

**Aplicatie pentru gestionarea programarilor in cabinetele veterinare**

**HomeVet**

Realizat de:

Std. Sg. DUMITRU Andreea-Ioana

Std. Sg. ISTRATE Stefan-Daniel

Grupa C113B

Indrumator:

NIȚĂ Ștefania

MARGHESCU Andrei

**CUPRINS**

**Capitolul 1 – Introducere**.................................................................................pag. 3

1.1. Scopul proiectului................................................................................pag. 3

1.2. Structura documentului.......................................................................pag. 3

**Capitolul 2 – Descrierea generala a produsului software**.............................pag. 4

2.1. Descrierea produsului software...........................................................pag. 4

2.2. Detalierea platformei SW/HW............................................................pag. 4

**Capitolul 3 – Detalierea functionalitatilor**.....................................................pag. 5

3.1. Prezentarea funcționalitatilor..............................................................pag. 5

3.2 Diagrama ER si Entity Framework..................................................pag. 18

3.2. Concluzii ................................................... ........................................pag. 18

**Capitolul 1 – Introducere**

* 1. **Scopul proiectului**

Proiectul se prezinta ca o aplicatie de tip site online, care are ca obiect de activitate gestionarea unui cabinet veterinar. Functionalitatile aplicatiei se bazeaza pe impartirea utilizatorilor in trei roluri, si anume medic, asistent si utilizator normal. Aceasta aplicatie are drept utilizatori tinta persoanele ce isi doresc sa isi faca o programare online la un cabinet veterinar, dar si medicii si asistentii care isi doresc sa isi aiba programarile intr-un mediu mai organizat si usor de accesat.

* 1. **Structura documentului**

Documentul este structurat pe trei capitole: capitolul 1 reprezinta introducerea, capitolul 2 prezinta o descriere detaliata a aplicatiei, iar capitolul 3 prezinta cerintele exacte ale aplicatiei.

**Capitolul 2 – Descrierea generala a produsului software**

**2.1. Descrierea produsului software**

Aplicatia este dezvoltata in limbajul de programare C#, iar interfata grafica este prietenoasa realizata cu ajutorul framework-ului WPF.

Conexiunea dintre aplicatie si baza de date se bazeaza pe Entity Framework, cu ajutorul caruia integram baza de date în cadrul aplicatiei astfel incat toate tabelele folosite se vor transforma in clase care pot fi instantiate in aplicatie.

Dupa ce ne-am asigurat ca aceasta conexiune este valida, urmatorul pas este reprezentat de autentificarea in sistem. Utilizatorul va fi intampinat de o interfata prietenoasa si isi va introduce datele de autentificare, dar va avea posibilitatea de a se si inregistra in sistem daca credentialele nu exista in baza de date.

După ce utilizatorul s-a autentificat cu succes, in functie de tipul sau, el va fi redirectionat catre pagina principala asociata.

Utilizatorul normal poate folosi aceasta aplicatie pentru a planifica o programare sau pentru a notifica situatiile de urgenta, poate pune intrebari, isi poate accesa lista cu animalele introduse in baza de date si poate crea programari pentru propriile animale la un cabinet veterinar dintr-o anumita locatie.

Medicul poate accepta sau respinge programarile create de utilizatori, poate raspunde la intrebarile adresate de utilizator, poate actualiza programarea daca trebuie colectata vreo proba, in care delega un asistent sa o colecteze fie de la domiciliu, fie de la cabinetul unde s-a facut programarea, isi poate accesa lista cu programari (doar cele marcate ca acceptate).

Asistentul medicului poate vizualiza lista cu programari impreuna cu cele marcate ca fiind necesara colectarea de probe.

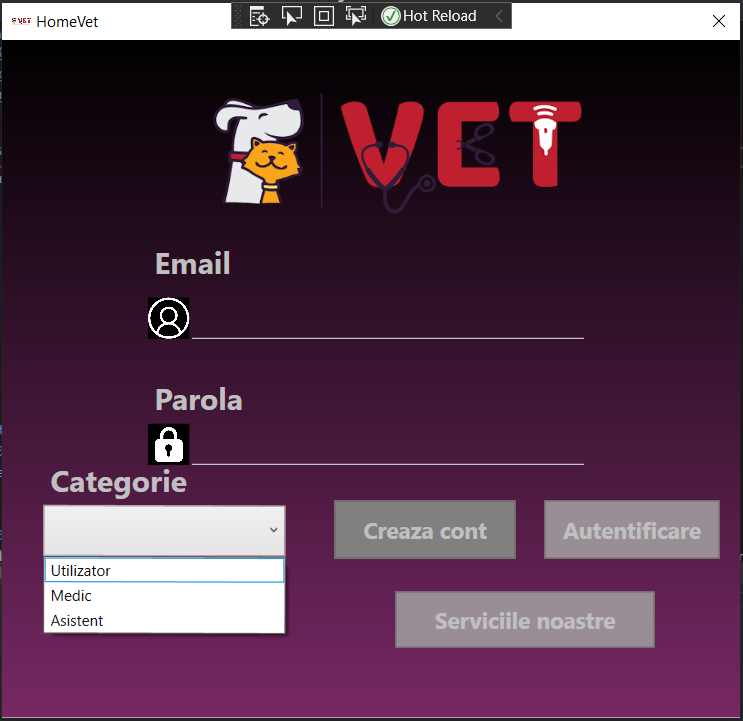
**2.2. Detalierea platformei SW/HW**

Produsul software este dezvoltat pentru dispozitivele pe care ruleaza sistemul de operare Windows 10. Am utilizat mediul de dezvoltare Microsoft Visual Studio, iar baza de date a fost implementata cu ajutorul Microsoft SQL Server . Din punct de vedere harware, este nevoie de un procesor cu o frecventa de 1.5 GHz sau mai mare, si o memorie RAM de peste 4 GB.

