



# **IT 309 SOFTWARE ENGINEERING**

## **PROJECT DOCUMENTATION**

### **MASIC WORKWEAR**

Prepared by:

**Džanin Mašić**

**Tijana Burazorović**

Proposed to:

**Nermina Durmić, Assist. Prof. Dr.**

**Aldin Kovačević, Teaching Assistant**

18.06.2023.

## Contents

1. Introduction.....	3
1.1. About the Project.....	3
1.2. Project Functionalities and Screenshots.....	3
2. Project Structure.....	8
2.1. Technologies .....	8
2.2. Database Entities.....	8
2.3. Design Patterns .....	9
2.4. Tests.....	10
3. Conclusion .....	15

# 1. Introduction

## 1.1. About the Project

Masic Workwear is an application for inventory management of a textile company. The application allows users to log in and log out, add, edit and delete materials, filter materials, search keywords and add colors. The app also shows total length of materials in each color, as well as a chart of this representation.

The application is deployed at the following link: <http://209.38.206.111/SoftwareEngineering>

## 1.2. Project Functionalities and Screenshots

As mentioned above, the application has following functionalities:

- **Login:**

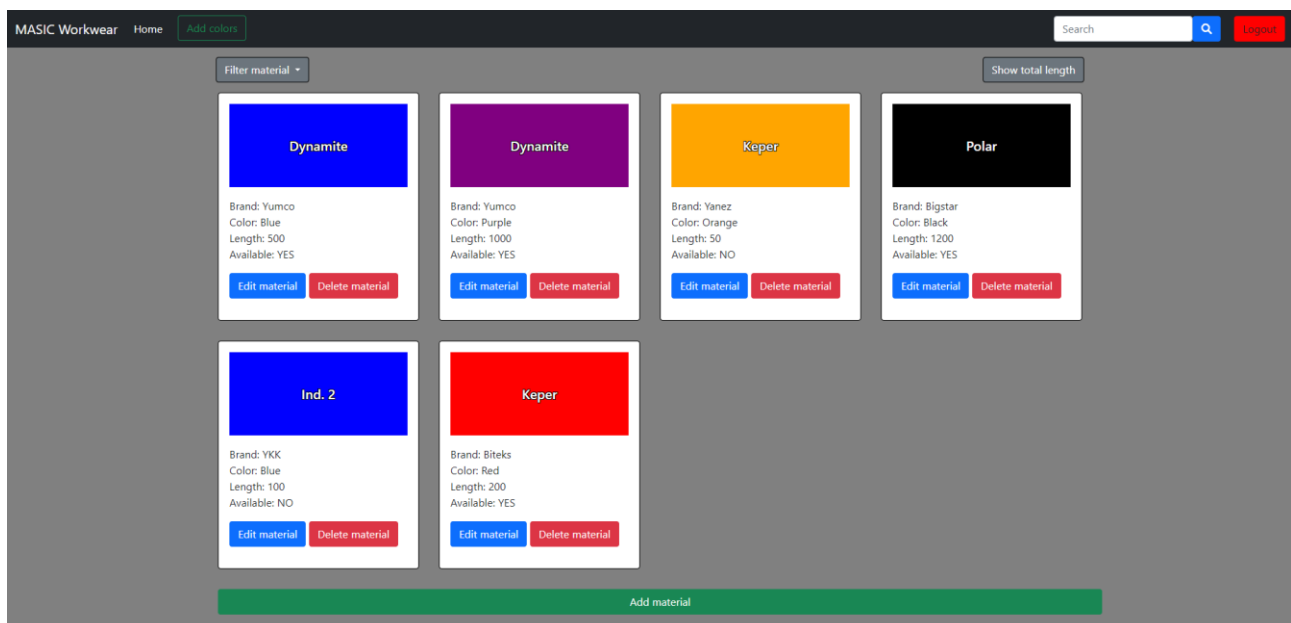
Email address

Password

Login

Here the admin can enter their email and password and log in to the application, which will lead them to the inventory application if the correct credentials are entered and the *Login* button is clicked.

