# Окно добавления Товара

## C#

using JewerlyStore.Classes;

using JewerlyStore.DB;

using Microsoft.Win32;

using System;

using System.IO;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

namespace JewerlyStore.View.Pages.FunctionsWithData

{

/// <summary>

/// Логика взаимодействия для AddPage.xaml

/// </summary>

public partial class AddPage : Page

{

public AddPage()

{

InitializeComponent();

//Инициализация данных в комбобокс

categoryCmb.ItemsSource = ConnectClass.db.Category.Select(item => item.Title).ToList();

}

//Кнопка назад

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

{

NavigationService.GoBack();

}

//Кнопка добавления

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

{

Jewelry newJewelry = new Jewelry();

Category newCategory = new Category();

Parameters newParameteres = new Parameters();

try

{

//Добавление изобржения

MemoryStream stream = new MemoryStream();

JpegBitmapEncoder encoder = new JpegBitmapEncoder();

encoder.Frames.Add(BitmapFrame.Create((BitmapImage)imgLoad.Source));

encoder.Save(stream);

newJewelry.JewImg = stream.ToArray();

//-------------------------------------------------------------------

newJewelry.JewName = jewNameTxb.Text;

newJewelry.Material = materialTxb.Text;

newJewelry.Pice = Convert.ToInt64(priceTxb.Text);

newJewelry.ParametersID = newParameteres.ID;

newJewelry.Count = Convert.ToInt32(txbCount.Text);

var jewCategory = ConnectClass.db.Category.FirstOrDefault(item => item.Title == categoryCmb.Text);

newJewelry.CategoryID = jewCategory.ID;

newParameteres.Height = heightTxb.Text;

newParameteres.Width = widthTxb.Text;

newParameteres.Weight = widthTxb.Text;

ConnectClass.db.Parameters.Add(newParameteres);

ConnectClass.db.Jewelry.Add(newJewelry);

ConnectClass.db.SaveChanges();

MessageBox.Show("Данные сохранены.", "Операция прошла успешно!", MessageBoxButton.OK, MessageBoxImage.Information);

NavigationService.GoBack();

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка: {ex.Message}", "Ошибка при сохранении!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

//Кнопка открытия изображения

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

{

try

{

OpenFileDialog file = new OpenFileDialog();

file.Filter = "Image (\*.png; \*.jpeg; \*.jpg;) | \*.png; \*.jpeg; \*.jpg;";

if (file.ShowDialog() == true)

{

BitmapImage imgBitmap = new BitmapImage(new Uri(file.FileName));

imgLoad.Source = imgBitmap;

}

}

catch (Exception)

{

MessageBox.Show("Выберите пожалуйста картинку.", "Произошла ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void priceTxb\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

e.Handled = "0123456789".IndexOf(e.Text) < 0;

}

private void heightTxb\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

e.Handled = "0123456789".IndexOf(e.Text) < 0;

}

private void widthTxb\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

e.Handled = "0123456789".IndexOf(e.Text) < 0;

}

private void weightTxb\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

e.Handled = "0123456789".IndexOf(e.Text) < 0;

}

private void txbCount\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

e.Handled = "0123456789".IndexOf(e.Text) < 0;

}

}

}

## XAML

<Page x:Class="JewerlyStore.View.Pages.FunctionsWithData.AddPage"

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:JewerlyStore.View.Pages.FunctionsWithData"

mc:Ignorable="d"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

d:DesignHeight="600" d:DesignWidth="1000"

Title="Добавление">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="0.2\*"/>

</Grid.RowDefinitions>

<Grid Grid.Row="0">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid Grid.Column="0">

<StackPanel VerticalAlignment="Center"

HorizontalAlignment="Center"

Orientation="Vertical">

<Image x:Name="imgLoad"

Width="200"

Height="200"

Margin="0,0,0,20"/>

<Button x:Name="openBtn"

materialDesign:ButtonAssist.CornerRadius="5"

Content="Выбрать"

Background="Black"

Foreground="White"

Width="200"

Height="35" Click="openBtn\_Click"/>

</StackPanel>

</Grid>

<Grid Grid.Column="1">

<StackPanel VerticalAlignment="Center"

HorizontalAlignment="Center"

Orientation="Vertical">

<StackPanel Orientation="Horizontal"

Margin="0,0,0,20">

<TextBox x:Name="jewNameTxb"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

FontSize="15"

materialDesign:HintAssist.Hint="Изделие"

Width="200"/>

</StackPanel>

<StackPanel Orientation="Horizontal"

Margin="0,0,0,20">

<ComboBox x:Name="categoryCmb"

Style="{StaticResource MaterialDesignFloatingHintComboBox}"

FontSize="15"

materialDesign:HintAssist.Hint="Категория"

Width="200"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBox x:Name="materialTxb"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

materialDesign:HintAssist.Hint="Материал"

FontSize="15"

Width="200"/>

</StackPanel>

<StackPanel Orientation="Horizontal"

Margin="0,0,0,20">

<TextBox x:Name="priceTxb"

Margin="0,10,0,0"

PreviewTextInput="priceTxb\_PreviewTextInput"

materialDesign:HintAssist.Hint="Цена"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

FontSize="15"

Width="200"/>

</StackPanel>

<StackPanel Orientation="Horizontal"

Margin="0,0,0,20">

<TextBox x:Name="heightTxb"

PreviewTextInput="heightTxb\_PreviewTextInput"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

materialDesign:HintAssist.Hint="Высота"

FontSize="15"

Width="200"/>

</StackPanel>

<StackPanel Orientation="Horizontal"

Margin="0,0,0,20">

<TextBox x:Name="widthTxb"

PreviewTextInput="widthTxb\_PreviewTextInput"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

materialDesign:HintAssist.Hint="Ширина"

FontSize="15"

Width="200"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBox x:Name="weightTxb"

PreviewTextInput="weightTxb\_PreviewTextInput"

materialDesign:HintAssist.Hint="Вес"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

FontSize="15"

Width="200"/>

</StackPanel>

<StackPanel>

<TextBox Name="txbCount"

PreviewTextInput="txbCount\_PreviewTextInput"

materialDesign:HintAssist.Hint="Количество товара"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

FontSize="15"

Margin="0,10,0,0"

Width="200"/>

</StackPanel>

</StackPanel>

</Grid>

</Grid>

<Grid Grid.Row="1">

<StackPanel VerticalAlignment="Center"

HorizontalAlignment="Center"

Orientation="Horizontal">

<Button x:Name="backBtn"

materialDesign:ButtonAssist.CornerRadius="5"

Content="Назад"

Background="Black"

Foreground="White"

Width="200"

Height="35"

Margin="0,0,20,0" Click="backBtn\_Click"/>

<Button x:Name="addBtn"

Content="Добавить"

Background="Black"

Foreground="White"

materialDesign:ButtonAssist.CornerRadius="5"

Width="200"

Height="35" Click="addBtn\_Click"/>

</StackPanel>

</Grid>

</Grid>

</Page>

# Окно Добавления клиентов

## C#

using JewerlyStore.Classes;

using JewerlyStore.DB;

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 JewerlyStore.View.Pages.FunctionsWithData

{

/// <summary>

/// Логика взаимодействия для CheckoutPage.xaml

/// </summary>

public partial class CheckoutPage : Page

{

public CheckoutPage()

{

InitializeComponent();

}

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

{

NavigationService.GoBack();

}

private void txbPhone\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

e.Handled = "0123456789+".IndexOf(e.Text) < 0;

}

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

{

try

{

Client client = new Client();

client.FirstName = txbFirstName.Text;

client.LastName = txbLastName.Text;

client.SecondName = txbSecondName.Text;

client.Phone = txbPhone.Text;

ConnectClass.db.Client.Add(client);

ConnectClass.db.SaveChanges();

MessageBox.Show("Клиент добавлен.", "Операция прошла успешно!", MessageBoxButton.OK, MessageBoxImage.Information);

NavigationService.Navigate(new PaymentPage());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

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

{

NavigationService.Navigate(new PaymentPage());

}

}

}

## XAML

<Page x:Class="JewerlyStore.View.Pages.FunctionsWithData.CheckoutPage"

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:JewerlyStore.View.Pages.FunctionsWithData"

mc:Ignorable="d"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

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

Title="Оформление заказа">

<Grid>

<StackPanel HorizontalAlignment="Center"