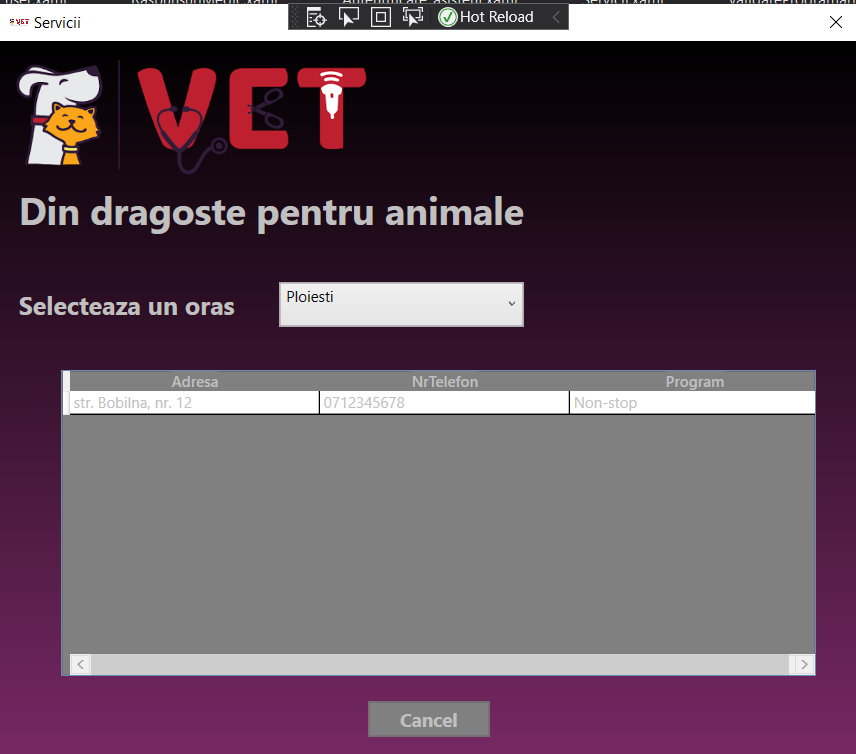
**Capitolul 3 - Detalierea functionalitatilor**

