

Habib University
Software Engineering - CS/CE 353/374

Project - Report
PAWSTrack



Instructor: Abdul Rahman Qaim

Lyeba Abid - la07309
Iqra Ahmed - ia07674
Ali Muhammad Asad - aa07190
Sadiqah Mushtaq sm07152

Contents

1	Project Aims, Objectives and Abstract	3
2	User Specifications	4
3	System Specifications	5
4	Functional and Non-Functional Requirements	7
4.1	Functional Requirements	7
4.2	Non-Functional Requirements	7
5	Mockups and Wireframes	8
6	UML Diagrams	10
6.1	Use Cases	10
6.2	Sequence Diagram	11
6.3	Class Diagram	12
6.4	ERD	13
6.5	Architecture	14
7	Requirement Traceability Matrix	15
8	DevOps User Stories, and Acceptance Criteria	16
8.1	Admin Stories	16
8.2	User Stories	19
9	Test Plan	22

1 Project Aims, Objectives and Abstract

PAWSTrack is a web-based shelter pet and record management system developed for the local shelter "PAWS" that is run by our own RA, Ahmed Bilal. Currently, the shelter does not have any proper management system, running and working purely on the efforts of some individuals who are managing the shelter. The shelter is in need of a system that can help them manage their pet records, inventory, staff, and vet records, and their finances / donations. The general public also has no formal way of contacting the shelter, or understanding their operations apart from their social media pages, which are also not comprehensive or efficient in providing information to the public.

PAWSTrack aims to streamline shelter operations for the shelter admin by providing efficient record keeping, enhancing workflow, and ensuring data security. The system will allow the admin to manage pet records including their adoption status, medical history and records, shelter inventory and stock, staff and vet's working at the shelter. The general public can also register on the web-page to understand more about the shelter operations, view current pets at the shelter either undergoing some medical treatment or if they are up for adoption. The users can also register their pets for treatments, or donate to the shelter out of good will. The app aims to create awareness amongst the general public while providing them with a platform to interact with the shelter.

Key features include a user-friendly interface, scalability, and robust privacy measures. PAWSTrack serves as an intuitive solution for managing pet records, veterinary services, and inventory, contributing to the overall well-being of the shelter and its furry residents.

2 User Specifications

We will mainly have two types of users; **Admin** and **Client**

1. Admin Users:

These users include the shelter staff, vets, manager / admin or any other person in charge. As the Admin, I should be able to:

- upload, and manage (update and delete) pet records, including details about each pet
- update adoption status of pets
- manage and view inventory stocks
- upload and manage information about vets / staff etc
- upload and manage pets currently being treated, along with their medical bills and expenditure
- upload and manage donations / sponsorships for certain items / pets
- manage donations received for the shelter
- make day to day updates about any specific user's pets currently being treated

2. Client Users:

As a user I should be able to:

- create a user account
- login to my user account
- manage my own profile including personal information, profile picture
- view information about the animals and pets currently at the shelter
- view information about the general pet care, environment and information about the shelter
- view pets or animals up for adoption, and their history if there is any
- access customer support channels for inquiries, or assistance (dedicated help center)
- make requests for pet adoptions (through dedicated help center right now)
- request for treatment for a pet, or if a stray animal / pet needs to be left at the shelter (through dedicated help center right now)

3 System Specifications

1. Admin Management

(a) *Admin Registration* (only done by another admin)

- i. Username
- ii. Email
- iii. Password

(b) *Admin Login*

- i. Username
- ii. Password

(c) *Account Management*

- i. Add / Edit profile picture
- ii. Change password
- iii. Change email

2. User Account Management

(a) *User Registration*

- i. Username
- ii. Email
- iii. Password

(b) *User Login*

- i. Username
- ii. Password

(c) *Account Management*

- i. Add / Edit profile picture
- ii. Change password
- iii. Change email

3. Dashboard

(a) *Admin Dashboard*

- i. View and manage pet records
- ii. View and manage user messages
- iii. add or remove staff

(b) *User Dashboard*

- i. View and edit personal details

- ii. View animals and pets at the shelter
- iii. View shelter information (about us page) to understand shelter operations
- iv. Contact customer support by leaving a message through the “Contact Us” page

4. **Pet Management (Admin Only)**

- (a) *Add new pet profiles*
 - i. Picture
 - ii. Name
 - iii. Age
 - iv. Type
 - v. Adoption status
 - vi. Adoption history
 - vii. Medical history
 - viii. Treatment status
 - ix. Treatment history
 - x. Treatment cost
 - xi. Living status (dead or alive)
- (b) Update / edit existing pet profiles
- (c) Delete pet profiles

5. **Customer Support Management (Users)**

- (a) *User can contact shelter by leaving a message*
 - i. Name
 - ii. Email
 - iii. Phone
 - iv. Subject
 - v. Message details / query / concern / request
- (b) User can contact shelter through telephone
- (c) User can visit the shelter by following the location given on the attached map

6. **Customer Support Management (Admin)**

- (a) Admin can view messages left by users
- (b) *Admin can respond to messages*
 - i. Email
 - ii. Phone

