Министерство образования Московской области

Государственное образовательное учреждение высшего образования Московской области

«Государственный гуманитарно-технологический университет»

**Ликино-Дулевский политехнический колледж – филиал ГГТУ**

**КУРСОВОЙ ПРОЕКТ**

Тема: Разработка программного модуля «Интернет-магазин электроники»

МДК 01.01 «Разработка программных модулей»

**Выполнил:**

Широнин Николай Анатольевич

студент группы ИСП.20А

09.02.07 Информационные системы и программирование

очной формы обучения

**Руководитель:**

Кузьмина Елена Евгеньевна­­­­­­­­­­­­­­­­­­

Оценка

\_\_\_\_\_\_\_\_\_/\_\_\_­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Подпись руководителя

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ликино-Дулёво

2023 год

Оглавление

[Введение 2](#_Toc131693636)

[1. Назначение разработки 2](#_Toc131693637)

[2. Требования к программному модулю 3](#_Toc131693638)

[2. 1 Требования к составу и параметрам технических средств: 3](#_Toc131693639)

[2. 2 Требования к информационной и программной совместимости: 3](#_Toc131693640)

[3. Руководство программиста 3](#_Toc131693641)

[3. 1 Разработка базы данных: 3](#_Toc131693642)

[3. 2 Разработка программного модуля: 10](#_Toc131693643)

[3. 2 Текст программы: 18](#_Toc131693644)

[4. Руководство пользователя 18](#_Toc131693645)

[4. 1 Выполнение программного модуля: 25](#_Toc131693646)

[4. 2 Сообщения пользователю: 26](#_Toc131693647)

[5. Тестирование программного модуля 28](#_Toc131693648)

[5. 1 Модульное тестирование 28](#_Toc131693649)

[5. 2 Тестовые сценарии 29](#_Toc131693650)

[6. Отладка программного модуля 30](#_Toc131693651)

[Заключение 31](#_Toc131693652)

[Список литературы 32](#_Toc131693653)

[Ссылки на источники 32](#_Toc131693654)

[Приложения 33](#_Toc131693655)

**Введение**

WPF — это сокращение от Windows Presentation Foundation, технологии, используемой для разработки графических интерфейсов пользователя (GUI) для приложений Windows. WPF была введена вместе с .NET Framework 3.0 и предоставляет разработчикам более широкие возможности для создания более интерактивных и более эстетически привлекательных пользовательских интерфейсов. WPF использует разметку XAML для определения пользовательского интерфейса и поддерживает различные функции, такие как векторная графика, анимация, прозрачность и трансформации. WPF также имеет множество встроенных элементов управления, таких как кнопки, списки и таблицы, которые могут быть настроены для удовлетворения конкретных потребностей приложения.

C# (произносится "си шарп") — это современный, объектно-ориентированный язык программирования, разработанный компанией Microsoft в рамках платформы .NET. Он был создан с целью быть простым, эффективным и легким в использовании, при этом достаточно мощным для создания сложных приложений. C# широко используется для разработки приложений для Windows, веб-приложений, игр и мобильных приложений для устройств на iOS и Android. Он также используется для программирования бэкэнда, такого как разработка веб-сервисов и API. Синтаксис C# похож на Java и C++, что упрощает изучение языка для разработчиков, знакомых с этими языками.

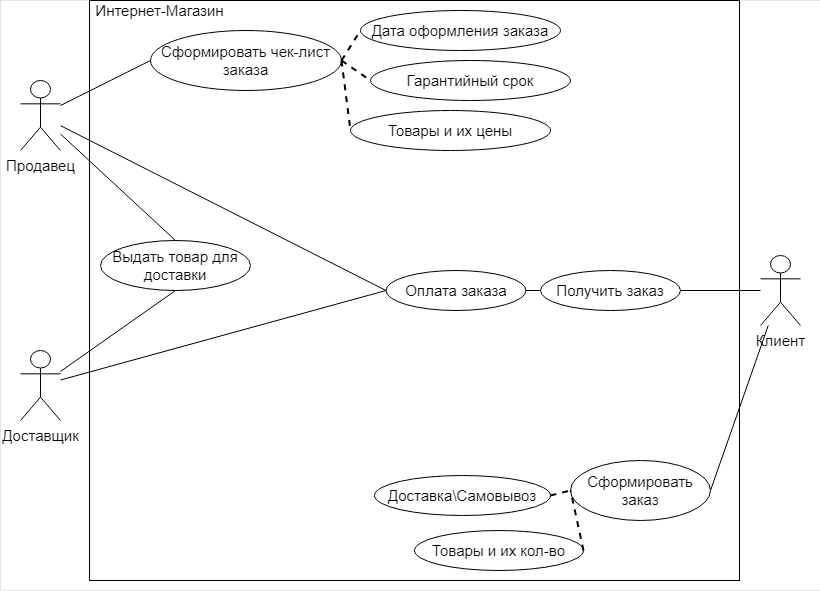
Интернет-магазины — это онлайн-платформы, где товары и услуги продаются через интернет. Они предоставляют потребителям возможность просматривать каталоги товаров, выбирать и заказывать товары, оплачивать их и получать доставку без необходимости посещать физический магазин. Интернет-магазины могут специализироваться на определенных товарных группах, таких как электроника, одежда, косметика, продукты питания и т.д. Они также могут предлагать различные услуги, такие как консультации, сборка и установка товаров, гарантийное обслуживание и т.д. Интернет-магазины стали очень популярными в последние годы, благодаря удобству и доступности покупок через интернет, а также возможности сравнивать цены и выбирать лучшие предложения.

1. **Назначение разработки**

Разработка проводится на основании приказа о курсовом проектировании.

Приложение должно выполнять следующие функции:

* Просмотр всех заказов из всех магазинов.
* Добавления и удаление Заказов.
* Поиск Заказа по критериям.
* Вывод отчёта в Excel
* Вывод чека в Word



Диаграмма№1 «use case»

1. **Требования к программному модулю**

## **2. 1 Требования к составу и параметрам технических средств:**

Таблица№1 «Минимальные характеристики компьютера»

|  |  |
| --- | --- |
| Процессор | i5 6600 |
| Память | От 4 ГБ |
| Разрешение экрана | От 1920x1080 |
| Устройства ввода | Клавиатура, мышь |
|  |  |

Таблица№2 «Оптимальные характеристики компьютера»

|  |  |
| --- | --- |
| Процессор | Ryzen 7 2700 |
| Память | От 8 ГБ |
| Разрешение экрана | От 1920x1080 |
| Устройства ввода | Клавиатура, мышь |

**2. 2 Требования к информационной и программной совместимости:**

Для корректной работы программы необходимо: иметь установленные приложения и программы: Visual Studio 2019-2022, Microsoft SQL Server Management Studio, MS Excel, MS Word.

1. **Руководство программиста**

## **3. 1 Разработка базы данных:**

В данной работе используется база данных «SQL в среде Microsoft SQL server management studio»

В базе данных существуют 10 связанных таблицы

Таблица Client состоит id клиента его ФИО и Телефонного номера.

Таблица№3 «Client»

|  |  |  |  |
| --- | --- | --- | --- |
| **Client** | | | |
| Поле | Тип | Атрибуты | Описание |
| ClientID | int | PK NN | Уникальный номер Клиента |
| FIO | Varchar (MAX) | AN | Фамилия, Имя, Отчество Клиента |
| PhoneNumber | Varchar (MAX) | AN | Телефонный номер Клиента |

Таблица WayToGet это таблица создана для выбора типа получения заказа, то есть Доставка или Самовывоз состоит из id и типа.

Таблица№4 «WayToGet»

|  |  |  |  |
| --- | --- | --- | --- |
| **WayToGet** | | | |
| Поле | Тип | Атрибуты | Описание |
| WayToGetID | int | PK NN | Уникальный номер типа получения |
| Type | Varchar (MAX) | AN | Тип получения |

Таблица Store хранит в себе id магазина, его электронную почту, адрес, уникальный код, и Фотографию Магазина.

Таблица№5 «Store»

|  |  |  |  |
| --- | --- | --- | --- |
| **Store** | | | |
| Поле | Тип | Атрибуты | Описание |
| StoreID | int | PK NN | Уникальный номер Магазина |
| Email | Varchar (MAX) | AN | Электронная почта Магазина |
| Adress | Varchar (MAX) | AN | Адрес Магазина |
| Code | int | AN | Уникальный код Магазина |
| Picture | Varchar (MAX) | AN | Путь к фотографии магазина |

Таблица Status Данная таблица служит для определения статуса Заказа она состоит из id и называния статуса.

Таблица№6 «Status»

|  |  |  |  |
| --- | --- | --- | --- |
| **Status** | | | |
| Поле | Тип | Атрибуты | Описание |
| StatusID | int | PK NN | Уникальный номер Статуса |
| SName | Varchar (MAX) | AN | Название Статуса |

Таблица Category нужна для определения категории товара состоит из id и названия категории

Таблица№7 «Category»

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | | | |
| Поле | Тип | Атрибуты | Описание |
| CategoryID | int | PK NN | Уникальный номер Категории |
| CName | Varchar (MAX) | AN | Название Категории |

Таблица Goods хранит в себе товары, состоит из id, Названия товара, Фирмы, Модели, Категории, Характеристик, Цены, Помесячной гарантии и Фотографии товара.

Таблица№8 «Goods»

|  |  |  |  |
| --- | --- | --- | --- |
| **Goods** | | | |
| Поле | Тип | Атрибуты | Описание |
| GoodsID | int | PK NN | Уникальный номер Товара |
| GName | Varchar (MAX) | AN | Название Товара |
| Firm | Varchar (MAX) | AN | Фирма Товара |
| Model | Varchar (MAX) | AN | Модель Товара |
| CategoryID | int | AN | Категория Товара |
| Specifications | Varchar (MAX) | AN | Характеристики Товара |
| Price | int | AN | Цена Товара |
| WarrantyPerMonth | int | AN | Гарантия Товара (помесячная) |
| GImage | Varchar (MAX) | AN | Фото Товара |

Таблица DeliveryEmployee хранит ФИО доставщиков и состоит из id и ФИО.

Таблица№9 «DeliveryEmployee»

|  |  |  |  |
| --- | --- | --- | --- |
| **DeliveryEmployee** | | | |
| Поле | Тип | Атрибуты | Описание |
| DeliveryEmployeeID | int | PK NN | Уникальный номер Доставшика |
| FIOEmployee | Varchar (MAX) | AN | ФИО Доставщика |

Таблица Delivery хранит адреса, по которым была произведена доставка состоит из id, адреса и ФИО доставщика.

Таблица№10 «Delivery»

|  |  |  |  |
| --- | --- | --- | --- |
| **Delivery** | | | |
| Поле | Тип | Атрибуты | Описание |
| DeliveryID | int | PK NN | Уникальный номер Доставки |
| Adress | Varchar (MAX) | AN | Адрес клиента |
| DeliveryEmployeeID | int | AN | ФИО доставщика |

Таблица Orders в таблице хранится информация кто в каком магазине купил товар и каким образом, состоит из id, id магазина, способу получения, доставка (если не самовывоз), Клиент, Статус заказа и время реализации заказа.

Таблица№11 «Orders»

|  |  |  |  |
| --- | --- | --- | --- |
| **Orders** | | | |
| Поле | Тип | Атрибуты | Описание |
| OrdersID | int | PK NN | Уникальный номер заказа |
| StoreID | int | AN | Уникальный номер магазина |
| WayToGetID | Int | AN | Уникальный номер типа получения |
| DeliveryID | Int | AN | Уникальный номер доставки |
| ClientID | Int | AN | Уникальный номер клиента |
| StatusID | int | AN | Уникальный номер статуса |
| DateTime | datetime | AN | Дата реализации заказа |

Таблица OrdersGoods сопоставляет заказы с товарами состоит из id, заказа, товара и его количества.

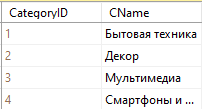
Таблица№10 «OrdersGoods»

|  |  |  |  |
| --- | --- | --- | --- |
| **Delivery** | | | |
| Поле | Тип | Атрибуты | Описание |
| id | int | PK NN | Уникальный номер OrdersGoods |
| OrdersID | int | AN | Уникальный номер заказа |
| GoodsID | int | AN | Уникальный номер товара |
| Quantity | int | AN | Количество товара |

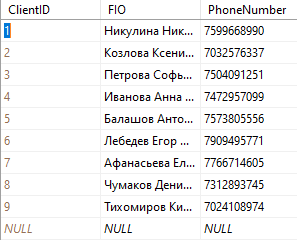
Для соединения таблиц использовались следующие типы связей:

 - один ко многим вида: not-**identifying relationship**

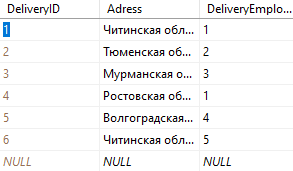
Заполнение базы данных:



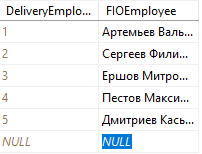
Рисунок№1 «Category»



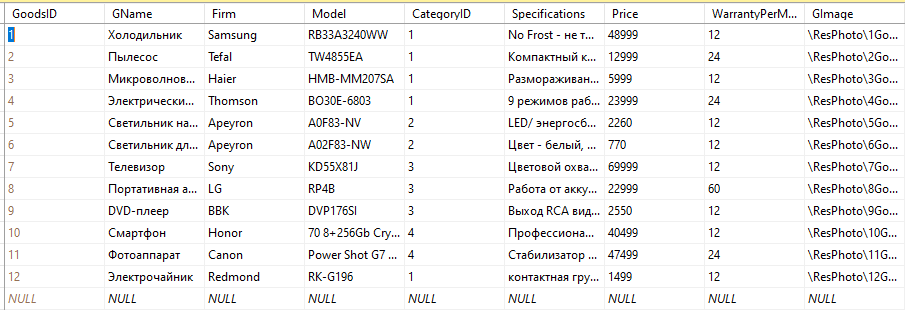
Рисунок№2 «Client»



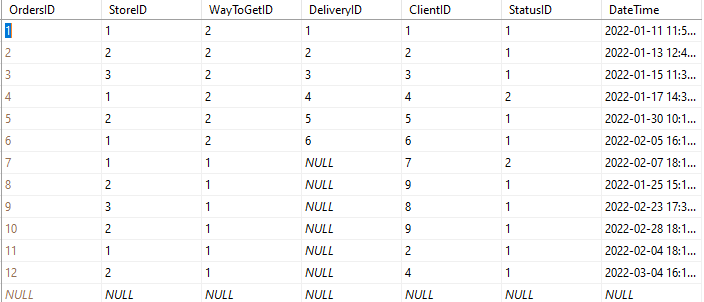
Рисунок№3 «Delivery»



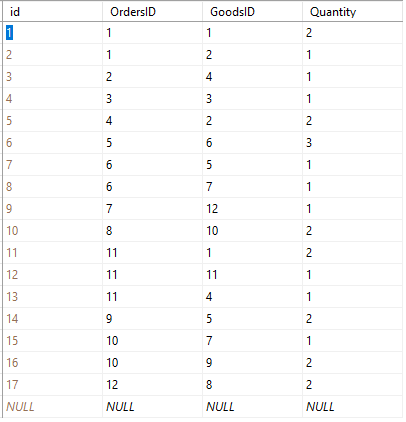
Рисунок№4 «DeliveryEmloyee»



