Pet Store

Pet Store

Management System

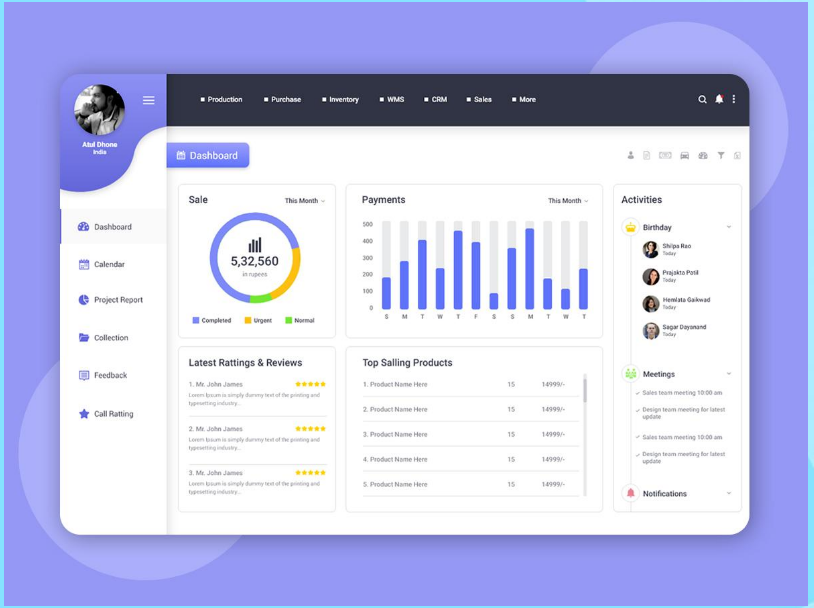
Emma De Barros

Raisa Stepanova-Timina

Elina Sardaryan

MS (Management System)

- Mix between CRM and EMS
- Computer software or application that uses a database to manage all content



Latest Rattings & Reviews

1. Mr. John James ★★★★★
Lorem Ipsum is simply dummy text of the printing and typesetting industry...

2. Mr. John James ★★★★★
Lorem Ipsum is simply dummy text of the printing and typesetting industry...

3. Mr. John James ★★★★★
Lorem Ipsum is simply dummy text of the printing and typesetting industry...

Top Salling Products

1. Product Name Here	15	14999/-
2. Product Name Here	15	14999/-
3. Product Name Here	15	14999/-
4. Product Name Here	15	14999/-
5. Product Name Here	15	14999/-

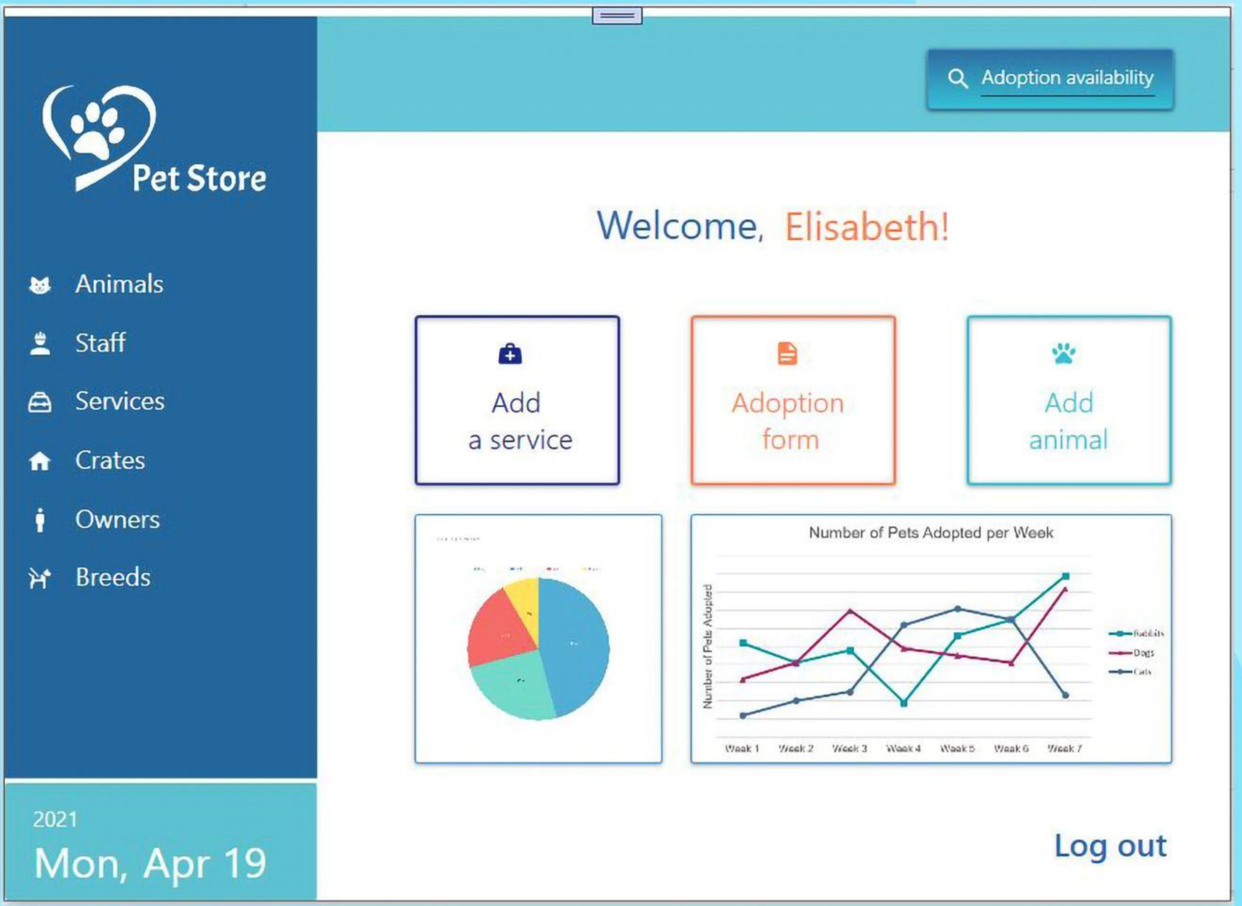
Shilpa Rao
Today

Prajakta Patil
Today

Hemlata Gaikwad
Today

Sagar Dayanand
Today

- ✓ Sales team meeting 10:00 am
- ✓ Design team meeting for latest update
- ✓ Sales team meeting 10:00 am
- ✓ Design team meeting for latest update

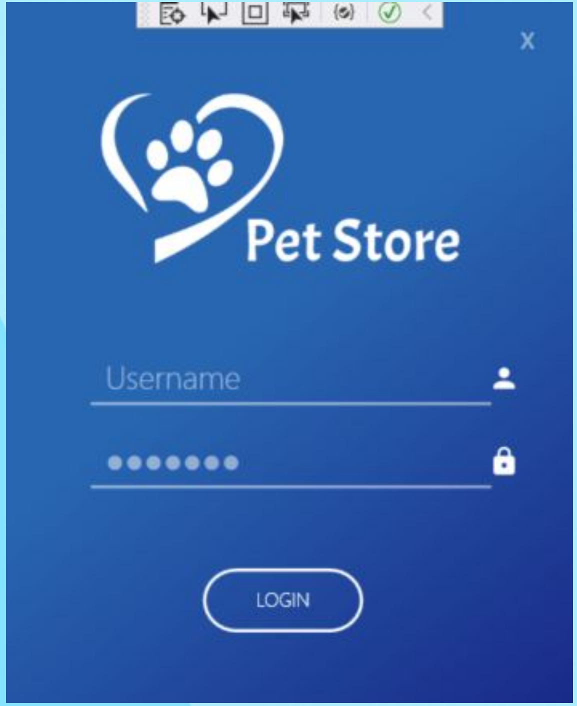


Purpose of Pet store MS is to give user a possibility to comfortably manipulate data related to animals.

