Software Design Document

<Sydney Airbnb Database Analysis>

Chelzie Castanares – s5259144

Thien Thao My Bui – s5273753

Table of Contents

[1.0 System Vision 3](#_Toc46748622)

[1.1 Problem Background 3](#_Toc46748623)

[1.2 System Overview 3](#_Toc46748624)

[1.3 Potential Benefits 3](#_Toc46748625)

[2.0 Requirements 4](#_Toc46748626)

[2.1 User Requirements 4](#_Toc46748627)

[2.2 Software Requirements 4](#_Toc46748628)

[2.3 Use Cases 4](#_Toc46748629)

[3.0 System Components and Software Design 5](#_Toc46748630)

[3.1 System Components 5](#_Toc46748631)

[3.2 Software Design 5](#_Toc46748632)

[4.0 User Interface Design 6](#_Toc46748633)

# System Vision

## Problem Background

Airbnb is a global business that specialises in providing accommodation but differs from traditional hotels and resorts. Airbnb acts a marketplace between homeowners who wish to rent out their place for accommodation and tourists looking for a place to stay. As the popularity of Airbnb increases worldwide, the need of updated systems is more prevalent within their business. Sydney is one of the most popularized tourist destinations within the world ranking at 4th in 2016, and therefore has been chosen as the subject to perform Airbnb data analysis using a visualization tool to understand and improve the business structure to increase efficiency.

Despite of the affordable price and the diversity of lodging options, local and international tourists visiting Sydney still have trouble finding their ideal Airbnb place due to an enormous list of all properties for rent across the whole city. Taken from a well-known source InsideAirbnb, a dataset collecting detailed listing activity of homestays in Sydney during 12/2018 - 12/2019 will be used to find out valuable insights for the tourists. Depending on the renter’s selected period and preference, the implemented system may help sorting out among a massive amount of available Airbnb accomodations and simplify the search process. This would ideally enhance the booking experience and ensure travellers to Sydney plan their vacation easier and have a more comfortable stay. In addition, for local businesses and individuals who run Airbnb properties in Sydney, this system would support them in property marketing and management strategies based on customer feedbacks as well.

## System Overview

The system should be able to perform a range of capabilities that’ll enable users to analyse and display results from Sydney’s Airbnb data based on selecting criteria, narrowing down the options into the most suitable listings.

The following include the system capabilities identified:

* Within a user-selected period, return all the listings in a specified suburb.
* Within a user-selected period, display the distribution of prices of properties using a chart.
* Within a user-selected period, retrieve all records that contain a specific keyword (e.g., pool, 2-bedroom).
* Analyse the comments relating to cleanliness of the property such as (dust, dirt etc.).
* Return all the reviews of a user-selected property.

## Potential Benefits

With the use of the of the data visualisation tool, it will help Airbnb customers understand the data behind Sydney’s homestay activity listings, which in turn will also assist the businesses to understand their consumers better. The following system will allow both tourists and private businesses to analyse popular locations of renting Airbnb’s and how well the previous customers enjoyed their stay based on reviews and feedbacks. This can enable the business to produce more value for consumers as it aims to improve problems based on data analysed.

The system is implemented with the intention to enhance the booking experience and ensure travellers to Sydney plan their vacation easier and have a more comfortable stay. In addition, for local businesses and individuals who run Airbnb properties in Sydney, this system would support them in property marketing and management strategies by identifying problems and tackling issues.

# Requirements

## User Requirements

In this section you detail how a user is supposed to interact with or use your program. What do they ***need*** to be able to do? This should all be from the end users perspective. Can be a combination of narrative text and listing of needs.

**Assignment note: You have not been given a client/user, so you can make one up. Who do you think would be using your software?**

Our client Airbnb has listed specific requirements to be included. The following system should include the following user requirements that enable the business to work on the issue.

|  |  |
| --- | --- |
| Requirements | Description |
| Property Search | The user is able to input the location of stay within their selected period date. |
| Property Price Search | The user is able to see the cost of property within their selected period date. |
| Keyword Search System | The user is able to search for properties containing the specific keyword/s they are looking for. |
| Comments Search System | The user is able to view all comments of selected property within their selected period date. |
| Review Search System | The user is able to look at the reviews of a certain property. |

## Software Requirements

In this section you detail what the requirements for the software are. What functionality will it provide? This is usually a formal listing, with requirements often using the word ‘Shall’. IE:

R1.1 The program shall accept multiple file names as arguments from the command line.

R1.2 Each file name can be a simple file name or include the full path of the file with one or more levels.

etc …

Can be primarily functional requirements, though you may include other types if you think of them.

