Software Design Document

<Project Name>

Student Names:

Ashley Pergoliti (s5311775)

Jonas Sajonas (s5284977)

Jodie Thomson (s118338)

Table of Contents

[1.0 System Vision 3](#_Toc142068931)

[1.1 Problem Background 3](#_Toc142068932)

[1.2 System Overview 3](#_Toc142068933)

[1.3 Potential Benefits 3](#_Toc142068934)

[2.0 Requirements 4](#_Toc142068935)

[2.1 User Requirements 4](#_Toc142068936)

[2.2 Software Requirements 4](#_Toc142068937)

[2.3 Use Cases & Use Case Diagrams 5](#_Toc142068938)

[3.0 Software Design and System Components 6](#_Toc142068939)

[3.1 Software Design 6](#_Toc142068940)

[3.2 System Components 6](#_Toc142068941)

[3.2.1 Functions 6](#_Toc142068942)

[3.2.2 Data Structures / Data Sources 6](#_Toc142068943)

[3.2.3 Detailed Design 6](#_Toc142068944)

[4.0 User Interface Design 7](#_Toc142068945)

[4.1 Structural Design 7](#_Toc142068946)

[4.2 Visual Design 7](#_Toc142068947)

# System Vision

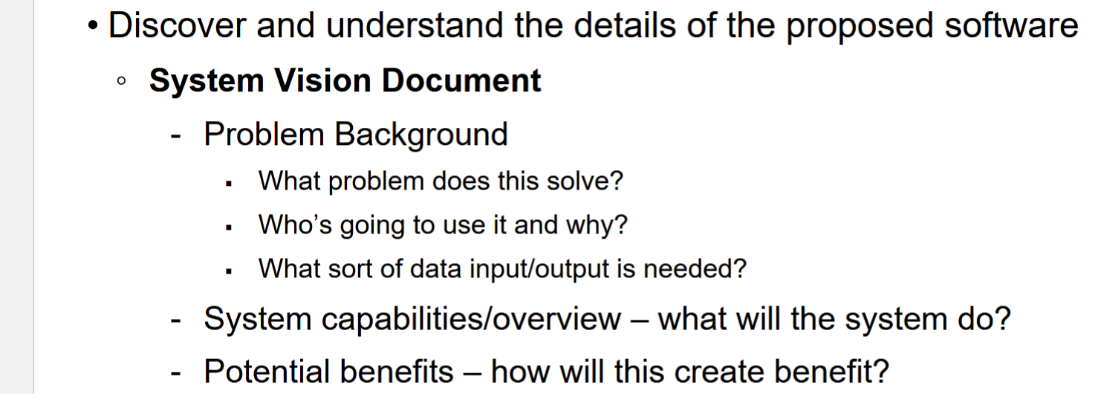
1. System Vision (10 marks)

Should include a background on the dataset, software overview and potential benefits of the software.

*Can incorporate maybe*

*This project will design and develop the software that reads data from the data set, performs user defined functions on the data and displays the results in a purpose-designed user interface. The capabilities will include:*

* *A search option that will allow a user to find listings targeting specific suburbs.*
* *For a user specified period, output a chart showing price distribution.*
* *For a user specified period and keywords, output records.*
* *Analysis of cleanliness from customer comments.*
* *Output charts showing variances in availability and associated pricing across the year - TBC*



## Problem Background

Excel sheets are widely used to store large sets of text data. Users face problems when trying to view these large sets of data. This can be due to there being many unnecessary sections of data. This makes it hard for a user to sort through the data for what they are looking for.

This applies to users trying to view data relating to Airbnb. Users want to pick a specific date and see information about different properties. This information includes the distribution of prices, listings by a specific suburb, records including user chosen keywords, and customers that commented on the cleanliness of the property.

It is recommended that a system is created to help these users view the data quickly and effectively.

## System Overview

This system should be able to complete the following:

* Create a user interface that has interactable buttons and input fields that change what is displayed on the interface.
* Read different CSV files.
* Display different data based on the options selected(options,keywords,ratings)

## Potential Benefits

This system will provide the following benefits for the users:

* View large sets of data through a smooth user-friendly interface.
* View this data in text form and graph form.
* Filter through the data for their specific chosen criteria.
* Speed up the process of analysing the data.
* Potential use to analyse other data.

# Requirements

2. Requirements (10 marks) There are 2 types of requirements to consider

## User Requirements

• User Requirements: How a user will interact with the program. What do they need to do ?

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?**

* Run the GUI script (run the application).
* Select a specific date period.
  + See the information of all suburbs listed under this date.
  + Able to view a chart that is created, which shows the distribution of prices of properties.
  + Able to enter a specific keyword to view all records that contain that keyword.
* Able to view and analyse places where customers have commented on the cleanliness of the Airbnb. ??
* Another unnamed “insight” or analysis tool of our choice. ??

## Software Requirements

Software Requirements: What functionality will the software provide (think functional 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 program shall be run as a single file in the command line.?
* The script shall open to a GUI interface.?
* The GUI interface shall have several buttons and other input fields that the user can interact with.
* These buttons shall trigger further actions and interfaces- a graph.

Functional Requirements:

* Read GUI script
* Display GUI
* Accept Input from user
* Display result of User interaction
* Read excel files
* Filter Keywords from the search query
* System returns none when search value has no matches
* Filter cleanliness keyword from comments
* Get ratings of listings
* Get Lists of Suburbs
* Get Dates of Listings
* Verify Search queries
* Produce charts
* Display all listings

(Check

Required Features:

For a user-selected period, report the information of all listings in a specified suburb

For a user-selected period, produce a chart to show the distribution of prices of properties

For a user-selected period, retrieve all records that contain a keyword (user entered), e.g. pool, pet.

