**Министерство образования Московской области**

**ГОУ ВО МО «Государственный гуманитарно-технологический университет» (ГГТУ) Ликино-Дулевский политехнический колледж- филиал ГГТУ**

Наименование колледжа/техникума

**О Т Ч Ё Т**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

(код и наименование ПМ)

Обучающейся\_\_\_\_Чулковой Дарья Романовны\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ф.И.О.

Курс \_\_\_\_3\_\_\_\_\_ группа \_\_\_ИСП.22А\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Специальность 09.02.07 Информационные системы и программирование \_\_\_

Место практики Ликино-Дулевский политехнический колледж – филиал ГГТУ

Период практики 25.11.2024 г. – 07.12.2024 г., 07.04.2025 г. - 19.04.2025 г.

Руководители практики

от колледжа \_\_\_\_\_\_\_\_ Кузьмина Елена Евгеньевна \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Гжегожевский Сергей Владимирович\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ветрова Карина Валерьевна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тишкин Валентин Васильевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ф.И.О.

Г. Ликино-Дулево

2025 г.

Оглавление

[Введение 3](#_Toc184378980)

[1. Разработка технического задания 4](#_Toc184378981)

[2. Проектирование и создание базы данных 8](#_Toc184378982)

[2.1 Таблицы данных 9](#_Toc184378983)

[3.Разработка интерфейса приложения 12](#_Toc184378984)

[4.Разработка программных моделей 17](#_Toc184378985)

[5.Разработка Api 29](#_Toc184378986)

[6.Разработка библиотеки класов 34](#_Toc184378987)

[7.Заключение 36](#_Toc184378988)

# Введение

Целью учебной практики является закрепление и углубление знаний, полученных обучающимися в процессе теоретического обучения по профессиональному модулю ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем».

Задачи практики:

1. Получение первичных профессиональных умений и навыков;

2. Подготовка студентов к осознанному и глубокому изучению общепрофессиональных и специальных дисциплин;

3. Получение практических профессиональных умений и навыков по специальности 09.02.07 Информационные системы и программирование.

С целью овладения видом деятельности «Разработка модулей программного обеспечения для компьютерных систем» обучающийся должен обладать соответствующими профессиональными компетенциями:

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием;

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием;

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств;

ПК 1.4. Выполнять тестирование программных модулей;

# **1. Разработка технического задания**

*1*. Разработать социальную сеть для организации путешествий, предлагая широкий спектр услуг, направленных на облегчение процесса планирования и бронирования.

* ***Сущности***:

Авиабилет : Дата вылета, Дата прибытия, Город отправления, Город назначения, Класс обслуживания, Цена, Авиакомпания, Самолёт.

Клиент : Фио, Почта, Телефон, Пароль.

Гостиница : Название гостиницы, Рейтинг, Тип номера, Цена.

Оплата : Способ оплаты, Статус транзакции

Бронирование : Состав заказа, Статус бронирования, Цена

Тур : Название, Описание, Цена, Количество мест, Дата начала, Дата окончания.

* ***Процессы***:

добавление туров гостиниц и авиабилетов ,просмотр бронирования ,добавление клиентов ,регистрация и авторизация

2. *Разработка ER – модели и блок-схемы бизнес-процессов*

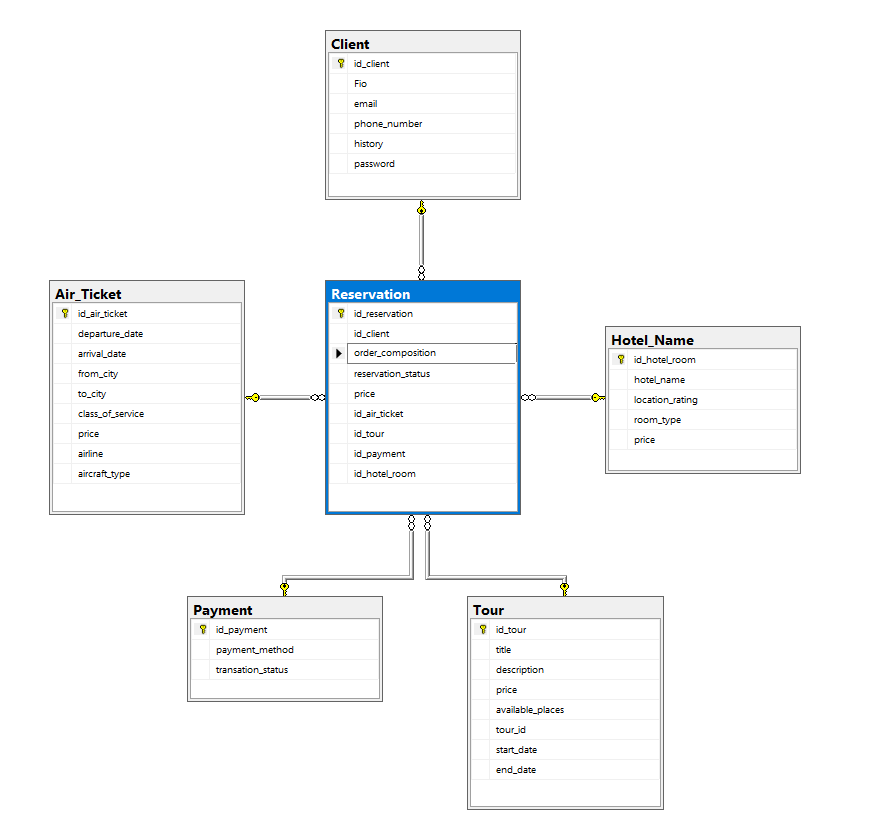


рис.1 «ER-диаграмма»

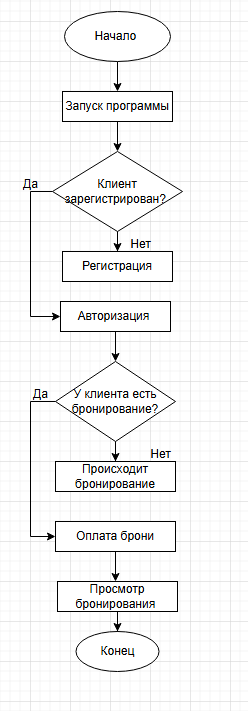


рис.2 «Блок-схема»

***3. Описание функциональных требований.***

1. Регистрация и авторизация
2. Брорирование туров ,авиабилетов и гостиниц
3. Просмотр бронирования
4. Добавление клиентов

***4. Описание нефункциональных требований.***

1. Быстрое время отклика
2. Защита данных
3. Интуитивное понятие интерфейса

***5. Анализ рисков и ограничений.***

1. Возможные сбои в системе, которые могут привести к недоступности услуг
2. Ошибки в программном обеспечении

# **2. Проектирование и создание базы данных**

*Цель:* Разработка структуры базы данных для хранения данных о пользователях, книгах, личных полках пользователя, отзывов, друзей

*Задание:* На основании описания предметной области, созданной ранее, разработать базу данных, выполнив следующие шаги:

СУБД: Microsoft SQL Server

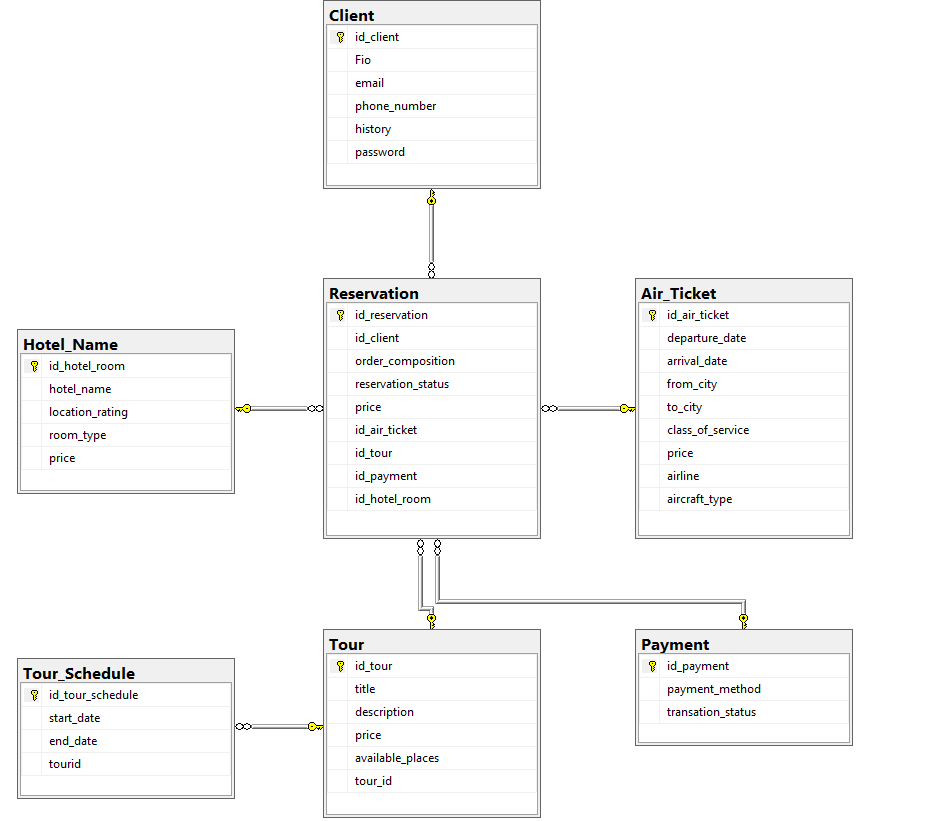
**

Рис.3 «ER-диаграмма»

## 2.1 Таблицы данных



Рис.4 «Авиабилеты»

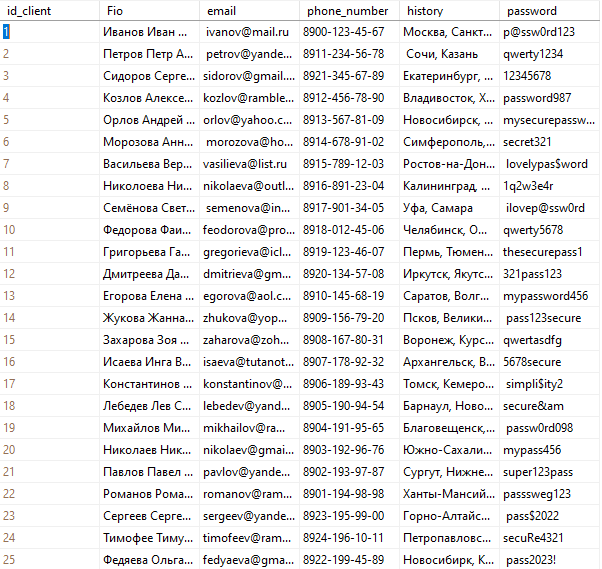


Рис.5 «Клиенты»

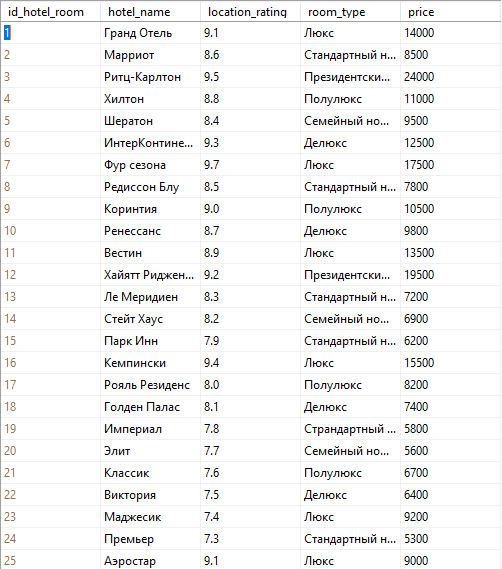


Рис.6 «Гостиницы»

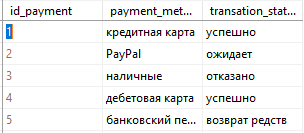


Рис.7 «Оплата»

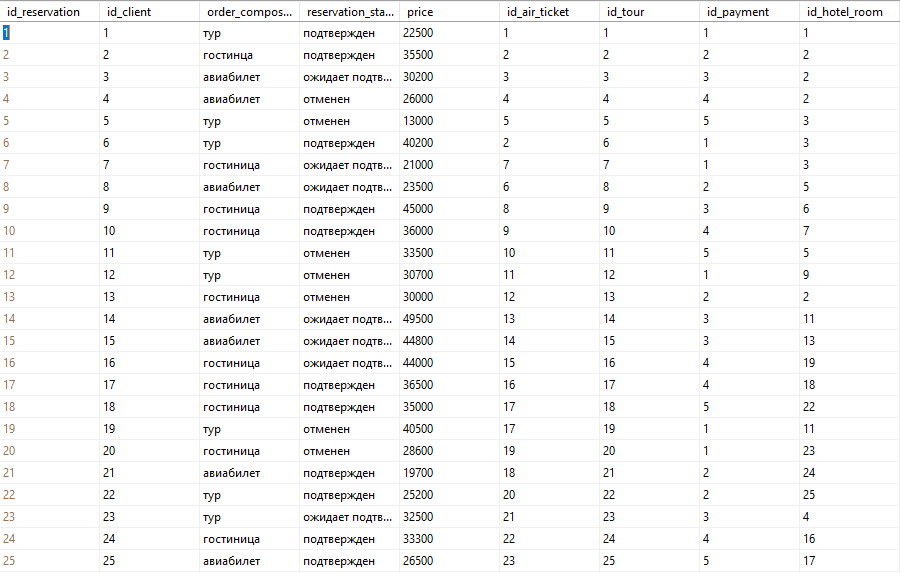


Рис.8 «Бронирование»

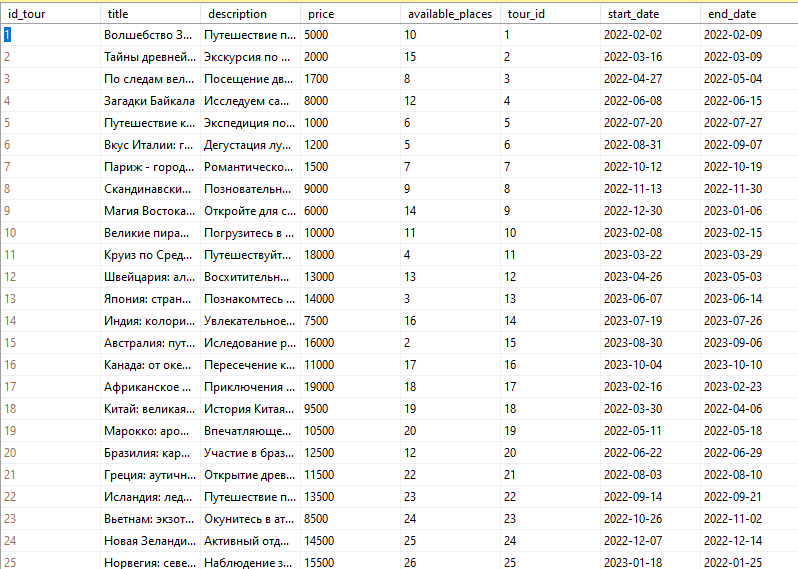


Рис.9 «Туры»