The following are non-functional requirements that focus on FURPS+ (Functionality, Usability, Reliability, Performance and Security):

|  |  |
| --- | --- |
| Non - Functional Requirements | |
| FURPS+ Category | Description |
| Usability | * User interface should be easy to use and navigate. * All requirements should be working smoothly with no errors. |
| Reliability | * System should be running at all times. * System should be error free. |
| Performance | * System response time should be quick when retrieving data e.g., 5 secs. * Should be able to store large amounts of data. |
| Security | * Website should be encrypted with HTTPS Protocol. |

## Use Cases & Use Case Diagrams

In this section you provide some use cases showing how people may use your software

A diagram of a person's search

Description automatically generated

|  |  |
| --- | --- |
| **Use Case ID** | 1 |
| **Use Case Name** | Property Search |
| **Actors** | User (Tourist) |
| **Description** | The user will be able to search for a property in an area within their selected dates. |
| **Flow of events** | 1) The user will access the system.  2) The user will select the period of date.  3) The user will input the suburb of stay.  4) The system will display all accommodation in that area within the selected period. |
| **Alternative Flow** | None |

|  |  |
| --- | --- |
| **Use Case ID** | 2 |
| **Use Case Name** | Property Price Search |
| **Actors** | User (Tourist) |
| **Description** | The user will be able to find property to stay in that fits their ideal budget range within their selected dates. |
| **Flow of events** | 1) The user will access the system.  2) The user will select the period of date.  3) The system will display all properties and their prices within selected period. |
| **Alternative Flow** | None |

|  |  |
| --- | --- |
| **Use Case ID** | 3 |
| **Use Case Name** | Keyword Search |
| **Actors** | User (Tourist) |
| **Description** | The user will be able to find all properties within their selected dates that match their ‘keyword’ description entered. |
| **Flow of events** | 1) The user will access the system.  2) The user will select the period of date.  3) The user will input desired ‘keywords’ that they wish to have in the property.  3) The system will display all properties that match ‘keyword’ description within the selected period. |

|  |  |
| --- | --- |
| **Use Case ID** | 4 |
| **Use Case Name** | Comments Search |
| **Actors** | User (Tourist) |
| **Description** | The system will show all comments relating to cleanliness of all properties within their selected dates. |
| **Flow of events** | 1) The user will access the system.  2) The user will select the period of date.  3) The system will display all comments relating to cleanliness for all properties within the selected period. |
| **Alternative Flow** | None |

|  |  |
| --- | --- |
| **Use Case ID** | 5 |
| **Use Case Name** | Review System |
| **Actors** | User (Tourist) |
| **Description** | The user is able to see all reviews on a selected property within their selected dates. |
| **Flow of events** | 1) The user will access the system.  2) The user will select the period of date.  3) The user will select a property.  3) The system will return all reviews on that property. |
| **Alternative Flow** | None |

# Software Design and System Components

## Software Design

A block diagram/flowchart of how your software might work

A diagram of a flowchart

Description automatically generated

Change enter desired keyword and display

Change review

## System Components

### Functions

Preliminary list of all functions in the software. For each function in the list the following information is provided:

* a brief description of what it does (1 or 2 sentences);
* a list of the input parameters, and their data types, and what they are used for;
* a list of any side effects caused by the function (ie change global or member variables, changes data passed by reference from calling function etc)
* a description of the function’s return value

*Function used: Loading data (pd.read\_csv – can be loaded from web or local computer/might not work for other file formats like html, sql*

*Extracting basic info – df.info(), df.describe -> identify feature types, statistical analysis and detect missing value if happened*

*Df.sort\_values -> sorting dât based on a specific criteria in ascending/descending order*

*Df.str.contains() // df.isin(*

|  |  |
| --- | --- |
| **Function** | loadData |
| **Description** | Inputting and loading csv data file from a website link or local computer (based on pandas library in python) |
| **Input parameters** | A website link / Directory to the csv file location in the computer |
| **Side effects** | None, as it is a read-only operation and should not modify the data in external sources |
| **Return value** | A DataFrame representing all the listings of the data |

|  |  |
| --- | --- |
| **Function** | priceReport |
| **Description** | Retrieving price of all the Airbnb listings in the database based on user’s chosen period |
| **Input parameters** | * Starting date * Ending date |
| **Side effects** | None |
| **Return value** | A visual representation (scatter plot/pie chart) to inform a summary report of the pricing range in the specified period |

|  |  |
| --- | --- |
| **Function** | areaFilter |
| **Description** | Retrieving specific Airbnb listings in the database based on user’s chosen suburbs |
| **Input parameters** | * Starting date * Ending date * Suburb name (not case sensitive |
| **Side effects** | None |
| **Return value** | A list of Airbnb properties that meet the specified period and location |

|  |  |
| --- | --- |
| **Function** | keywordSearch |
| **Description** | Searching for properties containing the specific keyword and return matching results |
| **Input parameters** | * Starting date * Ending date * Keywords |
| **Side effects** | May result in inconsistent results due to the case-sensitive context or typo mistake from either the input parameters or the description of the properties |
| **Return value** | A DataFrame containing all the matching results, including the Airbnb name (Id) and their related records from customers or the business’s descriptions. |

