Walmart Item Page Test Plan

#### 

#### **Introduction**

Test Eng team will create and cover the following scenarios for Walmart Item page via selenium with Java

#### **Testing Coverage**

Below are listed test scenario which Test Eng team will cover:

1. Verify user is not able to order past max quantity and correct error is shown
2. Verify description of product minimizes/ maximizes
3. Verify adding to cart and switching pages, cart should be same as before
4. Verify content of Title, description, specifications
5. Verify when user scrolls down, reviews appear or are loading within reasonable time
6. Verify valid options are enabled for find in store (Ex. By postal code, information cross reference correctly)
7. Verify user is able to click related products and return to previous screen at the same spot the user had left
8. Verify quick links provided at bottom of page redirect to correct pages
9. Verify Add to cart shows pop up w/ option to continue shopping or checkout
10. Verify “free return” navigates user to “Return policy” section which will be expanded if not already
11. Verify user is able to minimize and maximize product links (Ex. specification, find in store etc…)
12. Verify changing postal code updates respective results
13. Verify user is able to search reviews and results render

#### **Features not to be tested**

1. Tablets, iPads will not be tested
2. Mobile Phones will not be tested
3. End to End purchase workflow will not be covered
4. Entering and exiting to search browsers

#### **Item pass/fail criteria**

Test Eng team will not pass with any open P0, P1, P2 bugs. P3/P4 bugs are acceptable at Product owners /Business discretion.

#### **Assumptions**

Assumption will be that environment is up and running, and all applicable links redirect to the correct page are all working

#### **Test deliverables**

Test deliverables are the documents that will be delivered by the testing team at the end of testing process. This may include test cases, sample data, test report, issue log.

#### **Environmental needs**

We will testing in Walmarts Prod environment (walmart.ca)

#### **Schedule**

Testing effort by Test Eng team will take approx 2-3 days

**Scenario 1: Get Title of Page**

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.chrome.ChromeDriver;

**public** **class** Demo {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

System.*setProperty*("webdriver.chrome.driver", "//Users//rubelsingh//Documents//chromedriver");

WebDriver driver=**new** ChromeDriver();

driver.get("https://www.walmart.ca/en/ip/intex-metal-frame-pool/6000166640889");

System.***out***.println(driver.getTitle());

}

}

**Scenario 2: See more products by “Intex Development Co.”**

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.UnexpectedAlertBehaviour;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.remote.CapabilityType;

**import** org.openqa.selenium.remote.DesiredCapabilities;

**public** **class** Demo {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

System.*setProperty*("webdriver.chrome.driver", "//Users//rubelsingh//Documents//chromedriver");

WebDriver driver=**new** ChromeDriver();

DesiredCapabilites cap= DesiredCapabilities.*chrome*();

cap.setCapability(CapabilityType.***ACCEPT\_SSL\_CERTS***,**true**);

WebDriver driver1 = **new** ChromeDriver(capabilities);

driver1.get("https://www.walmart.ca/en/ip/intex-metal-frame-pool/6000166640889");

System.***out***.println(driver1.getTitle());

driver1.findElement(By.*cssSelector*("css-1syn49.elkyjhv0")).click();

}

}

**Scenario 3:**