

**NANYANG  
TECHNOLOGICAL  
UNIVERSITY**  
**SINGAPORE**

**SC2006 - Software Engineering**  
**Lab 1 Deliverables**

<b>Lab Group</b>	SCMA
<b>Team</b>	FetchMeHome
<b>Members</b>	BENJAMIN YEOH JUN JIE (U2321438C)
	BASKAR KAYALVIZHI ATHITHIYA (U2322885L)
	TIN JING LUN JAVIER (U2310906E)
	TRISTAN AMADEUS SURYA (U2320674L)
	TOMAR YASHWARDHAN SINGH (U2323694E)
	ZOU XUEHAO (U2322789L)

# Table of contents

<b>1. Documentation of Functional &amp; non-functional requirements</b>	<b>3</b>
A. Functional Requirements	3
B. Non-Functional Requirements	12
<b>2. Data Dictionary</b>	<b>13</b>
<b>3. Initial Use Case Model, consisting of Use Case diagram &amp; Use Case descriptions</b>	<b>15</b>
A. Use Case Diagram	15
B. Use Case Descriptions	16
<b>4. UI Mockups</b>	<b>69</b>

## **1. Documentation of functional and non-functional requirements**

### **A. Functional Requirements**

1. FetchMeHome shall allow Users to be authenticated and use the application.
  - 1.1. FetchMeHome shall allow Users to create an account
    - 1.1.1. FetchMeHome shall allow users to input their Full Name, E-Mail, Password and Phone No.
    - 1.1.2. FetchMeHome shall create an account with the information entered by the user.
  - 1.2. FetchMeHome shall allow Users to sign in using the account they have created previously
    - 1.2.1. FetchMeHome shall allow Users to enter their E-Mail and Password to log into the application.
    - 1.2.2. FetchMeHome shall mask the Password entered by the Users by replacing the actual text with dots unless the Users chooses to unmask it.
    - 1.2.3. If the E-Mail and Password do not match, FetchMeHome shall display “E-mail and Password do not match” to the user.
    - 1.2.4. If the E-mail and Password match, FetchMeHome shall log the User in and navigate the User to the home screen of the Application.
  - 1.3. FetchMeHome shall guide First-Time Users through the app.
  - 1.4. FetchMeHome shall allow Users to edit their profile details.

- 1.4.1. FetchMeHome shall allow Users to edit their name.
  - 1.4.2. FetchMeHome shall allow Users to edit their mobile number.
  - 1.4.3. FetchMeHome shall allow Users to edit their email.
  - 1.4.4. FetchMeHome shall allow Users to edit their profile photo.
  - 1.4.5. FetchMeHome shall allow Users to edit their location.
  - 1.4.6. FetchMeHome shall allow Users to edit their biography.
- 1.5. FetchMeHome shall allow Users to view and query all features
  - 1.5.1. FetchMeHome shall display all features via a list interface to provide a comprehensible view of all available options.
  - 1.5.2. FetchMeHome shall provide a find option that allows Users to find any missing pets near their vicinity.
  - 1.5.3. FetchMeHome shall provide an adopt option that allows Users to adopt a pet based on the listing.
2. FetchMeHome shall allow Users to view detailed information of all functions available on the mobile application.
  - 2.1. FetchMeHome shall allow Users to view details about Lost Pet listings.
    - 2.1.1. FetchMeHome shall display the Lost Pet button on the far left beside the Home button.
    - 2.1.2. FetchMeHome shall display all Lost Pets posts for Users to view.
    - 2.1.3. FetchMeHome shall allow Users to filter Lost Pets posts based on location.
    - 2.1.4. FetchMeHome shall allow Users to click on individual posts to expand and view more details.

- 2.1.5. FetchMeHome shall display a photo of the lost animal which will be uploaded by the respective pet owner.
  - 2.1.6. FetchMeHome shall display a real time map which shows the last seen location of the pet.
  - 2.1.7. FetchMeHome shall display descriptions of the animal which will be provided by the respective pet owner.
- 2.2. FetchMeHome shall allow Users to view details about Adoption of pets.
  - 2.2.1. FetchMeHome shall display the Adoption button on the far right beside the Home button.
  - 2.2.2. FetchMeHome shall allow Users to click on individual posts to expand and view more details.
  - 2.2.3. FetchMeHome shall display a photo of the pet for adoption which will be uploaded by the respective pet owner.
  - 2.2.4. FetchMeHome shall display descriptions of the pet which will be provided by the respective pet owner.
3. FetchMeHome shall allow Owners to perform Owner specific tasks
  - 3.1. FetchMeHome shall allow Owners to create new details of missing pets.
    - 3.1.1. Owners can upload images of the missing pet.
    - 3.1.2. Owners can enter the name of the missing pet.
    - 3.1.3. Owners can enter the age of the missing pet.
    - 3.1.4. Owners can enter the gender of the missing pet.
    - 3.1.5. Owners can enter the breed of the missing pet.
    - 3.1.6. Owners can enter the last seen location of the missing pet.

- 3.1.7. Owners can enter the activeness of the missing pet.
  - 3.1.8. Owners can enter the health condition of the missing pet.
  - 3.1.9. Owners can enter the notable features of the missing pet.
  - 3.1.10. Owners can enter a short description of the missing pet.
  - 3.1.11. Owners can enter the bounty reward for the missing pet.
  - 3.1.12. Owners can click the upload button to post their listing.
- 3.2. FetchMeHome shall allow Owners to edit details of missing pets.
- 3.2.1. FetchMeHome shall display an “Edit” button in the listing.
  - 3.2.2. Upon clicking the “Edit” button, FetchMeHome shall allow Owners to upload new images and/or edit any of their preferred descriptions.
  - 3.2.3. FetchMeHome shall display a “Update” button to allow Owners to save the changes they made permanently.
- 3.3. FetchMeHome shall allow Owners to verify images uploaded by PetFinders.
- 3.3.1. FetchMeHome shall display a “Verify” button on the listing.
  - 3.3.2. Upon clicking the “Verify” button, FetchMeHome shall take the Owner to a page of photos uploaded by PetFinder.
  - 3.3.3. FetchMeHome shall allow Owners to either accept or reject each request.
  - 3.3.4. FetchMeHome shall display an “Approve” and a “Deny” button.
  - 3.3.5. Upon clicking the “Approve” button, FetchMeHome shall provide the Owner with the contact details of the PetFinder.

- 3.3.6. If the “Deny” button is selected, FetchMeHome will remove the image uploaded by the PetFinder.
- 3.4. FetchMeHome shall allow Owners to flag PetFinders for misconduct.
  - 3.4.1. FetchMeHome shall display a “Flag” button on the listing.
  - 3.4.2. Upon clicking the “Flag” button, FetchMeHome shall take the Owner to a “Report User” page.
  - 3.4.3. FetchMeHome shall allow the Owner to enter a short description of the situation.
  - 3.4.4. Upon submitting the report, FetchMeHome shall notify the Admin.
- 3.5. FetchMeHome shall allow Owners to create new adoption pet listings.
  - 3.5.1. Owners can upload images of the pet for adoption.
  - 3.5.2. Owners can enter the name of the pet for adoption.
  - 3.5.3. Owners can enter the age of the pet for adoption.
  - 3.5.4. Owners can enter the species of the pet for adoption.
  - 3.5.5. Owners can enter the breed of the pet for adoption.
  - 3.5.6. Owners can enter a short description of the pet for adoption.
  - 3.5.7. Owners can click the upload button to post their listing.
- 3.6. FetchMeHome shall allow Owners to edit details of the pet for adoption.
  - 3.6.1. FetchMeHome shall display an “Edit” button in the listing.
  - 3.6.2. Upon clicking the “Edit” button, FetchMeHome shall allow Owners to upload new images and/or edit any of their preferred descriptions.

- 3.6.3. FetchMeHome shall display a “Update” button to allow Owners to save the changes they made permanently.
  - 3.7. FetchMeHome shall allow Owners to review Adopter requests for adoption.
    - 3.7.1. FetchMeHome shall display an “Accept” and a “Decline” button for the Owner to choose.
    - 3.7.2. Upon clicking the “Accept” button, FetchMeHome shall provide the contact details of the Adopter to the Owner to allow them to decide on the meeting details.
    - 3.7.3. Upon clicking the “Decline” button, FetchMeHome shall deny the Adopter request.
  - 3.8. FetchMeHome shall allow Owners to flag Adopters for misconduct.
    - 3.8.1. FetchMeHome shall display a “Flag” button on the listing.
    - 3.8.2. Upon clicking the “Flag” button, FetchMeHome shall take the Owner to a “Report User” page.
    - 3.8.3. FetchMeHome shall allow the Owner to enter a short description of the situation.
    - 3.8.4. Upon submitting the report, FetchMeHome shall notify the Administrator.
4. FetchMeHome shall allow PetFinders to perform PetFinder-specific tasks.
    - 4.1. FetchMeHome shall allow PetFinders to track and submit missing pets by uploading the photos of the pet found.

- 4.1.1. FetchMeHome shall use data provided by pet owners to predict the vicinity of the missing pet.
  - 4.1.2. FetchMeHome shall make use of the current weather conditions to adjust the predicted area of the missing pet.
  - 4.1.3. PetFinders can click on listings to see photos and descriptions of the pets.
  - 4.1.4. FetchMeHome shall display a “Report” button on the listing.
  - 4.1.5. PetFinders can scroll to the bottom of the listing to click the “Report” button, if they see a pet which matches the description.
  - 4.1.6. Upon clicking the “Report” button FetchMeHome shall allow PetFinder to upload the photos of the pet they found.
  - 4.1.7. FetchMeHome shall display a “Confirm” button.
  - 4.1.8. Upon clicking the “Confirm” button FetchMeHome shall send photos taken to the Owner for verification.
5. FetchMeHome shall allow Adopters to perform Adopter-specific functions.
    - 5.1. FetchMeHome shall allow Adopters to search for their desired pet to adopt.
      - 5.1.1. Adopters can enter the details of their desired pets and their specific requirements.
      - 5.1.2. FetchMeHome shall then display a list of matching pets up for adoption.
      - 5.1.3. FetchMeHome shall allow Adopters to click on individual posts to expand for more details and descriptions of the pet.

- 5.1.4. FetchMeHome shall display an “Adopt” button.
  - 5.1.5. Upon clicking the “Adopt” button FetchMeHome shall display a confirmation page with the details.
  - 5.1.6. FetchMeHome shall display a “Confirm” button for the Adopter to select.
  - 5.1.7. Upon clicking the “Confirm” button the details of the Adopter will be sent to the pet Owner for verification and approval.
- 5.2. FetchMeHome shall allow Adopters to flag Owners for misconduct.
  - 5.2.1. FetchMeHome shall display a “Flag” button on the listing.
  - 5.2.2. Upon clicking the “Flag” button, FetchMeHome shall take the Adopter to a “Report User” page.
  - 5.2.3. FetchMeHome shall allow the Adopter to enter a short description of the situation.
  - 5.2.4. Upon submitting the report, FetchMeHome shall notify the Administrator.
6. FetchMeHome shall allow Administrators to perform Administrators specific functions.
  - 6.1. FetchMeHome shall allow Administrators to review Users from the platform if there is any usage misconduct or violation of terms.
    - 6.1.1. If the Administrator deems the flagged Users to be at fault, FetchMeHome shall allow Administrators to ban the Users from the platform.

- 6.1.2. If the flagged User is not at fault, FetchMeHome shall allow Administrators to dismiss the report.
- 6.1.3. If any listing is reported by a user as inappropriate or misleading, FetchMeHome shall allow Administrators to remove them from the platform.
- 6.1.4. If the flagged listing is not inappropriate or misleading, FetchMeHome shall allow Administrators to dismiss the report.

