Software Design Document

**Airbnb Data Analysis Software**

**Student Names:**

Sarah Mitchell s5316578

Lina Kim s5188279

Sonya Jung s5276105

Table of Contents

[1.0 System Vision 2](#_Toc385899282)

[1.1 Problem Background 3](#_Toc1662632199)

[1.2 System Overview 3](#_Toc5863750)

[1.3 Potential Benefits 4](#_Toc1150217504)

[2.0 Requirements 5](#_Toc991798057)

[2.1 User Requirements 5](#_Toc1496000822)

[2.2 Software Requirements 5](#_Toc1127058108)

[2.3 Use Cases & Use Case Diagrams 5](#_Toc1965625470)

[3.0 Software Design and System Components 7](#_Toc595478406)

[3.1 Software Design 8](#_Toc1055477680)

[3.2 System Components 8](#_Toc977147064)

[3.2.1 Functions 8](#_Toc80781565)

[3.2.2 Data Structures / Data Sources 10](#_Toc459916480)

[3.2.3 Detailed Design 12](#_Toc2146927893)

[4.0 User Interface Design 12](#_Toc401773592)

[4.1 Structural Design 13](#_Toc1968948346)

[4.2 Visual Design 13](#_Toc1339919918)

# System Vision

## Problem Background

Our project aims to address the need for an efficient and user-friendly way to search and book short-term rentals for travellers and backpackers worldwide who are planning to visit Sydney, Australia. This interface will streamline the process of finding suitable accommodation, taking into account factors such as availability, location, price and user preferences. By providing travellers with a centralised platform for accessing Airbnb listings in Sydney, they will be able to make informed decisions and secure their accommodation hassle-free.

To achieve this, the system will require the following data inputs and outputs:

Data input:

* Location Preferences: Users should be able to input their desired suburb or areas within Sydney where they would like to stay.
* Travel Dates: Users need to specify their check-in and check-out dates to find available accommodations for their intended stay duration.
* Budget Range: Users may input their budget constraints (minimum and maximum prices) to filter listings that fall within their financial considerations.
* Keywords: Users could search for desired amenities such as Wi-Fi, kitchen access, pet-friendliness, etc by using keywords.
* Cleanliness: Users could select a desired level of cleanliness and listings that have been commented as ‘clean’, for example, would be shown.

Data Output:

* Search Results: The system will output a list of short-term rental listings that match the user's input criteria.
* Pricing Information: The system will display the pricing details for each listing.
* Suburb Property Report: Users can generate a comprehensive report detailing all available property listings within a specific suburb for the selected time period. This report will include key information about each listing, facilitating an overview of accommodation options in a particular area.
* Price Distribution Chart: A graphical chart will illustrate the distribution of property prices across the selected criteria. This chart will provide users with insights into the range of pricing options available, aiding in making informed decisions based on their budget considerations. Chart showing the distribution of property prices

## System Overview

The system will serve as a comprehensive and user-friendly platform for travellers and backpackers worldwide who are planning to visit Sydney, Australia to find short-term accommodation. Its primary functionalities include:

* Accommodation Search: Users can input their location preferences, travel dates, budget constraints, desired amenities, and cleanliness preferences to search for short-term rental listings in Sydney that match their criteria.
* Listing Details: The system will provide detailed information about each listing, including amenities, and pricing details. Users can explore these details to make informed decisions about their accommodation.
* Suburb Property Report: Users have the option to generate a comprehensive report that compiles information about all available property listings within a specific suburb for their selected time period. This report facilitates an overview of accommodation options in a particular area.
* Price Distribution Chart: The system will generate a graphical chart illustrating the distribution of property prices across the selected criteria. This chart provides users with insights into the range of pricing options available, helping them to make informed decisions based on their budget considerations.

In conclusion, the system will simplify the process of finding and booking short-term rentals in Sydney by offering a range of search options, detailed listing information, suburb-specific reports, and visual representations of price distributions. It aims to enhance the user experience for travellers and backpackers seeking accommodation in Sydney, Australia.

