Baza Jednostki Wojskowej

Spis treści

1. Wstęp

Przedstawiona baza danych umożliwia proste zarządzanie jednostką wojskową. W bazie zawarte są wszystkie dane każdego żołnierza, sprzętu jaki jest na wyposażeniu jednostki, oraz informacje odnośnie stopni i specjalizacji.

2. Analiza wymagań systemu

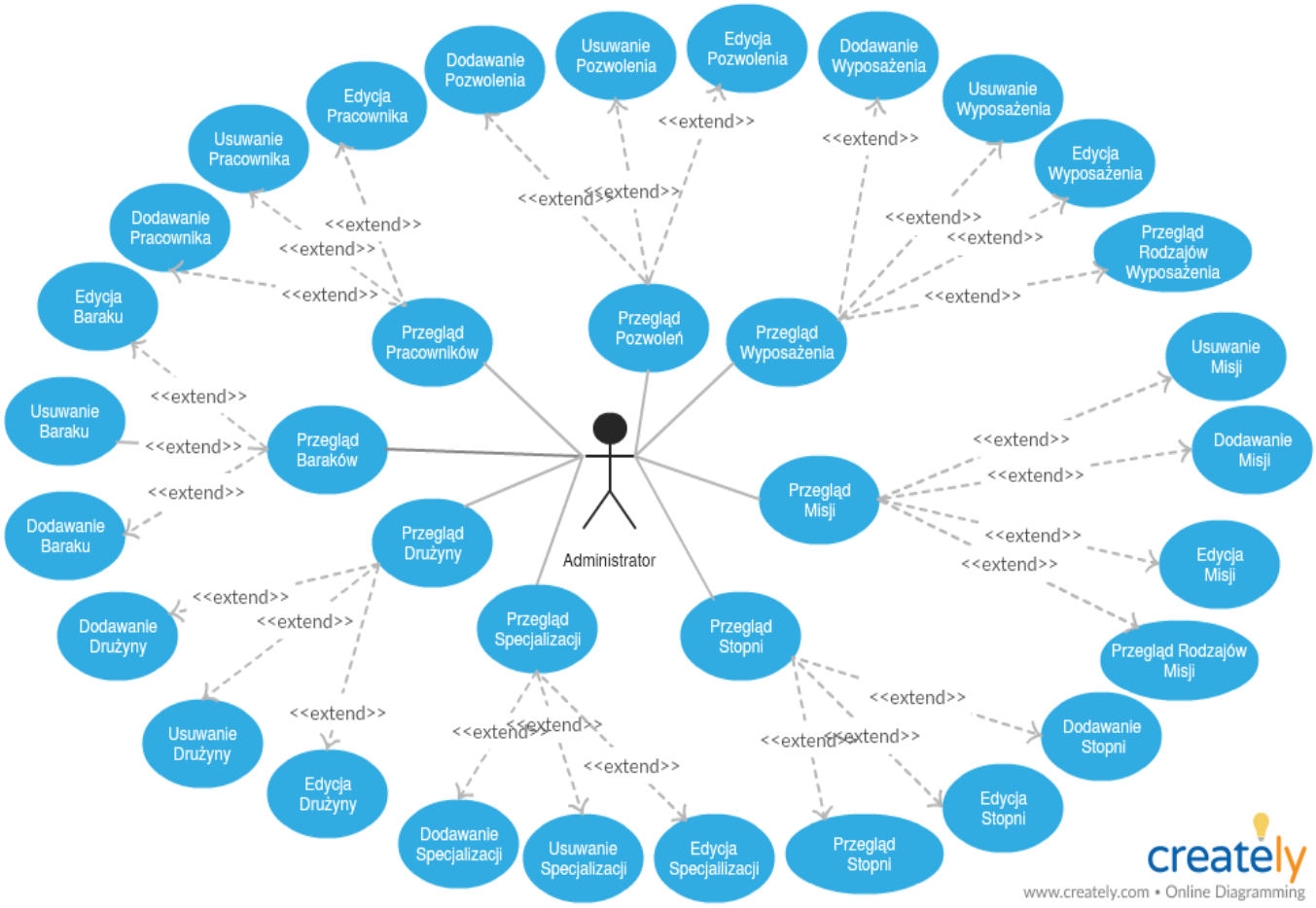
2.1 Wymagania funkcjonalne

- Wybór, które dane powinny być wyświetlane,
- Wyświetlanie danych w czytelny sposób,
- Dodawanie danych do konkretnych tabel,
- Edycja danych,
- Usuwanie danych,

