**SOFTWARE VERIFICATION, VALIDATION AND TESTING**

TESTING DOCUMENTATION

PROJECT NAME

Prepared by:

**Nejra Jerlagic**

**Nadja Cizmic  
Haris Sacic**

Proposed to:

**Samed Jukić, Assist. Prof. Dr.**

**Aldin Kovačević, Teaching Assistant**

22.10.2023.

Table of Contents

[1. Introduction 3](#_Toc125237668)

[1.1. About the Project 3](#_Toc125237669)

[1.2. Project Functionalities and Screenshots 3](#_Toc125237670)

[2. Test Plan 8](#_Toc125237671)

[2.1. Scope 8](#_Toc125237672)

[2.2. Testing Environment and Tools 8](#_Toc125237673)

[3. Test Execution 8](#_Toc125237674)

[3.1. Registratiom 8](#_Toc125237675)

[3.2. Login 9](#_Toc125237676)

[3.3. Forgot Password 10](#_Toc125237677)

[3.4. Back Functionality 10](#_Toc125237678)

[3.4. Filters and Bars 11](#_Toc125237679)

[3.5. Shopping Bag 13](#_Toc125237680)

[3.5. Customer Support 14](#_Toc125237681)

[3.5. Footer 14](#_Toc125237682)

[9. Conclusion 16](#_Toc125237683)

[9.2. Final Thoughts 16](#_Toc125237684)

# 1. Introduction

## 1.1. About the Project

We decided to test <https://www.mojbrend.ba>. This is a web application for purchasing clothes in Bosnia and Herzegovina. The brand has different web pages for other countries as well. It has basic authentication functionalities: Login, Registration. Password reset. It also has a shopping bag, filters, search function. The user can shop within different brands: Calvin Klein, Tommy Hilfiger, Replay, US Polo and many others. There is also a Chat option for contacting the brand and asking for any needed help.

## 1.2. Project Functionalities and Screenshots

LOGIN:

Graphical user interface, text, application

Description automatically generated with medium confidence

REGISTRATION:

Graphical user interface, text, application, chat or text message, email

Description automatically generated

RESET PASSWORD:

Graphical user interface, text, application

Description automatically generated

BRANDS BAR:

Table

Description automatically generated

SEARCH BAR:

Background pattern

Description automatically generated

FILTER:

Graphical user interface

Description automatically generated

SHOPPING BAG:

Graphical user interface, application

Description automatically generated with medium confidence

CHAT:

Graphical user interface, text, application, chat or text message

Description automatically generated

FAST LINKS:

Graphical user interface, application

Description automatically generated

OTHER COUNTRIES:

Text, application

Description automatically generated

# 2. Test Plan

## 2.1. Scope

We focused on testing the key functions, starting with login as the most important functionality. When we made sure we covered login, registration, forgot password and negative tests for the same, we then turned to testing out main functions for buying an item, such as shopping bag, drop downs, search bar and the filters for the items. The next goal was customer support, which is the Chat option. Then we covered the fast links and the links to the page for other countries.

We paid our attention to testing this app as a whole, with the biggest focus on the major functionalities. The details were the second priority.

## 2.2. Testing Environment and Tools

We were testing this app using Java programming language, with JUnit and Selenium framework.

# 3. Test Execution

In the “Test Execution” section, you need to list your **test scenarios** and **test cases**. *Test scenario* is a statement whose purpose is to inform what particular area of the application will be tested. Test scenarios consist of multiple *test cases*, which test specific parts of an application. You can use the following template approach for test scenarios and test cases, but you can also use a custom template or your own “design”, as long as you have all the necessary information.

