# SOFTWARE VERIFICATION, VALIDATION AND TESTING

## TESTING DOCUMENTATION

TESTING TWITTER

Prepared by:
**Karišik Elmedin**
**Gačanović Sead**

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

10.01.2021.

## Contents

1. Introduction ..... 3
1.1. About the Project ..... 3
1.2. Project Functionalities and Screenshots ..... 3
2. Test Plan ..... 6
2.1. Scope ..... 6
2.2. Testing Environment and Tools ..... 6
3. Test Execution ..... 6
3.1. Login ..... 6
3.2. Post Tweets ..... 7
3.3. Search ..... 8
3.4. Send Messages ..... 9
3.5. Follow and Unfollow ..... 10
3.6. Profile Page ..... 11
3.7. Logout ..... 12
4. Conclusion ..... 13
4.1. Testing Summary ..... 13
4.2. Final Thoughts ..... 13

# 1. Introduction

## 1.1. About the Project

The website that we will be testing for our project will be Twitter. Users post and interact with each other with messages, which are famously known as Tweets. Registered users can post, retweet, quote and like tweets, whereas unregistered users can only read tweets. Twitter can be accessed through its website or mobile app.

The social network has been created by Jack Dorsey, Noah Glass, Biz Stone and Evan Williams in March 2006 and it was launched in July of the same year. It is one of the biggest social networks in the whole world with more than 320 million monthly active users.

The website is created using Java, Ruby, Scala and Javascript and you can access it at https://twitter.com/home.

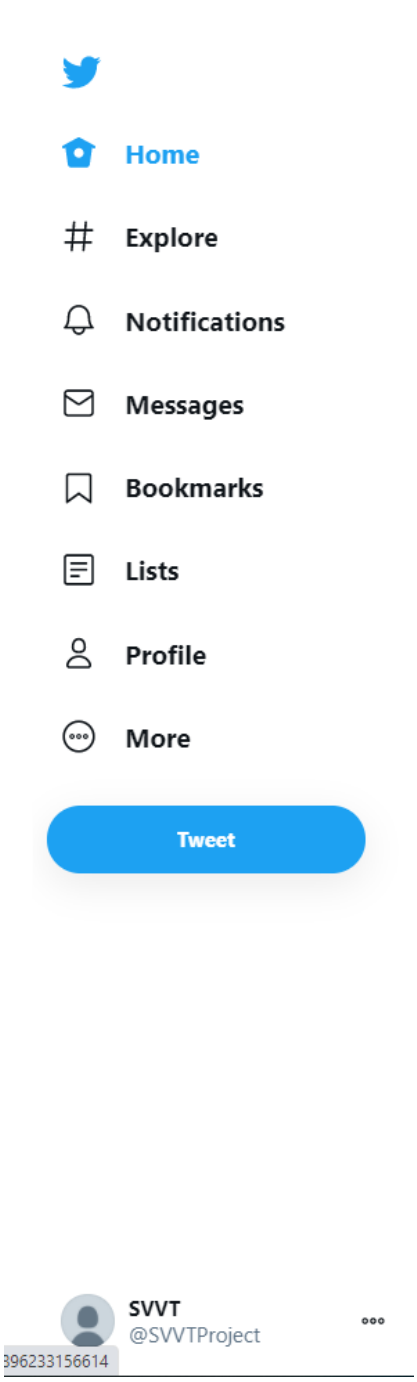
## 1.2. Project Functionalities and Screenshots

We will be looking at and testing Twitter from the perspective of a logged in user.

At the front-end level, the website is made up of multiple webpages, such as:

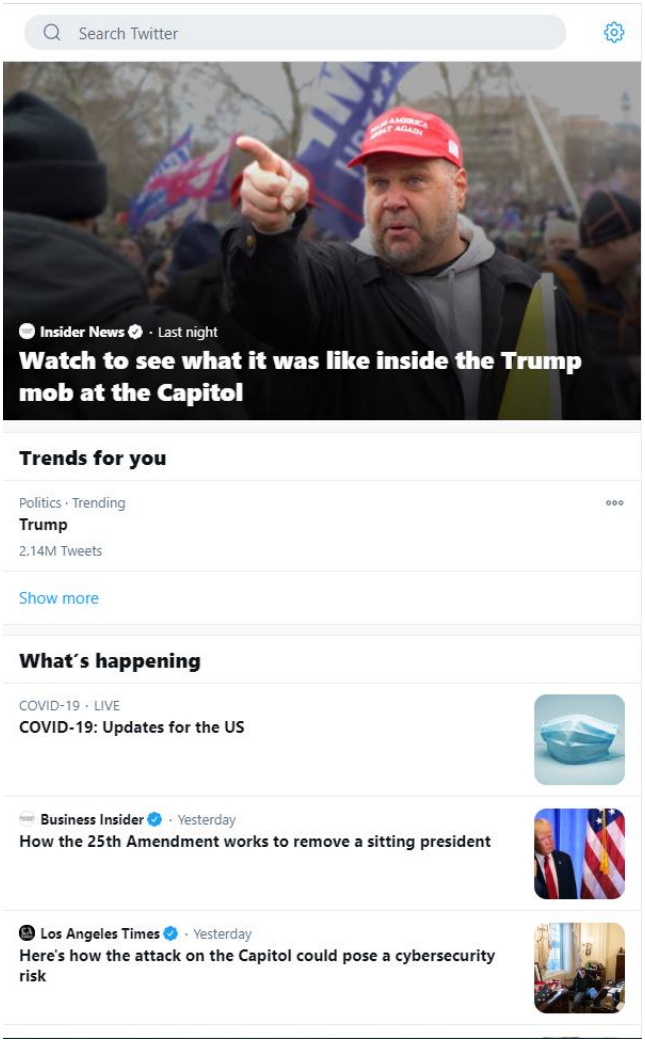
- **Sidebar**: It is present at all times and in all pages. It contains links to other pages: *Home* (which we end up in when we first open the website), *Explore*, *Notifications*, *Messages*, *Bookmarks*, *Lists*, *Profile*, *More*. On top of the sidebar is the Twitter logo, which when clicked takes us also to the homepage. Under the *More* button is the Tweet button which opens up a window from where we can write tweets. And finally, on the bottom of the sidebar is a user icon button, with the users profile photo, from which the user can manage multiple logged in accounts, add existing accounts or log out from the current account.
- **Home page** (https://twitter.com/home): contains **an input field** from which we can write and post tweets**; the Twitter timeline**, where we can see tweets and retweets from the people or topics we follow and accounts similar to theirs and, of course, ads.
- **Explore page** (https://twitter.com/explore): allows the user to search for other users, topics, hashtags and to see what's trending right now.
- **Notifications page** (https://twitter.com/notifications): where the user can see and check any notifications that he might receive.
- **Messages page** (https://twitter.com/messages): here the user can chat privately with any other user.
- **Bookmarks page** (https://twitter.com/i/bookmarks): where the tweets that the user has bookmarked are stored.
- **Lists page** (https://twitter.com/@username /lists): where you can see the lists that a user has created. (Instead @username in the link just type the username of the user who's lists you wish to see)

- **Profile page** (https://twitter.com/@username): where you can check out your profile or any other user's profile. (Just change @username with the username of the user who's profile you want to check out)
- **More button**: opens up a vertical menu from which we can adjust a lot of settings.