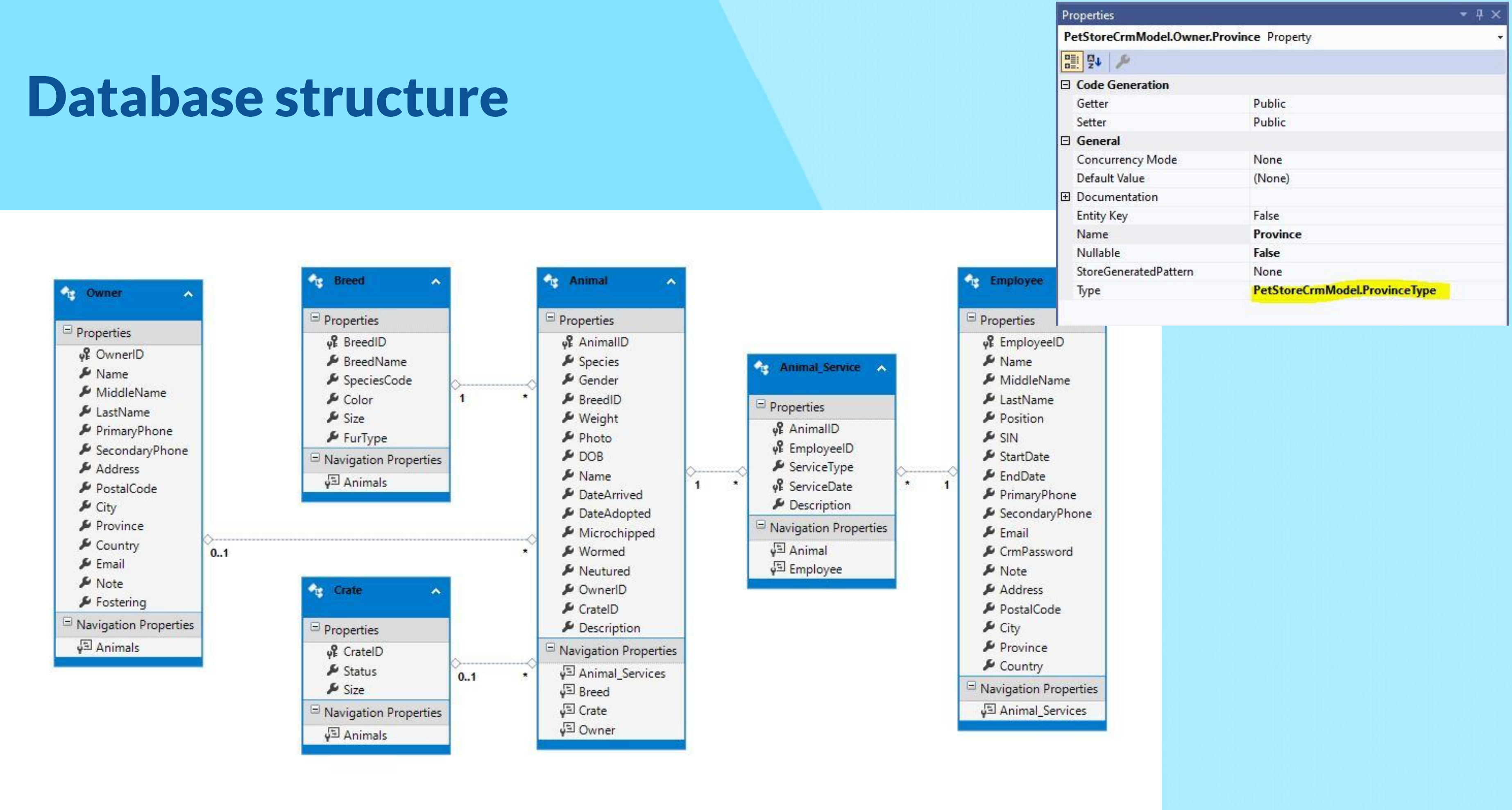
Current system was designed for shelters, however can be used for pet stores, and vet clinics as well.

Features

1. Create,Read, Update, Delete (CRUD)
    - animals
    - owners
    - services
    - employees
    - crates
2. Check availability of crates for newly added animals
3. Check adoption availability
4. Sort data
5. Export to csv (list of owners)
6. Print out adoption form + save to pdf format



Database structure



Owner

Properties
- OwnerID
- Name
- MiddleName
- LastName
- PrimaryPhone
- SecondaryPhone
- Address
- PostalCode
- City
- Province
- Country
- Email
- Note
- Fostering

Navigation Properties
- Animals

Breed

Properties
- BreedID
- BreedName
- SpeciesCode
- Color
- Size
- FurType

Navigation Properties
- Animals

Crate

Properties
- CrateID
- Status
- Size

Navigation Properties
- Animals

Animal

Properties
- AnimalID
- Species
- Gender
- BreedID
- Weight
- Photo
- DOB
- Name
- DateArrived
- DateAdopted
- Microchipped
- Wormed
- Neutured
- OwnerID
- CrateID
- Description

Navigation Properties
- Animal_Services
- Breed
- Crate
- Owner

Animal_Service

Properties
- AnimalID
- EmployeeID
- ServiceType
- ServiceDate
- Description

Navigation Properties
- Animal
- Employee

Employee

Properties
- EmployeeID
- Name
- MiddleName
- LastName
- Position
- SIN
- StartDate
- EndDate
- PrimaryPhone
- SecondaryPhone
- Email
- CrmPassword
- Note
- Address
- PostalCode
- City
- Province
- Country

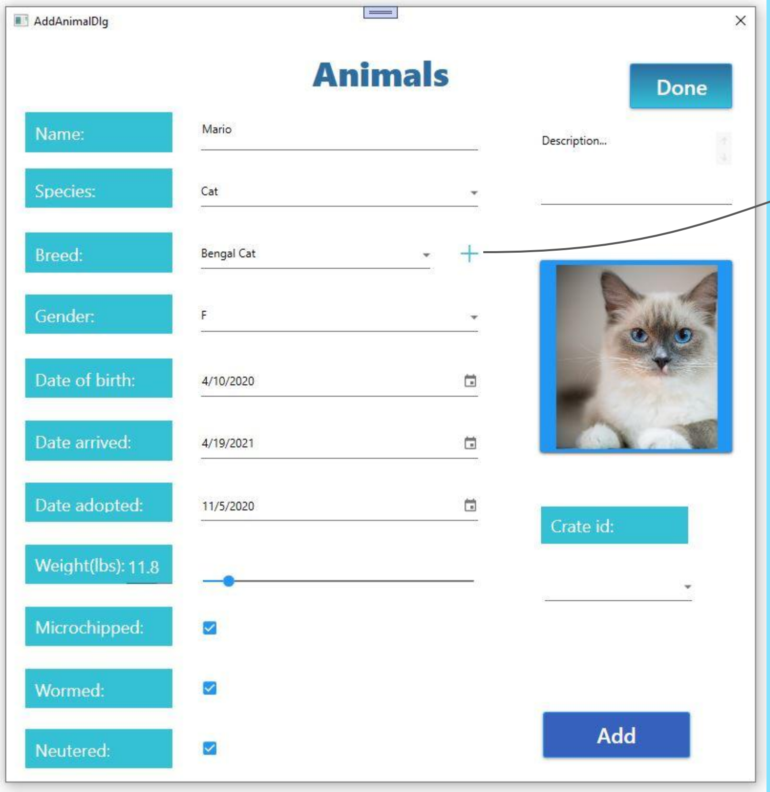
Navigation Properties
- Animal_Services

Properties

PetStoreCrmModel.Owner.Province Property

Code Generation	
Getter	Public
Setter	Public
General	
Concurrency Mode	None
Default Value	(None)
Documentation	
Entity Key	False
Name	**Province**
Nullable	**False**
StoreGeneratedPattern	None
Type	**PetStoreCrmModel.ProvinceType**

Adding a new breed name while adding an animal



AddAnimalDlg

Animals

Name: Mario

Species: Cat

Breed: Bengal Cat

Gender: F

Date of birth: 4/10/2020

Date arrived: 4/19/2021

Date adopted: 11/5/2020

Weight(lbs): 11.8

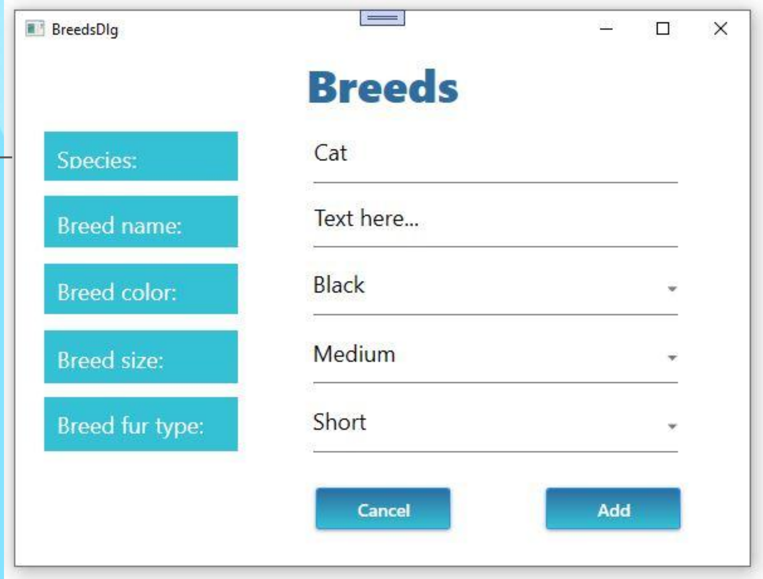
Microchipped: ☑

Wormed: ☑

Neutered: ☑

Description...

