Laboratorium P4 2022/2023

GR. 2a

Projekt 3
Obsługa własnej bazy danych

Przygotował:

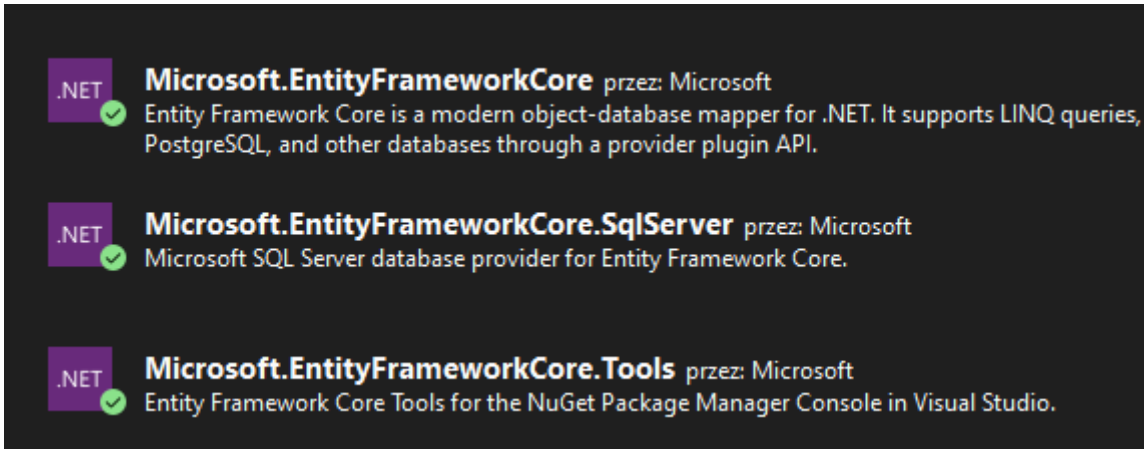
Jakub Janik

1. Cel

Celem projektu było stworzenie aplikacji umożliwiającej korzystanie z bazy danych oraz dodawanie elementów do niej.

2. Technologie

Projekt został stworzony w technologii .NET Freamwork w języku C#. Część wizualna została wykonana w WPF. Wykorzystana baza danych została utworzona w SSMS. Oraz zostały użyte następujące pakiety z Nuget



3. Klasy

```
namespace p4projekt3.Model
{
    [Table("Autobus")]
    Odwołania: 4
    public class AutobusModel
    {
        [Key]
        Odwołania: 3
        public string Nr_rejestracji { get; set; }

        1 odwołanie
        public virtual ICollection<KierowcaModel> KierowcaId { get; set; } = new List<KierowcaModel>();

        [Column(TypeName="nvarhcar(30)")]
        Odwołania: 3
        public string Model { get; set; }

        [Column(TypeName ="nvarchar(20)")]
        Odwołania: 3
        public string Marka { get; set; }

        [Column(TypeName="decimal(2)")]
        Odwołania: 3
        public string Rocznik { get; set; }

    }
}
```

Klasa Autobus która posiada klucz główny Nr_rejestracji oraz klucz obcy który zawiera Id kierowcy osiada osobne atrybuty niż klasa kierowca

```
namespace p4projekt3.Model
{
    [Table("Kierowca")]
    Odwołania: 5
    public class KierowcaModel
    {
        [Key]
        Odwołania: 0
        public int KierowcaId { get; set; }

        [Column(TypeName ="nvarchar(40)")]
        Odwołania: 3
        public string FirstName { get; set; }

        [Column(TypeName ="nvarchar(50)")]
        Odwołania: 3
        public string LastName { get; set; }

        [Column(TypeName ="int")]
        Odwołania: 3
        public int Nr_tel { get; set; }

        [Column(TypeName ="nvarchar(120)")]
        Odwołania: 3
        public string Adres { get; set; }

        [Column(TypeName ="date")]
        Odwołania: 3
        public DateTime Data_ur { get; set; }
    }
}
```