2.2 Wymagania niefunkcjonalne

- Wydajność,
- Niezawodność,
- Łatwość użycia,

2.3 Diagram przypadków użycia



3. Wykorzystane technologie

- Microsoft SQL Server 2017

- Język C#

- WPF

- Entity Framework

- Caliburn.Micro

4. Projekt aplikacji

4.1 Architektura aplikacji

- Models – przechowuje modele bazy danych,

- Views – przechowuje widoki aplikacji,

- ViewModels – odpowiada za logikę systemu,

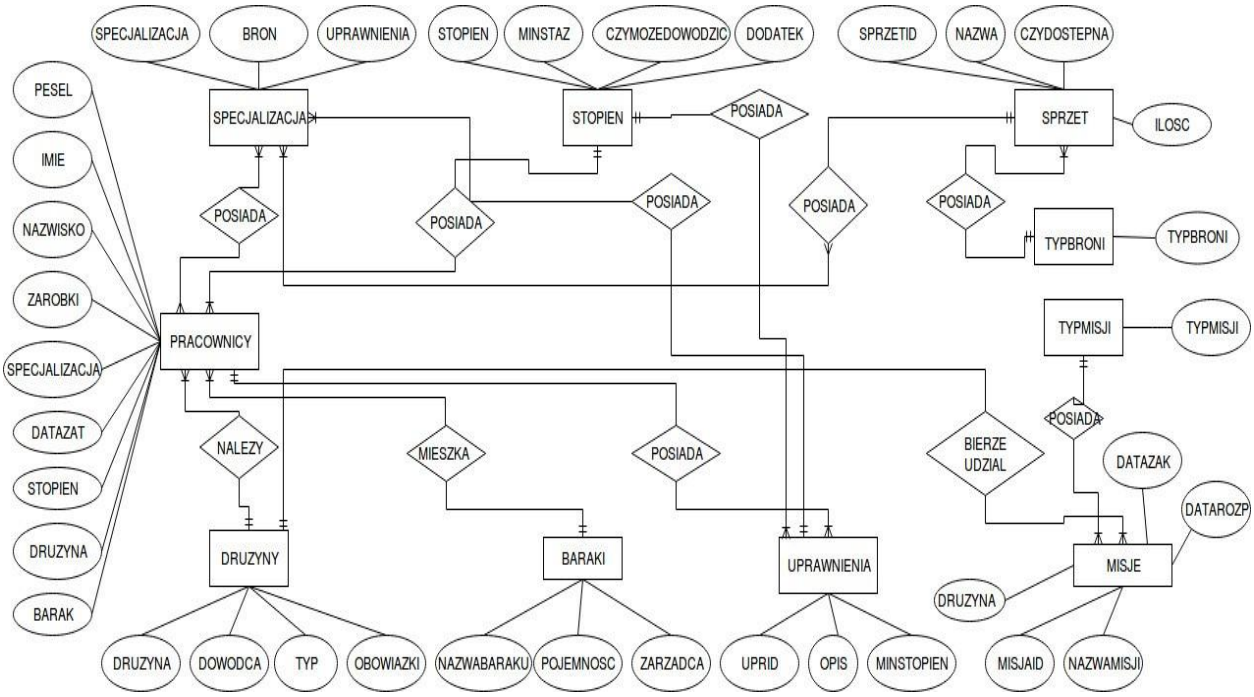
- Services – do obsługi bazy danych,

- DBInitializer – klasa odpowiadająca za wypełnienie nowej bazy przykładowymi danymi,

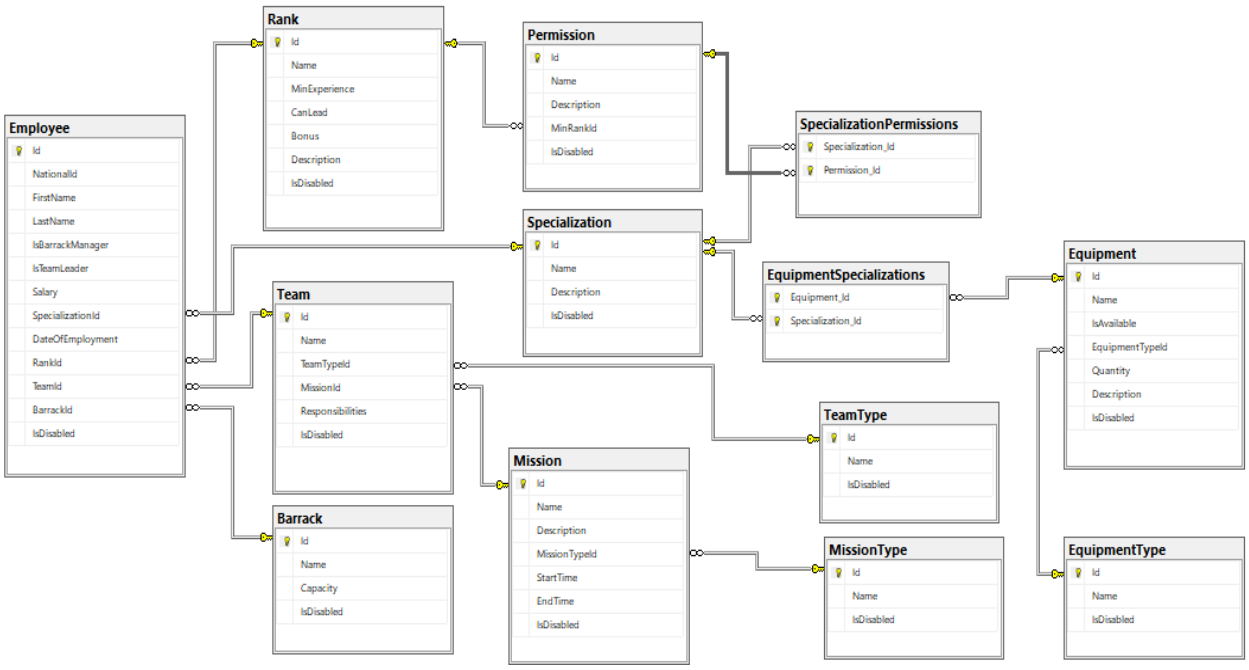
- DTO – Data Transfer Object – obiekty odpowiadające obiektom z tabel, na który pracujemy w BackEndzie, mogą zawierać pola, których nie ma w bazie danych, przez co umożliwiają nam łatwiejszą obsługę programu,

- Migrations – folder zawierający migracje bazy danych, co umożliwia nam zmianę struktury bazy danych lub wykonywania skryptów bez konieczności jej usuwania,

4.2 Projekt koncepcyjny bazy danych



4.3 Projekt schematu relacyjnego



4.4 Mapowanie klas na tabele bazodanowe

Code First poprzez Entity Framework.

