**TRƯỜNG ĐẠI HỌC VĂN LANG**

**KHOACÔNG NGHỆ THÔNG TIN**

A red text on a black background

Description automatically generated

**KIỂM THỬ TỰ ĐỘNG**

***QUẢN LÝ ANIME***

***GVHD: PHAN GIA PHƯỚC***

**SVTH**: Nguyễn Hoài Khương – 2174802010843

Võ Đăng Kỷ - 2174802010349

Trần Bảo Anh Huy – 2174802010829

Phạm Thành Tâm - 2174802010372

Mục Lục

[**1. Chức năng Đăng nhập/Đăng ký (Thành Tâm) 4**](#_Toc14064)

[1.1 Kiểm tra đăng nhập 4](#_Toc16907)

[1.2 Đăng ký 4](#_Toc15797)

[1.3 Code chức năng Đăng nhập/ Đăng ký 6](#_Toc30841)

[1.4 Kết quả chức năng Đăng nhập/ Đăng ký 9](#_Toc28883)

[**2. Chức năng Chia Sẻ (Hoài Khương) 9**](#_Toc14341)

[2.1 Kiểm tra chức năng chia sẻ 10](#_Toc4890)

[2.2 Code chức năng chia sẻ 11](#_Toc27529)

[2.3 Kết quả chức năng chia sẻ 12](#_Toc29218)

[**3. Chức năng Báo cáo (Hoài Khương) 12**](#_Toc2011)

[3.1 Kiểm tra chức năng báo cáo 13](#_Toc32408)

[3.2 Code chức năng báo cáo 13](#_Toc5461)

[3.3 Kết quả chức năng báo cáo 15](#_Toc14382)

[**4. Chức năng Hỗ trợ (Hoài Khương) 15**](#_Toc21154)

[4.1 Kiểm tra chức năng hỗ trợ 15](#_Toc4537)

[4.2 Code chức năng hỗ trợ 16](#_Toc8907)

[4.3 Kết quả chức năng báo cáo 18](#_Toc22398)

[**5. Chức năng Nhắn tin với người dùng (Hoài Khương) 18**](#_Toc18654)

[5.1 Kiểm tra chức năng nhắn tin với người dùng 18](#_Toc12485)

[5.2 Code chức năng nhắn tin với người dùng 19](#_Toc17832)

[5.3 Kết quả chức năng nhắn tin với người dùng 21](#_Toc24361)

[**6. Chức năng Tìm kiếm Anime (Đăng Kỷ) 21**](#_Toc4179)

[6.1 Kiểm tra chức năng tìm kiếm Anime 21](#_Toc12608)

[6.2 Code chức năng tìm kiếm Anime 22](#_Toc26860)

[6.3 Kết quả chức năng tìm kiếm Anime 23](#_Toc2948)

[**7. Chức năng Yêu thích Anime (Đăng Kỷ) 23**](#_Toc15384)

[7.1 Kiểm tra chức năng yêu thích Anime 23](#_Toc16819)

[7.2 Kiểm tra chức năng yêu thích Anime 24](#_Toc2442)

[7.3 Kết quả chức năng yêu thích Anime 25](#_Toc3069)

[**8. Chức năng Cập nhật ảnh đại diện (Đăng Kỷ) 26**](#_Toc2797)

[8.1 Kiểm tra chức năng Cập nhật ảnh đại diện 26](#_Toc32432)

[8.2 Code chức năng Cập nhật ảnh đại diện 26](#_Toc1891)

[8.1 Kết quả chức năng Cập nhật ảnh đại diện 28](#_Toc24647)

[**9. Chức năng So sánh (Đăng Kỷ) 28**](#_Toc24526)

[9.1 Kiểm tra chức năng so sánh 28](#_Toc22800)

[9.2 Code chức năng so sánh 28](#_Toc629)

[9.3 Kết quả chức năng so sánh 29](#_Toc4280)

[**10. Chức năng review (Thành Tâm) 29**](#_Toc10441)

[10.1 Kiểm tra chức năng review 29](#_Toc12671)

[10.2 Code chức năng review 30](#_Toc6224)

[10.3 Kết quả chức năng review 30](#_Toc17903)

[**11. Tính năng Kiểm tra nội dung ( Võ Đăng Kỷ) 31**](#_Toc10252)

[11.1 Kiểm tra định danh và text trên các tên trường 31](#_Toc19771)

[11.2 Code kiểm tra nội dung 31](#_Toc5578)

[11.3 Kết quả kiểm tra nội dung 32](#_Toc953)

[**12. Tính năng Tile (Anh Huy) 32**](#_Toc21314)

[12.2 Kiểm tra tính chính xác của tiêu đề trang. 32](#_Toc2072)

[12.2 Code tính năng tile 32](#_Toc28916)

[12.3Kết quả tính năng tile 33](#_Toc2870)

[**13 Tính năng Kiểm tra từ khóa ( Trần Bảo Anh Huy) 33**](#_Toc15648)

[13.1 Kiểm tra title và text trên các tên trường 33](#_Toc26568)

[13.2 Code Tính năng Kiểm tra từ khóa 34](#_Toc8013)

[13.3 Kết quả Tính năng Kiểm tra từ khóa 35](#_Toc10295)

[**14 Tính năng đánh giá ( Trần Bảo Anh Huy) 33**](#_Toc15648)

[14.1 Kiểm tra title và text trên các tên trường 33](#_Toc26568)

[14.2 Code Tính năng Kiểm tra đánh giá 34](#_Toc8013)

[14.3 Kết quả Tính năng Kiểm tra đánh giá 35](#_Toc10295)

[**15. Tính năng đăng nhập (POM) ( Võ Đăng Kỷ) 33**](#_Toc15648)

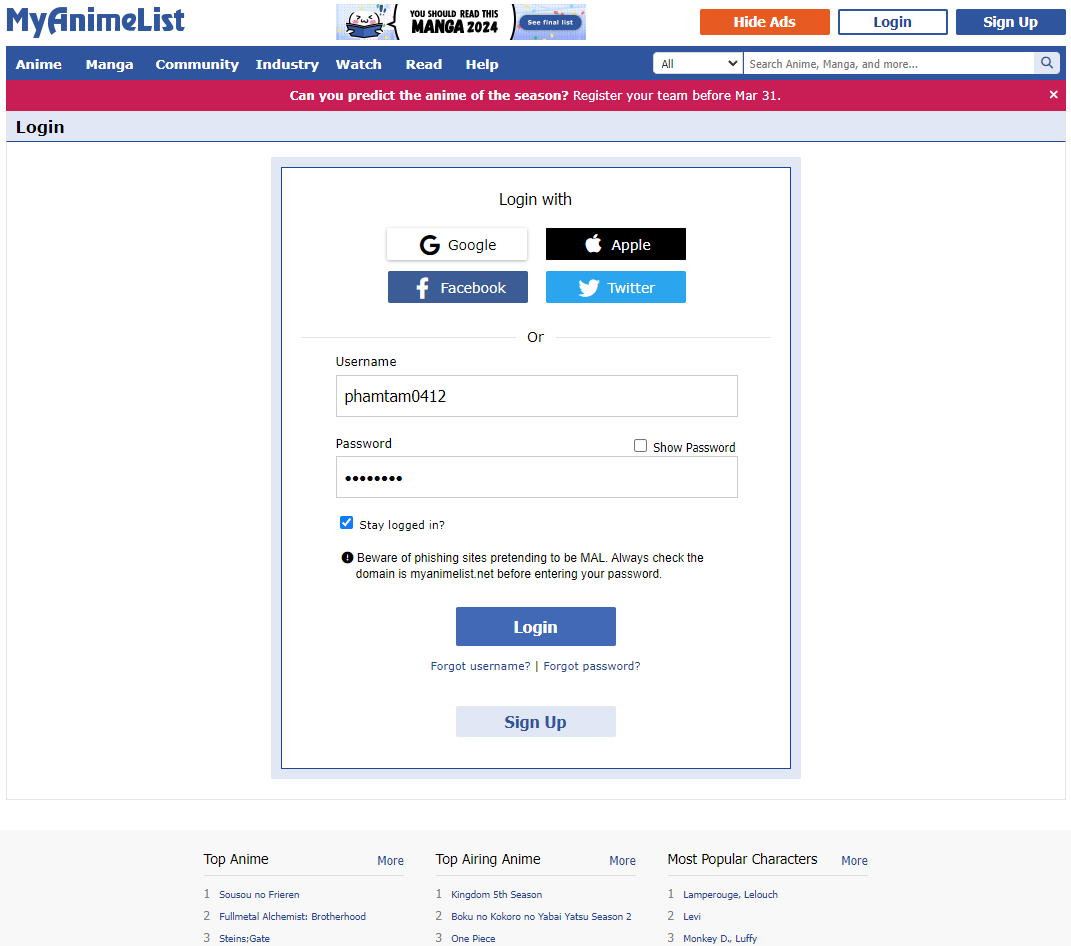
[[15.1 Kiểm tra chức](#_Toc26568) năng đăng nhập (POM) 33](#_Toc26568)

[15.2 Code Tính năng Kiểm tra đăng nhập 34](#_Toc8013)

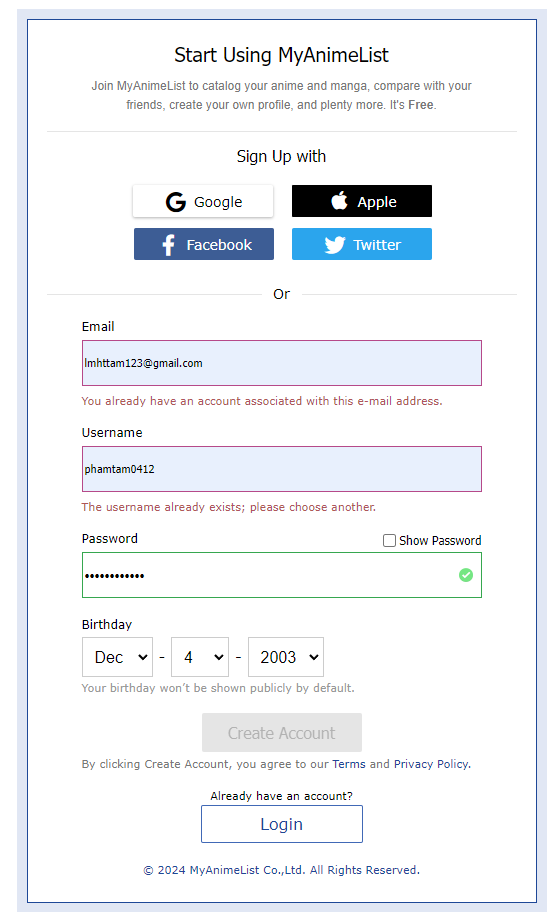
[15.3 Kết quả Tính năng Kiểm tra đăng nhập 35](#_Toc10295)

