Оглавление

[Код окна AuthWindow.xaml: 2](#_Toc129685834)

[Код окна AuthWindow.xaml.cs: 3](#_Toc129685835)

[Код она MainWindow.xaml: 4](#_Toc129685836)

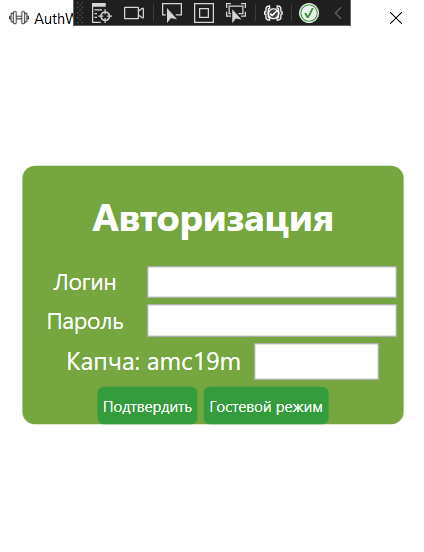
[Код окна MainWindow.xaml.cs: 4](#_Toc129685837)

[Код окна ShopingCartPage.xaml: 5](#_Toc129685838)

[Код окна ShopingCartPage.xaml.cs: 6](#_Toc129685839)

[Код окна AssortimentsPage.xaml: 9](#_Toc129685840)

[Код окна AssortimentsPage.xaml.cs: 11](#_Toc129685841)



# Код окна AuthWindow.xaml:

<Grid>

<Border HorizontalAlignment="Center" VerticalAlignment="Center" CornerRadius="10" Background="#75A640">

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock FontSize="30" HorizontalAlignment='Center' Margin='0 20 0 20' TextAlignment="Center" FontWeight="Bold" Foreground="White" Grid.Row="0" >Авторизация</TextBlock>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" Grid.Row='1'>

<TextBlock Foreground="White" Width="100" FontSize="18" TextAlignment="Center" >Логин</TextBlock>

<TextBox x:Name="Login" FontSize="18" Width="200" Margin="0 0 5 0"></TextBox>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" Grid.Row='2' Margin="0 5 0 0">

<TextBlock Foreground="White" Width="100" FontSize="18" TextAlignment="Center" >Пароль</TextBlock>

<TextBox x:Name="Password" FontSize="18" Width="200" Margin="0 0 5 0"></TextBox>

</StackPanel>

<StackPanel Grid.Row="3" Margin=" 0 5 20 5" Height="30" Orientation="Horizontal" HorizontalAlignment="Right">

<TextBlock x:Name="Code" Foreground="White" Margin="0 0 10 0" FontSize="20" TextAlignment="Left">Код:</TextBlock>

<TextBox x:Name="CodeInput" FontSize="18" Width="100"></TextBox>

</StackPanel>

<StackPanel Grid.Row="4" Orientation="Horizontal" Height="30" HorizontalAlignment="Center">

<Button x:Name="Ok" Click="Ok\_Click" Background="#349C3D" Margin="0 0 5 0" Foreground="White" Width="80">Подтвердить</Button>

<Button x:Name="Gost" Click="Gost\_Click" Background="#349C3D" Foreground="White" Width="100"> Гостевой режим </Button>

</StackPanel>

</StackPanel>

</Border>

</Grid>

# Код окна AuthWindow.xaml.cs:

public partial class AuthWindow : Window

{

private string \_code = string.Empty;

public AuthWindow()

{

InitializeComponent();

GenerateCode();

}

private string GenerateCode()

{

Random random= new Random();

List <string> chars = new List<string>() {"1","2","3","5","4","6","7","8","9", "a","b","c","d","f","h","g","p","m","j"};

string result = string.Empty;

for(int i = 0; i < 6; i++)

{

result += chars[random.Next(chars.Count)];

}

Code.Text = "Капча: " + result;

return \_code = result;

