



MINISTRY OF EDUCATION AND TRAINING

TRƯỜNG ĐẠI HỌC FPT

2012



FPT UNIVERSITY

Capstone Project Document

Pawnder: Pet Dating App

SEP490_G151	
Group Members	Nguyen Van Duc HE171702 Nguyen Ta Tri Cuong HE173555 Le Quoc Tuan HE171554 Doan Quoc Viet HE170543 Nguyen Minh Sang HE176048
Supervisor	Ngo Van Binh
Ext Supervisor	
Capstone Project code	DPDA

- Hanoi, December/2025 -

Table of Contents

Acknowledgement.....	3
Definition and Acronyms.....	4
I. Project Introduction.....	5
1. Overview.....	5
2. Product Background.....	5
3. Existing Systems.....	6
4. Business Opportunity.....	11
5. Software Product Vision.....	12
6. Project Scope & Limitations.....	13
II. Project Management Plan.....	16
1. Overview.....	16
2. Management Approach.....	23
3. Project Deliverables.....	25
4. Responsibility Assignments.....	26
5. Project Communications.....	27
6. Configuration Management.....	28
III. Software Requirement Specification.....	30
1. Product Overview.....	30
2. User Requirements.....	32
3. Functional Requirements.....	43
4. Non-Functional Requirements.....	56
5. Requirement Appendix.....	58
IV. Software Design Description.....	64
1. System Design.....	64
2. Database Design.....	70
3. Detailed Design.....	86
V. Software Testing Documentation.....	174
1. Scope of Testing.....	174
2. Test Strategy.....	175
3. Test Plan.....	180
4. Test Cases.....	181
5. Test Reports.....	182
VI. Release Package & User Guides.....	183
1. Deliverable Package.....	183
2. Installation Guides.....	184
3. User Manual.....	190

Acknowledgement

The completion of the Pawnder App project marks a pivotal chapter in our academic and personal journey—a journey filled with both formidable challenges and transformative growth. As we bring this report to its conclusion, we would like to take this opportunity to extend our deepest and most heartfelt gratitude to all those who have supported, guided, and inspired us throughout this endeavor.

First and foremost, we are profoundly grateful to our esteemed supervisor, Mr. Ngo Van Binh. His exceptional knowledge in software architecture and design, coupled with his unwavering dedication and insightful guidance, played a crucial role in the development of the Pawnder App system. His constructive feedback and encouragement empowered us to navigate the project's most complex obstacles with resilience and clarity. We are especially thankful for his patience and belief in our potential, which sustained our motivation during the most demanding phases of this project.

We would also like to express our sincere appreciation to FPT University for providing a nurturing academic environment and the necessary resources that made the realization of this project possible. The institution's commitment to innovation and excellence has been a continuous source of inspiration for us.

Our heartfelt thanks go to the dedicated lecturers of FPT University, whose unwavering support and mentorship throughout the semester and during our Capstone Project have been truly invaluable. Their passion for teaching, combined with their encouragement and wisdom, has not only shaped us into capable software engineers but also instilled in us the confidence to pursue our ambitions with purpose and determination.

Lastly, and most dearly, we wish to acknowledge the love and support of our families and friends. Their constant encouragement, understanding, and sacrifices provided us with the strength and emotional balance needed to undertake this demanding project. Without their unwavering presence, this achievement would not have been possible.

The Pawnder App—a pet dating and matchmaking application—stands as a testament to our collective effort, resilience, and commitment. By creating a platform that connects pet owners and helps pets find compatible companions for breeding, adoption, or friendship, we hope to contribute meaningfully to the pet-loving community in Vietnam and beyond. This project symbolizes not only our growth as aspiring software professionals but also the enduring support system that has surrounded us. We look back with pride and gratitude, and we move forward with renewed determination and a profound sense of purpose.

Pawnder Project

Hanoi, December 20th, 2025

Definition and Acronyms

Acronym	Definition
BR	Business Rule
ERD	Entity Relationship Diagram
GUI	Graphical User Interface
SRS	Software Requirement Specification
UAT	User Acceptance Test
UC	Use Case
API	Application Program Interface

I. Project Introduction

1. Overview

1.1 Project Information

- Project name: Pawnder: Pet Dating App
- Project code: DPDA
- Group name: SEP490_G151
- Software type: Mobile Application

1.2 Project Team

Full Name	Role	Email	Mobile
Nguyễn Văn Đức	Member	ducnvhe171702@fpt.edu.vn	0985068091
Nguyễn Minh Sáng	Leader	sangnmhe176048@fpt.edu.vn	0342362698
Nguyễn Tạ Trí Cường	Member	cuongntthe173555@fpt.edu.vn	0395866168
Lê Quốc Tuấn	Member	tuanlqhe171554@fpt.edu.vn	0909280703
Đoàn Quốc Việt	Member	vietdqhe170543@fpt.edu.vn	0947635868

2. Product Background

The pet industry is experiencing rapid growth worldwide, with pets becoming increasingly popular companions for millions of households. In recent years, pets are more often viewed as integral members of the family, influencing lifestyle choices and daily social interactions. Globally, the pet market continues to expand as spending on pet care, products, and services grows year-over-year.

Empirical evidence substantiates this shift in user demand toward more responsible and regulated breeding practices. A large-scale survey conducted by Woodhead *et al.* involving 986 dog owners in Australia indicates that the majority of respondents assign high importance to breeding dog welfare, while a substantial proportion support the implementation of stricter regulatory frameworks to safeguard animal welfare during breeding activities [1]. Furthermore, the study identifies a persistent information gap, as many respondents report limited access to reliable and transparent information regarding breeding standards, partner selection criteria, and breeder credibility. As a consequence, breeding-related decisions are frequently made through informal online communities or personal networks, which are widely perceived as inefficient and associated with elevated risks [1].

From the perspective of breeders, convergent challenges have been documented. The *Australian Dog Breeders Study* published by MDPI reveals that breeders increasingly recognize pet owners' expectations for comprehensive and well-documented pet information, encompassing health records, behavioral attributes, and breeding histories [2]. Nevertheless, the study emphasizes that existing mechanisms for connecting breeders and pet owners remain fragmented and insufficiently standardized, thereby constraining effective information exchange and limiting the accuracy and reliability of breeder-owner matching processes [2].

At the market level, these behavioral and perceptual trends are mirrored in the economic expansion of breeding-related services. The global pets breeding management market was estimated at approximately USD 2.1 billion in 2024 and is projected to reach nearly USD 3.8 billion by 2035, corresponding to a compound annual growth rate (CAGR) of around 6.0% [3]. This sustained growth

reflects increasing investment in data-driven, technology-enabled solutions aimed at improving transparency, efficiency, and welfare outcomes within breeding ecosystems.

Collectively, these findings delineate a pronounced gap between the escalating demand for transparent, welfare-oriented, and information-rich breeding practices and the limited availability of technological platforms capable of systematically supporting such requirements. Addressing this gap represents a significant opportunity for digital solutions that integrate structured data management, transparent interaction mechanisms, and trusted connectivity between pet owners and breeders.

Today, pet owners face several challenges in managing and connecting around their pets. They often struggle to find reliable platforms to connect with other pet owners in their local area, share meaningful stories within communities that truly understand pet ownership, and ensure safe opportunities for pets to socialize. In particular, activities such as finding suitable playmates, companions, or even coordinating responsible breeding connections are commonly handled through informal online groups, word-of-mouth networks, or chance encounters. These approaches are often inefficient, unverified, and may involve potential risks.

This gap presents a clear opportunity for a more specialized solution. Pet owners increasingly seek a platform that goes beyond basic photo sharing or casual interaction, offering practical tools tailored to real pet-related needs. Pawnder is designed to address this opportunity by providing a dedicated digital environment that supports structured pet profiles, interaction, and information sharing.

Pawnder reimagines how pet owners interact with each other and manage their pets in the digital environment. Inspired by familiar interaction patterns such as the swipe-to-match concept, the platform provides an intuitive yet practical way for users to explore connections and engage within a pet-centered community. Each pet is represented through a dedicated profile containing structured information such as breed, age, personality, and preferences, enabling more informed and meaningful interactions.

By combining social networking with targeted interaction features, Pawnder positions itself as more than a conventional pet application. The platform helps pet owners save time, reduce uncertainty, and build trusted connections within a focused ecosystem. Ultimately, Pawnder responds to the growing demand in the pet-tech space for a dedicated platform that supports responsible interaction, community building, and future technological expansion.

From an information technology perspective, the challenge of Pawnder lies not only in building a connection platform, but also in processing and standardizing pet data, which is inherently unstructured. Pet images uploaded by users vary significantly in terms of shooting angles, lighting conditions, and image quality, posing challenges for feature recognition and extraction. To address this, Pawnder applies artificial intelligence-based image recognition techniques in combination with a structured data system, aiming to ensure consistency, scalability, and reliability in pet profiles.

Reference :

[1] Woodhead, L., et al., *Perceptions of dog breeding practices, breeding dog welfare and companion dog acquisition in a self-selected sample of Australian adults*, Animal Welfare, vol. 27, no. 4, pp. 343–361, 2018.

[2] Australian Dog Breeders Study, *A Quantitative Exploration of Australian Dog Breeders' Breeding Practices*, Animals (MDPI), 2025.

[3] Euromonitor International, *Pet Care in Asia Pacific – Market Overview and Forecast*, 2024.

3. Existing Systems

3.1 Petfinder

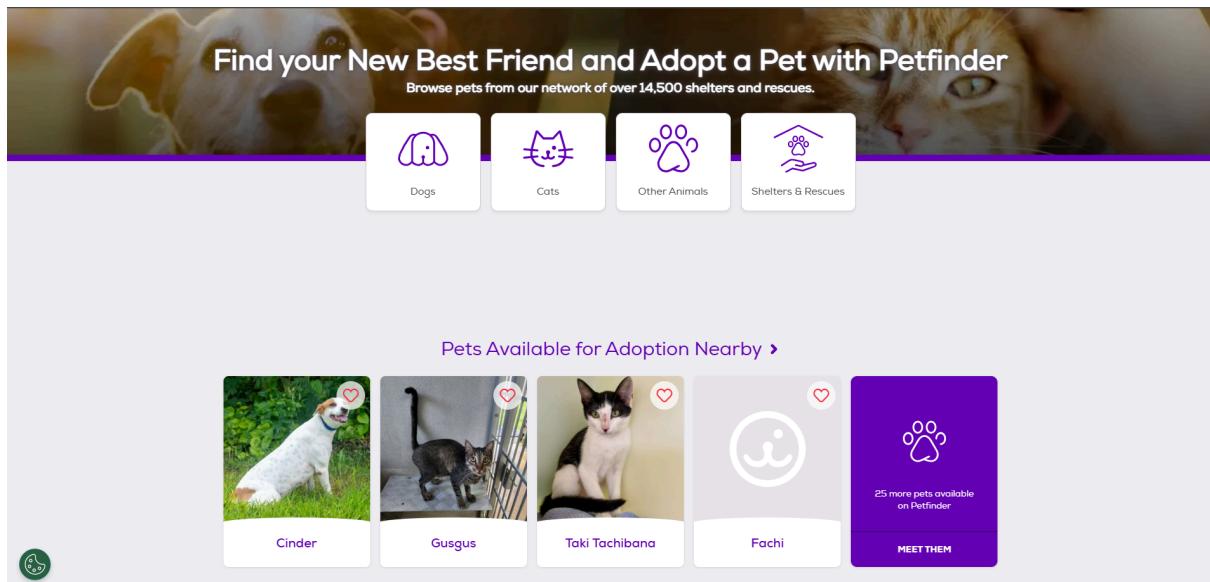


Figure 3.1. Systems Petfinder

- **Description:** Petfinder is a large-scale online platform designed to support pet adoption by connecting potential adopters with shelters, rescue organizations, and individual pet owners. The system primarily focuses on listing pets available for adoption (mainly dogs and cats) and providing descriptive information such as breed, age, personality, and health condition to support adoption decisions.
- **Link:** <https://www.petfinder.com/>
- **Actors:**
 - **Pet seekers:** Individuals searching for pets available for adoption based on predefined criteria such as breed, location, and age.
 - **Pet Provider:** Shelters, adoption centers, or individual pet owners who upload and manage pet profiles for adoption purposes.
 - **Platform Administrator:** Responsible for managing system content, approving listings, and maintaining overall platform operation.
- **Features:**
 - **Pet Discovery and Search:** Allows pet seekers to search and filter available pets based on attributes such as species, breed, age, and geographic location.
 - **Pet Profile Management:** Displays detailed pet profiles created and maintained by pet providers, including descriptive and health-related information.
 - **Adoption Connection Workflow:** Facilitates communication between pet seekers and pet providers through contact information or inquiry mechanisms.
 - **Multi-platform Availability:** Accessible via both web and mobile applications.
- **Pros:**
 - **Supports Structured Adoption Workflows:** Petfinder provides a structured process for adoption by centralizing pet listings from multiple shelters and organizations, reducing reliance on informal channels.
 - **Large and Diverse Pet Database:** Aggregates a significant number of pet profiles from various regions, increasing adoption opportunities and visibility for shelters.

- **Clear Role Separation:** Distinguishes clearly between pet seekers and pet providers, enabling focused system interactions aligned with adoption use cases.
- **User-friendly Interface:** Simplifies pet discovery through intuitive search and filtering mechanisms, lowering the barrier for first-time adopters.
- **Cons:**
 - **Adoption-centric System Design:** The platform is primarily designed for pet adoption and does not support ongoing social interaction or relationship-building between pets.
 - **Limited Species-specific Experience:** Petfinder does not provide specialized interaction models or features tailored specifically for cats or other individual species.
 - **Manual Profile Creation:** Pet profiles rely on manually entered information, with no AI-based image analysis or automated data standardization mechanisms.
 - **No Intelligent Pet-to-Pet Matching:** The system does not support matchmaking or compatibility analysis between pets, as its focus is on adoption rather than social or breeding-related use cases.

3.2 Tinder

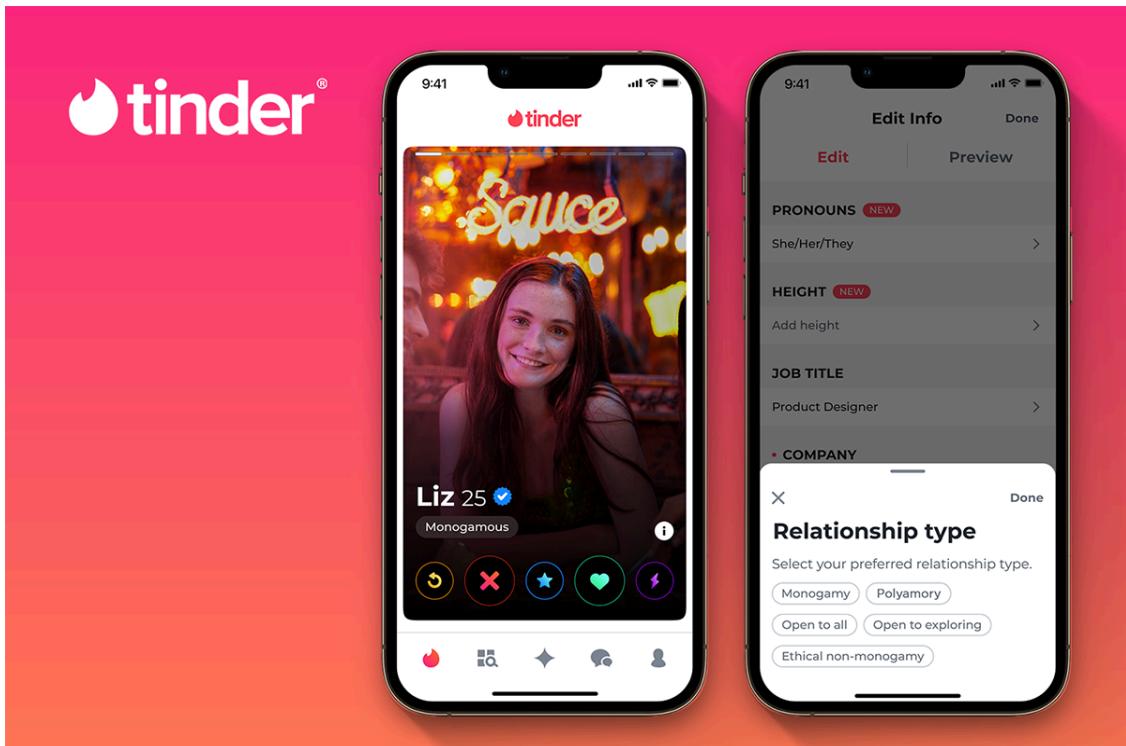


Figure 3.2. Systems Tinder

- **Description:** Tinder is a globally popular dating application that enables people to discover and connect with others through an intuitive swipe left/right interaction model. The platform is designed to facilitate fast user-to-user matching and communication, primarily within the context of human social relationships. Tinder's core experience is driven by profile presentation, user interaction signals (likes, passes), and system-level recommendation and ranking mechanisms.
- **Link:** <https://tinder.com/>
- **Actors:**

- **End User:** Individuals who create personal profiles, browse other profiles, and perform swipe actions to express interest.
 - **Platform Administrator/ Moderation Team:** Responsible for enforcing platform policies, managing reports, and maintaining platform integrity and safety controls.
 - **Subscription User (Secondary User):** End users who access premium features through paid plans.
- **Features:**
 - **User Profile Management:** Allows users to create and maintain personal profiles including photos and basic personal information.
 - **Swipe-based Interaction Mechanism:** Users swipe right to express interest or swipe left to skip; a match occurs when two users mutually express interest.
 - **Matching& Messaging:** Enables communication between matched users through an in-app chat mechanism.
 - **Discovery and Recommendation Support:** Supports user discovery based on factors such as location proximity and stated preferences, combined with platform-level ranking/recommendation logic.
 - **Pros:**
 - **User Profile Management:** Allows users to create and maintain personal profiles including photos and basic personal information.
 - **Swipe-based Interaction Mechanism:** Users swipe right to express interest or swipe left to skip; a match occurs when two users mutually express interest.
 - **Matching & Messaging:** Enables communication between matched users through an in-app chat mechanism.
 - **Discovery and Recommendation Support:** Supports user discovery based on factors such as location proximity and stated preferences, combined with platform-level ranking/recommendation logic.
 - **Cons:**
 - **Human-centric Domain:** Tinder is designed for human dating, so its matching objectives and data structures are not directly transferable to pet matchmaking without domain adaptation.
 - **Profile Data Dependence:** Profile accuracy depends largely on user-provided information, which may vary in reliability and does not inherently include structured biological or pet-specific attributes.
 - **Limited Domain-specific Safety & Verification for Pet Use Cases:** While Tinder includes safety and moderation controls for human interactions, it is not designed around pet-related risks (e.g., breeding safety, health history, pet identity verification).
 - **No Pet-specific Data Standardization Pipeline:** Tinder's system does not include mechanisms such as image-based pet profiling or automated standardization of pet attributes, which are critical for pet-oriented matching.

3.3 BarkBuddy

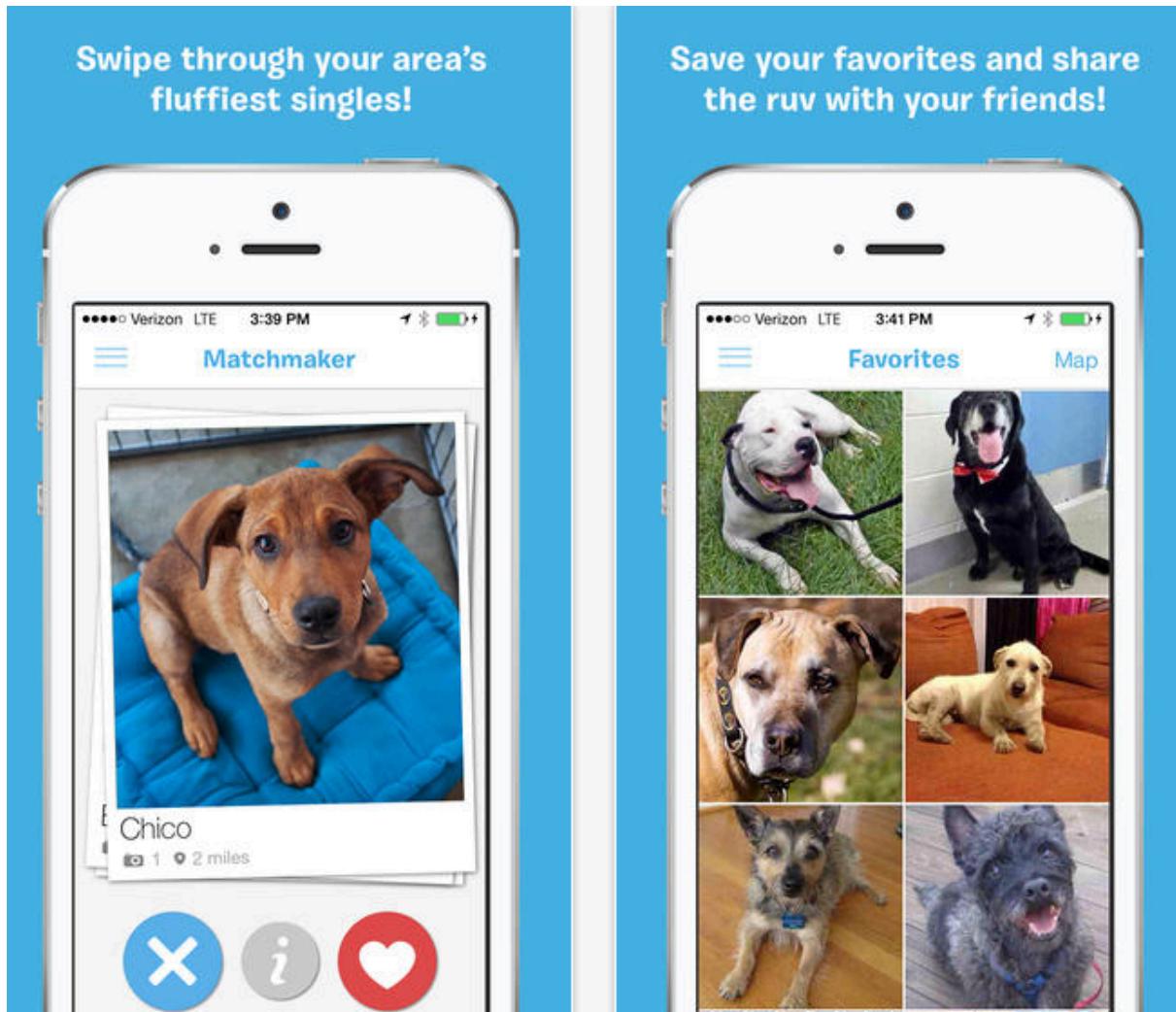


Figure 3.3. Systems BarkBuddy

- **Description:** BarkBuddy is a mobile application developed by BarkBox, designed to support dog adoption through a swipe-based interaction model similar to popular dating applications. The platform allows users to browse dog profiles from nearby rescue shelters and adoption centers, with the primary goal of increasing engagement and adoption rates through a simplified discovery experience.
- **Link:** <https://barkbuddy.com/>
- **Actors:**
 - **Pet Seeker (Primary User):** Individuals searching for dogs available for adoption and expressing interest through swipe-based interactions.
 - **Pet Provider (Secondary User):** Rescue shelters or adoption centers responsible for creating and managing dog profiles within the system.
 - **Platform Administrator:** Responsible for managing platform content, system maintenance, and operational oversight.
- **Features:**
 - **Swipe-based Pet Discovery:** Enables users to browse dog profiles using a left/right swipe interaction to express interest or skip.

- **Pet Profile Presentation:** Displays detailed dog profiles including breed, age, photos, and basic personality descriptions provided by shelters.
- **Location-based Discovery:** Supports finding adoptable dogs based on geographic proximity to the user.
- **Adoption Connection Workflow:** Connects interested users with nearby shelters or rescue organizations to initiate the adoption process.
- **Pros:**
 - **Engaging Interaction Model:** The swipe-based interface lowers user effort and increases engagement, making the adoption browsing process more approachable and interactive.
 - **Clear Adoption-oriented System Design:** BarkBuddy focuses specifically on facilitating dog adoption by streamlining the discovery and initial connection stages between adopters and shelters.
 - **Pet-focused User Experience:** Unlike general-purpose social platforms, BarkBuddy is tailored to pet-related use cases, aligning its features and content with the adoption context.
 - **Simple and Intuitive Interface:** The application emphasizes ease of use, enabling users to quickly explore multiple adoption options without complex workflows.
- **Cons:**
 - **Adoption-centric Scope:** BarkBuddy is designed exclusively for adoption workflows and does not support ongoing social interaction or pet-to-pet matchmaking beyond adoption use cases.
 - **Species Limitation:** The platform focuses solely on dogs and does not provide equivalent support for cats or other pet species.
 - **Manual Profile Data Dependency:** Pet profiles are created and maintained by shelters, relying on manually entered information without AI-based image analysis or automated data standardization.
 - **Limited Use of AI at the System Level:** While the platform supports location-based discovery and basic preference filtering, it does not integrate AI as a core component for pet profile analysis, compatibility assessment, or intelligent pet-to-pet matching.

3.4 Overall Existing Systems Conclusion

The analysis of existing systems highlights that current platforms address pet-related needs from different perspectives, including adoption-focused workflows (Petfinder, BarkBuddy) and interaction-driven matching models (Tinder). Each system effectively solves a specific problem within its domain, such as facilitating pet adoption or enabling fast human-to-human connections through intuitive interaction mechanisms.

However, from a system-level and data-processing perspective, these platforms exhibit several common limitations when considered in the context of pet social interaction and matchmaking. Pet adoption platforms primarily rely on manually created pet profiles and focus on discovery and connection rather than ongoing interaction or compatibility assessment between pets. Meanwhile, human dating applications demonstrate strong matching workflows and scalable interaction models but are not designed to handle pet-specific data structures, biological attributes, or domain-specific safety considerations.

Although some existing systems incorporate artificial intelligence in areas such as content recommendation, ranking, or moderation, AI is typically applied as a supporting feature rather than

being embedded into the core data-processing pipeline. In most cases, pet profile data is not standardized through automated image-based analysis, which limits the system's ability to support intelligent, data-driven pet-to-pet matching at scale.

This analysis reveals a clear functional and architectural gap in the current landscape: the absence of a specialized platform that integrates AI-assisted pet profile standardization with pet-centric interaction and matching mechanisms at the system level. Pawnder is positioned to address this gap by focusing on structured pet data generation, domain-specific interaction design, and AI-supported information assistance, particularly for cat owners. By combining lessons learned from existing systems with a pet-focused system architecture, Pawnder aims to provide a more reliable, scalable, and context-aware solution for pet social interaction and matchmaking.

4. Business Opportunity

4.1 Market & Current Problems:

The pet industry, particularly the cat segment, is experiencing rapid growth in Vietnam and other developing countries, as pet owners increasingly regard cats as integral members of their families. Along with this trend, the demand for reliable digital platforms that support interaction, connection, and information sharing among cat owners has grown significantly. However, despite this increasing demand, the current digital ecosystem still presents several fundamental limitations.

- **Lack of a dedicated, cat-focused information system:** Existing platforms do not provide a specialized information system designed specifically for cat-related interaction. As a result, interactions among cat owners are fragmented and not supported by a structured, domain-focused platform.
- **Manually created and unreliable pet profiles:** Pet profiles are primarily created through manual input, leading to inconsistent, incomplete, and unreliable data. This limits the accuracy of information and reduces trust in the profiles used for interaction and connection.
- **Absence of data verification and validation mechanisms:** The lack of systematic data verification and validation increases potential risks during the process of connection and matching, as users have limited means to assess the credibility of pet-related information.
- **Lack of integrated information support modules:** Current platforms do not integrate dedicated modules to support users with pet-related information or answer common questions, forcing users to rely on external sources that may be inconsistent or unreliable.
- **Fragmentation of pet-related functionalities across multiple platforms:** Pet-related services and functionalities are scattered across various independent platforms, resulting in a disconnected user experience and increased effort for pet owners to manage information and interactions.

4.2 Pawnder's Solution

Pawnder addresses existing challenges through a multi-layer technology pipeline:

- **Data Collection Layer:** Users upload pet images and basic information to the system.
- **AI Processing Layer:** The system applies image recognition models to analyze images and extract key features such as breed, coat color, and physical characteristics.
- **Data Standardization Layer:** The extracted information is transformed into structured pet profiles to support data storage and efficient querying.
- **Intelligent Matching Layer:** The matching algorithm recommends compatible pets based on extracted features, behavioral attributes, and geographic location.

- **Social Interaction Layer:** Users can communicate, share content, and build a community within a controlled and monitored environment.

4.3 Competitive Advantage

- Existing channels (e.g., Facebook, Zalo groups) primarily provide basic connections, lacking data verification mechanisms for pets.
- International applications (e.g., Petfinder) mainly focus on simple pet adoption or dating, with limited system-level support for automated feature extraction and intelligent matching based on image data.
- Pawnder differentiates itself as an all-in-one, data-driven platform that integrates AI-based profiling, intelligent matching, and a modular system architecture, optimized for local user behavior in the Southeast Asian market.

4.4 Trends & Growth Potential

- **AI adoption:** Users are increasingly open to AI-driven personalization in daily life.
- **Pet humanization:** Owners are willing to invest more in entertainment, care, and experiences for their pets.
- **Expanding pet services ecosystem:** Growth in grooming services, training centers, and breeding networks creates strong partnership opportunities for Pawnder.
- **Data-driven decision-making:** Pet owners are becoming more conscious of safety, compatibility, and reliability when choosing playmates or breeding partners, increasing demand for platforms that provide structured data, verification mechanisms, and intelligent matching support.

5. Software Product Vision

For pet owners seeking an easier, more reliable, and modern way to find suitable companions or breeding partners for their pets, **Pawnder** offers a comprehensive solution through an intelligent mobile application. Leveraging the power of artificial intelligence, Pawnder automatically analyzes pet photos to generate highly accurate profiles — including breed identification, physical characteristics, coat colors, and even predicted behavioral traits. Based on this information, the system intelligently suggests and matches pets with high compatibility, taking into account breed, biological factors, personality traits, needs, and geographic proximity.

Unlike traditional methods that depend on word of mouth, scattered online forums, or incomplete and unverified information, Pawnder provides a unified, transparent, and data-driven platform. This helps users save time, build trust, and minimize risks when choosing playmates or breeding partners for their pets. By digitizing and standardizing pet information, Pawnder creates a safe, controlled, and user-friendly environment where pet owners can confidently make decisions backed by accurate insights.

In addition to its product vision, Pawnder aims to build a system architecture that is scalable and adaptable over time. User and pet behavioral data are collected and analyzed to progressively improve the accuracy of the matching system. AI components are designed in a modular manner, allowing models to be replaced or upgraded in the future without affecting the overall system. This approach ensures that Pawnder not only performs effectively in the present, but also maintains a robust technological foundation for long-term development.

Our vision is to create a world where pet lovers are seamlessly connected through intelligent recommendations and trustworthy information—where every pet has access to compatible friends, healthy social interactions, and better opportunities for companionship. With Pawnder, we aim to strengthen the global pet-loving community, fostering a more connected, responsible, and caring ecosystem empowered by smart technology.

6. Project Scope & Limitations

6.1 Major Features

Among the system features, the team focuses on several core functionalities that involve significant technical challenges, including automatic pet profile generation using AI, intelligent matching algorithms, and a multi-role access control system (User – Expert – Admin). The remaining features primarily support user experience and contribute to the overall completeness of the system.

Feature ID	Features Description
General	
FE-01	Authentication: Allows users to register and log in using email and password, and recover forgotten passwords.
Admin Web	
FE-02	Content Management System: Administrators can manage pets, attributes, events, policies, prohibited words, reports, and payments to maintain the quality and security of the platform.
User	
FE-03	AI-powered Pet Profile Creation: Users can automatically create attractive and detailed pet profiles using AI technology based on uploaded photos.
FE-04	AI Chat Assistant: Users can chat with an AI assistant to receive personalized advice on pet care and behavior.
FE-05	Smart Pet Matching and Recommendations: Matches pets based on compatibility weights with swipe interactions and provides AI-based recommendations based on interests and location.
FE-06	Real-time Chat and Matching Management: Users can chat with matched users in real time, manage matchmaking connections, block unwanted users, and seamlessly coordinate pet meetups.
FE-07	Expert Consultations with AI Assessment: Users can consult certified pet experts and submit chats to AI for professional assessments, effectively combining AI with human

	expertise.
FE-08	Appointment Scheduling System: Users can schedule, manage, and track pet meetups and consultations with check-in and completion.
FE-09	Pet Events & Contests: Users can participate in pet contests, submit entries, and vote for pets in community events.
FE-10	Payment & Subscriptions: Users can purchase and manage premium subscription packages for advanced features.
FE-11	Reporting & Safety: Users can report inappropriate content or users and receive real-time notifications to maintain a safe community experience.
Expert	
FE-11	Expert Consultation Management: Experts can manage consultation requests, review AI conversations, chat with users, and provide professional feedback.



Figure 6.1. Major Features

6.2 Limitations & Exclusions

6.2.1 Limitations

ID	Name	Description
LI-01	Virtual Interaction Only	The system supports interactions and connections exclusively within a digital environment. Pawnder does not provide real-world intervention services such as direct pet breeding, veterinary consultation, or medical evaluation for pets.
LI-02	Manual Dependency in Data Verification	Although the system provides mechanisms for standardizing pet profiles, data accuracy remains partially dependent on user-submitted information. Pawnder has not yet implemented third-party data verification mechanisms (e.g., veterinary clinics or professional organizations).

LI-03	Rule-Based Matching Instead of AI-Driven Decision Making	Within the scope of the current project, the matching process is implemented based on predefined rules and criteria. Artificial intelligence is not applied for automated decision-making, adaptive matching, or user behavior optimization.
LI-04	Limited Geographic Availability	Location-based functionalities (e.g., regional matching) are supported only within predefined geographic areas and have not been deployed at a global scale.
LI-05	Online-Only System Operation	The system requires an active internet connection to operate. Offline functionality is not supported in the current version.
LI-06	Species Scope Limitation	The system currently focuses primarily on cats. Other pet species (such as dogs, birds, reptiles, etc.) are not supported in the present implementation.
LI-07	No direct integration with veterinary systems in the current implementation.	The application currently only facilitates connections between pet owners for breeding purposes conducted outside the platform and does not yet provide direct support or services from veterinary clinics or animal health professionals.

6.2.2 Exclusions

ID	Name	Description
EX-01	Language support	Currently, the app is only available in Vietnamese.
EX-02	Offline Support	The application does not support a mobile app
EX-03	Insurance	Pet insurance features are not included
EX-04	Third-party Integrations	No integration with external pet adoption platforms (e.g., Petfinder, BarkBuddy) for automatic data sync.
EX-05	E-commerce	The system does not support buying/selling pet products, accessories, or services.

II. Project Management Plan

1. Overview

1.1 Scope & Estimation

This section provides a structured view of a project's scope, breaking it down into manageable Work Breakdown Structure (WBS) items. Each WBS item represents a specific task or User Story within the project. For enhanced planning and resource allocation, complexity levels (such as simple, medium, and complex) are assigned to these items. Additionally, the estimated effort is quantified in man-days, indicating the anticipated number of working days required for each person to complete each task. Depending on the complexity and the amount of time taken on each task, we have defined 3 levels: simple, medium, and complex.

The definition of each level:

- **Simple:** Tasks that are straightforward, less complex, and require minimal resources and time to complete.
- **Medium:** Tasks with a moderate level of complexity, requiring a reasonable amount of resources and time to complete.
- **Complex:** Complex tasks that require significant resources and time to complete, potentially involving multiple steps and coordination between teams.

1.2 Project Objectives

We define one man-day as 8 hours. Our team works 8 hours per day from Monday to Friday, and 4 hours per day on Saturday and Sunday.

- The average working hours for one person per week is 48 hours (6 man-days).
- Total team capacity: $6 \text{ (man-days)} \times 14 \text{ (weeks)} \times 5 \text{ (members)} = 420 \text{ man-days}$.

#	WBS Item	Complexity	Est. Effort (man-days)
1	Sprint 1: 06/09/2025 - 19/09/2025		70
1.1	Project Initiating		22.5
1.1.1	Sprint planning meeting	Simple	0.5
1.1.2	Define project scope and objectives	Medium	5
1.1.3	Identify stakeholders and requirements	Complex	15
1.1.4	Create project charter	Simple	1
1.1.5	Define project team roles and responsibilities	Simple	1
1.2	Document		6
1.2.1	Prepare functional and customer requirements confirmation documents	Simple	2
1.2.2	Prepare Project Management Plan	Complex	4
1.3	Design		36
1.3.1	Build the web design system	Simple	2
1.3.2	Design web prototypes	Complex	12
1.3.3	Design database	Medium	8
1.3.4	Setup DevOps Google Cloud	Complex	14
1.4	Testing		5
1.4.1	Prepare the test plan	Medium	5
1.5	Sprint Retrospective		0.5

2	Sprint 2: 25/09/2025 - 14/10/2025			70.5
2.1	Organizing a meeting to confirm the functions and project requirements			1
2.2	Document			22
2.2.1	Prepare the SDS document	Complex	5.5	
2.2.2	Prepare the SRS document	Complex	16.5	
2.3	Design			31.5
2.3.1	Finish Web Prototypes	Medium	5.5	
2.3.2	Draw UCD	Medium	5.5	
2.3.3	Draw ERD	Medium	11	
2.3.4	Draw Package Diagram	Medium	5.5	
2.3.5	Define user flow	Medium	2.5	
2.3.6	Update database design	Medium	1.5	
2.4	Coding			12
2.4.1	Build the code base for the back-end user system	Medium	3	
2.4.2	Build the code base for the back-end admin system	Medium	3	
2.4.3	Build the code base for the front-end user system	Medium	3	
2.4.4	Build the code base for the front-end admin system	Medium	3	
2.5	Testing			3
2.5.1	Write UAE test cases	Medium	1.5	
2.5.2	Write ST test cases	Medium	1.5	
2.6	Sprint Retrospective			1
3	Sprint 3: 15/10/2025 - 01/11/2025			39
3.1	Organize a meeting to confirm the project design documents and specifications			1
3.2	Document			12
3.2.1	Update the requirements for the SDS document by comments	Complex	6	
3.2.2	Update the requirements for the SRS document by comments	Complex	6	
3.3	Design			7.5
3.3.1	Update diagrams by comments	Medium	5	
3.3.2	Update Web Prototypes	Medium	2.5	

3.4	Coding			15.5
3.4.1	Implement EPIC 1: Authentication for users(cont.)	Medium	6	
3.4.2	Implement EPIC 1: Authentication for admin(cont.)	Medium	6	
3.4.3	Build shared components for user web app	Medium	1.5	
3.4.4	Build shared components for Admin web app	Medium	2	
3.5	Testing			4
3.5.1	Write UT test cases	Medium	2	
3.5.1	Write UT test cases	Medium	2	
3.6	Sprint Retrospective			1
4	Sprint 4: 03/03/2025 - 16/03/2025			58.5
4.1	Document			9.5
4.1.1	Sprint planning	Simple	0.5	
4.1.2	Finish updating SDS document	Complex	7	
4.1.3	Finish updating SRS document	Complex	2	
4.2	Design			4
4.2.1	Finish updating diagrams	Complex	2	
4.2.2	Finish updating Web Prototypes	Medium	2	
4.3	Development			26.5
4.3.1	Implement user authentication & session management	Medium	1.5	
4.3.2	Implement user profile & address management	Complex	10	
4.3.4	Implement admin user management (view, ban, unban)	Complex	10	
4.3.6	Implement basic user interface pages	Simple	2	
4.3.7	Implement notification feature (send & view notifications)	Medium	3	
4.4	Testing			8
4.4.1	Unit testing for authentication & user features	Simple	2	
4.4.2	Integration testing for core user flows	Simple	2	
4.4.3	System testing for user interface	Simple	4	
4.5	Handover meeting – MVP phase 1			1
4.6	Sprint retrospective			0.5
5	Sprint 5: 17/03/2025 – 30/03/2025			83

5.2	Requirement & Design			1
5.3	Development			55.5
5.3.1	Implement pet/profile management for users	Complex	6	
5.3.2	Implement user preferences & attribute selection	Complex	6	
5.3.3	Implement matching & interaction between users	Complex	6	
5.3.4	Improve notification feature (read, unread status)	Medium	4	
5.3.5	Implement report & complaint feature	Complex	6	
5.3.6	Implement payment & membership feature	Medium	1.5	
5.3.7	Implement admin content management	Medium	1.5	
5.3.8	Complete user workflow based on pets & matching	Complex	7	
5.3.9	Complete profile & matching user interface	Simple	1.5	
5.3.10	Implement appointment & history management	Complex	7	
5.3.11	Implement user activity statistics & badges	Complex	7	
5.3.12	Implement admin attendance & monitoring features	Medium	4	
5.4	Testing			25
5.4.1	Unit testing for user management features	Simple	6	
5.4.2	Integration testing for end-to-end user flows	Simple	3	
5.4.3	System testing for admin functions	Simple	2	
5.4.4	System testing for end-user functions	Simple	4	
5.4.5	Unit testing for payment & notification	Medium	6	
5.4.6	Integration testing for payment processing	Simple	1	
5.4.7	Final system testing for admin & user roles	Simple	3	
5.5	Handover meeting – MVP phase 2			1
5.6	Sprint retrospective			0.5
6	Sprint 6: 31/03/2025 – 13/04/2025			68.5
6.1	Documentation			7
6.1.1	Update design documents for communication features	Medium	1	
6.1.2	Prepare sequence diagrams for chat & consultation	Medium	2	
6.1.3	Prepare class diagrams	Simple	2	
6.1.4	Complete web user guide	Medium	2	

6.2	Development			52
6.2.1	Implement user-to-user communication	Medium	4.5	
6.2.2	Implement consultation between user and expert	Complex	6	
6.2.3	Implement AI-assisted chat support	Complex	6	
6.2.4	Implement expert confirmation & approval workflow	Complex	6	
6.2.5	Implement blocking, reporting & usage limitation	Complex	6	
6.2.6	Implement admin notification management	Complex	6	
6.2.7	Finish updating Web Prototypes	Complex	6	
6.2.8	Implement advanced admin user control	Medium	1	
6.2.9	Complete user chat interface	Simple	1	
6.2.10	Complete expert chat interface	Medium	2	
6.2.11	Complete AI chat interface	Medium	2	
6.2.12	Complete notification interface	Medium	1.5	
6.2.13	Complete admin content management interface	Medium	2	
6.2.14	Finalize admin support tools	Medium	2	
6.3	Testing			7
6.3.1	Unit testing for communication features	Simple	2	
6.3.2	Integration testing for chat workflows	Simple	1	
6.3.3	System testing (user & admin)	Simple	2	
6.3.4	Heuristic usability evaluation	Simple	1	
6.3.5	Prepare acceptance testing document	Simple	1	
6.4	Handover meeting – MVP phase 3			1
6.5	Sprint retrospective			0.5
7	Sprint 7: 14/04/2025 – 27/04/2025			32.5
7.1	Requirement & Design			12.5
7.1.1	Finalize web user guide	Simple	2	
7.1.2	Update WBS & traceability documents	Simple	0.5	
7.1.3	Complete all required project documents	Simple	10	
7.2	Development			4

7.2.1	Fix user-side bugs & usability issues	Simple	2
7.2.2	Fix admin-side bugs & logic issues	Simple	2
7.3	Testing		4
7.3.1	Review & finalize UT, IT, ST, AT documents	Simple	4
7.4	Final handover meeting		1
7.5	Closing Stage		11
7.5.1	System handover & user training	Medium	1
7.5.2	Prepare & rehearse capstone presentation	Medium	10
Total Estimated Effort (man-days)			421.5

1.2.1 Test Phase Results Overview

#	Testing Stage	Test Coverage	No. of Defects	% of Defect	Notes
1	Unit Test	95%	0	0%	Written and run by software developers to ensure that a section of an application (known as the "unit") meets its design and behaves as intended.
2	Integration Test	100%	3	2.8%	Individual software modules are combined and tested as a group.
3	System Test	100%	0	0%	Performed on the entire system, tests not only the design but also the behavior and even the customer's expectations.
4	Acceptance Test	100%	8	27.59%	System users perform tests in line with what would occur in real-life scenarios.

1.2.2 Ensure implementation on schedule with allowable resources

- Sprint Timeliness Rate(%): 90%
- Allocated Effort: 3024 hours - 420 man-days

#	Members	Weekdays	Weekends
1	Nguyen Van Duc	6 hours	8 hours
2	Nguyen Minh Sang	6 hours	8 hours

3	Nguyen Ta Tri Cuong	6 hours	8 hours
4	Le Quoc Tuan	6 hours	8 hours
5	Doan Quoc Viet	6 hours	8 hours

1.3 Project Risks

#	Risk Description	Impact	Possibility	Response Plans
1	Team member injury or illness	Very High	Low	Activate emergency response: reassign tasks, reduce sprint workload, and support affected members.
2	Team members lacking time due to academic commitments	High	High	Set a clear time commitment from the start, daily progress updates, and ensure minimum working hours.
3	Incorrect cost estimation (Cloud, domain, ID photo storage, etc.)	Medium	High	Prepare a detailed budget, review regularly, and use free GCP services where possible.
4	Technical difficulties with .NET, React, GCP, etc.	Low	Medium	Internal knowledge sharing, self-study with official docs, and peer support within the team.
5	Loss of user data or system failure	Very High	Low	Set up automated backups, and regularly test backup/restore functions to ensure data safety.
6	API have error	Very High	Medium	Ask the instructor for advice

2. Management Approach

The Pawnder project will follow the Agile Scrum methodology, ensuring flexibility, transparency, and continuous improvement. Development will be organized into short sprints, with each sprint delivering functional features such as cat profiles, matchmaking, AI-based predictions, and social sharing.

Regular Scrum ceremonies—planning, daily stand-ups, reviews, and retrospectives—will promote strong collaboration among developers, designers, AI specialists, and product owners. This approach enables quick adaptation to feedback, early delivery of value, and consistent alignment with user needs.

2.1 Project Process

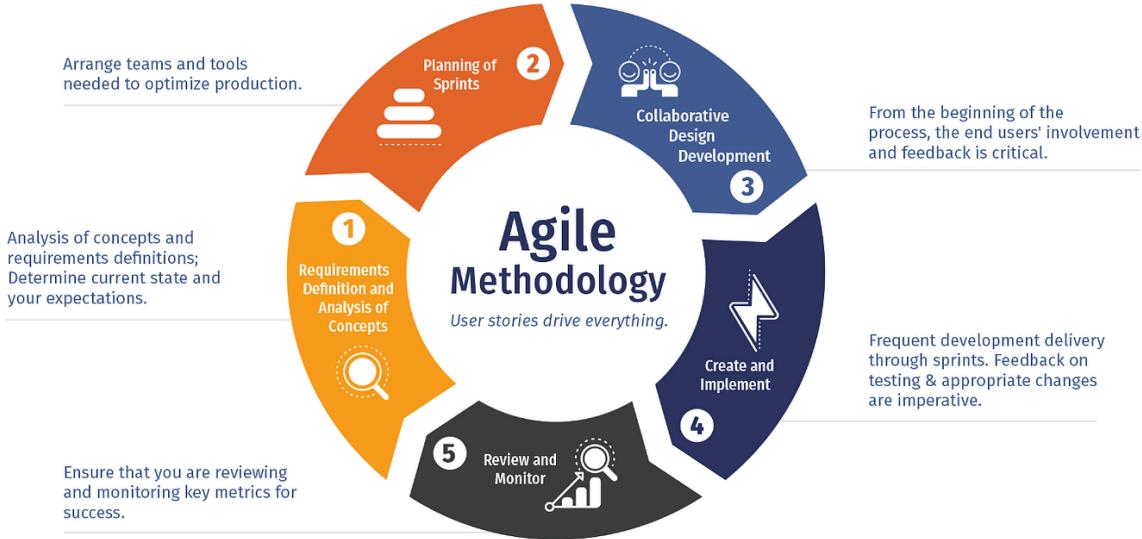


Figure 2.1: Agile Scrum methodology

The project is developed following the Agile Scrum methodology, which emphasizes flexibility, collaboration, and continuous improvement throughout the software development lifecycle. This approach allows the team to adapt quickly to changing requirements while ensuring consistent delivery of value to end users.

The process begins with Requirements Definition and Analysis, where business needs and user expectations are clearly identified and translated into user stories. Based on these requirements, the team conducts Sprint Planning to organize tasks, allocate resources, and define sprint goals.

Development is carried out through collaborative design and development, ensuring close cooperation among team members and active involvement from stakeholders. Features are then created and implemented incrementally within short, time-boxed iterations called sprints, typically lasting two weeks. Each sprint results in a potentially shippable product increment.

Finally, the team continuously reviews and monitors progress, evaluating performance metrics and gathering feedback to improve product quality and team efficiency. This iterative process ensures continuous enhancement, transparency, and alignment with user needs.

2.2 Quality Management

To ensure high-quality delivery of the **Pawnder** project, the team will apply a structured quality management approach throughout the development lifecycle. Quality objectives will be achieved through the following practices:

- **Defect Prevention:** The team will follow coding standards, naming conventions, and best practices to minimize defects from the start. Regular knowledge-sharing sessions and documentation will help reduce misunderstandings and improve consistency.

- **Reviewing:** Code reviews and design walkthroughs will be conducted at each sprint to identify issues early. This process also ensures alignment with requirements and promotes knowledge transfer within the team.
- **Unit Testing:** Each module, such as profile creation, matchmaking, and AI prediction, will undergo unit testing. This ensures components work correctly in isolation before moving to the next stage.
- **Integration Testing:** After unit testing, the system will validate how modules interact, focusing on data flow and functionality across features. This step ensures the app functions smoothly as a cohesive platform.
- **System Testing:** End-to-end testing will be performed to verify performance, usability, and reliability from the user's perspective. This guarantees that the final product meets both functional and non-functional requirements before release.

2.3 Training Plan

Training Area	Participants	When, Duration	Waiver Criteria
.NET	All member	Week 1 – 2 sessions (2 hours each)	Mandatory
React Native	All member	Week 2 – 3 sessions (2 hours each)	Mandatory
Git, Github	All member	Week 1 – 1 session (2.5 hours)	Mandatory
Visual Paradigm	All member	Week 2 – 3 sessions (1.5 hours each)	Mandatory
Testing (Unit & UAT)	All member	Week 4 – 2 sessions (2 hours each)	Mandatory

3. Project Deliverables

#	Deliverable	Due Date	Notes
1	SEP490_G151_Report1_ProjectIntroduction.docx	15/09/2025	Project introduction
2	SEP490_G151_Report2_ProjectManagementPlan.docx	22/09/2025	Project Management Plan
3	SEP490_G151_Report3_SoftwareRequirementSpecification.docx	06/10/2025	Software Requirement Specification (SRS)
4	SEP490_G151_Report4_SoftwareDesignSpecification.docx	20/10/2025	Software Design Specification (SDS)
5	SEP490_G151_Report5_TestDocumentation.docx	10/11/2025	Test Documentation

6	SEP490_G151_Report6_SoftwareUserGuides.docx	24/11/2025	Software User Guides
7	SEP490_G151_Report7_FinalProjectReport.docx	15/12/2025	Final Project Report

4. Responsibility Assignments

D~Do; R~Review; S~Support; I~Informed; <blank>- Omitted

Responsibility	SangNM	DucNV	CuongNTT	TuanLQ	VietDQ
Project Planning & Tracking	S	D	R	R	S
Prepare Project Introduction Document	S	S	D	R	S
Prepare SRS Document (Overview Part)	R	D	S	S	S
Prepare SRS Document (User Requirements)	D	R	S	S	S
Prepare SRS Document (Functional Requirements)	D	S	R	S	S
Prepare SRS Document (Non-Functional Requirements)	R	D	S	S	S
Prepare SRS Document (Use Case Diagrams)	S	R	D	S	S
Prepare SRS Document (Activity Diagrams)	R	S	D	S	S
Prepare SRS Document (System Models)	S	R	S	D	S
Prepare SDD Document (Architecture Design)	R	D	S	S	S
Prepare SDD Document (Database Design)	S	R	D	S	S

Prepare SDD Document (UI/UX Design)	R	S	D	S	S
Frontend	S	R	R	D	D
Backend	D	D	D	R	R
Integration & Testing	S	R	S	D	S
Deployment	R	S	D	S	S
Prepare Final Report	D	R	S	S	R
Prepare Final Presentation	R	S	D	S	R

5. Project Communications

Communication Item	Who/ Target	Purpose	When, Frequency	Type, Tool, Method(s)
Daily Stand-up Meeting	Entire team (developers, testers, product owner)	Update progress, share issues, and assign daily tasks	Daily, 15 minutes (every morning)	Face-to-face / Google Meet; task notes recorded in Jira
Sprint Planning Meeting	Product owner, Scrum master, development team	Define sprint goals, select and estimate backlog items	At the beginning of each sprint	Online meeting (Google Meet); backlog managed in Jira
Sprint Review Meeting	Development team, product owner, stakeholders	Demonstrate completed features and collect feedback	At the end of each sprint	Online presentation; feedback documented in Jira
Sprint Retrospective	Development team, Scrum master	Evaluate sprint performance and identify improvements	At the end of each sprint	Team discussion via Google Meet; action items recorded
Backlog Refinement	Product owner, development team	Clarify and prioritize product backlog items	Once per sprint (or as needed)	Jira backlog review; online discussion

Technical Discussion / Design Meeting	Developers, technical lead	Discuss system design and technical solutions	As needed	Google Meet / Chat; diagrams on Draw.io
---------------------------------------	----------------------------	---	-----------	---

6. Configuration Management

6.1 Document Management

We use Google Drive to manage project documents and their changes/versions, such as the image below:

Name	Owner	Date modified	File size
DOCS	me	19 Sept	—
Information and Task	Nguyen Minh...	13:42	9 KB
Giới thiệu về sản phẩm	Nguyen Minh...	18:16	779 KB

Figure 6.1: Document Management

Changes are tracked and different versions are managed using Google Drive, which allows us to manage project documents and their changes/versions efficiently. Each document's history is automatically saved, enabling us to view and restore previous versions if needed. This ensures that all changes are documented, and different versions are easily accessible for review and comparison.

6.2 Source Code Management

The source code will be managed by the team on GitHub:

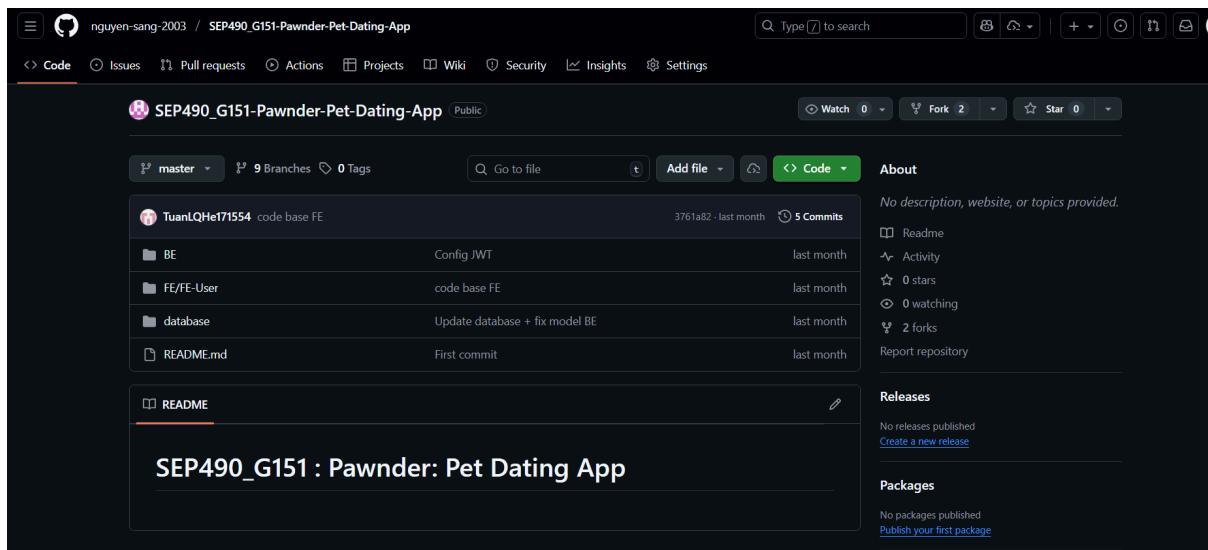


Figure 6.2: Source Code Management

For the branching strategy, we use the Gitflow branching strategy. This involves creating separate branches for features, releases, and hotfixes.

Code reviews will be conducted using pull requests. When a developer completes a feature or bug fix, they will create a pull request to merge their changes into the develop branch. The pull request will be reviewed by at least one other team member, who will check for code quality, adherence to coding standards, and potential issues. Feedback will be provided, and changes may be requested before the pull request is approved and merged.

The release management process involves the following steps:

- **Release Planning:** Identify the features and fixes to be included in the release.
- **Creating a Release Branch:** Create a release branch from the develop branch.
- **Final Testing and Bug Fixes:** Perform thorough testing and address any issues found.
- **Release Approval:** Obtain approval from stakeholders for the release.
- **Merging and Tagging:** Merge the release branch into the master branch and tag the release with a version number.
- **Deployment:** Deploy the new release to the production environment.
- **Post-Release Monitoring:** Monitor the release for any issues and address them as needed.

6.3 Tools & Infrastructures

Category	Tools / Infrastructure
Technology	React (FrontEnd), React Native(App), DotNet(BackEnd)
Database	PostgreSQL, Firebase
IDEs/Editors	Visual Studio Code, VS Code
Diagramming	DrawIO, Visual paradigm
Documentation	Ms Office, Google Docs/Sheets/Slides
Version Control	GitHub(Source Codes), Google Drive (Documents)
Deployment server	Auze cloud

III. Software Requirement Specification

1. Product Overview

Pawnder is a software platform designed to help pet owners manage pet profiles, find suitable companions for play or breeding, and connect with other owners through structured matching and communication features. The system addresses common challenges such as incomplete or inaccurate pet information, difficulties in locating compatible pets, and the lack of transparent and reliable digital tools to support informed decision-making.

Main Objectives:

- **Accurate Pet Profiles:** Users can upload pet images, and the system supports AI-assisted recognition of pet attributes such as breed estimation, age estimation, and observable traits, reducing manual input and improving profile consistency.
- **Smart Matching:** Pawnder suggests suitable pets for friendship or breeding based on predefined criteria, including breed, age, gender, location, and temperament compatibility.
- **Safety & Transparency:** Users can block unwanted interactions, report inappropriate behavior, and review system policies prior to accessing core features.
- **Premium Services:** Users may upgrade to VIP status via secure QR-based payment methods to access additional benefits, including increased daily interaction limits, extended AI usage quotas, and additional expert confirmation requests.
- **AI Chat with Expert Validation:** Users may interact with an AI-assisted chat feature. When required, AI-generated responses can be reviewed and validated by a designated Expert before being delivered to the user.
- **Pet Appointments:** Users can schedule meetups at pet-friendly locations with matched pet owners, including location-based check-in verification using GPS and geocoding services..
- **Pet Events:** Users can participate in photo and video contests, submit entries, vote for favorites, and view leaderboards with prize distribution information.

User Roles:

- **Pet Owners (Mobile App):** Pet Owners can register and authenticate accounts, manage personal and pet profiles, upload pet images for AI-assisted recognition, search and match with other pets, communicate with other owners and experts, schedule pet appointments, participate in events, report violations, use AI chat features, block or unblock users, review and accept system policies, receive notifications, and purchase or renew VIP services via QR-based payment.
- **Administrators (Web Portal):** Administrators manage user accounts, monitor service usage and history, handle user reports, configure system notifications, manage shared attributes (e.g. traits), moderate content (e.g., inappropriate language filtering), manage policies and policy versions, create and manage pet events, and monitor the queue and status of expert confirmation requests.
- **Experts (Web Portal):** Validate AI-generated responses (e.g., Chat AI outputs) and provide confirmation or rejection. Expert decisions are logged in the system and can be monitored by administrators. Experts can also directly chat with users for complex consultations.

External Integrations:

- **Payment Gateways (VietQR, Sepay):** For secure QR code transactions and subscription purchases with automated payment verification.
- **AI Services (Gemini AI):** For pet image recognition, intelligent matching, and conversational chat assistance.
- **Location Services (LocationIQ):** Used to support reverse geocoding and location-based features for pet appointments and check-in verification.
- **Note:** AI-based functions (e.g., image recognition, matching support, and chat assistance) are treated as internal system processing at the context level. In accordance with academic modeling requirements, the AI engine is not represented as a separate external actor in the context diagram.

The context diagram below illustrates the boundary of Pawnder and the data flows between external entities and the system for release 1.0.

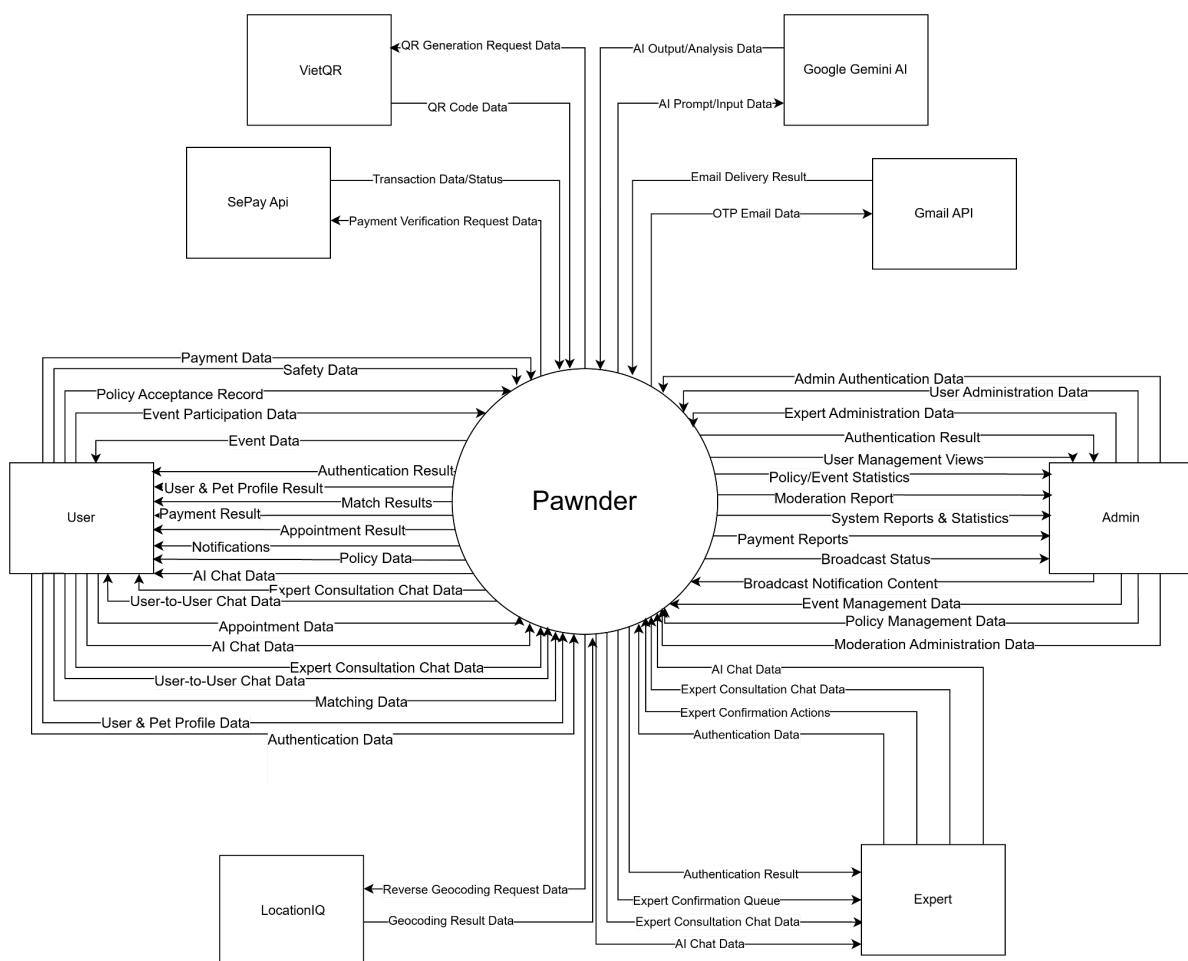


Figure 1. Context diagram of the Pawnder system

2. User Requirements

2.1 Actors

#	Actor	Description
1	User	The primary user of the mobile application. They register, create and manage pet profiles, search for suitable matches for dating or breeding, use chat features, and can purchase advanced services (VIP, Premium).
2	Admin	The administrator uses the web portal to manage accounts, pets, system rules, reports, services, notifications, and approve expert registrations.
3	Expert	A veterinary professional who registers a personal profile to provide online consultation services to users.
4	Payment Gateway	An external system (VietQR) that securely processes financial transactions and supports payment for user services.
5	AI Engine	An external tool (Google Gemini) that supports pet breed recognition from images and provides AI-powered chat assistance.
6	Notification System	An external component (Email Service/FCM) responsible for sending notifications (push/email) to users regarding match requests, messages, or system alerts.
7	Cloud Storage	An external service (Cloudinary) used for storing and retrieving user-uploaded media files such as pet photos.

2.2 Use Cases

2.2.1 Diagram(s)

2.2.1.1 Overview Use Case Diagram

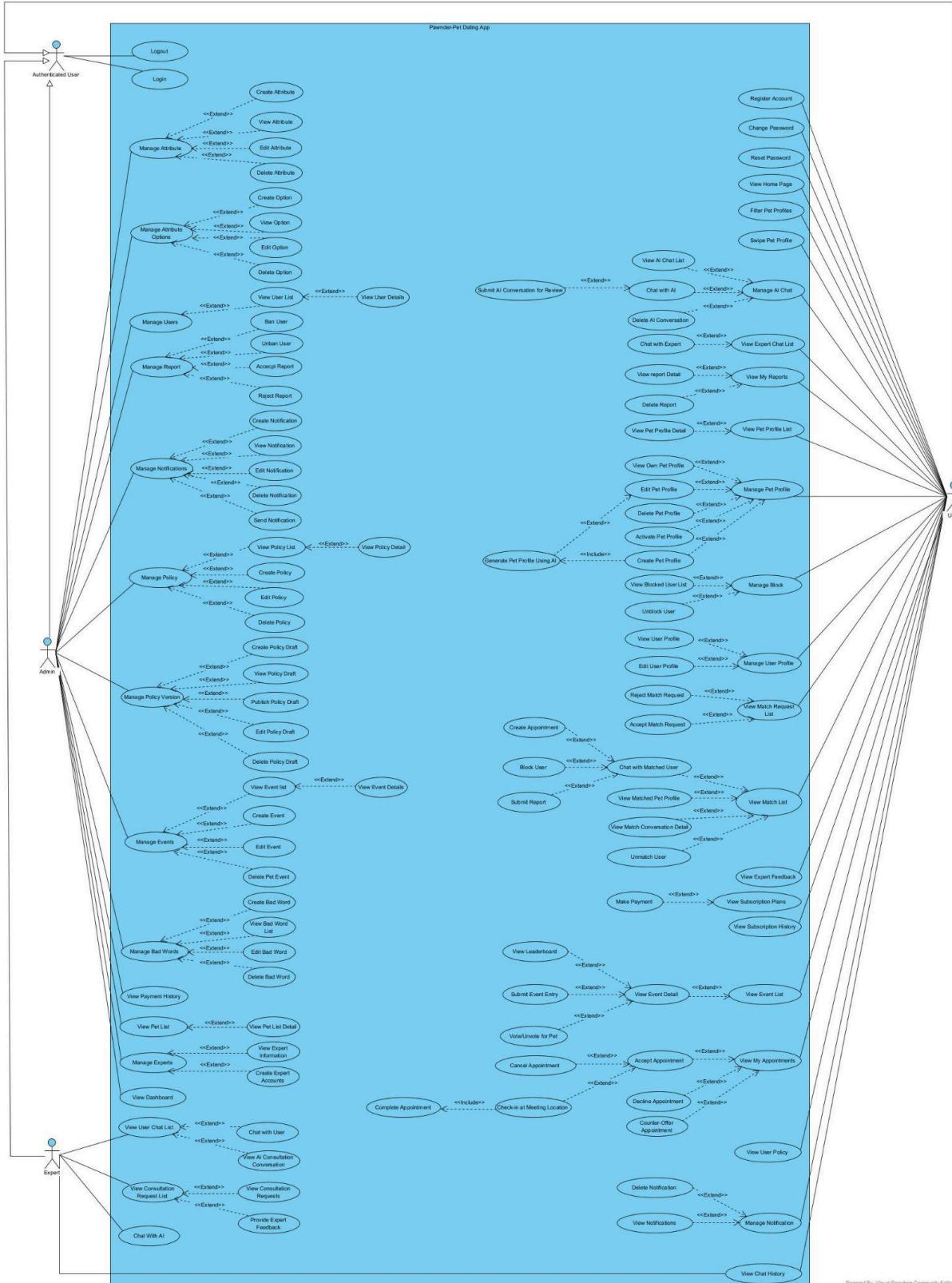


Figure 2.2.1.1: Overview Use Case Diagram

2.2.1.2 Detail User Diagram

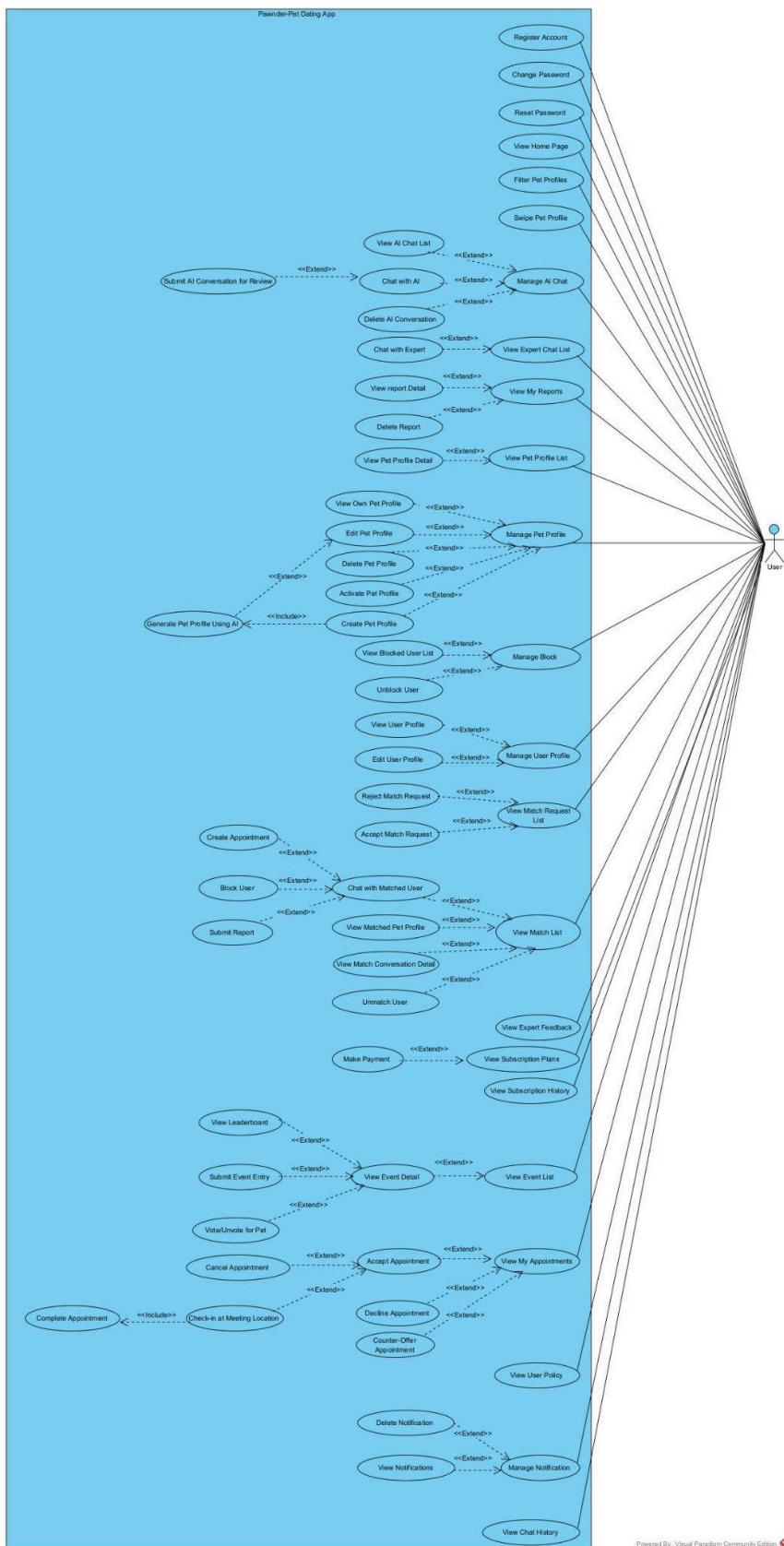


Figure 2.2.1.2: Detail User Diagram

2.2.1.2.1 Detail User Description:

ID	Use Case	Actors	Use Case Description
UC-01	Login	User, Expert, Admin	Allows users to log into the system by verifying credentials and redirecting them to the appropriate homepage based on role.
UC-02	Logout	User, Expert, Admin	Allows users to securely log out and terminate their active session.
UC-03	Register Account	User	Allows new users to create an account by providing required personal and login information.
UC-04	Reset Password	User	Allows users to reset their password by verifying their identity.
UC-05	Change Password	User	Allows users to change their current password to enhance security.
UC-06	View Home Page	User	Allows users to view the main homepage displaying featured pets, events, and updates.
UC-07	Filter Pet Profiles	User	Allows users to filter pet profiles based on preferences for matching.
UC-08	View Pet Profile Detail	User	Allows users to view detailed information of a selected pet profile.
UC-09	Swipe Pet Profile	User	Allows users to swipe pet profiles to like or skip matching opportunities.
UC-10	View AI Chat List	User	Allows users to view a list of AI chat conversations.
UC-11	Chat with AI	User	Allows users to interact with an AI assistant for pet-related consultation.
UC-12	Delete AI Conversation	User	Allows users to delete an AI conversation from their history.
UC-13	Submit AI Conversation for Review	User	Allows users to submit AI conversations for expert review.
UC-14	View Expert Feedback	User	Allows users to view expert feedback on submitted AI conversations.
UC-15	View Expert Chat List	User	Allows users to view a list of conversations with experts.

UC-16	Chat with Expert	User	Allows users to chat directly with experts for consultation.
UC-17	View Policy	User	Allows users to view published system policies.
UC-18	Accept Policy	User	Allows users to accept system policies before using system features.
UC-19	View My Appointments	User	Allows users to view their appointment list and details.
UC-20	Create Appointment	User	Allows users to create an appointment with matched users.
UC-21	Accept Appointment	User	Allows users to accept an appointment invitation.
UC-22	Decline Appointment	User	Allows users to decline an appointment invitation.
UC-23	Cancel Appointment	User	Allows users to cancel an existing appointment.
UC-24	Check-in at Appointment	User	Allows users to check in upon arrival at the meeting location.
UC-25	Complete Appointment	User	Allows users to mark an appointment as completed.
UC-26	Counter Offer Appointment	User	Allows users to delete notifications.
UC-26	View My Reports	User	Allows users to view their submitted reports and report details.
UC-27	Submit Report	User	Allows users to submit reports about inappropriate behavior or issues.
UC-28	View Subscription Plans	User	Allows users to view available subscription plans and VIP status.
UC-29	Make Payment	User	Allows users to make payments for subscriptions via QR code.
UC-30	View Subscription History	User	Allows users to view their payment history.
UC-31	View Own Pet Profile	User	Allows users to view their own pet profiles.

UC-32	Create Pet Profile	User	Allows users to create a new pet profile manually.
UC-33	Generate Pet Profile Using AI	User	Allows users to generate a pet profile automatically using AI.
UC-34	Edit Pet Profile	User	Allows users to edit existing pet profiles.
UC-35	Delete Pet Profile	User	Allows users to delete pet profiles.
UC-36	Set Active Pet	User	Allows users to set a pet as the active profile for matching.
UC-37	Block User	User	Allows users to block other users.
UC-38	View Blocked User List	User	Allows users to view their blocked user list.
UC-39	Unblock User	User	Allows users to unblock a user.
UC-40	View Match List	User	Allows users to view their list of matched users or pets.
UC-41	View Match Conversation Detail	User	Allows users to view conversation details with a matched user.
UC-42	Chat with Matched User	User	Allows users to chat with matched users.
UC-43	Respond to Match Request	User	Allows users to accept or reject a received match request.
UC-44	Unmatch User	User	Allows users to remove an existing match.
UC-45	View Event List	User	Allows users to view a list of pet events.
UC-46	View Event Detail	User	Allows users to view detailed information and leaderboard of an event.
UC-47	Submit Event Entry	User	Allows users to submit their pets to events.
UC-48	Vote / Unvote for Pet	User	Allows users to vote or remove votes for pets in events.
UC-49	View User Profile	User	Allows users to view their user profile.
UC-50	Edit User Profile	User	Allows users to edit their profile information.

UC-51	View Notifications	User	Allows users to view system notifications.
UC-52	Delete Notification	User	Allows users to delete notifications.

2.2.1.3: Detail Expert Diagram

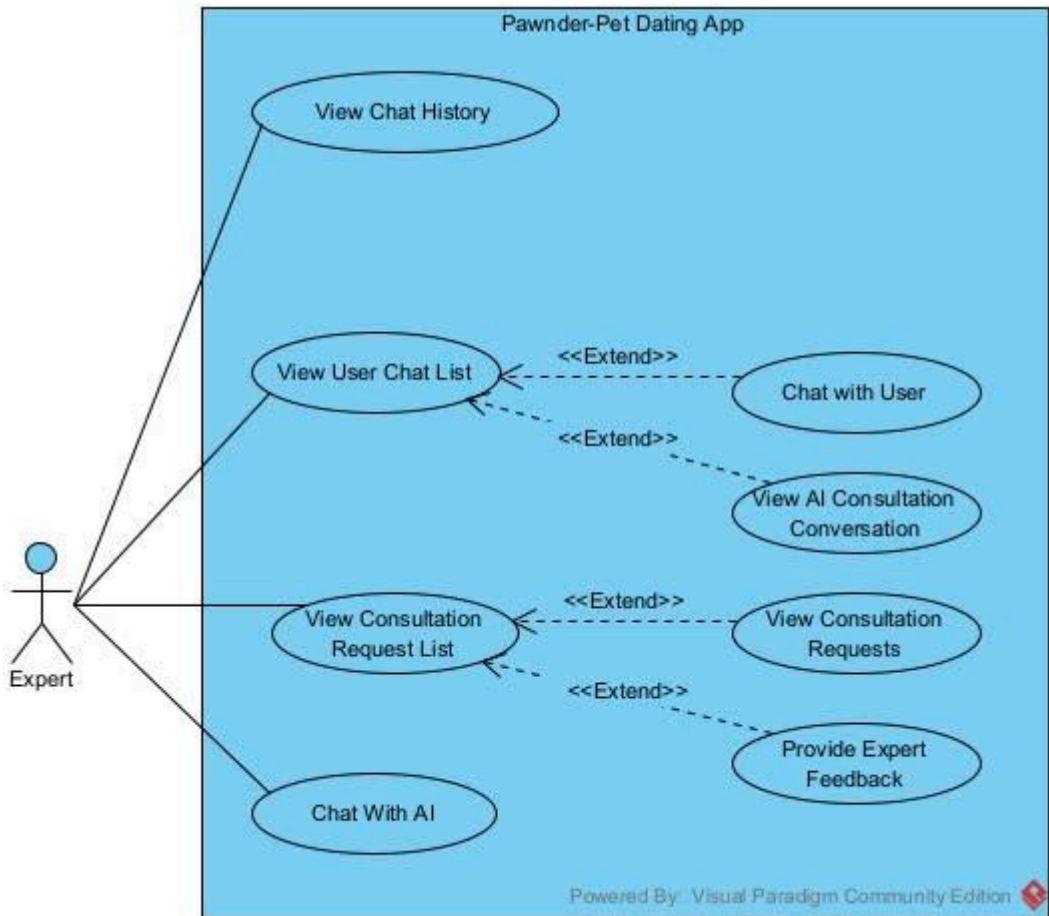


Figure 2.2.1.3: Detail Expert Diagram

2.2.1.3.1 Detail Expert Description:

ID	Use Case	Actors	Use Case Description
UC-53	View Consultation Request List	Expert	Allows experts to view consultation requests from users.
UC-54	Provide Expert Feedback	Expert	Allows experts to confirm consultation requests and provide feedback.
UC-55	View AI Consultation Conversation	Expert	Allows experts to review AI conversations submitted by users.

UC-56	Chat with AI	Expert	Allows experts to continue AI conversations for consultation.
UC-57	View User Chat List	Expert	Allows experts to view user chat lists.
UC-58	Chat with User	Expert	Allows experts to chat directly with users.
UC-59	View Chat History	Expert	Allows experts to view previous chat conversations with users for reference and follow-up purposes.
UC-60	Chat With AI	Expert	Allows experts to interact with the AI chatbot to get assistance or information about the platform.

2.2.1.4: Detail Admin Diagram

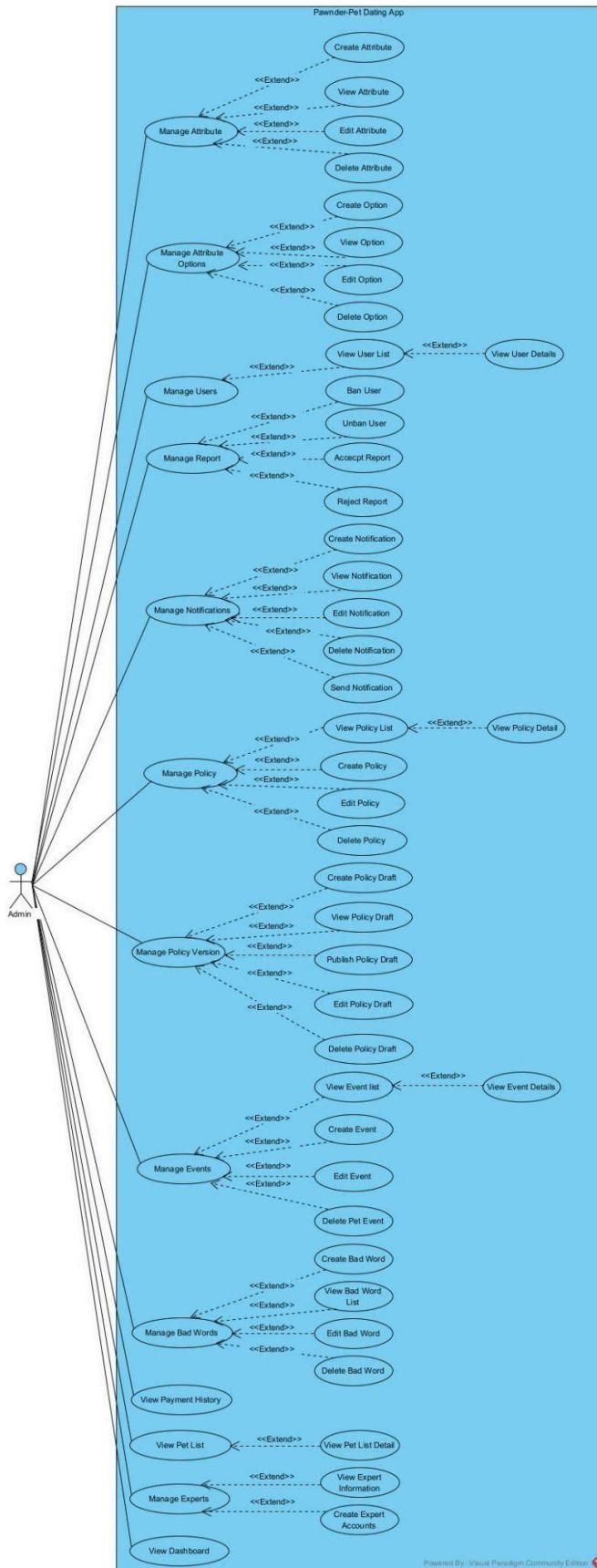


Figure 2.2.1.4: Detail Admin Diagram

2.2.1.4.1 Detail Admin Description:

ID	Use Case	Actors	Use Case Description
UC-61	View User List	Admin	Allows Admin to view all registered users.
UC-62	View User Detail	Admin	Allows Admin to view detailed user information.
UC-63	Ban User	Admin	Allows Admin to ban users for policy violations.
UC-64	Unban User	Admin	Allows Admin to restore access for banned users.
UC-65	Create Attribute	Admin	Allows Admin to create system attributes.
UC-66	View Attribute List	Admin	Allows Admin to view all attributes.
UC-67	Edit Attribute	Admin	Allows Admin to edit attributes.
UC-68	Delete Attribute	Admin	Allows Admin to delete attributes.
UC-69	View Expert Information	Admin	Allows Admin to view expert account information.
UC-70	Create Expert Account	Admin	Allows Admin to create new expert accounts.
UC-71	Create Option	Admin	Allows Admin to create system options.
UC-72	View Option List	Admin	Allows Admin to view options.
UC-73	Edit Option	Admin	Allows Admin to edit options.
UC-74	Delete Option	Admin	Allows Admin to delete options.
UC-75	Accept Report	Admin	Allows Admin to accept reported cases.
UC-76	Reject Report	Admin	Allows Admin to reject invalid reports.
UC-77	Create Event	Admin	Allows Admin to create pet events.
UC-78	View Event List	Admin	Allows Admin to view pet events.
UC-79	View Event Detail	Admin	Allows Admin to view event details.
UC-80	Edit Event	Admin	Allows Admin to edit events.
UC-81	Delete Event	Admin	Allows Admin to delete events.

UC-82	Create Notification	Admin	Allows Admin to create notifications.
UC-83	Send Notification	Admin	Allows Admin to send notifications to users.
UC-84	View Notification	Admin	Allows Admin to view notifications.
UC-85	Edit Notification	Admin	Allows Admin to edit notifications.
UC-86	Delete Notification	Admin	Allows Admin to delete notifications.
UC-87	Create Bad Word	Admin	Allows Admin to add prohibited chat words.
UC-88	View Bad Word List	Admin	Allows Admin to view prohibited words.
UC-89	Edit Bad Word	Admin	Allows Admin to edit prohibited words.
UC-90	Delete Bad Word	Admin	Allows Admin to delete prohibited words.
UC-91	Create Policy	Admin	Allows Admin to create a policy category.
UC-92	View Policy List	Admin	Allows Admin to view all policies.
UC-93	View Policy Detail	Admin	Allows Admin to view policy details.
UC-94	Edit Policy	Admin	Allows Admin to edit policies.
UC-95	Delete Policy	Admin	Allows Admin to delete policies.
UC-96	Create Policy Draft	Admin	Allows Admin to create a draft version of a policy.
UC-97	View Policy Draft	Admin	Allows Admin to view policy drafts.
UC-98	Edit Policy Draft	Admin	Allows Admin to edit policy drafts.
UC-99	Delete Policy Draft	Admin	Allows Admin to delete policy drafts.
UC-100	Publish Policy Draft	Admin	Allows Admin to publish policy drafts for user access.

UC-101	View Payment History	Admin	Allows Admin to view all payment transactions.
UC-102	View Pet list	Admin	Allows Admin to view all registered pets.
UC-103	View Pet List Detail	Admin	Allows Admin to view detailed pet information.

3. Functional Requirements

3.1 System Functional Overview

3.1.1 Screens Flow

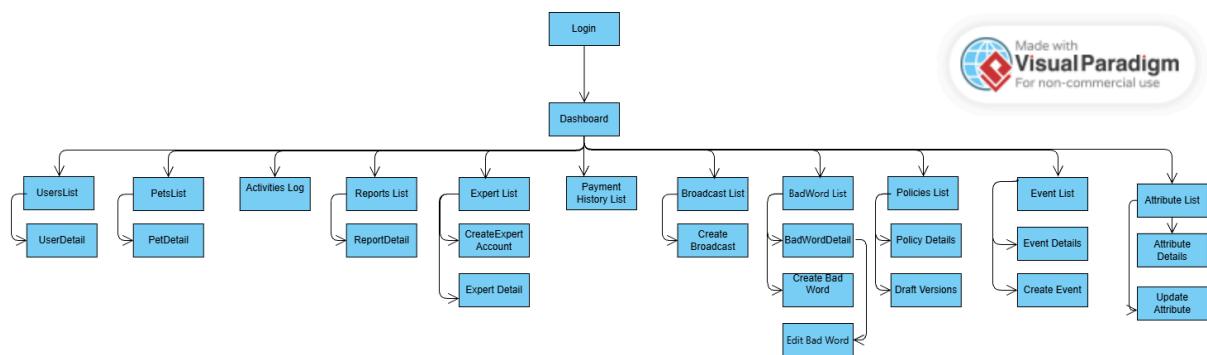


Figure 3.1.1.1. Admin Screen Flow

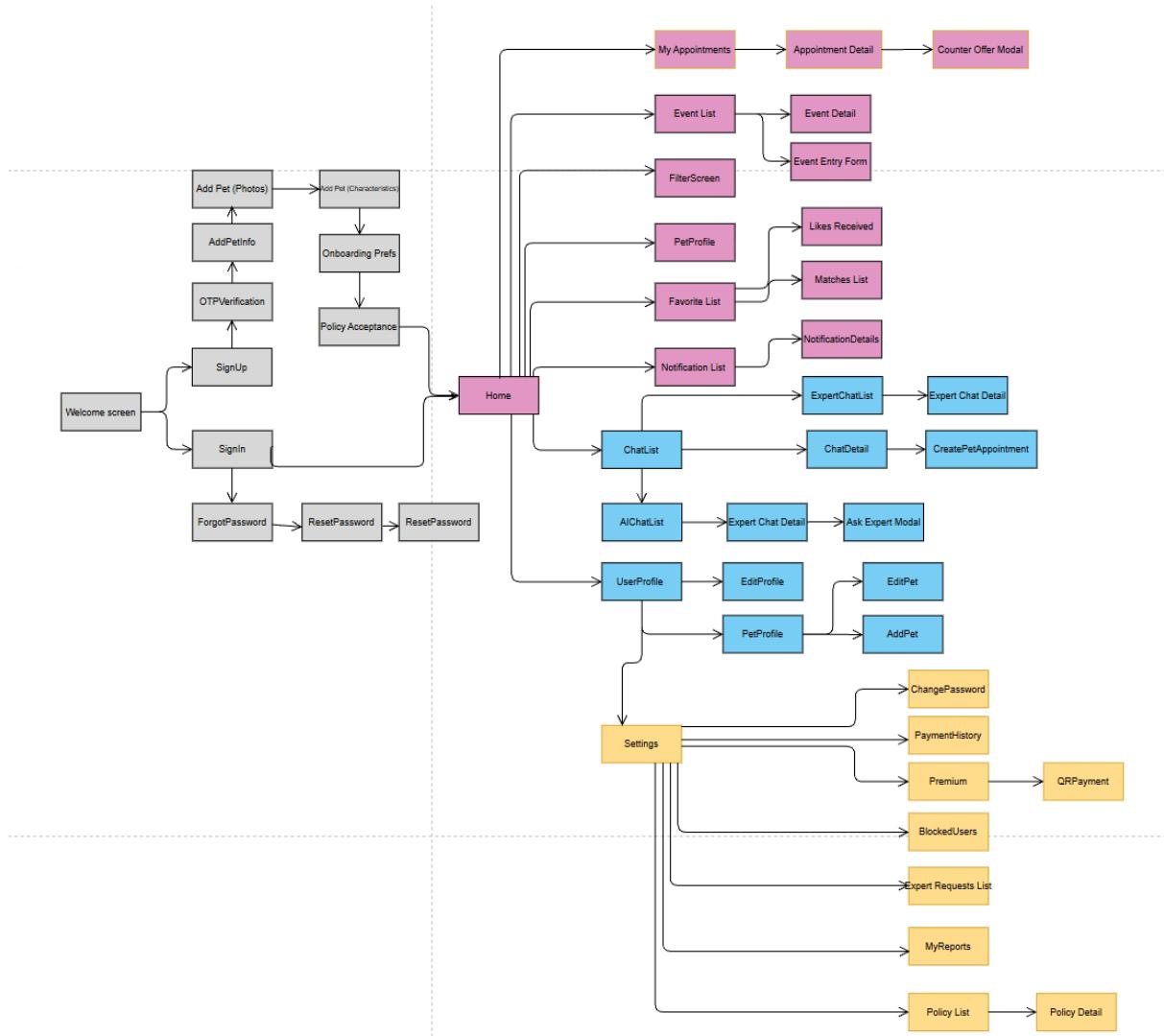


Figure 3.1.1.2. User Screen Flow

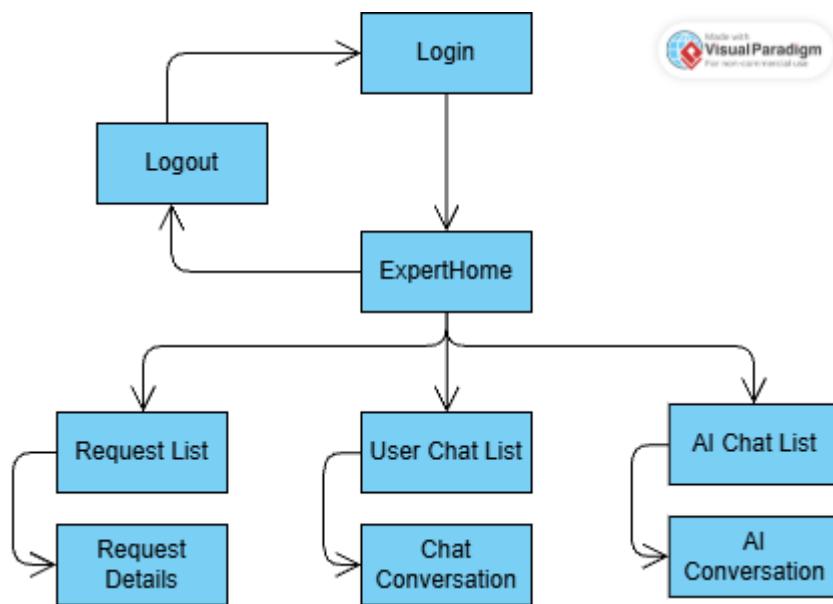


Figure 3.1.1.3. Expert Screen Flow

3.1.2 Screen Descriptions

3.1.2.1 User Application Screens (Mobile - React Native)

#	Feature	Description
1	SignUp	Enables new users to create an account by providing personal information, email, and password
2	OTPVerification	Allows users to verify their account using a One-Time Password sent to their email
3	AddPetInfo	Guides new users to add their first pet profile after successful registration
4	SignIn	Allows users to enter email and password to access the application
5	ForgotPassword	Allows users to request a password reset via OTP sent to their email
6	ResetPassword	Enables users to create a new password after OTP verification
7	Home	Displays the main dashboard with pet matching cards and quick access to all features
8	PetEvent	Shows a list of pet-related events that users can participate in
9	SendPetImage	Allows users to upload and submit pet images for event participation
10	VotePet	Enables users to vote for their favorite pets in event competitions
11	FilterScreen	Allows users to set preferences and filters for pet matching criteria
12	PetProfile	Displays detailed information about a pet including photos, attributes, and owner info
13	Favorite	Shows a list of pets that the user has liked or added to favorites

14	Notification	Displays a list of notifications including matches, messages, and system alerts
15	NotificationDetails	Shows detailed information about a specific notification
16	ExpertChatList	Displays a list of available pet experts for consultation
17	ExpertChat	Enables real-time chat communication with pet experts for advice
18	ChatList	Shows a list of all active chat conversations with matched users
19	ChatDetail	Displays the full conversation history and enables messaging with a matched user
20	CreatePetAppointment	Allows users to schedule a pet meetup appointment with matched users
21	AIChatList	Displays a list of AI chat sessions for pet-related queries
22	AIChat	Enables users to chat with AI assistant for pet care advice and information
23	ExpertConfirm	Shows AI responses pending expert confirmation for accuracy verification
24	UserProfile	Displays the user's personal profile information and their pets
25	EditProfile	Allows users to modify their personal information and profile settings
26	EditPet	Enables users to update their pet's information and photos
27	AddPet	Allows users to add a new pet profile to their account
28	Settings	Provides access to application settings and account management options
29	ChangePassword	Allows users to update their account password
30	PaymentHistory	Displays a history of all payment transactions made by the user
31	Premium	Shows premium subscription plans and VIP features available for purchase
32	QRPayment	Enables users to complete premium subscription payment via QR code

33	BlockedUsers	Displays a list of users that have been blocked by the current user
34	MyReports	Shows a list of reports submitted by the user regarding issues or violations
35	Policy	Provides access to view platform policies and terms of service
36	ViewListPolicy	Displays a list of all available policies and guidelines
37	ViewPolicyDetails	Shows the complete content of a specific policy document
38	AcceptListPolicyDetails	Allows users to review and accept updated policy terms
39	PetAppointment	Displays a list of scheduled pet meetup appointments
40	ViewPetAppointment	Shows detailed information about a specific pet appointment
41	EditPetAppointment	Allows users to modify or reschedule an existing pet appointment

3.1.2.2 Admin Portal Screens (Web - React)

#	Feature	Description
42	AdminLogin	Allows administrators to enter email and password to access the admin dashboard
43	Dashboard	Displays an overview of key metrics, statistics, and quick access to all management modules
44	ManageChatUser	Provides access to manage and monitor users participating in the chat system
45	ViewChatUser	Displays detailed information about chat users including their activity and chat history
46	ManageChatFilterRules	Provides tools to create, view, edit, and delete chat filter rules for content moderation
47	CreateChatFilterRules	Allows administrators to create new chat filter rules to block inappropriate content
48	ViewChatFilterRules	Displays a list of all existing chat filter rules with their configurations
49	EditChatFilterRules	Enables administrators to modify existing chat filter rules

50	DeleteChatFilterRules	Allows administrators to remove unwanted chat filter rules from the system
51	ReloadCacheChatFilterRules	Refreshes the cached chat filter rules to apply recent changes immediately
52	ManageEvent	Provides tools to manage pet events including creating, editing, and deleting events
53	CreatePetEvent	Allows administrators to create new pet-related events for users to participate
54	EditPetEvent	Enables administrators to modify details of existing pet events
55	DeletePetEvent	Allows administrators to remove pet events from the system
56	ViewPetEvent	Displays detailed information about a specific pet event
57	ManageUsers	Provides access to view user details and manage user accounts including banning
58	ViewDetails	Displays comprehensive information about a specific user account
59	BanUsers	Allows administrators to ban or suspend user accounts that violate platform policies
60	ManageExpert	Provides tools to manage expert accounts including creation and viewing expert information
61	CreateExpertAccount	Allows administrators to create new expert accounts for pet specialists
62	ViewExpertList	Displays a list of all registered experts on the platform
63	ViewExpertInformation	Shows detailed profile information of a specific expert
64	ManagePet	Provides access to view and manage pet profiles registered on the platform
65	ViewPet	Displays detailed information about a specific pet profile including owner and attributes

66	ManagePolicy	Provides tools to create, view, edit, and delete platform policies and guidelines
67	CreatePolicy	Allows administrators to create new platform policies or terms of service
68	ViewPolicy	Displays the content of existing platform policies
69	EditPolicy	Enables administrators to modify existing platform policies
70	DeletePolicy	Allows administrators to remove outdated or unnecessary policies
71	ManageReports	Provides access to view and handle user reports and complaints
72	ViewDetails (Reports)	Displays detailed information about a specific user report
73	EditReport	Allows administrators to update the status or add notes to a report
74	ManagePaymentHistory	Provides access to view payment transactions and revenue analytics
75	ViewHistory	Displays a list of all payment transactions with details
76	ViewRevenueChart	Shows graphical representation of revenue data and financial statistics
77	ManageAttribute	Provides tools to manage pet attributes used for matching and profile creation
78	ViewAttribute	Displays a list of all pet attributes in the system
79	EditAttribute	Allows administrators to modify existing pet attributes
80	DeleteAttribute	Allows administrators to remove pet attributes from the system

3.1.2.3 Expert Screens (Web)

#	Feature	Description
81	ExpertLogin	Allows experts to enter email and password to access the expert portal

82	ExpertHome	Displays the main dashboard with quick access to chat, AI consultation, and approval features
83	ChatWithUser	Provides access to communicate directly with users seeking expert advice
84	ViewChatHistory	Displays the complete conversation history with a specific user
85	ViewChatList	Shows a list of all active and past chat conversations with users
86	Send/Receive Messages	Enables real-time messaging between experts and users for consultation
87	ChatWithAI	Provides tools for experts to interact with the AI system for reference and testing
88	CreateChatWithAI	Allows experts to initiate a new conversation session with the AI assistant
89	GetChatWithAIList	Displays a list of all AI chat sessions created by the expert
90	GetAllChatWithAIContent	Shows the complete content and history of AI chat conversations
91	DeleteChatWithAI	Allows experts to remove AI chat sessions they no longer need
92	SendChatWithAIQuestion	Enables experts to send questions or queries to the AI for responses
93	ApproveAIChat	Provides tools to review and approve/reject AI-generated responses for user consultations
94	View Consultation Requests	Displays a list of pending consultation requests from users
95	ViewUser'sAIChatHistory	Shows the AI chat history of a user for context before providing expert advice
96	Accept/Reject Consultation Requests	Allows experts to accept or decline incoming consultation requests from users
97	ViewUnreadRequest	Displays a list of new consultation requests that haven't been reviewed yet
98	ViewReadedRequest	Shows consultation requests that have been viewed but not yet processed
99	Review/Reply Request	Enables experts to review AI-generated content and provide corrections or confirmations

3.1.3 Screen Authorization

#	Screen	User	Admin	Expert
1	Login	X	X	X
2	Register/SignUp	X		
3	OTPVerification	X		
4	ForgotPassword	X		
5	ResetPassword	X		
6	ChangePassword	X		
7	AddPetInfo	X		
8	Home/Dashboard	X	X	X
9	PetEvent	X		
10	SendPetImage	X		
11	VotePet	X		
12	FilterScreen	X		
13	PetProfile	X		
14	Favorite	X		
15	Notification	X	X	X
16	NotificationDetails	X	X	X
17	ExpertChatList	X		
18	ExpertChat	X		
19	ChatList	X		
20	ChatDetail/ChatHistory	X		
21	Send/Receive Messages	X		X
22	CreatePetAppointment	X		
23	AIChatList	X		X
24	AIChat	X		X
25	ExpertConfirm	X		
26	UserProfile	X		

27	EditProfile	X		
28	EditPet	X		
29	AddPet	X		
30	Settings	X		X
31	PaymentHistory	X		
32	Premium	X		
33	QRPayment	X		
34	BlockedUsers	X		
35	MyReports	X		
36	ViewListPolicy	X		X
37	ViewPolicyDetails	X		X
38	AcceptListPolicyDetails	X		X
39	PetAppointment	X		
40	ViewPetAppointment	X		
41	EditPetAppointment	X		
42	ManageChatUser		X	
43	ViewChatUser		X	
44	ManageChatFilterRules		X	
45	CreateChatFilterRules		X	
46	ViewChatFilterRules		X	
47	EditChatFilterRules		X	
48	DeleteChatFilterRules		X	
49	ReloadCacheChatFilterRules		X	
50	ManageEvent		X	
51	CreatePetEvent		X	
52	EditPetEvent		X	
53	DeletePetEvent		X	
54	ViewPetEvent		X	

55	ManageUsers		X	
56	ViewUserDetails		X	
57	BanUsers		X	
58	ManageExpert		X	
59	CreateExpertAccount		X	
60	ViewExpertList		X	
61	ViewExpertInformation		X	
62	ManagePet		X	
63	ViewPet		X	
64	ManagePolicy		X	
65	CreatePolicy		X	
66	EditPolicy		X	
67	DeletePolicy		X	
68	ManageReports		X	
69	ViewReportDetails		X	
70	EditReport		X	
71	ManagePaymentHistory		X	
72	ViewRevenueChart		X	
73	ManageAttribute		X	
74	ViewAttribute		X	
75	EditAttribute		X	
76	DeleteAttribute		X	
77	ChatWithUser			X
78	CreateChatWithAI			X
79	GetChatWithAIList			X
80	GetAllChatWithAIContent			X
81	DeleteChatWithAI			X
82	SendChatWithAIQuestion			X

83	ApproveAIChat			X
84	ViewConsultationRequests			X

3.1.4 Non-Screen Functions

#	Feature	System Function	Description
1	Payment Processing	VietQR Payment Gateway API	Handles secure bank transfer transactions using VietQR API to generate QR codes and process payment callbacks for VIP/Premium subscriptions.
2	Pet Matching	Pet Match Algorithm	Background service that processes user preferences (location, breed, age) and suggests compatible pet matches using filtering and ranking logic.
3	AI Image Analysis	Google Gemini Vision API	Analyzes uploaded pet photos to identify breed, detect potential health issues, and provide recommendations using AI-powered image recognition.
4	AI Chatbot	Google Gemini Chat API	Provides automated responses to user queries about pet care, breeding advice, and general questions using natural language processing.
5	Email Service	SMTP & Kickbox Integration	Sends OTP verification codes, password reset emails, and system notifications. Validates email addresses before sending using Kickbox API.
6	Image Storage	Cloudinary Cloud Storage	Stores and delivers user-uploaded pet photos with automatic optimization, CDN distribution, and URL generation for database storage.
7	Real-time Messaging	SignalR WebSocket	Enables real-time chat functionality between users and experts, including instant message delivery and online status indicators.
8	Authentication	JWT Token Management	Generates and validates JSON Web Tokens for secure user authentication across API requests with role-based access control.
9	Password Security	BCrypt Hashing	Encrypts user passwords using BCrypt algorithm with salt for secure storage and validation during login.
10	Notification System	Email Notification Service	Sends automated notifications to users about new matches, messages, expert replies, and system alerts via email.
11	Data Validation	Email & Input Validation	Validates user inputs, email formats, and data integrity before processing to prevent SQL injection and invalid data storage.

