Приложение А

Министерство образования и науки Нижегородской области

Государственное бюджетное профессиональное

образовательное учреждение

«Заволжский автомоторный техникум»

**Листинг**

Автоматизированной информационной

системы «Гостиница» на базе платформы .NET Framework

Разработала:

студентка группы ПС-20Б

Мухина Е.Е.

г. Заволжье

2022– 2023 уч. год

Листинг программного продукта представлен в электронном виде и хранится в папке Мухина Е.Е. в количестве печатных страниц.

Форма Авторизации

using System;  
using System.Windows;  
using Hotel.classes;  
using MySql.Data.MySqlClient;  
  
namespace Hotel.pages  
{  
 /// <summary>  
 /// Interaction logic for Authorization.xaml  
 /// </summary>  
 public partial class Authorization  
 {  
 public Authorization()  
 {  
 InitializeComponent();  
 }  
  
 //закрытие формы  
 private void CloseBut\_OnClick(object sender, RoutedEventArgs e)  
 {  
 MessageBoxResult askExit = MessageBox.Show("Вы хотите выйти из приложения \"Гостиница\"?", "ВЫХОД ИЗ ПРИЛОЖЕНИЯ", MessageBoxButton.YesNo, MessageBoxImage.Question);  
 if (askExit == MessageBoxResult.Yes)  
 {  
 BackUpSql.CreateBackUp(DateTime.Now.ToString("dd-MM-yyyy HH-mm-ss"));  
 Application.Current.Shutdown();  
 }  
 }  
  
 //кнопка авторизации, проверка введенных данных  
 private void LogInBut\_OnClick(object sender, RoutedEventArgs e)  
 {  
 if(Login.Text == "" && (Passwd.Password == "" || RepeatPasswd.Password == ""))  
 MessageBox.Show("Введите логин и пароль!", "ОШИБКА ЗАПОЛНЕНИЯ", MessageBoxButton.OK, MessageBoxImage.Warning);  
 else if(Login.Text == "")  
 MessageBox.Show("Введите логин!", "ОШИБКА ЗАПОЛНЕНИЯ", MessageBoxButton.OK, MessageBoxImage.Warning);  
 else if(Login.Text != "" && Passwd.Password == "" || RepeatPasswd.Password == "")  
 MessageBox.Show("Введите пароль!", "ОШИБКА ЗАПОЛНЕНИЯ", MessageBoxButton.OK, MessageBoxImage.Warning);  
 else  
 {  
 if(Passwd.Password != RepeatPasswd.Password)  
 MessageBox.Show("Пароль не совпадает!", "ОШИБКА ЗАПОЛНЕНИЯ", MessageBoxButton.OK, MessageBoxImage.Warning);  
 else  
 {  
 string login = Login.Text;  
 string passwd = GetHashPasswd.GetHash(Passwd.Password);  
   
 string logInQuery = string.Format("SELECT \* FROM hotel.user WHERE u\_login = '{0}' AND u\_password = '{1}';", login, passwd);  
 try  
 {  
 classes.MySql.Connection.Open();  
  
 MySqlCommand command = new MySqlCommand(logInQuery, classes.MySql.Connection);  
 var checkUser = command.ExecuteScalar();  
 if (checkUser == null)  
 MessageBox.Show("Неверно введен логин или пароль!", "ОШИБКА АВТОРИЗАЦИИ", MessageBoxButton.OK, MessageBoxImage.Error);  
 else  
 {  
 MySqlDataReader reader = command.ExecuteReader();  
 while (reader.Read())  
 {  
 WhichUserType.UserType = reader.GetString(4);  
 }  
 MessageBox.Show("Вы успешно авторизовались.\nДобро пожаловать!", "УСПЕШНАЯ АВТОРИЗАЦИЯ", MessageBoxButton.OK, MessageBoxImage.Information);  
   
 NavigationService.Navigate(new MainPage());  
 }  
 classes.MySql.Connection.Close();  
 }  
 catch (Exception ex)  
 {  
 MessageBox.Show(ex.Message, "ОШИБКА", MessageBoxButton.OK, MessageBoxImage.Error);  
 }  
 }  
 }  
 }  
 }  
}

Главная форма

using System;

using System.Collections.Generic;

using System.IO;

using System.Windows;

using System.Windows.Controls;

using Hotel.classes;

using MySql.Data.MySqlClient;

namespace Hotel.pages

{

/// <summary>

/// Interaction logic for MainPage.xaml

/// </summary>

public partial class MainPage : Page

{

public MainPage()

{

InitializeComponent();

}

//закрытие приложения, создание Backup

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

{

MessageBoxResult AskExit = MessageBox.Show("Вы хотите выйти из приложения \"Гостиница\"?", "ВЫХОД ИЗ ПРИЛОЖЕНИЯ", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (AskExit == MessageBoxResult.Yes)

{

BackUpSql.CreateBackUp(DateTime.Now.ToString("dd-MM-yyyy HH-mm-ss"));

Application.Current.Shutdown();

}

}

//кнопка "Выход из аккаунта"

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

{

var result = MessageBox.Show("Хотите выйти из аккаунта?", "ВЫХОД ИЗ АККАУНТА", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

NavigationService.Navigate(new Authorization());

}

}

//кнопка панели администратора

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

{

Panel.IsOpen = true;

}

//загрузка формы

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

{

classes.MySql.Connection.Close();

ArrivalDate.SelectedDate = DateTime.Now;

DepartureDate.SelectedDate = DateTime.Now.AddDays(1);

if (WhichUserType.UserType == "user")

{

AddUserBut.Visibility = Visibility.Hidden;

AdminPanel.Visibility = Visibility.Hidden;

Text\_.Text = "АИС Гостиница";

}

LoadData("SELECT \* FROM hotel.room;");

}

//загрузка данных из базы

private void LoadData(string query)

{

try

{

using (classes.MySql.Connection)

{

classes.MySql.Connection.Open();

using (MySqlCommand command = new MySqlCommand(query, classes.MySql.Connection))

{

List<Rooms> rooms = new List<Rooms>();

using (MySqlDataReader reader = command.ExecuteReader())

{

while (reader.Read())

{

rooms.Add(new Rooms { Name = reader.GetString(1), Area = Convert.ToInt32(reader.GetString(2)), VisCount = Convert.ToInt32(reader.GetString(3)), Cost = Convert.ToInt32(reader.GetString(4)), Availble = Convert.ToInt32(reader.GetString(5)), Image = String.Format(Directory.GetCurrentDirectory() + "\\images\\{0}", reader.GetString(6)) });

}

RoomsList.ItemsSource = rooms;

}

}

}

}

catch (Exception ex)

{

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

}

}

//поиск

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

{

LoadData(GetQuery());

}

//получение запроса в базу данных

private string GetQuery()

{

string sql = String.Empty;

if (SearchBox.Text != "") sql = String.Format("SELECT \* FROM hotel.room WHERE r\_name LIKE '%{0}%'", SearchBox.Text);

else sql = "SELECT \* FROM hotel.room";

switch (FilterBox.SelectedIndex)