Crate id:



BreedsDlg

Breeds

Species: Cat

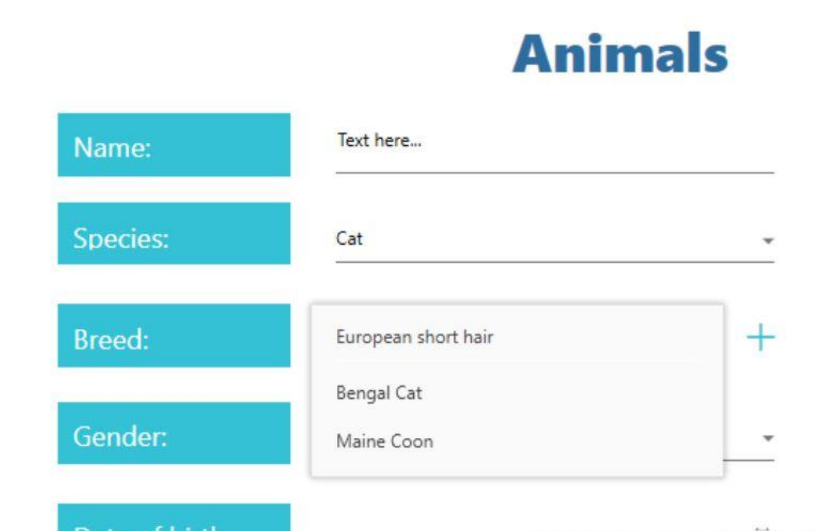
Breed name: Text here...

Breed color: Black

Breed size: Medium

Breed fur type: Short

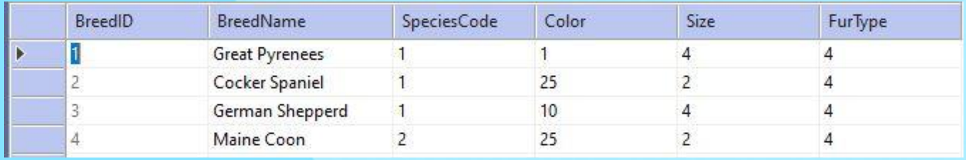
Challenges and solutions



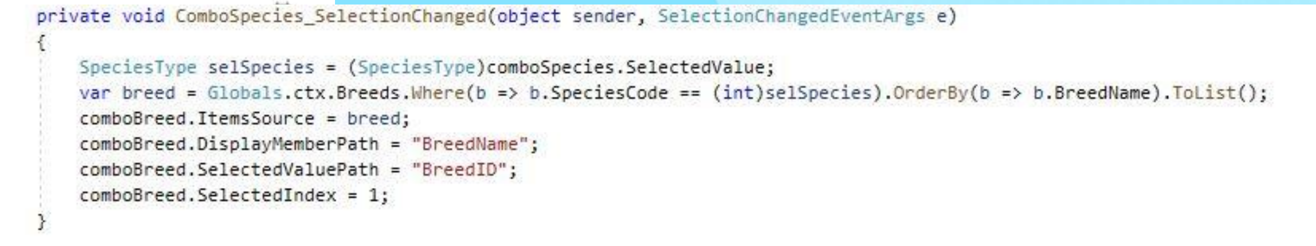
Animals

Name:	Text here...
Species:	Cat
Breed:	European short hair Bengal Cat Maine Coon
Gender:	
Date of birth:	

Display only the breed names of the same species type



BreedID	BreedName	SpeciesCode	Color	Size	FurType
1	Great Pyrenees	1	1	4	4
2	Cocker Spaniel	1	25	2	4
3	German Shepperd	1	10	4	4
4	Maine Coon	2	25	2	4

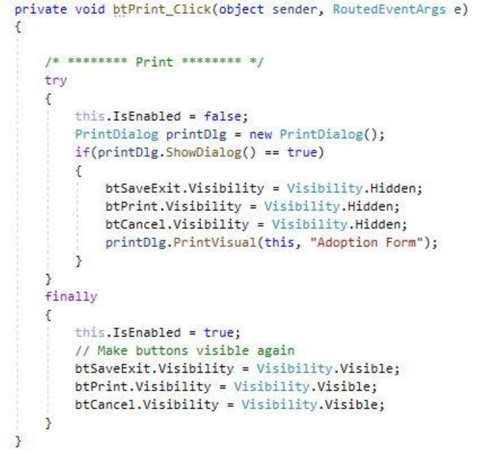


```
private void ComboSpecies_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    SpeciesType selSpecies = (SpeciesType)comboSpecies.SelectedValue;
    var breed = Globals.ctx.Breeds.Where(b => b.SpeciesCode == (int)selSpecies).OrderBy(b => b.BreedName).ToList();
    comboBreed.ItemsSource = breed;
    comboBreed.DisplayMemberPath = "BreedName";
    comboBreed.SelectedValuePath = "BreedID";
    comboBreed.SelectedIndex = 1;
}
```

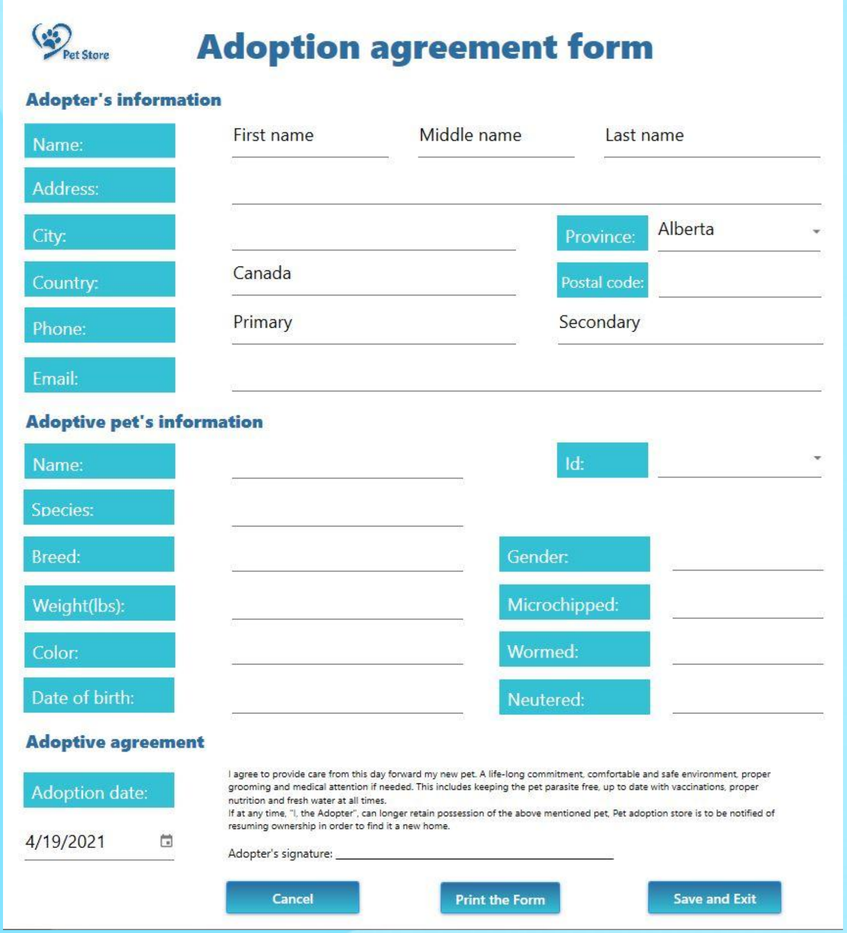
Adoption agreement

Fill out adoption form

- Add a new owner (Owners table)
- Add ownerID (Animals table)
- Add date of adoption (Animals table)
- Delete crate (Animals table)
- Change the status of crate(Crate table)



```
private void btPrint_Click(object sender, RoutedEventArgs e)
{

    /* ******** Print ******** */
    try
    {
        this.IsEnabled = false;
        PrintDialog printDlg = new PrintDialog();
        if(printDlg.ShowDialog() == true)
        {
            btSaveExit.Visibility = Visibility.Hidden;
            btPrint.Visibility = Visibility.Hidden;
            btCancel.Visibility = Visibility.Hidden;
            printDlg.PrintVisual(this, "Adoption Form");
        }
    }
    finally
    {
        this.IsEnabled = true;
        // Make buttons visible again
        btSaveExit.Visibility = Visibility.Visible;
        btPrint.Visibility = Visibility.Visible;
        btCancel.Visibility = Visibility.Visible;
    }
}
```



Adoption agreement form

Adopter's information

Name: First name ______ Middle name ______ Last name ______

Address: ______

City: ______ Province: Alberta

Country: Canada Postal code: ______

Phone: Primary ______ Secondary ______

Email: ______

Adoptive pet's information

Name: ______ Id: ______

Species: ______

Breed: ______ Gender: ______

Weight(lbs): ______ Microchipped: ______

Color: ______ Wormed: ______

Date of birth: ______ Neutered: ______

Adoptive agreement

Adoption date: 4/19/2021

I agree to provide care from this day forward my new pet. A life-long commitment, comfortable and safe environment, proper grooming and medical attention if needed. This includes keeping the pet parasite free, up to date with vaccinations, proper nutrition and fresh water at all times.