}

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

{

cfg.AuthUser = null;

cfg.IsAuth = false;

MainWindow window = new MainWindow();

window.Show();

this.Close();

}

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

{

List <User> users = App.db.Users.ToList();

User u = users.FirstOrDefault(el => el.UserLogin == Login.Text && el.UserPassword == Password.Text);

if (CodeInput.Text == \_code)

{

if (u != null)

{

MessageBox.Show($"Добро пожаловать, {u.UserName}.\nВаша роль: {u.Role.RoleName}");

cfg.AuthUser = u;

cfg.IsAuth = true;

MainWindow window = new MainWindow();

window.Show();

this.Close();

}

else

{

MessageBox.Show("Такого Пользователя не существует!");

}

}

else

{

MessageBox.Show("Вы не правильно ввели капчу, блокировка приложения на 10 сек.");

Thread.Sleep(10000);

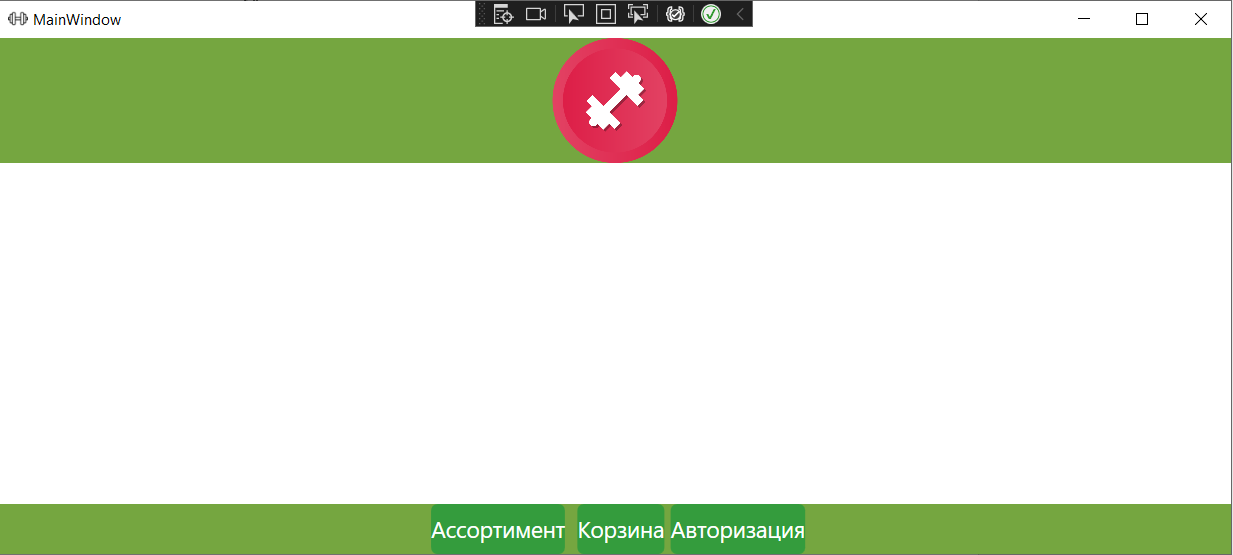
GenerateCode();

}

}

}

# Код она MainWindow.xaml:



<Grid.RowDefinitions>

<RowDefinition Height="100"/>

<RowDefinition/>

<RowDefinition Height="40" />

</Grid.RowDefinitions>

<Grid Grid.Row="2" Background="#75A640"/>

<StackPanel Background='#75A640'>

<Image Source="Resources/logo.png" Height="100" Margin="0 0 0 10"></Image>

</StackPanel>

<StackPanel Grid.Row="2" Orientation="Horizontal" HorizontalAlignment="Center">

<Button x:Name="Assortiment" Click="Assortiment\_Click" FontSize="18" Foreground="White" Background="#349C3D" Margin="5 0 5 0">Ассортимент</Button>