## B. Non-Functional Requirements

<b>Usability</b>	<ul style="list-style-type: none"><li>• 90% of users should be able to login in 5 min.</li><li>• 90% of users should be able to navigate the 2 different sub categories and home page.</li></ul>
<b>Reliability</b>	<ul style="list-style-type: none"><li>• The system should have at least 95% uptime to ensure users can always search for lost pets.</li><li>• Launching the app should be available in 5 seconds.</li></ul>

<b>Performance</b>	<ul style="list-style-type: none"> <li>The system should process the Users preferences and personality test results with the help of AI to return the most suitable pet to them.</li> </ul>
<b>Maintainability</b>	<ul style="list-style-type: none"> <li>Code should follow best practices (SOLID, clean architecture).</li> </ul>
<b>Scalability</b>	<ul style="list-style-type: none"> <li>The platform should support at least 1000 concurrent users without performance degradation.</li> <li>But if needed, scaling horizontally by increasing the quantity of servers to handle more traffic should be available.</li> </ul>
<b>Security</b>	<ul style="list-style-type: none"> <li>User authentication must use hashed and salted passwords (bcrypt/Argon2/PBKDF2).</li> <li>Prevent unauthorized access (e.g., Users should not have access to administrator privileges).</li> </ul>

## 2. Data Dictionary

Term	Definition
<b>User</b>	An individual who interacts with the FetchMeHome application. Users can be pet owners, finders, adopters or administrators.
<b>Authentication</b>	The process of verifying a user's identity through email and password.
<b>Profile</b>	A collection of user information, including full name, email, phone number, and profile picture.
<b>Weather Conditions</b>	Environmental factors like temperature, rainfall, and wind speed that may affect a pet's movement.
<b>Bounty Reward</b>	A monetary reward set by the pet owner to encourage finders to return their pet.
<b>Verification</b>	The process by which Owners approve or reject a PetFinder's reported pet sighting.
<b>Report User</b>	A feature allowing users to flag misconduct and notify administrators.
<b>Report Listing</b>	A feature to flag listings when it is fake or inappropriate and this will notify administrators.
<b>Lost Pet Listing</b>	A post created by an Owner containing details about a missing pet.
<b>Review Lost Pet Submission</b>	Owners can verify if the pet in the photo submission by the Pet Finder is really their pet or not.
<b>View Adoption Request</b>	For the owner to verify and give their contact details upon verification of the adopter's information.
<b>Adoption Listing</b>	A record created by an Owner containing details about a pet available for adoption.
<b>Pet Owner</b>	The Owner can post the listings for adoption requests and post the lost dog listing.
<b>PetFinder</b>	A person who finds the pet and/or uploads photos and/or return the pet to the owner

<b>Adopter</b>	A person who wishes to adopt a pet from the listings.
<b>Administrator</b>	An administrator who manages the platform and handles misconduct reports and listings.
<b>Kebab Menu</b>	A kebab menu is a menu icon that displays a list of options or actions when clicked.
<b>Personality Test</b>	A series of questions to determine which breed of pets are suitable for the user.
<b>Walkthrough</b>	A one time flow of events that guide the for first time users on how to navigate the different features of our app.

### 3. Initial Use Case Model, including Use Case Diagram and Use Case descriptions.

#### A. Use Case Diagram



## B. Use Case Descriptions

### Functional Requirement #1

#### I.I CreateAccount

Use Case ID:	#1-1		
Use Case Name:	CreateAccount		
Created By:	Baskar Kayalvizhi Athithiya	Last Updated By:	Baskar Kayalvizhi Athithiya
Date Created:	8/2/2025	Date Last Updated:	8/2/25

Actor:	User
Description:	Allows first-time users to create an Account.
Preconditions:	None
Postconditions:	An Account is created for the User.
Priority:	High
Frequency of Use:	High

Flow of Events:	<ol style="list-style-type: none"> <li>1. System will prompt the user to enter their Full Name, E-mail Address, Phone Number, Password, and Retype Password.</li> <li>2. System checks all the required fields.</li> <li>3. User selects “Validate Phone Number”.</li> <li>4. System generates a One-Time-Password (OTP) and sends the OTP to the User’s entered Phone Number.</li> <li>5. User enters the OTP.</li> <li>6. An Account is created for the User and the database is updated.</li> </ol>
Alternative Flows:	<p><u>AF-S2: User did not fill in all required fields in the form</u></p> <ol style="list-style-type: none"> <li>1. User did not fill in all required fields in the form.</li> <li>2. Form is not submitted, the system prompts User to fill in all required fields.</li> <li>3. System returns to step 1.</li> </ol> <p><u>AF-S3: User phone number is already registered</u></p> <ol style="list-style-type: none"> <li>1. User phone number is already registered with the app.</li> <li>2. System will prompt the User to login using the login use case.</li> </ol> <p><u>AF-S4: User OTP entered is incorrect</u></p> <ol style="list-style-type: none"> <li>1. User OTP entered is incorrect.</li> <li>2. System will prompt the User to enter the correct OTP.</li> <li>3. System returns to step 4.</li> </ol>
Exceptions:	None
Includes:	Walkthrough
Special Requirements:	System needs to validate User input data
Assumptions:	None

Notes and Issues:	None
-------------------	------

## I.II Login

Use Case ID:	#1-2		
Use Case Name:	Login		
Created By:	Baskar Kayalvizhi Athithiya	Last Updated By:	Baskar Kayalvizhi Athithiya
Date Created:	8/2/2025	Date Last Updated:	8/2/2025

Actor:	User
Description:	Allows the User to login using their phone number and password.
Preconditions:	Users must have an existing FetchMeHome account.
Postconditions:	The User is logged in to the FetchMeHome application and is navigated to the home screen of the application.
Priority:	High
Frequency of Use:	High

Flow of Events:	<ol style="list-style-type: none"> <li>1. System prompts the User to enter their phone number and password.</li> <li>2. The User enters their Phone number and Password into the respective fields.</li> <li>3. The User selects “Login”.</li> <li>4. System checks all the required fields.</li> <li>5. The User is navigated to the home screen of the FetchMeHome application.</li> </ol>
Alternative Flows:	<u>AF-S4: User did not fill in all required fields in the page</u> <ol style="list-style-type: none"> <li>4. User did not fill in all required fields in the page.</li> <li>5. Login is not processed, the system prompts User to fill in all required fields.</li> <li>6. System returns to step 1.</li> </ol>
Exceptions:	None
Includes:	None
Special Requirements:	System needs to validate the email and password provided by the User.
Assumptions:	User already has an existing FetchMeHome account.
Notes and Issues:	The Password will be masked as dots but the User can view them by pressing the eye icon.

### I.III EditProfile

Use Case ID:	#1-3
--------------	------

Use Case Name:	EditProfile		
Created By:	Baskar Kayalvizhi Athithiya	Last Updated By:	Baskar Kayalvizhi Athithiya
Date Created:	8/2/2025	Date Last Updated:	8/2/2025

Actor:	User
Description:	Users can edit the details of their profile such as phone number, email address, and name, and add a profile photo
Preconditions:	<ol style="list-style-type: none"> <li>1. The User must have an existing FetchMeHome application</li> <li>2. The User must be logged in to the FetchMeHome application</li> </ol>
Postconditions:	The information given by the User will be updated on their profile
Priority:	Low
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> <li>1. The User will click the “Edit Profile” button on their profile.</li> <li>2. The system shall prompt the User to key in the updated personal particulars.</li> <li>3. The User shall then click the “Confirm” button to save the changes made to their profile.</li> </ol>

Alternative Flows:	None
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	The User has an existing FetchMeHome account
Notes and Issues:	None

#### I.IV WalkThrough

Use Case ID:	#1-4		
Use Case Name:	WalkThrough		
Created By:	Baskar Kayalvizhi Athithiya	Last Updated By:	Baskar Kayalvizhi Athithiya
Date Created:	8/2/2025	Date Last Updated:	8/2/2025

Actor:	User
--------	------

Description:	Users who have just created an account will be introduced to the features of the FetchMeHome application.
Preconditions:	It must be the first time a User is logging in to the application after creating their FetchMeHome account
Postconditions:	The User has been introduced to the interface of the application as well as the features available in the application
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> <li>1. The User will click the “Login” button after creating a FetchMeHome account.</li> <li>2. Beside the icon of the AdoptAPet function, the System shall display a description of how to navigate it.</li> <li>3. The User shall then press the “I Understand” button.</li> <li>4. Beside the icon of the FindLostPet function, the System shall display a description of how to navigate it.</li> <li>5. The User shall then press the “I Understand” button.</li> </ol>
Alternative Flows:	None
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

## I.V ViewMissingPetList

Use Case ID:	#1-5		
Use Case Name:	ViewMissingPetList		
Created By:	Baskar Kayalvizhi Athithiya	Last Updated By:	Baskar Kayalvizhi Athithiya
Date Created:	8/2/2025	Date Last Updated:	8/2/2025

Actor:	User
Description:	Users can view all available Lost Pet listings
Preconditions:	The User must be logged in to the FetchMeHome application
Postconditions:	The User will be able to view all the listings of Lost Pets
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. The User shall click on the “LostPet” icon on the navigation bar of the home screen of the FetchMeHome application.</li> <li>2. The System shall then display all the posts of Lost Pets.</li> </ol>

	3. The User can then click on any individual post to view more details of the Lost Pets.
Alternative Flows:	None
Exceptions:	None
Includes:	ViewMap
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

#### I.VI ViewPetAdoptionList

Use Case ID:	#1-6		
Use Case Name:	ViewPetAdoptionList		
Created By:	Baskar Kayalvizhi Athithiya	Last Updated By:	Baskar Kayalvizhi Athithiya
Date Created:	8/2/2025	Date Last Updated:	8/2/2025

Actor:	User
Description:	Users can view all listings of pets that are up for adoption
Preconditions:	The User must be logged in to the FetchMeHome application
Postconditions:	The User will be able to view all listings of the pets that are up for adoption
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. User shall click the “AdoptPet” icon on the navigation bar of the home screen of the FetchMeHome application.</li> <li>2. The System shall then display all listings of Pets that are up for adoption.</li> <li>3. Users can click on any individual post to expand it and view more details of the pet that is up for adoption.</li> </ol>
Alternative Flows:	None
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

## I.VII ManageProfile

Use Case ID:	#1-7		
Use Case Name:	ManageProfile		
Created By:	Baskar Kayalvizhi Athithiya	Last Updated By:	Baskar Kayalvizhi Athithiya
Date Created:	10/2/2025	Date Last Updated:	10/2/2025

Actor:	User
Description:	Users can manage the status of their profile
Preconditions:	The User must be logged in to their FetchMeHome account
Postconditions:	The User's account details and status will be updated by the details provided by the User and the actions taken
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> <li>1. The User will click on the “Profile” icon on the navigation bar of the FetchMeHome home page</li> <li>2. The System shall then display the profile page of the User.</li> <li>3. The System shall then display the initial details provided by the User during Account creation.</li> </ol>

	<p>4. The System shall display the “Edit Profile” button and the “Delete Account” buttons at the bottom of the profile.</p> <p>5. If the User clicks on the “Edit Profile” button then they will use the included use case EditProfile to edit their profile.</p> <p>6. If the User clicks on the “Delete Account” button then they will use the included use case DeleteAccount to delete their profile.</p>
Alternative Flows:	None
Exceptions:	None
Includes:	EditProfile, DeleteAccount
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

## I.VIII DeleteAccount

Use Case ID:	#1-8		
Use Case Name:	DeleteAccount		
Created By:	Baskar Kayalvizhi Athithiya	Last Updated By:	Baskar Kayalvizhi Athithiya

Date Created:	10/2/2025	Date Last Updated:	10/2/2025
---------------	-----------	--------------------	-----------

Actor:	User
Description:	The User can delete their account
Preconditions:	The User must be logged in to the FetchMeHome application
Postconditions:	The User's Account will be deleted
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> <li>1. The User will click on the “Delete Account” button</li> <li>2. System shall then display the “Confirm” and “Cancel” buttons</li> <li>3. If the User clicks on the “Confirm” button the Account will be deleted</li> <li>4. The User will be transported back to their own Profile page</li> </ol>
Alternative Flows:	None
Exceptions:	<p><u>AF-S2: User clicks the “Cancel” button</u></p> <ol style="list-style-type: none"> <li>1. If the User clicks the “Cancel” button, the system will return to the profile page.</li> </ol>
Includes:	None

Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

### I.IX FlagListing

Use Case ID:	#1-9		
Use Case Name:	FlagListing		
Created By:	Benjamin Yeoh	Last Updated By:	Benjamin Yeoh
Date Created:	8/2/25	Date Last Updated:	8/2/25

Actor:	User
Description:	Allows Users to flag listings deemed inappropriate.
Preconditions:	User is logged in and authenticated.
Postconditions:	Administrators are notified.