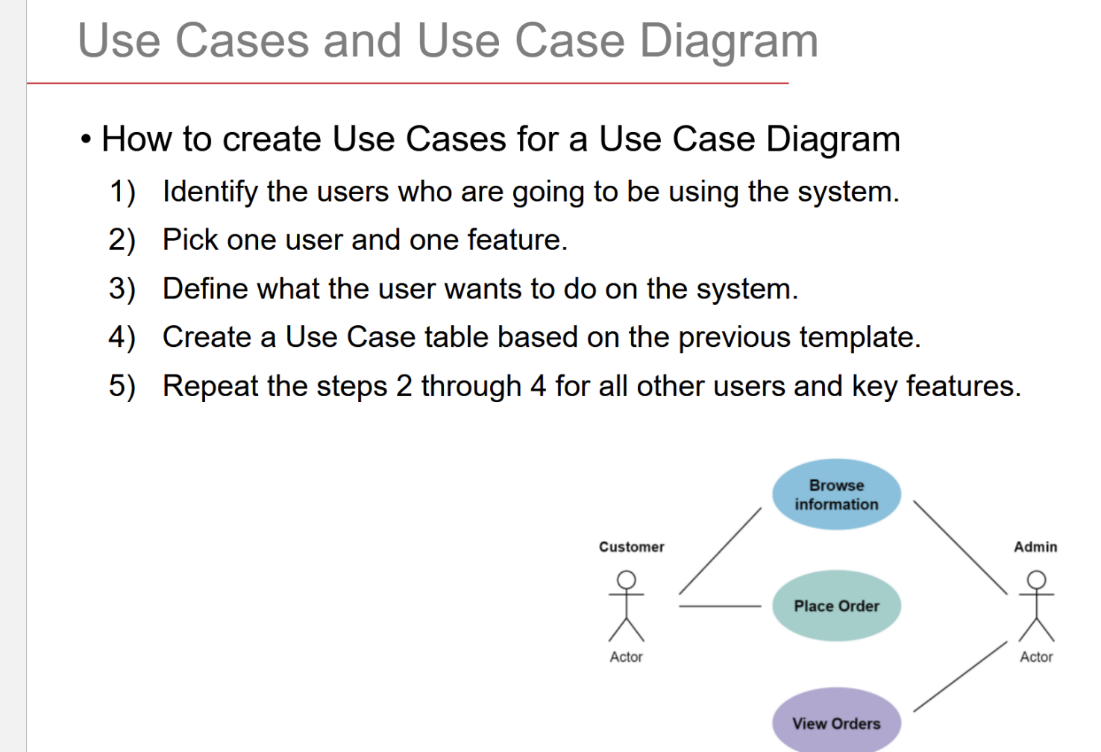
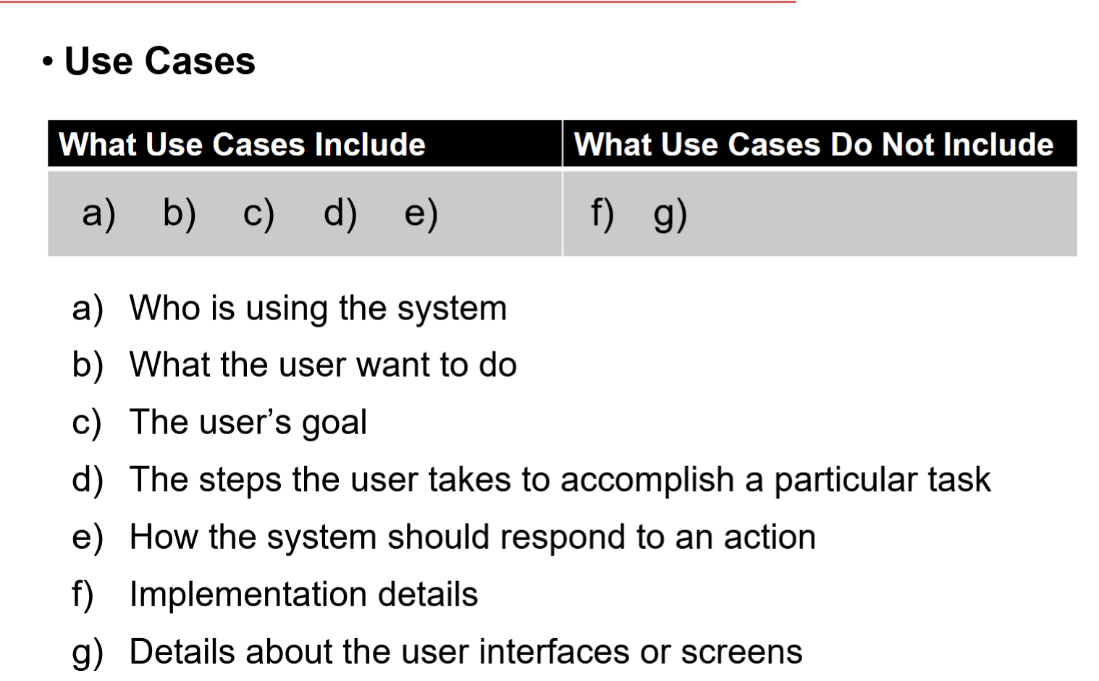
Analysing how many customers commented on factors related to cleanliness (multiple key words may be associated with cleanliness – justify your selection).

