App xmal

List<string> errors = new List<string>();

if (string.IsNullOrWhiteSpace(tbName.Text))

errors.Add("Услуга не может быть пустой");

if (!int.TryParse(tbCost.Text, out int cost) || cost <= 0)

errors.Add("Цена должна быть положительным числом");

if (!int.TryParse(tbSecond.Text, out int duration) || duration <= 0)

errors.Add("Длительность должна быть положительным числом");

if (errors.Count > 0)

{

MessageBox.Show(string.Join("\n", errors), "Ошибка ввода", MessageBoxButton.OK, MessageBoxImage.Error);

return; // Прерываем выполнение, если есть ошибки

}

var existingService = helper.GetContext().Service

.FirstOrDefault(s => s.Title.Equals(tbName.Text, StringComparison.OrdinalIgnoreCase));

if (existingService != null && currentService == null)

{

MessageBox.Show("Услуга с таким названием уже существует.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return; // Прерываем выполнение, если услуга уже существует

}

// Если текущая услуга существует, обновляем ее, иначе создаем новую

if (currentService != null)

{

currentService.Title = tbName.Text;

currentService.Cost = cost;

currentService.DurationInSeconds = duration;

currentService.Description = tbdescription.Text;

currentService.Discount = double.TryParse(tbDiscount.Text, out double discount) ? discount : 0;

if (!string.IsNullOrEmpty(selectedImagePath))

{

if (!currentService.MainImagePath.StartsWith("/Услуги салона красоты/"))

{

// Если не начинается, добавляем префикс

currentService.MainImagePath = "/Услуги салона красоты/" + selectedImagePath;

MessageBox.Show("Типо нет: " + currentService.MainImagePath);

}

else

{

// Если уже начинается с нужной строки, проверяем, не дублируем ли мы путь

string imageFileName = System.IO.Path.GetFileName(selectedImagePath);

string existingFileName = System.IO.Path.GetFileName(currentService.MainImagePath);

if (existingFileName != imageFileName)

{

// Добавляем только если имена файлов различаются

currentService.MainImagePath += selectedImagePath; // Или другой способ обработки

}

else

{

//MessageBox.Show("Изображение уже добавлено: " + currentService.MainImagePath);

}

}

}

// Обновляем существующую запись

helper.GetContext().SaveChanges();

MessageBox.Show("Услуга успешно отредактирована");

}

else

{

// Создаем новую услугу

Service newService = new Service

{

Title = tbName.Text,

Cost = cost,

DurationInSeconds = duration,

Description = tbdescription.Text,

Discount = double.TryParse(tbDiscount.Text, out double newDiscount) ? newDiscount : 0,

MainImagePath = !string.IsNullOrEmpty(selectedImagePath) ? "/Услуги салона красоты/" + selectedImagePath : null

};

helper.GetContext().Service.Add(newService);

helper.GetContext().SaveChanges();

MessageBox.Show("Услуга успешно сохранена!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

NavigationService.Navigate(new PageGlav(1));

}

}

Main Window xaml

<Window x:Class="WpfApp2.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:WpfApp2"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="148\*"/>

<ColumnDefinition Width="647\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

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

<RowDefinition Height="8\*"></RowDefinition>

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

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" Height="auto" Width="auto" Grid.ColumnSpan="2" Background="#FFDFE2FD">

<Button x:Name="btnBack" HorizontalAlignment="Right" Content="Назад" Width="135" Height="34" Click="btnBack\_Click"/>

</StackPanel>

<StackPanel Grid.Row="2" Height="auto" Width="auto" Grid.ColumnSpan="2" Background="#FFD8DBF5"/>

<Frame x:Name="frame" Height="auto" Width="auto" Grid.ColumnSpan="2" Margin="0,43,0,30" Grid.RowSpan="3" NavigationUIVisibility="Hidden"/>

<Image Source="C:\Users\user\source\repos\SalomKrasoti\SalomKrasoti\Logo\beauty\_logo.ico" HorizontalAlignment="Left" Height="29" Margin="0,1,0,0" VerticalAlignment="Top" Width="36"/>

</Grid>

</Window>

Main window cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WpfApp2.Models;

using WpfApp2.Pages;

using static WpfApp2.MainWindow;

using static WpfApp2.Pages.PageGlav;

namespace WpfApp2

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

frame.Content = new Pages.PageGlav();

}

public class helper

{

public static SalonKrasotiEntities ent;

public static SalonKrasotiEntities GetContext()

{

if (ent == null)

{

ent = new SalonKrasotiEntities();

}

return ent;

}

}

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

{

try

{

if (frame.CanGoBack) // Проверяем, можно ли вернуться назад

{

frame.GoBack();

}

else

{

// Можно добавить дополнительную логику, например, показать сообщение пользователю

MessageBox.Show("Нет предыдущей страницы для возврата.");

}

}

catch (InvalidOperationException ex)

{

// Обработка исключения, если оно все же возникло

MessageBox.Show($"Произошла ошибка: {ex.Message}");

}

}

}

}

Page glav xaml

<Page x:Class="WpfApp2.Pages.PageGlav"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:WpfApp2.Pages"

mc:Ignorable="d"