4 Functional and Non-Functional Requirements


4.1 Functional Requirements

- Admin Management - can manage accounts
- User Account Management
- Dashboard - can view relevant details
- Pet Management by adding, updating, or deleting pet details
- Customer Support Management
 - User can contact the shelter
 - Admin can respond to user queries and messages

4.2 Non-Functional Requirements

- Web Application only
- Performance: system should be able to handle multiple users at the same time without any significant performance degradation
- Every request should not take more than 2 seconds to complete
- Security: Ensure that users data is secure, and that unauthorized users cannot access admin only details and operations
- Usability: System should be user friendly, and easy to navigate by both the admin and the clients
- Compatibility: System should be compatible for various kinds of devices - in our case, various browsers
- Maintainability: more features can be added without affecting existing ones
- Reliability: system should not be down for long periods of time

5 Mockups and Wireframes



PAWTRACK
ANALYSIS & MONITORING

[Sign up with Google](#)

- OR -

Full Name

Email Address

Password

[Sign Up](#)

Already have an account? [Login](#)

(a) Sign-up / Register Screen



PAWTRACK
ANALYSIS & MONITORING

Frame 10
[Home](#) [Features](#) [Contact Us](#)Frame
[Login](#) [Sign Up](#)[Sign up with Google](#)

- OR -

Full Name

Email Address

Password

[Sign Up](#)

Already have an account? [Login](#)



PAWTRACK
ANALYSIS & MONITORING

[Settings](#) [Pets](#) [About Us](#) [Contact Us](#)

Account Settings



Full Name

Email Address

Phone

Profile pic

[Choose file](#)

© 2024 Pawstrack. All rights reserved.



PAWTRACK
ANALYSIS & MONITORING

[Settings](#) [Pets](#) [About Us](#) [Contact Us](#)

PAWTRACK
ANALYSIS & MONITORING

Our Mission

At Pawstrack, our mission is to provide innovative solutions for pet owners to track and monitor their beloved companions, ensuring their safety and well-being.

Our Team

We are a team of passionate individuals dedicated to creating cutting-edge technology for pet tracking. Our team includes experienced engineers, designers, and animal lovers who are committed to making a difference in the lives of pets and their owners.

Our Values

- Commitment to pet safety and happiness
- Innovation and continuous improvement
- Transparency and integrity in all our operations
- Customer satisfaction and support

© 2024 Pawstrack. All rights reserved.



PAWTRACK
ANALYSIS & MONITORING

[Settings](#) [Pets](#) [About Us](#) [Contact Us](#)

Get in Touch

If you'd like further information about PAWSTrack, or have any specific queries, please feel free to reach out to us!

Simply fill out the form, include your message, and we'll get back to you as soon as we can.

Name

Email

Subject

Phone

Message

© 2024 Pawstrack. All rights reserved.



PAWTRACK
ANALYSIS & MONITORING

[Settings](#) [Pets](#) [About Us](#) [Contact Us](#)

Pets

Picture	Name	Animal Type	Description	Adopted	Up for Adoption	Medical History	Deceased

© 2024 Pawstrack. All rights reserved.

