Министерство цифрового развития государственного управления, информационных технологий и связи Республики Татарстан

Государственное автономное профессиональное образовательное учреждение «Международный центр компетенций – Казанский техникум информационных технологий и связи»

Курсовой проект выполнен

и защищена с оценкой

Руководитель КП

« » 20 г

КУРСОВОЙ ПРОЕКТ

по МДК 01.01 Разработка программных модулей на тему:

Разработка standalone приложения для магазина перегонных аппаратов.

Руководитель И.И. Долгова

подпись, дата

Исполнитель,

студентка группы 321 Г.Ф. Гарифинова

ИСП подпись, дата

Казань, 2022

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc104541428)

[1 ОБОСНОВАНИЕ ВЫБОРА СРЕДЫ РАЗРАБОТКИ 4](#_Toc104541429)

[1.1 Язык программирования 4](#_Toc104541430)

[1.2 Платформа WPF 4](#_Toc104541431)

[1.3 База данных MongoDB 6](#_Toc104541432)

[1.4 Необходимые драйвера для MongoDB на C# 6](#_Toc104541433)

[2 РЕАЛИЗАЦИЯ ПРОЕКТА 7](#_Toc104541434)

[2.1 Подключение MonboDB 7](#_Toc104541435)

[2.2 Создание базы данных 8](#_Toc104541436)

[2.3 Создание приложения 9](#_Toc104541437)

[2.4 Работа за администратора 14](#_Toc104541438)

[ЗАКЛЮЧЕНИЕ 17](#_Toc104541439)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 19](#_Toc104541440)

[ПРИЛОЖЕНИЕ A 20](#_Toc104541441)

[ПРИЛОЖЕНИЕ B 22](#_Toc104541442)

[ПРИЛОЖЕНИЕ C 37](#_Toc104541443)

# ВВЕДЕНИЕ

Программный продукт «Магазин самогонных аппаратов», представляет собой приложение, предназначенное для просмотра и заказа товаров. В приложении имеется более 10 товаров с описанием доступных для заказа. Так же в приложении реализованы другие функции для больших возможностей, например:

* Регистрация пользователя;
* Добавление товара;
* Удаление товара;
* Удаление заказа;

При разработке данного продукта использовалась база данных MongoDB.

Для создания приложения была выбрана среда программирования Microsoft Visual Studio 2022, в которой, с помощью платформы пользовательского интерфейса WPF и было сделано приложение.

Для реализации программного продукта были определены и выполнены следующие задачи:

* изучение предметной области;
* разработка интерфейса;
* разработка структуры базы данных.

# 1 ОБОСНОВАНИЕ ВЫБОРА СРЕДЫ РАЗРАБОТКИ

1.1 Язык программирования

Для написания программы была выбрана среда программирования Visual Studio 2022 и язык программирования C#.

Был выбран данный язык программирования, потому что он основан на платформе .NET, которая позволяет запустить приложение на различных платформах. Также C# был выбран из-за ряда преимуществ:

* подлинная объектная ориентированность (всякая языковая сущность претендует на то, чтобы быть объектом);
* компонентно-ориентированное программирование;
* безопасный (по сравнению с языками C и C++) код;
* унифицированная система типизации;
* поддержка событийно-ориентированного программирования;
* «родной» язык для создания приложений в среде .NET;
* объединение лучших идей современных языков программирования: Java, C++, Visual Basic и др.

## 1.2 Платформа WPF

Windows Presentation Foundation (WPF) — аналог WinForms, система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML.

Технология WPF (Windows Presentation Foundation) является часть экосистемы платформы .NET и представляет собой подсистему для построения графических интерфейсов.

Если при создании традиционных приложений на основе WinForms за отрисовку элементов управления и графики отвечали такие части ОС Windows, как User32 и GDI+, то приложения WPF основаны на DirectX.

В этом состоит ключевая особенность рендеринга графики в WPF: используя WPF, значительная часть работы по отрисовке графики, как простейших кнопочек, так и сложных 3D-моделей, ложиться на графический процессор на видеокарте, что также позволяет воспользоваться аппаратным ускорением графики.

Преимущества WPF:

* Использование традиционных языков .NET-платформы - C# и VB.NET для создания логики приложения.
* Возможность декларативного определения графического интерфейса с помощью специального языка разметки XAML, основанном на xml и представляющем альтернативу программному созданию графики и элементов управления, а также возможность комбинировать XAML и C#/VB.NET.
* Независимость от разрешения экрана: поскольку в WPF все элементы измеряются в независимых от устройства единицах, приложения на WPF легко масштабируются под разные экраны с разным разрешением.
* Новые возможности, которых сложно было достичь в WinForms, например, создание трехмерных моделей, привязка данных, использование таких элементов, как стили, шаблоны, темы и др.
* Хорошее взаимодействие с WinForms, благодаря чему, например, в приложениях WPF можно использовать традиционные элементы управления из WinForms.
* Богатые возможности по созданию различных приложений: это и мультимедиа, и двухмерная и трехмерная графика, и богатый набор встроенных элементов управления, а также возможность самим создавать новые элементы, создание анимаций, привязка данных, стили, шаблоны, темы и многое другое.
* Аппаратное ускорение графики - вне зависимости от того, работаете ли вы с 2D или 3D, графикой или текстом, все компоненты приложения транслируются в объекты, понятные Direct3D, и затем визуализируются с помощью процессора на видеокарте, что повышает производительность, делает графику более плавной.
* Создание приложений под множество ОС семейства Windows - от Windows XP до Windows 10.

Чтобы создать WPF проект, нужно зайти в Visual Studio и в поисковике найти WPF (NET.Framework) или (NET.Core), после нажатия открывается новый проект.

## 1.3 База данных MongoDB

MongoDB — документоориентированная система управления базами данных, не требующая описания схемы таблиц. Считается одним из классических примеров NoSQL-систем, использует JSON-подобные документы и схему базы данных. Написана на языке C++.

Если в традиционном мире SQL есть таблицы, то в мире MongoDB есть коллекции. И если в реляционных БД таблицы хранят однотипные жестко структурированные объекты, то в коллекции могут содержать самые разные объекты, имеющие различную структуру и различный набор свойств.

## 1.4 Необходимые драйвера для MongoDB на C#

Для установки нужных драйверов для MongoDb нужно зайти в проект и открыть контекстное меню и нажать NuGet. После этого в поисковике нужно найти нужные драйвера. Пакетный менеджер предоставлен в рисунке 1.4.1.

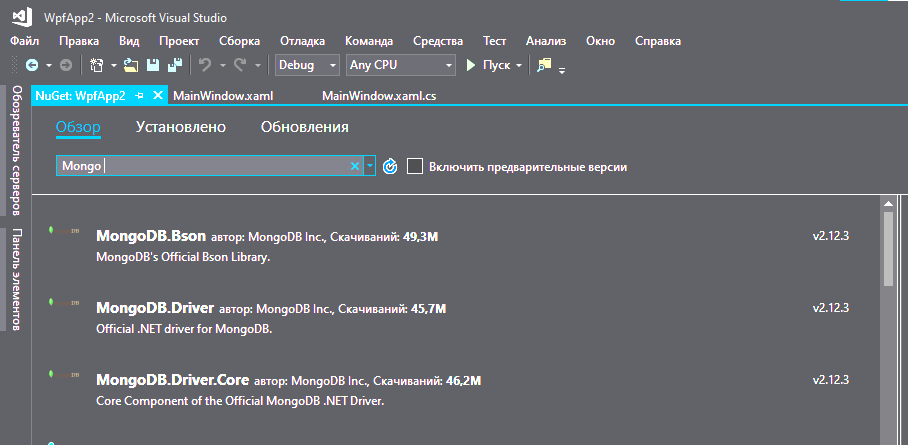


Рисунок 1.4.1 Пакетный менеджер

После нахождения нужных драйверов, необходимо скачать все 3 драйвера:

* MongoDB.Bson;
* MongoDB.Driver;
* MongoDB.Driver.Core.

# 2 РЕАЛИЗАЦИЯ ПРОЕКТА

## 2.1 Подключение MonboDB

Перед началом работы с проектом всегда необходимо открыть локальный сервер и MonboDB Compass.

Иконка для запуска локального сервера представлен на рисунке 2.1.1.



Рисунок 2.1.1 Иконка для запуска сервера

Иконка для запуска MonboDB Compass представлен на рисунке 2.1.2.



Рисунок 2.1.2 Иконка для запуска MonboDB Compass

Далее открываем Visual Studio и подключаем все необходимые пространства имен (Рис. 2.1.3).



Рисунок 2.1.3 Все необходимые пространства имен

## 2.2 Создание базы данных

Каждый объект в моей коллекции содержит 4 полей (Рис. 2.2.1), а именно:

* Уникальный номер товара (ID);
* Название товара;
* Описание товара;
* Цена товара;

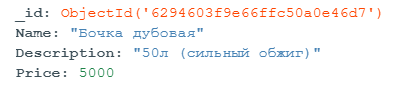


Рисунок 2.2.1 Пример полей

Точно таким же образом создается база для хранения информации о пользователях с одним лишь отличием: поля для этого будут другими. Для хранения информации о пользователях я решила использовать 8 полей (Рис. 2.2.2):

* Уникальный ключ пользователя(ID);
* Имя;
* Фамилия;
* Отчество;
* Дата рождения;
* Телефон;
* Пароль;
* Администратор ли он?

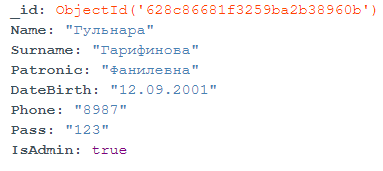


Рисунок 2.2.2 пример хранения информации о пользователях

Я разработала собственный простой для восприятия дизайн. При его создании я пользовалась удобствами WPF (Например, для создания кнопки я перетаскивала готовый элемент). Все это было сделано для того, чтобы интерфейс был под моим контролем, и никто не смог его сломать. Дизайн моего интерфейса(Рис.2.2.3).

