Use Case Documentation - Dog Rescue Application

This document outlines the key use cases for the Dog Rescue Application, detailing how different users interact with the system to achieve their goals.

1. Actors

The primary actors interacting with the system are:

- **Public User / Site Visitor:** Any individual accessing the public-facing website. This includes potential adopters, people wanting to report a found dog, or individuals interested in volunteering.
- **Administrator (Admin):** A privileged user responsible for managing the application's content and operations, such as dog profiles, rescue submissions, and volunteer applications.

---

2. Use Cases

Below are the detailed use cases, organized by actor.

2.1 Public User / Site Visitor Use Cases

These users can access the public parts of the website.

UC-001: View List of Available Dogs

- **Actor:** Public User
- **Goal:** To see a list of dogs currently available for adoption or fostering.
- **Description:** The user navigates to the "Dogs" or "Adopt" section of the website and is presented with a gallery or list of dogs.
- **Preconditions:**
  - The user has access to the Dog Rescue website.
- **Main Flow:**
  1. User accesses the website.
  2. User navigates to the page displaying dogs (e.g., "Our Dogs", "Adoptable Dogs").
  3. The system displays a list/gallery of dogs with summary information (e.g., name, breed, age, image).
  4. User can potentially filter or sort the list based on criteria like breed, size, age (if implemented).
- **Postconditions:**
  - The user has viewed the list of available dogs.
- **Alternative Flows:**

  - If no dogs are currently available, a message indicating this is displayed.

UC-002: View Detailed Dog Profile

- **Actor:** Public User
- **Goal:** To get more detailed information about a specific dog.
- **Description:** After viewing a list of dogs, the user selects a specific dog to see its full profile.
- **Preconditions:**
  - The user is viewing the list of available dogs (UC-001).
  - The selected dog exists in the system.
- **Main Flow:**
  1. User selects a dog from the list.
  2. The system displays the detailed profile page for that dog, including name, breed, age, size, gender, image(s), description, tags, and status.
- **Postconditions:**
  - The user has viewed the detailed information for the selected dog.
- **Alternative Flows:**
  - If the selected dog's profile is not found or an error occurs, an appropriate message is displayed.

UC-003: Submit a Rescue Request for a Found Dog

- **Actor:** Public User
- **Goal:** To inform the rescue organization about a dog that has been found and needs assistance.
- **Description:** The user fills out and submits a form with details about the found dog and their contact information.
- **Preconditions:**
  - The user has access to the Dog Rescue website.
- **Main Flow:**
  1. User navigates to the "Rescue Submission" or "Report a Found Dog" form.
  2. User fills in the required fields:
     - Dog's details (if known): name, breed, gender, age, size.
     - Location where the dog was found.
     - Description of the dog and situation.
     - Submitter's contact information: contact name, contact email, contact phone.
     - Optionally, upload or provide links to images of the dog.
  3. User submits the form.
  4. The system validates the input.
  5. If valid, the system saves the submission with a 'pending' status and displays a confirmation message to the user.
- **Postconditions:**
  - A new rescue submission record is created in the system.
  - The user receives confirmation of their submission.
- **Alternative Flows:**
  - If validation fails (e.g., missing required fields, invalid email format), the system displays error messages, and the user can correct the form.
  - If there's a system error during submission, an error message is displayed.

UC-004: Apply to Become a Volunteer

- **Actor:** Public User
- **Goal:** To submit an application to volunteer with the dog rescue organization.
- **Description:** The user fills out and submits a volunteer application form.
- **Preconditions:**
  - The user has access to the Dog Rescue website.
- **Main Flow:**
  1. User navigates to the "Volunteer" or "Get Involved" section and accesses the application form.
  2. User fills in the required fields:
     - Personal Information: name, email, phone.
     - Volunteer Preferences: volunteer type (e.g., Dog Walker, Foster Parent), availability.
     - Experience with animals.
     - Optionally, an additional message.
  3. User submits the form.
  4. The system validates the input.
  5. If valid, the system saves the volunteer application with a 'pending' status and displays a confirmation message to the user.
- **Postconditions:**
  - A new volunteer application record is created in the system.
  - The user receives confirmation of their application.
- **Alternative Flows:**
  - If validation fails, the system displays error messages, and the user can correct the form.
  - If there's a system error during submission, an error message is displayed.

2.2 Administrator (Admin) Use Cases

Admins have access to a secure backend or dashboard to manage the application's data.

