Pet Management System

Given Task

A simple web front end application (this can be MVC or blazor) for listing / editing / deleting a fictitious list of people, this should also list each of their pets. This web application should link to a back end api for the CRUD methods. A search facility would be an added bonus.

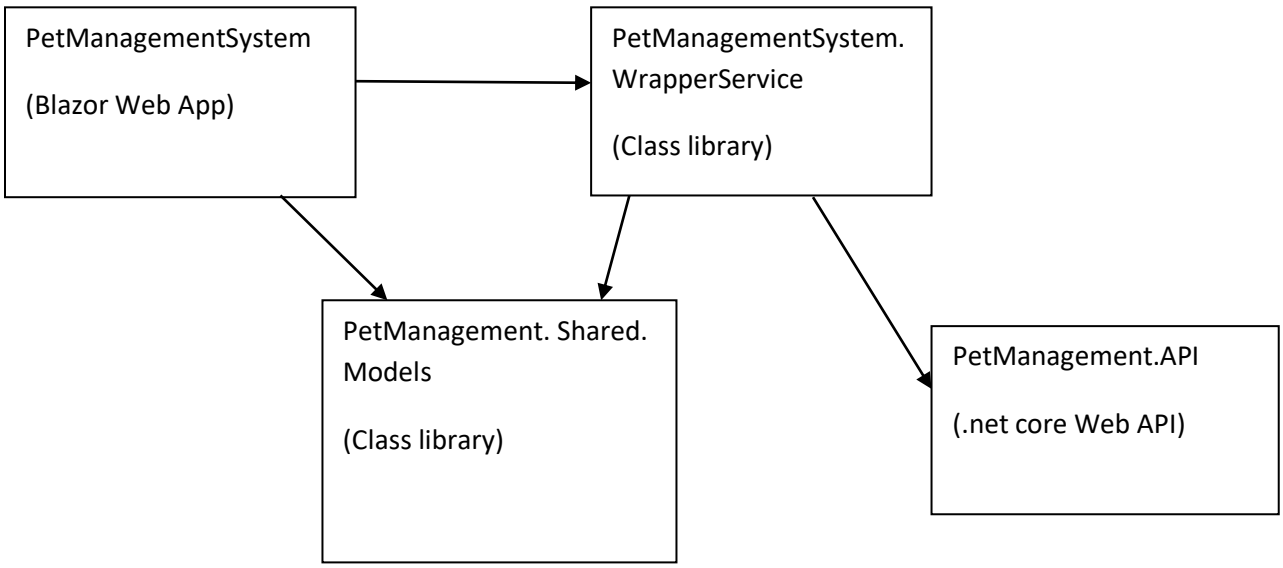
A back end api with the relevant controllers to Create, Update and Delete a persons details and their pets. The database can be localdb, mysql, Access or even an in memory structure of objects. Using Entity Framework would be a nice to have but not necessary.

Both projects should use c#, preferably .net core but other versions of .net can also be used.

Technologies Used

- Web Front End – Blazor Web App
- BackEnd – Asp .net core Web API
- Middleware services – Class library
- Database – localdb
- Entity Framework (Code First)
- Dependency Injection

Blazor Web App Architecture



PetManagementSystem (Frontend)

This project is client side front end application. I created **razor** pages to create interactive UI with the **Syncfusion.** I have added 2 web pages . 1. Save Pet and Owner details and 2.Display (It covers update, filtering and sorting) It connected with wrapperServices and shared.models.

