

BLG 475E: SOFTWARE QUALITY AND TESTING SPRING 2019-2020 HOMEWORK

3

REPORT

O. Kürşat Karayılan
150140011

In this homework we are asked to test given scenario by using java and selenium. First, I needed to describe to the project where my chromedriver file is, as seen below. The file is TestContext.java I edited in this case.

```
public Browser doCreateBrowser()
{
    // Optional, if not specified, WebDriver will search your path for chromedriver.
    System.setProperty("webdriver.chrome.driver", "/Users/kursat/Desktop/ders/2019 2020 bahar/BLG475E Test/hw3/migros/chromedriver");
}
```

Example Test

The example test in the project file we are given is Dana Yemeklik Kuşbaşı scenario. I edited the code since I skipped the login part.

```
public class BasketTest extends AbstractTest
{
    @Test
    public void testBasketItem()
    {
        //login(context.getInternalProps().getUsername(), context.getInternalProp
        //clearBasket();

        MainPage mainPage = new MainPage(browser);
        browser.waitForClick(mainPage.meatFishChickenMenu);
        String title1 = browser.getTitle();
        Assert.assertEquals("Et, Tavuk, Balık - Migros", title1);

        browser.waitForClick(mainPage.meatCategory);
        String title2 = browser.getTitle();
        Assert.assertEquals("Kırmızı Et - Migros", title2);

        browser.waitForClick(mainPage.meatKusbasi);
        String title3 = browser.getTitle();
        Assert.assertEquals("Dana Yemeklik Kuşbaşı Kg - Migros", title3);
    }
}
```

This code does the following steps.

- Opens migros.com.tr
- Clicks Et, Tavuk, Balık category
- Clicks Kırmızı Et
- Find Dana Yemeklik Kuşbaşı Kg item on the page then clicks

I checked in each page if title of the page corrects or not as you can see above.

```
@FindBy(linkText = "Et, Tavuk, Balık")
public WebElement meatFishChickenMenu;

@FindBy(css = "a[data-monitor-ga-action='Kırmızı Et']")
public WebElement meatCategory;

@FindBy(linkText = "Dana Yemeklik Kuşbaşı Kg")
public WebElement meatKusbasi;
```

- meatFishChickenMenu is used for clicking Et, Tavuk, Balık category. It works with linkText method.
- meatCategory is used for Kırmızı Et category. It looks for css of the links. If data-monitor-ga-action equals to Kırmızı Et, then it click that link.
- meatKusbasi finds Dana Yemeklik Kuşbaşı item and clicks it.

New Scenario

- It opens migros.com.tr
- Goes to Bebek, Oyuncak category
- Clicks Bebek Bezi subcategory
- Selects Prima then 4 Beden
- Clicks Önce En Yüksek Fiyat to sort items from highest to lowest
- Clicks first item

```
MainPage mainPage = new MainPage(browser);
Assert.assertTrue("Could not open Bebek Oyuncak category", browser.waitForClick(mainPage.bebekOyuncakMenu));
String title1 = browser.getTitle();
Assert.assertEquals("Bebek, Oyuncak - Migros", title1);

Assert.assertTrue("Could not open Bebek Bezi category", browser.waitForClick(mainPage.bebekBezi));
String title2 = browser.getTitle();
Assert.assertEquals("Bebek Bezleri, Bebek Bezi Kampanyaları - Migros", title2);

Assert.assertTrue("Could not open Prima page", browser.waitForClick(mainPage.prima));
String title3 = browser.getTitle();
Assert.assertEquals("Prima Bebek Bezleri, Bebek Bezi Kampanyaları - Migros", title3);

Assert.assertTrue("Could not open 4 Beden page", browser.waitForClick(mainPage.dortBeden));
String isChecked = mainPage.checkDortBeden.getAttribute("class");
Assert.assertEquals("checkbox_minimal-orange checked", isChecked);
```

In this scenario I checked if page titles are correct or not. Because when you go following pages, the titles change