1. **Chức năng Đăng nhập/Đăng ký (Thành Tâm)**

1.1 Kiểm tra đăng nhập



1.2 Đăng ký



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID\_TC | Title | Các bước thực hiện | Kết quả mong đợi | Kết quả thực tế | Mục tiêu | RS |
| TC\_001 | Đăng nhập bằng tên người dùng và mật khẩu | 1. Truy cập vào trang đăng nhập của ứng dụng. 2. Nhập tên người dùng và mật khẩu. 3. Nhấn nút đăng nhập. | Người dùng được chuyển hướng đến trang chính của ứng dụng sau khi đăng nhập thành công. | Trang chính của ứng dụng hiển thị sau khi đăng nhập thành công. | Kiểm tra tính năng đăng nhập bằng tên người dùng và mật khẩu. | Pass |
| TC\_002 | Đăng nhập bằng tài khoản Gmail | 1. Truy cập vào trang đăng nhập của ứng dụng. 2. Chọn phương thức đăng nhập bằng Gmail.  3. Nhập thông tin tài khoản Gmail.  4. Nhấn nút đăng nhập. | Người dùng được chuyển hướng đến trang chính của ứng dụng sau khi đăng nhập thành công. | Trang chính của ứng dụng hiển thị sau khi đăng nhập thành công. | Kiểm tra tính năng đăng nhập bằng tài khoản Gmail. | Pass |
| TC\_003 | Đăng nhập bằng tài khoản Facebook | 1. Truy cập vào trang đăng nhập của ứng dụng. 2. Chọn phương thức đăng nhập bằng Facebook. 3. Nhập thông tin tài khoản Facebook. 4. Nhấn nút đăng nhập. | Người dùng được chuyển hướng đến trang chính của ứng dụng sau khi đăng nhập thành công. | Trang chính của ứng dụng hiển thị sau khi đăng nhập thành công. | Kiểm tra tính năng đăng nhập bằng tài khoản Facebook. | Fail |
| TC\_004 | Đăng nhập bằng tài khoản Twitter | 1. Truy cập vào trang đăng nhập của ứng dụng. 2. Chọn phương thức đăng nhập bằng Twitter.  3. Nhập thông tin tài khoản Twitter.  4. Nhấn nút đăng nhập. | Người dùng được chuyển hướng đến trang chính của ứng dụng sau khi đăng nhập thành công. | Trang chính của ứng dụng hiển thị sau khi đăng nhập thành công. | Kiểm tra tính năng đăng nhập bằng tài khoản Twitter. | Fail |
| TC\_005 | Đăng nhập bằng tài khoản Apple | 1. Truy cập vào trang đăng nhập của ứng dụng. 2. Chọn phương thức đăng nhập bằng Apple.  3. Nhập thông tin tài khoản Apple.  4. Nhấn nút đăng nhập. | Người dùng được chuyển hướng đến trang chính của ứng dụng sau khi đăng nhập thành công. | Trang chính của ứng dụng hiển thị sau khi đăng nhập thành công. | Kiểm tra tính năng đăng nhập bằng tài khoản Apple. | Pass |

1.3 Code chức năng Đăng nhập/ Đăng ký

import java.io.FileReader;

import java.io.IOException;

import org.json.simple.JSONArray;

import org.json.simple.JSONObject;

import org.json.simple.parser.JSONParser;

import org.json.simple.parser.ParseException;

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.Test;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.DataProvider;

public class Login\_Register {

WebDriver driver;

String[][] data = null;

@Test(dataProvider = "dpLogin")

public void testLogin(String email, String password, String loginMethod) throws InterruptedException {

driver.get("https://myanimelist.net/login.php");

switch (loginMethod) {

case "Google":

loginWithGoogle(email, password);

break;

case "Facebook":

loginWithFacebook(email, password);

break;

case "Twitter":

loginWithTwitter(email, password);

break;

default:

System.out.println("Phương thức đăng nhập không được hỗ trợ: " + loginMethod);

}

}

public void loginWithGoogle(String email, String password) throws InterruptedException {

driver.findElement(By.id("google-button")).click();

Thread.sleep(2000);

driver.findElement(By.id("identifierId")).sendKeys(email + Keys.ENTER);

Thread.sleep(2000);

driver.findElement(By.name("password")).sendKeys(password + Keys.ENTER);

}

public void loginWithFacebook(String email, String password) throws InterruptedException {

driver.findElement(By.id("facebook-login-button")).click();

Thread.sleep(2000);

driver.findElement(By.id("email")).sendKeys(email);

driver.findElement(By.id("pass")).sendKeys(password);

driver.findElement(By.id("loginbutton")).click();

}

public void loginWithTwitter(String email, String password) throws InterruptedException {

driver.findElement(By.id("twitter-login-button")).click();

Thread.sleep(2000);

driver.findElement(By.id("username")).sendKeys(email);

driver.findElement(By.id("password")).sendKeys(password);

driver.findElement(By.id("loginbutton")).click();

}

@BeforeMethod

public void beforeMethod() {

System.out.println("Bắt đầu kiểm tra");

System.setProperty("webdriver.chrome.driver", "C:\\chromedriver.exe");

driver = new ChromeDriver();

driver.manage().window().maximize();

}

@AfterMethod

public void afterMethod() {

driver.quit();

}

@DataProvider(name = "dpLogin")

public String[][] readLoginData() throws IOException, ParseException {

return readData("D:\\testdata.json", "userlogins");

}

@DataProvider(name = "dpRegister")

public String[][] readRegisterData() throws IOException, ParseException {

return readData("D:\\testdata.json", "userregistrations");

}

public String[][] readData(String filePath, String dataKey) throws IOException, ParseException {

String email = null;

String username = null;

String pass = null;

String birthday = null;

String loginMethod = null;

FileReader reader = new FileReader(filePath);

JSONParser jsonParser = new JSONParser();

Object jsonObject = jsonParser.parse(reader);

JSONObject jsonObj = (JSONObject) jsonObject;

JSONArray array = (JSONArray) jsonObj.get(dataKey);

String arr[][] = new String[array.size()][5];

for (int i = 0; i < array.size(); i++) {

JSONObject userData = (JSONObject) array.get(i);

email = (String) userData.get("email");

username = (String) userData.get("username");

pass = (String) userData.get("password");

birthday = (String) userData.get("birthday");

loginMethod = (String) userData.get("loginMethod");

arr[i][0] = email;

arr[i][1] = username;

arr[i][2] = pass;

arr[i][3] = birthday;

arr[i][4] = loginMethod;

}

return arr;

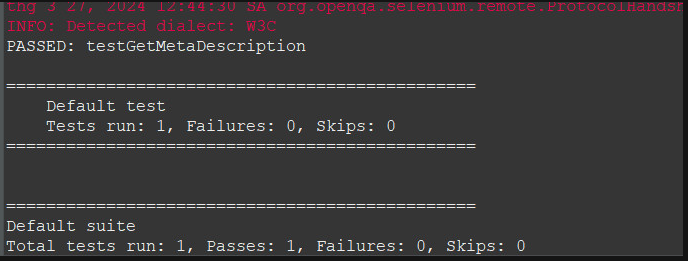
}

}

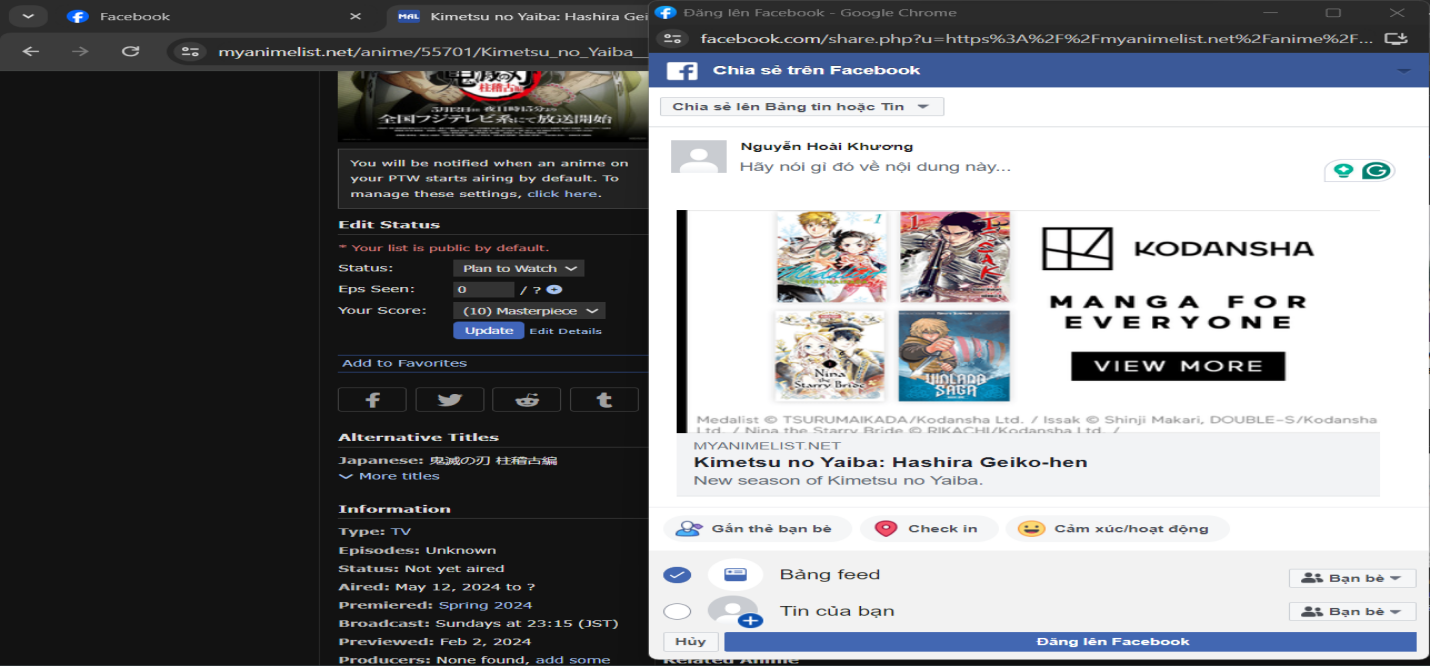
- Giải thích:

Ta sử dụng TestNG và chức năng Test kèm theo độ ưu tiên của priority nhập thông tin đăng hoặc cách đăng nhập khác nhau để vào trang chủ.