VerticalAlignment="Center"

Width="250">

<StackPanel VerticalAlignment="Top">

<TextBlock Text="Заполните данные покупателя: "

Margin="0,0,0,20"/>

</StackPanel>

<TextBox Name="txbLastName"

materialDesign:HintAssist.Hint="Фамилия"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"/>

<TextBox Name="txbFirstName"

materialDesign:HintAssist.Hint="Имя"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"/>

<TextBox Name="txbSecondName"

materialDesign:HintAssist.Hint="Отчество (необязательно)"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"/>

<TextBox Name="txbPhone"

PreviewTextInput="txbPhone\_PreviewTextInput"

materialDesign:HintAssist.Hint="Телефон"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"/>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Margin="0,10,0,0">

<Button Content="Сохранить"

Name="btnSave"

Width="120"

Height="25"

FontSize="11"

materialDesign:ButtonAssist.CornerRadius="5"

Foreground="White"

Background="Black" Click="btnSave\_Click"/>

<Button Content="Назад"

Margin="10,0,0,0"

materialDesign:ButtonAssist.CornerRadius="5"

Name="btnCancel"

Width="120"

Height="25"

FontSize="11"

Foreground="White"

Background="Black" Click="btnCancel\_Click"/>

</StackPanel>

<Button Content="Клиент существует"

Foreground="White"

Background="Black"

FontSize="11"

Margin="0, 10, 0, 0"

Height="25"

Name="btnNext"

materialDesign:ButtonAssist.CornerRadius="5"

Click="btnNext\_Click"/>

</StackPanel>

</Grid>

</Page>

# Окно редактирования Товара

## C#

using JewerlyStore.Classes;

using JewerlyStore.DB;

using Microsoft.Win32;

using System;

using System.IO;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Input;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

namespace JewerlyStore.View.Pages.FunctionsWithData

{

/// <summary>

/// Логика взаимодействия для EditPage.xaml

/// </summary>

public partial class EditPage : Page

{

private Jewelry selectedItem;

public EditPage(Jewelry selectedItem)

{

InitializeComponent();

this.selectedItem = selectedItem;

categoryCmb.ItemsSource = ConnectClass.db.Category.Select(item => item.Title).ToList();

//Передача изображения на страницу

if (selectedItem.JewImg != null)

{

BitmapImage bitmap = new BitmapImage();

using (MemoryStream stream = new MemoryStream(selectedItem.JewImg))

{

bitmap.BeginInit();

bitmap.CacheOption = BitmapCacheOption.OnLoad;

bitmap.StreamSource = stream;

bitmap.EndInit();

}

imgLoad.Source = bitmap;

}

jewNameTxb.Text = selectedItem.JewName;

categoryCmb.SelectedItem = selectedItem.Category.Title;

materialTxb.Text = selectedItem.Material;

priceTxb.Text = Convert.ToString(selectedItem.Pice);

heightTxb.Text = selectedItem.Parameters.Height;

widthTxb.Text = selectedItem.Parameters.Width;

weightTxb.Text = selectedItem.Parameters.Weight;

txbCount.Text = selectedItem.Count.ToString();

}

//Кнопка назад

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

{

NavigationService.GoBack();

}

//Кнопка изменения

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

{

try

{

var editJewelry = ConnectClass.db.Jewelry.FirstOrDefault(item => item.ID == selectedItem.ID);

//Добавление изображения

MemoryStream stream = new MemoryStream();

JpegBitmapEncoder encoder = new JpegBitmapEncoder();

encoder.Frames.Add(BitmapFrame.Create((BitmapImage)imgLoad.Source));

encoder.Save(stream);

editJewelry.JewImg = stream.ToArray();

editJewelry.JewName = jewNameTxb.Text;

editJewelry.Category.Title = categoryCmb.Text;

editJewelry.Material = materialTxb.Text;

editJewelry.Pice = Convert.ToInt64(priceTxb.Text);

editJewelry.Parameters.Height = heightTxb.Text;

editJewelry.Parameters.Width = widthTxb.Text;

editJewelry.Parameters.Weight = weightTxb.Text;

editJewelry.Count = Convert.ToInt32(txbCount.Text);

ConnectClass.db.SaveChanges();

MessageBox.Show("Данные успешно обновились.", "Операция прошла успешно!", MessageBoxButton.OK, MessageBoxImage.Information);

NavigationService.GoBack();

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка: {ex.Message}", "Ошибка при сохранении!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

//Кнопка открытия изображения

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

{

try

{

OpenFileDialog file = new OpenFileDialog();

file.Filter = "Image (\*.png; \*.jpeg; \*.jpg;) | \*.png; \*.jpeg; \*.jpg;";

if (file.ShowDialog() == true)

{

BitmapImage imgBitmap = new BitmapImage(new Uri(file.FileName));

imgLoad.Source = imgBitmap;

}

}

catch

{

MessageBox.Show("Выберите пожалуйста картинку.", "Произошла ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void priceTxb\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

e.Handled = "0123456789".IndexOf(e.Text) < 0;

}

private void heightTxb\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

e.Handled = "0123456789".IndexOf(e.Text) < 0;

}

private void widthTxb\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

e.Handled = "0123456789".IndexOf(e.Text) < 0;

}

private void weightTxb\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

e.Handled = "0123456789".IndexOf(e.Text) < 0;

}

private void txbCount\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

e.Handled = "0123456789".IndexOf(e.Text) < 0;

}

}

}

## XAML

<Page x:Class="JewerlyStore.View.Pages.FunctionsWithData.EditPage"

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:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:local="clr-namespace:JewerlyStore.View.Pages.FunctionsWithData"

mc:Ignorable="d"

d:DesignHeight="600" d:DesignWidth="1000"

Title="Обновление данных">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="0.2\*"/>

</Grid.RowDefinitions>

<Grid Grid.Row="0">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid Grid.Column="0">

<StackPanel VerticalAlignment="Center"

HorizontalAlignment="Center"

Orientation="Vertical">

<Image x:Name="imgLoad"

Width="200"

Height="200"

Margin="0,0,0,20"/>

<Button x:Name="openBtn"

materialDesign:ButtonAssist.CornerRadius="5"

Content="Выбрать"

Background="Black"

Foreground="White"

Height="35" Click="openBtn\_Click"/>

</StackPanel>

</Grid>

<Grid Grid.Column="1">

<StackPanel VerticalAlignment="Center"

HorizontalAlignment="Center"

Orientation="Vertical">

<StackPanel Orientation="Horizontal"

Margin="0,0,0,20">

<TextBox x:Name="jewNameTxb"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

FontSize="15"

materialDesign:HintAssist.Hint="Изделие"

Width="200"/>

</StackPanel>

<StackPanel Orientation="Horizontal"

Margin="0,0,0,20">

<ComboBox x:Name="categoryCmb"

Style="{StaticResource MaterialDesignFloatingHintComboBox}"

FontSize="15"

materialDesign:HintAssist.Hint="Категория"

Width="200"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBox x:Name="materialTxb"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

materialDesign:HintAssist.Hint="Материал"

FontSize="15"

Width="200"/>

</StackPanel>

<StackPanel Orientation="Horizontal"

Margin="0,0,0,20">

<TextBox x:Name="priceTxb"

PreviewTextInput="priceTxb\_PreviewTextInput"

materialDesign:HintAssist.Hint="Цена"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

FontSize="15"

Width="200"/>

</StackPanel>

<StackPanel Orientation="Horizontal"

Margin="0,0,0,20">

<TextBox x:Name="heightTxb"

PreviewTextInput="heightTxb\_PreviewTextInput"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

materialDesign:HintAssist.Hint="Высота"

FontSize="15"

Width="200"/>

</StackPanel>

<StackPanel Orientation="Horizontal"

Margin="0,0,0,20">

<TextBox x:Name="widthTxb"

PreviewTextInput="widthTxb\_PreviewTextInput"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

materialDesign:HintAssist.Hint="Ширина"

FontSize="15"

Width="200"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBox x:Name="weightTxb"

PreviewTextInput="weightTxb\_PreviewTextInput"

materialDesign:HintAssist.Hint="Вес"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

FontSize="15"

Width="200"/>

</StackPanel>

<StackPanel>

<TextBox Name="txbCount"

