Project Plan

**Airbnb Data Analysis Software**

**Student Names:**

Sarah Mitchell s5316578

Lina Kim s5188279

Sonya Jung s5276105

Table of Contents

[1.0 Introduction 2](#_Toc991249392)

[1.1 Background 3](#_Toc1226106345)

[1.2 Scope 3](#_Toc1316070197)

[1.3 Document contents 3](#_Toc58907274)

[2.0 Work Breakdown Structure 3](#_Toc962232036)

[3.0 Activity Definition & Estimation 5](#_Toc1907384245)

[4.0 Gantt Chart 8](#_Toc1490801526)

[5.0 Design (updated) 8](#_Toc18127114)

# Introduction

## Background

In this project, we are creating a user-friendly data analysis and visualisation tool tailored to the Sydney Airbnb Open dataset. This dataset contains a wealth of information about Airbnb listings in Sydney, encompassing property details, pricing, availability, and guest comments. Our primary goal is to design and implement a graphical user interface (GUI) that simplifies the process of extracting insights from this dataset. The tool will offer several crucial features, all of which will be spoken of in depth in the scope of the project.

The tool encompasses a range of essential features, which will be elaborated upon in detail within the project scope. These features aim to cater to diverse user needs, from travellers seeking the perfect stay to property owners looking to optimize their listings, and researchers analysing trends in the Sydney Airbnb market.

Our development approach covers software engineering, data analysis, and user experience design. We will leverage Python and popular libraries like pandas, matplotlib, and wxPython to craft an intuitive interface. Additionally, data preprocessing and analysis algorithms will be employed to uncover meaningful insights from the dataset.

The ultimate measure of success for this project will be twofold: the tool's usability and the value it brings to users. Our aim is to empower a wide-ranging user base, including travellers seeking memorable experiences, property owners striving to enhance their offerings, and researchers exploring the dynamics of the Airbnb market in Sydney. By facilitating data-driven decision-making, we envision not only simplifying the process but also enriching the overall Airbnb experience in this vibrant city.

## Scope

The interface will enable users to perform several visualisation and analysis tasks, including (for a user-selected period) reporting the information of all listings in a specified suburb, producing a chart to show the distribution of prices of properties and retrieving all records containing a user entered keyword such as ‘pool’ or ‘pet.’ In addition, the interface will analyse how many customers commented on factors related to cleanliness and only show the user properties that’s prices are within the minimum and maximum the user inputted.

## Document contents

This document will contain the Work Breakdown Structure (WBS), activity definition and estimation, and a Gantt chart. The WBS will provide a hierarchal illustration of the project, dividing it into smaller, more feasible tasks. Activity definition and estimation are pivotal procedures involving the identification and description of the specific tasks required to complete the project. Additionally, they assist in determining the essential resources, time, and effort for each of these tasks. The Gantt chart will offer a visual representation of the project’s timeline, incorporating task sequencing, resource allocation, progress tracking and other key information.

# Work Breakdown Structure

1. Initiating
   1. Assign project manager
2. Planning
   1. Create Scope Plan
   2. Create WBS and Schedule
   3. Software Requirements
   4. User Requirements
   5. Use cases & diagrams
3. Designing