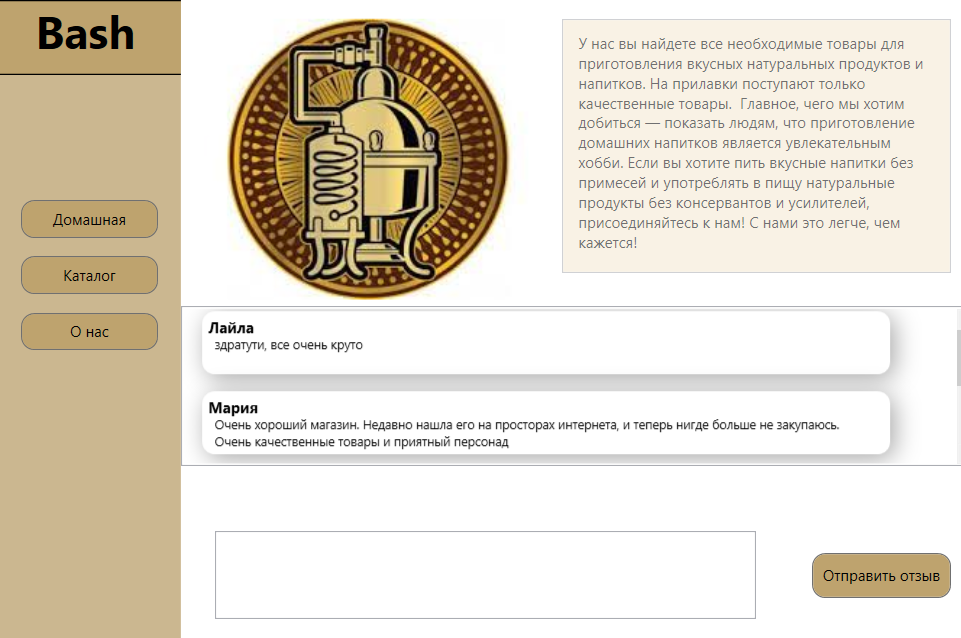


Рисунок 2.2.3 Дизайн интерфейса

## 2.3 Создание приложения

Для создания первой страницы я пользовалась интерфейсом WPF. Таким образом, я создала окно для регистрации и входа в моем приложении там описаны следующие текстбоксы:

* Имя пользователя;
* Фамилия пользователя;
* Отчество пользователя;
* Дата рождения;
* Логин для пользователя (это его Телефон);
* Пароль пользователя;
* Кнопка для регистрации;
* Кнопка для входа;

После того, как пользователь введет все данные и нажмёт соответствующую кнопку, он увидит сообщение об успешной регистрации или входе.



Рисунок 2.3.1 Регистрация и вход

После того, как пользователь зарегистрировался, он автоматически переходит на страницу «О нас» (Рис. 2.3.2). Здесь пользователь может прочесть о данном магазине и почитать отзывы других пользователей. При желании также может оставить собственный отзыв.

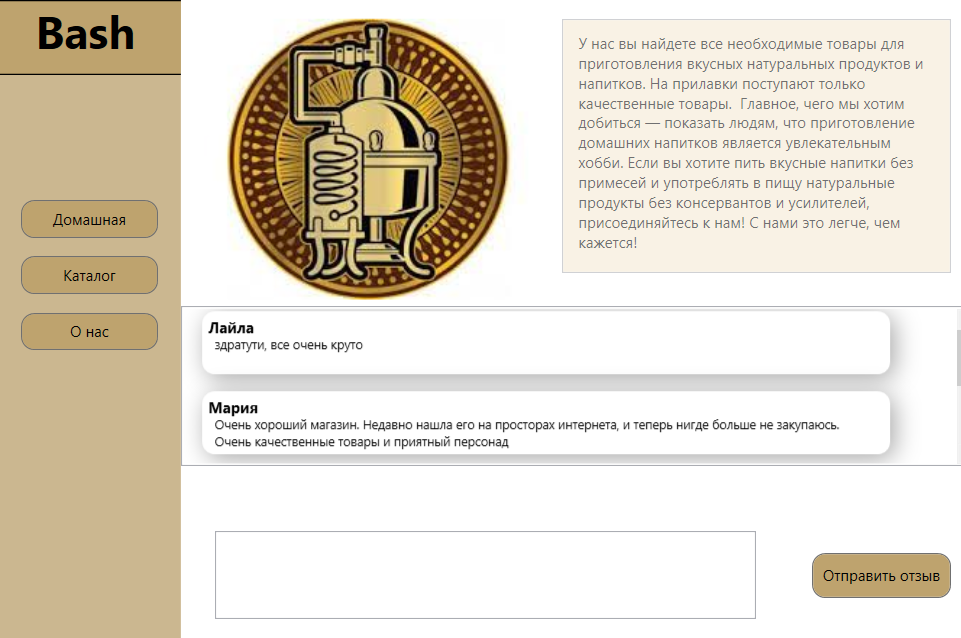


Рисунок 2.3.2 О нас

По нажатию на кнопку «Домашняя», пользователь переходит в личный кабинет. Здесь он может просмотреть его личную информацию. При необходимости есть возможность редактировать профиль. Также пользователь может просмотреть список сделанных им заказов.

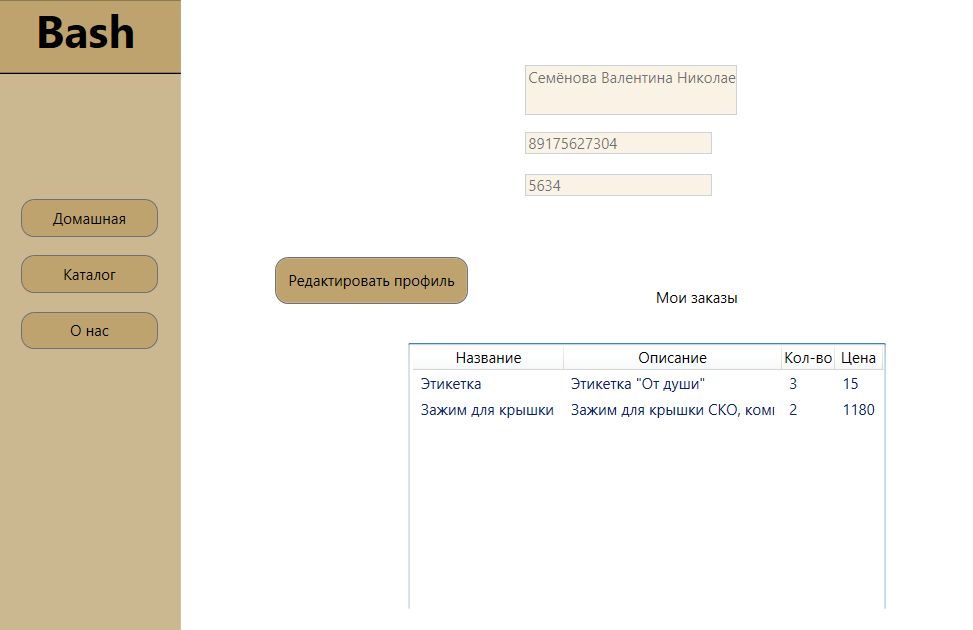


Рисунок 2.3.3 «Домашняя»

Чтобы отредактировать профиль, пользователь должен нажать кнопку «Редактировать профиль». После этого поля станут открытыми, и в них можно будет изменить информацию. Также появится кнопка «Сохранить», при нажатии на которую новая информация сохраниться в базе данных, а сама кнопка станет недоступной для пользователя (Рис. 2.3.4).



Рисунок 2.3.4 Редактирование профиля

Нажав на кнопку «Каталог», пользователь попадет на страницу со списком товаров, доступных для заказа (Рис. 2.3.5). На этой странице также присутствуют элементы для поиска определенного товара и сортировки по цене.

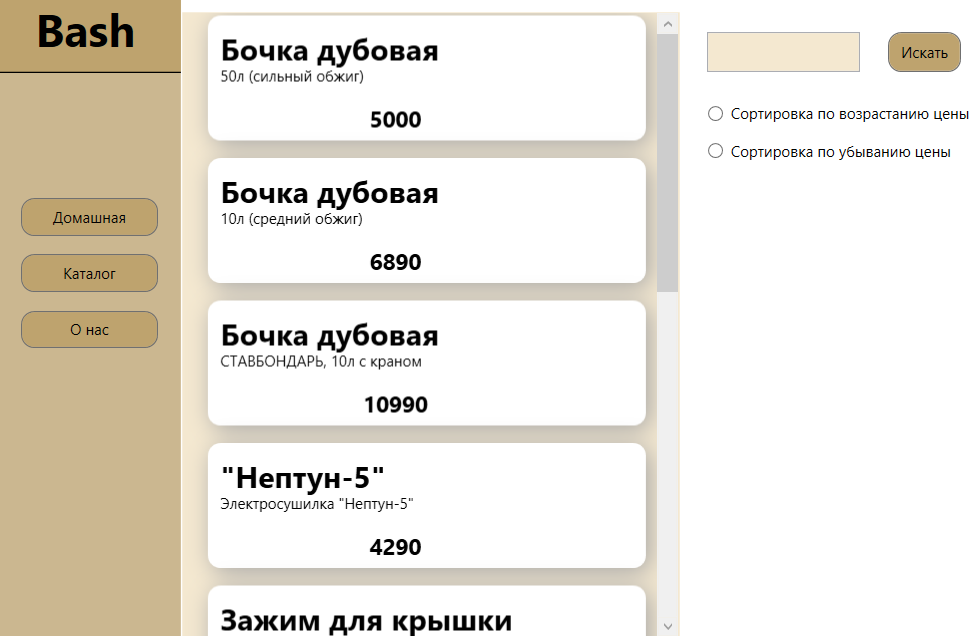


Рисунок 2.3.5 Окно для пользователя

Выбрав товар с помощью нажатия на элемент списка, пользователь переходит на новую страницу для заказа (Рис. 2.3.6). Здесь уже заполнена информация о товаре, нужно лишь выбрать количество товара. После нажатия на кнопку «Добавить заказа», заказа будет добавлен в базу данный и выведен в личном кабинете.