PreviewTextInput="txbCount\_PreviewTextInput"

materialDesign:HintAssist.Hint="Количество товара"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

FontSize="15"

Margin="0,10,0,0"

Width="200"/>

</StackPanel>

</StackPanel>

</Grid>

</Grid>

<Grid Grid.Row="1">

<StackPanel VerticalAlignment="Center"

HorizontalAlignment="Center"

Orientation="Horizontal">

<Button x:Name="backBtn"

materialDesign:ButtonAssist.CornerRadius="5"

Content="Назад"

Background="Black"

Foreground="White"

Width="200"

Height="35"

Margin="0,0,20,0" Click="backBtn\_Click"/>

<Button x:Name="editBtn"

Content="Изменить"

materialDesign:ButtonAssist.CornerRadius="5"

Background="Black"

Foreground="White"

Width="200"

Height="35" Click="editBtn\_Click"/>

</StackPanel>

</Grid>

</Grid>

</Page>

# Окно печатать чек

## C#

using JewerlyStore.Classes;

using JewerlyStore.DB;

using System;

using System.Linq;

using System.Windows;

using Word = Microsoft.Office.Interop.Word;

using System.Windows.Controls;

namespace JewerlyStore.View.Pages.FunctionsWithData

{

/// <summary>

/// Логика взаимодействия для GetCheckPage.xaml

/// </summary>

public partial class GetCheckPage : Page

{

private Check Check;

public GetCheckPage(Check check)

{

InitializeComponent();

this.Check = check;

txbFioClient.Text = check.Client.FullName;

txbJewName.Text = check.Jewelry.JewelryCheckGet;

txbDate.Text = check.Date.ToString();

txbCommission.Text = check.Commission.ToString();

txbTotalPrice.Text = check.TotalPrice.ToString();

txbCount.Text = check.Count.ToString();

}

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

{

PrintDialog print = new PrintDialog();

if (print.ShowDialog() == true)

{

print.PrintVisual(pirntStackpanel, "Check");

}

MessageBox.Show("Оформление чека прошло успешно.", "Чек сохранен!", MessageBoxButton.OK, MessageBoxImage.Information);

NavigationService.Navigate(new MenuPage());

}

}

}

## XAML

<Page x:Class="JewerlyStore.View.Pages.FunctionsWithData.GetCheckPage"

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:JewerlyStore.View.Pages.FunctionsWithData"

mc:Ignorable="d"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

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

Title="Получить чек"

FontSize="15">

<Grid>

<StackPanel HorizontalAlignment="Center"

VerticalAlignment="Center">

<StackPanel Orientation="Horizontal"

Name="pirntStackpanel"

Margin="40">

<StackPanel Margin="0,0,30,0">

<TextBlock Text="ФИО Клиента: "/>

<TextBlock Text="Изделие: "

Margin="0,10,0,10"/>

<TextBlock Text="Дата покупки: "/>

<TextBlock Text="Коммиссия: "

Margin="0,10,0,10"/>

<TextBlock Text="Коичество: "

Margin="0,0,0,10"/>

<TextBlock Text="Итого: "/>

</StackPanel>

<StackPanel Width="400">

<TextBlock Name="txbFioClient"

Text="Client FIO"/>

<TextBlock Name="txbJewName"

Margin="0,10,0,10"

Text="Jewelry Name"/>

<TextBlock Name="txbDate"

Text="Date"/>

<TextBlock Name="txbCommission"

Text="Commission"

Margin="0,10,0,10"/>

<TextBlock Name="txbCount" Text="Count"/>

<TextBlock Text="Total Price"

Name="txbTotalPrice"

Margin="0,10,0,0"/>

</StackPanel>

</StackPanel>

<StackPanel HorizontalAlignment="Center"

VerticalAlignment="Center"

Margin="20">

<Button Content="Печатать чек"

Width="200"

materialDesign:ButtonAssist.CornerRadius="10"

Foreground="White"

Background="Black"

Name="btnCheckGet"

Click="btnCheckGet\_Click"/>

</StackPanel>

</StackPanel>

</Grid>

</Page>

# История заказов

## C#

using JewerlyStore.Classes;

using System;

using System.Diagnostics;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Navigation;

using Word = Microsoft.Office.Interop.Word;

namespace JewerlyStore.View.Pages.FunctionsWithData

{

/// <summary>

/// Логика взаимодействия для HistoryPage.xaml

/// </summary>

public partial class HistoryPage : Page

{

public HistoryPage()

{

InitializeComponent();

ChecksList.ItemsSource = ConnectClass.db.Check.ToList();

}

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

{

NavigationService.GoBack();

}

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

{

ChecksList.ItemsSource = ConnectClass.db.Check.Where(item => item.Date == dtpDate.SelectedDate).ToList();

}

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

{

dtpDate.SelectedDate = null;

ChecksList.ItemsSource = ConnectClass.db.Check.ToList();

}

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

{

var word = new Word.Application();

try

{

var documents = word.Documents.Add();

var paragrah = word.ActiveDocument.Paragraphs.Add();

var tableRange = paragrah.Range;

var checksList = ConnectClass.db.Check.ToList();

var table = documents.Tables.Add(tableRange, checksList.Count, 6);

table.Borders.Enable = 1;

table.Cell(1, 1).Range.Text = "ФИО Клиента";

table.Cell(1, 2).Range.Text = "Изделие";

table.Cell(1, 3).Range.Text = "Продано шт.";

table.Cell(1, 4).Range.Text = "Телефон";

table.Cell(1, 5).Range.Text = "Дата";

table.Cell(1, 6).Range.Text = "Сумма";

int i = 2;

foreach (var item in checksList)

{

table.Cell(i, 1).Range.Text = item.Client.FullName;

table.Cell(i, 2).Range.Text = item.Jewelry.JewelryCheckGet;

table.Cell(i, 3).Range.Text = item.Count.ToString();

table.Cell(i, 4).Range.Text = item.Client.Phone;

table.Cell(i, 5).Range.Text = item.Date.ToString();

table.Cell(i, 6).Range.Text = item.TotalPrice.ToString();

i++;

}

documents.SaveAs2(Environment.CurrentDirectory + @"Checks.pdf", Word.WdSaveFormat.wdFormatPDF);

documents.Close(Word.WdSaveOptions.wdDoNotSaveChanges);

word.Quit(Word.WdSaveOptions.wdDoNotSaveChanges);

if (MessageBox.Show("Документ PDF успешно сформирован. Хотите открыть его?", "Операция прошла успешно!", MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)

{

Process.Start(Environment.CurrentDirectory + @"Checks.pdf");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, ex.Source + " выдал исключение!", MessageBoxButton.OK, MessageBoxImage.Error);

word.Quit(Word.WdSaveOptions.wdDoNotSaveChanges);

}

}

}

}

## XAML

<Page x:Class="JewerlyStore.View.Pages.FunctionsWithData.HistoryPage"

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:JewerlyStore.View.Pages.FunctionsWithData"

mc:Ignorable="d"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

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

Title="История заказов">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="8\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid Grid.Row="0">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<StackPanel HorizontalAlignment="Left"

Grid.Column="0"

Orientation="Horizontal">

<Button Content="Назад"

Width="100"

materialDesign:ButtonAssist.CornerRadius="5"

Foreground="White"

Background="Black"

Margin="5"

Click="btnBack\_Click"/>

</StackPanel>

<StackPanel Grid.Column="1"

Orientation="Horizontal">

<DatePicker Name="dtpDate"

Width="200"

Margin="10"

SelectedDateChanged="dtpDate\_SelectedDateChanged"

materialDesign:HintAssist.Hint="Выберите дату"

Style="{StaticResource MaterialDesignDatePicker}"/>

<Button Content="Сбросить"

Foreground="White"

Background="Black"

BorderBrush="Black"

materialDesign:ButtonAssist.CornerRadius="5"

Margin="10,0,0,0"

Click="AllCheck\_Click"/>

</StackPanel>

</Grid>

<Grid Grid.Row="1">

<ListView Name="ChecksList">

<ListView.ItemTemplate>

<DataTemplate>

<StackPanel>

<Label Content="{Binding ID}" ContentStringFormat="ID: {0}"/>

<Label Content="{Binding Client.FullName}" ContentStringFormat="ФИО Клиента: {0}"/>

<Label Content="{Binding Jewelry.JewelryCheckGet}" ContentStringFormat="Изделие: {0}"/>

