**Test Plan**

Types of testing that could be used:

Unit tests (functional)

System test (functional)

User acceptance testing [use case or state transition based] (functional and non-functional)

Exploratory testing (functional and non-functional)

**Unit Tests**

|  |  |  |  |
| --- | --- | --- | --- |
| **GIVEN** | **WHEN** | **THEN** | **RESULT** |
| * I have created a test object with shortString | * I call the inputAsStringArray method | * The result should match the ArrayList in the unit test | Pass |
| * I have created a test object with shortString | * I call the longestWordForDisplay method | * The result should be “The longest word is STRING at 6 letters long.” | Pass |
| * I have created a test object with shortString | * I call the longestWord method | * The result should match the ArrayList in the unit test | Pass |
| * I have defined a non-existent file location | * I call the convertTextFile method and pass in the file location | * The String output should be empty | Pass |
| * I have defined a file location that exists | * I call the convertTextFile method and pass in the file location | * The String output should be the same as longString | Pass |
| * I have created a test object with longString | * I call the countCharacters method and pass in inputAsArray and character ‘W’ | * The result should be 14 | Pass |
| * I have created a test object with shortString | * I call the countCharacters method and pass in inputAsArray and character ‘B’ | * The result should be 0 | Pass |
| * I have created a test object with longString | * I call the countCharacters method and pass in inputAsArray and character ‘,’ | * The result should be 9 | Pass |
| * I have created a test object with shortString | * I call the countCharacters method and pass in inputAsArray and character ‘-’ | * The result should be 0 | Pass |
| * I have created a test object with longString | * I call the countCharacters method and pass in inputAsArray and character ‘7’ | * The result should be 2 | Pass |
| * I have created a test object with shortString | * I call the countCharacters method and pass in inputAsArray and character ‘6’ | * The result should be 3 | Pass |
| * I have created a test object with an empty string | * I call the countCharacters method and pass in inputAsArray and character ‘”’ | * The result should be 0 | Pass |
| * I have created a test object with shortString | * I call the characterFrequency method | * The result should match the array in the unit test | Pass |
| * I have created a test object with longString | * I call the characterFrequency method | * The result should match the array in the unit test | Pass |
| * I have created a test object with an empty string | * I call the characterFrequency method | * The result should be an array of zeros | Pass |
| * I have created a test object with shortString | * I call the numberOfSpaces method | * The result should be 10 | Pass |
| * I have created a test object with longString | * I call the numberOfSpaces method | * The result should be 110 | Pass |
| * I have created a test object with an empty string | * I call the numberOfSpaces method | * The result should be 0 | Pass |
| * I have created a test object with shortString | * I call the numberOfWords method | * The result should be 11 | Pass |
| * I have created a test object with longString | * I call the numberOfWords method | * The result should be 111 | Pass |
| * I have created a test object with an empty string | * I call the numberOfWords method | * The result should be 0 | Pass |
| * I have created a test object with shortString | * I call the relativeFrequency method | * The result should match the array in the unit test | Pass |
| * I have created a test object with longString | * I call the relativeFrequency method | * The result should match the array in the unit test | Pass |
| * I have created a test object with an empty string | * I call the relativeFrequency method | * The result should be an array of zeros | Pass |
| * I have created a test object with shortString | * I call the highestCount method | * The result should be 7 | Pass |
| * I have created a test object with longString | * I call the highestCount method | * The result should be 56 | Pass |
| * I have created a test object with an empty string | * I call the highestCount method | * The result should be 0 | Pass |
| * I have created a test object with shortString | * I call the charsIncludingSpaces method | * The result should be 59 | Pass |
| * I have created a test object with longString | * I call the charsIncludingSpaces method | * The result should be 585 | Pass |
| * I have created a test object with an empty string | * I call the charsIncludingSpaces method | * The result should be 0 | Pass |
| * I have created a test object with shortString | * I call the charsExcludingSpaces method | * The result should be 49 | Pass |
| * I have created a test object with longString | * I call the charsExcludingSpaces method | * The result should be 475 | Pass |
| * I have created a test object with an empty string | * I call the charsExcludingSpaces method | * The result should be 0 | Pass |
| * I have created a test object with varietyString | * I call the numberOfRecognisedCharacters method | * The result should be 71 | Pass |
| * I have created a test object with an empty string | * I call the numberOfRecognisedCharacters method | * The result should be 0 | Pass |
| * I have created a test object with varietyString | * I call the numberOfUnrecognisedCharacters method | * The result should be 22 | Pass |
| * I have created a test object with an empty string | * I call the numberOfUnrecognisedCharacters method | * The result should be 0 | Pass |
| * I have created a test object with shortString | * I call the wordLengths method | * The result should match the array in the unit test | Pass |
| * I have created a test object with longString | * I call the wordLengths method | * The result should match the array in the unit test | Pass |
| * I have created a test object with shortString | * I call the wordLengthFrequencies method | * The result should match the array in the unit test | Pass |
| * I have created a test object with longString | * I call the wordLengthFrequencies method | * The result should match the array in the unit test | Pass |