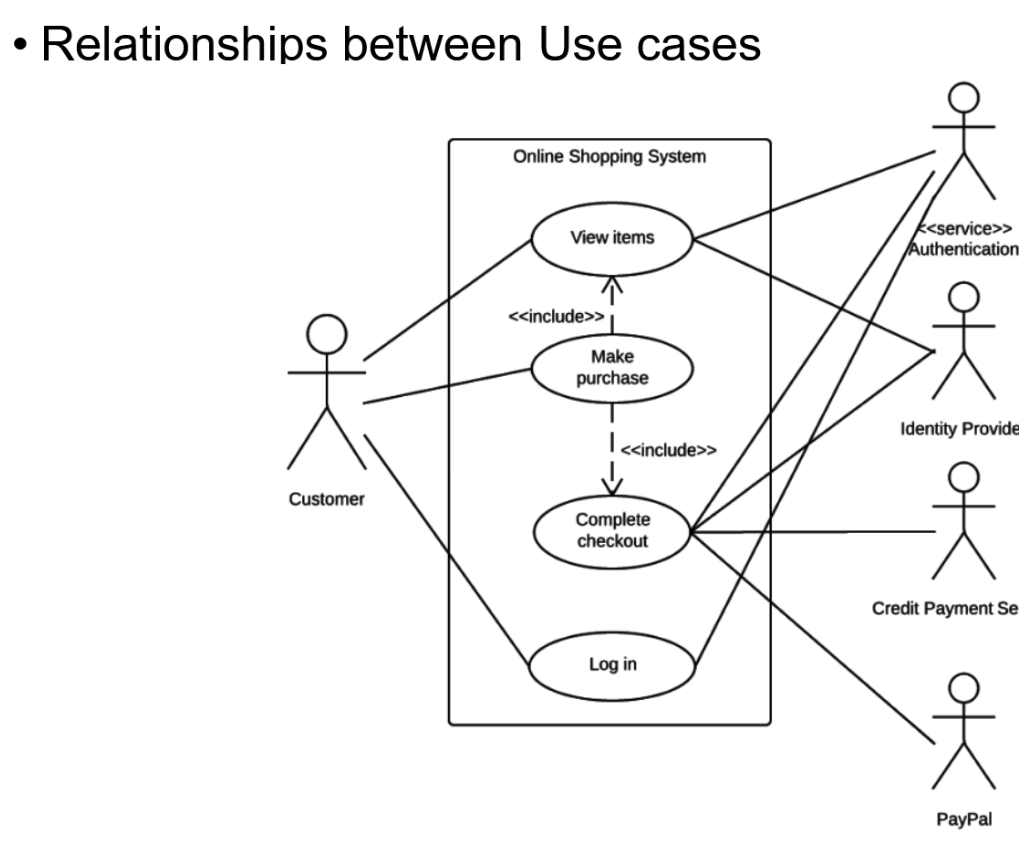
## Use Cases & Use Case Diagrams – need 2-3 • Use Cases: These Use Cases should show the blending of user and software requirements by identifying use cases and how the user will interact with the product. Any diagramming format is acceptable, but the diagrams should clearly display the sequence of events and interactions between the user and the software. Expecting about ~5 use cases (1 for each of the functions) and a few accompanying Use Case Diagrams.

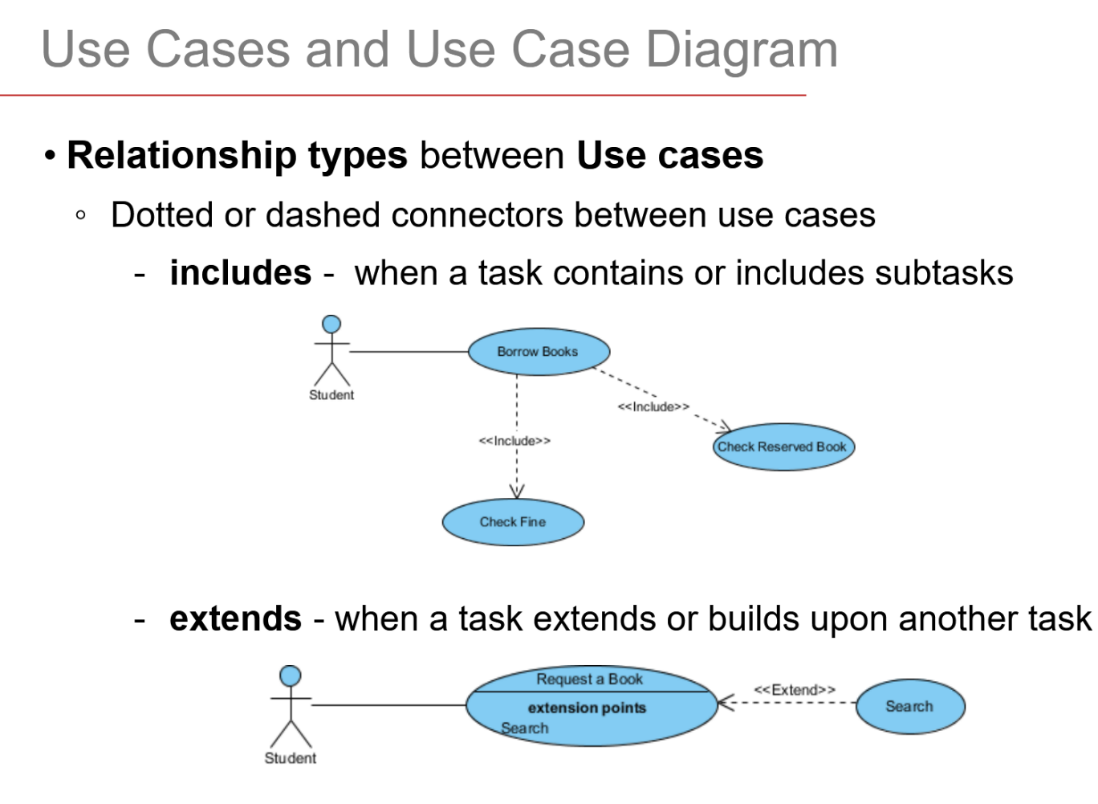
* In this section you provide some use cases showing how people may use your software.  
  *A search option that will allow a user to find listings targeting specific suburbs.*
* *For a user specified period, output a chart showing price distribution.*
* *For a user specified period and keywords, output records.*
* *Analysis of cleanliness from customer comments.*
* *Output charts showing variances in availability and associated pricing across the year - TBC*

