Word Count and Search & Replace

User Document

Reagan Schulte & Art Gomez

# Project Goal

The goal of this project is to implement a program that allows a user to gather information about a file and modify it easily. The program is designed to accept user input through console commands. The program has two use cases which are count and replace.

# Potential Users

The users of this program will be those who routinely work with text files. Examples include programmers, data scientists, and IT technicians.

# Use Cases

## Count

The count use case is started by the user choosing option 0 from the main menu. Option 0 allows the user to see all the words and the placement of those words by character count. First the user enters the name for the file they want to be counted by the system. Then by choosing option 0 a new object wordcount is created, and the file name is passed to this object. Then the file is found inside the object and is counted. Once the object has all the words and placements of those words, it prints out all the words to the user. This always the user to see what kind of words they are using in their file and how often they use them, if they want to replace certain types of words.

## Replace

The replace use case is initiated after the user enters the name of the file and chooses option 1 from the main menu. The file name is passed to the “replace” object constructor which then prompts the user for a word to replace. The user is then prompted for a word to replace the first word. If no errors occur, the program overwrites the original file with a new file that has each desired word replaced with its replacement. The user can only replace one word for each initiation. For example, if a user initiates the use case and replaces the word “foo” with the word “bar” then all the occurrences of “foo” in the file are then replaced with “bar.” If the user wishes to also replace “boo” with “far,” they must initiate the replace use case again through the main menu.

# Use Instructions

The program can be executed via a Linux bash shell and navigating to the folder in which the java files were downloaded. The files SearchReplace.java, replace.java, and wordcount.java are required to be in the same folder before compiling. Executing “javac SearchReplace.java” will compile the program. Entering “java SearchReplace” will launch the program. When entering a filename, the user may include a file path or may just enter the file name if it is located in the same directory as the program.