**System test**

Create a test object with shortString

Test each get method within TextAnalysis class

\*\*getBasicAnalysisForDisplay and getWordLengthFrequenciesDisplay have not been tested, they will be tested manually\*\*

**User acceptance test (primarily testing the user interface)**

\*\*Incorporated into this script is a non-functional performance test importing a 5000 word text file to confirm larger files can be handled without any impact on performance\*\*

|  |  |  |
| --- | --- | --- |
| **Step instructions** | **Expected behavior** | **Comments** |
| Run program | Program starts and offers 3 options – 1 Full mode, 2 Demo mode, 0 Exit the program |  |
| Select option 1 – Full mode | Interface offers 3 options – 1 Use the keyboard, 2 Import a .txt file, 0 Return to the main menu |  |
| Select option 1 – Use the keyboard | Interface asks user to enter text |  |
| Enter text of your choice (try and include a variety of character types) | Interface allows user to enter text. After pressing enter it should display the text entered and offer 7 options |  |
| Select option 1 – Show all analysis | Interface should display as per fig. 1 |  |
| Select option 2 – Show basic analysis | Interface should display as per fig. 2 |  |
| Select option 0 – Return to the main menu | Interface should offer 3 options - 1 Full mode, 2 Demo mode, 0 Exit the program |  |
| Select option 1 – Full mode | Interface offers 3 options – 1 Use the keyboard, 2 Import a .txt file, 0 Return to the main menu |  |
| Select option 2 – Import a .txt file | Interface should ask the user to enter the file location and give example |  |
| Enter “..\TestTextFiles\test.txt” | Interface should allow user to enter location. After pressing enter it should display the text entered and offer 7 options |  |
| Select option 3 – Show character frequency table | Interface should display as per fig. 3 |  |
| Select option 0 – Return to the main menu | Interface should offer 3 options - 1 Full mode, 2 Demo mode, 0 Exit the program |  |
| Select option 2 – Demo mode | Interface offers 3 options – 1 A short piece of text, 2 I A long piece of text imported from a .txt file, 0 Return to the main menu |  |
| Select option 1 – A short piece of text | It should display the text entered and offer 7 options |  |
| Select option 4 – Show character frequency bar chart | Interface should display as per fig. 4 |  |
| Select option 0 – Return to the main menu | Interface should offer 3 options - 1 Full mode, 2 Demo mode, 0 Exit the program |  |
| Select option 2 – Demo mode | Interface offers 3 options – 1 A short piece of text, 2 I A long piece of text imported from a .txt file, 0 Return to the main menu |  |
| Select option 2 – A long piece of text imported from a .txt file | It should display the text entered and offer 7 options |  |
| Select option 5 – Show all word lengths | Interface should display as per fig. 5 |  |
| Select option 6 – Save basic analysis to a .txt file | Interface should ask the user to enter the name of the file |  |
| Enter “usertest” | Interface allows user to enter text. After pressing enter it should confirm the location of the file. The file should be visible in the TestTextFiles folder within the program |  |
| Select option 0 – Return to the main menu | Interface should offer 3 options - 1 Full mode, 2 Demo mode, 0 Exit the program |  |
| Select option 0 – Exit the program | Interface should display a goodbye message and terminate |  |