<Button x:Name="OpenCards" FontSize="18" Click="OpenCards\_Click" Foreground="White" Background="#349C3D" Margin="5 0 5 0">Корзина</Button>

<Button x:Name="Auth" FontSize="18" Click="Auth\_Click" Foreground="White" Background="#349C3D">Авторизация</Button>

</StackPanel>

<Frame x:Name="MainFrame" Grid.Row="1"></Frame>

</Grid>

# Код окна MainWindow.xaml.cs:

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

cfg.MainFrame = MainFrame;

if (!cfg.IsAuth)

{

Auth.Content = "Войти";

}

}

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

{

cfg.MainFrame.Navigate(new AssortimentsPage());

}

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

{

cfg.MainFrame.Navigate(new ShopingCartPage());

}

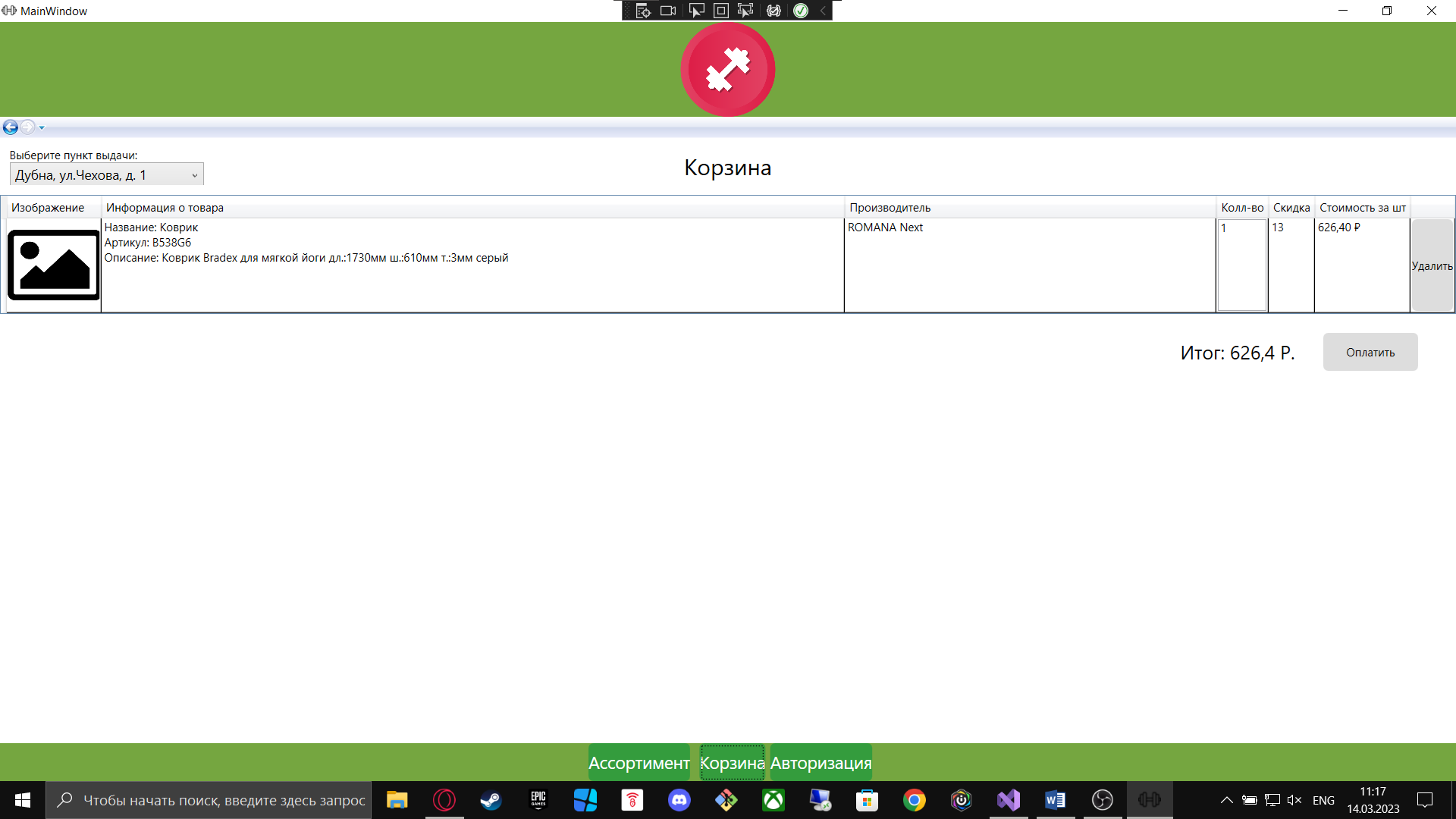
private void Auth\_Click(object sender, RoutedEventArgs e)

