

POLITECHNIKA KOSZALIŃSKA

Baza PKS

Dokumentacja z przedmiotu
systemy zarządzania relacyjnymi
bazami danych

Dawid Gawński

U-8421

Spis treści

1.	Opis zadania
2.	Wyznaczenia kategorii użytkowników i funkcji systemu dla każdej z tych kategorii.....
3.	Stworzenie diagramu przypadków użycia (UML).....
4.	Opis słownika danych.....
5.	Opis modelu konceptualnego
6.	Opis modelu fizycznego
7.	Kody SQL do tworzenia tabel bazy danych
8.	Kody zapytań SQL do bazy danych wraz z opisem
9.	Opis aplikacji lokalnej , diagramy klas aplikacji.....
a)	Opis funkcjonalności
b)	Wybrana technologia.....
c)	Diagram klas.....
d)	Opis struktury programu.....
e)	Opis najważniejszych funkcji programu.....
10.	Opis aplikacji webowej
a)	Opis funkcjonalności
b)	Wybrana technologia.....
c)	Graf przejść
d)	Opis struktury programu.....
10.	Wnioski

1. Opis zadania

Zadania w ramach przedmiotu z „Zarządzania relacyjnymi bazami danych” polegały na zaprojektowaniu bazy danych, utworzeniu aplikacji lokalnej oraz internetowej, która ma zarządzać tą bazą oraz dokumentacji.

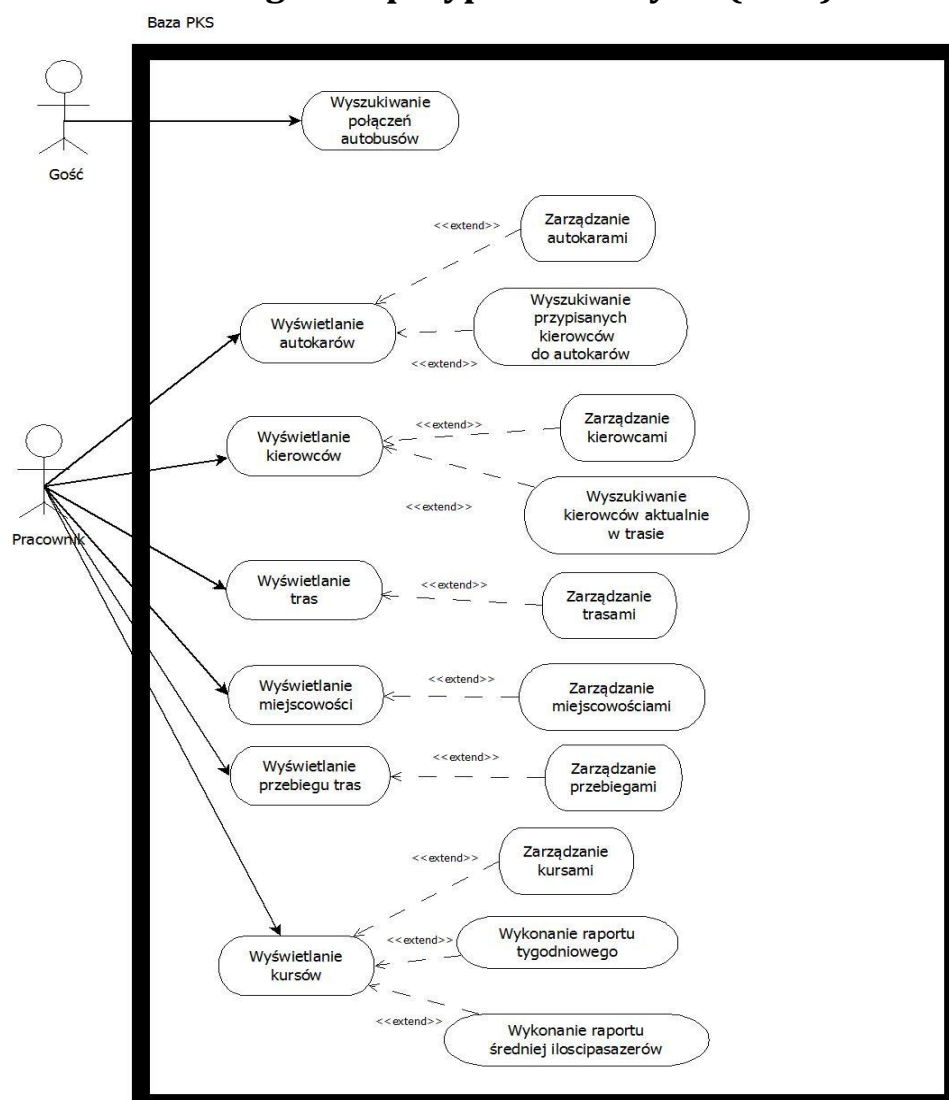
Mój projekt przeznaczony jest dla ludzi szukających połączeń autobusów PKS według wybranych przystanków, daty oraz czasu i dla pracowników, by w sposób łatwy byli w stanie zarządzać tymi połączeniami.

2. Wyznaczenia kategorii użytkowników i funkcji systemu dla każdej z tych kategorii

W oby dwóch aplikacjach zostały utworzone następujące kategorie użytkowników:

- użytkownik niezalogowany, który ma on dostęp do wyszukiwania połączeń autobusów potrzebnych dla tego użytkownika;
- użytkownik zalogowany, który jest pracownikiem firmy i ma możliwość edytowania wszystkich tabel.

3. Stworzenie diagramu przypadków użycia (UML)



4. Opis słownika danych

<u>ZBIÓR ENCJI „Kierowca”</u>				
Nazwa	Opis	Typ	Czy złożony?	Dziedzina atrybutu
ID_kierowcy	Numer jednoznacznie identyfikujący kierowcę	Liczba całkowita	Nie	Zbiór dwucyfrowych liczb całkowitych bez powtórzeń
Imie_Nazwisko	Imię i nazwisko kierowcy	Znakowy	Tak, Imię Nazwisko	Zbiór łańcuchów znakowych opisujących imię i nazwisko kierowcy

<u>ZBIÓR ENCJI „Autokar”</u>				
Nazwa	Opis	Typ	Czy złożony?	Dziedzina atrybutu
ID_autokaru	Numer jednoznacznie identyfikujący autokar	Liczba całkowita	Nie	Zbiór dwucyfrowych liczb całkowitych bez powtórzeń
Nr_rejestr	Numer rejestracyjny autokaru	Znakowy	Nie	Zbiór łańcuchów znakowych opisujących numer rejestracyjny autokaru
Marka	Marka autokaru	Znakowy	Nie	Zbiór łańcuchów znakowych opisujących markę autokaru
Model	Model autokaru	Znakowy	Nie	Zbiór łańcuchów znakowych opisujących model autokaru
Liczba_miejsc	Zmienna identyfikująca liczbę miejsc w autokarze	Liczba całkowita	Nie	Zbiór dwucyfrowych liczb całkowitych określających liczbę miejsc w autokarze

