Государственное бюджетное профессиональное образовательное учреждение «Кунгурский сельскохозяйственный колледж»

Специальность 09.02.07 «Информационные системы и программирование»

**КУРСОВОЙ ПРОЕКТ**

**по МДК 06.02 «Инженерно-техническая поддержка сопровождения ИС»**

**на тему: «Разработка ИС для салона красоты»**

Выполнил: студент

группы ИС-49

Букирев Дмитрий Витальевич

Проверил: преподаватель

Завьялова Н.Ю.

Кунгур 2022

Содержание

[Введение 3](#_Toc121470451)

[Анализ предметной области 4](#_Toc121470452)

[Построение БД 6](#_Toc121470453)

[Разработка приложения 8](#_Toc121470454)

[Руководство пользователя 12](#_Toc121470455)

[Заключение 13](#_Toc121470456)

[Список литературы 14](#_Toc121470457)

# Введение

Информационная система (ИС) – это программно-аппаратный комплекс, который используется для сбора, хранения, обработки и выдачи информации. Традиционно ИС работают с огромными объемами сложно структурированных данных. Интеграция программного, аппаратного обеспечения и данных позволяет автоматизировать управление информацией.

Предназначение ИС в том, чтобы своевременно обеспечивать персонал необходимой информацией в определенной области

Целью курсового проекта является разработка информационной системы «Салон красоты».

Задачами курсового проекта являются:

1. Анализ предметной области, создание Use Case диаграммы
2. Проектирование базы данных, создание ER диаграммы и словаря данных
3. Разработка приложения
4. Разработка руководства пользователя

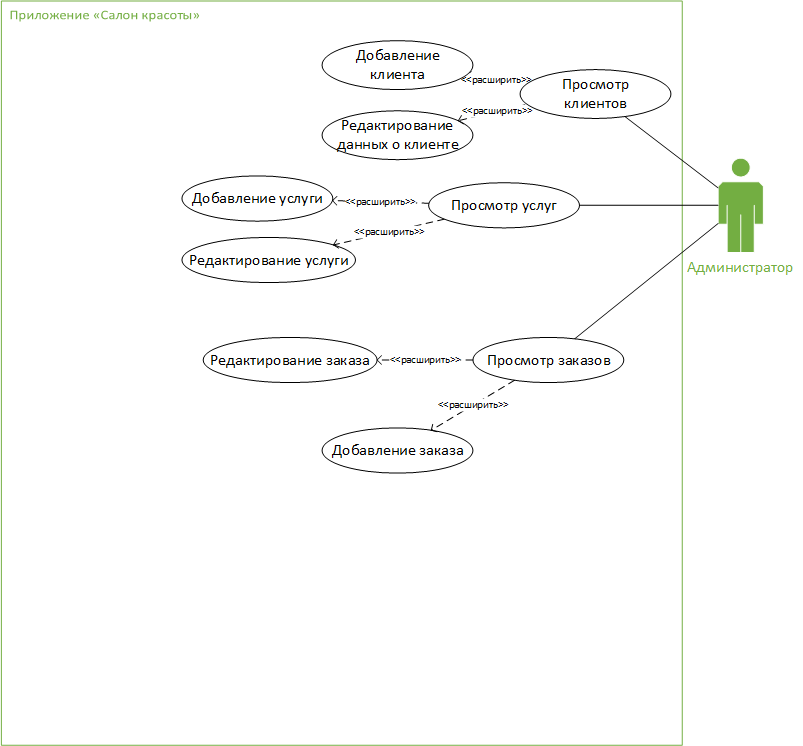
# Анализ предметной области

Основные процессы, на которых основывается деятельность салона: оказание услуг в сфере красоты (маникюр, педикюр, окрашивание волос, наращивание ресниц), продажа и заказ косметических средств, финансовый учет. Администратор непосредственно работает с клиентами (принимает заявки, анализирует отзывы и пожелания), составляет график смен. Закупка средств осуществляется менеджером по закупкам выбранных поставщиков. В случае если товар не соответствует требованиям салона, менеджер по закупкам может вернуть товар, если такое условие предусмотрено договором с поставщиком. Бухгалтер занимается финансовым учетом. Клиент может записаться заочно.