{

case 0:

sql += " ORDER BY cost";

break;

case 1:

sql += " ORDER BY visitors\_quant";

break;

case 2:

sql += " ORDER BY available\_quant";

break;

default:

sql += " ORDER BY cost";

break;

}

switch (SortBox.SelectedIndex)

{

case 0:

sql += " DESC";

break;

case 1:

sql += " ASC";

break;

}

return sql += ";";

}

private void \_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

LoadData(GetQuery());

}

//кнопка "Добавление пользователя"

private void AddUserBut\_OnClick(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new AddUser());

}

//кнопка "Клиенты"

private void ClientBut\_OnClick(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new ClientPage());

}

//кнопка "Оформить/Бронировать"

private void But\_OnClick(object sender, RoutedEventArgs e)

{

if (ArrivalDate.SelectedDate > DepartureDate.SelectedDate)

MessageBox.Show("Некорректная дата заезда", "ОШИБКА ЗАПОЛНЕНИЯ",

MessageBoxButton.OK, MessageBoxImage.Warning);

else

{

if (NumberOfVisit.SelectedItem != null)

{

ForBooking.DataArriv = ArrivalDate.SelectedDate.ToString();

ForBooking.DataDepart = DepartureDate.SelectedDate.ToString();

ForBooking.NumVis = int.Parse(NumberOfVisit.Text);

NavigationService.Navigate(new Booking());

}

else

MessageBox.Show("Выберите количество посетителей", "ОШИБКА ЗАПОЛНЕНИЯ",

MessageBoxButton.OK,MessageBoxImage.Information);

}

}

//кнопка "Номера"

private void RoomBut\_OnClick(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new RoomPage());

}

//контроль даты заезда

private void ArrivalDate\_OnSelectedDateChanged(object sender, SelectionChangedEventArgs e)

{

if (ArrivalDate.SelectedDate > DateTime.Now) But.Content = "Бронировать";

else But.Content = "Оформить";

}

//контроль даты выезда

private void DepartureDate\_OnSelectedDateChanged(object sender, SelectionChangedEventArgs e)

{

if (DepartureDate.SelectedDate < DateTime.Now) DepartureDate.SelectedDate = DateTime.Now.AddDays(1);

}

//кнопка "Список всех записей"

private void ViewBookBut\_OnClick(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new AllBookingPage());

}

}

}

Форма отображения бронирований

using System;

using System.Collections.Generic;

using System.Data;

using System.Windows;

using System.Windows.Input;

using Hotel.classes;

using MySql.Data.MySqlClient;

namespace Hotel.pages

{

public partial class AllBookingPage

{

//переменные для пагинации

private int \_curPage = 1, \_allPages = 0, \_limit = 2, \_allLines = 0, \_startLine = 0;

public AllBookingPage()

{

InitializeComponent();

}

//загрузка формы

private void AllBookingPage\_OnLoaded(object sender, RoutedEventArgs e)

{

string q = $"SELECT reg\_num, arrival\_date, departure\_date, (select surname from hotel.client where hotel.client.IDclient = booking.IDclient) as resident, (select r\_name from hotel.room where room.NUMroom = booking.NUMroom) as room, (select area from hotel.room where room.NUMroom = booking.NUMroom) as area,(select cost from hotel.room where room.NUMroom = booking.NUMroom) as cost, count\_day, (select f\_name from hotel.feed where feed.IDfeed = booking.IDfeed) as feed, amount FROM hotel.booking LIMIT {\_limit} OFFSET {\_startLine};";

LoadData(q);

\_allLines = CountRow();

\_allPages = (int)Math.Ceiling((double)\_allLines / (double)\_limit);

AllPage.Text = \_allPages.ToString();

}

//загрузка данных из базы, отображение в ItemsControl

private void LoadData(string query)

{

try

{

using (classes.MySql.Connection)

{

classes.MySql.Connection.Open();

using (MySqlCommand command = new MySqlCommand(query, classes.MySql.Connection))

{

List<Reservation> reservation = new List<Reservation>();

using (MySqlDataReader reader = command.ExecuteReader())

{

while (reader.Read())

{

reservation.Add(new Reservation { RegNum = reader.GetString(0),

ArrivalDate = reader.GetString(1).Substring(0, 10),

DepartureDate = reader.GetString(2).Substring(0, 10),

Client = reader.GetString(3), RoomName = reader.GetString(4),

RoomArea = reader.GetString(5), RoomCost = reader.GetString(6),

Feed = reader.GetString(8), Amount = reader.GetString(9)

});

}

ReservationList.ItemsSource = reservation;

}

}

}

}

catch (Exception e)

{

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

}

}

//получение количество строк в базе данных

private int CountRow()

{

int lines = 0;

using (classes.MySql.Connection)

{

DataTable dt = new DataTable();

classes.MySql.Connection.Open();

using (MySqlCommand command =

new MySqlCommand("SELECT \* FROM hotel.booking", classes.MySql.Connection))

{

using (MySqlDataReader reader = command.ExecuteReader())

{

dt.Load(reader);

lines = dt.Rows.Count;

}

}

}

return lines;

}

//закрытие формы

private void CloseBut\_OnClick(object sender, RoutedEventArgs e)

{

MessageBoxResult result = MessageBox.Show("Хотите вернуться на главное меню?", "ВЫХОД", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

NavigationService.Navigate(new MainPage());

}

}

//элемент пагинации, переход на предыдущую страницу

private void PreviousPage\_OnMouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

if (\_curPage > 1)

{

\_curPage -= 1;

CurrentPage.Text = \_curPage.ToString();

\_startLine -= 2;

string q =

$"SELECT reg\_num, arrival\_date, departure\_date, (select surname from hotel.client where hotel.client.IDclient = booking.IDclient) as resident, (select r\_name from hotel.room where room.NUMroom = booking.NUMroom) as room, (select area from hotel.room where room.NUMroom = booking.NUMroom) as area,(select cost from hotel.room where room.NUMroom = booking.NUMroom) as cost, count\_day, (select f\_name from hotel.feed where feed.IDfeed = booking.IDfeed) as feed, amount FROM hotel.booking LIMIT {\_limit} OFFSET {\_startLine};";

LoadData(q);

}

}

//элемент пагинации, переход на следующую страницу

private void NextPage\_OnMouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

if (\_curPage < \_allPages)

{

\_curPage += 1;

CurrentPage.Text = \_curPage.ToString();

\_startLine += 2;

string q =

$"SELECT reg\_num, arrival\_date, departure\_date, (select surname from hotel.client where hotel.client.IDclient = booking.IDclient) as resident, (select r\_name from hotel.room where room.NUMroom = booking.NUMroom) as room, (select area from hotel.room where room.NUMroom = booking.NUMroom) as area,(select cost from hotel.room where room.NUMroom = booking.NUMroom) as cost, count\_day, (select f\_name from hotel.feed where feed.IDfeed = booking.IDfeed) as feed, amount FROM hotel.booking LIMIT {\_limit} OFFSET {\_startLine};";

