Maciej Karpiński, Tomasz Jakubiak

# Dokumentacja Techniczna

## Programu komputerowego – Komis Samochodowy „Janusz”

W skład niniejszej dokumentacji wchodzą:

### Cel oraz założenia projektu

### Diagram UML – przypadki użycia

### Model danych

### Opis przypadków użycia

### model interfejsu użytkownika

### Diagram UML Sekwencji

### Fragmenty kodów źródłowych

### Podsumowanie

### Cel oraz założenia projektu

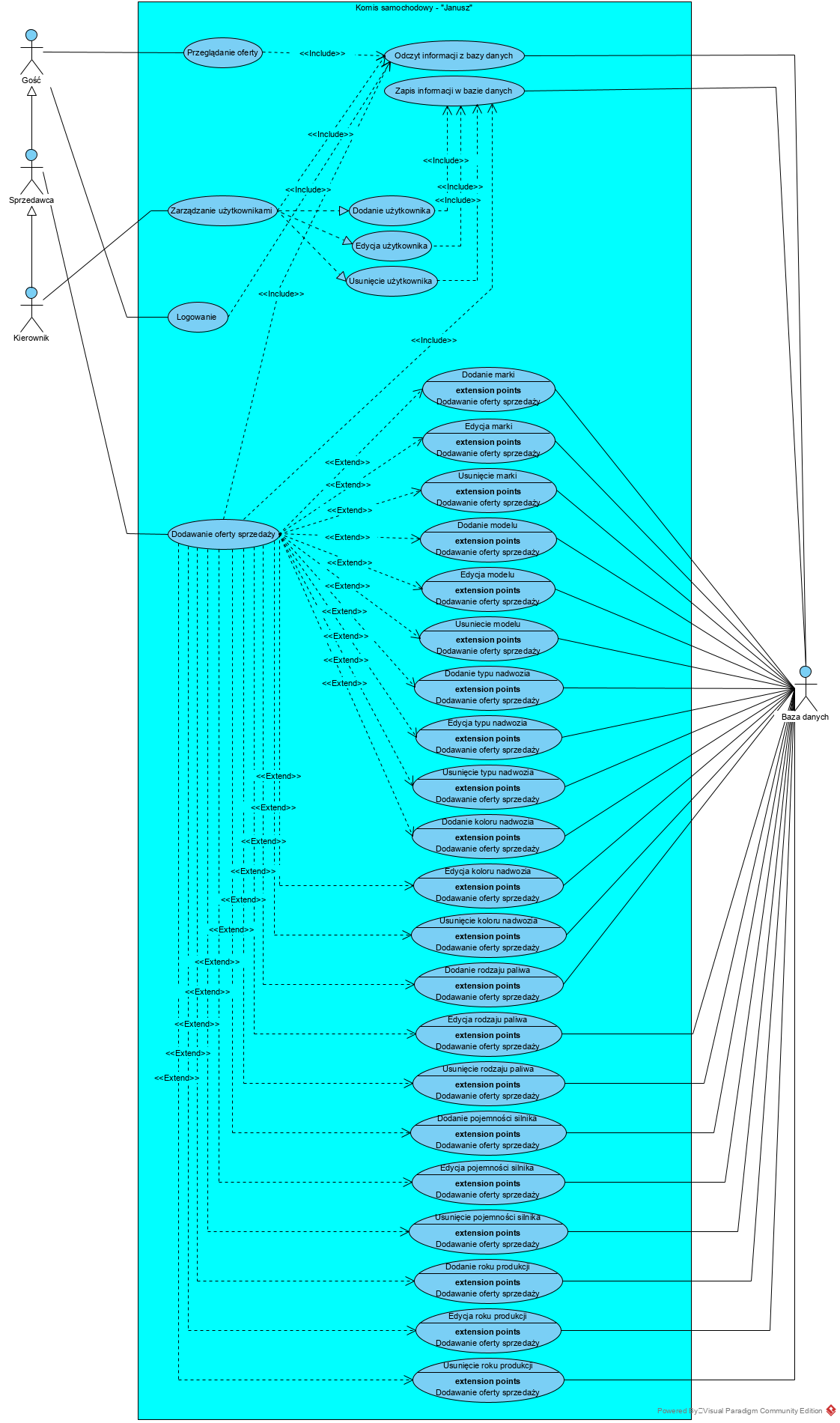
Celem stworzenia projektu jest zautomatyzowanie dodawania, przechowywania oraz wyświetlania ogłoszeń w komisie samochodowym „Janusz”.