<u>ZBIÓR ENCJI „Przebieg”</u>				
Nazwa	Opis	Typ	Czy złożony?	Dziedzina atrybutu
ID_przebiegu	Numer jednoznacznie identyfikujący przebieg pomiędzy miejscowościami	Liczba całkowita	Nie	Zbiór dwucyfrowych liczb całkowitych bez powtórzeń
Roznica_czasu	Różnica czasu pomiędzy daną miejscowością a miejscowością początkową	Czas	Nie	Zbiór łańcuchów znakowych opisujących imię i nazwisko pracownika
Kolejnosc	Zmienna identyfikująca kolejność występowania przystanku na trasie	Liczba całkowita	Nie	Zbiór dwucyfrowych liczb całkowitych określających kolejność przystanków

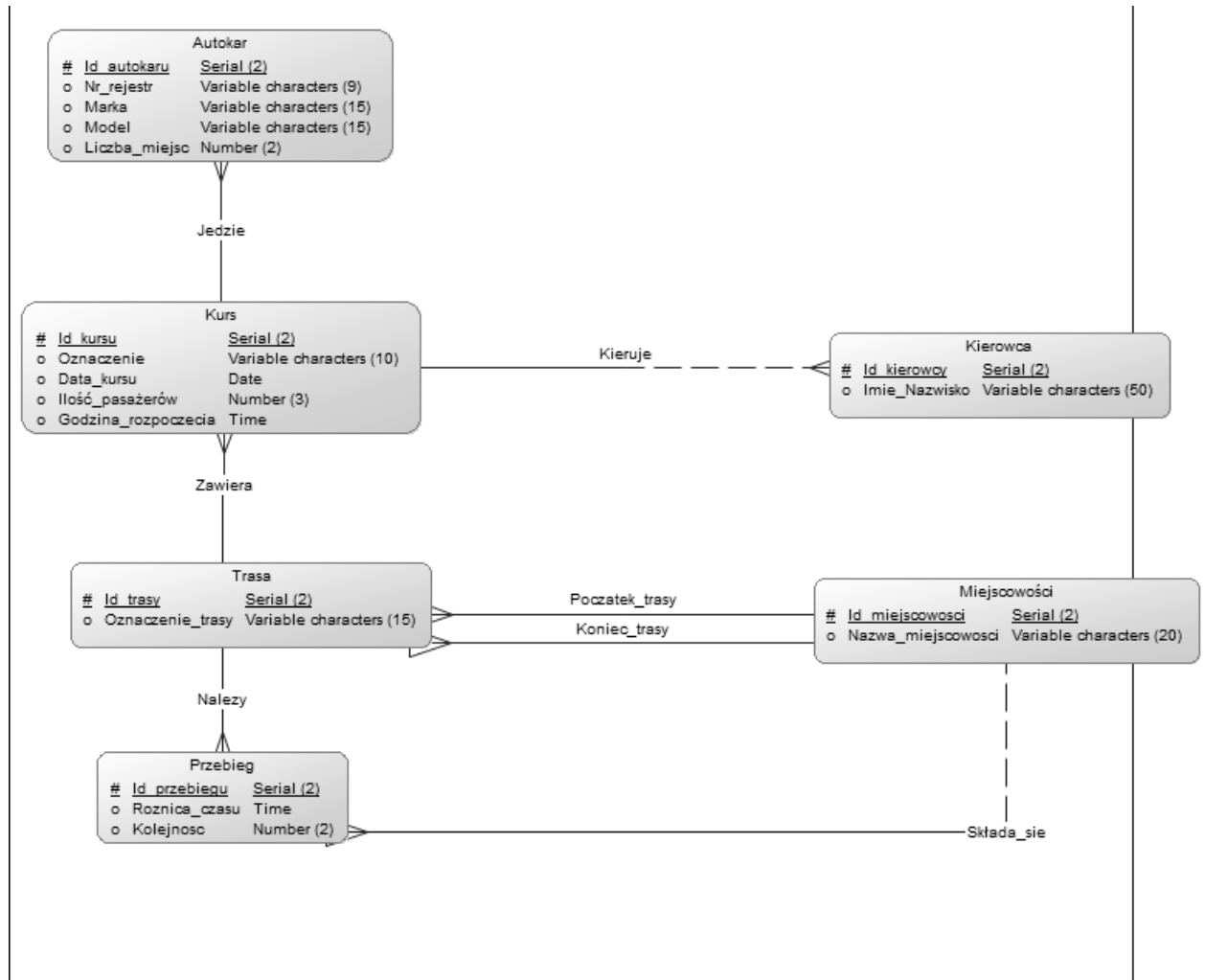
<u>ZBIÓR ENCJI „Miejscowości”</u>				
Nazwa	Opis	Typ	Czy złożony?	Dziedzina atrybutu
ID_miejscowosci	Numer jednoznacznie identyfikujący miejscowość	Liczba całkowita	Nie	Zbiór dwucyfrowych liczb całkowitych bez powtórzeń
Nazwa_miejscowosci	Nazwa miejscowości w której znajdują się przystanek	Znakowy	Tak	Zbiór łańcuchów znakowych opisujących nazwę miejscowości

<u>ZBIÓR ENCJI „Trasa”</u>				
Nazwa	Opis	Typ	Czy złożony?	Dziedzina atrybutu
ID_trasy	Numer jednoznacznie identyfikujący trasę	Liczba całkowita	Nie	Zbiór dwucyfrowych liczb całkowitych bez powtórzeń
Oznaczenie_trasy	Oznaczenie trasy	Znakowy	Nie	Zbiór łańcuchów znakowych opisujących trasę

