**Содержание**

[**MainWindow** 2](#_Toc127615358)

[**AvtorizPage** 3](#_Toc127615359)

[**CaptchaPage** 6](#_Toc127615360)

[**GostPage** 8](#_Toc127615361)

[**FarmZakazPage** 13](#_Toc127615362)

[**ClientPage** 16](#_Toc127615363)

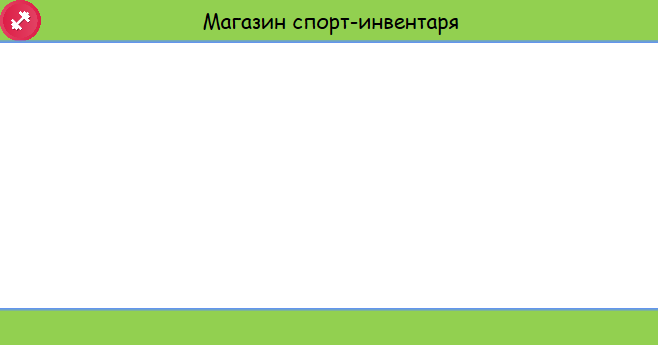
[**AdminPage** 21](#_Toc127615364)

[**ZakazPage** 27](#_Toc127615365)

[**AddAndEditPage** 31](#_Toc127615366)

[**ManaferPage** 35](#_Toc127615367)

# **MainWindow**



MainWindow.xaml

<Window x:Class="OOO\_S\_App.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:OOO\_S\_App"

mc:Ignorable="d"

Title="Магазин спортивных товаров" Height="450" Width="800" Style="{StaticResource window}" WindowState="Maximized">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<Grid Background="#FF92D050">

<Image Source="/Resources/logo.png" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<TextBlock Text="Магазин спорт-инвентаря" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="25"/>

</Grid>

<Grid Grid.Row="1">

<Frame Name="MainFrame" NavigationUIVisibility="Hidden"></Frame>

</Grid>

<Grid Grid.Row="2" Background="#FF92D050">

</Grid>

</Grid>

</Window>

MainWindow.xaml.cs

using OOO\_S\_App.Class;

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;

namespace OOO\_S\_App

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

MainFrame.Navigate(new Pages.AvtorizPage());

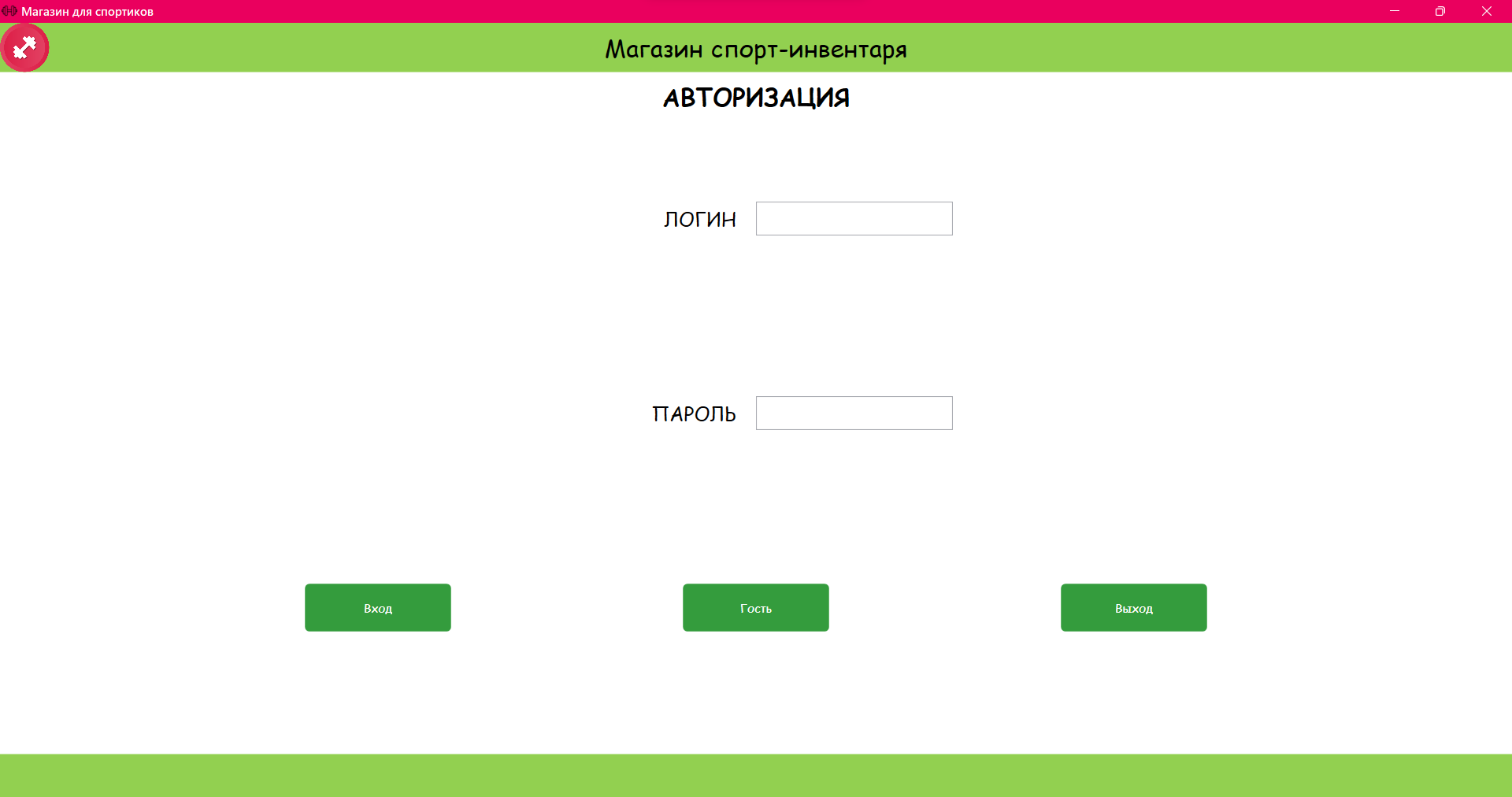
Manager.MainFrame = MainFrame;

}

}

}

# **AvtorizPage**



AvtorizPage.xaml

<Page x:Class="OOO\_S\_App.Pages.AvtorizPage"

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:OOO\_S\_App.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="AvtorizPage" Style="{StaticResource base\_page}">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<TextBlock Text="АВТОРИЗАЦИЯ" FontSize="25" Grid.ColumnSpan="2" VerticalAlignment="Center" HorizontalAlignment="Center" FontWeight="Bold"/>

<TextBlock Text="ЛОГИН" Grid.Row="1" FontSize="20" VerticalAlignment="Center" HorizontalAlignment="Right" Margin="0 0 20 0"/>

<TextBlock Text="ПАРОЛЬ" Grid.Row="2" FontSize="20" VerticalAlignment="Center" HorizontalAlignment="Right" Margin="0 0 20 0"/>

<TextBox x:Name="TB" Grid.Column="1" Grid.Row="1" Width="200" Height="35" VerticalAlignment="Center" HorizontalAlignment="Left"/>

<PasswordBox x:Name="PB" Grid.Row="2" Grid.Column="1" Width="200" Height="35" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<Button x:Name="gost" Content="Гость" ToolTip="\*Войти как гость\*" Style="{StaticResource ButtonRadius}" Width="150" Grid.Row="3" Grid.ColumnSpan="2" Height="50" Click="gost\_Click"/>

<Button x:Name="vhod" Content="Вход" HorizontalAlignment="Center" VerticalAlignment="Center" Click="vhod\_Click" Style="{StaticResource ButtonRadius}" Width="150" Height="50" Grid.Row="3"/>

<Button x:Name="omena" Content="Выход" HorizontalAlignment="Center" VerticalAlignment="Center" Click="omena\_Click" Style="{StaticResource ButtonRadius}" Width="150" Height="50" Grid.Row="3" Grid.Column="1"/>

</Grid>

</Page>

AvtorizPage.xaml.cs

using OOO\_S\_App.Base;

using OOO\_S\_App.Class;

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;

namespace OOO\_S\_App.Pages

{

/// <summary>

/// Логика взаимодействия для AvtorizPage.xaml

/// </summary>

public partial class AvtorizPage : Page

{

short \_attempts;

public AvtorizPage()

{

InitializeComponent();

\_attempts = 0;

}

//логика взаимодействия с кнопкой Гость

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

{

MessageBox.Show("Вход выполнен. Статус пользователя - Гость");

NavigationService.Navigate(new Pages.GostPage());

}

//логика взаимодействия с кнопкой Вход, учитывая введенные данные

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

{

try

{

\_attempts++;

User user = TradeEntities.GetContext().Users.FirstOrDefault(u => u.U\_Login == TB.Text && u.U\_Pass == PB.Password);

if (user != null)

{

if (user.U\_Role\_ID == 1)

{

Auth.U\_Familya = user.U\_Familya;

Auth.U\_Imya = user.U\_Imya;

Auth.U\_Otchestvo = user.U\_Otchestvo;

MessageBox.Show("Вход выполнен. Статус пользователя - Клиент");

NavigationService.Navigate(new Pages.ClientPage());

}

else if (user.U\_Role\_ID == 2)

{

Auth.U\_Familya = user.U\_Familya;

Auth.U\_Imya = user.U\_Imya;

Auth.U\_Otchestvo = user.U\_Otchestvo;

MessageBox.Show("Вход выполнен. Статус пользователя - Администратор");

NavigationService.Navigate(new Pages.AdminPage());

}

else if (user.U\_Role\_ID == 3)

{

Auth.U\_Familya = user.U\_Familya;

Auth.U\_Imya = user.U\_Imya;

Auth.U\_Otchestvo = user.U\_Otchestvo;

MessageBox.Show("Вход выполнен. Статус пользователя - Менеджер");

NavigationService.Navigate(new Pages.ManagerPage());

}

}

else

{

if (\_attempts < 2)

{

MessageBox.Show("Неверный логин и/или пароль");

}

else

{

NavigationService.Navigate(new Pages.CaptchaPage());

}

}

}

catch (Exception ex)

{

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

}

}

//логика взаимодействия с кнопкой Выход, где окно авторизации по нажатию на кнопку закрывается

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

{

App.Current.Shutdown();

}

}

}

# **CaptchaPage**



CaptchaPage.xaml

<Page x:Class="OOO\_S\_App.Pages.CaptchaPage"

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:OOO\_S\_App.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="CaptchaPage" Style="{StaticResource base\_page}">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="60"/>

<RowDefinition Height="60"/>

</Grid.RowDefinitions>

<Image Name="CaptchaImage" Source="{Binding CaptchaPath}" Width="480" Height="200" />

<TextBox Name="CaptchaTb" Grid.Row="1" VerticalAlignment="Center" FontSize="30" Width="300"/>

<Button Name="CheckCaptcha" Click="CheckCaptcha\_Click" Content="OK" Grid.Row="2" Style="{StaticResource ButtonRadius}" HorizontalAlignment="Center" VerticalAlignment="Center"/>

</Grid>

</Page>

CaptchaPage.xaml.cs

using OOO\_S\_App.Class;

using System;

using System.Collections.Generic;

using System.IO;

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;

namespace OOO\_S\_App.Pages

{

/// <summary>

/// Логика взаимодействия для CaptchaPage.xaml

/// </summary>

public partial class CaptchaPage : Page

{

string \_correctCaptcha { get; set; }

public bool \_isCorrect { get; set; } = false;

public CaptchaPage()