Aplikacja będzie bardzo intuicyjna i przyjazna użytkownikowi,

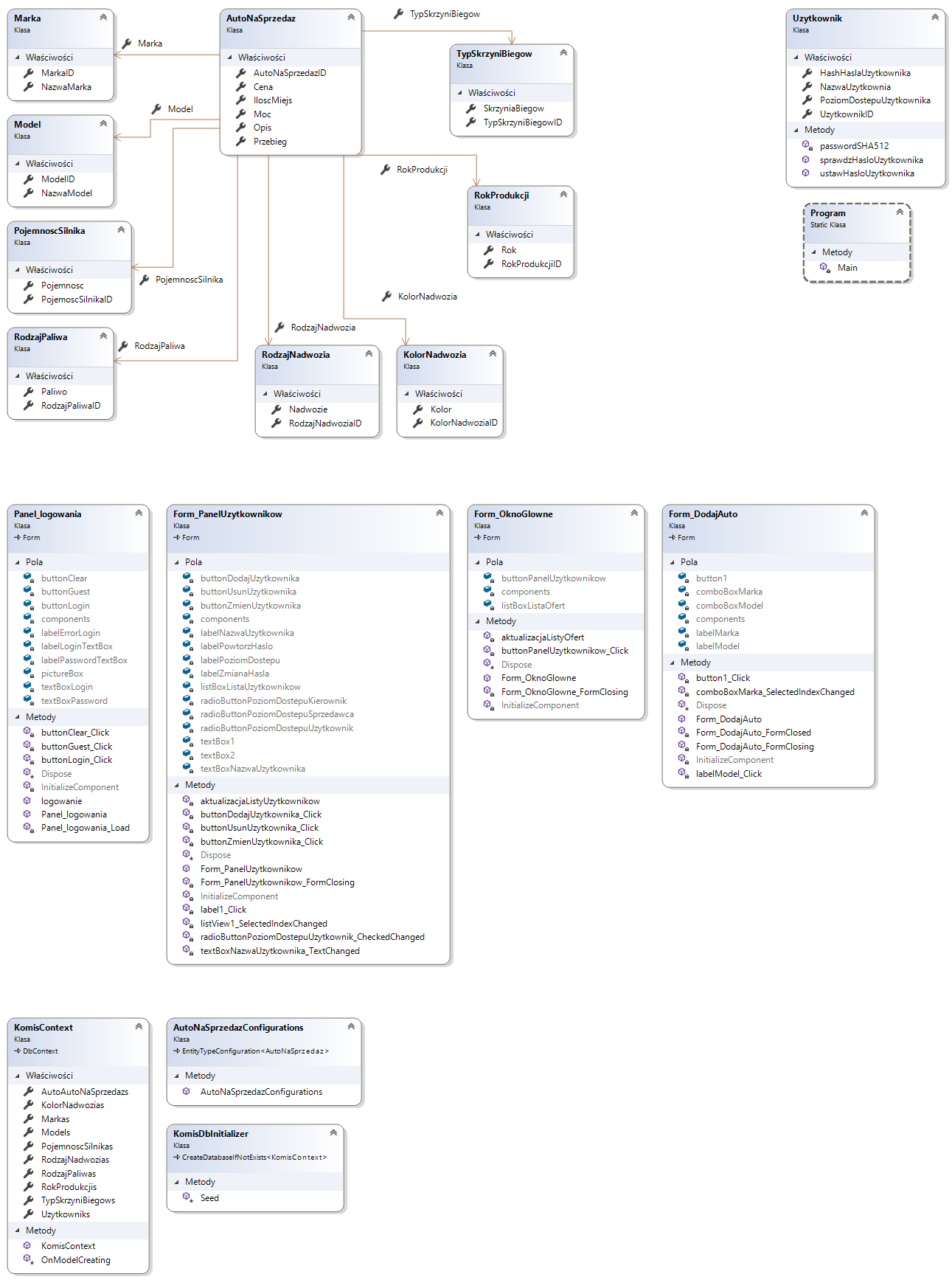
Dane będą zapisywane do bazy SQL lokalnej,

Dodawanie ogłoszeń będzie uproszczone poprzez wybieranie z gotowych danych wprowadzonych do bazy, ale użytkownik będzie miał możliwość dodawania własnych danych.