LoadData(q);

}

}

}

}

Форма Бронирования

using System;

using System.Globalization;

using System.IO;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Input;

using Hotel.classes;

using MySql.Data.MySqlClient;

using word = Microsoft.Office.Interop.Word;

namespace Hotel.pages

{

public partial class Booking

{

//переменные для создания документа Word

private readonly string \_clientProfile = Directory.GetCurrentDirectory() + @"\template\ClientProfile.docx";

private readonly string \_booking = Directory.GetCurrentDirectory() + @"\template\BookingConfirmation.docx";

private readonly string \_cheque = Directory.GetCurrentDirectory() + @"\template\Cheque.docx";

//ID

private double \_totalCost, \_foodCost;

private double \_roomCost, \_numDays;

private string \_nameRoom = string.Empty;

private int \_idfeed, \_idclient, \_numroom, \_age;

public Booking()

{

InitializeComponent();

}

//замена слов в документе

private void ReplaceWordStub(string stubToReplace, string text, word.Document wordDocument)

{

var range = wordDocument.Content;

range.Find.ClearFormatting();

range.Find.Execute(FindText: stubToReplace, ReplaceWith: text);

}

#region Формирование документа "Анкета клиента"

private void ClientProfBut\_OnMouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

string surname = Surname.Text;

string name = \_Name.Text;

string patr = Patr.Text;

string phoneNumber = PhoneBox.Text;

string seriesPas = PasSeries.Text, numberPas = PasNumber.Text;

string date = DateTime.Now.ToShortDateString();

string birthDay = Birthday.Text;

string arDate = ForBooking.DataArriv.Substring(0, 10);

string depDate = ForBooking.DataDepart.Substring(0, 10);

string issueDate = DateIssue.Text;

try

{

var wordApp = new word.Application();

wordApp.Visible = false;

var wordDoc = wordApp.Documents.Open(\_clientProfile);

ReplaceWordStub("<surname>", surname, wordDoc);

ReplaceWordStub("<name>", name, wordDoc);

ReplaceWordStub("<patr>", patr, wordDoc);

ReplaceWordStub("<birthday>", birthDay, wordDoc);

ReplaceWordStub("<phonenumber>", phoneNumber, wordDoc);

ReplaceWordStub("<seria>", seriesPas, wordDoc);

ReplaceWordStub("<number>", numberPas, wordDoc);

ReplaceWordStub("<date>", date, wordDoc);

ReplaceWordStub("<arrivaldate>", arDate, wordDoc);

ReplaceWordStub("<departuredate>", depDate, wordDoc);

ReplaceWordStub("<dateissue>", issueDate, wordDoc);

wordApp.Visible = true;

}

catch(Exception ex)

{

MessageBox.Show(ex.Message, "ОШИБКА");

}

}

#endregion

//закрытие формы

private void CloseBut\_OnClick(object sender, RoutedEventArgs e)

{

MessageBoxResult result = MessageBox.Show("Вернуться на главное в меню?", "ВЫХОД", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

NavigationService.Navigate(new MainPage());

}

}

#region Контроль ввода

private void FIO\_OnPreviewTextInput(object sender, TextCompositionEventArgs e)

{

if (!(Char.IsLetter(e.Text, 0)))

e.Handled = true;

}

private void PhoneBox\_OnPreviewTextInput(object sender, TextCompositionEventArgs e)

{

if (!(Char.IsDigit(e.Text, 0) && PhoneBox.Text.Length < 11))

e.Handled = true;

}

private void Date\_OnPreviewTextInput(object sender, TextCompositionEventArgs e)

{

if (Birthday.Text.Length >= 10 || (!(Char.IsDigit(e.Text, 0)) && e.Text != "."))

e.Handled = true;

}

private void Date1\_OnPreviewTextInput(object sender, TextCompositionEventArgs e)

{

if (DateIssue.Text.Length >= 10 || (!(Char.IsDigit(e.Text, 0)) && e.Text != "."))

e.Handled = true;

}

private void PasSeries\_OnPreviewTextInput(object sender, TextCompositionEventArgs e)

{

if (PasSeries.Text.Length >= 4 || !(Char.IsDigit(e.Text, 0)))

e.Handled = true;

}

private void PasNumber\_OnPreviewTextInput(object sender, TextCompositionEventArgs e)

{

if (PasNumber.Text.Length >= 6 || !(Char.IsDigit(e.Text, 0)))

e.Handled = true;

}

#endregion

//загрузка формы

private void Booking\_OnLoaded(object sender, RoutedEventArgs e)

{

ArrivalDate.SelectedDate = DateTime.Parse(ForBooking.DataArriv);

DepartureDate.SelectedDate = DateTime.Parse(ForBooking.DataDepart);

NumOfVis.Text = ForBooking.NumVis.ToString();

\_numDays = GetDays(DateTime.Parse(ForBooking.DataArriv), DateTime.Parse(ForBooking.DataDepart));

Days.Text = GetDays(DateTime.Parse(ForBooking.DataArriv), DateTime.Parse(ForBooking.DataDepart)).ToString(CultureInfo.CurrentCulture);

FillRoomBox(ForBooking.NumVis);

}

//заполнение элемента ComboBox

private void FillRoomBox(int numOfVis)

{

string fillQuery = $"SELECT \* FROM hotel.room WHERE visitors\_quant >= {numOfVis} AND available\_quant >=1 ORDER BY cost ASC;";

try

{

using (classes.MySql.Connection)

{

classes.MySql.Connection.Open();

MySqlCommand command = new MySqlCommand(fillQuery, classes.MySql.Connection);

MySqlDataReader reader = command.ExecuteReader();

using (reader)

{

while (reader.Read())

{

Room.Items.Add($"{reader.GetString(1)} | {reader.GetString(3)} чел. | {reader.GetString(2)} кв.м. ({reader.GetString(4)} руб./ночь)");

}

}

}

}

catch (Exception e)

{

MessageBox.Show(e.Message, "ОШИБКА ЗАПОЛНЕНИЯ", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

//расчет количества дней прожимания

private static double GetDays(DateTime start, DateTime end)

{

double days = Math.Floor((end - start).TotalDays);

return days;

}

//выбор "Питания"

private void Food\_OnSelectionChanged(object sender, SelectionChangedEventArgs e)

{

switch (Food.SelectedIndex)

{

case 1:

\_foodCost = 300 \* \_numDays;

\_idfeed = 1;

break;

case 2:

\_foodCost = 600 \* \_numDays;

\_idfeed = 2;

break;

case 3:

\_foodCost = 900 \* \_numDays;

\_idfeed = 3;

break;

default:

\_idfeed = 4;

break;

}

}

//выбор "Номера"

private void Room\_OnSelectionChanged(object sender, SelectionChangedEventArgs e)

{

string[] Cost = Room.SelectedItem.ToString().Split(' ', '(');

\_nameRoom = Cost[0];