|  |  |
| --- | --- |
| **Function** | commentSearch |
| **Description** | Retrieving the chosen property details with all the existed comments relating to cleanliness criteria |
| **Input parameters** | * Property ID * Keywords (ex: “dust”, “trash”, “dirt”, “clean”, “tidy”, “hygiene”, “sanitary”,…) |
| **Side effects** | Relying on keywords would not collect all the related comments, might miss some text data |
| **Return value** | A DataFrame representing the details of the matching comments |

|  |  |
| --- | --- |
| **Function** | reviewReport |
| **Description** | Retrieving all the available reviews and review scores regarding the selected property |
| **Input parameters** | Property ID |
| **Side effects** | None |
| **Return value** | * A DataFrame listing all the reviews and feedbacks from the customers. * A DataFrame regarding the average (mean) and mode of the review scores |

### Data Structures / Data Sources

List of all data structures in the software (eg linked lists, trees, arrays etc) or eternal data sources. For each data structure in the list the following information is provided:

* Type of structure (tree, list etc),
* Description of where and how it is used
* List of data members, and what each one is for do
* List of functions that use it

|  |  |
| --- | --- |
| **Types** | DataFrame |
| **Description** |  |
| **Data members** | * Column index: a list of strings (variable names) * Allowing for referencing and retrieval of data through unique labels * Row index: a list of integers (row numbers) / strings (propertyID) * Access and identify specific rows in the DataFrame * Series of data: |
| **Used functions** |  |

|  |  |
| --- | --- |
| **Types** |  |
| **Description** |  |
| **Data members** |  |
| **Used functions** |  |

### Detailed Design

Pseudocode for all non-standard / non-trivial algorithms that operate on data structures

Commenst of data = E.g get data

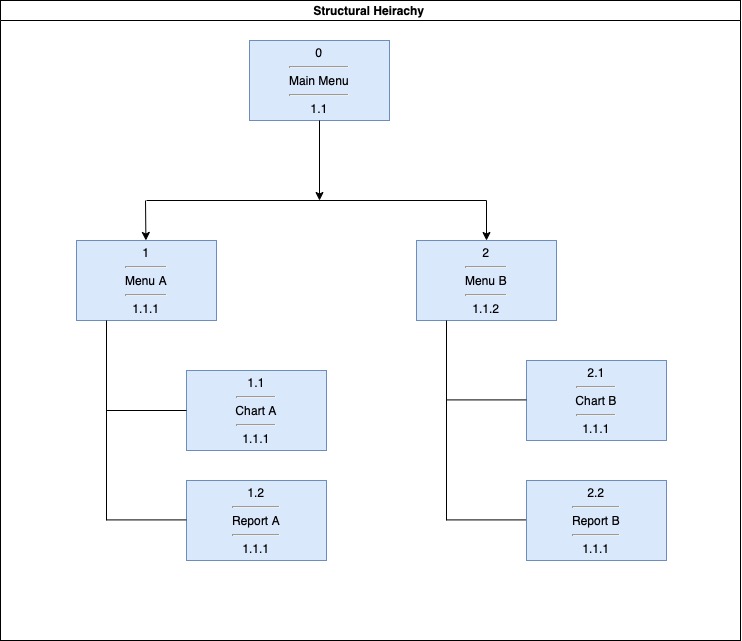
# User Interface Design

This is your initial interface design. Describe the tools you used for this design stage and any key findings that informed your design. This introduction is descriptive and should explain what you have completed for the actual design work you will present in the sub-sections below.

To increase usability for both the client (Airbnb) and the users, the following interface was designed with the idea of simplicity to make it easier to use and understand. The tools we used to achieve this was ‘draw.io’ a free online diagram software and ‘wireframe.cc’. Using this tool, we were able to create wireframes and a structural hierarchy to showcase the design of our system and how it will be used.

## Structural Design

Structural design refers to the navigational and information structure of your product – the structure that supports the interface layout. How will you structure your product? How will you group your information? How will you navigate through your product? Why? This can take the form of a diagram showing structure and hierarchy, supported by a discussion and justification of your choices. Why have you made these design choices? Describe and outline the structure of your interface and of your information.



The diagram above showcases the structural hierarchy for the Sydney Airbnb System. To ensure that our criteria of simplicity was met, we divided the requirements so that only two menu buttons would be needed to display the information. ‘Menu A’ will be the property details and will have ‘Chart A’ showing a graph of the accommodation prices all across Sydney. ‘Report A’ will display all accommodation available. ‘Menu B’ will display reviews and comments and will also have ‘Chart B’ which shows the review score of a property. ‘Report B’ will display all the comments of a selected property.

