Project Plan

Sydney Airbnb Data

Table of Contents

[1.0 Introduction 3](#_Toc142905926)

[1.1 Background 3](#_Toc142905927)

[1.2 Scope 3](#_Toc142905928)

[1.3 Document contents 3](#_Toc142905929)

[2.0 Work Breakdown Structure 4](#_Toc142905930)

[3.0 Activity Definition & Estimation 7](#_Toc142905931)

[4.0 Gantt Chart 11](#_Toc142905932)

# Introduction

## Background

In the dynamic landscape of modern decision-making, data-driven insights have emerged as the linchpin of informed strategies. Our current initiative converges at the crossroads of data analysis and the vibrant Sydney Airbnb market, encapsulating the essence of a meticulously curated dataset sourced from Kaggle.

With a resolute goal, we aim to engineer a sophisticated software solution that unlocks the latent insights embedded within this dataset. It spans a gamut of attributes, spanning from property features to guest feedback, offering a panoramic view into the intricate fabric of the Sydney Airbnb ecosystem.

Beyond its observational role, our software encompasses a spectrum of functionalities tailored to empower users. From analysing suburb-specific trends over time to visualizing price dynamics, our software crystallizes the intricate dynamics of the Sydney Airbnb domain.

Moreover, the software delves into guest sentiments, deciphering the nuances of customer comments—a pivotal factor in shaping guest experiences. Infused with innovation, the software introduces an element of novelty, elevating the overall analytical experience.

At its core, this initiative rests upon the foundation of a meticulously curated Kaggle dataset—a beacon of real-world data. It's a pivotal bedrock upon which our software thrives, unfurling a world of possibilities.

Beyond abstract data, this endeavour holds the potential to optimize rentals, enhance guest experiences, and empower property owners with actionable insights. As we embark on this data-driven journey, the Sydney Airbnb market beckons—an intricate narrative waiting to be decoded. Our software, an embodiment of this vision, stands ready to reshape how we navigate the realm of transient lodging.

## Scope

The scope of this project is to design and implement a sophisticated software application for analysing the dynamic Sydney Airbnb market. The software will offer users the ability to interact with a curated dataset, enabling data retrieval, visualization, and insights generation. Key functionalities include querying property information based on user-defined criteria, creating informative price distribution charts, conducting keyword-based searches, and analysing customer feedback on cleanliness.

Beyond the technical development, the project also encompasses essential documentation phases. A comprehensive project plan will outline the project's lifecycle from inception to deployment, ensuring seamless coordination and progress tracking. The software design document will detail the architecture, user interfaces, and system components, aligning development with user-cantered design principles.

The project's ultimate ambition reaches beyond code to cultivate an enriching data ecosystem. By empowering users with tools for informed decision-making, property owners for optimizing rentals, and data enthusiasts for exploration, this Endeavor aims to amplify the value of data analysis in the context of the vibrant Sydney Airbnb market.

In essence, the project's scope encompasses technical excellence, strategic planning, and the creation of an impactful platform that transforms raw data into actionable insights, driving informed choices and enhancing understanding of the Airbnb market in Sydney.

## Document contents

This section outlines the specific contents and structure of the document. It provides readers with an overview of what to expect in the following sections:

* Work Breakdown Structure (WBS):

This section presents a detailed breakdown of the project's tasks and subtasks. It outlines the hierarchical structure of the project, starting from the initial tasks and branching out into smaller components. Each task is assigned a unique ID, and its description, duration, and any dependencies are provided. The WBS offers a comprehensive view of the project's scope and helps in understanding the sequence of tasks and their relationships.

* Activity Definition & Estimation:

In this section, each activity identified in the WBS is defined in more detail. The purpose, specific steps, and estimated time required for each activity are elaborated upon. By breaking down the tasks further, the project team gains a clearer understanding of the work involved and the effort needed to complete each activity successfully.

* Gantt Chart:

The Gantt chart, included in this section, visually represents the project timeline. It plots the start and end dates of each task on a timeline, showcasing their dependencies and durations. The Gantt chart provides a clear visualization of how tasks are scheduled and when they are expected to be completed. This aids in tracking progress and managing resources effectively throughout the project.

