**A cat and dog looking at icons

AI-generated content may be incorrect.**

**Group G9**

**Social Network for Pets**

Ahmet Enes Şahan ID:26779932038 Requirement Analysis - E/R Diagram

Şevval Gürler ID:21917417002 Relational Database Schema - SQL Tables

**STEP 1**

**Objectives**

* Enable pet owners to create accounts
* Allow users to create profiles for their pets
* Facilitate interactions between pets (friendships, groups, etc.)
* Enable users to share posts and interact with other users
* Support social features such as messaging and commenting

**User Types**

* **Pet Owners**: Can add their pets, share posts, and message other users
* **Veterinarians**: Can share pet health-related information and interact with users
* **Admins**: Can manage the database and remove fake accounts

**Problems Solved**

* Offering a communication hub for pet owners to share experiences and support each other.
* Fostering pet communities through social interaction and networking
* Enabling veterinarians to engage with users

**Key Data and Relationships**

* Ownership relationship between **Users** and **Pets**
* Friendship relationships between **Pets**
* Messaging and following relationships between **Users**
* **Posts** and **Comments**

**Advanced SQL Queries**

* How many total posts does a specific user own?
* Which posts from a veterinarian have received the most engagement (likes, comments)?
* What is the average age of pets belonging to a specific breed (e.g., Labrador Retrievers)?
* Who are the pets with the most friends among pets?
* What are the most popular pets in a specific city?

**STEP 2**

**Entities and Attributes**

**1. Users**

This entity represents users who interact with the system. Users can be pet owners, veterinarians, or general users.

* **UsersID** (Primary Key) → A unique identifier for each user.
* **Name** → The full name of the user.
* **Email** (Unique) → A unique email address used for user authentication.
* **Password** → A hashed password for login security.
* **City** → The city where the user lives.
* **UsersType** → Defines the type of user (e.g., Pet Owner, Veterinarian).
* **RegistrationDate** → The date when the user registered in the system.

**2. AdminAction**

**This entity represents administrative actions performed by users with the "Admin" role to manage and moderate the system.**

* **ActionID (Primary Key) → A unique identifier for each administrative action.**
* **AdminID (Foreign Key → Users.UsersID) → References the admin user who performed the action.**
* **ActionType → Describes the type of action taken (e.g., "Delete Post", "Ban User", "Remove Comment").**
* **TargetUserID (Foreign Key → Users.UsersID) → Identifies the user who is the subject of the action (e.g., the banned user). *(nullable depending on action)***
* **TargetPostID (Foreign Key → Post.PostID) → Identifies the post affected by the action. *(nullable depending on action)***
* **ActionDate → The timestamp of when the action was performed.**
* **Description → Additional notes or context related to the action taken by the admin.**

**3. Pet**

This entity represents pets that belong to users.

* **PetID** (Primary Key) → A unique identifier for each pet.
* **Name** → The pet’s name.
* **Species** → The type of animal (e.g., Dog, Cat, Rabbit).
* **Breed** → The breed of the pet (e.g., Labrador Retriever, Persian Cat).
* **Age** → The age of the pet in years.
* **Gender** → The gender of the pet (Male/Female).
* **OwnerID** (Foreign Key → Users.UsersID) → References the user who owns the pet.
* **RegistrationDate** → The date when the pet was added to the system.

**4. Post**

This entity represents posts shared by users.

* **PostID** (Primary Key) → A unique identifier for each post.
* **Content** → The text content of the post.
* **ImageURL** → A link to an image attached to the post.
* **CreatedAt** → The timestamp when the post was created.
* **UsersID** (Foreign Key → Users.UsersID) → References the user who created the post.

**5. Comment**

This entity represents comments made on posts.

* **CommentID** (Primary Key) → A unique identifier for each comment.
* **Content** → The text content of the comment.
* **CreatedAt** → The timestamp when the comment was created.
* **UsersID** (Foreign Key → Users.UsersID) → References the user who wrote the comment.
* **PostID** (Foreign Key → Post.PostID) → References the post on which the comment was made.