{

}

}

# Код окна ShopingCartPage.xaml:



<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="60"></RowDefinition>

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

<RowDefinition Height="80"></RowDefinition>

</Grid.RowDefinitions>

<StackPanel Margin="10" >

<TextBlock Text="Выберите пункт выдачи: " />

<ComboBox Width="205" SelectionChanged="ComboPickup\_SelectionChanged" HorizontalAlignment="Left" Height="30" FontSize="14" x:Name="ComboPickup" DisplayMemberPath="FullAddress"/>

</StackPanel>

<TextBlock Text="Корзина" VerticalAlignment="Center" Grid.Row="0" HorizontalAlignment="Center" FontSize="24px"></TextBlock>

<DataGrid x:Name="DataGridSC" Grid.Row="1" AutoGenerateColumns="False" RowHeight="100" SelectedCellsChanged ="DataGridSC\_SelectedCellsChanged" LoadingRow="DataGridSC\_LoadingRow" IsReadOnly="True" SelectionMode="Single">

<DataGrid.Columns >

<DataGridTemplateColumn Width="100" Header="Изображение" >

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Image Source="{Binding Path=GetImage}" Width="100"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTextColumn Binding="{Binding ProductFullInfo}" Header="Информация о товара" Width="2\*"/>

<DataGridTextColumn Binding="{Binding Manufacturer.Name}" Header="Производитель" Width="1\*"/>

<DataGridTemplateColumn Width="auto" Header="Колл-во">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBox TextChanged="count\_TextChanged" x:Name="countText" Text="{Binding ProductQuantityInStock}"></TextBox>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTextColumn Binding="{Binding ProductDiscountAmount}" Header="Скидка" Width="auto"/>

<DataGridTextColumn Binding="{Binding ProductCostWithDiscount, StringFormat='c', ConverterCulture='ru-RU'}" Header="Стоимость за шт" Width="auto"/>

<DataGridTemplateColumn Width="auto" Header="">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnDelete" Content="Удалить" Click="BtnDelete\_Click" Margin="0"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<StackPanel Grid.Row="2" Orientation="Horizontal" HorizontalAlignment="Right" Margin="0 0 30 0">

<StackPanel Orientation="Horizontal" Margin="10">

<TextBlock VerticalAlignment="Center" FontSize="20" Text="Итог: "></TextBlock>

<TextBlock VerticalAlignment="Center" FontSize="20" x:Name="TotalSum" Text="0 р"></TextBlock>

</StackPanel>

<Button Content="Оплатить" Click="Order\_Click" Width="100" Height="40" Margin="20 10 10 10"></Button>

</StackPanel>

</Grid>

# Код окна ShopingCartPage.xaml.cs:

public partial class ShopingCartPage : Page

{

static private double \_sum = 0;

List<Product> shopingCart = new List<Product>();

int PickupPointId = 0;

public ShopingCartPage()

{

InitializeComponent();

ComboPickup.ItemsSource = App.db.OrderPickups.ToList();

ComboPickup.SelectedIndex= 0;

LoadDataGrid();

}

private void UpdatePage()

{

cfg.MainFrame.Navigate(new ShopingCartPage());