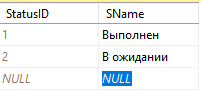
Рисунок№5 «Goods»



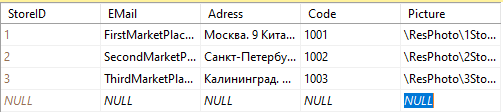
Рисунок№6 «Orders»



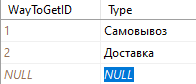
Рисунок№7 «OrdersGoods»



Рисунок№8 «Status»



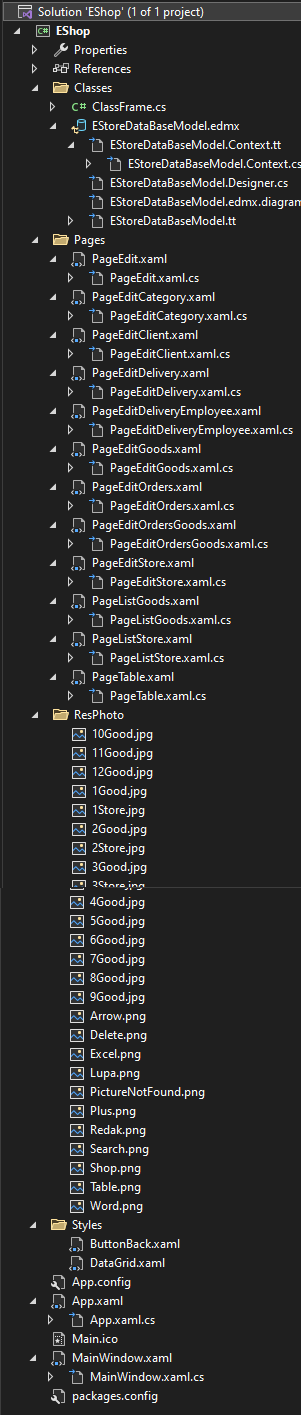
Рисунок№9 «Store»



Рисунок№10 «WayToGet»

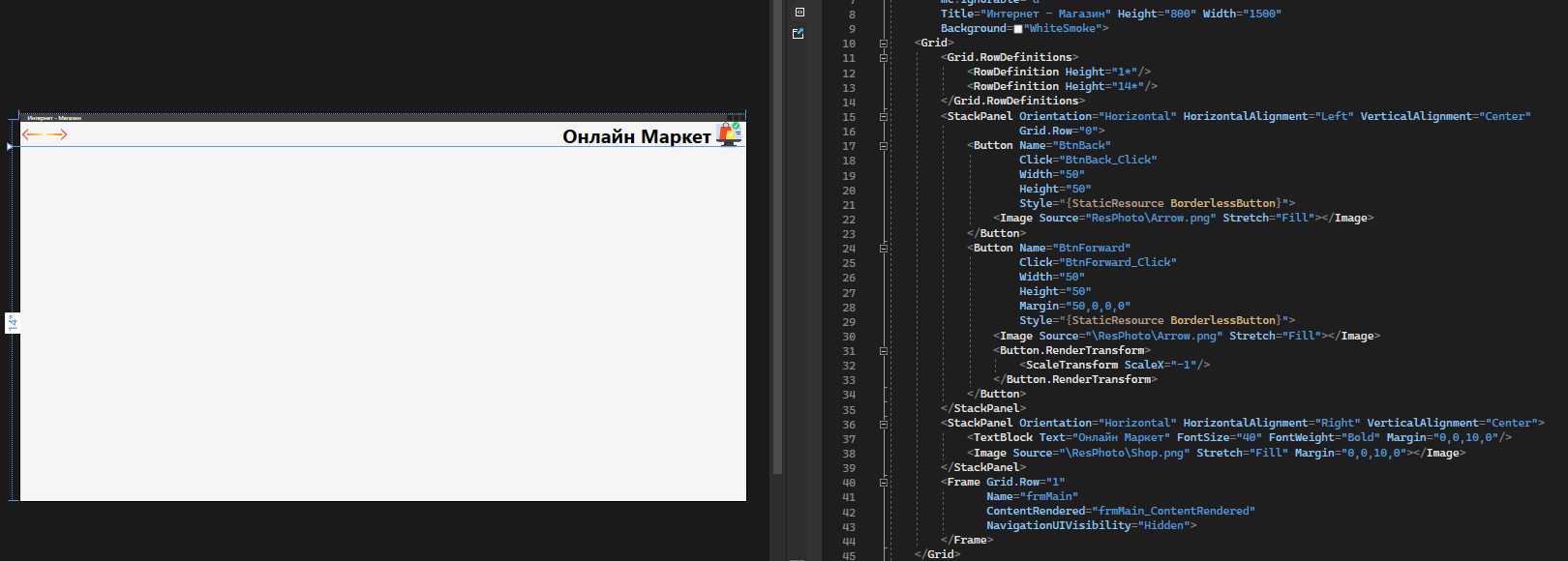
## **3. 2 Разработка программного модуля:**

Структура проекта для приложения выглядит так:



Рисунок№11 «Структура»

**Main Window.xaml (Содержит Заголовок)**



Рисунок№11 «Заголовок»

**Main Window.xaml.cs:**

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

ClassFrame.frmObj = frmMain;

frmMain.Navigate(new Pages.PageTable());

}

private void frmMain\_ContentRendered(object sender, EventArgs e)

{

if (frmMain.CanGoBack)

{

BtnBack.Visibility = Visibility.Visible;

}

else

{

BtnBack.Visibility = Visibility.Hidden;

}

if (frmMain.CanGoForward)

{

BtnForward.Visibility = Visibility.Visible;

}

else

{

BtnForward.Visibility = Visibility.Hidden;

}

}

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

{

frmMain.GoBack();

}

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

{

frmMain.GoForward();

}

}

**App.Xaml (Содержит ссылки на стили):**

[**Код**](#APPXAML)

В приложении было создано четыре основных папки: Classes, Pages, ResPhoto, Styles. В папке ResPhoto хранятся Фотографии Товаров, Магазинов и иконки для программы.

В папке Styles хранятся стили:

**ButtonBack.xaml (стиль для кнопок обратно и вперед):**

[Код](#ButtonBack)

**DataGrid.xaml (Стиль для таблицы):**

[Код](#DataGrid)

Папка Classes содержит в себе:

**ClassFrame.cs (Для загрузки страниц):**

internal class ClassFrame

{

public static Frame frmObj;

}

**EStoreDataBaseModel.edmx (Подключенная база данных). EntityFramework — это набор технологий в ADO.NET, которые поддерживают разработку программных приложений, ориентированных на данные.**

public partial class EStoreEntities : DbContext

{

private static EStoreEntities \_Context;

public EStoreEntities()

: base("name=EStoreEntities")

{

}

public static EStoreEntities GetContext()

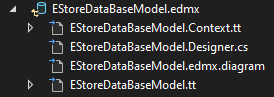
{

if (\_Context == null)

\_Context = new EStoreEntities();

return \_Context;

}



Рисунок№13 «Модель базы данных»

**Папка Pages содержит в себе (Для хранения страниц):**

**PageTable.xaml**

[Код](#PageTable)

**PageTabel.xaml.cs:**

using EShop.Classes;

using Microsoft.Office.Interop.Excel;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Button = System.Windows.Controls.Button;

using Excel = Microsoft.Office.Interop.Excel;

using Page = System.Windows.Controls.Page;

using Word = Microsoft.Office.Interop.Word;

public partial class PageTable : Page

{

private EStoreEntities \_context = new EStoreEntities();

public PageTable()

{

InitializeComponent();

DGridOrders.ItemsSource = EStoreEntities.GetContext().OrdersGoods.ToList();

OrderSelect.Text = Convert.ToInt32(OrderSelect.Text+1).ToString();

}

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

{

ClassFrame.frmObj.Navigate(new PageEdit((sender as Button).DataContext as OrdersGoods));

}

**Поиск по ФИО Клиента:**

private void TxtSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

DGridOrders.ItemsSource = EStoreEntities.GetContext().OrdersGoods.Where(x => x.Orders.Client.FIO.ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

}

**Очистка параметров поиска:**

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

{

TxtSearch.Text = null;

DGridOrders.ItemsSource = EStoreEntities.GetContext().OrdersGoods.ToList();

}

**Кнопки для навигации по вкладкам добавлений:**

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

{

ClassFrame.frmObj.Navigate(new PageEditCategory(null));

}

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

{

ClassFrame.frmObj.Navigate(new PageEditClient(null));

}

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

{

ClassFrame.frmObj.Navigate(new PageEditDelivery(null));

}

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

{

ClassFrame.frmObj.Navigate(new PageEditDeliveryEmployee(null));

}

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

{

ClassFrame.frmObj.Navigate(new PageEditGoods(null));

}

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

{

ClassFrame.frmObj.Navigate(new PageEditOrders(null));

}

private void BtnToOrders\_Goods\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageEditOrdersGoods(null));

}

private void BtnToStoreClick(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageEditStore(null));

}

**Кнопка для удаления записи:**

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

{

var OrdersForRemoving = DGridOrders.SelectedItems.Cast<OrdersGoods>().ToList();

if (MessageBox.Show($"Удалить {OrdersForRemoving.Count()} запись?",

"Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

try

{

EStoreEntities.GetContext().OrdersGoods.RemoveRange(OrdersForRemoving);

EStoreEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

DGridOrders.ItemsSource = EStoreEntities.GetContext().OrdersGoods.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

**Кнопка для навигации страниц просмотров:**

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

{

ClassFrame.frmObj.Navigate(new PageListGoods());

}

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

{

ClassFrame.frmObj.Navigate(new PageListStore());

}

**Кнопка для вывода отчёта в Excell:**

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

{

var app = new Excel.Application();

Excel.Workbook wb = app.Workbooks.Add();

Excel.Worksheet worksheet = app.Worksheets.Item[1];

int indexRows = 1;

worksheet.Cells[1][indexRows] = "Номер Заказа";

worksheet.Cells[2][indexRows] = "Адресс магазина";

worksheet.Cells[3][indexRows] = "Почта";

worksheet.Cells[4][indexRows] = "Категория товара";

worksheet.Cells[5][indexRows] = "Название товара";

worksheet.Cells[6][indexRows] = "Цена";

worksheet.Cells[7][indexRows] = "Кол-во";

worksheet.Cells[8][indexRows] = "Тип получения";

worksheet.Cells[9][indexRows] = "ФИО клиента";

worksheet.Cells[10][indexRows] = "Номер клиента";

worksheet.Cells[11][indexRows] = "Статус заказа";

worksheet.Cells[12][indexRows] = "Дата Реализации заказа";

var printItems = EStoreEntities.GetContext().OrdersGoods.ToList();

foreach (var item in printItems)

{

worksheet.Cells[1][indexRows + 1] = item.Orders.OrdersID;

worksheet.Cells[2][indexRows + 1] = item.Orders.Store.Adress;

worksheet.Cells[3][indexRows + 1] = item.Orders.Store.EMail;

worksheet.Cells[4][indexRows + 1] = item.Goods.Category.CName;

worksheet.Cells[5][indexRows + 1] = item.Goods.GName;

worksheet.Cells[6][indexRows + 1] = item.Goods.Price.ToString();

worksheet.Cells[7][indexRows + 1] = item.Quantity.ToString();

worksheet.Cells[8][indexRows + 1] = item.Orders.WayToGet.Type;

worksheet.Cells[9][indexRows + 1] = item.Orders.Client.FIO;

worksheet.Cells[10][indexRows + 1] = item.Orders.Client.PhoneNumber.ToString();

worksheet.Cells[11][indexRows + 1] = item.Orders.Status.SName;

worksheet.Cells[12][indexRows + 1] = item.Orders.DateTime.ToString();

indexRows++;

}

worksheet.Cells[13][3].Formula = "Кол-во Проданного товара =";

worksheet.Cells[14][3].Formula = $"=SUM(G2:G{indexRows})";

worksheet.Cells[13][4].Formula = "Прибыль за все товары =";

worksheet.Cells[14][4].Formula = $"=SUM(F2:F{indexRows})";

Excel.Range rangeBorders = worksheet.Range[worksheet.Cells[12][indexRows], worksheet.Cells[1][1]];

rangeBorders.Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle =

rangeBorders.Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle =

rangeBorders.Borders[Excel.XlBordersIndex.xlEdgeRight].LineStyle =

rangeBorders.Borders[Excel.XlBordersIndex.xlEdgeTop].LineStyle =

rangeBorders.Borders[Excel.XlBordersIndex.xlInsideHorizontal].LineStyle =

rangeBorders.Borders[Excel.XlBordersIndex.xlInsideVertical].LineStyle = Excel.XlLineStyle.xlContinuous;

worksheet.Columns.AutoFit();

app.Visible = true;

}

**Кнопка для вывода чека в Word:**

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

{

var Select = Convert.ToInt32(OrderSelect.Text);

var allOrders = \_context.OrdersGoods.Where(x => x.OrdersID == Select).ToList();

var application = new Word.Application();

Word.Document document = application.Documents.Add();

Word.Paragraph orderParagrapth12 = document.Paragraphs.Add();

Word.Range orderRange1 = orderParagrapth12.Range;

foreach (var currentGoods in allOrders)

{

orderRange1.Text = "Чек по заказу №" + currentGoods.Orders.OrdersID.ToString();

orderParagrapth12.set\_Style("Заголовок");

}

orderRange1.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

orderRange1.InsertParagraphAfter();

Word.Paragraph orderParagrapth = document.Paragraphs.Add();

Word.Range orderRange = orderParagrapth.Range;

foreach (var currentGoods in allOrders)

{

orderRange.Text = currentGoods.Orders.Client.FIO;

orderParagrapth.set\_Style("Заголовок");

}

orderRange.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

orderRange.InsertParagraphAfter();

Word.Paragraph tableParagraph = document.Paragraphs.Add();

Word.Range tableRange = tableParagraph.Range;

Word.Table paymentsTable = document.Tables.Add(tableRange, allOrders.Count() + 1, 7);

paymentsTable.Borders.InsideLineStyle = paymentsTable.Borders.OutsideLineStyle = Word.WdLineStyle.wdLineStyleSingle;

paymentsTable.Range.Cells.VerticalAlignment = Word.WdCellVerticalAlignment.wdCellAlignVerticalCenter;

Word.Range cellRange;

cellRange = paymentsTable.Cell(1, 1).Range;

cellRange.Text = "Фото";

cellRange = paymentsTable.Cell(1, 2).Range;

cellRange.Text = "Название";

cellRange = paymentsTable.Cell(1, 3).Range;

cellRange.Text = "Фирма";

cellRange = paymentsTable.Cell(1, 4).Range;

cellRange.Text = "Модель";

cellRange = paymentsTable.Cell(1, 5).Range;

cellRange.Text = "Цена, Р";

cellRange = paymentsTable.Cell(1, 6).Range;

cellRange.Text = "Кол-во";

cellRange = paymentsTable.Cell(1, 7).Range;

cellRange.Text = "Итог";

paymentsTable.Rows[1].Range.Bold = 1;

paymentsTable.Rows[1].Range.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

for (int i = 0; i < allOrders.Count(); i++)