3.1 UI design

3.2 Wireframes

3.3 Problem backgrounds

3.4 Software Design

3.5 System Overview

3.6 Potential Benefits

3.7 System components

1. Executing

3.1 Collect the raw data

3.2 Retrieve the raw data

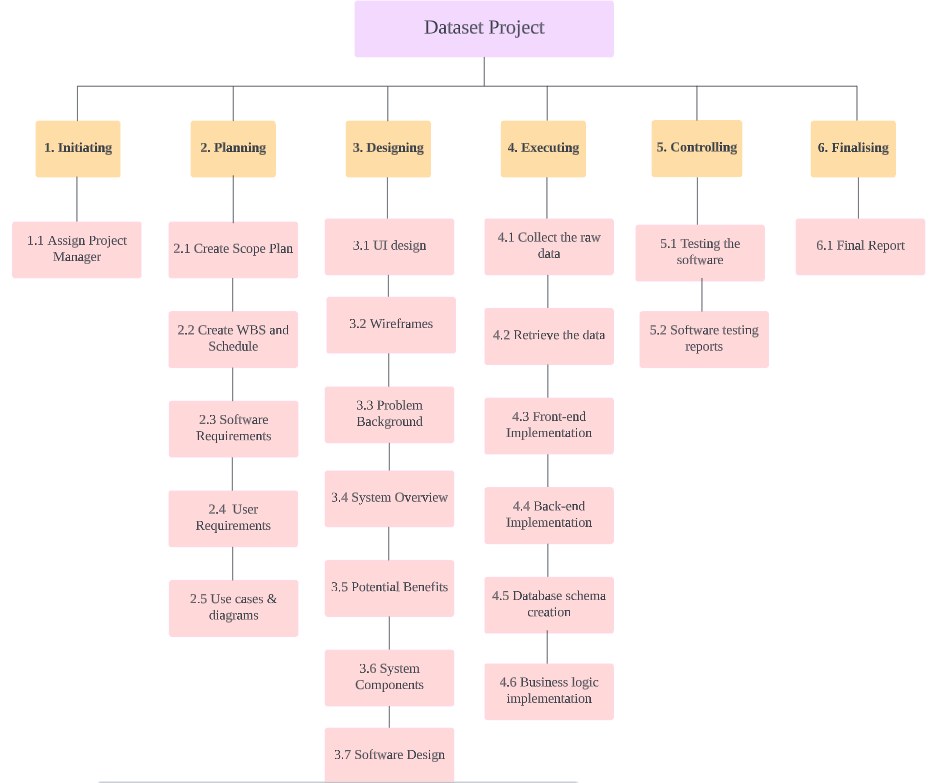
3.3 Front-end implementation

3.4 Back-end implementation

3.5 Database Schema Creation

3.7 Business logic implementation

1. Controlling
   1. Testing the software
   2. Software testing reports
2. Finalising
   1. Final Report



# Activity Definition & Estimation

1. Initiating
   1. Assign a project manager

* Activity Definition: Select and assign a project manager to oversee the project’s planning, execution, and completion.
* Time Estimation: 1 day.

1. Planning
   1. Create Scope Plan

* Activity Definition: Defining the scope of the project.
* Time Estimation: 7 days.
  1. Create WBS and Schedule
* Activity Definition: Break down the project into tasks and create a detailed project schedule.
* Time Estimation: 7 days.
  1. Software Requirements
* Activity Definition: Gather and document the software requirements that outline what the software needs to achieve.
* Time Estimation: 8 days.
  1. User Requirements
* Activity Definition: Gather and document the user requirements of the project to ensure the software meets user expectations and needs.
* Time Estimation: 8 days.
  1. Use cases & diagrams
* Activity Definition: Create use case scenarios and system diagrams based on the user requirements.
* Time Estimation: 8 days.

1. Designing
   1. UI design

* Activity Definition: Design the interface for the software.
* Time Estimation: 2 days.
  1. Wireframes
* Activity Definition: Develop wireframes to visualise the layout and structure of the user interface.
* Time Estimation: 3 days.
  1. Problem backgrounds
* Activity Definition: Research and understand the problems that the software aims to address.
* Time Estimation: 3 days.
  1. System Overview
* Activity Definition: Develop a system overview that will provide a high-level understanding of the software architecture, main components, and their interactions.
* Time Estimation: 5 days.
  1. Potential Benefits
* Activity Definition: Identify and document potential benefits that the software can bring to stakeholders.
* Time Estimation: 5 days.
  1. System components
* Activity Definition: Identify and document the various system components of the software.
* Time Estimation: 5 days.
  1. Software design
* Activity Definition: Develop the detailed software design. This will include its architecture, data flow, interfaces, and algorithms.
* Time Estimation: 3 days.

1. Executing
   1. Collect the raw data

* Activity Definition: Using Kaggle, download the CSV files for the Sydney Airbnb dataset.
* Time Estimation: 1 day.
  1. Retrieve the data
* Activity Definition: Using SQLite, input the Sydney Airbnb data into the software.
* Time Estimation: 5 days.
  1. Front-end Implementation
* Activity Definition: Develop the front-end components and user interface based on the UI design and wireframes.
* Time Estimation: 10 days.
  1. Back-end Implementation
* Activity Definition: Develop and implement the back-end logic, server-side components and functionality required for the software’s operation.
* Time Estimation: 10 Days.
  1. Database Schema creation
* Activity Definition: Design and create the database schema that will store and manage the software's data.
* Time Estimation: 1 Week.
  1. Business logic implementation
* Activity Definition: Implement the core business logic that drives the software's operations.
* Time Estimation: 3 days.

