

SE 3XA3: Test Report

MacSidenotes

Team #04

Josh Mitchell mitchjp3

Matthew Shortt shorttmk

December 7, 2016

Contents

1	Functional Requirements Evaluation	1
1.1	Sidebar - Typing Window	1
1.2	Note Taking	2
1.3	Note Saving	2
1.4	Master List	2
1.5	Revisit URL	3
1.6	Note Deletion	3
2	Nonfunctional Requirements Evaluation	4
2.1	Look and Feel	4
2.2	Usability and Humanity Requirements	5
2.3	Performance	5
3	Comparison to Existing Implementation	5
4	Automated Unit Testing	6
4.1	clickCounter	6
4.2	showList	6
4.3	emptyMasterList	6
4.4	showNotice	6
5	Changes Due to Testing	7
6	Trace to Requirements	7
7	Trace to Modules	8
8	Code Coverage Metrics	8
8.1	Function Coverage	8
8.2	Statement Coverage	8
8.3	Branch Coverage	9
8.4	Condition Coverage	9
9	Appendix	9
9.1	Symbolic Parameters	9
9.2	Usability Survey Questions	9

List of Tables

1	Revision History	ii
2	Trace Back to Requirements	7
3	Trace Back to Modules	8
4	Table of Symbolic Parameters	9

List of Figures

Table 1: **Revision History**

Date	Version	Notes
Dec 7th	1.0	Revision 1

This document describes the implementation of the [MacSidenotes Test Plan](#). It covers: the testing of Functional and Non-Functional Requirements, System and Unit Testing, and the tracing of Test cases to the modules and requirements.

1 Functional Requirements Evaluation

1.1 Sidebar - Typing Window

1. Sidebar Open

Type: Functional, Dynamic, Manual

Initial State: Default Chrome browser window.

Input: Extension icon click.

Expected Results: Sidebar will appear at the top right of the page.

Actual Results Match Expected Results: Yes

2. Sidebar Close-1

Type: Functional, Dynamic, Manual

Initial State: Sidebar appears at the top right of the page.

Input: Extension icon click.

Expected Results: Sidebar disappears leaving solely the browser page.

Actual Results Match Expected Results: Yes

3. Sidebar Close-2

Type: Functional, Dynamic, Manual

Initial State: Sidebar appears at the top right of the page.

Input: Click event on browser page.

Expected Results: Sidebar disappears leaving solely the browser page.

Actual Results Match Expected Results: Yes

1.2 Note Taking

Typing

1. Input 1

Type: Functional, Dynamic, Manual

Initial State: Blank text area.

Input: User keyboard input.

Expected Results: User keyboard input.

Actual Results Match Expected Results: Yes

1.3 Note Saving

Save Note

1. Save Note - 1

Type: Functional, Dynamic, Manual

Initial State: Text window with keyboard input.

Input: Click event on 'Save Note' button.

Expected Results: Confirmation Message 'Note Saved!'

Actual Results Match Expected Results: Yes

1.4 Master List

List Management

1. List View - Open

Type: Functional, Dynamic, Manual

Initial State: Sidebar with solely text area and buttons.

Input: Click event on 'List' button.

Expected Results: Master List shown below the contents of the Sidebar.

Actual Results Match Expected Results: Yes

2. List View - Closed

Type: Functional, Dynamic, Manual

Initial State: Sidebar with text area, buttons and Master List.

Input: Click event on 'List' button.

Expected Results: Master List disappears from the Sidebar.

Actual Results Match Expected Results: Yes

1.5 Revisit URL

Link to Previous Notes

1. Link to Previous Note

Type: Functional, Dynamic, Manual

Initial State: Master List contains at least 1 note and is visible.

Input: Click event on a URL in Master List

Expected Results: New Chrome tab opens at that URL.

Actual Results Match Expected Results: Yes

1.6 Note Deletion

Delete Note

1. Delete Note

Type: Functional, Dynamic, Manual

Initial State: Text Area with a note saved to it.

Input: Click event on 'Delete Note' button.

Expected Results: Deletion confirmation message.

Actual Results Match Expected Results: Yes

2. Delete Confirm - YES

Type: Functional, Dynamic, Manual

Initial State: 'Delete Note' has been clicked prompting confirmation.

Input: Click event on 'YES' button.

Expected Results: Confirmation of deletion message.

Actual Results Match Expected Results: Yes

3. Delete Confirm - NO

Type: Functional, Dynamic, Manual

Initial State: 'Delete Note' has been clicked prompting confirmation.

Input: Click event on 'NO' button.

Expected Results: Note is not deleted.

Actual Results Match Expected Results: Yes

2 Nonfunctional Requirements Evaluation

2.1 Look and Feel

Look

1. Sidebar Size

Type: Manual, Dynamic

Initial State: Default Chrome browser window.

Input: Extension icon click.

Output: Sidebar popup.