Klasa kierowca zawiera tylko klucz główny oraz posiada osobne atrybuty względem klasy Autobus

```
namespace p4projekt3.Model
{
    1 odwołanie
    public static class Configuration
    {
        1 odwołanie
        public static string ConnectionString { get; } =
            "Data Source=MAINPC\\SQLEXPRESS;Integrated Security=True;Connect Timeout=30" +
            ";Encrypt=False;Trust Server Certificate=False;Application Intent=ReadWrite;" +
            "Multi Subnet Failover=False";
    }
}
```

Klasa Configuration posiada tylko connection string

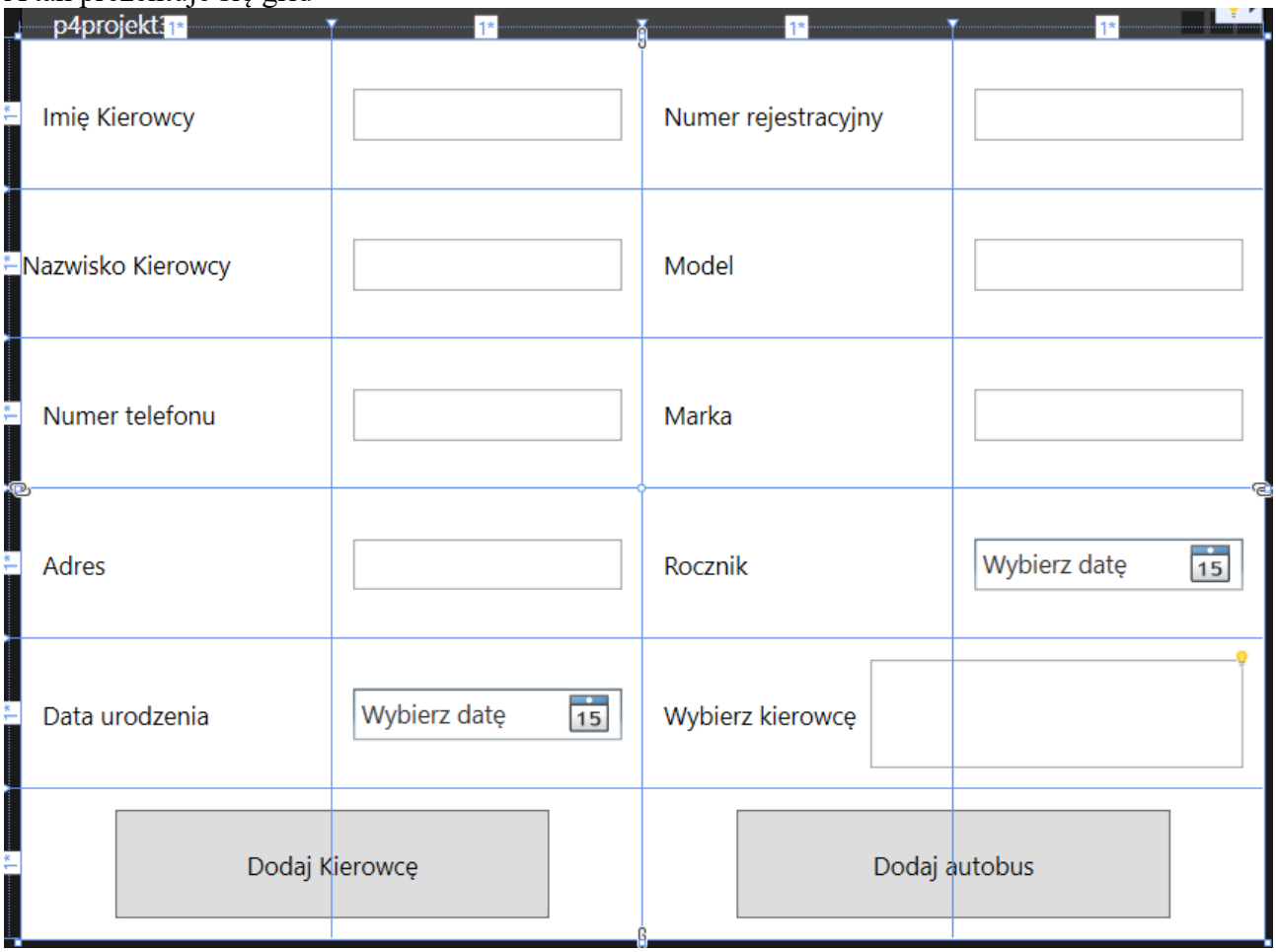
```
namespace p4projekt3.Model
{
    Odwołania: 5
    public class WorkDBContext : DbContext
    {
        Odwołania: 2
        public WorkDBContext()
        {

        }
        1 odwołanie
        public DbSet<KierowcaModel>Kierowca { get; set; }
        1 odwołanie
        public DbSet<AutobusModel>Autobus { get; set; }

        Odwołania: 0
        public WorkDBContext(DbContextOptions<WorkDBContext> options) : base(options) { }
        Odwołania: 0
        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            modelBuilder.Entity<AutobusModel>()
                .HasMany(x => x.KierowcaId);
        }
        Odwołania: 0
        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            optionsBuilder.UseSqlServer(Configuration.ConnectionString);
        }
    }
}
```