Priority:	Medium
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> <li>1. The System shall display a “FlagListing” button.</li> <li>2. If the User deems the listing as inappropriate, the User can click the “FlagListing” button.</li> <li>3. The System will prompt the User to fill in details of the scenario in a short description box.</li> <li>4. Checks are made to ensure the description box is filled.</li> <li>5. User clicks the “Submit” button, a notification is sent to the Administrators and the database is updated.</li> </ol>
Alternative Flows:	<p><u>AF-S4: User did not fill up required field in the form</u></p> <ol style="list-style-type: none"> <li>1. User did not fill up the required field in the form.</li> <li>2. User clicks the “Submit” button.</li> <li>3. Checks that the required field is not filled up.</li> <li>4. User is prompted to fill in the required field, no changes to the form.</li> <li>5. System returns to step 3.</li> </ol>
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

## I.X LogOut

Use Case ID:	#1-10		
Use Case Name:	LogOut		
Created By:	Baskar Kayalvizhi Athithiya	Last Updated By:	Baskar Kayalvizhi Athithiya
Date Created:	12/2/25	Date Last Updated:	12/2/25

Actor:	User
Description:	Allows Users to log out of their FetchMeHome account
Preconditions:	User is logged in and authenticated.
Postconditions:	User is logged out of the FetchMeHome application
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> <li>1. The System shall display a “Log Out” button in the profile of the User.</li> <li>2. The User shall then click the “Log Out” button.</li> </ol>

	<ol style="list-style-type: none"> <li>3. The System shall then display the “Confirm” and “Cancel” buttons.</li> <li>4. The User shall then click the “Confirm” button.</li> <li>5. The User shall then be logged out of the FetchMeHome application.</li> <li>6. The System shall then display the Login page of the FetchMeHome application.</li> </ol>
Alternative Flows:	<u>AF-S3: The User chooses to click the “Cancel” button instead</u> <ol style="list-style-type: none"> <li>1. The User shall click on the “Cancel” button.</li> <li>2. The System shall then return to the profile of the User.</li> </ol>
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

#### I.XI ViewMap

Use Case ID:	#1-11
Use Case Name:	ViewMap

Created By:	Baskar Kayalvizhi Athithiya	Last Updated By:	Baskar Kayalvizhi Athithiya
Date Created:	12/2/25	Date Last Updated:	12/2/25

Actor:	User
Description:	Allows Users to view the map which includes the last seen location of the lost pet as well as the surrounding weather conditions
Preconditions:	User is logged in and authenticated.
Postconditions:	User will know the last seen location as well as the surrounding weather conditions
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. The User shall click on the “Last Seen” button on the missing pet listing.</li> <li>2. The System shall then display the local map.</li> <li>3. The map shall include the last seen location of the missing pet which has been provided by the Pet Owner.</li> </ol>

	4. The map shall also display nearby weather conditions to assist the User with finding the missing pet.
Alternative Flows:	None
Exceptions:	Most recent data is unavailable due to connection and updating issues
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

## Functional Requirement #2

### II.I ManageRequest

Use Case ID:	#2-1		
Use Case Name:	ManageRequest		
Created By:	Benjamin Yeoh	Last Updated By:	Benjamin Yeoh

Date Created:	8/2/25	Date Last Updated:	8/2/25
---------------	--------	--------------------	--------

Actor:	Owner
Description:	Allows Owner to manage requests made by PetFinder and/or Adopter.
Preconditions:	Owner is logged in and is authenticated.
Postconditions:	None
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. System shall display a “ManageRequests” button in the profile page.</li> <li>2. Upon clicking the button, the Owner will be provided with 3 use cases:             <ol style="list-style-type: none"> <li>a. ViewRequestList</li> <li>b. ReviewAdoptionRequest</li> <li>c. ReviewLostPetSubmission</li> </ol> </li> <li>3. If the Owner selects the activity ViewRequestList, then the Owner uses the included use case ViewRequestList to view all requests directed to the Owner.</li> <li>4. If the Owner selects the activity ReviewAdoptionRequest, then the Owner uses the included use case</li> </ol>

	<p>ReviewAdoptionRequest to review adoption request(s) made by Adopter(s).</p> <p>5. If the Owner selects the activity ReviewLostPetSubmission, then the Owner uses the included use case ReviewLostPetSubmission to review missing pet report(s) made by PetFinder(s).</p>
Alternative Flows:	None
Exceptions:	None
Includes:	<ol style="list-style-type: none"> <li>1. ViewRequestList</li> <li>2. ReviewAdoptionRequest</li> <li>3. ReviewLostPetSubmission</li> </ol>
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

## II.II ViewRequestList

Use Case ID:	#2-2
--------------	------

Use Case Name:	ViewRequestList		
Created By:	Benjamin Yeoh	Last Updated By:	Benjamin Yeoh
Date Created:	8/2/25	Date Last Updated:	8/2/25

Actor:	Owner
Description:	Allows Owner to view all available requests / reports made by PetFinder and/or Adopter.
Preconditions:	Owner is logged in and is authenticated.
Postconditions:	None
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. System shall display all available request(s) / report(s) directed to the Owner.</li> <li>2. For each request made by the Adopter, the System shall display the Adopter's information and details.</li> </ol>

	3. For each report made by the PetFinder, the System shall display the PetFinder's information and details.
Alternative Flows:	None
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

### II.III ReviewAdoptionRequest

Use Case ID:	#2-3		
Use Case Name:	ReviewAdoptionRequest		
Created By:	Benjamin Yeoh	Last Updated By:	Benjamin Yeoh
Date Created:	8/2/25	Date Last Updated:	8/2/25

Actor:	Owner
Description:	Allows Owner to view all available adoption requests made by the Adopter.
Preconditions:	Owner is logged in and is authenticated.
Postconditions:	None
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. System shall display a list of all Adopter(s) adoption requests with their personality test results attached as an image.</li> <li>2. For each request, the System will provide an “Accept” and a “Decline” button for the Owner to determine if the respective Adopter is eligible for adoption.</li> <li>3. Upon clicking the “Accept” button for a particular request, the System will provide the Owner with the respective Adopter’s Phone No.</li> </ol>
Alternative Flows:	<p><u>AF-S2: Owner selects the “Decline” button</u></p> <ol style="list-style-type: none"> <li>1. If the Owner selects the “Decline” button, the System will delete the specific request.</li> <li>2. System returns to step 1.</li> </ol>

Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

### II.III ReviewLostPetSubmission

Use Case ID:	#2-3		
Use Case Name:	ReviewLostPetSubmission		
Created By:	Benjamin Yeoh	Last Updated By:	Benjamin Yeoh
Date Created:	8/2/25	Date Last Updated:	8/2/25

Actor:	Owner
--------	-------

Description:	Allows Owner to view all lost pet submission reports made by PetFinder.
Preconditions:	Owner is logged in and is authenticated.
Postconditions:	None
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. System shall display a list of all lost pet submission reports from PetFinder(s).</li> <li>2. For each report, the System shall display the uploaded image(s) provided by the PetFinder, as well as a “Accept” and a “Decline” button.</li> <li>3. Owner shall determine if any of the images uploaded is his pet.</li> <li>4. If there is an image match for a particular report, the Owner will click the “Accept” button, and the system will provide the Owner with the respective PetFinder Phone No.</li> </ol>
Alternative Flows:	<u>AF-S2: Owner selects the “Decline” button</u> <ol style="list-style-type: none"> <li>1. If the Owner selects the “Decline” button, the System will delete the specific report.</li> <li>2. System returns to step 1.</li> </ol>
Exceptions:	None

Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

#### II.IV CreateListing

Use Case ID:	#2-4		
Use Case Name:	CreateListing		
Created By:	Benjamin Yeoh	Last Updated By:	Benjamin Yeoh
Date Created:	8/2/25	Date Last Updated:	8/2/25

Actor:	Owner
Description:	Allows Owners to create listings for their missing pet(s) and/or pet(s) up for adoption.

Preconditions:	Owner is logged in and is authenticated.
Postconditions:	None
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. System shall display an “Add” icon in the home page.</li> <li>2. Upon clicking the “Add” icon, Owner will be provided with 2 use cases:             <ol style="list-style-type: none"> <li>a. CreateAdoptListing</li> <li>b. CreateMissingPetListing</li> </ol> </li> <li>3. If the Owner selects the activity CreateAdoptListing, then the Owner uses the included use case CreateAdoptListing to create a listing for their pet(s)to put up for adoption.</li> <li>4. If the Owner selects the activity CreateMissingPetListing, then the Owner uses the included use case CreateMissingPetListing to create a listing for their missing pet(s).</li> </ol>
Alternative Flows:	None
Exceptions:	None
Includes:	<ol style="list-style-type: none"> <li>1. CreateAdoptListing</li> <li>2. CreateMissingPetListing</li> </ol>
Special Requirements:	None

Assumptions:	None
Notes and Issues:	None

## II.V CreateAdoptListing

Use Case ID:	#2-5		
Use Case Name:	CreateAdoptListing		
Created By:	Benjamin Yeoh	Last Updated By:	Benjamin Yeoh
Date Created:	8/2/25	Date Last Updated:	8/2/25

Actor:	Owner
Description:	Allows Owner to create a listing for pets up for adoption.
Preconditions:	Owner is logged in and is authenticated.
Postconditions:	Listing is successfully posted.

Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. Owner fills in all the required forms presented to them.</li> <li>2. Owner clicks the “Submit” button, CreateAdoptListing is called.</li> <li>3. Checks are made to ensure all the required fields are filled up.</li> <li>4. Listing is created and added to the list of pets up for adoption.</li> </ol>
Alternative Flows:	<p><u>AF-S3: Owner did not fill up all required fields in the form</u></p> <ol style="list-style-type: none"> <li>1. Owner did not fill up all required fields in the form.</li> <li>2. Owner clicks the “Submit” button.</li> <li>3. The System flags out that some of the required fields are not filled up.</li> <li>4. System returns to step 1.</li> </ol>
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

## II.VI CreateMissingPetListing

Use Case ID:	#2-6		
Use Case Name:	CreateMissingPetListing		
Created By:	Benjamin Yeoh	Last Updated By:	Benjamin Yeoh
Date Created:	8/2/25	Date Last Updated:	8/2/25

Actor:	Owner
Description:	Allows Owner to create a listing for their missing pet(s).
Preconditions:	Owner is logged in and authenticated.
Postconditions:	Listing is successfully posted.
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. Owner fills in all the required forms presented to them.</li> <li>2. Owner clicks the “Submit” button, CreateMissingPetListing is called.</li> </ol>

	<ul style="list-style-type: none"> <li>3. Checks are made to ensure all the required fields are filled up.</li> <li>4. Listing is sent and updated to the database.</li> </ul>
Alternative Flows:	<p><u>AF-S3: Owner did not fill up all required fields in the form</u></p> <ul style="list-style-type: none"> <li>1. Owner did not fill up all required fields in the form.</li> <li>2. Owner clicks the “Submit” button, CreateMissingPetListing is called.</li> <li>3. Checks that some required fields are not filled up.</li> <li>4. The System flags out that some of the required fields are not filled up.</li> <li>5. System returns to step 1.</li> </ul>
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

## II.VII ManageMyListings

Use Case ID:	#2-7
Use Case Name:	ManageMyListings

Created By:	Benjamin Yeoh	Last Updated By:	Benjamin Yeoh
Date Created:	8/2/25	Date Last Updated:	8/2/25

Actor:	Owner
Description:	Allows Owners to manage all their available listing(s).
Preconditions:	Owner is logged in and is authenticated.
Postconditions:	None
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. System shall display a “ManageMyListings” button in the profile page.</li> <li>2. Upon clicking the “ManageMyListings” button, the system shall display all available listing(s) which the Owner had uploaded.</li> <li>3. In each listing, the System shall display a kebab menu icon.</li> <li>4. Upon clicking the icon, the System shall provide the Owner with 2 use cases:             <ol style="list-style-type: none"> <li>a. Edit</li> <li>b. Delete</li> </ol> </li> </ol>