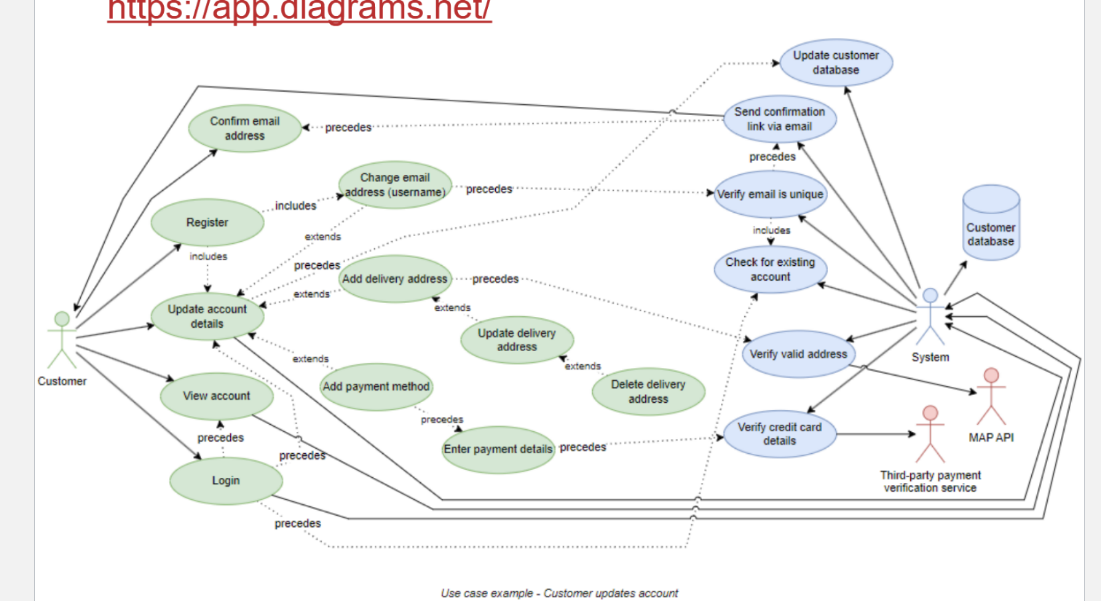
Beginning of use case diagram – idk if this is even close

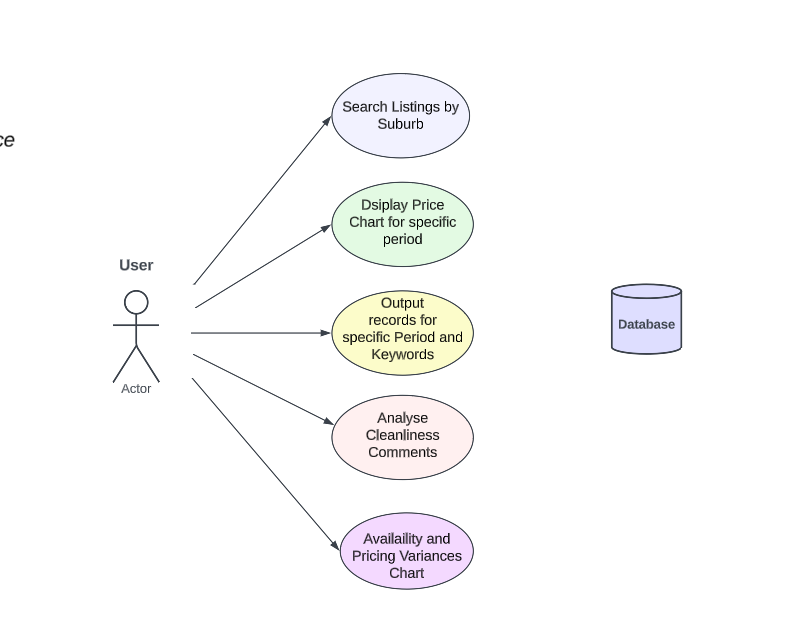
<https://lucid.app/lucidchart/05a101cd-3e34-424d-ae05-a2dfcf43707b/edit?viewport_loc=-1222%2C42%2C2758%2C1295%2C0_0&invitationId=inv_53257447-7ef1-422e-ac89-3be62abd9110>