By structuring the document with these sections, readers gain insights into the project's structure, its component tasks, their estimated durations, and the overall project timeline. This content ensures that stakeholders understand what the document contains and can navigate through the information effectively.

# Work Breakdown Structure

The Work Breakdown Structure (WBS) section serves as a comprehensive outline of the project's tasks and activities, offering a high-level overview of the project's scope and components. It presents a hierarchical breakdown of the entire project into manageable work units, each representing a distinct task or activity. The WBS provides a structured framework to capture all essential elements of the project, guiding the subsequent activities and aiding in effective project management.

In this section, the WBS is presented in the form of a table, with each row corresponding to a specific task or activity. The tasks are organized in a logical sequence, often categorized by project phases or themes. While the WBS focuses on outlining the scope of the project, it forms the foundation for generating the Activity Definition and subsequently scheduling tasks in the Gantt Chart. The WBS provides a visual roadmap that helps stakeholders, team members, and project managers gain a clear understanding of the project's composition and the sequence of tasks required for successful completion.

|  |  |  |  |
| --- | --- | --- | --- |
| **Work Breakdown Structure** | | | |
| **Task ID** | **Task Description** | **Duration** | **Predecessor** |
| **Project Initiation** | | | |
| 1 | Gather group members | 1 Day | 0 |
| 2 | Select dataset | 1 Day | 1 |
| 3 | Set up GitHub repository | 1 Day | 2 |
| **Preparatory Work** | | | |
| 4 | Understand dataset structure | 2 Days | 2 |
| 5 | Identify analysis tasks | 1 Day | 4 |
| 6 | Define additional analysis task | 1 Day | 5 |
| 7 | Discuss software design approach | 2 Days | 5 |
| **Prepare Project Plan** | | | |
| 8 | Create Project Overview | 1 Day | 4 |
| 9 | Develop Work-Breakdown Structure | 2 Days | 5 |
| 10 | Define Activity Definitions and estimates | 1 Day | 9 |
| 11 | Build Gantt chart | 2 Days | 10 |
| **Prepare Software Design Document** | | | |
| 12 | Create System Vision Statement | 1 Day | - |
| 13 | Identify and list user requirements | 1 Day | 12 |
| 14 | Identify and list software requirements | 1 Day | 12 |
| 15 | Define use cases and personas | 2 Days | 13,14 |
| 16 | List system components and design | 3 Days | 15 |
| **UI Wireframe Design** | | | |
| 17 | Design early user interface wireframe/mock-ups | 2 Days | 16 |
| 18 | Refine and finalize wireframe design | 1 Day | 17 |
| **Software Implementation and Testing** | | | |
| 19 | Implement Data Analysis Program |  |  |
| 19.1 | Set up project structure and environment | 1 Day |  |
| 19.2 | Implement data loading and preprocessing | 2 Days |  |
| 19.3 | Develop user interface components | 3 Days |  |
| 19.4 | Implement analysis algorithms | 4 Days |  |
| 19.5 | Integrate UI and analysis components | 2 Days |  |
| 20 | Perform Unit Testing |  |  |
| 20.1 | Create test cases for data loading | 1 Day |  |
| 20.2 | Execute unit tests for UI components | 2 Days |  |
| 20.3 | Conduct unit tests for analysis | 3 Days |  |
| 20.4 | verify UI and analysis integration | 2 Days |  |
| 20.5 | Perform unit testing and debugging | 2 Days |  |
| **Software Testing Report** | | | |
| 21 | Compile test cases and outcomes | 1 Day |  |
| 22 | Include coverage and evaluation details | 1 Day |  |
| 23 | Document acceptance testing results | 1 Day |  |
| 24 | Explain implementation and testing process | 1 Day |  |
| **Executive Summary** | | | |
| 25 | Summarize project objectives and outcomes | 1 Day |  |
| 26 | Highlight key achievements and results | 1 Day |  |
| 27 | Include main findings and recommendations | 1 Day |  |
| 28 | Edit and proofread summary | 1 Day |  |
| **Final Review and submission** | | | |
| 29 | Review Project plan, testing and summary reports | 2 Days |  |
| 30 | Revise and edit content | 2 Days |  |
| 31 | Track actual completion times | 1 Day |  |
| 32 | Regularly update Gantt chart | 1 Day |  |
| 33 | Git log text document | 1 Day |  |
| 34 | Finalize all documentation | 2 Days |  |
| **Final Review and submission** | | | |
| 35 | Revise and finalize evrything | 2 Days |  |
| 36 | Submit all project deliverables | 1 Day | - |