## Potential Benefits

There are many potential benefits of this system and extend to both travellers but also the broader tourism industry. These benefits include:

* Efficient accommodation selection: Travelers will benefit from a streamlined and efficient process for selecting accommodation that precisely aligns with their preferences and budget. This will save them valuable time and ensure a more enjoyable and comfortable stay in Sydney.
* Informed Decision-Making: The system empowers users with comprehensive information about each listing, including amenities, pricing, cleanliness, and user reviews. This enables travellers to make well-informed decisions, increasing overall satisfaction with their chosen accommodations.
* Market Insights: The price distribution chart provides valuable insights into the pricing dynamics of short-term rentals in Sydney. This can benefit both travellers and property owners by helping them gauge market trends and competitive pricing strategies.
* Promotion of Local Tourism: By facilitating easier and more convenient accommodation booking, the system contributes to the promotion of tourism in Sydney. This can boost the local economy, supporting businesses and job creation within the tourism sector.
* Enhanced User Experience: The user-friendly interface and interactive features make planning a trip to Sydney more enjoyable and less stressful. This, in turn, encourages travellers to explore the city and its attractions with greater confidence.
* Data-Driven Decision-Making: The suburb property reports can be invaluable for travellers planning to stay in specific areas, providing a wealth of data to aid in choosing the right neighbourhood. This data-driven approach ensures travellers have a deeper understanding of the local rental market.

In summary, the system has the potential to revolutionise the way travellers plan and book accommodation in Sydney, offering convenience, information, and insights that enhance the overall travel experience while contributing to the vitality of the local tourism sector.

# Requirements

## User Requirements

Our user would be backpackers and travellers who use Airbnb to choose their accommodation.

* As a user, I want to be able to find all the Airbnb in selected suburb and period.
* As a user, I want to be able to see the distribution of pricing in selected period.
* As a user, I want to be able to search a keyword and get all information about the keyword.
* As a user, I want to be able to view the results of how many customers commented on factors related to cleanliness.
* As a user, I want to be able to select a price range such as minimum price and maximum price and get a listing of all Airbnb.

## Software Requirements

**Functional Requirements:**

R1.1: The software shall allow users to specify a date range and a suburb for analysis.

R1.2: The software shall allow users to generate a report that provides information about all property listings within the specified suburb for the selected period.

R1.3: The software shall create a chart showing the distribution of property prices for the selected suburb and date range.

R1.4: The software shall provide the option to search for and retrieve all records within the specified period that contain specific keywords, such as "pool" or "pet."

R1.5: The software shall perform analysis on customer comments related to factors associated with cleanliness, identifying keywords like "clean," "hygiene," "tidy," etc.

R1.6: The software shall present with a summary of how many customers have commented positively or negatively on cleanliness-related factors.

R1.7: The software shall allow users to input their budget constraints to filter listings that fall within their financial considerations.