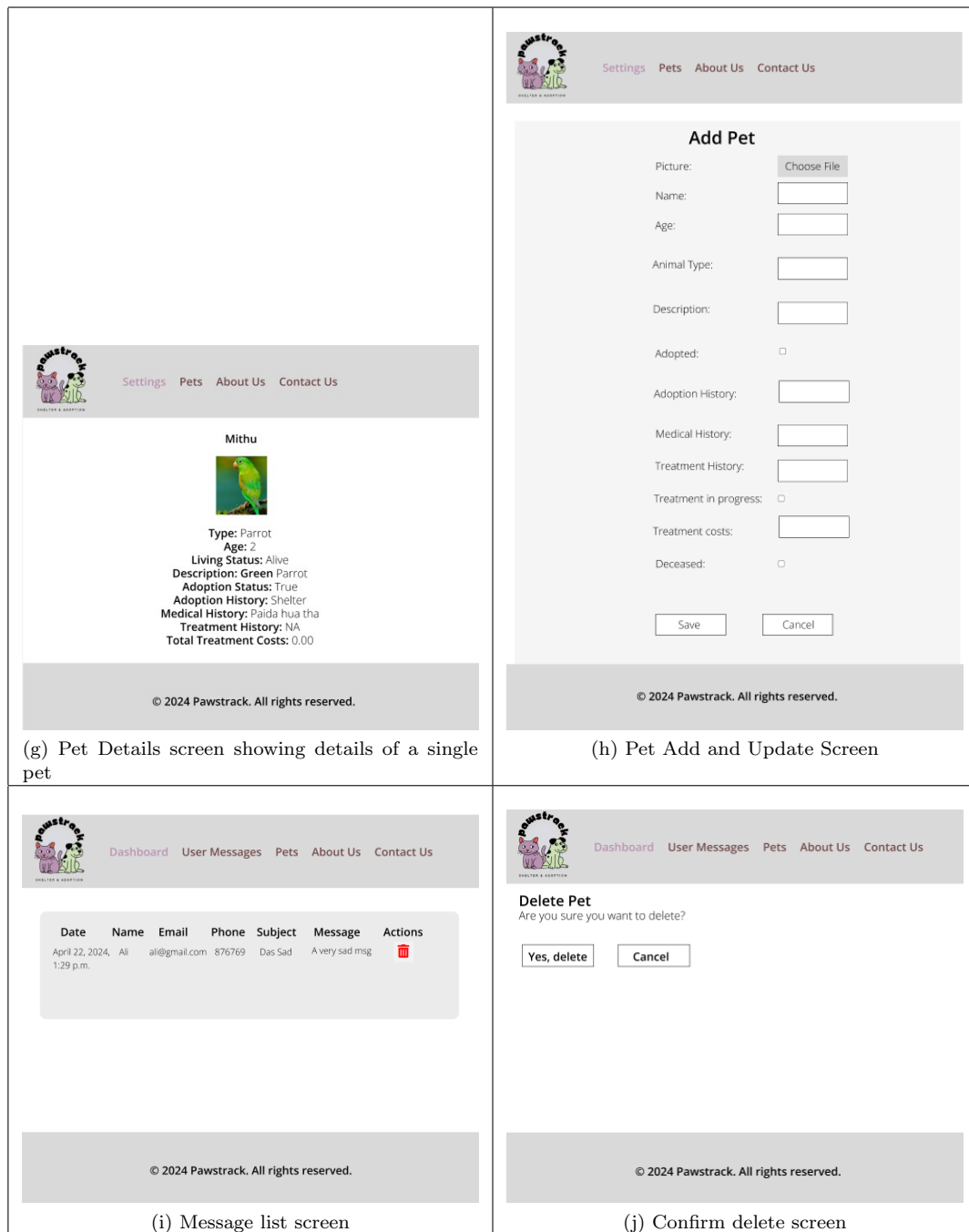


Figure 1: Mockups and Wireframes of the PAWSTrack Web Application

The above mockups and wireframes basically show the different screens of the web application. Some screens like the delete screen were reused, the add screen was used to update as well, and some screens are admin only.

6 UML Diagrams

This section will cover the UML diagrams for the system. The UML diagrams are a visual representation of the system's architecture, design, and functionality, and help in understanding the system's structure and behavior.

6.1 Use Cases



Figure 2: Use Case Diagram showing the different use cases of the system.

The above Use Case diagram effectively incorporates the different use cases of our system and the actors involved in each use case. The actors are the admin and the user/customer. The admin add new pets, view existing pets which extends to editing their details or medical records which further extends to editing the medical records as we need to be able to manage and update pet records. The admin can also delete a pet. Admin can also check messages, which they can choose to delete.

The user can view and update their own account, view pets, view existing pets and their details, and contact the shelter. The user can also view the shelter information.

6.2 Sequence Diagram

Based on the use cases, and provided system specifications, and functional requirements, we have created the following sequence diagram which encapsulates the requests / responses made from the system and to the database to fetch required information based on made requests.