**6. Likes**

This entity represents users liking posts.

* **LikeID** (Primary Key) → A unique identifier for each like.
* **UserID** (Foreign Key → Users.UsersID) → References the user who liked the post.
* **PostID** (Foreign Key → Post.PostID) → References the post that was liked.
* **LikedAt**→ The timestamp when the like was created.

**7. Friendship**

This entity represents friendships between pets.

* **FriendshipID** (Primary Key) → A unique identifier for each friendship connection.
* **Pet1ID** (Foreign Key → Pet.PetID) → References one pet in the friendship.
* **Pet2ID** (Foreign Key → Pet.PetID) → References the other pet in the friendship.
* **CreatedAt** → The timestamp when the friendship was formed.

**8. Message**

This entity represents private messages sent between users.

* **MessageID** (Primary Key) → A unique identifier for each message.
* **SenderID** (Foreign Key → Users.UsersID) → References the user who sent the message.
* **ReceiverID** (Foreign Key → Users.UsersID) → References the user who received the message.
* **Content** → The message content.
* **SentAt** → The timestamp when the message was sent.

**9. Veterinarian**

This entity represents veterinarians who provide medical support and advice for pets.

* **VeterinarianID** (Primary Key) → A unique identifier for each veterinarian.
* **Name** → The full name of the veterinarian.
* **ClinicName** → The name of the veterinarian’s clinic.
* **City** → The city where the veterinarian is located.
* **Email** (Unique) → A unique email address for contact purposes.

**Relationships and Their Cardinalities**

* **Users - Pet (1:M):**
  + A **user** can own **multiple pets**, but each **pet** belongs to **only one user**.
* **Users - Post (1:M):**
  + A **user** can create **multiple posts**, but each **post** belongs to **only one user**.
* **Users - Comment (1:M):**
  + A **user** can write **multiple comments**, but each **comment** belongs to **only one user**.
* **Users - Likes (M:M):**
  + A **user** can like **multiple posts**, and each **post** can have **multiple likes**.
* **Pet - Friendship (M:M):**
  + A **pet** can have **multiple friendships**, and a **friendship** is always between **two pets**.
* **Users - Messages (M:M):**
  + A **user** can send messages to **multiple users**, and each **message** is exchanged between **two users**.
* **Veterinarian - Post (1:M):**
  + A **veterinarian** can create **multiple posts**, but each **post** belongs to **only one veterinarian**.
* **Veterinarian – Users (M:M):**
  + One user can follow or communicate with more than one veterinarian. One veterinarian can provide services or counseling to multiple users.
* **Admin - Users (1:M):**
  + An admin can take actions on multiple users, but each action targets only one user.
* **Admin - Post (1:M):**
  + An admin can moderate multiple posts, but each moderation action affects only one post.

**E/R Diagram**

**A diagram of a network

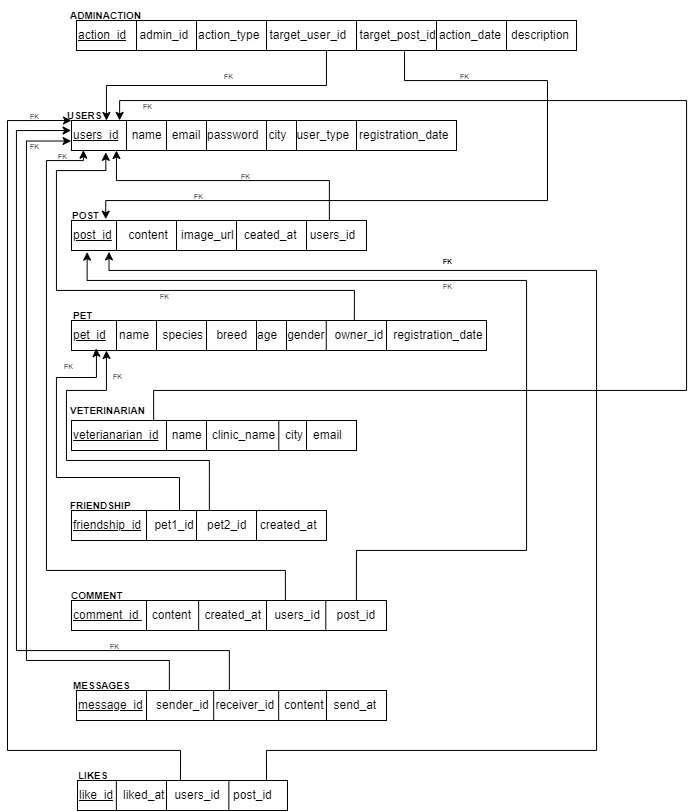
AI-generated content may be incorrect.**

