Software Design Document

<Project Name>

Student Names

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

## System Overview

## Potential Benefits

# 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?**

## 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.

## Use Cases & Use Case Diagrams

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

# Software Design and System Components

## Software Design

A block diagram/flowchart of how your software might work

## 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

### 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

### Detailed Design

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

# User Interface Design

For the overall design of our program, we utilised draw.io, a wireframing tool, to create a visual representation of the user interface. This enabled us to iteratively refine the layout and functionality before moving to the development phase. Our key findings that guided this design include:

* User-Centric Approach:

Our search functionality encourages users to filter their search based on their preferences. By allowing users to input their preferred suburb, desired date range, price range, minimum rating, property type and room type, we ensure that the search results are personalised to their requirements.

* Information Hierarchy:

We recognised the importance of presenting essential information upfront. The concise display of listing details, including bedrooms, bathrooms, ratings, prices, and property types, facilitates quick decision-making by providing users with the most pertinent information at a glance. Other information contained in the original datasets that we do not think is as important will not be displayed on the program to avoid overstimulating the user.

* Data Visualisation:

The inclusion of a price distribution chart enhances user understanding of the pricing landscape within their selected filters. This visual aid simplifies the interpretation of pricing ranges and supports users in making well-informed choices.

Key findings that influenced our visual design approach include:

* Colour Coding:

We incorporated colour coding to enhance user orientation. By using distinct blue buttons for data updates and orange buttons for opening new windows, users should find it easy to navigate through the interface. The strategic use of colour, helps users intuitively identify the purpose of each button and action, contributing to a smooth navigation experience.

* Readability and Typography:

Choosing simple and easy-to-read fonts like Georgia, Arial, or Times New Roman reflects our dedication to creating an interface that's visually appealing and user-friendly. This ensures that users can effortlessly grasp the information without any hassle.

* Clean Layout:

To maintain a clean and organised look, we chose to open new windows when creating a new dataset (keyword search, price distribution chart or suburb specific information). This decision enhances the user experience, preventing the interface from becoming cluttered with excessive data.

In the upcoming sub-sections, we delve into the finer details of the structural design, including the search functionality, result display, filters and sorting options, and additional features like the price distribution chart, keyword search, and suburb insights. This comprehensive design approach aims to provide users with an intuitive, informative, and visually pleasing interaction with the program.

## Structural Design

**Home Page:**

**Search Functionality:**

* Users use search bar to input their preferred suburb.
* Users can enter the date range they want to look at in the form DD/MM/YYYY to DD/MM/YYYY

No data will be displayed unless Suburb is entered. Users can initiate the search by clicking the "SEARCH" button.

**Result Display:**

A list of Airbnb listings matching the chosen suburb and date range is displayed.

Basic listing information is presented, including:

* Name
* Number of bedrooms
* Number of bathrooms
* Occupancy capacity
* Rating
* Price
* Property type (e.g., apartment, house, etc.)
* Superhost status

Users can access more details by clicking the "View More" button associated with each listing which will open the URL of the AirBnB listing in their preferred browser.

**Filters and Sorting:**

A sidebar accommodates filtering options for result refinement:

* Price Range: Users can set minimum and maximum price limits for result filtering.
* Rating: Users can exclude lower-rated listings by specifying a minimum rating.
* Property Type: Users can opt for specific accommodation types (e.g., apartments).
* Room Type: Users can opt for specific accommodation types (e.g., apartments).

Once entered, user will click on the “Apply” button and the dataset will be updated.

Sorting preferences can be tailored via dropdown menus when the title of a column in the dataset is clicked on:

* Name: A-Z or Z-A
* Type: A-Z or Z-A
* Bedrooms: Smallest to Largest or Largest to Smallest
* Bathrooms: Smallest to Largest or Largest to Smallest
* Occupants: Smallest to Largest or Largest to Smallest
* Rating: Smallest to Largest or Largest to Smallest
* Price: Smallest to Largest or Largest to Smallest

**Results Overview:**

An information snippet at the bottom right corner provides the count of listings that match the current filters.