12	Daily Limit Tracking	Rate Limiting Service	Tracks and enforces daily limits on user actions (likes, expert consultations) based on account type (Free vs Premium).
----	----------------------	-----------------------	---

3.1.5 Entity Relationship Diagram

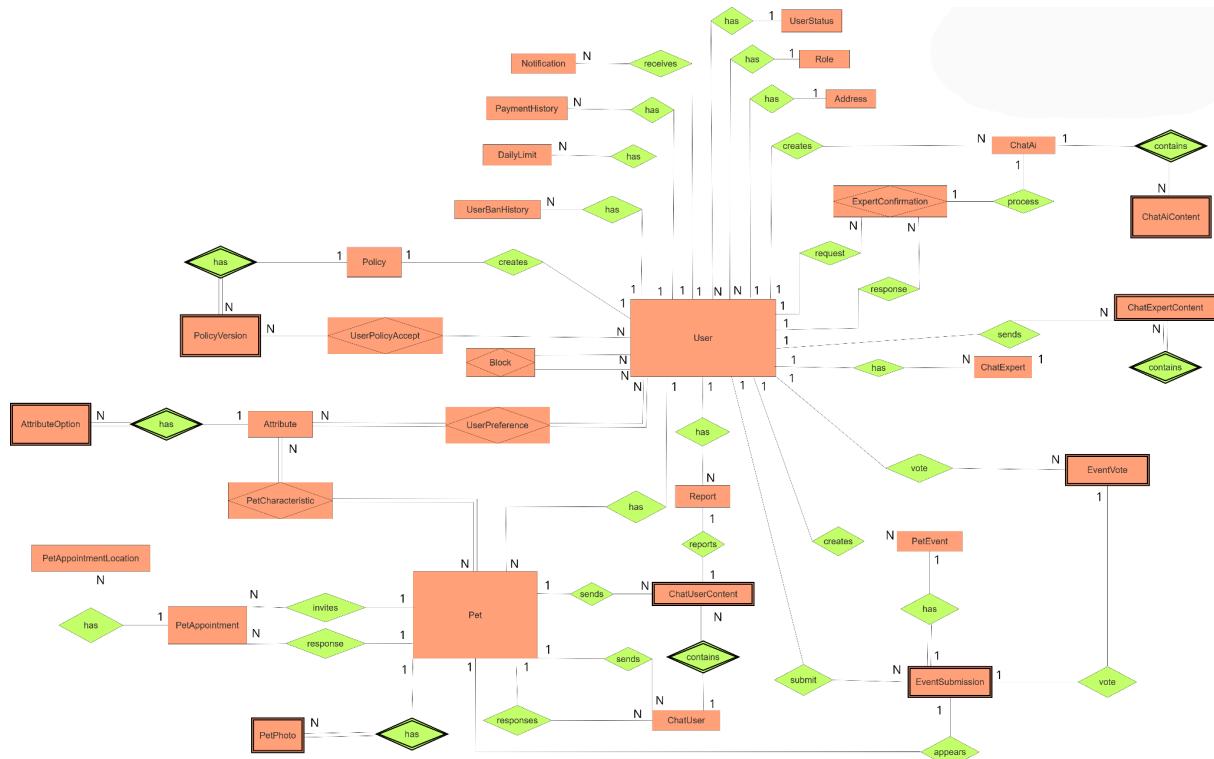


Figure 3.1.5. Entity Relationship Diagram of the Pawnder system

Entities Description:

#	Entity	Description
1	User	Represents a user account in the system.
2	Pet	Represents a pet profile owned by a user.
3	Role	Defines user roles in the system.
4	UserStatus	Defines account statuses for users.
5	Address	Represents a location address.
6	Notification	Represents a notification sent to a user.
7	PaymentHistory	Records a VIP subscription payment transaction.
8	DailyLimit	Tracks daily usage limit for a specific user action.
9	UserBanHistory	Records a ban history entry for a user.

10	Policy	Represents a system policy document.
11	PolicyVersion	Represents a version of a policy.
12	UserPolicyAccept	Records a user's acceptance of a policy version.
13	Attribute	Represents an attribute used for pet matching.
14	AttributeOption	Represents an option value for an attribute.
15	PetCharacteristic	Links a pet to an attribute with its value.
16	UserPreference	Links a user to their preferred attribute values for matching.
17	Block	Records a blocking relationship between two users.
18	ChatAI	Represents a chat session between a user and AI assistant.
19	ChatAiContent	Represents a message in an AI chat session.
20	ChatExpert	Represents a chat session between a user and an expert.
21	ChatExpertContent	Represents a message in an expert chat session.
22	ChatUser	Represents a match between two pets.
23	ChatUserContent	Represents a message in a user-to-user chat.
24	Report	Represents a user report about inappropriate content.
25	PetPhoto	Represents a photo of a pet.
26	PetAppointmentLocation	Represents a pet-friendly location for appointments.
27	PetAppointment	Represents an appointment between two matched pets.
28	PetEvent	Represents a pet photo/video contest event.
29	EventSubmission	Represents a contest entry submitted by a user.
30	EventVote	Represents a vote cast on a contest submission.
31	ExpertConfirmation	Represents a confirmation request linking AI chat to expert consultation.

3.2 Authentication & Onboarding

3.2.1 User Login

UC-001 Login

Use Case ID	UC-001	Uce Case Name	Login
Primary Actors	User (Pet Owner)	Secondary Actors	System

Description	Users authenticate themselves by entering their registered email and password to access the Pawnder system. The system validates credentials and grants access to the application.
Preconditions	User has a registered account in the system. User is on the Login screen. User is not currently logged in.
Postconditions	User is successfully authenticated. JWT authentication token is generated and stored locally. User is navigated to Home/Match Feed screen.
Normal Sequence/Flow	<ol style="list-style-type: none"> 1. User navigates to Login screen from Welcome screen or Register screen. 2. System displays Login screen with email and password input fields. 3. User enters email address in the email field. 4. User enters password in the password field. 5. User clicks the "Sign In" button. 6. System validates email format. 7. System sends authentication request to backend server. 8. Backend validates credentials against database (email and BCrypt-encrypted password). 9. Backend generates JWT token with 7-day expiration. 10. System stores authentication token in local secure storage. 11. System navigates user to Home/Match Feed screen.
Alternative Sequences/Flows	<p>A1. If email format is invalid → System displays error message: "Please enter a valid email address" below email field. Return to step 3 in normal flow.</p> <p>A2. If credentials are invalid → System sends authentication request to backend server. Backend finds no matching email or password does not match. System displays error message: "Invalid email or password". Return to step 3 in normal flow.</p> <p>A3. If user clicks "Forgot Password?" link → System navigates to Forgot Password screen (UC-004).</p> <p>A4. If user clicks "Sign Up" link → System navigates to Register screen (UC-003).</p>

Screen Design: Login Screen

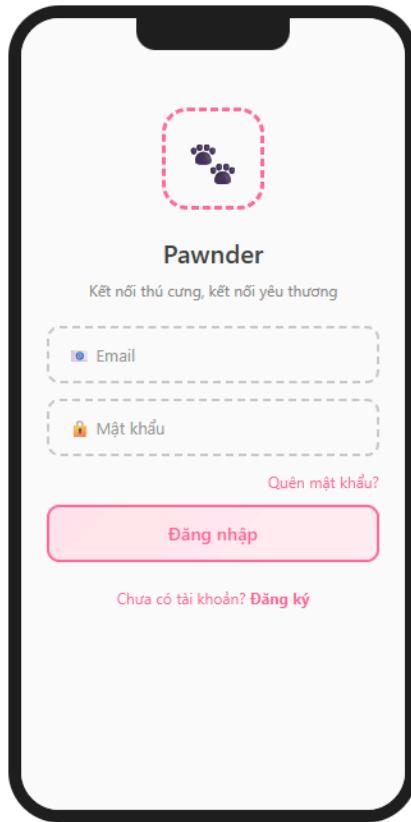


Figure 3.2.1: Login Screen Mockup

Sign In Screen

This screen allows the user to:

- Authenticate into the application using email and password credentials
- Navigate to registration page for new account creation
- Access password recovery functionality

Field Description

Field Name	Description
Branding Section	Displays app logo with pet photo at the top of the screen
Login Form	Contains input fields for email and password to authenticate user credentials
Action Buttons	Includes sign in button to submit credentials and navigation links for registration and password recovery

3.2.2 User Logout

UC-002 Logout

Use Case ID	UC002	Use Case Name	Logout
Primary Actors	User (Pet Owner), Expert, Admin	Secondary Actors	System
Description	This use case describes the process by which an authenticated user terminates their current session. The system invalidates the session credentials, clears local data, and returns the application to the unauthenticated state.		
Preconditions	1. User is currently logged into the system. 2. User has an active session.		
Postconditions	1. User session is terminated. 2. Local session credentials and user data are cleared. 3. User is redirected to Welcome Screen.		
Normal Sequence/Flow	1. User navigates to Settings Screen from the application. 2. System displays Settings Screen with account options. 3. User scrolls to "Danger Zone" section. 4. User taps the "Logout" button. 5. System displays a confirmation dialog. 6. User taps "Logout" to confirm. 7. System invalidates the session and clears local data. 8. System redirects User to Welcome Screen.		
Alternative Sequences/Flows	A1. If user cancels logout → 1. System closes the confirmation dialog. 2. System returns User to Settings Screen.		

Screen Design: Logout Screen

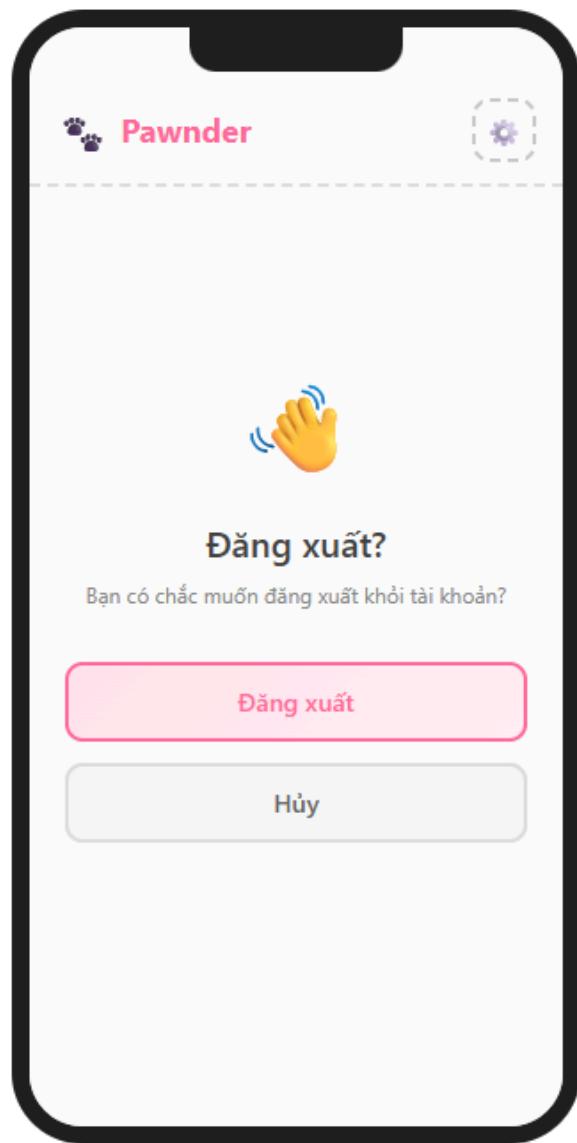


Figure 3.2.2: Logout Screen Mockup

Logout Screen

This screen allows the user to:

- Securely log out of the application and terminate the current session
- Confirm logout action before proceeding with session termination
- Cancel logout operation and return to the previous screen

Field Description

Field Name	Description
Confirmation Dialog	Displays a warning dialog asking the user to confirm they want to log out of the application

Logout Message	Shows a clear message explaining that logging out will end the current session and require re-authentication to access the app again
Action Buttons	Includes two buttons: "Cancel" to dismiss the dialog and return to settings, and "Logout" to confirm and proceed with logging out

3.2.3 User Registration

UC-003 Register

Use Case ID	UC003	Use Case Name	Register
Primary Actors	User (Pet Owner)	Secondary Actors	Email Service (SMTP)
Description	New users create an account by providing their full name, gender, email address, and password. The system validates the information and sends a verification OTP to the user's email.		
Preconditions	1. User is on the Register screen. 2. User does not have an existing account with the provided email.		
Postconditions	1. OTP is sent to user's email. 2. User data is temporarily stored. 3. User is navigated to OTP Verification screen.		
Normal Sequence/Flow	1. User navigates to Register screen from Welcome screen or Login screen. 2. System displays Register screen with name, gender, email, password, and confirm password input fields. 3. User enters full name in the name field. 4. User selects gender (Male/Female). 5. User enters email address in the email field. 6. User enters password in the password field. 7. User enters confirm password in the confirm password field. 8. User clicks the "Sign Up" button. 9. System validates all input fields. 10. System sends OTP to user's email. 11. System displays success message: "Check your email. OTP has been sent." 12. System navigates to OTP Verification screen with user data.		
Alternative Sequences/Flows	A1. If input validation fails → System displays specific error message for the invalid field (Name, Gender, Email, Password, Confirm password). Return to corresponding step. A2. If email already exists → System displays error message: "Email already exists." User can click "Sign In" link to navigate to Login screen. A3. If network error occurs →		

	<p>System displays error message: "Connection error. Please check your network." Return to step 8.</p> <p>A4. If user clicks "Sign In" link →</p> <p>System navigates to Login screen (UC-002).</p>
--	---

Screen Design: Register Screen

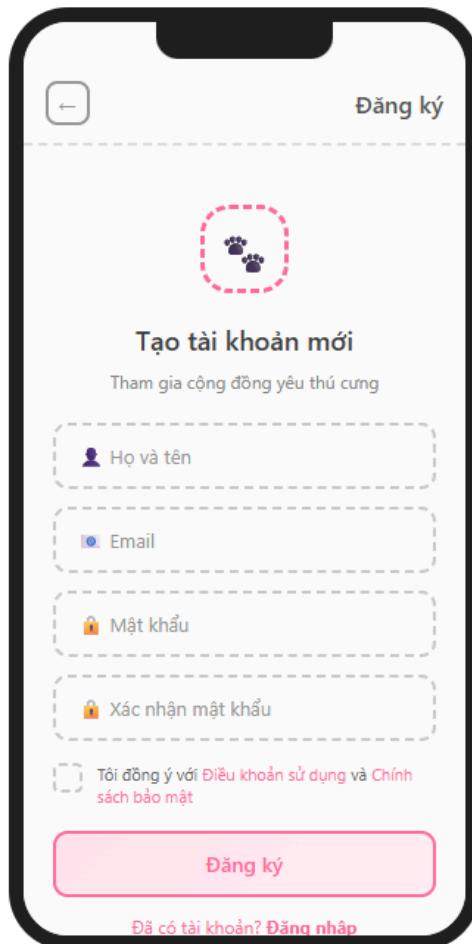


Figure 3.2.3: Register Screen Mockup

Sign Up Screen

This screen allows the user to:

- Create a new account by providing personal information and credentials
- Upload or capture a profile photo for their account
- Select their gender preference
- Accept terms and policy before registration
- Navigate back to sign in screen if already have an account

Field Description

Field Name	Description
------------	-------------

Profile Photo Section	Allows user to upload or capture profile photo with circular placeholder
Registration Form	Contains input fields for full name, gender selection, email, password, and password confirmation
Terms Agreement	Checkbox for user to accept terms and policy before registration
Action Buttons	Includes back button, sign up button to submit form, and link to navigate to sign in screen

3.2.4 Reset Password

UC-004 Reset Password

Use Case ID	UC004	Use Case Name	Forgot Password
Primary Actors	User (Pet Owner)	Secondary Actors	Email Service (SMTP)
Description	Users who forgot their password can request a password reset by providing their registered email address. The system validates the email and sends a 6-digit OTP code for verification.		
Preconditions	1. User has a registered account in the system. 2. User is on the Forgot Password screen.		
Postconditions	1. 6-digit OTP code is generated and sent to user's email. 2. User is navigated to Reset Password screen.		
Normal Sequence/Flow	1. User navigates to Forgot Password screen from Login screen. 2. System displays Forgot Password screen with email input field. 3. User enters registered email address in the email field. 4. User clicks the "Send OTP" button. 5. System validates email format. 6. System sends OTP to user's email. 7. System displays success message: "OTP sent to your email." 8. System navigates to Reset Password screen.		
Alternative Sequences/Flows	A1. If email is empty → System displays error message: "Please enter your email." Return to step 3. A2. If email format is invalid → System displays error message: "Please enter a valid email address." Return to step 3. A3. If email not found or OTP send fails → System displays error message from server. Return to step 3. A4. If user clicks "Back to Sign In" link → System navigates to Login screen (UC-002).		

Screen Design: Forgot Password Screen

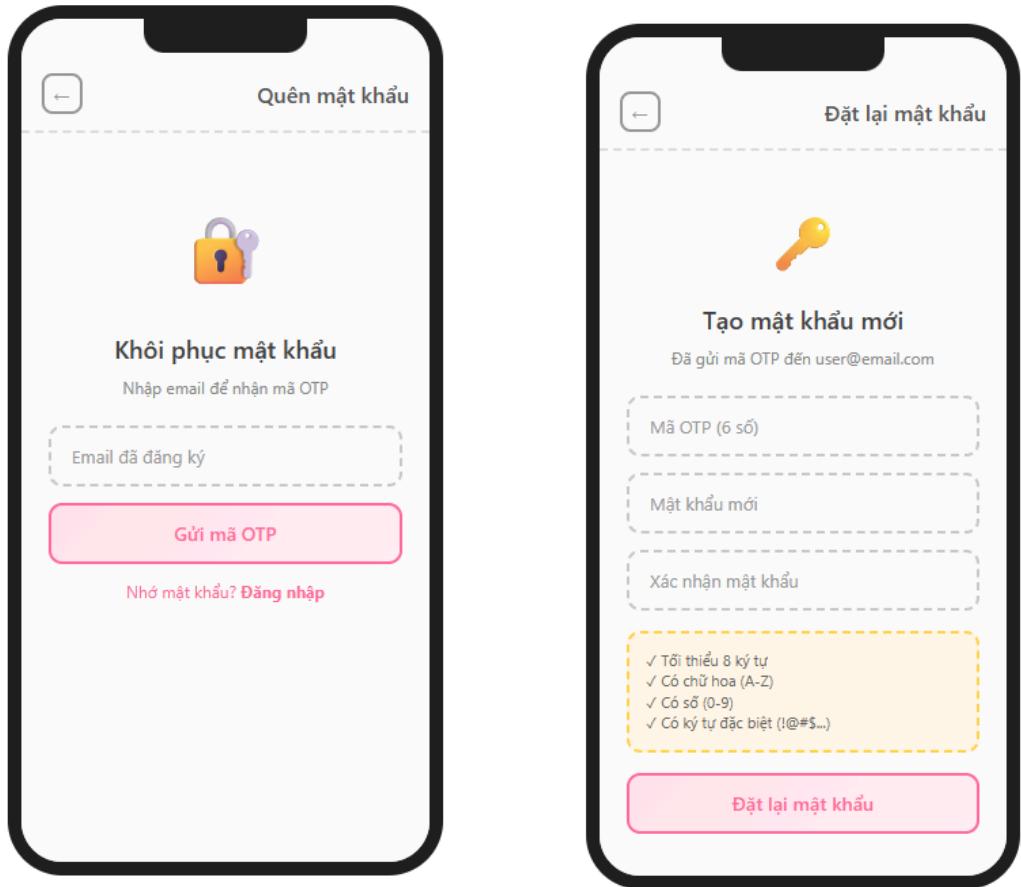


Figure 3.2.4: Forgot Password Screen Mockup

Reset Password Screen

This screen allows the user to:

- Request a password reset by providing their registered email address
- Receive a verification code to their email for password recovery
- Navigate back to the previous screen

Field Description

Field Name	Description
Header Section	Contains back button for navigation and screen title with instructional text
Email Form	Includes email input field with label and placeholder text for user to enter registered email address
Action Button	Send Code button to submit email and trigger verification code delivery

3.2.5. Change Password Screen

UC-005: Change Password

Use Case ID	UC005	Use Case Name	Change Password
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to change their account password. Users must enter current password, new password, and confirm new password. System validates password requirements (minimum 8 characters, contains uppercase, lowercase, number, special character, different from current, passwords match) with real-time visual feedback.		
Preconditions	1. User is logged into the application. 2. User has accessed Change Password from the Settings Screen.		
Postconditions	1. Password is changed successfully. 2. Success message is shown. 3. User is navigated back to Settings Screen.		
Normal Sequence/Flow	1. User taps "Change Password" from Settings Screen. 2. System displays Change Password Screen with three password input fields and requirements checklist. 3. User enters current password in the current password field. 4. User enters new password in the new password field. 5. System validates in real-time and shows visual checkmarks for met requirements when new password field is focused. 6. User enters confirm password in the confirm password field. 7. System shows checkmark icon if passwords match, X icon if they don't match. 8. User taps "Change Password" button. 9. System validates all fields. 10. System calls change password API. 11. System shows success alert with message from server. 12. User closes alert and system navigates back to Settings Screen.		
Alternative Sequences/Flows	A1. If user toggles password visibility → User taps eye icon. System toggles show/hide password. A2. If current password is empty → System shows warning: "Please enter your current password." Return to step 3. A3. If new password is empty → System shows warning: "Please enter your new password." Return to step 4. A4. If new password validation fails (length, case, number, special char) → System shows corresponding warning message. Return to step 4. A5. If new password is same as current → System shows warning: "New password must be different from current password." Return to step 4.		

	<p>A6. If passwords don't match → System shows warning: "Passwords do not match." Return to step 6.</p> <p>A7. If API fails → System shows error message from server. Return to step 3.</p> <p>A8. If user taps "Forgot your current password?" → System navigates to Forgot Password Screen.</p> <p>A9. If user taps back button → System navigates back to Settings Screen.</p>
--	---

Screen Design: Change Password Screen

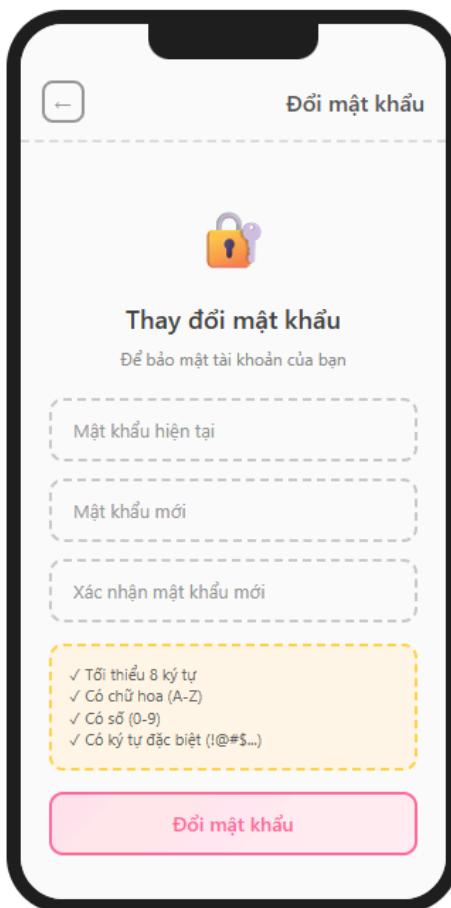


Figure 3.2.5: Change Password Screen Mockup

Change Password Screen

This screen allows the user to:

- Update account password with security requirements

- Verify current password before changing
- View password strength requirements

Field Description

Field Name	Description
Header Section	White header with back button and "Change Password" title
Security Icon	Pink circular icon with key symbol indicating password security feature
Instructions Text	Informational text "Make sure your new password is strong and secure"
Password Form	Three input fields for "Current Password", "New Password", and "Confirm New Password"
Password Requirements	Checklist showing requirements with checkmarks: "At least 8 characters", "Different from current password", "Passwords match"
Change Password Button	Pink gradient button to submit password change

3.2.6 View Home Page

UC-006 View Home Page

Use Case ID	UC006	Use Case Name	View Home Screen
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to view the main homepage displaying recommended pet cards in a swipeable stack format. Users can swipe right to like, swipe left to pass, or tap to view pet details.		
Preconditions	1. User is logged into the system. 2. User has at least one pet profile. 3. User has an active pet set.		
Postconditions	1. Home screen is displayed with recommended pet cards. 2. User can interact with pet cards (swipe, view details, navigate photos).		
Normal Sequence/Flow	1. User opens the application or navigates to Home Screen. 2. System displays loading skeleton with pet card placeholders. 3. System retrieves User ID and active Pet ID from local storage. 4. System fetches VIP status for current User.		

	<p>5. System fetches recommended pets from server based on User preferences and active pet.</p> <p>6. System preloads first 10 pet images for smooth display.</p> <p>7. System loads remaining pets in background.</p> <p>8. System displays pet cards in swipeable stack format showing photos, info, and match percentage.</p> <p>9. System displays header with logo, events, appointments, filters, and notifications.</p> <p>10. System displays bottom navigation bar (Home, Favorite, Chat, Profile).</p> <p>11. System connects to SignalR for real-time match notifications.</p>
Alternative Sequences/Flows	<p>A1. If no pets available → System displays empty state with "Reload" and "Adjust Filters" buttons.</p> <p>A2. If no active pet → System prompts User to create/activate a pet in Profile.</p> <p>A3. If user pulls to refresh → System reloads pets from server.</p> <p>A4. If user swipes right/taps Like → System animates "LIKE", sends request to server, and moves to next card.</p> <p>A5. If user swipes left/taps Pass → System animates "PASS" and moves to next card.</p> <p>A6. If user taps info/match badge → System navigates to Pet Profile Detail or opens Match Details Modal.</p> <p>A7. If user taps photo carousel → System navigates previous/next photo and updates dots.</p> <p>A8. If all pets swiped → System displays empty state with animated paw icon.</p> <p>A9. If login success detected → System shows "Welcome! Login successful" alert and clears flag.</p>

Screen Design: View Home Page

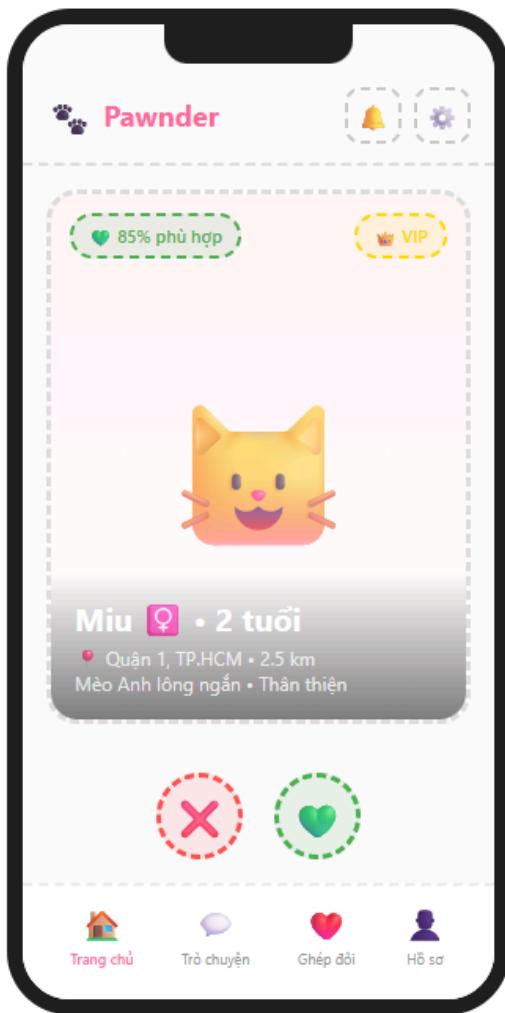


Figure 3.2.6: View Home Page Mockup

View Home Page

This screen allows the user to:

- Browse recommended pet profiles through swipeable cards
- Swipe right to like or swipe left to pass on pet profiles
- View pet photos, information, and match percentage
- Access events, appointments, filters, and notifications

Field Description

Field Name	Description
Header Section	Displays Pawnder logo and navigation icons (events, appointments, filters, notifications with badge)
Pet Card	Swipeable card showing pet photo, name, age, breed, distance, and bio with gradient overlay
Match Badge	Clickable percentage badge showing compatibility score with star icon

Pet Photos	Multiple photos with dots indicator and tap navigation to switch between images
Action Buttons	Heart button (like) and X button (pass) at bottom of card
Info Button	Button in top-right corner to view full pet profile details
Owner Info	Shows owner name with VIP badge if applicable
Bottom Navigation	Fixed navigation bar with Home, Chat, Matches, and Profile tabs

3.2.7 View Policies

UC-017 View Policies

Use Case ID	UC017	Use Case Name	View Application Policies
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to view application policies (Terms of Service, Privacy Policy, Community Guidelines) from Settings. Users can browse policy list, read full policy content with version information and published dates.		
Preconditions	1. User is logged into the system. 2. User has accessed Settings Screen.		
Postconditions	1. User has viewed the policy content. 2. System displays policy details successfully.		
Normal Sequence/Flow	1. User navigates to Settings Screen and taps "Policies" option. 2. System displays loading skeleton with policy item placeholders. 3. System fetches active policies from server. 4. System displays Policy List Screen with icons, names, versions, and dates. 5. User taps on a policy card to view details. 6. System displays loading indicator and fetches policy details from server. 7. System displays Policy Detail Screen with header, version, date, and scrollable content.		

	8. User reads through the policy content. 9. User taps back button to return to Policy List Screen.
Alternative Sequences/Flows	A1. If list is empty → System displays "No policies available." A2. If load fails → System displays error state with "Retry" button. A3. If user pulls to refresh → System reloads policies from server. A4. If detail load fails → System shows error and "Retry" option. A5. If direct navigation → System displays ResourceDetail with predefined content.

Screen Design: View Policies Screen

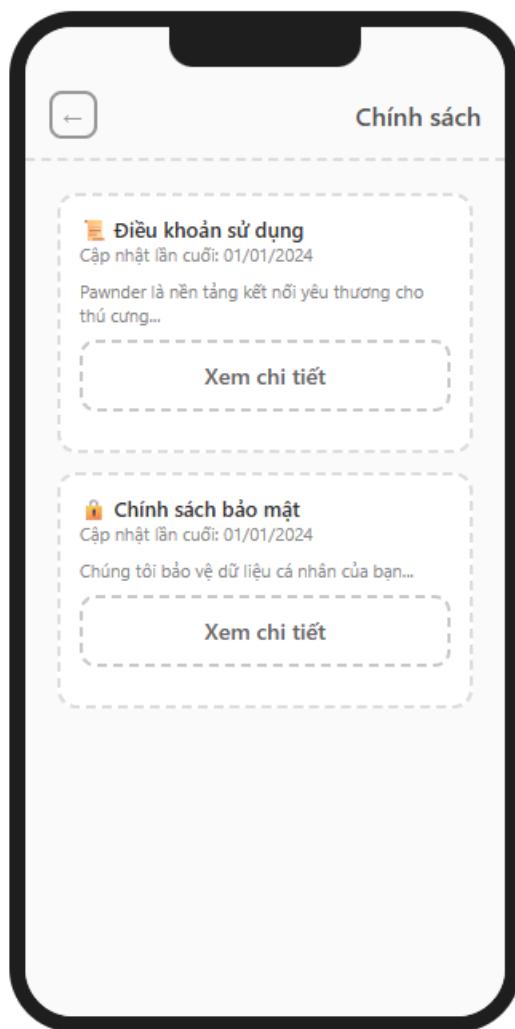


Figure : View Policies Screen Mockup

View & Accept Policies Screen

This screen allows the user to:

- View list of all active system policies
- Access detailed policy content by tapping on policy items

- Refresh policy list with pull-to-refresh gesture
- See policy version number and published date

Field Description

Field Name	Description
Header Section	Displays document icon, screen title "Accept Policies", and subtitle explaining the requirement
Policy Cards	Expandable cards showing policy name, version, and full content with checkbox for acceptance
Checkbox	Individual checkbox for each policy to indicate user has read and agrees to the terms
Info Card	Information banner with icon explaining that all policies must be accepted to continue
Accept All Button	Gradient button at bottom to submit acceptance (disabled until all policies are checked)

3.2.8 Accept Policies

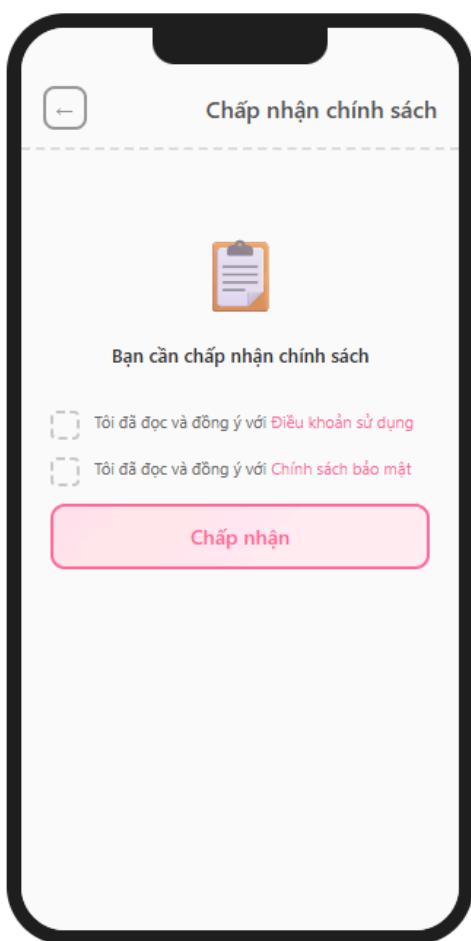
UC-018 Accept Policies

Use Case ID	UC018	Use Case Name	Accept Application Policies
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to accept application policies during onboarding or when policies are updated. Users must accept all required policies to proceed with registration or continue using the app.		
Preconditions	1. User is registering new account OR policy has been updated since last acceptance. 2. User has pending policies to accept.		
Postconditions	1. User acceptance is recorded in database with timestamp. 2. User is navigated to Home Screen.		
Normal Sequence/Flow	1. System detects User has pending policies to accept. 2. System displays Policy Acceptance Screen with list of pending policies. 3. User taps on a policy card to expand and view content. 4. User reads the policy content. 5. User taps checkbox to accept the policy. 6. User repeats steps 3-5 for all pending policies. 7. User taps "Accept All" button. 8. System sends acceptance requests to server. 9. System navigates to Home Screen.		

Alternative Sequences/Flows	<p>A1. If user taps "Accept All" without checking all policies → System shows warning alert: "Please accept all policies to continue." Return to step 3.</p> <p>A2. If acceptance request fails → System shows error alert with error message. Return to step 7.</p> <p>A3. If only one policy pending → System displays policy card in expanded state by default.</p> <p>A4. If policy was previously accepted (update) → System displays "Update" badge next to version number.</p>
-----------------------------	---

Screen Design: Accept Policies Screen

Figure : Accept Policies Screen Mockup



View & Accept Policies Screen

This screen allows the user to:

- Review all pending policies that require acceptance
- Read policy content by expanding each policy card
- Check individual policy checkboxes to indicate agreement

- Accept all policies at once after reviewing and checking all items

Field Description

Field Name	Description
Header Section	Displays document icon, screen title "Accept Policies", and subtitle explaining the requirement
Policy Cards	Expandable cards showing policy name, version, and full content with checkbox for acceptance
Checkbox	Individual checkbox for each policy to indicate user has read and agrees to the terms
Info Card	Information banner with icon explaining that all policies must be accepted to continue
Accept All Button	Gradient button at bottom to submit acceptance (disabled until all policies are checked)
Policy Counter	Shows progress text indicating how many policies have been checked out of total

3.3 Pet Profile Management

3.3.1 View Own Pet Profile

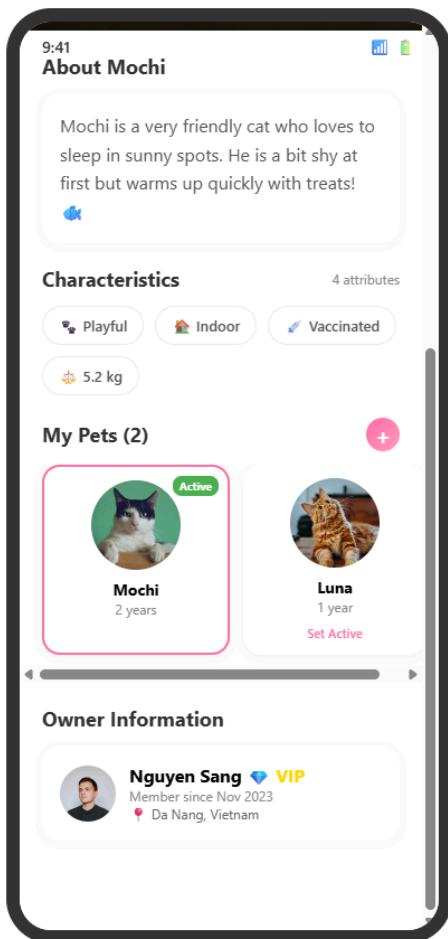
UC-031: View Own Pet Profile

Feature	Description
Use Case ID	UC-031
Use Case Name	View Own Pet Profile
Primary Actors	User (Pet Owner)
Secondary Actors	None
Description	Allows the user to view a list of all their registered pets and access options to add, edit, or manage them. The system retrieves the user's pets from the database and displays them in a card layout.

Preconditions	<ul style="list-style-type: none"> • User has successfully logged into the application. • User has navigated to the "Profile" tab.
Postconditions	<p>Success case:</p> <ul style="list-style-type: none"> • User sees a list of their registered pets with key details (name, breed, age, photo). • User can identify which pet is currently active for matching.
Normal Sequence/Flow	<p>Step 3.2.1: View My Pets List</p> <ol style="list-style-type: none"> 1. User taps the "Profile" icon on the bottom navigation bar. 2. System retrieves the list of pets associated with the user's account from the database. 3. System displays the My Pets Screen. 4. System renders a card for each pet containing: <ul style="list-style-type: none"> ○ Pet's profile photo (or placeholder). ○ Pet's name and gender symbol. ○ Pet's breed and age. ○ "Active" badge if the pet is currently selected for matching. 5. User views the list of pets. 6. User can tap on a pet card to view details (UC-009) or tap "Add New Pet" to create a profile (UC-008).
Alternative Sequences/Flows	<p>Step 3.2.1.1: User Has No Pets</p> <p>1-2. Same as normal flow.</p> <p>3. System detects that the user has 0 pets registered.</p> <p>4. System displays an empty state message (e.g., "No pets found").</p> <p>5. System highlights the "Add New Pet" button to encourage profile creation.</p> <p>Step 3.2.1.2: System Fails to Load Pets</p> <ol style="list-style-type: none"> 1. User navigates to the Profile tab. 2. System attempts to fetch pet data but encounters a network or server error. 3. System displays an error message (e.g., "Failed to load pets. Please try again."). 4. System provides a "Retry" button. 5. User taps "Retry" to attempt fetching data again.

Screen Design: View Own Pet Profile Screen

Figure : View Own Pet Profile



View Own Pet Profile

This screen allows the user to:

- View list of their own pets (maximum 3 pets)
- See which pet is currently active for matching
- Add new pet profile (if under limit)
- Edit or set active status for each pet

Field Description

Field Name	Description
Header Section	Displays "My Pets" title with count and "Add Pet" button
Pet Cards	Horizontal scrollable cards showing pet photo, name, breed, age with "Active" badge for current active pet

Active Badge	Green badge with checkmark indicating which pet is currently active for matching
Edit Button	Pencil icon button on each card to edit pet details
Set Active Button	Button to change active pet for matching (shown on inactive pets)
Add Pet Button	Gradient button to create new pet profile (disabled when 3 pets reached)

3.3.2. Create Pet Profile by Using AI

UC-032: Create Pet Profile by Using AI

Feature	Description
Use Case ID	UC-032
Use Case Name	Create Pet Profile by Using AI
Primary Actors	User (Pet Owner)
Secondary Actors	Google Gemini AI
Description	The user initiates and completes the multi-step wizard process to create a new, complete pet profile. This includes entering basic information, uploading photos, receiving AI analysis from Google Gemini, and finalizing characteristics before the profile is permanently saved to the system.
Preconditions	User is logged into the application. User is on the "My Pets" screen. User has not reached maximum pet limit (3 pets).
Postconditions	Complete pet profile is saved to the database. New pet appears in user's pet list. User is navigated back to "My Pets" screen.

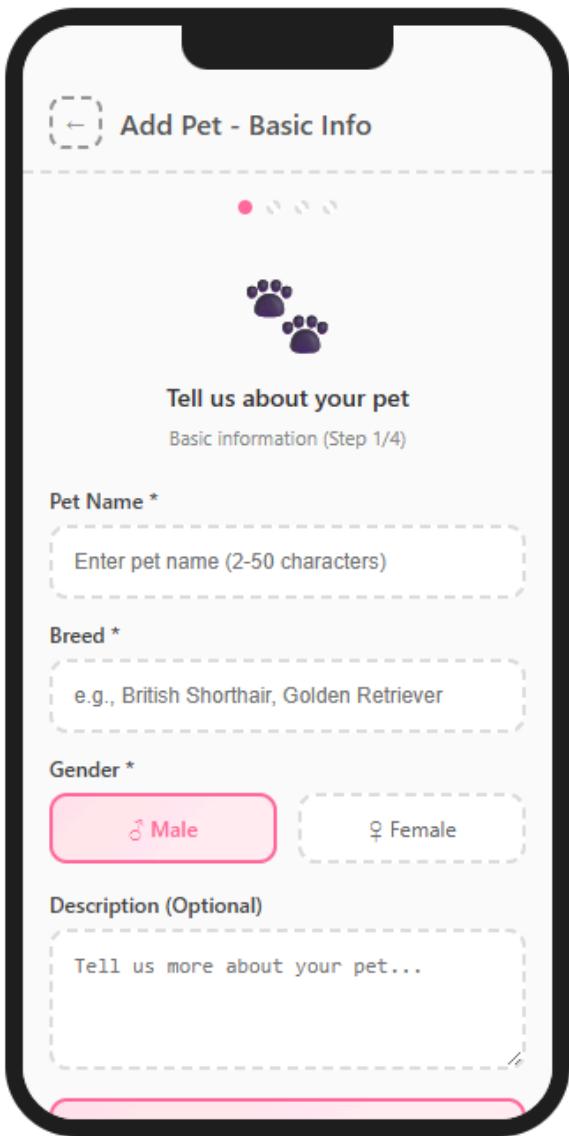
Normal Sequence/Flow	<p>: Create Pet Profile</p> <ol style="list-style-type: none"> 1. User taps the "Add New Pet" button on the My Pets screen. 2. System displays Screen 1 (Basic Info). 3. User enters name, breed, and gender and taps "Continue". 4. System validates and saves data temporarily. 5. System navigates to Screen 2 (Upload Photos). 6. User uploads at least one photo. 7. User taps the "Analyze with AI" button(Screen 3). 8. System calls the Google Gemini AI service. 9. System displays Screen (AI Pet Analysis) with analysis results. 10. User reviews the data. 11. User taps "Finish & Save" (or "Continue"). 12. System saves AI data and navigates to Screen 4 (Pet Characteristics). 13. User enters/adjusts final characteristics (Weight, Energy Level, etc.). 14. User taps the final "Finish & Save" button. 15. System performs final validation. 16. System saves the complete profile to the database. 17. System navigates back to the My Pets Screen.
----------------------	---

Alternative Sequences/Flows	<p>Step 3.3.2.1: Missing Required Fields</p> <p>User fails to complete required fields (e.g., Pet Name, Gender) on Screen 3.3.2.</p> <p>User taps "Continue".</p> <p>System detects validation error.</p> <p>System displays an error message below the missing field.</p> <p>System keeps the User on the current screen.</p> <p>User corrects the error and returns to Step 3 in the Normal Flow.</p> <p>Step 3.3.2.2: Cancel Creation</p> <p>User taps the "Back" or "Cancel" button on the header at any point during the wizard (Steps 1-17 of Normal Flow).</p> <p>System displays confirmation dialog warning about data loss.</p> <p>User confirms cancellation.</p> <p>System discards all entered/temporarily stored data.</p> <p>System deletes temporary pet record if created.</p> <p>System navigates back to the My Pets Screen.</p> <p>Step 3.3.3.1: No Photo Selected</p> <p>At Normal Flow Step 6, User attempts to proceed without uploading a minimum of one photo.</p> <p>User taps "Analyze with AI" or "Continue".</p> <p>System displays an error message: "Please upload at least one photo."</p> <p>System keeps the User on the current screen.</p> <p>Step 3.3.3.2: Skip AI Analysis</p> <p>After uploading photos (Normal Flow Step 6), User taps the "Skip AI" or "Continue Manually" button.</p> <p>System skips the AI analysis screen (3.3.4).</p> <p>System navigates directly to Screen 3.3.5 (Pet Characteristics).</p>
-----------------------------	---

	<p>The process continues from Step 13 in the Normal Flow with empty characteristic fields.</p> <p>Step 3.3.4.1: AI Analysis Failed</p> <p>At Normal Flow Step 8, System fails to get a response from the AI or analysis is inconclusive.</p> <p>System displays an error message: "Could not identify pet. Please enter details manually."</p> <p>User taps "Continue Manually".</p> <p>System navigates User to Screen 3.3.5 (Pet Characteristics).</p> <p>The process continues from Step 13 in the Normal Flow with empty characteristic fields.</p> <p>Step 3.3.5.1: Validation Error on Final Save</p> <p>At Normal Flow Step 13, User enters invalid data (e.g., negative weight) on Screen 3.3.5.</p> <p>User taps the final "Finish & Save" button.</p> <p>System detects validation error.</p> <p>System displays error message below the field.</p> <p>System keeps the User on the current screen.</p> <p>User corrects the error and returns to Step 14 in the Normal Flow.</p>
--	---

Screen Design 1: Create Pet Profile Screen

Figure: Create Pet Profile Screen Mockup



Pet Profile Screen

This screen allows the user to:

- Enter basic information about their pet including name and breed
- Navigate through a multi-step pet profile creation process
- Navigate back to the previous screen
- Proceed to the next step of pet profile setup

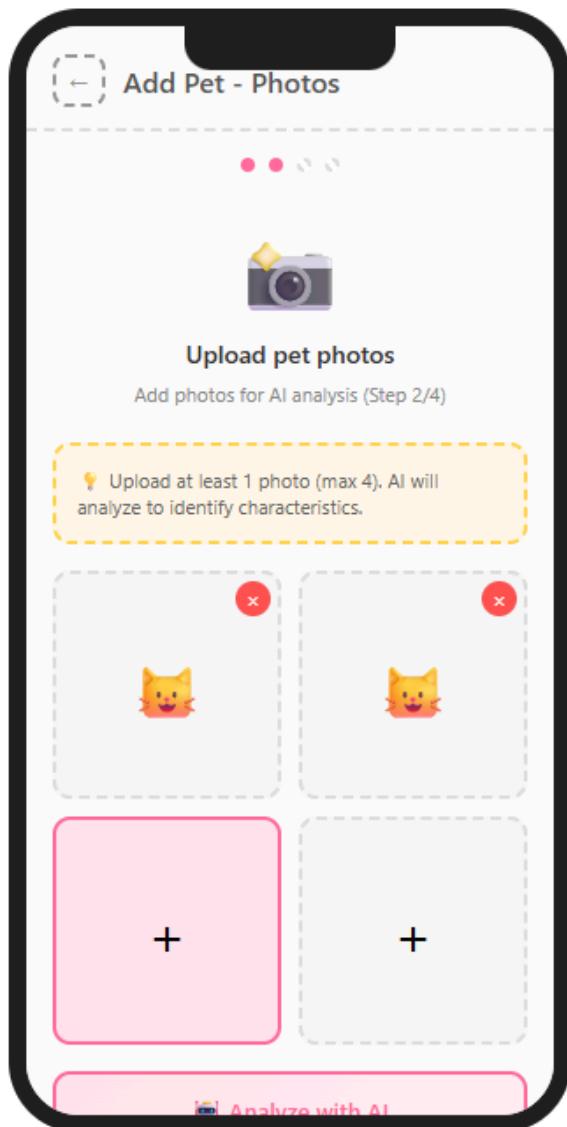
Field Description

Field Name	Description
Progress Section	Shows step indicator (Step 1 of 3) with visual progress bar and back button for navigation

Form Section	Contains input fields for pet's name and breed with labels and placeholder text to guide user input
Action Button	Continue button to validate inputs and proceed to next step of pet profile creation

Screen Design 2: Upload Pet Photos Screen

Figure: Upload Pet Photos Screen Mockup



Add Photos Screen

This screen allows the user to:

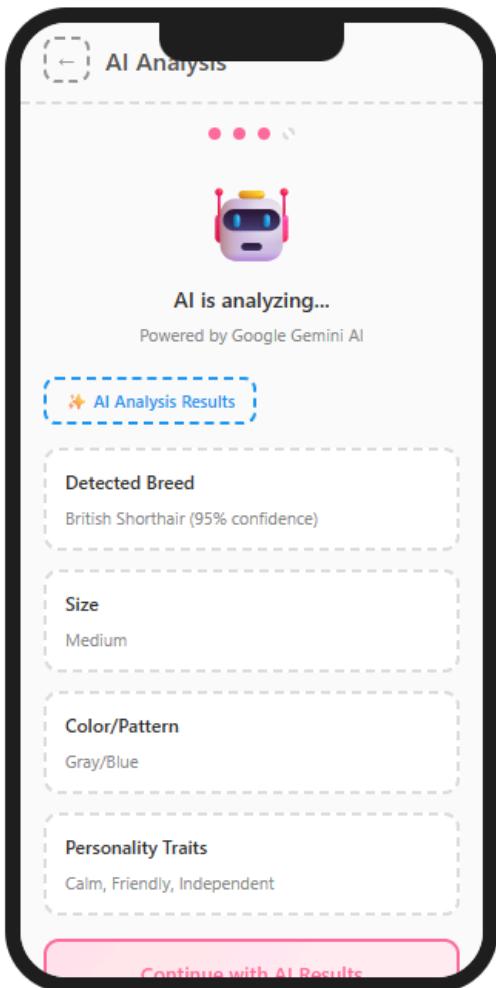
- Upload at least 3 photos of their pet for profile creation
- Select photos from device gallery or camera
- Navigate back to the previous step to edit pet information
- Proceed to the next step with AI analysis of uploaded photos

Field Description

Field Name	Description
Progress Section	Shows step indicator (Step 2 of 3) with visual progress bar and back button for navigation
Instruction Section	Contains screen title and text indicating minimum photo requirement
Photo Upload Grid	Displays uploaded photo thumbnails and empty slots with add button for selecting additional photos
Action Button	Continue with AI button to proceed to next step where system analyzes uploaded photos

Screen Design 3: AI Pet Analysis Screen

Figure: AI Pet Analysis Screen Mockup



AI Analysis Screen

This screen allows the user to:

- Review AI-generated analysis results of uploaded pet photos
- Verify automatically detected pet characteristics such as breed and color
- Navigate back to previous screen to re-upload photos if results are incorrect
- Proceed to next step with confirmed pet information

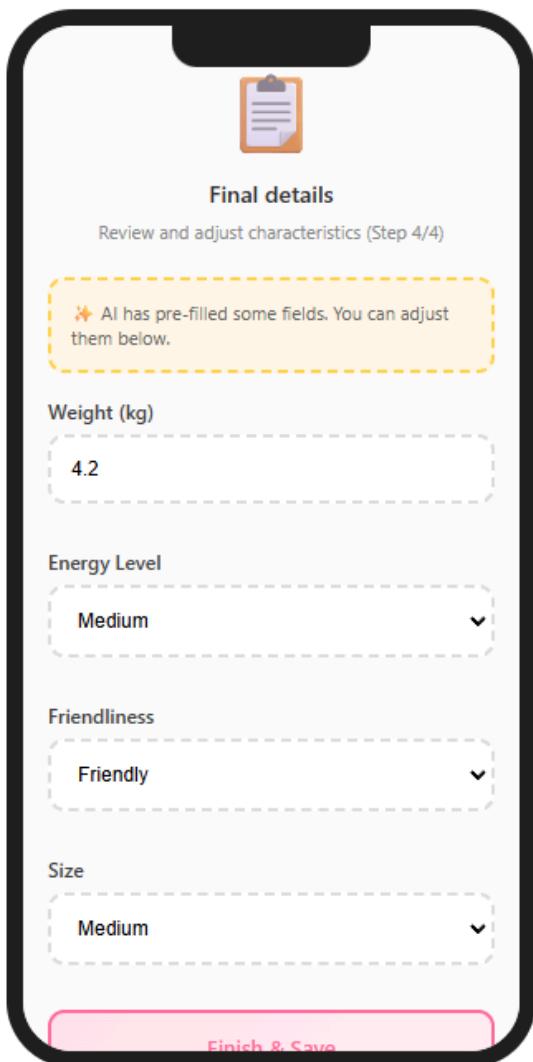
Field Description

Field Name	Description
Header Section	Contains back button and screen title with success message indicating analysis completion

Results Section	Displays AI-detected pet information including breed with confidence percentage and color in card format
Action Button	Continue button to accept AI analysis results and proceed to next step of profile creation

Screen Design 4: Pet Characteristics Screen

Figure: Pet Characteristics Screen Mockup



Characteristics Screen

This screen allows the user to:

- Enter pet's physical characteristics and weight information
- Select pet's personality traits from available options
- Complete the pet profile creation process

Field Description

Field Name	Description
Progress Indicator	Shows user's current position (Step 3 of 3) in the multi-step pet profile creation flow
Weight Input	Allows user to enter pet's weight in kilograms as numeric value
Personality Selection	Provides multiple personality trait options (Playful, Shy, Energetic) for user to select pet's behavior characteristics
Action Buttons	Contains back button for navigation to previous step and Finish Profile button to complete and save the pet profile

3.3.3. Edit Pet Profile Screen

UC-033: Edit Pet Profile

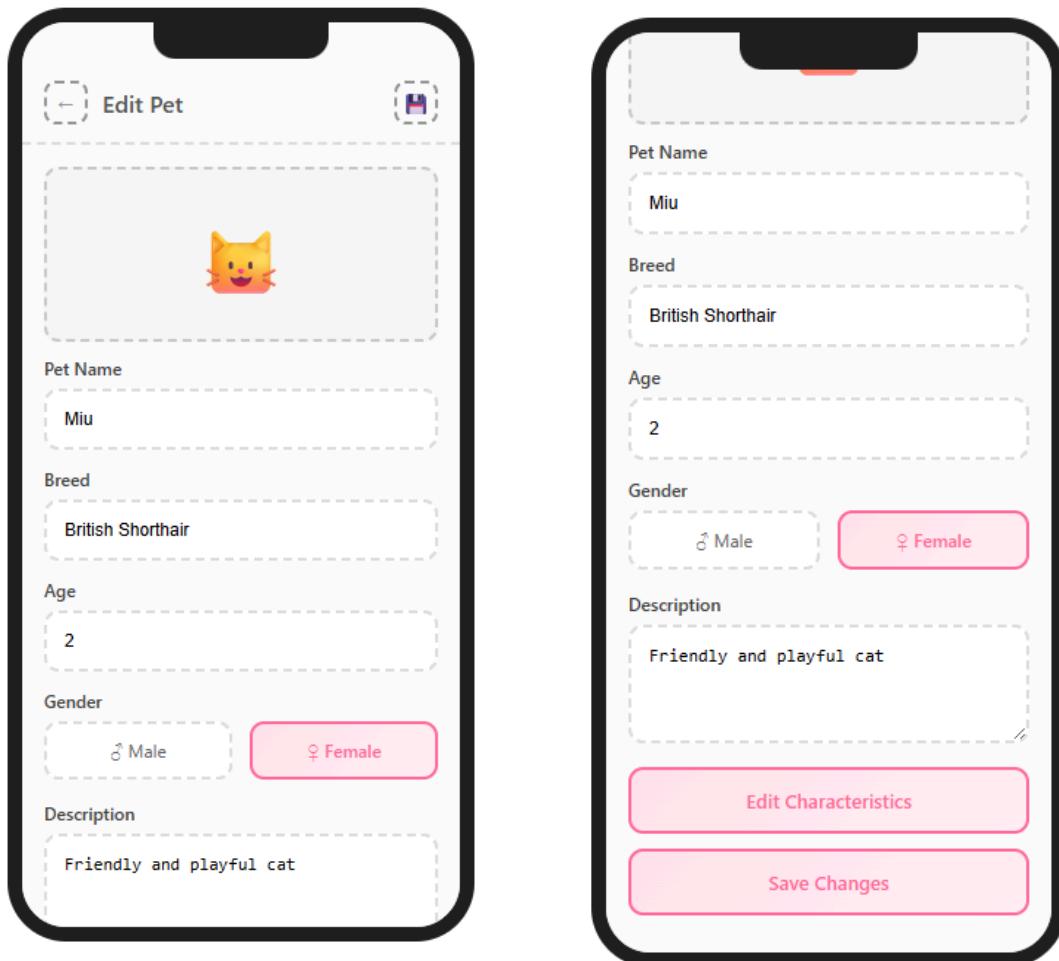
Use Case ID	UC033	Use Case Name	View Own Pet Profile
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows the user to view a list of all their registered pets and access options to add, edit, or manage them. The system retrieves the user's pets from the database and displays them in a card layout with the active pet's details prominently shown.		
Preconditions	1. User has successfully logged into the application. 2. User has navigated to the "Profile" tab.		
Postconditions	1. User sees their active pet's profile with photos, characteristics, and bio. 2. User sees a list of all registered pets with key details. 3. User can identify which pet is currently active for matching.		
Normal Sequence/Flow	1. User taps the "Profile" icon on the bottom navigation bar. 2. System displays loading indicator. 3. System retrieves user data, pets list, VIP status, and address from database. 4. System fetches characteristics and photos for active pet. 5. System displays Profile Screen showing: active pet photo carousel, name, gender, breed, age, bio, characteristics grid, and My Pets section. 6. Each pet card displays photo, name, gender, breed, and "Active" badge if selected. 7. User views the profile information.		

Alternative Sequences/Flows

- A1. If user has no pets → System displays "No pets found" and highlights "Add New Pet" button.
- A2. If load fails → System displays error alert.
- A3. If taps "Add New Pet" → System checks pet limit (max 3) before navigating to Add Pet screen.
- A4. If taps pet card → System navigates to Pet Profile Detail screen.
- A5. If taps "Set Active" → System updates active pet in database after confirmation.
- A6. If taps delete → System checks pet count (min 1) before deleting pet and updating active pet.
- A7. If taps edit → System navigates to Edit Pet or Edit Profile screen.

Screen Design: Edit Pet Profile Screen

Figure: Edit Pet Profile Screen Mockup



Edit Pet Profile Screen

This screen allows the user to:

- Update existing pet profile information including photos and basic details
- Edit pet characteristics and personality traits
- Save changes to pet profile

Field Description

Field Name	Description
Header Section	Contains back button for navigation and screen title "Edit Pet Profile"
Profile Photo Section	Displays current pet profile photo with edit button overlay and "Change pet photo" link to update main profile image
Pet Photos Section	Shows current uploaded photos (2/6) in grid layout with delete button on each photo and add button to upload additional photos up to maximum limit
Pet Information Form	Contains editable input fields for pet name, breed, age, gender selection buttons, and description text area
Action Buttons	Includes "Edit Characteristics" button to navigate to characteristics editing screen and "Save Changes" button to update all modified information

3.3.4. Delete Pet Profile

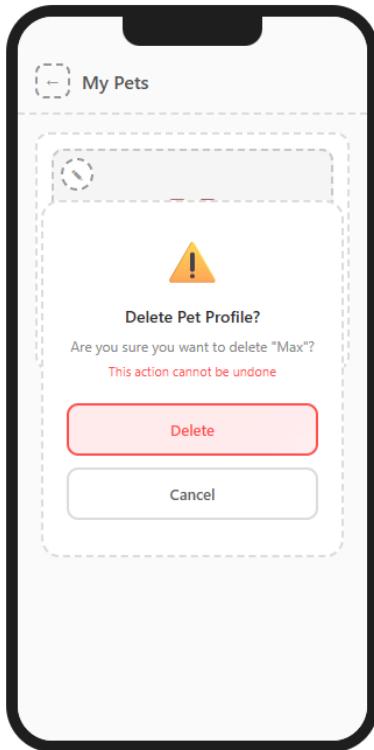
UC-034: Delete Pet Profile

Use Case ID	UC034	Use Case Name	Delete Pet Profile
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to delete their pet profile from the system. The system removes the pet data and automatically sets another pet as active if the deleted pet was the active one.		
Preconditions	1. User is logged into the system 2. User has at least two pet profiles. 3. User is on User Profile Screen.		
Postconditions	1. Pet profile is deleted from database. 2. If deleted pet was active, system sets another pet as active. 3. Pet list is updated on screen.		

Normal Sequence/Flow	<ol style="list-style-type: none"> 1. User navigates to User Profile Screen. 2. System displays a list of the User's pet profiles. 3. User taps the delete button (trash icon) on a pet card. 4. System displays a confirmation dialog. 5. User taps "Delete" to confirm. 6. System sends delete request to server. 7. System checks if deleted pet was active. 8. If deleted pet was active, system sets first remaining pet as active in database. 9. System reloads profile data from server. 10. System displays success message.
Alternative Sequences/Flows	<ol style="list-style-type: none"> A1. If user cancels delete → System closes confirmation dialog. System returns to User Profile Screen. A2. If user has only 1 pet → System shows warning: "Must have at least one pet." System does not allow deletion. A3. If delete request fails → System displays error alert with message. Pet remains in the list.

Screen Design: Delete Pet Profile

Figure: Delete Pet Profile Screen Mockup



Delete Pet Profile

This screen allows the user to:

- Delete a pet profile from their account
- Confirm deletion with warning about permanent action
- Cancel deletion if they change their mind

Field Description

Field Name	Description
Background Layer	Shows the pet list screen with pet cards visible behind the modal
Confirmation Dialog	Centered modal with warning icon, title, and confirmation message
Warning Message	Red text stating "This action cannot be undone"
Delete Button	Primary destructive action button with red styling
Cancel Button	Secondary button to abort deletion

3.3.5. Set Active Pet

UC-035: Set Active Pet

Use Case ID	UC035	Use Case Name	Set Active Pet
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to set a pet as the active profile for matching. The active pet is used for swiping, receiving likes, and matching with other pets.		
Preconditions	1. User is logged into the system. 2. User has at least one pet profile. 3. User is on User Profile Screen.		
Postconditions	1. Selected pet is set as active in database. 2. Previous active pet is deactivated. 3. Badge counts are refreshed for new active pet.		
Normal Sequence/Flow	1. User navigates to User Profile Screen. 2. System displays list of user's pet profiles with active pet highlighted. 3. User taps "Set Active" button on a pet card that is not currently active. 4. System displays confirmation dialog. 5. User taps "Confirm" to confirm. 6. System sends set-active request to server. 7. System invalidates local cache.		

	8. System reloads pets data from server. 9. System updates active pet state. 10. System refreshes badge counts for new active pet. 11. System displays success message.
Alternative Sequences/Flows	A1. If user cancels → System closes confirmation dialog. System returns to User Profile Screen. A2. If pet is already active → System displays info message: "This pet is already active." A3. If set-active request fails → System displays error alert. Active pet remains unchanged.

Screen Design: Set Active Pet Screen

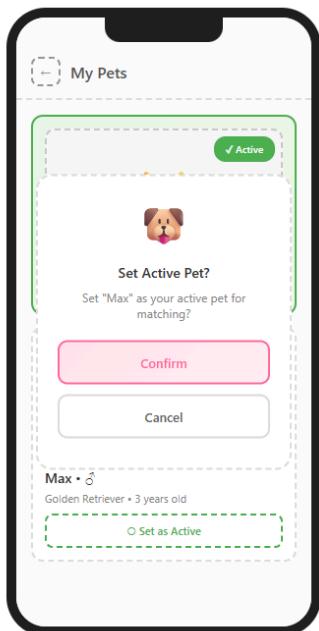


Figure: Set Active Pet Mockup

Set Active Pet Screen

This screen allows the user to:

- Set a specific pet as their active profile for matching
- Confirm which pet will be used for the matching system
- Cancel the action if they change their mind

Field Description

Field Name	Description

Background Layer	Shows the pet list with current active pet and other pets visible behind the modal
Confirmation Dialog	Centered modal with pet icon, title, and confirmation message
Pet Icon	Displays emoji representing the pet being set as active
Confirmation Message	Shows which pet will be set as active (e.g., "Set 'Max' as your active pet for matching?")
Confirm Button	Primary action button to set the selected pet as active
Cancel Button	Secondary button to abort the action

3.4 Matching & Discovery

3.4.1 Filter Pet Profiles

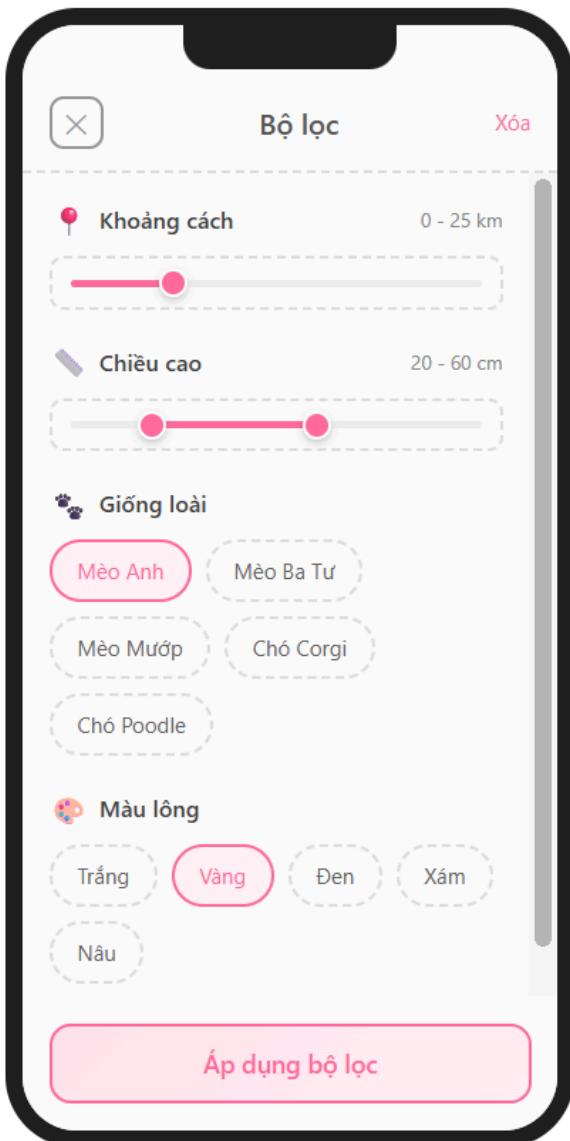
[UC-007:Filter Pet Profiles](#)

Use Case ID	UC007	Use Case Name	Set Match Filters
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to set preferences to filter pet profiles for matching. The system saves preferences and updates the Home screen to display pets matching the filter criteria.		
Preconditions	1. User is logged into the system. 2. User is on Home Screen.		
Postconditions	1. User preferences are saved to database. 2. Home screen displays pets matching the filter criteria.		
Normal Sequence/Flow	1. User taps the filter icon on the Home Screen header. 2. System navigates to Filter Screen. 3. System loads available filter attributes and user's existing preferences from server. 4. System displays filter options grouped by attribute type (breed, gender, distance, etc.). 5. User selects desired filter options by tapping chips or adjusting sliders. 6. User taps "Save Filter" button. 7. System validates and saves preferences to server. 8. System displays success message. 9. System navigates back to Home Screen. 10. Home Screen reloads with pets matching new filter criteria.		
Alternative Sequences/Flows	A1. If user taps "Clear All" → System clears all selected options. User can select new filters or save empty preferences.=		

- | | |
|--|---|
| | <p>A2. If user taps back button → System discards unsaved changes and navigates back to Home Screen without saving.</p> <p>A3. If save request fails → System displays error alert and user remains on Filter Screen.</p> |
|--|---|

Screen Design: Filter Pet Profiles Screen

Figure: Filter Pet Profiles Screen Mockup



Filter Pet Profiles

This screen allows the user to:

- Filter and search for pet profiles based on various criteria
- Apply multiple filter options to narrow down results

- View filtered pet profiles matching selected criteria

Field Description

Field Name	Description
Header Section	Contains back button for navigation and screen title "Filter Pets"
Search Bar	Text input field to search pets by name or breed
Species Filter	Selection buttons or dropdown to filter by pet type (Dog, Cat, etc.)
Gender Filter	Radio buttons or toggles to filter by Male/Female
Age Range Filter	Slider or input fields to set minimum and maximum age
Size Filter	Options to filter by pet size (Small, Medium, Large)
Location Filter	Distance radius selector or location input
Apply Button	Primary action button to apply selected filters and show results
Reset Button	Secondary button to clear all filters and reset to default

3.4.2 View Pet Profile Detail

UC-008: View Pet Profile Detail

Use Case ID	UC008	Use Case Name	View Pet Detail
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to view complete information about a pet from the match feed, including photos, characteristics, owner information, and match percentage details.		
Preconditions	1. User is logged into the application. 2. User is viewing Match Feed Screen or other screens with pet cards. 3. User has tapped the info button on a pet card.		
Postconditions	1. Pet detail information is displayed successfully. 2. User can navigate back or perform actions (Send Match Request/Block).		

Normal Sequence/Flow	<ol style="list-style-type: none"> 1. User taps the info button on a pet card in Match Feed Screen. 2. System displays loading indicator. 3. System retrieves User ID and active pet ID from storage. 4. System fetches pet data, photos, characteristics, and owner information from database. 5. System fetches match details (match percentage and matched attributes) if viewing other user's pet. 6. System displays Pet Detail Screen with photo carousel, pet info, location, match percentage, bio, characteristics, owner card, and action buttons. 7. User scrolls to view all information.
Alternative Sequences/Flows	<ol style="list-style-type: none"> A1. If user taps "Send Match Request" → System sends like request. If mutual match, shows "It's a Match!" alert, otherwise shows success message and navigates back. A2. If user taps "Block" → System shows confirmation. User confirms, system sends block request and navigates to Home Screen. A3. If user taps photo → System navigates previous/next photo in carousel. A4. If user taps match badge → System opens Match Details Modal. A5. If viewing own pet → System displays "Edit" button. User taps to navigate to Edit Pet screen. A6. If user taps back button → System navigates back to previous screen. A7. If load fails → System displays error alert and navigates back.

Screen Design: Pet Detail Preview Screen

Figure: Pet Detail Preview Screen Mockup



Pet Detail Screen

This screen allows the user to:

- View complete profile information of a specific pet
- Learn about pet's characteristics, personality, and physical attributes
- View owner information and their other pets
- Send match request or block user

Field Description

Field Name	Description
Quick Info Section	Displays pet's basic information in circular badge format showing breed, age, and gender at the top of screen
About Section	Contains section title and detailed text description of pet's personality, interests, and what they are looking for in a match
Characteristics Section	Shows pet attributes in pill-shaped badges including energy level, weight, personality traits, and lifestyle preferences
Owner Section	Displays owner's profile photo, name, premium membership badge, and arrow button to navigate to owner's full profile
Other Pets Section	Horizontal scrollable row showing circular thumbnails of other pets owned by the same owner with pet names below
Action Buttons	Contains primary "Send Match Request" button with heart icon and secondary "Block User" text link in pink/red color

3.4.3 Swipe Pet Profile

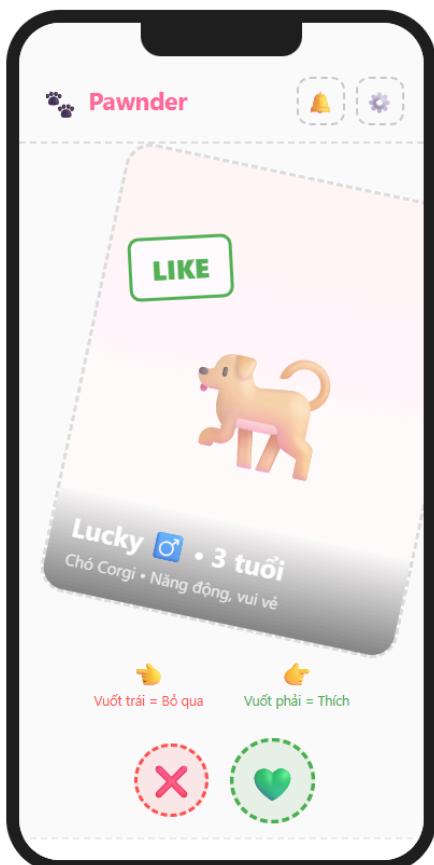
UC-009: Swipe Pet Profile

Use Case ID	UC009	Use Case Name	Swipe Pet Profile
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to swipe pet profiles to express interest or skip matching opportunities. The system sends like requests and handles match notifications.		
Preconditions	1. User is logged into the system. 2. User has an active pet set. 3. User is on Home Screen with pet cards displayed.		
Postconditions	1. If swiped right: Like request is sent to server.		

	<p>2. If mutual like: Match is created and both users are notified.</p> <p>3. Current pet card is removed and next card is displayed.</p>
Normal Sequence/Flow	<p>1. User views a pet card on the Home Screen.</p> <p>2. User swipes the card to the right (like) or to the left (pass).</p> <p>3. System animates the card flying off the screen in the swipe direction.</p> <p>4. System triggers haptic feedback.</p> <p>5. If user swipes right, system sends like request to server with fromUserId, toUserId, fromPetId, and toPetId.</p> <p>6. System resets card position and advances to next pet card in the stack.</p> <p>7. If mutual like is detected, system receives match notification via SignalR and displays match modal.</p>
Alternative Sequences/Flows	<p>A1. If user taps Like/Pass button → System triggers swipe animation and continues from step 3.</p> <p>A2. If swipe not far enough → System animates card back to center.</p> <p>A3. If no more pets → System displays empty state with "No more suitable pets" and "Reload" button.</p> <p>A4. If match limit reached → System shows limit modal with upgrade option.</p>

Screen Design: Swipe Pet Profile

Figure: Swipe Pet Profile Screen Mockup



Swipe Pet Profile

This screen allows the user to:

- View pet profiles one at a time in a card-based interface
- Swipe right to like or left to pass on pet profiles
- Tap action buttons to like, pass, or super like pets
- View detailed pet information and photos

Field Description

Field Name	Description
Header Section	Contains app logo, filter button, and settings/profile access
Pet Card	Large card displaying pet photo, name, breed, age, and distance
Photo Gallery	Swipeable photo carousel showing multiple pet images with indicators
Pet Information	Displays pet name, age, breed, and location distance
Info Button	Button to view full pet profile details and characteristics
Action Buttons	Bottom row with Pass (X), Super Like (Star), and Like (Heart) buttons
Swipe Gesture Area	Touch-enabled area allowing left/right swipe gestures for quick decisions

3.4.4. View Match List

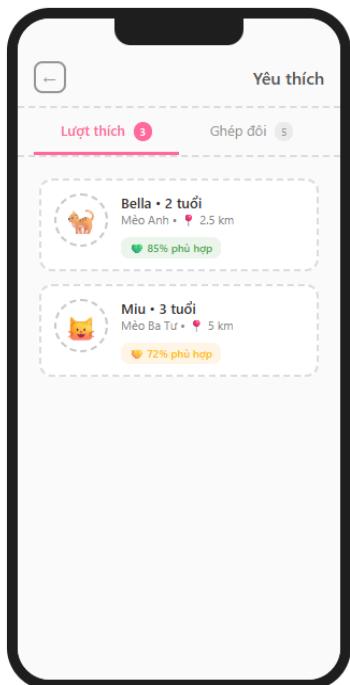
UC-039: View Match List

Use Case ID	UC039	Use Case Name	View Match List
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to view matched users or pets. Users can see likes received and existing matches, with options to accept, reject, chat, or unmatch.		
Preconditions	1. User is logged into the system. 2. User has an active pet set.		
Postconditions	1. User can view list of likes received and matches. 2. User can interact with each item (accept, reject, chat, unmatch).		

Normal Sequence/Flow	<ol style="list-style-type: none"> User taps the "Favorite" tab on the bottom navigation bar. System navigates to Favorite Screen and resets favorite badge count. System fetches likes received from server, filtered by active pet. System fetches match details (match percentage) for each pet. System displays two tabs: "Likes" and "Matches". System shows pet cards with photo carousel, name, gender, age, breed, owner name, and match percentage. In Likes tab, each card shows Pass (X) button and Match (heart) button. In Matches tab, each card shows Chat button and Unmatch option.
Alternative Sequences/Flows	<ol style="list-style-type: none"> If no likes/matches → System displays empty state with button to navigate to Home Screen. If user pulls down to refresh → System reloads likes and matches from server. If user taps pet card → System navigates to Pet Profile Detail Screen. If user taps Match button on a like → System sends like request, creates match, and moves card to Matches tab. If user taps Pass button on a like → System removes card from Likes list. If user taps Chat button on a match → System navigates to Chat Screen. If user taps Unmatch option → System shows confirmation. User confirms, system sends unmatch request and removes card.

Screen Design: View Match List

Figure: View Match List Mockup



View Match List

This screen allows the user to:

- View all matched pet profiles in a list or grid format
- See match percentage and compatibility scores
- Access chat with matched pet owners • Filter and sort matches by various criteria

Field Description

Field Name	Description
Header Section	Contains back button for navigation and screen title "Matches"
Filter/Sort Options	Tabs or buttons to filter matches (All, New, Favorites) and sort by match percentage or date
Match Cards	Grid or list of matched pet profiles showing pet photo, name, breed, and match percentage
Match Percentage Badge	Visual indicator showing compatibility score (e.g., 85% match)
Chat Button	Button or icon on each match card to open chat conversation
Favorite Icon	Heart icon to mark matches as favorites

3.4.5. Respond to Match Request

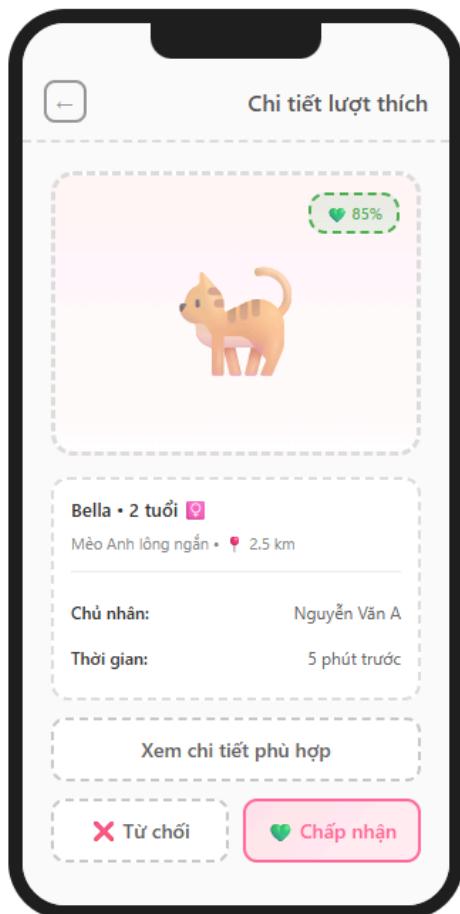
UC-042: Respond to Match Request

Use Case ID	UC042	Use Case Name	Respond to Match Request
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to accept or reject a match request (like received). When accepted, both users become matched and can chat.		
Preconditions	1. User is logged into the system. 2. User has an active pet set. 3. User has received at least one like/match request from another user.		
Postconditions	1. Match request status is updated in the system. 2. If accepted: both users become matched and can chat. 3. If declined: request is removed from the list.		
Normal Sequence/Flow	1. User views a pending match request card in the "Likes" tab. 2. User taps the heart button (Match) to accept the request. 3. System sends respond request with action "match" to server. 4. System updates match status to "matched".		

	<p>5. System displays Match Success modal with pet photo, name, owner name, and "Send Message" button.</p> <p>6. System moves pet card to "Matches" tab.</p> <p>7. System invalidates likes and chats cache.</p>
Alternative Sequences/Flows	<p>A1. If user declines match request → User taps X button (Pass). System sends respond request with action "pass" to server, removes card, and invalidates likes cache.</p> <p>A2. If user views pet profile before responding → User taps on pet card. System navigates to Pet Profile Detail Screen. User returns to Favorite Screen.</p> <p>A3. If user views match percentage details → User taps match percentage badge. System displays Match Details modal showing matched attributes.</p> <p>A4. If user navigates photo carousel → User taps left or right side of pet photo. System displays previous or next photo and updates dots.</p>

Screen Design: Respond to Match Request

Figure: Respond to Match Request Screen Mockup



Respond to Match Request

This screen allows the user to:

- View incoming match requests from other pet owners
- Accept or decline match requests
- See detailed information about the requesting pet and owner • Make informed decisions about potential matches

Field Description

Field Name	Description
Header Section	Contains back button for navigation and screen title "Match Request"
Pet Profile Card	Displays requesting pet's photo, name, breed, age, and location
Match Percentage	Shows compatibility score between the two pets
Pet Details Section	Shows additional information like personality traits, size, and characteristics
Owner Information	Brief info about the pet owner (name, verification status)
Accept Button	Primary action button with heart icon to accept the match request
Decline Button	Secondary action button with X icon to decline the match request
View Full Profile Link	Link to view complete pet profile before making decision

3.4.6. Unmatch User

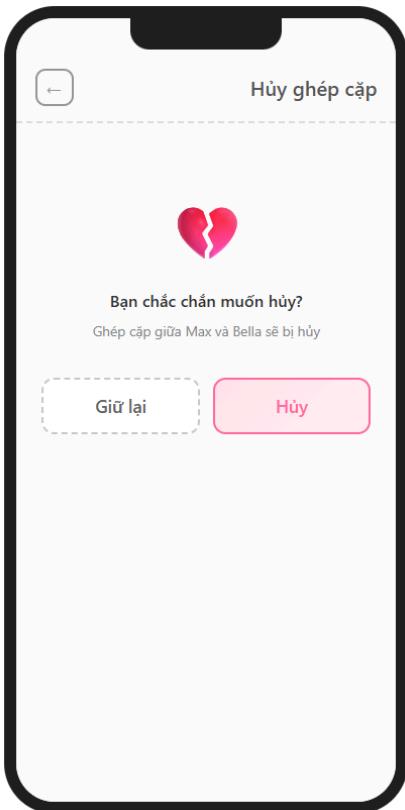
UC-043: Unmatch User

Use Case ID	UC043	Use Case Name	Unmatch User
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to remove a matched connection with another pet owner. The match is removed from both users' lists and chat history becomes inaccessible.		
Preconditions	1. User is logged into the system. 2. User has at least one existing match.		
Postconditions	1. Match is removed from both users' match lists. 2. Chat history between users is no longer accessible.		
Normal Sequence/Flow	1. User navigates to Favorite Screen and selects "Matches" tab. 2. User taps "Unmatch" button on a matched pet card. 3. System displays confirmation dialog with pet name. 4. User taps "Confirm" to proceed.		

	5. System sends unmatch request to server. 6. System removes match from list. 7. System displays success message.
Alternative Sequences/Flows	A1. If user cancels unmatch → User taps "Cancel" button. System closes confirmation dialog. A2. If unmatch request fails → System displays error alert. Match remains in list.

Screen Design: Unmatch User Screen

Figure: Unmatch User Screen Mockup



Unmatch User Screen

This screen allows the user to:

- Remove a matched connection with another user
- Confirm unmatch action with warning about consequences
- Optionally provide a reason for unmatched • Cancel the action if they change their mind

Field Description

Field Name	Description
Background Layer	Shows the match profile or chat screen behind the modal

Confirmation Dialog	Centered modal with warning icon and unmatched confirmation message
Warning Message	Text explaining that unmatched will remove the connection and chat history
Reason Selection	Dropdown or radio buttons to select reason for unmatched
Unmatch Button	Primary destructive action button with red styling to confirm unmatched
Cancel Button	Secondary button to abort the unmatched action

3.5 Communication

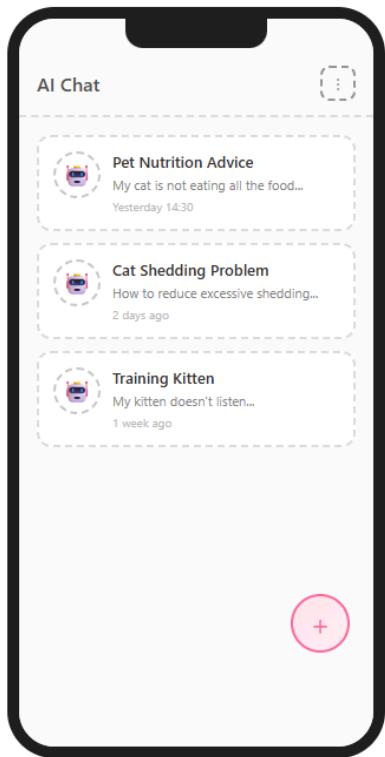
3.5.1 View AI Chat List

UC-010: View AI Chat List

Use Case ID	UC010	Use Case Name	View AI Chat List
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to view a list of AI chat conversations. Users can create new chats, rename existing chats, or delete chat sessions.		
Preconditions	1. User is logged into the system.		
Postconditions	1. User can view all their AI chat sessions. 2. User can create, rename, or delete chat sessions.		
Normal Sequence/Flow	1. User navigates to AI Chat List Screen. 2. System fetches all AI chat sessions for user. 3. System displays list of chat sessions with title, last message preview, timestamp, and message count. 4. System shows "New Chat" button at top. 5. System shows "Expert Confirmation" button to request expert review.		
Alternative Sequences/Flows	A1. If user taps "New Chat" → System creates new chat session and navigates to AI Chat Screen. A2. If user long-presses on chat item → System displays rename modal. User enters new title and saves. System updates chat title. A3. If no chat sessions exist → System displays empty state: "No conversations yet." A4. If user taps chat item → System navigates to AI Chat Screen with selected conversation. A5. If user swipes to delete chat → System shows delete confirmation. User confirms, system deletes chat session.		

Screen Design: Chat List Screen

Figure: Chat List Screen Mockup



Messages Screen

This screen allows the user to:

- View list of all AI chat conversations
- Access previous chat sessions with AI assistant
- See chat history and recent messages
- Start new AI chat conversations

Field Description

Field Name	Description
Header Section	Contains back button for navigation and screen title "AI Chat"
New Chat Button	Primary button to start a new AI conversation
Chat Item	Scrollable list of previous AI chat sessions with timestamps
Chat Item	Each item shows chat preview, last message, and date/time
AI Avatar/Icon	Icon or image representing the AI assistant
Empty State	Message and illustration shown when no chat history exists
Search Bar	Input field to search through chat history

3.5.2 Chat with AI

UC-011: Chat with AI

Use Case ID	UC011	Use Case Name	Chat with AI
Primary Actors	User (Pet Owner)	Secondary Actors	AI Service
Description	Allows users to interact with an AI assistant for pet-related consultation. The system saves chat history and updates token usage.		
Preconditions	1. User is logged into the system. 2. User has an existing AI chat session.		
Postconditions	1. User receives AI response to their question. 2. Chat history is saved. 3. Token usage is updated.		
Normal Sequence/Flow	1. User opens an AI chat session. 2. System loads chat history and displays previous messages. 3. User types a question in the input field. 4. User taps "Send" button. 5. System displays user message immediately. 6. System shows typing indicator. 7. System sends question to AI service. 8. System receives AI response. 9. System displays AI response with timestamp. 10. System updates token usage.		
Alternative Sequences/Flows	A1. If user taps suggestion chip → System auto-sends suggested question and continues from step 5. A2. If user taps "Ask Expert" on AI response → System displays expert selection modal. User selects expert and sends request. System shows success message. A3. If new chat session → System displays welcome message with suggestions and continues from step 3. A4. If AI service fails → System displays error message. User can retry sending message.		

Screen Design: Chat with AI Screen

Figure: Chat with AI Screen Mockup



Chat with AI Screen

This screen allows the user to:

- Have real-time conversations with AI assistant about pet care
- Ask questions and receive AI-generated responses
- View chat history within the conversation
- Send text messages and receive helpful pet-related advice

Field Description

Field Name	Description
Header Section	Contains back button, AI assistant name/title, and menu options
AI Avatar	Icon or image representing the AI assistant at top of screen
Chat Messages Area	Scrollable area displaying conversation history with user and AI messages
User Message Bubble	Right-aligned message bubbles showing user's sent messages
AI Message Bubble	Left-aligned message bubbles showing AI responses
Typing Indicator	Animated dots showing when AI is generating response
Message Input Field	Text input box at bottom for typing messages
Send Button	Button to send typed message to AI assistant

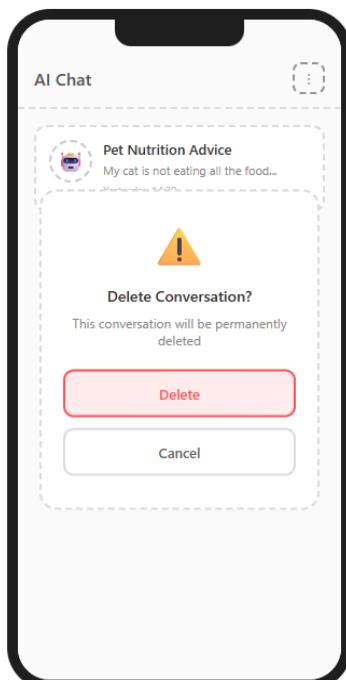
3.5.3. Delete AI Conversation

UC-012: Delete AI Conversation

Use Case ID	UC012	Use Case Name	Delete AI Conversation
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to delete an AI conversation from their history. The chat session and all messages are permanently removed.		
Preconditions	1. User is logged into the system. 2. User has at least one AI chat session.		
Postconditions	1. Chat session and all messages are permanently deleted. 2. Chat is removed from the list.		
Normal Sequence/Flow	1. User views AI Chat List Screen. 2. User taps delete (trash) button on a chat item. 3. System displays confirmation dialog with chat title. 4. User taps "Delete" to confirm. 5. System sends delete request to server. 6. System removes chat from list.		
Alternative Sequences/Flows	A1. If user cancels delete → User taps "Cancel" button. System closes confirmation dialog. A2. If delete request fails → System displays error alert. Chat remains in list.		

Screen Design: Delete AI Conversation Screen

Figure: Delete AI Conversation Screen Mockup



Delete AI Conversation

This screen allows the user to:

- Delete an AI chat conversation from history
- Confirm deletion with warning about permanent removal
- Cancel deletion if they change their mind

Field Description

Field Name	Description
Background Layer	Shows the AI chat list or conversation screen behind the modal
Confirmation Dialog	Centered modal with warning icon and deletion confirmation message
Warning Message	Text stating "This action cannot be undone" and that chat history will be permanently deleted
Delete Button	Primary destructive action button with red styling to confirm deletion
Cancel Button	Secondary button to abort the deletion action

3.5.4. Submit AI Conversation for Review

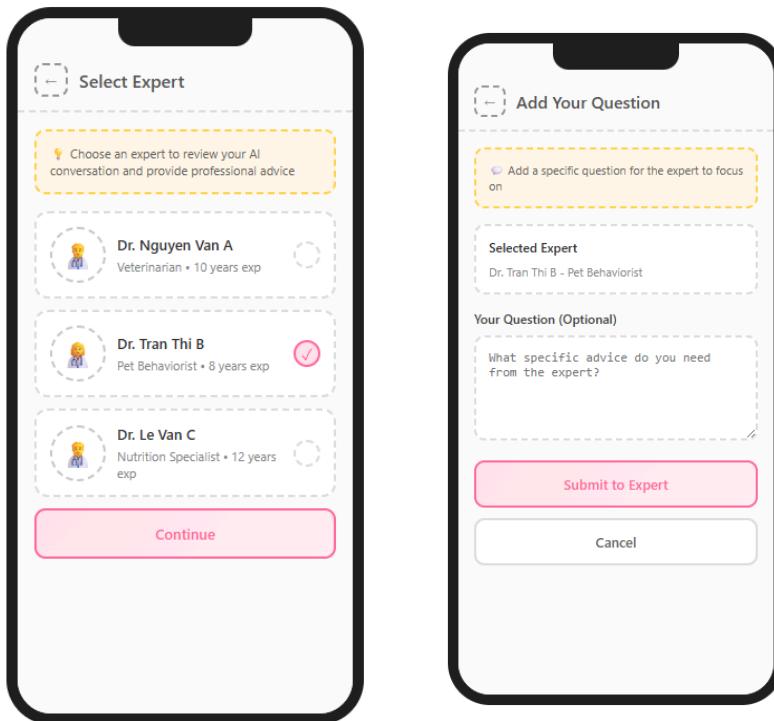
UC-013: Submit AI Conversation for Review

Use Case ID	UC013	Use Case Name	Submit AI Conversation for Review
Primary Actors	User (Pet Owner)	Secondary Actors	Expert
Description	Allows users to submit AI conversations for expert review. The expert receives a notification and can provide professional verification.		
Preconditions	1. User is logged into the system. 2. User has an AI chat with at least one AI response. 3. At least one expert is available in the system.		
Postconditions	1. Expert confirmation request is created. 2. Expert receives notification to review. 3. AI response is marked as "Sent to Expert".		

