

Sprawozdanie
Projektowanie Aplikacji Bazodanowych
Dział Kadr

Wykonane przez: Jarosław Jędruszcak

1. Opis projektu

Projekt ma służyć osobom odpowiedzialnym za rekrutację nowych pracowników dla firmy oraz osobom poszukującym pracy. Powinien być on zrealizowany w postaci aplikacji lokalnej oraz aplikacji WWW.

Aplikacja lokalna może być zrealizowana za pomocą dowolnego języka programowania obiektowego (np. C++, C#, Java itp.).

Aplikacja Internetowa musi być zrealizowana przy pomocy technologii ASP.NET.

Obydwie aplikacje powinny się realizować następujące funkcje:

- Przeglądanie ofert pracy firmy i możliwość zgłoszenia chęci udziału w rozmowie kwalifikacyjnej (podanie podstawowych danych).
- Zarządzanie (dodawanie, usuwanie) ofertami przez odpowiedzialną osobę.
- Zarządzanie (dodawanie, usuwanie) listą pracowników już zatrudnionych.
- Możliwość tworzenia list pracowników zatrudnionych i list wolnych etatów.
- Administrowania danymi – wprowadzenie, edytowanie i usuwanie podstawowych obiektów Bazy Danych.

Osoby poszukujące pracy powinny mieć możliwość przeglądania ofert firmy i w przypadku zainteresowania ofertą wypełniają podstawowy formularz CV (np. imię, nazwisko, wiek, rodzaj wykształcenia, miejsce zamieszkania, staż itp.).

Odpowiedzialny pracownik firmy powinien mieć możliwość dodawania nowych i usuwania nieaktualnych ofert pracy oraz edycji danych pracowników. Powinien również mieć on możliwość sporządzania list pracowników oraz list ofert pracy.

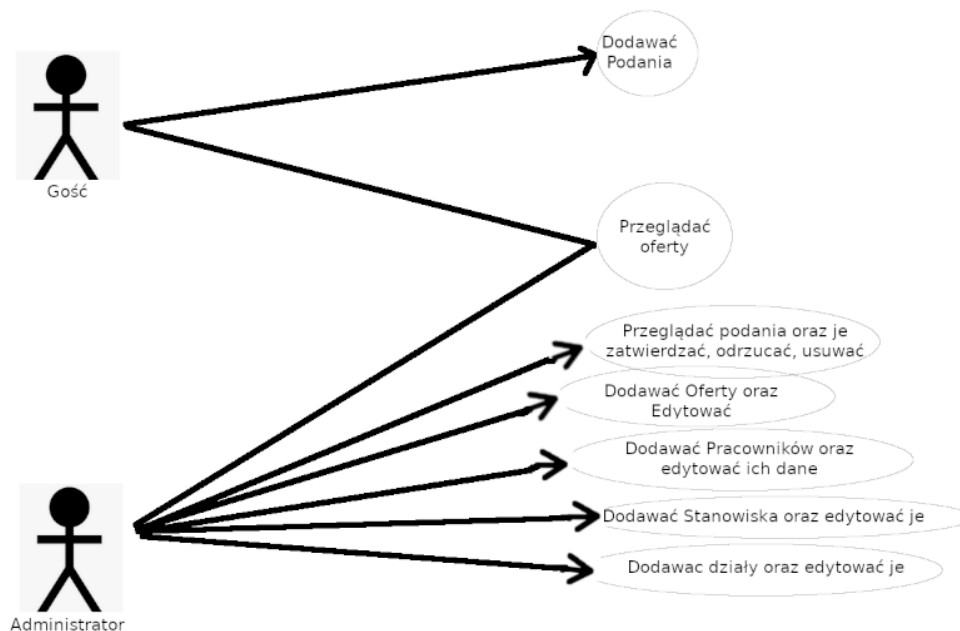
Baza powinna zawierać nie mniej niż 5 kluczowych encji, np.:

- PODANIA
- OFERTY
- PRACOWNICY
- DZIAŁ
- STANOWISKA

2. Kategorie użytkowników i funkcje programu.

Gość – może przeglądać oferty zamieszczone przez administratora działu kadr, oraz napisać podanie przy którym wybiera stanowisko oraz podaje swoje niezbędne dane.