## Use Cases & Use Case Diagrams

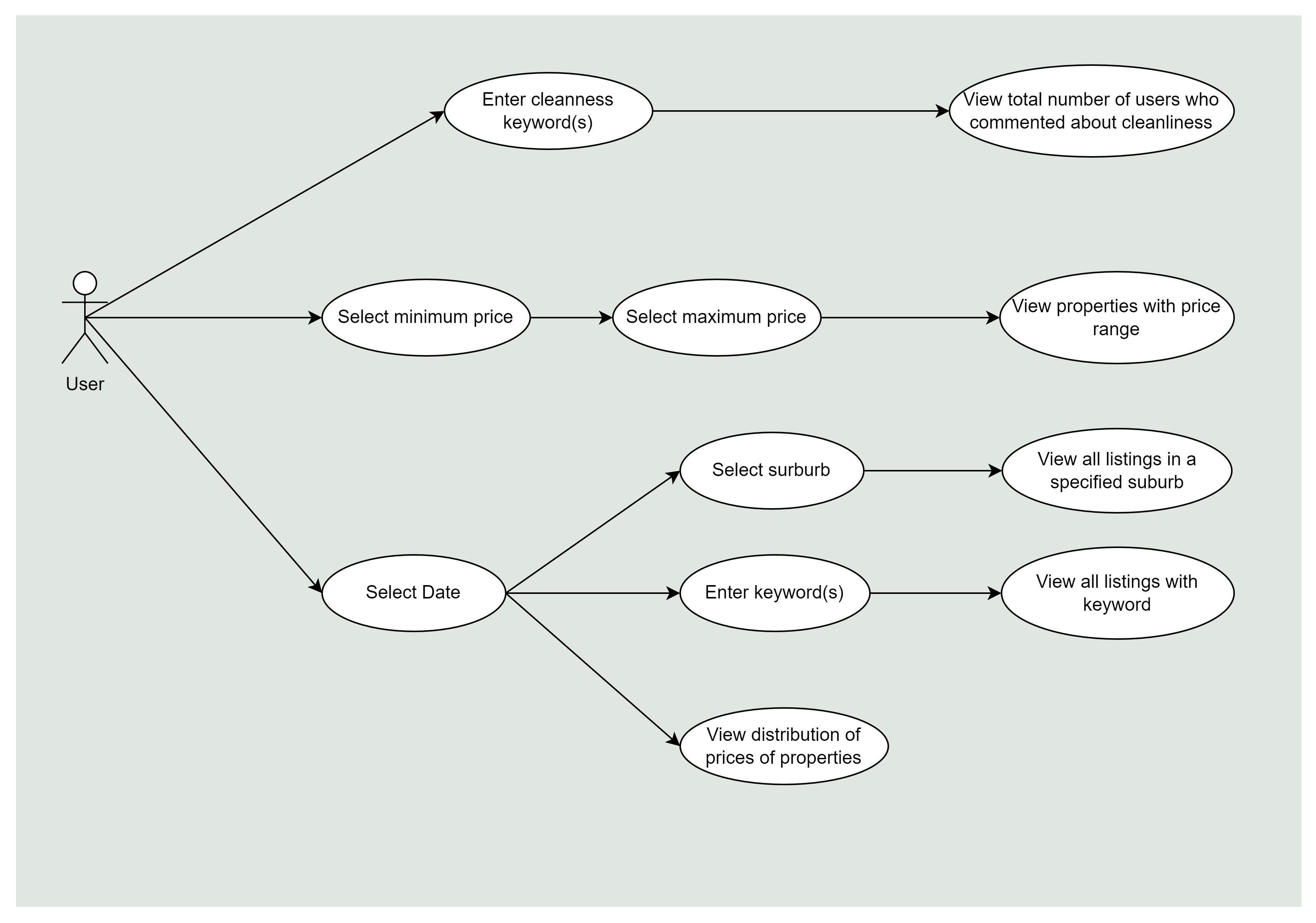
|  |  |
| --- | --- |
| **Use Case ID** | 1 |
| **Use Case Name** | View distribution of prices of properties |
| **Actors** | User |
| **Description** | The user will be able to see a chart with the distribution of prices of properties. |
| **Flow of Events** | 1. The user selects date 2. The system will display the chart with the distribution of prices of properties. |

|  |  |
| --- | --- |
| **Use Case ID** | 2 |
| **Use Case Name** | View total number of users who commented about cleanliness |
| **Actors** | User |
| **Description** | The user will be able to see the total number of properties with the comments related to cleanliness. |
| **Flow of Events** | 1. The user enters a keyword related to cleanliness e.g. hygienic, clean, tidy 2. The system will display the total number of users who made comments on cleanliness factor. |

|  |  |
| --- | --- |
| **Use Case ID** | 3 |
| **Use Case Name** | View properties with selected price range |
| **Actors** | User |
| **Description** | The user will be able to see all the properties with the desired price range. |
| **Flow of Events** | 1. The user selects minimum price of the property/day 2. The user selects maximum price of the property/day 3. The system will display all the properties with the desired price range. |

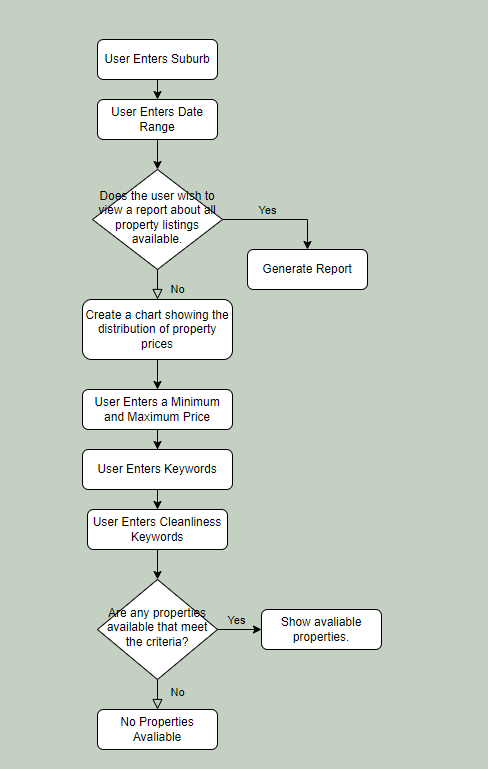
|  |  |
| --- | --- |
| **Use Case ID** | 4 |
| **Use Case Name** | View properties with use-entered keyword |
| **Actors** | User |
| **Description** | The user will be able to see all the properties with the entered keyword and date. |
| **Flow of Events** | 1. The user selects date 2. The user enters keywords e.g. pool, pet 3. The system will display all the properties with user-entered keyword for the selected date |

