Object Oriented Software Engineering

Terminal Lab Exam

Name: Aoun-Haider

ID: FA21-BSE-133

Submitted to: Mam Sana-Maqbool

Question: 01

Online Pet House is a web and android application where the purchase and sale of pets took place and vets are available for pet treatments. Today we came to know about the circumstances that most of the people depend on Local markets and nearby houses to buy a pet for their house.

Through this project, we are planning to digitalize this process through online marketing. Thus, individuals can also rely on other people across the district to buy a variety of breeds of pets. People could also be able to come to know about the different price ranges for the same breed offered by different people. Thus, our app provides a platform where the users could buy or sell pets by providing the contact details of the buyers/sellers. We are building the android application in a way that we create a login for all users. After logging in, the user is directed towards the homepage where he/she could see the pets available for buying. Besides these, one could see options for selling pets and managing his/her account. The details of the pet, some of its pictures, its age (rough one), etc. are all to be provided if one could sell a pet whereas the buyer can view pets classified into categories. Moreover, we are also providing a feature that allows the users to know about all the pet shops in the city so that they can buy pets or their food from there also. Thus, our app makes a perfect choice for the ones who love pets and seek variant breeds of them.

a. **Identify the attributes.**

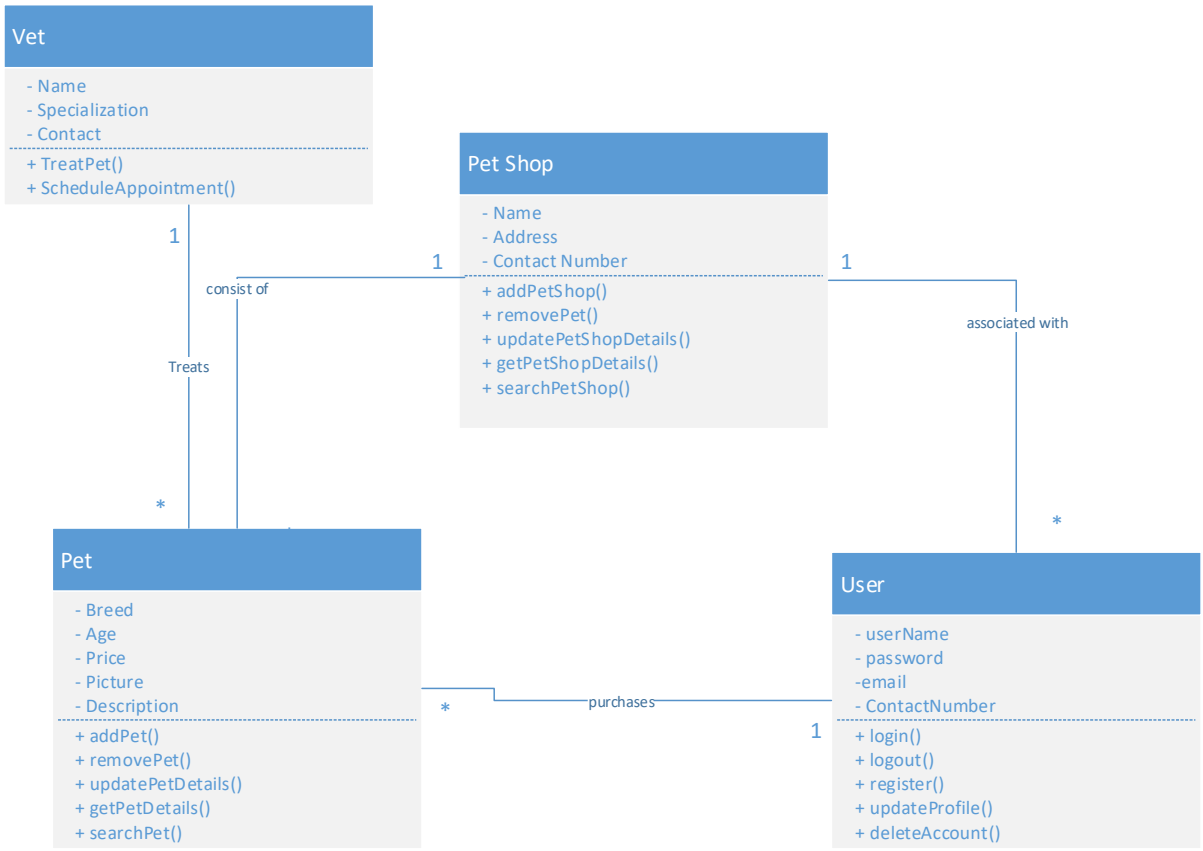
User: Username, Password, Email, Contact number