Administrator – jest to pracownik z Uprawnieniami „Administrator“. Pracownik ten ma możliwość przeglądania wszystkich danych zawartych w bazie danych oraz jest w stanie dodawać, edytować oraz usuwać z niej dane, wyjątkiem jest tabela o nazwie Podania gdzie można tylko zmienić stan podania jak i to podanie usunąć.

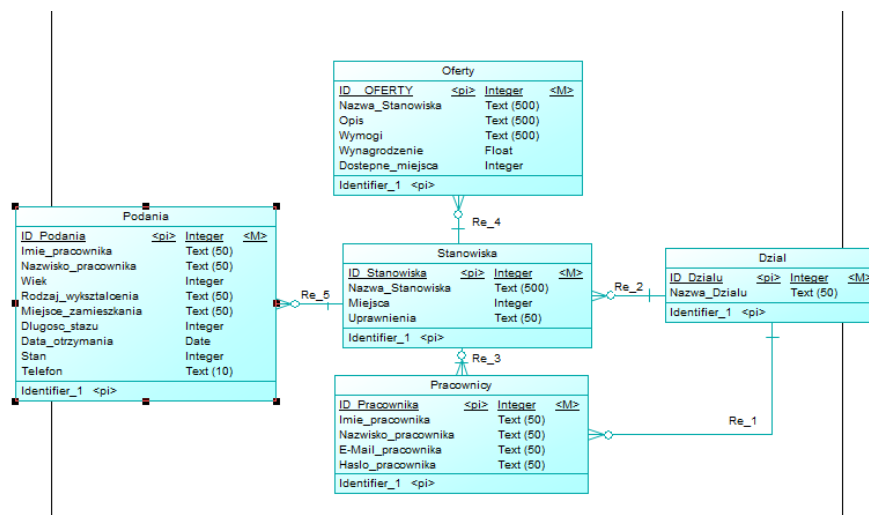


3. Opis Słownika danych

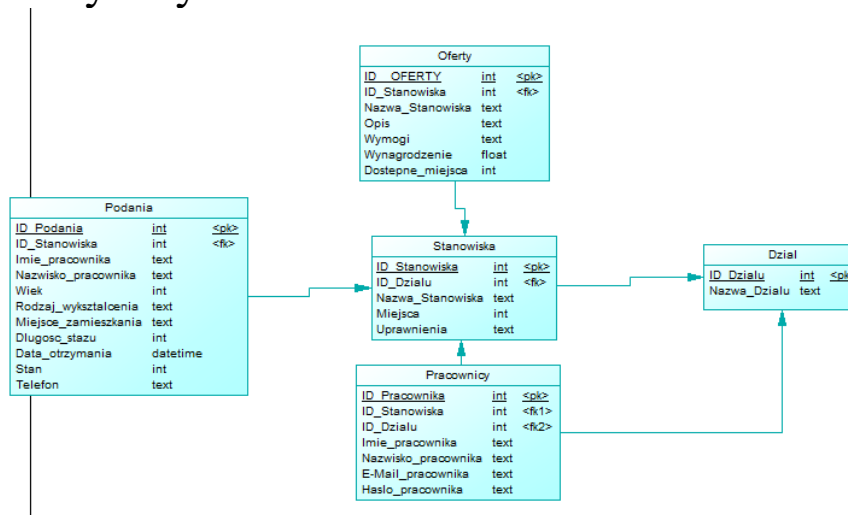
Słownik encji:

- Podania – Encja zawierająca podane podania w odpowiedzi na oferty.
- Oferty - Encja zawierająca zamieszczone oferty o prace, zamieszcza je osoba ze stanowiska posiadającego uprawnienia "rekrutacja".
- Pracownicy – Encja zawierająca liste pracowników pracujących pod stanowiskiem.
- Stanowiska – Encja zawierająca liste dostępnych stanowisk w danym dziale.
- Dział – Encja zawierająca liste Działów.