<Label Content="{Binding Count}" ContentStringFormat="Количество проданного товара: {0}"/>

<Label Content="{Binding Client.Phone}" ContentStringFormat="Телефон: {0}"/>

<Label Content="{Binding Date}" ContentStringFormat="Дата покупки: {0}"/>

<Label Content="{Binding TotalPrice}" ContentStringFormat="Итоговая цена: {0} рублей"/>

</StackPanel>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

<Grid Grid.Row="2">

<StackPanel HorizontalAlignment="Right"

VerticalAlignment="Center">

<Button Content="Вывести PDF"

Foreground="White"

Background="Black"

Width="200"

Margin="0,0,10,0"

materialDesign:ButtonAssist.CornerRadius="10"

Click="btnExportPDF\_Click"/>

</StackPanel>

</Grid>

</Grid>

</Page>

# Офрмление покупки

## C#

using JewerlyStore.Classes;

using JewerlyStore.DB;

using JewerlyStore.View.Pages.PlaceHolder;

using System;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

namespace JewerlyStore.View.Pages.FunctionsWithData

{

/// <summary>

/// Логика взаимодействия для PaymentPage.xaml

/// </summary>

public partial class PaymentPage : Page

{

private int \_balance = 1;

private int \_countJew { get; set; }

private int \_count = 1;

private float \_total { get; set; }

private float \_price { get; set; }

public PaymentPage()

{

InitializeComponent();

txbCount.Visibility = Visibility.Hidden;

cmbSelectClient.ItemsSource = ConnectClass.db.Client.ToList();

cmbSelectClient.DisplayMemberPath = "FullName";

cmbSelectJewely.ItemsSource = ConnectClass.db.Jewelry.ToList();

cmbSelectJewely.DisplayMemberPath = "JewelryGet";

}

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

{

try

{

Check check = new Check();

check.IDClient = (cmbSelectClient.SelectedItem as Client).ID;

check.IDJewelry = (cmbSelectJewely.SelectedItem as Jewelry).ID;

check.Date = DateTime.Now;

check.TotalPrice = \_total;

check.Count = \_count;

ConnectClass.db.Check.Add(check);

var selectedCount = ConnectClass.db.Jewelry.FirstOrDefault(item => item.ID == check.IDJewelry);

selectedCount.Count = \_balance;

ConnectClass.db.SaveChanges();

NavigationService.Navigate(new OrderDone(check));

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

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

{

\_countJew = (cmbSelectJewely.SelectedItem as Jewelry).Count;

\_total = (cmbSelectJewely.SelectedItem as Jewelry).Pice;

\_price = \_total;

txbCount.Visibility = Visibility.Visible;

if(\_countJew != 0)

{

\_count = 1;

txbCount.Text = \_count.ToString();

txbTotalPrice.Text = \_total.ToString();

}

else

{

MessageBox.Show("Данного товара нет в наличии.", "Внимание!", MessageBoxButton.OK, MessageBoxImage.Exclamation);

txbCount.Text = "0";

txbTotalPrice.Text = "0";

}

}

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

{

NavigationService.GoBack();

}

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

{

txbCount.Visibility = Visibility.Visible;

if(\_countJew != 0)

{

if (\_countJew == \_count)

{

MessageBox.Show("Вы первысили количество товара, которое имеется в наличии склада.", "Внимание!", MessageBoxButton.OK, MessageBoxImage.Exclamation);

}

else

{

\_count++;

\_balance = \_countJew - \_count;

\_total += \_price;

}

txbTotalPrice.Text = \_total.ToString();

txbCount.Text = \_count.ToString();

}

else

{

MessageBox.Show("Вы первысили количество товара, которое имеется в наличии склада.", "Внимание!", MessageBoxButton.OK, MessageBoxImage.Exclamation);

}

}

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

{

if (\_count != 0)

{

\_count--;

\_total -= \_price;

}

\_balance = \_countJew + \_count;

txbTotalPrice.Text = \_total.ToString();

txbCount.Text = \_count.ToString();

}

}

}

XAML

<Page x:Class="JewerlyStore.View.Pages.FunctionsWithData.PaymentPage"

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:JewerlyStore.View.Pages.FunctionsWithData"

mc:Ignorable="d"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

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

Title="Оформление покупки">

<Grid>

<StackPanel VerticalAlignment="Top"

HorizontalAlignment="Left">

<Button Name="btnBack"

Content="Назад"

Margin="10"

Foreground="White"

Background="Black"

Width="150"

Click="btnBack\_Click"

materialDesign:ButtonAssist.CornerRadius="10"/>

</StackPanel>

<StackPanel HorizontalAlignment="Center"

VerticalAlignment="Center">

<StackPanel VerticalAlignment="Top"

HorizontalAlignment="Left">

<TextBlock Text="Оформите заказ"

FontSize="14"/>

</StackPanel>

<ComboBox Name="cmbSelectClient"

Width="400"

Style="{StaticResource MaterialDesignFloatingHintComboBox}"

materialDesign:HintAssist.Hint="Выберите клиента"/>

<StackPanel Orientation="Horizontal">

<ComboBox Name="cmbSelectJewely"

Width="280"

SelectionChanged="cmbSelectJewely\_SelectionChanged"

Style="{StaticResource MaterialDesignFloatingHintComboBox}"

materialDesign:HintAssist.Hint="Выберите изделие"/>

<TextBlock Name="txbCount"

Width="30"

FontSize="15"

Text="1"

Margin="10,0,0,0"

VerticalAlignment="Center"/>

<Button Content="+"

Click="btnAddCount\_Click"

Foreground="White"

Background="Black"/>

<Button Content="-"

Foreground="White"

Background="Black"

Click="btnRemoveCount\_Click"/>

</StackPanel>

<TextBox Name="txbTotalPrice"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

materialDesign:HintAssist.Hint="Итоговая цена"/>

<Button Content="Офорить покупку"

Foreground="White"

Background="Black"

Margin="0,5,0,0"

Name="btnOrderDone"

Click="btnOrderDone\_Click"

materialDesign:ButtonAssist.CornerRadius="15"

/>

</StackPanel>

</Grid>

</Page>

# Окно PlaceHolder Полчуить чек

## C#

using JewerlyStore.DB;

using JewerlyStore.View.Pages.FunctionsWithData;

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 JewerlyStore.View.Pages.PlaceHolder

{

/// <summary>

/// Логика взаимодействия для OrderDone.xaml

/// </summary>

public partial class OrderDone : Page

{

private Check check;

public OrderDone(Check check)

{

InitializeComponent();

this.check = check;

}

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

{

NavigationService.Navigate(new MenuPage());

}

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

{

NavigationService.Navigate(new GetCheckPage(check));

}

}

}

## XAML

<Page x:Class="JewerlyStore.View.Pages.PlaceHolder.OrderDone"

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:JewerlyStore.View.Pages.PlaceHolder"

mc:Ignorable="d"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

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

Title="Покупка оформлена.">

<Grid>

<StackPanel VerticalAlignment="Top"

HorizontalAlignment="Left">

<Button Content="Отмена"

materialDesign:ButtonAssist.CornerRadius="5"

Name="btnCancel"

Margin="10"

Foreground="White"

Background="Black"

FontSize="15"

Click="btnCancel\_Click"

Width="150"/>

</StackPanel>

<StackPanel HorizontalAlignment="Center"

VerticalAlignment="Center">

<TextBlock Text="Заказ оформлен, перейдите дальше для получения чека."

