SOFTWARE VERIFICATION, VALIDATION AND TESTING

TESTING DOCUMENTATION

Testing OLX.ba

Prepared by:
Fatima Ademovic

Proposed to:
Samed Jukić, Assist. Prof. Dr.
Aldin Kovačević, Teaching Assistant

Sarajevo, 22.1.2023.

TABLE OF CONTENTS

Contents

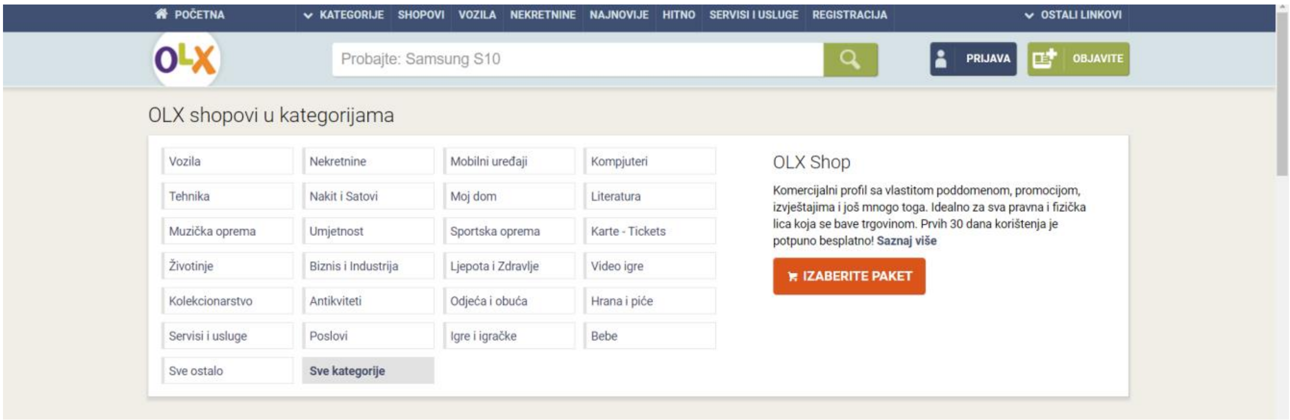
1. Introduction

1.1. About the Project

OLX is the world's leading shopping platform that is present in more than 30 countries around the world. OLX makes it easy to connect people to buy, sell or exchange products and services, and offers quick and easy posting of ads from any mobile device or computer. Every month, hundreds of millions of people use OLX to easily, safely and conveniently find their perfect home, buy or sell a car, find a great job, find or sell a wide range of products
At the beginning of 2015, PIK.ba became a member of the global classifieds network, OLX, continuing to operate under the local extension OLX.ba.
Link: ***https://www.olx.ba/***

1.2. Project Functionalities and Screenshots

-OLX makes it easy to connect people to buy, sell or exchange products and services



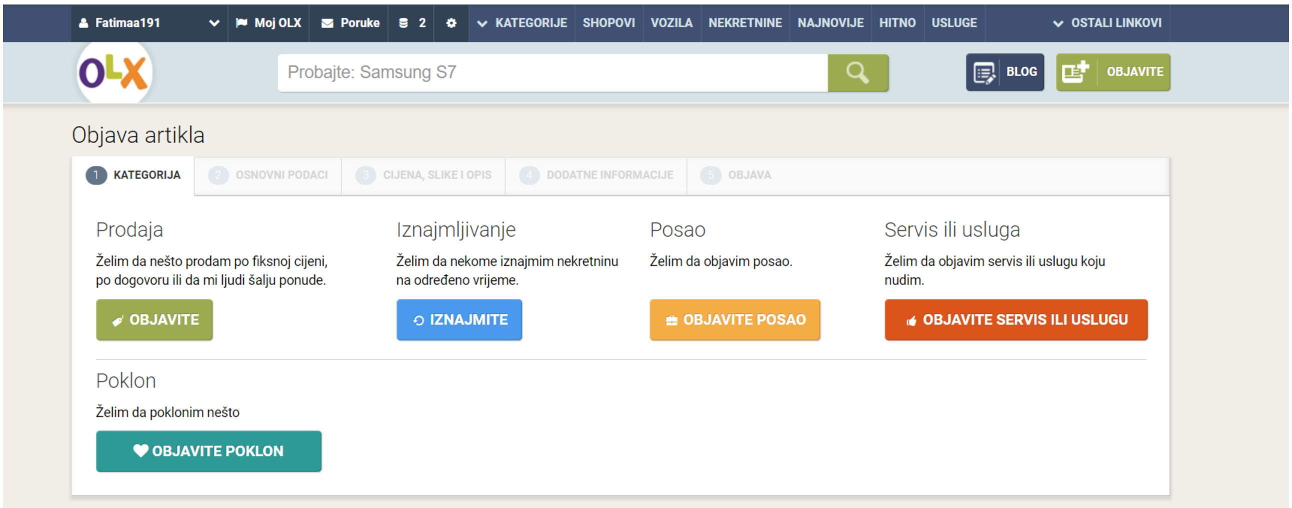
OLX shopovi u kategorijama

Vozila	Nekretnine	Mobilni uređaji	Kompjuteri
Tehnika	Nakit i Satovi	Moj dom	Literatura
Muzička oprema	Umjetnost	Sportska oprema	Karte - Tickets
Životinje	Biznis i Industrija	Ljepota i Zdravlje	Video igre
Kolekcionarstvo	Antikviteti	Odjeća i obuća	Hrana i piće
Servisi i usluge	Poslovi	Igre i igračke	Bebe
Sve ostalo	**Sve kategorije**		

OLX Shop

Komercijalni profil sa vlastitom poddomenom, promocijom, izvještajima i još mnogo toga. Idealno za sva pravna i fizička lica koja se bave trgovinom. Prvih 30 dana korištenja je potpuno besplatno! **Saznaj više**

-offers quick and easy ad posting



Objava artikla

Prodaja

Želim da nešto prodam po fiksnoj cijeni, po dogovoru ili da mi ljudi šalju ponude.