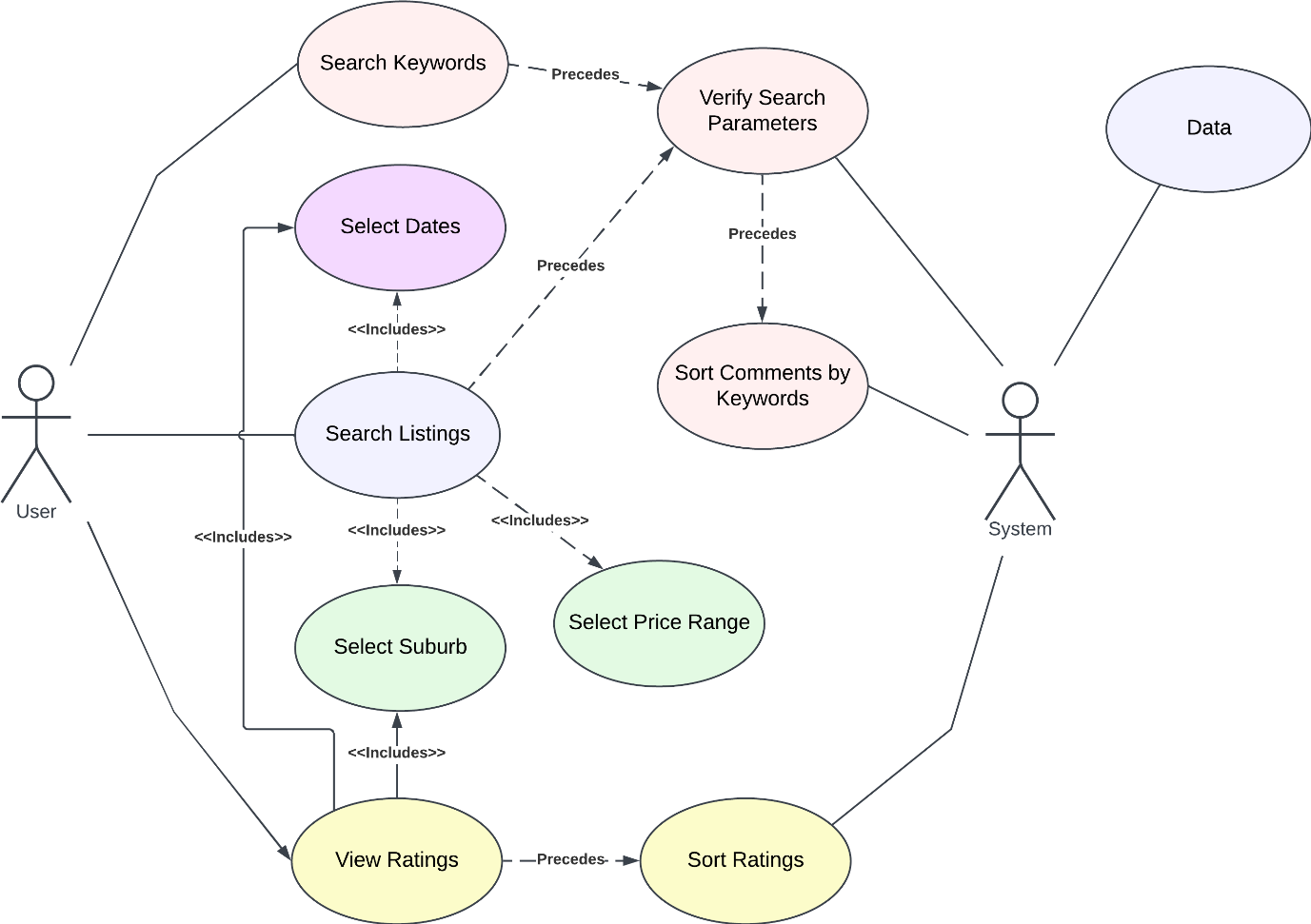












<<DRAFT>>

There was two ways the use cases were shown in the notes  
Jodie did some a couple of weeks ago but I think they got lost in the changed system back then – we now how two drafts, I added them below Jonas:

**Draft: Use Cases**

**The content notes show a couple of different ways**

|  |  |
| --- | --- |
| **Use case ID** | 1.0 |
| **Use Case Name** | Display Listings for Suburb |
| **Actors** | User |
| **Description** | For a user selected period and suburb, display all listings. |
| **Flow of Events** | 1. User Selects search suburb options 2. User Selects chosen time Frame from date pickers 3. User Enters chosen suburb 4. System Displays listings for the selected Suburb and time period |
| **Alternative flow** | 1. System returns no listings 2. User option to enter new dates and/or suburb |

|  |  |
| --- | --- |
| **Use case ID** | 2.0 |
| **Use Case Name** | Display Pricing Chart |
| **Actors** | User |
| **Description** | Display Chart for pricing distribution for specified time period |
| **Flow of Events** | 1. User Selects Chart option 2. User selects timeframe 3. System Displays chart |
| **Alternative Flow** | None  1. System returns no chart when it doesn’t have any match  2. User option to enter new dates and/or keywords. |

|  |  |
| --- | --- |
| **Use case ID** | 3.0 |
| **Use Case Name** | Display Records that contain Keywords |
| **Actors** | User |
| **Description** | Display all records that contain user selected keywords in a selected period |
| **Flow of Events** | 1. User Selects Keywords option 2. User Enters Keywords 3. User selects time period from date picker 4. System Displays all records that contain the keywords in that time period. |
| **Alternative Flow** | 1. System returns no listings 2. User option to enter new dates and/or keywords. |

|  |  |
| --- | --- |
| **Use case ID** | 4.0 |
| **Use Case Name** | Display Listings according to cleanliness |
| **Actors** | User |
| **Description** | Display analysis of cleanliness comments - ? Chart, stars? Most number of comments?  Display analysis of cleanliness from comments and ratings of the listing. |
| **Flow of Events** | 1. User Selects cleanliness analysis option 2. System Displays Results |
| **Alternative Flow?** | None |