- Bebek, Oyuncak category
- Bebek Bezi subcategory
- Prima selection

On the 4 Beden page I checked if the checkbox is checked or not. Because after you click 4 Beden option checkbox is checked.

```

browser.waitAndClick(mainPage.sirala);
Assert.assertTrue("Could not find sort from highest button", browser.waitAndClick(mainPage.enPahaliyaGore));
String biggest = mainPage.price.get(0).getText();
String replaceString = biggest.replace(" TL", "");
replaceString = replaceString.replace(",", ".");
biggest = replaceString;
float bigint = Float.parseFloat(biggest);
float bigint2 = bigint;

for(WebElement value: mainPage.price)
{
    String small = value.getText();
    String replacedString=small.replace(" TL", "");
    replacedString=replacedString.replace(",", ".");
    small = replacedString;
    float smallint = Float.parseFloat(small);
    if(bigint2 >= smallint)
    {
        bigint2 = smallint;
        System.out.println(bigint2);
    }
    else
    {
        Assert.assertTrue("Prices not descending", true);
    }
}

```

First the program clicks Sirala button to popup sort types. Sometimes they are not visible. Then it selects Önce En Yüksek Fiyat option. In this section I take all the prices on the page to check they are sorted correctly. I take them then convert to float without 'TL' and ',' characters. After that I check their order. Also printing them to console to see result.

```

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs,
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestio
ChromeDriver was started successfully.
Haz 19, 2020 10:14:10 ÖS org.openqa.selenium.remote.ProtocolHandshake createSessio
INFO: Detected dialect: W3C
159.85
159.85
159.85
139.85
139.85
139.85
139.85
139.85
94.9
79.9
79.9
79.9
79.9
79.9
79.9
71.18
71.18
71.18
71.18
(Prices not descending, true);

```

```

browser.waitForClick(mainPage.enPahalıUrun);
boolean sepet = browser.isElementDisplayed(mainPage.sepetEkle);
Assert.assertTrue("Ürün sayfasına ulaşamadı", sepet);
String enpahali = mainPage.enPahalıUrunFiyat.getText();
enpahali = enpahali.replace(" TL", "");
enpahali = enpahali.replace(",", ".");
float enpahaliFiyat = Float.parseFloat(enpahali);
Assert.assertEquals(bigint, enpahaliFiyat, 0.0001);

```

Also, to know if we are in the product page or not I checked if the page has Sepete Ekle button or not. If not, it gives assertion error "Ürün Sayfasına Ulaşamadı". Here again I checked if the price is the highest one from the previous page.

```

@FindBy(linkText = "Bebek, Oyuncak")
public WebElement bebekOyuncakMenu;

@FindBy(css = "a[data-monitor-ga-action='Bebek Bezi']")
public WebElement bebekBezi;

@FindBy(xpath = "//*[@id='page-area']/div/div/div[1]/div[1]/div/div[3]/ul/li/ul/li[1]/a/div/span[2]")
public WebElement prima;

@FindBy(css = "span[data-request-param='216']")
public WebElement checkDortBeden;

@FindBy(css = "a[data-monitor-ga-label='4 Beden']")
public WebElement dortBeden;

@FindBy(linkText = "Önce En Yüksek Fiyat")
public WebElement enPahaliyaGore;

@FindBy(xpath = "//*[@id='product-list-sort']/nav/button/div/span")
public WebElement sirala;

@FindBy(xpath = "//*[@id='product-list']/div[2]/div[1]/div/form/div[1]/h5/a")
public WebElement enPahaliUrun;

@FindBy(xpath = "//*[@id='store-product-primary-price']")
public WebElement enPahaliUrunFiyat;

@FindBy(xpath = "//*[@id='product-page']/div/div/div[1]/div[2]/div/div[2]/form/div/div[2]/button")
public WebElement sepetEkle;

@FindBy(xpath = "//*[@contains(@class, 'value')]")
public List<WebElement> price;

```

- bebekOyuncakMenu is looking for a link with text Bebek, Oyuncak
- bebekBezi is searching for a link element with the css has data-monitor-ga-action and equals to Bebek Bezi
- To click Prima, I used xpath
- To select 4 Beden option I used css method. It looks links with data-monitor-ga-label equals to 4 Beden
- Sirala clicks Sirala button to popup sorting options by its xpath
- enPahaliyaGore selects highest priced item sort
- enPahaliUrun clicks first item according to xpath.
- enPahaliUrunFiyat is takes the price text from product page by its xpath
- sepetEkle method finds Sepete Ekle button on the product page to check if we are in the product page or not.
- Price is a list of WebElements. I used this method to get all prices. It looks for elements in the page with class attribute equals to value.