}

private void UpdateTotalSumText(double sum)

{ TotalSum.Text = Convert.ToString(sum) + " Р."; }

public void LoadDataGrid()

{

List<OrderProduct> orderProducts = App.db.OrderProducts.Where(el => el.Order.UserID == cfg.AuthUser.UserID && el.Order.OrderStatusID == 1).ToList();

List<Product> allProduct = App.db.Products.ToList();

\_sum = 0;

foreach (OrderProduct product in orderProducts)

{

Product productadd = allProduct.Where(el=>el.ProductArticleNumber == product.ProductArticleNumber).FirstOrDefault();

if(productadd != null)

{

productadd.ProductQuantityInStock = product.Count ?? 0;

\_sum += productadd.ProductCostWithDiscount \* Convert.ToDouble(productadd.ProductQuantityInStock);

shopingCart.Add(productadd);

}

}

UpdateTotalSumText(\_sum);

DataGridSC.ItemsSource = shopingCart;

}

public void DeleteProduct(Product product)

{

shopingCart.Remove(product);

List<OrderProduct> orderProducts = App.db.OrderProducts.ToList();

OrderProduct findProduct = orderProducts.FirstOrDefault(el => el.ProductArticleNumber == product.ProductArticleNumber && el.Order.UserID == cfg.AuthUser.UserID && el.Order.OrderStatusID == 1);

if (findProduct != null)

{

App.db.OrderProducts.Remove(findProduct);

MessageBox.Show("Товар удален из корзины!");

}

else

{

MessageBox.Show("Товар уже удален!");

}

App.db.SaveChanges();

}

public void UpdateProductQuantity(Product product, int Value)

{

List<OrderProduct> orderProducts = App.db.OrderProducts.ToList();

OrderProduct findProduct = orderProducts.FirstOrDefault(el => el.ProductArticleNumber == product.ProductArticleNumber && el.Order.UserID == cfg.AuthUser.UserID && el.Order.OrderStatusID == 1);

if (findProduct != null)

{

findProduct.Count = Value;

}

else

{

MessageBox.Show("Товар уже удален!");

}

App.db.SaveChanges();

}

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

{ }

private void DataGridSC\_SelectedCellsChanged(object sender, SelectedCellsChangedEventArgs e)

{ }

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

{

var orders = App.db.Orders;

Order order = orders.FirstOrDefault(el => el.UserID == cfg.AuthUser.UserID && el.OrderStatusID == 1);

if (order != null)

{

if (PickupPointId == 0)

{

MessageBox.Show("Выбери пункт выдачи.");

return;

}

order.OrderCreate = DateTime.Now;

if (shopingCart.Count > 3)

{

order.OrderDeliveryDate = DateTime.Now.AddDays(3);

}

else

{

order.OrderDeliveryDate = DateTime.Now.AddDays(6);

}

order.OrderPickupPoint = PickupPointId;

order.OrderStatusID = 2;

App.db.SaveChanges();

UpdatePage();

}

}

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

{

Product selected = (sender as Button).DataContext as Product;

DeleteProduct(selected);

UpdatePage();

}

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

{

var ob = (sender as TextBox);

Product selected = (sender as TextBox).DataContext as Product;

if (ob.IsLoaded)

{

int selectedIndex = DataGridSC.SelectedIndex;

var text = ob.Text;

int number;

bool isNumber = int.TryParse(text, out number);

if (isNumber)

{

if(number == 0)

{

DeleteProduct(selected);

}

if(number > 0)

{

UpdateProductQuantity(selected, number);

}

if(number < 0)

{

MessageBox.Show("Колл-во не может быть меньше нуля");

}

UpdatePage();

