|  |  |
| --- | --- |
| Safe roads - Staying safe - NSW Centre for Road Safety  Project Plan  NSW Traffic Penalty Tool | Authors  Brianne Byer s5175100  Wonwoo Choi s5145987  Marco Querzola s5264979  2810ICT Software Technologies  Trimester 2, 2022 |

Table of Contents

[1.0 Introduction 2](#_Toc113182375)

[1.1 Background 2](#_Toc113182376)

[1.2 Scope 2](#_Toc113182377)

[1.3 Document Contents 2](#_Toc113182378)

[2.0 Work Breakdown Structure 3](#_Toc113182379)

[3.0 Activity Definition & Estimation 4](#_Toc113182380)

[4.0 Gantt Chart 8](#_Toc113182381)

# Introduction

## Background

Road rules are essential to abide by, as each law protects the driver, passenger, and other bystanders. However, traffic penalties are enforced to encourage responsible driving further when a driver disobeys. New South Wales is one of Australia's more lenient states regarding driving rules and penalties. Compared to other Australian states, New South Wales has had the most road deaths in the last 12 months. The community should be aware of traffic penalty trends to avoid danger or offending themselves. While said data is available to the public, it is often displayed and manipulated in an unfriendly manner. Creating a tool that focuses on traffic penalties in New South Wales is necessary.

## Scope

The main deliverable of the project is a user-friendly data analysis and visualisation software called the “NSW Traffic Penalty Tool” (NTPT). NTPT will identify significant traffic penalty trends, such as offence type, offence location, time of offence and many other variables. It is important to note the tool will utilise one dataset called “Australia NSW traffic penalty data 2011-2017”. Therefore, the project is restricted to said dataset and unable to assess other datasets available online. While the public will be able to use the software, the target users will be the government agency Transport for NSW (TfNSW). TfNSW is responsible for major road infrastructure, licensing of drivers and registration of motor vehicles. The NTPT will be built with open-source resources; no costs are associated to the project. Documentation and other assets of the project are accessible within the designated GitHub repository. Planning documentation for the project will be finalised by the 4th of September 2022. NTPT is to be fully operational by the 9th of October 2022.

## Document Contents

The Project Plan document contains an Introduction, Work Breakdown Structure, Activity Definition and Estimation, as well as a Gantt chart for the NTPT. Within the Introduction, a high-level overview of the background, schedule, cost, and limitations of the NTPT are established. Secondly, a work breakdown structure will demonstrate the key deliverables of the NTPT. The Activity Definition and Estimation section explores the dependency, duration, description, and identification of each project activity. Additionally, a precedence diagramming method is displayed, with the purpose to visualise all required activities. Lastly, the Gantt chart presents the NTPT project schedule which includes the owner of each activity, estimated time of completion, real time of completion, milestones, and other useful information.

# Work Breakdown Structure



Figure – NTPT Work Breakdown Structure

Figure 1 displays the main deliverable, key phases, and their associated components for the NTPT. Each work breakdown structure has four levels: main deliverable, key phases, work packages and activities. A top-down approach has been incorporated in forming Figure 1, with the largest item and main deliverable being the data analysis and visualisation software. The key phases include initialisation, planning, testing, backend, frontend, updates and closing. Each phase must be completed for the project to be completed.

# Activity Definition & Estimation

Time management is a significant aspect of project management. To reduce risk, each activity for the NTPT must be acknowledged. Figure 2 represents each activities duration, start time, finish time, identification, and relationships. Refer to Table 1 for a detailed description of each activity.