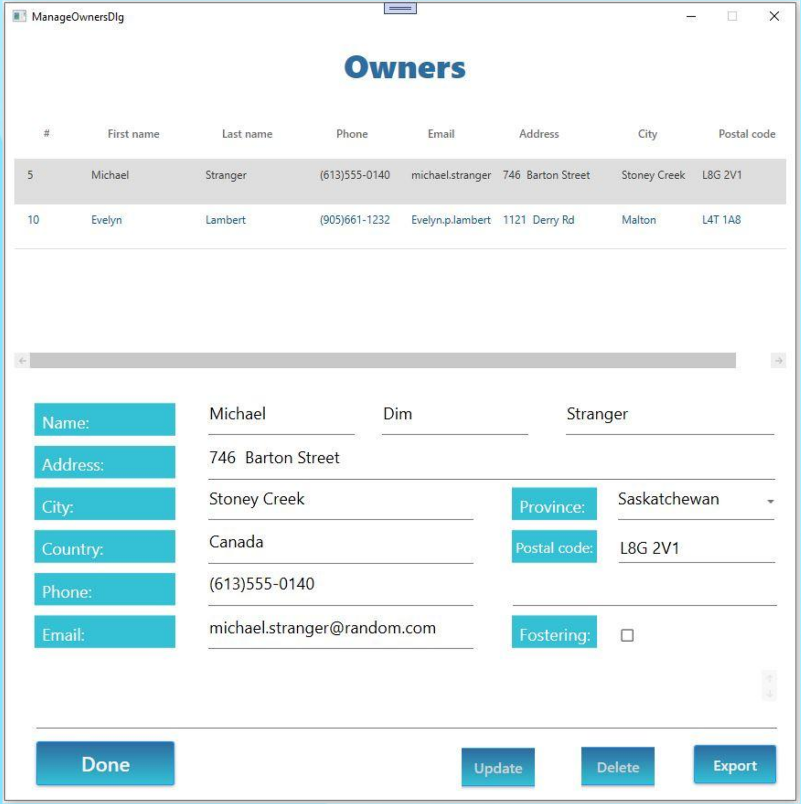
If at any time, "I, the Adopter", can longer retain possession of the above mentioned pet, Pet adoption store is to be notified of resuming ownership in order to find it a new home.

Adopter's signature: ______

Challenges and solutions

Delete owner

- First, delete owner id from Animals table(also adoption date)
- Assign a crate to the animal
- Change status of crate to occupied/true
- Only then delete the owner



Owners

#	First name	Last name	Phone	Email	Address	City	Postal code
5	Michael	Stranger	(613)555-0140	michael.stranger	746 Barton Street	Stoney Creek	L8G 2V1
10	Evelyn	Lambert	(905)661-1232	Evelyn.p.lambert	1121 Derry Rd	Malton	L4T 1A8

Name: Michael Dim Stranger

Address: 746 Barton Street

City: Stoney Creek Province: Saskatchewan

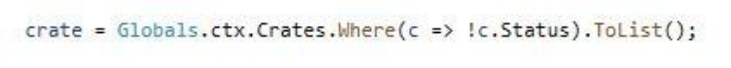
Country: Canada Postal code: L8G 2V1

Phone: (613)555-0140

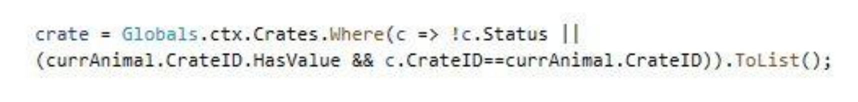
Email: michael.stranger@random.com Fostering: □

What I learned

Add VS Update Animals

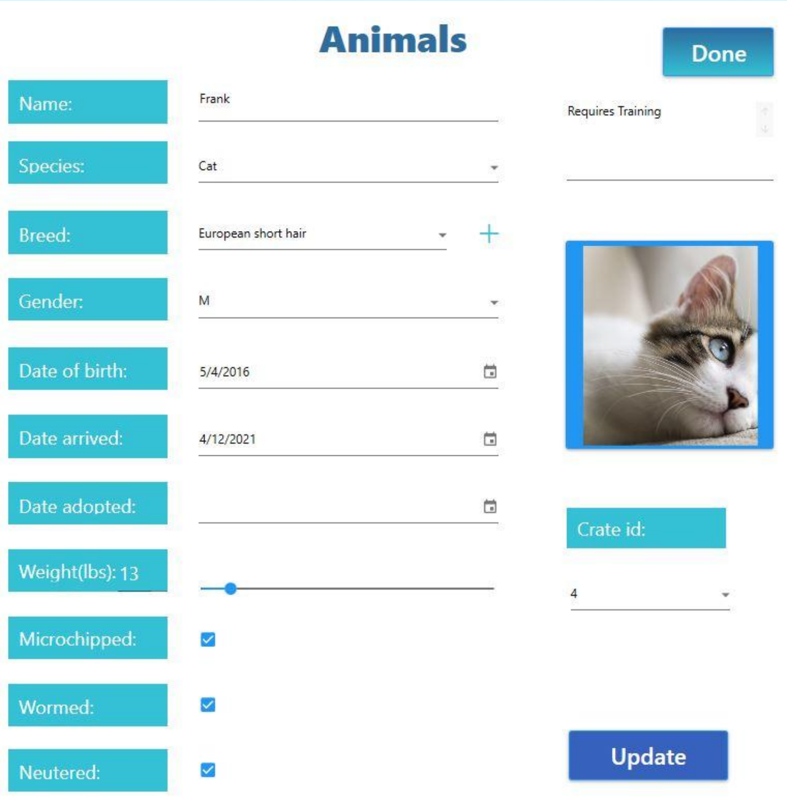


```
crate = Globals.ctx.Crates.Where(c => !c.Status).ToList();
```



```
crate = Globals.ctx.Crates.Where(c => !c.Status ||
(currAnimal.CrateID.HasValue && c.CrateID==currAnimal.CrateID)).ToList();
```

- Modify CrateID in Animals table -- > modify crate Status in Crates table



Animals

Name: Frank

Species: Cat

Breed: European short hair

Gender: M

Date of birth: 5/4/2016

Date arrived: 4/12/2021

Date adopted:

Weight(lbs): 13

Microchipped: ☑

Wormed: ☑

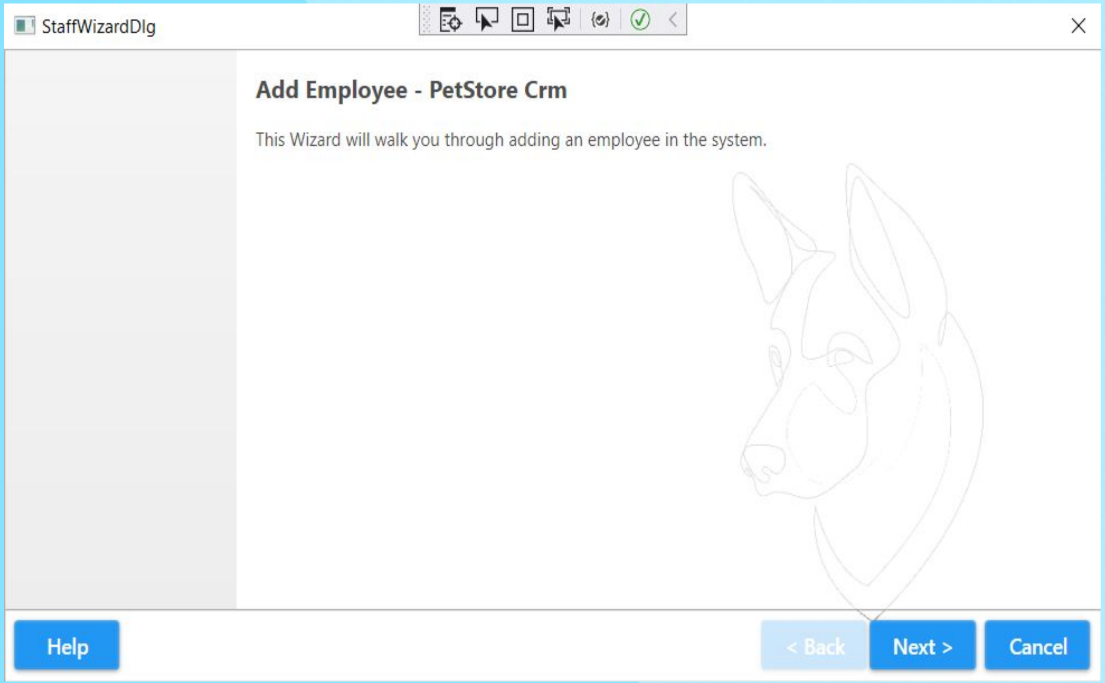
Neutered: ☑

Requires Training

Crate id: 4

Technologies Used:

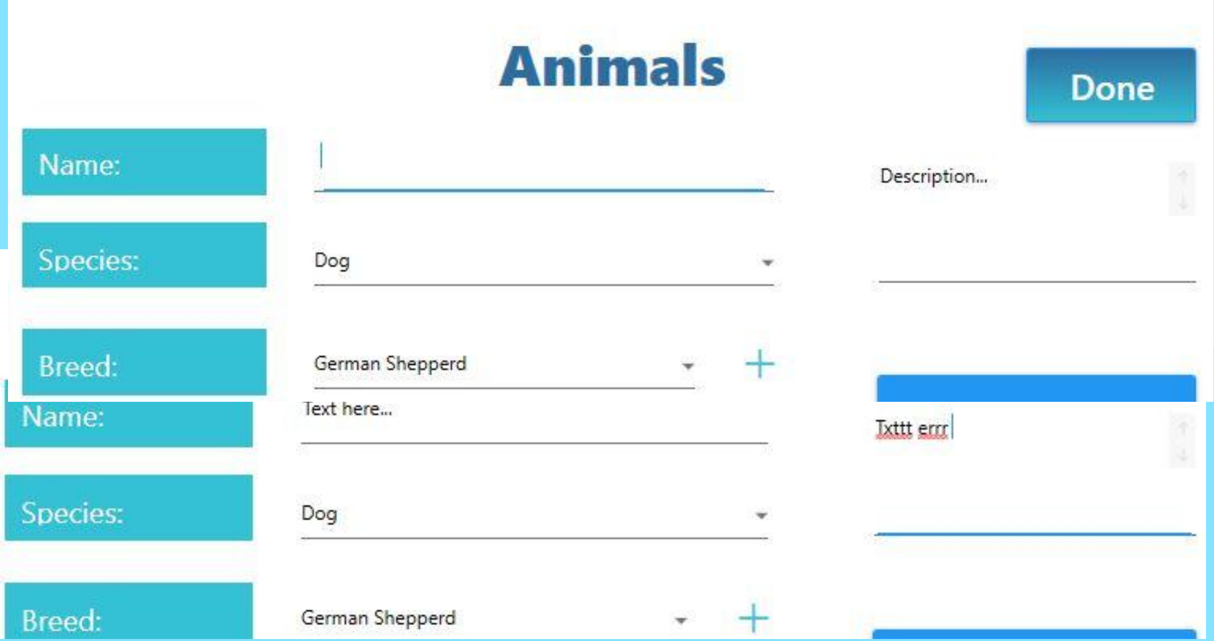
- Wizard with Extended WPF ToolKit
- Material Design
- Print - PDF
- LINQ
- CSV Helper
- DataGrid



Add Employee - PetStore Crm

This Wizard will walk you through adding an employee in the system.

Small challenges and solutions



Name:

Species: Dog

Breed: German Shepperd

Description...

Name: Text here...

Species: Dog

Breed: German Shepperd

Txttt errr

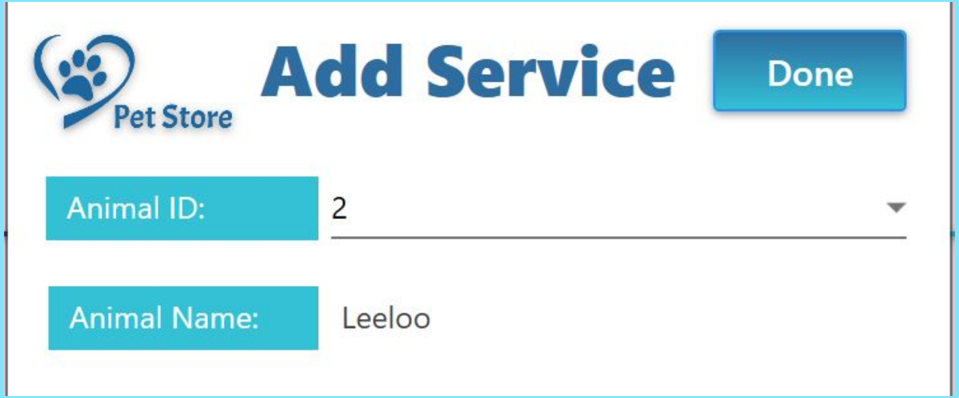
GotFocus=
"tbDescription_GotFocus"

LostFocus=
"tbDescription_LostFocus"

SpellCheck.IsEnabled=
"True"

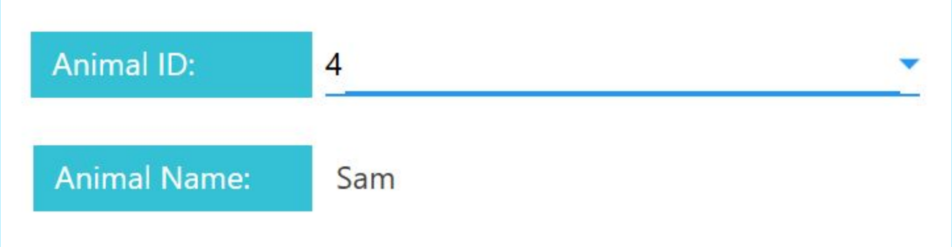
AcceptsReturn="True"

