SE 2226 Test Completion Report Template

# (Based on ISO/IEC/IEEE 2G11G-2 Section 7.4)

# *Rüzgar Erdem, Şehran Kartal*

# *Can Berk Soydan*

# Summary of testing performed

# — Test specifications were made, it covered 3 areas and amounted to 60 tests.

# — Test environment was established.

# — Tests were executed and results recorded.

# Deviations from planned testing

# It was not possible to perform load and penetration testing due to legal ambiguity.

# The tests were planned to be done entirely on Selenium IDE but it was realized that some parts would be done more efficiently using Selenium WebDriver scripts.

# Tests using Selenium WebDriver were found to be slower than desired and thus planned to be ported to another library but that was not possible due to time constraints.

# Tests which required sales amounts (such as sorting items by most bought) were not possible due to sale data being inaccessible without API keys or admin privileges.

# Test completion evaluation

# 13 out of the 60 test procedures executed did not pass testing. 8 of these failed tests were of the sorting functionality of the site while 5 of them were of the item filtering system.

# 5 of these tests were all related to the descending rating sorting. These tests are not pointers to a significant defect as we assume that the site uses a black box algorithm for the rating sort which considers review amounts, recent sales and other factors.

# Another 3 were related to a specific item: phones. Whilst other items like bags and personal computers did not show any problems with any functions tested phones always had one or two outliers in the data. We believe this may be due to a competing algorithm which weighs specific items more heavily either due to a defect or as a form of advertisement.

# The remaining 5 tests were related to item filters and were the cause of filter traits (such as item types) not being present in the titles of the items. This was a technical problem on the project’s part because it would be needed to open the links of every item to be able to check their descriptions and find the traits mentioned which would significantly slow down test run times. As such this is not assumed to be a defect.

# Tests pertaining to sales amounts (such as sorting items by most bought) were not possible as the sales data for items is inaccessible from outside the server.

# Black Box tests such as login sequences and add-to-card features were also tested and have all passed successfully. The use cases for these tests will be provided alongside the report. The tests for adding cards to the system have not been executed because of the inability to add fake cards.

# Factors that blocked progress

# Selenium IDE can have difficulties finding and clicking on specific elements of a webpage. It can also have problems typing into textboxes with limited characters. The best solution to these problems was changing target elements or using different functions.

# Another problem was that the Selenium WebDriver used to write filtering and sorting functionality tests was slower than desired. We attempted to port the tests over to a different framework but that did not prove possible due to functional and time constraints.

# Test measures

# 60 tests procedures were executed throughout the project. 13 tests failed and 3 defects were detected. 5 tests cases could not be run due to inaccessible data (these being sorting functions for sales amounts). 2 additional test cases couldn’t be run due to the inability to add fake debit cards to the system.

# Duration of test design and execution:

# — One week was spent on the production of test cases and procedures.

# — Two weeks were spent on establishing the test environment, test execution and defect finding.

# — Three hours were spent on the production of this report.

# Test deliverables

# — Test Plan

# — Decision Table for Site Login Flows

# — 6 Use Case Specifications for Black Box Tests

# — BVA Report for Item Filters

# Lessons learned

# The testing process would’ve been faster if appropriate developer APIs were provided for basic web scraping operations and data requests for items. Fortunately most of these could be acquired through scraping but some data was unavailable due to site, framework and time restrictions.

# It might also be more preferrable to use more fundamental scraping tools in the future to acquire the data as Selenium WebDriver was too slow for the basic scraping done as it emulated user interaction which was unnecessary.