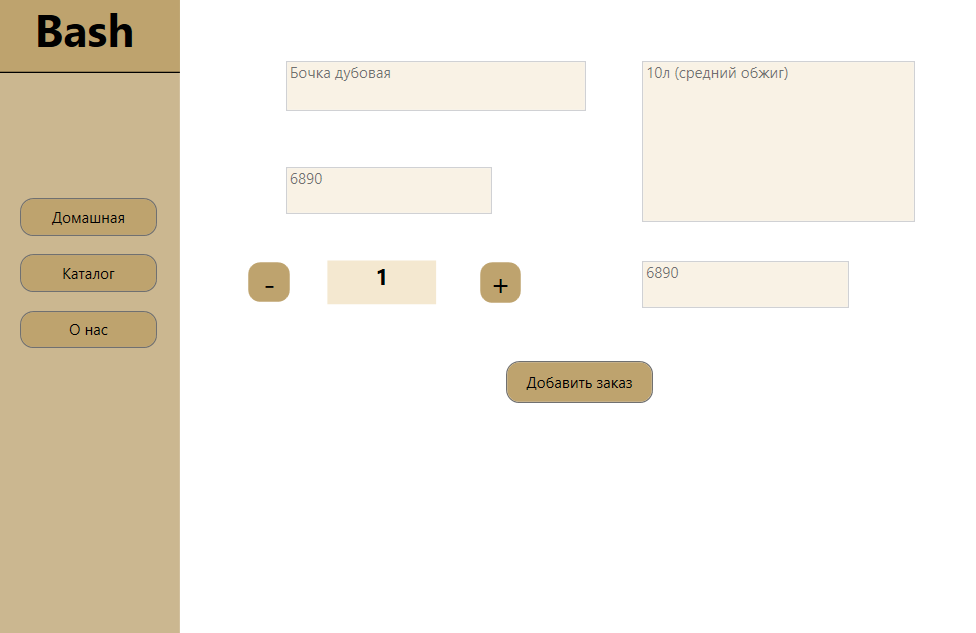


Рисунок 2.3.6 Страница для заказа

2.4 Работа за администратора

При работе за администратора приложения нужно будет выполнить почти такие же действия, как у пользователя, но чуть-чуть сложнее, во-первых, нужно в MongoDB выставить значение у пользователя IsAdmin : true, и после чего нажимаем кнопку «Update» это значит, что мы пользователю присваиваем значение администратора (Рис 2.4.1).



Рисунок 2.4.1 Делаем пользователя администратором

Затем администратор вводит свои данные в окне «Вход» и приложение его приветствует как администратора (Рис 2.4.2).

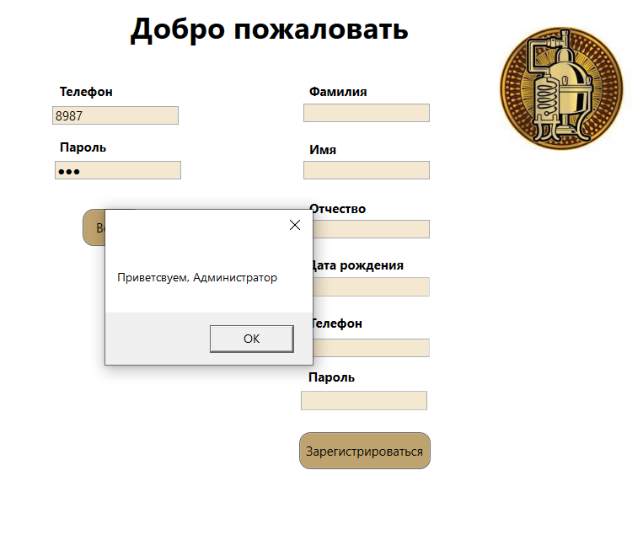


Рисунок 2.4.2 Уведомление для администратора

Когда администратор произвел вход в приложение, на странице «Каталог» добавляется кнопка «Добавить товар». (Рис 2.4.3).

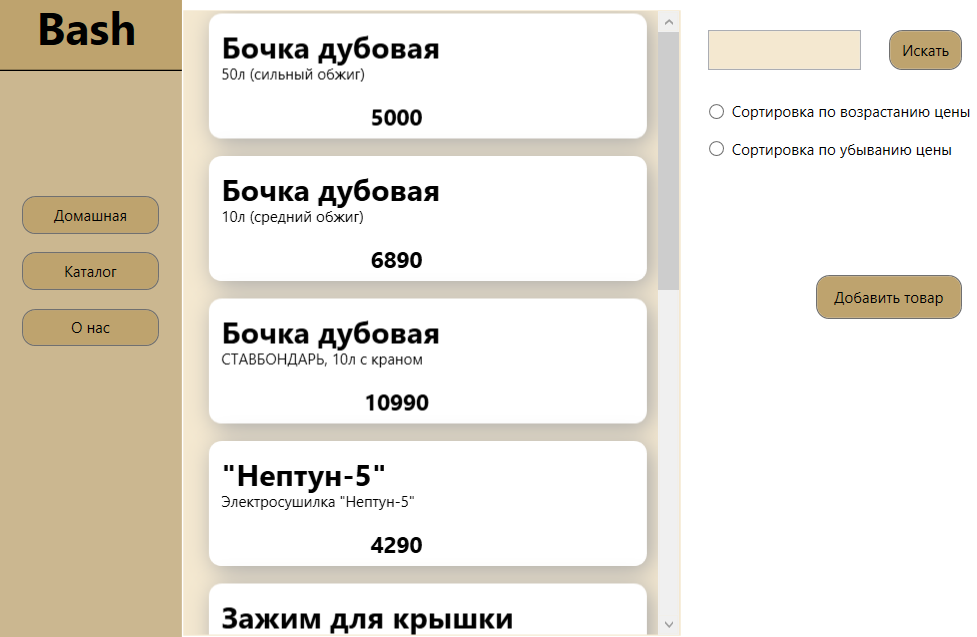


Рисунок 2.4.3 Каталог администратора

При нажатии на данную кнопку, открывается окно для добавления товара. (Рис 2.4.4).

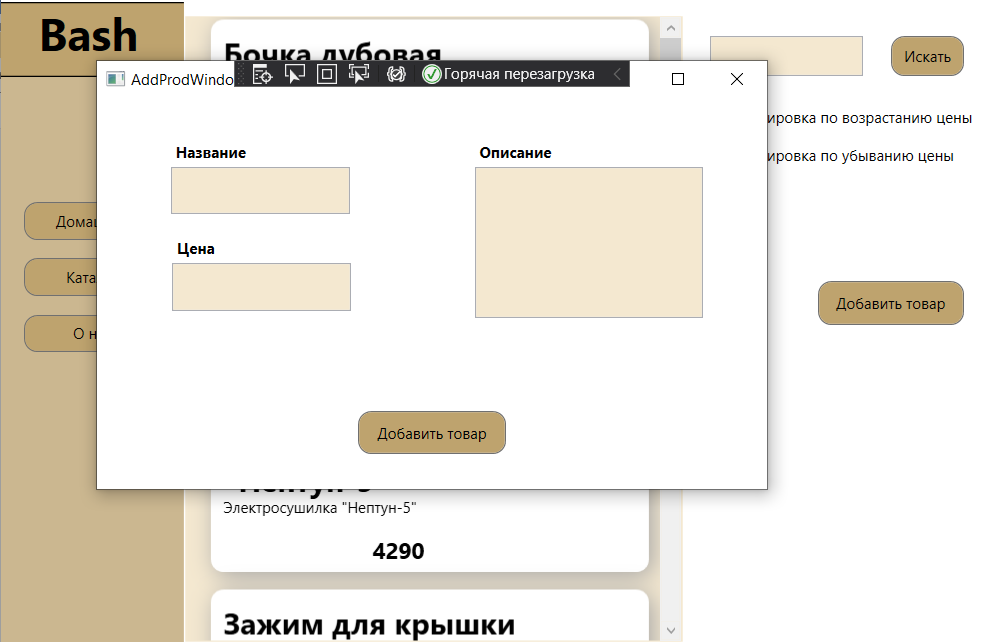


Рисунок 2.4.4 Окно добавления товара

Также на странице «Каталог» есть возможность удалить или редактировать товар, нажав на элемент списка (Рис 2.4.5).

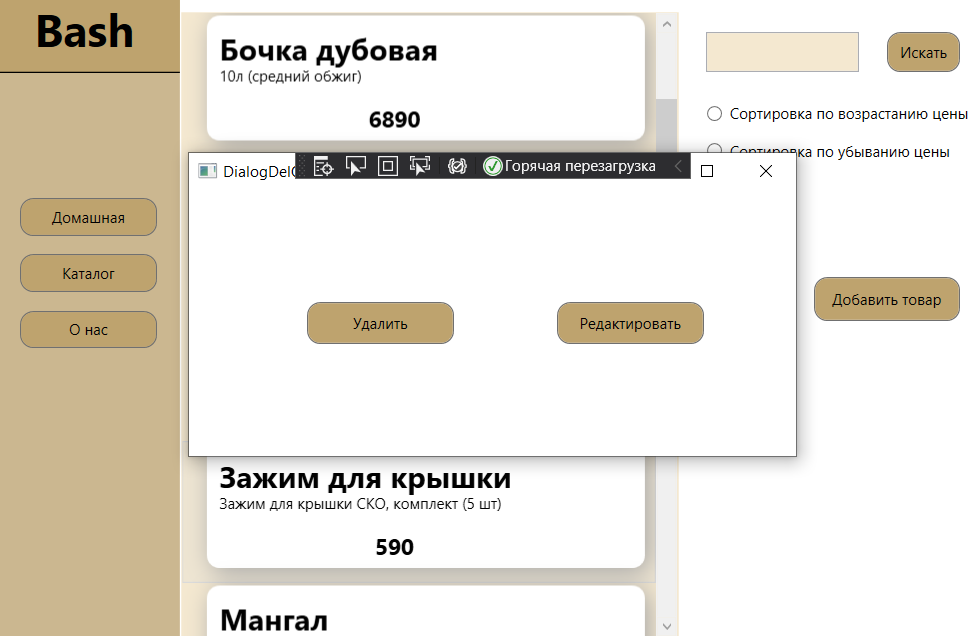


Рисунок 2.4.5 Удаление товара

На странице «Домашняя» администратор может просмотреть список заказов от всех пользователей (Рис 2.4.6).

