Software Testing Report

<Project Name>

Georgia Platt

Jackson Scown

Gia Huy Lieu

Table of Contents

[1.0 Unit Tests 3](file:///D:\Learning\SofwareTec\Assignment2\Software%20Testing%20Report.docx#_Toc49779837)

[2.0 Coverage Report 4](file:///D:\Learning\SofwareTec\Assignment2\Software%20Testing%20Report.docx#_Toc49779838)

[3.0 Requirements Acceptance Testing 5](file:///D:\Learning\SofwareTec\Assignment2\Software%20Testing%20Report.docx#_Toc49779839)

# Unit Tests

| **No** | **Test Case** | **Expected Results** | **Actual Results** |
| --- | --- | --- | --- |
| **1.0** | **Perform search functions:** |  |  |
| 1.1 | Test with dates:  Make sure date is valid | Only date types are permitted as input | Only date types are permitted as input |
| 1.2 | Test search  The test case will setup the suburb and dates that users want to stay. After triggering event search, it will display all available places that was filled if the test case is true | Display all available places related to user inputs. | Display all available places related to user inputs. |
| **2.0** | **Test apply filter functions:**  This test happens when user do the perform search functions, inputting suburbs and dates. In this test function, ‘Sydney’ is set |  |  |
| 2.1 | Filter with minimum range and maximum range is empty:    This test will ensure that the functions will work or not if maximum range doesn’t have value. | Assert error messages box isn’t displayed | No error messages box was displayed |
| 2.2 | Filter with maximum range and minimum range is empty:  This test will ensure that the functions will work or not if minimum range doesn’t have value. | Assert error messages box isn’t displayed | No error messages box was displayed |
| 2.3 | Filter with minimum range and maximum range:  This test will ensure that the functions will work or not if maximum range doesn’t have value. | Display error message and exit | Display error message and exit |
| 2.4 | Test filter data with price data is greater or equal than maximum range | Display all listings that have price data is less or equal than maximum range | Display all listings that have price data is less or equal than maximum range |
| 2.5 | Test filter data with price data is greater or equal than minimum range | Display all listings that have price data is maximum or equal than maximum range | Display all listings that have price data is maximum or equal than maximum range |
| 2.6 | Testing the filtering of data within a specified price range, with a minimum and maximum value | Display all listings within specified range, with a minimum and maximum | Display all listings within specified range, with a minimum and maximum |
| 2.7 | Testing the filtering of data for specified room type | Display all listings within specified room type | Display all listings within specified room type |
| 2.8 | Testing the filtering of data for specified property type | Display all listings within specified property type | Display all listings within specified property type |
| 2.9 | Testing the filtering of data for specified price range, property type and room type | Display all listings with specified attributes | Display all listings with specified attributes |
| 2.10 | Testing the filtering of data for an invalid price range: In this test, the text 'abc' was entered into the minimum price input | Display the errors | Display the errors |
| **3.0** | **Test price distribution chart function** |  |  |
| 3.1 | Testing the function to read the file by verifying the presence of values in the 'suburb,' 'property type,' and 'room type' fields | Display the presence of value in the 'suburb,' 'property type,' and 'room type' fields | Display the presence of value in the 'suburb,' 'property type,' and 'room type' fields |
| 3.2 | Testing the 'filter\_change' when users select an option from the dropdown menu. The 'filter\_change' flag is used to ensure that the user's selection in the dropdown menu is detected. | ‘filter\_change’ is true when users select | ‘filter\_change’ is true when users select |
| 3.3 | Testing whether 'filter\_change' is set back to 'False' after the drawing functions | ‘filter\_change’ is true after the drawing functions | ‘filter\_change’ is true after the drawing functions |
| 4.0 | Test keyword search function |  |  |
| 4.1 | Testing whether the file was successfully loaded | Load file successfully | Load file successfully |
| 4.2 | Test search for cleanliness | Display the data related to cleanliness comments | Display the data related to cleanliness comments |
| 5.0 | **Test suburb rating function** |  |  |
| 5.1 | Testing the function to read the file by verifying the presence of values in the 'property type,' and 'room type' fields | Display the presence of value in the 'property type,' and 'room type' fields | Display the presence of value in the 'suburb,' 'property type,' and 'room type' fields |
| 5.2 | Test the average rating of all suburbs | Display the average rating of all suburbs | Display the average rating of all suburbs |
| 5.3 | Test the average price of all suburbs | Display the average price of all suburbs | Display the average price of all suburbs |

# Coverage Report

Function coverage:

Unit tests cover the behaviour of ‘perform\_search’, ‘applyFilters’, ‘priceDistSearch, ‘priceCustSearch’, ‘keywordSearch’ method, ensuring that it performs its intended task of searching and filtering data based on user inputs. This includes testing the entire functions ’logic.

Statement coverage:

Unit tests cover every statement within the listed method above. This means that each line of code in the method is executed at least once during the tests. These covers:

* Scenarios where users input both invalid price ranges, dates, and typical filter criteria.
* Scenarios where file cannot load or there is no file.
* Scenarios where users can do the keyword search and read about cleanliness comments

Branch coverage:

The unit tests validate both the true and false branches of conditional statements within the listed method above. This includes:

* Testing different conditions related to data filtering such as ‘price minimum’ or ‘price maximum’ field is not filled.
* Testing flag in ‘priceDistSearch’ and ‘priceCustSearch’ to ensure that users has set values for data.

Condition coverage:

Tests cover all possible combinations of conditions within single statements, including logical AND and OR conditions. This ensures that the method behaves correctly under various input conditions and that different combinations of filters are tested.

# Requirements Acceptance Testing

| **Software  Requirement No** | **Test** | **Implemented (Full /Partial/ None)** | **Test Results (Pass/ Fail)** | **Comments (for partial implementation or failed test results)** |
| --- | --- | --- | --- | --- |
| 1 | Users shall be able to input the name of a suburb in Sydney to narrow down data analysis to a specific area | Full | Pass |  |
| 2 | The dataset shall be updated monthly to ensure access to the latest data | None |  |  |
| 3 | Users shall have the ability to perform keyword searches (e.g., "pool" or "pet-friendly") to filter listings based on amenities and features | Full | Pass |  |
| 4 | Users shall have the option to filter listings based on various criteria, such as price range, number of bedrooms, or property type | Full | Pass |  |
| 5 | The system shall calculate the number of customers who commented on cleanliness. | None |  |  |
| 6 | The system shall provide a tool for users to analyse cleanliness-related keywords in customer reviews | Full | Pass |  |
| 7 | Users shall be able to generate a price distribution chart for the selected time, illustrating the distribution of property prices | Partial | Pass | Users shall be able to generate a price distribution from file csv. |