Software Testing Report

<Sydney Database Analysis>

Thien Thao My Bui – s5273753

Chelzie Castanares – s5259144

Table of Contents

[1.0 Unit Tests 3](#_Toc49779837)

[2.0 Coverage Report 4](#_Toc49779838)

[3.0 Requirements Acceptance Testing 5](#_Toc49779839)

# Unit Tests

|  |  |  |  |
| --- | --- | --- | --- |
| No | Test Case | Expected Results | Actual Results |
| 1.0 | Data Storage | Can handle any input CSV file | Error: Exceeds data limit of 350MB. |
| 1.1 | Data Loading and Preprocessing Time | Ensure data cleaning and preprocessing functions work properly | Long loading time  Some columns have mixtype |
| 1.2 | Data Format | DataFrame | DataFrame  Error: Some columns have mixtype value |
| 2.0 | Keyword Data Retrieve | Return all keyword-containing records | Return all keyword-containing records |
| 2.1 | Keyword Search Accuracy | Accurate with ability to ignore case-sensitive | Accurate with ability to ignore case-sensitive |
| 2.2 | Keyword Finding Accuracy | Return records with exact matching keyword | Error: Some simple word might be mistaken with other words containing the exact prefix |
| 2.3 | No Keyword found in the dataset | Display error message | Display error message |
| 2.4 | Accepting multiple keywords | Return all records containing at least one of the keywords | Error: Return records containing all the keywords |
| 3.0 | Default Criteria Data Retrieve | Return all records related to the set-up criteria | Return all records related to the set-up criteria |
| 3.1 | Accepting multiple keywords in the default criteria | Return all records containing at least one of the keywords form the set-up criteria | Return all records containing at least one of the keywords form the set-up criteria |
| 4.0 | Date Ranges Input | Return all available records in the specified period | Return all available records in the specified period |
| 4.1 | Missing Date Range Input | Display error message | Display error message |
| 5.0 | Suburb Selection | Return records from all of the selected options | Return all records from all of the selected options |
| 5.1 | No suburb option input | Return all the records | Return all the records |
| 6.0 | Data Visualization | Accurate figure and icon object based on the value of the data | Accurate figure and icon based on the value of the data |

\*Fixes for Test Case 1:

1.0. By optimize and config Streamlit library that can have an upload data limit of 2GB.

1.1. Using session\_state to store the previous loaded dataframe, helping the system to run and process faster

1.2. Adding dtype for the specified warning columns

# Coverage Report

As our Python files were built mainly through script code, not defining specific functions or classes, the analysis report will focus specifically at the statement level instead.

* Homepage.py: There are 15 out of 16 statements (lines of code) that were executed during the test, resulting in 89% coverage

A black background with white text

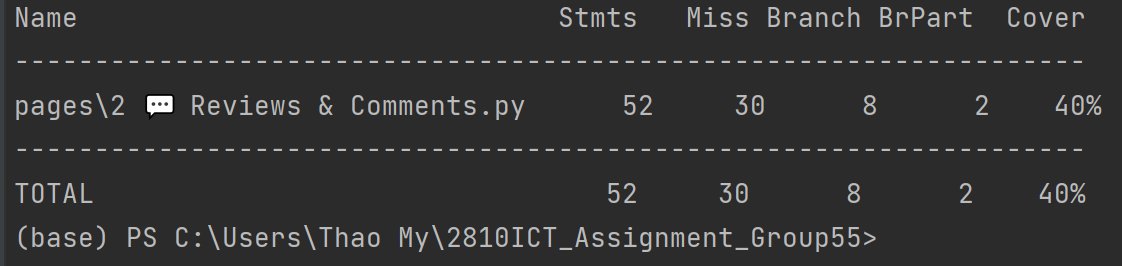
Description automatically generated

* Property Details.py: Out of 47 statements (lines of code), 56% of them were executed during the test.

A screenshot of a computer

Description automatically generated

* Reviews & Comments.py: Only 40% of 52 statements (lines of code) that were executed during the test.



The difference between the Coverage Percentage in each python script file is due to a heavy number of statements (lines of code) lying in between different branches in the python files. Branches typically occur within conditional statements (“if-else” blocks). These coverage percentages were calculated based on the statements within a partial of branch parts being executed. As a result, some other opposite branches within conditional statements were not executed, leading to all statements residing in these branches not being fully tested.

Therefore, for further implementation of the project, it's recommended to structure the statements into functions or classes to facilitate more granular and meaningful testing, plus addressing missed branches to handle different scenarios and conditions correctly.

# Requirements Acceptance Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Software Requirement No | Test | Implemented (Full /Partial/ None) | Test Results (Pass/ Fail) | Comments (for partial implementation or failed test results) |
| Property Search | The users are able to input the location of stay within their selected period date. | Full | Pass |  |
| Property Price Search | The users are able to see the cost of all properties within their selected period date. | Full | Pass |  |
| Keyword Search System | The user is able to search for properties containing the specific keyword/s they are looking for. | Full | Pass | Cannot work accurately for multiple keywords |
| Comments Search System | The users are able to view all comments of selected property related to their keywords/criteria they want to explore. | Full | Pass |  |
| Review Search System | The users are able to look at the review scores of a certain property. | Full | Pass |  |
| Usability | User interface should be easy to use and navigate. | Full | Pass |  |
| All requirements should be working smoothly with no errors. | Full | Pass |  |
| Reliability | System should be running at all times. | Partial | Fail | System does not run at all times but runs when activated |
| System should be error free. | Full | Pass |  |
| Performance | System response time should be quick when retrieving data e.g., 5 secs. | Partial | Fail | System response time is a bit slow, taking about 30sec – 1 minute to boot or get results but system still works. |
| Should be able to store large amounts of data. | Full | Pass |  |
| Security | Website should be encrypted with HTTPS Protocol. | Full | Pass |  |