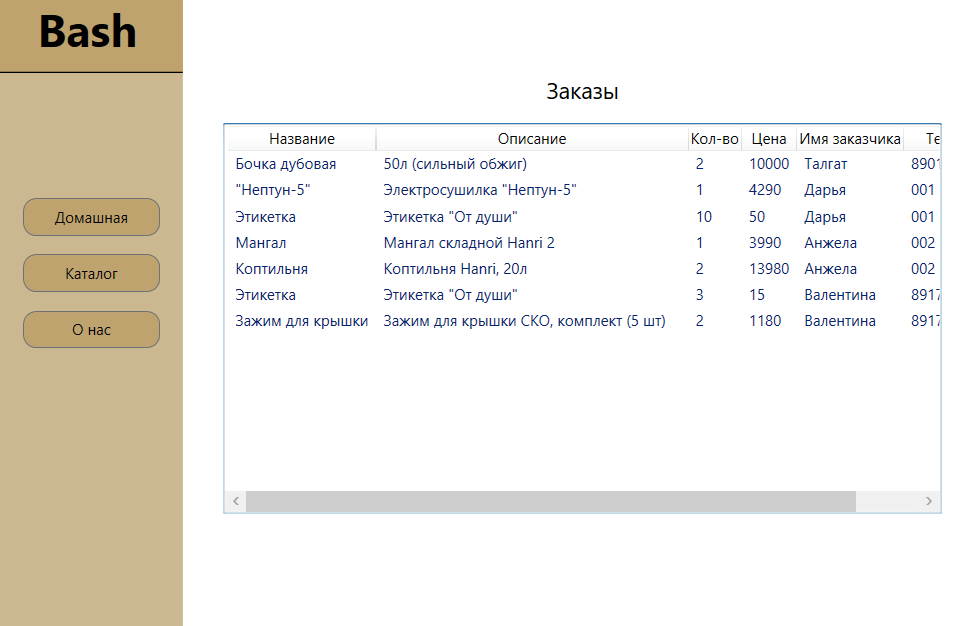


Рисунок 2.4.5 Страница администратора для просмотра заказов

Администратор также может удалить выбранный заказ (Рис 2.5.7).

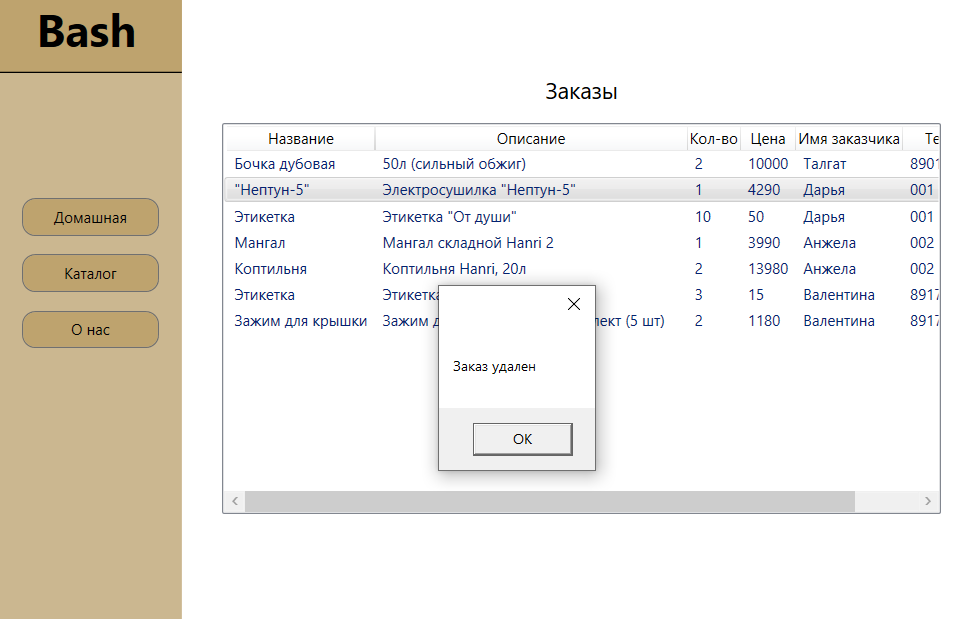


Рисунок 2.4.7 Удаление заказа

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта была сделана работоспособная программа. С помощью WPF был создан удобный пользовательский интерфейс, благодаря которому работать с программой сможет даже неопытный пользователь. Также с помощью языка программирования C# был реализован весь функционал программы.

В процессе разработки была использована база данных Mongo DB, которая позволяет хранить определённый объём информации. Данная БД проявила себя как надёжная и функциональная, так как при работе с ней никаких проблем не возникало. Также можно расширять базу данных с течением времени, наполняя её дополнительной информацией. Были реализованы и основные операции для работы с БД, такие как визуальное отображение данных, возможность добавления и удаления записей, а также выбор определенных данных из БД.

В качестве дальнейшего развития возможна доработка программы в виде обновлений, основанная на рекомендациях пользователей, в целях улучшить данный продукт в лучшую сторону.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Введение в WPF [Электронный ресурс]. – режим доступа: https://metanit.com/sharp/wpf/1.php;
2. Язык программирования C# [Электронный ресурс]. – режим доступа: https://btfr.cc/1ia
3. Справочник [Электронный ресурс]. – режим доступа: <https://ru.wikipedia.org/wiki/MongoDB>;
4. Справочник [Электронный ресурс]. – режим доступа: <https://metanit.com/nosql/mongodb/1.1.php>;
5. Документация по C# [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/csharp/>;
6. Справочник [Электронный ресурс]. – режим доступа: https://metanit.com/nosql/mongodb/2.10.php;
7. Справочник [Электронный ресурс]. – режим доступа: https://metanit.com/nosql/mongodb/2.13.php;
8. Справочник [Электронный ресурс]. – режим доступа: https://coderlessons.com/tutorials/bazy-dannykh/uchitsia-mongodb/mongodb-sortirovka-zapisei;
9. Справочник [Электронный ресурс]. – режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/api/system.collections.generic.list-1.find?view=net-6.0>;

ПРИЛОЖЕНИЕ A

**Техническое задание…**

**Вводная часть:**описание, общие положения, назначение и цели проекта.

Разработать standalone приложение для магазина. Данные будут браться из базы данных MongoDB. Приложение должно быть разработано на платформе WPF. Приложение будет цениться людьми, желающими готовить напитки и закуски лично с помощью проверенных и качественных товаров.

**Основная:**содержит функциональные и технические требования.

Реализовать список всех доступных товаров для заказа.

Обязательно должна быть регистрация пользователя его панель и кнопки, и ещё панель администратора, в которой будет реализовано добавление или удаление из базы. Проект должен быть сделан из страниц.

1. Цветовая палитра белый/бежевый/коричневый цвета
2. Возможность регистрации как клиент
3. Возможность регистрации как администратор
4. Добавление данных у администратора, так и удаление из базы

**Начальный экран**

1. При старте программы на белом фоне будут поля для регистрации и авторизации пользователя.

**Форма начального экрана**

1. Текстовые поля для регистрации:
   1. Для ввода фамилии пользователя.
   2. Для ввода имени пользователя.
   3. Для ввода отчества пользователя.
   4. Для ввода даты рождения.
   5. Для ввода телефона.
   6. Для ввода пароля.
2. Текстовые поля для авторизации:

2.1Для ввода телефона.

2.2Для ввода пароля.

1. Кнопки:

2.1 Для авторизации в программу.

2.2 Для регистрации в программе.

**Форма каталога**

1. Список товаров
2. Кнопки
   1. Сортировка по цене
   2. Поиск по названию товара
   3. Добавить товар (Только Админ)

**Формы для добавление товара**

* + - 1. Текстовые поля
  1. Название товара
  2. Описание товара
  3. Цена товара

1. Кнопки

2.1Кнопка Добавить товар

**Формы личного кабинета**

* + - 1. Список заказов (всех для админа)
      2. Текстовые поля

2.1ФИО пользователя

2.2Телефон

2.3Пароль

3.Кнопки

3.1 Кнопка Редактировать профиль

3.2 Кнопка Сохранить

# ПРИЛОЖЕНИЕ B

using MongoDB.Bson.Serialization.Attributes;

using MongoDB.Bson;

using System.Linq;

using MongoDB.Driver;

namespace BashShop.Data

{

public class Users

{

[BsonId]

[BsonIgnoreIfDefault]

public ObjectId Id { get; set; }

[BsonElement]

public string Name { get; set; }

[BsonElement]

public string Surname { get; set; }

[BsonElement]

public string Patronic { get; set; }

[BsonElement]

public string DateBirth { get; set; }

[BsonElement]

public string Phone { get; set; }

[BsonElement]

public string Pass { get; set; }

[BsonElement]

public byte[] Photo { get; set; }

[BsonElement]

public bool IsAdmin { get; set; }

public Users(string name, string surname, string patronic, string datebirth, string phone, string pass, byte[] photo, bool isadmin = false)

{

Name = name;

Surname = surname;

Patronic = patronic;

DateBirth = datebirth;

Phone = phone;

Pass = pass;

IsAdmin = isadmin;

Photo = photo;

}

public Users(string phone, string pass)

{

Phone = phone;

Pass = pass;

}

public async static void AddUser(Users user)

{

MongoClient client = new MongoClient();

var db = client.GetDatabase("BashShop");

var collection = db.GetCollection<Users>("users");

await collection.InsertOneAsync(user);

}

public static bool LogInUser(Users user)

{

MongoClient client = new MongoClient();

var db = client.GetDatabase("BashShop");

var collection = db.GetCollection<Users>("users");

var users = collection.Find(x => true).ToList();

App.user = users.Where(x => x.Phone == user.Phone && x.Pass == user.Pass).FirstOrDefault();

if (App.user == null)

return false;

return true;

}

public static void EditProfile()

{

MongoClient client = new MongoClient();

var db = client.GetDatabase("BashShop");

var data = db.GetCollection<Users>("users");

var UpdateDef = Builders<Users>.Update.Set("Name", App.user.Name).Set("Surname", App.user.Surname).Set("Patronic", App.user.Patronic).Set("Phone", App.user.Phone).Set("Pass", App.user.Pass);

data.UpdateOne(basa => basa.Id == App.user.Id, UpdateDef);

}

public void Update()

{

MongoClient client = new MongoClient();

var db = client.GetDatabase("ReabilitionCenter");

var collection = db.GetCollection<Users>("users");

var UpdateDef = Builders<Users>.Update.Set("Photo", App.user.Photo);

collection.UpdateOne(basa => basa.Id == App.user.Id, UpdateDef);