FontSize="22"

FontFamily="century gothic"/>

<Button Foreground="White"

Background="Black"

Content="Получить чек"

FontSize="15"

Name="btnGetCheck"

Click="btnGetCheck\_Click"

FontFamily="century gothic"

materialDesign:ButtonAssist.CornerRadius="10"

Margin="0,10,0,0"/>

</StackPanel>

</Grid>

</Page>

# Окно авторизации

## C#

using JewerlyStore.Classes;

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 JewerlyStore.View.Pages

{

/// <summary>

/// Логика взаимодействия для AutorizationPage.xaml

/// </summary>

public partial class AutorizationPage : Page

{

public AutorizationPage()

{

InitializeComponent();

}

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

{

try

{

var currentUser = ConnectClass.db.SignIn.FirstOrDefault(item => item.LogIn == loginTxb.Text && item.Password == passTxb.Password);

if (currentUser != null)

{

switch (currentUser.RoleID)

{

case "A":

NavigationService.Navigate(new MenuPage());

break;

}

}

else

{

throw new Exception("Проверьте корректность введенных данных!");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, ex.Source + " выдал исключение", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

Application.Current.Shutdown();

}

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

{

NavigationService.Navigate(new RegistrationPageView());

}

}

}

## XAML

<Page x:Class="JewerlyStore.View.Pages.AutorizationPage"

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:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:local="clr-namespace:JewerlyStore.View.Pages"

mc:Ignorable="d"

d:DesignHeight="600" d:DesignWidth="1000"

Title="Вход">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid Background="#FFECECEC" Grid.Column="1">

<StackPanel VerticalAlignment="Center"

HorizontalAlignment="Center"

Orientation="Vertical">

<StackPanel Orientation="Horizontal"

Margin="0,0,0,10">

<TextBox x:Name="loginTxb"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

materialDesign:HintAssist.Hint="Имя пользователя"

FontSize="15"

Width="250"/>

</StackPanel>

<StackPanel Orientation="Horizontal"

Margin="0,0,0,10">

<PasswordBox x:Name="passTxb"

FontSize="15"

Style="{StaticResource MaterialDesignFloatingHintPasswordBox}"

materialDesign:HintAssist.Hint="Пароль"

Width="250"/>

</StackPanel>

<StackPanel Orientation="Horizontal"

VerticalAlignment="Center"

HorizontalAlignment="Right">

<Button x:Name="comeBtn"

materialDesign:ButtonAssist.CornerRadius="5"

Content="Войти"

Width="120"

Height="35"

Background="Black"

Foreground="White"

Click="comeBtn\_Click"

IsDefault="True"/>

<Button Content="Отмена"

materialDesign:ButtonAssist.CornerRadius="5"

Background="Black"

Margin="10,0,0,0"

Foreground="White"

Width="120"

Height="35"

Name="btnCancel"

Click="btnCancel\_Click"/>

</StackPanel>

<StackPanel>

<Button Foreground="White"

Background="Black"

materialDesign:ButtonAssist.CornerRadius="10"

Content="Регистрация"

Margin="0,5,0,0"

Click="btnRegistration\_Click"/>

</StackPanel>

</StackPanel>

</Grid>

<Grid Grid.Column="0"

Background="#FFECECEC">

<Image Source="/Resources/la.png"/>

</Grid>

</Grid>

</Page>

# Окно рабочей зоны с товарами

## C#

using JewerlyStore.Classes;

using JewerlyStore.DB;

using JewerlyStore.View.Pages.FunctionsWithData;

using JewerlyStore.View.Windows;

using System;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Navigation;

using Word = Microsoft.Office.Interop.Word;

namespace JewerlyStore.View.Pages

{

/// <summary>

/// Логика взаимодействия для MainViewPage.xaml

/// </summary>

public partial class MainViewPage : Page

{

public MainViewPage()

{

InitializeComponent();

}

//Инициализация

private void Page\_Loaded(object sender, RoutedEventArgs e)

{

listDataView.ItemsSource = ConnectClass.db.Jewelry.ToList();

}

//Поиск

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

{

listDataView.ItemsSource = ConnectClass.db.Jewelry.Where(item => item.JewName.Contains(searchTxb.Text) || item.Material.Contains(searchTxb.Text) || item.Category.Title.Contains(searchTxb.Text)).ToList();

}

//Кнопка назад

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

{

NavigationService.GoBack();

}

//Добавление

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

{

NavigationService.Navigate(new AddPage());

}

//Удаление

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

{

try

{

Jewelry removeJewerly = (Jewelry)listDataView.SelectedItem;

if(removeJewerly != null)

{

if(MessageBox.Show("Вы точно хотите удалить выбранный элемент? Данные будут удалены навсегда!", "Вы уверены?", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

ConnectClass.db.Jewelry.Remove(removeJewerly);

ConnectClass.db.SaveChanges();

Page\_Loaded(null, null);

}

}

else

{

throw new Exception("Выберите элемент!");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, ex.Source + " выдал исключение", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

//Экспорт данных в PDF

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

{

ExportWindow window = new ExportWindow();

window.ShowDialog();

}

//Двойное нажатие - изменение

private void listDataView\_MouseDoubleClick(object sender, System.Windows.Input.MouseButtonEventArgs e)

{

Jewelry editJewelry = (Jewelry)listDataView.SelectedItem;

if(editJewelry != null)

{

NavigationService.Navigate(new EditPage(editJewelry));

}

}

}

}

## XAML

<Page x:Class="JewerlyStore.View.Pages.MainViewPage"

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:JewerlyStore.View.Pages"

mc:Ignorable="d" Loaded="Page\_Loaded"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

d:DesignHeight="600" d:DesignWidth="1000"

Title="Рабочая зона">

<Grid Background="White">

<Grid.RowDefinitions>

<RowDefinition Height="0.2\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="0.2\*"/>

</Grid.RowDefinitions>

<Grid Grid.Row="0">

<StackPanel VerticalAlignment="Center"

HorizontalAlignment="Center"

Orientation="Horizontal">

<TextBox x:Name="searchTxb"

FontSize="15"

materialDesign:HintAssist.Hint="Поиск"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

Width="250"

TextChanged="searchTxb\_TextChanged"/>

</StackPanel>

</Grid>

<Grid Grid.Row="1">

<ListView x:Name="listDataView" MouseDoubleClick="listDataView\_MouseDoubleClick">

<ListView.ItemTemplate>

<DataTemplate>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid Grid.Column="0" Margin="0,0,30,0">

<Image Source="{Binding JewImg}"

Width="200"

Height="200"/>

</Grid>

<Grid Grid.Column="1">

<StackPanel VerticalAlignment="Center"

HorizontalAlignment="Center"

Orientation="Vertical">

<Label Content="{Binding JewName}" ContentStringFormat="Название: {0}" FontSize="14" Foreground="Black" FontWeight="Bold"/>

<Label Content="{Binding Category.Title}" ContentStringFormat="Категория: {0}" FontSize="14" Foreground="Black"/>

<Label Content="{Binding Material}" ContentStringFormat="Материал: {0}" FontSize="14" Foreground="Black"/>

<Label Content="{Binding Pice}" ContentStringFormat="Цена: {0} руб." FontSize="14" Foreground="Black"/>

<Label Content="{Binding Parameters.Height}" ContentStringFormat="Высота изделия: {0} см." FontSize="14" Foreground="Black"/>

<Label Content="{Binding Parameters.Width}" ContentStringFormat="Ширина изделия: {0} см." FontSize="14" Foreground="Black"/>

<Label Content="{Binding Parameters.Weight}" ContentStringFormat="Вес изделия: {0} г." FontSize="14" Foreground="Black"/>

<Label Content="{Binding Count}" ContentStringFormat="Количество: {0} шт." FontSize="14" Foreground="Black"/>

</StackPanel>

</Grid>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

<Grid Grid.Row="2">

<StackPanel VerticalAlignment="Center"

HorizontalAlignment="Left"

Margin="20">

<Button x:Name="backBtn"

materialDesign:ButtonAssist.CornerRadius="5"

Content="Назад"

Background="Black"

Foreground="White"

Width="150"

Height="35" Click="backBtn\_Click"/>

</StackPanel>

<StackPanel VerticalAlignment="Center"

HorizontalAlignment="Center"

Orientation="Horizontal">

<Button x:Name="removeBtn"

Content="Удалить"

Background="Black"

Foreground="White"

materialDesign:ButtonAssist.CornerRadius="5"

Width="150"

Height="35"

Margin="0,0,20,0" Click="removeBtn\_Click"/>

<Button x:Name="addBtn"

Content="Добавить"

Background="Black"

Foreground="White"

materialDesign:ButtonAssist.CornerRadius="5"

Width="150"

Height="35" Click="addBtn\_Click"/>

</StackPanel>

<StackPanel VerticalAlignment="Center"

HorizontalAlignment="Right"

Margin="20">

<Button x:Name="pdfBtn"

Content="Вывести"

Background="Black"

Foreground="White"

Width="150"

materialDesign:ButtonAssist.CornerRadius="5"