Pet: Breed, Age, Price, Pictures, Description

Pet Shop: Name, Address, Contact number

Vet: Name, Specialization, Contact

b. Draw a class diagram of the given case study.



c. Identify the operations.

User: login(), logout(), register(), updateProfile(), deleteAccount()

Pet: addPet(), removePet(), updatePetDetails(), getPetDetails(), searchPet()

Pet Shop: addPetShop(), removePetShop(), updatePetShopDetails(), getPetShopDetails(), searchPetShop()

Vet: TreatPet(), ScheduleAppointment()

d. Identify the relationships i-e Generalizations, aggregation, dependency and associations.

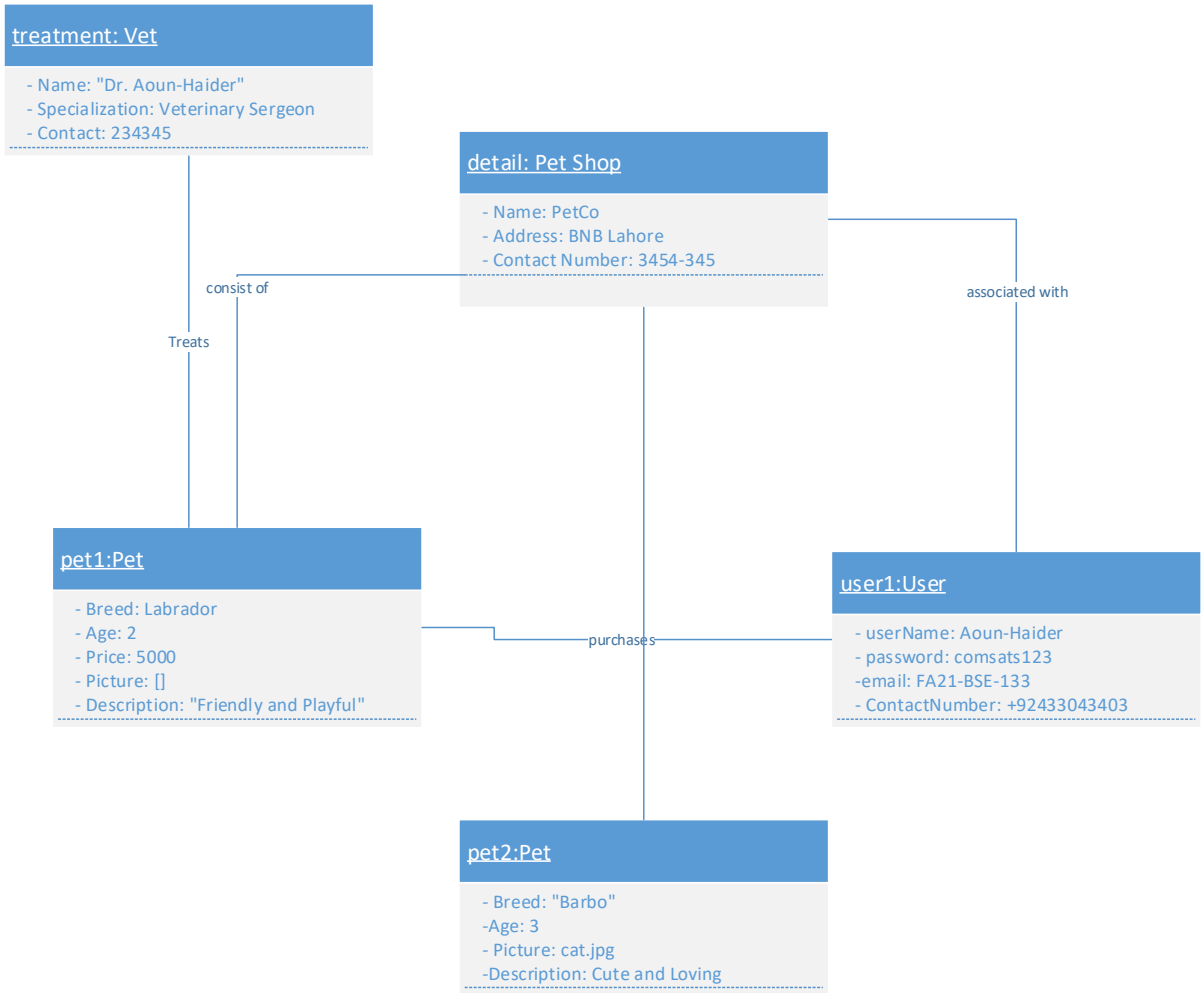
Generalization: No generalization relationships are evident in the given scenario.

