

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

The Sydney Airbnb rental market is a vibrant and varied environment with a wide selection of homes for short-term stays. Users looking to rent or invest in real estate may find it difficult to sort through the massive amount of data and make wise choices. Making decisions in this market can be strongly influenced by an understanding of price trends, property attributes, and customer feedback.

A big issue potential renters or investors may face with is Information Overload. Numerous characteristics are included in this information, including location, property type, amenities, and client feedback. Finding the best solutions after navigating through this information becomes a difficult undertaking.

In addition to Information Overload, the complexity of pricing in the Sydney Airbnb market adds to the decision-making challenge. The property's location in relation to important events and attractions as well as the provision of special amenities might have an impact on rental rates. Deep data research is necessary to understand pricing trends and determine the best times of year to book or list properties. Users could find it difficult to determine fair price without a thorough tool to analyse pricing distributions across time periods, which could result in missed opportunities or unsatisfactory rental plans. Dealing with this pricing complexity is therefore essential for the everyday customer.

## System Overview

We Propose a simple software system that will assist customers to narrow down their searches and requirements for Airbnb's, specifically in Sydney. Our system intends to simplify the decision-making process for users looking for short-term accommodations in the dynamic Sydney rental market by utilising the power of data analysis and visualisation. The system will behave as a simple program that analyses datasets from Sydney’s Airbnb environment and allow users to interact and extract specifications they require, of said datasets.

The system's fundamental feature allows users to choose specific suburbs, time periods, and keywords to analyse and visualise pertinent data. The software processes data on property listings, pricing trends, and customer reviews by interacting with the sizable Sydney Airbnb dataset. A graphical user interface will make it simple for consumers to obtain key information that influences their renting decisions.

## Potential Benefits

Potential benefits include a customer's Insight into Price trends, Transparent Property Attributes and Time Management.

Insight into Price Trends.

One of the key advantages of our proposed software system lies in its ability to provide users with insights into price trends within the Sydney Airbnb rental market. By allowing users to input specific time periods for analysis, the software generates comprehensive visualizations that outline how rental prices fluctuate over different seasons and events. This feature assists potential renters and property investors alike in making well-informed decisions about the optimal times to book or list properties. The clear visualization of price distributions empowers users to grasp the range of prices they can expect to encounter, thus preventing underestimation or overpayment. With these insights, users can strategically plan their rental experiences, avoiding the rush during peak seasons and seizing opportunities when prices are most favourable.

Transparent Property Attributes.

Another key benefit of our system is the transparency of property attributes. Having access to clear and thorough information about property features and services becomes crucial as customers browse through the wide variety of short-term accommodations available on the Sydney Airbnb market. By allowing customers to enter criteria, such as desired suburbs and keywords, to filter listings depending on their preferences, our programme streamlines the process. With the use of this tool, consumers can save time and effort by only seeing listings that match their specific needs. The system enables visitors to choose the accommodations that best meet their needs by showing property qualities including location, property type, and amenities that are offered.

Time Optimization.

A key benefit for users navigating the Sydney Airbnb market is time management. Making informed decisions takes much less time when a plethora of property-related data can be accessed and analysed quickly. Users may rapidly select listings based on their preferred suburbs, discover transparent property attributes, and gain insights on price trends instead of wading through mountains of information. This simplified procedure not only improves user experience but also makes it possible for users (such as investors) to quickly take advantage of rental opportunities. Users can concentrate their efforts on analysing qualities that most closely match their tastes and goals by cutting down on the time spent on data analysis.

# Requirements

## User Requirements

The expected user of the system will be an average traveller who makes their way into Sydney for a holiday stay.

The User must be able to do the following in order to fully utilize the program:

* Launch the Program.

This includes having up to date computers (phones, laptops, etc...) that can run modern software.

* Enter their preferences.
* Customize their search.
* Navigate results.
* Exit the program.

## Software Requirements

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.

R1.3 The program shall provide a user-friendly interface for interactions.

R1.4 The program shall offer users their preferred suburb for accommodation searches.

R1.5 The program shall enable users to specify the time of their stay using start and end dates.

R1.6 The program shall offer a visualization feature to display the distribution of rental prices within user-defined time periods.

R1.7 The program shall allow users to conduct keyword-based searches by inputting amenities or preferences such as "pool" or "pet."

R1.8 The program shall process user inputs to filter property listings based on selected suburbs, time periods, and keywords.

R1.9 The program shall provide transparent property attributes including location, property type, amenities, and customer feedback.

R1.10 The program shall perform cleanliness analysis by identifying keywords related to cleanliness in customer reviews.

R1.11 The program shall display clear and understandable visualizations/listings of price distributions and cleanliness insights.

R1.12 The program shall be able to analyse the datasets and differentiate the property types respectfully and be able to generate listings of specific types respective of the user’s preference.

R1.13 The program shall be designed to handle various edge cases, such as invalid inputs and missing data, to ensure robustness.

R1.14 The program shall provide an option for users to exit the software when their interactions are complete.

## Use Cases & Use Case Diagrams

Below are 5 Use Cases and their respective diagrams for the use cases when using the system;

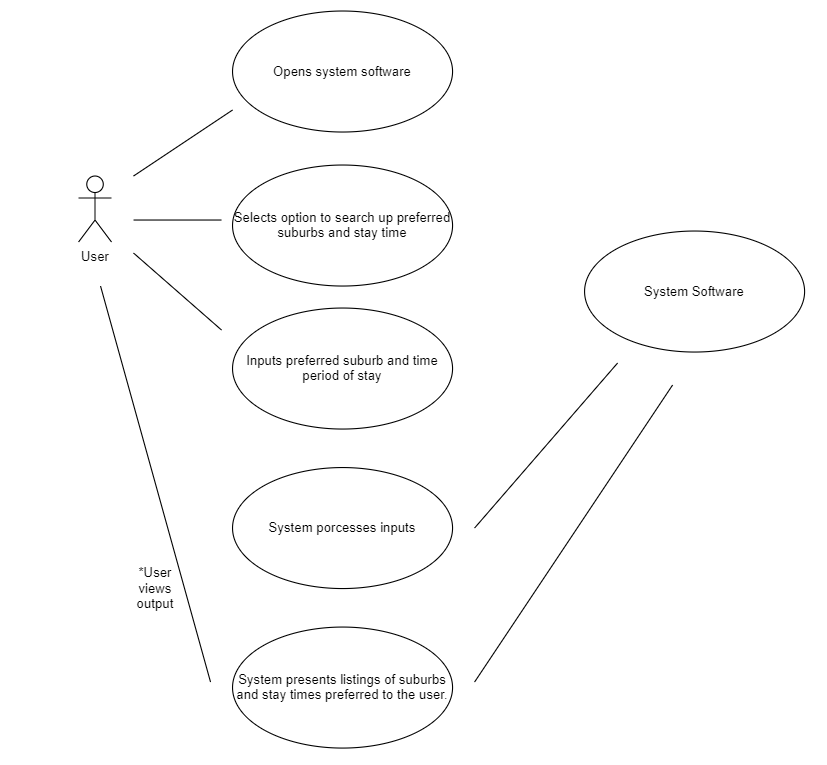
* Search by Suburb and Time Period
* Visualization of Price Distribution
* Searching by Keyword
* Analysing Cleanliness by Comments
* Property Type Preference

To note it will a paramount functional requirement that the System provide a simple, unclustered User Interface.