**STEP 3**

**Social Network for Pets - Relational Schema**

**Tables and Attributes**

1. **Users** (UsersID, Name, Email, Password, City, UsersType, RegistrationDate)
2. AdminAction (ActionID, AdminID, ActionType, TargetUsersID → UsersID, TargetPostID → PostID, ActionDate, Description)
3. **Pet** (PetID, Name, Species, Breed, Age, Gender, OwnerID → UsersID, RegistrationDate)
4. **Post** (PostID, Content, ImageURL, CreatedAt, UsersID → UsersID)
5. **Comment** (CommentID, Content, CreatedAt, UserID → UsersID, PostID → PostID)
6. **Likes** (LikeID, LikedAt, UsersID → UsersID, PostID → PostID)
7. **Messages** (MessageID, Content, SentAt, SenderID → UsersID, ReceiverID → UsersID)
8. **Friendship** (FriendshipID, Pet1ID → PetID, Pet2ID → PetID, CreatedAt)
9. **Veterinarian** (VeterinarianID → UsersID, Name, ClinicName, City, Email)



**SQL CREATE TABLE StatementA screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**System Functionalities**

**1.Registration**

A screenshot of a computer

AI-generated content may be incorrect.

To become a member of the PetSocial platform, it is necessary to complete the registration form on this page accurately and thoroughly. A user account can be created by specifying a username, a valid email address, and a secure password. When choosing a password, it is important to meet the security requirements (minimum 8 characters, uniqueness, inclusion of numeric and special characters). Upon completion of the registration process, users will be able to take advantage of all the features the platform offers**.**

**2.Pet Owner Login**

**A screen shot of a login screen

AI-generated content may be incorrect.**

Access to the PetSocial platform requires the entry of user credentials (username and password). The secure login system ensures the protection of user accounts.

**3.Home Page**

**A screenshot of a computer

AI-generated content may be incorrect.**

This page allows registered users to manage their personal profile and pet relationships on the platform. In the 'Your Pets' section, users can view their registered pets, add new pets, or update information about their existing pets. The 'Your Pets' Friendships' section allows users to manage their connections with other pet owners and initiate new friendships. Additionally, users can adjust their account settings and log out of the platform from this page.

