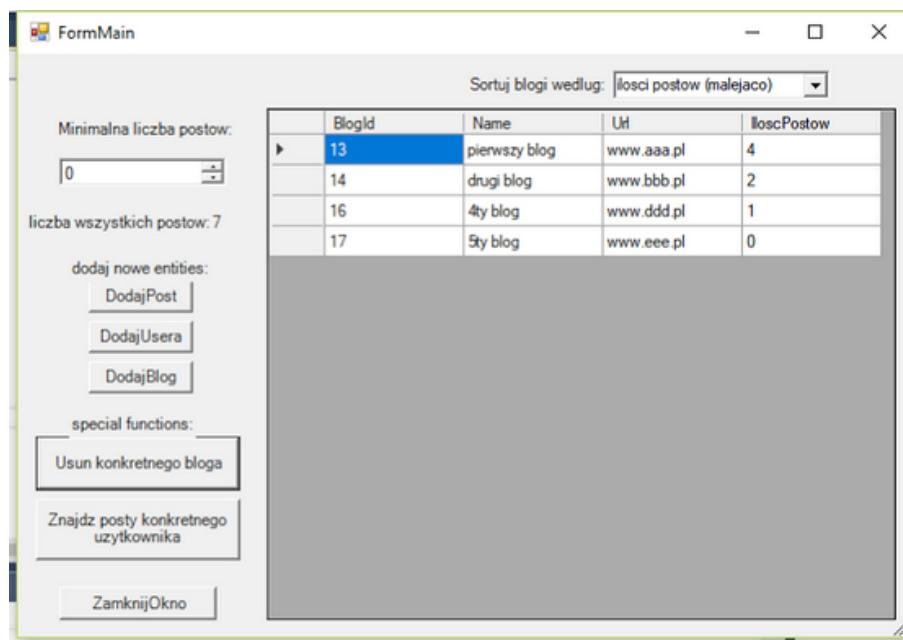


# BAZY DANYCH SPRAWOZDANIE .Net, Entity Framework

Ogolny wyglad programu (glowny form):



Program spełnia funkcje:

- 1) dodawania nowych entities ( posta/usera/bloga )
- 2) glowny form pokazuje wszystkie blogi, ktore mozemy posortowac
- 3) usuniecie konkretnego bloga
- 4) glowny form pokazuje wszystkie blogi, dropdownowi “minimalna liczba postow” mozemy pokazac tylko te, ktore maja nie mniej postow niz wskazana przez nas liczba.
- 5) Znajdowanie postow konkretnego uzytkownika

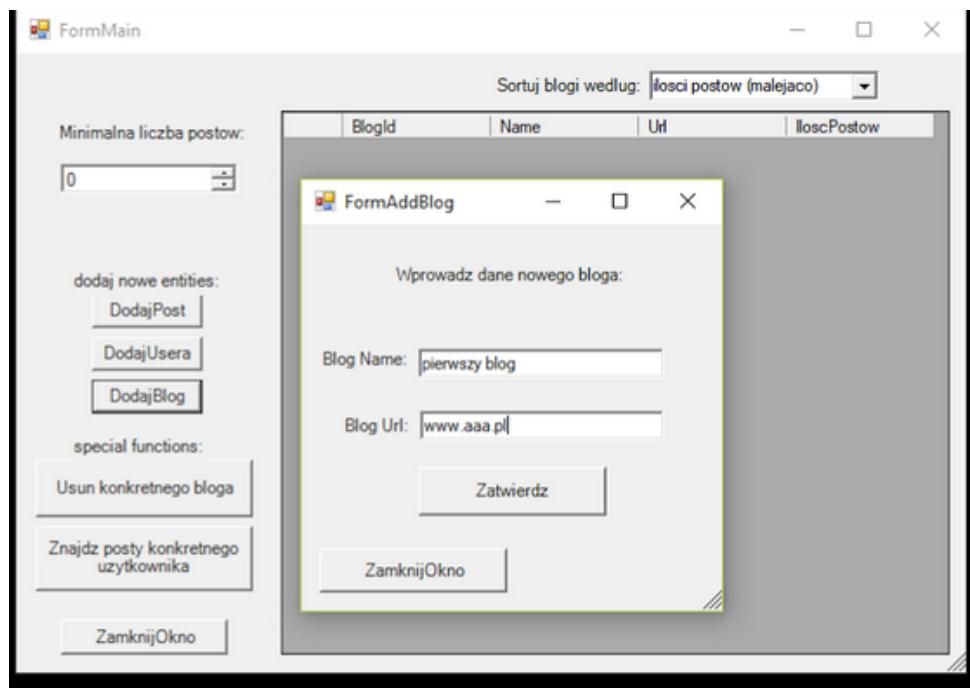
**(pierwsze 15 stron to screny samego programu oraz opis jego dzialania, po nich jest widoczny kod programu).**

## 1) dodawanie nowych entities

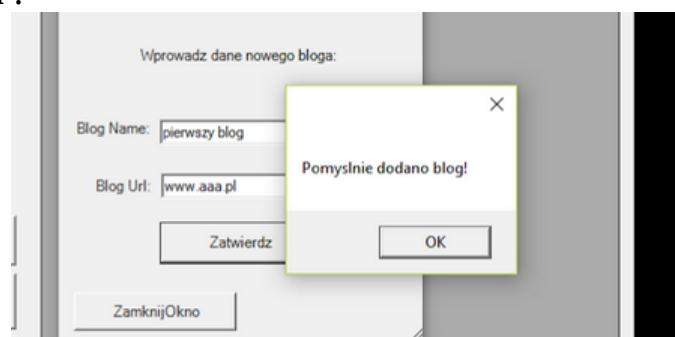
-po dodaniu nowego bloga/posta wszystkie bogi sa automatycznie odswiezane w naszym glownym widoku wiec widzimy zmiany bez potrzeby jakiegokolwiek odswiezania naszego programu.

### 1a) dodawanie nowego bloga

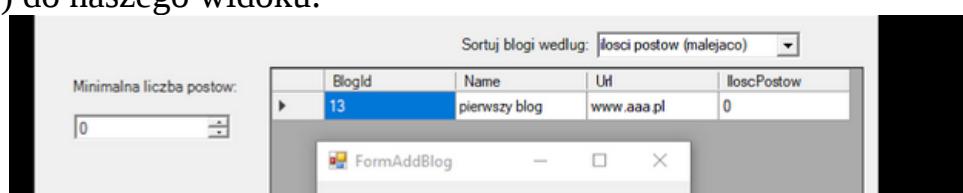
-Po wcisnieciu buttona “DodajBlog” widocznego na glownej formie wyświetla sie nam forma odpowiedzialna za dodawanie nowego bloga:



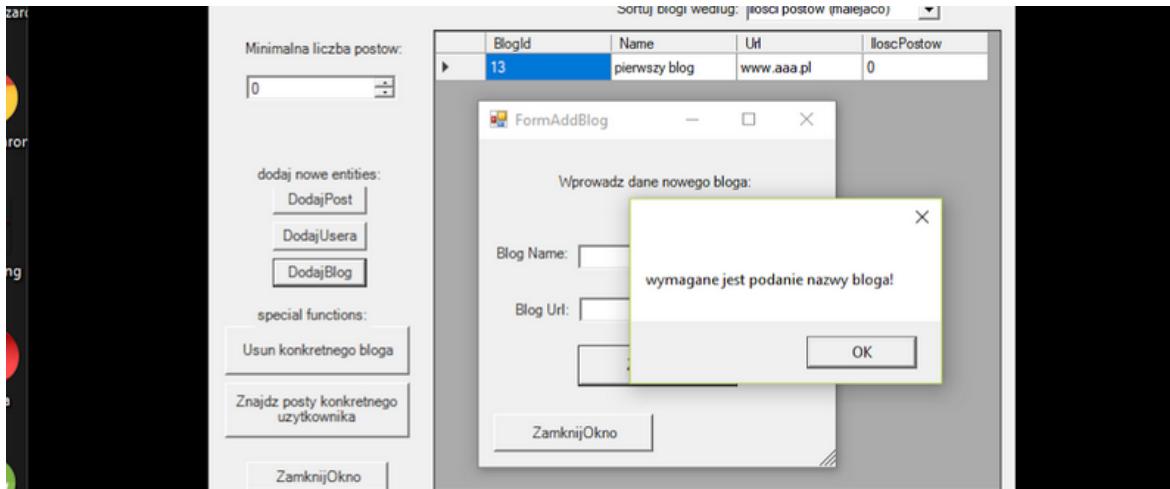
-po wypelnieniu wymaganych pol wyświetli nam sie okno dialogowe informujace nas o dodaniu bloga :



-I od razu mozemy zauwazyc dodanie nowego bloga (z danymi, ktore wczesniej podalismy) do naszego widoku:

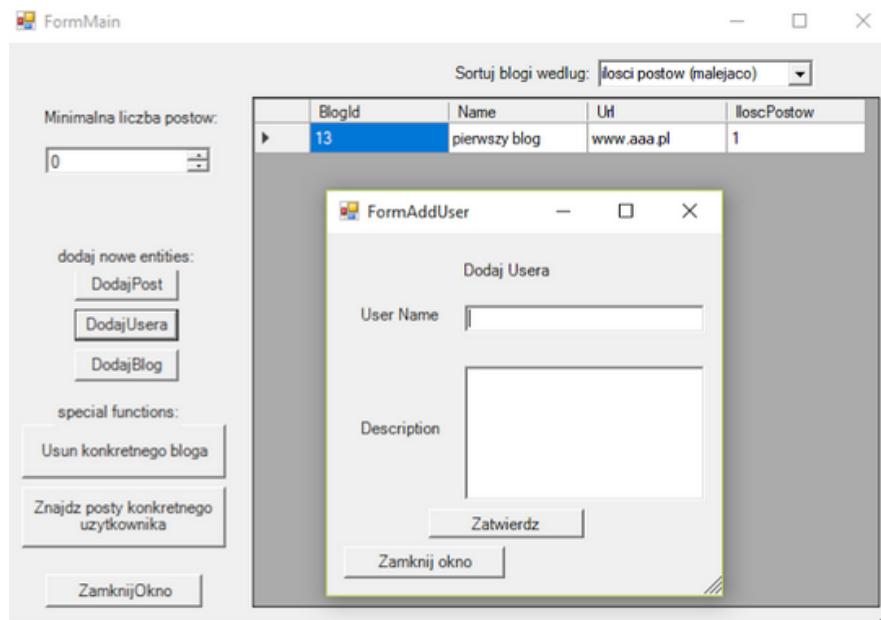


-przy dodawaniu bloga wymagana jest nazwa bloga (url nie jest wymagany), gdy sprobujemy dodac blog bez wprowadzenia nazwy, dostaniemy komunikat:



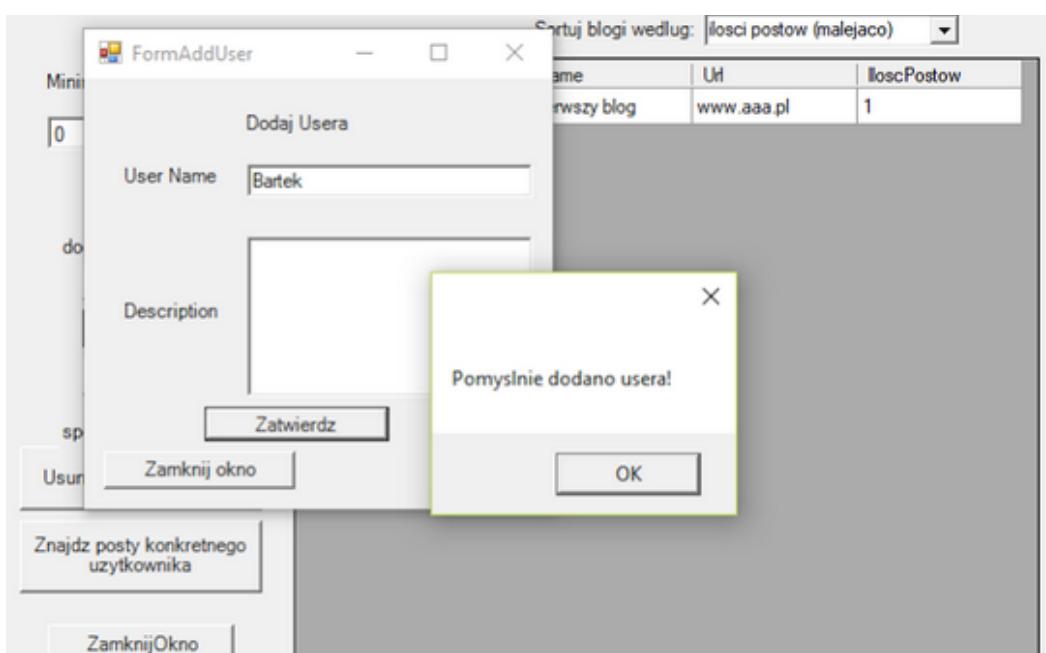
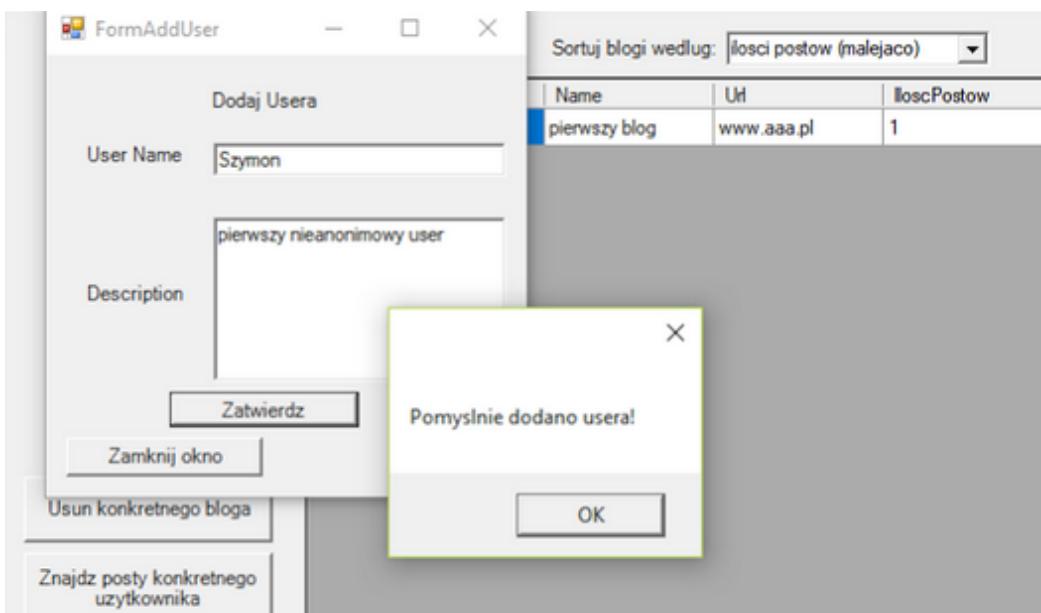
### 1b) dodawanie nowego usera

-Po wcisnieciu buttona “Dodaj Usera” widocznego na glownej formie wyświetla sie nam forma odpowiedzialna za dodawanie nowego usera:

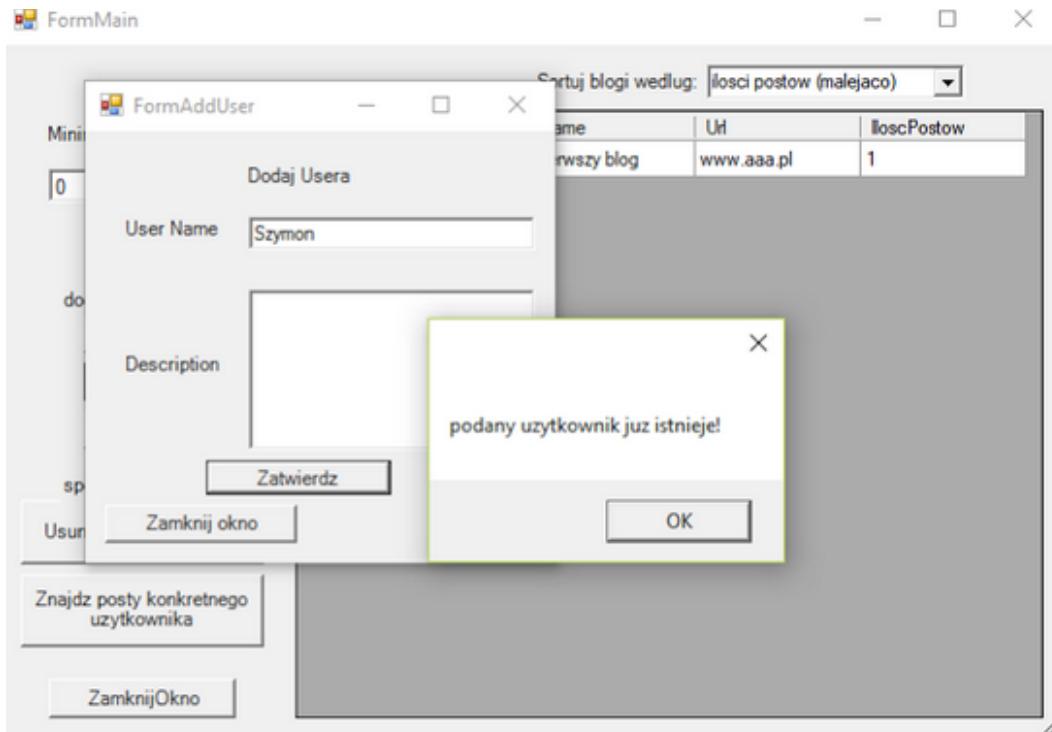


-Zalozylem, ze User Name (primary key) jest wymagany a pole Description nie jest.

