Information system for pet services

(Project for Systems III)

Dana Donevska

Definition of the problem

The majority of individuals who have pets develop strong emotional bonds with them. Caring for these animals is an integral part of their daily routine. However, there are situations when unavoidable commitments, such as important meetings or business trips, require them to be away from their pets. Finding a safe and comfortable environment for their pets, one that closely resembles their own home, can be a challenging task. Even when there's ample time for planning, the only available options often involve pet hotels or shelters, which may not provide the ideal experience for pets and owners alike. Alternatively, entrusting a relative or friend with pet care can be uncertain, as they might lack the necessary knowledge and skills.

This situation can lead to various issues, ranging from pets having a distressing time, owners worrying about their well-being, pets experiencing loneliness, a potential breakdown in trust between the pet and owner, and, in extreme cases, pets developing behavioral problems or falling ill. Furthermore, the lack of a reliable system for pet care and advice can leave pet owners feeling isolated and helpless when facing pet-related problems, whether it's illness, fur issues, or the need for general advice and guidance.

To address these challenges, as an exercise in the course Systems III, we propose the development of an integrated system, a web application that serves as an all-in-one solution for pet owners. This system will include:

1. An online vet forum where pet owners can seek advice, share experiences, and connect with veterinary professionals or other people in such situations to address their pet's health and behavior concerns.

2. Pet sitting services that offer a safe and home-like environment for pets when their owners must be away, ensuring their well-being and comfort.

I believe this information system will be of great value in Koper, as well as in every other part of Slovenia, and abroad. By creating this holistic system, I aim to provide a seamless and trustworthy solution for pet owners, enhancing the overall quality of life for both pets and their devoted human companions.

Functional and nonfunctional requirements

Functional Requirements

The system should enable the following functionalities:

1. The system should allow every user to make an account and insert all the necessary information (basic information: name, surname, age, home address, city, email, phone number and their role in the system), as well as their pets (pet information: name, type, breed, age, character description).

    1.1. The system should allow the user to modify their personal information and delete their account (only if the user hasn't made a deal valid at that time, or in the future).

    1.2. The system should allow the administrator to delete any profile that might be violating the system's purpose or its users.

2. The system should allow pet caretakers to make a pet-sitting listing for the period in which they can take care of a certain type of pet.

    2.1. Pet caretakers should be able to put a pricing for the service they provide.

3. The system should allow users to filter through the listings (categories: city, time period, ratings, price and type of pet), and contact (via call or email) many Pet caretakers in the process of finding their best match.

    3.1. Pet caretakers should be able to accept or decline service requests/deals.

4. Pet owners should be able to make a deal with specific details (time period, pet, food, personal preferences and a mobile application to keep in touch as a description), which the Pet sitters will need to accept or decline.

5. The users won't be able to make a deal for more than 2 weeks at a time.

6. After the service is completed, Pet owners should be able to leave ratings for the pet caretaker.

7. The system should contain a forum, where each user will be able to make a post (with an option to add a photo), ask a question related to their pets health, well being and behavior (with an additional option to get responses just from vets if desired), and get comments on it from more experienced users and veterinarians.

8. The system should send notifications and reminders to users about the status of their deals, upcoming pet care appointments, questions on the forum (vets), change of their rating.

9. The system can only be used by registered users. Non-registered users will be able to see just the basic information about the system (what is it used for and instructions on how it works).

    9.1. While registering each vet user will have to go through a procedure, waiting for the system admin to confirm the profile.

