Software Testing Report

<Analysis tools of Victoria State Accident >

S5288771 Chi Pang Cheng

S5323472 Kwangseok Choi

Table of Contents

[1.0 Unit Tests 3](#_Toc49779837)

[2.0 Coverage Report 4](#_Toc49779838)

[3.0 Requirements Acceptance Testing 5](#_Toc49779839)

# Unit Tests

The table below shows the results of testing the code used to develop Victoria's accident analysis software. A total of 12 functions were tested using pytest.

| **No** | **Test Case** | **Expected Results** | **Actual Results** |
| --- | --- | --- | --- |
| 1.0 | LoadCSV |  |  |
| 1.1 | Set first load | Fetching first load | Fetching first load |
| 1.2 | Load data frame, grid rows/columns | Data frame loaded, grid contains rows and columns | Data frame loaded, grid contains rows and columns |
| 2.0 | Add Controls For Column |  |  |
| 2.1 | Adding controls for column should create checkbox and text control for each column name. | Checkbox and text match the number of column | Checkbox and text match the number of column |
| 3.0 | On Checkbox Change |  |  |
| 3.1 | Change the state of checkbox | “Checked\_checkboxes” count will be incremented by one | “Checked\_checkboxes” count will be incremented by one |
| 4.0 | SearchCSV |  |  |
| 4.1 | Searching with all checked checkbox | Selected grid displayed | Selected grid displayed |
| 5.0 | Reset |  |  |
| 5.1 | Resetting checkboxes | “checked checkboxes” set to zero | “checked checkboxes” set to zero |
| 6.0 | Calculate Statistics |  |  |
| 6.1 | Calculation statistics with checked checkboxes. | Get statistics of checked items | Get statistics of checked items |
| 7.0 | Get Unique Column Values |  |  |
| 7.1 | Get unique column values | Get unique column values and return dict | Get unique column values and return dict |
| 8.0 | Get Column Value Count |  |  |
| 8.1 | Counting specific values from column | return Integer count | return Integer count |
| 9.0 | Generate Chart |  |  |
| 9.1 | Generate chart with invalid header | Get assertion error and re-select the axis | Get assertion error and re-select the axis |
| 9.2 | Generate chart with valid Y-axis | Generate chart with selected axis | Generate chart with selected axis |
| 10.0 | Update Gauge |  |  |
| 10.1 | Update gauge, set value to specified value | Gauge control's value set to the specified value. | Gauge control's value set to the specified value. |
| 11.0 | Log\_action |  |  |
| 11.1 | Add action text into log | Action text added to log | Action text added to log |
| 12.0 | Update Statistics |  |  |
| 12.1 | Statistics will be available after first load | After first load, statistics will be displayed | After first load, statistics will be displayed |
| 12.2 | Update stats with no checkboxes, display specific message. | “Please check at least one checkbox for statistics analysis.” This message will be displayed | “Please check at least one checkbox for statistics analysis.” This message will be displayed |
| 12.3 | Update stats with more than five checkboxes, display specific message. | “Please check no more than five checkboxes for statistics analysis. This message will be displayed | “Please check no more than five checkboxes for statistics analysis. This message will be displayed |
| 12.4 | Valid statistics calculation | Statistics will be available | Statistics will be available |

# Coverage Report

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

The photo below shows the results of a coverage test of Victoria's accident analysis software. There are a total of 119 statements, and you can see that all statements are working. Also, every function in the program is called at least once. In the if statement, both true case and false case were tested.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

# 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 | Accept multiple file names as arguments from the command line | Partial | Fail | Software accept only one CSV file at once. But you can re-submit file. |
| 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 | None | Fail | Users are not allowed to select folder. They can select only CSV file. |
| 5 | File name can be a simple file name or include the full path of the file with one or more levels | None | Fail | Users select file through file open dialog, so no need to enter file name. |
| 6 | file names must start with an alphabetical character | None | Fail | File names can start with number. |
| 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 | Full |  |
| 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) | Full | Pass |  |