Project Plan

Sydney’s Airbnb Data App

**Assignment Group 31**

Fizza Batool - s5160137

Uyen Nguyen - s5306965

Table of Contents

[1.0 Introduction 3](#_Toc144029737)

[1.1 Background 3](#_Toc144029738)

[1.2 Scope 3](#_Toc144029739)

[1.3 Document contents 4](#_Toc144029740)

[2.0 Work Breakdown Structure 5](#_Toc144029741)

[3.0 Activity Definition & Estimation 6](#_Toc144029742)

[4.0 Gantt Chart 9](#_Toc144029743)

[5.0 Bibliography 10](#_Toc144029744)

# Introduction

## Background

Sydney was announced as the 4th top city in the world in 2016. Its reputation has spread globally, and its name has become prominent in the world today. To enable continued growth and economic benefits for Sydney, its residents and visitors, a technology application will be created as part of the ‘InsideAirbnb’ initiative to describe the listing activities of residents in Sydney, Australia. The Kaggle.com dataset of Airbnb activity in Sydney will be used and integrated in the software application to report on all listings in a specific suburb, create a distribution of prices chart, retrieve records by keyword, and retrieve comments related to factors keywords such as cleanliness, along the number of listings in a specified suburb. As such, the Griffith University ICT team (group #) has been assigned the job of creating a software application designed to extract the dataset and retrieve information to make informed decisions about Sydney’s Airbnb market for clients such as Ray White and other real estate agencies, making assumptions about the changed tourism landscape of Australia in relation to increased concentration of visitors, rental and accommodation prices, and overall cleanliness issues and other problems associated with specific suburban areas.

## Scope

The application will include the design of a user-friendly interface with the following functions:

* For a user-selected period, report the information of all listings in a specified suburb.
* For a user-selected period, produce a chart to show the distribution of prices of properties.
* For a user-selected period, retrieve all records that contain a keyword (user entered), e.g., pool, pet.
* Retrieve how many customers commented on factors related to cleanliness (multiple key words may be associated with cleanliness).
* For a user-selected period, report the number of listings in a specified suburb.

The application software is developed to work on Windows 10 Operating Systems. Any machine running this OS will be able to run this piece of software on their local device. The focus of this software will be to disseminate information by retrieval and visual display. In addition, the user will be able to define search parameters that to analyse records which contain particular words.

There will be three members working together on the application using Python 3, wxBuilder, SQL and Excel. ICT group member will be working on one function and using GitHub to post changes and modifications. The hours of work required for each function is displayed below:

**Figure 1. Details of Application Features and Work Hours**

|  |  |
| --- | --- |
| Reporting listings of specified suburb | 1 person required/30hrs of work |
| Produce a chart of distribution of prices | 1 person required/30hrs of work |
| Retrieve records of keywords | 1 person required/30hrs of work |
| Retrieve number of comments on cleanliness | 1 person required/30hrs of work |
| Retrieve number of listings in a specified suburb | 1 person required/30hrs of work |

## Document contents

As part of a collaboration with Griffith University’s software development ICT students, the dataset for Sydney’s Airbnb will be researched and used to prepare for an interactive application with the required features of reporting information of all listings in a specified suburb, displaying a chart related to the distribution of property prices in specified suburbs, retrieving all records that contain user entered information, analysing how many customers commented on cleanliness, and the number of listings in a specified suburb. In this document, the ‘Project Plan’ contains a work-breakdown structure which records the list of features for the application and various tasks required to complete those features. This document also contains an estimated timeframe for each task’s completion along with their definitions. Lastly, this document contains a visual representation through a Gantt chart which represents the schedule.

# Work Breakdown Structure

|  |  |  |
| --- | --- | --- |
| **Work Breakdown Structure** | | |
| **Task ID** | **Task Component** | **Status (Not started, In progress, Done)** |
| **Initialisation 1.0** | |  |
| 1.1 | Meet with group members | Done |
| 1.2 | Debrief project | Done |
| 1.3 | Installation and dataset | Done |
| 1.4 | Assign roles and responsibilities | Done |
| 1.5 | Documentation | Done |
| **Planning and Analysis 2.0** | |  |
| 2.1 | Background | Done |
| 2.2 | Assess and identify available data | Done |
| 2.3 | Create Work-Breakdown Structure | Done |
| 2.4 | Activity definition and estimation | Done |
| 2.5 | Gantt chart | Done |
| **Design 3.0** | |  |
| 3.1 | System vision | Done |
| 3.2 | System overview | Done |
| 3.3 | Present benefits | Done |
| 3.4 | Requirements/user requirements, software requirements | Done |
| 3.5 | Use cases | Done |
| 3.6 | System component and software design | Done |
| 3.7 | User interface design | Done |
| 3.8 | Structural design | Done |
| 3.9 | Visual design | Done |
| **Development 4.0** | |  |
| 4.1 | Develop user data | Done |
| 4.2 | Create user interface | Done |
| 4.2 | Create algorithms to produce features | Done |
| 4.3 | Implement coding logic | Done |
| 4.4 | Log and upload | Done |
| 4.5 | Revision and testing | Done |
| **Testing 5.0** | |  |
| 5.1 | Unit testing | Done |
| 5.2 | Closed beta testing | Done |
| 5.3 | Manual and document testing | Done |
| 5.4 | User accepting testing | Done |
| 5.5 | Report and documentation | Done |
| **Deployment 6.0** | |  |
| 6.1 | Provide user manual | Done |
| 6.2 | Deploy | Done |
| 6.3 | Analyse and document the software | Done |