**3.1. Prezentarea functionalitatilor**

# Fereastra principala



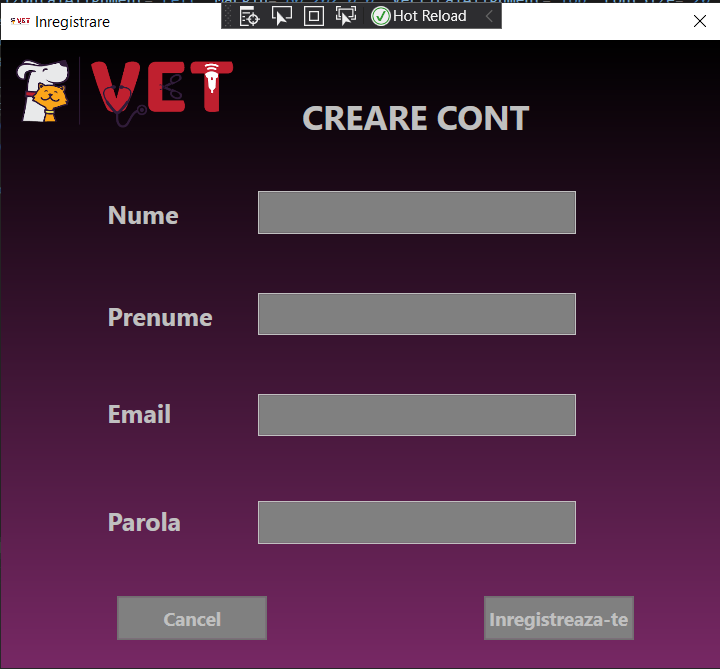
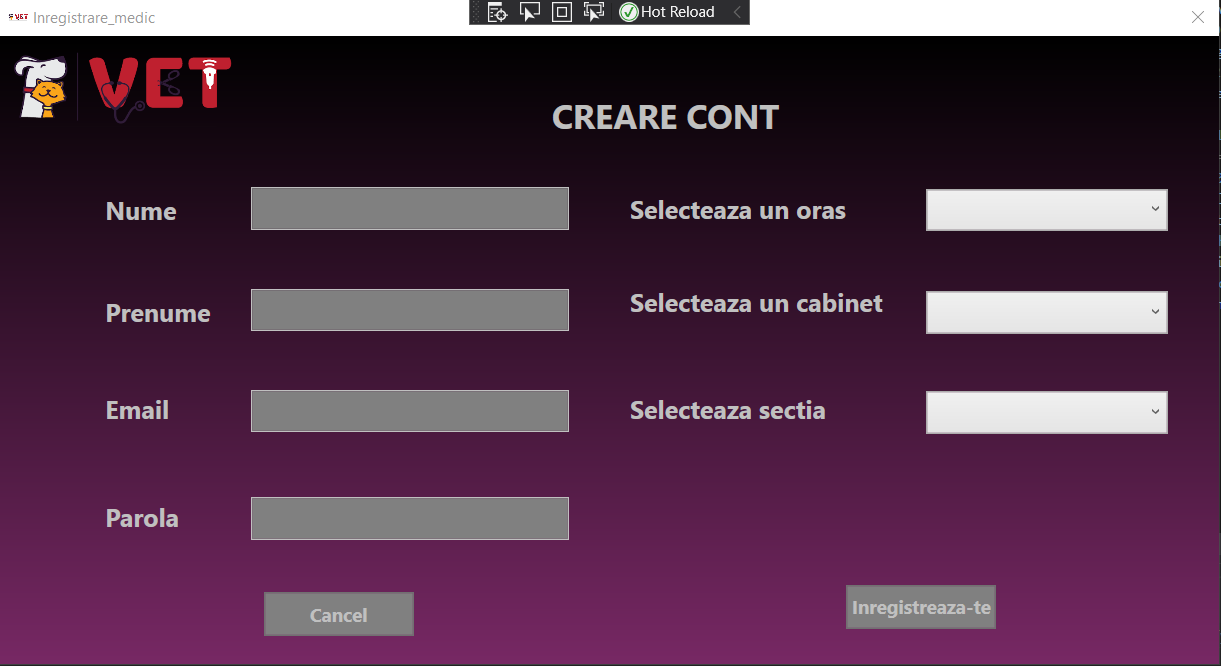
Aici utilizatorii introduc credentialele in functie de tipul selectat din combo box(medic, asistent, user normal). Cele 3 butoane vor crea noi ferestre.

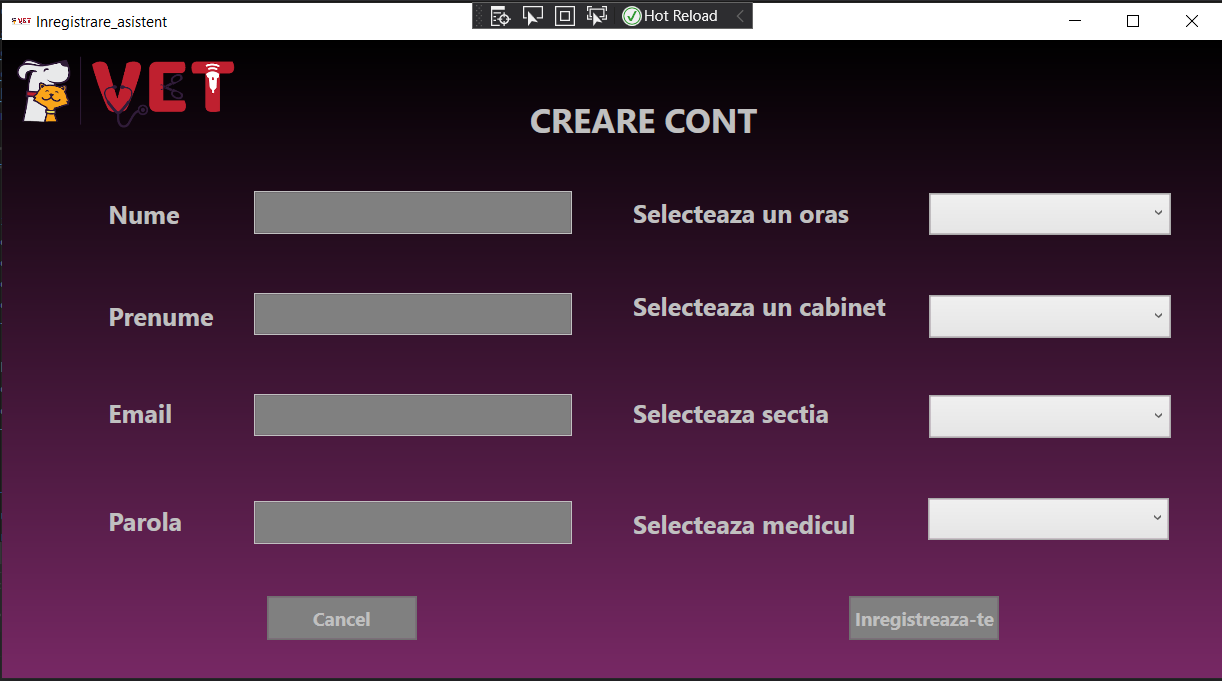
1. Serviciile noastre

In aceasta fereastra sunt prezentate orasele in care clinica veterinara HomeVet isi desfasoara activitatea. Dupa selectarea unui oras, vor aparea in datagrid informatii despre acel cabinet precum adresa, numarul de telefon si programul acestuia. Butonul „Cancel” va duce utilizatorul in pagina principala a aplicatiei.

