Anonymouth

Acceptance Test Plan

V1.5

Armon Entezari
Karan Hansaria
Chris McGuire
Jon McGrath
Paul Vu

Advisor: Rachel Greenstadt

Table of Contents

1. Introduction

1.1. Background

The purpose of this document is to describe the tests necessary for insurance that Anonymouth works properly. Properly means Anonymouth has the functionality it detailed in its “Functional Requirements”. The SRS of Anonymouth has a “Functional Requirements” section.

1.2. Structure of the Document

This document is divided into six different sections:

- Section 2 (Test Approach) – Describes how testing will be performed on Anonymouth
- Section 3 (Test Assumptions and Exclusions) – Lists all the previous assumptions and items that will not be covered with this test plan
- Section 4 (Entry and Exit Criteria) – What must be satisfied before and after running the tests
- Section 5 (Testing Participants) – The responsibilities each member of the testing team will complete
- Section 6 (Test Cases) – Individual test cases done on Anonymouth

1.3. References

[1] - Button, A. Mouse. "What Is Graphical User Interface? - A Word Definition From the Webopedia Computer Dictionary." *Webopedia: Online Computer Dictionary for Computer and Internet Terms and Definitions*. Web. 24 Nov. 2010. <http://www.webopedia.com/TERM/G/Graphical_User_Interface_GUI.html>.

[2] - "Unit Testing." *MSDN | Microsoft Development, Subscriptions, Resources, and More*. Web. 28 Nov. 2010. <http://msdn.microsoft.com/en-us/library/aa292197(VS.71).aspx>.

[3] - "What Is Integration Testing? - Definition from Whatis.com." *SearchSoftwareQuality.com*. TechTarget. Web. 28 Nov. 2010. <http://searchsoftwarequality.techtarget.com/definition/integration-testing>.

[4] - "corpus." *Dictionary.com Unabridged*. Random House, Inc. 02 Dec. 2010. <Dictionary.com http://dictionary.reference.com/browse/corpus>.

Also referenced are items from the requirements document of Anonymouth.

1.4. Glossary

- Test Team Lead – The person responsible to make sure the Acceptance Test conforms to the specifications listed in this document
- Testers – People who run the individual test cases
- Clientele – Users and advisors to Anonymouth
- Unit Test - Take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect [2].
- System Test – Test which tests the software as a whole for a match between its requirements and how it behaves in certain situations. It is done after all other tests.
- Integration Test - A software development process which program units are combined and tested as groups in multiple ways [3]. It is done after Unit Testing
- Functional Requirements – The internal "organs" of software and how they work
- Nonfunctional Requirements – Non-internal requirements for software
- Requirements Document (SRS) – Encompasses every aspect of software
- Graphical User Interface (GUI) – Acronym for Graphical User Interface. It is a program interface that takes advantage of the computer's graphics capabilities to make the program easier to use [1].
- Corpus - A large or complete collection of writings [4]

2. Test Approach

A simple route to testing would just do the tests required and ensure the output from those tests were the expected output. However, a very detailed plan for the tests is essential for the optimal results from testing. This plan includes agreements between the developer and client, a checklist to validate every tester completed their testing, and giving the results to the client.

The Unit, Integration, and System Tests precede the Acceptance Test Plan. It will not be as rigorous as those tests and they must be complete to the satisfaction of the clientele before

the start of the Acceptance Test Plan. The Acceptance Test Plan essentially tests every requirement explained in the SRS.

3. Test Assumptions and Exclusions

   3.1. Assumptions

   - The other tests (Unit, Integration, and System) completed without problems
   - The latest version of Anonymouth is used for testing
   - The non-functional requirements such as the installation of Python and Stylo are already done
   - Anonymouth's latest software documentation has been obtained

   3.2. Exclusions

   - Non-functional requirements in Anonymouth's requirements document
   - Performance, architecture design, and usability of Anonymouth

4. Entry and Exit Criteria

   4.1. Entry Criteria

   - Assumptions (3.1)
   - Agreement on a start date and time between the developer and client
   - Notification upon start from the team test leader to our clientele and testers

   4.2. Exit Criteria

   - The tested functional requirements have the correct post condition(s) -> Success
   - Any successful test case shall be reported to the appropriate people listed in (5.4)
   - The tested functional requirements do not have the correct post condition(s) -> Failure
   - Any failed test case shall be reported to the appropriate people listed in (5.3) and rescheduled for a later date after the action has been fixed

5. Testing Participants

   5.1. Roles and Responsibilities

   - Test Team Lead: Armon Entezari
   - Testers: Paul Vu, Jon McGrath, Chris McGuire, Karan Hansaria
   - Clientele: Dr. Rachel Grennstadt, Michael Brennan, Sadia Afroz

   5.2. Training Requirements

Testers involved with the Acceptance Test Plan should have experience using the GUI of Anonymouth. As well, they must have read Anonymouth's published software documentation located on their website.

5.3. Problem Reporting

Upon a problem encountered during testing, the tester should note it. Then, when all the test cases are finished, a report will be drafted with the test cases that failed with a possible reason for why it might have failed. This report will be sent to all testing participants.