1.4 Kết quả chức năng Đăng nhập/ Đăng ký



1. **Chức năng Chia Sẻ (Hoài Khương)**

2.1 Kiểm tra chức năng chia sẻ 

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID\_TC | Title | Các bước thực hiện | Kết quả mong đợi | Kết quả thực tế | Mục tiêu | RS |
| TC\_01 | Kiểm tra nội dung đăng nhập | 1. Mở form  2. Điền thông tin tài khoản và mật khẩu  3. Xem thông báo | Đăng nhập thành công | Đăng nhập thành công | Đăng nhập vào tài khoản để có thể chọn bộ anime yêu thích để chia sẻ | Pass |
| TC\_2 | Tìm kiếm bộ anime muốn chia sẻ | 1. Vào mục tìm kiếm nhập tên bộ anime  2. Nhập tên anime  3. Tìm kiếm | Tìm kiếm thành công | Tìm kiếm thất bại | Để tìm được bộ anime để chia sẻ cho bạn bè theo ý muốn | Fail |
| TC\_3 | Chia sẽ anime | 1. Chọn mục anime muốn chia sẻ  2. Bấm nút chia sẽ  3. Xác nhận | Chia sẽ thành công | Chia sẽ thất bại | Chia sẽ được anime cho bạn bè | Fail |

2.2 Code chức năng chia sẻ

package QLAnime;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class Anime\_Share {

private WebDriver driver;

*@BeforeClass*

public void setUp() {

System.*setProperty*("webdriver.chrome.driver", "C:\\chromedriver.exe");

driver = new ChromeDriver();

driver.get("https://myanimelist.net/");

}

*@Test*(priority = 1)

public void testLogin() {

WebElement loginButton = driver.findElement(By.*xpath*("//a[contains(text(),'Login')]"));

loginButton.click();

WebElement usernameField = driver.findElement(By.*id*("loginUserName"));

WebElement passwordField = driver.findElement(By.*id*("login-password"));

WebElement loginSubmitButton = driver.findElement(By.*xpath*("//input[@value='Login']"));

usernameField.sendKeys("langxitrum");

passwordField.sendKeys("030102Ky");

loginSubmitButton.click();

try {

Thread.*sleep*(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

*@Test*(priority = 2, dependsOnMethods = {"testLogin"})

public void testSearch() throws InterruptedException {

WebElement loginseach = driver.findElement(By.*name*("topkeyword"));

loginseach.click();

WebElement searchtext = driver.findElement(By.*id*("topSearchText"));

searchtext.sendKeys("Kimetsu no Yaiba: Hashira Geiko-hen");

WebElement searchButton = driver.findElement(By.*xpath*("//\*[@id=\"topSearchButon\"]/i"));

searchButton.click();

try {

Thread.*sleep*(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

WebElement result = driver.findElement(By.*className*("js-categories-seasonal"));

Assert.*assertTrue*(result.getText().toLowerCase().contains("Kimetsu no Yaiba: Hashira Geiko-hen"), "Search failed!");

}

*@Test*(priority = 3)

public void click() throws InterruptedException {

WebElement select = driver.findElement(By.*xpath*("//\*[@id=\"#revAreaAnimeHover55701\"]"));

select.click();

Thread.*sleep*(2000);

WebElement share = driver.findElement(By.*xpath*("//\*[@id=\"content\"]/table/tbody/tr/td[1]/div/div[7]/a[1]/div/i"));

share.click();

Thread.*sleep*(2000);

}

*@AfterClass*

public void tearDown() {

try {

Thread.*sleep*(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

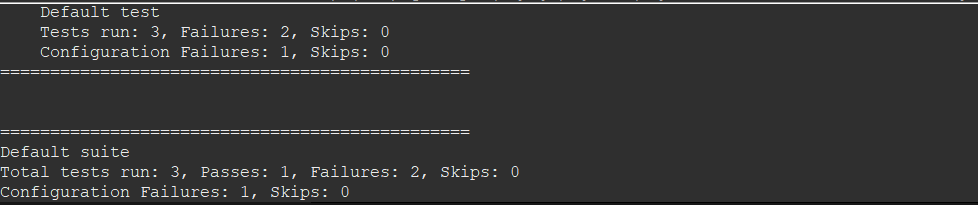
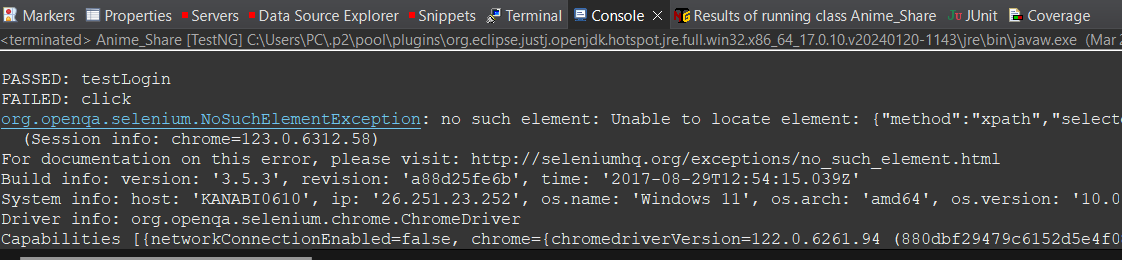
driver.quit();

}

}

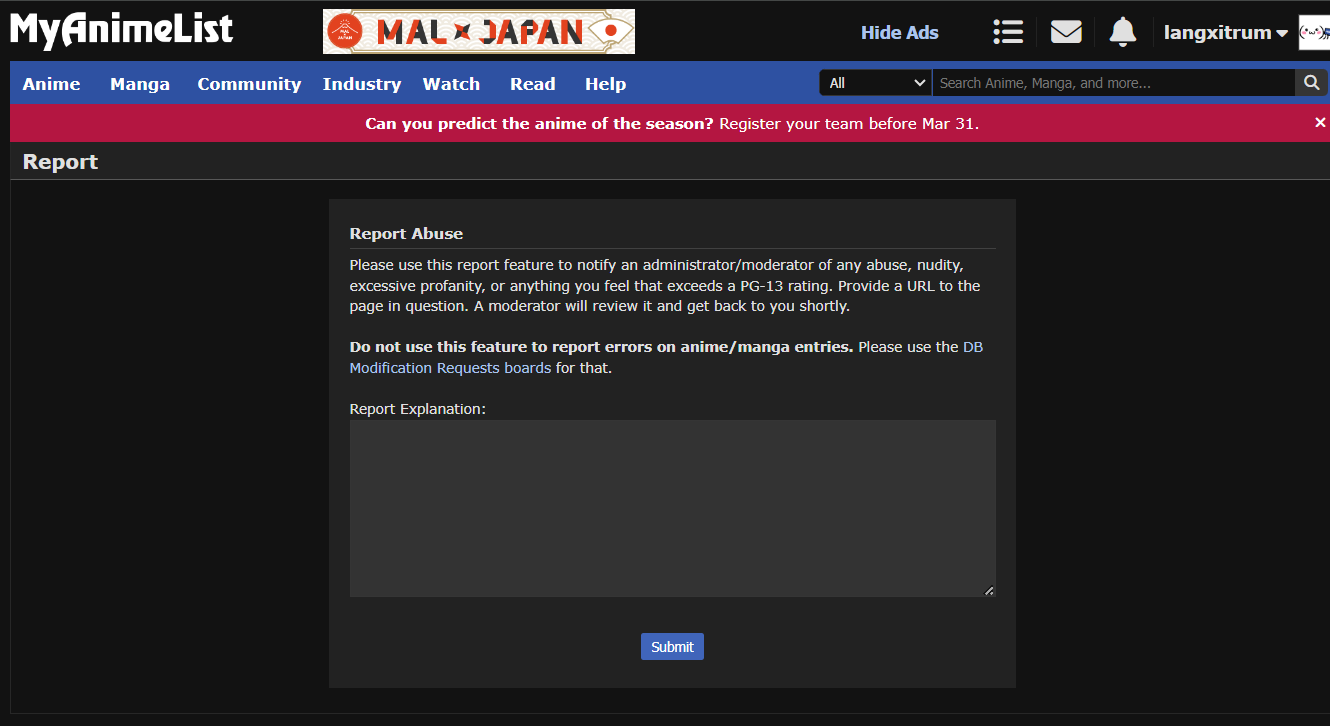
- Giải thích: Ta sử dụng TestNG và chức năng Test kèm theo độ ưu tiên của priority sau khi đăng nhập thành công sẽ đến bước tìm kiếm bộ anime muốn chia sẻ sau đó chọn qua mục để chia sẻ và xác nhận.

2.3 Kết quả chức năng chia sẻ



**3. Chức năng Báo cáo (Hoài Khương)**

3.1 Kiểm tra chức năng báo cáo



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID\_TC | Title | Các bước thực hiện | Kết quả mong đợi | Kết quả thực tế | Mục tiêu | RS |
| TC\_01 | Kiểm tra nội dung đăng nhập | 1. Mở form  2. Điền thông tin tài khoản và mật khẩu  3. Xem thông báo | Đăng nhập thành công | Đăng nhập thành công | Đăng nhập vào tài khoản để có thể chọn bộ anime yêu thích để chia sẻ | Pass |
| TC\_2 | Xác nhận chọn phần hỗ trợ để báo cáo | 1. Vào mục Help  2. Chọn Report | Chọn báo cáo thành công | Chọn báo cáo thành công | Để có thể mở form điền dữ liệu báo cáo | Pass |
| TC\_3 | Điền form và xác nhận báo cáo | 1. Điền thông tin cần báo cáo  2. Bấm nút xác nhận | Gửi thành công | Gửi thành công | Báo cáo đến trang chủ để biết khắc phục | Pass |

3.2 Code chức năng báo cáo

package QLAnime;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class AnimeReport {

private WebDriver driver;

*@BeforeClass*

public void setUp() {

System.*setProperty*("webdriver.chrome.driver", "C:\\chromedriver.exe");

driver = new ChromeDriver();

driver.get("https://myanimelist.net/");

driver.manage().window().maximize();

}

*@Test*(priority = 1)

public void TestLoginAnime() {

WebElement loginButton = driver.findElement(By.*xpath*("//a[contains(text(),'Login')]"));

loginButton.click();

WebElement usernameField = driver.findElement(By.*id*("loginUserName"));

WebElement passwordField = driver.findElement(By.*id*("login-password"));

WebElement loginSubmitButton = driver.findElement(By.*xpath*("//input[@value='Login']"));

usernameField.sendKeys("langxitrum"); //

passwordField.sendKeys("030102Ky"); //

loginSubmitButton.click();

try {

Thread.*sleep*(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

*@Test*(priority = 2)

public void testReport() throws InterruptedException {

WebElement searchButton = driver.findElement(By.*xpath*("//\*[@id=\"nav\"]/li[7]/a"));

searchButton.click();

WebElement searchButton1 = driver.findElement(By.*xpath*("//\*[@id=\"nav\"]/li[7]/ul/li[5]/a"));

searchButton1.click();

try {

Thread.*sleep*(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

*@Test*(priority = 3)

public void testSubmit() throws InterruptedException {

WebElement abuse\_report = driver.findElement(By.*id*("abuse\_report"));

abuse\_report.sendKeys("Thanks"); //

WebElement SubmitButton = driver.findElement(By.*xpath*("//\*[@id=\"dialog\"]/tbody/tr/td/form/div[2]/input"));

SubmitButton.click();

