Pet Clinic

Your task will be to continue developing a B.E. system for a Veterinary Clinic that tracks information on pets, owners, appointments, veterinarians, and clinics. The B.E. should be a .NET API project with a SQL database controller by Entity Framework code first.:

- Please find the instruction and backbone code here
- Use Visual Studio as you IDE. Download free version here https://visualstudio.microsoft.com/downloads/
- Any additional technical help can be found online
- You are free to refactor any code
- You can use sql as docker image as local database https://learn.microsoft.com/en-us/sql/linux/quickstart-install-connect-docker?view=sql-server-ver16&tabs=cli&pivots=cs1-bash
- Do not hesitate to reach out for help to Christian.gjedrem@cegl.com

Background

Our company has been tasked by a leading pet clinic chain to develop a comprehensive booking system that will streamline their appointment scheduling process and enhance the overall experience for pet owners and clinic staff. As pet ownership continues to rise and veterinary care becomes increasingly specialized, the need for a user-friendly and efficient booking system has become paramount.

Customer Profile:

Our client is a reputable chain of pet clinics with multiple locations across the country. With a dedicated team of veterinarians and support staff, they provide a wide range of veterinary services, including wellness exams, vaccinations, surgical procedures, and specialized treatments.

Project Objectives:

The primary objective of this project is to develop a robust booking system that enables pet owners to easily schedule appointments for their pets while providing clinic staff with the necessary tools to manage appointments effectively. The system should be intuitive, reliable, and capable of handling a high volume of appointments across multiple clinic locations.

Key Features:

Pet Management: Allow pet owners to register their pets in the system, providing essential details such as pet name, breed, age, and medical history.

Appointment Scheduling: Enable pet owners to schedule appointments for various services, including routine check-ups, vaccinations, grooming, and medical consultations.

Clinic Location Management: Provide clinic staff with the ability to manage clinic locations, including address details, operating hours, and available services.

Veterinarian Management: Allow clinic administrators to manage veterinarian profiles, including their specialties, availability, and clinic schedules.

Appointment Tracking: Enable clinic staff to track and manage appointments, including scheduling, rescheduling, and cancellations, while maintaining accurate records of past and upcoming appointments.

Technical Requirements

The system should be developed using the .NET Entity framework, ensuring compatibility with the client's existing infrastructure and technologies.

Database management should be handled using a reliable and scalable database system, such as SQL Server to ensure data integrity and performance. For local development use docker desktop with sql server image or an in-memory database.

Entity Relationship

Our solution architect together with the customer has come up with a database entity relationship diagram provided in the attached image.

Sprints

Each sprint will involve backend development tasks to implement the features.

Testing and quality assurance activities will be conducted throughout each sprint to ensure the reliability and functionality of the system. Use the testing project to write unit tests and e2e tests to ensure overall functionality.

Sprint 0: Deployment to Cloud

Draw a diagram and explain what Azure services are best to use in serving the APIs to a F.E.

What services should you use for handling access management and log-in?

What technologies can you use to deploy this service to cloud?

Sprint 1: Pet Management and Clinic Location Management

Owner registration

As a person who owns pets and want to take care of them, I need to be able to register, edit and delete my information for the pet clinic.

When registering I should not be able to enter incorrect information or provide illegal information.

Endpoints

- Registration
  - Should be a POST request in ClientController.RegisterClient
  - Should not be able to register twice in the system
  - Should Pass validation before registering to the database
  - Information to register
    - Name
      - Not empty
      - Should be First name and Last name
      - Not more than 100 characters
    - Birthdate
      - Legal date
      - Not empty
      - Not more than 100 years old
    - Address Information in separate field as specified in ERD
      - Not empty
      - No more than 200 characters
      - Valid ZIP (4 Characters)
    - Sex
      - Not empty
      - Male or Female
    - Phone number
      - Not empty
      - Legal Norwegian number
    - Email
      - Not empty
      - Legal email
- EditDetails
  - Should be a PUT request in ClientController.EditClientDetails
  - Should pass validation before editing details in the database
  - Information to edit:
    - Everything as in ClientController.RegisterClient but the sex and birthdate
- Delete client
  - Delete client by id with owned pets(Sprint 1)
  - Pets should also be deleted if a client is to be deleted
  - Client should not have any active appointments(Sprint 3)
    - If there are return bad request with information of appointments:
    - “Cannot delete as your pet {PetName} have active appointments - [[{StartTime}, {EndTime} at {ClinicName} with {Vet Name}]],[[{StartTime}, {EndTime} at {ClinicName} with {Vet Name}]]….
- Get Client by Id
  - Return information on client by Id:
    - Name
    - Birthdate