Non-Functional Requirements

Table 1: Non-functional requirements

Requirement Type	Explanation
Performance	☐ Since it will only be available in Slovenia, the system must support a minimum throughput rate of 500 concurrent users at a time. ☐ The system should respond to user interactions within 2 seconds for normal operations.
Information	☐ Pet-related information, like vet forum posts, should always be up to date, changing as soon as something new is added. ☐ Deal history should be kept current, with any new deals or changes showing up within 24h.
Economy	☐ The development should be completed within 10 months from the start of the development process (November, 2024).
Control (and Security)	☐ The system should include authentication and authorization mechanisms. ☐ The system should implement robust security measures to protect user data and payment information.
Service	☐ The system will be available for people who are passionate about pets, located in Slovenia. ☐ There will be 2 main user types: Pet owners and Pet sitters. ☐ Having in mind that the system will not be the main way of keeping people in touch, the system must maintain an uptime of at least 95%. ☐ The system will be accessible as a website, requiring no app downloads. It will be primarily optimized for computers and devices with larger screens, ensuring an excellent user experience for users accessing the website on desktops and tablets. ☐ The system will include a section with easy instructions, and FAQs to help users clarify all functionalities.

Feasibility study

Technical Feasibility

The proposed solution is technically feasible, utilizing Node.js and Express for efficient and scalable web development. The choice of SQL for the database aligns with industry standards, and hosting the server on faculty infrastructure ensures reliability and performance. Additionally, incorporating modular components allows for easy future enhancements and adaptability to evolving project needs.

Operational Feasibility

Operationally, the solution is feasible, with a focus on aligning the platform with Slovenian laws and regulations related to pet care, privacy, and data security. Additionally, understanding and accommodating cultural preferences and practices specific to Slovenia, and the whole Eastern Europe will be critical for user engagement and acceptance. Establishing clear agreements and protocols with pet sitters, walkers, and other stakeholders further supports operational feasibility.

Economic Viability

Economically, the proposed system is viable within the Slovenian market. The pet care industry in Slovenia presents opportunities for revenue generation through service fees, subscriptions, and potential partnerships. The decision to integrate with an external payment system reduces development costs and leverages existing reliable solutions. A comprehensive financial analysis, accounting for local market dynamics and potential growth, will provide a clearer picture of the system's economic viability.

Organizational and Social Acceptability

Organizational and social acceptability are likely high given the system's alignment with the needs of Slovenian pet owners and service providers. Socially, the system caters specifically to the Slovenian audience, addressing their unique preferences and practices in pet care. Collaborating with local veterinary professionals and ensuring user-friendly features will contribute to the system's acceptance within the community.

Logical design

Table 2: Matrix user role/functions

Functions	Pet Owner	Pet Caretaker	Veterinarian	Administrator
Create a profile	✔	✔	✔	✘
Modify/Delete the profile	✔	✔	✔	✘
Receive notifications (mails)	✔	✔	✔	✔
Create posts	✔	✔	✔	✘
Comment on posts for vets	✘	✘	✔	✘
Comment	✔	✔	✔	✘
Delete own posts and comments	✔	✔	✔	✘
Delete post (all)	✘	✘	✘	✔
Delete comments (all)	✘	✘	✘	✔
Approve vet profiles	✘	✘	✘	✔
Make a deal	✔	✔	✔	✘
Accept/Reject a deal	✘	✔	*depends	✘
Leave a review	✔	✘	*depends	✘
Create listing for pet sitting	✘	✔	*depends	✘
Delete own listings	✘	✔	*depends	✘
Delete listings (all)	✘	✘	✘	✔
Filter listings	✔	✔	✔	✔
Sort posts and listings	✔	✔	✔	✔

*This depends on the type of the user, either Pet Owner or Pet Sitter/Caretaker. A veterinarian can be anyone

Entity Relational Diagram (ERD)