	<p>5. If the Owner selects the activity Edit, he will use the included use case Edit to edit his listing.</p> <p>6. If the Owner selects the activity Delete, he will use the included use case Delete to delete his listing.</p>
Alternative Flows:	None
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	Owner has existing listings.
Notes and Issues:	None

## II.VIII EditListing

Use Case ID:	#2-8		
Use Case Name:	EditListing		
Created By:	Benjamin Yeoh	Last Updated By:	Benjamin Yeoh
Date Created:	8/2/25	Date Last Updated:	8/2/25

Actor:	Owner
Description:	Allows Owners to edit listings for their missing pet(s) and/or pet(s) up for adoption.
Preconditions:	Owner is logged in and is authenticated.
Postconditions:	Listing is successfully edited.
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> <li>1. The Owner selects the “Edit” button, he would be allowed to edit on his selected listing.</li> <li>2. System shall display a “Submit” button.</li> <li>3. Upon clicking the “Submit” button, the system checks that all required fields are filled up.</li> <li>4. The listing is updated.</li> </ol>
Alternative Flows:	<p><u>AF-S3: Owner did not fill up all required fields in the form</u></p> <ol style="list-style-type: none"> <li>1. Owner did not fill up all required fields in the form.</li> <li>2. Owner clicks the “Submit” button, EditMissingPetListing is called.</li> <li>3. Checks that some required fields are not filled up.</li> <li>4. System returns to step 1.</li> </ol>
Exceptions:	None

Includes:	None
Special Requirements:	None
Assumptions:	Owner has existing listings.
Notes and Issues:	None

## II.IX DeleteListing

Use Case ID:	#2-9		
Use Case Name:	DeleteListing		
Created By:	Benjamin Yeoh	Last Updated By:	Benjamin Yeoh
Date Created:	8/2/25	Date Last Updated:	8/2/25

Actor:	Owner
Description:	Owner is able to remove his listing(s).

Preconditions:	Owner is logged in and authenticated.
Postconditions:	Listing is successfully deleted
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> <li>1. The Owner selects the “Delete” button.</li> <li>2. System shall display a “Confirm” button and a “Cancel” button.</li> <li>3. Upon clicking the “Confirm” button, the listing is removed from the database.</li> </ol>
Alternative Flows:	<p><u>AF-S2: Owner clicks the “Cancel” button</u></p> <ol style="list-style-type: none"> <li>1. If Owner clicks the “Cancel” button, the System will not remove the listing.</li> </ol>
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	Owner has existing listings
Notes and Issues:	None

## II.X FlagUser

Use Case ID:	#2-10		
Use Case Name:	FlagUser		
Created By:	Benjamin Yeoh	Last Updated By:	Benjamin Yeoh
Date Created:	8/2/25	Date Last Updated:	8/2/25

Actor:	Owner/Adopter/PetFinder
Description:	Allows Owner/Adopter/PetFinder to flag Users deemed inappropriate.
Preconditions:	Owner/Adopter/PetFinder is logged in and authenticated.
Postconditions:	Administrators are notified.
Priority:	Medium
Frequency of Use:	Low
Flow of Events:	1. The System shall display a “FlagUser” icon.

	<ol style="list-style-type: none"> <li>2. If the Owner/Adopter/PetFinder deems the User's actions (online or physical) as inappropriate, the Owner/Adopter/PetFinder can click the "FlagUser" icon.</li> <li>3. The System will prompt the Owner/Adopter/PetFinder to fill in details of the scenario in a short description box and/or any images / supporting evidence.</li> <li>4. Checks are made to ensure the description box is filled.</li> <li>5. Owner/Adopter/PetFinder clicks the "Submit" button, a notification is sent to the Administrators and the database is updated.</li> </ol>
Alternative Flows:	<p><u>AF-S4: Owner did not fill up required field in the form</u></p> <p>Owner/Adopter/PetFinder did not fill up the required field in the form.</p> <p>Owner/Adopter/PetFinder clicks the "Submit" button.</p> <p>Checks that the required field is not filled up.</p>
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	Owner has communicated to PetFinder and/or Adopter.
Notes and Issues:	None

## Functional Requirement #3

### III.I SubmitLostPet

Use Case ID:	#3-1		
Use Case Name:	SubmitLostPet		
Created By:	Baskar Kayalvizhi Athithiya	Last Updated By:	Baskar Kayalvizhi Athithiya
Date Created:	8/2/25	Date Last Updated:	9/2/25

Actor:	PetFinder
Description:	PetFinder can report missing pets in their vicinity.
Preconditions:	<ol style="list-style-type: none"><li>1. PetFinder is logged in and authenticated.</li><li>2. PetFinder can only report the missing animal through the corresponding LostPet listing</li></ol>
Postconditions:	The Owner will be notified.
Priority:	High

Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. System shall display a list of all Lost Pet(s).</li> <li>2. PetFinder clicks on the respective LostPet listing which corresponds to the pet they found.</li> <li>3. System shall display a “Report” button.</li> <li>4. If the PetFinder clicks on the “Report” button, the System will direct the PetFinder to upload image(s) of the pet.</li> <li>5. System displays a “Upload” and “Cancel” option.</li> <li>6. PetFinder clicks on the “Upload” button and uploads image(s) of the pet as an image attachment with a short optional description.</li> <li>7. System displays a “Confirm” and a “Cancel” button.</li> <li>8. PetFinder clicks on the “Confirm” button. Uploaded images will be sent to the Owner and he/she will be notified.</li> </ol>
Alternative Flows:	<u>AF-S5: PetFinder clicks “Cancel” button</u> <ol style="list-style-type: none"> <li>1. If the PetFinder clicks the “Cancel” button, the report will be dismissed and images will not be sent to the Owner.</li> <li>2. The PetFinder will be brought back to the selected LostPet Listing.</li> </ol>
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	The Pet found matches the image in the listing.
Notes and Issues:	None

## Functional Requirement #4

### IV.I SubmitAdoptionRequest

Use Case ID:	#4-1		
Use Case Name:	SubmitAdoptionRequest		
Created By:	Baskar Kayalvizhi Athithiya	Last Updated By:	Baskar Kayalvizhi Athithiya
Date Created:	11/2/25	Date Last Updated:	11/2/25

Actor:	Adopter
Description:	Adopters can send an Adoption Request to Owners for pet(s) they are interested in.
Preconditions:	Adopter is logged in and authenticated.
Postconditions:	The Adoption Request is successfully sent to the Owner of the Pet
Priority:	High

Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. The System displays the details of the Pet that is up for adoption.</li> <li>2. The System displays the “Adopt” button.</li> <li>3. The Adopter clicks on the “Adopt” button.</li> <li>4. System prompts the Adopter to fill in the following details:             <ol style="list-style-type: none"> <li>a. Age</li> <li>b. Gender</li> <li>c. Message to be sent to the Owner</li> </ol> </li> <li>5. System displays the “Confirm” and “Cancel” button.</li> <li>6. The Adopter clicks on the “Confirm” button.</li> <li>7. System sends the Adoption Request to the Owner of the Pet Listing.</li> </ol>
Alternative Flows:	<u>AF-S5: PetFinder clicks “Cancel” button</u> <ol style="list-style-type: none"> <li>1. Adopter terminates the adoption process and the request is not sent to the Owner.</li> <li>2. Adopter is brought back to the Pet Details page.</li> </ol>
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

#### IV.II CancelAdoptionRequest

Use Case ID:	#4-2		
Use Case Name:	CancelAdoptionRequest		
Created By:	Baskar Kayalvizhi Athithiya	Last Updated By:	Baskar Kayalvizhi Athithiya
Date Created:	11/2/25	Date Last Updated:	11/2/25

Actor:	Adopter
Description:	Adopter can cancel any Adoption Request they had previously sent to the Owner
Preconditions:	<ol style="list-style-type: none"> <li>1. Adopter is logged in and authenticated.</li> <li>2. Adopter had already sent an Adoption Request to the Owner</li> </ol>
Postconditions:	The Adoption Request sent to the Owner has been successfully removed.
Priority:	Medium

Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> <li>1. System displays the list of all adoption requests sent by the Adopter</li> <li>2. Adopter clicks on the kebab menu on the top right-hand side of the desired adoption request.</li> <li>3. The Kebab menu shows “Info” and “Cancel Request” buttons.</li> <li>4. Adopter clicks on “Cancel Request”.</li> <li>5. System requests Adopter to confirm cancellation request.</li> <li>6. Adopter confirms cancellation request.</li> <li>7. The Adopter successfully deletes the adoption request.</li> </ol>
Alternative Flows:	<p><u>AF-S3: Adopter clicks “Info” button</u></p> <ol style="list-style-type: none"> <li>1. System will display the pet details of the adoption request.</li> <li>2. System returns back to step 1.</li> </ol> <p><u>AF-S4: Adopter does not confirm the “Cancel Request”</u></p> <ol style="list-style-type: none"> <li>1. Selected adoption requests will not be changed.</li> <li>2. Adopter will be brought back to his/her list of adoption requests.</li> </ol>
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

#### IV.III ViewSubmittedRequestList

Use Case ID:	#4-3		
Use Case Name:	ViewSubmittedRequestList		
Created By:	Baskar Kayalvizhi Athithiya	Last Updated By:	Baskar Kayalvizhi Athithiya
Date Created:	11/2/25	Date Last Updated:	11/2/25

Actor:	Adopter
Description:	Adopters will be able to view all the adoption requests they have made for the pets that they are interested in
Preconditions:	<ol style="list-style-type: none"> <li>1. The Adopter is logged in authenticated by the System</li> <li>2. The Adopter must have previously made a successful adoption request to the Owner</li> </ol>
Postconditions:	The Adopter will be able to view the list of adoption requests they had made.
Priority:	Medium

Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> <li>1. System shows the list of all adoption requests that the Adopter had submitted.</li> <li>2. Adopter is able to view individual adoption request status by clicking the “View Status” button in each listing.</li> </ol>
Alternative Flows:	None
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

#### IV.IV PersonalityTest

Use Case ID:	#4-4		
Use Case Name:	PersonalityTest		
Created By:	Benjamin Yeoh	Last Updated By:	Benjamin Yeoh

Date Created:	8/2/25	Date Last Updated:	8/2/25
---------------	--------	--------------------	--------

Actor:	Adopter
Description:	Adopters can take a personality test to determine which pet is suitable for them.
Preconditions:	Adopter is logged in and authenticated.
Postconditions:	None
Priority:	Low
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> <li>1. System shall display a “Personality” button in the profile page.</li> <li>2. If the Adopter selects the “Personality” button, he will be prompted to fill in a personality form with required fields.</li> <li>3. The System checks for all required fields to be filled up.</li> <li>4. The form is sent to the respective Pet Owner for him to evaluate the Adopter’s suitability for adoption.</li> </ol>
Alternative Flows:	<p><u>AF-S3: Adopter did not fill up all required fields in the form</u></p> <ol style="list-style-type: none"> <li>1. Adopter did not fill up all required fields in the form.</li> <li>2. Adopter clicks the “Submit” button.</li> <li>3. System checks that all the required fields are not filled up.</li> </ol>

	4. The form is not sent to the Owner, the System prompts the Adopter to fill up all required fields in the form.
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	Adopter wishes to adopt a pet from a particular listing.
Notes and Issues:	None

## Functional Requirement #5

### V.I ReviewActivity

Use Case ID:	#5-1		
Use Case Name:	ReviewActivity		
Created By:	Benjamin Yeoh	Last Updated By:	Benjamin Yeoh
Date Created:	8/2/25	Date Last Updated:	8/2/25