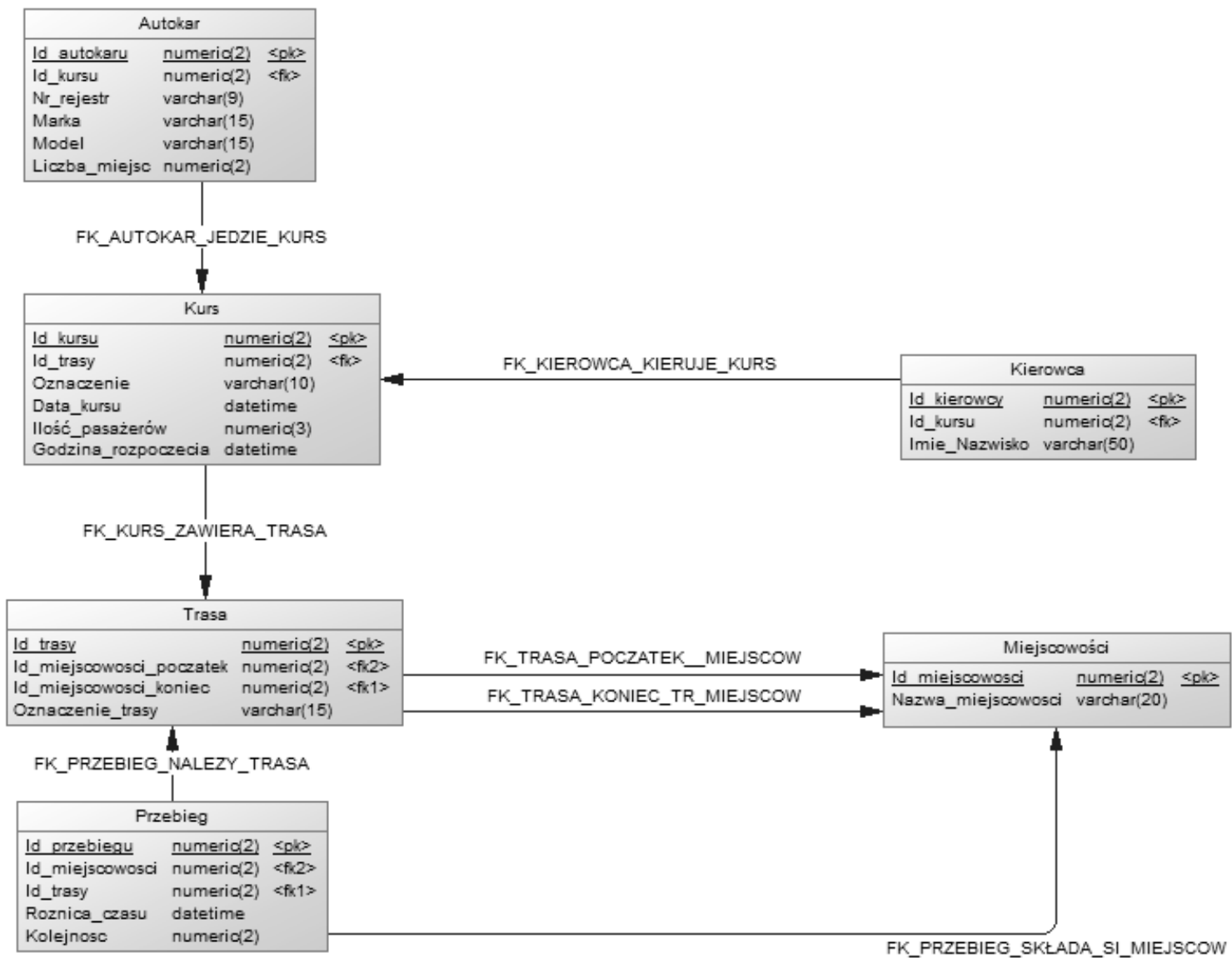
<u>ZBIÓR ENCJI „Kurs”</u>				
Nazwa	Opis	Typ	Czy złożony?	Dziedzina atrybutu
ID_kursu	Numer jednoznacznie identyfikujący kurs	Liczba całkowita	Nie	Zbiór dwucyfrowych liczb całkowitych bez powtórzeń
Oznaczenie	Oznaczenie kursu	Znakowy	Nie	Zbiór łańcuchów znakowych opisujących kurs

Data_kursu	Data wystąpienia kursu	Data	Tak Dzień Miesiąc Rok	Data opisująca datę wystąpienia kursu
Ilosc_pasazerow	Ilość pasażerów, która skorzystała z danego kursu	Liczba całkowita	Nie	Zbiór trzycyfrowych liczb całkowitych określających liczbę pasażerów
Godzina_rozpoczecia	Godzina rozpoczęcia kursu na przystanku początkowym	Czas	Nie	Zbiór godzin z minutami określających rozpoczęcie danego kursu

5. Opis modelu konceptualnego



6.Opis modelu fizycznego



7. Kody SQL do tworzenia tabel bazy danych

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

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

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

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

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

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

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

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

```
if exists (select 1
  from sysindexes
  where id = object_id('Przebieg')
  and name = 'Należy_FK'
  and indid > 0
```

```

        and indid < 255)
drop index Przebieg.Nalezy_FK
go

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

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

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

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

```

```

/*=====*/
/* Table: Autokar */
/*=====*/

```

```

create table Autokar (
    Id_autokaru      numeric(2)      identity,
    Id_kursu         numeric(2)      not null,
    Nr_rejestr       varchar(9)      null,
    Marka            varchar(15)     null,
    Model            varchar(15)     null,
    Liczba_miejsc    numeric(2)      null,
    constraint PK_AUTOKAR primary key nonclustered (Id_autokaru)
)
go

```

```

/*=====*/
/* Index: Jedzie_FK */
/*=====*/

```

```

create index Jedzie_FK on Autokar (
    Id_kursu ASC
)
go

```

```

/*=====*/
/* Table: Kierowca */
/*=====*/

```

```

create table Kierowca (
    Id_kierowcy      numeric(2)      identity,
    Id_kursu         numeric(2)      null,
    Imie_Nazwisko    varchar(50)    null,
    constraint PK_KIEROWCA primary key nonclustered (Id_kierowcy)
)
go

