Министерство образования Московской области

Государственное образовательное учреждение высшего образования Московской области

«Государственный гуманитарно-технологический университет»

**Промышленно-экономический колледж**

**КУРСОВОЙ ПРОЕКТ**

Разработка программного модуля информационной системы «Гостиничный комплекс»

МДК 01.01 «Разработка программных модулей»

**Выполнил:**

Полуянская Анастасия

Георгиевна

студентка группы ИСП.21.1А\_\_\_\_\_\_\_\_\_\_\_\_\_

09.02.07 Информационные системы и программирование)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

очной формы обучения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Руководитель:**

Кузьмина Елена Евгеньевна­­­­­­­­­­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка

\_\_\_\_\_\_\_\_\_/\_\_\_­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Подпись руководителя

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Орехово-Зуево

2024 год

Оглавление

[**Введение** 3](#_Toc162637572)

[**1.** **Назначение разработки** 4](#_Toc162637573)

[**2.** **Требования к программному модулю.** 5](#_Toc162637574)

[**2.1.** **Требования к функциональным характеристикам** 5](#_Toc162637575)

[**2.2.** **Требования к составу и параметрам технических средств** 5](#_Toc162637576)

[**2.3.** **Требования к информационной и программной совместимости** 6](#_Toc162637577)

[**3.** **Руководство программиста.** 7](#_Toc162637578)

[**3.1.** **Разработка базы данных** 7](#_Toc162637579)

[**3.2.** **Разработка программного модуля** 13](#_Toc162637580)

[**3.3.** **Текст программы** 21](#_Toc162637581)

[**4.** **Руководство пользователя** 27](#_Toc162637582)

[**4.1.** **Выполнение программного модуля** 27](#_Toc162637583)

[**4.2.** **Сообщения пользователю** 32](#_Toc162637584)

[**5.** **Тестирование и отладка программного модуля.** 35](#_Toc162637585)

[**Заключение.** 38](#_Toc162637586)

[**Список литературы:** 39](#_Toc162637587)

**Введение**

Гостиничный комплекс - средство размещения, состоящее из определённого количества номеров, имеющее единое руководство, предоставляющее набор услуг.

Гостиничный комплекс "Лазурный Берег" представляет собой программное приложение, разработанное для обеспечения эффективного администрирования и управления гостиничным комплексом. Это приложение предназначено для автоматизации основных процессов управления гостиничным бизнесом, обеспечивая удобный доступ к данным и инструментам управления для администраторов комплекса через интерфейс программы.

1. **Назначение разработки**

Целью создания программного модуля для администрирования гостиничного комплекса "Лазурный Берег" является предоставление решения для управления гостиничным бизнесом, работы с базой данных по предметной области. Модуль предназначен для упрощения мониторинга и ведения базы данных.

1. **Требования к программному модулю.**
   1. **Требования к функциональным характеристикам**

Программный модуль информационной системы «Лазурный Берег» должен помочь сотрудникам автоматизировать обработку данных по клиентам и бронированиям. Также он должен упростить оформление документации по бронированиям.

Следюущие функции должны быть реализованы в приложении:

* Авторизация в программе при запуске.
* Просмотр данных из базы данных.
* Корректный вывод данных из базы данных.
* Добавление новых данных.
* Изменение существующих данных.
* Удаление существующих данных.
* Переходы по страницам приложения.
* Вывод на печать данных о бронированиях.
* Поиск по фамилиям клиентов.
  1. **Требования к составу и параметрам технических средств**

Таблица 1. «Минимальные характеристики компьютера»

|  |  |
| --- | --- |
| Минимальные характеристики компьютера | |
| Процессор | AMD Athlon, Intel Core i3-7100 |
| Оперативная память | От 4Гб |
| Память | От 1Гб |
| Разрешение экрана | 1240x768 |
| Устройство ввода | Клавиатура, мышь |

Таблица 2. «Оптимальные характеристики компьютера»

|  |  |
| --- | --- |
| Оптимальные характеристики компьютера | |
| Процессор | AMD Ryzen 5 3600, Intel Core i5-10600К |
| Оперативная память | От 8 Гб |
| Память | От 1 Гб |
| Разрешение экрана | 1920x1080 |
| Устройство ввода | Клавиатура, мышь |

* 1. **Требования к информационной и программной совместимости**

Для корректной работы программы необходимо:

1. **ОС Windows 10/11 –** операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией **Microsoft** в рамках семейства **Windows NT**.
2. **Microsoft Word 2016 –** текстовый редактор, предназначенный для создания, редактирования, просмотра и форматирования текстов. Используется для ведения хода разработки и составление отчета, чтобы подвести итог работы программы.
3. **SQLite** **–** компактная встраиваемая СУБД.
4. **Visual Studio 2022** (и выше) **–** интегрированная среда разработки программного обеспечения и ряд других инструментов. Используется как основная среда разработки приложения.
5. **Язык программирования С# -** объектно-ориентированный язык программирования общего назначения. Используется как основной язык программирования приложения.
6. **ADO.NET Entity Framework** – объектно-ориентированная технология доступа к данным.
7. **Руководство программиста.**

Для выполнения приложения необходима была нужна база данных, где будет храниться информация о клиентах, бронированиях, номерах и сотрудниках. Была выбрана СУБД **SQLite**.

**SQLite–** бесплатная компактная реляционная база данных.

Плюсы:

* Быстрая и эффективная работа.
* Независимость от сервера.
* Компактность.
* Простота использования.
* Поддержка многопользовательской среды.

Минусы:

* Ограниченные типы данных.
* Отсутствие некоторых продвинутых функций.

Мой выбор остановился на SQLite так как это отличное решение для небольших проектов, не требующих сложных возможностей масштабирования.

* 1. **Разработка базы данных**
* Создание Базы данных «hotel» с помощью DB Browser (SQLite)

Новая базы данных – «ЛКМ».

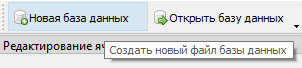


Рисунок 1. «Создание БД»

В открывшемся проводнике ввести название базы данных «hotel» и нажать «сохранить».

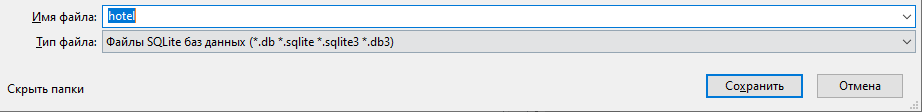


Рисунок 2. «Название БД»

* Создание таблиц



Рисунок 3. «Создание таблиц»

Таблица «Бронирования» со столбцами типами данных. В этой таблице хранится информация о бронированиях:

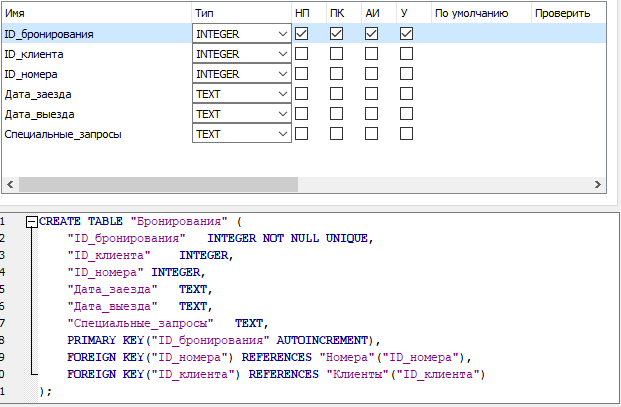


Рисунок 4. Таблица «Бронирования»

Таблица «Клиенты» со столбцами и типами данных. В этой таблице хранится информация о клиентах:

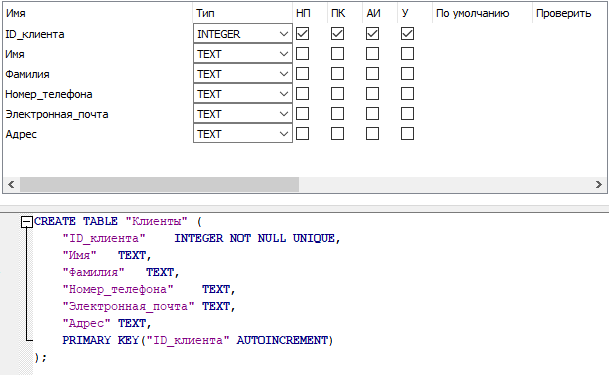


Рисунок 5. Таблица «Клиенты»

Таблица «Номера» со столбцами и типами данных. В этой таблице хранится информация о номерах, возможных для бронирования в гостинице:

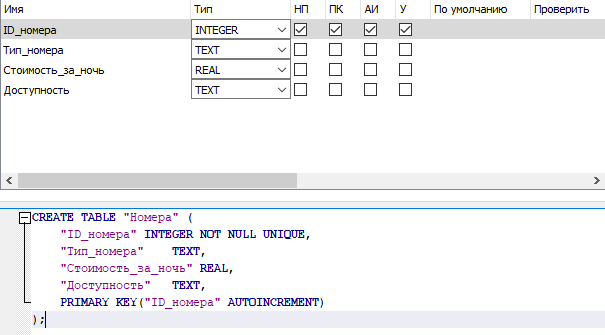


Рисунок 6. Таблица «Номера»

Таблица «Сотрудники» со столбцами и типами данных. В этой таблице хранится информация о сотрудниках, работающих в гостинице:

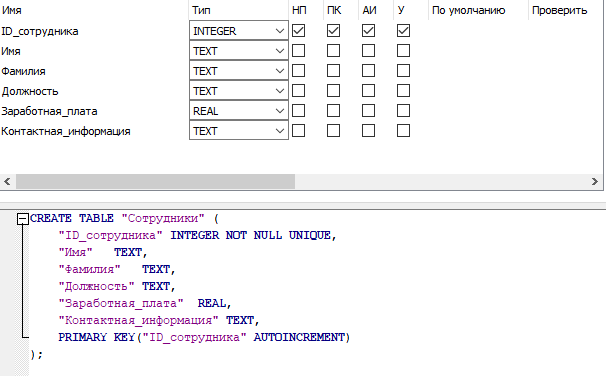


Рисунок 7. Таблица «Сотрудники»

Таблица «Услуги» со столбцами и типами данных. В этой таблице хранится информация об услугах, предоставляемых гостиницей:

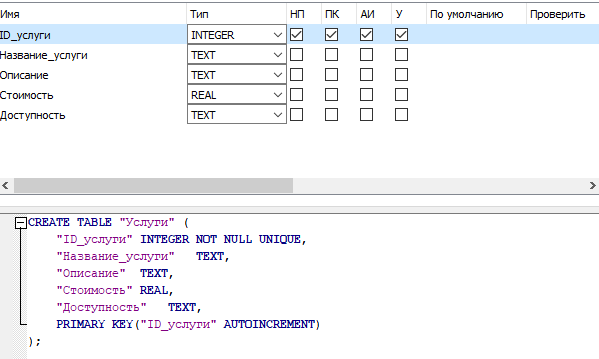


Рисунок 8. Таблица «Услуги»

Таблица «Users» со столбцами и типами данных. В этой таблице хранится информация для авторизации пользователя в приложении:

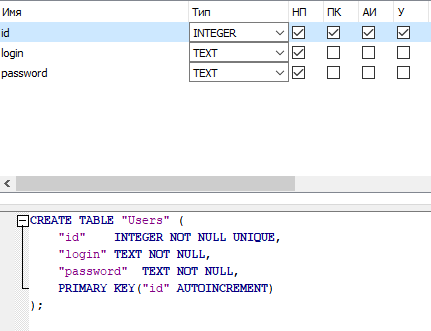


Рисунок 9. Таблица «Users»

* ER диаграммы базы данных:

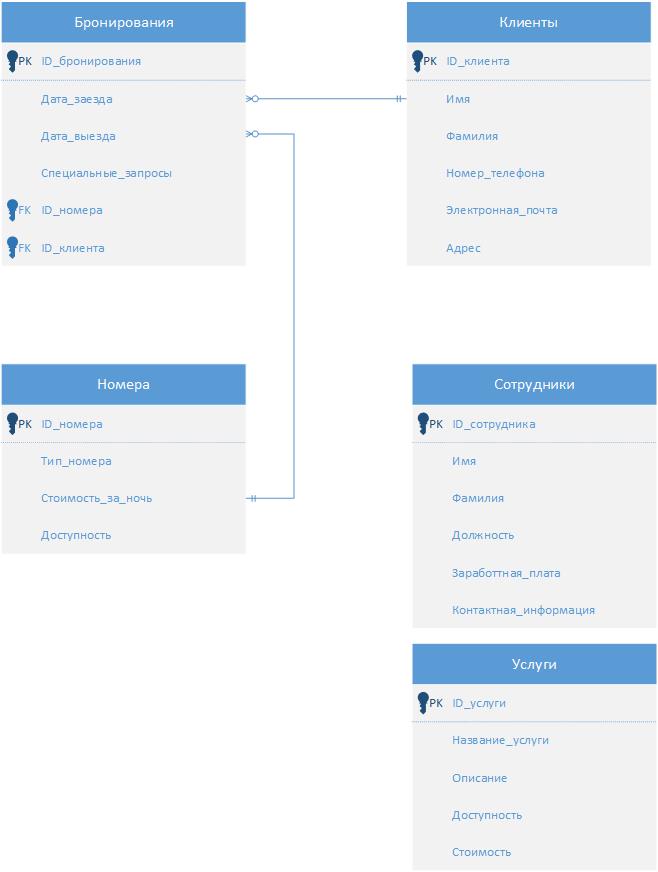


Рисунок 10. ER диаграмма БД

* 1. **Разработка программного модуля**

Программа разработана с использование MS Visual Studio 2022.

Шаги по созданию программы:

1. Создание проекта:

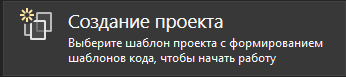


Рисунок 11. Создание проекта

1. Тип проекта:

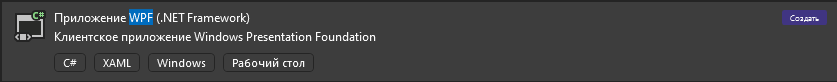


Рисунок 12. Тип проекта

1. Обозреваель решений:

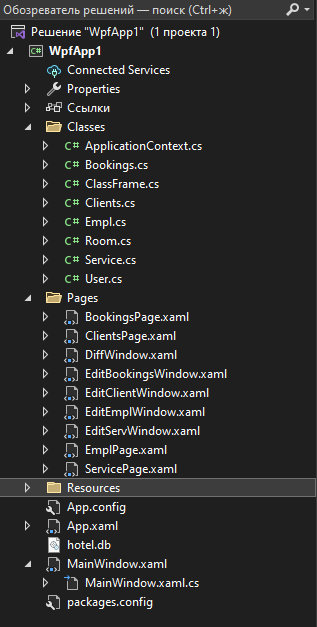


Рисунок 13. Обозреватель решений

1. Присоединение БД:
2. Созданный ранее файл базы данных нужно перенести в файлы проекта:

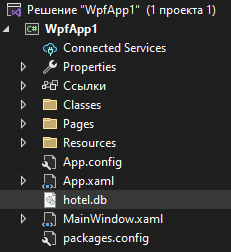


Рисунок 14. Файл базы данных в обозревателе решений

1. В файле App.config нужно прописать код для подключения к базе данных:

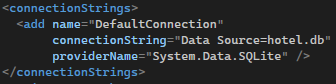


Рисунок 15. Строка подключения к БД

1. Создать класс для использования таблиц бд:

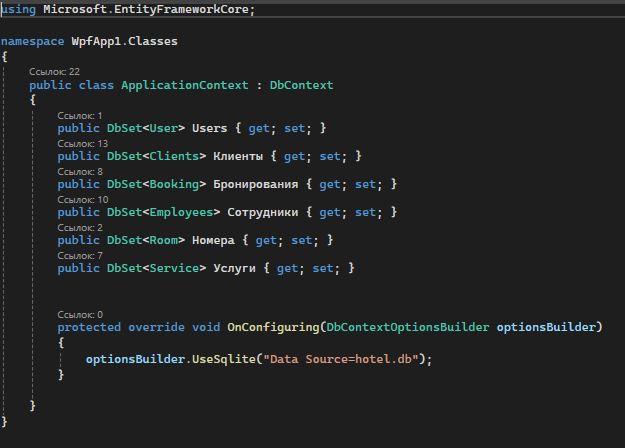


Рисунок 16. ApplicationContext.cs

1. Создать классы для таблиц бд подобно этому:

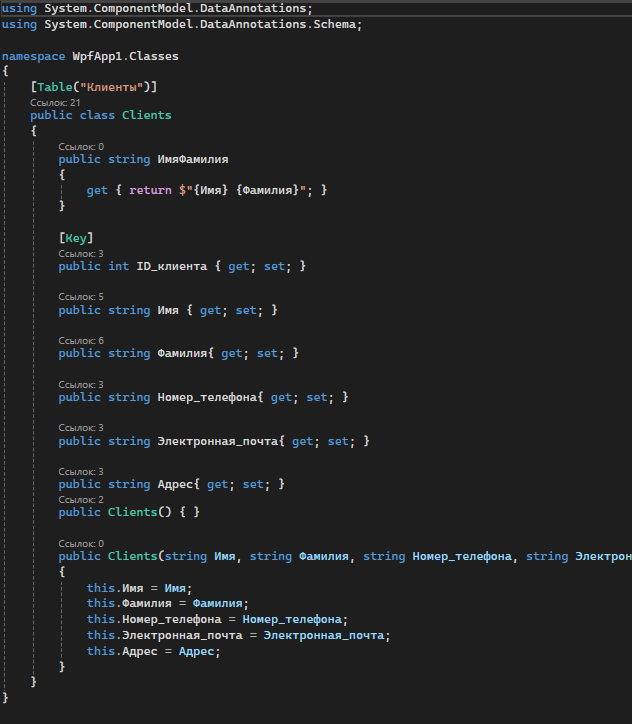


Рисунок 17. Clients.cs

1. Создание страниц и окон.
2. Добавить-Страница(WPF)

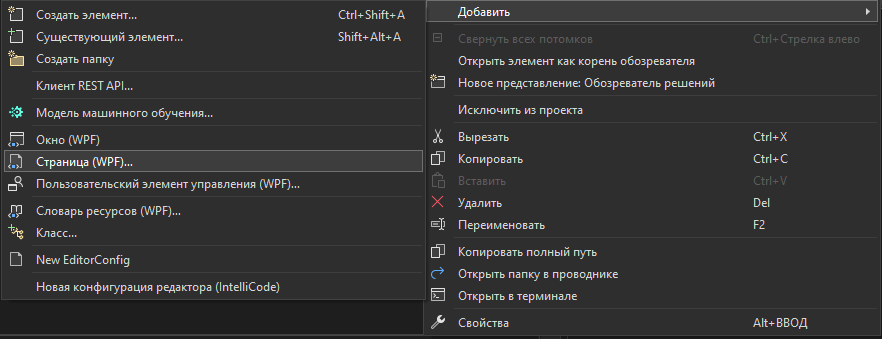


Рисунок 18. Добавление страницы

1. Добавить-Окно(WPF)

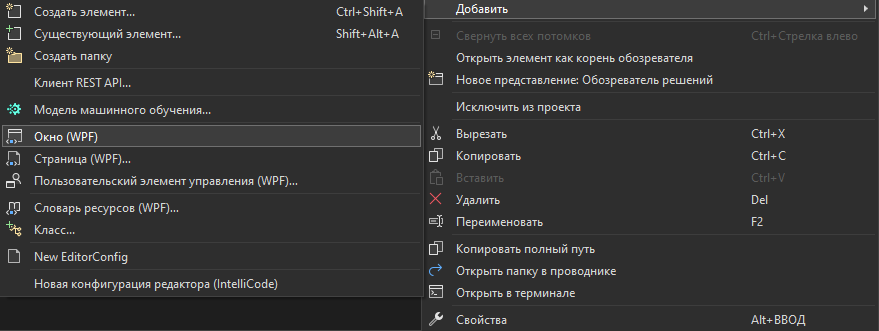


Рисунок 19. Добавление окна

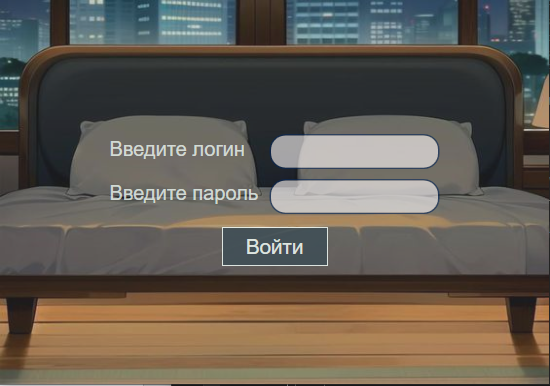


Рисунок 20. Окно «MainWindow»

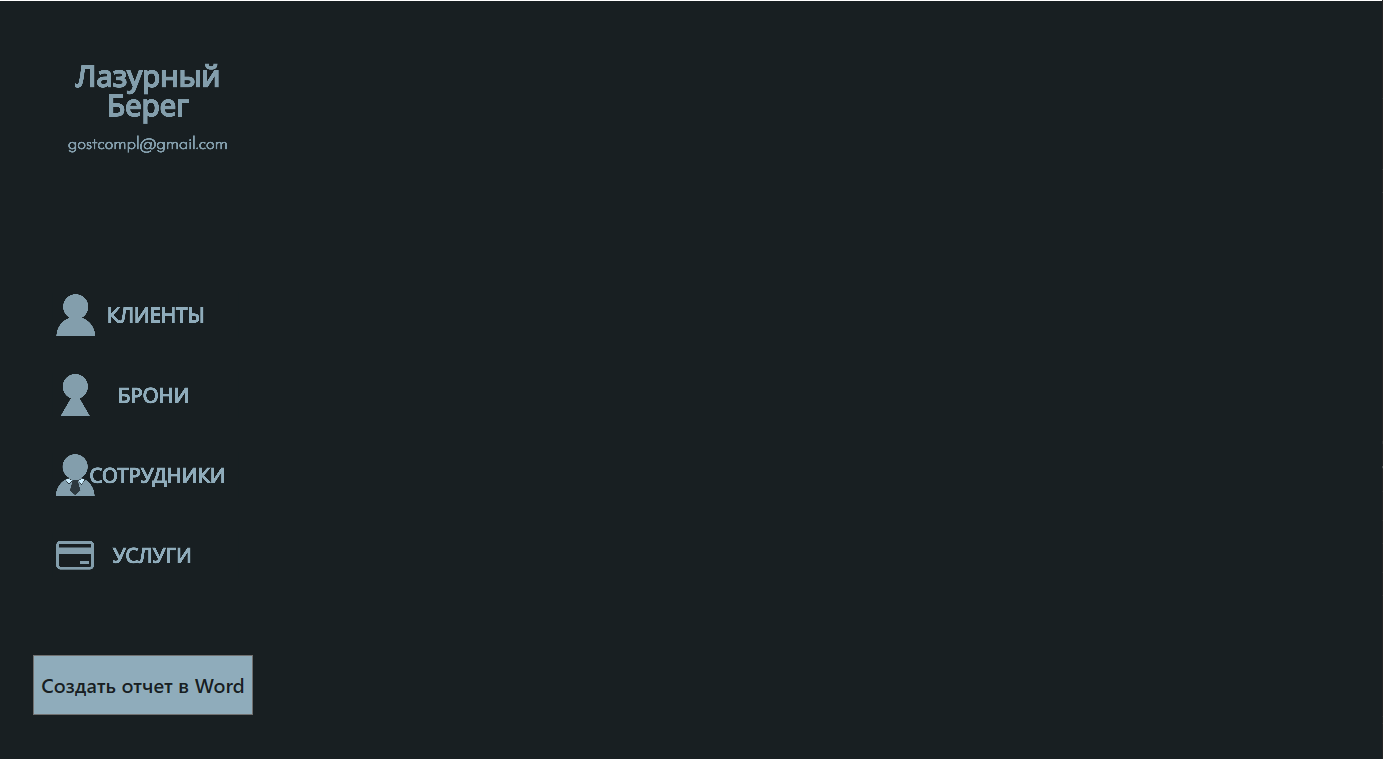


Рисунок 21. Страница «DiffWindow»

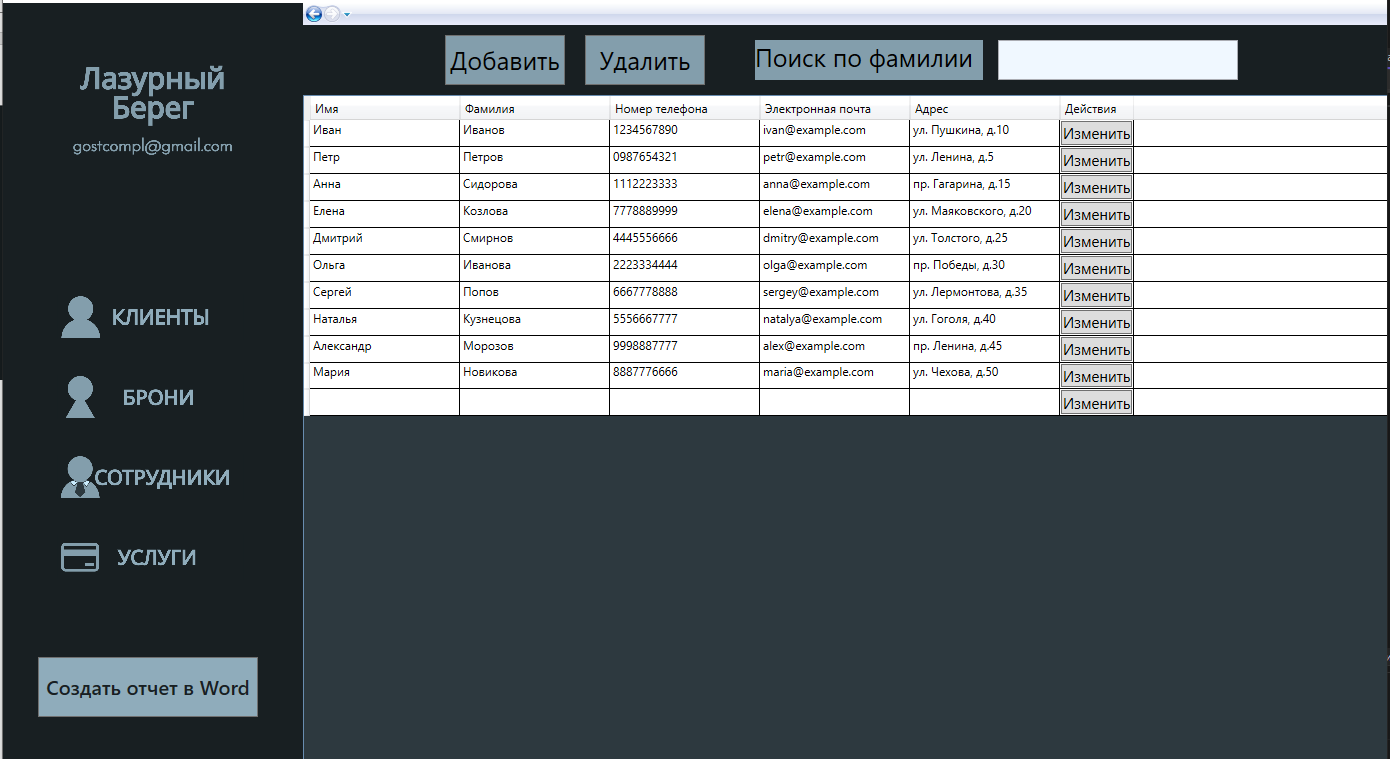


Рисунок 22. Страница «ClientsPage»

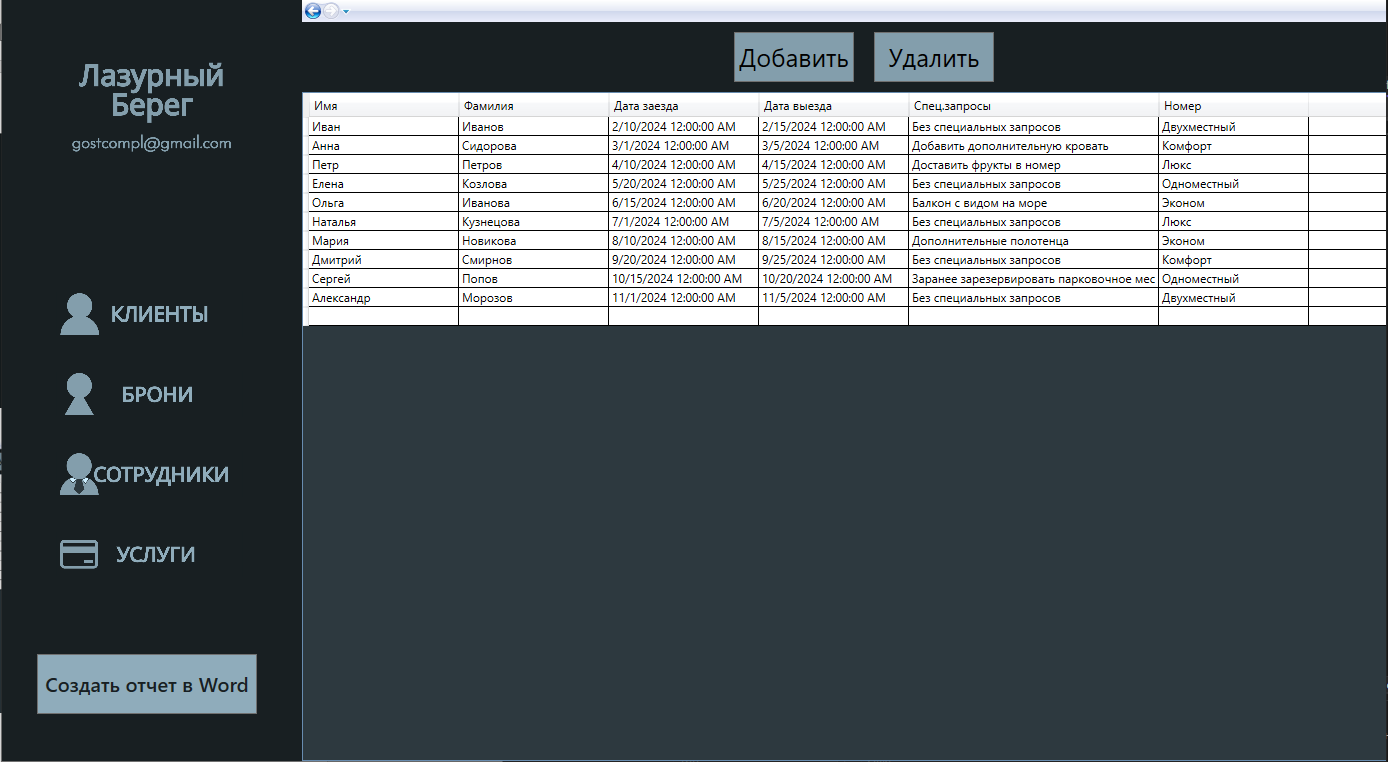


Рисунок 23. Страница «BookingsPage»

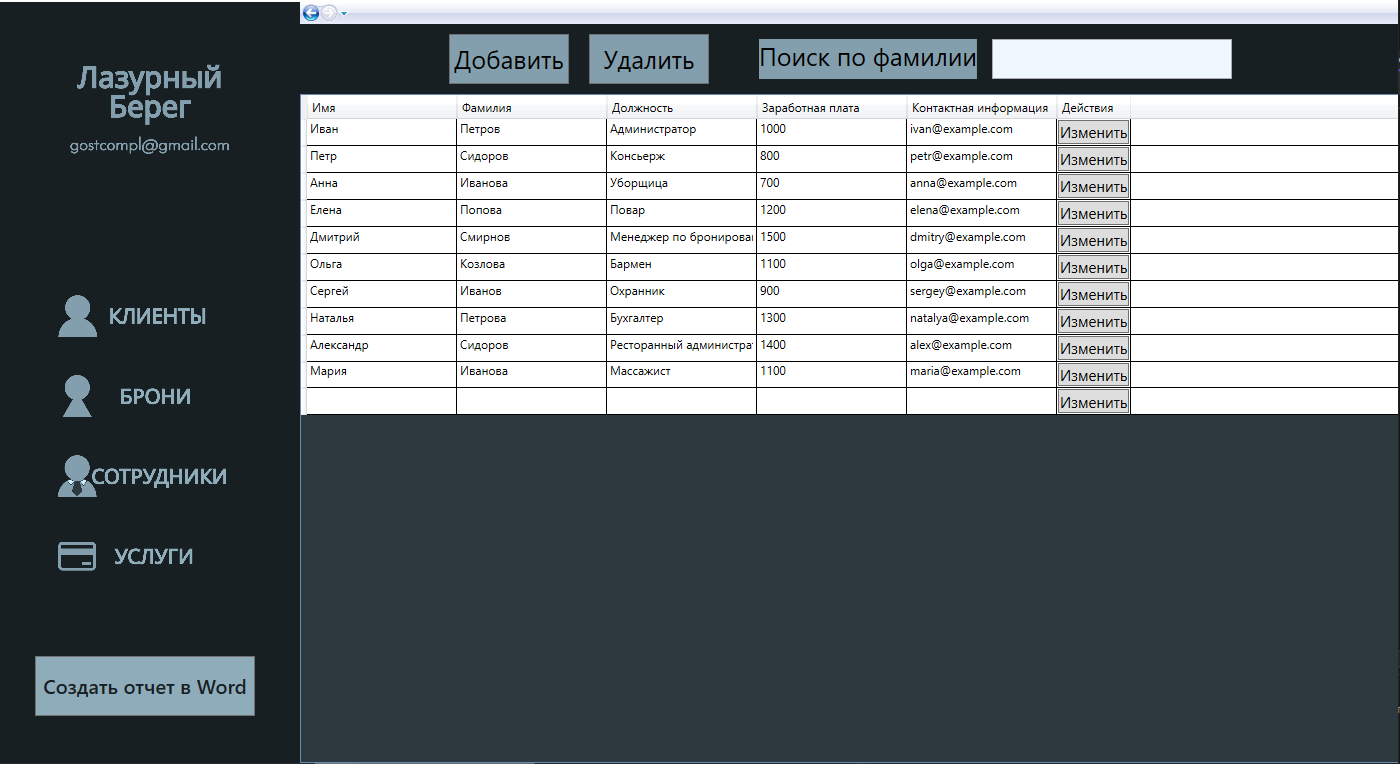


Рисунок 24. Страница «EmplPage»

