

**NANYANG
TECHNOLOGICAL
UNIVERSITY**
SINGAPORE

SC2006 - Software Engineering
Lab 1 Deliverables

| | |
|------------------|-----------------------------------------|
| Lab Group | SCMA |
| Team | FetchMeHome |
| Members | 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

| | |
|----------------------------------------------------------------------------------------------|-----------|
| 1. Documentation of Functional & non-functional requirements | 3 |
| A. Functional Requirements | 3 |
| B. Non-Functional Requirements | 8 |
| 2. Data Dictionary | 10 |
| 3. Initial Use Case Model, consisting of Use Case diagram & Use Case descriptions | 12 |
| A. Use Case Diagram | 12 |
| B. Use Case Descriptions | 13 |
| 4. UI Mockups | 70 |

1. Documentation of functional and non-functional requirements

A. Functional Requirements

1. FetchMeHome shall allow Users to be authenticated and use the website.
 - 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 log 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 to protect the User's privacy.
 - 1.2.3. FetchMeHome shall verify if the E-Mail and the password keyed in by the User are valid before letting them log in.
 - 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 names.
 - 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 perform general tasks without a user account.
 - 2.1. Users must be able to view lost pet listings.
 - 2.1.1. FetchMeHome shall display pet details for each lost pet listing.
 - 2.1.2. FetchMeHome shall display a real-time map which shows the last seen location of the pet.
 - 2.1.3. FetchMeHome shall allow Users to filter lost pet listings based on the type of pet and the last seen location of the lost pet.
 - 2.1.4. FetchMeHome shall allow Users to report inappropriate lost pet listings.
 - 2.2. Users must be able to view pet adoption listings.
 - 2.2.1. FetchMeHome shall display pet details for each adoption listing
 - 2.2.2. FetchMeHome shall allow Users to filter pet adoption listings based on the type of pet.

- 2.2.3. FetchMeHome shall allow Users to report inappropriate pet adoption listings.
 - 2.3. Users must be able to view the Frequently Asked Questions (FAQ) page.
3. FetchMeHome shall allow Owners to perform Owner specific tasks
 - 3.1. FetchMeHome shall allow Owners to create new lost pet listings.
 - 3.1.1. Owners can upload images of the lost pet.
 - 3.1.2. Owners can enter the name of the lost pet.
 - 3.1.3. Owners can enter the age of the lost pet.
 - 3.1.4. Owners can enter the type of the lost pet.
 - 3.1.5. Owners can enter the last seen location of the lost pet.
 - 3.1.6. Owners can enter a short description of the lost pet.
 - 3.1.7. Owners can enter the bounty reward for the lost pet.
 - 3.2. FetchMeHome shall allow Owners to interact with PetFinders
 - 3.2.1. FetchMeHome shall allow Owners to verify images submitted by PetFinders.
 - 3.2.2. FetchMeHome shall allow Owners to flag PetFinders for misconduct.
 - 3.3. FetchMeHome shall allow Owners to create new adoption pet listings.
 - 3.3.1. Owners must upload images of the pet for adoption.
 - 3.3.2. Owners must enter the name of the pet for adoption.
 - 3.3.3. Owners must enter the age of the pet for adoption.
 - 3.3.4. Owners must enter the type of the pet for adoption.
 - 3.3.5. Owners must enter a short description of the pet for adoption.

3.4. FetchMeHome shall allow Owners to edit details of their existing listings.

3.4.1. FetchMeHome shall allow Owners to upload new images and/or edit any of their desired attributes.

3.5. FetchMeHome shall allow Owners to interact with Adopters.

3.5.1. FetchMeHome shall allow Owners to review Adopters' adoption requests.

3.5.2. FetchMeHome shall allow Owners to contact Adopters to liaise meeting details.

3.5.3. FetchMeHome shall allow Owners to flag Adopters for misconduct.

4. FetchMeHome shall allow PetFinders to perform PetFinder-specific tasks.

4.1. FetchMeHome shall allow PetFinders to report found pets.

4.2. FetchMeHome shall allow PetFinders to flag inappropriate Owners.

5. FetchMeHome shall allow Adopters to perform Adopter-specific functions.

5.1. FetchMeHome shall allow Adopters to submit adoption requests for their desired pet.

5.1.1. Adopters must enter their email.

5.1.2. Adopters must enter their phone number.

5.1.3. Adopters must state their current housing situation.

5.1.4. Adopters must fill in their past history with pets.

5.1.5. Adopters must fill in their current list of pets.

5.2. FetchMeHome shall allow Adopters to flag Owners for misconduct.

5.2.1. FetchMeHome shall allow the Adopter to enter a short description of the situation.

6. FetchMeHome shall allow Administrators to perform Administrator specific functions.

6.1. FetchMeHome shall allow Administrators to review reports made.

6.1.1. FetchMeHome shall allow Administrators to ban the Users from the platform.

6.1.2. FetchMeHome shall allow Administrators to remove listings from the platform.

6.1.3. FetchMeHome shall allow Administrators to dismiss the report.

B. Non-Functional Requirements

| | |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Usability | <ul style="list-style-type: none">• 90% of users should be able to login in 5 min.• 90% of users should be able to navigate the 2 different sub categories and home page. |
| Reliability | <ul style="list-style-type: none">• The system should have at least 95% uptime to ensure users can always search for lost pets.• Launching the app should be available in 5 seconds. |
| Performance | <ul style="list-style-type: none">• The system should process the Users preferences and personality test results to return the most suitable pet to them in 20 seconds.• The system should login in the user within 10 seconds.• The system should display the list of lost pets and pets for adoption in 5 seconds. |
| Maintainability | <ul style="list-style-type: none">• Code should follow best practices (SOLID, clean architecture). |
| Scalability | <ul style="list-style-type: none">• The platform should support at least 1000 concurrent users |

| | |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>without performance degradation.</p> <ul style="list-style-type: none">• But if needed, scaling horizontally by increasing the quantity of servers to handle more traffic should be available. |
| Security | <ul style="list-style-type: none">• User authentication must use hashed and salted passwords (bcrypt/Argon2/PBKDF2).• Prevent unauthorized access (e.g., Users should not have access to administrator privileges). |

