**ONLINE PET ADOPTION SYSTEM**

**By:**

**Chavas, Mary Shelireen D.**

**Requirement Specification Document**

**INTRODUCTION**

**Purpose of the Document**

The purpose of this document is to define the requirements for the Online Pet Adoption System. It serves as a reference for developers, designers, testers, and stakeholders to ensure a clear and shared understanding of the system’s functional and non-functional requirements.

**Overview of the Software System Being Developed**

The Online Pet Adoption System is a web-based platform designed to facilitate the adoption of pets from animal shelters and rescue organizations. The system enables users to browse available pets, view detailed profiles, and submit adoption applications online. Administrators and shelter staff can manage pet listings and review applications. The system aims to streamline the adoption process, increase pet visibility, and promote responsible pet ownership.

**Scope of the Requirements Specification**

This requirements specification document covers:

Functional requirements (features and behaviors the system must support)

Non-functional requirements (performance, security, usability.)

User roles and permissions

System constraints and assumptions

Acceptance criteria for successful implementation

**FUNCTIONAL REQUIREMENTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement ID | Requirement Description | Priority | Dependencies | Acceptance Criteria |
| FR-01 | Users must be able to register for an account using an account. | High | None | User can create an account and receive a confirmation message. |
| FR-02 | Users must be able to log in and log out securely. | High | FR-01 | User can log in with valid credentials and is redirected to the dashboard; logout end the session. |
| FR-03 | User can browse a list of available pets with basic information (name, species, etc.). | High | None | Pet list is displayed with at least name, species, and status for each pet. |
| FR-04 | Users can view detailed profiles for each pet. | high | FR-03 | Clicking a pet shows a detail page with full information and images. |
| FR-05 | Users can submit adoption applications for available pets | high | FR-01, FR-04 | User can fill and submit an application form; confirmation is shown. |
| FR-06 | Admins can add, edit or remove pet listings. | high | Admin log in (FR-02) | Admin dashboard allows CRUD operations on pet records. |
| FR-07 | Admins can review and update the status of adoption applications. | High | FR-05, Admin, log in | Admin can view applications and change their status (pending approved rejected.) |
| FR-08 | Password reset functionality for users who forget their password. | Medium | FR-01 | User can request a password reset and set a new password via email link. |
| FR-09 | Search and filter pets by species, breed, or status. | Medium | FR-03 | User can filter/search pet listings and see relevant results. |
| FR-10 | Responsive design for usability on desktop devices. | Medium | All UI features | System display correctly and is usable on various screen sizes. |

**NON-FUNCTIONAL REQUIREMENTS**

**Performance**

The system shall respond to user actions (page loads, form submissions) within 2 seconds under normal operating conditions.

The system shall support at least 100 concurrent users without significant degradation in performance.

Database queries shall be optimized to return results within 1 second for standard operations.

**Usability**

The user interface shall be intuitive and easy to navigate for users of all technical backgrounds.

The system shall provide clear feedback for user actions (successful submissions, errors).

The application shall be responsive and usable on both desktop and mobile devices.

Accessibility standards (WCAG 2.1) shall be considered to support users with disabilities.

**Reliability**

The system shall be available 99% of the time, excluding scheduled maintenance.

The system shall recover gracefully from unexpected failures, with minimal data loss.

Regular backups of the database shall be performed to prevent data loss.

**Security**

All user passwords shall be stored as salted hashes.

The system shall enforce authentication for all sensitive operations.

User roles and permissions shall be strictly enforced to prevent unauthorized access.

All data transmitted between client and server shall be encrypted using HTTPS.

The system shall be protected against common web vulnerabilities (SQL injection, XSS, CSRF).

**Scalability**

The system architecture shall allow for horizontal scaling to accommodate increased user load.

The database design shall support efficient growth in the number of users, pets, and applications.

Static assets shall be served via a CDN to handle high traffic volumes.

**Maintainability**

The codebase shall be modular and well-documented to facilitate future enhancements.

Configuration files and environment variables shall be used for environment-specific settings.

Automated tests shall be implemented to ensure system stability during updates.

The system shall support easy deployment and rollback procedures.

**USE CASES**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | Use Case Name | Description | Actors | Preconditions | Postconditions | Alternate Flows/Exceptions |
| UC-01 | User Registration | A new user creates an account by providing required information. | User | User is not logged in. | User account is created and user is logged in or prompted to log in. | Email already in use, valid input registration fails |
| UC-02 | User login | A registered user logs in with valid credentials | user | User has a registered account | User authenticated and redirected to dashboard | Invalid credentials account locked log in fails |
| UC-03 | Browse pet | User views a list of available pets for adoption | User-Guest | None | List of pets displayed | No pets available database error |
| UC-04 | View pet details | User select a pet to view detailed information and image | User and guest | At least one pet exists | Pet detailed page is displayed | Pet not found. Image missing |
| UC-05 | Submit adoption application | User submits an application for a specific pet | User | User is logged in pet is available | Application is submitted and confirmation is shown | Pet no longer available validation error |
| UC-06 | Admin adds edits or removes pet listings | Admin adds edits or removes pet listing | admin | Admin is logged in | Pet listing is updated in the system | Invalid data database error |
| UC-07 | Admin review applications | Admin reviews and updates the status of adoption applications | admin | Admin is logged in applications exist | Application status is updated | Application not found invalid status |
| UC-08 | Password reset | User requests a password reset and sets a new password via email link | user | User has a registered email | Password is updated user can log in | Email not found link expired. |