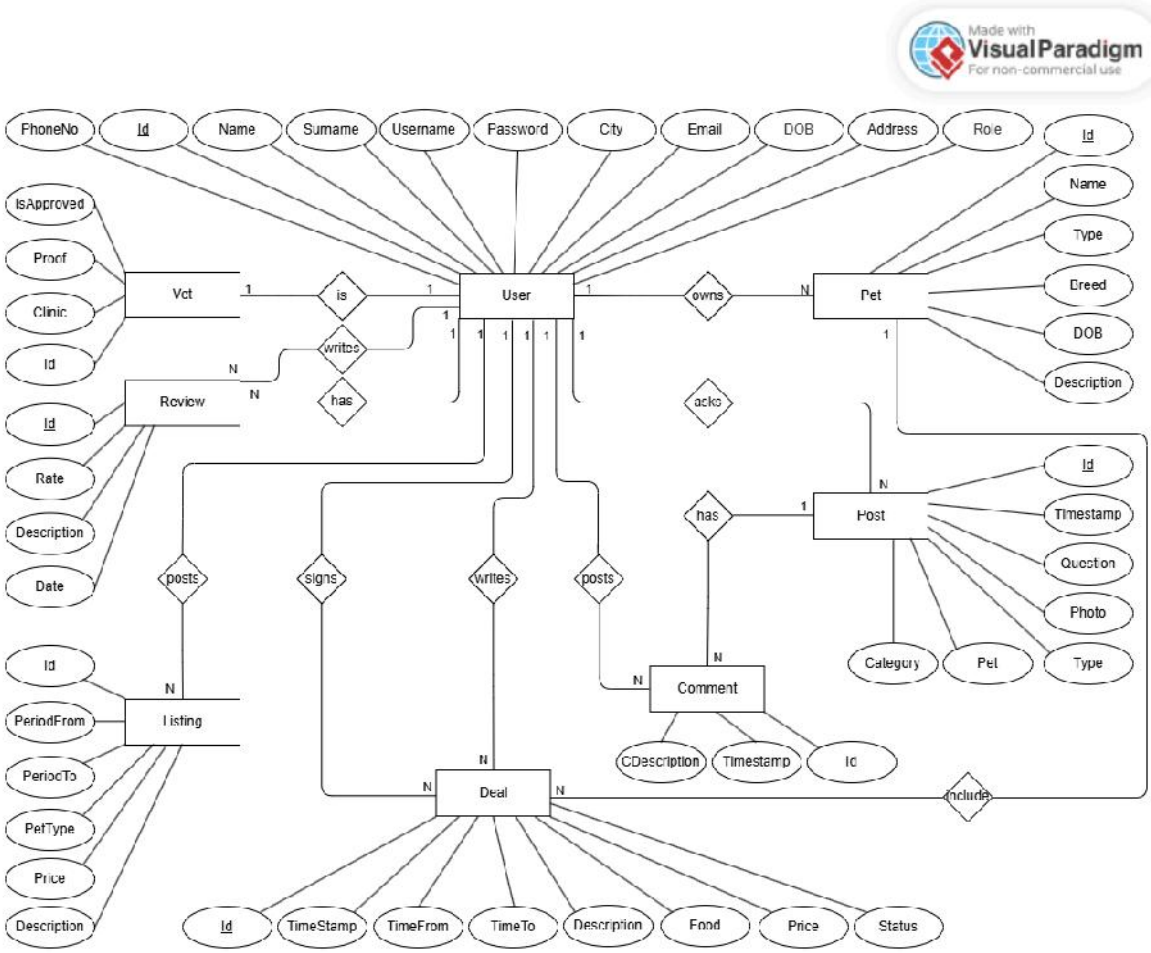

Figure 1: Entity relational diagram

Note: The ResetPasswordToken entity is not included in the ER diagram because it is not specific to the core functionality of the system and the system does not directly depend on it. However, it is included in the supplementary diagrams below for completeness.