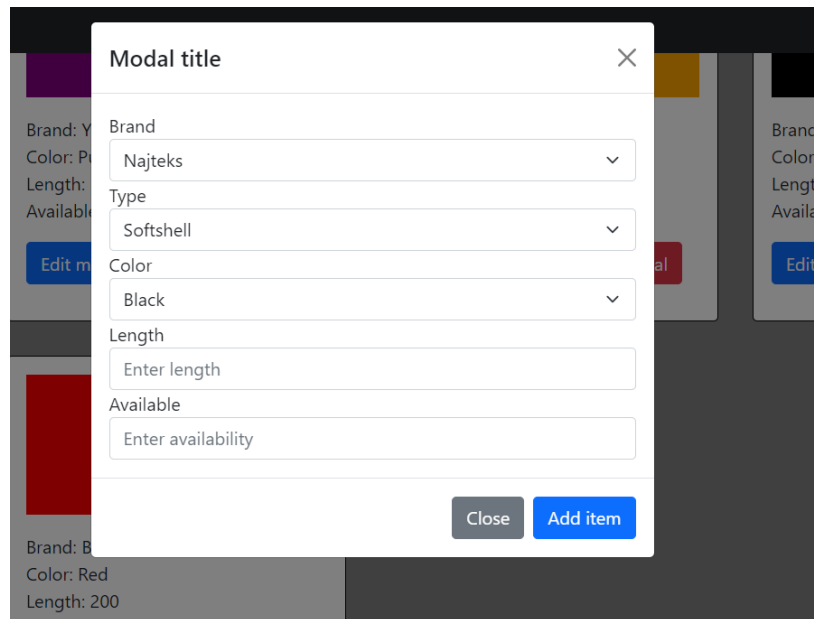
- **Home Page:**



Above is the home page of the application where admin can choose what they want to do. All of the main functionalities can be accessed from here, i.e.: add material, edit material, filter material, delete material, search, add color and show total length.

- **Add Material:**

When the user clicks on the *Add material* button, they are led to the following modal:

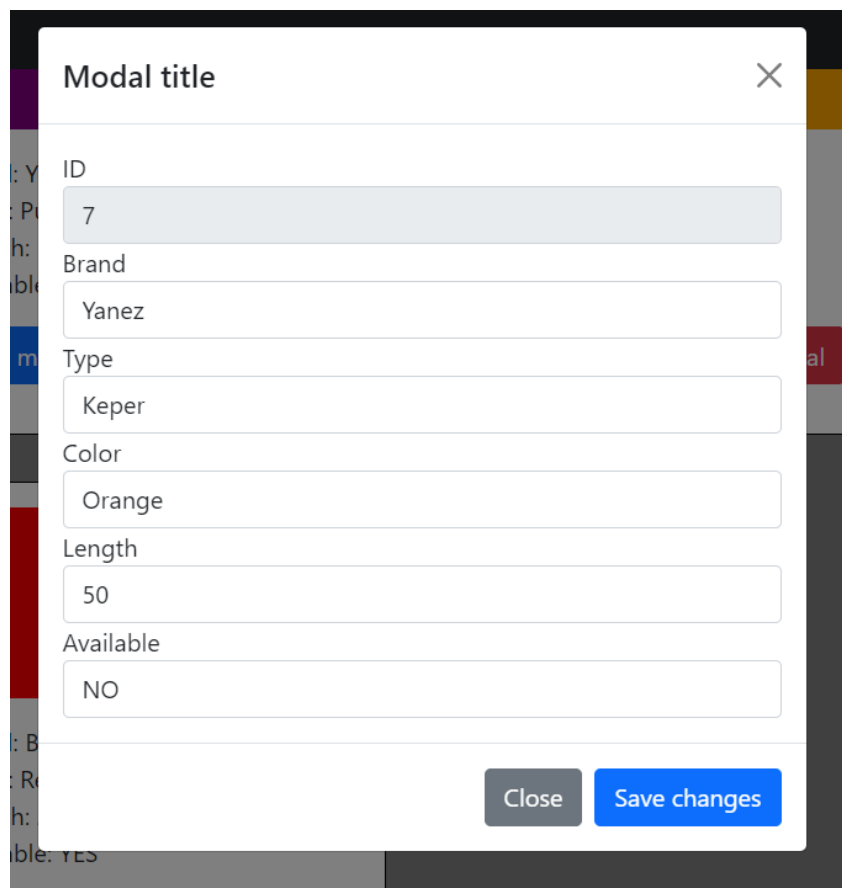


A screenshot of a modal window titled "Modal title" with a close button (X) in the top right corner. The form contains the following fields: "Brand" (a dropdown menu with "Najteks" selected), "Type" (a dropdown menu with "Softshell" selected), "Color" (a dropdown menu with "Black" selected), "Length" (a text input field with the placeholder "Enter length"), and "Available" (a text input field with the placeholder "Enter availability"). At the bottom right of the modal are two buttons: "Close" (grey) and "Add item" (blue).

They can select the brand, type and color of the material here and enter the length and availability of the material.