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

Title="PageGlav">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="41\*" />

<RowDefinition Height="327\*" />

<RowDefinition Height="18\*" />

</Grid.RowDefinitions>

<Grid.Resources>

<DataTemplate x:Key="MyCol">

<Border BorderBrush="#CCCCCC" BorderThickness="1" CornerRadius="5" Padding="10">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<!-- Для текста слева -->

<ColumnDefinition Width="Auto" />

<!-- Для процента справа -->

</Grid.ColumnDefinitions>

<StackPanel Grid.Column="0">

<TextBlock FontFamily="Segoe UI" HorizontalAlignment="Left" TextWrapping="Wrap">

<Run Text="{Binding ID}" FontWeight="Bold" FontSize="16"/> <!-- Увеличиваем размер шрифта -->

<Run Text=": " FontSize="16"/> <!-- Увеличиваем размер шрифта -->

<Run Text="{Binding Title}" FontWeight="Bold" FontSize="16"/> <!-- Увеличиваем размер шрифта -->

</TextBlock>

<TextBlock Text="{Binding Cost, StringFormat='{}{0} рублей'}"/>

<!-- Можно также увеличить размер здесь -->

</StackPanel>

<TextBlock Grid.Column="1" FontFamily="Segoe UI" HorizontalAlignment="Right">

<TextBlock.Style>

<Style TargetType="TextBlock">

<Setter Property="FontSize" Value="16"/>

<!-- Устанавливаем размер шрифта -->

<Setter Property="Text" Value="{Binding Discount, StringFormat='{}{0} %'}"/>

<Style.Triggers>

<DataTrigger Binding="{Binding Discount}" Value="{x:Null}">

<Setter Property="Text" Value=""/>

</DataTrigger>

<DataTrigger Binding="{Binding Discount}" Value="0">

<Setter Property="Text" Value=""/>

</DataTrigger>

</Style.Triggers>

</Style>

</TextBlock.Style>

</TextBlock>

</Grid>

</Border>

</DataTemplate>

</Grid.Resources>

<DataGrid AutoGenerateColumns="False" x:Name="serviceGrid" Grid.Row="1" IsReadOnly="True" GridLinesVisibility="None">

<DataGrid.Columns>

<DataGridTemplateColumn CellTemplate="{StaticResource MyCol}" Width="\*"/>

</DataGrid.Columns>

</DataGrid>

<Grid Grid.Row="0">

<StackPanel Grid.Column="0" Orientation="Horizontal" VerticalAlignment="Center" HorizontalAlignment="Left">

<TextBox x:Name="SearchBox" Width="295" ></TextBox>

<ComboBox x:Name="DiscountBox" Width="97" SelectionChanged="DiscountBox\_SelectionChanged" />

<ComboBox x:Name ="SortBox" VerticalAlignment="Center" Height="41" Width="163" SelectionChanged="SortBox\_SelectionChanged"/>

<Button x:Name = "btncreate" Width="189" Content="Добавить услугу" Click="btncreate\_Click" />

</StackPanel>

</Grid>

<Grid Grid.Row="3">

</Grid>

</Grid>

</Page>

Page glav cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WpfApp2.Models;

using static WpfApp2.MainWindow;

namespace WpfApp2.Pages

{

/// <summary>

/// Логика взаимодействия для PageGlav.xaml

/// </summary>

public partial class PageGlav : Page

{

private List<Service> listspisokservice;

public PageGlav()

{

InitializeComponent();

listspisokservice = helper.GetContext().Service.ToList();

SortBox.Items.Add("Все");

SortBox.Items.Add("0 - 5%");

SortBox.Items.Add("5 - 15%");

//SortBox.Items.Add("15 - 30%");

//SortBox.Items.Add("30 - 70%");

//SortBox.Items.Add("70 - 100%");

SortBox.SelectedIndex = 0;

DiscountBox.Items.Add("Без сортировки");

DiscountBox.Items.Add("По возрастанию");

DiscountBox.Items.Add("По убыванию");

DiscountBox.SelectedIndex = 0;

Load();

}

public void Load()

{

var serviceQuery = listspisokservice.AsQueryable();

switch(SortBox.SelectedItem.ToString())

{

case "0 - 5%":

serviceQuery = serviceQuery.Where(s => s.Discount == null || (s.Discount >= 0 && s.Discount < 5));

break;

case "5 - 15%":

serviceQuery = serviceQuery.Where(s => s.Discount >= 5 && s.Discount < 15);

break;

case "Все":

default:

break;

}

string selectedDiscount = DiscountBox.SelectedItem?.ToString();

switch (selectedDiscount)

{

case "По возрастанию":

serviceQuery = serviceQuery.OrderBy(s => s.Cost); // Сортировка по возрастанию

break;

case "По убыванию":

serviceQuery = serviceQuery.OrderByDescending(s => s.Cost); // Сортировка по убыванию

break;

case "Без сортировки":

default:

break;

}

// Применяем изменения к ItemsSource

var filteredServices = serviceQuery.ToList();

// Обновляем ItemsSource

serviceGrid.ItemsSource = filteredServices;

foreach (var service in filteredServices)