Iznajmljivanje

Želim da nekome iznajmim nekretninu na određeno vrijeme.

Posao

Želim da objavim posao.

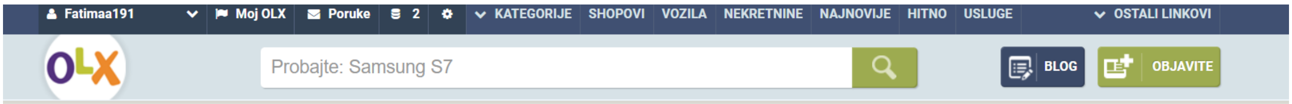
Servis ili usluga

Želim da objavim servis ili uslugu koju nudim.

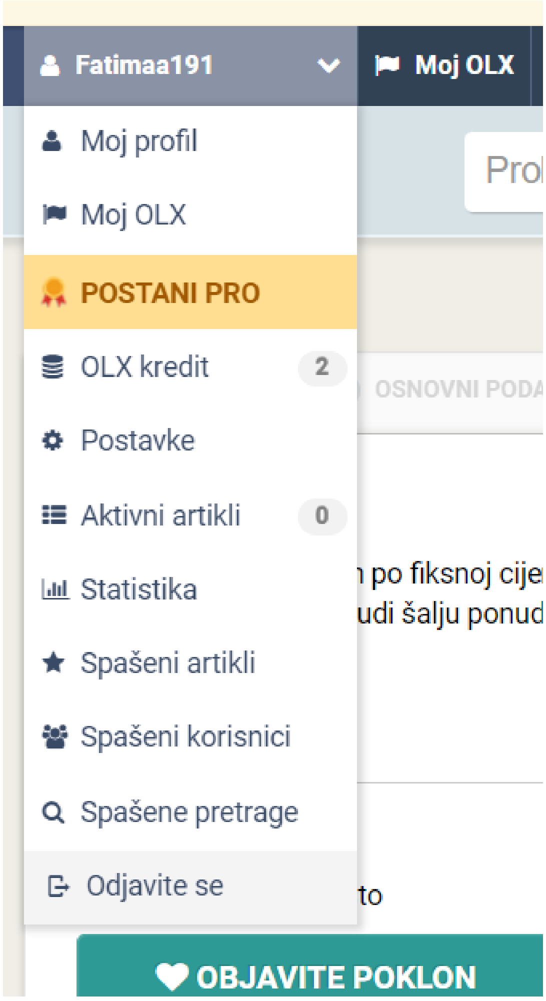
Poklon

Želim da poklonim nešto

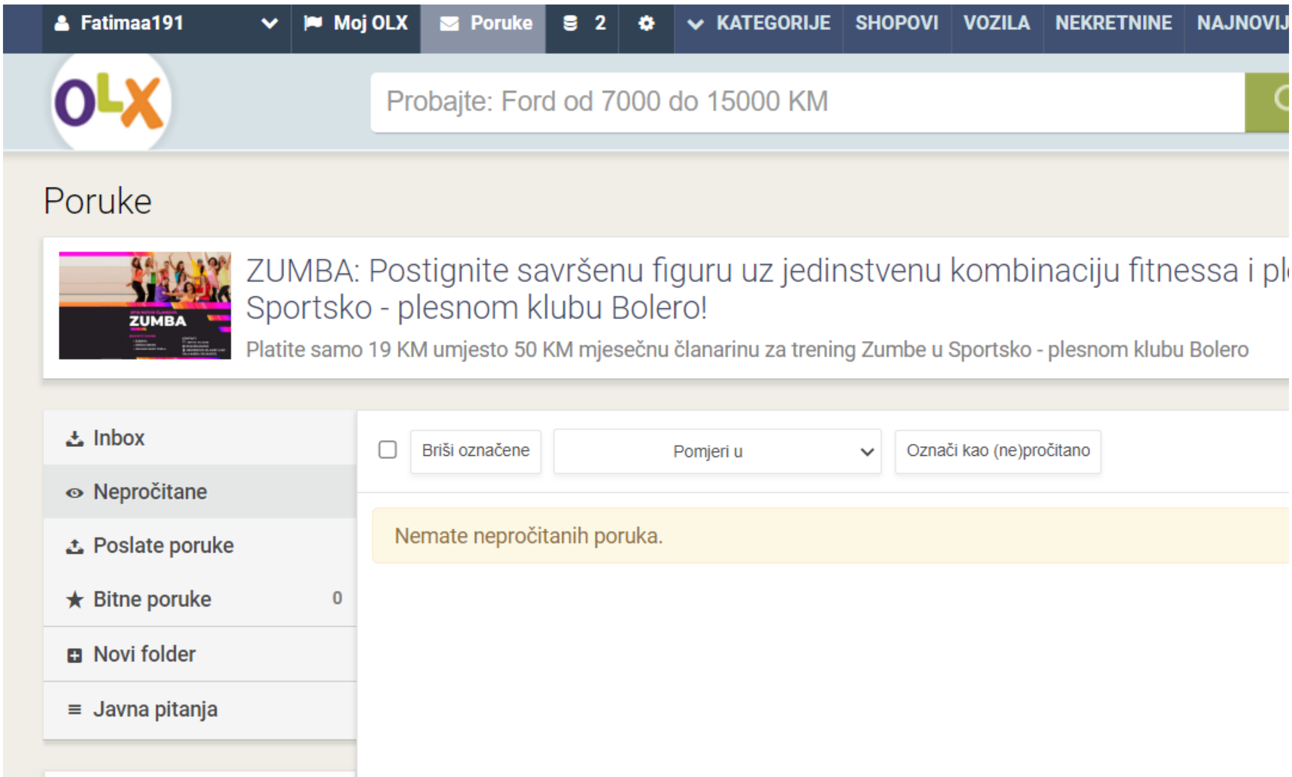
-search bar for easily, safely and conveniently find your perfect home, buy or sell a car, find a great job, and find or sell a wide range of products



-functionalities such as logging in, logging out, registration



-OLX makes it easy to connect people with the goal of buying, selling or exchanging through the messaging feature



Poruke

ZUMBA: Postignite savršenu figuru uz jedinstvenu kombinaciju fitnessa i pl
Sportsko - plesnom klubu Bolero!

