SOFTWARE VERIFICATION, VALIDATION AND TESTING

TESTING DOCUMENTATION

Pegasus testing

Prepared by:
Naida Smailbegović
Lejla Srna
Danis Halkić

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

20.01.2023.

TABLE OF CONTENTS

Contents

1. Introduction

1.1. About the Project

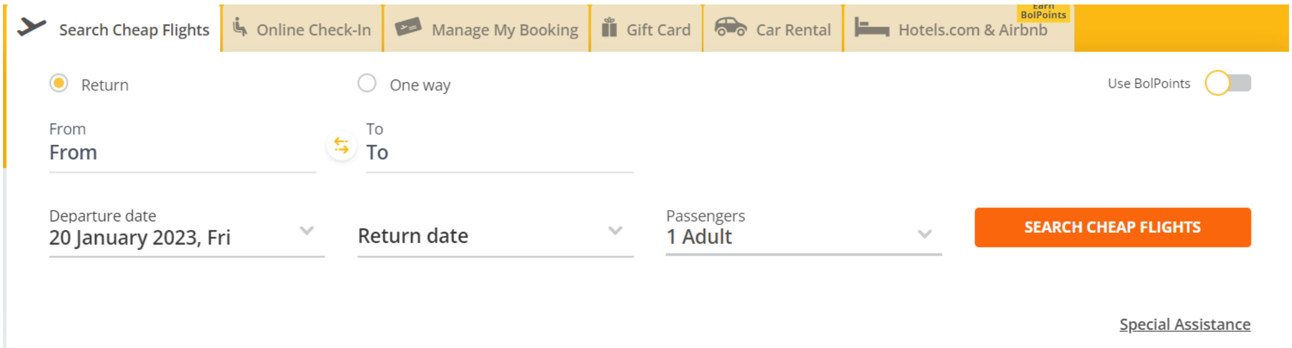
Pegasus is the web page used for booking flights for the Pegasus company. You can search for cheap flights, become their member, rent cars as well as book hotels.

This is the link for the homepage:

https://www.flypgs.com/en

1.2. Project Functionalities and Screenshots

Searching the flight, we need to pick cities of departure and where we want to go, pick date of departure and returning date as well as how many passengers are traveling.



● Return ○ One way

Use BolPoints

From
From

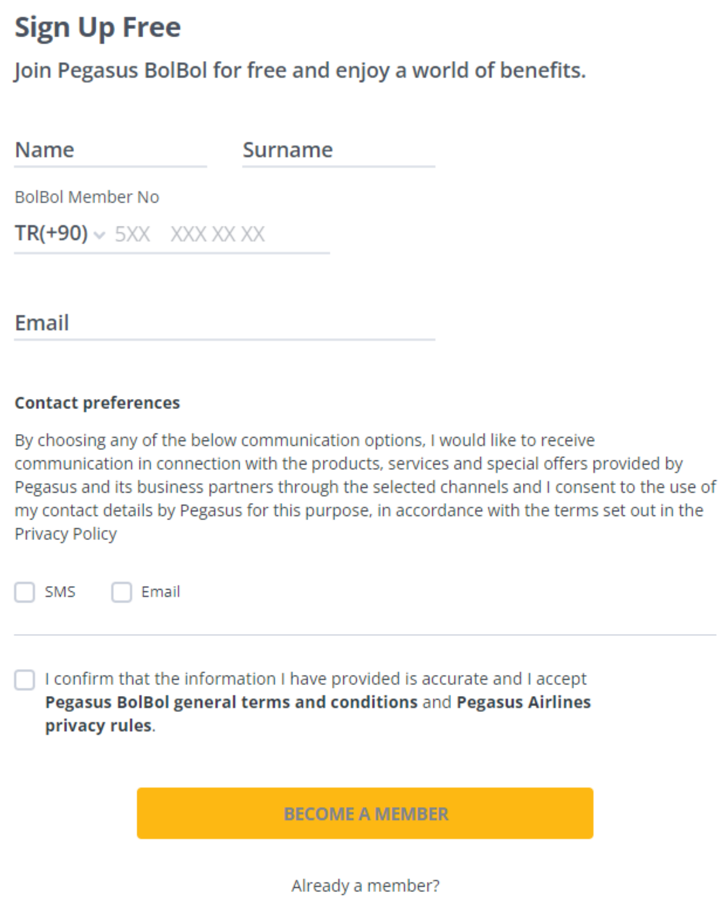
To
To

Departure date
20 January 2023, Fri

Return date

Passengers
1 Adult

Becoming the “BolBol member”, which is the way to collect points for flights and use them for discounts etc.



Sign Up Free

Join Pegasus BolBol for free and enjoy a world of benefits.

Name

Surname

BolBol Member No
TR(+90) 5XX XXX XX XX

Email

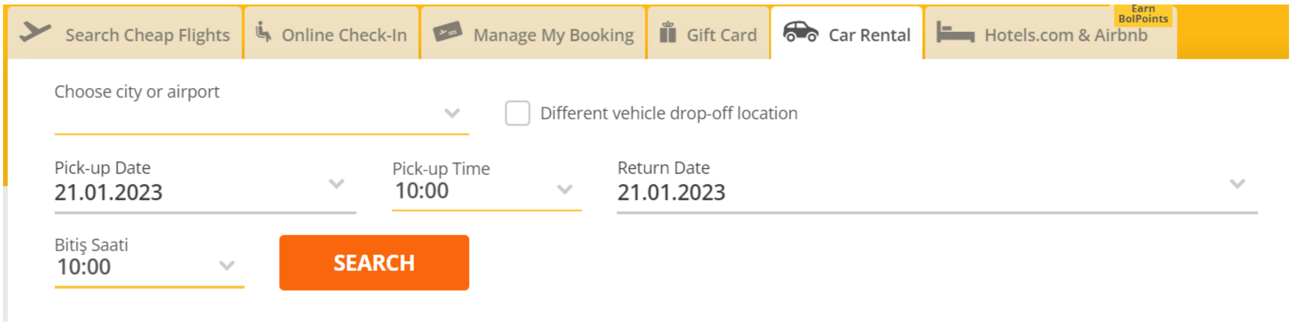
Contact preferences

By choosing any of the below communication options, I would like to receive communication in connection with the products, services and special offers provided by Pegasus and its business partners through the selected channels and I consent to the use of my contact details by Pegasus for this purpose, in accordance with the terms set out in the Privacy Policy

□ SMS □ Email

□ I confirm that the information I have provided is accurate and I accept **Pegasus BolBol general terms and conditions** and **Pegasus Airlines privacy rules**.

Renting a car is also an important functionality of this website.



Choose city or airport

□ Different vehicle drop-off location

Pick-up Date 21.01.2023

Pick-up Time 10:00

Return Date 21.01.2023

Bitiş Saati 10:00

2. Test Plan

2.1. Scope

We want to cover the whole webpage, but really focus on the main parts.

The main one is searching for flights, which we will test.

We think that the important part of this website is becoming their member, although we will not test does it really affect the price of the flight, since we need to have some “BOLBOL” points already.

We will also test some basic functionalities like changing the language, navigation bar and searching on the page.

2.2. Testing Environment and Tools

We used Selenium test environment, together with Java.