Actor:	Administrator
Description:	Administrators will be brought to the administrator page where he/she can perform actions that can only be completed by administrators.
Preconditions:	Administrator is logged in and authenticated.
Postconditions:	Affected Users will be notified.
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. System displays the list of all of the reports submitted by users.</li> <li>2. Administrator selects an activity and views the details of the submitted report.</li> <li>3. System prompts the Administrator to select the desired activity: DeleteListing, BanUser, DismissReport.</li> <li>4. If the Administrator selects the activity DeleteListing, then the Administrator uses the included use case DeleteListing to delete the selected listing.</li> <li>5. If the Administrator selects the activity BanUser, then the Administrator uses the included use case BanUser to ban the selected user.</li> <li>6. If the Administrator selects the activity DismissReport, then the Administrator will be directed to the included use case DismissReport to delete the report.</li> </ol>

Alternative Flows:	None
Exceptions:	None
Includes:	<ol style="list-style-type: none"> <li>1. DeleteListing</li> <li>2. BanUser</li> <li>3. DismissReport</li> </ol>
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

## V.II BanUser

Use Case ID:	#5-2		
Use Case Name:	BanUser		
Created By:	Benjamin Yeoh	Last Updated By:	Benjamin Yeoh
Date Created:	8/2/25	Date Last Updated:	8/2/25

Actor:	Administrator
Description:	Administrators can ban Users deemed inappropriate.
Preconditions:	Administrator is logged in and authenticated.
Postconditions:	Banned Users will be notified.
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. System displays a “Ban User” button.</li> <li>2. Administrator clicks on the “Ban User” button.</li> <li>3. System shall display a “Confirm” and a “Cancel” button.</li> <li>4. Upon clicking the “Confirm” button, the Administrator bans the selected User.</li> <li>5. System disables the user’s account.</li> <li>6. The banned user is notified of the ban.</li> </ol>
Alternative Flows:	<p><u>AF-S3: Administrator clicks the “Cancel” button</u></p> <ol style="list-style-type: none"> <li>1. If the Administrator clicks the “Cancel” button, the System will return to step 1.</li> </ol>
Exceptions:	None
Includes:	None

Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

### V.III DeleteListing

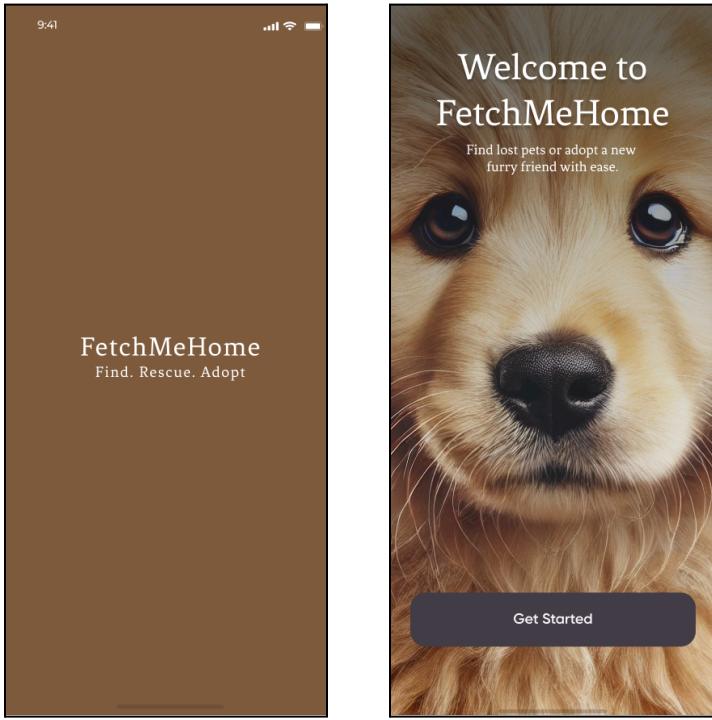
Use Case ID:	#5-3		
Use Case Name:	DeleteListing		
Created By:	Benjamin Yeoh	Last Updated By:	Benjamin Yeoh
Date Created:	8/2/25	Date Last Updated:	8/2/25

Actor:	Administrator
Description:	Administrator can remove listing that he/she deems inappropriate
Preconditions:	If the actor is an Administrator, the administrator must be logged in and verified.

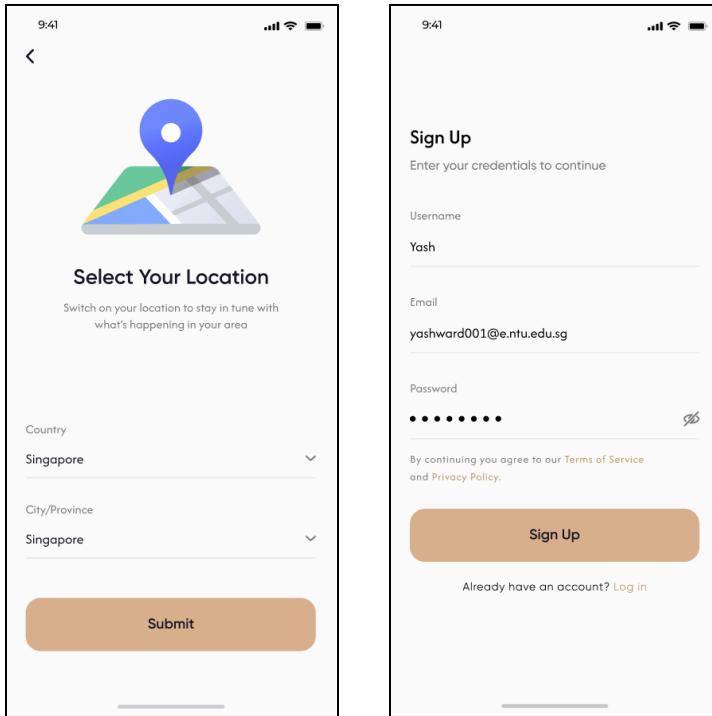
Postconditions:	Listing is deleted
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. User selects a the listing that he/she wants to delete</li> <li>2. User clicks the kebab menu and selects the “Delete Listing” icon.</li> <li>3. System prompts Administrator’s confirmation and deletes the listing.</li> <li>4. Owner of the listing is notified of the deletion.</li> </ol>
Alternative Flows:	<p><u>AF-S3: Administrator cancels the confirmation:</u></p> <ol style="list-style-type: none"> <li>1. Administrator selects the “Cancel” icon when prompted for deletion confirmation.</li> <li>2. Administrator is brought back to the interface list of submitted reports.</li> </ol>
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

## 4. UI MockUps

### 4.1 Introduction Page



### 4.2 Sign Up



## 4.3 Login

9:41

Mobile Number

Mobile Number  
+65 |

Or connect with social media

G Continue with Google

f Continue with Facebook

9:41

Enter your mobile number

Mobile Number  
SG +65 |

9:41

Enter your 4-digit code

Code  
| - - - |

Resend Code

1 2 3  
4 5 6  
7 8 9  
PQRS TUV WXYZ  
+ \* # 0 ☒

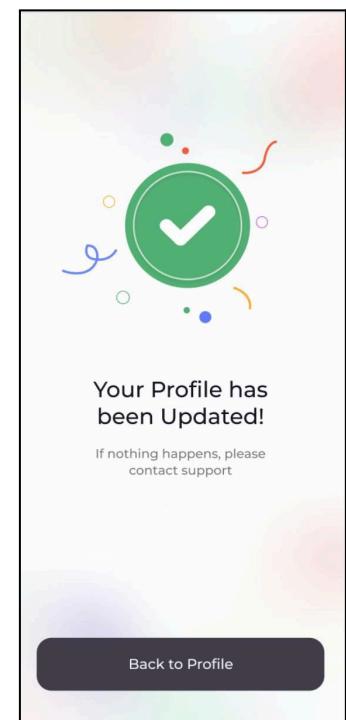
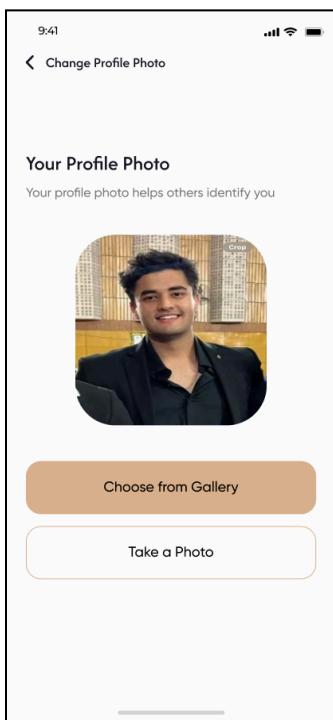
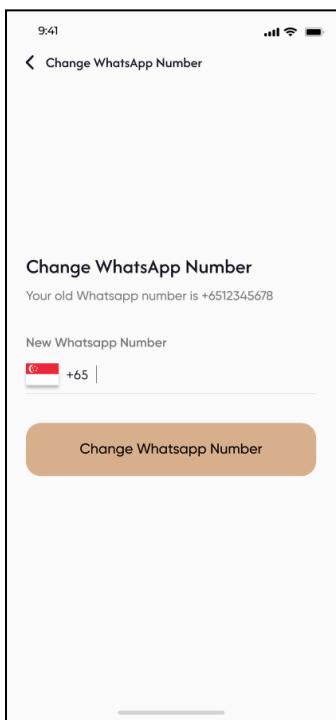
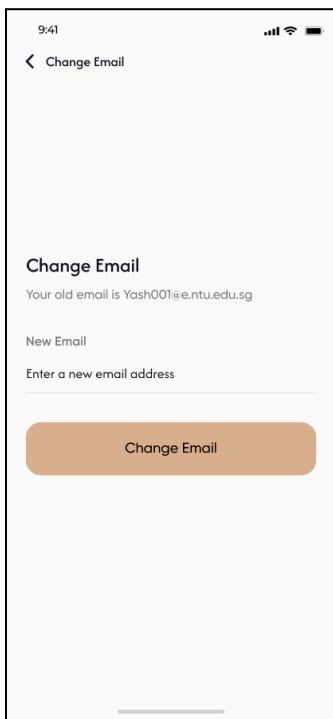
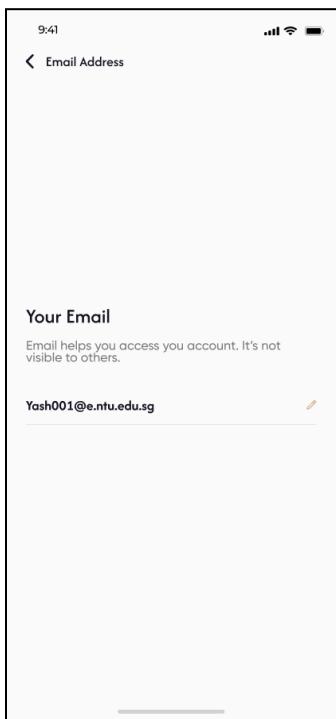
1 2 3  
4 5 6  
7 8 9  
PQRS TUV WXYZ  
+ \* # 0 ☒

## 4.4 Walkthrough

The image displays six screenshots of a mobile application interface, arranged in a 2x3 grid. The app appears to be a platform for users to report lost or found pets and for others to search for them.

- Screenshot 1: Lost Pets**  
Shows a list of lost pets. One listing for "Shiro" (Siberian Husky) is highlighted. A callout bubble says: "This is the lost pets page. You can find lost pets here." with a button "Ok, I understand." Below it, another listing for "Bob" (Golden Retriever) is shown. A callout bubble says: "This is the lost pets page. You can find lost pets here." with a button "Ok, I understand."
- Screenshot 2: Adopt a Pet**  
Shows a list of adoptable pets. One listing for "Buck" (Siberian Husky) is highlighted. A callout bubble says: "This is the adopt pets page. You can find adopt pets here." with a button "Ok, I understand." Below it, another listing for "Bob" (Rottweiler) is shown. A callout bubble says: "This is the adopt pets page. You can find adopt pets here." with a button "Ok, I understand."
- Screenshot 3: Lost Pets**  
Shows a detailed view of a lost pet, "Buck". A callout bubble says: "Here, you can see the details of the pet." with a button "Ok, I understand." Below the details, there are sections for "Owner Details" (Max, 30 Years Old), "Health Conditions" (Apples Are Nutritious. Apples May Be Good For Weight Loss. Apples May Be Good For Your Heart. As Part Of A Heartful And Varied Diet.), and "Owner Details" (Rating: 5 stars).
- Screenshot 4: Adopt a Pet**  
Shows a detailed view of an adoptable pet, "Bruno". A callout bubble says: "Here, you can see the details of the pet." with a button "Ok, I understand." Below the details, there are sections for "Last Seen" (Pasir Ris Public Park, 12:09 PM, Near Central Pond), "Special Traits" (Rating: 5 stars), and "Owner Details" (Rating: 5 stars).
- Screenshot 5: Create Listing**  
Shows the process of creating a new listing. It includes fields for "Name" (Nan), "Breed" (Golden Retriever), "Description", and "Health Conditions". A callout bubble says: "This is the page to create a new Listing." with a button "Ok, I understand."
- Screenshot 6: User Details**  
Shows user profile details for "Yash". It includes a profile picture, a rating of 5 stars, and a message: "Hello! I am a new user!". A callout bubble says: "Here, you can rate, restrict, block, or Flag listings" with a button "Ok, I understand." Below the details, there are sections for "Restrict", "Block", and "Flag".