{

InitializeComponent();

\_correctCaptcha = Captcha.CreateCaptcha();

CaptchaImage.Source = new BitmapImage(new Uri($@"{Directory.GetCurrentDirectory()}\Image\{\_correctCaptcha}.png"));

}

//проверка написания капчи, где при корректном прохождении пользователь переходит обратно на страницу авторизации, при некорректном - выводится уведомление об ошибке

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

{

if (\_correctCaptcha != CaptchaTb.Text)

{

MessageBox.Show("Каптча введена не верно");

NavigationService.Navigate(new Pages.CaptchaPage());

}

\_isCorrect = true;

if (\_correctCaptcha == CaptchaTb.Text)

{

NavigationService.Navigate(new Pages.AvtorizPage());

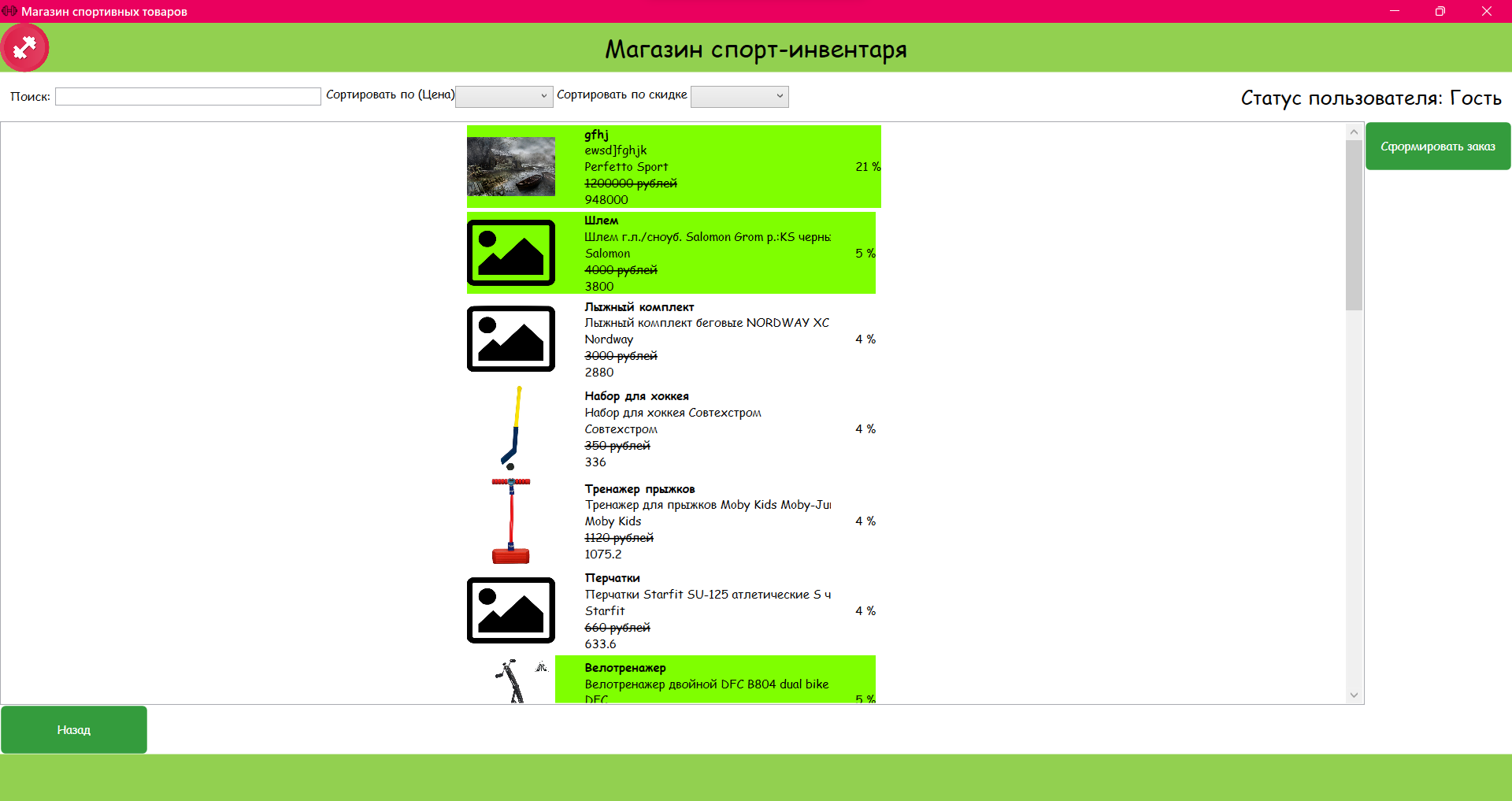
}

}

}

}

# **GostPage**



GostPage.xaml

<Page x:Class="OOO\_S\_App.Pages.GostPage"

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:OOO\_S\_App.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="GostPage" Style="{StaticResource base\_page}" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="150"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<StackPanel Grid.Column="0" Grid.ColumnSpan="2" HorizontalAlignment="Right" VerticalAlignment="Center" Margin="10" Orientation="Vertical">

<TextBlock Text="Статус пользователя: Гость" FontSize="20"/>

</StackPanel>

<Grid>

<WrapPanel VerticalAlignment="Center" HorizontalAlignment="Left" Margin="10 0 0 0">

<StackPanel Orientation="Horizontal" VerticalAlignment="Center">

<TextBlock Text="Поиск:" />

<TextBox x:Name="TBP" Width="270" Margin="5,0,5,0" TextChanged="TBP\_TextChanged"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Сортировать по (Цена)"/>

<ComboBox Name="CBS" Width="100" DisplayMemberPath="" SelectionChanged="CBS\_SelectionChanged">

<ComboBoxItem Content="По возрастанию"/>

<ComboBoxItem Content="По убыванию"/>

</ComboBox>

</StackPanel>

<StackPanel Orientation="Horizontal" VerticalAlignment="Center">

<TextBlock Text=" Сортировать по скидке "/>

<ComboBox Name="CBBEB" SelectionChanged="CBBEB\_SelectionChanged" Width="100">

<ComboBoxItem Content="Все"></ComboBoxItem>

<ComboBoxItem Content="0-10%"></ComboBoxItem>

<ComboBoxItem Content="11-15%"></ComboBoxItem>

<ComboBoxItem Content="16+%"></ComboBoxItem>

</ComboBox>

</StackPanel>

</WrapPanel>

</Grid>

<Grid Grid.Row="1">

<Grid.RowDefinitions>

</Grid.RowDefinitions>

<ListView x:Name="LVTovar" Grid.Row="1" ScrollViewer.HorizontalScrollBarVisibility="Disabled">

<ListView.ItemsPanel >

<ItemsPanelTemplate>

<WrapPanel Orientation="Vertical" HorizontalAlignment="Center"/>

</ItemsPanelTemplate>

</ListView.ItemsPanel>

<ListView.ItemTemplate>

<DataTemplate>

<Grid Background="{Binding GetColor}">

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<StackPanel Orientation="Horizontal" Height="auto" Width="auto">

<Image Width="90" HorizontalAlignment="Center" VerticalAlignment="Center" Source="{Binding Path=GetPhoto}"/>

<StackPanel Grid.Column="1" Margin="30 0 25 0" Width="250" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock FontWeight="Bold" Text="{Binding Path=T\_Spisok\_ID}"/>

<TextBlock Text="{Binding Path=T\_Opis}"/>

<TextBlock Text="{Binding Path=Proizvod.P\_Name}"/>

<TextBlock Text="{Binding Path=T\_Stoimost, StringFormat={}{0} рублей}" TextDecorations="{Binding Path=GetTextDecotation}"/>

<TextBlock Text="{Binding ProductPriceWithDiscount}"/>

</StackPanel>

<StackPanel Grid.Column="2" Margin="0 0 0 0" VerticalAlignment="Center" HorizontalAlignment="Right">

<TextBlock Text="{Binding Path=T\_Destv\_Skidka, StringFormat={}{0} %}"/>

</StackPanel>

</StackPanel>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

<Button x:Name="farm\_zakaz" Content="Сформировать заказ" HorizontalAlignment="Center" VerticalAlignment="Top" Click="farm\_zakaz\_Click" Style="{StaticResource ButtonRadius}" Grid.Column="1" Width="150" Height="50" Grid.Row="1"/>

<Button x:Name="back" Content="Назад" Grid.Row="2" Style="{StaticResource ButtonRadius}" Width="150" Height="50" VerticalAlignment="Center" HorizontalAlignment="Left" Click="back\_Click"/>

</Grid>

</Page>

GostPage.xaml.cs

using OOO\_S\_App.Base;

using OOO\_S\_App.Class;

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;

namespace OOO\_S\_App.Pages

{

/// <summary>

/// Логика взаимодействия для GostPage.xaml

/// </summary>

public partial class GostPage : Page

{

public GostPage()

{

InitializeComponent();

}

//кнопка назад возвращает пользователя на предыдущую страницу

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

{

NavigationService.Navigate(new Pages.AvtorizPage());

}

//кнопка перехода на формирование заказа

private void farm\_zakaz\_Click(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new Pages.FarmZakazPage(null));

}

//оператор случая, необходимый для корректного отображения данных

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

TradeEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

LVTovar.ItemsSource = TradeEntities.GetContext().Tovars.OrderBy(p => p.T\_Artikyl\_ID).ToList();

}

}

//поиск по наименованию товара

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

{

UpdateData();

}

//сортировка по стоимости товара

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

{

UpdateData();

}

//логика для осуществления поиска и сортировки по стоимости

private void UpdateData()

{

var currentProduct = TradeEntities.GetContext().Tovars.OrderBy(p => p.T\_Spisok\_ID).ToList();

currentProduct = currentProduct.Where(p => p.T\_Spisok\_ID.ToLower().Contains(TBP.Text.ToLower())).ToList();

if (CBS.SelectedIndex >= 0)

{

if (CBS.SelectedIndex == 0)

{

currentProduct = currentProduct.OrderBy(p => p.T\_Stoimost).ToList();

}

if (CBS.SelectedIndex == 1)

{

currentProduct = currentProduct.OrderByDescending(p => p.T\_Stoimost).ToList();

}

}

LVTovar.ItemsSource = currentProduct;

}

//фильтрация по скидке товара

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

{

UpdateData1();

}

//логика для осуществления фильтрации, сортировки и поиска

private void UpdateData1()

{

var currentProduct = TradeEntities.GetContext().Tovars.OrderBy(p => p.T\_Spisok\_ID).ToList();

currentProduct = currentProduct.Where(p => p.T\_Spisok\_ID.ToLower().Contains(TBP.Text.ToLower())).ToList();

if (CBS.SelectedIndex >= 0)

{

if (CBS.SelectedIndex == 0)

{

currentProduct = currentProduct.OrderBy(p => p.T\_Stoimost).ToList();

}

if (CBS.SelectedIndex == 1)

{

currentProduct = currentProduct.OrderByDescending(p => p.T\_Stoimost).ToList();

}

}

if (CBBEB.SelectedIndex > 0)

{

int a = 0;

int b = 0;

switch (CBBEB.SelectedIndex)

{

case 1:

a = 0;

b = 10;

break;

case 2:

a = 11;

b = 15;

break;

case 3:

a = 16;

b = 100;

break;

}

currentProduct = currentProduct.Where(p => p.T\_Destv\_Skidka > a && p.T\_Destv\_Skidka <= b).ToList();