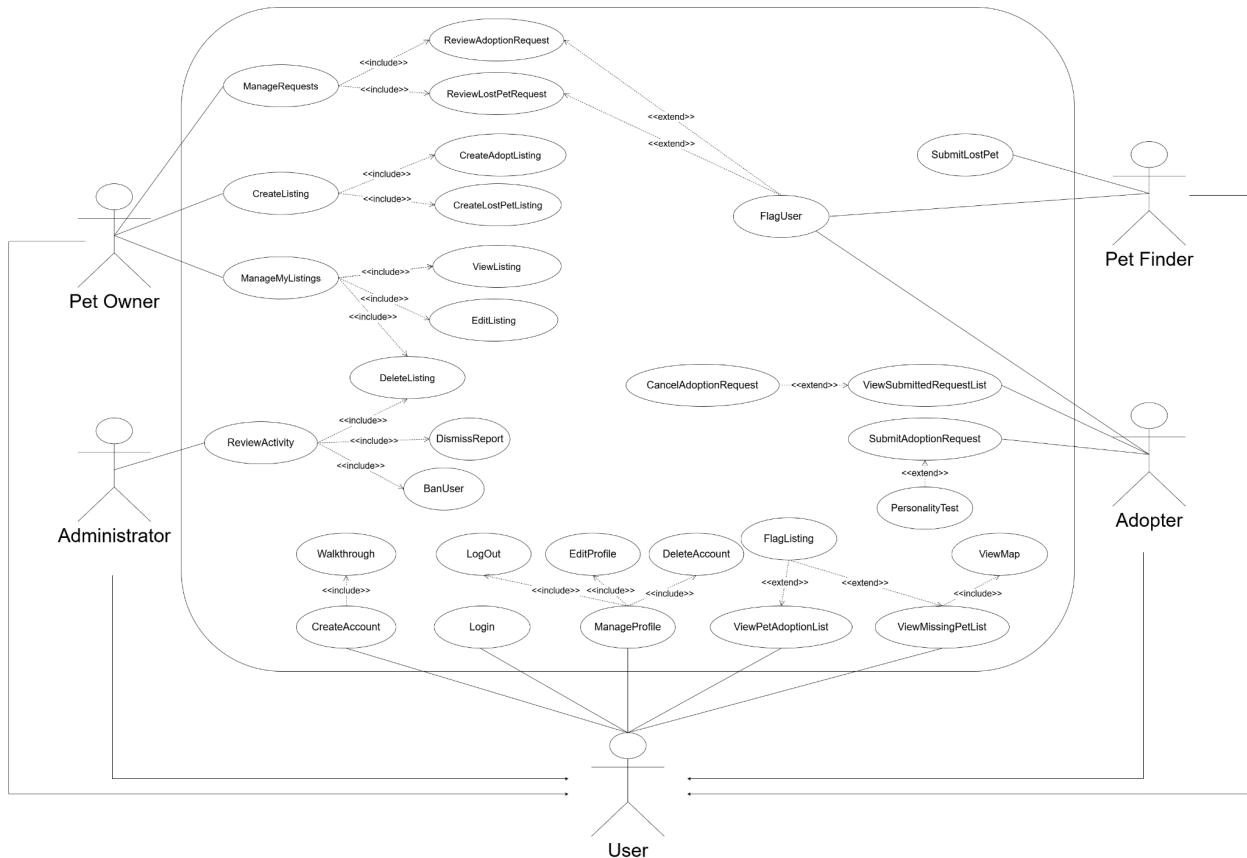
2. Data Dictionary

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

| | |
|-------------------------|-----------------------------------------------------------------------------------------------------------------|
| Adopter | A person who wishes to adopt a pet from the listings. |
| Administrator | An administrator who manages the platform and handles misconduct reports and listings. |
| Personality Test | A series of questions to determine which breed of pets are suitable for the user. |
| Walkthrough | A one time flow of events that guide the 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"> 1. System will prompt the user to enter their Full Name, E-mail Address, Phone Number, Password, and Retype Password. 2. System checks all the required fields. 3. User selects “Validate Phone Number”. 4. System generates a One-Time-Password (OTP) and sends the OTP to the User’s entered Phone Number. 5. User enters the OTP. 6. An Account is created for the User and the database is updated. |
| 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"> 1. User did not fill in all required fields in the form. 2. Form is not submitted, the system prompts User to fill in all required fields. 3. System returns to step 1. <p><u>AF-S3: User phone number is already registered</u></p> <ol style="list-style-type: none"> 1. User phone number is already registered with the app. 2. System will prompt the User to login using the login use case. <p><u>AF-S4: User OTP entered is incorrect</u></p> <ol style="list-style-type: none"> 1. User OTP entered is incorrect. 2. System will prompt the User to enter the correct OTP. 3. System returns to step 4. |
| 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"> 1. System prompts the User to enter their phone number and password. 2. The User enters their Phone number and Password into the respective fields. 3. The User selects “Login”. 4. System checks all the required fields. 5. The User is navigated to the home screen of the FetchMeHome application. |
| Alternative Flows: | <u>AF-S4: User did not fill in all required fields in the page</u> <ol style="list-style-type: none"> 4. User did not fill in all required fields in the page. 5. Login is not processed, the system prompts User to fill in all required fields. 6. System returns to step 1. |
| 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 WalkThrough

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

| | | | |
|----------------|-----------------------------|--------------------|-----------------------------|
| 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"> 1. The User will click the “Login” button after creating a FetchMeHome account. 2. Beside the icon of the AdoptAPet function, the System shall display a description of how to navigate it. 3. The User shall then press the “I Understand” button. 4. Beside the icon of the FindLostPet function, the System shall display a description of how to navigate it. |

| | |
|-----------------------|---------------------------------------------------------|
| | 5. The User shall then press the “I Understand” button. |
| Alternative Flows: | None |
| Exceptions: | None |
| Includes: | None |
| Special Requirements: | None |
| Assumptions: | None |
| Notes and Issues: | None |

I.IV ViewMissingPetList

| | | | |
|----------------|-----------------------------|--------------------|-----------------------------|
| Use Case ID: | #1-4 | | |
| 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"> 1. The User shall click on the “LostPet” icon on the navigation bar of the home screen of the FetchMeHome application. 2. The System shall then display all the posts of Lost Pets. 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.I ViewPetAdoptionList

| | | | |
|----------------|-----------------------------|--------------------|-----------------------------|
| Use Case ID: | #1-5 | | |
| 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"> 1. User shall click the “AdoptPet” icon on the navigation bar of the home screen of the FetchMeHome application. |

| | |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>2. The System shall then display all listings of Pets that are up for adoption.</p> <p>3. Users can click on any individual post to expand it and view more details of the pet that is up for adoption.</p> |
| Alternative Flows: | None |
| Exceptions: | None |
| Includes: | None |
| Special Requirements: | None |
| Assumptions: | None |
| Notes and Issues: | None |

I.VI ManageProfile

| | | | |
|----------------|-----------------------------|--------------------|-----------------------------|
| Use Case ID: | #1-6 | | |
| 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"> 1. The User will click on the “Profile” icon on the navigation bar of the FetchMeHome home page 2. The System shall then display the profile page of the User. 3. The System shall then display the initial details provided by the User during Account creation. 4. The System shall display the “Edit Profile” button and the “Delete Account” buttons at the bottom of the profile. 5. If the User clicks on the “Edit Profile” button then they will use the included use case EditProfile to edit their profile. 6. If the User clicks on the “Delete Account” button then they will use the included use case DeleteAccount to delete their profile. 7. If the User clicks on the “Log Out” button then they will use the included use case LogOut to log out of their account. |
| Alternative Flows: | None |

| | |
|-----------------------|------------------------------------|
| Exceptions: | None |
| Includes: | EditProfile, DeleteAccount, LogOut |
| Special Requirements: | None |
| Assumptions: | None |
| Notes and Issues: | None |

I.VII EditProfile

| | | | |
|----------------|-----------------------------|--------------------|-----------------------------|
| Use Case ID: | #1-7 | | |
| 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"> 1. The User must have an existing FetchMeHome application 2. The User must be logged in to the FetchMeHome application |
| 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"> 1. The User will click the “Edit Profile” button on their profile. 2. The system shall prompt the User to key in the updated personal particulars. 3. The User shall then click the “Confirm” button to save the changes made to their profile. |
| Alternative Flows: | None |
| Exceptions: | None |
| Includes: | None |
| Special Requirements: | None |
| Assumptions: | The User has an existing FetchMeHome account |
| 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"> 1. The User will click on the “Delete Account” button 2. System shall then display the “Confirm” and “Cancel” buttons 3. If the User clicks on the “Confirm” button the Account will be deleted |

| | |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>4. The User will be transported back to their own Profile page</p> |
| Alternative Flows: | None |
| Exceptions: | <p><u>AF-S2: User clicks the “Cancel” button</u></p> <p>1. If the User clicks the “Cancel” button, the system will return to the profile page.</p> |
| Includes: | None |
| Special Requirements: | None |
| Assumptions: | None |
| Notes and Issues: | None |

I.IX LogOut

| | | | |
|----------------|-----------------------------|------------------|-----------------------------|
| Use Case ID: | #1-9 | | |
| 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"> 1. The System shall display a “Log Out” button in the profile of the User. 2. The User shall then click the “Log Out” button. 3. The System shall then display the “Confirm” and “Cancel” buttons. 4. The User shall then click the “Confirm” button. 5. The User shall then be logged out of the FetchMeHome application. 6. The System shall then display the Login page of the FetchMeHome application. |
| Alternative Flows: | <u>AF-S3: The User chooses to click the “Cancel” button instead</u> <ol style="list-style-type: none"> 1. The User shall click on the “Cancel” button. |

| | |
|-----------------------|-------------------------------------------------------------|
| | 2. The System shall then return to the profile of the User. |
| Exceptions: | None |
| Includes: | None |
| Special Requirements: | None |
| Assumptions: | None |
| Notes and Issues: | None |

I.X FlagListing

| | | | |
|----------------|---------------|--------------------|---------------|
| Use Case ID: | #1-10 | | |
| 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"> 1. The System shall display a “FlagListing” button. 2. If the User deems the listing as inappropriate, the User can click the “FlagListing” button. 3. The System will prompt the User to fill in details of the scenario in a short description box. 4. Checks are made to ensure the description box is filled. 5. User clicks the “Submit” button, a notification is sent to the Administrators and the database is updated. |
| Alternative Flows: | <p><u>AF-S4: User did not fill up required field in the form</u></p> <ol style="list-style-type: none"> 1. User did not fill up the required field in the form. 2. User clicks the “Submit” button. 3. Checks that the required field is not filled up. 4. User is prompted to fill in the required field, no changes to the form. 5. System returns to step 3. |
| 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"> 1. The User shall click on the “Last Seen” button on the missing pet listing. 2. The System shall then display the local map. 3. The map shall include the last seen location of the missing pet which has been provided by the Pet Owner. 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 ManageRequests

| | | | |
|----------------|----------------|--------------------|---------------|
| Use Case ID: | #2-1 | | |
| Use Case Name: | ManageRequests | | |
| 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"> 1. System shall display a “ManageRequests” button in the profile page. 2. Upon clicking the button, the Owner will be provided with 2 use cases: <ol style="list-style-type: none"> a. ReviewAdoptionRequest b. ReviewLostPetRequest 3. If the Owner selects the activity ReviewAdoptionRequest, then the Owner uses the included use case ReviewAdoptionRequest to review adoption request(s) made by Adopter(s). 4. If the Owner selects the activity ReviewLostPetRequest, then the Owner uses the included use case ReviewLostPetRequest to review missing pet report(s) made by PetFinder(s). |
| Alternative Flows: | None |
| Exceptions: | None |
| Includes: | <ol style="list-style-type: none"> 1. ReviewAdoptionRequest 2. ReviewLostPetRequest |
| Special Requirements: | None |
| Assumptions: | None |
| Notes and Issues: | None |

II.II 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"> 1. System shall display a list of all Adopter(s) adoption requests with their personality test results attached as an image. 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. 3. Upon clicking the “Accept” button for a particular request, the System will provide the Owner with the respective Adopter’s Phone No. |
| Alternative Flows: | <u>AF-S2: Owner selects the “Decline” button</u> <ol style="list-style-type: none"> 1. If the Owner selects the “Decline” button, the System will delete the specific request. 2. System returns to step 1. |
| Exceptions: | None |
| Includes: | None |
| Special Requirements: | None |
| Assumptions: | None |
| Notes and Issues: | None |