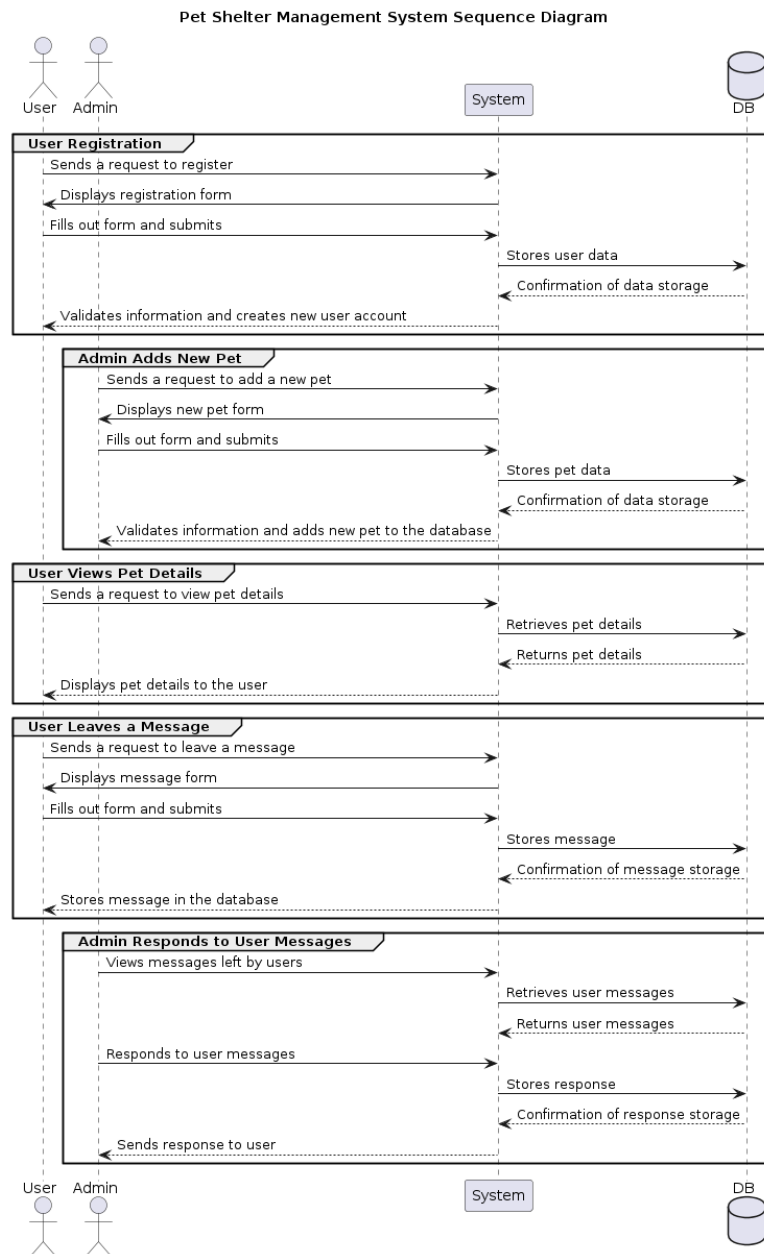


Figure 3: Sequence Diagram showing user/admin-system interaction

6.3 Class Diagram

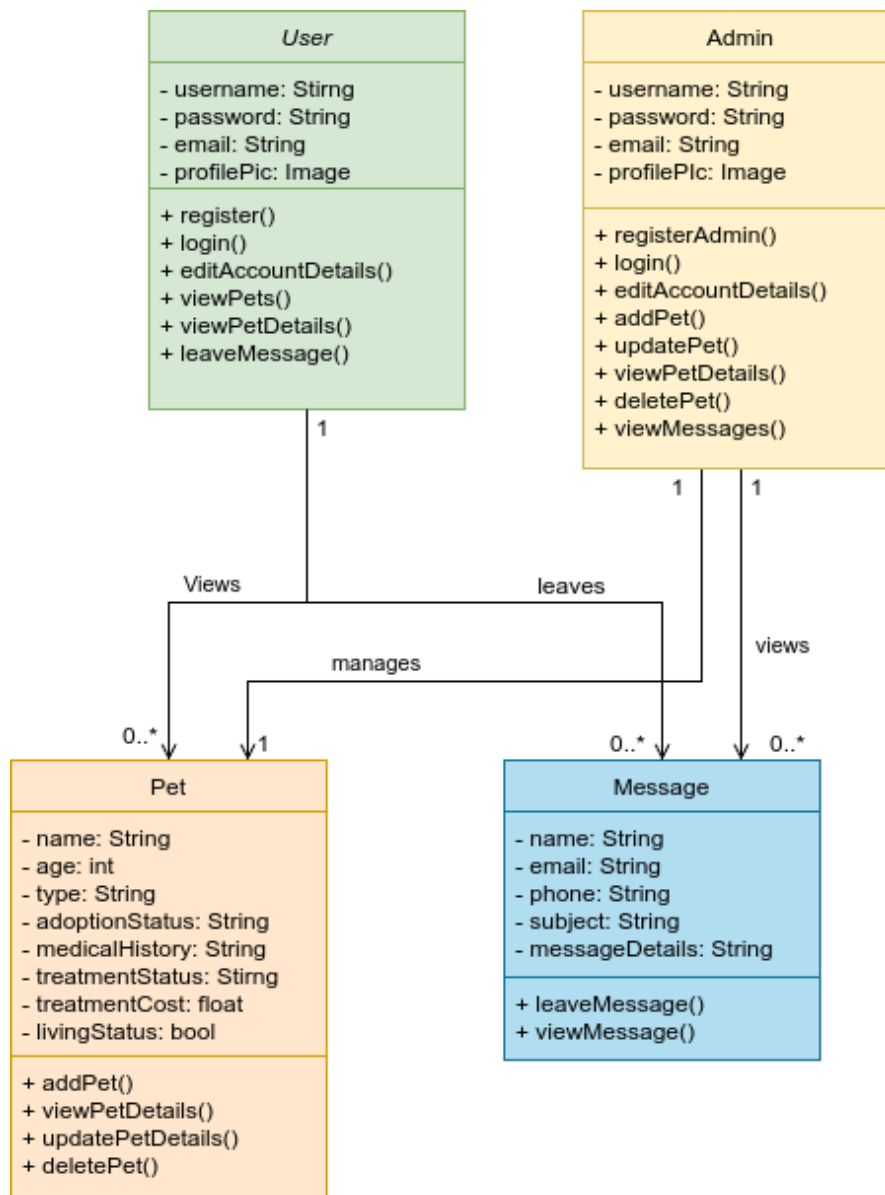


Figure 4: Class Diagram showing the different classes and their relationships

A user can view multiple pets or a pet can be viewed by multiple users. A user can also leave multiple messages, but a message can only be left by one user. An admin can manage multiple pets, and can also respond to multiple messages.