# 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.*

Project Initiation:

1. Gather Group Members (1 Day):

* Contact potential team members, discussing their skills and availability.
* Formally invite selected members to join the project team.

1. Select Dataset (1 Day):

* Research available Airbnb datasets on platforms like Kaggle and data repositories.
* Evaluate dataset options based on size, relevance, and data quality.
* Choose the most suitable dataset for the project's analysis goals.

1. Set up GitHub Repository (1 Day):

* Create a new GitHub repository for the project.
* Configure repository settings, including access permissions and collaboration guidelines.
* Initialize the repository with essential files, such as a README and license.

1. Preparatory Work: 4. Understand Dataset Structure (2 Days):

* Download and explore the selected dataset's contents.
* Identify the various data fields, their types, and potential relationships.
* Document an overview of the dataset's structure for reference.

1. Identify Analysis Tasks (1 Day):

* Brainstorm potential analyses to extract insights from the dataset.
* Prioritize analysis tasks based on their relevance and potential impact.
* Document a list of analysis objectives and associated tasks.

1. Define Additional Analysis Task (1 Day):

* Collaboratively brainstorm to identify an extra analysis task beyond the initial list.
* Evaluate the feasibility and significance of the additional task.
* Document the details and purpose of the new analysis task.

1. Discuss Software Design Approach (1 Day):

* Conduct a team meeting to discuss software design considerations.
* Analyse various architectural approaches and their pros and cons.
* Choose a suitable design approach based on the project's objectives.

Prepare Project Plan:

1. Create Project Overview (1 Day):

* Write a concise overview of the project, including its goals, scope, and significance.
* Summarize the dataset's relevance and potential impact on the software.
* Provide an outline of the document's contents.

1. Develop Work-Breakdown Structure (1 Day):

* Break down the project into smaller tasks and sub-tasks.
* Organize these tasks in a hierarchical structure, showing their relationships.
* Ensure that each task is well-defined and contributes to the project's objectives.

1. Define Activity Definitions and Estimates (1 Day):

* Elaborate on each task in the work breakdown structure.
* Specify the purpose, expected outcomes, and deliverables of each task.
* Estimate the time required for each task based on its complexity and dependencies.

1. Build Gantt Chart (2 Days):

* Translate the activity definitions into a visual timeline.
* Establish task dependencies to create a sequential order of execution.
* Assign estimated durations to each task and generate a Gantt chart.

Prepare Software Design Document:

1. Create System Vision Statement (1 Day):

* Define the purpose and objectives of the software.
* Describe the problem domain and how the software addresses it.
* Highlight the potential benefits and impacts of the software.

1. Identify and List User Requirements (1 Day):

* Gather requirements by engaging potential users or stakeholders.
* List the features, functionalities, and interactions expected from the software.
* Ensure that the requirements are clear, concise, and aligned with user needs.

1. Identify and List Software Requirements (2 Days):

* Derive technical requirements from the user requirements.
* Detail the software functionalities, data handling, and processing capabilities.
* Ensure that the software requirements are complete and unambiguous.

1. Define Use Cases and Personas (2 Days):

* Describe the scenarios in which users will interact with the software.
* Create use case diagrams to visualize user interactions and system responses.
* Develop user personas to better understand the software's target audience.

1. List System Components and Design (3 Days):

* Identify the major components/modules of the software.
* Describe the purpose and responsibilities of each component.
* Sketch a high-level design or flowchart depicting the interactions between components.

UI Wireframe Design:

1. Design Early User Interface Wireframe/Mock-ups (2 Days):

* Create initial wireframes illustrating the layout and structure of the user interface.
* Focus on representing key elements and interactions without detailed visuals.
* Gather feedback from team members to refine the wireframes.

