Министерство образования и науки Российской Федерации Федеральное агентство по образованию Федеральное государственное бюджетное образовательное учреждение высшего образования «Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Лабораторная работа №4

по курсу «Базы данных»

Выполнил студент группы ИВТ-31\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Бикметов И.Р/

Проверил доцент кафедры ЭВМ \_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Клюкин В.Л./

Киров 2025

1. Задание

- разработать приложение на C# для работы с БД

- приложение должно выполнять следующие действия: отображение БД, добавление записи, удаление записи, сортировка БД

2. Диаграмма классов для работы с БД

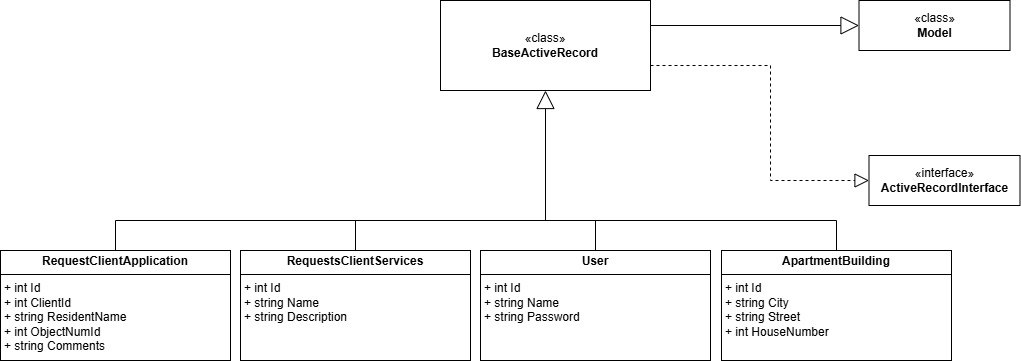


Рисунок 1 – диаграмма классов

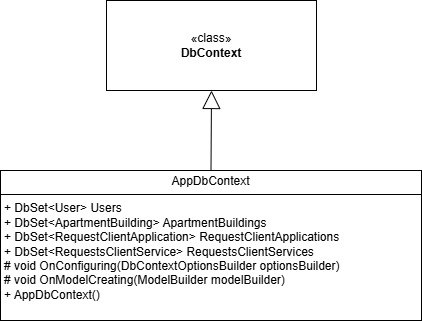


Рисунок 2 – диаграмма классов

3. Листинг кода

using DBAdmin.entity;

using Microsoft.EntityFrameworkCore;

using static Microsoft.EntityFrameworkCore.DbLoggerCategory.Database;

namespace DBAdmin.context

{

public class AppDbContext : DbContext

{

public DbSet<User> Users { get; set; } = null!;

public DbSet<UserPosition> UserPositions { get; set; } = null!;

public DbSet<RequestsClientService> RequestsClientServices { get; set; } = null!;

public DbSet<RequestServiceToPosition> RequestServiceToPositions { get; set; } = null!;

public DbSet<ApartmentBuilding> ApartmentBuildings { get; set; } = null!;

public DbSet<ApartmentBuildingObjectType> ApartmentBuildingObjectTypes { get; set; } = null!;

public DbSet<RequestsService> RequestsServices { get; set; } = null!;

public DbSet<RequestClientApplication> RequestClientApplications { get; set; } = null!;

public DbSet<RequestPerformerService> RequestPerformerServices { get; set; } = null!;

public AppDbContext()

{

Database.EnsureCreated();

}

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseNpgsql("Host=localhost;Port=5432;Database=mydb;Username=postgres;Password=13579");

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

// RequestServiceToPosition -> RequestClientService

modelBuilder.Entity<RequestServiceToPosition>()

.HasOne(r => r.Service)

.WithMany()

.HasForeignKey(r => r.ServiceId);

// RequestServiceToPosition -> UserPosition

modelBuilder.Entity<RequestServiceToPosition>()

.HasOne(r => r.Position)

.WithMany()

.HasForeignKey(r => r.PositionId);

//RequestsService -> RequestServiceToPosition

modelBuilder.Entity<RequestsService>()

.HasOne(r => r.ServiceToPosition)