{

var currentGood = allOrders[i];

cellRange = paymentsTable.Cell(i + 2, 1).Range;

Word.InlineShape imageShape = cellRange.InlineShapes.AddPicture(AppDomain.CurrentDomain.BaseDirectory + "..\\..\\" + currentGood.Goods.GImage);

imageShape.Width = imageShape.Height = 40;

cellRange.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

cellRange = paymentsTable.Cell(i + 2, 2).Range;

cellRange.Text = currentGood.Goods.GName;

cellRange = paymentsTable.Cell(i + 2, 3).Range;

cellRange.Text = currentGood.Goods.Firm;

cellRange = paymentsTable.Cell(i + 2, 4).Range;

cellRange.Text = currentGood.Goods.Model;

cellRange = paymentsTable.Cell(i + 2, 5).Range;

cellRange.Text = currentGood.Goods.Price.ToString();

cellRange = paymentsTable.Cell(i + 2, 6).Range;

cellRange.Text = currentGood.Quantity.ToString();

cellRange = paymentsTable.Cell(i + 2, 7).Range;

cellRange.Text = (currentGood.Goods.Price \* currentGood.Quantity).ToString();

}

Word.Paragraph FinalPriceParagraph = document.Paragraphs.Add();

Word.Range FinalPriceRange = FinalPriceParagraph.Range;

int sum = 0;

for (int i = 0; i < allOrders.Count(); i++)

{

var currentGoodss = allOrders[i];

sum += Convert.ToInt32(allOrders[i].Goods.Price) \* Convert.ToInt32(allOrders[i].Quantity);

}

FinalPriceRange.Text = "Итоговая цена: " + (sum).ToString() + "руб.";

FinalPriceParagraph.set\_Style("Заголовок");

Word.Paragraph FinalPriceParagraph1 = document.Paragraphs.Add();

Word.Range FinalPriceRange1 = FinalPriceParagraph.Range;

foreach (var currentGoodsss in allOrders)

{

FinalPriceRange1.Text = "Дата реализация заказа: " + currentGoodsss.Orders.DateTime.ToString();

orderParagrapth.set\_Style("Заголовок");

}

orderRange.InsertParagraphAfter();

application.Visible = true;

}

}

**PageEdit.xaml**

[Код](#PageEdit)

**PageEditCategory.xaml**

[Код](#PageEditCategory)

**PageEditClient.xaml**

[Код](#PageEditClient)

**PageEditDelivery.xaml**

[Код](#PageEditDelivery)

**PageEditDeliveryEmployee.xaml**

[Код](#PageEditDeliveryEmployee)

**PageEditGoods.xaml**

[Код](#PageEditGoods)

**PageEditOrders.xaml**

[Код](#PageEditOrders)

**PageEditOrdersGoods.xaml**

[Код](#PageEditOrdersGoods)

**PageEditStore.xaml**

[Код](#PageEditStore)

**PageListGoods.xaml**

[Код](#PageListGoods)

**PageListStore.xaml**

[Код](#PageListStore)

## **3. 2 Текст программы:**

**PageEdit.xaml.cs**

[Код](#PageEditcs)

**PageEditCategory.xaml.cs**

[Код](#PageEditCategorycs)

**PageEditClient.xaml.cs**

[Код](#PageEditClientcs)

**PageEditDelivery.xaml.cs**

[Код](#PageEditDeliverycs)

**PageEditDeliveryEmployee.xaml.cs**

[Код](#PageEditDeliveryEmployeecs)

**PageEditGoods.xaml.cs**

[Код](#PageEditGoodscs)

**PageEditOrders.xaml.cs**

[Код](#PageEditOrderscs)

**PageEditOrdersGoods.xaml.cs**

[Код](#PageEditOrdersGoodscs)

**PageEditStore.xaml.cs**

[Код](#PageEditStorecs)

**PageListGoods.xaml.cs**

[Код](#PageListGoodscs)

**PageListStore.xaml.cs**

[Код](#PageListStorecs)

1. **Руководство пользователя**

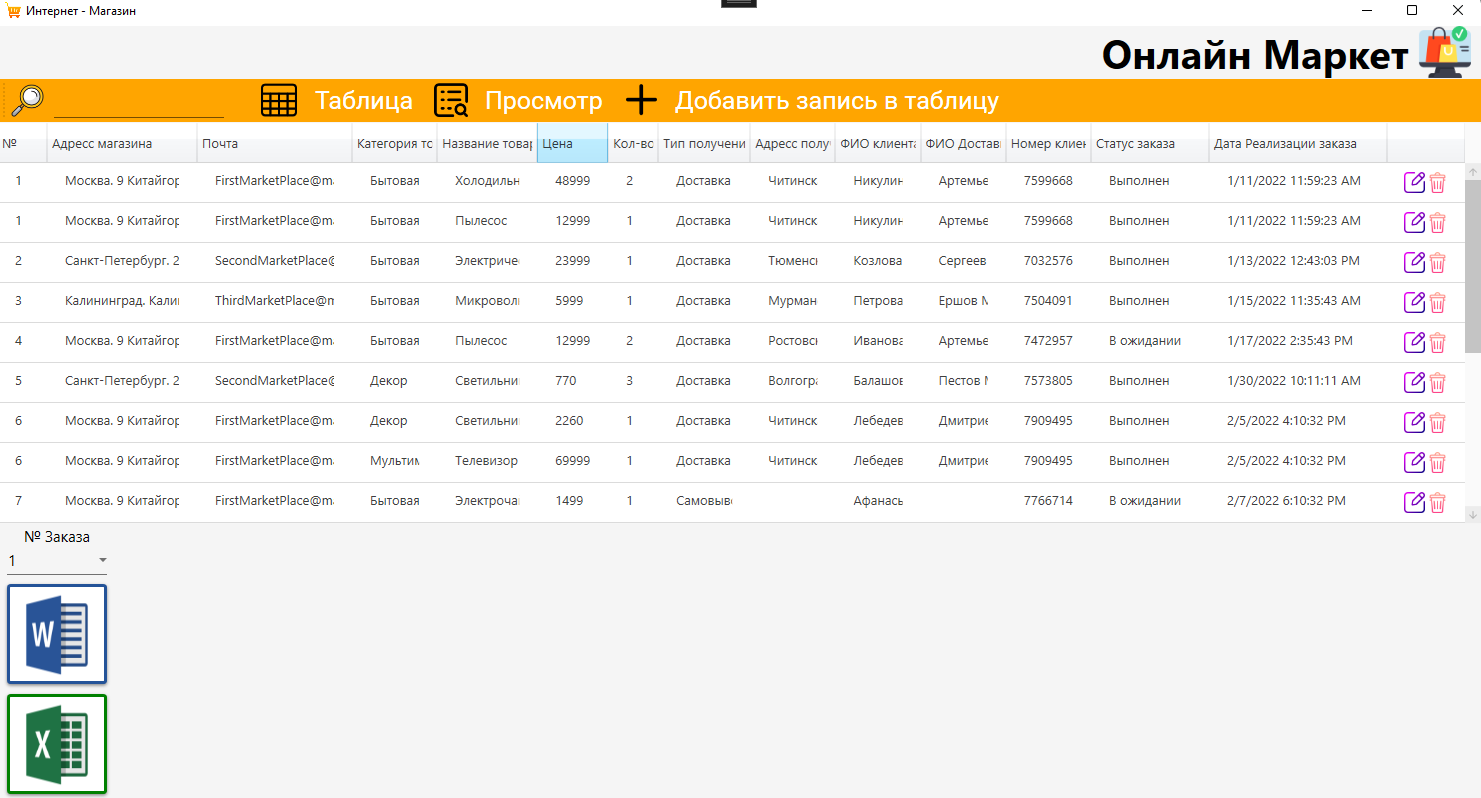
Для использования приложение необходимо:

1.Запустить exe файл приложения (EShop)



Рисунок№14 «exe файл программы»

2.После запуска открывается главное окно. Где можно выводить отчёт Excel и чек в Word, редактировать таблицу удалять записи, поиск по фамилии, убрать параметры поиск, а также переходить на другие страницы для просмотра и добавления записей.



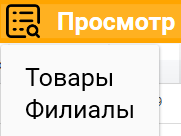
Рисунок№15 «Главная страница»

Для вывода в word надо выбрать номер заказа.



Рисунок№16 «Выбор заказа для вывода»

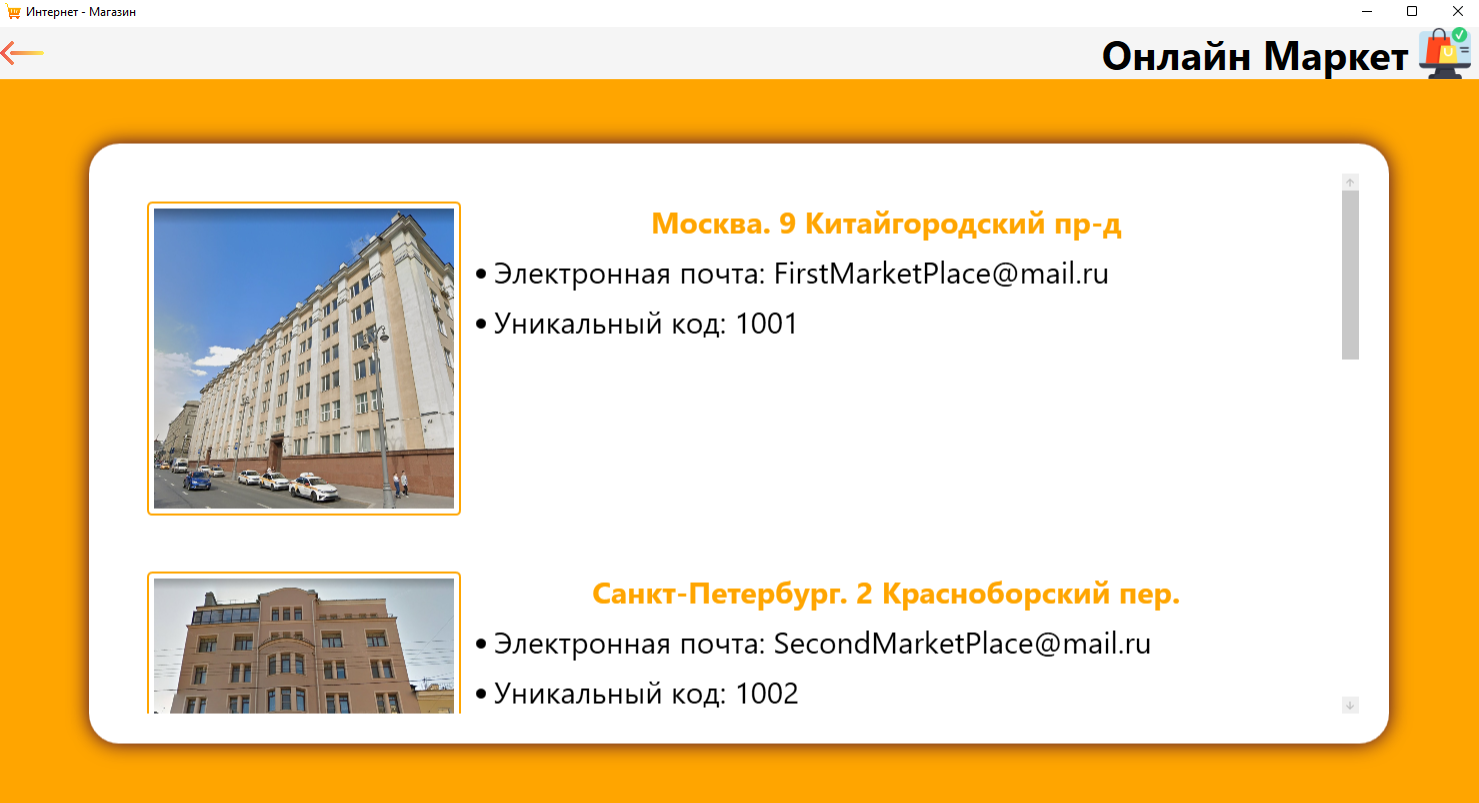
3.Для просмотра надо нажить просмотр и выбрать категорию.



Рисунок№17 «Выбор просмотра»

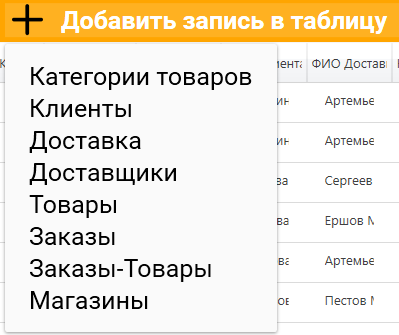


Рисунок№18 «Просмотр товары»



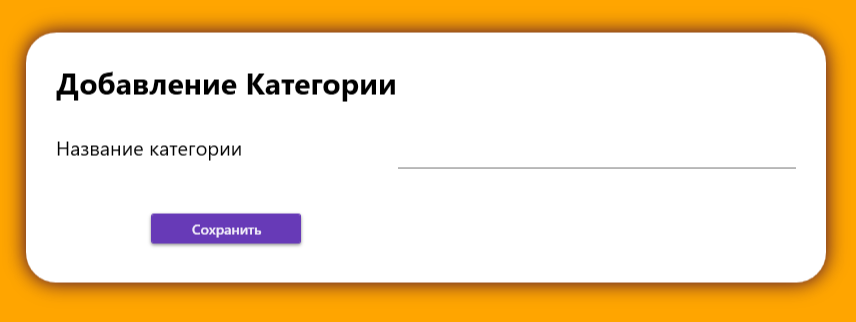
Рисунок№18 «Просмотр магазины»

Для добавления нужно выбрать таблицу из вкладки Добавить запись в таблицу.

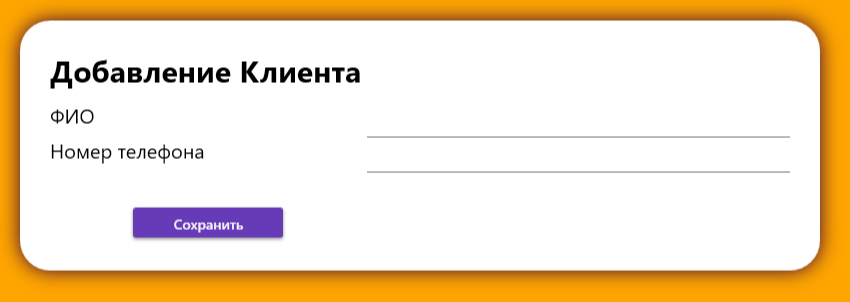


Рисунок№18 «Выбор таблиц для добавления»

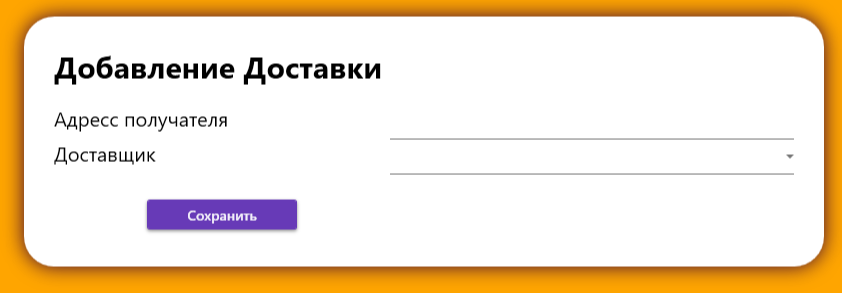
4.Для добавления заполняем строки и нажимаем сохранить.