## 3.1. Registration

A user want to register for the application.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Registration | | | | |
| **Description:**  This is a test for registration to the app | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Load base url  2. Click on “profile”  3. Click on “Registrujte se”  4. Enter $name  5. Enter $surname  6. Enter $email  6. Enter $password  7. Click on “Kreiraj korisnicki profil”  8. The user is registered | **Test Data:**  $name:Nadja  $surname:Cizmic  $email: nadjacizmic@hotmail.com | **Expected Result:**  The user is registered. | **Actual Result:**  The user is registered. | **Status:**  PASS |
| **Notes:** | | | | |



## 3.2. Login

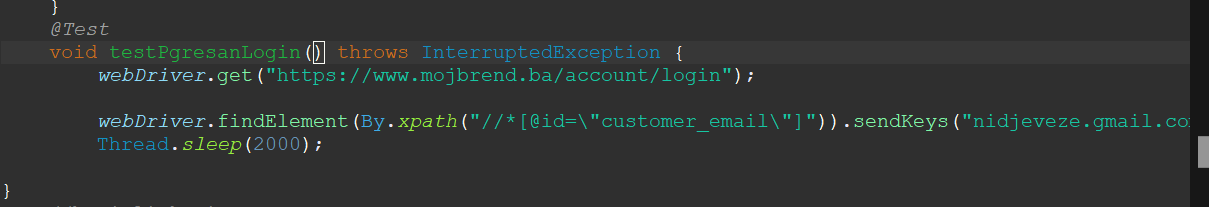
A user wants to log in the application.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Login | | | | |
| **Description:**  This is a positive test for login functionality | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Load base url  2. Click on “profile”  3. Enter email  4. Enter password  5. Click “Prijavi se”  6. The user is logged in  7. Click “Povratak na kupovinu”  8. Click “Odjava” | **Test Data:**  $email:nadjacizmic@hotmail.com  $password: Lozinka123 | **Expected Result:**  The user is logged in. | **Actual Result:**  The user is logged in. | **Status:**  PASS |
| **Notes:** \ | | | | |
|  | | | | |

Text

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Login Negative | | | | |
| **Description:**  This is a negative test for login functionality | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Load base url  2. Click on “profile”  3. Enter email  4. Enter password  5. Click “Prijavi se”  6. The user is stays on the same page | **Test Data:**  $email:nadjacizmic  $password: Lozinka123 | **Expected Result:**  The user is not logged in. | **Actual Result:**  The user is not logged in. | **Status:**  PASS |
| **Notes:** \ | | | | |



## 3.3. Forgot Password

A user forgets the password for the application.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Forgot Password | | | | |
| **Description:**  This is a test for dealing with a forgotten password | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Load base url  2. Click on “profile”  3. Enter email  4. Click on “Zaboravili ste sifru?”  5. Enter $email2  6. Click “Izvrsi reset”  6. The user is stays on the same page  7. Enter $email  8. Click “Izvrsi reset”  9. The page reports that the email is sent | **Test Data:**  $email:nadjacizmic@hotmail.com  $email2:nidjeveze@hotmail.com | **Expected Result:**  Web page says that the reset is sent. | **Actual Result:**  Web page says that the reset is sent. | **Status:**  PASS |
| **Notes:** | | | | |

Text

Description automatically generated with low confidence

## 3.4. Back Functionality

A user wants to go back to shopping.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Going Back and Main brands | | | | |
| **Description:**  This is a test for checking main brands and returning to the start page | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Load base url  2. Click on “Calvin Klein”  3. “Calvin Klein muski I zenski proizvodi” is shown  4. Click on “Tommy Hilfiger”  5. “Tommy Hilfiger” muski I zenski proizvodi” is shown  6. Click on “Replay”  7. “Replay” muski I zenski proizvodi” is shown  8. Click on “US Polo Assn.”  9. “US Polo Assn.” muski I zenski proizvodi” is shown | **Test Data:** | **Expected Result:**  The user is returned to a base url. | **Actual Result:**  The user is returned to a base url. | **Status:**  PASS |
| **Notes:** | | | | |



## 3.4. Filters and Bars

A user wants to search the options for shopping.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Search Bar | | | | |
| **Description:**  This is a test for searching the web app. | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Load base url  2. Click on Search bar  3. Enter $brand  4. Click on the search icon on the right  5. Tommy Hilfiger items show | **Test Data:**  $brand:Tommy Hilfiger | **Expected Result:**  Tommy Hilfiger items show. | **Actual Result:**  Tommy Hilfiger items show. | **Status:**  PASS |
| **Notes:** | | | | |