### Diagram UML – przypadki użycia



### Model danych



### Opis przypadków użycia

Proces logowania się użytkowników z różnymi uprawnieniami i możliwością korzystania z wybranych formularzy.

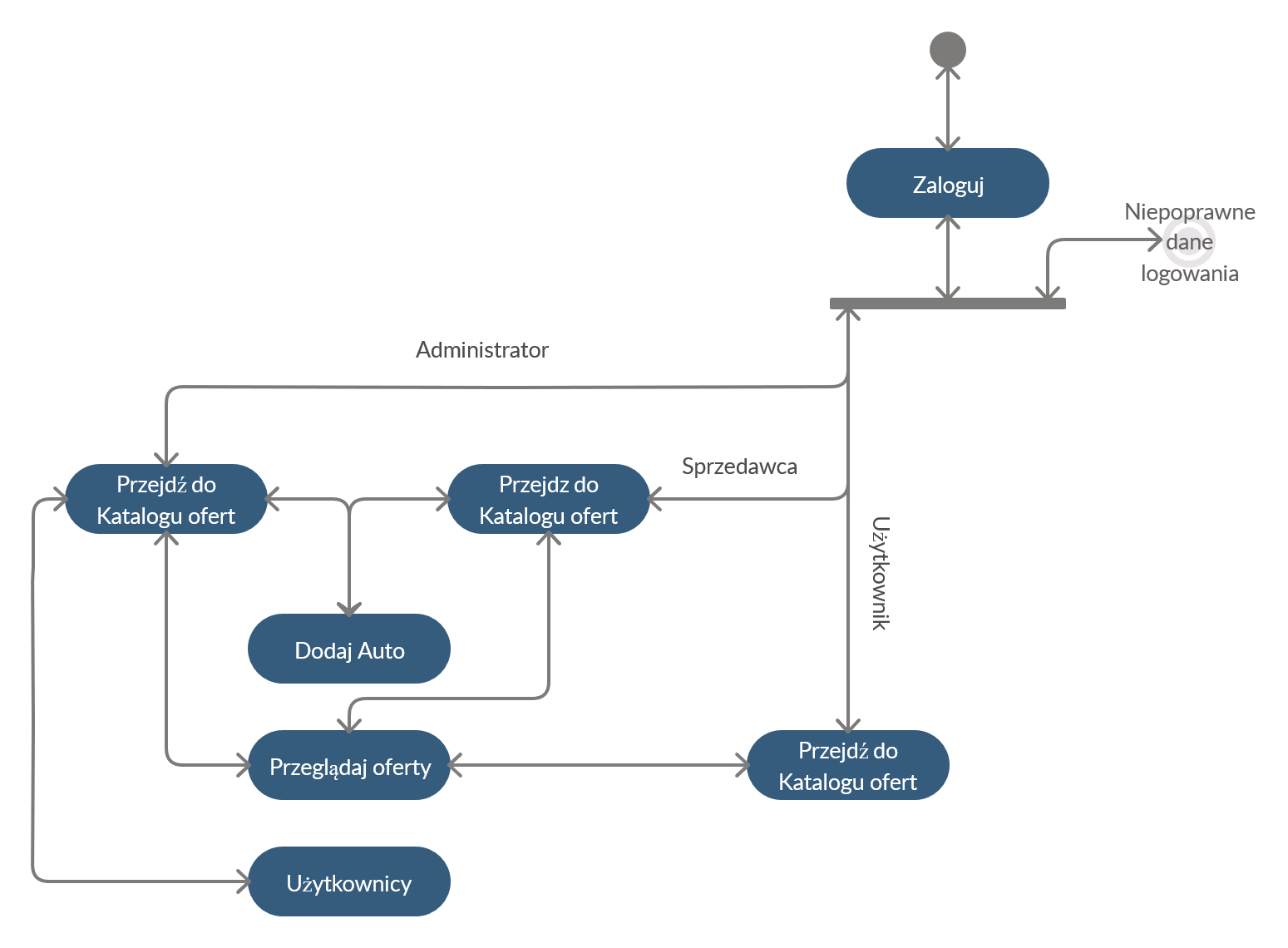
Program na początku posiada dwa konta : konto Administratora z pełnymi uprawnieniami oraz konto Gościa z uprawnieniami użytkownika.