**DATA REQUIREMENTS**

**Data Entities and Attributes**

|  |  |
| --- | --- |
| Entity | Attributes |
| User | User\_id (PK), username, password\_hash, email, role (adopted/admin/shelter) created\_At |
| Pet | Pet\_id (PK) Name, species, age, gender, description, status, created\_At |
| Adoption Application | Application\_id (PK) user\_id (FK) pet\_id (FK) Status, application\_date, notes |
| Pet Image (optional) | Image\_id (PK), pet\_id (FK) image\_url |

**Relationships between data entities**

* User to Adoption Application

One to many (A user can submit multiple adoption applications.)

Pet to AdoptionApplication:

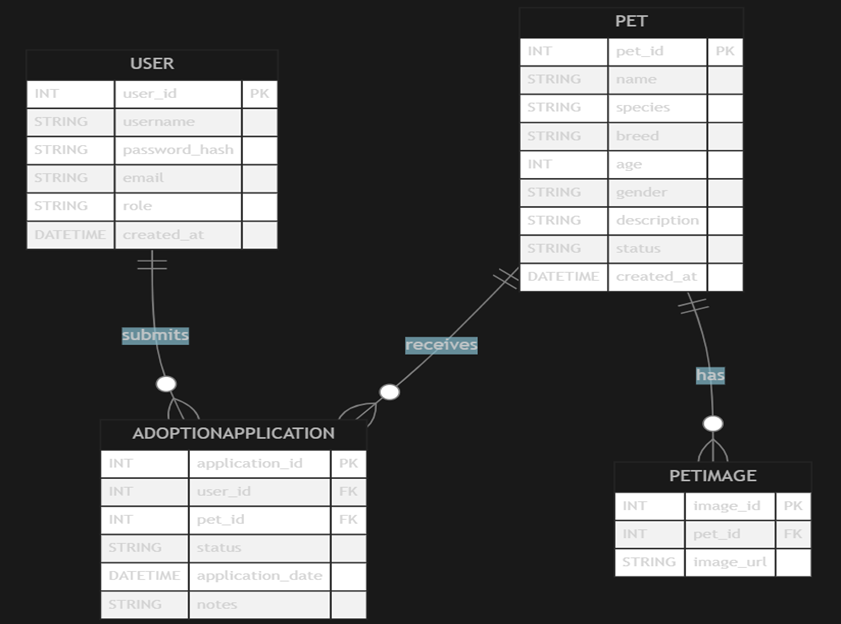
One to many (A pet can have multiple adoption applications.)

* Pet to Pet Image

One to many (a pet can have multiple images)

* AdoptionApplication links user to pet:

**MANY TO ONE FOR BOTH USER AND PET**



**ASSUMPTIONS AND CONSTRAINTS**

**Assumptions**

Users have access to a modern web browser and a stable internet connection.

Administrators and shelter staff are trained to use the system’s admin features.

All pet and user data entered into the system is accurate and up-to-date.

Email services (for registration, password reset.) are available and reliable.

The system will be deployed in a secure, managed hosting environment.

The number of concurrent users and pets will be within the expected capacity defined in the non-functional requirements.

Third-party services (email, image hosting) used by the system will remain available and maintain their APIs.

**Constraints**

The system must comply with relevant data protection and privacy regulations (GDPR, local laws).

Only the specified user roles (adopter, admin, shelter staff) are supported.

The system does not include real-time communication or chat features.

The database technology is limited to those supported by the deployment environment (SQLite for development, PostgreSQL/MySQL for production).

The system must use HTTPS for all client-server communications.

The user interface must be accessible and responsive, but advanced accessibility features (screen reader optimization) are not required in the initial release.

Integration with external systems is limited to those specified in the requirements (email, image hosting).

**Glossary**

|  |  |
| --- | --- |
| Term | Definition |
| Adopter | A user who browses pets and submits adoption applications |
| Admin | A user with privileges to manage pet listings and review applications |
| Pet | An Animal available for adoption in the system |
| Adoption Application | A form submitted by an adopter to request adoption of a pet |
| User Account | A registered profile required to access system features. |
| Authentication | The process of verifying a user’s identity. |
| Authorization | The process of granting access to features based on user roles. |