

SOFTWARE VERIFICATION, VALIDATION AND TESTING

TESTING DOCUMENTATION

PROJECT NAME: GENELEC.BA

Prepared by:
Mirza Nikolić
Damir Todić
Anesa Čutuna

Proposed to:
SamedJukić, Assist. Prof. Dr.
AldinKovačević, Teaching Assistant

Date of submission: 22.01.2023.

TABLE OF CONTENTS:

1. Introduction	3
1.1. About the Project	3
1.2. Project Functionalities and Screenshots	3
2. Test Plan	5
2.1. Scope	5
2.2. Testing Environment and Tools	5
3. Test Execution	5
3.1. Registration test	14
3.2. Login test	15
3.3. Category Test	16
3.4. Search Test	17
3.5. LogOut Test	18
3.6. Decrease one Test	19
3.7. Wish list display Test	20
3.8. Product Sorting Test	21
3.9. Add to cart from shop page	22
3.10. Delete from Cart Test	23
3.11. Cart Increase Item by two Test	24
3.12. Subscribe to News letter Test	25
3.13.No Products Found Test	26
3.14. Get tags test	27
3.15. Remove from Wish List Test	28
4. Conclusion	29
4.1. Testing Summary	29
4.2. Final Thoughts	29

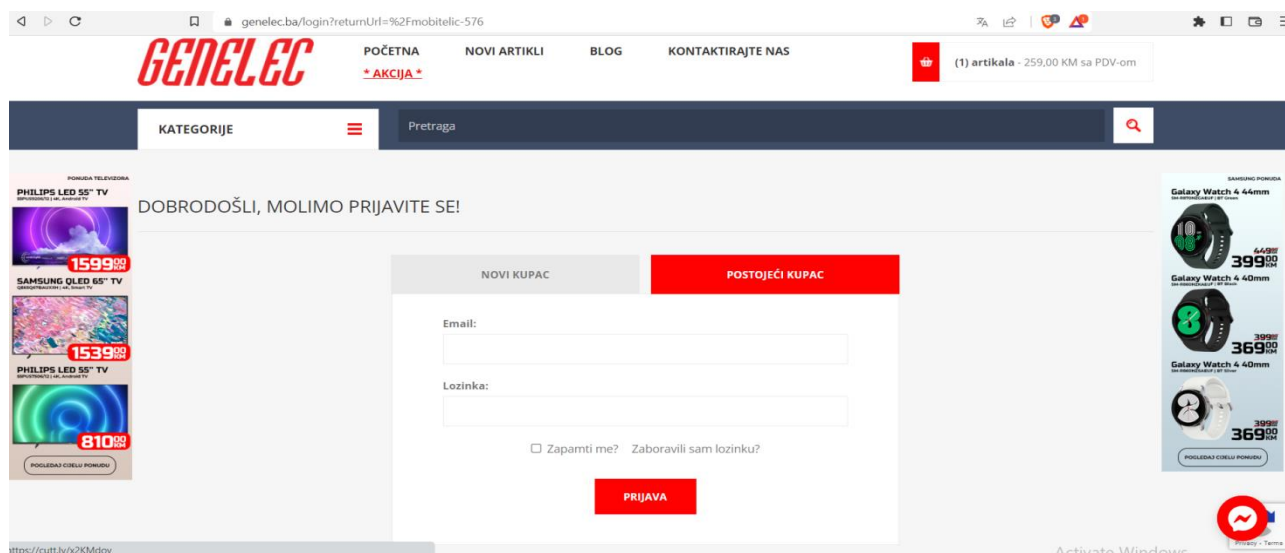
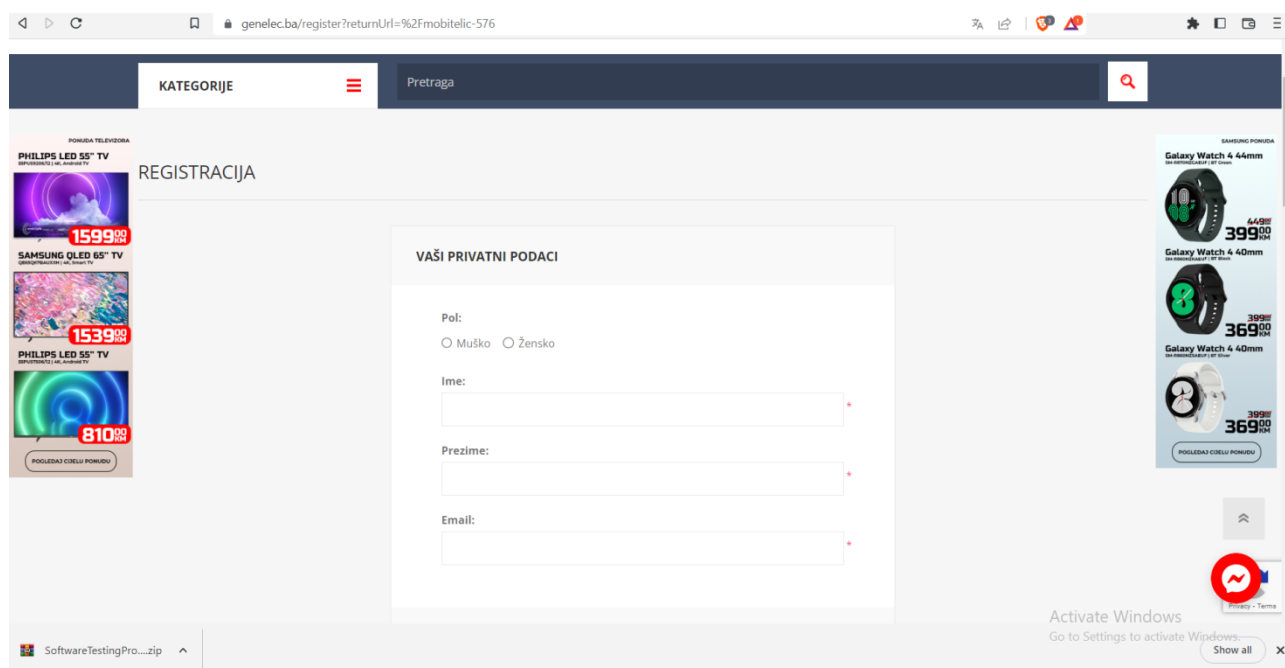
1. Introduction