Text

Description automatically generated

A user wants to filter the options for shopping.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Filter | | | | |
| **Description:**  This is a test for filtering the options for shopping. | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Load base url  2. Click on the “Sezonsko snizenje” window  3. Tick the $brand  4. The first item is by the $brand  5. Untick the $brand  6. Tick the $brand2  The first item is by the $brand2 | **Test Data:**  $brand:Calvin Klein  $brand2: Diesel | **Expected Result:**  The correct bands are shown after filtering. | **Actual Result:**  The correct bands are shown after filtering. | **Status:**  PASS |
| **Notes:** | | | | |

Text

Description automatically generated

A user wants to to find the right size.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Size | | | | |
| **Description:**  This is a test for choosing the right size. | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Load base url  2. Click on the “Sezonsko snizenje” window  3. Click on “Velicina”  4. Tick the $size  5. Click on the first item  6. The size of the first item is $size | **Test Data:**  $size:S | **Expected Result:**  The correct size is shown for the product. | **Actual Result:**  The correct size is shown for the product. | **Status:**  FAIL |
| **Notes:** | | | | |

Text

Description automatically generated

A user wants to search the other brands.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Other Brands | | | | |
| **Description:**  This is a test for searching the other brands. | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Load base url  2. Navigate to the dropdown “Vise Brendova” and click on “Diesel” | **Test Data:** | **Expected Result:**  Brand “Diesel” is in the dropdown for “Vise brendova”. | **Actual Result:**  Brand “Diesel” is in the dropdown for “Vise brendova”. | **Status:**  PASS |
| **Notes:** | | | | |
|  | | | | |

Text

Description automatically generated

## 3.5. Shopping Bag

A user wants to check the shopping bag.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Shopping Bag | | | | |
| **Description:**  This is a test for shopping bag | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Load base url  2. Click “Prijava”  3. Click “Povratak na kupovinu”  4. Click “Calvine Klein  5. Click some article and then click “Dodaj u korpu”  6. Click “Nastavi kupovinu”  7. Click some article and then click “Dodaj u korpu”  8. Click on “Korpa” and assert that there are 2 items in it | **Test Data:** | **Expected Result:**  For chart to have 2 items in it | **Actual Result:**  Chart has 2 items | **Status:**  PASS |
| **Notes:** | | | | |

Text

Description automatically generated

A user wants to buy instantly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Buy Instantly | | | | |
| **Description:** | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Load base url  2. Click “Prijava”  3. Click “Povratak na kupovinu”  4. Click “Calvine Klein  5. Click some article and then click “Dodaj u korpu”  6. Click “Kupi odmah”  7. Fill contact form  8. Click “Prelazak na dostavu” | **Test Data:** | **Expected Result:**  To buy items in the chart | **Actual Result:**  Items in chart were bought | **Status:**  PASS |
| **Notes:** | | | | |

Text

Description automatically generated

A user wants to remove items from the bag.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Remove From the Bag | | | | |
| **Description:**  This is a test for removing the items from the bag | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Load base url  2. Click “Prijava”  3. Click “Povratak na kupovinu”  4. Click “Tommy Hilfiger  5. Click some article and then click “Dodaj u korpu”  6. Click “Korpa”  7. Click remove buton “x” | **Test Data:** | **Expected Result:**  To remove added item from chart | **Actual Result:**  Item removed | **Status:**  PASS |
| **Notes:** | | | | |