6.4 ERD

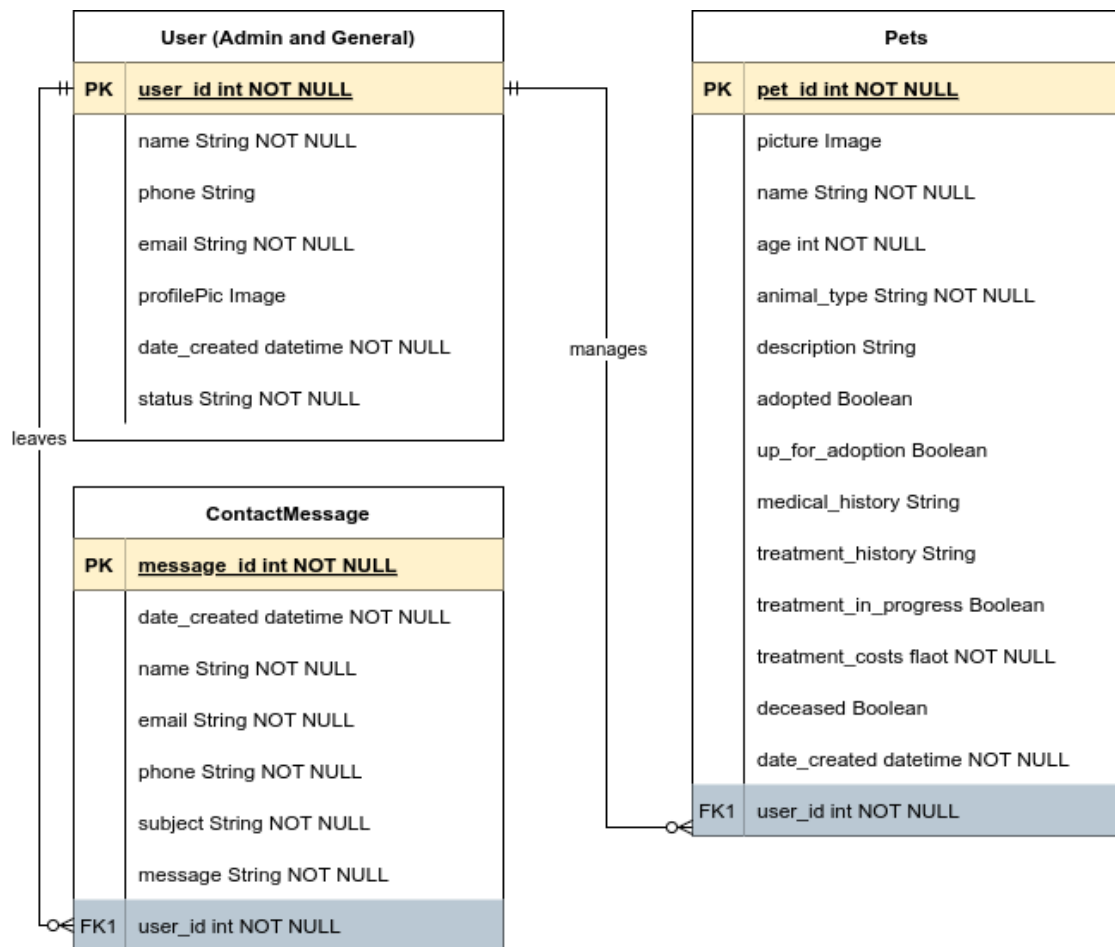


Figure 5: Entity Relationship Diagram showing the different entities and their relationships

In the above ERD diagram, we represent both the general user and admin by one table since they mostly contain the same information, apart from status which basically tells if the user is an admin or not. Each user has their own unique id. Each user can leave multiple messages, however, one message belongs to only one user, hence we define a one-to-many relationship between the user and message table, with the `user_id` being the foreign key in the message table. Similarly, each pet can be viewed by multiple users, however, there is no relation defined for this since it is not necessary for the system. However, one admin can manage multiple pets, hence we define a one-to-many relationship between the admin and pet table, with the `user_id` being the foreign key in the pet table. Each pet also has a unique `pet_id` associated with it, which is used to identify each pet and display that pet's information.

