

DOKUMENTACJA ADMINISTRATION PANEL - BUSINESS CODING

Spis treści

Spis treści	2
1. INFORMACJE OGÓLNE	3
2. CHARAKTERYSTYKA PROGRAMU	4
2.1 WPROWADZENIE DO PROJEKTU	4
2.2 WPROWADZENIE DO PROGRAMU	4
2.3 WYKORZYSTANE TECHNOLOGIE	4
3. POCZĄTI PRACY	5
3.1 LOGOWANIE	5
3.2 WCZYTYWANIE DANYCH Z TABEL	6
4. DODAWANIE NOWYCH REKORDÓW DO TABELI	8
4.1 DODAWANIE DO USERS	8
4.2 DODAWANIE DO COMPANIES	9
4.3 DODAWANIE DO DESCRIPTION	10
4.4 DODAWANIE DO EMPLOYEES	10
4.5 DODAWANIE DO PROJECTS	11
4.6 DODAWANIE DO TYPESOFEMPLOYEES	12
4.7 DODAWANIE DO TYPESOFFICES	13
5. MODYFIKOWANIE REKORDÓW W TABELI	14
5.1 MODYFIKOWANIE W USERS	14
5.2 MODYFIKOWANIE W COMPANIES	14
5.3 MODYFIKOWANIE W DESCRIPTION	15
5.4 MODYFIKOWANIE W EMPLOYEES	15
5.5 MODYFIKOWANIE W PROJECTS	15
5.6 MODYFIKOWANIE W TYPESOFEMPLOYEES	15
5.7 MODYFIKOWANIE W TYPESOFFICES	15
6. USUWANIE REKORDÓW Z TABELI	16
7. KORZYSTANIE Z OKIENKA SQL QUERY	18
8. ARCHITEKTURA PROGRAMU	19
8.1 ADMINISTRATION PANEL	19
8.2 DATABASE ENTITY FRAMEWORK	21

1. INFORMACJE OGÓLNE

Nazwa projektu: *Business Coding*

Nazwa programu: Administration Panel

Wersja programu: 1.0

Autorzy: Piotr Gawęda, Tomasz Tomczykiewicz

Data opracowania: 4.11.2013

Krótki opis programu: Program służy do zarządzania bazą danych gry (dodawanie, modyfikowanie i usuwanie rekordów w tabelach).

2. CHARAKTERYSTYKA PROGRAMU

2.1 WPROWADZENIE DO PROJEKTU

Projekt *Business Coding* będzie turową grą internetową, w której gracz tworzy własne przedsiębiorstwo programistyczne. Na początku gry, gracz dostaje pewien zasób gotówki, który może przeznaczyć na zatrudnianie nowych pracowników oraz wynajęcie biura.

Użytkownik może podejmować nowe projekty, za które dostaje wynagrodzenie, ale musi zmieścić się w określonej liczbie tur. Za każdy ukończony projekt, gracz otrzyma respekt. Na podstawie respektu, będzie prowadzony ranking graczy.

2.2 WPROWADZENIE DO PROGRAMU

Program *Administration Panel* umożliwia administratorowi zarządzanie światem gry. Aby móc zarządzać bazą danych gry, trzeba podać właściwy login i hasło. Po zalogowaniu można dodawać, modyfikować i usuwać użytkowników, projekty, typy pracowników, typy biur, pracowników, firmy.

Ponadto można pisać zapytania, które są wykonywane bezpośrednio na bazie danych gry.

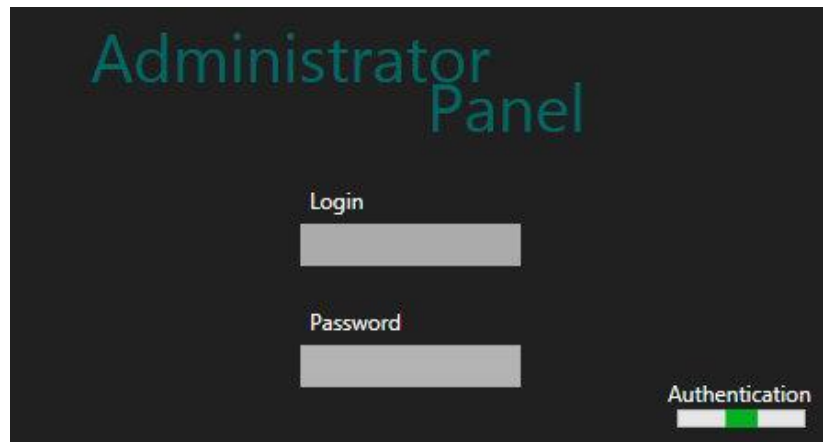
2.3 WYKORZYSTANE TECHNOLOGIE

- WPF (Windows Presentation Foundation) – API programu
- Entity Framework 6 – dostęp do bazy danych

3. POCZĄTI PRACY

3.1 LOGOWANIE

Po uruchomieniu programu, pokazuje się nam ekran logowania.

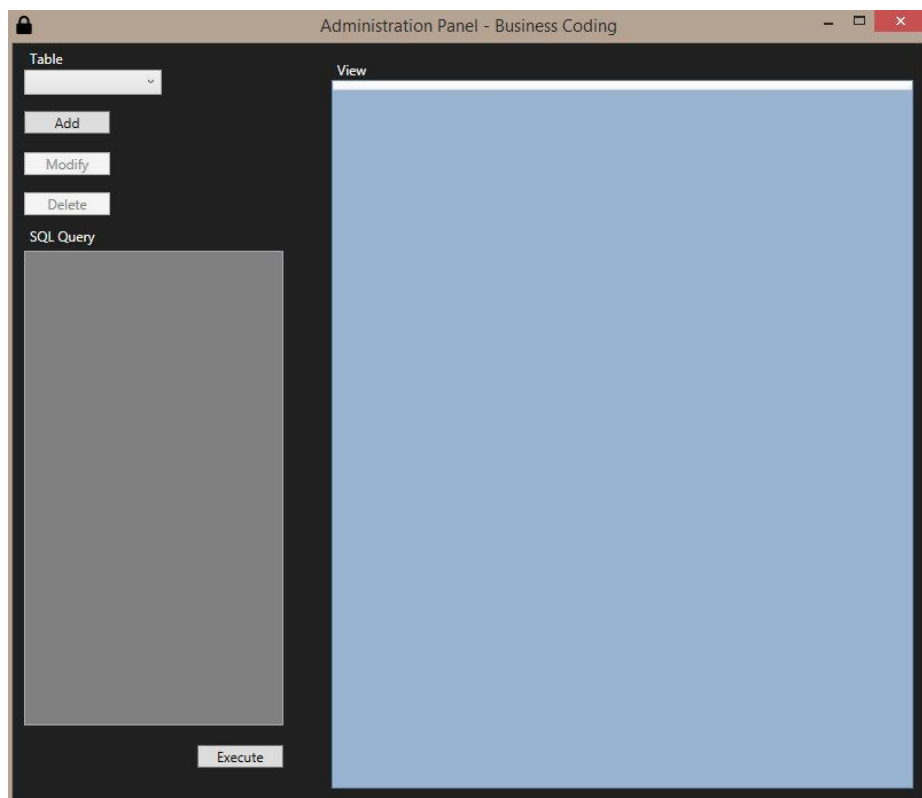


Rysunek 1 Ekran logowania

Aby się zalogować podajemy:

- **Login** – *admin*
- **Password** - *Admin123*

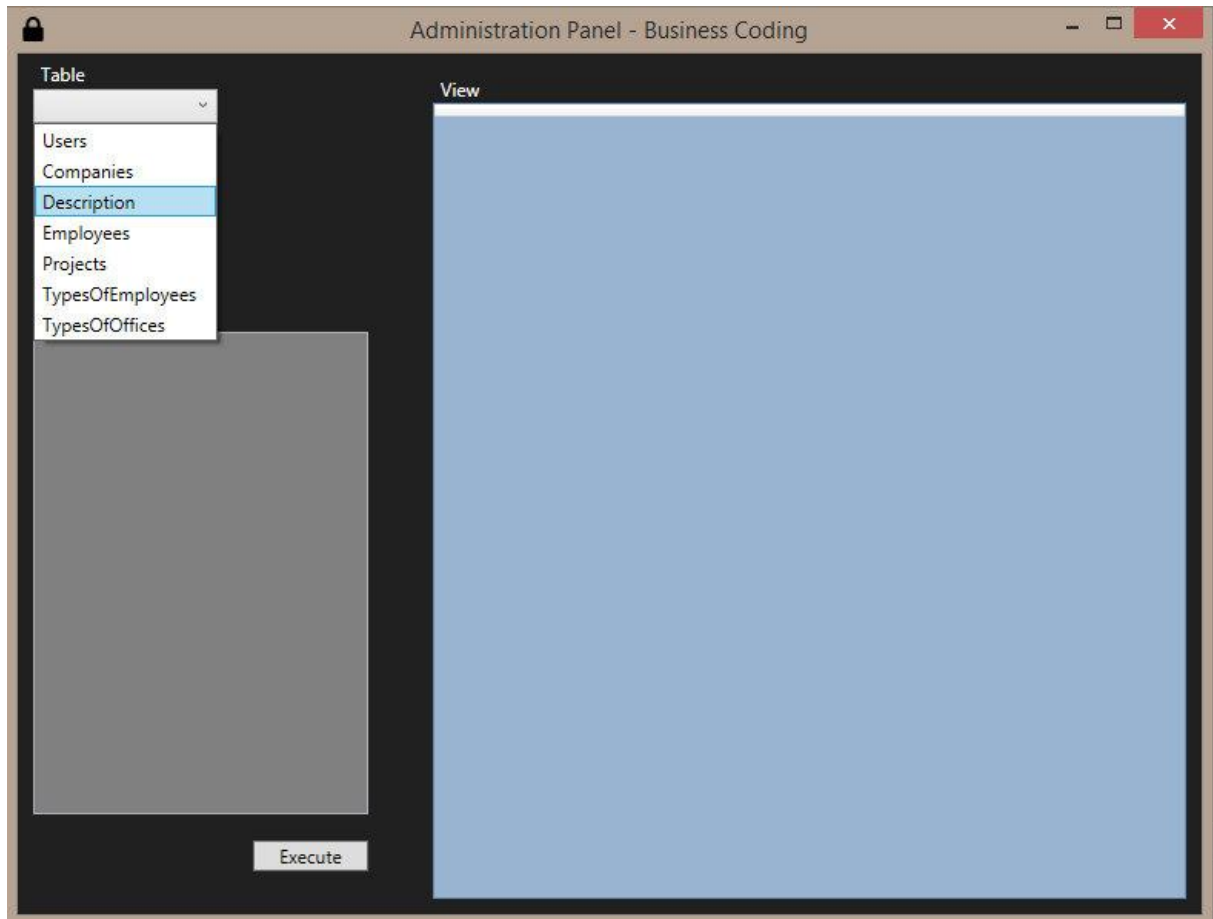
Po poprawnym podaniu danych, pojawia się nam kolejne okno z właściwym programem.



Rysunek 2 Ekran główny

3.2 WCZYTYWANIE DANYCH Z TABEL

Aby zobaczyć jakie dane są w tabeli, wybieramy z comboboxa (tuż pod napisem „Table”) tabelę, która nas interesuje.