ArmyBaseContext – klasa odpowiadająca za szkielet bazy danych.

```
namespace ArmyBase.Models
{
    using ArmyBase.Models.Initializer;
    using ArmyBase.ViewModels;
    using Caliburn.Micro;
    using System;
    using System.Data.Entity;
    using System.Linq;
    using System.Threading.Tasks;

    public class ArmyBaseContext : DbContext
    {
        // Your context has been configured to use a 'Model2' connection string from your application's
        // configuration file (App.config or Web.config). By default, this connection string targets the
        // 'ArmyBase.Models.Model2' database on your LocalDb instance.
        //
        // If you wish to target a different database and/or database provider, modify the 'Model2'
        // connection string in the application configuration file.
        public ArmyBaseContext()
            : base("name=ArmyBaseContext")
        {
        }

        // Add a DbSet for each entity type that you want to include in your Models. For more information
        // on configuring and using a Code First Models, see http://go.microsoft.com/fwlink/?LinkId=390109.

        // public virtual DbSet<MyEntity> MyEntities { get; set; }

        public DbSet<Barrack> Barracks { get; set; }
        public DbSet<Employee> Employees { get; set; }
        public DbSet<Equipment> Equipments { get; set; }
        public DbSet<EquipmentType> EquipmentTypes { get; set; }
        public DbSet<Mission> Missions { get; set; }
        public DbSet<MissionType> MissionTypes { get; set; }
        public DbSet<Permission> Permissions { get; set; }
        public DbSet<Rank> Ranks { get; set; }
        public DbSet<Specialization> Specializations { get; set; }
        public DbSet<Team> Teams { get; set; }
        public DbSet<TeamType> TeamTypes { get; set; }
    }

    //public class MyEntity
    //{
    //    public int Id { get; set; }
    //    public string Name { get; set; }
    //}
}
```

Mapowanie tabel na modele:

Wykorzystane adnotacje (DataAnnotations):

-`Table` – adnotacja poprzez którą nadajemy nazwę tabeli,

- `Key` – klucz główny tabeli,

- `Required` – adnotacja wymagająca podania wartości pola, możliwość utworzenia walidacji poprzez nadanie atrybutu `ErrorMessage`,

- `ForeignKey` – nadanie klucza obcego, który zostaje powiązany z elementem z innej tabeli, o tym samym typie.

Relacje:
- one-to-many – relacja realizowana poprzez nadanie klucza obcego obiektowi w jednej tabeli o typie X, gdzie następnie w tabeli tego obiektu (typ X) tworzymy kolekcję ICollection złożoną z obiektów typu Y,

- many-to-many – relacja tworzona poprzez utworzenie w obu modelach o różnych typach kolekcji ICollection z obiektami typów z drugiej tabeli.

```
namespace ArmyBase.Models
{
    using System;
    using System.Collections.Generic;
    using System.ComponentModel.DataAnnotations;
    using System.ComponentModel.DataAnnotations.Schema;
    using System.Data.Entity.Spatial;

    [Table("Barrack")]
    public class Barrack
    {
        [Key]
        public int Id { get; set; }

        [Required(ErrorMessage =("Barack's name is required"))]
        public string Name { get; set; }

        [Required]
        public int Capacity { get; set; }

        public ICollection<Employee> Employee { get; set; }

        public bool IsDisabled { get; set; } = false;
    }
}
```

```
[Table("Employee")]
public class Employee
{
    [Key]
    public int Id { get; set; }

    [Required(ErrorMessage = ("National ID is required"))]
    public int NationalId { get; set; }

    [Required(ErrorMessage = ("First name is required"))]
    public string FirstName { get; set; }

    [Required(ErrorMessage = ("Last name is required"))]
    public string LastName { get; set; }

    public bool IsBarrackManager { get; set; }

    public bool IsTeamLeader { get; set; }

    public double Salary { get; set; }

    [ForeignKey("Specialization")]
    public int? SpecializationId { get; set; }

    public Specialization Specialization { get; set; }

    [Required]
    public DateTime DateOfEmployment { get; set; }

    [ForeignKey("Rank")]
    public int? RankId { get; set; }

    public Rank Rank { get; set; }

    [ForeignKey("Team")]
    public int? TeamId { get; set; }

    public Team Team { get; set; }

    [ForeignKey("Barrack")]
    public int? BarrackId { get; set; }

    public Barrack Barrack { get; set; }

    public bool IsDisabled { get; set; } = false;
}
```

```
[Table("Equipment")]
public class Equipment
{
    [Key]
    public int Id { get; set; }

    [Required(ErrorMessage = ("Name is required"))]
    public string Name { get; set; }

    public bool IsAvailable { get; set; }

    [ForeignKey("EquipmentType")]
    [Required(ErrorMessage = ("Equipment type is required"))]
    public int? EquipmentTypeId { get; set; }

    public EquipmentType EquipmentType { get; set; }

    [Required(ErrorMessage = ("Quantity is required"))]
    public int Quantity { get; set; }

    public string Description { get; set; }

    [ForeignKey("Id")]
    public ICollection<Specialization> Specialization { get; set; }

    public bool IsDisabled { get; set; } = false;
}
```

```
[Table("EquipmentType")]
public class EquipmentType
{
    [Key]
    public int Id { get; set; }

    [Required(ErrorMessage = ("Name is required"))]
    public string Name { get; set; }

    public ICollection<Equipment> Equipment { get; set; }

    public bool IsDisabled { get; set; } = false;
}
```

```
[Table("Mission")]
public class Mission
{
    [Key]
    public int Id { get; set; }

    [Required(ErrorMessage = ("Name is required"))]
    public string Name { get; set; }

    public string Description { get; set; }

    [ForeignKey("MissionType")]
    [Required(ErrorMessage = ("Mission type is required"))]
    public int? MissionTypeId { get; set; }

    public MissionType MissionType { get; set; }

    [Required(ErrorMessage = ("Start time is required"))]
    public DateTime StartTime { get; set; }

    public DateTime? EndTime { get; set; }

    public ICollection<Team> Team { get; set; }

    public bool IsDisabled { get; set; } = false;
}
```

```
[Table("MissionType")]
public class MissionType
{
    [Key]
    public int Id { get; set; }

    [Required(ErrorMessage = ("Name is required"))]
    public string Name { get; set; }

    public ICollection<Mission> Mission { get; set; }

    public bool IsDisabled { get; set; } = false;
}
```

```
[Table("Permission")]
public class Permission
{
    [Key]
    public int Id { get; set; }

    [Required(ErrorMessage = ("Name is required"))]
    public string Name { get; set; }

    public string Description { get; set; }

    [ForeignKey("MinRank")]
    [Required(ErrorMessage = ("Minimum rank is required"))]
    public int? MinRankId { get; set; }

    public Rank MinRank { get; set; }

    public ICollection<Specialization> Specialization { get; set; }

    public bool IsDisabled { get; set; } = false;

}
```

```
[Key]
public int Id { get; set; }

[Required(ErrorMessage = ("Name is required"))]
public string Name { get; set; }

[Required(ErrorMessage = ("Minimal experience is required"))]
public int MinExperience { get; set; }

public bool CanLead { get; set; }

public int? Bonus { get; set; }

public string Description { get; set; }

public ICollection<Employee> Employee { get; set; }

public ICollection<Permission> Permission { get; set; }

public bool IsDisabled { get; set; } = false;
```

```
[Table("Specialization")]
public class Specialization
{
    [Key]
    public int Id { get; set; }

    [Required(ErrorMessage = ("Name is required"))]
    public string Name { get; set; }

    public string Description { get; set; }

    [ForeignKey("Specialization")]
    public ICollection<Equipment> Equipment { get; set; }

    [ForeignKey("Specialization")]
    public ICollection<Permission> Permission { get; set; }

    [ForeignKey("Specialization")]
    public ICollection<Employee> Employee { get; set; }

    public bool IsDisabled { get; set; } = false;
}
```