```

```

/*=====*/
/* Index: Kieruje_FK */
/*=====*/

```

```

create index Kieruje_FK on Kierowca (
    Id_kursu ASC
)
go

```

```

/*=====*/
/* Table: Kurs */
/*=====*/
create table Kurs (
    Id_kursu      numeric(2)      identity,
    Id_trasy      numeric(2)      not null,
    Oznaczenie    varchar(10)     null,
    Data_kursu    datetime        null,
    Ilość_pasażerów numeric(3)     null,
    Godzina_roz poczenia datetime  null,
    constraint PK_KURS primary key nonclustered (Id_kursu)
)
go

/*=====*/
/* Index: Zawiera_FK */
/*=====*/
create index Zawiera_FK on Kurs (
    Id_trasy ASC
)
go

/*=====*/
/* Table: Miejscowości */
/*=====*/
create table Miejscowości (
    Id_miejscowosci numeric(2)      identity,
    Nazwa_miejscowosci varchar(20)  null,
    constraint PK_MIEJSCOWOŚCI primary key nonclustered (Id_miejscowosci)
)
go

/*=====*/
/* Table: Przebieg */
/*=====*/
create table Przebieg (
    Id_przebiegu    numeric(2)      identity,
    Id_miejscowosci numeric(2)      not null,
    Id_trasy        numeric(2)      not null,
    Roznica_czasu   datetime        null,
    Kolejnosć      numeric(2)      null,
    constraint PK_PRZEBIEG primary key nonclustered (Id_przebiegu)
)
go

/*=====*/
/* Index: Należy_FK */
/*=====*/
create index Należy_FK on Przebieg (
    Id_trasy ASC
)
go

/*=====*/
/* Index: Składa_sie_FK */
/*=====*/
create index Składa_sie_FK on Przebieg (
    Id_miejscowosci ASC
)
go

/*=====*/
/* Table: Trasa */
/*=====*/
create table Trasa (
    Id_trasy      numeric(2)      identity,
    Id_miejscowosci_poczatek numeric(2)      not null,
    Id_miejscowosci_koniec numeric(2)      not null,
    Oznaczenie_trasy varchar(15)     null,
    constraint PK_TRASA primary key nonclustered (Id_trasy)
)
go

```

```

/*=====*/
/* Index: Koniec_trasy_FK */
/*=====*/
create index Koniec_trasy_FK on Trasa (
Id_miejscowosci_koniec ASC
)
go

/*=====*/
/* Index: Poczatek_trasy_FK */
/*=====*/
create index Poczatek_trasy_FK on Trasa (
Id_miejscowosci_poczatek ASC
)
go

```

8. Kody zapytań SQL do bazy danych wraz z opisem

1.Otrzymać przebieg trasy kursu pomiędzy punktem startowym a końcowy

```

Select m.Nazwa_miejscowosci
From Trasa AS t
Inner join Kurs AS k
ON t.Id_trasy=k.Id_trasy
Inner join Przebieg AS p
on p.Id_trasy=t.Id_trasy
Inner join Miejscowosci AS m
on m.Id_miejscowosci=p.Id_miejscowosci
Where( t.Id_miejscowosci_poczatek!=p.Id_miejscowosci OR t.Id_miejscowosci_koniec!=p.Id_miejscowosci )
and k.Id_kursu=2
Order by p.Kolejnosc

```

2.Otrzymać kierowców którzy są aktualnie w trasie

```

Select k.Imie_Nazwisko
From Kierowca AS k
Inner Join kurs AS ku
ON k.Id_kursu=ku.Id_kursu
WHERE ku.Data_kursu=CONVERT(date,SYSDATETIME()) and
ku.Godzina_rozpoczecia<CONVERT(time,SYSDATETIME())

```

3.Otrzymać raport tygodniowy na temat ilości pasażerów korzystających z danego kursu

```

Select SUM(Ilość_pasażerów) AS Tygodniowy
FROM Kurs
WHERE Data_kursu<CONVERT(date,SYSDATETIME())

```

4.Otrzymać listę połączeń pomiędzy miejscowościami dla różnych kursów

```
Select k.Id_kursu,m.Nazwa_miejscowosci,k.Godzina_roz poczeczia,p.Roznica_czasu
From Kurs AS k
INNER JOIN Trasa AS t
ON k.Id_trasy=t.Id_trasy
INNER JOIN Przebieg AS p
ON p.Id_trasy=t.Id_trasy
INNER JOIN Miejscowości AS m
ON m.Id_miejscowosci=p.Id_miejscowosci
WHERE p.Id_miejscowosci=5 OR p.Id_miejscowosci=1
Order by p.Roznica_czasu
```

5.Otrzymać raport na temat ilości pasażerów korzystających z danego kursu

```
Select (SUM(Ilość_pasażerów)/COUNT(Id_kursu)) AS Tygodniowy
FROM Kurs AS k
INNER JOIN Trasa AS t
on k.Id_trasy=t.Id_trasy
WHERE Data_kursu<=CONVERT(date,SYSDATETIME()) and k.Id_kursu=2
```

6.Otrzymać raport harmonogramu ruchu pojazdu po trasie, skojarzoną z wybranym kursem

```
Select m.Nazwa_miejscowosci,k.Godzina_roz poczeczia,p.Roznica_czasu
From Trasa AS t
Inner join Kurs AS k
ON t.Id_trasy=k.Id_trasy
Inner join Przebieg AS p
on p.Id_trasy=t.Id_trasy
Inner join Miejscowości AS m
on m.Id_miejscowosci=p.Id_miejscowosci
Where( t.Id_miejscowosci_poczatek!=p.Id_miejscowosci OR
t.Id_miejscowosci_koniec!=p.Id_miejscowosci ) and k.Id_kursu=2
Order by p.Kolejność
```

7.Otrzymać raport pro przeznaczenie kierowców pojazdom

```
Select k.Imie_Nazwisko
FROM Kierowca AS k
INNER JOIN Kurs AS ku
ON ku.Id_kursu=k.Id_kursu
INNER JOIN Autokar AS a
on a.Id_kursu=k.Id_kursu
```

9. Opis aplikacji lokalnej , diagramy klas aplikacji

a) Opis funkcjonalności

Aplikacja lokalna udostępnia następujące funkcjonalności:

- Przeglądanie połączeń autokarów, po uprzednim podaniu dodatkowych parametrów;
- Administrowanie autokarami (usuwanie, edycja, dodawanie) przez upoważnionego pracownika;
- Wyszukiwanie przypisanych kierowców do wybranego autokaru;
- Administrowanie kierowcami (usuwanie, edycja, dodawanie) przez upoważnionego pracownika;
- Wyszukiwanie kierowców, którzy są w aktualnej chwili w trasie;
- Administrowanie trasami (edycja, usuwanie) przez upoważnionego pracownika;
- Administrowanie miejscowościami (edycja, usuwanie) przez upoważnionego pracownika;
- Administrowanie przebiegami pomiędzy przystankami (edycja, usuwanie) przez upoważnionego pracownika;
- Administrowanie kursami (edycja, usuwanie) przez upoważnionego pracownika;