}}}

using MongoDB.Bson;

using MongoDB.Bson.Serialization.Attributes;

using MongoDB.Driver;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace BashShop.Data

{

public class Products

{

[BsonId]

[BsonIgnoreIfDefault]

public ObjectId Id { get; set; }

[BsonElement]

public string Name { get; set; }

[BsonElement]

public string Description { get; set; }

[BsonElement]

public double Price { get; set; }

public Products(string name, string des, double price)

{

Name = name;

Description = des;

Price = price;

}

public async static void AddProd(Products prod)

{

MongoClient client = new MongoClient();

var db = client.GetDatabase("BashShop");

var collection = db.GetCollection<Products>("prods");

await collection.InsertOneAsync(prod);

}

public async static Task<List<Products>> GetProd()

{

MongoClient client = new MongoClient();

var db = client.GetDatabase("BashShop");

var collection = db.GetCollection<Products>("prods");

return collection.Find(x => true).ToList();

}

public static void DeleteProd(Products prod)

{

MongoClient client = new MongoClient();

var db = client.GetDatabase("BashShop");

var collection = db.GetCollection<Products>("prods");

collection.DeleteOne(x => x.Id == prod.Id);

}

public static void EditProd()

{

MongoClient client = new MongoClient();

var db = client.GetDatabase("BashShop");

var data = db.GetCollection<Products>("prods");

var UpdateDef = Builders<Products>.Update.Set("Name", App.products.Name).Set("Description", App.products.Description).Set("Price", App.products.Price);

data.UpdateOne(basa => basa.Id == App.products.Id, UpdateDef);

}

public static async Task<List<Products>> SearchList(string word)

{

MongoClient client = new MongoClient();

var db = client.GetDatabase("BashShop");

var collection = db.GetCollection<Products>("prods");

return collection.Find(x => x.Name == word).ToList();

}

public static async Task<List<Products>> SortingList()

{

MongoClient client = new MongoClient();

var db = client.GetDatabase("BashShop");

var collection = db.GetCollection<Products>("prods");

//return collection.Find(x => true).Sort("prods":1, sort).ToList();

var w = from c in collection.AsQueryable<Products>()

orderby c.Price

select c;

return w.ToList();

}

public static async Task<List<Products>> SortingLisrMinus()

{

MongoClient client = new MongoClient();

var db = client.GetDatabase("BashShop");

var collection = db.GetCollection<Products>("prods");

//return collection.Find(x => true).Sort("prods":1, sort).ToList();

var w = from c in collection.AsQueryable<Products>()

orderby c.Price descending

select c;

return w.ToList();

}}}

using MongoDB.Bson;

using MongoDB.Bson.Serialization.Attributes;

using MongoDB.Driver;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace BashShop.Data

{

public class Orders

{

[BsonId]

[BsonIgnoreIfDefault]

public ObjectId id { get; set; }

[BsonElement]

public string NameProd { get; set; }

[BsonElement]

public double PriceProds { get; set; }

[BsonElement]

public string DesProd { get; set; }

[BsonElement]

public int Count { get; set; }

[BsonElement]

public string NameUser { get; set; }

[BsonElement]

public string PhoneUser { get; set; }

public Orders(string name, double price, string desprod, int count, string nameus, string phoneus)

{

NameProd = name;

PriceProds = price;

DesProd = desprod;

Count = count;

NameUser = nameus;

PhoneUser = phoneus;

}

public async static void AddOrder(Orders or)

{

MongoClient client = new MongoClient();

var db = client.GetDatabase("BashShop");

var collection = db.GetCollection<Orders>("orders");

await collection.InsertOneAsync(or);

}

public async static Task<List<Orders>> GetInfoOrder()

{

MongoClient client = new MongoClient();

var db = client.GetDatabase("BashShop");

var collection = db.GetCollection<Orders>("orders");

return collection.Find(x => x.PhoneUser == App.user.Phone).ToList();

}

public async static Task<List<Orders>> GetInfoAllOrders()

{

MongoClient client = new MongoClient();

var db = client.GetDatabase("BashShop");

var collection = db.GetCollection<Orders>("orders");

return collection.Find(x => true).ToList();

}

public static void DeleteOrder(Orders order)

{

MongoClient client = new MongoClient();

var db = client.GetDatabase("BashShop");

var collection = db.GetCollection<Orders>("orders");

collection.DeleteOne(x=>x.id == order.id);

}}}

using MongoDB.Bson;

using MongoDB.Bson.Serialization.Attributes;

using MongoDB.Driver;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace BashShop.Data

{

public class Reviews

{

[BsonId]

[BsonIgnoreIfDefault]

public ObjectId id { get; set; }

[BsonElement]

public string Review { get; set; }

[BsonElement]

public string NameUs { get; set; }

public Reviews(string rev, string nameus)

{

Review = rev;

NameUs = nameus;

}

public async static void AddReview(Reviews rev)

{

MongoClient client = new MongoClient();

var db = client.GetDatabase("BashShop");

var collection = db.GetCollection<Reviews>("reviews");

await collection.InsertOneAsync(rev);

}

public async static Task<List<Reviews>> GetInfoReview()

{

MongoClient client = new MongoClient();

var db = client.GetDatabase("BashShop");

var collection = db.GetCollection<Reviews>("reviews");

return collection.Find(x => true).ToList();

}}}

<Window x:Class="BashShop.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:BashShop"

mc:Ignorable="d"

Title="MainWindow" Height="550" Width="800" ResizeMode="NoResize">

<Grid x:Name="MainRoot">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="149\*"/>

<ColumnDefinition Width="651\*"/>

</Grid.ColumnDefinitions>

<Frame x:Name="MainFrame" Source="/Pages/AboutUsPage.xaml" NavigationUIVisibility="Hidden" Grid.Column="1"/>

<Rectangle Fill="#CBB790" HorizontalAlignment="Center" Height="534" Stroke="Black" VerticalAlignment="Center" Width="150"/>

<Rectangle Fill="#BEA36E" HorizontalAlignment="Center" Height="60" Stroke="Black" VerticalAlignment="Top" Width="150"/>

<Button x:Name="About\_UsBtn" Style="{DynamicResource ButtonStyle}" Content="О нас" Background="#BEA36E" HorizontalAlignment="Center" Margin="0,250,0,0" VerticalAlignment="Top" Height="30" Width="110" Click="About\_Us\_Click" />

<Button x:Name="CatalogBtn" Style="{DynamicResource ButtonStyle}" Content="Каталог" Background="#BEA36E" HorizontalAlignment="Center" Margin="0,205,0,0" VerticalAlignment="Top" Height="30" Width="110" Click="Catalog\_Click" />

<Button x:Name="HomePageBtn" Style="{DynamicResource ButtonStyle}" Content="Домашная" Background="#BEA36E" HorizontalAlignment="Center" Margin="0,160,0,0" VerticalAlignment="Top" Height="30" Width="110" Click="HomePage\_Click"/>

<Label Content="Bash" FontWeight="Bold" HorizontalAlignment="Left" Margin="25,2,0,0" VerticalAlignment="Top" FontSize="36" FontFamily="Kurale"/>

</Grid>

</Window>

using System.Windows;

using BashShop.APages;

using BashShop.Pages;

namespace BashShop

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public bool IsAdmin;

public MainWindow(bool id)

{

InitializeComponent();

IsAdmin = id;

}

private void HomePage\_Click(object sender, RoutedEventArgs e)

{

if (App.user.IsAdmin == true)

{

MainFrame.Navigate(new AHomePage());

}

else

MainFrame.Navigate(new HomePage());

}

private void Catalog\_Click(object sender, RoutedEventArgs e)

{

if (App.user.IsAdmin == true)

{

MainFrame.Navigate(new ACatalogPage());

}

else

MainFrame.Navigate(new CatalogPage());

}

private void About\_Us\_Click(object sender, RoutedEventArgs e)

{

MainFrame.Navigate(new AboutUsPage());

}}}

<Window x:Class="BashShop.Windows.LogWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:BashShop.Windows"

mc:Ignorable="d"

Title="LogWindow" Height="600" Width="650">

<Grid Background="White">

<Label Content="Добро пожаловать" FontWeight="Bold" FontSize="28" HorizontalAlignment="Left" Margin="121,23,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="PhoneLogTB" Background="#F4E8D0" HorizontalAlignment="Left" Margin="52,122,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"/>

<Label Content="Телефон" FontWeight="Bold" HorizontalAlignment="Left" Margin="54,95,0,0" VerticalAlignment="Top"/>

<PasswordBox x:Name="PassLog" Background="#F4E8D0" HorizontalAlignment="Left" Margin="54,174,0,0" VerticalAlignment="Top" Width="120"/>

<Label Content="Пароль" FontWeight="Bold" HorizontalAlignment="Left" Margin="54,148,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="SurnameTB" Background="#F4E8D0" HorizontalAlignment="Left" Margin="290,120,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"/>

<Label Content="Фамилия" FontWeight="Bold" HorizontalAlignment="Left" Margin="291,95,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="NameTB" Background="#F4E8D0" HorizontalAlignment="Left" Margin="290,174,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"/>

<Label Content="Имя" FontWeight="Bold" HorizontalAlignment="Left" Margin="291,150,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="PhoneRegTB" MaxLength="11" Background="#F4E8D0" HorizontalAlignment="Left" Margin="290,343,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"/>

<Label Content="Телефон" FontWeight="Bold" HorizontalAlignment="Left" Margin="291,315,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="DateBTB" MaxLength="10" Background="#F4E8D0" HorizontalAlignment="Left" Margin="290,285,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"/>

<Label Content="Дата рождения" FontWeight="Bold" HorizontalAlignment="Left" Margin="289,260,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="PatronicTB" Background="#F4E8D0" HorizontalAlignment="Left" Margin="290,230,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"/>

<Label Content="Отчество" FontWeight="Bold" HorizontalAlignment="Left" Margin="291,206,0,0" VerticalAlignment="Top"/>

<Button Style="{DynamicResource ButtonStyle}" x:Name="LogIn" Height="35" Content="Войти" Background="#BEA36E" HorizontalAlignment="Left" Margin="81,220,0,0" VerticalAlignment="Top" Width="58" Click="LogIn\_Click"/>

