

SE 3XA3: Test Report

ReTouch

Team #7, ReTouchers
Abrar Attia - attiaa1
Susan Fayez - fayezs
Mediha Munim - munimm

December 6, 2017

Contents

1	Functional Requirements Evaluation	1
2	Nonfunctional Requirements Evaluation	2
2.1	Usability	2
2.2	Performance and Robustness	2
2.3	Cultural	3
3	Comparison to Existing Implementation	3
4	Unit Testing	4
5	Changes Due to Testing	4
6	Automated Testing	4
7	Trace to Requirements	8
8	Trace to Modules	8
9	Code Coverage Metrics	8

List of Tables

1	Revision History	i
2	Trace Between Tests and Requirements	9
3	Trace Between Tests and Modules	10

List of Figures

Table 1: **Revision History**

Date	Version	Notes
Dec 6	1.0	Test Report Updated

1 Functional Requirements Evaluation

The description of all tests is provided. The purpose of the following tests is to demonstrate that all the specified functional requirements were met. Also, that a user would be able to use and run through the application successfully. The tests focus on the methods and requirements for lesson selection, execution, mistakes and correctness, generation and display.

Test Name: LS

Results: Lesson Selection - The user is able to select a lesson and is redirected to the lesson screen.

Test Name: LE

Results: Lesson Execution - The lesson runs and tracks the current character. The user is able to complete a line and press enter to move on to the next line or complete the lesson.

Test Name: LMC

Results: Lesson Mistakes and Corrections - The system highlights incorrect characters and corrects them when user fixes the mistake.

Test Name: LG

Results: Lesson Generation - The system generates lessons that contain a maximum number of characters per line.

Test Name: LD

Results: Lesson Display - The system displays all the required results of the user on the lesson screen and in the results page.

2 Nonfunctional Requirements Evaluation

The nonfunctional requirement evaluation was based off of manual testing that involved running a user survey. The ReTouch system was tested by a selected test group of individuals. This allowed for a demonstration of the usability of the application from the perspective of people with different levels of technical expertise. Participants were asked to run through the ReTouch application as well as the original project; KTouch, and answer a set of questions. The supervisors (ReTouch Developers) recorded the answers and any additional questions or concerns that the participants had.

2.1 Usability

Test Name: UH1

Initial State: The Lesson selected and about to begin.

Input/Condition: The user asked to complete the lesson, with no further guidance.

Expected Output: The user should be able to complete the lesson.

Result: All participants were able to successfully complete the full lesson without assistance from the supervisors.

Test Name: UH2

Initial State: The application unopened

Input/Condition: The user, asked to complete the process of opening the application, selecting a Lesson, and viewing the results without further guidance

Expected Output: The user should be able to easily navigate the three stages of the application.

Result: 91 percent of all users scaled the application to be 4+ on a 5 point scale of ease of use.

2.2 Performance and Robustness

Test Name: P1

Initial State: During the Lesson.

Input/Condition: The user works on the Lesson.

Expected Output: The system will respond to the user input within 1 second.

Result: No complaints or noticing of any lagging by the participants.

Test Name: P2

Initial State: During the Lesson.

Input/Condition: The user works on the Lesson.

Expected Output: The timer displayed to the user will be accurate and begin once the lesson loads and will end once the user hits enter on the last line to complete the Lesson

Result: No record of incorrect elapsed time for lesson completion.

Test Name: P5

Initial State: Any point in using the application

Input/Condition: The user uses the application.

Expected Output: The application will not interfere with the users machine.

Result: No documented occurrences of the application interfering with the users machine when participants ran through the program.

2.3 Cultural

Test Name: C1

Initial State: The application is unopened.

Input/Condition: A user will be asked to open and go through the three stages of the application (Lesson Selection, Lesson, and Results).

Expected Output: The user shall not be offended by any of the content within the application.

Result: All participants answered that they were not offended by anything within the application.

3 Comparison to Existing Implementation

The ReTouch application was compared to the original application KTouch through the manual testing. This included conducting a user survey and allowing participants to run through both programs for comparison. The results of the survey included: Question: Which program was easier to use KTouch or ReTouch? The results found that 90.9 percent of the participants selected that ReTouch is easier to use in comparison to KTouch.

4 Unit Testing

Unit testing was completed using the JUnit unit testing framework and AssertJ which is used to simulate user interaction and test the required functional tests. The tests and their results are shown within automated testing.

5 Changes Due to Testing

After manual and automated testing were completed, a few changes to the system were made. After conducting the user surveys, it was advised by participants to include instructional based pop-ups that would allow the user to know what to do next. This was implemented by including "Press Backspace" and "Press Enter" pop-ups that clearly notified users when they make a mistake or need to press enter to move onto the next line.

In addition, the appearance of the GUI was also altered to make it more user friendly. This included adding coloured backgrounds to each page of the application and consistent text throughout to help instruct the user on what to do in each page.

6 Automated Testing

Lesson Selection

Test: FT1

Initial State: Lesson selection (start) screen.

Input: The desired Lesson.

Expected Output: User redirected to beginning of Lesson screen.

Result: PASS; System directs user automatically to the selected lesson screen when the start button is selected.

Lesson Execution

Test: FT4

Initial State: Beginning of the Lesson screen.

Input: A desired Lesson.

Expected Output: The Lesson is run, the system sets the current character to the first character in the Lesson.

Result: PASS; System confirms that the first character in the line is the cur-

rent character.

Test: FT4

Initial State: Beginning of the Lesson screen.

Input: A desired Lesson.

Expected Output: The Lesson is run, the system sets the current character to the first character in the Lesson.

Result: PASS; System confirms that the first character in the line is the current character.

Test: FT5

Initial State: During a Lesson.