# 3.Разработка интерфейса приложения

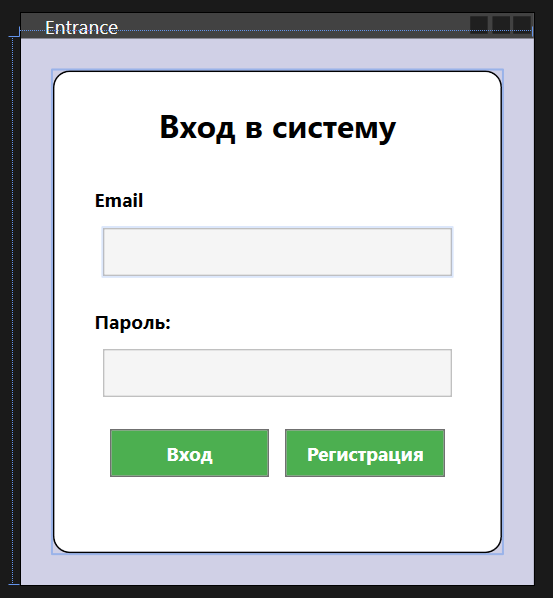


Рис.10 «Авторизация»

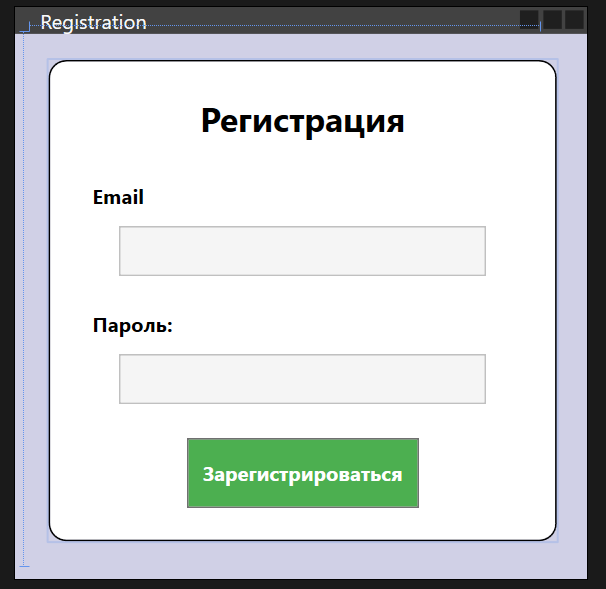


Рис.11 «Регистриция»

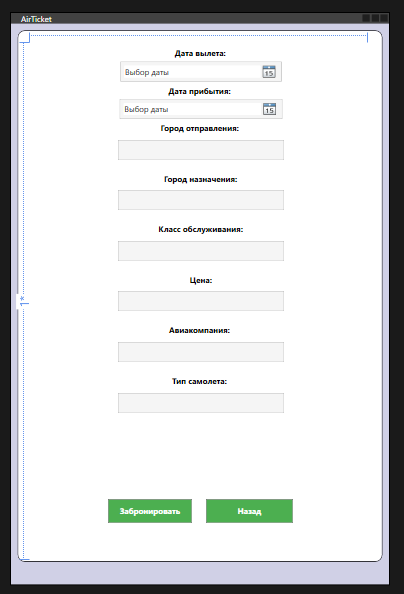


Рис.12 «Бронирование Авиабилета»

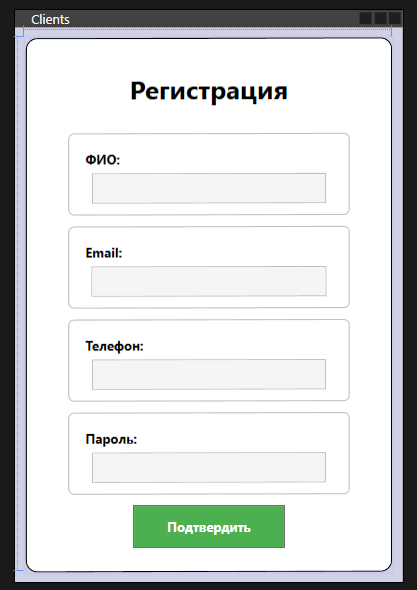


Рис.13 «Регистрация клиента»

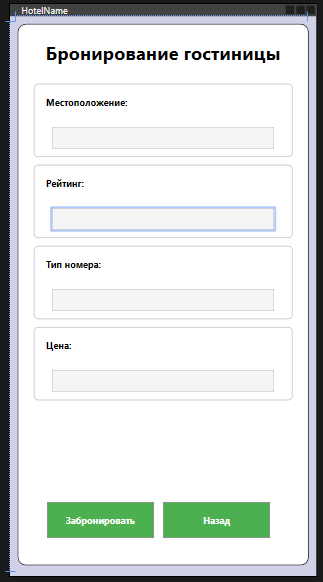


Рис.14 «Бронирование гостиницы»

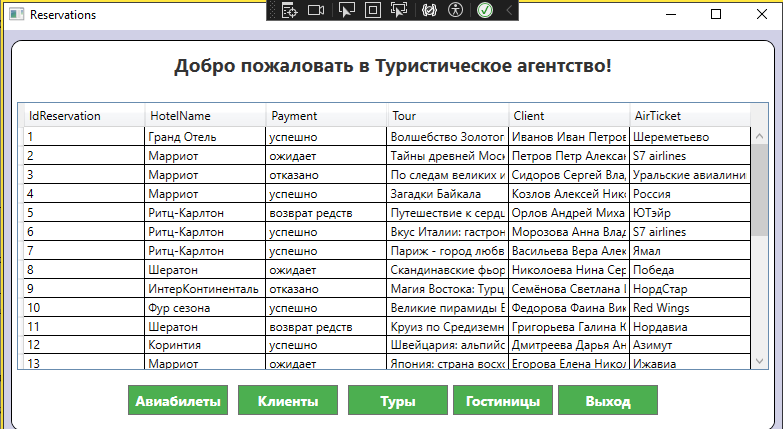


Рис.15 «Окно бронирования»

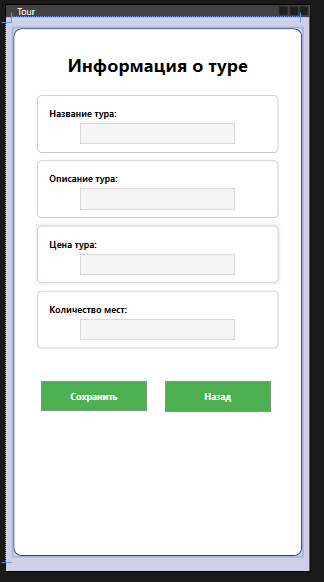


Рис.16 «Бронирование тура»

# 4.Разработка программных моделей

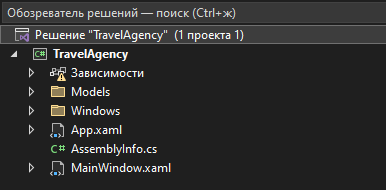


Рис. «Сткуртура проекта»

*Код входа в систему*

<Grid>

<Grid x:Name="LoginGrid" Visibility="Visible" Margin="20">

<Border CornerRadius="10" Background="White" BorderThickness="1" BorderBrush="Black" Padding="20">