- **Edit Material:**

When clicking on the *Edit material* button on the material card, the user is shown the following modal:

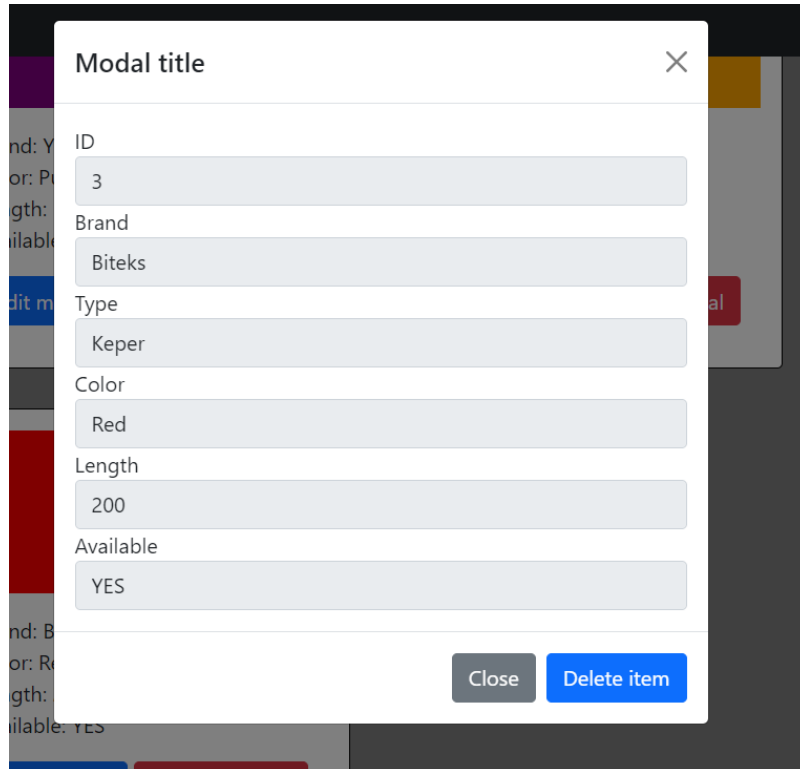


A screenshot of a modal window titled "Modal title" with a close button (X) in the top right corner. The form contains the following fields: "ID" (a text input field with the value "7"), "Brand" (a text input field with the value "Yanez"), "Type" (a text input field with the value "Keper"), "Color" (a text input field with the value "Orange"), "Length" (a text input field with the value "50"), and "Available" (a text input field with the value "NO"). At the bottom right of the modal are two buttons: "Close" (grey) and "Save changes" (blue).

Here they can change the data about the product, with brand, type and color fields being of a select type, length being the number field and Available being the text field.

- **Delete Material:**

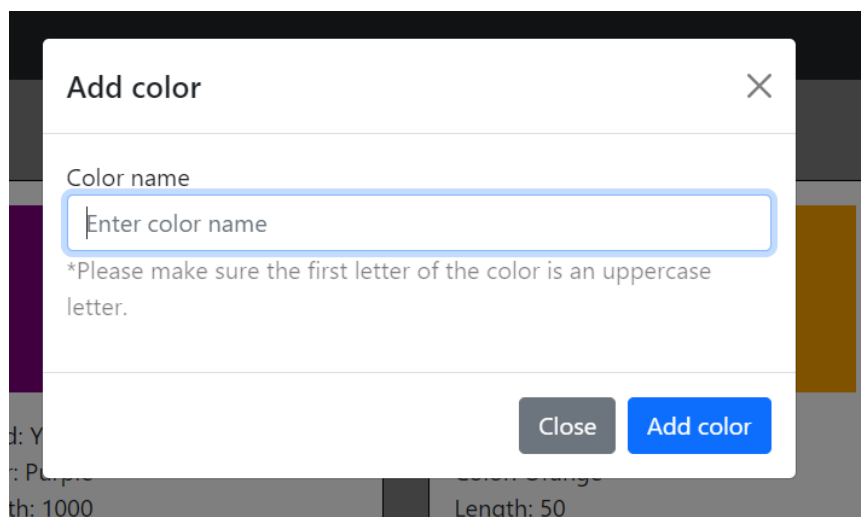
By clicking on the delete material button, the modal for deleting the material is shown, where on clicking the *Delete item* button would delete the material:



A modal window titled "Modal title" with a close button (X) in the top right corner. The modal contains several input fields for editing product data: ID (with value 3), Brand (with value Biteks), Type (with value Keper), Color (with value Red), Length (with value 200), and Available (with value YES). At the bottom right, there are two buttons: "Close" and "Delete item".