Platite samo 19 KM umjesto 50 KM mjesečnu članarinu za trening Zumbe u Sportsko - plesnom klubu Bolero

Nemate nepročitanih poruka.

2. Test Plan

2.1. Scope

I'm testing OLX.ba for this project. I'm testing a few of the application's features, including opening the page, determining whether the name is "OLX.ba - Svijet Kupoprodaje," whether a menu is present, whether the sign-in button is functional, whether the sign-out button is present, and a few other features among the many that olx.ba offers.

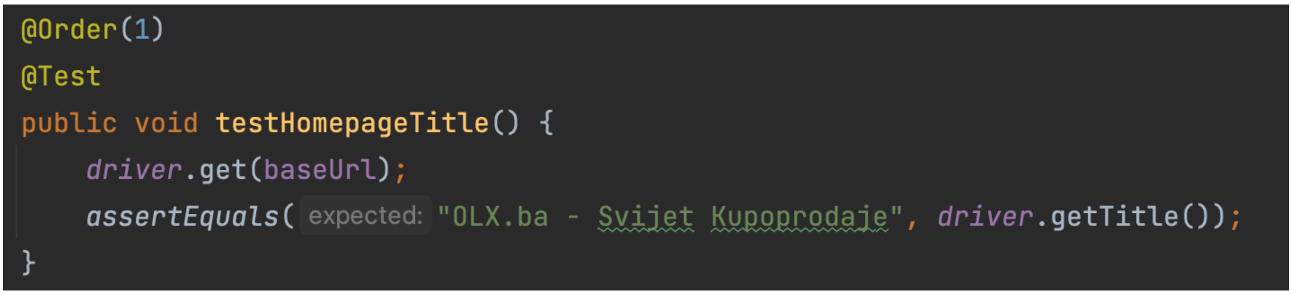
2.2. Testing Environment and Tools

I used Selenium for testing. Environment I used is IntelliJ and programming language that I used is Java.

3. Test Execution

3.1. Test Home page title

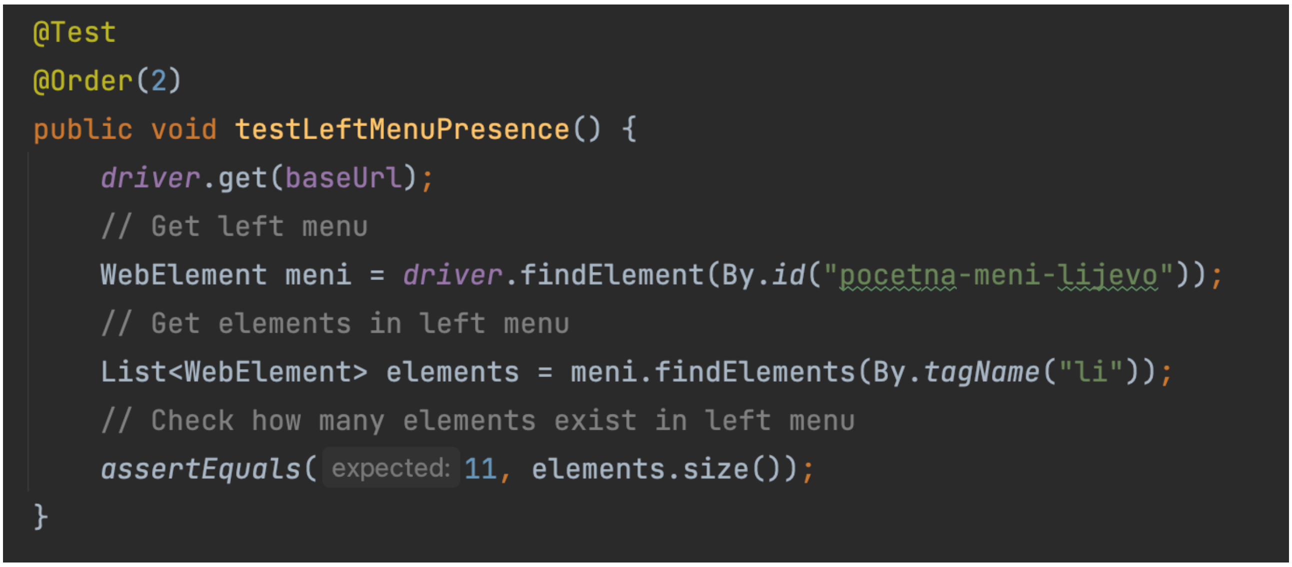
Test Name: Home page title test				
Description: Test of opening the page and whether the name is "OLX.ba - Svijet Kupoprodaje"				
Pre-condition(s):				
Test Steps: 1.Goes to home page Olx.ba	**Test Data:**	**Expected Result:** Opens page	**Actual Result:** **Opens page**	**Status:** PASS
Notes:				



```java
@Order(1)
@Test
public void testHomepageTitle() {
    driver.get(baseUrl);
    assertEquals( expected: "OLX.ba - Svijet Kupoprodaje", driver.getTitle());
}
```