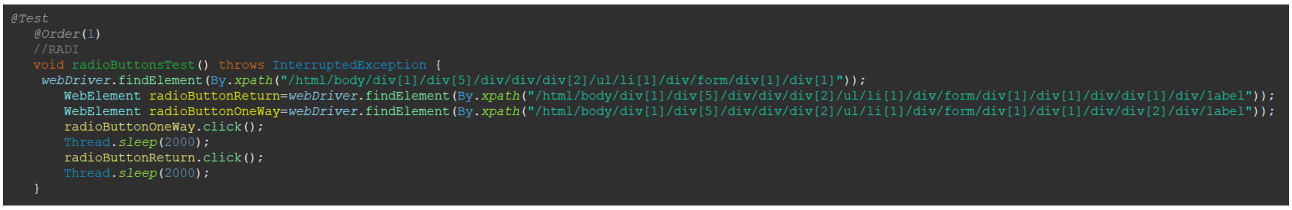
Some of the Selenium tools that we used are: Selenium IDE, Selenium WebDriver.

3. Test Execution

3.1. Radio Buttons test

On the home page in the Searching for flights part there is two radio buttons that we should select. They are used for user to choose if the want a return or one-way ticket.

Test Name: Radio Buttons				
Description: We want to test radio buttons which are in searching for flight part.				
Pre-condition(s):				
Test Steps: 1. Go to the home page. 2. Search for the part where the radio buttons are. 3. Click on the One Way radio button. 4. Click on the Return radio button.	**Test Data:** - we are using radio buttons which have its values.	**Expected Result:** Expected result was for the return radio button to be selected.	**Actual Result:** Return radio button is selected.	**Status:** PASS
Notes:				

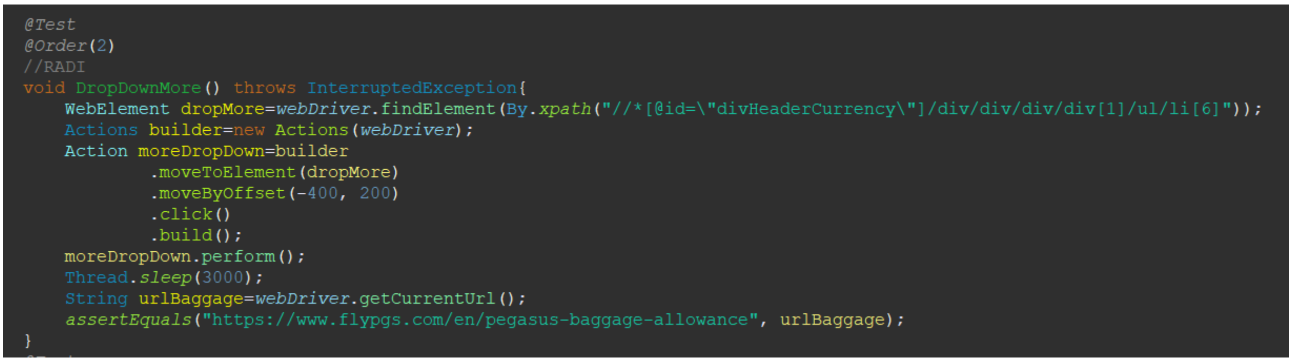


```
@Test
   @Order(1)
   //RADI
   void radioButtonsTest() throws InterruptedException {
    webDriver.findElement(By.xpath("/html/body/div[1]/div[5]/div/div/div[2]/ul/li[1]/div/form/div[1]/div[1]"));
       WebElement radioButtonReturn=webDriver.findElement(By.xpath("/html/body/div[1]/div[5]/div/div/div[2]/ul/li[1]/div/form/div[1]/div[1]/div/div[1]/div/label"));
       WebElement radioButtonOneWay=webDriver.findElement(By.xpath("/html/body/div[1]/div[5]/div/div/div[2]/ul/li[1]/div/form/div[1]/div[1]/div/div[2]/div/label"));
       radioButtonOneWay.click();
       Thread.sleep(2000);
       radioButtonReturn.click();
       Thread.sleep(2000);
   }
```

3.2. Navigation bar>More testing

On the home page there is a navigation bar. In this navigation bar we have many options and More is one of them. Under this option which is a drop down there is many more services. We want to test one of them.

<table>
<tr><td colspan="5"><b>Test Name:</b> More options test</td></tr>
<tr><td colspan="5"><b>Description:</b> Check if this option has workable url’s in the dropdown.</td></tr>
<tr><td colspan="5"><b>Pre-condition(s):</b></td></tr>
<tr><td><b>Test Steps:</b>
1. Open the Homepage
2. Find the “More” option in the navigation bar
3. Hover over it and the dropdown menu should appear
4. Get to the option “Baggage allowance” and click on it
5. Assert if the url worked.</td><td><b>Test Data:</b></td><td><b>Expected Result:</b>
The user is taken to the correct new page when clicking on the link.</td><td><b>Actual Result:</b>
The user is taken to the correct new page when clicking on the link.</td><td><b>Status:</b>
PASS</td></tr>
<tr><td colspan="5"><b>Notes:</b></td></tr>
</table>

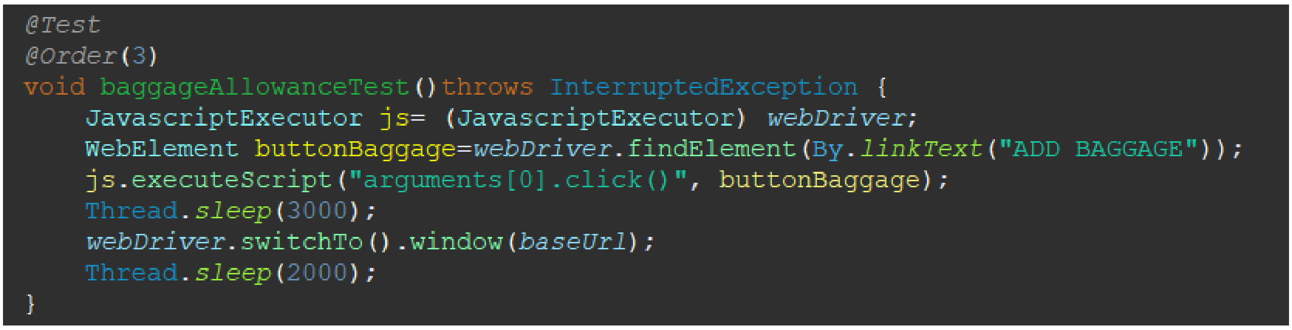


```java
@Test
@Order(2)
//RADI
void DropDownMore() throws InterruptedException{
    WebElement dropMore=webDriver.findElement(By.xpath("//*[@id=\"divHeaderCurrency\"]/div/div/div/div[1]/ul/li[6]"));
    Actions builder=new Actions(webDriver);
    Action moreDropDown=builder
            .moveToElement(dropMore)
            .moveByOffset(-400, 200)
            .click()
            .build();
    moreDropDown.perform();
    Thread.sleep(3000);
    String urlBaggage=webDriver.getCurrentUrl();
    assertEquals("https://www.flypgs.com/en/pegasus-baggage-allowance", urlBaggage);
}
```

3.3.Baggage allowance page

After getting on this new page we want to test if the button on it works. This button allows you to add baggage on your already bought tickets, which is the reason we stopped on just opening this page.