Height="35" Click="pdfBtn\_Click"/>

</StackPanel>

</Grid>

</Grid>

</Page>

# Окно меню

## C#

using JewerlyStore.View.Pages.FunctionsWithData;

using JewerlyStore.View.Windows;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Navigation;

namespace JewerlyStore.View.Pages

{

/// <summary>

/// Логика взаимодействия для MenuPage.xaml

/// </summary>

public partial class MenuPage : Page

{

public MenuPage()

{

InitializeComponent();

}

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

{

NavigationService.Navigate(new CheckoutPage());

}

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

{

NavigationService.Navigate(new MainViewPage());

}

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

{

About about = new About();

about.ShowDialog();

}

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

{

NavigationService.Navigate(new HistoryPage());

}

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

{

if (MessageBox.Show("Вы действительно хотите выйти?", "Уверены.", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

NavigationService.Navigate(new AutorizationPage());

}

}

}

## XAML

<Page x:Class="JewerlyStore.View.Pages.MenuPage"

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:JewerlyStore.View.Pages"

mc:Ignorable="d" FontSize="18"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

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

Title="Меню">

<Grid>

<StackPanel HorizontalAlignment="Center"

VerticalAlignment="Center">

<Button Content="Оформить заказ"

materialDesign:ButtonAssist.CornerRadius="5"

Name="btnCheckout"

Foreground="White"

Background="Black"

Width="350"

Click="btnCheckout\_Click"/>

<Button Content="Товар"

materialDesign:ButtonAssist.CornerRadius="5"

Name="btnProducts"

Foreground="White"

Background="Black"

Click="btnProducts\_Click"

Margin="0,10,0,10"/>

<Button Content="История заказа"

Foreground="White"

materialDesign:ButtonAssist.CornerRadius="5"

Background="Black"

Name="btnHistory"

Click="btnHistory\_Click"/>

<Button Content="О нас"

Name="btnAbout"

Click="btnAbout\_Click"

Margin="0,10,0,10"

materialDesign:ButtonAssist.CornerRadius="5"

Foreground="White"

Background="Black"/>

<Button Content="Выйти"

Name="btnExit"

Click="btnExit\_Click"

materialDesign:ButtonAssist.CornerRadius="5"

Foreground="White"

Background="Black"/>

</StackPanel>

</Grid>

</Page>

# Окно регистрации

## C#

using JewerlyStore.Classes;

using JewerlyStore.DB;

using JewerlyStore.HelperClass;

using System;

using System.Linq;

using System.Net;

using System.Net.Mail;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Navigation;

namespace JewerlyStore.View.Pages

{

/// <summary>

/// Interaction logic for RegistrationPageView.xaml

/// </summary>

public partial class RegistrationPageView : Page

{

public RegistrationPageView()

{

InitializeComponent();

}

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

{

try

{

if (HelperObject.IsValidEmail(txbEmail.Text))

{

if (ConnectClass.db.Regisration.Count(item => item.Email == txbEmail.Text) > 0)

{

MessageBox.Show("Учётная запись существует. Пожалуйста, если вы потеряли доступ, обратитесь к Администартору Системы.",

"В доступе отказано!", MessageBoxButton.OK, MessageBoxImage.Warning);

return;

}

else

{

SendMessage();

Regisration regisration = new Regisration();

SignIn signIn = new SignIn();

signIn.LogIn = txbEmail.Text;

signIn.Password = currentPassword;

signIn.RoleID = "A";

regisration.Email = txbEmail.Text;

regisration.FirstName = txbFirstName.Text;

regisration.LastName = txbLastName.Text;

regisration.Patronymic = txbPatronymic.Text;

regisration.NumberContract = txbNumberContract.Text;

regisration.Date = DateTime.Now;

regisration.IDUser = signIn.ID;

ConnectClass.db.SignIn.Add(signIn);

ConnectClass.db.Regisration.Add(regisration);

ConnectClass.db.SaveChanges();

}

}

else

{

MessageBox.Show("Некорректный формат почты! Пожалуйста, внимательно прочтите, какой набор символов вы ввели и повторите попытку!",

"Некорректный формать логина!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, ex.Source, MessageBoxButton.OK, MessageBoxImage.Error);

}

}

public static string GenericPassword()

{

// Получаем количество слов и букв за слово.

int num\_letters = 8;

int num\_words = 1;

// Создаем массив букв, которые мы будем использовать.

char[] letters = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdifghijklmnopqrstuvwxyz123456789".ToCharArray();

// Создаем генератор случайных чисел.

Random rand = new Random();

// Сделайте слово.

string word = "";

// Делаем слова.

for (int i = 1; i <= num\_words; i++)

{

for (int j = 1; j <= num\_letters; j++)

{

// Выбор случайного числа от 0 до 25

// для выбора буквы из массива букв.

int letter\_num = rand.Next(0, letters.Length - 1);

// Добавить письмо.

word += letters[letter\_num];

}

}

return word;

}

static string currentPassword = GenericPassword();

private async void SendMessage()

{

try

{ // отправитель - устанавливаем адрес и отображаемое в письме имя

MailAddress from = new MailAddress("jewralyapp@mail.ru", "Admin JewralyApp SoftWare");

// кому отправляем

MailAddress to = new MailAddress(txbEmail.Text);

// создаем объект сообщения

MailMessage m = new MailMessage(from, to);

// тема письма

m.Subject = "Администратор одобрил ваш запрос!";

// текст письма

m.Body = "<h2>Ваш логин: " + txbEmail.Text + ", Ваш пароль: " + currentPassword + " </h2>";

// письмо представляет код html

m.IsBodyHtml = true;

// адрес smtp-сервера и порт, с которого будем отправлять письмо

SmtpClient smtp = new SmtpClient("smtp.mail.ru", 587);

// логин и пароль

smtp.Credentials = new NetworkCredential("jewralyapp@mail.ru", "@Lacky12345");

smtp.EnableSsl = true;

await smtp.SendMailAsync(m);

Console.Read();

MessageBox.Show("Ваш запрос отправлен, Администратор системы рассмотрит ваш запрос как можно быстрее. Дождитесь пожалуйста ответа.",

"Запрос успешно отправлен!", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

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

}

}

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

{

NavigationService.GoBack();

}

}

}

## XAML

<Page x:Class="JewerlyStore.View.Pages.RegistrationPageView"

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:JewerlyStore.View.Pages"

mc:Ignorable="d" FontSize="16"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

d:DesignHeight="600" d:DesignWidth="1000"

Title="Регистрация пользователя">

<Grid>

<StackPanel HorizontalAlignment="Right"

VerticalAlignment="Top"

Margin="10">

<Button Content="Назад"

Foreground="White"

Background="Black"

Width="150"

Click="btnBack\_Click"

materialDesign:ButtonAssist.CornerRadius="5"/>

</StackPanel>

<StackPanel HorizontalAlignment="Center"

VerticalAlignment="Center">

<TextBox Width="250"

Name="txbEmail"

materialDesign:HintAssist.Hint="Введите почту: "

Style="{StaticResource MaterialDesignFloatingHintTextBox}"/>

<TextBox Width="250"

Name="txbFirstName"

materialDesign:HintAssist.Hint="Имя: "

Style="{StaticResource MaterialDesignFloatingHintTextBox}"/>

<TextBox Width="250"

Name="txbLastName"

materialDesign:HintAssist.Hint="Фамилия: "

Style="{StaticResource MaterialDesignFloatingHintTextBox}"/>

<TextBox Width="250"

Name="txbPatronymic"

materialDesign:HintAssist.Hint="Отчество (необязательно)"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"/>

<TextBox Width="250"

Name="txbNumberContract"

materialDesign:HintAssist.Hint="Номер договора: "

Style="{StaticResource MaterialDesignFloatingHintTextBox}"/>

<Button Foreground="White"

Background="Black"

Margin="0, 10,0,0"

Content="Сохранить"

materialDesign:ButtonAssist.CornerRadius="10"

Click="btnSave\_Click"/>

</StackPanel>

</Grid>

</Page>

# Дополнительное окно

## C#

using JewerlyStore.Classes;

using JewerlyStore.DB;

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.IO;

using System.Linq;

using System.Text.Encodings.Web;

using System.Text.Json;

using System.Text.Unicode;

using System.Windows;

using Word = Microsoft.Office.Interop.Word;

namespace JewerlyStore.View.Windows

{

/// <summary>

/// Interaction logic for ExportWindow.xaml

/// </summary>

public partial class ExportWindow : Window

{

public ExportWindow()