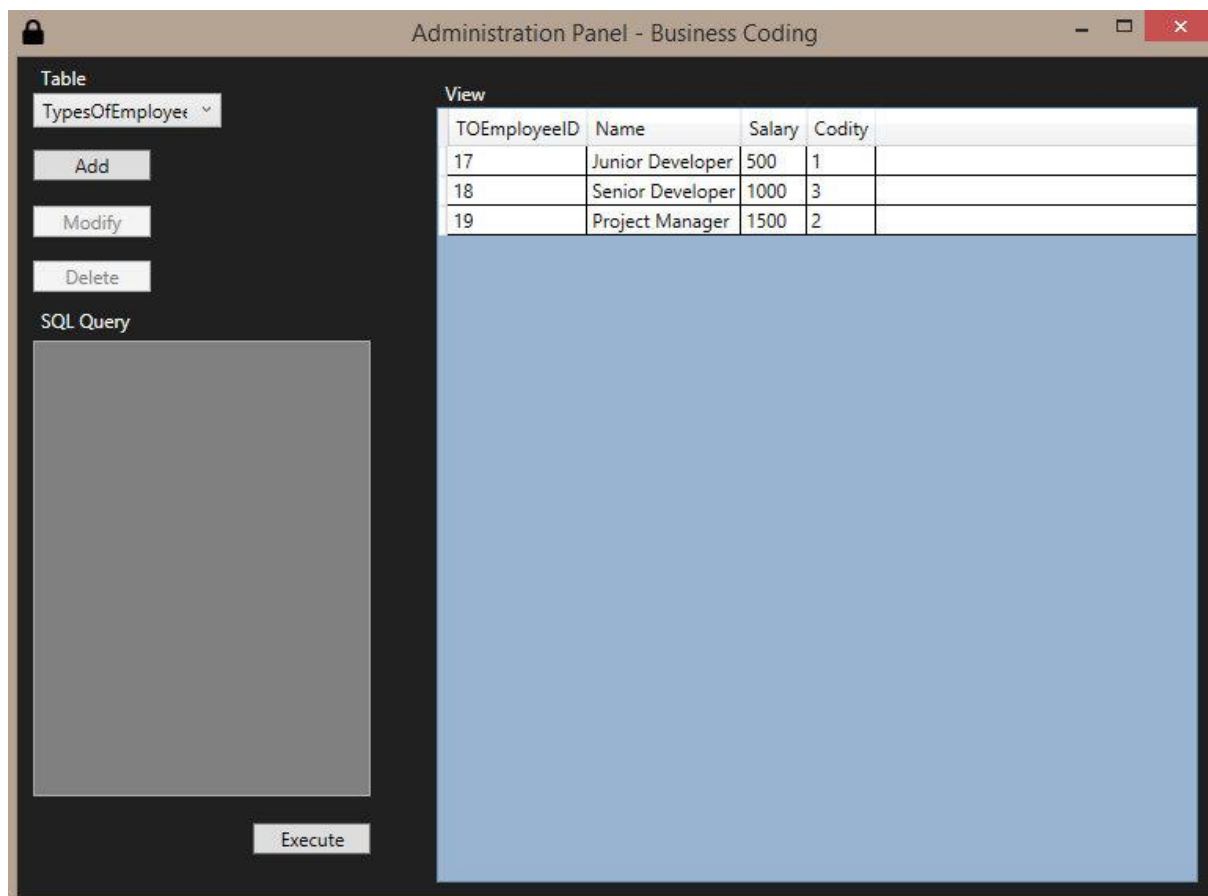


Rysunek 3 Wybór tabeli

Do wyboru mamy następujące tabele:

- **Users** – zawiera dane o użytkownikach zarejestrowanych w grze
- **Description** – zbiór opisów do projektów
- **Companies** – opisuje firmy użytkowników
- **Employees** – zbiór wszystkich pracowników w grze
- **Projects** – zbiór wszystkich projektów w grze
- **TypesOfEmployees** – zawiera dane o typach pracowników
- **TypesOfOffices** – zawiera dane o typach biur

Klikamy na interesującą nas tabelę i po chwili otrzymujemy podgląd.



Rysunek 4 Podgląd tabeli TypesOfEmployees

4. DODAWANIE NOWYCH REKORDÓW DO TABELI

Aby dodać rekord do tabeli, wybieram najpierw tabelę, do jakiej chcemy dodać rekord. Następnie wciskamy przycisk „Add” i otwiera nam się właściwy formularz.

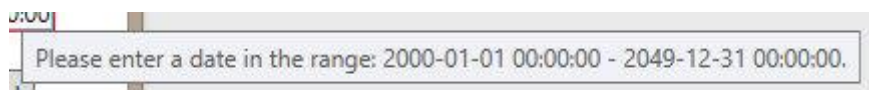
Aby zapisać wprowadzone zmiany, klikamy na przycisk „Confirm”, aby pominąć zmiany, wciskamy „Cancel”.

4.1 DODAWANIE DO USERS

W formularzu mamy następujące pola:

- **UserID** (*int*)– klucz główny w tabeli, nie jest edytowalny
- **Login** (*string*)– login gracza, nie może być pusty
- **Password** (*string*) – hasło gracza, nie może być puste
- **Email** (*string*)– email gracza, musi być to napis „typu” mail, nie może być pusty
- **Activation Date** (*DateTime*)– data założenia konta, musi być podane w formacie jak w formularzu (minimalna data 2000-01-01, maksymalna data 2049-12-31), nie może być puste
- **Last Login** (*DateTime*) – data ostatniego logowania, musi być podane w formacie jak w formularzu (minimalna data 2000-01-01, maksymalna data 2049-12-31), nie może być puste

W przypadku błędnego podania danych, wokół pola tekstowego pojawi się czerwona obwódka. Po najechaniu na nią pojawi się komunikat błędu.



Rysunek 5 Komunikat błędu

A screenshot of a Windows-style dialog box titled "Entry Form User". It contains several input fields: "UserID:" (disabled), "Login:", "Password:", "Email:", "Activation Date:", and "Last Login:". The "Activation Date" and "Last Login" fields contain the text "1/1/0001 12:00:00" and are highlighted with red rectangular borders. At the bottom of the dialog are two buttons: "Confirm" and "Cancel".

Rysunek 6 Formularz User


4.2 DODAWANIE DO COMPANIES

W formularzu mamy następujące pola:

- **CompanyID** (*int*) – klucz główny w tabeli, nie jest edytowalny
- **Name** (*string*) – nazwa firmy, nie może być pusta
- **Respect** (*int*) - respekt firmy, zakres 0-214783647
- **Money** (*double*) – stan konta firmy, zakres 0-100000000000
- **Cost Of Living** (*double*) – koszt utrzymania firmy, zakres 0-100000000000
- **TOOfficeID** (*int*) – klucz obcy, odwołanie do typu biura, nie może być puste
- **UserID** (*int*) – klucz obcy, odwołanie do konta gracza, nie może być puste

Pola **TOOfficeID** i **UserID** muszą mieć referencję do istniejących pól, inaczej podczas dodawania pojawi nam się ostrzeżenie i nie dodamy rekordu do tabeli.

W przypadku błędnego podania danych, wokół pola tekstowego pojawi się czerwona obwódka. Po najechaniu na nią pojawi się komunikat błędu.



The image shows a Windows-style dialog box titled "Entry Form Company". It contains several input fields with labels: "CompanyID:", "Name:", "Respect", "Money:", "Cost Of Living:", "TOOfficeID:", and "UserID:". The "Name:", "TOOfficeID:", and "UserID:" fields have red rectangular borders around them, indicating they contain invalid data. The "Respect", "Money:", and "Cost Of Living:" fields contain the number "0". At the bottom of the dialog are two buttons: "Confirm" and "Cancel".

Rysunek 7 Formularz Company

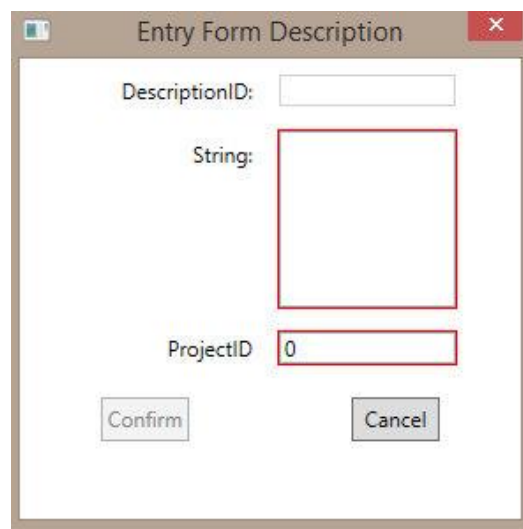
4.3 DODAWANIE DO DESCRIPTION

W formularzu mamy następujące pola:

- **DescriptionID** (*int*) – klucz główny w tabeli, nie jest edytowalny
- **Name** (*string*) – opis, nie może być puste
- **ProjectID** (*int*) – klucz obcy, odwołanie do projektu, nie może być puste

Pole **ProjectID** musi mieć referencję do istniejących pól, inaczej podczas dodawania pojawi nam się ostrzeżenie i nie dodamy rekordu do tabeli.