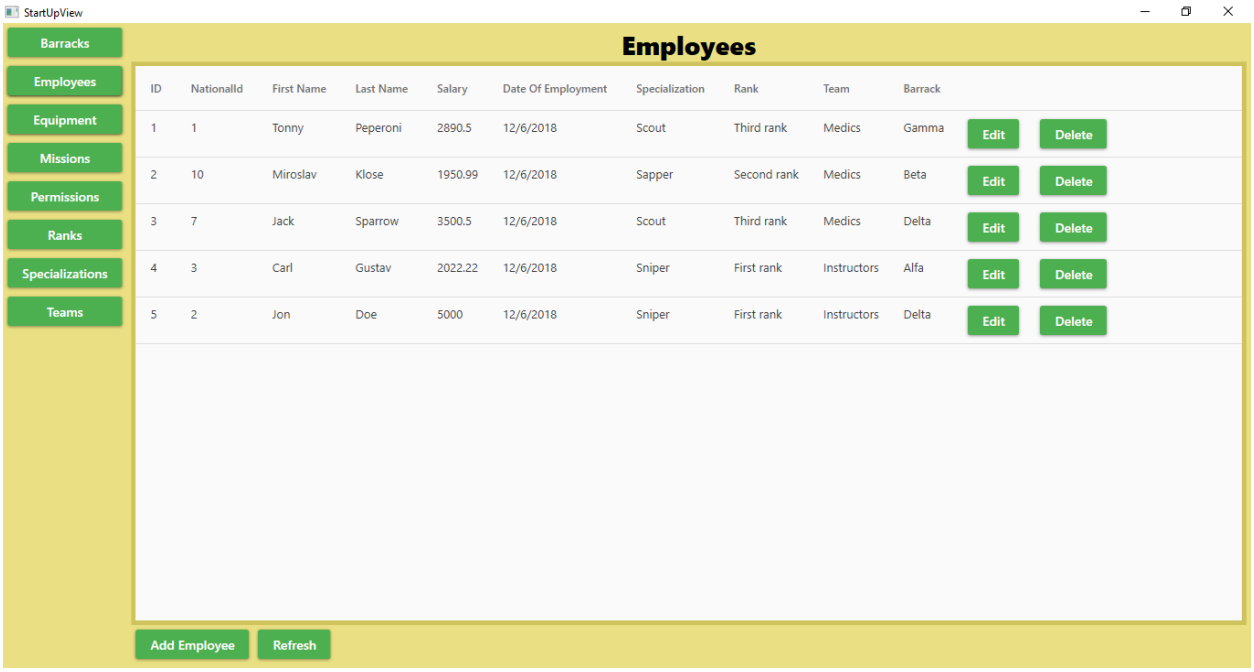
```
[Table("Team")]
public class Team
{
    [Key]
    public int Id { get; set; }

    [Required(ErrorMessage = ("Name is required"))]
    public string Name { get; set; }

    [ForeignKey("TeamType")]
    [Required(ErrorMessage = ("Team type is required"))]
    public int? TeamTypeId { get; set; }

    public TeamType TeamType { get; set; }

    [ForeignKey("Mission")]
    public int? MissionId { get; set; }

    public Mission Mission { get; set; }

    public string Responsibilities { get; set; }

    public ICollection<Employee> Employee { get; set; }

    public bool IsDisabled { get; set; } = false;

}
```