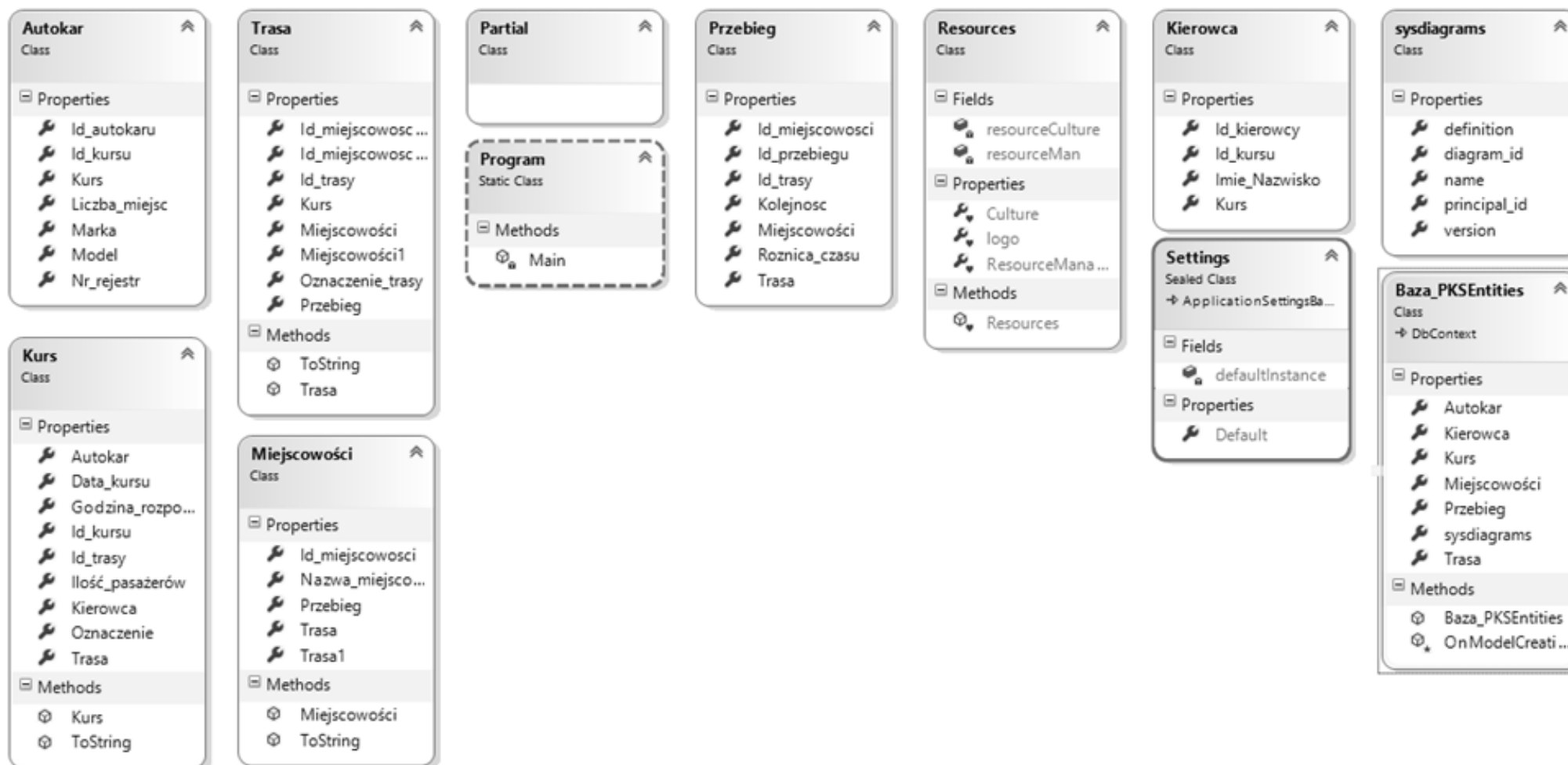
- Generowanie raportu tygodniowego związanego z liczbą pasażerów danego kursu;
- Generowanie raportu liczącego średnią liczbę pasażerów od początku istnienia danego kursu

b) Wybrana technologia

Aplikacja została zrealizowana w C# przy użyciu środowiska Microsoft Visual Studio 2015 oraz następującej biblioteki:

- Entity Framework 6.0 w celu połączenia bazy danych z aplikacją;

c) Diagram klas



frmmiestowosci

Class
→ Form

Fields

- bsmiestowosci
- btcbmiestowo...
- btdodajmiestow...
- btedytujmiestow...
- btusunmiestow...
- components
- db
- dgmiejscowosci
- ld_miejscowosci
- lbdanemiejscow...
- nazwamiejscow...
- pblogo

Methods

- btcbmiestowo...
- btdodajmiestow...
- btedytujmiestow...
- btusunmiestow...
- Dispose
- frmmiestowosci
- InitializeCompo...

frmprzebieg

Class
→ Form

Fields

- bsprzebieg
- bstrasy
- btcbprzebieg
- btdodajprzebieg
- btedytujprzebieg
- btusunprzebieg
- components
- dataGridView2
- db
- dgprzebieg
- kolejnoscDataG...
- lbdaneprzebieg
- max
- Miejscowosci
- oznaczenietras...
- pblogo
- roznicaczasuDa...

Methods

- btcbprzebieg_C...
- btdodajprzebie...
- btedytujprzebie...
- btusunprzebieg...
- dataGridView2_...
- Dispose
- frmprzebieg
- getMax
- InitializeCompo...

frmprzebiegedy...

Class
→ Form

Fields

- bsmiasto
- btaneditprzebieg
- btokeditprzebieg
- cbmiasto
- components
- db
- lbidtrasy
- lbkolejka
- lbrozczasu
- pblogo
- przebieg
- tbidtrasy
- tbkolejnosc
- tbroznicaczasu
- toadd
- trasa

Properties

- kolejka

Methods

- btaneditprzebi...
- btokeditprzebi...
- Dispose
- frmprzebieged...
- InitializeCompo...

frmtrasy

Class
→ Form

Fields

- bsmiastokon
- bsmiastopocz
- bstrasy
- btcbtrasy
- btdodajtrasy
- btedytujtrasy
- btusuntrasy
- components
- db
- dgtrasy
- lbdanetrasy
- Miejscowosci
- Miejscowosci1
- oznaczenietras...
- pblogo

Methods

- btcbtrasy_Click
- btdodajtrasy_Cl...
- btedytujtrasy_C...
- btusuntrasy_Click
- Dispose
- frmtrasy
- InitializeCompo...

frmtrasyedycja

Class
→ Form

Fields

- bsmiastokon
- bsmiastopocz
- btanedittrasy
- btokedittrasy
- cbmiastokoniec
- cbmiastopocz
- components
- db
- label3
- lbidmiejscakon
- lbidmiejscapocz
- pblogo
- tboznacztasy
- toadd
- trasa

Properties

- route

Methods

- btanedittrasy_C...
- btokedittrasy_C...
- Dispose
- frmtrasyedycja
- InitializeCompo...

frmmiestowosc...

