Test Plan

KidText

Trent Apple, Chris Bauschka, Tyler Hoffman, Elizabeth Murphy, Aaron Petry, Kyle Ribordy, Brittany Vacchiano

Our design plan is to implement the functionality in a modular fashion. This will allow automated testing of a majority of the planned functionality.

Testing KidText

This testing suite contains both manual and automated tests. The manual tests can be performed by following the directions supplied below. All of the automated unit tests are located in kttest.java. Each automated unit test has a comment header similar to the tests defined below to allow efficient searching for a specific test. To run all the unit tests, run kttest all or simply kttest. Specific unit tests can be run by supplying the test identification code. For example, if kt_ss000 needs to be run, run kttest kt_ss000.

Test Cases

Each test case consists of three parts. Part one (A) is the identification code, a short title, and a severity level. The severity level is a linear scale from 1 to 4 with 1 being the most severe to 4 being the least severe. Part two (B) is the instructions required to manually run the test. Many of the tests also have automated unit test counterparts that can be run through kttest. Part three (C) is the expected result if the program performs as documented (i.e. the test fails to detect an error). An example can be seen below:

A. kt_ss000 - String Statistics: word count - Severity 4
B. Select the Statistics menu item and then click the Word Count item in the dropdown menu.
C. A popup window is created with the count of the number of words in the

document.

1. Unit Tests

1.1. Replace

A. Overall:

i. kt_rp000 - Replace Overall: String in Text - Severity 1
Input: Search for valid string present in file
Output: Should highlight all instances of the string.

ii. kt_rp001 - Replace Overall: String Not in Text, Severity 1
Input: Search for valid string not present in file
Output: Should receive message stating that no instances were found.

iii. kt_rp002 - Replace Overall: No Input - Severity 2
Input: Search with no input
Output: Should receive error message

iv. kt_rp003 - Replace Overall: Blank File - Severity 2
Input: Search empty file
Output: Should receive error message

v. kt_rp004 - Replace Overall: String too Long - Severity 3
Input: Search for string longer than file
Output: Should receive error message

B. Case Sensitive

i. kt_rp005 - Replace Case Sensitive: Same Case - Severity 1
Input: Search for string with matching case settings
Output: Should highlight all instances of the string.

ii. kt_rp006 - Replace Case Sensitive: Different Case - Severity 1
Input: Search for string only found in different case settings (i.e., search for “hello” when text contains “Hello”)
Output: should receive message stating that no instances were found

C. Case Insensitive

i. kt_rp007 - Replace Case Insensitive: Different Case - Severity 1
Input: Search for string found in different case settings (i.e., search for “hello” when text contains “Hello”).
Output: Should highlight all instances of the string.

ii. kt_rp008 - Replace Case Insensitive: Both Cases - Severity 2
Input: Search for string found with both same and different case settings.

Output: Should highlight all instances of the string.

D. Whole Word
    i. kt_rp009 - Replace Whole Word: Word Present - Severity 1
    Input: Search for word found in text
    Output: Should highlight all instances of string
    ii. kt_rp010 - Replace Whole Word: Only String Present - Severity 1
    Input: Search for word found in text only as part of another word (i.e., search for “heart” when text contains “heartache”).
    Output: Should receive message stating that no instances were found.
    iii. kt_rp011- Replace Whole Word: Both Present - Severity 2
    Input: Search for string found in text both as whole word and as part of another word.
    Output: Should only highlight instances where string is separate word.

E. Context (based upon selection)
    i. kt_rp012 - Replace Context: String in Selection - Severity 1
    Input: Search for string found in selection.
    Output: Should highlight all instances of the string.
    ii. kt_rp013 - Replace Context: String Only in Text - Severity 1
    Input: Search for string not found in selection but found in text.
    Output: Should receive message stating that no instances were found.
    iii. kt_rp013 - Replace Context: String Not in Text - Severity 2
    Input: Search for string not found in text.
    Output: Should receive message stating that no instances were found.

F. Wildcard
    i. kt_rp015 - Replace Wildcard: Wildcard in Text - Severity 1
    Input: Search for properly formatted wildcard string found in text.
    Output: Should highlight all instances of string.
    ii. kt_rp016 - Replace Wildcard: Wildcard Not in Text - Severity 1
    Input: Search for properly formatted wildcard string not found in text.
    Output: Should receive message stating that no instances were found.
    iii. kt_rp017 - Replace Wildcard: Bad Wildcard - Severity 2
    Input: Search for improperly formatted wildcard string.