<StackPanel Margin="0,0,0,-11">

<TextBlock Text="Вход в систему" FontSize="20" FontWeight="Bold"

HorizontalAlignment="Center" Margin="0,0,0,20"/>

<Label Content="Email" FontWeight="Bold"/>

<TextBox Name="txtEmailTextBox" Width="218" Height="30" Margin="0,5,0,15"

Padding="5" Background="#F5F5F5" BorderBrush="#C0C0C0" BorderThickness="1"/>

<Label Content="Пароль:" FontWeight="Bold"/>

<TextBox Name="txtPasswordTextBox" Width="218" Height="30" Margin="0,5,0,20"

Background="#F5F5F5" BorderBrush="#C0C0C0" BorderThickness="1"/>

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

<Button Content="Вход" Width="100" Height="30" Name="btnEnter"

Click="btnEnter\_Click" Background="#4CAF50"

Foreground="White" FontWeight="Bold" Margin="0,0,10,0"/>

<Button Content="Регистрация" Width="100" Height="30" Name="btnReg"

Click="btnReg\_Click" Background="#4CAF50"

Foreground="White" FontWeight="Bold"/>

</StackPanel>

</StackPanel>

</Border>

</Grid>

</Grid>

*Логика для входа в систему*

public partial class MainWindow : Window

{

private List<Client> registere;

public MainWindow()

{

InitializeComponent();

}

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

{

string email = txtEmailTextBox.Text;

string password = txtPasswordTextBox.Text;

var aaaa = AuthenticateClient(email, password);

if (aaaa != null)

{

WindowReservation windowReservation = new WindowReservation();

windowReservation.Show();

this.Close();

}

else

{

MessageBox.Show("Неправильная почта или пароль", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

WindowRegistration windowRegistration = new WindowRegistration();

windowRegistration.Show();

this.Close();

}

private bool AuthenticateClient(string email, string password)

{

using (TurAgenstvooContext context = new TurAgenstvooContext())

{

var client = context.Clients.FirstOrDefault(e => e.Email == email && e.Password == password);

return client != null;

}

}

}

*Код для авторизации*

<Grid Margin="10,0,27,7">

<Border CornerRadius="10" Background="White" BorderThickness="1"

BorderBrush="Black" Padding="20" Margin="10,0,-9,0" Height="286" VerticalAlignment="Center">

<StackPanel Margin="0,0,0,-11">

<TextBlock Text="Регистрация" FontSize="20" FontWeight="Bold"

HorizontalAlignment="Center" Margin="0,0,0,20"/>

<Label Content="Email" FontWeight="Bold"/>

<TextBox Name="EmailTextBox" Width="218" Height="30"

Margin="0,5,0,15" Padding="5"

Background="#F5F5F5" BorderBrush="#C0C0C0"

BorderThickness="1"/>

<Label Content="Пароль:" FontWeight="Bold"/>

<PasswordBox Name="PasswordTextBox" Width="218" Height="30"

Margin="0,5,0,20"

Background="#F5F5F5" BorderBrush="#C0C0C0"

BorderThickness="1"/>

<StackPanel Orientation="Vertical" Width="141" >

<Button Content="Зарегистрироваться" Width="138" Height="42"

Click="ButtonRegister\_Click" Background="#4CAF50"

Foreground="White" FontWeight="Bold"/>

</StackPanel>

</StackPanel>

</Border>

</Grid>

*Логика для авторизации*

public partial class WindowRegistration : Window

{

Client newClient = new Client();

public WindowRegistration()

{

InitializeComponent();

}

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

{

string email = EmailTextBox.Text;

string password = PasswordTextBox.Password;

if (email == null)

{

MessageBox.Show("Введите почту!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

else

{

if (password == null)

{

MessageBox.Show("Введите пароль!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

else

{

using (TurAgenstvooContext db = new TurAgenstvooContext())

{

var aunt = db.Clients.FirstOrDefault(u => u.Email == email);

if (aunt != null)

{

MessageBox.Show("Пользователь с такой почтой уже существует!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

else

{

newClient.Email = email;

newClient.Password = password;

db.Clients.Add(newClient);

db.SaveChanges();

MainWindow mainWindow = new MainWindow();

mainWindow.Show();

this.Close();

}

}

}

}

}

}

*Код для бронирования авиабилета*

<Border CornerRadius="10" Background="White" BorderThickness="1"

BorderBrush="Black" Padding="20" Margin="10,10,10,34">

<Grid Height="775" VerticalAlignment="Top">

<Grid.RowDefinitions>

<RowDefinition/>

</Grid.RowDefinitions>

<StackPanel Margin="-16,0,-19,-14">

<Label Content="Дата вылета:" FontWeight="Bold" Width="89"/>

<DatePicker x:Name="txtdpStartDate" Width="205"

Background="#F5F5F5" BorderBrush="#C0C0C0"

BorderThickness="1" Padding="5" Text="{Binding DepartureDate}"/>

<Label Content="Дата прибытия:" FontWeight="Bold" Width="108"/>

<DatePicker x:Name="txtdpEndDate" Width="205"

Background="#F5F5F5" BorderBrush="#C0C0C0"

BorderThickness="1" Padding="5" Text="{Binding ArrivalDate}"/>

<Label Content="Город отправления:" FontWeight="Bold" Width="131"/>

<TextBox x:Name="txtDepartureCityFrom" Width="250" Height="30"

Margin="0,5,0,15" Padding="5"

Background="#F5F5F5" BorderBrush="#C0C0C0"

BorderThickness="1" Text="{Binding FromCity}"/>

<Label Content="Город назначения:" FontWeight="Bold" Width="121"/>

<TextBox x:Name="txtDestinationCityTo" Width="250" Height="30"

Margin="0,5,0,15" Padding="5"

Background="#F5F5F5" BorderBrush="#C0C0C0"

BorderThickness="1" Text="{Binding ToCity}"/>

<Label Content="Класс обслуживания:" FontWeight="Bold" Width="138"/>

<TextBox x:Name="txtClassType" Width="250" Height="30"

Margin="0,5,0,15" Padding="5"

Background="#F5F5F5" BorderBrush="#C0C0C0"

BorderThickness="1" Text="{Binding ClassOfService}"/>

<Label Content="Цена:" FontWeight="Bold" Width="44"/>

<TextBox x:Name="txtPrice" Width="250" Height="30"

Margin="0,5,0,15" Padding="5" Background="#F5F5F5" BorderBrush="#C0C0C0"

BorderThickness="1" Text="{Binding Price}"/>

<Label Content="Авиакомпания:" FontWeight="Bold" Width="106"/>

<TextBox x:Name="txtAirline" Width="250" Height="30"

Margin="0,5,0,15" Padding="5"

Background="#F5F5F5" BorderBrush="#C0C0C0"

BorderThickness="1" Text="{Binding Airline}"/>

<Label Content="Тип самолета:" FontWeight="Bold" Width="97"/>

<TextBox x:Name="txtAircraftType" Width="250" Height="30"

Margin="0,5,0,20" Padding="5"

Background="#F5F5F5" BorderBrush="#C0C0C0"

BorderThickness="1" Text="{Binding AircraftType}"/>

</StackPanel>

<Button Content="Забронировать" Height="36"

Click="BookingButton\_Click" Background="#4CAF50"

Foreground="White" FontWeight="Bold" Margin="114,682,264,57"/>

<Button Content="Назад" Height="36"

Click="Button\_Click" Background="#4CAF50"