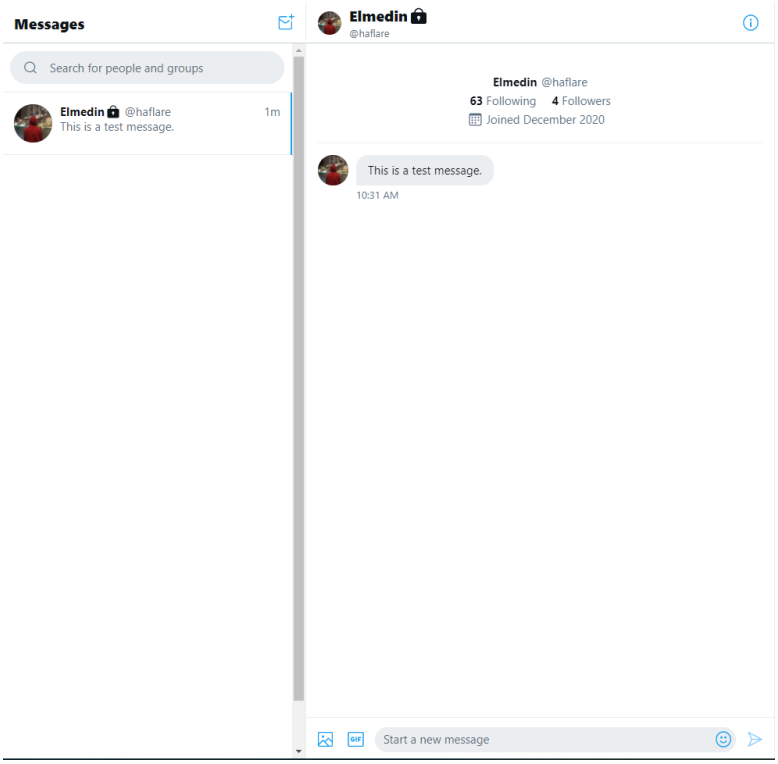

*1. Sidebar*


*2. Home page*

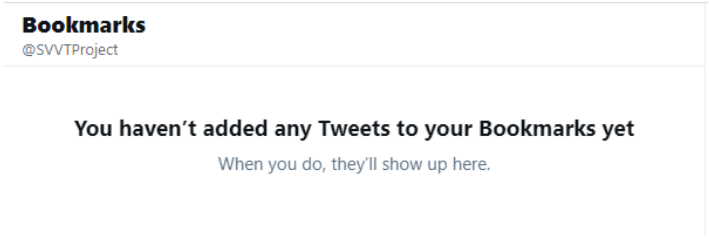



*3. Explore page*

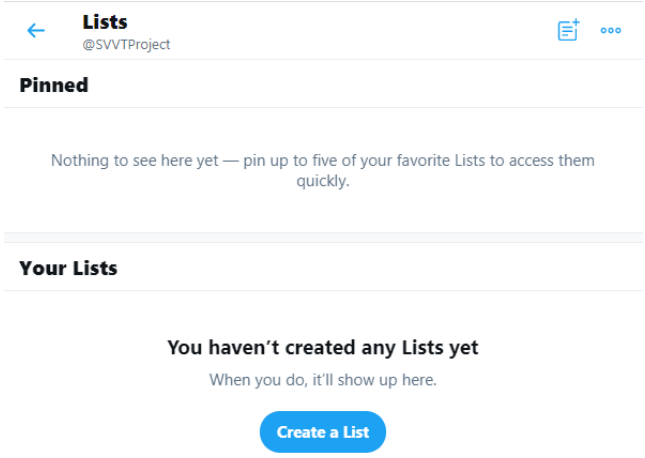

*4. Notifications page*




*5. Messages page*




*6. Bookmarks page*




*7. Lists page*


*8. Profile page*

## 2. Test Plan

## 2.1. Scope

Our testing will try to cover all the major features of Twitter, such as; logging in, logging out, writing/posting tweets, opening different web pages, searching for users, following, unfollowing, etc.

Since we don't have any access to the back-end code, we will only be doing black-box testing.

## 2.2. Testing Environment and Tools

The testing will be done by using JUnit, which is an xUnit framework for Java, which came preinstalled with Eclipse, which is the IDE that we will be using to perform the testing.

We will also be using Selenium, which is also a Java framework, in conjunction with JUnit. The integration of Selenium and Java will be made possible through the Selenium Webdriver library and package for Java. The web driver will use the Google Chrome executable, and the tests will be written and run in Eclipse Java IDE 2019.

## 3. Test Execution

## 3.1. Login

User wants to be able to log in to his profile and use Twitter.

<table>
<tr><td colspan="5">Test Name: Login Test</td></tr>
<tr><td colspan="5">Description: Check if a registered user is able to log in</td></tr>
<tr><td colspan="5">Pre-condition(s): Selenium Web driver is loaded and user already knows the password and username/email/phone number of a registered account.</td></tr>
<tr><td>Test Steps:<br><br>1. Go to https://twitter.com/<br>2. Enter email/phone number/username<br>3. Enter the password<br>4. Press “Log in” button</td><td>Test Data:<br><br>- username: SVVTProject<br>- password: svvttesting</td><td>Expected Result:<br><br>The user is successfully navigated to the home page of his profile</td><td>Actual Result:<br><br>The user is successfully navigated to the home page of his profile</td><td>Status:<br><br>PASS</td></tr>
<tr><td colspan="5">Notes:</td></tr>
</table>

```java
@TestMethodOrder(OrderAnnotation.class)
class TwitterTest {

    private static WebDriver webDriver;
    private static String baseUrl;

    @BeforeAll
    static void setUp() {
        System.setProperty("webdriver.chrome.driver", "/home/elmedin/Documents/code/chrome_driver/chromedriver");
        webDriver = new ChromeDriver();
        baseUrl = "https://twitter.com/";
    }

    @Test
    @Order(1)
    public void testLogin() throws InterruptedException {
        webDriver.get(baseUrl);
        webDriver.manage().window().maximize();
        Thread.sleep(1000);

        WebElement email = webDriver.findElement(By.xpath("/html/body/div/div/div/div/main/div/div/div/div[1]/div[1]/div/form/div/div[1]/di
        WebElement password = webDriver.findElement(By.xpath("/html/body/div/div/div/div/main/div/div/div/div[1]/div[1]/div/form/div/div[2]
        WebElement loginButton = webDriver.findElement(By.xpath("//*[@id=\"react-root\"]/div/div/div/main/div/div/div/div[1]/div[1]/div/for

        email.sendKeys("SVVTProject");
        Thread.sleep(500);

        password.sendKeys("svvttesting");
        Thread.sleep(500);

        loginButton.click();
        Thread.sleep(500);

        WebElement weUsername = webDriver.findElement(By.xpath("/html/body/div/div/div/div[2]/header/div/div/div/div[2]/div/div/div[2]/div/
        String username = weUsername.getText();

        assertEquals("@SVVTProject", username);

        Thread.sleep(3000);
    }
```