}

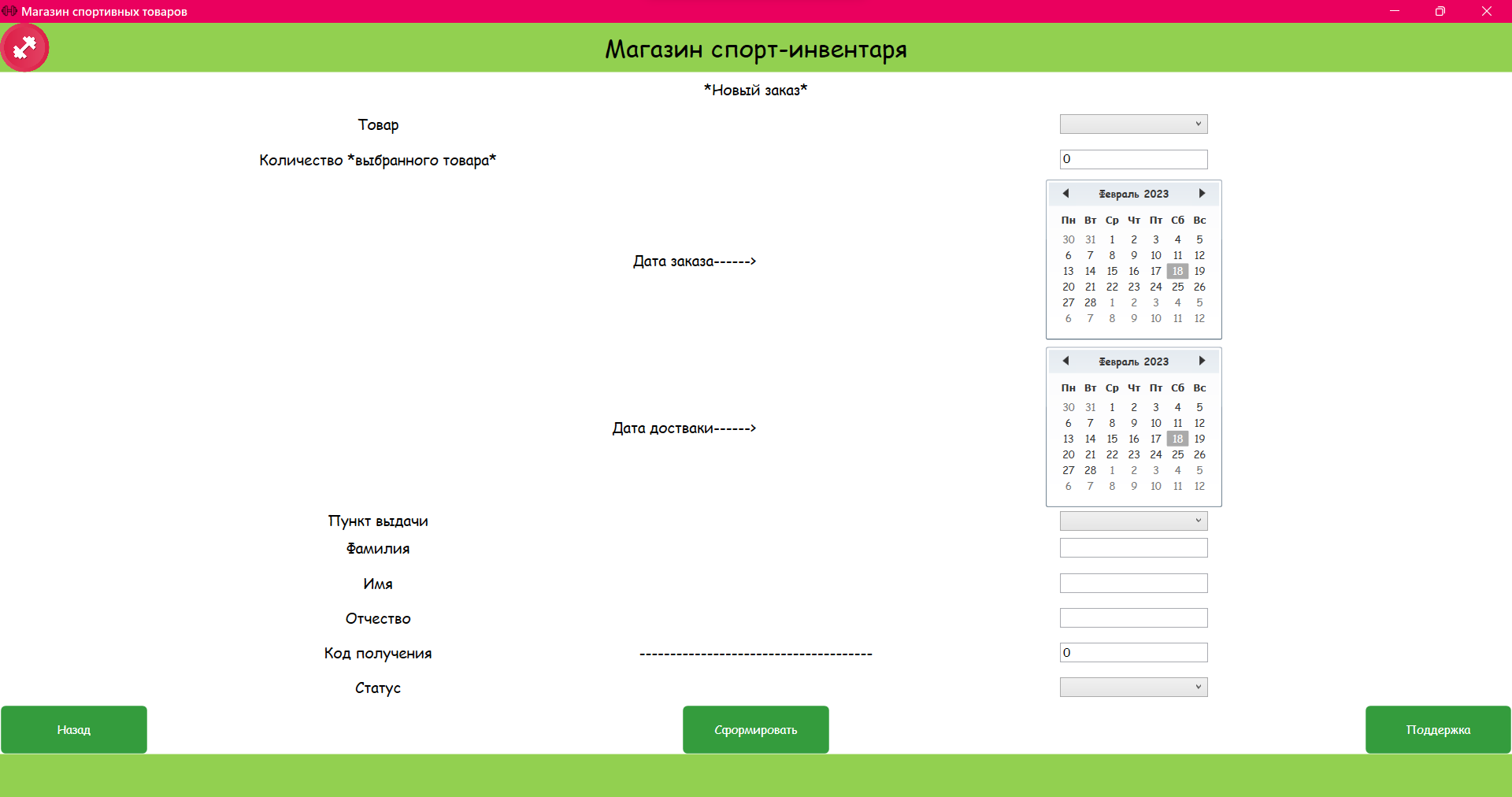
LVTovar.ItemsSource = currentProduct;

}

}

}

# **FarmZakazPage**



FarmZakaz.xaml

<Page x:Class="OOO\_S\_App.Pages.FarmZakazPage"

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:OOO\_S\_App.Pages"

mc:Ignorable="d"

d:DesignHeight="950" d:DesignWidth="1200"

Title="FarmZakazPage" Style="{StaticResource base\_page}">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="170"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<Button x:Name="back" Content="Назад" Grid.Row="11" Style="{StaticResource ButtonRadius}" Width="150" Height="50" VerticalAlignment="Center" HorizontalAlignment="Left" Click="back\_Click"/>

<Button x:Name="new" Content="Сформировать" Grid.Row="11" Grid.ColumnSpan="2" Style="{StaticResource ButtonRadius}" Width="150" Height="50" VerticalAlignment="Center" HorizontalAlignment="Center" Click="new\_Click"/>

<Button x:Name="TP" Content="Поддержка" Grid.Row="11" Grid.Column="1" Style="{StaticResource ButtonRadius}" Width="150" Height="50" VerticalAlignment="Center" HorizontalAlignment="Right" Click="TP\_Click"/>

<TextBlock Text="Товар" VerticalAlignment="Center" Grid.Row="1" HorizontalAlignment="Center" FontSize="15"/>

<TextBlock Text="Количество \*выбранного товара\*" Grid.Row="2" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="15"/>

<TextBlock Text="Дата заказа------>" Grid.Row="3" VerticalAlignment="Center" HorizontalAlignment="Right" FontSize="15"/>

<TextBlock Text="Дата достваки------>" Grid.Row="4" VerticalAlignment="Center" HorizontalAlignment="Right" FontSize="15"/>

<TextBlock Text="Пункт выдачи" Grid.Row="5" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="15"/>

<TextBlock Text="Фамилия" Grid.Row="6" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="15"/>

<TextBlock Text="Имя" Grid.Row="7" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="15"/>

<TextBlock Text="Отчество" Grid.Row="8" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="15"/>

<TextBlock Text="--------------------------------------" Grid.ColumnSpan="2" Grid.Row="9" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="15"/>

<TextBlock Text="\*Новый заказ\*" VerticalAlignment="Center" Grid.ColumnSpan="2" HorizontalAlignment="Center" FontSize="15"/>

<Calendar x:Name="calend" Grid.Row="3" Grid.Column="1" FirstDayOfWeek="Monday" Visibility="Visible" HorizontalAlignment="Center" SelectedDate="{Binding Z\_Data\_Zakaza}" Width="auto" Height="auto"/>

<Calendar x:Name="calend1" Grid.Row="4" Grid.Column="1" FirstDayOfWeek="Monday" SelectedDate="{Binding Z\_Data\_Dostav}" HorizontalAlignment="Center" VerticalAlignment="Center" Width="auto" Height="auto"/>

<ComboBox x:Name="CBTovar" Grid.Row="1" Grid.Column="1" SelectedItem="{Binding Tovar}" DisplayMemberPath="T\_Artikyl\_ID" Width="150" Height="20"/>

<TextBox x:Name="TBKol" Grid.Column="1" Grid.Row="2" Text="{Binding Z\_Kolich}" Width="150" Height="20" HorizontalAlignment="Center" VerticalAlignment="Center"/>

<ComboBox x:Name="CBPV" Grid.Row="5" Grid.Column="1" SelectedItem="{Binding Pynkt\_vid}" DisplayMemberPath="P\_Ylica" Width="150" Height="20" Margin="225,1,225,0" VerticalAlignment="Top"/>

<TextBox x:Name="TBFam" Grid.Column="1" Grid.Row="6" Text="{Binding Z\_Famil}" Width="150" Height="20" HorizontalAlignment="Center" VerticalAlignment="Center"/>

<TextBox x:Name="TBIm" Grid.Column="1" Grid.Row="7" Text="{Binding Z\_Imya}" Width="150" Height="20" HorizontalAlignment="Center" VerticalAlignment="Center"/>

<TextBox x:Name="TBOtch" Grid.Column="1" Grid.Row="8" Text="{Binding Z\_Otchestvo}" Width="150" Height="20" HorizontalAlignment="Center" VerticalAlignment="Center"/>

<TextBlock Text="Код получения" Grid.Row="9" FontSize="15" VerticalAlignment="Center" HorizontalAlignment="Center"/>

<TextBlock Text="Статус" Grid.Row="10" FontSize="15" VerticalAlignment="Center" HorizontalAlignment="Center"/>

<TextBox x:Name="TBKP" Grid.Column="1" Visibility="Visible" Text="{Binding Z\_Kod\_Polych}" Grid.Row="9" Width="150" Height="20" HorizontalAlignment="Center" VerticalAlignment="Center" />

<ComboBox x:Name="TBStatus" Visibility="Visible" Grid.Column="1" Grid.Row="10" SelectedItem="{Binding Status}" DisplayMemberPath="S\_Name" Width="150" Height="20" HorizontalAlignment="Center" VerticalAlignment="Center" />

</Grid>

</Page>

FarmZakaz.xaml.cs

using OOO\_S\_App.Base;

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;

namespace OOO\_S\_App.Pages

{

/// <summary>

/// Логика взаимодействия для FarmZakazPage.xaml

/// </summary>

public partial class FarmZakazPage : Page

{

public static Zakaz \_zak { get; set; }

public FarmZakazPage(Zakaz selectedep)

{

InitializeComponent();

\_zak = selectedep ?? new Zakaz();

DataContext = \_zak;

CBTovar.ItemsSource = TradeEntities.GetContext().Tovars.ToList();

CBPV.ItemsSource = TradeEntities.GetContext().Pynkt\_vid.ToList();

TBStatus.ItemsSource= TradeEntities.GetContext().Status.ToList();

}

//кнопка назад возвращает пользователя на предыдущую страницу

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

{

NavigationService.GoBack();

}

//кнопка оформления заказа с уведомлением об успешном оформлении

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

{

try

{

\_zak.Z\_Pynkt\_vid\_ID = (CBPV.SelectedItem as Pynkt\_vid).Pynkt\_vid\_ID;

\_zak.Z\_Sostav = (CBTovar.SelectedItem as Tovar).T\_Artikyl\_ID;

\_zak.Z\_Status\_ID = (TBStatus.SelectedItem as Status).Status\_ID;

if (\_zak.Zakaz\_ID == 0)

{

TradeEntities.GetContext().Zakazs.Add(\_zak);

}

TradeEntities.GetContext().SaveChanges();

MessageBox.Show("Вы оформили заказ!");

NavigationService.GoBack();

}

catch (Exception ex)

{

MessageBox.Show(ex.InnerException.Message);