<Button Style="{DynamicResource ButtonStyle}" x:Name="LogUp" Content="Зарегистрироваться" Background="#BEA36E" HorizontalAlignment="Left" Margin="286,432,0,0" VerticalAlignment="Top" Height="35" Width="125" Click="LogUp\_Click"/>

<Image HorizontalAlignment="Left" Height="144" Margin="457,47,0,0" VerticalAlignment="Top" Width="157" Source="/img/logo.jpg"/>

<PasswordBox x:Name="PassReg" Background="#F4E8D0" HorizontalAlignment="Left" Margin="288,393,0,0" VerticalAlignment="Top" Width="120"/>

<Label Content="Пароль" FontWeight="Bold" HorizontalAlignment="Left" Margin="290,366,0,0" VerticalAlignment="Top"/>

</Grid>

</Window>

using BashShop.Data;

using System.Windows;

namespace BashShop.Windows

{

/// <summary>

/// Логика взаимодействия для LogWindow.xaml

/// </summary>

public partial class LogWindow : Window

{

public LogWindow()

{

InitializeComponent();

}

private void LogIn\_Click(object sender, RoutedEventArgs e)

{

if (PhoneLogTB.Text == "" || PassLog.Password == "")

{

MessageBox.Show("Введите все данные");

}

else

{

Users us = new Users(PhoneLogTB.Text, PassLog.Password);

Users.LogInUser(us);

if (App.user.IsAdmin == true)

{

MessageBox.Show("Приветсвуем, Администратор");

MainWindow mw = new MainWindow(us.IsAdmin);

mw.Show();

this.Close();

}

else

{

MessageBox.Show("Успешный вход");

MainWindow mw = new MainWindow(us.IsAdmin);

mw.Show();

this.Close();

}

}

}

private void LogUp\_Click(object sender, RoutedEventArgs e)

{

if (NameTB.Text == "" || SurnameTB.Text == "" || PatronicTB.Text == "" || DateBTB.Text == "" || PhoneRegTB.Text == "" || PassReg.Password == "")

{

MessageBox.Show("Введите все данные");

}

else

{

Users us = new Users(NameTB.Text, SurnameTB.Text, PatronicTB.Text, DateBTB.Text, PhoneRegTB.Text, PassReg.Password, null);

Users.AddUser(us);

MessageBox.Show("Вы зарегистрировались!");

MainWindow mw = new MainWindow(us.IsAdmin);

mw.Show();

this.Close();

App.user = us;

}}}}

<Page x:Class="BashShop.Pages.AboutUsPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:BashShop.Pages"

mc:Ignorable="d"

d:DesignHeight="550" d:DesignWidth="650"

Title="AboutUsPage">

<Grid Background="White">

<Image HorizontalAlignment="Left" Height="243" Margin="0,15,0,0" VerticalAlignment="Top" Width="300" Source="/img/logo.jpg"/>

<TextBox IsEnabled="False" Text="У нас вы найдете все необходимые товары для приготовления вкусных натуральных продуктов и напитков. На прилавки поступают только качественные товары.

Главное, чего мы хотим добиться — показать людям, что приготовление домашних напитков является увлекательным хобби. Если вы хотите пить вкусные напитки без примесей и употреблять в пищу натуральные продукты без консервантов и усилителей, присоединяйтесь к нам! С нами это легче, чем кажется!" Background="#F4E8D0" Padding="10" TextWrapping="Wrap" HorizontalAlignment="Left" Margin="305,15,0,0" VerticalAlignment="Top" Height="203" Width="311"/>

<ListView x:Name="ReviewsLV" Margin="0,245,0,140">

<ListView.ItemTemplate>

<DataTemplate>

<Border Background="White" CornerRadius="10" Margin="10,0,0,10">

<Border.Effect>

<DropShadowEffect BlurRadius="30" Color="LightGray" />

</Border.Effect>

<Canvas Height="50" Width="550" HorizontalAlignment="Center">

<TextBlock

Text="{Binding NameUs}"

FontWeight="Bold"

FontSize="12"

Margin="5"

Width="300"

TextAlignment="Left"

Background="Transparent"/>

<TextBlock

Text="{Binding Review}"

FontSize="10"

Margin="10,20"

Width="500"

Height="300"

TextAlignment="Left"

TextWrapping="Wrap"

Background="Transparent"/>

</Canvas>

</Border>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<TextBox x:Name="DesRev" HorizontalAlignment="Left" Margin="27,425,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="433" Height="70"/>

<Button x:Name="AddRevBtn" Style="{DynamicResource ButtonStyle}" Background="#BEA36E" Content="Отправить отзыв" HorizontalAlignment="Left" Margin="505,442,0,0" VerticalAlignment="Top" Height="36" Width="111" Click="AddRevBtn\_Click"/>

</Grid>

</Page>

using BashShop.Data;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

namespace BashShop.Pages

{

/// <summary>

/// Логика взаимодействия для AboutUsPage.xaml

/// </summary>

public partial class AboutUsPage : Page

{

public AboutUsPage()

{

InitializeComponent();

GetInfoReviews();

}

private void AddRevBtn\_Click(object sender, RoutedEventArgs e)

{

if (DesRev.Text == " ")

{

MessageBox.Show("Введите все данные");

}

else

{

Reviews rev = new Reviews(DesRev.Text, App.user.Name);

Reviews.AddReview(rev);

GetInfoReviews();

DesRev.Text = "";

}}

public async Task GetInfoReviews()

{

ReviewsLV.ItemsSource = await Reviews.GetInfoReview();

}}}

<Page x:Class="BashShop.Pages.CatalogPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:BashShop.Pages"

mc:Ignorable="d"

d:DesignHeight="550" d:DesignWidth="650"

Title="CatalogPage">

<Grid Background="White">

<ListView

Name="listprod" Margin="0,10,0,0" SelectionChanged="listprod\_SelectionChanged"

Background="#F4E8D0" BorderBrush="Transparent"

HorizontalContentAlignment="Center" Width="400"

HorizontalAlignment="Left">

<ListView.ItemTemplate>

<DataTemplate>

<Border Background="White" CornerRadius="10" Margin="10,0,0,10">

<Border.Effect>

<DropShadowEffect BlurRadius="30" Color="LightGray" />

</Border.Effect>

<Canvas Height="100" Width="350" HorizontalAlignment="Center">

<TextBlock

Text="{Binding Name}"

FontWeight="Bold"

FontSize="24"

Margin="10"

Width="300"

TextAlignment="Left"

Background="Transparent"/>

<TextBlock

Text="{Binding Description}"

FontSize="12"

Margin="10,40"

Width="300"

TextAlignment="Left"

TextWrapping="Wrap"

Background="Transparent"/>

<TextBlock

Text="{Binding Price}"

FontWeight="Bold"

FontSize="18"

Margin="0,70"

Width="300"

TextAlignment="Center"

Background="Transparent"/>

</Canvas>

</Border>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<TextBox x:Name="SearchWord" HorizontalAlignment="Left" Margin="421,27,0,0" Background="#F4E8D0" TextWrapping="Wrap" VerticalAlignment="Top" Width="122" Height="32"/>

<Button Style="{DynamicResource ButtonStyle}" x:Name="SearchBtn" Height="32" Content="Искать" Background="#BEA36E" HorizontalAlignment="Left" Margin="566,27,0,0" VerticalAlignment="Top" Width="58" Click="SearchBtn\_Click"/>

<RadioButton x:Name="Sort" Content="Сортировка по возрастанию цены" HorizontalAlignment="Left" Margin="421,85,0,0" VerticalAlignment="Top" Height="25" Width="219" Click="Sort\_Click"/>

<RadioButton x:Name="Sort\_Copy" Content="Сортировка по убыванию цены" HorizontalAlignment="Left" Margin="421,115,0,0" VerticalAlignment="Top" Height="25" Width="219" Click="Sort\_Copy\_Click"/>

</Grid>

</Page>

using BashShop.Data;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

namespace BashShop.Pages

{

/// <summary>

/// Логика взаимодействия для CatalogPage.xaml

/// </summary>

public partial class CatalogPage : Page

{

public CatalogPage()

{

InitializeComponent();

GetInfoProdAsync();

}

public async Task GetInfoProdAsync()

{

listprod.ItemsSource = await Products.GetProd();

}

private void listprod\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

var lstp = listprod.SelectedItem as Products;

this.NavigationService.Navigate(new OrderPage(lstp));

}

public async Task GetSearchList()

{

listprod.ItemsSource = await Products.SearchList(SearchWord.Text);

}

public async Task GetSotringList()

{

listprod.ItemsSource = await Products.SortingList();

}

public async Task GetSotringListMinus()

{

listprod.ItemsSource = await Products.SortingLisrMinus();

}

private void SearchBtn\_Click(object sender, RoutedEventArgs e)

{

GetSearchList();

}

private void Sort\_Click(object sender, RoutedEventArgs e)

{

GetSotringList();

}

private void Sort\_Copy\_Click(object sender, RoutedEventArgs e)

{

GetSotringListMinus();

}}}

<Page x:Class="BashShop.Pages.OrderPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:BashShop.Pages"

mc:Ignorable="d"

d:DesignHeight="550" d:DesignWidth="650"

Title="OrderPage">

<Grid Background="White">

<TextBox x:Name="NameProd" Text="{}" Background="#F4E8D0" IsEnabled="False" HorizontalAlignment="Left" Margin="85,50,0,0" VerticalAlignment="Top" Width="240" Height="40"/>

<TextBox x:Name="DesProd" Text="{}" Background="#F4E8D0" IsEnabled="False" HorizontalAlignment="Left" Margin="370,50,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="218" Height="129"/>

<TextBox x:Name="PriceProd" Text="{} за шт." Background="#F4E8D0" IsEnabled="False" HorizontalAlignment="Left" Margin="85,135,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="165" Height="38"/>

<Button Style="{DynamicResource ButtonStyle}" x:Name="AddOrderBtn" Height="34" Content="Добавить заказ" Background="#BEA36E" HorizontalAlignment="Center" VerticalAlignment="Top" Width="118" Click="AddOrderBtn\_Click" Margin="0,290,0,0"/>