|  |  |
| --- | --- |
| **Use Case ID** | 5 |
| **Use Case Name** | View properties with specified suburb |
| **Actors** | User |
| **Description** | The user will be able to see all the properties with the selected suburb and date. |
| **Flow of Events** | 1. The user selects desired date 2. The user selects suburb in Sydney 3. The system will display all the properties with date and selected suburb. |



# Software Design and System Components

## Software Design



## System Components

### Functions

**Function 1) Selecting Suburb**

User can select suburbs where they want to book an Airbnb from the provided list of suburbs. Data type is “string”, and the data is used for the lists of suburbs.

**Input parameters:** addProperty(Suburbs), getProperty

**Return value:** The Airbnb’s shown to the user will be in the suburb they selected.

**Function 2) Selecting Minimum Price and Maximum Price**

User can select their own price range due to their own financial status to choose Airbnb. Data type is “integer” and used for the price range.

**Input parameters:** addPrice()

**Return value:** The Airbnb's that shown to the user would be within the specified price range.

**Function 3) Selecting Date**

User can pick their Check-in Date and Check-Out Date for their trip. Data type is “integer” and used for date and month.

**Input parameters:** addDateData(), getDateData()

**Return value:**  The Airbnb's that shown to the user would be within the specified time period.

**Function 4) Searching keyword**

User can search any keywords and get results from the keyword they searched. Data type is “string” and used for user to search for any keywords.

**Input parameters:**

incrementKeywordCount(keyword)

if the keyword exist in dic

Increment the associated value by 1

If no, add the keyword as a key and set the value to 1

getKeywordCount(keyword)

If yes return to associated value

If no, return 0

**Return Value:** The Airbnb's shown to the user will include the keyword(s) they entered.

**Function 5) Cleanliness-related keywords**

User can search a keyword such as “clean”, “tidy”, “hygienic” and get results of comments that are mentioning cleanliness-related keywords. Data type is “string” which user can search for the cleanliness-related keywords.

**Input parameters:**

incrementKeywordCount(keyword)

if the keyword exist in dic

Increment the associated value by 1

If no, add the keyword as a key and set the value to 1

GetKeywordCount(keyword)

If yes return to associated value

If no, return 0

**Return Value:** The Airbnb's shown to the user will include comments from past visitors which contain the relevant cleanliness keyword(s) the user entered.

### Data Structures / Data Sources

**1.**

**Type:** dict

**Description:** This data structure employs a dictionary where each key represents a combination of suburb name and selected date. The associated value is a list containing multiple property dictionaries, each containing attributes like id, name, price, and so on.

**Usage:** This structure facilitates the reporting of property details for a specified suburb and date.

**Data Members:**

**Key:** Combination of suburb name and selected date.

**Value:** List of property dictionaries containing attributes like id, name, price, etc.

**Functions:** addProperty(suburb, selected\_date, property\_dict): Adds a property dictionary to the list for the specified suburb and date.

getProperties(suburb, selected\_date): Retrieves the list of property dictionaries for the given suburb and date.

**2.**

**Type:** dict and list

**Description:** query the listing, information corresponding to a certain date and user entered keyword. The data structure is a dictionary with key=date, value=list, and multiple data are stored in the list, indicating multiple property information.

**Usage:** This structure stores property listing information based on date and keywords entered by the user.

**Data members:** date - report the information of all listings in a specified suburb

Data - retrieve all records that contain a keyword, e.g. pool, pet.

**List of functions that use it:** addDateData(), getDateData()

**3.**

**Type:** dict

**Description:** The hash table is used to count and analyse customer comments related to cleanliness. It efficiently stores keywords and their corresponding counts.

**Usage:** This structure facilitates analysing how many customers commented on cleanliness-related factors.

**Data Members:**

Keys: Cleanliness-related keywords (e.g., "clean", "tidy", "hygienic" etc).