Normal Sequence/Flow	<ol style="list-style-type: none"> User views an AI response in the chat. User taps "Ask Expert" button on the AI response. System loads and displays list of available experts. User selects an expert from the list. System shows question input form. User enters specific question for the expert. User taps "Send" to submit. System creates expert confirmation request with chatAId, expertId, and user question. System marks AI response as "Sent to Expert". System displays success message.
Alternative Sequences/Flows	<p>A1. If user changes selected expert → User taps "Change" button. System returns to expert selection list.</p> <p>A2. If user cancels request → User taps outside modal or close button. System closes modal.</p> <p>A3. If no experts available → System displays message: "No experts available." System closes modal.</p> <p>A4. If submit request fails → System displays error alert. User can retry submission.</p>

Screen Design: Submit AI Conversation for Review

Figure:Submit AI Conversation for Review Mockup



Submit AI Conversation for Review

This screen allows the user to:

- Report problematic or inappropriate AI responses

- Submit AI conversation for admin review
- Provide feedback about AI behavior or accuracy
- Add optional comments explaining the issue

Field Description

Field Name	Description
Header Section	Contains back button for navigation and screen title "Report AI Conversation"
Conversation Preview	Shows snippet or summary of the AI conversation being reported
Reason Selection	Radio buttons or dropdown to select reason for review (Inappropriate, Inaccurate, Offensive, Other)
Comments Field	Text area for user to provide additional details or context about the issue
Submit Button	Primary action button to send the report to admin review
Cancel Button	Secondary button to abort the submission

3.5.5. View Expert Feedback

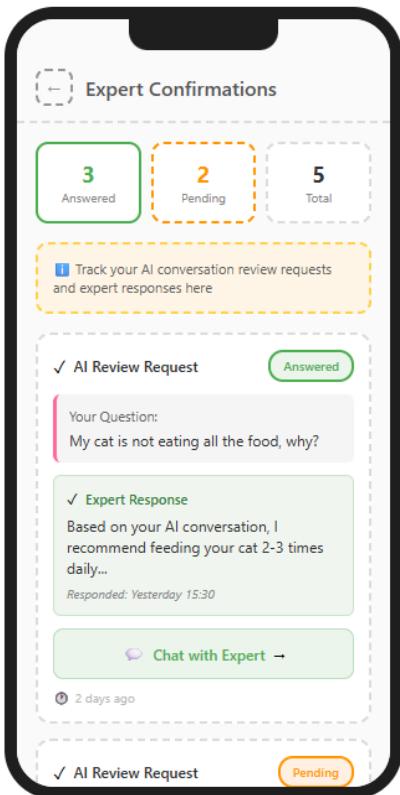
UC-014: View Expert Feedback

Use Case ID	UC014	Use Case Name	View Expert Feedback
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to view feedback provided by experts on their AI conversations. Users can see request statuses and filter by answered or pending.		
Preconditions	1. User is logged into the system.		
Postconditions	1. User can view all expert confirmation requests and their statuses.		
Normal Sequence/Flow	1. User navigates to Expert Confirmation Screen. 2. System fetches all expert confirmation requests for user. 3. System displays statistics: Answered count, Pending count, Total count. 4. System displays list of requests with status badge, user question, expert response (if answered), and timestamp. 5. User can tap filter tabs to view: All, Answered, or Pending requests.		
Alternative Sequences/Flows	A1. If user taps "Chat with Expert" on answered request → System creates or retrieves existing expert chat and navigates to Expert Chat Screen		

	<p>A2. If user pulls down to refresh → System reloads requests from server and continues from step 3.</p> <p>A3. If user taps stat card to filter → System filters list by selected status and continues from step 4.</p> <p>A4. If no requests exist → System displays empty state: "No expert confirmations yet."</p> <p>A5. If user taps request item → System expands item to show full details.</p>
--	--

Screen Design: View Expert Feedback Screen

Figure: View Expert Feedback Screen Mockup



View Expert Feedback Screen

This screen allows the user to:

- View feedback and advice from pet care experts
- Read expert responses to submitted questions
- See expert ratings and credentials
- Access detailed expert recommendations

Field Description

Field Name	Description
Header Section	Contains back button for navigation and screen title "Expert Feedback"
Expert Profile Card	Shows expert's photo, name, credentials, and specialization
Question Summary	Displays the original question or topic submitted by user
Expert Response	Detailed feedback and advice provided by the expert
Response Date	Timestamp showing when expert provided the feedback
Rating Section	Star rating or thumbs up/down to rate the expert's response
Follow-up Button	Button to ask additional questions or request clarification

3.5.6. View Expert Chat List

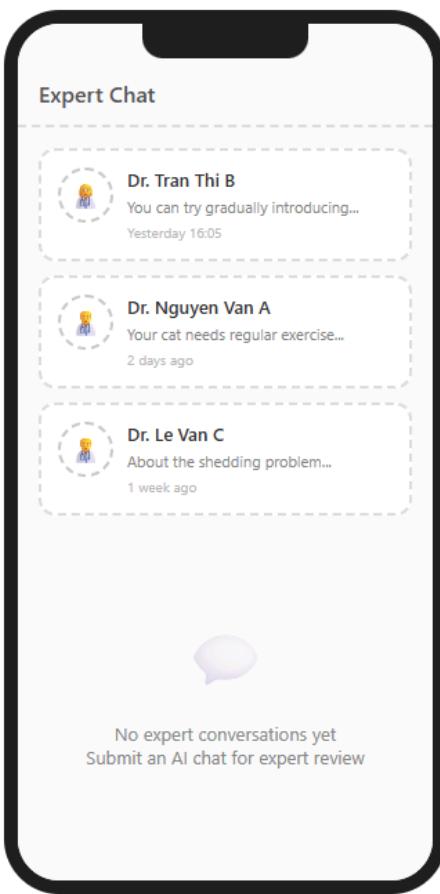
UC-015: View Expert Chat List

Use Case ID	UC015	Use Case Name	View Expert Chat List
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to view a list of expert chat conversations. Users can see existing chats with experts and start new conversations with available experts.		
Preconditions	1. User is logged into the system.		
Postconditions	1. User can view all their expert chat sessions. 2. User can start new chat with available experts.		
Normal Sequence/Flow	1. User navigates to Expert Chat List Screen. 2. System fetches all expert chat sessions for user. 3. System sorts chats by most recent message time. 4. System displays list of expert chats with: expert avatar, name, last message preview, timestamp, and unread badge. 5. System shows "New Chat" button to start conversation with expert. 6. System connects to SignalR for real-time message updates.		
Alternative Sequences/Flows	A1. If user taps "New Chat" → System displays expert selection, user selects expert, system creates/retrieves chat and navigates to Expert Chat Screen. A2. If user taps chat item → System navigates to Expert Chat Screen. A3. If user pulls down to refresh → System reloads expert chats from server.		

- | | |
|--|---|
| | <p>A4. If no chats exist → System displays empty state: "No expert conversations yet" with "Start Chat" button.</p> <p>A5. If new message via SignalR → System updates chat item, moves it to top, and shows unread badge.</p> <p>A6. If create chat fails → System displays error alert.</p> |
|--|---|

Screen Design: View Expert Chat List Screen

Figure: View Expert Chat List Screen Mockup



View Expert Chat List Screen

This screen allows the user to:

- View list of all conversations with pet care experts
- Access ongoing and past expert consultations
- See unread messages and response status
- Start new expert consultation requests

Field Description

Field Name	Description
Header Section	Contains back button for navigation and screen title "Expert Chats"

New Consultation Button	Primary button to request a new expert consultation
Chat List	Scrollable list of expert conversations with status indicators
Chat Item	Each item shows expert photo, name, specialization, last message preview, and timestamp
Status Badge	Indicator showing chat status (Active, Waiting for Response, Completed)
Unread Count Badge	Number badge showing unread messages from expert
Filter Tabs	Tabs to filter by status (All, Active, Completed)
Empty State	Message and illustration shown when no expert chats exist

3.5.6. Chat with Expert

UC-016: Chat with Expert

Use Case ID	UC016	Use Case Name	Chat with Expert
Primary Actors	User (Pet Owner)	Secondary Actors	Expert
Description	Allows users to chat with veterinary experts in real-time. Messages are sent via SignalR and chat history is saved.		
Preconditions	1. User is logged into the system. 2. User has an existing expert chat session.		
Postconditions	1. Messages are sent and received in real-time. 2. Chat history is saved.		
Normal Sequence/Flow	1. User opens Expert Chat Screen from Expert Chat List or notification. 2. System connects to SignalR for real-time messaging. 3. System loads chat history from server. 4. System displays messages with expert badge for expert messages. 5. User types message in input field. 6. User taps "Send" button. 7. System displays message with "sending" status. 8. System sends message to server via SignalR. 9. System updates message status to "sent". 10. When expert replies, system receives message via SignalR and displays it.		
Alternative Sequences/Flows	A1. If user pulls down to refresh → System reloads messages from server and continues from step 4. A2. If real-time message received → System receives message via SignalR, adds to chat, and auto-scrolls to bottom.		

	A3. If send message fails → System displays message with "failed" status. User can tap to retry. A4. If SignalR connection lost → System attempts to reconnect automatically and shows status indicator.
--	---

Screen Design: Chat with Expert Screen

Figure: Chat with Expert Screen Mockup



Chat with Expert Screen

This screen allows the user to:

- Have real-time conversations with verified pet care experts
- Send messages and receive professional advice
- Share photos or documents related to pet concerns
- View expert's credentials and specialization

Field Description

Field Name	Description
Header Section	Contains back button, expert name, specialization, and online status indicator

Expert Profile Info	Shows expert's photo, credentials, and verification badge
Chat Messages Area	Scrollable area displaying conversation history between user and expert
User Message Bubble	Right-aligned message bubbles showing user's sent messages
Expert Message Bubble	Left-aligned message bubbles showing expert's responses
Typing Indicator	Shows when expert is typing a response
Message Input Field	Text input box at bottom for typing messages
Attachment Button	Button to attach photos or documents to message
Send Button	Button to send typed message to expert

3.5.6. View Match Conversation Detail

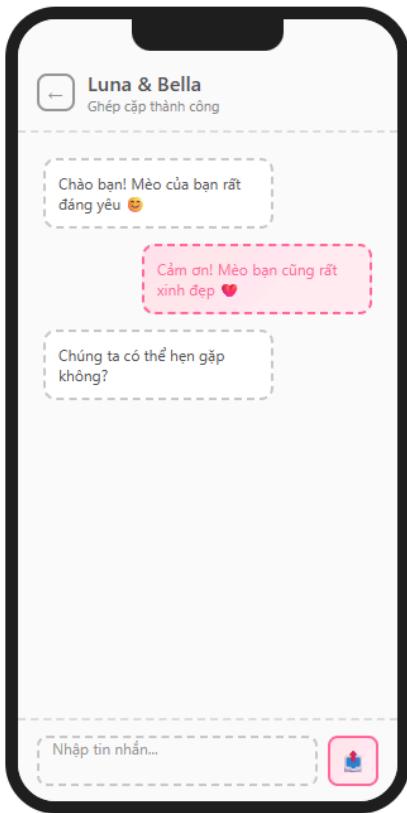
UC-040: View Match Conversation Detail

Use Case ID	UC040	Use Case Name	View Match Conversation
Primary Actors	User	Secondary Actors	Matched User
Description	Allows users to view and interact with match conversation details. Messages are displayed with read status and real-time messaging is enabled.		
Preconditions	1. User is logged into the system. 2. User has an existing match with another user.		
Postconditions	1. Messages are displayed with read status. 2. Chat is marked as read. 3. Real-time messaging is enabled.		
Normal Sequence/Flow	1. User taps on a chat from Chat List Screen. 2. System connects to SignalR for real-time messaging. 3. System loads chat messages and user/pet avatars. 4. System displays messages grouped by date with text, timestamp, read status, and sender avatar. 5. System shows online/offline status of other user. 6. User types message in input field. 7. User taps "Send" button. 8. System displays message with "sending" status. 9. System sends message to server via SignalR. 10. System updates message status to "sent". 11. When other user replies, system receives message via SignalR and displays it.		

Alternative Sequences/Flows	<p>A1. If user taps calendar icon → System navigates to Create Appointment Screen.</p> <p>A2. If user taps menu > "View Profile" → System fetches other user's pet info and navigates to Pet Profile Screen.</p> <p>A3. If user taps menu > "Block"/"Unmatch" → System shows confirmation dialog, executes request, and navigates to Chat List Screen.</p> <p>A4. If user long-presses message → System displays message menu (e.g., Report).</p> <p>A5. If other user is typing → System receives typing event via SignalR and displays "Typing..." status.</p>
------------------------------------	--

Screen Design: View Match Conversation Detail Screen

Figure: View Match Conversation Detail Screen Mockup



View Match Conversation Detail Screen

This screen allows the user to:

- View detailed information about a matched pet and owner
- See full pet profile and characteristics
- Access conversation history and chat
- View match percentage and compatibility details

Field Description

Field Name	Description
Header Section	Contains back button, matched pet name, and menu options (unmatch, report)
Pet Photo Gallery	Swipeable carousel showing all photos of the matched pet
Match Percentage Badge	Displays compatibility score between the two pets
Pet Information Section	Shows pet name, breed, age, gender, size, and location
Personality Traits	Lists pet's characteristics and personality attributes
Owner Information	Brief details about the pet owner with verification status
Chat Button	Primary action button to open or continue conversation
Create Appointment Button	Button to schedule a meetup with the matched pet
Additional Actions	Options like favorite, share profile, or unmatch

3.5.7. Chat with Matched User

UC-041: Chat with Matched User

Use Case ID	UC041	Use Case Name	Send Message with BadWord Filter
Primary Actors	User	Secondary Actors	BadWord Filter System
Description	Allows users to send messages to matched users. Messages are filtered for inappropriate content before being delivered in real-time.		
Preconditions	1. User is logged into the system. 2. User is viewing a match conversation. 3. SignalR connection is established.		
Postconditions	1. Message is sent and displayed in conversation. 2. Other user receives message in real-time.		
Normal Sequence/Flow	1. User types message in input field. 2. System detects typing and sends typing indicator to other user via SignalR. 3. User taps "Send" button. 4. System adds message to chat with "sending" status and clears input field. 5. System sends message to server. 6. Server filters message for bad words. 7. System updates message status to "sent". 8. Other user receives message via SignalR in real-time.		

Alternative Sequences/Flows	<p>A1. If message received → System receives message via SignalR, adds to chat, auto-scrolls, and marks as read.</p> <p>A2. If other user is typing → System receives typing event and displays animated typing indicator (disappears after 3s of inactivity).</p> <p>A3. If send fails → System displays message with "failed" status. User can tap to retry.</p> <p>A4. If message contains bad words → Server filters inappropriate content and system displays filtered message.</p>
------------------------------------	--

Screen Design: Chat with Matched User

Figure: Chat with Matched User Screen Mockup



Chat with Matched User Screen

This screen allows the user to:

- Have real-time conversations with matched pet owners
- Send text messages and share photos
- Create appointment invitations within chat
- View chat history and message status

Field Description

Field Name	Description
Header Section	Contains back button, matched user's name, pet photo, and menu options
Match Info Banner	Shows match percentage and pet names at top of chat
Chat Messages Area	Scrollable area displaying conversation history between matched users
User Message Bubble	Right-aligned message bubbles showing sent messages with read status
Typing Indicator	Shows when the other user is typing
Message Input Field	Text input box at bottom for typing messages
Send Button	Button to send typed message

3.6 Payment & Subscription

3.6.1. View Subscription Plans

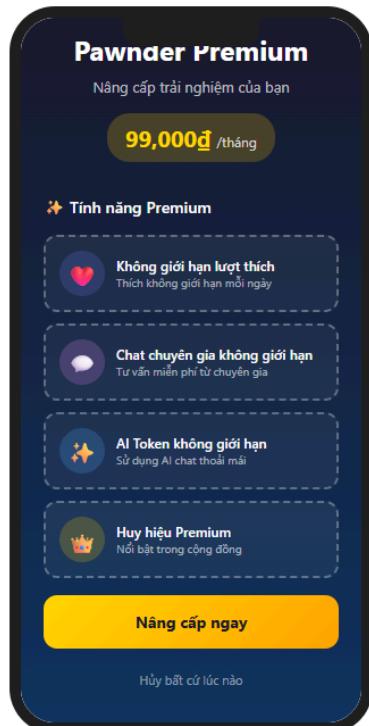
[UC-028: View Subscription Plans](#)

Use Case ID	UC028	Use Case Name	Process QR Payment
Primary Actors	User (Pet Owner)	Secondary Actors	Payment Gateway (VietQR), Banking System
Description	Displays QR code for VietQR payment. Users scan the QR code using their banking app to complete payment for premium subscription.		
Preconditions	1. User is logged into the application. 2. User has selected a premium plan from Premium Plans Screen. 3. User has valid planId, amount, and duration parameters.		
Postconditions	1. QR code is generated and displayed. 2. User completes payment via banking app. 3. Payment history is created in database. 4. User subscription is activated. 5. User is navigated to Home Screen.		

Normal Sequence/Flow	<ol style="list-style-type: none"> 1. User arrives from Premium Plans Screen with selected plan details. 2. System displays loading state: "Creating QR code..." 3. System generates QR code (280x280px) with payment information. 4. System displays Payment Info Card with service package, duration, and amount. 5. System displays QR code. 6. System shows 3-step instructions: Open banking app → Select QR scan → Scan and confirm. 7. User scans QR code with banking app and completes payment. 8. User returns to app and taps "Payment Completed" button. 9. System shows processing state: "Processing..." 10. System creates payment history in database. 11. System activates user subscription. 12. System shows success alert: "Payment successful!" 13. User taps "Great!" button, system navigates to Home Screen.
Alternative Sequences/Flows	<ol style="list-style-type: none"> A1. If QR generation fails → System shows error and "Retry" button. A2. If payment confirmation fails → System shows error alert. User can retry or log in again. A3. If user taps back button → System navigates back to Premium Plans Screen. A4. If user session invalid → System shows error and navigates to Login Screen.

Screen Design: View Subscription Plans

Figure: View Subscription Plans Mockup



View Subscription Plans Screen

This screen allows the user to:

- View available premium subscription plans and pricing
- Compare features between free and premium tiers
- Select a subscription plan to purchase
- See current subscription status if already subscribed

Field Description

Field Name	Description
Header Section	Contains back button and screen title "Premium Plans" or "Upgrade"
Current Plan Badge	Shows user's current subscription status if applicable
Plan Cards	Cards displaying different subscription tiers (Basic/Free, Premium, VIP) with pricing
Plan Features List	Bullet points showing features included in each plan
Price Display	Shows subscription cost per month/year with any discounts
Popular/Recommended Badge	Highlights the most popular or recommended plan
Select Plan Button	Primary action button on each card to choose and purchase plan
Terms and Conditions Link	Link to subscription terms and auto-renewal information

3.6.2. Make Payment

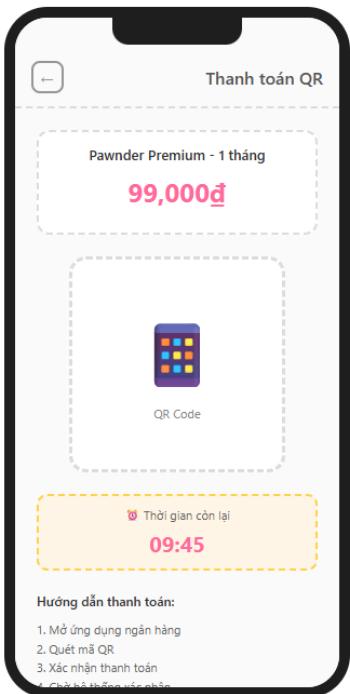
UC-029: Make Payment

Use Case ID	UC029	Use Case Name	Make Payment
Primary Actors	User (Pet Owner)	Secondary Actors	Payment Gateway (VietQR), Banking System
Description	Displays QR code for VietQR payment. Users scan the QR code using their banking app to complete payment for premium subscription.		
Preconditions	1. User is logged into the application. 2. User has selected a premium plan from Premium Plans Screen.		
Postconditions	1. QR code is generated and displayed.		

	<p>2. User completes payment via banking app.</p> <p>3. Payment history is created.</p> <p>4. User is navigated to Home Screen.</p>
Normal Sequence/Flow	<p>1. User arrives from Premium Plans Screen with selected plan details.</p> <p>2. System displays Payment Info Card showing service package, duration, and amount.</p> <p>3. System shows loading indicator while generating QR code.</p> <p>4. System generates and displays QR code (280x280px).</p> <p>5. System shows 3-step instructions for payment.</p> <p>6. User scans QR code with banking app and completes payment.</p> <p>7. User returns to app and taps "Payment Completed" button.</p> <p>8. System processes payment and creates payment history.</p> <p>9. System shows success alert: "Payment successful!"</p> <p>10. User taps "Great!" button.</p> <p>11. System navigates to Home Screen.</p>
Alternative Sequences/Flows	<p>A1. If QR generation fails → System shows error message with retry button. User taps "Retry" to regenerate QR code.</p> <p>A2. If payment confirmation fails → System shows error alert. User can retry or log in again.</p> <p>A3. If user taps back button → System navigates back to Premium Plans Screen without processing payment.</p>

Screen Design: Make Payment Screen

Figure: Make Payment Screen Mockup



Payment Screen

This screen allows the user to:

- Complete payment for premium subscription
- Select payment method (credit card, e-wallet, QR code)
- Review order summary and pricing
- Confirm and process payment transaction

Field Description

Field Name	Description
Header Section	Contains back button and screen title "Payment"
Order Summary Card	Shows selected plan, duration, and total price
QR Code Display	Shows scannable QR code with payment timer
Payment Instructions	Brief text explaining how to complete payment
Confirm Payment Button	Primary action button to complete transaction

3.6.3. View Payment History

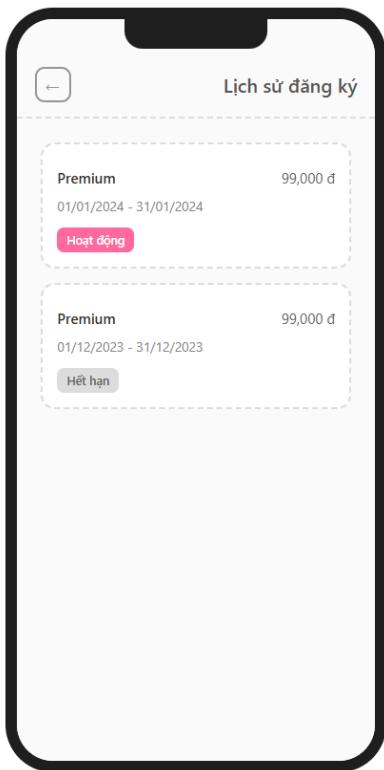
UC-030: View Payment History

Use Case ID	UC030	Use Case Name	View Payment History
Primary Actors	User	Secondary Actors	System
Description	Displays a list of all payment transactions made by the user for premium subscriptions. Shows transaction details including service name, date range, amount, and status.		
Preconditions	1. User is logged into the application. 2. User has accessed Payment History from Settings Screen.		
Postconditions	1. Payment history list is displayed. 2. Summary card shows total spending and transaction count.		
Normal Sequence/Flow	1. User taps "Payment History" option from Settings Screen. 2. System displays loading state while fetching data. 3. System loads payment history for current user. 4. System calculates summary statistics: total spending and transaction count. 5. System displays Summary Card showing total amount and transaction count. 6. System displays payment list with transaction details. 7. User can scroll through the list.		

Alternative Sequences/Flows	<p>A1. If no transactions found → System displays empty state with "Upgrade to Premium" button. User taps to navigate to Premium Plans Screen.</p> <p>A2. If load fails → System displays error message with retry button. User taps "Retry" to reload data.</p> <p>A3. If user pulls down to refresh → System reloads payment history and updates display.</p> <p>A4. If user taps back button → System navigates back to Settings Screen.</p>
-----------------------------	---

Screen Design: Payment History Screen

Figure: Payment History Screen Mockup



Payment History Screen

This screen allows the user to:

- View all past payment transactions and subscription purchases
- Check total spending and transaction count
- Review payment status and details for each transaction

Field Description

Field Name	Description
Header Section	White header with back button and "Payment History" title

Summary Card	Pink gradient card displaying total spent "440,000đ" and total transaction count "2 Giao dịch"
Transaction List	Scrollable list of payment records showing subscription status (Premium Active/Expired), purchase date, renewal date, payment method with QR icon, amount in bold, and green "Thành công" (Success) status label
Transaction Items	Each item includes checkmark icon, subscription name, date range, payment method indicator, and transaction amount with status

3.7. User Profile

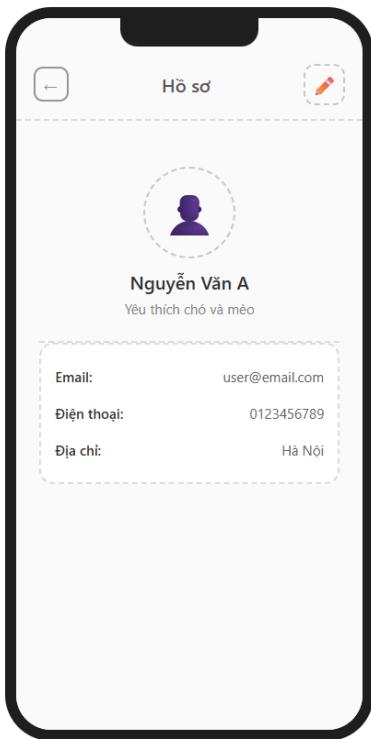
3.7.1. User Profile Screen

UC-048: View User Profile

Use Case ID	UC048	Use Case Name	View User Profile
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to view their own profile including personal info, pet list, and active pet details.		
Preconditions	1. User is logged into the system.		
Postconditions	1. User can view their profile information and pet list.		
Normal Sequence/Flow	1. User taps "Profile" tab on bottom navigation bar. 2. System fetches user data, pets list, and VIP status. 3. System identifies active pet and loads photos/characteristics. 4. System displays Profile Screen with: photo carousel, pet info (name, gender, age, breed), bio, and top 6 characteristics. 5. System renders "My Pets" list (max 3) and Owner info card. 6. System displays VIP badge if applicable. 7. User navigates through pet photos or views detailed info.		
Alternative Sequences/Flows	A1. If > 6 characteristics → System shows first 6 and "Show more" button to expand. A2. If user taps settings icon → System navigates to Settings Screen. A3. If user taps edit button → System navigates to Edit Pet Screen. A4. If user taps "Add Pet" → System checks pet count; if < 3 navigates to Add Pet Screen, if = 3 shows "Maximum pets reached" alert. A5. If user taps pet card → System navigates to Pet Profile Detail Screen. A6. If load fails → System displays error alert.		

Screen Design: User Profile Screen

Figure: User Profile Screen Mockup



Pet Profile Detail Screen

This screen allows the user to:

- View their own profile information and settings
- See account details and verification status
- Access profile editing options
- View subscription status and account statistics

Field Description

Field Name	Description
Header Section	Contains back button, screen title, and edit profile button
Profile Photo	Displays user's profile picture with option to change
User Information	Shows name, email, phone number, and verification badges

3.7.2. Edit Profile Screen

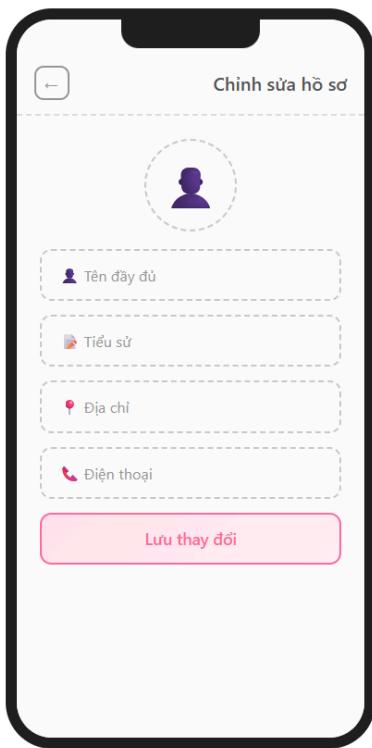
UC-049: Edit User Profile

Use Case ID	UC049	Use Case Name	Edit User Profile
Primary Actors	User (Pet Owner)	Secondary Actors	GPS Service, Location Service

Description	Allows users to edit their profile information including name, gender, and location.
Preconditions	1. User is logged into the application. 2. User has accessed Edit Profile from User Profile Screen.
Postconditions	1. User information is updated in database. 2. Success message is shown. 3. User is navigated back to Profile Screen.
Normal Sequence/Flow	1. User taps "Edit" button in Owner Info section on Profile Screen. 2. System loads current user data. 3. System displays Edit Profile Screen with full name (editable), email (read-only), gender selection, and location section with GPS button. 4. User edits full name and selects gender. 5. User taps "Save Changes" button. 6. System validates name (not empty, min 2 characters). 7. System sends update request to server. 8. System displays success message and navigates back to Profile Screen.
Alternative Sequences/Flows	A1. Update location via GPS: User taps "Get GPS Location". System requests permission, retrieves coordinates, updates address, and displays new location (Ward, District, City). A2. If validation fails: System displays error message for invalid field. User remains on Edit Profile Screen. A3. If update request fails: System displays error alert. User can retry saving. A4. If user taps back button: System discards unsaved changes and navigates back to Profile Screen.

Screen Design: Edit Profile Screen

Figure: Edit Profile Screen Mockup



Edit Profile Screen

This screen allows the user to:

- Update personal information including name, email, and gender
- Enable and update GPS location for matching
- Save profile changes

Field Description

Field Name	Description
Header Section	White header with back button and "Edit Profile" title
Personal Information Section	Form fields for "Full Name" showing "Nguyen Sang", "Email" with placeholder, and "Gender" selection with three options (Male selected in pink, Female, Other)
Location GPS Section	Checkbox with informational text about GPS usage, pink "Lấy vị trí GPS" button to enable location, and current location display showing address "Long Thanh My, District 9, Ho Chi Minh City" with map pin icon
Save Button	Pink gradient button labeled "Save Changes" at bottom to submit all profile updates

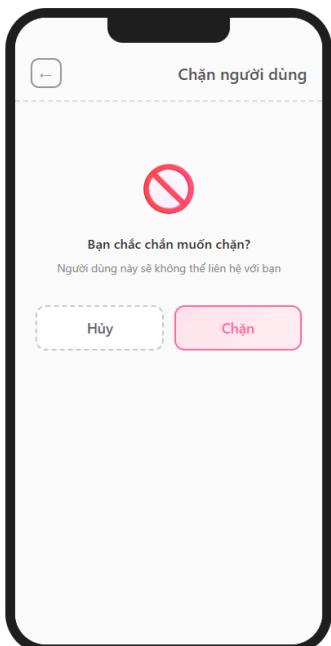
3.7.2. Block User

UC-036: Block User

Use Case ID	UC036	Use Case Name	Block User
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to block another user from the chat conversation. Blocked users cannot send messages or view the blocker's profile.		
Preconditions	1. User is logged into the system. 2. User is viewing a chat conversation with another user.		
Postconditions	1. Target user is added to blocked list. 2. Match is removed. 3. User is redirected to Chat List.		
Normal Sequence/Flow	1. User opens chat conversation with another user. 2. User taps menu icon (three dots). 3. System displays menu options. 4. User taps "Block" option. 5. System displays confirmation dialog with user name. 6. User taps "Confirm" to proceed. 7. System sends block request to server. 8. System displays success message. 9. System navigates back to Chat List Screen.		
Alternative Sequences/Flows	A1. If user cancels block → User taps "Cancel" button. System closes confirmation dialog. A2. If block request fails → System displays error alert. User remains in chat conversation.		

Screen Design: Blocked Users Screen

Figure: Blocked Users Screen Mockup



Blocked Users Screen

This screen allows the user to:

- Block a user to prevent them from contacting or viewing profile
- Confirm block action with warning message
- Cancel the block action if they change their mind

Field Description

Field Name	Description
Header Section	Contains back button and screen title "Chặn người dùng"
Warning Icon	Large block icon indicating blocking action
Confirmation Message	Text asking "Bạn chắc chắn muốn chặn?"
Warning Text	Explains that blocked user cannot contact them
Cancel Button	Secondary button to abort blocking action
Block Button	Primary action button to confirm blocking the user

3.7.2. View Blocked User List

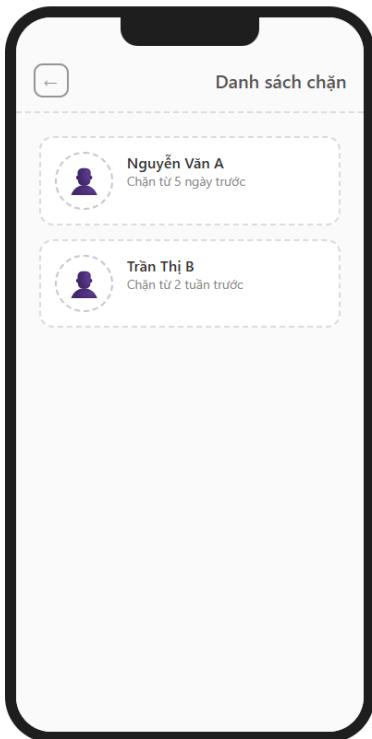
UC-037: View Blocked User List

Use Case ID	UC037	Use Case Name	View Blocked User List
Primary Actors	User	Secondary Actors	System
Description	Allows users to view the list of users they have blocked with option to unblock.		
Preconditions	1. User is logged into the system.		
Postconditions	1. User can view all blocked users with option to unblock.		
Normal Sequence/Flow	1. User navigates to Settings > Blocked Users. 2. System fetches blocked users list from server. 3. System displays list of blocked users with user name and blocked time (relative format). 4. Each user card shows "Unblock" button.		
Alternative Sequences/Flows	A1. If no blocked users → System displays empty state: "No blocked users." A2. If user taps "Unblock" button → See UC-038 Unblock User. A3. If load fails → System displays error message with retry button.		

	A4. If user pulls down to refresh → System reloads blocked users list.
--	--

Screen Design: View Blocked User List Screen

Figure:View Blocked User List Mockup



View Blocked User List

This screen allows the user to:

- View list of all blocked users
- See when each user was blocked
- Access options to unblock users

Field Description

Field Name	Description
Header Section	Contains back button and screen title "Danh sách chặn"
Blocked User List	Scrollable list of blocked users with their information
User Card	Each item shows user avatar, name, and blocked date
User Avatar	Profile picture placeholder for blocked user
User Name	Display name of the blocked user
Block Date	Shows how long ago the user was blocked (e.g., "Chặn từ 5 ngày trước")

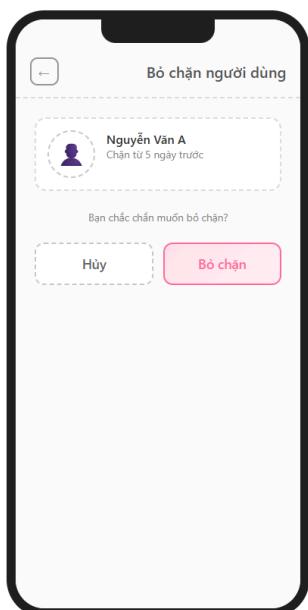
3.7.2. Unblock User

UC-038: Unblock User

Use Case ID	UC038	Use Case Name	Unblock User
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to unblock a previously blocked user from the Blocked Users list in Settings.		
Preconditions	1. User is logged into the system. 2. User has at least one blocked user.		
Postconditions	1. Target user is removed from blocked list. 2. Target user can now interact with the user again.		
Normal Sequence/Flow	1. User views Blocked Users list in Settings. 2. User taps "Unblock" button on a blocked user card. 3. System displays confirmation dialog with user name. 4. User taps "Unblock" to confirm. 5. System sends unblock request to server. 6. System removes user from blocked list. 7. System displays success message.		
Alternative Sequences/Flows	A1. If user cancels unblock → User taps "Cancel" button. System closes confirmation dialog. A2. If unblock request fails → System displays error alert. User remains in blocked list.		

Screen Design: Unblock User Screen

Figure: Unblock User Screen Mockup



Unblock User

This screen allows the user to:

- Unblock a previously blocked user
- Confirm unblock action
- Cancel the unblock action if they change their mind

Field Description

Field Name	Description
Header Section	Contains back button and screen title "Bỏ chặn người dùng"
User Card	Shows blocked user's avatar, name, and blocked date
Confirmation Message	Text asking "Bạn chắc chắn muốn bỏ chặn?"
Cancel Button	Secondary button to abort unblocking action
Unblock Button	Primary action button to confirm unblocking the user

3.8. Settings & Notification

3.8.1. View Notifications

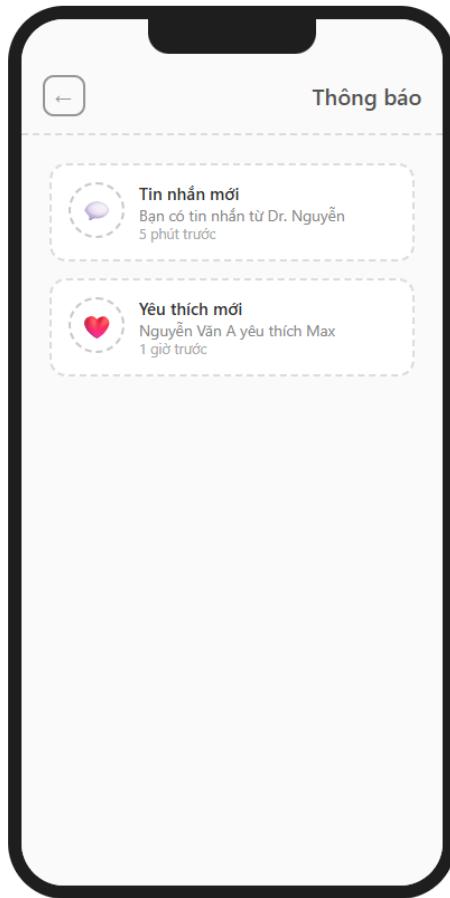
UC-050: View Notifications

Use Case ID	UC050	Use Case Name	View Notifications
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to view all notifications including system alerts, expert replies, and appointment updates.		
Preconditions	1. User is logged into the system.		
Postconditions	1. User can view all notifications with read/unread status.		
Normal Sequence/Flow	1. User taps notification icon or navigates to Notifications Screen. 2. System fetches notifications for user. 3. System displays notification list with filter tabs: All, Unread, Appointment, System, Expert. 4. Each notification shows icon, title, message preview, timestamp, and unread indicator. 5. User taps on a notification. 6. System marks notification as read. 7. System displays notification detail modal or navigates to related screen.		

Alternative Sequences/Flows	A1. Filter: User taps filter tab; System filters list by selected type. A2. Mark all as read: User taps checkmark icon; System marks all as read and updates count to 0. A3. Refresh: User pulls down to refresh; System reloads data from server. A4. Navigation: If event/expert notification tapped, System navigates to Event List or Expert Chat Screen. A5. Empty state: If no notifications match filter, System displays empty state message.
-----------------------------	---

Screen Design: View Notifications

Figure: View Notifications Mockup



View Notifications Screen

This screen allows the user to:

- View all notifications from the system
- See different types of notifications (expert responses, likes, matches, appointments, system updates)
- Mark notifications as read
- Access related content by tapping notifications

Field Description

Field Name	Description
Header Section	Contains back button, screen title "Thông báo", and "Đánh dấu đã đọc" action
Notification List	Scrollable list of notifications ordered by time
Notification Item	Each item shows icon, title, description, and timestamp
Notification Title	Brief title describing the notification
Notification Description	Detailed message about the notification
Timestamp	Shows when notification was received (e.g., "5 phút trước", "1 giờ trước")

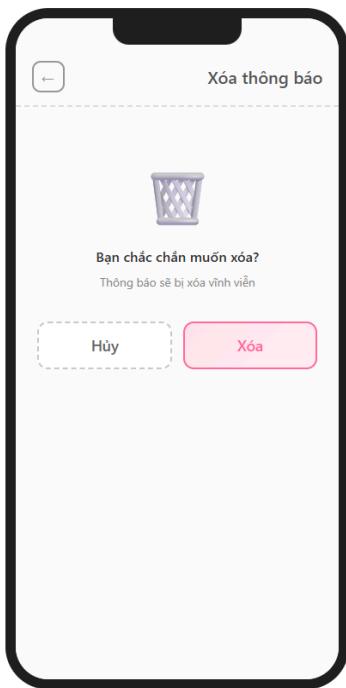
3.8.2. Delete Notification

UC-051: Delete Notification

Use Case ID	UC051	Use Case Name	Delete Notification
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to delete a notification from their notification list.		
Preconditions	1. User is logged into the system. 2. User has at least one notification.		
Postconditions	1. Notification is permanently deleted from the system.		
Normal Sequence/Flow	1. User views Notifications Screen. 2. User long-presses on a notification item. 3. System triggers delete confirmation. 4. System sends delete request to server. 5. System removes notification from list. 6. System displays success message.		
Alternative Sequences/Flows	A1. Delete from modal: User opens notification detail modal, taps "Delete" button. System sends request, closes modal, and removes item from list. A2. If delete request fails: System displays error alert. Notification remains in list.		

Screen Design:Delete Notification Screen

Figure:Delete Notification Screen Mockup



Delete Notification Screen

This screen allows the user to:

- Delete a notification from the list
- Confirm deletion with warning about permanent removal
- Cancel deletion if they change their mind

Field Description

Field Name	Description
Header Section	Contains back button and screen title "Xóa thông báo"
Delete Icon	Large trash icon indicating deletion action
Confirmation Message	Text asking "Bạn chắc chắn muốn xóa?"

3.8.4. My Reports Screen

UC-026: View My Reports

Use Case ID	UC026	Use Case Name	View My Reports
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Displays a list of reports submitted by the user. Shows report details including reported user, reason, message content, status, and resolution.		

Preconditions	1. User is logged into the application. 2. User has accessed My Reports from Settings Screen.
Postconditions	1. Reports are displayed with statistics. 2. User can filter by status (All/Resolved/Pending). 3. User can view report details and resolution.
Normal Sequence/Flow	1. User taps "My Reports" from Settings Screen. 2. System displays loading state while fetching data. 3. System loads reports and calculates statistics: Total, Resolved, and Pending count. 4. System displays My Reports Screen with stats cards, filter info, and report list. 5. User can filter by status, scroll through list, and pull down to refresh.
Alternative Sequences/Flows	A1. Filter by status: User taps "Total", "Resolved", or "Pending" stat cards; System filters list accordingly. A2. Resolution display: If resolved, report card shows user question and resolution in green box. A3. Data unavailable: If reported user/message is deleted, System shows warning: "Information unavailable". A4. Refresh: User pulls down to refresh; System reloads data from server. A5. Empty state: If no reports exist, System displays message: "No reports yet."

Screen Design: My Reports Screen

Figure: My Reports Screen Mockup



Report Management Screen

This screen allows the user to:

- View list of all submitted reports
- See report status (pending, resolved, rejected)
- Check details of each report including reported user and reason
- Track report progress and outcomes

Field Description

Field Name	Description
Header Section	Contains back button and screen title "Báo cáo của tôi"
Report List	Scrollable list of submitted reports ordered by date
Report Card	Each card shows report type, status badge, reported user, reason, and timestamp
Status Badge	Color-coded indicator showing report status (Pending/yellow, Resolved/green, Rejected/red)
Reported User Info	Shows name of the reported user
Description Preview	Shows brief excerpt of report details if provided
Timestamp	Shows when report was submitted

3.8.4. Submit Report

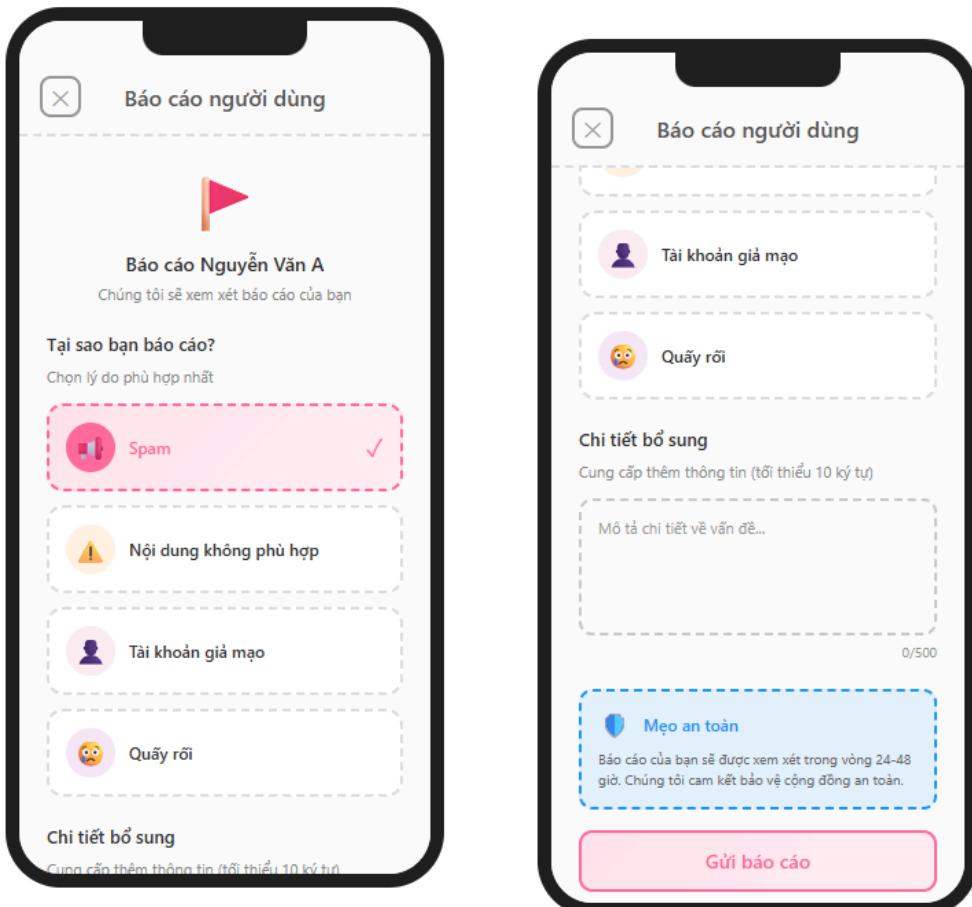
UC-027: Submit Report

Use Case ID	UC027	Use Case Name	Submit Report
Primary Actors	User (Pet Owner)	Secondary Actors	System
Description	Allows users to report inappropriate messages or users. Users select a reason from predefined options and submit the report for admin review.		
Preconditions	1. User is logged into the system. 2. User is viewing a chat conversation. 3. User has selected a message to report.		
Postconditions	1. Report is submitted to system. 2. Report is added to user's report list. 3. Admin receives notification for review.		

Normal Sequence/Flow	<ol style="list-style-type: none"> User long-presses on a message in chat. System displays message menu. User taps "Report" option. System displays Report Modal with list of report reasons. User selects a reason from predefined options (Inappropriate Content, Spam, etc.). User taps "Next" button. System displays confirmation screen with selected reason. User taps "Submit Report" button. System sends report to server with IDs and reason. System displays success message and returns to chat.
Alternative Sequences/Flows	<p>A1. If "Other" reason: User enters custom reason in text field. System continues from step 6.</p> <p>A2. If user taps back button: System returns to reason selection screen.</p> <p>A3. If user closes modal: System resets selection and closes modal.</p> <p>A4. If submit fails: System displays error alert. User can retry.</p> <p>A5. If no reason selected: "Next" button remains disabled.</p>

Screen Design: Submit Report Screen

Figure: Submit Report Screen Mockup



Submit Report Screen

This screen allows the user to:

- Report a user for policy violations
- Select reason for reporting from predefined categories
- Provide additional details about the issue
- Submit report for admin review

Field Description

Field Name	Description
Header Section	Contains close button and screen title "Báo cáo người dùng"
Report Icon & Info	Shows warning icon, reported user name, and explanation text
Reason Selection	List of selectable reason cards (Spam, Inappropriate Content, Fake Account, Harassment, Other)
Selected Reason Indicator	Checkmark showing currently selected reason
Description Field	Text area for additional details (minimum 10 characters, max 500)
Character Counter	Shows current character count (0/500)
Safety Tips Banner	Information box explaining review process and timeline
Submit Button	Primary action button to send report for review

3.9 Appointment & Event Management

3.9.1. View My Appointments

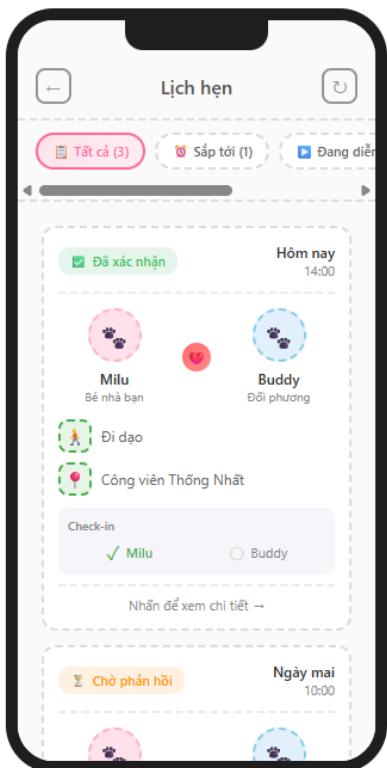
UC-019: View My Appointments

Primary Actors	User	Secondary Actors	None
Description	Allows users to view their appointment list with filtering options.		
Preconditions	User is logged into the system.		
Postconditions	User can view and filter their appointments by status.		
Normal Sequence/Flow	1. User navigates to the My Appointments Screen from Settings or a Match Chat. 2. System fetches the User's appointments from the server. 3. System displays filter tabs with count badges: - All - Upcoming - Ongoing - Completed - Past 4. System displays the appointment list sorted by date (newest first). 5. Each appointment card shows: - Status badge with icon		

	<ul style="list-style-type: none"> - Date and time - Inviter pet name and Invitee pet name - Activity type (Walk / Cafe / Playdate) - Location name - Check-in status (for confirmed or ongoing appointments) <p>6. User can tap on an appointment card to view details.</p>
Alternative Sequences/Flows	<p>A1: Filter by Status At Step 4, User taps a filter tab (Upcoming / Ongoing / Completed / Past): 1. System filters appointments by the selected status. 2. System updates the list display. Use case continues.</p> <p>A2: Pull to Refresh At any step, User pulls down on the list: 1. System shows a refresh indicator. 2. System reloads appointments from the server. 3. System updates the list. Use case continues.</p> <p>A3: No Appointments At Step 4, if no appointments exist for the selected filter: 1. System displays an empty state with a calendar icon. 2. System shows a message based on the selected filter type. Use case ends.</p>

Screen Design: View My Appointments Screen

Figure: View My Appointments Mockup



This screen allows the user to:

- View list of all appointments with filter options
- See appointment status and details
- Check pet match information and activity type
- Monitor check-in status for confirmed appointments
- Access appointment details by tapping cards

Field Description

Field Name	Description
Header Section	Contains back button, screen title "Lịch hẹn", and refresh button
Filter Tabs	Horizontal scrollable tabs (All, Upcoming, Ongoing, Completed, Past) with count badges
Appointment List	Scrollable list of appointment cards
Appointment Card	Shows status badge, date/time, pet match info, activity, location, and check-in status
Status Badge	Color-coded indicator (Confirmed/green, Pending/yellow, Completed/purple, etc.)
Pet Match Section	Displays both pets with avatars, names, roles, and heart icon between them
Activity Icon	Shows activity type icon (walk, cafe, playdate) with label
Location Info	Displays meeting location name
Check-in Status	Shows check-in status for both pets with checkmarks
Card Footer	"Nhấn để xem chi tiết" text with arrow icon
Empty State	Message shown when no appointments match selected filter

3.9.2. Create Appointment

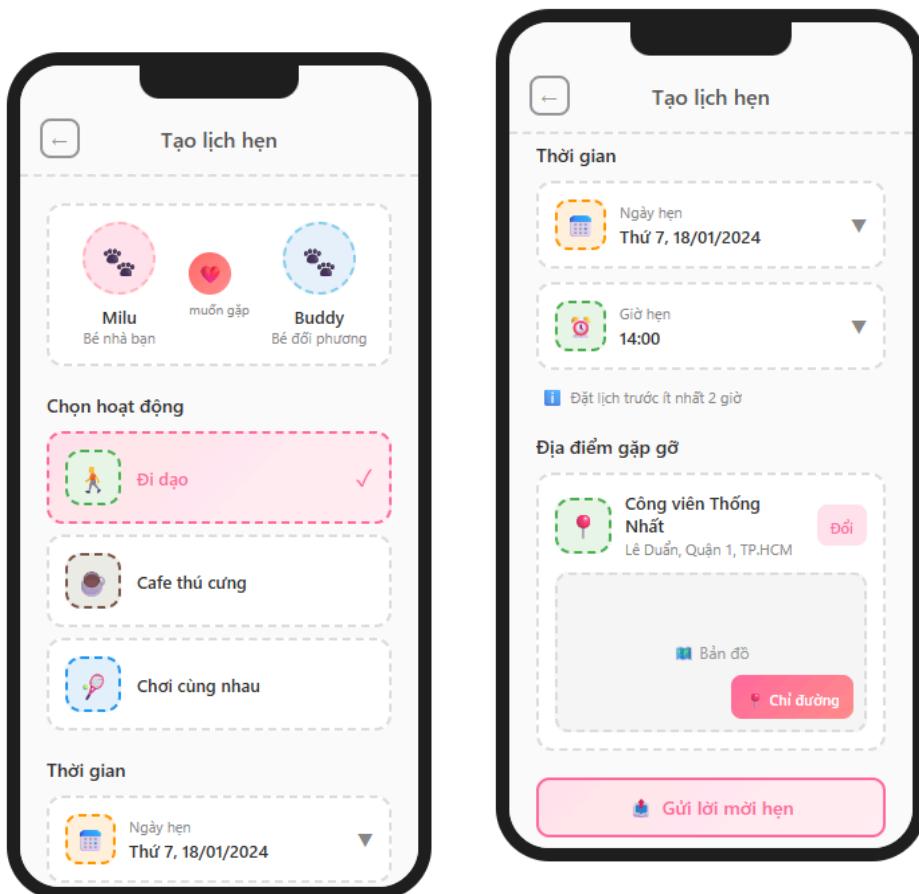
UC-020: Create Appointment

Primary Actors	User	Secondary Actors	None
Description	Allows users to create a new appointment with a matched pet.		
Preconditions	User is logged into the system. User has an active match. No pending appointment exists for this match.		
Postconditions	Appointment is created with "pending" status and invitation sent to the other user.		

Normal Sequence/Flow	<ol style="list-style-type: none"> User taps "Create Appointment" from the Match Chat Screen. System validates preconditions (a match exists and no pending appointment). System displays the Create Appointment Screen with inviter and invitee pet info. User selects an activity type (Walk / Cafe / Playdate). User selects a date and time (minimum 2 hours in advance). User taps "Select Location". System navigates to the Location Picker Screen. User selects a preset or custom location on the map. System returns to the Create Appointment Screen with the selected location. User taps "Send Invitation". System creates the appointment and sends a notification to the invitee. System displays a success message.
Alternative Sequences/Flows	<p>A1: View Appointment Detail</p> <p>At Step 12, User taps "View Detail":</p> <ol style="list-style-type: none"> System navigates to the Appointment Detail Screen. Use case ends.

Screen Design: Create Appointment

Figure: Create Appointment Screen Mockup



This screen allows the user to:

- Create a new appointment invitation with a matched pet
- Select activity type for the meetup
- Choose date and time for the appointment • Pick a meeting location on map
- Send appointment invitation to matched user

Field Description

Field Name	Description
Header Section	Contains back button and screen title "Tạo lịch hẹn"
Pet Match Card	Shows both pets (inviter and invitee) with avatars, names, and heart icon
Activity Selection	List of selectable activity cards (Walk, Cafe, Playdate) with icons
Selected Activity Indicator	Checkmark showing currently selected activity
Date Picker	Input field to select appointment date with calendar icon
Time Picker	Input field to select appointment time with clock icon
Booking Rule Hint	Information text about minimum advance booking requirement (2 hours)
Location Picker	Card showing selected location with name, address, and map preview
Location Info	Displays meeting location name
Map Preview	Small map showing selected location with marker
Change Location Button	Button to modify selected location
Submit Button	Primary action button "Gửi lời mời hẹn" to send invitation

3.9.3. Accept Appointment

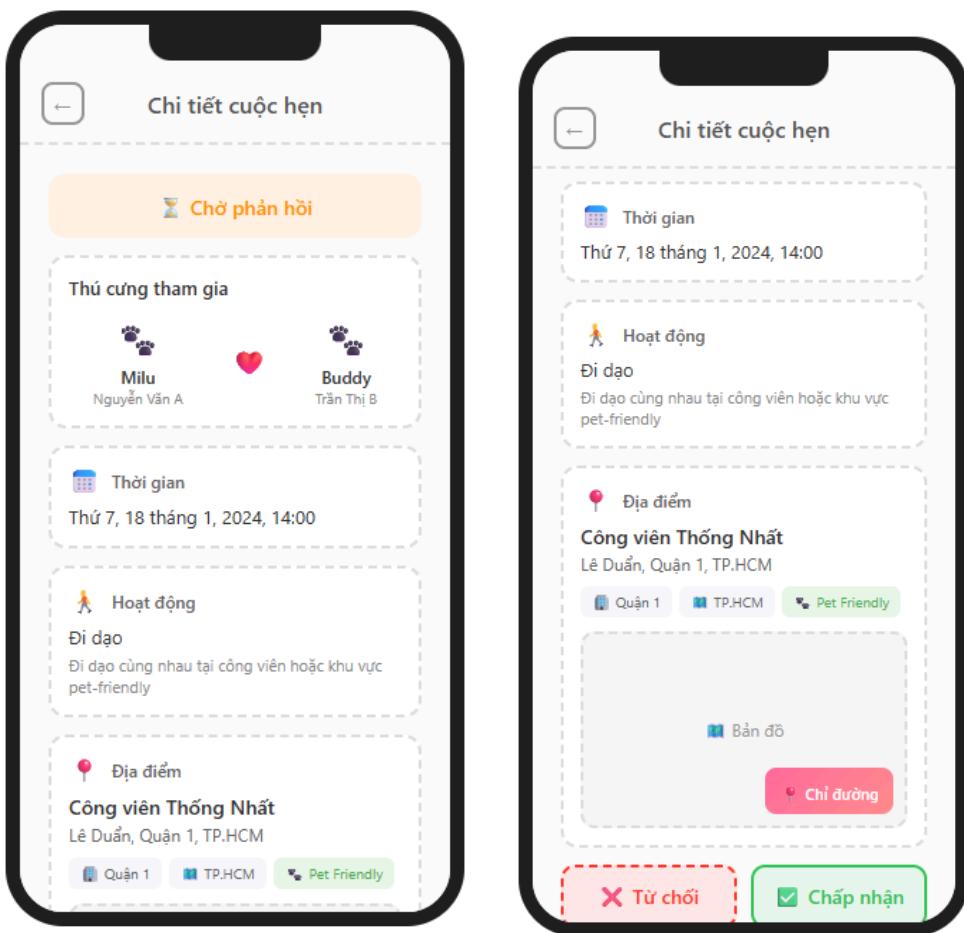
UC-021: Accept Appointment

Primary Actors	User	Secondary Actors	None
Description	Allows users to accept a pending appointment invitation.		
Preconditions	User is logged into the system. User has a pending appointment where they are the current decision maker.		
Postconditions	Appointment status changes to "confirmed".		

Normal Sequence/Flow	<ol style="list-style-type: none"> User opens the Appointment Detail Screen for a pending appointment. System displays appointment details with Accept and Decline buttons. User taps the "Accept" button. System displays a confirmation dialog. User confirms acceptance. System sends an accept request to the server. System updates the appointment status to "confirmed". System sends a notification to the other User. System displays a success message.
Alternative Sequences/Flows	<p>A1: Cancel Confirmation At Step 5, User taps "Cancel":</p> <ol style="list-style-type: none"> System closes the confirmation dialog. Use case ends.

Screen Design: Accept Appointment Screen

Figure: Accept Appointment Screen Mockup



This screen allows the user to:

- View detailed information about appointment invitation
- See pet participants, activity, date/time, and location
- Accept or decline the appointment request

- View location on map before deciding

Field Description

Field Name	Description
Header Section	Contains back button and screen title "Chi tiết cuộc hẹn"
Status Badge	Shows appointment status (Pending/yellow)
Pet Participants Section	Displays both pets with avatars, names, and owner names with heart icon
Date/Time Info	Shows appointment date and time with calendar icon
Activity Info	Displays activity type (Walk, Cafe, Playdate) with icon and description
Location Section	Shows location name, address, and additional tags (district, city, pet-friendly)
Map Preview	Interactive map showing appointment location with marker and navigation button
Action Buttons	Two buttons - Decline (red) and Accept (green) for responding to invitation

3.9.4. Decline Appointment

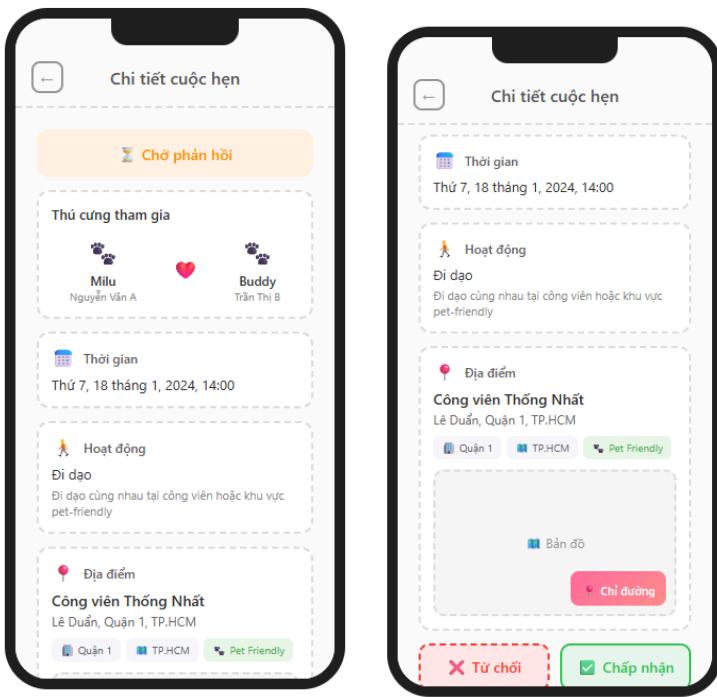
UC-022: Decline Appointment

Primary Actors	User	Secondary Actors	None
Description	Allows users to decline a pending appointment invitation.		
Preconditions	User is logged into the system. User has a pending appointment where they are the current decision maker.		
Postconditions	Appointment status changes to "rejected".		
Normal Sequence/Flow	1. User opens the Appointment Detail Screen for a pending appointment. 2. System displays appointment details with Accept and Decline buttons. 3. User taps the "Decline" button. 4. System displays a modal asking for the decline reason. 5. User enters the decline reason. 6. User taps "Confirm". 7. System sends a decline request to the server. 8. System updates the appointment status to "rejected". 9. System sends a notification to the other User. 10. System displays a success message.		

<p>Alternative Sequences/Flows</p>	<p>A1: Cancel Decline</p> <p>At Step 5, User taps "Cancel":</p> <ol style="list-style-type: none"> 1. System closes the modal. <p>Use case ends.</p>
---	--

Screen Design: Decline Appointment Screen

Figure: Decline Appointment Screen Mockup



This screen allows the user to:

- View detailed information about appointment invitation

- See pet participants, activity, date/time, and location
- Accept or decline the appointment request
- View location on map before deciding

Field Description

Field Name	Description
Header Section	Contains back button and screen title "Chi tiết cuộc hẹn"
Status Badge	Shows appointment status (Pending/yellow)
Pet Participants Section	Displays both pets with avatars, names, and owner names with heart icon
Date/Time Info	Shows appointment date and time with calendar icon
Activity Info	Displays activity type (Walk, Cafe, Playdate) with icon and description
Location Section	Shows location name, address, and additional tags (district, city, pet-friendly)
Map Preview	Interactive map showing appointment location with marker and navigation button
Action Buttons	Two buttons - Decline (red) and Accept (green) for responding to invitation

3.9.5. Cancel Appointment

UC-023: Cancel Appointment

Primary Actors	User	Secondary Actors	None
Description	Allows users to cancel an appointment they are participating in.		
Preconditions	User is logged into the system. User is a participant in the appointment. Appointment status is not cancelled, completed, no_show, or rejected.		
Postconditions	Appointment status changes to "cancelled".		
Normal Sequence/Flow	1. User opens the Appointment Detail Screen. 2. User taps "Cancel Appointment" button. 3. System displays a modal asking for the cancel reason. 4. User enters the cancel reason. 5. User taps "Cancel Appointment". 6. System sends a cancel request to the server. 7. System updates the appointment status to "cancelled". 8. System sends a notification to the other User. 9. System displays a success message.		

Alternative Sequences/Flows	<p>A1: Appointment Soon At Step 2, if the appointment is within 2 hours: 1. System displays a warning "Appointment is about to start. Are you sure?". 2. User taps "Still Cancel". Use case continues from Step 3.</p> <p>A2: User Cancels At Step 2, User taps "No": 1. System closes the warning dialog. Use case ends.</p>
-----------------------------	--

Screen Design: Cancel Appointment Screen

Figure: Cancel Appointment Screen Mockup



This screen allows the user to:

- Cancel an existing appointment
- Provide reason for cancellation
- Confirm cancellation action

Field Description

Field Name	Description
Modal Overlay	Semi-transparent background dimming the screen behind
Confirmation Dialog	Centered modal with cancellation form
Dialog Title	Text "Lý do hủy cuộc hẹn"
Reason Input Field	Text area for entering cancellation reason
Cancel Button	Secondary button to abort cancellation ("Không")
Confirm Button	Primary destructive button to confirm cancellation ("Hủy cuộc hẹn") with red styling

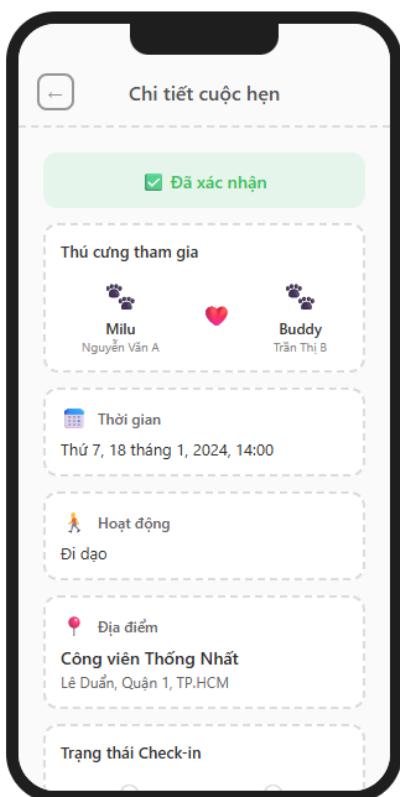
3.9.6. Check-in at Appointment

UC-024: Check-in at Appointment

Primary Actors	User	Secondary Actors	None
Description	Allows users to check-in at the appointment location using GPS.		
Preconditions	User is logged into the system. Appointment status is "confirmed" or "on_going". User has not checked in yet. Current time is within check-in window (30 minutes before to 30 minutes after appointment time).		
Postconditions	User is marked as checked in. If both users checked in, status changes to "on_going".		
Normal Sequence/Flow	1. User opens the Appointment Detail Screen for a confirmed appointment. 2. User taps the "Check-in" button. 3. System requests location permission. 4. User grants the location permission. 5. System retrieves the current GPS location. 6. System validates the User is within 500m of the appointment location. 7. System sends a check-in request to the server. 8. System updates the check-in status. 9. System displays a success message.		
Alternative Sequences/Flows	A1: Both Checked In At Step 8, if both Users have checked in: 1. System updates the appointment status to "on_going". 2. System displays a celebration message. Use case ends.		

Screen Design:Check-in at Appointment Screen

Figure: Check-in at Appointment Screen Mockup



This screen allows the user to:

- Check-in at appointment location using GPS
- View appointment details before check-in
- See check-in status of both participants
- Confirm presence at meeting location

Field Description

Field Name	Description
Header Section	Contains back button and screen title "Chi tiết cuộc hẹn"
Status Badge	Shows appointment status (Confirmed/green)
Pet Participants Section	Displays both pets with avatars, names, and owner names
Date/Time Info	Shows appointment date and time
Activity Info	Displays activity type with icon
Location Info	Shows meeting location name and address
Check-in Status Section	Displays check-in status for both pets with checkmark indicators

Check-in Button	Primary action button with location icon to perform GPS check-in
Location Requirement Info	Text explaining 100m radius requirement for check-in

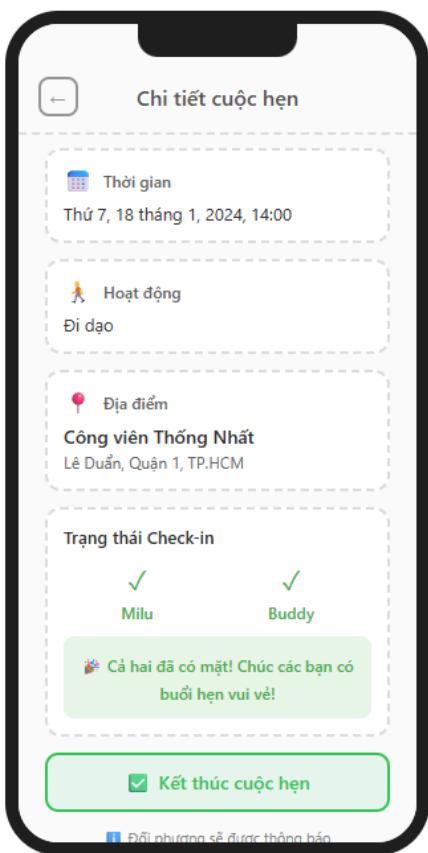
3.9.7. Complete Appointment

UC-025: Complete Appointment

Primary Actors	User	Secondary Actors	None
Description	Allows users to manually complete an ongoing appointment.		
Preconditions	User is logged into the system. User is a participant in the appointment. Appointment status is "on-going" (both users checked in). Current time is after appointment time.		
Postconditions	Appointment status changes to "completed".		
Normal Sequence/Flow	1. User opens the Appointment Detail Screen for an ongoing appointment. 2. User taps the "Complete Appointment" button. 3. System displays a confirmation dialog. 4. User confirms completion. 5. System sends a complete request to the server. 6. System updates the appointment status to "completed". 7. System sends a notification to the other User. 8. System displays a success message.		
Alternative Sequences/Flows	A1: Cancel Completion At Step 4, User taps "Cancel": 1. System closes the confirmation dialog. Use case end		

Screen Design:Complete Appointment Screen

Figure: Complete Appointment Screen Mockup



This screen allows the user to:

- Mark appointment as completed after meetup
- View appointment details and check-in status
- Confirm completion when both pets have checked in
- End the ongoing appointment

Field Description

Field Name	Description
Header Section	Contains back button and screen title "Chi tiết cuộc hẹn"
Status Badge	Shows appointment status (On-going/blue)
Pet Participants Section	Displays both pets with avatars, names, and owner names
Date/Time Info	Shows appointment date and time
Activity Info	Displays activity type with icon
Location Info	Shows meeting location name and address

Check-in Status Section	Displays both pets with green checkmarks indicating successful check-in
Success Banner	Green banner with celebration message "Cả hai đã có mặt! Chúc các bạn có buổi hẹn vui vẻ!"
Complete Button	Primary action button "Kết thúc cuộc hẹn" with green styling
Notification Info	Text explaining that other party will be notified

3.9.8. View Event List

UC-044: View Event List

Primary Actors	User	Secondary Actors	None
Description	Allows users to view list of pet contest events with filtering options.		
Preconditions	User is logged into the system		
Postconditions	User can view and filter events by status.		
Normal Sequence/Flow	1. User navigates to the Event List Screen from Settings. 2. System fetches active events from the server. 3. System displays filter tabs with count badges: All, Active, Upcoming, Completed. 4. System displays the event list with cards showing: cover image, title, status badge, submission count, vote count, and time remaining. 5. User can tap on an event card to view details.		
Alternative Sequences/Flows	A1: Filter by Status At Step 3, User taps a filter tab: 1. System filters events by the selected status. 2. System updates the list display. Use case continues. A2: Pull to Refresh At any step, User pulls down on the list: 1. System shows a refresh indicator. 2. System reloads events from the server. Use case continues. A3: No Events At Step 3, if no events exist for the selected filter: 1. System displays an empty state with a calendar icon. 2. System shows an appropriate message. Use case ends.		

Screen Design:View Event List Screen

Figure: View Event List Screen Mockup



This screen allows the user to:

- View list of all pet-related events
- Filter events by status (All, Ongoing, Upcoming, Completed)
- See event details including participants and votes
- Access individual event details

Field Description

Field Name	Description
Header Section	Contains back button and screen title "Sự kiện"
Filter Tabs	Horizontal scrollable tabs (All, Ongoing, Upcoming, Completed)
Event List	Scrollable list of event cards

Event Card	Shows event icon, name, date range, status badge, description, and statistics
Event Icon	Large emoji representing event type
Event Name	Title of the event
Event Date	Date range showing start and end dates
Status Badge	Color-coded indicator (Ongoing/green, Upcoming/yellow, Completed/gray)
Event Description	Brief description of the event
Statistics Row	Shows participant count and vote count

3.9.9. View Event Detail

UC-045: View Event Detail

Primary Actors	User	Secondary Actors	None
Description	Allows users to view detailed information of a pet contest event.		
Preconditions	User is logged into the system. Event exists.		
Postconditions	User can view event details, submissions, and leaderboard.		
Normal Sequence/Flow	1. User taps on an event card from the Event List. 2. System fetches event details from the server. 3. System displays the Event Detail Screen with: cover image (tappable for full screen), title and status badge, description, prize information, timeline (start time, submission deadline, end time), countdown timer, statistics (submission count, vote count). 4. System displays two tabs: Submissions and Leaderboard. 5. System shows the submissions grid with pet photos. 6. User can tap on a submission to view it in a modal.		
Alternative Sequences/Flows	A1: View Leaderboard At Step 4, User taps the "Leaderboard" tab: 1. System fetches leaderboard data. 2. System displays a ranked list of submissions by vote count. Use case continues. A2: View Winners At Step 3, if the event is completed and has winners: 1. System displays a Winners section with the top 3 submissions. Use case continues. A3: View Full Cover Image At Step 3, User taps the cover image: 1. System displays a full-screen image modal. 2. User taps close to dismiss. Use case continues.		

Screen Design: View Event Detail Screen

Figure: View Event Detail Screen Mockup



This screen allows the user to:

- View detailed information about a specific event
- See event date, location, and description
- Register to participate in the event
- Access full event information before joining

Field Description

Field Name	Description
Header Section	Contains back button and screen title "Chi tiết sự kiện"
Event Icon	Large emoji representing the event type
Event Name	Title of the event
Event Date	Shows event date
Event Location	Displays venue or location name
Event Description	Detailed description of the event

Join Button	Primary action button "Tham gia sự kiện" to register for event
-------------	--

3.9.10. Submit Event Entry

UC-046: Submit Event Entry

Primary Actors	User	Secondary Actors	None
Description	Allows users to submit their pet's photo/video to participate in an event.		
Preconditions	User is logged into the system. Event status is "active" (submission period open). User has not already submitted to this event. User has at least one pet profile.		
Postconditions	User's submission is created and visible in the event.		
Normal Sequence/Flow	1. User taps the "Join Contest" button on the Event Detail Screen. 2. System displays the Submit Entry Screen. 3. System loads the User's pet list. 4. User selects a pet from the dropdown. 5. User taps the media selector to choose a photo or video. 6. System opens the device gallery. 7. User selects an image or video. 8. System displays a media preview. 9. User optionally enters a caption (maximum 500 characters). 10. User taps "Submit Entry". 11. System uploads the media to cloud storage. 12. System creates the submission record. 13. System displays a success message. 14. System navigates back to the Event Detail Screen.		
Alternative Sequences/Flows	A1: Single Pet At Step 3, if the User has only one pet: 1. System auto-selects that pet. Use case continues from Step 5. A2: Change Media At Step 8, User taps "Change Photo": 1. System opens the gallery again. Use case continues from Step 7.		

Screen Design: Submit Event Entry Screen

Figure: Submit Event Entry Screen Mockup



This screen allows the user to:

- Register pet to participate in an event
- Select which pet to enter
- Provide description for the entry
- Upload photos for the event submission

Field Description

Field Name	Description
Header Section	Contains back button and screen title "Đăng ký tham gia"
Pet Selection Field	Dropdown or input to choose which pet to register
Description Field	Text input for brief description of entry
Photo Upload Field	Input to add photos for the event entry
Submit Button	Primary action button "Gửi đăng ký" to submit event registration

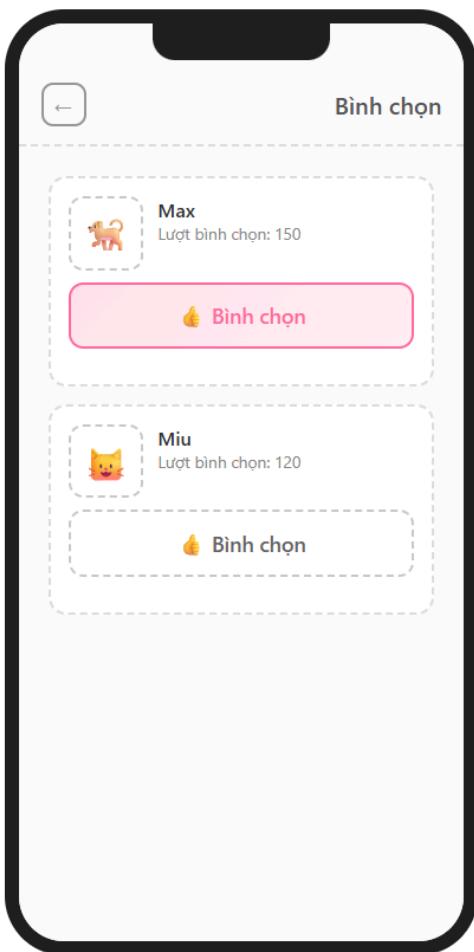
3.9.11. Vote / Unvote for Pet

UC-047: Vote / Unvote for Pet

Primary Actors	User	Secondary Actors	None
Description	Allows users to vote or remove vote for a submission in an event.		
Preconditions	User is logged into the system. Event is in voting period (active or submission_closed status). User is not the owner of the submission.		
Postconditions	Vote is added or removed from the submission.		
Normal Sequence/Flow	1. User views a submission in the Event Detail (grid or modal). 2. User taps the heart (vote) button on the submission. 3. System checks if the User has already voted. 4. If the User has not voted, the System adds the vote (optimistic update). 5. System sends a vote request to the server. 6. System updates the vote count display. 7. The heart icon changes to the filled state.		
Alternative Sequences/Flows	A1: Unvote At Step 3, if the User has already voted: 1. System removes the vote (optimistic update). 2. System sends an unvote request to the server. 3. System decreases the vote count. 4. The heart icon changes to the outline state. Use case ends. A2: Vote from Modal At Step 1, User opens the submission modal: 1. User taps the vote button in the modal. Use case continues from Step 3. A3: Vote from Leaderboard At Step 1, User views the Leaderboard tab: 1. User taps on a leaderboard item. 2. System opens the submission modal. Use case continues from Step 2.		

Screen Design: Vote / Unvote for Pet Screen

Figure: Vote / Unvote for Pet Screen Mockup



This screen allows the user to:

- View list of pets participating in an event
- Vote for favorite pets in the competition
- See current vote counts for each pet
- Unvote if already voted

Field Description

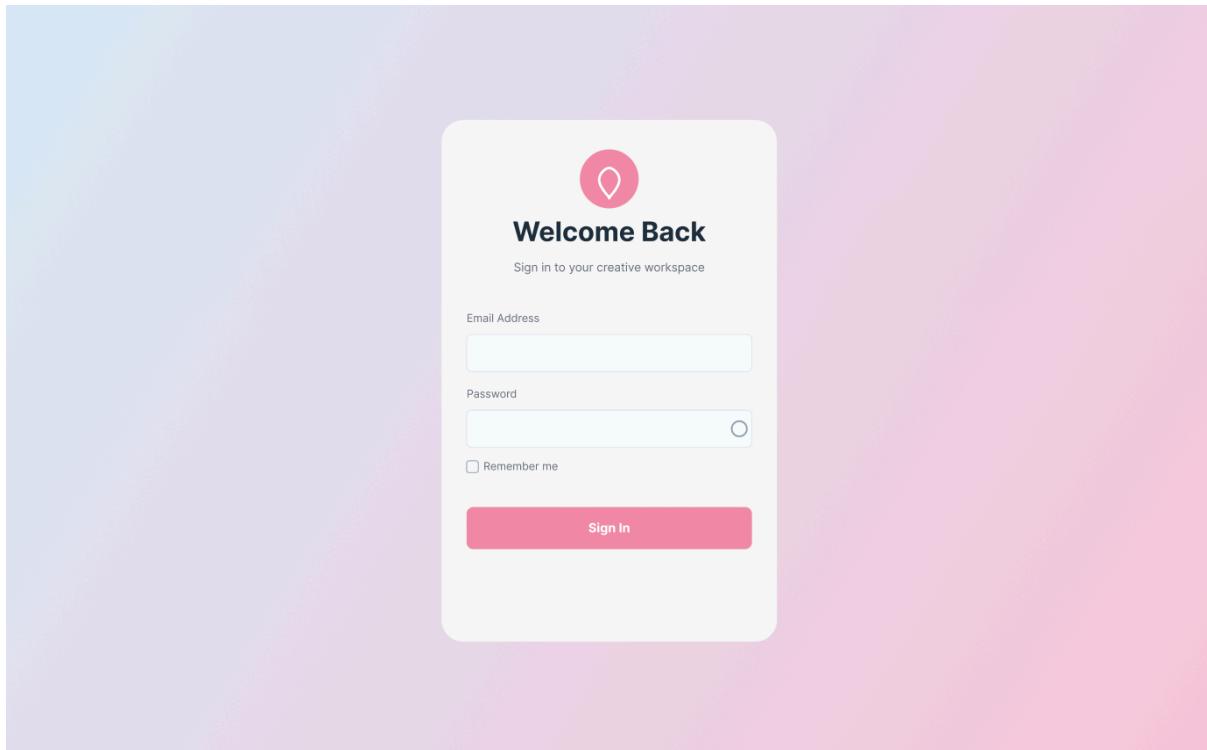
Field Name	Description
Header Section	Contains back button and screen title "Bình chọn"
Pet Entry List	Scrollable list of participating pets
Pet Entry Card	Shows pet icon, name, and current vote count
Pet Icon	Avatar or emoji representing the pet
Pet Name	Name of the participating pet

Vote Count	Displays number of votes received
Vote Button	Primary action button to cast vote (filled) or secondary button if already voted

3.10 Admin/ Expert Login

UC-038: Admin/Expert Login

Primary Actors	Admin ,Expert	Secondary Actors	None
Description	The Admin or Expert logs into the Pawnder Admin System using the email and password provided by the platform. The system validates the credentials and grants access to the appropriate dashboard (Admin Dashboard or Expert Workspace).		
Preconditions	<p>The Admin/Expert account already exists in the system.</p> <p>The email and password are stored and valid in the database.</p> <p>The account is not disabled or locked.</p> <p>The user is currently on the Login Screen.</p>		
Postconditions	<p>The user is successfully authenticated.</p> <p>A login session is created.</p> <p>The system redirects the user to the correct interface:</p> <ul style="list-style-type: none"> ● Admin → Admin Dashboard ● Expert → Expert Workspace 		
Normal Sequence/Flow	<ol style="list-style-type: none"> 1. The user opens the Login page. 2. The system displays the login form. 3. The user enters their email address. 4. The user enters their password. 5. (Optional) The user selects the “Remember me” option. 6. The user clicks the Sign In button. 7. The system validates the entered credentials. 8. If the credentials are correct, the system creates an authentication session. 9. The system redirects the user to the appropriate dashboard (Admin Dashboard or Expert Workspace). 		
Alternative Sequences/Flows	<p>A1 – Incorrect Password</p> <ol style="list-style-type: none"> 1. The user enters an incorrect password. 2. The user clicks “Sign In”. 3. The system displays an error message: “Incorrect email or password.” 4. The user re-enters their credentials and attempts to log in again. <p>A2 – Email Not Found</p> <ol style="list-style-type: none"> 1. The user enters an email address that does not exist in the system. 2. The user clicks “Sign In”. 3. The system displays an error message: “Account does not exist.” <p>A3 – Account Disabled</p> <ol style="list-style-type: none"> 1. The user enters a valid email and password. 2. The user clicks “Sign In”. 3. The system detects that the account is disabled and displays: “Your account has been disabled.” <p>A4 – Forgot Password (if available)</p> <ol style="list-style-type: none"> 1. The user selects “Forgot password?”. 2. The user enters their email address. 3. The system sends a password reset link to the user’s email. 		



Field Description

Field Name	Description
Email	Input field where the admin/expert enters their registered email address to log in.
Password	Password input field with masked characters for security during login.
Remember me	Checkbox that allows the system to keep the admin/expert logged in on the current device.
Sign In	Primary action button. When clicked, the system validates the credentials and grants access if authenticated.

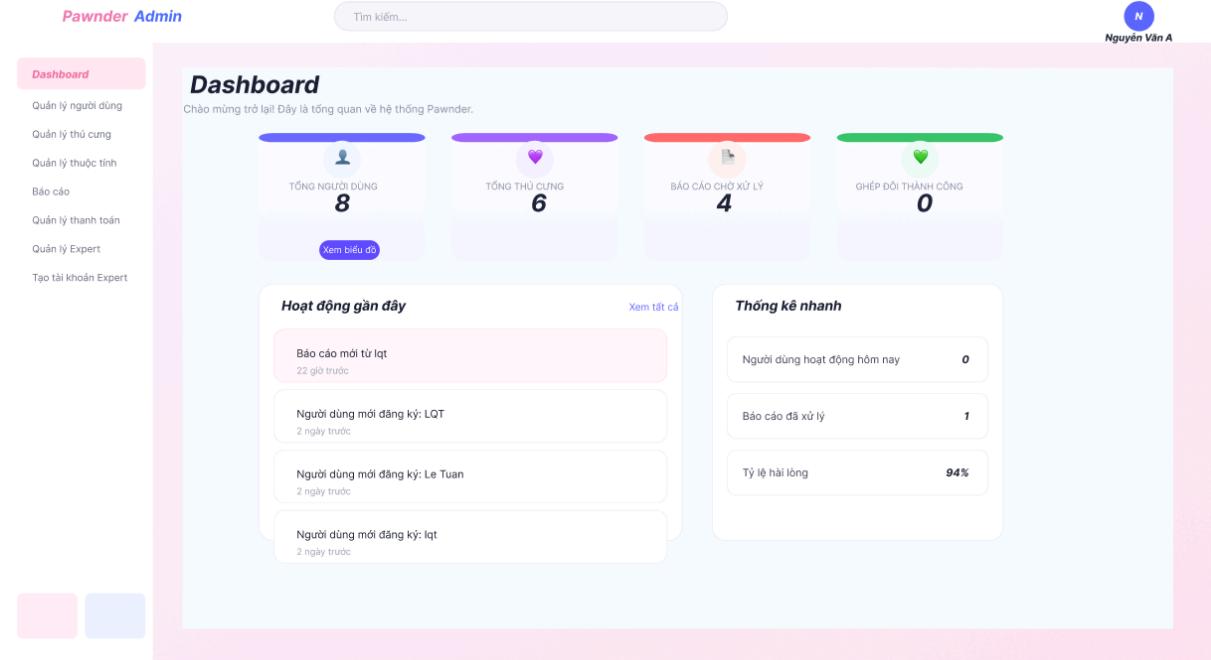
3.11 Admin Services

3.11.1 Admin dashboard

UC-058: Admin dashboard

Primary Actors	Admin	Secondary Actors	None
Description	<p>The Admin Dashboard provides a real-time overview of the Pawnder system. It displays key metrics and activity to help administrators monitor the platform effectively. Main information includes:</p> <ul style="list-style-type: none"> ● Total number of users ● Total number of pets ● Number of pending reports ● Total successful matches ● Recent activities 		

	<ul style="list-style-type: none"> Quick statistics (daily active users, resolved reports, satisfaction rate) <p>The dashboard serves as the central monitoring hub for the administrator.</p>
Preconditions	<p>Admin must be successfully logged in. Stable communication between dashboard service and database. Existing data for users, pets, reports, and activities.</p>
Postconditions	<p>Success case:</p> <ul style="list-style-type: none"> Dashboard loads and displays all metrics and components correctly. Admin can navigate to other modules from the dashboard. <p>Failure case:</p> <ul style="list-style-type: none"> Some or all widgets fail to load → appropriate error messages appear.
Normal Sequence/Flow	<ol style="list-style-type: none"> Admin logs in and is redirected to the Dashboard. System sends queries to fetch: <ul style="list-style-type: none"> total users total pets pending reports successful matches System requests Recent Activities. System retrieves Quick Stats from Payment/Activity Service. Dashboard renders: <ul style="list-style-type: none"> Four KPI cards Recent Activities list Quick Stats panel Admin scrolls, views charts, or opens “See all”. Admin navigates to other management modules (User, Pet, Report, Payment...).
Alternative Sequences/Flows	<p>A1 – No data available</p> <ul style="list-style-type: none"> Database returns empty results. → KPIs display 0, Recent Activities show “No recent activity”. <p>A2 – Data fetch error</p> <ul style="list-style-type: none"> Dashboard fails to connect to the database. → Display: “Failed to load data. Please try again.” <p>A3 – Partial failure</p> <ul style="list-style-type: none"> Some APIs fail while others respond normally. → Faulted widgets display “N/A”. <p>A4 – Timeout</p> <ul style="list-style-type: none"> Data retrieval exceeds the timeout limit. → Loading spinner → “Request timed out. Please reload.”



Field Description

Field Name	Description
Dashboard Title	The page title displayed at the top - "Dashboard"
Welcome Message	Welcome message shown below the title - "Chào mừng trở lại! Đây là tổng quan về hệ thống Pawnder."
Total Users	Card displaying the total number of users in the system, includes a "View Chart" button to show user growth chart
Total Pets	Card displaying the total number of pets registered in the system
Pending Reports	Card displaying the number of reports with "Pending" status waiting for admin review
Successful Matches	Card displaying the total number of successful matches between pets
User Growth Chart	Area Chart displaying cumulative number of users by month (last 12 months), can be toggled on/off
Recent Activities	Section displaying list of recent activities (new user registrations, new reports), shows maximum 2 items with "View All" button
Active Users Today	Quick stat showing number of users active today (based on UpdatedAt or CreatedAt within today)
Processed Reports	Quick stat showing number of reports that have been processed (status = "Resolved" or "Rejected")

3.11.2 User management

UC-60: View User List

Primary Actors	Admin
Secondary Actors	None
Description	This use case allows the Admin to view a list of all users registered in the Pawnder system, including their basic information, account status, verification status, and summary statistics. The Admin can search, filter, and review user records for management purposes.
Preconditions	Admin has successfully logged into the Admin Portal. Admin has permission to access User Management module.
Postconditions	User list is displayed successfully. Admin can proceed to view user details or perform management actions (e.g., ban/unban).
Normal Sequence/Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin selects “Quản lý người dùng” (User Management) from the sidebar menu. 3. System sends a request to retrieve user data from the database. 4. System loads user statistics including total users, normal users, premium users, and verified users. 5. System displays the user list table with user information. 6. Admin can search users by name, email, or username. 7. Admin can filter users by account status.
Alternative Sequences/Flows	<p>A1 – No data available - Database returns no user records. → System displays an empty state message indicating no users found.</p> <p>A2 – Data fetch error - System fails to retrieve user data due to server error. → System displays an error message: “Failed to load user data. Please try again.”</p> <p>A3 – Permission denied - Admin does not have sufficient permission to access User Management. → System blocks access and displays an authorization error message.</p> <p>A4 – Timeout - Data retrieval exceeds the allowed response time. → System displays a timeout message and allows retry.</p>

Pawnder Admin

Dashboard

Quản lý người dùng

Quản lý thú cưng
Quản lý thuộc tính
Báo cáo
Quản lý thanh toán
Quản lý Expert
Tạo tài khoản Expert

Quản lý người dùng

Danh sách tất cả người dùng trong hệ thống

Tìm kiếm theo tên, email, username...

Tất cả trạng thái

TỔNG NGƯỜI DÙNG: 3
NORMAL: 3
PREMIUM: 0
ĐÃ XÁC THỰC: 3

Avatar	Thông tin cá nhân	Liên hệ	Trạng thái	Xác thực	Thống kê	Ngày tạo	Hành động
TT Trần Thị B @user_NG	expert@pawnder.com	NORMAL	✓ Đã xác thực	KHÔNG ÁP DỤNG CHO EXPERT	25/11/2025 17:31:37	View Ban	
LM Lê Minh C @user_Nam	user1@example.com	NORMAL	✓ Đã xác thực	THỦ CUNG: 1	25/11/2025 17:31:37	View Ban	
LM Lê Minh D @user_2	user2@pawnder.com	NORMAL	✓ Đã xác thực	THỦ CUNG: 1	25/11/2025	View Ban	

Field Name	Description
Avatar	Displays the user's profile avatar or initials if no avatar is available.
Full Name	Displays the full name of the user.
Username	Displays the unique username of the user.
Email	Displays the registered email address of the user.
Account Status	Indicates the current status of the user account (e.g., Normal, Premium).
Verification Status	Indicates whether the user account has been verified.
Statistics	Displays summary information related to the user (e.g., number of pets, expert status).
Created Date	Displays the date and time when the user account was created.
Actions	Provides available management actions for the selected user (e.g., view details, ban/unban).

UC-61: View User Detail

Primary Actors	Admin
Secondary Actors	None

Description	This use case allows the Admin to view detailed information of a selected user, including personal information, contact details, account status, verification status, activity statistics, and related data such as pets and matching history.
Preconditions	Admin has successfully logged into the Admin Portal. Admin has permission to access User Management. The selected user exists in the system.
Postconditions	User detail information is displayed successfully. Admin can navigate between detail tabs or return to the user list.
Normal Sequence/Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin navigates to “Quản lý người dùng” (User Management) from the sidebar menu. 3. System displays the list of users. 4. Admin selects the View (eye) icon corresponding to a user record. 5. System sends a request to retrieve detailed information of the selected user. 6. System displays the User Detail screen with overview information. 7. System displays user detail sections including personal information, contact details, account status, verification status, and timestamps. 8. Admin may switch between tabs such as Personal Information, Pets, Matching, and Activities.
Alternative Sequences/Flows	<p>A1 – User not found - Selected user does not exist or has been deleted. → System displays an error message and redirects Admin back to the user list.</p> <p>A2 – Data retrieval error - System fails to load user detail due to server error. → System displays an error message and allows retry.</p> <p>A3 – Permission denied - Admin does not have permission to view user details. → System blocks access and displays an authorization error message.</p> <p>A4 – Timeout - User detail request exceeds response time limit. → System displays a timeout message and provides retry option.</p>

The screenshot shows a user profile for 'Lê Minh C' (LM). At the top, there's a back button labeled 'Quay lại danh sách' and a title 'Chi tiết người dùng'. Below the title, the user's name 'Lê Minh C' is displayed with an icon, followed by 'N/A • Nam' and a green checkmark indicating 'Đã xác thực' (verified). To the right, there are three boxes: '1 THỦ CƯNG' (1 pet), '0 GIHÉP ĐỒI' (0 matches), and '5/1/2026 THAM GIA' (joined on 5/1/2026). Below these are four tabs: 'Thông tin cá nhân' (Personal Information) which is selected, 'Thú cưng' (Pet), 'Ghép đôi' (Match), and 'Hoạt động' (Activity).

Thông tin liên hệ

- Email: user@example.com

Thông tin cá nhân

- Giới tính: Nam
- Vai trò: User

Trạng thái tài khoản

- Trạng thái: NORMAL
- Xác thực: Đã xác thực

Thời gian

- Ngày tạo: 13:46:27 5/1/2026
- Cập nhật cuối: 21:28:36 17/1/2026

Field Name	Description
Avatar	Displays the user's profile avatar or initials if no avatar is available.
Full Name	Displays the full name of the user.
Gender	Displays the gender of the user.
Role	Displays the role of the account (e.g., User).
Email	Displays the registered email address of the user.
Account Status	Displays the current status of the user account (e.g., Normal).
Verification Status	Indicates whether the user account has been verified.
Pets Count	Displays the total number of pets owned by the user.
Matches Count	Displays the total number of successful matches of the user.
Join Date	Displays the date when the user joined the system.
Created Date	Displays the date and time when the user account was created.
Last Updated	Displays the date and time of the last account update.

Tabs Navigation	Allows Admin to switch between user detail sections (Personal Information, Pets, Matching, Activities).
-----------------	---

UC-62: Ban User

Primary Actors	Admin
Secondary Actors	
Description	This use case allows the Admin to temporarily or permanently ban a user who violates platform policies. The Admin specifies the ban duration and provides a reason for the ban. Once confirmed, the system updates the user's account status and restricts access accordingly.
Preconditions	Admin has successfully logged into the Admin Portal. Admin has permission to manage users. The selected user exists and is not currently banned.
Postconditions	User account status is updated to Banned. Ban duration and reason are stored in the system. User access to the system is restricted until the ban expires or is manually lifted.
Normal Sequence/Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin navigates to “Quản lý người dùng” (User Management). 3. System displays the list of users. 4. Admin selects the “Xử lý sai phạm” (Ban) action corresponding to a user. 5. System displays the Ban User modal dialog. 6. Admin selects a ban duration (e.g., 1 day, 3 days, 7 days, 1 month, 3 months, permanent). 7. System automatically calculates and displays the ban expiration time. 8. Admin enters the reason for banning the user. 9. Admin clicks “Xác nhận ban” (Confirm Ban). 10. System validates input data. 11. System updates the user status to Banned and saves ban information. 12. System closes the modal and updates the user list.
Alternative Sequences/Flows	<p>A1 – Missing ban reason - Admin attempts to confirm without entering a ban reason. → System displays a validation message and blocks submission.</p> <p>A2 – User already banned - Selected user is already under an active ban. → System displays a warning message and disables ban action.</p> <p>A3 – Permission denied - Admin does not have permission to ban users. → System blocks the action and displays an authorization error message.</p> <p>A4 – System error / Timeout - System fails to process the ban request or exceeds response time. → System displays an error message and allows retry.</p>

Xử lý sai phạm - Lê Minh C



Ban người dùng

Thời gian ban:

1 ngày



Lý do ban: *

Nhập lý do ban người dùng...