II.III ReviewLostPetRequest

| | | | |
|----------------|----------------------|--------------------|---------------|
| Use Case ID: | #2-3 | | |
| Use Case Name: | ReviewLostPetRequest | | |
| 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 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"> System shall display a list of all lost pet reports from PetFinder(s). |

| | |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <ol style="list-style-type: none"> 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. 3. Owner shall determine if any of the images uploaded is his pet. 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. |
| Alternative Flows: | <u>AF-S2: Owner selects the “Decline” button</u> <ol style="list-style-type: none"> 1. If the Owner selects the “Decline” button, the System will delete the specific report. 2. System returns to step 1. |
| 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"> 1. System shall display an “Add” icon in the home page. 2. Upon clicking the “Add” icon, Owner will be provided with 2 use cases: <ol style="list-style-type: none"> a. CreateAdoptListing b. CreateLostPetListing |

| | |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>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.</p> <p>4. If the Owner selects the activity CreateLostPetListing, then the Owner uses the included use case CreateLostPetListing to create a listing for their missing pet(s).</p> |
| Alternative Flows: | None |
| Exceptions: | None |
| Includes: | <ol style="list-style-type: none"> 1. CreateAdoptListing 2. CreateLostPetListing |
| 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"> 1. Owner fills in all the required forms presented to them. 2. Owner clicks the “Submit” button, CreateAdoptListing is called. 3. Checks are made to ensure all the required fields are filled up. 4. Listing is created and added to the list of pets up for adoption. |
| 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"> 1. Owner did not fill up all required fields in the form. 2. Owner clicks the “Submit” button. |

| | |
|-----------------------|------------------------------------------------------------------------------------------------------------------------|
| | <p>3. The System flags out that some of the required fields are not filled up.</p> <p>4. System returns to step 1.</p> |
| Exceptions: | None |
| Includes: | None |
| Special Requirements: | None |
| Assumptions: | None |
| Notes and Issues: | None |

II.VI CreateLostPetListing

| | | | |
|----------------|----------------------|--------------------|---------------|
| Use Case ID: | #2-6 | | |
| Use Case Name: | CreateLostPetListing | | |
| 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"> 1. Owner fills in all the required forms presented to them. 2. Owner clicks the “Submit” button, CreateLostPetListing is called. 3. Checks are made to ensure all the required fields are filled up. 4. Listing is sent and updated to the database. |
| 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"> 1. Owner did not fill up all required fields in the form. 2. Owner clicks the “Submit” button, CreateLostPetListing is called. 3. Checks that some required fields are not filled up. 4. The System flags out that some of the required fields are not filled up. 5. System returns to step 1. |
| 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"> 1. System shall display a “MyPanel” button in the profile page. 2. Upon clicking the “MyPanel” button, the system shall display all available listing(s) which the Owner had uploaded. 3. The System shall provide the Owner with 2 use cases for each listing: <ol style="list-style-type: none"> a. Edit b. Delete 4. If the Owner selects the activity Edit, he will use the included use case Edit to edit his listing. 5. If the Owner selects the activity Delete, he will use the included use case Delete to delete his listing. |
| Alternative Flows: | None |
| Exceptions: | None |
| Includes: | <ol style="list-style-type: none"> 1. ViewListing 2. EditLising 3. DeleteListing |
| Special Requirements: | None |

| | |
|-------------------|------------------------------|
| Assumptions: | Owner has existing listings. |
| Notes and Issues: | None |

II.VIII ViewListing

| | | | |
|----------------|---------------|--------------------|---------------|
| Use Case ID: | #2-8 | | |
| Use Case Name: | ViewListing | | |
| 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 view 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"> 1. System shall display a “MyPanel” button in the profile page. 2. Upon clicking the “MyPanel” button, the system shall display all available listing(s) which the Owner had uploaded. |
| Alternative Flows: | None |
| Exceptions: | None |
| Includes: | None |
| Special Requirements: | None |
| Assumptions: | Owner has existing listings. |
| Notes and Issues: | None |

II.IV EditListing

| | |
|--------------|------|
| Use Case ID: | #2-9 |
|--------------|------|

| | | | |
|----------------|---------------|--------------------|---------------|
| 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"> 1. The Owner selects the “Edit” button, he would be allowed to edit on his selected listing. 2. System shall display a “Submit” button. 3. Upon clicking the “Submit” button, the system checks that all required fields are filled up. 4. The listing is updated. |

| | |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Alternative Flows: | <u>AF-S3: Owner did not fill up all required fields in the form</u> 1. Owner did not fill up all required fields in the form. 2. Owner clicks the “Submit” button, EditListing is called. 3. Checks that some required fields are not filled up. 4. System returns to step 1. |
| Exceptions: | None |
| Includes: | None |
| Special Requirements: | None |
| Assumptions: | Owner has existing listings. |
| Notes and Issues: | None |

II.X DeleteListing

| | | | |
|----------------|---------------|--------------------|---------------|
| Use Case ID: | #2-10 | | |
| 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"> 1. The Owner selects the “Delete” button. 2. System shall display a “Confirm” button and a “Cancel” button. 3. Upon clicking the “Confirm” button, the listing is removed from the database. |
| Alternative Flows: | <p><u>AF-S2: Owner clicks the “Cancel” button</u></p> <ol style="list-style-type: none"> 1. If Owner clicks the “Cancel” button, the System will not remove the listing. |
| Exceptions: | None |
| Includes: | None |
| Special Requirements: | None |

| | |
|-------------------|-----------------------------|
| Assumptions: | Owner has existing listings |
| Notes and Issues: | None |

II.XI FlagUser

| | | | |
|----------------|---------------|--------------------|---------------|
| Use Case ID: | #2-11 | | |
| 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: | <ol style="list-style-type: none"> 1. The System shall display a “FlagUser” icon. 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. 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. 4. Checks are made to ensure the description box is filled. 5. Owner/Adopter/PetFinder clicks the “Submit” button, a notification is sent to the Administrators and the database is updated. |
| 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"> 1. PetFinder is logged in and authenticated. 2. PetFinder can only report the missing animal through the corresponding LostPet listing |

| | |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Postconditions: | The Owner will be notified. |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | <ol style="list-style-type: none"> 1. System shall display a list of all Lost Pet(s). 2. PetFinder clicks on the respective LostPet listing which corresponds to the pet they found. 3. System shall display a “Report” button. 4. If the PetFinder clicks on the “Report” button, the System will direct the PetFinder to upload image(s) of the pet. 5. System displays a “Upload” and “Cancel” option. 6. PetFinder clicks on the “Upload” button and uploads image(s) of the pet as an image attachment with a short optional description. 7. System displays a “Confirm” and a “Cancel” button. 8. PetFinder clicks on the “Confirm” button. Uploaded images will be sent to the Owner and he/she will be notified. |
| Alternative Flows: | <u>AF-S5: PetFinder clicks “Cancel” button</u> <ol style="list-style-type: none"> 1. If the PetFinder clicks the “Cancel” button, the report will be dismissed and images will not be sent to the Owner. 2. The PetFinder will be brought back to the selected LostPet Listing. |
| 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"> 1. The System displays the details of the Pet that is up for adoption. 2. The System displays the “Adopt” button. 3. The Adopter clicks on the “Adopt” button. 4. System prompts the Adopter to fill in the following details: <ol style="list-style-type: none"> a. Age b. Gender c. Message to be sent to the Owner 5. System displays the “Confirm” and “Cancel” button. 6. The Adopter clicks on the “Confirm” button. 7. System sends the Adoption Request to the Owner of the Pet Listing. |
| Alternative Flows: | <u>AF-S5: PetFinder clicks “Cancel” button</u> <ol style="list-style-type: none"> 1. Adopter terminates the adoption process and the request is not sent to the Owner. 2. Adopter is brought back to the Pet Details page. |
| 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"> 1. Adopter is logged in and authenticated. 2. Adopter had already sent an Adoption Request to the Owner |
| 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"> 1. System displays the list of all adoption requests sent by the Adopter 2. Adopter clicks on the kebab menu on the top right-hand side of the desired adoption request. 3. The Kebab menu shows “Info” and “Cancel Request” buttons. 4. Adopter clicks on “Cancel Request”. 5. System requests Adopter to confirm cancellation request. 6. Adopter confirms cancellation request. 7. The Adopter successfully deletes the adoption request. |
| Alternative Flows: | <p><u>AF-S3: Adopter clicks “Info” button</u></p> <ol style="list-style-type: none"> 1. System will display the pet details of the adoption request. 2. System returns back to step 1. <p><u>AF-S4: Adopter does not confirm the “Cancel Request”</u></p> <ol style="list-style-type: none"> 1. Selected adoption requests will not be changed. 2. Adopter will be brought back to his/her list of adoption requests. |
| 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"> 1. The Adopter is logged in authenticated by the System 2. The Adopter must have previously made a successful adoption request to the Owner |
| Postconditions: | The Adopter will be able to view the list of adoption requests they have made. |
| Priority: | Medium |
| Frequency of Use: | Medium |
| Flow of Events: | <ol style="list-style-type: none"> 1. System shows the list of all adoption requests that the Adopter had submitted. 2. Adopter is able to view individual adoption request status by clicking the “View Status” button in each listing. |
| 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"> 1. System shall display a “Personality” button in the profile page. 2. If the Adopter selects the “Personality” button, he will be prompted to fill in a personality form with required fields. 3. The System checks for all required fields to be filled up. 4. The form is sent to the respective Pet Owner for him to evaluate the Adopter’s suitability for adoption. |
| Alternative Flows: | <u>AF-S3: Adopter did not fill up all required fields in the form</u> <ol style="list-style-type: none"> 1. Adopter did not fill up all required fields in the form. 2. Adopter clicks the “Submit” button. 3. System checks that all the required fields are not filled up. 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: | 1. System displays the list of all of the reports submitted by users. |

| | |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <ol style="list-style-type: none"> 2. Administrator selects an activity and views the details of the submitted report. 3. System prompts the Administrator to select the desired activity: DeleteListing, BanUser, DismissReport. 4. If the Administrator selects the activity DeleteListing, then the Administrator uses the included use case DeleteListing to delete the selected listing. 5. If the Administrator selects the activity BanUser, then the Administrator uses the included use case BanUser to ban the selected user. 6. If the Administrator selects the activity DismissReport, then the Administrator will be directed to the included use case DismissReport to delete the report. |
| Alternative Flows: | None |
| Exceptions: | None |
| Includes: | <ol style="list-style-type: none"> 1. DeleteListing 2. BanUser 3. DismissReport |
| 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"> 1. System displays a “Ban User” button. 2. Administrator clicks on the “Ban User” button. 3. System shall display a “Confirm” and a “Cancel” button. 4. Upon clicking the “Confirm” button, the Administrator bans the selected User. 5. System disables the user’s account. 6. The banned user is notified of the ban. |

| | |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| Alternative Flows: | <u>AF-S3: Administrator clicks the “Cancel” button</u> 1. If the Administrator clicks the “Cancel” button, the System will return to step 1. |
| 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"> 1. User selects a the listing that he/she wants to delete 2. User clicks the kebab menu and selects the “Delete Listing” icon. 3. System prompts Administrator’s confirmation and deletes the listing. 4. Owner of the listing is notified of the deletion. 5. Listing is removed from the database. |
| Alternative Flows: | <p><u>AF-S3: Administrator cancels the confirmation:</u></p> <ol style="list-style-type: none"> 1. Administrator selects the “Cancel” icon when prompted for deletion confirmation. |

| | |
|-----------------------|------------------------------------------------------------------------------|
| | 2. Administrator is brought back to the interface list of submitted reports. |
| Exceptions: | None |
| Includes: | None |
| Special Requirements: | None |
| Assumptions: | None |
| Notes and Issues: | None |