Klasa WorkDBContext posiada deklaracje związku 1 do N

4. Grid
Poniżej znajduje się cały kod grid’a

```
        <TextBlock Grid.Row="0" Grid.Column="0" VerticalAlignment="Center" Margin="10">Imię Kierowcy</TextBlock>
    <TextBox x:Name="ImieTextBox"  Grid.Row="0" Grid.Column="1" MaxHeight="25" Margin="10" Text="{Binding FirstName}" TextChanged="TextBox_TextChanged"></TextBox>

        <TextBlock Grid.Row="1" Grid.Column="0" VerticalAlignment="Center" >Nazwisko Kierowcy</TextBlock>
    <TextBox x:Name="NazwiskoTextBox" Grid.Row="1" Grid.Column="1" MaxHeight="25" Margin="10" Text="{Binding LastName}"></TextBox>

    <TextBlock Grid.Row="2" Grid.Column="0" VerticalAlignment="Center" Margin="10" >Numer telefonu</TextBlock>
    <TextBox x:Name="NrTelTextBox" Grid.Row="2" Grid.Column="1" MaxHeight="25" Margin="10" Text="{Binding Nr_tel}"></TextBox>

    <TextBlock Grid.Row="3" Grid.Column="0" VerticalAlignment="Center" Margin="10">Adres</TextBlock>
    <TextBox x:Name="AdresTextBox" Grid.Row="3" Grid.Column="1" MaxHeight="25" Margin="10" Text="{Binding Adres}"></TextBox>

    <TextBlock Grid.Row="4" Grid.Column="0" VerticalAlignment="Center" Margin="10">Data urodzenia</TextBlock>
    <DatePicker x:Name="Data_urTextBox" Grid.Row="4" Grid.Column="1" MaxHeight="25" Margin="10" Text="{Binding Data_ur}"></DatePicker>

        <Button x:Name="Kierowca_button" Grid.Row="5" Margin="45,10,45,10" Grid.ColumnSpan="2" Click="Button_Click_kierowca">Dodaj Kierowcę</Button>

    <TextBlock Grid.Row="0" Grid.Column="2" VerticalAlignment="Center" Margin="10" >Numer rejestracyjny</TextBlock>
    <TextBox x:Name="Nr_rejestracjiTextBox" Grid.Row="0" Grid.Column="3" MaxHeight="25" Margin="10" Text="{Binding Nr_rejestracyjny}"></TextBox>

    <TextBlock Grid.Row="1" Grid.Column="2" VerticalAlignment="Center" Margin="10">Model</TextBlock>
    <TextBox x:Name="ModelTextBox" Grid.Row="1" Grid.Column="3" MaxHeight="25" Margin="10" Text="{Binding Model}" TextChanged="TextBox_TextChanged_1"></TextBox>

    <TextBlock Grid.Row="2" Grid.Column="2" VerticalAlignment="Center" Margin="10">Marka</TextBlock>
    <TextBox x:Name="MarkaTextBox" Grid.Row="2" Grid.Column="3" MaxHeight="25" Margin="10" Text="{Binding Marka}"></TextBox>

    <TextBlock Grid.Row="3" Grid.Column="2" VerticalAlignment="Center" Margin="10" >Rocznik</TextBlock>
    <DatePicker x:Name="RocznikDatePicker" Grid.Row="3" Grid.Column="3" MaxHeight="25" Margin="10" Text="{Binding Rocznik}"></DatePicker>

    <TextBlock Grid.Row="4" Grid.Column="2" VerticalAlignment="Center" Margin="10" >Wybierz kierowcę</TextBlock>
    <ListBox x:Name="KierowcyListBox" Grid.Row="4" Grid.Column="2" Grid.ColumnSpan="2" Margin="110,10,10,10" SelectionMode="Multiple"
             ItemsSource="{Binding ElementName=ImieTextBox, Mode=OneWay}">
        <ListBox.ItemTemplate>
            <DataTemplate>
                <TextBlock Text="{Binding FullName}" />
            </DataTemplate>
        </ListBox.ItemTemplate>
    </ListBox>

    <Button x:Name="Autobus_button" Grid.Row="5" Grid.Column="2" Margin="45,10,45,10" Grid.ColumnSpan="2" Click="Button_Click_autobus" >Dodaj autobus</Button>
    </Grid>
```

