Text File Analyzer

Software Requirements Specification (SRS)

Team Members:

Justin Dierken

Travis Delly

Brandon Quan

Alex Cure

Revision History

Date	Revision	Description	Author
10/16/17	1.0	First commit.	Travis Delly
10/22/17	1.05	All files created.	Travis Delly
10/22/17	1.1	First commit.	Alex Cure
10/22/17	1.15	Second commit.	Alex Cure
10/22/17	1.2	public to private variables.	Alex Cure
10/22/17	1.25	Add the file IO class.	Brandon Quan
10/22/17	1.3	Added name variable.	Alex Cure
10/22/17	1.35	Merge branch master.	Alex Cure
10/22/17	1.40	Uppercase S on string.	Travis Delly
10/25/17	1.45	Finished File IO class.	Brandon Quan
10/25/17	1.5	Database handler.	Travis Delly
10/25/17	1.55	Merge with master.	Travis Delly
10/25/17	1.6	Cleaned up code.	Travis Delly
10/29/17	1.65	Changes for Justin.	Justin Dierken
10/29/17	1.7	Adde remove punctuation	Brandon Quan
10/29/17	1.75	Changed Return type	Brandon Quan
10/29/17	2.0	Finalized tested, bring all together	Travis Delly

Table of Contents

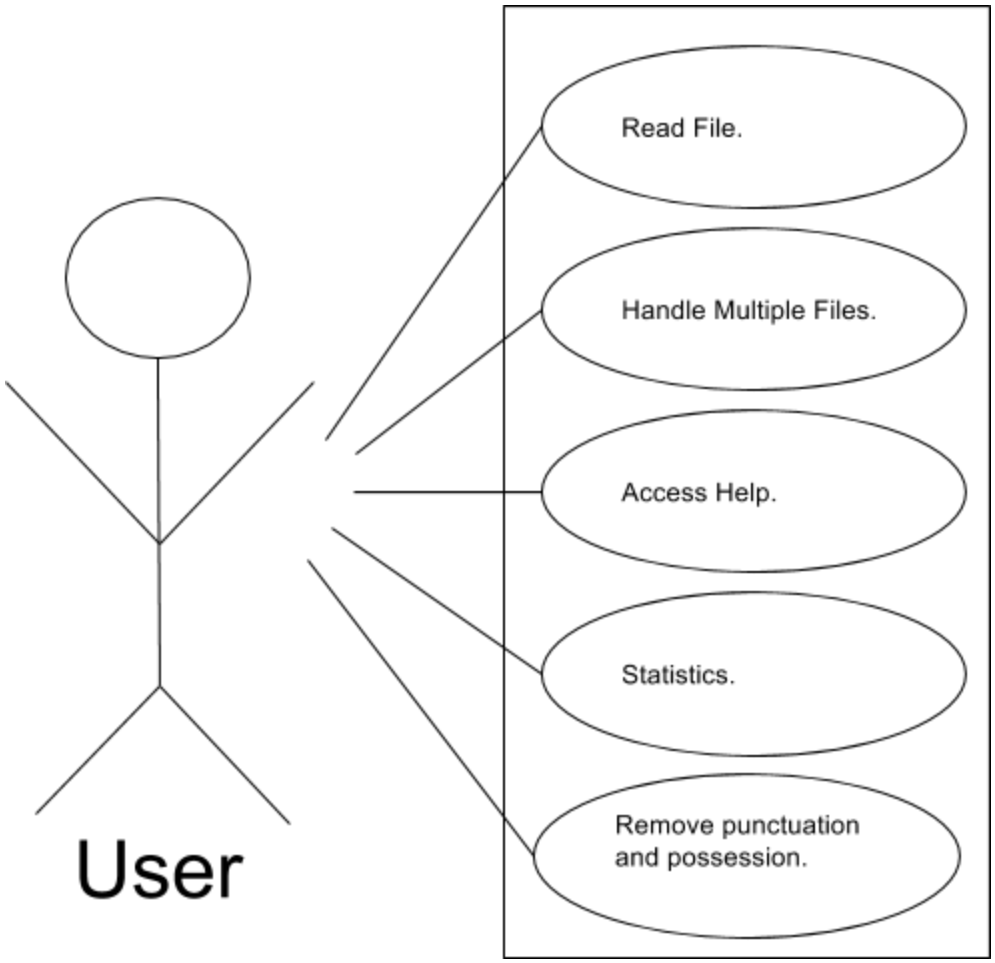
1. Project Overview

1.1. Overview

This program is to take in a text file and analyze it. During the analyzing it will determine the number of lines, number of blank lines, number of spaces, number of words, average characters per line, average word length, and most common words. It will also during the analyzing determine all of the above with punctuation and possession removed from the text file. It will then update the history containing all files analyzed with their name and date. It will also update the averages across all files. The program will contain a easy to use GUI, which will have options to input a text file, analyze said text file, a tab for history that will display date and file names that have been analyzed, a tab for analysis that will display averages across all files and individual statistics for each file, and a tab for help which will display help information for the program.

2. Use Cases

2.1. Use Case Diagram



Provide a use case diagram of this software product.

2.2. Use Case Scenarios

Provide a list of use case scenarios of this software product.

- As a user, I want to remove punctuation.
- As a user, I want to access the statistics of a text file.
- As a user, I want to read a file.
- As a user, I want to handle multiple files.
- As a user, I want to access help.

3. Test Plan

3.1. Test Cases

Specify required behavior of the software product and include parameters such as possible input, expected output, on error, etc.

	Feature	Description	Possible Input	Expected Output
1	Upload File.	Click on the Browse button and choose a file from directory.	Text File	Success message.
2	Upload File.	Click on the Browse button and choose a file from directory.	Binary File	Error message: Only text files are accepted.