Test Name: Testing page that we opened				
Description: Check if the button on this page works and does it open new tab.				
Pre-condition(s):				
Test Steps: 1. New page is opened 2. Click on the button “Add Baggage” 3. It will open a new tab with new page for selecting the baggage 4. Return to the homepage	**Test Data:**	**Expected Result:** The user is taken to a new page when clicking on the button, and taken back on the homepage.	**Actual Result:** The user is taken to a new page when clicking on the button, and taken back to the homepage, but the test fails saying that the window of a homepage does not exist.	**Status:** FAIL
Notes:				



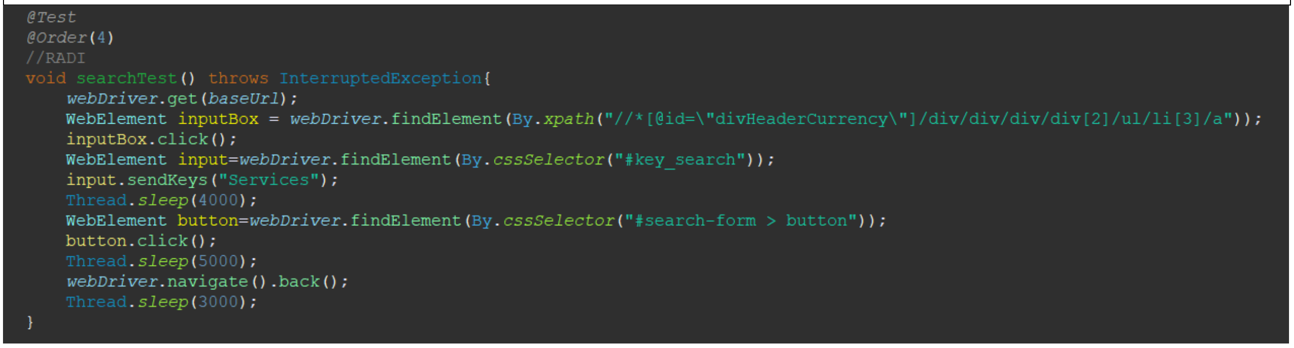
```
@Test
@Order(3)
void baggageAllowanceTest()throws InterruptedException {
    JavascriptExecutor js= (JavascriptExecutor) webDriver;
    WebElement buttonBaggage=webDriver.findElement(By.linkText("ADD BAGGAGE"));
    js.executeScript("arguments[0].click()", buttonBaggage);
    Thread.sleep(3000);
    webDriver.switchTo().window(baseUrl);
    Thread.sleep(2000);
}
```

3.4.Testing Search on the page

On the homepage there is a search in the right upper corner which is used for searching through the page. We want to test if it works and will it lead us to the searched page.

Test Name: Test Search				
Description: Check if the search works				
Pre-condition(s):				
Test Steps: 1. Find the Search box and click on it 2. Send data to the search input	**Test Data:** We need input for the search which is “Services” in this case	**Expected Result:** The user is navigated to the search results, and then back to	**Actual Result:** The user is navigated to the search results, and then back to	**Status:** PASS

3. Search for it 4. Wait for the results and go back to the homepage		homepage.	homepage.	
Notes:				

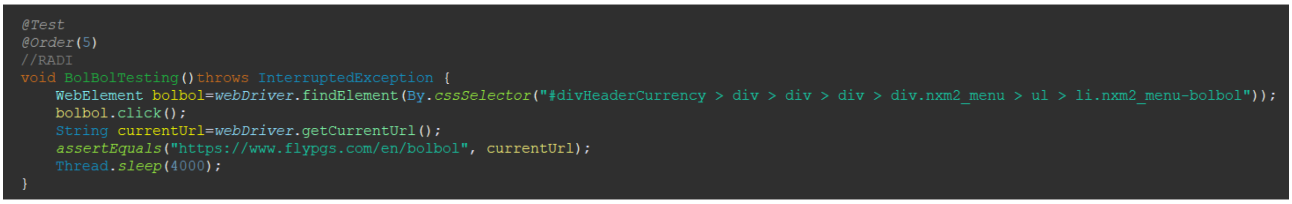


```
@Test
@Order(4)
//RADI
void searchTest() throws InterruptedException{
    webDriver.get(baseUrl);
    WebElement inputBox = webDriver.findElement(By.xpath("//*[@id=\"divHeaderCurrency\"]/div/div/div/div[2]/ul/li[3]/a"));
    inputBox.click();
    WebElement input=webDriver.findElement(By.cssSelector("#key_search"));
    input.sendKeys("Services");
    Thread.sleep(4000);
    WebElement button=webDriver.findElement(By.cssSelector("#search-form > button"));
    button.click();
    Thread.sleep(5000);
    webDriver.navigate().back();
    Thread.sleep(3000);
}
```

3.5.Testing BolBol button

On the navigation bar There is a BOLBOL button which will lead us to the bolbol page.

Test Name: BolBol button				
Description: Check if the button works				
Pre-condition(s):				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**
1. Find the button and click on it 2. Assert if the new page is correct		User is lead to the new page.	User is lead to the new page.	PASS
Notes:				



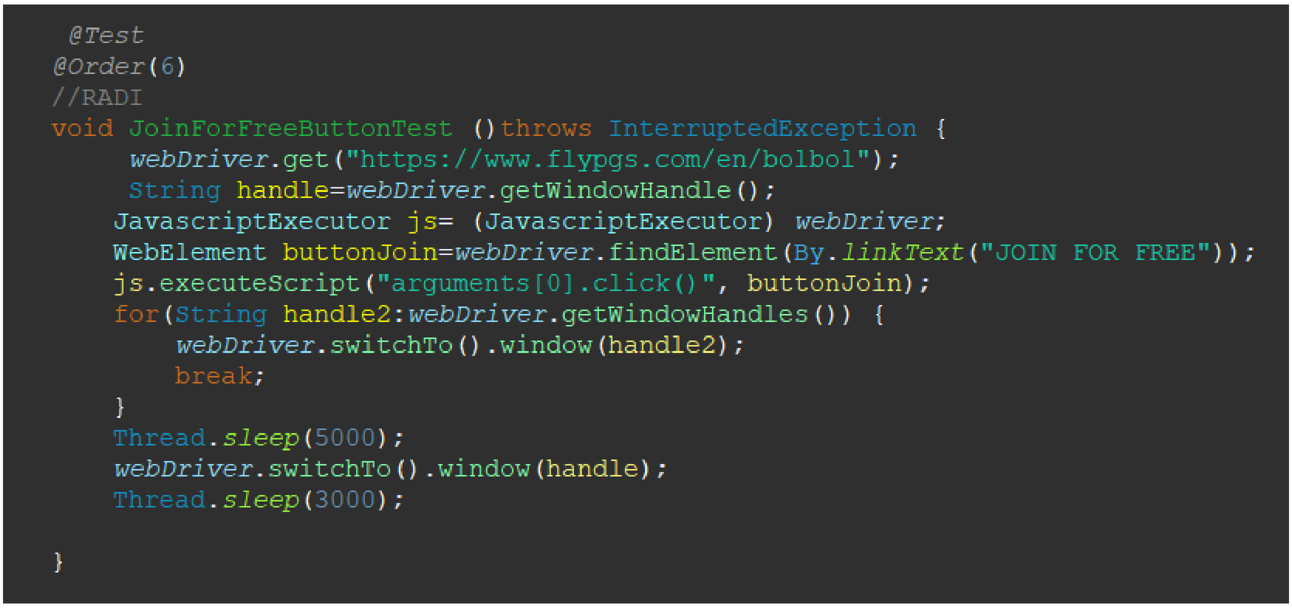
```
@Test
@Order(5)
//RADI
void BolBolTesting()throws InterruptedException {
    WebElement bolbol=webDriver.findElement(By.cssSelector("#divHeaderCurrency > div > div > div > div.nxm2_menu > ul > li.nxm2_menu-bolbol"));
    bolbol.click();
    String currentUrl=webDriver.getCurrentUrl();
    assertEquals("https://www.flypgs.com/en/bolbol", currentUrl);
    Thread.sleep(4000);
}
```

3.6.Testing button for joining

When we get to the BolBol page we have a button which will lead us to the form for becoming a member.