Use Case 1. User searching Suburb and Time Period on Software System

|  |  |
| --- | --- |
| **Use Case ID** | 1 |
| **Use Case Name** | Search by Suburb and Time Period |
| **Primary Actors** | User |
| **Description** | The software is used by the User to narrow down Airbnb accommodation options in Sydney specific to the User’s requirements of suburb and specific time frame. |
| **Pre-Conditions** | The user has launched the Airbnb data analysis and visualization program and is presented with the user interface. |
| **Flow of Events** | 1. User opens the software 2. User selects Option to search by suburb and time period. 3. User inputs suburb and stay dates. 4. User submits Input. 5. System processes input and retrieves listings. 6. System displays listing details to the user. |
| **Post-Conditions** | User receives an organised list of available accommodations in the chosen suburb for the specified time period. |
| **Alternate Flow** | Invalid inputs – System prompts user to re-enter information.  No listings found – System informs user and suggests adjusting parameters. |

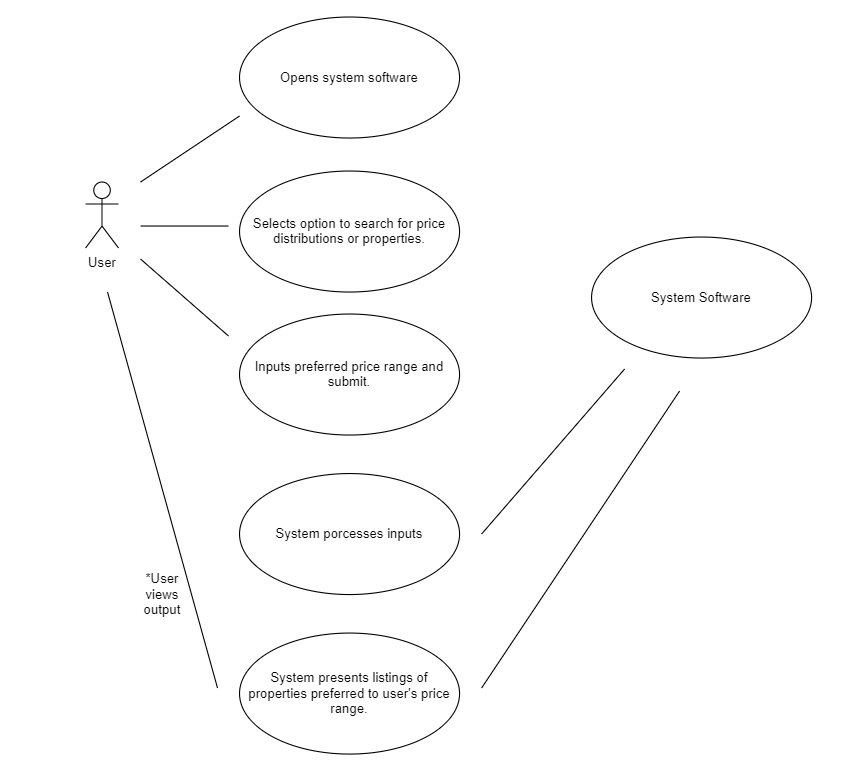
Use Case Diagram 1. Use Case ID 1.



Use Case 2. User Visualising Price Distribution on System Software

|  |  |
| --- | --- |
| **Use Case ID** | 2 |
| **Use Case Name** | Visualise Price Distribution |
| **Primary Actors** | User |
| **Description** | A user interacts with the system to produce a visual depiction of the distribution of Airbnb properties in Sydney. |
| **Pre-Conditions** | The user has launched the Airbnb data analysis and visualization program and is presented with the user interface. |
| **Flow of Events** | 1. User opens the software. 2. The user selects the option to search for price distributions of the properties. 3. User inputs their preferred price ranges. 4. User submits input. 5. System processes the user’s input and scans datasets accordingly against prices of properties. 6. The system generates and produces the user with a visualisation/listing of properties preferred to the user’s price ranges. |
| **Post-Conditions** | The user is provided with a visualisation that displays the distribution of property prices. |
| **Alternate Flow** | If there is an error during the processing of the user's input:  The system displays an error message and guides the user to retry generating the price distribution. |

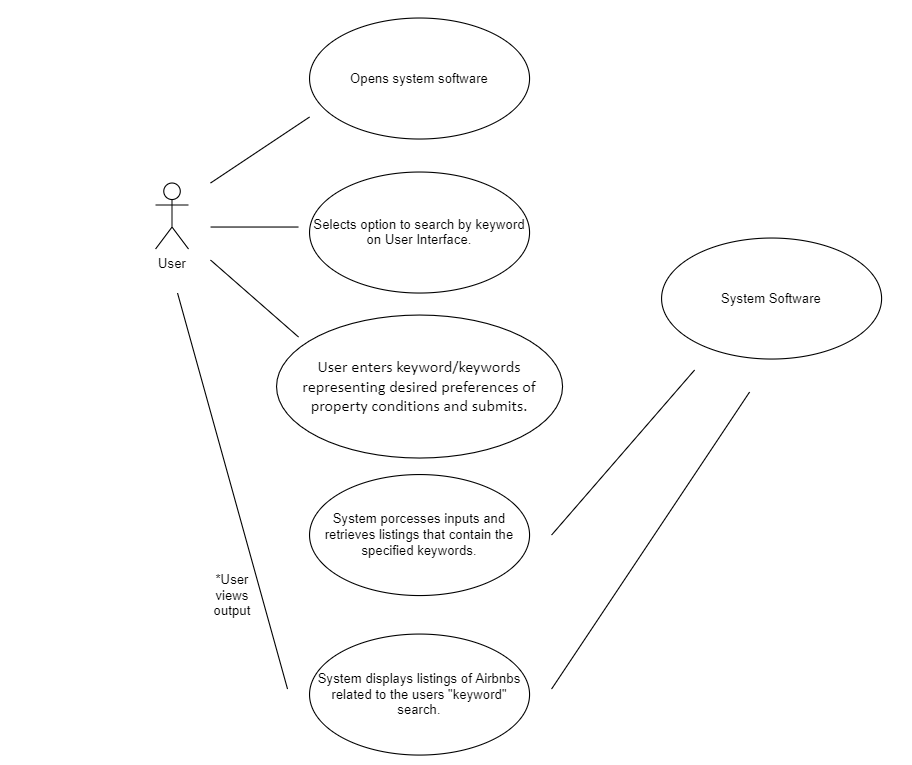
Use Case Diagram 2. Use Case ID 2.



Use Case 3. User searching for Airbnb's by keyword.

|  |  |
| --- | --- |
| **Use Case ID** | 3 |
| **Use Case Name** | Search by Keyword |
| **Primary Actors** | User |
| **Description** | The user interacts with the system to look up specific Airbnb listings using keywords they have entered. By using keywords like "pool," "pet," and other suitable terms, the user hopes to filter and find Airbnb descriptions or even ads linked to the Airbnb properties. |
| **Pre-Conditions** | The user has launched the Airbnb data analysis and visualization program and is presented with the user interface. |
| **Flow of Events** | 1. User opens Airbnb dataset system software. 2. User selects option to search by keyword on the UI. 3. User enters keyword/keywords representing desired preferences of property conditions. 4. User submits input. 5. The system processes the user's input and retrieves listings that contain the specified keywords. 6. The system displays listings of Airbnb’s that have descriptions/ads that contain terms related to the user’s search. |
| **Post-Conditions** | The user is presented with a list of Airbnb listings that contain the specified keywords, helping them find accommodations that cater to their preferences. |
| **Alternate Flow** | If no listings match the entered keywords:  The system informs the user that no listings match the criteria and suggests refining the search. |