# Activity Definition & Estimation

|  |  |  |
| --- | --- | --- |
| **Activity Definition & Estimate** | | |
| **Task ID** | **Task Definition** | **Task Duration** |
| **Initialisation** | | |
| 1.1 | Meet with group members | 1 day |
| 1.2 | Debrief project | 1 day |
| 1.3 | Install required software and retrieve dataset | 1 day |
| 1.4 | Assign roles and responsibilities | 1 day |
| 1.5 | Define programming and documentation requirements (GitHub) | 1 day |
| **Planning and Analysis** | | |
| 2.1 | Introduce the project background | 1 day |
| 2.2 | Assess the data collected including the data necessary to complete each feature | 1 day |
| 2.3 | List out all the project activities in a hierarchy chart | 1 day |
| 2.4 | Define the tasks in the WBS and estimate the time needed to complete each task | 1day |
| 2.5 | Display the general schedule and track the actual time of each task | 1 day |
| **Design** | | |
| 3.1 | Introduce the system vision for this software | 1 day |
| 3.2 | Describe the deliverables, functions, data,  and information | 1 day |
| 3.3 | Briefly identify the potential benefits of this project for customers and clients | 1 day |
| 3.4 | Introduce the requirements, describe the requirements of this application in relation to users and this software | 2 days |
| 3.5 | Identify Use Case(s) and create Use case diagrams | 2 days |
| 3.6 | Describe the software functions, data structures/data sources, and present several detailed designs of software through pseudocode | 2 days |
| 3.7 | Describe the tools used for this design | 2 days |
| 3.8 | Define the navigation layout and structure of the software | 1 day |
| 3.9 | Detailed visual design such as layout, visual elements, icons, graphics, style, colour, font, and general screen design | 3 days |
| **Development** | |  |
| 4.1 | Collect dataset from Kaggle.com and transfer data into database | 5 days |
| 4.2 | Create the software interface for both the main menu and feature pages through python | 3 days |
| 4.3 | Use of Python and SQL query to create the algorithms, sorting, and retrieval of data,  then create functions to display data in  graphs | 1 day |
| 4.4 | Use of dev team to apply logic to conditions to all functions | 14 days |
| 4.5 | Upload code to GitHub; log findings and errors, clearly label and comment on code | 1 (cont’d) |
| 4.6 | Revise and test code, identify issues, continue development, and apply different scenarios to test each feature | 1 day |
| **Testing** | |  |
| 5.1 | Review all code, functionality, and apply final testing scenarios to test the software | 1 day |
| 5.2 | Present the application to closed beta testers (customers) | 1 day |
| 5.3 | Report and document of all testing conducted in beta including a checklist of successes and errors encountered | 1 day |
| 5.4 | Present the application to the client(s) for open beta testing | 1 day |
| 5.5 | Create and finalise report and documentation of software | 2 days |
| **Deployment** | |  |
| 6.1 | Create user manual (READ.ME) file | 1 day |
| 6.2 | Finalise software and deploy it to the public (clients(s)) | 1 day |
| 6.3 | Analyse the success of the software reflecting on customer and client feedback, document the feedback | 1 day |

\*There are only slight changes made with variation of a day, but no significant change of plan was witnessed.

3.5 The change in use case was made to have better understanding of the requirement so that the client can be delivered a satisfying product.

3.6 Software design had some feedback, therefore there was further time taken to complete the task.

4.1 There were a lot of csv files to decide from which ones we needed the most and compliment the code with them.

5.5 Documentation was slightly lengthy as the function 5 needed to be decided.

The initialisation phase involves identifying the problem and creating a project charter which is a formal document outlining the project purpose and scope, understanding the project, and meeting with group members for follow-ups. This activity is estimated to take approximately two weeks. In this phase, the team will get to know each other through in-person laboratories and other communicating platforms such as Teams. Weekly expectations and goals will be set up and must be discussed. Each student will outline their expectations and abilities. Students will formalise their group agreement on L@Griffith. A GitHub repository will also be created and used to store the assignment task, and each group member will be added to the repository with full-privileged permissions. Each member must pull this repository to their local system.

During the planning and analysis phase, the ICT project team will discuss the dataset by retrieving and viewing the data from Kaggle.com. The group will evaluate the viability of each dataset and choose the dataset according to viability. The project overview, background, and scope must be written in this phase. The project plan will be researched and addressed, and the contents will be documented. Approximately two weekly meetings will take place after each week’s laboratory to discuss the written project plan.

The design phase involves the collaborative effort of the team to design the software application and its user interface and provide a detailed system design of what this piece of software will look like along with what each component should be able to do. This phase comprises the initial and end software design. The design stage is estimated to take about one and a half weeks and each software component and cog will be documented. The software design will be shown visually.

The development phase encompasses the construction of the user interface design and backend logic. This is somewhat a time-consuming process spanning across three to four weeks. The key role involves the development of the interface using wxBuilder and wxPython. The software design must be followed including the implementation of each component outlined in the design phase. The parameters will also be defined in addition to the functions for each cog in the interface of the software application and transferred to each team member’s local Python IDE.

The testing phase will allow the software to gauge its relative success in the public along with identifying any errors in its functionality. This step is completed to ensure the software is aligned with the requirements of the users. This step is also completed to ensure feedback of bugs and rectification. If any identified errors are observed, they are reported, logged, and expected to be fixed before the final release of the software to major clients. This testing phase spans across one week of the project’s life cycle.

Deployment marks the final phase of the project’s life cycle which involves the release of the software from production and into the public. This step is expected to take a couple of days and involves the distribution of the user manual, deployment, and analysis of the success of the software through ongoing customer and client feedback.

# Gantt Chart



# Bibliography

Xie, T. (2016). Sydney Airbnb Open Data [Public raw data]. Retrieved from <https://www.kaggle.com/datasets/tylerx/sydney-airbnb-open-data>