

Colegiul Național „Mihai Viteazul” Turda  
Matematică-informatică, intensiv informatică  
Atestat informatică

## **REGISTRUL ANIMALELOR DINTR-UN CABINET VETERINAR**

Autor  
**Szabo Andreea Georgiana**

Profesori îndrumători  
**prof. Miron Florin**  
**prof. Pop Daniela**

2022

# **CUPRINS**

1. Prezentarea generală a temei	
1.1. Informații generale despre limbajul C# .....	3
1.2. Scurta descriere a aplicației.....	3
1.3. Motivul alegerii temei.....	3
2. Resurse hardware și software necesare	
2.1. Hardware.....	4
2.2. Arhitecturi acceptate.....	4
2.3. Software.....	4
3. Realizarea aplicației/ Ghidul programatorului	
3.1. Descriere meniuri.....	5
3.2. Structuri baze de date.....	7
3.3. Secvențe reprezentative de cod sursă (+descrierea lor).....	8
4. Utilizarea aplicației /Ghidul utilizatorului	
4.1. Ghidul medicului veterinar.....	13
4.2. Ghidul proprietarului unui animal de companie.....	13
5. Extinderi posibile ale aplicației.....	14
6. Bibliografie.....	14

# **1. Prezentarea generală a temei**

## **1.1. Informații generale**

### **1.1.1. VISUAL STUDIO**

Visual Studio include un set complet de instrumente de dezvoltare pentru generarea de aplicații ASP.NET, Servicii Web XML, aplicații desktop și aplicații mobile. Visual Basic, Visual C++, Visual C# și Visual J# toate folosesc același mediu de dezvoltare integrat (IDE) care le permite partajarea instrumentelor și facilitează crearea de soluții folosind mai multe limbaje de programare. Aceste limbaje permit să beneficieze de caracteristicile .NET Framework care oferă acces la tehnologii cheie care simplifică dezvoltarea de aplicații web ASP și XML Web Services cu Visual Web Developer.

### **1.1.2. C#**

C# este un limbaj de programare orientat pe obiecte conceput de Microsoft la sfârșitul anilor 90. A fost conceput ca un concurent pentru limbajul Java. Ca și acesta, C# este un derivat al limbajului de programare C++ și simplifică mult scrierea de programe pentru sistemul de operare Windows.

## **1.2. Scurta descriere a aplicației**

Aplicația presupune memorarea anumitor date pentru fiecare animal ce aparține de cabinetul veterinar respectiv. Acestea se refera la: tip(caine, pisica, etc), nume, rasa, greutate, gen(masculin sau feminin), data nasterii.

Pe langa acestea, se memorează programările fiecărui animal, atât cele precedente (de când este înregistrat la cabinetul respectiv), cât și următoarele, bolile de care suferă în prezent, cu tratamentele corespunzătoare și date de contact ale stăpânului.

Conține și o lista cu prețuri care urmează sa îi fie utilă atât medicului, pentru anumite statistici, cât și proprietarului, pentru o informare cât mai rapidă.

După cum am menționat mai sus, aplicația este atât la dispoziția medicului, cât și la dispoziția proprietarilor de animale, fie că aceștia doresc doar sa se informeze în privința costurilor sau doresc sa afle istoricul medical al animalului lor. Medicul, în meniul său are la dispoziție o mulțime de operații cum ar fi adăugare, ștergerea sau editarea informațiilor. Acestora li se mai adauga statisticile, posibilitatea de a vedea ce proprietari nu s-au prezentat cu animalele lor la controalele precedente și o reprezentare a animalelor care au de facut vizita medicala într-o anumita zi.

## **1.3. Motivul alegerii temei**

Cand eram mica spuneam tuturor ca îmi doresc sa devin medic veterinar pentru că iubesc animalele. Ulterior, am ajuns la concluzia că nu sunt făcută pentru a fi doctor și că pasiunea pentru informatica este mult mai mare. În aceasta aplicatie am dorit să mă pun în locul unui medic veterinar, însă doar pentru a crea informatic ceva ce ar putea să-i fie folositor. Astfel, mi-am îndeplinit oarecum o dorință din copilărie.

## **2. Resurse hardware și software necesare**

### **2.1. Hardware (Cerințe minime de sistem)**

- Procesor 1.6 GHz
- 1GB RAM
- 5,5 GB spațiu pe disc
- Unitate optică cu DVD-ROM
- Conexiune la internet

### **2.2. Arhitecturi acceptate**

- 32-bit
- 64-bit

### **2.3. Software**

- Windows Vista( x86, x64)
- Windows XP( x86, x64)
- Windows 7( x86, x64)
- Windows 8( x86, x64)
- Windows 10( x86, x64)
- Microsoft .NET Framework 4
- Microsoft Server SQL 2008 service pack 3
- Microsoft Server SQL R2 service pack 1

### 3. Realizarea aplicației/ Ghidul programatorului

#### 3.1. Descriere meniuri

Aplicația are la bază două meniuri invizibile inițial.

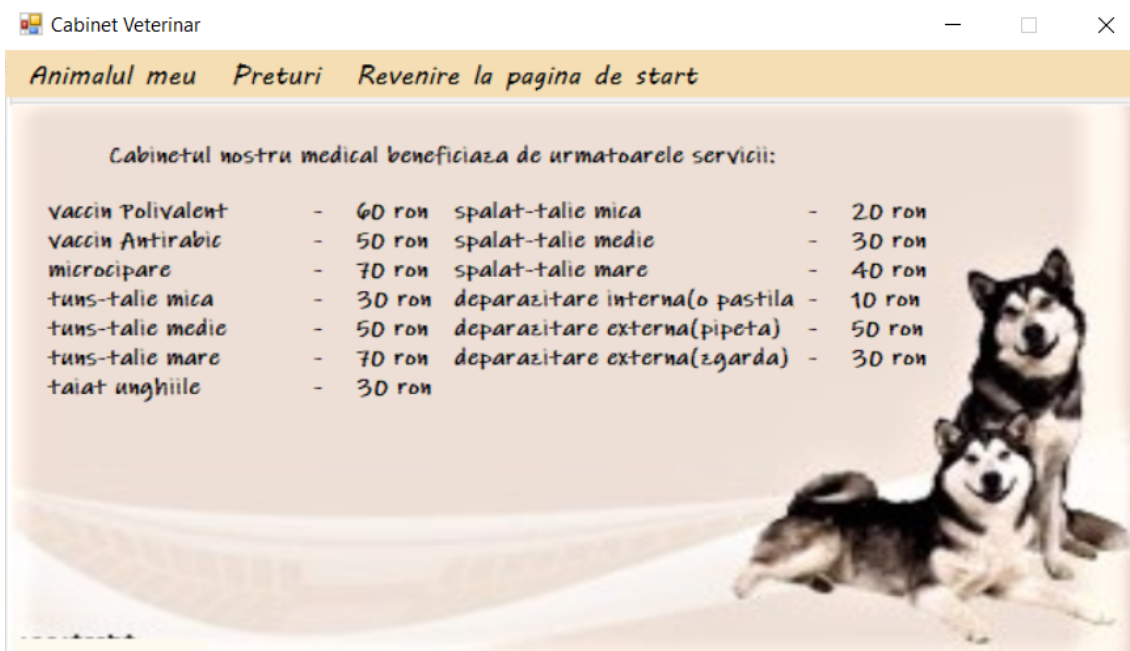
Pagina de start impune utilizatorului să aleagă dacă vizitează pagina ca proprietar de animal sau ca medic.



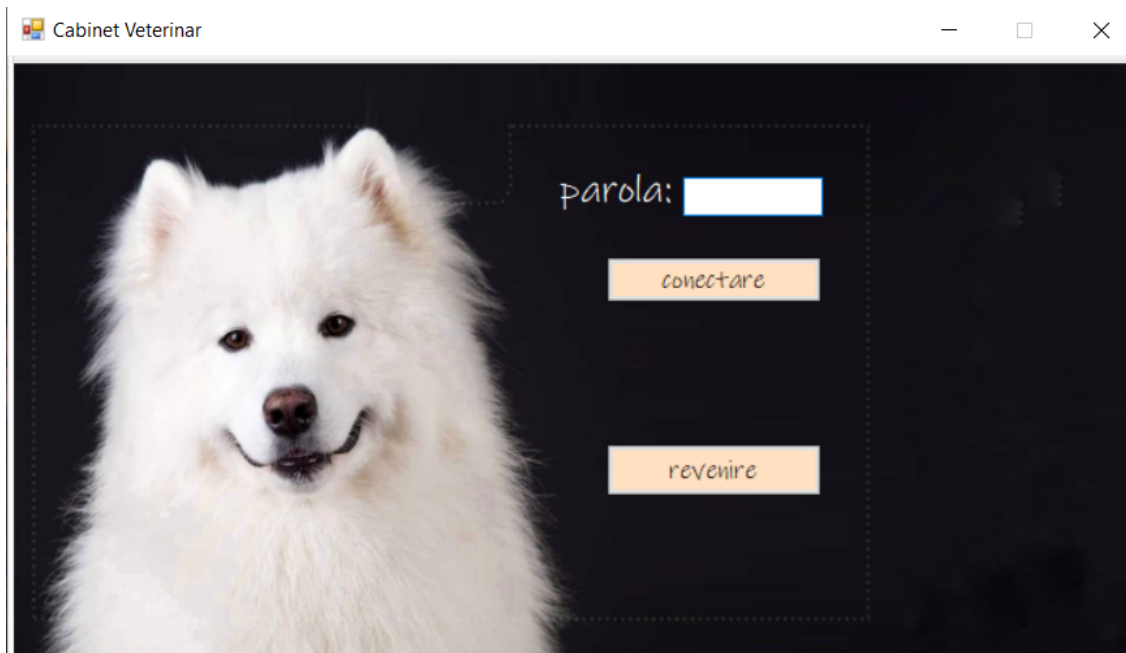
În cazul în care selectează “proprietar” va deveni vizibil unul dintre meniuri și va fi distribuit către un tabpage nou cu toate procedurile alături de prețurile corespunzătoare. Această opțiune apare și în meniu sub denumirea "Prețuri".

Alta opțiune ar fi "Animalul meu", care conține 4 subcategorii: “caracteristici generale”, “viitoare programari”, “consultații neefectuate” și “tratamente”.

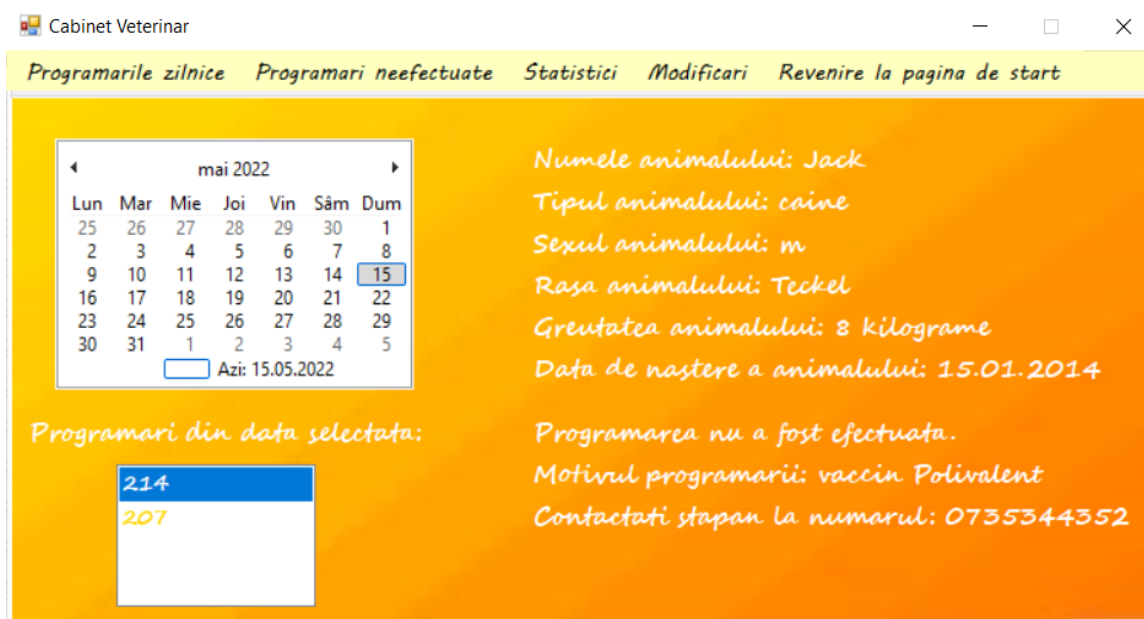
Ultima opțiune este cea de revenire la pagina de start.



În cazul în care utilizatorul selectează opțiunea “Doctor” este direcționat către o pagina unde i se va cere o parolă (parola este prestabilită și va fi încredințată doar medicului cabinetului sau mai multor medici în cazul în care aceștia lucrează în același loc cu pacienți comuni). Când parola introdusă este corectă, medicului îi va apărea o alta fereastră și un alt meniu, altfel, va primi un mesaj care îl înștiințează că parola este greșită și are oricând posibilitatea revenirii la pagina de start.



Fereastră care i se deschide când parola este corectă are un calendar în care este preselectată data de azi și alături o listă cu toate codurile animalelor ce au programări în ziua curentă. Aceasta se găsește și în meniu sub denumirea de “Programările zilnice”.



În meniu se află și opțiunea de “Programari neefectuate” care redirectioneaza medicul spre o fereastră cu o lista a animalelor ce nu s-au prezentat.

O alta opțiune este cea de statistici cu subcategoriile “ultima luna” și “luna viitoare”, fiecare cu două subcategorii identice.

A patra opțiune este cea de modificare, unde, pentru fiecare tip de informații, se poate adăuga, edita sau șterge.

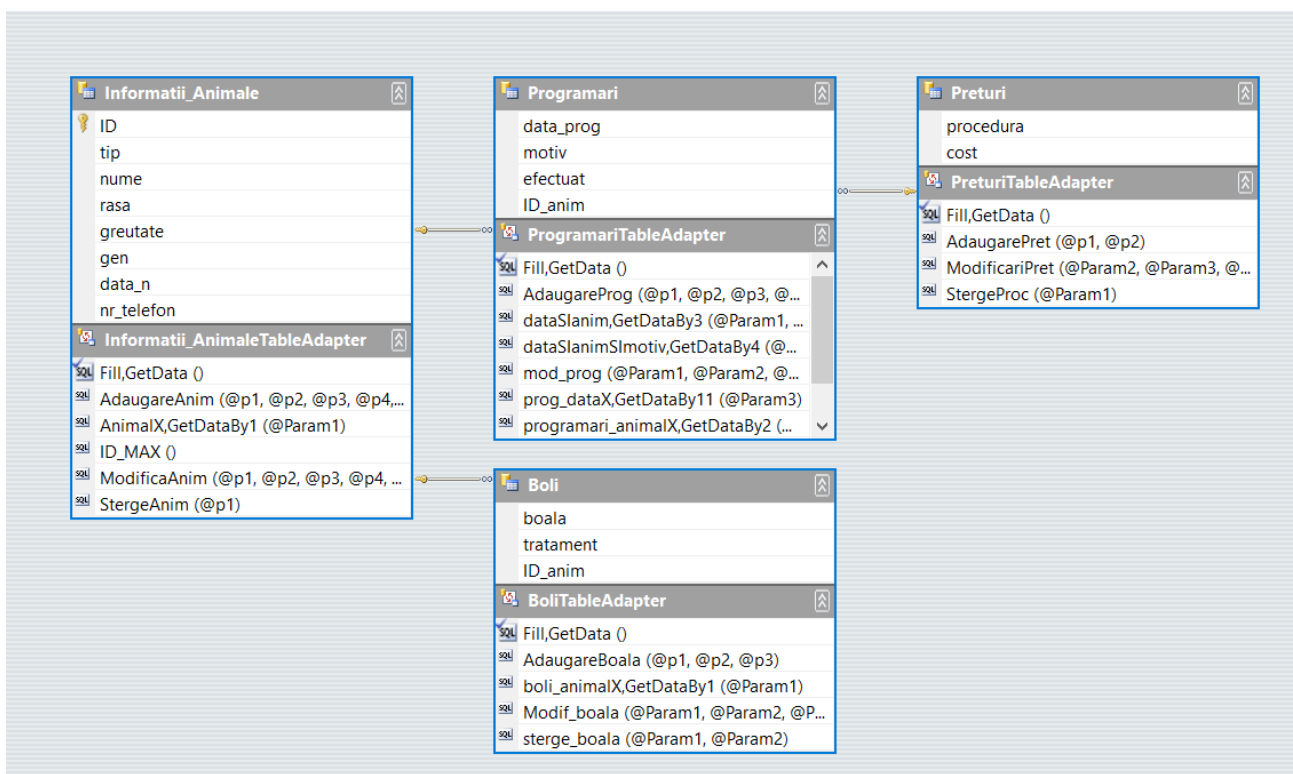
Ultima opțiune este de revenire la pagina de start.



### 3.2. Structuri baze de date

Baza de date este una singura si poarta denumirea de “registru”.

Aceasta este alcatuita din 4 tabele conectate între ele.



Primul tabel este cel principal cu informațiile despre animale, fiecărui animal i se atribuie aici un ID unic care va fi folosit în conexiunea cu celelalte tabele.

Al doilea tabel conține toate programările, fiecare animal având posibilitatea de a avea nenumărate programări.

Al treilea este asemănător cu al doilea, doar ca aici sunt menționate bolile și tratamentele animalelor.

Al patrulea conține informații despre preț în funcție de procedură. Coloana “procedură” este legată de coloana “motiv” din tabela cu programări, pentru a se putea efectua mai ușor operațiile din cadrul statisticilor.

### 3.3. Secvențe reprezentative de cod sursă (+descrierea lor)

#### 3.3.1. Afisarea programarilor din ziua curentă în funcție de data selectata pe calendar

Pentru aceasta am creat un tabpage nou unde utilizatorul este trimis atunci când apasă pe opțiunea "programări viitoare" din meniu (același cod este accesat și în urma introducerii parolei corecte).

Datei selectate pe calendar îi este atribuită ziua curentă și celorlalte obiecte le sunt atribuite valori nule.

În urma unei selecții pe calendar se apelează o interogare creată precedent (aceasta transmite toate datele despre animalele care au programare într-o zi anume).

Pe baza interogării se completează lista cu toate codurile animalelor înregistrate.

```
private void programarileZilniceToolStripMenuItem_Click(object sender, EventArgs e)
{
    listBox3.Items.Clear();
    tabControl1.SelectedTab = tabPage4;
    monthCalendar1.SelectionStart = DateTime.Now;
    DateTime dt = monthCalendar1.SelectionStart;
    label6.Text = "";
    label20.Text = "";
}

private void monthCalendar1_DateChanged(object sender, DateRangeEventArgs e)
{
    listBox3.Items.Clear();
    label6.Text = "";
    label20.Text = "";
    DateTime dt = monthCalendar1.SelectionStart;
    programariTableAdapter.prog_dataX(registruDataSet.Programari, dt.ToShortDateString());
    DataTable tab = registruDataSet.Programari;
    for (int i = 0; i < tab.Rows.Count; i++)
    {
        listBox3.Items.Add(tab.Rows[i]["ID_anim"]);
    }
}
```

În urma selectării unui cod din lista, interogarea se execută din nou și informațiile sunt preluate și sunt tipărite pe ecran într-o formă accesibilă și însoțită de specificații.



```

private void listBox3_SelectedIndexChanged(object sender, EventArgs e)
{
    if (listBox3.SelectedIndex != -1)
    {
        label6.Text = "";
        label20.Text = "";
        DateTime dt = monthCalendar1.SelectionStart;
        programariTableAdapter.prog_dataX(registruDataSet.Programari, dt.ToShortDateString());
        DataTable tab = registruDataSet.Programari;
        label6.Text = "Numele animalului: " + tab.Rows[listBox3.SelectedIndex]["nume"] + "\n";
        label6.Text += "Tipul animalului: " + tab.Rows[listBox3.SelectedIndex]["tip"] + "\n";
        label6.Text += "Sexul animalului: " + tab.Rows[listBox3.SelectedIndex]["gen"] + "\n";
        label6.Text += "Rasa animalului: " + tab.Rows[listBox3.SelectedIndex]["rasa"] + "\n";
        label6.Text += "Greutatea animalului: " + tab.Rows[listBox3.SelectedIndex]["greutate"] + " kilograme" + "\n";
        label6.Text += "Data de nastere a animalului: " + tab.Rows[listBox3.SelectedIndex]["data_n"] + "\n";
        if (bool.Parse(tab.Rows[listBox3.SelectedIndex]["efectuat"].ToString()))
            label20.Text += "Programarea a fost efectuata.\n";
        else
            label20.Text += "Programarea nu a fost efectuata. \n";
        label20.Text += "Motivul programarii: " + tab.Rows[listBox3.SelectedIndex]["motiv"] + "\n";
        label20.Text += "Contactati stapan la numarul: " + tab.Rows[listBox3.SelectedIndex]["nr_telefon"] + "\n";
    }
}

```

### 3.3.2. Codul de modificare a unei programări

În urma apăsării pe butonul editare, medicul este trimis către un tab unde se află o listă gata completată cu toate codurile animalelor (prin interogarea “Fill”, tot în momentul apăsării).

Medicul va avea de selectat unul dintre coduri (animalul a cărui programare dorește să o editeze) și îi vor apărea în cea de a doua listă toate programările ce corespund animalului (această operație se realizează, de asemenea, printr-o interogare).

```

private void editareToolStripMenuItem2_Click(object sender, EventArgs e)
{
    tabControl1.SelectedTab = tabPage17;
    listBox15.Items.Clear();
    listBox16.Items.Clear();
    listBox9.Items.Clear();
    radioButton4.Checked = false;
    radioButton3.Checked = false;
    informatii_AnimaleTableAdapter.Fill(registruDataSet.Informatii_Animale);
    DataTable tab = registruDataSet.Informatii_Animale;
    for (int i = 0; i < tab.Rows.Count; i++)
    {
        listBox15.Items.Add(tab.Rows[i]["ID"]);
    }
}

private void listBox15_SelectedIndexChanged(object sender, EventArgs e)
{
    if (listBox15.SelectedIndex != -1)
    {
        listBox9.Items.Clear();
        listBox16.Items.Clear();
        programariTableAdapter.programari_animalX(registruDataSet.Programari, listBox15.SelectedItem.ToString());
        DataTable tab = registruDataSet.Programari;
        for (int i = 0; i < tab.Rows.Count; i++)
        {
            listBox9.Items.Add(tab.Rows[i]["data_prog"]);
        }
    }
}

```

În cea de a treia listă, în momentul selectării unei date calendaristice, pe prima linie se enunțează motivul (lista este utilă atunci când există 2 programări în aceeași zi). În a patra listă apar toate posibilele motive pentru un consult și se poate selecta un alt motiv dacă se dorește. La selecție nu se întâmplă nimic, doar la apăsarea pe buton. În partea de jos dintre cele 2 butoane se bifează automat cel care corespunde cu datele memorate despre programare și aceasta se poate modifica.

```
private void listBox9_SelectedIndexChanged(object sender, EventArgs e)
{
    if (listBox9.SelectedIndex != -1)
    {
        listBox16.Items.Clear();
        listBox17.Items.Clear();
        programariTableAdapter.programari_animalX(registruDataSet.Programari, listBox15.SelectedItem.ToString());
        DataTable tab = registruDataSet.Programari;
        listBox17.Items.Add(tab.Rows[listBox9.SelectedIndex]["motiv"]);
        listBox16.Items.Clear();
        preturiTableAdapter.Fill(registruDataSet.Preturi);
        DataTable tab2 = registruDataSet.Preturi;
        for (int i = 0; i < tab2.Rows.Count; i++)
        {
            listBox16.Items.Add(tab2.Rows[i]["procedura"]);
        }
        if ((bool)tab.Rows[listBox9.SelectedIndex]["efectuat"] == false)
        {
            radioButton2.Checked = true;
        }
        else
        {
            radioButton1.Checked = true;
            listBox16.SelectedItem = tab.Rows[listBox9.SelectedIndex]["motiv"];
        }
    }
}
```

Prin apăsarea butonului “modifica” se va face modificarea printr-o interogare de tip “update” unde motivul, data și ID-ul animalului se regăsesc în selecțiile din primele trei liste.

Am ales această variantă pentru a nu permite utilizatorului să greșască un cuvânt și, astfel, să se piardă conexiunea dintre tabele.

```

private void button15_Click_1(object sender, EventArgs e)
{
    if (listBox9.SelectedIndex != -1 && listBox15.SelectedIndex != -1 && listBox16.SelectedIndex != -1)
    {
        if (radioButton2.Checked == false)
        {
            if (radioButton1.Checked == true)
            {
                programariTableAdapter.mod_prog(listBox16.SelectedItem.ToString(), true, listBox15.SelectedItem.ToString(),
                    listBox9.SelectedItem.ToString(), listBox17.Items[0].ToString());
                listBox9.Items.Clear();
                listBox17.Items.Clear();
                listBox16.Items.Clear();
                radioButton1.Checked = false;
                radioButton2.Checked = false;
            }
        }
        else
        {
            if (radioButton1.Checked == false)
            {
                programariTableAdapter.mod_prog(listBox16.SelectedItem.ToString(), false, listBox15.SelectedItem.ToString(),
                    listBox9.SelectedItem.ToString(), listBox17.Items[0].ToString());
                listBox9.Items.Clear();
                listBox17.Items.Clear();
                listBox16.Items.Clear();
                radioButton1.Checked = false;
                radioButton2.Checked = false;
            }
        }
    }
}

```

### 3.3.3. Afișarea tratamentului pentru boala unui anumit animal

Odată cu selectarea opțiunii "tratamente" utilizatorul este direcționat către un tabpage nou unde i se cere introducerea codului animalului. (mesajul apare pe ecran)

```

private void tratamentPentruBolileDeCareSuferaToolStripMenuItem_Click(object sender, EventArgs e)
{
    tabControl1.SelectedTab = tabPage9;
    label19.Text = "";
    textBox5.Text = "";
    label17.Text = "";
    comboBox1.Visible = false;
}

```

La apăsarea pe butonul “cauta”, în cazul în care codul este corect și animalul suferă de cel puțin o boală, îi va apărea o lista ascunsă cu bolile acestuia(aflăte prin apelarea unei interogări). In caz contrar, va fi înștiințat corespunzător.

```

private void button8_Click(object sender, EventArgs e)
{
    label19.Text = "";
    comboBox1.Text = "";
    comboBox1.Visible = false;
    comboBox1.Items.Clear();
    string t = textBox5.Text.Trim();
    label17.Text = "";
    if (t.Length > 0)
    {
        boliTableAdapter.boli_animalX(registruDataSet.Boli, t);
        DataTable tab = registruDataSet.Boli;
        if (tab.Rows.Count > 0)
        {
            for (int i = 0; i < tab.Rows.Count; i++)
                comboBox1.Items.Add(tab.Rows[i]["boala"]);
            label17.Text = "Animalul dumneavoastra, " + tab.Rows[0]["nume"] + ", \n" + "sufera in prezent urmatoarele boli: ";
            comboBox1.Visible = true;
        }
        else
        {
            informatii_AnimaleTableAdapter.AnimalX(registruDataSet.Informatii_Animale, t);
            DataTable tab1 = registruDataSet.Informatii_Animale;
            if (tab1.Rows.Count > 0)
                label17.Text = "Animalul dumneavoastra, in prezent, \nnu sufera de nicio boala. ";
            else
                label17.Text = "Nu exista niciun animal inregistrat \ncu acest ID.";
        }
    }
}

```

La un click pe unul dintre elementele listei se va reține elementul și se va folosi în aceeași interogare care află bolile animalului. În funcție de linia la care se afla în listă cuvântul selectat se află despre care boala este vorba și se tipărește pe ecran tratamentul.

```

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (comboBox1.SelectedIndex != -1)
    {
        string t = textBox5.Text.Trim();
        if (t.Length > 0)
        {
            boliTableAdapter.boli_animalX(registruDataSet.Boli, t);

            DataTable tab = registruDataSet.Boli;
            object o = tab.Rows[comboBox1.SelectedIndex]["tratament"];
            if (o.ToString() != "")
            {
                label19.Text = "Pentru boala " + comboBox1.SelectedItem + " tratamentul recomandat \nde medic este: ";
                label19.Text += tab.Rows[comboBox1.SelectedIndex]["tratament"] + ".";
            }
            else
                label19.Text = "Pentru boala " + comboBox1.SelectedItem + " nu exista \nun tratament recomandat de medic.";
        }
    }
}

```

## **4. Utilizarea aplicației /Ghidul utilizatorului**

### **4.1. Ghidul medicului veterinar**

În urma deschiderii aplicației se selectează opțiunea “doctor”.

În fereastra nou apărută se introduce parola prestabilită (care se cunoaște).

În cazul în care aceasta este corectă apare o altă fereastră și un meniu în partea de sus a ecranului.

Fereastra respectivă are un calendar în care este preselectată data de azi și alături o listă cu toate codurile animalelor ce au programări în ziua curentă. La un click pe o altă dată calendaristică vor apărea codurile animalelor cu programări în ziua selectată. Aceasta fereastra se găsește și în meniu sub denumirea de “Programările zilnice”.

La un click pe unul dintre codurile ce apar în listă, în aceeași fereastră apar toate informațiile despre animalul cu acel ID, despre programarea respectivă și date de contact ale stăpânului.

Tot în meniul de sus apar și “Programari neefectuate”. La o apăsare pe aceasta opțiune apare o lista cu toate programările din trecut care au fost notate ca și neefectuate. Odată cu selectarea unui element din lista apar informațiile despre animalul care nu a fost adus la programare și numărul de telefon al stăpânului.

La opțiunea “statistici” se afla subcategoriile “ultima luna” și “luna viitoare”, fiecare cu două subcategorii identice. Apasand pe “Încasări” se afișază într-o fereastră nouă toate încasarile din ultima luna/din luna următoare. Opțiunea “Servicii de același tip” calculează, tot așa, totalul procedurilor de un fel anume.

Opțiunea a patra este cea de modificare unde, la adăugare se introduc pentru fiecare tip de date în parte, date noi ce vor rămâne în memorie. La modificare se selectează dintre datele existente ce se dorește a se modifica și în casetele respective se înlocuiește vechea informație cu informația nouă dorită. La ștergere se selectează tot dintre datele existente ceva ce se va șterge definitiv din memorie.

Ultima opțiune este cea prin care revine la pagina inițială.

### **4.2. Ghidul proprietarului unui animal de companie**

În urma apăsării pe butonul “proprietar” de pe pagina de start va apărea un meniu în partea de sus și utilizatorul va fi directionat către o fereastră ce conține lista cu tarifele și respectiv procedurile pe care cabinetul le are în vedere. Aceasta fereastra poate fi accesată și apasand opțiunea “prețuri” din meniu.

La apăsarea pe opțiunea “Animalul meu” vor apărea trei subopțiuni, printre care cea de “caracteristici generale”. La un click pe aceasta va apărea o fereastră cu o caseta unde se cere să se introducă codul animalului și să se apese pe buton. La apăsare vor apărea caracteristicile reținute în registrul medical.

Pe același principiu sunt și programările viitoare și cele neefectuate. În plus, acolo utilizatorul este informat și despre suma pe care o are de achitat pentru consultația respectivă. Dacă nu există astfel de programări pentru animalul său i se va aduce la cunoștință.

La subopțiunea “tratamente”, dacă animalul suferă de anumite boli îi va apărea o caseta din care are de selectat boala. În cazul în care medicul a notat un tratament pentru boala respectivă, stăpânului îi va fi menționat mai jos pentru a ști exact ce are de căutat. În cazul în care animalul nu mai suferă de nicio boală sau boala nu necesită tratament acesta va fi informat corespunzător.

A treia opțiune este cea prin care revine la pagina inițială.

## **5. Extinderi posibile ale aplicației**

- O interfață grafică mai bună.
- Adăugarea unor statistici pe întregul an.
- Adăugarea unei comenzi ce îi afișează medicului toate animalele ce au programări pe un motiv anume.

## **6. Bibliografie**

- [https://ro.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://ro.wikipedia.org/wiki/Microsoft_Visual_Studio)
- [https://ro.wikipedia.org/wiki/C\\_sharp](https://ro.wikipedia.org/wiki/C_sharp)