Data Dictionary

Table3: Data dictionary

Entity	Description	Attribute	Type	Description of attribute
User	User of the system	Id (PK)	int(11)	Identity of the user
		Name	varchar(24)	User's first name
		Surname	varchar(24)	User's last name
		Username	varchar(24)	User's username
		Password	varchar(24)	User's password
		City	varchar(15)	City that is closest to the user (chosen from selected options)
		Email	varchar(320)	User's email address
		DOB	date	User's date of birth
		Address	varchar(60)	User's home address
		PhoneNo	varchar(9)	User's mobile phone number
		Role	varchar(9)	User's role (Petsitter or Petowner)
Vet	Veterinarian	Id (PK)	int(11)	Identity of the vet
		Clinic	varchar(30)	Vet's work place
		Proof	longblob	A scan of the veterinarian's diploma/certificate
		IsApproved	tinyint(1)	Admin's confirmation that the person is a veterinarian
Pet	User's pet	Id (PK)	int(11)	Identity of the pet
		Name	varchar(24)	Name of the pet
		Type	varchar(7)	Animal type of the pet
		Breed	varchar(48)	Breed type of the pet
		DOB	date	Date of birth of the pet
		Description	varchar(500)	Pet's behavior description
Review	User's review	Id (PK)	int(11)	Identity of the review
		Rate	tinyint(1)	Number rate/rating of the review (1 to 5)
		Date	date	The date on which the review is posted
		Description	varchar(500)	Review's explanation
Post	A question posted on the forum	Id (PK)	int(11)	Identity of the post
		Timestamp	timestamp	The time when the post is created
		Question	varchar(250)	The question asked

		Photo*	longblob	Photo posted with the question
		Type	tinyint(1)	To indicate whether the question is asking only the vets
		Pet	varchar(7)	Pet the post is about
		Category	varchar(8)	Category the post is about
Comment	Comment on a post from the forum	Id (PK)	int(11)	Identification of the comment
		CDescription	varchar(250)	The comment written
		Timestamp	timestamp	The time when the comment is created
Deal	Specific deal between two people and a pet	Id (PK)	int(11)	Identity of a deal
		TimeStamp	timestamp	The time when the deal is signed
		TimeFrom	date	The date when the service starts
		TimeTo	date	The date when the service ends
		Description	varchar(500)	Description of everything that is important for the Pet Sitter
		Food	varchar(24)	The pet's food preference
		Price	float	The price for which both sides have agreed
		Status	varchar(10)	Status of the deal (sent, accepted, rejected, done)
Listing	A listing created by Pet caretaker	Id (PK)	int(11)	Identity of a listing
		PeriodFrom	date	The date when the listing is valid from (start date)
		PeriodTo	date	The date when the listing is valid to (end date)
		PetType	varchar(7)	Types of the pets that are included
		Price	float	The price range for the service provided (it usually depends on the number of days and type/breed of pet)
		Description	varchar(500)	Description about the care, the place, everything that would be useful and important to know
PasswordResetTokens	Tokens for users that want to change their passwords	Id (PK)	int(11)	Identity of the token
		Token	varchar(255)	Token string
		ExpiresIn	bigint(20)	Time till when it is valid