        - Address Information in separate field as specified in ERD
        - Sex
        - Phone number
        - Email
        - Pets
        - PetName
        - PetType
        - PetBreed
        - MedicalHistory
        - Appointments (Sprint 3)
            - AppointmentStartTime
            - AppointmentEndTime
            - AppointmentClinicName
            - AppointmentVetName
- Get Client by Clinic Id
    - Return clients that have or had appointments in a clinic id (Sprint3). Should have the following information:
        - ClientId
        - ClientName
        - ClientEmail
        - ClientPhoneNumber
        - Pets
            - PetName
            - PetType
            - Appointments (Sprint 3)
        - ClinicName
        - NumberOfAppointments

Pet Registration

As a pet owner, I want to register my pet in the system by providing essential details for my pet: name, breed, age, and medical history. When registering I should not provide incorrect information as specified.

- Registration
    - Should be a POST request in PetController.RegisterPet
    - Should not be able to register twice in the system for the same pet
    - Should Pass validation before registering to the database
    - Information
        - Name
        - Not empty
        - Maximum of 100 characters
        - Type
        - Only one of the following:
            - Cat
            - Dog

        - Fish/Aquatic
        - Bird
        - Reptile
        - Insect
    - Birthyear
        - Not empty
        - Not older than 200 years ago
    - Breed
        - Not empty
    - Medical history
        - Can be empty
        - Not more than 5000 characters long
- EditDetails
    - Should be a PUT request in PetController.EditPetDetails
    - Should pass validation before editing details in the database.
    - Information to edit should be everything but Type and Birthyear
- Delete Pet
    - Delete pet by id without deleting appointments (Sprint 2)
    - Pet should not have any active appointments(Sprint 2)
        - If a pet has active appoint return the following error message “Cannot delete as your pet {PetName} have active appointments - [[{StartTime}, {EndTime} at {ClinicName} with {Vet Name}]], [[{StartTime}, {EndTime} at {ClinicName} with {Vet Name}]]….
- Get Pet by Id
    - Return information on pet by Id:
        - PetName
        - PetType
        - PetBreed
        - MedicalHistory
        - Appointments (Sprint 3)
            - AppointmentStartTime
            - AppointmentEndTime
            - AppointmentClinicName
            - AppointmentVetName
- Get Pets by Client Id
    - Return pets for an owner id with the following information
        - PetId
        - PetName
        - PetType
        - Appointments
            - ClinicName
            - NumberOfAppointments
- Get Pets by Clinic Id
    - Return pets for a clinic there they had or have an appointment (Sprint 3)

        - PetId
        - PetName
        - PetType
        - NumberOfAppointments

Clinics

As a clinic staff member, I want to be able to add and manage clinic including location and address details, operating hours, and available services. As a Client I need to be able to filter clinics based on their specialty. When registering or editing I should not provide incorrect information as specified below.

Endpoints

- ClinicController.UpsertClinic
    - Registering and Editing Clinic with valid information
    - Should not be able to register the same clinic twice
    - Information to add
        - Name
            - Not empty
            - Maximum 200 characters
        - Location, separate fields with validation as indicated for client
        - Operating Hours
            - Opening Hours with weekday specified
            - Closing Hours with weekday specified
            - Both should be a valid opening hours
            - Opening hour should be before closing hour for the same day
            - Closing hour should be after opening hour for the same day,
        - Phone number with validation as before
        - Email with validation as before
        - Maximum number of vets
        - Number of active vets
            - Should specify how many vets can be on duty at the same time
            - Minimum 1

