Software Testing Report

Sydney Airbnb Data Analysis Tool

s5291506 – Jamil Deris  
s5287914 – Tanish Dhir  
s5295636 – Arjan Dangol

Table of Contents

[1.0 Unit Tests 3](#_Toc49779837)

[2.0 Coverage Report 4](#_Toc49779838)

[3.0 Requirements Acceptance Testing 5](#_Toc49779839)

# Unit Tests

This specific format of the table enables you to make a comparison, between the real outcomes of every test case. This assists the developer in spotting any inconsistencies or problems within their code. If required the developer can expand this table by incorporating test cases that are relevant, to their project and provide supplementary information like the date when the tests were performed the name of the tester and any remarks or comments regarding the test results.

(In this table you fill out details about what unit tests you have done using the unittest module)

| **No** | **Test Case** | **Expected Results** | **Actual Results** |
| --- | --- | --- | --- |
| **1.0** | **WordCount Functions** |  |  |
| 1.1 | Test with a valid input file | Count of words in the file matches the expected count |  |
| 1.2 | Test with a non-existent file | Exception Handled |  |
| 1.3 | Test with an empty input file | Display error message and exit |  |
| 1.4 | Test with a file containing special characters | Count of words in the file matches the expected count |  |
| 1.5 | Test with a large input file | Count of words in the file matches the expected count |  |
| **2.0** | **Histogram Functions** |  |  |
| 2.1 | Test with a valid input dictionary | Generate a histogram chart that matches the expected histogram |  |
| 2.2 | Test with an empty input dictionary | Display error message and exit |  |
| 2.3 | Test with a dictionary containing negative values | Display error message and exit |  |
| 2.4 | Test with a dictionary containing non-integer values | Display error message and exit |  |
| 2.5 | Test with a dictionary containing duplicate keys | Generate a histogram chart that includes all unique keys |  |

# Coverage Report

A description of the coverage of your unit tests, including how you evaluated coverage (function, statement, branch, condition)

# Requirements Acceptance Testing

(You will need to fill out the column on the left with the requirements listed in software design documents and the columns on the right with the results of your own testing)

| **Software  Requirement No** | **Test** | **Implemented (Full /Partial/ None)** | **Test Results (Pass/ Fail)** | **Comments (for partial implementation or failed test results)** |
| --- | --- | --- | --- | --- |
| 1 | The program shall accept user input through a list of suburbs |  |  |  |
| 2 | The program shall show what rooms are available and show brief description about them |  |  |  |
| 3 | It shall have a date option to see for a particular date |  |  |  |
| 4 | It shall present a price for each room including details of payment |  |  |  |
| 5 | 5 It shall display number of customers who gave feedback chart of a room on basis of cleanliness and similar keywords like environment, tidy, etc. These will have values which indicate what is the review of the room. These are selected as tourists will look for these features in a room as it makes their visit a pleasant experience. |  |  |  |
| 6 | The program shall present a price distribution chart when user selects a suburb and click on view price distribution chart |  |  |  |
| 7 | The program shall present how many times a property has been used for a user selected date |  |  |  |
| 8 | Directory/level names must start with an alphabetical character to be considered valid |  |  |  |
| 9 | The program should be able to accept as many levels for each file name as the user wants to input. This is limited only by the number of levels allowed in Windows (approximately 120) |  |  |  |