3.2. Test Menu presence

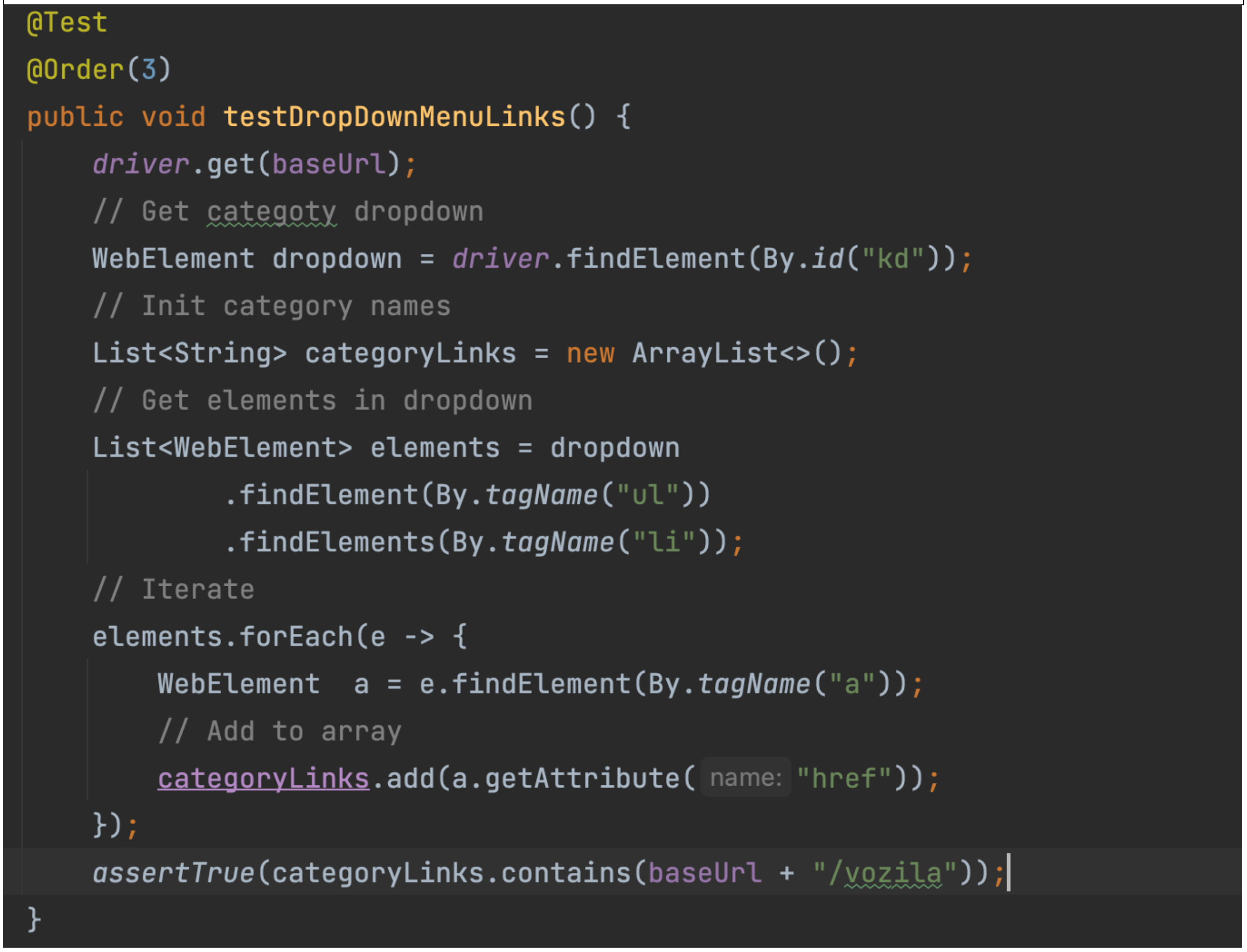
Test Name: Menu presence test				
Description: Test if there is a menu on the left and if there are 11 elements in it.				
Pre-condition(s):				
Test Steps: 1. Goes to page 2. Checks if there is menu left 3. If there is menu, checks if how much elements is there In menu	**Test Data:**	**Expected Result:** there is a menu on the left and if there are 11 elements in it.	**Actual Result:** there is a menu on the left and if there are 11 elements in it.	**Status:** PASS
Notes:				



```java
@Test
@Order(2)
public void testLeftMenuPresence() {
    driver.get(baseUrl);
    // Get left menu
    WebElement meni = driver.findElement(By.id("pocetna-meni-lijevo"));
    // Get elements in left menu
    List<WebElement> elements = meni.findElements(By.tagName("li"));
    // Check how many elements exist in left menu
    assertEquals( expected: 11, elements.size());
}
```