5.4. Successful Cases Reporting

Test cases that succeed are given a "checkmark" for completion. These will be sent as a report to all testing participants noting that the test cases completed without problems.

6. Test Cases

Name: Call Stylo (2.1, 0010)

Precondition: A writing sample will be in the document buffer

Action: Call the Stylo tool from our tool

Post condition: Stylo returns a list of features (2.1, 0020)

Name: Stylo will return a list of features (2.1, 0020)

Precondition: Stylo will be called (2.1, 0010)

Action: Receive the list of features

Post condition: Assure the list of features matches those in their requirements document

---

Name: Rate feature (2.1, 0021)

Precondition: Features will be returned (2.1, 0020)

Action: Rate a certain feature and its importance in identifying an author

Post condition: N/A

Name: Application startup (2.2.1, 0050)

Precondition: N/A

Action: N/A

Post condition: Main screen loads alone

Name: Blank windows (2.2.1, 0060)

Precondition: Application startup (2.2.1, 0050)

Action: All windows within the application will be blank

Post condition: N/A

Name: Loaded project (2.2.1, 0070)

Precondition: A project will be loaded from the project wizard or creating a new one in the buffer

Action: All windows within the application will not be blank

Post condition: N/A

Name: Correct percentages (2.2.1.1, 0110)

Precondition: A document has been submitted

Action: The percentages of likeliness are integers between 0 and 100

Post condition: N/A

Name: Loaded project (2.2.1.1, 0113)

Precondition: The submit button has not been pressed

Action: Ensure that the “current” and “new” percentages are grayed out

Post condition: N/A

Name: Select feature (2.2.1.2, 0160)

Precondition: Project exists and has been submitted at least once before

Action: Click on a feature in the features/ suggestions list

Post condition: Text area focuses on feature

-Text corresponding to feature is highlighted

-Tree expands to show suggestions

Name: Select suggestion (2.2.1.2, 0170)

Precondition: Feature in feature list has been selected

Action: Click on a suggestion in the features/ suggestions list

Post condition: Text area focuses on feature

-Text corresponding to feature is highlighted

Name: Open suggestion submenu (2.2.1.2, 0190)

Precondition: Suggestion in feature list has been selected

Action: Right-click on a suggestion in the features/ suggestions list

Post condition: Sub-menu opens revealing the three choices: replace, compare, ignore

---

Name: Select ‘replace’ from suggestion submenu (2.2.1.2, 0200)

Precondition: Suggestion submenu is open

Action: Click on ‘replace’ in the submenu

Post condition: Text area updates with replacement of selected text

-Percentage likeness adjusts accordingly underneath text area

---

Name: Select ‘compare’ from suggestion submenu (2.2.1.2, 0210)

Precondition: Suggestion submenu is open

Action: Click on ‘compare’ in the submenu

Post condition: Percentage likeness adjusts accordingly underneath text area

-Clicking anywhere else will reset the percentage to its pre-compare value

---

Name: Select ‘ignore’ from suggestion submenu (2.2.1.2, 0220)

Precondition: Suggestion submenu is open

Action: Click on ‘ignore’ in the submenu

Post condition: Suggestion is removed from the suggestions list

---

Name: Select ‘undo ignore’ from suggestion area (2.2.1.2, 0240)

Precondition: Right click in suggestion area

Action: Click on ‘undo ignore’ in the submenu

Post condition: The suggestion previously ignored will be redisplayed in the suggestion area

---

Name: Load previous version (2.2.1.3, 0280)

Precondition: Text has been submit for review more than once

Action: Click on item in the history area

Post condition: Text area updates with previous version

-Suggestions update with previous version

-Percentages update with previous version

Name: Submit piece (2.2.1.4, 0300)

Precondition: Suggestion(s) have been generated by our tool

Action: Click on a color in the toolbox

Post condition: Only the suggestion of that color clicked will be displayed in the suggestion area

---

Name: Default filter (2.2.1.4, 0310)

Precondition: Suggestion(s) have been generated by our tool

Action: N/A

Post condition: All the suggestions are displayed in the suggestion area

---

Name: Move legend toolbox (2.2.1.4, 0320)

Precondition: N/A

Action: The legend toolbox will move when it is dragged

Post condition: The legend toolbox will move to the desired position of the user

---

Name: Turn off legend toolbox (2.2.1.4, 0330)

Precondition: N/A

Action: From the Windows menu on the toolbar, the user will turn off the legend toolbox

Post condition: The legend toolbox no longer appears

Name: Submit piece (2.2.1.5, 0350)

Precondition: Text area is not empty

Action: Click on ‘submit’ button

Post condition: Piece is submit to Stylo for evaluation

-Percentages update to reflect evaluation

-Features list filled out

-History item added

Name: Select ‘Wizard’ from new project popup (2.2.2, 0380)

Precondition: New project selected

Action: Click on ‘Wizard’ option in the new project popup

Post condition: First step of the Wizard appears

Name: Click “Yes” (2.2.2, 0390)