- **Add colors:**

In the nav bar there is the *Add colors* button, which will open the following modal:

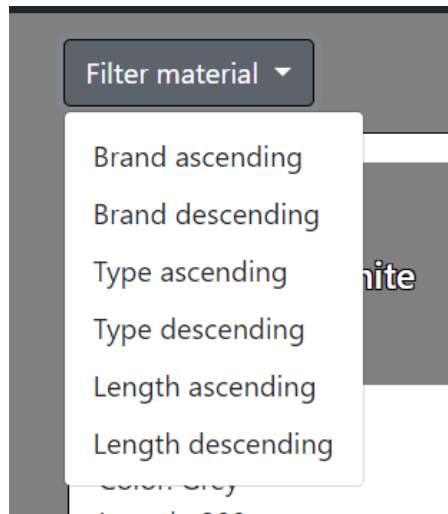


A modal window titled "Add color" with a close button (X) in the top right corner. It features a text input field labeled "Color name" with a placeholder "Enter color name". Below the input field, there is a note: "\*Please make sure the first letter of the color is an uppercase letter." At the bottom right, there are two buttons: "Close" and "Add color".

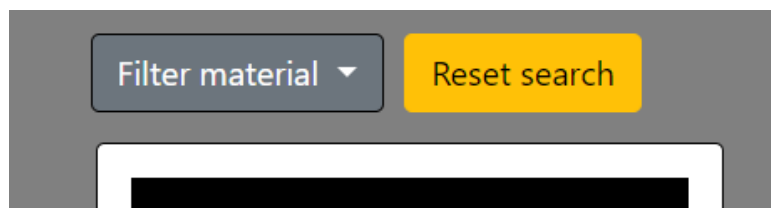
The user can type in the color they want to add and if it is not already in the database, it will be added.

- **Filter Material:**

The user can filter material by clicking on Filter material button. They will be presented with the following options:



When clicking one option, the material will automatically be filtered and the user can choose to reset the filter by clicking on the *Reset search* button:

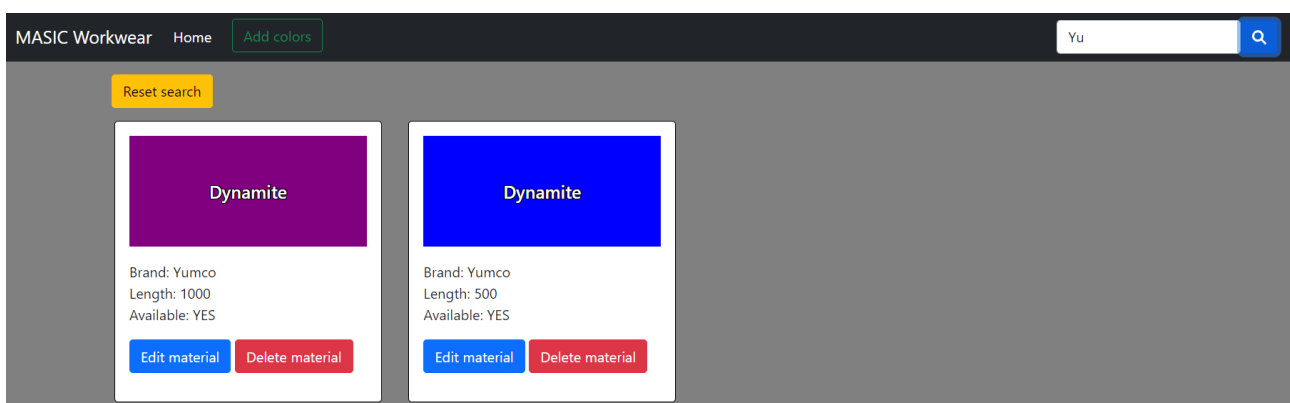


- **Search:**

User can search for the materials by using the search field in the nav bar:



Here, once a search word is entered and the search icon is clicked, the app will display the matching results:



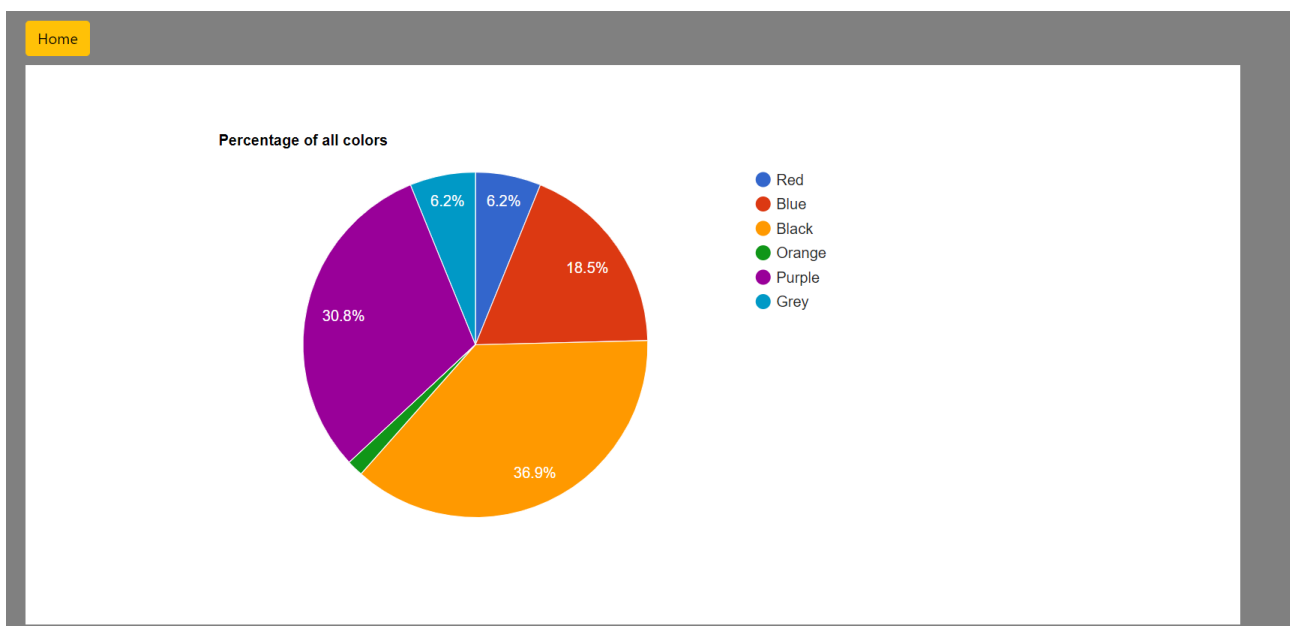
The user can reset the search by clicking on the *Reset search* button.

- **Show Total Length:**

By clicking the *Show total length* button on the home page, the following will be shown:

<a href="#">Home</a> <a href="#">Chart version</a>		
#	Color	Length
1	Red	200
2	Blue	600
3	Black	1200
4	Orange	50
5	Purple	1000
6	Grey	200

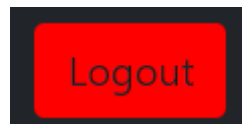
By clicking on the *Chart version* button, the pie chart of all materials will be shown:



When clicking on the *Home* button, the user will be redirected back to the home page.

- **Logout:**

In the nav bar there is a logout button that logs the user out and redirects to the login page.



## 2. Project Structure

### 2.1. Technologies

Masic Workwear is developed using the following technologies, frameworks and tools:

- Flight PHP
- HTML
- CSS
- JavaScript
- firebase/php-jwt
- Swagger
- Google Charts

The project follows PSR-1 coding standard for the backend, implementing the following rules:

- PHP code **MUST** use the long `<?php ?>` tags or the short-echo `<?= ?>` tags; no other tag variations are allowed.
- Files must use UTF-8 encoding.
- Files **SHOULD** either declare symbols (classes, functions, constants, etc.) or cause side-effects (e.g.: generate output, change .ini settings, etc.) but **SHOULD NOT** do both.
  - The phrase “side effects” means execution of logic not directly related to declaring classes, functions, constants, etc., merely from including the file.
- Namespaces and classes **MUST** follow an “autoloading” PSR.
  - Each class is in a file by itself, and is in a namespace of at least one level.
- Class names **MUST** be declared in *StudlyCaps* (PascalCase)
  - every word in the class name starts with a capital letter
  - no spaces or underscores
- Class constants **MUST** be declared in *all uppercase* with *underscore separators*.
- Method names **MUST** be declared in *camelCase*.

