Software Testing Report

SYD Airbnb Application

Viet My Tran Le s5235877

William Luvant s5287580

Liangxian Zhao s2869753

Table of Contents

[1.0 Unit Tests 3](#_Toc49779837)

[2.0 Coverage Report 4](#_Toc49779838)

[3.0 Requirements Acceptance Testing 5](#_Toc49779839)

# Unit Tests (Below Unit test are my hypothesis, feel free to make any change. I asked lecturer, we don’t need to test GUI window)

**Total tests =？**

| **No** | **Test Case** | **Expected Results** | **Actual Results** |
| --- | --- | --- | --- |
| **1.0** | **Create Table Function** |  |  |
| 1.1 | Test without existing SQL database | Dataset Completed! | Dataset Completed! |
| 1.2 | Test with existing SQL database | Exception handled: pass and print "Data Exists!" | Exception handled: pass and print "Data Exists!" |
| **2.0** | **Insert Table Function** |  |  |
| 2.1 | Test without existing SQL database | Exception handled: pass and print "Not Exist!" | Exception handled: pass and print "Not Exist!" |
| 2.2 | Test with already inserted SQL database | Exception handled: pass and print "Data Already Exists!" | Exception handled: pass and print "Data Already Exists!" |
| 2.3 | Test with existing uninserted SQL database | Data Inserted! | Data Inserted! |
| **3.0** | **Display Main Window Function** |  |  |
| 3.1 | Test previous window opened | Previous window closed and show main window | Previous Window Close = Close  Main Window Show = Show |
| 3.2 | Test no previous window opened | Pass closing previous window and show main window | Previous Window Close = None  Main Window Show = Show |
| **4.0** | **Details by Time range: Initial Table Data Fetch Function** |  |  |
| 4.1 | Test initial table data fetch | Data is fetched (length of data > 0) |  |
| **5.0** | **Inspection Details (Time), Property price Function** | *\*print is for test purposes, in program print is wx.MessageBox* |  |
| 5.1 | Test with correct date format | Data is fetched (length of data > 0) |  |
| 5.2 | Test with wrong date format | Exception handled: pass and print "Not Found: Please enter right date format" |  |
| 5.3 | Test with letters | Exception handled: pass and print "Not Found: Please enter value within date range / incorrect date" | Exception handled: pass and print "Not Found: Please enter value within date range / incorrect date" |
| 5.4 | Test with out-of-range date | Exception handled: pass and print "Not Found: Please enter value within date range / incorrect date" | Exception handled: pass and print "Not Found: Please enter value within date range / incorrect date" |
| **6.0** | **Keyword Search Function** |  |  |
| 6.1 | Test with correct date format | Data is fetched (length of data > 0) |  |
| 6.2 | Test with wrong date format | Exception handled: pass and print " Not Found: Please enter a keyword / right date format" | Exception handled: pass and print " Not Found: Please enter a keyword / right date format" |
| 6.3 | Test with letters | Exception handled: pass and print "Not Found: Keyword is not found in date range / Date out of range / Incorrect Date" | Exception handled: pass and print "Not Found: Keyword is not found in date range / Date out of range / Incorrect Date" |
| 6.4 | Test with out-of-range date | Exception handled: pass and print "Not Found: Keyword is not found in date range / Date out of range / Incorrect Date" | Exception handled: pass and print "Not Found: Keyword is not found in date range / Date out of range / Incorrect Date" |
| 6.5 | Test without keyword | Exception handled: pass and print "Not Found: Please enter a keyword / right date format" | Exception handled: pass and print "Not Found: Please enter a keyword / right date format" |
| 6.6 | Test with keyword not in date range | Exception handled: pass and print "Not Found: Keyword is not found in date range / Date out of range / Incorrect Date" | Exception handled: pass and print "Not Found: Keyword is not found in date range / Date out of range / Incorrect Date" |
| **7.0** | **Score-rating Function** |  |  |
| 7.1 | Test with existing selection | Data is fetched (length of data > 0) |  |
| *\* Tests don't include the GUI window creation, but just simple closing previous window and opening the selected requirement window.* | | | |
| **8.0** | **Display Individual Feature Window Function** |  |  |
| 8.1 | Test display window | Window displayed | Previous Window Close = Close  Main Window Show = Show |

# Coverage Report (hypothesis)

Below is coverage report generated through coverage.py. All functions/units tested are nestled in one python file (units.py), these are the 'data processing functions' pulled out from the program code. This report covers the coverage of test\_units.py (file using python's unittest to test each unit) with function test cases taken from units.py. All tests passed.

Main page of coverage report:

Coverage report for units.py:

Coverage for test\_units.py:

# Requirements Acceptance Testing (hypothesis)

| **Software  Requirement No** | **Test** | **Implemented (Full /Partial/ None)** | **Test Results (Pass/ Fail)** | **Comments (for partial implementation or failed test results)** |
| --- | --- | --- | --- | --- |
| 1 | Display 5 buttons function |  |  |  |
| 2 | Display price field to receive user input |  |  |  |
| 3 | Display start and end dates to choose from |  |  | **Replaced with text input, date range information and input validation.** |
| 4 | Display earliest to latest date range |  |  |  |
| 5 | Display list of keywords to choose from |  |  |  |
| 6 | Display list of score rating to choose from |  |  |  |
| 7 | Give option to reset the function (input field) |  |  | For practicality and to reduce redundancy, reset the input field by typing. |
| 8 | Display appropriate table/graph respective of requirement |  |  |  |
| 9 | Give option to continue to another session |  |  |  |
| 10 | Give option to exit the program |  |  | For practicality and to reduce redundancy, exit is through the window button on top left corner. |