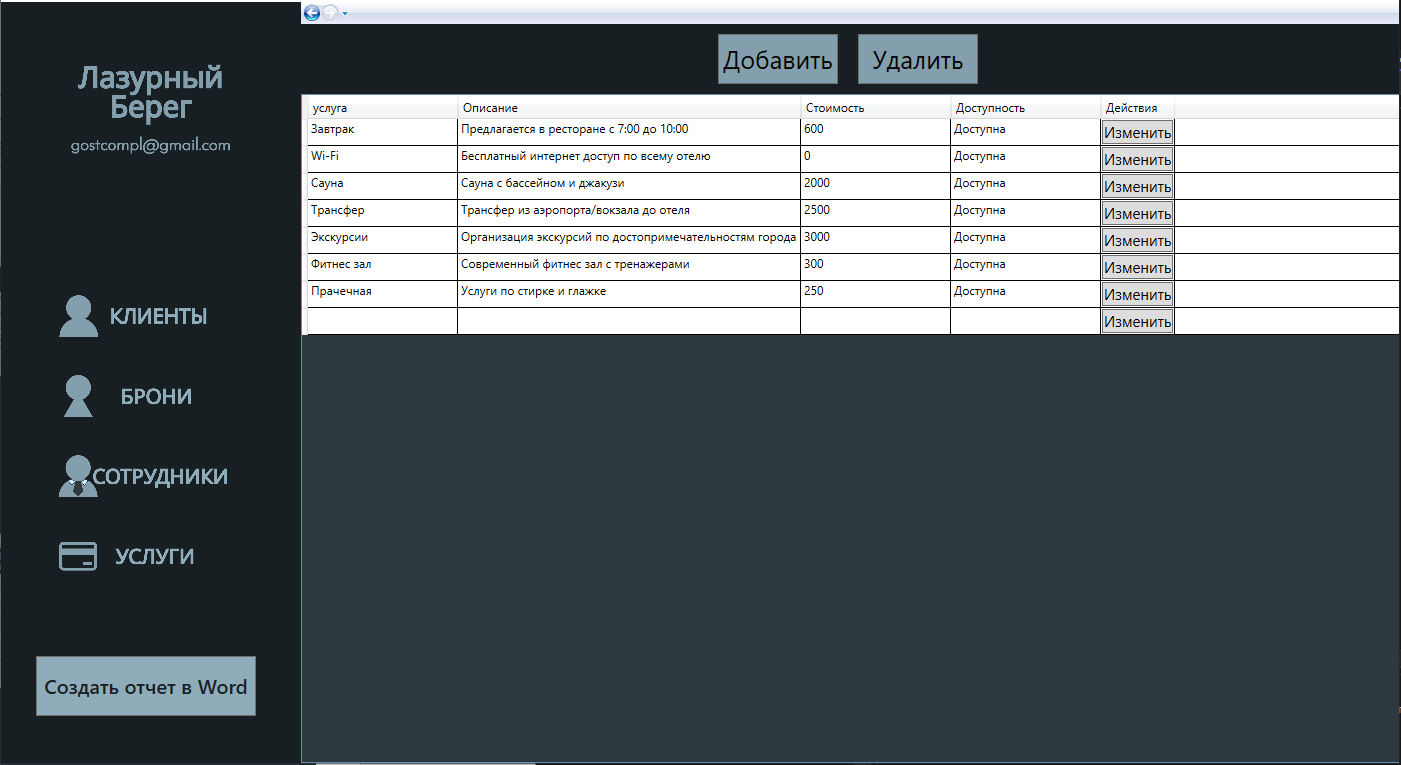


Рисунок 25. Страница «ServicePage»

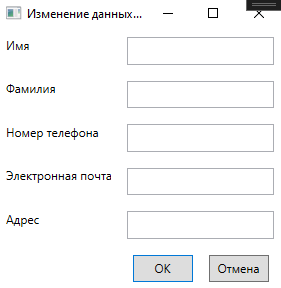


Рисунок 26. Окно «EditClientWindow»

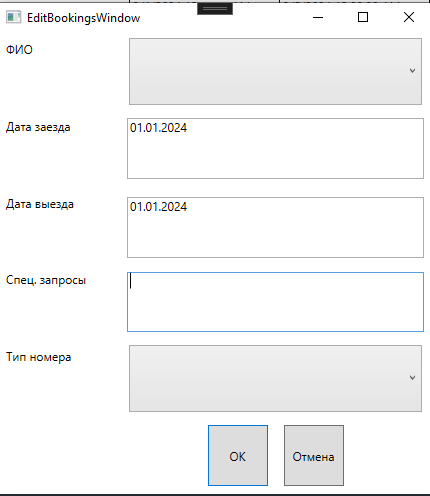


Рисунок 27. Окно «EditBookingsWidow»

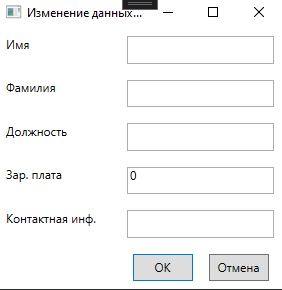


Рисунок 28. Окно «EditEmplWindow»

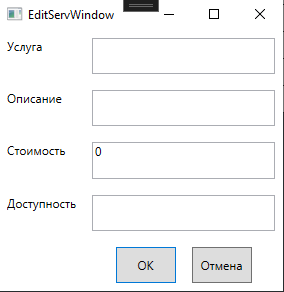


Рисунок 29. Окно «EditServWidow»

* 1. **Текст программы**

Страница MainWindow – страница авторизации пользователя в приложении.

MainWindow.xaml.cs код:

using System.Linq;

using System.Windows;

using System.Windows.Media;

using WpfApp1.Classes;

using WpfApp1.Pages;

namespace WpfApp1

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

}

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

{

string Login = textBoxLogin.Text.Trim();

string Pass = textBoxPassword.Password.Trim();

if (Login.Length < 1)

{

textBoxLogin.ToolTip = "Данные некорректны";

textBoxLogin.Background = Brushes.Red;

}

else if (Pass.Length < 1)

{

textBoxPassword.ToolTip = "Данные некорректны";

textBoxPassword.Background = Brushes.Red;

}

else

{

textBoxLogin.ToolTip = "";

textBoxLogin.Background = Brushes.Transparent;

textBoxPassword.ToolTip = "";

textBoxPassword.Background = Brushes.Transparent;

User auth = null;

using (ApplicationContext db = new ApplicationContext())

{

auth = db.Users.Where(b => b.Login == Login && b.Password == Pass).FirstOrDefault();

}

if (auth != null)

{

MessageBox.Show("Успешно");

DiffWindow newWindow = new DiffWindow();

newWindow.Show();

this.Close();

}

else

{

MessageBox.Show("Данные неверны");

}

}

}

}

}

В окне DiffWindow находятся все навигационные кнопки и кнопка вывода в Word.

DiffWindow.xaml код:

<Window x:Class="WpfApp1.Pages.DiffWindow"

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:WpfApp1.Pages"

mc:Ignorable="d"

Title="" Height="800" Width="1400">

<Grid Background="#181F22">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="300" ></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

</Grid.ColumnDefinitions>

<Frame Grid.Column="1" x:Name="info"/>

<StackPanel>

<Image Source="\Resources\5.png" Height="169" Width="290" Margin="5 20 5 5"/>

</StackPanel>

<WrapPanel Background="#181F22" VerticalAlignment="Top" HorizontalAlignment="Center" Height="585" Width="280" RenderTransformOrigin="0.5,0.5" Margin="0,199,0,0">

<WrapPanel.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform/>

<RotateTransform Angle="0"/>

<TranslateTransform/>

</TransformGroup>

</WrapPanel.RenderTransform>

<Button Name="clients\_btn" BorderBrush="#181F22" Click="clients\_btn\_click" Width="195" Height="50" Margin="40 90 5 25" Background="#181F22" >

<Image Source="\Resources\1.png" Height="45" Width="263"/>

</Button>

<Button Name="bookings\_btn" BorderBrush="#181F22" Click="bookings\_btn\_click" Width="195" Height="50" Margin="40 5 5 25" Background="#181F22">

<Image Source="\Resources\2.png" Height="45" Width="263"/>

</Button>

<Button Name="empl\_btn" BorderBrush="#181F22" Click="empl\_btn\_click" Width="195" Height="50" Margin="40 5 5 25" Background="#181F22">

<Image Source="\Resources\3.png" Height="45" Width="191"/>

</Button>

<Button Name="serv\_btn" BorderBrush="#181F22" Click="serv\_btn\_click" Width="195" Height="50" Margin="40 5 5 25" Background="#181F22">

<Image Source="\Resources\4.png" Height="45" Width="263"/>

</Button>

<Button FontSize="20" Content="Создать отчет в Word" Click="CreateWordReport\_Click" Width="220" Height="60" Margin="25 50 5 25" Background="#FF8FACBB" FontFamily="Segoe UI Semibold" Foreground="#FF181F22"/>

</WrapPanel>

</Grid>

</Window>

DiffWindow.xaml.сs код:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows;

using WpfApp1.Classes;

using Microsoft.Office.Interop.Word;

namespace WpfApp1.Pages

{

public partial class DiffWindow : System.Windows.Window

{

public class BookingInfo

{

public string Имя { get; set; }

public string Фамилия { get; set; }

public string Тип\_номера { get; set; }

public DateTime? Дата\_заезда { get; set; }

public DateTime? Дата\_выезда { get; set; }

}

public List<BookingInfo> GetBookingWithClientAndRoomInfo()

{

using (var db = new ApplicationContext())

{

var bookingsInfo = (from booking in db.Бронирования

join client in db.Клиенты on booking.ID\_клиента equals client.ID\_клиента

join room in db.Номера on booking.ID\_номера equals room.ID\_номера

select new BookingInfo

{

Имя = client.Имя,

Фамилия = client.Фамилия,

Тип\_номера = room.Тип\_номера,

Дата\_заезда = booking.Дата\_заезда,

Дата\_выезда = booking.Дата\_заезда

}).ToList();

return bookingsInfo;

}

}

ApplicationContext db = new ApplicationContext();

private BookingsPage \_viewModel;

public DiffWindow()

{

WindowStartupLocation = WindowStartupLocation.CenterScreen;

InitializeComponent();

\_viewModel = new BookingsPage();

DataContext = \_viewModel;

}

private void clients\_btn\_click(object sender, RoutedEventArgs e)

{

Pages.ClientsPage clientsPage = new Pages.ClientsPage();

info.NavigationService.Navigate(clientsPage);

}

private void bookings\_btn\_click(object sender, RoutedEventArgs e)

{

Pages.BookingsPage bookingsPage = new Pages.BookingsPage();

info.NavigationService.Navigate(bookingsPage);

}

private void empl\_btn\_click(object sender, RoutedEventArgs e)

{

Pages.EmplPage emplPage = new Pages.EmplPage();

info.NavigationService.Navigate(emplPage);

}

private void serv\_btn\_click(object sender, RoutedEventArgs e)

{

Pages.ServicePage emplPage = new Pages.ServicePage();

info.NavigationService.Navigate(emplPage);

}

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

{

List<Booking> bookings = db.Бронирования.ToList();

CreateWordDocument(bookings);

}

Код для вывода данных о бронированиях в Word:

public void CreateWordDocument(List<Booking> bookings)

{

string desktopPath = Environment.GetFolderPath(Environment.SpecialFolder.Desktop);

Microsoft.Office.Interop.Word.Application wordApp = new Microsoft.Office.Interop.Word.Application();

Document doc = wordApp.Documents.Add();

Microsoft.Office.Interop.Word.Table table = doc.Tables.Add(doc.Range(), bookings.Count + 1, 5);

table.Cell(1, 1).Range.Text = "Имя";

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

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

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

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

var bookingsInfo = GetBookingWithClientAndRoomInfo();

int rowIndex = 2;

foreach (var bookingInfo in bookingsInfo)

{

table.Cell(rowIndex, 1).Range.Text = bookingInfo.Имя;

table.Cell(rowIndex, 2).Range.Text = bookingInfo.Фамилия;

table.Cell(rowIndex, 3).Range.Text = bookingInfo.Тип\_номера;

table.Cell(rowIndex, 4).Range.Text = bookingInfo.Дата\_заезда.ToString();

table.Cell(rowIndex, 5).Range.Text = bookingInfo.Дата\_выезда.ToString();

rowIndex++;

}

string filename = System.IO.Path.Combine(desktopPath, "Отчет\_о\_бронированиях.docx");

doc.SaveAs2(filename);

MessageBox.Show("Документ успешно создан и сохранён на рабочем столе.", " ", MessageBoxButton.OK, MessageBoxImage.Information);

doc.Close();

wordApp.Quit();

}

}

}

Код для загрузки данных из БД:

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

{

db.Database.EnsureCreated();

db.Клиенты.Load();

clientsList.ItemsSource = db.Клиенты.Local.ToObservableCollection();

}

Код для поиска клиента по его фамилии:

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

{

string search = txtSearchClient.Text;

clientsList.ItemsSource = db.Клиенты

.Where(x => x.Фамилия.ToLower().Contains(search))

.ToList();

}

Код для редактирования записей БД:

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

{

var button = sender as Button;

var client = button?.DataContext as Clients;

if (client != null)

{

var editClientWindow = new EditClientWindow(client);

editClientWindow.ShowDialog();

}

else

{

EditClientWindow editClientWindow = new EditClientWindow(new Clients());

if (editClientWindow.ShowDialog() == true)

{

Clients clients = editClientWindow.Клиенты;

db.Клиенты.Add(clients);

db.SaveChanges();

clientsList.ItemsSource = db.Клиенты.ToList();

}

}

}

Код для добавления и удаления записи БД:

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

{

EditClientWindow editClientWindow = new EditClientWindow(new Clients());

if (editClientWindow.ShowDialog() == true)

{

Clients clients = editClientWindow.Клиенты;

db.Клиенты.Add(clients);

db.SaveChanges();

clientsList.ItemsSource = db.Клиенты.ToList();

}

}

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

{

Clients selectedClient = clientsList.SelectedItem as Clients;

if (selectedClient == null)

return;

db.Клиенты.Remove(selectedClient);

db.SaveChanges();

clientsList.ItemsSource = db.Клиенты.ToList();

}

Код для объеденения и загрузки таблиц:

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

{

db.Database.EnsureCreated();

db.Бронирования.Load();

DataContext = db.Бронирования.Local.ToObservableCollection();

try

{

var bookingsWithClients = db.Бронирования

.Include(b => b.Клиенты)

.Include(b => b.Номера)

.ToList();

bookingsDataGrid.ItemsSource = bookingsWithClients;

}

catch (Exception ex)

{

HandleException(ex);

}

}

1. **Руководство пользователя**
   1. **Выполнение программного модуля**

Для запуска приложения необходимо запустить .exe файл

Окно авторизации при запуске:

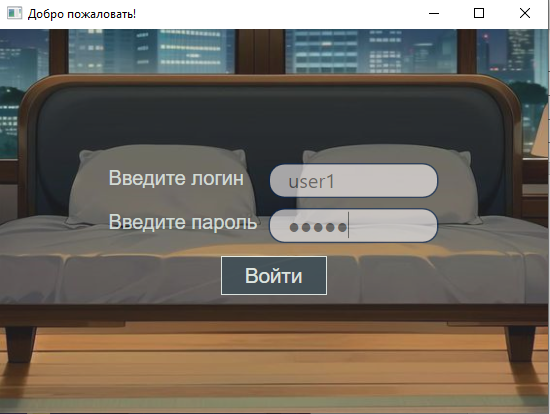


Рисунок 30. Окно авторизации

Страница отображения данных БД:

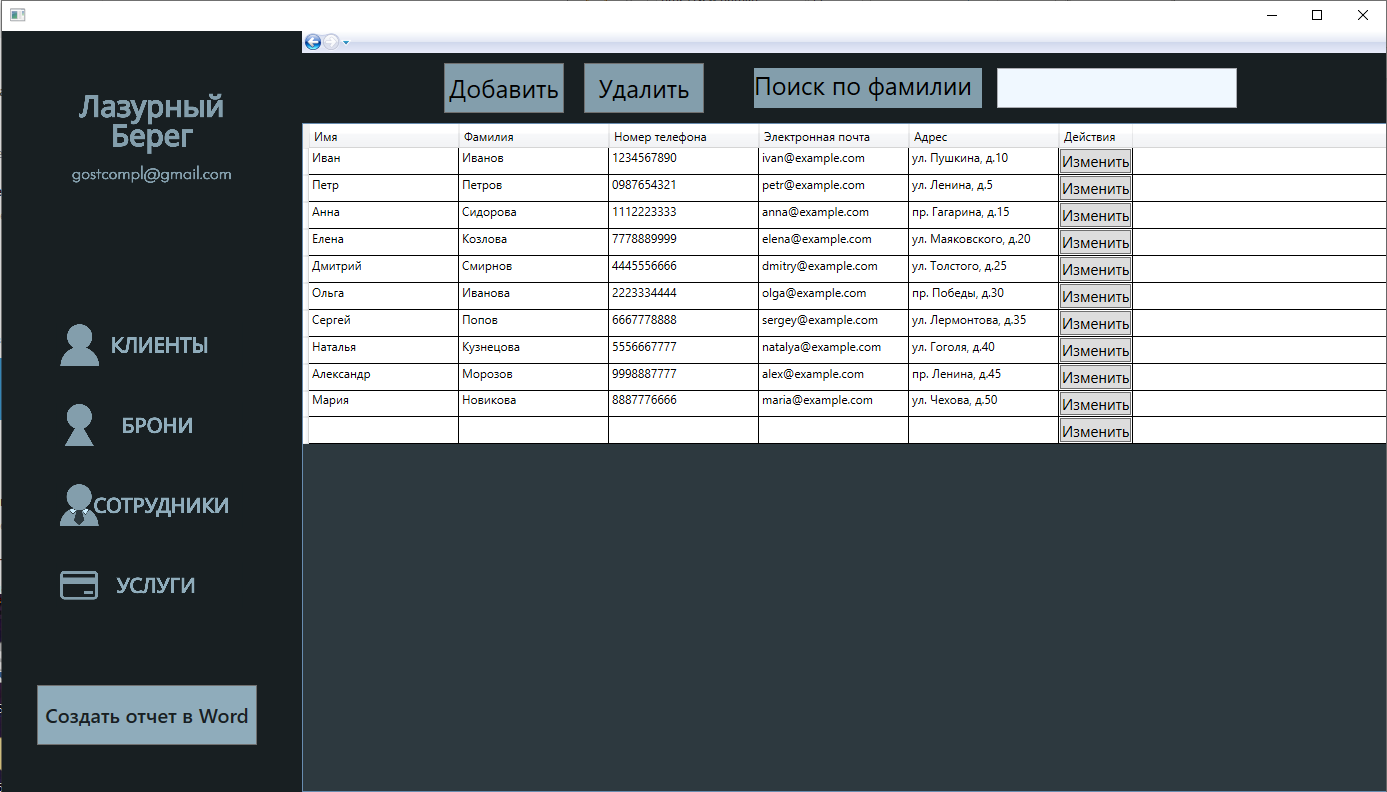


Рисунок 31. Данные таблицы «Клиенты»

Добавление новой записи в таблицу:

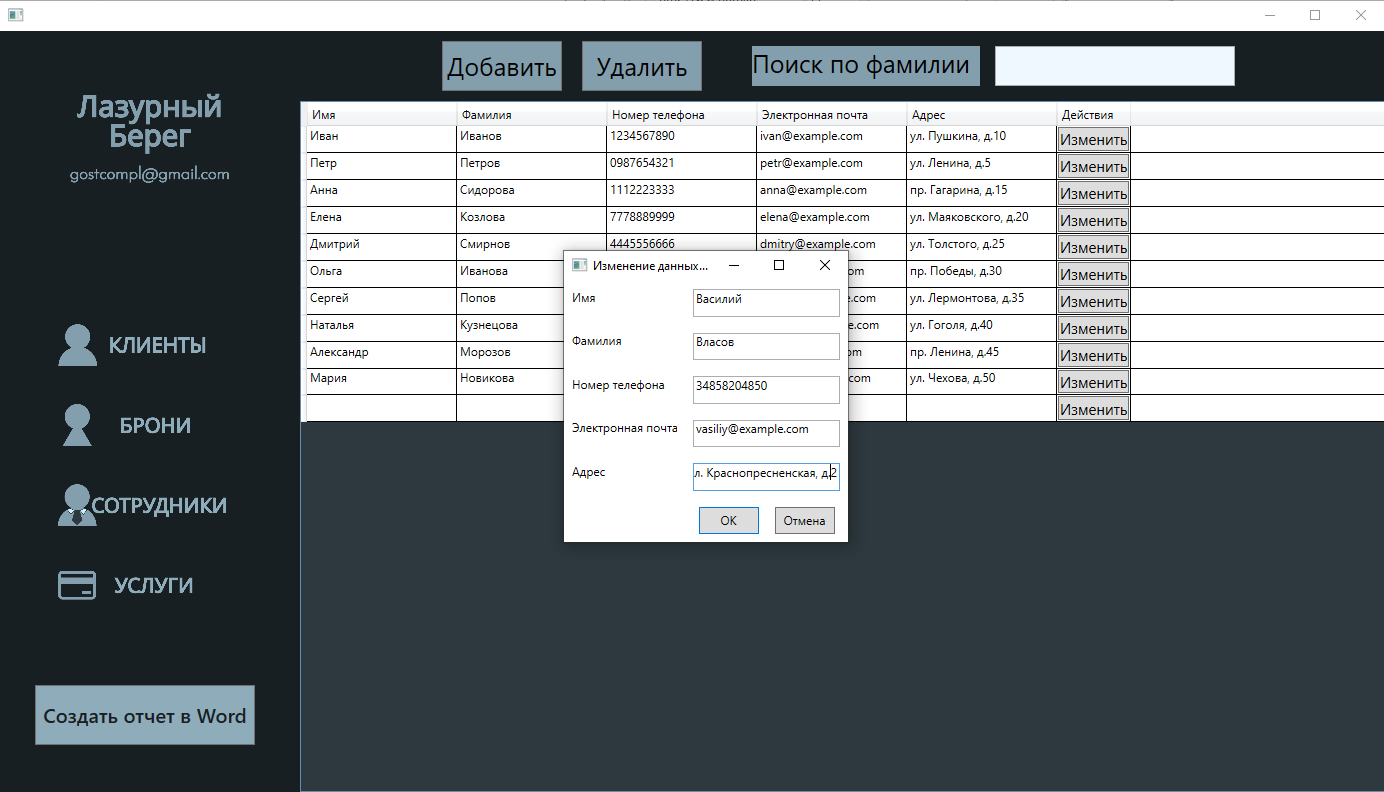


Рисунок 32. Окно добавления новой записи

Результат добавления записи в таблицу:

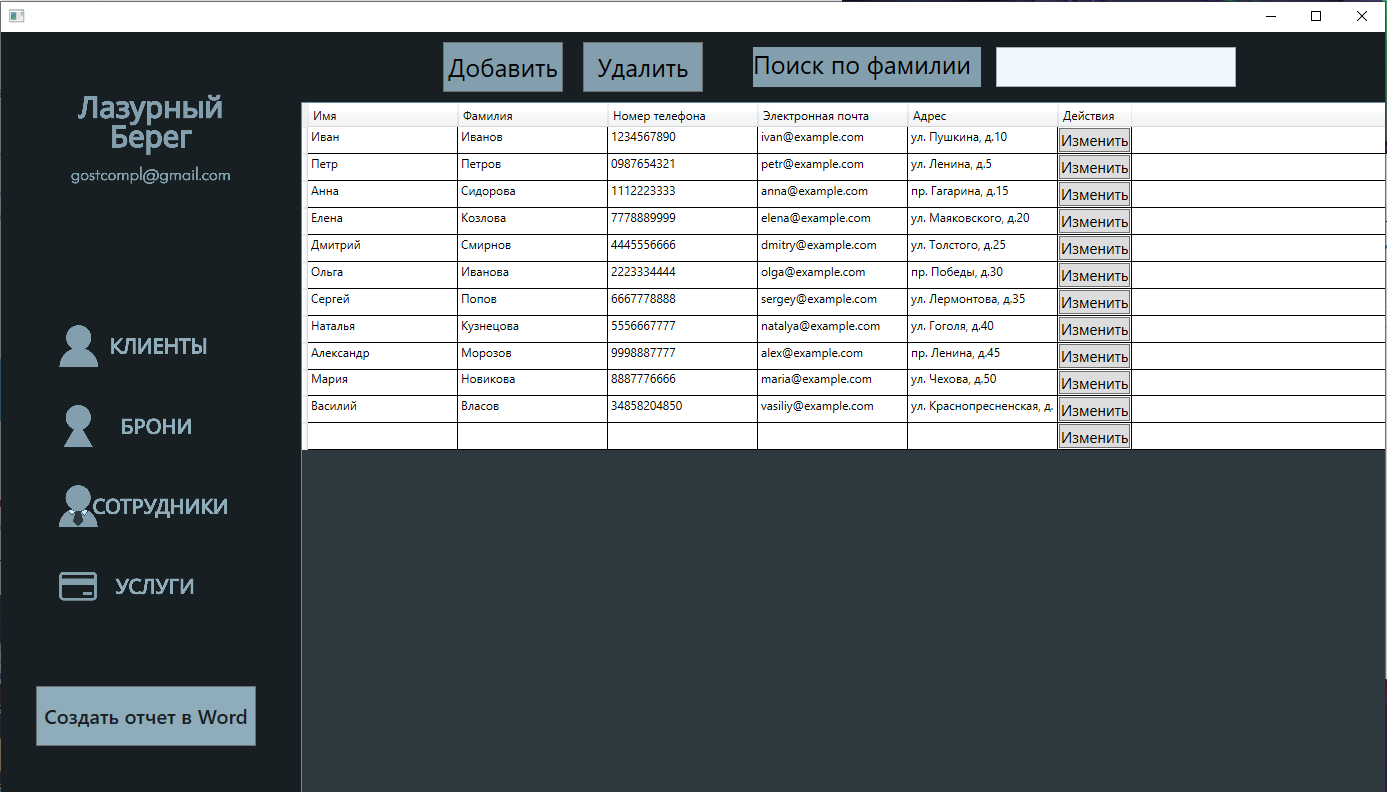


Рисунок 33. Новая запись в таблице

Для удаления записи необходимо ее выделить:

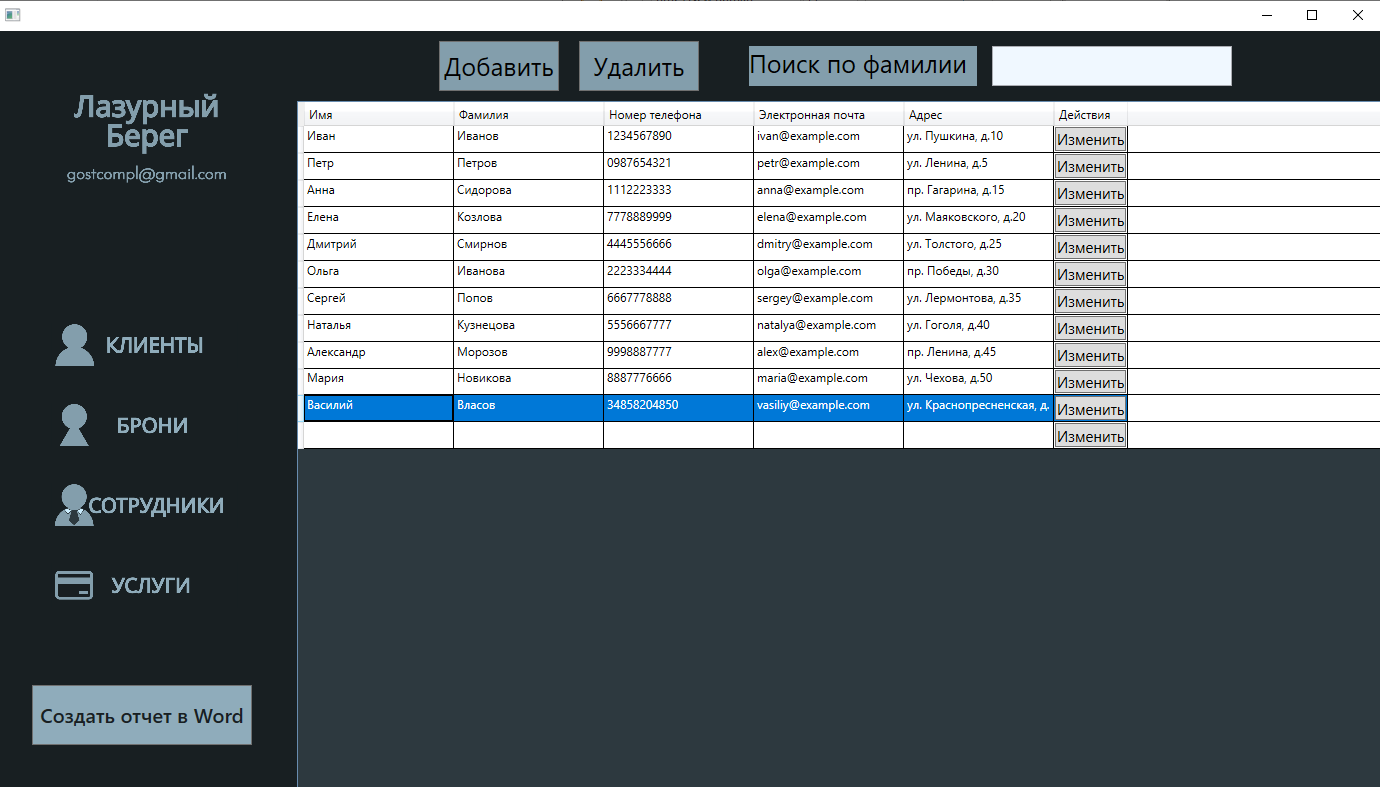


Рисунок 34. Выделенная запись

Для удаления записи нужно нажать кнопку «Удалить»:

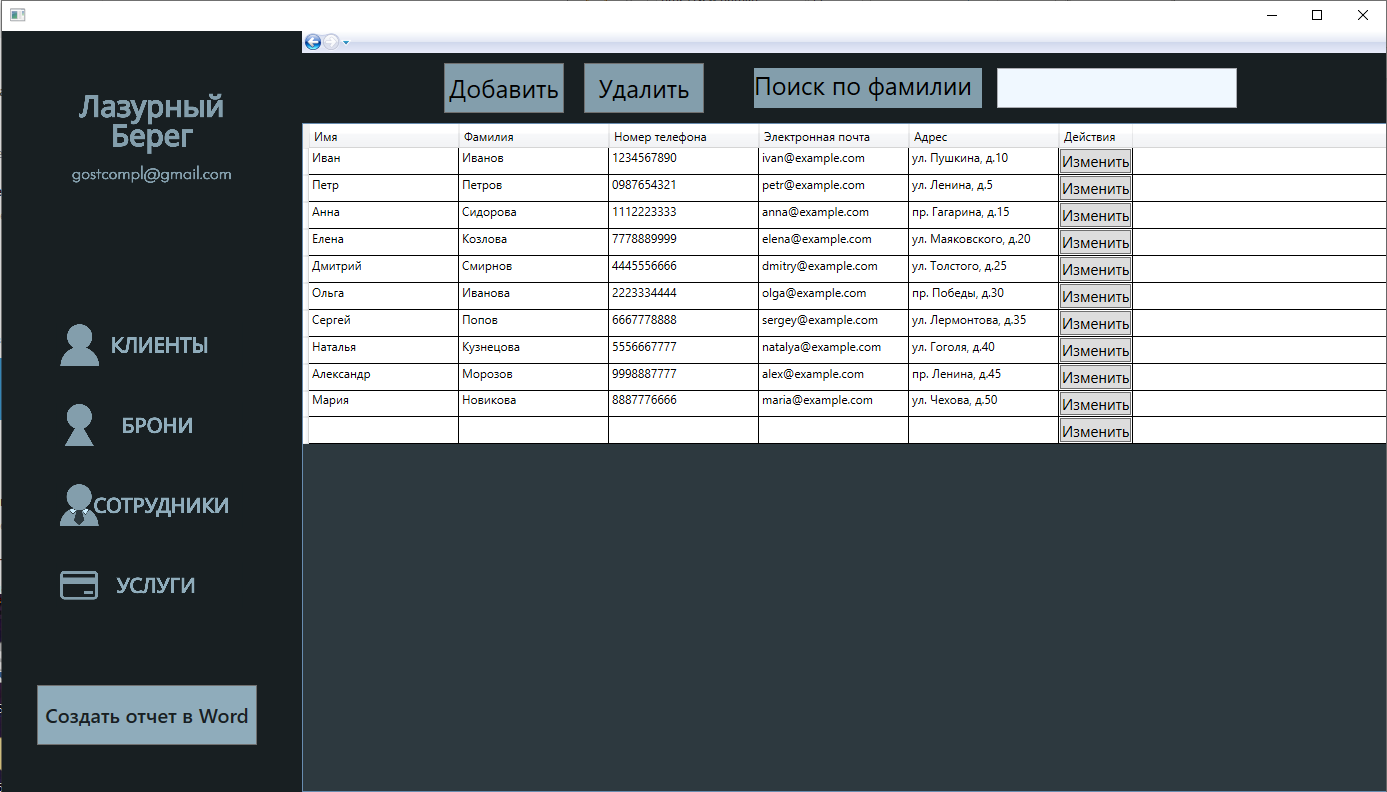


Рисунок 35. Запись удалена

Поиск клиента по его фамилии:

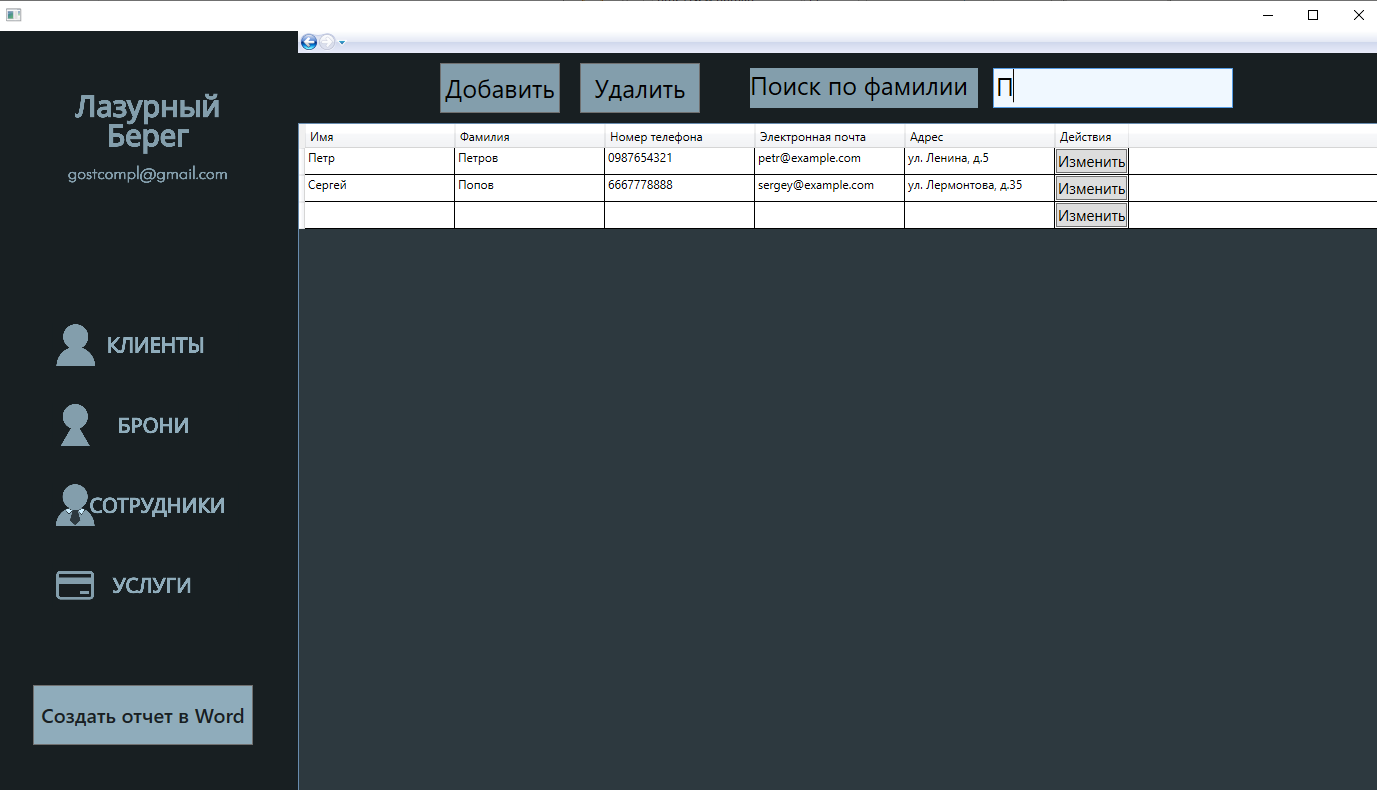


Рисунок 36. Результат поиска

По нажатии на кнопку «Создать отче в Word» на рабочем столе создается документ:



Рисунок 37. Документ, созданный на рабочем столе

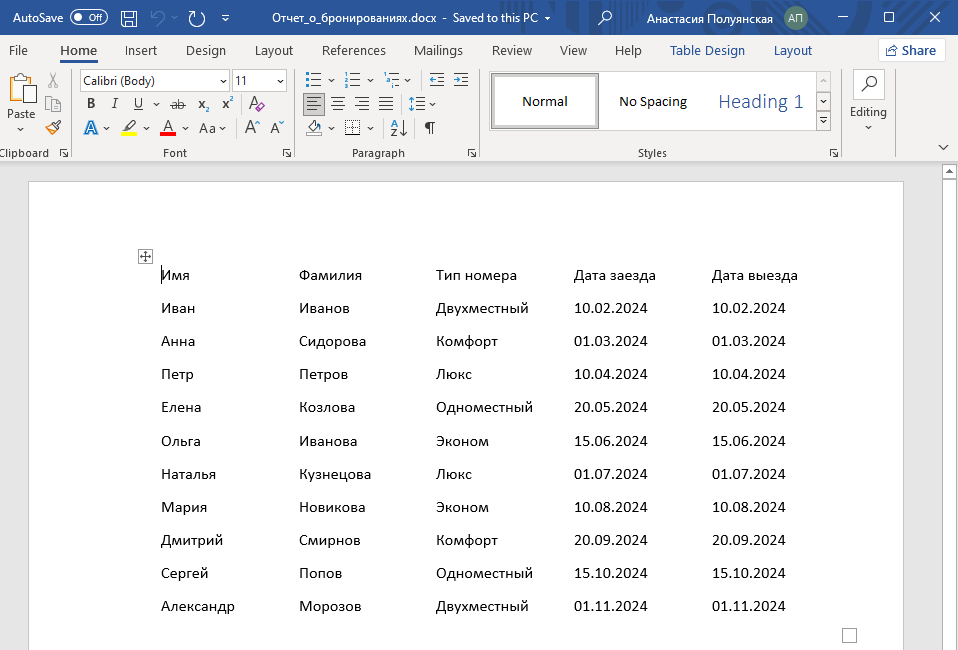


Рисунок 38. Содержимое созданного документа

* 1. **Сообщения пользователю**

Сообщение при вводе неверных значений при авторизации:

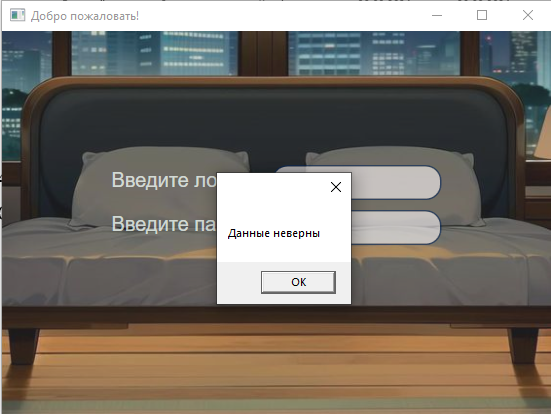


Рисунок 39. Сообщение «Данные неверны»

Сообщение при вводе верных значений при авторизации:

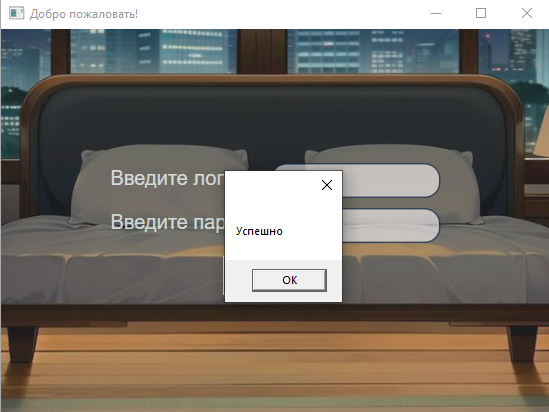


Рисунок 40. Сообщение «Успешно»

Сообщение о сохранении данных в БД:

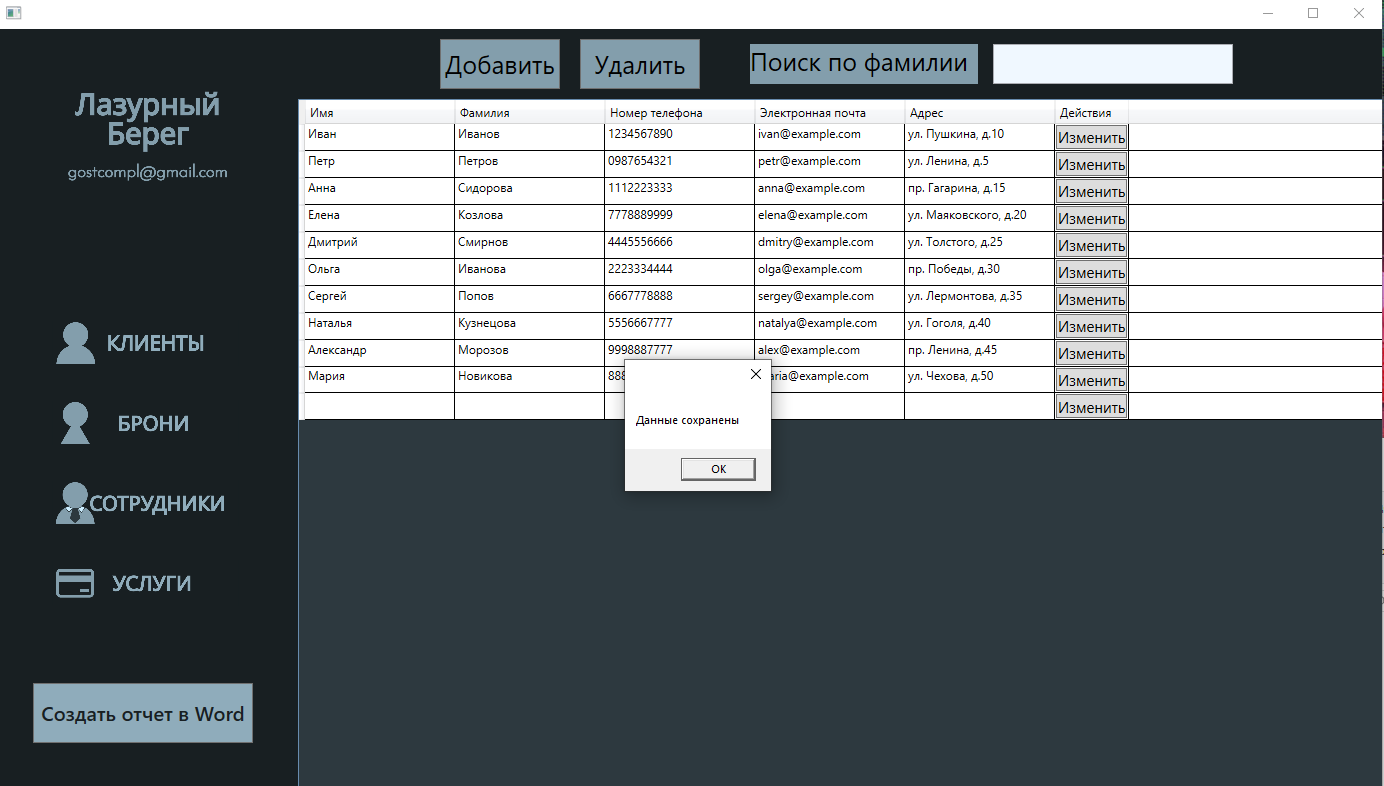


Рисунок 41. Сообщение «Данные сохранены»

Сообщение об удалении данных:

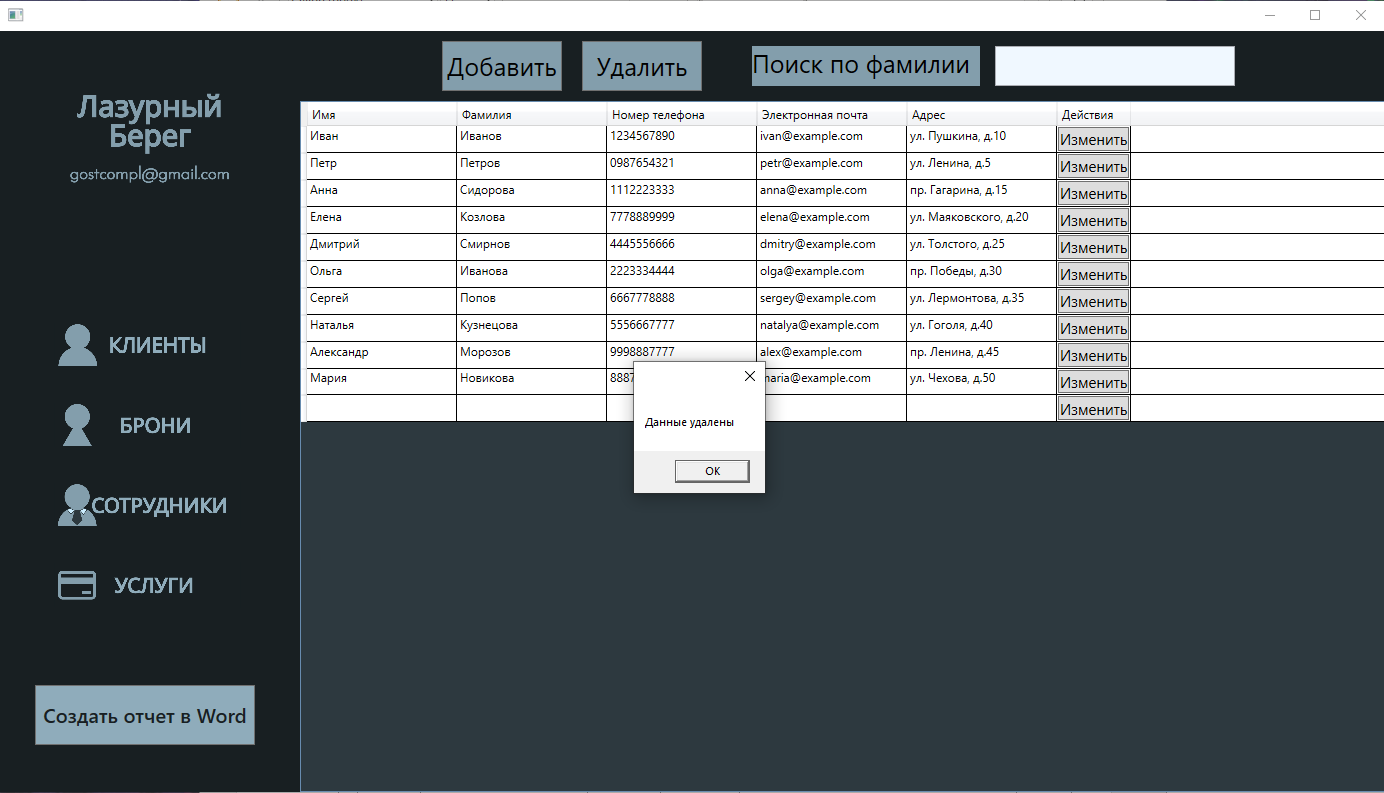


Рисунок 42. Сообщение «Данные удалены»

Ошибка при не выборе данных при добавлении записи в таблицу Бронирования:

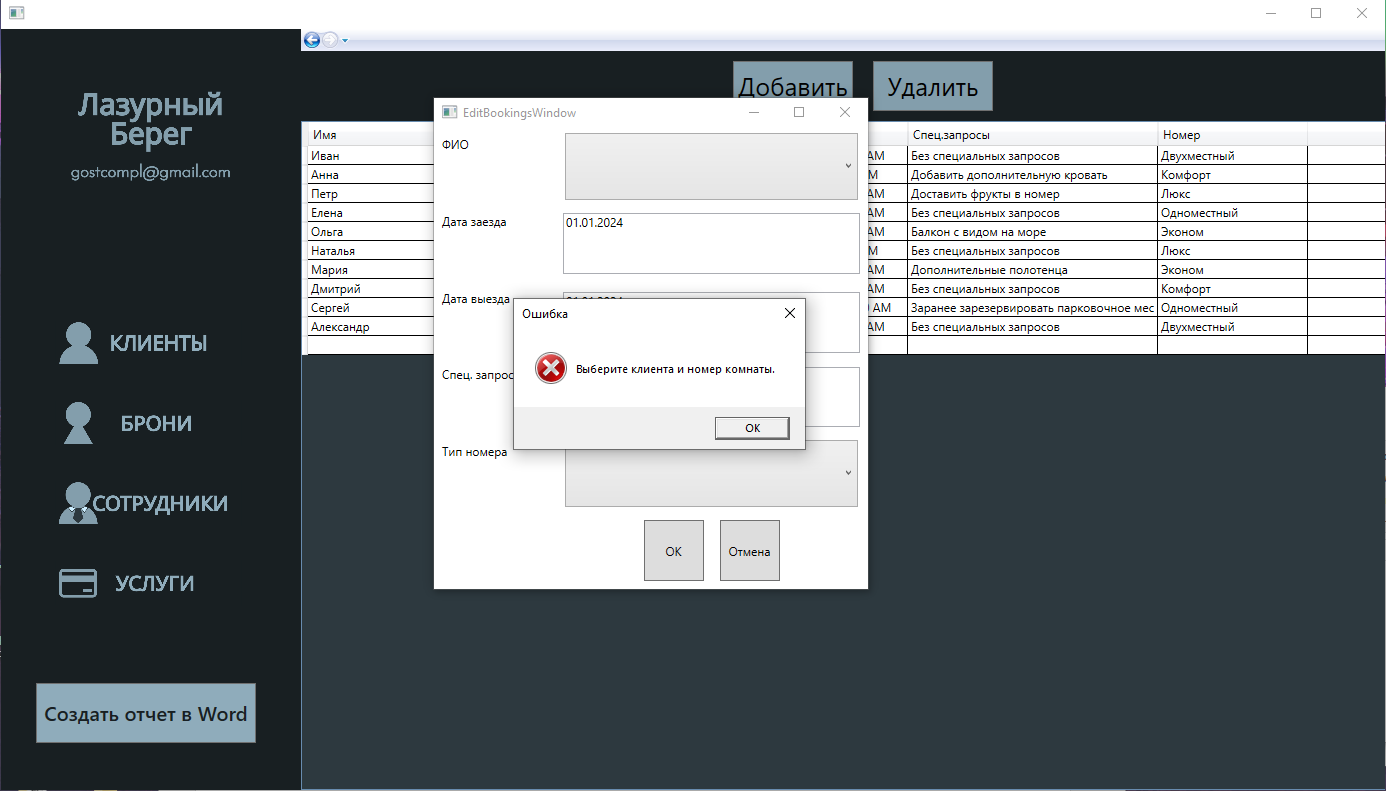


Рисунок 43. Сообщение «Выберите клиента и номер комнаты»

Сообщении о сохранении отчетаWord на рабочем столе:

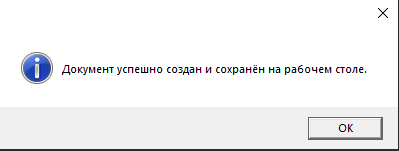
=

Рисунок 44. Сообщение «Документ успешно создан и сохранен на рабочем столе»

1. **Тестирование и отладка программного модуля.**

**Тест-кейс:** **Добавление нового клиента**

**Описание:** Проверка функции добавления нового клиента в систему.

**Шаги**:

1. Запустить приложение "Лазурный Берег".
2. Перейти в раздел "Клиенты".
3. Нажать кнопку "Добавить клиента".
4. Ввести данные нового клиента в соответствующие поля:
   * Имя
   * Фамилия
   * Номер телефона
   * Адрес электронной почты
   * Дополнительные данные (при необходимости)
5. Нажать кнопку "Сохранить".

**Ожидаемый результат:**

* После нажатия кнопки "Сохранить" новый клиент должен быть добавлен в базу данных.
* В списке клиентов должен появиться новый клиент с введенными данными.

**Предусловия:**

* Приложение "Лазурный Берег" запущено и работает корректно.

**Фактический результат:**

* Новый клиент появился в списке клиентов.
* Новый клиент отображается в списке клиентов при последующем открытии раздела "Клиенты".

**Отладка:**

При переходе на страницу бронирований была ошибка загрузки базы данных Бронирования:

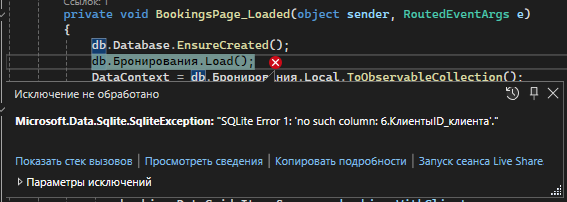


Рисунок 45. Демонстрация ошибки

Ошибка была исправлен добавлением строк с определением внешних ключей в классе Booking.cs:

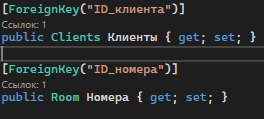


Рисунок 46. Код для исправления ошибки

**Заключение.**

В рамках данного курсового проекта было разработано приложение для администрирования гостиничного комплекса "Лазурный Берег". Целью проекта было создание удобного и эффективного инструмента для управления информацией о клиентах, номерах, бронированиях и других аспектах работы гостиничного бизнеса.

В процессе разработки были выполнены следующие задачи:

* Разработана база данных для хранения информации о клиентах, номерах, бронированиях, сотрудниках и услугах.
* Создано приложение на базе технологии WPF (Windows Presentation Foundation) для обеспечения удобного и интуитивно понятного пользовательского интерфейса.
* Реализованы функциональности добавления, удаления, редактирования и поиска информации о клиентах, номерах и бронированиях.
* Обеспечена интеграция с базой данных для сохранения и получения данных о гостиничном комплексе.

Дальнейшее развитие приложения может включать в себя расширение функциональности, улучшение пользовательского интерфейса, добавление дополнительных возможностей для управления.

**Список литературы:**

1. Федорова Г.Н. Разработка модулей программного обеспечения для компьютерных систем: учебник для студентов СПО /Г.Н. Федорова. 2-е изд., стер. - М.: Издательский центр «Академия»,2018. - 384с.
2. Федорова Г.Н. Осуществление интеграции программных модулей: учебник для студентов СПО / Г.Н. Федорова. 2-е изд., стер. - М.: Издательский центр «Академия»,2018. - 285 с.
3. Царёв, Р.Ю. Алгоритмы и структуры данных (CDIO): учебник /Р.Ю. Царёв, А.В. Прокопенко; Министерство образования и науки Российской Федерации, Сибирский Федеральный университет. - Красноярск: СФУ, 2016 -204 с. : ил. - Библиогр. в кн. - ISBN 978-5-7638-3388-1; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=497016
4. Гниденко, И. Г. Технология разработки программного обеспечения : учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва: Издательство Юрайт, 2020 — 235 с.
5. C# 7.0. Справочник. Полное описание языка.: Пер. с англ. – СпБ.: ООО “Альфакнига”, 2018 – 1024 с. : ил. – Парал. тит. англ.
6. Microsoft Visual C#. Подробное руководство. 8-е изд. – СПб.: Питер, 2017 – 848 с.: ил. – (Серия «Библиотека программиста»).
7. Программирование на C# для начинающих. Основные сведения / Алексей Васильев. – Москва : Эксмо, 2018 – 592 с. – (Российский компьютерный бестселлер).
8. Язык программирования C# 7 и платформы .NET и .NET Core, 8-е изд. : Пер. с англ. – СПб. : ООО “Диалектика”, 2018 – 1328 с. : ил. – Парал. тит. англ.
9. Оппель, Эндрю Дж. SQL. Полное руководство / Оппель Эндрю Дж.. - М.: Диалектика / Вильямс, 2016 - 902 c.

Приложение 1. Текст программы.

Страница BookingPage.

.xaml:

<Page x:Class="WpfApp1.Pages.BookingsPage"

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:WpfApp1.Pages"

mc:Ignorable="d"

d:DesignHeight="800" d:DesignWidth="1400"

Title="BookingsPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="70"></RowDefinition>

<RowDefinition></RowDefinition>

</Grid.RowDefinitions>

<DataGrid x:Name="bookingsDataGrid" Width="1520" Grid.Row="1" ItemsSource="{Binding}" AutoGenerateColumns="False" Background="#2D393F">

<DataGrid.Columns>

<DataGridTextColumn Width="150" Header="Имя" Binding="{Binding Клиенты.Имя}"/>

<DataGridTextColumn Width="150" Header="Фамилия" Binding="{Binding Клиенты.Фамилия}"/>

<DataGridTextColumn Width="150" Header="Дата заезда" Binding="{Binding Дата\_заезда}"/>

<DataGridTextColumn Width="150" Header="Дата выезда" Binding="{Binding Дата\_выезда}"/>

<DataGridTextColumn Width="250" Header="Спец.запросы" Binding="{Binding Специальные\_запросы}"/>

<DataGridTextColumn Width="150" Header="Номер" Binding="{Binding Номера.Тип\_номера}"/>

</DataGrid.Columns>

</DataGrid>

<StackPanel Width="1520" Orientation="Horizontal" HorizontalAlignment="Center">

<Button Content="Добавить" Height="50" Width="120" Background="#839EAC" FontSize="25" Margin="650 10 10 10" Click="Add\_Click" />

<Button Content="Удалить" Height="50" Width="120" Background="#839EAC" FontSize="25" Margin="10" Click="Delete\_Click" />

</StackPanel>

</Grid>

</Page>

.xaml.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using WpfApp1.Classes;

using Microsoft.EntityFrameworkCore;

namespace WpfApp1.Pages

{

public partial class BookingsPage : Page

{

ApplicationContext db = new ApplicationContext();

public BookingsPage()

{

InitializeComponent();

Loaded += BookingsPage\_Loaded;

}

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

{

db.Database.EnsureCreated();

db.Бронирования.Load();

DataContext = db.Бронирования.Local.ToObservableCollection();

try

{

var bookingsWithClients = db.Бронирования

.Include(b => b.Клиенты)

.Include(b => b.Номера)

.ToList();

bookingsDataGrid.ItemsSource = bookingsWithClients;

}

catch (Exception ex)

{

HandleException(ex);

}

}

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

{

using (var db = new ApplicationContext())

{

var clients = db.Клиенты.ToList();

var rooms = db.Номера.ToList();

EditBookingsWindow editBookingsWindow = new EditBookingsWindow(new Booking(), clients, rooms);

if (editBookingsWindow.ShowDialog() == true)

{

Booking booking = editBookingsWindow.Бронирования;

db.Бронирования.Add(booking);

db.SaveChanges();

bookingsDataGrid.ItemsSource = db.Бронирования.ToList();

bookingsDataGrid.Items.Refresh();

}

}

}

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

{

Booking selectedBooking = bookingsDataGrid.SelectedItem as Booking;

if (selectedBooking == null)

return;

using (var db = new ApplicationContext())

{

db.Бронирования.Remove(selectedBooking);

db.SaveChanges();

}

var bookingsList = bookingsDataGrid.ItemsSource as List<Booking>;

if (bookingsList != null)

{

bookingsList.Remove(selectedBooking);

bookingsDataGrid.Items.Refresh();

MessageBox.Show("Данные удалены", "", MessageBoxButton.OK);

}

}

private void HandleException(Exception ex)

{

throw new NotImplementedException();

}

}

}

Страница ClientsPage.

.xaml:

<Page x:Class="WpfApp1.Pages.ClientsPage"

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:WpfApp1.Pages"

mc:Ignorable="d"

d:DesignHeight="800" d:DesignWidth="1400"

Title="ClientsPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="70"></RowDefinition>

<RowDefinition></RowDefinition>

</Grid.RowDefinitions>

<DataGrid Width="1520" Grid.Row="1" x:Name="clientsList" ItemsSource="{Binding}" AutoGenerateColumns="False" Background="#2D393F">

<DataGrid.Columns>

<DataGridTextColumn Width="150" Header="Имя" Binding="{Binding Имя}"/>

<DataGridTextColumn Width="150" Header="Фамилия" Binding="{Binding Фамилия}"/>

<DataGridTextColumn Width="150" Header="Номер телефона" Binding="{Binding Номер\_телефона}"/>

<DataGridTextColumn Width="150" Header="Электронная почта" Binding="{Binding Электронная\_почта}"/>

<DataGridTextColumn Width="150" Header="Адрес" Binding="{Binding Адрес}"/>

<DataGridTemplateColumn

Header="Действия"

Width="Auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnEdit"

Click="BtnEdit\_Click"

Content="Изменить"

FontSize="15"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<StackPanel Width="1400" Orientation="Horizontal" HorizontalAlignment="Center" >

<Button Content="Добавить" Height="50" Width="120" Background="#839EAC" FontSize="25" Margin="300 10 10 10" Click="Add\_Click" />

<Button Content="Удалить" Height="50" Width="120" Background="#839EAC" FontSize="25" Margin="10" Click="Delete\_Click" />

<TextBlock Height="40" Background="#839EAC" FontSize="25" Text="Поиск по фамилии" Margin="40 5 5 5 " Width="228"/>

<TextBox x:Name="txtSearchClient"

Background="AliceBlue"

HorizontalAlignment="Left"

TextChanged="txtSearchClient\_TextChanged"

Width="240"

Height="40"

Margin="10 5 5 5"

FontSize="25"/>

</StackPanel>

</Grid>

</Page>

.xaml.cs:

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using WpfApp1.Classes;

using Microsoft.EntityFrameworkCore;

namespace WpfApp1.Pages

{

public partial class ClientsPage : Page

{

ApplicationContext db = new ApplicationContext();

public ClientsPage()

{

InitializeComponent();

Loaded += ClientsPage\_Loaded;

}

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

{

db.Database.EnsureCreated();

db.Клиенты.Load();

clientsList.ItemsSource = db.Клиенты.Local.ToObservableCollection();

}

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

{

string search = txtSearchClient.Text;

clientsList.ItemsSource = db.Клиенты

.Where(x => x.Фамилия.ToLower().Contains(search))

.ToList();

}

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

{

var button = sender as Button;

var client = button?.DataContext as Clients;

if (client != null)

{

var editClientWindow = new EditClientWindow(client);

editClientWindow.ShowDialog();

}

else

{

EditClientWindow editClientWindow = new EditClientWindow(new Clients());

if (editClientWindow.ShowDialog() == true)

{

Clients clients = editClientWindow.Клиенты;

db.Клиенты.Add(clients);

db.SaveChanges();

clientsList.ItemsSource = db.Клиенты.ToList();

}

}

}

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

{

EditClientWindow editClientWindow = new EditClientWindow(new Clients());

if (editClientWindow.ShowDialog() == true)

{

Clients clients = editClientWindow.Клиенты;

db.Клиенты.Add(clients);

db.SaveChanges();

clientsList.ItemsSource = db.Клиенты.ToList();

}

}

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

{

Clients selectedClient = clientsList.SelectedItem as Clients;

if (selectedClient == null)

return;

db.Клиенты.Remove(selectedClient);

db.SaveChanges();

clientsList.ItemsSource = db.Клиенты.ToList();

MessageBox.Show("Данные удалены", "", MessageBoxButton.OK);

}

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

{

Clients selectedClient = clientsList.SelectedItem as Clients;

if (selectedClient == null)

return;

EditClientWindow editClientWindow = new EditClientWindow(selectedClient);

if (editClientWindow.ShowDialog() == true)

{

Clients editedClient = editClientWindow.Клиенты;

Clients existingClient = db.Клиенты.Find(editedClient.ID\_клиента);

if (existingClient != null)

{

existingClient.Имя = editedClient.Имя;

existingClient.Фамилия = editedClient.Фамилия;

existingClient.Номер\_телефона = editedClient.Номер\_телефона;

existingClient.Электронная\_почта = editedClient.Электронная\_почта;

existingClient.Адрес = editedClient.Адрес;

db.SaveChanges();

clientsList.ItemsSource = db.Клиенты.ToList();

}

}

}

}

}

Окно DiffWindow.