W przypadku błędnego podania danych, wokół pola tekstowego pojawi się czerwona obwódka. Po najechnaniu na nią pojawi się komunikat błędu.



Rysunek 8 Formularz Description

4.4 DODAWANIE DO EMPLOYEES

W formularzu mamy następujące pola:

- **EmployeeID** (*int*) – klucz główny w tabeli, nie jest edytowalny
- **Date Of Hiring** (*DateTime*) – data zatrudnienia, może być puste
- **CompanyID** (*int*) – klucz obcy, odwołanie do firmy, może być puste
- **TOEmployeeID** (*int*) – klucz obcy, odwołanie do typu pracownika, nie może być puste

Pola **TOEmployeeID** i **CompanyID** muszą mieć referencję do istniejących pól, inaczej podczas dodawania pojawi nam się ostrzeżenie i nie dodamy rekordu do tabeli.

W przypadku błędnego podania danych, wokół pola tekstowego pojawi się czerwona obwódka. Po najechnaniu na nią pojawi się komunikat błędu.

The image shows a Windows-style dialog box titled "Entry Form Employee". It has a standard title bar with a close button (X). The dialog contains four text input fields arranged vertically. The first three are empty, and the fourth, labeled "TOEmployeeID:", contains the number "0". Below the fields are two buttons: "Confirm" and "Cancel".

Rysunek 9 Formularz Employee

4.5 DODAWANIE DO PROJECTS

W formularzu mamy następujące pola:

- **ProjectID** (*int*) – klucz główny w tabeli, nie jest edytowalny
- **Name** (*string*) – nazwa projektu, nie może być pusta
- **Complexity** (*short*) – złożoność projektu, określa trudność projektu, zakres 1-100, nie może być puste
- **Time Rounds** (*short*) – długość w turach wykonywania projektu, zakres 1-100, nie może być puste
- **Time To End** (*short*) – ile zostało do końca projektu w razie jego podjęcia, zakres 0-100, nie może być puste
- **Gratification** (*double*) – wynagrodzenie, zakres 1-1000000, nie może być puste
- **Min Respect** (*int*) – minimalny respekt jaki firma musi posiadać, aby podjąć projekt, zakres 1 - 1000000, nie może być puste
- **Min Codity** (*short*) - minimalny respekt jaki firma musi posiadać, aby podjąć projekt, zakres 1-1000000, nie może być puste
- **CompanyID** (*int*) – klucz obcy, odwołanie do firmy, która podjęła projekt, może być puste

Pole **CompanyID** musi mieć referencję do istniejących pól, inaczej podczas dodawania pojawi nam się ostrzeżenie i nie dodamy rekordu do tabeli.

W przypadku błędnego podania danych, wokół pola tekstowego pojawi się czerwona obwódka. Po najechnaniu na nią pojawi się komunikat błędu.

The image shows a Windows-style dialog box titled "Entry Form Project". It contains several text input fields arranged vertically. The fields are labeled as follows: "ProjectID:", "Name:", "Complexity:", "Time Rounds:", "Time To End:", "Gratification:", "Min Respect:", "Min Cidity:", and "CompanyID:". The "Complexity:", "Time Rounds:", "Gratification:", "Min Respect:", and "Min Cidity:" fields have the number "0" entered. The "Name:" field has a red border, indicating a validation error. At the bottom of the dialog, there are two buttons: "Confirm" and "Cancel".

Rysunek 10 Formularz Project

4.6 DODAWANIE DO TYPESOFEMPLOYEES

W formularzu mamy następujące pola:

- **TOEmployeeID** (*int*) – klucz główny w tabeli, nie jest edytowalny
- **Name** (*string*) – nazwa typu pracownika, nie może być puste
- **Salary** (*double*) – pensja pracownika, zakres 0-5000, nie może być puste
- **Codity** (*short*) – kodowalność pracownika, zakres 0-100, nie może być puste

W przypadku błędnego podania danych, wokół pola tekstowego pojawi się czerwona obwódka. Po najechaniu na nią pojawi się komunikat błędu.

Entry Form Type Of Employee

TOEmployeeID:

Name:

Salary:

Codity:

Confirm Cancel

Rysunek 11 Formularz Type Of Employee

4.7 DODAWANIE DO TYPESOFFICES

W formularzu mamy następujące pola:

- **TOEmployeeID** (*int*) – klucz główny w tabeli, nie jest edytowalny
- **Name** (*string*) – nazwa typu biura, nie może być puste
- **Capacity** (*short*) – pojemność biura, zakres 1-100, nie może być puste
- **Cost Of Hiring** (*double*) – koszt wynajęcia biura, zakres 0-100000, nie może być puste

W przypadku błędnego podania danych, wokół pola tekstowego pojawi się czerwona obwódka. Po najechniu na nią pojawi się komunikat błędu.

Entry Form TypeOfOffice

TOOfficelD:

Name:

Capacity:

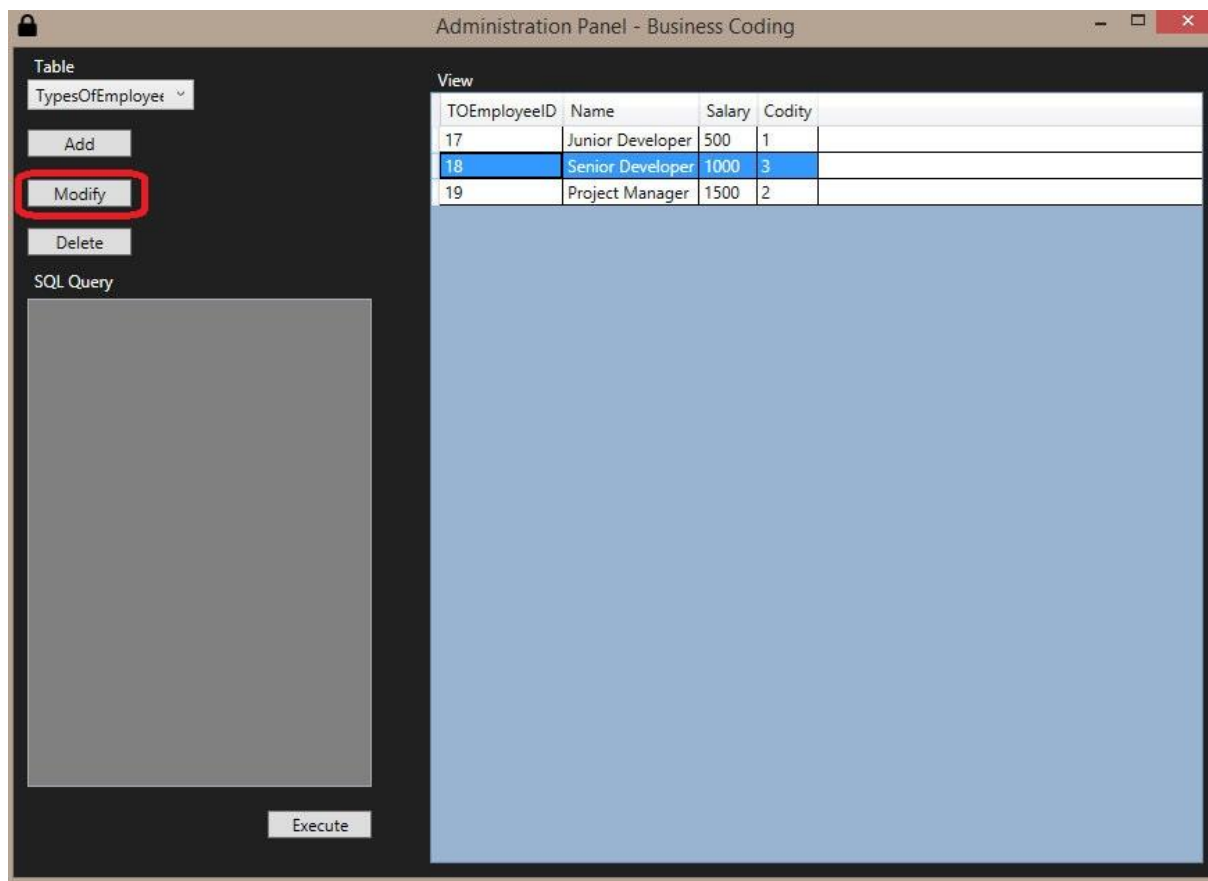
Cost Of Hiring:

Confirm Cancel

Rysunek 12 Formularz Type Of Office

5. MODYFIKOWANIE REKORDÓW W TABELI

Aby zmodyfikować rekord z tabeli, najpierw musi ją wczytać. Następnie w obszarze „View”, zaznaczamy rekord, który chcemy zmodyfikować i wciskamy przycisk „Modify”.



Rysunek 13 Przycisk Modify

Aby zapisać wprowadzone zmiany, klikamy na przycisk „Confirm”, aby pominąć zmiany, wciskamy „Cancel”.

5.1 MODYFIKOWANIE W USERS

Modyfikowanie rekordów odbywa się na tej samej zasadzie, co dodawanie. Jedynie większość pól już jest uzupełniona. Pola są tego samego typu i mają taką samą wymagalność.

5.2 MODYFIKOWANIE W COMPANIES

Modyfikowanie rekordów odbywa się na tej samej zasadzie, co dodawanie. Jedynie większość pól już jest uzupełniona. Pola są tego samego typu i mają taką samą wymagalność.

5.3 MODYFIKOWANIE W DESCRIPTION

Modyfikowanie rekordów odbywa się na tej samej zasadzie, co dodawanie. Jedynie większość pól już jest uzupełniona. Pola są tego samego typu i mają taką samą wymagalność.

5.4 MODYFIKOWANIE W EMPLOYEES

Modyfikowanie rekordów odbywa się na tej samej zasadzie, co dodawanie. Jedynie większość pól już jest uzupełniona. Pola są tego samego typu i mają taką samą wymagalność.

5.5 MODYFIKOWANIE W PROJECTS

Modyfikowanie rekordów odbywa się na tej samej zasadzie, co dodawanie. Jedynie większość pól już jest uzupełniona. Pola są tego samego typu i mają taką samą wymagalność.

5.6 MODYFIKOWANIE W TYPESOFEMPLOYEES

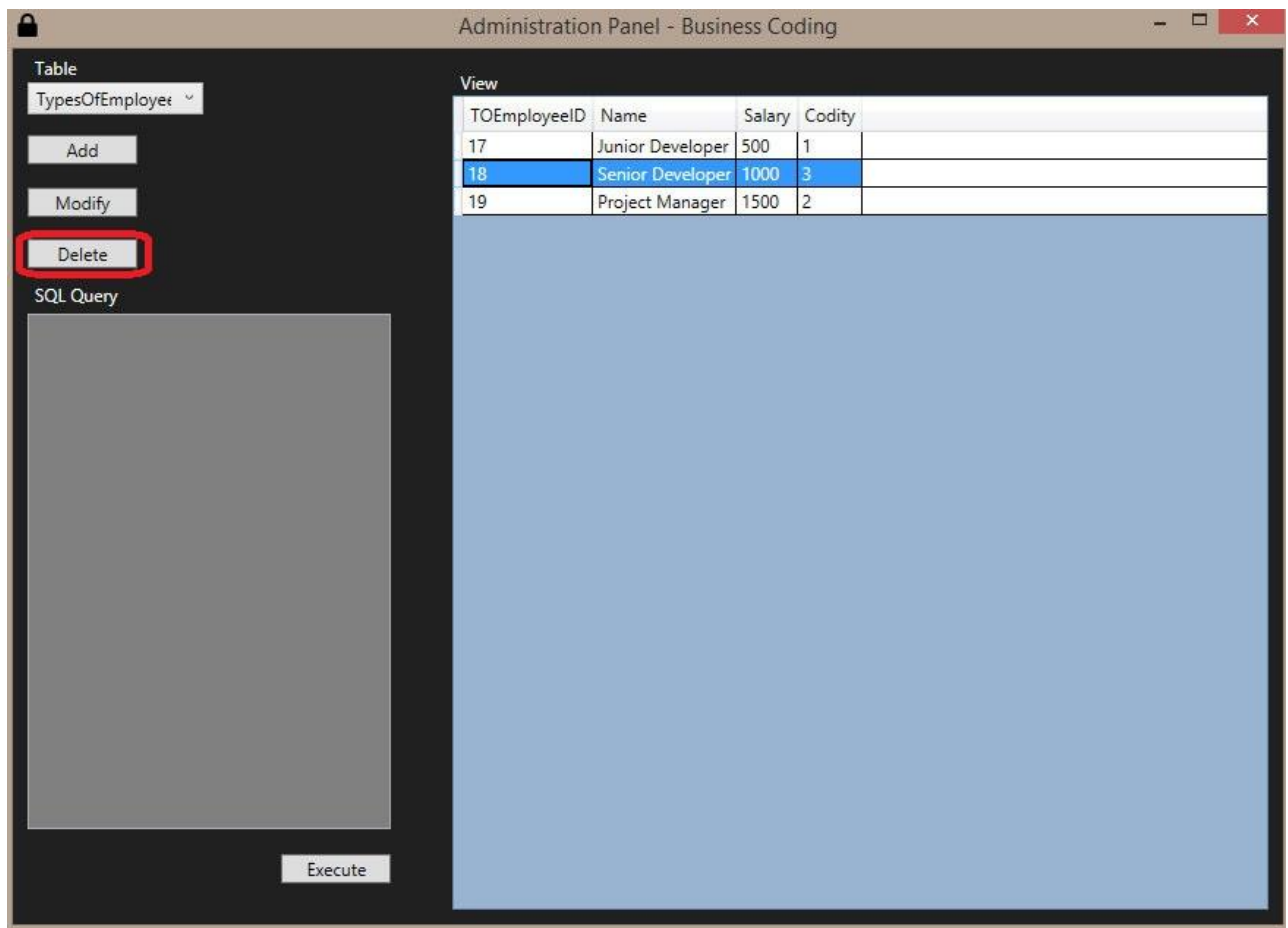
Modyfikowanie rekordów odbywa się na tej samej zasadzie, co dodawanie. Jedynie większość pól już jest uzupełniona. Pola są tego samego typu i mają taką samą wymagalność.