V.IV DismissReport

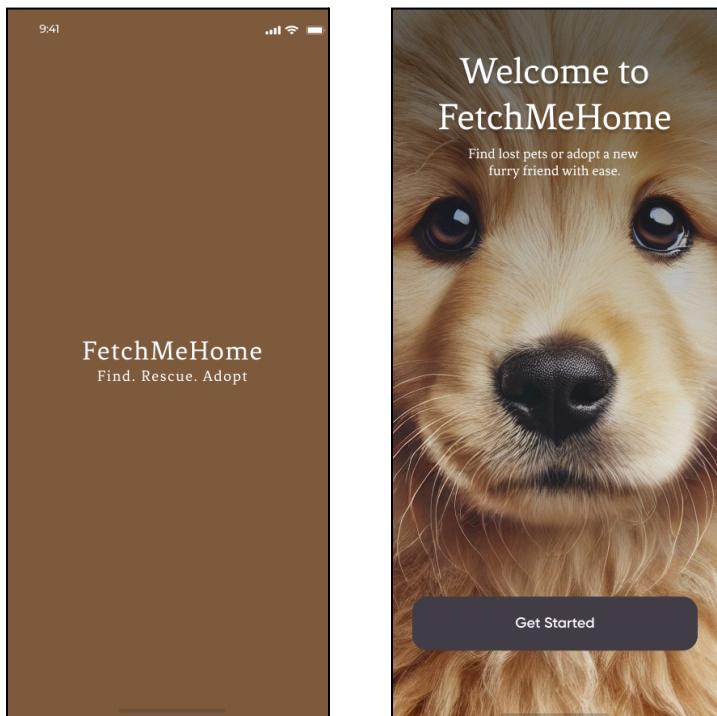
| | | | |
|----------------|---------------|--------------------|---------------|
| Use Case ID: | #5-4 | | |
| Use Case Name: | DismissReport | | |
| Created By: | Benjamin Yeoh | Last Updated By: | Benjamin Yeoh |
| Date Created: | 8/2/25 | Date Last Updated: | 8/2/25 |

| | |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Actor: | Administrator |
| Description: | Administrator can remove reports that he/she deems illogical |
| Preconditions: | Administrator is logged in and authenticated. |
| Postconditions: | Report is deleted |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | <ol style="list-style-type: none"> 1. System displays a “Dismiss Report” button. 2. Administrator clicks on the “Dismiss Report” button. 3. System shall display a “Confirm” and a “Cancel” button. 4. Upon clicking the “Confirm” button, the Administrator removes the selected report. 5. Report is removed from the database. |
| Alternative Flows: | <p><u>AF-S3: Administrator cancels the confirmation:</u></p> <ol style="list-style-type: none"> 1. Administrator selects the “Cancel” icon when prompted for deletion confirmation. 2. Administrator is brought back to the interface list of submitted reports. |
| Exceptions: | None |
| Includes: | None |