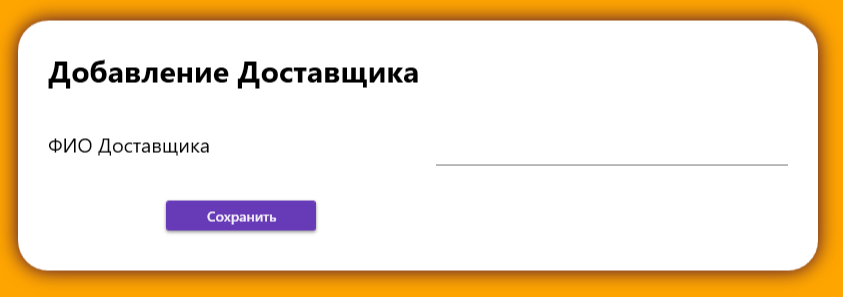
Рисунок№19 «Таблица для добавления категории»



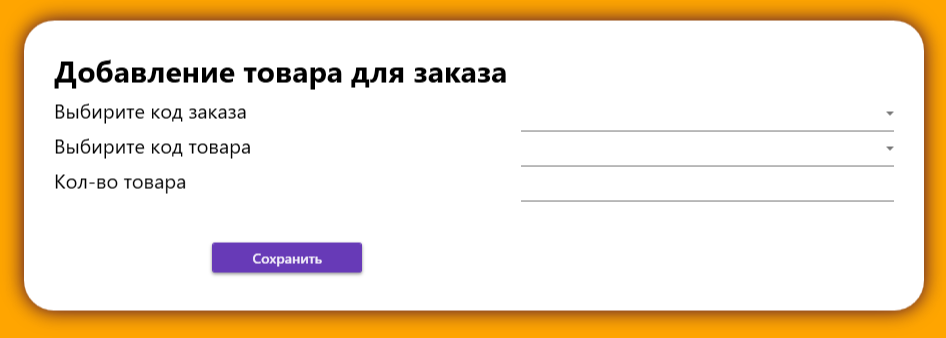
Рисунок№20 «Таблица для добавления клиента»



Рисунок№21 «Таблица для добавления доставки»

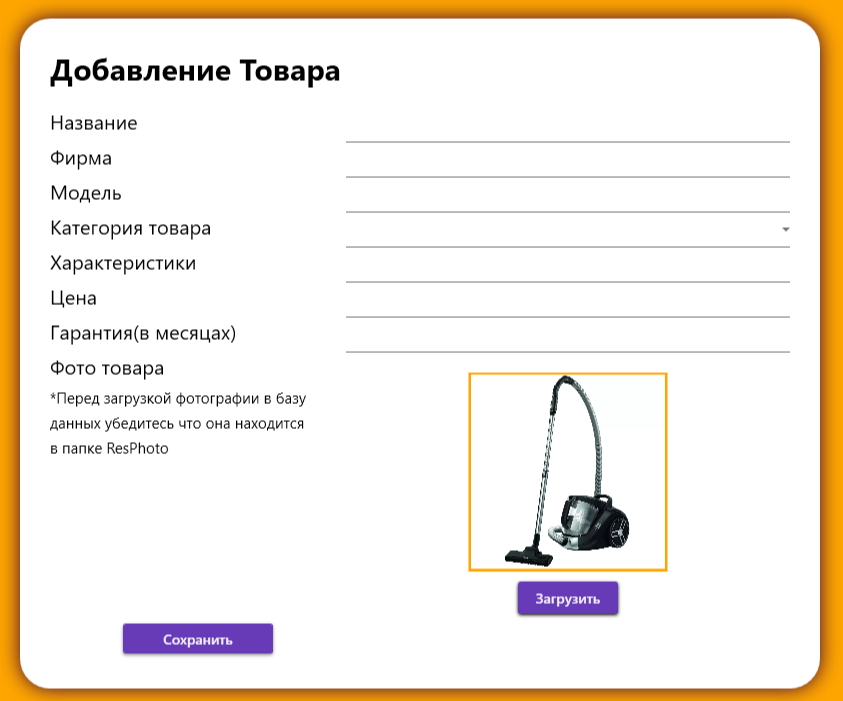


Рисунок№22 «Таблица для добавления доставщика»

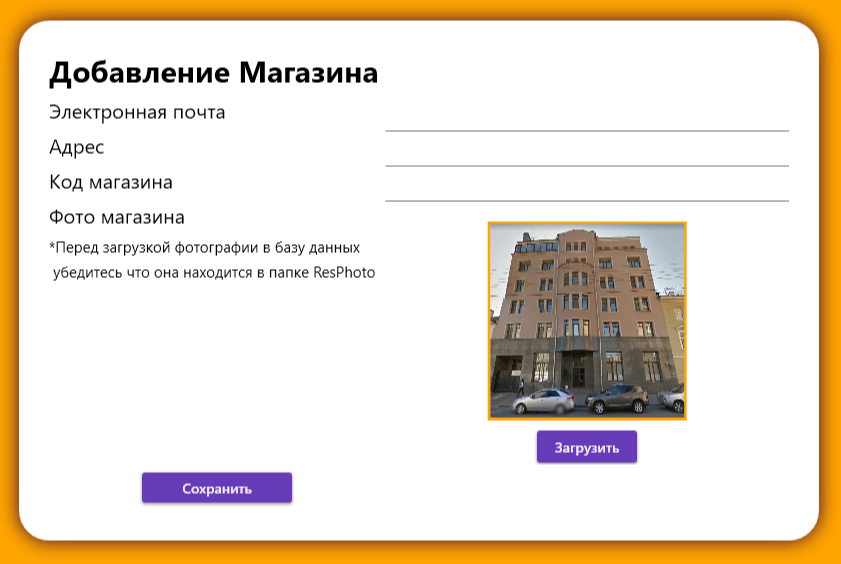


Рисунок№23 «Таблица для добавления товара для заказа»

Таблицы Товары и Магазины имеют добавления фото, для его загрузки в базу данных надо фотографию перекинуть в папку ResPhoto и загрузить через программу.

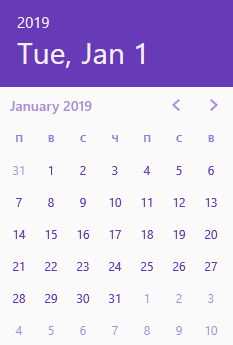


Рисунок№24 «Таблица для добавления товара»

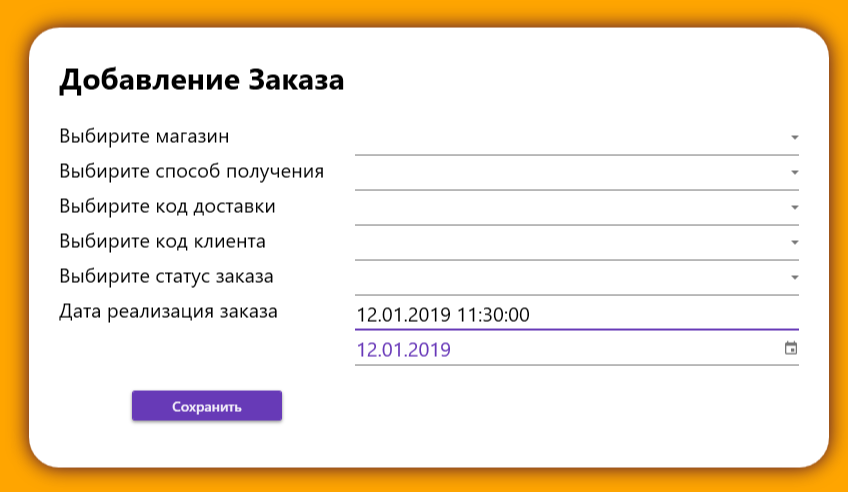


Рисунок№24 «Таблица для добавления магазина»

Для добавления Заказа нужно указать дату и время, для этого нужно выбрать дату и поменять время.

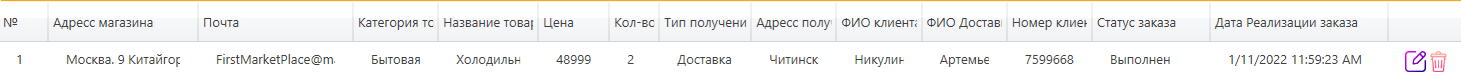


Рисунок№23 «Календарь для выбора даты»



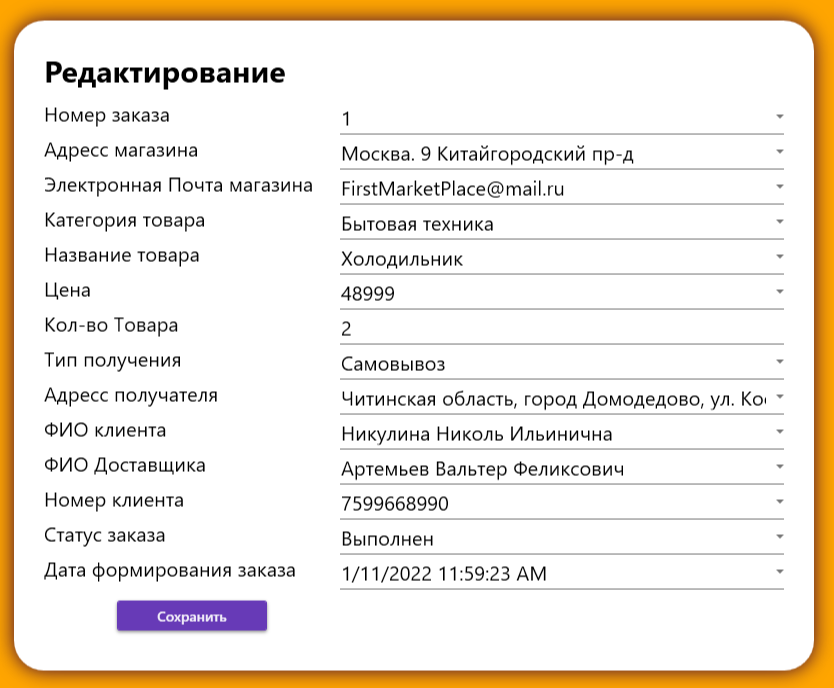
Рисунок№26 «Таблица для добавления заказа»

5.Для редактирования нажимаем на кнопку в записи из таблицы.



Рисунок№27 «Запись таблицы»

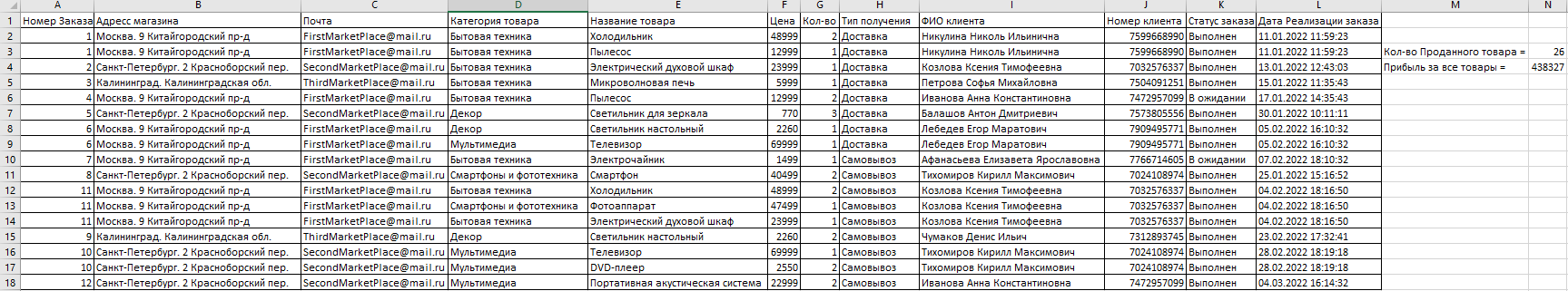
Выбираем данные и сохраняем.



Рисунок№28 «Страница редактирования»

**4. 1 Выполнение программного модуля:**

Вывод в Excel:



Рисунок№29 «Отчёт в Excel»

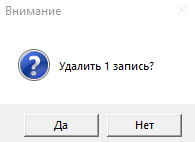
Вывод Чека в Word:



Рисунок№30 «Чек Word»

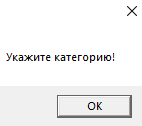
## **4. 2 Сообщения пользователю:**

1.При удалении записи высвечивается диалоговое окно (Удалить запись?).

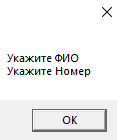


Рисунок№31 «Диалоговое окно удаления записи»

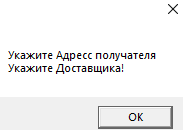
2.Сообщения, когда не вписаны данные в строку добавления.



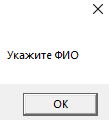
Рисунок№31 «Диалоговое окно добавления категории»



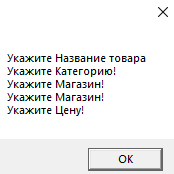
Рисунок№32 «Диалоговое окно добавления клиента»



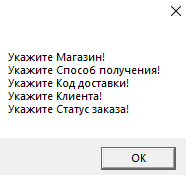
Рисунок№33 «Диалоговое окно добавления доставки»



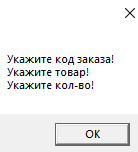
Рисунок№34 «Диалоговое окно добавления доставщика»



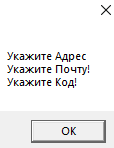
Рисунок№35 «Диалоговое окно добавления товара»



Рисунок№36 «Диалоговое окно добавления заказа»



Рисунок№37 «Диалоговое окно добавления товаров для заказа»



Рисунок№38 «Диалоговое окно добавления магазина»

1. **Тестирование программного модуля**

Тестирование программного обеспечения — процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом (ISO/IEC TR 19759:2005).

**5. 1 Модульное тестирование**

1.Для разработки модульных тестов можно проверить цену товара Price. В данном тесте Y=1 равен положительным эфектом а Y=0 равен негативному.

Тест1: Цена не может равняться 0 и не может быть отрицательной.

[TestMethod]

public void PriceO\_Or\_minus1()

{

int Price = -3;

int expected\_Y = 0;

int actual\_Y = c.Formula(Price);

Assert.AreEqual(expected\_Y, actual\_Y);

}

Тест2: Цена не может равняться миллиарду или быть больше.

