

Sydney AirBnB Search

Software Design Document



September 4, 2023

Griffith University

Yiling Shou s5255523 Victor Mou s5269386 Luke Brady s5293182

Table of Contents

[1.0 System Vision 2](#_Toc144751343)

[1.1 Problem Background 2](#_Toc144751344)

[1.2 System Overview 2](#_Toc144751345)

[1.3 Potential Benefits 2](#_Toc144751346)

[2.0 Requirements 3](#_Toc144751347)

[2.1 User Requirements 3](#_Toc144751348)

[2.2 Software Requirements 3](#_Toc144751349)

[2.3 Use Cases & Use Case Diagrams 4](#_Toc144751350)

[3.0 Software Design and System Components 5](#_Toc144751351)

[3.1 Software Design 5](#_Toc144751352)

[3.2 System Components 5](#_Toc144751353)

[3.2.1 Functions 5](#_Toc144751354)

[Search: 5](#_Toc144751355)

[Listing: 6](#_Toc144751356)

[Review: 6](#_Toc144751357)

[Suburb: 6](#_Toc144751358)

[3.2.2 Data Structures / Data Sources 6](#_Toc144751359)

[3.2.3 Detailed Design 6](#_Toc144751360)

[4.0 User Interface Design 7](#_Toc144751361)

[4.1 Structure Design 7](#_Toc144751362)

[4.2 Visual Design 8](#_Toc144751363)

# System Vision

## Problem Background

Sydney’s Airbnb listings are so numerous that it is difficult to find any individual listing in the dataset. Software to categorise and refine a user’s results will greatly improve customer experience, as they will only see results that are relevant to them. They will be able to select a specific suburb, price range, features, room type and cleanliness, then decide where they will stay from the results that remain.

## System Overview

The system will be able to take a user's search input and output the listings that match this input. A user will be able to enter one or many suburbs, and only receive results for listings within the specified suburb or suburbs. If no suburb is entered, all suburbs will be returned.

A user can also input a price range by specifying maximum and minimum values. The search will return all listings within the specified price range. If one or both of the parameters are not entered, then there will be no limit placed upon the price of listings for any parameter that is missing.

A user will be able to refine the results by searching for words such as 'clean' in the user comments. If a reviewer has mentioned the word clean in their review, the review will appear. The user will then be able to select each review to search the reviews more closely. This will allow them to eliminate results with a comment containing 'not clean', or something similar.

A user can specify from a drop-down menu whether they wish to have a listing which is a whole house/appartment, a private room or other. If no response is selected, then all results will be returned. Otherwise, the listings will be refined to only listings that meet the room type requirement.

Finally, a user can specify keywords in a 'description' search box. If their keywords appear in the summary, space, description, experiences\_offered or neighbourhood\_overview categories for that listing, the listing will be displayed. Otherwise, it will not.

## Potential Benefits

The software is being created to gain insights into Airbnb listings in Sydney through reporting information related to each listing, as well as an overview of all listings.

One potential benefit of the software is users being more aware of fair pricing of listings in their area, so they do not overpay for their accommodation. This will encourage hosts to price their listings more competitively, so that they receive guests.

Through this software, users will be able to compare listings with each other and will be able to make more informed decisions about their accommodation selection. They will be able to search for listings in the suburb they want, with added specifications, including the number of bedrooms and bathrooms, and then choose a listing from the results that are returned.

# Requirements

## User Requirements

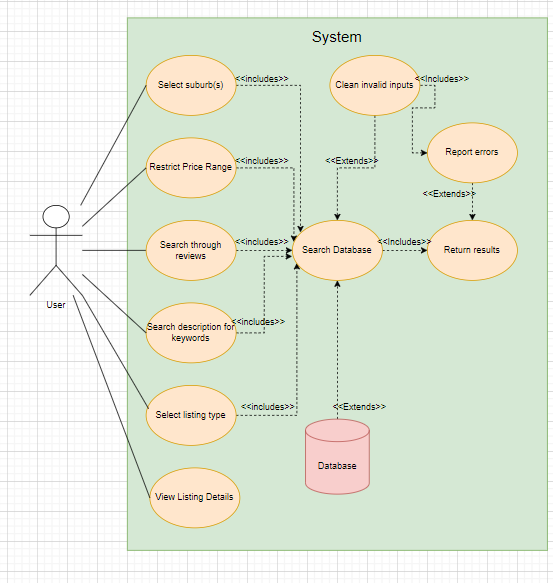
* User can select suburb.
* User can restrict price range.
* User can search through reviews.
* User can search for places with specific key words.
* User can view the details of a listing.

Each of the input fields in the system are able to take valid user inputs and are able to check for validation. The suburb field only takes inputs that match a suburb name. If an invalid suburb is entered, it is ignored. For the price range, only integer or floating-point values are accepted. Other inputs are ignored. If the minimum value equals or exceeds the maximum value, both values are to be ignored. If the minimum value exceeds the highest price or the maximum value is less than the lowest price, they will be ignored. When a user searches for the descriptions or reviews, if their keywords do not match any results, that keyword is to be ignored. If multiple keywords are searched for, the system will search the database for each word individually. The property type field takes values from a drop-down menu and can thus only take three different values: whole house/appt, private room, or other. If no listings pass the validation, then an error will be displayed, asking the user to broaden their search. If any of the inputs were ignored, the system will display the results that were ignored.