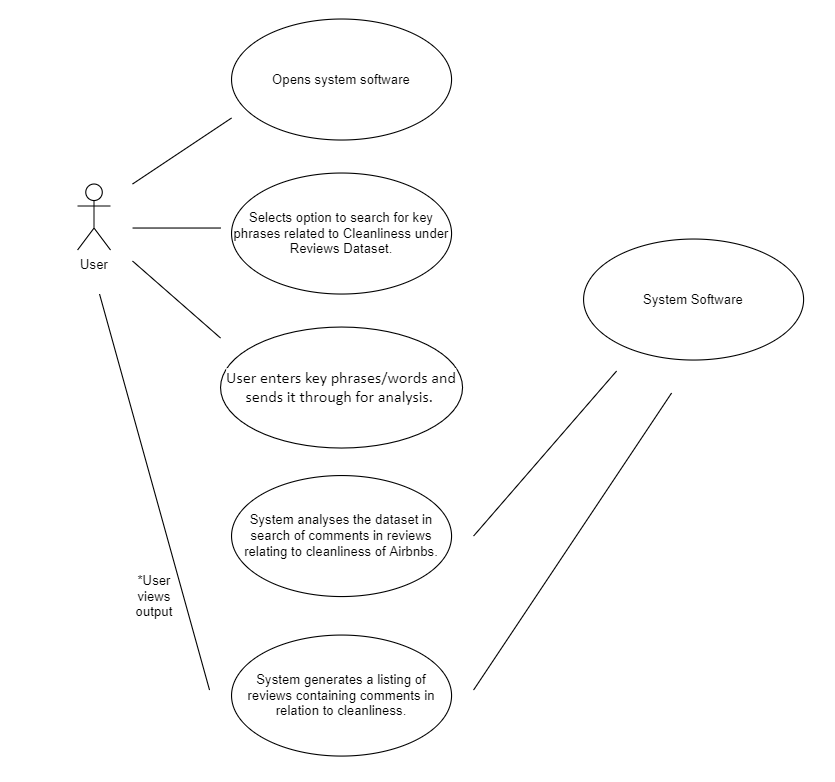
Use Case Diagram 3. Use Case ID 3.



Use Case 4. Analyse Cleanliness by Comments

|  |  |
| --- | --- |
| **Use Case ID** | 4 |
| **Use Case Name** | Analysing Cleanliness by searching Comments |
| **Primary Actors** | User |
| **Description** | In this use case, the system examines user reviews of Airbnb rentals to assess the standard of cleanliness and pinpoint elements that affect hygiene. The user searches in the system for reviews that highlight cleanliness aspects of Airbnb’s in Sydney. The user does this by inputting keywords or rather synonyms of words and phrases related to “clean”, such words as “tidy” and “spotless”. |
| **Pre-Conditions** | The system has access to customer reviews and associated data for Airbnb listings. |
| **Flow of Events** | 1. User opens System Software. 2. Under the “Reviews” selection of datasets, the user inputs key phrases or words in relation to cleanliness. 3. Once the input is sent through, the system analyses the dataset in search of comments related to the search. 4. The system generates a listing of comments and reviews in relation to cleanliness comments/words. |
| **Post-Conditions** | The system provides a report showcasing the number of cleanliness-related comments for each Airbnb listing. |
| **Alternate Flow** | If no cleanliness-related keywords are found in the reviews:  The system may display a message indicating that there are no comments specifically related to cleanliness. |

Use Case Diagram 4. Use Case ID 4.

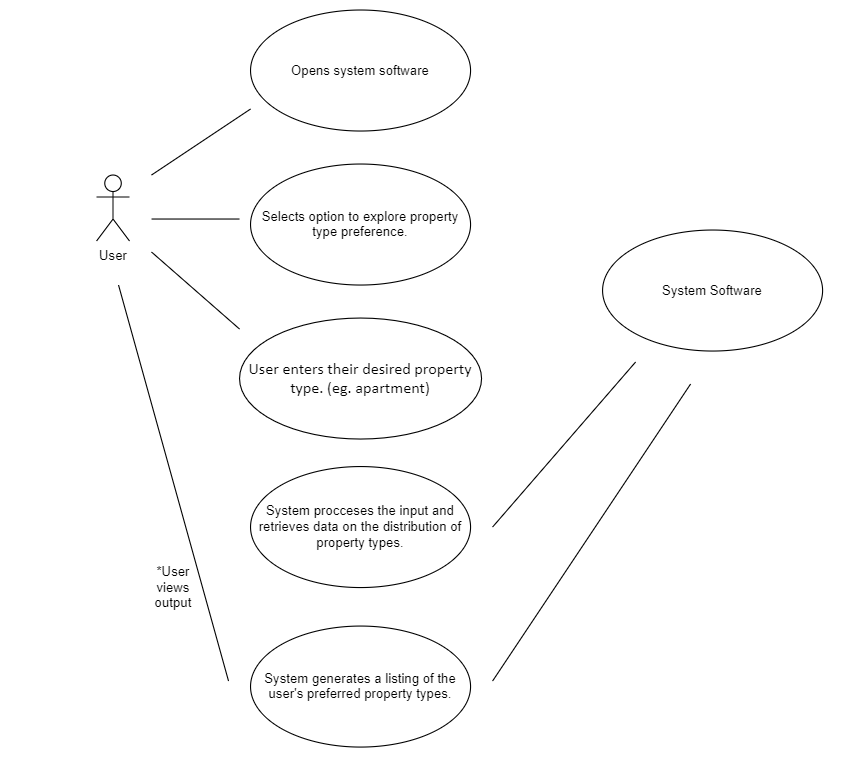


Use Case 5. Property Type Preference.

|  |  |
| --- | --- |
| **Use Case ID** | 5 |
| **Use Case Name** | Property Type Preference. |
| **Primary Actors** | User |
| **Description** | The user searches for their preferred Airbnb accommodation type. The user aims to understand the distribution and availability of various property types, such as apartments, houses, or shared rooms. |
| **Pre-Conditions** | The user has launched the Airbnb data analysis and visualization program and is presented with the user interface. |
| **Flow of Events** | 1. The user selects the option to explore property type preference. 2. The user inputs their preferred accommodation type (house, apartments, etc...). 3. The system processes the user's input and retrieves data on the distribution of property types in Sydney’s Airbnb scene across the dataset by analysing property descriptions and reviews. 4. The system generates a listing of the user’s desired property type as per their input has requested. |
| **Post-Conditions** | The user is presented with a visualization that showcases the distribution of property types, helping them identify their preferred property type for accommodations. |
| **Alternate Flow** | If there is an error during the processing of the user's input:  The system displays an error message and guides the user to retry inputting the property type preference. |

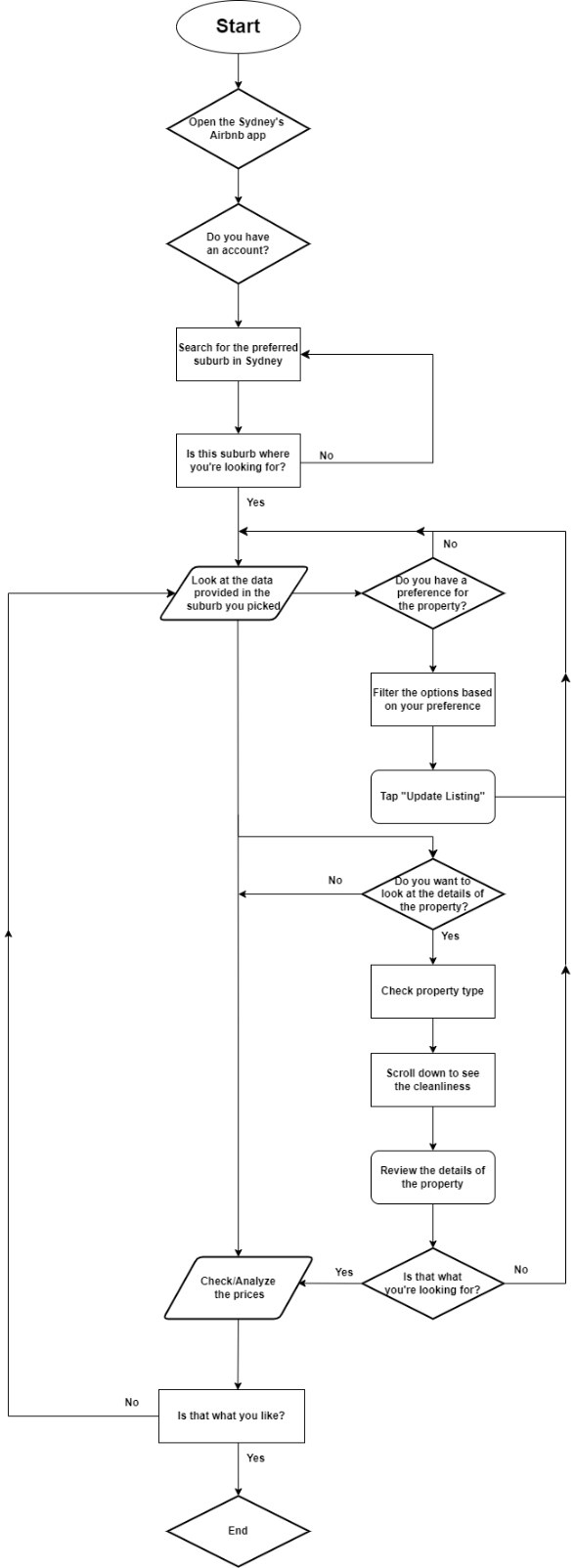
Notes: Only One property type can be requested at a time.