Test Name: Test Button for joining				
Description: Check if the search works				
Pre-condition(s):				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**

1. Getting on BolBol page 2. Finding a button for joining 3. Click on it 4. Get to the page for signing up		The user is lead to the page for signing up.	The user is lead to the page for signing up.	PASS
Notes:				



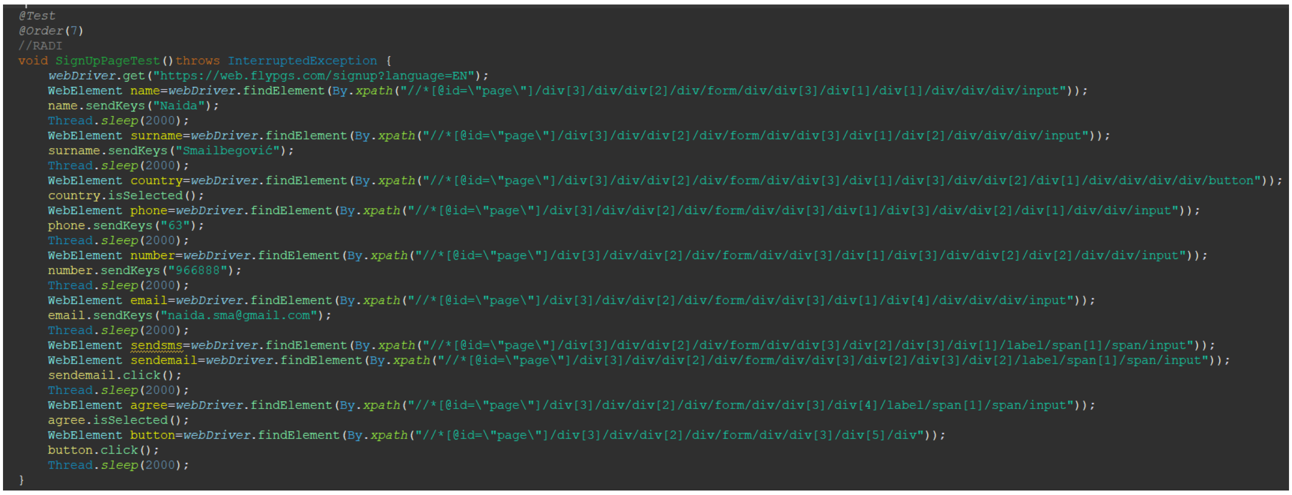
```
 @Test
@Order(6)
//RADI
void JoinForFreeButtonTest ()throws InterruptedException {
     webDriver.get("https://www.flypgs.com/en/bolbol");
     String handle=webDriver.getWindowHandle();
    JavascriptExecutor js= (JavascriptExecutor) webDriver;
    WebElement buttonJoin=webDriver.findElement(By.linkText("JOIN FOR FREE"));
    js.executeScript("arguments[0].click()", buttonJoin);
    for(String handle2:webDriver.getWindowHandles()) {
        webDriver.switchTo().window(handle2);
        break;
    }
    Thread.sleep(5000);
    webDriver.switchTo().window(handle);
    Thread.sleep(3000);

}
```

3.7.Testing signup form

After clicking on the button the signup page which has form on it will be opened. On this page we can find checkboxes which will be tested too. We need to send it values and test if everything works correctly.

Test Name: Signup form				
Description: Check if the form works				
Pre-condition(s):				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**
1. After opening this page we need to find every part of the form and send values to it 2. Confirming contact preferences	Name Surname Country Phone number E-mail	Everything in the form works correctly.	Everything in the form works correctly.	PASS

3. Agreeing on the terms and conditions 4. Click on the Become a member button				
Notes: We used a wrong Telephone number on purpose so that we don’t actually become a member.				

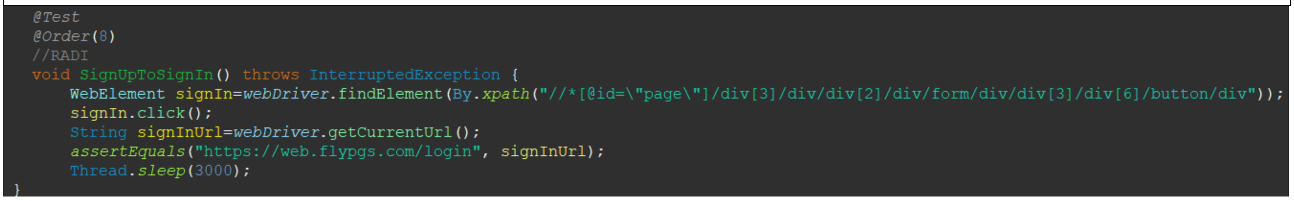


```
@Test
@Order(7)
//RADI
void SignUpPageTest()throws InterruptedException {
    webDriver.get("https://web.flypgs.com/signup?language=EN");
    WebElement name=webDriver.findElement(By.xpath("//*[@id=\"page\"]/div[3]/div/div[2]/div/form/div/div[3]/div[1]/div[1]/div/div/div/input"));
    name.sendKeys("Naida");
    Thread.sleep(2000);
    WebElement surname=webDriver.findElement(By.xpath("//*[@id=\"page\"]/div[3]/div/div[2]/div/form/div/div[3]/div[1]/div[2]/div/div/div/input"));
    surname.sendKeys("Smailbegović");
    Thread.sleep(2000);
    WebElement country=webDriver.findElement(By.xpath("//*[@id=\"page\"]/div[3]/div/div[2]/div/form/div/div[3]/div[1]/div[3]/div/div[2]/div[1]/div/div/div/div/button"));
    country.isSelected();
    WebElement phone=webDriver.findElement(By.xpath("//*[@id=\"page\"]/div[3]/div/div[2]/div/form/div/div[3]/div[1]/div[3]/div/div[2]/div[1]/div/div/input"));
    phone.sendKeys("63");
    Thread.sleep(2000);
    WebElement number=webDriver.findElement(By.xpath("//*[@id=\"page\"]/div[3]/div/div[2]/div/form/div/div[3]/div[1]/div[3]/div/div[2]/div[2]/div/div/input"));
    number.sendKeys("966888");
    Thread.sleep(2000);
    WebElement email=webDriver.findElement(By.xpath("//*[@id=\"page\"]/div[3]/div/div[2]/div/form/div/div[3]/div[1]/div[4]/div/div/div/input"));
    email.sendKeys("naida.sma@gmail.com");
    Thread.sleep(2000);
    WebElement sendsms=webDriver.findElement(By.xpath("//*[@id=\"page\"]/div[3]/div/div[2]/div/form/div/div[3]/div[2]/div[3]/div[1]/label/span[1]/span/input"));
    WebElement sendemail=webDriver.findElement(By.xpath("//*[@id=\"page\"]/div[3]/div/div[2]/div/form/div/div[3]/div[2]/div[3]/div[2]/label/span[1]/span/input"));
    sendemail.click();
    Thread.sleep(2000);
    WebElement agree=webDriver.findElement(By.xpath("//*[@id=\"page\"]/div[3]/div/div[2]/div/form/div/div[3]/div[4]/label/span[1]/span/input"));
    agree.isSelected();
    WebElement button=webDriver.findElement(By.xpath("//*[@id=\"page\"]/div[3]/div/div[2]/div/form/div/div[3]/div[5]/div"));
    button.click();
    Thread.sleep(2000);
}
```

3.8.Testing Already a member link

At the end of the form and below a button there is a link that will lead us to the Login page if we already are a member.