* attributes marked with the asterisk symbol "*" are not mandatory

Relational Model

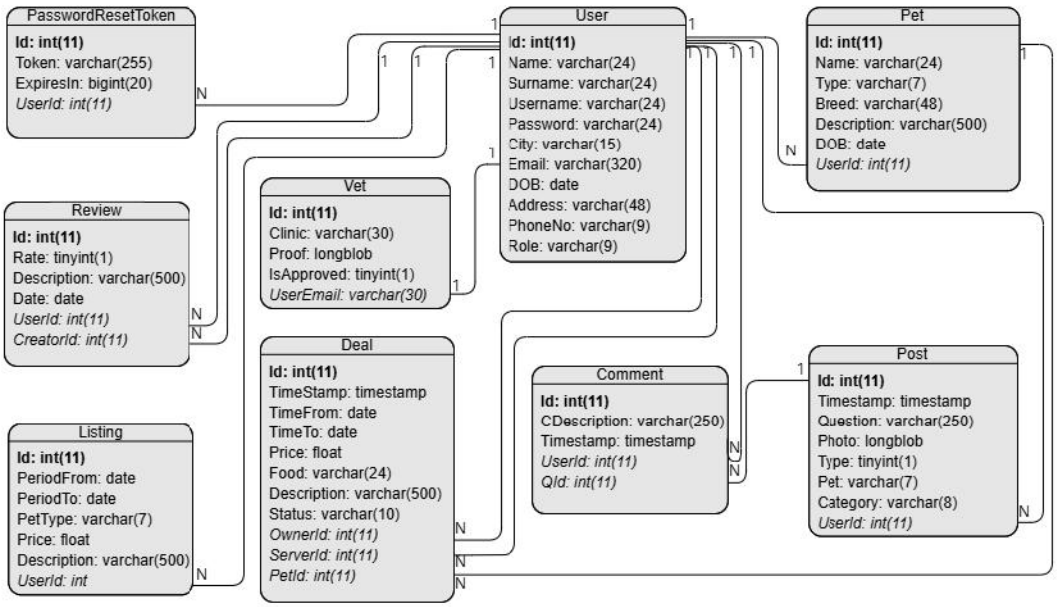

Figure 2: Relational model

This model is normalized to NF3.