Output: Should receive error message.

G. Regex
    i. kt_rp018 - Replace Regex: Regex in Text - Severity 1
    Input: Search for properly formatted regex found in text.
    Output: Should highlight all instances of string.
    ii. kt_rp019 - Replace Regex: Regex Not in Text - Severity 1
    Input: Search for properly formatted regex not found in text.
    Output: Should receive message stating that no instances were found.
    iii. kt_rp020: Bad Regex - Severity 2
    Input: Search for improperly formatted regex.
    Output: Should receive error message.

1.2. Compare/Diff

A. Normal Input
    i. kt_cd000 - Compare/Diff: Normal Input - Severity 1
    ii. Select the Compare menu item and then select the text document to compare the text with.
    iii. A popup window with the difference between the two documents will appear.

B. Empty Input
    i. kt_cd001 - Compare/Diff: Empty Input - Severity 2
    ii. Select the Compare menu item and then select the text document to test the empty string with.
    iii. A popup window will appear with the difference between the two documents.

C. Identical Input
    i. kt_cd002 - Compare/Diff: Identical Input - Severity 1
    ii. Select the Compare menu item and then select the text document identical to the current text.
    iii. A popup window will appear showing that the files are identical.

D. Illegal Input
    i. kt_cd003 - Compare/Diff: Bad Input - Severity 3
    ii. Select the Compare menu item and then select the binary file to attempt to compare.
    iii. A popup window will appear showing that the file is illegal and cannot be compared.

E. Large Input
    i. kt_cd004 - Compare/Diff: Large Input - Severity 2

ii. Select the Compare menu item and then select the large input text document to compare.
iii. A popup window will appear with the difference between the two documents.

1.3. Text Statistics

A. Word Count

i. kt_ts001 - Word Count: Normal input - Severity 1
Input: A string with no special characters, words separated by spaces (ex. “this is a string without any trip ups”).
Output: An accurate count of words.

ii. kt_ts002 - Word Count: Empty input - Severity 3
Input: An empty string.
Output: A count of 0 words.

iii. kt_ts003 - Word Count: Special Characters - Severity 3
Input: Text separated by special characters (-_.!? etc) (ex. “This is a string with special.characters_which may cause problems”).
Output: A correct count of words (only spaces break apart words).

B. Number of Brackets

i. kt_ts004 - Bracket Count: Even brackets - Severity 1
Input: A string with an even number of brackets.
Output: A count of the brackets in the string.

ii. kt_ts005 - Bracket Count: Odd brackets - Severity 2
Input: A string with an odd number of brackets.
Output: A count of the brackets, as well as an alert that there is an odd number.

iii. kt_ts006 - Bracket Count: Empty string - Severity 2
Input: An empty string.
Output: A bracket count of 0.

iv. kt_ts007 - Bracket Count: No brackets - Severity 3
Input: A string containing no brackets.
Output: A bracket count of 0.

C. Number of Sentences

i. kt_ts008 - Sentence Count: One sentence - Severity 1
Input: A string with one sentence.
Output: A sentence count of 1.

ii. kt_ts009 - Sentence Count: Multiple sentences - Severity 1
Input: A string with multiple sentences.
Output: An accurate count of sentences.

iii. kt_ts010 - Sentence Count: Repeating periods - Severity 2

Input: A string with sentences separated by more than one period.
Output: An accurate count of sentences--not periods.

iv. kt_ts011 - Sentence Count: Empty string - Severity 3
Input: An empty string.
Output: A sentence count of 0.

D. Number of Spaces

i. kt_ts012 - Spaces Count: Empty string - Severity 3
Input: An empty string.
Output: A space count of 0.

ii. kt_ts013 - Spaces Count: String with no spaces - Severity 2
Input: A non-empty string containing no spaces.
Output: A space count of 0.

iii. kt_ts014 - Spaces Count: String with spaces - Severity 1
Input: A string containing spaces.
Output: An accurate count of the spaces in the string.