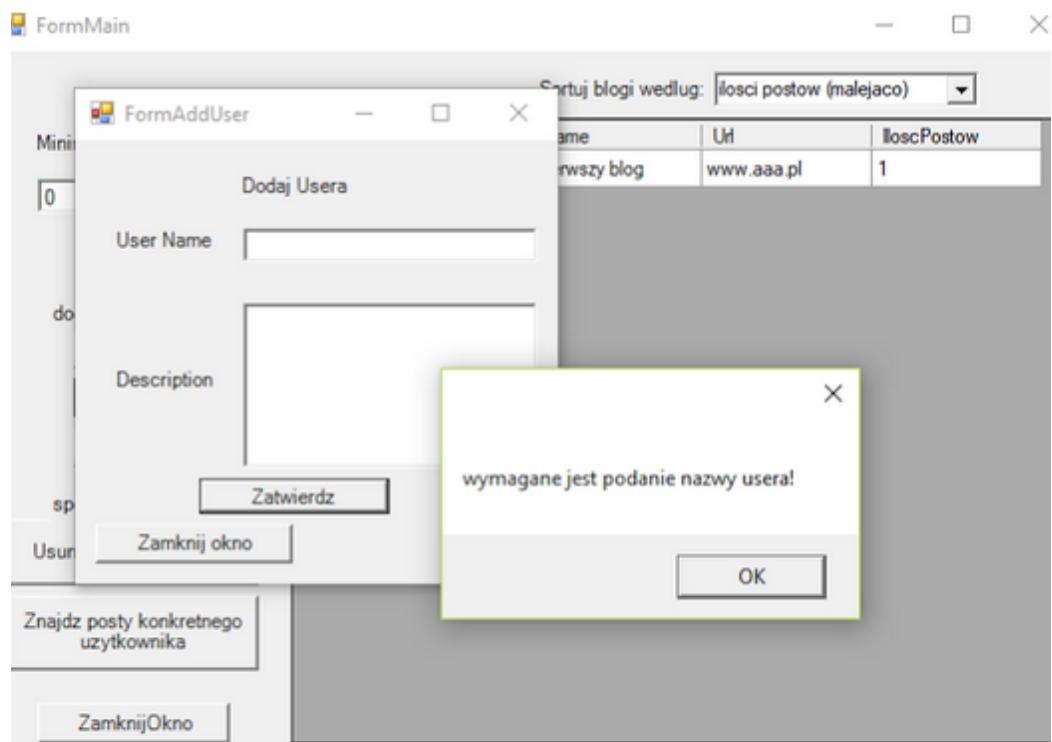
-Gdy wypełnimy wymagane pola dostaniemy komunikat :



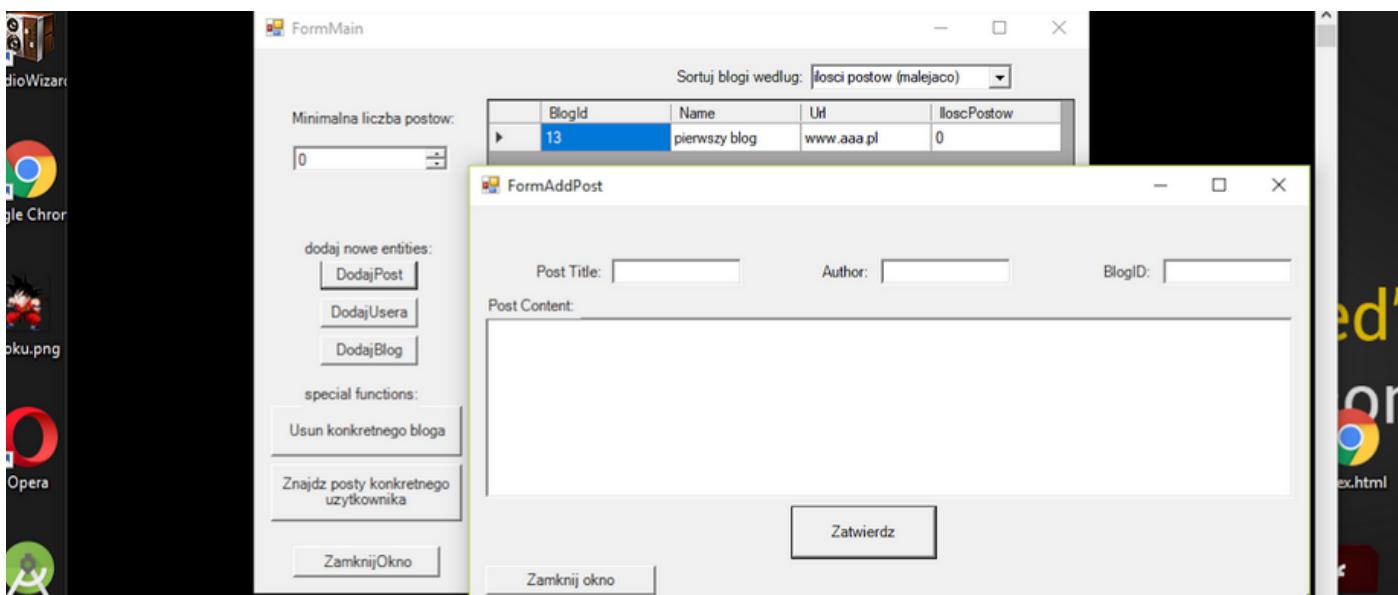
-Gdy ponownie wprowadzimy ten sam User Name (lub po prostu User Name, który już jest zajęty), dostaniemy komunikat :



-Gdy sprobujemy dodaj usera, bez wypełnionego wymaganego pola, dostaniemy komunikat :



## 1c) dodawanie nowego posta

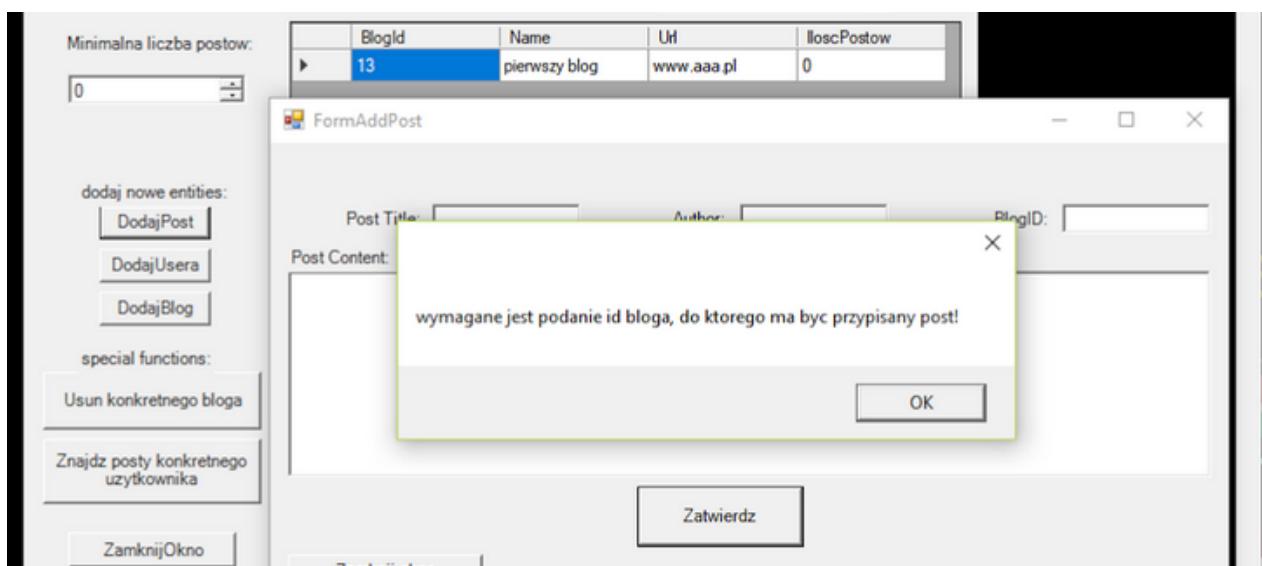


-Po wcisnieciu buttona “Dodaj Post” widocznego na glownej formie wyświetla sie nam forma odpowiedzialna za dodawanie nowego posta:

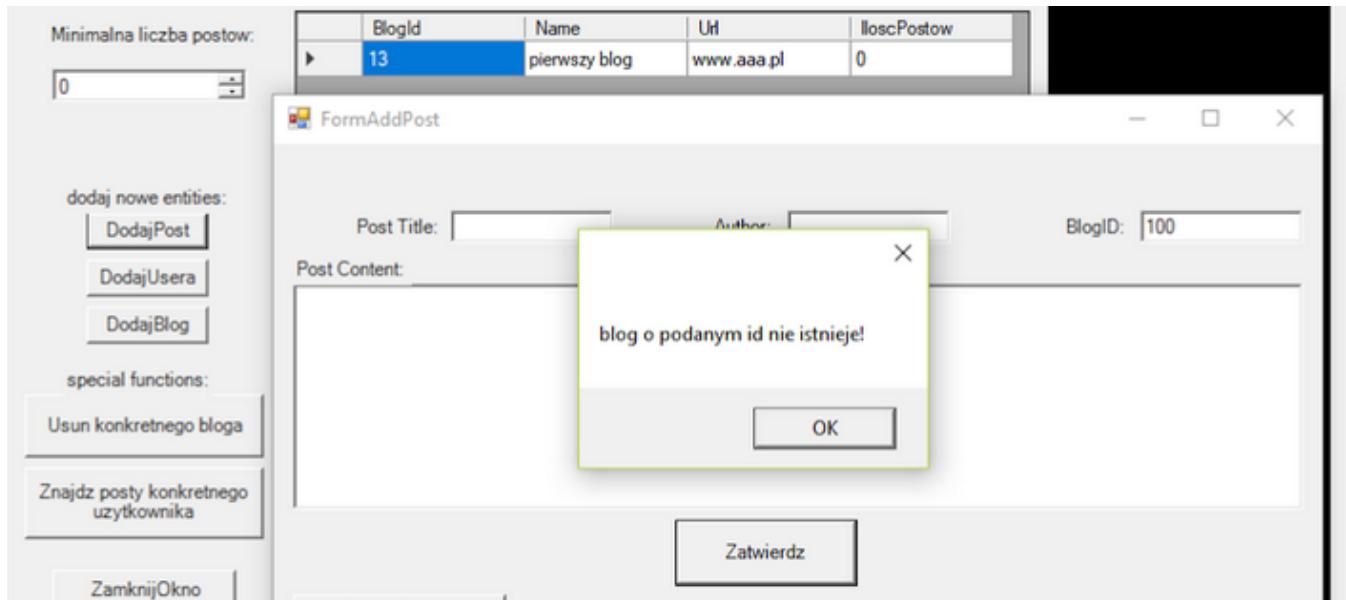
-przyjalem, ze wszystkie pola, poza Authorem, sa wymagane (gdy nie podamy authora to post dostanie wlasciciela (usera) o nicku “anonymous” I jego post bedzie anonimowy).