5.7 MODYFIKOWANIE W TYPESOFOFFICES

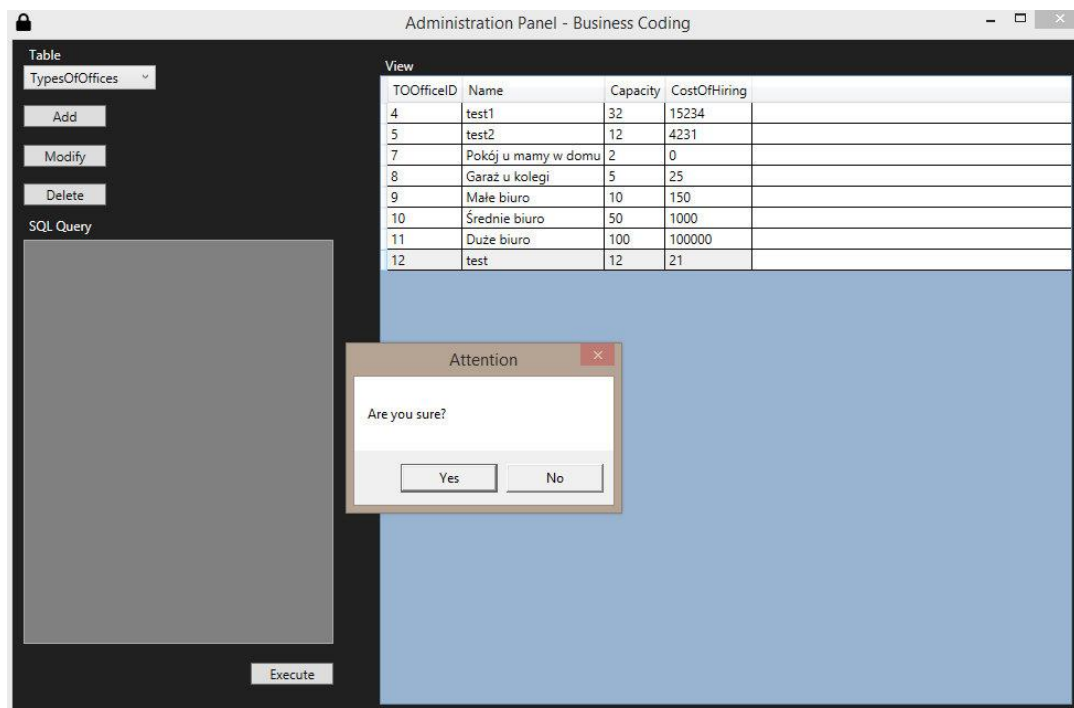
Modyfikowanie rekordów odbywa się na tej samej zasadzie, co dodawanie. Jedynie większość pól już jest uzupełniona. Pola są tego samego typu i mają taką samą wymagalność.

6. USUWANIE REKORDÓW Z TABELI

Aby usunąć rekord z tabeli, najpierw musi ją wczytać. Następnie w obszarze „View”, zaznaczamy rekord, który chcemy usunąć i wciskamy przycisk „Delete”. Potwierdzamy przyciskiem „OK”.

