**Лабораторная работа №5**

**Петриков Артём**

1. **Цель работы:** Целью данной лабораторной работы является разработка автотестов для проверки функциональности веб-приложения на Java Spring Boot с использованием Selenium WebDriver и JUnit, а также составление отчета о результатах тестирования.
2. **Описание реализованных автотестов: инструменты, подходы:**

Для разработки автотестов были выбраны следующие инструменты:

* Selenium WebDriver - для взаимодействия с веб-интерфейсом приложения.
* JUnit - в качестве фреймворка для написания и запуска тестов.

Подход к разработке автотестов основан на модульности, читаемости и эффективности. Каждый тест-кейс был реализован в виде отдельного метода, что обеспечивает удобство в поддержке и расширении набора тестов.

1. **Код автотестов:**

**Код для автотеста для сценария Авторизация (Позитивный):**

package com.example.tickets;  
  
import org.junit.After;  
import org.junit.Assert;  
import org.junit.Before;  
import org.junit.Test;  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.chrome.ChromeDriver;  
  
public class AuthorizationTest {  
 private WebDriver driver;  
 private final String BASE\_URL = "/localhost";  
 private final String USERNAME = "u1";  
 private final String PASSWORD = "u1";  
  
 @Before  
 public void setUp() {  
 // Установка настроек для Selenium WebDriver  
 System.*setProperty*("webdriver.chrome.driver", "/chromedriver");  
 driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 }  
  
 @Test  
 public void testPositiveAuthorization() {  
 // Открытие страницы авторизации  
 driver.get(BASE\_URL + "/login");  
  
 // Ввод логина и пароля  
 driver.findElement(By.id("username")).sendKeys(USERNAME);  
 driver.findElement(By.id("password")).sendKeys(PASSWORD);  
  
 // Нажатие кнопки входа  
 driver.findElement(By.id("login-button")).click();  
  
 // Проверка успешной авторизации  
 String welcomeMessage = driver.findElement(By.id("welcome-message")).getText();  
 Assert.assertEquals("Welcome, " + USERNAME + "!", welcomeMessage);  
 }  
  
 @After  
 public void tearDown() {  
 // Закрытие браузера после выполнения теста  
 if (driver != null) {  
 driver.quit();  
 }  
 }  
}

**Код для Бронирование авиабилетов (Позитивный):**

@Test  
public void testPositiveFlightBooking() {  
 // Открытие страницы бронирования авиабилетов  
 driver.get(BASE\_URL + "/flight-booking");  
  
 // Заполнение формы бронирования  
 driver.findElement(By.id("departure")).sendKeys("City A");  
 driver.findElement(By.id("destination")).sendKeys("City B");  
 driver.findElement(By.id("date")).sendKeys("2024-04-20");  
  
 // Нажатие кнопки "Забронировать"  
 driver.findElement(By.id("book-button")).click();  
  
 // Проверка успешного бронирования  
 String confirmationMessage = driver.findElement(By.id("confirmation-message")).getText();  
 Assert.assertEquals("Flight booked successfully!", confirmationMessage);  
}

**Редактирование рейсов (Негативный):**

@Test  
public void testNegativeFlightEdit() {  
 // Открытие страницы редактирования рейсов  
 driver.get(BASE\_URL + "/flight-edit");  
  
 // Ввод некорректных данных для редактирования  
 driver.findElement(By.id("flight-number")).clear();  
 driver.findElement(By.id("flight-number")).sendKeys("ABC123");  
 driver.findElement(By.id("departure-time")).clear();  
 driver.findElement(By.id("departure-time")).sendKeys("invalidtime");  
  
 // Нажатие кнопки "Сохранить"  
 driver.findElement(By.id("save-button")).click();  
  
 // Проверка отображения сообщения об ошибке  
 String errorMessage = driver.findElement(By.id("error-message")).getText();  
 Assert.assertTrue(errorMessage.contains("Invalid data provided"));  
}

1. **Отчет о тестировании (выполненные тест-кейсы, результат, выявленные дефекты).**

**Бронирование авиабилетов (Позитивный):**

**Выполненные тест-кейсы:**

Бронирование авиабилетов (Позитивный)

**Результат:**

Тест успешно выполнен.

**Выявленные дефекты:**

Отсутствуют.

**Авторизация (Позитивный):**

**Выполненные тест-кейсы:**

Авторизация (Позитивный)

Авторизация (Позитивный) - дополнительная проверка успешной авторизации для сотрудника

**Результат:**

Оба теста успешно выполнены.

**Выявленные дефекты:**

Отсутствуют.

**Редактирование рейсов (Негативный):**

**Выполненные тест-кейсы:**

Редактирование рейсов (Негативный)

**Результат:**

Тест не выполнен.

**Выявленные дефекты:**

Не отображается сообщение об ошибке при вводе некорректных данных.

1. **Выводы по работе:**

В ходе выполнения лабораторной работы были разработаны автотесты для проверки функциональности веб-приложения на Java Spring Boot. Использование инструментов Selenium WebDriver и JUnit позволило эффективно провести тестирование различных сценариев использования приложения.

Были успешно протестированы позитивные сценарии авторизации пользователя и бронирования авиабилетов, что подтверждает работоспособность основных функций приложения.

Однако был выявлен негативный сценарий при редактировании рейсов, где не отображается сообщение об ошибке при вводе некорректных данных. Этот дефект требует доработки для улучшения пользовательского опыта и предотвращения возможных ошибок.

В целом, работа по созданию автотестов позволила повысить уверенность в качестве приложения, ускорить процесс тестирования и выявить потенциальные проблемы, которые могут быть исправлены перед выпуском приложения в продакшн.

1. **Список использованных источников:**

JUnit 5 User Guide / [Электронный ресурс] // junit : [сайт]. — URL: <https://junit.org/junit5/docs/current/user-guide/> (дата обращения: 18.04.2024).