4. Model konceptualny



5. Model fizyczny



6. Wygenerowany kod SQL

```

/*=====*/
/* DBMS name:    Microsoft SQL Server 2012          */
/* Created on:   06.01.2019 15:18:57                */
/*=====*/

```

```

if exists (select 1
from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('OFERTY') and o.name = 'FK_OFERTY_RE_4_STANOWIS')
alter table OFERTY
drop constraint FK_OFERTY_RE_4_STANOWIS
go

```

```

if exists (select 1
from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('PODANIA') and o.name = 'FK_PODANIA_RE_5_STANOWIS')

```

```

alter table PODANIA
  drop constraint FK_PODANIA_RE_5_STANOWIS
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('PRACOWNICY') and o.name = 'FK_PRACOWNI_RE_1_DZIAL')
alter table PRACOWNICY
  drop constraint FK_PRACOWNI_RE_1_DZIAL
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('PRACOWNICY') and o.name =
'FK_PRACOWNI_RE_3_STANOWIS')
alter table PRACOWNICY
  drop constraint FK_PRACOWNI_RE_3_STANOWIS
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('STANOWISKA') and o.name = 'FK_STANOWIS_RE_2_DZIAL')
alter table STANOWISKA
  drop constraint FK_STANOWIS_RE_2_DZIAL
go

if exists (select 1
  from sysobjects
  where id = object_id('DZIAL')
  and type = 'U')
drop table DZIAL
go

if exists (select 1
  from sysindexes
  where id = object_id('OFERTY')
  and name = 'RE_4_FK'
  and indid > 0
  and indid < 255)
drop index OFERTY.RE_4_FK
go

if exists (select 1
  from sysobjects
  where id = object_id('OFERTY')
  and type = 'U')
drop table OFERTY
go

if exists (select 1

```

```

        from sysindexes
        where id = object_id('PODANIA')
        and name = 'RE_5_FK'
        and indid > 0
        and indid < 255)
drop index PODANIA.RE_5_FK
go

if exists (select 1
        from sysobjects
        where id = object_id('PODANIA')
        and type = 'U')
drop table PODANIA
go

if exists (select 1
        from sysindexes
        where id = object_id('PRACOWNICY')
        and name = 'RE_1_FK'
        and indid > 0
        and indid < 255)
drop index PRACOWNICY.RE_1_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('PRACOWNICY')
        and name = 'RE_3_FK'
        and indid > 0
        and indid < 255)
drop index PRACOWNICY.RE_3_FK
go

if exists (select 1
        from sysobjects
        where id = object_id('PRACOWNICY')
        and type = 'U')
drop table PRACOWNICY
go

if exists (select 1
        from sysindexes
        where id = object_id('STANOWISKA')
        and name = 'RE_2_FK'
        and indid > 0
        and indid < 255)
drop index STANOWISKA.RE_2_FK
go

if exists (select 1

```

```

        from sysobjects
        where id = object_id('STANOWISKA')
        and type = 'U')
drop table STANOWISKA
go

/*=====*/
/* Table: DZIAL */
/*=====*/
create table DZIAL (
    ID_DZIALU      int          not null IDENTITY,
    NAZWA_DZIALU   text          null,
    constraint PK_DZIAL primary key nonclustered (ID_DZIALU ASC)
)
go

/*=====*/
/* Table: OFERTY */
/*=====*/
create table OFERTY (
    ID_OFERTY      int          not null IDENTITY,
    ID_STANOWISKA  int          not null,
    NAZWA_STANOWISKA text      null,
    OPIS           text          null,
    WYMOGI         text          null,
    WYNAGRODZENIE  float        null,
    DOSTEPNE_MIEJSCA int        null,
    constraint PK_OFERTY primary key nonclustered (ID_OFERTY ASC)
)
go

/*=====*/
/* Index: RE_4_FK */
/*=====*/
create index RE_4_FK on OFERTY (
    ID_STANOWISKA ASC
)
go

/*=====*/
/* Table: PODANIA */
/*=====*/
create table PODANIA (
    ID_PODANIA      int          not null IDENTITY,
    ID_STANOWISKA  int          not null,
    IMIE_PRACOWNIKA text        null,
    NAZWISKO_PRACOWNIKA text    null,
    WIEK            int          null,
    RODZAJ_WYKSZTALCENIA text    null,
    MIEJSCE_ZAMIESZKANIA text    null,

