the app should use to meet user needs.

Estimated Time: Around 2 days

(c). Generate Use Cases: Create a diagram to show how users will interact with the app and what it will do for them in different scenarios.

Estimated Time: About 3 days

3.Visualise how software will work.

(a). Employ Block Diagram/Flowchart: Create a simple diagram that shows how different parts of the software will work together, like building blocks or a step-by-step map.

Estimated Time: About 5 days

4.Identifying all system components

(a). List all functions: Make a list of all the things software will be able to do.

Estimated Time: About 2 days

(b). List all Data Structures/Data Sources: Identify where software will get and store information.

Estimated Time: Around 3 days

(c). Provide Detailed Design: Create a plan that explains exactly how all the different parts of your software will work together.

Estimated Time: About 4 days

5.Designing User Interface

(a). Outline Structure Design: Sketch out a basic layout of how the app will look and where things will go.

Estimated Time: About 3 days

(b). Detail Visual Design: Add colours, buttons, and other visual elements to make the app look good and easy to use.

Estimated Time: Around 3 days

6.Using Version control

(a). Employ GIT and GitHub: Use tools like GIT and GitHub to keep track of changes and work together on the software.

Estimated Time: During whole project

PartB

1.Update Part A

(a). Update Plan Project/ Gantt Chart: Make changes to project plan and timeline as needed based on progress.

Estimated Time: Anytime when PartB starting.

(b). Update Software Design document:

1.Edit Needed requirement: Modify the requirements that have changed or need to be adjusted.

2.Edit the way the software works: Update how the software works based on any new insights.

3.Edit System components: Make changes to the different parts of the software.

4.Edit Structure and user designs: Adjust the layout and visual elements of the app.

Estimated Time: Anytime when PartB starting.

2.Develop testing plan

(a). Develop Unit tests: Create small tests to check that different parts of your software are working correctly.

Estimated Time: About 3 days

(b). Describe Coverage Test Results: Test Results: Explain how much of your software is being tested and what the results are.

Estimated Time: Around 4 days

3.Program the application

(a). Employ Python: Write the code for app using the Python programming language.

Estimated Time: About 2 weeks

(b). Empty Data Analysis Program: Develop a basic program to analyse data, even if it doesn't do everything yet.

Estimated Time: Around 1 week

4.Summarise analysis

(a). Describe Results: Write about what you found during your analysis and any insights you gained.

Estimated Time: About 3 days

5.Using Version control

(a).Employ GIT and GitHub: Use tools like GIT and GitHub to keep track of changes and work together on the software.

Estimated Time: During whole project

# Gantt Chart

*This section should contain your Gantt chart. The items in the Gantt chart should match the activity definition from section 3. You should also submit your Gantt chart file separately.*