- ClinicController.GetClinicsBytIds
    - Get Clincs information by ids
    - For clinics stored should return the following information
        - ClinicId
        - Name
        - Location as one field
        - Operating Hours, array of opening hours and closing hours in the format of "dddd HH:mm"

        - Phone number
        - Email
    - Vets (Sprint 2)
        - VetName
        - Position
        - Email Phone
- Number of Appointments
    - Number of Clients
    - Number of Pets
    - Available services
        - Available services as per Veterinarian specialties
    - Maximum number of active vets

- ClinicController.GetClinicBySpeciality
    - Get Clincs information by speciality (Sprint 2)
    - Following information should be returned:
        - ClinicId
        - Name
        - Location as one field
        - Phone
        - Email
        - Specialties (Sprint 2)
- ClinicController.GetSpecialities
    - Get all available specialties for all clinics. (Sprint 2)
    - Should return:
        - Specialty Name
        - Number of Vets with this specialties across the clinics

Sprint 2: Veterinarian Management

User Stories:

As a clinic administrator, I want to manage veterinarian profiles, including their specialties, availability, and clinic schedules.

Veterinarians

As a clinic administrator I want to manage veterinarian profiles. I want to register, edit and delete them as well as add specialties to them.

Endpoints

- VetController.UpsertVet
    - Insert or Edit vet with valid data
    - Should not be able to add specialty her
    - Only one vet per clinic for the same name
    - Information to add/edit:
        - Name
            - Validation as with client name
            - Only on add
        - Birthdate
            - Validation as with client birthdate
            - Only on add
        - Address
            - Validation as with client address
        - Position
            - One of the following
                - Technician
                - Groomer
                - Doctor
                - Head Doctor
        - Phone
            - Validation as with client phone number
        - Email
            - Validation as with client email
- VetController.DeleteVet
    - Delete vet
    - Should not be able to delete vet with active schedules(Sprint 2)
    - Return ErrorMessage “Cannot delete as vet {VetName} have active schedules - [[{StartTime}, {EndTime} at {ClinicName} with {Vet Name}]], [[{StartTime}, {EndTime} at {ClinicName} with {Vet Name}]]….

- VetController.GetVetsByIdsAndMonth
    - Return Vet information from specified Ids:
        - VetId
        - VetName
        - VetPhoneNumber
        - VetEmail
        - VetSpecialites
- SpecialtyName
    - Schedules(vet schedules for specified month, sprint 2)
        - ScheduleStartDate
        - ScheduleEndDate
        - ScheduleType

- VetController.GetVetsByClinicId
    - Return Vet information for vets that have schedule for the clinic specified
    - Information to return:
        - VetId
        - VetName
        - VetPhoneNumber
        - VetEmail
- VetController.GetVetsBySpecialties
    - Return vets across the clinics that have the specified specialty
    - Information to return:
        - VetName
        - VetPhoneNumber
        - VetEmail
- VetController.AddSpecialityToVet
    - Should not be able to add if it is already with the vet
    - Ensure that no duplicates are added to the specialty Table

Clinic Schedules

As a clinic administrator I should be able to see the overview of a clinic's schedule in their opening hours. As a clinic administrator I should be able to set Vets on duty during clinic's opening hours. AS a clinic administrator I should not be able to set more Vets on the same time than specified in the clinic's number of active vets.

Endpoint

- ScheduleController.RegisterSchedule
    - Inserting information on a schedule
        - ClinicId
            - if a schedule already exists within the times specified, ensure that the number of different vets for the same time are not larger than the maximum number of vets for a clinic
        - Start time
            - Should be after or as the clinic opens
            - Should not be after End time
        - End Time
            - Should be before or as the clinic closes
            - Should not be before Start time
            - minimum of 15 minutes after Start time
        - VetId
            - The vet that is on active duty