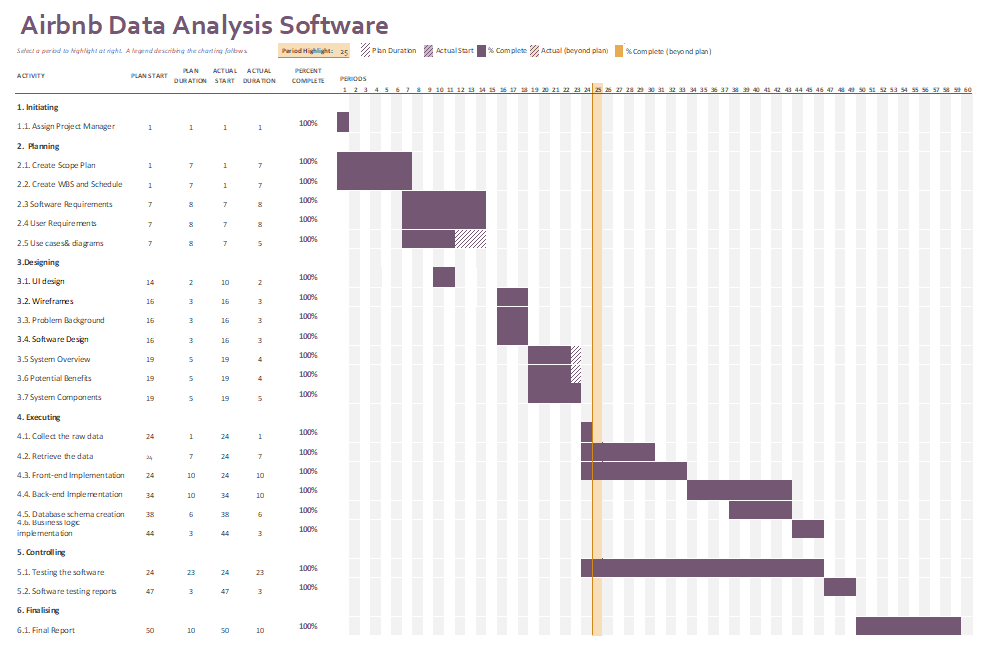
1. Controlling
   1. Testing the software

* Activity Definition: Conduct thorough testing of the software to identify and fix any bugs or issues.
* Time Estimation: 23 days.
  1. Software testing reports
* Activity Definition: Generate testing reports to document the testing process and outcomes, including any bugs or errors that have been fixed.
* Time Estimation: 3 days.

1. Finalising
   1. Final Report

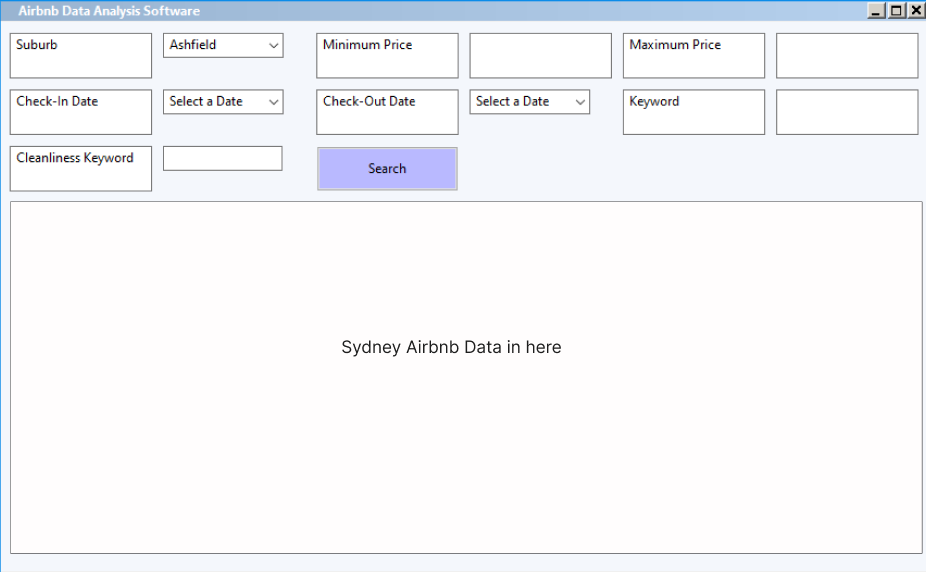
* Activity Definition: Prepare a final report that includes the problem background, planning, execution, controlling and assessed outcomes.
* Time Estimation: 10 days.

# Gantt Chart



# 5.0 Design (updated)

Initially, the design of the Airbnb system was as follows:



However, one page containing all the requirements can make it more difficult by having to keep track of all the fields listed on top. This not only slows down the process of reading and loading a file but also, make it harder to achieve the requirement goals. Instead, when the user enters the requirement button, only corresponding fields shows.