.xaml:

<Window x:Class="WpfApp1.Pages.DiffWindow"

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:WpfApp1.Pages"

mc:Ignorable="d"

Title="" Height="800" Width="1400">

<Grid Background="#181F22">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="300" ></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

</Grid.ColumnDefinitions>

<Frame Grid.Column="1" x:Name="info"/>

<StackPanel>

<Image Source="\Resources\5.png" Height="169" Width="290"

Margin="5 20 5 5"/>

</StackPanel>

<WrapPanel Background="#181F22" VerticalAlignment="Top" HorizontalAlignment="Center" Height="585" Width="280" RenderTransformOrigin="0.5,0.5" Margin="0,199,0,0">

<WrapPanel.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform/>

<RotateTransform Angle="0"/>

<TranslateTransform/>

</TransformGroup>

</WrapPanel.RenderTransform>

<Button Name="clients\_btn" BorderBrush="#181F22" Click="clients\_btn\_click" Width="195" Height="50" Margin="40 90 5 25" Background="#181F22" >

<Image Source="\Resources\1.png" Height="45" Width="263"/>

</Button>

<Button Name="bookings\_btn" BorderBrush="#181F22" Click="bookings\_btn\_click" Width="195" Height="50" Margin="40 5 5 25" Background="#181F22">

<Image Source="\Resources\2.png" Height="45" Width="263"/>

</Button>

<Button Name="empl\_btn" BorderBrush="#181F22" Click="empl\_btn\_click" Width="195" Height="50" Margin="40 5 5 25" Background="#181F22">

<Image Source="\Resources\3.png" Height="45" Width="191"/>

</Button>

<Button Name="serv\_btn" BorderBrush="#181F22" Click="serv\_btn\_click" Width="195" Height="50" Margin="40 5 5 25" Background="#181F22">

<Image Source="\Resources\4.png" Height="45" Width="263"/>

</Button>

<Button FontSize="20" Content="Создать отчет в Word" Click="CreateWordReport\_Click" Width="220" Height="60" Margin="25 50 5 25" Background="#FF8FACBB" FontFamily="Segoe UI Semibold" Foreground="#FF181F22"/>

</WrapPanel>

</Grid>

</Window>

.xaml.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows;

using WpfApp1.Classes;

using Microsoft.Office.Interop.Word;

namespace WpfApp1.Pages

{

public partial class DiffWindow : System.Windows.Window

{

public class BookingInfo

{

public string Имя { get; set; }

public string Фамилия { get; set; }

public string Тип\_номера { get; set; }

public DateTime? Дата\_заезда { get; set; }

public DateTime? Дата\_выезда { get; set; }

}

public List<BookingInfo> GetBookingWithClientAndRoomInfo()

{

using (var db = new ApplicationContext())

{

var bookingsInfo = (from booking in db.Бронирования

join client in db.Клиенты on booking.ID\_клиента equals client.ID\_клиента

join room in db.Номера on booking.ID\_номера equals room.ID\_номера

select new BookingInfo

{

Имя = client.Имя,

Фамилия = client.Фамилия,

Тип\_номера = room.Тип\_номера,

Дата\_заезда = booking.Дата\_заезда,

Дата\_выезда = booking.Дата\_заезда

}).ToList();

return bookingsInfo;

}

}

ApplicationContext db = new ApplicationContext();

private BookingsPage \_viewModel;

public DiffWindow()

{

WindowStartupLocation = WindowStartupLocation.CenterScreen;

InitializeComponent();

\_viewModel = new BookingsPage();

DataContext = \_viewModel;

}

private void clients\_btn\_click(object sender, RoutedEventArgs e)

{

Pages.ClientsPage clientsPage = new Pages.ClientsPage();

info.NavigationService.Navigate(clientsPage);

}

private void bookings\_btn\_click(object sender, RoutedEventArgs e)

{

Pages.BookingsPage bookingsPage = new Pages.BookingsPage();

info.NavigationService.Navigate(bookingsPage);

}

private void empl\_btn\_click(object sender, RoutedEventArgs e)

{

Pages.EmplPage emplPage = new Pages.EmplPage();

info.NavigationService.Navigate(emplPage);

}

private void serv\_btn\_click(object sender, RoutedEventArgs e)

{

Pages.ServicePage emplPage = new Pages.ServicePage();

info.NavigationService.Navigate(emplPage);

}

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

{

List<Booking> bookings = db.Бронирования.ToList();

CreateWordDocument(bookings);

}

public void CreateWordDocument(List<Booking> bookings)

{

string desktopPath = Environment.GetFolderPath(Environment.SpecialFolder.Desktop);

Microsoft.Office.Interop.Word.Application wordApp = new Microsoft.Office.Interop.Word.Application();

Document doc = wordApp.Documents.Add();

Microsoft.Office.Interop.Word.Table table = doc.Tables.Add(doc.Range(), bookings.Count + 1, 5);

table.Cell(1, 1).Range.Text = "Имя";

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

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

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

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

var bookingsInfo = GetBookingWithClientAndRoomInfo();

int rowIndex = 2;

foreach (var bookingInfo in bookingsInfo)

{

table.Cell(rowIndex, 1).Range.Text = bookingInfo.Имя;

table.Cell(rowIndex, 2).Range.Text = bookingInfo.Фамилия;

table.Cell(rowIndex, 3).Range.Text = bookingInfo.Тип\_номера;

table.Cell(rowIndex, 4).Range.Text = bookingInfo.Дата\_заезда.ToString();

table.Cell(rowIndex, 5).Range.Text = bookingInfo.Дата\_выезда.ToString();

rowIndex++;

}

string filename = System.IO.Path.Combine(desktopPath, "Отчет\_о\_бронированиях.docx");

doc.SaveAs2(filename);

MessageBox.Show("Документ успешно создан и сохранён на рабочем столе.", " ", MessageBoxButton.OK, MessageBoxImage.Information);

doc.Close();

wordApp.Quit();

}

}

}

Окно EditBookingsWindow.

.xaml:

<Window x:Class="WpfApp1.Pages.EditBookingsWindow"

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:WpfApp1.Pages"

mc:Ignorable="d"

Title="EditBookingsWindow" Height="500" Width="450">

<Window.Resources>

<Style TargetType="TextBlock">

<Setter Property="Margin" Value="8" />

</Style>

<Style TargetType="TextBox">

<Setter Property="Margin" Value="8" />

</Style>

<Style TargetType="Button">

<Setter Property="MinWidth" Value="60" />

<Setter Property="Margin" Value="8" />

</Style>

</Window.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition />

<RowDefinition />

<RowDefinition />

<RowDefinition />

<RowDefinition />

<RowDefinition />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="121" />

<ColumnDefinition />

</Grid.ColumnDefinitions>

<TextBlock Text="ФИО" Margin="8,8,8,8" />

<TextBlock Text="Дата заезда" Grid.Row="1" Margin="8,8,8,8" />

<TextBlock Text="Дата выезда" Grid.Row="2" Margin="8,8,8,8" />

<TextBlock Text="Спец. запросы" Grid.Row="3" Margin="8,8,8,8" />

<TextBlock Text="Тип номера" Grid.Row="4" Margin="8,8,8,8" />

<TextBox Text="{Binding Бронирования.Дата\_заезда, StringFormat=dd.MM.yyyy}" Grid.Column="1" Grid.Row="1" Margin="8,8,8,8" />

<TextBox Text="{Binding Бронирования.Дата\_выезда, StringFormat=dd.MM.yyyy}" Grid.Column="1" Grid.Row="2" Margin="8,10,8,6" />

<TextBox Text="{Binding Бронирования.Специальные\_запросы}" Grid.Column="1" Grid.Row="3" Margin="8,8,8,8" />

<ComboBox Name="CmbName"

ItemsSource="{Binding Клиенты}"

SelectedValue="{Binding Бронирования.ID\_клиента}"

DisplayMemberPath="ИмяФамилия"

Margin="10,5,10,5"

Grid.Column="1"/>

<ComboBox Grid.Row="4"

Name="CmbRoom"

ItemsSource="{Binding Номера}"

SelectedValue="{Binding Бронирования.ID\_номера}"

DisplayMemberPath="Тип\_номера"

Margin="10,5,10,5"

Grid.Column="2"/>

<StackPanel HorizontalAlignment="Center" Orientation="Horizontal" Grid.Row="5" Grid.Column="1" Width="152">

<Button IsDefault="True" Click="Accept\_Click" >OK</Button>

<Button IsCancel="True" >Отмена</Button>

</StackPanel>

</Grid>

</Window>

.xaml.cs:

using System.Collections.Generic;

using System.Windows;

using WpfApp1.Classes;

namespace WpfApp1.Pages

{

public partial class EditBookingsWindow : Window

{

public Booking Бронирования { get; private set; }

public List<Clients> Клиенты { get; set; }

public List<Room> Номера { get; set; }

public EditBookingsWindow(Booking booking, List<Clients> clients, List<Room> rooms)

{

InitializeComponent();

Бронирования = booking;

Клиенты = clients;

Номера = rooms;

DataContext = this;

WindowStartupLocation = WindowStartupLocation.CenterScreen;

}

void Accept\_Click(object sender, RoutedEventArgs e)

{

Clients selectedClient = (Clients)CmbName.SelectedItem;

Room selectedRoom = (Room)CmbRoom.SelectedItem;

if (selectedClient == null || selectedRoom == null)

{

MessageBox.Show("Выберите клиента и номер комнаты.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

Бронирования.ID\_клиента = selectedClient.ID\_клиента;

Бронирования.ID\_номера = selectedRoom.ID\_номера;

DialogResult = true;

MessageBox.Show("Данные сохранены", "", MessageBoxButton.OK);

}

}

}

Окно EditClientWindow.

.xaml:

<Window x:Class="WpfApp1.Pages.EditClientWindow"

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:WpfApp1.Pages"

mc:Ignorable="d"

Title="Изменение данных клиента" Height="300" Width="300">

<Window.Resources>

<Style TargetType="TextBlock">

<Setter Property="Margin" Value="8" />

</Style>

<Style TargetType="TextBox">

<Setter Property="Margin" Value="8" />

</Style>

<Style TargetType="Button">

<Setter Property="MinWidth" Value="60" />

<Setter Property="Margin" Value="8" />

</Style>

</Window.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition />

<RowDefinition />

<RowDefinition />

<RowDefinition />

<RowDefinition />

<RowDefinition />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="121" />

<ColumnDefinition />

</Grid.ColumnDefinitions>

<TextBlock Text="Имя" Margin="8,8,8,8" />

<TextBlock Text="Фамилия" Grid.Row="1" Margin="8,8,8,8" />

<TextBlock Text="Номер телефона" Grid.Row="2" Margin="8,8,8,8" />

<TextBlock Text="Электронная почта" Grid.Row="3" Margin="8,8,8,8" />

<TextBlock Text="Адрес" Grid.Row="4" Margin="8,8,8,8" />

<TextBox Text="{Binding Имя}" Grid.Column="1" Margin="8,8,8,8" />

<TextBox Text="{Binding Фамилия}" Grid.Column="1" Grid.Row="1" Margin="8,8,8,8" />

<TextBox Text="{Binding Номер\_телефона}" Grid.Column="1" Grid.Row="2" Margin="8,8,8,8" />

<TextBox Text="{Binding Электронная\_почта}" Grid.Column="1" Grid.Row="3" Margin="8,8,8,8" />

<TextBox Text="{Binding Адрес}" Grid.Column="1" Grid.Row="4" Margin="8,8,8,8" />

<StackPanel HorizontalAlignment="Center" Orientation="Horizontal" Grid.Row="5" Grid.Column="1" Width="152">

<Button IsDefault="True" Click="Accept\_Click" >OK</Button>

<Button IsCancel="True" >Отмена</Button>

</StackPanel>

</Grid>

</Window>

.xaml.cs:

using System.Windows;

using WpfApp1.Classes;

namespace WpfApp1.Pages

{

public partial class EditClientWindow : Window

{

public Clients Клиенты { get; private set; }

public EditClientWindow(Clients clients)

{

InitializeComponent();

Клиенты = clients;

DataContext = Клиенты;

WindowStartupLocation = WindowStartupLocation.CenterScreen;

}

void Accept\_Click(object sender, RoutedEventArgs e)

{

DialogResult = true;

MessageBox.Show("Данные сохранены", "", MessageBoxButton.OK);

}

}

}

Окно EditEmplWindow.

.xaml:

<Window x:Class="WpfApp1.Pages.EditEmplWindow"

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:WpfApp1.Pages"

mc:Ignorable="d"

Title="Изменение данных сотрудника" Height="300" Width="300">

<Window.Resources>

<Style TargetType="TextBlock">

<Setter Property="Margin" Value="8" />

</Style>

<Style TargetType="TextBox">

<Setter Property="Margin" Value="8" />

</Style>

<Style TargetType="Button">

<Setter Property="MinWidth" Value="60" />

<Setter Property="Margin" Value="8" />

</Style>

</Window.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition />

<RowDefinition />

<RowDefinition />

<RowDefinition />

<RowDefinition />

<RowDefinition />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="121" />

<ColumnDefinition />

</Grid.ColumnDefinitions>

<TextBlock Text="Имя" Margin="8,8,8,8" />

<TextBlock Text="Фамилия" Grid.Row="1" Margin="8,8,8,8" />

<TextBlock Text="Должность" Grid.Row="2" Margin="8,8,8,8" />

<TextBlock Text="Зар. плата" Grid.Row="3" Margin="8,8,8,8" />

<TextBlock Text="Контактная инф." Grid.Row="4" Margin="8,8,8,8" />

<TextBox Text="{Binding Имя}" Grid.Column="1" Margin="8,8,8,8" />

<TextBox Text="{Binding Фамилия}" Grid.Column="1" Grid.Row="1" Margin="8,8,8,8" />

<TextBox Text="{Binding Должность}" Grid.Column="1" Grid.Row="2" Margin="8,8,8,8" />

<TextBox Text="{Binding Заработная\_плата}" Grid.Column="1" Grid.Row="3" Margin="8,8,8,8" />

<TextBox Text="{Binding Контактная\_информация}" Grid.Column="1" Grid.Row="4" Margin="8,8,8,8" />

<StackPanel HorizontalAlignment="Center" Orientation="Horizontal" Grid.Row="5" Grid.Column="1" Width="152">

<Button IsDefault="True" Click="Accept\_Click" >OK</Button>

<Button IsCancel="True" >Отмена</Button>

</StackPanel>

</Grid>

</Window>

.xaml.cs:

using System.Windows;

using WpfApp1.Classes;

namespace WpfApp1.Pages

{

public partial class EditEmplWindow : Window

{

public Employees Сотрудники { get; private set; }

public EditEmplWindow(Employees employees)

{

InitializeComponent();

Сотрудники = employees;

DataContext = Сотрудники;

WindowStartupLocation = WindowStartupLocation.CenterScreen;

}

void Accept\_Click(object sender, RoutedEventArgs e)

{

DialogResult = true;

MessageBox.Show("Данные сохранены", "", MessageBoxButton.OK);

}

}

}

Окно EditServWindow.

.xaml:

<Window x:Class="WpfApp1.Pages.EditServWindow"

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:WpfApp1.Pages"

mc:Ignorable="d"

Title="EditServWindow" Height="300" Width="300">

<Window.Resources>

<Style TargetType="TextBlock">

<Setter Property="Margin" Value="8" />

</Style>

<Style TargetType="TextBox">

<Setter Property="Margin" Value="8" />

</Style>

<Style TargetType="Button">

<Setter Property="MinWidth" Value="60" />

<Setter Property="Margin" Value="8" />

</Style>

</Window.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition />

<RowDefinition />

<RowDefinition />

<RowDefinition />

<RowDefinition />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="85" />

<ColumnDefinition />

</Grid.ColumnDefinitions>

<TextBlock Text="Услуга" Margin="8,8,8,8" />

<TextBlock Text="Описание" Grid.Row="1" Margin="8,8,8,8" />

<TextBlock Text="Стоимость" Grid.Row="2" Margin="8,8,8,8" />

<TextBlock Text="Доступность" Grid.Row="3" Margin="8,8,8,8" />

<TextBox Text="{Binding Название\_услуги}" Grid.Column="1" Margin="8,8,8,8" />

<TextBox Text="{Binding Описание}" Grid.Column="1" Grid.Row="1" Margin="8,8,8,8" />

<TextBox Text="{Binding Стоимость}" Grid.Column="1" Grid.Row="2" Margin="8,8,8,8" />

<TextBox Text="{Binding Доступность}" Grid.Column="1" Grid.Row="3" Margin="8,8,8,8" />

<StackPanel HorizontalAlignment="Center" Orientation="Horizontal" Grid.Row="5" Grid.Column="1" Width="152">

<Button IsDefault="True" Click="Accept\_Click" >OK</Button>

<Button IsCancel="True" >Отмена</Button>

</StackPanel>

</Grid>

</Window>

.xaml.cs:

using System.Windows;

using WpfApp1.Classes;

namespace WpfApp1.Pages

{

public partial class EditServWindow : Window

{

public Service Услуги { get; private set; }

public EditServWindow(Service service)

{

InitializeComponent();

Услуги = service;

DataContext = Услуги;

WindowStartupLocation = WindowStartupLocation.CenterScreen;

}

void Accept\_Click(object sender, RoutedEventArgs e)

{

DialogResult = true;

MessageBox.Show("Данные сохранены", "", MessageBoxButton.OK);

}

}

}

Страница EmplPage.

