**Приложение**

<Window x:Class="Fitnes.SotrudnikRaspisanie"

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:Fitnes"

mc:Ignorable="d"

Title="Расписание" Height="775" Width="1050" Loaded="Window\_Loaded" WindowStartupLocation="CenterScreen" ResizeMode="NoResize" WindowStyle="None" MouseDown="Window\_MouseDown">

<Window.Resources>

<Style TargetType="{x:Type TextBox}">

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

<Setter Property="BorderBrush" Value="#FFFF8000"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="Template">

<Setter.Value>

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

<Border Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}">

<ScrollViewer x:Name="PART\_ContentHost" />

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

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

<Setter Property="BorderBrush" Value="White"/>

</Trigger>

</Style.Triggers>

</Style>

<Style TargetType="{x:Type PasswordBox}">

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

<Setter Property="BorderBrush" Value="#FFFF8000"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="Template">

<Setter.Value>

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

<Border Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}">

<ScrollViewer x:Name="PART\_ContentHost" />

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

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

<Setter Property="BorderBrush" Value="White"/>

</Trigger>

</Style.Triggers>

</Style>

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

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

<Setter Property="BorderBrush" Value="#FFFF8000"/>

<Setter Property="Foreground" Value="#FFFF8000"/>

<Setter Property="BorderThickness" Value="2"/>

<Setter Property="Template">

<Setter.Value>

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

<Border Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}">

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

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

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

<Setter Property="Background" Value="#FFFF8000"/>

<Setter Property="Foreground" Value="White"/>

</Trigger>

</Style.Triggers>

</Style>

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

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

<Setter Property="BorderBrush" Value="#FFFF2F00"/>

<Setter Property="Foreground" Value="#FFFF2F00"/>

<Setter Property="BorderThickness" Value="2"/>

<Setter Property="Template">

<Setter.Value>

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

<Border Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}">

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

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

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

<Setter Property="Background" Value="#FFFF2F00"/>

<Setter Property="Foreground" Value="White"/>

</Trigger>

</Style.Triggers>

</Style>

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

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

<Setter Property="BorderBrush" Value="#FFFF8000"/>

<Setter Property="BorderThickness" Value="2"/>

<Setter Property="Template">

<Setter.Value>

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

<Border Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}">

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

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

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

<Setter Property="Background" Value="#FFFF8000"/>

</Trigger>

</Style.Triggers>

</Style>

<Style x:Key="CBox" TargetType="{x:Type ComboBox}">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="ComboBox">

<Grid>

<ToggleButton Grid.Column="2" Focusable="false" IsChecked="{Binding Path=IsDropDownOpen,Mode=TwoWay,RelativeSource={RelativeSource TemplatedParent}}" >

<ToggleButton.Template>

<ControlTemplate>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="5\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<Border x:Name="Border" Grid.ColumnSpan="2" Background="#FF32363A" BorderBrush="#FFFF8000" BorderThickness="1" />

<Border Grid.Column="0" Margin="1" Background="#FF32363A" BorderThickness="1" />

<Path x:Name="Arrow" Grid.Column="1" Fill="#FFFF7F12" HorizontalAlignment="Center" VerticalAlignment="Center" Data="M 0 0 L 4 4 L 8 0 Z"/>

</Grid>

<ControlTemplate.Triggers>

<Trigger Property="ToggleButton.IsMouseOver" Value="true">

<Setter TargetName="Border" Property="Background" Value="#FFFF7F12" />

</Trigger>

<Trigger Property="ToggleButton.IsChecked" Value="true">

<Setter TargetName="Border" Property="Background" Value="#FF32363A" />

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</ToggleButton.Template>

</ToggleButton>

<ContentPresenter Name="ContentSite" IsHitTestVisible="False" Content="{TemplateBinding SelectionBoxItem}" ContentTemplate="{TemplateBinding SelectionBoxItemTemplate}" ContentTemplateSelector="{TemplateBinding ItemTemplateSelector}" Margin="3" />

<TextBox x:Name="PART\_EditableTextBox" Visibility="Hidden" IsReadOnly="{TemplateBinding IsReadOnly}"/>

<Popup Name="Popup" Placement="Bottom" IsOpen="{TemplateBinding IsDropDownOpen}" AllowsTransparency="True" Focusable="False" PopupAnimation="Slide">

<Grid Name="DropDown" SnapsToDevicePixels="True" MinWidth="{TemplateBinding ActualWidth}" MaxHeight="{TemplateBinding MaxDropDownHeight}">

<Border x:Name="DropDownBorder" Background="#FF2E2F31" />

<ScrollViewer SnapsToDevicePixels="True">

<StackPanel IsItemsHost="True" />

</ScrollViewer>

</Grid>

</Popup>

</Grid>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

</Window.Resources>

<Grid Background="#FF32363A">

<Grid HorizontalAlignment="Center" Height="88" VerticalAlignment="Bottom" Width="1050" Background="#FF2E2F31" Margin="0,412,0,0">

<ItemsControl HorizontalAlignment="Center" VerticalAlignment="Center" Height="92" Width="1050">

<ItemsControl.ItemsPanel>

<ItemsPanelTemplate>

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

</ItemsPanelTemplate>

</ItemsControl.ItemsPanel>

<Button x:Name="RaspBtn" HorizontalAlignment="Left" Margin="0,2,0,0" VerticalAlignment="Top" Width="124" Height="90" FontSize="24" Click="RaspBtn\_Click" Style="{StaticResource Menu}">

<StackPanel Orientation="Vertical">

<Image Width="50" Source="/Resources/Images/schedule.png"/>

<Label FontSize="16" Padding="0" Foreground="White" Content="Расписание"/>

</StackPanel>

</Button>

<Button x:Name="TrenBtn" HorizontalAlignment="Left" Margin="-2,2,0,0" VerticalAlignment="Top" Width="126" Height="90" FontSize="24" Click="TrenBtn\_Click" Style="{StaticResource Menu}">

<StackPanel Orientation="Vertical">

<Image Width="35" Source="/Resources/Images/training.png"/>

<Label FontSize="16" Padding="0" Foreground="White" Content="Тренировки"/>

</StackPanel>

</Button>

<Button x:Name="KlientBtn" HorizontalAlignment="Left" Margin="-2,2,0,0" VerticalAlignment="Top" Width="124" Height="90" FontSize="24" Click="KlientBtn\_Click" Style="{StaticResource Menu}">

<StackPanel Orientation="Vertical">

<Image Width="65" Source="/Resources/Images/clients.png"/>

<Label FontSize="16" Padding="0" Foreground="White" Content="Клиенты" HorizontalAlignment="Center"/>

</StackPanel>

</Button>

<Button x:Name="TrenerBtn" HorizontalAlignment="Left" Margin="-2,2,0,0" VerticalAlignment="Top" Width="124" Height="90" FontSize="24" Click="TrenerBtn\_Click" Style="{StaticResource Menu}">

<StackPanel Orientation="Vertical">

<Image Width="35" Source="/Resources/Images/coach.png"/>

<Label FontSize="16" Padding="0" Foreground="White" Content="Тренеры" HorizontalAlignment="Center"/>

</StackPanel>

</Button>

<Button x:Name="AbonBtn" HorizontalAlignment="Left" Margin="-2,2,0,0" VerticalAlignment="Top" Width="124" Height="90" FontSize="24" Click="AbonBtn\_Click" Style="{StaticResource Menu}">

<StackPanel Orientation="Vertical">

<Image Width="60" Source="/Resources/Images/subscription.png"/>

<Label FontSize="16" Padding="0" Foreground="White" Content="Абонемент" HorizontalAlignment="Center"/>

</StackPanel>

</Button>

<Button x:Name="ExitBtn" HorizontalAlignment="Left" Margin="-2,2,0,0" VerticalAlignment="Top" Width="126" Height="90" FontSize="24" Click="ExitBtn\_Click" Style="{StaticResource Exit}">

<StackPanel Orientation="Vertical">

<Image Width="50" Source="/Resources/Images/exit.png"/>

<Label FontSize="16" Padding="0" Foreground="red" Content="Выход"/>

</StackPanel>

</Button>

</ItemsControl>

</Grid>

<Label Content="Расписание" HorizontalContentAlignment="Center" VerticalContentAlignment="Stretch" FontWeight="Bold" FontSize="48" Foreground="White" Margin="0,0,0,690"/>

<Image Name="LogoBtn" Source="/Resources/Images/logo.png" HorizontalAlignment="Left" Height="75" VerticalAlignment="Top" Width="75" Margin="10,10,0,0" MouseLeftButtonDown="LogoBtn\_MouseLeftButtonDown"/>

<ListView x:Name="list" ScrollViewer.HorizontalScrollBarVisibility="Disabled" Margin="10,151,10,204" HorizontalContentAlignment="Center" FontSize="18" Background="#FF32363A" BorderBrush="#FFFF8011" Foreground="White" SelectionMode="Single" d:ItemsSource="{d:SampleData ItemCount=5}" Width="auto">

<ListView.ItemTemplate>

<DataTemplate>

<Border BorderBrush="#FFFF8001" BorderThickness="1" Width="auto" Height="100" Padding="10" Background="#FF2E2F31">

<Grid Height="310" Background="{Binding BackColor}">

<StackPanel Grid.Row="1">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="35"/>

<ColumnDefinition Width="225"/>

<ColumnDefinition Width="225"/>

<ColumnDefinition Width="175"/>

<ColumnDefinition Width="150"/>

<ColumnDefinition Width="100"/>

<ColumnDefinition Width="75"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Row="0" Grid.Column="0">

<TextBlock Text="ID" FontWeight="Bold" HorizontalAlignment="Center"/>

<TextBlock Text="{Binding id\_raspisaniya}" HorizontalAlignment="Center"/>

</StackPanel>

<StackPanel Grid.Row="0" Grid.Column="1">

<TextBlock Text="Тренер" FontWeight="Bold" HorizontalAlignment="Center"/>

<TextBlock HorizontalAlignment="Center" TextWrapping="Wrap" TextAlignment="Center">

<Run Text="{Binding FamT, Mode=OneWay}"/>

<Run Text="{Binding ImyaT, Mode=OneWay}"/>

<Run Text="{Binding OtchT, Mode=OneWay}"/>

</TextBlock>

</StackPanel>

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

<TextBlock Text="Клиент" FontWeight="Bold" HorizontalAlignment="Center"/>

<TextBlock HorizontalAlignment="Center" TextWrapping="Wrap" TextAlignment="Center">

<Run Text="{Binding Familiya, Mode=OneWay}"/>

<Run Text="{Binding Imya, Mode=OneWay}"/>

<Run Text="{Binding Otchestvo, Mode=OneWay}"/>

</TextBlock>

</StackPanel>

<StackPanel Grid.Row="0" Grid.Column="3" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Тренировка" FontWeight="Bold" HorizontalAlignment="Center"/>

<TextBlock Text="{Binding Trenirovka}" HorizontalAlignment="Center" TextWrapping="Wrap" Height="auto" VerticalAlignment="Center"/>

</StackPanel>

<StackPanel Grid.Row="0" Grid.Column="4">

<TextBlock Text="Зал" FontWeight="Bold" HorizontalAlignment="Center" TextWrapping="Wrap"/>

<TextBlock Text="{Binding Zal}" HorizontalAlignment="Center" TextWrapping="Wrap" Height="auto" VerticalAlignment="Center"/>

</StackPanel>

<StackPanel Grid.Row="0" Grid.Column="5">

<TextBlock Text="Дата" FontWeight="Bold" HorizontalAlignment="Center"/>

<TextBlock Text="{Binding Data, StringFormat=d}" HorizontalAlignment="Center" TextWrapping="Wrap" Height="auto" VerticalAlignment="Center"/>

</StackPanel>

<StackPanel Grid.Row="0" Grid.Column="6">

<TextBlock Text="Время" FontWeight="Bold" HorizontalAlignment="Center"/>

<TextBlock Text="{Binding Vremya}" HorizontalAlignment="Center" TextWrapping="Wrap" Height="auto" VerticalAlignment="Center"/>

</StackPanel>

</Grid>

</StackPanel>

</Grid>

</Border>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<TextBox x:Name="txtSearch" HorizontalAlignment="Left" VerticalAlignment="Top" Height="40" Margin="890,106,0,0" TextWrapping="Wrap" Width="150" FontSize="24" TextChanged="txtSearch\_TextChanged" Style="{StaticResource {x:Type TextBox}}"/>

<Button x:Name="SortBtn" Content="Сортировать" HorizontalAlignment="Left" Margin="10,106,0,0" VerticalAlignment="Top" Width="175" Height="40" FontSize="24" Click="Sort\_Click" FontWeight="SemiBold" Style="{StaticResource Btn}"/>

<Button x:Name="ObnovBtn" Content="Обновить" HorizontalAlignment="Left" Margin="190,106,0,0" VerticalAlignment="Top" Width="175" Height="40" FontSize="24" Click="Refresh\_Click" FontWeight="SemiBold" Style="{StaticResource Btn}"/>

<Grid Margin="0,576,0,0" HorizontalAlignment="Center" VerticalAlignment="Top" Width="1050">

<ItemsControl BorderBrush="#FFFF8000" Background="#FF2E2F31" Margin="0,0,0,-5">

<ItemsControl Margin="0,10,0,0" Height="91">

<ItemsControl Margin="3,0,0,0" HorizontalAlignment="Center" VerticalAlignment="Center">

<ItemsControl.ItemsPanel>

<ItemsPanelTemplate>

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

</ItemsPanelTemplate>

</ItemsControl.ItemsPanel>

<ComboBox x:Name="IDTxt" FontSize="18" Style="{StaticResource CBox}" Foreground="White" Width="50" Margin="0,0,5,0"/>

<ComboBox x:Name="TrenerTxt" FontSize="18" Style="{StaticResource CBox}" Foreground="White" Width="130" Margin="0,0,5,0"/>

