

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.
- **Registered User:** An authenticated user who can submit adoption applications, access a personal dashboard, and submit rescue reports.
- **Administrator (Admin):** A privileged user responsible for managing the application's content and operations, such as dog profiles, rescue submissions, volunteer applications, and adoption 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" 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, Event Helper, Kennel Assistant, Other), availability (Weekdays, Weekends, Evenings, Mornings, Flexible).
 - 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 Registered User Use Cases

These users have authenticated accounts and can access additional features.

UC-005: User Registration

Actor: Public User

Goal: To create an account to access user-specific features.

Description: The user registers for an account with the Dog Rescue application.

Preconditions:

- The user has access to the Dog Rescue website.

Main Flow:

1. User navigates to the registration page.
2. User fills in required fields: username, password, name, email, phone (optional).
3. User submits the registration form.
4. The system validates the input and checks for duplicate usernames/emails.
5. If valid, the system creates a new user account with 'user' role and 'active' status.
6. User is automatically logged in and receives a confirmation.

Postconditions:

- A new user account is created in the system.
- The user is authenticated and can access user-specific features.

Alternative Flows:

- If validation fails or username/email already exists, appropriate error messages are displayed.
- If there's a system error, an error message is displayed.

UC-006: User Login

Actor: Registered User

Goal: To authenticate and access user-specific features.

Description: The user provides their credentials to access their account.

Preconditions:

- The user has a registered account.

Main Flow:

1. User navigates to the login page.
2. User enters their username and password.
3. User submits the login form.
4. The system validates the credentials.
5. If valid, the system grants access and redirects to the appropriate page.

Postconditions:

- The user is authenticated and has access to user-specific features.

Alternative Flows:

- If credentials are invalid, an error message is displayed.
- If the account is inactive, an appropriate message is displayed.

UC-007: Submit Adoption Application

Actor: Registered User

Goal: To apply for adopting a specific dog.

Description: The authenticated user submits an adoption application for a dog they're interested in.

Preconditions:

- User is logged in (UC-006).
- User is viewing a dog's profile (UC-002).

Main Flow:

1. User navigates to the adoption application form for a specific dog.
2. User fills in optional application notes.
3. User submits the application.
4. The system validates the input and checks for duplicate applications.
5. If valid, the system creates a new application with 'Pending' status.
6. User receives confirmation of their application.

Postconditions:

- A new adoption application record is created in the system.
- The user receives confirmation of their submission.

Alternative Flows:

- If the user has already applied for this dog, an error message is displayed.
- If validation fails, appropriate error messages are displayed.

UC-008: View Personal Dashboard

Actor: Registered User

Goal: To view their personal information, rescue submissions, and adoption applications.

Description: The authenticated user accesses a dashboard showing their personal data and activities.

Preconditions:

- User is logged in (UC-006).

Main Flow:

1. User navigates to their personal dashboard.
2. The system displays:
 - User's personal information.

- List of their rescue submissions.
 - List of their adoption applications with current statuses.
3. User can view details of each submission/application.

Postconditions:

- The user has viewed their personal dashboard information.

Alternative Flows:

- If no submissions or applications exist, appropriate messages are displayed.

UC-009: Withdraw Adoption Application

Actor: Registered User

Goal: To withdraw a previously submitted adoption application.

Description: The user can withdraw their adoption application if it hasn't been processed yet.

Preconditions:

- User is logged in (UC-006).
- User has a pending or under review adoption application.

Main Flow:

1. User navigates to their dashboard or application details.
2. User selects the application to withdraw.
3. User confirms the withdrawal.
4. The system updates the application status to 'Withdrawn'.

Postconditions:

- The adoption application status is updated to 'Withdrawn'.

Alternative Flows:

- If the application has already been processed (approved/rejected), withdrawal is not allowed.

2.3 Administrator (Admin) Use Cases

Admins have access to a secure backend 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 (hidden route: `/admin-login-7a91b523e61`).
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, an appropriate message is displayed.

UC-102: Admin Registration (with Secret Key)

Actor: Administrator

Goal: To create a new admin account using a secret key.

Description: A new admin can register using a secret key for security.

Preconditions:

- The person has the secret key for admin registration.

Main Flow:

1. Admin navigates to the admin login page.
2. Admin switches to registration mode.
3. Admin fills in username, password, and secret key.
4. Admin submits the registration form.
5. The system validates the secret key and creates the admin account.
6. Admin is automatically logged in.

Postconditions:

- A new admin account is created in the system.
- The admin is authenticated and has access to administrative features.

Alternative Flows:

- If the secret key is invalid, an error message is displayed.
- If username already exists, an error message is displayed.

UC-103: 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.

Preconditions:

- Admin is logged in (UC-101).