Class
→ Form

Fields

- btaneditmiejsc...
- btokeditmiejsc...
- components
- lbnazwamiejscow...
- miejscowosc
- pblogo
- tbnazwamiejsc...
- toadd

Properties

- city

Methods

- btaneditmiejsc...
- btokeditmiejsc...
- Dispose
- frmmiestowosc...
- InitializeCompo...

frmkierowcy

Class
→ Form

Fields

- bskierowcy
- bskierowcywtr
- bskurs
- btcbkierowcy
- btdodajkierowcy
- btedytujkierowcy
- btszukajkierowcy
- btusunmierowcy
- components
- dataGridView1
- dataGridView2
- db
- gr
- imieNazwiskoD...
- imieNazwiskoD...
- Kurs
- label1
- lbdanekierowcy
- pblogo
- ts

Methods

- btcbkierowcy_C...
- btdodajkierowc...
- btedytujkierow...
- btszukajkierow...
- btusunmierowcy...
- Dispose
- frmkierowcy
- InitializeCompo...

frmkierowcyedy...

Class
→ Form

Fields

- btaneditkierowcy
- btokeditkierowcy
- components
- kierowca
- kurs
- label1
- label2
- pblogo
- tbidkursu
- tbimienazwisko
- toadd

Properties

- Driver

Methods

- btaneditkierow...
- btokeditkierow...
- Dispose
- frmkierowcyed...
- InitializeCompo...

frmkursy

Class
→ Form

Fields

- bskursy
- btcbkursy
- btdodajkursy
- btedyujkursy
- btrtyg
- btsrednia
- btusunkursy
- components
- datakursuData...
- db
- dgkursy
- godzinarozpoc...
- iloscPasazerow...
- label1
- label2
- label3
- label4
- label5
- pblogo
- tbgodzrozp
- tbidtrasy
- tbiloscPasazerow
- tbOznaczenieKur...
- toadd

Properties

- position

Methods

- btaneditkurs_Cl...
- btokeditkurs_Cl...
- Dispose
- frmkursyedycja
- InitializeCompo...

frmkursyedycja

Class
→ Form

Fields

- btaneditkurs
- btokeditkurs
- components
- dtpkurs
- kurs
- label1
- label2
- label3
- label4
- label5
- pblogo
- tbgodzrozp
- tbidtrasy
- tbiloscPasazerow
- tbOznaczenieKur...
- toadd

Properties

- position

Methods

- btaneditkurs_Cl...
- btokeditkurs_Cl...
- Dispose
- frmkursyedycja
- InitializeCompo...

frmautokaryedy...

Class
→ Form

Fields

- autokar
- btaneditautokar
- btokeditautokar
- components
- db
- label1
- label2
- label3
- label4
- label5
- lbidkursu
- pblogo
- tbidkursu
- tbmarka
- tbmiejsca
- tbmodel
- tbnrrejst
- toadd

Properties

- Bus

Methods

- btaneditautoka...
- btokeditautoka...
- Dispose
- frmautokaryed...
- InitializeCompo...

frmdyzurny

Class
→ Form

Fields

- btautokary
- btokierowcy
- btkurs
- btmiejscowosci
- btprzebieg
- bttrasy
- btwyloguj
- components
- pblogo

Methods

- btautokary_Click
- btokierowcy_Click
- btkurs_Click
- btmiejscowosci...
- btprzebieg_Click
- bttrasy_Click
- btwyloguj_Click
- Dispose
- frmdyzurny
- InitializeCompo...

frmPasazer

Class
→ Form

Fields

- bskoniec
- bskurs
- bsmiej
- bspoczek
- bsprzebieg
- btlog
- btwyszukaj
- cbprzystanekkon
- cbprzystanekp...
- components
- dataGridView1
- dataGridView2
- datakursuData...
- db
- dtpdatapaszera
- lbdata
- lbdo
- lbgodzina
- lbprzystanekko...
- lbprzystanekpo...
- lbz
- Miejscowosci
- oznaczenieData...
- pblogo
- roznicaCzasuDa...
- tbgodzina

Methods

- btlog_Click
- btwyszukaj_Click
- dataGridView1_...
- Dispose
- frmPasazer
- frmPasazer_Load
- InitializeCompo...

frmlog

Class
→ Form

Fields

- btcblog
- btlog
- components
- lbhaslo
- lblogin
- pblogo
- tbhaslo1
- tblogin1

Methods

- btcblog_Click
- btlog_Click
- Dispose
- frmlog
- InitializeCompo...

frmautokary

Class
→ Form

Fields

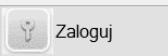
- bsautokar
- bskierowca
- btcbautokary
- btdodajautokary
- btedyujautokary
- btszukajautokary
- btusunautokary
- components
- db
- dgautokary
- dgkierowcy
- imieNazwiskoD...
- Kurs
- label2
- lbdata
- lbdataautoakru
- liczbamiejscDat...
- markaDataGrid...
- modelDataGrid...
- nrrejestrDataGr...
- pblogo

Methods

- btcbautokary_C...
- btdodajautokar...
- btedyujautoka...
- btszukajautoka...
- btusunautokary...
- Dispose
- frmautokary
- InitializeCompo...

d) Opis struktury programu

Wyszukiwarka połączeń PKS

Przystanek początkowy
Z:

Przystanek końcowy
Do:

Podaj datę:

Podaj godzinę:



Oznaczenie	Data kursu	Godzina odjazdu	Miejscowości
------------	------------	-----------------	--------------

Logowanie





Login:


Hasło:


 


Panel administracyjny





 Autokary

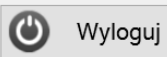
 Kierowcy

 Trasy


 Kursy

 Miejscowości

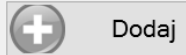
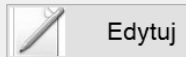
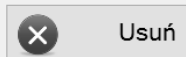
 Przebieg



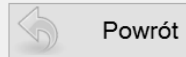
Trasy



Dane trasy

Miejscowość początek	Miejscowość koniec	Oznaczenie trasy
----------------------	--------------------	------------------



+

Dodaj

Edytuj

x

Usuń

Dane przebiegu

Oznaczenie_trasy	Miejscowości	Roznica_czasu	Kolejnosc
------------------	--------------	---------------	-----------

Powrót

+

Dodaj

Edytuj

x

Usuń

Dane miejscowości

Id_miejscowosci	Nazwa_miejscowosci
-----------------	--------------------

Powrót

+

Dodaj

Edytuj

x

Usuń

Dane kursu

Oznaczenie	Data_kursu	Ilość_pasażerów	Godzina_rozpoczec	Trasa
------------	------------	-----------------	-------------------	-------

Oznaczenie kursu:

Σ

Tygodniowy

\bar{x}

Średnia

Ilość pasażerów:

Powrót

Wyszukaj

+

Dodaj

Edytuj

x

Usuń

Oznaczenie:

Data kursu:

wtorek . 31 maja 2016

Ilość pasażerów:

Godzina rozpoczęcia:

Oznaczenie trasy:

OK

Anuluj

Powrót

e) Opis najważniejszych funkcji programu

```
public frmAutokary()  
{  
    InitializeComponent();  
    bsAutokar.DataSource = db.Autokar.ToList();  
}
```

Rysunek Funkcja pobierająca wartości z bazy danych

```

private void btdodajautokary_Click(object sender, EventArgs e)
{
    frmautokaryedycja add = new frmautokaryedycja(null);
    if (add.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    {
        db.Autokar.Add(add.Bus);
        db.SaveChanges();
        bsautokar.DataSource = db.Autokar.ToList();
    }
}

private void btedyujautokary_Click(object sender, EventArgs e)
{
    frmautokaryedycja edit = new frmautokaryedycja(bsautokar.Current as Autokar);
    edit.ShowDialog();
    db.SaveChanges();
    bsautokar.DataSource = db.Autokar.ToList();
}

private void btusunautokary_Click(object sender, EventArgs e)
{
    Autokar toremove = bsautokar.Current as Autokar;
    db.Autokar.Remove(toremove);
    db.SaveChanges();
    bsautokar.DataSource = db.Autokar.ToList();
}

```

Rysunek Fragment kodu odpowiedzialny za usuwanie wiesz z bazy danych

10. Opis aplikacji webowej

a) Opis funkcjonalności

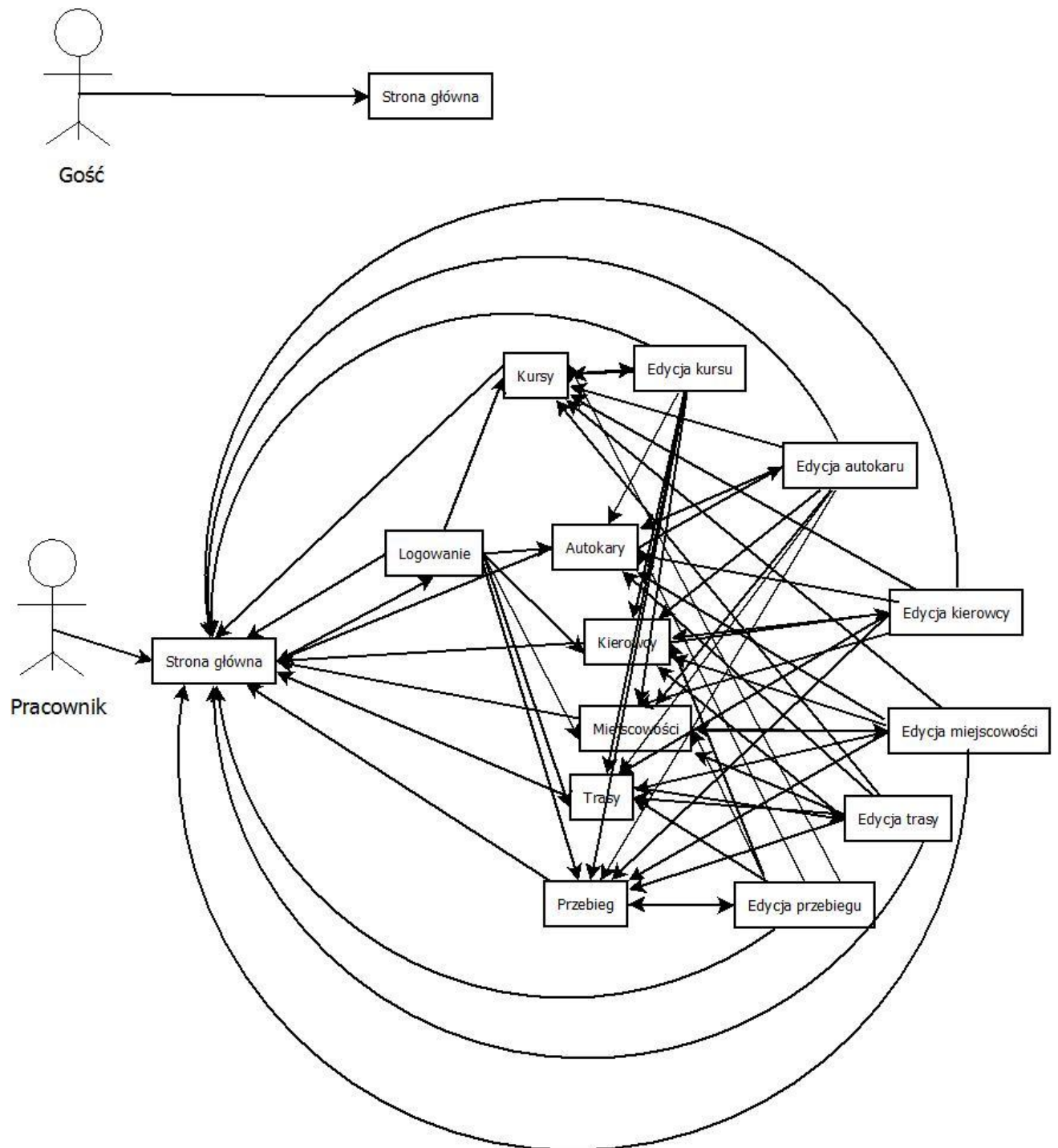
Aplikacja webowa udostępnia następujące funkcjonalności:

- Przeglądanie połączeń autokarów, po uprzednim podaniu dodatkowych parametrów;
- Administrowanie autokarami (usuwanie, edycja, dodawanie) przez upoważnionego pracownika;
- Wyszukiwanie przypisanych kierowców do wybranego autokaru;
- Administrowanie kierowcami (usuwanie, edycja, dodawanie) przez upoważnionego pracownika;
- Wyszukiwanie kierowców, którzy są w aktualnej chwili w trasie;
- Administrowanie trasami (edycja, usuwanie) przez upoważnionego pracownika;
- Administrowanie miejscowościami (edycja, usuwanie) przez upoważnionego pracownika;
- Administrowanie przebiegami pomiędzy przystankami (edycja, usuwanie) przez upoważnionego pracownika;
- Administrowanie kursami (edycja, usuwanie) przez upoważnionego pracownika;
- Generowanie raportu tygodniowego związanego z liczbą pasażerów danego kursu;
- Generowanie raportu liczącego średnią liczbę pasażerów od początku istnienia danego kursu