## 3.2. Post Tweets

User wants to be able to post Tweets from his profile.

| **Test Name:** Tweeting Test | | | | |
|---|---|---|---|---|
| **Description:** Check if user is able to post Tweets | | | | |
| **Pre-condition(s):** Selenium Web driver is loaded and user is logged in. | | | | |
| **Test Steps:**<br><br>1. Press the Tweet button on the sidebar<br>2. Enter tweet text in the window that should pop up (possible to enter custom text through the Eclipse console)<br>3. Press “Tweet” in the lower right corner of the window | **Test Data:**<br><br>- any input we provide through the console | **Expected Result:**<br><br>The tweet is posted and displayed on top of the timeline | **Actual Result:**<br><br>The tweet is posted and displayed on top of the timeline | **Status:**<br><br>PASS |
| **Notes:** | | | | |

```java
    @Test
    @Order(2)
    public void testTweet() throws InterruptedException {
        @SuppressWarnings("resource")
        Scanner scanner = new Scanner(System.in);

        WebElement tweetButton = webDriver.findElement(By.xpath("/html/body/div/div/div/div[2]/header/div/div/div/div[1]/div[3]/a/div"));
        tweetButton.click();
        Thread.sleep(1000);

        WebElement tweetTextField = webDriver.findElement(By.xpath("/html/body/div/div/div/div[1]/div[2]/div/div/div/div/div/div[2]/div[2]/
        String tweetText = scanner.nextLine();
        tweetTextField.sendKeys(tweetText);

        WebElement sendTweetButton = webDriver.findElement(By.xpath("/html/body/div/div/div/div[1]/div[2]/div/div/div/div/div/div[2]/div[2]
        sendTweetButton.click();

        Thread.sleep(3000);

        WebElement confirmation = webDriver.findElement(By.xpath("/html/body/div/div/div/div[2]/main/div/div/div/div[1]/div/div[4]/div/div/
        String confirmationString = confirmation.getText();

        Thread.sleep(1000);

        assertTrue(confirmationString.contains(tweetText));

        Thread.sleep(3000);
    }
```

## 3.3. Search

User wants to be able to search for other user's profiles.

| **Test Name:** Search Test | | | | |
|---|---|---|---|---|
| **Description:** Check if the searching mechanism is working and user can search for other users | | | | |
| **Pre-condition(s):** Selenium Web driver is loaded and user is logged in. | | | | |
| **Test Steps:**<br><br>1. Click on the Explore button in the sidebar<br>2. Type in a user's name or username in the search-box<br>3. Press enter | **Test Data:**<br><br>- the input we gave in our test was @KingJames, the username of NBA player, LeBron James | **Expected Result:**<br><br>The user is taken to https://twitter.com/KingJames | **Actual Result:**<br><br>The user is taken to https://twitter.com/KingJames | **Status:**<br><br>PASS |
| **Notes:** | | | | |

```java
    @Test
    @Order(3)
    public void testSearch() throws InterruptedException {
        WebElement exploreButton = webDriver.findElement(By.xpath("/html/body/div/div/div/div[2]/header/div/div/div/div[1]/div[2]/nav/a[2]/
        exploreButton.click();
        Thread.sleep(3000);

        WebElement searchBar = webDriver.findElement(By.xpath("/html/body/div/div/div/div[2]/main/div/div/div/div[1]/div/div[1]/div[1]/div/
        searchBar.sendKeys("@KingJames");
        searchBar.sendKeys(Keys.RETURN);
        Thread.sleep(3000);

        WebElement firstResult = webDriver.findElement(By.xpath("/html/body/div/div/div/div[2]/main/div/div/div/div[1]/div/div[2]/div/div/s
        firstResult.click();
        Thread.sleep(3000);

        WebElement userName = webDriver.findElement(By.xpath("/html/body/div/div/div/div[2]/main/div/div/div/div[1]/div/div[2]/div/div/div[
        String userNameText = userName.getText();

        assertEquals("LeBron James", userNameText);

        Thread.sleep(3000);
    }
```

