Software Testing Report

<Project Name>

Student Names

Table of Contents

[1.0 Unit Tests 3](#_Toc49779837)

[2.0 Coverage Report 4](#_Toc49779838)

[3.0 Requirements Acceptance Testing 5](#_Toc49779839)

# Unit Tests

(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 a wrong filename | Display an Error Message | Exception Handled |
| 1.2 | Test empty input file | Display an Error Message | Exception Handled |
| **2.0** | **Histogram Functions** |  |  |
| 2.1 | Empty input dictionary | Display an Error Message | Display an Error Message |

# Coverage Report

The unit tests comprehensively cover various aspects, ensuring the successful passage of all tests. Evaluation of coverage includes assessing functions, statements, branches, and conditions to guarantee a robust and error-resistant system. Notably, the tests incorporate exception handling mechanisms both at the database level and during query execution, as well as for potential errors related to incorrect file inputs. This comprehensive approach to testing aims to validate the functionality, accuracy, and reliability of the system, minimizing the likelihood of issues arising in real-world scenarios.

# Requirements Acceptance Testing

| **Software  Requirement No** | **Test** | **Implemented (Full /Partial/ None)** | **Test Results (Pass/ Fail)** | **Comments (for partial implementation or failed test results)** |
| --- | --- | --- | --- | --- |
| 1 | Accept multiple file names as arguments from the command line |  |  | Not Applicable |
| 2 | Display the details of all valid files | Full | Pass |  |
| 3 | Display an appropriate message if a file does not exist or if a file name is invalid | Full | Pass |  |
| 4 | Display a message if an argument is a directory instead of a file | Full | Pass |  |
| 5 | File name can be a simple file name or include the full path of the file with one or more levels | Full | Pass | File should be in the same folder, because of secret key access by streamlit |
| 6 | file names must start with an alphabetical character | Full | Pass |  |
| 7 | Valid file name extensions must be 3 or 4 alphabetical characters preceded by a dot) | Full | Pass |  |
| 8 | Directory/level names must start with an alphabetical character to be considered valid | Full | Pass |  |
| 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) |  |  | Not Applicable |