6.5 Architecture

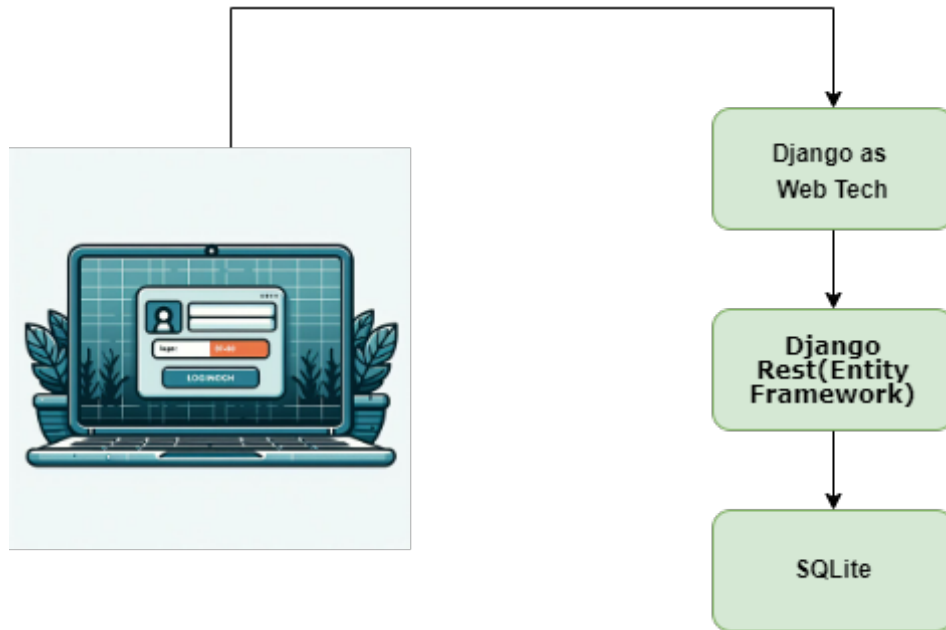


Figure 6: Architecture Diagram showing the different components of the system

Our architectural diagram is defined as shown above. We have made a web application using the python's Django as our main web technology. With Django, we've used basic HTML, CSS, and JavaScript for implementing various functionalities for the web-pages. For the API calls and handling, we've used Django Rest Framework which is a powerful and flexible toolkit for building Web APIs, efficiently handling the requests and responses. Our data is being stored in a SQLite database, a lightweight file based database which is easy to use and manage.

7 Requirement Traceability Matrix

S.No	Module Name	Applicable Roles	Description
1	Admin Management	Admin	Admin can register Admin can login Admin can manage account
2	User Account Management	User	User can register User can login User can manage account
3	Dashboard	Admin	Admin can view and manage pet records Admin can view and manage user messages Admin can add or remove staff
		User	User can view and edit personal details User can view animals and pets at the shelter User can view shelter information User can contact customer support
4	Pet Management	Admin	Admin can add new pet profiles Admin can update or edit existing pet profiles Admin can delete pet profiles
5	Customer Support Management	Admin	Admin can view messages left by users Admin can respond to messages
		User	User can contact shelter by leaving a message User can contact shelter through telephone

8 DevOps User Stories, and Acceptance Criteria

8.1 Admin Stories

1. As an Admin, I can make daily updates on pets

Description: As an Admin, I want to make day-to-day updates about pets currently being held at the shelter. This can include their daily pictures, wallpapers, their treatment if any, their livelihood, their adoption status, new animals, etc.

Acceptance Criteria:

1. Pet Updates:

- Admin can upload one picture per pet profile daily.
- Mandatory fields (name, species, breed, age, gender, ID) are accurately populated and give errors if wrong data types are used for specific fields.
- Treatment information and adoption status are correctly displayed for each pet.

2. Livelihood Updates:

- Admin can post updates on pets' general well-being.
- Updates include behavior, interactions, and special needs.

3. Adoption Status Updates:

- Admin can update adoption status (available, pending, adopted) for each pet.
- Users can view real-time adoption status changes on pet profiles.

4. New Arrivals Updates:

- Admin promptly adds profiles for new animals.
- Correctly captures species, breed, age, and gender for each new arrival.
- Includes at least one picture for each new arrival.

2. As an Admin, I can manage detailed pet records

Description: As an Admin, I want the ability to upload and manage detailed records for each pet. This includes uploading images, maintaining a comprehensive medical history, and updating the adoption status. Efficient pet record management ensures accurate information about each animal in the shelter.

Acceptance Criteria:

1. There should be an option to add a new pet.
2. No record should be created without a pet name.
3. No record should be created without specifying the pet type.
4. No record should be created without providing pet description, history, and setting an adoption status.
5. It should be possible to create a record without uploading a pet image.

6. The submit button should successfully create a pet record when clicked.
7. Pet records should be editable.
8. Pet records should be deletable.

3. As an Admin, I can manage pet treatments

Description: As an Admin, I want to upload and manage information about pets currently undergoing treatment. This includes tracking medical bills and expenditures associated with the treatment. Efficient management of treatment records is essential for providing optimal care to the animals.

Acceptance Criteria:

1. Add a pet to the treatment page.
2. Add the vet treating the pet to the treatment page.
3. Add a description about the treatment.
4. Add medication for the treatment.
5. Add day-to-day cost for the treatment.
6. If the cost is covered, the total cost should be updated accordingly.

4. As an Admin, I can manage adoption status and history

Description: As an Admin, I want the capability to update the adoption status of pets and track adoption applications efficiently. This functionality is crucial for managing the adoption process, ensuring transparency, and matching pets with suitable adopters.

Acceptance Criteria:

1. A pet can be added without any adoption history.
2. It should be possible to set adoption status for any pet successfully.
3. Adoption history can be updated successfully.
4. Adoption status and history should be successfully displayed.

5. As an Admin, I can manage inventory stocks

Description: As an Admin, I want to manage and view inventory stocks easily. This involves the ability to add, update, or delete stock items as needed. Efficient inventory management ensures the availability of necessary supplies for the shelter's operations.

Acceptance Criteria:

1. Add inventory items.
2. Add inventory item details.
3. Add inventory item quantity.
4. Add inventory item price.
5. Add inventory item shelf-life if needed.

6. Update inventory item quantity/price as needed.
7. Delete an inventory item.

6. As an Admin, I can maintain vets and staff information

Description: As an Admin, I want to upload and manage information about vets and staff members, including their roles and responsibilities within the shelter. This functionality ensures that the team's details are organized and accessible for efficient collaboration.