[TestMethod]

public void Price1000000000()

{

int Price = 1000000000;

int expected\_Y = 0;

int actual\_Y = c.Formula(Price);

Assert.AreEqual(expected\_Y, actual\_Y);

}

Тест3: Тест если цена соответствующая.

[TestMethod]

public void PriceIsNormal()

{

int Price = 40200;

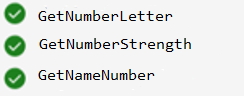
int expected\_Y = 1;

int actual\_Y = c.Formula(Price);

Assert.AreEqual(expected\_Y, actual\_Y);

}

Итог:



Рисунок№39 «Итоги тестов»

**5. 2 Тестовые сценарии**

Таблица№11 «Аннотация теста»

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| Название проекта | UnitTestForEshop |
| Рабочая версия | 1.0 |
| Имя тестирующего | Широнин Николай Анатольевич |
| Дата(ы) теста | 06.04.2023 |

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| Тестовый пример # | PriceO\_Or\_minus1 |
| Приоритет тестирования  (Низкий/Средний/Высокий) | Высокий |
| Заголовок/название теста | Проверка данных ввода |
| Краткое изложение теста | При сохранении информации цены она не может быть нулём или отрицательным. |
| Этапы теста | Зайти на страницу добавления, указать следующие значения: Название, Фирма, Модель, Категория товара, Характеристики, Цена, Гарантия, Фото товара. |
| Тестовые данные | ФМО: Холодильник, LG, K2000, Бытовая техника, технология Frost, -3, 12. |
| Ожидаемый результат | Y=0, тоесть ошибка |
| Фактический результат | Запись не сохранена |
| Предварительное условие | Должна быть кнопка для перехода на страницу добавления и сохранения данных, элементы управления куда будут заполняться данные и кнопка для сохранения этих данных. |
| Постусловие | Данные не были добавлены в таблицу. |
| Статус (Зачет/Незачет) | Не зачёт |

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| Тестовый пример # | Price1000000000 |
| Приоритет тестирования  (Низкий/Средний/Высокий) | Средний |
| Заголовок/название теста | Проверка данных ввода |
| Краткое изложение теста | При сохранении информации цена не может быть больше миллиарда. |
| Этапы теста | Зайти на страницу добавления, указать следующие значения: Название, Фирма, Модель, Категория товара, Характеристики, Цена, Гарантия, Фото товара. |
| Тестовые данные | ФМО: Холодильник, LG, K2000, Бытовая техника, технология Frost, 1000000000, 12. |
| Ожидаемый результат | Y=0, тоесть ошибка |
| Фактический результат | Запись не сохранена |
| Предварительное условие | Должна быть кнопка для перехода на страницу добавления и сохранения данных, элементы управления куда будут заполняться данные и кнопка для сохранения этих данных. |
| Постусловие | Данные не были добавлены в таблицу. |
| Статус (Зачет/Незачет) | Не зачёт |

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| Тестовый пример # | PriceIsNormal |
| Приоритет тестирования  (Низкий/Средний/Высокий) | Низкий |
| Заголовок/название теста | Проверка данных ввода |
| Краткое изложение теста | Цена яляется допустимой. |
| Этапы теста | Зайти на страницу добавления, указать следующие значения: Название, Фирма, Модель, Категория товара, Характеристики, Цена, Гарантия, Фото товара. |
| Тестовые данные | ФМО: Холодильник, LG, K2000, Бытовая техника, технология Frost, 40200, 12. |
| Ожидаемый результат | Y=1, тоесть пройдено |
| Фактический результат | Запись сохранена |
| Предварительное условие | Должна быть кнопка для перехода на страницу добавления и сохранения данных, элементы управления куда будут заполняться данные и кнопка для сохранения этих данных. |
| Постусловие | Данные были добавлены в таблицу. |
| Статус (Зачет/Незачет) | Зачёт |

**6.**  **Отладка программного модуля**

1.Записи не сохранялись в базу данных.

Для исправления надо было поставить автозаполнения id



Рисунок№40 «Автозаполнение»

2.Не удалось преобразовать int в int

Для решения надо было int поменять в var



Рисунок№41 «Исправленный выбор заказа»

**Заключение**

На основании проведенной работы по созданию программы для учета заказов можно сделать вывод, что такая программа является важным инструментом для управления бизнесом. Она позволяет автоматизировать процессы учета заказов, что упрощает работу менеджеров и сокращает время, затрачиваемое на выполнение этих задач.

В процессе разработки программы были использованы современные технологии программирования, такие как Entity Framework и Microsoft sql Server, что обеспечивает высокую производительность и удобство использования.

Кроме того, были реализованы функции для мониторинга заказов, их добавления и редактирования, вывода отчёта в Excel и чеков для покупателей через Word, что позволяет более эффективно управлять бизнесом и принимать правильные решения на основе полученных данных.

В целом, созданная программа для учета заказов является полезным инструментом для любого бизнеса и может значительно улучшить процессы управления заказами и учета финансовых операций.

**Список литературы**

1. Агуров, Павел C#. Сборник рецептов / Павел Агуров. - М.: "БХВ-Петербург", 2012. - 432 c.

2. Албахари, Джозеф C# 3.0. Справочник / Джозеф Албахари, Бен Албахари. - М.: БХВ-Петербург, 2012. - 944 c.

3. Вагнер, Билл С# Эффективное программирование / Билл Вагнер. - М.: ЛОРИ, 2013. - 320 c.

4. Зиборов, В.В. Visual C# 2012 на примерах / В.В. Зиборов. - М.: БХВ-Петербург, 2013. - 480 c.

5. Зиборов, Виктор Visual C# 2010 на примерах / Виктор Зиборов. - М.: "БХВ-Петербург", 2011. - 432 c.

6. Ишкова, Э. А. Самоучитель С#. Начала программирования / Э.А. Ишкова. - М.: Наука и техника, 2013. - 496 c.

**Ссылки на источники**

1. Официальная документация Microsoft по c#

<https://learn.microsoft.com/ru-ru/dotnet/csharp/>

1. Метанит

<https://metanit.com/sharp/>

1. Форум по c#

<https://www.cyberforum.ru/csharp-net/>

**Приложения**

Приложение №1. Текст программы

**App.Xaml**

<Application.Resources>

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

<ResourceDictionary Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.Light.xaml" />

<ResourceDictionary Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.Defaults.xaml" />

<ResourceDictionary Source="pack://application:,,,/MaterialDesignColors;component/Themes/Recommended/Primary/MaterialDesignColor.DeepPurple.xaml" />

<ResourceDictionary Source="pack://application:,,,/MaterialDesignColors;component/Themes/Recommended/Accent/MaterialDesignColor.Lime.xaml" />

<ResourceDictionary Source="Styles/ButtonBack.xaml"/>

<ResourceDictionary Source="Styles/DataGrid.xaml"/>

</ResourceDictionary.MergedDictionaries>

</ResourceDictionary>

</Application.Resources>

**ButtonBack.xaml**

<Style TargetType="{x:Type Button}" x:Key="BorderlessButton">

<Setter Property="Background" Value="Transparent" />

<Setter Property="BorderThickness" Value="0" />

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type Button}">

<Border Padding="{TemplateBinding Padding}" Background="{TemplateBinding Background}">

<ContentPresenter VerticalAlignment="Center" HorizontalAlignment="Center" />

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

**DataGrid.xaml**

<Style x:Key="DataGridRowStyle" TargetType="{x:Type DataGridRow}">

<Setter Property="Background" Value="{DynamicResource {x:Static SystemColors.WindowBrushKey}}"/>

<Setter Property="SnapsToDevicePixels" Value="true"/>

<Setter Property="Validation.ErrorTemplate" Value="{x:Null}"/>

<Setter Property="ValidationErrorTemplate">

<Setter.Value>

<ControlTemplate>

<TextBlock Foreground="Red" Margin="2,0,0,0" Text="!" VerticalAlignment="Center"/>

</ControlTemplate>

</Setter.Value>

</Setter>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type DataGridRow}">

<Border x:Name="DGR\_Border" Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}" SnapsToDevicePixels="True">

<SelectiveScrollingGrid>

<SelectiveScrollingGrid.ColumnDefinitions>

<ColumnDefinition Width="Auto"/>

<ColumnDefinition Width="\*"/>

</SelectiveScrollingGrid.ColumnDefinitions>

<SelectiveScrollingGrid.RowDefinitions>

<RowDefinition Height="40"/>

<RowDefinition Height="\*"/>

</SelectiveScrollingGrid.RowDefinitions>

<DataGridCellsPresenter Grid.Column="1" ItemsPanel="{TemplateBinding ItemsPanel}" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}"/>

<DataGridDetailsPresenter Grid.Column="1" Grid.Row="1" SelectiveScrollingGrid.SelectiveScrollingOrientation="{Binding AreRowDetailsFrozen, ConverterParameter={x:Static SelectiveScrollingOrientation.Vertical}, Converter={x:Static DataGrid.RowDetailsScrollingConverter}, RelativeSource={RelativeSource AncestorType={x:Type DataGrid}}}" Visibility="{TemplateBinding DetailsVisibility}"/>

<DataGridRowHeader Grid.RowSpan="2" SelectiveScrollingGrid.SelectiveScrollingOrientation="Vertical" Visibility="{Binding HeadersVisibility, ConverterParameter={x:Static DataGridHeadersVisibility.Row}, Converter={x:Static DataGrid.HeadersVisibilityConverter}, RelativeSource={RelativeSource AncestorType={x:Type DataGrid}}}"/>

</SelectiveScrollingGrid>

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

<Trigger Property="IsNewItem" Value="True">

<Setter Property="Margin" Value="{Binding NewItemMargin, RelativeSource={RelativeSource AncestorType={x:Type DataGrid}}}"/>

</Trigger>

</Style.Triggers>

</Style>

<Style x:Key="ColumnHeaderGripperStyle" TargetType="{x:Type Thumb}">

<Setter Property="Width" Value="8"/>

<Setter Property="Height" Value="40"/>

<Setter Property="Background" Value="Transparent"/>

<Setter Property="Cursor" Value="SizeWE"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type Thumb}">

<Border Background="{TemplateBinding Background}" Padding="{TemplateBinding Padding}"/>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="DataGridColumnHeaderStyle" TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="VerticalContentAlignment" Value="Center"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type DataGridColumnHeader}">

<Grid>

<theme:DataGridHeaderBorder Background="{TemplateBinding Background}"

BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}"

IsPressed="{TemplateBinding IsPressed}" IsHovered="{TemplateBinding IsMouseOver}" IsClickable="{TemplateBinding CanUserSort}" Padding="{TemplateBinding Padding}" SortDirection="{TemplateBinding SortDirection}" SeparatorVisibility="{TemplateBinding SeparatorVisibility}" SeparatorBrush="{TemplateBinding SeparatorBrush}">

<ContentPresenter HorizontalAlignment="{TemplateBinding HorizontalContentAlignment}" RecognizesAccessKey="True" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}" VerticalAlignment="{TemplateBinding VerticalContentAlignment}"/>

</theme:DataGridHeaderBorder>

<Thumb x:Name="PART\_LeftHeaderGripper" HorizontalAlignment="Left" Style="{StaticResource ColumnHeaderGripperStyle}"/>

<Thumb x:Name="PART\_RightHeaderGripper" HorizontalAlignment="Right" Style="{StaticResource ColumnHeaderGripperStyle}"/>

</Grid>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

**PageTable.xaml**

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="14\*"/>

<RowDefinition Height="11\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0">

<ToolBar Height="45" VerticalAlignment="Top" Background="Orange">

<Image Source="/ResPhoto/Lupa.png" Height="33"/>

<TextBox Name="TxtSearch" Foreground="White" FontSize="18" Height="34" Width="170" TextChanged="TxtSearch\_TextChanged" Margin="10,0,0,0"></TextBox>

<Menu Height="34" Margin="35,0,0,0" Background="Orange">

<MenuItem Header="Таблица" Foreground="White" FontSize="25" VerticalAlignment="Center">

<MenuItem.Icon>

<Image Source="/ResPhoto/Table.png" Margin="-20,-10,-13,-10"></Image>

</MenuItem.Icon>

<MenuItem Header="Очистить"

Click="BtnClearAll\_Click" Foreground="Black"></MenuItem>

</MenuItem>

<MenuItem Header="Просмотр" Foreground="White" FontSize="25" VerticalAlignment="Center">

<MenuItem.Icon>

<Image Source="/ResPhoto/Search.png" Margin="-20,-9,-16,-9"></Image>

</MenuItem.Icon>

<MenuItem Header="Товары"

Name="ToListGoods" Click="ToListGoods\_Click" Foreground="Black"></MenuItem>

<MenuItem Header="Филиалы"

Name="ToListStore" Click="ToListStore\_Click" Foreground="Black"></MenuItem>

</MenuItem>

<MenuItem Header="Добавить запись в таблицу" Foreground="White" FontSize="25" VerticalAlignment="Center">

<MenuItem.Icon>

<Image Source="/ResPhoto/Plus.png" Margin="-9,-10,-6,-8"></Image>

</MenuItem.Icon>

<MenuItem Header="Категории товаров"

Click="BtnToCategory\_Click" Foreground="Black"></MenuItem>

<MenuItem Header="Клиенты"

Click="BtnToClient\_Click" Foreground="Black"></MenuItem>

<MenuItem Header="Доставка"

Click="BtnToDelivery\_Click" Foreground="Black"></MenuItem>

<MenuItem Header="Доставщики"

Click="BtnToDeliveryEmployee\_Click" Foreground="Black"></MenuItem>

<MenuItem Header="Товары"

Click="BtnToGoods\_Click" Foreground="Black"></MenuItem>

<MenuItem Header="Заказы"

Click="BtnToOrders\_Click" Foreground="Black"></MenuItem>

<MenuItem Header="Заказы-Товары"

Click="BtnToOrders\_Goods\_Click" Foreground="Black"></MenuItem>

<MenuItem Header="Магазины"

Click="BtnToStoreClick" Foreground="Black"></MenuItem>

</MenuItem>

</Menu>

</ToolBar>

</StackPanel>

<DataGrid ColumnHeaderStyle="{DynamicResource DataGridColumnHeaderStyle}"

RowStyle="{DynamicResource DataGridRowStyle}"

x:Name="DGridOrders"

AutoGenerateColumns="False" IsReadOnly="True"

Margin="0,45,0,0" Height="400" VerticalAlignment="Top" Width="1485"

HorizontalAlignment="Left">

<DataGrid.Columns>

<DataGridTextColumn

Header="№" Width="50"

Binding="{Binding Orders.OrdersID}"/>

<DataGridTextColumn

Header="Адресс магазина" Width="150"

Binding="{Binding Orders.Store.Adress}"/>

<DataGridTextColumn

Header="Почта" Width="155"

Binding="{Binding Orders.Store.EMail}"/>

<DataGridTextColumn

Header="Категория товара" Width="\*"

Binding="{Binding Goods.Category.CName}"/>

<DataGridTextColumn

Header="Название товара" Width="100"

Binding="{Binding Goods.GName}"/>

<DataGridTextColumn

Header="Цена" Width="71"

Binding="{Binding Goods.Price}"/>

<DataGridTextColumn

Header="Кол-во" Width="50"