1.1. About the Project

The name of the web page our project is based on is Genelec d.o.o. It is an electronic device store which was the first online store to offer item delivery across the entire country of BiH. It was established in 1989. Their web page offers a lot of options for testing since it has a lot of categories for the electronics It sells. You have the options of buying items (adding and deleting to and from shopping cart etc.) and more.

1.2. Project Functionalities and Screenshots

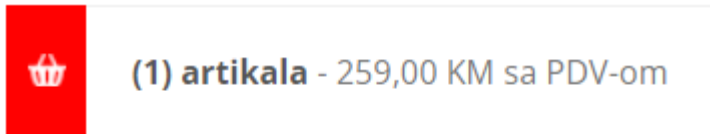
In this project we will test the functionalities such as Registering, Logging in, Logging out of the page.



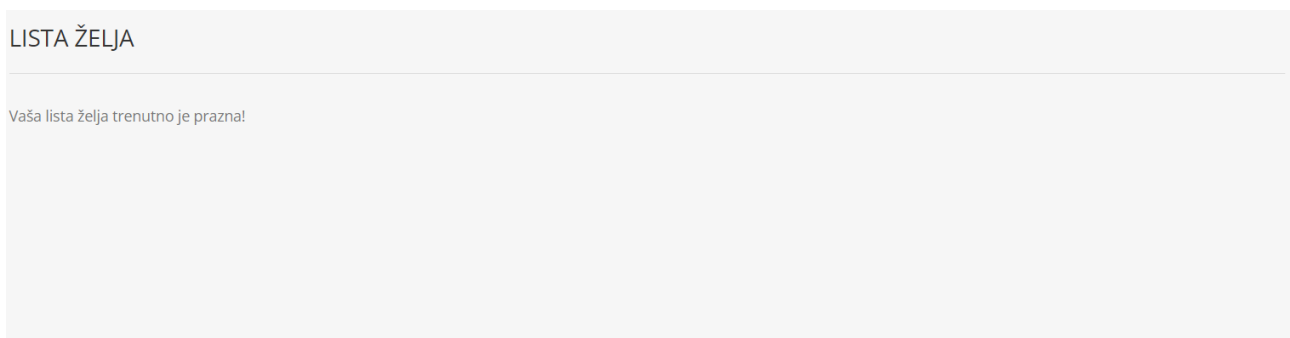
We will also be testing the Search Bar of the web page and it's functionality



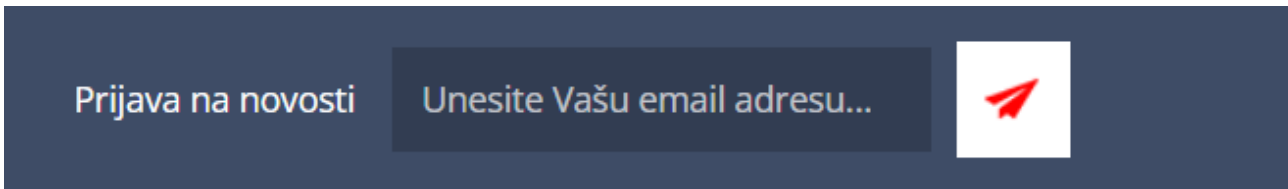
We will be testing the cart functionalities such as add to cart, deleting from cart etc.



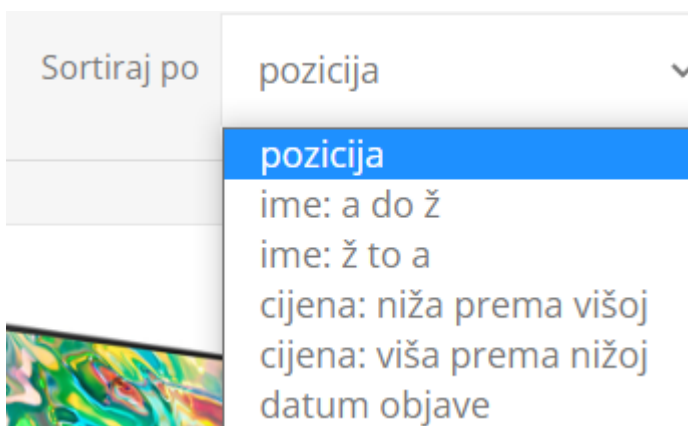
We will be testing the functionality of the page wish list



We will also be testing the subscribe to newsletter functionality



We will test the product sorting functionality



2. Test Plan

2.1. Scope

As mentioned above we will be focused on testing multiple functionalities of the page. We will test if we can successfully register on the page. Then we will test if we can successfully log in with a created account as well as if we can successfully log out of an account. We will test if we can successfully add/remove items from the shopping cart. We will test if the Newsletter works if an invalid e mail address is inputted. We will check if the search bar works correctly by inputting valid and invalid search arguments. We will test if the product sorting menu works. Of course we cannot cover every element and every functionality of the web page and we will focus on the above mentioned functionalities and elements.