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

4. UI MockUps

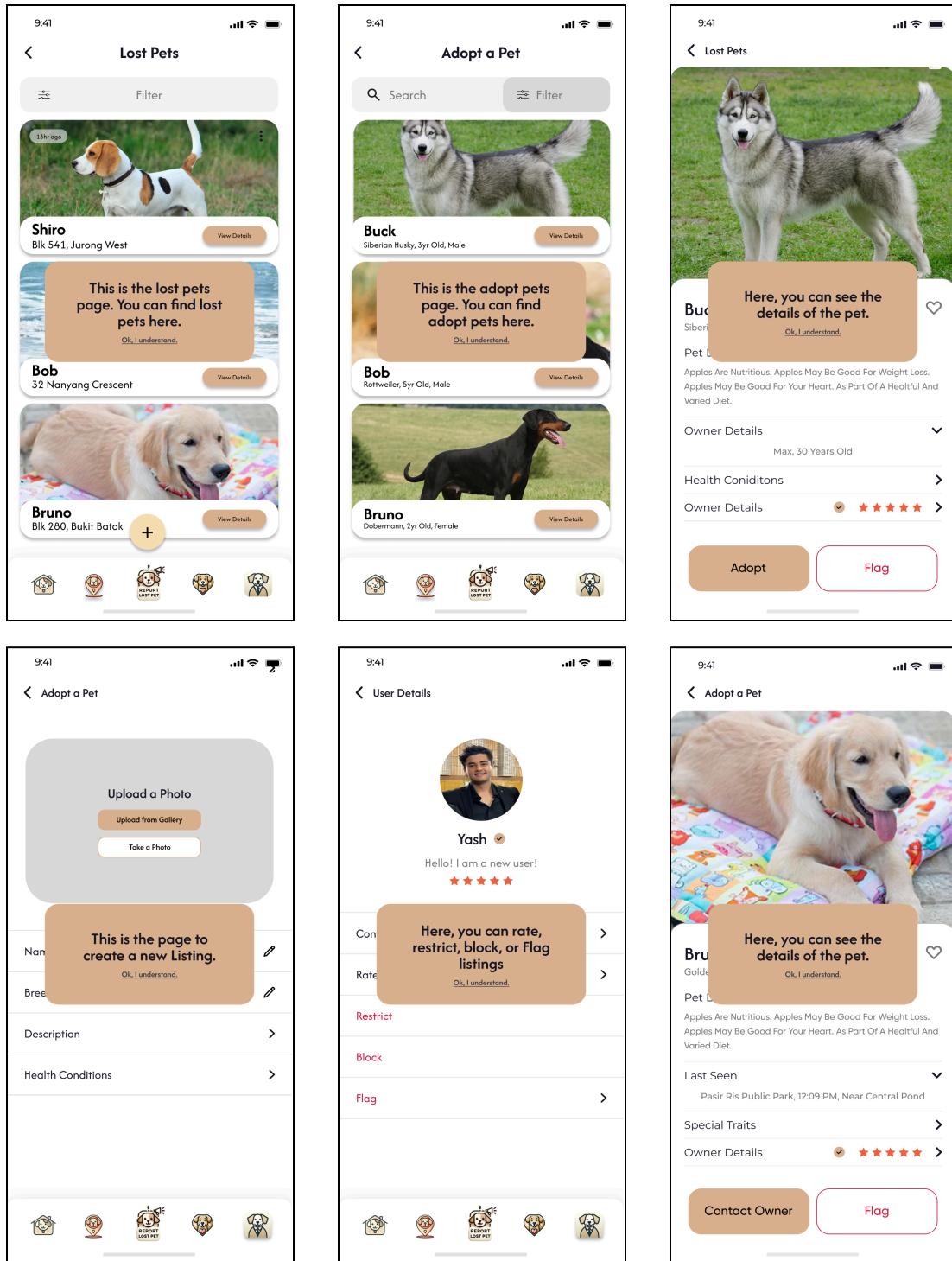
4.1 Introduction Page



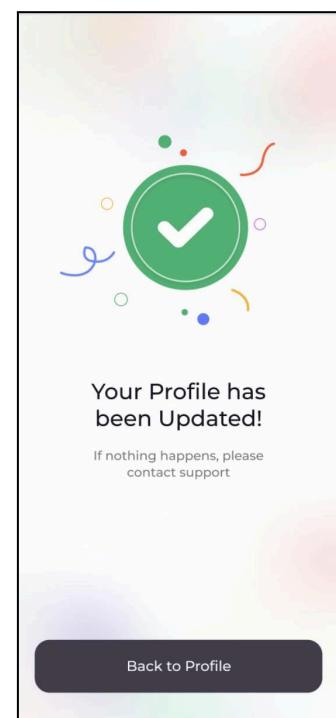
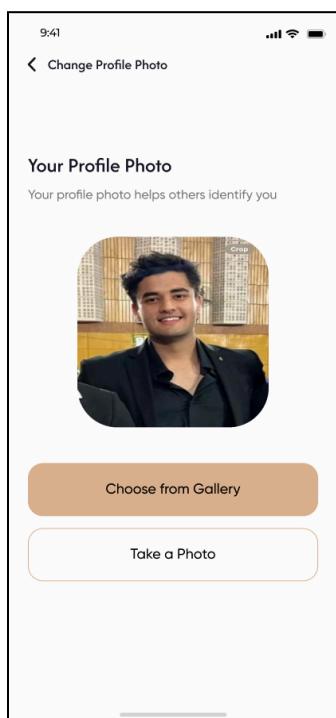
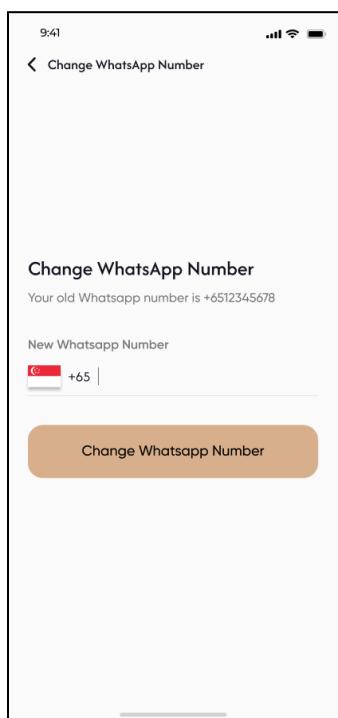
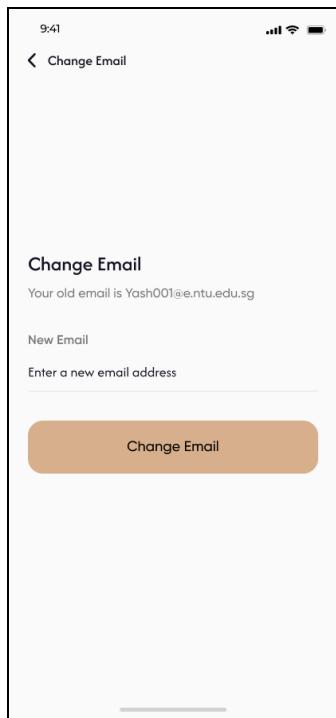
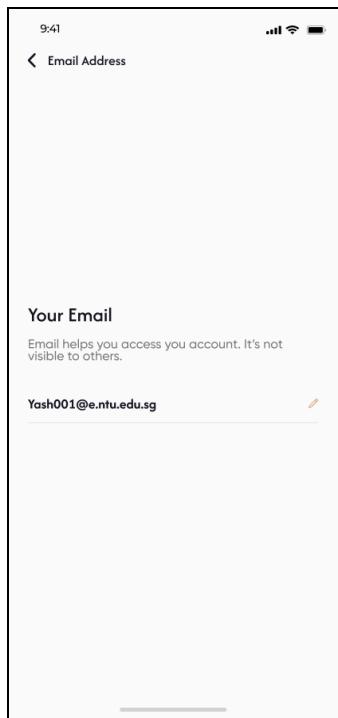
4.2 Sign Up

4.3 Login

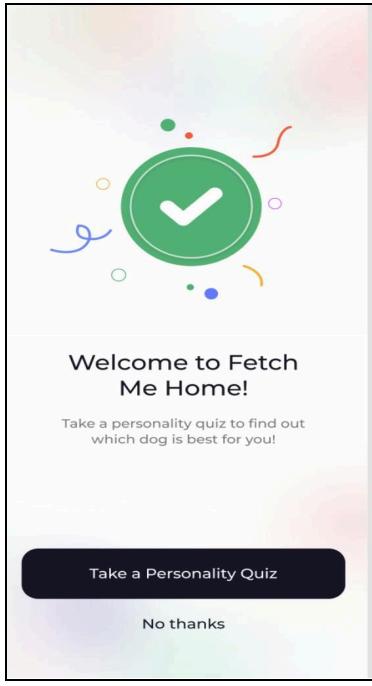
4.4 Walkthrough



4.5 Account settings



4.6 Welcome to FetchMeHome / Personality Quiz



9:41

Personality Test

Rate these statements from 1-5

I like hanging out with new people

Big no No Neutral Yes Big yes

I like hanging out with new people

Big no No Neutral Yes Big yes

I like hanging out with new people

Big no No Neutral Yes Big yes

I like hanging out with new people

Big no No Neutral Yes Big yes

I like hanging out with new people

Big no No Neutral Yes Big yes

Submit

Skip this section

9:41

Personality Test

Rate these statements from 1-5

You are a Dog Person! 🐕

🐶 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.

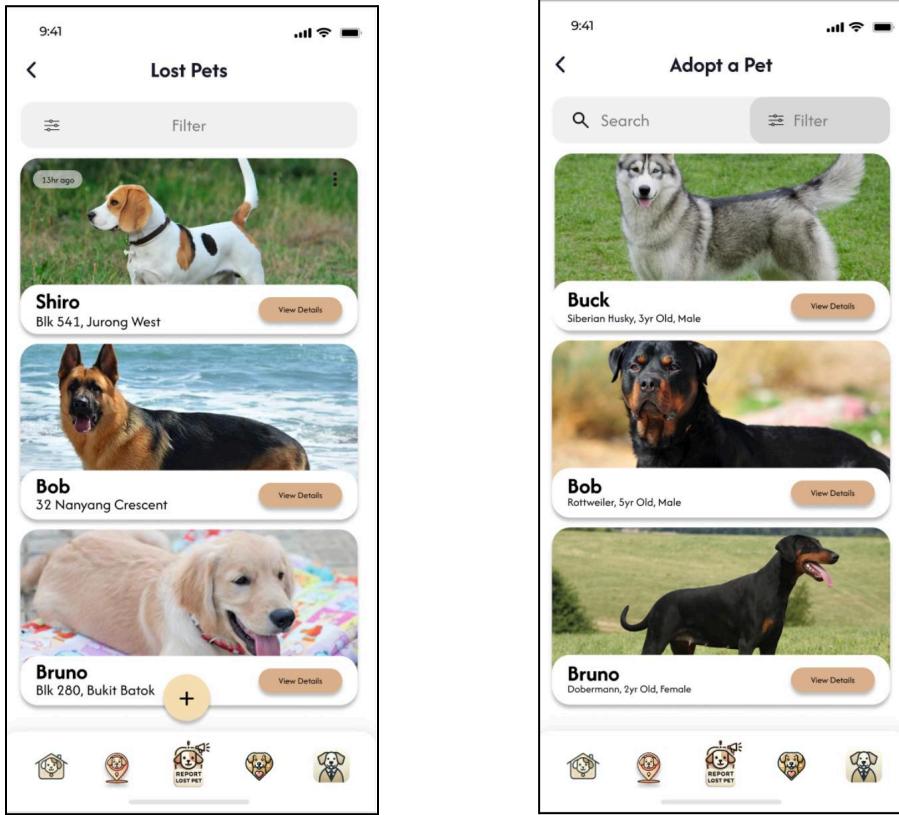
🐱 Mostly B's - You Are A Cat!
Independent Yet Affectionate On Your Terms, You Enjoy Cozying Up At Home But Also Need Your Space. A Low-Maintenance But Loving Pet Like A Cat Is Perfect For You.

🐦 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.