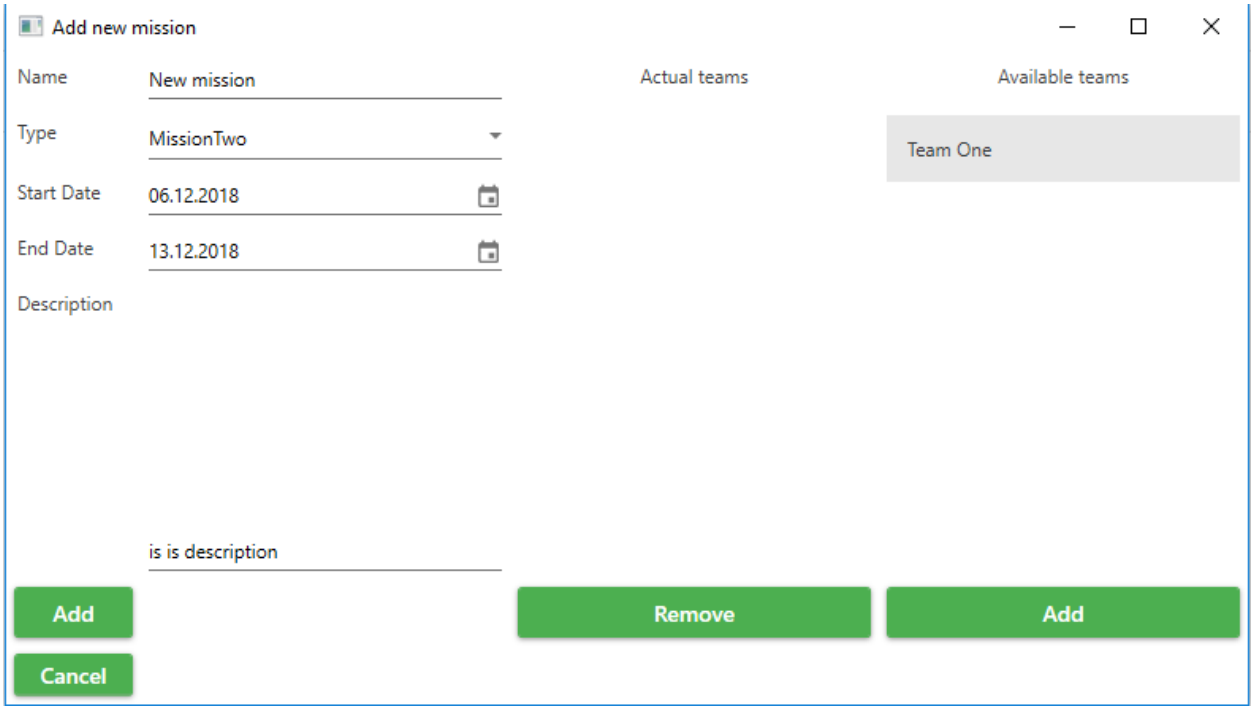
```
[Table("TeamType")]
public class TeamType
{
    [Key]
    public int Id { get; set; }

    [Required(ErrorMessage = ("Name is required"))]
    public string Name { get; set; }

    public ICollection<Team> Team { get; set; }

    public bool IsDisabled { get; set; } = false;
}
```