Foreground="White" FontWeight="Bold" Margin="261,682,113,57"/>

</Grid>

</Border>

*Логика для бронирования авиабилета*

public partial class WindowAirTicket : Window

{

private AirTicket airTicket = new AirTicket();

public WindowAirTicket()

{

InitializeComponent();

}

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

{

using (TurAgenstvooContext context = new TurAgenstvooContext())

{

airTicket.Airline = txtAirline.Text;

airTicket.AircraftType = txtAircraftType.Text;

airTicket.DepartureDate = DateOnly.Parse(txtdpStartDate.Text);

airTicket.ArrivalDate = DateOnly.Parse(txtdpEndDate.Text);

airTicket.FromCity = txtDepartureCityFrom.Text;

airTicket.ToCity = txtDestinationCityTo.Text;

airTicket.ClassOfService = txtClassType.Text;

airTicket.Price = Convert.ToDouble(txtPrice.Text);

try

{

context.AirTickets.Add(airTicket);

context.SaveChanges();

MessageBox.Show("Успешно");

}

catch (Exception ex)

{

MessageBox.Show("Ошибка");

}

}

}

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

{

this.Close();

}

}

*Логика для регистрации клиента*

public partial class WindowClients : Window

{

public WindowClients()

{

InitializeComponent();

}

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

{

using (TurAgenstvooContext context = new TurAgenstvooContext())

{

Client newClient = new Client

{

Fio = txtFio.Text,

Email = txtemail.Text,

PhoneNumber = txtPhone.Text,

Password = txtPassword.Text,

};

try

{

context.Clients.Add(newClient);

context.SaveChanges();

MessageBox.Show("Клиент добавлен");

}

catch (Exception ex)

{

MessageBox.Show("Ошибка");

}

}

}

}

*Код для бронирования отеля*

<Grid Margin="10,10,10,4">

<Border CornerRadius="10" Background="White" BorderThickness="1" BorderBrush="Black" Padding="20" Margin="0,0,0,10">

<Grid Margin="0,0,0,29">

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<TextBlock Grid.Row="0" Text="Бронирование гостиницы" FontSize="24" FontWeight="Bold"

HorizontalAlignment="Center" Margin="0,0,0,20"/>

<!-- Поля ввода -->

<Border Grid.Row="2" BorderThickness="1" BorderBrush="#C0C0C0" CornerRadius="5" Margin="0,5">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Label Grid.Row="0" Content="Местоположение:" FontWeight="Bold" Margin="10"/>

<TextBox Grid.Row="1" Name="txtLocation" Width="290" Background="#F5F5F5"

BorderBrush="#C0C0C0" BorderThickness="1" Padding="5" Margin="10" Text="{Binding HotelName1}"/>

</Grid>

</Border>

<Border Grid.Row="3" BorderThickness="1" BorderBrush="#C0C0C0" CornerRadius="5" Margin="0,5">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Label Grid.Row="0" Content="Рейтинг:" FontWeight="Bold" Margin="10"/>

<TextBox Grid.Row="1" Name="txtLocationRating" Width="290" Background="#F5F5F5"

BorderBrush="#C0C0C0" BorderThickness="1" Padding="5" Margin="10" Text="{Binding LocationRating}"/>

</Grid>

</Border>

<Border Grid.Row="4" BorderThickness="1" BorderBrush="#C0C0C0" CornerRadius="5" Margin="0,5">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Label Grid.Row="0" Content="Тип номера:" FontWeight="Bold" Margin="10"/>

<TextBox Grid.Row="1" Name="txtRoomType" Width="290" Background="#F5F5F5"

BorderBrush="#C0C0C0" BorderThickness="1" Padding="5" Margin="10" Text="{Binding RoomType}"/>

</Grid>

</Border>

<Border Grid.Row="5" BorderThickness="1" BorderBrush="#C0C0C0" CornerRadius="5" Margin="0,5">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Label Grid.Row="0" Content="Цена:" FontWeight="Bold" Margin="10"/>

<TextBox Grid.Row="1" Name="txtPrice" Width="290" Background="#F5F5F5"

BorderBrush="#C0C0C0" BorderThickness="1" Padding="5" Margin="10" Text="{Binding Price}"/>

</Grid>

</Border>

</Grid>

</Border>

<Button x:Name="btnSubmit" Content="Забронировать" Click="btnSubmit\_Click"

Background="#4CAF50" Foreground="White" FontWeight="Bold" Width="140" Margin="38,625,202,46"/>

<Button x:Name="Back" Content="Назад" Click="Back\_Click" Background="#4CAF50"

Foreground="White" FontWeight="Bold" Width="140"

Margin="190,625,50,46"/>

</Grid>

*Логика для бронирования отеля*

public partial class WindowHotelName : Window

{

private HotelName HotelName = new HotelName();

public WindowHotelName()

{

InitializeComponent();

}

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

{

using (TurAgenstvooContext context = new TurAgenstvooContext())

{

HotelName.HotelName1 = txtLocation.Text;

HotelName.LocationRating = txtLocationRating.Text;

HotelName.RoomType = txtRoomType.Text;

HotelName.Price = Convert.ToDouble(txtPrice.Text);

try

{

context.HotelNames.Add(HotelName);

context.SaveChanges();

MessageBox.Show("Гостиничный номер забронирован!");

}

catch (Exception ex)

{

MessageBox.Show("Ошибка");

}

}

}

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

{

this.Close();

}

}

*Код для окна бронирования*

<Grid>

<Grid>

<Border BorderBrush="Black" BorderThickness="1" CornerRadius="10" Margin="10"

Background="White">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*" />

<RowDefinition Height="Auto" />

</Grid.RowDefinitions>

<TextBlock Text="Добро пожаловать в Туристическое агентство!"

FontSize="19"

FontWeight="Bold"

Foreground="#333"

HorizontalAlignment="Center"

VerticalAlignment="Top"

Margin="0,10,0,0" Grid.Row="0"/>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Center"

Grid.Row="1"

Margin="10">

<Button Content="Авиабилеты" Click="Flights\_Click"

Height="30" Width="100"

Background="#4CAF50" Foreground="White"

FontWeight="Bold" FontSize="14" Margin="5"/>

<Button Content="Клиенты" Click="Clients\_Click"

Height="30" Width="100"

Background="#4CAF50" Foreground="White"

FontWeight="Bold" FontSize="14" Margin="5"/>

<Button Content="Туры" Click="Tour\_Click"

Height="30" Width="100"

Background="#4CAF50" Foreground="White"

FontWeight="Bold" FontSize="14" Margin="5"/>

<Button Content="Гостиницы" Click="Hotels\_Click"

Height="30" Width="100"

Background="#4CAF50" Foreground="White"

FontWeight="Bold" FontSize="14"/>

<Button Content="Выход" Click="End\_Click"

Height="30" Width="100"

Background="#4CAF50" Foreground="White"

FontWeight="Bold" FontSize="14" Margin="5"/>

</StackPanel>

</Grid>

</Border>

</Grid>

<DataGrid Margin="16,72,16,71" Name="dtgreserv" AutoGenerateColumns="False">

<DataGrid.Columns>

<DataGridTextColumn Header="IdReservation"

Binding="{Binding IdReservation}" Width="\*"/>

<DataGridTextColumn Header="HotelName"

Binding="{Binding IdHotelRoomNavigation.HotelName1}" Width="\*"/>

<DataGridTextColumn Header="Payment"

Binding="{Binding IdPaymentNavigation.TransationStatus}" Width="\*"/>

<DataGridTextColumn Header="Tour"

Binding="{Binding IdTourNavigation.Title}" Width="\*"/>

<DataGridTextColumn Header="Client"

Binding="{Binding IdClientNavigation.Fio}" Width="\*"/>

<DataGridTextColumn Header="AirTicket"

Binding="{Binding IdAirTicketNavigation.Airline}" Width="\*"/>

</DataGrid.Columns>

</DataGrid>

</Grid>

*Логика для окна бронирования*

public partial class WindowReservation : Window

{

public WindowReservation()

{

InitializeComponent();

LoadData();

}

public void LoadData()

{

using (TurAgenstvooContext context = new TurAgenstvooContext())

{

var data = context.Reservations.Include(o => o.IdClientNavigation).Include(o => o.IdAirTicketNavigation).