Object-oriented analysis

UML Class Diagram

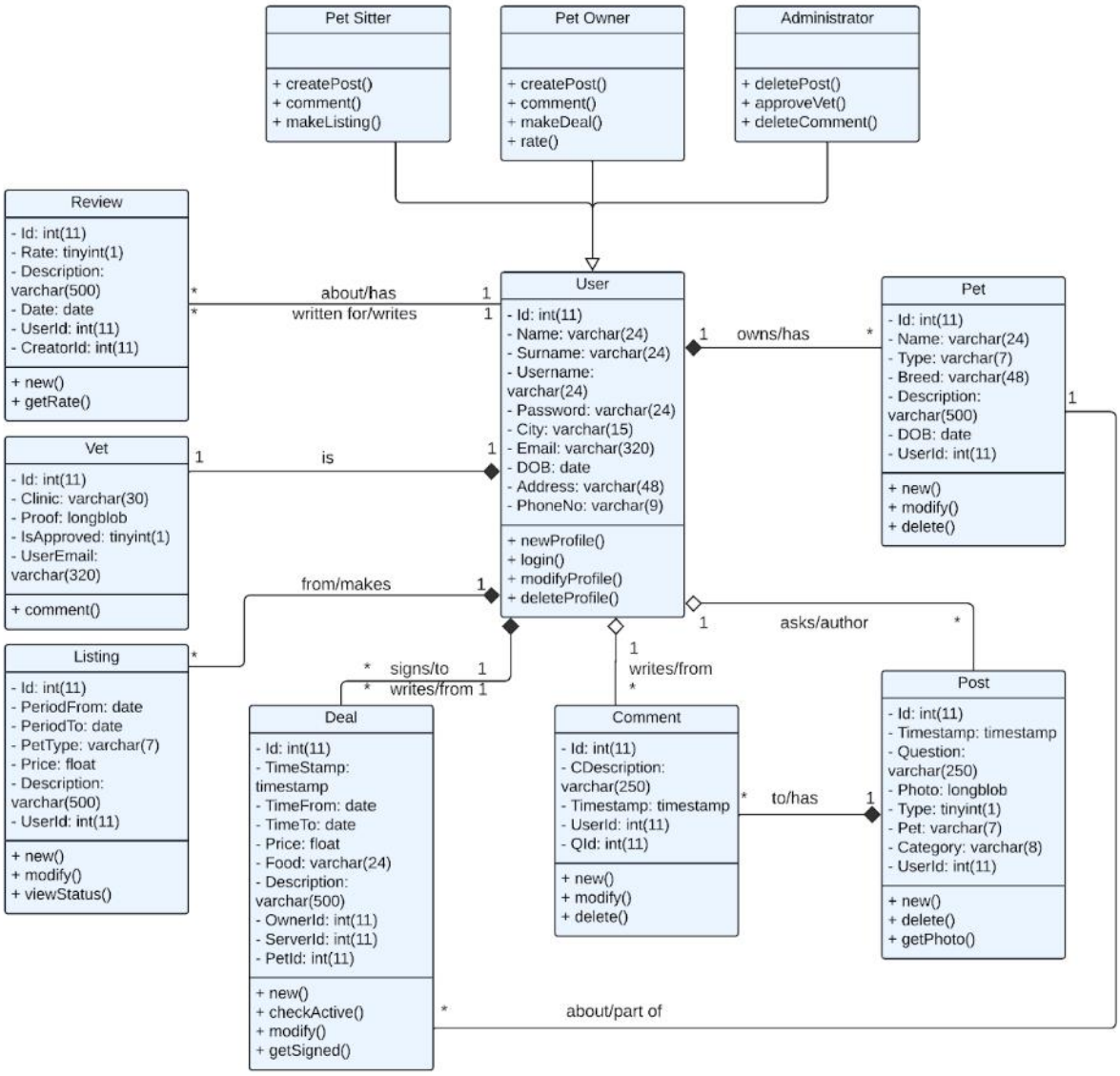

Figure 3: UML class diagram

UML Sequence Diagrams

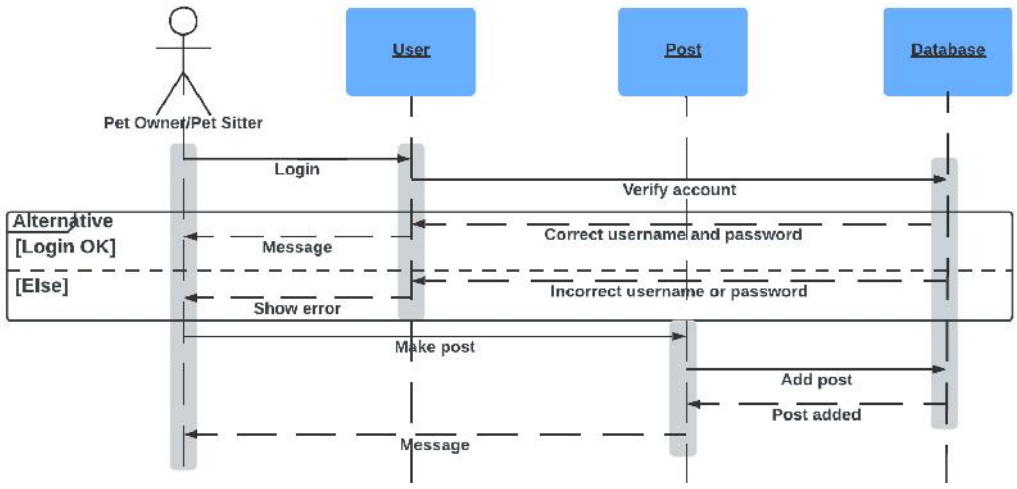

Figure 4: UML sequence diagram for making a post