Precondition: Project Wizard selected

Action: Click on “Yes” option when asked if they will like to import a document

Post condition: File browser window opens

Name: Click “Cancel” (2.2.2, 0395)

Precondition: User selects to import a document

Action: Click on “Cancel” for the file browser window

Post condition: Application returns to the import document window

---

Name: Click “Open” (2.2.2, 0395)

Precondition: No file has been selected in the file browser window

Action: Click on “Open”

Post condition: Application returns to the import document window

---

Name: Click “Open” (2.2.2, 0396)

Precondition: File has been selected in the file browser window

Action: Click on “Open”

Post condition: Application returns to the project wizard and the file is displayed in the document area

---

Name: Click “No” (2.2.2, 0397)

Precondition: Project Wizard selected

Action: Click on “No” to import a document

Post condition: Text area of application remains blank and the corpus selection window will be opened

---

Name: Click red “x” (2.2.2, 0420)

Precondition: At least one document has been added to the corpus

Action: Click on a red “x” for an individual document

Post condition: Document will be removed from the corpus

Name: Save project (2.2.2, 0430)

Precondition: Corpus documents have been choosen

Action: User wished to save the project to a file

Post condition: The project file has been saved in the correct format to the correct destination on the hard disk

---

Name: Click “Okay” (2.2.2, 0440)

Precondition: Project saved to a file

Action: Click “Okay”

Post condition: Application returns to the main screen

Name: New project (2.2.3, 0480)

Precondition: New Project selected from File menu

Action: Click on “New Project” in File menu

Post condition: Windows load empty

-Option to use wizard available

-Options relating to project that were disabled are now enabled

---

Name: Open project (2.2.3, 0510)

Precondition: Open Project selected from File menu

Action: Click on “Open Project” in File menu

Post condition: Windows load filled with its saved state

-Options relating to project that were disabled are now enabled

---

Name: Save project (2.2.3, 0530)

Precondition: Project exists in current session

Action: Click on ‘Save project’ in the File menu

Post condition: Popup dialog to select file from explorer

-Saves information to file on disk

Name: Click “Save/Save As” (2.2.3, 0541)

Precondition: User clicks “Save/Save As” from File menu and saves the file to a file that already exists

Action: Prompt the user if they wish to overwrite the existing project file

Post condition: Application returns to the import document window

---

Name: List most recent project files (2.2.3, 0550)

Precondition: Users hovers over Recent Projects in the File menu

Action: Hover over Recent Projects

Post condition: A maximum of five recent project files are displayed in a tab to the immediate right of the Recent Projects selection

---

Name: Export project (2.2.3, 0560)

Precondition: Project exists in current session

Action: Click on ‘Export project’ in the File menu

Post condition: Popup dialog to select file from explorer

-Save file as user’s desired text format

Name: Close project (2.2.3, 0570)

Precondition: N/A

Action: Click on ‘Close’ in the File menu

Post condition: Popup dialog to select file from explorer (if not recently saved)

-Close program

Name: Undo step (2.2.4, 0600)

Precondition: Project exists, has been modified

Action: Click on ‘Undo’ in the Edit menu

Post condition: Revert change

Name: Redo step (2.2.4, 0610)

Precondition: Project exists, has been modified and user has performed an undo

Action: Click on ‘Redo’ in the Edit menu

Post condition: Recommit change

Name: Copy (2.2.4, 0620)

Precondition: Project exists, text selected

Action: Click on ‘Copy’ in the Edit menu

Post condition: Copy text and copy to clipboard

Name: Cut (2.2.4, 0630)

Precondition: Project exists, text selected

Action: Click on ‘Cut’ in the Edit menu

Post condition: Delete text and copy to clipboard

Name: Paste (2.2.4, 0640)

Precondition: Project exists, text selected or cursor has position

Action: Click on ‘Paste’ in the Edit menu

Post condition: Paste text and leave in clipboard

Name: Find (2.2.4, 0651)

Precondition: Find has been selected from the Edit menu

Action: Click on ‘Find’ from the find window

Post condition: The first occurrence of whatever will be in the find area will be highlighted in the document area

---

Name: Manage Corpus (2.2.5, 0680)

Precondition: Project exists

Action: Click on ‘Manage corpus’ in the Tools menu

Post condition: Popup window to manage corpus with add, remove

---

Name: Click ‘x’ (2.2.5, 0682)

Precondition: The manage corpus window has been opened

Action: Click on a ‘x’ next to a document listed in the window

Post condition: Document will be removed from the corpus

Name: Click ‘Add Document’ (2.2.5, 0683)

Precondition: The manage corpus window has been opened

Action: Click on ‘Add Document’ button in the window

Post condition: User selects a file they will add to the corpus

---

Name: Click Tutorial (2.2.7, 0730)

Precondition: The help menu has been selected

Action: Click on ‘Tutorial’

Post condition: A window with a step-by-step guide on using Anonymouth will be displayed

---

Name: Click Contents (2.2.7, 0740)

Precondition: The help menu has been selected

Action: Click on ‘Contents’

Post condition: A help document with frequently asked questions will be displayed

---