2.2. Testing Environment and Tools

For this project we are using Java as our programming language, Selenium and JUNIT 5 for Web page testing and Page Object Models (POM). We are using the Chrome Web Driver which has to be imported in each test file using the appropriate path on your PC. Also Selenium dependencies (JAR Files) have to be imported into the project in order for the Test Files to work properly.

3. Test Execution

In order to execute tests, one should navigate to the test section and execute tests with JUnit. Tests can be manually executed one by one, or by TEST SUITES in groups. We have made different classes to separate base functionalities for pages which can be seen in the 'lib' folder, and we will be taking their methods from test classes which will be shown below as screenshots, so please refer to those methods.

We have six different screens that were tested in this project and these are their corresponding classes are as follows:

Cart page:

12 usages Mirza Nikolic *

```
public class CartPage {
```

7 usages

```
private final WebDriver driver;
```

3 usages

```
private final By QtyInput = By.className("qty-input");
```

1 usage

```
private final By UpdateButton = By.className("update-cart-button");
```

1 usage

```
private final By DeleteCheck = By.name("removefromcart");
```

1 usage

```
private final By NoData = By.className("no-data");
```

4 usages Mirza Nikolic

```
public int getQtyInput() {  
    String qty = driver.findElement(QtyInput).getAttribute( name: "value");  
    return Integer.valueOf(qty).intValue();  
}
```

1 usage Mirza Nikolic

```
public void decreaseQtyInput(int qty) {  
    WebElement element = driver.findElement(QtyInput);  
    int currentQty = Integer.valueOf(element.getAttribute( name: "value")).intValue();  
    element.clear();  
    element.sendKeys(String.valueOf( currentQty - qty));  
}
```

1 usage Mirza Nikolic

```
public void IncreaseQtyInput(int qty) {  
    WebElement element = driver.findElement(QtyInput);  
    int currentQty = Integer.valueOf(element.getAttribute( name: "value")).intValue();  
    element.clear();  
    element.sendKeys(String.valueOf( currentQty + qty));  
}
```

3 usages Mirza Nikolic

```
public void clickOnUpdateCart() { driver.findElement(UpdateButton).click(); }
```

2 usages Mirza Nikolic

```
public void clickOnDeleteCheck() { driver.findElement(DeleteCheck).click(); }
```

1 usage Mirza Nikolic

```
public Boolean isCartEmpty() { return driver.findElement(NoData).getText().trim().equals("Vaša korpa je prazna!"); }
```

Home Page:

• Mirza Nikolic *

```
public class HomePage {  
    16 usages  
    private final WebDriver driver;  
  
    1 usage  
    private final By HeaderLinksWrapper = By.className("header-links-wrapper");  
    1 usage  
    private final By LogoutNav = By.className("ico-logout");  
    1 usage  
    private final By WishListNav = By.className("ico-wishlist");  
    1 usage  
    private final By PhonesCategory = By.linkText("Mobilni telefoni");  
    1 usage  
    private final By Iphone11Card = By.linkText("Apple iPhone 11 64GB White");  
    2 usages  
    private final By SearchBar = By.id("small-searchterms");  
    1 usage  
    private final By SearchItem = By.className("instant-search-item");  
    1 usage  
    private final By MainCart = By.className("cart-trigger");  
    1 usage  
    private final By CartButton = By.className("add-to-cart-button");  
    1 usage  
    private final By CartContent = By.className("content");  
    1 usage  
    private final By NewsletterEmail = By.id("newsletter-email");  
    1 usage  
    private final By SubscribeButton = By.id("newsletter-subscribe-button");  
    1 usage  
    private final By SubscribeResultContent = By.id("newsletter-result-block");  
    1 usage  
    private final By SearchButton = By.className("search-box-button");  
}
```

```

    Mirza Nikolic
public HomePage(WebDriver _driver) { driver = _driver; }

3 usages Mirza Nikolic
public void clickOnHeaderLinksWrapper() { driver.findElement(HeaderLinksWrapper).click(); }

1 usage Mirza Nikolic
public void clickOnLogout() { driver.findElement(LogoutNav).click(); }

1 usage Mirza Nikolic
public void clickOnPhonesCategory() {
    driver.findElement(PhonesCategory).click();
}

1 usage Mirza Nikolic
public void clickOnMobilePhone() { driver.findElement(Iphone11Card).click(); }

2 usages Mirza Nikolic
public void clickOnCartButton() { driver.findElement(CartButton).click(); }

2 usages Mirza Nikolic
public void clickOnSearchBar() { driver.findElement(SearchBar).click(); }

2 usages Mirza Nikolic
public void clickOnMainCart() { driver.findElement(MainCart).click(); }

2 usages Mirza Nikolic
public void setSearchText(String searchText) { driver.findElement(SearchBar).sendKeys(searchText); }

1 usage Mirza Nikolic
public void clickOnSearchItem() { driver.findElement(SearchItem).click(); }

1 usage Mirza Nikolic
public void clickOnWishList() { driver.findElement(WishListNav).click(); }