## 3.4. Send Messages

User wants to be able to send direct messages to other users.

| **Test Name:** Message Test | | | | |
|---|---|---|---|---|
| **Description:** Check if user is able to send direct messages to other users | | | | |
| **Pre-condition(s):** Selenium Web driver is loaded and user is logged in. | | | | |
| **Test Steps:**<br><br>1. Click on the Messages button in the sidebar<br>2. Choose a user who will receive the direct message<br>3. Writes the message and press enter to send it | **Test Data:**<br><br>- any input we provide through the console (similar to the Tweet Test example) | **Expected Result:**<br><br>The message is sent | **Actual Result:**<br><br>The message is sent | **Status:**<br><br>PASS |
| **Notes:** | | | | |

```
113    @Test
114    @Order(4)
115    public void testMessage() throws InterruptedException {
116        @SuppressWarnings("resource")
117        Scanner scanner = new Scanner(System.in);
118
119        WebElement messagesButton = webDriver.findElement(By.xpath("/html/body/div/div/div/div[2]/header/div/div/div/div[1]/div[2]/nav/a[4]
120        messagesButton.click();
121        Thread.sleep(3000);
122
123        WebElement userToChatWith = webDriver.findElement(By.xpath("/html/body/div/div/div/div[2]/main/div/div/div/section[1]/div[2]/div/di
124        userToChatWith.click();
125        Thread.sleep(500);
126
127        WebElement messageInput = webDriver.findElement(By.xpath("/html/body/div/div/div/div[2]/main/div/div/div/section[2]/div[2]/div/div/
128        messageInput.click();
129        String message = scanner.nextLine();
130        messageInput.sendKeys(message);
131        messageInput.sendKeys(Keys.RETURN);
132        Thread.sleep(1000);
133
134        String pageSource = webDriver.getPageSource();
135
136        assertTrue(pageSource.contains(message));
137
138        Thread.sleep(3000);
139    }
```

## 3.5. Follow and Unfollow

User wants to be able to follow and unfollow other users at any time.

| **Test Name:** Follow/Unfollow Test | | | | |
|---|---|---|---|---|
| **Description:** Check if the user can successfully follow and unfollow other users | | | | |
| **Pre-condition(s):** Selenium Web driver is loaded and user is logged in. | | | | |
| **Test Steps:**<br><br>1. Visit a user's profile as shown in Search Test example<br>2. Press Follow/Unfollow button | **Test Data:**<br><br>- the input we gave in our test was @EdDzeko, the username of football player, Edin Džeko | **Expected Result:**<br><br>The user from the input was followed and unfollowed | **Actual Result:**<br><br>The user from the input was followed and unfollowed | **Status:**<br><br>PASS |
| **Notes:** | | | | |

```
141     @Test
142     @Order(5)
143     public void testFollowUnfollow() throws InterruptedException {
144         WebElement exploreButton = webDriver.findElement(By.xpath("/html/body/div/div/div/div[2]/header/div/div/div/div[1]/div[2]/nav/a[2]/
145         exploreButton.click();
146         Thread.sleep(3000);
147
148         WebElement searchBar = webDriver.findElement(By.xpath("/html/body/div/div/div/div[2]/main/div/div/div/div[1]/div/div[1]/div[1]/div/
149         searchBar.sendKeys("@EdDzeko");
150         searchBar.sendKeys(Keys.RETURN);
151         Thread.sleep(3000);
152
153         WebElement firstResult = webDriver.findElement(By.xpath("/html/body/div/div/div/div[2]/main/div/div/div/div[1]/div/div[2]/div/div/s
154         firstResult.click();
155         Thread.sleep(3000);
156
157         WebElement followButton = webDriver.findElement(By.xpath("/html/body/div/div/div/div[2]/main/div/div/div/div[1]/div/div[2]/div/div/
158
159         assertNotEquals("Following", followButton.getText());
160
161         followButton.click();
162         Thread.sleep(1000);
163
164         assertEquals("Following", followButton.getText());
165
166         followButton.click();
167         Thread.sleep(1000);
168
169         WebElement confirmUnfollowButton = webDriver.findElement(By.xpath("/html/body/div/div/div/div[1]/div[2]/div/div/div/div/div/div[2]/
170         confirmUnfollowButton.click();
171         Thread.sleep(1000);
172
173         assertNotEquals("Following", followButton.getText());
174
175         Thread.sleep(3000);
176     }
```