# Creaza cont

In cazul in care un utilizator doreste sa isi creeze cont, acesta va accesa butonul „Creaza cont”. In functie de categoria din care, acest buton va crea ferestre de inregistrare pentru fiecare user in parte. Dupa completarea campurilor, valorile introduse vor fi introduse in baza de date, iar utilizatorul se poate conecta direct din pagina principala.

* Inregistrare Utilizator
* Inregistrare Medic
* Inregistrare Asistent



In plus parola este stocata in baza de date dupa ce este supusa unui proces de hash-ing cu alogrimtul sha256.

byte[] passwdBytes = Encoding.UTF8.GetBytes(parola);

using (var sha256 = SHA256.Create())

{

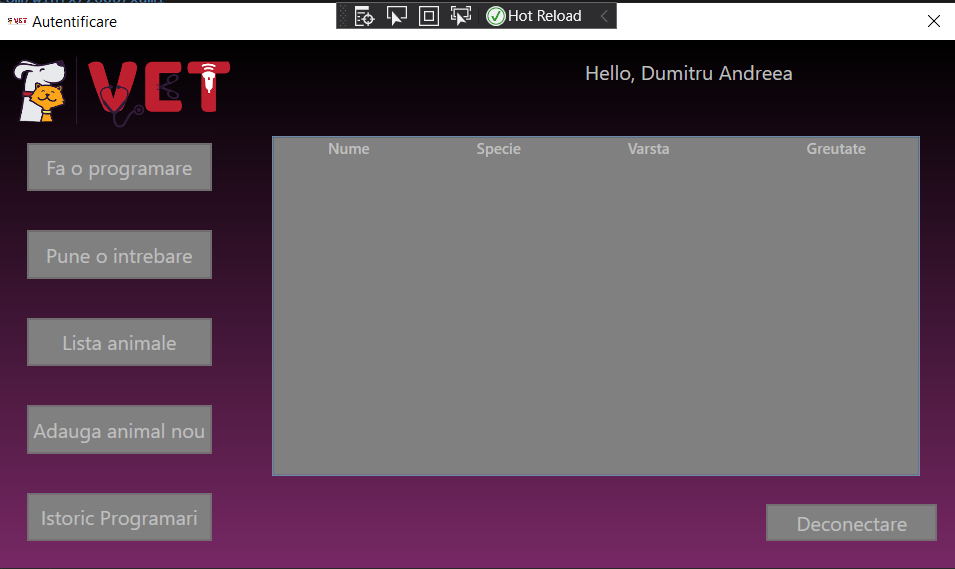
byte[] hash = sha256.ComputeHash(passwdBytes);

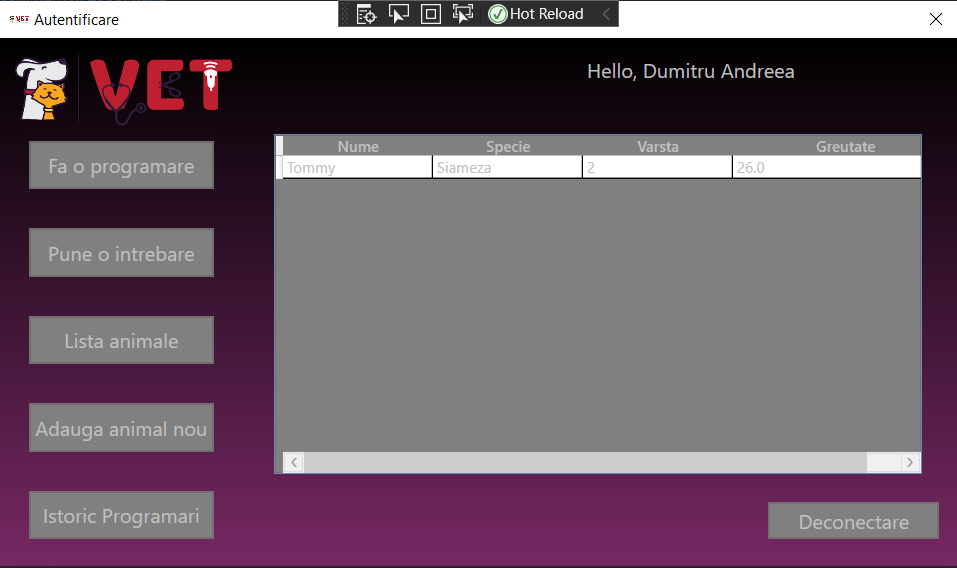
hashedPass = Encoding.UTF8.GetString(hash);