}

}

//кнопка нажатие на которую выдает уведомление о контактах технической поддержки

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

{

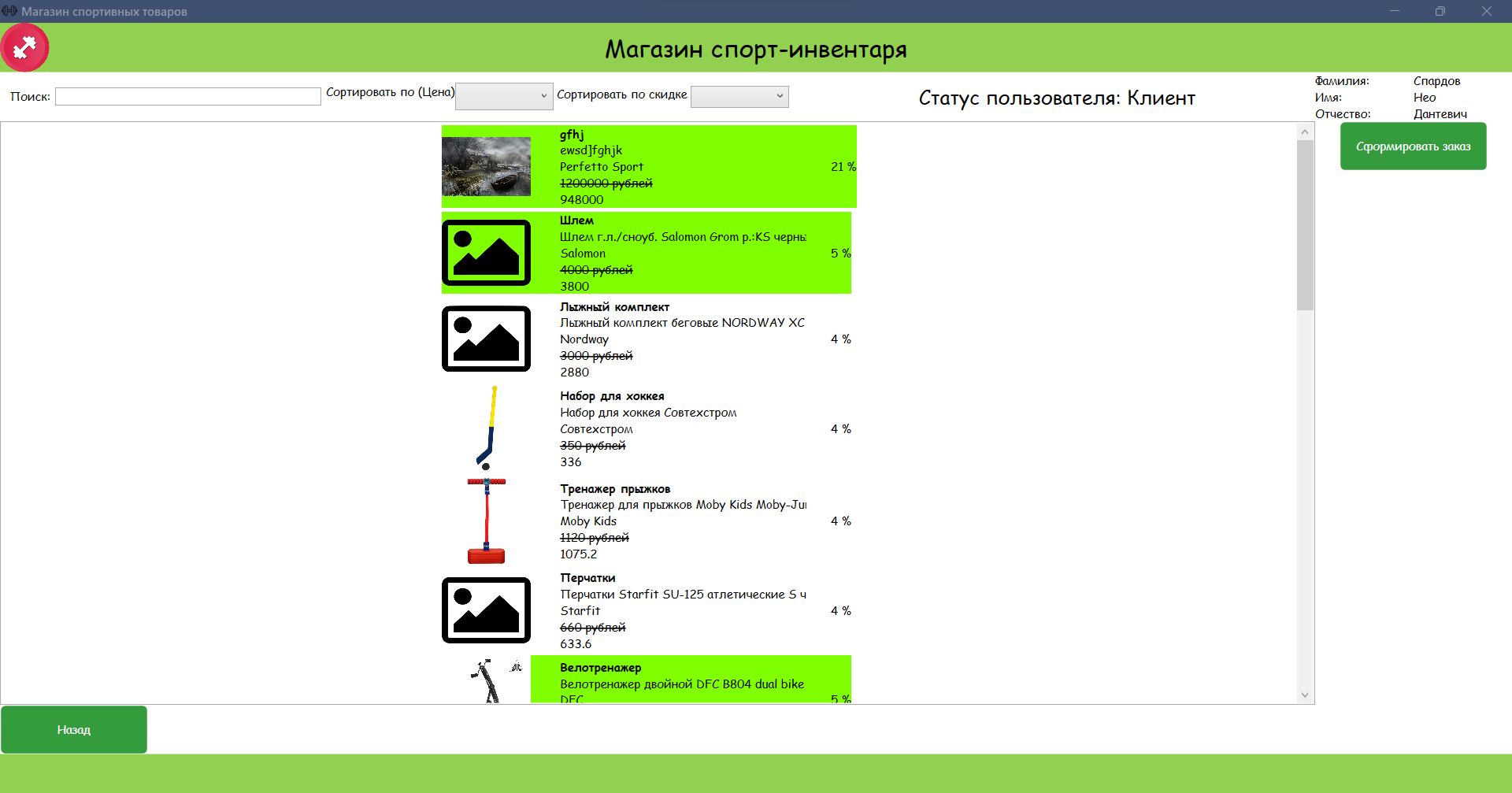
MessageBox.Show("Тел: 89178918913");

}

}

}

# **ClientPage**



ClientPage.xaml

<Page x:Class="OOO\_S\_App.Pages.ClientPage"

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:OOO\_S\_App.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="ClientPage" Style="{StaticResource base\_page}" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="100"/>

<ColumnDefinition Width="100"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<Grid>

<WrapPanel VerticalAlignment="Center" HorizontalAlignment="Left" Margin="10 0 0 0">

<StackPanel Orientation="Horizontal" VerticalAlignment="Center">

<TextBlock Text="Поиск:" />

<TextBox x:Name="TBP" Width="270" Margin="5,0,5,0" TextChanged="TBP\_TextChanged"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Сортировать по (Цена)"/>

<ComboBox Name="CBS" DisplayMemberPath="" Width="100" SelectionChanged="CBS\_SelectionChanged">

<ComboBoxItem Content="По возрастанию"/>

<ComboBoxItem Content="По убыванию"/>

</ComboBox>

</StackPanel>

<StackPanel Orientation="Horizontal" VerticalAlignment="Center">

<TextBlock Text=" Сортировать по скидке "/>

<ComboBox Name="CBBEB" SelectionChanged="CBBEB\_SelectionChanged" Width="100">

<ComboBoxItem Content="Все"></ComboBoxItem>

<ComboBoxItem Content="0-10%"></ComboBoxItem>

<ComboBoxItem Content="11-15%"></ComboBoxItem>

<ComboBoxItem Content="16+%"></ComboBoxItem>

</ComboBox>

</StackPanel>

<StackPanel Grid.Column="0" Grid.ColumnSpan="2" HorizontalAlignment="Right" VerticalAlignment="Center" Orientation="Vertical">

<TextBlock Text=" Статус пользователя: Клиент" FontSize="20"/>

</StackPanel>

</WrapPanel>

</Grid>

<Grid Grid.Row="1">

<Grid.RowDefinitions>

</Grid.RowDefinitions>

<ListView x:Name="LVTovar" Grid.Row="1" ScrollViewer.HorizontalScrollBarVisibility="Disabled">

<ListView.ItemsPanel >

<ItemsPanelTemplate>

<WrapPanel Orientation="Vertical" HorizontalAlignment="Center"/>

</ItemsPanelTemplate>

</ListView.ItemsPanel>

<ListView.ItemTemplate>

<DataTemplate>

<Grid Background="{Binding GetColor}">

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<StackPanel Orientation="Horizontal" Height="auto" Width="auto">

<Image Width="90" HorizontalAlignment="Center" VerticalAlignment="Center" Source="{Binding Path=GetPhoto}"/>

<StackPanel Grid.Column="1" Margin="30 0 25 0" Width="250" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock FontWeight="Bold" Text="{Binding Path=T\_Spisok\_ID}"/>

<TextBlock Text="{Binding Path=T\_Opis}"/>

<TextBlock Text="{Binding Path=Proizvod.P\_Name}"/>

<TextBlock Text="{Binding Path=T\_Stoimost, StringFormat={}{0} рублей}" TextDecorations="{Binding Path=GetTextDecotation}"/>

<TextBlock Text="{Binding ProductPriceWithDiscount}"/>

</StackPanel>

<StackPanel Grid.Column="2" Margin="0 0 0 0" VerticalAlignment="Center" HorizontalAlignment="Right">

<TextBlock Text="{Binding Path=T\_Destv\_Skidka, StringFormat={}{0} %}"/>

</StackPanel>

</StackPanel>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

<StackPanel Grid.Column="1" Orientation="Vertical" HorizontalAlignment="Left" VerticalAlignment="Top">

<TextBlock Text="Фамилия:"/>

<TextBlock Text="Имя:"/>

<TextBlock Text="Отчество:"/>

</StackPanel>

<StackPanel Grid.Column="2" Orientation="Vertical" Width="100" HorizontalAlignment="Center" VerticalAlignment="Top">

<TextBlock x:Name="TbFamil" Text="{Binding U\_Familya}"/>

<TextBlock x:Name="TbImya" Text="{Binding U\_Imya}"/>

<TextBlock x:Name="TbOtchestvo" Text="{Binding U\_Otchestvo}"/>

</StackPanel>

<Button x:Name="farm\_zakaz" Content="Сформировать заказ" HorizontalAlignment="Center" VerticalAlignment="Top" Click="farm\_zakaz\_Click" Style="{StaticResource ButtonRadius}" Grid.ColumnSpan="2" Grid.Column="1" Width="150" Height="50" Grid.Row="1"/>

<Button x:Name="back" Content="Назад" Grid.Row="2" Style="{StaticResource ButtonRadius}" Width="150" Height="50" VerticalAlignment="Center" HorizontalAlignment="Left" Click="back\_Click"/>

</Grid>

</Page>

ClientPage.xaml.cs

using OOO\_S\_App.Base;

using OOO\_S\_App.Class;

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;

namespace OOO\_S\_App.Pages

{

/// <summary>

/// Логика взаимодействия для ClientPage.xaml

/// </summary>

public partial class ClientPage : Page

{

public ClientPage()

{

InitializeComponent();

DataContext = TradeEntities.GetContext().Users.ToList();

List<User> users = TradeEntities.GetContext().Users.ToList();

TbFamil.Text = Auth.U\_Familya;

TbImya.Text = Auth.U\_Imya;

TbOtchestvo.Text = Auth.U\_Otchestvo;

}

//кнопка назад возвращает пользователя на предыдущую страницу

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

{

NavigationService.Navigate(new Pages.AvtorizPage());

}

//оператор случая, необходимый для корректного отображения данных

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

TradeEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

LVTovar.ItemsSource = TradeEntities.GetContext().Tovars.OrderBy(p => p.T\_Artikyl\_ID).ToList();

}

}

//кнопка, по нажатию на которую пользователь будет перенаправлен на страницу заказа

private void farm\_zakaz\_Click(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new Pages.FarmZakazPage(null));

}

//поиск по наименованию товара

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

{

UpdateData();

}

//сортировка по стоимости товара

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

{

UpdateData();

}

//логика для осуществления поиска и сортировки по стоимости

private void UpdateData()

{

var currentProduct = TradeEntities.GetContext().Tovars.OrderBy(p => p.T\_Spisok\_ID).ToList();

currentProduct = currentProduct.Where(p => p.T\_Spisok\_ID.ToLower().Contains(TBP.Text.ToLower())).ToList();

if (CBS.SelectedIndex >= 0)

{

if (CBS.SelectedIndex == 0)

{

currentProduct = currentProduct.OrderBy(p => p.T\_Stoimost).ToList();

}

if (CBS.SelectedIndex == 1)

{

currentProduct = currentProduct.OrderByDescending(p => p.T\_Stoimost).ToList();

}

}

LVTovar.ItemsSource = currentProduct;

}

//фильтрация по скидке товара

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

{

UpdateData1();

}

//логика для осуществления фильтрации, сортировки и поиска

private void UpdateData1()

{

var currentProduct = TradeEntities.GetContext().Tovars.OrderBy(p => p.T\_Spisok\_ID).ToList();

currentProduct = currentProduct.Where(p => p.T\_Spisok\_ID.ToLower().Contains(TBP.Text.ToLower())).ToList();

if (CBS.SelectedIndex >= 0)

{

if (CBS.SelectedIndex == 0)

{

currentProduct = currentProduct.OrderBy(p => p.T\_Stoimost).ToList();

}

if (CBS.SelectedIndex == 1)

{

currentProduct = currentProduct.OrderByDescending(p => p.T\_Stoimost).ToList();

}

}

if (CBBEB.SelectedIndex > 0)

{

int a = 0;

int b = 0;

switch (CBBEB.SelectedIndex)

{

case 1:

a = 0;

b = 10;

break;

case 2:

a = 11;

b = 15;

break;

case 3:

a = 16;

b = 100;

break;

}

currentProduct = currentProduct.Where(p => p.T\_Destv\_Skidka > a && p.T\_Destv\_Skidka <= b).ToList();