```



```

4 usages  Mirza Nikolic
public int getQtyInput() {
    String qty = driver.findElement(QtyInput).getAttribute( name: "value");
    return Integer.valueOf(qty).intValue();
}

1 usage  Mirza Nikolic
public void decreaseQtyInput(int qty) {
    WebElement element = driver.findElement(QtyInput);
    int currentQty = Integer.valueOf(element.getAttribute( name: "value")).intValue();
    element.clear();
    element.sendKeys(String.valueOf( currentQty - qty));
}

1 usage  Mirza Nikolic
public void IncreaseQtyInput(int qty) {
    WebElement element = driver.findElement(QtyInput);
    int currentQty = Integer.valueOf(element.getAttribute( name: "value")).intValue();
    element.clear();
    element.sendKeys(String.valueOf( currentQty + qty));
}

3 usages  Mirza Nikolic
public void clickOnUpdateCart() { driver.findElement(UpdateButton).click(); }

2 usages  Mirza Nikolic
public void clickOnDeleteCheck() { driver.findElement>DeleteCheck).click(); }

1 usage  Mirza Nikolic
public Boolean isCartEmpty() { return driver.findElement>NoData).getText().trim().equals("Vaša korpa je prazna!"); }

```

Login page:

```

6 usages  Mirza Nikolic
public LoginPage(WebDriver _driver) { driver = _driver; }

2 usages  Mirza Nikolic
public void clickOnHeaderLinksWrapper() { driver.findElement(HeaderLinksWrapper).click(); }

2 usages  Mirza Nikolic
public void clickOnLoginLink() { driver.findElement(LoginNav).click(); }

2 usages  Mirza Nikolic
public void setEmail(String email) {
    driver.findElement>EmailInput).sendKeys(email);
}

2 usages  Mirza Nikolic
public void setPassword(String password) { driver.findElement>PasswordInput).sendKeys(password); }

2 usages  Mirza Nikolic
public void clickOnRememberMe() { driver.findElement(RememberMe).click(); }

2 usages  Mirza Nikolic
public void loginSubmit() { driver.findElement(LoginSubmit).click(); }

5 usages  Mirza Nikolic
public void login() {
    clickOnHeaderLinksWrapper();
    clickOnLoginLink();
    setEmail("mirza.nikolic@stu.ibu.edu.ba");
    setPassword("mirzastudent123");
    clickOnRememberMe();
    loginSubmit();
}

```

Register page:

```
1 usage  👤 Mirza Nikolic
public void clickOnHeaderLinksWrapper() { driver.findElement(HeaderLinksWrapper).click(); }

1 usage  👤 Mirza Nikolic
public void clickOnRegisterLink() { driver.findElement(RegisterNav).click(); }

1 usage  👤 Mirza Nikolic
public void clickOnGenderMale() { driver.findElement(GenderMale).click(); }

1 usage  👤 Mirza Nikolic
public void setFirstName(String firstName) { driver.findElement(FirstName).sendKeys(firstName); }

1 usage  👤 Mirza Nikolic
public void setLastName(String lastName) { driver.findElement(LastName).sendKeys(lastName); }

1 usage  👤 Mirza Nikolic
public void setEmail(String email) { driver.findElement(Email).sendKeys(email); }

1 usage  👤 Mirza Nikolic
public void setCompany(String company) { driver.findElement(Company).sendKeys(company); }

1 usage  👤 Mirza Nikolic
public void setStreetAddress(String streetAddress) { driver.findElement(StreetAddress).sendKeys(streetAddress); }

1 usage  👤 Mirza Nikolic
public void setZipPostalCode(String zipPostalCode) { driver.findElement(ZipPostalCode).sendKeys(zipPostalCode); }

1 usage  👤 Mirza Nikolic
public void setCity(String city) { driver.findElement(City).sendKeys(city); }

1 usage  👤 Mirza Nikolic
public void setPhone(String phone) { driver.findElement(Phone).sendKeys(phone); }

1 usage  👤 Mirza Nikolic
public void clickOnNewsletter() { driver.findElement(Newsletter).click(); }
```

```
1 usage  👤 Mirza Nikolic
public void clickOnHeaderLinksWrapper() { driver.findElement(HeaderLinksWrapper).click(); }

1 usage  👤 Mirza Nikolic
public void clickOnRegisterLink() { driver.findElement(RegisterNav).click(); }

1 usage  👤 Mirza Nikolic
public void clickOnGenderMale() { driver.findElement(GenderMale).click(); }

1 usage  👤 Mirza Nikolic
public void setFirstName(String firstName) { driver.findElement(FirstName).sendKeys(firstName); }

1 usage  👤 Mirza Nikolic
public void setLastName(String lastName) { driver.findElement(LastName).sendKeys(lastName); }

1 usage  👤 Mirza Nikolic
public void setEmail(String email) { driver.findElement(Email).sendKeys(email); }

1 usage  👤 Mirza Nikolic
public void setCompany(String company) { driver.findElement(Company).sendKeys(company); }

1 usage  👤 Mirza Nikolic
public void setStreetAddress(String streetAddress) { driver.findElement(StreetAddress).sendKeys(streetAddress); }

1 usage  👤 Mirza Nikolic
public void setZipPostalCode(String zipPostalCode) { driver.findElement(ZipPostalCode).sendKeys(zipPostalCode); }

1 usage  👤 Mirza Nikolic
public void setCity(String city) { driver.findElement(City).sendKeys(city); }

1 usage  👤 Mirza Nikolic
public void setPhone(String phone) { driver.findElement(Phone).sendKeys(phone); }

1 usage  👤 Mirza Nikolic
public void clickOnNewsletter() { driver.findElement(Newsletter).click(); }
```

