**Autocomplete Feature – Test Plan**

Trunk Club Assignment

Date: 10th Feb 2019

Lakshmy Balachandran

lakshmibalachandran87@gmail.com

Contents

[Introduction 2](#_Toc688204)

[Tools 2](#_Toc688205)

[Glossary of terms 2](#_Toc688206)

[Test Cases for identified Scenarios 3](#_Toc688207)

[Scenario 1: API Validation 3](#_Toc688208)

[Scenario 2: UI Validation 6](#_Toc688209)

[Scenario 3: Performance Testing 10](#_Toc688210)

[Scenario 4: Load Testing 11](#_Toc688211)

[Automation 11](#_Toc688212)

# Introduction

This document provides a high-level summary of test considerations for the Autocomplete feature in <www.amazon.com>. The scope of this test plan includes the following validation/ testing scenarios:

* API Validation
* UI Validation
* Performance Testing (non-functional)
* Load Testing (non-functional)

# Tools

List of tools to be used for automating the above validation scenarios:

|  |  |
| --- | --- |
| API validation | Postman |
| UI validation | Eclipse, Selenium, Java, TestNG |
| Performance Testing | JMeter |
| Load Testing | JMeter |

# Cof terms

Below is a glossary of terms used in the test cases that follows, as it applies to the application www.amazon.com that is being tested:

|  |  |
| --- | --- |
| Search field | The product search field that is used for testing the Autocomplete feature |
| Keyword | The keyword entered in the search field to yield Autocomplete suggestions |
| Suggestions | List of product suggestions provided in the Autocomplete dropdown as a keyword is entered in the search field |
| Department Dropdown | Dropdown list located immediate left to the Autocomplete text box for selecting the product departments |
| Submit Button | Button located immediate right to the Autocomplete text box for submitting the keyword search |
| Fuzzy Matching | Technique that suggest correct matches for a keyword even if minor spelling errors are found in the keyword. (Level of Fuzzy Matching differs and it should be identified from the business requirements of the application) |

# Test Cases for identified Scenarios

## Scenario 1: API Validation

|  |  |
| --- | --- |
| **Test Case #1:** Verify maximum display limit of Autocomplete suggestions list.   * In case of amazon.com, verify that autocomplete suggestions is kept to a maximum of 10 suggestions, even when more than 10 suggestionsexist * Note: If keyword starts with ‘#’, the max limit for suggestions seem to be 3 instead of 10 | Postman |
| **Test Case #2:** Verify the accuracy of suggestions provided in Autocomplete list   * Suggestions for keyword “b” should include only strings starting with “b”. * Suggestions for keyword “bath” should include only strings starting with “bath” | Postman |
| **Test Case #3:** Verify that distinct suggestions are returned by Autocomplete, with no duplicate suggestions returned   * Suggestions for keyword “bath” should have distinct suggestions, all starting with “bath”. * For example, “bath toys” or “bath tub” should not occur more than once, when keyword is “bath”. | Postman |
| **Test Case #4:** Verify that Autocomplete suggestions are not case-sensitive to the keyword used   * Suggestions for keyword “bath”, “Bath” or “BATH” should all be same | Postman |
| **Test Case# 5:** Verify that Autocomplete reacts as per the business requirements when keywords start with a special character   * Autocomplete ignores special characters such as $, @, % etc., when keywords start with that special character * Note that special character ‘#’ alone is treated differently for amazon.com. | Postman |
| **Test Case#6:** Verify the suggestions List Retrieval Pattern in the Autocompletion Feature   * Identify the suggestion lists follows the design algorithm. * Suggestion provided for letter “N” will show the suggestion list based on the pattern algorithm mentioned in the business requirements (for example, the suggestions list will be shown based on ascending order sorting or most rated items etc.) | Postman |
| **Test Case# 7:** Verify the behavior of suggestions list of the Autocompletion Feature after the addition of a new product to the database   * Identify the autocomplete feature behavior after a new product addition into the database * Add value “ZAA” to the database. Make sure that this is the first entry with ZAA * Suggestions provided for word “ZAA” should also contain the newly added value | Postman |
| **Test Case# 7:** Verify the behavior of suggestions list of the Autocompletion Feature after the deletion of an existing product from the database   * Identify the autocomplete feature behavior after the deletion an existing product in the database * Delete product name “ZAA” from the database. * Suggestions provided for word “ZAA” should not contain the recently deleted product name | Postman |
| **Special Feature in Amazon:** Please note I have identified “FuzzyMatching” technique in the Amazon Autocomplete Feature  **Test Case# 8:** Verification of FuzzyMatching algorithm in Autocomplete Feature   * Suggestions provided for misspelled word “Computor” should be the same as those provided for the original word “Computer” | Postman |
| **Test Case# 9:** Verify no missing data in the suggestion list of Autocomplete Feature   * The Autocomplete Feature should retrieve all possible matching suggestions (Assuming, maximum number of suggestions for the Autocomplete suggestion list is equal to 10) * Make sure that there is no Autocomplete suggestion list for word “Z1” under the Baby Department. Add “Five” products to the Product Database with names Z11, Z12, Z13, Z14, Z15 under the Baby Department * Suggestions provided for “Z1” should retrieve all values including Z11 – Z15 | Postman |
| **Test Case# 10:** Verify the department details is existing for the First AutoComplete Suggestion   * The suggestion list should show the product department details only for the first suggestion * For example, the keyword “Shoe” should retrieve “shoe rack” as first product suggestion with department details “In Home & Kitchen” along with other product suggestions | Postman |
| **Test Case# 11:** Verify the suggestion list retrieval for multiple keyword concatenation   * Suggestions for keyword “bath + tub” should limit number of suggestions to “three” in the suggestion list * Suggestions for keyword “bath + tub” should retrieve suggestions which have both keywords “bath” and “tub” | Postman |