Thời gian ban sẽ hết hạn vào:

01:46:50 19/1/2026

Hủy

Xác nhận ban

Field Name	Description
Ban Duration	Dropdown list allowing Admin to select the ban duration (e.g., 1 day, 3 days, 7 days, 1 month, 3 months, permanent).
Ban Reason	Text area allowing Admin to enter the reason for banning the user (required).
Ban Expiration Time	Displays the calculated date and time when the ban will expire based on the selected duration.

Cancel Button	Allows Admin to cancel the ban action and close the modal without saving changes.
Confirm Ban Button	Confirms the ban action and submits the ban information to the system.

UC-63: Unban User

Primary Actors	Admin
Secondary Actors	None
Description	This use case allows the Admin to remove an active ban from a user account. After confirmation, the system restores the user's account status and allows the user to access the system again.
Preconditions	<ul style="list-style-type: none"> - Admin has successfully logged into the Admin Portal. - Admin has permission to manage users. - The selected user exists and is currently in Banned status.
Postconditions	<ul style="list-style-type: none"> - User account status is updated from Banned to Normal. - Ban record is closed or marked as inactive. - A success notification is displayed to the Admin.
Normal Sequence/Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin navigates to “Quản lý người dùng” (User Management). 3. System displays the list of users with their current statuses. 4. Admin selects the “Gỡ ban” (Unban) action corresponding to a banned user. 5. System displays the Unban User confirmation modal with current ban information. 6. Admin reviews the ban status information. 7. Admin clicks “Xác nhận gỡ ban” (Confirm Unban). 8. System validates the request. 9. System updates the user account status to Normal. 10. System displays a success notification message. 11. System refreshes the user list with the updated status.
Alternative Sequences/Flows	<p>A1 – User not banned - Selected user is not in Banned status. → System disables the unban action or displays a warning message.</p> <p>A2 – Permission denied - Admin does not have permission to unban users. → System blocks the action and displays an authorization error message.</p> <p>A3 – System error / Timeout - System fails to process the unban request or exceeds response time. → System displays an error message and allows retry.</p>

Gỡ ban người dùng - Lê Minh C



Thông tin ban hiện tại:

Lý do ban:

d

Thời gian ban còn lại:

23 giờ 59 phút 54 giây

Hết hạn vào:

01:53:29 19/1/2026

Bị ban từ:

01:53:30 18/1/2026



Bạn có chắc chắn muốn gỡ ban cho người dùng này không?

Hủy

Xác nhận gỡ ban

Field Name	Description
Ban Status	Displays the current ban status of the selected user (e.g., BANNED).
Ban Information	Displays information indicating that the user is currently banned.
Warning Message	Displays a warning message asking Admin to confirm the unban action.
Cancel Button	Allows Admin to cancel the unban action and close the modal.
Confirm Unban Button	Confirms the unban action and submits the request to the system.
Success Notification	Displays a notification message indicating the user has been unbanned successfully.

3.11.3 Attribute management

UC-64: Create Attribute

Primary Actors	Admin
Secondary Actors	None
Description	This use case allows the Admin to create a new attribute that can be used in the system for managing pet information. The Admin defines the attribute name, value type, and optional unit before saving it to the system.
Preconditions	Admin has successfully logged into the Admin Portal. Admin has permission to access Attribute Management.
Postconditions	A new attribute is created and stored in the system. The attribute is displayed in the attribute list and becomes available for option management.
Normal Sequence/Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin navigates to “Quản lý thuộc tính” (Attribute Management) from the sidebar menu. 3. System displays the Attribute Management screen. 4. Admin enters the Attribute Name. 5. Admin selects the Value Type of the attribute. 6. Admin enters the Unit (if applicable). 7. Admin clicks “Tạo thuộc tính” (Create Attribute). 8. System validates the input data. 9. System saves the new attribute to the database. 10. System updates and displays the attribute in the attribute list.
Alternative Sequences/Flows	<p>A1 – Missing required fields - Admin submits the form without entering required information (e.g., attribute name or value type). → System displays validation messages and blocks submission.</p> <p>A2 – Duplicate attribute name - Attribute name already exists in the system. → System displays an error message indicating duplicate attribute.</p> <p>A3 – Permission denied - Admin does not have permission to create attributes. → System blocks the action and displays an authorization error message.</p> <p>A4 – System error / Timeout - System fails to create the attribute due to server error or timeout. → System displays an error message and allows retry.</p>

Tạo mới thuộc tính

Tên thuộc tính *

Vd: Màu lông

Kiểu giá trị

Đơn vị

Chọn kiểu giá trị

Vd: kg, cm

Tạo thuộc tính

Danh sách thuộc tính

Nhấn vào một thuộc tính để quản lý option

Tìm kiếm...

Hình dạng đầu	Sửa Xoá
Kiểu: string • Đơn vị: N/A	Sửa Xoá
Hình dạng mõm	Sửa Xoá
Kiểu: string • Đơn vị: N/A	Sửa Xoá
Màu lông	Sửa Xoá
Kiểu: string • Đơn vị: N/A	Sửa Xoá
Dộ dài lông	Sửa Xoá
Kiểu: string • Đơn vị: N/A	Sửa Xoá
Kiểu lông	Sửa Xoá
Kiểu: string • Đơn vị: N/A	Sửa Xoá

Field Name	Description
Attribute Name	Text field allowing Admin to enter the name of the attribute (required).
Value Type	Dropdown allowing Admin to select the data type of the attribute (e.g., string, number).
Unit	Optional text field allowing Admin to specify the measurement unit of the attribute (e.g., kg, cm).
Create Attribute Button	Submits the form and creates a new attribute in the system.
Validation Message	Displays error messages when required fields are missing or invalid.

UC-65: View Attribute List

Primary Actors	Admin
Secondary Actors	None
Description	This use case allows the Admin to view the list of attributes currently available in the system. The Admin can search for attributes, select an attribute to manage its options, and perform edit or delete actions.
Preconditions	Admin has successfully logged into the Admin Portal. Admin has permission to access Attribute Management.
Postconditions	Attribute list is displayed successfully. Admin may proceed to edit, delete, or manage options of a selected attribute.
Normal Sequence/Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin navigates to “Quản lý thuộc tính” (Attribute Management) from the sidebar menu.

	<p>3. System retrieves and displays the list of attributes.</p> <p>4. Admin views attribute information including name, value type, and unit.</p> <p>5. Admin enters a keyword into the search field to filter attributes (optional).</p> <p>6. System updates the attribute list based on the search result.</p> <p>7. Admin selects an attribute to manage its options or clicks Edit/Delete actions.</p>
Alternative Sequences/Flows	<p>A1 – No attribute available - System returns an empty attribute list. → System displays an empty state message indicating no attributes found.</p> <p>A2 – Attribute not found - Search keyword does not match any attribute. → System displays an empty result message.</p> <p>A3 – Permission denied - Admin does not have permission to view attributes. → System blocks access and displays an authorization error message.</p> <p>A4 – System error / Timeout - System fails to retrieve attribute data or exceeds response time. → System displays an error message and allows retry.</p>

Field Name	Description
Attribute Name	Displays the name of the attribute.
Value Type	Displays the data type of the attribute (e.g., string).
Unit	Displays the unit of measurement associated with the attribute (or N/A if not applicable).
Search Field	Allows Admin to search attributes by name.
Edit Action	Allows Admin to edit the selected attribute.

Delete Action	Allows Admin to delete the selected attribute.
Attribute Selection	Allows Admin to select an attribute to manage its options.

UC-66: Edit Attribute

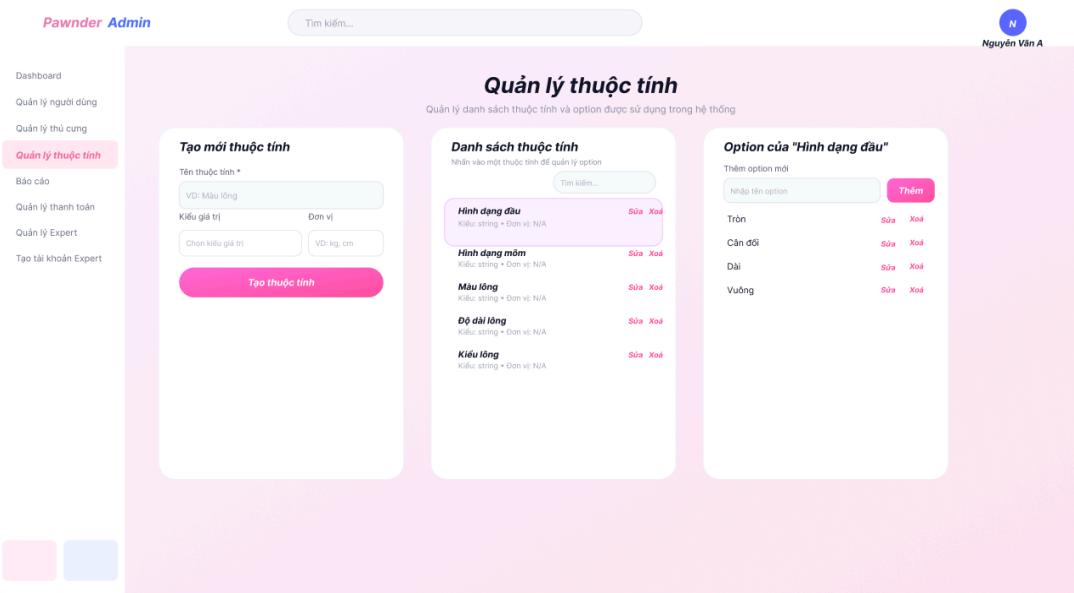
Primary Actors	Admin
Secondary Actors	None
Description	This use case allows the Admin to update an existing attribute by modifying its basic information, including attribute name, value type, and unit.
Preconditions	Admin has successfully logged into the Admin Portal. Admin has permission to access Attribute Management. The selected attribute exists in the system.
Postconditions	Attribute information is updated and saved successfully. Updated attribute information is reflected in the attribute list.
Normal Sequence/Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin navigates to “Quản lý thuộc tính” (Attribute Management). 3. System displays the list of attributes. 4. Admin selects an attribute from the attribute list. 5. System loads the selected attribute information into the Update Attribute form. 6. Admin edits attribute fields (attribute name, value type, unit). 7. Admin clicks “Lưu thay đổi” (Save Changes). 8. System validates the updated data. 9. System updates the attribute information in the database. 10. System refreshes and displays the updated attribute information in the attribute list.
Alternative Sequences/Flows	<p>A1 – Missing required fields - Admin leaves required fields empty (e.g., attribute name) and clicks Save. → System displays validation messages and blocks submission.</p> <p>A2 – Duplicate attribute name - Updated attribute name already exists in the system. → System displays a duplicate name error message and blocks saving.</p> <p>A3 – Permission denied - Admin does not have permission to update attributes. → System blocks the action and displays an authorization error message.</p> <p>A4 – System error / Timeout - System fails to update attribute data due to server error or timeout. → System displays an error message and allows retry.</p> <p>A4 – Delete option not allowed (business constraint) - Admin attempts to delete an option that is currently in use by existing pet profiles. → System displays a warning message and blocks deletion.</p>

Field Name	Description
Attribute Name	Text field allowing Admin to update the name of the attribute (required).
Value Type	Dropdown allowing Admin to update the data type of the attribute (e.g., string).
Unit	Optional field allowing Admin to update the measurement unit (e.g., kg, cm).
Save Changes Button	Saves updated attribute information to the system.
Attribute List	Displays all attributes; selecting one loads its details into the update form.
Validation Message	Displays errors for missing or invalid attribute fields.

UC-67: Delete Attribute

Primary Actors	Admin
Secondary Actors	System
Description	This use case allows the Admin to delete an existing attribute from the system. The system requires confirmation before performing the delete operation.
Preconditions	Admin has successfully logged into the Admin Portal.

	Admin has permission to access Attribute Management. The selected attribute exists in the system.
Postconditions	The selected attribute is removed from the system. The attribute list is refreshed to reflect the deletion.
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin navigates to “Quản lý thuộc tính” (Attribute Management). 3. System displays the list of attributes. 4. Admin clicks “Xóa” (Delete) on the target attribute in the attribute list. 5. System displays a confirmation popup: “Bạn có chắc chắn muốn xóa thuộc tính này?” with OK and Hủy options. 6. Admin clicks OK to confirm deletion. 7. System deletes the selected attribute from the database. 8. System refreshes the attribute list. 9. System displays a success notification to the Admin (optional).
Alternative Sequences / Flows	<p>A1 – Cancel deletion - Admin clicks Hủy on the confirmation popup. → System closes the popup and no data is changed.</p> <p>A2 – Attribute not found - The selected attribute no longer exists (deleted by another operation). → System displays an error message and refreshes the attribute list.</p> <p>A3 – Delete not allowed (in use) (business constraint) - The attribute is currently being used in existing pet profiles. → System blocks deletion and displays a warning message.</p> <p>A4 – Permission denied - Admin does not have permission to delete attributes. → System blocks the action and displays an authorization error message.</p> <p>A5 – System error / Timeout - System fails to process the delete request due to server error or timeout. → System displays an error message and allows retry.</p>



The screenshot shows the Pawnder Admin application interface. On the left, there's a sidebar with navigation links like Dashboard, User Management, Supplier Management, and Product Management. The Product Management section is highlighted in pink and contains a sub-link for Attribute Management, which is also highlighted in pink. The main content area has a header "Quản lý thuộc tính" (Attribute Management) and a sub-header "Quản lý danh sách thuộc tính và option được sử dụng trong hệ thống". There are three main sections:

- Tạo mới thuộc tính**: A form to create a new attribute with fields for Name (Vd: Mẫu lông), Type (Kiểu giá trị), Unit (Đơn vị), and Description (Chọn kiểu giá trị). A "Tạo thuộc tính" button is at the bottom.
- Danh sách thuộc tính**: A list of existing attributes with columns for Name, Type, Unit, and Actions (Sửa, Xoá).
- Option của "Hình dạng đầu"**: A list of options for the "Hình dạng đầu" attribute, showing name, type, unit, and actions (Sửa, Xoá). The options listed are Tròn, Côn đồi, Dài, and Vuông.

 The top right corner shows a user profile for Nguyễn Văn A.

Field Name	Description
Delete Attribute Action	Allows Admin to request deletion of a selected attribute from the attribute list.
Confirmation Popup	Displays a confirmation message before deleting an attribute.
OK Button	Confirms the deletion action and proceeds with removing the attribute.
Cancel Button (Hủy)	Cancels deletion and closes the popup without changes.
Success Notification	Displays a success message after deletion is completed (if implemented).

3.11.4 Expert management

UC-68: View Expert Information

Primary Actors	Admin
Secondary Actors	System
Description	This use case allows the Admin to view detailed information of an expert account. From the expert detail screen, the Admin can update the expert's basic information (excluding email) and reset the expert's password when needed.
Preconditions	Admin has successfully logged into the Admin Portal. Admin has permission to access Expert Management. The selected expert exists in the system.
Postconditions	Expert information is displayed successfully. If Admin updates the expert information, changes are saved and reflected in the system. If Admin resets the password, the system generates a new password for the expert (password is not displayed).
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin navigates to “Quản lý Expert” (Expert Management) from the sidebar menu. 3. System displays the expert list. 4. Admin clicks “Chi tiết” (Detail) for a selected expert. 5. System retrieves and displays the expert detail information. 6. Admin reviews the expert information. 7. Admin edits allowed fields (e.g., full name, gender). 8. Admin clicks “Lưu thông tin” (Save Information). 9. System validates the input and updates the expert profile in the database.

	<p>10. System displays a success message and refreshes the updated information.</p> <p>Reset password (optional within the same screen):</p> <p>11. Admin clicks “Đặt lại mật khẩu” (Reset Password).</p> <p>12. System generates a new password for the expert account and updates it in the database.</p> <p>13. System displays a confirmation message indicating the password has been reset (the password is not shown).</p>
Alternative Sequences / Flows	<p>A1 – Invalid input data</p> <ul style="list-style-type: none"> - Admin enters invalid information (e.g., empty required name) and clicks Save. → System displays validation messages and blocks saving. <p>A2 – Email is not editable</p> <ul style="list-style-type: none"> - Admin attempts to modify email. → System keeps email field fixed/read-only and does not allow changes. <p>A3 – Expert not found</p> <ul style="list-style-type: none"> - Selected expert does not exist or has been removed. → System displays an error message and redirects back to the expert list. <p>A4 – Permission denied</p> <ul style="list-style-type: none"> - Admin does not have permission to view/update expert information. → System blocks the action and displays an authorization error message. <p>A5 – System error / Timeout</p> <ul style="list-style-type: none"> - System fails to load/update expert data or reset password due to server error/timeout. → System displays an error message and allows retry.

Tim kiếm...

Quản lý Expert

Xem danh sách, chi tiết và cập nhật thông tin tài khoản chuyên gia.

ID	HỌ TÊN	EMAIL	NGÀY TẠO	HÀNH ĐỘNG
2	Trần Thị B	expert@pawnder.com	25/11/2025	Chi tiết

Tổng: 1 expert

Field Name	Description
Back to Expert List Button	Navigates Admin back to the expert list screen.

Full Name	Text field allowing Admin to view and update the expert's full name.
Email	Displays expert email address (read-only; cannot be edited).
Gender	Dropdown allowing Admin to select/update the expert's gender.
Save Information Button	Saves updated expert information to the system.
Reset Password Section	Displays instructions for resetting password (password is not displayed after reset).
Reset Password Button	Generates a new password for the expert account and updates it in the system.
Validation Message	Displays errors when input data is missing or invalid.
Success Notification	Displays confirmation messages after saving information or resetting password.

UC-69: Create Expert Account

Primary Actors	Admin
Secondary Actors	System
Description	This use case allows the Admin to create a new expert account in the system. The Admin provides basic information (full name and gender). The system automatically generates the expert email and a suggested password. Admin may regenerate/change the suggested password before creating the account.
Preconditions	Admin has successfully logged into the Admin Portal. Admin has permission to access Expert Management. The system is able to generate a unique expert email.
Postconditions	A new expert account is created successfully. The generated email and password are associated with the expert account. The new expert appears in the expert list (including quick list panel).
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin navigates to “Tạo tài khoản Expert” (Create Expert Account) from the sidebar menu. 3. System displays the Create Expert Account form and the quick expert list panel. 4. Admin enters the expert's Full Name. 5. Admin selects the expert's Gender. 6. System automatically generates the expert Email based on system rules (e.g., expert + smallest available number). 7. System generates a Suggested Password for the expert account. 8. (Optional) Admin clicks “Đổi mật khẩu” (Change Password) to regenerate/change the suggested password.

	<p>9. Admin clicks “Tạo tài khoản” (Create Account).</p> <p>10. System validates input data.</p> <p>11. System creates the expert account and stores the generated credentials.</p> <p>12. System updates the expert list and displays a success notification.</p>
Alternative Sequences / Flows	<p>A1 – Missing required fields</p> <ul style="list-style-type: none"> - Admin clicks Create Account without entering required information (e.g., full name). → System displays validation messages and blocks submission. <p>A2 – Email generation failed (duplicate/conflict)</p> <ul style="list-style-type: none"> - System cannot generate a unique email due to conflict or rule failure. → System displays an error message and requests retry. <p>A3 – Permission denied</p> <ul style="list-style-type: none"> - Admin does not have permission to create expert accounts. → System blocks the action and displays an authorization error message. <p>A4 – System error / Timeout</p> <ul style="list-style-type: none"> - System fails to create the account due to server error or timeout. → System displays an error message and allows retry.

The screenshot shows the Pawnder Admin dashboard with a sidebar containing navigation links like Dashboard, Quản lý người dùng, Quản lý thú cưng, Quản lý thuộc tính, Báo cáo, Quản lý thanh toán, Quản lý Expert, and Tạo tài khoản Expert. The main content area is titled "Tạo tài khoản chuyên gia" and contains a form for creating a new expert account. The form fields are: Họ và tên * (Full Name), Giới tính (Gender), Email tự động (Auto-generated Email), and Mật khẩu để xuất (Password). A note below the email field states: "Hệ thống cấp email dạng expert + số thứ tự nhỏ nhất chưa dùng". To the right, there is a sidebar titled "Danh sách nhanh" showing a list of existing experts: Trần Thị B (expert@pawnder.com). The overall interface is clean with a light blue and white color scheme.

Field Name	Description
Full Name	Text field allowing Admin to enter the expert's full name (required).
Gender	Dropdown allowing Admin to select the expert's gender.
Auto-generated Email	Read-only field showing the email generated automatically by the system for the expert account.

Suggested Password	Displays the system-generated password suggestion for the expert account.
Change Password Button	Regenerates/changes the suggested password before account creation.
Create Account Button	Submits the form and creates a new expert account.
Quick Expert List Panel	Displays a quick list of existing experts for reference.
Validation Message	Displays errors when required fields are missing or invalid.
Success Notification	Displays a confirmation message after expert account creation.

3.11.5 Option management

UC-70: Create Option

Primary Actors	Admin
Secondary Actors	System
Description	This use case allows the Admin to create a new option value for a selected attribute. The Admin selects an attribute from the attribute list, enters the option name, and adds it to the system.
Preconditions	Admin has successfully logged into the Admin Portal. Admin has permission to access Option Management. The selected attribute exists in the system.
Postconditions	A new option is created and stored under the selected attribute. The option list is updated and displays the newly created option.
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin navigates to “Quản lý thuộc tính” (Attribute Management) where options are managed per attribute. 3. System displays the attribute list. 4. Admin selects an attribute from the attribute list. 5. System displays the options of the selected attribute in the option panel. 6. Admin enters an option name in the New Option input field. 7. Admin clicks “Thêm” (Add). 8. System validates the input data. 9. System creates the new option and stores it in the database. 10. System refreshes the option list and displays the newly added option.
Alternative Sequences / Flows	<p>A1 – No attribute selected - Admin attempts to add an option without selecting an attribute. → System displays a message prompting Admin to select an attribute first.</p> <p>A2 – Missing option name - Admin clicks Add with an empty option name.</p>

	<ul style="list-style-type: none"> → System displays validation message and blocks submission. <p>A3 – Duplicate option name</p> <ul style="list-style-type: none"> - The option name already exists under the selected attribute. <ul style="list-style-type: none"> → System displays a duplicate option error message and blocks creation. <p>A4 – Permission denied</p> <ul style="list-style-type: none"> - Admin does not have permission to create options. <ul style="list-style-type: none"> → System blocks the action and displays an authorization error message. <p>A5 – System error / Timeout</p> <ul style="list-style-type: none"> - System fails to create the option due to server error or timeout. <ul style="list-style-type: none"> → System displays an error message and allows retry.
--	---

Field Name	Description
Attribute Selection	Allows Admin to select an attribute from the attribute list to manage its options.
Option Panel Title	Displays the selected attribute name to indicate which attribute the options belong to.
New Option Input	Text field allowing Admin to enter the name/value of the new option (required).

Add Button (Thêm)	Submits the new option and creates it under the selected attribute.
Option List	Displays all existing options under the selected attribute.
Validation Message	Displays errors when input is missing/invalid or option is duplicated.

UC-71: View Option List

Primary Actors	Admin
Secondary Actors	System
Description	This use case allows the Admin to view the list of options associated with a selected attribute. The system displays all existing option values under the chosen attribute.
Preconditions	Admin has successfully logged into the Admin Portal. Admin has permission to access Option Management. The attribute list is available in the system.
Postconditions	The option list of the selected attribute is displayed successfully. Admin can proceed to create, edit, or delete options if permitted.
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin navigates to “Quản lý thuộc tính” (Attribute Management) where options are managed per attribute. 3. System displays the attribute list. 4. Admin selects an attribute from the attribute list. 5. System retrieves all options associated with the selected attribute. 6. System displays the option list under the option section (below “Thêm option mới”).
Alternative Sequences / Flows	<p>A1 – No attribute selected - Admin has not selected any attribute. → System displays an instruction message (e.g., “Hãy chọn một thuộc tính ở danh sách bên cạnh.”).</p> <p>A2 – No options available - The selected attribute has no options. → System displays an empty state message for the option list.</p> <p>A3 – Attribute not found - Selected attribute does not exist or cannot be loaded. → System displays an error message and refreshes the attribute list.</p> <p>A4 – Permission denied - Admin does not have permission to view options. → System blocks access and displays an authorization error message.</p> <p>A5 – System error / Timeout - System fails to retrieve options due to server error or timeout. → System displays an error message and allows retry.</p>

Quản lý thuộc tính

Quản lý danh sách thuộc tính và option được sử dụng trong hệ thống

Danh sách thuộc tính

Nhấn vào một thuộc tính để quản lý option

Tìm kiếm...

Hình dạng đầu

Kiểu: string * Đơn vị: N/A

Sửa Xoá

Hình dạng mõm

Kiểu: string * Đơn vị: N/A

Sửa Xoá

Màu lông

Kiểu: string * Đơn vị: N/A

Sửa Xoá

Độ dài lông

Kiểu: string * Đơn vị: N/A

Sửa Xoá

Kiểu lông

Kiểu: string * Đơn vị: N/A

Sửa Xoá

Option của "Hình dạng đầu"

Thêm option mới

Nhập tên option

Thêm

Tròn

Sửa Xoá

Cân đối

Sửa Xoá

Dài

Sửa Xoá

Vuông

Sửa Xoá

Field Name	Description
Attribute Selection	Allows Admin to select an attribute whose options will be displayed.
Option Section Title	Displays the selected attribute name (e.g., “Option của ‘Hình dạng đầu’”).
Option List	Displays all option values associated with the selected attribute.
Empty State Message	Indicates that no options exist for the selected attribute (if applicable).
Instruction Message	Prompts Admin to select an attribute when none is selected.

UC-72: Edit Option

Primary Actors	Admin
Secondary Actors	System

Description	This use case allows the Admin to update the name/value of an existing option under a selected attribute. The option is edited inline in the option panel and saved to the system.
Preconditions	Admin has successfully logged into the Admin Portal. Admin has permission to access Option Management. An attribute is selected and its option list is displayed. The selected option exists in the system.
Postconditions	The selected option is updated successfully. The option list is refreshed and displays the updated option value.
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin navigates to the attribute screen where options are managed per attribute. 3. Admin selects an attribute to view its options. 4. System displays the option list for the selected attribute. 5. Admin clicks “Sửa” (Edit) on a specific option. 6. System switches the option panel to Edit Mode, displaying the “Cập nhật option” input field with the current option value and buttons “Hủy” (Cancel) and “Lưu” (Save). 7. Admin updates the option value in the input field. 8. Admin clicks “Lưu” (Save). 9. System validates the updated value. 10. System updates the option in the database. 11. System exits edit mode and refreshes the option list showing the updated value.
Alternative Sequences / Flows	<p>A1 – Cancel edit</p> <ul style="list-style-type: none"> - Admin clicks “Hủy” (Cancel). → System exits edit mode and keeps the original option unchanged. <p>A2 – Missing option value</p> <ul style="list-style-type: none"> - Admin clears the input and clicks Save. → System displays validation message and blocks saving. <p>A3 – Duplicate option value</p> <ul style="list-style-type: none"> - Updated option value already exists under the same attribute. → System displays a duplicate option error message and blocks saving. <p>A4 – Option not found</p> <ul style="list-style-type: none"> - The selected option no longer exists (deleted by another operation). → System displays an error message and refreshes the option list. <p>A5 – Permission denied</p> <ul style="list-style-type: none"> - Admin does not have permission to edit options. → System blocks the action and displays an authorization error message. <p>A6 – System error / Timeout</p> <ul style="list-style-type: none"> - System fails to update the option due to server error or timeout. → System displays an error message and allows retry.

Option của "Hình dạng đầu"

Cập nhật option

Tròn

Hủy

Lưu

Tròn

Sửa Xóa

Cân đối

Sửa Xóa

Dài

Sửa Xóa

Vuông

Sửa Xóa

Field Name	Description
Edit Action (Sửa)	Enables edit mode for a selected option.
Update Option Input	Text field allowing Admin to modify the option value (required).
Save Button (Lưu)	Saves the updated option value to the system.
Cancel Button (Hủy)	Cancels editing and exits edit mode without changes.
Option List	Displays all options under the selected attribute and reflects updates after saving.
Validation Message	Displays errors when the option value is empty/invalid or duplicated.
Edit Mode	UI state where the option panel shows the update input and action buttons.

UC-73: Delete Option

Primary Actors	Admin
Secondary Actors	System
Description	This use case allows the Admin to delete an existing option belonging to a selected attribute. The system requires confirmation before deleting the option.
Preconditions	Admin has successfully logged into the Admin Portal. Admin has permission to access Option Management. An attribute is selected and its option list is displayed. The selected option exists in the system.
Postconditions	The selected option is removed from the system. The option list is refreshed and no longer displays the deleted option.
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin navigates to the attribute management screen where options are displayed per attribute. 3. Admin selects an attribute to view its option list. 4. System displays all options of the selected attribute. 5. Admin clicks “Xóa” (Delete) on a specific option. 6. System displays a confirmation popup asking for deletion confirmation (OK / Hủy). 7. Admin clicks OK to confirm deletion. 8. System deletes the selected option from the database. 9. System refreshes the option list and removes the deleted option from the UI. 10. System displays a success notification (if implemented).
Alternative Sequences / Flows	<p>A1 – Cancel deletion - Admin clicks Hủy on the confirmation popup. → System closes the popup and no data is changed.</p> <p>A2 – Option not found - The selected option no longer exists (deleted by another operation). → System displays an error message and refreshes the option list.</p> <p>A3 – Delete not allowed (in use) (business constraint) - The option is currently being used in existing pet profiles. → System blocks deletion and displays a warning message.</p> <p>A4 – Permission denied - Admin does not have permission to delete options. → System blocks the action and displays an authorization error message.</p> <p>A5 – System error / Timeout - System fails to process the delete request due to server error or timeout. → System displays an error message and allows retry.</p>

Option của "Hình dạng đầu"

Cập nhật option

Tròn

Hủy

Lưu

Tròn

Sửa Xóa

Cân đối

Sửa Xóa

Dài

Sửa Xóa

Vuông

Sửa Xóa

Field Name	Description
Delete Option Action (Xóa)	Allows Admin to request deletion of a selected option.
Confirmation Popup	Displays a confirmation message before deleting the option.
OK Button	Confirms the delete action and proceeds with option deletion.
Cancel Button (Hủy)	Cancels deletion and closes the popup without changes.
Option List	Displays all options under the selected attribute and refreshes after deletion.
Success Notification	Displays a success message after the option is deleted (if implemented).
Error Message	Displays error or warning messages when deletion is blocked or fails.

3.11.6 Report management

UC-74: Accept Report

Primary Actors	Admin
Secondary Actors	System
Description	This use case allows the Admin to process an incoming report by reviewing its details and accepting the report. The Admin provides a handling response which is saved in the system and sent to the reporting user.
Preconditions	Admin has successfully logged into the Admin Portal. Admin has permission to access Report Management. At least one report exists in the system. The selected report is in Pending (Đang chờ) status.
Postconditions	The report status is updated to Processed / Accepted (Đã xử lý). Admin's response is saved in the system. The response is sent to the reporting user. The report timeline is updated to reflect the handling action.
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin navigates to “Quản lý báo cáo” (Report Management). 3. System displays the report list with status indicators (Pending / Processed / Rejected). 4. Admin clicks “Xử lý báo cáo” action on a report in the list. 5. System displays the Report Detail page, including report reason, reporter, reported user/content, report content, and timeline. 6. Admin reviews the report details. 7. Admin clicks “Xử lý báo cáo” (Accept Report) button. 8. System displays a modal dialog “Nhập phản hồi xử lý” with a text input field. 9. Admin enters the handling response. 10. Admin clicks “Lưu phản hồi” (Save Response). 11. System validates the input response. 12. System updates the report status to Đã xử lý and saves the response. 13. System sends the response to the reporting user. 14. System updates the report timeline and displays a success notification.
Alternative Sequences / Flows	<p>A1 – Cancel processing - Admin closes the response modal or clicks Hủy. → System closes the modal and no changes are made.</p> <p>A2 – Missing response content - Admin submits the response with empty content. → System displays a validation message and blocks submission.</p> <p>A3 – Report already processed - Admin attempts to process a report that is already Đã xử lý / Tùy chối. → System blocks the action and displays an informational message.</p>

	<p>A4 – Permission denied</p> <ul style="list-style-type: none"> - Admin does not have permission to process reports. → System blocks the action and displays an authorization error message. <p>A5 – System error / Timeout</p> <ul style="list-style-type: none"> - System fails to save the response due to server error or timeout. → System displays an error message and allows retry.
--	--

The screenshot shows the Pawnder Admin application. On the left, there's a sidebar with navigation links like 'Bảng điều khiển', 'Quản lý người dùng', 'Quản lý thú cưng', 'Quản lý thuộc tính', 'Quản lý Policy' (with sub-options 'Policies' and 'Draft Versions'), 'Quản lý từ điển', and a 'Báo cáo' section which is currently selected. The 'Báo cáo' section contains links for 'Quản lý thanh toán', 'Quản lý sự kiện', 'Thông báo hệ thống', 'Quản lý Expert', and '+ Tạo tài khoản Expert'. The main content area is titled 'Chi tiết báo cáo #1'. It shows a report with ID #1, status 'DANG CHỜ' (Pending), reporter 'Lê Minh C' (@user), and reported user 'Lê Minh C' (@user). The 'Lý do báo cáo' (Reason for report) is noted as 'Người dùng bên kia gửi tin nhắn lặp lại gây phiền.' (The user on the other side keeps sending repetitive messages causing trouble.). The 'Mô tả chi tiết' (Description) states 'Người dùng bên kia gửi tin nhắn lặp lại gây phiền.' (The user on the other side keeps sending repetitive messages causing trouble.). The 'Nội dung bị báo cáo' (Content reported) includes a message 'Chào bạn, tôi muốn nhờ bạn tư vấn cho thú cưng của tôi' (Hello you, I want to ask you for advice about my pet). Below this is a 'Timeline' section with a note 'Báo cáo được tạo - 13:46:27 5/1/2026'. At the bottom are two buttons: a green '✓ Xử lý báo cáo' (Handle report) and a red '✗ Từ chối báo cáo' (Reject report).

Field Name	Description
Report List	Displays all reports with ID, reporter, reported user/content, reason, status, date, and actions.
Process Report Action	Allows Admin to start processing a pending report from the report list.
Report Detail Page	Displays full information of the selected report, including reason, description, reported content, and timeline.
Accept Report Button (Xử lý báo cáo)	Initiates the report acceptance and opens the response input modal.
Response Input Field	Text area allowing Admin to enter a handling response (required).
Save Response Button (Lưu phản hồi)	Saves the handling response and completes the report processing.
Cancel Button (Hủy)	Cancels the processing action and closes the response modal.

Report Status	Indicates the current state of the report (Pending / Processed / Rejected).
Timeline	Displays chronological events related to the report lifecycle.
Notification Message	Displays success or error messages after processing actions.

UC-75: Reject Report

Primary Actors	Admin
Secondary Actors	System
Description	This use case allows the Admin to review a report and reject it if the report is considered invalid or does not violate system policies. The Admin provides a rejection response which is saved and sent to the reporting user.
Preconditions	<ul style="list-style-type: none"> - Admin has successfully logged into the Admin Portal. - Admin has permission to access Report Management. - At least one report exists in the system. - The selected report is in Pending (Đang chờ) status.
Postconditions	<ul style="list-style-type: none"> - The report status is updated to Rejected (Từ chối). - Admin's rejection response is saved in the system. - The response is sent to the reporting user. - The report timeline is updated to reflect the rejection action.
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin navigates to “Quản lý báo cáo” (Report Management). 3. System displays the report list with status indicators. 4. Admin clicks “Xử lý báo cáo” action on a pending report. 5. System displays the Report Detail page. 6. Admin reviews the report information, reported content, and description. 7. Admin clicks “Từ chối báo cáo” (Reject Report) button. 8. System displays a modal dialog “Nhập phản hồi xử lý” with a text input field. 9. Admin enters the rejection reason/response. 10. Admin clicks “Lưu phản hồi” (Save Response). 11. System validates the input response. 12. System updates the report status to Từ chối and saves the response. 13. System sends the rejection response to the reporting user. 14. System updates the report timeline and displays a success notification.
Alternative Sequences / Flows	<p>A1 – Cancel rejection</p> <ul style="list-style-type: none"> - Admin closes the modal or clicks Hủy. → System closes the modal and no changes are made. <p>A2 – Missing response content</p> <ul style="list-style-type: none"> - Admin submits an empty rejection response. → System displays a validation message and blocks submission. <p>A3 – Report already processed</p> <ul style="list-style-type: none"> - Admin attempts to reject a report that is already Đã xử lý / Từ chối. → System blocks the action and displays an informational message.

	<p>A4 – Permission denied</p> <ul style="list-style-type: none"> - Admin does not have permission to reject reports. → System blocks the action and displays an authorization error message. <p>A5 – System error / Timeout</p> <ul style="list-style-type: none"> - System fails to save the rejection response due to server error or timeout. → System displays an error message and allows retry.
--	---

Field Name	Description
Report List	Displays all reports with ID, reporter, reported user/content, reason, status, date, and actions.
Report Detail Page	Displays full information of the selected report, including report reason, description, reported content, and timeline.
Reject Report Button (Từ chối báo cáo)	Allows Admin to reject the selected pending report.
Response Input Field	Text area allowing Admin to enter the rejection reason (required).
Save Response Button (Lưu phản hồi)	Saves the rejection response and updates the report status.
Cancel Button (Hủy)	Cancels the rejection action and closes the response modal.
Report Status	Indicates the current state of the report (Pending / Processed / Rejected).

Timeline	Displays chronological events related to the report lifecycle.
Notification Message	Displays success or error messages after rejecting a report.

3.11.7 Event management

UC-76: Create Event

Primary Actors	Admin
Secondary Actors	System
Description	This use case allows the Admin to create a new event in the Pawnder Admin Portal. The Admin provides event information such as event name, description, cover image, start time, submission deadline, end time, reward points, and prize description. After submission, the system validates and stores the event, then displays it in the event list.
Preconditions	<ul style="list-style-type: none"> - Admin has successfully logged into the Admin Portal. - Admin has permission to access Event Management.
Postconditions	<ul style="list-style-type: none"> - A new event is successfully created and stored in the system. - The created event appears in the Event List with the appropriate status based on system rules.
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin navigates to Event Management. 3. System displays the event list and the + Create Event button. 4. Admin clicks + Create Event. 5. System displays the Create Event form. 6. Admin enters the Event Name, optional Description, and selects a Cover Image by uploading a file or entering an image URL. 7. Admin enters Start Time, Submission Deadline, End Time, Reward Points, and optional Prize Description. 8. Admin clicks Create Event. 9. System validates the input data. 10. System saves the new event to the database. 11. System displays a success notification and refreshes or redirects to the Event List, showing the newly created event.
Alternative Sequences / Flows	<p>A1 – Missing required fields: Admin leaves required fields empty (e.g., Event Name, Start Time, Submission Deadline, End Time) and clicks Create → System displays validation messages and blocks submission.</p> <p>A2 – Invalid date/time logic: Admin enters invalid time values (e.g., End Time earlier than Start Time, or Submission Deadline outside the event duration) → System displays an error message and blocks submission.</p> <p>A3 – Invalid cover image: Admin uploads an unsupported image format, exceeds size limits, or enters an invalid image URL → System displays an error and requests correction.</p> <p>A4 – Permission denied: Admin does not have permission to create events → System blocks the action and displays an authorization error.</p>

	A5 – System error / Timeout: The system fails to create the event due to a server error or timeout → System displays an error message and allows retry.
--	---

The screenshot shows the Pawnder Admin interface for creating an event. The left sidebar has a navigation menu with various options like 'Bảng điều khiển', 'Quản lý người dùng', 'Quản lý thú cưng', 'Quản lý thuộc tính', 'Quản lý Policy', 'Quản lý từ cảm', and 'Báo cáo'. A pink highlighted section in the sidebar says 'Quản lý sự kiện' and lists 'Thông báo hệ thống', 'Quản lý Expert', and '+ Tạo tài khoản Expert'. The main area is a form titled 'Tạo sự kiện *'. It has fields for 'Tên sự kiện *' (with placeholder 'VD: Mèo ngáo xuân nhất 2026'), 'Mô tả' (with placeholder 'Mô tả chi tiết về cuộc thi...'), 'Ảnh bìa' (with 'Upload ảnh' and 'Nhập URL' buttons and a placeholder 'https://example.com/cover.jpg'), 'Thời gian bắt đầu *' (date picker), 'Hạn nộp bài *' (date picker), 'Thời gian kết thúc *' (date picker), 'Điểm thường' (with value '0'), and 'Mô tả giải thưởng' (with placeholder 'VD: Top 3 nhận 100/50/30 điểm VIP'). At the bottom are 'Hủy' and 'Tạo sự kiện' buttons.

Field Name	Description
Event List Page	Displays the list of events, including search, status filter, event table, and the Create Event button.
+ Create Event Button	Opens the Create Event form.
Event Name	Text input for the event name (required).
Description	Text area for detailed event description (optional).
Cover Image – Upload	Allows Admin to upload an image file as the event cover (optional, subject to format and size rules).
Cover Image – URL	Allows Admin to provide an image URL as the event cover (optional).
Start Time	Date-time picker defining when the event starts (required).
Submission Deadline	Date-time picker defining the deadline for submissions (required).

End Time	Date-time picker defining when the event ends (required).
Reward Points	Numeric input for reward points granted to winners (default value may be 0).
Prize Description	Text area describing prizes or rewards (optional).
Cancel Button	Cancels event creation and returns to the Event List without saving data.
Create Event Button	Submits the form, triggers validation, and creates the event.
Validation Message	Displays errors for missing fields, invalid formats, or incorrect time logic.
Success Notification	Displays confirmation that the event has been created successfully (if implemented).

UC-77: View Event List

Primary Actor	Admin
Secondary Actor	System
Description	This use case allows the Admin to view the list of events in the system. The Admin can browse events, search by event name, and filter events by status (e.g., Ongoing, Voting Ended, Finished).
Preconditions	<ul style="list-style-type: none"> - Admin has successfully logged into the Admin Portal. - Admin has permission to access Event Management.
Postconditions	<ul style="list-style-type: none"> - The event list is displayed according to the selected search keyword and filter criteria. - No system data is modified.
Normal Sequence/Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin navigates to “Quản lý Sự kiện” (Event Management) from the sidebar menu. 3. System displays the event list with default settings (all events, all statuses). 4. Admin views the list of events, including event name, status, time, number of submissions, vote count, and creation date. 5. Admin enters a keyword in the Search by event name input field (optional). 6. Admin selects a value from the Status filter dropdown (optional). 7. System filters and refreshes the event list based on the search keyword and selected status. 8. Admin may navigate between pages using pagination controls (if available).
Alternative Sequences/Flows	A1 – No matching events found

	<ul style="list-style-type: none"> Admin searches or filters with criteria that match no events. → System displays an empty list or a “No events found” message. <p>A2 – Invalid or empty search input</p> <ul style="list-style-type: none"> Admin enters an empty or invalid keyword. → System ignores the input and displays the full event list. <p>A3 – Permission denied</p> <ul style="list-style-type: none"> Admin does not have permission to access Event Management. → System blocks access and displays an authorization error message. <p>A4 – System error / Timeout</p> <ul style="list-style-type: none"> System fails to retrieve the event list due to a server error or timeout. → System displays an error message and allows the Admin to retry.
--	--

TÊN SỰ KIỆN	TRẠNG THÁI	THỜI GIAN	BÀI ĐƯỢC THI	LUỢT VOTE	THAO TÁC
Mèo ngủ xuất tahats test...	ĐÃ KẾT THÚC	20:34 14-01 — 20:36 14-01	1	1	
Mèo ngủ xáu nhất 2026 gửi ảnh mèo của bạn nhé...	ĐÃ KẾT THÚC	00:09 14-01 — 00:23 14-01	1	1	
test test...	HẾT HẠN VOTE	00:18 12-01 — 00:23 12-01	0	0	
test teste...	HẾT HẠN VOTE	00:16 12-01 — 00:18 12-01	0	0	
test teste...	HẾT HẠN VOTE	00:14 12-01 — 00:16 12-01	0	0	
Mèo ngủ xáu nhất 2026 Hãy gửi ảnh bé mèo của bạn vào đây kkk...	ĐÃ KẾT THÚC	23:46 11-01 — 23:53 11-01	1	1	

Field Name	Description
Event List	Displays all events with basic information such as name, status, time, number of submissions, votes, and creation date.
Search Input	Allows Admin to search events by event name.
Status Filter	Dropdown allowing Admin to filter events by status (e.g., All, Ongoing, Voting Ended, Finished).
Pagination Controls	Allows Admin to navigate between pages of the event list (if applicable).

View Event Action	Allows Admin to view detailed information of a selected event (if implemented).
-------------------	---

UC-78: View Event Detail

Primary Actor	Admin
Secondary Actor	System
Description	This use case allows the Admin to view detailed information of a selected event, including event metadata, submissions, voting statistics, and winner ranking.
Preconditions	<ul style="list-style-type: none"> - Admin has successfully logged into the Admin Portal. - The selected event exists in the system.
Postconditions	<ul style="list-style-type: none"> - Event details are displayed for review. - No data is modified in this use case.
Normal Sequence/Flow	<p>Admin navigates to Event Management. System displays the event list. Admin clicks “Xem chi tiết” (View Detail) on a selected event. System displays the Event Detail page, including:</p> <ul style="list-style-type: none"> • Event name, description, cover image • Event status • Start time, submission deadline, end time • Reward information • Total number of submissions • Total number of votes <p>System displays Top participants / winners (if available). System displays the list of submitted entries with:</p> <ul style="list-style-type: none"> • Pet image • Pet name • Owner • Vote count • Submission time
Alternative Sequences/Flows	<p>A1 – Event not found → System displays an error message and redirects back to the event list.</p>

The screenshot shows the Pawnder Admin dashboard with the following details:

- Left Sidebar:**
 - Pawnder Admin
 - Bảng điều khiển
 - Quản lý người dùng
 - Quản lý thú cưng
 - Quản lý thuộc tính
 - Quản lý Policy
 - Quản lý từ cảm
 - Báo cáo
 - Quản lý sự kiện
 - Thông báo hệ thống
 - Quản lý Expert
 - Tạo tài khoản Expert
- Main Content Area:**
 - Event Details:** Mèo ngủ xuất nhanh (DÃ KẾT THÚC), test, Bắt đầu: 20:34 14/01/2026, Hạn nộp: 20:35 14/01/2026, Kết thúc: 20:36 14/01/2026.
 - Voting Results:**

1	BÀI ĐỨC THI	1	LƯỢT VOTE
Top 3 Người thắng cuộc	MyMy	1 votes	
 - Submission Table:**

Bài dự thi (1)		Bảng xếp hạng			
#	ÁNH	PET	CHỦ NHÂN	VOTES	NGÀY NỘP
1		MyMy	Lê Minh D	1	20:34 14/01/2026

Field Name	Description
Back to Event List Button	Allows Admin to return to the event list page.
Edit Event Button	Allows Admin to navigate to the event edit page (if editing is supported).
Event Cover Image	Displays the cover image of the selected event.
Event Name	Displays the name of the event.
Event Status	Displays the current status of the event (e.g., Ongoing, Voting Ended, Finished).
Event Description	Displays the detailed description of the event.
Start Time	Displays the start date and time of the event.
Submission Deadline	Displays the deadline for submitting entries to the event.
End Time	Displays the end date and time of the event.
Reward Information	Displays the reward or prize description of the event.
Total Submissions	Displays the total number of submissions for the event.
Total Votes	Displays the total number of votes received in the event.

Top Winners Section	Displays the top-ranked participants based on voting results (if available).
Winner Rank Badge	Indicates the ranking position of winners (e.g., 1st place).
Participant Name	Displays the name of the pet or participant in the event.
Owner Name	Displays the name of the user who submitted the entry.
Vote Count	Displays the number of votes received by each submission.
Submission List Tab	Displays the list of submitted entries for the event.
Ranking Tab	Displays ranking or leaderboard information based on votes (if implemented).
Submission Image	Displays the image submitted by the participant.
Submission Time	Displays the time when the entry was submitted.

UC-79: Edit Event

Primary Actor	Admin
Secondary Actors	System
Description	This use case allows the Admin to edit an existing event by accessing the event detail page, modifying permitted event information, and saving the changes.
Preconditions	Admin is logged into the Pawnder Admin system. The event already exists in the system. Admin has permission to manage events.
Postconditions	Event information is updated successfully. Updated information is displayed on the Event Detail page and Event List.
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin navigates to Event Management. 2. System displays the event list. 3. Admin selects an event and clicks View Detail. 4. System displays the Event Detail screen. 5. Admin clicks the Edit button. 6. System displays the Edit Event form with pre-filled data. 7. Admin modifies the desired event information. 8. Admin clicks Save Changes. 9. System validates the input data. 10. System saves the updated information and returns to the Event Detail page.
Alternative Sequences / Flows	<p>A1 – Invalid Input:</p> <ul style="list-style-type: none"> - If required fields are missing or invalid, the system displays error messages and does not save changes.

A2 – Restricted Fields:

- If the event has started, the Start Time field cannot be edited.
- If the submission deadline has passed, the Submission Deadline field cannot be edited.

A3 – Cancel Editing:

- Admin clicks Cancel.
- System discards all changes and returns to the Event Detail page.

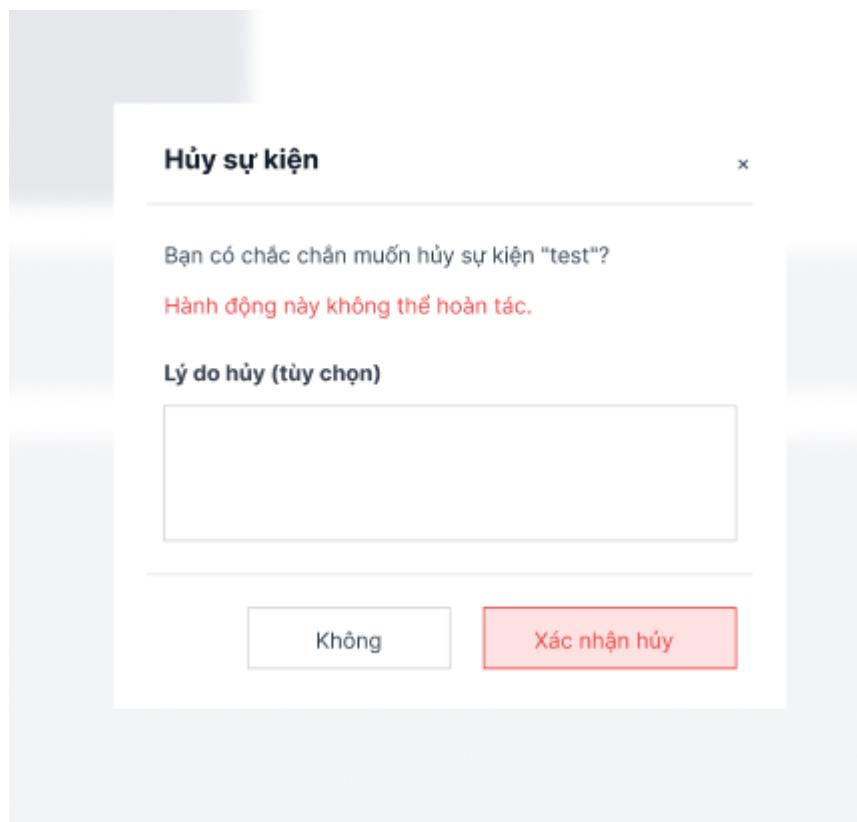
The screenshot shows the 'Chỉnh sửa sự kiện' (Edit Event) page. On the left, there's a sidebar with navigation links: Bảng điều khiển, Quản lý người dùng, Quản lý thú cưng, Quản lý thuộc tính, Quản lý Policy, Quản lý từ cấm, Báo cáo, and Quản lý sự kiện (selected). The main area has a title '← Quay lại' and 'Chỉnh sửa sự kiện'. It contains fields for 'Tên sự kiện' (Mèo ngủ xuất náhats), 'Mô tả' (test), 'Thời gian' (Start: 01/14/2026 08:34 PM, End: 01/14/2026 08:35 PM), 'Mô tả giải thưởng' (test), and an 'Ảnh bìa sự kiện' section with a preview, upload button, and URL input field (https://example.com/cover.jpg). Buttons at the bottom include 'Hủy' and 'Lưu thay đổi'.

Field Name	Description
Event Name	Editable text field for updating the event title. This field is required.
Description	Text area allowing Admin to update the event description.
Cover Image	Allows Admin to upload a new image or enter an image URL for the event cover.
Start Time	Displays the event start time. This field is disabled if the event has already started.
Submission Deadline	Displays the submission deadline. This field is disabled if the deadline has passed.
End Time	Allows Admin to update the event end time. Must be later than the start time.

Reward Points	Numeric field to update reward points for the event.
Reward Description	Text area describing rewards for winners.
Save Changes Button	Saves all updated event information after validation.
Cancel Button	Cancels editing and returns to the Event Detail page without saving changes.

UC-80: Delete Event

Primary Actors	Admin
Secondary Actors	System
Description	This use case allows the Admin to cancel (delete) an existing event that is either ongoing or has reached the voting deadline. The event will be marked as canceled after confirmation.
Preconditions	Admin is logged into the system. The selected event exists. The event status is Ongoing or Voting Ended.
Postconditions	The event is marked as Canceled. The cancellation reason (if provided) is saved in the system. The event is no longer available for participation or voting.
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin navigates to Event Management. 2. Admin selects an event and clicks View Event Detail. 3. Admin clicks the Cancel Event button. 4. The system displays a confirmation dialog. 5. Admin enters an optional cancellation reason. 6. Admin clicks Confirm Cancel. 7. The system cancels the event and updates its status.
Alternative Sequences / Flows	<p>A1: Admin clicks Cancel → The system closes the dialog and no changes are made.</p> <p>A2: System error occurs during cancellation → The system displays an error message and keeps the event unchanged.</p>



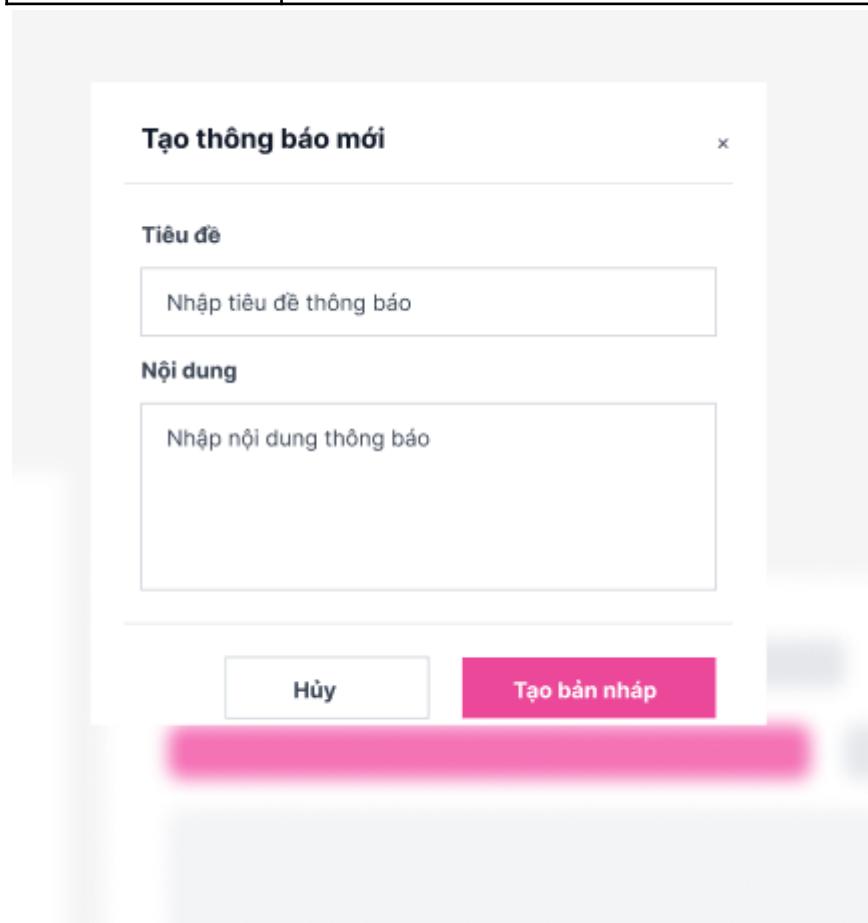
Field Name	Description
Event Information	Displays event name, cover image, description, status, start time, submission deadline, end time, and reward details.
Cancel Event Button	Allows Admin to initiate the event cancellation process.
Confirmation Dialog	A modal dialog asking Admin to confirm the cancellation action.
Cancellation Reason Input	Optional text field allowing Admin to enter a reason for canceling the event.

3.11.8 Notification management

UC-81: Create Notification

Primary Actors	Admin
Secondary Actors	System
Description	This use case allows the Admin to create a new system notification and save it as a draft before sending it to users.
Preconditions	Admin is logged into the Admin Dashboard. Admin has permission to manage system notifications.

Postconditions	A new notification is successfully created and saved in Draft status. The notification appears in the Draft list.
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin navigates to System Notifications. 2. Admin clicks Create Notification. 3. The system displays the Create Notification modal. 4. Admin enters the notification title and content. 5. Admin clicks Create Draft. 6. The system saves the notification as a draft and closes the modal. 7. The draft notification is displayed in the Draft list.
Alternative Sequences / Flows	<p>A1: Admin clicks Cancel → The system closes the modal without saving.</p> <p>A2: Required fields are empty → The system displays validation error messages and prevents saving.</p>



Field Name	Description
Create Notification Button	Allows Admin to open the Create Notification modal.
Notification Title	Text field for entering the notification title.
Notification Content	Text area for entering the notification message content.
Create Draft Button	Saves the notification as a draft without sending it.

Cancel Button	Closes the modal without saving any data.
Draft Notification List	Displays all notifications saved in Draft status.

UC-82: Send Notification

Primary Actors	Admin
Secondary Actors	System
Description	This use case allows the Admin to send a system notification that is currently saved as a draft to users.
Preconditions	Admin is logged into the Admin Dashboard. At least one notification exists in Draft status.
Postconditions	The selected notification is successfully sent to users. The notification status changes from Draft to Sent. The notification appears in the Sent list.
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin navigates to System Notifications. 2. The system displays the list of draft notifications. 3. Admin selects a draft notification. 4. Admin clicks the Send button. 5. The system sends the notification to users. 6. The system updates the notification status to Sent. 7. The notification is moved to the Sent list.
Alternative Sequences / Flows	<p>A1: Admin cancels the send action → The notification remains in Draft status.</p> <p>A2: System error occurs while sending → The system displays an error message and keeps the notification in Draft status.</p>

Pawnder Admin

- Bảng điều khiển
- Quản lý người dùng
- Quản lý thủ cung
- Quản lý thuộc tính
- Quản lý Policy
- Quản lý từ cấm
- Báo cáo
- Quản lý thanh toán
- Quản lý sự kiện

Thông báo hệ thống

Tạo thông báo

Bản nháp (1) Đã gửi (1)

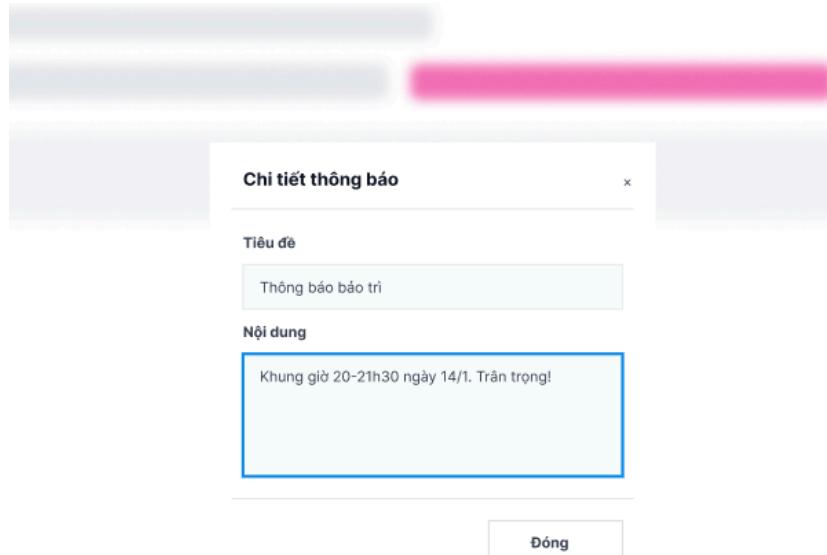
Tiêu đề	Nội dung	Người tạo	Ngày gửi	Thao tác
Thông báo bảo trì	Khung giờ 20-21h30 ngày 14/1. Trân trọng!	Nguyễn Văn A	18:06:24 15/1/2026	Xem ĐÃ GỬI

Field Name	Description
Draft Notification List	Displays all notifications saved in Draft status.
Notification Title	Shows the title of the draft notification.
Notification Content	Shows the content of the draft notification.
Send Button	Sends the selected draft notification to users.
Edit Button	Allows Admin to edit the draft notification before sending.
Delete Button	Allows Admin to delete the draft notification.
Sent Notification Tab	Displays all notifications that have been successfully sent.

UC-83: View Notification

Primary Actors	Admin
Secondary Actors	System
Description	This use case allows the Admin to view details of a notification that has already been sent to users. Admin can access the list of sent notifications and open a specific notification to review its title and content in read-only mode.

Preconditions	<ul style="list-style-type: none"> - Admin has successfully logged into the Admin Portal. - Admin has permission to access Notification Management. - At least one notification has been sent.
Postconditions	<ul style="list-style-type: none"> - The selected notification details are displayed to the Admin. - No data is modified during this process.
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin navigates to System Notification Management. 3. System displays two tabs: Draft and Sent. 4. Admin selects the Sent tab. 5. System displays a list of sent notifications. 6. Admin clicks the View button of a selected notification. 7. System displays a popup/modal showing notification details (title and content). 8. Admin reviews the notification information. 9. Admin clicks Close to exit the detail view.
Alternative Sequences / Flows	<p>A1 – No sent notifications - The sent notification list is empty. → System displays a message indicating no sent notifications are available.</p> <p>A2 – Notification not found - The selected notification no longer exists or cannot be loaded. → System displays an error message and refreshes the sent list.</p> <p>A3 – Permission denied - Admin does not have permission to view notifications. → System blocks access and displays an authorization error message.</p> <p>A4 – System error - System fails to load notification details. → System displays an error message and allows retry.</p>

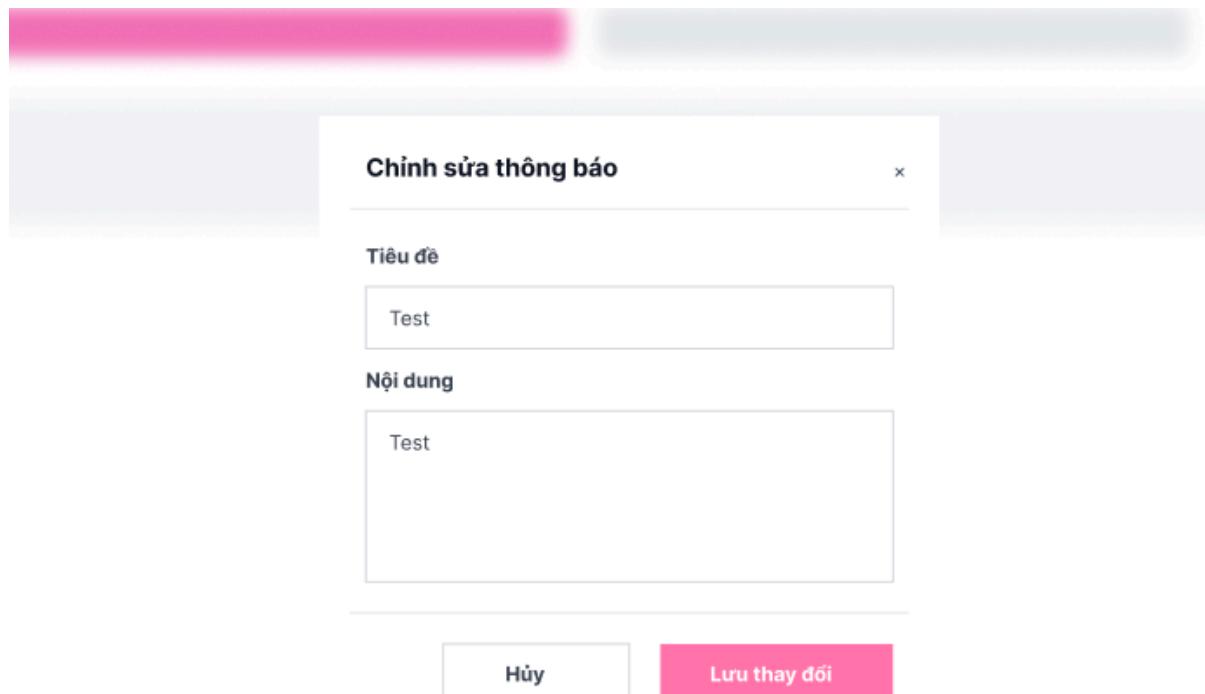


Field Name	Description
Sent Tab	Displays all notifications that have already been sent to users.
Notification List	Shows a list of sent notifications with basic information such as title, content preview, creator, and sent date.
View Button	Allows Admin to open and view detailed information of a selected sent notification.
Notification Detail Modal	Displays the full title and content of the selected notification in read-only mode.
Title Field	Shows the title of the sent notification.
Content Field	Shows the content/body of the sent notification.
Close Button	Closes the notification detail modal and returns to the sent notification list.

UC-84: Edit Notification

Primary Actors	Admin
Secondary Actors	System
Description	This use case allows the Admin to edit an existing notification draft. Admin can modify the notification title and content before saving the changes. Only notifications in Draft status can be edited.
Preconditions	-- Admin has successfully logged into the Admin Portal. - Admin has permission to access Notification Management. - The selected notification exists and is in Draft status.
Postconditions	- The notification draft is updated with the new information. - The updated notification remains in Draft status. - The notification list reflects the updated information.
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin navigates to System Notification Management. 3. System displays the Draft notification list. 4. Admin selects a draft notification and clicks the Edit button. 5. System opens the Edit Notification modal/form with existing data populated. 6. Admin modifies the notification title and/or content. 7. Admin clicks Save Changes. 8. System validates the input data. 9. System saves the updated notification draft. 10. System displays a success message and refreshes the draft list.

Alternative Sequences / Flows	A1 – Cancel editing - Admin clicks Cancel or closes the modal. → System discards changes and returns to the draft list. A2 – Invalid input - Required fields (title or content) are empty or invalid. → System displays validation errors and prevents saving. A3 – Notification not editable - The notification has already been sent. → System blocks editing and displays a warning message. A4 – System error - System fails to update the notification. → System displays an error message and allows retry.
-------------------------------	--



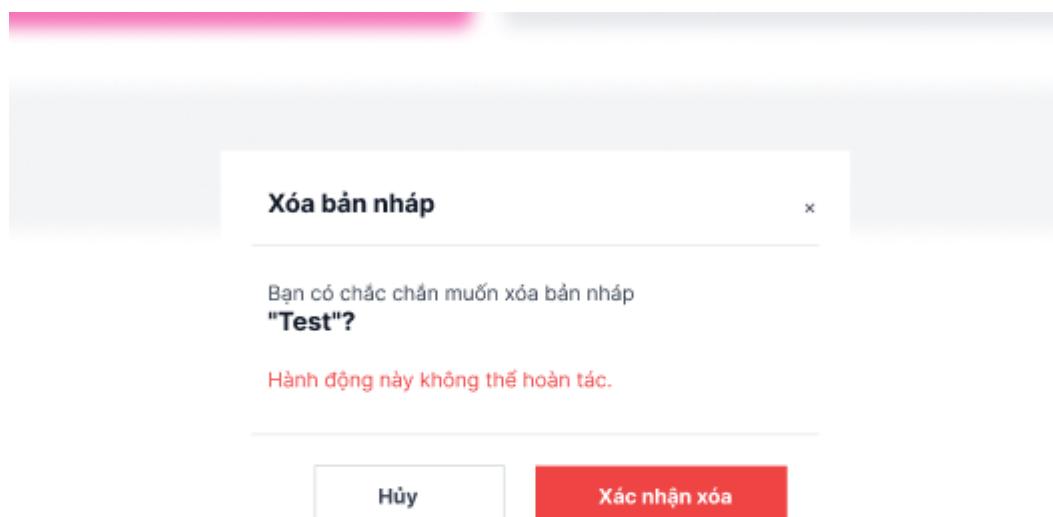
The screenshot shows a modal window titled "Chỉnh sửa thông báo". Inside the modal, there are two input fields: one for the title ("Tiêu đề") containing "Test" and one for the content ("Nội dung") also containing "Test". At the bottom of the modal are two buttons: "Hủy" (Cancel) and a pink "Lưu thay đổi" (Save changes) button.

Field Name	Description
Draft Notification List	Displays all notifications currently in Draft status.
Edit Button	Allows Admin to open the selected draft notification for editing.
Edit Notification Modal	A popup/modal that allows Admin to modify notification details.
Title Field	Editable input field for the notification title.
Content Field	Editable text area for the notification content.

Save Changes Button	Saves the updated notification information.
Cancel Button	Cancels the edit operation and closes the modal without saving changes.
Validation Message	Displays error messages when input data is invalid.
Success Notification	Displays a confirmation message after successful update.

UC-85: Delete Notification

Primary Actors	Admin
Secondary Actors	System
Description	This use case allows the Admin to delete an existing notification draft from the system. The deletion action requires confirmation via a popup dialog to prevent accidental removal. Only notifications in Draft status can be deleted.
Preconditions	Admin has successfully logged into the Admin Portal. Admin has permission to manage system notifications. The selected notification exists and is in Draft status.
Postconditions	The selected notification is permanently removed from the system. The notification no longer appears in the Draft list. The system updates the notification list accordingly.
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin logs into the Admin Portal. 2. Admin navigates to System Notification Management. 3. System displays the Draft notification list. 4. Admin selects a draft notification and clicks the Delete button. 5. System displays a confirmation popup dialog. 6. Admin clicks Confirm Delete. 7. System deletes the notification from the database. 8. System displays a success message. 9. System refreshes the Draft notification list.
Alternative Sequences / Flows	<p>A1 – Cancel editing - Admin clicks Cancel or closes the modal. → System discards changes and returns to the draft list.</p> <p>A2 – Invalid input - Required fields (title or content) are empty or invalid. → System displays validation errors and prevents saving.</p> <p>A3 – Notification not editable - The notification has already been sent. → System blocks editing and displays a warning message.</p> <p>A4 – System error - System fails to update the notification. → System displays an error message and allows retry.</p>



Field Name	Description
Draft Notification List	Displays all notifications that are currently in Draft status.
Delete Button	Allows Admin to initiate deletion of a selected draft notification.
Confirmation Popup	A dialog that asks Admin to confirm or cancel the deletion action.
Confirm Delete Button	Confirms deletion and permanently removes the notification.
Cancel Button	Cancels the delete action and closes the confirmation popup.
Success Message	Displays confirmation that the notification was deleted successfully.
Error Message	Displays when the system fails to delete the notification.

3.11.9 Bad Word management

UC-86: Create Bad Word

Primary Actors	Admin
Secondary Actors	System

Description	This use case allows Admin to create a new bad word in the system by providing required information such as word content, severity level, category, and activation status.
Preconditions	- Admin is logged in successfully. - Admin has permission to manage bad words.
Postconditions	A new bad word is created and stored in the system. The bad word is displayed in the bad word list. If "Activate immediately" is selected, the bad word becomes active.
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin navigates to Bad Word Management. 2. Admin clicks Add Bad Word button. 3. System displays the Add New Bad Word popup. 4. Admin enters the bad word content. 5. Admin selects severity level (Level 1 or Level 2). 6. Admin selects category. 7. Admin chooses activation option (optional). 8. Admin clicks Add New button. 9. System validates input data. 10. System saves the new bad word and closes the popup. 11. System updates and displays the bad word list.
Alternative Sequences / Flows	<p>4a. Missing required fields → System shows validation error and prevents submission.</p> <p>9a. Bad word already exists → System shows duplication warning.</p> <p>8a. Admin clicks Cancel → Popup closes, no data is saved.</p>

Thêm từ cấm mới

Từ cấm *

Mức độ

Danh mục

Kích hoạt ngay

Hủy

Thêm mới

Field Name	Description
Bad Word	Text input for entering the prohibited word content.
Severity Level	Dropdown allowing Admin to select the severity of the bad word (e.g., Level 1 – Mild, Level 2 – Severe).
Category	Dropdown allowing Admin to classify the bad word (e.g., Offensive, Scam).
Activate Immediately	Checkbox allowing Admin to activate the bad word immediately after creation.
Add New Button	Submits the form and creates a new bad word.
Cancel Button	Cancels the action and closes the popup without saving data.

UC-87: View Bad Word List

Primary Actors	Admin
Secondary Actors	System
Description	This use case allows Admin to view the list of bad words currently stored in the system, including their severity level, category, type, and activation status.
Preconditions	Admin is logged in successfully. Admin has permission to access Bad Word Management.
Postconditions	The bad word list is displayed successfully. Admin can view summary statistics and detailed information of each bad word.
Normal Sequence/Flow	<ol style="list-style-type: none"> 1. Admin clicks Bad Word Management in the sidebar. 2. System loads the bad word management page. 3. System displays summary statistics (Total, Active, Level 1, Level 2). 4. System displays the list of bad words with detailed columns. 5. Admin views the bad word list.
Alternative Sequences/Flows	<ol style="list-style-type: none"> 2a. No bad words exist → System displays an empty list message. 2b. System fails to load data → System shows an error notification.

Pawnder Admin

Nguyễn Văn A

Bảng điều khiển
Quản lý người dùng
Quản lý thú cưng
Quản lý thuộc tính
Quản lý Policy

Quản lý từ cấm

Tổng số: 8 | Dang hoạt động: 8 | Level 1: 3 | Level 2: 5

Reload Cache | + Thêm từ cấm

Tìm kiếm từ cấm... | Tất cả Level | Tất cả danh mục | Tất cả trạng thái

TÙ CẤM	LEVEL	DANH MỤC	LOẠI	TRẠNG THÁI	THAO TÁC
lừa đảo	NÂNG (LV2)	Scam	Text	DANG BẤT	...

Field Name	Description
Total Bad Words	Displays the total number of bad words in the system.
Active Bad Words	Displays the number of currently active bad words.
Level 1 Count	Displays the number of Level 1 (mild) bad words.
Level 2 Count	Displays the number of Level 2 (severe) bad words.
Search Input	Allows Admin to search bad words by keyword.
Level Filter	Dropdown allowing Admin to filter bad words by severity level.
Category Filter	Dropdown allowing Admin to filter bad words by category.
Status Filter	Dropdown allowing Admin to filter bad words by activation status.
Bad Word Column	Displays the prohibited word content.
Level Column	Displays the severity level of the bad word.
Category Column	Displays the category assigned to the bad word.
Type Column	Displays the type of bad word (e.g., Text).
Status Column	Displays the activation status of the bad word.
Action Column	Provides actions such as view, edit, activate/deactivate, or delete.

UC-88: Edit Bad Word

Primary Actors	Admin
Secondary Actors	System
Description	This use case allows the Admin to edit an existing bad word in the bad word list, including updating its content, severity level, category, and activation status.
Preconditions	Admin is logged in to the Pawnder Admin system. The bad word list already contains at least one bad word.
Postconditions	The selected bad word is updated successfully. The updated information is reflected immediately in the bad word list. If activated, the new configuration is applied to the content moderation system.
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin navigates to Bad Word Management. 2. System displays the list of bad words. 3. Admin clicks the Edit icon on a specific bad word. 4. System opens the Edit Bad Word modal. 5. Admin updates the bad word information (word, level, category, status). 6. Admin clicks Update. 7. System validates the input and saves the changes. 8. System displays a success message and updates the list.
Alternative Sequences / Flows	<p>A1: Admin clicks Cancel → System closes the modal without saving changes.</p> <p>A2: Invalid or empty input → System displays validation error messages and does not save changes.</p>

Chỉnh sửa từ cấm

Từ cấm *

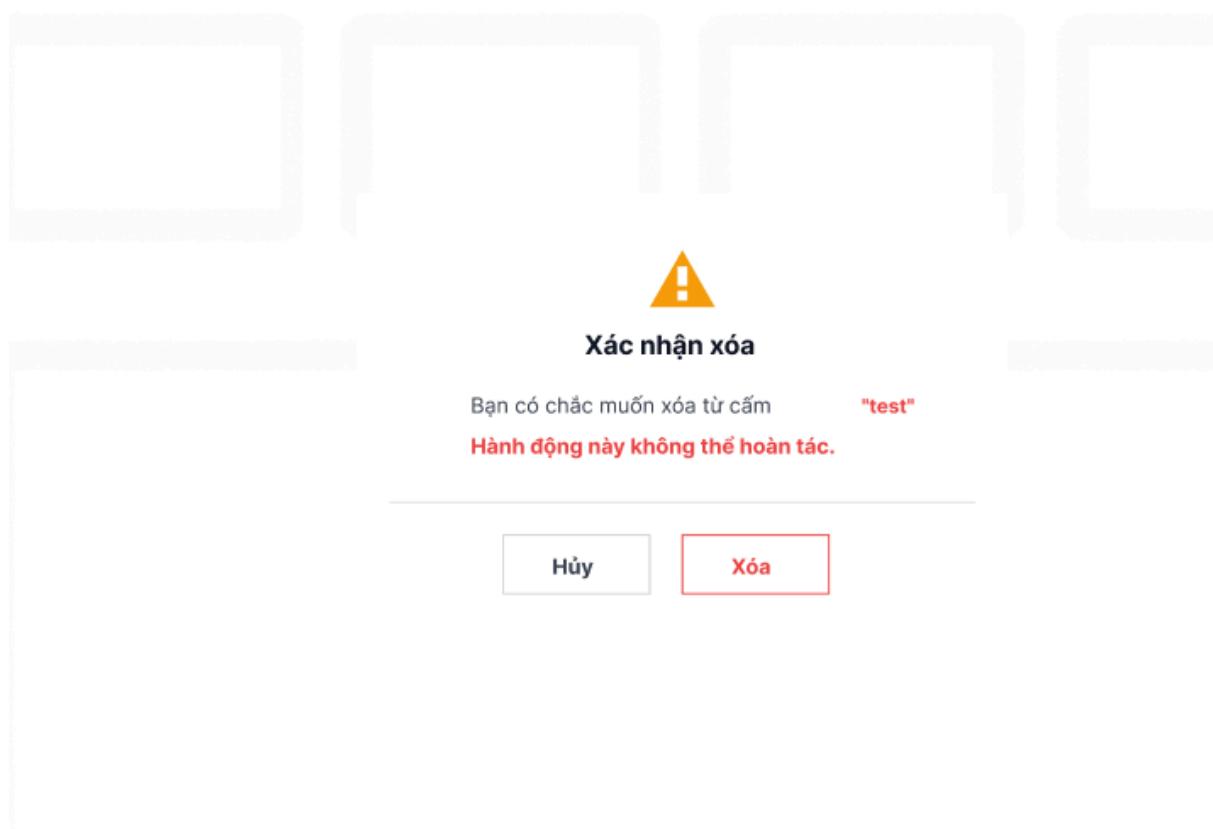
Mức độ

Danh mục

Field Name	Description
Bad Word	The word or phrase that is restricted and needs to be moderated. This field is required.
Severity Level	Indicates the seriousness of the bad word (e.g., Level 1 – Mild, Level 2 – Severe).
Category	Defines the classification of the bad word (e.g., Offensive, Scam).
Activate Immediately	A checkbox to determine whether the bad word is enabled right after updating.
Update Button	Saves the updated bad word information and applies the changes.
Cancel Button	Closes the edit modal without saving any changes.

UC-89: Delete Bad Word

Primary Actors	Admin
Secondary Actors	System
Description	Allows Admin to delete an existing bad word from the bad word list.
Preconditions	Admin is logged in Bad word list is available
Postconditions	Selected bad word is permanently removed from the system Bad word list is updated
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin navigates to Bad Word Management. 2. System displays the list of bad words. 3. Admin clicks the Delete icon of a selected bad word. 4. System displays a confirmation popup. 5. Admin clicks Delete to confirm. 6. System deletes the bad word and refreshes the list.
Alternative Sequences / Flows	4a. Admin clicks Cancel → System closes popup, no data is changed. 5a. System error occurs → System shows an error message and keeps the bad word unchanged.



Field Name	Description
Bad Word List	Displays all existing bad words with level, category, type, and status.
Delete Icon	Allows Admin to initiate deletion of a selected bad word.
Confirmation Popup	Displays a warning message that deletion is irreversible.
Cancel Button	Closes the popup without deleting the bad word.
Delete Button	Confirms deletion and permanently removes the bad word.

3.11.10 Policy management

UC-90: Create Policy

Primary Actors	Admin
Secondary Actors	System
Description	This use case allows the Admin to create a new policy by entering policy information such as policy code, policy name, display order, description, and consent requirement.

Preconditions	Admin is logged into the system. Admin has permission to manage policies.
Postconditions	A new policy is successfully created. The policy appears in the policy list with version v1.
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin navigates to Policy Management. 2. Admin selects Policies from the menu. 3. Admin clicks the Create New Policy button. 4. The system displays the Create Policy modal form. 5. Admin enters required policy information. 6. Admin clicks Create Policy. 7. The system validates the input data. 8. The system saves the new policy and closes the modal. 9. The system updates and displays the policy list.
Alternative Sequences / Flows	<p>5a. Missing required fields: - The system displays validation error messages and prevents submission.</p> <p>7a. Invalid policy code format: - The system displays an error and requires correction before creating the policy.</p> <p>6a. Admin clicks Cancel: - The system closes the modal without saving any data.</p>

Field Name	Description

Policy Code	A unique identifier for the policy (uppercase letters and underscores only).
Display Order	Determines the display sequence of the policy in the system.
Policy Name	The official name of the policy shown to users.
Description	A brief explanation of the policy's purpose and content.
Require User Consent	Indicates whether users must accept this policy before using the system.
Create Policy Button	Submits the form and creates a new policy if all inputs are valid.
Cancel Button	Closes the form without saving any changes.

UC-91: View Policy List

Primary Actors	Admin
Secondary Actors	System
Description	This use case allows the Admin to view the list of existing policies in the system, including their basic information such as policy name, code, status, consent requirement, and current version.
Preconditions	Admin is logged into the system. Admin has permission to access policy management.
Postconditions	The policy list is displayed successfully. No data is modified.
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin navigates to Policy Management. 2. Admin selects Policies from the menu. 3. The system retrieves all available policies. 4. The system displays the policy list screen with policy information.
Alternative Sequences / Flows	<p>3a. No policies available: - The system displays an empty list with a notification message.</p> <p>2a. Access denied: - The system prevents access and displays an authorization error message.</p>

Field Name	Description
Policy Name	Displays the name and short description of the policy.
Policy Code	Displays the unique identifier code of the policy.
Status	Indicates whether the policy is active or inactive.
Mandatory	Indicates whether user consent is required for this policy.
Current Version	Displays the latest version of the policy.
Search Input	Allows Admin to search policies by name or code.
Status Filter	Allows Admin to filter policies by status.
Type Filter	Allows Admin to filter policies by policy type.
Action Buttons	Provides actions such as view details, edit, or delete policy.

UC-92: View Policy Detail

Primary Actors	Admin
Secondary Actors	System
Description	Allows the Admin to view detailed information of a Policy, including general policy information and the list of its versions. The Admin can also view detailed content of each policy version.

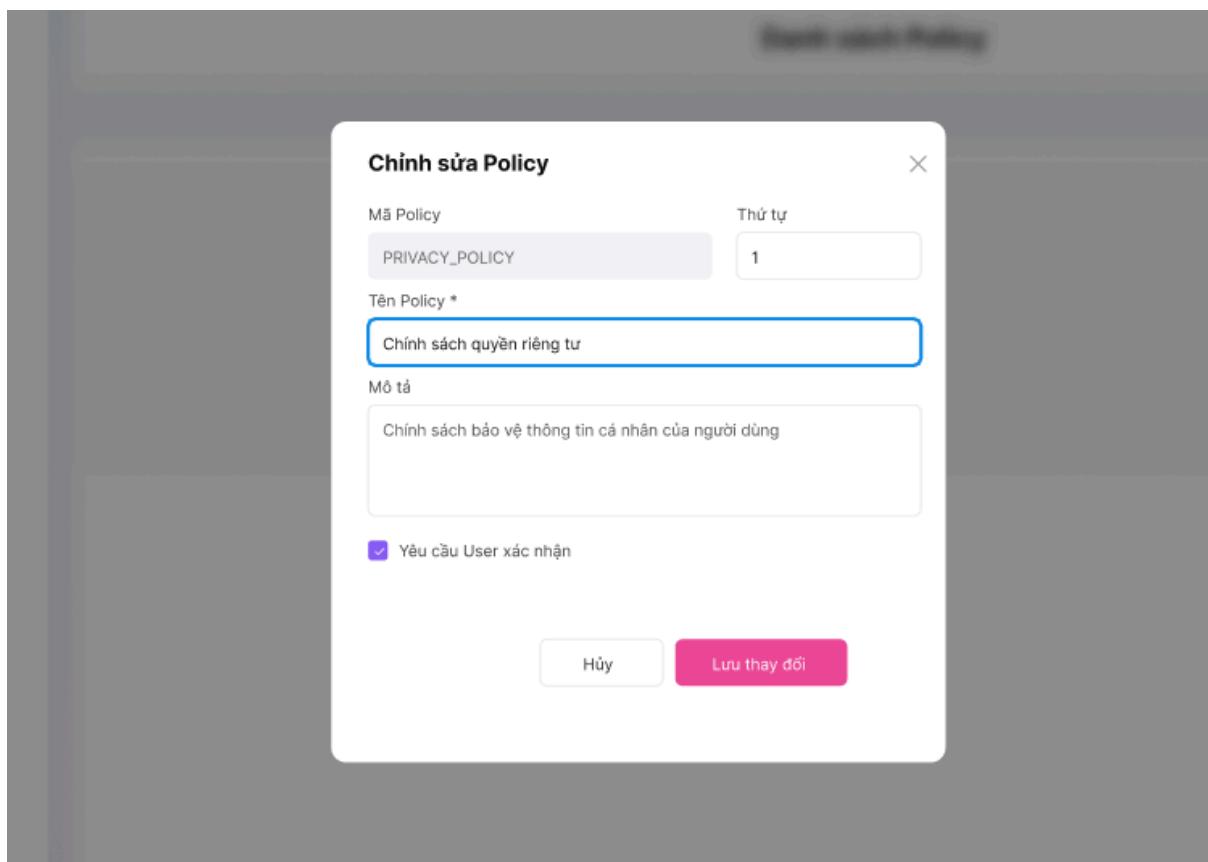
Preconditions	Admin has successfully logged into the system The Policy already exists in the system
Postconditions	Policy detailed information is displayed Admin can view the list of policy versions and access details of each version
Normal Sequence / Flow	1. Admin clicks Policy Management from the menu. 2. Admin selects Policies. 3. The system displays the policy list. 4. Admin clicks the View (eye icon) of a policy. 5. The system displays the Policy Detail screen, including policy information and the list of versions.
Alternative Sequences / Flows	A1: The policy does not exist → The system displays an error message. A2: Admin clicks View on a specific version → The system displays detailed content of the selected version.

Field Name	Description
Policy Code	Unique identifier code of the policy.
Policy Name	Name of the policy.
Description	Short description of the policy content.
Display Order	Order in which the policy is displayed.

Require Confirmation	Indicates whether users are required to accept the policy.
Status	Current status of the policy (Active / Inactive).
Total Versions	Total number of versions of the policy.
Accepted Users	Number of users who have accepted the policy.
Not Accepted Users	Number of users who have not accepted the policy.
Acceptance Rate	Percentage of users who have accepted the policy.

UC-93: Edit Policy

Primary Actors	Admin
Secondary Actors	System
Description	This use case allows the Admin to edit the information of an existing policy, including policy name, display order, description, and confirmation requirement.
Preconditions	<ul style="list-style-type: none"> - Admin is logged into the system - The policy already exists
Postconditions	<ul style="list-style-type: none"> - The policy information is updated successfully - The updated data is saved in the system
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin navigates to Policy Management. 2. Admin selects Policies. 3. The system displays the policy list. 4. Admin clicks the Edit icon of a policy. 5. The system displays the Edit Policy popup. 6. Admin updates the required information. 7. Admin clicks Save Changes. 8. The system validates and updates the policy. 9. The system displays the updated policy in the list.
Alternative Sequences / Flows	<p>A1: Admin clicks Cancel → The popup is closed without saving changes.</p> <p>A2: Required fields are empty or invalid → The system displays validation error messages.</p>

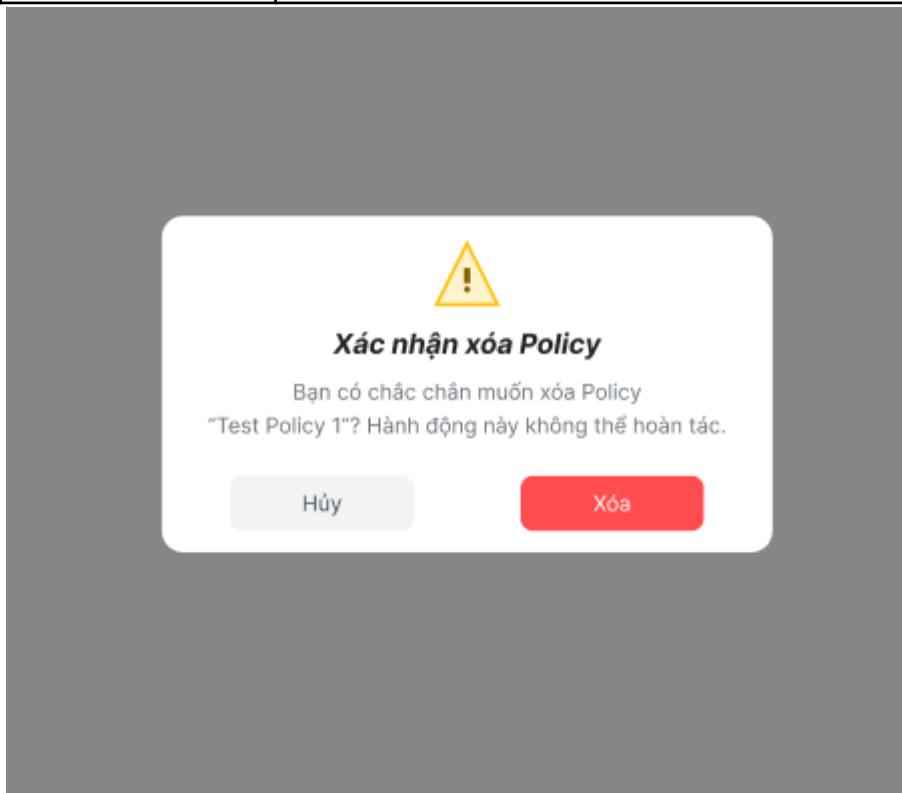


Field Name	Description
Policy Code	Displays the unique code of the policy (read-only).
Display Order	Determines the order in which the policy is shown.
Policy Name	Allows Admin to edit the policy name.
Description	Allows Admin to edit a short description of the policy.
Require User Confirmation	Checkbox indicating whether users must accept the policy.
Save Changes Button	Saves all updated policy information.
Cancel Button	Cancels editing and closes the popup.

UC-94: Delete Policy

Primary Actors	Admin
Secondary Actors	System
Description	Allows the Admin to delete an existing Policy from the system. The system requires confirmation before permanently deleting the policy.

Preconditions	Admin has successfully logged into the system The Policy exists in the system
Postconditions	The selected Policy is removed from the system after confirmation The policy list is updated accordingly
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin clicks Policy Management from the menu. 2. Admin selects Policies. 3. The system displays the policy list. 4. Admin clicks the Delete (trash icon) of a policy. 5. The system displays a confirmation popup. 6. Admin confirms the deletion. 7. The system deletes the policy and refreshes the policy list.
Alternative Sequences / Flows	<p>A1: Admin cancels the confirmation popup → The policy is not deleted. A2: The policy does not exist or has been deleted → The system displays an error message.</p>



Field Name	Description
Confirmation Message	Message informing that the delete action is irreversible.
Cancel Button	Cancels the delete action and closes the popup.
Confirm Delete Button	Confirms deletion and permanently removes the policy.

3.11.11 Policy Draft management

UC-95: Create Policy Draft

Primary Actor	Admin
Secondary Actor	System
Description	Allows Admin to create a new draft version for an existing Policy. The draft contains the updated policy content and changelog information before being published.
Preconditions	Admin has successfully logged into the system At least one Policy already exists
Postconditions	A new Policy draft version is created and saved in Draft status The draft version appears in the Draft Versions list
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin clicks Policy Management on the menu 2. Admin selects Draft Versions 3. System displays the Draft Version Management screen 4. Admin selects a Policy from the Policy list 5. Admin clicks Create New Version 6. System displays the Create Policy Draft form 7. Admin enters version title, policy content, and changelog 8. Admin clicks Create Version 9. System saves the draft and displays it in the draft list
Alternative Sequences / Flows	<p>A1: Required fields are missing → System displays validation error messages</p> <p>A2: Admin clicks Cancel → System discards input and returns to Draft Version list</p>

Quản lý Version nhập

Nhập của "Chính sách quyền riêng tư"

+ Tạo mới

v2.0
Tạo: 17/1/2026

Tạo Version mới

Tiêu đề *

VD: Điều khoản v2.0

Nội dung *

Nội dung đầy đủ của Policy...

Changelog

Mô tả thay đổi...

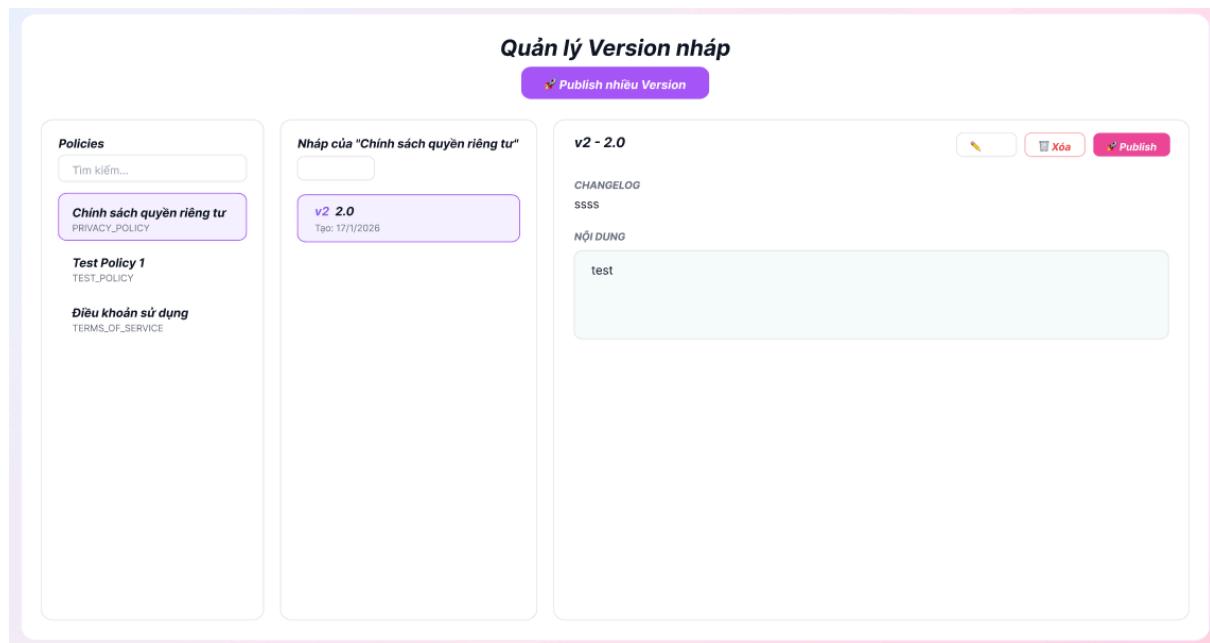
Hủy

Tạo Version

Field Name	Description
Policy	The selected Policy for which the draft version is created
Version Title	Title or name of the draft version (e.g., Privacy Policy v2.0)
Content	Full content of the Policy version
Changelog	Description of changes compared to the previous version
Status	Status of the version (Draft)
Created Date	Date and time when the draft version is created
Action	Create or cancel draft creation

UC-96: View Policy Draft

Primary Actor	Admin
Secondary Actor	System
Description	Allows Admin to view all draft versions of a selected Policy and see the detailed content of a specific draft version.
Preconditions	Admin has successfully logged into the system At least one Policy exists in the system
Postconditions	The list of draft versions for the selected Policy is displayed The content of a selected draft version is shown
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin clicks Policy Management on the menu 2. Admin selects Draft Versions 3. System displays the Draft Version Management screen 4. Admin selects a Policy from the Policy list (Panel 1) 5. System displays the list of draft versions of the selected Policy (Panel 2) 6. Admin selects a specific draft version 7. System displays the detailed content of the selected version (Panel 3)
Alternative Sequences / Flows	<p>A1: Selected Policy has no draft versions → System displays “No draft versions available”</p> <p>A2: Admin does not select any version → System displays a hint message to select a version</p>



Field Name	Description
Version	Version identifier of the draft
Title	Title of the draft version
Status	Status of the version (Draft)
Created Date	Date and time when the draft was created

UC-97: Edit Policy Draft

Primary Actor	Admin
Secondary Actor	System
Description	Allows Admin to edit an existing draft version of a Policy by updating its content and changelog before saving the changes.
Preconditions	Admin has successfully logged into the system At least one Policy exists The selected Policy has at least one draft version
Postconditions	The draft version is updated successfully The updated information is saved and displayed
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin clicks Policy Management on the menu 2. Admin selects Draft Versions 3. System displays the Draft Version Management screen 4. Admin selects a Policy from the Policy list 5. System displays the list of draft versions of the selected Policy 6. Admin selects a draft version 7. System displays the version details

	<p>8. Admin clicks the Edit button 9. Admin updates the version information (title, content, changelog) 10. Admin clicks Save 11. System validates the input and saves the changes 12. System displays the updated draft version details</p>
Alternative Sequences / Flows	<p>A1: Required fields are missing → System displays validation error messages A2: Admin cancels editing → No changes are saved</p>

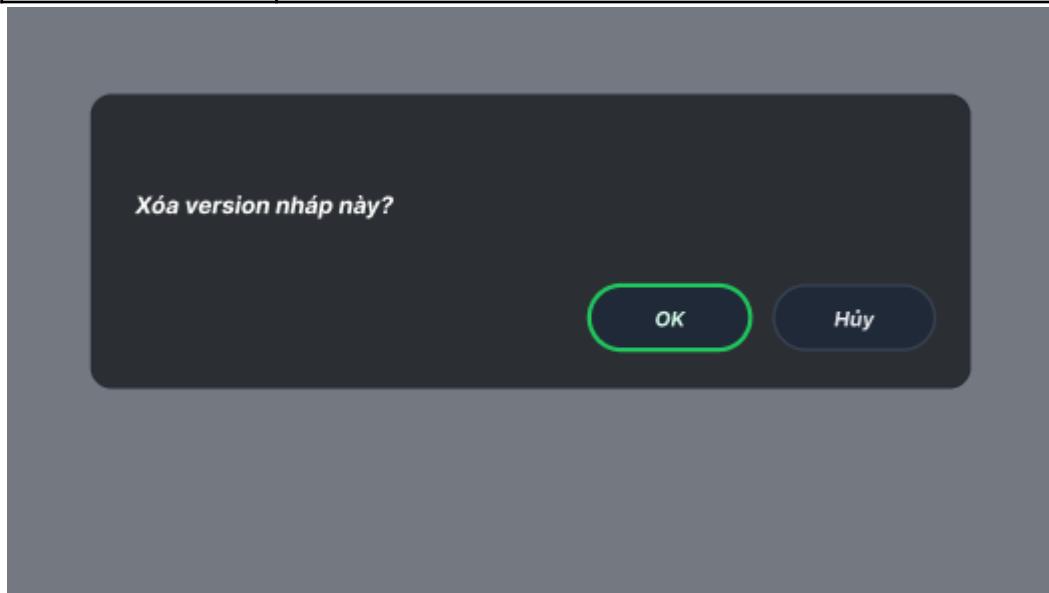
The screenshot displays a user interface for managing policy drafts. On the left, a sidebar lists versions with 'v2 2.0' highlighted. The main area shows the details for 'v2 - 2.0', including the title '2.0' and content 'test'. A 'Changelog' section contains the text 'SSSS'. At the top right are buttons for 'Hủy' (Cancel) and 'Lưu' (Save).