```

```

    DLUGOSC_STAZU      int          null,
    DATA_OTRZYMANIA   datetime     null,
    STAN               int          null,
    TELEFON            text         null,
    constraint PK_PODANIA primary key nonclustered (ID_PODANIA ASC)
)
go

/*=====*/
/* Index: RE_5_FK                                     */
/*=====*/
create index RE_5_FK on PODANIA (
ID_STANOWISKA ASC
)
go

/*=====*/
/* Table: PRACOWNICY                                   */
/*=====*/
create table PRACOWNICY (
    ID_PRACOWNIKA      int          not null IDENTITY,
    ID_STANOWISKA      int          not null,
    ID_DZIALU          int          not null,
    IMIE_PRACOWNIKA    text         null,
    NAZWISKO_PRACOWNIKA text       null,
    E_MAIL_PRACOWNIKA  text         null,
    HASLO_PRACOWNIKA   text         null,
    constraint PK_PRACOWNICY primary key nonclustered (ID_PRACOWNIKA ASC)
)
go

/*=====*/
/* Index: RE_3_FK                                     */
/*=====*/
create index RE_3_FK on PRACOWNICY (
ID_STANOWISKA ASC
)
go

/*=====*/
/* Index: RE_1_FK                                     */
/*=====*/
create index RE_1_FK on PRACOWNICY (
ID_DZIALU ASC
)
go

/*=====*/
/* Table: STANOWISKA                                   */
/*=====*/

```



```

create table STANOWISKA (
  ID_STANOWISKA      int          not null IDENTITY,
  ID_DZIALU           int          not null,
  NAZWA_STANOWISKA   text         null,
  MIEJSCA             int          null,
  UPRAWNIENIA         text         null,
  constraint PK_STANOWISKA primary key nonclustered (ID_STANOWISKA ASC)
)
go

/*=====*/
/* Index: RE_2_FK                                     */
/*=====*/
create index RE_2_FK on STANOWISKA (
  ID_DZIALU ASC
)
go

alter table OFERTY
  add constraint FK_OFERTY_RE_4_STANOWIS foreign key (ID_STANOWISKA)
    references STANOWISKA (ID_STANOWISKA)
go

alter table PODANIA
  add constraint FK_PODANIA_RE_5_STANOWIS foreign key (ID_STANOWISKA)
    references STANOWISKA (ID_STANOWISKA)
go

alter table PRACOWNICY
  add constraint FK_PRACOWNI_RE_1_DZIAL foreign key (ID_DZIALU)
    references DZIAL (ID_DZIALU)
go

alter table PRACOWNICY
  add constraint FK_PRACOWNI_RE_3_STANOWIS foreign key (ID_STANOWISKA)
    references STANOWISKA (ID_STANOWISKA)
go

alter table STANOWISKA
  add constraint FK_STANOWIS_RE_2_DZIAL foreign key (ID_DZIALU)
    references DZIAL (ID_DZIALU)
go

```

7. Podstawowe rodzaje zapytań

Podstawowe rodzaje zapytań:

1. Otrzymać listę pracowników wyznaczonego działu.

```
SELECT Pracownicy.*, Dzial.Nazwa_Dzialu
FROM Dzial INNER JOIN Pracownicy ON Dzial.ID_Dzialu = Pracownicy.ID_Dzial
WHERE ((Dzial.Nazwa_Dzialu)="TU WYZNACZAM DZIAŁ");
```

2. Otrzymać listę kierowników działów.

```
SELECT Pracownicy.*, Stanowiska.Uprawnienia
FROM Stanowiska INNER JOIN Pracownicy ON Stanowiska.ID_Stalowiska =
Pracownicy.ID_Stalowiska
WHERE ((Stanowiska.Uprawnienia)="Kierownik");
```