Furthermore, the HTML on frontend follows Google’s HTML coding conventions, as specified here:

<https://google.github.io/styleguide/htmlcssguide.html>

### 2.2. Database Entities

Masic Workwear database contains following entities:

- admins → admin users that can log in
- brands → brands that provide materials
- colors → all colors of materials
- material → all materials added by admins
- types → material types



## 2.3. Design Patterns

The following design patterns have been implemented in the project:

1. Singleton pattern: used in the backend, in the files:

- rest/dao/AdminDao.class.php
- rest/dao/BaseDao.class.php
- rest/dao/MaterialDao.class.php
- rest/dao/BrandDao.class.php
- rest/dao/ColorDao.class.php
- rest/dao/TypeDao.class.php
- rest/services/AdminService.class.php
- rest/services/BaseService.class.php
- rest/services/MaterialService.class.php
- rest/services/BrandService.class.php
- rest/services/ColorService.class.php
- rest/services/TypeService.class.php

The singleton pattern was used to make a single instance of both the database (in the BaseDao) and all of the daos (in the respective Dao files). This way, instead of making a new dao every time it is called, the service instead calls the getInstance() function that will make the new instance of dao only if the instance of that dao is not already set, and return the existing instance if it exists. We decided to use this pattern because of the resource sharing between the classes, time and memory saving since each instance is created only once and instance control since it prevents creation of copies of the same object.

2. Front controller pattern: used in the backend, in the file:

- rest/index.php

Front controller design pattern handles all requests sent to the application. This way, the security of the application is improved, since all requests need to pass through the index.php for authorization in order to be redirected to the desired routes. Thus, index.php has all of the daos and routes and registers all services through Flight.

## 2.4. Tests

The tests can be found inside the github repository in tests/tests.zip. The zipped folder contains the entire java project used for Selenium testing the website.

The following tests were developed for testing the app:

1. Login Test:
  - Logs into the application
  - Checks the current URL
  - Checks the background color of the "Logout" button
  - Clicks the "Logout" button
  - Checks current URL
  - Checks if the title of the email field matches

```
@Test
void test() throws InterruptedException {
    WebDriver.get(baseUrl);
    Thread.sleep(3000);

    WebDriver.findElement(By.xpath("/html/body/main/div/div/form/div[1]/input")).sendKeys("dzanin.masic@stu.ibu.edu.ba");
    Thread.sleep(3000);

    WebDriver.findElement(By.xpath("/html/body/main/div/div/form/div[2]/input")).sendKeys("dzanin");
    Thread.sleep(3000);

    WebDriver.findElement(By.xpath("/html/body/main/div/div/form/button")).click();

    Thread.sleep(3000);
    String url = WebDriver.getCurrentUrl();
    assertEquals("http://209.38.206.111/SoftwareEngineering/index.html", url);

    String buttonColor = WebDriver.findElement(By.xpath("/html/body/main/div/div/div[2]/div/div/button[1]")).getCssValue("font-weight");
    assertEquals("400", buttonColor);
    Thread.sleep(3000);

    WebDriver.findElement(By.xpath("/html/body/main/div/div/div[2]/div/div/button[1]")).click();
    Thread.sleep(3000);

    String modalTitle = WebDriver.findElement(By.xpath("/html/body/div[1]/div/div/div[1]/h5")).getText();
    assertEquals("Modal title", modalTitle);

    WebElement selectElement = WebDriver.findElement(By.id("type_id"));
    Select dropdown = new Select(selectElement);
    dropdown.selectByValue("1");
    Thread.sleep(3000);

    WebDriver.findElement(By.xpath("/html/body/div[1]/div/div/div[3]/button[2]")).click();
    Thread.sleep(3000);

    String typeText = WebDriver.findElement(By.xpath("/html/body/main/div/div/div[2]/div/div/h5")).getText();
    assertEquals("Softshell", typeText);
    Thread.sleep(3000);
}
```

2. Chart Test:
  - Logs into the application
  - Checks the current URL
  - Checks the background color of a button
  - Clicks the "Show total length" button
  - Clicks the "Chart version" button
  - Checks that the "Home" button indeed says "Home"
  - Clicks the "Home" button and goes to the home-page

```

@Test
void test() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/main/div/div/form/div[1]/input")).sendKeys("dzanin.masic@stu.ibu.edu.ba");
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/main/div/div/form/div[2]/input")).sendKeys("dzanin");
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/main/div/div/form/button")).click();

    Thread.sleep(3000);
    String url = webDriver.getCurrentUrl();
    assertEquals("http://209.38.206.111/SoftwareEngineering/index.html", url);

    String buttonColor = webDriver.findElement(By.xpath("/html/body/main/div/div/div[1]/div/button")).getCssValue("background-color");
    String colorToHex = Color.fromString(buttonColor).asHex();
    assertEquals("#6c757d", colorToHex);
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/main/div/div/div[1]/div/button")).click();
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/main/div/div/div/form/button[2]")).click();
    Thread.sleep(3000);

    String chartTitle = webDriver.findElement(By.xpath("/html/body/main/div/div/div[1]/form/button")).getText();
    assertEquals(true, chartTitle.contains("Home"));
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/main/div/div/div[1]/form/button")).click();
    Thread.sleep(3000);
}