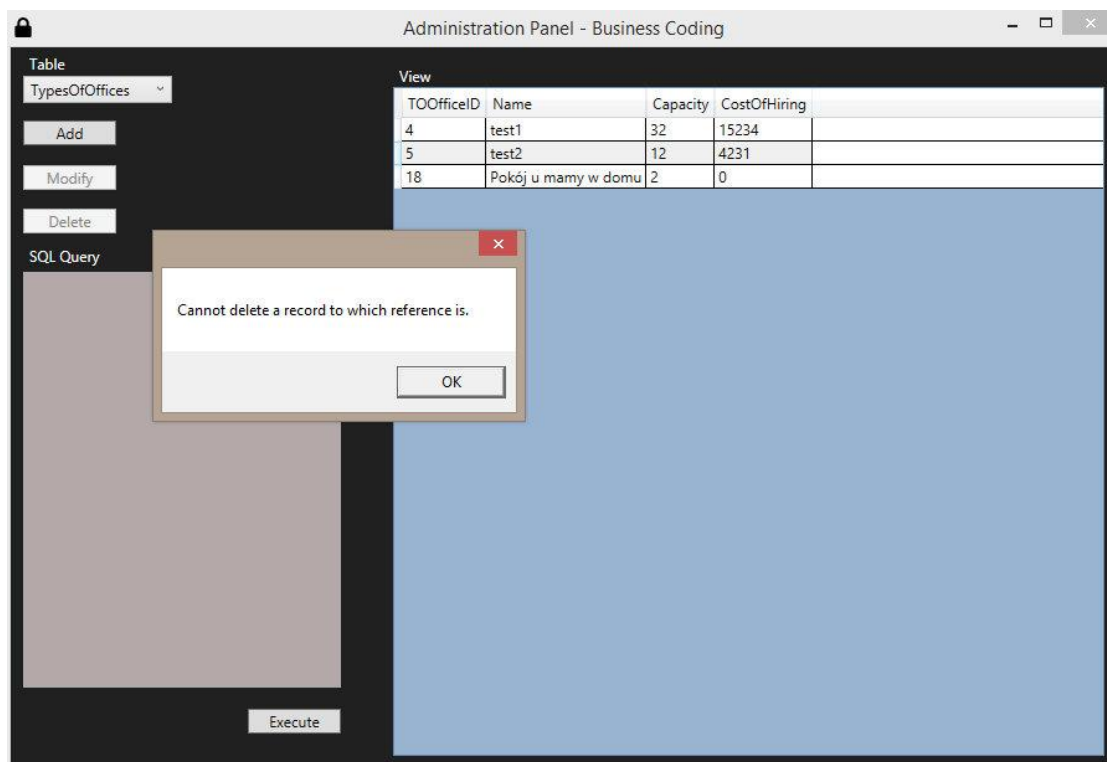


Rysunek 14 Przycisk Delete



Rysunek 15 Okno potwierdzenia usuwania

Jeśli do usuwanego rekordu jest odwołanie z innego rekordu, to program nie pozwoli na usunięcie i pojawi się stosowna informacja.

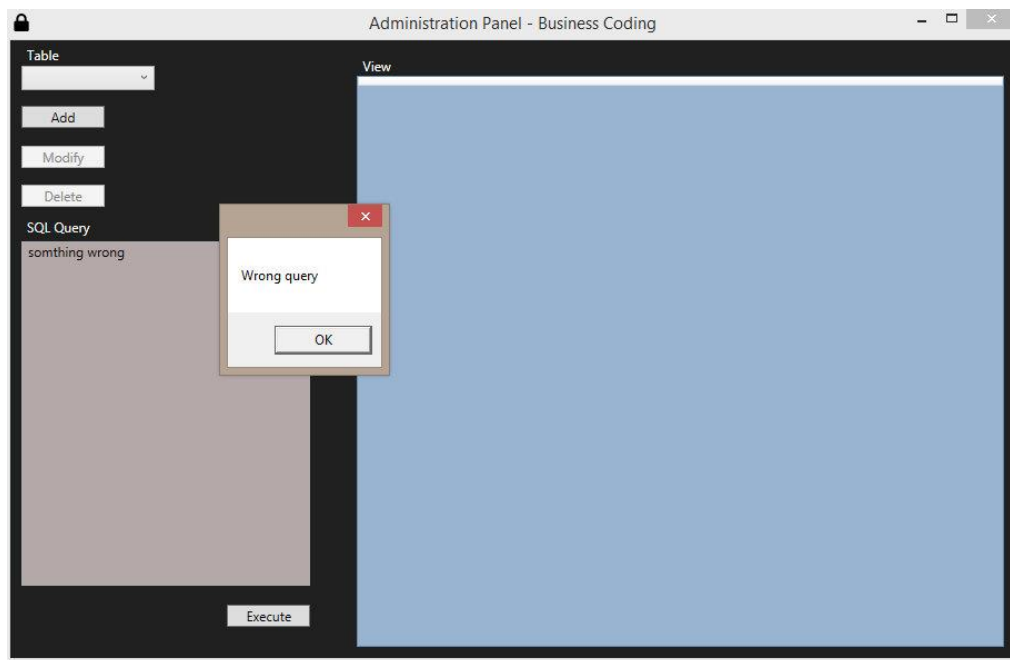


Rysunek 16 Próba usunięcia obiektu z referec

7. KORZYSTANIE Z OKIENKA SQL QUERY

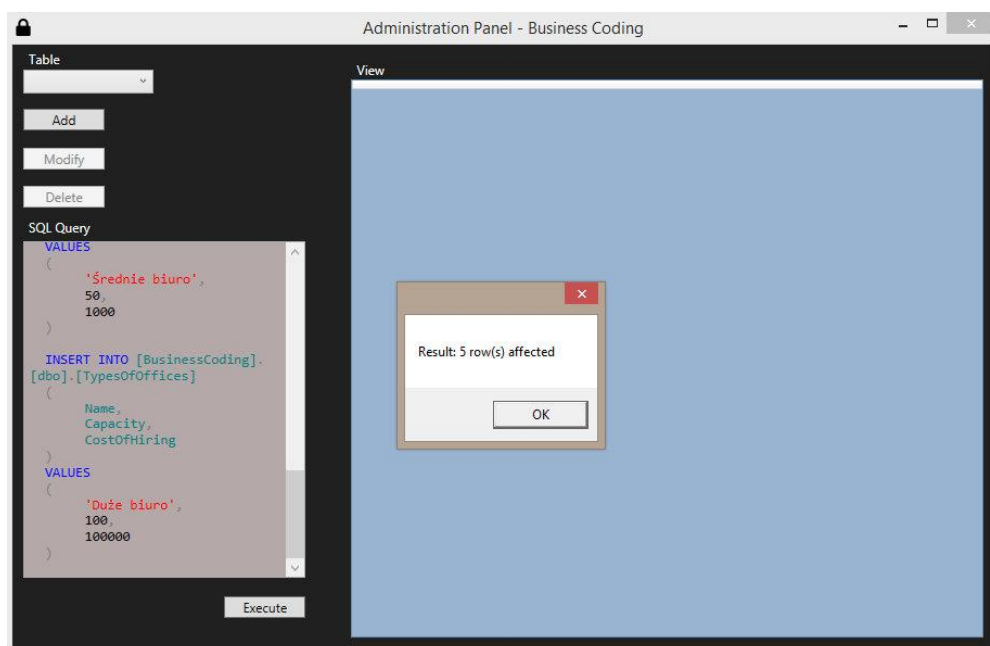
Program umożliwia również pisanie zapytań w języku SQL bezpośrednio do bazy danych. Aby to uczynić, klikamy na pole pod napisem „SQL Query” i piszemy zapytanie. Gdy już napiszemy zapytanie wciskamy przycisk „Execute”.

W przypadku źle napisanego zapytania, program zwróci nam okno z takową informacją.



Rysunek 17 Błędne zapytanie

W przeciwnym wypadku, otrzymamy informację o wyniku zapytania.