Field Name	Description
Version Title	Title or name of the policy draft version
Content	Full content of the policy version
Changelog	Description of changes made in this version
Status	Status of the version (Draft)
Action – Save	Saves the updated draft version
Action – Cancel	Cancels editing without saving changes
Version Title	Title or name of the policy draft version

UC-98: Delete Policy Draft

Primary Actor	Admin
Secondary Actor	System

Description	Allows Admin to delete an existing draft version of a Policy. This action permanently removes the draft version from the system.
Preconditions	<ul style="list-style-type: none"> - Admin has successfully logged into the system - At least one Policy exists - The selected Policy has at least one draft version
Postconditions	<ul style="list-style-type: none"> - The selected draft version is deleted from the system - The draft version list is updated
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin clicks Policy Management on the menu 2. Admin selects Draft Versions 3. System displays the Draft Version Management screen 4. Admin selects a Policy from the Policy list 5. System displays the list of draft versions of the selected Policy 6. Admin selects a draft version 7. System displays the draft version details 8. Admin clicks the Delete button 9. System displays a confirmation popup 10. Admin confirms the deletion 11. System deletes the draft version 12. System refreshes the draft version list
Alternative Sequences / Flows	<p>A1: Admin cancels deletion → Draft version is not deleted</p> <p>A2: Draft version does not exist → System displays an error message</p>

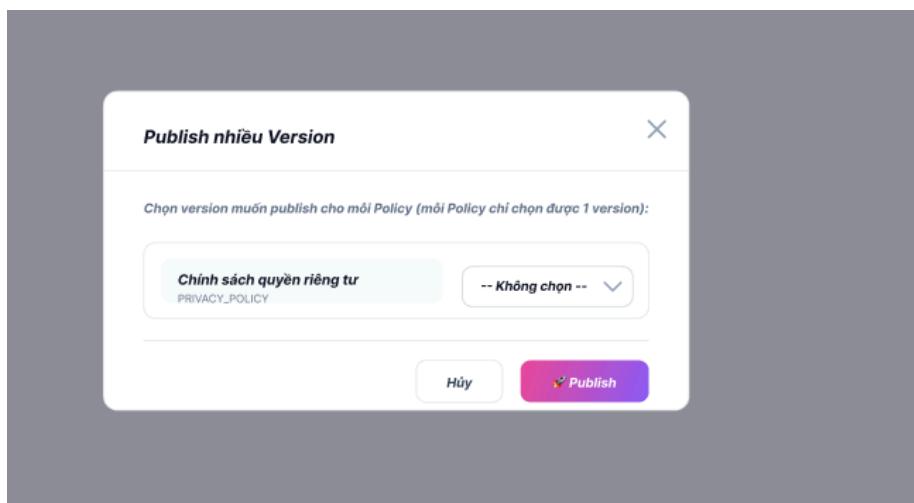


Field Name	Description
Policy Name	Name of the Policy containing the draft version
Version	Version number of the draft version
Status	Status of the version (Draft)

Action – Delete	Triggers the delete draft version action
Confirmation Popup	Confirms whether the Admin wants to delete the draft version
Action – Confirm	Confirms deletion and removes the draft version
Action – Cancel	Cancels deletion without any changes

UC-99: Publish Policy Draft

Primary Actors	Admin
Secondary Actors	System
Description	Allows Admin to publish one or multiple draft versions of Policies at the same time. For each Policy, Admin can select only one draft version to publish, which will become the active version of that Policy.
Preconditions	<ul style="list-style-type: none"> - Admin has successfully logged into the system - At least one Policy has draft versions available
Postconditions	<ul style="list-style-type: none"> - Selected draft versions are published successfully - Published versions become active versions of their corresponding Policies - Draft versions are updated accordingly
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin clicks Policy Management on the menu 2. Admin selects Draft Versions 3. System displays the Draft Version Management screen 4. Admin clicks Publish Multiple Versions 5. System displays a popup with a list of Policies and dropdowns for version selection 6. Admin selects one draft version for each Policy (optional) 7. Admin clicks Publish 8. System publishes the selected versions and updates Policy status
Alternative Sequences / Flows	<p>A1: No version selected → System disables Publish button or shows validation message</p> <p>A2: Selected version is invalid or already published → System displays error message</p> <p>A3: Admin clicks Cancel → Popup closes, no changes applied</p>



Field Name	Description
Policy Name	Name of the Policy
Policy Code	Unique identifier of the Policy
Version Dropdown	Dropdown list of available draft versions for the Policy
Selected Version	The draft version chosen to be published
Publish Button	Confirms publishing selected versions
Cancel Button	Cancels the publishing action and closes the popup

3.11.12 Payment management

UC-100: View Payment History

Primary Actors	Admin
Secondary Actors	System
Description	Allows Admin to view and monitor payment transactions related to Premium upgrades and user payments. The screen provides summary statistics, transaction lists, and revenue charts for tracking payment performance.
Preconditions	- Admin has successfully logged into the system
Postconditions	- Payment history screen is displayed - Admin can view payment statistics, transaction list, and revenue chart
Normal Sequence / Flow	1. Admin clicks Payment Management on the menu 2. System displays the Payment Management screen

	<p>3. System shows total revenue, total transactions, and payment statistics 4. System displays the payment transaction list 5. Admin clicks Chart to view revenue visualization</p>
Alternative Sequences / Flows	<p>A1: No payment data available → System displays “No transactions recorded” message A2: Admin searches by keyword → System filters transaction list accordingly</p>

The screenshot shows the Pawnder Admin interface. On the left, there's a sidebar with navigation links: Dashboard, Quản lý người dùng, Quản lý thú cưng, Quản lý thuộc tính, Báo cáo, Quản lý thanh toán (which is highlighted), Quản lý Expert, and Tạo tài khoản Expert. The main content area has a title "Quản lý thanh toán" and a subtitle "Theo dõi các giao dịch nâng cấp Premium và thanh toán của người dùng." It displays two summary metrics: Tổng doanh thu (119.000 đ) and Tổng giao dịch (3). Below these are three transaction rows:

NGƯỜI DÙNG	SỐ TIỀN	GÓI PREMIUM	TRẠNG THÁI	NGÀY BẮT ĐẦU	NGÀY KẾT THÚC
Lê Minh D	99.000 đ	Gói VIP 1 tháng	✓ ACTIVE	27/11/2025	27/12/2025
Lê Tuấn	10.000 đ	Gói VIP 1 tháng	✓ ACTIVE	27/11/2025	27/12/2025
Lê Minh C	10.000 đ	Gói VIP 1 tháng	✓ ACTIVE	27/11/2025	27/12/2025

Field Name	Description
Total Revenue	Total amount of revenue generated from payments
Total Transactions	Total number of payment transactions
Chart Button	Displays revenue chart visualization
Search Bar	Allows searching by email, user name, or transaction ID
User	User who made the payment
Amount	Payment amount
Premium Package	Premium plan purchased
Status	Payment status (Success / Pending / Failed)
Start Date	Premium subscription start date
End Date	Premium subscription end date
Transaction List	List of recorded payment transactions

3.11.13 Pet management

UC-101: View Pet List

Primary Actors	Admin
Secondary Actors	System
Description	Allows Admin to view a list of all pets registered in the system. The list displays basic pet information, owner details, and provides access to view detailed pet profiles.
Preconditions	- Admin has successfully logged into the system
Postconditions	- Pet list screen is displayed - Admin can view basic information of each pet
Normal Sequence / Flow	1. Admin clicks Pet Management on the menu 2. System displays the Pet List screen 3. System shows a list of all pets with images, pet information, and owner details
Alternative Sequences / Flows	A1: No pets exist in the system → System displays an empty state message A2: Admin uses search bar → System filters pet list based on entered keyword

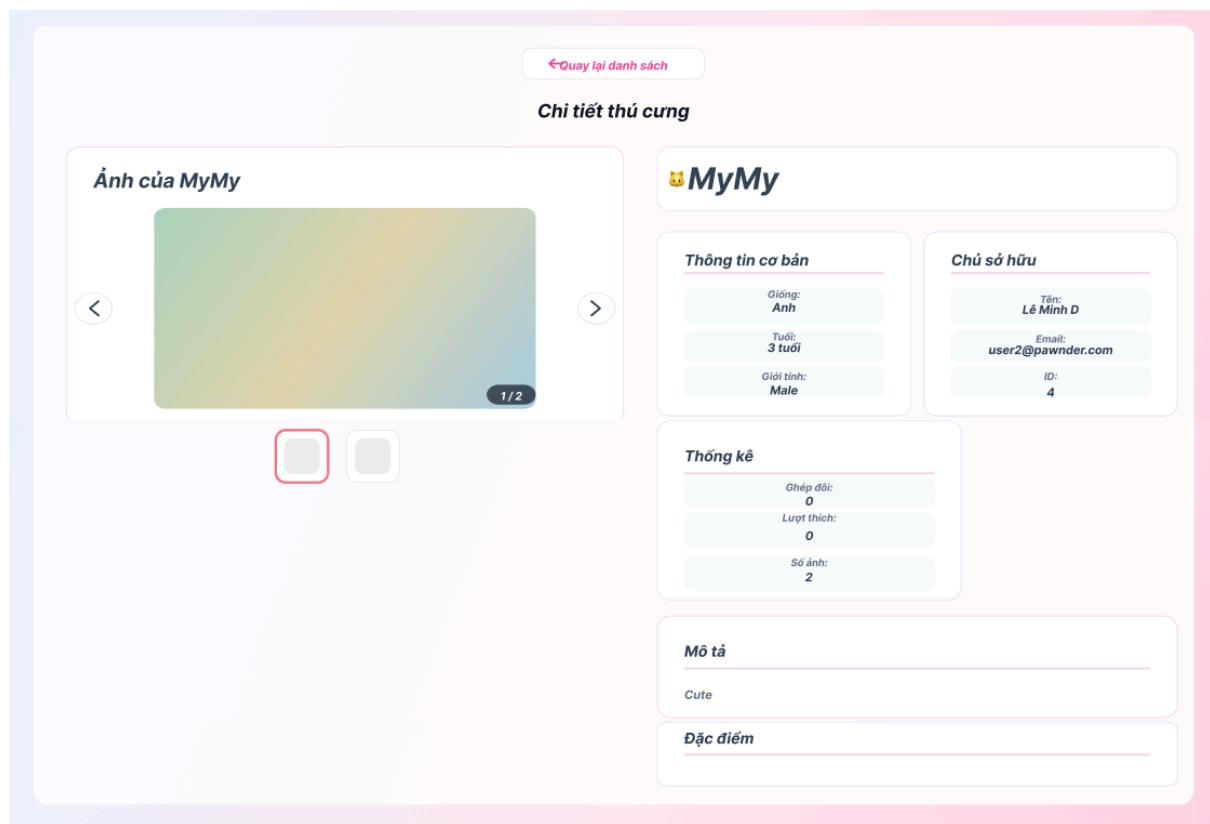
The screenshot shows the Pawnder Admin application's Pet Management screen. The title is "Quản lý thú cưng". It lists five pets: Mila (Golden Retriever, 3 years old), Luna (Poodle, 2 years old), Frank (Vietnam, Male, 0 years old), Lucas (Egypt, Male, 0 years old), and Lyn (Russia, Male, 0 years old). Each pet card includes its name, breed, age, and a link to its profile. The interface features a search bar at the top, a sidebar with various administrative links, and a navigation bar at the bottom.

Field Name	Description
Search Bar	Allows searching pets by name, breed, or owner
Pet Image	Avatar image of the pet

Pet Name	Name of the pet
Breed	Pet breed
Gender	Gender of the pet
Age	Age of the pet
Description	Short description of the pet
Owner Name	Name of the pet owner
Owner Email	Email address of the pet owner
Owner ID	Unique identifier of the owner
Action	Button to view detailed pet information

UC-102: View Pet Detail

Primary Actors	Admin
Secondary Actors	System
Description	Allows Admin to view detailed information of a selected pet, including images, basic information, owner details, statistics, description, and physical characteristics.
Preconditions	Admin has successfully logged into the system The pet already exists in the system
Postconditions	Pet detail information is displayed correctly No data is modified
Normal Sequence / Flow	<ol style="list-style-type: none"> 1. Admin clicks Pet Management from the menu 2. System displays the pet list screen 3. Admin clicks the View Detail (eye icon) of a pet 4. System displays the Pet Detail screen with full information
Alternative Sequences / Flows	A1: Pet does not exist → System displays an error message A2: Admin clicks Back to List → System returns to the pet list screen



Field Name	Description
Image Gallery	Displays all images of the selected pet
Image Navigation	Allows Admin to navigate between images
Pet Name	Name of the pet
Breed	Breed of the pet
Age	Age of the pet
Gender	Gender of the pet
Owner Name	Name of the pet owner
Owner ID	Unique identifier of the pet owner
Matches Count	Number of matches the pet has participated in
Likes Count	Number of likes received by the pet
Total Images	Total number of images uploaded for the pet
Description	Short description of the pet's personality or behavior
Height	Height of the pet (cm)
Weight	Weight of the pet (kg)

3.12 Expert Services

3.12.1 View Consultation Request List

UC-053: View Consultation Request List

Primary Actors	Expert	Secondary Actors	None
Description	Allows experts to view list of AI consultation verification requests from users.		
Preconditions	Expert is logged into the system.		
Postconditions	Expert can view and filter consultation requests by status		
Normal Sequence/Flow	1. Expert navigates to the Notifications Screen from the sidebar. 2. System fetches consultation requests from the server. 3. System displays statistics: Total requests, Pending, Confirmed. 4. System displays a request table with columns: #, User name/email, User question, Created date, Status, Actions. 5. Expert can search by User name. 6. Expert can toggle between "Pending" and "All Processed" views.		
Alternative Sequences/Flows	A1: View Processed Requests At Step 4, Expert clicks "View All Processed": 1. System filters to show only confirmed requests. 2. The table displays an additional "Content" column with expert notes. Use case continues. A2: Refresh Data At Step 4, Expert clicks "Refresh": 1. System reloads requests from the server. Use case continues. A3: Search by User At Step 5, Expert enters a search term: 1. System filters requests by User name. Use case continues.		

Pawnder Admin

Thông báo chờ xác nhận

Chat với người dùng

Tìm kiếm...

Trần Thị B
Người dùng

Quản lý thông báo

Xác nhận và xử lý các yêu cầu xác thực thông tin AI từ người dùng.

Xem tất cả đã xử lý | Làm mới

TỔNG SỐ THÔNG BÁO: 6 | CHỜ XỬ LÝ: 1 | ĐÃ XÁC NHẬN: 5

#	Người dùng	Loại	Tiêu đề	Nội dung	Ngày tạo	Trạng thái	Thao tác
1	Lê Minh C	Xác nhận thông tin AI	Chat #1	Người dùng cần xác nhận chuyên giá cho câu trả lời...	17:31 25/11/2025	CHỜ XỬ LÝ	Xác nhận

Thông báo chờ xác nhận

Chat với người dùng

Quản lý thông báo

Xác nhận và xử lý các yêu cầu xác thực thông tin AI từ người dùng.

Xem chờ xử lý

Làm mới

TỔNG SỐ THÔNG BÁO

6

CHỜ XỬ LÝ

1

ĐÃ XÁC NHẬN

5

#	Người dùng	Loại	Tiêu đề	Nội dung	Ngày tạo	Trạng thái	Thao tác
1	Lê Minh D	Xác nhận thông tin AI	Chat #2	Chuyên gia đã kiểm tra	17:31 25/11/2025	ĐÃ XÁC NHẬN	
2	Le Tuan	Xác nhận thông tin AI	Chat #6	sai hết rồi	08:14 27/11/2025	ĐÃ XÁC NHẬN	
3	Iqt	Xác nhận thông tin AI	Chat #9	Sai quá sai	08:26 27/11/2025	ĐÃ XÁC NHẬN	
4	Iqt	Xác nhận thông tin AI	Chat #10	hay đây	08:29 27/11/2025	ĐÃ XÁC NHẬN	
5	Iqt	Xác nhận thông tin AI	Chat #11	Thêm tí cay cay rất ngon bạn nhé!	13:51 28/11/2025	ĐÃ XÁC NHẬN	

Xác nhận thông báo

X

Thông tin thông báo

Người dùng: Lê Minh C (user1@example.com)

Loại: Xác nhận thông tin AI

Ngày tạo: 17:31 25/11/2025

Trạng thái:

CHỜ XỬ LÝ

File đoạn chat



chat_1.txt

40 tin nhắn - Nhấn để xem

Xem file

Đánh giá và ghi chú từ chuyên gia

Hãy đánh giá tính chính xác của câu trả lời và bổ sung thông tin hữu ích cho người dùng.

Ví dụ: "Thông tin AI đúng nhưng cần bổ sung..." hoặc "Thông tin AI cần điều chỉnh..."

Hủy

Xác nhận và gửi

Field Description

Field Name	Description
Page Title	The main heading displayed at the top - "Quản lý thông báo" (Notification Management)
Page Subtitle	Descriptive text below the title - "Xác nhận và xử lý các yêu cầu xác thực thông tin AI từ người dùng." (Confirm and process AI information verification requests from users.)
Search Input	Text input field with placeholder "Tìm kiếm theo tên người dùng..." (Search by user name...) - filters notifications by user name in real-time
Search Clear Button	X icon button that appears when search term is entered - clears the search input when clicked
View All Processed Button	Gradient button labeled "Xem tất cả đã xử lý" (View all processed) - toggles between viewing pending and processed notifications. Changes to "Xem chờ xử lý" (View pending) when showing processed
Refresh Button	Secondary button labeled "Làm mới" (Refresh) - reloads notifications from backend
Loading State	Message "Đang tải dữ liệu..." (Loading data...) displayed while fetching notifications
Error Alert	Red alert banner displaying error message when data fetching fails
Total Notifications Card	Summary card displaying total number of notifications, labeled "Tổng số thông báo" (Total Notifications)
Pending Notifications Card	Summary card displaying count of notifications with "pending" status, labeled "Chờ xử lý" (Pending)
Confirmed Notifications Card	Summary card displaying count of confirmed notifications, labeled "Đã xác nhận" (Confirmed)
# Column	Table column displaying sequential number for each notification (1, 2, 3...)
User Column	Table column labeled "Người dùng" (User) displaying user name and email
User Name	The full name of the user who requested verification
User Email	The email address of the user
User Question Column	Table column labeled "Câu hỏi người dùng" (User Question) displaying the user's question or request message

Content Column	Table column labeled "Nội dung" (Content) displaying expert's note for confirmed notifications, or "-" for pending ones
Created Date Column	Table column labeled "Ngày tạo" (Created Date) displaying when the notification was created, formatted as HH:mm DD/MM/YYYY
Status Column	Table column labeled "Trạng thái" (Status) displaying status badge
Status Badge	Colored badge showing status: "Chờ xử lý" (Pending - yellow), "Đã xác nhận" (Confirmed - green), "Đã từ chối" (Rejected - red)
Action Column	Table column labeled "Thao tác" (Action) containing action buttons
Confirm Button	Green button labeled "Xác nhận" (Confirm) - displayed for pending notifications, opens confirmation modal
View Button	Blue button with eye icon - displayed for confirmed/rejected notifications, opens detail modal
Empty State Message	Message displayed in table: "Không có thông báo nào chờ xử lý" (No pending notifications) or "Không có thông báo nào đã xử lý" (No processed notifications) based on filter
Pagination Controls	Navigation controls: Previous button with arrow icon, numbered page buttons, Next button with arrow icon - displays when total pages > 1
Confirmation Modal	Modal dialog that opens when clicking Confirm or View button
Modal Title	Heading "Xác nhận thông báo" (Confirm notification) in the modal header
Modal Close Button	X button in the modal header to dismiss the modal
Notification Information Section	Section in modal displaying notification details
User Information	Detail row showing "Người dùng:" (User:) with user name and email
Created Date	Detail row showing "Ngày tạo:" (Created Date:) with formatted date
Status Display	Detail row showing "Trạng thái:" (Status:) with status badge
User Question Section	Section in modal displaying the user's question
Question Content	Detail row showing "Nội dung câu hỏi:" (Question content:) with the user's question text

Chat History Section	Section in modal displaying chat history file card (if available)
Chat File Card	Clickable card displaying chat file icon, filename (chat_[chatAild].txt), message count, and "Xem file" (View file) / "Đang mở" (Open) action
Chat History Panel	Right panel in modal displaying full chat history when chat file is opened
Chat History Title	Heading "Lịch sử chat" (Chat History) in the right panel
Chat Messages	List of chat messages showing sender name, timestamp, and message content
User Message	Chat message from user, displayed with user's name and timestamp
AI Message	Chat message from Pawnder AI, displayed with "Pawnder AI" sender and timestamp
Expert Note Section (Pending)	Section for pending notifications with textarea for expert to enter evaluation and notes
Note Instruction	Helper text: "Hãy đánh giá tính chính xác của câu trả lời và bổ sung thông tin hữu ích cho người dùng." (Please evaluate the accuracy of the answer and add useful information for the user.)
Note Textarea	Multi-line text input with placeholder "Ví dụ: 'Thông tin AI đúng nhưng cần bổ sung...' hoặc 'Thông tin AI cần điều chỉnh...'" (E.g.: 'AI information is correct but needs to add...' or 'AI information needs adjustment...')
Expert Note Display (Confirmed)	Section for confirmed notifications displaying the note that was sent to the user
Note Display	Read-only text showing the expert's note that was sent to the user
Modal Footer Buttons (Pending)	Footer buttons for pending notifications: "Hủy" (Cancel) and "Xác nhận và gửi" (Confirm and send)
Confirm and Send Button	Primary button that submits the confirmation - disabled when note is empty or submitting. Shows "Đang gửi..." (Sending...) while processing
Cancel Button	Secondary button to close the modal without saving

3.12.1 Confirm Consultation Request

UC-053: Confirm Consultation Request

Primary Actors	Expert	Secondary Actors	None
Description	Allows experts to confirm and respond to AI consultation verification requests.		

Preconditions	Expert is logged into the system. Consultation request exists with "pending" status.
Postconditions	Request status changes to "confirmed". User receives notification with expert feedback.
Normal Sequence/Flow	<ol style="list-style-type: none"> Expert clicks the "Confirm" button on a pending request. System displays a confirmation modal with: User info, User question, and AI chat history. Expert reviews the AI conversation. Expert enters a confirmation note or feedback. Expert clicks "Confirm". System sends the confirmation to the server. System updates the request status to "confirmed". System sends a notification to the User. System displays a success message. System switches the view to "All Processed".
Alternative Sequences/Flows	<p>A1: View Chat History At Step 3, Expert clicks "View Chat History": 1. System expands to show the full AI conversation. Use case continues.</p> <p>A2: Chat with AI At Step 3, Expert clicks "Chat with AI": 1. System clones the AI chat. 2. System navigates to the Expert Chat AI Screen. Use case ends.</p> <p>A3: Close Modal At Step 4, Expert clicks close: 1. System closes the modal without saving. Use case ends.</p>

Pawnder Admin

Thống báo chờ xác nhận

Chat với người dùng

Tìm kiếm...

Trần Thị B
Người dùng

Quản lý thông báo

Xác nhận và xử lý các yêu cầu xác thực thông tin AI từ người dùng.

Xem tất cả đã xử lý | Làm mới

TỔNG SỐ THÔNG BÁO: 6 | CHỜ XỬ LÝ: 1 | ĐÃ XÁC NHẬN: 5

#	Người dùng	Loại	Tiêu đề	Nội dung	Ngày tạo	Trạng thái	Thao tác
1	Lê Minh C	Xác nhận thông tin AI	Chat #1	Người dùng cần xác nhận chuyên gia cho câu trả lời...	17:31 25/11/2025	Chờ xác nhận	Xác nhận

Pawnder Admin

Tim kiếm...

Trần Thị B
Người dùng

Thông báo chờ xác nhận

Chat với người dùng

Quản lý thông báo

Xác nhận và xử lý các yêu cầu xác thực thông tin AI từ người dùng.

Xem chờ xử lý | Làm mới

TỔNG SỐ THÔNG BÁO	CHỜ XỬ LÝ	ĐÃ XÁC NHẬN
6	1	5

#	Người dùng	Loại	Tiêu đề	Nội dung	Ngày tạo	Trạng thái	Thao tác
1	Lê Minh D	Xác nhận thông tin AI	Chat #2	Chuyên gia đã kiểm tra	17:31 25/11/2025	ĐÃ XÁC NHẬN	
2	Le Tuan	Xác nhận thông tin AI	Chat #6	sai hết rồi	08:14 27/11/2025	ĐÃ XÁC NHẬN	
3	lqt	Xác nhận thông tin AI	Chat #9	Sai quá sai	08:26 27/11/2025	ĐÃ XÁC NHẬN	
4	lqt	Xác nhận thông tin AI	Chat #10	hay đấy	08:29 27/11/2025	ĐÃ XÁC NHẬN	
5	lqt	Xác nhận thông tin AI	Chat #11	Thêm tí cay cay rất ngọt bạn nhé!	13:51 28/11/2025	ĐÃ XÁC NHẬN	

Xác nhận thông báo

Thông tin thông báo

Người dùng: Lê Minh C (user1@example.com)

Loại: Xác nhận thông tin AI

Ngày tạo: 17:31 25/11/2025

Trạng thái: **CHỜ XỬ LÝ**

File đoạn chat

chat_1.txt
40 tin nhắn · Nhấn để xem | Xem file

Đánh giá và ghi chú từ chuyên gia

Hãy đánh giá tinh chính xác của câu trả lời và bổ sung thông tin hữu ích cho người dùng.

Ví dụ: "Thông tin AI đúng nhưng cần bổ sung..." hoặc "Thông tin AI cần điều chỉnh..."

Hủy | **Xác nhận và gửi**

Field Description

Field Name	Description
Page Title	The main heading displayed at the top - "Quản lý thông báo" (Notification Management)

Page Subtitle	Descriptive text below the title - "Xác nhận và xử lý các yêu cầu xác thực thông tin AI từ người dùng." (Confirm and process AI information verification requests from users.)
Search Input	Text input field with placeholder "Tìm kiếm theo tên người dùng..." (Search by user name...) - filters notifications by user name in real-time
Search Clear Button	X icon button that appears when search term is entered - clears the search input when clicked
View All Processed Button	Gradient button labeled "Xem tất cả đã xử lý" (View all processed) - toggles between viewing pending and processed notifications. Changes to "Xem chờ xử lý" (View pending) when showing processed
Refresh Button	Secondary button labeled "Làm mới" (Refresh) - reloads notifications from backend
Loading State	Message "Đang tải dữ liệu..." (Loading data...) displayed while fetching notifications
Error Alert	Red alert banner displaying error message when data fetching fails
Total Notifications Card	Summary card displaying total number of notifications, labeled "Tổng số thông báo" (Total Notifications)
Pending Notifications Card	Summary card displaying count of notifications with "pending" status, labeled "Chờ xử lý" (Pending)
Confirmed Notifications Card	Summary card displaying count of confirmed notifications, labeled "Đã xác nhận" (Confirmed)
# Column	Table column displaying sequential number for each notification (1, 2, 3...)
User Column	Table column labeled "Người dùng" (User) displaying user name and email
User Name	The full name of the user who requested verification
User Email	The email address of the user
User Question Column	Table column labeled "Câu hỏi người dùng" (User Question) displaying the user's question or request message
Content Column	Table column labeled "Nội dung" (Content) displaying expert's note for confirmed notifications, or "-" for pending ones
Created Date Column	Table column labeled "Ngày tạo" (Created Date) displaying when the notification was created, formatted as HH:mm DD/MM/YYYY

Status Column	Table column labeled "Trạng thái" (Status) displaying status badge
Status Badge	Colored badge showing status: "Chờ xử lý" (Pending - yellow), "Đã xác nhận" (Confirmed - green), "Đã từ chối" (Rejected - red)
Action Column	Table column labeled "Thao tác" (Action) containing action buttons
Confirm Button	Green button labeled "Xác nhận" (Confirm) - displayed for pending notifications, opens confirmation modal
View Button	Blue button with eye icon - displayed for confirmed/rejected notifications, opens detail modal
Empty State Message	Message displayed in table: "Không có thông báo nào chờ xử lý" (No pending notifications) or "Không có thông báo nào đã xử lý" (No processed notifications) based on filter
Pagination Controls	Navigation controls: Previous button with arrow icon, numbered page buttons, Next button with arrow icon - displays when total pages > 1
Confirmation Modal	Modal dialog that opens when clicking Confirm or View button
Modal Title	Heading "Xác nhận thông báo" (Confirm notification) in the modal header
Modal Close Button	X button in the modal header to dismiss the modal
Notification Information Section	Section in modal displaying notification details
User Information	Detail row showing "Người dùng:" (User:) with user name and email
Created Date	Detail row showing "Ngày tạo:" (Created Date:) with formatted date
Status Display	Detail row showing "Trạng thái:" (Status:) with status badge
User Question Section	Section in modal displaying the user's question
Question Content	Detail row showing "Nội dung câu hỏi:" (Question content:) with the user's question text
Chat History Section	Section in modal displaying chat history file card (if available)
Chat File Card	Clickable card displaying chat file icon, filename (chat_[chatAId].txt), message count, and "Xem file" (View file) / "Đang mở" (Open) action

Chat History Panel	Right panel in modal displaying full chat history when chat file is opened
Chat History Title	Heading "Lịch sử chat" (Chat History) in the right panel
Chat Messages	List of chat messages showing sender name, timestamp, and message content
User Message	Chat message from user, displayed with user's name and timestamp
AI Message	Chat message from Pawnder AI, displayed with "Pawnder AI" sender and timestamp
Expert Note Section (Pending)	Section for pending notifications with textarea for expert to enter evaluation and notes
Note Instruction	Helper text: "Hãy đánh giá tính chính xác của câu trả lời và bổ sung thông tin hữu ích cho người dùng." (Please evaluate the accuracy of the answer and add useful information for the user.)
Note Textarea	Multi-line text input with placeholder "Ví dụ: 'Thông tin AI đúng nhưng cần bổ sung...' hoặc 'Thông tin AI cần điều chỉnh...'" (E.g.: 'AI information is correct but needs to add...' or 'AI information needs adjustment...')
Expert Note Display (Confirmed)	Section for confirmed notifications displaying the note that was sent to the user
Note Display	Read-only text showing the expert's note that was sent to the user
Modal Footer Buttons (Pending)	Footer buttons for pending notifications: "Hủy" (Cancel) and "Xác nhận và gửi" (Confirm and send)
Confirm and Send Button	Primary button that submits the confirmation - disabled when note is empty or submitting. Shows "Đang gửi..." (Sending...) while processing
Cancel Button	Secondary button to close the modal without saving

3.12.1 View AI Consultation Conversation

UC-054: View AI Consultation Conversation

Primary Actors	Expert	Secondary Actors	None
Description	Allows experts to view the AI conversation history that user submitted for verification.		
Preconditions	Expert is logged into the system. Chat conversation exists with user. SignalR connection is established.		
Postconditions	Messages are exchanged between expert and user in real-time.		
Normal Sequence/Flow	1. Expert opens a consultation request detail. 2. System fetches the AI chat history from the server. 3. System displays the conversation with: User messages (questions), AI responses (answers), and timestamps for each message.		

	4. Expert reviews the conversation to verify AI accuracy.
Alternative Sequences/Flo	A1: Empty History At Step 2, if no chat history exists: 1. System displays a fallback message. Use case ends.

Pawnder Admin

Tìm kiếm...

Trần Thị B
Người dùng

Thông báo chờ xác nhận

Chat với người dùng

Quản lý thông báo

Xác nhận và xử lý các yêu cầu xác thực thông tin AI từ người dùng.

Xem tất cả đã xử lý

Làm mới

TỔNG SỐ THÔNG BÁO

6

CHỜ XỬ LÝ

1

ĐÃ XÁC NHẬN

5

#	Người dùng	Loại	Tiêu đề	Nội dung	Ngày tạo	Trạng thái	Thao tác
1	Lê Minh C	Xác nhận thông tin AI	Chat #1	Người dùng cần xác nhận chuyên gia cho câu trả lời...	17:31 25/11/2025	CHỜ XỬ LÝ	Xác nhận

Pawnder Admin

Tìm kiếm...

Trần Thị B
Người dùng

Thông báo chờ xác nhận

Chat với người dùng

Quản lý thông báo

Xác nhận và xử lý các yêu cầu xác thực thông tin AI từ người dùng.

Xem chờ xử lý

Làm mới

TỔNG SỐ THÔNG BÁO

6

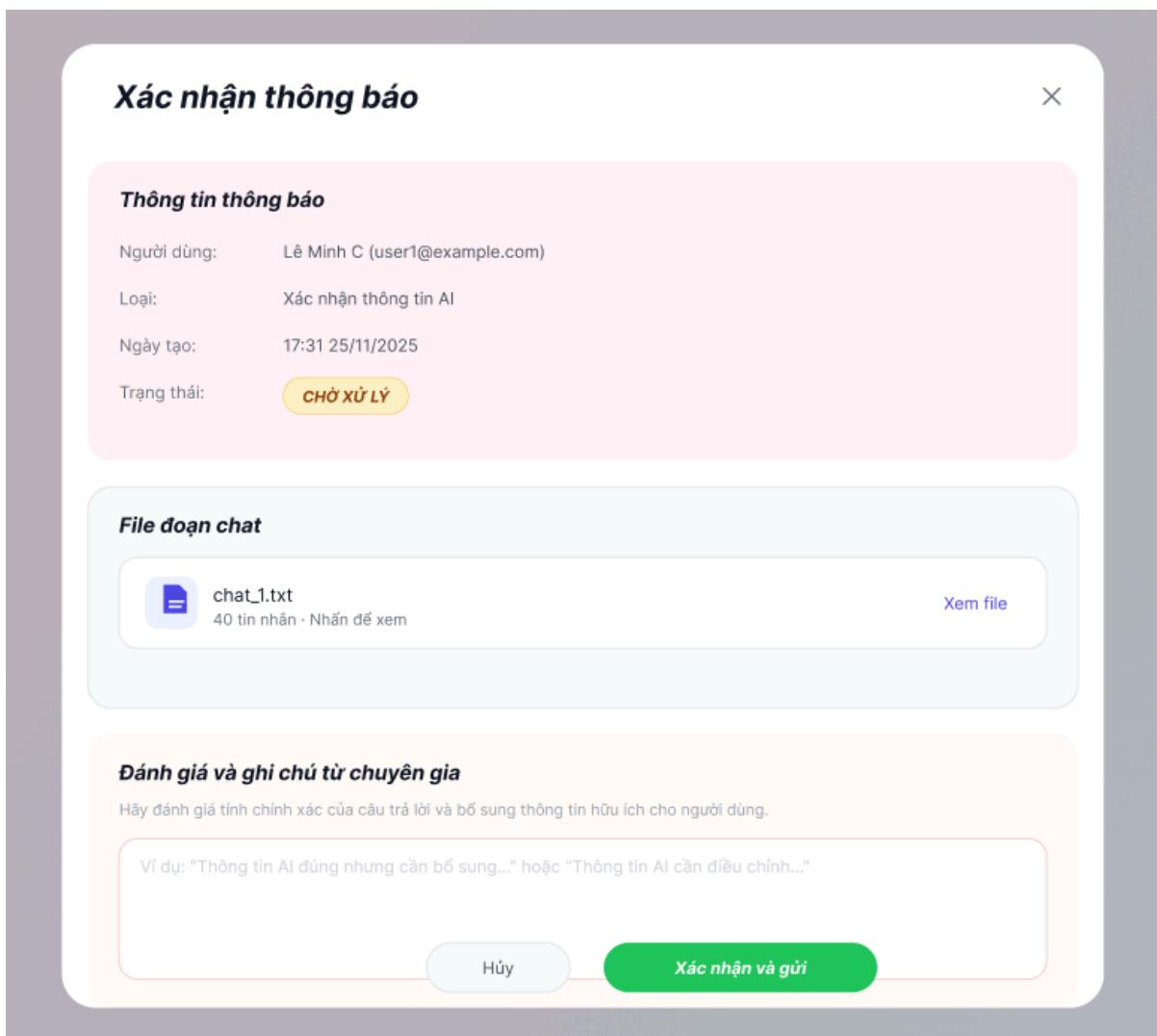
CHỜ XỬ LÝ

1

ĐÃ XÁC NHẬN

5

#	Người dùng	Loại	Tiêu đề	Nội dung	Ngày tạo	Trạng thái	Thao tác
1	Lê Minh D	Xác nhận thông tin AI	Chat #2	Chuyên gia đã kiểm tra	17:31 25/11/2025	ĐÃ XÁC NHẬN	<input checked="" type="radio"/>
2	Le Tuan	Xác nhận thông tin AI	Chat #6	sai hết rồi	08:14 27/11/2025	ĐÃ XÁC NHẬN	<input checked="" type="radio"/>
3	Iqt	Xác nhận thông tin AI	Chat #9	Sai quá sai	08:26 27/11/2025	ĐÃ XÁC NHẬN	<input checked="" type="radio"/>
4	Iqt	Xác nhận thông tin AI	Chat #10	hay đây	08:29 27/11/2025	ĐÃ XÁC NHẬN	<input checked="" type="radio"/>
5	Iqt	Xác nhận thông tin AI	Chat #11	Thêm tí cay cay rất ngon bạn nhé!	13:51 28/11/2025	ĐÃ XÁC NHẬN	<input checked="" type="radio"/>



Field Description

Field Name	Description
Page Title	The main heading displayed at the top - "Quản lý thông báo" (Notification Management)
Page Subtitle	Descriptive text below the title - "Xác nhận và xử lý các yêu cầu xác thực thông tin AI từ người dùng." (Confirm and process AI information verification requests from users.)
Search Input	Text input field with placeholder "Tìm kiếm theo tên người dùng..." (Search by user name...) - filters notifications by user name in real-time
Search Clear Button	X icon button that appears when search term is entered - clears the search input when clicked
View All Processed Button	Gradient button labeled "Xem tất cả đã xử lý" (View all processed) - toggles between viewing pending and processed

	notifications. Changes to "Xem chờ xử lý" (View pending) when showing processed
Refresh Button	Secondary button labeled "Làm mới" (Refresh) - reloads notifications from backend
Loading State	Message "Đang tải dữ liệu..." (Loading data...) displayed while fetching notifications
Error Alert	Red alert banner displaying error message when data fetching fails
Total Notifications Card	Summary card displaying total number of notifications, labeled "Tổng số thông báo" (Total Notifications)
Pending Notifications Card	Summary card displaying count of notifications with "pending" status, labeled "Chờ xử lý" (Pending)
Confirmed Notifications Card	Summary card displaying count of confirmed notifications, labeled "Đã xác nhận" (Confirmed)
# Column	Table column displaying sequential number for each notification (1, 2, 3...)
User Column	Table column labeled "Người dùng" (User) displaying user name and email
User Name	The full name of the user who requested verification
User Email	The email address of the user
User Question Column	Table column labeled "Câu hỏi người dùng" (User Question) displaying the user's question or request message
Content Column	Table column labeled "Nội dung" (Content) displaying expert's note for confirmed notifications, or "-" for pending ones
Created Date Column	Table column labeled "Ngày tạo" (Created Date) displaying when the notification was created, formatted as HH:mm DD/MM/YYYY
Status Column	Table column labeled "Trạng thái" (Status) displaying status badge
Status Badge	Colored badge showing status: "Chờ xử lý" (Pending - yellow), "Đã xác nhận" (Confirmed - green), "Đã từ chối" (Rejected - red)
Action Column	Table column labeled "Thao tác" (Action) containing action buttons
Confirm Button	Green button labeled "Xác nhận" (Confirm) - displayed for pending notifications, opens confirmation modal

View Button	Blue button with eye icon - displayed for confirmed/rejected notifications, opens detail modal
Empty State Message	Message displayed in table: "Không có thông báo nào chờ xử lý" (No pending notifications) or "Không có thông báo nào đã xử lý" (No processed notifications) based on filter
Pagination Controls	Navigation controls: Previous button with arrow icon, numbered page buttons, Next button with arrow icon - displays when total pages > 1
Confirmation Modal	Modal dialog that opens when clicking Confirm or View button
Modal Title	Heading "Xác nhận thông báo" (Confirm notification) in the modal header
Modal Close Button	X button in the modal header to dismiss the modal
Notification Information Section	Section in modal displaying notification details
User Information	Detail row showing "Người dùng:" (User:) with user name and email
Created Date	Detail row showing "Ngày tạo:" (Created Date:) with formatted date
Status Display	Detail row showing "Trạng thái:" (Status:) with status badge
User Question Section	Section in modal displaying the user's question
Question Content	Detail row showing "Nội dung câu hỏi:" (Question content:) with the user's question text
Chat History Section	Section in modal displaying chat history file card (if available)
Chat File Card	Clickable card displaying chat file icon, filename (chat_[chatAild].txt), message count, and "Xem file" (View file) / "Đang mở" (Open) action
Chat History Panel	Right panel in modal displaying full chat history when chat file is opened
Chat History Title	Heading "Lịch sử chat" (Chat History) in the right panel
Chat Messages	List of chat messages showing sender name, timestamp, and message content
User Message	Chat message from user, displayed with user's name and timestamp
AI Message	Chat message from Pawnder AI, displayed with "Pawnder AI" sender and timestamp

Expert Note Section (Pending)	Section for pending notifications with textarea for expert to enter evaluation and notes
Note Instruction	Helper text: "Hãy đánh giá tính chính xác của câu trả lời và bổ sung thông tin hữu ích cho người dùng." (Please evaluate the accuracy of the answer and add useful information for the user.)
Note Textarea	Multi-line text input with placeholder "Ví dụ: 'Thông tin AI đúng nhưng cần bổ sung...' hoặc 'Thông tin AI cần điều chỉnh...'" (E.g.: 'AI information is correct but needs to add...' or 'AI information needs adjustment...')
Expert Note Display (Confirmed)	Section for confirmed notifications displaying the note that was sent to the user
Note Display	Read-only text showing the expert's note that was sent to the user
Modal Footer Buttons (Pending)	Footer buttons for pending notifications: "Hủy" (Cancel) and "Xác nhận và gửi" (Confirm and send)
Confirm and Send Button	Primary button that submits the confirmation - disabled when note is empty or submitting. Shows "Đang gửi..." (Sending...) while processing
Cancel Button	Secondary button to close the modal without saving

3.12.1 Chat with AI (Expert)

UC-055: Chat with AI (Expert)

Primary Actors	Expert	Secondary Actors	None
Description	Allows experts to chat with AI assistant for research or testing purposes.		
Preconditions	Expert is logged into the system.		
Postconditions	Expert can create and manage AI chat conversations.		
Normal Sequence/Flow	1. Expert navigates to the "Chat with AI" Screen from the sidebar. 2. System loads the Expert's AI chat list. 3. System displays the chat list sidebar and the chat window. 4. Expert selects a chat or creates a new chat. 5. Expert types a question in the input field. 6. Expert clicks "Send". 7. System sends the question to the AI. 8. System displays the AI response. 9. The chat is updated and moved to the top of the list.		
Alternative Sequences/Flows	A1: Create New Chat At Step 4, Expert clicks "Create New Chat": 1. System creates a new AI chat. 2. System selects the new chat. 3. The chat window shows an empty state. Use case continues from Step 5. A2: Delete Chat At Step 4, Expert clicks delete on a chat: 1. System shows a confirmation dialog. 2. Expert confirms deletion.		

	<p>3. System deletes the chat. 4. System selects the next available chat. Use case continues. A3: Cloned Chat At Step 2, if navigated from a consultation request: 1. System auto-selects the cloned chat. Use case continues from Step 5.</p>
--	---

Pawnder Admin

Tìm kiếm...

Trần Thị B
Người dùng

Thông báo chờ xác nhận

Chat với người dùng

Quản lý thông báo

Xác nhận và xử lý các yêu cầu xác thực thông tin AI từ người dùng.

Xem tất cả đã xử lý

Làm mới

TỔNG SỐ THÔNG BÁO

6

CHỜ XỬ LÝ

1

ĐÃ XÁC NHẬN

5

#	Người dùng	Loại	Tiêu đề	Nội dung	Ngày tạo	Trạng thái	Thao tác
1	Lê Minh C	Xác nhận thông tin AI	Chat #1	Người dùng cần xác nhận chuyên gia cho câu trả lời...	17:31 25/11/2025	CHỜ XỬ LÝ	Xác nhận

Pawnder Admin

Tìm kiếm...

Trần Thị B
Người dùng

Thông báo chờ xác nhận

Chat với người dùng

Quản lý thông báo

Xác nhận và xử lý các yêu cầu xác thực thông tin AI từ người dùng.

Xem chờ xử lý

Làm mới

TỔNG SỐ THÔNG BÁO

6

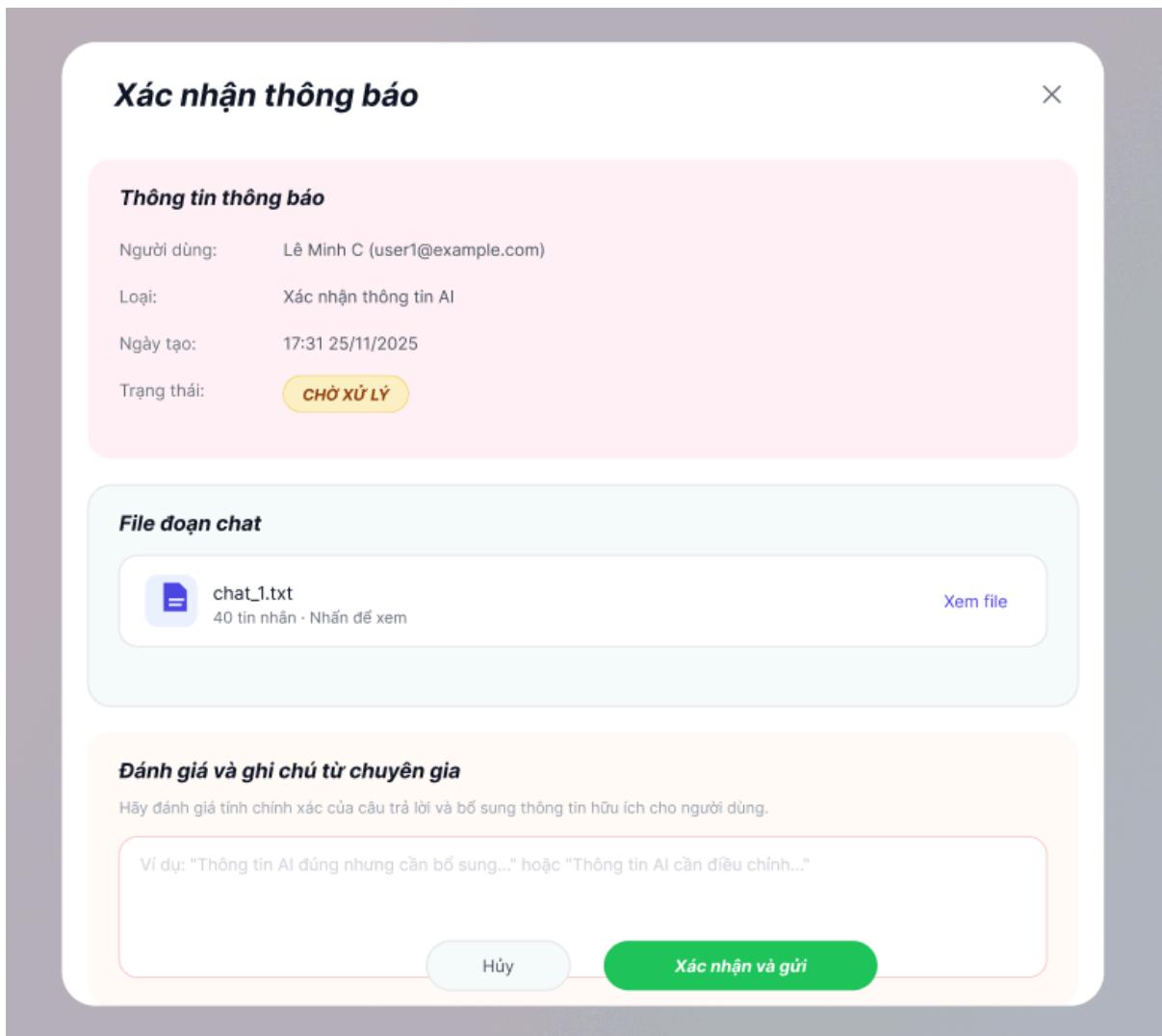
CHỜ XỬ LÝ

1

ĐÃ XÁC NHẬN

5

#	Người dùng	Loại	Tiêu đề	Nội dung	Ngày tạo	Trạng thái	Thao tác
1	Lê Minh D	Xác nhận thông tin AI	Chat #2	Chuyên gia đã kiểm tra	17:31 25/11/2025	ĐÃ XÁC NHẬN	<input checked="" type="radio"/>
2	Le Tuan	Xác nhận thông tin AI	Chat #6	sai hết rồi	08:14 27/11/2025	ĐÃ XÁC NHẬN	<input checked="" type="radio"/>
3	Iqt	Xác nhận thông tin AI	Chat #9	Sai quá sai	08:26 27/11/2025	ĐÃ XÁC NHẬN	<input checked="" type="radio"/>
4	Iqt	Xác nhận thông tin AI	Chat #10	hay đây	08:29 27/11/2025	ĐÃ XÁC NHẬN	<input checked="" type="radio"/>
5	Iqt	Xác nhận thông tin AI	Chat #11	Thêm tí cay cay rất ngon bạn nhé!	13:51 28/11/2025	ĐÃ XÁC NHẬN	<input checked="" type="radio"/>



Field Description

Field Name	Description
Page Title	The main heading displayed at the top - "Quản lý thông báo" (Notification Management)
Page Subtitle	Descriptive text below the title - "Xác nhận và xử lý các yêu cầu xác thực thông tin AI từ người dùng." (Confirm and process AI information verification requests from users.)
Search Input	Text input field with placeholder "Tìm kiếm theo tên người dùng..." (Search by user name...) - filters notifications by user name in real-time
Search Clear Button	X icon button that appears when search term is entered - clears the search input when clicked
View All Processed Button	Gradient button labeled "Xem tất cả đã xử lý" (View all processed) - toggles between viewing pending and processed

	notifications. Changes to "Xem chờ xử lý" (View pending) when showing processed
Refresh Button	Secondary button labeled "Làm mới" (Refresh) - reloads notifications from backend
Loading State	Message "Đang tải dữ liệu..." (Loading data...) displayed while fetching notifications
Error Alert	Red alert banner displaying error message when data fetching fails
Total Notifications Card	Summary card displaying total number of notifications, labeled "Tổng số thông báo" (Total Notifications)
Pending Notifications Card	Summary card displaying count of notifications with "pending" status, labeled "Chờ xử lý" (Pending)
Confirmed Notifications Card	Summary card displaying count of confirmed notifications, labeled "Đã xác nhận" (Confirmed)
# Column	Table column displaying sequential number for each notification (1, 2, 3...)
User Column	Table column labeled "Người dùng" (User) displaying user name and email
User Name	The full name of the user who requested verification
User Email	The email address of the user
User Question Column	Table column labeled "Câu hỏi người dùng" (User Question) displaying the user's question or request message
Content Column	Table column labeled "Nội dung" (Content) displaying expert's note for confirmed notifications, or "-" for pending ones
Created Date Column	Table column labeled "Ngày tạo" (Created Date) displaying when the notification was created, formatted as HH:mm DD/MM/YYYY
Status Column	Table column labeled "Trạng thái" (Status) displaying status badge
Status Badge	Colored badge showing status: "Chờ xử lý" (Pending - yellow), "Đã xác nhận" (Confirmed - green), "Đã từ chối" (Rejected - red)
Action Column	Table column labeled "Thao tác" (Action) containing action buttons
Confirm Button	Green button labeled "Xác nhận" (Confirm) - displayed for pending notifications, opens confirmation modal

View Button	Blue button with eye icon - displayed for confirmed/rejected notifications, opens detail modal
Empty State Message	Message displayed in table: "Không có thông báo nào chờ xử lý" (No pending notifications) or "Không có thông báo nào đã xử lý" (No processed notifications) based on filter
Pagination Controls	Navigation controls: Previous button with arrow icon, numbered page buttons, Next button with arrow icon - displays when total pages > 1
Confirmation Modal	Modal dialog that opens when clicking Confirm or View button
Modal Title	Heading "Xác nhận thông báo" (Confirm notification) in the modal header
Modal Close Button	X button in the modal header to dismiss the modal
Notification Information Section	Section in modal displaying notification details
User Information	Detail row showing "Người dùng:" (User:) with user name and email
Created Date	Detail row showing "Ngày tạo:" (Created Date:) with formatted date
Status Display	Detail row showing "Trạng thái:" (Status:) with status badge
User Question Section	Section in modal displaying the user's question
Question Content	Detail row showing "Nội dung câu hỏi:" (Question content:) with the user's question text
Chat History Section	Section in modal displaying chat history file card (if available)
Chat File Card	Clickable card displaying chat file icon, filename (chat_[chatAild].txt), message count, and "Xem file" (View file) / "Đang mở" (Open) action
Chat History Panel	Right panel in modal displaying full chat history when chat file is opened
Chat History Title	Heading "Lịch sử chat" (Chat History) in the right panel
Chat Messages	List of chat messages showing sender name, timestamp, and message content
User Message	Chat message from user, displayed with user's name and timestamp
AI Message	Chat message from Pawnder AI, displayed with "Pawnder AI" sender and timestamp

Expert Note Section (Pending)	Section for pending notifications with textarea for expert to enter evaluation and notes
Note Instruction	Helper text: "Hãy đánh giá tính chính xác của câu trả lời và bổ sung thông tin hữu ích cho người dùng." (Please evaluate the accuracy of the answer and add useful information for the user.)
Note Textarea	Multi-line text input with placeholder "Ví dụ: 'Thông tin AI đúng nhưng cần bổ sung...' hoặc 'Thông tin AI cần điều chỉnh...'" (E.g.: 'AI information is correct but needs to add...' or 'AI information needs adjustment...')
Expert Note Display (Confirmed)	Section for confirmed notifications displaying the note that was sent to the user
Note Display	Read-only text showing the expert's note that was sent to the user
Modal Footer Buttons (Pending)	Footer buttons for pending notifications: "Hủy" (Cancel) and "Xác nhận và gửi" (Confirm and send)
Confirm and Send Button	Primary button that submits the confirmation - disabled when note is empty or submitting. Shows "Đang gửi..." (Sending...) while processing
Cancel Button	Secondary button to close the modal without saving

3.12.1 View User Chat List

UC-056: View User Chat List

Primary Actors	Expert	Secondary Actors	None
Description	Allows experts to view list of users who have initiated chat conversations.		
Preconditions	Expert is logged into the system.		
Postconditions	Expert can view and select user chats.		
Normal Sequence/Flow	1. Expert navigates to the "Chat with Users" Screen from the sidebar. 2. System establishes a SignalR connection for real-time updates. 3. System fetches the Expert's chat list from the server. 4. System displays the chat list sorted by the most recent message. 5. Each chat item shows: User avatar or initial, User name, and User email. 6. Expert can click on a chat to view messages.		

Alternative Sequences/Flows	<p>A1: No Chats At Step 4, if no chats exist: 1. System displays the message "No conversations yet". Use case ends.</p> <p>A2: New Message Received At any step, if SignalR receives a new message: 1. System updates the chat list order (newest first). 2. If the message belongs to the selected chat, it appears in the chat window. Use case continues.</p>
-----------------------------	--

Pawnder Admin

Tìm kiếm...

Trần Thị B
Người dùng

Thông báo chờ xác nhận

Chat với người dùng

Quản lý thông báo

Xác nhận và xử lý các yêu cầu xác thực thông tin AI từ người dùng.

Xem tất cả đã xử lý

Làm mới

TỔNG SỐ THÔNG BÁO

6

CHỜ XỬ LÝ

1

ĐÃ XÁC NHẬN

5

#	Người dùng	Loại	Tiêu đề	Nội dung	Ngày tạo	Trạng thái	Thao tác
1	Lê Minh C	Xác nhận thông tin AI	Chat #1	Người dùng cần xác nhận chuyên gia cho câu trả lời...	17:31 25/11/2025	CHỜ XỬ LÝ	Xác nhận

Pawnder Admin

Tìm kiếm...

Trần Thị B
Người dùng

Thông báo chờ xác nhận

Chat với người dùng

Quản lý thông báo

Xác nhận và xử lý các yêu cầu xác thực thông tin AI từ người dùng.

Xem chờ xử lý

Làm mới

TỔNG SỐ THÔNG BÁO

6

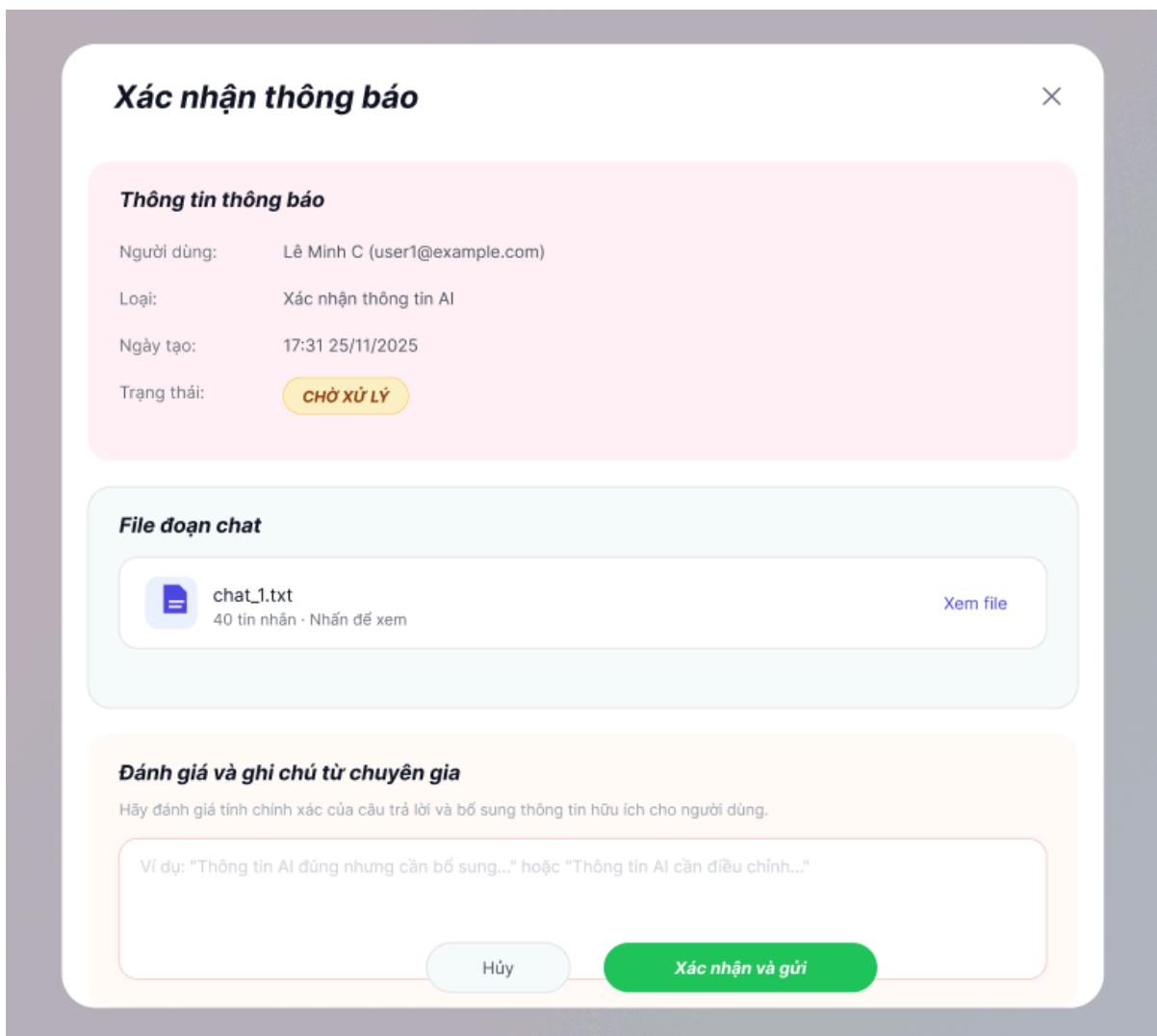
CHỜ XỬ LÝ

1

ĐÃ XÁC NHẬN

5

#	Người dùng	Loại	Tiêu đề	Nội dung	Ngày tạo	Trạng thái	Thao tác
1	Lê Minh D	Xác nhận thông tin AI	Chat #2	Chuyên gia đã kiểm tra	17:31 25/11/2025	ĐÃ XÁC NHẬN	<input checked="" type="radio"/>
2	Le Tuan	Xác nhận thông tin AI	Chat #6	sai hết rồi	08:14 27/11/2025	ĐÃ XÁC NHẬN	<input checked="" type="radio"/>
3	Iqt	Xác nhận thông tin AI	Chat #9	Sai quá sai	08:26 27/11/2025	ĐÃ XÁC NHẬN	<input checked="" type="radio"/>
4	Iqt	Xác nhận thông tin AI	Chat #10	hay đây	08:29 27/11/2025	ĐÃ XÁC NHẬN	<input checked="" type="radio"/>
5	Iqt	Xác nhận thông tin AI	Chat #11	Thêm tí cay cay rất ngon bạn nhé!	13:51 28/11/2025	ĐÃ XÁC NHẬN	<input checked="" type="radio"/>



Field Description

Field Name	Description
Page Title	The main heading displayed at the top - "Quản lý thông báo" (Notification Management)
Page Subtitle	Descriptive text below the title - "Xác nhận và xử lý các yêu cầu xác thực thông tin AI từ người dùng." (Confirm and process AI information verification requests from users.)
Search Input	Text input field with placeholder "Tìm kiếm theo tên người dùng..." (Search by user name...) - filters notifications by user name in real-time
Search Clear Button	X icon button that appears when search term is entered - clears the search input when clicked
View All Processed Button	Gradient button labeled "Xem tất cả đã xử lý" (View all processed) - toggles between viewing pending and processed

	notifications. Changes to "Xem chờ xử lý" (View pending) when showing processed
Refresh Button	Secondary button labeled "Làm mới" (Refresh) - reloads notifications from backend
Loading State	Message "Đang tải dữ liệu..." (Loading data...) displayed while fetching notifications
Error Alert	Red alert banner displaying error message when data fetching fails
Total Notifications Card	Summary card displaying total number of notifications, labeled "Tổng số thông báo" (Total Notifications)
Pending Notifications Card	Summary card displaying count of notifications with "pending" status, labeled "Chờ xử lý" (Pending)
Confirmed Notifications Card	Summary card displaying count of confirmed notifications, labeled "Đã xác nhận" (Confirmed)
# Column	Table column displaying sequential number for each notification (1, 2, 3...)
User Column	Table column labeled "Người dùng" (User) displaying user name and email
User Name	The full name of the user who requested verification
User Email	The email address of the user
User Question Column	Table column labeled "Câu hỏi người dùng" (User Question) displaying the user's question or request message
Content Column	Table column labeled "Nội dung" (Content) displaying expert's note for confirmed notifications, or "-" for pending ones
Created Date Column	Table column labeled "Ngày tạo" (Created Date) displaying when the notification was created, formatted as HH:mm DD/MM/YYYY
Status Column	Table column labeled "Trạng thái" (Status) displaying status badge
Status Badge	Colored badge showing status: "Chờ xử lý" (Pending - yellow), "Đã xác nhận" (Confirmed - green), "Đã từ chối" (Rejected - red)
Action Column	Table column labeled "Thao tác" (Action) containing action buttons
Confirm Button	Green button labeled "Xác nhận" (Confirm) - displayed for pending notifications, opens confirmation modal

View Button	Blue button with eye icon - displayed for confirmed/rejected notifications, opens detail modal
Empty State Message	Message displayed in table: "Không có thông báo nào chờ xử lý" (No pending notifications) or "Không có thông báo nào đã xử lý" (No processed notifications) based on filter
Pagination Controls	Navigation controls: Previous button with arrow icon, numbered page buttons, Next button with arrow icon - displays when total pages > 1
Confirmation Modal	Modal dialog that opens when clicking Confirm or View button
Modal Title	Heading "Xác nhận thông báo" (Confirm notification) in the modal header
Modal Close Button	X button in the modal header to dismiss the modal
Notification Information Section	Section in modal displaying notification details
User Information	Detail row showing "Người dùng:" (User:) with user name and email
Created Date	Detail row showing "Ngày tạo:" (Created Date:) with formatted date
Status Display	Detail row showing "Trạng thái:" (Status:) with status badge
User Question Section	Section in modal displaying the user's question
Question Content	Detail row showing "Nội dung câu hỏi:" (Question content:) with the user's question text
Chat History Section	Section in modal displaying chat history file card (if available)
Chat File Card	Clickable card displaying chat file icon, filename (chat_[chatAild].txt), message count, and "Xem file" (View file) / "Đang mở" (Open) action
Chat History Panel	Right panel in modal displaying full chat history when chat file is opened
Chat History Title	Heading "Lịch sử chat" (Chat History) in the right panel
Chat Messages	List of chat messages showing sender name, timestamp, and message content
User Message	Chat message from user, displayed with user's name and timestamp
AI Message	Chat message from Pawnder AI, displayed with "Pawnder AI" sender and timestamp

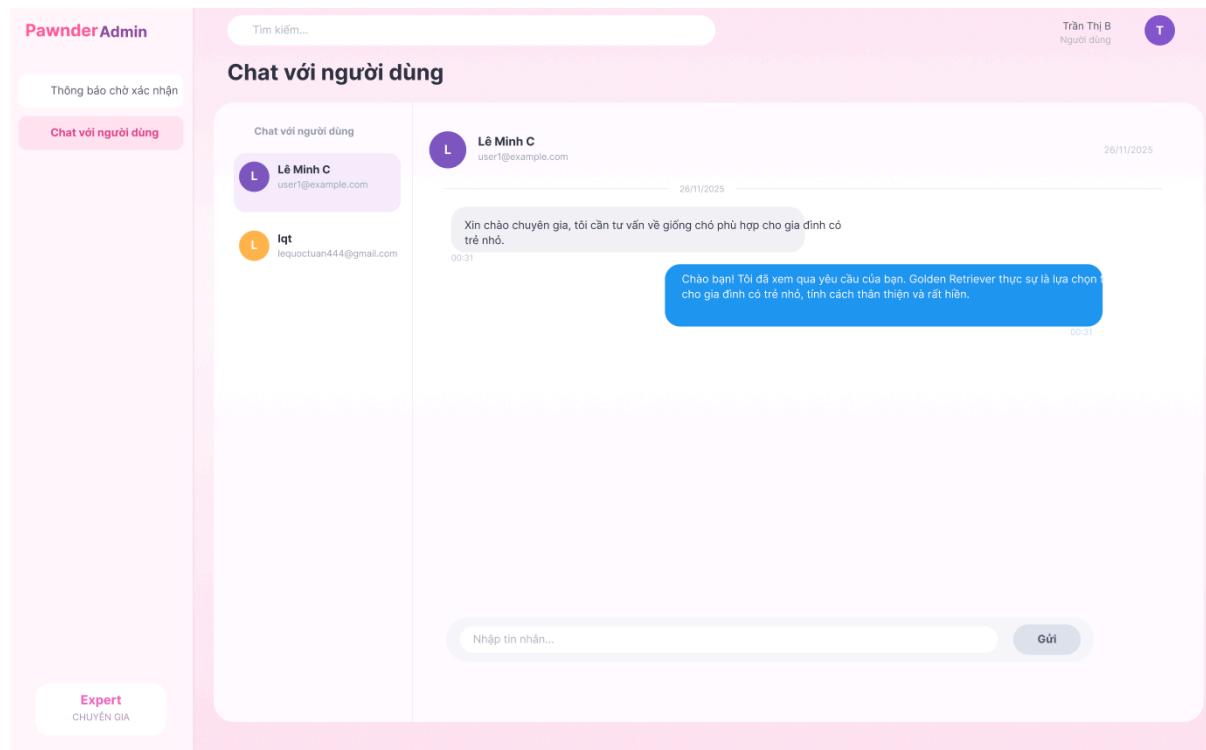
Expert Note Section (Pending)	Section for pending notifications with textarea for expert to enter evaluation and notes
Note Instruction	Helper text: "Hãy đánh giá tính chính xác của câu trả lời và bổ sung thông tin hữu ích cho người dùng." (Please evaluate the accuracy of the answer and add useful information for the user.)
Note Textarea	Multi-line text input with placeholder "Ví dụ: 'Thông tin AI đúng nhưng cần bổ sung...' hoặc 'Thông tin AI cần điều chỉnh...'" (E.g.: 'AI information is correct but needs to add...' or 'AI information needs adjustment...')
Expert Note Display (Confirmed)	Section for confirmed notifications displaying the note that was sent to the user
Note Display	Read-only text showing the expert's note that was sent to the user
Modal Footer Buttons (Pending)	Footer buttons for pending notifications: "Hủy" (Cancel) and "Xác nhận và gửi" (Confirm and send)
Confirm and Send Button	Primary button that submits the confirmation - disabled when note is empty or submitting. Shows "Đang gửi..." (Sending...) while processing
Cancel Button	Secondary button to close the modal without saving

3.12.2 Chat with User

UC-057: Chat with User

Primary Actors	Expert	Secondary Actors	None
Description	Allows experts to send and receive messages with users in real-time.		
Preconditions	Expert is logged into the system. Chat conversation exists with user. SignalR connection is established.		
Postconditions	Messages are exchanged between expert and user in real-time.		
Normal Sequence/Flow	1. Expert selects a user chat from the chat list. 2. System joins the SignalR chat group. 3. System loads chat messages from the server. 4. System displays messages with: sent messages (right side), received messages (left side), timestamps, and date dividers. 5. Expert types a message in the input field. 6. Expert clicks "Send" or presses Enter. 7. System adds the message optimistically (immediate display). 8. System sends the message via the API.		

	<p>9. The message status updates to "sent".</p> <p>10. The User receives the message via SignalR.</p>
Alternative Sequences/Flows	<p>A1: Receive Message</p> <p>At any step, when the User sends a message:</p> <ol style="list-style-type: none"> 1. SignalR receives the message event. 2. System adds the message to the chat window. 3. System scrolls to the bottom. <p>Use case continues.</p> <p>A2: Switch Chat</p> <p>At Step 1, Expert selects a different chat:</p> <ol style="list-style-type: none"> 1. System leaves the current SignalR group. 2. System joins the new chat group. 3. System loads the new chat messages. <p>Use case continues from Step 4.</p>



Field Description

Field Name	Description
Page Title	The heading displayed at the top of the left sidebar - "Chat với người dùng" (Chat with users)
Mock Data Indicator	Optional indicator "Đang dùng Mock Data" (Using Mock Data) displayed when USE_MOCK_DATA is true

Loading State	Message "Đang tải..." (Loading...) with subtitle "Đang tải danh sách chat..." (Loading chat list...) displayed while fetching chats
Chat List Sidebar	Left sidebar displaying list of users available for chat
Empty Chat List Message	Message "Chưa có cuộc trò chuyện nào" (No conversations yet) displayed when no chats exist
Chat Item	Individual user entry in the chat list - clickable to select a chat
Selected Chat Item	Chat item with light blue background indicating it's currently selected
User Avatar	Circular avatar displaying user's profile picture or first letter of user's name as placeholder
User Name (in list)	The full name of the user displayed in each chat item
User Email (in list)	The email address of the user displayed below the name in each chat item
Chat Window	Right section displaying the selected chat conversation
Chat Header	Header section at the top of the chat window showing selected user information
Chat Header Avatar	Circular avatar in the header matching the selected user's avatar from the list
Chat Header Name	The full name of the selected user displayed in the header
Chat Header Email	The email address of the selected user displayed below the name in the header
Chat Messages Area	Scrollable area displaying the conversation messages
Date Divider	Horizontal divider showing date label (e.g., "Hôm nay" (Today), "Hôm qua" (Yesterday), or formatted date) - appears when date changes between messages
Message Bubble	Individual message container displaying message content and timestamp
Sent Message	Message bubble aligned to the right (expert's messages) with distinct styling
Received Message	Message bubble aligned to the left (user's messages) with distinct styling
Message Content	The text content of the message displayed in the message bubble
Message Timestamp	Time displayed below each message, formatted as HH:mm in Vietnamese timezone

Empty Chat State	Message "Chọn một cuộc trò chuyện để bắt đầu" (Select a conversation to start) displayed when no chat is selected
Message Input Form	Form at the bottom of the chat window for sending messages
Message Input Field	Text input field with placeholder "Nhập tin nhắn..." (Enter message...) - disabled while sending
Send Button	Button labeled "Gửi" (Send) - submits the message. Displays "Đang gửi..." (Sending...) while processing. Disabled when input is empty or sending
SignalR Connection	Real-time WebSocket connection for receiving new messages instantly - automatically connects when component mounts
Auto-scroll	Automatic scrolling to bottom of messages when new messages arrive or chat is selected

4. Non-Functional Requirements

4.1 External Interfaces

4.1.1 User Interfaces

- **UI-01:** The application shall provide a modern, clean, and intuitive interface optimized for both mobile (User App - React Native) and web (Admin & Expert Portal - React.js).
- **UI-02:** All user-facing screens shall maintain consistency in typography, color scheme, button styles, and layout structure.
- **UI-03:** The system shall support Vietnamese (vi-VN) as the primary language for all screens.

4.1.2 Software Interfaces

- **SI-01:** The system shall connect to a PostgreSQL database hosted on Render for all persistent data storage.
- **SI-02:** The system shall integrate with Gmail OAuth2 API for sending verification emails, OTP, and notifications.
- **SI-03:** The system shall integrate with VietQR/SePay API for payment processing and automated transaction verification.
- **SI-04:** The system shall use Cloudinary API for image storage, optimization, and retrieval.
- **SI-05:** The system shall integrate with Google Gemini AI API to perform pet image analysis (breed, characteristics, traits) and provide conversational responses for the in-app AI Chat feature.
- **SI-06:** The system shall integrate with LocationIQ API for geocoding (converting GPS coordinates to address).
- **SI-07:** The system shall integrate with Kickbox API for email validation.

4.1.3 Communications Interfaces

- **CI-01:** All data communication must use HTTPS protocol to ensure end-to-end encryption.
- **CI-02:** SignalR (WebSocket) shall be used for real-time features such as chat messaging, typing indicators, match notifications, and badge updates.

4.2 Quality Attributes

4.2.1 Usability

- **USA-01:** All navigation flows shall require no more than 3 steps to reach a primary action.
- **USA-02:** Forms must include inline validation and user-friendly error messages.

- **USA-03:** Icons, labels, and micro-interactions must comply with Mobile UX Guidelines.

4.2.2 Reliability

- **REL-01:** The system shall achieve 99% uptime (excluding scheduled maintenance).
- **REL-02:** Daily database backups must be automatically created and stored for at least 7 days.
- **REL-03:** The system shall implement retry logic for failed external API calls.

4.2.3 Performance

- **PER-01:** Average API response time shall be ≤ 5 seconds under normal load.
- **PER-02:** The system must support 200 concurrent users and scale to 1000 under peak load.
- **PER-03:** Image upload must complete within 10 seconds for files <10MB.

4.2.4 Security

- **SEC-01:** Passwords must be hashed using BCrypt with salt.
- **SEC-02:** Only Admin users have permission to modify or delete system-level data.
- **SEC-03:** JWT authentication shall be used for all mobile and web clients.

4.2.5 Maintainability

- **MAI-01:** Codebase must follow modular architecture to ensure easy extension.
- **MAI-02:** All API endpoints must be documented using Swagger/OpenAPI.
- **MAI-03:** Source code must follow consistent naming conventions and project structure.

4.2.6 Portability

- **POR-01:** Mobile app shall support Android 10+.
- **POR-02:** Web app must be compatible with Chrome, CocCoc and Safari.
- **POR-03:** Containerization using Docker is recommended to ensure deployment portability.

4.2.7 Support Documents

- **SUD-01:** A comprehensive User Guide must be provided for User App, Admin Portal, and Expert Portal.
- **SUD-02:** Every new release shall include Release Notes documenting features, improvements, and bug fixes.

5. Requirement Appendix

5.1 Business Rules

ID	Rule Definition
BR-01	Users must be at least 18 years old to register and use the application.
BR-02	Each account can own a maximum of 3 pets; creation beyond this is blocked.
BR-03	Each pet can have up to 4 photos; only JPG/PNG ≤10MB; exactly one primary photo.
BR-04	A pet profile must have at least one (1) photo and required basic information (name, breed entered as free text, gender, valid age greater than 0 and within the system-defined limit) before it can be set to Active; setting a pet profile to Active shall automatically set all other pet profiles of the same user to Inactive.
BR-05	Only mutually matched pets can exchange messages; blocked/reported users cannot interact.
BR-06	Pets shown for matching must be Active, not deleted, have ≥1 non-deleted photo, and ≥1 characteristic.

BR-07	Matching/recommendation considers user preferences, pet characteristics, and distance (when available), excludes blocked and pending/accepted matches, and returns up to 20 results.
BR-08	Blocking/Reporting: self-block/report is forbidden; blocking/reporting soft-deletes related chats/matches and hides both users from each other.
BR-09	Only Admin can ban/unban; bans can be temporary or permanent; banned users cannot log in or message; expired bans are lifted on login/refresh.
BR-10	Daily limits: request_match (Free 10/day, VIP 30/day); expert_confirm (Free 2, VIP 10); expert_chat (Free 5, VIP 15); AI chat tokens (Free 10k/day, VIP 50k/day).
BR-12	Expert can only view and process Expert Confirmations assigned to them
BR-13	When an expert confirmation is marked "confirmed", a notification must be sent to the user.
BR-14	User/content reports must be reviewed by Admin; violating chats/matches are locked/removed.
BR-15	VIP purchase: disallow if VIP is already active; addInfo must follow userId{X}months{Y} format; amount > 0; activate only after VietQR/SePay verification.
BR-16	On VIP expiry, downgrade user status to normal; expired payments move to pending status.
BR-17	Payment history must be retained for audit/reconciliation purposes.
BR-18	Chat history for user-user and user-expert is stored; AI chat history is retained when expert verification is requested.
BR-19	AI chat respects daily token quota based on user status (Free/VIP).
BR-20	Address: latitude must be -90 to 90, longitude must be -180 to 180; geocoding rate-limit 1 request/sec/user; each user has a single address record (updated in place).
BR-21	Attributes/options drive classification/filtering/matching; characteristics and preferences must reference valid Attributes; MinValue ≤ MaxValue; numeric ranges: age 0–25 years, weight 0.5–12 kg, height 15–40 cm.
BR-22	Passwords require ≥8 characters with uppercase, lowercase, digit, and special character; always hashed; password change requires the old password and must differ from it.
BR-23	Email must be valid, ≤150 chars, unique; FullName ≤100 chars; Gender in allowed set; text fields respect DB length limits.
BR-24	Upload requires at least 1 photo before saving a new pet; cannot delete the last photo if the pet is Active.
BR-25	Pet image analysis: up to 4 photos per request; AI extracts characteristics; only cat images are analyzed.
BR-26	Platform-role enforcement: Admin/Expert log in via web admin; User via mobile; refresh tokens must match DB.
BR-27	Search distance is configurable (e.g., 0–100 km; null = unlimited); if location data is missing, distance filtering is skipped.
BR-28	Admin can create/update/delete Expert accounts and manage user roles/status.
BR-29	Admin can reassign Expert Confirmation to another Expert (only when status is pending).
BR-30	Admin can manage Attributes and AttributeOptions (create, update, delete).
BR-31	Expert can chat with Users via ChatExpert; can view User's ChatAI history when confirmation is requested.
BR-32	Experts can update ExpertConfirmation status (pending → confirmed/rejected) with a response message.
BR-33	Banned words must be classified into three (03) severity levels: Mild, Severe, and Critical, as the basis for message content handling.
BR-34	For Mild banned words, the system must automatically mask all or part of the violating content with replacement characters and still allow the message to be sent.

BR-35	For Severe or Critical banned words, the system must completely reject the message and notify the user of the violation.
BR-36	If a message contains multiple banned words of different severity levels, the system must apply the highest severity level to determine the handling method.
BR-37	The system must normalize message content (including but not limited to: removing special characters, excess whitespace, case conversion, and detecting substitute character patterns) to prevent any attempts to bypass the banned word filter.
BR-38	The system must support banned word detection through both plain text strings and pattern/regex expressions to ensure accuracy and scalability.
BR-39	The banned word filter mechanism must be applied to all chat content, including but not limited to: User-to-User chat and User-to-Expert chat.
BR-40	Only users with Administrator roles are permitted to create, edit, delete, or classify the banned word list.
BR-41	Only mutually matched users can create appointments with each other.
BR-42	Appointment date/time must be in the future (at least 2 hours from current time).
BR-43	Users cannot create new appointments if a pending/confirmed appointment exists with the same match.
BR-44	Minimum 10 messages required between users before creating an appointment.
BR-45	Both pet profiles must be complete (name, breed, photo) before creating appointment
BR-46	Only the current decision user can accept, decline, or counter-offer an appointment.
BR-47	Maximum 3 counter-offers allowed per appointment.
BR-48	Check-in requires user location within 100 meters of appointment location, and within 30 minutes before to 90 minutes after scheduled time.
BR-49	Appointments auto-expire if not responded before scheduled time; auto-marked NO SHOW if no check-in within 90 minutes; auto-complete if on going for 90 minutes.
BR-50	Only Admin can create, update, or cancel pet events.
BR-51	Event timeline must follow: StartTime < SubmissionDeadline < EndTime; StartTime must be in future.
BR-52	Users can only submit to events after StartTime and before SubmissionDeadline.
BR-53	Each user can submit only one entry per event.
BR-54	Submission pets must belong to the submitting user and not be deleted.
BR-55	Users cannot vote for their own submissions.
BR-56	Each user can vote only once per submission; can unvote before the event ends.
BR-57	Event status transitions automatically: upcoming → active → submission_closed → voting_ended → completed.
BR-58	Notifications sent to all users (Role_Name="user") when an event is created, updated, or cancelled.
BR-59	Only Admin can create, update, delete policies and policy versions.
BR-60	Policy codes and names must be unique within the system; codes are case-insensitive for uniqueness check.
BR-61	Each policy can have only one ACTIVE version; publishing new version sets previous to INACTIVE.
BR-62	Only DRAFT versions can be edited or deleted.
BR-63	Policies with requireConsent=true must be accepted by users to continue using the app.
BR-64	User policy acceptance is tracked with timestamp; old accepts are invalidated when a new version is published.
BR-65	blocks API requests if the user has pending required policies.
BR-66	OTP codes are valid for 5 minutes from creation time; expired or invalid OTP codes are rejected.
BR-67	Access token expires after 60 minutes (1 hour); refresh token expires after 30 days.

BR-68	Bad word list is cached for 30 minutes to optimize performance; cache automatically refreshes after expiration.
BR-69	When uploading the first photo for a pet with no photos, that photo must be automatically set as primary.

5.2 Common Requirements

5.2.1 Authentication & Authorization

- All API endpoints require JWT token authentication except login, registration, and OTP verification.
- Access token expiration time is 6 hours (360 minutes), with refresh token valid for 30 days.
- Role-based access control: Admin, Expert, and User roles with different permission levels.
- Passwords must be at least 8 characters, containing uppercase, lowercase, numbers, and special characters.
- Passwords are hashed using BCrypt; legacy SHA256 hashes are auto-upgraded on login.

5.2.2 Data Validation

- All user inputs must be validated on both client and server side.
- Email addresses must be valid format, unique, and ≤ 150 characters.
- FullName must be ≤100 characters; Gender must be in the allowed set.
- Date fields must be in ISO 8601 format.
- Numeric fields must be validated for range (e.g., latitude -90 to 90, longitude -180 to 180).

5.2.3 Image Handling

- All images are uploaded to Cloudinary cloud storage (folder: pawnder/pets).
- Maximum file size: 10MB per image.
- Supported formats: JPEG, PNG.
- Each pet can have up to 4 photos; exactly one must be marked as primary.
- Pet image analysis uses Google Gemini AI; only cat images are accepted.

5.2.4 Real-time Communication

- SignalR is used for real-time chat messaging between users and experts.
- WebSocket transport with compression enabled for faster transmission.
- KeepAlive interval: 10 seconds; Client timeout: 20 seconds.
- Connection state is monitored with automatic reconnection on failure.
- Supports: chat messages, typing indicators, read receipts, match notifications, and badge updates.

5.2.5 Error Handling

- All errors are logged with timestamp, request path, and error details.
- User-facing error messages are clear and in Vietnamese.
- System errors do not expose sensitive information (database structure, API keys, etc.).
- Global exception handler returns standardized JSON error responses.

5.2.6 Performance Requirements

- Database queries use Entity Framework Core with PostgreSQL.
- Retry on failure enabled (max 3 retries, 5 seconds delay).
- Command timeout: 30 seconds.
- Matching/recommendation returns a maximum of 20 results.
- SignalR message buffer: 32KB maximum.

5.2.7 Security Requirements

- All API communications use HTTPS.
- Passwords are hashed using BCrypt (auto-upgrade from legacy SHA256).
- JWT tokens are validated for issuer, audience, lifetime, and signing key.
- SQL injection prevention through Entity Framework parameterized queries.

- CORS configured for specific origins only.

5.2.8 Accessibility Requirements

- Cloudinary: Image storage and optimization.
- Google Gemini AI: Pet image analysis and AI chat.
- VietQR/SePay: Payment verification for VIP subscription.
- LocationIQ: Geocoding for address resolution (rate-limit 1 req/sec/user).
- Kickbox: Email verification.
- Gmail OAuth2: Email sending for OTP.

5.2.9 Localization

- Application supports Vietnamese (vi-VN) as the primary language.
- All user-facing messages are in Vietnamese.
- Currency is displayed in Vietnamese Dong (VND).
- Date/time follows Vietnamese locale standards.

5.2.10 Data Privacy

- User data is only collected for application functionality.
- Users can request data export and deletion (GDPR compliance).
- Third-party services (Cloudinary, Google Gemini) must comply with data protection agreements.
- Analytics data is anonymized and aggregated.

5.3 Application Messages List

#	Message code	Message Type	Context	Content
1	MSG01	Alert dialog	Please enter the ban reason!	Vui lòng nhập lý do ban!
2	MSG02	Alert dialog	User banned successfully!	Đã ban người dùng thành công!
3	MSG03	Alert dialog	Cannot ban user. Please try again later.	Không thể ban người dùng. Vui lòng thử lại sau.
4	MSG04	Alert dialog	User unbanned successfully!	Đã gỡ ban người dùng thành công!
5	MSG05	Alert dialog	Please enter a note before confirming.	Vui lòng nhập ghi chú trước khi xác nhận.
6	MSG06	Alert dialog	Cannot identify expert. Please login again.	Không thể xác định chuyên gia. Vui lòng đăng nhập lại.
7	MSG07	Alert dialog	Notification confirmed successfully.	Đã xác nhận thông báo thành công.
8	MSG08	Alert dialog	Cannot confirm notification. Please try again.	Không thể xác nhận thông báo. Vui lòng thử lại.
9	MSG09	Alert dialog	Cannot send messages. Please login again.	Không thể gửi tin nhắn. Vui lòng đăng nhập lại.
10	MSG10	Alert dialog	Cannot send messages. Please try again.	Không thể gửi tin nhắn. Vui lòng thử lại.
12	MSG12	Inline loading text	Loading page	Đang tải dữ liệu...
13	MSG13	Inline error state	Cannot generate QR code. Please try again.	Không thể tạo mã QR. Vui lòng thử lại.
14	MSG14	Alert dialog	User info not found. Please login again.	Không tìm thấy thông tin người dùng. Vui lòng đăng nhập lại.

15	MSG15	Alert dialog	Invalid user information.	Thông tin người dùng không hợp lệ.
16	MSG16	Alert dialog	Payment successful! You have upgraded to VIP	Thanh toán thành công! Bạn đã nâng cấp lên VIP
17	MSG17	Alert dialog	Cannot complete payment. Please try again.	Không thể hoàn tất thanh toán. Vui lòng thử lại.
18	MSG18	Alert dialog	Payment error – An error occurred. Please try again later.	Lỗi thanh toán – Đã có lỗi xảy ra. Vui lòng thử lại sau.
19	MSG19	Alert dialog	Failed to Save – Unknown error. Please try again.	Failed to Save – Unknown error. Please try again.
20	MSG20	Alert dialog	Conversation not found.	Không tìm thấy cuộc trò chuyện.
21	MSG21	Alert dialog	Cannot load messages.	Không thể tải tin nhắn.
22	MSG22	Alert dialog	Cannot send messages. Please try again.	Không thể gửi tin nhắn. Vui lòng thử lại.
23	MSG23	Alert dialog	Cannot load chat history.	Không thể tải lịch sử chat.
24	MSG24	Alert dialog	Please create a new conversation first.	Vui lòng tạo cuộc trò chuyện mới trước.
25	MSG25	Alert dialog	AI is overloaded – Processing taking long. Please try again in seconds. AI is overloaded – Processing taking long. Please try again in seconds.	AI đang quá tải – AI đang mất nhiều thời gian để xử lý. Vui lòng thử lại sau vài giây.
26	MSG26	Alert dialog	An error occurred with AI. Please try again.	Có lỗi xảy ra với AI. Vui lòng thử lại.
27	MSG27	Alert dialog	Cannot send messages. Please check the connection.	Không thể gửi tin nhắn. Vui lòng kiểm tra kết nối.
28	MSG28	Custom alert modal	Please enter your question.	Vui lòng nhập câu hỏi của bạn.
29	MSG29	Custom alert modal	Please save conversation before requesting an expert.	Vui lòng lưu cuộc trò chuyện trước khi yêu cầu chuyên gia.
30	MSG30	Custom alert modal	Cannot send request. Please try again.	Không thể gửi yêu cầu. Vui lòng thử lại.
31	MSG31	Limit modal	You have run out of expert confirmation quota for today!	Bạn đã hết lượt xác nhận chuyên gia hôm nay!
32	MSG32	Custom alert modal	Request sent! Your request has been sent to an expert...	Đã gửi yêu cầu! Yêu cầu của bạn đã được gửi đến chuyên gia...
33	MSG33	Alert dialog	Cannot create chat. Please try again later.	Không thể tạo chat. Vui lòng thử lại sau.
34	MSG34	Alert dialog	Cannot delete conversation.	Không thể đổi tên cuộc trò chuyện.
36	MSG36	Alert dialog	AIChatListScreen.tsx	Không thể xóa cuộc trò chuyện.
37	MSG37	Inline validation text	Email is required.	Email là bắt buộc.
38	MSG38	Inline validation text	Email is invalid.	Email không hợp lệ.

39	MSG39	Inline validation text	Password is required.	Mật khẩu là bắt buộc.
40	MSG40	Inline validation text	Passwords must be at least 6 characters.	Mật khẩu phải có ít nhất 6 ký tự.
41	MSG41	Inline error banner	Login failed. Please check the information again.	Đăng nhập thất bại. Vui lòng kiểm tra lại thông tin.
42	MSG42	Alert dialog	User information not found.	Không tìm thấy thông tin người dùng.
43	MSG43	Alert dialog	User information not found.	Không tìm thấy thông tin người dùng.

IV. Software Design Description

1. System Design

1.1 System Architecture

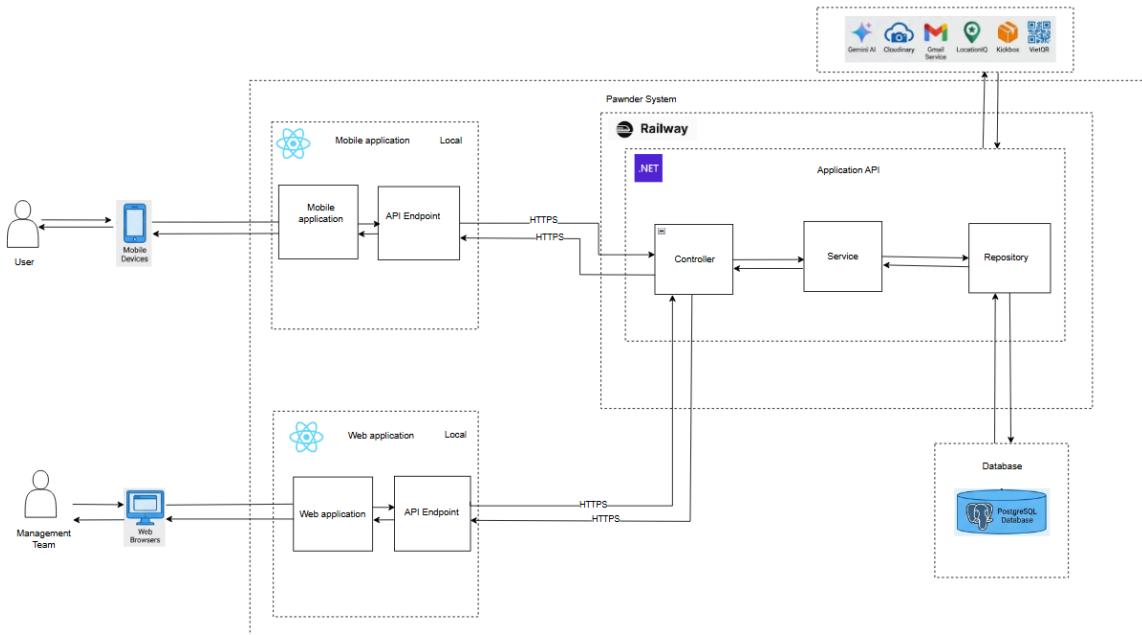


Figure 1.1. System Architecture of the Pawnder system

The Overall Architectural Diagram (Figure 1.1) illustrates the high-level components of the Pawnder system and their interactions. Key components include:

1. Actor:

No	Actor	Role	Interface
1	User	End users who use the app to find and connect pets	Mobile App (React Native)
2	Expert	Experts who verify pet information	Internet Browser (Web)
3	Administrator	System administrators	Internet Browser (Web)

2. Backend Layer:

No	Component	Function
1	Backend API	Handles business logic, JWT authentication, data management controllers
2	SignalR Hub	Handles real-time features: user-to-user chat, expert chat, match/like notifications, typing indicators

3. External service:

No	Service	Function
1	Cloudinary	Pet image storage and management
2	Google Gemini AI	Pet image analysis, auto-generated descriptions, AI chatbot
3	Gmail API	Email notifications, OTP verification
4	VietQR	QR code payment generation for VIP upgrades
5	Kickbox	Email address validation
6	LocationIQ	Reverse geocoding - converts GPS coordinates to readable addresses (city, district, ward)

1.2 Package Diagram

1.2.1 Package Backend

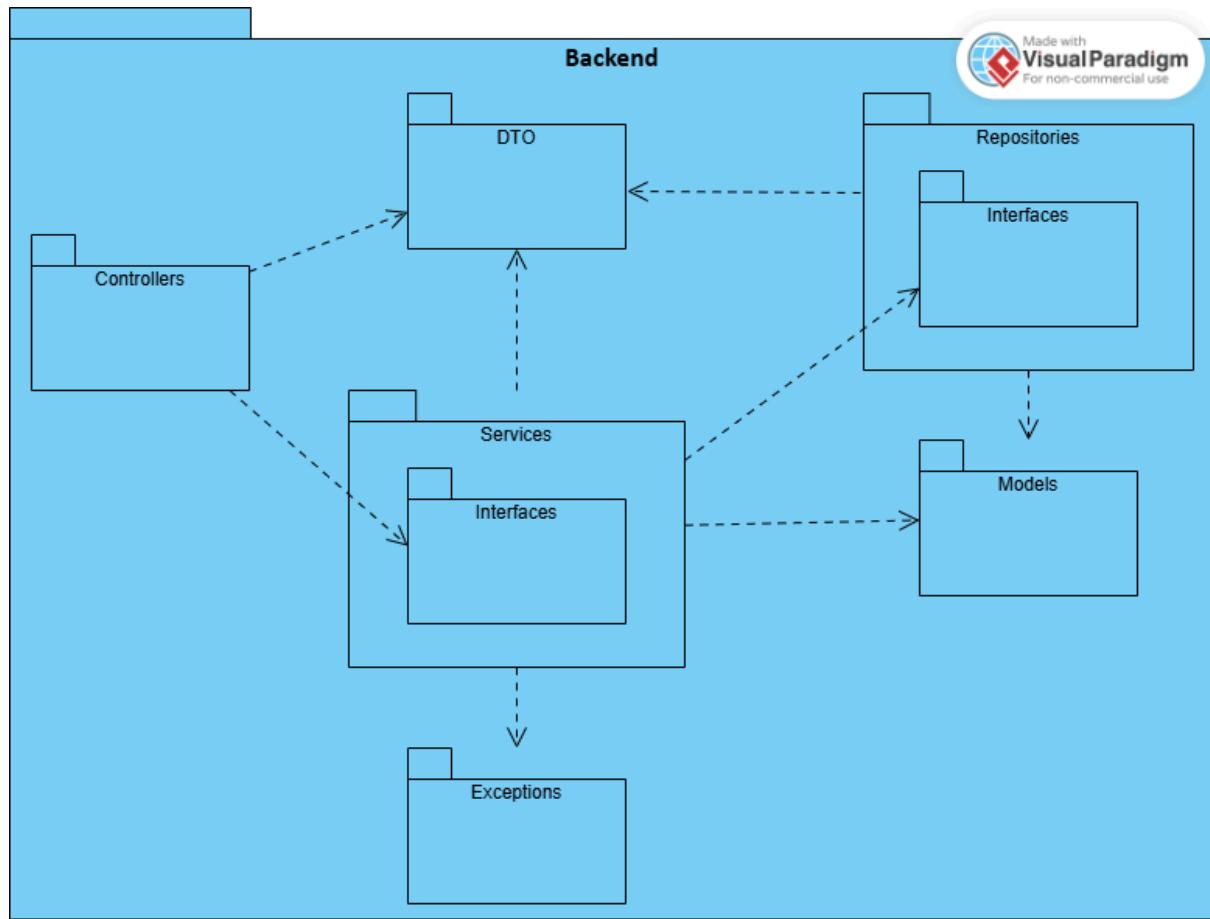


Figure 1.2.1. Backend Package Diagram

Package Descriptions

No	Package	Description
1	Controllers	API Layer - Receives and processes HTTP requests, returns HTTP responses. Contains controller classes that handle application API endpoints.
2	Services	Business Logic Layer - Handles business logic, validation, and orchestration. Contains service classes that implement the main business operations of the application.
3	Services.Interfaces	Service Contracts - Defines interfaces for services. Provides contracts/interfaces to ensure consistency and facilitate testing and mocking.
4	Repositories	Data Access Layer - Queries database, performs CRUD operations. Contains repository classes that interact directly with the database.
5	Repositories.Interfaces	Repository Contracts - Defines interfaces for repositories. Provides contracts/interfaces to ensure consistency and facilitate testing and mocking.