## Visual Design

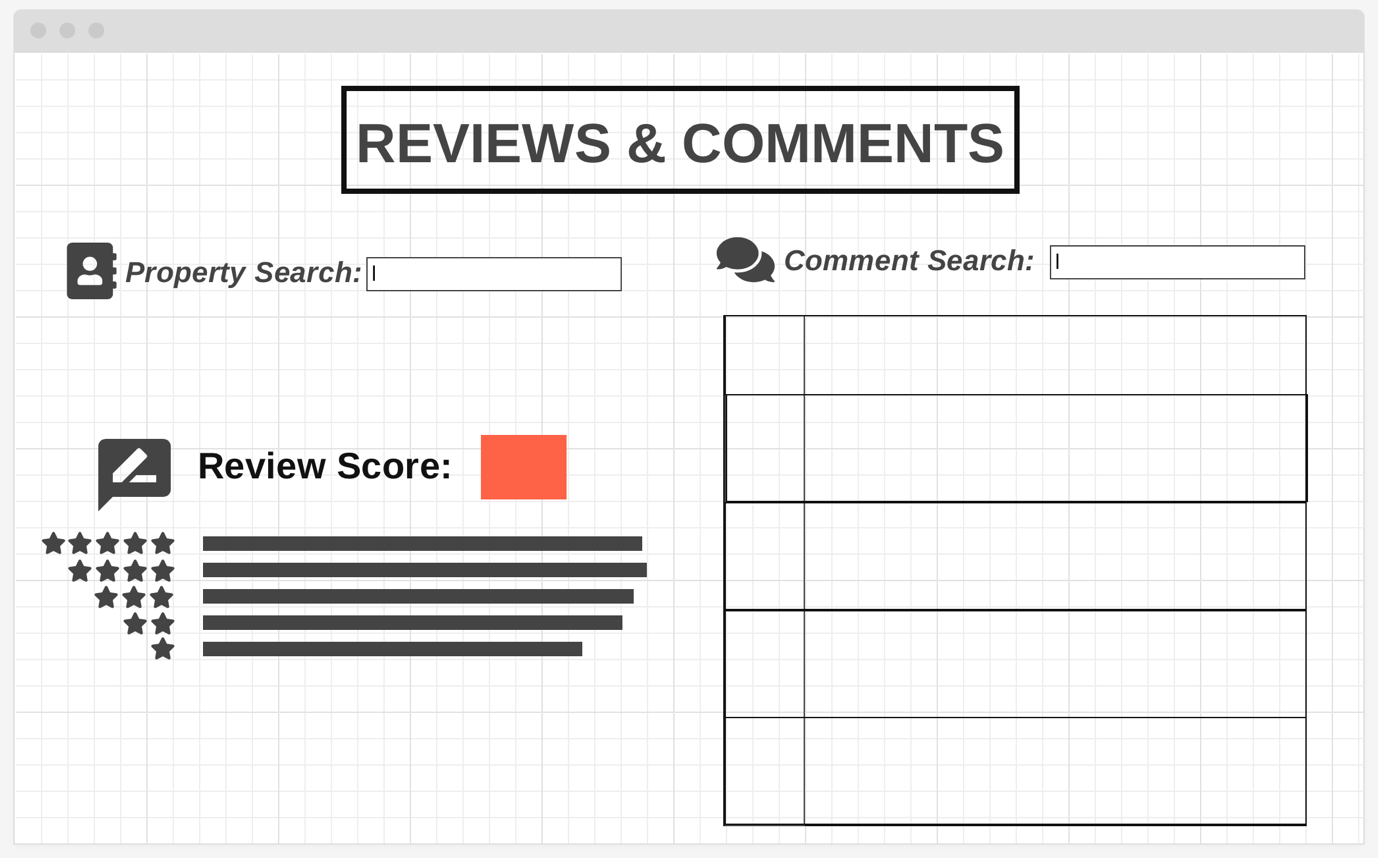
Detail your visual design: Layout, visual elements, icons, graphics, style, colour, fonts general screen designs. This can be sketches, wireframes, mockups etc, supported by a discussion, explanation, and justification of your choices.



The wireframe above is the ‘Main Menu’ and shows the overall layout of the system. As mentioned in the introduction, the design is made with simplicity to enable users to use the interface with ease. The design has all buttons at the top of the website and then displays the results at the bottom of the webpage.



The second wireframe above is the ‘Menu A’ and is the Property Details section. This section allows users to input a suburb and select a date period which in turn will show all the available properties. This webpage also allows users to compare the prices of accommodation across Sydney. Icons were used to guide users on what needs to be inputted.



The third wireframe above is the ‘Menu B’ and is the Reviews and Comments webpage. This section allows users to input view the review score of a selected property. It also allows users to find comments about a property such as the cleanliness.