Aggregation: There is a possible aggregation relationship between User and Pet, where a User can have multiple Pets.

Dependency: There is a dependency relationship between User and Pet Shop, as the User needs information about Pet Shops.

Association: There is an association relationship between User and Pet, as a User can buy or sell Pets. One vet can have many pets under treatment. Association exist b/w Pet and Vet class.

e. Identify objects and draw an object diagram based on class diagram drawn on scenario.



f. Identify the design pattern exist in drawn class diagram

Abstraction occurrence pattern exist in class diagram because user can purchase multiple pets and more instances are required to process.

g. Also implement it.

```
interface PetOwner {

   void addPet(Pet pet);
```

```
    void removePet(Pet pet);
}

interface MedicalProfessional {
    void treatPet(Pet pet);
    void scheduleAppointment();
}

class User implements PetOwner {
    private String username;
    private String password;
    private String email;
    private String contact;
    private List<Pet> pets;

    public User(String username, String password, String email, String
contact) {
        this.username = username;
        this.password = password;
        this.email = email;
        this.contact = contact;
        this.pets = new ArrayList<>();
    }

    @Override
    public void addPet(Pet pet) {
        pets.add(pet);
```

```
    }

    @Override
    public void removePet(Pet pet) {
        pets.remove(pet);
    }

    // Getters and setters for the attributes

    // Other methods and functionalities
}

class Pet {
    private String breed;
    private int age;
    private double price;
    private List<String> pictures;
    private String description;

    public Pet(String breed, int age, double price, List<String> pictures, String
description) {
        this.breed = breed;
        this.age = age;
        this.price = price;
        this.pictures = pictures;
        this.description = description;
    }
```

```
    // Getters and setters for the attributes

    // Other methods and functionalities
}

class PetShop {
    private String name;
    private String address;
    private String contact;

    public PetShop(String name, String address, String contact) {
        this.name = name;
        this.address = address;
        this.contact = contact;
    }

    // Getters and setters for the attributes

    // Other methods and functionalities
}

class Vet implements MedicalProfessional {
    private String name;
    private String specialization;
    private String contact;
```

```
    public Vet(String name, String specialization, String contact) {
        this.name = name;
        this.specialization = specialization;
        this.contact = contact;
    }

    @Override
    public void treatPet(Pet pet) {
        // Code to treat the pet
    }

    @Override
    public void scheduleAppointment() {
        // Code to schedule an appointment
    }

    // Getters and setters for the attributes

    // Other methods and functionalities
}

public class Main {
    public static void main(String[] args) {
        // Create instances of User, Pet, PetShop, and Vet
        PetOwner user = new User("JohnDoe", "password",
"johndoe@example.com", "555-1234");
```

```
        Pet pet = new Pet("Labrador", 2, 500.0, new ArrayList<>(), "Friendly and
playful");

        PetShop petShop = new PetShop("PetCo", "123 Main St", "555-5678");

        MedicalProfessional vet = new Vet("Dr. Smith", "Veterinary Surgeon",
"555-9876");


        // Add the pet to the user's list of pets

        user.addPet(pet);


        // Perform other operations and interactions between objects

    }

}
```

Question: 02