Задачи, решаемые ИС:

1. Хранение информации о заказах
2. Хранение информации об услугах
3. Хранение информации о клиентах

Диаграмма Use-case

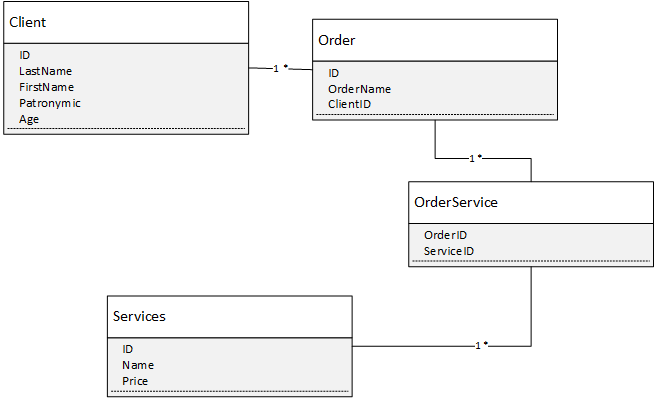
  
Рисунок 1 – Диаграмма Use-case

Администратор может:

* Просматривать услуги
  + Добавлять услуги
  + Изменять услуги
* Просматривать клиентов
  + Добавлять клиентов
  + Изменять клиентов
* Просматривать заказы
  + Добавлять заказы
  + Изменять заказы

# Построение БД

ER диаграмма

  
Рисунок 2 – ER диаграмма

* 1. Словарь данных

Таблица 1 – Сущность Клиенты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Client** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **ID** | Int | Y |  |
|  | **LastName** | VARCHAR (50) | Y |  |
|  | **FirstName** | VARCHAR (50) | Y |  |
|  | **Patronymic** | VARCHAR (50) |  |  |
|  | **Age** | smallint | Y |  |
|  | **Image** | Image |  |  |

Таблица 2 – Сущность Заказы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Order** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **ID** | Int | Y |  |
|  | **OrderName** | Varchar(50) | Y |  |
| FK | **ClientID** | int | Y |  |
|  | **Price** | Money | Y |  |

Таблица 3 – Сущность ЗаказыУслуги

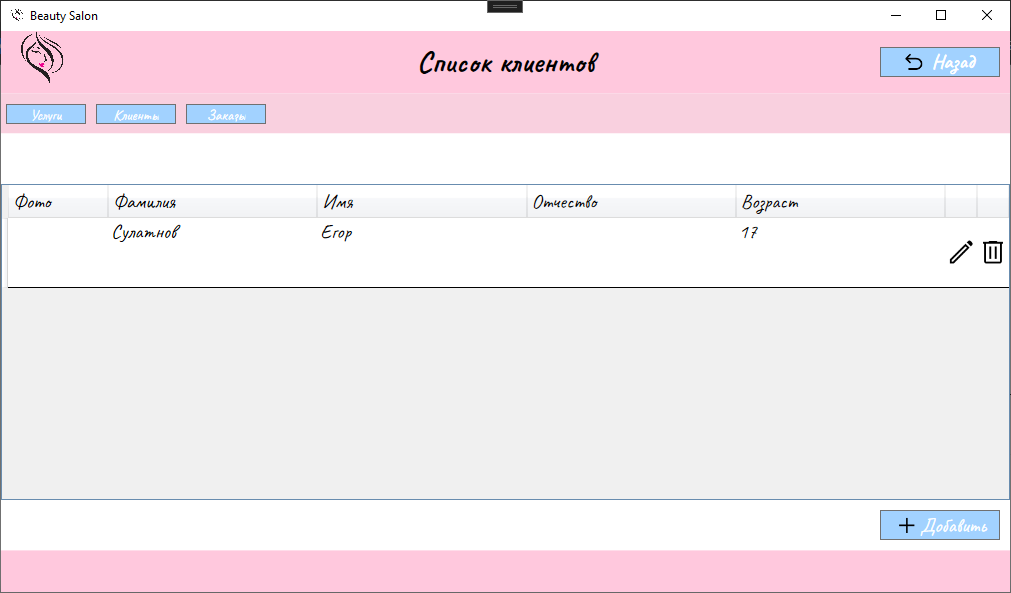
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OrderService** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **ID** | INT | Y |  |
| FK | **OrderID** | INT | Y |  |
| FK | **ServiceID** | INT | Y |  |

Таблица 4 – Сущность Услуги

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **ID** | INT | Y |  |
|  | **Name** | VARCHAR (50) | Y |  |
|  | **Price** | Money | Y |  |