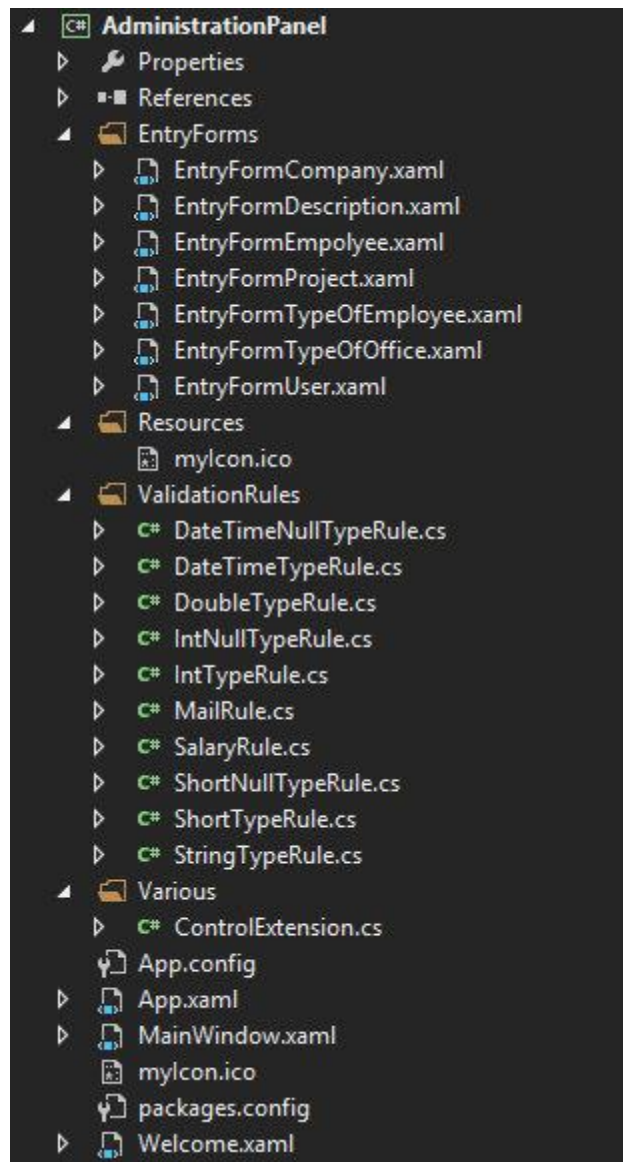
Rysunek 18 Wynik zapytania

8. ARCHITEKTURA PROGRAMU

Program składa się z dwóch projektów:

- AdministrationPanel – API programu oraz cała jego obsługa
- DatabaseEntityFramework – model bazy danych oraz dostęp do bazy danych

8.1 ADMINISTRATION PANEL



Rysunek 19 Drzewo AdministrationPanel

- **EntryForm** (folder) – zawiera wszystkie formularze, które są używane podczas dodawania jak i modyfikowania rekordów
- **Resources** (folder) – zawiera plik z ikonką
- **ValidationRules** (folder) – zawiera metody walidacji dla różnych typów pól
 - **DateTimeNullRule** – walidacja dla zmiennych typu DateTime, które mogą być również nullami
 - **DateTimeRule** – walidacja dla zmiennych typu DateTime, które nie mogą być nullami

- **Various** (*folder*) – zawiera inne dodatkowe klasy
- **MainWindow** (*plik*) – kod głównego okna programu
- **Welcome.xaml** (*plik*) + kod okna logowania

```

namespace AdministrationPanel
{
    /// <summary> ...
    public partial class MainWindow : Window
    {
        Fields - selectedItem

        Fields

        Constructors

        Controls

        Buttons

        Creating Objects From Entry Form

        Display Records

        Add Records

        Modify Records

        Delete Records

        Additional Methods

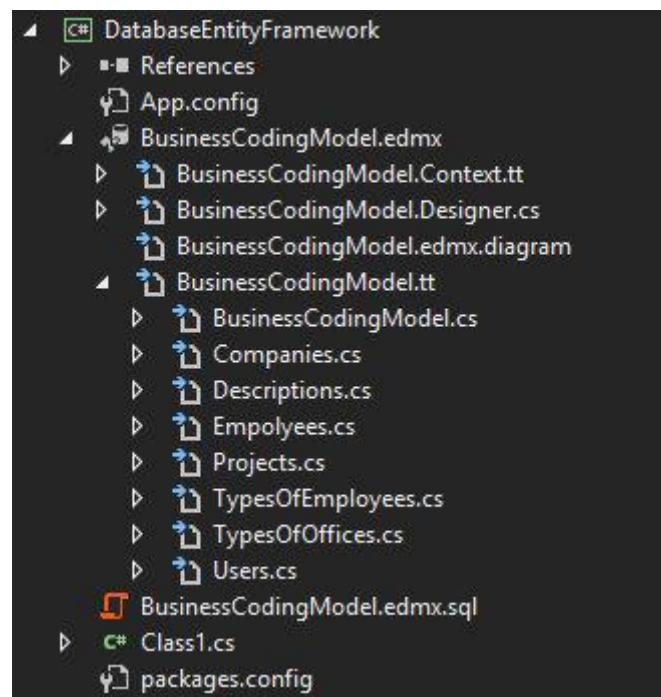
        Syntax Highlighting
    }
}

```

Rysunek 20 MainWindow.xaml.cs

- **Fields - selectedItem** – zmienne opisujące zaznaczone rekordy
- **Fields** – zmienne pomocnicze
- **Constructors** – konstruktor okna głównego
- **Controls** – obsługa kontrolek, które nie są Buttonami
- **Buttons** – obsługa przycisków
- **Creating Objects From Entry Form** – metody tworzące obiekty z danych z formularzy
- **Display Records** – metody wyświetlające wszystkie rekordy
- **Add Records** – metody dodające rekordy do bazy
- **Modify Records** – metody modyfikujące rekordy bazy
- **Delete Records** – metody usuwające rekordy z bazy
- **Additional Methods** – dodatkowe metody
- **Syntax Highlighting** – metody służące do podświetlania składni zapytań

8.2 DATABASE ENTITY FRAMEWORK



Rysunek 21 Drzewo DatabaseEntityFramework

- **BusinessCodingModel.edmx** – model bazy danych
 - **BusinessCodingModel.tt** – zawiera klasy encji wygenerowanych z modelu
- **BusinessCodingModel.edmx.sql** – zapytanie SQL tworzące bazę danych
- **Class1.cs** – klasa z funkcją main

```

public partial class TypesOfOffices : INotifyPropertyChanged
{
    public event PropertyChangedEventHandler PropertyChanged;

    virtual protected void OnPropertyChanged(string propertyName)
    {
        if (PropertyChanged != null)
            PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
    }

    private int t00fficeID;
    private string name;
    private short capacity;
    private double costOfHiring;

    public int T00fficeID
    {
        get
        {
            return t00fficeID;
        }
        set
        {
            t00fficeID = value;
            OnPropertyChanged("T00fficeID");
        }
    }

    public string Name[...]

    public short Capacity[...]

    public double CostOfHiring[...]
}

```

Rysunek 22 Klasa TypesOfOffices