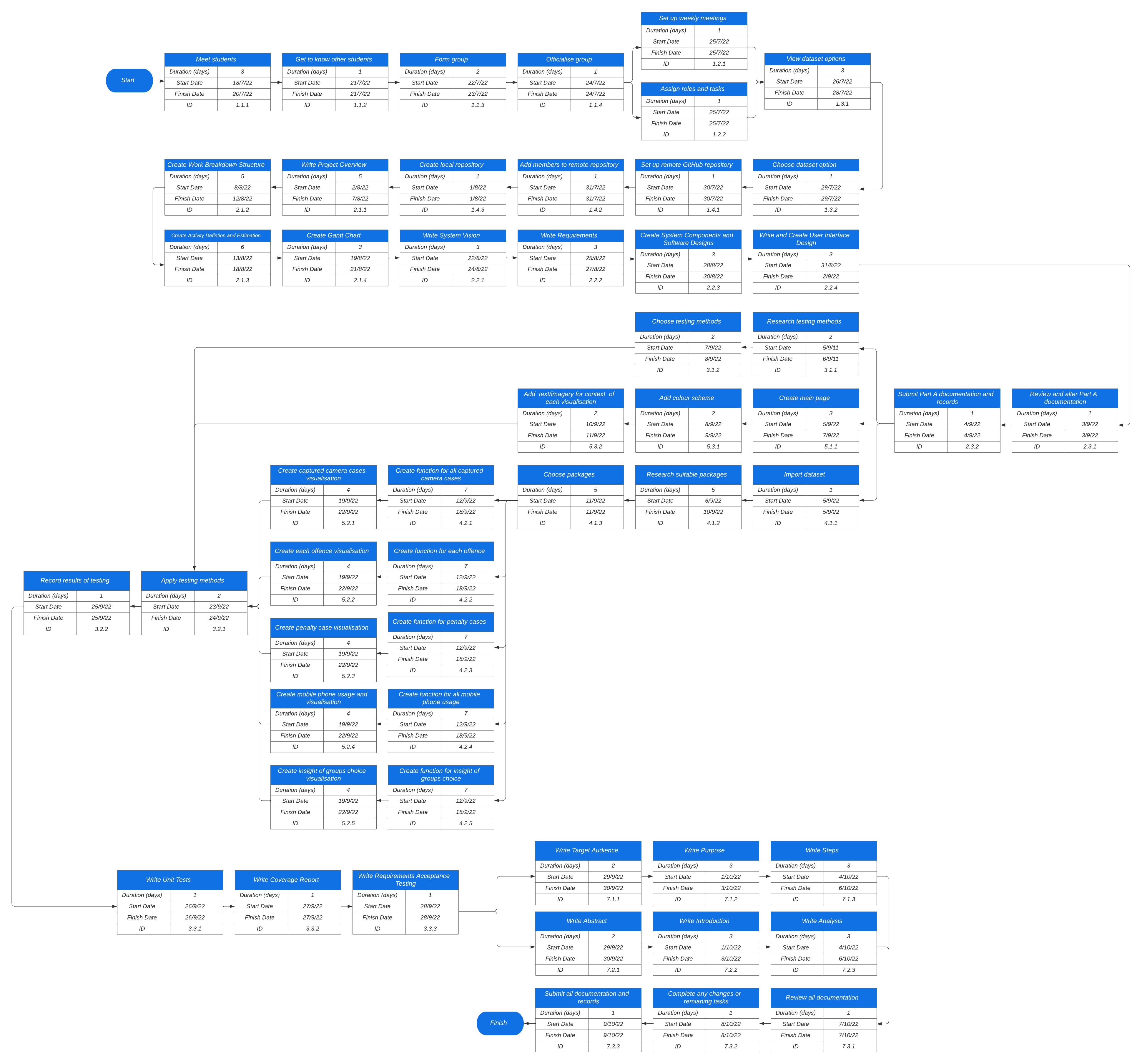


Figure – NTPT Precedence Diagramming Method

Table - Activity Description

|  |  |
| --- | --- |
| Activity | Description |
| 1.1.1 | Meet students from the online lectures, in-person labs, Microsoft Teams and other social media platforms. |
| 1.1.2 | Get to know other students by communicating online and in-person labs. Share information regarding career goals, current degree, and other relevant topics. |
| 1.1.3 | To form a group, expectations and goals must be discussed. Each student is to outline expectations of course (e.g. a grade of 7,6,5, or 4), time schedule and abilities. Expectations are agreed upon. |
| 1.1.4 | Students formalise a group on L@G Blackboard. |
| 1.2.1 | Set up a weekly meeting according to discussing schedules and time constraints. This information was previously discussed in 1.1.2 and 1.1.3. |
| 1.2.2 | Assign tasks according to each students’ strengths, weaknesses, and skills they want to improve. This information was previously discussed in 1.1.2 and 1.1.3. |
| 1.3.1 | View dataset options by reading listed data sets. From each dataset, review individual website. |
| 1.3.2 | Group discusses dataset together, agrees on chosen dataset according to interest. |
| 1.4.1 | Member of group to create a remote GitHub repository. This includes creating a GitHub account, creating repository, and selecting appropriate settings. |
| 1.4.2 | Member of group to add each member to remote Github repository. Members include student group and staff members (e.g. Gervase Tuxworth). |
| 1.4.3 | Each member of group (students) must pull repository onto their local machine. |
| 2.1.1 | Project Overview must be written for project. The background of the project must be researched and addressed. The scope of the project must be brainstormed and addressed. The contents of the document must be identified. |
| 2.1.2 | Work Breakdown Structure of project must be created. Group (students) must brainstorm main deliverable, key phases, work packages and activities associated to the product. From the brainstorm, a diagram is formed. |
| 2.1.3 | Activity Definition and Estimation of project must be created. Each task required for the project is identified. Each task is assigned an identification number, an estimated duration and description. Optionally, the precedence diagramming method can be utilised. |
| 2.1.4 | From 2.1.3, a Gantt Chart can be formed. Each key phase, work package and activity is to be visually represented on the chart. |
| 2.2.1 | System Vision must be written for project. The problem background must be researched and addressed. The system overview is to be addressed. The potential benefits are to be addressed. |
| 2.2.2 | Requirements must be written for project. The user requirements must be researched and addressed. The software requirements are to be addressed. The use cases are to be identified and visually represented. |
| 2.2.3 | Software Design and System Components must be written for project. The software design will be shown visually through a flowchart. Functions, data structures and detailed design of the project, must be researched and addressed. |
| 2.2.4 | User Interface must be written a displayed visually for project. Research must be undertaken that supports structural design choices. The visual design wireframes will be created. |
| 2.3.1 | Group (students) to review each section of Part A and make any final adjustments. |
| 2.3.2 | Group member to submit all documentation required for first submission. |
| 3.1.1 | Research testing methods by exploring well-known and recommended methods online. |
| 3.1.2 | Choose testing methods from research (3.1.1). |
| 3.2.1 | After each function and visualisation of the project is completed, apply chosen testing methods. |
| 3.2.2 | Records results of each testing result, from applying testing methods (3.2.1). |
| 3.3.1 | Unit Tests must be written for project. From the information and results gathered, fill out the designated table. |
| 3.3.2 | Coverage Report must be written for project. From research (3.1.1, 3.1.2) and unit testing (3.3.1), describe how coverage was evaluated. |
| 3.3.3 | Requirements Acceptance Testing must be written for project. From the information and results gathered, fill out the designated table. |