## 4.5 Account settings



## 4.6 Welcome to FetchMeHome / Personality Quiz

The image displays three sequential screenshots from a mobile application titled "FetchMeHome".

**Screenshot 1: Welcome Screen**  
The screen features a colorful, abstract background with a central green circle containing a white checkmark. Below the circle, the text "Welcome to Fetch Me Home!" is displayed in a bold, black font. A smaller text below it reads: "Take a personality quiz to find out which dog is best for you!". At the bottom, there are two buttons: a dark blue button labeled "Take a Personality Quiz" and a light blue button labeled "No thanks".

**Screenshot 2: Personality Test Screen**  
The screen shows the title "Personality Test" at the top, followed by the instruction "Rate these statements from 1-5". Below this, there are five identical statement blocks, each consisting of a statement and a rating scale from "Big no" to "Big yes".

- I like hanging out with new people

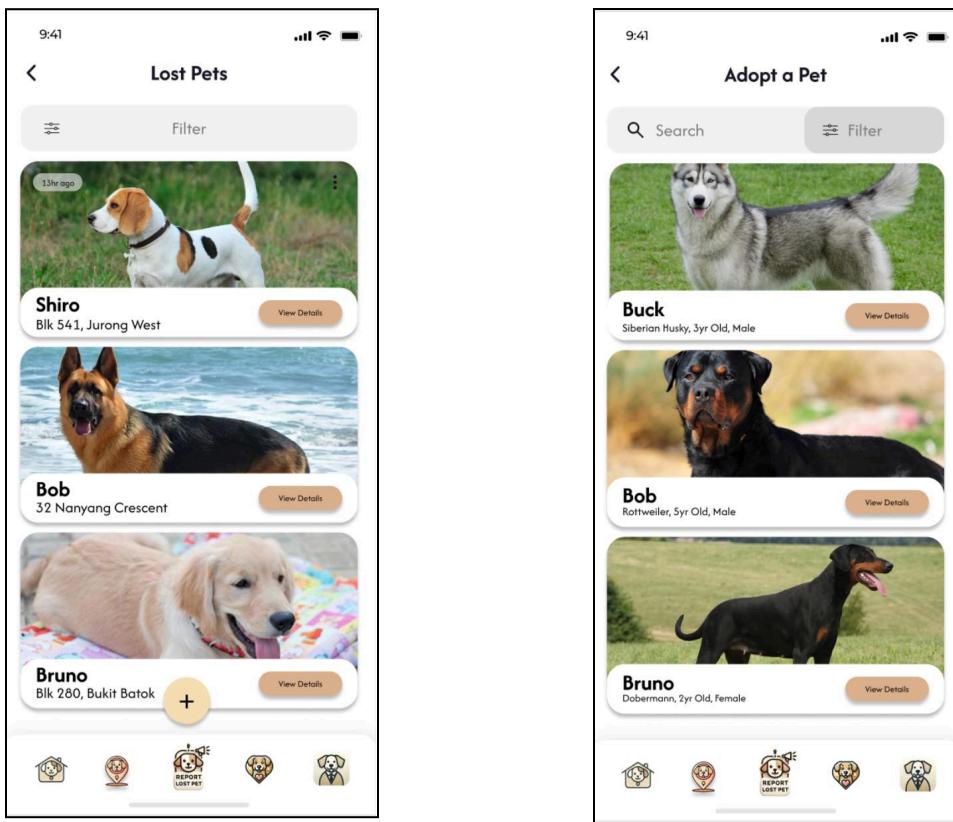
At the bottom of the screen are two buttons: a large orange "Submit" button and a smaller "Skip this section" button.

**Screenshot 3: Result Screen**  
The screen shows the title "Personality Test" at the top, followed by the instruction "Rate these statements from 1-5". Below this, the text "You are a Dog Person! 🐕" is displayed. Further down, there are four sections corresponding to the letter grades:

- Mostly A's - You Are A Dog!  
Loyal, Energetic, And Social, You Thrive On Companionship And Adventure. You Need A Pet That Can Keep Up With Your Active Lifestyle And Return Your Affectionate Nature.
- Mostly B's - You Are A Cat!  
Independent Yet Affectionate On Your Terms, You Enjoy Cozying Up At Home But Also Need Your Space. A Low-Maintenance But Loving Pet Like A Cat Is Perfect For You.
- Mostly C's - You Are A Bird!  
You Enjoy Peaceful Environments, Value Independence, And Love A Good Routine. A Bird's Calming Presence And Ability To Entertain With Song Make It A Great Match.
- Mostly D's - You Are A Reptile!  
Unique, Curious, And Often Unpredictable, You Enjoy Observing And Learning Rather Than Constant Socializing. A Reptile Like A Gecko Or Snake Would Fit Your Chill Yet Fascinating Personality.

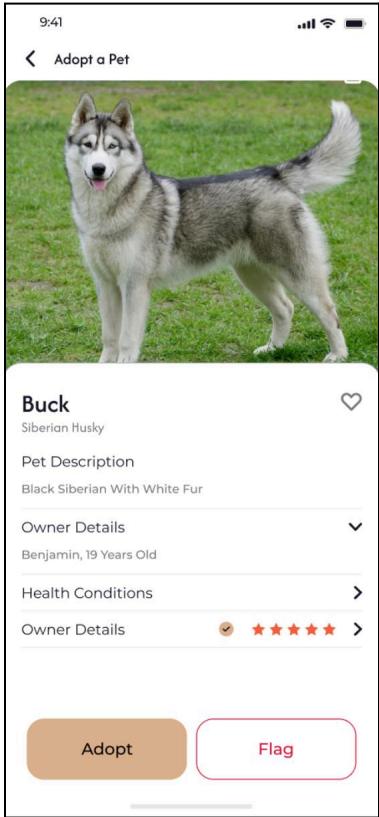
At the bottom of the screen is a large orange "Save Results" button.

#### 4.7. View Pets up for Adoption / Missing Pets

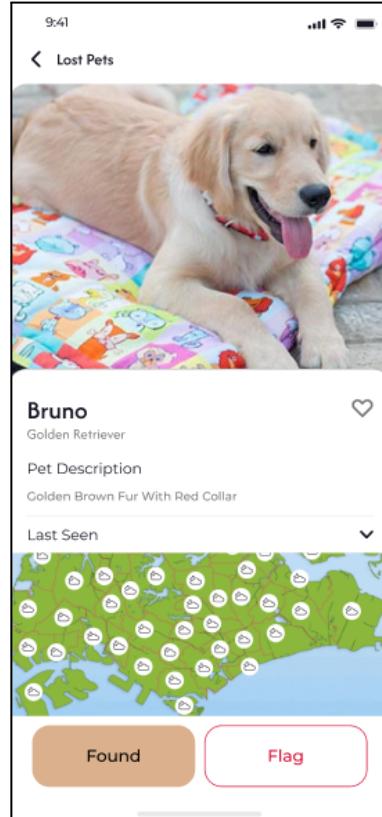


#### 4.8 View details of listings

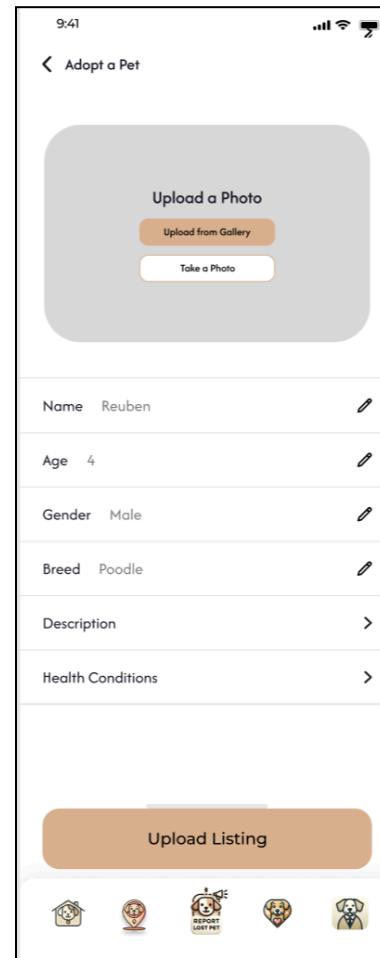
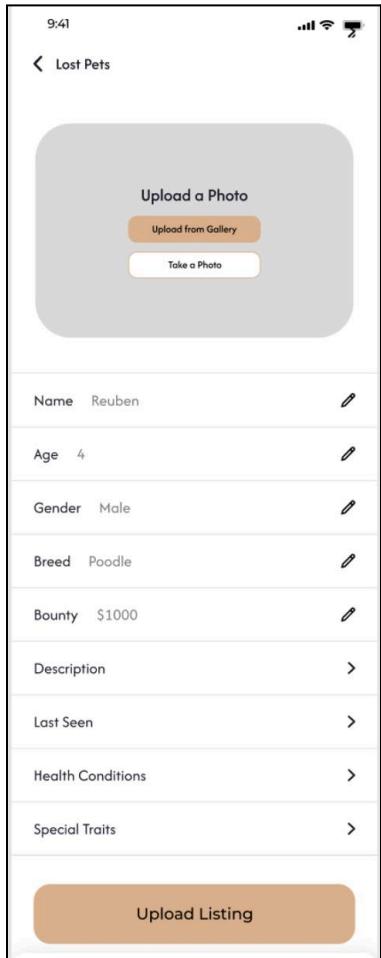
##### AdoptPetListing



##### LostPetListing



## 4.9 Create Listing for Lost Pets and Pets that are up for Adoption



## 4.10 Edit Listing for Lost Pets and Missing Pets

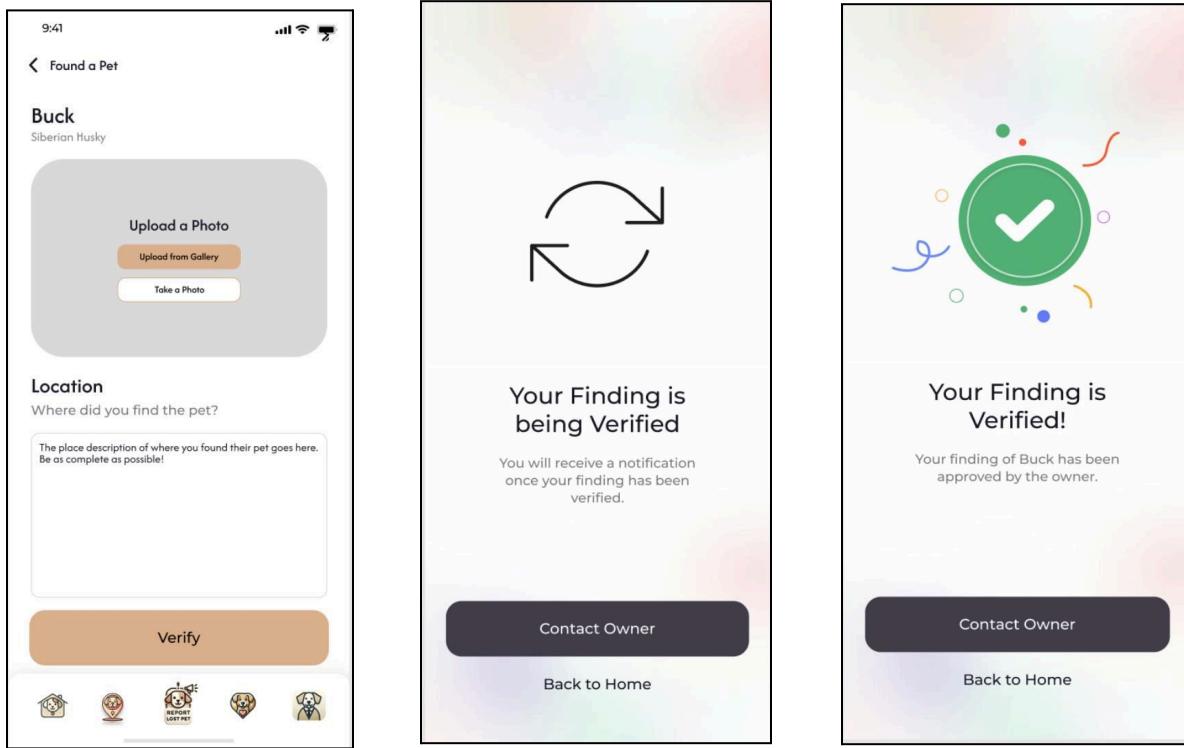
The image displays five wireframe screens for a mobile application, arranged in two rows. The top row contains three screens: 'Pet Description', 'Last Seen', and 'Health Conditions'. The bottom row contains two screens: 'Special Traits' and a summary screen.