}

else

{ MessageBox.Show("Не должны быть букв и знаков!"); }

}

}

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

{

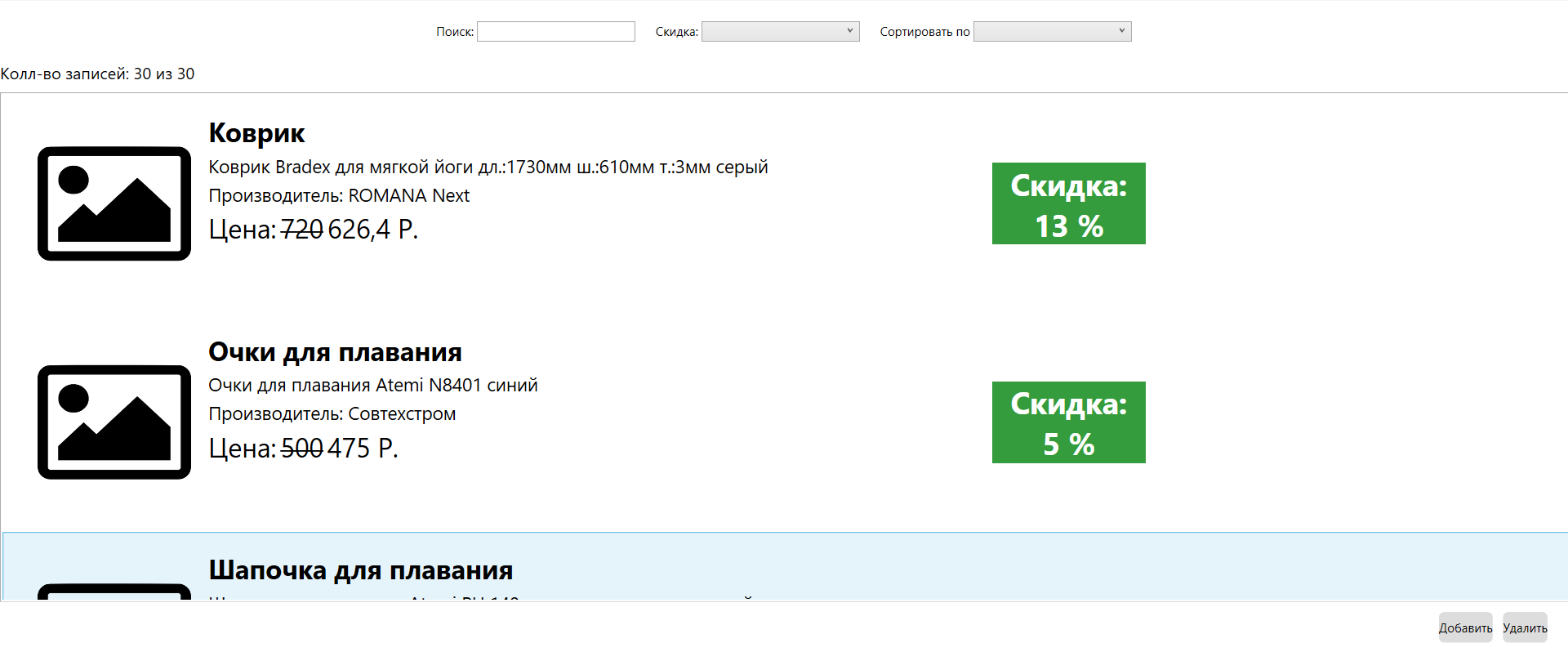
OrderPickup pickup = ComboPickup.SelectedValue as OrderPickup;

PickupPointId = pickup.OrderPickupPointID;

}

}

# Код окна AssortimentsPage.xaml:



<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="60"></RowDefinition>

<RowDefinition Height="30"></RowDefinition>

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

<RowDefinition Height="50"></RowDefinition>

</Grid.RowDefinitions>

<WrapPanel Grid.Row="0" Orientation="Horizontal" Margin="10" HorizontalAlignment="Center">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Left" Margin="10">