Shop page:

```
private final By SortInput = By.id("products-orderby");
1 usage
private final By ProductCartButton = By.className("product-box-add-to-cart-button");
1 usage
private final By NoResultMsg = By.className("no-result");

4 usages  Mirza Nikolic
public ShopPage(WebDriver _driver) { driver = _driver; }

1 usage  Mirza Nikolic
public void clickOnSortInput() { driver.findElement(SortInput).click(); }

1 usage  Mirza Nikolic
public void sortByDescendingNameOrder() {
    Select select = new Select(driver.findElement(SortInput));
    select.selectByValue("6");
}

1 usage  Mirza Nikolic
public void clickOnAddToCartButton() { driver.findElement(ProductCartButton).click(); }

1 usage  Mirza Nikolic
public void inputMinPrice(){
    JavascriptExecutor js = (JavascriptExecutor) driver;
    js.executeScript( script: "document.getElementById('slider').setAttribute('data-selectedminvalue', 500)");
}

no usages  Mirza Nikolic
public void inputMaxPrice(){
    JavascriptExecutor js = (JavascriptExecutor) driver;
    js.executeScript( script: "document.getElementById('slider').setAttribute('data-selectedmaxvalue', '1000')");
}

1 usage  Mirza Nikolic
public Boolean noResults() {
    return driver.findElement(NoResultMsg).getText().trim().equals("No products were found that matched your criteria.");
}
```

Wishlist page:

```
3 usages  Mirza Nikolic *
public class WishlistPage {
    3 usages
    WebDriver driver;

    1 usage
    private final By UpdateWishListButton = By.className("update-wishlist-button");
    1 usage
    private final By NoData = By.className("no-data");

    1 usage  Mirza Nikolic
    public WishlistPage(WebDriver _driver) { driver = _driver; }

    1 usage  Mirza Nikolic
    public void clickOnUpdateWishListButton() { driver.findElement(UpdateWishListButton).click(); }

    1 usage  Mirza Nikolic
    public Boolean isWishlistEmpty() {
        return driver.findElement(NoData).getText().trim().equals("Vaša lista želja trenutno je prazna!");
    }
}
```

New articles page:

```
3 usages
public class NewArticlesPage {
    2 usages
    WebDriver driver;

    1 usage
    private final By viewAllButton = By.linkText("Pogledaj sve");

    1 usage
    public NewArticlesPage(WebDriver _driver) {
        driver = _driver;
    }

    1 usage
    public void clickOnViewAll() {
        driver.findElement(viewAllButton).click();
    }
}
```

3.1. Registration test

For the purpose of creating orders and purchases, the user wants to create profile on the platform. This feature provides many advantages for the users as it opens doors for all additional and best Genelec.ba services.

Test Name: Register Test				
Description: Create an account for the user				
Pre-condition(s): The only condition is that you need to insert a valid e-mail in the test file which has not already been registered on genelec.ba				
Test Steps:	Test Data:	Expected Result:	Actual Result:	Status:
1. 1. Go to the landing page and click on the 'my profile' button 2. Click on the Register button 3. Enter Gender, first name, last name, email, company, street address, zip postal code, city, phone 4. uncheck the news letter option 5. Enter password and confirm password 6. Click on Register button	<ul style="list-style-type: none">- Male- Selenium- Project- mirza.nikolic@stu.ibu.edu.ba- Burch- Kosevo 24- 71000- Sarajevo- 062786574- mirzastudent123- mirzastudent123	The user is successfully registered	The user is successfully registered	PASS
Notes:				

```
1 Mirza Nikolic
2 @Test
3 public void Run() {
4     driver.get("https://www.genelec.ba");
5     try {
6         Thread.sleep(6000);
7         driver.manage().window().maximize();
8         registerPage.clickOnHeaderLinksWrapper();
9         registerPage.clickOnRegisterLink();
10        registerPage.clickOnGenderMale();
11        registerPage.setFirstName("Selenium");
12        registerPage.setLastName("Project");
13        registerPage.setEmail("mirza.nikolic@stu.ibu.edu.ba");
14        registerPage.setCompany("Burch");
15        registerPage.setStreetAddress("Kosevo 24");
16        registerPage.setZipPostalCode("71000");
17        registerPage.setCity("Sarajevo");
18        registerPage.setPhone("062 786-574");
19        registerPage.clickOnNewsletter();
20        registerPage.setPassword("mirzastudent123");
21        registerPage.setConfirmPassword("mirzastudent123");
22        registerPage.clickConfirmButton();
23    } catch (Exception e) {
24        System.out.println(e.getMessage());
25    }
26 }
```

3.2. Login test

The user that has already created the account, can login whenever needed.