<ComboBox x:Name="KlientTxt" FontSize="18" Style="{StaticResource CBox}" Foreground="White" Width="130" Margin="0,0,5,0"/>

<ComboBox x:Name="TrenTxt" FontSize="18" Style="{StaticResource CBox}" Foreground="White" Width="130" Margin="0,0,5,0"/>

<ComboBox x:Name="ZalTxt" FontSize="18" Style="{StaticResource CBox}" Foreground="White" Width="130" Margin="0,0,5,0"/>

<DatePicker x:Name="DataTxt" HorizontalAlignment="Left" VerticalAlignment="Top" Height="40" Width="130" FontSize="18" PreviewTextInput="DataTxt\_PreviewTextInput" BorderBrush="#FFFF8000" Background="#FF32363A" Foreground="White" Margin="0,0,5,0">

<DatePicker.Resources>

<Style TargetType="DatePickerTextBox">

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

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

<Setter Property="Foreground" Value="White"/>

<Setter Property="FontSize" Value="18"/>

<Setter Property="Padding" Value="2"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="DatePickerTextBox">

<Border BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="1" Background="{TemplateBinding Background}">

<ScrollViewer x:Name="PART\_ContentHost" />

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

</DatePicker.Resources>

</DatePicker>

<ComboBox x:Name="DenTxt" FontSize="18" Style="{StaticResource CBox}" Foreground="White" Width="130" Margin="0,0,5,0"/>

<TextBox x:Name="VremyaTxt" HorizontalAlignment="Left" VerticalAlignment="Top" Height="40" TextWrapping="Wrap" Width="130" FontSize="18" PreviewTextInput="VremyaTxt\_PreviewTextInput" Style="{StaticResource {x:Type TextBox}}" Margin="0,0,5,0" Background="#FF32363A"/>

</ItemsControl>

<ItemsControl Height="91" HorizontalAlignment="Center">

<ItemsControl.ItemsPanel>

<ItemsPanelTemplate>

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

</ItemsPanelTemplate>

</ItemsControl.ItemsPanel>

<Button x:Name="DobavBtn" Content="Добавить" HorizontalAlignment="Left" VerticalAlignment="Top" Width="150" Height="40" FontSize="20" Click="DobavBtn\_Click" FontWeight="SemiBold" Style="{StaticResource Btn}" Margin="0,10,0,0"/>

<Button x:Name="IzmBtn" Content="Изменить" HorizontalAlignment="Left" VerticalAlignment="Top" Width="150" Height="40" FontSize="20" Click="IzmBtn\_Click" FontWeight="SemiBold" Style="{StaticResource Btn}" Margin="5,10,0,0"/>

<Button x:Name="UdalitBtn" Content="Удалить" HorizontalAlignment="Left" VerticalAlignment="Top" Width="150" Height="40" FontSize="20" Click="UdalitBtn\_Click" FontWeight="SemiBold" Style="{StaticResource Exit}" Margin="5,10,0,0"/>

</ItemsControl>

</ItemsControl>

</ItemsControl>

</Grid>

<ComboBox x:Name="DenSortTxt" FontSize="24" Style="{StaticResource CBox}" Foreground="White" Margin="735,106,165,629" SelectionChanged="SortTxt\_SelectionChanged" Width="150"/>

<ComboBox x:Name="ZalSortTxt" FontSize="24" Style="{StaticResource CBox}" Foreground="White" Margin="580,106,320,629" SelectionChanged="SortTxt\_SelectionChanged" Width="150"/>

<ComboBox x:Name="TrenSortTxt" FontSize="24" Style="{StaticResource CBox}" Foreground="White" Margin="425,106,475,629" SelectionChanged="SortTxt\_SelectionChanged" Width="150"/>

</Grid>

</Window>

using Fitnes.Resources;

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.Shapes;

namespace Fitnes

{

/// <summary>

/// Логика взаимодействия для SotrudnikRaspisanie.xaml

/// </summary>

public partial class SotrudnikRaspisanie : Window

{

user155\_dbEntities db = new user155\_dbEntities();

public SotrudnikRaspisanie()

{

InitializeComponent();

}

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

{

var Join = (from a in db.Fitnes\_Raspisanie\_Trenirovok

join b in db.Fitnes\_Klient

on a.id\_klienta equals b.id\_klienta

join c in db.Fitnes\_Trenirovka\_Klienta

on a.id\_trenirovki equals c.id\_trenirovki

join d in db.Fitnes\_Zal

on a.id\_zala equals d.id\_zala

join f in db.Fitnes\_Trener

on a.id\_trenera equals f.id\_trenera

join g in db.Fitnes\_Dni\_Nedeli

on a.id\_dni\_nedeli equals g.id\_dni\_nedeli

select new { a.id\_raspisaniya, FamT = f.Familiya, ImyaT = f.Imya, OtchT = f.Otchestvo, b.Familiya, b.Imya, b.Otchestvo, Trenirovka = c.Nazvanie, Zal = d.Nazvanie, Data = a.Data\_trenirovki, Vremya = a.Vremya\_trenirovki, Den = g.Nazvanie });

list.ItemsSource = Join.ToList();

DenSortTxt.SelectedIndex = 0;

ZalSortTxt.SelectedIndex = 0;

TrenSortTxt.SelectedIndex = 0;

txtSearch.Text = "Поиск";

txtSearch.GotFocus += RemoveText;

txtSearch.LostFocus += AddText;

DataTxt.SelectedDate = DateTime.Now;

VremyaTxt.Text = "Время";

VremyaTxt.GotFocus += RemoveText1;

VremyaTxt.LostFocus += AddText1;

var idd = db.Fitnes\_Raspisanie\_Trenirovok.Select(w => w.id\_raspisaniya).ToList();

IDTxt.Items.Add("ID");

foreach (int item in idd)

{

IDTxt.Items.Add(item);

}

IDTxt.Text = "ID";

var trener = db.Fitnes\_Trener.Select(w => w.Familiya + " " + w.Imya + " " + w.Otchestvo).ToList();

TrenerTxt.Items.Add("Тренер");

foreach (string item in trener)

{

TrenerTxt.Items.Add(item);

}

TrenerTxt.Text = "Тренер";

var klient = db.Fitnes\_Klient.Select(w => w.Familiya + " " + w.Imya + " " + w.Otchestvo).ToList();

KlientTxt.Items.Add("Клиент");

foreach (string item in klient)

{

KlientTxt.Items.Add(item);

}

KlientTxt.Text = "Клиент";

var tren = db.Fitnes\_Trenirovka\_Klienta.Select(w => w.Nazvanie).ToList();

TrenTxt.Items.Add("Тренировка");

TrenSortTxt.Items.Add("Тренировка");

foreach (string item in tren)

{

TrenTxt.Items.Add(item);

TrenSortTxt.Items.Add(item);

}

TrenTxt.Text = "Тренировка";

TrenSortTxt.Text = "Тренировка";

var zal = db.Fitnes\_Zal.Select(w => w.Nazvanie).ToList();

ZalTxt.Items.Add("Зал");

ZalSortTxt.Items.Add("Зал");

foreach (string item in zal)

{

ZalTxt.Items.Add(item);

ZalSortTxt.Items.Add(item);

}

ZalTxt.Text = "Зал";

ZalSortTxt.Text = "Зал";

var den = db.Fitnes\_Dni\_Nedeli.Select(w => w.Nazvanie).ToList();

DenTxt.Items.Add("День недели");

DenSortTxt.Items.Add("День недели");

foreach (string item in den)

{

DenTxt.Items.Add(item);

DenSortTxt.Items.Add(item);

}

DenTxt.Text = "День недели";

DenSortTxt.Text = "День недели";

}

public void RemoveText(object sender, EventArgs e)

{

if (txtSearch.Text == "Поиск")

{

txtSearch.Text = "";

}

}

public void AddText(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(txtSearch.Text))

txtSearch.Text = "Поиск";

}

public void RemoveText1(object sender, EventArgs e)

{

if (VremyaTxt.Text == "Время")

{

VremyaTxt.Text = "";

}

}

public void AddText1(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(VremyaTxt.Text))

VremyaTxt.Text = "Время";

}

private void Window\_MouseDown(object sender, MouseButtonEventArgs e)

{

if (e.LeftButton == MouseButtonState.Pressed)

DragMove();

}

private void LogoBtn\_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

SotrudnikGlavnaya a = new SotrudnikGlavnaya();

a.Show();

this.Close();

}

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

{

SotrudnikRaspisanie a = new SotrudnikRaspisanie();

a.Show();

this.Close();

}

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

{

SotrudnikTrenirovki a = new SotrudnikTrenirovki();

a.Show();

this.Close();

}

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

{

SotrudnikKlienti a = new SotrudnikKlienti();

a.Show();

this.Close();

}

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

{

SotrudnikTrenera a = new SotrudnikTrenera();

a.Show();

this.Close();

}

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

{

SotrudnikAbonementi a = new SotrudnikAbonementi();

a.Show();

this.Close();

}

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

{

MessageBoxResult result = MessageBox.Show("Вы уверены что хотите выйти из аккаунта?", "Сообщение", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

MainWindow a = new MainWindow();

a.Show();

this.Close();

}

}

public int sort = 0;

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

{

var Join = (from a in db.Fitnes\_Raspisanie\_Trenirovok

join b in db.Fitnes\_Klient

on a.id\_klienta equals b.id\_klienta

join c in db.Fitnes\_Trenirovka\_Klienta

on a.id\_trenirovki equals c.id\_trenirovki

join d in db.Fitnes\_Zal

on a.id\_zala equals d.id\_zala

join f in db.Fitnes\_Trener

on a.id\_trenera equals f.id\_trenera

join g in db.Fitnes\_Dni\_Nedeli

on a.id\_dni\_nedeli equals g.id\_dni\_nedeli

select new { a.id\_raspisaniya, FamT = f.Familiya, ImyaT = f.Imya, OtchT = f.Otchestvo, b.Familiya, b.Imya, b.Otchestvo, Trenirovka = c.Nazvanie, Zal = d.Nazvanie, Data = a.Data\_trenirovki, Vremya = a.Vremya\_trenirovki, Den = g.Nazvanie });

if (DenSortTxt.Text != "День недели" && ZalSortTxt.Text != "Зал" && TrenSortTxt.Text != "Тренировка")

{

if (sort == 0)

{

list.ItemsSource = Join.Where(w => w.Den == DenSortTxt.Text && w.Zal == ZalSortTxt.Text && w.Trenirovka == TrenSortTxt.Text).OrderByDescending(i => i.id\_raspisaniya).ToList();

sort = 1;

}

else

{

list.ItemsSource = Join.Where(w => w.Den == DenSortTxt.Text && w.Zal == ZalSortTxt.Text && w.Trenirovka == TrenSortTxt.Text).OrderBy(i => i.id\_raspisaniya).OrderBy(a => a.id\_raspisaniya).ToList();

sort = 0;

}

}

else if (DenSortTxt.Text != "День недели" && ZalSortTxt.Text != "Зал" && TrenSortTxt.Text == "Тренировка")

{

if (sort == 0)

{

list.ItemsSource = Join.Where(w => w.Den == DenSortTxt.Text && w.Zal == ZalSortTxt.Text).OrderByDescending(i => i.id\_raspisaniya).ToList();

sort = 1;

}

else

{

list.ItemsSource = Join.Where(w => w.Den == DenSortTxt.Text && w.Zal == ZalSortTxt.Text).OrderBy(i => i.id\_raspisaniya).OrderBy(a => a.id\_raspisaniya).ToList();

sort = 0;

}

}

else if (DenSortTxt.Text != "День недели" && ZalSortTxt.Text == "Зал" && TrenSortTxt.Text != "Тренировка")

{

if (sort == 0)

{

list.ItemsSource = Join.Where(w => w.Den == DenSortTxt.Text && w.Trenirovka == TrenSortTxt.Text).OrderByDescending(i => i.id\_raspisaniya).ToList();

sort = 1;

}

else

{

list.ItemsSource = Join.Where(w => w.Den == DenSortTxt.Text && w.Trenirovka == TrenSortTxt.Text).OrderBy(i => i.id\_raspisaniya).OrderBy(a => a.id\_raspisaniya).ToList();

sort = 0;

}

}

else if (DenSortTxt.Text == "День недели" && ZalSortTxt.Text != "Зал" && TrenSortTxt.Text != "Тренировка")

{

if (sort == 0)

{

list.ItemsSource = Join.Where(w => w.Zal == ZalSortTxt.Text && w.Trenirovka == TrenSortTxt.Text).OrderByDescending(i => i.id\_raspisaniya).ToList();

sort = 1;

}

else

{

list.ItemsSource = Join.Where(w => w.Zal == ZalSortTxt.Text && w.Trenirovka == TrenSortTxt.Text).OrderBy(i => i.id\_raspisaniya).OrderBy(a => a.id\_raspisaniya).ToList();

sort = 0;

}

}

else if (DenSortTxt.Text != "День недели" && ZalSortTxt.Text == "Зал" && TrenSortTxt.Text == "Тренировка")

{

if (sort == 0)

{

list.ItemsSource = Join.Where(w => w.Den == DenSortTxt.Text).OrderByDescending(i => i.id\_raspisaniya).ToList();

sort = 1;

}

else

{

list.ItemsSource = Join.Where(w => w.Den == DenSortTxt.Text).OrderBy(i => i.id\_raspisaniya).OrderBy(a => a.id\_raspisaniya).ToList();

sort = 0;

}

}

else if (DenSortTxt.Text == "День недели" && ZalSortTxt.Text != "Зал" && TrenSortTxt.Text == "Тренировка")

{

if (sort == 0)

{

list.ItemsSource = Join.Where(w => w.Zal == ZalSortTxt.Text).OrderByDescending(i => i.id\_raspisaniya).ToList();

sort = 1;

}

else

{

list.ItemsSource = Join.Where(w => w.Zal == ZalSortTxt.Text).OrderBy(i => i.id\_raspisaniya).OrderBy(a => a.id\_raspisaniya).ToList();

sort = 0;