}

*@AfterClass*

public void tearDown() {

try {

Thread.*sleep*(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

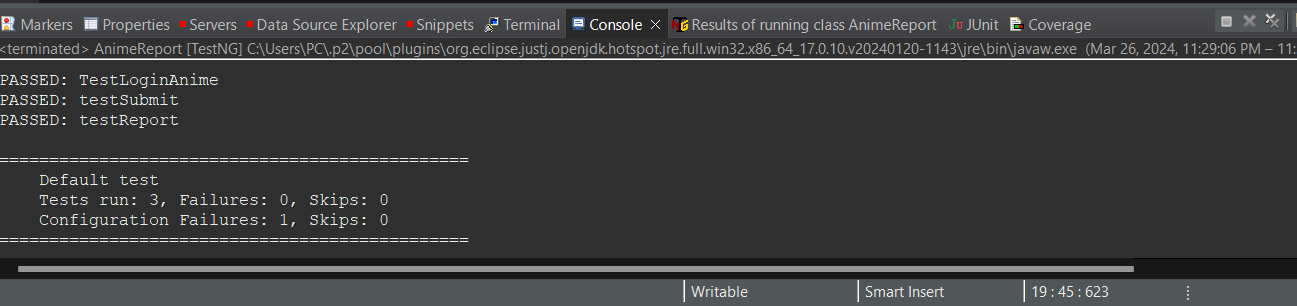
driver.quit();

}

}

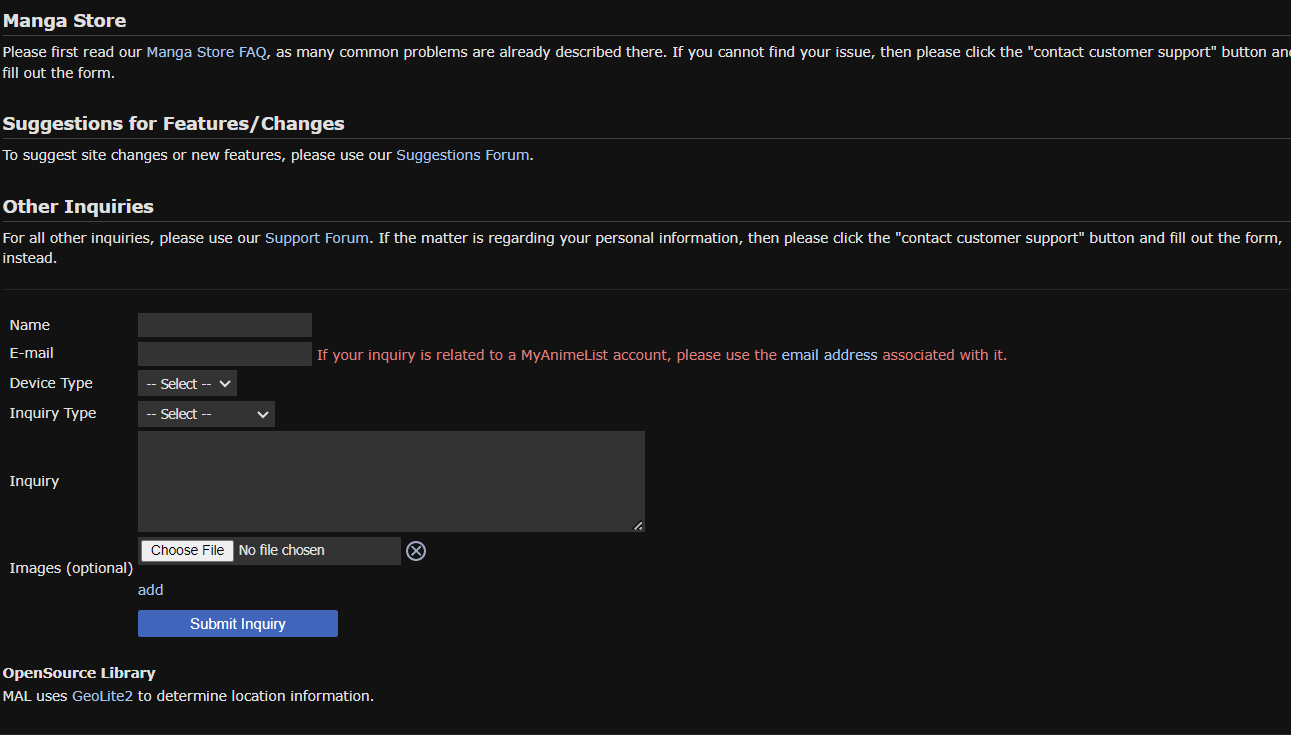
- Giải thích: Ta sử dụng TestNG và chức năng Test kèm theo độ ưu tiên của priority sau khi đăng nhập thành công sẽ đến bước tìm phần hỗ trợ và báo cáo sau khi mở form và điền nội dung báo cáo và xác nhận gửi.

3.3 Kết quả chức năng báo cáo



**4. Chức năng Hỗ trợ (Hoài Khương)**

4.1 Kiểm tra chức năng hỗ trợ



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID\_TC | Title | Các bước thực hiện | Kết quả mong đợi | Kết quả thực tế | Mục tiêu | RS |
| TC\_01 | Kiểm tra nội dung đăng nhập | 1. Mở form  2. Điền thông tin tài khoản và mật khẩu  3. Xem thông báo | Đăng nhập thành công | Đăng nhập thành công | Đăng nhập vào tài khoản để có thể chọn bộ anime yêu thích để chia sẻ | Pass |
| TC\_2 | Xác nhận chọn phần hỗ trợ để vào form | 1. Vào mục Help  2. Chọn Suportive | Vào form hỗ trợ | Vào form hỗ trợ | Để có thể mở form điền dữ liệu hỗ trợ | Pass |
| TC\_3 | Điền form và xác nhận hỗ trợ | 1. Điền thông tin cần báo cáo hỗ trợ  2. Bấm nút xác nhận | Gửi thành công | Gửi thất bại | Báo cáo thông tin hỗ trợ đến trang chủ để biết khắc phục cho mình | Fail |

4.2 Code chức năng hỗ trợ

package QLAnime;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.support.ui.Select;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class AnimeSuport {

private WebDriver driver;

*@BeforeClass*

public void setUp() {

System.*setProperty*("webdriver.chrome.driver", "C:\\chromedriver.exe");

driver = new ChromeDriver();

driver.get("https://myanimelist.net/");

driver.manage().window().maximize();

}

*@Test*(priority = 1)

public void testLogin() {

WebElement loginButton = driver.findElement(By.*xpath*("//a[contains(text(),'Login')]"));

loginButton.click();

WebElement usernameField = driver.findElement(By.*id*("loginUserName"));

WebElement passwordField = driver.findElement(By.*id*("login-password"));

WebElement loginSubmitButton = driver.findElement(By.*xpath*("//input[@value='Login']"));

usernameField.sendKeys("langxitrum");

passwordField.sendKeys("030102Ky");

loginSubmitButton.click();

try {

Thread.*sleep*(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

*@Test*(priority = 2)

public void testSupport() throws InterruptedException {

WebElement searchButton = driver.findElement(By.*xpath*("//\*[@id=\"nav\"]/li[7]/a"));

searchButton.click();

WebElement searchButton1 = driver.findElement(By.*xpath*("//\*[@id=\"nav\"]/li[7]/ul/li[2]/a"));

searchButton1.click();

WebElement searchSupport = driver.findElement(By.*xpath*("//\*[@id=\"content\"]/div[4]/span"));

searchSupport.click();

try {

Thread.*sleep*(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

*@Test*(priority = 3)

public void testSubmit() throws InterruptedException {

WebElement name = driver.findElement(By.*id*("name"));

WebElement email = driver.findElement(By.*id*("email"));

WebElement inquiryy = driver.findElement(By.*id*("inquiry"));

name.sendKeys("Khuong"); //

[email.sendKeys("kanabi29030610@gmail.com](mailto:email.sendKeys(%22kanabi29030610@gmail.com)"); //

inquiryy.sendKeys("so good");

WebElement device = driver.findElement(By.*id*("device\_type"));

Select se = new Select(device);

se.selectByIndex(2);

Thread.*sleep*(2000);

WebElement inquiry = driver.findElement(By.*id*("inquiry\_type"));

Select se1 = new Select(inquiry);

se1.selectByIndex(2);

Thread.*sleep*(2000);

WebElement SubmitButton = driver.findElement(By.*xpath*("//\*[@id=\"content\"]/div[5]/form/table/tbody/tr[7]/td[2]"));

SubmitButton.click();

}

*@AfterClass*

public void tearDown() {

try {

Thread.*sleep*(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

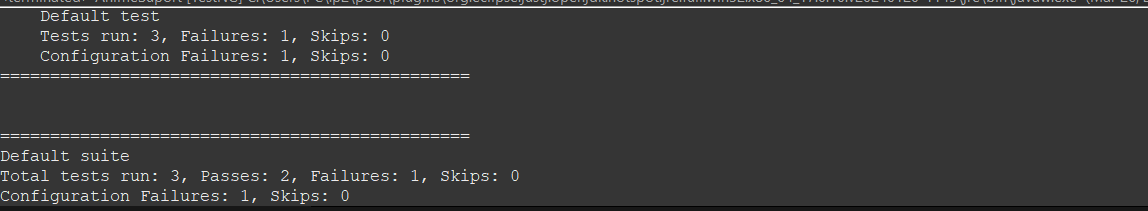
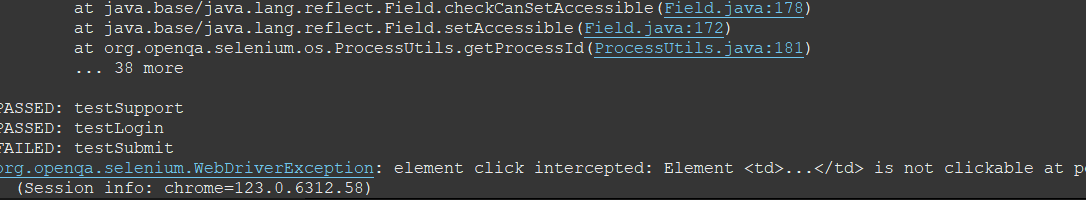
driver.quit();