Use Case Diagram 5. Use Case ID 5.



# Software Design and System Components

## Software Design



## System Components

### Functions

**Function: get\_listings\_in\_suburb**

Description: Retrieves information about all listings in a specified suburb for a user-selected period.

Input Parameters: suburb (string), start\_date (date), end\_date (date).

Side Effects: None.

Return Value: List of listing information (objects).

**Function: generate\_price\_distribution\_chart**

Description: Creates a chart showing the distribution of prices of properties for a user-selected period.

Input Parameters: listings (list of objects), start\_date (date), end\_date (date).

Side Effects: None.

Return Value: Chart image file path or bytes.

**Function: search\_by\_keyword**

Description: Retrieves all records containing a user-specified keyword for a user-selected period

Input Parameters: keyword (string), start\_date (date), end\_date (date).

Side Effects: None.

Return Value: List of matching records.

**Feature 4: Cleanliness Analysis** 4. **Function: analyze\_cleanliness\_comments**

Description: Analyzes comments related to cleanliness factors and returns the number of customers who mentioned them.

Input Parameters: reviews (list of objects), cleanliness\_keywords (list of strings).

Side Effects: None.

Return Value: Number of customers who commented on cleanliness.

**Feature 5: Custom Insight or Analysis Tool** 5. **Function: custom\_insight\_analysis**

Description: Performs a custom analysis or insight generation based on user-defined criteria.

Input Parameters: criteria (various data types), additional\_parameters (various data types).

Side Effects: Depends on the specific analysis performed.

Return Value: Analysis result or insights.

**1) User Registration:**

Description: Allows users to create new accounts.

Input Parameters: User details (name: string, email: string, password: string).

Side Effects: Creates a new user account in the database.

Return Value: None.

**2) User Login:**

Description: Allows users to log in to their accounts.

Input Parameters: User credentials (email: string, password: string).

Side Effects: Sets the user's login session.

Return Value: User ID or success indicator.

**3) Search Listings:**

Description: Enables users to search for properties based on various criteria.

Input Parameters: Location: string, dates: date range, preferences: object.

Side Effects: None.

Return Value: List of matching property IDs.

**4) View Listing Details:**

Description: Displays detailed information about a selected listing.

Input Parameters: Listing ID: string.

Side Effects: None.

Return Value: Listing details object.

**5) Submit Booking Request:**

Description: Allows users to submit booking requests for specific dates.

Input Parameters: Listing ID: string, requested dates: date range, guest count: integer.

Side Effects: Adds a booking request to the database.

Return Value: Booking request ID.

**6) Check Availability:**

Description: Checks if a property is available for the requested dates.

Input Parameters: Listing ID: string, requested dates: date range.

Side Effects: None.

Return Value: Availability status (boolean).

**7) Confirm Booking:**

Description: Confirms a booking after availability is confirmed.

Input Parameters: Booking details: object, payment information: object.

Side Effects: Confirms the booking in the database, updates property availability.

Return Value: Booking confirmation ID.

**8) Process Payment:**

Description: Handles payment processing for confirmed bookings.

Input Parameters: Payment details: object.

Side Effects: Processes the payment through a payment gateway.

Return Value: Payment success status (boolean).

**9) Host Interaction:**

Description: Facilitates communication between guests and hosts.

Input Parameters: Messages: string, requests: object, responses: object.

Side Effects: Updates message history.

Return Value: None.

**10) Send Check-In Instructions:**

Description: Sends check-in instructions to guests before arrival.

Input Parameters: Booking details: object.

Side Effects: Sends email or notification to the guest.

Return Value: None.

**11) Guest Check-Out:**

Description: Marks the end of a guest's stay.

Input Parameters: Check-out date: date.

Side Effects: Updates booking status and availability.

Return Value: None.

**12) Leave Reviews and Ratings:**

Description: Allows guests and hosts to leave reviews and ratings.

Input Parameters: Review text: string, rating: integer.

Side Effects: Updates review data in the database.

Return Value: Review submission status (boolean).

**13) Manage User Profile:**

Description: Lets users view and update their profile information.

Input Parameters: Profile details: object, updates: object.

Side Effects: Updates user profile data in the database.

Return Value: Updated profile data.

**14) Notifications:**

Description: Sends notifications to users for various events.

Input Parameters: Event triggers: object.

Side Effects: Sends notifications to users.

Return Value: None.

**15) Customer Support:**

Description: Provides customer support to users.

Input Parameters: Support request details: object.

Side Effects: None.

Return Value: Support response object.

### Data Structures / Data Sources

**1) Database (Relational Database Management System):**

Type of Structure: Relational Database

Description: Stores all application data including user profiles, property listings, bookings, reviews, and more.

Data Members: Tables with various fields for different data types (e.g., Users, Listings, Bookings).

Functions Using It: Almost all functions interact with the database for data retrieval, updates, and storage.

**List of Listings:**

* Type of Structure: List (Array or Linked List)
* Description: Stores property listings.
* Data Members: Each listing is an object with attributes like ID, location, price, etc.
* Functions Using It: Search functions, price distribution, suburb listings, keyword search.

**List of Bookings:**

* Type of Structure: List (Array or Linked List)
* Description: Stores booking information.
* Data Members: Each booking is an object with attributes like ID, guest ID, listing ID, dates, status.
* Functions Using It: Booking functions, availability checks, conflicts resolution.

**List of Reviews:**

* Type of Structure: List (Array or Linked List)
* Description: Stores user reviews.
* Data Members: Each review is an object with attributes like ID, reviewer ID, reviewed ID, text, rating.
* Functions Using It: Review aggregation, cleanliness analysis, keyword search.

**2) User Profile Object:**

Type of Structure: Object

Description: Represents user profile information.

Data Members: User ID, name, email, password, preferences, etc.

Functions Using It: User registration, user login, manage profile.

**3) Property Listing Object:**

Type of Structure: Object

Description: Represents a property listing with details.

Data Members: Listing ID, location, description, photos, amenities, availability, etc.

Functions Using It: Search listings, view listing details, submit booking request.

**4) Booking Object:**

Type of Structure: Object

Description: Represents a booking request or confirmed booking.

Data Members: Booking ID, guest ID, listing ID, dates, status, payment details, etc.

Functions Using It: Submit booking request, confirm booking, process payment.

**5) Message Thread (Linked List or Array of Objects):**

Type of Structure: Linked List or Array

Description: Manages communication between users (guests and hosts).