}

}

else if (DenSortTxt.Text == "День недели" && ZalSortTxt.Text == "Зал" && TrenSortTxt.Text != "Тренировка")

{

if (sort == 0)

{

list.ItemsSource = Join.Where(w => w.Trenirovka == TrenSortTxt.Text).OrderByDescending(i => i.id\_raspisaniya).ToList();

sort = 1;

}

else

{

list.ItemsSource = Join.Where(w => w.Trenirovka == TrenSortTxt.Text).OrderBy(i => i.id\_raspisaniya).OrderBy(a => a.id\_raspisaniya).ToList();

sort = 0;

}

}

else

{

if (sort == 0)

{

list.ItemsSource = Join.OrderByDescending(i => i.id\_raspisaniya).ToList();

sort = 1;

}

else

{

list.ItemsSource = Join.OrderBy(i => i.id\_raspisaniya).ToList();

sort = 0;

}

}

}

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

{

var Join = (from a in db.Fitnes\_Raspisanie\_Trenirovok

join b in db.Fitnes\_Klient

on a.id\_klienta equals b.id\_klienta

join c in db.Fitnes\_Trenirovka\_Klienta

on a.id\_trenirovki equals c.id\_trenirovki

join d in db.Fitnes\_Zal

on a.id\_zala equals d.id\_zala

join f in db.Fitnes\_Trener

on a.id\_trenera equals f.id\_trenera

join g in db.Fitnes\_Dni\_Nedeli

on a.id\_dni\_nedeli equals g.id\_dni\_nedeli

select new { a.id\_raspisaniya, FamT = f.Familiya, ImyaT = f.Imya, OtchT = f.Otchestvo, b.Familiya, b.Imya, b.Otchestvo, Trenirovka = c.Nazvanie, Zal = d.Nazvanie, Data = a.Data\_trenirovki, Vremya = a.Vremya\_trenirovki, Den = g.Nazvanie });

list.ItemsSource = Join.ToList();

DenSortTxt.SelectedIndex = 0;

ZalSortTxt.SelectedIndex = 0;

TrenSortTxt.SelectedIndex = 0;

}

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

{

try

{

var Join = (from a in db.Fitnes\_Raspisanie\_Trenirovok

join b in db.Fitnes\_Klient

on a.id\_klienta equals b.id\_klienta

join c in db.Fitnes\_Trenirovka\_Klienta

on a.id\_trenirovki equals c.id\_trenirovki

join d in db.Fitnes\_Zal

on a.id\_zala equals d.id\_zala

join f in db.Fitnes\_Trener

on a.id\_trenera equals f.id\_trenera

join g in db.Fitnes\_Dni\_Nedeli

on a.id\_dni\_nedeli equals g.id\_dni\_nedeli

select new { a.id\_raspisaniya, FamT = f.Familiya, ImyaT = f.Imya, OtchT = f.Otchestvo, b.Familiya, b.Imya, b.Otchestvo, Trenirovka = c.Nazvanie, Zal = d.Nazvanie, Data = a.Data\_trenirovki, Vremya = a.Vremya\_trenirovki, Den = g.Nazvanie });

if (DenSortTxt.Text != "День недели" && ZalSortTxt.Text != "Зал" && TrenSortTxt.Text != "Тренировка")

{

list.ItemsSource = Join.Where(

w => w.Zal == ZalSortTxt.Text && w.Den == DenSortTxt.Text && w.Trenirovka == TrenSortTxt.Text && (w.Familiya.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Imya.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Otchestvo.ToLower().Contains(txtSearch.Text.ToLower())

|| w.FamT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.ImyaT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.OtchT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Trenirovka.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Zal.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Data.ToString().Contains(txtSearch.Text.ToLower())

|| w.Vremya.ToLower().Contains(txtSearch.Text.ToLower()))

).ToList();

}

else if (DenSortTxt.Text != "День недели" && ZalSortTxt.Text != "Зал" && TrenSortTxt.Text == "Тренировка")

{

list.ItemsSource = Join.Where(

w => w.Zal == ZalSortTxt.Text && w.Den == DenSortTxt.Text && (w.Familiya.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Imya.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Otchestvo.ToLower().Contains(txtSearch.Text.ToLower())

|| w.FamT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.ImyaT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.OtchT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Trenirovka.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Zal.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Data.ToString().Contains(txtSearch.Text.ToLower())

|| w.Vremya.ToLower().Contains(txtSearch.Text.ToLower()))

).ToList();

}

else if (DenSortTxt.Text != "День недели" && ZalSortTxt.Text == "Зал" && TrenSortTxt.Text != "Тренировка")

{

list.ItemsSource = Join.Where(

w => w.Trenirovka == TrenSortTxt.Text && w.Den == DenSortTxt.Text && (w.Familiya.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Imya.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Otchestvo.ToLower().Contains(txtSearch.Text.ToLower())

|| w.FamT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.ImyaT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.OtchT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Trenirovka.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Zal.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Data.ToString().Contains(txtSearch.Text.ToLower())

|| w.Vremya.ToLower().Contains(txtSearch.Text.ToLower()))

).ToList();

}

else if (DenSortTxt.Text == "День недели" && ZalSortTxt.Text != "Зал" && TrenSortTxt.Text != "Тренировка")

{

list.ItemsSource = Join.Where(

w => w.Trenirovka == TrenSortTxt.Text && w.Zal == ZalSortTxt.Text && (w.Familiya.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Imya.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Otchestvo.ToLower().Contains(txtSearch.Text.ToLower())

|| w.FamT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.ImyaT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.OtchT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Trenirovka.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Zal.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Data.ToString().Contains(txtSearch.Text.ToLower())

|| w.Vremya.ToLower().Contains(txtSearch.Text.ToLower()))

).ToList();

}

else if (DenSortTxt.Text != "День недели" && ZalSortTxt.Text == "Зал" && TrenSortTxt.Text == "Тренировка")

{

list.ItemsSource = Join.Where(

w => w.Den == DenSortTxt.Text && (w.Familiya.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Imya.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Otchestvo.ToLower().Contains(txtSearch.Text.ToLower())

|| w.FamT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.ImyaT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.OtchT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Trenirovka.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Zal.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Data.ToString().Contains(txtSearch.Text.ToLower())

|| w.Vremya.ToLower().Contains(txtSearch.Text.ToLower()))

).ToList();

}

else if (DenSortTxt.Text == "День недели" && ZalSortTxt.Text != "Зал" && TrenSortTxt.Text == "Тренировка")

{

list.ItemsSource = Join.Where(

w => w.Zal == ZalSortTxt.Text && (w.Familiya.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Imya.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Otchestvo.ToLower().Contains(txtSearch.Text.ToLower())

|| w.FamT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.ImyaT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.OtchT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Trenirovka.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Zal.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Data.ToString().Contains(txtSearch.Text.ToLower())

|| w.Vremya.ToLower().Contains(txtSearch.Text.ToLower()))

).ToList();

}

else if (DenSortTxt.Text == "День недели" && ZalSortTxt.Text == "Зал" && TrenSortTxt.Text != "Тренировка")

{

list.ItemsSource = Join.Where(

w => w.Trenirovka == TrenSortTxt.Text && (w.Familiya.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Imya.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Otchestvo.ToLower().Contains(txtSearch.Text.ToLower())

|| w.FamT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.ImyaT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.OtchT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Trenirovka.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Zal.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Data.ToString().Contains(txtSearch.Text.ToLower())

|| w.Vremya.ToLower().Contains(txtSearch.Text.ToLower()))

).ToList();

}

else if (DenSortTxt.Text == "День недели" && ZalSortTxt.Text == "Зал" && TrenSortTxt.Text == "Тренировка")

{

list.ItemsSource = Join.Where(

w => w.Familiya.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Imya.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Otchestvo.ToLower().Contains(txtSearch.Text.ToLower())

|| w.FamT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.ImyaT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.OtchT.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Trenirovka.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Zal.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Data.ToString().Contains(txtSearch.Text.ToLower())

|| w.Vremya.ToLower().Contains(txtSearch.Text.ToLower())

).ToList();

}

var den = DenSortTxt.SelectedValue as String;

var zal = ZalSortTxt.SelectedValue as String;

var tren = TrenSortTxt.SelectedValue as String;

if (txtSearch.Text == "")

{

if (DenSortTxt.SelectedIndex != 0 && ZalSortTxt.SelectedIndex != 0 && TrenSortTxt.SelectedIndex != 0)

{

list.ItemsSource = Join.Where(w => w.Den == den && w.Zal == zal && w.Trenirovka == tren).ToList();

}

else if (DenSortTxt.SelectedIndex != 0 && ZalSortTxt.SelectedIndex != 0 && TrenSortTxt.SelectedIndex == 0)

{

list.ItemsSource = Join.Where(w => w.Den == den && w.Zal == zal).ToList();

}

else if (DenSortTxt.SelectedIndex != 0 && ZalSortTxt.SelectedIndex == 0 && TrenSortTxt.SelectedIndex != 0)

{

list.ItemsSource = Join.Where(w => w.Den == den && w.Trenirovka == tren).ToList();

}

else if (DenSortTxt.SelectedIndex == 0 && ZalSortTxt.SelectedIndex != 0 && TrenSortTxt.SelectedIndex != 0)

{

list.ItemsSource = Join.Where(w => w.Zal == zal && w.Trenirovka == tren).ToList();

}

else if (DenSortTxt.SelectedIndex != 0 && ZalSortTxt.SelectedIndex == 0 && TrenSortTxt.SelectedIndex == 0)

{

list.ItemsSource = Join.Where(w => w.Den == den).ToList();

}

else if (DenSortTxt.SelectedIndex == 0 && ZalSortTxt.SelectedIndex != 0 && TrenSortTxt.SelectedIndex == 0)

{

list.ItemsSource = Join.Where(w => w.Zal == zal).ToList();

}

else if (DenSortTxt.SelectedIndex == 0 && ZalSortTxt.SelectedIndex == 0 && TrenSortTxt.SelectedIndex != 0)

{

list.ItemsSource = Join.Where(w => w.Trenirovka == tren).ToList();

}

else

{

list.ItemsSource = Join.ToList();

}

}

else if (txtSearch.Text == "Поиск")

{

if (DenSortTxt.SelectedIndex != 0 && ZalSortTxt.SelectedIndex != 0 && TrenSortTxt.SelectedIndex != 0)

{

list.ItemsSource = Join.Where(w => w.Den == den && w.Zal == zal && w.Trenirovka == tren).ToList();

}

else if (DenSortTxt.SelectedIndex != 0 && ZalSortTxt.SelectedIndex != 0 && TrenSortTxt.SelectedIndex == 0)

{

list.ItemsSource = Join.Where(w => w.Den == den && w.Zal == zal).ToList();

}

else if (DenSortTxt.SelectedIndex != 0 && ZalSortTxt.SelectedIndex == 0 && TrenSortTxt.SelectedIndex != 0)

{

list.ItemsSource = Join.Where(w => w.Den == den && w.Trenirovka == tren).ToList();

}

else if (DenSortTxt.SelectedIndex == 0 && ZalSortTxt.SelectedIndex != 0 && TrenSortTxt.SelectedIndex != 0)

{

list.ItemsSource = Join.Where(w => w.Zal == zal && w.Trenirovka == tren).ToList();

}

else if (DenSortTxt.SelectedIndex != 0 && ZalSortTxt.SelectedIndex == 0 && TrenSortTxt.SelectedIndex == 0)

{

list.ItemsSource = Join.Where(w => w.Den == den).ToList();

}

else if (DenSortTxt.SelectedIndex == 0 && ZalSortTxt.SelectedIndex != 0 && TrenSortTxt.SelectedIndex == 0)

{

list.ItemsSource = Join.Where(w => w.Zal == zal).ToList();

}

else if (DenSortTxt.SelectedIndex == 0 && ZalSortTxt.SelectedIndex == 0 && TrenSortTxt.SelectedIndex != 0)

{

list.ItemsSource = Join.Where(w => w.Trenirovka == tren).ToList();

}

else

{

list.ItemsSource = Join.ToList();

}

}

}

catch

{

MessageBox.Show("Произошла ошибка на сервере!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

try

{

if (TrenerTxt.Text != "Тренер" && KlientTxt.Text != "Клиент" && TrenTxt.Text != "Тренировка" && ZalTxt.Text != "Зал" && DataTxt.Text != "Дата" && VremyaTxt.Text != "Время")

{

MessageBoxResult result = MessageBox.Show("Вы уверены что хотите добавить новую запись?", "Сообщение", MessageBoxButton.YesNo, MessageBoxImage.Question);

var trener = db.Fitnes\_Trener.Where(w => w.Familiya + " " + w.Imya + " " + w.Otchestvo == TrenerTxt.Text).Select(w => w.id\_trenera).FirstOrDefault();

var fio = db.Fitnes\_Klient.Where(w => w.Familiya + " " + w.Imya + " " + w.Otchestvo == KlientTxt.Text).Select(w => w.id\_klienta).FirstOrDefault();

var tren = db.Fitnes\_Trenirovka\_Klienta.Where(w => w.Nazvanie == TrenTxt.Text).Select(w => w.id\_trenirovki).FirstOrDefault();

var zal = db.Fitnes\_Zal.Where(w => w.Nazvanie == ZalTxt.Text).Select(w => w.id\_zala).FirstOrDefault();

var den = db.Fitnes\_Dni\_Nedeli.Where(w => w.Nazvanie == DenTxt.Text).Select(w => w.id\_dni\_nedeli).FirstOrDefault();

if (result == MessageBoxResult.Yes)

{

Fitnes\_Raspisanie\_Trenirovok izm = new Fitnes\_Raspisanie\_Trenirovok();

izm.id\_klienta = fio;

izm.id\_trenirovki = tren;

izm.id\_trenera = trener;

izm.id\_zala = zal;

izm.id\_dni\_nedeli = den;

izm.Data\_trenirovki = Convert.ToDateTime(DataTxt.Text);

izm.Vremya\_trenirovki = VremyaTxt.Text;

db.Fitnes\_Raspisanie\_Trenirovok.Add(izm);

db.SaveChanges();

var Join = (from a in db.Fitnes\_Raspisanie\_Trenirovok

join b in db.Fitnes\_Klient

on a.id\_klienta equals b.id\_klienta

join c in db.Fitnes\_Trenirovka\_Klienta

on a.id\_trenirovki equals c.id\_trenirovki

join d in db.Fitnes\_Zal

on a.id\_zala equals d.id\_zala

join f in db.Fitnes\_Trener

on a.id\_trenera equals f.id\_trenera

join g in db.Fitnes\_Dni\_Nedeli

on a.id\_dni\_nedeli equals g.id\_dni\_nedeli

select new { a.id\_raspisaniya, FamT = f.Familiya, ImyaT = f.Imya, OtchT = f.Otchestvo, b.Familiya, b.Imya, b.Otchestvo, Trenirovka = c.Nazvanie, Zal = d.Nazvanie, Data = a.Data\_trenirovki, Vremya = a.Vremya\_trenirovki, Den = g.Nazvanie });

list.ItemsSource = Join.ToList();

MessageBox.Show("Запись успешно добавлена!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

IDTxt.Items.Clear();

var idd = db.Fitnes\_Raspisanie\_Trenirovok.Select(w => w.id\_raspisaniya).ToList();

IDTxt.Items.Add("ID");

foreach (int item in idd)

{

IDTxt.Items.Add(item);

}

IDTxt.Text = "ID";

}

}

else

{

MessageBox.Show("Заполните все поля!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

catch

{

MessageBox.Show("Произошла ошибка на сервере!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

try

{

if (IDTxt.Text == "ID")

{

MessageBox.Show("Выберите ID записи для изменения!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

else if (TrenerTxt.Text == "Тренер" && KlientTxt.Text != "Клиент" && TrenTxt.Text == "Тренировка" && ZalTxt.Text == "Зал" && DataTxt.Text == "Дата" && VremyaTxt.Text == "Время")

{

MessageBox.Show("Заполните хотя бы одно поле!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

else

{

MessageBoxResult result = MessageBox.Show("Сохранить новые данные?", "Сообщение", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

var num = Convert.ToInt16(IDTxt.Text);

var izm = db.Fitnes\_Raspisanie\_Trenirovok.Where(w => w.id\_raspisaniya == num).FirstOrDefault();

var trener = db.Fitnes\_Trener.Where(w => w.Familiya + " " + w.Imya + " " + w.Otchestvo == TrenerTxt.Text).Select(w => w.id\_trenera).FirstOrDefault();

var fio = db.Fitnes\_Klient.Where(w => w.Familiya + " " + w.Imya + " " + w.Otchestvo == KlientTxt.Text).Select(w => w.id\_klienta).FirstOrDefault();

var tren = db.Fitnes\_Trenirovka\_Klienta.Where(w => w.Nazvanie == TrenTxt.Text).Select(w => w.id\_trenirovki).FirstOrDefault();

var zal = db.Fitnes\_Zal.Where(w => w.Nazvanie == ZalTxt.Text).Select(w => w.id\_zala).FirstOrDefault();

var den = db.Fitnes\_Dni\_Nedeli.Where(w => w.Nazvanie == DenTxt.Text).Select(w => w.id\_dni\_nedeli).FirstOrDefault();

if (TrenerTxt.Text != "")

{

izm.id\_trenera = trener;

}

if (KlientTxt.Text != "")

{

izm.id\_klienta = fio;

}

if (TrenTxt.Text != "")

{

izm.id\_trenirovki = tren;

}

if (ZalTxt.Text != "")

{

izm.id\_zala = zal;

}

if (DenTxt.Text != "")

{

izm.id\_dni\_nedeli = den;

}

if (DataTxt.Text != "")

{

izm.Data\_trenirovki = Convert.ToDateTime(DataTxt.Text);

}

if (VremyaTxt.Text != "")

{

izm.Vremya\_trenirovki = VremyaTxt.Text;

}

db.SaveChanges();

var Join = (from a in db.Fitnes\_Raspisanie\_Trenirovok

join b in db.Fitnes\_Klient

on a.id\_klienta equals b.id\_klienta

join c in db.Fitnes\_Trenirovka\_Klienta

on a.id\_trenirovki equals c.id\_trenirovki

join d in db.Fitnes\_Zal

on a.id\_zala equals d.id\_zala

join f in db.Fitnes\_Trener

on a.id\_trenera equals f.id\_trenera

join g in db.Fitnes\_Dni\_Nedeli

on a.id\_dni\_nedeli equals g.id\_dni\_nedeli

select new { a.id\_raspisaniya, FamT = f.Familiya, ImyaT = f.Imya, OtchT = f.Otchestvo, b.Familiya, b.Imya, b.Otchestvo, Trenirovka = c.Nazvanie, Zal = d.Nazvanie, Data = a.Data\_trenirovki, Vremya = a.Vremya\_trenirovki, Den = g.Nazvanie });

list.ItemsSource = Join.ToList();

MessageBox.Show("Новые данные сохранены!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

}

catch

{

MessageBox.Show("Произошла ошибка на сервере!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

try

{

if (IDTxt.Text != "ID")

{

MessageBoxResult result = MessageBox.Show("Вы уверены что хотите удалить запись?", "Сообщение", MessageBoxButton.YesNo, MessageBoxImage.Question);

var num = Convert.ToInt16(IDTxt.Text);

var izm = db.Fitnes\_Raspisanie\_Trenirovok.Where(w => w.id\_raspisaniya == num).FirstOrDefault();

if (result == MessageBoxResult.Yes)

{

db.Fitnes\_Raspisanie\_Trenirovok.Remove(izm);

db.SaveChanges();

var Join = (from a in db.Fitnes\_Raspisanie\_Trenirovok

join b in db.Fitnes\_Klient

on a.id\_klienta equals b.id\_klienta

join c in db.Fitnes\_Trenirovka\_Klienta

on a.id\_trenirovki equals c.id\_trenirovki

join d in db.Fitnes\_Zal

on a.id\_zala equals d.id\_zala

join f in db.Fitnes\_Trener

on a.id\_trenera equals f.id\_trenera

join g in db.Fitnes\_Dni\_Nedeli

on a.id\_dni\_nedeli equals g.id\_dni\_nedeli

select new { a.id\_raspisaniya, FamT = f.Familiya, ImyaT = f.Imya, OtchT = f.Otchestvo, b.Familiya, b.Imya, b.Otchestvo, Trenirovka = c.Nazvanie, Zal = d.Nazvanie, Data = a.Data\_trenirovki, Vremya = a.Vremya\_trenirovki, Den = g.Nazvanie });

list.ItemsSource = Join.ToList();

MessageBox.Show("Запись успешно удалена!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

IDTxt.Items.Clear();

var idd = db.Fitnes\_Raspisanie\_Trenirovok.Select(w => w.id\_raspisaniya).ToList();

IDTxt.Items.Add("ID");

foreach (int item in idd)

{

IDTxt.Items.Add(item);

}

IDTxt.Text = "ID";

}

}

else

{

MessageBox.Show("Заполните поле ID!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

catch

{

MessageBox.Show("Произошла ошибка на сервере!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

if (!System.Text.RegularExpressions.Regex.IsMatch(e.Text, "^[0-9.-]"))

{

e.Handled = true;

}

}

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

{

if (!System.Text.RegularExpressions.Regex.IsMatch(e.Text, "^[0-9:-]"))

{

e.Handled = true;

}

}

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

{

var Join = (from a in db.Fitnes\_Raspisanie\_Trenirovok

join b in db.Fitnes\_Klient

on a.id\_klienta equals b.id\_klienta

join c in db.Fitnes\_Trenirovka\_Klienta

on a.id\_trenirovki equals c.id\_trenirovki

join d in db.Fitnes\_Zal

on a.id\_zala equals d.id\_zala

join f in db.Fitnes\_Trener

on a.id\_trenera equals f.id\_trenera

join g in db.Fitnes\_Dni\_Nedeli

on a.id\_dni\_nedeli equals g.id\_dni\_nedeli

select new { a.id\_raspisaniya, FamT = f.Familiya, ImyaT = f.Imya, OtchT = f.Otchestvo, b.Familiya, b.Imya, b.Otchestvo, Trenirovka = c.Nazvanie, Zal = d.Nazvanie, Data = a.Data\_trenirovki, Vremya = a.Vremya\_trenirovki, Den = g.Nazvanie });

var den = DenSortTxt.SelectedValue as String;

var zal = ZalSortTxt.SelectedValue as String;

var tren = TrenSortTxt.SelectedValue as String;

if (DenSortTxt.SelectedIndex != 0 && ZalSortTxt.SelectedIndex != 0 && TrenSortTxt.SelectedIndex != 0)

{

list.ItemsSource = Join.Where(w => w.Den == den && w.Zal == zal && w.Trenirovka == tren).ToList();

}

else if (DenSortTxt.SelectedIndex != 0 && ZalSortTxt.SelectedIndex != 0 && TrenSortTxt.SelectedIndex == 0)

{

list.ItemsSource = Join.Where(w => w.Den == den && w.Zal == zal).ToList();

}

else if (DenSortTxt.SelectedIndex != 0 && ZalSortTxt.SelectedIndex == 0 && TrenSortTxt.SelectedIndex != 0)

{

list.ItemsSource = Join.Where(w => w.Den == den && w.Trenirovka == tren).ToList();

}

else if (DenSortTxt.SelectedIndex == 0 && ZalSortTxt.SelectedIndex != 0 && TrenSortTxt.SelectedIndex != 0)

{

list.ItemsSource = Join.Where(w => w.Zal == zal && w.Trenirovka == tren).ToList();

}

else if (DenSortTxt.SelectedIndex != 0 && ZalSortTxt.SelectedIndex == 0 && TrenSortTxt.SelectedIndex == 0)

{

list.ItemsSource = Join.Where(w => w.Den == den).ToList();

}

else if (DenSortTxt.SelectedIndex == 0 && ZalSortTxt.SelectedIndex != 0 && TrenSortTxt.SelectedIndex == 0)

{

list.ItemsSource = Join.Where(w => w.Zal == zal).ToList();

}

else if (DenSortTxt.SelectedIndex == 0 && ZalSortTxt.SelectedIndex == 0 && TrenSortTxt.SelectedIndex != 0)

{

list.ItemsSource = Join.Where(w => w.Trenirovka == tren).ToList();

}

else

{

list.ItemsSource = Join.ToList();

}

}

}

}

<Window x:Class="Fitnes.SotrudnikTrenera"

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:Fitnes"

mc:Ignorable="d"

Title="Тренеры" Height="775" Width="1050" Loaded="Window\_Loaded" WindowStartupLocation="CenterScreen" ResizeMode="NoResize" WindowStyle="None" MouseDown="Window\_MouseDown">

<Window.Resources>

<Style TargetType="{x:Type TextBox}">

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

<Setter Property="BorderBrush" Value="#FFFF8000"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="Template">

<Setter.Value>

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

<Border Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}">

<ScrollViewer x:Name="PART\_ContentHost" />

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

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

<Setter Property="BorderBrush" Value="White"/>

</Trigger>

</Style.Triggers>

</Style>

<Style TargetType="{x:Type PasswordBox}">

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

<Setter Property="BorderBrush" Value="#FFFF8000"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="Template">

<Setter.Value>

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

<Border Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}">

<ScrollViewer x:Name="PART\_ContentHost" />

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

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

<Setter Property="BorderBrush" Value="White"/>

</Trigger>

</Style.Triggers>

</Style>

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

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

<Setter Property="BorderBrush" Value="#FFFF8000"/>

<Setter Property="Foreground" Value="#FFFF8000"/>

<Setter Property="BorderThickness" Value="2"/>

<Setter Property="Template">

<Setter.Value>

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

<Border Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}">

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

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

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

<Setter Property="Background" Value="#FFFF8000"/>

<Setter Property="Foreground" Value="White"/>

</Trigger>

</Style.Triggers>

</Style>

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

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

<Setter Property="BorderBrush" Value="#FFFF2F00"/>

<Setter Property="Foreground" Value="#FFFF2F00"/>

<Setter Property="BorderThickness" Value="2"/>

<Setter Property="Template">

<Setter.Value>

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

<Border Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}">

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

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

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

<Setter Property="Background" Value="#FFFF2F00"/>

<Setter Property="Foreground" Value="White"/>

</Trigger>

</Style.Triggers>

</Style>

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

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

<Setter Property="BorderBrush" Value="#FFFF8000"/>

<Setter Property="BorderThickness" Value="2"/>

<Setter Property="Template">

<Setter.Value>

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

<Border Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}">

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

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

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

<Setter Property="Background" Value="#FFFF8000"/>

</Trigger>

</Style.Triggers>

</Style>

<Style x:Key="CBox" TargetType="{x:Type ComboBox}">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="ComboBox">

<Grid>

<ToggleButton Grid.Column="2" Focusable="false" IsChecked="{Binding Path=IsDropDownOpen,Mode=TwoWay,RelativeSource={RelativeSource TemplatedParent}}" >

<ToggleButton.Template>

<ControlTemplate>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="5\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<Border x:Name="Border" Grid.ColumnSpan="2" Background="#FF32363A" BorderBrush="#FFFF8000" BorderThickness="1" />

<Border Grid.Column="0" Margin="1" Background="#FF32363A" BorderThickness="1" />

<Path x:Name="Arrow" Grid.Column="1" Fill="#FFFF7F12" HorizontalAlignment="Center" VerticalAlignment="Center" Data="M 0 0 L 4 4 L 8 0 Z"/>

</Grid>

<ControlTemplate.Triggers>

<Trigger Property="ToggleButton.IsMouseOver" Value="true">

<Setter TargetName="Border" Property="Background" Value="#FFFF7F12" />

</Trigger>

<Trigger Property="ToggleButton.IsChecked" Value="true">

<Setter TargetName="Border" Property="Background" Value="#FF32363A" />

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</ToggleButton.Template>

</ToggleButton>

<ContentPresenter Name="ContentSite" IsHitTestVisible="False" Content="{TemplateBinding SelectionBoxItem}" ContentTemplate="{TemplateBinding SelectionBoxItemTemplate}" ContentTemplateSelector="{TemplateBinding ItemTemplateSelector}" Margin="3" />

<TextBox x:Name="PART\_EditableTextBox" Visibility="Hidden" IsReadOnly="{TemplateBinding IsReadOnly}"/>

<Popup Name="Popup" Placement="Bottom" IsOpen="{TemplateBinding IsDropDownOpen}" AllowsTransparency="True" Focusable="False" PopupAnimation="Slide">

<Grid Name="DropDown" SnapsToDevicePixels="True" MinWidth="{TemplateBinding ActualWidth}" MaxHeight="{TemplateBinding MaxDropDownHeight}">

<Border x:Name="DropDownBorder" Background="#FF2E2F31" />

<ScrollViewer SnapsToDevicePixels="True">

<StackPanel IsItemsHost="True" />

</ScrollViewer>

</Grid>

</Popup>

</Grid>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

</Window.Resources>

<Grid Background="#FF32363A">

<Grid HorizontalAlignment="Center" Height="88" VerticalAlignment="Bottom" Width="1050" Background="#FF2E2F31" Margin="0,412,0,0">

<ItemsControl HorizontalAlignment="Center" VerticalAlignment="Center" Height="92" Width="800">

<ItemsControl.ItemsPanel>

<ItemsPanelTemplate>

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

</ItemsPanelTemplate>

</ItemsControl.ItemsPanel>

<Button x:Name="RaspBtn" HorizontalAlignment="Left" Margin="0,2,0,0" VerticalAlignment="Top" Width="124" Height="90" FontSize="24" Click="RaspBtn\_Click" Style="{StaticResource Menu}">

<StackPanel Orientation="Vertical">

<Image Width="50" Source="/Resources/Images/schedule.png"/>

<Label FontSize="16" Padding="0" Foreground="White" Content="Расписание"/>

</StackPanel>

</Button>

<Button x:Name="TrenBtn" HorizontalAlignment="Left" Margin="-2,2,0,0" VerticalAlignment="Top" Width="126" Height="90" FontSize="24" Click="TrenBtn\_Click" Style="{StaticResource Menu}">

<StackPanel Orientation="Vertical">

<Image Width="35" Source="/Resources/Images/training.png"/>

<Label FontSize="16" Padding="0" Foreground="White" Content="Тренировки"/>

</StackPanel>

</Button>

<Button x:Name="KlientBtn" HorizontalAlignment="Left" Margin="-2,2,0,0" VerticalAlignment="Top" Width="124" Height="90" FontSize="24" Click="KlientBtn\_Click" Style="{StaticResource Menu}">

<StackPanel Orientation="Vertical">

<Image Width="65" Source="/Resources/Images/clients.png"/>

<Label FontSize="16" Padding="0" Foreground="White" Content="Клиенты" HorizontalAlignment="Center"/>

</StackPanel>

</Button>

<Button x:Name="TrenerBtn" HorizontalAlignment="Left" Margin="-2,2,0,0" VerticalAlignment="Top" Width="124" Height="90" FontSize="24" Click="TrenerBtn\_Click" Style="{StaticResource Menu}">

<StackPanel Orientation="Vertical">

<Image Width="35" Source="/Resources/Images/coach.png"/>

<Label FontSize="16" Padding="0" Foreground="White" Content="Тренеры" HorizontalAlignment="Center"/>

</StackPanel>

</Button>

<Button x:Name="AbonBtn" HorizontalAlignment="Left" Margin="-2,2,0,0" VerticalAlignment="Top" Width="124" Height="90" FontSize="24" Click="AbonBtn\_Click" Style="{StaticResource Menu}">

<StackPanel Orientation="Vertical">

<Image Width="60" Source="/Resources/Images/subscription.png"/>

<Label FontSize="16" Padding="0" Foreground="White" Content="Абонемент" HorizontalAlignment="Center"/>

</StackPanel>

</Button>

<Button x:Name="ExitBtn" HorizontalAlignment="Left" Margin="-2,2,0,0" VerticalAlignment="Top" Width="126" Height="90" FontSize="24" Click="ExitBtn\_Click" Style="{StaticResource Exit}">

<StackPanel Orientation="Vertical">

<Image Width="50" Source="/Resources/Images/exit.png"/>

<Label FontSize="16" Padding="0" Foreground="red" Content="Выход"/>

</StackPanel>

</Button>

</ItemsControl>

</Grid>

<Label Content="Тренеры" HorizontalContentAlignment="Center" VerticalContentAlignment="Stretch" FontWeight="Bold" FontSize="48" Foreground="White" Margin="0,0,0,407"/>

<Image Name="LogoBtn" Source="/Resources/Images/logo.png" HorizontalAlignment="Left" Height="75" VerticalAlignment="Top" Width="75" Margin="10,10,0,0" MouseLeftButtonDown="LogoBtn\_MouseLeftButtonDown"/>

<ListView x:Name="list" ScrollViewer.HorizontalScrollBarVisibility="Disabled" Margin="10,151,10,204" HorizontalContentAlignment="Center" FontSize="18" Background="#FF32363A" BorderBrush="#FFFF8011" Foreground="White" SelectionMode="Single" d:ItemsSource="{d:SampleData ItemCount=5}" Width="auto">

<ListView.ItemTemplate>

<DataTemplate>

<Border BorderBrush="#FFFF8001" BorderThickness="1" Width="auto" Height="100" Padding="10" Background="#FF2E2F31">

<Grid Height="310" Background="{Binding BackColor}">

<StackPanel Grid.Row="1">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="35"/>

<ColumnDefinition Width="350"/>

<ColumnDefinition Width="200"/>

<ColumnDefinition Width="200"/>

<ColumnDefinition Width="200"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Row="0" Grid.Column="0">

<TextBlock Text="ID" FontWeight="Bold" HorizontalAlignment="Center"/>

<TextBlock Text="{Binding id\_trenera}" HorizontalAlignment="Center"/>

</StackPanel>

<StackPanel Grid.Row="0" Grid.Column="1">

<TextBlock Text="ФИО" FontWeight="Bold" HorizontalAlignment="Center"/>

<TextBlock HorizontalAlignment="Center" TextWrapping="Wrap" TextAlignment="Center">

<Run Text="{Binding Familiya, Mode=OneWay}"/>

<Run Text="{Binding Imya, Mode=OneWay}"/>

<Run Text="{Binding Otchestvo, Mode=OneWay}"/>

</TextBlock>

</StackPanel>

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

<TextBlock Text="Телефон" FontWeight="Bold" HorizontalAlignment="Center"/>

<TextBlock Text="{Binding Telefon}" HorizontalAlignment="Center" TextWrapping="Wrap" Height="auto" VerticalAlignment="Center"/>

</StackPanel>

<StackPanel Grid.Row="0" Grid.Column="3">

<TextBlock Text="Логин" FontWeight="Bold" HorizontalAlignment="Center" TextWrapping="Wrap"/>

<TextBlock Text="{Binding Login}" HorizontalAlignment="Center" TextWrapping="Wrap" Height="auto" VerticalAlignment="Center"/>

</StackPanel>

<StackPanel Grid.Row="0" Grid.Column="4">

<TextBlock Text="Пароль" FontWeight="Bold" HorizontalAlignment="Center" TextWrapping="Wrap"/>

<TextBlock Text="{Binding Password}" HorizontalAlignment="Center" TextWrapping="Wrap" Height="auto" VerticalAlignment="Center"/>

</StackPanel>

</Grid>

</StackPanel>

</Grid>

</Border>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<TextBox x:Name="txtSearch" HorizontalAlignment="Left" VerticalAlignment="Top" Height="40" Margin="790,106,0,0" TextWrapping="Wrap" Width="250" FontSize="24" TextChanged="txtSearch\_TextChanged" Style="{StaticResource {x:Type TextBox}}"/>

<Button x:Name="SortBtn" Content="Сортировать" HorizontalAlignment="Left" Margin="10,106,0,0" VerticalAlignment="Top" Width="175" Height="40" FontSize="24" Click="Sort\_Click" FontWeight="SemiBold" Style="{StaticResource Btn}"/>

<Button x:Name="ObnovBtn" Content="Обновить" HorizontalAlignment="Left" Margin="190,106,0,0" VerticalAlignment="Top" Width="175" Height="40" FontSize="24" Click="Refresh\_Click" FontWeight="SemiBold" Style="{StaticResource Btn}"/>

<Grid Margin="0,576,0,0" HorizontalAlignment="Center" VerticalAlignment="Top" Width="1050">

<ItemsControl BorderBrush="#FFFF8000" Background="#FF2E2F31" Margin="0,0,0,-5">

<ItemsControl Margin="0,10,0,0" Height="91" Width="800">

<ItemsControl Margin="3,0,0,0" HorizontalAlignment="Center" VerticalAlignment="Center">

<ItemsControl.ItemsPanel>

<ItemsPanelTemplate>

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

</ItemsPanelTemplate>

</ItemsControl.ItemsPanel>

<ComboBox x:Name="IDTxt" FontSize="18" Style="{StaticResource CBox}" Foreground="White" Width="50" Margin="0,0,5,0"/>

<TextBox x:Name="FamTxt" HorizontalAlignment="Left" VerticalAlignment="Top" Height="40" TextWrapping="Wrap" Width="110" FontSize="18" PreviewTextInput="FamTxt\_PreviewTextInput" Style="{StaticResource {x:Type TextBox}}" Margin="0,0,5,0" Background="#FF32363A"/>

<TextBox x:Name="ImyaTxt" HorizontalAlignment="Left" VerticalAlignment="Top" Height="40" TextWrapping="Wrap" Width="110" FontSize="18" PreviewTextInput="ImyaTxt\_PreviewTextInput" Style="{StaticResource {x:Type TextBox}}" Margin="0,0,5,0" Background="#FF32363A"/>

<TextBox x:Name="OtchTxt" HorizontalAlignment="Left" VerticalAlignment="Top" Height="40" TextWrapping="Wrap" Width="110" FontSize="18" PreviewTextInput="OtchTxt\_PreviewTextInput" Style="{StaticResource {x:Type TextBox}}" Margin="0,0,5,0" Background="#FF32363A"/>

<TextBox x:Name="TelTxt" HorizontalAlignment="Left" VerticalAlignment="Top" Height="40" TextWrapping="Wrap" Width="110" FontSize="18" PreviewTextInput="TelTxt\_PreviewTextInput" Style="{StaticResource {x:Type TextBox}}" Margin="0,0,5,0" Background="#FF32363A"/>

<TextBox x:Name="LoginTxt" HorizontalAlignment="Left" VerticalAlignment="Top" Height="40" TextWrapping="Wrap" Width="110" FontSize="18" PreviewTextInput="LoginTxt\_PreviewTextInput" Style="{StaticResource {x:Type TextBox}}" Margin="0,0,5,0" Background="#FF32363A"/>

<TextBox x:Name="PassTxt" HorizontalAlignment="Left" VerticalAlignment="Top" Height="40" TextWrapping="Wrap" Width="110" FontSize="18" PreviewTextInput="PassTxt\_PreviewTextInput" Style="{StaticResource {x:Type TextBox}}" Margin="0,0,5,0" Background="#FF32363A"/>

</ItemsControl>

<ItemsControl Height="91" HorizontalAlignment="Center">

<ItemsControl.ItemsPanel>

<ItemsPanelTemplate>

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

</ItemsPanelTemplate>

</ItemsControl.ItemsPanel>

<Button x:Name="DobavBtn" Content="Добавить" HorizontalAlignment="Left" VerticalAlignment="Top" Width="150" Height="40" FontSize="20" Click="DobavBtn\_Click" FontWeight="SemiBold" Style="{StaticResource Btn}" Margin="0,10,0,0"/>

<Button x:Name="IzmBtn" Content="Изменить" HorizontalAlignment="Left" VerticalAlignment="Top" Width="150" Height="40" FontSize="20" Click="IzmBtn\_Click" FontWeight="SemiBold" Style="{StaticResource Btn}" Margin="5,10,0,0"/>

<Button x:Name="UdalitBtn" Content="Удалить" HorizontalAlignment="Left" VerticalAlignment="Top" Width="150" Height="40" FontSize="20" Click="UdalitBtn\_Click" FontWeight="SemiBold" Style="{StaticResource Exit}" Margin="5,10,0,0"/>

</ItemsControl>

</ItemsControl>

</ItemsControl>

</Grid>

</Grid>

</Window>

using Fitnes.Resources;

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.Shapes;

namespace Fitnes

{

/// <summary>

/// Логика взаимодействия для SotrudnikTrenera.xaml

/// </summary>

public partial class SotrudnikTrenera : Window

{

user155\_dbEntities db = new user155\_dbEntities();

public SotrudnikTrenera()

{

InitializeComponent();

}

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

{

list.ItemsSource = db.Fitnes\_Trener.ToList();

txtSearch.Text = "Поиск";

txtSearch.GotFocus += RemoveText;

txtSearch.LostFocus += AddText;

FamTxt.Text = "Фамилия";

FamTxt.GotFocus += RemoveText1;

FamTxt.LostFocus += AddText1;

ImyaTxt.Text = "Имя";

ImyaTxt.GotFocus += RemoveText2;

ImyaTxt.LostFocus += AddText2;

OtchTxt.Text = "Отчество";

OtchTxt.GotFocus += RemoveText3;

OtchTxt.LostFocus += AddText3;

TelTxt.Text = "Телефон";

TelTxt.GotFocus += RemoveText4;

TelTxt.LostFocus += AddText4;

LoginTxt.Text = "Логин";

LoginTxt.GotFocus += RemoveText5;

LoginTxt.LostFocus += AddText5;

PassTxt.Text = "Пароль";

PassTxt.GotFocus += RemoveText6;

PassTxt.LostFocus += AddText6;

var idd = db.Fitnes\_Trener.Select(w => w.id\_trenera).ToList();

IDTxt.Items.Add("ID");

foreach (int item in idd)

{

IDTxt.Items.Add(item);

}

IDTxt.Text = "ID";

}

public void RemoveText(object sender, EventArgs e)

{

if (txtSearch.Text == "Поиск")

{

txtSearch.Text = "";

}

}

public void AddText(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(txtSearch.Text))

txtSearch.Text = "Поиск";

}

public void RemoveText1(object sender, EventArgs e)

{

if (FamTxt.Text == "Фамилия")

{

FamTxt.Text = "";

}

}

public void AddText1(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(FamTxt.Text))

FamTxt.Text = "Фамилия";

}

public void RemoveText2(object sender, EventArgs e)

{

if (ImyaTxt.Text == "Имя")

{

ImyaTxt.Text = "";

}

}

public void AddText2(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(ImyaTxt.Text))

ImyaTxt.Text = "Имя";

}

public void RemoveText3(object sender, EventArgs e)

{

if (OtchTxt.Text == "Отчество")

{

OtchTxt.Text = "";

}

}

public void AddText3(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(OtchTxt.Text))

OtchTxt.Text = "Отчество";

}

public void RemoveText4(object sender, EventArgs e)

{

if (TelTxt.Text == "Телефон")

{

TelTxt.Text = "";

}

}

public void AddText4(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(TelTxt.Text))

TelTxt.Text = "Телефон";

}

public void RemoveText5(object sender, EventArgs e)

{

if (LoginTxt.Text == "Логин")

{

LoginTxt.Text = "";

}

}

public void AddText5(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(LoginTxt.Text))

LoginTxt.Text = "Логин";

}

public void RemoveText6(object sender, EventArgs e)

{

if (PassTxt.Text == "Пароль")

{

PassTxt.Text = "";

}

}

public void AddText6(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(PassTxt.Text))

PassTxt.Text = "Пароль";

}

private void Window\_MouseDown(object sender, MouseButtonEventArgs e)

{

if (e.LeftButton == MouseButtonState.Pressed)

DragMove();

}

private void LogoBtn\_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

SotrudnikGlavnaya a = new SotrudnikGlavnaya();

a.Show();

this.Close();

}

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

{

SotrudnikRaspisanie a = new SotrudnikRaspisanie();

a.Show();

this.Close();

}

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

{

SotrudnikTrenirovki a = new SotrudnikTrenirovki();

a.Show();

this.Close();

}

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

{

SotrudnikKlienti a = new SotrudnikKlienti();

a.Show();

this.Close();

}

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

{

SotrudnikTrenera a = new SotrudnikTrenera();

a.Show();

this.Close();

}

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

{

SotrudnikAbonementi a = new SotrudnikAbonementi();

a.Show();

this.Close();

}

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

{

MessageBoxResult result = MessageBox.Show("Вы уверены что хотите выйти из аккаунта?", "Сообщение", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

MainWindow a = new MainWindow();

a.Show();

this.Close();

}

}

public int sort = 0;

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

{

if (sort == 0)

{

list.ItemsSource = db.Fitnes\_Trener.OrderByDescending(i => i.Familiya).ToList();

sort = 1;

}

else

{

list.ItemsSource = db.Fitnes\_Trener.OrderBy(a => a.Familiya).ToList();

sort = 0;

}

}

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

{

list.ItemsSource = db.Fitnes\_Trener.ToList();

}

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

{

try

{

list.ItemsSource = db.Fitnes\_Trener.Where(

w => w.Familiya.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Imya.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Otchestvo.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Telefon.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Login.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Password.ToLower().Contains(txtSearch.Text.ToLower())

).ToList();

if (txtSearch.Text == "")

{

list.ItemsSource = db.Fitnes\_Trener.ToList();

}

else if (txtSearch.Text == "Поиск")

{

list.ItemsSource = db.Fitnes\_Trener.ToList();

}

}

catch

{

MessageBox.Show("Произошла ошибка на сервере!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

try

{

if (FamTxt.Text != "Фамилия" && ImyaTxt.Text != "Имя" && OtchTxt.Text != "Отчество" && TelTxt.Text != "Телефон" && LoginTxt.Text != "Логин" && PassTxt.Text != "Пароль")

{

MessageBoxResult result = MessageBox.Show("Вы уверены что хотите добавить новую запись?", "Сообщение", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

Fitnes\_Trener izm = new Fitnes\_Trener();

izm.Familiya = FamTxt.Text;

izm.Imya = ImyaTxt.Text;

izm.Otchestvo = OtchTxt.Text;

izm.Telefon = TelTxt.Text;

izm.Login = LoginTxt.Text;

izm.Password = PassTxt.Text;

db.Fitnes\_Trener.Add(izm);

db.SaveChanges();

list.ItemsSource = db.Fitnes\_Trener.ToList();

MessageBox.Show("Запись успешно добавлена!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

IDTxt.Items.Clear();

var idd = db.Fitnes\_Trener.Select(w => w.id\_trenera).ToList();

IDTxt.Items.Add("ID");

foreach (int item in idd)

{

IDTxt.Items.Add(item);

}

IDTxt.Text = "ID";

}

}

else

{

MessageBox.Show("Заполните все поля!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

catch

{

MessageBox.Show("Произошла ошибка на сервере!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

try

{

if (IDTxt.Text == "ID")

{

MessageBox.Show("Выберите ID записи для изменения!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

else if (FamTxt.Text == "Фамилия" && ImyaTxt.Text == "Имя" && OtchTxt.Text == "Отчество" && TelTxt.Text == "Телефон" && LoginTxt.Text != "Логин" && PassTxt.Text != "Пароль")

{

MessageBox.Show("Заполните хотя бы одно поле!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

else

{

MessageBoxResult result = MessageBox.Show("Сохранить новые данные?", "Сообщение", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

var num = Convert.ToInt16(IDTxt.Text);

var izm = db.Fitnes\_Trener.Where(w => w.id\_trenera == num).FirstOrDefault();

if (FamTxt.Text != "Фамилия")

{

izm.Familiya = FamTxt.Text;

}

if (ImyaTxt.Text != "Имя")

{

izm.Imya = ImyaTxt.Text;

}

if (OtchTxt.Text != "Отчество")

{

izm.Otchestvo = OtchTxt.Text;

}

if (TelTxt.Text != "Телефон")

{

izm.Telefon = TelTxt.Text;

}

if (LoginTxt.Text != "Логин")

{

izm.Login = LoginTxt.Text;

}

if (PassTxt.Text != "Пароль")

{

izm.Password = PassTxt.Text;

}

db.SaveChanges();

list.ItemsSource = db.Fitnes\_Trener.ToList();

MessageBox.Show("Новые данные сохранены!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

}

catch

{

MessageBox.Show("Произошла ошибка на сервере!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

try

{

if (IDTxt.Text != "ID")

{

MessageBoxResult result = MessageBox.Show("Вы уверены что хотите удалить запись?", "Сообщение", MessageBoxButton.YesNo, MessageBoxImage.Question);

var num = Convert.ToInt16(IDTxt.Text);

var izm = db.Fitnes\_Trener.Where(w => w.id\_trenera == num).FirstOrDefault();

if (result == MessageBoxResult.Yes)

{

db.Fitnes\_Trener.Remove(izm);

db.SaveChanges();

list.ItemsSource = db.Fitnes\_Trener.ToList();

MessageBox.Show("Запись успешно удалена!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

IDTxt.Items.Clear();

var idd = db.Fitnes\_Trener.Select(w => w.id\_trenera).ToList();

IDTxt.Items.Add("ID");

foreach (int item in idd)

{

IDTxt.Items.Add(item);

}

IDTxt.Text = "ID";

}

}

else

{

MessageBox.Show("Заполните поле ID!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

catch

{

MessageBox.Show("Произошла ошибка на сервере!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

if (!System.Text.RegularExpressions.Regex.IsMatch(e.Text, "^[а-яА-Я]"))

{

e.Handled = true;

}

}

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

{

if (!System.Text.RegularExpressions.Regex.IsMatch(e.Text, "^[а-яА-Я]"))

{

e.Handled = true;

}

}

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

{

if (!System.Text.RegularExpressions.Regex.IsMatch(e.Text, "^[а-яА-Я]"))

{

e.Handled = true;

}

}

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

{

if (!System.Text.RegularExpressions.Regex.IsMatch(e.Text, "^[0-9+()-]"))

{

e.Handled = true;

}

}

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

{

if (!System.Text.RegularExpressions.Regex.IsMatch(e.Text, "^[a-zA-Z0-9.,@!;:]"))

{

e.Handled = true;

}

}

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

{

if (!System.Text.RegularExpressions.Regex.IsMatch(e.Text, "^[a-zA-Z0-9.,@!;:]"))

{

e.Handled = true;

}

}

}

}

<Window x:Class="Fitnes.SotrudnikTrenirovki"

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:Fitnes"

mc:Ignorable="d"

Title="Тренировки" Height="750" Width="1050" Loaded="Window\_Loaded" WindowStartupLocation="CenterScreen" ResizeMode="NoResize" WindowStyle="None" MouseDown="Window\_MouseDown">

<Window.Resources>

<Style TargetType="{x:Type TextBox}">

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

<Setter Property="BorderBrush" Value="#FFFF8000"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="Template">

<Setter.Value>

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

<Border Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}">

<ScrollViewer x:Name="PART\_ContentHost" />

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

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

<Setter Property="BorderBrush" Value="White"/>

</Trigger>

</Style.Triggers>

</Style>

<Style TargetType="{x:Type PasswordBox}">

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

<Setter Property="BorderBrush" Value="#FFFF8000"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="Template">

<Setter.Value>

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

<Border Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}">

<ScrollViewer x:Name="PART\_ContentHost" />

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

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

<Setter Property="BorderBrush" Value="White"/>

</Trigger>

</Style.Triggers>

</Style>

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

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

<Setter Property="BorderBrush" Value="#FFFF8000"/>

<Setter Property="Foreground" Value="#FFFF8000"/>

<Setter Property="BorderThickness" Value="2"/>

<Setter Property="Template">

<Setter.Value>

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

<Border Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}">

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

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

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

<Setter Property="Background" Value="#FFFF8000"/>

<Setter Property="Foreground" Value="White"/>

</Trigger>

</Style.Triggers>

</Style>

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

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

<Setter Property="BorderBrush" Value="#FFFF2F00"/>

<Setter Property="Foreground" Value="#FFFF2F00"/>

<Setter Property="BorderThickness" Value="2"/>

<Setter Property="Template">

<Setter.Value>

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

<Border Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}">

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

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

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

<Setter Property="Background" Value="#FFFF2F00"/>

<Setter Property="Foreground" Value="White"/>

</Trigger>

</Style.Triggers>

</Style>

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

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

<Setter Property="BorderBrush" Value="#FFFF8000"/>

<Setter Property="BorderThickness" Value="2"/>

<Setter Property="Template">

<Setter.Value>

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

<Border Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}">

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

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

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

<Setter Property="Background" Value="#FFFF8000"/>

</Trigger>

</Style.Triggers>

</Style>

<Style x:Key="CBox" TargetType="{x:Type ComboBox}">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="ComboBox">

<Grid>

<ToggleButton Grid.Column="2" Focusable="false" IsChecked="{Binding Path=IsDropDownOpen,Mode=TwoWay,RelativeSource={RelativeSource TemplatedParent}}" >

<ToggleButton.Template>

<ControlTemplate>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="5\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<Border x:Name="Border" Grid.ColumnSpan="2" Background="#FF2E2F31" BorderBrush="#FFFF8000" BorderThickness="1" />

<Border Grid.Column="0" Margin="1" Background="#FF2E2F31" BorderThickness="1" />

<Path x:Name="Arrow" Grid.Column="1" Fill="#FFFF7F12" HorizontalAlignment="Center" VerticalAlignment="Center" Data="M 0 0 L 4 4 L 8 0 Z"/>

</Grid>

<ControlTemplate.Triggers>

<Trigger Property="ToggleButton.IsMouseOver" Value="true">

<Setter TargetName="Border" Property="Background" Value="#FFFF7F12" />

</Trigger>

<Trigger Property="ToggleButton.IsChecked" Value="true">

<Setter TargetName="Border" Property="Background" Value="#FF2E2F31" />

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</ToggleButton.Template>

</ToggleButton>

<ContentPresenter Name="ContentSite" IsHitTestVisible="False" Content="{TemplateBinding SelectionBoxItem}" ContentTemplate="{TemplateBinding SelectionBoxItemTemplate}" ContentTemplateSelector="{TemplateBinding ItemTemplateSelector}" Margin="3" />

<TextBox x:Name="PART\_EditableTextBox" Visibility="Hidden" IsReadOnly="{TemplateBinding IsReadOnly}"/>

<Popup Name="Popup" Placement="Bottom" IsOpen="{TemplateBinding IsDropDownOpen}" AllowsTransparency="True" Focusable="False" PopupAnimation="Slide">

<Grid Name="DropDown" SnapsToDevicePixels="True" MinWidth="{TemplateBinding ActualWidth}" MaxHeight="{TemplateBinding MaxDropDownHeight}">

<Border x:Name="DropDownBorder" Background="#FF2E2F31" />

<ScrollViewer SnapsToDevicePixels="True">

<StackPanel IsItemsHost="True" />

</ScrollViewer>

</Grid>

</Popup>

</Grid>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

</Window.Resources>

<Grid Background="#FF32363A">

<Grid HorizontalAlignment="Center" Height="88" VerticalAlignment="Bottom" Width="1050" Background="#FF2E2F31" Margin="0,412,0,0">

<ItemsControl HorizontalAlignment="Center" VerticalAlignment="Center" Height="92" Width="800">

<ItemsControl.ItemsPanel>

<ItemsPanelTemplate>

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

</ItemsPanelTemplate>

</ItemsControl.ItemsPanel>

<Button x:Name="RaspBtn" HorizontalAlignment="Left" Margin="0,2,0,0" VerticalAlignment="Top" Width="124" Height="90" FontSize="24" Click="RaspBtn\_Click" Style="{StaticResource Menu}">

<StackPanel Orientation="Vertical">

<Image Width="50" Source="/Resources/Images/schedule.png"/>

<Label FontSize="16" Padding="0" Foreground="White" Content="Расписание"/>

</StackPanel>

</Button>

<Button x:Name="TrenBtn" HorizontalAlignment="Left" Margin="-2,2,0,0" VerticalAlignment="Top" Width="126" Height="90" FontSize="24" Click="TrenBtn\_Click" Style="{StaticResource Menu}">

<StackPanel Orientation="Vertical">

<Image Width="35" Source="/Resources/Images/training.png"/>

<Label FontSize="16" Padding="0" Foreground="White" Content="Тренировки"/>

</StackPanel>

</Button>

<Button x:Name="KlientBtn" HorizontalAlignment="Left" Margin="-2,2,0,0" VerticalAlignment="Top" Width="124" Height="90" FontSize="24" Click="KlientBtn\_Click" Style="{StaticResource Menu}">

<StackPanel Orientation="Vertical">

<Image Width="65" Source="/Resources/Images/clients.png"/>

<Label FontSize="16" Padding="0" Foreground="White" Content="Клиенты" HorizontalAlignment="Center"/>

</StackPanel>

</Button>

<Button x:Name="TrenerBtn" HorizontalAlignment="Left" Margin="-2,2,0,0" VerticalAlignment="Top" Width="124" Height="90" FontSize="24" Click="TrenerBtn\_Click" Style="{StaticResource Menu}">

<StackPanel Orientation="Vertical">

<Image Width="35" Source="/Resources/Images/coach.png"/>

<Label FontSize="16" Padding="0" Foreground="White" Content="Тренеры" HorizontalAlignment="Center"/>

</StackPanel>

</Button>

<Button x:Name="AbonBtn" HorizontalAlignment="Left" Margin="-2,2,0,0" VerticalAlignment="Top" Width="124" Height="90" FontSize="24" Click="AbonBtn\_Click" Style="{StaticResource Menu}">

<StackPanel Orientation="Vertical">

<Image Width="60" Source="/Resources/Images/subscription.png"/>

<Label FontSize="16" Padding="0" Foreground="White" Content="Абонемент" HorizontalAlignment="Center"/>

</StackPanel>

</Button>

<Button x:Name="ExitBtn" HorizontalAlignment="Left" Margin="-2,2,0,0" VerticalAlignment="Top" Width="126" Height="90" FontSize="24" Click="ExitBtn\_Click" Style="{StaticResource Exit}">

<StackPanel Orientation="Vertical">

<Image Width="50" Source="/Resources/Images/exit.png"/>

<Label FontSize="16" Padding="0" Foreground="red" Content="Выход"/>

</StackPanel>

</Button>

</ItemsControl>

</Grid>

<Label Content="Тренировки" HorizontalContentAlignment="Center" VerticalContentAlignment="Stretch" FontWeight="Bold" FontSize="48" Foreground="White" Margin="0,0,0,649"/>

<Image Name="LogoBtn" Source="/Resources/Images/logo.png" HorizontalAlignment="Left" Height="75" VerticalAlignment="Top" Width="75" Margin="10,10,0,0" MouseLeftButtonDown="LogoBtn\_MouseLeftButtonDown"/>

<ListView x:Name="list" ScrollViewer.HorizontalScrollBarVisibility="Disabled" Margin="10,151,260,108" HorizontalContentAlignment="Center" FontSize="18" Background="#FF32363A" BorderBrush="#FFFF8011" Foreground="White" SelectionMode="Single" d:ItemsSource="{d:SampleData ItemCount=5}">

<ListView.ItemTemplate>

<DataTemplate>

<Border BorderBrush="#FFFF8001" BorderThickness="1" Width="740" Height="115" Padding="10" Background="#FF2E2F31">

<Grid Height="310" Background="{Binding BackColor}">

<StackPanel Grid.Row="1">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="50"/>

<ColumnDefinition Width="200"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Row="0" Grid.Column="0">

<TextBlock Text="ID" FontWeight="Bold" HorizontalAlignment="Center"/>

<TextBlock Text="{Binding id\_trenirovki}" HorizontalAlignment="Center"/>

</StackPanel>

<StackPanel Grid.Row="0" Grid.Column="1">

<TextBlock Text="Название" FontWeight="Bold" HorizontalAlignment="Center"/>

<TextBlock Text="{Binding Nazvanie}" HorizontalAlignment="Center"/>

</StackPanel>

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

<TextBlock Text="Описание" FontWeight="Bold" HorizontalAlignment="Center"/>

<TextBlock Text="{Binding Opisanie}" HorizontalAlignment="Center" TextWrapping="Wrap"/>

</StackPanel>

</Grid>

</StackPanel>

</Grid>

</Border>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<TextBox x:Name="txtSearch" HorizontalAlignment="Left" VerticalAlignment="Top" Height="40" Margin="540,106,0,0" TextWrapping="Wrap" Width="250" FontSize="24" TextChanged="txtSearch\_TextChanged" Style="{StaticResource {x:Type TextBox}}"/>

<Button x:Name="SortBtn" Content="Сортировать" HorizontalAlignment="Left" Margin="10,106,0,0" VerticalAlignment="Top" Width="175" Height="40" FontSize="24" Click="Sort\_Click" FontWeight="SemiBold" Style="{StaticResource Btn}"/>

<Button x:Name="ObnovBtn" Content="Обновить" HorizontalAlignment="Left" Margin="190,106,0,0" VerticalAlignment="Top" Width="175" Height="40" FontSize="24" Click="Refresh\_Click" FontWeight="SemiBold" Style="{StaticResource Btn}"/>

<Grid Margin="795,106,10,108">

<ScrollViewer BorderBrush="#FFFF8000" Background="#FF2E2F31" VerticalScrollBarVisibility="Visible">

<ItemsControl Margin="0,10,0,0" Height="337" >

<ComboBox x:Name="IDTxt" Height="40" Width="175" FontSize="18" Style="{StaticResource CBox}" Margin="0,0,0,10" Foreground="White"/>

<TextBox x:Name="NazvTxt" Height="40" TextWrapping="Wrap" Width="175" FontSize="18" PreviewTextInput="NazvTxt\_PreviewTextInput" Style="{StaticResource {x:Type TextBox}}" Margin="0,0,0,10"/>

<TextBox x:Name="OpisTxt" Height="75" TextWrapping="Wrap" Width="175" FontSize="18" PreviewTextInput="OpisTxt\_PreviewTextInput" Style="{StaticResource {x:Type TextBox}}" Margin="0,0,0,10"/>

<Button x:Name="DobavBtn" Content="Добавить" HorizontalAlignment="Center" VerticalAlignment="Top" Width="175" Height="40" FontSize="20" Click="DobavBtn\_Click" FontWeight="SemiBold" Style="{StaticResource Btn}" Margin="0,0,0,10"/>

<Button x:Name="IzmBtn" Content="Изменить" HorizontalAlignment="Center" VerticalAlignment="Top" Width="175" Height="40" FontSize="20" Click="IzmBtn\_Click" FontWeight="SemiBold" Style="{StaticResource Btn}" Margin="0,0,0,10"/>

<Button x:Name="UdalitBtn" Content="Удалить" HorizontalAlignment="Center" VerticalAlignment="Top" Width="175" Height="40" FontSize="20" Click="UdalitBtn\_Click" FontWeight="SemiBold" Style="{StaticResource Exit}"/>

</ItemsControl>

</ScrollViewer>

</Grid>

</Grid>

</Window>

using Fitnes.Resources;

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.Shapes;

namespace Fitnes

{

/// <summary>

/// Логика взаимодействия для SotrudnikTrenirovki.xaml

/// </summary>

public partial class SotrudnikTrenirovki : Window

{

user155\_dbEntities db = new user155\_dbEntities();

public SotrudnikTrenirovki()

{

InitializeComponent();

}

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

{

list.ItemsSource = db.Fitnes\_Trenirovka\_Klienta.ToList();

txtSearch.Text = "Поиск";

txtSearch.GotFocus += RemoveText;

txtSearch.LostFocus += AddText;

NazvTxt.Text = "Название";

NazvTxt.GotFocus += RemoveText1;

NazvTxt.LostFocus += AddText1;

OpisTxt.Text = "Описание";

OpisTxt.GotFocus += RemoveText2;

OpisTxt.LostFocus += AddText2;

var idd = db.Fitnes\_Trenirovka\_Klienta.Select(w => w.id\_trenirovki).ToList();

IDTxt.Items.Add("ID");

foreach (int item in idd)

{

IDTxt.Items.Add(item);

}

IDTxt.Text = "ID";

}

public void RemoveText(object sender, EventArgs e)

{

if (txtSearch.Text == "Поиск")

{

txtSearch.Text = "";

}

}

public void AddText(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(txtSearch.Text))

txtSearch.Text = "Поиск";

}

public void RemoveText1(object sender, EventArgs e)

{

if (NazvTxt.Text == "Название")

{

NazvTxt.Text = "";

}

}

public void AddText1(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(NazvTxt.Text))

NazvTxt.Text = "Название";

}

public void RemoveText2(object sender, EventArgs e)

{

if (OpisTxt.Text == "Описание")

{

OpisTxt.Text = "";

}

}

public void AddText2(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(OpisTxt.Text))

OpisTxt.Text = "Описание";

}

private void Window\_MouseDown(object sender, MouseButtonEventArgs e)

{

if (e.LeftButton == MouseButtonState.Pressed)

DragMove();

}

private void LogoBtn\_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

SotrudnikGlavnaya a = new SotrudnikGlavnaya();

a.Show();

this.Close();

}

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

{

SotrudnikRaspisanie a = new SotrudnikRaspisanie();

a.Show();

this.Close();

}

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

{

SotrudnikTrenirovki a = new SotrudnikTrenirovki();

a.Show();

this.Close();

}

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

{

SotrudnikKlienti a = new SotrudnikKlienti();

a.Show();

this.Close();

}

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

{

SotrudnikTrenera a = new SotrudnikTrenera();

a.Show();

this.Close();

}

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

{

SotrudnikAbonementi a = new SotrudnikAbonementi();

a.Show();

this.Close();

}

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

{

MessageBoxResult result = MessageBox.Show("Вы уверены что хотите выйти из аккаунта?", "Сообщение", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

MainWindow a = new MainWindow();

a.Show();

this.Close();

}

}

public int sort = 0;

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

{

if (sort == 0)

{

list.ItemsSource = db.Fitnes\_Trenirovka\_Klienta.OrderByDescending(i => i.Nazvanie).ToList();

sort = 1;

}

else

{

list.ItemsSource = db.Fitnes\_Trenirovka\_Klienta.OrderBy(a => a.Nazvanie).ToList();

sort = 0;

}

}

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

{

list.ItemsSource = db.Fitnes\_Trenirovka\_Klienta.ToList();

}

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

{

try

{

list.ItemsSource = db.Fitnes\_Trenirovka\_Klienta.ToList().Where(

w => w.Nazvanie.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Opisanie.ToLower().Contains(txtSearch.Text.ToLower())

).ToList();

if (txtSearch.Text == "")

{

list.ItemsSource = db.Fitnes\_Trenirovka\_Klienta.ToList();

}

else if (txtSearch.Text == "Поиск")

{

list.ItemsSource = db.Fitnes\_Trenirovka\_Klienta.ToList();

}

}

catch

{

MessageBox.Show("Произошла ошибка на сервере!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

try

{

if (NazvTxt.Text != "Название" && OpisTxt.Text != "Описание")

{

MessageBoxResult result = MessageBox.Show("Вы уверены что хотите добавить новую запись?", "Сообщение", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

Fitnes\_Trenirovka\_Klienta izm = new Fitnes\_Trenirovka\_Klienta();

izm.Nazvanie = NazvTxt.Text;

izm.Opisanie = OpisTxt.Text;

db.Fitnes\_Trenirovka\_Klienta.Add(izm);

db.SaveChanges();

list.ItemsSource = db.Fitnes\_Trenirovka\_Klienta.ToList();

MessageBox.Show("Запись успешно добавлена!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

IDTxt.Items.Clear();

var idd = db.Fitnes\_Trenirovka\_Klienta.Select(w => w.id\_trenirovki).ToList();

IDTxt.Items.Add("ID");

foreach (int item in idd)

{

IDTxt.Items.Add(item);

}

IDTxt.Text = "ID";

}

}

else

{

MessageBox.Show("Заполните все поля!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

catch

{

MessageBox.Show("Произошла ошибка на сервере!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

try

{

if (IDTxt.Text == "ID")

{

MessageBox.Show("Выберите ID записи для изменения!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

else if (NazvTxt.Text == "Название" && OpisTxt.Text == "Описание")

{

MessageBox.Show("Заполните хотя бы одно поле!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

else

{

MessageBoxResult result = MessageBox.Show("Сохранить новые данные?", "Сообщение", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

var num = Convert.ToInt16(IDTxt.Text);

var izm = db.Fitnes\_Trenirovka\_Klienta.Where(w => w.id\_trenirovki == num).FirstOrDefault();

if (NazvTxt.Text != "Название")

{

izm.Nazvanie = NazvTxt.Text;

}

if (OpisTxt.Text != "Описание")

{

izm.Opisanie = OpisTxt.Text;

}

db.SaveChanges();

list.ItemsSource = db.Fitnes\_Trenirovka\_Klienta.ToList();

MessageBox.Show("Новые данные сохранены!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

}

catch

{

MessageBox.Show("Произошла ошибка на сервере!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

try

{

if (IDTxt.Text != "ID")

{

MessageBoxResult result = MessageBox.Show("Вы уверены что хотите удалить запись?", "Сообщение", MessageBoxButton.YesNo, MessageBoxImage.Question);

var num = Convert.ToInt16(IDTxt.Text);

var izm = db.Fitnes\_Trenirovka\_Klienta.Where(w => w.id\_trenirovki == num).FirstOrDefault();

if (result == MessageBoxResult.Yes)

{

db.Fitnes\_Trenirovka\_Klienta.Remove(izm);

db.SaveChanges();

list.ItemsSource = db.Fitnes\_Trenirovka\_Klienta.ToList();

MessageBox.Show("Запись успешно удалена!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

IDTxt.Items.Clear();

var idd = db.Fitnes\_Trenirovka\_Klienta.Select(w => w.id\_trenirovki).ToList();

IDTxt.Items.Add("ID");

foreach (int item in idd)

{

IDTxt.Items.Add(item);

}

IDTxt.Text = "ID";

}

}

else

{

MessageBox.Show("Заполните поле ID!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

catch

{

MessageBox.Show("Произошла ошибка на сервере!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

if (!System.Text.RegularExpressions.Regex.IsMatch(e.Text, "^[а-яА-Я0-9.,!:;]"))

{

e.Handled = true;

}

}

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

{

if (!System.Text.RegularExpressions.Regex.IsMatch(e.Text, "^[а-яА-Я0-9.,!:;]"))

{

e.Handled = true;

}

}

}

}

<Window x:Class="Fitnes.TrenerKlienti"

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:Fitnes"

mc:Ignorable="d"

Title="Клиенты" Height="775" Width="1050" Loaded="Window\_Loaded" WindowStartupLocation="CenterScreen" ResizeMode="NoResize" WindowStyle="None" MouseDown="Window\_MouseDown">

<Window.Resources>

<Style TargetType="{x:Type TextBox}">

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

<Setter Property="BorderBrush" Value="#FFFF8000"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="Template">

<Setter.Value>

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

<Border Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}">

<ScrollViewer x:Name="PART\_ContentHost" />

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

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

<Setter Property="BorderBrush" Value="White"/>

</Trigger>

</Style.Triggers>

</Style>

<Style TargetType="{x:Type PasswordBox}">

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

<Setter Property="BorderBrush" Value="#FFFF8000"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="Template">

<Setter.Value>

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

<Border Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}">

<ScrollViewer x:Name="PART\_ContentHost" />

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

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

<Setter Property="BorderBrush" Value="White"/>

</Trigger>

</Style.Triggers>

</Style>

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

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

<Setter Property="BorderBrush" Value="#FFFF8000"/>

<Setter Property="Foreground" Value="#FFFF8000"/>

<Setter Property="BorderThickness" Value="2"/>

<Setter Property="Template">

<Setter.Value>

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

<Border Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}">

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

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

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

<Setter Property="Background" Value="#FFFF8000"/>

<Setter Property="Foreground" Value="White"/>

</Trigger>

</Style.Triggers>

</Style>

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

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

<Setter Property="BorderBrush" Value="#FFFF2F00"/>

<Setter Property="Foreground" Value="#FFFF2F00"/>

<Setter Property="BorderThickness" Value="2"/>

<Setter Property="Template">

<Setter.Value>

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

<Border Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}">

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

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

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

<Setter Property="Background" Value="#FFFF2F00"/>

<Setter Property="Foreground" Value="White"/>

</Trigger>

</Style.Triggers>

</Style>

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

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

<Setter Property="BorderBrush" Value="#FFFF8000"/>

<Setter Property="BorderThickness" Value="2"/>

<Setter Property="Template">

<Setter.Value>

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

<Border Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}">

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

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Style.Triggers>

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

<Setter Property="Background" Value="#FFFF8000"/>

</Trigger>

</Style.Triggers>

</Style>

</Window.Resources>

<Grid Background="#FF32363A">

<Grid HorizontalAlignment="Center" Height="88" VerticalAlignment="Bottom" Width="1050" Background="#FF2E2F31" Margin="0,412,0,0">

<ItemsControl HorizontalAlignment="Center" VerticalAlignment="Center" Height="92" Width="800">

<ItemsControl.ItemsPanel>

<ItemsPanelTemplate>

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

</ItemsPanelTemplate>

</ItemsControl.ItemsPanel>

<Button x:Name="RaspBtn" HorizontalAlignment="Left" VerticalAlignment="Top" Width="124" Height="90" FontSize="24" Click="RaspBtn\_Click" Style="{StaticResource Menu}" Margin="0,2,0,0">

<StackPanel Orientation="Vertical">

<Image Width="50" Source="/Resources/Images/schedule.png"/>

<Label FontSize="16" Padding="0" Foreground="White" Content="Расписание"/>

</StackPanel>

</Button>

<Button x:Name="TrenBtn" HorizontalAlignment="Left" VerticalAlignment="Top" Width="126" Height="90" FontSize="24" Click="TrenBtn\_Click" Style="{StaticResource Menu}" Margin="-2,2,0,0">

<StackPanel Orientation="Vertical">

<Image Width="35" Source="/Resources/Images/training.png"/>

<Label FontSize="16" Padding="0" Foreground="White" Content="Тренировки"/>

</StackPanel>

</Button>

<Button x:Name="KlientBtn" HorizontalAlignment="Left" VerticalAlignment="Top" Width="124" Height="90" FontSize="24" Click="KlientBtn\_Click" Style="{StaticResource Menu}" Margin="-2,2,0,0">

<StackPanel Orientation="Vertical">

<Image Width="65" Source="/Resources/Images/clients.png"/>

<Label FontSize="16" Padding="0" Foreground="White" Content="Клиенты"/>

</StackPanel>

</Button>

<Button x:Name="ExitBtn" HorizontalAlignment="Left" VerticalAlignment="Top" Width="126" Height="90" FontSize="24" Click="ExitBtn\_Click" Style="{StaticResource Exit}" Margin="-2,2,0,0">

<StackPanel Orientation="Vertical">

<Image Width="50" Source="/Resources/Images/exit.png"/>

<Label FontSize="16" Padding="0" Foreground="red" Content="Выход"/>

</StackPanel>

</Button>

</ItemsControl>

</Grid>

<Label Content="Клиенты" HorizontalContentAlignment="Center" VerticalContentAlignment="Stretch" FontWeight="Bold" FontSize="48" Foreground="White" Margin="0,0,0,674"/>

<Image Name="LogoBtn" Source="/Resources/Images/logo.png" HorizontalAlignment="Left" Height="75" VerticalAlignment="Top" Width="75" Margin="10,10,0,0" MouseLeftButtonDown="LogoBtn\_MouseLeftButtonDown"/>

<ListView x:Name="list" ScrollViewer.HorizontalScrollBarVisibility="Disabled" Margin="10,151,10,108" HorizontalContentAlignment="Center" FontSize="18" Background="#FF32363A" BorderBrush="#FFFF8011" Foreground="White" SelectionMode="Single" d:ItemsSource="{d:SampleData ItemCount=5}">

<ListView.ItemTemplate>

<DataTemplate>

<Border BorderBrush="#FFFF8001" BorderThickness="1" Width="990" Height="115" Padding="10" Background="#FF2E2F31">

<Grid Height="310" Background="{Binding BackColor}">

<StackPanel Grid.Row="1">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="350"/>

<ColumnDefinition Width="250"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Row="0" Grid.Column="0">

<TextBlock Text="ФИО" FontWeight="Bold" HorizontalAlignment="Center"/>

<TextBlock HorizontalAlignment="Center" TextWrapping="Wrap" TextAlignment="Center">

<Run Text="{Binding Familiya, Mode=OneWay}"/>

<Run Text="{Binding Imya, Mode=OneWay}"/>

<Run Text="{Binding Otchestvo, Mode=OneWay}"/>

</TextBlock>

</StackPanel>

<StackPanel Grid.Row="0" Grid.Column="1">

<TextBlock Text="Телефон" FontWeight="Bold" HorizontalAlignment="Center"/>

<TextBlock Text="{Binding Telefon}" HorizontalAlignment="Center"/>

</StackPanel>

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

<TextBlock Text="Email" FontWeight="Bold" HorizontalAlignment="Center"/>

<TextBlock Text="{Binding Email}" HorizontalAlignment="Center"/>

</StackPanel>

</Grid>

</StackPanel>

</Grid>

</Border>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<TextBox x:Name="txtSearch" HorizontalAlignment="Left" VerticalAlignment="Top" Height="40" Margin="790,106,0,0" TextWrapping="Wrap" Width="250" FontSize="24" TextChanged="txtSearch\_TextChanged" Style="{StaticResource {x:Type TextBox}}"/>

<Button x:Name="SortBtn" Content="Сортировать" HorizontalAlignment="Left" Margin="10,106,0,0" VerticalAlignment="Top" Width="175" Height="40" FontSize="24" Click="Sort\_Click" FontWeight="SemiBold" Style="{StaticResource Btn}"/>

<Button x:Name="ObnovBtn" Content="Обновить" HorizontalAlignment="Left" Margin="190,106,0,0" VerticalAlignment="Top" Width="175" Height="40" FontSize="24" Click="Refresh\_Click" FontWeight="SemiBold" Style="{StaticResource Btn}"/>

</Grid>

</Window>

using Fitnes.Resources;

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.Shapes;

using System.Resources;

namespace Fitnes

{

/// <summary>

/// Логика взаимодействия для TrenerKlienti.xaml

/// </summary>

public partial class TrenerKlienti : Window

{

user155\_dbEntities db = new user155\_dbEntities();

public TrenerKlienti()

{

InitializeComponent();

}

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

{

var Join = (from a in db.Fitnes\_Raspisanie\_Trenirovok

join b in db.Fitnes\_Klient

on a.id\_klienta equals b.id\_klienta

join c in db.Fitnes\_Trener

on a.id\_trenera equals c.id\_trenera

select new { c.id\_trenera, b.Familiya, b.Imya, b.Otchestvo, b.Telefon, b.Email });

list.ItemsSource = Join.Where(w => w.id\_trenera == PublicClass.PublicID).ToList();

txtSearch.Text = "Поиск";

txtSearch.GotFocus += RemoveText;

txtSearch.LostFocus += AddText;

}

public void RemoveText(object sender, EventArgs e)

{

if (txtSearch.Text == "Поиск")

{

txtSearch.Text = "";

}

}

public void AddText(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(txtSearch.Text))

txtSearch.Text = "Поиск";

}

private void Window\_MouseDown(object sender, MouseButtonEventArgs e)

{

if (e.LeftButton == MouseButtonState.Pressed)

DragMove();

}

private void LogoBtn\_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

TrenerGlavnaya a = new TrenerGlavnaya();

a.Show();

this.Close();

}

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

{

TrenerRaspisanie a = new TrenerRaspisanie();

a.Show();

this.Close();

}

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

{

TrenerTrenirovki a = new TrenerTrenirovki();

a.Show();

this.Close();

}

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

{

TrenerKlienti a = new TrenerKlienti();

a.Show();

this.Close();

}

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

{

MessageBoxResult result = MessageBox.Show("Вы уверены что хотите выйти из аккаунта?", "Сообщение", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

MainWindow a = new MainWindow();

a.Show();

this.Close();

}

}

public int sort = 0;

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

{

var Join = (from a in db.Fitnes\_Raspisanie\_Trenirovok

join b in db.Fitnes\_Klient

on a.id\_klienta equals b.id\_klienta

join c in db.Fitnes\_Trener

on a.id\_trenera equals c.id\_trenera

select new { c.id\_trenera, b.Familiya, b.Imya, b.Otchestvo, b.Telefon, b.Email });

if (sort == 0)

{

list.ItemsSource = Join.Where(w => w.id\_trenera == PublicClass.PublicID).OrderByDescending(i => i.Familiya).ToList();

sort = 1;

}

else

{

list.ItemsSource = Join.Where(w => w.id\_trenera == PublicClass.PublicID).OrderBy(a => a.Familiya).ToList();

sort = 0;

}

}

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

{

var Join = (from a in db.Fitnes\_Raspisanie\_Trenirovok

join b in db.Fitnes\_Klient

on a.id\_klienta equals b.id\_klienta

join c in db.Fitnes\_Trener

on a.id\_trenera equals c.id\_trenera

select new { c.id\_trenera, b.Familiya, b.Imya, b.Otchestvo, b.Telefon, b.Email });

list.ItemsSource = Join.Where(w => w.id\_trenera == PublicClass.PublicID).ToList();

}

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

{

try

{

var Join = (from a in db.Fitnes\_Raspisanie\_Trenirovok

join b in db.Fitnes\_Klient

on a.id\_klienta equals b.id\_klienta

join c in db.Fitnes\_Trener

on a.id\_trenera equals c.id\_trenera

select new { c.id\_trenera, b.Familiya, b.Imya, b.Otchestvo, b.Telefon, b.Email });

list.ItemsSource = Join.Where(

w => w.id\_trenera == PublicClass.PublicID && (w.Familiya.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Imya.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Otchestvo.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Telefon.ToLower().Contains(txtSearch.Text.ToLower())

|| w.Email.ToLower().Contains(txtSearch.Text.ToLower()))

).ToList();

if (txtSearch.Text == "")

{

list.ItemsSource = Join.Where(w => w.id\_trenera == PublicClass.PublicID).ToList();

}

else if (txtSearch.Text == "Поиск")

{

list.ItemsSource = Join.Where(w => w.id\_trenera == PublicClass.PublicID).ToList();

}

}

catch

{

MessageBox.Show("Произошла ошибка на сервере!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

}