## Scenario 2: UI Validation

|  |  |
| --- | --- |
| **Test Case #1:** Verify that Autocomplete suggestions are displayed in the webpage   * Entering any keyword in search field should provide a dropdown of relevant Autocomplete suggestions | Eclipse, Selenium, Java, TestNG |
| **Test Case #2:** Verify that user is able to use mouse to select an item from the Autocomplete suggestions   * Enter a keyword “computer” and user should be able to select the item “computer chair” from the list of suggestions using the mouse. * Expected result is user is redirected to search results page for “computer chair” | Eclipse, Selenium, Java, TestNG |
| **Test Case #3:** Verify that user is able to use keyboard and navigate to an item from the Autocomplete suggestions   * Enter a keyword “computer” and user should be able to navigate to the suggestion “computer chair” using ‘up’ and ‘down’ arrow keys. * Expected result is the item “computer chair” is highlighted as user navigates to that item. | Eclipse, Selenium, Java, TestNG |
| **Test Case #4:** Verify that user is able to use keyboard to select an item from the Autocomplete suggestions   * Enter a keyword “computer” and navigate to the suggestion “computer chair” using ‘up’ and ‘down’ arrow keys and hit ‘Enter. * Expected result is user is redirected to search results page for “computer chair” | Eclipse, Selenium, Java, TestNG |
| **Test Case #5:** Verify Autocomplete suggestions are department dependent   * If a non-existing keyword is searched in a selected department, then there will be no suggestions shown even though the product may exist in the other department * For example, select “Alexa Skills” from the Department Dropdown and search for keyword “ZX” in the search field. The expected result should not show any Autocomplete suggestions as there are no such products under the “Alexa Skills” department product database | Eclipse, Selenium, Java, TestNG |
| **Test Case #6:** Validate the Autocomplete Feature behavior when no suggestions selected from the suggestion list  - Without selecting anything from the autocomplete suggestion list for a keyword “lamp”, click on the submit button. The user should be able to see the results for “lamp” | Eclipse, Selenium, Java, TestNG |
| **Test Case #7:** Validate the Autocomplete Feature behavior when no keyword is entered in the Autocomplete search field  - Select All Departments and click on submit button without entering anything in the Autocomplete Feature textbox. The program should behave as per the given business requirements | Eclipse, Selenium, Java, TestNG |
| **Test Case# 8:** Verification of enter -removal - reenter behavior of Autocomplete Feature  - Identify the behavior of Autocomplete Feature when a keyword is typed, removed without selecting anything in the suggestion list and retype a new keyword  -For example, first enter a keyword “lamp” into the autocomplete search field. Without selecting anything from the autocomplete suggestion list, remove the keyword and retype a new keyword “Toy” to the search field. The user should be able to see the results of most recently selected value from the autocompletion list. | Eclipse, Selenium, Java, TestNG |
| **Test Case# 9:** Validation of Copy $ Paste functionality in the search field of the Autocomplete Feature  -The search field should be able to accept keywords through copy- paste and retrieve suggestions list based on that | Eclipse, Selenium, Java, TestNG |
| **Test Case#10:** Verify the validity of suggestions in Autocomplete Feature  -Make sure that all the suggestions in the autocomplete list results a valid search result  -For example, all the 10 suggestions for the keyword “tomato” should retrieve valid search results | Eclipse, Selenium, Java, TestNG |