**4.Adding Pets**

**A screenshot of a computer

AI-generated content may be incorrect.**

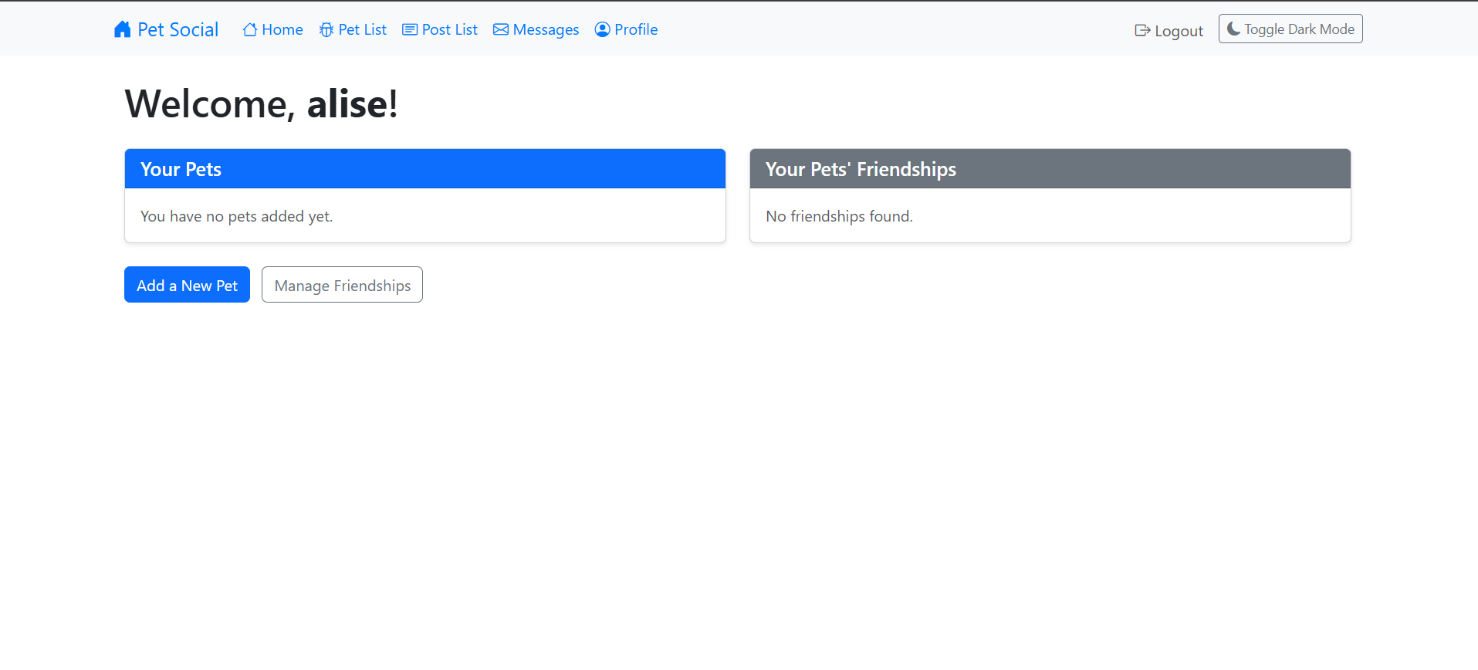
Users can add their pets to the platform and create a profile. This profile should include basic information such as the pet's name, pet's species, breed, age and gender.

**5.Pets ListA screenshot of a computer

AI-generated content may be incorrect.**

This section contains a list of the pets the user has registered on the platform and their detailed profiles. Users can update their pets' information from this section.

**6.Pets Friendship**



This feature enables users to connect with other pet owners who have similar pets by filtering based on criteria such as species, breed, age, and interests. This allows users to organize joint activities, share experiences, and establish new social circles. The platform has taken the necessary security measures to ensure that users can interact in a safe and controlled environment.

**7.Post**

**A screenshot of a computer

AI-generated content may be incorrect.**

The Pet Social Posts page serves as a central platform for pet owners to interact, share information, and exchange experiences. Users can view, comment on, and like posts from other members, thereby fostering community engagement. Content related to specific topics can be easily accessed via hashtags. The Pet Social Posts page aims to provide pet owners with a supportive environment where they can gain knowledge and assist one another by sharing their experiences. User privacy and respectful interactions are prioritized.

**8.Message**

**A white background with blue text

AI-generated content may be incorrect.**

This page is designed for registered users to send private messages to other users on the platform. The recipient of the message is selected from the 'Receiver' section, and the message text is entered in the 'Content' section. The message is sent to the recipient by pressing the 'Send' button.