Binding="{Binding Quantity}"/>

<DataGridTextColumn

Header="Тип получения" Width="92"

Binding="{Binding Orders.WayToGet.Type}"/>

<DataGridTextColumn

Header="Адресс получателя" Width="\*"

Binding="{Binding Orders.Delivery.Adress}"/>

<DataGridTextColumn

Header="ФИО клиента" Width="\*"

Binding="{Binding Orders.Client.FIO}"/>

<DataGridTextColumn

Header="ФИО Доставщика" Width="\*"

Binding="{Binding Orders.Delivery.DeliveryEmployee.FIOEmployee}"/>

<DataGridTextColumn

Header="Номер клиента" Width="\*"

Binding="{Binding Orders.Client.PhoneNumber}"/>

<DataGridTextColumn

Header="Статус заказа" Width="118"

Binding="{Binding Orders.Status.SName}"/>

<DataGridTextColumn

Header="Дата Реализации заказа" Width="178"

Binding="{Binding Orders.DateTime}"/>

<DataGridTemplateColumn Width="Auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Center">

<Button Name="BtnEdit"

Click="BtnEdit\_Click"

Width="23"

Height="23"

Style="{StaticResource BorderlessButton}">

<Image Source="\ResPhoto\Redak.png" Stretch="Fill"></Image>

</Button>

<Button Name="BtnDelete"

Click="BtnDelete\_Click"

Width="23"

Height="23"

Style="{StaticResource BorderlessButton}">

<Image Source="\ResPhoto\Delete.png" Stretch="Fill"></Image>

</Button>

</StackPanel>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<StackPanel Orientation="Vertical" Grid.Row="1" HorizontalAlignment="Left" Margin="10,10,0,0">

<ComboBox x:Name="OrderSelect" FontSize="15">

<ComboBoxItem>1</ComboBoxItem>

<ComboBoxItem>2</ComboBoxItem>

<ComboBoxItem>3</ComboBoxItem>

<ComboBoxItem>4</ComboBoxItem>

<ComboBoxItem>5</ComboBoxItem>

<ComboBoxItem>6</ComboBoxItem>

<ComboBoxItem>7</ComboBoxItem>

<ComboBoxItem>8</ComboBoxItem>

<ComboBoxItem>9</ComboBoxItem>

<ComboBoxItem>10</ComboBoxItem>

<ComboBoxItem>11</ComboBoxItem>

<ComboBoxItem>12</ComboBoxItem>

<ComboBoxItem>13</ComboBoxItem>

<ComboBoxItem>14</ComboBoxItem>

<ComboBoxItem>15</ComboBoxItem>

<ComboBoxItem>16</ComboBoxItem>

<ComboBoxItem>17</ComboBoxItem>

<ComboBoxItem>18</ComboBoxItem>

<ComboBoxItem>19</ComboBoxItem>

<ComboBoxItem>20</ComboBoxItem>

<ComboBoxItem>21</ComboBoxItem>

<ComboBoxItem>22</ComboBoxItem>

<ComboBoxItem>23</ComboBoxItem>

<ComboBoxItem>24</ComboBoxItem>

<ComboBoxItem>25</ComboBoxItem>

<ComboBoxItem>26</ComboBoxItem>

<ComboBoxItem>27</ComboBoxItem>

<ComboBoxItem>28</ComboBoxItem>

<ComboBoxItem>29</ComboBoxItem>

<ComboBoxItem>30</ComboBoxItem>

<ComboBoxItem>31</ComboBoxItem>

<ComboBoxItem>32</ComboBoxItem>

<ComboBoxItem>33</ComboBoxItem>

<ComboBoxItem>34</ComboBoxItem>

<ComboBoxItem>35</ComboBoxItem>

<ComboBoxItem>36</ComboBoxItem>

<ComboBoxItem>37</ComboBoxItem>

<ComboBoxItem>38</ComboBoxItem>

<ComboBoxItem>39</ComboBoxItem>

<ComboBoxItem>40</ComboBoxItem>

<ComboBoxItem>41</ComboBoxItem>

<ComboBoxItem>42</ComboBoxItem>

<ComboBoxItem>43</ComboBoxItem>

<ComboBoxItem>44</ComboBoxItem>

<ComboBoxItem>45</ComboBoxItem>

</ComboBox>

<Button

Width="100"

Height="100"

Background="White"

Margin="0,10,0,0"

BorderBrush="#245398"

BorderThickness="3"

Name="BtnWord"

Click="BtnWord\_Click">

<Image Source="\ResPhoto\Word.png" Stretch="Fill" HorizontalAlignment="Center"></Image>

</Button>

<Button

Width="100"

Height="100"

Background="White"

Margin="0,10,0,0"

BorderBrush="Green"

BorderThickness="3"

Name="BtnExcel"

Click="BtnExcel\_Click">

<Image Source="\ResPhoto\Excel.png" Stretch="Fill" HorizontalAlignment="Center"></Image>

</Button>

</StackPanel>

</Grid>

**PageEdit.xaml**

<Grid>

<Border Height="650" Width="800" CornerRadius="30" Margin="15" Background="White" VerticalAlignment="Center" Padding="30" >

<Border.Effect>

<DropShadowEffect BlurRadius="30" Color="Purple" ShadowDepth="0">

</DropShadowEffect>

</Border.Effect>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="10\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="2\*"/>

<ColumnDefinition Width="3\*"/>

</Grid.ColumnDefinitions>

<TextBlock Text="Редактирование" FontSize="30" FontWeight="Bold"/>

<StackPanel Grid.Row="1">

<TextBlock Text="Номер заказа" Height="35" FontSize="20"/>

<TextBlock Text="Адресс магазина" Height="35" FontSize="20"/>

<TextBlock Text="Электронная Почта магазина" Height="35" FontSize="20"/>

<TextBlock Text="Категория товара" Height="35" FontSize="20"/>

<TextBlock Text="Название товара" Height="35" FontSize="20"/>

<TextBlock Text="Цена" Height="35" FontSize="20"/>

<TextBlock Text="Кол-во Товара" Height="35" FontSize="20"/>

<TextBlock Text="Тип получения" Height="35" FontSize="20"/>

<TextBlock Text="Адресс получателя" Height="35" FontSize="20"/>

<TextBlock Text="ФИО клиента" Height="35" FontSize="20"/>

<TextBlock Text="ФИО Доставщика" Height="35" FontSize="20"/>

<TextBlock Text="Номер клиента" Height="35" FontSize="20"/>

<TextBlock Text="Статус заказа" Height="35" FontSize="20"/>

<TextBlock Text="Дата формирования заказа" Height="35" FontSize="20"/>

</StackPanel>

<StackPanel Grid.Row="1" Grid.Column="1">

<ComboBox Name="CmbOrdersID"

SelectedValue="{Binding OrdersID}" FontSize="20" Height="35"/>

<ComboBox Name="CmbStoreAdress"

SelectedValue="{Binding OrdersID}" FontSize="20" Height="35"/>

<ComboBox Name="CmbEMail"

SelectedValue="{Binding OrdersID}" FontSize="20" Height="35"/>

<ComboBox Name="CmbCName"

SelectedValue="{Binding GoodsID}" FontSize="20" Height="35"/>

<ComboBox Name="CmbGName"

SelectedValue="{Binding GoodsID}" FontSize="20" Height="35"/>

<ComboBox Name="CmbPrice"

SelectedValue="{Binding GoodsID}" FontSize="20" Height="35"/>

<TextBox Name="TxtQuantity"

Text="{Binding Quantity}" FontSize="20" Height="35"/>

<ComboBox Name="CmbType"

SelectedValue="{Binding OrdersID}" FontSize="20" Height="35"/>

<ComboBox Name="CmbAdress"

SelectedValue="{Binding OrdersID}" FontSize="20" Height="35"/>

<ComboBox Name="CmbFIO"

SelectedValue="{Binding OrdersID}" FontSize="20" Height="35"/>

<ComboBox Name="CmbFIOEmployee"

SelectedValue="{Binding OrdersID}" FontSize="20" Height="35"/>

<ComboBox Name="CmbPhoneNumber"

SelectedValue="{Binding OrdersID}" FontSize="20" Height="35"/>

<ComboBox Name="CmbSName"

SelectedValue="{Binding OrdersID}" FontSize="20" Height="35"/>

<ComboBox Name="CmbDateTime"

SelectedValue="{Binding OrdersID}" FontSize="20" Height="35"/>

</StackPanel>

<StackPanel Grid.Row="2" Grid.Column="0" VerticalAlignment="Center">

<Button Content="Сохранить"

Width="150"

Height="30"

Name="BtnSave"

Click="BtnSave\_Click"/>

</StackPanel>

</Grid>

</Border>

</Grid>

**PageEditCategory.xaml**

<Grid>

<Border Height="250" Width="800" CornerRadius="30" Margin="15" Background="White" VerticalAlignment="Center" Padding="30" >

<Border.Effect>

<DropShadowEffect BlurRadius="30" Color="Purple" ShadowDepth="0">

</DropShadowEffect>

</Border.Effect>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="6\*"/>

<RowDefinition Height="6\*"/>

<RowDefinition Height="4\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="6\*"/>

<ColumnDefinition Width="7\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Row="1">

<TextBlock Text="Название категории" Height="35" FontSize="20"/>

</StackPanel>

<TextBlock Text="Добавление Категории" FontSize="30" FontWeight="Bold"/>

<StackPanel Grid.Row="1" Grid.Column="1">

<TextBox Name="TxtGName"

Text="{Binding CName}" FontSize="20" Height="35"/>

</StackPanel>

<StackPanel Grid.Row="2" Grid.Column="0" VerticalAlignment="Center">

<Button Content="Сохранить"

Width="150"

Height="30"

Name="BtnSave"

Click="BtnSave\_Click"/>

</StackPanel>

</Grid>

</Border>

</Grid>

**PageEditClient.xaml**

<Grid>

<Border Height="250" Width="800" CornerRadius="30" Margin="15" Background="White" VerticalAlignment="Center" Padding="30" >

<Border.Effect>

<DropShadowEffect BlurRadius="30" Color="Purple" ShadowDepth="0">

</DropShadowEffect>

</Border.Effect>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="3\*"/>

<RowDefinition Height="6\*"/>

<RowDefinition Height="2\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="3\*"/>

<ColumnDefinition Width="4\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Row="1">

<TextBlock Text="ФИО" Height="35" FontSize="20"/>

<TextBlock Text="Номер телефона" Height="35" FontSize="20"/>

</StackPanel>

<TextBlock Text="Добавление Клиента" FontSize="30" FontWeight="Bold"/>

<StackPanel Grid.Row="1" Grid.Column="1">

<TextBox Name="TxtFIO"

Text="{Binding FIO}" FontSize="20" Height="35"/>

<TextBox Name="TxtPhone"

Text="{Binding PhoneNumber}" FontSize="20" Height="35"/>

</StackPanel>

<StackPanel Grid.Row="2" Grid.Column="0" VerticalAlignment="Center">

<Button Content="Сохранить"

Width="150"

Height="30"

Name="BtnSave"

Click="BtnSave\_Click"/>

</StackPanel>

</Grid>

</Border>

</Grid>

**PageEditDelivery.xaml**

<Grid>

<Border Height="250" Width="800" CornerRadius="30" Margin="15" Background="White" VerticalAlignment="Center" Padding="30" >

<Border.Effect>

<DropShadowEffect BlurRadius="30" Color="Purple" ShadowDepth="0">

</DropShadowEffect>

</Border.Effect>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="4\*"/>

<RowDefinition Height="6\*"/>

<RowDefinition Height="3\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="5\*"/>

<ColumnDefinition Width="6\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Row="1">

<TextBlock Text="Адресс получателя" Height="35" FontSize="20"/>

<TextBlock Text="Доставщик" Height="35" FontSize="20"/>

</StackPanel>

<TextBlock Text="Добавление Доставки" FontSize="30" FontWeight="Bold"/>

<StackPanel Grid.Row="1" Grid.Column="1">

<TextBox Name="TxtAdress"

Text="{Binding Adress}" FontSize="20" Height="35"/>

<ComboBox Name="CmbEmployee"

SelectedValue="{Binding DeliveryEmployeeID}" FontSize="20" Height="35"/>

</StackPanel>

<StackPanel Grid.Row="2" Grid.Column="0" VerticalAlignment="Center">

<Button Content="Сохранить"

Width="150"

Height="30"

Name="BtnSave"

Click="BtnSave\_Click"/>

</StackPanel>

</Grid>

</Border>

</Grid>

**PageEditDeliveryEmployee.xaml**

<Grid>

<Border Height="250" Width="800" CornerRadius="30" Margin="15" Background="White" VerticalAlignment="Center" Padding="30" >

<Border.Effect>

<DropShadowEffect BlurRadius="30" Color="Purple" ShadowDepth="0">

</DropShadowEffect>

</Border.Effect>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="8\*"/>

<RowDefinition Height="6\*"/>

<RowDefinition Height="5\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="11\*"/>

<ColumnDefinition Width="10\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Row="1">

<TextBlock Text="ФИО Доставщика" Height="35" FontSize="20"/>

</StackPanel>

<TextBlock Text="Добавление Доставщика" FontSize="30" FontWeight="Bold"/>

<StackPanel Grid.Row="1" Grid.Column="1">

<TextBox Name="TxtEmployee"

Text="{Binding FIOEmployee}" FontSize="20" Height="35"/>

</StackPanel>

<StackPanel Grid.Row="2" Grid.Column="0" VerticalAlignment="Center">

<Button Content="Сохранить"

Width="150"

Height="30"

Name="BtnSave"

Click="BtnSave\_Click"/>

</StackPanel>

</Grid>

</Border>

</Grid>

**PageEditGoods.xaml**

<Grid>

<Border Height="670" Width="800" CornerRadius="30" Margin="15" Background="White" VerticalAlignment="Center" Padding="30" >

<Border.Effect>

<DropShadowEffect BlurRadius="30" Color="Purple" ShadowDepth="0">

</DropShadowEffect>

</Border.Effect>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="3\*"/>

<RowDefinition Height="26\*"/>

<RowDefinition Height="2\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="2\*"/>

<ColumnDefinition Width="3\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Row="1">

<TextBlock Text="Название" Height="35" FontSize="20"/>

<TextBlock Text="Фирма" Height="35" FontSize="20"/>

<TextBlock Text="Модель" Height="35" FontSize="20"/>

<TextBlock Text="Категория товара" Height="35" FontSize="20"/>

<TextBlock Text="Характеристики" Height="35" FontSize="20"/>

<TextBlock Text="Цена" Height="35" FontSize="20"/>

<TextBlock Text="Гарантия(в месяцах)" Height="35" FontSize="20"/>

<TextBlock Text="Фото товара" Height="35" FontSize="20"/>

<TextBlock Text="\*Перед загрузкой фотографии в базу" Height="35" FontSize="15"/>

<TextBlock Text="данных убедитесь что она находится" Height="35" FontSize="15" Margin="0,-10,0,0"/>

<TextBlock Text="в папке ResPhoto" Height="35" FontSize="15" Margin="0,-10,0,0"/>