Acceptance Criteria:

1. Admin should be allowed to update the vet's information.
2. A new vet cannot be left nameless, or without a description or contact information.
3. Information about the vet should be displayed correctly.

7. As an Admin, I can manage pet treatments

Description: As an Admin, I want to upload and manage information about pets currently undergoing treatment. This includes tracking medical bills and expenditures associated with the treatment. Efficient management of treatment records is essential for providing optimal care to the animals.

Acceptance Criteria:

1. Add a pet to the treatment page.
2. Add the vet treating the pet.
3. Add a description about the treatment.
4. Add medication for the treatment.
5. Add day-to-day cost for the treatment.
6. If the cost is covered, the total cost should be updated accordingly.

8. As an Admin, I can login to dashboard securely:

Description: As an Admin, I want to securely log in to the system to access the admin dashboard and perform various administrative tasks. This includes ensuring that only authorized personnel can access sensitive information and functionalities.

Acceptance Criteria:

1. Admin ID should be unique.
2. The system should deny access if the credentials are incorrect.
3. The system should allow access if the credentials are correct.
4. Try to login with only username, and only password.

9. As an Admin, I can manage donations and sponsorships

Description: As an Admin, I want to handle and manage donations and sponsorships for items or specific pets. This involves keeping track of received donations and ensuring transparency in financial transactions related to the shelter's operations.

Acceptance Criteria:

1. A donor or sponsor should be able to contact via customer support.
2. A donation or sponsorship should be logged into the system and linked to the pet or inventory if needed.
3. Donor details should be successfully displayed.
4. Sponsor details should be successfully displayed.
5. Donation amount/sponsorship amount should be updated.

8.2 User Stories

1. As a User, I can view adoption candidates

Description: As a General User, I want to view pets or animals available for adoption and their history, if available, to make informed adoption decisions.

Acceptance Criteria:

- System should successfully display list of pets available for adoption.
- On selecting to view a pet, the adoption status should display if the pet is available or not.
- The adoption history and relevant details about the pet should be displayed.

2. As a User, I can access customer support

Description: As a User with an Account, I want to access customer support channels for inquiries or assistance, such as a dedicated help center or information section, so that I can seek help or a clarification whenever needed.

Acceptance Criteria:

- Provide customer service to customers.
- Provide contact information about the team/shelter.
- Working email channel.
- Allow message box on the page.
- User notified when their query is sent.

3. As a User, I can understand Shelter operations

Description: As a General User, I want to view information about general pet care, the shelter environment, and relevant information to better understand the shelter's operations.

Acceptance Criteria:

- Provide customer service to customers.

- Provide contact information about the team/shelter.
- Working email channel.
- Allow message box on the page.
- User notified when their query is sent.

4. As a User, I can login to my account and manage it

Description: As a user with an account, I want to be able to manage my account details, such as name, password, email, contact number, and track my pets etc.

Acceptance Criteria:

- Authenticate the user's credentials.
- Try logging in with only username, and with only password.
- Forget password button.
- Try logging in with incorrect credentials.
- Update username.
- Update password.
- Update account picture.
- Update further details.
- Update pets.
- View and track pets.

5. As a User, I can create my account

Description: As a User with an Account, I want to create my account, including personal information such as email, username, password, so that I can personalize my experience on the platform.

Acceptance Criteria:

- UserID must be unique.
- Authenticate user credentials.
- Try registering with duplicate credentials.
- Try registering with only email, or username, or password.
- Try registering with invalid password.
- Allow access if account created.

6. As a User, I can register and track my pet

Description: As a User with an Account, I want to register my own pet to the shelter for treatment, track their day-to-day status, and make payments for any associated expenditures, so that I can ensure proper care for my pet.

Acceptance Criteria:

- Allow user to register a pet.
- Pet name.
- Pet type.
- Pet description.
- Pet image (optional).
- Show pet's status.
- Show pet's medical bills, and history.
- Allow payment option.

7. As a User, I can leave my feedback

Description: As a User with an Account, I want to leave feedback, reviews, comments, and ratings for the shelter's services and my overall experience, so that I can contribute to improving the shelter's services and share my experience with others.

Acceptance Criteria:

- Allow the user to leave feedback.
- Allow user to give a review and ratings.
- Reviews should be displayed correctly.

8. As a User, I can make adoption requests

Description: As a General User, I want to make requests for pet adoptions easily through the system, so that I can initiate the adoption process conveniently.

Acceptance Criteria:

- Display pets available for adoption.
- Request for adoption form/button available for pet.
- Notify the user if their adoption request has been sent forward.
- Notify user when their request is processed.

9. As a User, I can make donations

Description: As a User with an Account, I want to make one-time or recurring donations, either general or specific, through a secure payment processing system, so that I can support the shelter financially and contribute to its cause.

Acceptance Criteria:

- Allow a page where donations can be made via bank or card etc. OTP should be sent to provided contact info.
- Receipt should be generated to
- show payment processed successfully.
- Receipt should not be generated, an error thrown when not processed.

