Ekta Dosad

Automation of end-to-end user e-commerce transaction flow
GitHub URL: https://github.com/ektadosad/Walmart-Transaction_flow.git

Reasoning behind technical choice- Page Factory Design Pattern:
Disadvantage of Selenium test cases is that it leads to an unmaintainable project because of the delicacy (duplicated usage of locators). We have to walk through the whole test code to adjust locators if it is changed, which is time a time consuming process. A better approach to script maintenance can be achieved by separating the abstraction of test objects and test scripts.

I chose Page Factory design pattern for this project because it provides us an ability to reuse the Page Factory Web elements across different classes/methods while reducing multiple findElement calls. Moreover, this approach helped me in reducing code-duplicity from my functional test cases and enabled me to write modular test cases in much optimized manner to conduct testing of various web elements.

In a nut-shell, because of following advantages over other approaches, I chose Page Factory Design Pattern:
- Readability of scripts.
- Reduce or Eliminate duplication.
- Easy to Maintain.
- Re-usability of code to implement more tests.
- Can be used in any kind of framework such as Data Driven, Keyword or Modular Driven.

Technology:
- **Testing tool:** Selenium Web Driver
- **Programming language:** Java

Framework:
- TestNG
- ReportNG

Design Pattern:
- Page Factory Design Pattern

Coverage:
- Functional Tests
- Appearance/User Experience Testing

Software required:
- Eclipse IDE, TestNG Eclipse plug-in, ReportNG

This approach covers all the 5 scenarios given in the requirement document. I might have used data driven or keyword driven approach to run the test case multiple times for different users by reading log in details and item keywords from excel.
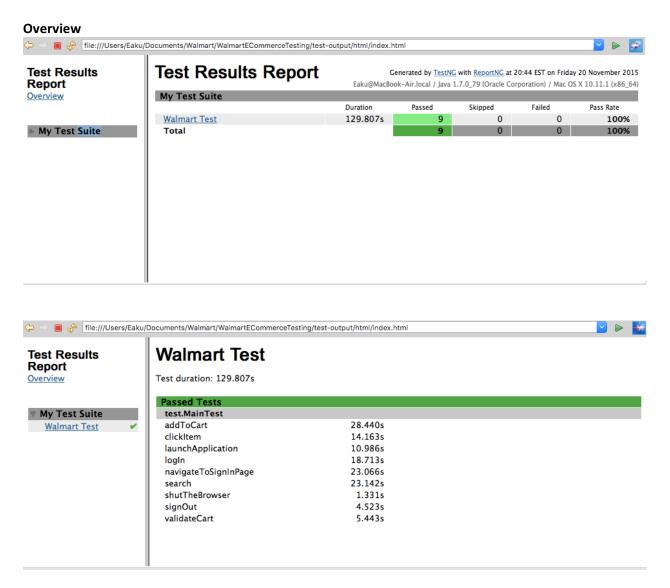
Instructions for running the code:
- Open project properties
- Add TestNG and ReportNG Jar Files to the project.
- Disable default TestNG listeners.

Ekta Dosad

- Open src folder and run testing.xml as TestNG Suit.
- Open index.html (under test output) in browser to see test report.

Screenshot of HTML report through ReportNG:

Overview

Test Results Report
Overview

Test Results Report

Generated by TestNG with ReportNG at 20:44 EST on Friday 20 November 2015
Eaku@MacBook–Air.local / Java 1.7.0_79 (Oracle Corporation) / Mac OS X 10.11.1 (x86_64)

My Test Suite

	Duration	Passed	Skipped	Failed	Pass Rate
Walmart Test	129.807s	9	0	0	100%
Total		9	0	0	100%

▶ **My Test Suite**

file:///Users/Eaku/Documents/Walmart/WalmartECommerceTesting/test-output/html/index.html

Test Results Report
Overview

Walmart Test

Test duration: 129.807s

▼ **My Test Suite**
  Walmart Test ✔

Passed Tests

test.MainTest

addToCart	28.440s
clickItem	14.163s
launchApplication	10.986s
logIn	18.713s
navigateToSignInPage	23.066s
search	23.142s
shutTheBrowser	1.331s
signOut	4.523s
validateCart	5.443s

Note: Regarding requirements that are not listed in the homework assignment:

- Negative scenarios: I noticed that few items have option of selecting size and color. If the item is not available in particular or size, then either we can search for different item or can fail the test as "the item is out of stock". So, even after implement size and color selection option, there is a possibility that the test will fail (if item is out of stock).

     Right now, I am choosing items at random. As mentioned by the Hiring Manager. These scenarios are being ignored for now and only items that have quantity are added (as quantity web element is present for all the items). I have implemented selection of quantity (no. of items you want to add in your cart). The requirement of color and size selection can be implemented in the same way if needed.

Ekta Dosad

- As mentioned by the hiring manager, I am randomly selecting an item from the result set (by selecting location of random items). This kind of selection can not be linked to a particular item requirement (As we don't know what item will be present at that location). Due to frequent changes in the site, locator of elements can change and selenium web driver can not locate the element if its locator has changed.

I could have written better tests if above requirements were mentioned clearly in the homework assignment.

- **Issue with the site:** The site takes a lot of time to load, which leads to timeout and causes the test to fail with "Unable to find element with id…" error.

Solution: I have implemented explicit wait and have given 35 sec wait. But if the site takes more than 35 sec and the test fails.