## 3.6. Profile Page

User wants to be able to access his profile page easily and at any time.

| **Test Name:** Profile Page Test | | | | |
|---|---|---|---|---|
| **Description:** Check if the user can access his own profile page | | | | |
| **Pre-condition(s):** Selenium Web driver is loaded and user is logged in. | | | | |
| **Test Steps:**<br><br>1. Click on the Profile button in the sidebar | **Test Data:** | **Expected Result:**<br><br>The user is taken to his profile page | **Actual Result:**<br><br>The user is taken to his profile page | **Status:**<br><br>PASS |
| **Notes:** | | | | |

```
178     @Test
179     @Order(6)
180     public void testProfilePage() throws InterruptedException {
181         WebElement profileButton = webDriver.findElement(By.xpath("/html/body/div/div/div/div[2]/header/div/div/div/div[1]/div[2]/nav/a[7]/
182         profileButton.click();
183         Thread.sleep(3000);
184 
185         WebElement name = webDriver.findElement(By.xpath("/html/body/div/div/div/div[2]/main/div/div/div/div[1]/div/div[2]/div/div/div[1]/d
186         String nameString = name.getText();
187 
188         WebElement username = webDriver.findElement(By.xpath("/html/body/div/div/div/div[2]/main/div/div/div/div[1]/div/div[2]/div/div/div[
189         String usernameString = username.getText();
190 
191         String url = webDriver.getCurrentUrl();
192 
193         assertEquals("SVVT", nameString);
194         assertEquals("@SVVTProject", usernameString);
195         assertEquals("https://twitter.com/SVVTProject", url);
196 
197         Thread.sleep(3000);
198     }
199 
```

## 3.7. Logout

User wants to be able to log out of his Twitter profile at any time.

| **Test Name:** Logout Test | | | | |
|---|---|---|---|---|
| **Description:** Check if user can easily log out of his account | | | | |
| **Pre-condition(s):** Check if the user can access his own profile page | | | | |
| **Test Steps:**<br><br>1. Click on the last section in the sidebar<br>2. Click “Log out @SVVTProject”<br>3. Confirm logout by clicking on “Log out” in the window that pops up | **Test Data:** | **Expected Result:**<br><br>The user is logged out and taken to the login/registration page | **Actual Result:**<br><br>The user is logged out and taken to the login/registration page | **Status:**<br><br>PASS |
| **Notes:** | | | | |

```
    @Test
    @Order(7)
    public void testLogout() throws InterruptedException {
        webDriver.get(baseUrl);
        webDriver.manage().window().maximize();
        Thread.sleep(1000);

        WebElement dots = webDriver.findElement(By.xpath("/html/body/div/div/div/div[2]/header/div/div/div/div[2]/div/div/div[3]"));
        dots.click();
        Thread.sleep(1000);

        WebElement logoutButton = webDriver.findElement(By.xpath("/html/body/div/div/div/div[1]/div[2]/div/div/div[2]/div/div[2]/div/div/di
        logoutButton.click();
        Thread.sleep(1000);

        WebElement logoutConfirmButton = webDriver.findElement(By.xpath("/html/body/div/div/div/div[1]/div[2]/div/div/div/div/div/div[2]/di
        logoutConfirmButton.click();
        Thread.sleep(1000);

        String logoutUrl = webDriver.getCurrentUrl();

        assertTrue(logoutUrl.contains("logout"));

        Thread.sleep(3000);
    }

    @AfterAll
    static void tearDown() {
        webDriver.close();
    }

}
```

# 4. Conclusion

## 4.1. Testing Summary

| Testing Tool | Total Tests | Passed Tests | Failed Tests |
|---|---|---|---|
| Selenium and JUnit | 7 | 7 | 0 |

## 4.2. Final Thoughts

The project, or should we say website, that we tested was very well designed and implemented. I think that we did a good job with our tests. We wrote only 7 tests, but they were rather complex and contain multiple tests in themselves.
We did make one observation that is a flaw in design in our eyes, and that is that there are two buttons (the Button with the Twitter icon and the Home Page button in the sidebar) which lead to the same page, the Home Page.