```

### 3. Delete Test:

- Logs into the application
- Checks the current URL
- Checks the font weight of the "Delete" button
- Clicks the "Delete" button
- Checks the modal title
- Clicks the "Delete" button inside modal

```

@Test
void test() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/main/div/div/form/div[1]/input")).sendKeys("dzanin.masic@stu.ibu.edu.ba");
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/main/div/div/form/div[2]/input")).sendKeys("dzanin");
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/main/div/div/form/button")).click();

    Thread.sleep(3000);
    String url = webDriver.getCurrentUrl();
    assertEquals("http://209.38.206.111/SoftwareEngineering/index.html", url);

    String buttonFontWeight = webDriver.findElement(By.xpath("/html/body/main/div/div/div[2]/div/div/button[2]")).getCssValue("font-weight");
    assertEquals("400", buttonFontWeight);
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/main/div/div/div[2]/div/div/button[2]")).click();
    Thread.sleep(3000);

    String modalTitle = webDriver.findElement(By.xpath("/html/body/div[2]/div/div/div[1]/h5")).getText();
    assertEquals("Modal title", modalTitle);

    webDriver.findElement(By.xpath("/html/body/div[2]/div/div/div[3]/button[2]")).click();
    Thread.sleep(3000);
}

```

#### 4. Add Test:

- Logs into the application
- Checks the current URL
- Checks the background color of the "Add" button
- Clicks the "Add" button
- Fills out the modal with information
- Clicks the "Add" button inside modal
- Checks if the newly added material brand matches

```
@test
void test() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/main/div/div/form/div[1]/input")).sendKeys("dzanin.masic@stu.ibu.edu.ba");
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/main/div/div/form/div[2]/input")).sendKeys("dzanin");
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/main/div/div/form/button")).click();

    Thread.sleep(3000);
    String url = webDriver.getCurrentUrl();
    assertEquals("http://209.38.206.111/SoftwareEngineering/index.html", url);

    String buttonColor = webDriver.findElement(By.xpath("/html/body/main/div/div/button")).getCssValue("background-color");
    String colorToHex = Color.fromString(buttonColor).asHex();
    assertEquals("#198754", colorToHex);
    Thread.sleep(3000);

    JavascriptExecutor jsExecutor = (JavascriptExecutor) webDriver;
    jsExecutor.executeScript("window.scrollTo(0, document.body.scrollHeight)");
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/main/div/div/button")).click();
    Thread.sleep(3000);

    String modalTitle = webDriver.findElement(By.xpath("/html/body/div[3]/div/div/form/div[1]/h5")).getText();
    assertEquals("Modal title", modalTitle);

    WebElement selectElement = webDriver.findElement(By.id("select-brand"));
    Select dropdown = new Select(selectElement);
    dropdown.selectByValue("6");
    Thread.sleep(3000);

    WebElement selectElement2 = webDriver.findElement(By.id("select-color"));
    Select dropdown2 = new Select(selectElement2);
    dropdown2.selectByValue("7");
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/div[3]/div/div/form/div[2]/div[4]/input")).sendKeys("4000");
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/div[3]/div/div/form/div[2]/div[5]/input")).sendKeys("NO");
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/div[3]/div/div/form/div[3]/button[2]")).click();
    Thread.sleep(3000);

    String brandText = webDriver.findElement(By.xpath("/html/body/main/div/div/div[2]/div/div/p[2]")).getText();
    assertEquals(true, brandText.contains("Klopman"));
    Thread.sleep(3000);
}
```