6	Models	Entity Layer - Database entities, DbContext, and domain models. Contains entity classes representing database tables and DbContext for managing database connections.
7	DTO	Data Transfer Objects - Request/Response models, API contracts. Contains DTO classes used to transfer data between layers, ensuring data is formatted correctly according to API requirements.
8	Exceptions	Custom Exceptions - Custom error handling, error management. Contains custom exception classes to handle special error cases in the application.

1.2.2 Package Frontend For User

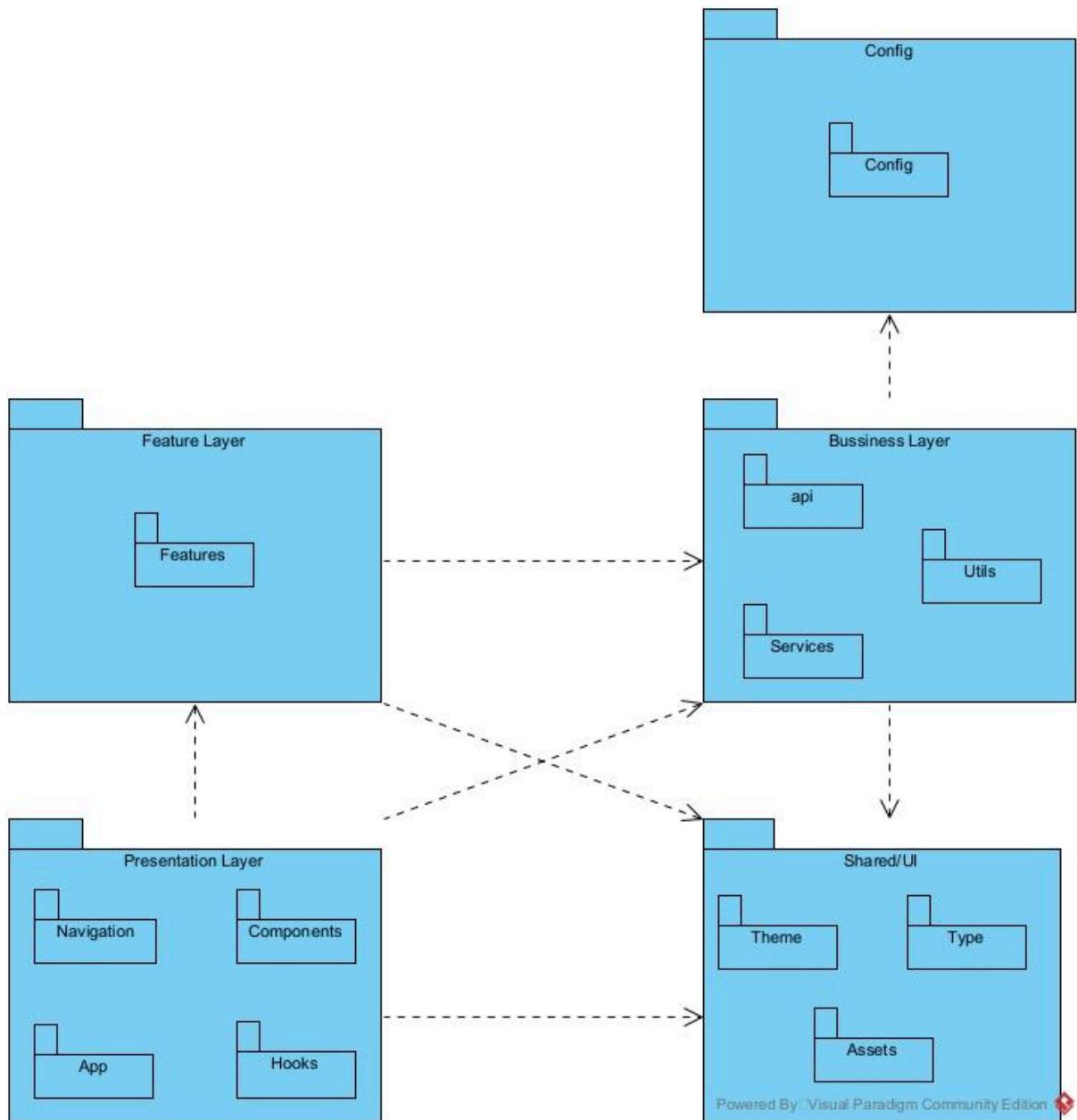


Figure 1.2.2. Package Frontend For User Diagram

Package Descriptions

No	Package	Description
1	api	Contains API client functions and endpoints. Handles all HTTP requests to the backend, including authentication, data fetching, and data submission.
2	app	Contains Redux store configuration, hooks, and global application state management. Manages application-wide state and data flow.
3	components	Contains shared, reusable React Native components used across the application. Includes modals, skeletons, navigation components, and other common UI elements.
4	config	Contains application configuration files, API endpoints, and environment settings. Centralizes configuration management for the application.
5	features	Contains feature-based modules organized by domain. Each feature module includes screens, components, and related logic for a specific application feature.
6	hooks	Contains custom React hooks for reusable logic, API cancellation, badge notifications, and custom alerts. Provides shared hook functionality across the application.
7	navigation	Contains navigation setup, route definitions, and navigation configuration. Manages application routing and screen transitions.
8	services	Contains service classes for location services, navigation services, and SignalR real-time communication. Handles external service integrations and real-time features.
9	theme	Contains theme definitions, colors, and styling constants. Centralizes application theming and color management.
10	types	Contains TypeScript type definitions and global type declarations. Provides type safety and "IntelliSense" support across the application.
11	utils	Contains utility functions for API caching, retry logic, image optimization, JWT handling, storage, and other helper functions. Provides shared utility functionality.
12	assets	Contains static files including images, icons, fonts, and other media resources used throughout the application. Includes avatar images, welcome screens, and UI graphics.

1.2.3 Package Frontend For Admin/Expert

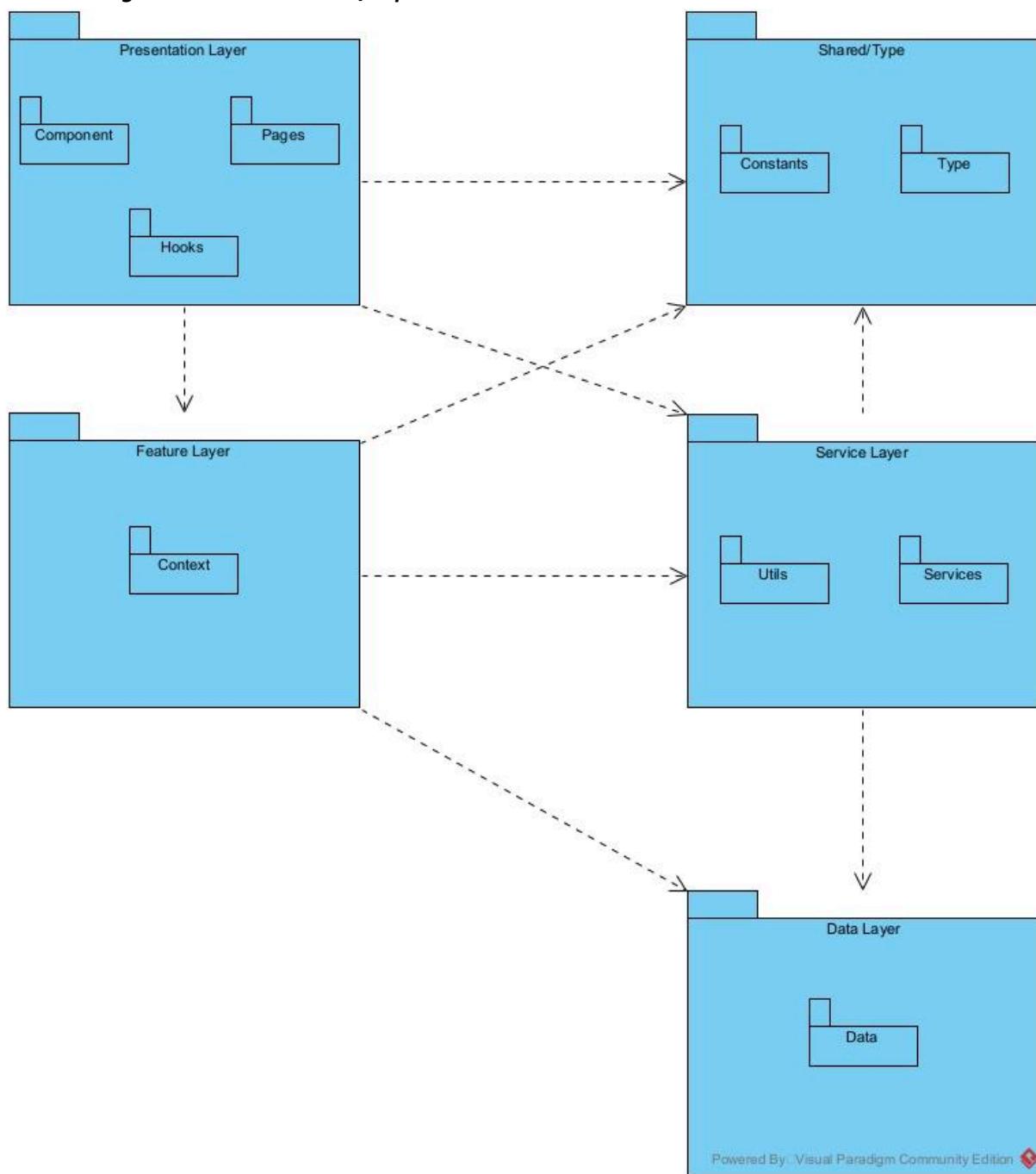


Figure 1.2.3. Package Frontend For Admin/Expert Diagram

Package Descriptions

No	Package	Description
----	---------	-------------

1	components	Contains shared, reusable React components used across the admin application. Includes common components like Header, Sidebar, Layout components, and protected routes.
2	constants	Contains constant values, configuration constants, and application-wide static values used throughout the admin application.
3	context	Contains React Context providers for authentication, notifications, and theme management. Manages global application state using React Context API.
4	data	Contains mock data files for development and testing purposes. Includes sample data for payments, pets, reports, user notifications, and users.
5	hooks	Contains custom React hooks for reusable logic and shared hook functionality across the admin application.
6	pages	Contains page-level components organized by feature. Includes pages for authentication, dashboard, expert management, payment management, pet management, report management, user management, and activities.
7	services	Contains service classes for API communication, authentication, and business logic. Includes API client, dashboard service, expert service, notification service, pet service, report service, and user service.
8	types	Contains TypeScript type definitions and type declarations. Provides type safety and “IntelliSense” support across the admin application.
9	utils	Contains utility functions for formatting (currency, date), validation (email), JWT handling, storage, debouncing, and other helper functions. Provides shared utility functionality.

2. Database Design

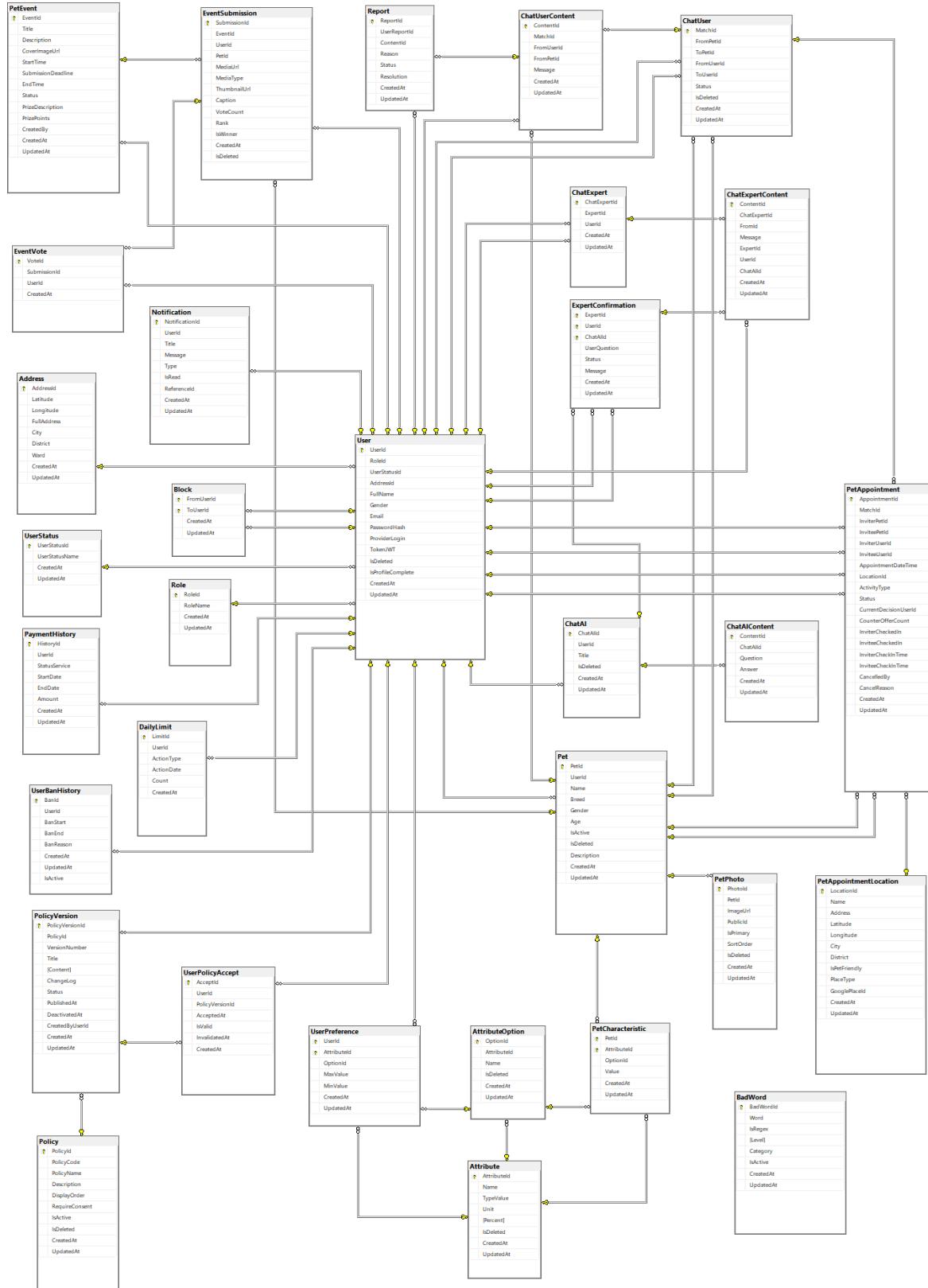


Figure 2. Database Design

2.1 Role

Description: This table stores roles available in the system (e.g., Admin, User).

No	Field	Data Type	Data Size	Constraint	Description
1	RoleId	integer		PK	Primary key, unique identifier for the role.
2	RoleName	varchar	50	NN	Name of the role.
3	CreatedAt	timestamp		NN	Timestamp when the role record was created.
4	UpdatedAt	timestamp		NN	Timestamp when the role record was last updated.

2.2. UserStatus

Description: This table stores possible statuses for users (e.g., Active, Banned, Pending).

No	Field	Data Type	Data Size	Constraint	Description
1	UserStatusId	integer	–	PK	Primary key, unique identifier for user status.
2	UserStatusName	varchar	50	NN	Name of the status.
3	CreatedAt	timestamp	–	NN	Timestamp when created.
4	UpdatedAt	timestamp	–	NN	Timestamp when updated.

2.3 Address

Description: This table stores user address information.

No	Field	Data Type	Data Size	Constraint	Description
1	AddressId	integer	–	PK	Primary key.
2	Latitude	numeric	9,6	NN	Latitude coordinate.
3	Longitude	numeric	9,6	NN	Longitude coordinate.
4	FullAddress	text	–	NN	Complete address text.
5	City	varchar	100	–	City name.
6	District	varchar	100	–	District name.
7	Ward	varchar	100	–	Ward name.

2.4 User

Description: This table stores user account information.

No	Field	Data Type	Data Size	Constraint	Description
1	UserId	integer	–	PK	Primary key.
2	RoleId	integer	–	FK	References Role table.

3	UserStatusId	integer	–	FK	References UserStatus table.
4	AddressId	integer	–	FK	References Address table.
5	FullName	varchar	100	NN	Full name of the user.
6	Gender	varchar	10	NN	Gender (Male/Female/Other).
7	Email	varchar	150	UN	User email must be unique.
8	PasswordHash	text	–	NN	Encrypted password.
9	ProviderLogin	varchar	50	–	Login method (e.g., Google).
10	TokenJWT	text	–	–	JWT token for session.
11	IsDeleted	boolean	–	NN	Soft delete flag.
12	IsProfileComplete	boolean	–	NN	Whether the user profile is complete.
13	CreatedAt	timestamp	–	NN	Created timestamp.
14	UpdatedAt	timestamp	–	NN	Updated timestamp.

2.5 Attribute

Description: This table stores characteristics/attributes that can be used to describe pets or user preferences.

No	Field	Data Type	Data Size	Constraint	Description
1	Attributeld	integer	–	PK	Primary key, unique identifier for the attribute.
2	Name	varchar	100	UN, NN	Name of the attribute (e.g., Coat Length, Color eye).
3	TypeValue	varchar	50	NN	Data type of the value (e.g., option, range, text).
4	Unit	varchar	20	–	Unit of measurement if numeric (e.g., cm, kg).
5	Percent	numeric	5,2	NN	Weight percentage for scoring.
6	IsDeleted	boolean	–	NN	Soft delete flag.
7	CreatedAt	timestamp	–	NN	Timestamp when created.
8	UpdatedAt	timestamp	–	NN	Timestamp when last updated.

2.6 AttributeOption

Description: This table stores selectable options for attributes that use dropdown/choice values.

No	Field	Data Type	Data Size	Constraint	Description
1	OptionId	integer	–	PK, UN, NN	Primary key.
2	AttributId	integer	–	FK, NN	References Attribute table.
3	Name	Character varying	100	NN	Display name of the option.
4	IsDeleted	boolean	–	NN	Soft delete flag.
5	CreatedAt	timestamp	–	NN	Created timestamp.
6	UpdatedAt	timestamp	–	NN	Updated timestamp.

2.7 UserPreference

Description: This table stores the user's preferred pet characteristics.

No	Field	Data Type	Data Size	Constraint	Description
1	UserId	integer	–	PK, FK	References User table.
2	AttributId	integer	–	PK, FK	References Attribute table.
3	OptionId	integer	–	FK	References AttributeOption table (nullable).
4	MaxValue	integer	–	–	Maximum acceptable numeric value for range attributes.
5	MinValue	integer	–	–	Minimum acceptable numeric value for range attributes.
6	CreatedAt	timestamp	–	NN	Created timestamp.
7	UpdatedAt	timestamp	–	NN	Updated timestamp.

2.8 Pet

Description: This table stores registered pets owned by users.

No	Field	Data Type	Data Size	Constraint	Description
1	PetId	integer	–	PK	Primary key.
2	UserId	integer	–	FK	Pet owner, references User table.
3	Name	varchar	100	NN	Pet's name.
4	Breed	varchar	100	NN	Breed type.
5	Gender	varchar	10	NN	Gender of the pet.
6	Age	integer	–	NN	Age in years.

7	IsActive	boolean	–	NN	Whether the pet is visible/active.
8	IsDeleted	boolean	–	NN	Soft delete flag.
9	Description	text	–	–	Additional details about the pet.
10	CreatedAt	timestamp	–	NN	Created timestamp.
11	UpdatedAt	timestamp	–	NN	Updated timestamp.

2.9 PetPhoto

Description: This table stores pet images.

No	Field	Data Type	Data Size	Constraint	Description
1	Photoid	integer	–	PK	Primary key.
2	PetId	integer	–	FK	References Pet table.
3	ImageUrl	text	–	NN	URL of the image.
4	PublicId	text	–	NN	Cloud storage identifier.
5	IsPrimary	boolean	–	NN	Whether this is the main photo.
6	SortOrder	integer	–	NN	Ordering index for multiple photos.
7	IsDeleted	boolean	–	NN	Soft delete flag.
8	CreatedAt	timestamp	–	NN	Created timestamp.
9	UpdatedAt	timestamp	–	NN	Updated timestamp.

2.10 PetCharacteristic

Description: This table stores attribute values for each pet.

No	Field	Data Type	Data Size	Constraint	Description
1	PetId	integer	–	PK, FK	References Pet table.
2	Attributoid	integer	–	PK, FK	References Attribute table.
3	OptionId	integer	–	FK	References AttributeOption table (nullable).
4	Value	integer	–	–	Numeric value if TypeValue is range.
5	CreatedAt	timestamp	–	NN	Created timestamp.
6	UpdatedAt	timestamp	–	NN	Updated timestamp.

2.11 ChatAI

Description: This table stores roles available in the system (e.g., Admin, User).

No	Field	Data Type	Data Size	Constraint	Description
1	ChatAId	integer	–	PK	Primary key.
2	UserId	integer	–	FK	References User table; session owner.
3	Title	varchar	200	NN	Display title for the chat session.
4	IsDeleted	boolean	–	NN	Soft delete flag.
5	CreatedAt	timestamp	–	NN	Created timestamp.
6	UpdatedAt	timestamp	–	NN	Updated timestamp.

2.12 ChatAIContent

Description: This table stores messages exchanged in a ChatAI session.

No	Field	Data Type	Data Size	Constraint	Description
1	ContentId	integer	–	PK	Primary key.
2	ChatAId	integer	–	FK	References ChatAI table.
3	Question	text	–	NN	User's query text.
4	Answer	text	–	NN	AI's response text.
5	CreatedAt	timestamp	–	NN	Created timestamp.
6	UpdatedAt	timestamp	–	NN	Updated timestamp.

2.13 ExpertConfirmation

Description: This table stores confirmation/verification actions performed by experts on ChatAI results.

No	Field	Data Type	Data Size	Constraint	Description
1	ExpertId	integer	–	PK, FK	References User table; expert performing confirmation.
2	UserId	integer	–	PK, FK	References User table; user who asked AI.
3	ChatAId	integer	–	PK, FK	References ChatAI table.
4	Status	text	–	NN	Confirmation status (e.g., Approved, Rejected).

5	Message	text	–	–	Additional explanation from the expert.
6	CreatedAt	timestamp	–	NN	Created timestamp.
7	UpdatedAt	timestamp	–	NN	Updated timestamp.

2.14 ChatUser

Description: This table stores matching/relationship status between two users for chatting.

No	Field	Data Type	Data Size	Constraint	Description
1	MatchId	integer	–	PK	Primary key.
2	FromUserId	integer	–	FK	Initiator, references User table.
3	ToUserId	integer	–	FK	Receiver, references User table.
4	Status	varchar	50	NN	Match/chat status (e.g., Approved, Blocked).
5	IsDeleted	boolean	–	NN	Soft delete flag.
6	CreatedAt	timestamp	–	NN	Created timestamp.
7	UpdatedAt	timestamp	–	NN	Updated timestamp.

2.15 ChatUserContent

Description: This table stores message content exchanged between matched users.

No	Field	Data Type	Data Size	Constraint	Description
1	ContentId	integer	–	PK	Primary key.
2	MatchId	integer	–	FK	References ChatUser table.
3	FromUserId	integer	–	FK	References User table; sender.
4	Message	text	–	NN	Chat message text.
5	CreatedAt	timestamp	–	NN	Created timestamp.
6	UpdatedAt	timestamp	–	NN	Updated timestamp.

2.16 Report

Description: This table stores reports of inappropriate behavior in chats.

No	Field	Data Type	Data Size	Constraint	Description
1	ReportId	integer	–	PK	Primary key.

2	UserReportId	integer	–	FK	References User table; who reported.
3	ContentId	integer	–	FK	References ChatUserContent table.
4	Reason	text	–	NN	Reason for reporting.
5	Status	Character varying	50	NN	Status (Pending, Reviewed).
6	Resolution	text	–	–	Result or admin action taken.
7	CreatedAt	timestamp	–	NN	Created timestamp.
8	UpdatedAt	timestamp	–	NN	Updated timestamp.

2.17 Block

Description: This table stores records of users blocking another user.

No	Field	Data Type	Data Size	Constraint	Description
1	FromUserId	integer	–	PK, FK	References User; who blocks.
2	ToUserId	integer	–	PK, FK	References User; who is blocked.
3	CreatedAt	timestamp	–	NN	Created timestamp.
4	UpdatedAt	timestamp	–	NN	Updated timestamp.

2.18 PaymentHistory

Description: This table stores subscription or service purchase history.

No	Field	Data Type	Data Size	Constraint	Description
1	HistoryId	integer	–	PK	Primary key.
2	UserId	integer	–	FK	References User table.
3	StatusService	Character varying	100	NN	Status Service (e.g., Active, Waiting, Processing).
4	StartDate	date	–	NN	Subscription start date.
5	EndDate	date	–	NN	Subscription end date.
6	CreatedAt	timestamp	–	NN	Created timestamp.
7	UpdatedAt	timestamp	–	NN	Updated timestamp.
8	Amount	numeric	10,2	NN	The amount the user transfers.

2.19 Notification

Description: This table stores notifications for users from the system.

No	Field	Data Type	Data Size	Constraint	Description
1	NotificationId	integer	–	PK	Primary key.
2	UserId	integer	–	FK	References User table.
3	Title	Character varying	200	NN	Notification title text.
4	Message	text	–	NN	Notification message content.
5	CreatedAt	timestamp	–	NN	Created timestamp.
6	UpdatedAt	timestamp	–	NN	Updated timestamp.

2.20 UserBanHistory

Description: Stores a user's ban records — start/end timestamps, reason, active flag, and created/updated timestamps.

No	Field	Data Type	Data Size	Constraint	Description
1	NotificationId	integer	–	PK	Primary key.
2	UserId	integer	–	FK	References User table.
3	BanStart	timestamp	–	NN	Start ban timestamp.
4	BanEnd	timestamp	–	NN	End ban timestamp.
5	BanReason	text	–	NN	Reasons for users being banned.
6	CreatedAt	timestamp	–	NN	Created timestamp.
7	UpdatedAt	timestamp	–	NN	Updated timestamp.
8	IsActive	boolean	–	NN	Is the record valid?

2.21 DailyLimit

Description: Records per-user daily action limits and usage for a given action type and date, including the count and creation time.

No	Field	Data Type	Data Size	Constraint	Description
1	LimitId	integer	–	PK	Primary key.
2	UserId	integer	–	FK	References User table.
3	ActionType	Character varying	100	NN	Type of action.
4	ActionDate	date	–	NN	Date apply limit.
5	Count	integer	–	NN	Number of times the action was performed during the day.
6	CreatedAt	timestamp	–	NN	Record creation time.

2.22 ChatExpert

Description: Represents a conversation between a user and an expert — references the expert and user and contains creation/update timestamps.

No	Field	Data Type	Data Size	Constraint	Description
1	ChatExpertId	integer	—	PK	Primary key.
2	ExpertId	integer	—	FK	References ChatExpertConfirmation table.
3	UserId	integer	—	FK	References User table.
4	CreatedAt	timestamp	—	NN	Created timestamp.
5	UpdatedAt	timestamp	—	NN	Updated timestamp.

2.23 ChatExpertContent

Description: Contains messages within a ChatExpert conversation — message text, sender, optional links to expert/AI records, and created/updated timestamps.

No	Field	Data Type	Data Size	Constraint	Description
1	ContentId	integer	—	PK	Primary key.
2	ChatExpertId	integer	—	FK	References ChatExpert table.
3	FromId	integer	—	FK	References User table.
4	Message	text	—	NN	Message content.
5	ExpertId	integer	—	FK	References ExpertConfirmation table.
6	UserId	integer	—	NN	References User table.
7	ChatAid	integer	—	FK	References ChatAi table.
8	CreateAt	timestamp	—	NN	Created timestamp.
9	UpdatedAt	timestamp	—	NN	Updated timestamp.

2.24 BadWord

Description: This table stores bad words/phrases for content moderation and filtering.

No	Field	Data Type	Data Size	Constraint	Description
1	BadWordId	serial	—	PK	Primary key, auto-increment.
2	Word	varchar	200	NN	The bad word or phrase to filter.
3	IsRegex	boolean	—	—	Whether the word is a regex pattern. Default: FALSE.

4	Level	int	–	NN	Severity level (1 = low, 2 = medium, 3 = high).
5	Category	varchar	50	–	Category of the bad word.
6	IsActive	boolean	–	–	Whether this filter is active. Default: TRUE.
7	CreatedAt	timestamp	–	–	Record creation timestamp. Default: NOW().
8	UpdatedAt	timestamp	–	–	Record last update timestamp. Default: NOW().

2.25 Policy

Description: This table stores policy information (Terms of Service, Privacy Policy, etc.).

No	Field	Data Type	Data Size	Constraint	Description
1	PolicyId	serial	–	PK	Primary key, auto-increment.
2	PolicyCode	varchar	50	NN	Unique code for the policy.
3	PolicyName	varchar	200	NN	Name of the policy.
4	Description	text	–	–	Description of the policy.
5	DisplayOrder	integer	–	–	Order for displaying policies. Default: 0.
6	RequireConsent	boolean	–	–	Whether user consent is required. Default: TRUE.
7	IsActive	boolean	–	–	Whether this policy is active. Default: TRUE.
8	IsDeleted	boolean	–	–	Soft delete flag. Default: FALSE.
9	CreatedAt	timestamp	–	–	Record creation timestamp. Default: NOW().
10	UpdatedAt	timestamp	–	–	Record last update timestamp. Default: NOW().

2.26 PolicyVersion

Description: This table stores version history of each policy.

No	Field	Data Type	Data Size	Constraint	Description
1	PolicyVersionId	serial	–	PK	Primary key, auto-increment.
2	PolicyId	integer	–	FK	Reference to Policy table. On Delete Cascade.
3	VersionNumber	integer	–	NN	Version number of the policy.

4	Title	varchar	300	NN	Title of this version.
5	Content	text	–	NN	Full content of the policy.
6	ChangeLog	text	–	–	Description of changes in this version.
7	Status	varchar	20	NN	Status: DRAFT, ACTIVE, INACTIVE. Default: DRAFT.
8	PublishedAt	timestamp	–	–	When this version was published.
9	DeactivatedAt	timestamp	–	–	When this version was deactivated.
10	CreatedByUserId	integer	–	FK	User who created this version. On Delete Set Null.
11	CreatedAt	timestamp	–	–	Record creation timestamp. Default: NOW().
12	UpdatedAt	timestamp	–	–	Record last update timestamp. Default: NOW().

2.27 UserPolicyAccept

Description: This table stores user acceptance history for policy versions.

No	Field	Data Type	Data Size	Constraint	Description
1	AcceptId	bigserial	–	PK	Primary key, auto-increment.
2	UserId	integer	–	FK	Reference to User table. On Delete Cascade.
3	PolicyVersionId	integer	–	FK	Reference to PolicyVersion table. On Delete Cascade.
4	AcceptedAt	timestamp	–	NN	When user accepted the policy. Default: NOW().
5	IsValid	boolean	–	–	Whether this acceptance is still valid. Default: TRUE.
6	InvalidatedAt	timestamp	–	–	When this acceptance was invalidated.
7	CreatedAt	timestamp	–	–	Record creation timestamp. Default: NOW().

2.28 PetAppointmentLocation

Description: This table stores pet-friendly locations for appointments.

No	Field	Data Type	Data Size	Constraint	Description
1	LocationId	serial	–	PK	Primary key, auto-increment.

2	Name	varchar	200	NN	Name of the location.
3	Address	text	–	NN	Full address of the location.
4	Latitude	decimal	(9,6)	NN	GPS latitude coordinate.
5	Longitude	decimal	(9,6)	NN	GPS longitude coordinate.
6	City	varchar	100	–	City name.
7	District	varchar	100	–	District name.
8	IsPetFriendly	boolean	–	–	Whether location is pet-friendly. Default: TRUE.
9	PlaceType	varchar	50	–	Type: park, pet_cafe, vet_clinic, custom.
10	GooglePlaceId	varchar	255	–	Google Maps Place ID.
11	CreatedAt	timestamp	–	–	Record creation timestamp. Default: NOW().
12	UpdatedAt	timestamp	–	–	Record last update timestamp. Default: NOW().

2.29 PetAppointment

Description: This table stores pet meeting appointments between matched users.

No	Field	Data Type	Data Size	Constraint	Description
1	AppointmentId	serial	–	PK	Primary key, auto-increment.
2	MatchId	int	–	FK	Reference to ChatUser (match).
3	InviterPetId	int	–	FK	Pet ID of the inviter.
4	InviteePetId	int	–	FK	Pet ID of the invitee.
5	InviterUserId	int	–	FK	User ID of the inviter.
6	InviteeUserId	int	–	FK	User ID of the invitee.
7	AppointmentDateTime	timestamp	–	NN	Scheduled date and time.
8	LocationId	int	–	FK	Reference to PetAppointmentLocation.
9	ActivityType	varchar	50	NN	Type: walk, cafe, playdate.
10	Status	varchar	30	–	Status: pending, confirmed, rejected, cancelled, on_going, completed, no_show. Default: pending.

11	CurrentDecisionUserId	int	–	FK	User who needs to make decision.
12	CounterOfferCount	int	–	–	Number of counter offers (max 3). Default: 0.
13	InviterCheckedIn	boolean	–	–	Whether inviter has checked in. Default: FALSE.
14	InviteeCheckedIn	boolean	–	–	Whether invitee has checked in. Default: FALSE.
15	InviterCheckInTime	timestamp	–	–	Inviter check-in timestamp.
16	InviteeCheckInTime	timestamp	–	–	Invitee check-in timestamp.
17	CancelledBy	int	–	FK	User who cancelled the appointment.
18	CancelReason	text	–	–	Reason for cancellation.
19	CreatedAt	timestamp	–	–	Record creation timestamp. Default: NOW().
20	UpdatedAt	timestamp	–	–	Record last update timestamp. Default: NOW().

2.30 PetEvent

Description: This table stores online pet photo/video competition events.

No	Field	Data Type	Data Size	Constraint	Description
1	EventId	serial	–	PK	Primary key, auto-increment.
2	Title	varchar	200	NN	Event title.
3	Description	text	–	–	Event description.
4	CoverImageUrl	varchar	500	–	Cover/poster image URL.
5	StartTime	timestamp	–	NN	Event start time.
6	SubmissionDeadline	timestamp	–	NN	Deadline for submissions.
7	EndTime	timestamp	–	NN	Event end time (voting ends).
8	Status	varchar	30	–	Status: upcoming, active, submission_closed, voting_ended, completed, cancelled. Default: upcoming.

9	PrizeDescription	text	–	–	Description of prizes.
10	PrizePoints	int	–	–	Points awarded to winners. Default: 0.
11	CreatedBy	int	–	FK	Admin who created the event.
12	CreatedAt	timestamp	–	–	Record creation timestamp. Default: NOW().
13	UpdatedAt	timestamp	–	–	Record last update timestamp. Default: NOW().

2.31 EventSubmission

Description: This table stores user submissions for pet events.

No	Field	Data Type	Data Size	Constraint	Description
1	SubmissionId	serial	–	PK	Primary key, auto-increment.
2	EventId	int	–	FK	Reference to PetEvent.
3	UserId	int	–	FK	User who submitted.
4	PetId	int	–	FK	Pet in the submission.
5	MediaUrl	varchar	500	NN	URL of submitted media.
6	MediaType	varchar	20	NN	Type: image, video.
7	ThumbnailUrl	varchar	500	–	Thumbnail URL for videos.
8	Caption	varchar	500	–	Caption for the submission.
9	VoteCount	int	–	–	Total votes received. Default: 0.
10	Rank	int	–	–	Final ranking (1, 2, 3 for top 3).
11	IsWinner	boolean	–	–	Whether this is a winning entry. Default: FALSE.
12	CreatedAt	timestamp	–	–	Record creation timestamp. Default: NOW().
13	IsDeleted	boolean	–	–	Soft delete flag. Default: FALSE.

2.32 EventVote

Description: This table stores votes for event submissions.

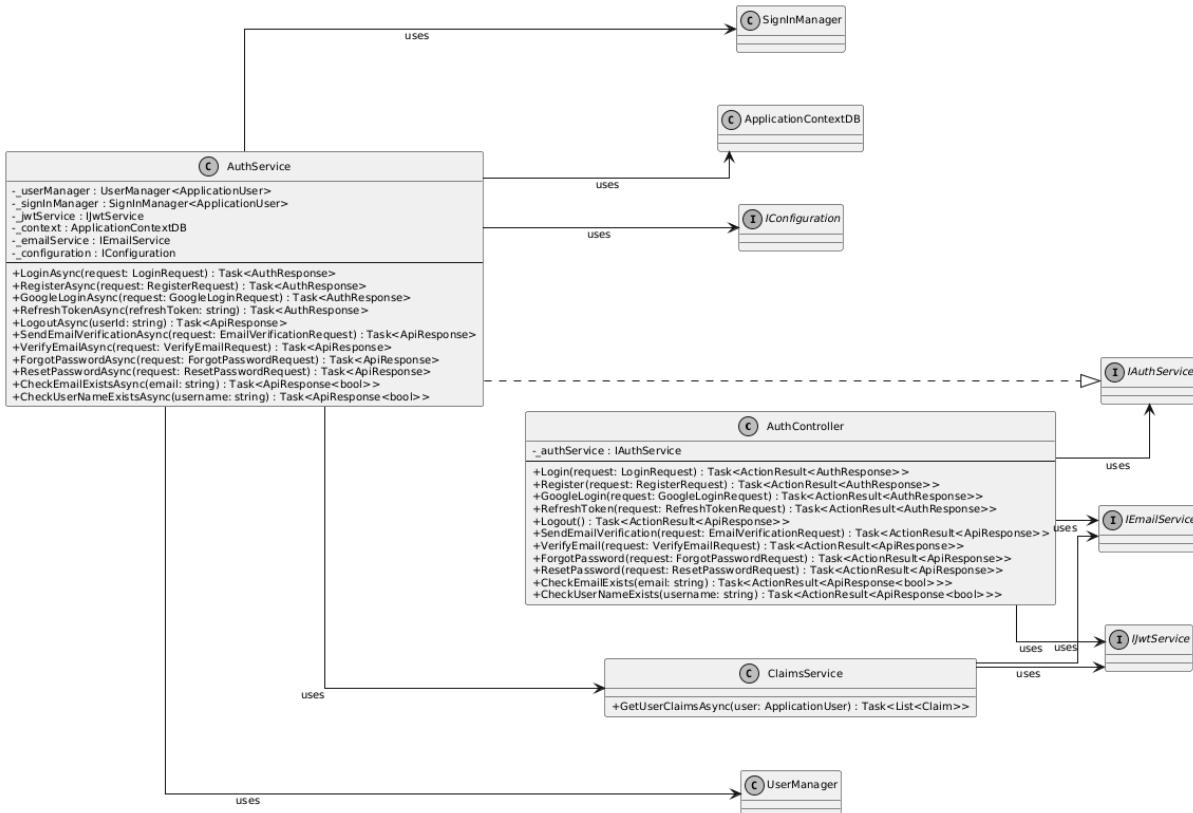
No	Field	Data Type	Data Size	Constraint	Description
1	VoteId	serial	–	PK	Primary key, auto-increment.
2	SubmissionId	int	–	FK	Reference to EventSubmission. On Delete Cascade.

3	UserId	int	-	FK	User who voted.
4	CreatedAt	timestamp	-	-	Vote timestamp. Default: NOW().

3. Detailed Design

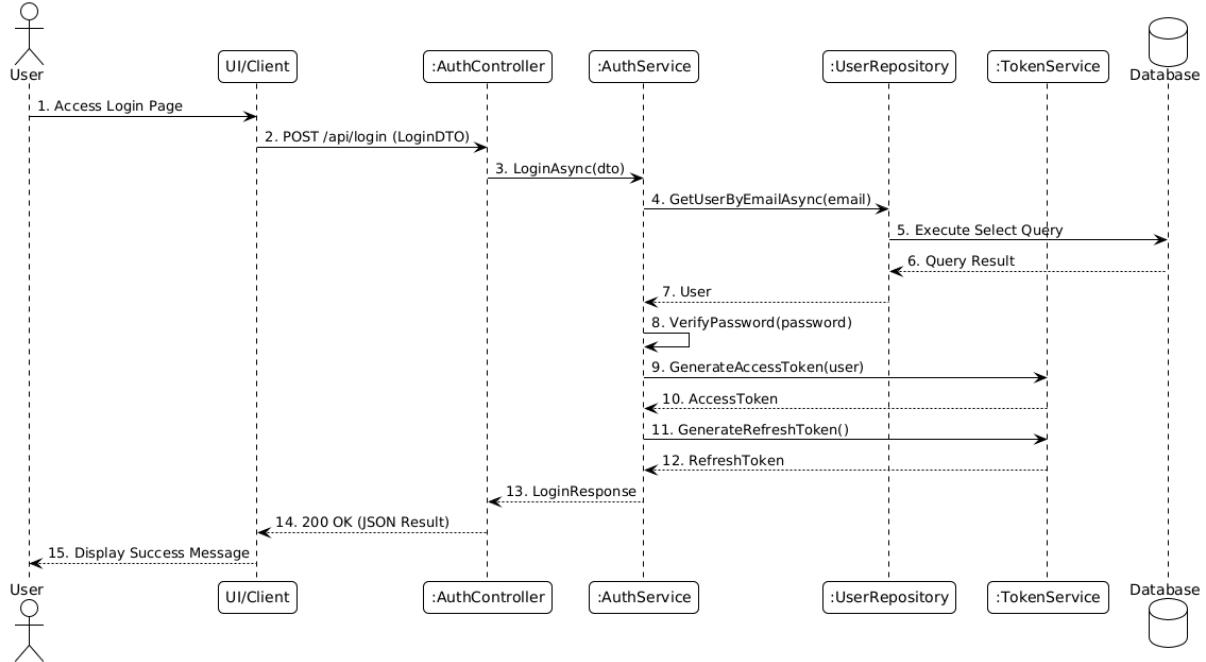
3.1 Authentication & Authorization

3.1.1 Class Diagram

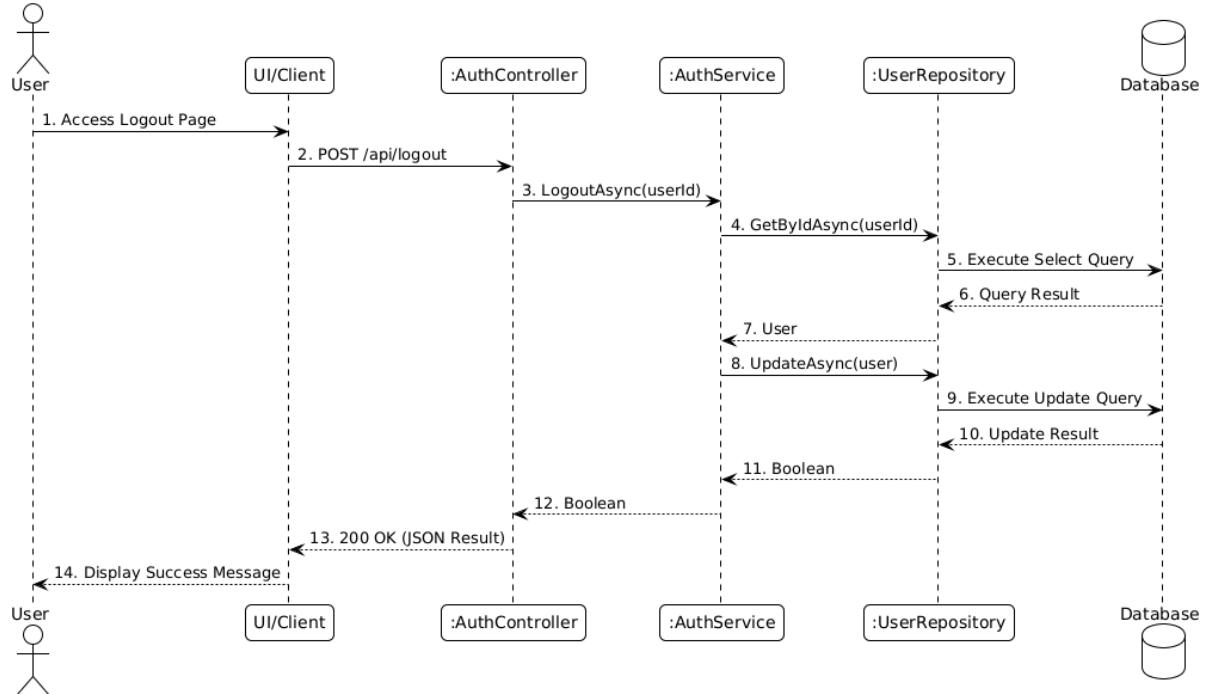


3.1.2 Sequence Diagram

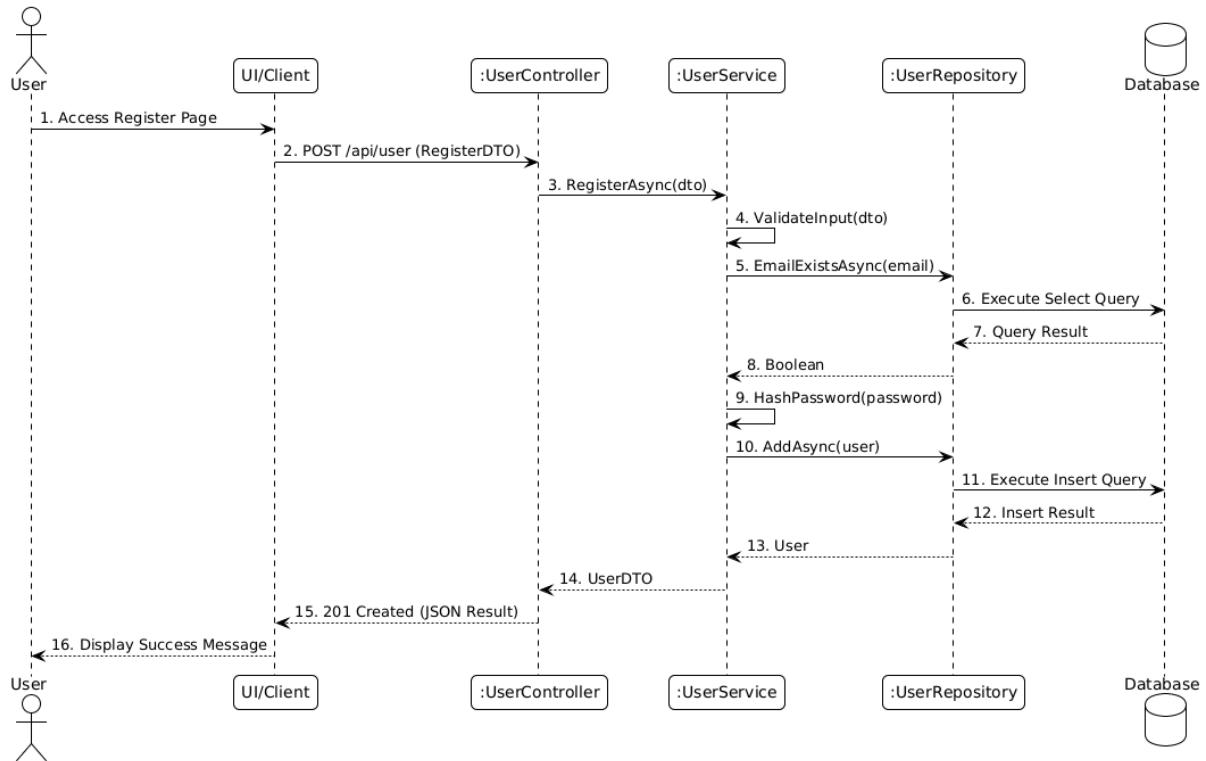
3.1.2.1 Login



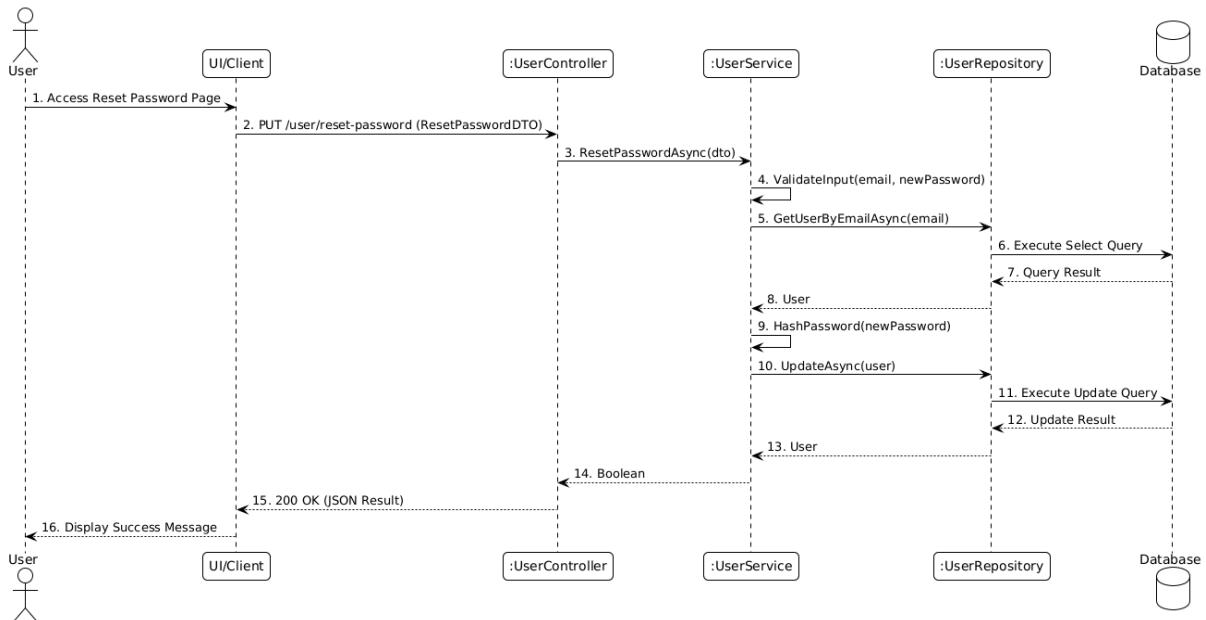
3.1.2.2 Logout



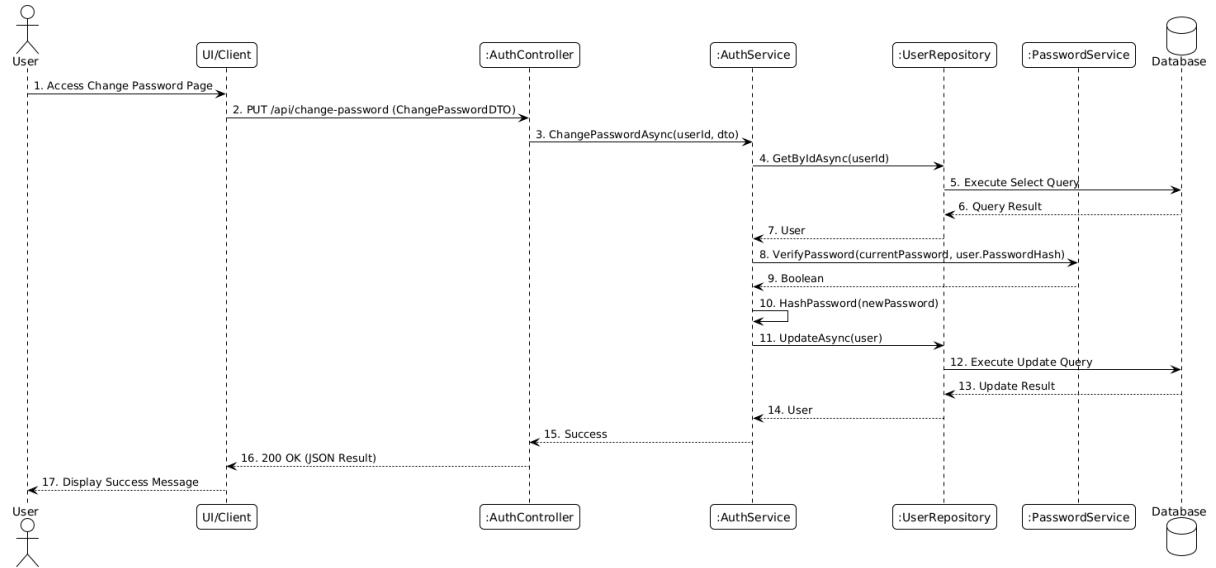
3.1.2.3 Register Account



3.1.2.4 Reset Password

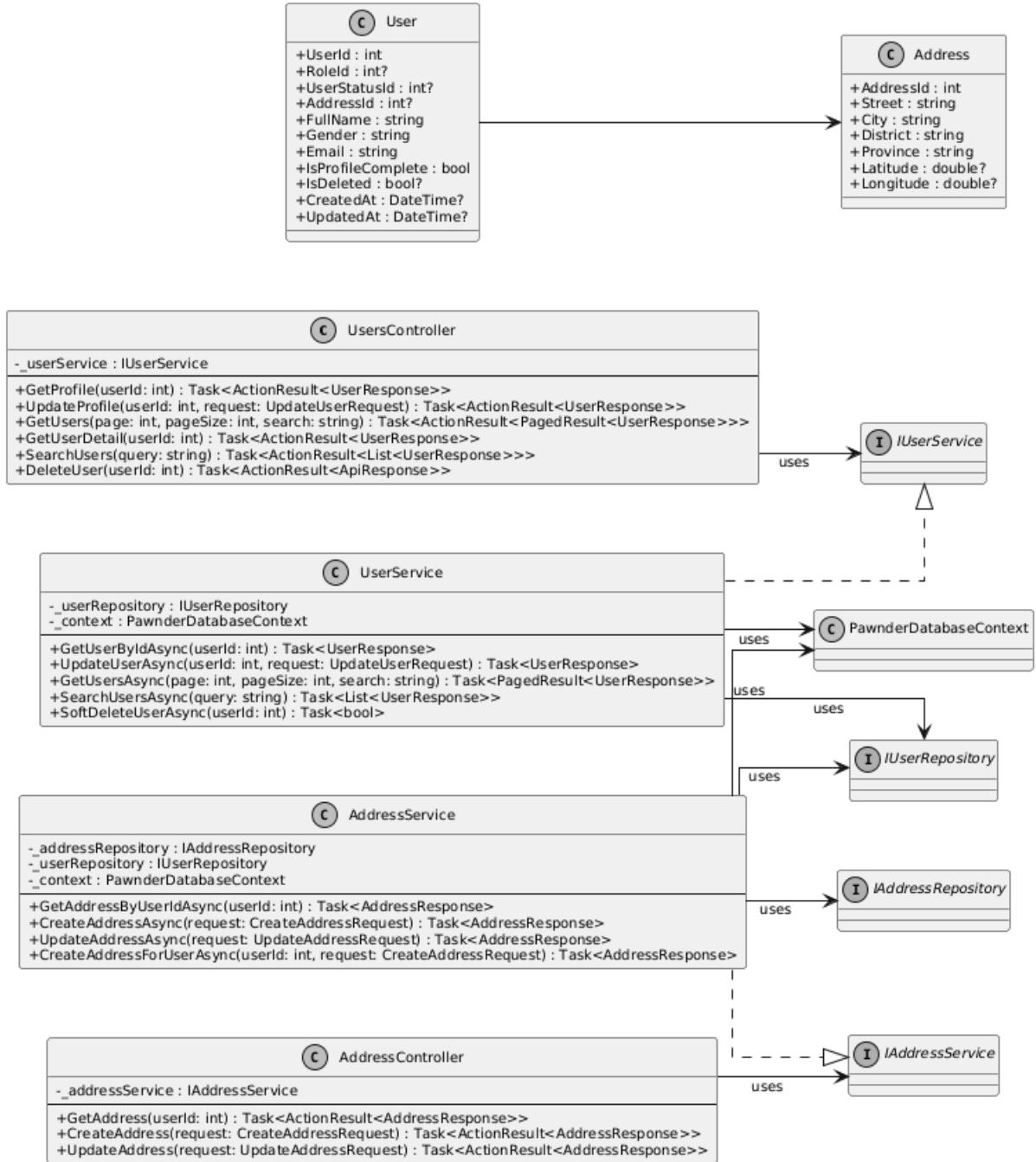


3.1.2.5 Change Password



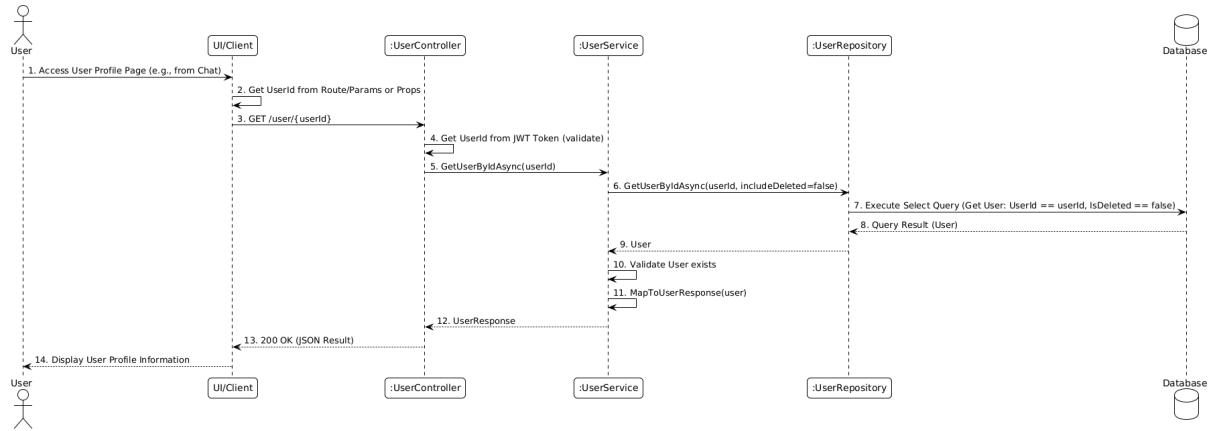
3.2 User Profile Management

3.2.1 Class Diagram

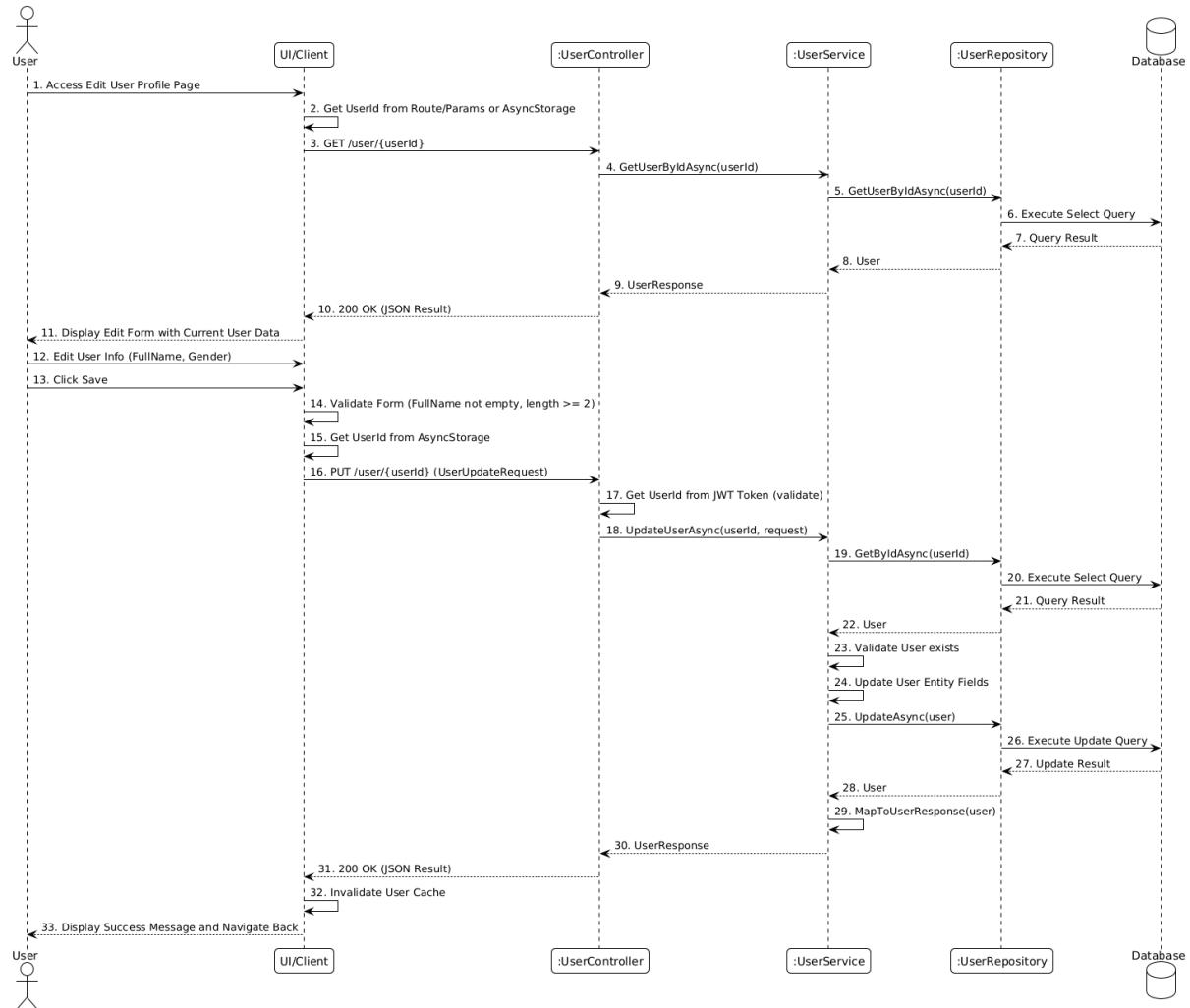


3.2.2 Sequence Diagram

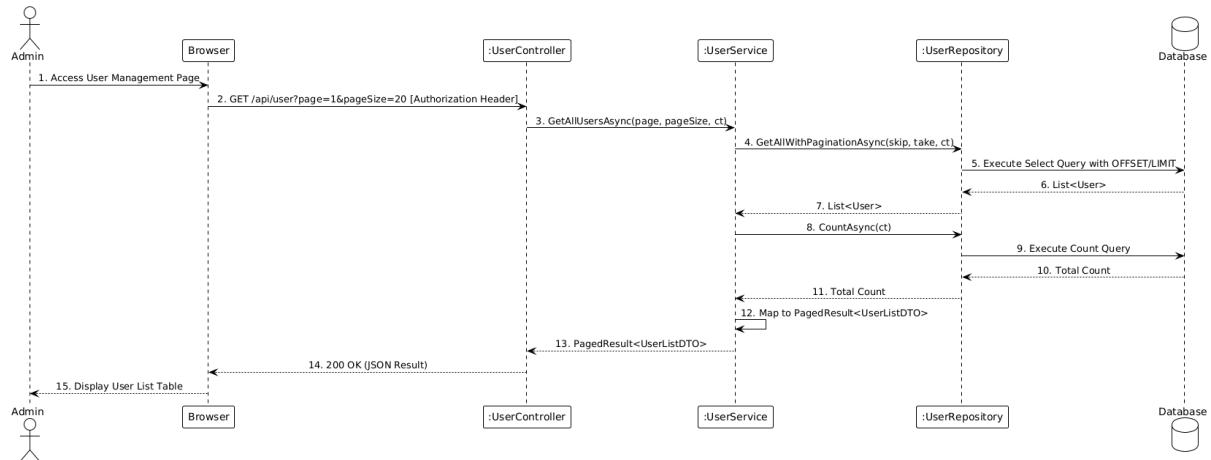
3.2.2.1 View User Profile



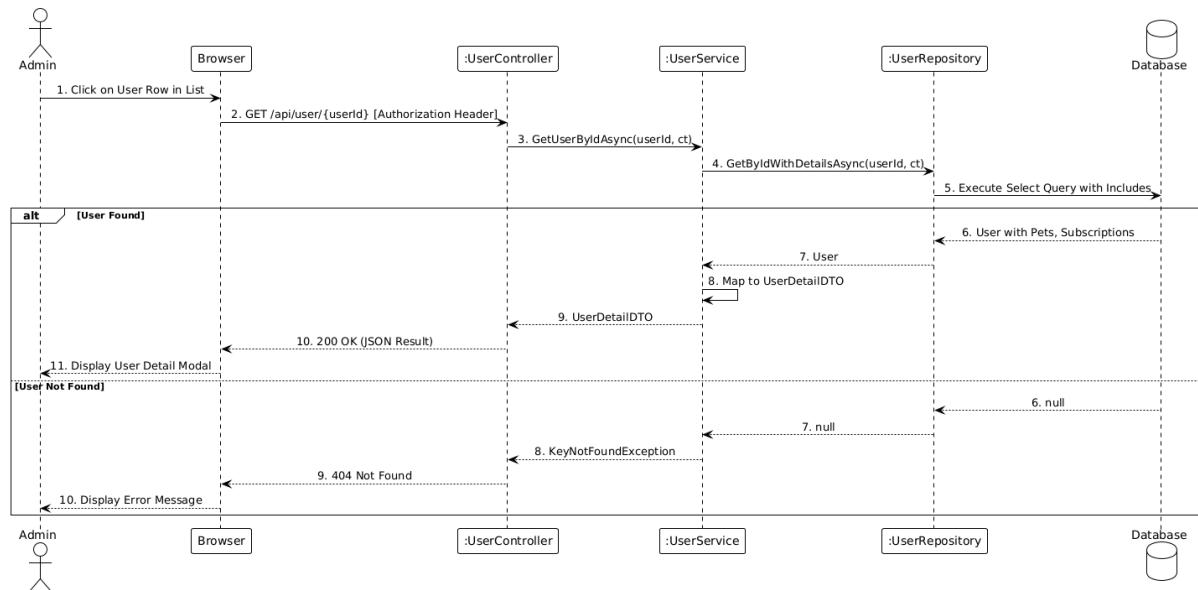
3.2.2.2 Edit User Profile



3.2.2.3 View User List

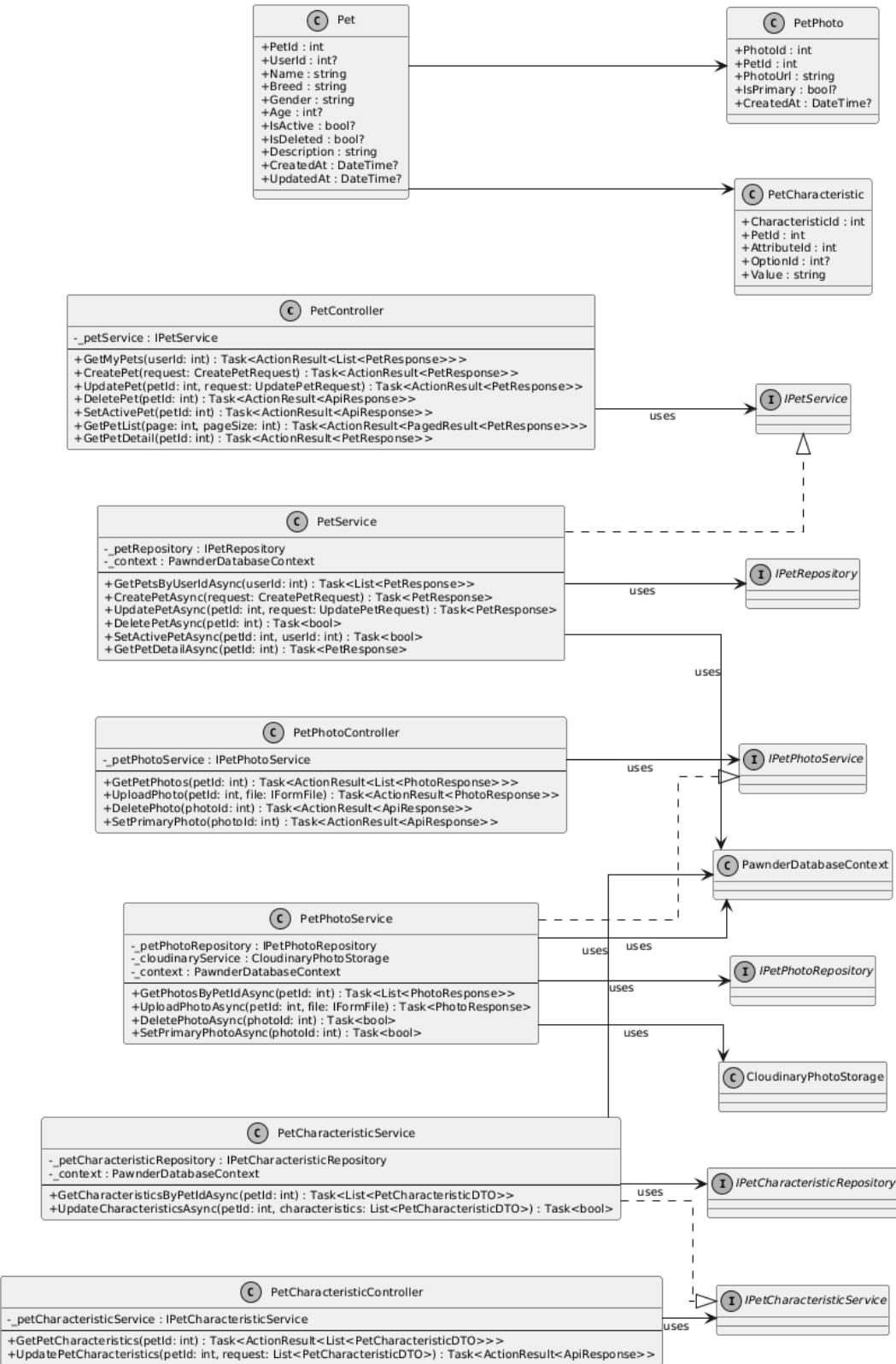


3.2.2.4 View User Detail



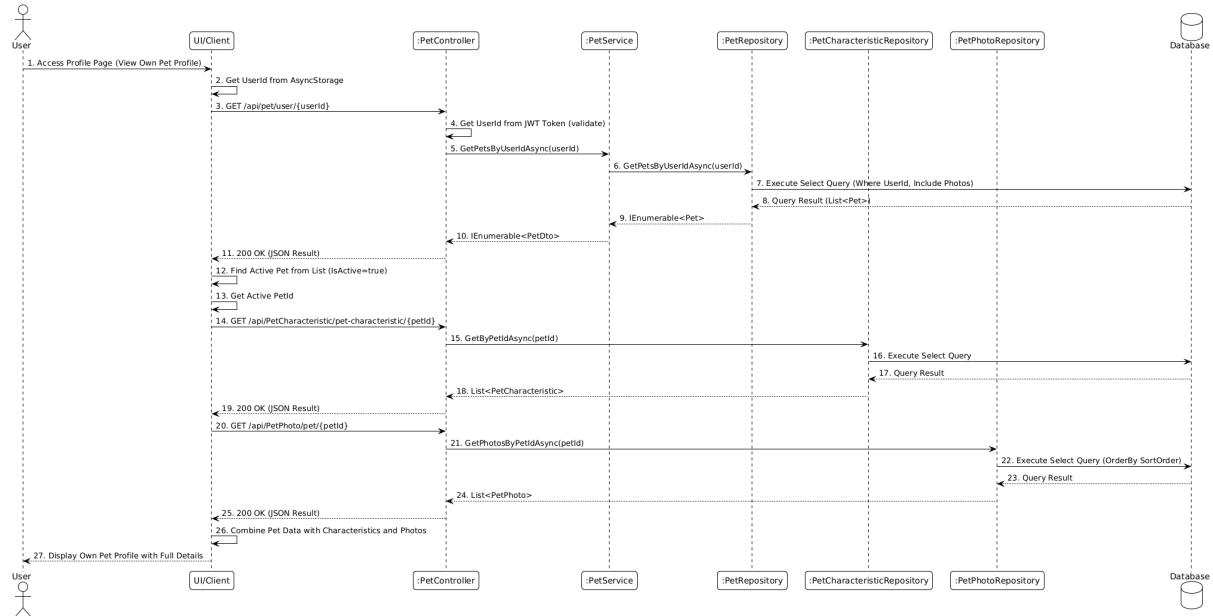
3.3 Pet Profile Management

3.3.1 Class Diagram

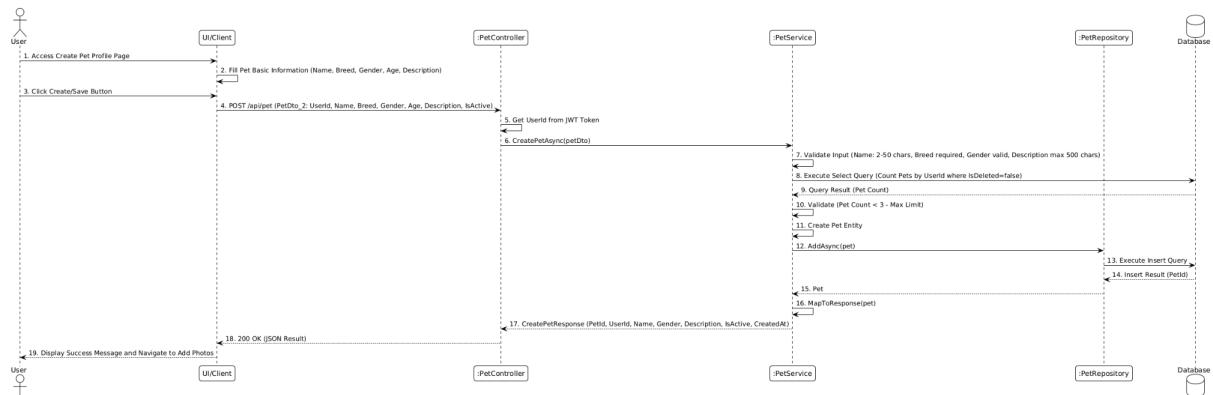


3.3.2 Sequence Diagram

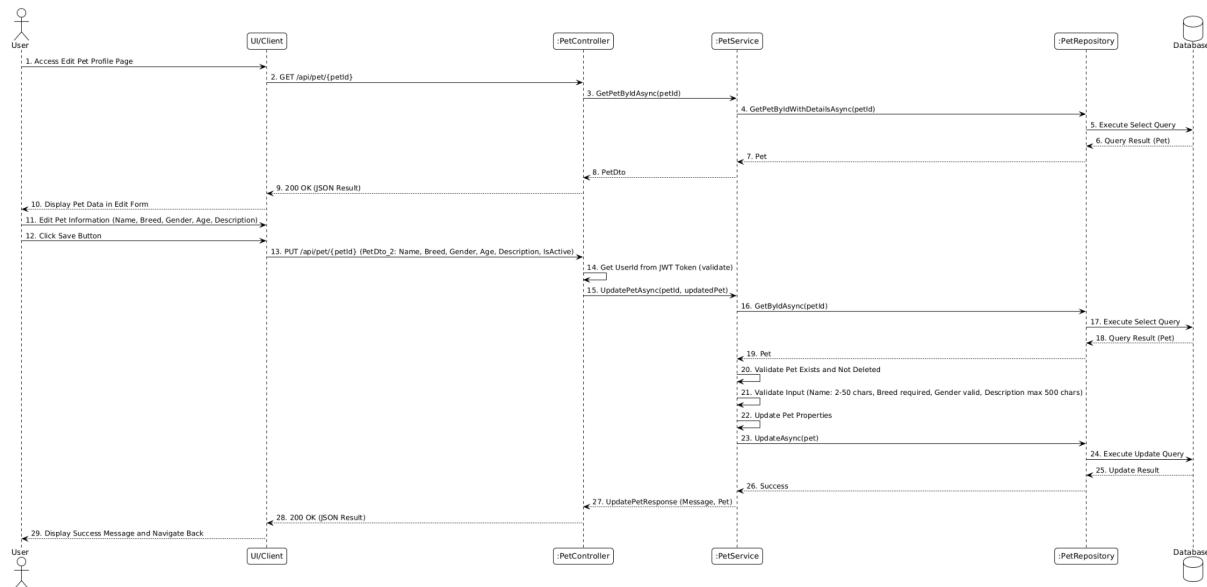
3.3.2.1 View Own Pet Profile



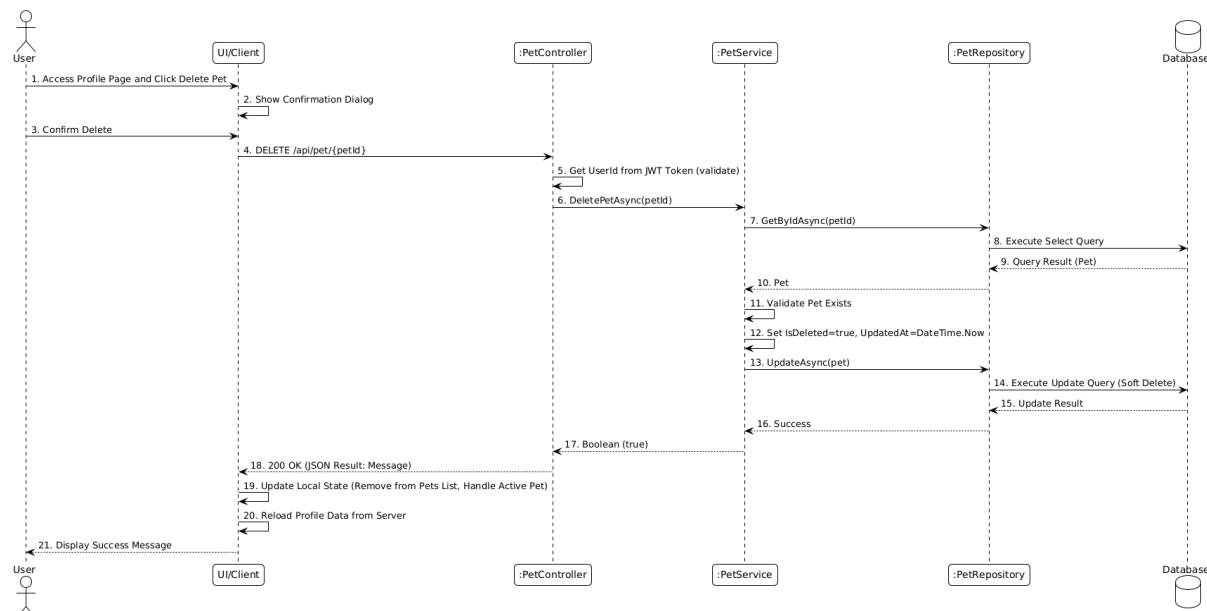
3.3.2.2 Create Pet Profile



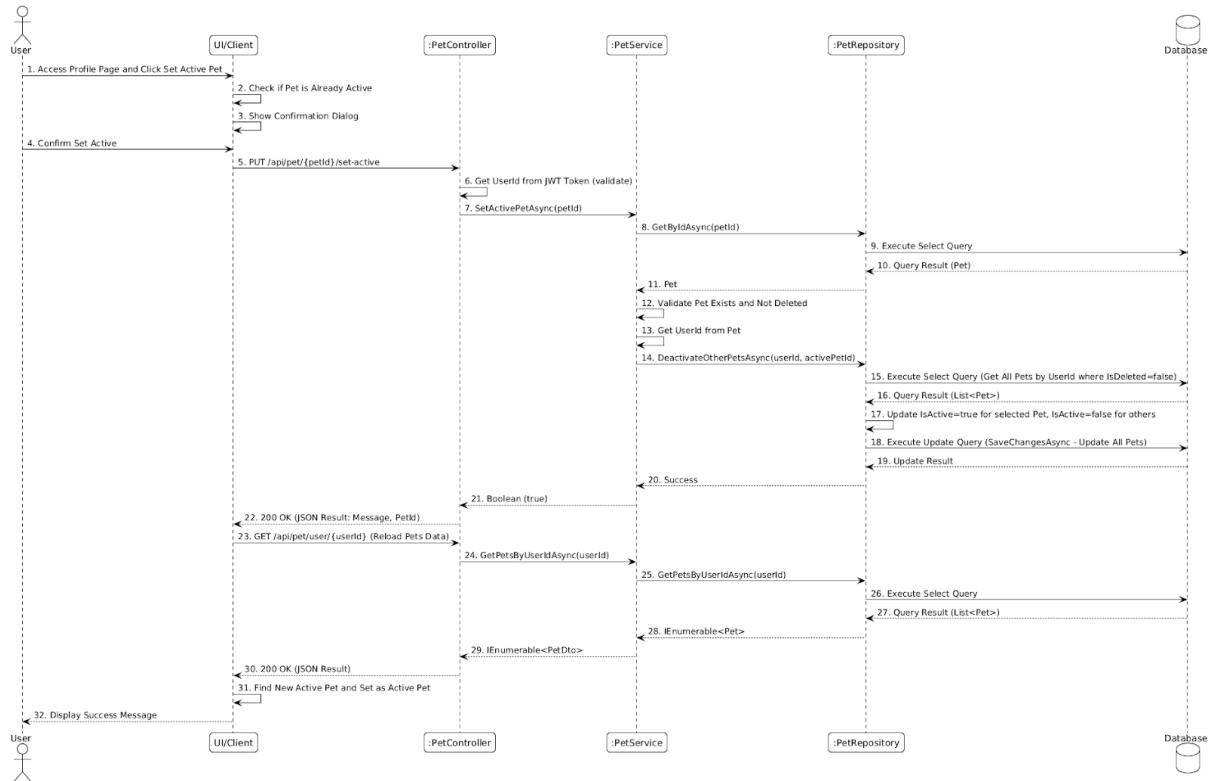
3.3.2.3 Edit Pet Profile



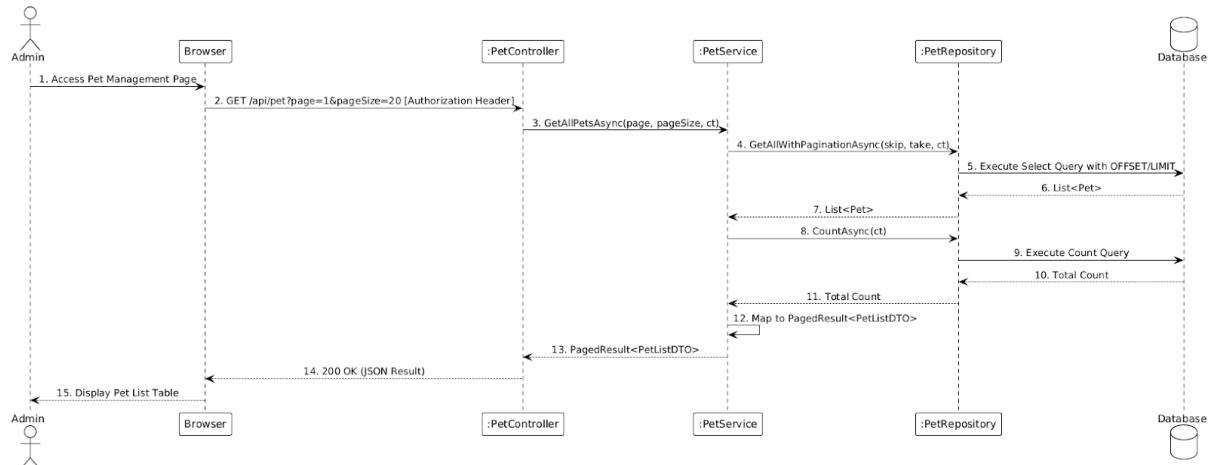
3.3.2.4 Delete Pet Profile



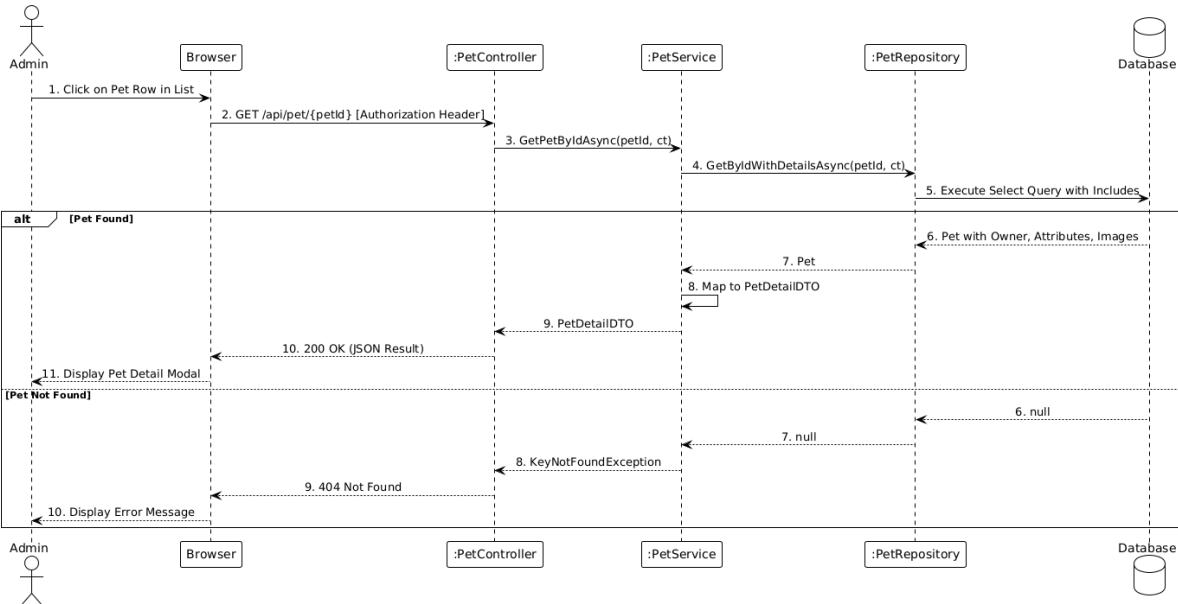
3.3.2.5 Set Active Pet Profile



3.3.2.6 View Pet List

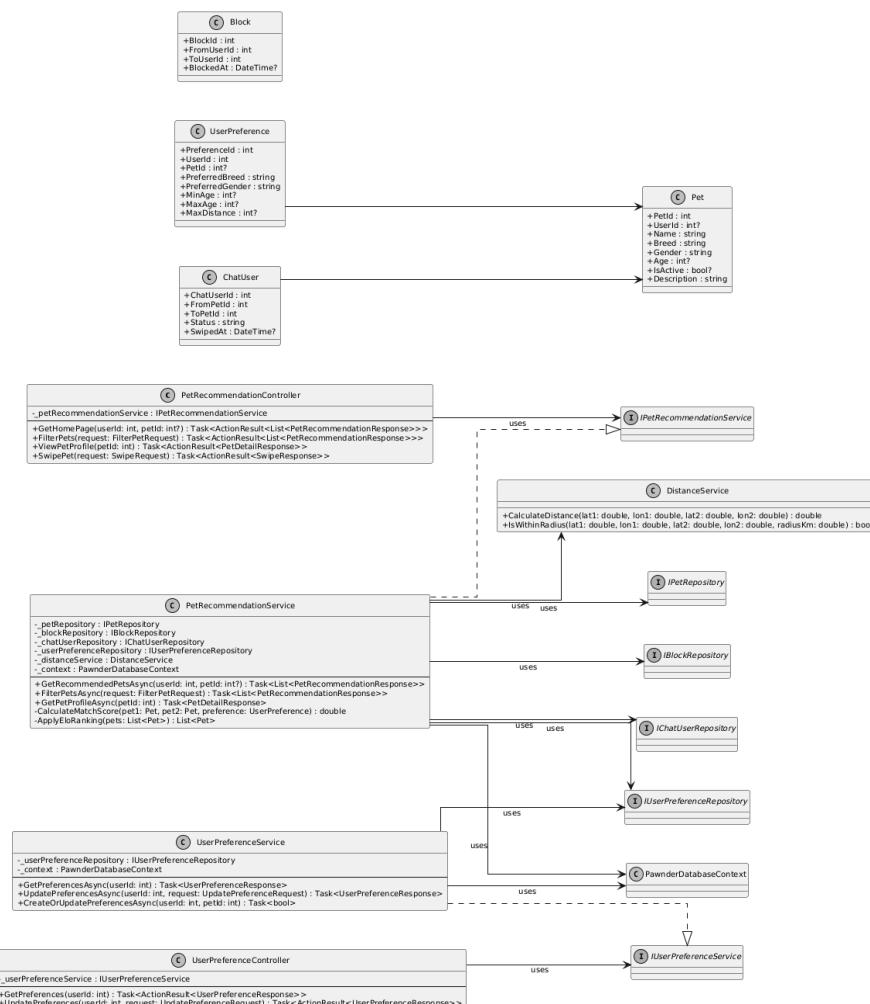


3.3.2.7 View Pet Detail



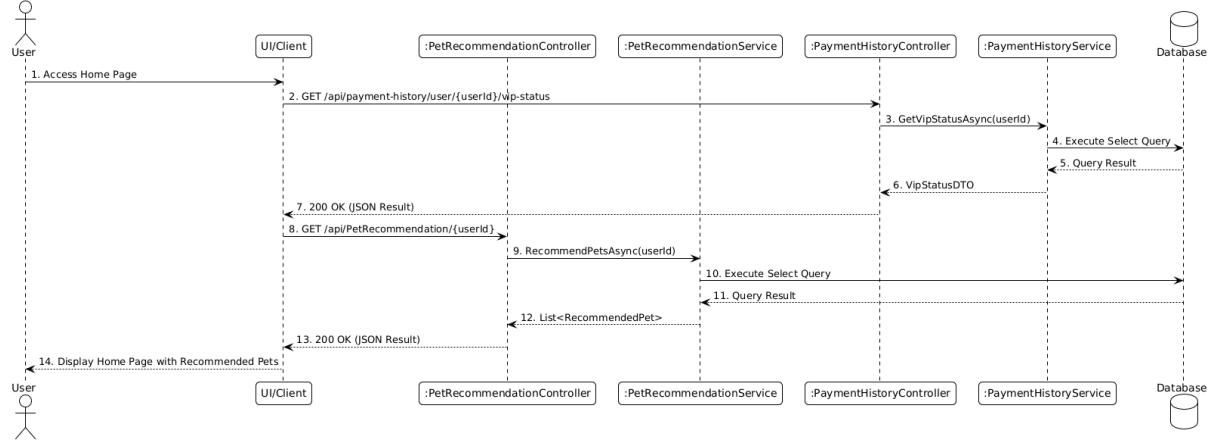
3.4 Pet Discovery & Swiping

3.4.1 Class Diagram

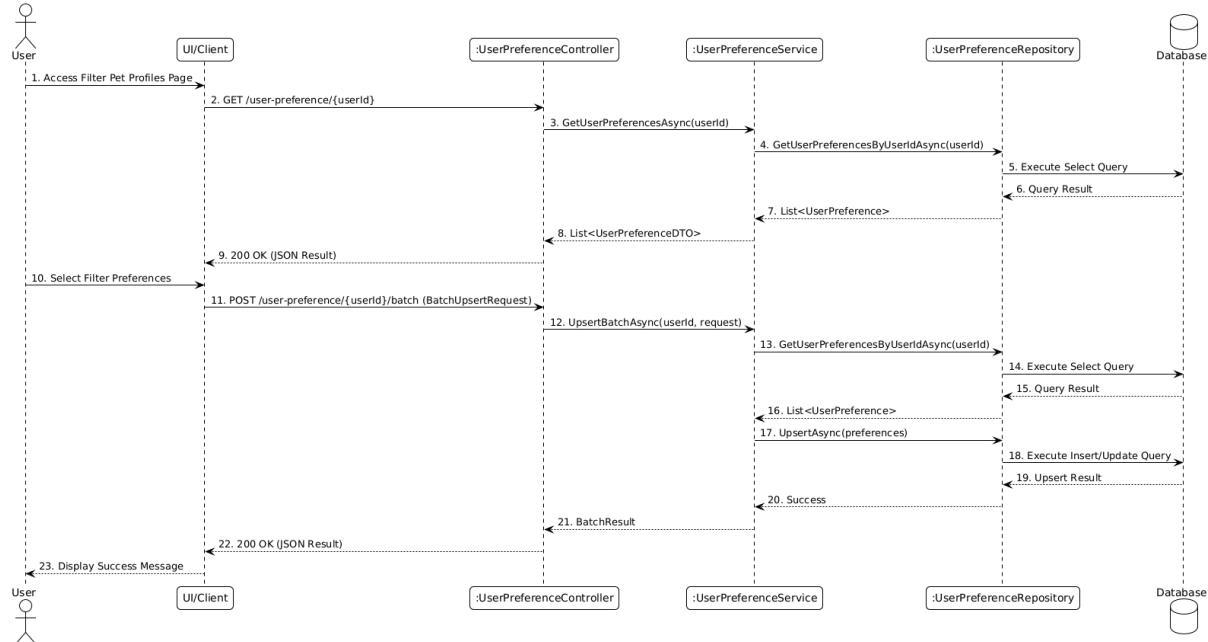


3.4.2 Sequence Diagram

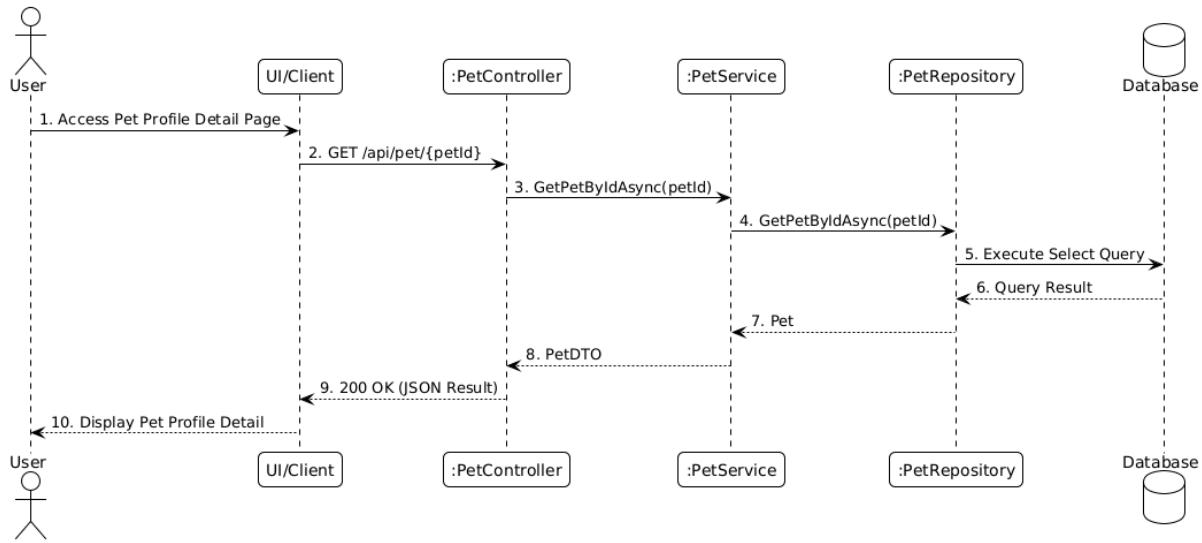
3.4.2.1 View Home Page



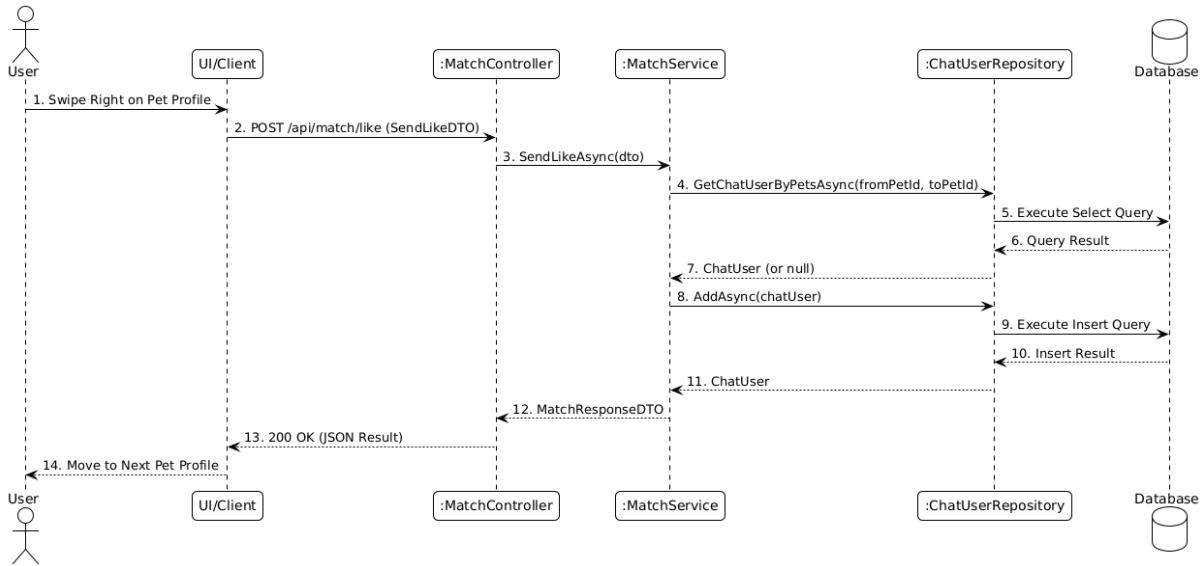
3.4.2.2 Filter Pet Profile



3.4.2.3 View Pet Profile

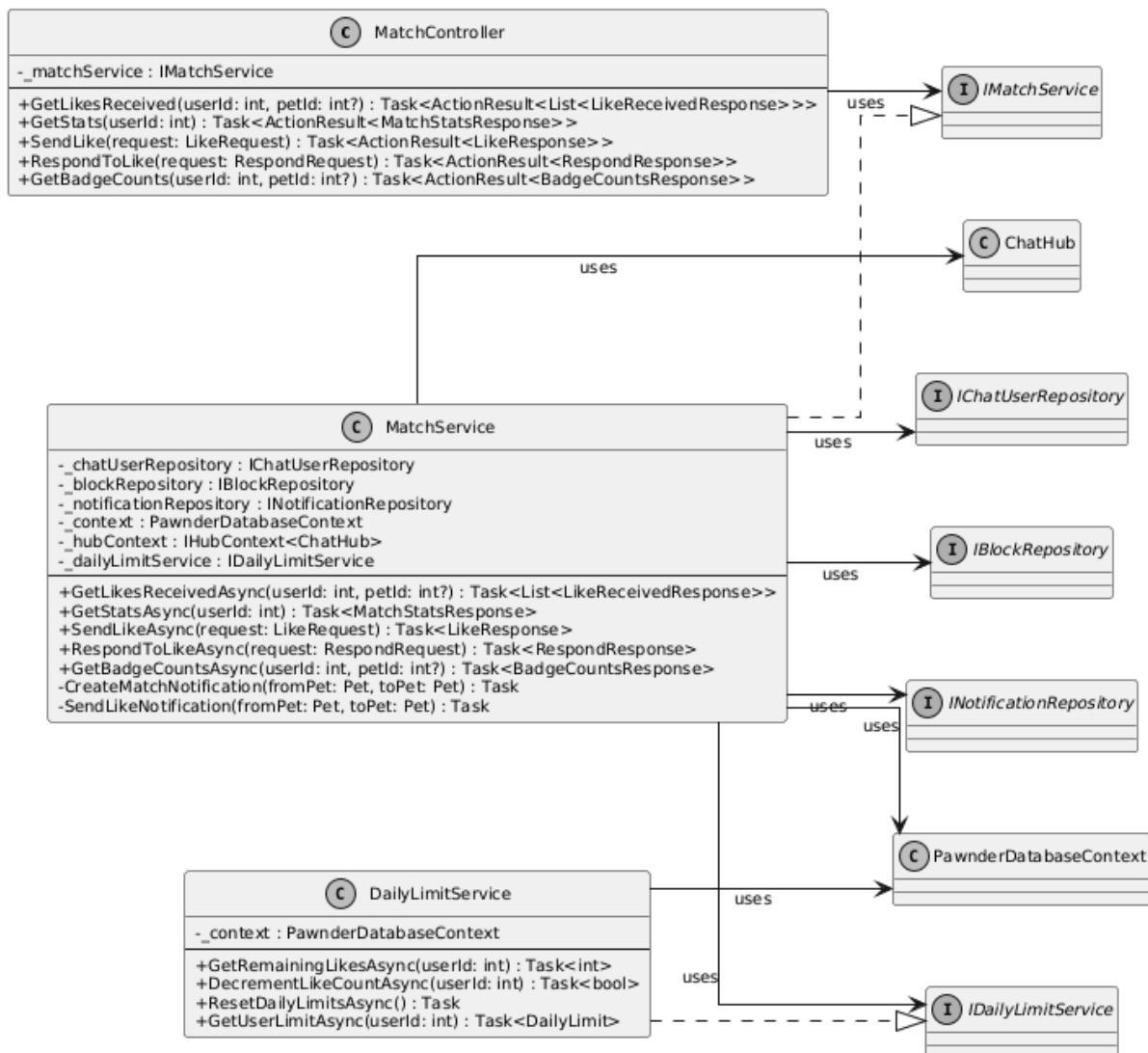
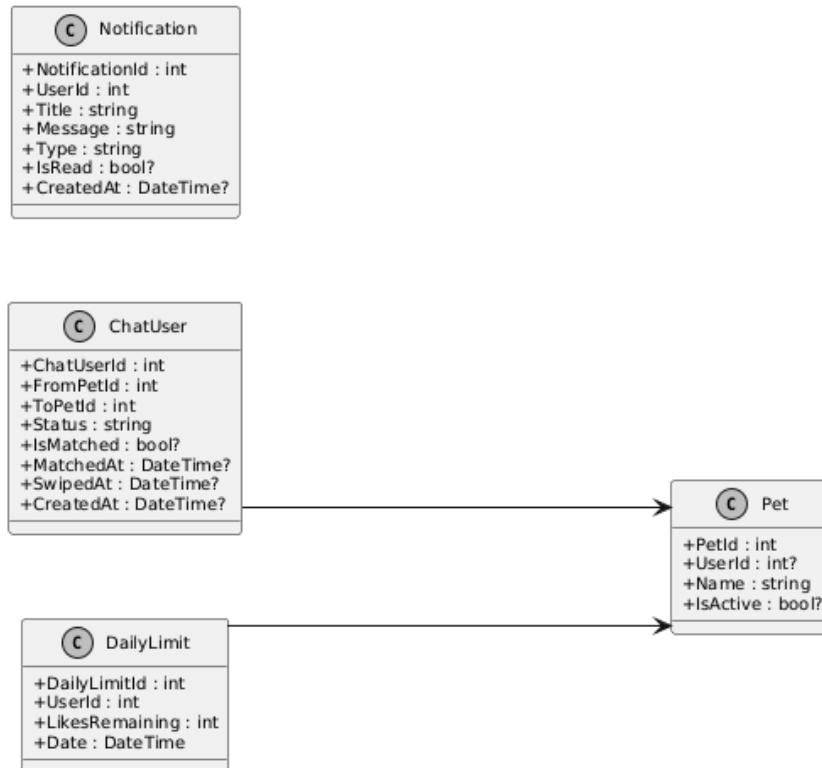