Fig. 1-5 below

Fig. 1

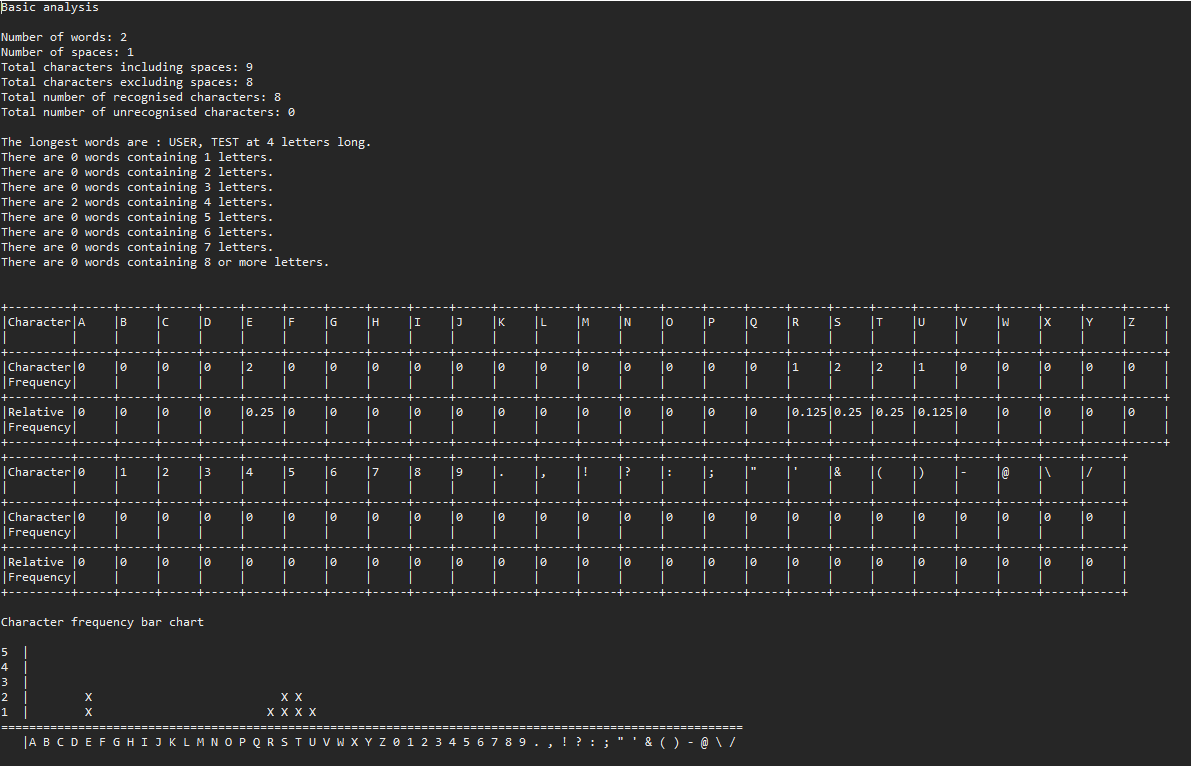
 

Fig. 2



Fig. 3