Data Members: Messages (text, sender, timestamp).

Functions Using It: Host interaction, communication between users.

**6) Review Object:**

Type of Structure: Object

Description: Represents a user review for a property or a guest/host.

Data Members: Review ID, reviewer ID, reviewed ID, text, rating, timestamp.

Functions Using It: Leave reviews and ratings.

**7) Notification Queue (Priority Queue or List of Objects):**

Type of Structure: Priority Queue or List

Description: Holds notifications for users to be delivered.

Data Members: Notifications (message, type, timestamp).

Functions Using It: Notifications for various events. Functions Using It: Notifications for various events, dispatching notifications.

**8) Search Preferences Object:**

Type of Structure: Object

Description: Represents user preferences for searching listings.

Data Members: Location, dates, amenities, price range, etc.

Functions Using It: Search listings based on preferences.

**9) Support Request Object:**

Type of Structure: Object

Description: Represents a user's support request.

Data Members: Request ID, user ID, issue description, status, timestamp.

Functions Using It: Customer support interaction.

**10) External APIs (Geolocation, Payment Gateways, etc.):**

Type of Structure: External Data Source

Description: Interfaces with external services like geolocation for maps, payment gateways for payments, etc.

Data Members: API endpoints, authentication tokens, request/response formats.

Functions Using It: Geolocation services, payment processing.

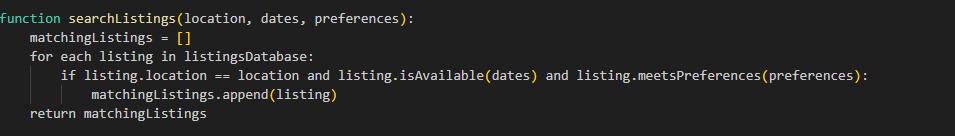
**Chart Data Structure (e.g., Dictionary or Custom Class):**

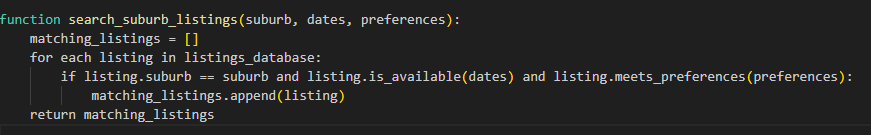
* Type of Structure: Custom Data Structure
* Description: Stores data for generating charts.
* Data Members: Various attributes based on the chart's data requirements.
* Functions Using It: Price distribution chart generation, custom insight analysis.

**Cleanliness Keywords List:**

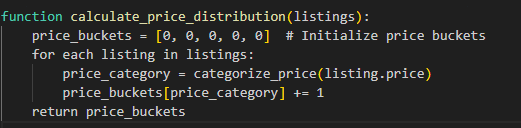
* Type of Structure: List (Array or Linked List)
* Description: Stores keywords associated with cleanliness comments.
* Data Members: List of cleanliness-related keywords (strings).
* Functions Using It: Cleanliness analysis, review aggregation.

### Detailed Design

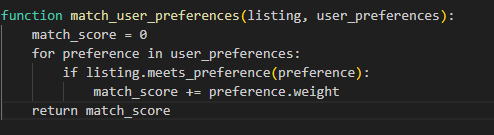
Search Listing:  
  


**Suburb Listing Algorithm:**   


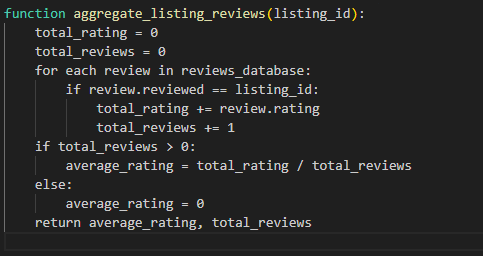
**Price Distribution Algorithm:**



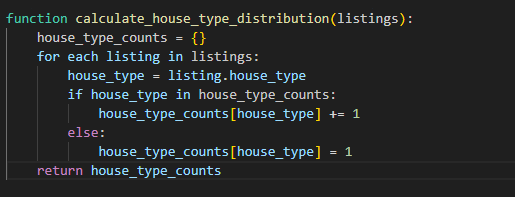
**User Preference Algorithm:**



**Review Algorithm:**



**House Type Distribution Algorithm:**



# User Interface Design

For this initial interface design, the goal is to create a simple and user-friendly website that allows users to compare prices and explore listings in specific suburbs within Sydney. This design aims to provide a seamless and efficient user experience, focusing on ease of navigation, search functionality, and displaying relevant information.  
  
**Conceptual Design Phase:**

Initiate the design process by brainstorming and creating conceptual sketches, either on paper or using a digital sketching tool. This step aids in visualizing layout concepts and page structures, providing an initial foundation for the design.  
  
**Key Findings and Considerations:**

**User Persona and Needs:** Understand the primary user demographic and their specific requirements. Tailor the design to cater to the needs of potential users, whether they are tourists, business travellers, or residents.

**Simplified Navigation**: Prioritize a straightforward and user-friendly navigation system. Include easily accessible menu options or a prominently placed search bar to ensure users can quickly find listings by suburb and compare prices.

**Suburb Selection:** Decide on an intuitive method for users to select specific suburbs. Options include a dropdown menu, an interactive map interface, or a search bar with autocomplete suggestions for suburbs.

**Filtering and Sorting:** Provide users with filtering options to refine their search results based on criteria such as price range, property type, and amenities. Additionally, implement sorting features to allow users to arrange listings by price, rating, or other relevant factors.  
  
**Sub-Sections:**

In your actual design work, consider creating sub-sections that focus on specific aspects of the interface, such as:

**Homepage:** Design the main landing page, showcasing featured listings and the search functionality prominently.

**Search Results Page:** Create wireframes for the page where users view search results, including listing cards and filtering options.

**Listing Details Page:** Design the layout for individual listing pages, displaying comprehensive information about a property.

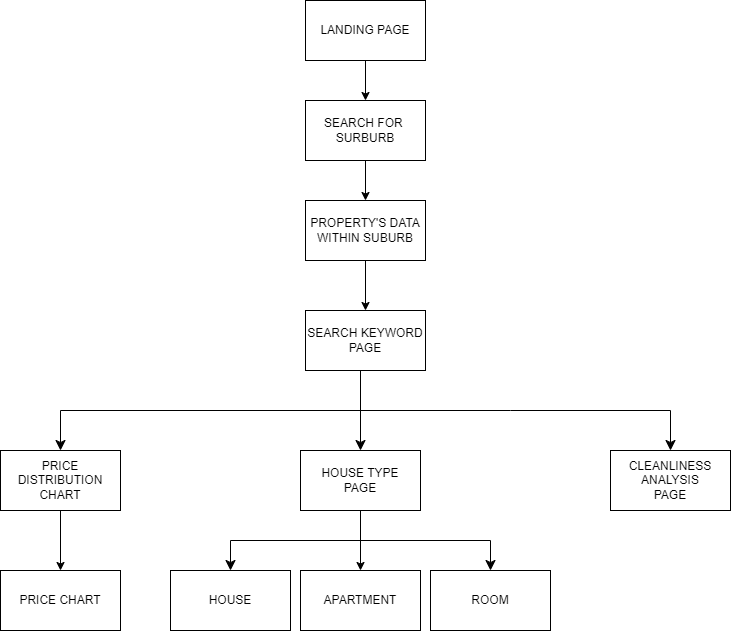
**Navigation and Menus:** Sketch out the navigation menu, header, and footer elements for consistent site-wide navigation.

**Suburb Selection:** Detail how users will select suburbs, whether through a dropdown, map, or search bar.

**Filtering and Sorting:** Design wireframes for filtering and sorting options to help users refine their search results.

