

Software Engineering  
 Phase III

By: Helisa Ormëni

Saeda Haka

Epoka University, Tirana Albania 2024

* Software design and modeling

**Group name:** Pet Care Vet

**Software Architecture**

* System Architecture

1. **Client Interface:** 
   1. This component represents the front-end interface accessible to users (pet owners) and admins.
   2. It includes user interfaces for viewing vet profiles, booking vaccine appointments, and managing vet profiles and appointment schedules (for admins).
   3. Technologies: HTML, CSS, JavaScript
2. **User Management Microservice:**
   1. Manages user authentication, registration, and authorization.
   2. Stores user data and authentication tokens.
   3. Technologies: Node.js, Express.js, MongoDB or PostgreSQL for data storage.
3. **Vet Profile Microservice:**
   1. Handles vet profiles, including creation, retrieval, updating, and deletion.
   2. Stores vet profile data and qualifications.
   3. Technologies: Node.js, Express.js, MongoDB, or PostgreSQL for data storage.
4. **Appointment Management Microservice:**
   1. Manages vaccine appointment scheduling.
   2. Allows users to book, reschedule, and cancel appointments.
   3. Notifies users and vets about upcoming appointments.
   4. Technologies: Node.js, Express.js, MongoDB or PostgreSQL for data storage.
5. **Admin Panel:**
   1. Provides interfaces for admin users to manage vet profiles and appointment schedules.
   2. Allows admins to add, edit, or remove vet profiles and appointments.
   3. Technologies: HTML, CSS, JavaScript, back-end APIs for data manipulation.
6. **Database:**
   1. Stores all application data, including user information, vet profiles, appointment schedules, etc.
   2. Technologies: MySQL, XAMPP

* Component diagram

**Sequence Diagram**

A white sheet of paper with black text

Description automatically generated

1. The client interface sends a request to view vet profiles.

2. The user management microservice receives the request and authenticates the user.

3. Upon successful authentication, the user management microservice requests the vet profile microservice to retrieve the vet profile data.

4. The vet profile microservice processes the request and sends back the vet profile data to the user management microservice.

5. The client interface displays the vet profile to the user.

6. The user requests to book an appointment.

7. The user management microservice checks the availability of the vet for appointments.

8. The appointment management microservice retrieves available slots for the vet.

9. Upon selecting a slot, the appointment management microservice books the appointment.

10. The appointment management microservice confirms the booking.

11. The client interface displays a confirmation message to the user.

**Database Design:**

Table: Reservation

id (Primary Key): Unique identifier for each appointment.

pet\_owner\_name: Name of the pet owner booking the appointment.

vet\_id: References the unique identifier of the vet assigned to the appointment.

reservation date: Date of the appointment reservation.

reservation\_time: Time of the appointment reservation.

status: Indicates the status of the appointment (e.g., confirmed, pending, canceled).

This design ensures that data is organized efficiently with minimal redundancy. Each appointment is uniquely identified by its ID, and the table captures all relevant details related to the appointment, including the pet owner's name, the assigned vet, reservation date and time, and appointment status.

* **Modeling**

A screenshot of a computer program

Description automatically generated

In this activity diagram:

- The process starts with the user initiating the appointment booking.

- The user views vet profiles and selects a vet profile.

- The user books an appointment with the selected vet.

- The system confirms the appointment.

- Finally, the appointment booking process ends.