System Test Plan

Instructions. In this section, you must provide your system test plan with at least 5 test cases. **If you want to provide more than 5 test cases, add an appendix at the end of this document with the 6th, 7th, etc. test cases so that page numbers for all sections match the required template!**

Make sure:

- You provide your sample test data
- Test IDs are uniquely identified and descriptive
- Test descriptions are fully specified with complete inputs, specific values, and preconditions
    - 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)

Test Data:
For our system test cases, we will use the following test files:
input1-1.txt:

```
A flea and a fly in a flue
Were imprisoned so what could they do
Said the fly let us flee
Let us fly said the flea
So they flew through a flaw in the flue
Ogden Nash
```

input1-2.txt:

```
A flea and a fly in 4 flue
Were imprisoned so what could they do
Said the 5 let us flee
Let 18 5 said 16 2
```

```
So 13 flew through 4 flaw 6 16 7
Ogden Nash
```

input2.txt:

```
Foo
```

empty.txt (empty):

Test ID	Description	Expected Results	Actual Results

Test #1
 	
testID:
 	
testCompressFile
 	
Strategy:
 	
(Equivalence class - compressing a file)
	Preconditions:
	
	● CompressionUI has loaded successfully
	● The file input1-1.txt exists
	
	Steps:
	
	1. Enter "input/input1-1.txt"
	2. Enter "compress"
		The compressed file contents display in the program as:
		`Compressed Output {`
		`  Line 1:A flea and a fly in 4 flue`
		`  Line 2:Were imprisoned so what could they do`
		`  Line 3:Said the 5 let us flee`
		`  Line 4:Let 18 5 said 16 2`
		`  Line 5:So 13 flew through 4 flaw 6 16 7`
		`  Line 6:Ogden Nash`
		`}`
			Displayed contents:
			`Compressed Output {`
			`  Line 1:A flea and a fly in 4 flue`
			`  Line 2:Were imprisoned so what could they do`
			`  Line 3:Said the 5 let us flee`
			`  Line 4:Let 18 5 said 16 2`
			`  Line 5:So 13 flew through 4 flaw 6 16 7`
			`  Line 6:Ogden Nash`
			`}`

Test #2
 	
testID:
 	
testDecompressFile
 	
Strategy:
 	
(Equivalence class - decompressing a file)
	Preconditions:
	
	● CompressionUI has loaded successfully
	● The file input1-2.txt exists
	
	Steps:
	
	1. **Enter “input/input1-2.txt”**
	2. **Enter “decompress”**
		The decompressed file contents display in the program as:
		`Decompressed Output {`
		`  Line 1:A flea and a fly in a flue`
		`  Line 2:Were imprisoned so what could they do`
		`  Line 3:Said the fly let us flee`
		`  Line 4:Let us fly said the flea`
		`  Line 5:So they flew through a flaw in the flue`
		`  Line 6:Ogden Nash`
		`}`
			Displayed contents:
			`Decompressed Output {`
			`  Line 1:A flea and a fly in a flue`
			`  Line 2:Were imprisoned so what could they do`
			`  Line 3:Said the fly let us flee`
			`  Line 4:Let us fly said the flea`
			`  Line 5:So they flew through a flaw in the flue`
			`  Line 6:Ogden Nash`
			`}`

Test #3
 	
testID:
 	
testCompressOneWordFile
 	
Strategy:
 	
(Boundary value - input single word file, file is valid but no compression is done)
	Preconditions:
	
	● CompressionUI has loaded successfully
	● The file input2.txt exists
	
	Steps:
	
	1. **Enter “input/input2.txt”**
	2. **Enter “compress”**
		The compressed file contents display in the program as:
		`Compressed Output {`
		`  Line 1:Foo`
		`}`
			Displayed contents:
			`Compressed Output {`
			`  Line 1:Foo`
			`}`

Test #4 **testID:** **testInvalidFileCompression** **Strategy:** **(Exception/unexpected input - try compressing empty file)**	**Preconditions:** ● CompressionUI has loaded successfully ● The file empty.txt exists **Steps:** 1. **Enter “input/empty.txt”** 2. **Enter “compress”**	The program displays: `“The provided input file has no text to compress.”`	Displayed message: `“The provided input file has no text to compress.”`

Test #5 **testID:** **testInvalidFileDecompression** **Strategy:** **(Exception/unexpected input - try decompressing empty file)**	**Preconditions:** 1. CompressionUI has loaded successfully 2. The file empty.txt exists **Steps:** **3. Enter “input/empty.txt”** **4. Enter “decompress”**	The program displays: `“The provided input file has no text to decompress.”`	Displayed message: `“The provided input file has no text to decompress.”`