

Міністерство освіти і науки України
НТУУ «КПІ ім. Ігоря Сікорського»
Навчально-науковий інститут атомної та теплової енергетики
Кафедра цифрових технологій в енергетиці

Лабораторна робота №6
з дисципліни «Декларативне програмування графічних
інтерфейсів»
«Творче завдання з розробки застосунку WPF з
використанням баз даних»
Варіант №19

Виконав: Студент групи ТР-12
Ковальов Олександр
Перевірів: доцент, к.ф.-м.н.
Тарнавський Ю. А.

Мета роботи. Закріпити базові знання і вміння з використання в WPF технології EF Core і Scaffolding.

Завдання. Розробити повнофункціональний застосунок WPF з власною базою даних.

Файли XAML та приєднаного коду додайте у звіт з виконання лабораторної роботи.

AlarmRepository.cs:

```
using Lab6.Context;
using Lab6.Entities;
using Microsoft.EntityFrameworkCore;

namespace Lab6.Repositories;

public static class AlarmRepository
{
    private static ApplicationDbContext _dbContext;

    public static DbSet<Alarm> AlarmSet { get; }

    static AlarmRepository()
    {
        _dbContext = new ApplicationDbContext();
        AlarmSet = _dbContext.Alarms;

        CheckAlarmRelevance();
    }

    private static async void SaveChanges()
    {
        await _dbContext.SaveChangesAsync();
    }

    public static void CheckAlarmRelevance()
    {
        var forDeletion = new List<Guid>();

        foreach (var record in AlarmSet)
        {
            if (record.Datetime.CompareTo(DateTime.Now) <= 0)
                forDeletion.Add(new Guid(record.Id));
        }

        bool areThereDeprecatedAlarms = forDeletion.Count != 0;

        if (!areThereDeprecatedAlarms) return;

        foreach (var id in forDeletion)
        {
            RemoveRecord(id);
        }

        SaveChanges();
    }

    public static Alarm AddRecord(string title, DateTime dateTime)
    {
        var alarm = new Alarm(title, dateTime);

        AlarmSet.Add(alarm);

        SaveChanges();

        return alarm;
    }

    public static Alarm EditRecord(Guid recordId, string title, DateTime dateTime, bool isAlarmEnabled)
    {
        var record = GetRecord(recordId);
        record.Title = title;
        record.Datetime = dateTime;
        record.IsAlarmEnabled = isAlarmEnabled;

        SaveChanges();
    }
}
```

```

        return record;
    }

    public static Alarm GetRecord(Guid id)
    {
        var alarm = AlarmSet.Find(id.ToByteArray());

        if (alarm is null)
            throw new ArgumentException("There's no alarms with that GUID");

        return alarm;
    }

    public static Alarm RemoveRecord(Guid id)
    {
        var alarm = GetRecord(id);

        AlarmSet.Remove(alarm);

        SaveChanges();

        return alarm;
    }
}

```

Alarm.cs:

```

namespace Lab6.Entities;

public partial class Alarm
{
    public byte[] Id { get; set; } = null!;

    public string Title { get; set; } = null!;

    public bool IsAlarmEnabled { get; set; }

    public DateTime Datetime { get; set; }

    public Alarm()
    {
    }

    public Alarm(string title, DateTime dateTime)
    {
        Id = Guid.NewGuid().ToByteArray();
        Title = title;
        IsAlarmEnabled = true;
        Datetime = dateTime;
    }
}

```

ApplicationDbContext.cs:

```

using System.Configuration;
using Lab6.Entities;
using Microsoft.EntityFrameworkCore;

namespace Lab6.Context;

public partial class ApplicationDbContext : DbContext
{
    public ApplicationDbContext()
    {
    }

    public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
        : base(options)
    {
    }

    public virtual DbSet<Alarm> Alarms { get; set; }

    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
    }
}

```

```

var connectionString = ConfigurationManager.
    ConnectionStrings["ApplicationDatabase"].ConnectionString;

var dataDirectory = AppDomain.CurrentDomain.GetData("DataDirectory")!.ToString();
if (!string.IsNullOrEmpty(dataDirectory))
{
    connectionString = connectionString
        .Replace("|DataDirectory|", dataDirectory);
}

optionsBuilder.UseSqlite(connectionString);
}

protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder.Entity<Alarm>(entity =>
    {
        entity.Property(e => e.Id).HasColumnName("id");
        entity.Property(e => e.Datetime)
            .HasColumnType("DATETIME")
            .HasColumnName("datetime");
        entity.Property(e => e.IsAlarmEnabled)
            .HasColumnType("BOOLEAN")
            .HasColumnName("is_alarm_enabled");
        entity.Property(e => e.Title)
            .HasColumnType("NVARCHAR(100)")
            .HasColumnName("title");
    });

    modelBuilder.OnModelCreatingPartial(modelBuilder);
}

partial void OnModelCreatingPartial(ModelBuilder modelBuilder);
}

```

CreateTables.sql:

```

CREATE TABLE Alarms
(
    id          BLOB PRIMARY KEY NOT NULL,
    title       NVARCHAR(100)    NOT NULL,
    is_alarm_enabled BOOLEAN      NOT NULL,
    datetime    DATETIME         NOT NULL
);

```

Github Repository: <https://github.com/xairaven/KPI-DPGI/tree/Lab6/Lab6>

Висновок: за результатами виконання цієї лабораторної роботи були набуті практичні навички роботи з Entity Framework.