E. Search Result Count

i. kt_ts015 - Search Result Count: Not found - Severity 2
Input: A string and a search term not inside said string.
Output: A search result count of 0.

ii. kt_ts016 - Search Result Count: No search term - Severity 2
Input: A string and an empty search term.
Output: A search result count of 0.

iii. kt_ts017 - Search Result Count: Highlighted term - Severity 1
Input: A user highlights a term and clicks “Find.”
Output: A correct count of the term’s occurrences in the document.

iv. kt_ts018 - Search Result Count: Inputted term - Severity 1
Input: A user clicks “Find” and types in a search term.
Output: A correct count of the term’s occurrences in the document.

v. kt_ts019 - Search Result Count: Different Cases - Severity 2
Input: A user selects “Case Sensitive Search.”
Output: A count of the term only matching in matching case.

vi. kt_ts020 - Search Result Count: Indifferent Cases - Severity 2
Input: A user selects “Case Insensitive Search.”
Output: A count of the term in any matching case.

1.4. Basic Cipher

A. Encrypt

i. kt_bc001 - Basic Cipher ROT13 Encrypt: Normal mixed case input - Severity 1
   Input: A string with no special characters containing at least five (5) unique mixed-case letters and at least two being more than 13 letters apart in the alphabet(e.g. AbRsT).
   Output: A shifted / rotated variation by 13 characters based upon the inputted data where lowercase stays lowercase, and uppercase remains uppercase (e.g. NoEfG).
ii. kt_bc002 - Basic Cipher ROT-N Encrypt: Normal mixed case input - Severity 1
   Input: A string with no special characters containing at least five (5) unique mixed-case letters and at least two being more than 26 - N letters apart in the alphabet (e.g. AbRsT).
   Output: A shifted / rotated variation by N characters based upon the inputted data where lowercase stays lowercase, and uppercase remains uppercase (e.g. NoEfG).
iii. kt_bc003 - Basic Cipher ROT-N Encrypt: Spaces and special characters - Severity 1
   Input: A string with letters, special characters, and spaces of length 5 or more.
   Output: A properly N shifted string, but with only the letters shifted. The special characters should remain the same.

B. Decrypt
   i. kt_bc004 - Basic Cipher ROT13 Decrypt: Normal mixed case input - Severity 1
      Input: A shifted / rotated variation by 13 characters or previously encrypted by the ROT13 Encrypt function (e.g. NoEfG)
      Output: A string shifted to the left by 13 characters to display the unencrypted version of the text (e.g. AbRsT).
   ii. kt_bc005 - Basic Cipher ROT-N Decrypt: Normal mixed case input - Severity 1
      Input: A shifted / rotated variation by N characters or previously encrypted by the ROT-N Encrypt function with letters of mixed case (e.g. NoEfG)
      Output: A string shifted to the left by N characters to display the unencrypted version of the text where lowercase stays lowercase, and uppercase remains uppercase (e.g. AbRsT).
   iii. kt_bc006 - Basic Cipher ROT-N Decrypt: Spaces and special characters - Severity 1

Input: A string with letters, special characters, and spaces of length 5 or more.
Output: A properly N shifted string, but with only the letters shifted. The special characters should remain the same.

1.5. Syntax Highlighting

A. Java Syntax

i. kt_sh001 - Single Keyword Highlight
Input: A single Java keyword in the text box (e.g. “public”)
Output: Word should highlight in another color signifying its a keyword.

ii. kt_sh002 - Single Keyword-within-word
Input: A single word that contains a Java keyword (e.g. catpublic)
Output: Word should not highlight.

iii. kt_sh003 - Single Comment Highlight
Input: A single comment (/* comment */)
Output: the comment should be in the color assigned to comments.

iv. kt_sh004 - Single line comment
Input: a comment appearing at the end of a line of code (e.g. // comment)
Output: the comment should be in the color assigned to comments.

v. kt_sh005 - Single Function Highlight (function())
Input: a single function
Output: the word of the function declaration should be highlighted in the color assigned to functions.

vi. kt_sh006 - Single Constant Highlight
Input: A capital letter constant (e.g. NUM_LOOPS)
Output: Word should be highlighted in color assigned to constants

vii. kt_sh007 - Complex file highlight
Input: A large 100+ line .java file
Output: A properly highlighted file according to Java syntax.