-dodawanie nowego posta odbywa sie po wypełnieniu odpowiednich pol I wcisnieciu przycisku “zatwierdz”

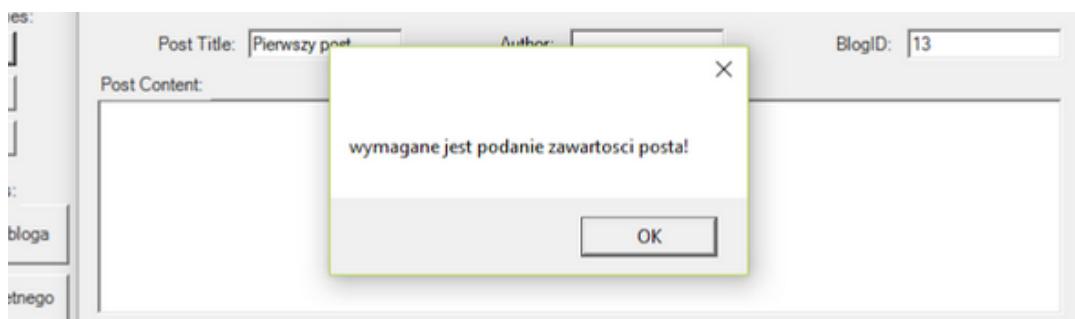
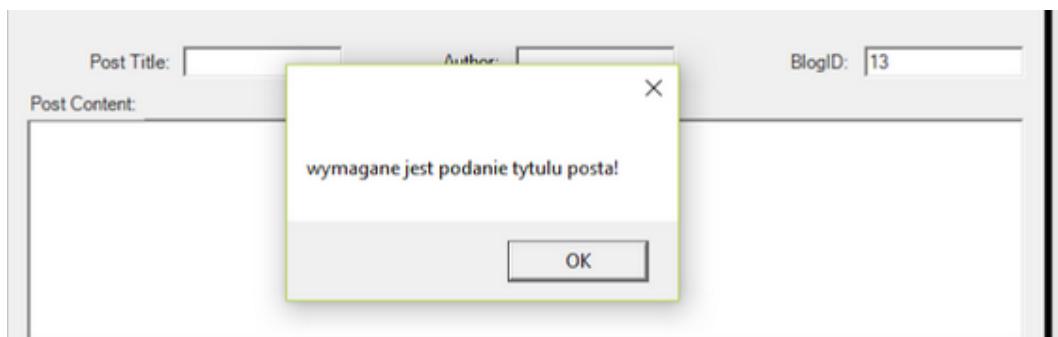
-gdy sprobujemy dodac post bez wprowadzonego id bloga, do którego ma post przynalezec, dostaniemy komunikat :



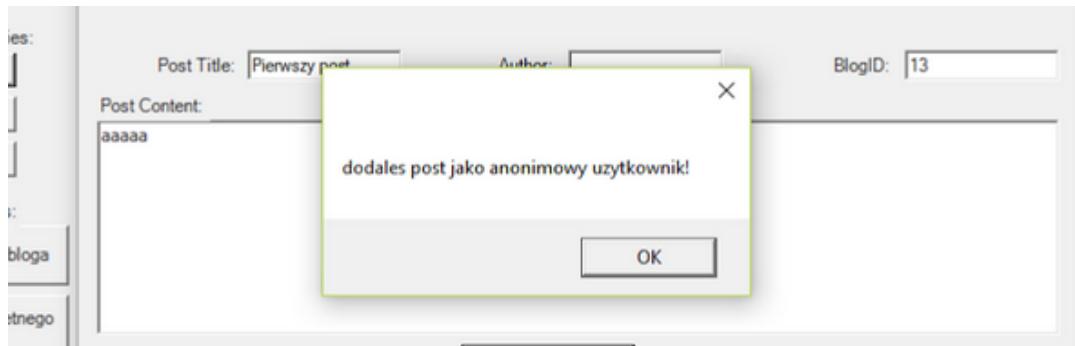
-natomiast gdy podamy nieistniejące id bloga dostaniemy komunikat :



-wymagane jest również podanie tytułu posta oraz kontentu posta ( komunikaty pojawiające się gdy ich nie podamy widoczne sa ponizej ).

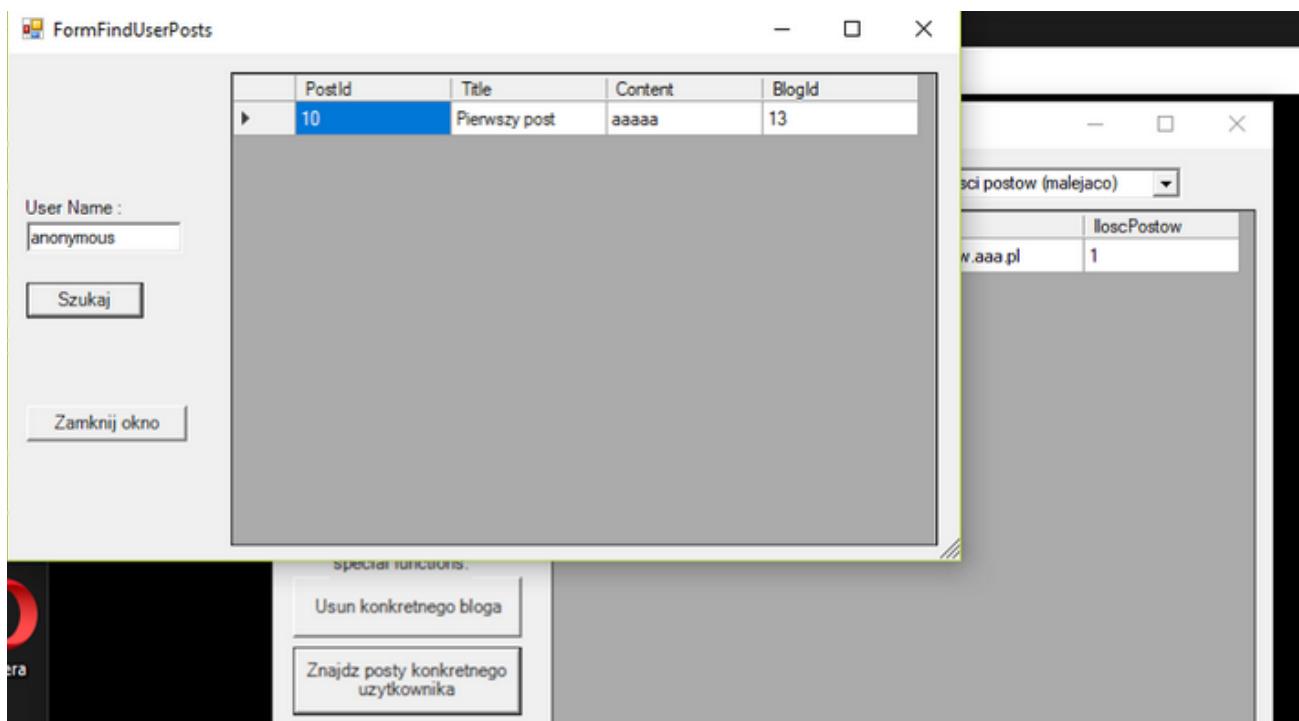


-gdy podamy wszystkie parametry, poza autorem, dostaniemy komunikat :



post został dodany.

Tutaj korzystajac z przycisku “znajdz posty konkretnego uzytkownika” (ktory opisze poznniej) mozemy zobaczyć, ze powyzszy post został prawidłowo przypisany do uzytkownika anonymous, mozemy również zauwazyc, ze w kolumnie “IloscPostow” dla naszego (jak na razie jedynego) bloga widnieje jedynka (post, który właśnie dodaliśmy):



## 2) sortowanie blogow

-w prawym gornym rogu dodalem mozliwosc sortowania blogow ze wzgledu na :

- a) blogID
- b) url
- c) nazwe bloga
- d) ilosc postow

- Po kazdej zmienie sposobu sortowania, blogi sa automatycznie sortowane wedlug wybranego sposobu sortowania.

