Pet Adoption System

Title	**Pet Adoption System**
Description	The Pet Adoption System is designed to streamline the process of adopting pets from shelters. This system will manage pet profiles, adopter information, and the adoption process, ensuring a smooth and efficient experience for both the shelters and the adopters. The system will also handle notifications and updates regarding the adoption status. The main entities of the Pet Adoption System project are **(may be more according to your implementation and logic)** : 1. Pet ○ Attributes**:** ID, name, species, breed, age, health status. 2. Adopter ○ Attributes**:** ID, name, contact information, adoption history. 3. Shelter ○ Attributes**:** ID, name, location, contact information. 4. Adoption ○ Attributes: ID, pet, adopter, adoption date, status. 5. User ○ Attributes**:** ID, username, password, role (admin, adopter).
Deliverables	1. Class diagram for the project (Printed) 2. Java project includes all classes and functions in the description. 3. Documentation that contains a. your own system description b. input and output scenarios
Bonus	1. **Powerful GUI** 2. **Advanced Search:** Implement advanced search functionality for pets based on various criteria (age, breed, ...). 3. **Notification System**: Implement a system to notify adopters about the status of their requests (approved or denied) via email (or simple notifications within the system). 4. **Analytics:** Provide analytics and reports on adoption trends and shelter performance.

System Functionalities	**1. User Management:** **a. Register and manage user profiles (admin, adopter).** **b. Login and logout functionality.** **2. Pet Management:** **a. Add, edit, and delete pet profiles.** **b. Display pet details including health status and adoption availability.** **3. Adopter Management:** **a. Add, edit, and delete adopter profiles.** **b. Track adoption history and status.** **4. Shelter Management:** **a. Add, edit, and delete shelter information.** **b. Manage the list of pets in each shelter.** **5. Adoption Process:** **a. Allow adopters to request pet adoptions.** **b. Admins can approve or reject adoption requests.** **c. Track the status of each adoption request.** **6. Notifications(Bouns):** **a. Send notifications to users about adoption status and updates.** **b. Display notifications in the user interface.** **7. Reporting (Bouns):** a. Number of Adoptions: i. Number of adoptions per shelter. ii. Monthly or yearly adoption trends. b. Adopter Statistics: i. List of adopters with the highest number of adoptions. ii. Demographic information of adopters (age, location, etc.). c. Pet Statistics: i. Most adopted species and breeds. ii. Average time pets spend in the shelter before adoption. d. Shelter Performance: i. Number of pets currently available for adoption in each shelter. ii. Adoption rate per shelter. e. Adoption Request Statistics: i. Number of adoption requests received and processed.

What System Should do	1. Manage User Accounts: Allow users to create and manage their accounts, ensuring secure access to the system. 2. Handle Pet Profiles: Enable shelters to add and update pet profiles, including detailed information about each pet. 3. Facilitate Adoptions: Streamline the adoption process by allowing adopters to request adoptions and admins to manage these requests. 4. Send Notifications: Keep users informed about the status of their adoption requests and other relevant updates. 5. Generate Reports: Provide insights into adoption trends and shelter performance through detailed reports and analytics.
Notes	1. You should implement all concepts of OOP. 2. Each member **MUST** work on at least one of the required classes besides file processing or GUI. (**Individual marks**) 3. The evaluation will be mainly based on the student's ability to use and apply OOP concepts and the explanation of the code. 4. You must deliver the Class Diagram for the project. 5. You must apply exception handling. 6. Using Files is mandatory (Not Database) 7. Any project must have at least **8 classes** 8. Regarding **files:** - You must have only two functions for file reading and writing. - You should read data once at the beginning of your run then do your operations and access the code then save in files at the end of your program.