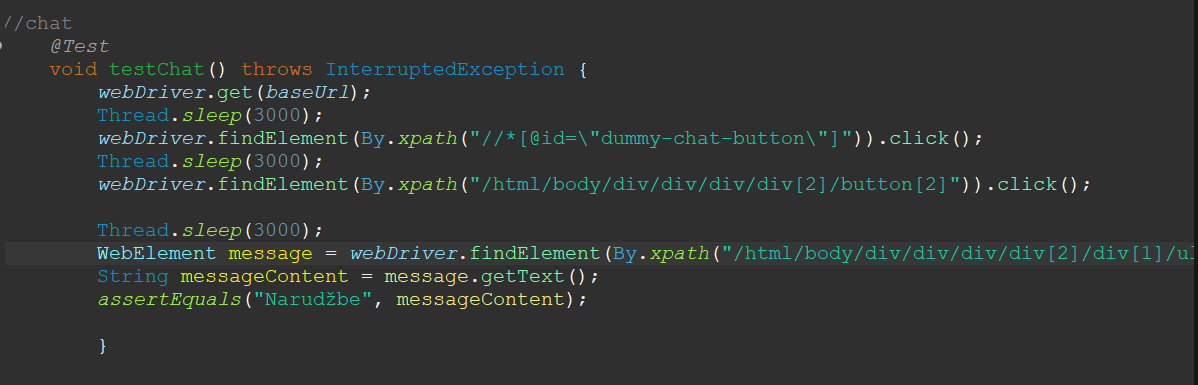
Text

Description automatically generated

## 3.5. Customer Support

A user wants to reach out for help.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Chat | | | | |
| **Description:**  This is a test for the chat functionality. | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Load base url  2. Click on the Chat icon in the down right corner  3. Click on “Troskovi dostave”  4. The message contains the word “Narudzbe” | **Test Data:** | **Expected Result:**  The message contains the word “Narudzbe”. | **Actual Result:**  The message contains the word “Narudzbe”. | **Status:**  FAIL |
| **Notes:** | | | | |
|  | | | | |
|  | | | | |



A user wants to be on the list for newsletter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Newsletter | | | | |
| **Description:**  This is a test for newsletter | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Load base url  2. Find “Zelim biti u trendu!”  3. Enter $email in the box below it  4. Click “Prijavi me” | **Test Data:**  $email: nadjacizmic@hotmail.com | **Expected Result:**  The user sent the email successfully. | **Actual Result:**  The user sent the email successfully. | **Status:**  PASS |
| **Notes:** | | | | |

Graphical user interface

Description automatically generated with low confidence

## 3.5. Footer

A user wants to find web pages of other countries.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Other Countries | | | | |
| **Description:**  This is a test for finding the web pages for other countries. | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Load base url  2. Click on “www.mojbrend.co.rs”  3. The said url opens  4. Click on “www.mojbrend.hr”  5. The said url opens  6. Click on “www.mojbrend.me”  7. The said url opens | **Test Data:** | **Expected Result:**  The correct link opens. | **Actual Result:**  The correct link opens. | **Status:**  FAIL |
| **Notes:** | | | | |

Text

Description automatically generated

A user wants to use the fast links to access another page of the app.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Fast Links | | | | |
| **Description:**  This is a test for fast links. | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Load base url  2. Click on “O nama”  3. The heading “O nama” is shown  4. Click on “Kontakt”  5. The heading “Kontakt” is shown  6. Click on “Politika privatnosti”  7. The heading “Pravila zastite privatnosti” is shown  8. Click on “Opsti prodajni uslovi”  9. The heading “Uvjeti pruzanja usluge” is shown  10. Click on “Kolacici”  11. The heading “Pravna obavijest” is shown | **Test Data:**  $brand:Tommy Hilfiger | **Expected Result:**  The correct link opens. | **Actual Result:**  The correct link opens. | **Status:**  PASS |
| **Notes:** | | | | |

A screenshot of a computer

Description automatically generated with medium confidence

# 9. Conclusion

9.1. Testing Summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Testing Tool** | **Total Tests** | **Passed Tests** | **Failed Tests** |
| Selenium  JUnit  Java | 16 | 13 | 3 |

## 9.2. Final Thoughts

As we worked on automating QA testing for the mojbrend.ba website using Selenium, we learned a lot about the capabilities and limitations of the tool. We discovered that Selenium's ability to interact with web pages in the same way a human user would made it an excellent choice for testing the user interface of the website. We were able to create scripts that navigated through the website, filled out forms, and clicked buttons to ensure that all of the website's features were working properly.

During the process of automating QA testing for the mojbrend.ba website using Selenium was a valuable learning experience. We gained a deeper understanding of the tool and its capabilities, as well as a better appreciation for the importance of automated testing in the software development process.