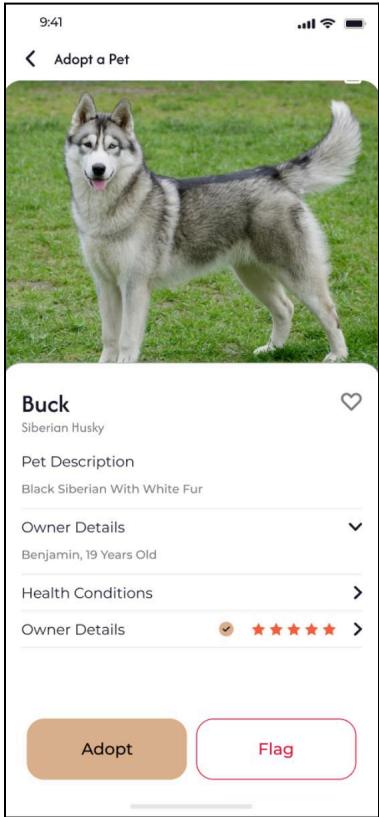
Save Results

4.7. View Pets up for Adoption / Missing Pets

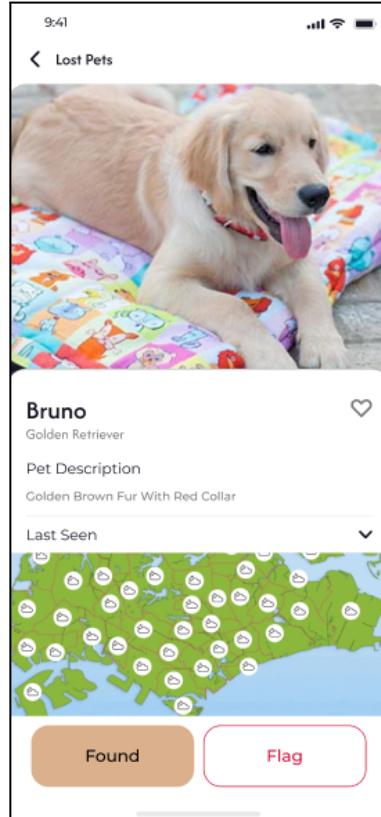


4.8 View details of listings

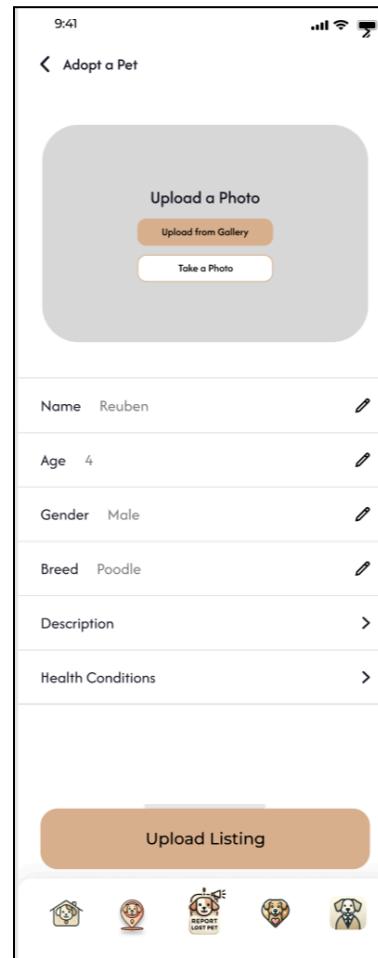
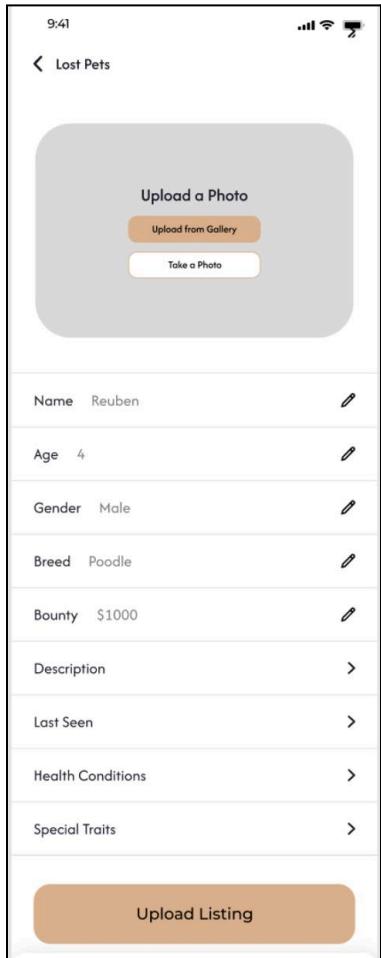
AdoptPetListing



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

9:41

◀ Description

Pet Description

Tell us something about your pet

Your pet description goes here. Be as complete as possible!

Update Pet Description

[Back to Previous Page](#)

9:41

◀ Last Seen

Last Seen

Where did you last see your pet.

Your pet's last known location goes here. Be as complete as possible!

Update Last Seen

[Back to Previous Page](#)

9:41

◀ Special Traits

Special Traits

Tell us something unique about your pet.

Your pet's special traits goes here. Be as complete as possible!

Update Special Traits

[Back to Previous Page](#)

9:41

◀ Health Conditions

Health Conditions

How's your pet doing?

Your pet's health conditions goes here. Be as complete as possible!

Update Health Conditions

[Back to Previous Page](#)



Your Listing has been Updated

If nothing happens, please contact support

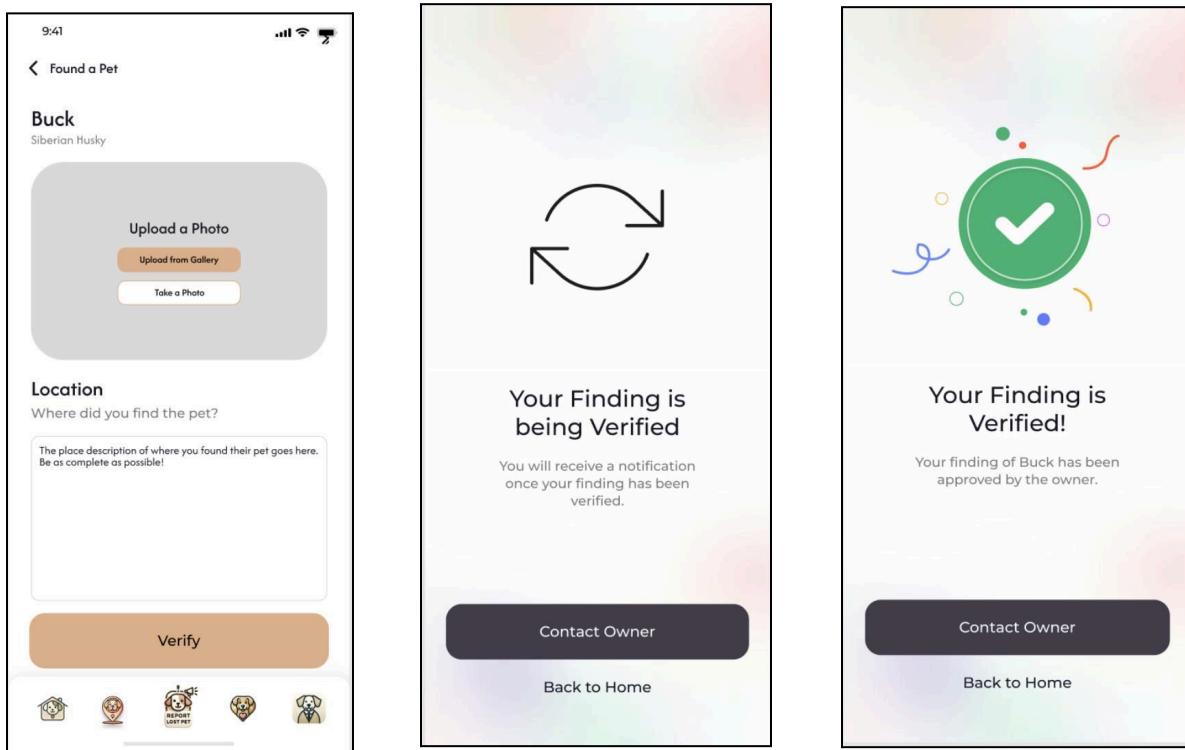
Back to Home

Undo Changes

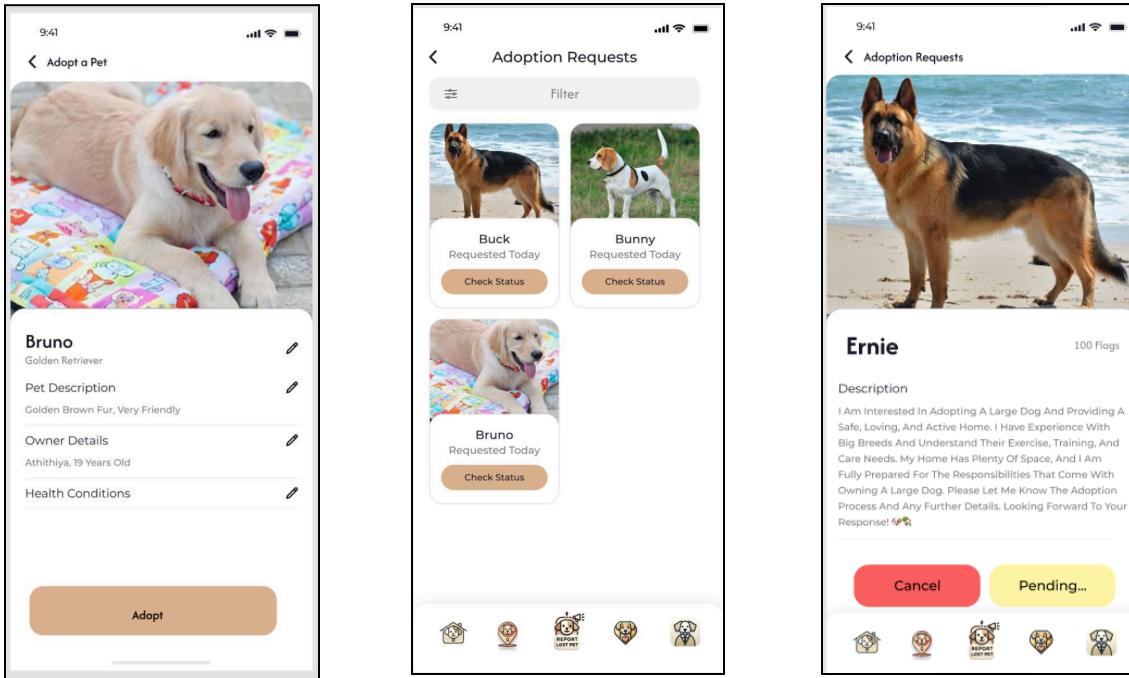
4.11 Delete Listing



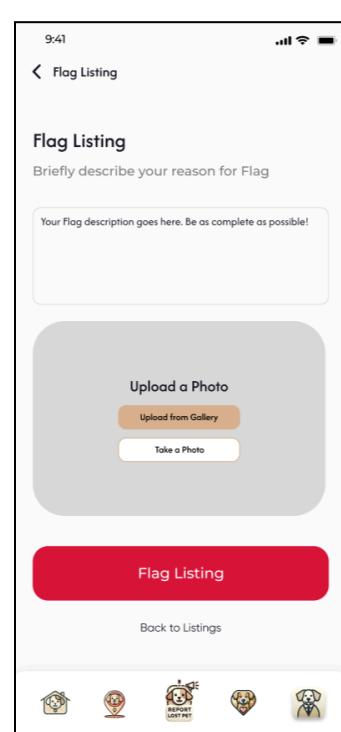
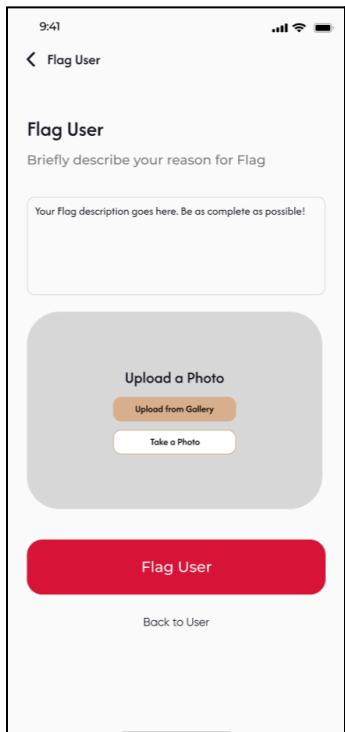
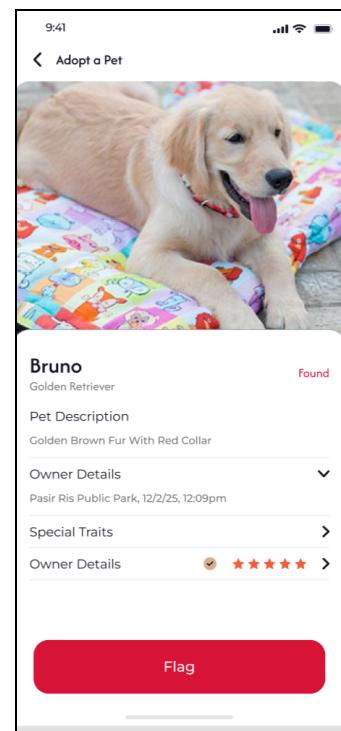
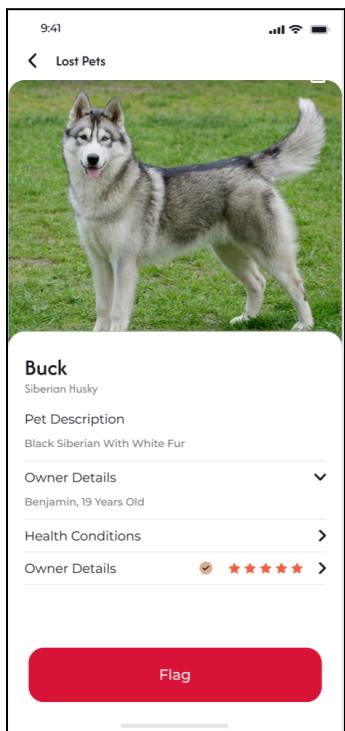
4.12 PetFinders can Submit Lost Report for Missing Pets