Combobox



Animal ID: 2

Animal Name: Leeloo

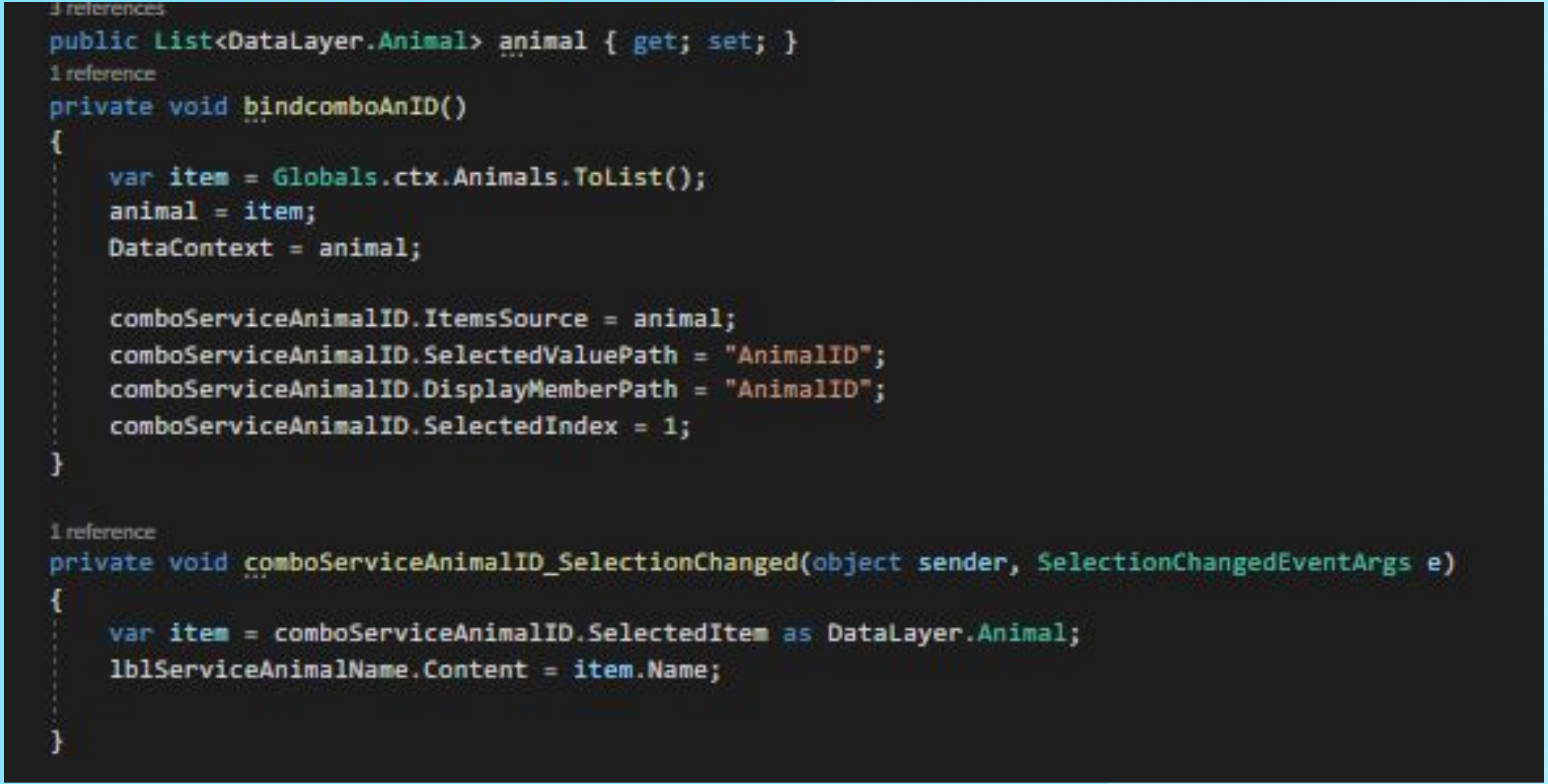


Animal ID: 4

Animal Name: Sam

Animal name changes accordingly to its ID

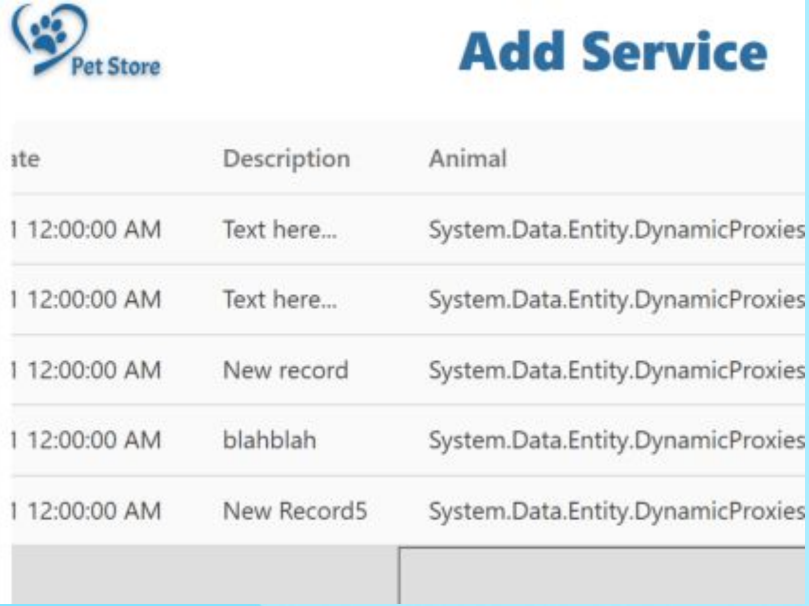
Challenge #1 Combobox



```
3 references
public List<DataLayer.Animal> animal { get; set; }
1 reference
private void bindcomboAnID()
{
    var item = Globals.ctx.Animals.ToList();
    animal = item;
    DataContext = animal;

    comboServiceAnimalID.ItemsSource = animal;
    comboServiceAnimalID.SelectedValuePath = "AnimalID";
    comboServiceAnimalID.DisplayMemberPath = "AnimalID";
    comboServiceAnimalID.SelectedIndex = 1;
}

1 reference
private void comboServiceAnimalID_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    var item = comboServiceAnimalID.SelectedItem as DataLayer.Animal;
    lblServiceAnimalName.Content = item.Name;

}
```

Challenge #1 DataGrid

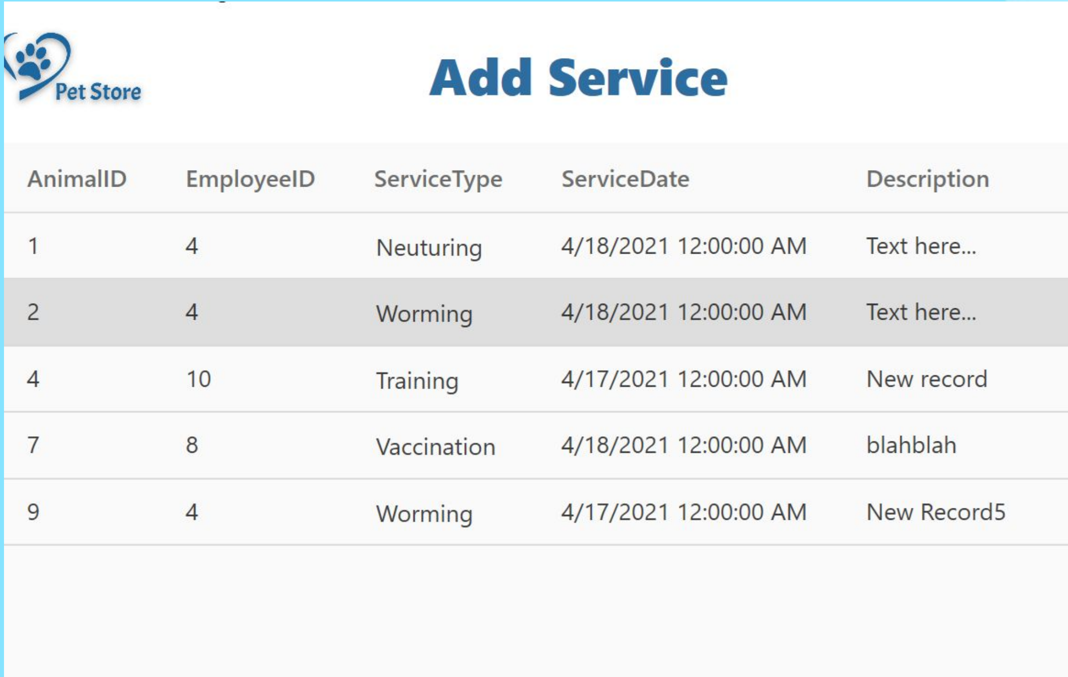
```
var anSer = from a
in Globals.ctx.Animal_Services
                    select a;
```



Add Service

ate	Description	Animal
1 12:00:00 AM	Text here...	System.Data.Entity.DynamicProxies
1 12:00:00 AM	Text here...	System.Data.Entity.DynamicProxies
1 12:00:00 AM	New record	System.Data.Entity.DynamicProxies
1 12:00:00 AM	blahblah	System.Data.Entity.DynamicProxies
1 12:00:00 AM	New Record5	System.Data.Entity.DynamicProxies

Solution

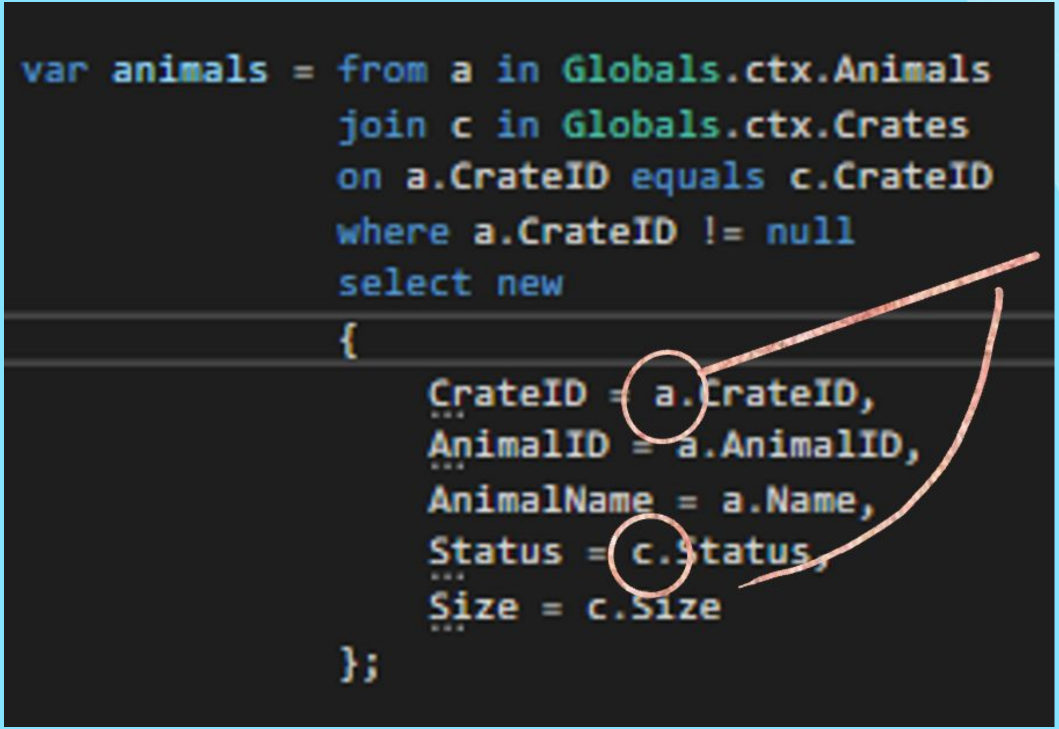


Add Service

AnimalID	EmployeeID	ServiceType	ServiceDate	Description
1	4	Neutering	4/18/2021 12:00:00 AM	Text here...
2	4	Worming	4/18/2021 12:00:00 AM	Text here...
4	10	Training	4/17/2021 12:00:00 AM	New record
7	8	Vaccination	4/18/2021 12:00:00 AM	blahblah
9	4	Worming	4/17/2021 12:00:00 AM	New Record5

```
var anSer = from a in
Globals.ctx.Animal_Services

        select new

        {

            AnimalID = a.AnimalID,

            EmployeeID = a.EmployeeID,

            ServiceType = a.ServiceType,

            ServiceDate = a.ServiceDate,

            Description = a.Description

        };
```