3.3 Test DropDown Menu Links

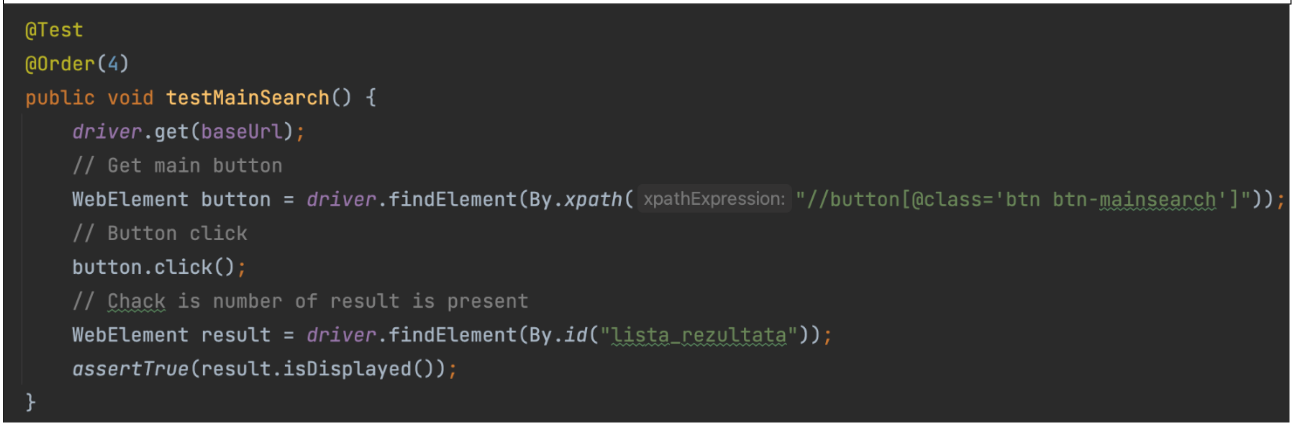
Test Name: DropDown Menu Links test				
Description: Test to check if the CATEGORIES dropdown menu contains a link to the vehicle category				
Pre-condition(s):				
Test Steps: 1. Goes to page 2. Get category dropdown 3. Get all elements in dropdown 4. Goes through array of elements 5. Checks if there is category "vozila"	**Test Data:**	**Expected Result:** Dropdown menu contains a link to the vehicle category.	**Actual Result: Dropdown menu contains a link to the vehicle category.**	**Status:** PASS
Notes:				



```java
@Test
@Order(3)
public void testDropDownMenuLinks() {
    driver.get(baseUrl);
    // Get categoty dropdown
    WebElement dropdown = driver.findElement(By.id("kd"));
    // Init category names
    List<String> categoryLinks = new ArrayList<>();
    // Get elements in dropdown
    List<WebElement> elements = dropdown
            .findElement(By.tagName("ul"))
            .findElements(By.tagName("li"));
    // Iterate
    elements.forEach(e -> {
        WebElement  a = e.findElement(By.tagName("a"));
        // Add to array
        categoryLinks.add(a.getAttribute( name: "href"));
    });
    assertTrue(categoryLinks.contains(baseUrl + "/vozila"));
}
```

3.4 Test Main Search

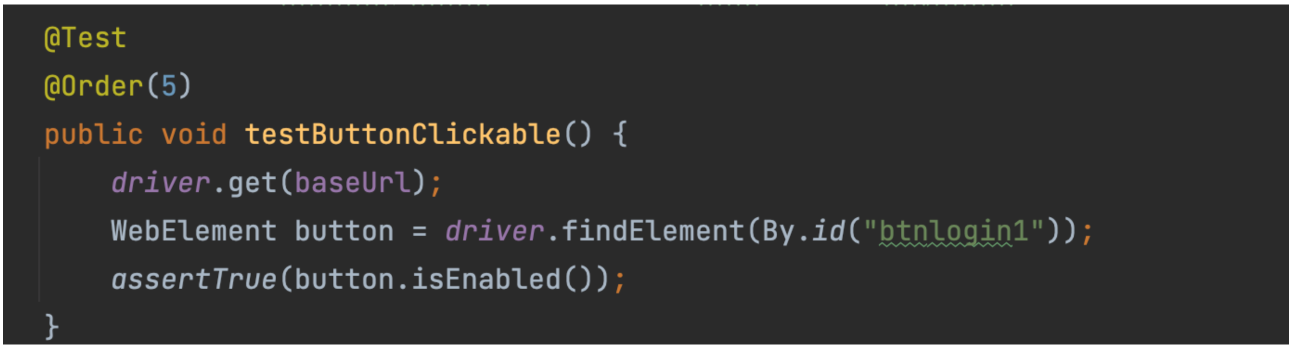
Test Name: Main Search test				
Description:				
Pre-condition(s):				
Test Steps: 1. Checks if there is search button 2. If there is clicks button 3. Check if number of result is present	**Test Data:**	**Expected Result:** Results are displayed	**Actual Result:** **Results are displayed.**	**Status:** PASS
Notes:				



```
@Test
@Order(4)
public void testMainSearch() {
    driver.get(baseUrl);
    // Get main button
    WebElement button = driver.findElement(By.xpath( xpathExpression: "//button[@class='btn btn-mainsearch']"));
    // Button click
    button.click();
    // Chack is number of result is present
    WebElement result = driver.findElement(By.id("lista_rezultata"));
    assertTrue(result.isDisplayed());
}
```