<TextBlock Text="Поиск: " VerticalAlignment="Center"/>

<TextBox Width="155" x:Name="TBoxSearch" TextChanged="TBoxSearch\_TextChanged"/>

</StackPanel>

<StackPanel Orientation="Horizontal" Margin="10">

<TextBlock Text="Скидка: " VerticalAlignment="Center"/>

<ComboBox Width="155" x:Name="ComboDiscount" SelectionChanged="ComboDiscount\_SelectionChanged">

<ComboBoxItem Content="Все диапазоны"/>

<ComboBoxItem Content="От 0 до 9.99%"/>

<ComboBoxItem Content="От 10 до 14.99%"/>

<ComboBoxItem Content="От 15 %"/>

</ComboBox>

</StackPanel>

<StackPanel Orientation="Horizontal" Margin="10">

<TextBlock Text="Сортировать по " VerticalAlignment="Center"/>

<ComboBox Width="155" x:Name="ComboSort" SelectionChanged="ComboSort\_SelectionChanged">

<ComboBoxItem Content="Цена (По возрастанию)"/>

<ComboBoxItem Content="Цена (По убыванию)"/>

</ComboBox>

</StackPanel>

</WrapPanel>

<TextBlock x:Name="CountProducts" FontSize="16" Grid.Row="1">10 bp 100</TextBlock>

<ListBox x:Name='AssortimentList' Width="1600" Grid.Row="2" HorizontalAlignment="Left" VerticalAlignment="Center" Background="#fff" >

<ListBox.ContextMenu>

<ContextMenu>

<MenuItem x:Name="OpenCardAdd" Header="Добавить в корзину" Click="OpenCardAdd\_Click"/>

</ContextMenu>

</ListBox.ContextMenu>

<ListBox.ItemTemplate>

<DataTemplate>

<Grid Margin="15">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="180"></ColumnDefinition>

<ColumnDefinition Width="770"></ColumnDefinition>

<ColumnDefinition Width="150"></ColumnDefinition>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="180"></RowDefinition>

</Grid.RowDefinitions>

<Image Source="{Binding GetImage}" Width="150" Grid.Column="0"></Image>

<StackPanel Grid.Column="1" >

<TextBlock TextAlignment="Left" FontSize="26" Text="{Binding ProductName}" Margin="2" FontWeight="Bold"/>

<TextBlock TextAlignment="Left" FontSize="18" Text="{Binding ProductDescription}" Margin="2" />

<TextBlock TextAlignment="Left" FontSize="18" Text="{Binding GetManufacturer}" Margin="2" />

<StackPanel Orientation="Horizontal">

<TextBlock FontSize="26" Margin="2">Цена: </TextBlock>

<TextBlock TextAlignment="Left" FontSize="26" Text="{Binding ProductCost}" Margin="2">

<TextBlock.TextDecorations>

<TextDecoration Location="Strikethrough">

</TextDecoration>

</TextBlock.TextDecorations>

</TextBlock>

<TextBlock TextAlignment="Left" FontSize="26" Text="{Binding GetPrice}" Margin="2"/>

</StackPanel>

</StackPanel>

<StackPanel Grid.Column="3" Width="150" Background="#349C3D" VerticalAlignment="Center" >

<TextBlock TextAlignment="Center" Foreground="White" FontWeight="Bold" FontSize="30">Скидка:</TextBlock>

<TextBlock TextAlignment="Center" Foreground="White" FontWeight="Bold" FontSize="30" Text="{Binding GetDiscount}"></TextBlock>

</StackPanel>

</Grid>

</DataTemplate>

</ListBox.ItemTemplate>

</ListBox>

<StackPanel Grid.Row="3" Orientation="Horizontal" Margin="10" HorizontalAlignment="Right">

<Button Height="30 " HorizontalAlignment="Left">Добавить</Button>

<Button Height="30" Margin="10 0">Удалить</Button>

</StackPanel>

</Grid>

# Код окна AssortimentsPage.xaml.cs:

public partial class AssortimentsPage : Page

{

private int \_countProduct = 0;

private int \_countProductWithFilter = 0;

public AssortimentsPage()