Test Name: Test Already a member button				
Description: Check if the button works				
Pre-condition(s):				
Test Steps: 1. Finding the button. 2. Click on it. 3. Assert new url with Login page url.	**Test Data:**	**Expected Result:** The user is lead to the page for logging in.	**Actual Result:** The user is lead to the page for logging in.	**Status:** PASS
Notes:				

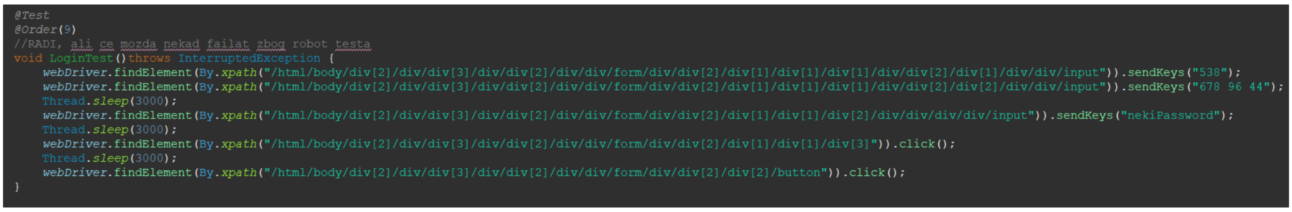


```
@Test
@Order(8)
//RADI
void SignUpToSignIn() throws InterruptedException {
    WebElement signIn=webDriver.findElement(By.xpath("//*[@id=\"page\"]/div[3]/div/div[2]/div/form/div/div[3]/div[6]/button/div"));
    signIn.click();
    String signInUrl=webDriver.getCurrentUrl();
    assertEquals("https://web.flypgs.com/login", signInUrl);
    Thread.sleep(3000);
}
```

3.9.Testing Login page

After getting on Login page we need to test if this form works.

Test Name: Test Login form				
Description: Check if the form works				
Pre-condition(s):				
Test Steps: 1. Sending the values to the form 2. Checking the Not a robot checkbox 3. Clicking a Login button	**Test Data:** Phone number Password	**Expected Result:** The form works, but the user does not exist.	**Actual Result:** The form works, but the user does not exist.	**Status:** PASS
Notes: "I'm not a robot" checkbox might not work because the page may need multiple recognitions.				

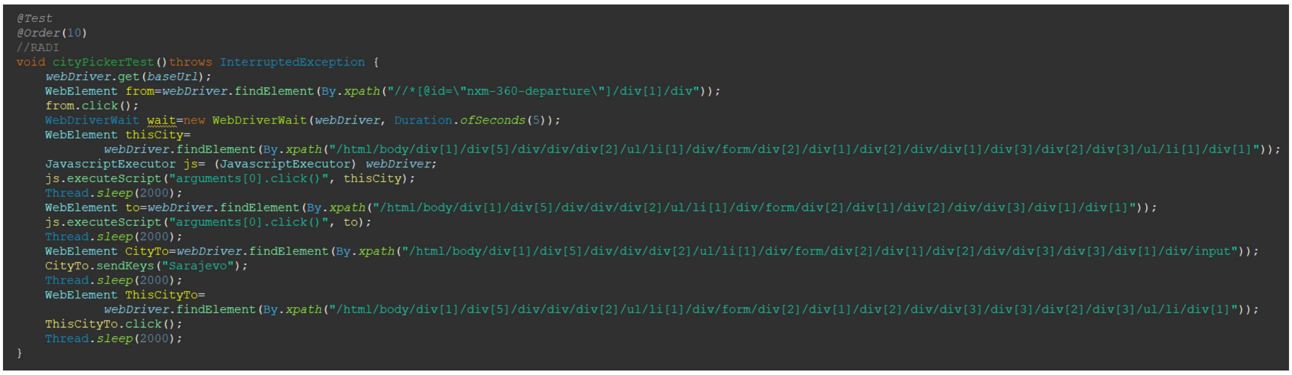


```
@Test
@Order(9)
//RADI, ali ce mozda nekad failat zbog robot testa
void LoginTest()throws InterruptedException {
    webDriver.findElement(By.xpath("/html/body/div[2]/div/div[3]/div/div[2]/div/div/form/div/div[2]/div[1]/div[1]/div[1]/div/div[2]/div[1]/div/div/input")).sendKeys("538");
    webDriver.findElement(By.xpath("/html/body/div[2]/div/div[3]/div/div[2]/div/div/form/div/div[2]/div[1]/div[1]/div[1]/div/div[2]/div[2]/div/div/input")).sendKeys("678 96 44");
    Thread.sleep(3000);
    webDriver.findElement(By.xpath("/html/body/div[2]/div/div[3]/div/div[2]/div/div/form/div/div[2]/div[1]/div[1]/div[2]/div/div/div/div/input")).sendKeys("nekiPassword");
    Thread.sleep(3000);
    webDriver.findElement(By.xpath("/html/body/div[2]/div/div[3]/div/div[2]/div/div/form/div/div[2]/div[1]/div[1]/div[3]")).click();
    Thread.sleep(3000);
    webDriver.findElement(By.xpath("/html/body/div[2]/div/div[3]/div/div[2]/div/div/form/div/div[2]/div[2]/button")).click();
}
```

3.10.Testing Picking the City

This is part of the Searching for flights. We need to pick city of departure and where we want to return. This works as a dropdown menu but we can also send values to it. We will test both.

Test Name: City picker Test				
Description: Check if dropdown menus and sending keys work				
Pre-condition(s):				
Test Steps: 1. Find City of departure part and click on it 2. Choose option from the dropdown menu 3. Click on the part where we choose city we want to fly to 4. Send value to it 5. Choose that city	**Test Data:** City To: Sarajevo	**Expected Result:** Both cities where selected.	**Actual Result:** Both cities where selected.	**Status:** PASS
Notes:				