<Button Style="{DynamicResource ButtonStyle}" x:Name="MinusBtn" BorderBrush="White" Content="-" FontSize="24" Background="#BEA36E" HorizontalAlignment="Left" Margin="54,210,0,0" VerticalAlignment="Top" Height="34" Width="35" Click="MinusBtn\_Click"/>

<Button Style="{DynamicResource ButtonStyle}" x:Name="PlusBtn" BorderBrush="White" Content="+" FontSize="24" Background="#BEA36E" HorizontalAlignment="Left" Margin="239,210,0,0" VerticalAlignment="Top" Height="35" Width="35" Click="PlusBtn\_Click"/>

<TextBlock x:Name="CountProd" HorizontalAlignment="Left" IsEnabled="False" Background="#F4E8D0" Margin="118,210,0,0" Text="1" FontWeight="Bold" FontSize="18" TextWrapping="Wrap" Height="35" TextAlignment="Center" VerticalAlignment="Top" Width="87"/>

<TextBox x:Name="PriceProds" Text="" Background="#F4E8D0" IsEnabled="False" HorizontalAlignment="Left" Margin="370,210,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="165" Height="38"/>

</Grid>

</Page>

using BashShop.Data;

using System;

using System.Windows;

using System.Windows.Controls;

namespace BashShop.Pages

{

/// <summary>

/// Логика взаимодействия для OrderPage.xaml

/// </summary>

public partial class OrderPage : Page

{

Products order;

public OrderPage(Products or)

{

InitializeComponent();

order = or;

NameProd.Text = or.Name;

DesProd.Text = or.Description;

PriceProd.Text = or.Price.ToString();

PriceProds.Text = or.Price.ToString();

}

private void AddOrderBtn\_Click(object sender, RoutedEventArgs e)

{

Orders or = new Orders(NameProd.Text, Convert.ToDouble(PriceProds.Text), DesProd.Text, Convert.ToInt32(CountProd.Text), App.user.Name, App.user.Phone);

Orders.AddOrder(or);

MessageBox.Show("Заказ отправлен");

}

private void MinusBtn\_Click(object sender, RoutedEventArgs e)

{

int count = Convert.ToInt32(CountProd.Text);

if (count == 1)

{

CountProd.Text = "1";

}

else

{

count--;

CountProd.Text = count.ToString();

PriceProds.Text = (count \* order.Price).ToString();

}}

private void PlusBtn\_Click(object sender, RoutedEventArgs e)

{

int count = Convert.ToInt32(CountProd.Text);

double pr = Convert.ToDouble(PriceProd.Text);

if (count >= 50)

{

CountProd.Text = "50";

}

else

{

count++;

CountProd.Text = count.ToString();

PriceProds.Text = (count \* pr).ToString();

}}}}

<Page x:Class="BashShop.Pages.HomePage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:BashShop.Pages"

mc:Ignorable="d"

d:DesignHeight="550" d:DesignWidth="650"

Title="HomePage">

<Grid Background="White">

<Image x:Name="UserImg" HorizontalAlignment="Left" Height="151" Margin="65,30,0,0" VerticalAlignment="Top" Width="174" Source="/img/profil.png" PreviewMouseDown="UserImg\_PreviewMouseDown"/>

<TextBox x:Name="FIOTB" Background="#F4E8D0" IsEnabled="False" HorizontalAlignment="Left" Margin="275,53,0,0" VerticalAlignment="Top" Width="170" Height="40"/>

<TextBox x:Name="PhoneLB" Text="{Binding Phone}" Background="#F4E8D0" IsEnabled="False" HorizontalAlignment="Left" Margin="275,106,0,0" VerticalAlignment="Top" Width="150"/>

<TextBox x:Name="PassLB" Text="{Binding Pass}" Background="#F4E8D0" IsEnabled="False" HorizontalAlignment="Left" Margin="275,140,0,0" VerticalAlignment="Top" Width="150"/>

<Button x:Name="UpdateProfilBtn" Style="{DynamicResource ButtonStyle}" Background="#BEA36E" Content="Редактировать профиль" Padding="10" HorizontalAlignment="Left" Margin="75,206,0,0" VerticalAlignment="Top" Click="UpdateProfilBtn\_Click"/>

<ListView x:Name="UsersLV" Height="250" Margin="182,275,75,25" SelectionChanged="UsersLV\_SelectionChanged">

<ListView.View>

<GridView>

<GridViewColumn DisplayMemberBinding="{Binding NameProd}" Header="Название"/>

<GridViewColumn DisplayMemberBinding="{Binding DesProd}" Header="Описание"/>

<GridViewColumn DisplayMemberBinding="{Binding Count}" Header="Кол-во"/>

<GridViewColumn DisplayMemberBinding="{Binding PriceProds}" Header="Цена"/>

</GridView>

</ListView.View>

</ListView>

<Label Content="Мои заказы" HorizontalAlignment="Left" Margin="375,225,0,0" VerticalAlignment="Top"/>

<Button x:Name="SaveUpdateBtn" Content="Сохранить" Style="{DynamicResource ButtonStyle}" Background="#BEA36E" HorizontalAlignment="Left" Margin="75,275,0,0" VerticalAlignment="Top" Height="34" Width="77" Click="SaveUpdateBtn\_Click"/>

</Grid>

</Page>

using BashShop.Data;

using BashShop.Windows;

using Microsoft.Win32;

using System;

using System.IO;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Input;

using System.Windows.Media.Imaging;

namespace BashShop.Pages

{

/// <summary>

/// Логика взаимодействия для HomePage.xaml

/// </summary>

public partial class HomePage : Page

{

public HomePage()

{

InitializeComponent();

SaveUpdateBtn.Visibility = Visibility.Hidden;

FIOTB.Text = App.user.Surname + " " + App.user.Name + " " + App.user.Patronic;

DataContext = App.user;

GetInfoOrderAsync();

}

public async Task GetInfoOrderAsync()

{

UsersLV.ItemsSource = await Orders.GetInfoOrder();

}

private void UpdateProfilBtn\_Click(object sender, RoutedEventArgs e)

{

FIOTB.IsEnabled = true;

PhoneLB.IsEnabled = true;

PassLB.IsEnabled = true;

SaveUpdateBtn.Visibility = Visibility.Visible;

}

private void SaveUpdateBtn\_Click(object sender, RoutedEventArgs e)

{

FIOTB.IsEnabled = false;

PhoneLB.IsEnabled = false;

PassLB.IsEnabled = false;

SaveUpdateBtn.Visibility = Visibility.Hidden;

string[] fio = FIOTB.Text.Split(' ');

App.user.Surname = fio[0];

App.user.Name = fio[1];

App.user.Patronic = fio[2];

App.user.Phone = PhoneLB.Text;

App.user.Pass = PassLB.Text;

Users.EditProfile();

}

private void UsersLV\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

DialogDelWindow dia = new DialogDelWindow();

if (dia.ShowDialog()==true)

{

var item = UsersLV.SelectedItem as Orders;

Orders.DeleteOrder(item);

MessageBox.Show("Заказ удален");

GetInfoOrderAsync();

}

}

private void UserImg\_PreviewMouseDown(object sender, MouseButtonEventArgs e)

{

OpenFileDialog openFileDialog = new OpenFileDialog();

openFileDialog.Filter = "Image files|\*.bmp;\*.jpg;\*.png|All files|\*.\*";

openFileDialog.FilterIndex = 1;

if (openFileDialog.ShowDialog() == true)

{

App.user.Photo = File.ReadAllBytes(openFileDialog.FileName);

BitmapImage image = new BitmapImage();

image.BeginInit();

image.UriSource = new Uri(openFileDialog.FileName);

image.EndInit();

UserImg.Source = image;

App.user.Update();

}}}}

<Window x:Class="BashShop.Windows.DialogDelWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:BashShop.Windows"

mc:Ignorable="d"

Title="DialogDelWindow" Height="250" Width="450">

<Grid>

<Button Style="{DynamicResource ButtonStyle}" Height="35" Content="Нет" Background="#BEA36E" FontSize="18" HorizontalAlignment="Left" Margin="284,137,0,0" VerticalAlignment="Top" Width="71" Click="Button\_Click\_1"/>

<Button Style="{DynamicResource ButtonStyle}" Height="35" Content="Да" Background="#BEA36E" FontSize="18" HorizontalAlignment="Left" Margin="81,137,0,0" VerticalAlignment="Top" Width="72" Click="Button\_Click"/>

<Label Content="Удалить заказ?" FontSize="24" HorizontalAlignment="Center" Margin="0,47,0,0" VerticalAlignment="Top"/>

</Grid>

</Window>

using System.Windows;

namespace BashShop.Windows

{

/// <summary>

/// Логика взаимодействия для DialogDelWindow.xaml

/// </summary>

public partial class DialogDelWindow : Window

{

public DialogDelWindow()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

this.DialogResult = true;

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

this.Close();

}}}

<Page x:Class="BashShop.APages.ACatalogPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:BashShop.APages"

mc:Ignorable="d"

d:DesignHeight="550" d:DesignWidth="650"

Title="ACatalogPage">

<Grid Background="White">

<ListView

Name="listprod" Margin="0,10,0,0" SelectionChanged="listprod\_SelectionChanged"

Background="#F4E8D0" BorderBrush="Transparent"

HorizontalContentAlignment="Center" Width="400"

HorizontalAlignment="Left">

<ListView.ItemTemplate>

<DataTemplate>

<Border Background="White" CornerRadius="10" Margin="10,0,0,10">

<Border.Effect>

<DropShadowEffect BlurRadius="30" Color="LightGray" />

</Border.Effect>

<Canvas Height="100" Width="350" HorizontalAlignment="Center">

<TextBlock

Text="{Binding Name}"

FontWeight="Bold"

FontSize="24"

Margin="10"

Width="300"

TextAlignment="Left"

Background="Transparent"/>

<TextBlock

Text="{Binding Description}"

FontSize="12"

Margin="10,40"

Width="300"

TextAlignment="Left"

TextWrapping="Wrap"

Background="Transparent"/>

<TextBlock

Text="{Binding Price}"

FontWeight="Bold"

FontSize="18"

Margin="0,70"

Width="300"

TextAlignment="Center"