</StackPanel>

<TextBlock Text="Добавление Товара" FontSize="30" FontWeight="Bold"/>

<StackPanel Grid.Row="1" Grid.Column="1">

<TextBox Name="TxtGName"

Text="{Binding GName}" FontSize="20" Height="35"/>

<TextBox Name="TxtFirm"

Text="{Binding Firm}" FontSize="20" Height="35"/>

<TextBox Name="TxtModel"

Text="{Binding Model}" FontSize="20" Height="35"/>

<ComboBox Name="CmbCategory"

SelectedValue="{Binding CategoryID}" FontSize="20" Height="35"/>

<TextBox Name="TxtSpecifications"

Text="{Binding Specifications}" FontSize="20" Height="35"/>

<TextBox Name="TxtPrice"

Text="{Binding Price}" FontSize="20" Height="35"/>

<TextBox Name="TxtWarranty"

Text="{Binding WarrantyPerMonth}" FontSize="20" Height="35"/>

<StackPanel Name="FillNull" HorizontalAlignment="Center" VerticalAlignment="Center" Margin="0,20,0,0">

<Rectangle Width="199" Height="199" Fill="White" Stroke="Orange" StrokeThickness="3"/>

<Path Data="M0 0L200 200M200 0L0 200" Height="199" Width="199" Stroke="Orange" Margin="0 -200 0 0" StrokeThickness="1"/>

</StackPanel>

<Image Name="imageArtist" Width="194" Height="194" Margin="0 -200 0 0" Source="{Binding GImage}" Stretch="UniformToFill"/>

<Button Content="Загрузить" Width="100" Name="ImageLoad" Click="ImageLoad\_Click" Margin="10"/>

</StackPanel>

<StackPanel Grid.Row="2" Grid.Column="0" VerticalAlignment="Center">

<Button Content="Сохранить"

Width="150"

Height="30"

Name="BtnSave"

Click="BtnSave\_Click"/>

</StackPanel>

</Grid>

</Border>

</Grid>

**PageEditOrders.xaml**

<Grid>

<Border Height="440" Width="800" CornerRadius="30" Margin="15" Background="White" VerticalAlignment="Center" Padding="30" >

<Border.Effect>

<DropShadowEffect BlurRadius="30" Color="Purple" ShadowDepth="0">

</DropShadowEffect>

</Border.Effect>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="4\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="2\*"/>

<ColumnDefinition Width="3\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Row="1">

<TextBlock Text="Выбирите магазин" Height="35" FontSize="20"/>

<TextBlock Text="Выбирите способ получения" Height="35" FontSize="20"/>

<TextBlock Text="Выбирите код доставки" Height="35" FontSize="20"/>

<TextBlock Text="Выбирите код клиента" Height="35" FontSize="20"/>

<TextBlock Text="Выбирите статус заказа" Height="35" FontSize="20"/>

<TextBlock Text="Дата реализация заказа" Height="35" FontSize="20"/>

</StackPanel>

<TextBlock Text="Добавление Заказа" FontSize="30" FontWeight="Bold"/>

<StackPanel Grid.Row="1" Grid.Column="1">

<ComboBox Name="CmbTxtStoreID"

SelectedValue="{Binding StoreID}" FontSize="20" Height="35"/>

<ComboBox Name="CmbTxtWayToGetID"

SelectedValue="{Binding WayToGetID}" FontSize="20" Height="35"/>

<ComboBox Name="CmbDeliveryID"

SelectedValue="{Binding DeliveryID}" FontSize="20" Height="35"/>

<ComboBox Name="CmbClientID"

SelectedValue="{Binding ClientID}" FontSize="20" Height="35"/>

<ComboBox Name="CmbStatusID"

SelectedValue="{Binding StatusID}" FontSize="20" Height="35"/>

<TextBox x:Name="TxtDateTime"

Text="{Binding DateTime}" FontSize="20" Height="35"/>

<DatePicker x:Name="DP1" FontSize="20" Height="35"

Foreground="#6737b8"

AllowDrop="True"

SelectedDateFormat="Short"

DisplayDate="2019-01-01"

SelectedDateChanged="DP1\_SelectedDateChanged">

</DatePicker>

<ComboBox x:Name="cboTP" FontSize="20" Visibility="Collapsed">

<ComboBoxItem> 10:00 </ComboBoxItem>

</ComboBox>

</StackPanel>

<StackPanel Grid.Row="2" Grid.Column="0" VerticalAlignment="Center">

<Button Content="Сохранить"

Width="150"

Height="30"

Name="BtnSave"

Click="BtnSave\_Click"/>

</StackPanel>

</Grid>

</Border>

</Grid>

**PageEditOrdersGoods.xaml**

<Grid>

<Border Height="290" Width="900" CornerRadius="30" Margin="15" Background="White" VerticalAlignment="Center" Padding="30" >

<Border.Effect>

<DropShadowEffect BlurRadius="30" Color="Purple" ShadowDepth="0">

</DropShadowEffect>

</Border.Effect>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="2\*"/>

<RowDefinition Height="6\*"/>

<RowDefinition Height="2\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="5\*"/>

<ColumnDefinition Width="4\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Row="1">

<TextBlock Text="Выбирите код заказа" Height="35" FontSize="20"/>

<TextBlock Text="Выбирите код товара" Height="35" FontSize="20"/>

<TextBlock Text="Кол-во товара" Height="35" FontSize="20"/>

</StackPanel>

<TextBlock Text="Добавление товара для заказа" FontSize="30" FontWeight="Bold"/>

<StackPanel Grid.Row="1" Grid.Column="1">

<ComboBox Name="CmbOrdersID"

SelectedValue="{Binding OrdersID}" FontSize="20" Height="35"/>

<ComboBox Name="CmbGoodsID"

SelectedValue="{Binding GoodsID}" FontSize="20" Height="35"/>

<TextBox Name="TxtQuantity"

Text="{Binding Quantity}" FontSize="20" Height="35"/>

</StackPanel>

<StackPanel Grid.Row="2" Grid.Column="0" VerticalAlignment="Center">

<Button Content="Сохранить"

Width="150"

Height="30"

Name="BtnSave"

Click="BtnSave\_Click"/>

</StackPanel>

</Grid>

</Border>

</Grid>

**PageEditStore.xaml**

<Grid>

<Border Height="520" Width="800" CornerRadius="30" Margin="15" Background="White" VerticalAlignment="Center" Padding="30" >

<Border.Effect>

<DropShadowEffect BlurRadius="30" Color="Purple" ShadowDepth="0">

</DropShadowEffect>

</Border.Effect>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="8\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="5\*"/>

<ColumnDefinition Width="6\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Row="1">

<TextBlock Text="Электронная почта" Height="35" FontSize="20"/>

<TextBlock Text="Адрес" Height="35" FontSize="20"/>

<TextBlock Text="Код магазина" Height="35" FontSize="20"/>

<TextBlock Text="Фото магазина" Height="35" FontSize="20"/>

<TextBlock Text="\*Перед загрузкой фотографии в базу данных" Height="35" FontSize="15"/>

<TextBlock Text=" убедитесь что она находится в папке ResPhoto" Height="35" FontSize="15" Margin="0,-10,0,0"/>

</StackPanel>

<TextBlock Text="Добавление Магазина" FontSize="30" FontWeight="Bold"/>

<StackPanel Grid.Row="1" Grid.Column="1">

<TextBox Name="TxtEMail"

Text="{Binding EMail}" FontSize="20" Height="35"/>

<TextBox Name="TxtAdress"

Text="{Binding Adress}" FontSize="20" Height="35"/>

<TextBox Name="TxtCode"

Text="{Binding Code}" FontSize="20" Height="35"/>

<StackPanel Name="FillNull" HorizontalAlignment="Center" VerticalAlignment="Center" Margin="0,20,0,0">

<Rectangle Width="199" Height="199" Fill="White" Stroke="Orange" StrokeThickness="3"/>

<Path Data="M0 0L200 200M200 0L0 200" Height="199" Width="199" Stroke="Orange" Margin="0 -200 0 0" StrokeThickness="1"/>

</StackPanel>

<Image Name="imageArtist" Width="194" Height="194" Margin="0 -200 0 0" Source="{Binding Picture}" Stretch="UniformToFill"/>

<Button Content="Загрузить" Width="100" Name="ImageLoad" Click="ImageLoad\_Click" Margin="10"/>

</StackPanel>

<StackPanel Grid.Row="2" Grid.Column="0" VerticalAlignment="Center">

<Button Content="Сохранить"

Width="150"

Height="30"

Name="BtnSave"

Click="BtnSave\_Click"/>

</StackPanel>

</Grid>

</Border>

</Grid>

**PageListGoods.xaml**

<Grid>

<Border Height="600" Width="1300" CornerRadius="30" Margin="15" Background="White" VerticalAlignment="Center" Padding="30" >

<Border.Effect>

<DropShadowEffect BlurRadius="30" Color="Purple" ShadowDepth="0">

</DropShadowEffect>

</Border.Effect>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="auto"></RowDefinition>

<RowDefinition Height="\*"></RowDefinition>

</Grid.RowDefinitions>

<ListView Grid.Row="1" Name="LviewGoods">

<ListView.ItemTemplate>

<DataTemplate>

<Grid Margin="20" Width="1150">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="2\*"/>

<ColumnDefinition Width="5\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Column="0" HorizontalAlignment="Left">

<Border BorderBrush="Orange" BorderThickness="2" CornerRadius="4">

<Image Width="300" Height="300" Grid.Row="1" Stretch="UniformToFill" HorizontalAlignment="Center" Margin="5">

<Image.Source>

<Binding Path="GImage">

<Binding.TargetNullValue>

<ImageSource>/ResPhoto/PictureNotFound.png</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

</Border>

</StackPanel>

<StackPanel Orientation="Horizontal" Grid.Column="1" VerticalAlignment="Top" HorizontalAlignment="Center">

<TextBlock Text="{Binding GName}" FontSize="30" FontWeight="Bold" Foreground="Orange"></TextBlock>

<TextBlock Text=": " FontSize="30" FontWeight="Bold" Foreground="Black"></TextBlock>

<TextBlock Text="{Binding Firm}" FontSize="30" FontWeight="Bold" Foreground="Black"></TextBlock>

<TextBlock Text=" " FontSize="30" FontWeight="Bold"></TextBlock>

<TextBlock Text="{Binding Model}" FontSize="30" FontWeight="Bold" Foreground="Black"></TextBlock>

</StackPanel>

<StackPanel Orientation="Horizontal" Grid.Column="1" HorizontalAlignment="Left" Margin="0,50,0,0">

<Ellipse Fill="Black" Width="10" Height="10" Margin="0,-220,0,0"/>

<TextBlock Text=" Цена: " FontSize="30" Foreground="Black"></TextBlock>

<TextBlock Text="{Binding Price}" FontSize="30" Foreground="Black"></TextBlock>

<TextBlock Text=" руб." FontSize="30" Foreground="Black"></TextBlock>

</StackPanel>

<StackPanel Orientation="Horizontal" Grid.Column="1" HorizontalAlignment="Left" Margin="0,100,0,0">

<Ellipse Fill="Black" Width="10" Height="10" Margin="0,-170,0,0"/>

<TextBlock Text=" Гарантия: " FontSize="30" Foreground="Black"></TextBlock>

<TextBlock Text="{Binding WarrantyPerMonth}" FontSize="30" Foreground="Black"></TextBlock>

<TextBlock Text=" мес." FontSize="30" Foreground="Black"></TextBlock>

</StackPanel>

<StackPanel Orientation="Horizontal" Grid.Column="1" HorizontalAlignment="Left" Margin="0,150,0,0">

<Ellipse Fill="Black" Width="10" Height="10" Margin="0,-120,0,0"/>

<TextBlock Text=" Характеристики:" FontSize="30" Foreground="Black"></TextBlock>

</StackPanel>

<StackPanel Grid.Column="1" HorizontalAlignment="Left" Margin="0,200,0,0">

<TextBlock Text="{Binding Specifications}" FontSize="16.1" Foreground="Black"></TextBlock>

</StackPanel>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

</Border>

</Grid>

**PageListStore.xaml**

<Grid>

<Border Height="600" Width="1300" CornerRadius="30" Margin="15" Background="White" VerticalAlignment="Center" Padding="30" >

<Border.Effect>

<DropShadowEffect BlurRadius="30" Color="Purple" ShadowDepth="0">

</DropShadowEffect>

</Border.Effect>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="auto"></RowDefinition>

<RowDefinition Height="\*"></RowDefinition>

</Grid.RowDefinitions>

<ListView Grid.Row="1" Name="LviewStore">

<ListView.ItemTemplate>

<DataTemplate>

<Grid Margin="20" Width="1150">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="2\*"/>

<ColumnDefinition Width="5\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Column="0" HorizontalAlignment="Left">

<Border BorderBrush="Orange" BorderThickness="2" CornerRadius="4">

<Image Width="300" Height="300" Grid.Row="1" Stretch="UniformToFill" HorizontalAlignment="Center" Margin="5">

<Image.Source>

<Binding Path="Picture">

<Binding.TargetNullValue>

<ImageSource>/ResPhoto/PictureNotFound.png</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

</Border>

</StackPanel>

<StackPanel Orientation="Horizontal" Grid.Column="1" VerticalAlignment="Top" HorizontalAlignment="Center">

<TextBlock Text="{Binding Adress}" FontSize="30" FontWeight="Bold" Foreground="Orange"></TextBlock>

</StackPanel>

<StackPanel Orientation="Horizontal" Grid.Column="1" HorizontalAlignment="Left" Margin="0,50,0,0">

<Ellipse Fill="Black" Width="10" Height="10" Margin="0,-220,0,0"/>

<TextBlock Text=" Электронная почта: " FontSize="30" Foreground="Black"></TextBlock>

<TextBlock Text="{Binding EMail}" FontSize="30" Foreground="Black"></TextBlock>

</StackPanel>

<StackPanel Orientation="Horizontal" Grid.Column="1" HorizontalAlignment="Left" Margin="0,100,0,0">

<Ellipse Fill="Black" Width="10" Height="10" Margin="0,-170,0,0"/>

<TextBlock Text=" Уникальный код: " FontSize="30" Foreground="Black"></TextBlock>

<TextBlock Text="{Binding Code}" FontSize="30" Foreground="Black"></TextBlock>

</StackPanel>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

</Border>

</Grid>

**PageEdit.xaml.cs**

public partial class PageEdit : Page

{

private OrdersGoods \_currentOrder = new OrdersGoods();

public PageEdit(OrdersGoods selectedOrder)

{

InitializeComponent();

CmbOrdersID.ItemsSource = EStoreEntities.GetContext().Orders.ToList();

CmbOrdersID.SelectedValuePath = "OrdersID";

CmbOrdersID.DisplayMemberPath = "OrdersID";

CmbDateTime.ItemsSource = EStoreEntities.GetContext().Orders.ToList();

CmbDateTime.SelectedValuePath = "OrdersID";

CmbDateTime.DisplayMemberPath = "DateTime";

CmbStoreAdress.ItemsSource = EStoreEntities.GetContext().Store.ToList();

CmbStoreAdress.SelectedValuePath = "StoreID";