- Pokaz praktyczny :

(sortowanie wedlug blogID (rosnaco) jest defaultowym sortowaniem)

Sortuj blogi wedlug: blogID (rosnaco)				
	BlogId	Name	Url	IloscPostow
▶	13	pierwszy blog	www.aaa.pl	4
	14	drugi blog	www.bbb.pl	2
	15	trzeci blog	www.ccc.pl	1
	16	4ty blog	www.ddd.pl	1
	17	5ty blog	www.eee.pl	0

Sortuj blogi wedlug: blogID (malejaco)				
	BlogId	Name	Url	IloscPostow
▶	17	5ty blog	www.eee.pl	0
	16	4ty blog	www.ddd.pl	1
	15	trzeci blog	www.ccc.pl	1
	14	drugi blog	www.bbb.pl	2
	13	pierwszy blog	www.aaa.pl	4

Sortuj blogi wedlug: nazwa bloga (alfabetycznie)				
	BlogId	Name	Url	IloscPostow
▶	16	4ty blog	www.ddd.pl	1
	17	5ty blog	www.eee.pl	0
	14	drugi blog	www.bbb.pl	2
	13	pierwszy blog	www.aaa.pl	4
	15	trzeci blog	www.ccc.pl	1

Sortuj blogi wedlug: nazwa bloga (z-a)				
	BlogId	Name	Url	IloscPostow
▶	13	pierwszy blog	www.aaa.pl	4
	14	drugi blog	www.bbb.pl	2
	15	trzeci blog	www.ccc.pl	1
	16	4ty blog	www.ddd.pl	1
	17	5ty blog	www.eee.pl	0

Sortuj blogi wedlug: url bloga (alfabetycznie)				
	BlogId	Name	Url	IloscPostow
▶	13	pierwszy blog	www.aaa.pl	4
	14	drugi blog	www.bbb.pl	2
	15	trzeci blog	www.ccc.pl	1
	16	4ty blog	www.ddd.pl	1
	17	5ty blog	www.eee.pl	0

Sortuj blogi według: url bloga (z-a)

	BlogId	Name	Url	IloscPostow
▶	17	5ty blog	www.eee.pl	0
	16	4ty blog	www.ddd.pl	1
	15	trzeci blog	www.ccc.pl	1
	14	drugi blog	www.bbb.pl	2
	13	pierwszy blog	www.aaa.pl	4

Sortuj blogi według: Ilosci postow (malejaco)

	BlogId	Name	Url	IloscPostow
▶	13	pierwszy blog	www.aaa.pl	4
	14	drugi blog	www.bbb.pl	2
	15	trzeci blog	www.ccc.pl	1
	16	4ty blog	www.ddd.pl	1
	17	5ty blog	www.eee.pl	0

Sortuj blogi według: Ilosci postow (rosnaco)

	BlogId	Name	Url	IloscPostow
▶	17	5ty blog	www.eee.pl	0
	15	trzeci blog	www.ccc.pl	1
	16	4ty blog	www.ddd.pl	1
	14	drugi blog	www.bbb.pl	2
	13	pierwszy blog	www.aaa.pl	4

### 3) usuwanie konkretnego bloga

-Na potrzeby tego podpunktu dodalem label, ktory pokazuje liczbe wszystkich postow we wszystkich blogach (pod dropdownem “minimalna liczba postow”).

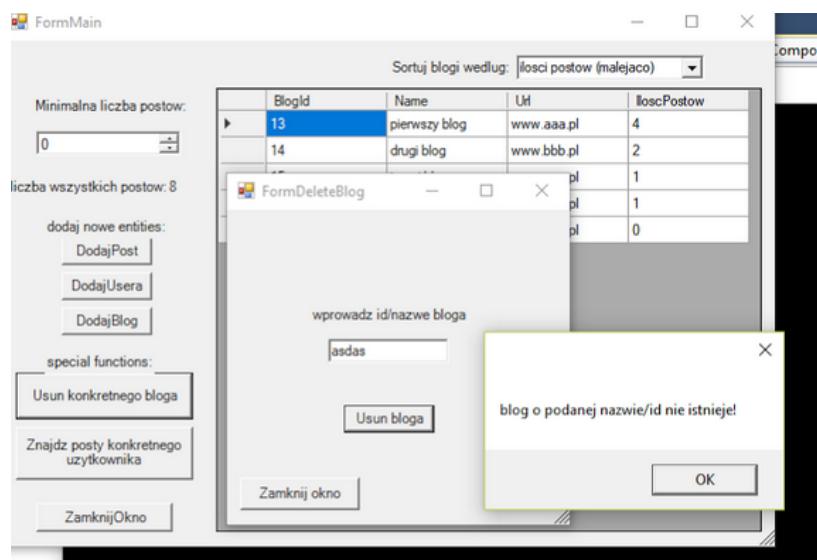
The screenshot shows a Windows application window titled "FormMain". At the top right is a dropdown menu labeled "Sortuj blogi wedlug: ilosci postow (malejaco)". Below it is a search bar with the placeholder "Minimalna liczba postow: 0". To the right is a table with columns: BlogId, Name, Url, and IloscPostow. The table contains the following data:

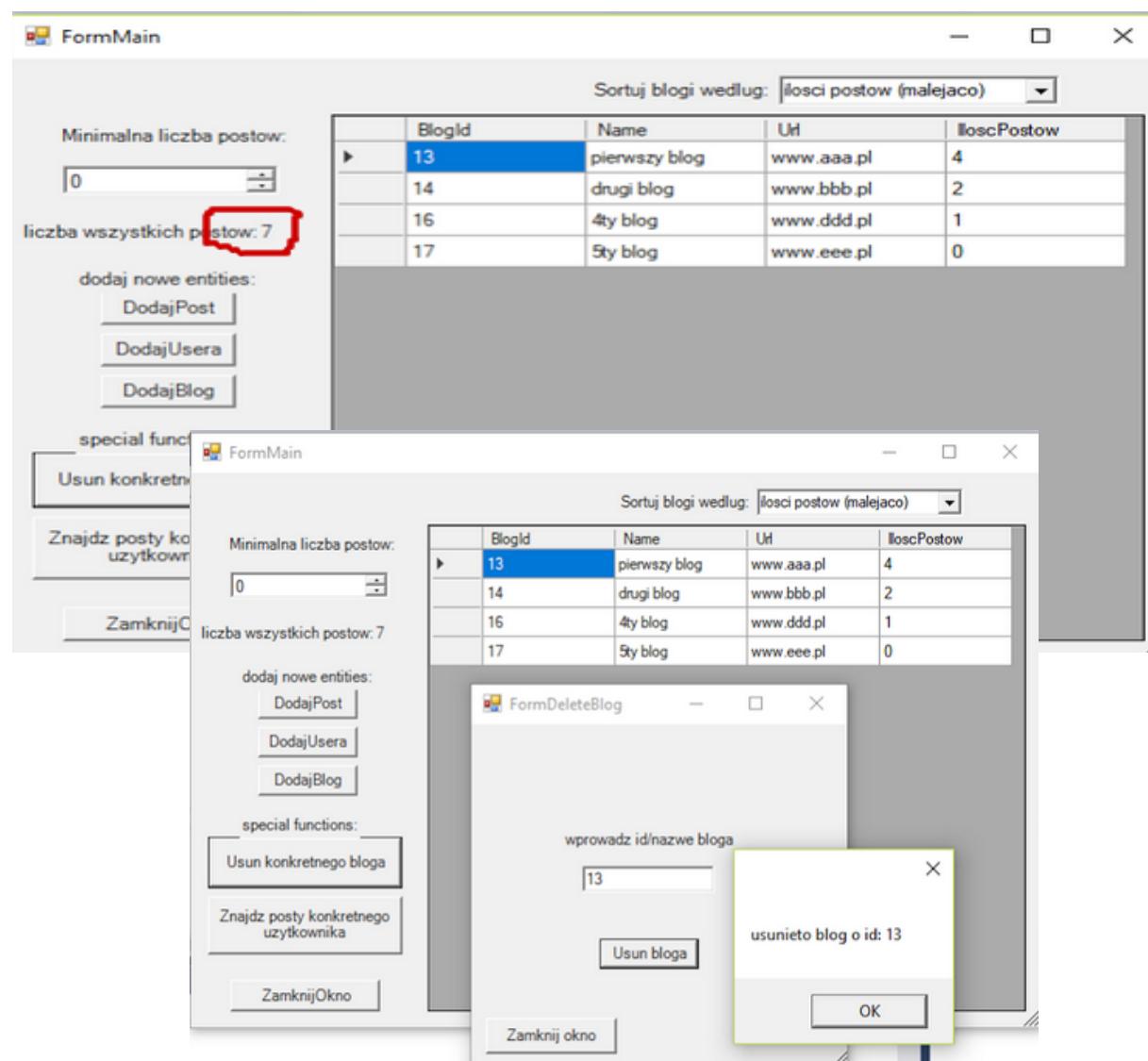
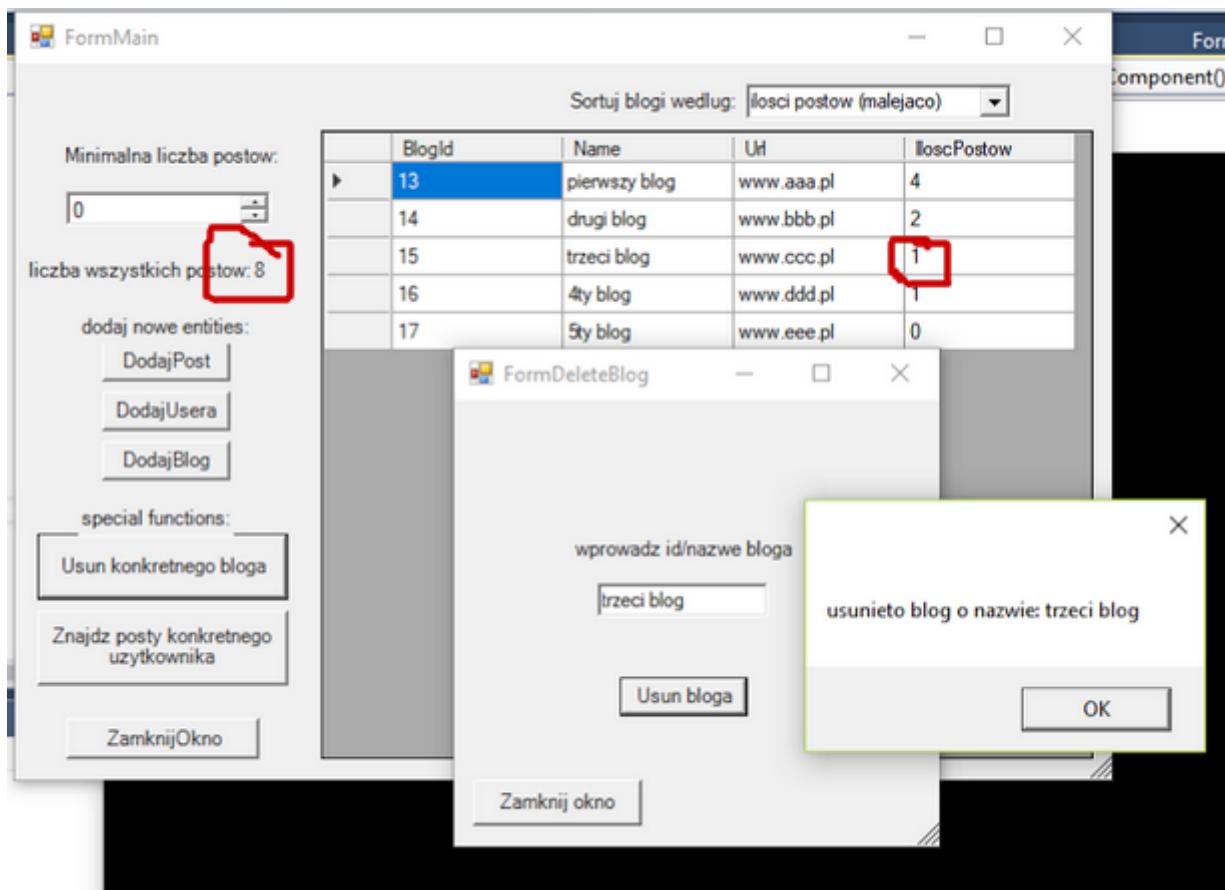
BlogId	Name	Url	IloscPostow
13	pierwszy blog	www.aaa.pl	4
14	drugi blog	www.bbb.pl	2
15	trzeci blog	www.ccc.pl	1
16	4ty blog	www.ddd.pl	1
17	5ty blog	www.eee.pl	0