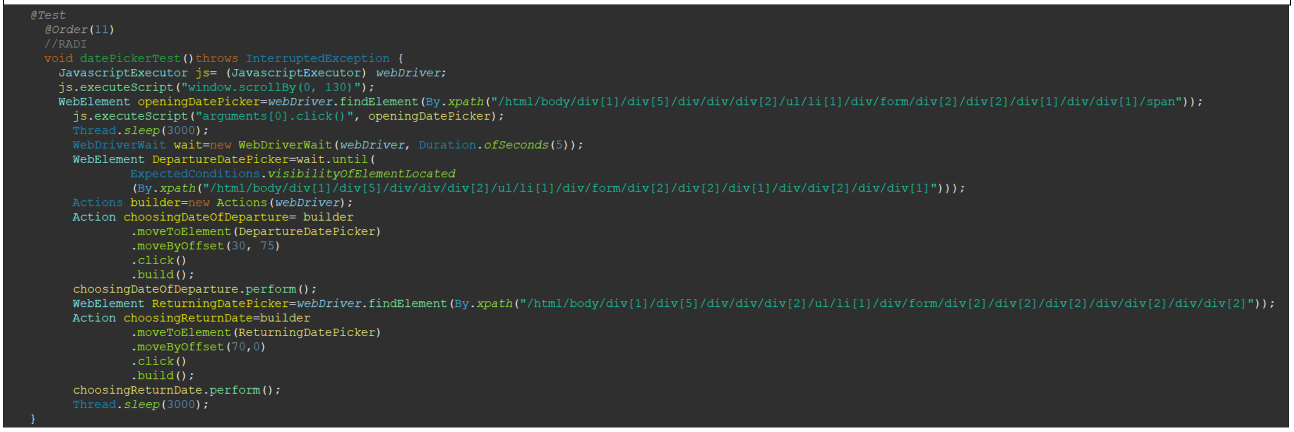


```
@Test
@Order(10)
//RADI
void cityPickerTest()throws InterruptedException {
    webDriver.get(baseUrl);
    WebElement from=webDriver.findElement(By.xpath("//*[@id=\"nxm-360-departure\"]/div[1]/div"));
    from.click();
    WebDriverWait wait=new WebDriverWait(webDriver, Duration.ofSeconds(5));
    WebElement thisCity=
            webDriver.findElement(By.xpath("/html/body/div[1]/div[5]/div/div/div[2]/ul/li[1]/div/form/div[2]/div[1]/div[2]/div/div[1]/div[3]/div[2]/div[3]/ul/li[1]/div[1]"));
    JavascriptExecutor js= (JavascriptExecutor) webDriver;
    js.executeScript("arguments[0].click()", thisCity);
    Thread.sleep(2000);
    WebElement to=webDriver.findElement(By.xpath("/html/body/div[1]/div[5]/div/div/div[2]/ul/li[1]/div/form/div[2]/div[1]/div[2]/div/div[3]/div[1]/div[1]"));
    js.executeScript("arguments[0].click()", to);
    Thread.sleep(2000);
    WebElement CityTo=webDriver.findElement(By.xpath("/html/body/div[1]/div[5]/div/div/div[2]/ul/li[1]/div/form/div[2]/div[1]/div[2]/div/div[3]/div[3]/div[1]/div/input"));
    CityTo.sendKeys("Sarajevo");
    Thread.sleep(2000);
    WebElement ThisCityTo=
            webDriver.findElement(By.xpath("/html/body/div[1]/div[5]/div/div/div[2]/ul/li[1]/div/form/div[2]/div[1]/div[2]/div/div[3]/div[3]/div[2]/div[3]/ul/li/div[1]"));
    ThisCityTo.click();
    Thread.sleep(2000);
}
```

3.11.Testing Date Picker

This is part of the Searching for flights. We need to pick Departure Date and when do we want to return.

Test Name: Date Picker Test				
Description: Check if the date picker works				
Pre-condition(s):				
Test Steps: 1. Opening Date picker 2. Moving to the departure date part 3. Clicking on the 26th of January 4. Clicking on the Return date 5. Moving to the Return date picker 6. Clicking on the 9th of February	**Test Data:**	**Expected Result:** Dates where chosen.	**Actual Result:** Dates where chosen.	**Status:** PASS
Notes:				

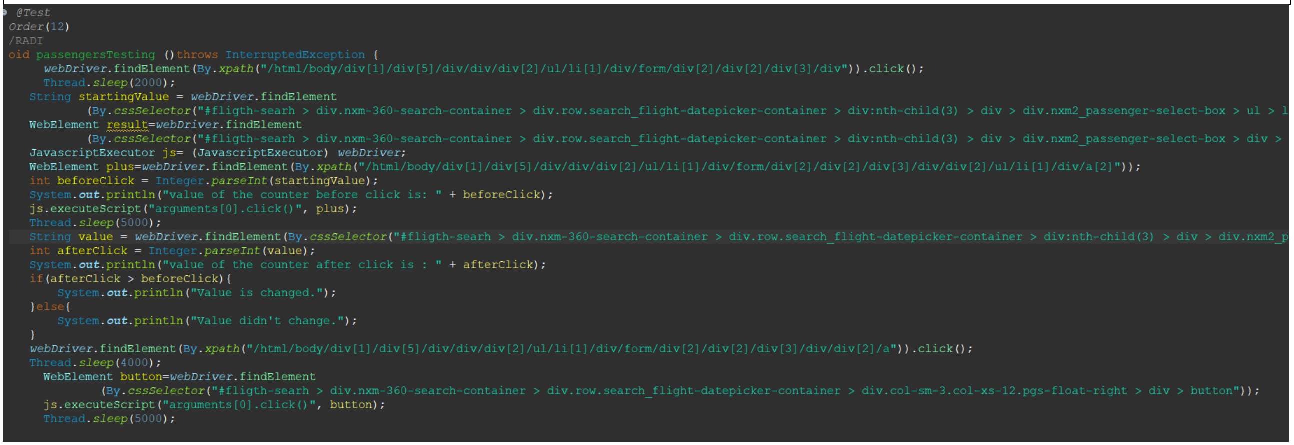


```
@Test
  @Order(11)
  //RADI
  void datePickerTest()throws InterruptedException {
    JavascriptExecutor js= (JavascriptExecutor) webDriver;
    js.executeScript("window.scrollBy(0, 130)");
    WebElement openingDatePicker=webDriver.findElement(By.xpath("/html/body/div[1]/div[5]/div/div/div[2]/ul/li[1]/div/form/div[2]/div[2]/div[1]/div/div[1]/span"));
      js.executeScript("arguments[0].click()", openingDatePicker);
      Thread.sleep(3000);
      WebDriverWait wait=new WebDriverWait(webDriver, Duration.ofSeconds(5));
      WebElement DepartureDatePicker=wait.until(
              ExpectedConditions.visibilityOfElementLocated
              (By.xpath("/html/body/div[1]/div[5]/div/div/div[2]/ul/li[1]/div/form/div[2]/div[2]/div[1]/div/div[2]/div/div[1]")));
      Actions builder=new Actions(webDriver);
      Action choosingDateOfDeparture= builder
              .moveToElement(DepartureDatePicker)
              .moveByOffset(30, 75)
              .click()
              .build();
      choosingDateOfDeparture.perform();
      WebElement ReturningDatePicker=webDriver.findElement(By.xpath("/html/body/div[1]/div[5]/div/div/div[2]/ul/li[1]/div/form/div[2]/div[2]/div[2]/div/div[2]/div/div[2]"));
      Action choosingReturnDate=builder
              .moveToElement(ReturningDatePicker)
              .moveByOffset(70,0)
              .click()
              .build();
      choosingReturnDate.perform();
      Thread.sleep(3000);
}
```

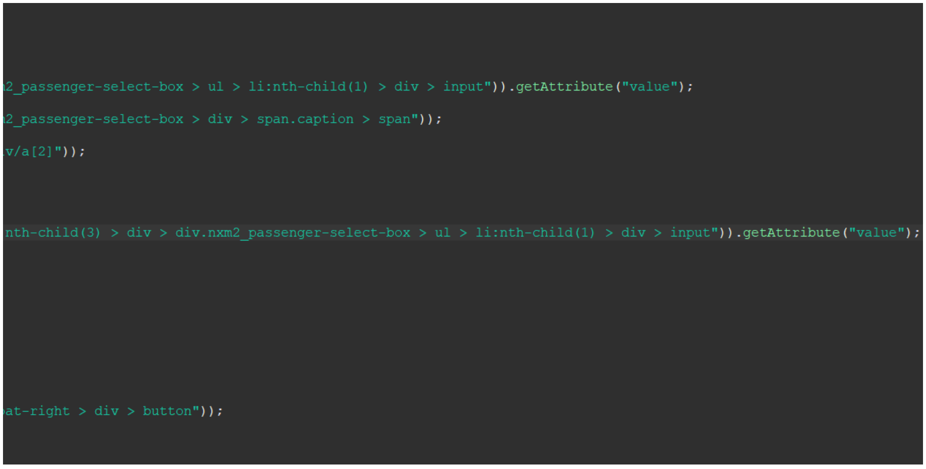
3.12.Passengers Testing

This is part of the Searching for flights. We need to choose how many passengers are going.