.WithMany()

.HasForeignKey(r => r.ServiceId);

//RequestsService -> RequestClientApplication

modelBuilder.Entity<RequestsService>()

.HasOne(r => r.ClientApplication)

.WithMany()

.HasForeignKey(r => r.RequestId);

// RequestClientApplication -> User

modelBuilder.Entity<RequestClientApplication>()

.HasOne(r => r.Client)

.WithMany()

.HasForeignKey(r => r.ClientId);

// RequestClientApplication -> ApartmentBuildingObjectType

modelBuilder.Entity<RequestClientApplication>()

.HasOne(r => r.ObjectType)

.WithMany()

.HasForeignKey(r => r.ObjectTypeId);

// RequestClientApplication -> ApartmentBuilding

modelBuilder.Entity<RequestClientApplication>()

.HasOne(r => r.Apartment)

.WithMany()

.HasForeignKey(r => r.ObjectNumId);

// RequestPerformerService -> RequestClientApplication

modelBuilder.Entity<RequestPerformerService>()

.HasOne(r => r.ClientApplication)

.WithMany()

.HasForeignKey(r => r.RequestId);

// RequestPerformerService -> RequestsService

modelBuilder.Entity<RequestPerformerService>()

.HasOne(r => r.RequestService)

.WithMany()

.HasForeignKey(r => r.ServiceOnRequestId);

// RequestPerformerService -> User

modelBuilder.Entity<RequestPerformerService>()

.HasOne(r => r.Implementer)

.WithMany()

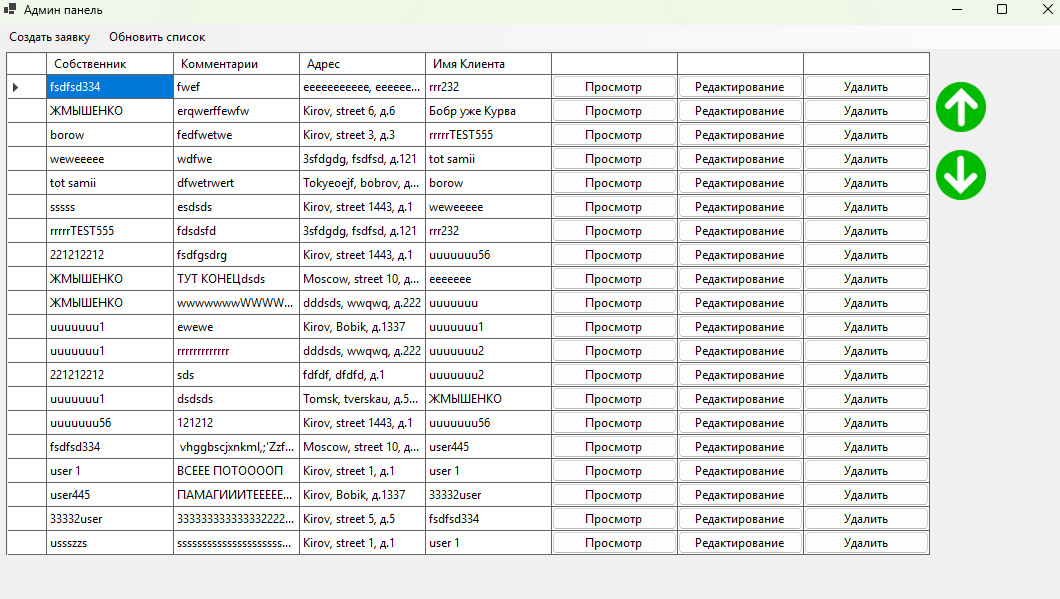
.HasForeignKey(r => r.ImplementerId);