Input: User types any character.

Expected Output: The system moves the current character value to the next one in the Lesson.

Result: PASS; System successfully tracks the current character and moves the value to the next value in the lesson whenever a character is entered.

Test: FT6

Initial State: During a Lesson

Input: User types any character in the Lesson.

Expected Output: The system sets that particular character in the Lesson as completed and the cursor moves on to the next character.

Result: PASS; System changes the colour of the character to correct and moves the cursor over to the next character (current character).

Test: FT8(i)

Initial State: During the Lesson, when the user reaches the end of a line having completed the correct characters.

Input: The user hits the enter key.

Expected Output: If the user can move on to the next line.

Result: PASS; The system moves the cursor over to the next line when the enter key is passed into the system.

Test: FT8(ii)

Initial State: During the Lesson, when the user reaches the end of a line, having made mistakes.

Input: The user hits the enter key.

Expected Output: The user may not move on the the next line.
Result: PASS; The system does not allow the cursor to move onto the next line unless all prior characters are correct.

Test: FT14(i)

Initial State: During the Lesson, when the user has finished typing everything correctly.

Input: The user hits the enter key.

Expected Output: The Lesson page is closed

Result: PASS; The system closed the lesson page once the enter key is sent through at the end of the lesson.

Test: FT14(ii)

Initial State: During the Lesson, when the user has finished typing everything but has made uncorrected mistakes.

Input: The user hits the enter key.

Expected Output: The Lesson page does not terminate until the user goes back and corrects their mistakes.

Result: PASS; The lesson page does not close until all prior characters are correct.

Lesson Mistakes and Corrections

Test: FT7

Initial State: During the Lesson.

Input: The user enters an incorrect character.

Expected Output: The system highlights the the character that was entered incorrectly, and if the user continues typing, each character after the initial incorrect character is highlighted as incorrect.

Result: PASS; The system recognizing incorrect characters and highlights them.

Test: FT9

Initial State: During the Lesson.

Input: The user enters the backspace key.

Expected Output: The current character is moved to the previous character in the Lesson. If the current character is the first in the Lesson and backspace is entered, nothing happens. Result: PASS; Nothing occurs when backspace

is entered at the current character.

Test: FT10

Initial State: During the Lesson.

Input: User input for the Lesson, with some mistakes.

Expected Output: The number of characters entered incorrectly is stored in a variable.

Result: PASS; The value of all the incorrect characters entered are stored in the module correctly.

Lesson Generation

Test: FT2

Initial State: Lesson selection screen

Input: The user selects a Lesson

Expected Output: The system will generate a Lesson - a list of characters of less than MAXLESSON characters, including spaces

Result: PASS; The system confirms that every line in the lesson does not surpass the maximum.

Lesson Display

Test: FT3

Initial State: The beginning of the Lesson.

Input: The user works on the Lesson.

Expected Output: The system displays the list of characters in the Lesson. The list of characters will be presented on separate lines, with each line than constant MAX LINE. Result: PASS; The system displays all the characters on the lesson page and the number of characters per line never exceeds the maximum.

Test: FT11

Initial State: During the Lesson.

Input: The user works on the Lesson.

Expected Output: The system displays the elapsed time, from when the use began the Lesson.

Result: PASS; The system displays the correct elapsed time.

Test: FT12

Initial State: During the Lesson.

Input: The user works on the Lesson.

Expected Output: The system displays the users typing accuracy.

Result: PASS; The system displays the correct accuracy.

Test: FT13

Initial State: During the Lesson

Input: The user works on the Lesson

Expected Output: The system displays the users typing speed

Result: PASS; The system displays the correct typing speed.

Test: FT15

Initial State: The Lesson is completed

Input: The user hits enter after they finish the Lesson

Expected Output: The results (time, typing accuracy, and typing speed) are displayed to the user.

Result: PASS; The system correctly displays all the user results on the final results page.

7 Trace to Requirements

The trace between the tests and the requirements is shown in Table 2 (below).

8 Trace to Modules

The trace between the tests and the modules is shown in Table 3 (below).

9 Code Coverage Metrics

The developers of the ReTouh application have estimated that roughly 70 percent code coverage is achieved throughout the tests. This number is based off the fact that the modules that have been covered by the tests contain the most code and the most necessary code since it conducts most of the functionality of the application. This is evident within the trace to modules table.

Test	Requirements
Functional Requirements Testing	
FT16	FREQ1
FT17	FREQ17
FT4	FREQ4
FT5	FREQ5
FT6	FREQ6
FT8(i)	FREQ8
FT8(ii)	FREQ8
FT14(i)	FREQ14
FT14(ii)	FREQ14
FT7	FREQ7
FT9	FREQ9
FT10	FREQ10
FT2	FREQ2
FT3	FREQ3
FT11	FREQ11
FT12	FREQ12
FT13	FREQ13
FT15	FREQ15
Non-functional Requirements Testing	
LF3	LF3
UH1	UH1
UH2	UH2
UH3	UH3
UH4	UH4
P1	P1
P2	P2
P5	P5
OE1	OE1
OE2	OE2
S1	S1
C1	C1

Table 2: Trace Between Tests and Requirements

Test	Modules
Automated Testing	
FT1	M5
FT16	M10, M3
FT17	M10, M3
FT4	M2, M6
FT5	M2, M6
FT6	M2, M6
FT8(i)	M2, M6
FT8(ii)	M2, M6
FT14(i)	M6
FT14(ii)	M6
FT7	M2, M6
FT9	M2, M6
FT10	M6
FT2	M4, M5
FT3	M2, M3, M5
FT11	M3, M9
FT12	M3, M9
FT13	M3, M9
FT15	M6, M7, M8

Table 3: Trace Between Tests and Modules