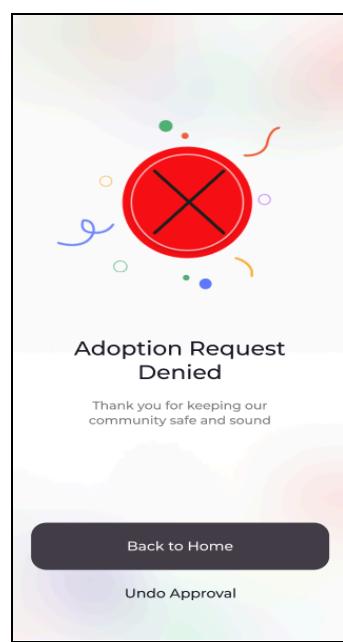
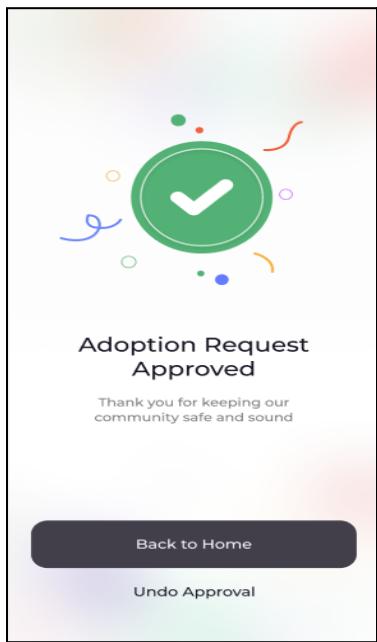
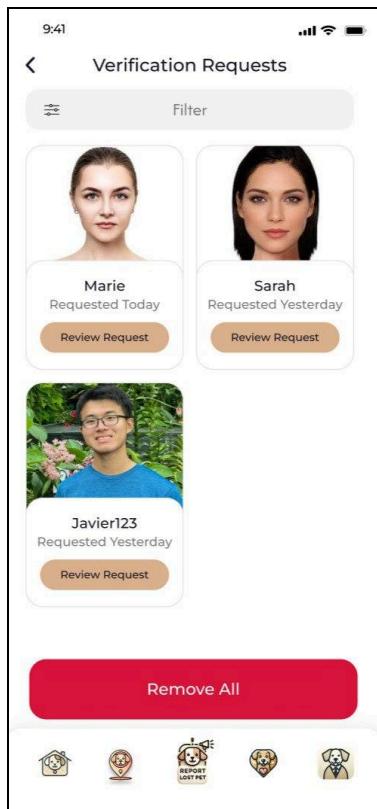
4.13 Adopters can submit adoption request and check their status of request



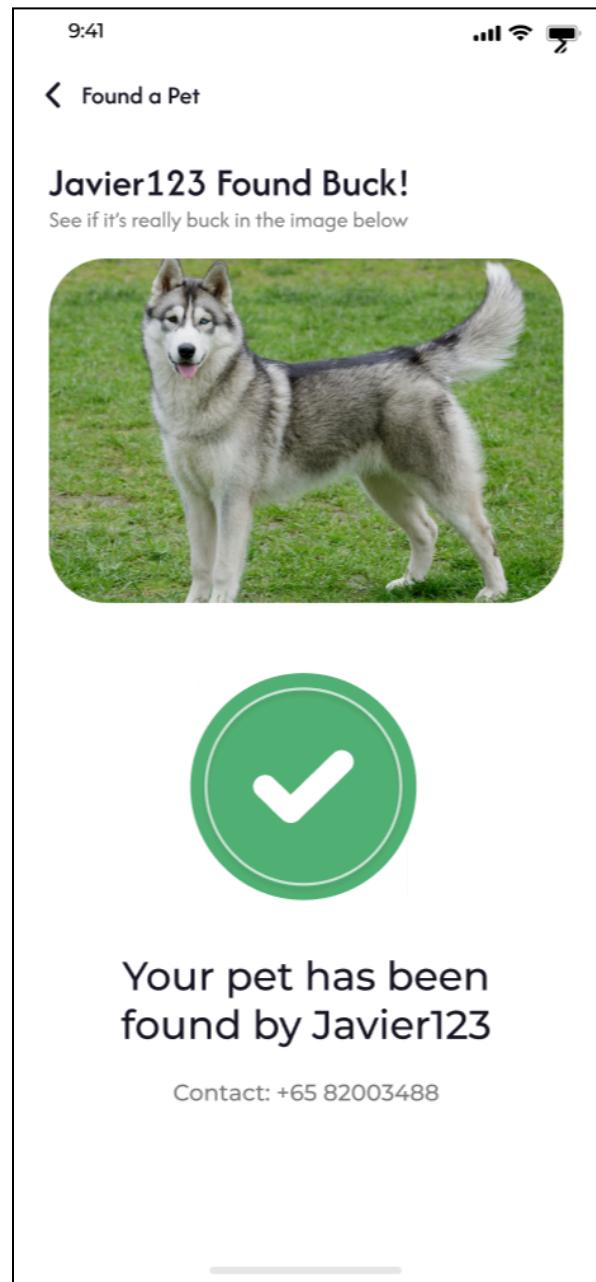
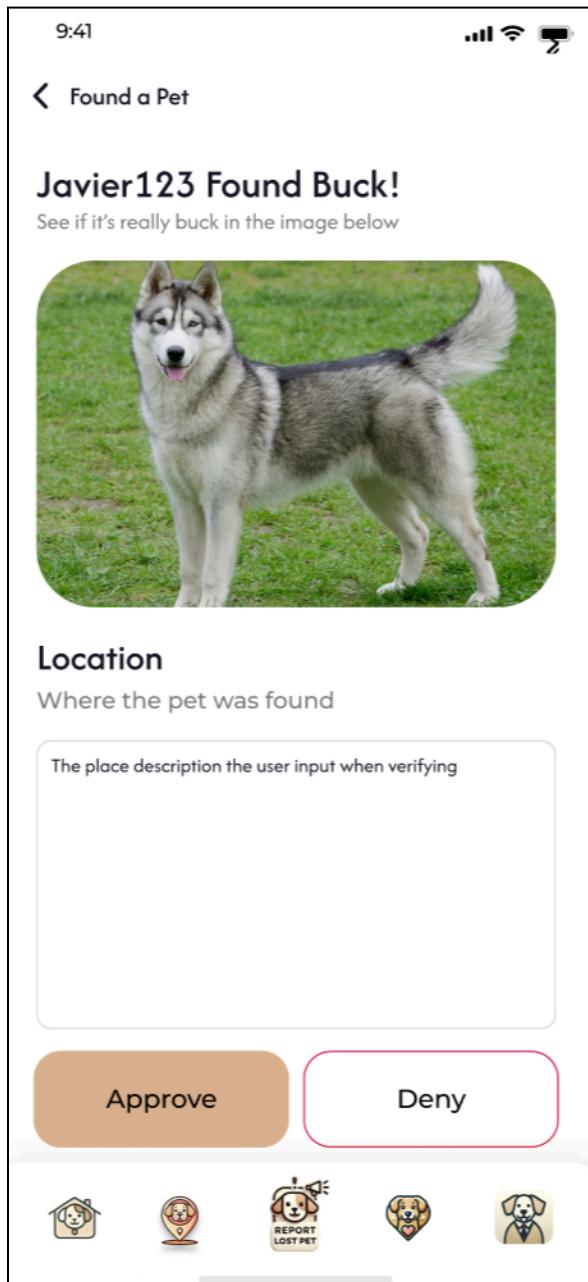
4.14 Adopters and Pet Finders can flag adoption and missing pet listings as well Owners