Join

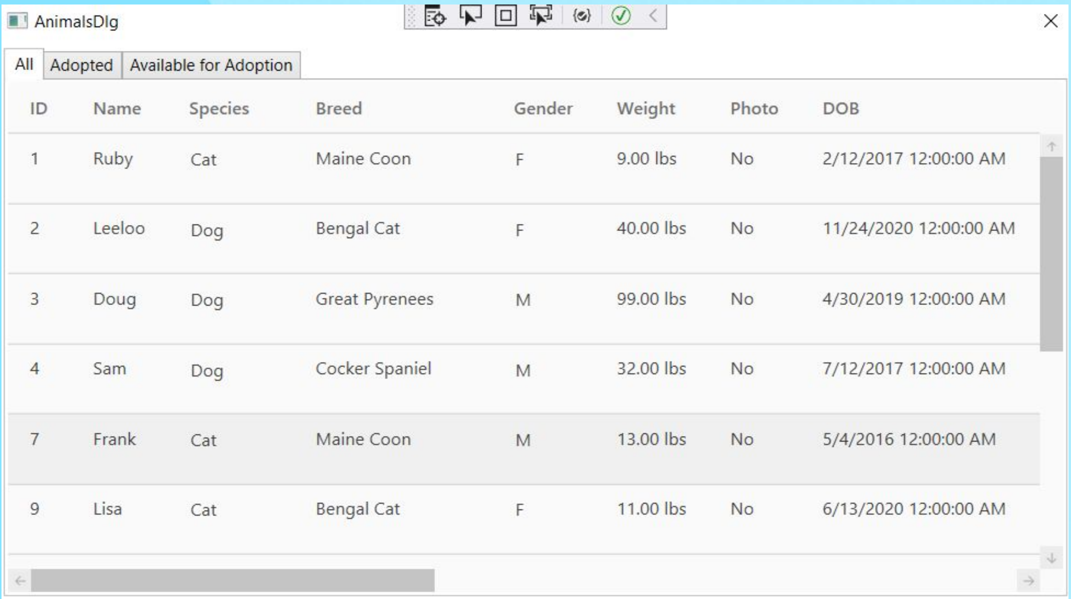


```
var animals = from a in Globals.ctx.Animals
              join c in Globals.ctx.Crates
              on a.CrateID equals c.CrateID
              where a.CrateID != null
              select new
              {
                  CrateID = a.CrateID,
                  AnimalID = a.AnimalID,
                  AnimalName = a.Name,
                  Status = c.Status,
                  Size = c.Size
              };
```

Presents information in a very neat way combining data from different tables

Challenge - Anonymous Types 🤯

- DataGrid in with TabControl
- != ListView
- ReadOnly property
- Display data from other entity (Owner's name e.g.)

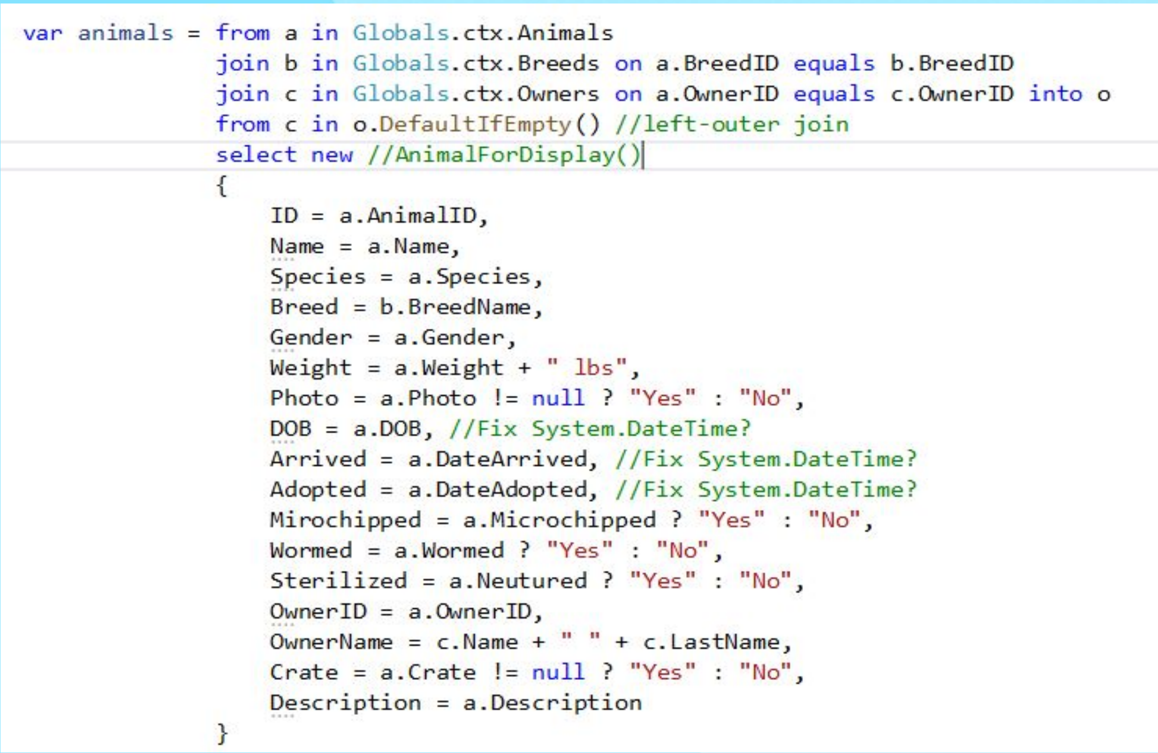


ID	Name	Species	Breed	Gender	Weight	Photo	DOB
1	Ruby	Cat	Maine Coon	F	9.00 lbs	No	2/12/2017 12:00:00 AM
2	Leeloo	Dog	Bengal Cat	F	40.00 lbs	No	11/24/2020 12:00:00 AM
3	Doug	Dog	Great Pyrenees	M	99.00 lbs	No	4/30/2019 12:00:00 AM
4	Sam	Dog	Cocker Spaniel	M	32.00 lbs	No	7/12/2017 12:00:00 AM
7	Frank	Cat	Maine Coon	M	13.00 lbs	No	5/4/2016 12:00:00 AM
9	Lisa	Cat	Bengal Cat	F	11.00 lbs	No	6/13/2020 12:00:00 AM

LINQ

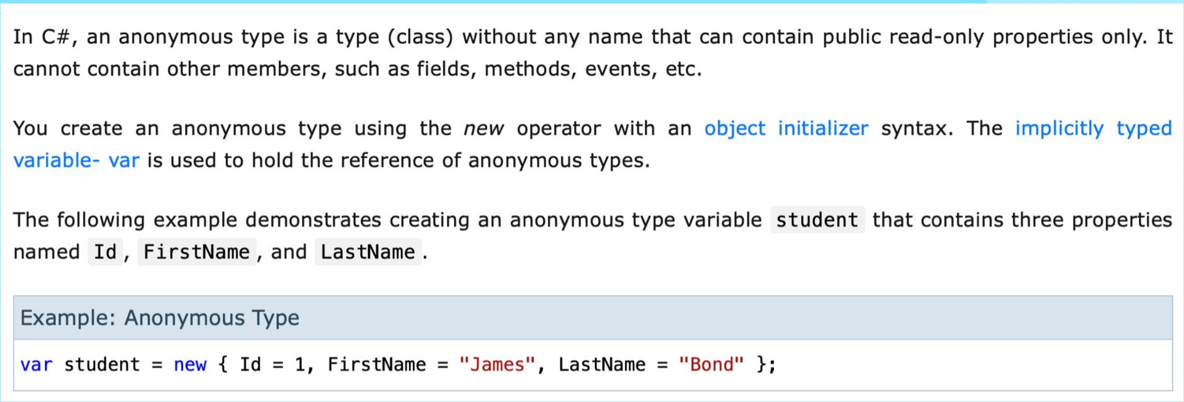
- Access data from SelectedItem != ListView because of anonymous types!