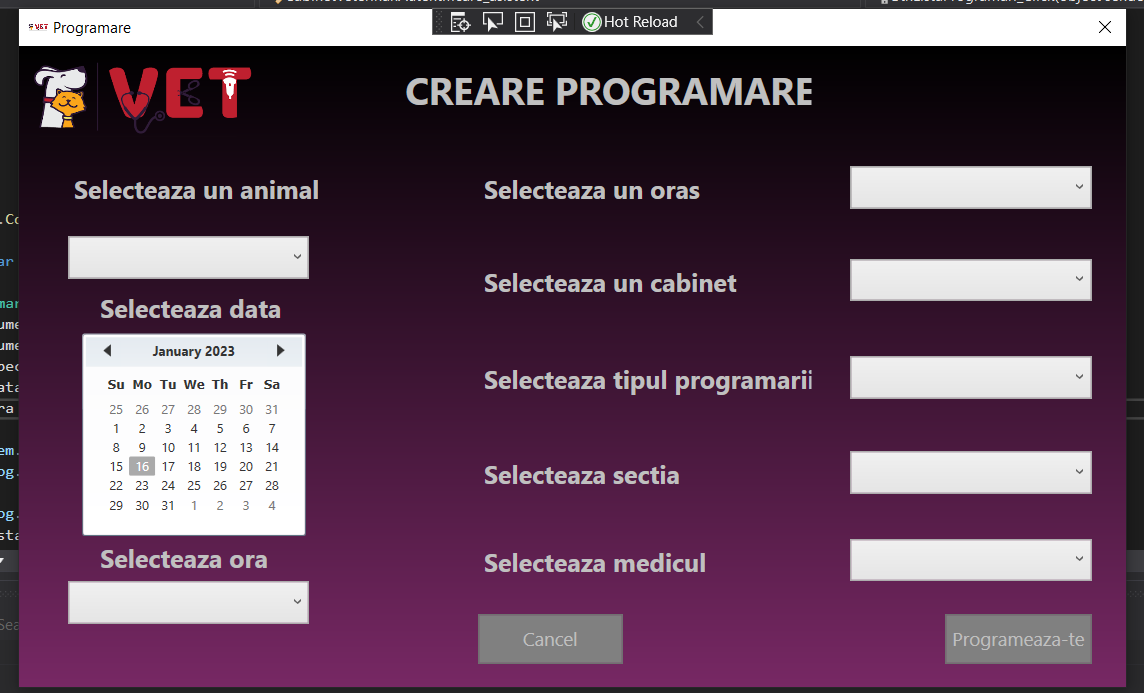
// Store the `storedPasswordHash` in your database

}

# Fereastra autentificare Utilizator Normal

In aceasta fereastra, utilizatorul poate realiza mai multe actiuni precum: crearea unei programari pentru animalele din lista sa, isi poate adauga noi animale in lista, poate vizualiza lista programarilor sau poate adresa intrebari medicilor.

* Lista animale
* Fa o programare



private void btnProgramare\_Click(object sender, RoutedEventArgs e)

{

if (cbMedic.SelectedItem!=null)

{

string tipProg = cbTipProgramare.SelectedItem.ToString();

string tip = "";

if (tipProg == "Urgenta")

tip = "D";

else if (tipProg == "Control")

tip = "N";

var newProg = new Programari()

{

idPacient = idA,

idMedic = getIdMedic(),

DataProgramare = calendar.SelectedDate.Value,

Ora = cbOra.SelectedItem.ToString(),

Tip = tip,

StatusProgramare = "Pending"

};

context.Programari.Add(newProg);

context.SaveChanges();

MessageBox.Show("Programare realizata cu succes!", "Succes", MessageBoxButton.OK, MessageBoxImage.Information);

this.Close();

}

}

Aceasta portiune din cod creaza o noua inregistrare pentru o programare facute de catre userul normal pe care o va introduce in baza de date.

public void LoadListaAnimale()

{

var animals = (from a in context.Pacienti

where a.idUtilizator == idUser

select a.Nume).Distinct();

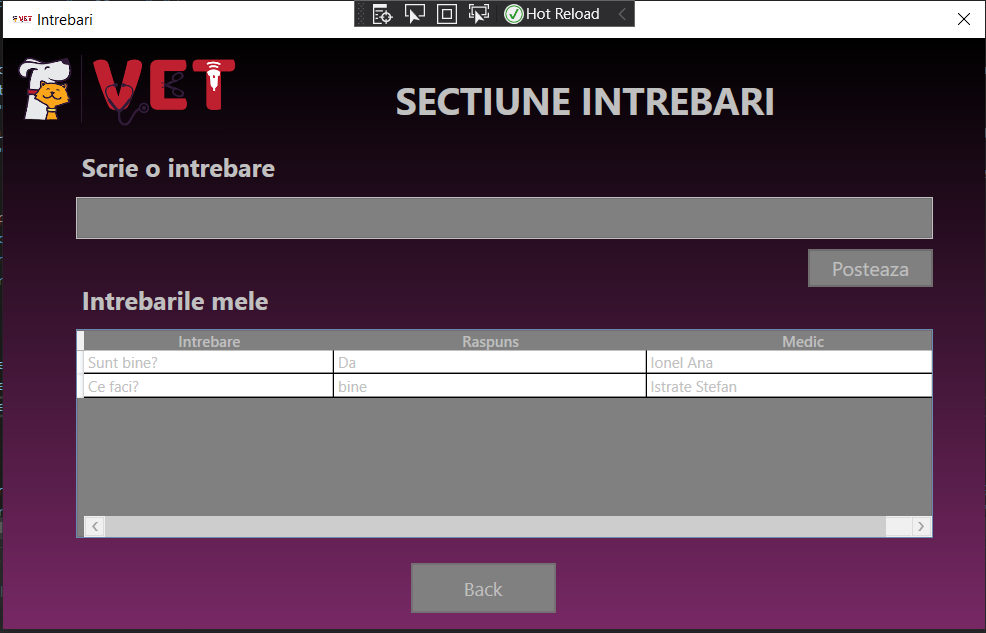
foreach (var c in animals)

