PetApp Adoption - System Description

Introduction

The PetApp Adoption System is a comprehensive application designed to facilitate the process of pet adoption. It provides a user-friendly interface for both adopters and administrators, ensuring efficient management of shelters, pets, and adoption requests. The system is built using JavaFX and includes robust functionalities to manage users, pets, adoptions, notifications, and reports.

System Components

AdoptionPetApp Class

The AdoptionPetApp class is the main entry point for the Pet Adoption Application, extending JavaFX's Application. It initializes data, establishes relationships between users, pets, shelters, and adoptions, and launches the user interface.

Key Attributes

users: List of registered users (User.txt).

pets: List of available pets (Pet.txt).

shelters: List of shelters (Shelter.txt).

adoptions: List of adoption records (Adoption.txt).

notifications: List of notifications (notification.txt).

ActiveUserid: Current user's ID.

indexActiveUser: Current user's index.

Key Methods

start: Displays the login screen (login.fxml).

main:

Loads data from files.

Links pets, adoptions, adopters, and shelters.

Launches the application.

1. User Class

Represents a general user in the system. Stores user details like ID, username, password, and role. Handles login functionality and logout event.

Key Methods:

- Log_in: Verifies user credentials.

- logout: Logs the user out and redirects to the login page.

- delete_account: Abstract method to delete the user's account, to be implemented in subclasses.

2. Admin Class (extends User)

Represents an admin user with additional privileges to manage shelters and pets.

Key Attributes:

- Associated shelter and shelter ID.

Key Methods:

- Sing_up: Registers a new admin and associates a shelter.

- add_shelter: Creates a new shelter and links it to the admin.

- Editadmin: Updates the admin's username and password.

- delete_account: Deletes the admin's account and associated shelter.

3. Shelter Class

Represents a shelter that contains pets and is managed by an admin.

Key Attributes:

- Shelter details (ID, name, location, contact info), pets, and admin ID.

Key Methods:

- add_Pet: Adds a new pet to the shelter.

- EditShelter: Updates shelter details.

- DeleteShelter: Removes the shelter and its pets from the system.

- toString: Returns a formatted string representation of the shelter's details.

4. Pet Class

Represents a pet in a shelter, with attributes like name, species, breed, age, health, and availability.

Key Methods:

- Remove_Pet: Marks a pet as unavailable and removes it from the shelter.

- UpdatePet: Updates the pet's details like name, species, breed, age, and health.

- toString: Returns a formatted string representation of the pet's details.

5. GeneralFunctions Class

Provides essential utility functions for managing file operations and generating unique IDs. Ensures structured data storage and retrieval across the application.

Key Methods:

- WriteInFile: Serializes entities (e.g., adopters, pets, shelters, etc.) and writes them to a file.

- ReadFromFile: Deserializes data from a file to retrieve stored entities.

- GenerateUniqueID: Generates unique identifiers for users, shelters, pets, adoptions, and notifications.

6. Adoption Class

Manages adoption processes by linking adopters with pets and tracking adoption statuses and details.

Key Attributes:

- Adopter and pet details.

- Adoption date and status.

Key Methods:

- editAdoption: Updates adoption status or pet availability.

7. Adopter Class (extends User)

Represents individuals seeking to adopt pets, extending general user functionality with adoption-related attributes.

Key Attributes:

- Contact information, age, location, and adoption history.

Key Methods:

-Sing_up: Facilitates adopter registration.

- editProfile: Updates adopter details.

- deleteAccount: Deletes adopter account from the system.

-compareTo: Implements Comparable based on Adoption history length for reporting.

8. Report Class

Provides detailed statistics and analytics to support shelter management and decision-making.

Key Attributes:

- Total adoption requests.

- Accepted and ongoing requests.

- Adoption success rate.

- Number of available pets.

- Most active adopters.

Key Methods:

- Generates reports with detailed metrics to enhance adoption processes.

9. Search Class

Provides filtering functionality to search for pets based on specific criteria like name, age, breed, and species.

Key Attributes:

- filterpets: A list to store filtered pets.

Key Methods:

- searchprocess: Filters a given list of pets using query parameters like name, age, breed, and species.

10. Notification Class

Manages notifications sent to admins or adopters regarding adoption requests and their status.

Key Attributes:

- notifiId: Unique identifier for the notification.

- sendTOId: Target user ID (admin or adopter).

- username: Username of the sender.

- isOpened: Boolean flag indicating whether the notification has been viewed.

Key Methods:

- sendadminnotif: Displays a popup notification for admins about a new adoption request.

- sendadopternotif: Displays a popup notification for adopters regarding the status of their adoption request.

- Getters and setters for accessing and modifying notification details.

- toString: Returns a string representation of the notification object.

11. Controllers

Manages user interface actions and interactions within the application.

Key Controllers:

- adoptionrequest Controller: Handles adoption requests, including viewing, accepting, or rejecting requests.

- ReportController: Displays adoption-related reports and statistics, including active adopters and shelter metrics.

- LoginController: Manages login and authentication processes.

- AdminMenuController: Facilitates admin dashboard functionalities like pet management and navigation.

- AdminProfileController: Allows admins to manage their profile and shelter details.

-AdopterMenuController: Manages adopter actions like viewing and adopting pets, searching, accessing profiles, and managing adoption history.

-AdopterProfileController: Handles profile management, including editing personal information, navigation, and account deletion.

-AdoptionHistory: Manages the adopter's adoption history, allowing navigation to profile and main menu. It displays adoption details in a table and handles account deletion and logout processes.

-signup1controller: Handles navigation between admin and adopter signup screens, and manages application exit, saving data to files.

-Signupadoptercontroller: Manages adopter signup, validates input, and navigates to the login screen. Also handles application exit and data saving.

-Signupadmincontroller: Handles the admin signup process, validates input, and navigates to the login screen upon successful signup. Also manages application exit, saving data to files.