Program posiada 3 rodzaje uprawnień:

Administrator – konto z najwyższymi uprawnieniami ma dostęp do wszystkich formularzy : Przeglądanie katalogu ofert , obsługa ofert oraz obsługa kont w systemie.

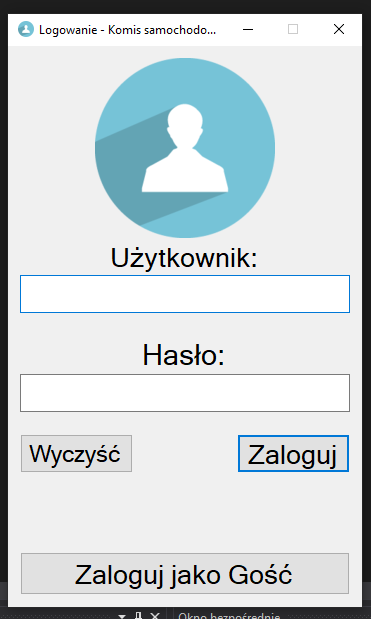
Sprzedawca – konto posiadające dostęp do dwóch formularzy : Przeglądanie katalogu ofert oraz obsługa ofert.

Użytkownik – konto posiadające tylko możliwość przeglądania ofert znajdujących się w systemie