- Pet Description:** Shows a text input field for pet description with placeholder text 'Your pet description goes here. Be as complete as possible!'. Includes 'Update Pet Description' and 'Back to Previous Page' buttons.
- Last Seen:** Shows a text input field for last seen location with placeholder text 'Your pet's last known location goes here. Be as complete as possible!'. Includes 'Update Last Seen' and 'Back to Previous Page' buttons.
- Health Conditions:** Shows a text input field for health conditions with placeholder text 'Your pet's health conditions goes here. Be as complete as possible!'. Includes 'Update Health Conditions' and 'Back to Previous Page' buttons.
- Special Traits:** Shows a text input field for special traits with placeholder text 'Your pet's special traits goes here. Be as complete as possible!'. Includes 'Update Special Traits' and 'Back to Previous Page' buttons.
- Summary Screen:** Displays a green circular icon with a checkmark and the message 'Your Listing has been Updated'. It includes 'Back to Home' and 'Undo Changes' buttons, along with a note: 'If nothing happens, please contact support'.

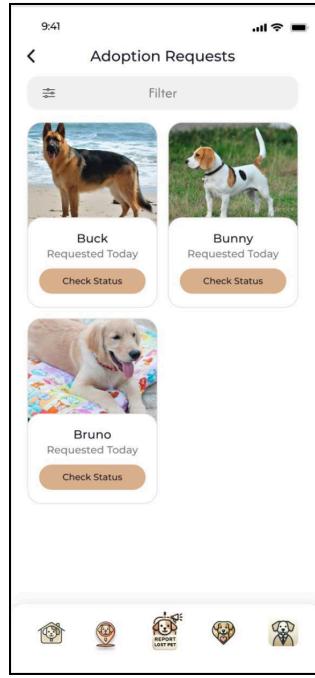
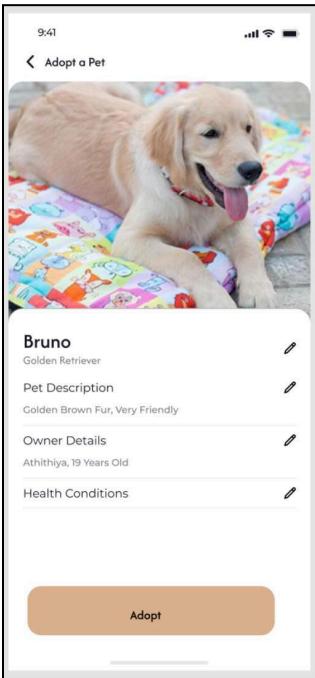
## 4.11 Delete Listing



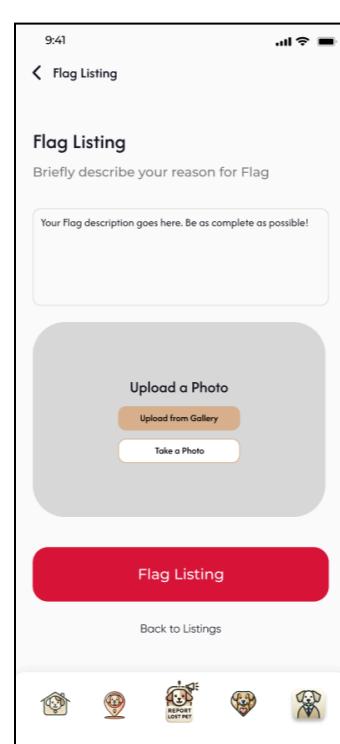
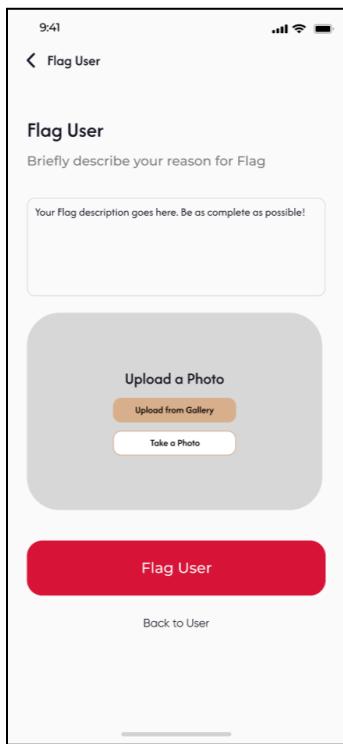
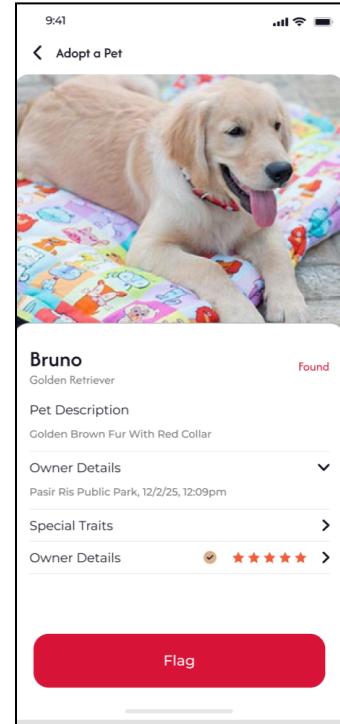
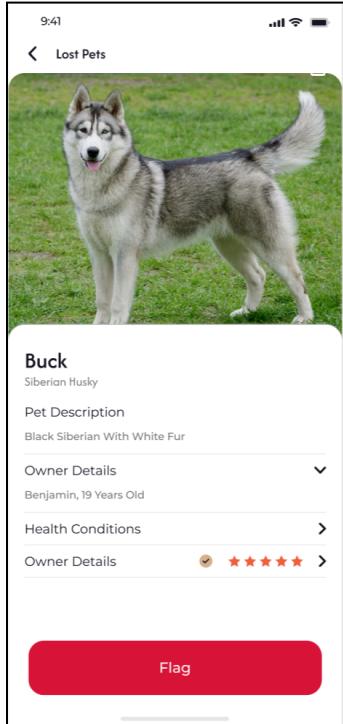
#### 4.12 PetFinders can Submit Lost Report for Missing Pets



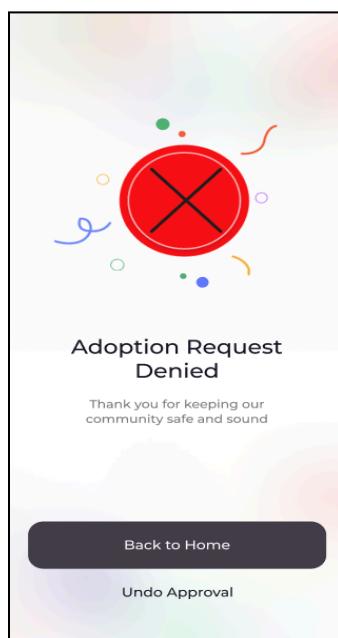
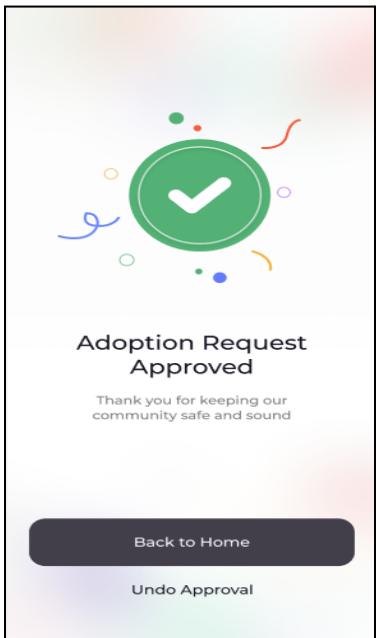
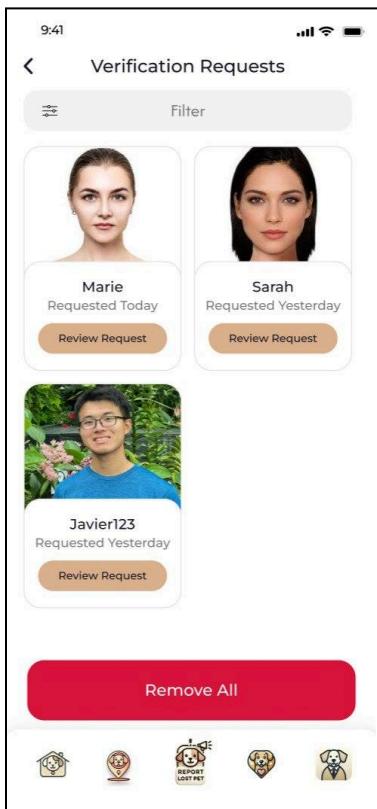
#### 4.13 Adopters can submit adoption request and check their status of request