3. Otrzymać listę posadowych stanowisk działu.

```
SELECT Oferty.*, Stanowiska.Nazwa_Stalowiska, Dzial.Nazwa_Dzialu
FROM (Dzial INNER JOIN Stanowiska ON Dzial.ID_Dzialu = Stanowiska.ID_Dzialu) INNER
JOIN Oferty ON Stanowiska.ID_Stalowiska = Oferty.ID_Stalowiska
WHERE ((Dzial.Nazwa_Dzialu)="TU WYZNACZAM DZIAŁ");
```

4. Otrzymać listę wolnych etatów.

```
SELECT Stanowiska.*
FROM Stanowiska
WHERE (Stanowiska.Miejsca > 0);
```

5. Otrzymać listę zgłoszonych ofert na wyznaczony termin

```
SELECT Podania.*
FROM Podania
WHERE ((Podania.Data_Otrzymania) = "TU WYZNACZAM TERMIN")
```

6. Otrzymać listę oraz liczbę osób zatrudnionych na odpowiednich s tanowiskach w różnych działach.

```
SELECT Pracownicy.Imie_pracownika, Stanowiska.Nazwa_Stalowiska
FROM Stanowiska INNER JOIN Pracownicy ON Stanowiska.ID_Stalowiska = Pracownicy.ID_
Stanowiska
WHERE (((Stanowiska .Nazwa_Stalowiska)="TU PODAC NAZWE"));
SELECT Count(Pracownicy.Imie_pracownika) AS PoliczOfImie_pracownika,
Stanowiska.Nazwa_Stalowiska
FROM Stanowiska INNER JOIN Pracownicy ON Stanowiska.ID_Stalowiska =
Pracownicy.ID_Stalowiska
GROUP BY Stabowiska.Nazwa_Stalowiska
```

HAVING (((Stanowiska.Nazwa_Stalowiska)="TU PODAC NAZWE"));

7. Otrzymać listę kandydatów na odpowiedni stanowiska według zgłoszonych ofert.

```
SELECT Podania.*, Stanowiska.Nazwa_Stalowiska  
FROM Stanowiska INNER JOIN Podania ON Stanowiska.ID_Stalowiska =  
Podania.ID_Stalowiska  
WHERE (((Stanowiska.Nazwa_Stalowiska)="TU PODAC NAZWE"));
```

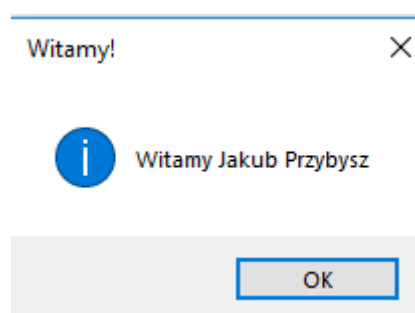
8. Aplikacja lokalna

Aplikacja lokalna została stworzona przy użyciu języka C#. Aplikacja łączy się z lokalną bazą danych, pobiera z niej dane, edytuje oraz usuwa, a sama baza danych przechowuje wszelkie zmiany.

- Logowanie się administratora



- Po poprawnym podaniu danych



- Okno administratora

The screenshot shows a web application window titled 'PrzeglądanieBDAdmin'. It features a navigation bar with buttons for 'Pracownicy', 'Stanowiska', 'Podania', 'Oferty', and 'Działy'. Below the navigation bar is a table with 14 rows of employee data. The table columns are: Lp., Stanowisko, Dział, Imię, Nazwisko, E-mail, Hasło, and Edytowanie. The 'Edytowanie' column contains buttons numbered 1 through 13. A 'Dodaj nowego' button is located at the bottom right of the table.