Main Flow (Add Dog):

1. Admin navigates to the "Dog Management" 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 rescueld).
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 "Dog Management" 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 "Dog Management" section and selects a dog to delete.
2. Admin confirms the deletion.
3. The system removes the dog's record from the Dogs collection.

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-104: Manage Rescue Submissions

Actor: Administrator

Goal: To review and process incoming rescue requests.

Description: Admins can view submitted rescue requests, update their status, 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', 'rescued', or 'closed').
5. Admin may add internal notes or assign tasks related to the submission.
6. Optionally, if a dog from a submission is rescued, the system can automatically create a new dog entry.

Postconditions:

- The status and details of rescue submissions are updated in the RescueSubmissions collection.
- If status is changed to 'rescued', a new dog entry may be created.

Alternative Flows:

- Admin might contact the submitter directly based on the provided contact information.

UC-105: Manage Volunteer Applications

Actor: Administrator

Goal: To review and process volunteer applications.

Description: Admins can view submitted volunteer applications, update their status, and contact applicants.

Preconditions:

- Admin is logged in (UC-101).

Main Flow:

1. Admin navigates to the "Volunteer Management" 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.
6. The system automatically sends email notifications for status changes.

Postconditions:

- The status and details of volunteer applications are updated in the Volunteers collection.
- Email notifications are sent to applicants for status changes.

Alternative Flows:

- Admin might contact the applicant via email or phone for an interview or to provide further information.

UC-106: Manage Adoption Applications

Actor: Administrator

Goal: To review and process adoption applications.

Description: Admins can view all adoption applications, update their status, and manage the adoption process.

Preconditions:

- Admin is logged in (UC-101).

Main Flow:

1. Admin navigates to the "Adoption Applications" section.
2. The system displays a list of all applications, filterable by status.
3. Admin selects an application to view its details.
4. Admin can update the application status (e.g., from 'Pending' to 'Under Review', 'Approved', 'Rejected', or 'Withdrawn').
5. Admin may add internal notes.
6. The system automatically sends email notifications for status changes.

Postconditions:

- The status and details of adoption applications are updated in the Applications collection.
- Email notifications are sent to applicants for status changes.

Alternative Flows:

- Admin might contact the applicant via email or phone for additional information.

UC-107: View Dashboard Statistics

Actor: Administrator

Goal: To view an overview of the rescue organization's activities and metrics.

Description: Admins can access a dashboard showing key statistics and recent activities.

Preconditions:

- Admin is logged in (UC-101).

Main Flow:

1. Admin accesses the main dashboard.
2. The system displays:
 - Total number of dogs.
 - Number of adopted dogs.
 - Total number of volunteers.
 - Pending applications count.
 - Rescue submissions count.
 - Pending rescues count.
3. Admin can view recent activities and quick action buttons.

Postconditions:

- The admin has viewed the dashboard statistics and can take appropriate actions.

Alternative Flows:

- If no data exists, appropriate messages are displayed.

3. System Features and Capabilities

3.1 Authentication and Security

- JWT-based authentication for both users and admins
- Separate authentication flows for different user types
- Protected routes with middleware validation
- Password hashing and secure storage

3.2 Email Notifications

- Automatic email notifications for volunteer application status changes
- Automatic email notifications for adoption application status changes
- Confirmation emails for submissions and applications

3.3 Data Management

- CRUD operations for all major entities (dogs, volunteers, rescue submissions, applications)
- Status tracking and workflow management
- Image handling for dogs and rescue submissions
- Data validation and error handling

3.4 User Experience

- Responsive design with Material-UI components
- Intuitive navigation and user interface
- Form validation and error messaging
- Loading states and user feedback

4. Technical Implementation Notes

4.1 Backend Architecture

- Node.js with Express.js framework
- MongoDB with Mongoose ODM
- JWT middleware for route protection
- RESTful API design

4.2 Frontend Architecture

- React with TypeScript
- Material-UI for component library
- React Router for navigation

- Context API for state management

4.3 Database Models

- **Admin:** username, password, name, email, role, status
- **User:** username, password, name, email, phone, role, status
- **Dog:** name, breed, age, size, gender, image, description, tags, status, rescued
- **Volunteer:** name, email, phone, volunteerType, availability, experience, message, status
- **RescueSubmission:** name, breed, gender, age, size, location, description, contact info, status
- **Application:** user, dog, status, notes, timestamps

5. Future Enhancements

5.1 Planned Features

- Enhanced filtering and search capabilities for dogs
- Advanced reporting and analytics
- Mobile application
- Integration with external services (veterinary, microchipping)

5.2 Security Improvements

- Role-based access control (RBAC)
- Audit logging
- Two-factor authentication
- API rate limiting

This document provides a comprehensive overview of the current use cases and system capabilities. Additional use cases may be identified as the application evolves and new features are implemented.