}

}

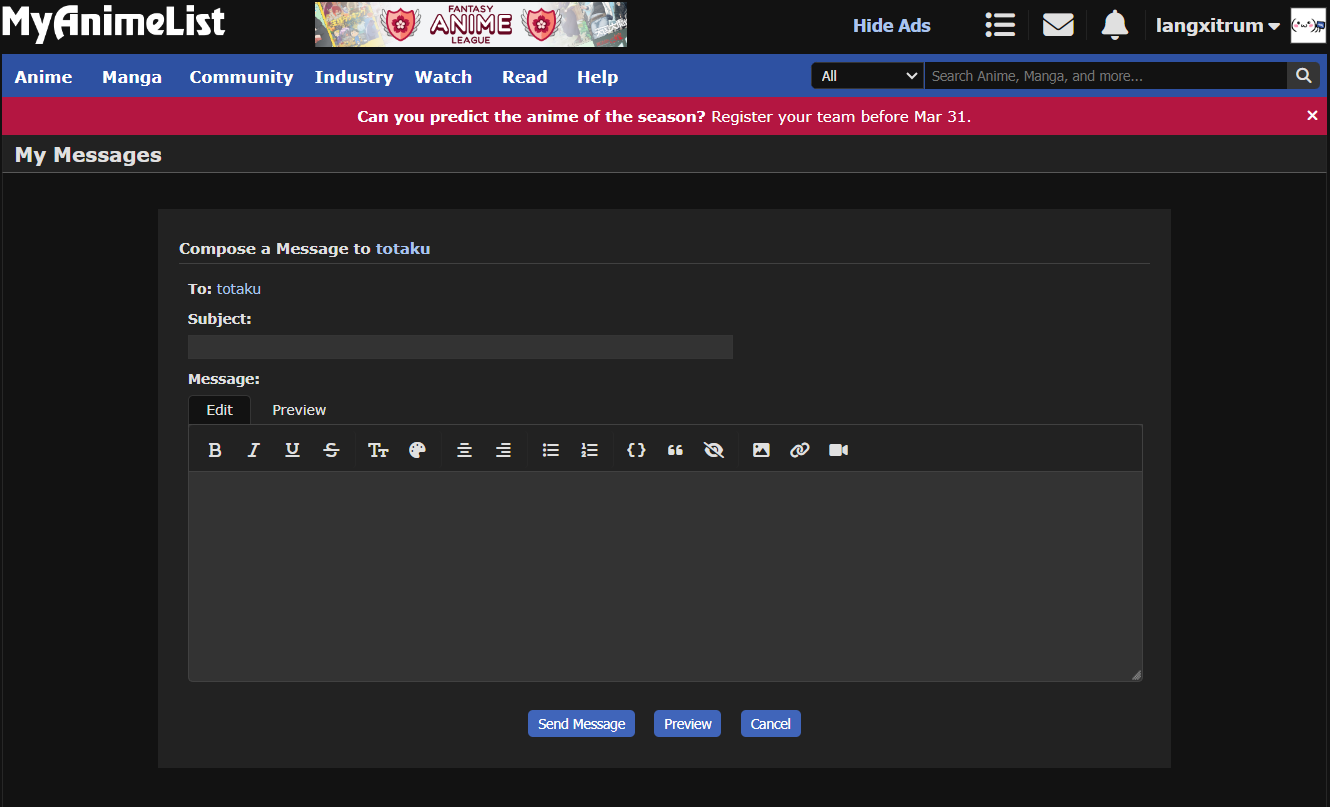
- Giải thích: Ta sử dụng TestNG và chức năng Test kèm theo độ ưu tiên của priority sau khi đăng nhập thành công sẽ đến bước tìm phần hỗ trợ và báo cáo sau khi mở form và điền nội dung cần hỗ trợ và xác nhận gửi.

4.3 Kết quả chức năng báo cáo



**5. Chức năng Nhắn tin với người dùng (Hoài Khương)**

5.1 Kiểm tra chức năng nhắn tin với người dùng



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID\_TC | Title | Các bước thực hiện | Kết quả mong đợi | Kết quả thực tế | Mục tiêu | RS |
| TC\_01 | Kiểm tra nội dung đăng nhập | 1. Mở form  2. Điền thông tin tài khoản và mật khẩu  3. Xem thông báo | Đăng nhập thành công | Đăng nhập thành công | Đăng nhập vào tài khoản để có thể chọn bộ anime yêu thích để chia sẻ | Pass |
| TC\_2 | Vào hộp thư và tìm kiếm người dùng cần gửi nội dung | 1. Vào thư messenger  2. Nhập tên người dùng  3. Xác nhận tìm kiếm | Tìm kiếm thành công | Tìm kiếm thành công | Để có thể vào được người dùng đó để gửi lời nhắn | Pass |
| TC\_3 | Điền form gửi lời nhắn | 1. Điền thông tin lời nhắn  2. Bấm nút xác nhận gửi | Gửi thành công | Gửi thất bại | Gửi lời nhắn đến người mình cần gửi | Fail |

5.2 Code chức năng nhắn tin với người dùng

package QLAnime;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class AnimeSendMessengerUser {

private WebDriver driver;

*@BeforeClass*

public void setUp() {

System.*setProperty*("webdriver.chrome.driver", "C:\\chromedriver.exe");

driver = new ChromeDriver();

driver.get("https://myanimelist.net/");

driver.manage().window().maximize();

}

*@Test*(priority = 1)

public void testLogin() {

WebElement loginButton = driver.findElement(By.*xpath*("//a[contains(text(),'Login')]"));

loginButton.click();

WebElement usernameField = driver.findElement(By.*id*("loginUserName"));

WebElement passwordField = driver.findElement(By.*id*("login-password"));

WebElement loginSubmitButton = driver.findElement(By.*xpath*("//input[@value='Login']"));

usernameField.sendKeys("langxitrum"); //

passwordField.sendKeys("030102Ky"); //

loginSubmitButton.click();

try {

Thread.*sleep*(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

*@Test*(priority = 2)

public void testSearchUser() throws InterruptedException {

WebElement searchButton = driver.findElement(By.*xpath*("//\*[@id=\"header-menu\"]/div[4]/a/i"));

searchButton.click();

Thread.*sleep*(3000);

WebElement searchButton1 = driver.findElement(By.*xpath*("//\*[@id=\"sendmsg\"]/a"));

searchButton1.click();

Thread.*sleep*(3000);

WebElement user = driver.findElement(By.*id*("toName"));

user.sendKeys("Wikanabi53");

Thread.*sleep*(3000);

WebElement searchButton2 = driver.findElement(By.*xpath*("//\*[@id=\"advancedSearchResultList\"]/div[1]/div/a/div/span"));

searchButton2.click();

try {

Thread.*sleep*(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

*@Test*(priority = 3)

public void testSubmit() throws InterruptedException {

WebElement subject = driver.findElement(By.*name*("subject"));

WebElement text = driver.findElement(By.*xpath*("//\*[@id=\"dialog\"]/tbody/tr/td/form/table/tbody/tr/td/div[3]/div[1]/div/textarea"));

subject.sendKeys("Add friend"); //

text.sendKeys("I am Kanabi"); //

Thread.*sleep*(2000);

WebElement submit = driver.findElement(By.*xpath*("//\*[@id=\"dialog\"]/tbody/tr/td/form/div/input[2]"));

submit.click();

Thread.*sleep*(2000);

}

*@AfterClass*

public void tearDown() {

try {

Thread.*sleep*(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

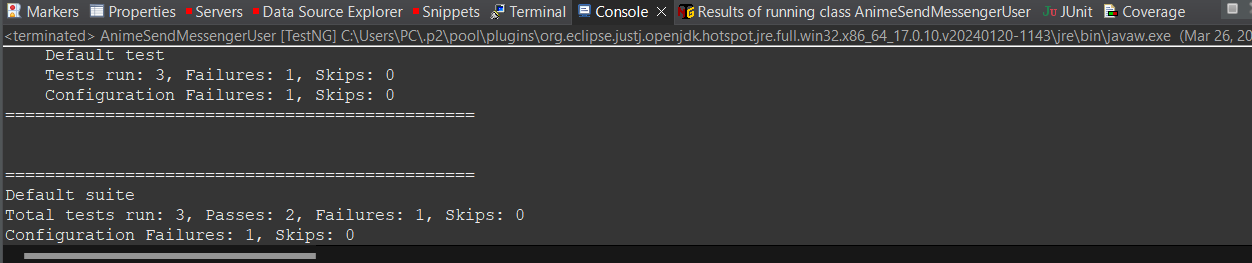
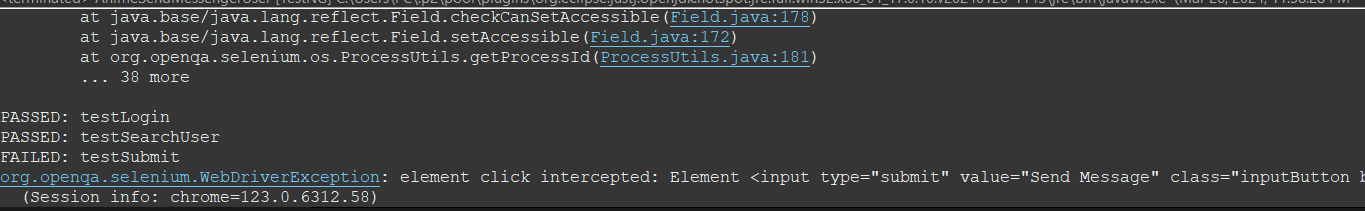
driver.quit();

}

}

- Giải thích: Ta sử dụng TestNG và chức năng Test kèm theo độ ưu tiên của priority sau khi đăng nhập thành công sẽ đến bước tìm mục lá thư và nhập tên người dùng sau đó điền nội dung gửi trong form và xác nhận gửi

5.3 Kết quả chức năng nhắn tin với người dùng



**6. Chức năng Tìm kiếm Anime (Đăng Kỷ)**

6.1 Kiểm tra chức năng tìm kiếm Anime

A screenshot of a computer

Description automatically generated

Giải thích: sử dụng TestNG và chức năng Test kèm theo độ ưu tiên của priority sau khi đăng nhập thành công sẽ đến bước tiếp theo nhấn vào nút tìm kiếm seach từ khóa Naruto và so sanh Naruto đó có tồn tại không.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID\_TC | Title | Các bước thực hiện | Kết quả mong đợi | Kết quả thực tế | Mục tiêu | Yêu cầu Đặc biệt |
| 1 | AnimeSearchTest | 1. Mở trang web MyAnimeList.Sử dụng thanh tìm kiếm để nhập từ khóa "anime". 3. Nhấn Enter hoặc nút tìm kiếm. | Hiển thị danh sách các anime liên quan đến từ khóa "anime". | Danh sách anime liên quan hiển thị đúng. | Kiểm tra tính năng tìm kiếm anime trên MyAnimeList. | pass |

6.2 Code chức năng tìm kiếm Anime

- Code: package KiemThu\_LyThuyet;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

public class AnimeSearchTest {

public String baseUrl = "<https://myanimelist.net/>";

String driverPath = "C:\\chromedriver.exe";

public WebDriver driver;

*@BeforeTest*

public void launchBrowser() {

System.***out***.println("Chạy trình duyệt chrome");

System.*setProperty*("webdriver.chrome.driver", driverPath);

driver = new ChromeDriver();

driver.get(baseUrl);

driver.manage().window().maximize();

}

*@Test*(priority = 1)

public void testLogin() {

WebElement loginButton = driver.findElement(By.*xpath*("//a[contains(text(),'Login')]"));

loginButton.click();

WebElement usernameField = driver.findElement(By.*id*("loginUserName"));

