Task #1

This project models a ***Pet Adoption Management System*** that supports multiple user roles and coordinates both the management of animals and the end-to-end adoption process.

The **primary purpose** of the system is to simplify the adoption process by maintaining accurate animal records, processing applications transparently, and ensuring that each adoption is ethically and medically approved. The system also ensures data integrity, secure access control, and traceability across all operations, from animal registration to adoption contract generation

The system involves several **interacting roles** that reflect different perspectives and responsibilities within the adoption process. Each **actor** performs specific operations that contribute to the overall workflow, as described below.:

- **Adopter**
    - The *Adopter* is an external user who interacts with the system to find and adopt animals. They can register, browse the list of available pets, filter results based on preferences (species, breed, or age), and submit adoption applications. The adopter can also upload required documents (such as identification or proof of residence) and track the status of their applications.
- **Shelter worker**
    - The *Shelter workers* are internal users responsible for managing animals within the shelter. Their tasks include registering new animals, updating animal profiles (health status, breed, or availability), and reviewing adoption applications submitted by adopters. They can approve or reject adoption requests, track the status of animals (available, reserved, adopted, or on medical hold), and maintain up-to-date shelter records.
- **Veterinarian**
    - The *Veterinarian* is responsible for maintaining the medical records of animals. They add new medical entries, record vaccination history, treatments, sterilization data, and monitor animals' readiness for adoption. Veterinarians can also create emergency medical entries when immediate intervention is required. Once an adoption is approved, they can generate or verify the animal's health certification included in the adoption contract.
- **System Administrator**
    - The *System Administrator* manages the technical and organizational aspects of the system. Their responsibilities include configuring system settings, managing user accounts, assigning roles, and performing data backups. They also generate statistical reports on adoption rates, animal health, and shelter performance. Administrators ensure that user access rights align with organizational policies and that all operations are securely logged.

Core system structure (classes and relationships):

The system's foundation lies in the **class model**, which defines the structure of data and their relationships.

The **User** class is an abstract entity from which all user roles-*Administrator*, *ShelterWorker*, *Veterinarian*, and *Adopter*-inherit. Each user has common attributes such as userID, name, email, phone, registrationDate, and role. Specialized subclasses extend this base with their specific data and operations (for example, the *Administrator* class includes operations like generateSystemReport() and manageUserAccounts()).

The **Animal** class represents individual pets available for adoption, including attributes such as animalID, name, species, breed, gender, dateOfBirth, temperament, and status. The status is defined by the **AdoptionStatus** enumeration (AVAILABLE, RESERVED, ADOPTED, MEDICAL_HOLD).

Each animal belongs to exactly one **Shelter**, while each shelter may host multiple animals (1 to * relationship). The shelter also maintains capacity, contact details, and availability metrics.

Each **Animal** can have multiple **MedicalRecords**, each created and maintained by a *Veterinarian*. A *MedicalRecord*stores data about the animal's health status, medical treatments, and notes by the vet.

The **AdoptionRequest** class links *Adopters* and *Animals* by capturing the adopter's motivation, request date, and application status. Once a request is approved, an **AdoptionContract** is generated, binding one adopter and one animal in a finalized legal adoption.

Additionally, the *Animal* class has two subclasses:

- **Dog**, which adds the attribute isTrained : Boolean.
- **Cat**, which adds the attribute isIndoorCat : Boolean.

This structure supports species-specific data without redundancy, ensuring scalability and clarity within the model.

Functional overview (Use case categories):

The **Use Case Model** illustrates the interactions between system actors and main functional processes. Instead of describing each use case in full detail, the system's functionality can be grouped into four main categories:

1. Adopter-Related Operations

Adopters can interact with the system through the following features:

- **User Registration** - create a personal account.
- **Browse Available Pets** - view and filter animals based on species, breed, or status.
- **Submit Adoption Application** - apply to adopt a chosen animal.
- **Upload Supporting Documents** (*extend relationship*) - attach identification or proof of eligibility to an adoption request.

These actions use *include* and *extend* relationships (e.g., “Browse Available Pets” includes “Search Pets” and “Filter Results”).

2. Shelter Management Operations

Shelter Staff manage the day-to-day operations through:

- **Register New Animal** - add new animals with all relevant data.
- **Update Animal Profile** - edit animal information and change availability status.
- **Review and Approve/Reject Adoption Applications** - assess adopter requests and finalize decisions.
- **Track Animal Status** - monitor availability and movement of animals between shelters or homes.

3. Veterinarian Operations

Veterinarians handle:

- **Manage Medical Records** - record treatments and vaccination data.
- **Emergency Medical Entry** (*extend relationship*) - create urgent medical updates when necessary.
- **Generate Adoption Contract** - verify animal health and provide approval for adoption completion.

4. Administrative Operations

System Administrators perform high-level management tasks, such as:

- **System Configuration** – adjust system parameters and access rules.
- **Manage User Profiles** – create, update, or deactivate user accounts.
- **Role-Based Management** – control user permissions.
- **Generate Reports** – produce adoption statistics and shelter performance summaries.
- **Backup Database** – ensure data integrity and disaster recovery.

Integration between structure and function

Both the class and use-case models reflect different perspectives of the same system:

- The **class model** represents the *static structure* - how entities such as *User*, *Animal*, and *AdoptionRequest* are related.
- The **use case model** represents the *dynamic behavior* - how users interact with the system to perform specific actions.

The integration of inheritance, associations, and enumerations in the class model directly supports the interactions defined in the use case model. For example:

- The AdoptionRequest and AdoptionContract classes enable the *Submit Application* and *Generate Contract* use cases.
- The MedicalRecord class supports the *Manage Medical Records* process.

- The AdoptionStatus enumeration allows the *Track Animal Status* operation.

This consistency ensures that the functional and structural aspects of the system remain aligned.

In general, the *Pet Adoption Management System* provides a unified platform that connects all key participants in the adoption process - adopters, shelter staff, veterinarians, and administrators.
It combines a clear class structure with a comprehensive set of functional use cases, ensuring that every process - from animal intake to contract signing - is transparent, efficient, and secure.

The design follows best modeling practices, combining structural and behavioral modeling elements such as generalization, associations, enumerations, and use-case relationships (*include* and *extend*). Additionally, it integrates sustainability principles that reinforce both organizational efficiency and ethical animal care.