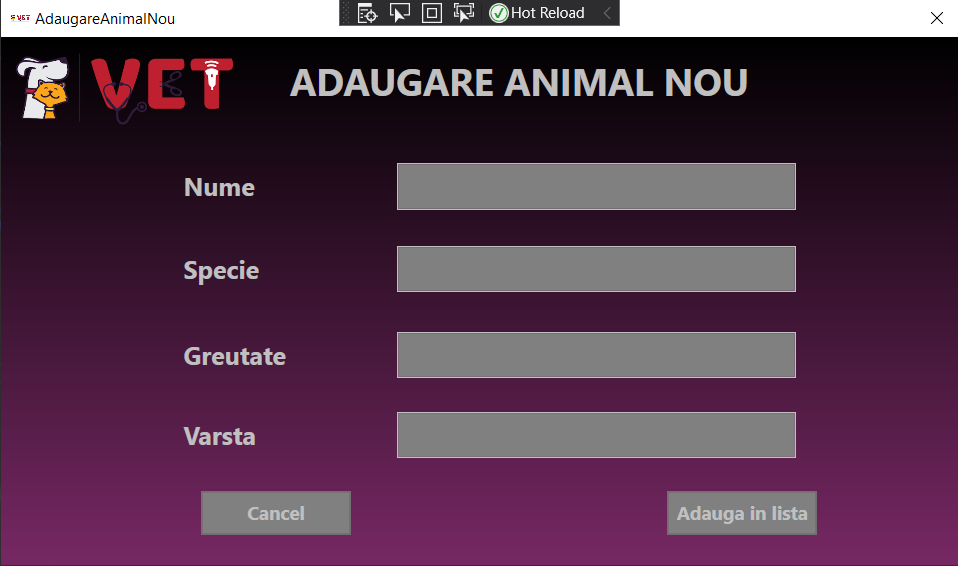
cbListaAnimale.Items.Add(c.ToString());

}

Aceasta functie selecteaza din baza de date toate animalele ce corespund utilizatorului respectiv pe care ulterior le va introduce intr-un combo box si le va afisa.

* Pune o intrebare



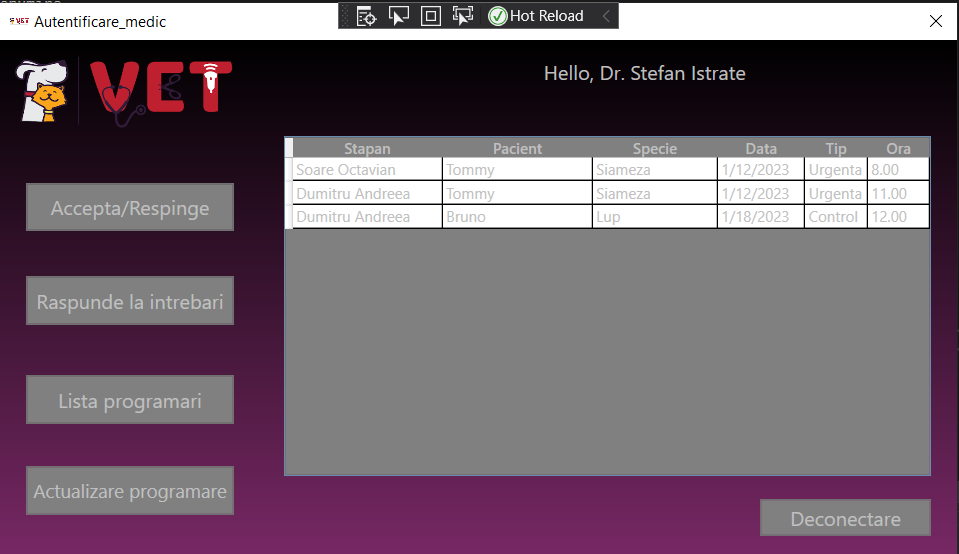
* Adauga animal nou
* Istoric programari