Include(o => o.IdTourNavigation).Include(o => o.IdPaymentNavigation).Include(o => o.IdHotelRoomNavigation).ToList();

dtgreserv.ItemsSource = data;

DataContext = data;

}

}

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

{

MainWindow mainWindow = new MainWindow();

mainWindow.Show();

this.Close();

}

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

{

WindowTour mainWindow = new WindowTour();

mainWindow.Show();

}

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

{

WindowHotelName windowHotelName = new WindowHotelName();

windowHotelName.Show();

}

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

{

WindowAirTicket windowAirTicket = new WindowAirTicket();

windowAirTicket.Show();

}

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

{

WindowClients windowClients = new WindowClients();

windowClients.Show();

}

}

*Код для бронирования тура*

<Grid Margin="10">

<Border CornerRadius="10" Background="White" BorderThickness="1"

BorderBrush="Black" Padding="20" Height="703" VerticalAlignment="Bottom" Margin="0,0,0,10">

<StackPanel Margin="10,10,10,161">

<TextBlock Text="Информация о туре" FontSize="24" FontWeight="Bold"

HorizontalAlignment="Center" Margin="0,0,0,20"/>

<Border BorderThickness="1" BorderBrush="#C0C0C0" CornerRadius="5" Margin="0,5">

<StackPanel Margin="10">

<Label Content="Название тура:" FontWeight="Bold"/>

<TextBox x:Name="txtTitle" Width="205"

Background="#F5F5F5" BorderBrush="#C0C0C0"

BorderThickness="1" Padding="5" Text="{Binding Title}"/>

</StackPanel>

</Border>

<Border BorderThickness="1" BorderBrush="#C0C0C0" CornerRadius="5" Margin="0,5">

<StackPanel Margin="10">

<Label Content="Описание тура:" FontWeight="Bold"/>

<TextBox x:Name="txtDescription" Width="205"

Background="#F5F5F5" BorderBrush="#C0C0C0"

BorderThickness="1" Padding="5" Text="{Binding Description}"/>

</StackPanel>

</Border>

<Border BorderThickness="1" BorderBrush="#C0C0C0" CornerRadius="5" Margin="0,5">

<StackPanel Margin="10">

<Label Content="Цена тура:" FontWeight="Bold"/>

<TextBox x:Name="txtPrice" Width="205"

Background="#F5F5F5" BorderBrush="#C0C0C0"

BorderThickness="1" Padding="5" Text="{Binding Price}"/>

</StackPanel>

</Border>

<Border BorderThickness="1" BorderBrush="#C0C0C0" CornerRadius="5" Margin="0,5">

<StackPanel Margin="10">

<Label Content="Количество мест:" FontWeight="Bold"/>

<TextBox x:Name="txtSeats" Width="205"

Background="#F5F5F5" BorderBrush="#C0C0C0"

BorderThickness="1" Padding="5" Text="{Binding AvailablePlaces}"/>

</StackPanel>

</Border>

</StackPanel>

</Border>

<Button Name="btnSave" Content="Сохранить"

HorizontalAlignment="Left" Margin="36,458,0,189"

Click="btnSave\_Click" Background="#4CAF50"

Foreground="White" FontWeight="Bold" Height="40" Width="140"/>

<Button Name="btnBack" Content="Назад"

HorizontalAlignment="Left" Margin="199,462,0,192"

Click="btnBack\_Click" Background="#4CAF50"

Foreground="White" FontWeight="Bold" Height="40" Width="140"/>

</Grid>

*Логика для бронирования тура*

public partial class WindowTour : Window

{

public WindowTour()

{

InitializeComponent();

}

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

{

using (TurAgenstvooContext context = new TurAgenstvooContext())

{

if (string.IsNullOrEmpty(txtTitle.Text) ||

string.IsNullOrEmpty(txtDescription.Text) || string.IsNullOrEmpty(txtPrice.Text) ||

string.IsNullOrEmpty(txtSeats.Text))

{

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

return;

}

else

{

string title = txtTitle.Text;

string description = txtDescription.Text;

float price = float.Parse(txtPrice.Text, NumberStyles.Any, CultureInfo.InvariantCulture);

int AvailablePlaces = int.Parse(txtSeats.Text);

Tour tour = new Tour

{

Title = title,

Description = description,

Price = price,

AvailablePlaces = AvailablePlaces

};

try

{

context.Tours.Add(tour);

context.SaveChanges();

MessageBox.Show("Тур успешно сохранен!", "Успешная запись", MessageBoxButton.OK, MessageBoxImage.Information);

ClearFields();

}

catch (FormatException ex)

{

MessageBox.Show("Некорректный формат данных. Проверьте вводимые значения.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

}

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

{

this.Close();

}

private void ClearFields()

{

txtTitle.Clear();

txtDescription.Clear();

txtPrice.Clear();

txtSeats.Clear();

}

}

# 5.Разработка Api

**API (Application programming interface) —** это**набор определённых правил и команд, благодаря которым приложения связываются между собой и обмениваются данными.**

Он предлагает разработчикам набор запросов, протоколов, подпрограмм и инструментов для того, чтобы быстро извлекать и обмениваться информацией между приложениями.

TurAgenstvoController

using Microsoft.AspNetCore.Mvc;

using Microsoft.EntityFrameworkCore;

using WebApplication1.Models;

namespace WebApplication1.Controllers

{

[Route("api/[controller]")]

[ApiController]

public class TurAgenstvoController : Controller

{

private readonly TurAgenstvoContext \_context;

public TurAgenstvoController(TurAgenstvoContext context)

{

\_context = context;

}

// GET: api/products

[HttpGet]

public async Task<ActionResult<IEnumerable<Client>>> GetAll()

{

return await \_context.Clients.ToListAsync();

}

// GET: api/products/{id}

[HttpGet("{id}")]

public async Task<ActionResult<Client>> GetById(int Id)

{

var Clients = await \_context.Clients.FindAsync(Id);

if (Clients is null)

return NotFound();

return Clients;

}

// POST: api/products

[HttpPost]

public async Task<IActionResult> Create(Client Client)

{

\_context.Clients.Add(Client);

await \_context.SaveChangesAsync();

return CreatedAtAction(nameof(GetById), new { id = Client.IdClient }, Client);

}

// PUT: api/products/5

// The method below will handle the PUT requests

[HttpPut("{id}")]

public async Task<IActionResult> PutIdClient(int id, Client Client)

{

if (id != Client.IdClient)

{

return BadRequest();

}

\_context.Entry(Client).State = EntityState.Modified;

try

{

await \_context.SaveChangesAsync();

}

catch (DbUpdateConcurrencyException)

{

if (!ClientExists(id))

{

return NotFound();

}

else

{

throw;

}

}

return NoContent();

}

private bool ClientExists(int id)