\_roomCost = Double.Parse(Cost[8]) \* \_numDays;

using (classes.MySql.Connection)

{

classes.MySql.Connection.Open();

MySqlCommand command = new MySqlCommand($"SELECT NUMroom FROM hotel.room WHERE r\_name = '{\_nameRoom}' AND cost = {Cost[8]};", classes.MySql.Connection);

\_numroom = Convert.ToInt32(command.ExecuteScalar());

}

}

//расчет итоговой суммы

private void calcCost()

{

\_totalCost = \_roomCost + \_foodCost;

AllSum.Text = \_totalCost.ToString(CultureInfo.CurrentCulture);

}

//кнопка "Расчитать"

private void CalcCost\_OnClick(object sender, RoutedEventArgs e)

{

calcCost();

}

//оформление заселения/брони, добавление данных в базу

private void CheckIntoRoomBut\_OnClick(object sender, RoutedEventArgs e)

{

if (Surname.Text == "" || \_Name.Text == "" || Patr.Text == "" || Birthday.Text == "" ||

PhoneBox.Text == "" || PasSeries.Text == "" || PasNumber.Text == "" || DateIssue.Text == "")

MessageBox.Show("Заполните все поля с данными о клиенте", "ОШИБКА ЗАПОЛНЕНИЯ", MessageBoxButton.OK,

MessageBoxImage.Warning);

else if (Food.SelectedItem == null)

MessageBox.Show("Выберите категорию питания", "ОШИБКА ЗАПОЛНЕНИЯ", MessageBoxButton.OK,

MessageBoxImage.Warning);

else if (Room.SelectedItem == null)

MessageBox.Show("Выберите гостиницный номер", "ОШИБКА ЗАПОЛНЕНИЯ", MessageBoxButton.OK,

MessageBoxImage.Warning);

else

{

calcCost();

CalcAge calcAge = new CalcAge(Birthday.Text);

\_age = calcAge.Age();

string[] dateBirth = Birthday.Text.Split('.');

string[] dateissue = DateIssue.Text.Split('.');

string[] dateArrival = ArrivalDate.SelectedDate.ToString().Split('.', ' ');

string[] dateDeparture = DepartureDate.SelectedDate.ToString().Split('.', ' ');

#region Добавление клиента

string addClientQuery = $"INSERT INTO hotel.client (surname, name, patronymic, birth\_date, age, phone\_number, passport\_series, passport\_number, date\_of\_issue) VALUES ('{Surname.Text}', '{\_Name.Text}', '{Patr.Text}', '{dateBirth[2]}-{dateBirth[1]}-{dateBirth[0]}', {\_age}, {PhoneBox.Text}, {PasSeries.Text}, {PasNumber.Text}, '{dateissue[2]}-{dateissue[1]}-{dateissue[0]}'); select last\_insert\_id();";

try

{

using (classes.MySql.Connection)

{

classes.MySql.Connection.Open();

MySqlCommand command = new MySqlCommand(addClientQuery, classes.MySql.Connection);

\_idclient = Convert.ToInt32(command.ExecuteScalar());

}

}

catch (Exception ex)

{

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

}

#endregion

#region Добавление записи о бронировании номера

int quantRoom = 0;

using (classes.MySql.Connection)

{

classes.MySql.Connection.Open();

string findQuantRoom = $"SELECT available\_quant FROM hotel.room WHERE NUMroom = {\_numroom};";

MySqlCommand command = new MySqlCommand(findQuantRoom, classes.MySql.Connection);

quantRoom = Convert.ToInt32(command.ExecuteScalar());

}

string bookingQuery = $"INSERT INTO hotel.booking (arrival\_date, departure\_date, IDclient, NUMroom, count\_day, IDfeed, amount) VALUES ('{dateArrival[2]}-{dateArrival[1]}-{dateArrival[0]}', '{dateDeparture[2]}-{dateDeparture[1]}-{dateDeparture[0]}', {\_idclient}, {\_numroom}, {\_numDays}, {\_idfeed}, {\_totalCost});";

classes.MySql.AddCommand(bookingQuery);

if (ForBooking.DataArriv.Substring(0, 10).Equals(DateTime.Now.ToShortDateString()))

{

string updateRoomInfo = $"UPDATE hotel.room SET available\_quant = {quantRoom - 1} WHERE NUMroom = {\_numroom};";

classes.MySql.AddCommand(updateRoomInfo);

CreateWordDocCheque();

}

else

{

MessageBox.Show("Бронирование (возможно другой код)");

CreateWordDocBooking();

}

#endregion

}

}

#region Формирование документа "Подтверждение брони"

private void CreateWordDocBooking()

{

string surname = Surname.Text;

string name = \_Name.Text;

string patr = Patr.Text;

string datenow = DateTime.Now.ToShortDateString(), timenow = DateTime.Now.ToShortTimeString();

string arDate = ForBooking.DataArriv.Substring(0, 10);

string depDate = ForBooking.DataDepart.Substring(0, 10);

string residents = ForBooking.NumVis.ToString();

string food = Food.Text;

string room = Room.Text;

string nights = Days.Text;

try

{

var wordApp = new word.Application();

wordApp.Visible = false;

var wordDoc = wordApp.Documents.Open(\_booking);

ReplaceWordStub("<surname>", surname, wordDoc);

ReplaceWordStub("<name>", name, wordDoc);

ReplaceWordStub("<patr>", patr, wordDoc);

ReplaceWordStub("<date\_now>, ", datenow, wordDoc);

ReplaceWordStub("<time\_now>", timenow, wordDoc);

ReplaceWordStub("<n>", name.Substring(0, name.Length - (name.Length - 1)), wordDoc);

ReplaceWordStub("<p>.", patr.Substring(0, patr.Length - (patr.Length - 1)), wordDoc);

ReplaceWordStub("<surname>", surname, wordDoc);

ReplaceWordStub("<arrival\_date>", arDate, wordDoc);

ReplaceWordStub("<departure\_date>", depDate, wordDoc);

ReplaceWordStub("<residents>", residents, wordDoc);

ReplaceWordStub("<food>", food, wordDoc);

ReplaceWordStub("<room>", room, wordDoc);

ReplaceWordStub("<days>", nights, wordDoc);

wordApp.Visible = true;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "ОШИБКА");

}

}

#endregion

#region Формирование документа "Чек о заселении"

private void CreateWordDocCheque()

{

string client = $"{Surname.Text} {\_Name.Text} {Patr.Text}";

string datenow = DateTime.Now.ToShortDateString();

string arDate = ForBooking.DataArriv.Substring(0, 10);

string depDate = ForBooking.DataDepart.Substring(0, 10);

string food = Food.Text;

string room = Room.Text;

string nights = Days.Text;

string amount = AllSum.Text;

Random random = new Random();

string regNum = random.Next(0, 900).ToString();

try

{

var wordApp = new word.Application();

wordApp.Visible = false;

var wordDoc = wordApp.Documents.Open(\_booking);

ReplaceWordStub("<client>", client, wordDoc);