}

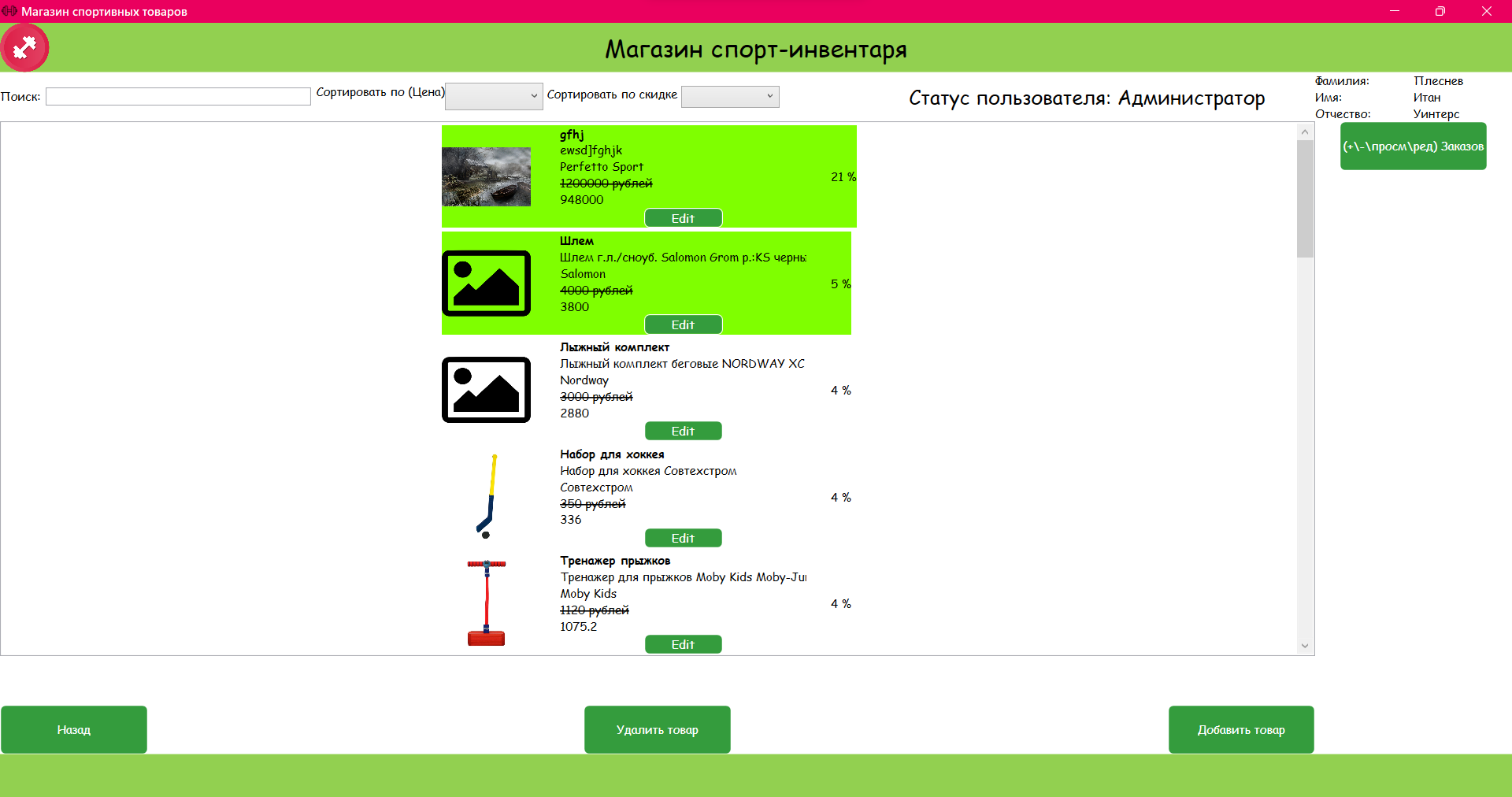
LVTovar.ItemsSource = currentProduct;

}

}

}

# **AdminPage**



AdminPage.xaml

<Page x:Class="OOO\_S\_App.Pages.AdminPage"

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:OOO\_S\_App.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="AdminPage" Style="{StaticResource base\_page}" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="100"/>

<ColumnDefinition Width="100"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="50"/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<Grid Grid.Row="1">

<Grid.RowDefinitions>

</Grid.RowDefinitions>

<ListView x:Name="LVTovar" Grid.Row="1" ScrollViewer.HorizontalScrollBarVisibility="Disabled">

<ListView.ItemsPanel >

<ItemsPanelTemplate>

<WrapPanel Orientation="Vertical" HorizontalAlignment="Center"/>

</ItemsPanelTemplate>

</ListView.ItemsPanel>

<ListView.ItemTemplate>

<DataTemplate>

<Grid Background="{Binding GetColor}">

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<StackPanel Orientation="Horizontal" Height="auto" Width="auto">

<Image Width="90" HorizontalAlignment="Center" VerticalAlignment="Center" Source="{Binding Path=GetPhoto}"/>

<StackPanel Grid.Column="1" Margin="30 0 25 0" Width="250" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock FontWeight="Bold" Text="{Binding Path=T\_Spisok\_ID}"/>

<TextBlock Text="{Binding Path=T\_Opis}"/>

<TextBlock Text="{Binding Path=Proizvod.P\_Name}"/>

<TextBlock Text="{Binding Path=T\_Stoimost, StringFormat={}{0} рублей}" TextDecorations="{Binding Path=GetTextDecotation}"/>

<TextBlock Text="{Binding ProductPriceWithDiscount}"/>

<Button x:Name="EditBtn" Content="Edit" Click="EditBtn\_Click" VerticalAlignment="Center" Style="{StaticResource ButtonRadius}" Width="80" HorizontalAlignment="Center"/>

</StackPanel>

<StackPanel Grid.Column="2" Margin="0 0 0 0" VerticalAlignment="Center" HorizontalAlignment="Right">

<TextBlock Text="{Binding Path=T\_Destv\_Skidka, StringFormat={}{0} %}"/>

</StackPanel>

</StackPanel>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

<Grid>

<WrapPanel VerticalAlignment="Center" HorizontalAlignment="Left">

<StackPanel Orientation="Horizontal" VerticalAlignment="Center">

<TextBlock Text="Поиск:" />

<TextBox x:Name="TBP" Width="270" Margin="5,0,5,0" TextChanged="TBP\_TextChanged"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Сортировать по (Цена)"/>

<ComboBox Name="CBS" Width="100" DisplayMemberPath="" SelectionChanged="CBS\_SelectionChanged">

<ComboBoxItem Content="По возрастанию"/>

<ComboBoxItem Content="По убыванию"/>

</ComboBox>

</StackPanel>

<StackPanel Orientation="Horizontal" VerticalAlignment="Center">

<TextBlock Text=" Сортировать по скидке "/>

<ComboBox Name="CBBEB" SelectionChanged="CBBEB\_SelectionChanged" Width="100">

<ComboBoxItem Content="Все"/>

<ComboBoxItem Content="0-10%"/>

<ComboBoxItem Content="11-15%"/>

<ComboBoxItem Content="16+%"/>

</ComboBox>

</StackPanel>

<StackPanel Grid.Column="0" Grid.ColumnSpan="2" HorizontalAlignment="Right" VerticalAlignment="Center" Orientation="Vertical">

<TextBlock Text=" Статус пользователя: Администратор" FontSize="20"/>

</StackPanel>

</WrapPanel>

</Grid>

<StackPanel Grid.Column="1" Orientation="Vertical" HorizontalAlignment="Left" VerticalAlignment="Center">

<TextBlock Text="Фамилия:"/>

<TextBlock Text="Имя:"/>

<TextBlock Text="Отчество:"/>

</StackPanel>

<StackPanel Grid.Column="2" Orientation="Vertical" Width="100" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock x:Name="TbFamil" Text="{Binding U\_Familya}"/>

<TextBlock x:Name="TbImya" Text="{Binding U\_Imya}"/>

<TextBlock x:Name="TbOtchestvo" Text="{Binding U\_Otchestvo}"/>

</StackPanel>

<Button x:Name="back" Content="Назад" Grid.Row="3" Style="{StaticResource ButtonRadius}" Width="150" Height="50" VerticalAlignment="Center" HorizontalAlignment="Left" Click="back\_Click"/>

<Button x:Name="zakaz" Content="(+\-\просм\ред) Заказов" HorizontalAlignment="Left" VerticalAlignment="Top" Click="zakaz\_Click" Style="{StaticResource ButtonRadius}" Grid.ColumnSpan="2" Grid.Column="1" Width="150" Height="50" Grid.Row="1" Margin="25,0,0,0"/>

<Button x:Name="addtovar" Grid.Row="3" HorizontalAlignment="Right" Style="{StaticResource ButtonRadius}" VerticalAlignment="Center" Content="Добавить товар" Click="addtovar\_Click" Width="150" Height="50"/>

<Button x:Name="delete" Grid.Row="3" HorizontalAlignment="Center" Style="{StaticResource ButtonRadius}" VerticalAlignment="Center" Content="Удалить товар" Click="delete\_Click" Width="150" Height="50"/>

</Grid>

</Page>

AdminPage.xaml.cs

using OOO\_S\_App.Base;

using OOO\_S\_App.Class;

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;

namespace OOO\_S\_App.Pages

{

/// <summary>

/// Логика взаимодействия для AdminPage.xaml

/// </summary>

public partial class AdminPage : Page

{

public AdminPage()

{

InitializeComponent();

DataContext = TradeEntities.GetContext().Users.ToList();

List<User> users = TradeEntities.GetContext().Users.ToList();

TbFamil.Text = Auth.U\_Familya;

TbImya.Text = Auth.U\_Imya;

TbOtchestvo.Text = Auth.U\_Otchestvo;

}

//кнопка назад возвращает пользователя на предыдущую страницу

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

{

NavigationService.Navigate(new Pages.AvtorizPage());

}

//оператор случая, необходимый для корректного отображения данных

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

TradeEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

LVTovar.ItemsSource = TradeEntities.GetContext().Tovars.OrderBy(p => p.T\_Artikyl\_ID).ToList();

}

}

//кнопка, по нажатию на которую пользователь будет перенаправлен на страницу заказа

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

{

NavigationService.Navigate(new Pages.ZakazPage());

}

//кнопка, по нажатию на которую пользователь будет перенаправлен на страницу добавления товара

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

{

NavigationService.Navigate(new Pages.AddAndEditTovarPage(null));

}

//поиск по наименованию товара

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

{

UpdateData();

}

//сортировка по стоимости товара

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

{

UpdateData();

}

//логика для осуществления поиска и сортировки по стоимости

private void UpdateData()

{

var currentProduct = TradeEntities.GetContext().Tovars.OrderBy(p => p.T\_Spisok\_ID).ToList();

currentProduct = currentProduct.Where(p => p.T\_Spisok\_ID.ToLower().Contains(TBP.Text.ToLower())).ToList();

if (CBS.SelectedIndex >= 0)

{

if (CBS.SelectedIndex == 0)

{

currentProduct = currentProduct.OrderBy(p => p.T\_Stoimost).ToList();

}

if (CBS.SelectedIndex == 1)

{

currentProduct = currentProduct.OrderByDescending(p => p.T\_Stoimost).ToList();

}

}

LVTovar.ItemsSource = currentProduct;

}

//кнопка, нажатие на которую ведет к удалению записи из базы данных

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

{

delete\_tovar();

}

private void delete\_tovar()

{

var selectedTovar = LVTovar.SelectedItems.Cast<Tovar>().ToList();

MessageBoxResult messageBoxResult = MessageBox.Show($"Удалить {selectedTovar.Count()} записей??", "Удаление", MessageBoxButton.OKCancel, MessageBoxImage.Question);

if (messageBoxResult == MessageBoxResult.OK)

{

try

{

Tovar x = selectedTovar[0];

TradeEntities.GetContext().Tovars.Remove(x);

TradeEntities.GetContext().SaveChanges();

MessageBox.Show("Записи удалены");

List<Tovar> tovars = TradeEntities.GetContext().Tovars.OrderBy(p => p.T\_Kolich).ToList();

LVTovar.ItemsSource = null;

LVTovar.ItemsSource = tovars;