# Разработка приложения

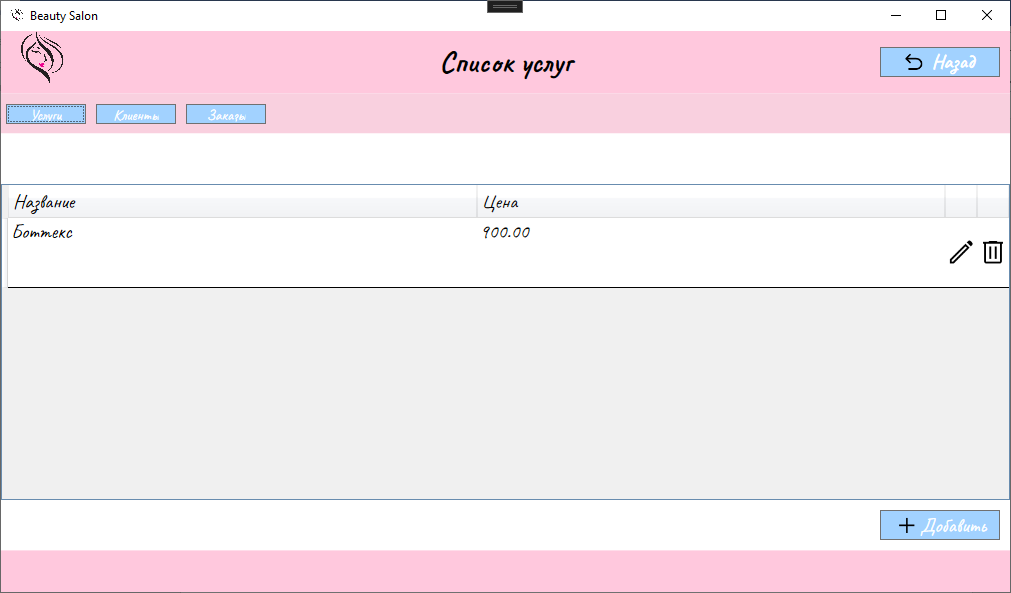
Список клиентов

  
Рисунок 3 – Окно «Список клиентов»

Разметка см. Приложение 1

Код см. Приложение 3

Список услуг

  
Рисунок 4 – Окно «Список услуг»

Разметка аналогична окну «Список клиентов» см. Приложение 1

Код см. Приложение 4

Список заказов

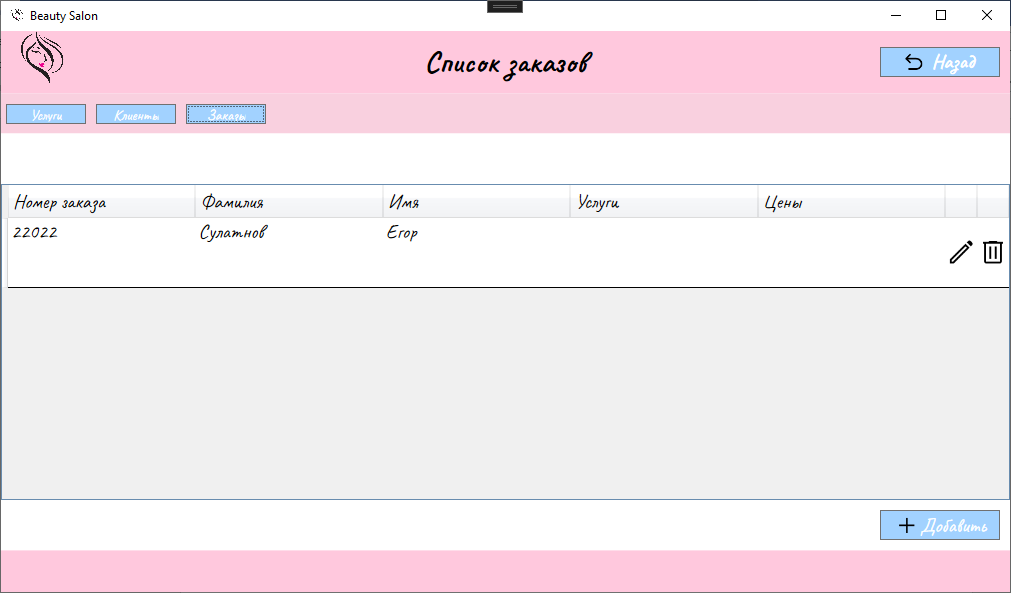
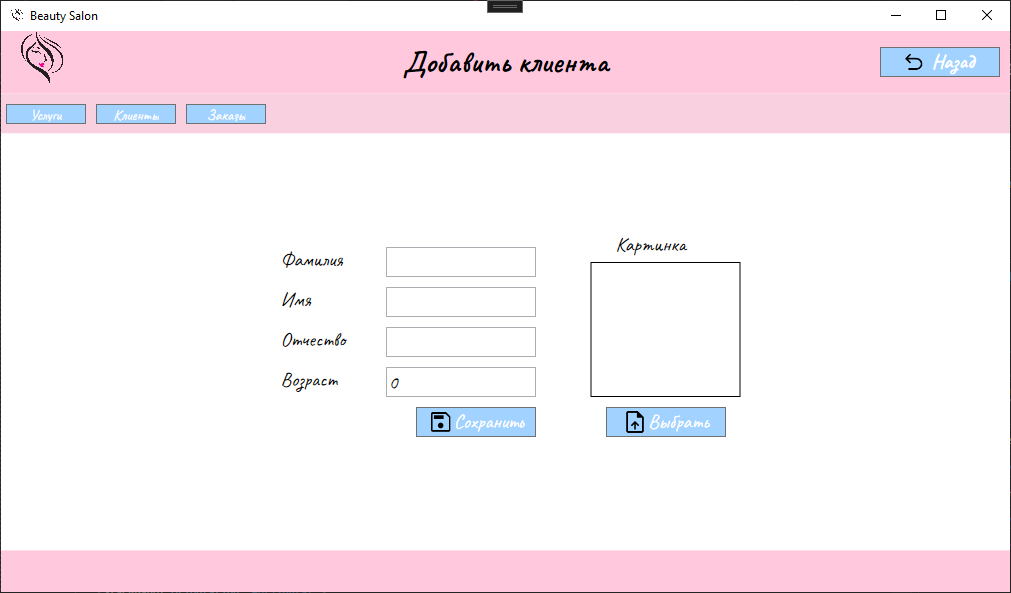