ReplaceWordStub("<date\_now>", datenow, wordDoc);

ReplaceWordStub("<arrival\_date>", arDate, wordDoc);

ReplaceWordStub("<departure\_date>", depDate, wordDoc);

ReplaceWordStub("<food>", food, wordDoc);

ReplaceWordStub("<room>", room, wordDoc);

ReplaceWordStub("<days>", nights, wordDoc);

ReplaceWordStub("<amount>", amount, wordDoc);

ReplaceWordStub("<regnum>", regNum, wordDoc);

ReplaceWordStub("<cost\_room>", amount, wordDoc);

ReplaceWordStub("<cost\_feed>", regNum, wordDoc);

wordApp.Visible = true;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "ОШИБКА");

}

}

#endregion

}

}

Форма Клиенты

using System;

using System.Data;

using System.IO;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Input;

using Hotel.classes;

using MySql.Data.MySqlClient;

namespace Hotel.pages

{

public partial class ClientPage

{

private readonly DataTable \_table = new DataTable();

//ID

private int \_id;

//проверка на выбранную строку

private bool \_checkedStr;

public ClientPage()

{

InitializeComponent();

}

//загрузка данных из базы

private void LoadDb(string query)

{

try

{

\_table.Clear();

using (classes.MySql.Connection)

{

classes.MySql.Connection.Open();

MySqlCommand command = new MySqlCommand(query, classes.MySql.Connection);

MySqlDataAdapter adapter = new MySqlDataAdapter(command);

adapter.Fill(\_table);

ClientDG.ItemsSource = \_table.DefaultView;

}

}

catch (Exception ex)

{

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

}

}

//поиск

private void SearchBox\_OnTextChanged(object sender, TextChangedEventArgs e)

{

LoadDb(String.Format("SELECT surname AS 'Фамилия', name AS 'Имя', patronymic AS 'Отчество', SUBSTRING(birth\_date, 1, 10) AS 'ДатаРождения', age AS 'Возраст', phone\_number AS 'Телефон', passport\_series AS 'Серия', passport\_number AS 'Номер', SUBSTRING(date\_of\_issue, 1, 10) AS 'ДатаВыдачи' FROM hotel.client WHERE surname LIKE '%{0}%';", SearchBox.Text));

}

//кнопка возвращения в главное меню

private void ReturnBut\_OnClick(object sender, RoutedEventArgs e)

{

MessageBoxResult result = MessageBox.Show("Хотите закрыть панель управления клиентами?", "ВЫХОД", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

NavigationService.Navigate(new MainPage());

}

}

//получение ID выбранной строки в базе

private void ClientDG\_OnMouseDoubleClick(object sender, MouseButtonEventArgs e)

{

\_checkedStr = true;

var surname = \_table.DefaultView[ClientDG.SelectedIndex]["Фамилия"];

if (surname != null)

{

try

{

using (classes.MySql.Connection)

{

classes.MySql.Connection.Open();

MySqlCommand command = new MySqlCommand(String.Format("SELECT \* FROM hotel.client WHERE client.surname = '{0}';", surname), classes.MySql.Connection);

MySqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

\_id = Convert.ToInt32(reader.GetString(0));

SurnameBox.Text = reader.GetString(1);

NameBox.Text = reader.GetString(2);

PatrBox.Text = reader.GetString(3);

string tmps = reader.GetString(4);

string[] tmp = tmps.Split(' ');

BirthDateBox.Text = tmp[0];

PhoneBox.Text = reader.GetString(6);

PasSeries.Text = reader.GetString(7);

PasNumber.Text = reader.GetString(8);

tmps = reader.GetString(9);

tmp = tmps.Split(' ');

DateIssue.Text = tmp[0];

}

}

}

catch (Exception ex)

{

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

}

}

}

#region Контроль ввода данных

private void SurnameBox\_OnPreviewTextInput(object sender, TextCompositionEventArgs e)

{

if (!(Char.IsLetter(e.Text, 0)))

e.Handled = true;

}

private void PasSeries\_OnPreviewTextInput(object sender, TextCompositionEventArgs e)

{

if (PasSeries.Text.Length >= 4 || !(Char.IsDigit(e.Text, 0)))

e.Handled = true;

}

private void PasNumber\_OnPreviewTextInput(object sender, TextCompositionEventArgs e)

{

if (PasNumber.Text.Length >= 6 || !(Char.IsDigit(e.Text, 0)))

e.Handled = true;

}

private void DateIssue\_OnPreviewTextInput(object sender, TextCompositionEventArgs e)

{

if ((DateIssue.Text.Length >= 10 || !(Char.IsDigit(e.Text, 0))) && (BirthDateBox.Text.Length >= 10 || !(Char.IsDigit(e.Text, 0))))

e.Handled = true;

}

private void PhoneBox\_OnPreviewTextInput(object sender, TextCompositionEventArgs e)

{

if (PhoneBox.Text.Length >= 11 || !(Char.IsDigit(e.Text, 0)))

e.Handled = true;

}

#endregion

//удаление выбранной строку из базы

private void DeleteBut\_OnMouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

if (\_checkedStr == false)

MessageBox.Show("Выберите клиента которого хотите удалить из базы", "УДАЛЕНИЕ", MessageBoxButton.OK, MessageBoxImage.Error);

else

{

try

{

using (classes.MySql.Connection)

{

classes.MySql.Connection.Open();

MessageBoxResult result = MessageBox.Show("Вы действительно хотите удалить клиента?", "УДАЛЕНИЕ КЛИЕНТА", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

MySqlCommand command = new MySqlCommand(String.Format("DELETE FROM hotel.client WHERE IDclient = {0};", \_id), classes.MySql.Connection);

command.ExecuteNonQuery();

SurnameBox.Clear();

NameBox.Clear();

PatrBox.Clear();

BirthDateBox.Clear();

PhoneBox.Clear();

PasSeries.Clear();

PasNumber.Clear();

DateIssue.Clear();

}

}

LoadDb("SELECT surname AS 'Фамилия', name AS 'Имя', patronymic AS 'Отчество', SUBSTRING(birth\_date, 1, 10) AS 'ДатаРождения', age AS 'Возраст', phone\_number AS 'Телефон', passport\_series AS 'Серия', passport\_number AS 'Номер', SUBSTRING(date\_of\_issue, 1, 10) AS 'ДатаВыдачи' FROM hotel.client;");

\_checkedStr = false;

}

catch (Exception ex)

{

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

}

}

}

//обновление данных в базе

private void DoneBut\_OnMouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

if (\_checkedStr == false)

MessageBox.Show("Выберите клиента, которого хотите отредактировать", "РЕДАКТИРОВАНИЕ", MessageBoxButton.OK, MessageBoxImage.Error);

else

{

if (SurnameBox.Text == "" || NameBox.Text == "" || PatrBox.Text == "" || BirthDateBox.Text == "" || PhoneBox.Text == "" || PasSeries.Text == "" || PasNumber.Text == "" || DateIssue.Text == "")