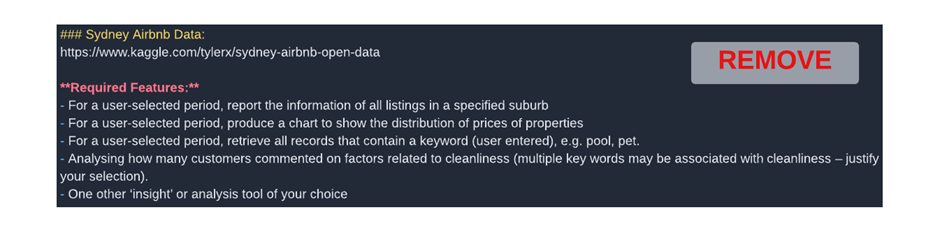
|  |  |
| --- | --- |
| **Use case ID** | 5.0 |
| **Use Case Name** | Display chart Availability and pricing  Display Listings with Ratings above 80 |
| **Actors** | User |
| **Description** | Display chart of availability and pricing across a calendar year  Display chart of Listings with ratings above 80 with highest on top. |
| **Flow of Events** | 1. User Chooses Pricing and Availability 2. User Chooses Calendar Year/s from selectors 3. System Displays one Chart for each year selected, showing both availability and price variances.   1. User chooses Listing and Ratings  2. User Chooses Calendar Year/s from selectors  3. System displays chart listings with highest ratings on top to bottom. |
| **Alternative Flow** | 1. System returns no Listing  2. User options to enter new dates. |

# 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

**Do we want separate functions for queries in each function?**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | **Description** | **Parameters** | **Side Effects** | **Return Value** | **Notes** |
| **def readExcel(fileName)** | A function that will be used to read the CSV files to grab the data. | The fileName variable will be a string, that will specify the name of the file. | There will be a global variable that will change to hold the dataframe of the Excel data. | Returns the dataframe of the data (of type DataFrame). This is a data representation of the CSV file. |  |
| **def selectDate(startDate,endDate)** | A function that will grab the date from the user-selected period. | The startDate parameter will hold the start date picked by the user in the start calendar.  The endDate parameter will hold the end date picked by the user in the end calendar. | This will set a global variable that holds the date range which will be used for other functions. | Return the date range. |  |
| **def startSession()** | A function that will initialise the application. It will create the graphical interface of the application, and it will call readExcel() to grab the required data from the Excel files. | No input parameters required. | Global variables will be changed from the readExcel() function. | No return value required. |  |
| **def displayPriceChart()** | This function will display a price chart for a user-selected period, to show the distribution of the prices of properties. It will be the command of a button. | No input parameters required. | This will create a matplotlib graph showing the data. | No return value required. | This function will use the data returned by getPriceChartData() |
| **def getPriceChartData(from, to, property)** | Using the date parameters, a query is created, and a list of property prices is returned. | From and to, will be the global variables that are holding the start and end dates the user selected respectively. Property will be a string variable holding an input from the user. | This will populate a variable with data, which will be used for displayPriceChart(). | The return value is a list of property prices. |  |
| **def displaySuburbListings()** | This function will display a chart with all listings for a specified suburb for a user-selected period. |  |  |  | This function will use the data returned by getSuburbListings(). |
| **def getSuburbListings(from, to, suburb)** | Using the 3 input parameters, a query is created, and the data received is all listings for the properties in that range. | From and to, will be the global variables holding the start and end dates the user selected respectively. Suburb will be a string variable holding an input from the user. |  | This function returns and displays all the listings based on the input variables. |  |
| **def displayKeywordResults()** |  |  |  | Returns the list of records that match the keywords and dates. |  |
| **def getKeywordResults(keyWords,from,to)** | Using the 3 input parameters, a query is created, and the data received is all the listings/records that are based on the parameters. | The keyWords variable will be an array of the cleaned input added by the user. From and to, will be the global variables holding the start and end dates the user selected respectively. |  | This function will return the data with the list of records that contain the keywords in that period. |  |
| **def displayOtherThing()** |  |  |  |  |  |
| **def getOtherThing(variable,variable)** |  |  |  | This function will return **x** data. |  |
| **def cleanUserInput(input)** | This function will take in the input of a user and clean the input. Remove special characters, split the input on the spaces. | The input variable ‘input’, will be the input entered by the user. |  | This function will return the array of the cleaned user input. |  |
| **def cleanExcelData(dataframe)** | This function will take in the dataframe, and will remove any records that are missing data in certain columns. | The input variable will be the dataframe of each data set. |  | This function will return the cleaned dataframe. |  |
| **def displayErrorMessage(errorMessage)** | This function will display an error message on screen for the user, on an error being triggered. | The input variable is errorMessage, which will be a string holding the error message that will be displayed on screen. |  | This function will return the error message. |  |
| **def cleanAnalysis()** |  |  |  |  |  |
| **def removeExtraColumns(dataframe)** | This function will remove columns from the input dataframe, that are not going to be used for displaying data. |  |  |  |  |
| **def clearSearchQuery()** | This function will clear the search box | The query is the input variable. | When search button is clicked it will not return anything. | Returns the search query as null |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### 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

<https://www.geeksforgeeks.org/data-structures/>

ChatGPT:

Creating a software app that takes Excel sheets and performs analysis involves dealing with data manipulation and storage. You'll need to choose appropriate data structures to efficiently handle the data. Here are some data structures you might consider using:

1. **Data Frame:** Data frames are a common choice for working with tabular data, such as Excel sheets. They allow you to store data in rows and columns, making it easy to perform operations on specific columns or rows. Libraries like Pandas (Python), data.table (R), or DataFrame (Julia) provide efficient data frame implementations.
2. **Arrays or Lists:** Depending on the complexity of your analysis, you might need arrays or lists to hold individual data points or values. These can be used to store data before processing it into a more structured format.
3. **Dictionaries or Hash Maps:** Dictionaries are useful for storing key-value pairs. You could use them to map column names or headers to their respective data, making it easier to reference and manipulate specific columns in your analysis.
4. **Graphs or Trees (if needed):** If your analysis involves relationships or hierarchies among data points, you might need to use graph or tree structures. These can help you represent and analyze connections between different elements in your data.
5. **Custom Objects:** Depending on the complexity of your analysis and the specific requirements of your app, you might define custom classes or structs to represent the data in a more meaningful way. This can help encapsulate data and relevant methods for analysis.
6. **Caches or Memoization (for optimization):** Depending on the type of analysis you're performing, you might need to cache or memoize intermediate results to improve performance. This can involve using data structures like caches or lookup tables to store and retrieve computed values.
7. **Stacks and Queues (if applicable):** If your analysis involves sequential processing or managing tasks in a certain order, stacks (last-in, first-out) or queues (first-in, first-out) can be helpful data structures.
8. **Sorting and Searching Data Structures:** Depending on the nature of your analysis, you might need sorting and searching algorithms or data structures like binary trees or hash-based structures to efficiently organize and find data.

Remember that the choice of data structures depends on the specific requirements of your analysis and the programming language you're using. Make sure to choose structures that align with the efficiency and performance goals of your app. For instance, Python with Pandas is a popular choice for data analysis due to its robust data frame implementation, while languages like R and Julia also have strong data manipulation capabilities.

### Detailed Design

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

# User Interface Design

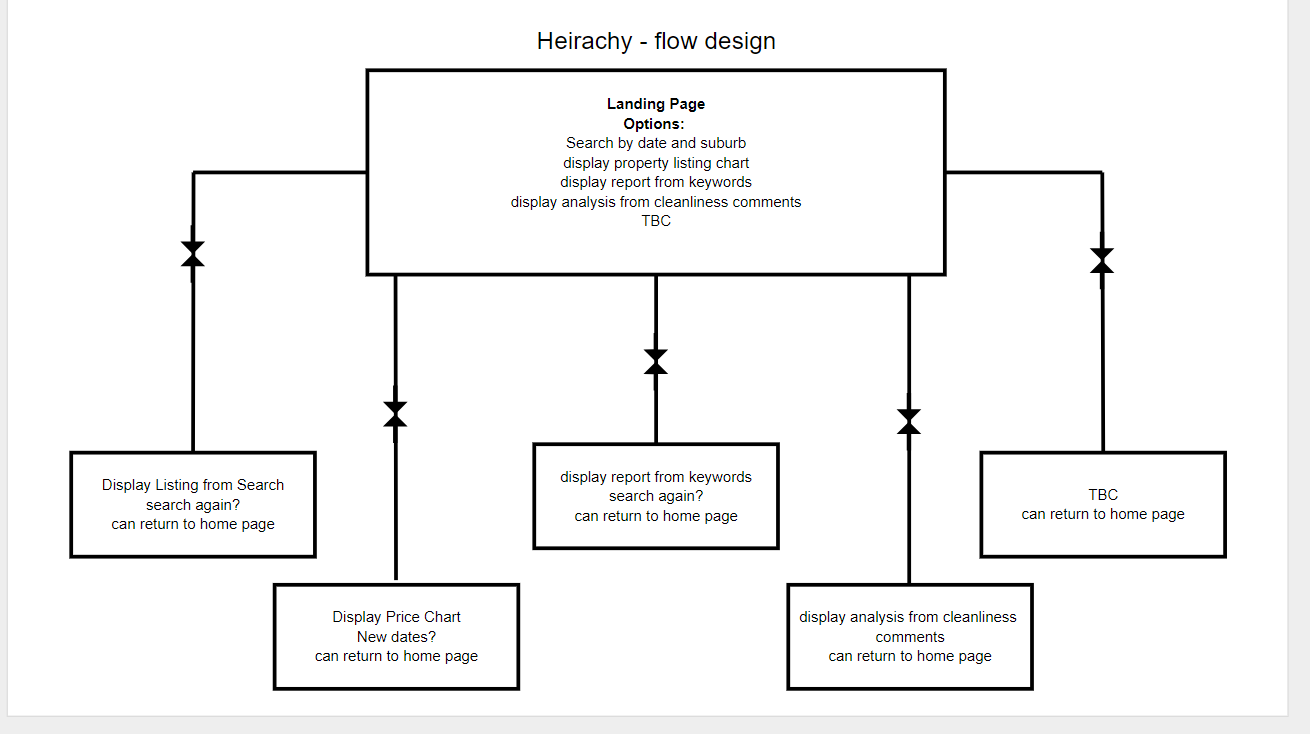
This is your initial interface design. Describe the tools you used for this design stage and any key findings that informed your design. **This introduction is descriptive and should explain what you have completed for the actual design work you will present in the sub-sections below.**

## 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.

Did this draft a couple of weeks ago, may or may not be suitable. A version of it may be fine with the detailed discussion -