3	Opening File and validating a file.	Click on analyze, readFile() will occur.	Text file	Nothing.
4	Access help.	Click on help.	Text File	Display help.
5	Removing punctuation and possess from analysis.	When analyze occurs, analysis of punctuation and possession removed will occur also.	Text File	Updated text file.
6	Reporting number of lines.	Click on analyze, readFile() operation will occur.	Text File	Update the object in the FileAnalyzer class.
7	Reporting number of blank lines.	Click on analyze, readFile() operation will occur.	Text File	Update the object in the FileAnalyzer class.
8	Reporting number of spaces.	Click on analyze, readFile() operation will occur.	Text File	Update the object in the FileAnalyzer class.
9	Reporting number of words.	Click on analyze, readFile() operation will occur.	Text File	Update the object in the FileAnalyzer class.

10	Reporting average characters per line.	Click on analyze, readFile() operation will occur.	Text File	Update the object in the FileAnalyzer class.
11	Reporting average word length.	Click on analyze, readFile() operation will occur.	Text File	Update the object in the FileAnalyzer class.
12	Reporting most common words.	Click on analyze, readFile() operation will occur.	Text File	Update the object in the FileAnalyzer class.
13	Reporting history of all files processed.	Click on analyze, insertRow() operation will occur.	Text File	Update database handler.
14	Reporting averages across all files.	Click on analysis tab, average will be displayed at the top.	Text File	Return averages.

4. Project Plan

4.1. Plan for the First Deliverable

Specify the plan for the first deliverable including the list of tasks with a description, assignment of tasks to team members, etc.

	Task	Description	Assignment
1	Class Diagram.	Class Diagram of the project will be drawn using the in Google docs.	Justin
2	Use of GitHub.	Commit updates as they occurred.	Alex, Travis, Justin, Brandon.
3	File Analyzer Class	File analyzer class to hold statistics of individual files.	Alex

4	GUI	Created GUI class so the user could perform all of the actions desired on text file.	Justin
5	File IO Class	Creates the object and then analysis on the inputted file.	Brandon
6	Database handler, Main	Handles the history and averages.	Travis
7	Executing program.	Pull parts together, check functionality.	Alex, Travis, Justin, Brandon

4.2. Plan for the Second Deliverable

	Task	**Description**	**Assignment**
1	Update GUI	Make user interface easier to understand.	Justin
2	Update documentation	Update documentation to include more test cases.	Alex
3	Perform test cases.	Perform test cases, update program to fix errors found in test cases.	Alex, Travis, Justin, Brandon.
3	Plan for deliverable 3	Discuss and generate a plan for deliverable 3.	Alex, Travis, Justin, Brandon.

5. Class Diagram

5.1. Class Diagram