**Фархутдинов Айдар**

**Цель работы:**

Целью данной лабораторной работы является разработка автотестов для проверки функциональности веб-приложения WorkSwiper с использованием Selenium WebDriver и JUnit, а также составление отчета о результатах тестирования.

**Описание реализованных автотестов: инструменты и 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.

**Код автотестов:**

1. Код для автотеста сценария Авторизация (Позитивный):

package com.example.tests;

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 = "http://localhost";

private final String USERNAME = "employee";

private final String PASSWORD = "password";

@Before

public void setUp() {

System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

driver = new ChromeDriver();

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

}

@Test

public void testPositiveAuthorization() {

driver.get(BASE\_URL + "/login");

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

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

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

String welcomeMessage = driver.findElement(By.id("welcome-message")).getText();

Assert.assertEquals("Welcome " + USERNAME + "!", welcomeMessage);

}

@After

public void tearDown() {

if (driver != null) {

driver.quit();

}

}

}

1. Код для автотеста сценария Создание задач (Позитивный):

@Test

public void testPositiveTaskCreation() {

driver.get(BASE\_URL + "/create-task");

driver.findElement(By.id("task-title")).sendKeys("New Task");

driver.findElement(By.id("task-description")).sendKeys("Task description");

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

String confirmationMessage = driver.findElement(By.id("confirmation-message")).getText();

Assert.assertEquals("Task created successfully!", confirmationMessage);

}

1. Код для автотеста сценария Выбор задач (Негативный):

@Test

public void testNegativeTaskSelection() {

driver.get(BASE\_URL + "/task-selection");

driver.findElement(By.id("select-all")).click();

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

String errorMessage = driver.findElement(By.id("error-message")).getText();

Assert.assertTrue(errorMessage.contains("No more tasks available"));

}

**Отчет по тестированию:**

**Выполненные тест-кейсы и результаты:**

1. **Авторизация (Позитивный)**
   * **Результат**: Тест успешно выполнен.
   * **Выявленные дефекты**: Отсутствуют.
2. **Создание задач (Позитивный)**
   * **Результат**: Тест успешно выполнен.
   * **Выявленные дефекты**: Отсутствуют.
3. **Выбор задач (Негативный)**
   * **Результат**: Тест выполнен, выявлена ошибка.
   * **Выявленные дефекты**: Появление пустых карточек после выбора всех задач.

**Список выявленных дефектов:**

1. **Дефект 1**:
   * **Краткое описание**: Появление пустых карточек после выбора всех задач.
   * **Серьезность**: Значительная (major)
   * **Приоритет**: Средний

**Выводы по работе:**

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

Позитивные тесты подтвердили работоспособность основных функций приложения, таких как авторизация и создание задач. Негативный тест выявил дефект при выборе задач, где после выбора всех доступных задач появляются пустые карточки. Этот дефект требует доработки для улучшения пользовательского опыта и предотвращения возможных ошибок.

Работа по созданию автотестов позволила повысить уверенность в качестве приложения, ускорить процесс тестирования и выявить потенциальные проблемы до выпуска приложения в продакшн.

**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).