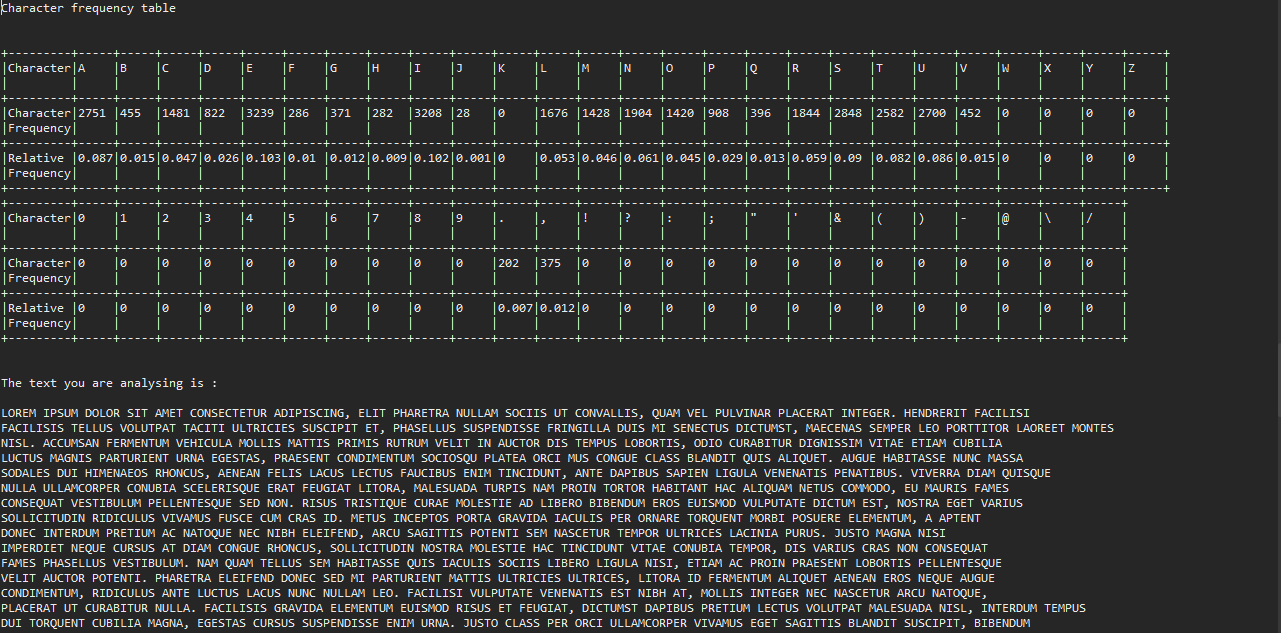


Fig. 4

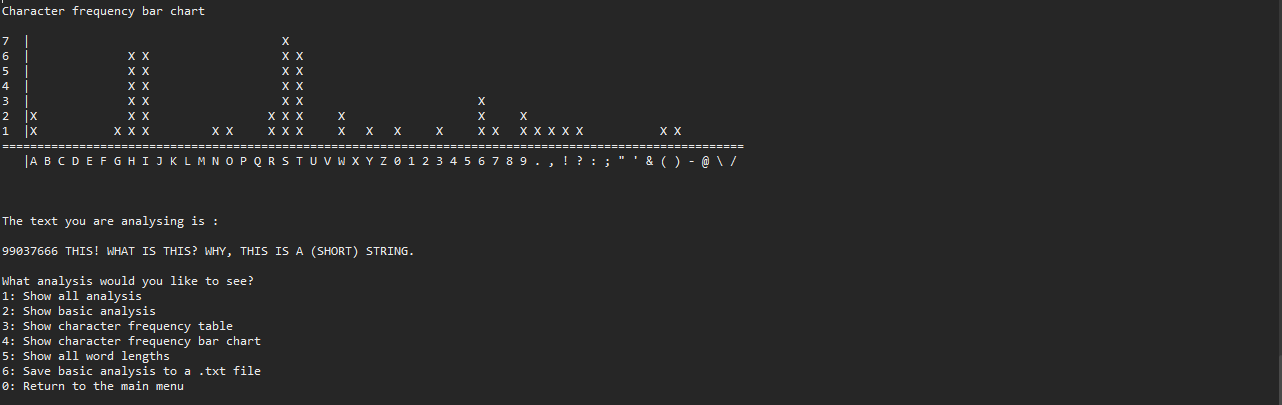


Fig. 5