{

return \_context.Clients.Any(e => e.IdClient == id);

}

// DELETE: api/products/5

[HttpDelete("{id}")]

public async Task<IActionResult> DeleteClient(int id)

{

var Client = await \_context.Clients.FindAsync(id);

if (Client == null)

{

return NotFound();

}

\_context.Clients.Remove(Client);

await \_context.SaveChangesAsync();

return NoContent();

}

}

}

*GET*

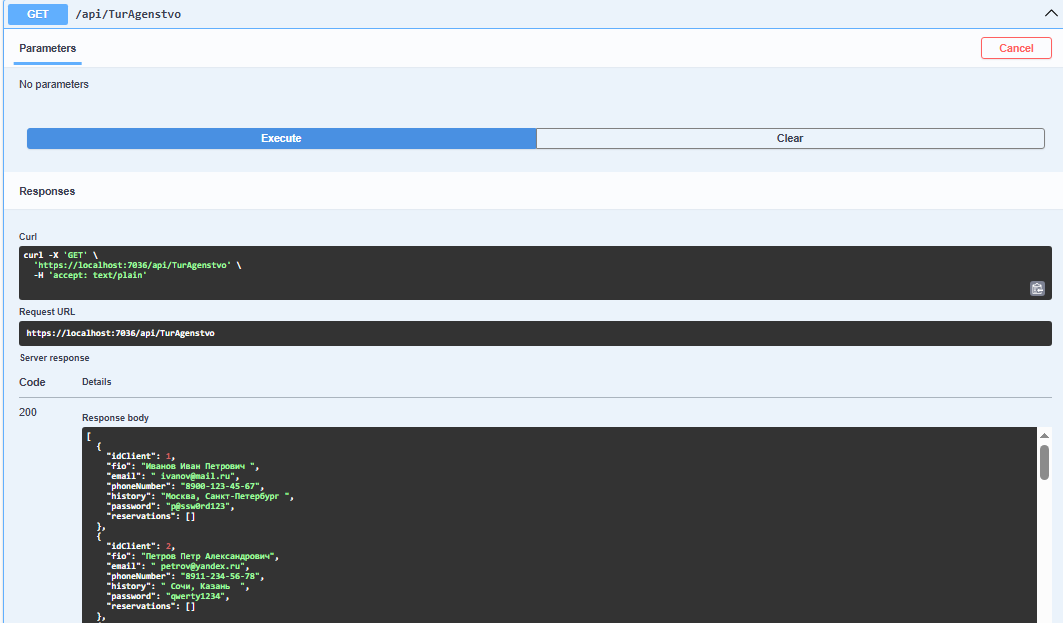


Рис.17«Просмотр всех данных»

*DELETE*

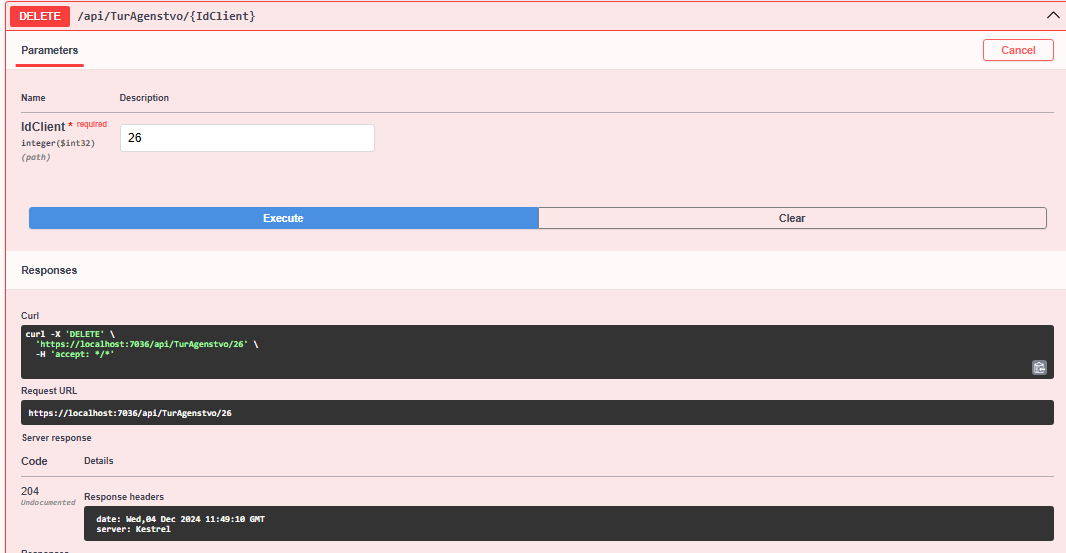


Рис18. «Удаление по id»

*GET по id*

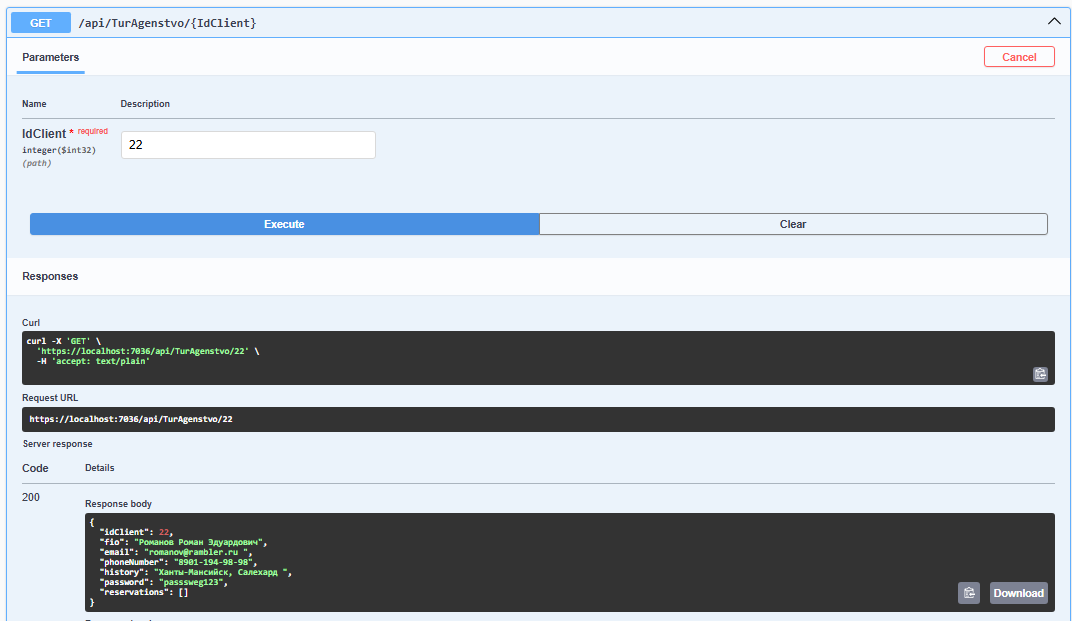


Рис.19 «Просмотр по id»