Expected Results: Sidebar will appear at the top right of the page and does not take up more than SIDEBAR_WIDTH.

Actual Results Match Expected Results: Yes

2.2 Usability and Humanity Requirements

Usability & Humanity

1. Communication - English

Type: Manual, Static, Dynamic

Initial State: Varied

Input: N/A

Output: Messages in English

Expected Results: All text is understandable and in English

Actual Results Match Expected Results: Yes

2.3 Performance

Speed

1. Extension Response

Type: Manual, Dynamic

Initial State: Varied

Input: User input.

Output: Extension output.

Expected Results: Extension use is not too slow as to interrupt user's flow.

Actual Results Match Expected Results: Yes

3 Comparison to Existing Implementation

The original project [Sidenotes](#) is defunct and no longer works. Sidenotes relied on a Dropbox API that is now deprecated, and thus MacSidenotes can't be compared to the original implementation.

4 Automated Unit Testing

Due to the nature of the design, all unit tests are automated using Jasmine. Jasmine works by specifying different "specs" that include assertions made about the program's reactions to input. Certain input is fed to the program being tested, and Jasmine asserts what the outcome should be if the test is successful. The results are outputted to an html file where they can be viewed.

To view the results of the automated tests, download the git project and run `Doc/TestReport/jasmine/SpecRunner.html`

4.1 `clickCounter`

Type: Unit, Dynamic, Automated

Initial State: Varied

Input: Click event on 'Show List' button.

Expected Results: numClicks incremented

Actual Results Match Expected Results: Yes

4.2 `showList`

Type: Unit, Dynamic, Automated

Initial State: Master List not visible

Input: 2 Click events on 'Show List' button.

Expected Results: Master List appear, then disappear

Actual Results Match Expected Results: Yes

4.3 `emptyMasterList`

Type: Unit, Dynamic, Automated

Initial State: Master List has no items

Input: Add items to Master List, then call `emptyMasterList()`

Expected Results: Remove all rows from Master List

Actual Results Match Expected Results: Yes

4.4 `showNotice`

Type: Unit, Dynamic, Automated

Initial State: saveNotice and deleteNotice icons not visible
Input: 2 Click events on 'Save Note' and 'Delete Note" buttons.
Expected Results: saveNotice and deleteNotice icons appear
Actual Results Match Expected Results: Yes

5 Changes Due to Testing

The most significant changes to the extension were made after completing Usability Testing. Users suggested that the "http://www." be removed from the URL to increase the aesthetic appeal of the UI.

Users also showed interest in a 'flat' design to the UI buttons and a more cohesive interface. These ideas were incorporated into the design of the extension.

6 Trace to Requirements

Table 2: Trace Back to Requirements

Test #	Requirement #
1.1	F.1
1.2	F.2
1.3	F.3
1.4	F.4
1.5	F.5
1.6	F.6
2.1	NF.2
2.1	NF.5
2.1	NF.6

7 Trace to Modules

Table 3: Trace Back to Modules

Test #	Module #
1.1	M8
1.2	M9
1.3	M2
1.4	M3
1.5	M3
1.6	M2 & M6
2.1	M8
2.1	M8
2.1	M1
4.1	M3
4.2	M3
4.3	M4
4.4	M6 & M8

8 Code Coverage Metrics

8.1 Function Coverage

With the exception of 2 functions that are only used for debugging purposes, every function implemented in the extension is called at least once.

8.2 Statement Coverage

Not counting the statements within the 2 aforementioned debugging functions, every statement in the extension is executed.

8.3 Branch Coverage

All of the branches that are integral to the core functionality of the extension are covered. The only ones that are not covered are within the aforementioned debugging functions or would not break the extension if they malfunctioned (eg. text truncating).

8.4 Condition Coverage

Due to the lack of total Branch Coverage (as addressed above), total Condition Coverage is not achieved. The most important boolean sub-expressions are evaluated to both true and false, though.

9 Appendix

9.1 Symbolic Parameters

The definition of the test cases will call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

Table 4: Table of Symbolic Parameters

Symbolic Constants	Value
RESPONSE_TIME	The maximum amount of time the system has to respond to a user interaction. This number is 2 seconds.
SIDEBAR_WIDTH	The maximum width that the Sidebar should be upon first opened. The maximum sidebar width is 30% of a users full browser size.

9.2 Usability Survey Questions

During October of 2016 Matthew asked his two roommates about the usability of the extension. At this point the product was at the proof of concept stage.

Some of the questions asked were:

What do you like about the product?

What do you think could be improved?

What features would you like to see added to the product?

Would you use the product?

Both roommates liked the product as-is due to the facility in which it can be used in tandem with it's practicality. Some suggestion put forward included:

Bullet Points

Font Colours/Sizes/Effects

Highlights

The ability to add pictures

Notification if page has note on it already

Titles for notes

Overall the meeting was very successful and produced many great ideas that could find their way onto the final product. Both roommates said they would definitely use the product.