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

The project's goal is to create a piece of software that examines and offers insights from Sydney Airbnb data. The dataset, which comes from Kaggle, comprises facts on Sydney-area Airbnb listings, including property information, cost, and client feedback. The software will include several features, including the ability to retrieve data regarding listings in each suburb for a user-specified time frame, create price distribution charts, look up records using keywords, analyse customer comments regarding cleanliness, and offer one additional insight or analysis tool.

In an era driven by data, where insights are the currency of informed decision-making, the significance of data analysis in shaping industries and facilitating growth cannot be overstated. The profound impact of data analysis is vividly showcased through the lens of the Sydney Airbnb market. As the vibrant city of Sydney continues to attract both tourists and residents alike, the world of Airbnb rentals within its bounds has emerged as a dynamic ecosystem. Understanding this ecosystem—comprising listings, pricing patterns, guest preferences, and other variables—holds the key to optimizing rentals, enhancing guest experiences, and maximizing property owners' returns.

The foundation of this project rests upon a carefully curated dataset sourced from Kaggle, a platform known for housing a plethora of real-world datasets. This dataset encapsulates a comprehensive spectrum of information about Airbnb listings in Sydney, ranging from property attributes to guest reviews. The potential insights lying dormant within this data serve as the bedrock for developing a sophisticated software application.

## Scope

The scope of this project encompasses the development of a software application to analyze the Sydney Airbnb data, focusing on specific features mentioned above. The software will allow users to interact with the data, visualize it through charts, and gain insights based on their queries. The project also includes creating project planning documentation and a software design document.

The scope of this endeavor stretches beyond the mere development of a software application; it encapsulates the birth of a tool that empowers users to unravel the complex tapestry of Sydney's Airbnb market. At the heart of this endeavor lies the creation of a user-centric platform that not only provides an avenue for data exploration but also presents it in a visually intuitive manner.

This project will embrace a multifaceted approach, encompassing not only the technical development of the software but also meticulous planning and strategic design. The software's core functionalities will allow users to conduct advanced data queries, visualize data trends, and gain valuable insights that can guide their decision-making processes. By fostering a seamless interaction between users and data, the software bridges the gap between raw data and actionable intelligence.

Furthermore, this project will extend its reach to documentation. A comprehensive project plan will meticulously outline the journey from inception to deployment. This plan will act as a guiding compass, ensuring that every phase of the project is well-orchestrated and that progress is tracked effectively. In tandem, a detailed software design document will lay the blueprint for the software's architecture, user interfaces, and system components.

The scope, thus, is not only to create a software application but to craft a comprehensive ecosystem that amplifies the value of data. This ecosystem will empower users with the tools to make well-informed decisions, property owners to optimize their listings, and data enthusiasts to explore the nuances of one of the world's most exciting Airbnb markets.

In essence, this project's scope extends beyond lines of code; it encompasses the power to transform data into insights and insights into action. Through a harmonious blend of technical prowess, strategic planning, and user-centered design, this project sets out to create a tangible impact in the realm of data analysis and Airbnb market understanding.

## Document contents

This project plan document outlines the overall plan for the development of the software application. It includes a Work Breakdown Structure (WBS) to breakdown the project into manageable tasks, Activity Definition and Estimation to estimate the effort required for each task, and a Gantt chart to schedule and track the project progress.

This project plan is a meticulously curated guide that illuminates the path from conceptualization to realization. Each section has been thoughtfully designed to provide clear insights and actionable steps. Here's a glimpse of what's in store:

Work Breakdown Structure: This section presents a detailed hierarchical breakdown of tasks, offering a bird's-eye view of the project's architecture and scope.

Activity Definition & Estimation: Dive into the specifics of each task, from purpose to expected outcomes, all while gaining an understanding of the time required for successful completion.

Gantt Chart: Visualize the project's timeline and task dependencies in a dynamic Gantt chart, translating the project plan into an easily digestible format.

Intricately woven together, these sections provide a comprehensive understanding of the project's intricacies and strategic approach. Whether you're a team member, stakeholder, or project manager, this document serves as a guiding compass, leading you through a journey of discovery, development, and innovation

*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.*

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 |
| **Update Project Plan** | | | |
| 19 | Revise and finalize project plan | 3 Days | - |
| 20 | Track actual completion times | 1 Day | 10 |
| 21 | Regularly update Gantt chart | 1 Day | 10 |
| 22 | Regular GitHub commits and pushes | 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):

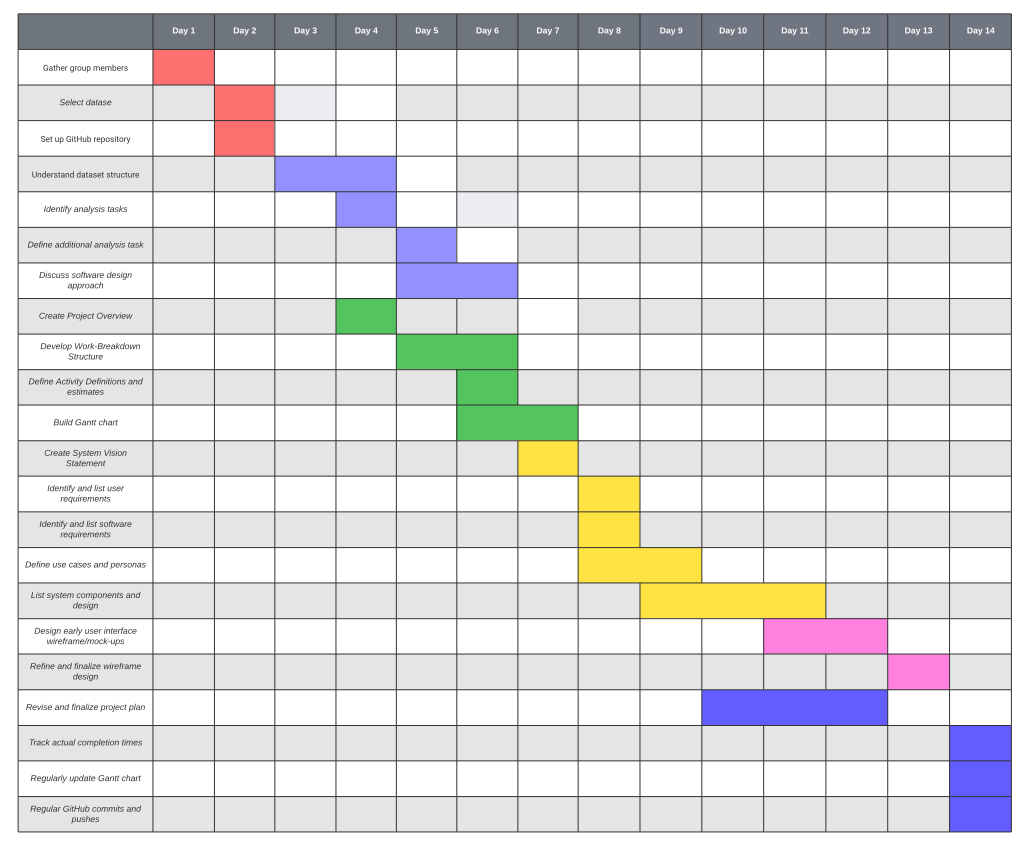
* Continuously update the Gantt chart to reflect completed tasks and milestones.
* Adjust the chart to account for any changes in task dependencies or timelines.
* Keep the Gantt chart accessible and shareable for all team members.

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

*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.*

**