3.5 Test Clickable Button

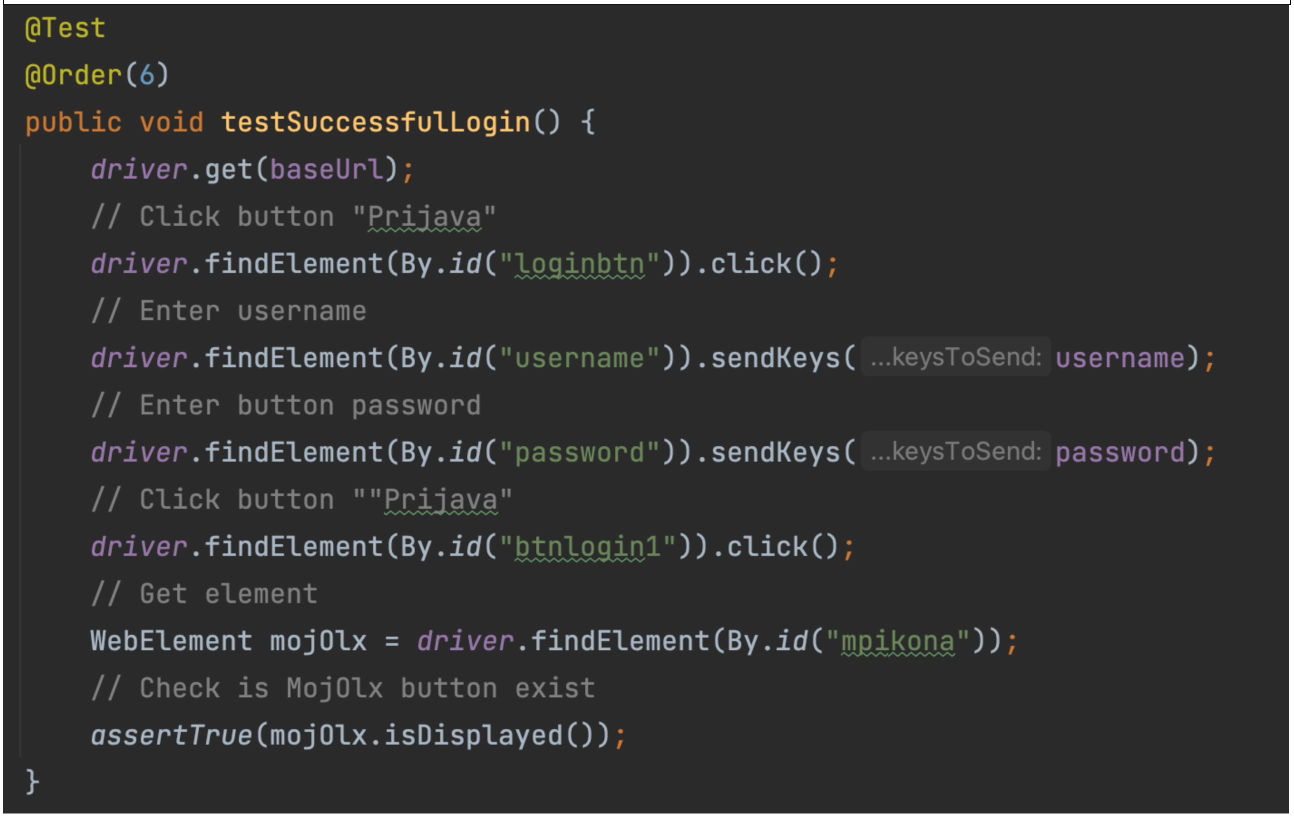
Test Name: Clickable Login Button test				
Description: **Test whether there is a login button and whether it can be clicked**				
Pre-condition(s):				
Test Steps: 1. Checks if button "Prijava" is clickable	**Test Data:**	**Expected Result:** There is log in button and can be clicked.	**Actual Result:** **There is log in button and can be clicked.**	**Status:** PASS
Notes:				



```
@Test
@Order(5)
public void testButtonClickable() {
    driver.get(baseUrl);
    WebElement button = driver.findElement(By.id("btnlogin1"));
    assertTrue(button.isEnabled());
}
```

3.6. Test Successful Login

Test Name: Successful Login test				
Description: Test for login, if the login is successful, it will appear in MojOlx in the navbar				
Pre-condition(s):				
Test Steps: 1. Clicks “Prijava buton” 2. Enter usernme and password 3. Clicks second “Prijava” button 4. Checks if login successful if there is MojOlx button	**Test Data:** **Username** **Password**	**Expected Result:** User is successfully logged in	**Actual Result:** **User is successfully logged in**	**Status:** PASS
Notes:				

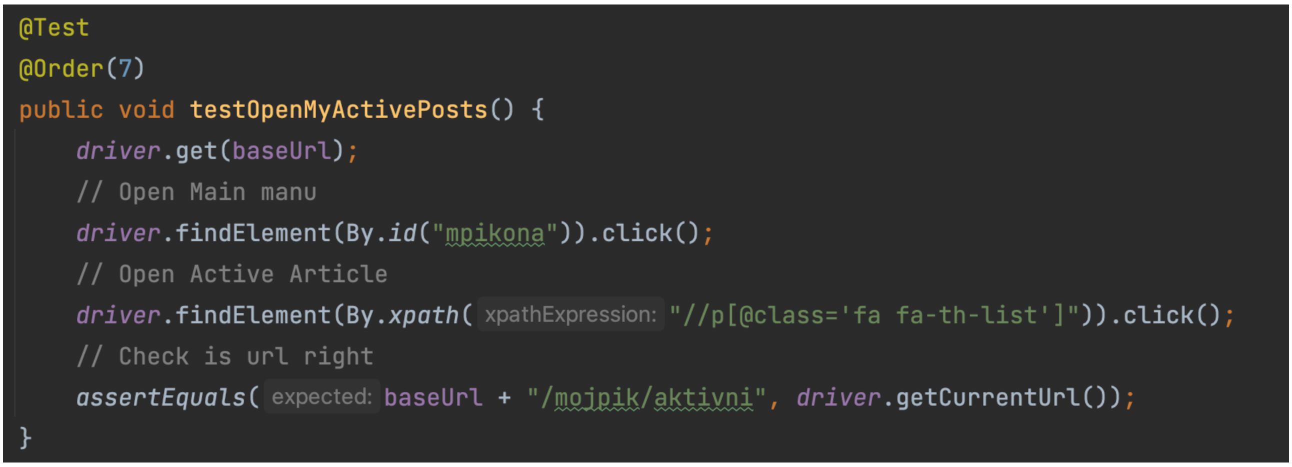


```
@Test
@Order(6)
public void testSuccessfulLogin() {
    driver.get(baseUrl);
    // Click button "Prijava"
    driver.findElement(By.id("loginbtn")).click();
    // Enter username
    driver.findElement(By.id("username")).sendKeys( ...keysToSend: username);
    // Enter button password
    driver.findElement(By.id("password")).sendKeys( ...keysToSend: password);
    // Click button ""Prijava"
    driver.findElement(By.id("btnlogin1")).click();
    // Get element
    WebElement mojOlx = driver.findElement(By.id("mpikona"));
    // Check is MojOlx button exist
    assertTrue(mojOlx.isDisplayed());
}
```

3.7. Test Open Active posts

Test Name: Open Active posts test				
Description: Test whether it opens my active items through Main Menu -> Active items				
Pre-condition(s):				
Test Steps: 1. If user is logged in (if user is not	**Test Data:**	**Expected Result:** Opens Active items through	**Actual Result:** **Opens Active items through**	**Status:** PASS