}

catch (Exception ex)

{

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

}

}

}

//кнопка, по нажатию на которую пользователь будет перенаправлен на страницу редактирования

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

{

NavigationService.Navigate(new Pages.AddAndEditTovarPage((Tovar)(sender as Button).DataContext));

}

//фильтрация по скидке товара

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

{

UpdateData1();

}

//логика для осуществления фильтрации, сортировки и поиска

private void UpdateData1()

{

var currentProduct = TradeEntities.GetContext().Tovars.OrderBy(p => p.T\_Spisok\_ID).ToList();

currentProduct = currentProduct.Where(p => p.T\_Spisok\_ID.ToLower().Contains(TBP.Text.ToLower())).ToList();

if (CBS.SelectedIndex >= 0)

{

if (CBS.SelectedIndex == 0)

{

currentProduct = currentProduct.OrderBy(p => p.T\_Stoimost).ToList();

}

if (CBS.SelectedIndex == 1)

{

currentProduct = currentProduct.OrderByDescending(p => p.T\_Stoimost).ToList();

}

}

if (CBBEB.SelectedIndex > 0)

{

int a = 0;

int b = 0;

switch (CBBEB.SelectedIndex)

{

case 1:

a = 0;

b = 10;

break;

case 2:

a = 11;

b = 15;

break;

case 3:

a = 16;

b = 100;

break;

}

currentProduct = currentProduct.Where(p => p.T\_Destv\_Skidka > a && p.T\_Destv\_Skidka <= b).ToList();

}

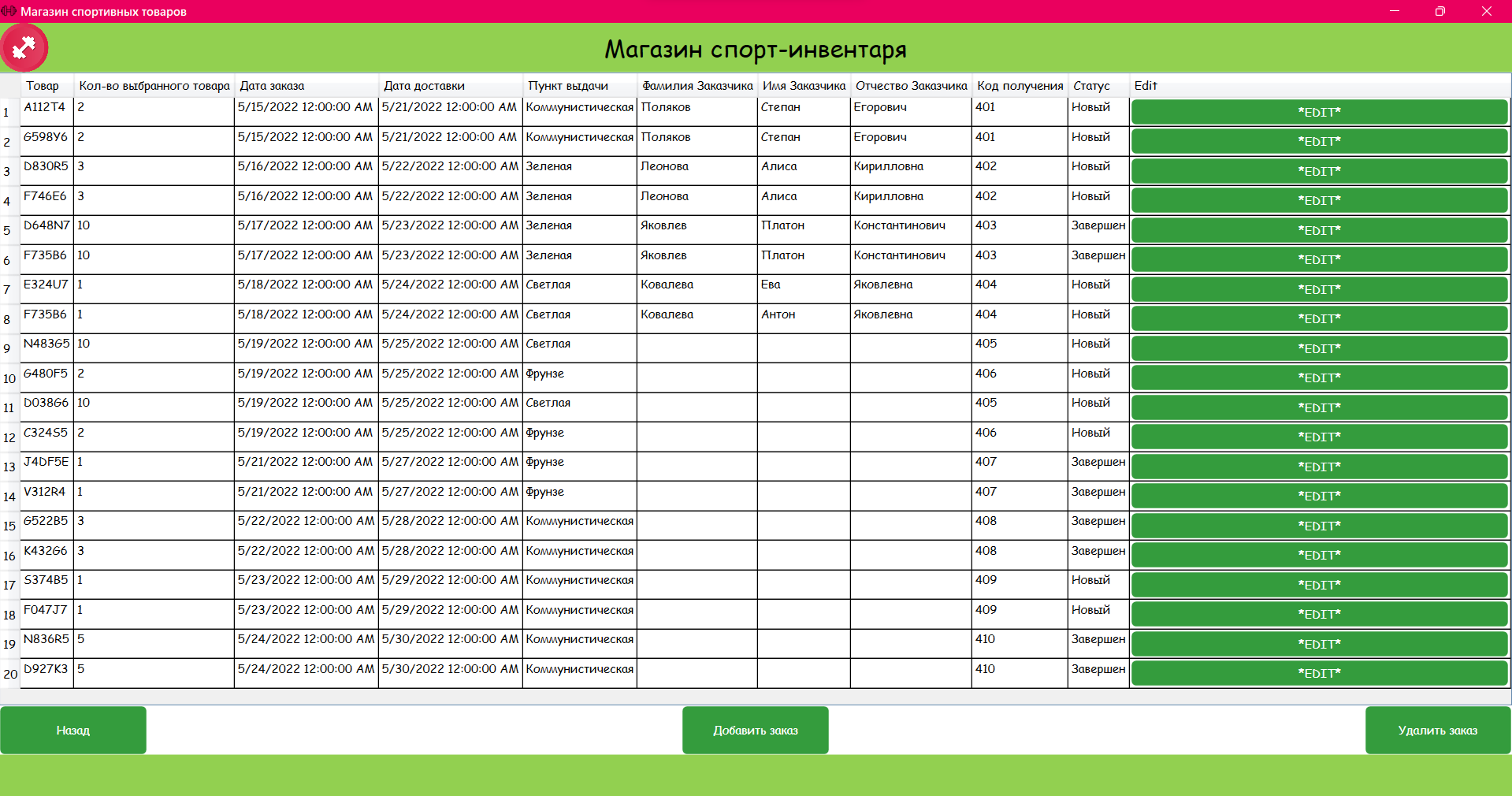
LVTovar.ItemsSource = currentProduct;

}

}

}

# **ZakazPage**



ZakazPage.xaml

<Page x:Class="OOO\_S\_App.Pages.ZakazPage"

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:OOO\_S\_App.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="ZakazPage" Style="{StaticResource base\_page}" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<DataGrid x:Name="Zakaz" AutoGenerateColumns="False" IsReadOnly="True" RowHeight="30" SelectionMode="Single" LoadingRow="Zakaz\_LoadingRow" SelectionChanged="Zakaz\_SelectionChanged">

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding Z\_Sostav}" Header="Товар" Width="auto"/>

<DataGridTextColumn Binding="{Binding Z\_Kolich}" Header="Кол-во выбранного товара" Width="auto"/>

<DataGridTextColumn Binding="{Binding Z\_Data\_Zakaza}" Header="Дата заказа" Width="auto"/>

<DataGridTextColumn Binding="{Binding Z\_Data\_Dostav}" Header="Дата доставки" Width="auto"/>

<DataGridTextColumn Binding="{Binding Pynkt\_vid.P\_Ylica}" Header="Пункт выдачи" Width="auto"/>

<DataGridTextColumn Binding="{Binding Z\_Famil}" Header="Фамилия Заказчика" Width="auto"/>

<DataGridTextColumn Binding="{Binding Z\_Imya}" Header="Имя Заказчика" Width="auto"/>

<DataGridTextColumn Binding="{Binding Z\_Otchestvo}" Header="Отчество Заказчика" Width="auto"/>

<DataGridTextColumn Binding="{Binding Z\_Kod\_Polych}" Header="Код получения" Width="auto"/>

<DataGridTextColumn Binding="{Binding Status.S\_Name}" Header="Статус" Width="auto"/>

<DataGridTemplateColumn Width="\*" Header="Edit">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnEdit" Style="{StaticResource ButtonRadius}" Content="\*EDIT\*" Click="BtnEdit\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<Button x:Name="addzakaz" Content="Добавить заказ" Grid.Row="1" Style="{StaticResource ButtonRadius}" Width="150" Height="50" VerticalAlignment="Center" HorizontalAlignment="Center" Click="addzakaz\_Click"/>

<Button x:Name="deletezakaz" Content="Удалить заказ" Grid.Row="1" Style="{StaticResource ButtonRadius}" Width="150" Height="50" VerticalAlignment="Center" HorizontalAlignment="Right" Click="deletezakaz\_Click"/>

<Button x:Name="back" Content="Назад" Grid.Row="1" Style="{StaticResource ButtonRadius}" Width="150" Height="50" VerticalAlignment="Center" HorizontalAlignment="Left" Click="back\_Click"/>

</Grid>

</Page>

ZakazPage.xaml.cs

using OOO\_S\_App.Base;

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;

namespace OOO\_S\_App.Pages

{

/// <summary>

/// Логика взаимодействия для ZakazPage.xaml

/// </summary>

public partial class ZakazPage : Page

{

public ZakazPage()

{

InitializeComponent();

List<Zakaz> zak = TradeEntities.GetContext().Zakazs.OrderBy(p => p.Z\_Data\_Dostav).ToList();

Zakaz.ItemsSource = zak;

}

//кнопка назад возвращает пользователя на предыдущую страницу

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

{

NavigationService.GoBack();

}

//кнопка перехода на страницу формирования заказа

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

{

NavigationService.Navigate(new Pages.FarmZakazPage(null));

}

//кнопка, нажатие на которую ведет к удалению заказа из базы данных

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

{

var \_selectedzakaz = Zakaz.SelectedItems.Cast<Zakaz>().ToList();

MessageBoxResult messageBoxResult = MessageBox.Show($"Удалить {\_selectedzakaz.Count()} записей???",

"Удаление", MessageBoxButton.OKCancel, MessageBoxImage.Question);

if (messageBoxResult == MessageBoxResult.OK)

{

try

{

Zakaz x = \_selectedzakaz[0];

TradeEntities.GetContext().Zakazs.Remove(x);

TradeEntities.GetContext().SaveChanges();

MessageBox.Show("Записи удалены");

List<Zakaz> tours = TradeEntities.GetContext().Zakazs.OrderBy(p => p.Z\_Data\_Dostav).ToList();

Zakaz.ItemsSource = null;

Zakaz.ItemsSource = tours;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString(), "Ошибка удаления", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

private void Zakaz\_LoadingRow(object sender, DataGridRowEventArgs e)

{

e.Row.Header = (e.Row.GetIndex() + 1).ToString();

}

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

{

}

//кнопка, по нажатию на которую пользователь будет перенаправлен на страницу редактирования

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

{

NavigationService.Navigate(new Pages.FarmZakazPage((sender as Button).DataContext as Zakaz));

}

//оператор случая, необходимый для корректного отображения данных

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

List<Zakaz> zak = TradeEntities.GetContext().Zakazs.OrderBy(p => p.Z\_Data\_Dostav).ToList();