CmbStoreAdress.DisplayMemberPath = "Adress";

CmbEMail.ItemsSource = EStoreEntities.GetContext().Store.ToList();

CmbEMail.SelectedValuePath = "StoreID";

CmbEMail.DisplayMemberPath = "EMail";

CmbCName.ItemsSource = EStoreEntities.GetContext().Category.ToList();

CmbCName.SelectedValuePath = "CategoryID";

CmbCName.DisplayMemberPath = "CName";

CmbGName.ItemsSource = EStoreEntities.GetContext().Goods.ToList();

CmbGName.SelectedValuePath = "GoodsID";

CmbGName.DisplayMemberPath = "GName";

CmbPrice.ItemsSource = EStoreEntities.GetContext().Goods.ToList();

CmbPrice.SelectedValuePath = "GoodsID";

CmbPrice.DisplayMemberPath = "Price";

CmbType.ItemsSource = EStoreEntities.GetContext().WayToGet.ToList();

CmbType.SelectedValuePath = "WayToGetID";

CmbType.DisplayMemberPath = "Type";

CmbAdress.ItemsSource = EStoreEntities.GetContext().Delivery.ToList();

CmbAdress.SelectedValuePath = "DeliveryID";

CmbAdress.DisplayMemberPath = "Adress";

CmbFIO.ItemsSource = EStoreEntities.GetContext().Client.ToList();

CmbFIO.SelectedValuePath = "ClientID";

CmbFIO.DisplayMemberPath = "FIO";

CmbFIOEmployee.ItemsSource = EStoreEntities.GetContext().DeliveryEmployee.ToList();

CmbFIOEmployee.SelectedValuePath = "DeliveryEmployeeID";

CmbFIOEmployee.DisplayMemberPath = "FIOEmployee";

CmbPhoneNumber.ItemsSource = EStoreEntities.GetContext().Client.ToList();

CmbPhoneNumber.SelectedValuePath = "ClientID";

CmbPhoneNumber.DisplayMemberPath = "PhoneNumber";

CmbSName.ItemsSource = EStoreEntities.GetContext().Status.ToList();

CmbSName.SelectedValuePath = "StatusID";

CmbSName.DisplayMemberPath = "SName";

if (selectedOrder != null)

\_currentOrder = selectedOrder;

DataContext = \_currentOrder;

}

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

{

StringBuilder error = new StringBuilder();

if (\_currentOrder.Quantity < 1)

error.AppendLine("Кол-во не может быть меньше 1!");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

//если пользователь новый

if (\_currentOrder.id == 0)

EStoreEntities.GetContext().OrdersGoods.Add(\_currentOrder);

try

{

EStoreEntities.GetContext().SaveChanges();

MessageBox.Show("Данные сохранены");

ClassFrame.frmObj.Navigate(new PageTable());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

**PageEditCategory.xaml.cs**

public partial class PageEditCategory : Page

{

private Category \_currentCategory = new Category();

public PageEditCategory(Category selectedCategory)

{

InitializeComponent();

if (selectedCategory != null)

\_currentCategory = selectedCategory;

DataContext = \_currentCategory;

}

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

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentCategory.CName))

error.AppendLine("Укажите категорию!");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

//если пользователь новый

if (\_currentCategory.CategoryID == 0)

EStoreEntities.GetContext().Category.Add(\_currentCategory);

try

{

EStoreEntities.GetContext().SaveChanges();

MessageBox.Show("Данные сохранены");

ClassFrame.frmObj.Navigate(new PageTable());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

**PageEditClient.xaml.cs**

public partial class PageEditClient : Page

{

private Client \_currentClient = new Client();

public PageEditClient(Client selectedClient)

{

InitializeComponent();

if (selectedClient != null)

\_currentClient = selectedClient;

DataContext = \_currentClient;

}

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

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentClient.FIO))

error.AppendLine("Укажите ФИО");

if (string.IsNullOrWhiteSpace(\_currentClient.PhoneNumber))

error.AppendLine("Укажите Номер");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

//если пользователь новый

if (\_currentClient.ClientID == 0)

EStoreEntities.GetContext().Client.Add(\_currentClient);

try

{

EStoreEntities.GetContext().SaveChanges();

MessageBox.Show("Данные сохранены");

ClassFrame.frmObj.Navigate(new PageTable());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

**PageEditDelivery.xaml.cs**

public partial class PageEditDelivery : Page

{

private Delivery \_currentDelivery = new Delivery();

public PageEditDelivery(Delivery selectedDelivery)

{

InitializeComponent();

CmbEmployee.ItemsSource = EStoreEntities.GetContext().DeliveryEmployee.ToList();

CmbEmployee.SelectedValuePath = "DeliveryEmployeeID";

CmbEmployee.DisplayMemberPath = "FIOEmployee";

if (selectedDelivery != null)

\_currentDelivery = selectedDelivery;

DataContext = \_currentDelivery;

}

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

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentDelivery.Adress))

error.AppendLine("Укажите Адресс получателя");

if (\_currentDelivery.DeliveryEmployeeID == null)

error.AppendLine("Укажите Доставщика!");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

//если пользователь новый

if (\_currentDelivery.DeliveryID == 0)

EStoreEntities.GetContext().Delivery.Add(\_currentDelivery);

try

{

EStoreEntities.GetContext().SaveChanges();

MessageBox.Show("Данные сохранены");

ClassFrame.frmObj.Navigate(new PageTable());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

**PageEditDeliveryEmployee.xaml.cs**

public partial class PageEditDeliveryEmployee : Page

{

private DeliveryEmployee \_currentEmployee = new DeliveryEmployee();

public PageEditDeliveryEmployee(DeliveryEmployee selectedDeliveryEmployee)

{

InitializeComponent();

if (selectedDeliveryEmployee != null)

\_currentEmployee = selectedDeliveryEmployee;

DataContext = \_currentEmployee;

}

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

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentEmployee.FIOEmployee))

error.AppendLine("Укажите ФИО");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

//если пользователь новый

if (\_currentEmployee.DeliveryEmployeeID == 0)

EStoreEntities.GetContext().DeliveryEmployee.Add(\_currentEmployee);

try

{

EStoreEntities.GetContext().SaveChanges();

MessageBox.Show("Данные сохранены");

ClassFrame.frmObj.Navigate(new PageTable());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

**PageEditGoods.xaml.cs**

public partial class PageEditGoods : Page

{

private Goods \_currentGoods = new Goods();

string imgLoc = "пусто";

public PageEditGoods(Goods selectedGoods)

{

InitializeComponent();

CmbCategory.ItemsSource = EStoreEntities.GetContext().Category.ToList();

CmbCategory.SelectedValuePath = "CategoryID";

CmbCategory.DisplayMemberPath = "CName";

if (selectedGoods != null)

\_currentGoods = selectedGoods;

DataContext = \_currentGoods;

}

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

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentGoods.GName))

error.AppendLine("Укажите Название товара");

if (\_currentGoods.CategoryID == null)

error.AppendLine("Укажите Категорию!");

if (\_currentGoods.Firm == null)

error.AppendLine("Укажите Магазин!");

if (\_currentGoods.Model == null)

error.AppendLine("Укажите Магазин!");

if (\_currentGoods.Price == null)

error.AppendLine("Укажите Цену!");

if (\_currentGoods.Price < 50)

error.AppendLine("Цена не может быть меньше 50р!");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

//если пользователь новый

if (\_currentGoods.GoodsID == 0)

EStoreEntities.GetContext().Goods.Add(\_currentGoods);

try

{

if (imgLoc != null && imgLoc != "пусто")

{

byte[] img = null;

FileStream fs = new FileStream(imgLoc, FileMode.Open, FileAccess.Read);

BinaryReader br = new BinaryReader(fs);

img = br.ReadBytes((int)fs.Length);

string filename = imgLoc.Substring(imgLoc.LastIndexOf('\\') + 1);

\_currentGoods.GImage = String.Concat("/ResPhoto/", filename);

}

EStoreEntities.GetContext().SaveChanges();

MessageBox.Show("Данные сохранены");

ClassFrame.frmObj.Navigate(new PageTable());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

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

{

try

{

OpenFileDialog dlg = new OpenFileDialog

{

Filter = "JPG Files (\*.jpg)|\*.jpg|PNG Files (\*.png)|\*.png|GIF Files (\*.gif)|\*.gif|All Files (\*.\*)|\*.\*",

Title = "Выберите фото/изображение лекарства"

};

if (dlg.ShowDialog() == true)

{

imgLoc = dlg.FileName.ToString();

imageArtist.Source = new BitmapImage(new Uri(imgLoc));

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

**PageEditOrders.xaml.cs**

public partial class PageEditOrders : Page

{

private Orders \_currentOrders = new Orders();

public PageEditOrders(Orders selectedOrders)

{

InitializeComponent();

CmbTxtStoreID.ItemsSource = EStoreEntities.GetContext().Store.ToList();

CmbTxtStoreID.SelectedValuePath = "StoreID";

CmbTxtStoreID.DisplayMemberPath = "Adress";

CmbTxtWayToGetID.ItemsSource = EStoreEntities.GetContext().WayToGet.ToList();

CmbTxtWayToGetID.SelectedValuePath = "WayToGetID";

CmbTxtWayToGetID.DisplayMemberPath = "Type";

CmbDeliveryID.ItemsSource = EStoreEntities.GetContext().Delivery.ToList();

CmbDeliveryID.SelectedValuePath = "DeliveryID";

CmbDeliveryID.DisplayMemberPath = "DeliveryID";

CmbClientID.ItemsSource = EStoreEntities.GetContext().Client.ToList();

CmbClientID.SelectedValuePath = "ClientID";

CmbClientID.DisplayMemberPath = "FIO";

CmbStatusID.ItemsSource = EStoreEntities.GetContext().Status.ToList();

CmbStatusID.SelectedValuePath = "StatusID";

CmbStatusID.DisplayMemberPath = "SName";

if (selectedOrders != null)

\_currentOrders = selectedOrders;

DataContext = \_currentOrders;

}

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

{

StringBuilder error = new StringBuilder();

if (\_currentOrders.StoreID== null)

error.AppendLine("Укажите Магазин!");

if (\_currentOrders.WayToGetID == null)

error.AppendLine("Укажите Способ получения!");

if (\_currentOrders.DeliveryID == null)

error.AppendLine("Укажите Код доставки!");

if (\_currentOrders.ClientID == null)

error.AppendLine("Укажите Клиента!");

if (\_currentOrders.StatusID == null)

error.AppendLine("Укажите Статус заказа!");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

//если пользователь новый

if (\_currentOrders.OrdersID == 0)

EStoreEntities.GetContext().Orders.Add(\_currentOrders);

try

{

EStoreEntities.GetContext().SaveChanges();

MessageBox.Show("Данные сохранены");

ClassFrame.frmObj.Navigate(new PageTable());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

private void DP1\_SelectedDateChanged(object sender, SelectionChangedEventArgs e)

{

string t = cboTP.Text;

string d = DP1.Text;

DateTime dt = DateTime.Parse(d +" "+ t);

TxtDateTime.Text= dt.ToString();

}

}

**PageEditOrdersGoods.xaml.cs**

public partial class PageEditOrdersGoods : Page

{

private OrdersGoods \_currentOrdersGoods = new OrdersGoods();

public PageEditOrdersGoods(OrdersGoods selectedOrdersGoods)

{

InitializeComponent();

CmbOrdersID.ItemsSource = EStoreEntities.GetContext().Orders.ToList();

CmbOrdersID.SelectedValuePath = "OrdersID";

CmbOrdersID.DisplayMemberPath = "OrdersID";

CmbGoodsID.ItemsSource = EStoreEntities.GetContext().Goods.ToList();

CmbGoodsID.SelectedValuePath = "GoodsID";

CmbGoodsID.DisplayMemberPath = "GName";

if (selectedOrdersGoods != null)

\_currentOrdersGoods = selectedOrdersGoods;

DataContext = \_currentOrdersGoods;

}

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

{

StringBuilder error = new StringBuilder();

if (\_currentOrdersGoods.OrdersID == null)

error.AppendLine("Укажите код заказа!");

if (\_currentOrdersGoods.GoodsID == null)

error.AppendLine("Укажите товар!");

if (\_currentOrdersGoods.Quantity == null)

error.AppendLine("Укажите кол-во!");

if (\_currentOrdersGoods.Quantity < 1)

error.AppendLine("Кол-во не может быть меньше 1!");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

//если пользователь новый

if (\_currentOrdersGoods.id == 0)

EStoreEntities.GetContext().OrdersGoods.Add(\_currentOrdersGoods);

try

{

EStoreEntities.GetContext().SaveChanges();

MessageBox.Show("Данные сохранены");

ClassFrame.frmObj.Navigate(new PageTable());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

**PageEditStore.xaml.cs**

public partial class PageEditStore : Page

{

private Store \_currentStore = new Store();

string imgLoc = "пусто";

public PageEditStore(Store selectedStore)

{

InitializeComponent();

if (selectedStore != null)

\_currentStore = selectedStore;

DataContext = \_currentStore;

}

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

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentStore.Adress))

error.AppendLine("Укажите Адрес");

if (\_currentStore.EMail == null)

error.AppendLine("Укажите Почту!");

if (\_currentStore.Code == null)

error.AppendLine("Укажите Код!");

if (\_currentStore.Code < 0)

error.AppendLine("Код не может быть отрицательным!");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

//если пользователь новый

if (\_currentStore.StoreID == 0)

EStoreEntities.GetContext().Store.Add(\_currentStore);

try

{

if (imgLoc != null && imgLoc != "пусто")

{

byte[] img = null;

FileStream fs = new FileStream(imgLoc, FileMode.Open, FileAccess.Read);

BinaryReader br = new BinaryReader(fs);

img = br.ReadBytes((int)fs.Length);

string filename = imgLoc.Substring(imgLoc.LastIndexOf('\\') + 1);

\_currentStore.Picture = String.Concat("/ResPhoto/", filename);

}

EStoreEntities.GetContext().SaveChanges();

MessageBox.Show("Данные сохранены");

ClassFrame.frmObj.Navigate(new PageTable());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

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

{

try

{

OpenFileDialog dlg = new OpenFileDialog

{

Filter = "JPG Files (\*.jpg)|\*.jpg|PNG Files (\*.png)|\*.png|GIF Files (\*.gif)|\*.gif|All Files (\*.\*)|\*.\*",

Title = "Выберите фото/изображение лекарства"

};

if (dlg.ShowDialog() == true)

{

imgLoc = dlg.FileName.ToString();

imageArtist.Source = new BitmapImage(new Uri(imgLoc));

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

**PageListGoods.xaml.cs**

public partial class PageListGoods : Page

{

public PageListGoods()

{

InitializeComponent();

var \_currentGoods = EStoreEntities.GetContext().Goods.ToList();

LviewGoods.ItemsSource = \_currentGoods;

}

}

**PageListStore.xaml.cs**

public partial class PageListStore : Page

{

public PageListStore()

{

InitializeComponent();

var \_currentStore = EStoreEntities.GetContext().Store.ToList();

LviewStore.ItemsSource = \_currentStore;

}

}