{

InitializeComponent();

}

private void ExportPDF()

{

var word = new Word.Application();

try

{

var document = word.Documents.Add();

var paragraph = word.ActiveDocument.Paragraphs.Add();

var tableRange = paragraph.Range;

var jewerlyList = ConnectClass.db.Jewelry.ToList();

var table = document.Tables.Add(tableRange, jewerlyList.Count, 7);

table.Borders.Enable = 1;

table.Cell(1, 1).Range.Text = "Наименование";

table.Cell(1, 2).Range.Text = "Категория";

table.Cell(1, 3).Range.Text = "Материал";

table.Cell(1, 4).Range.Text = "Цена";

table.Cell(1, 5).Range.Text = "Высота";

table.Cell(1, 6).Range.Text = "Ширина";

table.Cell(1, 7).Range.Text = "Вес";

table.Cell(1, 8).Range.Text = "Количество";

int i = 2;

foreach (var item in jewerlyList)

{

table.Cell(i, 0).Range.Text = item.ID.ToString();

table.Cell(i, 1).Range.Text = item.JewName;

table.Cell(i, 2).Range.Text = item.Category.Title;

table.Cell(i, 3).Range.Text = item.Material;

table.Cell(i, 4).Range.Text = (item.Pice).ToString();

table.Cell(i, 5).Range.Text = item.Parameters.Height;

table.Cell(i, 6).Range.Text = item.Parameters.Width;

table.Cell(i, 7).Range.Text = item.Parameters.Weight;

table.Cell(i, 7).Range.Text = item.Count.ToString();

i++;

}

document.SaveAs2(Environment.CurrentDirectory + @"parts.pdf", Word.WdSaveFormat.wdFormatPDF);

document.Close(Word.WdSaveOptions.wdDoNotSaveChanges);

word.Quit(Word.WdSaveOptions.wdDoNotSaveChanges);

MessageBox.Show("Сохранение прошло успешно!", "Сохранено!", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, ex.Source + " выдал исключение!", MessageBoxButton.OK, MessageBoxImage.Error);

word.Quit(Word.WdSaveOptions.wdDoNotSaveChanges);

}

}

private void ExportJSON()

{

var jewelry = ConnectClass.db.Jewelry.ToList();

var option = new JsonSerializerOptions() { Encoder = JavaScriptEncoder.Create(UnicodeRanges.All) };

string line = JsonSerializer.Serialize<List<Jewelry>>(jewelry, option);

File.WriteAllText(Environment.CurrentDirectory + @"parts.json", line);

MessageBox.Show("Сохранение прошло успешно!", "Сохранено!", MessageBoxButton.OK, MessageBoxImage.Information);

}

private void ExportCSV()

{

using (FileStream stream = new FileStream(Environment.CurrentDirectory + @"parts.csv", FileMode.Create))

{

using (StreamWriter writer = new StreamWriter(stream))

{

var service = ConnectClass.db.Jewelry.ToList();

writer.WriteLine("Название;Категория;Материал;Цена;Высота;Ширина;Вес;Количество");

foreach (var item in service)

{

writer.WriteLine($"{item.JewName};{item.JewName};{item.Category.Title};{item.Material};{item.Pice};{item.Parameters.Height};{item.Parameters.Width};{item.Parameters.Weight};{item.Count}");

}

}

}

MessageBox.Show("Сохранение прошло успешно!", "Сохранено!", MessageBoxButton.OK, MessageBoxImage.Information);

}

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

{

ExportPDF();

if (MessageBox.Show("Документ PDF успешно сформирован. Хотите открыть его?", "Операция прошла успешно!", MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)

{

Process.Start(Environment.CurrentDirectory + @"parts.pdf");

}

this.Close();

}

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

{

ExportCSV();

if (MessageBox.Show("Документ CSV успешно сформирован. Хотите открыть его?", "Операция прошла успешно!", MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)

{

Process.Start(Environment.CurrentDirectory + @"parts.csv");

}

this.Close();

}

}

}

## XAML

<Window x:Class="JewerlyStore.View.Windows.ExportWindow"

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:JewerlyStore.View.Windows"

mc:Ignorable="d"

WindowStartupLocation="CenterScreen"

WindowStyle="SingleBorderWindow"

MinHeight="200" MaxHeight="200"

MinWidth="350" MaxWidth="350"

Icon="/Resources/store.ico"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

Title="Вывести" Height="200" Width="350">

<Grid>

<StackPanel HorizontalAlignment="Center"

VerticalAlignment="Center">

<TextBlock Text="В каком формате вывести данные?"

FontSize="18"

Margin="0,0,0,10"/>

<StackPanel Orientation="Horizontal">

<Button Width="150"

Click="btnPDF\_Click"

Content="PDF"

materialDesign:ButtonAssist.CornerRadius="5"

Foreground="White"

Background="Black"/>

<Button Width="150"

Margin="5,0,0,0"

Click="btnCSV\_Click"

materialDesign:ButtonAssist.CornerRadius="5"

Content="CSV"

Foreground="White"

Background="Black"/>

</StackPanel>

</StackPanel>

</Grid>

</Window>

# База данных

USE [JewelryStore]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Category] Script Date: 6/7/2021 12:04:07 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Category](

[ID] [int] IDENTITY(1,1) NOT NULL,

[Title] [nvarchar](100) NOT NULL,

CONSTRAINT [PK\_Category\_ID] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Check] Script Date: 6/7/2021 12:04:07 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Check](

[ID] [int] IDENTITY(1,1) NOT NULL,

[IDClient] [int] NOT NULL,

[IDJewelry] [int] NOT NULL,

[Date] [date] NOT NULL,

[Commission] [float] NULL,

[TotalPrice] [float] NOT NULL,

[Count] [int] NOT NULL,

CONSTRAINT [PK\_Check] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Client] Script Date: 6/7/2021 12:04:07 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Client](

[ID] [int] IDENTITY(1,1) NOT NULL,

[FirstName] [nvarchar](50) NOT NULL,

[LastName] [nvarchar](50) NOT NULL,

[SecondName] [nvarchar](50) NULL,

[Phone] [char](30) NOT NULL,

CONSTRAINT [PK\_Client] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Jewelry] Script Date: 6/7/2021 12:04:07 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Jewelry](

[ID] [int] IDENTITY(1,1) NOT NULL,

[JewImg] [image] NULL,

[JewName] [nvarchar](100) NOT NULL,

[CategoryID] [int] NOT NULL,

[Material] [nvarchar](100) NOT NULL,

[Pice] [bigint] NOT NULL,

[ParametersID] [int] NOT NULL,

[Count] [int] NOT NULL,

CONSTRAINT [PK\_Jewerly\_ID] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Parameters] Script Date: 6/7/2021 12:04:07 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Parameters](

[ID] [int] IDENTITY(1,1) NOT NULL,

[Height] [nvarchar](100) NOT NULL,

[Width] [nvarchar](100) NOT NULL,

[Weight] [nvarchar](100) NOT NULL,

CONSTRAINT [PK\_Parameters\_ID] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Regisration] Script Date: 6/7/2021 12:04:07 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Regisration](

[ID] [int] IDENTITY(1,1) NOT NULL,

[Email] [nvarchar](50) NOT NULL,

[FirstName] [nvarchar](50) NOT NULL,

[LastName] [nvarchar](50) NOT NULL,

[Patronymic] [nvarchar](50) NULL,

[NumberContract] [char](30) NOT NULL,

[Date] [date] NOT NULL,

[IDUser] [int] NOT NULL,

CONSTRAINT [PK\_Regisration] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Role] Script Date: 6/7/2021 12:04:07 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Role](

[ID] [nchar](1) NOT NULL,

[Title] [nvarchar](20) NOT NULL,

CONSTRAINT [PK\_Role\_ID] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[SignIn] Script Date: 6/7/2021 12:04:07 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[SignIn](

[ID] [int] IDENTITY(1,1) NOT NULL,

[LogIn] [nvarchar](100) NOT NULL,

[Password] [nvarchar](100) NOT NULL,

[RoleID] [nchar](1) NOT NULL,

CONSTRAINT [PK\_SignIn\_ID] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

SET IDENTITY\_INSERT [dbo].[Category] ON

INSERT [dbo].[Category] ([ID], [Title]) VALUES (1, N'Золотой')