UC-101: Log In to Admin Dashboard

- **Actor:** Administrator
- **Goal:** To securely access the administrative functions of the application.
- **Description:** The admin provides their credentials (username and password) to gain access.
- **Preconditions:**
  - The admin has an existing, approved admin account.
  - The admin knows their username and password.
- **Main Flow:**
  1. Admin navigates to the admin login page.
  2. Admin enters their username and password.
  3. Admin submits the login form.
  4. The system validates the credentials against the `Admins` collection.
  5. If credentials are valid, the system grants access to the admin dashboard.
- **Postconditions:**

  - The admin is logged in and has access to administrative features.
- **Alternative Flows:**
  - If credentials are invalid, an error message is displayed.
  - If the account is pending or locked, an appropriate message is displayed.
  - Forgot password (if implemented): Admin can initiate a password reset process.

UC-102: Manage Dog Profiles (Add, Edit, Delete)

- **Actor:** Administrator
- **Goal:** To maintain an up-to-date database of dogs.
- **Description:** Admins can add new dogs, update information for existing dogs, or remove dogs that are no longer with the rescue (e.g., adopted, transferred).
- **Preconditions:**
  - Admin is logged in (UC-101).
- **Main Flow (Add Dog):**
  1. Admin navigates to the "Manage Dogs" section and chooses to "Add New Dog."
  2. Admin fills in the dog's details (name, breed, age, size, gender, image URL, description, tags, status, optional rescueId).
  3. Admin saves the new dog profile.
  4. The system validates and saves the new dog to the `Dogs` collection.
- **Main Flow (Edit Dog):**
  1. Admin navigates to the "Manage Dogs" section and selects a dog to edit.
  2. Admin modifies the desired fields.
  3. Admin saves the changes.
  4. The system validates and updates the dog's record in the `Dogs` collection.
- **Main Flow (Delete Dog):**
  1. Admin navigates to the "Manage Dogs" section and selects a dog to delete.
  2. Admin confirms the deletion.
  3. The system removes the dog's record from the `Dogs` collection (or marks it as inactive/archived).
- **Postconditions:**
  - The `Dogs` collection is updated according to the admin's actions.
- **Alternative Flows:**
  - Validation errors prevent saving incorrect data.
  - Confirmation prompts before deletion to prevent accidental data loss.

UC-103: Manage Rescue Submissions

- **Actor:** Administrator
- **Goal:** To review and process incoming rescue requests.
- **Description:** Admins can view submitted rescue requests, update their status (e.g., 'pending', 'processing', 'rescued', 'closed'), and take appropriate actions.
- **Preconditions:**
  - Admin is logged in (UC-101).
- **Main Flow:**
  1. Admin navigates to the "Rescue Submissions" section.

2. The system displays a list of submissions, filterable by status.
3. Admin selects a submission to view its details.
4. Admin can update the status of the submission (e.g., from 'pending' to 'processing' after contacting the submitter, or to 'rescued' if the dog is taken into care).
5. Admin may add internal notes or assign tasks related to the submission (if supported).
6. Optionally, if a dog from a submission is added to the system, the `rescueId` in the `Dogs` collection can be linked.

- **Postconditions:**
  - The status and details of rescue submissions are updated in the `RescueSubmissions` collection.
- **Alternative Flows:**
  - Admin might contact the submitter directly based on the provided contact information.

UC-104: Manage Volunteer Applications

- **Actor:** Administrator
- **Goal:** To review and process volunteer applications.
- **Description:** Admins can view submitted volunteer applications, update their status (e.g., 'pending', 'approved', 'rejected'), and contact applicants.
- **Preconditions:**
  - Admin is logged in (UC-101).
- **Main Flow:**
  1. Admin navigates to the "Volunteer Applications" section.
  2. The system displays a list of applications, filterable by status.
  3. Admin selects an application to view its details.
  4. Admin can update the status of the application (e.g., from 'pending' to 'approved' or 'rejected').
  5. Admin may add internal notes.
- **Postconditions:**
  - The status and details of volunteer applications are updated in the `Volunteers` collection.
- **Alternative Flows:**
  - Admin might contact the applicant via email or phone for an interview or to provide further information.

(Potentially) UC-105: Manage Admin Accounts

- **Actor:** Administrator (Super Admin, if roles are granular)
- **Goal:** To manage other admin users.
- **Description:** A higher-level admin might be able to add new admins, change their status, or reset passwords. (This depends on the complexity of admin roles implemented).
- **Preconditions:**
  - Admin is logged in (UC-101) and has sufficient privileges.
- **Main Flow:**
  1. Admin navigates to user management section.
  2. Admin can add, edit (e.g., status), or remove other admin accounts.
- **Postconditions:**
  - The `Admins` collection is updated.

This document provides a foundational set of use cases. Additional use cases may be identified as the application evolves.