| 4.1.1 | View dataset format, ensure dataset is added to repository for use. |
| 4.1.2 | Research popular packages, popular databases and other resources that could be useful for project. This includes resources for back-end and front-end of project, such as data plotting and user interface packages for Python. It is important to consider dataset format. |
| 4.1.3 | After research, choose resources that are deemed most suitable for the team. This is according to the advantages, disadvantages and if any, previous experience in the chosen resources. |
| 4.2.1 | Create functions required for all captured camera cases as identified in chosen dataset. This includes calling values from the dataset according to user input, and then displaying in a line chart. Refer to the Visual Design section of the Software Design documentation. |
| 4.2.2 | Create functions required for each offence as identified in chosen dataset. This includes calling values from the dataset according to user input, and then displaying in a pie chart. Refer to the Visual Design section of the Software Design documentation. |
| 4.2.3 | Create functions required for penalty cases as identified in chosen dataset. This includes calling values from the dataset according to user input, and then displaying in a line chart. Refer to the Visual Design section of the Software Design documentation. |
| 4.2.4 | Create functions required for all mobile phone usage as identified in chosen dataset. This includes calling values from the dataset according to user input, and then displaying in a bar chart. Refer to the Visual Design section of the Software Design documentation. |
| 4.2.5 | Create functions required for cases in a school zone as identified in chosen dataset. This includes calling values from the dataset according to user input, and then displaying in a bar chart. Refer to the Visual Design section of the Software Design documentation. |
| 5.1.1 | Create main page according to Visual Design section of the Software Design documentation. |
| 5.2.1 | Create captured camera cases visualisation based on Visual Design section of the Software Design documentation. |
| 5.2.2 | Create each offence visualisation based on Visual Design section of the Software Design documentation. |
| 5.2.3 | Create penalty case visualisation based on Visual Design section of the Software Design documentation. |
| 5.2.4 | Create mobile phone usage visualisation based on Visual Design section of the Software Design documentation. |
| 5.2.5 | Create cases in a school zone visualisation based Visual Design section of the Software Design documentation. |
| 5.3.1 | Add colour to main page and each graph/data plot generated by functions. Refer to the Visual Design section of the Software Design documentation. |
| 5.3.2 | Add text and imagery for each visualisation generated by their specific function. Further context should be added to why data is being displayed and how it can be used. Follow the Visual Design section of the Software Design documentation. |
| 6.1.1 | Continuously sync local and remote repositories through the duration of the project. |
| 6.2.1 | Continuously change documentation from first submission, according to adjustments and changed plans. |
| 7.1.1 | Write target audience for User Manual documentation. |
| 7.1.2 | Write purpose for User Manual documentation. |
| 7.1.3 | Write steps for User Manual documentation. |
| 7.2.1 | Write abstract for Executive Summary documentation. This is a summary of findings from analysis. |
| 7.2.2 | Write introduction for Executive Summary documentation. This includes the purpose of the report, date range covered, and different analysis tasks performed. |
| 7.2.3 | Write analysis for Executive Summary documentation. Choose a 12-month period to view each feature of the data tool (as dictated in the Visual Design section of Software Design Documentation). Discuss results and display images for evidence. |
| 7.3.1 | Group to review all documentation, identifying completed and/or incomplete sections. |
| 7.3.2 | Group to make any changes to section or complete incomplete tasks before final submission. |
| 7.3.3 | Once documentation is reviewed and approved by all group members, all records and documentation will be submitted. |