1. Refine and Finalize Wireframe Design (1 Day):

* Incorporate feedback and adjustments into the wireframes.
* Ensure that the wireframes accurately reflect the intended user interface.
* Finalize the wireframes for further design and implementation.

Update Project Plan:

1. Revise and Finalize Project Plan (2 Days):

* Review and update the project plan based on any changes or additions.
* Ensure that all tasks and their dependencies are accurately represented.
* Validate that the estimated timeframes align with the project's scope and objectives.

1. Track Actual Completion Times (1 Day):

* Record the actual start and completion times for completed tasks.
* Compare actual times with estimated times to monitor project progress.
* Identify any discrepancies and adjust the project plan as needed.

1. Regularly Update Gantt Chart (1 Day):

* Continuously update the Gantt chart to reflect completed tasks and milestones.
* Adjust the chart to account for any changes in task dependencies or timelines.
* Keep the Gantt chart accessible and shareable for all team members.

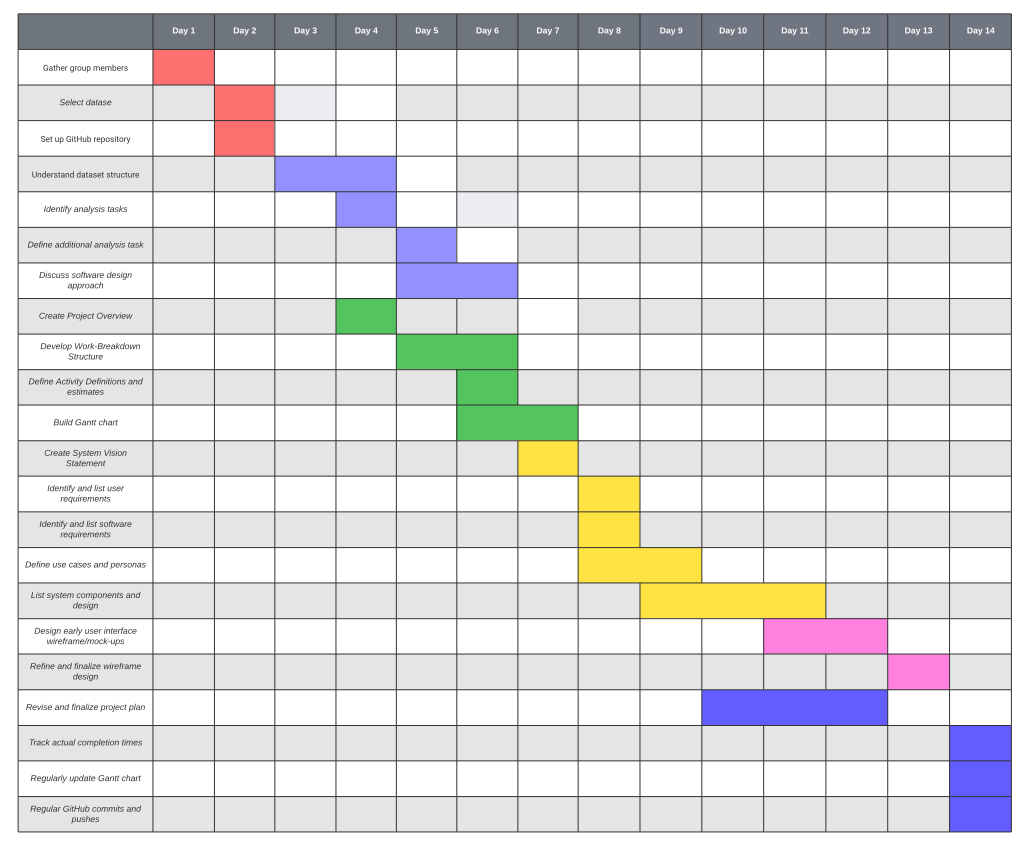
1. Regular GitHub Commits and Pushes (1 Day):

* Maintain a regular schedule of committing and pushing code to the GitHub repository.
* Ensure that team members follow version control practices consistently.
* Document commits and pushes to track changes and contributions.

# Gantt Chart

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.*

**