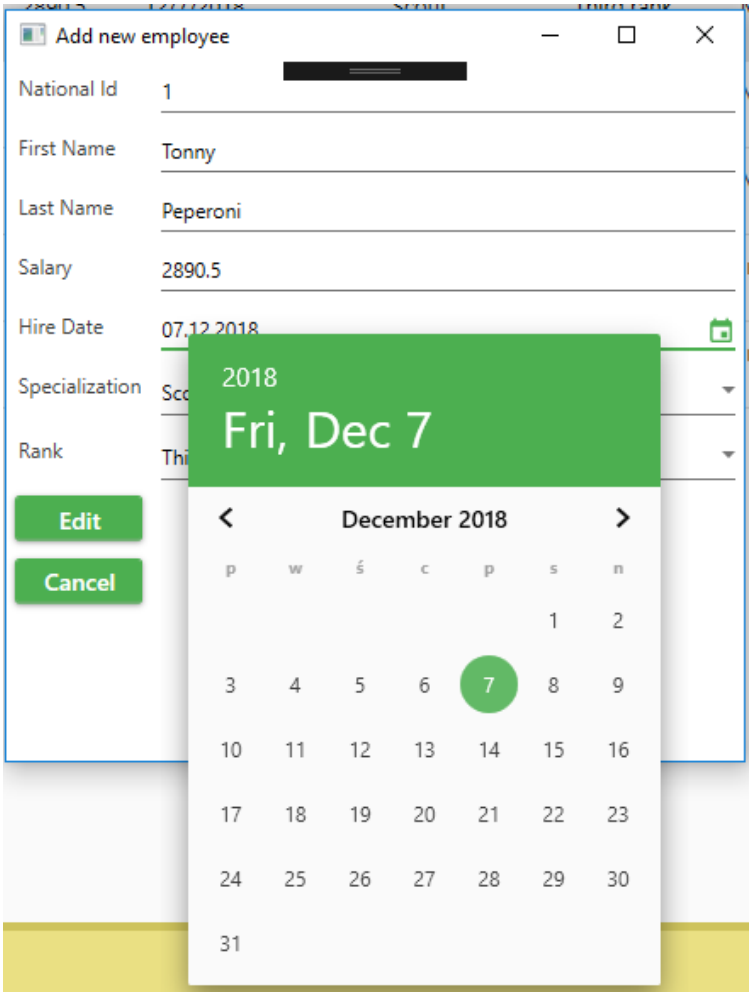
5. Funkcjonalność aplikacji

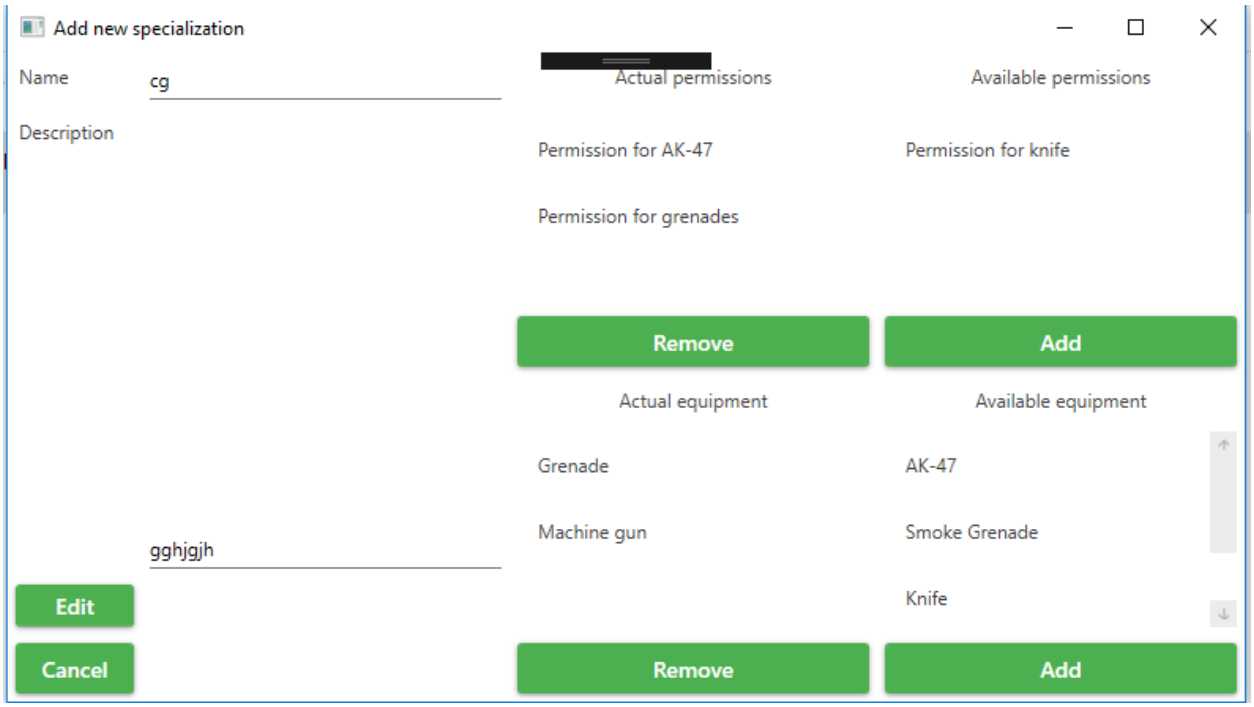
- Przegląd, dodawanie, usuwanie i edycja pracowników,
- Przegląd, dodawanie, usuwanie i edycja baraków(koszar) (zarządzanie listą zamieszkałych pracowników w nowych jak i istniejących barakach),
- Przegląd, dodawanie, usuwanie i edycja wyposażenia oraz jego typu,
- Przegląd, dodawanie, usuwanie i edycja misji oraz ich typów (zarządzanie listą zaangażowanych w misję drużyn w nowych jak i istniejących misjach),
- Przegląd, dodawanie, usuwanie i edycja pozwoleń,
- Przegląd, dodawanie, usuwanie i edycja stopni,
- Przegląd, dodawanie, usuwanie i edycja specjalizacji (zarządzanie listami zezwoleń oraz wyposażenia dla istniejących jak i nowych specjalizacji),
- Przegląd, dodawanie, usuwanie i edycja drużyn oraz ich typów (zarządzanie listą pracowników należących do drużyny),
- Wykrywanie czy istnieje baza danych,
- Generowanie nowej bazy danych wraz z przykładowymi danymi.

6. Interfejs użytkownika




Add new mission
Name
New mission
Actual teams
Available teams
Type
MissionTwo
Team One
Start Date
06.12.2018
End Date
13.12.2018
Description
is is description
Add
Remove
Add
Cancel


Add new employee
National Id
1
First Name
Tonny
Last Name
Peperoni
Salary
2890.5
Hire Date
07.12.2018
Specialization
Rank
Edit
Cancel
2018
Fri, Dec 7
December 2018
p w ś c p s n
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31



7. Podsumowanie

Aplikacja spełnia wszystkie wymagane funkcjonalności. Prosty interfejs sprawia, że obsługa aplikacji jest łatwa. Wszystkie zmiany wprowadzane przez użytkownika w aplikacji są uaktualniane w bazie danych.

Dodatek A: Skrypty tworzące obiekty baz danych

```
CREATE DATABASE [ArmyBase]
GO
USE [ArmyBase]
GO
/****** Object:  Table [dbo].[__MigrationHistory]    Script Date: 07.12.2018 00:31:00 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[__MigrationHistory](
	[MigrationId] [nvarchar](150) NOT NULL,
	[ContextKey] [nvarchar](300) NOT NULL,
	[Model] [varbinary](max) NOT NULL,
	[ProductVersion] [nvarchar](32) NOT NULL,
 CONSTRAINT [PK_dbo.__MigrationHistory] PRIMARY KEY CLUSTERED
(
	[MigrationId] ASC,
	[ContextKey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