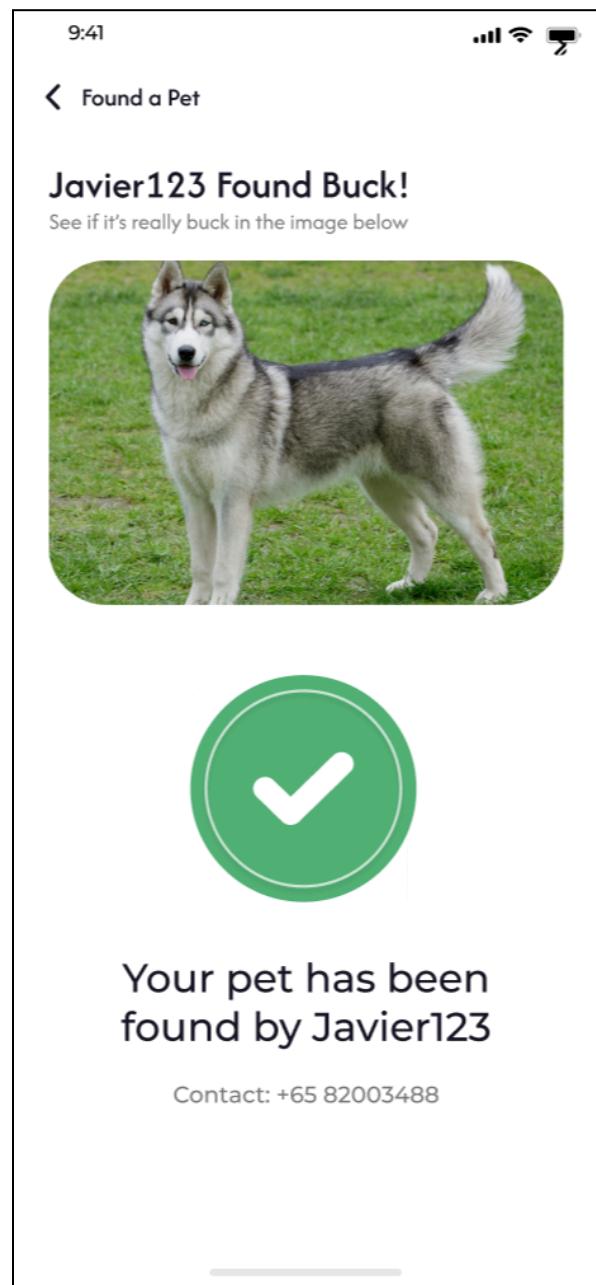
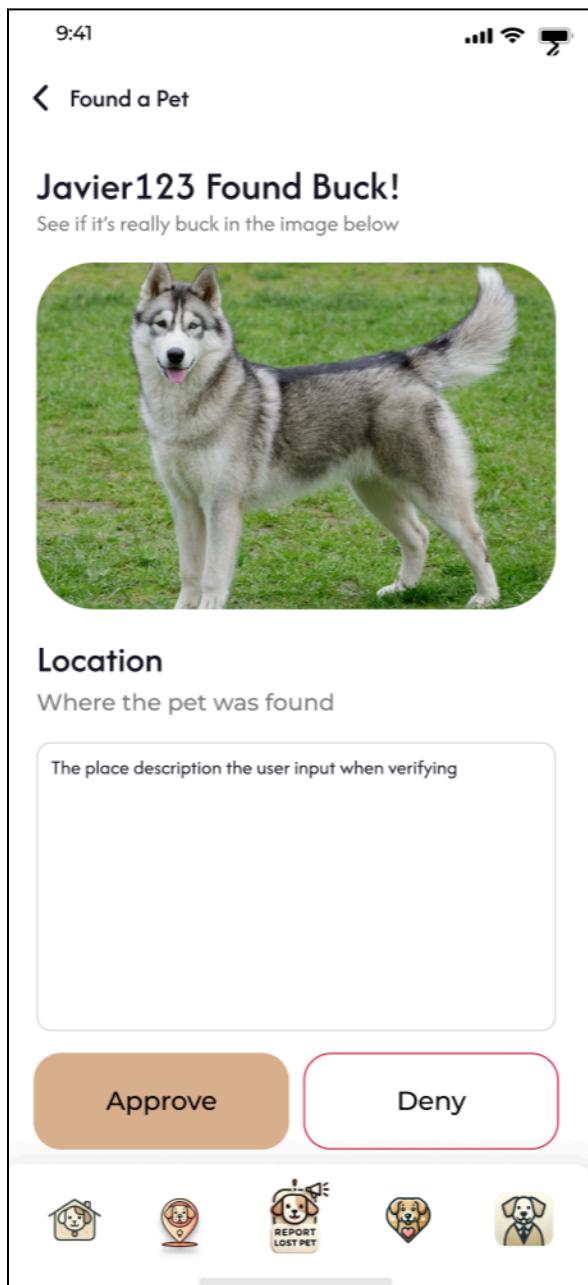
## 4.14 Adopters and Pet Finders can flag adoption and missing pet listings as well Owners



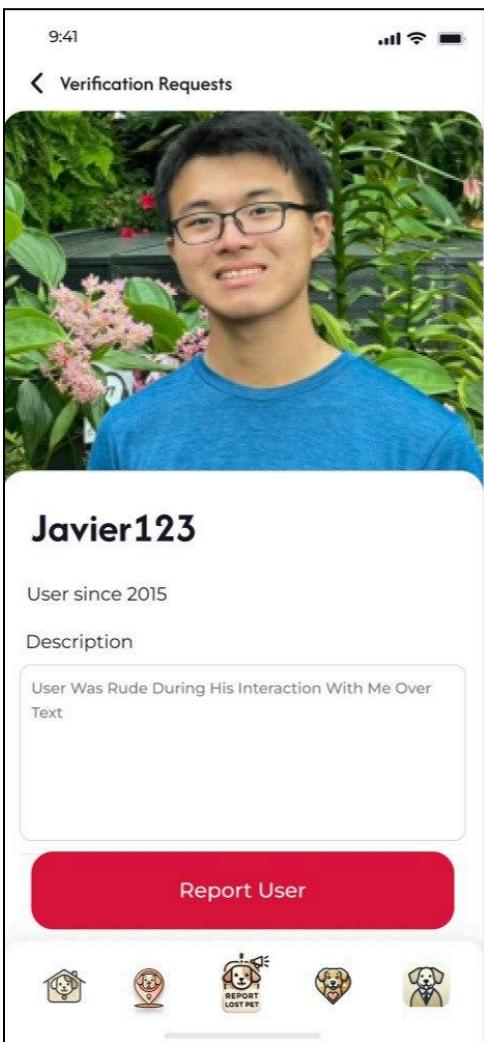
#### 4.15 Owners can view adoption requests, as well as approve or deny them



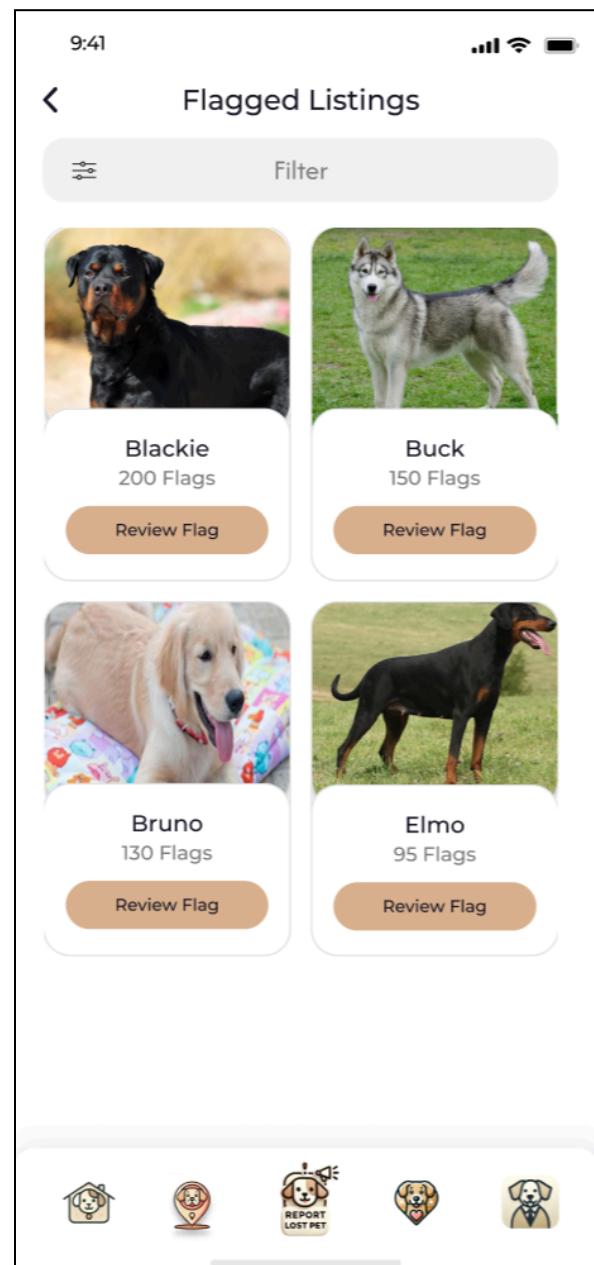
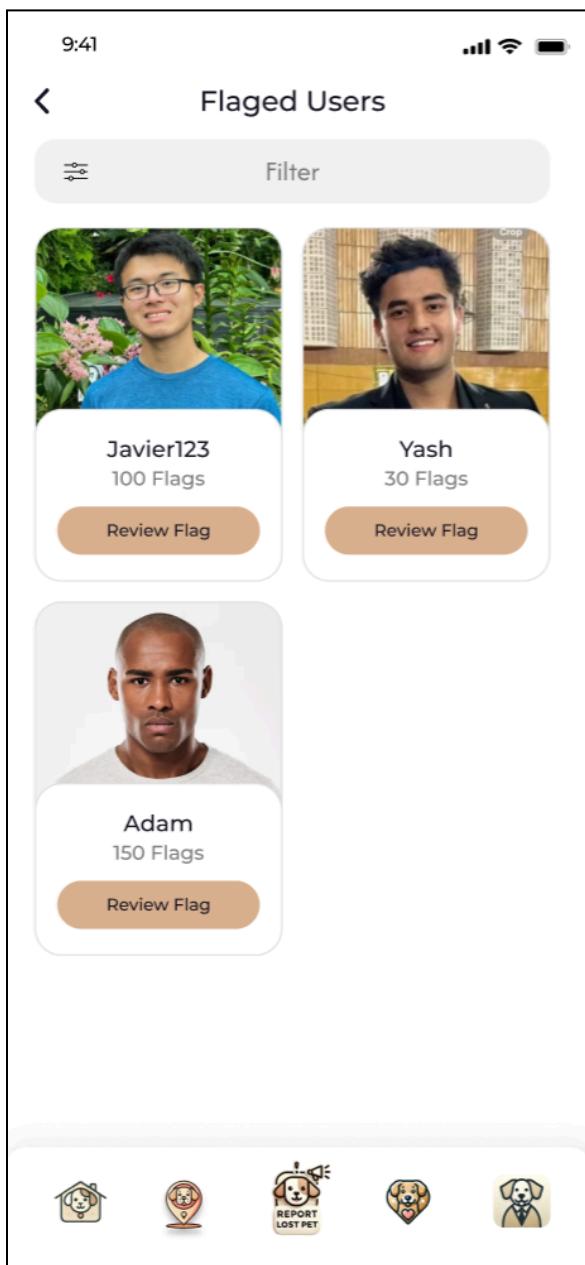
4.16 Owners can approve lost pet submission reports, as well as approve or deny them



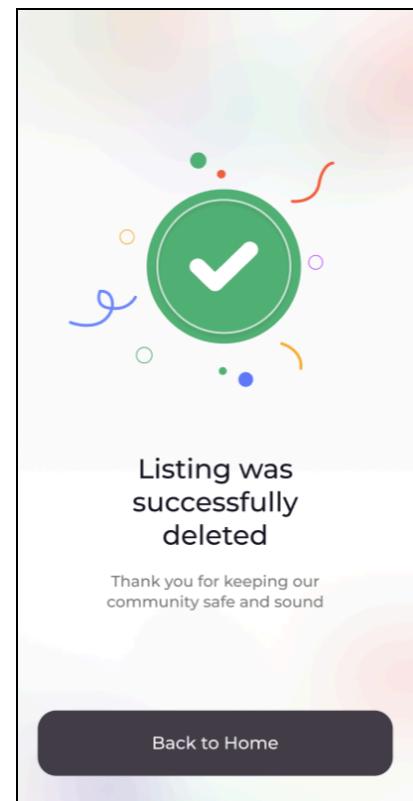
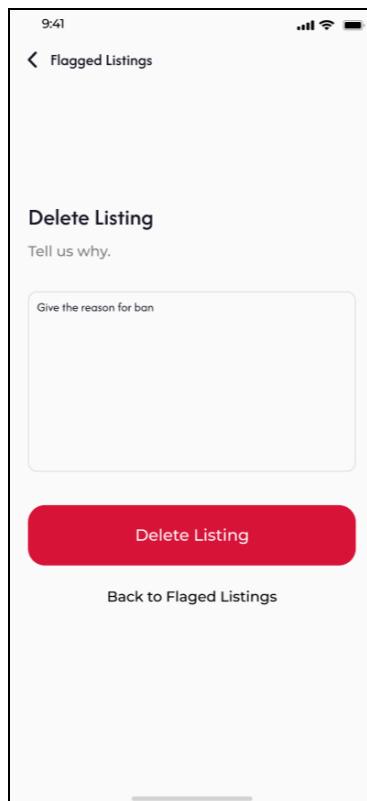
#### 4.17 Owner reporting Adopter / PetFinder for misconduct



#### 4.18 Administrators can view reports submitted by Users



#### 4.19 Administrator can remove flag or remove the flagged listing



#### 4.20 Administrator can BanUser or Dismiss Report