## Structural Design

**Interface Structure:**



**Homepage:** The homepage serves as the entry point, featuring a search bar prominently to encourage users to begin their search immediately. It also includes:

**Featured Listings:** A section showcasing popular or recommended listings.

**Quick Filters:** Options for users to filter listings by popular criteria (e.g., price range, property type).

**Suburb Selection:** A clear method for users to select a specific suburb.

**Search Results Page:** This page displays listings matching the user's search criteria. The structure includes:

**List/Grid View**: Users can switch between list and grid views for listing cards.

**Sorting Options**: Filters for sorting search results (e.g., price, rating).

**Listing Details Page**: Accessed by clicking on a listing card, this page provides comprehensive information about a property.

**Property Details:** Specifics like price, location, and amenities.

***Information Structure:***

**Property Listings:** Listings are organized by suburb and include details like property type, price, amenities.

**Reviews:** User-generated reviews are linked to specific listings and hosts, providing valuable information for potential guests.

**Suburbs and Locations:** Information about different suburbs in Sydney is available, including popular attractions and amenities.

**Navigation:**

**Primary Navigation:** A top navigation bar or menu provides access to essential sections such as Home, Listings, User Profile, and Suburbs.

**Search Bar:** A prominent search bar is available on every page, allowing users to search for listings by suburb, date, or keywords.

**Justification of Design Choices:**

**Homepage Emphasis:** Placing the search bar on the homepage encourages users to start their search immediately, which is the primary goal of the website.

**Search Results Flexibility**: Offering list and grid views, sorting, and filtering options on the search results page provides users with flexibility and customization in exploring listings.

**Comprehensive Listing Details:** The listing details page provides all relevant information in one place, simplifying the booking process and enhancing user trust.

**User-Centric Approach:** User profiles and booking management options cater to the needs of both guests and hosts, making it a user-centric platform.

**Consistent Navigation:** A consistent navigation structure and search bar placement across pages ensure a seamless and intuitive user experience.

**Structured Suburb Information:** Organizing information by suburb allows users to explore different areas of Sydney, helping them make informed decisions about their stay

## Visual Design

**Layout:**

* The layout is clean and organized, with a focus on usability.
* Each page has a centered container to ensure content is not too wide on larger screens.
* Page titles are centered and prominently displayed at the top.
* Input elements, buttons, and results are neatly organized within the container.

**Visual Elements:**

* Input fields for selecting suburbs, dates, keywords, and house types are styled as dropdowns or date pickers for ease of use.
* Buttons are styled with a blue primary color to make them stand out.
* Tables are used to display data with column headers for clarity.
* Charts and graphs are included for visual data representation.

**Icons and Graphics:**

* Icons can be added next to input fields to enhance user understanding (e.g., a calendar icon next to the date picker).
* Placeholder images can be used for charts and graphs in the prototype.

**Style:**

* The overall style is minimalist and modern.
* Consistent spacing and alignment are used to maintain a neat appearance.
* Borders and shadows can be added to input fields and buttons to make them appear interactive.

**Color:**

* A primary color like blue (#007bff) is used for buttons and important elements to create a visual hierarchy.
* A neutral background color (e.g., white or light gray) is used for the main content area.
* Charts and graphs can use a variety of colors to represent data points.

**Fonts:**

* Standard web-safe fonts like Arial or sans-serif are used for readability.
* Font sizes are consistent and chosen for readability on various screen sizes.

**General Screen Designs:**

1. **Suburb Listings Page:**

A screenshot of a computer

Description automatically generated

* + Input fields for suburb selection and date picking.
  + A "Generate Report" button.
  + A table displaying listing information.

1. **Price Distribution Chart Page:**

A screen shot of a chart

Description automatically generated

* + A date picker.
  + A "Generate Chart" button.
  + A placeholder chart image.

1. **Keyword Search Page:**

A screenshot of a computer

Description automatically generated

* + An input field for entering keywords.
  + A date picker.
  + A "Search" button.
  + A table displaying records.

1. **Cleanliness Analysis Page:**

A screenshot of a computer

Description automatically generated

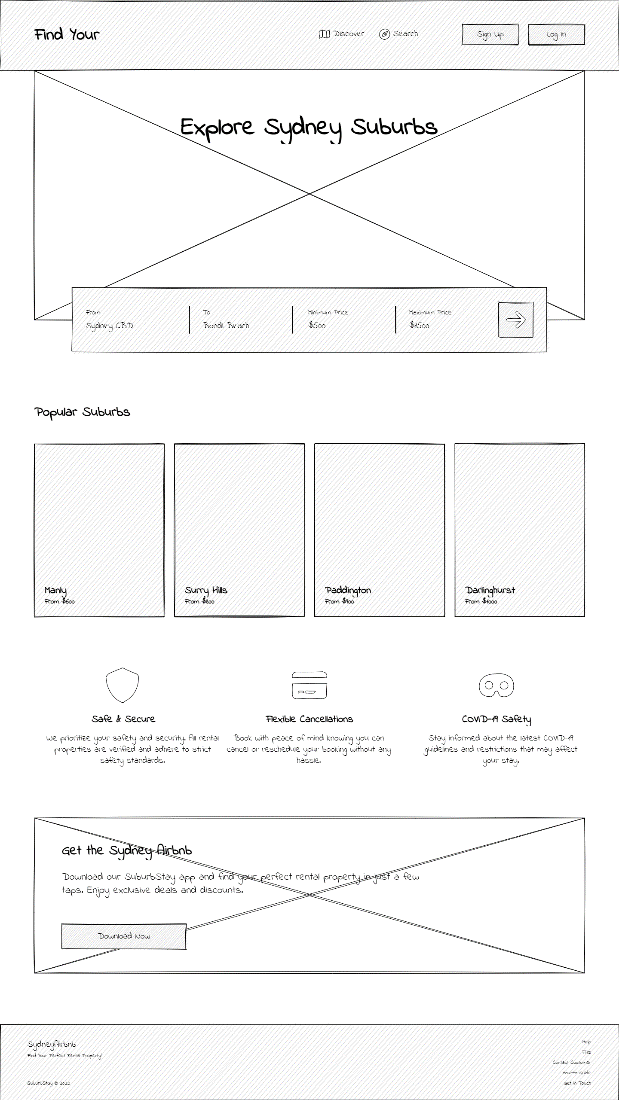
* + A keyword selection interface.
  + A date picker.
  + A "Generate Analysis" button.
  + A bar chart or data summary.

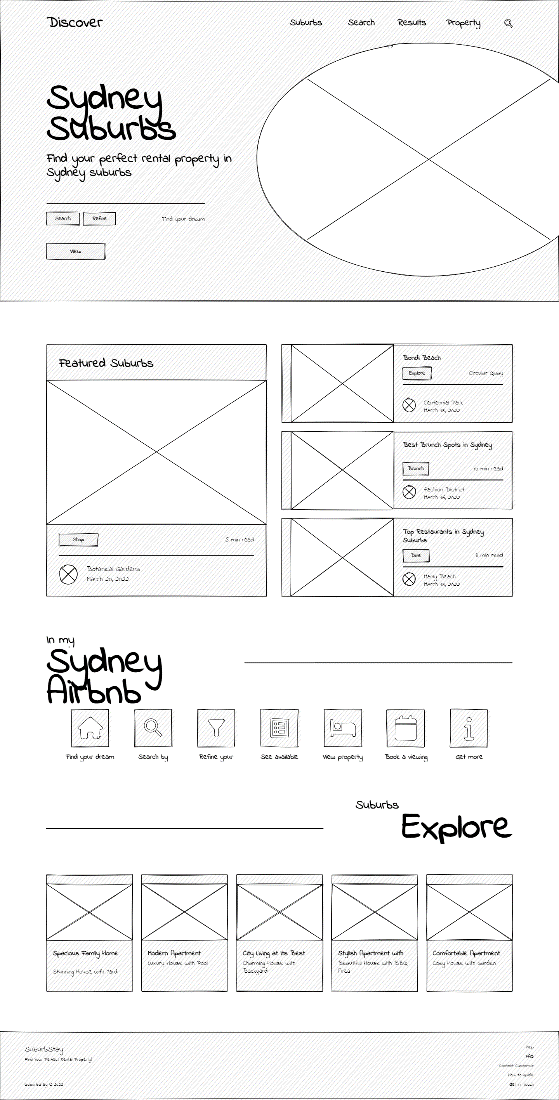
1. **Additional Insight Page (House Type Analysis):**