## Software Requirements

* The system can search the database.
* The system can report errors.
* The system can clean invalid user input.
* the system can return results.

## Use Cases & Use Case Diagrams



# Software Design and System Components

## Software Design

A screen shot of a computer

Description automatically generatedA screenshot of a graph

Description automatically generated

## System Components

### Functions

## Search:

* loadData() – loads the data from the database and selects all entries that match any specifications provided.

## Listing:

* getPrice() – returns the price of a listing from the results uploaded from the database.
* getDescription() – returns the description of a listing from the results uploaded from the database.
* getListingType() – returns the listing type of a listing from the results uploaded from the database.
* displayData() – displays all data collected for user viewing.

## Review:

* getReview() – returns an instance of a review from a listing from the results uploaded from the database.

## Suburb:

* getSuburb() – returns the specific suburb that matches a listing.
* getDescription() – returns the suburb description from the database.

### Data Structures / Data Sources

To complete the requirements of the software, the functions and classes of the system must be clearly displayed. Each listing is an instance of the listing class, and each contains an ID value, a price, a description and a listing type. They also have zero to many reviews, each of which has a text and a date. Each review only belongs to one listing, making the relationship between listings and reviews a one-to-many relationship. Each suburb has a name and description and can contain zero to many listings. Each listing belongs to only one suburb, making the relationship between listings and suburbs another one-to-many relationship. Hosts have an ID, a first name and a last name, and can create, update or delete listings for their properties. A host may have one or more listings, but each listing belongs to one host or host family. This represents a one-to-many relationship between hosts and listings. Guests have a first name and a last name and can place reviews or search the database. Each guest can place many reviews or searches, but each individual search or review is made by a singular guest. The search class extends the listing class, as the search returns only the listings that match the search parameters.

### Detailed Design

**loadData():**

Connect to database.

Check each line of the database for entries that match user specifications.

return these lines only.

**displayData(lines):**

Takes an array of listings as a parameter.

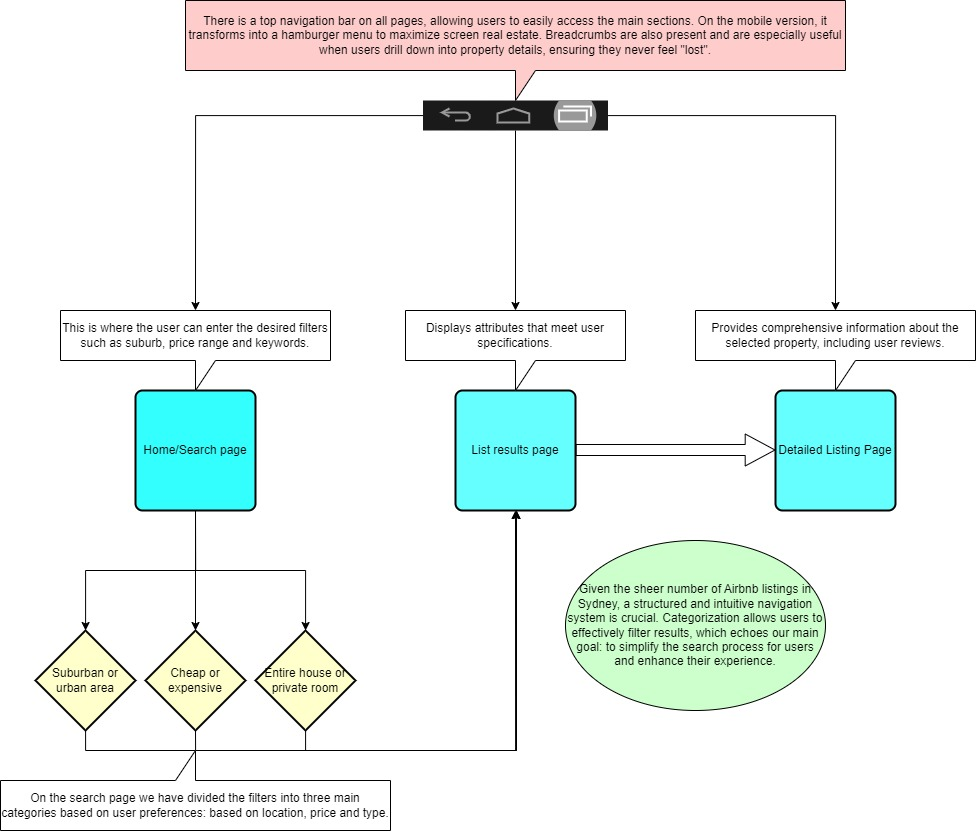
Loops through each element of the array.

For each element, display it on the screen.

# User Interface Design

Our software interface was designed to simplify searching and categorizing Sydney Airbnb listings, and we used Figma, a collaborative interface design tool. This decision was driven by Figma’s real-time collaboration capabilities, which allow our team to work together seamlessly and with versatility during the structural and visual design stages. During our design process, group members shared their own experiences using Airbnb. After discussions with each other and combined with feedback from users who have used Airbnb online, we found that users tend to prioritize intuitive navigation, simple design, and informative feedback. So, when designing user interfaces, we try to avoid presenting too much information or complex workflows at once, where users may find the interface overwhelming. We will use a simpler design, such as using clearer icons, consistent style, colour, and font to avoid users finding some icons or buttons confusing.

## Structure Design



## Visual Design