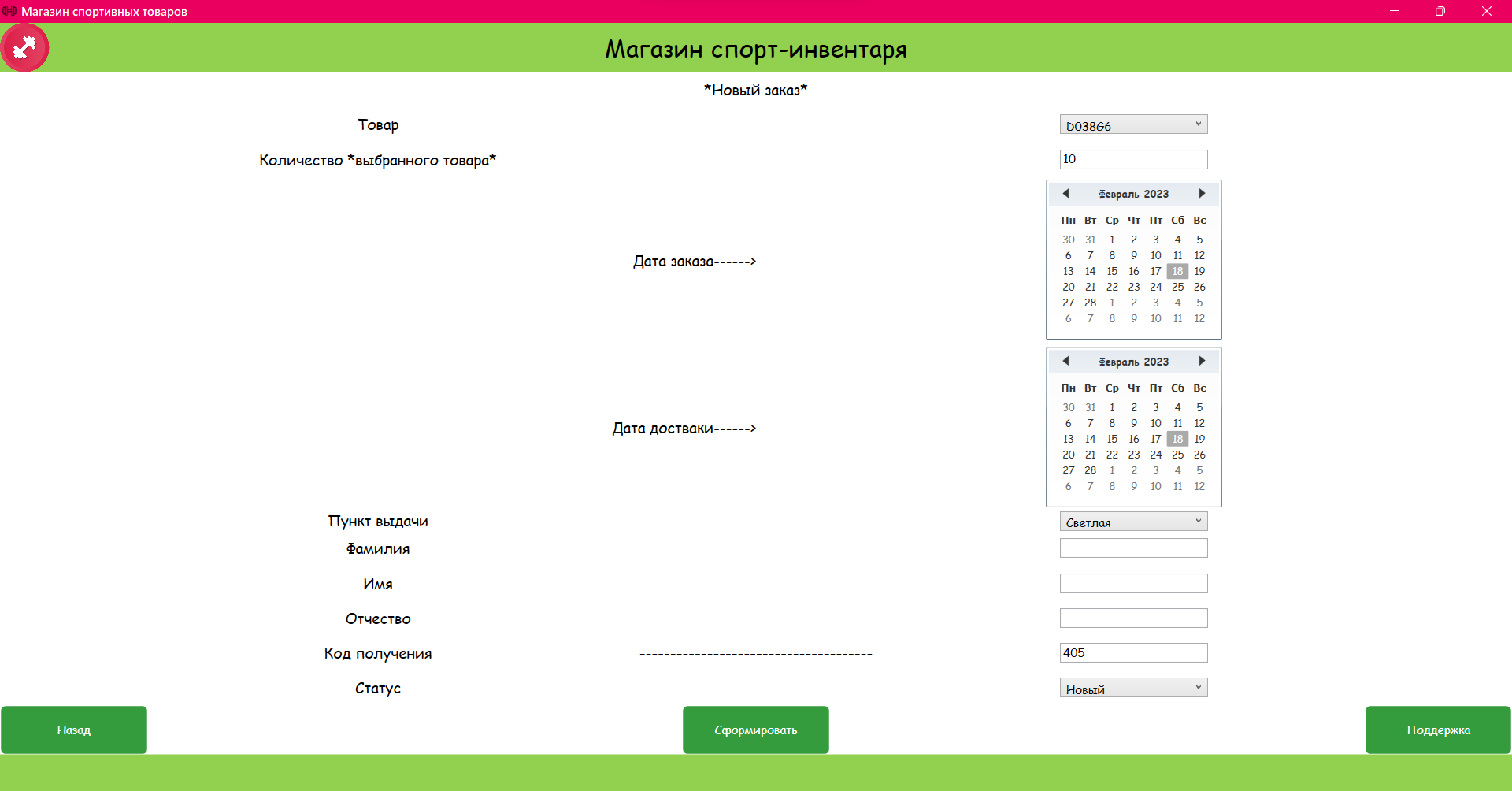
Zakaz.ItemsSource = zak;

}

}

}

# **AddAndEditPage**



AddAndEditPage.xaml

<Page x:Class="OOO\_S\_App.Pages.AddAndEditTovarPage"

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:OOO\_S\_App.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="AddAndEditTovarPage" Style="{DynamicResource base\_page}">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<TextBlock x:Name="ArtikylID" Text="ID" Height="40" HorizontalAlignment="Center"/>

<TextBlock Text="Товар" Height="40" FontSize="20" Margin="5" Grid.Row="1" HorizontalAlignment="Center"/>

<TextBlock Text="Единица измерения" Height="40" FontSize="20" Margin="5" Grid.Row="2" HorizontalAlignment="Center"/>

<TextBlock Text="Стоимость" Height="40" FontSize="20" Margin="5" Grid.Row="3" HorizontalAlignment="Center"/>

<TextBlock Text="Размер" Height="40" FontSize="20" Margin="5" Grid.Row="4" HorizontalAlignment="Center"/>

<TextBlock Text="Производитель" Height="40" FontSize="20" Margin="5" Grid.Row="5" HorizontalAlignment="Center"/>

<TextBlock Text="Поставщик" Height="40" FontSize="20" Margin="5" Grid.Row="6" HorizontalAlignment="Center"/>

<TextBlock Text="Тип товара" Height="40" FontSize="20" Margin="5" Grid.Row="7" HorizontalAlignment="Center"/>

<TextBlock Text="Скидка" Height="40" FontSize="20" Margin="5" Grid.Row="8" HorizontalAlignment="Center"/>

<TextBlock Text="Кол-во" Height="40" FontSize="20" Margin="5" Grid.Row="9" HorizontalAlignment="Center"/>

<TextBlock Text="Описание товара" Height="40" FontSize="20" Margin="5" Grid.Row="10" HorizontalAlignment="Center"/>

<TextBlock Text="Фото" Height="40" FontSize="20" Margin="177,4.6,177,0" Grid.Row="12" HorizontalAlignment="Center" Grid.RowSpan="2" VerticalAlignment="Top"/>

<TextBlock Text="Действующая скидка" FontSize="20" Margin="5" Grid.Row="11" HorizontalAlignment="Center"/>

<Button x:Name="LoadPhotoBtn" Content="Загрузить фото" VerticalAlignment="Center" HorizontalAlignment="Center" Style="{StaticResource ButtonRadius}" Grid.Row="12" Grid.Column="1" Width="120" Height="30" Click="LoadPhotoBtn\_Click"/>

<TextBox x:Name="T\_Artikyl\_ID" Width="350" Height="25" Grid.Column="1" Text="{Binding T\_Artikyl\_ID}"/>

<TextBox x:Name="SpisokCb" Grid.Row="1" Grid.Column="1" Text="{Binding T\_Spisok\_ID}" Width="350" Height="20"/>

<ComboBox x:Name="EdCb" Grid.Row="2" Grid.Column="1" SelectedItem="{Binding Ed\_Izm}" DisplayMemberPath="E\_Name" Width="350" Height="20"/>

<TextBox x:Name="T\_Stoimost" Width="350" Height="25" Grid.Column="1" Grid.Row="3" Text="{Binding T\_Stoimost}"/>

<TextBox x:Name="T\_Razmer\_Max\_Skidki" Width="350" Height="25" Grid.Column="1" Grid.Row="4" Text="{Binding T\_Razmer\_Max\_Skidki}"/>

<ComboBox x:Name="ProizCb" Grid.Row="5" Grid.Column="1" SelectedItem="{Binding Proizvod}" DisplayMemberPath="P\_Name" Width="350" Height="20"/>

<ComboBox x:Name="PostvCb" Grid.Row="6" Grid.Column="1" SelectedItem="{Binding Postavchik}" DisplayMemberPath="P\_Name" Width="350" Height="20"/>

<ComboBox x:Name="TypeCb" Grid.Row="7" Grid.Column="1" SelectedItem="{Binding Type\_Tovara}" DisplayMemberPath="T\_Name" Width="350" Height="20"/>

<TextBox x:Name="T\_Destv\_Skidka" Width="350" Height="25" Grid.Column="1" Grid.Row="8" Text="{Binding T\_Destv\_Skidka}"/>

<TextBox x:Name="T\_Kolich" Width="350" Height="25" Grid.Column="1" Grid.Row="9" Text="{Binding T\_Kolich}"/>

<TextBox x:Name="T\_Opis" Width="350" Height="25" Grid.Column="1" Grid.Row="10" Text="{Binding T\_Opis}"/>

<Button x:Name="back" Content="Назад" Grid.Row="13" Style="{StaticResource ButtonRadius}" Width="150" Height="50" VerticalAlignment="Center" HorizontalAlignment="Left" Click="back\_Click"/>

<Button x:Name="save" Content="Сохранить" Grid.Row="13" Style="{StaticResource ButtonRadius}" Width="150" Height="50" VerticalAlignment="Center" Grid.Column="1" HorizontalAlignment="Right" Click="save\_Click"/>

</Grid>

</Page>

AddAndEditPage.xaml.cs

using OOO\_S\_App.Base;

using System;

using System.Collections.Generic;

using System.IO;

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 Microsoft.Win32;

namespace OOO\_S\_App.Pages

{

/// <summary>

/// Логика взаимодействия для AddAndEditTovarPage.xaml

/// </summary>

public partial class AddAndEditTovarPage : Page

{

private Tovar \_currentZakazchik = new Tovar();

public string \_photoDirectory = $@"{Directory.GetCurrentDirectory()}\Images\";

private string \_photoPath;

private string \_photoName;

public AddAndEditTovarPage(Tovar selectedZakazchik)

{

InitializeComponent();

\_currentZakazchik = selectedZakazchik ?? new Tovar();

EdCb.ItemsSource = TradeEntities.GetContext().Ed\_Izm.ToList();

ProizCb.ItemsSource = TradeEntities.GetContext().Proizvods.ToList();

PostvCb.ItemsSource = TradeEntities.GetContext().Postavchiks.ToList();

TypeCb.ItemsSource = TradeEntities.GetContext().Type\_Tovara.ToList();

DataContext = \_currentZakazchik;

}

//кнопка назад отправляет пользователя на предыдущую страницу

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

{

NavigationService.GoBack();

}

//проверка на ошибки возвращает аргумент при наличии оных

private StringBuilder CheckFileds()

{

StringBuilder s = new StringBuilder();

return s;

}

//кнопка сохранить работает, если данные в необходимых полях введены корректно, тогда выводитсся уведомление об успешном внесении изменений, в ином случае выводится ошибка

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

{

StringBuilder \_error = CheckFileds();

if (\_error.Length > 0)

{

MessageBox.Show(\_error.ToString());

return;

}

try

{

if (\_photoPath != null)

{

\_currentZakazchik.T\_Photo = \_photoName;

File.Copy(\_photoPath, \_photoDirectory + \_photoName);

}

\_currentZakazchik.T\_Ed\_Izm\_ID = (EdCb.SelectedItem as Ed\_Izm).Ed\_Izm\_ID;

\_currentZakazchik.T\_Proizvod\_ID = (ProizCb.SelectedItem as Proizvod).Proizvod\_ID;

\_currentZakazchik.T\_Postavshik\_ID = (PostvCb.SelectedItem as Postavchik).Postavchik\_ID;

\_currentZakazchik.T\_Type\_Tovar\_ID = (TypeCb.SelectedItem as Type\_Tovara).Type\_Tovara\_ID;

if (\_currentZakazchik.T\_Artikyl\_ID != null)

{

TradeEntities.GetContext().Tovars.Add(\_currentZakazchik);

}

TradeEntities.GetContext().SaveChanges();

MessageBox.Show("Запись изменена");

NavigationService.Navigate(new Pages.AdminPage());

}

catch (Exception ex)

{

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

}

}

//при взаимодействии с кнопкой загрузить изображение открывается окно, где можно указать путь к изображению, соответствующему заданным критериям\

//в случае, если иображение не соответствует заданным параметрам - выводится ошибка

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

{

OpenFileDialog openFileDialog = new OpenFileDialog();

openFileDialog.Filter = "JPG Files (\*.jpg)|\*.jpg|PNG Files (\*.png)|\*.png";

if (openFileDialog.ShowDialog() == false)

{

return;

}

FileInfo fileInfo = new FileInfo(openFileDialog.FileName);

if (fileInfo.Length > 8 \* 1024 \* 1024 \* 6)

{

MessageBox.Show("Размер фото не должен превышать 6 мб");

return;

}

\_photoName = Guid.NewGuid().ToString();