```
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[Barrack]    Script Date: 07.12.2018 00:31:00 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Barrack](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[Name] [nvarchar](max) NOT NULL,
	[Capacity] [int] NOT NULL,
	[IsDisabled] [bit] NOT NULL,
 CONSTRAINT [PK_dbo.Barrack] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[Employee]    Script Date: 07.12.2018 00:31:00 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Employee](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[NationalId] [int] NOT NULL,
	[FirstName] [nvarchar](max) NOT NULL,
	[LastName] [nvarchar](max) NOT NULL,
	[IsBarrackManager] [bit] NOT NULL,
	[IsTeamLeader] [bit] NOT NULL,
	[Salary] [float] NOT NULL,
	[SpecializationId] [int] NULL,
	[DateOfEmployment] [datetime] NOT NULL,
	[RankId] [int] NULL,
	[TeamId] [int] NULL,
	[BarrackId] [int] NULL,
	[IsDisabled] [bit] NOT NULL,
 CONSTRAINT [PK_dbo.Employee] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[Equipment]    Script Date: 07.12.2018 00:31:00 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Equipment](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[Name] [nvarchar](max) NOT NULL,
```

```
	[IsAvailable] [bit] NOT NULL,
	[EquipmentTypeId] [int] NOT NULL,
	[Quantity] [int] NOT NULL,
	[Description] [nvarchar](max) NULL,
	[IsDisabled] [bit] NOT NULL,
 CONSTRAINT [PK_dbo.Equipment] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[EquipmentSpecializations]    Script Date: 07.12.2018 00:31:00 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[EquipmentSpecializations](
	[Equipment_Id] [int] NOT NULL,
	[Specialization_Id] [int] NOT NULL,
 CONSTRAINT [PK_dbo.EquipmentSpecializations] PRIMARY KEY CLUSTERED
(
	[Equipment_Id] ASC,
	[Specialization_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/****** Object:  Table [dbo].[EquipmentType]    Script Date: 07.12.2018 00:31:00 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[EquipmentType](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[Name] [nvarchar](max) NOT NULL,
	[IsDisabled] [bit] NOT NULL,
 CONSTRAINT [PK_dbo.EquipmentType] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[Mission]    Script Date: 07.12.2018 00:31:00 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Mission](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[Name] [nvarchar](max) NOT NULL,
	[Description] [nvarchar](max) NULL,
```

```
	[MissionTypeId] [int] NOT NULL,
	[StartTime] [datetime] NOT NULL,
	[EndTime] [datetime] NULL,
	[IsDisabled] [bit] NOT NULL,
 CONSTRAINT [PK_dbo.Mission] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[MissionType]    Script Date: 07.12.2018 00:31:00 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[MissionType](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[Name] [nvarchar](max) NOT NULL,
	[IsDisabled] [bit] NOT NULL,
 CONSTRAINT [PK_dbo.MissionType] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[Permission]    Script Date: 07.12.2018 00:31:00 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Permission](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[Name] [nvarchar](max) NOT NULL,
	[Description] [nvarchar](max) NULL,
	[MinRankId] [int] NOT NULL,
	[IsDisabled] [bit] NOT NULL,
 CONSTRAINT [PK_dbo.Permission] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[Rank]    Script Date: 07.12.2018 00:31:00 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Rank](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[Name] [nvarchar](max) NOT NULL,
```

```
	[MinExperience] [int] NOT NULL,
	[CanLead] [bit] NOT NULL,
	[Bonus] [int] NULL,
	[Description] [nvarchar](max) NULL,
	[IsDisabled] [bit] NOT NULL,
 CONSTRAINT [PK_dbo.Rank] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[Specialization]    Script Date: 07.12.2018 00:31:00 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Specialization](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[Name] [nvarchar](max) NOT NULL,
	[Description] [nvarchar](max) NULL,
	[IsDisabled] [bit] NOT NULL,
 CONSTRAINT [PK_dbo.Specialization] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[SpecializationPermissions]    Script Date: 07.12.2018 00:31:00 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[SpecializationPermissions](
	[Specialization_Id] [int] NOT NULL,
	[Permission_Id] [int] NOT NULL,
 CONSTRAINT [PK_dbo.SpecializationPermissions] PRIMARY KEY CLUSTERED
(
	[Specialization_Id] ASC,
	[Permission_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/****** Object:  Table [dbo].[Team]    Script Date: 07.12.2018 00:31:01 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Team](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[Name] [nvarchar](max) NOT NULL,
```

```
	[TeamTypeId] [int] NOT NULL,
	[MissionId] [int] NULL,
	[Responsibilities] [nvarchar](max) NULL,
	[IsDisabled] [bit] NOT NULL,
 CONSTRAINT [PK_dbo.Team] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[TeamType]    Script Date: 07.12.2018 00:31:01 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[TeamType](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[Name] [nvarchar](max) NOT NULL,
	[IsDisabled] [bit] NOT NULL,
 CONSTRAINT [PK_dbo.TeamType] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
ALTER TABLE [dbo].[Barrack] ADD  DEFAULT ((0)) FOR [IsDisabled]
GO
ALTER TABLE [dbo].[Employee] ADD  DEFAULT ((0)) FOR [IsDisabled]
GO
ALTER TABLE [dbo].[Equipment] ADD  DEFAULT ((0)) FOR [IsDisabled]
GO
ALTER TABLE [dbo].[EquipmentType] ADD  DEFAULT ((0)) FOR [IsDisabled]
GO
ALTER TABLE [dbo].[Mission] ADD  DEFAULT ((0)) FOR [IsDisabled]
GO
ALTER TABLE [dbo].[MissionType] ADD  DEFAULT ((0)) FOR [IsDisabled]
GO
ALTER TABLE [dbo].[Permission] ADD  DEFAULT ((0)) FOR [IsDisabled]
GO
ALTER TABLE [dbo].[Rank] ADD  DEFAULT ((0)) FOR [IsDisabled]
GO
ALTER TABLE [dbo].[Specialization] ADD  DEFAULT ((0)) FOR [IsDisabled]
GO
ALTER TABLE [dbo].[Team] ADD  DEFAULT ((0)) FOR [IsDisabled]
GO
ALTER TABLE [dbo].[TeamType] ADD  DEFAULT ((0)) FOR [IsDisabled]
GO
ALTER TABLE [dbo].[Employee]  WITH CHECK ADD  CONSTRAINT [FK_dbo.Employee_dbo.Barrack_BarrackId]
FOREIGN KEY([BarrackId])
REFERENCES [dbo].[Barrack] ([Id])
GO
```