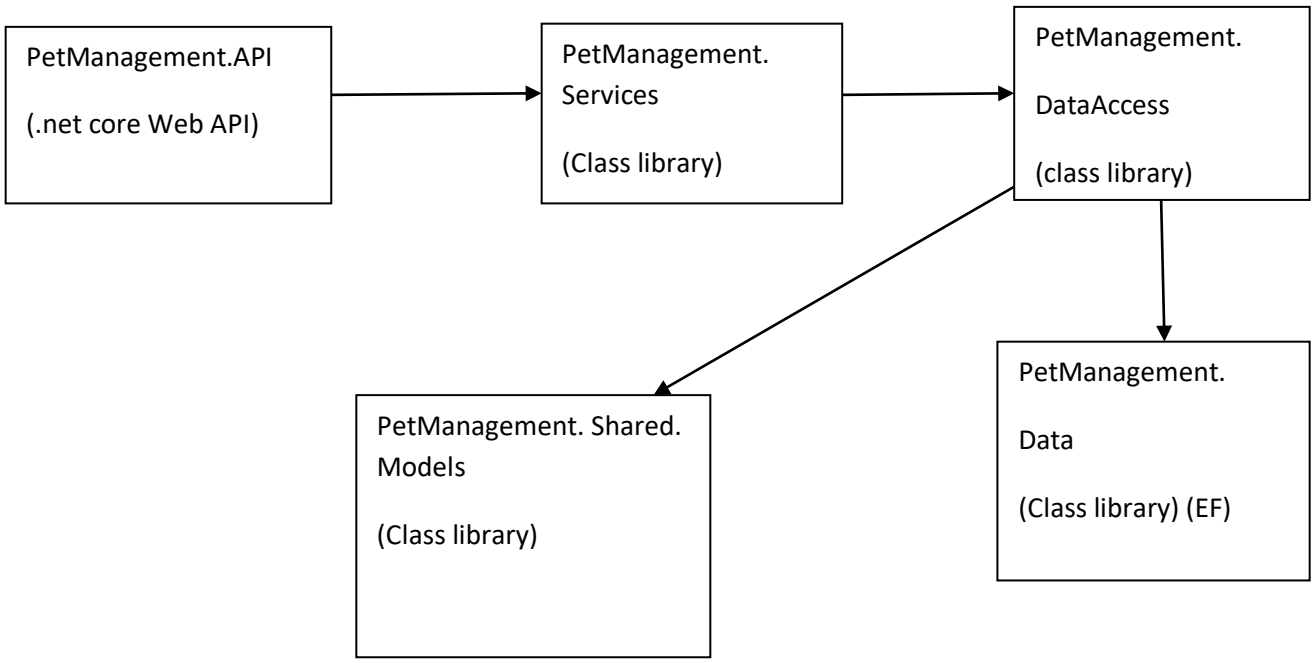
WrapperService

I created this project to separate from the blazor to handle validation, fetching data from the API, any data massaging and security. I wrote the Http Request calls in this layer

PetManagement.API

This is the API application I am connecting from the Blazor app. This will get the Url request from the wrapperService and perfom the API operation and return the result to the wrapper service. Will explain more below.

API Architecture



PetManagement.Shared.Models

This project is shared between API and Blazor. I created this to avoid redundancy in the code. All the models which are common to both the projects will be place here. I stored view models which will be

sent by API and received by Blazor application. In the future enum and constant variable can be added to this project

PetManagement.API

This is the API application. I created PetManagement controller with 5 action methods which will be discussed below. I mentioned the Route with controller and action which is called by Blazor App

PetManagement.Service

This is the service layer in the API. All the application logic is placed in here. It is the middleware between API layer and Data Access layer.

DataAccess layer

All the database related CRUD operations are placed here to segregate the database activities. This layer protects the database layer by mirroring the models. This is the only layer which connects directly to the data layer

Data layer

All the entity models and the DBcontext are placed in this layer. I performed code first approach to create the tables. After creating all the models I migrated to the Database to create tables or modify any.

Table Design

I created 3 simple table to perform the basic operations

- Species (Type of the Pet Animal)
- Person (Owner of the Pet)
- PetOwner (It has the owner and Pet combined information)

Functionalities

Save Pet and Owner details

I saved some Species(Pet type) to the database table and showing it in the drop down box and same for the persons. There is another text box to type the pet name vale and click the save button to save the data.

Display Pet and Owner details

I developed **Grid** to display all the records. It shows Id, Pet type, Owner name, pet name.

Sorting – I enabled sorting operation for Owner name and pet name column

Grouping – Enabled grouping functionality for pet type so that all the similar species sit together on the grid.

Filtering(Searching) – I enabled filtering for Owner name and Pet name column to search the particular record.

Update pet name - I developed the Pet name column with the edit option so that any new name can be added to the grid and updated in the database.

Future Enhancement

Technical enhancement

- Add TDD to the project which ensures the code quality
- Validation on the UI page
- Delete functionality
- Adding few more UI pages to get more input details about the pet and owner

Functionality Enhancement

- Adding more columns to the pet and owner table to store more realistic data
- Work down the validation properties to all the fields
- Adding feature to enter person and pet type information in the UI

Thank you for this opportunity

Terry & Mike