Background="Transparent"/>

</Canvas>

</Border>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<Button Style="{DynamicResource ButtonStyle}" x:Name="AddProdBtn" Height="35" Content="Добавить товар" Background="#BEA36E" HorizontalAlignment="Left" Margin="507,223,0,0" VerticalAlignment="Top" Width="117" Click="AddProdBtn\_Click" />

<TextBox x:Name="SearchWord" HorizontalAlignment="Left" Margin="421,27,0,0" Background="#F4E8D0" TextWrapping="Wrap" VerticalAlignment="Top" Width="122" Height="32"/>

<Button Style="{DynamicResource ButtonStyle}" x:Name="SearchBtn" Height="32" Content="Искать" Background="#BEA36E" HorizontalAlignment="Left" Margin="566,27,0,0" VerticalAlignment="Top" Width="58" Click="SearchBtn\_Click"/>

<RadioButton x:Name="Sort" Content="Сортировка по возрастанию цены" HorizontalAlignment="Left" Margin="421,85,0,0" VerticalAlignment="Top" Height="25" Width="219" Click="Sort\_Click"/>

<RadioButton x:Name="Sort\_Copy" Content="Сортировка по убыванию цены" HorizontalAlignment="Left" Margin="421,115,0,0" VerticalAlignment="Top" Height="25" Width="219" Click="Sort\_Copy\_Click"/>

</Grid>

</Page>

using BashShop.Data;

using BashShop.Windows;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

namespace BashShop.APages

{

/// <summary>

/// Логика взаимодействия для ACatalogPage.xaml

/// </summary>

public partial class ACatalogPage : Page

{

public ACatalogPage()

{

InitializeComponent();

GetInfoProdAsync();

}

public async Task GetInfoProdAsync()

{

listprod.ItemsSource = await Products.GetProd();

}

private void AddProdBtn\_Click(object sender, RoutedEventArgs e)

{

AddProdWindow apw = new AddProdWindow();

apw.Show();

}

private void listprod\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

var lstp = listprod.SelectedItem as Products;

DialogDelOrEditWindow ddoew = new DialogDelOrEditWindow(lstp);

ddoew.Show();

}

public async Task GetSearchList()

{

listprod.ItemsSource = await Products.SearchList(SearchWord.Text);

}

public async Task GetSotringList()

{

listprod.ItemsSource = await Products.SortingList();

}

public async Task GetSotringListMinus()

{

listprod.ItemsSource = await Products.SortingLisrMinus();

}

private void SearchBtn\_Click(object sender, RoutedEventArgs e)

{

GetSearchList();

}

private void Sort\_Click(object sender, RoutedEventArgs e)

{

GetSotringList();

}

private void Sort\_Copy\_Click(object sender, RoutedEventArgs e)

{

GetSotringListMinus();

}}}

<Window x:Class="BashShop.APages.AddProdWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:BashShop.APages"

mc:Ignorable="d"

Title="AddProdWindow" Height="350" Width="550">

<Grid>

<TextBox x:Name="NameProd" Background="#F4E8D0" HorizontalAlignment="Left" Margin="59,55,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="143" Height="38"/>

<Label Content="Название" FontWeight="Bold" HorizontalAlignment="Left" Margin="58,30,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="DesProd" Background="#F4E8D0" HorizontalAlignment="Left" Margin="302,55,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="183" Height="121"/>

<Label Content="Описание" FontWeight="Bold" HorizontalAlignment="Left" Margin="301,30,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="PriceProd" Background="#F4E8D0" HorizontalAlignment="Left" Margin="60,132,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="143" Height="38"/>

<Label Content="Цена" FontWeight="Bold" HorizontalAlignment="Left" Margin="59,107,0,0" VerticalAlignment="Top"/>

<Button Style="{DynamicResource ButtonStyle}" x:Name="AddProdBtn" Height="35" Content="Добавить товар" Background="#BEA36E" HorizontalAlignment="Center" Margin="0,250,0,0" VerticalAlignment="Top" Width="118" Click="AddProdBtn\_Click" />

</Grid>

</Window>

using BashShop.Data;

using System;

using System.Windows;

namespace BashShop.APages

{

/// <summary>

/// Логика взаимодействия для AddProdWindow.xaml

/// </summary>

public partial class AddProdWindow : Window

{

public AddProdWindow()

{

InitializeComponent();

}

private void AddProdBtn\_Click(object sender, RoutedEventArgs e)

{

if (NameProd.Text == "" || DesProd.Text == "" || PriceProd.Text == "")

{

MessageBox.Show("Введите все данные");

}

else

{

Products pr = new Products(NameProd.Text, DesProd.Text, Convert.ToDouble(PriceProd.Text));

Products.AddProd(pr);

MessageBox.Show("Товар добавлен");

NameProd.Text = "";

DesProd.Text = "";

PriceProd.Text = "";

}}}}

<Window x:Class="BashShop.Windows.DialogDelOrEditWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:BashShop.Windows"

mc:Ignorable="d"

Title="DialogDelOrEditWindow" Height="250" Width="500">

<Grid>

<Button Style="{DynamicResource ButtonStyle}" x:Name="DeleteProd" Height="34" Content="Удалить" Background="#BEA36E" HorizontalAlignment="Left" VerticalAlignment="Center" Width="118" Margin="94,0,0,0" Click="DeleteProd\_Click"/>

<Button Style="{DynamicResource ButtonStyle}" x:Name="EditProd" Height="34" Content="Редактировать" Background="#BEA36E" HorizontalAlignment="Left" VerticalAlignment="Center" Width="118" Margin="294,0,0,0" Click="EditProd\_Click"/>

</Grid>

</Window>

using BashShop.Data;

using System.Windows;

namespace BashShop.Windows

{

/// <summary>

/// Логика взаимодействия для DialogDelOrEditWindow.xaml

/// </summary>

public partial class DialogDelOrEditWindow : Window

{

public Products products;

public DialogDelOrEditWindow(Products prod)

{

InitializeComponent();

products = prod;

}

private void DeleteProd\_Click(object sender, RoutedEventArgs e)

{

Products.DeleteProd(products);

MessageBox.Show("Товар удален");

this.Close();

}

private void EditProd\_Click(object sender, RoutedEventArgs e)

{

EditWindow apw = new EditWindow(products);

apw.Show();

}}}

<Window x:Class="BashShop.Windows.EditWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:BashShop.Windows"

mc:Ignorable="d"

Title="EditWindow" Height="350" Width="550">

<Grid>

<Button Style="{DynamicResource ButtonStyle}" x:Name="EdirProd" Height="35" Content="Сохранить изменения" Background="#BEA36E" HorizontalAlignment="Left" Margin="355,250,0,0" VerticalAlignment="Top" Width="130" Click="EdirProd\_Click" />

<TextBox x:Name="NameProd" Background="#F4E8D0" HorizontalAlignment="Left" Margin="59,55,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="143" Height="38"/>

<Label Content="Название" FontWeight="Bold" HorizontalAlignment="Left" Margin="58,30,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="DesProd" Background="#F4E8D0" HorizontalAlignment="Left" Margin="302,55,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="183" Height="121"/>

<Label Content="Описание" FontWeight="Bold" HorizontalAlignment="Left" Margin="301,30,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="PriceProd" Background="#F4E8D0" HorizontalAlignment="Left" Margin="60,132,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="143" Height="38"/>

<Label Content="Цена" FontWeight="Bold" HorizontalAlignment="Left" Margin="59,107,0,0" VerticalAlignment="Top"/>

</Grid>

</Window>

using BashShop.Data;

using System;

using System.Windows;

namespace BashShop.Windows

{

/// <summary>

/// Логика взаимодействия для EditWindow.xaml

/// </summary>

public partial class EditWindow : Window

{

Products products;

public EditWindow(Products prod)

{

InitializeComponent();

products = prod;

NameProd.Text = prod.Name;

PriceProd.Text = prod.Price.ToString();

DesProd.Text = prod.Description;

}

private void EdirProd\_Click(object sender, RoutedEventArgs e)

{

App.products.Name = NameProd.Text;

App.products.Price = Convert.ToInt32(PriceProd.Text);

App.products.Description = DesProd.Text;

Products.EditProd();

}}}

<Page x:Class="BashShop.APages.AHomePage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:BashShop.APages"

mc:Ignorable="d"

d:DesignHeight="550" d:DesignWidth="650"

Title="AHomePage">

<Grid Background="White">

<ListView x:Name="AllOrdersLV" SelectionChanged="AllOrdersLV\_SelectionChanged" Margin="32,100,32,100">

<ListView.View>

<GridView>

<GridViewColumn DisplayMemberBinding="{Binding NameProd}" Header="Название"/>

<GridViewColumn DisplayMemberBinding="{Binding DesProd}" Header="Описание"/>

<GridViewColumn DisplayMemberBinding="{Binding Count}" Header="Кол-во"/>

<GridViewColumn DisplayMemberBinding="{Binding PriceProds}" Header="Цена"/>

<GridViewColumn DisplayMemberBinding="{Binding NameUser}" Header="Имя заказчика"/>

<GridViewColumn DisplayMemberBinding="{Binding PhoneUser}" Header="Телефон"/>

</GridView>

</ListView.View>

</ListView>

<Label Content="Заказы" FontSize="18" HorizontalAlignment="Center" Margin="0,56,0,0" VerticalAlignment="Top"/>

</Grid>

</Page>

using BashShop.Data;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

namespace BashShop.APages

{

/// <summary>

/// Логика взаимодействия для AHomePage.xaml

/// </summary>

public partial class AHomePage : Page

{

public AHomePage()

{

InitializeComponent();

GetInfoAllOrderAsynk();

}

public async Task GetInfoAllOrderAsynk()

{

AllOrdersLV.ItemsSource = await Orders.GetInfoAllOrders();

}

private void AllOrdersLV\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

var item = AllOrdersLV.SelectedItem as Orders;

Orders.DeleteOrder(item);

MessageBox.Show("Заказ удален");

GetInfoAllOrderAsynk();

}}}

# ПРИЛОЖЕНИЕ C

Перечень файлов на диске:

* BashShop – программа “Магазин перегонных аппаратов”;
* Отчёт по курсовой.docх – пояснительная записка к проекту;
* Users.json – файл для базы данных к данному продукту;
* Products.json – файл для базы данных к данному продукту;
* Orders.json – файл для базы данных к данному продукту;
* Reviews.json – файл для базы данных к данному продукту;