Test Name: Login test				
Description: Login to the account				
Pre-condition(s): have an account				
Test Steps:	Test Data:	Expected Result:	Actual Result:	Status:
1. Go to the landing page - mojprofil and click on the 'Login' button 2. Enter an email and password 3. Click on the remember me 5. Click again on the login button	<input type="checkbox"/> Email: mirza.nikolic@stu.ibu.edu.ba <input type="checkbox"/> Password: mirzastudent123	To successfully login to the account and navigate to the home page.	The user is successfully logged in and redirected to the home page.	PASS
Notes: The user must provide correct data, otherwise the 'incorrect credentials' will be shown.				

```

Mirza Nikolic
@Test
public void Run() throws InterruptedException {
    driver.get("https://www.genelec.ba");
    try {
        Thread.sleep( millis: 6000);
    } catch (Exception exception) {
        System.out.println(exception.getMessage());
    }
    driver.manage().window().maximize();
    loginPage.clickOnHeaderLinksWrapper();
    loginPage.clickOnLoginLink();
    loginPage.setEmail("mirza.nikolic@stu.ibu.edu.ba");
    loginPage.setPassword("mirzastudent123");
    loginPage.clickOnRememberMe();
    loginPage.loginSubmit();
    Thread.sleep( millis: 3000);
    assertEquals(driver.getCurrentUrl(), actual: "https://genelec.ba/login?returnUrl=%2F");
}
}

```

3.3. Category Test

Test Name: Category Cart				
Description: Traverse category page and add an item to cart				
Pre-condition(s):				
Test Steps:	Test Data:	Expected Result:	Actual Result:	Status:
<ol style="list-style-type: none">1. Click on the Category Button2. Click on the searched for Mobile Device3. Add it to cart4. Assure that the item is added to cart	<ul style="list-style-type: none">- "Genelac" link- Mobilnitel efon- Apple iPhone 11 64GB White	Category page will be traversed for the desired device and it will add it to cart	The desired item is added to cart	PASS
Notes:				

```

Mirza Nikolic
@Test
public void Run() {
    driver.get("https://www.genelec.ba");
    try {
        Thread.sleep( millis: 6000);
        driver.manage().window().maximize();
        homePage.clickOnPhonesCategory();
        homePage.clickOnMobilePhone();
        homePage.clickOnCartButton();
        Thread.sleep( millis: 2000);
        assertTrue(homePage.isItemAddedToCart());
    } catch (Exception exception) {
        System.out.println(exception.getMessage());
    }
}
}
```


3.4. Search Test

Test Name: Search				
Description: Search for an item and add it to cart				
Pre-condition(s):				
Test Steps:	Test Data:	Expected Result:	Actual Result:	Status:
<ol style="list-style-type: none">1. Click on Search Bar2. Type in Iphone 133. Click on the searched item4. Add it to cart5. Check if it is added to cart	https://www.genelec.ba - Iphone 13	The searched for item is added to cart	The searched for item is added to cart	PASS
Notes:				

```
└─ Mirza Nikolic
@Test
public void Run() {
    driver.get("https://www.genelec.ba");
    try {
        Thread.sleep( millis: 6000);
        driver.manage().window().maximize();
        Thread.sleep( millis: 1000);
        homePage.clickOnSearchBar();
        homePage.setSearchText("Iphone 13");
        Thread.sleep( millis: 3000);
        homePage.clickOnSearchItem();
        homePage.clickOnCartButton();
        Thread.sleep( millis: 3000);
        assertTrue(homePage.isItemAddedToCart());
    } catch (Exception exception) {
        System.out.println(exception.getMessage());
    }
}
```

3.5. LogOut Test

Users can log out whenever they want and after they have finished using Genelec.ba 's services.

Test Name: LogOut				
Description: Log out from the profile through My account				
Pre-condition(s):				
Test Steps:	Test Data:	Expected Result:	Actual Result:	Status:
<ol style="list-style-type: none">1. Repeat the Log in Process2. Click on “MojRačun” in the Header Links Wrapper3. Click on Log Out4. Assert that the user is logged out	<ul style="list-style-type: none">- Genelec link- Log in data	The user is successfully logged in and then successfully logged out.	The user is successfully logged in and then successfully logged out.	PASS
Notes:				

```
Mirza Nikolic
@Test
public void Run() {
    driver.get("https://www.genelec.ba");
    try {
        Thread.sleep( millis: 6000);
        driver.manage().window().maximize();
        homePage.clickOnHeaderLinksWrapper();
        loginPage.login();
        Thread.sleep( millis: 5000);
        homePage.clickOnHeaderLinksWrapper();
        homePage.clickOnLogout();
        Thread.sleep( millis: 2000);
        assertEquals( expected: "https://genelec.ba/", driver.getCurrentUrl());
    } catch (Exception exception) {
        System.out.println(exception.getMessage());
    }
}
```

3.6. Decrease one Test

Test Name: Decrease one				
Description: Decreasing the amount of a single item in the shopping cart by one				
Pre-condition(s): The user needs to have a multiple amount of a single Item in the cart before the test starts				
Test Steps:	Test Data:	Expected Result:	Actual Result:	Status:
<ol style="list-style-type: none"> 1. Repeat the login process 2. Click on the main cart button 3. Get the current amount of the item in the cart 4. Decrease the amount by one 5. Update the changes 6. Assert that the amount has been decreased by one 	https://www.genel.ec.ba Log in data	The amount of the item in cart will be decreased by one	The amount of the item in the cart has been decreased by one	PASS
Notes:				

```

@Test
public void Run() {
    driver.get("https://www.genelec.ba");
    try {
        Thread.sleep( millis: 6000);
        driver.manage().window().maximize();
        loginPage.login();
        Thread.sleep( millis: 5000);
        homePage.clickOnMainCart();
        int beforeQty = cartPage.getQtyInput();
        cartPage.decreaseQtyInput(1);
        cartPage.clickOnUpdateCart();
        Thread.sleep( millis: 1000);
        int afterQty = cartPage.getQtyInput();
        assertEquals(beforeQty, actual: afterQty + 1);
    } catch (Exception exception) {
        System.out.println(exception.getMessage());
    }
}