9 Test Plan

1. Admin Registration and Login:

- **Test Case 1:** Test admin registration with valid credentials.
 - **Steps:**
 - (a) Navigate to the registration page (register).
 - (b) Fill out the registration form with valid admin credentials.
 - (c) Submit the form.
 - **Expected Result:** Admin account should be created successfully, and the user should be redirected to the login page (login).
- **Test Case 2:** Test admin login with registered credentials.
 - **Steps:**
 - (a) Navigate to the login page (login).
 - (b) Enter the registered admin username and password.
 - (c) Submit the login form.
 - **Expected Result:** Admin should be logged in successfully and redirected to the admin home page (home).
- **Test Case 3:** Test admin login with invalid credentials (incorrect username/password).
 - **Steps:**
 - (a) Navigate to the login page (login).
 - (b) Enter incorrect admin username and/or password.
 - (c) Submit the login form.
 - **Expected Result:** Admin should not be able to log in, and an appropriate error message should be displayed.

2. Admin Home Page:

- **Test Case 4:** Test whether admin users are redirected to the correct home page after logging in.
 - **Steps:**
 - (a) Log in as an admin user.
 - (b) Verify the URL or page content to ensure redirection to the admin home page (home).
 - **Expected Result:** Admin user should be redirected to the admin home page after successful login.

3. Viewing and Managing Pets:

- **Test Case 5:** Test if admin users can view a list of pets (pet.list).

- **Steps:**
 - (a) Log in as an admin user.
 - (b) Navigate to the pets list page (pet_list).
- **Expected Result:** Admin should be able to view a list of pets.
- **Test Case 6:** Test if admin users can view details of a specific pet (pet_detail).
 - **Steps:**
 - (a) Log in as an admin user.
 - (b) Navigate to the details page of a specific pet (pet_detail).
 - **Expected Result:** Admin should be able to view detailed information about the selected pet.
- **Test Case 7:** Test adding a new pet (pet_create).
 - **Steps:**
 - (a) Log in as an admin user.
 - (b) Navigate to the page for adding a new pet (pet_create).
 - (c) Fill out the pet creation form with valid information.
 - (d) Submit the form.
 - **Expected Result:** A new pet should be added to the database, and the admin should be redirected to the pet list page (pet_list).
- **Test Case 8:** Test updating an existing pet's information (pet_update).
 - **Steps:**
 - (a) Log in as an admin user.
 - (b) Navigate to the edit page of an existing pet (pet_update).
 - (c) Modify the pet's information.
 - (d) Submit the form.
 - **Expected Result:** The pet's information should be updated in the database, and the admin should be redirected to the pet list page (pet_list).
- **Test Case 9:** Test deleting a pet record (pet_delete).
 - **Steps:**
 - (a) Log in as an admin user.
 - (b) Navigate to the delete page of an existing pet (pet_delete).
 - (c) Confirm the deletion.
 - **Expected Result:** The selected pet record should be deleted from the database, and the admin should be redirected to the pet list page (pet_list).

4. Viewing and Managing Messages:

- **Test Case 10:** Test if admin users can view a list of messages (messages).
 - **Steps:**
 - (a) Log in as an admin user.
 - (b) Navigate to the messages list page (messages).
 - **Expected Result:** Admin should be able to view a list of messages sent by customers.
- **Test Case 11:** Test if admin users can view details of a specific message (message_detail).
 - **Steps:**
 - (a) Log in as an admin user.
 - (b) Navigate to the details page of a specific message (message_detail).
 - **Expected Result:** Admin should be able to view detailed information about the selected message.
- **Test Case 12:** Test deleting a message (message_delete)
 - **Steps:**
 - (a) Log in as an admin user.
 - (b) Navigate to the delete page of an existing message (message_delete).
 - (c) Confirm the deletion.
 - **Expected Result:** The selected message should be deleted from the database, and the admin should be redirected to the messages list page (messages).

5. Access Control:

- **Test Case 13:** Test whether non-admin users are redirected away from admin-specific pages.
 - **Steps:**
 - (a) Log in as a non-admin user (e.g., a customer).
 - (b) Attempt to access admin-specific pages such as pet management or message management.
 - **Expected Result:** Non-admin users should be redirected to their respective pages (e.g., user page) and should not be able to access admin-specific functionalities.
- **Test Case 14:** Test whether admin-only functionalities are accessible only to users with admin privileges.
 - **Steps:**
 - (a) Log in as an admin user.
 - (b) Attempt to access admin-only functionalities.

- **Expected Result:** Admin users should have access to admin-only functionalities such as pet management and message management.
- **Test Case 15:** Test whether users with customer privileges can access their own pages (user page) but not admin-specific pages.
 - **Steps:**
 - (a) Log in as a customer user.
 - (b) Attempt to access their own page (user-page) and admin-specific pages.
 - **Expected Result:** Customer users should be able to access their own page but should be redirected away from admin-specific functionalities.

6. Profile Creation:

- **Test Case 16:** Test whether a customer profile is created upon admin registration.
 - **Steps:**
 - (a) Register as an admin user.
 - (b) Verify if a corresponding customer profile is created in the database.
 - **Expected Result:** Upon successful admin registration, a customer profile should be created in the database linked to the admin user.