<https://app.moqups.com/MB7IIIM7ZgfQlAGiH55X215qIB9xROpA/view/page/a421b9a23>

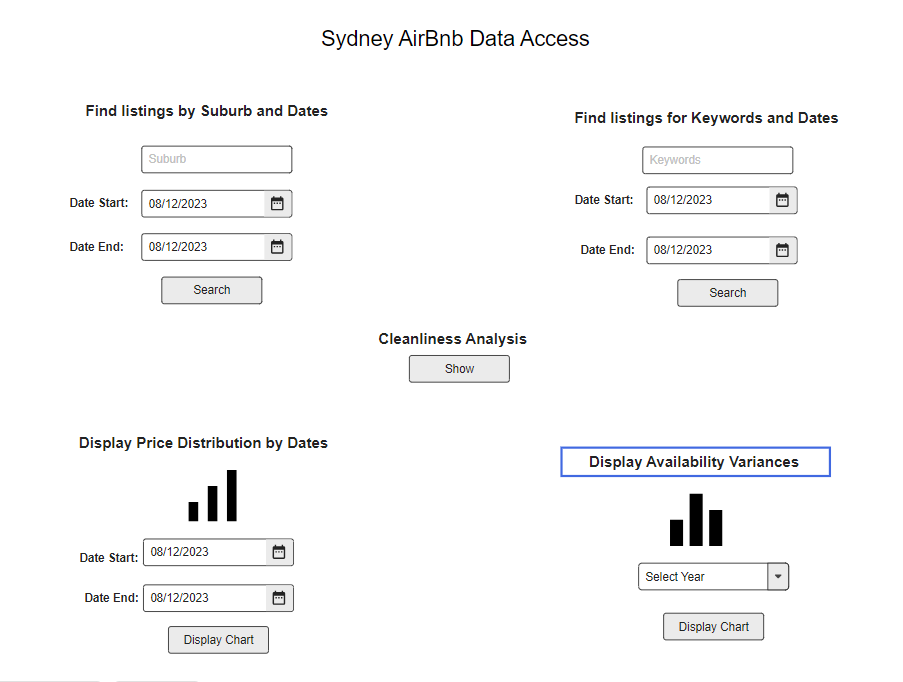


## Visual Design

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.

**This is a rough draft – no idea how we want to display the options yet – not like this – same link as heirachy -**

**Question do we want all options to appear ready to use on landing area?**



NEED:

* Choose colour palette
* Choose Fonts
* Screen design – wireframes + story bard?
* Discussion

Marking Guide  
Part A – Project Management (worth 25% of total course mark)  
There are 3 key components to this:  
1. Project Plan and Gantt Chart (40 marks)  
2. Software Design Document (50 marks)  
3. Software Version Control (10 marks)  
Marked out of 100, then times by 0.25 to get total mark. Unless otherwise specified, all  
group members should receive same mark.  
Project Plan and Gantt Chart (40 marks)  
In the Project Plan.docx  
1. Introduction (5 marks)  
Should contain an overview of the project (from a project management/component perspective)  
and mention the scope and outline of the project management document  
2. Work Breakdown Structure (WBS) (10 marks)  
Should be a breakdown of all the different activities involved in completing the project. For Part A  
this should contain all of the work involved including preparing the project plan and software  
design document, as well as all related preparatory/organisation work. For Part B this should include  
all of the required implementation, testing and reporting activities. This can be somewhat high level  
for Part B, but should still contain some reasonable assumptions. This should be presented as some  
form of diagram/hierarchy.  
3. Activity Definition & Estimation (10 marks)  
For each item in the WBS, the item should be explained in detail and include a time estimate that  
is reasonable.  
4. Gantt chart (15 marks)  
All of the items in the Activity definition should be listed in the Gantt chart with the relevant  
estimates and scheduling. The students should have also tracked the actual start time and time  
taken. Also need to submit the Gantt chart.xlsx

Software Design Document (50 marks)  
In the Software Design Document.docx  
1. System Vision (10 marks)  
Should include a background on the dataset, software overview and potential benefits of  
the software.  
2. Requirements (10 marks)  
There are 2 types of requirements to consider:  
• User Requirements: How a user will interact with the program. What do they need to do ?  
• Software Requirements: What functionality will the software provide (think functional  
requirements)  
• Use Cases: These Use Cases should show the blending of user and software requirements by  
identifying use cases and how the user will interact with the product. Any diagramming format  
is acceptable, but the diagrams should clearly display the sequence of events and interactions  
between the user and the software. Expecting about ~5 use cases (1 for each of the functions)  
and a few accompanying Use Case Diagrams.  
3. Software Components and Software Design (15 marks)  
• Software Design: Flow chart / block diagram (5 marks)  
• Software Components: Functions, Classes/Data Structures, Algorithms (10 marks)  
There should be a listing of the main functions (I would expect at least a loadData function and  
some kind of display function), the main classes/data structures used and potentially a description  
of any algorithms that might be used for data analysis. Each component should have the relevant  
information (name, type, details).  
4. User Interface Design (15 marks)  
Structural design should almost be like a flowchart/hierarchy chart showing the structure of  
the interface. What screens/components and how do they interact.  
Wireframes/mock ups of the interface. Should have clearly labelled interface components. No hand  
drawing (must be digital design). Different screens/menus/options should have their own  
wireframes. No colour/graphics required – just position/size of components and component layout.  
Additional design information can be included here too.  
Software Version Control (10 marks)  
Git\_log.txt : A Git repository is correctly used for all contributions to the project. A Git log is  
attached and shows regular commits and pushes from all group members.