WORD COUNTER

Godwill Afolabi

Department of Information Technology

Augsburg University

February 2023

SUMMARY AND PURPOSE

1.0 Summary and Purpose

1.1 Summary and Purpose

Summary and Purpose:

The word counter program is a GUI application which can open up a text document, read in and count all the occurrences of each word, and write the counts into a separate text file.

The word counter program helps to accurately count the number of words in a document and the occurrences of every word and its functionalities are tested to ensure the correct behavior.

1.1 System Requirements

- Java Development Kit (JDK) version 8 or higher
- Text Document in plain text format
- JUnit5 standalone-console jar

2.0 SYSTEM FEATURES

- Presents a GUI
- Allows user to select a file for word count
- Reading in a text document
- Removes punctuations, hyphens, and other unwanted characters
- Counts the occurrence of every word
- Writes the counts into a separate output text file
- Provides a way to count the total number of words in the document
- Tests all the methods in the program

2.0 Requirements

2.1 Functional Requirements

- The program must present the user with a GUI with an option to select and submit a file for the word count
- The program must allow the user to choose a specific text file on their drive for the word count
- The program must read the document and counts the occurrences of every word
- The program must remove all punctuations, hyphens, and other unwanted characters from the document
- The program must output another text file with the results
- The program must get a total count of all the words in the document
- The program must tests all the methods and report whether they passed or failed

2.1 Non-Functional Requirements

- The program must present a nice GUI for the file selection
- The program must handle all small and large input files
- The program must tests all the methods
- The program must be easy to use and require minimal user inputs
- The program must be reliable and produce accurate results
- The program must be efficient and use minimal system resources (no unnecessary methods or unused resources)

3.0 Architecture

The project uses different Java imports/libraries to execute the word count including: BufferedReader, BufferedWriter, FileReader, FileWriter, HashMap, and Map. It has 5 different classes: Controller, WordCount, WordCountGUI, WordCountModel, and WordCountTest.

WordCount: The main driver class for the application. It creates a WordCountGUI object and makes it visible.

WordCountGUI: It is the graphical user interface for the application. An instance of this GUI contains a reference to the controller.

WordCountTest: It is the test class for the Word Count application. It tests all the methods.

4.0 UML Diagram