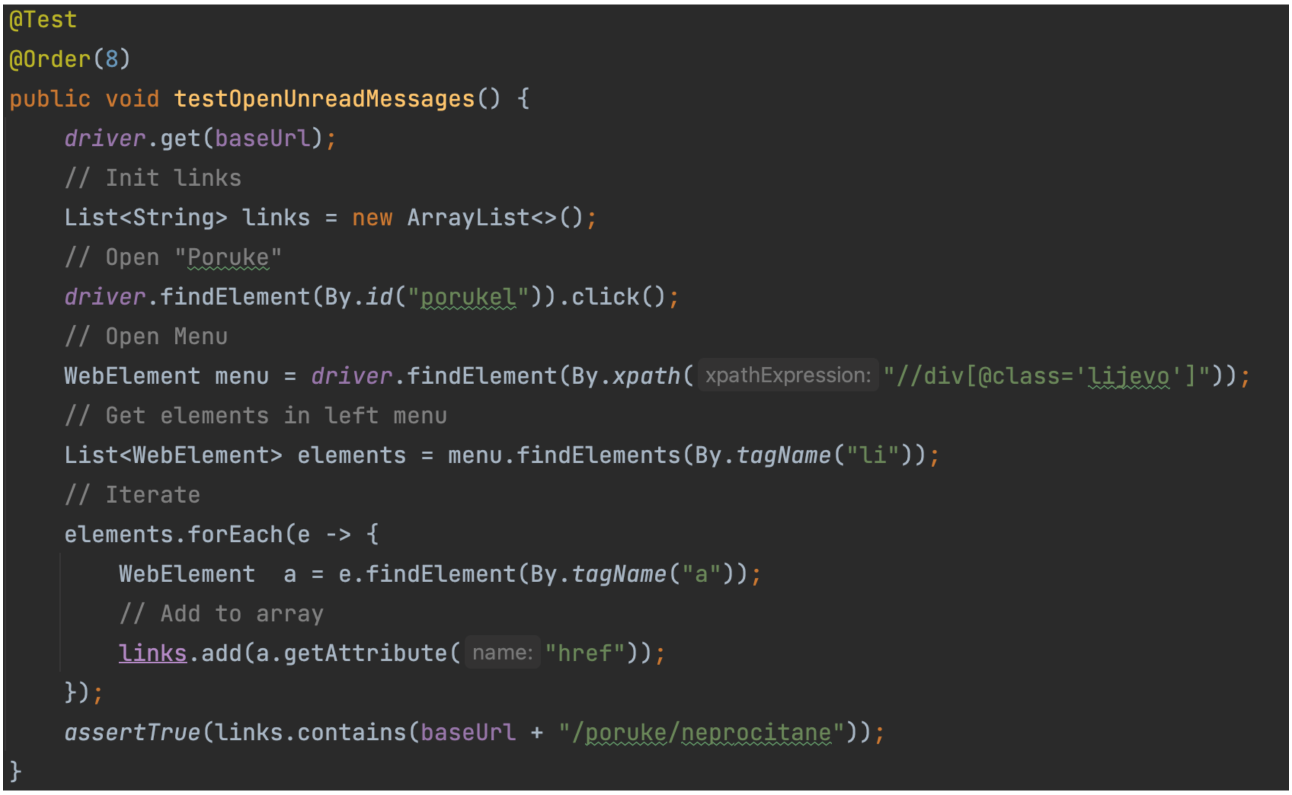
looged in test will fail) checks if there id Active elements in menu 2. Checks if url is baseUrl + \mojpik\aktivni		Main Menu	**Main Menu**	
Notes:				



```
@Test
@Order(7)
public void testOpenMyActivePosts() {
    driver.get(baseUrl);
    // Open Main manu
    driver.findElement(By.id("mpikona")).click();
    // Open Active Article
    driver.findElement(By.xpath( xpathExpression: "//p[@class='fa fa-th-list']")).click();
    // Check is url right
    assertEquals( expected: baseUrl + "/mojpik/aktivni", driver.getCurrentUrl());
}
```

3.8 Test Open Unread Messages

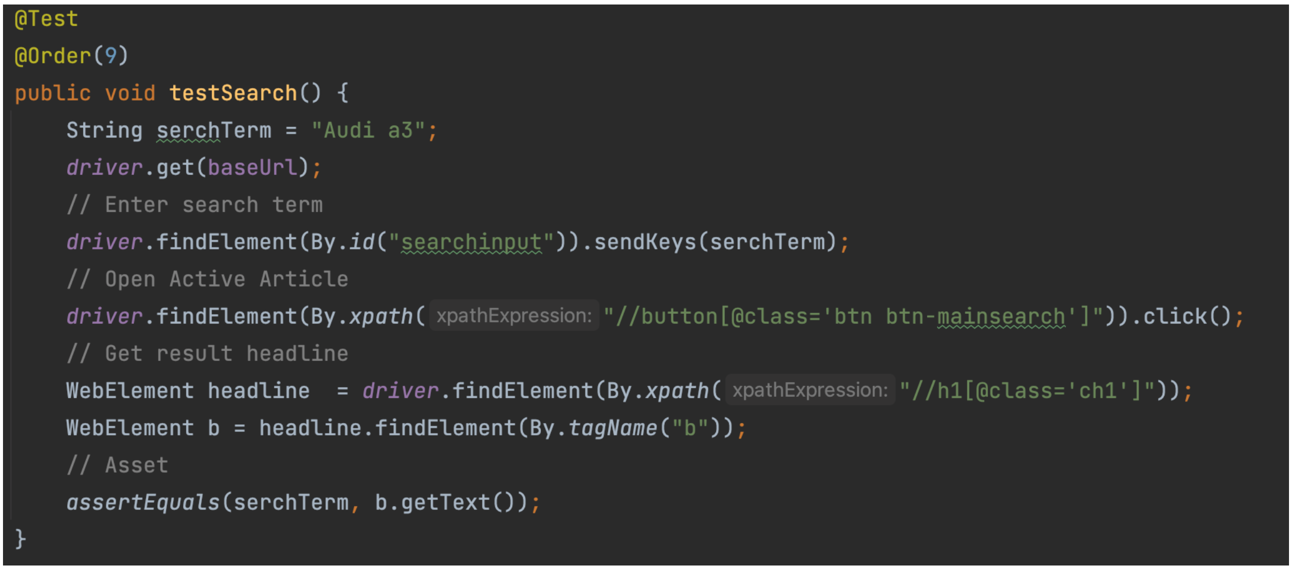
Test Name: Open unread messages test				
Description: Test to see if it opens unread messages through the Messages->Unread menu				
Pre-condition(s):				
Test Steps: 1. Clicks on messages button in menu (if user is logged in) 2. Then it goes through list on left 3. Checks if there is url with /poruke/neprocitane 4. Opens unreadMessages	**Test Data:**	**Expected Result:** Unread Messages are opened through Messages > Unread messages	**Actual Result: Unread Messages are opened through Messages > Unread messages**	**Status:** PASS
Notes:				



```
@Test
@Order(8)
public void testOpenUnreadMessages() {
    driver.get(baseUrl);
    // Init links
    List<String> links = new ArrayList<>();
    // Open "Poruke"
    driver.findElement(By.id("porukel")).click();
    // Open Menu
    WebElement menu = driver.findElement(By.xpath( xpathExpression: "//div[@class='lijevo']"));
    // Get elements in left menu
    List<WebElement> elements = menu.findElements(By.tagName("li"));
    // Iterate
    elements.forEach(e -> {
        WebElement  a = e.findElement(By.tagName("a"));
        // Add to array
        links.add(a.getAttribute( name: "href"));
    });
    assertTrue(links.contains(baseUrl + "/poruke/neprocitane"));
}
```