```

3.7. Wish list display Test

Test Name: Wish List Display				
Description: Open the Wish List				
Pre-condition(s):				
Test Steps:	Test Data:	Expected Result:	Actual Result:	Status:
<ol style="list-style-type: none">1. Click on Header Links Wrapper2. Click on Wish List Button in the drop down menu3. Assert that the page you are on is truly the wish list page	<ul style="list-style-type: none">- https://www.genelec.ba- https://genelec.ba/wishlist	The program will open the wish list page	The program has opened the wish list page	PASS
Notes:				

```

Mirza Nikolic
@Test
public void Run() {
    driver.get("https://www.genelec.ba");
    try {
        Thread.sleep( millis: 6000);
        driver.manage().window().maximize();
        homePage.clickOnHeaderLinksWrapper();
        homePage.clickOnWishList();
        assertEquals(driver.getCurrentUrl(), actual: "https://genelec.ba/wishlist");
    } catch (Exception exception) {
        System.out.println(exception.getMessage());
    }
}
}
```

3.8. Product Sorting Test

Test Name: Product sorting				
Description: Sort the products by name				
Pre-condition(s):				
Test Steps:	Test Data:	Expected Result:	Actual Result:	Status:
<ol style="list-style-type: none">1. Click on Sorting Drop down menu2. Put Sort by Name in reverse	"https://genelec.ba/notebookc-106"	The items are sorted by name in reverse	The items are sorted by name in reverse	PASS
Notes:				

```

Mirza Nikolic
@Test
public void Run() {
    driver.get("https://genelec.ba/notebookc-106");
    try {
        Thread.sleep( millis: 6000);
        driver.manage().window().maximize();
        shopPage.clickOnSortInput();
        shopPage.sortByDescendingNameOrder();
    } catch (Exception exception) {
        System.out.println(exception.getMessage());
    }
}
}
```

3.9. Add to cart from shop page

Test Name: Add to card from shop page				
Description: Add an item from the page to the cart				
Pre-condition(s):				
Test Steps:	Test Data:	Expected Result:	Actual Result:	Status:
<ol style="list-style-type: none">1. Add the first item in the opened list to cart2. Assert that the item is added to cart	"https://genelec.ba/notebookc-106"	The item is added to cart	The item is added to cart	PASS
Notes:				

```

Mirza Nikolic
@Test
public void Run() {
    driver.get("https://genelec.ba/notebookc-106");
    try {
        Thread.sleep( millis: 6000);
        driver.manage().window().maximize();
        shopPage.clickOnAddToCartButton();
        Thread.sleep( millis: 2000);
        assertTrue(homePage.isItemAddedToCart());
    } catch (Exception exception) {
        System.out.println(exception.getMessage());
    }
}
}
```

3.10. Delete from Cart Test

Test Name: Delete from Cart				
Description: Delete the item that is in the cart				
Pre-condition(s): One item is already in the cart when the test runs				
Test Steps:	Test Data:	Expected Result:	Actual Result:	Status:
<ol style="list-style-type: none">1. Repeat the Log in process2. Open the cart3. Check the delete checkmark on the item4. Update the cart so that the item is deleted5. Assert that the item is deleted	https://www.genelec.ba <ul style="list-style-type: none">- login data- https://genelec.ba/cart	The item in the cart will be deleted	The item in the cart was deleted	PASS
Notes:				

```
Mirza Nikolic
@Test
public void Run() {
    driver.get("https://www.genelec.ba");
    try {
        Thread.sleep(6000);
        driver.manage().window().maximize();
        LoginPage.login();
        Thread.sleep(5000);
        driver.get("https://genelec.ba/cart");
        Thread.sleep(5000);
        cartPage.clickOnDeleteCheck();
        cartPage.clickOnUpdateCart();
        assertTrue(cartPage.isCartEmpty());
    } catch (Exception exception) {
        System.out.println(exception.getMessage());
    }
}
```

3.11. Cart Increase Item by two Test

Test Name: Cart In crease Item by two				
Description: Increase the quantity of a single item in the shopping cart by 2				
Pre-condition(s):Have one item in the shopping cart				
Test Steps:	Test Data:	Expected Result:	Actual Result:	Status:
<ol style="list-style-type: none">1. Repeat the log in process2. Click on the main cart button3. Get the current amount of the item in the cart4. Increase the amount by 25. Update the changes that were made6. Assure that the amount was increased by 2	https://www.genelec.ba	The amount of the item in the shopping cart should be increased by 2	The amount of the item in the shopping cart was increased by 2	PASS
Notes:				

```

Mirza Nikolic
@Test
public void Run() {
    driver.get("https://www.genelec.ba");
    try {
        Thread.sleep(6000);
        driver.manage().window().maximize();
        LoginPage.login();
        Thread.sleep(5000);
        homePage.clickOnMainCart();
        int beforeQty = cartPage.getQtyInput();
        cartPage.IncreaseQtyInput(2);
        Thread.sleep(1000);
        cartPage.clickOnUpdateCart();
        Thread.sleep(1000);
        int afterQty = cartPage.getQtyInput();
        assertEquals(afterQty, beforeQty + 2);
    } catch (Exception exception) {
        System.out.println(exception.getMessage());
    }
}
}
```