**Open new window buttons (orange boxes on wireframes):**

* Price Distribution Chart
* Keyword Search
* Suburb Insights

**Price Distribution Chart:**

* Displays a visual diagram of the price range for all previously selected filters by user (suburb, date range, min rating, property type and room type)

**Keyword Search:**

* Users can search for specific terms or select predefined keywords (e.g., cleanliness, pool, parking, wi-fi, air conditioning, etc.).
* Once selected, user clicks “Go” button which displays data filtered by their keyword choice.

When the user clicks on the **“cleanliness”** checkbox, the dataset will display all listings that contain the following words:

* Clean
* Cleanliness
* Hygiene
* Tidy
* Tidiness
* Sanitary
* Neat
* Spotless
* Well-kept
* Dust
* Dirty
* Cleaned
* Unclean
* Dusty
* Unsanitary
* Messy
* Filthy
* Disgusting
* Shiny
* New

When the user clicks on the **“pool”** checkbox, the dataset will display all listings that contain the following words:

* Pool
* Swim
* Swimming
* Spa
* Jacuzzi

When the user clicks on the **“parking”** checkbox, the dataset will display all listings that contain the following words:

* Parking
* Garage
* Carport
* Vehicle
* Bike
* Automobile
* Parked
* Driveway
* Car
* Van
* Bus

When the user clicks on the **“wifi”** checkbox, the dataset will display all listings that contain the following words:

* WiFi
* Wi-Fi
* Internet
* WLAN
* Wireless
* Network
* Online
* Web
* NBN
* Broadband
* Hotspot

When the user clicks on the **“airconditioning”** checkbox, the dataset will display all listings that contain the following words:

* Airconditioning
* Aircon
* AC
* A/C
* Heating
* Cooling
* Fans
* Cold
* Hot
* Temperature
* Climate
* Air

**Suburb Insights:**

* User selects property type and room type then clicks Search button.
* Window displays a list containing all suburbs in original dataset sorted by average price for the user specified property type and room.
* Window displays a list containing all suburbs in original dataset sorted by average rating for the user specified property type and room.

**Justification:**

**Search Functionality:**

The search functionality is the core feature of the program. It allows users to find Airbnb listings based on their preferences. By including options to input the suburb and date range, it provides users with the flexibility to narrow down their search results.

**Results Display:**

The displayed information about each listing is concise and relevant, giving users a quick overview of what they can expect from the listings. Providing essential details like number of bedrooms, bathrooms, occupancy capacity, rating, price, and property type helps users make informed decisions without having to look through a gigantic dataset to find what they are looking for.

**Filters and Sorting:**

Filters and sorting options enhance the user experience by allowing users to refine their search results according to their specific criteria. This flexibility increases the likelihood of users finding listings that best match their preferences and requirements.

**Price Distribution Chart:**

The price distribution chart is an visual aid to help users understand the pricing landscape of the available listings. This feature can be especially helpful for users who want to see the range of prices within their chosen filters visually rather than in a dataset.

**Keyword Search:**

Including a keyword search feature helps users find listings with specific amenities or features that are important to them. This allows users to quickly narrow down their choices based on their preferences, saving time and effort.

**Suburb Insights:**

The suburb insights feature offers users valuable data for decision-making. Sorting suburbs by average price and average rating based on specific property and room types helps users identify potential suburbs that align with their budget and preferences.

**UI Design Choices:**

The use of checkboxes and dropdown menus for filtering and sorting offers a user-friendly interface. The layout of the filters and sorting options is designed to help users easily navigate and adjust their search criteria.

**View More Button:**

Providing a "View More" button that opens the Airbnb listing URL in the user's preferred browser ensures that users have easy access to detailed information about a listing. This button is designed to encourage users to explore the listing further and pull more traffic to the AirBnB website.

## Visual Design

This program is intentionally designed to be simple and user-friendly, ensuring even those unfamiliar with data can easily navigate through it. The user interface incorporates blue buttons for updating data within the current window, while orange buttons are used to open new windows. Each window is dedicated to specific content and features an appropriate title. To enhance readability, a basic and easily legible font like Georgia, Arial, or Times New Roman will be selected.

When creating a new dataset, it automatically opens in a separate window to maintain a clean and organized appearance, avoiding any visual clutter. The overall screen layout consists of a title, filters positioned either to the left or above the dataset, buttons to activate these filters, and a streamlined dataset displaying only relevant information rather than overwhelming users with extensive data.







