FINAL PROJECT MILESTONE 1

Ujwal Chandrashekar **Group 2**

Shreyas Pogal Naveen

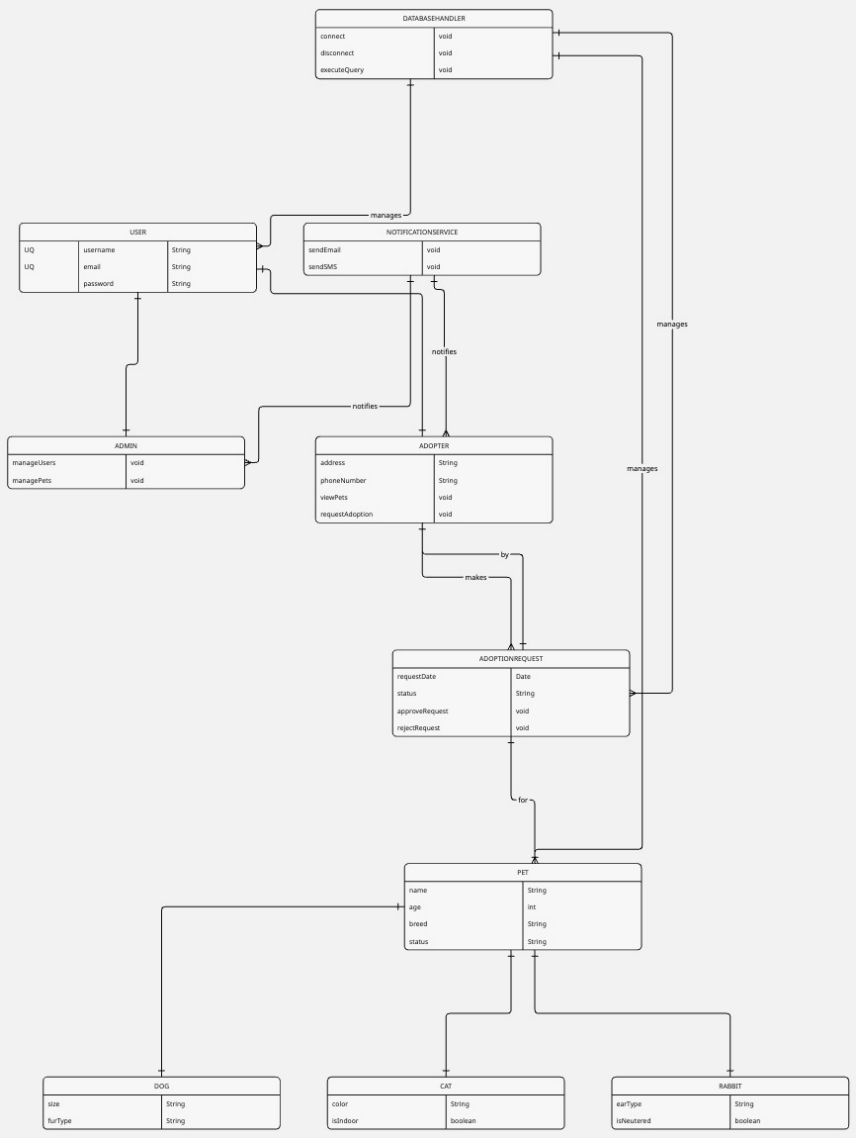
Project Topic: Pet Adoption Management System

Problem Statement

Many animal shelters face challenges in efficiently managing pet adoption processes. Adopters often find it difficult to search for pets, while shelter staff struggle to keep track of pet availability, adoption requests, and user details. This project aims to create a Pet Adoption Management System that allows shelters to add and manage pet listings and users to search for pets and apply for adoption online. The system will simplify the process through a user-friendly interface, streamline the approval workflow for staff, and improve adoption success rates by automating and organizing the adoption process.