A screenshot of a computer

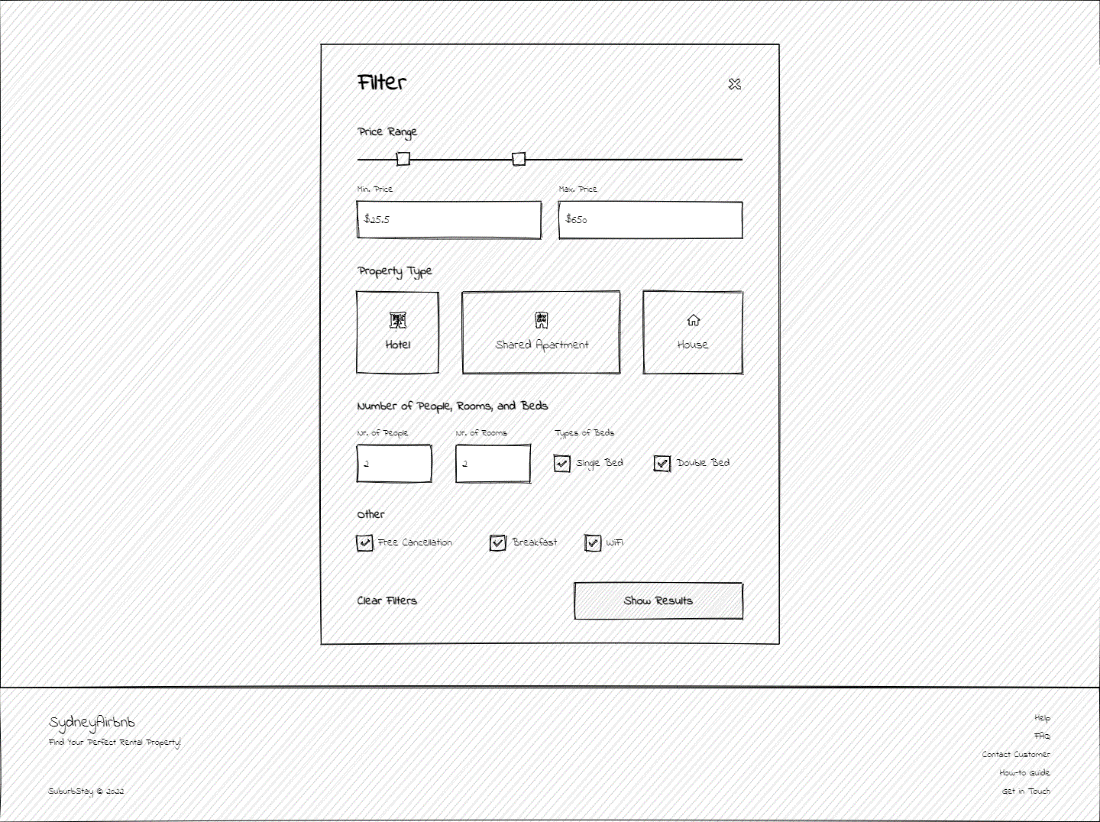
Description automatically generated

* + A dropdown menu for selecting house types.
  + Parameters or options related to the chosen insight.
  + A "Generate Insight" button.
  + Charts, tables, or summaries for displaying insights

Using sketches in the design process offers a rapid and flexible way to ideate and iterate on design concepts. Sketches are low-fidelity, focusing on fundamental layout and structure without getting bogged down in fine visual details, making them ideal for the early design stages. They are a cost-effective means of communicating ideas with stakeholders and team members, conveying the overall concept efficiently. Sketching provides the freedom to choose between traditional paper sketching or digital tools, adapting to individual preferences. This approach is particularly valuable when exploring multiple design directions and laying the foundation of a project's design, offering a quick and accessible method for conceptualizing ideas.  
  
**Landing Page:**  
  
The landing page features a captivating hero image of Sydney as its background and a central search bar for a clean and inviting start. Warm colors create an inviting atmosphere, with bold fonts for clear headers. Minimal navigation options focus on initiating user searches.  
  
**Discover Page:**

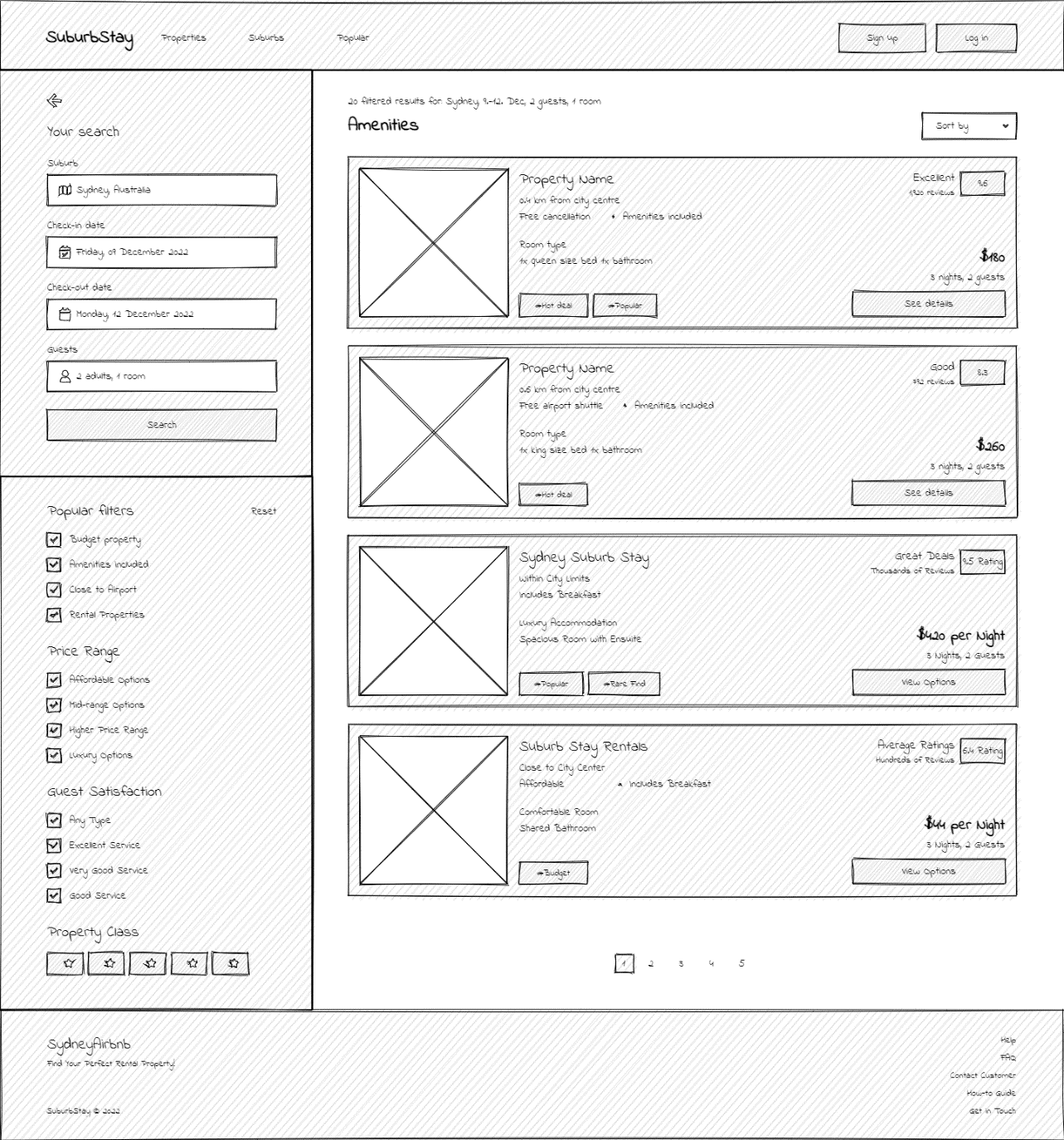


The discover page showcases property listings in a clean, neutral layout with grid-based cards. Icons and subtle color accents aid user interactions, while clear typography ensures readability. This design encourages exploration and comparison.  
  