MessageBox.Show("Все полня должны быть заполнены", "ОШИБКА ЗАПОЛНЕНИЯ", MessageBoxButton.OK, MessageBoxImage.Error);

else

{

try

{

string[] tmp = BirthDateBox.Text.Split('.');

string birthDate = String.Format("{0}-{1}-{2}", tmp[2], tmp[1], tmp[0]);

CalcAge calcAge = new CalcAge(BirthDateBox.Text);

int age = calcAge.Age();

tmp = DateIssue.Text.Split('.');

string dateIssue = String.Format("{0}-{1}-{2}", tmp[2], tmp[1], tmp[0]);

string query = string.Format("UPDATE hotel.client SET surname = '{0}', name = '{1}', patronymic = '{2}', birth\_date = '{3}', age = {4}, phone\_number = '{5}', passport\_series = '{6}', passport\_number = '{7}', date\_of\_issue = '{8}' WHERE client.IDclient = {9};", SurnameBox.Text, NameBox.Text, PatrBox.Text, birthDate, age, PhoneBox.Text, PasSeries.Text, PasNumber.Text, dateIssue, \_id);

using (classes.MySql.Connection)

{

classes.MySql.Connection.Open();

MySqlCommand command = new MySqlCommand(query, classes.MySql.Connection);

command.ExecuteNonQuery();

}

LoadDb("SELECT surname AS 'Фамилия', name AS 'Имя', patronymic AS 'Отчество', SUBSTRING(birth\_date, 1, 10) AS 'ДатаРождения', age AS 'Возраст', phone\_number AS 'Телефон', passport\_series AS 'Серия', passport\_number AS 'Номер', SUBSTRING(date\_of\_issue, 1, 10) AS 'ДатаВыдачи' FROM hotel.client;");

}

catch (Exception ex)

{

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

}

SurnameBox.Clear();

NameBox.Clear();

PatrBox.Clear();

BirthDateBox.Clear();

PhoneBox.Clear();

PasSeries.Clear();

PasNumber.Clear();

DateIssue.Clear();

\_checkedStr = false;

}

}

}

//экспорт данных

private void ExpBut\_OnMouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

try

{

using(classes.MySql.Connection)

{

classes.MySql.Connection.Open();

MySqlCommand comand = new MySqlCommand("SELECT \* FROM hotel.client;", classes.MySql.Connection);

MySqlDataReader reader = comand.ExecuteReader();

DataTable clientTable = new DataTable();

clientTable.Load(reader);

using (StreamWriter writer = new StreamWriter($@"{Directory.GetCurrentDirectory()}\\ExportCsv\\ClientData.csv", false))

{

for(int columnCount = 0; columnCount <= clientTable.Columns.Count - 1; ++columnCount)

{

if (columnCount != clientTable.Columns.Count - 1) writer.Write(clientTable.Columns[columnCount].ColumnName + ";");

else writer.Write(clientTable.Columns[columnCount].ColumnName);

}

writer.Write("\n");

foreach(DataRow row in clientTable.Rows)

{

string str = String.Join(";", row.ItemArray);

writer.WriteLine(str);

}

}

}

MessageBox.Show("Данные о клиентах выгружены!", "ЭКСПОРТ ДАННЫХ", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

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

}

}

//загрузка формы

private void ClientPage\_OnLoaded(object sender, RoutedEventArgs e)

{

LoadDb("SELECT surname AS 'Фамилия', name AS 'Имя', patronymic AS 'Отчество', SUBSTRING(birth\_date, 1, 10) AS 'ДатаРождения', age AS 'Возраст', phone\_number AS 'Телефон', passport\_series AS 'Серия', passport\_number AS 'Номер', SUBSTRING(date\_of\_issue, 1, 10) AS 'ДатаВыдачи' FROM hotel.client;");

}

}

}

Форма Номера

using System;

using System.Data;

using System.Windows;

using System.Windows.Input;

using System.IO;

using MySql.Data.MySqlClient;

namespace Hotel.pages

{

public partial class RoomPage

{

private readonly DataTable \_table = new DataTable();

private int \_id;

private bool \_checkedStr;

public RoomPage()

{

InitializeComponent();

}

//загрузка данных из базы

private void LoadDb(string query)

{

try

{

\_table.Clear();

using(classes.MySql.Connection)

{

classes.MySql.Connection.Open();

MySqlCommand command= new MySqlCommand(query, classes.MySql.Connection);

MySqlDataAdapter adapter = new MySqlDataAdapter(command);

adapter.Fill(\_table);

RoomDG.ItemsSource = \_table.DefaultView;

}

}

catch(Exception ex)

{

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

}

}

//кнопка возвращения в главное меню

private void ReturnBut\_OnClick(object sender, RoutedEventArgs e)

{

MessageBoxResult result = MessageBox.Show("Хотите закрыть панель управления гостиничными номерами?", "ВЫХОД", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

NavigationService.Navigate(new MainPage());

}

}

//получение ID выбранной строки

private void RoomDG\_OnMouseDoubleClick(object sender, MouseButtonEventArgs e)

{

\_checkedStr = true;

try

{

\_id = Convert.ToInt32(\_table.DefaultView[RoomDG.SelectedIndex]["№"]);

}

catch(Exception)

{

MessageBox.Show("Некорректно выбрана строка", "ОШИБКА", MessageBoxButton.OK, MessageBoxImage.Error);

}

if (\_id != -1)

{

try

{

using(classes.MySql.Connection)

{

classes.MySql.Connection.Open();

MySqlCommand command= new MySqlCommand(String.Format("SELECT \* FROM hotel.room WHERE NUMroom = {0}", \_id), classes.MySql.Connection);

MySqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

\_id= Convert.ToInt32(reader.GetString(0));

NameBox.Text = reader.GetString(1);

AreaBox.Text = reader.GetString(2);

VisitQuant.Text = reader.GetString(3);

CostBox.Text = reader.GetString(4);

AvailableQuant.Text = reader.GetString(5);

}

}

}

catch(Exception ex )

{

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

}

}

}

#region Контроль ввода

private void NameBox\_OnPreviewTextInput(object sender, TextCompositionEventArgs e)

{

if (Char.IsDigit(e.Text, 0))

e.Handled = true;

}

private void AreaBox\_OnPreviewTextInput(object sender, TextCompositionEventArgs e)

{

if (!(Char.IsDigit(e.Text, 0)))

e.Handled = true;

}

#endregion

//обновление данных в базе

private void DoneBut\_OnMouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

if(\_checkedStr == false)

MessageBox.Show("Выберите номер, данные о котором хотите отредактировать", "Редактирование номера ", MessageBoxButton.OK, MessageBoxImage.Error);

else

{

if(NameBox.Text == "" || AreaBox.Text == "" || CostBox.Text == "" || VisitQuant.Text == "" || AvailableQuant.Text == "")