Рисунок 5 – Окно «Список заказов»

Разметка аналогична окну «Список клиентов» см. Приложение 1

Код см. Приложение 5

Добавление/редактирование данных о клиенте

  
Рисунок 6 – Окно «Добавление/Редактирование клиента»

Разметка см. Приложение 2

Код см. Приложение 8

Добавление/Редактирование услуг

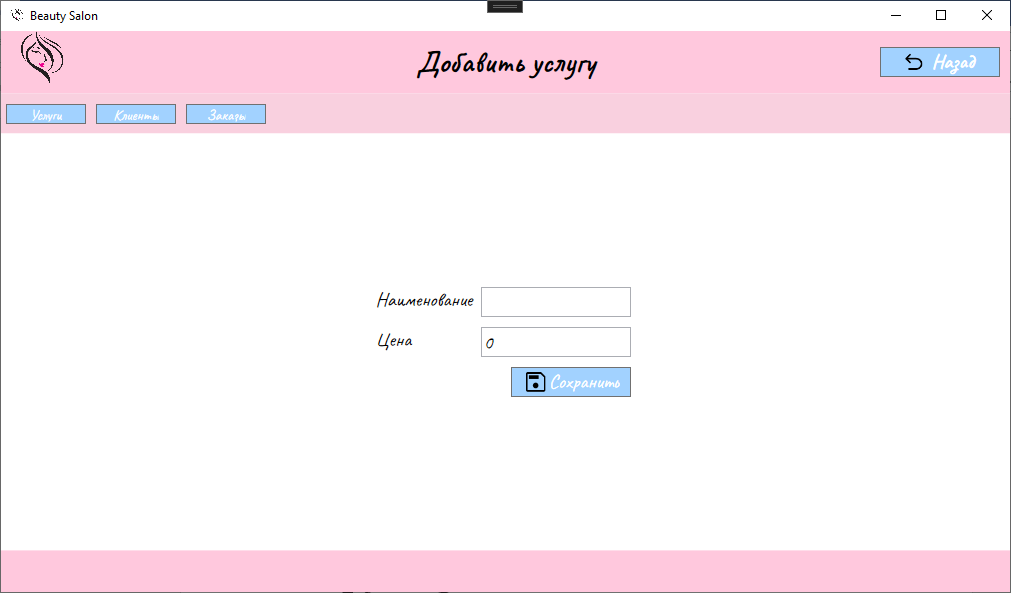


Рисунок 7 – Окно «Добавление/Редактирование услуг»

