# INTRODUCTION

## Identification

This Software Requirements Specification (SRS) document establishes the overall software requirements for the Word Statistics software being created. This SRS can be used as a basis for design and testing of the Word Statistics software.

## Software Overview

“In this project you are writing a program to perform various word statistics of a given document (as a string). The initial requirement is to count the frequency of each unique word. The code should support combinations of space, tab, and newline characters as separators.”

This is the guideline and requirements handed down by the instructor and it will be the leading development guideline throughout the software’s lifecycle.

**2.0 REQUIREMENTS**

**2.1 Functional Requirements**

* This program is to take a block of text from the user through an input initially prompted by the program and count every unique combination of letters and symbols that form a string that is separated by white space. The program will output each word read and output the number of times each unique word was read.
* White space is to be considered a space, tab or newline.
* The program will treat a “word” as any combination of characters (a string) that are separated by white space.
  + The characters used can be any eligible char that the user can input, such as but not limited to letters in any language, numbers or symbols.
  + Capitalized and lowercase characters are considered different and thus two strings with identical characters excluding a capitalized character and lowercase character will be considered different. (ex: “Test” and “test” will be considered different strings.)
* The input of words will be given by the user manually upon running of the program. The program will not read from a file of any sort, the program will strictly read from the input given by the user at the beginning of the program
* A “word” can be any number of characters. There is no limit on the amount of characters per “word”.
* Output will be presented in the console (or equivalent depending on compiler used.) via a print statement. Each unique “word” will have a value associated with it for how many times it appears in the given input text

**2.3 Non-Functional Requirements**

* There are no explicit non-functional requirements. The code will not have any memory leaks or segmentation faults.
* The code will be extremely usable, requiring only a minor read-me that explains how to use the code and what will be expected from the code’s output.
* There will be almost an infinite mean time between failures, as the code will be simple enough to not use a significant amount of memory or resources.
* There is no requirement for the compiler used for the program. All that matters is that it is a python compiler.

**2.5 Test Case Requirements**

* There will be a test case to test that the code works with a “word” composed of English letters.
* There will be a test case to test that the code works with a “word” that contains only numbers.
* There will be a test case to test that the code works with a “word” that contains only symbols
* There will be a test case to test that the code works with a “word” that contains English letters and numbers.
* There will be a test case to test that the code works with a “word” that contains English letters and symbols.
* There will be a test case to test that the code works with a “word” that contains symbols and numbers.
* There will be a test case to test that the code works with a “word” that contains English letters, symbols and numbers.
* There will be a test case to test each type of white space (newline, tab, space).