**9.Message Notification**

**A screenshot of a computer

AI-generated content may be incorrect.**

**10.Profile**

**A blue and white line

AI-generated content may be incorrect.**

This page allows registered users to view and update their personal information (first name, last name, email address, city). The user type (e.g., Pet Owner) and profile picture can also be managed from this section. Users can access and view related data via the 'Pets', 'Posts', and 'Likes' tabs. Changes made can be saved to the system using the 'Update Profile' button.

**11.Vetenerian Login**

**A screenshot of a computer

AI-generated content may be incorrect.**

This platform is designed to help veterinarians easily manage pet information, appointments, and health notes. To add a new appointment, simply select your pet's name, enter the appointment date and reason, and click the "Add Appointment" button.

You can also add health notes about your pets. To do so, choose the relevant pet, write detailed notes, and click the "Add Note" button.

**12.Admin Login**

**A screenshot of a computer

AI-generated content may be incorrect.**

This admin panel allows the administrator to view and edit all users on the platform. The list shows usernames, email addresses, and user types. By using the "Edit" button next to each user, the admin can update their information. Additionally, the admin can manage posted content, moderate comments, access statistics, and view advanced reports. All these tasks are performed by the administrator to ensure the smooth operation of the platform**.**

**13.Advanced Report**

**A screenshot of a computer

AI-generated content may be incorrect.**

The sections include:

* Total Posts by User: Displays the total number of posts made by a specific user (ID: 1).
* Top 5 Veterinarian Posts by Engagement (Likes + Comments): Lists the most engaging veterinarian posts based on likes, comments, and overall engagement.
* Average Age of Pets by Breed: A dropdown menu to select a breed; currently, no data is available for the selected breed.
* Top 5 Pets with Most Friends: Shows pets with the highest number of friends; in this case, "Kızıl" has 1 friend.
* Top 5 Popular Pets by City: A dropdown menu to select a city; currently, no popular pets are found for "Lake Amy."

At the bottom, there is a button labeled "Back to Admin Dashboard" which navigates back to the main dashboard.

**14.Dark Mode**

**A screenshot of a computer

AI-generated content may be incorrect.**

Dark mode is designed to improve user experience in low-light conditions. This mode reduces eye strain by changing the color scheme of the interface.

**Challenges & Improvements:**

Customization of User Model: Extending Django's default user model presented initial complexities. To address this, we implemented Django's AbstractUser model. This customization allowed us to seamlessly integrate veterinarian-specific fields (e.g., clinic name, specialization) directly into the user profile, enhancing the application's ability to cater to different user roles.

Media Files: Configuring file paths for photo uploads and serving MEDIA\_URL required careful attention. The primary challenge was ensuring correct access permissions and storage management. By implementing a dedicated media storage strategy and carefully configuring Django's MEDIA\_ROOT and MEDIA\_URL settings, we successfully enabled users to upload and display images, enriching the user experience.

Data Relationships: Designing relationships between users, pets, and posts demanded careful consideration to ensure data integrity and efficient querying. To address this, we implemented appropriate foreign key constraints and indexes in the database. This improved query performance and ensured that data was consistently linked across different parts of the application.

**Interface Challenges:** Achieving full responsiveness across all pages was initially incomplete. After identifying key areas lacking mobile optimization, we implemented a mobile-first design approach, utilizing CSS frameworks like Bootstrap to ensure seamless adaptation to various screen sizes. This significantly improved the user experience on mobile devices, leading to increased engagement.

Advanced Reporting with SQL Queries: The admin panel needs to output advanced reports about system usage. Creating SQL queires for different metrics such as calculating engagement (likes and comments) on veterinarian posts required complex calculations. Creating custom SQL query methods that directly query to database models made reporting functionalities more efficient.

**Conclusion:**

The aim of this assignment was to evaluate the development process of the PetSocial web application. Throughout the project, a social platform was created using Django, enabling users to interact. The key finding is that Django allows for rapid prototyping and facilitates the development process. The most important result is the valuable experience gained in web application development through this project.