Разметка аналогична окну «Добавление окна» см. Приложение 2

Код см. Приложение 6

Добавление заказа

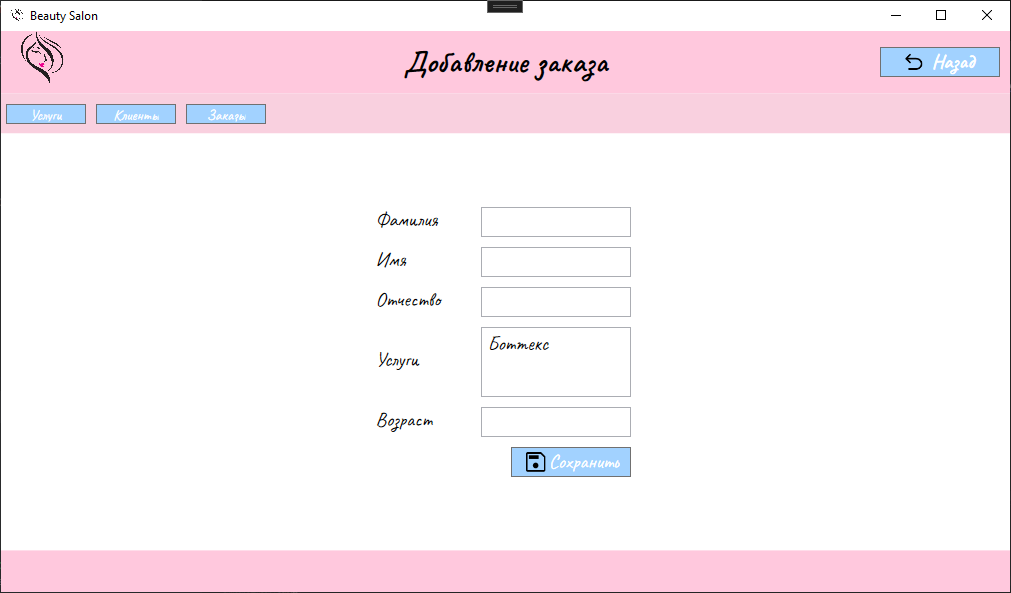


Рисунок 8 – Окно «Добавление заказа»

Разметка аналогична окну «Добавление окна» см. Приложение 2

Код см. Приложение 7

# Руководство пользователя

* Приложение может не запуститься с первого раза из-за этого может потребоваться повторный запуск приложения.
* Для редактирования данных требуется нажать на (значок карандаша), редактирования заказов не реализовано.
* Для удаления данных требуется нажать на (значок корзины).

Известные проблемы

* Не отображаются услуги и их цена в окне заказы.
* Не работает редактирование заказов.

# Заключение

В результате проделанной работы были получены следующие результаты:

1. Рассмотрена деятельность автосалона, создана диаграмма Use Case
2. Спроектирована база данных, создана ER диаграмма, создан словарь данных
3. Разработано приложение
4. Разработано руководство пользователя

В курсовом проекте были разработаны Use Case и ER диаграммы в графическом редакторе Microsoft Visio.

Приложение было разработано на объектно-ориентированном языке C# в визуальной среде Visual Studio.

Для создания базы данных было использовано Microsoft SQL Server.

При разработке курсового проекта были реализованы полученные теоретические знания.

Все поставленные цели и задачи выполнены.

# Список литературы

1. Руководство по WPF  
   <https://metanit.com/sharp/wpf/>
2. Описание салона красоты  
   <https://scienceforum.ru/2019/article/2018011560>

Приложения

Приложение 1

Разметка окна «Список клиентов»

<Page x:Class="Beauty\_Salon.Pages.ClientListPage"

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:Beauty\_Salon.Pages"

mc:Ignorable="d"

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

Title="Список клиентов" FontFamily="/Beauty Salon;component/Resources/Fonts/#Caveat" FontSize="20" Loaded="Page\_Loaded">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<DataGrid Grid.Row="1" x:Name="DGclient">

<DataGrid.Columns>

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

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Image MaxHeight="70" Source="{Binding Image}"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTextColumn Header="Фамилия" Binding="{Binding LastName}" Width="\*"/>

<DataGridTextColumn Header="Имя" Binding="{Binding FirstName}" Width="\*"/>

<DataGridTextColumn Header="Отчество" Binding="{Binding Patronymic}" Width="\*"/>

<DataGridTextColumn Header="Возраст" Binding="{Binding Age}" Width="\*"/>

<DataGridTemplateColumn>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Image MaxHeight="30" Source="/resources/edit.png" x:Name="ImgEdit" MouseLeftButtonDown="ImgEdit\_MouseLeftButtonDown"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn >

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Image MaxHeight="30" Source="/resources/delete.png" x:Name="ImgDelete" MouseLeftButtonDown="ImgDelete\_MouseLeftButtonDown"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<Button Grid.Row="2" x:Name="BtnAdd" HorizontalAlignment="Right" Click="BtnAdd\_Click">

<StackPanel Orientation="Horizontal">

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

<TextBlock Text="Добавить"/>

</StackPanel>

</Button>

</Grid>

</Page>

Приложение 2

Разметка окна «Добавить клиента»

<Page x:Class="Beauty\_Salon.Pages.AddEditClient"

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:Beauty\_Salon.Pages"

mc:Ignorable="d"

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

Title="Добавить клиента" FontFamily="/Beauty Salon;component/Resources/Fonts/#Caveat">

<Grid>

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

<StackPanel Margin="30">

<StackPanel Orientation="Horizontal">

<TextBlock Text="Фамилия" Style="{StaticResource TBmain}"/>

<TextBox x:Name="TboxLname"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Имя" Style="{StaticResource TBmain}"/>

<TextBox x:Name="TboxFname"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Отчество" Style="{StaticResource TBmain}"/>

<TextBox x:Name="TboxPatronymic"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Возраст" Style="{StaticResource TBmain}"/>

<TextBox x:Name="TboxAge">

<TextBox.Text>

<Binding Path="Age" UpdateSourceTrigger="PropertyChanged">

<Binding.ValidationRules>

<ExceptionValidationRule/>

</Binding.ValidationRules>

</Binding>

</TextBox.Text>

</TextBox>

</StackPanel>

<Button x:Name="BtnSave" HorizontalAlignment="Right" Margin="5" Click="BtnSave\_Click">

<StackPanel Orientation="Horizontal">

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

<TextBlock Text="Сохранить"/>

</StackPanel>

</Button>

</StackPanel>

<StackPanel Margin="20">

<TextBlock Text="Картинка" Style="{StaticResource TBmain}"/>

<Border BorderBrush="Black" BorderThickness="1" Width="150" Height="135">

<Image x:Name="ClientImage"/>

</Border>

<Button x:Name="BtnSelectImg" Click="BtnSelectImg\_Click">

<StackPanel Orientation="Horizontal">

<Image Source="/resources/upload.png"/>

<TextBlock Text="Выбрать"/>

</StackPanel>

</Button>

</StackPanel>

</StackPanel>

</Grid>

</Page>

Приложение 3

Код окна «Список клиентов»

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Input;

using System.Windows.Navigation;

namespace Beauty\_Salon.Pages

{

/// <summary>

/// Логика взаимодействия для ClientListPage.xaml

/// </summary>

public partial class ClientListPage : Page

{

public ClientListPage()

{

InitializeComponent();

}

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

{

UpdateClients();

}

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

{

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

}

private void UpdateClients()

{

var clients = App.Context.Clients.ToList();

DGclient.ItemsSource = clients;

}

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

{

var currentClient = (sender as Image).DataContext as Entities.Client;

NavigationService.Navigate(new Pages.AddEditClient(currentClient));

}

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

{

var currentClient = (sender as Image).DataContext as Entities.Client;

if (MessageBox.Show($"Вы уверены, что хотите удалить клиента: " + $"{currentClient.LastName}?", "Внимание",

MessageBoxButton.YesNo, MessageBoxImage.Warning) == MessageBoxResult.Yes)

{

App.Context.Clients.Remove(currentClient);

App.Context.SaveChanges();

}

UpdateClients();

}

}

}

Приложение 4

Код окна «Список услуг»

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Input;

using System.Windows.Navigation;

namespace Beauty\_Salon.Pages

{

/// <summary>

/// Логика взаимодействия для ServiceListPage.xaml

/// </summary>

public partial class ServiceListPage : Page

{

public ServiceListPage()

{

InitializeComponent();

UpdateServices();

}

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

{

UpdateServices();

}

private void UpdateServices()

{

var service = App.Context.Services.ToList();

DGservice.ItemsSource = service;

}

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

{

var currentService = (sender as Image).DataContext as Entities.Service;

NavigationService.Navigate(new Pages.AddEditServicePage(currentService));

UpdateServices();