Test Name: Date Picker Test				
Description: Check if the date picker works				
Pre-condition(s):				
Test Steps: 1. Clicking on the passengers part 2. Then we need to click on the plus to choose more and more passengers 3. Click on the plus in the Adults part just once 4. Click on the Done button	**Test Data:**	**Expected Result:** There are 2 Adults passengers chosen.	**Actual Result:** There are 2 Adults passengers chosen.	**Status:** PASS
Notes:				



```
 @Test
Order(12)
/RADI
oid passengersTesting ()throws InterruptedException {
     webDriver.findElement(By.xpath("/html/body/div[1]/div[5]/div/div/div[2]/ul/li[1]/div/form/div[2]/div[2]/div[3]/div")).click();
     Thread.sleep(2000);
   String startingValue = webDriver.findElement
           (By.cssSelector("#fligth-searh > div.nxm-360-search-container > div.row.search_flight-datepicker-container > div:nth-child(3) > div > div.nxm2_passenger-select-box > ul > l
   WebElement result=webDriver.findElement
           (By.cssSelector("#fligth-searh > div.nxm-360-search-container > div.row.search_flight-datepicker-container > div:nth-child(3) > div > div.nxm2_passenger-select-box > div >
   JavascriptExecutor js= (JavascriptExecutor) webDriver;
   WebElement plus=webDriver.findElement(By.xpath("/html/body/div[1]/div[5]/div/div/div[2]/ul/li[1]/div/form/div[2]/div[2]/div[3]/div/div[2]/ul/li[1]/div/a[2]"));
   int beforeClick = Integer.parseInt(startingValue);
   System.out.println("value of the counter before click is: " + beforeClick);
   js.executeScript("arguments[0].click()", plus);
   Thread.sleep(5000);
   String value = webDriver.findElement(By.cssSelector("#fligth-searh > div.nxm-360-search-container > div.row.search_flight-datepicker-container > div:nth-child(3) > div > div.nxm2_p
   int afterClick = Integer.parseInt(value);
   System.out.println("value of the counter after click is : " + afterClick);
   if(afterClick > beforeClick){
       System.out.println("Value is changed.");
   }else{
       System.out.println("Value didn't change.");
   }
   webDriver.findElement(By.xpath("/html/body/div[1]/div[5]/div/div/div[2]/ul/li[1]/div/form/div[2]/div[2]/div[3]/div/div[2]/a")).click();
   Thread.sleep(4000);
     WebElement button=webDriver.findElement
             (By.cssSelector("#fligth-searh > div.nxm-360-search-container > div.row.search_flight-datepicker-container > div.col-sm-3.col-xs-12.pgs-float-right > div > button"));
     js.executeScript("arguments[0].click()", button);
     Thread.sleep(5000);
```



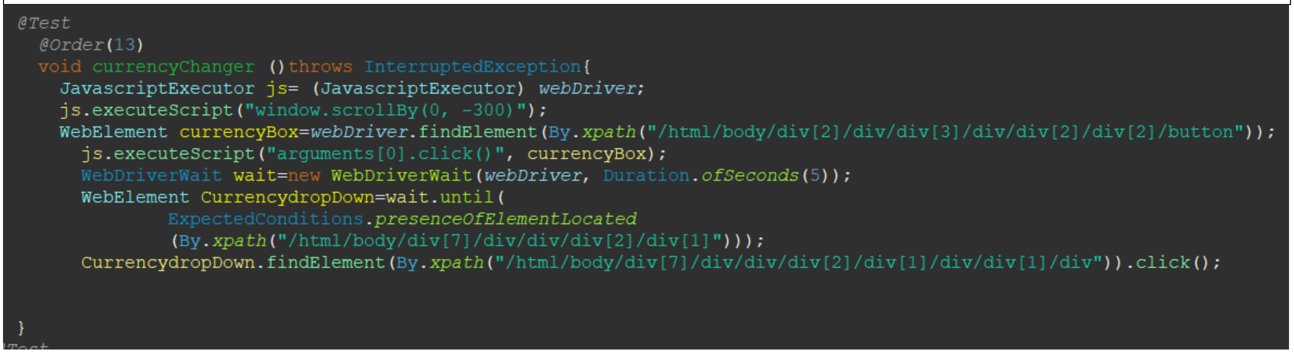
```
2_passenger-select-box > ul > li:nth-child(1) > div > input")).getAttribute("value");

2_passenger-select-box > div > span.caption > span"));

v/a[2]"));




nth-child(3) > div > div.nxm2_passenger-select-box > ul > li:nth-child(1) > div > input")).getAttribute("value");








at-right > div > button"));
```

3.13.Currency changer test

When we search flights in the right corner there is a box that we can use to change currency.

Test Name: Currency changer test
Description: Check if the currency changer works
Pre-condition(s):

Test Steps: 1. Click on the currency box 2. Wait for the dropdown 3. Click on the currency that we want	**Test Data:**	**Expected Result:** The currency should change.	**Actual Result:** For some reason we can’t locate the element and click on the currency to change.	**Status:** FAIL
Notes:				

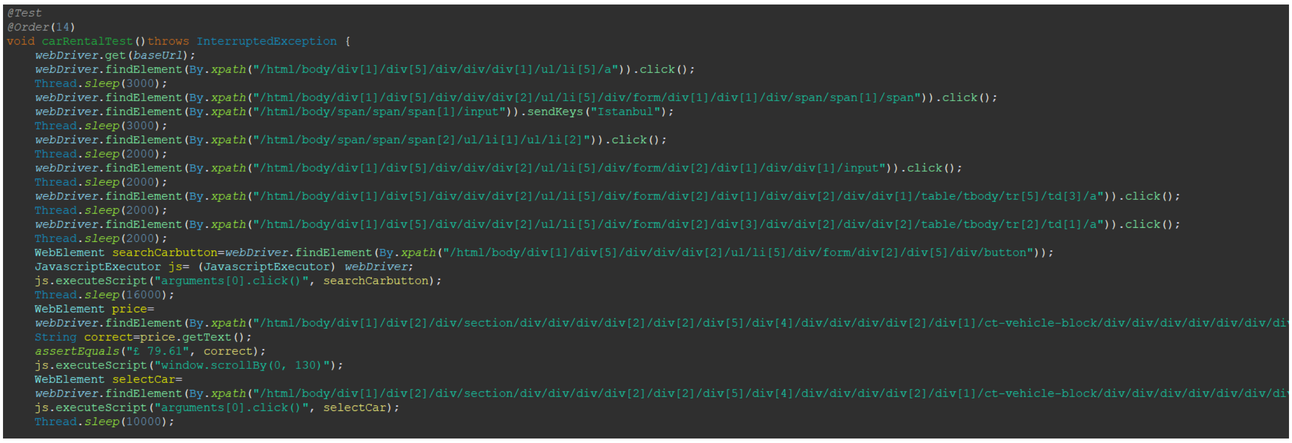


```
@Test
  @Order(13)
  void currencyChanger ()throws InterruptedException{
    JavascriptExecutor js= (JavascriptExecutor) webDriver;
    js.executeScript("window.scrollBy(0, -300)");
    WebElement currencyBox=webDriver.findElement(By.xpath("/html/body/div[2]/div/div[3]/div/div[2]/div[2]/button"));
      js.executeScript("arguments[0].click()", currencyBox);
      WebDriverWait wait=new WebDriverWait(webDriver, Duration.ofSeconds(5));
      WebElement CurrencydropDown=wait.until(
              ExpectedConditions.presenceOfElementLocated
              (By.xpath("/html/body/div[7]/div/div/div[2]/div[1]")));
      CurrencydropDown.findElement(By.xpath("/html/body/div[7]/div/div/div[2]/div[1]/div/div[1]/div")).click();


}
```

3.14.Car rental test

Over the Searching for flights there is a button that leads us to the Car rental page. On the page there should be a list of cars.