### model interfejsu użytkownika

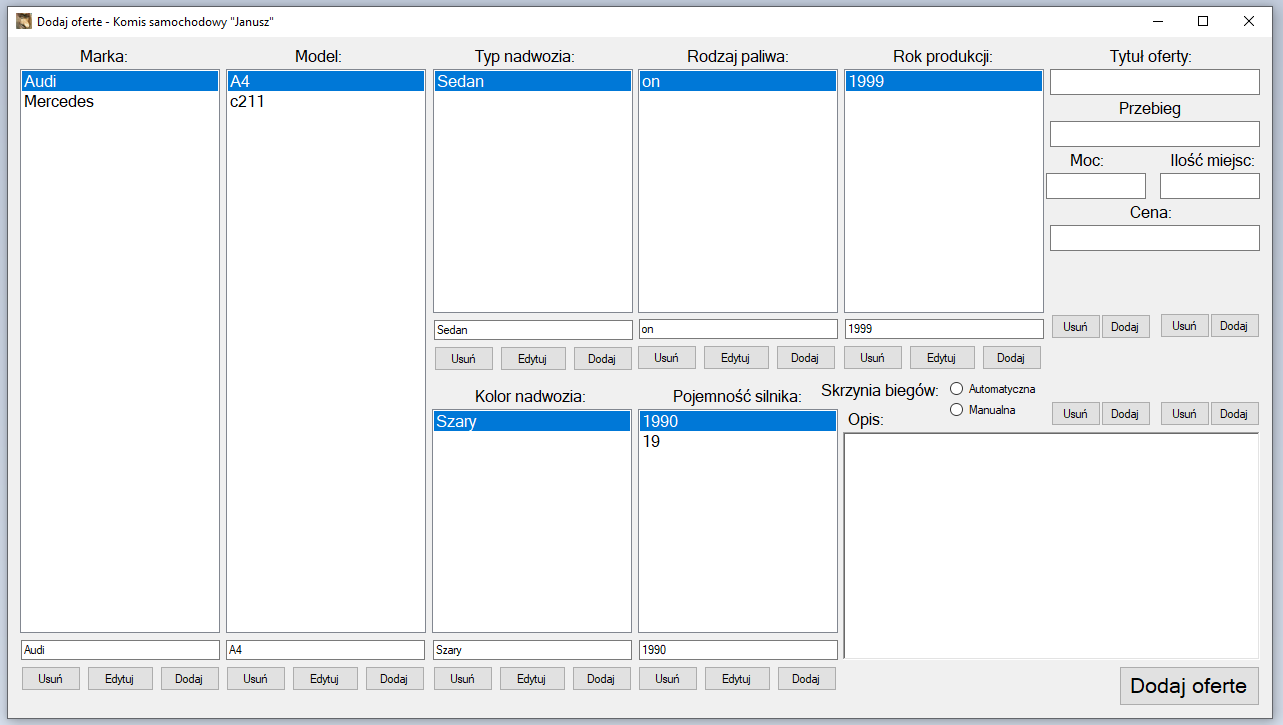
Panel logowania



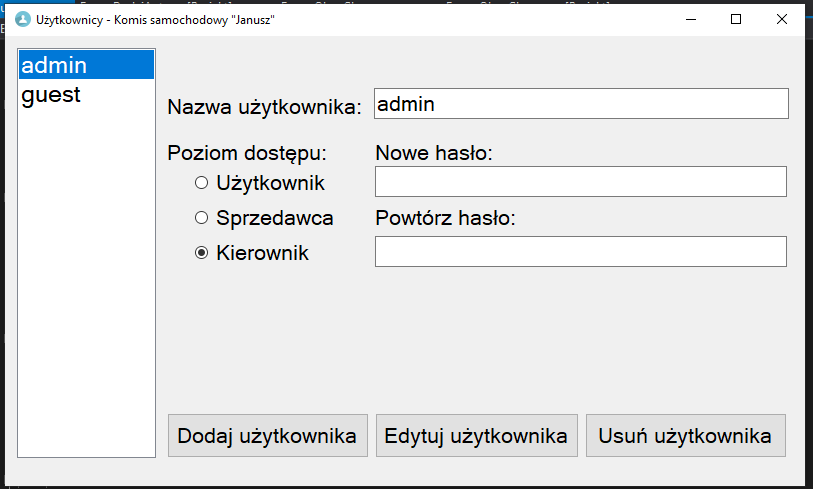
Panel ogłoszeń



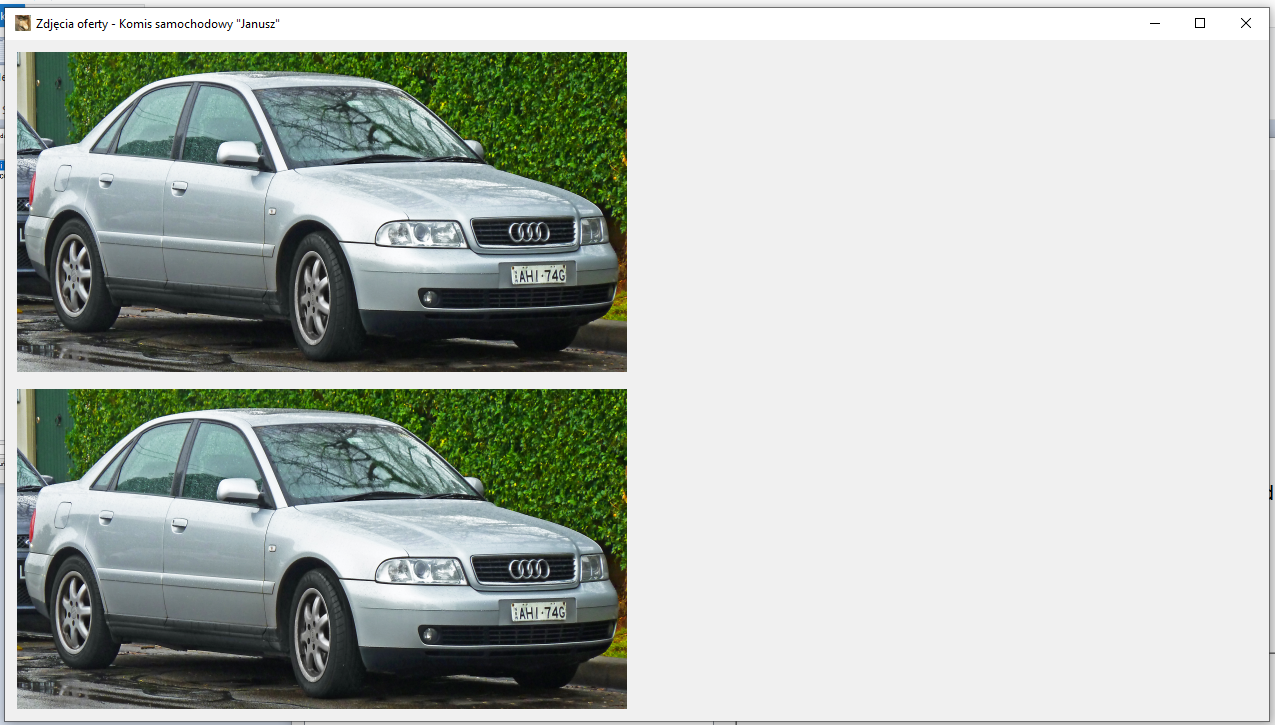
Panel dodawania nowego ogłoszenia



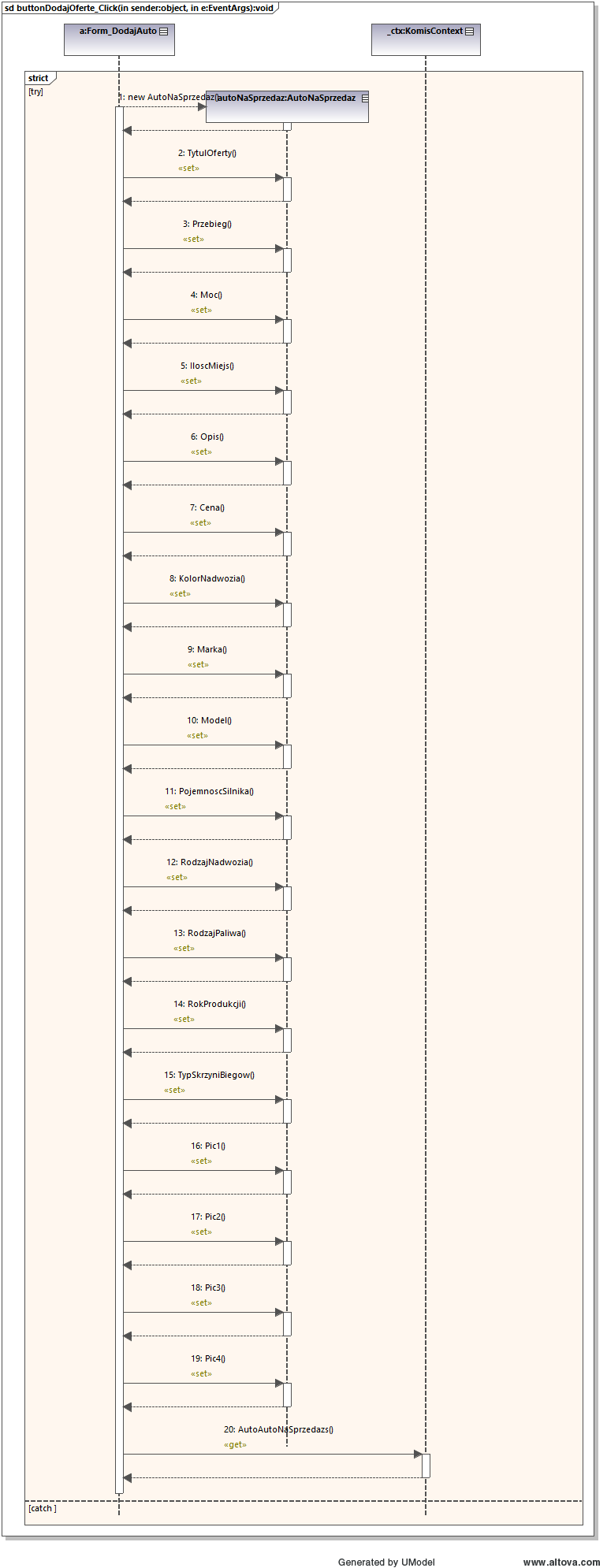
Panel zarządzania użytkownikami



Panel przeglądania zdjęcie w ogłoszeniu



### Diagram UML Sekwencji



Aby odczytać wszystkie Diagramy sekwencji UML otwórz plik „UML Sekwencje” w folderze aplikacji przy użyciu programu Altova UModel.