A tak prezentuje się grid



5. Przyciski

Za dodanie kierowców odpowiada ten przycisk

```
private void Button_Click_kierowca(object sender, RoutedEventArgs e)
{
    KierowcaModel kierowca = new KierowcaModel()
    {
        FirstName = ImieTextBox.Text,
        LastName = NazwiskoTextBox.Text,
        Nr_tel = int.Parse(NrTelTextBox.Text),
        Adres = AdresTextBox.Text,
        Data_ur = Data_urTextBox.SelectedDate.Value
    };

    using (var dbContext = new WorkDBContext())
    {
        dbContext.Kierowca.Add(kierowca);
        dbContext.SaveChanges();
    }

    ImieTextBox.Text = string.Empty;
    NazwiskoTextBox.Text = string.Empty;
    NrTelTextBox.Text = string.Empty;
    AdresTextBox.Text = string.Empty;
    Data_urTextBox.SelectedDate = null;

    MessageBox.Show("Kierowca został dodany.");
}
```

A z dodanie autobusów dopowiada ten przycisk

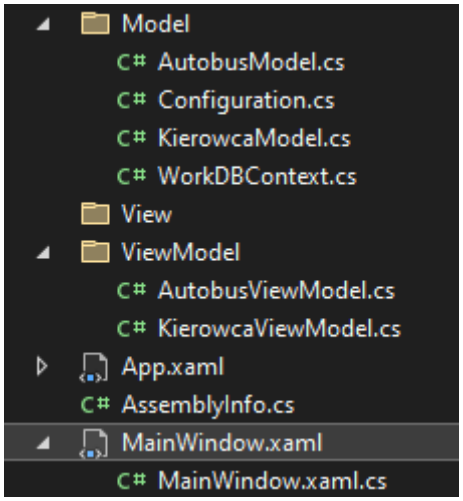
```
private void Button_Click_autobus(object sender, TextChangedEventArgs e)
{
    AutobusModel autobus = new AutobusModel()
    {
        Nr_rejestracji = Nr_rejestracjiTextBox.Text,
        Model = ModelTextBox.Text,
        Marka = MarkaTextBox.Text,
        Rocznik = RocznikDatePicker.Text
    };



    using (var dbContext = new WorkDBContext())
    {
        dbContext.Autobus.Add(autobus);
        dbContext.SaveChanges();
    }
    Nr_rejestracjiTextBox.Text = string.Empty;
    ModelTextBox.Text = string.Empty;
    MarkaTextBox.Text = string.Empty;
    RocznikDatePicker.SelectedDate = null;
    KierowcyListBox.SelectedItems.Clear();

    MessageBox.Show("Autobus został dodany.");
}
```