*POST*

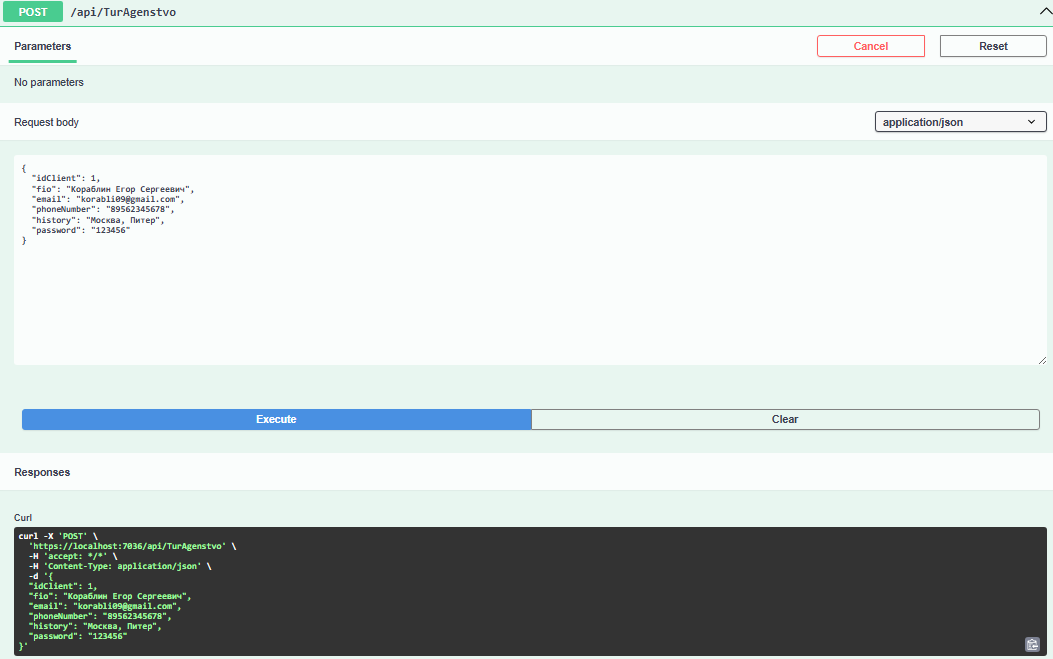


Рис.20 «Добавление клиента»

*PUT*

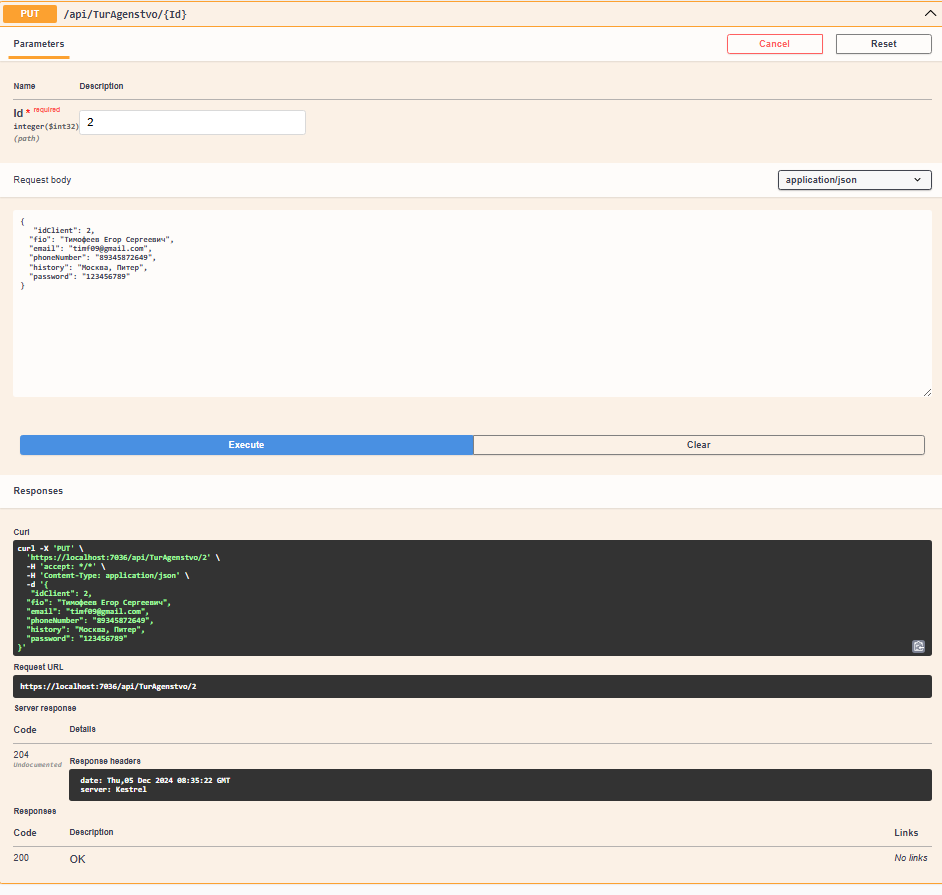


Рис. 21«Обновление клиента»

# 6.Разработка библиотеки класов

*Описание класса*

public class Class1

{

private readonly TurAgenstvooContext \_context;

public Class1(TurAgenstvooContext context)

{

\_context = context;

}

public string Signin(string Email, string Password)

{

if (string.IsNullOrEmpty(Email) || string.IsNullOrEmpty(Password))

{

return "Введите почту и пароль!";

}

var clients = \_context.Clients.FirstOrDefault(u => u.Email == Email && u.Password == Password);

if (clients != null)

{

return "Вход в систему";

}

else

{

return "Неверная почта или пароль!";

}

}

}

*Код тестирования*

[TestClass]

public sealed class Test1

{

private TurAgenstvooContext \_context;

private Class1 \_authorization;

[TestMethod]

public void Initialize()

{

var options = new DbContextOptionsBuilder<TurAgenstvooContext>()

.UseSqlServer("Server=44-1\\SQLEXPRESS;Database=TurAgenstvoo;TrustServerCertificate=True;Trusted\_Connection=True;")

.Options;

\_context = new TurAgenstvooContext(options);

\_context.Database.EnsureCreated();

\_authorization = new Class1(\_context);

}

[TestMethod]

public void Login\_ShouldReturnEnter\_ForValidClientCredentials()

{

string email = "fedyaeva@gmail.com"; // Логин

string password = "pass2023!"; // Пароль

string result = \_authorization.Signin(email, password);

Assert.AreEqual("Вход в систему", result);

}

[TestMethod]

public void Login\_ShouldReturnError\_ForValidClientCredentials()

{

string login = "12345678"; // Логин

string password = "12345678"; // Пароль

string result = \_authorization.Signin(login, password);

Assert.AreEqual("Вход в систему", result);

}

}

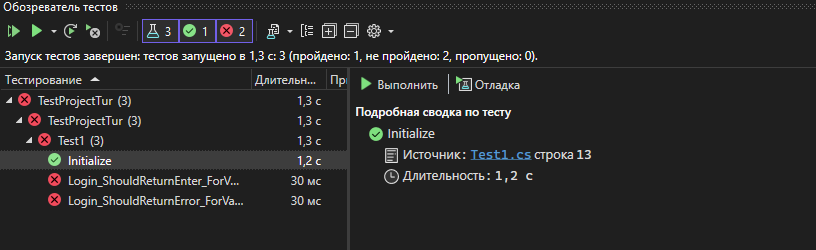


Рис.22 «Тест»

# 7.Заключение

В ходе выполнения данного проекта была проведена разработка технического задания, проектирование и создание базы данных, а также разработаны интерфейс приложения, программные модели, API и библиотека классов. Все этапы разработки были успешно завершены, что позволило создать функциональное приложение, соответствующее требованиям технического задания.

Разработанная база данных обеспечивает надежное хранение и обработку информации, а удобный интерфейс позволяет пользователям легко взаимодействовать с системой. Программные модели обеспечивают логику работы приложения, а API предоставляет возможность интеграции с другими системами. Библиотека классов облегчает дальнейшее расширение и поддержку системы.