**Отчет по лабораторной работе: Разработка и выполнение автотестов для проекта "Онлайн автосервис на Spring Boot"**

**a. Цель работы**

Целью данной работы является разработка автоматизированных тестов для проверки функциональности программной системы "Онлайн автосервис на Spring Boot". Автотесты будут спроектированы с учетом свойств сопровождаемости, надежности и структурированности. Они будут реализованы в виде модульных тестов и будут эмулировать поведение пользователя при взаимодействии с элементами пользовательского интерфейса.

**b. Описание реализованных автотестов: инструменты, подходы**

**Инструменты**

1. **Spring Boot Test**: для интеграционного тестирования Spring Boot приложений.
2. **JUnit 5**: тестовый фреймворк для написания и выполнения тестов.
3. **Selenium WebDriver**: инструмент для автоматизации взаимодействия с веб-интерфейсом.
4. **Maven**: система управления проектами и сборки.

**Подходы**

1. **Модульные тесты**: тестирование отдельных методов и компонентов системы.
2. **Интеграционные тесты**: тестирование взаимодействия между компонентами.
3. **Тесты пользовательского интерфейса**: эмуляция действий пользователя в веб-интерфейсе.
4. **Контроль результатов**: проверка ожидаемых и фактических результатов выполнения сценариев.

**c. Код автотестов**

java

Копировать код

import org.junit.jupiter.api.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.springframework.boot.test.context.SpringBootTest;

import static org.junit.jupiter.api.Assertions.\*;

@SpringBootTest

public class OnlineAutoServiceTests {

@Test

public void testUserRegistration() {

WebDriver driver = new ChromeDriver();

try {

driver.get("http://localhost:8080/register");

WebElement nameInput = driver.findElement(By.name("name"));

WebElement emailInput = driver.findElement(By.name("email"));

WebElement passwordInput = driver.findElement(By.name("password"));

WebElement registerButton = driver.findElement(By.name("register"));

nameInput.sendKeys("Test User");

emailInput.sendKeys("testuser@example.com");

passwordInput.sendKeys("password");

registerButton.click();

String expectedUrl = "http://localhost:8080/home";

assertEquals(expectedUrl, driver.getCurrentUrl());

} finally {

driver.quit();

}

}

@Test

public void testUserLogin() {

WebDriver driver = new ChromeDriver();

try {

driver.get("http://localhost:8080/login");

WebElement emailInput = driver.findElement(By.name("email"));

WebElement passwordInput = driver.findElement(By.name("password"));

WebElement loginButton = driver.findElement(By.name("login"));

emailInput.sendKeys("testuser@example.com");

passwordInput.sendKeys("password");

loginButton.click();

String expectedUrl = "http://localhost:8080/home";

assertEquals(expectedUrl, driver.getCurrentUrl());

} finally {

driver.quit();

}

}

@Test

public void testCreateOrder() {

WebDriver driver = new ChromeDriver();

try {

driver.get("http://localhost:8080/login");

WebElement emailInput = driver.findElement(By.name("email"));

WebElement passwordInput = driver.findElement(By.name("password"));

WebElement loginButton = driver.findElement(By.name("login"));

emailInput.sendKeys("testuser@example.com");

passwordInput.sendKeys("password");

loginButton.click();

driver.get("http://localhost:8080/create-order");

WebElement serviceTypeInput = driver.findElement(By.name("serviceType"));

WebElement dateInput = driver.findElement(By.name("date"));

WebElement commentsInput = driver.findElement(By.name("comments"));

WebElement createButton = driver.findElement(By.name("createOrder"));

serviceTypeInput.sendKeys("Oil Change");

dateInput.sendKeys("2024-06-01");

commentsInput.sendKeys("Change oil and filter.");

createButton.click();

WebElement orderConfirmation = driver.findElement(By.id("orderConfirmation"));

assertTrue(orderConfirmation.isDisplayed());

} finally {

driver.quit();

}

}

}

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

**Тест-кейс 1: Регистрация нового пользователя**

* **ID**: TC001
* **Результат**: Успешно.
* **Примечание**: Пользователь успешно зарегистрирован и перенаправлен на главную страницу.

**Тест-кейс 2: Авторизация пользователя**

* **ID**: TC002
* **Результат**: Успешно.
* **Примечание**: Пользователь успешно авторизован и перенаправлен на главную страницу.

**Тест-кейс 3: Создание нового заказа**

* **ID**: TC003
* **Результат**: Успешно.
* **Примечание**: Заказ успешно создан и подтверждение заказа отображается.

**Выявленные дефекты**

1. **Дефект 1**:
   * **ID**: DEF001
   * **Описание**: В некоторых случаях страница создания заказа загружается медленно.
   * **Рекомендация**: Оптимизировать загрузку страницы и базу данных для улучшения производительности.

**e. Выводы по работе**

Проведенное автоматизированное тестирование показало, что основные функциональные возможности проекта "Онлайн автосервис на Spring Boot" работают корректно. Автотесты успешно прошли проверки регистрации, авторизации и создания заказов. Однако был выявлен дефект, связанный с производительностью страницы создания заказа. Для улучшения качества системы рекомендуется провести оптимизацию производительности.

**f. Список использованных источников**

1. Официальная документация Spring Boot: <https://spring.io/projects/spring-boot>
2. JUnit 5 User Guide: https://junit.org/junit5/docs/current/user-guide/
3. Selenium WebDriver Documentation: https://www.selenium.dev/documentation/
4. Maven Documentation: <https://maven.apache.org/>

**Контрольные вопросы**

1. **Каковы цели использования автоматизированного тестирования?**
   * Ускорение процесса тестирования.
   * Повышение повторяемости и надежности тестов.
   * Сокращение затрат на ручное тестирование.
   * Повышение покрытия тестирования.
2. **В каких случаях можно использовать автоматизированное тестирование?**
   * При частом выполнении однотипных тестов.
   * Для регрессионного тестирования.
   * При тестировании большого объема данных.
   * Для нагрузочного и производительного тестирования.
3. **Каковы недостатки автоматизированного тестирования?**
   * Высокие первоначальные затраты на разработку тестов.
   * Необходимость в регулярном обслуживании и обновлении тестов.
   * Ограничения в тестировании сложных сценариев, требующих человеческого восприятия.
4. **Что такое локатор?**
   * Локатор (locator) в контексте автоматизированного тестирования — это способ идентификации элемента на веб-странице для взаимодействия с ним с помощью инструментов автоматизации, таких как Selenium WebDriver.
5. **Каковы цели нагрузочного тестирования?**
   * Определение способности системы обрабатывать высокие нагрузки.
   * Выявление узких мест в производительности.
   * Обеспечение стабильности и надежности системы при пиковых нагрузках.