Values: Count of comments mentioning the cleanliness-related keywords.

**Functions:**

incrementKeywordCount(keyword): Increments the count for a specific cleanliness keyword.

getKeywordCount(keyword): Retrieves the count for a specific cleanliness keyword.

**4.**

**Type:** Array

**Description:** This array is used to contain all prices of Airbnb listings.

**Usage:** This structure holds the prices so when the user selects minimum and maximum, they are shown Airbnb listings within the specified price range.

**Data Members:**

List of Prices: This is the primary data member, which is an array or list that holds the prices of Airbnb listings: Each element in the array represents the price of a single Airbnb listing.

**Functions:**

addPrice(price): This function allows you to add a new price to the array. You provide the price as a parameter, and it gets appended to the end of the array.

getPricesInRange(min\_price, max\_price): This function retrieved all the prices within a specified range. You pass in the minimum and maximum prices as parameters, and it returns a list of prices that fall within that range.

sortPrices(): This function sorts the prices in the array in ascending or descending order. You can specify the sorting order as a parameter.

getMinimumPrice(): This function returns the minimum price in the array.

getMaximumPrice(): This function returns the maximum price in the array.

removePrice(Price): This function allows you to remove a specific price from the array. You provide the price you want to remove as the parameter.

getAveragePrice(): This function calculates and returns the average (mean) price of all Airbnb listings in the array.

getTotalCount(): This function returns the total number of prices (the length of the array).

### Detailed Design

class Data(object):

def\_\_init\_\_(self):

self.initial ={“id”: None,

“name”: None,

“host\_id”: None,

“host\_name”: None,

“neighbourhood”: None,

“latitude”: None,

“longitude”: None,

“room\_type”: None,

“price”: None,

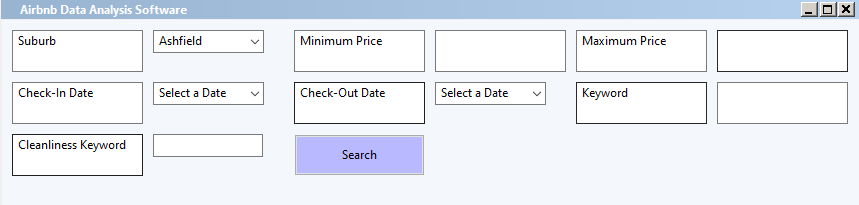
“available”: None}

self.data = self.initial

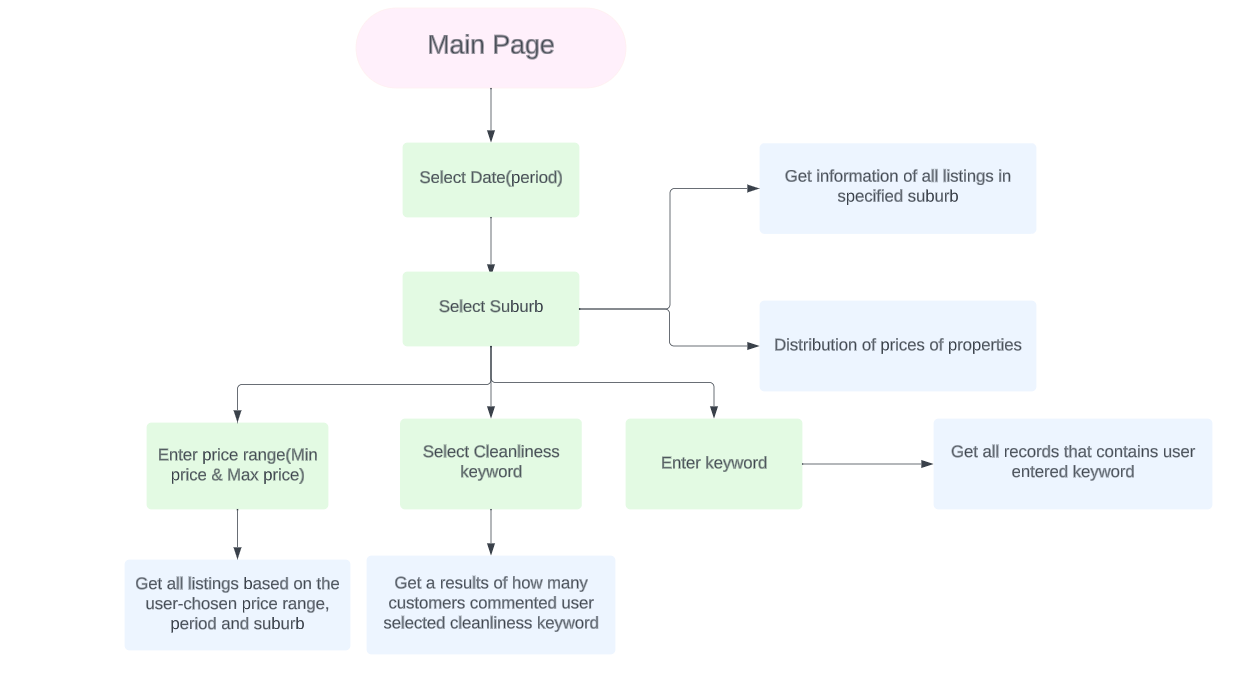
def getData(self):