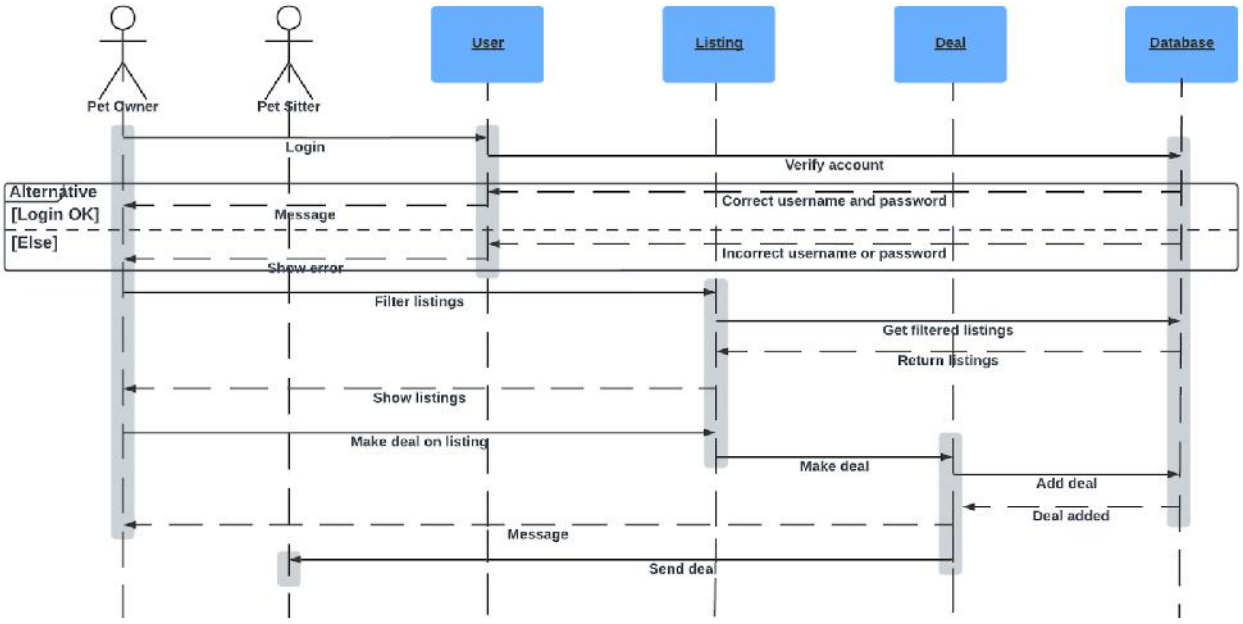

Figure 5: UML sequence diagram for making a deal

Process modeling

Functional Decomposition Diagram

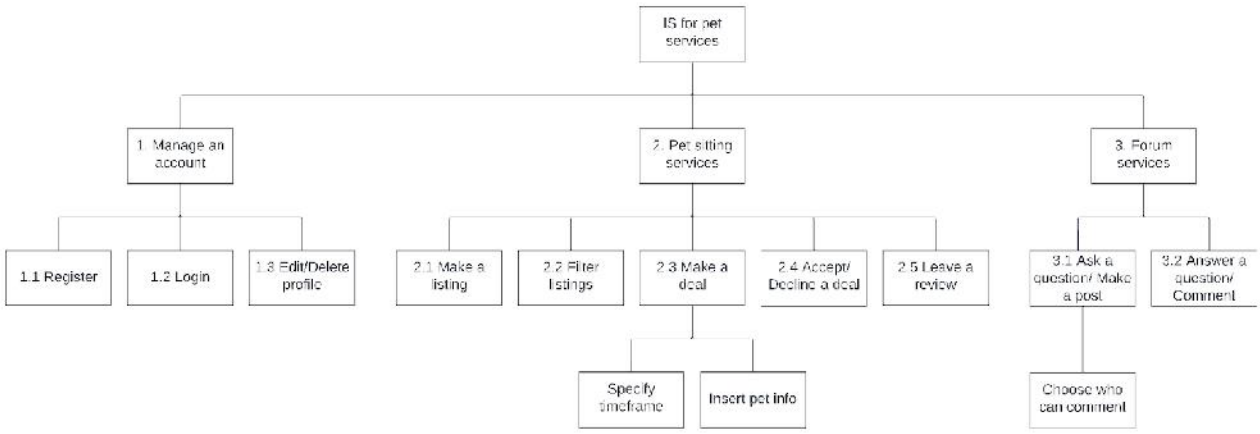

Figure 6: Functional decomposition diagram

Data Flow Diagrams

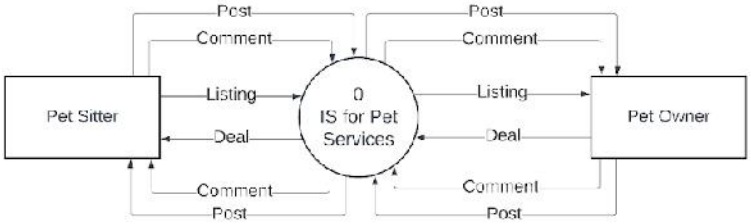

