Software Testing Report

Victorian Accident Analysis Project

Eric Martin, Harris Huntington

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** | **Import functions** |  |  |
| 1.1 | Test a wrong filename | Display error message and exit |  |
| 1.2 | Test an empty/invalid file | Display error message and exit | Display error message and exit |
| **2.0** | **Data functions** |  |  |
| 2.1 | Check valid date | Display error message and exit |  |
| 2.2 | Test infoByTime loads data | Display error message and exit |  |
| 2.3 | Test accidentByHour loads data | Display error message and exit |  |
| 2.4 | Test keywordByTime loads data | Display error message and exit |  |
| 2.5 | Test weekdayAnalysis loads data | Display error message and exit |  |
| 2.6 | Test alcoholType loads data | Display error message and exit |  |
| 2.7 | Test valid keyword | Display error message and exit |  |
| **3.0** | **Display Functions** |  |  |
| 3.1 | Test infoByTime displays info | Display error message and exit |  |
| 3.2 | Test accidentByHour displays info | Display error message and exit |  |
| 3.3 | Test keywordByTime displays info | Display error message and exit |  |
| 3.4 | Test weekdayAnalysis displays info | Display error message and exit |  |
| 3.5 | Test alcoholType displays info | Display error message and exit |  |

# 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 | Test a wrong filename (1.1) | Partial | Fail | Filename could not be accessed |
| 1 | Test an empty/invalid file (1.2) | Full | Pass |  |
| 2 | Check valid date (2.1) | Partial | Fail | Date didn’t work |
| 3 | Tests 3.1-3.4 | Partial | Fail | Could not detect displaying |
| 4 | Test valid keyword (2.7) | Partial | Fail | Could not detect keyword |
| 5 | Test keywordByTime displays info (3.3) | Partial | Fail | Could not detect displaying |
| 6 | Test keywordByTime loads data (2.4) | Partial | Fail | Could not detect loading |
| 7 | Tests 2.2 – 2.6 | Partial | Fail | Could not detect Loading |
| 8 | Tests 2.6/3.5 | Partial | Fail | Could not detect loading/displaying |
| 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) |  |  |  |