```
ALTER TABLE [dbo].[Employee] CHECK CONSTRAINT [FK_dbo.Employee_dbo.Barrack_BarrackId]
GO
ALTER TABLE [dbo].[Employee]  WITH CHECK ADD  CONSTRAINT [FK_dbo.Employee_dbo.Rank_RankId] FOREIGN
KEY([RankId])
REFERENCES [dbo].[Rank] ([Id])
GO
ALTER TABLE [dbo].[Employee] CHECK CONSTRAINT [FK_dbo.Employee_dbo.Rank_RankId]
GO
ALTER TABLE [dbo].[Employee]  WITH CHECK ADD  CONSTRAINT
[FK_dbo.Employee_dbo.Specialization_SpecializationId] FOREIGN KEY([SpecializationId])
REFERENCES [dbo].[Specialization] ([Id])
GO
ALTER TABLE [dbo].[Employee] CHECK CONSTRAINT [FK_dbo.Employee_dbo.Specialization_SpecializationId]
GO
ALTER TABLE [dbo].[Employee]  WITH CHECK ADD  CONSTRAINT [FK_dbo.Employee_dbo.Team_TeamId] FOREIGN
KEY([TeamId])
REFERENCES [dbo].[Team] ([Id])
GO
ALTER TABLE [dbo].[Employee] CHECK CONSTRAINT [FK_dbo.Employee_dbo.Team_TeamId]
GO
ALTER TABLE [dbo].[Equipment]  WITH CHECK ADD  CONSTRAINT
[FK_dbo.Equipment_dbo.EquipmentType_EquipmentTypeId] FOREIGN KEY([EquipmentTypeId])
REFERENCES [dbo].[EquipmentType] ([Id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Equipment] CHECK CONSTRAINT [FK_dbo.Equipment_dbo.EquipmentType_EquipmentTypeId]
GO
ALTER TABLE [dbo].[EquipmentSpecializations]  WITH CHECK ADD  CONSTRAINT
[FK_dbo.EquipmentSpecializations_dbo.Equipment_Equipment_Id] FOREIGN KEY([Equipment_Id])
REFERENCES [dbo].[Equipment] ([Id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[EquipmentSpecializations] CHECK CONSTRAINT
[FK_dbo.EquipmentSpecializations_dbo.Equipment_Equipment_Id]
GO
ALTER TABLE [dbo].[EquipmentSpecializations]  WITH CHECK ADD  CONSTRAINT
[FK_dbo.EquipmentSpecializations_dbo.Specialization_Specialization_Id] FOREIGN KEY([Specialization_Id])
REFERENCES [dbo].[Specialization] ([Id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[EquipmentSpecializations] CHECK CONSTRAINT
[FK_dbo.EquipmentSpecializations_dbo.Specialization_Specialization_Id]
GO
ALTER TABLE [dbo].[Mission]  WITH CHECK ADD  CONSTRAINT [FK_dbo.Mission_dbo.MissionType_MissionTypeId]
FOREIGN KEY([MissionTypeId])
REFERENCES [dbo].[MissionType] ([Id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Mission] CHECK CONSTRAINT [FK_dbo.Mission_dbo.MissionType_MissionTypeId]
GO
ALTER TABLE [dbo].[Permission]  WITH CHECK ADD  CONSTRAINT [FK_dbo.Permission_dbo.Rank_MinRankId]
FOREIGN KEY([MinRankId])
REFERENCES [dbo].[Rank] ([Id])
```

```
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Permission] CHECK CONSTRAINT [FK_dbo.Permission_dbo.Rank_MinRankId]
GO
ALTER TABLE [dbo].[SpecializationPermissions]  WITH CHECK ADD  CONSTRAINT
[FK_dbo.SpecializationPermissions_dbo.Permission_Permission_Id] FOREIGN KEY([Permission_Id])
REFERENCES [dbo].[Permission] ([Id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[SpecializationPermissions] CHECK CONSTRAINT
[FK_dbo.SpecializationPermissions_dbo.Permission_Permission_Id]
GO
ALTER TABLE [dbo].[SpecializationPermissions]  WITH CHECK ADD  CONSTRAINT
[FK_dbo.SpecializationPermissions_dbo.Specialization_Specialization_Id] FOREIGN KEY([Specialization_Id])
REFERENCES [dbo].[Specialization] ([Id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[SpecializationPermissions] CHECK CONSTRAINT
[FK_dbo.SpecializationPermissions_dbo.Specialization_Specialization_Id]
GO
ALTER TABLE [dbo].[Team]  WITH CHECK ADD  CONSTRAINT [FK_dbo.Team_dbo.Mission_MissionId] FOREIGN
KEY([MissionId])
REFERENCES [dbo].[Mission] ([Id])
GO
ALTER TABLE [dbo].[Team] CHECK CONSTRAINT [FK_dbo.Team_dbo.Mission_MissionId]
GO
ALTER TABLE [dbo].[Team]  WITH CHECK ADD  CONSTRAINT [FK_dbo.Team_dbo.TeamType_TeamTypeId] FOREIGN
KEY([TeamTypeId])
REFERENCES [dbo].[TeamType] ([Id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Team] CHECK CONSTRAINT [FK_dbo.Team_dbo.TeamType_TeamTypeId]
GO
```