MessageBox.Show("Все полня должны быть заполнены", "Ошибка заполнения", MessageBoxButton.OK, MessageBoxImage.Error);

else

{

try

{

string query = String.Format("UPDATE hotel.room SET r\_name = '{0}', area = {1}, visitors\_quant = {2}, cost = {3}, available\_quant = {4} WHERE room.NUMroom = {5};", NameBox.Text, AreaBox.Text, VisitQuant.Text, CostBox.Text, AvailableQuant.Text, \_id);

using(classes.MySql.Connection)

{

classes.MySql.Connection.Open();

MySqlCommand command = new MySqlCommand(query, classes.MySql.Connection);

command.ExecuteNonQuery();

}

LoadDb("SELECT NUMroom AS '№', r\_name AS 'Название', area AS 'Площадь', visitors\_quant AS 'Посетителей', cost AS 'Стоимость', available\_quant AS 'Свободных' FROM hotel.room;");

NameBox.Clear();

AreaBox.Clear();

CostBox.Clear();

VisitQuant.Text = String.Empty;

AvailableQuant.Text = String.Empty;

}

catch(Exception ex)

{

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

}

}

}

}

//добавление новой записи в базу

private void AddBut\_OnMouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

if(\_checkedStr)

{

MessageBox.Show("Нельзя добавить уже сущестующий номер", "ОШИБКА ДОБАВЛЕНИЯ", MessageBoxButton.OK, MessageBoxImage.Information);

NameBox.Clear();

AreaBox.Clear();

CostBox.Clear();

VisitQuant.Text = String.Empty;

AvailableQuant.Text = String.Empty;

\_checkedStr = false;

}

else

{

if (NameBox.Text == "" || AreaBox.Text == "" || CostBox.Text == "" || VisitQuant.Text == "" || AvailableQuant.Text == "")

MessageBox.Show("Все полня должны быть заполнены", "ОШИБКА ЗАПОЛНЕНИЯ", MessageBoxButton.OK, MessageBoxImage.Error);

else

{

try

{

string query = String.Format("INSERT INTO hotel.room (r\_name, area, visitors\_quant, cost, available\_quant) VALUES ('{0}', {1}, {2}, {3}, {4});", NameBox.Text, AreaBox.Text, VisitQuant.Text, CostBox.Text, AvailableQuant.Text);

using (classes.MySql.Connection)

{

classes.MySql.Connection.Open();

MessageBoxResult result = MessageBox.Show("Вы действительно хотите добавить новую комнату в базу?", "ДОБАВЛЕНИЕ ДАННЫХ", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

MySqlCommand command = new MySqlCommand(query, classes.MySql.Connection);

command.ExecuteNonQuery();

NameBox.Clear();

AreaBox.Clear();

CostBox.Clear();

VisitQuant.Text = String.Empty;

AvailableQuant.Text = String.Empty;

}

}

LoadDb("SELECT NUMroom AS '№', r\_name AS 'Название', area AS 'Площадь', visitors\_quant AS 'Посетителей', cost AS 'Стоимость', available\_quant AS 'Свободных' FROM hotel.room;");

}

catch (Exception ex)

{

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

}

}

}

}

//удаление выбранной записи

private void DeleteBut\_OnMouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

if(\_checkedStr == false)

MessageBox.Show("Выберите номер который хотите удалить из базы", "УДАЛЕНИЕ", MessageBoxButton.OK, MessageBoxImage.Error);

else

{

try

{

using(classes.MySql.Connection)

{

classes.MySql.Connection.Open();

MessageBoxResult result = MessageBox.Show("Вы действительно хотите удалить этот номер из базы?", "УДАЛЕНИЕ", MessageBoxButton.YesNo, MessageBoxImage.Question);

if(result == MessageBoxResult.Yes)

{

MySqlCommand command = new MySqlCommand(String.Format("DELETE FROM hotel.room WHERE NUMroom = {0};", \_id), classes.MySql.Connection);

command.ExecuteNonQuery();

NameBox.Clear();

AreaBox.Clear();

CostBox.Clear();

VisitQuant.Text = String.Empty;

AvailableQuant.Text = String.Empty;

}

}

LoadDb("SELECT NUMroom AS '№', r\_name AS 'Название', area AS 'Площадь', visitors\_quant AS 'Посетителей', cost AS 'Стоимость', available\_quant AS 'Свободных' FROM hotel.room;");

\_checkedStr = false;

}

catch (Exception ex)

{

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

}

}

}

//загрузка формы

private void RoomPage\_OnLoaded(object sender, RoutedEventArgs e)

{

LoadDb("SELECT NUMroom AS '№', r\_name AS 'Название', area AS 'Площадь', visitors\_quant AS 'Посетителей', cost AS 'Стоимость', available\_quant AS 'Свободных' FROM hotel.room;");

}

//эскпорт данных

private void ExpBut\_OnMouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

try

{

using(classes.MySql.Connection)

{

classes.MySql.Connection.Open();

MySqlCommand comand = new MySqlCommand("SELECT \* FROM hotel.room;", classes.MySql.Connection);

MySqlDataReader reader = comand.ExecuteReader();

DataTable roomTable = new DataTable();

roomTable.Load(reader);

using (StreamWriter writer = new StreamWriter($@"{Directory.GetCurrentDirectory()}\\ExportCsv\\RoomData.csv", false))

{

for(int columnCount = 0; columnCount <= roomTable.Columns.Count - 1; ++columnCount)

{

if (columnCount != roomTable.Columns.Count - 1) writer.Write(roomTable.Columns[columnCount].ColumnName + ";");

else writer.Write(roomTable.Columns[columnCount].ColumnName);

}

writer.Write("\n");

foreach(DataRow row in roomTable.Rows)

{

string str = String.Join(";", row.ItemArray);

writer.WriteLine(str);

}

}

}

MessageBox.Show("Данные о номерах гостиницы выгружены!", "ЭКСПОРТ ДАННЫХ", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

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

}

}

}

}

Форма Данные о клиентах

using System;

using System.Data;

using System.IO;

using System.Windows;

using System.Windows.Input;

using Hotel.classes;

using MySql.Data.MySqlClient;

namespace Hotel.pages

{

public partial class UserDataBase

{

private readonly DataTable \_table = new DataTable();

private string \_pwd = string.Empty;

private int \_id;

public UserDataBase()

{

InitializeComponent();

}

//загрузка данных из базы

private void LoadDb()

{

try

{

\_table.Clear();

using (classes.MySql.Connection)

{

classes.MySql.Connection.Open();

MySqlCommand command = new MySqlCommand("SELECT u\_name AS 'NAME', u\_login AS 'LOGIN', u\_password AS 'PASSWORD', u\_type AS 'TYPE' FROM hotel.user;", classes.MySql.Connection);

MySqlDataAdapter adapter = new MySqlDataAdapter(command);

adapter.Fill(\_table);

DataUser.ItemsSource = \_table.DefaultView;

}

}

catch (Exception ex)

{

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

}

}

//кнопка возвращения на главную формы

private void ReturnBut\_OnClick(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new AddUser());

}

//закрытие формы

private void CloseBut\_OnClick(object sender, RoutedEventArgs e)