return self.data

# User Interface Design

We used wxpython and figma to design our wireframe. We used wxpython for the wireframe because our software will be made from wxpython so we can get the exact design of the menu bar from it. Also, figma was used for the future database section which will have a chart of Airbnb data under the menu bar. Every time user search for their preferences in the menu bar, the results will be shown in the data chart below.

## Structural Design



To get information of all listings in user-selected suburb, user can first select a period for their future trip and then select a specific suburb. For the distribution of prices of Airbnb properties user can select a period then select a suburb. Users also can enter a keyword to get all records that contains user entered keyword. They can get results of how many customers commented the user-selected cleanliness keyword and they can also set a range of price by entering minimum and maximum price to get all listings based on the chosen price range. To get all the information, user must choose the period and a suburb. We made these design because it is user friendly, and user can easily get access to the data they want.

## Visual Design

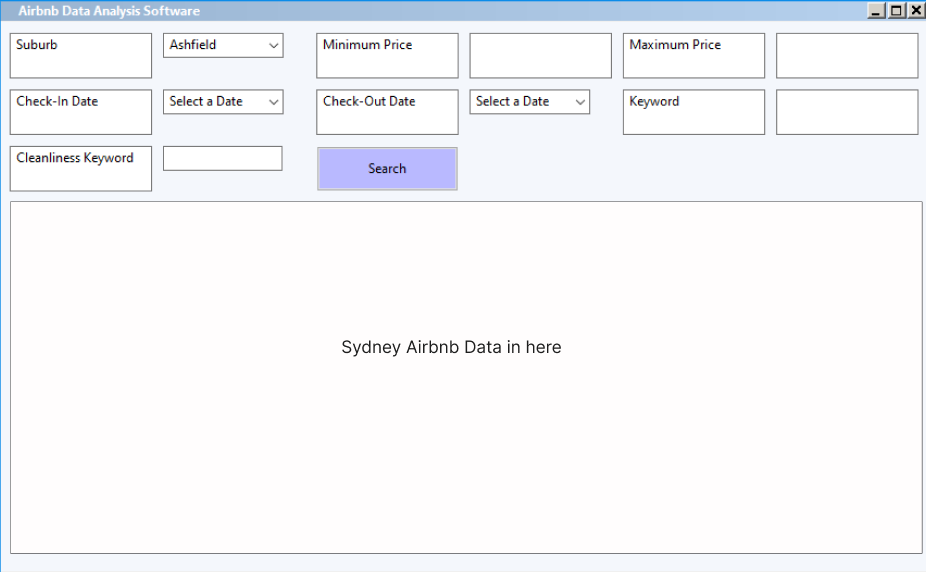


Image above is the design of the main page. Background colour will be light blue (191,205,219), and there are 7 selection types that user can choose from (suburb, minimum price, maximum price, check-in date, check-out date, search keyword, and search cleanness comments).

Using light blue as a background will help user to see selection menu bars more easily than the white background. Bottom part will have a chart of Sydney Airbnb data. Also, having a white text box for all the menu and data chart help user to easily visualise the whole menu and the data. We used “wxComboBox” for searching suburb, select a date and searching for cleanliness keyword. For minimum price, maximum price and keyword, we used “wxTextCtrl” to allow users to enter manually. We used “wxButton” for the search so user can search their data through the button to get the data of their inputs. For “Search” button, we used the button colour as RGB: 185,185,255 because it needs to be more visualise than the other search engines so user can easily find it.