SYSTEM TEST PLAN

Instructions. In this section, you must provide your system test plan with at least 5 test cases. Make sure:

- ✓ You describe how to setup the system to begin system testing
- ✓ Test IDs are uniquely identified and descriptive
- ✓ Test descriptions are fully specified with complete inputs, specific values, and preconditions
    - o Be sure to provide SPECIFIC INPUTs and VALUEs so that your test cases are repeatable
- ✓ Expected results are fully specified with specific output values
- ✓ All tests cover scenarios based on the problem statement
- ✓ All tests cover unique scenarios for the system
- ✓ All strategies for system testing are demonstrated in the tests (testing equivalence classes, testing boundary values, testing exceptions/unexpected inputs)

For test cases the file text.txt, containing the following will be used.

the farmer was working hard to grow his crops all summer so that his crops would be ready for everyone to enjoy

The file filter.txt will also be used containing the following,

the to

Test ID	Description	Expected Results	Actual Results
testLoadInput **(Equivalence class – loading text from file)**	**Preconditions**: • WordCloudUI has been loaded successfully. • text.txt and filter.txt exist **Steps**: 1. Specify the input file that contains the text of the literature. 2. **input/input.txt** 3. You may also specify the input file that contains the text of filter words, or type na. 4. **input/filter.txt**	text.txt and filter.txt are successfully read into the program	text.txt and filter.txt were successfully read into the program
testFrequency **(Exception/unexpected input – a word that is not contained in the text)**	**Preconditions**: • WordCloudUI has been loaded successfully. • text.txt and filter.txt have	The word (tomato) is contained in the text 0 times.	The word (tomato) is contained in the text 0 times.

	been loaded successfully **Steps**: 1. **Frequency** 2. The word to get the frequency of: **tomato**		
testMostFrequentWords 1 **(Boundary Value – testing for the minimum number of words that can be displayed 1)**	**Preconditions**: • WordCloudUI has been loaded successfully. • text.txt and filter.txt have been loaded successfully **Steps**: **1.** **Words** 2. The number of words you would like to generate a report for: **1**	MostFrequentWords[crops – 2]	MostFrequentWords[crops – 2]
testMostFrequentWords <0 **(Exception/unexpected input – giving a number of words that is less than 1)**	**Preconditions**: • WordCloudUI has been loaded successfully. • text.txt and filter.txt have been loaded successfully **Steps**: 1. **Words** **2.** The number of words you would like to generate a report for: **-5**	MostFrequentWords[Number of words must be greater than 0.]	MostFrequentWords[Number of words must be greater than 0.]
testWordCloudMax **(Boundary Value – giving a number of words greater than the total number of words in the text)**	**Preconditions**: • WordCloudUI has been loaded successfully. • text.txt and filter.txt have been loaded successfully **Steps**: **1.** **Cloud** 2. The output file of the word cloud: **output/output.h**	*all words (except the and so) will be printed in a word cloud with “crops” and “his” being the largest words with all other words printed at the same size.	All words are printed with “his” and “crops” being the largest and a different color. The title “WordCloud” also appears in the top left corner.

	tml 3. The title of the word cloud: **WordCloud** 4. The number of words to include in the word cloud: **35** 5. Check output		