6. Code behind

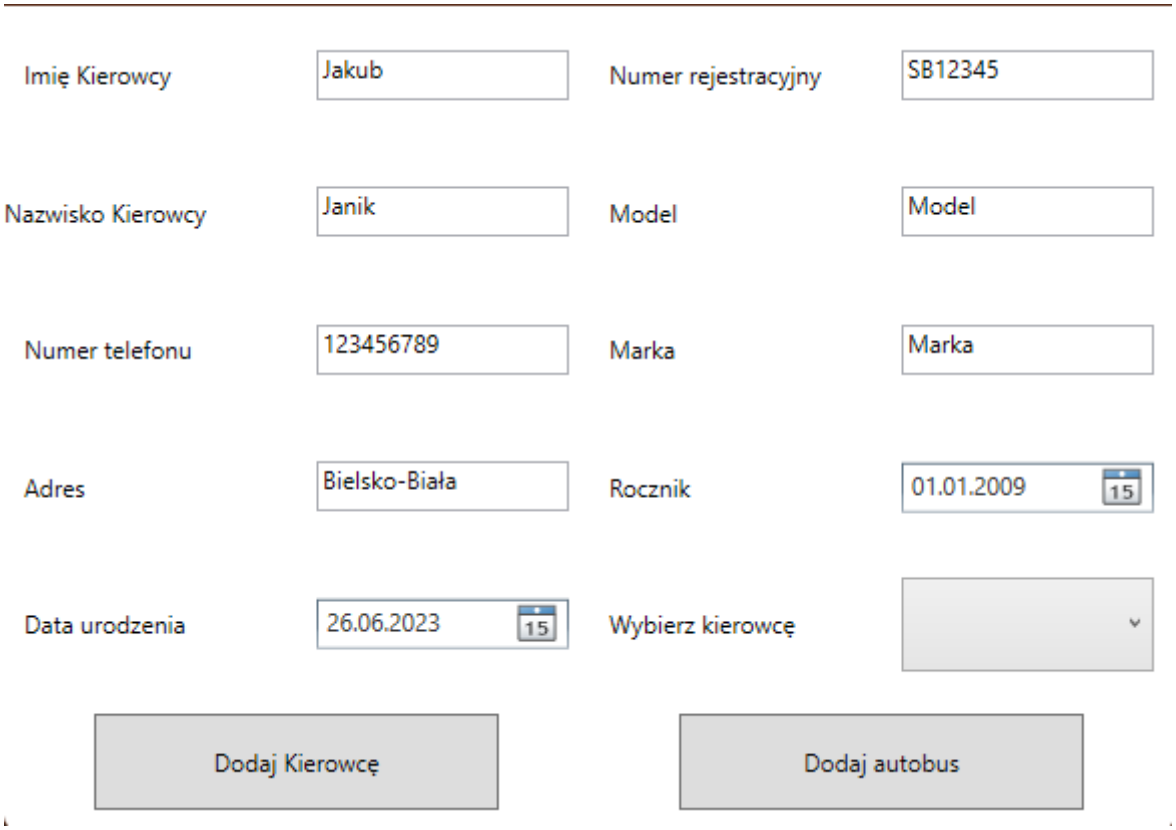
```
namespace p4projekt3
{
    /// <summary>
    /// Interaction logic for App.xaml
    /// </summary>
    Odwołania: 3
    public partial class App : Application
    {
    }
}
```

Jest pusty tak jak to powinno wyglądać w modelu MVVM

7. Lista wszystkich plików



8. Wynik działania aplikacji



Powyższy projekt przedstawia okno które pokazuje się po uruchomieniu aplikacji.
Po dodaniu kierowców możemy ich wybrać z listy wyboru która zaciąga dane z lokalnej bazy danych. Niestety z powodów technicznych nie mogłem zaprezentować działania listy.