INSERT [dbo].[Category] ([ID], [Title]) VALUES (2, N'Рубиновый')

INSERT [dbo].[Category] ([ID], [Title]) VALUES (3, N'Алмазное')

INSERT [dbo].[Category] ([ID], [Title]) VALUES (4, N'Арт-декор')

INSERT [dbo].[Category] ([ID], [Title]) VALUES (5, N'Ретро')

SET IDENTITY\_INSERT [dbo].[Category] OFF

GO

SET IDENTITY\_INSERT [dbo].[Check] ON

INSERT [dbo].[Check] ([ID], [IDClient], [IDJewelry], [Date], [Commission], [TotalPrice], [Count]) VALUES (1, 1, 2, CAST(N'2021-05-30' AS Date), NULL, 360000, 3)

INSERT [dbo].[Check] ([ID], [IDClient], [IDJewelry], [Date], [Commission], [TotalPrice], [Count]) VALUES (2, 1, 3, CAST(N'2021-05-30' AS Date), NULL, 900000, 3)

INSERT [dbo].[Check] ([ID], [IDClient], [IDJewelry], [Date], [Commission], [TotalPrice], [Count]) VALUES (3, 2, 4, CAST(N'2021-05-30' AS Date), NULL, 500000, 1)

INSERT [dbo].[Check] ([ID], [IDClient], [IDJewelry], [Date], [Commission], [TotalPrice], [Count]) VALUES (4, 1, 1, CAST(N'2021-05-30' AS Date), NULL, 300000, 3)

INSERT [dbo].[Check] ([ID], [IDClient], [IDJewelry], [Date], [Commission], [TotalPrice], [Count]) VALUES (5, 1, 5, CAST(N'2021-05-30' AS Date), NULL, 800000, 2)

SET IDENTITY\_INSERT [dbo].[Check] OFF

GO

SET IDENTITY\_INSERT [dbo].[Client] ON

INSERT [dbo].[Client] ([ID], [FirstName], [LastName], [SecondName], [Phone]) VALUES (1, N'Зарина', N'Ахмедова', N'', N'+79286549090 ')

INSERT [dbo].[Client] ([ID], [FirstName], [LastName], [SecondName], [Phone]) VALUES (2, N'Маргарита', N'Миронова', N'', N'79298770098 ')

SET IDENTITY\_INSERT [dbo].[Client] OFF

GO

SET IDENTITY\_INSERT [dbo].[Jewelry] ON

INSERT [dbo].[Jewelry] ([ID], [JewImg], [JewName], [CategoryID], [Material], [Pice], [ParametersID], [Count]) VALUES (1, , N'Подарочное кольцо', 1, N'Белое золото', 100000, 1, 10)

INSERT [dbo].[Jewelry] ([ID], [JewImg], [JewName], [CategoryID], [Material], [Pice], [ParametersID], [Count]) VALUES (2, , N'Кольцо', 2, N'Красный рубин', 120000, 2, 17)

INSERT [dbo].[Jewelry] ([ID], [JewImg], [JewName], [CategoryID], [Material], [Pice], [ParametersID], [Count]) VALUES (3, Ожирелье', 4, N'Белое золото', 300000, 3, 17)

INSERT [dbo].[Jewelry] ([ID], [JewImg], [JewName], [CategoryID], [Material], [Pice], [ParametersID], [Count]) VALUES (4, Ожирелье', 5, N'Белое золото', 500000, 4, 1)

INSERT [dbo].[Jewelry] ([ID], [JewImg], [JewName], [CategoryID], [Material], [Pice], [ParametersID], [Count]) VALUES (5, Обручальные кольца', 1, N'Белое золото', 400000, 5, 0)

SET IDENTITY\_INSERT [dbo].[Jewelry] OFF

GO

SET IDENTITY\_INSERT [dbo].[Parameters] ON

INSERT [dbo].[Parameters] ([ID], [Height], [Width], [Weight]) VALUES (1, N'5', N'5', N'5')

INSERT [dbo].[Parameters] ([ID], [Height], [Width], [Weight]) VALUES (2, N'5', N'5', N'5')

INSERT [dbo].[Parameters] ([ID], [Height], [Width], [Weight]) VALUES (3, N'150', N'70', N'70')

INSERT [dbo].[Parameters] ([ID], [Height], [Width], [Weight]) VALUES (4, N'230', N'90', N'90')

INSERT [dbo].[Parameters] ([ID], [Height], [Width], [Weight]) VALUES (5, N'4', N'4', N'4')

SET IDENTITY\_INSERT [dbo].[Parameters] OFF

GO

SET IDENTITY\_INSERT [dbo].[Regisration] ON

INSERT [dbo].[Regisration] ([ID], [Email], [FirstName], [LastName], [Patronymic], [NumberContract], [Date], [IDUser]) VALUES (1, N'abdulkhak1m@mail.ru', N'Abdulkhakim', N'Magomedov', NULL, N'434243 ', CAST(N'2021-06-06' AS Date), 1)

INSERT [dbo].[Regisration] ([ID], [Email], [FirstName], [LastName], [Patronymic], [NumberContract], [Date], [IDUser]) VALUES (2, N'thevckit@gmail.com', N'fjlads', N'jads;', N'', N'fjldsajf321 ', CAST(N'2021-06-06' AS Date), 2)

SET IDENTITY\_INSERT [dbo].[Regisration] OFF

GO

INSERT [dbo].[Role] ([ID], [Title]) VALUES (N'A', N'Admin')

GO

SET IDENTITY\_INSERT [dbo].[SignIn] ON

INSERT [dbo].[SignIn] ([ID], [LogIn], [Password], [RoleID]) VALUES (1, N'a', N'a', N'A')

INSERT [dbo].[SignIn] ([ID], [LogIn], [Password], [RoleID]) VALUES (2, N'thevckit@gmail.com', N'VjG8Q1LO', N'A')

SET IDENTITY\_INSERT [dbo].[SignIn] OFF

GO

ALTER TABLE [dbo].[Check] WITH CHECK ADD CONSTRAINT [FK\_Check\_Client] FOREIGN KEY([IDClient])

REFERENCES [dbo].[Client] ([ID])

GO

ALTER TABLE [dbo].[Check] CHECK CONSTRAINT [FK\_Check\_Client]

GO

ALTER TABLE [dbo].[Check] WITH CHECK ADD CONSTRAINT [FK\_Check\_Jewelry] FOREIGN KEY([IDJewelry])

REFERENCES [dbo].[Jewelry] ([ID])

GO

ALTER TABLE [dbo].[Check] CHECK CONSTRAINT [FK\_Check\_Jewelry]

GO

ALTER TABLE [dbo].[Jewelry] WITH CHECK ADD CONSTRAINT [FK\_Jewerly\_CategoryID\_Category\_ID] FOREIGN KEY([CategoryID])

REFERENCES [dbo].[Category] ([ID])

GO

ALTER TABLE [dbo].[Jewelry] CHECK CONSTRAINT [FK\_Jewerly\_CategoryID\_Category\_ID]

GO

ALTER TABLE [dbo].[Jewelry] WITH CHECK ADD CONSTRAINT [FK\_Jewerly\_ParametersID\_Parameters\_ID] FOREIGN KEY([ParametersID])

REFERENCES [dbo].[Parameters] ([ID])

GO

ALTER TABLE [dbo].[Jewelry] CHECK CONSTRAINT [FK\_Jewerly\_ParametersID\_Parameters\_ID]

GO

ALTER TABLE [dbo].[Regisration] WITH CHECK ADD CONSTRAINT [FK\_Regisration\_SignIn] FOREIGN KEY([IDUser])

REFERENCES [dbo].[SignIn] ([ID])

ON UPDATE CASCADE

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[Regisration] CHECK CONSTRAINT [FK\_Regisration\_SignIn]

GO

ALTER TABLE [dbo].[SignIn] WITH CHECK ADD CONSTRAINT [FK\_SignIn\_RoleID\_Role\_ID] FOREIGN KEY([RoleID])

REFERENCES [dbo].[Role] ([ID])

GO

ALTER TABLE [dbo].[SignIn] CHECK CONSTRAINT [FK\_SignIn\_RoleID\_Role\_ID]

GO