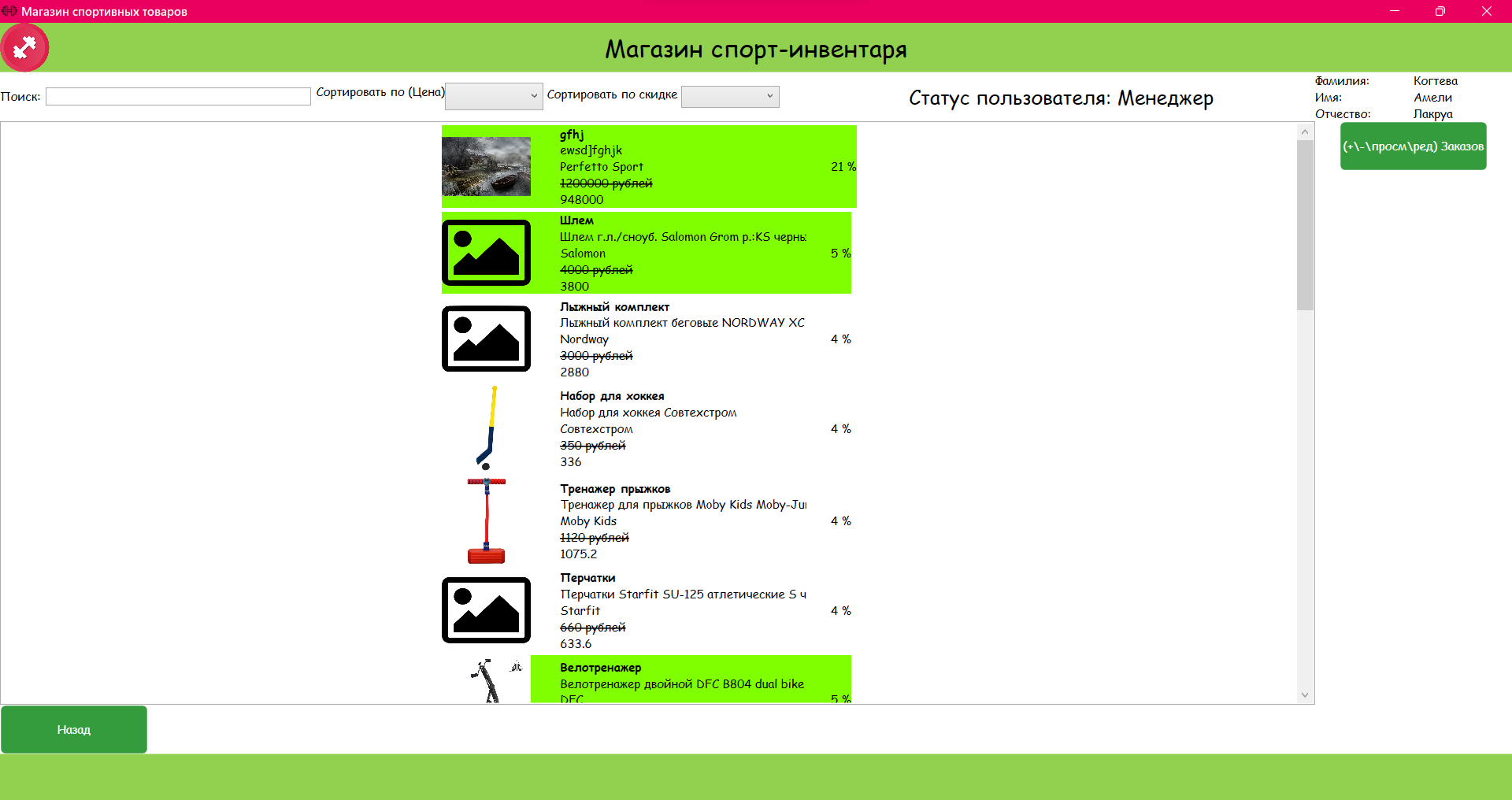
\_photoPath = fileInfo.FullName;

}

}

}

# **ManaferPage**



ManagerPage.xaml

<Page x:Class="OOO\_S\_App.Pages.ManagerPage"

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:OOO\_S\_App.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="ManagerPage" Style="{StaticResource base\_page}" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="100"/>

<ColumnDefinition Width="100"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<Grid>

<WrapPanel VerticalAlignment="Center" HorizontalAlignment="Left">

<StackPanel Orientation="Horizontal" VerticalAlignment="Center">

<TextBlock Text="Поиск:" />

<TextBox x:Name="TBP" Width="270" Margin="5,0,5,0" TextChanged="TBP\_TextChanged"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Сортировать по (Цена)"/>

<ComboBox Name="CBS" DisplayMemberPath="" Width="100" SelectionChanged="CBS\_SelectionChanged">

<ComboBoxItem Content="По возрастанию"/>

<ComboBoxItem Content="По убыванию"/>

</ComboBox>

</StackPanel>

<StackPanel Orientation="Horizontal" VerticalAlignment="Center">

<TextBlock Text=" Сортировать по скидке "/>

<ComboBox Name="CBBEB" SelectionChanged="CBBEB\_SelectionChanged" Width="100">

<ComboBoxItem Content="Все"></ComboBoxItem>

<ComboBoxItem Content="0-10%"></ComboBoxItem>

<ComboBoxItem Content="11-15%"></ComboBoxItem>

<ComboBoxItem Content="16+%"></ComboBoxItem>

</ComboBox>

</StackPanel>

<StackPanel Grid.Column="0" Grid.ColumnSpan="2" HorizontalAlignment="Right" VerticalAlignment="Center" Orientation="Vertical">

<TextBlock Text=" Статус пользователя: Менеджер" FontSize="20"/>

</StackPanel>

</WrapPanel>

</Grid>

<Grid Grid.Row="1">

<Grid.RowDefinitions>

</Grid.RowDefinitions>

<ListView x:Name="LVTovar" Grid.Row="1" ScrollViewer.HorizontalScrollBarVisibility="Disabled">

<ListView.ItemsPanel >

<ItemsPanelTemplate>

<WrapPanel Orientation="Vertical" HorizontalAlignment="Center"/>

</ItemsPanelTemplate>

</ListView.ItemsPanel>

<ListView.ItemTemplate>

<DataTemplate>

<Grid Background="{Binding GetColor}">

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<StackPanel Orientation="Horizontal" Height="auto" Width="auto">

<Image Width="90" HorizontalAlignment="Center" VerticalAlignment="Center" Source="{Binding Path=GetPhoto}"/>

<StackPanel Grid.Column="1" Margin="30 0 25 0" Width="250" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock FontWeight="Bold" Text="{Binding Path=T\_Spisok\_ID}"/>

<TextBlock Text="{Binding Path=T\_Opis}"/>

<TextBlock Text="{Binding Path=Proizvod.P\_Name}"/>

<TextBlock Text="{Binding Path=T\_Stoimost, StringFormat={}{0} рублей}" TextDecorations="{Binding Path=GetTextDecotation}"/>

<TextBlock Text="{Binding ProductPriceWithDiscount}"/>

</StackPanel>

<StackPanel Grid.Column="2" Margin="0 0 0 0" VerticalAlignment="Center" HorizontalAlignment="Right">

<TextBlock Text="{Binding Path=T\_Destv\_Skidka, StringFormat={}{0} %}"/>

</StackPanel>

</StackPanel>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

<StackPanel Grid.Column="1" Orientation="Vertical" HorizontalAlignment="Left" VerticalAlignment="Top">

<TextBlock Text="Фамилия:"/>

<TextBlock Text="Имя:"/>

<TextBlock Text="Отчество:"/>

</StackPanel>

<StackPanel Grid.Column="2" Orientation="Vertical" Width="100" HorizontalAlignment="Center" VerticalAlignment="Top">

<TextBlock x:Name="TbFamil" Text="{Binding U\_Familya}"/>

<TextBlock x:Name="TbImya" Text="{Binding U\_Imya}"/>

<TextBlock x:Name="TbOtchestvo" Text="{Binding U\_Otchestvo}"/>

</StackPanel>

<Button x:Name="back" Content="Назад" Grid.Row="2" Style="{StaticResource ButtonRadius}" Width="150" Height="50" VerticalAlignment="Center" HorizontalAlignment="Left" Click="back\_Click"/>

<Button x:Name="zakaz" Content="(+\-\просм\ред) Заказов" HorizontalAlignment="Center" VerticalAlignment="Top" Click="zakaz\_Click" Style="{StaticResource ButtonRadius}" Grid.ColumnSpan="2" Grid.Column="1" Width="150" Height="50" Grid.Row="1"/>

</Grid>

</Page>

ManagerPage.xaml.cs

using OOO\_S\_App.Base;

using OOO\_S\_App.Class;

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;

namespace OOO\_S\_App.Pages

{

/// <summary>

/// Логика взаимодействия для ManagerPage.xaml

/// </summary>

public partial class ManagerPage : Page

{

public ManagerPage()

{

InitializeComponent();

DataContext = TradeEntities.GetContext().Users.ToList();

List<User> users = TradeEntities.GetContext().Users.ToList();

TbFamil.Text = Auth.U\_Familya;

TbImya.Text = Auth.U\_Imya;

TbOtchestvo.Text = Auth.U\_Otchestvo;

}

//кнопка назад возвращает пользователя на предыдущую страницу

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

{

NavigationService.Navigate(new Pages.AvtorizPage());

}

//оператор случая, необходимый для корректного отображения данных

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

TradeEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

LVTovar.ItemsSource = TradeEntities.GetContext().Tovars.OrderBy(p => p.T\_Artikyl\_ID).ToList();

}

}

//кнопка, по нажатию на которую пользователь будет перенаправлен на страницу заказа

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

{

NavigationService.Navigate(new Pages.ZakazPage());

}

//поиск по наименованию товара

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

{

UpdateData();

}

//сортировка по стоимости товара

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

{

UpdateData();

}

//логика для осуществления поиска и сортировки по стоимости

private void UpdateData()

{

var currentProduct = TradeEntities.GetContext().Tovars.OrderBy(p => p.T\_Spisok\_ID).ToList();

currentProduct = currentProduct.Where(p => p.T\_Spisok\_ID.ToLower().Contains(TBP.Text.ToLower())).ToList();

if (CBS.SelectedIndex >= 0)

{

if (CBS.SelectedIndex == 0)

{

currentProduct = currentProduct.OrderBy(p => p.T\_Stoimost).ToList();

}

if (CBS.SelectedIndex == 1)

{

currentProduct = currentProduct.OrderByDescending(p => p.T\_Stoimost).ToList();

}

}

LVTovar.ItemsSource = currentProduct;

}

//фильтрация по скидке товара

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

{

UpdateData1();

}

//логика для осуществления фильтрации, сортировки и поиска

private void UpdateData1()

{

var currentProduct = TradeEntities.GetContext().Tovars.OrderBy(p => p.T\_Spisok\_ID).ToList();

currentProduct = currentProduct.Where(p => p.T\_Spisok\_ID.ToLower().Contains(TBP.Text.ToLower())).ToList();

if (CBS.SelectedIndex >= 0)

{

if (CBS.SelectedIndex == 0)

{

currentProduct = currentProduct.OrderBy(p => p.T\_Stoimost).ToList();

}

if (CBS.SelectedIndex == 1)

{

currentProduct = currentProduct.OrderByDescending(p => p.T\_Stoimost).ToList();

}

}

if (CBBEB.SelectedIndex > 0)

{

int a = 0;

int b = 0;

switch (CBBEB.SelectedIndex)

{

case 1:

a = 0;

b = 10;

break;

case 2:

a = 11;

b = 15;

break;

case 3:

a = 16;

b = 100;

break;

}

currentProduct = currentProduct.Where(p => p.T\_Destv\_Skidka > a && p.T\_Destv\_Skidka <= b).ToList();

}

LVTovar.ItemsSource = currentProduct;

}

}

}