Figure 7: Contextual data flow diagram

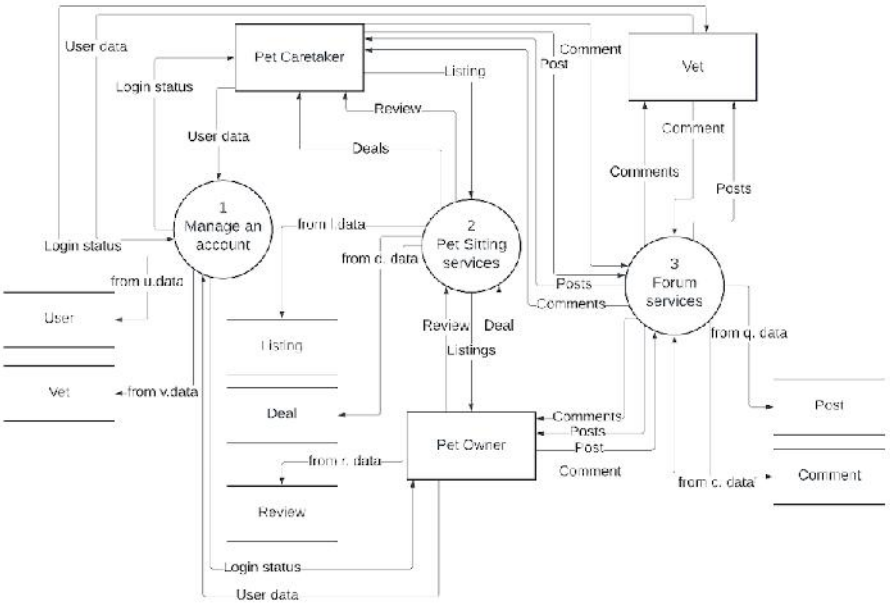

Figure 8: System level data flow diagram

Physical design

Physical Data Model

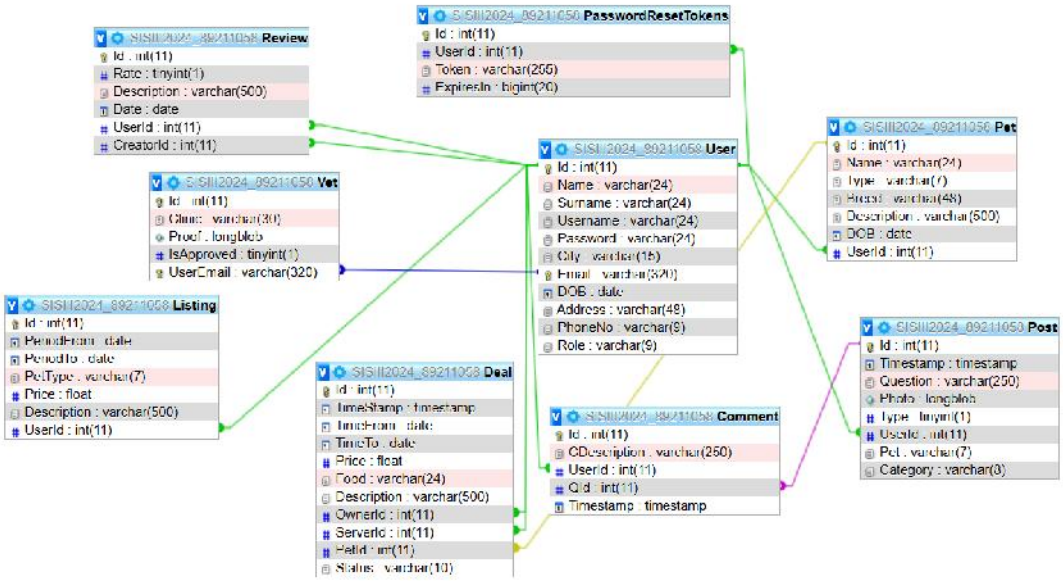

Figure 9: Physical data model