}

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

{

var currentService = (sender as Image).DataContext as Entities.Service;

if (MessageBox.Show($"Вы уверены, что хотите удалить услугу: " + $"{currentService.Name}?", "Внимание",

MessageBoxButton.YesNo, MessageBoxImage.Warning) == MessageBoxResult.Yes)

{

App.Context.Services.Remove(currentService);

App.Context.SaveChanges();

}

UpdateServices();

}

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

{

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

UpdateServices();

}

}

}

Приложение 5

Код окна «Список заказов»

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Input;

using System.Windows.Navigation;

namespace Beauty\_Salon.Pages

{

/// <summary>

/// Логика взаимодействия для OrderListPage.xaml

/// </summary>

public partial class OrderListPage : Page

{

public OrderListPage()

{

InitializeComponent();

}

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

{

UpdateOrders();

}

private void UpdateOrders()

{

var orders = App.Context.Orders.ToList();

var orderService = App.Context.OrderServices.ToList();

DGorders.ItemsSource = orders;

DGorders.DataContext = orderService;

}

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

{

NavigationService.Navigate(new AddEditOrderPage());

UpdateOrders();

}

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

{

var currentOrder = (sender as Image).DataContext as Entities.Order;

if (MessageBox.Show($"Вы уверены, что хотите удалить заказ: " + $"{currentOrder.OrderName}?", "Внимание",

MessageBoxButton.YesNo, MessageBoxImage.Warning) == MessageBoxResult.Yes)

{

App.Context.Orders.Remove(currentOrder);

App.Context.SaveChanges();

}

UpdateOrders();

}

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

{

UpdateOrders();

}

}

}

Приложение 6

Код окна «Добавление/Редактирования услуг»

using Beauty\_Salon.Entities;

using System.Linq;

using System.Text;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Navigation;

namespace Beauty\_Salon.Pages

{

/// <summary>

/// Логика взаимодействия для AddEditServicePage.xaml

/// </summary>

public partial class AddEditServicePage : Page

{

private Entities.Service currentService = null;

public AddEditServicePage()

{

InitializeComponent();

DataContext = new Service();

}

public AddEditServicePage(Entities.Service service)

{

InitializeComponent();

DataContext = service;

currentService = service;

Title = "Редактировать услугу";

TboxName.Text = currentService.Name;

TboxPrice.Text = currentService.Price.ToString();

}

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

{

var errorMessage = CheckError();

if (errorMessage.Length > 0)

{

MessageBox.Show(errorMessage, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

else

{

if (currentService == null)

{

var service = new Service()

{

Name = TboxName.Text,

Price = int.Parse(TboxPrice.Text)

};

App.Context.Services.Add(service);

App.Context.SaveChanges();

}

else

{

currentService.Name = TboxName.Text;

currentService.Price = int.Parse(TboxName.Text);

App.Context.SaveChanges();

}

NavigationService.GoBack();

}

}

private string CheckError()

{

var errorBuilder = new StringBuilder();

if (string.IsNullOrEmpty(TboxName.Text))

errorBuilder.AppendLine("Название услуги обязательно для заполнения");

var serviceFromDB = App.Context.Services.ToList()

.FirstOrDefault(p => p.Name.ToLower() == TboxName.Text.ToLower());

if (serviceFromDB != null)

errorBuilder.AppendLine("Такая услуга уже есть в базе данных");

decimal cost = 0;

if (decimal.TryParse(TboxPrice.Text, out cost) == false || cost <= 0)

errorBuilder.AppendLine("Стоимость услуги должна быть положительным числом");

if (errorBuilder.Length > 0)

errorBuilder.Insert(0, "Устраните следующие ошибки:\n");

return errorBuilder.ToString();

}

}

}

Приложение 7

Код окна «Добавление/Редактирования заказов»

using System;

using System.Linq;

using System.Text;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Navigation;

namespace Beauty\_Salon.Pages

{

/// <summary>

/// Логика взаимодействия для AddEditOrderPage.xaml

/// </summary>

public partial class AddEditOrderPage : Page

{

Entities.Client \_currentClient = null;

Entities.Order \_currentOrder = null;

Entities.OrderService \_currentOrderService = null;

public AddEditOrderPage()

{

InitializeComponent();

DataContext = new Entities.Order();

DataContext = new Entities.Client();

DataContext = new Entities.OrderService();

LboxService.ItemsSource = App.Context.Services.ToList();

}

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

{

var errorMessage = ErrorCheck();

if (errorMessage.Length > 0)

{

MessageBox.Show(errorMessage, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

else

{

if (\_currentClient == null)