Lp.	Stanowisko	Dział	Imię	Nazwisko	E-mail	Hasło	Edytowanie
1	Kierownik	Frontend	Robot	Robotewski	ro	ro	24
2	Kierownik	Backend	Konrad	Gajewski	ga@co.net	admin	1
3	Kierownik	Frontend	Jakub	Przybysz	prac1@co.net	prac	2
4	Kierownik	Frontend	Gabriela	Olejniczak	prac2@co.net	prac	3
5	Kierownik	Backend	Zofia	Bielecka	prac3@co.net	prac	4
6	Programista	Frontend	Wiktoria	Ostrowska	prac4@co.net	prac	5
7	Sprzedaz	Frontend	Eryka	Wróblewska	prac5@co.net	prac	6
8	Programista	Obsługi	Jakub	Zak	prac6@co.net	prac	7
9	Programista	Obsługi	Antonina	Sawicka	prac7@co.net	prac	8
10	Programista	Backend	Karol	Nowak	prac8@co.net	prac	9
11	Programista	Obsługi	Marek	Zak	prac9@co.net	prac	10
12	Kierownik	Frontend	Oliwia	Kwiatkowska	prac10@co.net	prac	11
13	Kierownik	Frontend	Mikolaj	Janicki	prac11@co.net	prac	12
14	Programista	Backend	Dawid	Jablonski	prac12@co.net	prac	13

Dodaj nowego

- Dodawanie Pracownika

The screenshot shows a web application window titled 'Dodawanie_Pracownika'. It contains a form with the following fields: 'Stanowisko' (dropdown menu with 'Programista' selected), 'Dział' (dropdown menu with 'Frontend' selected), 'Imię' (text input), 'Nazwisko' (text input), 'E-mail' (text input), and 'Hasło' (text input). There is a green circular button with a left arrow on the bottom left and a 'Dodaj' button on the bottom right.

9. Aplikacja internetowa

Aplikacja internetowa została stworzona przy użyciu języka C# w technologii ASP.net. Została zaprogramowana tak, by łączyć się z lokalną bazą danych, pobierać dane i je w możliwy sposób edytować.

- Logowanie

Logowanie!

E-mail:	<input type="text" value="prac1@co.net"/>
---------	---

Hasło:	<input type="text" value="prac"/>
--------	-----------------------------------

- Po poprawnym podaniu danych

Logowanie!

E-mail:	<input type="text" value="prac1@co.net"/>
---------	---

Hasło:	<input type="text" value="prac"/>
--------	-----------------------------------

Witamy **Jakub Przybysz**

- Okno administratora

Panel Administratora

Pracownicy Stanowiska Podania Oferty Działy

Wróć

Pracownicy

Stanowisko	Dział	Imię	Nazwisko	E-mail	Hasło	
Nowe Stanowisko	Frontend	Joanna	Szulc	szu@co.net	admin	Edytuj
Kierownik	Backend	Konrad	Gajewski	ga@co.net	admin	Edytuj
Kierownik	Frontend	Jakub	Przybysz	prac1@co.net	prac	Edytuj
Kierownik	Frontend	Gabriela	Olejniczak	prac2@co.net	prac	Edytuj
Kierownik	Backend	Zofia	Bielecka	prac3@co.net	prac	Edytuj
Programista	Frontend	Wiktoria	Ostrowska	prac4@co.net	prac	Edytuj
Programista	Obsługi konsumenta	Eryk	Wróblewski	prac5@co.net	prac	Edytuj
Programista	Obsługi konsumenta	Jakub	Zak	prac6@co.net	prac	Edytuj
Programista	Obsługi konsumenta	Antonina	Sawicka	prac7@co.net	prac	Edytuj
Programista	Backend	Karol	Nowak	prac8@co.net	prac	Edytuj
Programista	Obsługi konsumenta	Marek	Zak	prac9@co.net	prac	Edytuj
Kierownik	Frontend	Oliwia	Kwiatkowska	prac10@co.net	prac	Edytuj
Kierownik	Frontend	Mikołaj	Janicki	prac11@co.net	prac	Edytuj
Programista	Backend	Dawid	Jablonski	prac12@co.net	prac	Edytuj
Programista	Frontend	Marek	Zielinski	prac13@co.net	prac	Edytuj
Programista	Frontend	Patryk	Bak	prac14@co.net	prac	Edytuj
Programista	Frontend	Jakub	Bak	prac15@co.net	prac	Edytuj

- Dodawanie Pracownika

Dodawanie Pracownika

Dział

Stanowisko

Imię pracownika:

Nazwisko pracownika:

E-mail:

Hasło: