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)

ЛАБОРАТОРНЕАЯ РАБОТА №4

по дисциплине

«Поддержка и тестирование программных модулей»

Работу выполнил

Студент группы 4335

Желваков А.С.

Принял

Преподаватель Максимов Р.С.

Казань 2025

**Задание на лабораторную работу**

1. К уже разработанной ранее программе (с регистрацией и авторизацией) и подключением к БД.
2. Разработать новое окно:

* Для пользователя, окно вывода товаров (динамический вывод из БД).
* Для администратора, форму работы с товарами в БД.
* (С возможностью изменения, добавления и удаления товара)

1. Покрыть Unit - тестами "Вывод товаров", "Добавление нового товара", "Изменение ранее добавленного товара", "Удаление товара".
2. Определить степень покрытия кода.
3. Оформить отчёт.

**Результат выполнения работы**

На рисунке 1 представлено стартовое окно приложения (Страница авторизации).

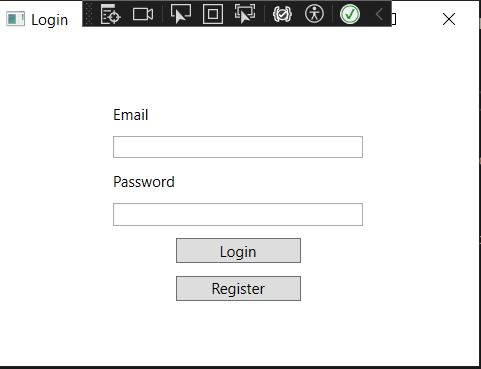


Рисунок 1 – Страница авторизации

На рисунке 2 представлена страница пользователя.

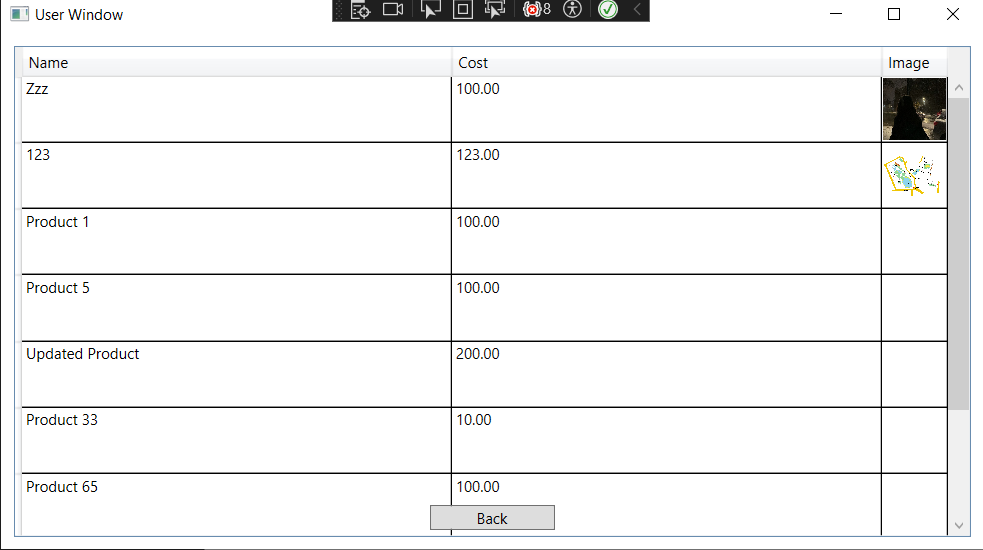


Рисунок 2 – Страница пользователя

На рисунке 3 представлена страница администратора.

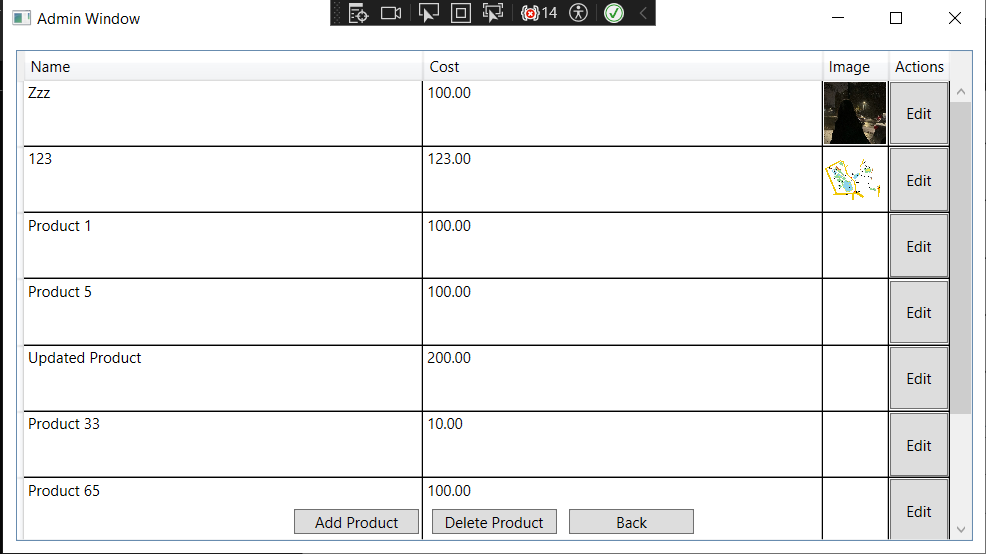


Рисунок 3 – Страница администратора

На рисунке 4 представлена страница изменения и добавления товара.

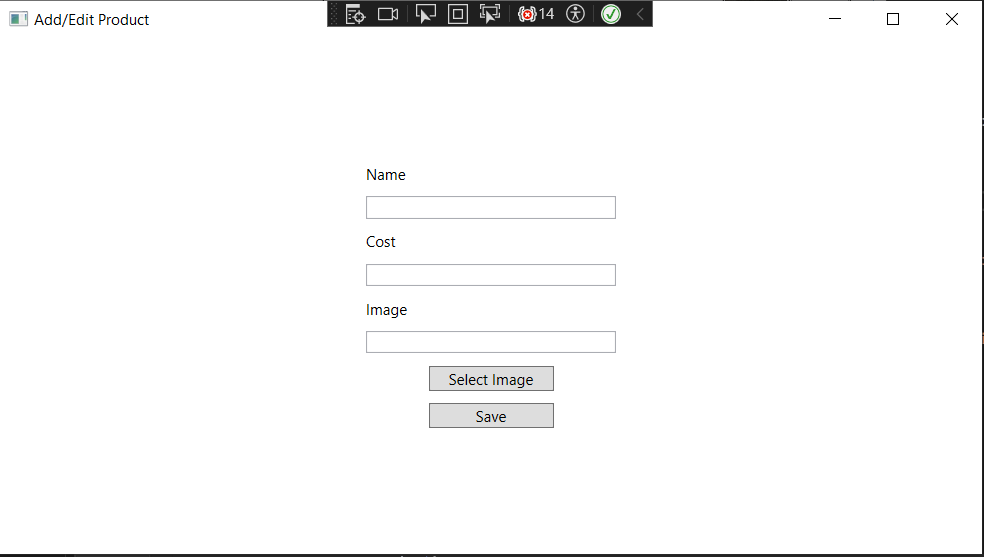


Рисунок 4 – Страница изменения и добавления товара

На рисунке 5 представлено выполнение модульного тестирования методов обращения с товарами.

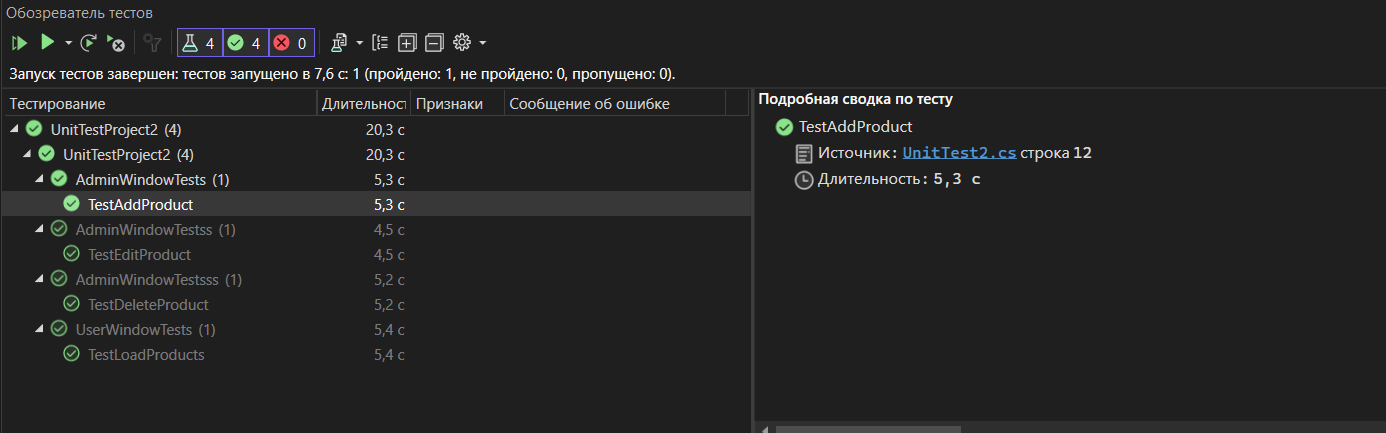


Рисунок 5 – Результат выполнения тестирования

*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);

}

}

}

}