## Scenario 3: Performance Testing

|  |  |
| --- | --- |
| **Scenario#17:** Performance Test after the Addition of Autocomplete Feature  **Description:** Conduct performance testing in various browsers/ OS in order to make sure that there the no performance degradation after the addition of the new Autocomplete Selection Feature  **Example Test Case# 19:** Verify the Performance Benchmark | Performance test with JMeter |

## Scenario 4: Load Testing

|  |  |
| --- | --- |
| **Scenario# 19** Load Test the pages with new Autocomplete selection Feature  **Description:** Conduct Load testing in order to make sure that the system can handle expected amount of load addition of the new Autocomplete Selection Feature  **Example Test Case# 20:** Verify the Load test with different number of users | Load test with JMeter |

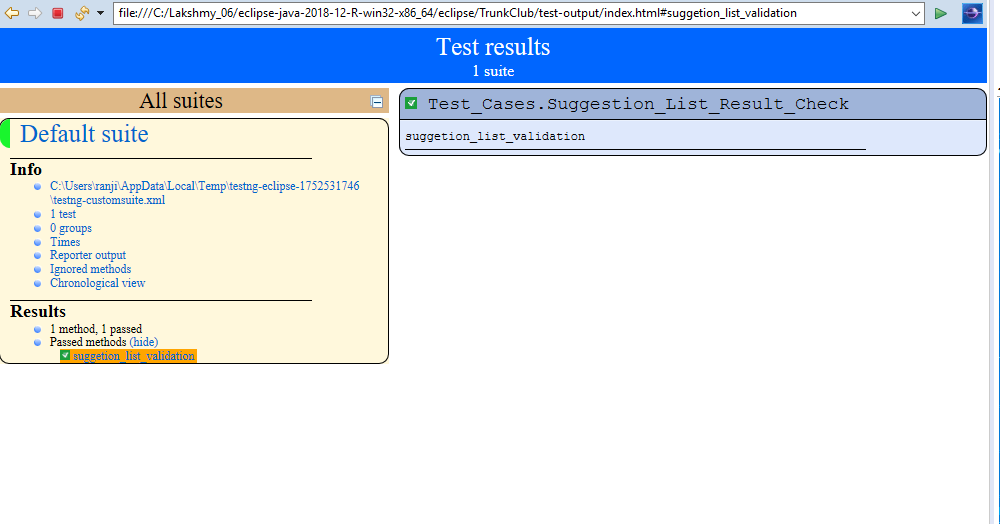
# Automation

Please provide a link to a GitHub repository containing your test automation. Please include compilation and execution instructions in the readme.

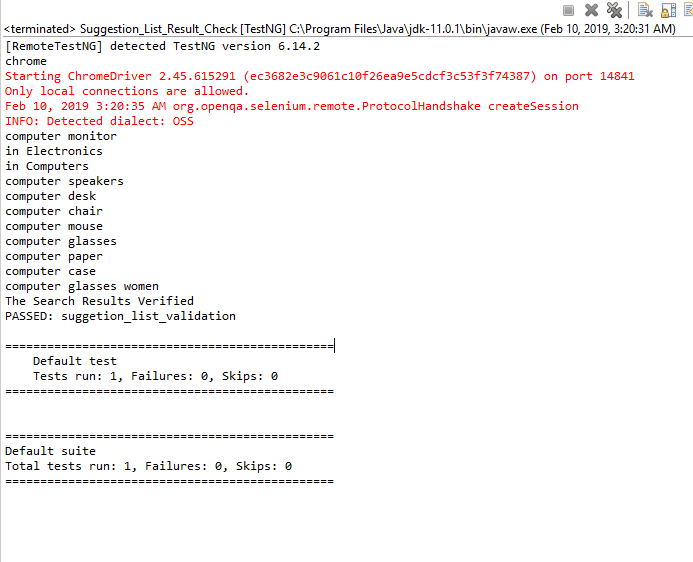
Below is the selected scenario for automation.

|  |  |
| --- | --- |
| **Test Case #2:** Verify that user is able to use mouse to select an item from the Autocomplete suggestions   * Enter a word “COMPUTER” into the text field with Autocomplete feature * **Expected Result:** Suggestions provided by the AutoComplete feature should start with “Computer” * Click on one of the suggestions in the list using mouse pointer * **Expected Result:** Make sure that the user is directed the selected item page and able to verify the matching results | UI Automation with Eclipse, Selenium, Java, TestNG |

## Test Report:



**Console Output:**

****