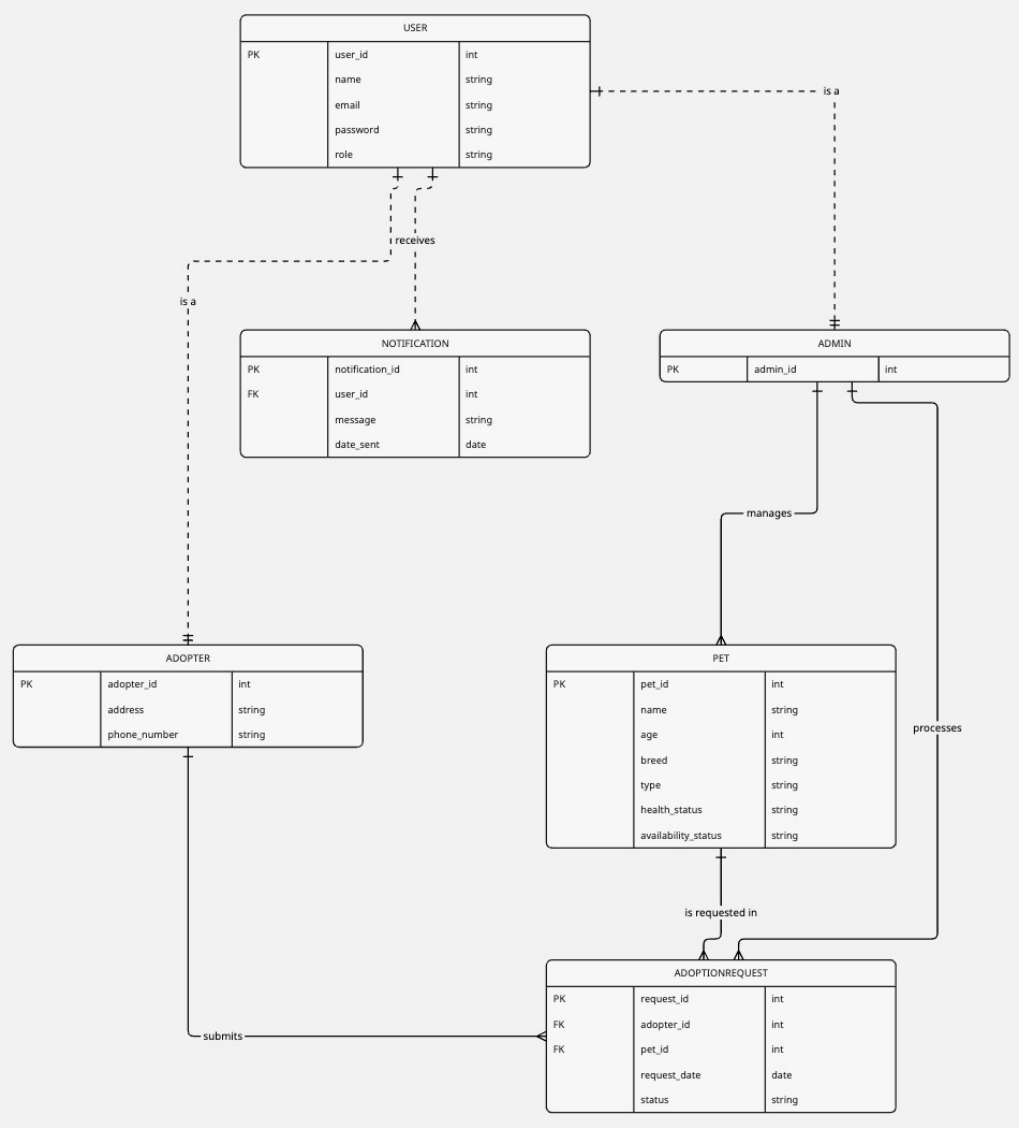
UML Diagram

The UML Class Diagram models the object-oriented structure of the Pet Adoption Management System. It outlines the key classes such as User, Admin, Adopter, Pet, AdoptionRequest, NotificationService, and DatabaseHandler. The diagram highlights inheritance relationships (e.g., Admin and Adopter are specialized forms of User) and clearly defines class attributes and methods. It also showcases class interactions such as how Adopters submit adoption requests, Admins manage pets, and NotificationService notifies users. This diagram emphasizes the use of core OOP principles like abstraction, inheritance, encapsulation, and method definition in line with Java development practices.



ER Diagram:

The Entity-Relationship (ER) Diagram represents the database structure of the Pet Adoption Management System. It includes entities such as User, Admin, Adopter, Pet, AdoptionRequest, and Notification, along with their primary keys, attributes, and foreign key relationships. The diagram captures how users receive notifications, adopters submit adoption requests, and admins manage and process pets. The use of "is-a" relationships denotes inheritance, while labeled associations like submits, manages, and processes define clear relational flows. This ER diagram provides a normalized view of how data is organized and related across the system.



Object-Oriented Concepts

1. Abstraction – Abstract class for defining pet attributes (type, age, health status).
2. Inheritance – Base class Pet inherited by Dog, Cat, and Rabbit.
3. Polymorphism – Overloading and overriding methods for adding and displaying pet details.
4. Encapsulation – Using private attributes and public getter/setter methods.
5. Composition – AdoptionRequest will be composed of User and Pet objects.
6. Interface – An interface for defining notification strategies (email, SMS).

Tech Stack

1. Backend: Java
2. Frontend: HTML, CSS
3. Data Handling: CSV files for storing pet and user data
4. Database: CSV file storage
5. IDE: Eclipse

Functionalities for Milestone 2

1. User Registration and Login: Create and authenticate users (Admin and Adopter).
2. Add and Display Pets: Admin should be able to add, edit, and delete pet records.
3. Search and Filter: Users should be able to search for pets by breed, age, and type.
4. Adoption Request: Allow users to apply for adoption.
5. Notification System: Notify users about the status of their adoption requests.

Contributions

1. Ujwal Chandrashekar – Backend development (User authentication, Adoption process)
2. Shreyas Pogal Naveen – Frontend development (HTML, CSS)
3. Ujwal Chandrashekar – Data Handling (CSV file management, Database connection)
4. Shreyas Pogal Naveen – Testing and debugging