  e.g. (Animal)grid.SelectedItem;



```
var animals = from a in Globals.ctx.Animals
              join b in Globals.ctx.Breeds on a.BreedID equals b.BreedID
              join c in Globals.ctx.Owners on a.OwnerID equals c.OwnerID into o
              from c in o.DefaultIfEmpty() //left-outer join
              select new //AnimalForDisplay()
              {
                  ID = a.AnimalID,
                  Name = a.Name,
                  Species = a.Species,
                  Breed = b.BreedName,
                  Gender = a.Gender,
                  Weight = a.Weight + " lbs",
                  Photo = a.Photo != null ? "Yes" : "No",
                  DOB = a.DOB, //Fix System.DateTime?
                  Arrived = a.DateArrived, //Fix System.DateTime?
                  Adopted = a.DateAdopted, //Fix System.DateTime?
                  Mirochipped = a.Microchipped ? "Yes" : "No",
                  Wormed = a.Wormed ? "Yes" : "No",
                  Sterilized = a.Neutured ? "Yes" : "No",
                  OwnerID = a.OwnerID,
                  OwnerName = c.Name + " " + c.LastName,
                  Crate = a.Crate != null ? "Yes" : "No",
                  Description = a.Description
              }
```

But… what are anonymous types??? 🤔



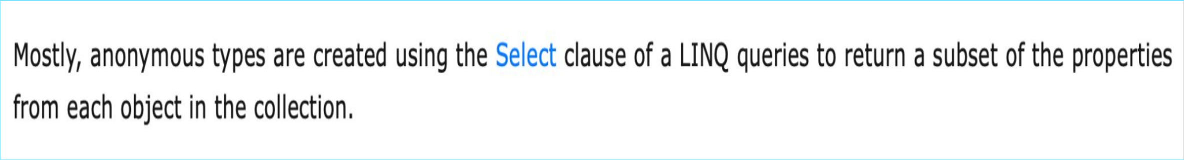
In C#, an anonymous type is a type (class) without any name that can contain public read-only properties only. It cannot contain other members, such as fields, methods, events, etc.

You create an anonymous type using the *new* operator with an object initializer syntax. The implicitly typed variable- var is used to hold the reference of anonymous types.

The following example demonstrates creating an anonymous type variable `student` that contains three properties named `Id`, `FirstName`, and `LastName`.

Example: Anonymous Type

```
var student = new { Id = 1, FirstName = "James", LastName = "Bond" };
```

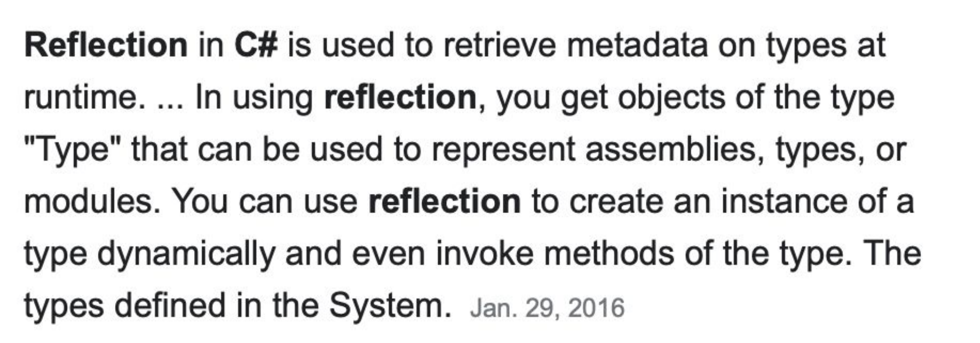


Mostly, anonymous types are created using the Select clause of a LINQ queries to return a subset of the properties from each object in the collection.

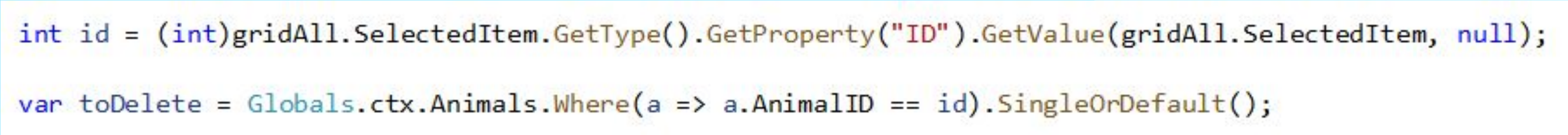
- Properties of AT are read only
- LINQ with anonymous types
- e.g. selecting AnimalID and turning it into ID
- Couldn’t read properties with the dot notation....

After a lot of research... 🥵

- Reflection OR create a display class
- AT derived from System.Object class → compiler generates a class with auto-generated name and applies appropriate type to each property based on the value expression
- GetType() method to see the name

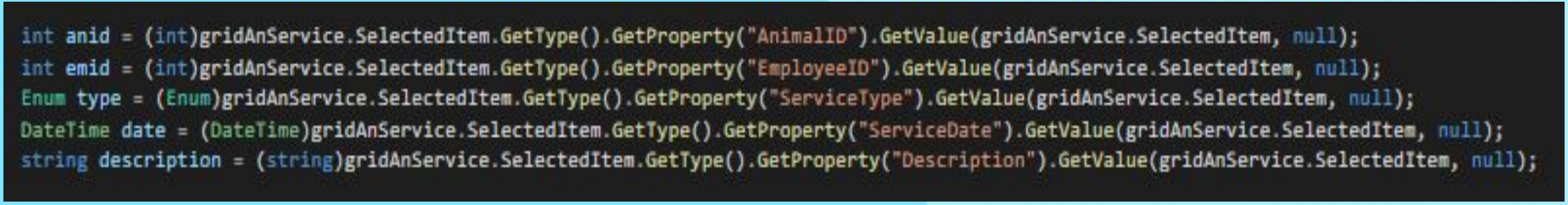


Reflection in **C#** is used to retrieve metadata on types at runtime. ... In using **reflection**, you get objects of the type "Type" that can be used to represent assemblies, types, or modules. You can use **reflection** to create an instance of a type dynamically and even invoke methods of the type. The types defined in the System. Jan. 29, 2016

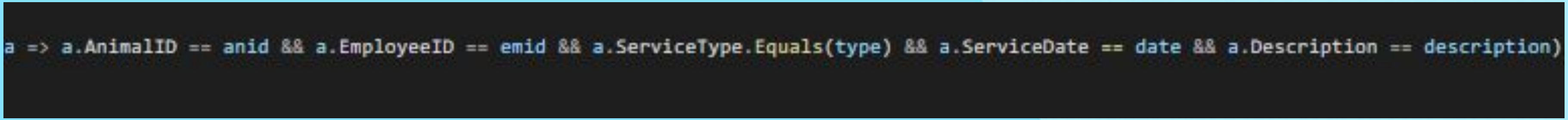


```
int id = (int)gridAll.SelectedItem.GetType().GetProperty("ID").GetValue(gridAll.SelectedItem, null);

var toDelete = Globals.ctx.Animals.Where(a => a.AnimalID == id).SingleOrDefault();
```

Clustered key



```
int anid = (int)gridAnService.SelectedItem.GetType().GetProperty("AnimalID").GetValue(gridAnService.SelectedItem, null);
int emid = (int)gridAnService.SelectedItem.GetType().GetProperty("EmployeeID").GetValue(gridAnService.SelectedItem, null);
Enum type = (Enum)gridAnService.SelectedItem.GetType().GetProperty("ServiceType").GetValue(gridAnService.SelectedItem, null);
DateTime date = (DateTime)gridAnService.SelectedItem.GetType().GetProperty("ServiceDate").GetValue(gridAnService.SelectedItem, null);
string description = (string)gridAnService.SelectedItem.GetType().GetProperty("Description").GetValue(gridAnService.SelectedItem, null);
```



```
a => a.AnimalID == anid && a.EmployeeID == emid && a.ServiceType.Equals(type) && a.ServiceDate == date && a.Description == description)
```

Requires the combination of all the fields in the row!

If we had 1 more week… 💭

- LiveCharts for Diagrams in MainWindow
- Edit option for Services, Employees
- Search option
- Installer

Summary

What was done and with what result…

- Add Employee with Wizard
- View Animals
- Add Animals
- Update Animals
- Add Owners
- Update Owner
- Delete Owner
- View Available and Occupied Crates
- Add Breeds
- Update Breeds
- Print Form
- Login Page with Exit - Logout

https://bitbucket.org/emma96/petstore/src/master/

https://trello.com/b/wqi5e4m1/petstore-project