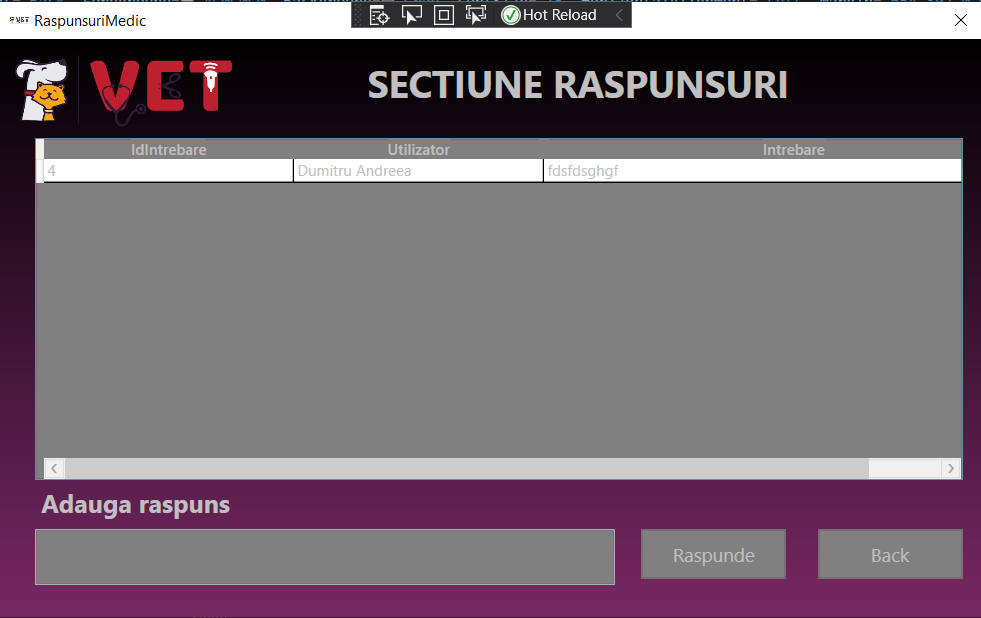


# Fereastra Autentificare Medic

In aceasta fereastra, medicul poate accepta sau respinge programarile create la cabinetul sau, poate raspunde la intrebarile puse de catre orice utilizator, isi poate vedea lista cu animalele programate la control sau poate actualiza o programare in cazul in care este necesara colectarea unor probe de catre asistentul pe care il va desemna in realizarea acestei actiuni fie acasa, fie la domiciliul pacientului.

* Lista programari



* Raspunde intrebari

private void btnRaspunde\_Click(object sender, RoutedEventArgs e)

{

if (ok == 1)

{

var intrebare = (from i in context.Intrebari

where i.idIntrebare == id

select i).Single();

intrebare.StatusIntrebare = "DA";

intrebare.MesajRaspuns = txtRaspuns.Text;

intrebare.idMedic = idMedic;

context.SaveChanges();

LoadIntrebari();

}

else

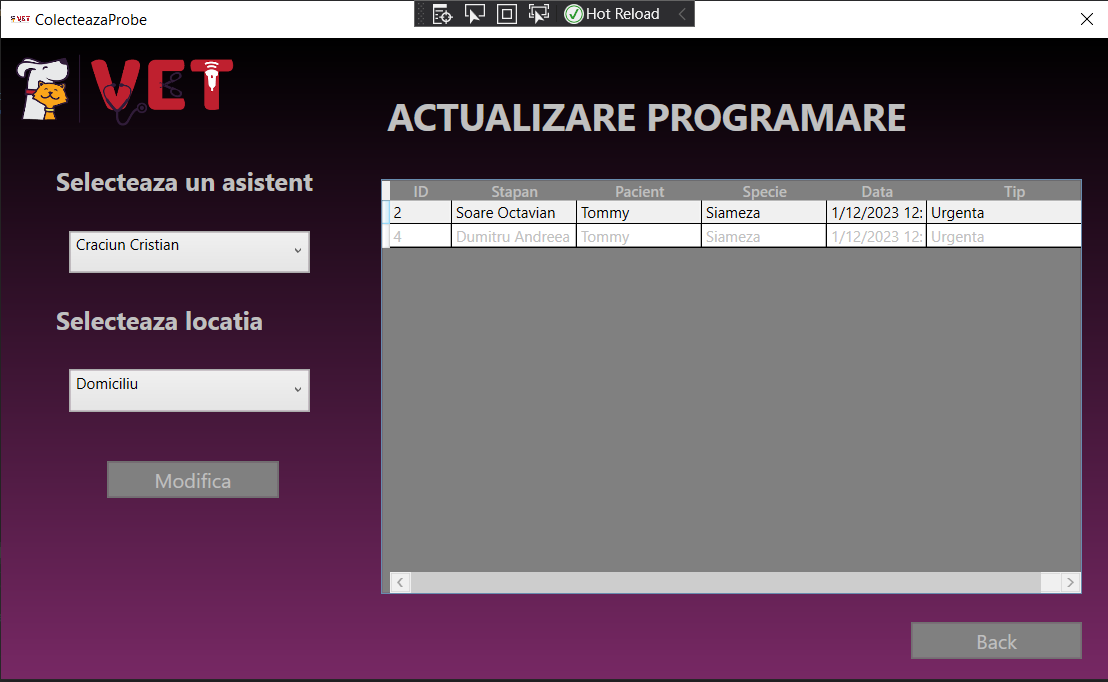
MessageBox.Show("Selecteaza intrebarea", "ERROR", MessageBoxButton.OK,MessageBoxImage.Error);

txtRaspuns.Clear();