3.12. Subscribe to News letter Test

Test Name: Subscribe to News letter				
Description: Check that the newsletter will not accept an invalid e mail address				
Pre-condition(s):				
Test Steps:	Test Data:	Expected Result:	Actual Result:	Status:
<ol style="list-style-type: none">1. Set the newsletter e mail in the newsletter input section2. Click on the subscribe button3. Assert that the E mail is not accepted (not a valid e mail)	<ul style="list-style-type: none">- https://genelec.ba- Invalid e-mail	The page should return that the e mail entered was invalid	The page indeed showed that the e mail entered is invalid	FAILED
Notes:				

```
Mirza Nikolic
@Test
public void Run() {
    driver.get("https://genelec.ba");
    try {
        Thread.sleep( millis: 6000);
        driver.manage().window().maximize();
        homePage.setNewsletterEmail("gossokutrasi-614yopmail.com");
        homePage.clickOnSubscribeButton();
        Thread.sleep( millis: 1000);
        assertTrue(homePage.isSubscriptionFailed());
    } catch (Exception exception) {
        System.out.println(exception.getMessage());
    }
}
```

3.13.No Products Found Test

Test Name: No products found				
Description: Test that the page will not return items for a random invalid search string				
Pre-condition(s):				
Test Steps:	Test Data:	Expected Result:	Actual Result:	Status:
<ol style="list-style-type: none">1. Click on the Search bar2. Set the text to the invalid search string3. Click on the search button4. Assert that there are no search results	<ul style="list-style-type: none">- https://genelec.ba- Search Text	The page should not return any items for the invalid search string because an item like that does not exist	The page indeed did not return any results	PASS
Notes:				

```
Mirza Nikolic
@Test
public void Run() {
    driver.get("https://genelec.ba");
    try {
        Thread.sleep( millis: 6000);
        driver.manage().window().maximize();
        homePage.clickOnSearchBar();
        homePage.setSearchText("blablabla");
        homePage.clickOnSearchButton();
        Thread.sleep( millis: 3000);
        assertTrue(shopPage.noResults());
    } catch (Exception exception) {
        System.out.println(exception.getMessage());
    }
}
```

3.14. Get tags test

Test Name: Find tags				
Description: Test will navigate to the tags list and assert link changes during the process.				
Pre-condition(s): /				
Test Steps:	Test Data:	Expected Result:	Actual Result:	Status:
<ul style="list-style-type: none">- Click on the “Novi artikli” tab in the menu bar- Click on “Pogledaj sve”- Assert that the link has changed to: https://genelec.ba/producttag/all	<ul style="list-style-type: none">- https://genelec.ba- https://genelec.ba/producttag/all	The test should assert that the current url is equal to: https://genelec.ba/producttag/all after finding new tags.	The page changes were indeed behaving as expected.	PASS
Notes:				

```
Mirza Nikolic *
@Test
public void Run() {
    driver.get("https://genelec.ba/");
    try {
        Thread.sleep(1000);
        driver.manage().window().maximize();
        homePage.clickOnNewArticlesButton();
        Thread.sleep(2000);
        newArticlesPage.clickOnViewAll();
        Thread.sleep(1000);
        assertEquals("expected: \"https://genelec.ba/producttag/all\", driver.getCurrentUrl());
    } catch (Exception exception) {
        System.out.println(exception.getMessage());
    }
}
```

3.15. Remove from Wish List Test

Test Name: Remove From Wish List				
Description: Delete an item from the wish list				
Pre-condition(s): Wish List Needs to have an item in it				
Test Steps:	Test Data:	Expected Result:	Actual Result:	Status:
<ol style="list-style-type: none">1. Repeat the Log in Process2. Go to Wish List page3. Check the Item Delete in Wish List Button4. Update the wish list5. Assert that the item in the wish list has been deleted	<ul style="list-style-type: none">- https://www.genelec.ba- https://genelec.ba/wishlist	The Item in the wish list is deleted	The Item in the wish list is deleted	PASS
Notes:				

```
@Test
public void Run() {
    driver.get("https://genelec.ba");
    try {
        Thread.sleep( millis: 6000);
        driver.manage().window().maximize();
        loginPage.login();
        Thread.sleep( millis: 2000);
        driver.get("https://genelec.ba/wishlist");
        Thread.sleep( millis: 2000);
        cartPage.clickOnDeleteCheck();
        wishlistPage.clickOnUpdateWishlistButton();
        assertTrue(wishlistPage.isWishlistEmpty());
    } catch (Exception exception) {
        System.out.println(exception.getMessage());
    }
}
```

4. Conclusion

4.1. Testing Summary

Testing Tool	Total Tests	Passed Tests	Failed Tests
JUnit5 and Selenium	15	#15	#0

4.2. Final Thoughts

We have learned a lot about automation tests during this period of time. We think that our tests are really nicely implemented because firstly they work, and secondly the base functionalities are separated in different classes which makes everything look clean and really clear.

