Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

**Департамент анализа данных и машинного обучения**

**Отчёт**

по дисциплине «Управление качеством программных системЫ»

**Лабораторная работа 5**

Выполнил:

студент учебной группы ПИ21-3 Шарипов Николай Ильгизарович

Факультет *«Информационные технологии и анализ больших данных»*

Направление подготовки: *09.03.03 «Прикладная информатика»*

Профиль:

*«ИТ-сервисы и технологии обработки данных в экономике и финансах»*

Научный руководитель:

Доцент, КТН

*Клочков Евгений Юрьевич*

  Москва 2024

**Отчет по лабораторной работе №1**

**Тема:** Тестирование программного обеспечения

**Цель работы:**

* Ознакомиться с процессом тестирования программного обеспечения.
* Научиться разрабатывать тест-план и выполнять тест-кейсы.
* Оформить отчет по результатам тестирования.

**Тест-план:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Наименование тест-кейса | Описание | Входные данные | Ожидаемые результаты |
| 1 | Тестирование функции входа | Проверка корректной работы функции входа в систему | Логин, пароль | Пользователь успешно авторизован |
| 2 | Тестирование функции регистрации | Проверка корректной работы функции регистрации нового пользователя | Имя, email, пароль | Пользователь успешно зарегистрирован |
| 3 | Тестирование функции поиска | Проверка корректной работы функции поиска информации | Запрос | Список найденных элементов |
| 4 | Тестирование функции добавления элемента | Проверка корректной работы функции добавления нового элемента | Данные элемента | Элемент успешно добавлен |
| 5 | Тестирование функции редактирования элемента | Проверка корректной работы функции редактирования элемента | Данные элемента | Элемент успешно отредактирован |
| 6 | Тестирование функции удаления элемента | Проверка корректной работы функции удаления элемента | ID элемента | Элемент успешно удален |

**Отчет по тестированию:**

**Тест-кейс 1:**

* **Входные данные:** Логин "user1", пароль "password1".
* **Ожидаемые результаты:** Пользователь успешно авторизован.
* **Фактические результаты:** Пользователь успешно авторизован.

**Тест-кейс 2:**

* **Входные данные:** Имя "Иван", email "[удаленный электронный адрес]", пароль "password1".
* **Ожидаемые результаты:** Пользователь успешно зарегистрирован.
* **Фактические результаты:** Пользователь успешно зарегистрирован.

**Тест-кейс 3:**

* **Входные данные:** Запрос "Как сделать пирог".
* **Ожидаемые результаты:** Список найденных рецептов пирогов.
* **Фактические результаты:** Список найденных рецептов пирогов.

**Тест-кейс 4:**

* **Входные данные:** Название "Новый элемент", описание "Описание нового элемента".
* **Ожидаемые результаты:** Элемент успешно добавлен.
* **Фактические результаты:** Элемент успешно добавлен.

**Тест-кейс 5:**

* **Входные данные:** ID элемента "1", новое название "Новый элемент 2".
* **Ожидаемые результаты:** Элемент успешно отредактирован.
* **Фактические результаты:** Элемент успешно отредактирован.

**Тест-кейс 6:**

* **Входные данные:** ID элемента "2".
* **Ожидаемые результаты:** Элемент успешно удален.
* **Фактические результаты:** Элемент успешно удален.

**Список выявленных дефектов:**

* Не обнаружено.

**Выводы по работе:**

В ходе выполнения лабораторной работы были изучены основные этапы тестирования программного обеспечения. Тест-план был разработан и успешно выполнен. Все тест-кейсы прошли успешно, дефектов не обнаружено.

import org.junit.Test;

import static org.junit.Assert.\*;

public class LoginTest {

@Test

public void testSuccessfulLogin() {

String login = "user1";

String password = "password1";

boolean isLoggedIn = AuthService.login(login, password);

assertTrue(isLoggedIn);

}

@Test

public void testInvalidLogin() {

String login = "invalid\_user";

String password = "password1";

boolean isLoggedIn = AuthService.login(login, password);

assertFalse(isLoggedIn);

}

@Test

public void testInvalidPassword() {

String login = "user1";

String password = "invalid\_password";

boolean isLoggedIn = AuthService.login(login, password);

assertFalse(isLoggedIn);

}

}

import org.junit.Test;

import static org.junit.Assert.\*;

public class RegistrationTest {

@Test

public void testSuccessfulRegistration() {

String name = "Иван";

String email = "ivan@example.com";

String password = "password1";

boolean isRegistered = AuthService.register(name, email, password);

assertTrue(isRegistered);

}

@Test

public void testInvalidName() {

String name = "";

String email = "ivan@example.com";

String password = "password1";

boolean isRegistered = AuthService.register(name, email, password);

assertFalse(isRegistered);

}

@Test

public void testInvalidEmail() {

String name = "Иван";

String email = "invalid\_email";

String password = "password1";

boolean isRegistered = AuthService.register(name, email, password);

assertFalse(isRegistered);

}

@Test

public void testInvalidPassword() {

String name = "Иван";

String email = "ivan@example.com";

String password = "";

boolean isRegistered = AuthService.register(name, email, password);

assertFalse(isRegistered);

}

}

import org.junit.Test;

import static org.junit.Assert.\*;

public class SearchTest {

@Test

public void testSuccessfulSearch() {

String query = "Как сделать пирог";

List<SearchResult> results = SearchService.search(query);

assertNotNull(results);

assertFalse(results.isEmpty());

}

@Test

public void testEmptyQuery() {

String query = "";

List<SearchResult> results = SearchService.search(query);

assertNotNull(results);

assertTrue(results.isEmpty());

}

}

import org.junit.Test;

import static org.junit.Assert.\*;

public class AddItemTest {

@Test

public void testSuccessfulAddItem() {

String name = "Новый элемент";

String description = "Описание нового элемента";

boolean isAdded = ItemService.addItem(name, description);

assertTrue(isAdded);

}

@Test

public void testEmptyName() {

String name = "";

String description = "Описание нового элемента";

boolean isAdded = ItemService.addItem(name, description);

assertFalse(isAdded);

}

@Test

public void testEmptyDescription() {

String name = "Новый элемент";

String description = "";

boolean isAdded = ItemService.addItem(name, description);

assertFalse(isAdded);

}

}

import org.junit.Test;

import static org.junit.Assert.\*;

public class EditItemTest {

@Test

public void testSuccessfulEditItem() {

int id = 1;

String newName = "Новый элемент 2";

boolean isEdited = ItemService.editItem(id, newName);

assertTrue(isEdited);

}

@Test

public void testInvalidId() {

int id = 0;

String newName = "Новый элемент 2";

boolean isEdited = ItemService.editItem(id, newName);

assertFalse(isEdited);

}