}

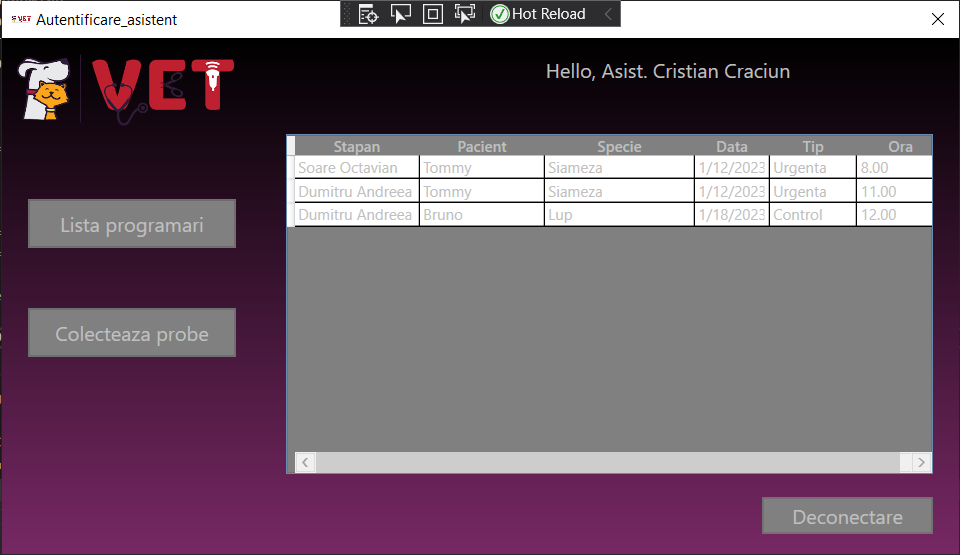
Aceasta functie actualizeaza in baza de date tabelul Intrebari, repectiv campurile status intrebare, mesaj raspuns si id medic in functie de intrebarea selectata.

* Actualizare programare

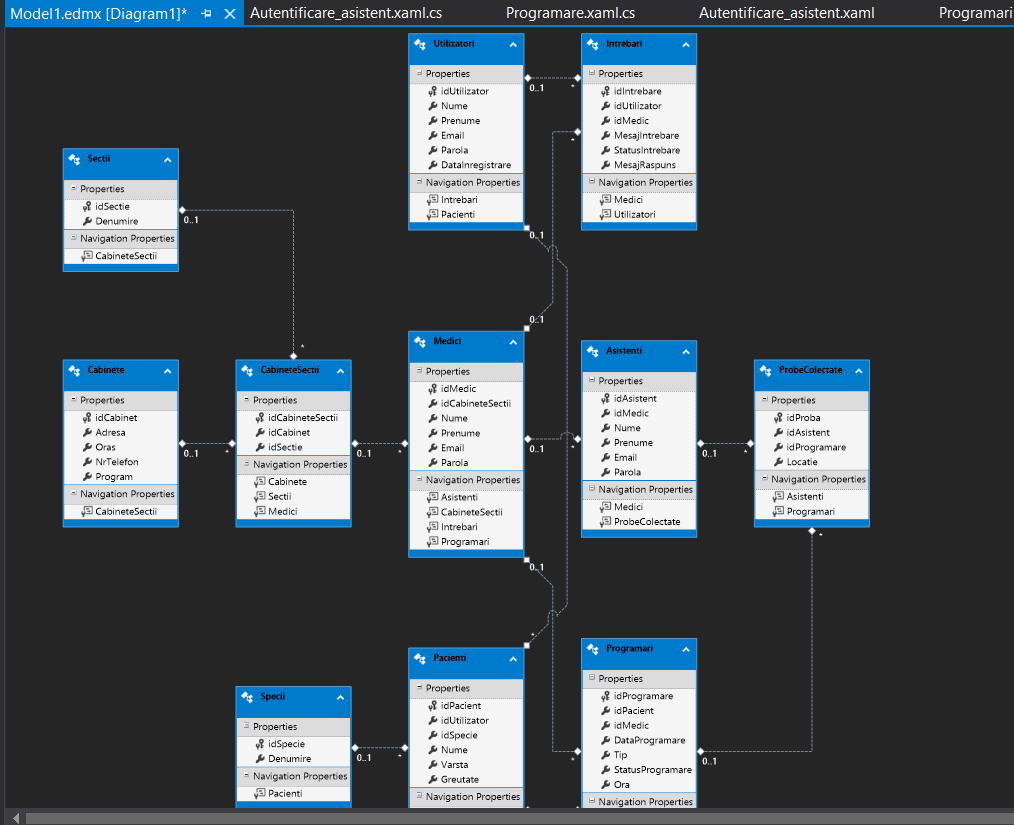


# Fereastra Asistent

In aceasta fereastra, asistentul poate vizualiza programarile de la medicul asupra caruia se afla in subordine si programarile asupra carora necesita sa colecteze probe.

* Lista programari + colecteaza probe

**3.2 Diagrama ER si Entity Framework**



**3.3 Concluzii**

In concluzie, aceasta aplicatie realizeaza functionalitatile de baza necesare gestionarii unui cabinet veterinar.