3.4.2.4 Swipe Pet Profile



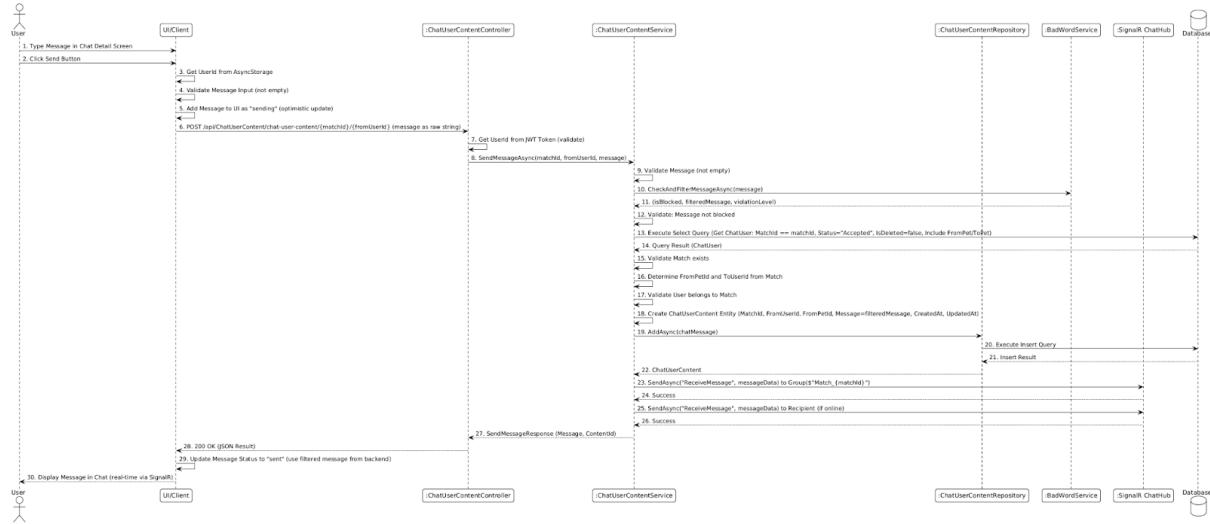
3.5 Matching

3.5.1 Class Diagram

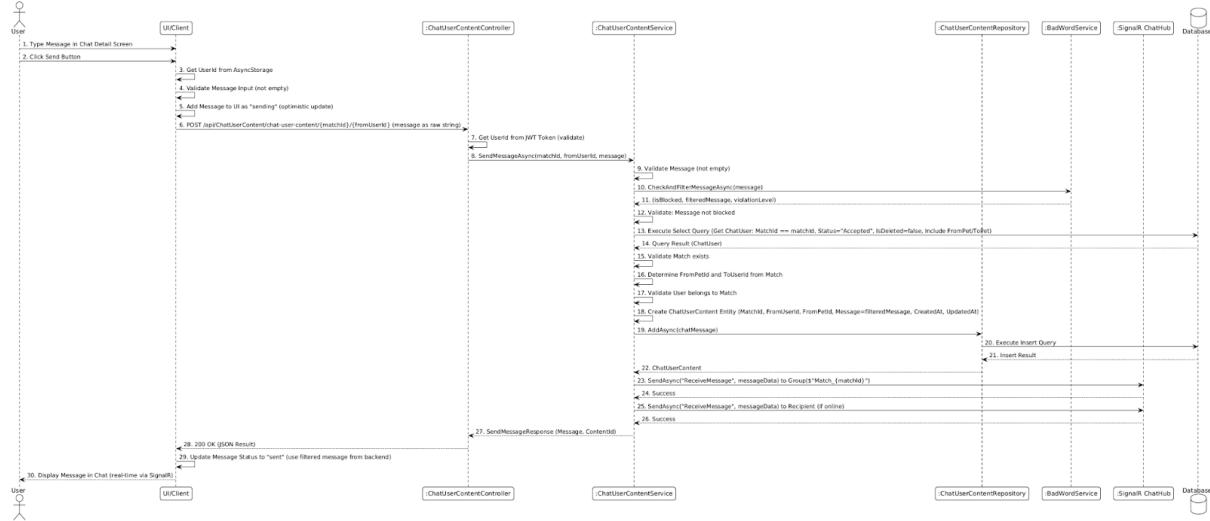


3.5.2 Sequence Diagram

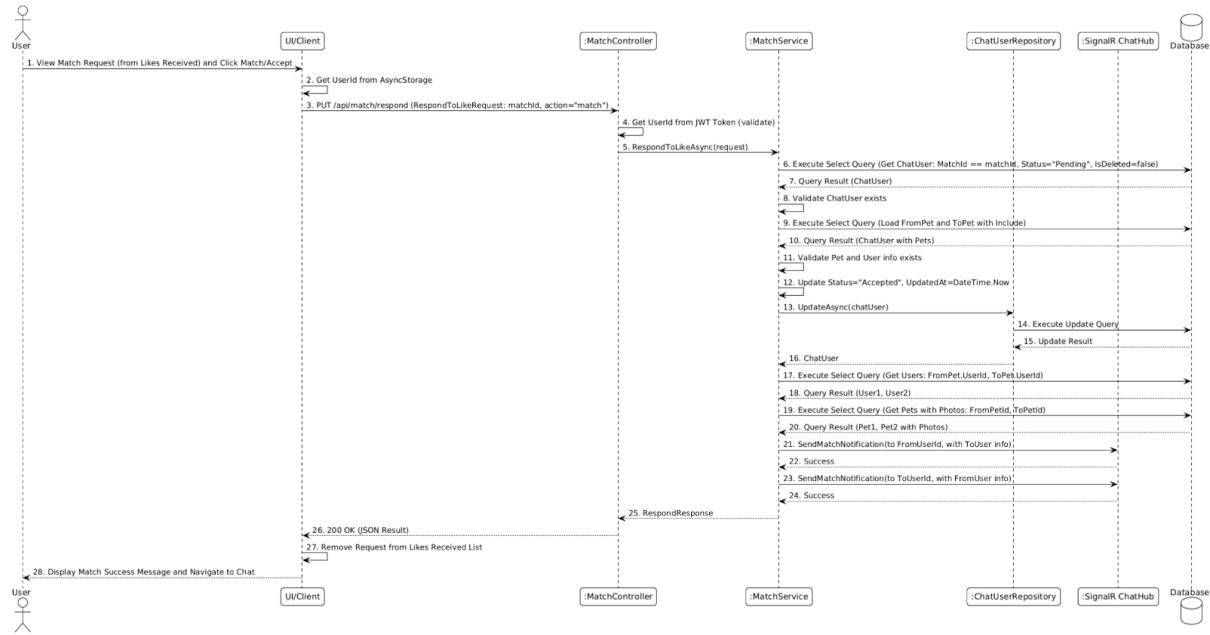
3.5.2.1 View Match List



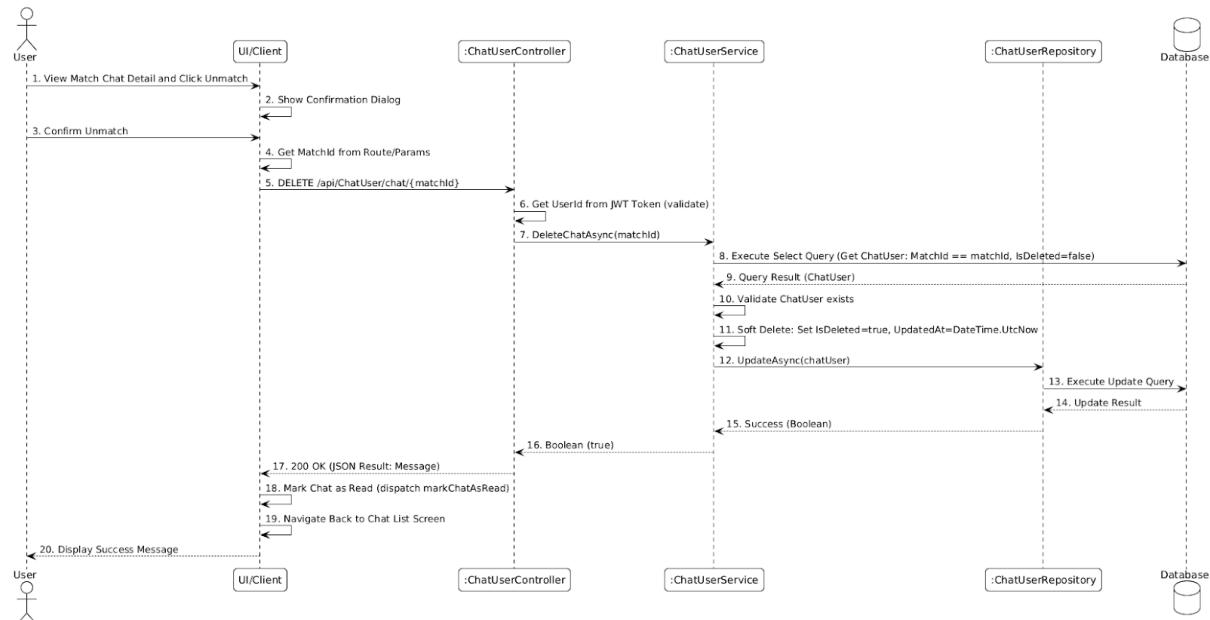
3.5.2.2 Chat with Matched User



3.5.2.3 Respond to Match Request

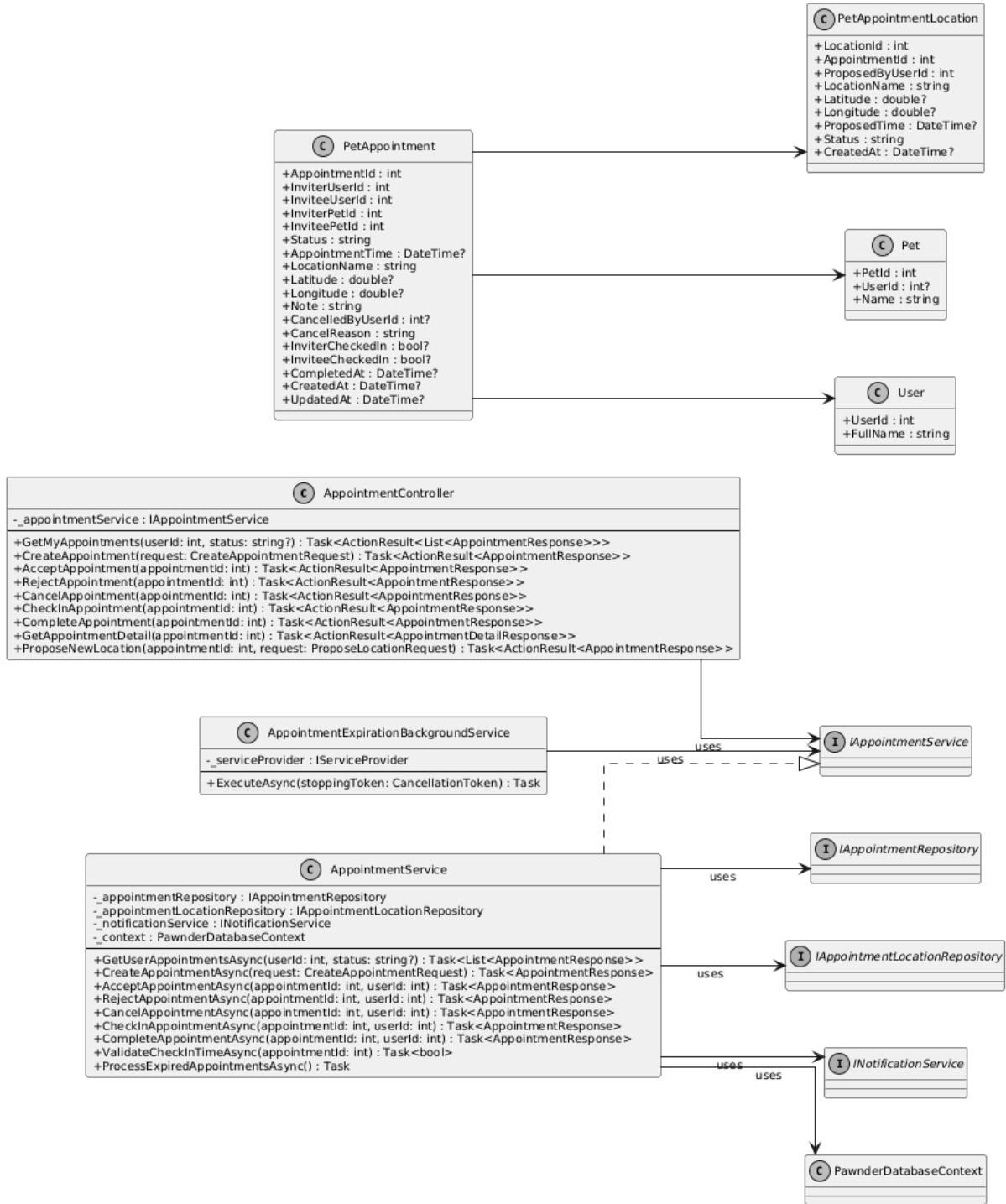


3.5.2.4 Unmatch User



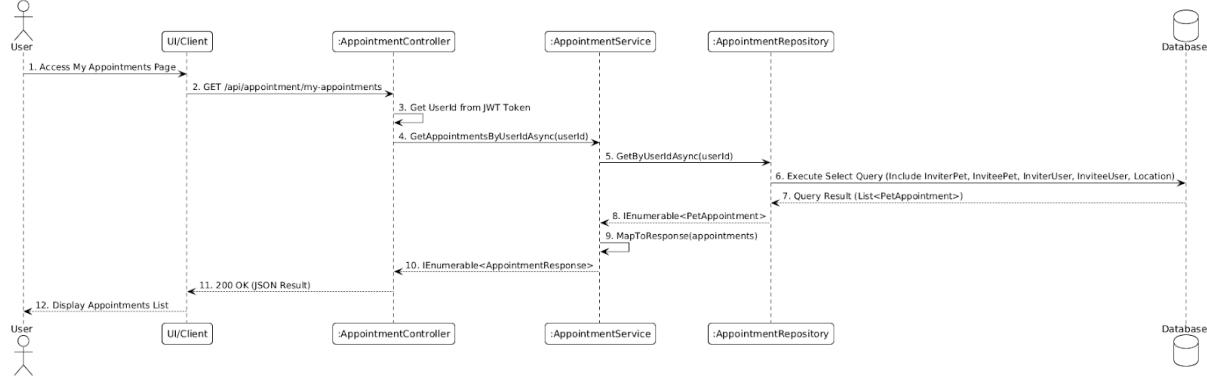
3.6 Appointment

3.6.1 Class Diagram

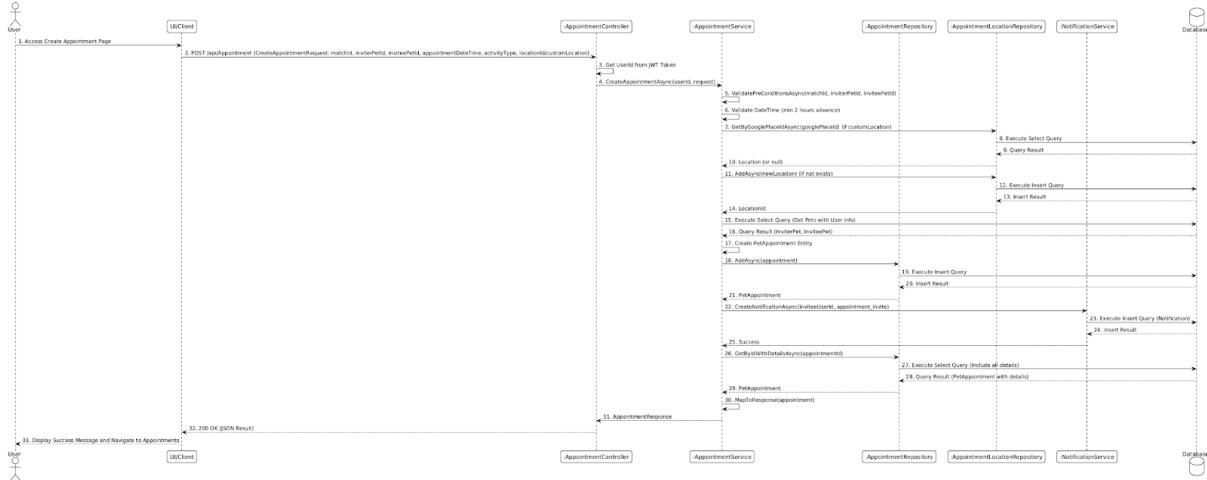


3.6.2 Sequence Diagram

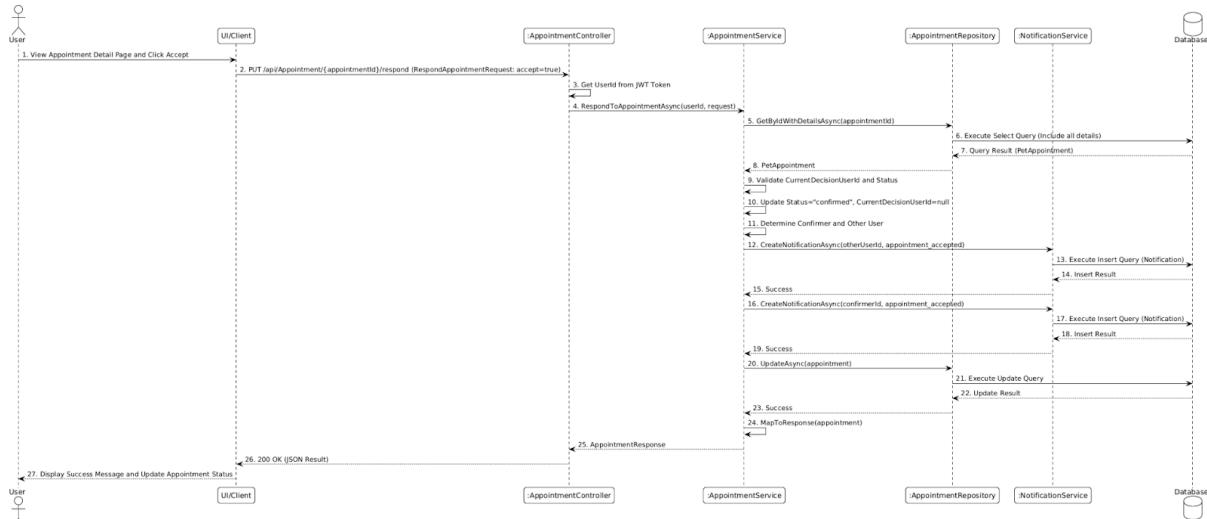
3.6.2.1 View My Appointment



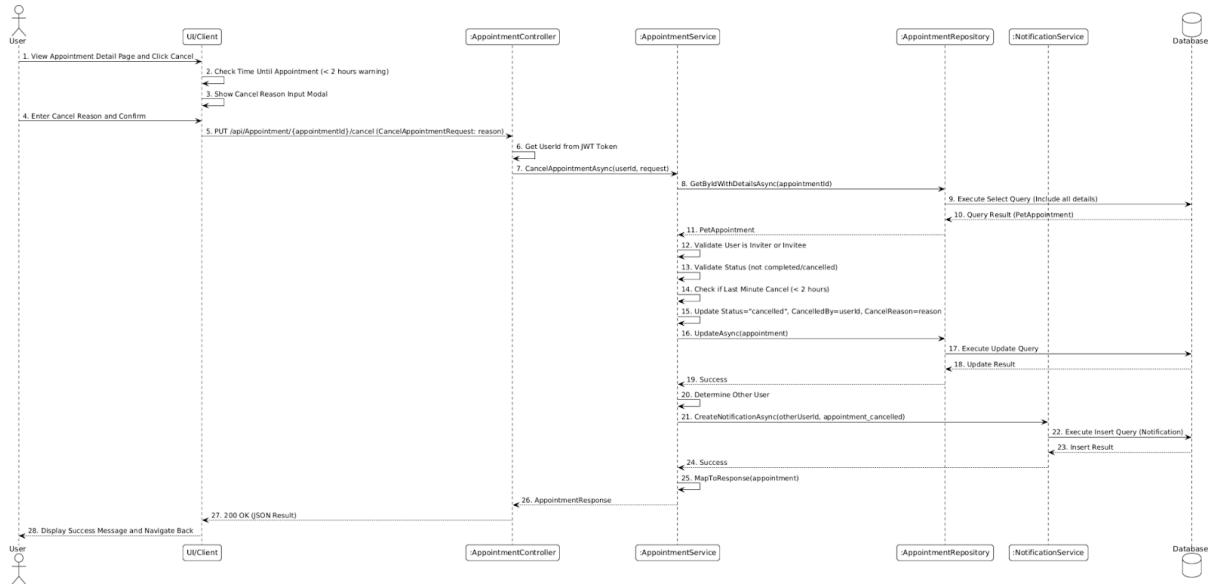
3.6.2.2 Create Appointment



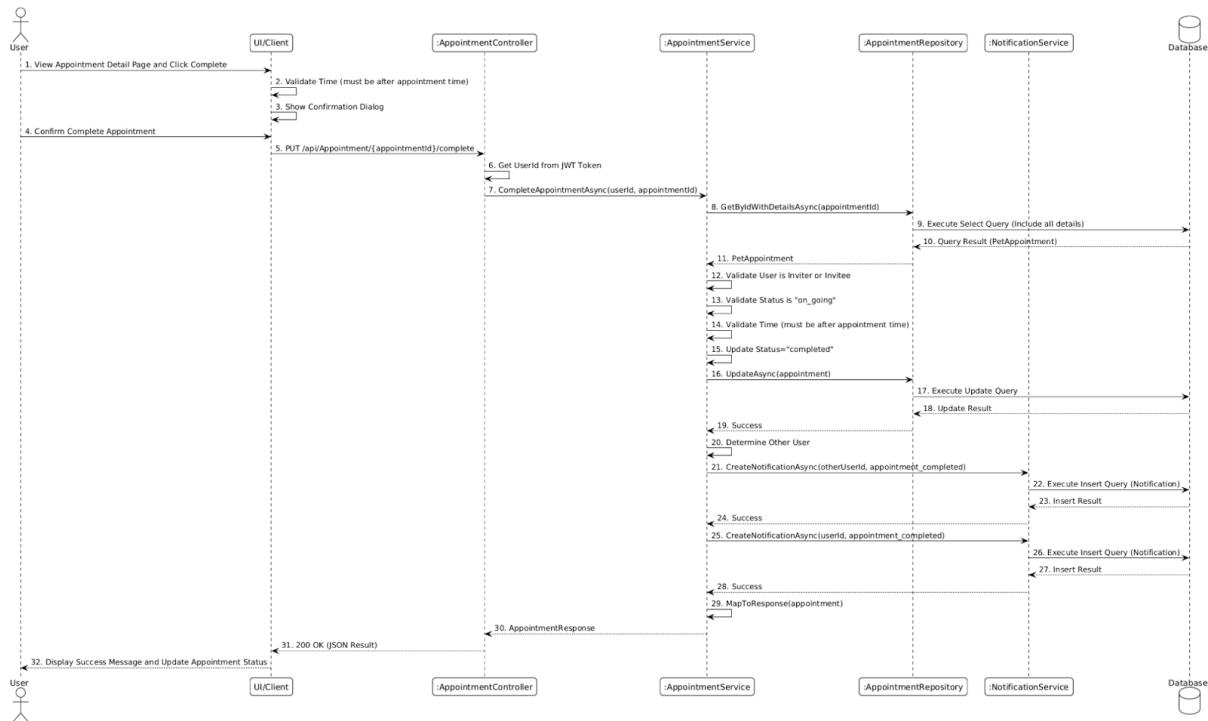
3.6.2.3 Accept Appointment



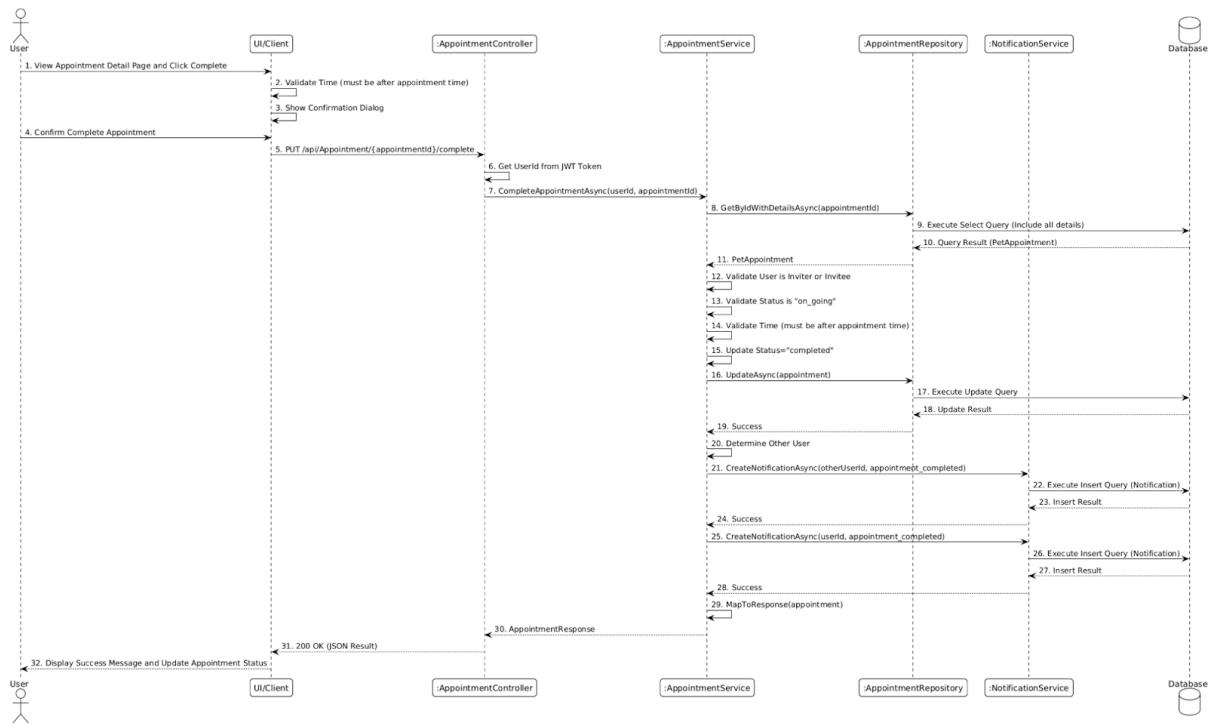
3.6.2.4 Cancel Appointment



3.6.2.5 Check-in at Appointment

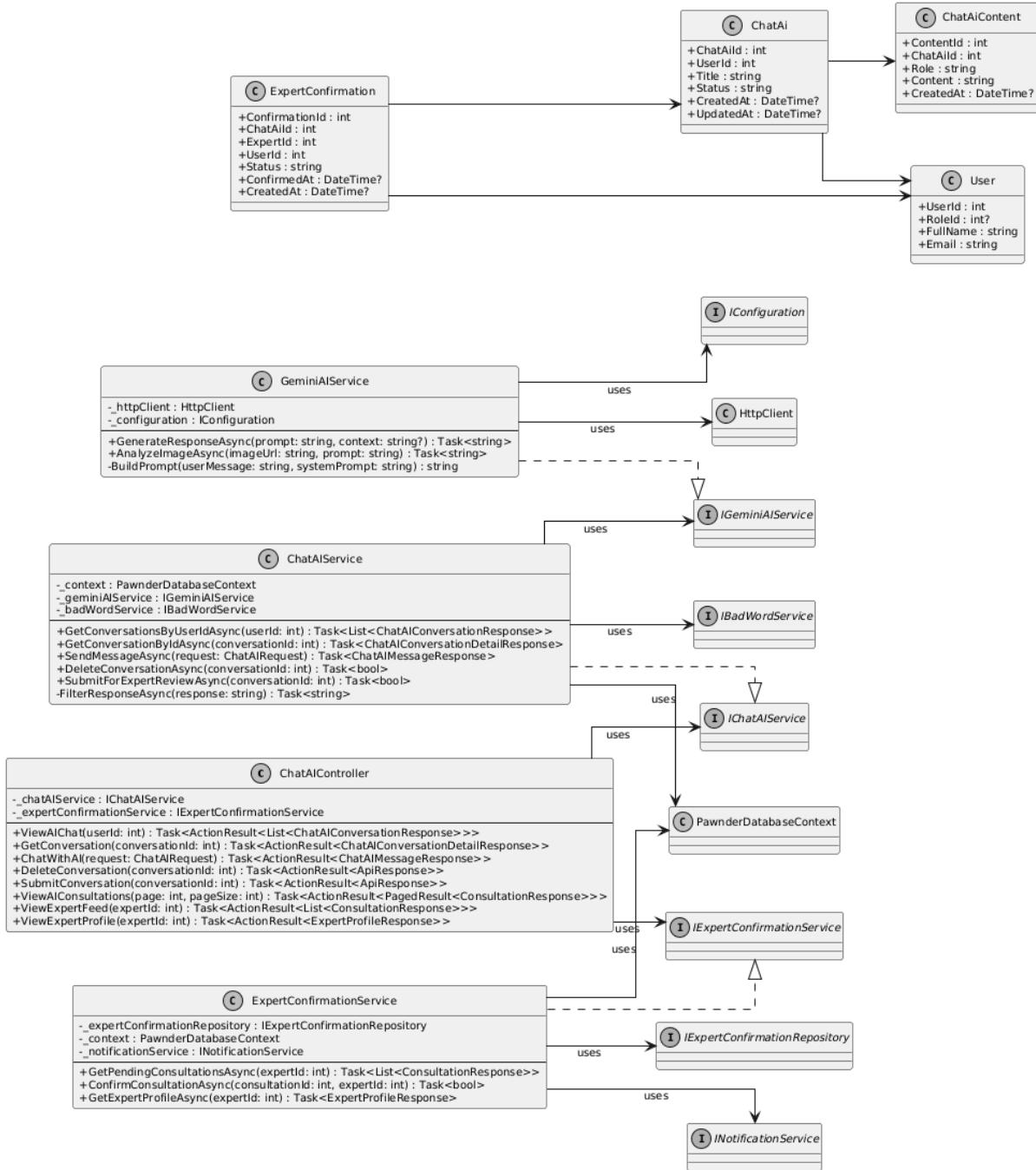


3.6.2.6 Complete Appointment



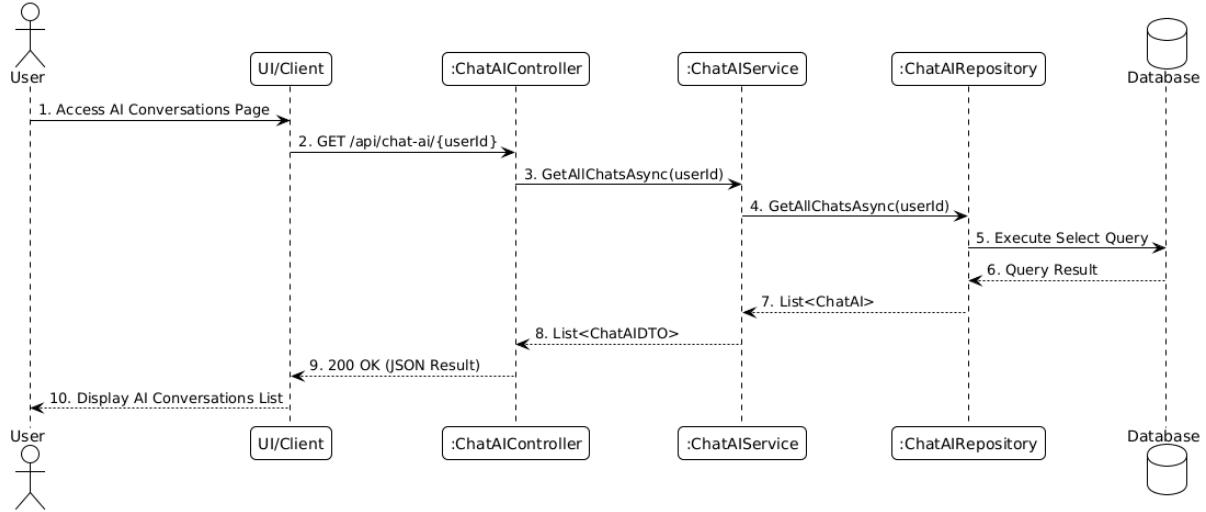
3.7 AI Consultation

3.7.1 Class Diagram

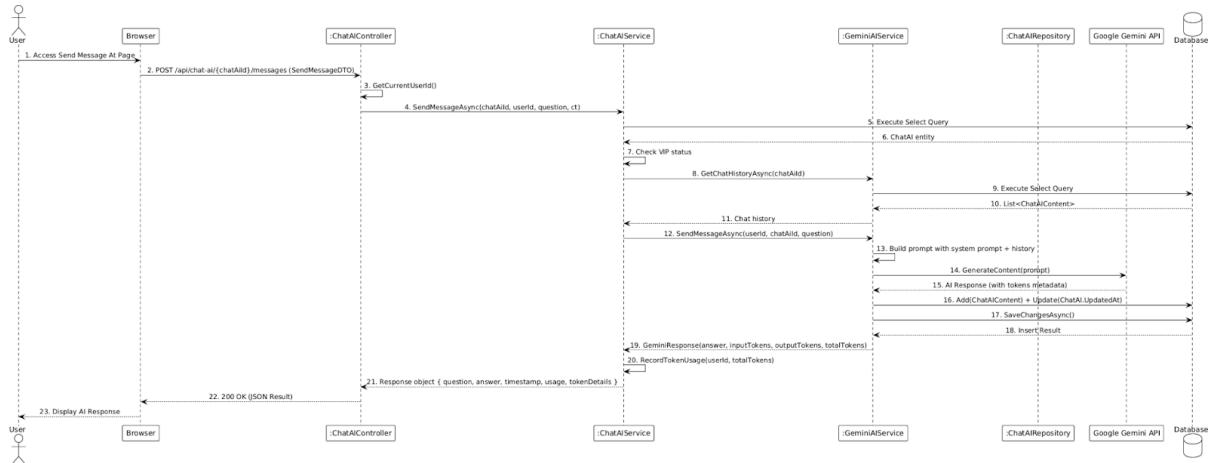


3.7.2 Sequence Diagram

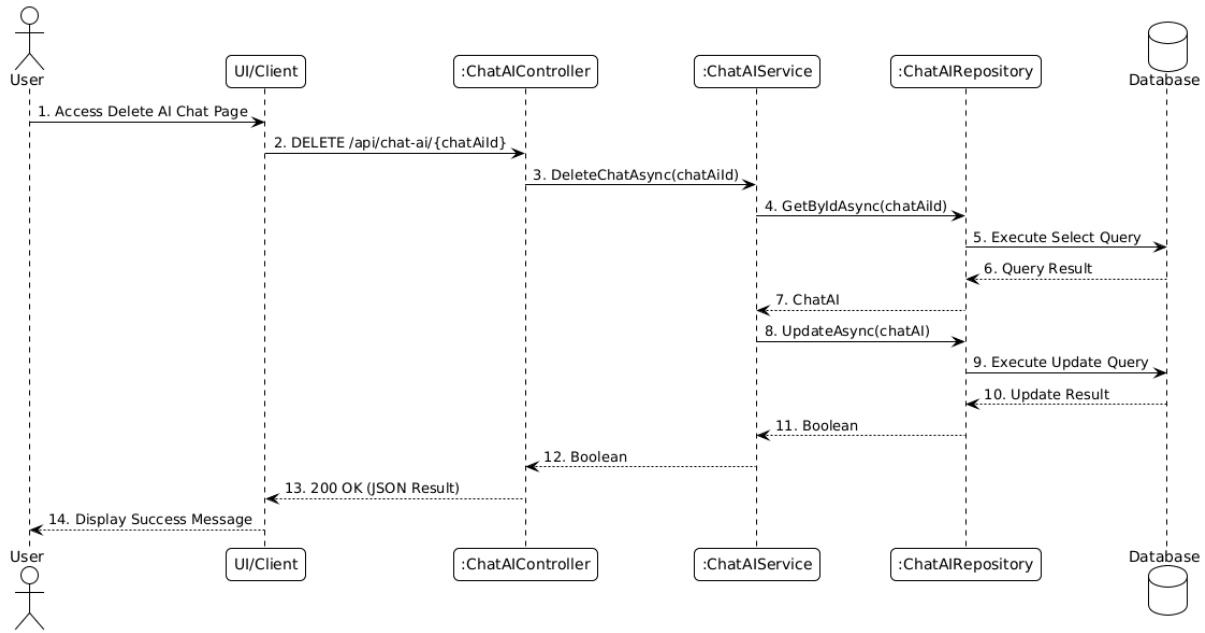
3.7.2.1 View AI Chat



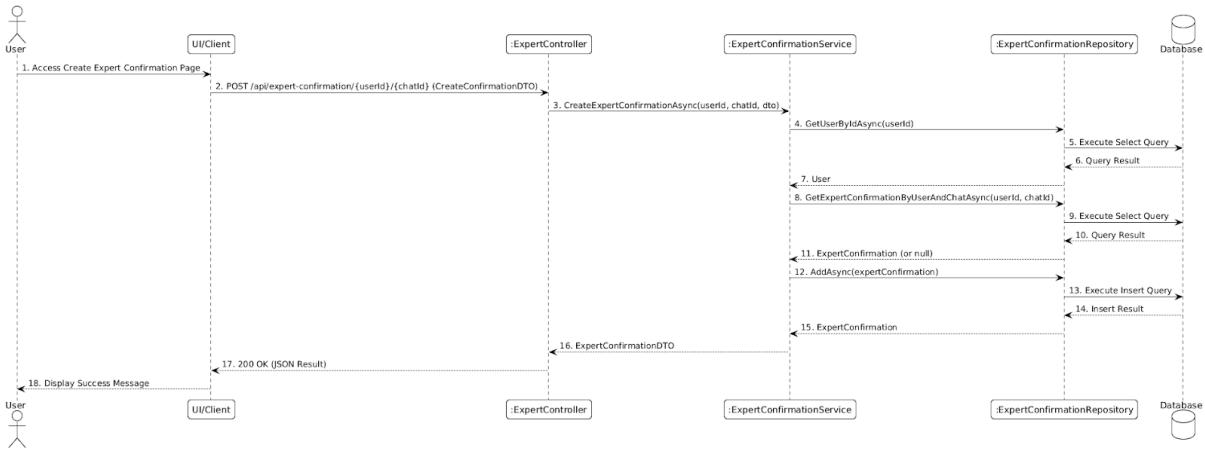
3.7.2.2 Chat with AI



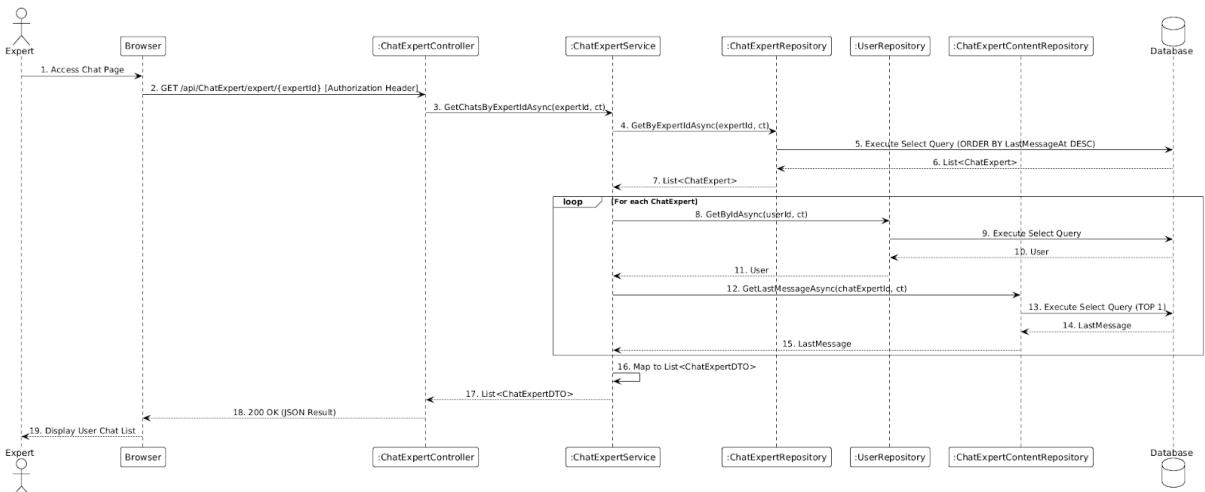
3.7.2.3 Delete AI Conversation



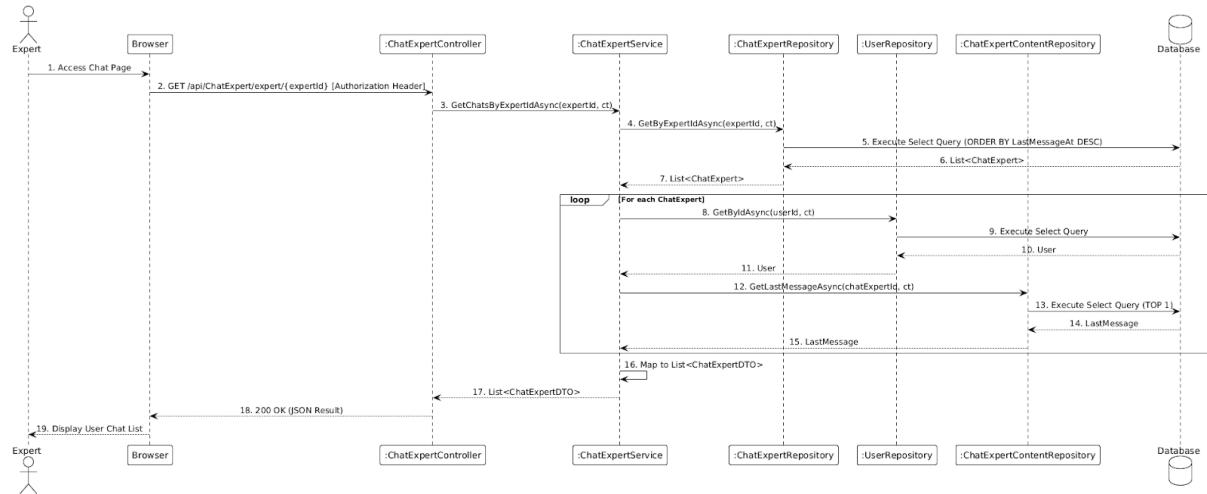
3.7.2.4 Submit AI Conversation



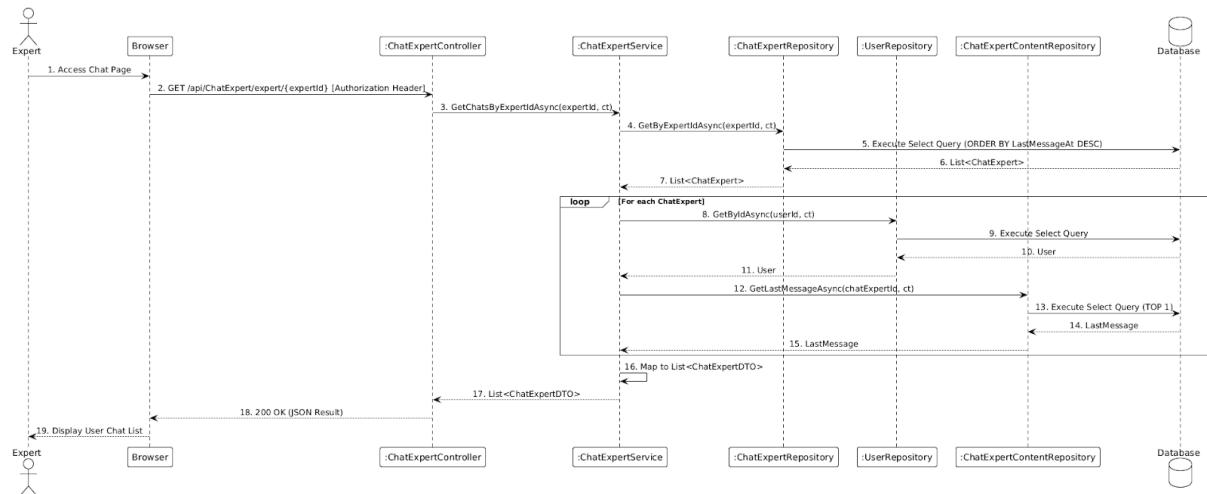
3.7.2.5 View AI Consultation



3.7.2.6 View Expert Feed

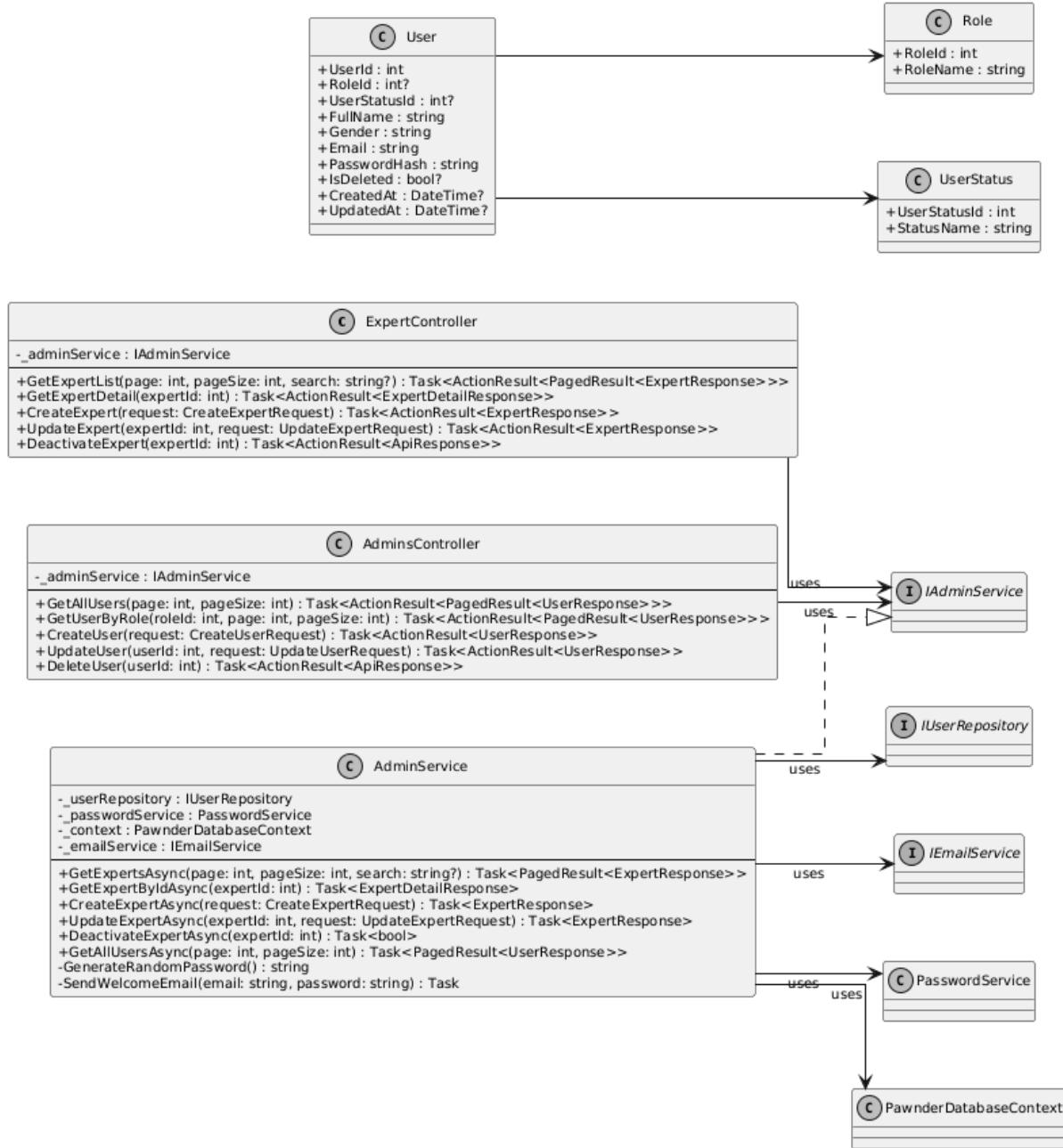


3.7.2.7 View Expert Profile



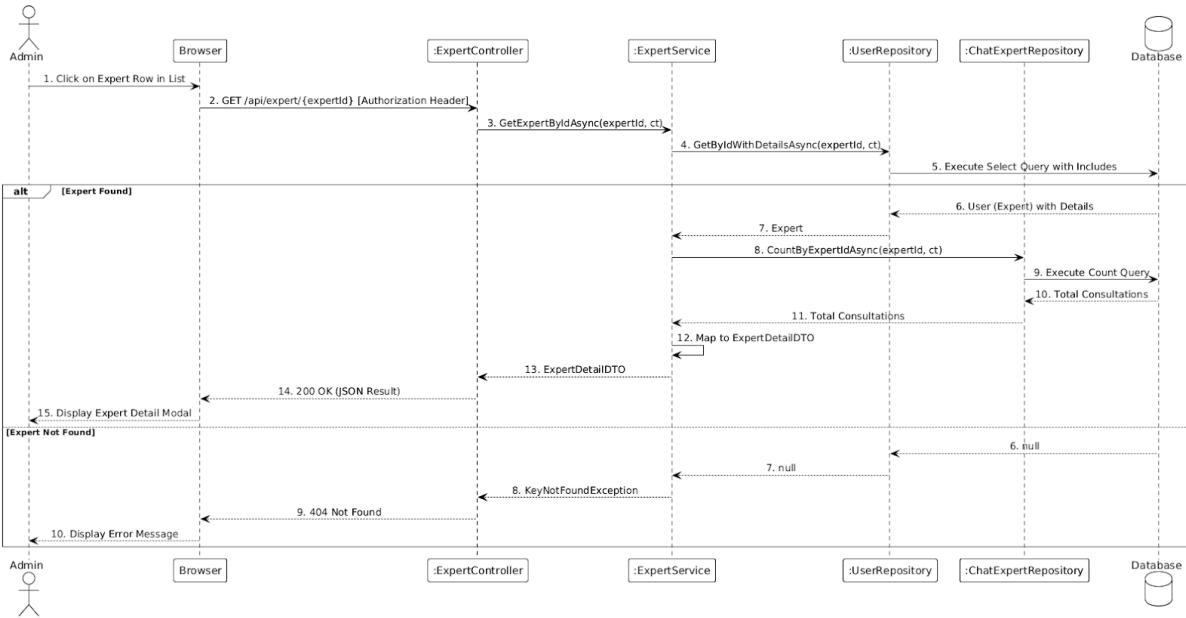
3.8 Expert Management

3.8.1 Class Diagram

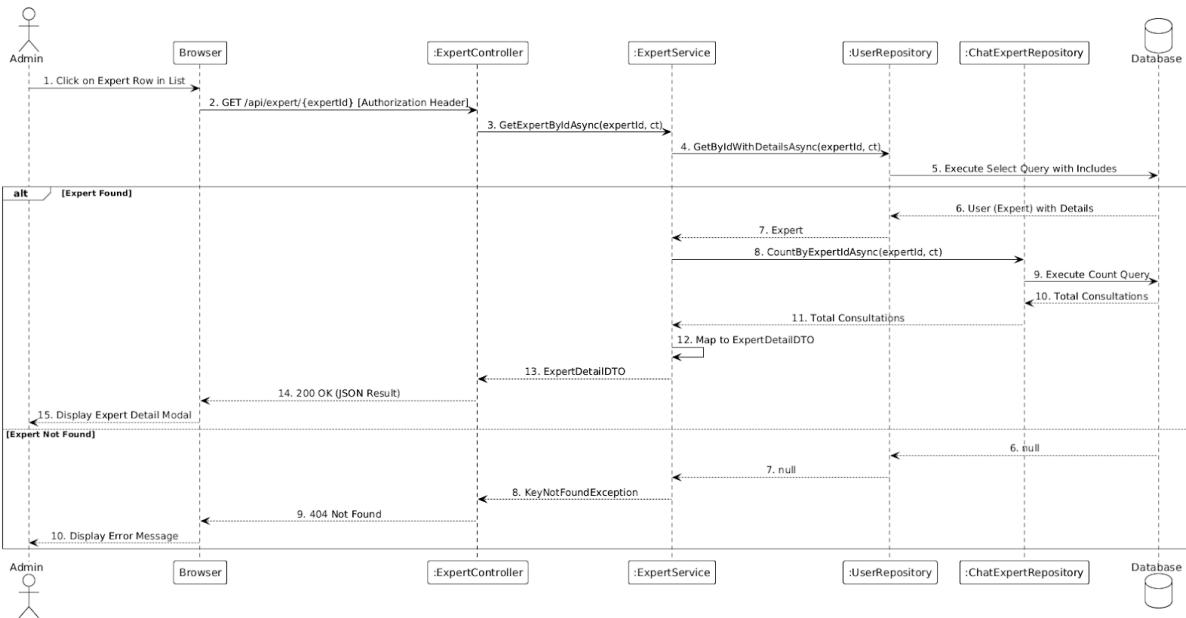


3.8.2 Sequence Diagram

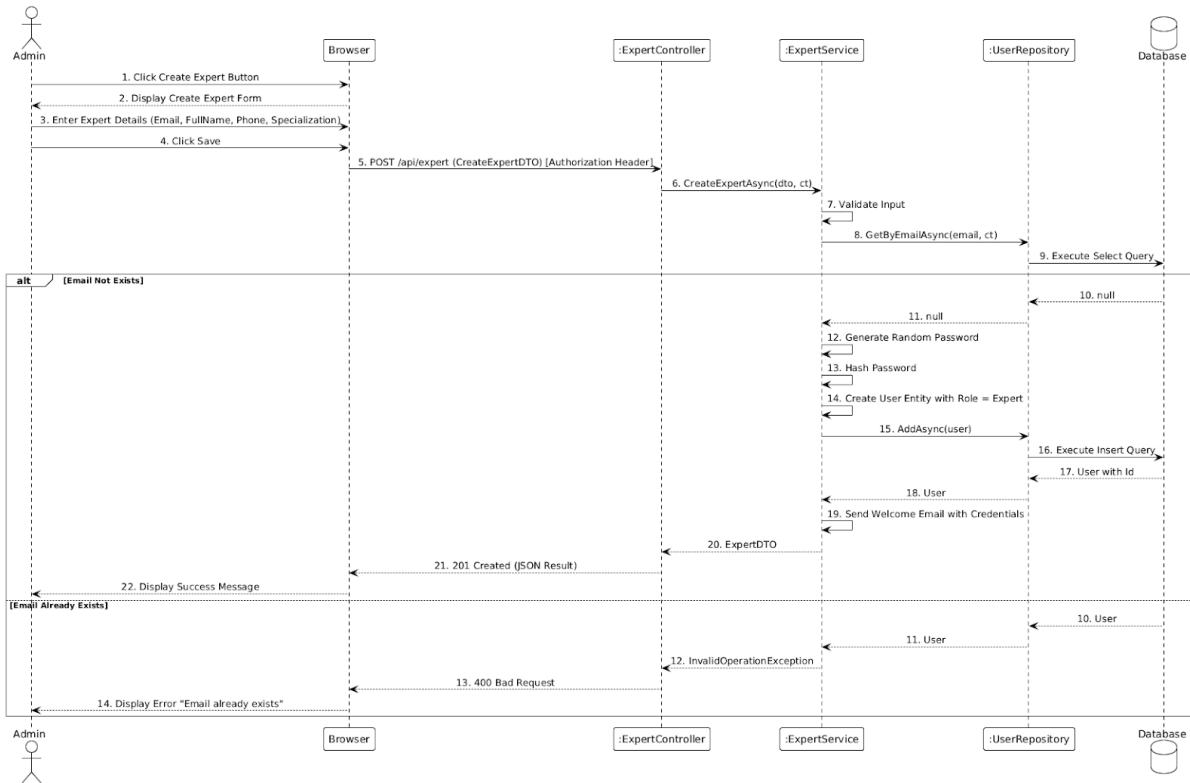
3.8.2.1 View Expert List



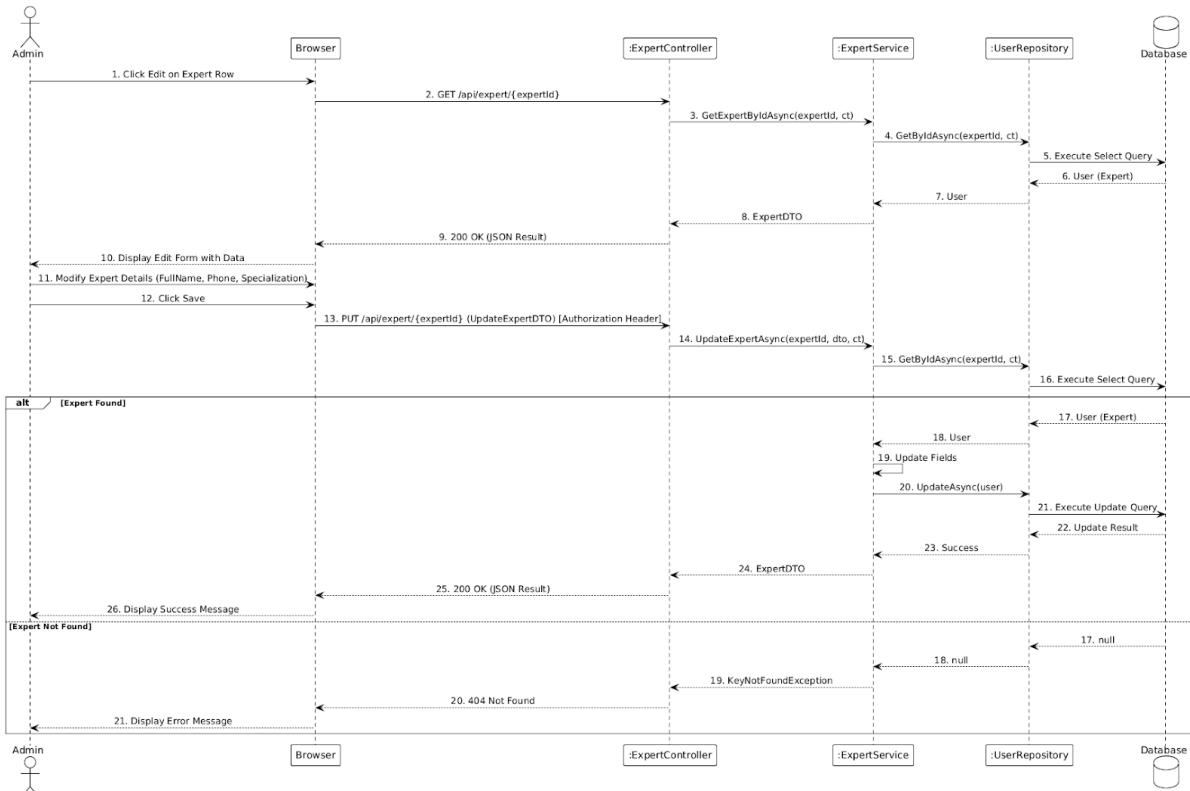
3.8.2.2 View Expert Detail



3.8.2.3 Create Expert Account



3.8.2.4 Edit Expert Account



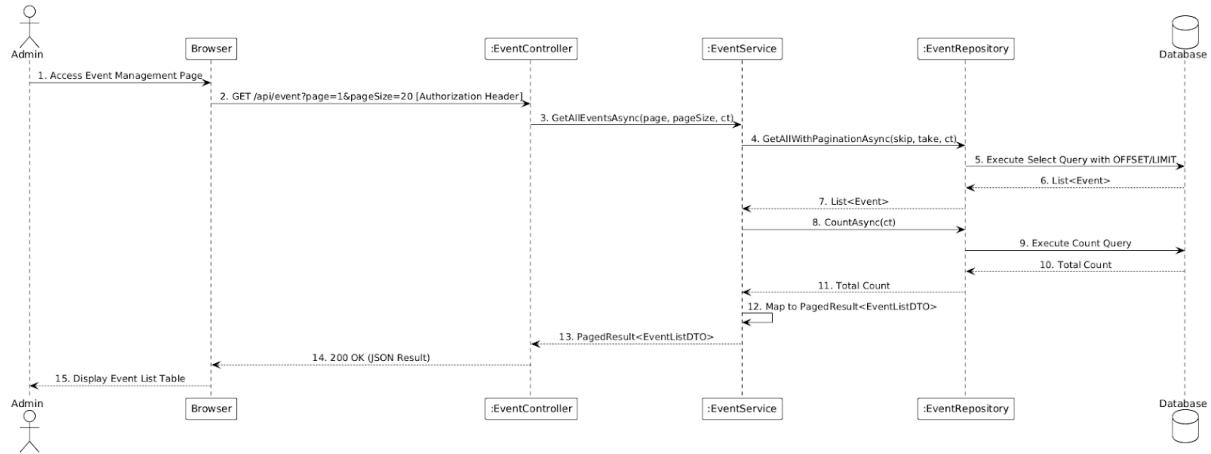
3.9 Event Management

3.9.1 Class Diagram

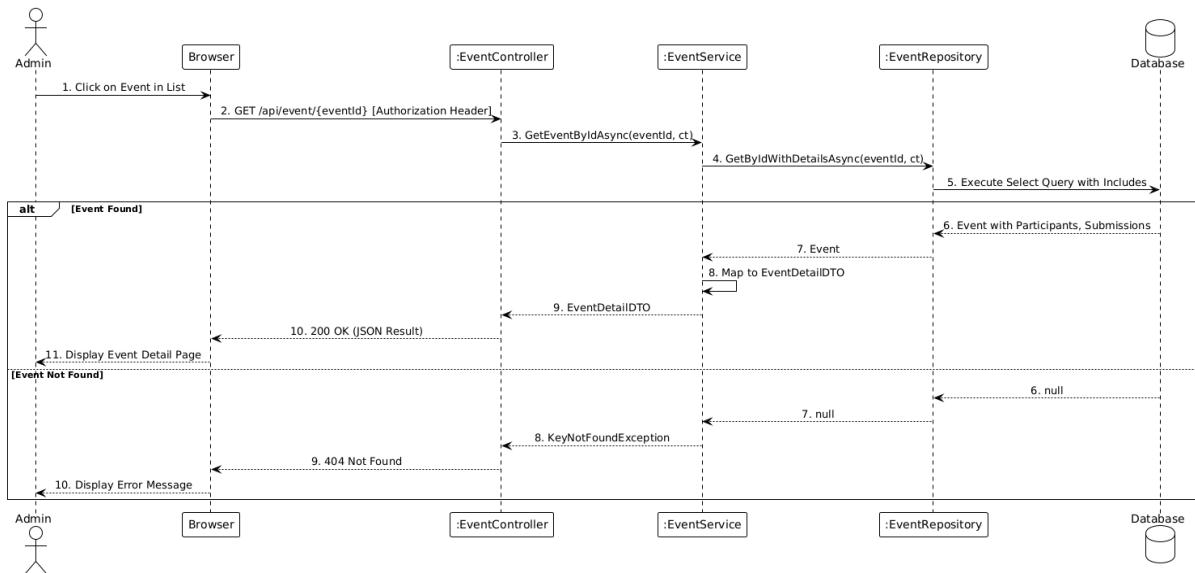


3.9.2 Sequence Diagram

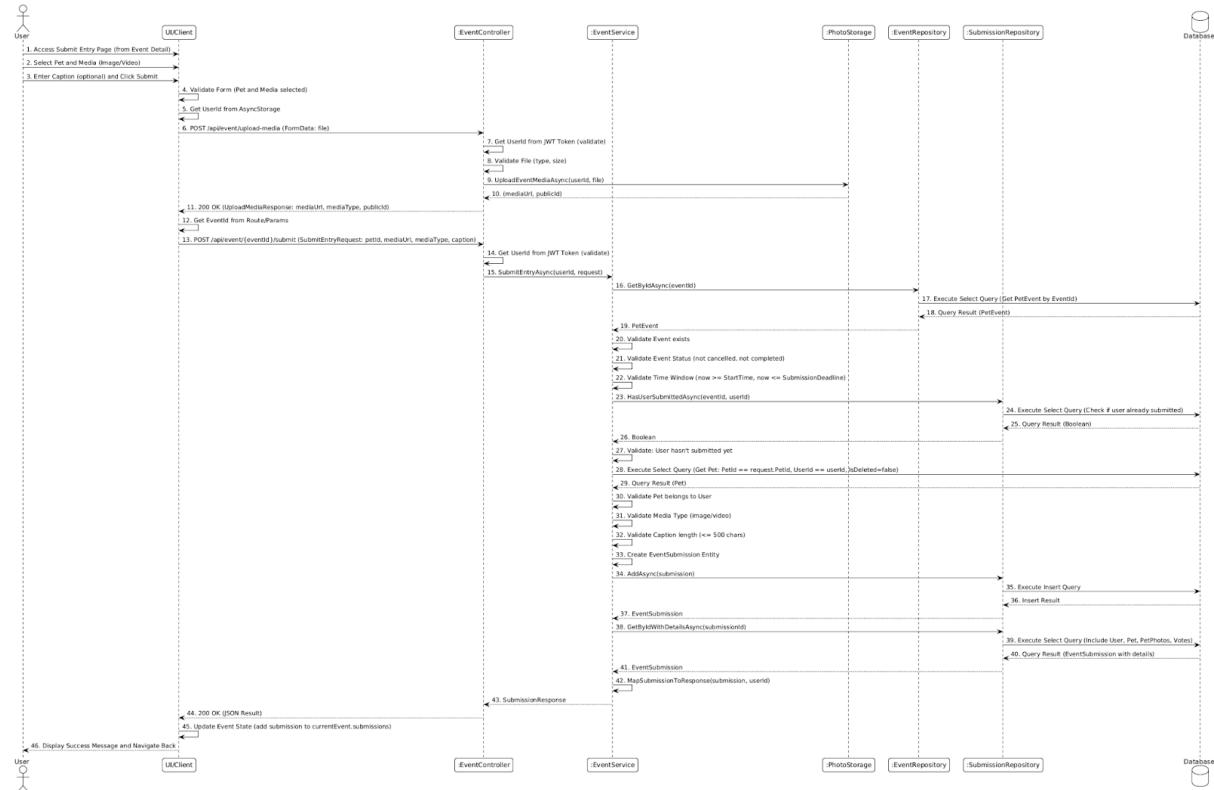
3.9.2.1 View Event List



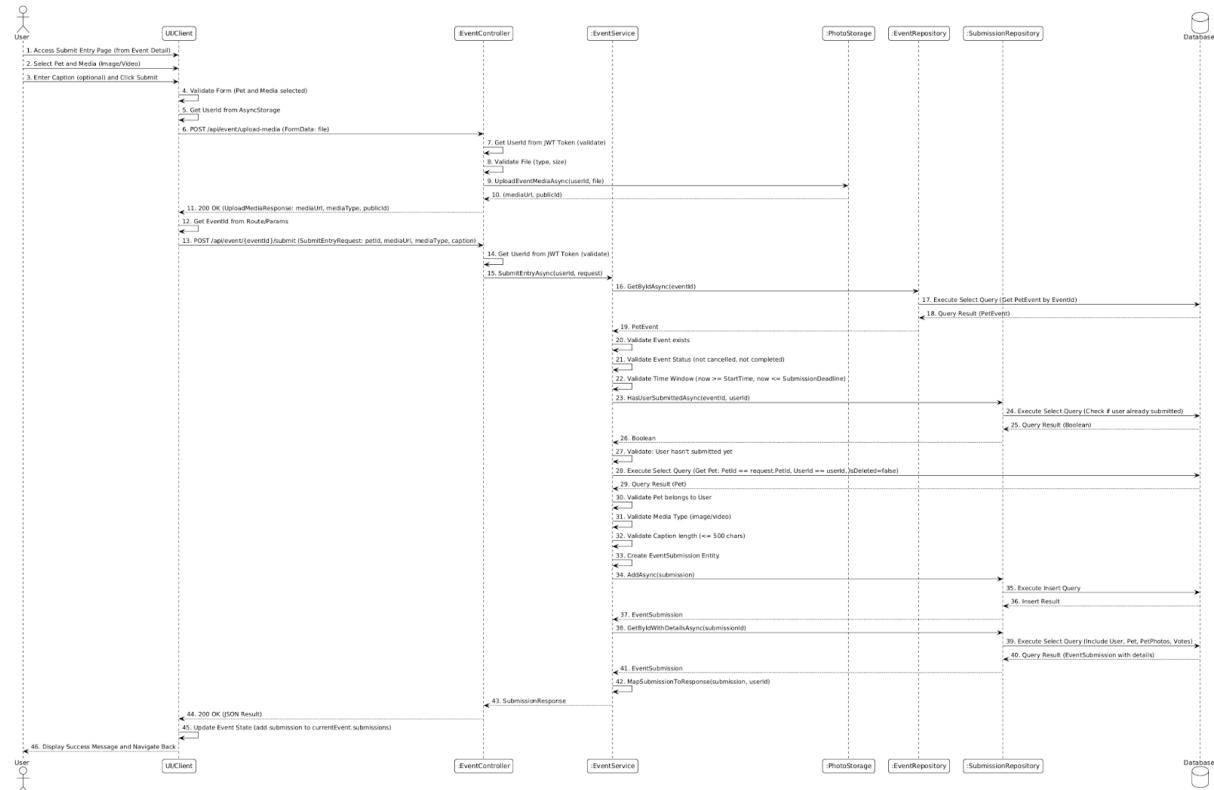
3.9.2.2 View Event Detail



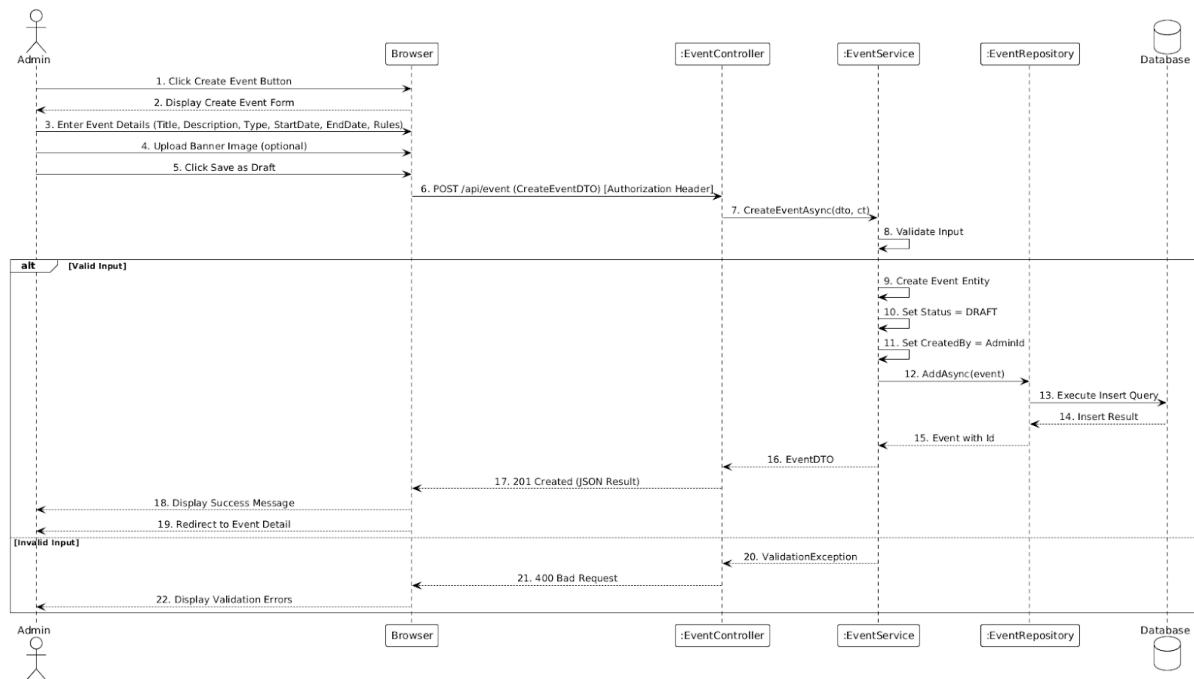
3.9.2.3 Submit Event Entry



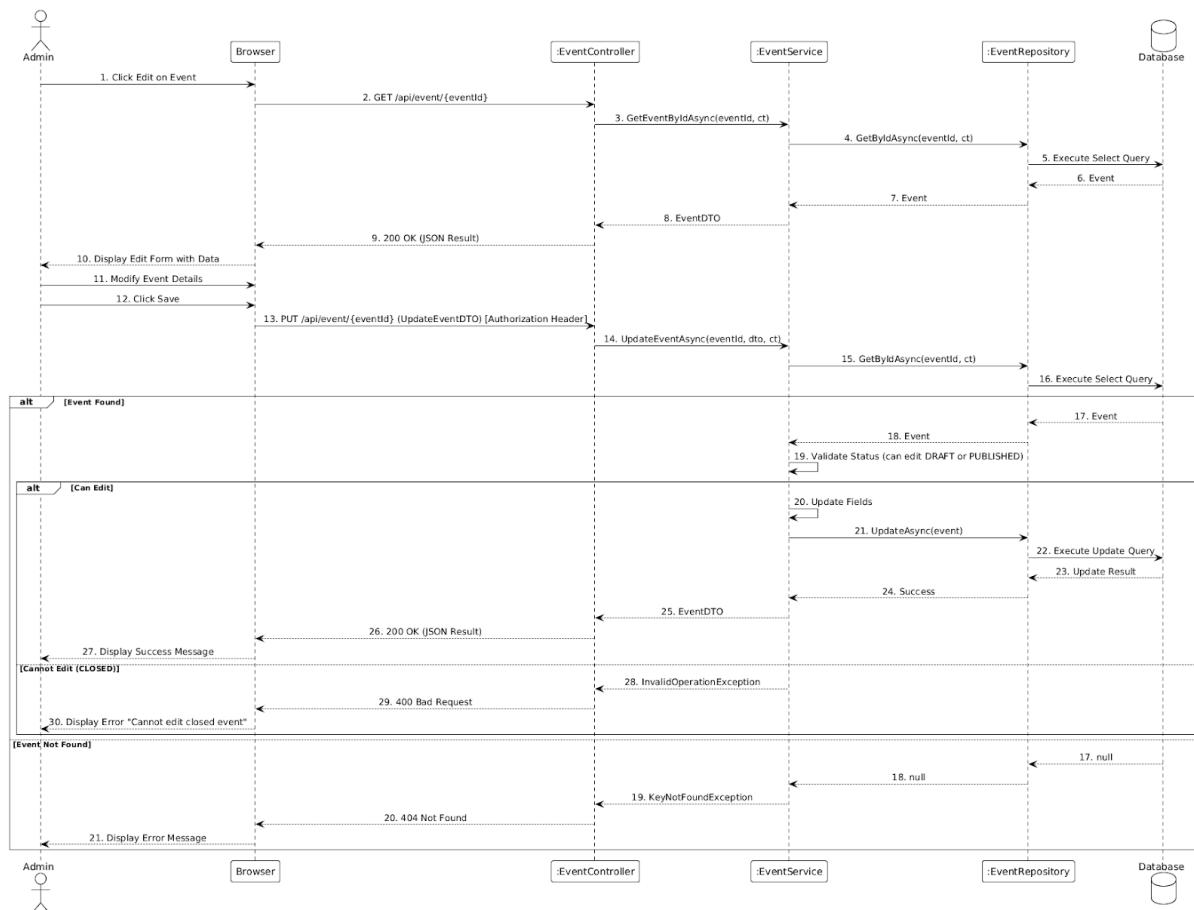
3.9.2.4 Vote / Unvote for Event



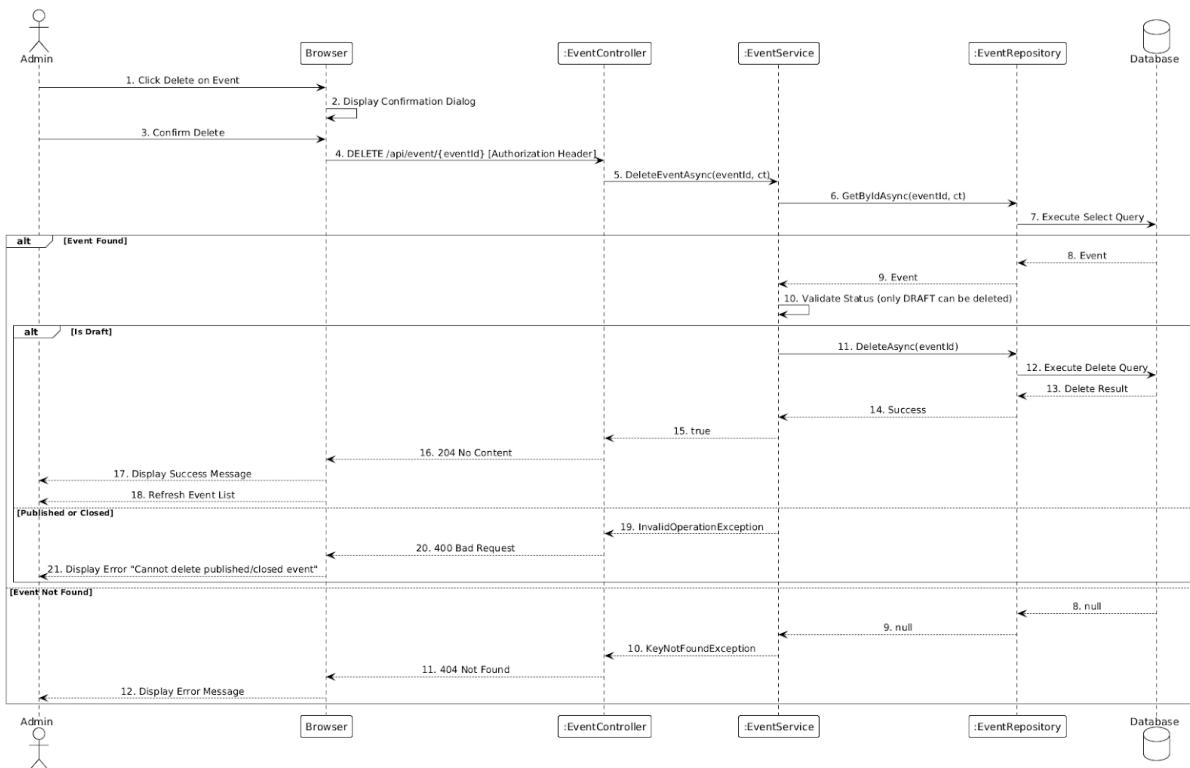
3.9.2.5 Create Event



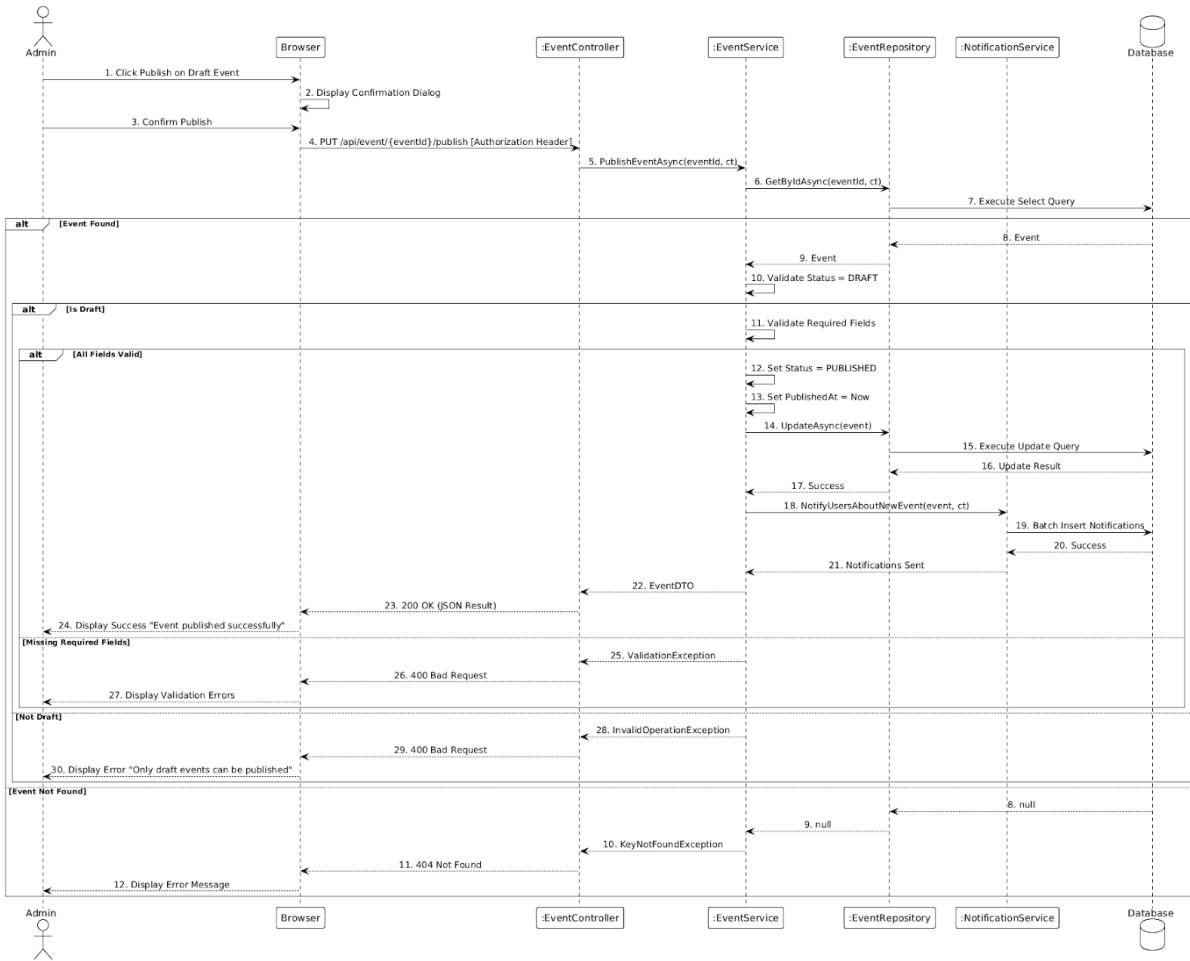
3.9.2.6 Edit Event



3.9.2.7 Delete Event

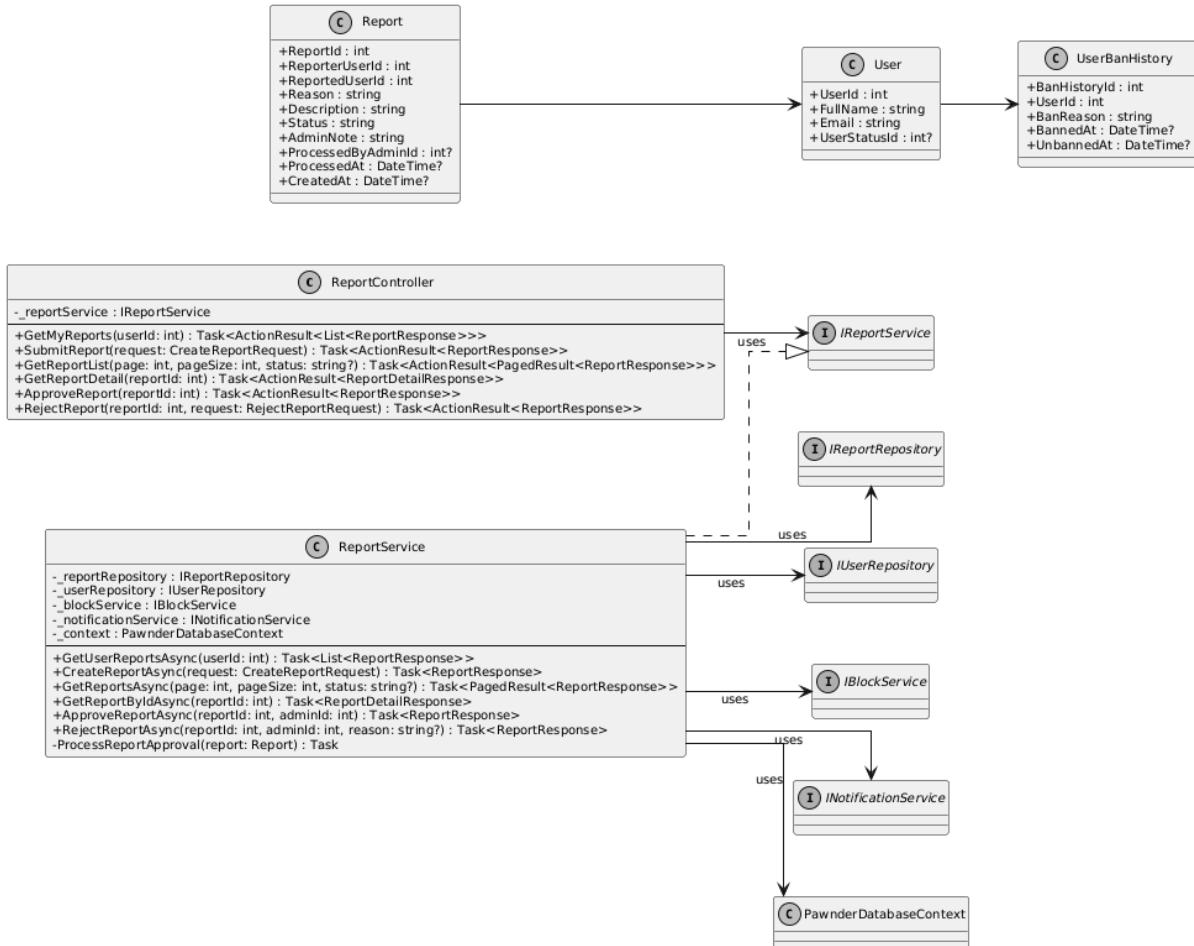


3.9.2.8 Publish Event



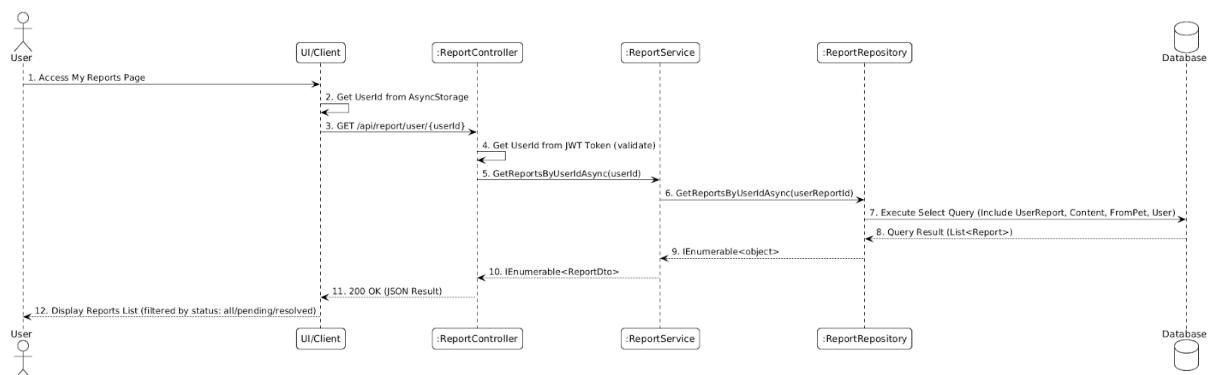
3.10 Report Management

3.10.1 Class Diagram