## 5. Edit Test:

- Logs into the application
- Checks the current URL
- Checks the font weight of the "Edit" button
- Clicks the "Edit" button
- Checks the modal title
- Changes value inside modal
- Checks if the newly edited material type matches

```
@test
void test() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/main/div/div/form/div[1]/input")).sendKeys("dzanin.masic@stu.ibu.edu.ba");
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/main/div/div/form/div[2]/input")).sendKeys("dzanin");
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/main/div/div/form/button")).click();

    Thread.sleep(3000);
    String url = webDriver.getCurrentUrl();
    assertEquals("http://209.38.206.111/SoftwareEngineering/index.html", url);

    String buttonColor = webDriver.findElement(By.xpath("/html/body/main/div/div/form/button")).getCssValue("background-color");
    String colorToHex = Color.fromString(buttonColor).asHex();
    assertEquals("#198754", colorToHex);
    Thread.sleep(3000);

    JavascriptExecutor jsExecutor = (JavascriptExecutor) webDriver;
    jsExecutor.executeScript("window.scrollTo(0, document.body.scrollHeight)");
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/main/div/div/form/button")).click();
    Thread.sleep(3000);

    String modalTitle = webDriver.findElement(By.xpath("/html/body/div[3]/div/div/form/div[1]/h5")).getText();
    assertEquals("Modal title", modalTitle);

    WebElement selectElement = webDriver.findElement(By.id("select-brand"));
    Select dropdown = new Select(selectElement);
    dropdown.selectByValue("6");
    Thread.sleep(3000);

    WebElement selectElement2 = webDriver.findElement(By.id("select-color"));
    Select dropdown2 = new Select(selectElement2);
    dropdown2.selectByValue("7");
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/div[3]/div/div/form/div[2]/div[4]/input")).sendKeys("4000");
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/div[3]/div/div/form/div[2]/div[5]/input")).sendKeys("NO");
    Thread.sleep(3000);

    webDriver.findElement(By.xpath("/html/body/div[3]/div/div/form/div[3]/button[2]")).click();
    Thread.sleep(3000);

    String brandText = webDriver.findElement(By.xpath("/html/body/main/div/div/div[2]/div/div/p[2]")).getText();
    assertEquals(true, brandText.contains("Klopman"));
    Thread.sleep(3000);
}
```

## 6. Search Test:

- Checks the background color of the "Login" button
- Logs into the application
- Checks the current URL
- Writes text inside the search field
- Clicks the search button
- Checks the type
- Goes back to home-page
- Checks the border color of the "Add color" button
- Clicks the "Logout" button
- Checks the current URL

```
@Test
void test() throws InterruptedException {
    WebDriver.get(baseUrl);
    Thread.sleep(3000);

    String buttonColor = WebDriver.findElement(By.xpath("/html/body/main/div/div/form/button")).getCssValue("background-color");
    String colorToHex = Color.fromString(buttonColor).asHex();
    assertEquals("#0d6efd", colorToHex);
    Thread.sleep(1000);

    WebDriver.findElement(By.xpath("/html/body/main/div/div/form/div[1]/input")).sendKeys("dzanin.masic@stu.ibu.edu.ba");
    Thread.sleep(3000);

    WebDriver.findElement(By.xpath("/html/body/main/div/div/form/div[2]/input")).sendKeys("dzanin");
    Thread.sleep(3000);

    WebDriver.findElement(By.xpath("/html/body/main/div/div/form/button")).click();
    Thread.sleep(3000);
    String url = WebDriver.getCurrentUrl();
    assertEquals("http://209.38.206.111/SoftwareEngineering/index.html", url);

    WebDriver.findElement(By.id("search-material")).sendKeys("polar");
    Thread.sleep(3000);
    WebDriver.findElement(By.xpath("/html/body/header/nav/div/div/form[1]/div[2]/button")).click();
    Thread.sleep(3000);

    String materialType = WebDriver.findElement(By.xpath("/html/body/main/div/div/div/div/div/div/h5")).getText();
    assertEquals("Polar", materialType);

    Thread.sleep(3000);
    WebDriver.findElement(By.xpath("/html/body/main/div/div/form/button")).click();

    Thread.sleep(3000);

    String addColorButton = WebDriver.findElement(By.xpath("/html/body/header/nav/div/div/div/div/div/div/li[2]/button")).getCssValue("border-color");
    String addColorToHex = Color.fromString(addColorButton).asHex();
    assertEquals("#198754", addColorToHex);
    Thread.sleep(1000);

    WebDriver.findElement(By.xpath("/html/body/header/nav/div/div/form[2]/button")).click();
    Thread.sleep(1000);

    String url2 = WebDriver.getCurrentUrl();
    assertEquals("http://209.38.206.111/SoftwareEngineering/login.html", url2);
}
```

### **3. Conclusion**

We have managed to develop the application and follow the standards that made our code more understandable and easier to work with. Furthermore, we realized the importance of implementing design patterns in the code, as they have many advantages, including better security and readability and more abstraction. We wish we had more free time to implement and experiment with more design patterns while developing the application.

Throughout this project we learned the importance of writing a clean, understandable code that follows standards. Even though throughout our education we worked only in small teams or individually, the career in development often requires working in big teams, which is much easier if everyone follows certain standards when coding. Furthermore, implementing patterns can benefit the development of application in many ways and it is important to understand the most widely used standards to know to recognize them and how to implement them.