Test Name: Car rental test				
Description: Check if the cars are found and selecting one of them				
Pre-condition(s):				
Test Steps: 1. Click on the Car rental button. 2. Send values to the input places. 3. Click on the Search button. 4. Page will be open with the list of car. 5. Test the price of the first car. 6. Select the car.	**Test Data:** Send Istanbul to the city input.	**Expected Result:** The car is selected.	**Actual Result:** The car is selected.	**Status:** PASS
Notes: The test might fail due to the change of the price of car which we can’t be sure of or predict.				

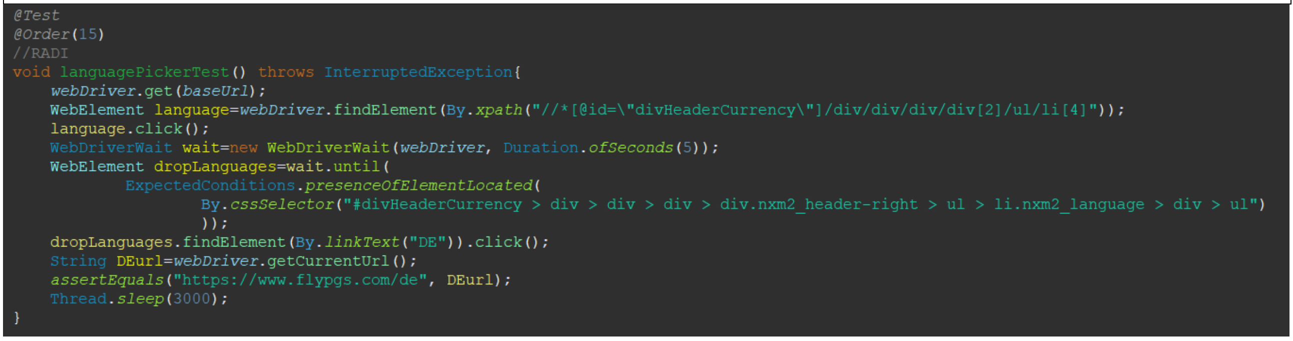


```
@Test
@Order(14)
void carRentalTest()throws InterruptedException {
    webDriver.get(baseUrl);
    webDriver.findElement(By.xpath("/html/body/div[1]/div[5]/div/div/div[1]/ul/li[5]/a")).click();
    Thread.sleep(3000);
    webDriver.findElement(By.xpath("/html/body/div[1]/div[5]/div/div/div[2]/ul/li[5]/div/form/div[1]/div[1]/div/span/span[1]/span")).click();
    webDriver.findElement(By.xpath("/html/body/span/span/span[1]/input")).sendKeys("Istanbul");
    Thread.sleep(3000);
    webDriver.findElement(By.xpath("/html/body/span/span/span[2]/ul/li[1]/ul/li[2]")).click();
    Thread.sleep(2000);
    webDriver.findElement(By.xpath("/html/body/div[1]/div[5]/div/div/div[2]/ul/li[5]/div/form/div[2]/div[1]/div/div[1]/input")).click();
    Thread.sleep(2000);
    webDriver.findElement(By.xpath("/html/body/div[1]/div[5]/div/div/div[2]/ul/li[5]/div/form/div[2]/div[1]/div/div[2]/div/div[1]/table/tbody/tr[5]/td[3]/a")).click();
    Thread.sleep(2000);
    webDriver.findElement(By.xpath("/html/body/div[1]/div[5]/div/div/div[2]/ul/li[5]/div/form/div[2]/div[3]/div/div[2]/div/div[2]/table/tbody/tr[2]/td[1]/a")).click();
    Thread.sleep(2000);
    WebElement searchCarbutton=webDriver.findElement(By.xpath("/html/body/div[1]/div[5]/div/div/div[2]/ul/li[5]/div/form/div[2]/div[5]/div/button"));
    JavascriptExecutor js= (JavascriptExecutor) webDriver;
    js.executeScript("arguments[0].click()", searchCarbutton);
    Thread.sleep(16000);
    WebElement price=
    webDriver.findElement(By.xpath("/html/body/div[1]/div[2]/div/section/div/div/div/div[2]/div[2]/div[5]/div[4]/div/div/div/div[2]/div[1]/ct-vehicle-block/div/div/div/div/div/div/di
    String correct=price.getText();
    assertEquals("£ 79.61", correct);
    js.executeScript("window.scrollBy(0, 130)");
    WebElement selectCar=
    webDriver.findElement(By.xpath("/html/body/div[1]/div[2]/div/section/div/div/div/div[2]/div[2]/div[5]/div[4]/div/div/div/div[2]/div[1]/ct-vehicle-block/div/div/div/div/div/div/di
    js.executeScript("arguments[0].click()", selectCar);
    Thread.sleep(10000);
```

3.15.Changing the language of the page

In the right corner of the page there is a menu for changing the language of the page.

Test Name: Date Picker Test				
Description: Check if the language will be changed.				
Pre-condition(s):				
Test Steps: 1. Click on the part where the languages are 2. Select DE from the menu 3. Assert that the page is on DE language	**Test Data:**	**Expected Result:** Page will be in German language.	**Actual Result:** Page will be in German language.	**Status:** PASS
Notes:				



```
@Test
@Order(15)
//RADI
void languagePickerTest() throws InterruptedException{
    webDriver.get(baseUrl);
    WebElement language=webDriver.findElement(By.xpath("//*[@id=\"divHeaderCurrency\"]/div/div/div[2]/ul/li[4]"));
    language.click();
    WebDriverWait wait=new WebDriverWait(webDriver, Duration.ofSeconds(5));
    WebElement dropLanguages=wait.until(
            ExpectedConditions.presenceOfElementLocated(
                    By.cssSelector("#divHeaderCurrency > div > div > div > div.nxm2_header-right > ul > li.nxm2_language > div > ul")
                    ));
    dropLanguages.findElement(By.linkText("DE")).click();
    String DEurl=webDriver.getCurrentUrl();
    assertEquals("https://www.flypgs.com/de", DEurl);
    Thread.sleep(3000);
}
```

9. Conclusion

9.1. Testing Summary

Testing Tool	Total Tests	Passed Tests	Failed Tests
We used Selenium test environment, together with Java. Some of the Selenium tools that we used are: Selenium IDE, Selenium WebDriver.	15 total tests	13 of passing tests	2 of failing tests: Currency changer test Baggage allowance page

9.2. Final Thoughts

Our final conclusion is that the site is built pretty good. It has some really good functionalities that are implemented in a really good way. It is easy for a user to use them and to get around the page. Major thing that we noticed and that is a very big flaw is the speed of the page. It needs so much time to load certain parts which is why we often had to use waits and Thread.sleep, which is probably a consequence of the next thing that we want to list. We noticed that it was really hard sometimes to get to the correct path/name/selector to use for certain elements. There is so many divisions and links that we had to go through and test clicking on all of them, which was really time consuming. Many of the buttons don't have accessible name and dropdowns menus are made with divisions. But at the end, we are very satisfied with the results and the experience that we gathered testing this page.