МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Казанский национальный исследовательский технический университет

им. А.Н. Туполева – КАИ»

Институт компьютерных технологий и защиты информации

Отделение СПО ИКТЗИ (Колледж информационных технологий)

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине

Поддержка и тестирование программных модулей

Работу выполнил

Студент гр.4335

Желваков А.С.

Принял

Преподаватель

Максимов Р. С.

Казань 2025

**Задание на лабораторную работу**

1. К уже разработанной ранее программе в 4 лабораторной работе.
2. Выполнить системное тестирование.
3. Пройти все этапы системного тестирования.
4. Оформить отчёт.

**Результат выполнения работы**

Тест-План

* Заголовок / Версия / Автор

Заголовок: Тест-план для приложения 4 лабораторной работы по поддержке и тестированию программных модулей.

Версия: 1.0

Автор: Павлова Анастасия 4335

* Техническое задание на продукт или иная документация:
* Приложение, разработанное в 4 лабораторной работе по поддержке и тестированию программных модулей предназначено для просмотра и администрирования товаров.
* Задачи / функциональность, которая должна быть протестирована:

1. Авторизация в приложении
2. Вывод товаров
3. Администрирование товаров
4. Пользовательский интерфейс

* Виды проводимого тестирования:

1. Авторизация в приложении:

* Авторизация пользователя
* Авторизация администратора

1. Вывод товаров:

* Вывод товаров на странице администратора
* Вывод товаров на странице пользователя

1. Администрирование товаров:

* Проверка добавления товара
* Проверка изменения товара
* Проверка удаления товара

1. Пользовательский интерфейс:

* Проверка доступности кнопки выхода
* Проверка интерфейса
* Список тестовой документации (тест-кейсы, тест-сьюты)

1. Тест-сьюты

Таблица 1

| ID | Автор | Приор. | Заголовок | Список тест-кейсов |
| --- | --- | --- | --- | --- |
| 1 | user | 1 | Модуль авторизации | * 1. Проверка авторизации пользователя   2. Проверка авторизации администратора |
| 2 | user | 1 | Модуль вывода товаров | * 1. Проверка вывода товаров на странице пользователя   2. Проверка вывода товаров на странице администратора |
| 3 | user | 1 | Модуль изменения товаров | * 1. Проверка добавления товара   2. Проверка обновления товара   3. Проверка удаления товара |
| 4 | user | 2 | Интерфейс приложения | * 1. Проверка доступности кнопки выхода   2. Проверка интерфейса |

1. Тест-кейсы

Таблица 2

| ID | Описание (Тип) | Предусловия | Шаги | Ожидаемый результат |
| --- | --- | --- | --- | --- |
| 1.1 | Проверка авторизации пользователя (позитивный) | 1. Пользователь находится на странице авторизации | 1. Ввести в поля «Email» и «Пароль» электронную почту и пароль пользователя соответственно | 1.Пользователь успешно переходит на страницу пользователя |
| 1.2 | Проверка авторизации администратора (позитивный) | 1. Пользователь находится на странице авторизации | 1. Ввести в поля «Email» и «Пароль» электронную почту и пароль администратора соответственно | 1. Пользователь успешно переходит на страницу администратора |
| 2.1 | Проверка вывода товаров на странице пользователя (позитивный) | 1. Пользователь находится на странице авторизации | 1. Войти на страницу пользователя | 1. Все товары отображаются |
| 2.2 | Проверка вывода товаров на странице администратора (позитивный) | 1. Пользователь находится на странице авторизации | 1. Войти на страницу администратора | 1. Товары корректно отображаются в таблице |

Продолжение таблицы 2

| ID | Описание (Тип) | Предусловия | Шаги | Ожидаемый результат |
| --- | --- | --- | --- | --- |
| 3.1 | Проверка добавления товара (позитивный) | 1. Пользователь находится на странице администратора | 1. Нажать на кнопку «Добавить» | 1. Пользователь переходит на страницу добавления, валидация не позволяет ввести некорректные данные |
| 3.2 | Проверка обновления товара (позитивный) | 1. Пользователь находится на странице администратора | 1. Нажать на кнопку «Изменить» любого товара | 1. Пользователь переходит на страницу изменения товара, данные уже загружены в поля, валидация не позволяет ввести некорректные данные |
| 3.3 | Проверка удаления товара (позитивный) | 1. Пользователь находится на странице администратора | 1. Выделить один или несколько товаров 2. Нажать на кнопку «Удалить» | 1. Страница обновляется |
| 4.1 | Проверка доступности кнопки выхода (позитивный) | 1. Пользователь находится на странице авторизации | 1. Нажать на кнопку «Выход» | 1. Пользователь вышел из системы |
| 4.2 | Проверка интерфейса (негативный) | 1. Пользователь открывает окно приложения | 1. Пользователь авторизовался и начинает пользоваться системой | 1. Окна имеют привлекательный и красочный вид |

* Список инструментов тестирования:
* В ходе тестирования не было задействовано ни одного инструмента.
* Сервер (или иное расположение тестируемой программной системы)
* Локальное хранилище
* Ответственные лица (ФИО/Должность/Занятие):
* ФИО: Павлова Анастасия Анатольевна
* Должность: Студент
* Занятие: Тестировщик
* График тестирования:
* Начало: 06.02.2025 9:00
* Конец: 06.02.2025 11:20
* Оценка риска:
* Следует тщательнее продумать интерфейс приложения для более корректного и привлекательного отображения и его использования
* Примечание:
* Отсутствует