# Gantt Chart

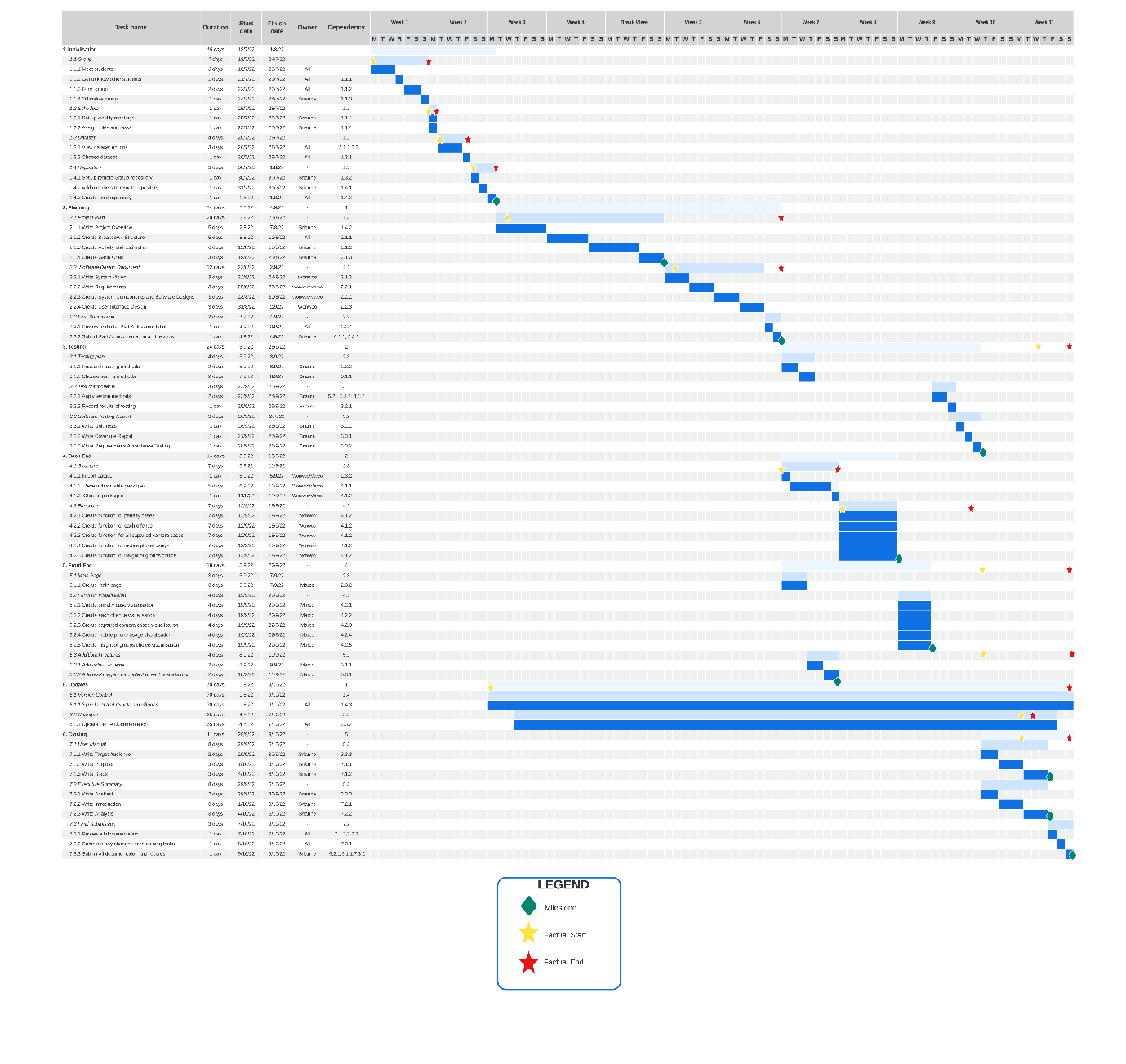


Figure – Gantt Chart

Figure 3 represents each activities expected start and end dates, with major milestones (green diamond) identified throughout the project. The factual start and end dates (yellow and red star respectively) of each milestone has also been addressed.