WebElement passwordField = driver.findElement(By.*id*("login-password"));

WebElement loginSubmitButton = driver.findElement(By.*xpath*("//input[@value='Login']"));

usernameField.sendKeys("langxitrum");

passwordField.sendKeys("030102Ky");

loginSubmitButton.click();

try {

Thread.*sleep*(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

*@Test*(priority = 2, dependsOnMethods = {"testLogin"})

public void testSearch() {

WebElement searchBox = driver.findElement(By.*name*("topkeyword"));

searchBox.click();

WebElement searchText = driver.findElement(By.*id*("topSearchText"));

searchText.sendKeys("Naruto: Shippuuden Movie 2 - Kizuna");

WebElement searchButton = driver.findElement(By.*xpath*("//\*[@id=\"topSearchButon\"]/i"));

searchButton.click();

// Chờ cho trang load

try {

Thread.*sleep*(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

WebElement searchResult = driver.findElement(By.*xpath*("//a[contains(text(),'Naruto: Shippuuden Movie 2 - Kizuna')]"));

searchResult.click();

// Chờ cho trang chi tiết load

try {

Thread.*sleep*(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

String pageTitle = driver.getTitle();

// So sánh tiêu đề của trang với tiêu đề mong đợi

String expectedTitle = "Naruto: Shippuuden Movie 2 - Kizuna - MyAnimeList.net";

Assert.*assertEquals*(pageTitle, expectedTitle, "Page title is not as expected.");

}

*@AfterTest*

public void terminateBrowser() {

driver.quit();

}

}

6.3 Kết quả chức năng tìm kiếm Anime

A black background with white text

Description automatically generated

**7. Chức năng Yêu thích Anime (Đăng Kỷ)**

7.1 Kiểm tra chức năng yêu thích Anime

A screenshot of a computer

Description automatically generated

7.2 Kiểm tra chức năng yêu thích Anime

Code: package KiemThu\_LyThuyet;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

public class yeuthichanime {

public String baseUrl = "https://myanimelist.net/";

String driverPath = "C:\\chromedriver.exe";

public WebDriver driver;

*@BeforeTest*

public void launchBrowser() {

System.***out***.println("Chạy trình duyệt chrome");

System.*setProperty*("webdriver.chrome.driver", driverPath);

driver = new ChromeDriver();

driver.get(baseUrl);

driver.manage().window().maximize();

}

*@Test*(priority = 1)

public void testLogin() {

WebElement loginButton = driver.findElement(By.*xpath*("//a[contains(text(),'Login')]"));

loginButton.click();

WebElement usernameField = driver.findElement(By.*id*("loginUserName"));

WebElement passwordField = driver.findElement(By.*id*("login-password"));

WebElement loginSubmitButton = driver.findElement(By.*xpath*("//input[@value='Login']"));

usernameField.sendKeys("langxitrum");

passwordField.sendKeys("030102Ky");

loginSubmitButton.click();

try {

Thread.*sleep*(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

*@Test*(priority = 2, dependsOnMethods = {"testLogin"})

public void testSearch() {

WebElement searchBox = driver.findElement(By.*name*("topkeyword"));

searchBox.click();

WebElement searchText = driver.findElement(By.*id*("topSearchText"));

searchText.sendKeys("Naruto: Shippuuden Movie 2 - Kizuna");

WebElement searchButton = driver.findElement(By.*xpath*("//\*[@id=\"topSearchButon\"]/i"));

searchButton.click();

try {

Thread.*sleep*(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

WebElement searchResult = driver.findElement(By.*xpath*("//a[contains(text(),'Naruto: Shippuuden Movie 2 - Kizuna')]"));

searchResult.click();

try {

Thread.*sleep*(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

String pageTitle = driver.getTitle();

String expectedTitle = "Naruto: Shippuuden Movie 2 - Kizuna - MyAnimeList.net";

Assert.*assertEquals*(pageTitle, expectedTitle, "Page title is not as expected.");

}

*@Test*(priority = 3, dependsOnMethods = {"testSearch"})

public void testAddToFavorites() {

WebElement addToFavoritesButton = driver.findElement(By.*xpath*("//a[@id='myinfo\_status']"));

addToFavoritesButton.click();

try {

Thread.*sleep*(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

*@AfterTest*

public void terminateBrowser() {

driver.quit();

}

}

+ Giải thích:

| ID Test case | Các bước thực hiện | Kết quả mong đợi | Kết quả thực tế | Mục tiêu | RS |
| --- | --- | --- | --- | --- | --- |
| yeuthichanime | 1. Truy cập vào trang MyAnimeList.. Tìm kiếm anime mong muốn bằng cách sử dụng chức năng tìm kiếm. Chọn anime từ kết quả tìm kiếm để xem thông tin chi tiết. Nhấn vào nút "Thêm vào yêu thích" hoặc tương tự. | Anime được thêm vào danh sách yêu thích của người dùng | Anime được thêm vào danh sách yêu thích thành công | Kiểm tra tính năng thêm anime yêu thích. | Pass |

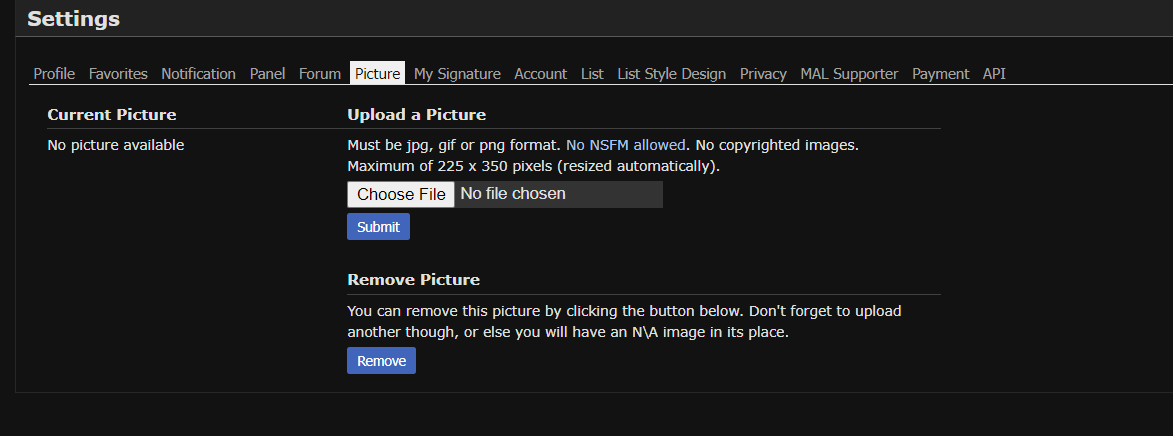
7.3 Kết quả chức năng yêu thích Anime

A screenshot of a computer program

Description automatically generated

**8. Chức năng Cập nhật ảnh đại diện (Đăng Kỷ)**

8.1 Kiểm tra chức năng Cập nhật ảnh đại diện



8.2 Code chức năng Cập nhật ảnh đại diện

Code: package KiemThu\_LyThuyet;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.awt.AWTException;

import java.awt.Robot;

import java.awt.Toolkit;

import java.awt.datatransfer.StringSelection;

import java.awt.event.KeyEvent;

public class upanhdaidien {

public static void main(String[] args) throws InterruptedException, AWTException {

System.*setProperty*("webdriver.chrome.driver", "C:\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.manage().window().maximize();

// Mở trang web MyAnimeList

driver.get("https://myanimelist.net/");

try {

WebElement loginButton = driver.findElement(By.*xpath*("//a[contains(text(),'Login')]"));

loginButton.click();

WebElement usernameField = driver.findElement(By.*id*("loginUserName"));

WebElement passwordField = driver.findElement(By.*id*("login-password"));

WebElement loginSubmitButton = driver.findElement(By.*xpath*("//input[@value='Login']"));

usernameField.sendKeys("langxitrum");

passwordField.sendKeys("030102Ky");

loginSubmitButton.click();

Thread.*sleep*(3000);

driver.get("https://myanimelist.net/editprofile.php?go=picture");

Thread.*sleep*(3000);

driver.findElement(By.*name*("file")).click();

driver.switchTo().activeElement();

Thread.*sleep*(2000);

StringSelection ss = new StringSelection("C:\\Users\\PC\\eclipse-workspace\\KTATUOU4\\src\\TH6\\anh.jpg");

Toolkit.*getDefaultToolkit*().getSystemClipboard().setContents(ss, null);

Robot robot = new Robot();

robot.delay(1000);

robot.keyPress(KeyEvent.***VK\_CONTROL***);

robot.keyPress(KeyEvent.***VK\_V***);

robot.keyRelease(KeyEvent.***VK\_V***);

robot.keyRelease(KeyEvent.***VK\_CONTROL***);

robot.delay(1000);

robot.keyPress(KeyEvent.***VK\_ENTER***);

robot.keyRelease(KeyEvent.***VK\_ENTER***);

Thread.*sleep*(3000);

} catch (Exception e) {

e.printStackTrace();

} finally {

driver.quit();

}

}

}

+ Giải thích:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID Test case | Các bước thực hiện | Kết quả mong đợi | Kết quả thực tế | Mục tiêu | RS |
| upanhdaidien | 1. Truy cập vào trang cài đặt profile.. Tìm và chọn tùy chọn "Thay đổi ảnh đại diện".. Chọn ảnh từ thiết bị của bạn.. Xác nhận việc thay đổi ảnh đại diện. | Ảnh đại diện mới được hiển thị trên profile của người dùng | Ảnh đại diện mới được hiển thị trên profile của người dùng | Kiểm tra tính năng up ảnh đại diện cho profile | Pass |
| upanhdaidien | 1. Truy cập vào trang cài đặt profile.. Tìm và chọn tùy chọn "Thay đổi ảnh đại diện". Thử chọn một file không hỗ trợ hoặc quá lớn. Xác nhận việc thay đổi ảnh đại diện. | Hiển thị thông báo lỗi hoặc không cho phép tải lên ảnh. | Hiển thị thông báo lỗi hoặc không cho phép tải lên ảnh. | Kiểm tra tính năng xử lý ảnh đại diện không hợp lệ | Pass |

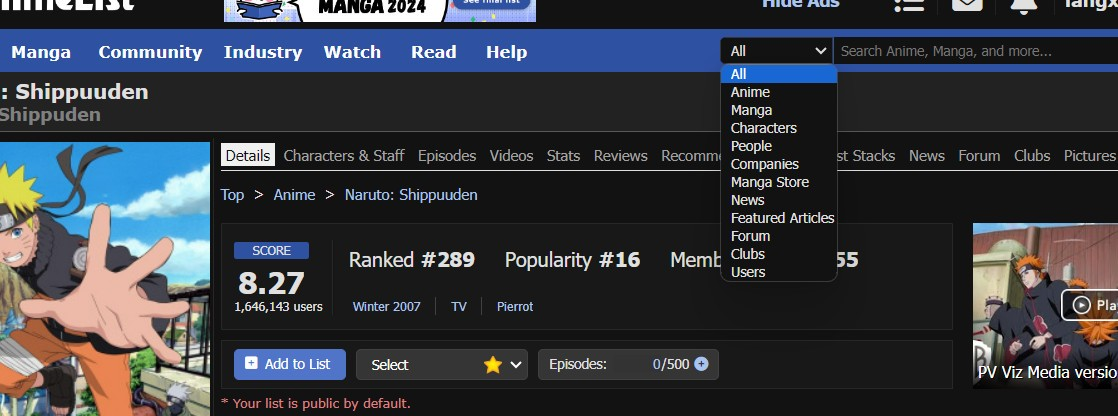
8.1 Kết quả chức năng Cập nhật ảnh đại diện