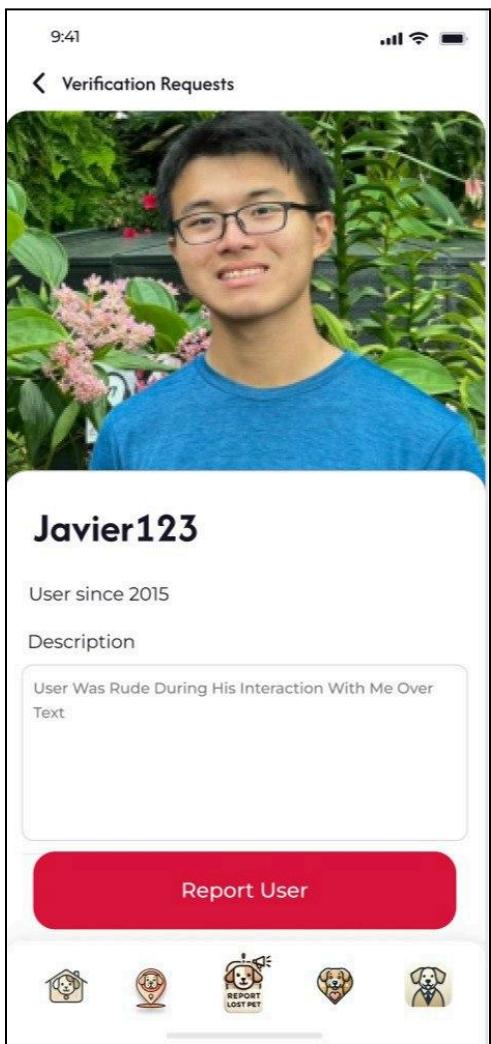
4.15 Owners can view adoption requests, as well as approve or deny them



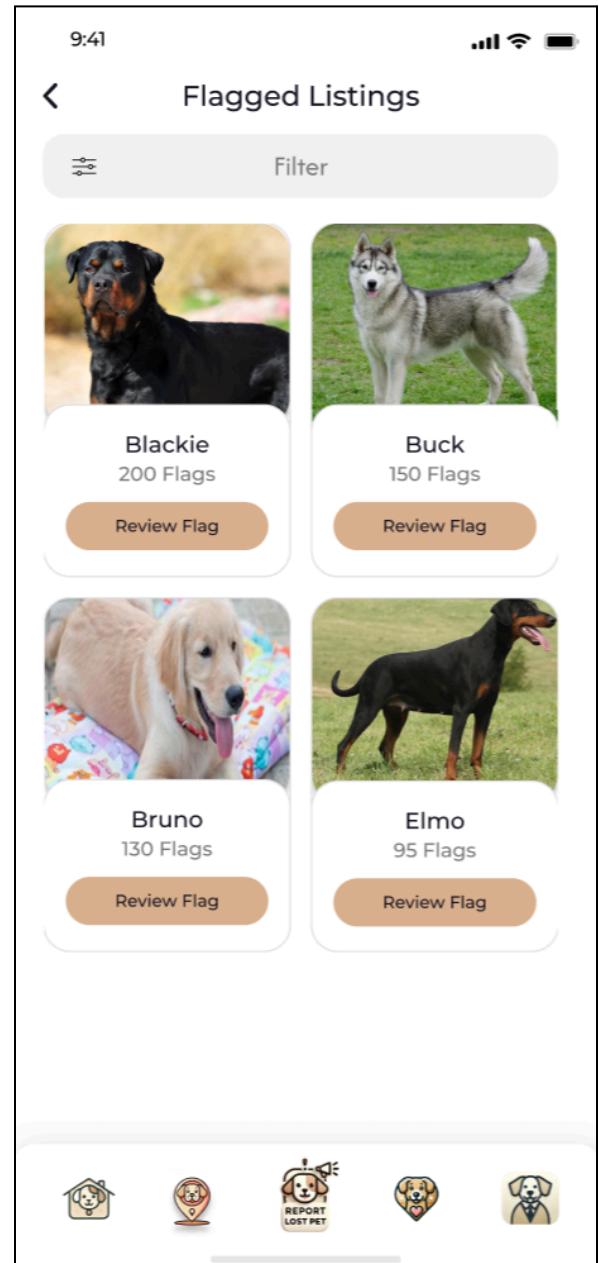
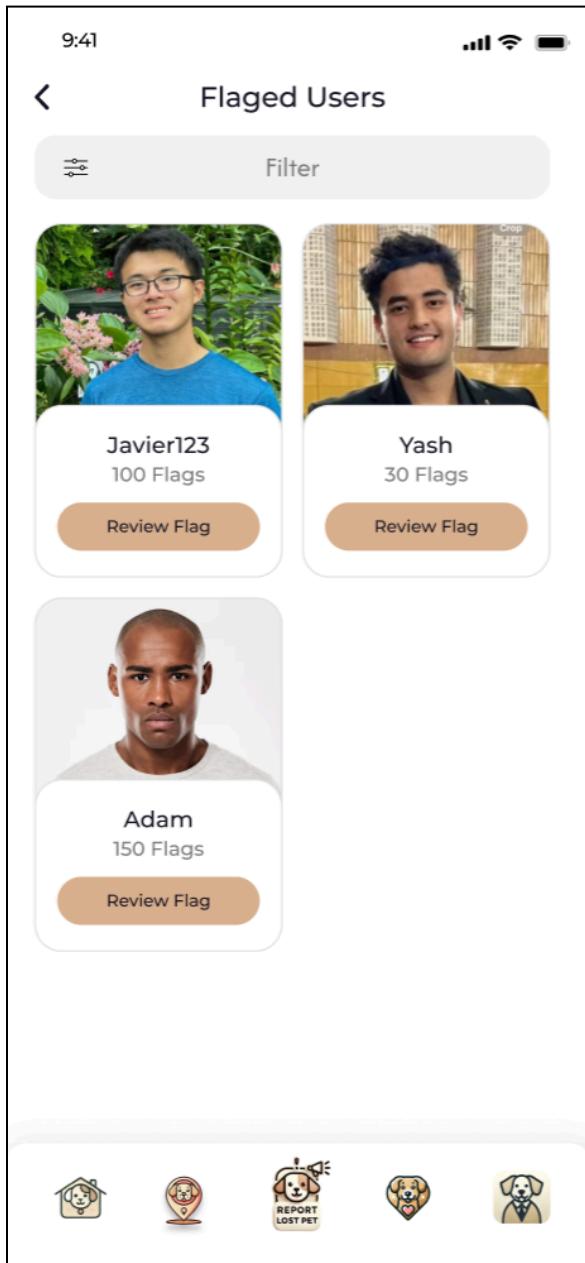
4.16 Owners can approve lost pet reports, as well as approve or deny them.



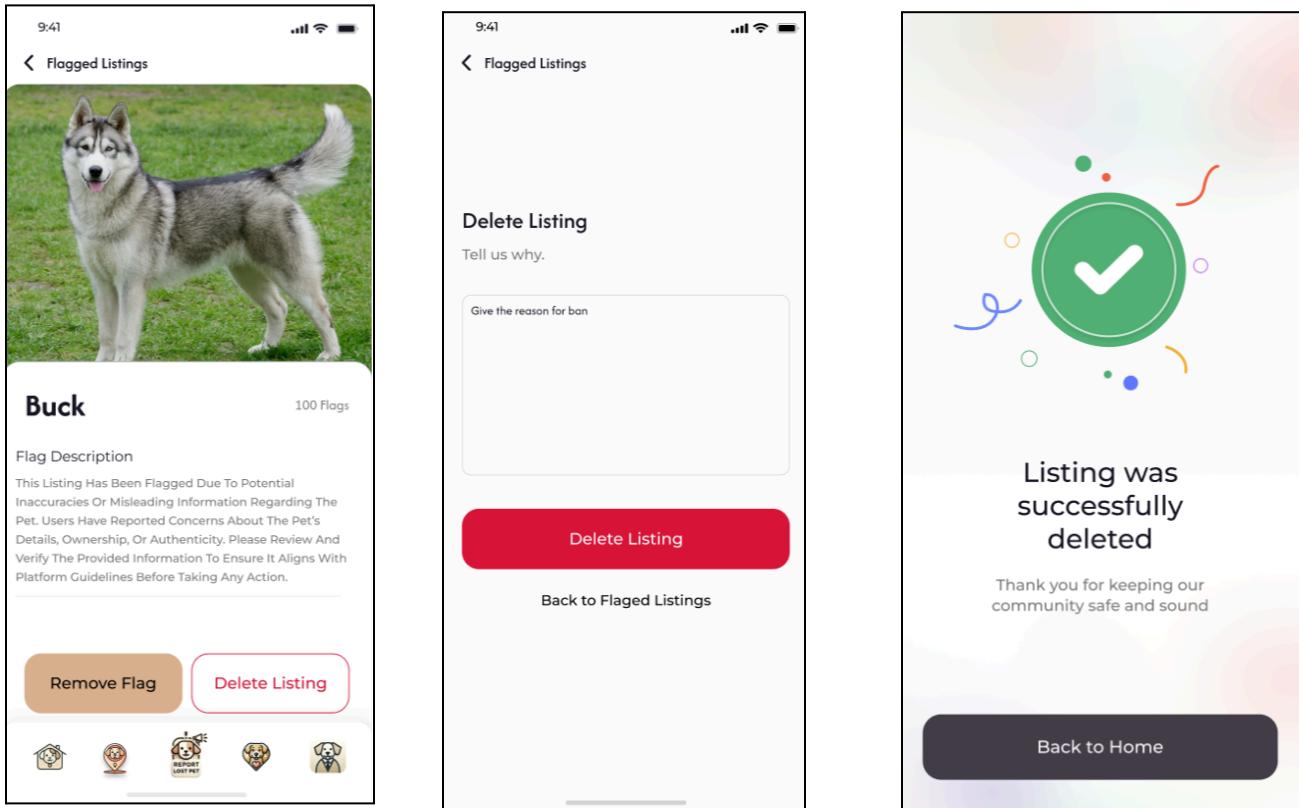
4.17 Owner reporting Adopter / PetFinder for misconduct



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