{

var result = MessageBox.Show("Хотите вернуться в главное меню?", "ВЫХОД",

MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

NavigationService.Navigate(new MainPage());

}

}

//получение ID выбранной строки

private void DataUser\_OnMouseDoubleClick(object sender, MouseButtonEventArgs e)

{

var login = \_table.DefaultView[DataUser.SelectedIndex]["LOGIN"];

if (login == null)

{

MessageBox.Show("Строка не выбрана", "ОШИБКА", MessageBoxButton.OK, MessageBoxImage.Error);

}

else

{

string query = $"SELECT IDuser, u\_name, u\_login, u\_password FROM hotel.user WHERE hotel.user.u\_login = '{login}';";

using(classes.MySql.Connection)

{

classes.MySql.Connection.Open();

MySqlCommand command = new MySqlCommand(query, classes.MySql.Connection);

MySqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

\_id = reader.GetInt32(0);

UsernameBox.Text = reader.GetString(1);

LoginBox.Text = reader.GetString(2);

\_pwd = reader.GetString(3);

}

}

}

}

//генерация пароля

private void GeneratePwd\_OnMouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

PasswdBox.Password = GenerateRwd.GeneratePasswd();

}

//удаление записи из базы

private void DeleteBut\_OnMouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

try

{

using (classes.MySql.Connection)

{

classes.MySql.Connection.Open();

MessageBoxResult result = MessageBox.Show($"Вы действительно хотите удалить пользователя\n{UsernameBox.Text}?", "УДАЛЕНИЕ", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

MySqlCommand command = new MySqlCommand($"DELETE FROM hotel.user WHERE u\_login = '{LoginBox.Text}';", classes.MySql.Connection);

command.ExecuteNonQuery();

//classes.MySql.connection.Close();

}

}

LoadDb();

}

catch(Exception ex)

{

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

}

UsernameBox.Clear();

LoginBox.Clear();

PasswdBox.Clear();

}

//обновление данных в базе

private void DoneBut\_OnMouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

if (UsernameBox.Text == String.Empty || LoginBox.Text == String.Empty)

MessageBox.Show("Все поля должны быть заполнены!", "ОШИБКА ЗАПОЛНЕНИЯ", MessageBoxButton.OK, MessageBoxImage.Warning);

else if (PasswdBox.Password != "" && PasswdBox.Password != " ")

{

if (PasswdBox.Password.Length >= 8)

\_pwd = PasswdBox.Password;

else

MessageBox.Show("Пароль должен быть не меньше восьми символов!", "ОШИБКА ЗАПОЛНЕНИЯ", MessageBoxButton.OK, MessageBoxImage.Warning);

}

else

{

try

{

using (classes.MySql.Connection)

{

classes.MySql.Connection.Open();

string query = $"UPDATE hotel.user SET u\_name = '{UsernameBox.Text}', u\_login = '{LoginBox.Text}', u\_password = '{\_pwd}' WHERE IDuser = {\_id}";

MySqlCommand command = new MySqlCommand(query, classes.MySql.Connection);

command.ExecuteNonQuery();

}

LoadDb();

}

catch (Exception ex)

{

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

}

UsernameBox.Clear();

LoginBox.Clear();

PasswdBox.Clear();

}

}

//загрузка формы

private void UserDataBase\_OnLoaded(object sender, RoutedEventArgs e)

{

LoadDb();

}

//экспорт данных

private void ExpBut\_OnMouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

try

{

using(classes.MySql.Connection)

{

classes.MySql.Connection.Open();

MySqlCommand comand = new MySqlCommand("SELECT \* FROM hotel.user;", classes.MySql.Connection);

MySqlDataReader reader = comand.ExecuteReader();

DataTable userTable = new DataTable();

userTable.Load(reader);

using (StreamWriter writer = new StreamWriter($@"{Directory.GetCurrentDirectory()}\\ExportCsv\\UserData.csv", false))

{

for(int columnCount = 0; columnCount <= userTable.Columns.Count - 1; ++columnCount)

{

if (columnCount != userTable.Columns.Count - 1) writer.Write(userTable.Columns[columnCount].ColumnName + ";");

else writer.Write(userTable.Columns[columnCount].ColumnName);

}

writer.Write("\n");

foreach(DataRow row in userTable.Rows)

{

string str = String.Join(";", row.ItemArray);

writer.WriteLine(str);

}

}

}

MessageBox.Show("Данные о пользователях системы выгружены!", "ЭКСПОРТ ДАННЫХ", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

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

}

}

}

}

Форма Добавления нового пользователя

using System.Windows;

using System.Windows.Input;

using Hotel.classes;

namespace Hotel.pages

{

public partial class AddUser

{

public AddUser()

{

InitializeComponent();

}

//закрытие формы

private void CloseBut\_OnClick(object sender, RoutedEventArgs e)

{

var result = MessageBox.Show("Хотите вернуться в главное меню?", "ВЫХОД", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

NavigationService.Navigate(new MainPage());

}

}

#region Контроль ввода данных

private void UsernameBox\_OnPreviewTextInput(object sender, TextCompositionEventArgs e)

{

if(!(char.IsLetter(e.Text, 0) || (e.Text == ".")))

e.Handled = true;

}

private void LoginBox\_OnPreviewTextInput(object sender, TextCompositionEventArgs e)

{

if (e.Text == "'")

e.Handled = true;

}

#endregion

//генерация пароля

private void GeneratePwd\_OnClick(object sender, RoutedEventArgs e)

{

PasswdBox.Password = GenerateRwd.GeneratePasswd();

}

//кнопка добавления нового пользователя

private void SignInBut\_OnClick(object sender, RoutedEventArgs e)

{

string username = UsernameBox.Text;

string login = LoginBox.Text;

string passwd = GetHashPasswd.GetHash(PasswdBox.Password);

if(username == string.Empty || login == string.Empty || passwd == string.Empty)

{

MessageBox.Show("Все поля должны быть заполнены!", "ОШИБКА ЗАПОЛНЕНИЯ", MessageBoxButton.OK, MessageBoxImage.Information);

}

else if(passwd.Length < 8)

{

MessageBox.Show("Пароль должен быть не меньше 8 символов!", "ОШИБКА ЗАПОЛНЕНИЯ", MessageBoxButton.OK, MessageBoxImage.Information);

}

else

{

string queryadd = $"INSERT INTO user (u\_name, u\_login, u\_password, u\_type) VALUES ('{username}','{login}','{passwd}','user');";

MessageBoxResult boxResult = MessageBox.Show($"Вы действительно хотите добавить пользователя {login} ({username})?", "ДОБАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯ", MessageBoxButton.YesNo, MessageBoxImage.Question);

if(boxResult == MessageBoxResult.Yes)

{

classes.MySql.AddCommand(queryadd);

UsernameBox.Clear();

LoginBox.Clear();

PasswdBox.Clear();

}

}

}

//кнопка перехода на форму "База данных пользователей"

private void UserDBBut\_OnClick(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new UserDataBase());

}

}

}