A screen shot of a computer

Description automatically generated

**9. Chức năng So sánh (Đăng Kỷ)**

9.1 Kiểm tra chức năng so sánh



9.2 Code chức năng so sánh

package KiemThu\_LyThuyet;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.support.ui.Select;

public class Sosanh {

public static void main(String[] args) throws InterruptedException {

System.setProperty("webdriver.chrome.driver", "C:\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.manage().window().maximize();

driver.get("https://myanimelist.net/");

//ket qua mong đợi

List<String> expect\_result = Arrays.asList(new String[] {"All", "Anime", "Manga", "Characters"});

// Kết quả thực tế

WebElement ActualDropdown = driver.findElement(By.id("topSearchValue"));

Select actual = new Select(ActualDropdown);

ActualDropdown.click();

Thread.sleep(2000);

ActualDropdown.click();

List<WebElement> actualListElements = actual.getOptions();

List<String> actual\_result = new ArrayList<String>();

for (WebElement options : actualListElements) {

actual\_result.add(options.getText());

}

if (expect\_result.equals(actual\_result)) {

System.out.println("Kết quả thực tế đúng với quả mong đợi");

} else {

System.out.println("Kết quả thực tế không đúng với quả mong đợi");

System.out.println("Kết quả mong đợi: " + expect\_result.toString());

System.out.println("Kết quả thực tế là: " + actual\_result.toString());

}

String default\_value\_expect = "Anime";

if(actual.getFirstSelectedOption().getText().equals(default\_value\_expect))

{

System.out.println("Giá trị mặc định của DropDown List đúng - pass");

}

else {

System.out.println("Giá trị mặc định của DropDown List không đúng - FAIL");

System.out.println("Kết quả mong đợi là: " + default\_value\_expect);

System.out.println("Kết quả thực tế là" + actual.getFirstSelectedOption().getText());

}

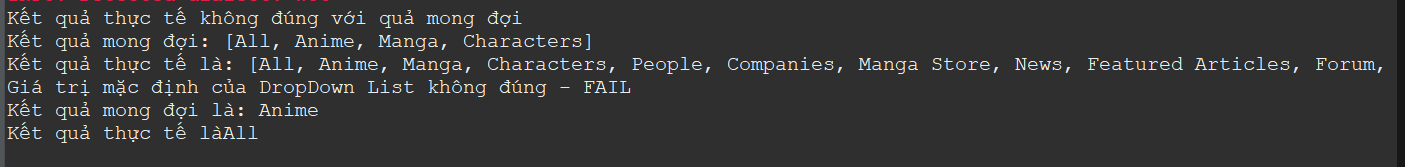
driver.close();

}

}

Giải thích: Sau khi đăng nhập thành công kiểm tra kết quả thực tế và kết quả mong đợi có giống nhau hay không và hiện ra kết quả

9.3 Kết quả chức năng so sánh



1. **Chức năng review (Thành Tâm)**
   1. Kiểm tra chức năng review

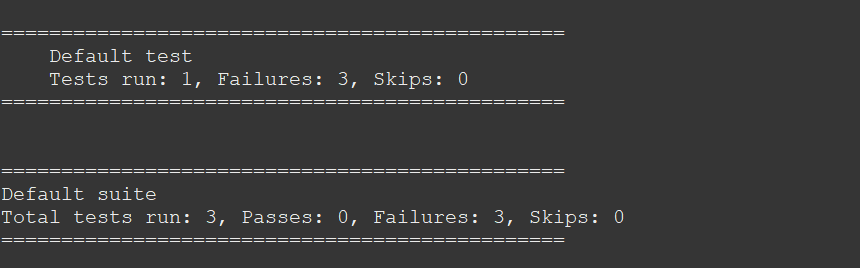
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID Test case** | **Các bước thực hiện** | **Kết quả mong đợi** | **Kết quả thực tế** | **Mục tiêu** | **RS** |
| Review | 1. Truy cập vào trang MyAnimeList.. Tìm kiếm anime mong muốn bằng cách sử dụng chức năng tìm kiếm. Chọn anime từ kết quả tìm kiếm để xem thông tin chi tiết. Nhấn vào nút "Thêm vào yêu thích" hoặc tương tự. | Anime được thêm vào danh sách yêu thích của người dùng | Anime được thêm vào danh sách yêu thích thành công | Kiểm tra tính năng thêm anime yêu thích. | Pass |

* 1. Code chức năng review



- Giải thích: Ta sử dụng TestNG và chức năng Test kèm theo độ ưu tiên của priority sau khi đăng nhập thành công sẽ đến bước kiểm tra các nút button review, kiểm tra xong sẽ hiển thị kết quả

* 1. Kết quả chức năng review



**11. Tính năng Kiểm tra nội dung ( Võ Đăng Kỷ)**

11.1 Kiểm tra định danh và text trên các tên trường

A screenshot of a computer

Description automatically generated

11.2 Code kiểm tra nội dung

Code: package KiemThu\_LyThuyet;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

public class KiemtraNoiDung {

    WebDriver driver;

    @BeforeTest

    public void setUp() throws InterruptedException {

        System.setProperty("webdriver.chrome.driver", "c:\\chromedriver.exe");

        driver = new ChromeDriver();

        driver.manage().window().maximize();

        driver.get("https://myanimelist.net/");

        Thread.sleep(2000);

    }

    @Test

    public void testNavigation() throws InterruptedException {

        driver.navigate().to("https://myanimelist.net/anime/1735/Naruto\_\_Shippuuden");

        Thread.sleep(2000);

        String dinhdanh = driver.getWindowHandle();

        System.out.println(dinhdanh);

        System.out.println(driver.getCurrentUrl());

        driver.navigate().back();

        Thread.sleep(2000);

        driver.navigate().forward();

        Thread.sleep(2000);

        driver.navigate().refresh();

        Thread.sleep(1000);

        System.out.println(driver.getTitle());

    }

    @AfterTest

    public void tearDown() {

        driver.quit();

    }

}

+ Giải thích: Ta sử dụng TestNG và chức năng Test kèm theo độ ưu tiên của priority sau khi đăng nhập thành công sẽ đến bước kiểm tra định danh và text trên các tên trường, kiểm tra xong sẽ hiển thị kết quả

| ID Test case | Các bước thực hiện | Kết quả mong đợi | Kết quả thực tế | Mục tiêu | RS |
| --- | --- | --- | --- | --- | --- |
| TC\_01 | Truy cập vào trang cần kiểm tra nội dung. Xem nội dung trang hiển thị. | Hiển thị nội dung hợp lệ, so sánh định danh, so sánh xem Shippuden (Naruto Shippuden) có toàn tại hay không | Nội dung trang hiển thị đúng và không có lỗi. | Kiểm tra tính chính xác và độ tin cậy của nội dung trang. | Pass |
| TC\_02 |  |  |  |  |  |

11.3 Kết quả kiểm tra nội dung

A screen shot of a computer

Description automatically generated

**12. Tính năng Tile (Anh Huy)**

12.2 Kiểm tra tính chính xác của tiêu đề trang.

A screenshot of a computer

Description automatically generated

12.2 Code tính năng tile

Code: package KiemThu\_LyThuyet;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class kiemtraTile {

    public static void main(String[] args) throws InterruptedException {

        // TODO Auto-generated method stub

        System.setProperty("webdriver.chrome.driver", "c:\\chromedriver.exe");

        WebDriver driver = new ChromeDriver();

        driver.manage().window().maximize();

        driver.navigate().to("https://myanimelist.net/");

        Thread.sleep(3000);

        String KetquaMongDoi = "MyAnimeList.net - Anime and Manga Database and Community";

        String KetquaThucTe = driver.getTitle();

        if (KetquaThucTe.contentEquals(KetquaMongDoi)) {

            System.out.println("PASS");

            System.out.println("Kết quả mong đợi là: " + KetquaMongDoi);

            System.out.println("Kết quả thực tế là: " + KetquaThucTe);

        } else {

            System.out.println("FAIL");

            System.out.println("Kết quả mong đợi là: " + KetquaMongDoi);

            System.out.println("Kết quả thực tế là: " + KetquaThucTe);

        }

    driver.close();

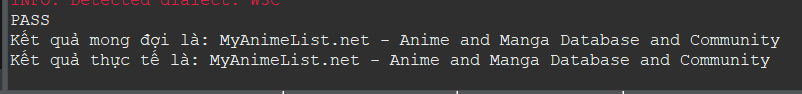
    }

}

- Giải thích: Ta sử dụng TestNG và chức năng Test kèm theo độ ưu tiên của priority sau khi đăng nhập thành công sẽ đến bước kiểm tra Kiểm tra tính chính xác của tiêu đề trang.

| ID Test case | Các bước thực hiện | Kết quả mong đợi | Kết quả thực tế | Mục tiêu | RS |
| --- | --- | --- | --- | --- | --- |
| kiemtraTile | Đăng Nhập thành công | Hiển thị tiêu đề trang chính xác là "MyAnimeList.net ". | Tiêu đề trang chính xác là "MyAnimeList.net ". | Kiểm tra tính chính xác của tiêu đề trang. | Pass |

12.3Kết quả tính năng tile



**13 Tính năng Kiểm tra từ khóa ( Trần Bảo Anh Huy)**

13.1 Kiểm tra title và text trên các tên trường

A screenshot of a video game

Description automatically generated

13.2 Code Tính năng Kiểm tra từ khóa

Code: package KiemThu\_LyThuyet;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

import org.testng.Reporter;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

public class Kiemtratukhoa {

    private WebDriver driver;

    @BeforeTest

    public void setUp() {

        System.setProperty("webdriver.chrome.driver", "c:\\chromedriver.exe");

        driver = new ChromeDriver();

    }

    @Test

    public void testGetMetaDescription() {

        driver.get("https://myanimelist.net/anime/1735/Naruto\_\_Shippuuden");

        // Lấy nội dung

        WebElement metaDescriptionElement = driver.findElement(By.cssSelector("meta[name='description']"));

        String metaDescriptionContent = metaDescriptionElement.getAttribute("content");

        // Kiểm tra

        Assert.assertTrue(metaDescriptionContent.contains("Looking for information on the anime Naruto: Shippuuden (Naruto Shippuden)?"));