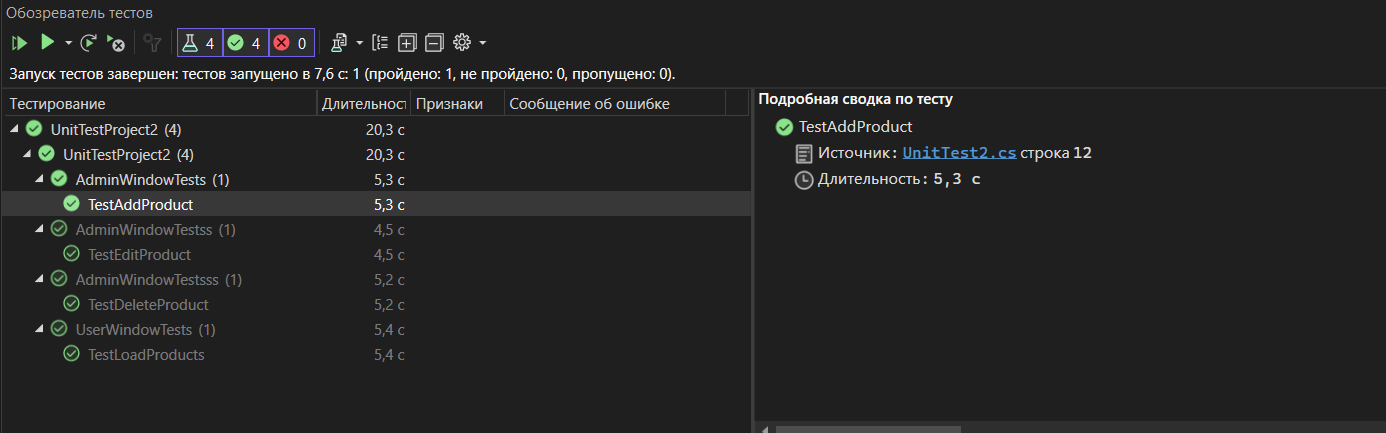


Рисунок 1 – Результат выполнения тестирования

*Листинг тестов*

UserWindowTests:

using Microsoft.EntityFrameworkCore;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using WpfApp1;

namespace UnitTestProject2

{

[TestClass]

public class UserWindowTests

{

[TestMethod]

public void TestLoadProducts()

{

// Arrange

string productName1 = "Product 33";

decimal productCost1 = 10.00m;

string imagePath1 = "image121.jpg";

long productId1;

// Act

using (var context = new XRENOTAEntities())

{

var product1 = new Products { Name = productName1, Cost = productCost1, Image\_path = imagePath1 };

context.Products.Add(product1);

context.SaveChanges();

productId1 = product1.Product\_id;

}

// Assert

using (var context = new XRENOTAEntities())

{

var productInDb1 = context.Products.Find(productId1);

Assert.IsNotNull(productInDb1, $"Product not found");

// 6. Дополнительно можно проверить свойства продуктов

Assert.AreEqual(productName1, productInDb1.Name, $"Incorrect name");

Assert.AreEqual(productCost1, productInDb1.Cost, $"Incorrect cost ");

Assert.AreEqual(imagePath1, productInDb1.Image\_path, $"Incorrect image path ");

}

}

}

}

AdminWindowTests:

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using System.Linq;

using WpfApp1;

namespace UnitTestProject2

{

[TestClass]

public class AdminWindowTests

{

[TestMethod]

public void TestAddProduct()

{

using (var context = new XRENOTAEntities())

{

var newProduct = new Products { Name = "Product 77", Cost = 100, Image\_path = "image177.jpg" };

context.Products.Add(newProduct);

context.SaveChanges();

}

using (var context = new XRENOTAEntities())

{

var productInDb = context.Products.FirstOrDefault(p => p.Name == "Product 77");

Assert.IsNotNull(productInDb, "Product was not added to the database");

Assert.AreEqual(100, productInDb.Cost, "Product cost is incorrect");

Assert.AreEqual("image177.jpg", productInDb.Image\_path, "Image path is incorrect");

}

}

}

}

AdminWindowTestss:

using Microsoft.EntityFrameworkCore;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using WpfApp1;

namespace UnitTestProject2

{

[TestClass]

public class AdminWindowTestss

{

[TestMethod]

public void TestEditProduct()

{

using (var context = new XRENOTAEntities())

{

var product = new Products { Name = "Product 66", Cost = 100, Image\_path = "image123.jpg" };

context.Products.Add(product);

context.SaveChanges();

var productToUpdate = context.Products.Find(product.Product\_id);

productToUpdate.Name = "Updated Product";

productToUpdate.Cost = 200;

context.SaveChanges();

var updatedProduct = context.Products.Find(product.Product\_id);

Assert.AreEqual("Updated Product", updatedProduct.Name);

Assert.AreEqual(200, updatedProduct.Cost);

}

}

}

}

AdminWindowTestsss

using Microsoft.EntityFrameworkCore;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using WpfApp1;

namespace UnitTestProject2

{

[TestClass]

public class AdminWindowTestsss

{

[TestMethod]

public void TestDeleteProduct()

{

using (var context = new XRENOTAEntities())

{

var product = new Products { Name = "Product 2", Cost = 100, Image\_path = "image222.jpg" };

context.Products.Add(product);

context.SaveChanges();

context.Products.Remove(product);

context.SaveChanges();

var deletedProduct = context.Products.Find(product.Product\_id);

Assert.IsNull(deletedProduct);

}

}

}

}