Software Testing Report

VicCrashStats App

Taylor Edgerton

Davinder Grewal

Table of Contents

[1.0 Unit Tests 3](#_Toc116409248)

[2.0 Coverage Report 5](#_Toc116409249)

[3.0 Requirements Acceptance Testing 6](#_Toc116409250)

[Software Design Document Software Requirements 6](#_Toc116409251)

[Software Requirement Acceptance table 7](#_Toc116409252)

# Unit Tests

| **No** | **Test Case** | **Expected Results** | **Actual Results** |
| --- | --- | --- | --- |
| **1.0** | **GetData Class Methods** |  |  |
| 1.1 | accidentAlcohol method connects to database and returns data | Returns Tuple data type | Returns Tuple data type |
| 1.2 | accidentAlchohol method returns correct data for barchart | Returns Data Length of 2 | Returns Data Length of 2 |
| 1.3 | getAccidentByKeyword method connects to database, accepts parsed string and returns data | Returns list data type | Returns a list data type |
| 1.4 | getAccidentsByDate method connects to database accepts parsed string and returns data | Returns list data type | Returns list data type |
| **2.0** | **Reading CSV File and creating Database** |  |  |
| 2.1 | Pandas reads CSV | Variable containing CSV is not none | Is Not None |
| 2.2 | Pandas Database Initialises | Variable containing CSV converted to SQL is not none | Is Not None |
| 2.3 | Pandas Database returns Dataframe | Variable returns type pandas.core.frame.Dataframe | Returns pandas.core.frame.Dataframe |
| 2.4 | Sqlite accepts a query and returns data | Variable containing cursor.execute is not none | Variable containing cursor.execute is not none |
| 2.5 | Sqlite Returns Data | Query result is returned as a List | Query result returns as a list |
| **3.0** | MatPlotLib Bar chart Creation |  |  |
| 3.1 | MatPlotLib Plot is created | Variable containing plot is not none | Variable containing plot is not none |
| 3.2 | MatPlotLib plot is a bar chart type | Variable containing barchart is of type plt.matplotlib.container.BarContainer | Variable containing barchart is of type plt.matplotlib.container.BarContainer |

# Coverage Report

All backend SQL and Database connection modules and functions used by the program are tested to ensure that all the data returned into the program are the correct data types are not none. This is required as it is difficult to check a specific SQL queries result as it will return different results based on the database used, although sqlite module returns a list data type. SQL and Database functions were tested for returning a value after connection and returning a list datatype after a query.

Matplotlib plot data was inserted into the testing class to ensure a connection to matplotlib library and the creation of a bar chart. Wxpython is difficult to test and time consuming as the app will initialise in order to read the class data, although functions sending and retrieving data to the application are tested.

Data that is parsed from the wxpython app into the GetData class for use in sql queries is tested within the GetData class for the correct list or tuple data types and length to ensure that the data being received by wxpython will be displayed correctly by the app.

# Requirements Acceptance Testing

## Software Design Document Software Requirements

|  |  |
| --- | --- |
| Software Requirements | |
| ID | Description |
| R1 | The program shall accept a database from a CSV file to perform functions on the data and create an output |
| R2 | The program will be programmed in Python |
| R3 | The data used will be a CSV file that will be inserted into the program by a programmer |
| R4 | The program will apply the CSV file as an SQLite database. |
| R5 | Each input will apply an SQL query and the output will be based on that Query |
| R6 | The program will use SQL queries to access data from the database |
| R7 | The program will use libraries to create data visualisation elements |
| R8 | The program will use a python GUI library wxPython |

## Software Requirement Acceptance table

| **Software  Requirement No** | **Test** | **Implemented (Full /Partial/ None)** | **Test Results (Pass/ Fail)** | **Comments (for partial implementation or failed test results)** |
| --- | --- | --- | --- | --- |
| R1 | The program shall accept a database from a CSV file to perform functions on the data and create an output | Partial | Fail | 3 of 5 options were successfully completed |
| R2 | The program will be programmed in Python | Full | Pass |  |
| R3 | The data used will be a CSV file that will be inserted into the program by a programmer | Full | Pass | Pandas is used to read the csv file |
| R4 | The program will apply the CSV file as an SQLite database. | Full | Pass | Pandas is used to create a .db file from the CSV |
| R5 | Each input will apply an SQL query and the output will be based on that Query | Partial | Pass | Inputs are formatted into the correct SQL query and Sqlite3 is used to query the .db file |
| R6 | The program will use SQL queries to access data from the database | Full | Pass | Sqlite3 allows SQL queries to return data from the database file |
| R7 | The program will use libraries to create data visualisation elements | Partial | Pass | A Basic MatPlotLib Bar chart is created, although using less data than expected |
| R8 | The program will use a python GUI library wxPython | Partial | Pass | wxPython was successfully used, although many visual elements described in the software design document are missing or incomplete. |