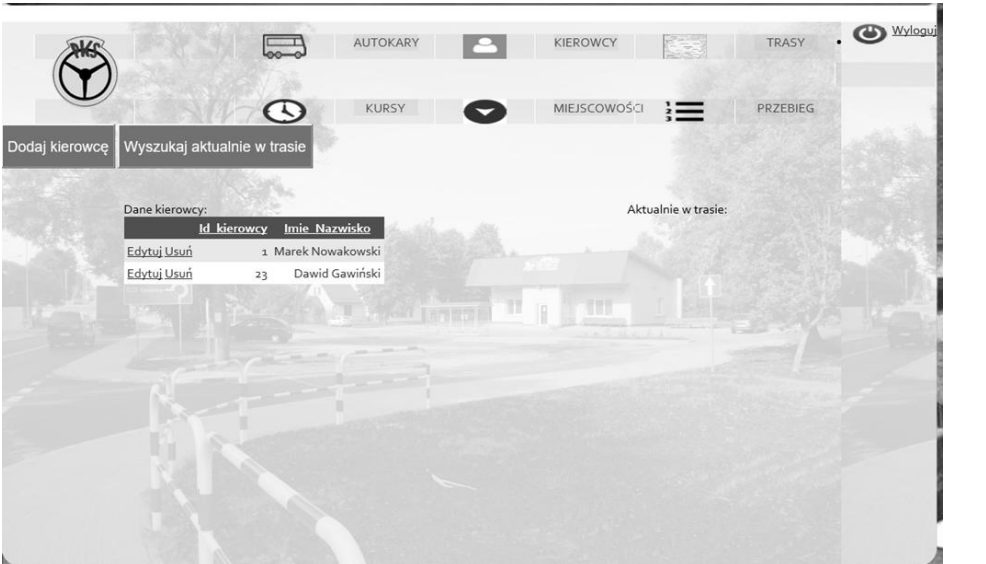
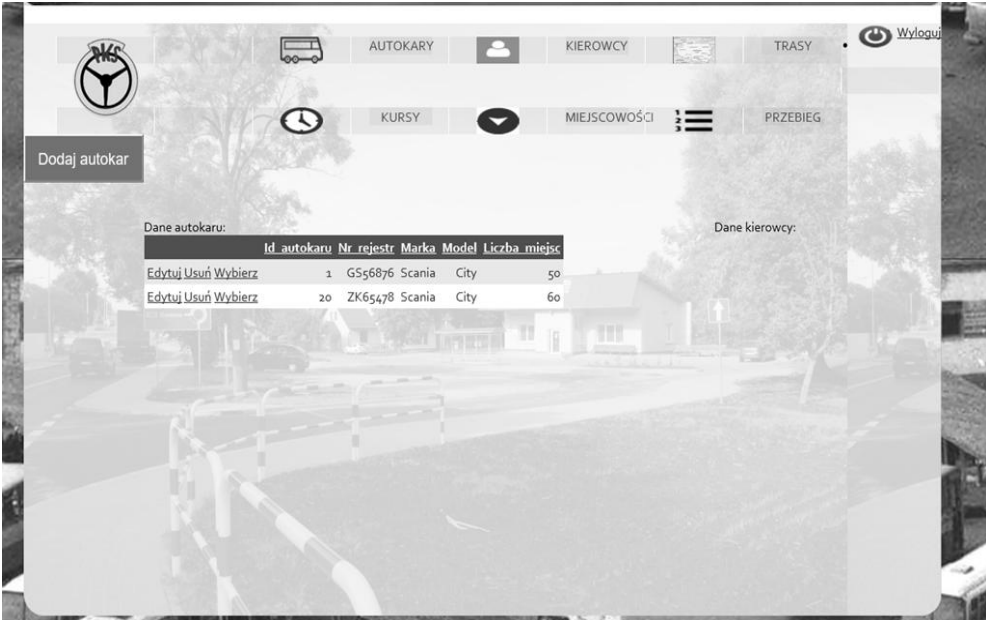
b) Wybrana technologia

Aplikacja została zrealizowana w asp.net WebForms przy użyciu środowiska Visual Studio 2015

c) Graf przejść



d) Opis struktury programu





AUTOKARY



KIEROWCY



TRASY

Wyloguj



KURSY



MIEJSCOWOŚCI



PRZEBIEG

Numer rejestracyjny:

Marka:

Model:

Ilość miejsc:

Id kursu:

Potwierdź

10. Wnioski

Poprawne zaprojektowanie bazy danych umożliwiło mi w sposób łatwy napisanie aplikacji lokalnej i internetowej.

Moja baza danych została postawiona na serwerze Microsoft SQL Server 2014. Wcześniej poznaliśmy Sybase, który według mnie jest o wiele gorszy. Microsoft SQL Server pozwala nam w prostszy sposób zarządzać danymi i je przeglądać.

Program PowerDesigner w bardzo łatwy sposób pozwala nam zaprojektować model konceptualny naszej bazy oraz pozwala wygenerować nam z niego model fizyczny i gotowy skrypt SQL. W przypadku Sybase skrypt nie zawsze wykonywany jest w 100%, natomiast w przypadku Microsoft SQL Server skrypt wykonywany jest poprawnie, a więc ułatwia nam to pracę.

Aplikacja lokalna napisana w języku C# do poprawnego działania potrzebowała dołączenia odpowiedniej biblioteki do połączenia z bazą danych. Prace przy pisaniu aplikacji przebiegały bez problemu, wcześniej napisane kwerendy działały poprawnie.

Technologia ASP.NET jest bardzo dobrą technologią do tworzenia aplikacji internetowych. Pozwala na bardzo proste zbudowanie szkieletu strony oraz zarządzanie użytkownikami zalogowanymi i niezalogowanymi. Technologia posiada także role, którymi możemy nadawać odpowiednie uprawnienia danym użytkownikom.

Aplikacja internetowa napisana w technologii asp.net WebForms była znacznie łatwiejsza do zaimplementowania niż ta w C#. Poprawne napisanie kontrolerów umożliwiło mi wygenerowanie potrzebnych widoków, takich jak okna dodawania nowych wierszy do bazy czy wyświetlania encji. Dzięki narzuconemu z góry frameworkowi kod był znacznie lepiej zorganizowany, każdą funkcjonalność można o wiele szybciej odszukać niż tą w C#. Utworzenie aplikacji w ASP.NET przy użyciu WebForms sprawiło, że wszelkie potrzebne biblioteki do połączenia z bazą danych (entity framework) zostały zainstalowane automatycznie przy tworzeniu projektu.

Według mojej oceny napisanie aplikacji w ASP.NET było znacznie łatwiejsze niż napisanie jej w C#. Język ten jest zdecydowanie bardziej intuicyjny, a zaimplementowane w nim rozwiązania sprawiają, że niektóre rzeczy można wykonać szybciej, lepiej i łatwiej.