{

if (string.IsNullOrEmpty(service.MainImagePath))

{

service.MainImagePath = "Услуги салона красоты/96.png"; // Путь к изображению по умолчанию

}

}

}

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

{

Load();

}

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

{

Load();

}

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

{

PageEdit pageEdit = new PageEdit();

NavigationService.Navigate(pageEdit);

}

}

}

Page Edit xaml

<Page x:Class="WpfApp2.Pages.PageEdit"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:WpfApp2.Pages"

mc:Ignorable="d"

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

Title="PageEdit">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<Grid x:Name="gridInfoClient" Grid.ColumnSpan="3" Margin="190,0,200,50">

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="1.5\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="1.5\*"/>

</Grid.ColumnDefinitions>

<TextBlock x:Name="lblID" Grid.Row="0" Grid.Column="0" Text="ID:" VerticalAlignment="Center" HorizontalAlignment="Right"/>

<TextBox x:Name="tbID" IsReadOnly="True" Width="100" Grid.Row="0" Grid.Column="1" Margin="10" HorizontalAlignment="Left" VerticalAlignment="Center" IsEnabled="False"/>

<TextBlock Text="Наименование услуги:" Grid.Row="1" Grid.Column="0" VerticalAlignment="Center" HorizontalAlignment="Right"/>

<TextBox x:Name="tbName" Width="100" Grid.Row="1" Grid.Column="1" Margin="10" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<TextBlock Text="Стоимость" Grid.Row="2" Grid.Column="0" VerticalAlignment="Center" HorizontalAlignment="Right"/>

<TextBox x:Name="tbCost" Width="100" Grid.Row="2" Grid.Column="1" Margin="10" HorizontalAlignment="Left" VerticalAlignment="Center" />

<TextBlock Text="Длительность" Grid.Row="3" Grid.Column="0" VerticalAlignment="Center" HorizontalAlignment="Right"/>

<TextBox x:Name="tbSecond" Width="100" Grid.Row="3" Grid.Column="1" Margin="10" HorizontalAlignment="Left" VerticalAlignment="Center" />

<TextBlock Text="Описание\*:" Grid.Row="4" Grid.Column="0" VerticalAlignment="Center" HorizontalAlignment="Right"/>

<TextBox x:Name="tbdescription" Width="100" Grid.Row="4" Grid.Column="1" Margin="10" HorizontalAlignment="Left" VerticalAlignment="Center" />

<TextBlock Text="Скидка:" Grid.Row="5" Grid.Column="0" VerticalAlignment="Center" HorizontalAlignment="Right"/>

<TextBox x:Name="tbDiscount" Width="100" Grid.Row="5" Grid.Column="1" Margin="10" HorizontalAlignment="Left" VerticalAlignment="Center" />

<StackPanel Grid.Row="8" Grid.Column="0" Grid.ColumnSpan="2" Orientation="Horizontal" VerticalAlignment="Bottom" Height="83" Margin="0,0,0,-27">

<Button x:Name="btnEdit" Content="Сохранить изменение" Height="50" Width="264" Click="btnEdit\_Click"/>

</StackPanel>

</Grid>

</Grid>

</Page>

Page Edit cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WpfApp2.Models;

using static WpfApp2.MainWindow;

namespace WpfApp2.Pages

{

/// <summary>

/// Логика взаимодействия для PageEdit.xaml

/// </summary>

public partial class PageEdit : Page

{

Service sr = new Service();

private Service currentService;

private string selectedImagePath;

public string seldnull = "";

public PageEdit()

{

InitializeComponent();

}

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

{

List<string> errors = new List<string>();

if (string.IsNullOrWhiteSpace(tbName.Text))

errors.Add("Услуга не может быть пустой");

if (!int.TryParse(tbCost.Text, out int cost) || cost <= 0)

errors.Add("Цена должна быть положительным числом");

if (!int.TryParse(tbSecond.Text, out int duration) || duration <= 0)

errors.Add("Длительность должна быть положительным числом");

if (errors.Count > 0)

{

MessageBox.Show(string.Join("\n", errors), "Ошибка ввода", MessageBoxButton.OK, MessageBoxImage.Error);

return; // Прерываем выполнение, если есть ошибки

}

var existingService = helper.GetContext().Service

.FirstOrDefault(s => s.Title.Equals(tbName.Text, StringComparison.OrdinalIgnoreCase));

if (existingService != null && currentService == null)

{

MessageBox.Show("Услуга с таким названием уже существует.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return; // Прерываем выполнение, если услуга уже существует

}

// Создаем новую услугу

Service newService = new Service

{

Title = tbName.Text,

Cost = cost,

DurationInSeconds = duration,

Description = tbdescription.Text,

Discount = double.TryParse(tbDiscount.Text, out double newDiscount) ? newDiscount : 0,

MainImagePath = !string.IsNullOrEmpty(selectedImagePath) ? "/Услуги салона красоты/" + selectedImagePath : null

};

helper.GetContext().Service.Add(newService);

helper.GetContext().SaveChanges();

MessageBox.Show("Услуга успешно сохранена!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

NavigationService.Navigate(new PageGlav());

}

}

}