### Fragmenty kodów źródłowych

1.

bool adminExist = false, guestExist = false;

using (var ctx = new KomisContext())

{

//Sprawdzanie czy w bazie danych istnieją domyślni użytkownicy admin i guest

List<Uzytkownik> listaUzytkownikow = ctx.Uzytkowniks.ToList();

foreach (var VARIABLE in listaUzytkownikow)

{

if (VARIABLE.NazwaUzytkownia == "admin")

{

adminExist = true;

break;

}

}

foreach (var VARIABLE in listaUzytkownikow)

{

if (VARIABLE.NazwaUzytkownia == "guest")

{

guestExist = true;

break;

}

}

//W przypadku braku domyślnych użytkowników admin i guest dodaje ich do bazy danych

if (adminExist == false)

{

var admin = new Uzytkownik();

admin.NazwaUzytkownia = "admin";

admin.ustawHasloUzytkownika("admin");

admin.PoziomDostepuUzytkownika = 2;

ctx.Uzytkowniks.Add(admin);

ctx.SaveChanges();

}

if (guestExist == false)

{

var guest = new Uzytkownik();

guest.NazwaUzytkownia = "guest";

guest.ustawHasloUzytkownika("guest");

guest.PoziomDostepuUzytkownika = 0;

ctx.Uzytkowniks.Add(guest);

ctx.SaveChanges();

}

}

2.

private void Form\_OknoGlowne\_Load(object sender, EventArgs e)

{

Debug.WriteLine("Okno główne, przyjęty parametr: " + PoziomLogowania);

//przyjecie wartości z panelu logowania

switch (PoziomLogowania)

{

case 0:

Debug.WriteLine("switch 0");

break;

case 1:

Debug.WriteLine("switch 1");

buttonDodajAuto.Visible = true;

buttonUsun.Visible = true;

break;

case 2:

Debug.WriteLine("switch 2");

buttonPanelUzytkownikow.Visible = true;

buttonDodajAuto.Visible = true;

buttonUsun.Visible = true;

break;

}

}

3.

public class Marka

{

[Key]

public int MarkaID { get; set; }

public string NazwaMarka { get; set; }

}

4.

public class Uzytkownik

{

[Key]

public int UzytkownikID { get; set; }

public string NazwaUzytkownia { get; set; }

public string HashHaslaUzytkownika { get; private set; }

public int PoziomDostepuUzytkownika { get; set; }

//ustawia hasło użytkownika jako hash

public void ustawHasloUzytkownika(string haslo)

{

this.HashHaslaUzytkownika = passwordSHA512(haslo);

}

//porównuje hasło wporawdzone z hasłem użytkownika

public bool sprawdzHasloUzytkownika(string haslo)

{

if (HashHaslaUzytkownika == passwordSHA512(haslo))

{

return true;

}

return false;

}

//szyfrowanie hasła przy pomocy SHA512

private string passwordSHA512(string haslo)

{

var bytes = System.Text.Encoding.UTF8.GetBytes(haslo);

using (var hash = System.Security.Cryptography.SHA512.Create())

{

var hashedInputBytes = hash.ComputeHash(bytes);

var hashedInputStringBuilder = new System.Text.StringBuilder(128);

foreach (var b in hashedInputBytes)

{

hashedInputStringBuilder.Append(b.ToString("X2"));

}

return hashedInputStringBuilder.ToString();

}

}

}

### Podsumowanie

Program został wykonany w Visual Studio 2019, przy użyciu Microsoft SQL Server.

Wszystkie założenia zostały spełnione zgodnie z wytycznymi.