WordCount

Software Requirements Specification

1.0.0

June 28 2016

Steven Mayner

Lead Software Engineer

Bentley Kang

Software Vice-Engineer

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 6/14/2016 | 0.0.1 First Draft | Mayner & Kang |  | |
| 6/17/2016 | 0.0.2 Rough Draft | Mayner & Kang |  | |
| 6/28/2016 | 1.0.0 First Release Draft | Mayner & Kang |  | |
|  |  |  |  | |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | Eduardo | Stakeholder |  | |
|  | TJ | Teacher’s Assistant |  | |

**Table of Contents**

Revision History ii

Document Approval ii

1. Introduction 1

Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.5 Overview 1

2. General Description 1

2.1 Product Perspective 1

2.2 Product Functions 2

2.3 User Characteristics 2

2.4 General Constraints 2

2.5 Assumptions and Dependencies 2

2.6 Risk Assessments 2

3. Specific Requirements 2

3.1 External Interface Requirements 2

3.1.1 User Interfaces 2

3.1.2 Hardware Interfaces 2

3.1.3 Software Interfaces 2

3.1.4 Communications Interfaces 3

3.2 Functional Requirements 3

3.2.1 Input Text 3

3.2.2 Counting System 3

3.2.3 Calculations 3

3.2.4 Output Report 3

3.2.4 Error Handling 3

3.3 Use Cases 3

3.3.1 Use Case #1 3

3.3.2 Use Case #2 4

3.3.3 Use Case #3 4

3.5 Non-Functional Requirements 5

3.5.1 Performance 5

3.5.2 Reliability 5

3.5.3 Availability 5

3.5.4 Security 5

3.5.5 Maintainability 5

3.5.6 Portability 5

3.8 Logical Database Requirements 5

# 1. Introduction

## Purpose

This SRS is presented for the usage of the Lead Software Engineer and Vice-Engineer to organize and clearly state the requirements of the WordCount project to stakeholder(s) to ensure a satisfactory deliverable product.

## 1.2 Scope

The ultimate goal of this project is to produce WordCount, a Python program operated via the command line, for use in analysis of student essays in a digital format. WordCount takes a text file as input and reports the average number of words used per sentence as output. The application of this is to facilitate the teacher’s students in writing shorter, more concise sentences as part of writing their essays.

To succeed in this goal, WordCount must be accessible and easily used by college students and faculty with little or no technical experience, require little or no maintenance, and maximize compatibility across platforms and operating systems.

## 1.3 Definitions, Acronyms, and Abbreviations

Counted word: For the purposes of the WordCount project, a counted word is a word longer than three letters. Only counted words are considered for the total sum used in the average formula.

Sentence: For the purposes of WordCount, a sentence is defined as a string of ending in a period, exclamation point, question mark, comma, colon, and semi-colon.

Word: For the purposes of WordCount, a word is defined as a string of alphabetical characters (a through z) excepting abbreviations such as “St.” or “Mr.” Addressed more specifically later.

## 1.5 Overview

Following this section is a description of WordCount’s traits as a program and a product, followed by more specific elaboration on the use cases and requirements expected of the program, and finally is a brief discussion of how this document itself will be updated.

# 2. General Description

The primary factor in the project’s functional and non-functional requirements is the varied and unpredictable nature of the user. Access to other software/hardware and technical skill cannot be assumed and must be accounted for. The user base is very large, and may include individuals requiring accessibility aid.

## 2.1 Product Perspective

The Microsoft Office Word have words counter on the bottom left corner. It counts pages, words, characters, paragraphs, etc. However it doesn’t show length of sentence and average length of sentence. Also Microsoft Office Word work in real time. However our program will not run in real time. The user will use the program when they are finished with the report. And the program will only output the average length of sentence. And only able to input .txt report.

## 2.2 Product Functions

The software will count the total words from the report and divide by the total count of sentences, and it will display that number. The product will also be able to process files individually or in batches, calculating the average across all input files.

## 2.3 User Characteristics

The user(s) will have an unknown level of technical experience. English proficiency is assumed. Users may be students or teachers.

## 2.4 General Constraints

The system must be compatible with accessibility devices such as text-to-speech and impaired vision displays. The system must be portable and able to run on any operating system given the presence of Python 3.

## 2.5 Assumptions and Dependencies

The read file is assumed to be .txt, and the program will not function if it reads a different file.

Python 3 must be present on user system.

## 2.6 Risk Assessments

Project is very low risk. The most significant risk is the small possibility of the stakeholder abandoning the project in favor of a different tactic for achieving their goal.

The secondary risk is Python rolling out an update that renders some segment of the code defective.

The third major risk is the small size of the team. With so few members, illness or injury may paralyze the project.

# 3. Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

The user will interface via the terminal.

### 3.1.2 Hardware Interfaces

The system will receive input via keyboard and output via display.

### 3.1.3 Software Interfaces

The system interfaces with Python and the operating system.

### 3.1.4 Communications Interfaces

The system will be distributed via USB port or electronic mail.

## 3.2 Functional Requirements

### 3.2.1 Input Text

Given a proper command, the system should read text off the indicated file for use.

### 3.2.2 Counting System

System should only count words longer than 3 letters, and should only count sentences ending with the defined punctuation “.” “?” “,” “:” “;”).

A word is defined as a string of alphabetical letters (a through z). Abbreviations shorter than 3 characters, beginning with an uppercase letter, and ending with a “.” are not words or sentences.

### 3.2.3 Calculations

System should accurately calculate average words per sentence of the read text, defined as the mean.

System should accurately calculate the average of a series of averages when processing several reports at once.

### 3.2.4 Output Report

System should make calculated average available to output methods.

### 3.2.5 Error Handling

Given improper input such as an incorrect file path or incorrect syntax, the system should not crash. The system should output a user-friendly message to correct the error.

## 3.3 Use Cases

### 3.3.1 Use Case #1

Using the program to read a single text file

### 3.3.2 Use Case #2

Program successfully accesses a file

### 3.3.3 Use Case #3

The program fails to access a file.

## 3.5 Non-Functional Requirements

### 3.5.1 Performance

System should output a report for an up to 50,000-character essay within 3 seconds 95% of the time.

System should output reports both to the command display and to a text file for batch processing.

### 3.5.2 Reliability

System should not crash when given arbitrary input. System should return a user-friendly error message to correct the input.

### 3.5.3 Availability

System should be available for use at any time once installed locally on the user’s system. System should output a report after correct input with no additional user input.

### 3.5.4 Security

System should not open any security vulnerabilities on the platform.

### 3.5.5 Maintainability

System should not require future updates within the next 4 years, and should not contain any bugs.

### 3.5.6 Portability

System should operate on any operating system given the constraint (Python), and should not rely on the presence of third-party files or programs to function.

## 3.8 Logical Database Requirements

No database will be required for this program.