3.9. Test Search

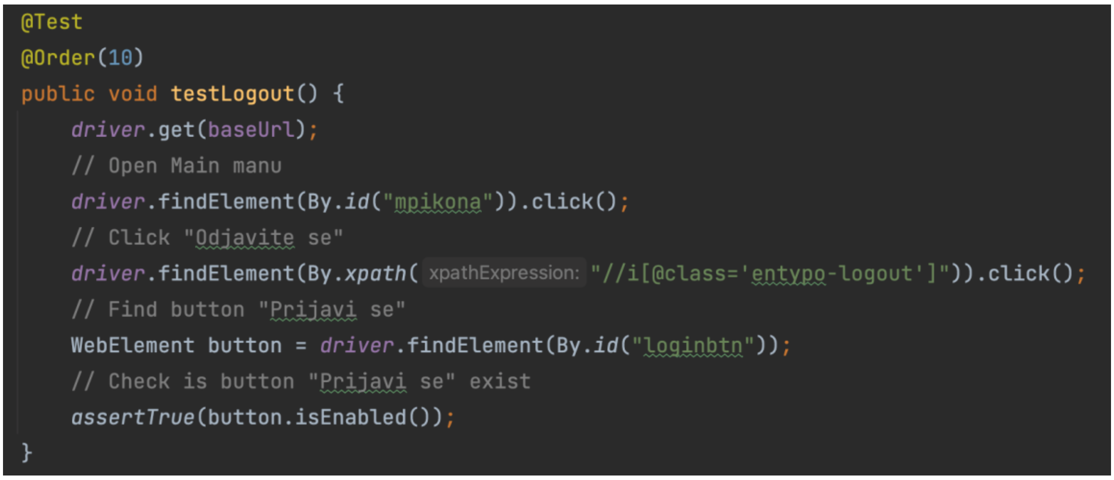
Test Name: Test Search				
Description: Test the search bar to check whether the search term will be entered in the headline results				
Pre-condition(s):				
Test Steps: 1. When search term is entered and button search is clicked 2. Checks if there is headline of searched item	**Test Data:** **Search term**	**Expected Result:** The search term will be entered in the headline results	**Actual Result:** **The search term will be entered in the headline results**	**Status:** PASS
Notes:				



```java
@Test
@Order(9)
public void testSearch() {
    String serchTerm = "Audi a3";
    driver.get(baseUrl);
    // Enter search term
    driver.findElement(By.id("searchinput")).sendKeys(serchTerm);
    // Open Active Article
    driver.findElement(By.xpath( xpathExpression: "//button[@class='btn btn-mainsearch']")).click();
    // Get result headline
    WebElement headline  = driver.findElement(By.xpath( xpathExpression: "//h1[@class='ch1']"));
    WebElement b = headline.findElement(By.tagName("b"));
    // Asset
    assertEquals(serchTerm, b.getText());
}
```

3.10 Test Logout

Test Name: Log out test				
Description: Test if logout works				
Pre-condition(s):				
Test Steps: 1. First it checks if there is “Odjavite se” button (if there is user is logged in already) 2. Clicks that button 3. Then it checks if there is “Prijavi se” button 4. If there is that means user is logged out.	**Test Data:**	**Expected Result:** Log out of the site	**Actual Result:** **Log out of the site**	**Status:** PASS
Notes:				



```java
@Test
@Order(10)
public void testLogout() {
    driver.get(baseUrl);
    // Open Main manu
    driver.findElement(By.id("mpikona")).click();
    // Click "Odjavite se"
    driver.findElement(By.xpath( xpathExpression: "//i[@class='entypo-logout']")).click();
    // Find button "Prijavi se"
    WebElement button = driver.findElement(By.id("loginbtn"));
    // Check is button "Prijavi se" exist
    assertTrue(button.isEnabled());
}
```

9. Conclusion

9.1. Testing Summary

Testing Tool	**Total Tests**	**Passed Tests**	**Failed Tests**
IntelliJ (IDE), Selenium	10	#10 of passing tests	#0 of failing tests

9.2. Final Thoughts

I gained a lot of knowledge about how to use Selenium and web page testing in general through this project. All the tests in this project are done simply and with understanding for all the essential functions of the site.