.xaml:

<Page x:Class="WpfApp1.Pages.EmplPage"

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:WpfApp1.Pages"

mc:Ignorable="d"

d:DesignHeight="800" d:DesignWidth="1400"

Title="EmplPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="70"></RowDefinition>

<RowDefinition></RowDefinition>

</Grid.RowDefinitions>

<DataGrid Width="1520" Grid.Row="1" x:Name="emplList" ItemsSource="{Binding}" AutoGenerateColumns="False" Background="#2D393F">

<DataGrid.Columns>

<DataGridTextColumn Width="150" Header="Имя" Binding="{Binding Имя}"/>

<DataGridTextColumn Width="150" Header="Фамилия" Binding="{Binding Фамилия}"/>

<DataGridTextColumn Width="150" Header="Должность" Binding="{Binding Должность}"/>

<DataGridTextColumn Width="150" Header="Заработная плата" Binding="{Binding Заработная\_плата}"/>

<DataGridTextColumn Width="150" Header="Контактная информация" Binding="{Binding Контактная\_информация}"/>

<DataGridTemplateColumn

Header="Действия"

Width="Auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Name="BtnEdit"

Click="BtnEdit\_Click"

Content="Изменить"

FontSize="15"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<StackPanel Width="1400" Orientation="Horizontal" HorizontalAlignment="Center">

<Button Content="Добавить" Height="50" Width="120" Background="#839EAC" FontSize="25" Margin="300 10 10 10" Click="Add\_Click" />

<Button Content="Удалить" Height="50" Width="120" Background="#839EAC" FontSize="25" Margin="10 10 10 10" Click="Delete\_Click" />

<TextBlock Height="40" Background="#839EAC" FontSize="25" Text="Поиск по фамилии" Margin="40 5 5 5"/>

<TextBox x:Name="txtSearchEmpl"

Background="AliceBlue"

HorizontalAlignment="Left"

TextChanged="txtSearchEmpl\_TextChanged"

Width="240"

Height="40"

Margin="10 5 5 5"

FontSize="25"/>

</StackPanel>

</Grid>

</Page>

.xaml.cs:

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using WpfApp1.Classes;

using Microsoft.EntityFrameworkCore;

namespace WpfApp1.Pages

{

public partial class EmplPage : Page

{

ApplicationContext db = new ApplicationContext();

public EmplPage()

{

InitializeComponent();

Loaded +=EmplPage\_Loaded;

}

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

{

db.Database.EnsureCreated();

db.Сотрудники.Load();

DataContext = db.Сотрудники.Local.ToObservableCollection();

}

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

{

string search = txtSearchEmpl.Text;

emplList.ItemsSource = db.Сотрудники

.Where(x => x.Фамилия.ToLower().Contains(search))

.ToList();

}

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

{

var button = sender as Button;

var empl = button?.DataContext as Employees;

if (empl != null)

{

var editClientWindow = new EditEmplWindow(empl);

editClientWindow.ShowDialog();

}

else

{

EditEmplWindow editEmplWindow = new EditEmplWindow(new Employees());

if (editEmplWindow.ShowDialog() == true)

{

Employees employees = editEmplWindow.Сотрудники;

using (var db = new ApplicationContext())

{

db.Сотрудники.Add(employees);

db.SaveChanges();

emplList.ItemsSource = db.Сотрудники.ToList();

}

emplList.Items.Refresh();

}

}

}

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

{

EditEmplWindow editEmplWindow = new EditEmplWindow(new Employees());

if (editEmplWindow.ShowDialog() == true)

{

Employees employees = editEmplWindow.Сотрудники;

using (var db = new ApplicationContext())

{

db.Сотрудники.Add(employees);

db.SaveChanges();

emplList.ItemsSource = db.Сотрудники.ToList();

}

emplList.Items.Refresh();

}

}

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

{

Employees selectedempl = emplList.SelectedItem as Employees;

if (selectedempl == null)

return;

using (var db = new ApplicationContext())

{

db.Сотрудники.Remove(selectedempl);

db.SaveChanges();

emplList.ItemsSource = db.Сотрудники.ToList();

}

emplList.Items.Refresh();

MessageBox.Show("Данные удалены", "", MessageBoxButton.OK);

}

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

{

Employees selectedempl = emplList.SelectedItem as Employees;

if (selectedempl == null)

return;

EditEmplWindow editEmplWindow = new EditEmplWindow(selectedempl);

if (editEmplWindow.ShowDialog() == true)

{

using (var db = new ApplicationContext())

{

Employees editedEmpl = editEmplWindow.Сотрудники;

Employees existingEmpl = db.Сотрудники.Find(editedEmpl.ID\_сотрудника);

if (existingEmpl != null)

{

existingEmpl.Имя = editedEmpl.Имя;

existingEmpl.Фамилия = editedEmpl.Фамилия;

existingEmpl.Должность = editedEmpl.Должность;

existingEmpl.Заработная\_плата = editedEmpl.Заработная\_плата;

existingEmpl.Контактная\_информация = editedEmpl.Контактная\_информация;

db.SaveChanges();

emplList.Items.Refresh();

}

}

}

}

}

}

Страница ServicePage.

.xaml:

<Page x:Class="WpfApp1.Pages.ServicePage"

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:WpfApp1.Pages"

mc:Ignorable="d"

d:DesignHeight="800" d:DesignWidth="1400"

Title="ServicePage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="70"></RowDefinition>

<RowDefinition></RowDefinition>

</Grid.RowDefinitions>

<DataGrid Width="1520" Grid.Row="1" x:Name="servList" ItemsSource="{Binding}" AutoGenerateColumns="False" Background="#2D393F">

<DataGrid.Columns>

<DataGridTextColumn Width="150" Header="услуга" Binding="{Binding Название\_услуги}"/>

<DataGridTextColumn Width="150" Header="Описание" Binding="{Binding Описание}"/>

<DataGridTextColumn Width="150" Header="Стоимость" Binding="{Binding Стоимость}"/>

<DataGridTextColumn Width="150" Header="Доступность" Binding="{Binding Доступность}"/>

<DataGridTemplateColumn

Header="Действия"

Width="Auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnEdit"

Click="Edit\_Click"

Content="Изменить"

FontSize="15"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<StackPanel Width="284" Orientation="Horizontal" HorizontalAlignment="Center" >

<Button Content="Добавить" Height="50" Width="120" Background="#839EAC" FontSize="25" Margin="10" Click="Add\_Click" />

<Button Content="Удалить" Height="50" Width="120" Background="#839EAC" FontSize="25" Margin="10" Click="Delete\_Click" />

</StackPanel>

</Grid>

</Page>

.xaml.cs:

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using WpfApp1.Classes;

using Microsoft.EntityFrameworkCore;

namespace WpfApp1.Pages

{

public partial class ServicePage : Page

{

ApplicationContext db = new ApplicationContext();

public ServicePage()

{

InitializeComponent();

Loaded += ServPage\_Loaded;

}

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

{

db.Database.EnsureCreated();

db.Услуги.Load();

DataContext = db.Услуги.Local.ToObservableCollection();

}

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

{

EditServWindow editServWindow = new EditServWindow(new Service());

if (editServWindow.ShowDialog() == true)

{

Service service = editServWindow.Услуги;

using (var db = new ApplicationContext())

{

db.Услуги.Add(service);

db.SaveChanges();

servList.ItemsSource = db.Услуги.ToList();

}

servList.Items.Refresh();

}

}

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

{

Service selectedServ = servList.SelectedItem as Service;

if (selectedServ == null)

return;

using (var db = new ApplicationContext())

{

db.Услуги.Remove(selectedServ);

db.SaveChanges();

servList.ItemsSource = db.Услуги.ToList();

}

servList.Items.Refresh();

MessageBox.Show("Данные удалены", "", MessageBoxButton.OK);

}

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

{

Service selectedServ = servList.SelectedItem as Service;

if (selectedServ == null)

return;

EditServWindow editServWindow = new EditServWindow(selectedServ);

if (editServWindow.ShowDialog() == true)

{

using (var db = new ApplicationContext())

{

Service editedServ = editServWindow.Услуги;

Service existingServ = db.Услуги.Find(editedServ.ID\_услуги);

if (existingServ != null)

{

existingServ.Название\_услуги = editedServ.Название\_услуги;

existingServ.Описание = editedServ.Описание;

existingServ.Стоимость = editedServ.Стоимость;

existingServ.Доступность = editedServ.Доступность;

db.SaveChanges();

servList.Items.Refresh();

}

}

}

}

}

}

Окно MainWidow.

.xaml:

<Window x:Class="WpfApp1.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:WpfApp1"

mc:Ignorable="d"

Title="Добро пожаловать!" Height="423" Width="565">

<Grid >

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid.Background>

<ImageBrush ImageSource="\Resources\lg.jpg" Opacity="0.9"/>

</Grid.Background>

<StackPanel Orientation="Vertical"

VerticalAlignment="Center" Margin="76,0,76,0" Height="160" Grid.ColumnSpan="2">

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Center">

<TextBlock Text="Введите логин" FontFamily="Arial"

Opacity="0.75"

FontSize="20"

Foreground="#F5FFFA"

Margin="5 25 5 5" Width="150" Height="30"/>

<Border BorderBrush="#052659" Margin="5 22 5 5" CornerRadius="15" BorderThickness="1.5" Background="#FFFAFA" Opacity="0.6" Height="35" Width="170">

<TextBox x:Name="textBoxLogin" Height="33" FontSize="20" Background="#FFFAFA" BorderThickness="0" Opacity="0.6" Margin="14,0,14,0" Padding="2" BorderBrush="#FFFAFA"/>

</Border>

</StackPanel>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Center">

<TextBlock Text="Введите пароль" FontFamily="Arial"

FontSize="20"

Opacity="0.75"

Foreground="#F5FFFA"

Margin="5" Width="150" Height="30"/>

<Border BorderBrush="#052659" Margin="5 5 5 5" CornerRadius="15" BorderThickness="1.5" Background="#FFFAFA" Opacity="0.6" Height="35" Width="170">

<PasswordBox x:Name="textBoxPassword" Height="33" FontSize="20" BorderThickness="0" Background="#FFFAFA" Opacity="0.6" Margin="14,0,14,0" Padding="2" BorderBrush="#FFFAFA"/>

</Border>

</StackPanel>

<Button

Content="Войти" FontFamily="Arial"

x:Name="enter"

Width="106"

Height="39"

Margin="8"

Background="#3d4d55"

Foreground="#F5FFFA"

Opacity="0.83"

BorderThickness="1"

BorderBrush="#F5FFFA"

FontSize="20.5"

Click="Button\_Auth\_Click">

</Button>

</StackPanel>

</Grid>

</Window>

.xaml.cs:

using System.Linq;

using System.Windows;

using System.Windows.Media;

using WpfApp1.Classes;

using WpfApp1.Pages;

namespace WpfApp1

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

}

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

{

string Login = textBoxLogin.Text.Trim();

string Pass = textBoxPassword.Password.Trim();

if (Login.Length < 1)

{

textBoxLogin.ToolTip = "Данные некорректны";

textBoxLogin.Background = Brushes.Red;

}

else if (Pass.Length < 1)

{

textBoxPassword.ToolTip = "Данные некорректны";

textBoxPassword.Background = Brushes.Red;

}

else

{

textBoxLogin.ToolTip = "";

textBoxLogin.Background = Brushes.Transparent;

textBoxPassword.ToolTip = "";

textBoxPassword.Background = Brushes.Transparent;

User auth = null;

using (ApplicationContext db = new ApplicationContext())

{

auth = db.Users.Where(b => b.Login == Login && b.Password == Pass).FirstOrDefault();

}

if (auth != null)

{

MessageBox.Show("Успешно");

DiffWindow newWindow = new DiffWindow();

newWindow.Show();

this.Close();

}

else

{

MessageBox.Show("Данные неверны");

}

}

}

}

}

Классы контекста:

* 1. using Microsoft.EntityFrameworkCore;

namespace WpfApp1.Classes

{

public class ApplicationContext : DbContext

{

public DbSet<User> Users { get; set; }

public DbSet<Clients> Клиенты { get; set; }

public DbSet<Booking> Бронирования { get; set; }

public DbSet<Employees> Сотрудники { get; set; }

public DbSet<Room> Номера { get; set; }

public DbSet<Service> Услуги { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseSqlite("Data Source=hotel.db");

}

}

}

* 1. using System;

using System.ComponentModel.DataAnnotations.Schema;

using System.ComponentModel.DataAnnotations;

namespace WpfApp1.Classes

{

[Table("Бронирования")]

public class Booking

{

[Key]

public int ID\_бронирования { get; set; }

public int ID\_клиента { get; set; }

public int ID\_номера { get; set; }

public DateTime Дата\_заезда { get; set; }

public DateTime Дата\_выезда { get; set; }

public string Специальные\_запросы { get; set; }

[ForeignKey("ID\_клиента")]

public Clients Клиенты { get; set; }

[ForeignKey("ID\_номера")]

public Room Номера { get; set; }

public Booking() { }

public Booking(int ID\_клиента, int ID\_номера, DateTime Дата\_заезда, DateTime Дата\_выезда, string Специальные\_запросы)

{

this.ID\_клиента = ID\_клиента;

this.ID\_номера = ID\_номера;

this.Дата\_заезда = Дата\_заезда;

this.Дата\_выезда = Дата\_выезда;

this.Специальные\_запросы = Специальные\_запросы;

}

}

}

* 1. using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace WpfApp1.Classes

{

[Table("Клиенты")]

public class Clients

{

public string ИмяФамилия

{

get { return $"{Имя} {Фамилия}"; }

}

[Key]

public int ID\_клиента { get; set; }

public string Имя { get; set; }

public string Фамилия{ get; set; }

public string Номер\_телефона{ get; set; }

public string Электронная\_почта{ get; set; }

public string Адрес{ get; set; }

public Clients() { }

public Clients(string Имя, string Фамилия, string Номер\_телефона, string Электронная\_почта, string Адрес)

{

this.Имя = Имя;

this.Фамилия = Фамилия;

this.Номер\_телефона = Номер\_телефона;

this.Электронная\_почта = Электронная\_почта;

this.Адрес = Адрес;

}

}

}

* 1. using System.ComponentModel.DataAnnotations.Schema;

using System.ComponentModel.DataAnnotations;

namespace WpfApp1.Classes

{

[Table("Сотрудники")]

public class Employees

{

[Key]

public int ID\_сотрудника { get; set; }

public string Имя { get; set; }

public string Фамилия { get; set; }

public string Должность { get; set; }

public double Заработная\_плата { get; set; }

public string Контактная\_информация { get; set; }

public Employees() { }

public Employees(string Имя, string Фамилия, string Должность, double Заработная\_плата, string Контактная\_информация)

{

this.Имя = Имя;

this.Фамилия = Фамилия;

this.Должность = Должность;

this.Заработная\_плата = Заработная\_плата;

this.Контактная\_информация = Контактная\_информация;

}

}

}

* 1. using System.ComponentModel.DataAnnotations.Schema;

using System.ComponentModel.DataAnnotations;

namespace WpfApp1.Classes

{

[Table("Номера")]

public class Room

{

[Key]

public int ID\_номера { get; set; }

public string Тип\_номера { get; set; }

public double Стоимость\_за\_ночь { get; set; }

public string Доступность { get; set; }

public Room() { }

public Room(string Тип\_номера, double Стоимость\_за\_ночь, string Доступность)

{

this.Тип\_номера = Тип\_номера;

this.Стоимость\_за\_ночь = Стоимость\_за\_ночь;

this.Доступность = Доступность;

}

}

}

* 1. using System.ComponentModel.DataAnnotations.Schema;

using System.ComponentModel.DataAnnotations;

namespace WpfApp1.Classes

{

[Table("Услуги")]

public class Service

{

[Key]

public int ID\_услуги { get; set; }

public string Название\_услуги { get; set; }

public string Описание { get; set; }

public double Стоимость { get; set; }

public string Доступность { get; set; }

public Service() { }

public Service(string Название\_услуги, string Описание, double Стоимость, string Доступность)

{

this.Название\_услуги = Название\_услуги;

this.Описание = Описание;

this.Стоимость = Стоимость;

this.Доступность = Доступность;

}

}

}

* 1. using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace WpfApp1.Classes

{

[Table("Users")]

public class User

{

[Key]

public int ID { get; set; }

private string login, password;

public string Login

{

get { return login; }

set { login = value; }

}

public string Password

{

get { return password; }

set { password = value; }

}

public User() { }

public User(string login, string password)

{

this.login = login;

this.password = password;

}

}

}