        //  in ra màn hình

        Reporter.log("Meta Description: " + metaDescriptionContent);

    }

    @AfterTest

    public void tearDown() {

        driver.quit();

    }

}

- Giải thích: Ta sử dụng TestNG và chức năng Test kèm theo độ ưu tiên của priority sau khi đăng nhập thành công sẽ đến bước kiểm tra Kiểm Kiểm tra title và text trên các tên trường

| ID Test case | Các bước thực hiện | Kết quả mong đợi | Kết quả thực tế | Mục tiêu | RS |
| --- | --- | --- | --- | --- | --- |
| Kiemtratukhoa | Không cần bất kỳ bước nào (trang đã được truy cập trước đó). | Hiển thị từ khóa "Looking for information on the anime Naruto: Shippuuden (Naruto Shippuden)?" | Hiển thị từ khóa "Looking for information on the anime Naruto: Shippuuden (Naruto Shippuden)?" | Kiểm tra tính chính xác của hiển thị từ khóa. | Pass |

13.3 Kết quả Tính năng Kiểm tra từ khóa

A screen shot of a computer

Description automatically generated

**14 Tính năng đánh giá ( Trần Bảo Anh Huy)**

14.1 Kiểm tra mức độ đánh giá hài long anime

A screenshot of a video game

Description automatically generated

14.2 Code Tính năng Kiểm tra từ khóa

package KiemThu\_LyThuyet;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.Select;

import org.openqa.selenium.support.ui.WebDriverWait;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class MyAnimeListRatingTest {

private WebDriver driver;

@BeforeClass

public void setUp() {

System.setProperty("webdriver.chrome.driver", "C:\\chromedriver.exe");

driver = new ChromeDriver();

}

@Test

public void login() {

driver.get("https://myanimelist.net/");

try {

WebElement loginButton = driver.findElement(By.xpath("//a[contains(text(),'Login')]"));

loginButton.click();

WebElement usernameField = driver.findElement(By.id("loginUserName"));

WebElement passwordField = driver.findElement(By.id("login-password"));

WebElement loginSubmitButton = driver.findElement(By.xpath("//input[@value='Login']"));

usernameField.sendKeys("langxitrum");

passwordField.sendKeys("030102Ky");

loginSubmitButton.click();

WebDriverWait wait = new WebDriverWait(driver, 10);

wait.until(ExpectedConditions.urlContains("myanimelist.net")); // Wait for login success

Thread.sleep(3000); // Wait for page to load after login

} catch (Exception e) {

e.printStackTrace();

}

}

@Test(dependsOnMethods = "login")

public void rateAnime() {

String animeUrl = "https://myanimelist.net/anime/5114/Fullmetal\_Alchemist\_\_Brotherhood"; // Example URL

int rating = 10; // Rating value (1-10)

driver.get(animeUrl);

WebDriverWait wait = new WebDriverWait(driver, 9);

wait.until(ExpectedConditions.presenceOfElementLocated(By.id("myinfo\_score")));

WebElement ratingDropdown = driver.findElement(By.id("myinfo\_score"));

Select select = new Select(ratingDropdown);

select.selectByValue(String.valueOf(rating));

}

@AfterClass

public void tearDown() {

driver.quit();

}

}

- Giải thích: Ta sử dụng TestNG và chức năng Test kèm theo độ ưu tiên của priority sau khi đăng nhập thành công sẽ đến bước kiểm tra Kiểm thử tính năng đánh giá mức độ hài long từ 1 sao đến 10 sao.

| **Test case** | **Các bước thực hiện** | **Kết quả mong đợi** | **Kết quả thực tế** | **Mục tiêu** | **RS chức năng đánh giá** |
| --- | --- | --- | --- | --- | --- |
| TC\_01 | 1. Mở trình duyệt và truy cập MyAnimeList. | - Đăng nhập thành công.  - Chọn anime.  - Đánh giá anime. | - Hiển thị trang chi tiết của anime đã đánh giá. | Đảm bảo người dùng có thể đánh giá anime một cách chính xác. | Hệ thống cho phép đăng nhập và đánh giá anime. |

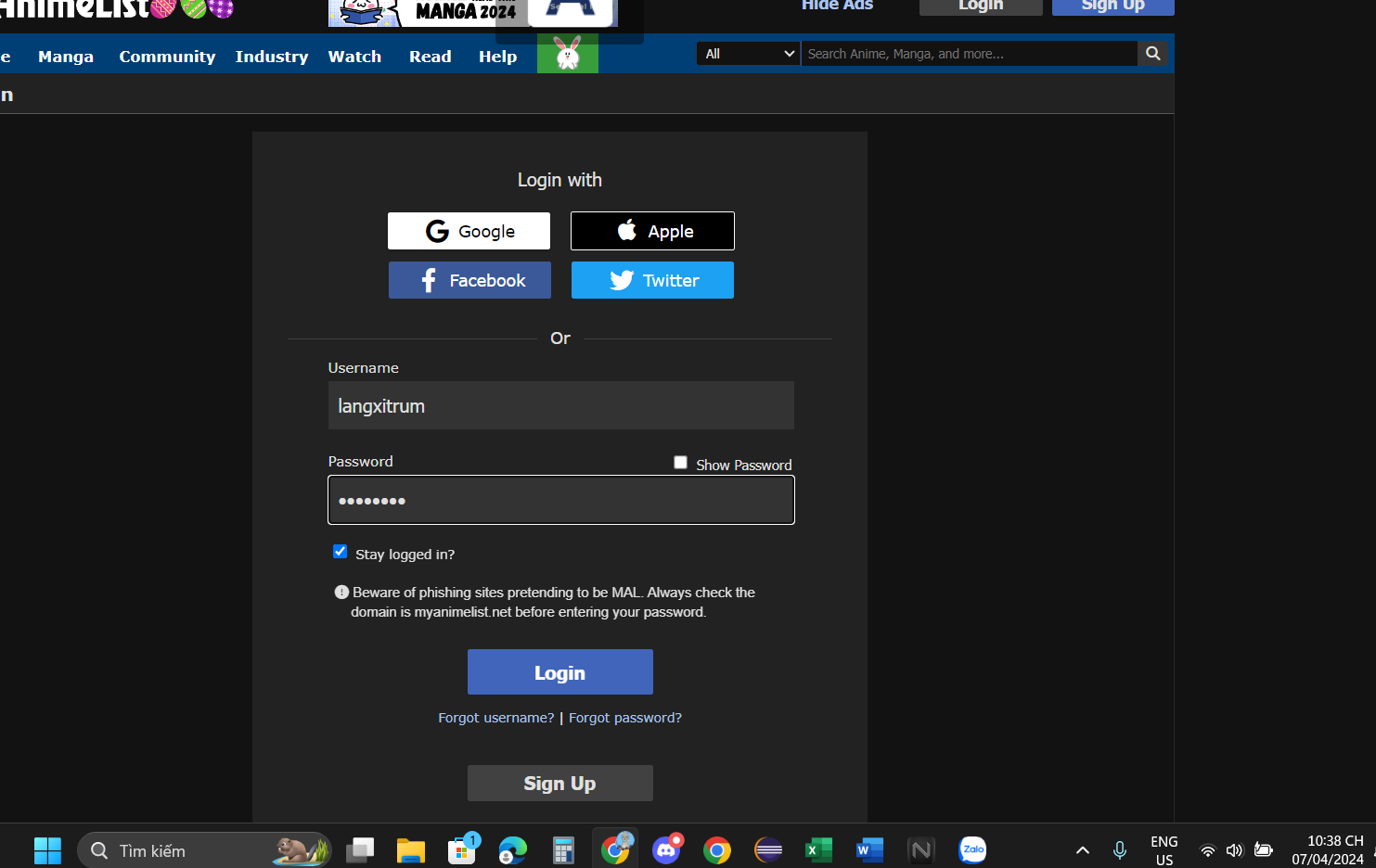
14.3 Kết quả Tính năng Kiểm tra từ khóa

A screenshot of a computer program

Description automatically generated

[**15. Tính năng đánh giá ( Võ Đăng Kỷ)**](#_Toc15648)

[15.1 Kiểm tra chức](#_Toc26568) năng đăng nhập qua Page Object Model (POM)



[15.2 Code Tính năng Kiểm tra đăng nhập](#_Toc8013)

Kiểm tra xác thực

package base;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import page.LoginPage;

public class Authentication {

WebDriver driver;

public Authentication(WebDriver driver) {

this.driver = driver;

}

public void login() throws InterruptedException {

// Khởi tạo trình duyệt Chrome

System.setProperty("webdriver.chrome.driver", "C:\\chromedriver.exe");

driver = new ChromeDriver();

driver.manage().window().maximize();

// Khởi tạo LoginPage

LoginPage loginPage = new LoginPage(driver);

// Mở trang đăng nhập của MyAnimeList

driver.get("https://myanimelist.net/login.php");

// Đăng nhập với thông tin tài khoản cụ thể

loginPage.login("langxitrum", "030102Ky");

}

}

**Phước thức đăng nhập**  
package page;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

public class LoginPage {

private WebDriver driver;

// Các phần tử trên trang đăng nhập

private By usernameField = By.id("loginUserName");

private By passwordField = By.id("login-password");

private By loginButton = By.xpath("//input[@value='Login']");

// Constructor nhận vào một đối tượng WebDriver

public LoginPage(WebDriver driver) {

this.driver = driver;

}

// Phương thức để nhập tên đăng nhập

public void enterUsername(String username) {

WebElement usernameElement = driver.findElement(usernameField);

usernameElement.clear();

usernameElement.sendKeys(username);

}

// Phương thức để nhập mật khẩu

public void enterPassword(String password) {

WebElement passwordElement = driver.findElement(passwordField);

passwordElement.clear();

passwordElement.sendKeys(password);

}

// Phương thức để nhấn nút đăng nhập

public void clickLoginButton() {

WebElement loginButtonElement = driver.findElement(loginButton);

loginButtonElement.click();

}

// Phương thức để đăng nhập với tên đăng nhập và mật khẩu cụ thể

public void login(String username, String password) {

enterUsername(username);

enterPassword(password);

clickLoginButton();

}

}

Thực hiện lấy từ Authentication để đăng nhập.

package testcase;

import org.openqa.selenium.WebDriver;

import org.testng.annotations.AfterClass;

import org.testng.annotations.Test;

import base.Authentication;

public class LoginPage {

WebDriver driver;

*@Test*

public void testLogin() throws InterruptedException {

// Khởi tạo Authentication

Authentication authentication = new Authentication(driver);

// Thực hiện đăng nhập

authentication.login();

// Thêm các kiểm tra hoặc hành động tiếp theo sau khi đăng nhập

}

*@AfterClass*

public void tearDown() {

driver.quit();

}

}

[**15.3 Kết quả Tính năng Kiểm tra đăng**](#_Toc10295) **nhập qua POM**

A screenshot of a computer program

Description automatically generated