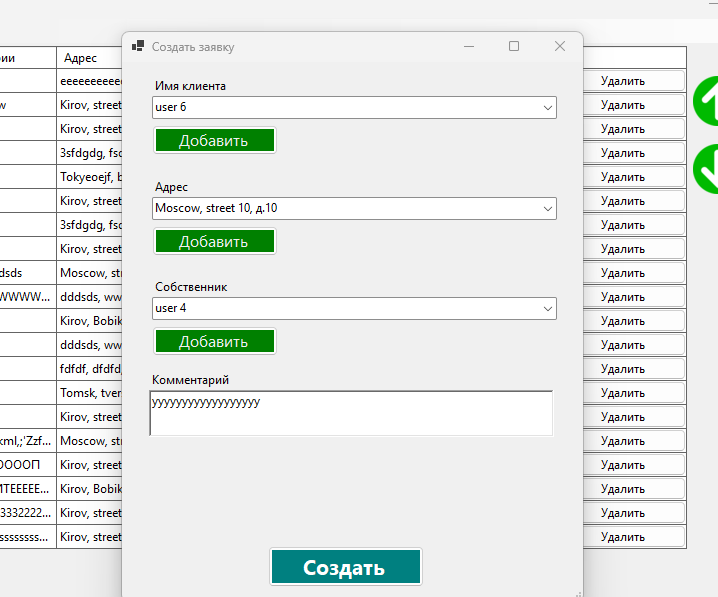
}

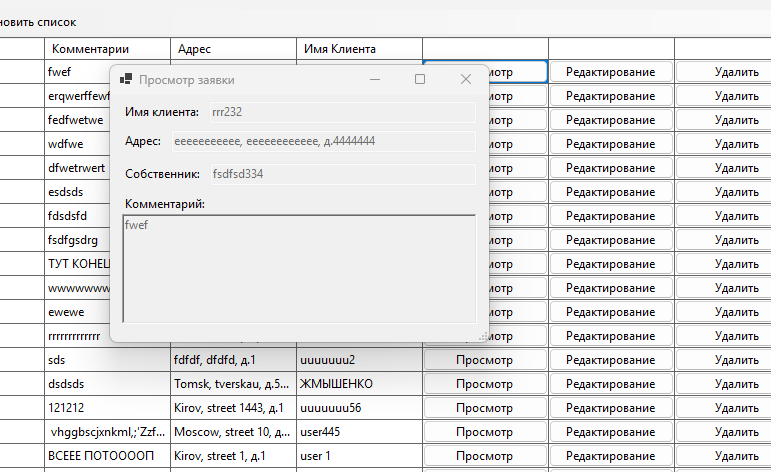
}

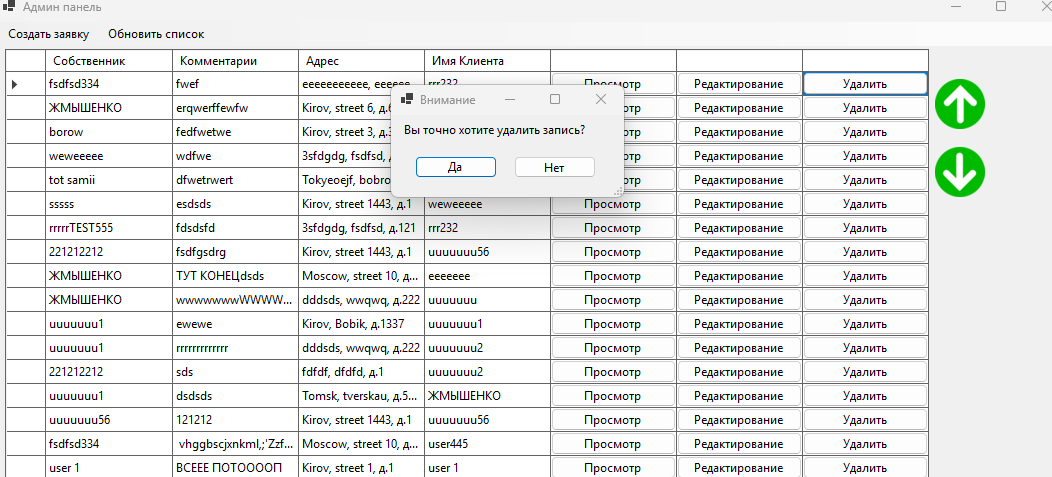
}

4. Экранные формы









5. Вывод

В ходе выполнения лабораторной работы были получены навыки разработки приложения для работы с бд.