{

var client = new Entities.Client

{

LastName = TboxLname.Text,

FirstName = TboxFname.Text,

Patronymic = TboxPatronymic.Text,

Age = short.Parse(TboxAge.Text)

};

var order = new Entities.Order

{

OrderName = DateTime.Now.ToString()

};

var orderService = new Entities.OrderService

{

ServiceID = int.Parse(LboxService.SelectedValue.ToString())

};

App.Context.Clients.Add(client);

App.Context.Orders.Add(order);

App.Context.OrderServices.Add(orderService);

App.Context.SaveChanges();

}

NavigationService.GoBack();

}

}

private string ErrorCheck()

{

var errorBuilder = new StringBuilder();

if (string.IsNullOrWhiteSpace(TboxLname.Text))

errorBuilder.AppendLine("Фамилия обязательня для заполнения");

if (string.IsNullOrWhiteSpace(TboxFname.Text))

errorBuilder.AppendLine("Имя обязательня для заполнения");

short age = 0;

if (short.TryParse(TboxAge.Text, out age) == false || age <= 0 || age >= 200)

errorBuilder.AppendLine("Возраст должен быть положительным числом и не больше 200");

if (errorBuilder.Length > 0)

errorBuilder.Insert(0, "Устраните следующие ошибки:\n");

return errorBuilder.ToString();

}

}

}

Приложение 8

Код окна «Добавление/Редактирования клиентов»

using Microsoft.Win32;

using System.IO;

using System.Text;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Media;

using System.Windows.Navigation;

namespace Beauty\_Salon.Pages

{

/// <summary>

/// Логика взаимодействия для AddEditClient.xaml

/// </summary>

public partial class AddEditClient : Page

{

private Entities.Client \_currentClient = null;

private byte[] \_mainImageData;

public AddEditClient()

{

InitializeComponent();

DataContext = new Entities.Client();

}

public AddEditClient(Entities.Client client)

{

InitializeComponent();

DataContext = client;

\_currentClient = client;

Title = "Редактировать данные о клиенте";

TboxLname.Text = \_currentClient.LastName;

TboxFname.Text = \_currentClient.FirstName;

TboxPatronymic.Text = \_currentClient.Patronymic;

TboxAge.Text = \_currentClient.Age.ToString();

if (\_currentClient.Image != null)

ClientImage.Source = (ImageSource) new ImageSourceConverter().ConvertFrom(\_currentClient.Image);

}

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

{

var errorMessage = ErrorCheck();

if (errorMessage.Length > 0 )

{

MessageBox.Show(errorMessage,"Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

else

{

if (\_currentClient == null)

{

var client = new Entities.Client

{

LastName = TboxLname.Text,

FirstName = TboxFname.Text,

Patronymic = TboxPatronymic.Text,

Age = short.Parse(TboxAge.Text),

Image = \_mainImageData

};

App.Context.Clients.Add(client);

App.Context.SaveChanges();

}

else

{

\_currentClient.LastName = TboxLname.Text;

\_currentClient.FirstName = TboxFname.Text;

\_currentClient.Patronymic = TboxPatronymic.Text;

\_currentClient.Age = short.Parse(TboxAge.Text);

if (\_mainImageData != null)

\_currentClient.Image = \_mainImageData;

App.Context.SaveChanges();

}

NavigationService.GoBack();

}

}

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

{

OpenFileDialog ofd = new OpenFileDialog();

ofd.Filter = "Image | \*.png; \*.jpg; \*.jpeg";

if (ofd.ShowDialog() == true)

{

\_mainImageData = File.ReadAllBytes(ofd.FileName);

ClientImage.Source = (ImageSource)new ImageSourceConverter().ConvertFrom(\_mainImageData);

}

}

private string ErrorCheck()

{

var errorBuilder = new StringBuilder();

if (string.IsNullOrWhiteSpace(TboxLname.Text))

errorBuilder.AppendLine("Фамилия обязательня для заполнения");

if (string.IsNullOrWhiteSpace(TboxFname.Text))

errorBuilder.AppendLine("Имя обязательня для заполнения");

short age = 0;

if (short.TryParse(TboxAge.Text, out age) == false || age <= 0 || age >=200)

errorBuilder.AppendLine("Возраст должен быть положительным числом и не больше 200");

if (errorBuilder.Length > 0)

errorBuilder.Insert(0, "Устраните следующие ошибки:\n");

return errorBuilder.ToString();

}

}

}