**Search Page:**



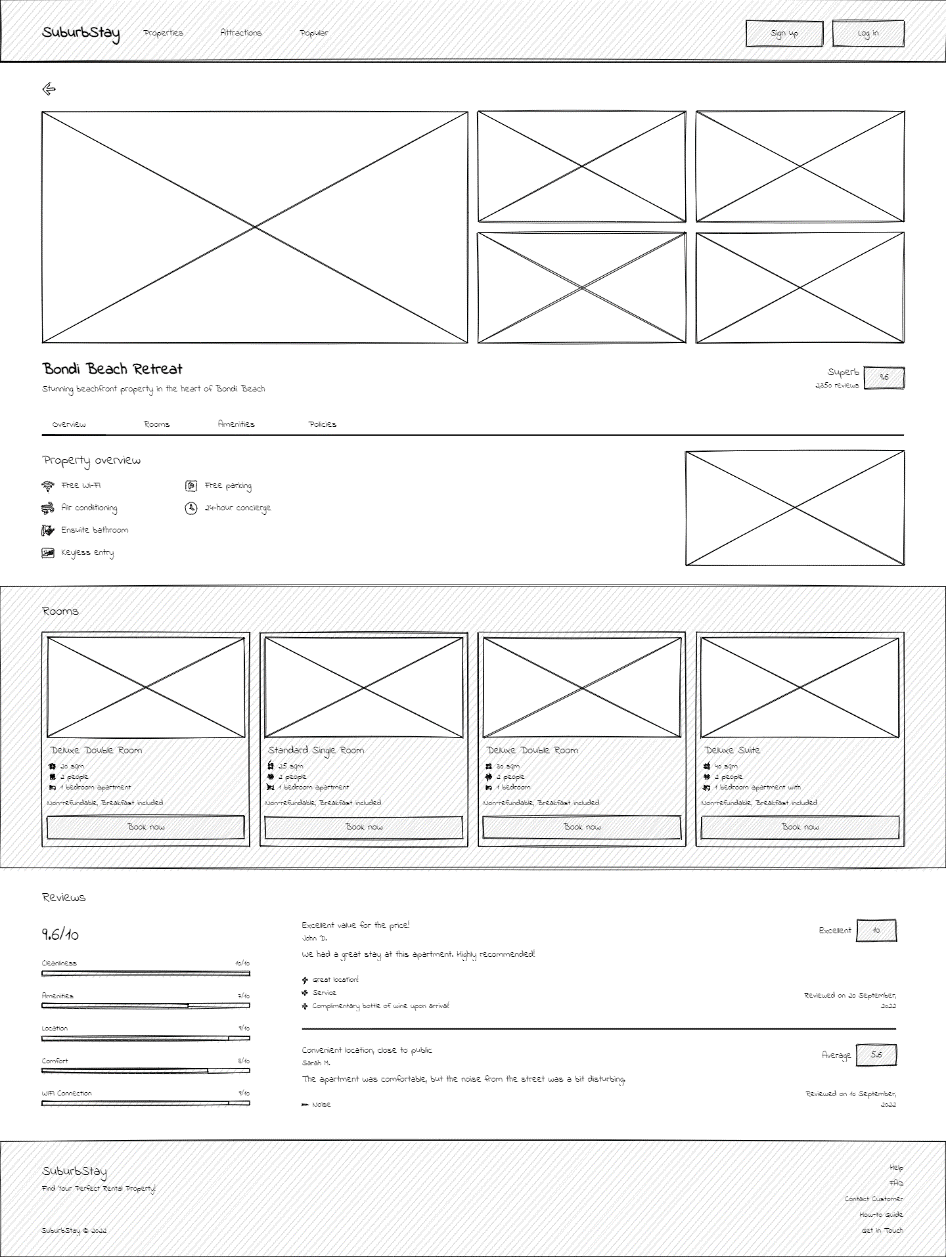
The search page maintains a clean and focused layout with a prominent search bar and a light, readable background. Subtle color accents distinguish the filtering panel. User-friendly filters make it easy for users to refine their searches.

**Search Results:**



Search results are displayed in list or grid views, with alternating background colors in the list view. Color coding aids quick decision-making for price ranges and availability. Sorting and filtering options are accessible at the top for user control.

**Property Detailed Page:**



Property detail pages maintain a cohesive design, offering comprehensive information in a single-scroll layout. Carefully selected fonts ensure readability, and icons enhance actions like favoriting properties. High-quality images, availability calendars, and reviews create a user-focused experience.

**Visual Elements:**  
  
The visual elements of the website have been meticulously designed to ensure a clean and user-friendly interface. The layout is characterized by its simplicity, focusing on providing a seamless user experience. Each page is structured logically, guiding users from the initial search for listings through to the detailed property information. Icons play a crucial role in enhancing user interactions. They are used consistently throughout the site to represent actions such as favoriting properties, checking availability, or sharing listings on social media. These icons are not only visually appealing but also intuitively convey functionality, adding valuable visual cues to the user interface. High-quality property images are strategically placed to showcase listings effectively. Vector graphics are employed for icons and buttons, ensuring crisp and clear visuals across various screen sizes. The overall style is modern and minimalistic, avoiding unnecessary ornamentation and clutter to prioritize clarity, usability, and a user-centric experience.

**Color Scheme:**  
In terms of the color scheme, careful consideration has been given to evoke a warm and inviting atmosphere. The hero image of Sydney on the landing page is overlaid with a subtle color filter, enhancing the visual appeal of the image while maintaining text legibility. The background color of listing cards on the discover page and search results is kept clean and neutral. This choice ensures that property images and details take center stage, providing a visually pleasing yet unobtrusive backdrop for property information. Subtle color accents are applied to buttons and interactive elements to draw users' attention to actions such as "View Details" or "Book Now." These accents create a visually pleasing contrast without overwhelming the design.  
  
**Fonts:**  
Regarding fonts, the website's typography choices have been made to prioritize readability and consistency. Bold and attractive fonts are used for headers and titles, making them visually distinct and emphasizing their importance. This ensures that users can quickly identify key sections and headings. For body text, paragraphs, and general content, a legible sans-serif font is employed. The font size has been carefully selected to ensure ease of reading on screens of all sizes, contributing to an accessible and user-friendly design.

**Justification:**

Clean and Uncluttered: The design choices aim for simplicity and ease of use to ensure that users can quickly find the information they need without distractions.

Responsive Design: A responsive layout adapts to various screen sizes, ensuring a consistent and user-friendly experience on both desktop and mobile devices.

Consistency: Consistent use of fonts, colors, and icons across the interface helps users navigate the website intuitively.

Clear Hierarchy: Headings, fonts, and contrasting colors are used to establish a clear hierarchy of information, making it easy for users to scan and find what they're looking for.

User-Centric: The design choices prioritize the user's experience, focusing on readability, accessibility, and an efficient search and browsing process.