-jak mozna wywnioskowac z powyzszego screna, label spelnia swoje zadanie.

-Po wcisnieciu przycisku “Usun konkretnego bloga” widocznego w lewym dolnym rogu glownej formy, zostanie wyswietlona forma odpowiedzialna za usuwanie blogow.

-W formie jest wymagane wprowadzenie nazwy lub id bloga, ktorego chcemy usunac, gdy wprowadzimy dane poprawnie blog jest usuwany oraz WSZYSTKIE POSTY, ktore znajdowaly sie w tym blogu rowniez sa usuwane razem z nim. Gdy wprowadzimy zle dane (zly id/zla nazwa) zostanie wyswietlona wiadomosc informujaca nas o tym.





FormMain

Sortuj blogi według: ilosci postow (malejaco)

	BlogId	Name	Url	iloscPostow
▶	14	drugi blog	www.bbb.pl	2
	16	4ty blog	www.ddd.pl	1
	17	5ty blog	www.eee.pl	0

Minimalna liczba postow:

liczba wszystkich postow: 3

dodaj nowe entities:

special functions:

## 4) dropdown “minimalna liczba postow”

-dropdown ma za zadanie ograniczyc wypisywane blogi do blogow, ktorych liczba wszystkich postow jest wieksza lub rowna liczbie, ktorej podamy w dropdownie.

-gdy dropdown bedzie wskazywal “0” wszystkie blogi beda widoczne:

Sortuj blogi wedlug: ilosc postow (malejaco)

	BlogId	Name	Url	iloscPostow
▶	13	pierwszy blog	www.aaa.pl	4
	14	drugi blog	www.bbb.pl	2
	15	trzeci blog	www.ccc.pl	1
	16	4ty blog	www.ddd.pl	1
	17	5ty blog	www.eee.pl	0

-kolejno zmiany w dropdownie zmniejszaja liczbe pokazanych blogow:

Sortuj blogi wedlug: ilosc postow (malejaco)

	BlogId	Name	Url	iloscPostow
▶	13	pierwszy blog	www.aaa.pl	4
	14	drugi blog	www.bbb.pl	2
	15	trzeci blog	www.ccc.pl	1
	16	4ty blog	www.ddd.pl	1

Sortuj blogi wedlug: ilosc postow (malejaco)

	BlogId	Name	Url	iloscPostow
▶	13	pierwszy blog	www.aaa.pl	4
	14	drugi blog	www.bbb.pl	2

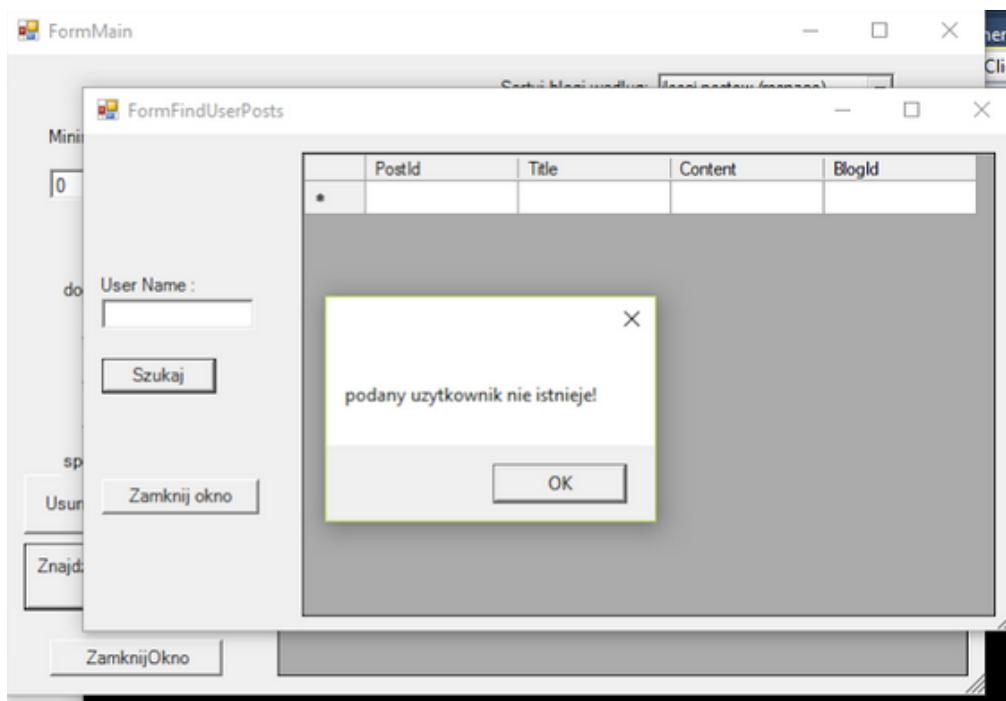
	BlogId	Name	Url	iloscPostow
▶				

## 5) Znajdowanie postow konkretnego uzytkownika

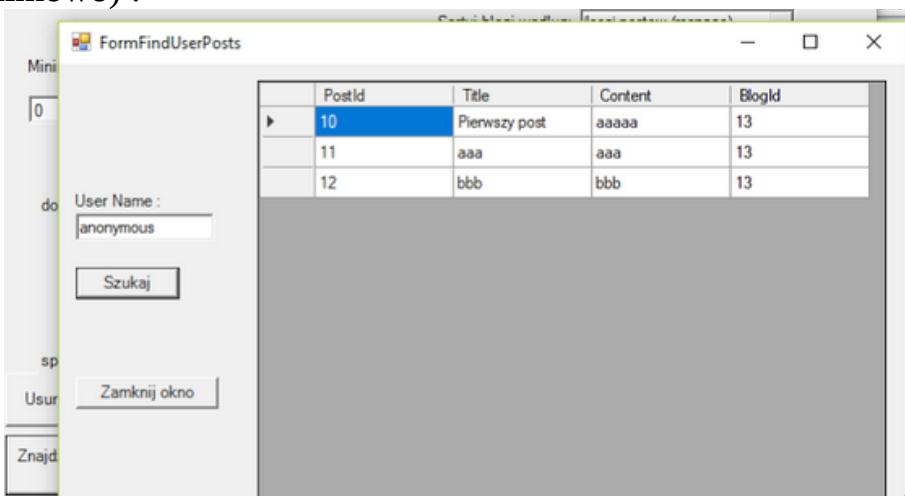
-po wcisnieciu przycisku “znajdz posty konkretnego uzytkownika” widocznego w lewym dolnym rogu glownej formy, zostanie otwarta nowa forma, ktoria jest odpowiedzialna za znajdowanie postow konkretnego uzytkownika.

-wymagane jest podanie nazwy uzytkownika, po zatwierdzeniu wcisnieciem przypisku “szukaj” gdy nazwa zostanie podana zostana wyswietlone wszystkie posty, ktore posiada uzytkownik.

-gdy podamy zla nazwe uzytkownika (jej brak lub nazwa uzytkownika, ktoria nie istnieje) :



-tutaj mozemy zobaczyć posty “uzytkownika” anonymous (czyli wszystkie posty dodane anonimowo) :



# Kod Programu :

a) klasa kontekstowa + klasa Blog/Post/User

```
9  namespace FinalEF
10 {
11     class Program
12     {
13         static void Main(string[] args)
14         {
15             using (var db = new BlogContext())
16             {
17                 FormMain f = new FormMain();
18                 f.ShowDialog();
19             }
20         }
21     }
22
23     public class Blog
24     {
25         public int BlogId { get; set; }
26         public string Name { get; set; }
27         public string Url { get; set; }
28         public int IloscPostow { get; set; }
29         public virtual List<Post> Posts { get; set; }
30     }
31
32     public class Post
33     {
34         public int PostId { get; set; }
35         public string Title { get; set; }
36         public string Content { get; set; }
37
38         public int BlogId { get; set; }
39         public virtual Blog Blog { get; set; }
40
41         public string UserName { get; set; }
42         public virtual User User { get; set; }
43     }
44
45     public class User {
46         [Key]
47         public string UserName { get; set; }
48         public string Description { get; set; }
49     }
50 }
```

```
51  public class BlogContext : DbContext
52  {
53      public DbSet<Blog> Blogs { get; set; }
54      public DbSet<Post> Posts { get; set; }
55      public DbSet<User> Users { get; set; }
56
57      protected override void OnModelCreating(DbModelBuilder modelBuilder)
58      {
59          modelBuilder.Entity<User>().Property(u => u.Description).HasColumnName("DetailedDescription");
60          modelBuilder.Entity<Post>().Property(u => u.UserName).HasColumnName("Author");
61          modelBuilder.Entity<Blog>().Property(b => b.Name).IsRequired();
62      }
63  }
```

-zmiany jakie zostaly wprowadzone (w porownaniu do tego co robilismy na zajeciach) to dodane pole “iloscpstow” do blogow (niestety nie udalo mi sie znalezc trogerow w entity frameworku wiec pozostalo zrobienie ich bezposrednio w bazie danych lub pomaganie sobie funkcjami “trigero podobnymi”).

b) Glowna forma :

-poza zwyklymi listenerami, ktore po prostu nasluchuja na klikniecie buttona I gdy button zostanie klikniety otwieraja nowa forme stworzylem dwie funkcje pomocnicze: “odswiez” oraz “aktualizujIloscPostow”. Funkcja “odswiez” jest odpowiedzialna za posortowanie naszego viewgrida zawierajacego wszystkie blogi wedlug odpowiedniej opcji (wybranej w dropdownie odpowiedzialnym za sposoby sortowania), ta funkcja jest odpalana za kazdym razem gdy zachodzi jakas zmiana na blogach. Funkcja “aktualizujIloscPostow” jest odpalana za kazdym razem gdy nastepuje zmiana na postach (jest to taki ala trigger, ktory czuwa nad zmianami dokonowanymi nad postami I aktualizuje odpowiednio baze oraz nasza forme).

```
11 namespace FinalEF
12 {
13     public partial class FormMain : Form
14     {
15         public static int sortCheckBox = 0;
16         private int minNumOfPosts = 0;
17         private int numberofPosts = 0;
18         public FormMain()
19         {
20             InitializeComponent();
21         }
22
23         private void FormMain_Load(object sender, EventArgs e)
24         {
25
26             comboBox1.SelectedIndex = 6;
27             this.akutalizujIloscPostow();
28             this.odswiez();
29
30         }
31
32         private void akutalizujIloscPostow()
33         {
34             using (var db = new BlogContext())
35             {
36                 int iloscPostow = db.Posts.Count();
37                 this.numberofPosts = iloscPostow;
38                 label6.Text = numberofPosts.ToString();
39             }
40         }
41
42         //dodaj Post
43         private void button4_Click(object sender, EventArgs e)
44         {
45             FormAddPost f = new FormAddPost();
46             f.ShowDialog();
47             this.akutalizujIloscPostow();
48             this.odswiez();
49
50
51
52
53
54         //dodaj Blog
55         private void button1_Click(object sender, EventArgs e)
56         {
57             FormAddBlog f = new FormAddBlog();
58             f.ShowDialog();
59             this.odswiez();
60         }
61         //zamknij okno
62         private void button2_Click(object sender, EventArgs e)
63         {
64             this.Close();
65         }
66
67 }
```

```
    f.ShowDialog();
    this.akutalizujIloscPostow();
    this.odswiez();
}

}

//dodaj Blog
private void button1_Click(object sender, EventArgs e)
{
    FormAddBlog f = new FormAddBlog();
    f.ShowDialog();
    this.odswiez();
}
//zamknij okno
private void button2_Click(object sender, EventArgs e)
{
    this.Close();
}
```

```
private void odswiez()
{
    using (var db = new BlogContext())
    {
        if (sortCheckBox == 0)
        {
            dataGridView1.DataSource = db.Blogs.Where(b => b.IloscPostow >= minNumOfPosts).ToList();
        }
        else if (sortCheckBox == 1)
        {
            dataGridView1.DataSource = db.Blogs.Where(b => b.IloscPostow >= minNumOfPosts).OrderByDescending(b => b.BlogId).ToList();
        }
        else if (sortCheckBox == 2)
        {
            dataGridView1.DataSource = db.Blogs.Where(b => b.IloscPostow >= minNumOfPosts).OrderBy(b => b.Name).ToList();
        }
        else if (sortCheckBox == 3)
        {
            dataGridView1.DataSource = db.Blogs.Where(b => b.IloscPostow >= minNumOfPosts).OrderByDescending(b => b.Name).ToList();
        }
        else if (sortCheckBox == 4)
        {
            dataGridView1.DataSource = db.Blogs.Where(b => b.IloscPostow >= minNumOfPosts).OrderBy(b => b.Url).ToList();
        }
        else if (sortCheckBox == 5)
        {
            dataGridView1.DataSource = db.Blogs.Where(b => b.IloscPostow >= minNumOfPosts).OrderByDescending(b => b.Url).ToList();
        }
        else if (sortCheckBox == 6)
        {
            dataGridView1.DataSource = db.Blogs.Where(b => b.IloscPostow >= minNumOfPosts).OrderByDescending(b => b.IloscPostow).ToList();
        }
        else if (sortCheckBox == 5)
        {
            dataGridView1.DataSource = db.Blogs.Where(b => b.IloscPostow >= minNumOfPosts).OrderByDescending(b => b.Url).ToList();
        }
        else if (sortCheckBox == 6)
        {
            dataGridView1.DataSource = db.Blogs.Where(b => b.IloscPostow >= minNumOfPosts).OrderByDescending(b => b.IloscPostow).ToList();
        }
    }
}
```

```
115 //dropdown z opcjami
116 private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
117 {
118     sortCheckBox = comboBox1.SelectedIndex;
119
120     this.odswiez();
121 }
122
123 private void label1_Click(object sender, EventArgs e)
124 {
125 }
126
127
128 //dodaj usera
129 private void button5_Click(object sender, EventArgs e)
130 {
131     FormAddUser f = new FormAddUser();
132     f.ShowDialog();
133 }
134
135 private void numericUpDown1_ValueChanged(object sender, EventArgs e)
136 {
137     this.minNumOfPosts = int.Parse(numericUpDown1.Value.ToString());
138     this.odswiez();
139 }
140 //find particular user posts
141 private void button3_Click(object sender, EventArgs e)
142 {
143     FormFindUserPosts f = new FormFindUserPosts();
144     f.ShowDialog();
145 }

146
147 //delete particular blog
148 private void button6_Click(object sender, EventArgs e)
149 {
150     FormDeleteBlog f = new FormDeleteBlog();
151     f.ShowDialog();
152     this.akutalizujIloscPostow();
153     this.odswiez();
154 }
155 }
156 }
```

c) Forma odpowiedzialna za dodawanie blogow :

```
11 | namespace FinalEF
12 | {
13 |     public partial class FormAddBlog : Form
14 |     {
15 |         public FormAddBlog()
16 |         {
17 |             InitializeComponent();
18 |         }
19 |
20 |         private void FormAddBlog_Load(object sender, EventArgs e)
21 |         {
22 |
23 |         }
24 |         //zakoncz
25 |         private void button1_Click(object sender, EventArgs e)
26 |         {
27 |             this.Close();
28 |         }
29 |
30 |         //zatwierdz
31 |         private void button2_Click(object sender, EventArgs e)
32 |         {
33 |             using (var db = new BlogContext())
34 |             {
35 |                 if (textBox2.Text == "")
36 |                     MessageBox.Show("wymagane jest podanie nazwy bloga!");
37 |                 else
38 |                 {
39 |                     string url;
40 |                     if (textBox1.Text == "")
41 |                     {
42 |                         url = "brak url.";
43 |                     }
44 |                     else
45 |                     {
46 |                         url = textBox1.Text;
47 |                     }
48 |                 }
49 |                 var blog = new Blog();
50 |                 blog.Name = textBox2.Text;
51 |                 blog.Url = url;
52 |                 db.Blogs.Add(blog);
53 |                 db.SaveChanges();
54 |                 MessageBox.Show("Pomyślnie dodano blog!");
55 |                 this.Close();
56 |             }
57 |         }
58 |     }
59 | }
```

#### d) Forma odpowiedzialna za dodawanie posta

```
11 namespace FINALTER
12 {
13     public partial class FormAddPost : Form
14     {
15         public FormAddPost()
16         {
17             InitializeComponent();
18         }
19
20         private void FormAddPost_Load(object sender, EventArgs e)
21         {
22
23
24
25         private void button2_Click(object sender, EventArgs e)
26         {
27             this.Close();
28         }
29
30         private void button1_Click(object sender, EventArgs e)
31         {
32             using (var db = new BlogContext())
33             {
34                 if (textBox1.Text == "")
35                 {
36                     MessageBox.Show("wymagane jest podanie id bloga, do ktorego ma byc przypisany post!");
37                     return;
38                 }
39                 else
40                 {
41                     try
42                     {
43                         int id = int.Parse(textBox1.Text);
44                         var query = from b in db.Blogs
45                                     where b.BlogId == id
46                                     select b.BlogId;
47                         if (!query.Contains<int>(id))
48
49                         if (!query.Contains<int>(id))
50                         {
51                             MessageBox.Show("blog o podanym id nie istnieje!");
52                             return;
53                         }
54                     }
55                     catch {
56                         MessageBox.Show("blog o podanym id nie istnieje! (prawdopodobnie bledny format id!)");
57                         return;
58                     }
59                 }
60                 if (textBox3.Text == "")
61                 {
62                     MessageBox.Show("wymagane jest podanie tytulu posta!");
63                     return;
64                 }
65                 if (richTextBox1.Text == "")
66                 {
67                     MessageBox.Show("wymagane jest podanie zawartosci posta!");
68                     return;
69                 }
70                 else
71                 {
72                     string author;
73                     if (textBox2.Text == "")
74                     {
75                         author = "anonymous";
76                         //dodaj anonymousa do db
77                         try
78                         {
79                             var anonymousID = from u in db.Users
80                                     where u.UserName == author
81                                     select u.UserName;
82                             if (!anonymousID.Contains<string>(author))
83                             {
84                                 var user = new User();
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
```

```

86             user.Description = "anonymous user";
87             user.UserName = author;
88             db.Users.Add(user);
89             db.SaveChanges();
90         }
91     }
92     catch
93     {
94         MessageBox.Show("blad przy dodawaniu anonimowego posta!");
95         return;
96     }
97 }
98 else
99 {
100     author = textBox2.Text;
101     try
102     {
103         var query = from u in db.Users
104                         where u.UserName == author
105                         select u.UserName;
106         if (!query.Contains<string>(author))
107         {
108             MessageBox.Show("podany użytkownik nie istnieje!");
109             return;
110         }
111     }
112     catch
113     {
114         MessageBox.Show("blad przy dodawaniu posta (validacja użytkownika)");
115         return;
116     }
117 }
118 var post = new Post();
119 int blogid = int.Parse(textBox1.Text);
120 post.BlogId = blogid;
121 db.Blogs.FirstOrDefault(b=>b.BlogId== blogid).IloscPostow+=1;

123         db.Blogs.FirstOrDefault(b=>b.BlogId== blogid).IloscPostow+=1;
124         post.UserName = author;
125         post.Title = textBox3.Text;
126         post.Content = richTextBox1.Text;
127         db.Posts.Add(post);
128         db.SaveChanges();
129         MessageBox.Show("Pomyślnie dodano post!");
130         this.Close();
131     }
132 }
133 }
134
135 //blogID      = textBox1
136 //author       = textBox2
137 //post title   = textBox3
138 //text Content = richTextBox1
139 }
140
141 }

```

e) forma odpowiedzialna za dodawanie userow

```
11 namespace FinalEF
12 {
13     public partial class FormAddUser : Form
14     {
15         public FormAddUser()
16         {
17             InitializeComponent();
18         }
19
20         private void FormAddUser_Load(object sender, EventArgs e)
21         {
22
23         }
24         //zakoncz
25         private void button2_Click(object sender, EventArgs e)
26         {
27             this.Close();
28         }
29         //zatwierdz
30         private void button1_Click(object sender, EventArgs e)
31         {
32             using (var db = new BlogContext())
33             {
34                 if (textBox1.Text == "")
35                 {
36                     MessageBox.Show("wymagane jest podanie nazwy usera!");
37                 }
38                 else
39                 {
40                     string description;
41                     if (richTextBox1.Text == "")
42                     {
43                         description = "brak opisu.";
44                     }
45                     else
46                     {
47                         description = richTextBox1.Text;
48
49                         var query = from u in db.Users
50                                     where u.UserName == textBox1.Text
51                                     select u.UserName;
52                         if (query.Contains<string>(textBox1.Text))
53                         {
54                             MessageBox.Show("podany uzytkownik juz istnieje!");
55                             return;
56                         }
57                         else
58                         {
59
60                             var user = new User();
61                             user.UserName = textBox1.Text;
62                             user.Description = description;
63                             db.Users.Add(user);
64                             db.SaveChanges();
65                             MessageBox.Show("Pomyślnie dodano usera!");
66                             this.Close();
67                         }
68                     }
69                 }
70             }
71         }
72 }
```

f) forma odpowiedzialna za usuwanie blogow :

```
11 namespace FinalEF
12 {
13     public partial class FormDeleteBlog : Form
14     {
15         public FormDeleteBlog()
16         {
17             InitializeComponent();
18         }
19
20         private void button2_Click(object sender, EventArgs e)
21         {
22             this.Close();
23         }
24
25         private void button1_Click(object sender, EventArgs e)
26         {
27             using (var db = new BlogContext())
28             {
29
30                 var query = from b in db.Blogs
31                             where b.BlogId.ToString() == textBox1.Text
32                             select b.BlogId.ToString();
33                 if (!query.Contains<string>(textBox1.Text))
34                 {
35                     string name = textBox1.Text;
36                     query = from b in db.Blogs
37                             where b.Name == name
38                             select b.Name;
39                     if (!query.Contains<string>(name))
40                     {
41                         MessageBox.Show("blog o podanej nazwie/id nie istnieje!");
42                         return;
43                     }
44                 else
45                 {
46                     var id = db.Blogs.Where(b => b.Name == name).ToArray()[0].BlogId;
47                     foreach (var entity in db.Posts)
48                     {
49                         if (entity.BlogId == id)
50                         {
51                             db.Posts.Remove(entity);
52                         }
53                     }
54                     db.SaveChanges();
55
56                     db.Blogs.Remove(db.Blogs.FirstOrDefault(b => b.Name == name));
57
58                     db.SaveChanges();
59                     MessageBox.Show("usunieto blog o nazwie: " + name);
60                 }
61             }
62         else
63         {
64             var id = int.Parse(textBox1.Text);
65             //removing posts
66             foreach (var entity in db.Posts)
67             {
68                 if(entity.BlogId == id)
69                 {
70                     db.Posts.Remove(entity);
71                 }
72             }
73             db.SaveChanges();
74             //removing blogs
75             db.Blogs.Remove(db.Blogs.FirstOrDefault(b => b.BlogId == id));
76             db.SaveChanges();
77             MessageBox.Show("usunieto blog o id: " + id.ToString());
78         }
79         this.Close();
80     }
81 }
82 }
```

g) forma odpowiedzialna za znajdowanie postow konkretnego usera :

```
11 |namespace FinalEF
12 |
13 |{
14 |    public partial class FormFindUserPosts : Form
15 |
16 |        public FormFindUserPosts()
17 |        {
18 |            InitializeComponent();
19 |
20 |        }
21 |
22 |        private void button1_Click(object sender, EventArgs e)
23 |        {
24 |            using (var db = new BlogContext())
25 |            {
26 |                var author = textBox1.Text;
27 |
28 |                var query = from u in db.Users
29 |                            where u.UserName == author
30 |                            select u.UserName;
31 |                if (!query.Contains<string>(author))
32 |                {
33 |                    MessageBox.Show("podany uzytkownik nie istnieje!");
34 |                    return;
35 |                }
36 |
37 |                var query2 = db.Posts.Where(p => p.UserName == author);
38 |                if (!query2.Any())
39 |                {
40 |                    MessageBox.Show("podany uzytkownik nie napisal zadnych postow!");
41 |                    return;
42 |                }
43 |
44 |                dataGridView1.DataSource = query2.ToList();
45 |            }
46 |
47 |            private void button2_Click(object sender, EventArgs e)
48 |
49 |            private void button2_Click(object sender, EventArgs e)
50 |            {
51 |                this.Close();
52 |            }
53 |
54 |            private void FormFindUserPosts_Load(object sender, EventArgs e)
55 |            {
56 |                // TODO: This line of code loads data into the '_FinalEF_BlogContextDataSet2.Posts' table. You can move, or remove it
57 |                //this.postsTableAdapter.Fill(this._FinalEF_BlogContextDataSet2.Posts);
58 |            }
59 |
60 |        }
```