                - Should exist within the clinic
                - Should not already be scheduled for that time
            - Type
                - Should specify if it is
                - Appointment(Sprint 3)
                - Scheduled Operation
                - Active duty
- ScheduleController.EditSchedule
    - Editing Shedule, the following information can be edited
    - If the schedule is an appointment:
    - cannot be changed:
        - This should happen in the appointment controller
    - If it is an active duty schedule
        - EndTime and StartTime can be changed
            - Validation as in Insert
        - VetId can be changed
            - if vet is in the clinic
            - If vet does not have an active schedule in that time already
            - Error message “The vet {vetId} is already scheduled {StartTime} – {EndTime}”
    - If it is a Scheduled Operation
        - EndTime and StartTime can be changed
        - Validation as in Insert
        - VetId can be changed
            - If vet is in the clinic
            - If vet does not have an active schedule in that time already
            - Error message “The vet {vetId} is already scheduled {StartTime} – {EndTime} at {ClinicName}”
            - Vet can only be doctor or head doctor
- ScheduleController.DeleteSchedule
    - Can only be deleted if a schedule is not an appointment
    - If the deleted schedule is an active duty, delete and return message:
    - “Vets missing on active duty {StartTime} – {EndTime} at {ClinicName}”
- ScheduleController.GetSchedulesByClinicIdandMonth
    - Return the following information for a clinic for the specified month:
        - ScheduleStart
        - ScheduleEnd
        - VetName
        - ScheduleType
        - PetName if it is an appointment
        - AppointmentComment if it is an appointment

- ScheduleController.GetScheduleById
    - Return the following information for a scheduled for the specified id:
        - ScheduleStart
        - ScheduleEnd
        - VetId
        - VetName
        - VetPhoneNumber
        - VetEmail
        - VetSpecialties
        - ScheduleType
        - PetId
        - PetName if it is an appointment
        - PetMedicalHistory if it is an appointment
        - ClientId
        - ClientName if it is an appointment
        - ClientEmail if it is an appointment
        - ClientPhoneNumber if it is an appointment
        - AppointmentComment if it is an appointment

Sprint 3: Appointment Scheduling

User Stories:

As a pet owner, I want to schedule appointments for various services, including routine check-ups, vaccinations, grooming, and medical consultations. An Appointment should have the following Endpoints

- AppointmentController.RegisterAppointment
    - I should be able to register an appointment for my pet
    - Information to add:
        - PetId
            - Id of the pet
            - Should exist
        - StartDate
            - Should be after opening hours for the clinic
        - EndDate
            - Should be before closing hours
            - The interval for the appointment should be inside a schedule with an active vet on duty
        - Type, one of the following
            - Grooming
                - A groomer should be on duty
            - Routine check-up

            - Doctor, Head doctor or technician should be on duty
        - Vaccination
            - Doctor, Head, technician or a groomer should be on duty
        - Medical Consultation
        - Doctor or Head doctor should be on duty
    - Status, one of the following
        - Booked
        - Completed
        - Billed
        - Cancelled
- AppointmentController.EditAppointmentDetails
    - Change the Start and EndDate for the appointment
    - Should be within opening hours and have a fitting vet on duty
- AppointmentController.DeleteAppointment
    - Delete appointment
    - Cannot delete if an appointment is completed or billed
- AppointmentController.ChangeAppointmentVet
    - Change the vet for the appointment
    - Can only be vet from this clinic that is on active duty and in a fitting role
    - Previous vet should have this schedule changed to active duty
    - Wanted Vet should have a new schedule with the appointment details
    - The active duty schedule need to be replaced with appointment details as well as the consistency of active duty for that day should be kept intact
- AppointmentController.CancelAppointment
    - Set status to cancelled
    - Only for booked appointments
- AppointmentController.ChangeAppointmentStatus
    - Change to Completed or Booked

End to end testing

Use this documentation to set up integration tests for api endpoints https://learn.microsoft.com/en-us/aspnet/core/test/integration-tests?view=aspnetcore-8.0

Bonus task

How would you set up an email notification system for completed registration, completed appointment scheduling and billing?

How would you approach the problem of changing number of vets for a clinic? What would happen if you want to reduce number of vets if there are active schedules? What are the possible scenarios and how would you handle them?