**Module 7: Testing Assignment**

Eric Klausner

Technical Writing for Software CEN-3100C

Professor Mary Walauskis

June 25, 2023

The unit tests for TextAnalyzerGUI were designed to make sure that the program initiates and functions properly after it has been updated to a Graphical User Interface (GUI). I attempted to implement the tests using JUnit, which is a popular unit testing framework for Java, and was required for the assignment.

The first test, testGUIInitiation(), verifies the correct initiation of the GUI components. It does this by first obtaining the main frame of the GUI using the TestAnalyzerGUI.getframe() methods. The test is supposed to check that the rame is not null and has the expected title, which is set as “The Raven: Word Occurrences.”

The test then examines the components within the frame's content pane. It checks that there are two components: a JPanel and a JScrollPane. These components are responsible for containing the main panel and the result text area, respectively. The test validates that the first component is a JPanel and contains a single JButton with the text "Start Program". The second component is then verified to be a JScrollPane, and its viewport is expected to contain a JTextArea. The test confirms that the JTextArea is not editable, indicating that the user cannot modify the contents.

The second test, testStartButtonAction(), focuses on the functionality of the "Start Program" button. The test begins by retrieving the frame, panel, start button, and result text area from the GUI using appropriate method calls. The test ensures that the start button is initially enabled and the result text area is empty.

The test simulates a button click using the doClick() method on the start button. After the click, the test verifies that the start button is disabled and the result text area is no longer empty, indicating that the GUI has processed the click and performed the desired actions.