{

InitializeComponent();

var products = App.db.Products.ToList();

\_countProduct = products.Count;

\_countProductWithFilter = \_countProduct;

UpdateCountProdutsText();

AssortimentList.ItemsSource = products;

}

private void UpdateData()

{

var currentGoods = App.db.Products.ToList();

currentGoods = currentGoods.Where(p => p.ProductName.ToLower().Contains(TBoxSearch.Text.ToLower())).ToList();

if (ComboDiscount.SelectedIndex > 0)

{

if (ComboDiscount.SelectedIndex == 1)

currentGoods = currentGoods.Where(p => p.ProductDiscountAmount >= 0 && p.ProductDiscountAmount < 10).ToList();

if (ComboDiscount.SelectedIndex == 2)

currentGoods = currentGoods.Where(p => p.ProductDiscountAmount >= 10 && p.ProductDiscountAmount < 15).ToList();

if (ComboDiscount.SelectedIndex == 3)

currentGoods = currentGoods.Where(p => p.ProductDiscountAmount >= 15).ToList();

}

if (ComboSort.SelectedIndex >= 0)

{

if (ComboSort.SelectedIndex == 0)

currentGoods = currentGoods.OrderBy(p => p.ProductCostWithDiscount).ToList();

if (ComboSort.SelectedIndex == 1)

currentGoods = currentGoods.OrderByDescending(p => p.ProductCostWithDiscount).ToList();

}

\_countProductWithFilter = currentGoods.Count;

UpdateCountProdutsText();

AssortimentList.ItemsSource = currentGoods;

}

private void UpdateCountProdutsText ()

{

CountProducts.Text = $"Колл-во записей: {\_countProductWithFilter} из {\_countProduct}";

}

private Order OrderCheck()

{

var orders = App.db.Orders;

Order order = orders.FirstOrDefault(el => el.UserID == cfg.AuthUser.UserID && el.OrderStatusID == 1);

return order;

}

private int OrderAdd()

{

Random random = new Random();

Order newOrder = new Order { OrderStatusID = 1, OrderCreate = DateTime.Now, OrderDeliveryDate = DateTime.Now.AddDays(3), OrderPickupPoint = 5, OrderPickupCode = Convert.ToString(random.Next(0, 10000)), UserID = cfg.AuthUser.UserID };

App.db.Orders.Add(newOrder);

App.db.SaveChanges();

Order order = OrderCheck();

if(order == null)

{

return 0;

}

return order.OrderID;

}

private void ProductFromOrderAdd(int orderId, Product newproduct)

{

OrderProduct orderProduct = App.db.OrderProducts.FirstOrDefault(el => el.OrderID == orderId && el.ProductArticleNumber == newproduct.ProductArticleNumber);

if(orderProduct == null)

{

App.db.OrderProducts.Add(new OrderProduct { OrderID = orderId, ProductArticleNumber = newproduct.ProductArticleNumber, Count = 1 });

}

else

{

orderProduct.Count++;

}

App.db.SaveChanges();

}

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

{

if (cfg.IsAuth)

{

Order order = OrderCheck();

Product selectedProduct = AssortimentList.SelectedItems[0] as Product;

if (order == null)

{

int orderid = OrderAdd();

ProductFromOrderAdd(orderid, selectedProduct);

}

else

{

ProductFromOrderAdd(order.OrderID, selectedProduct);

}

}

else

{

MessageBox.Show("Вы не авторизированы");

}

}

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

{

UpdateData();

}

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

{

UpdateData();

}

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

{

UpdateData();

}

}

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

{

if (MessageBox.Show("удалить?", "уведомление", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes) ;

{

var CurrentUser = AssortimentList.SelectedItem as Product;

App.db.Products.Remove(CurrentUser);

App.db.SaveChanges();

AssortimentList.ItemsSource = App.db.Products.ToList();

MessageBox.Show("Успешно", "Удалено");

}

}

}

}