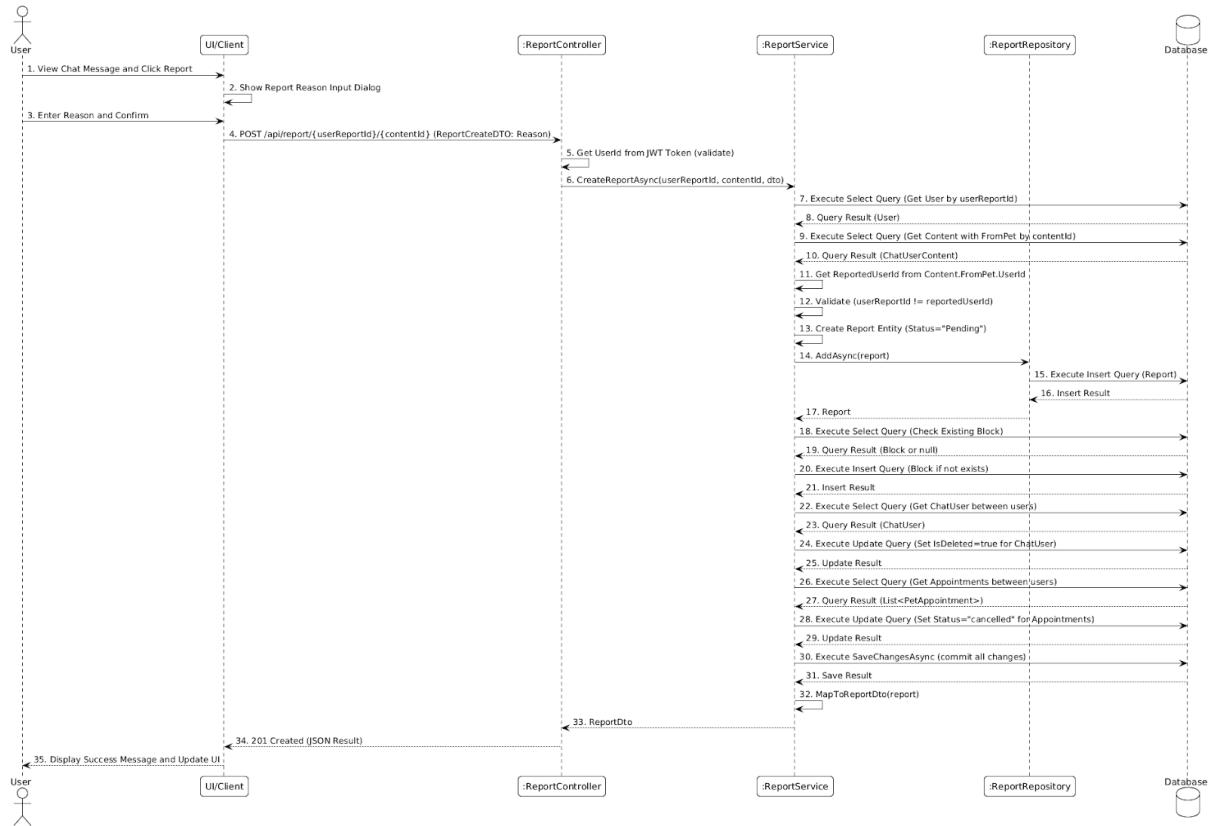


3.10.2 Sequence Diagram

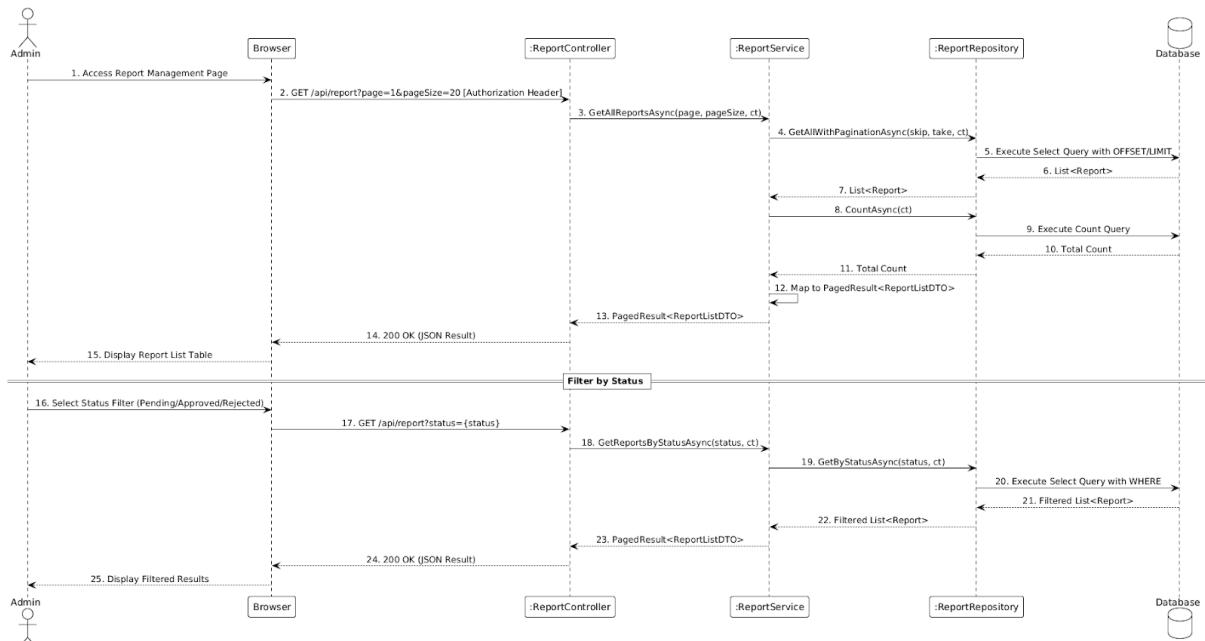
3.10.2.1 View My Report



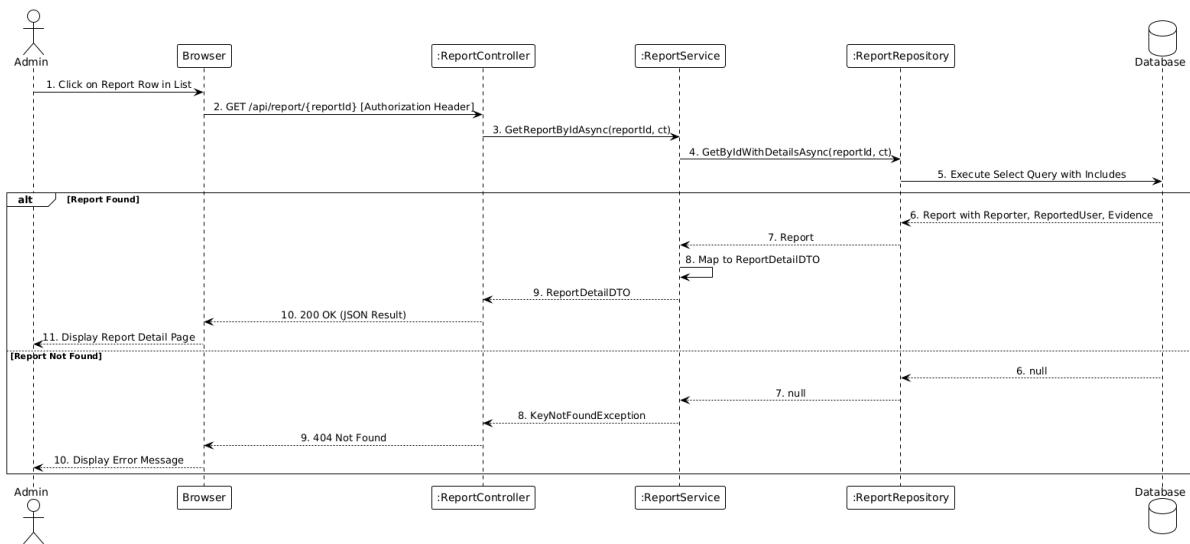
3.10.2.2 Submit Report



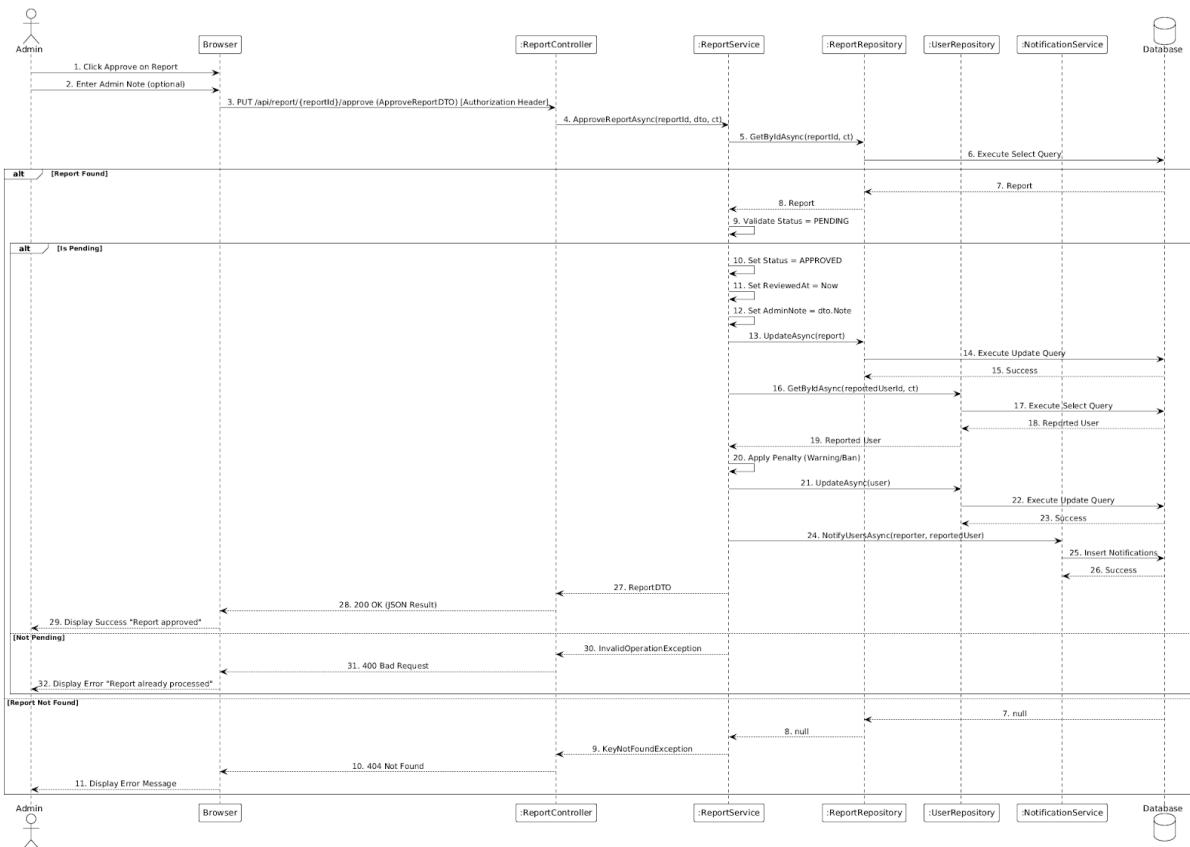
3.10.2.3 View Report List



3.10.2.4 View Report Detail

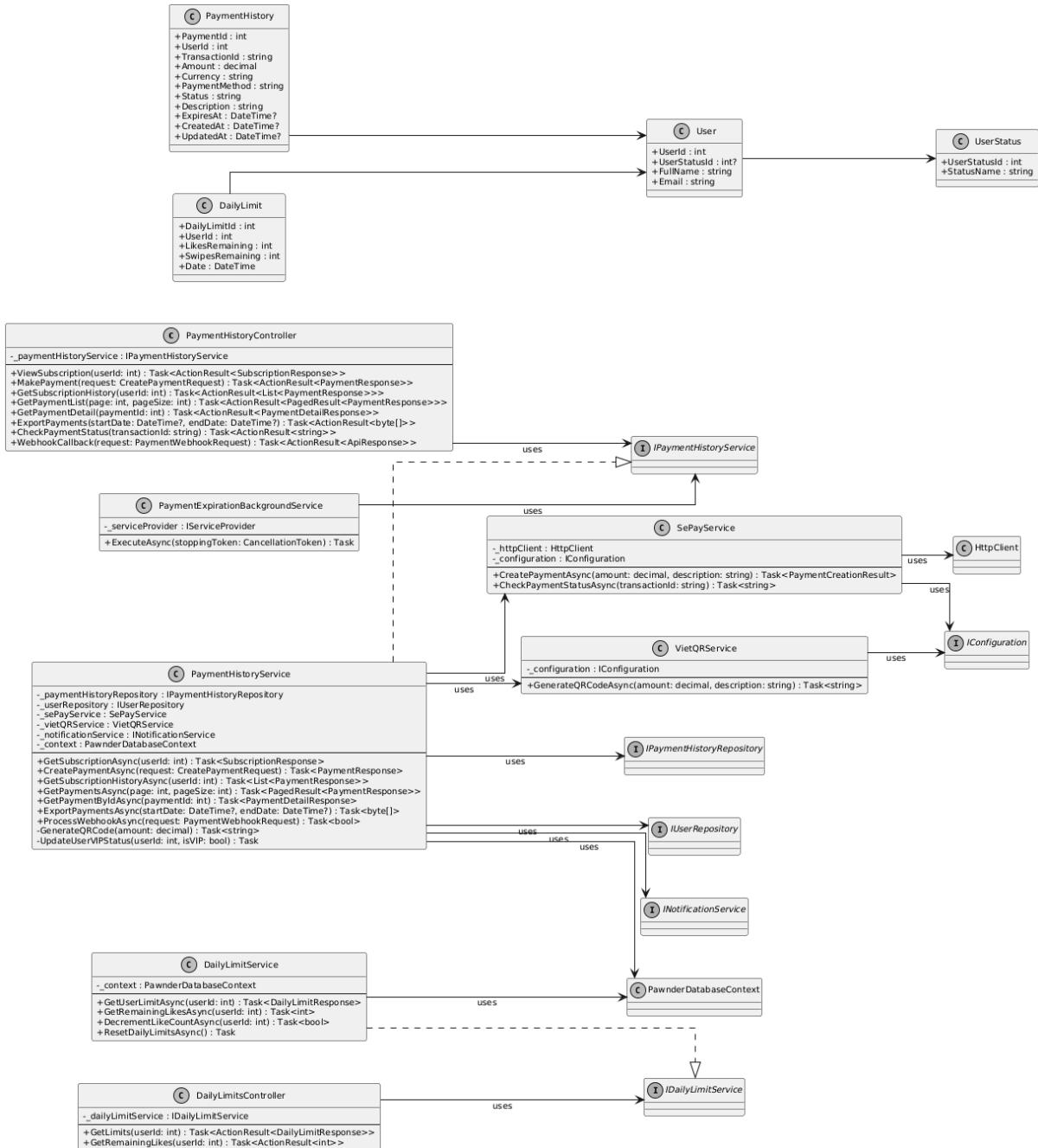


3.10.2.5 Approve Report



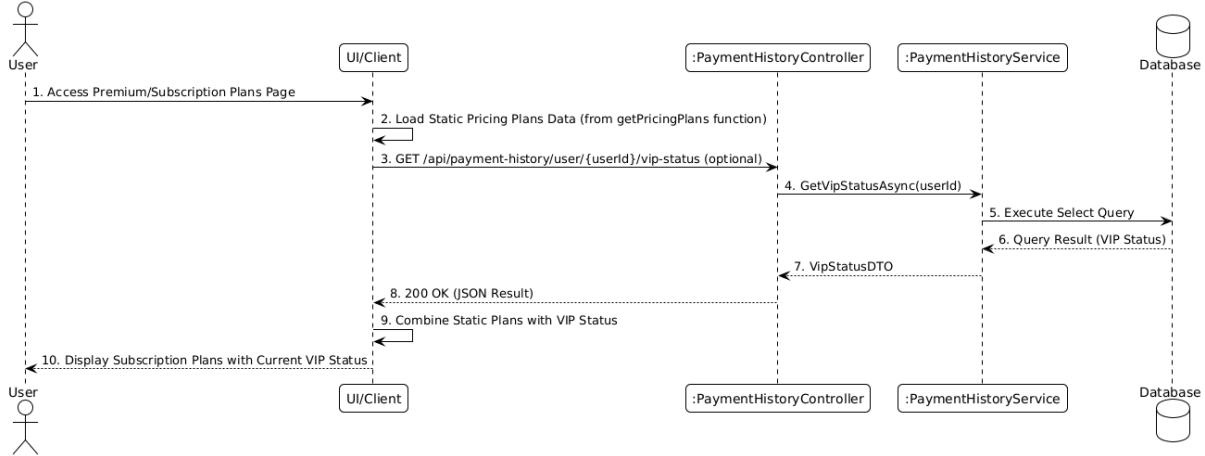
3.11 Subscription & Payment

3.11.1 Class Diagram

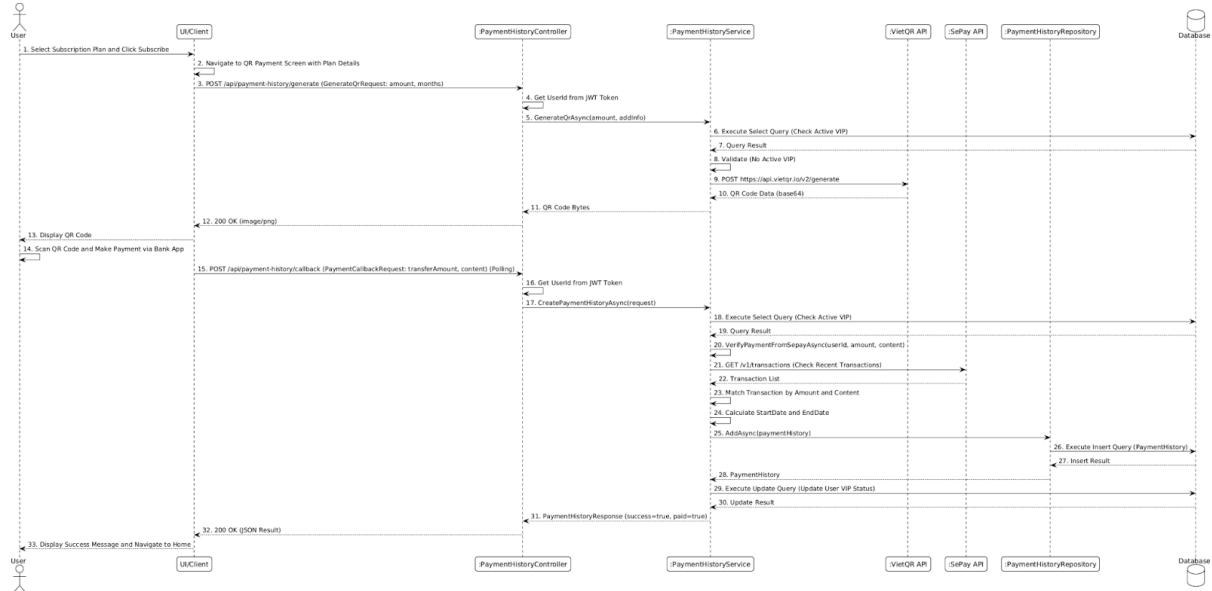


3.11.2 Sequence Diagram

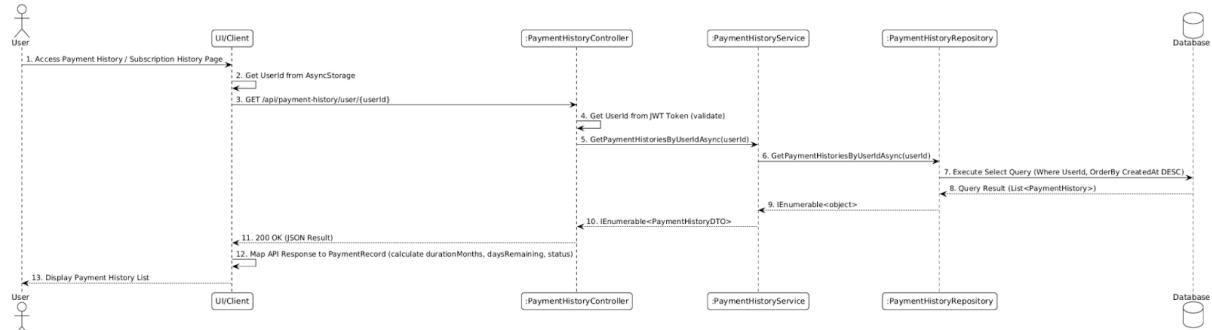
3.11.2.1 View Subscription



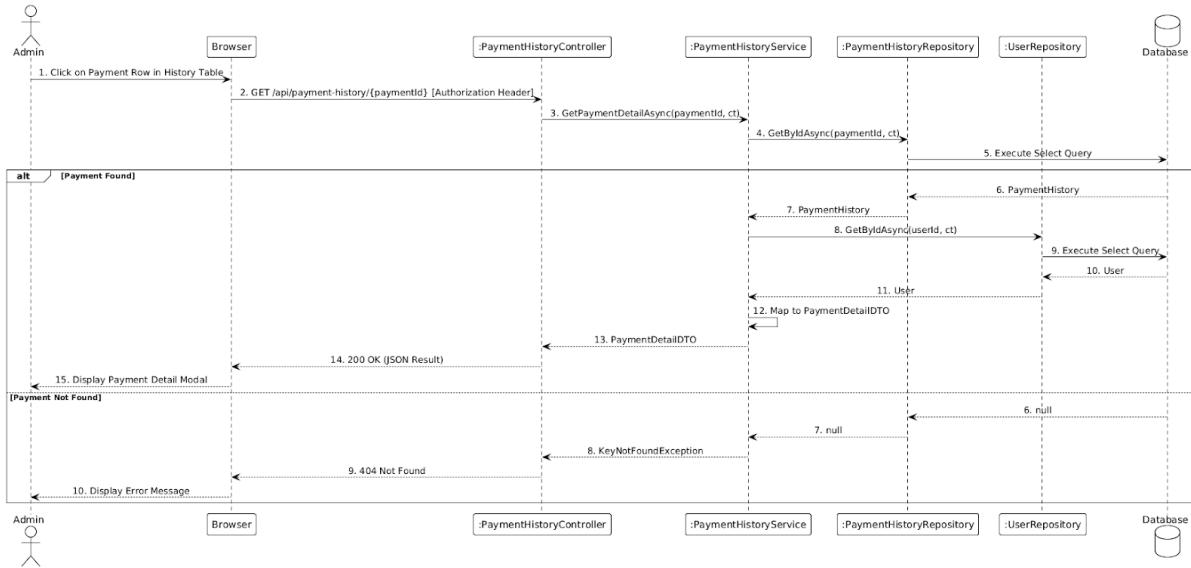
3.11.2.2 Make Payment



3.11.2.3 View Subscription History

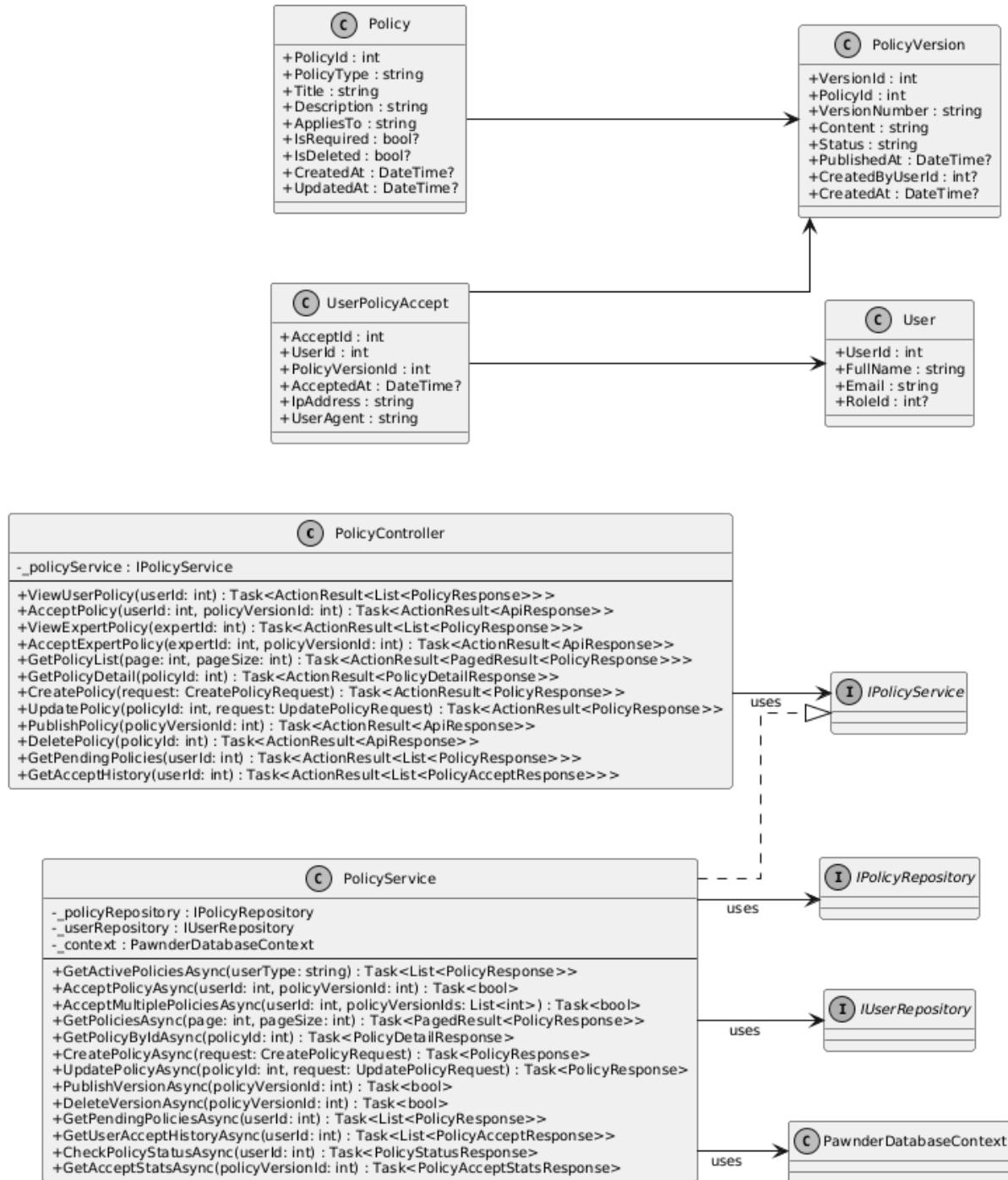


3.11.2.4 View Payment Detail



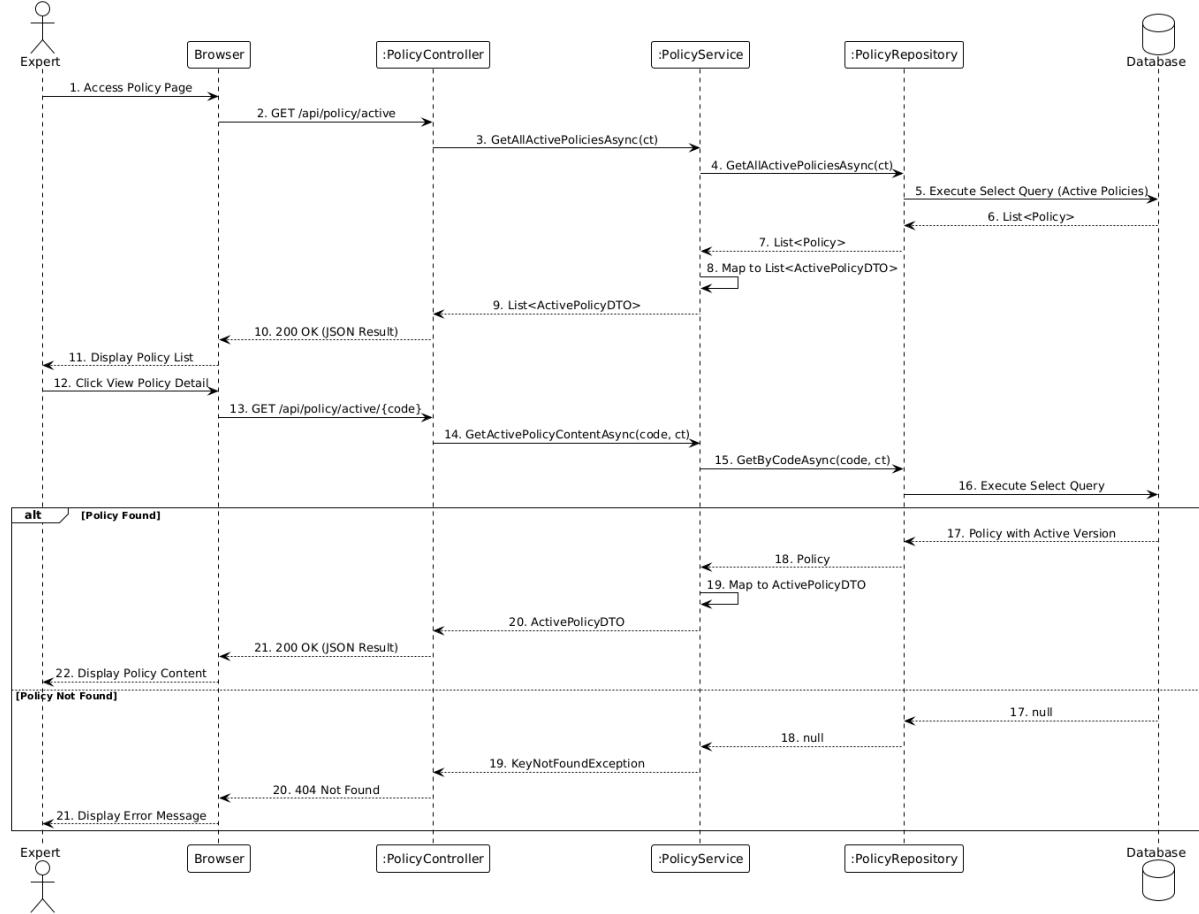
3.12 Policy Management

3.12.1 Class Diagram

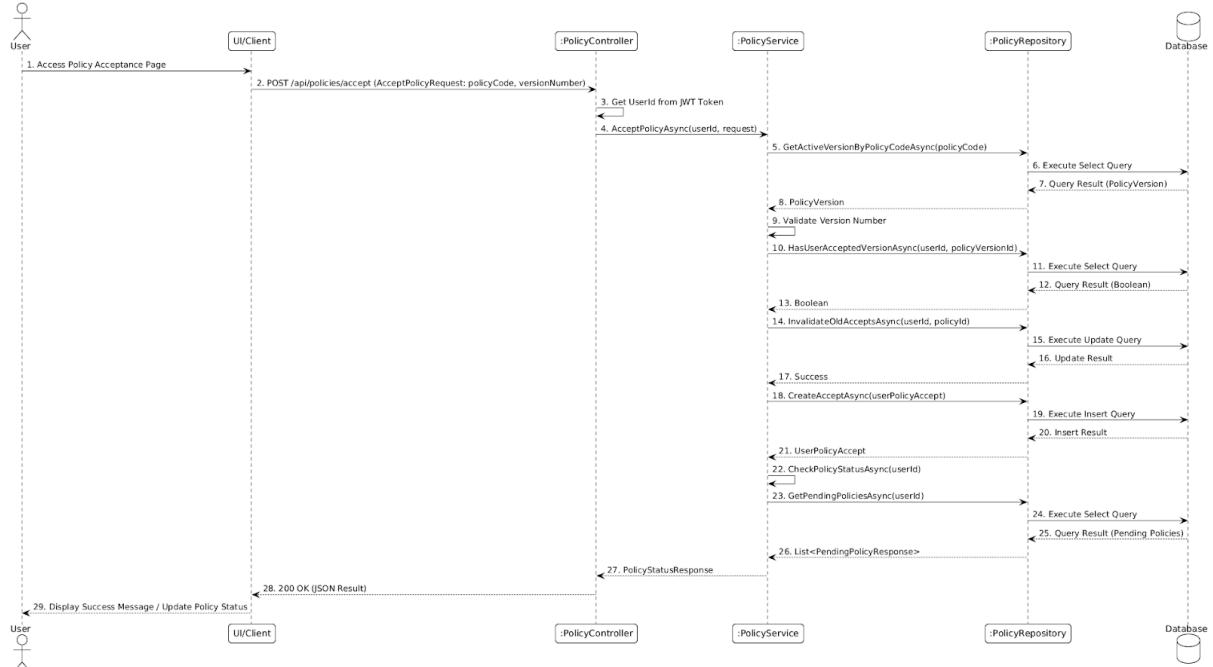


3.12.2 Sequence Diagram

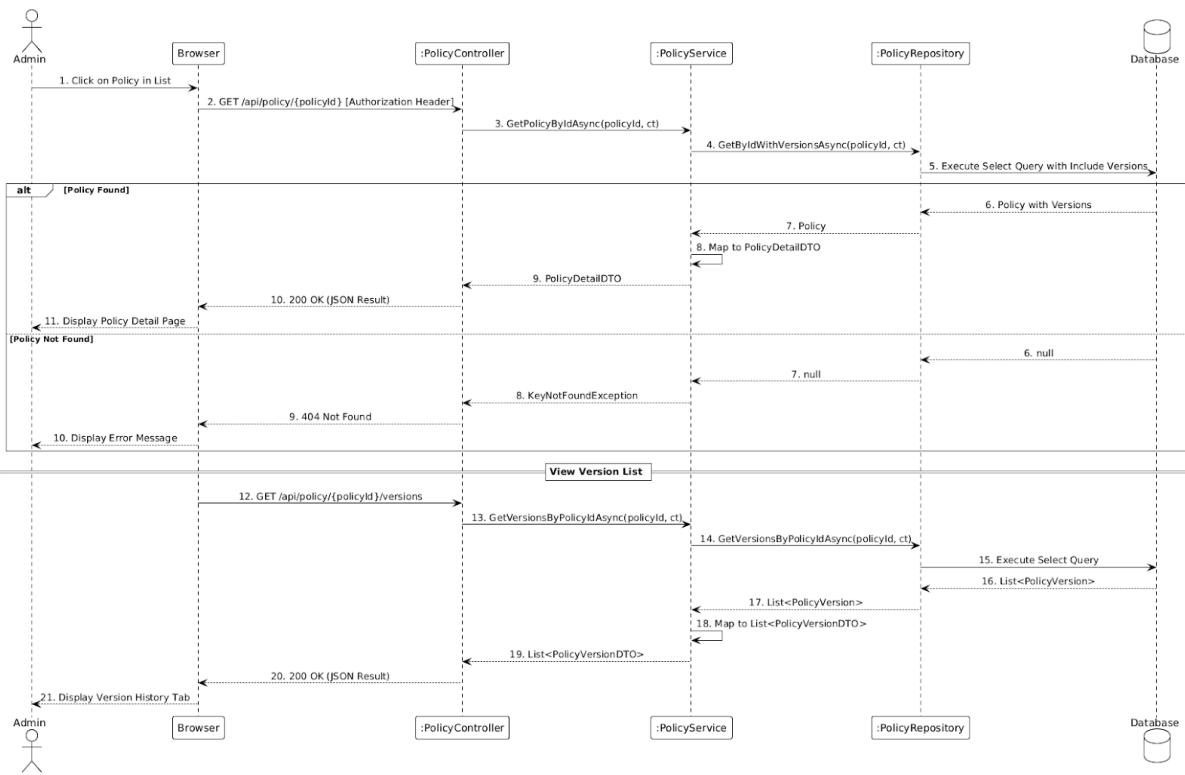
3.12.2.1 View User Policy



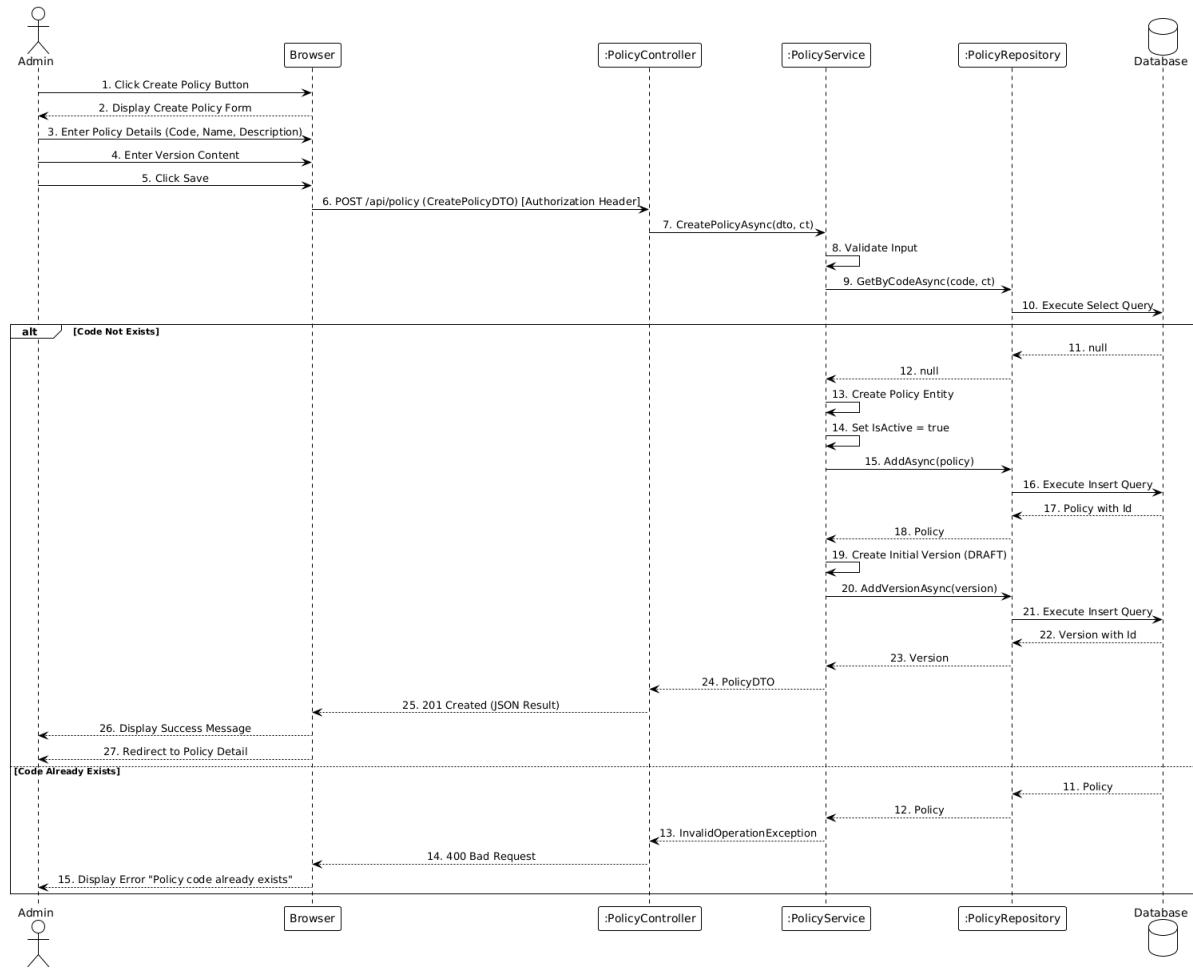
3.12.2.2 Accept Policy



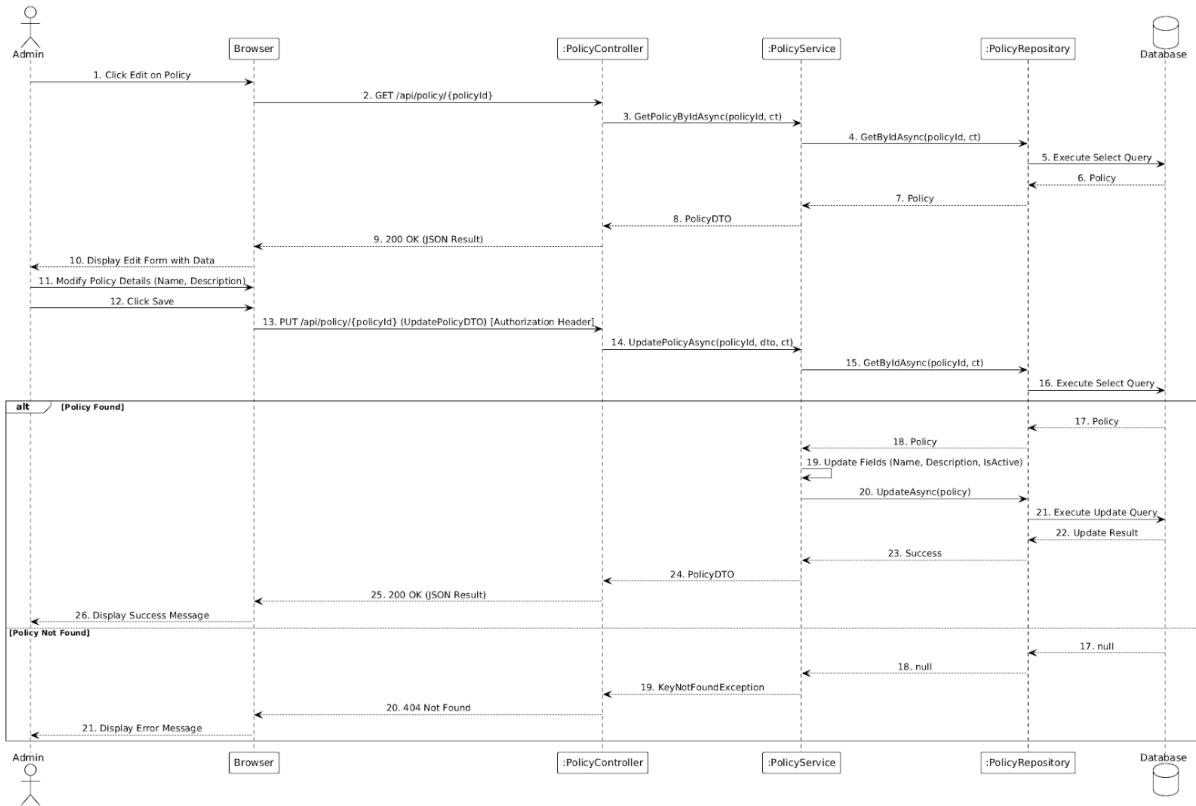
3.12.2.3 View Policy Detail



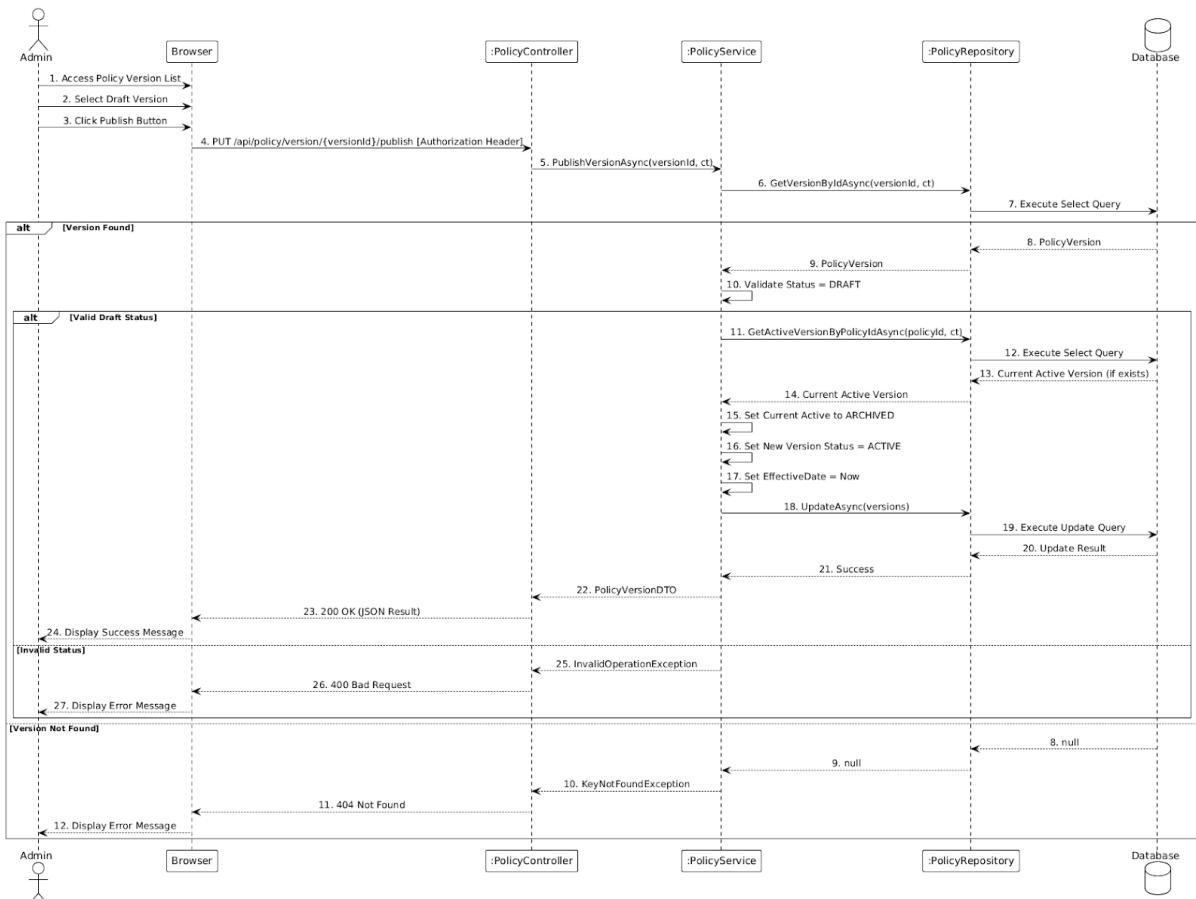
3.12.2.4 Create Policy



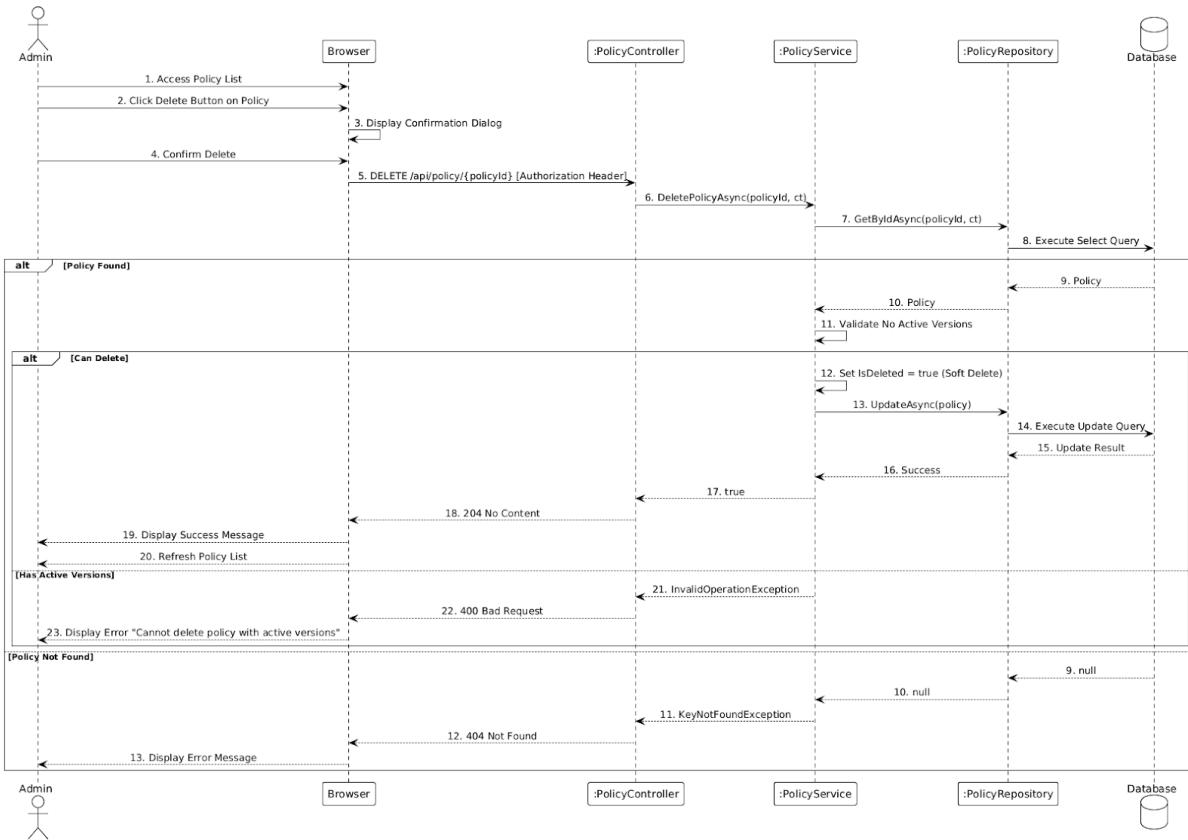
3.12.2.5 Edit Policy



3.12.2.6 Publish Policy

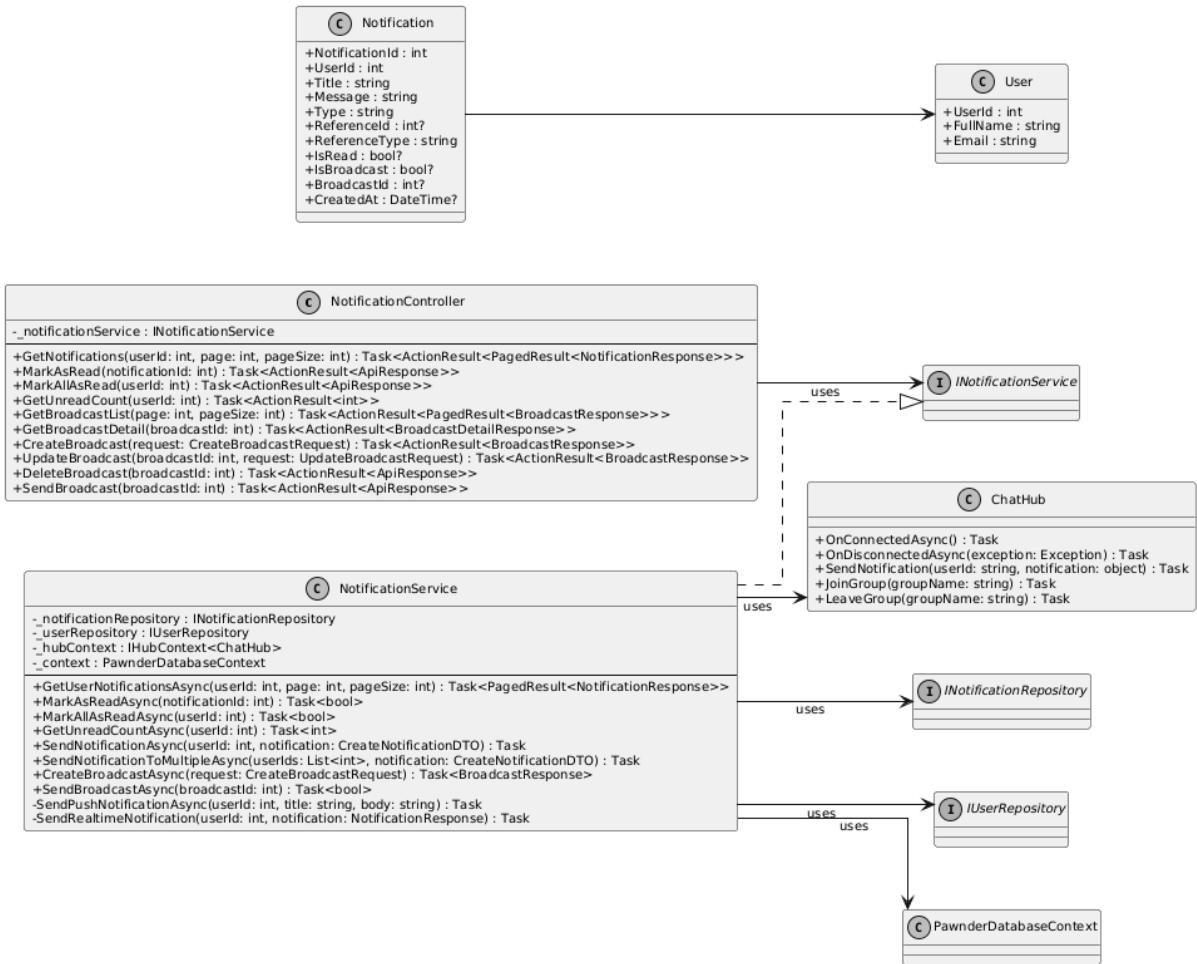


3.12.2.7 Delete Policy



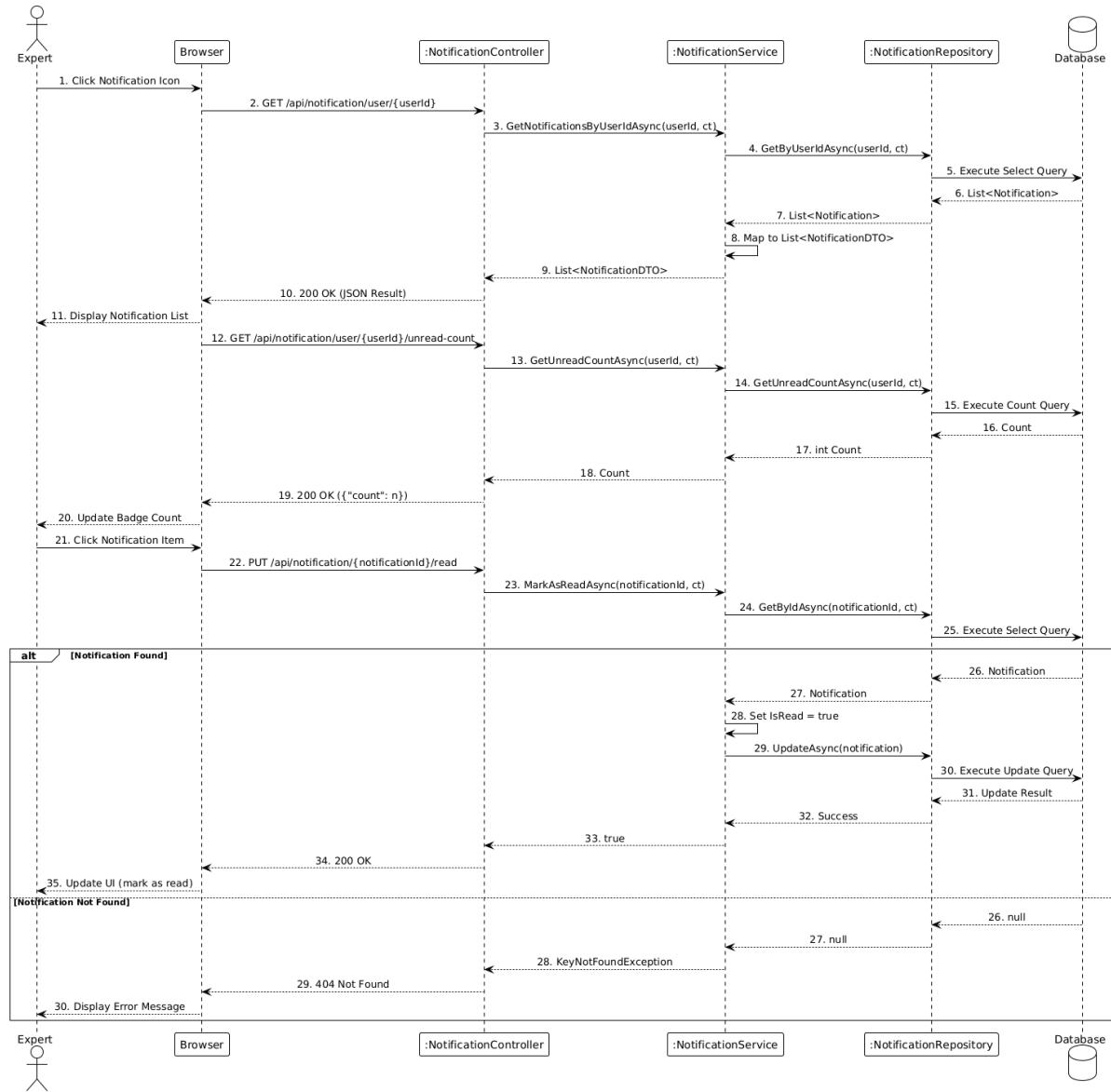
3.13 Notification & Broadcast

3.13.1 Class Diagram

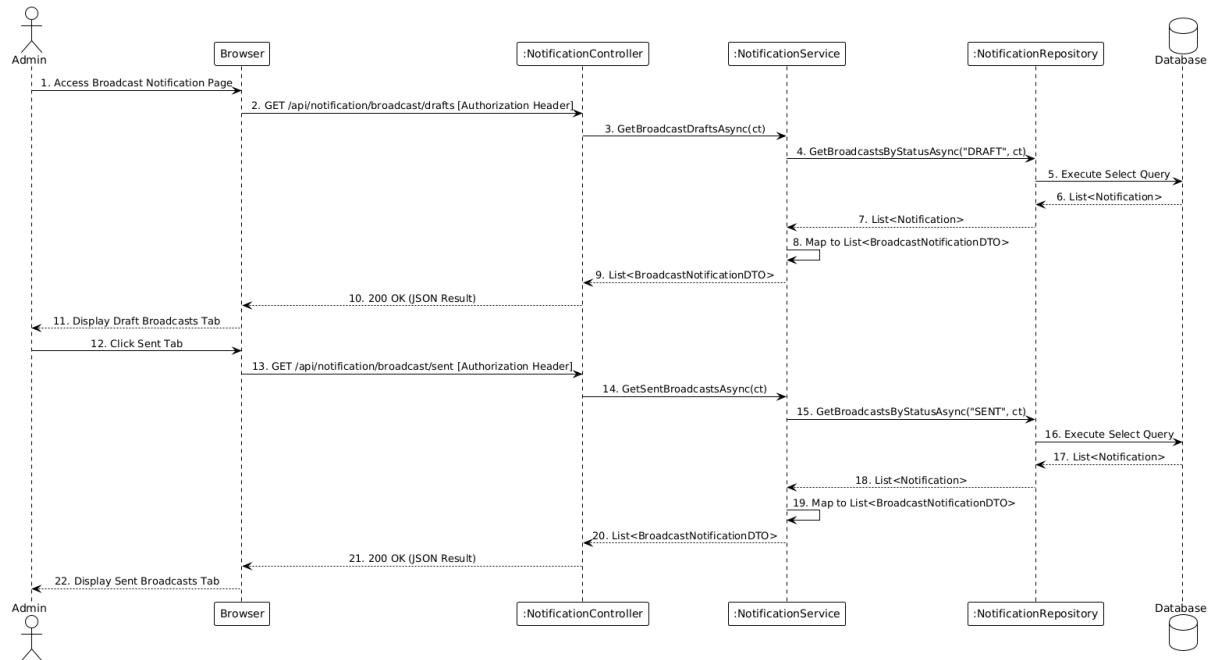


3.13.2 Sequence Diagram

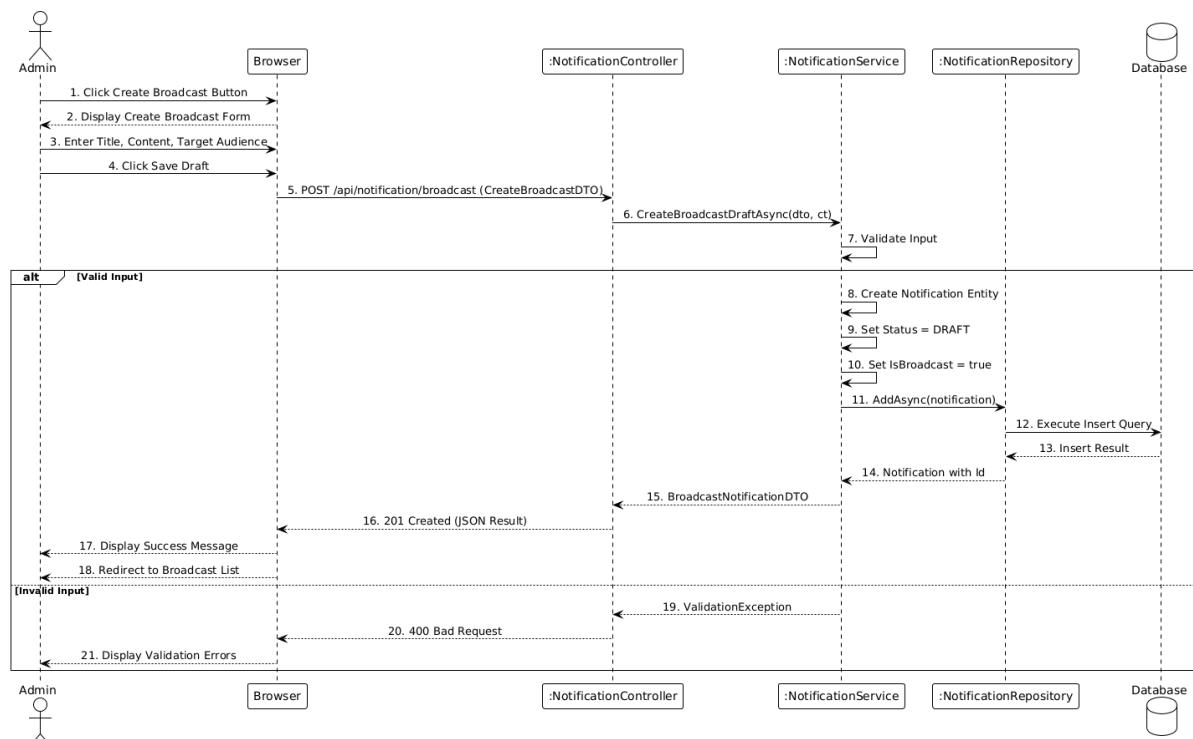
3.13.2.1 View Notification



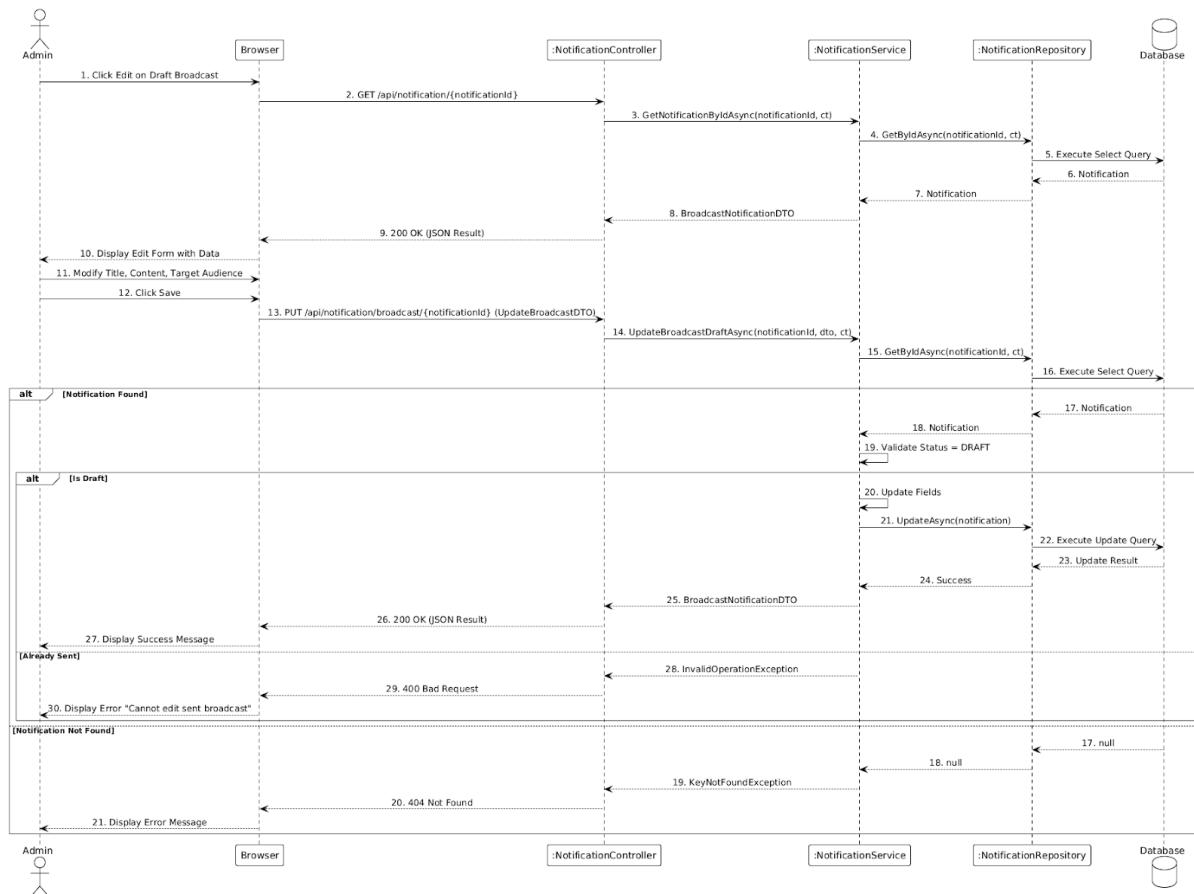
3.13.2.2 View Broadcast List



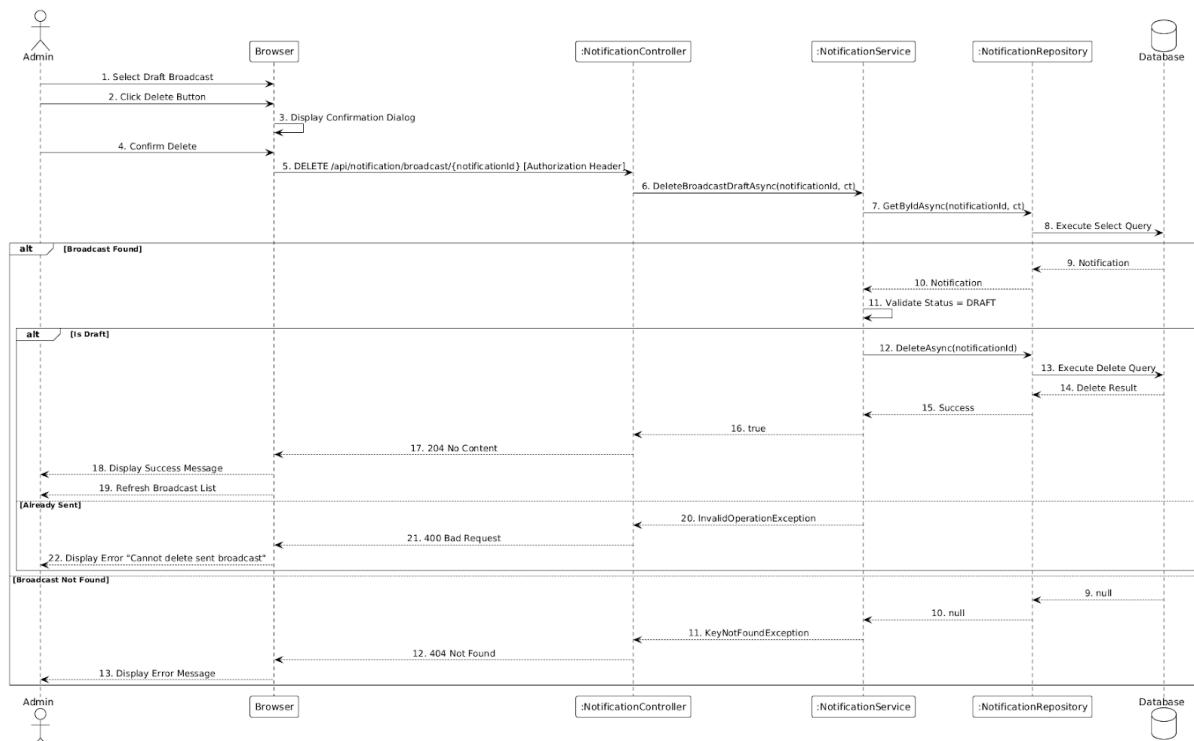
3.13.2.3 Create Broadcast



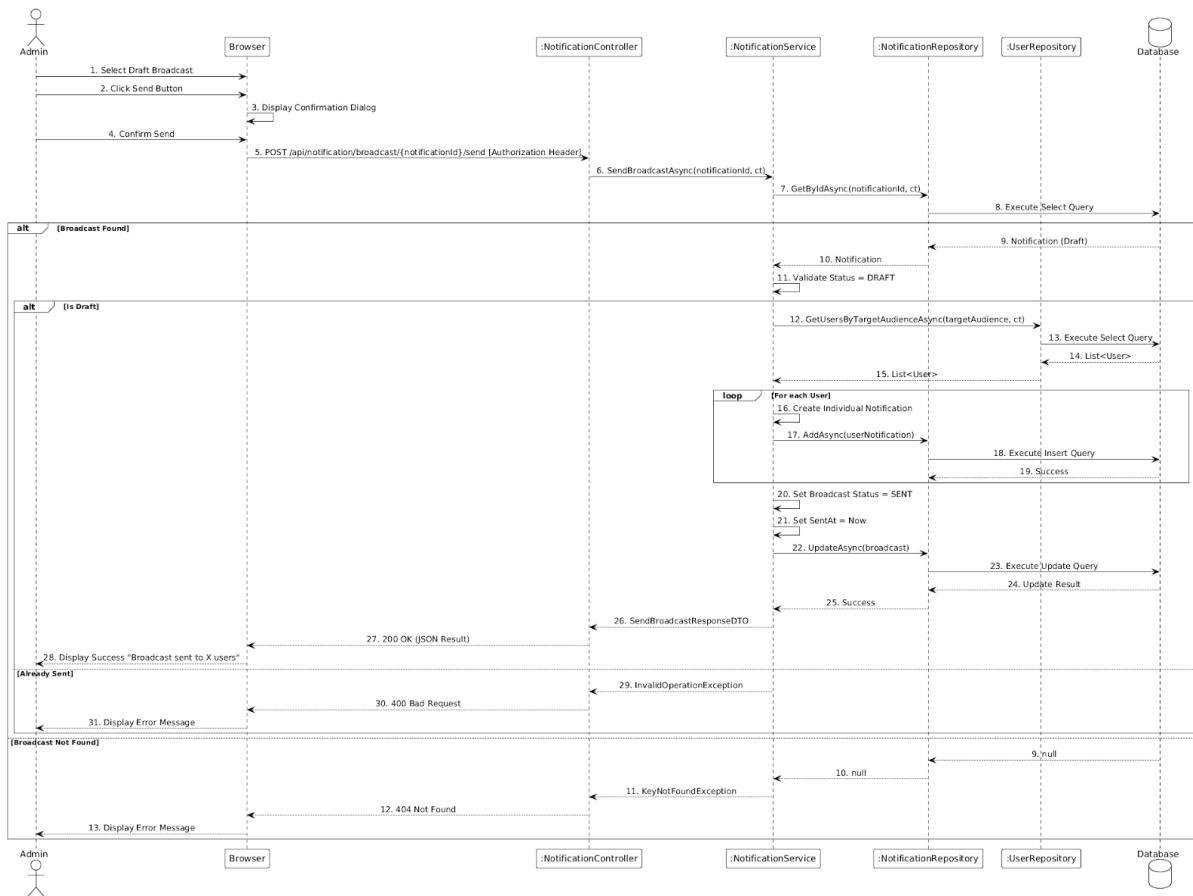
3.13.2.4 Edit Broadcast



3.13.2.5 Delete Broadcast

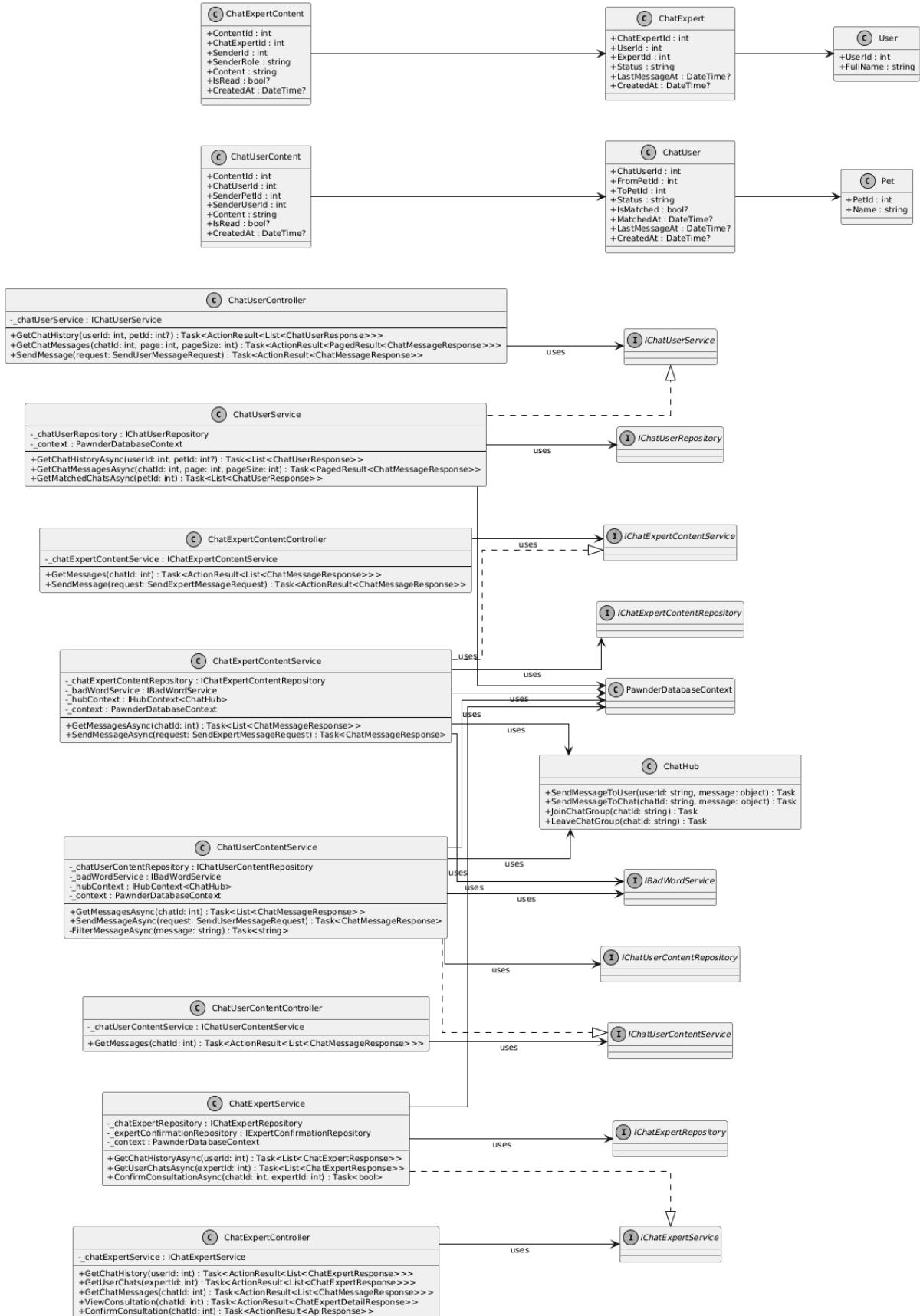


3.13.2.5 Send Broadcast



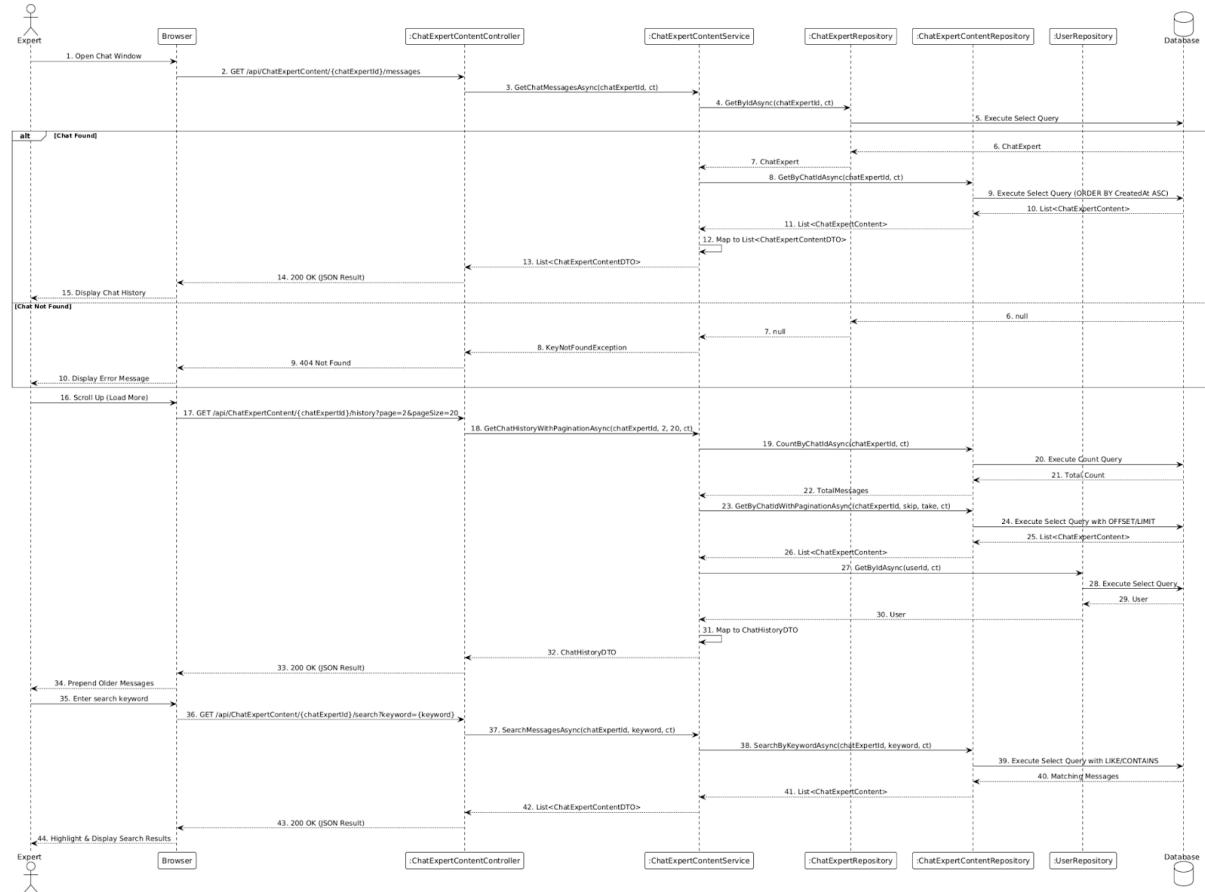
3.14 Chat & Messaging

3.14.1 Class Diagram

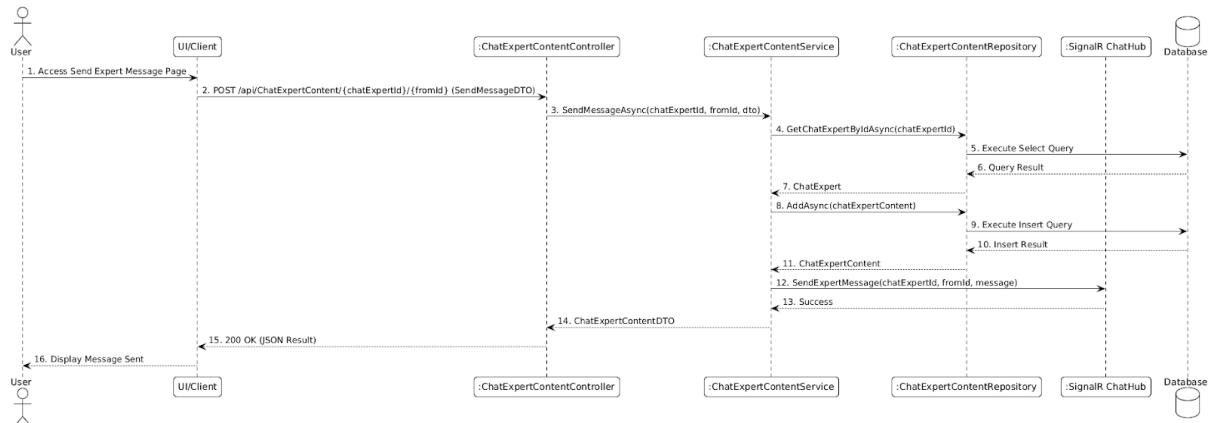


3.14.2 Sequence Diagram

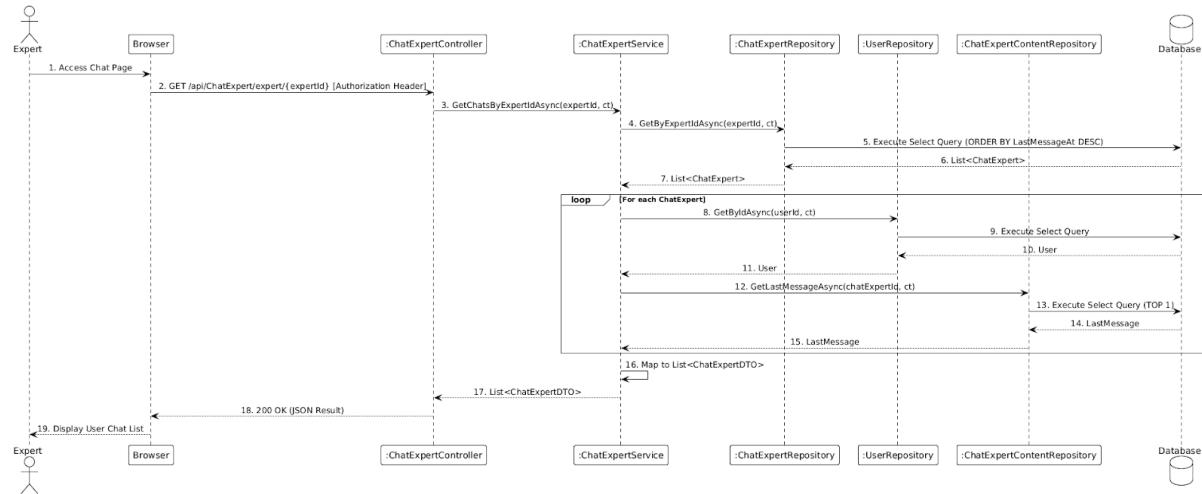
3.14.2.1 View Chat History



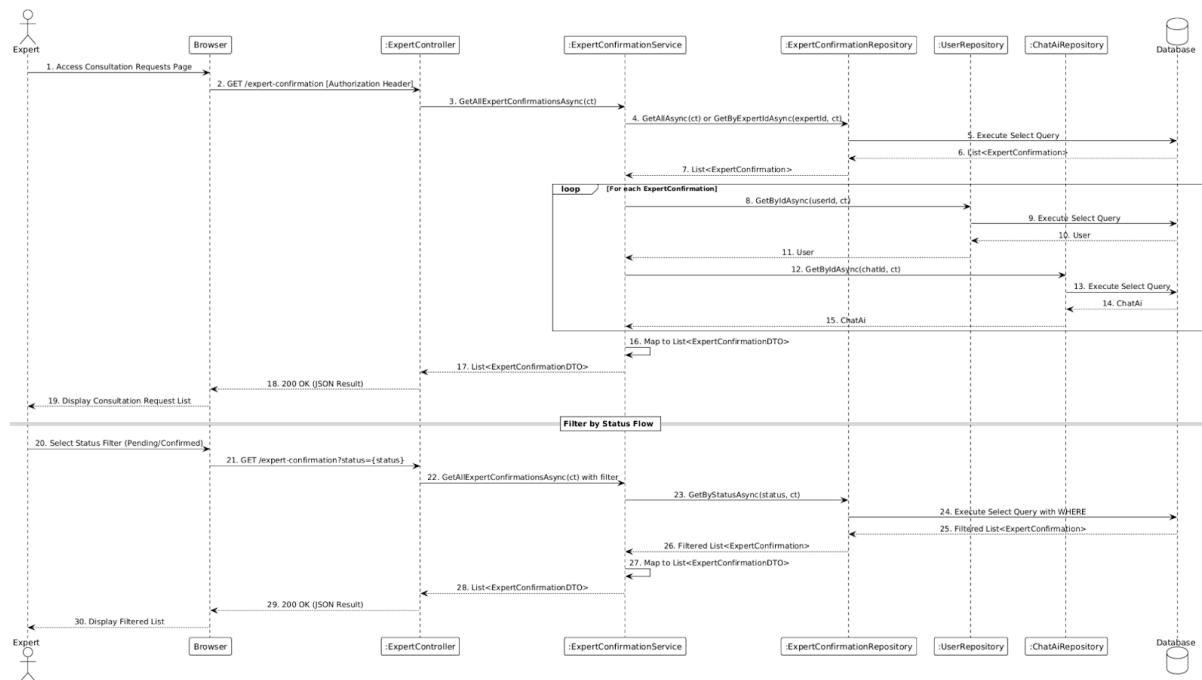
3.14.2.2 Chat with Expert



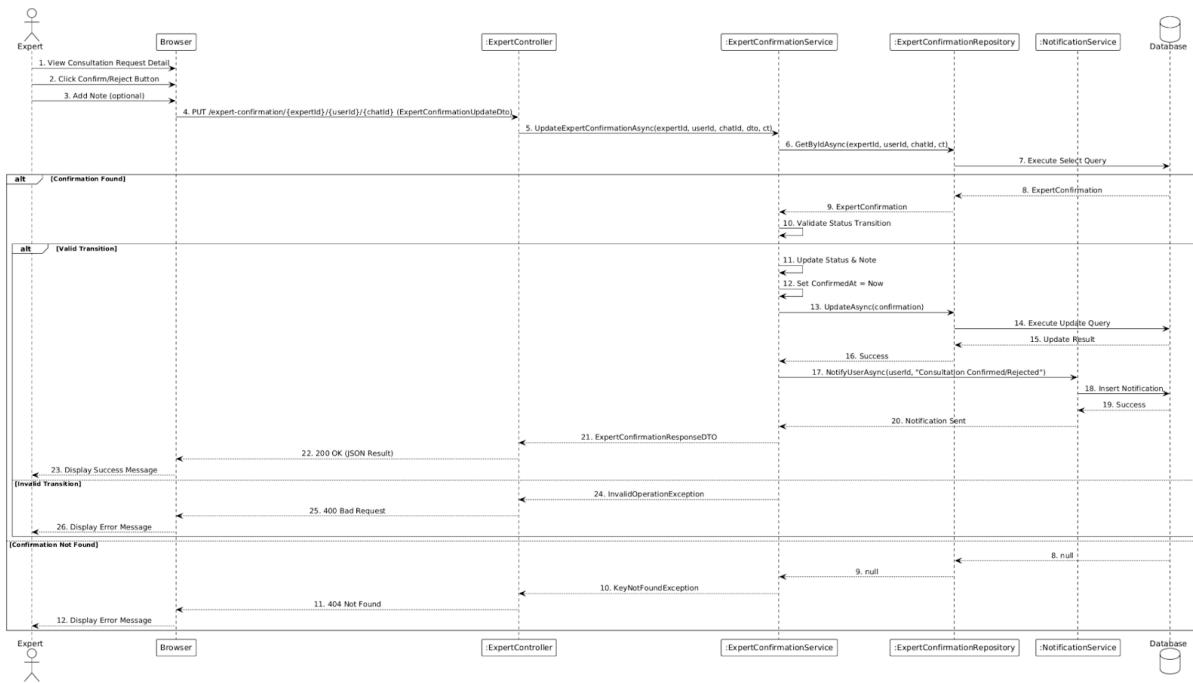
3.14.2.5 View User Chat List



3.14.2.6 View Consultation

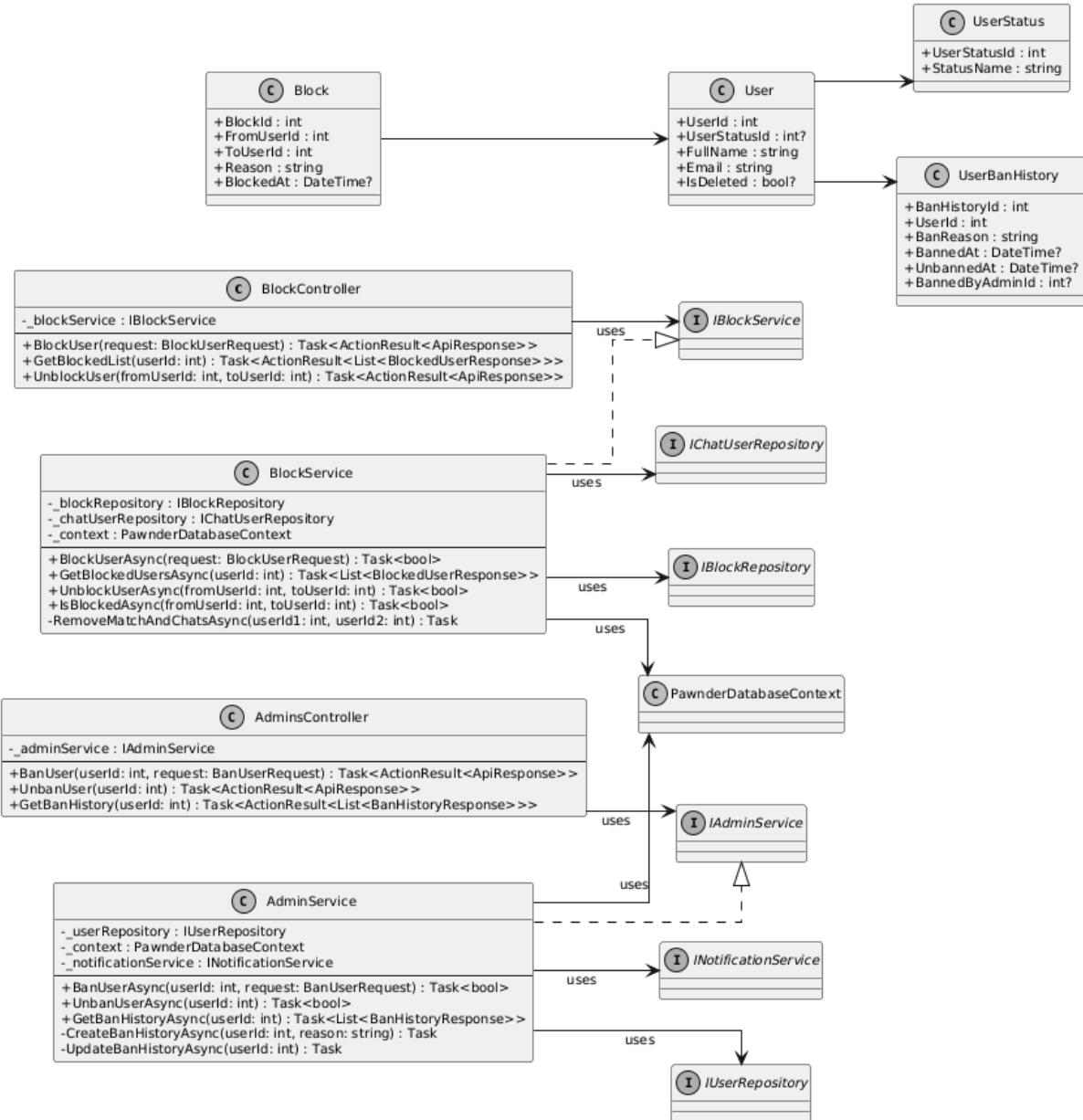


3.14.2.7 Confirm Consultation



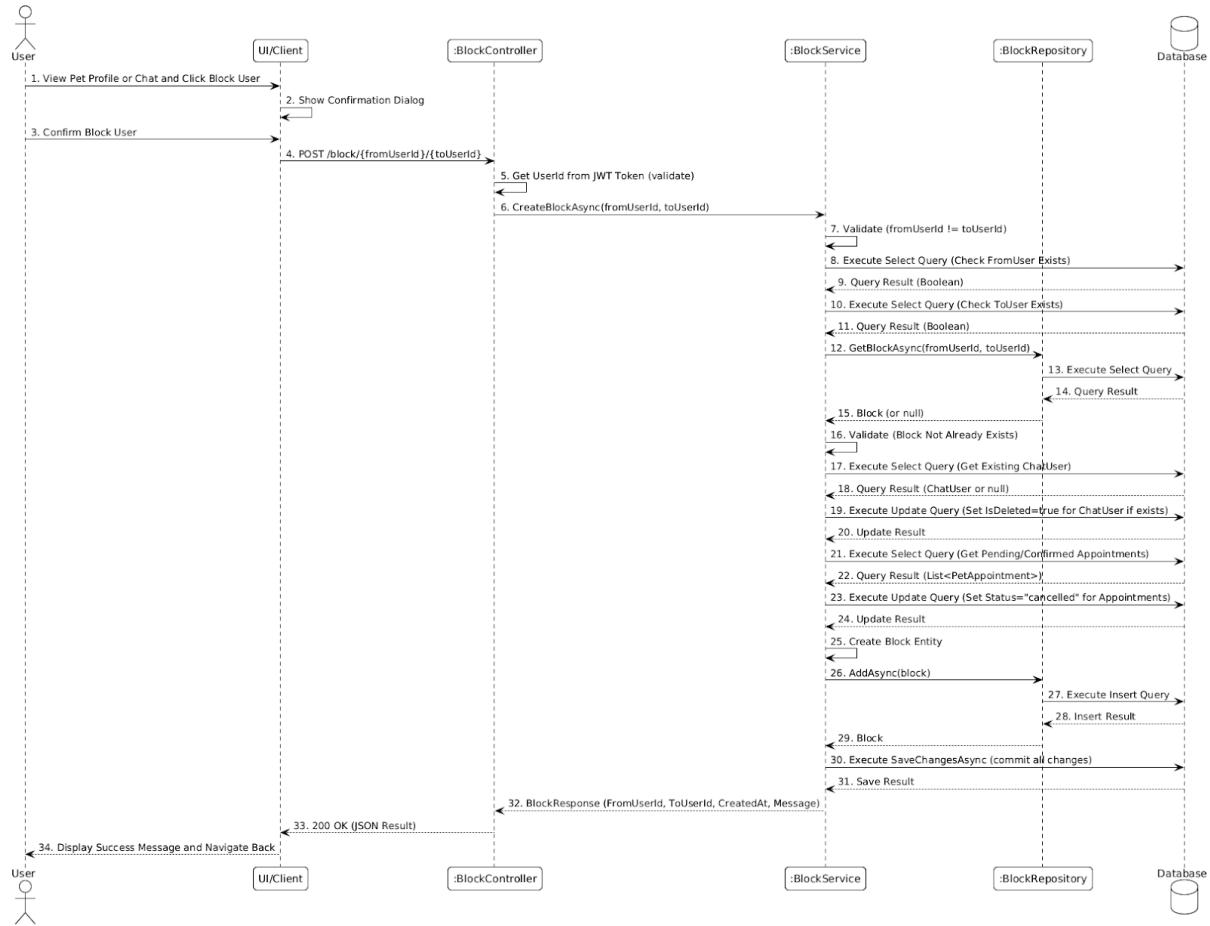
3.15 Block & Privacy

3.15.1 Class Diagram

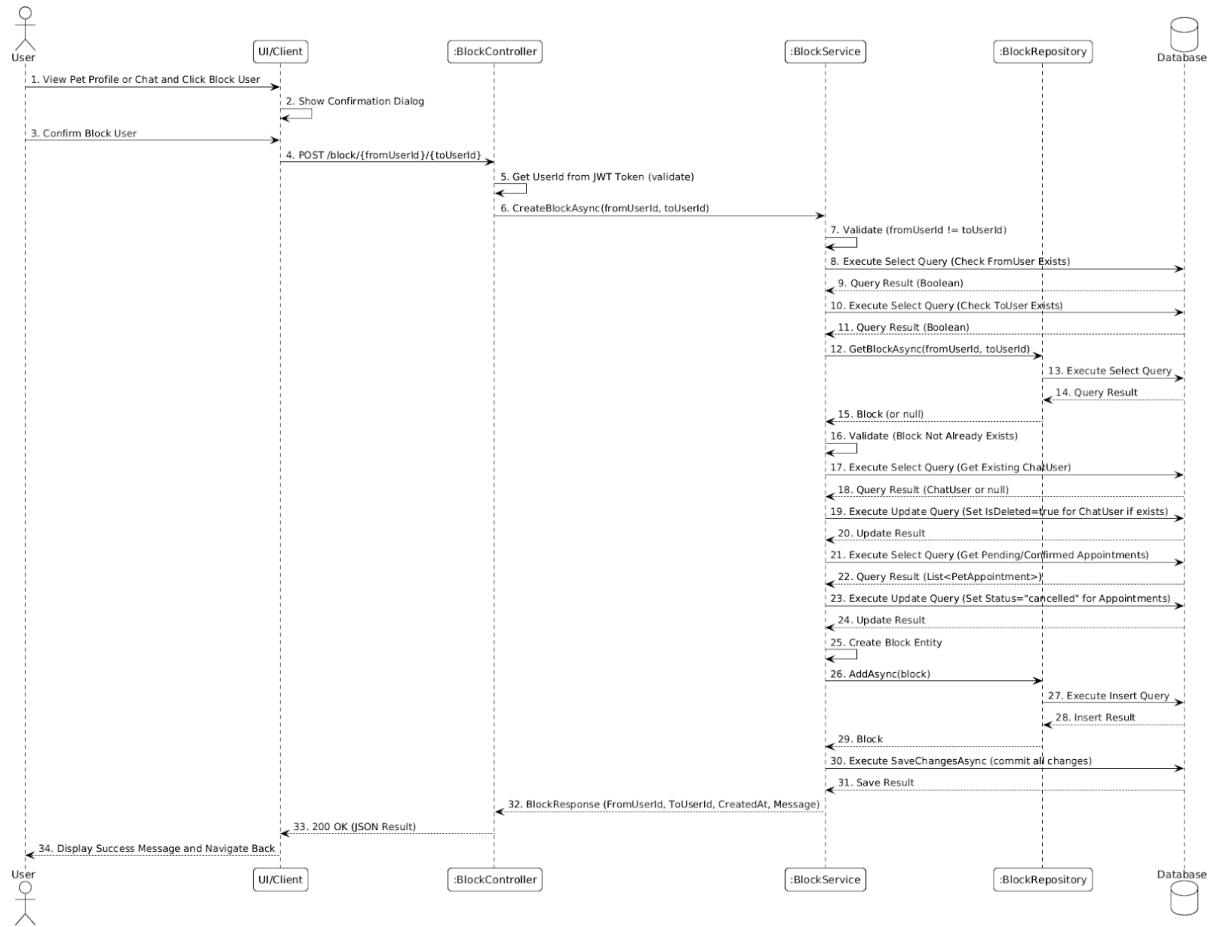


3.15.2 Sequence Diagram

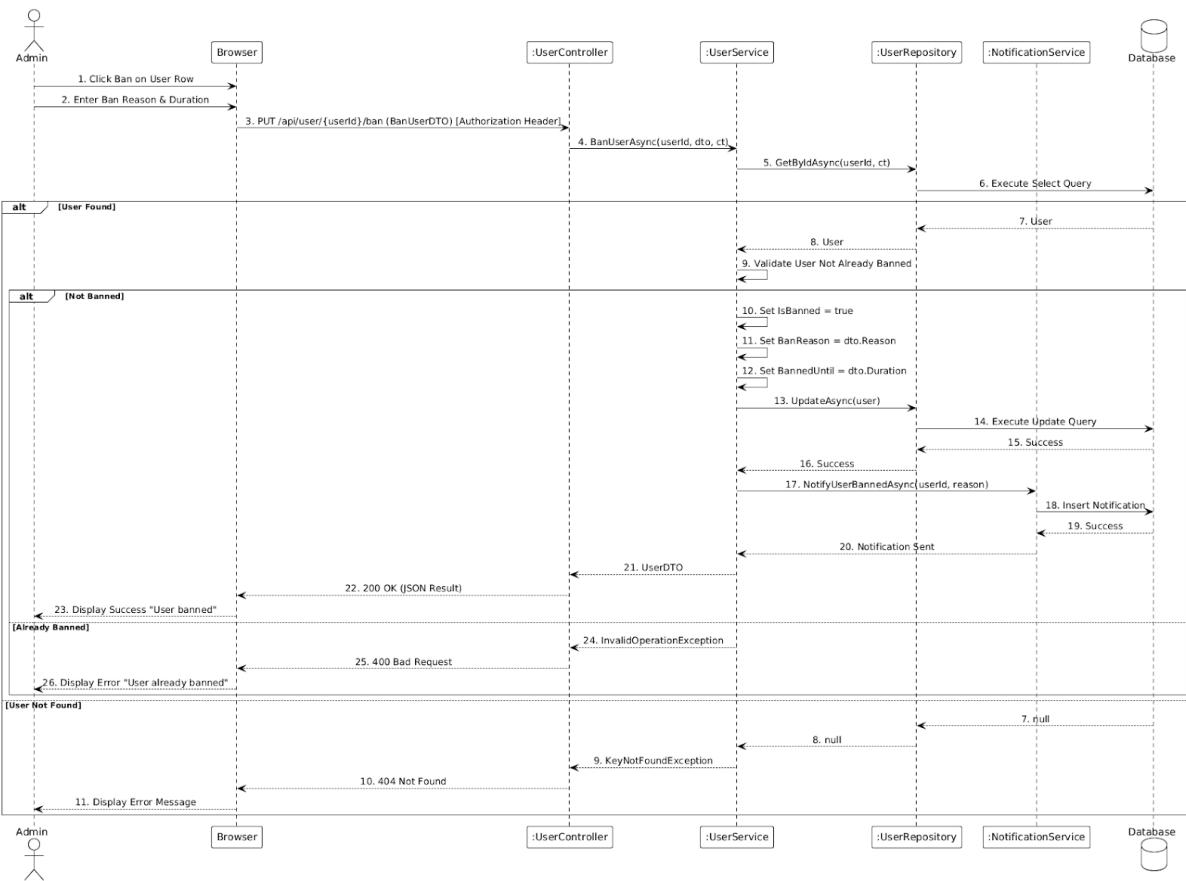
3.15.2.1 Block User



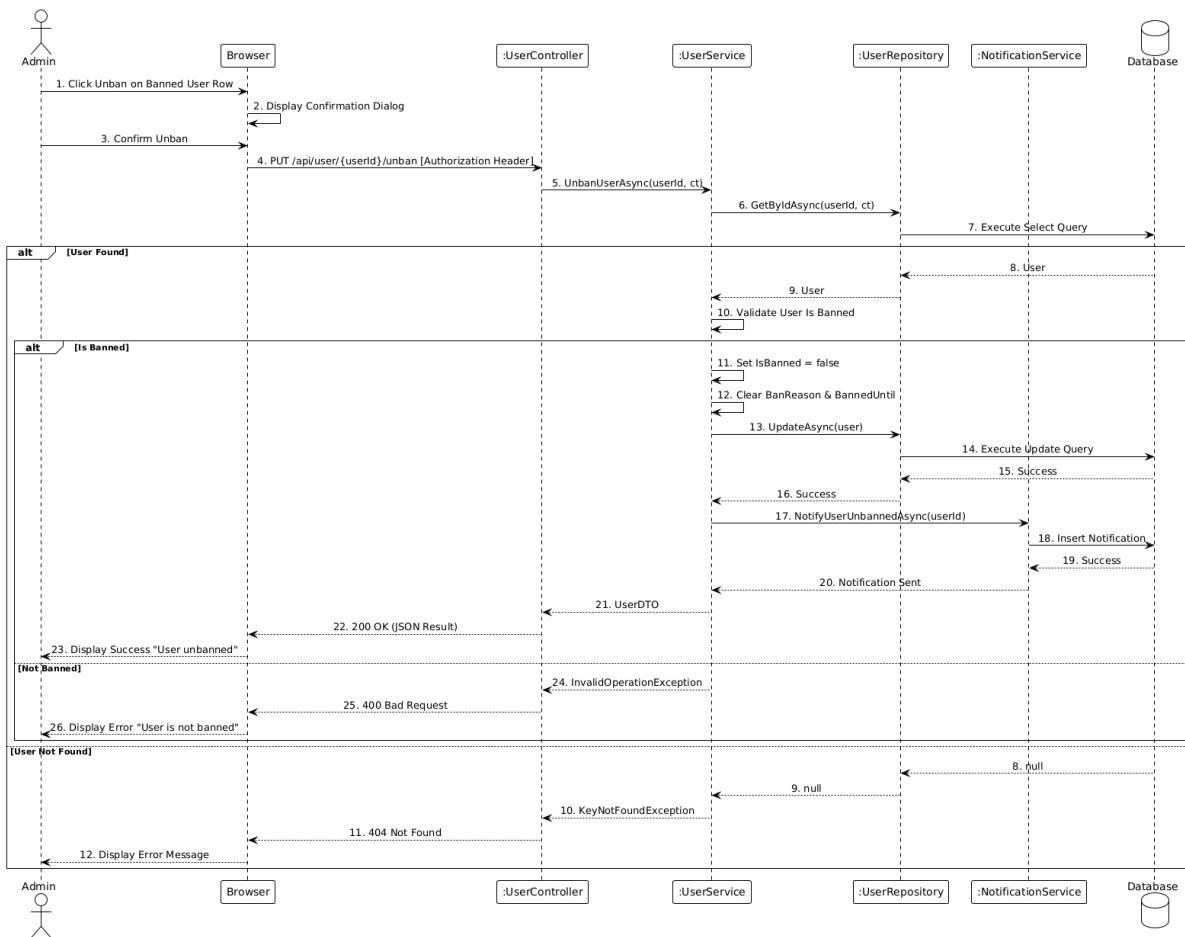
3.15.2.2 View Blocked List



3.15.2.3 Ban User

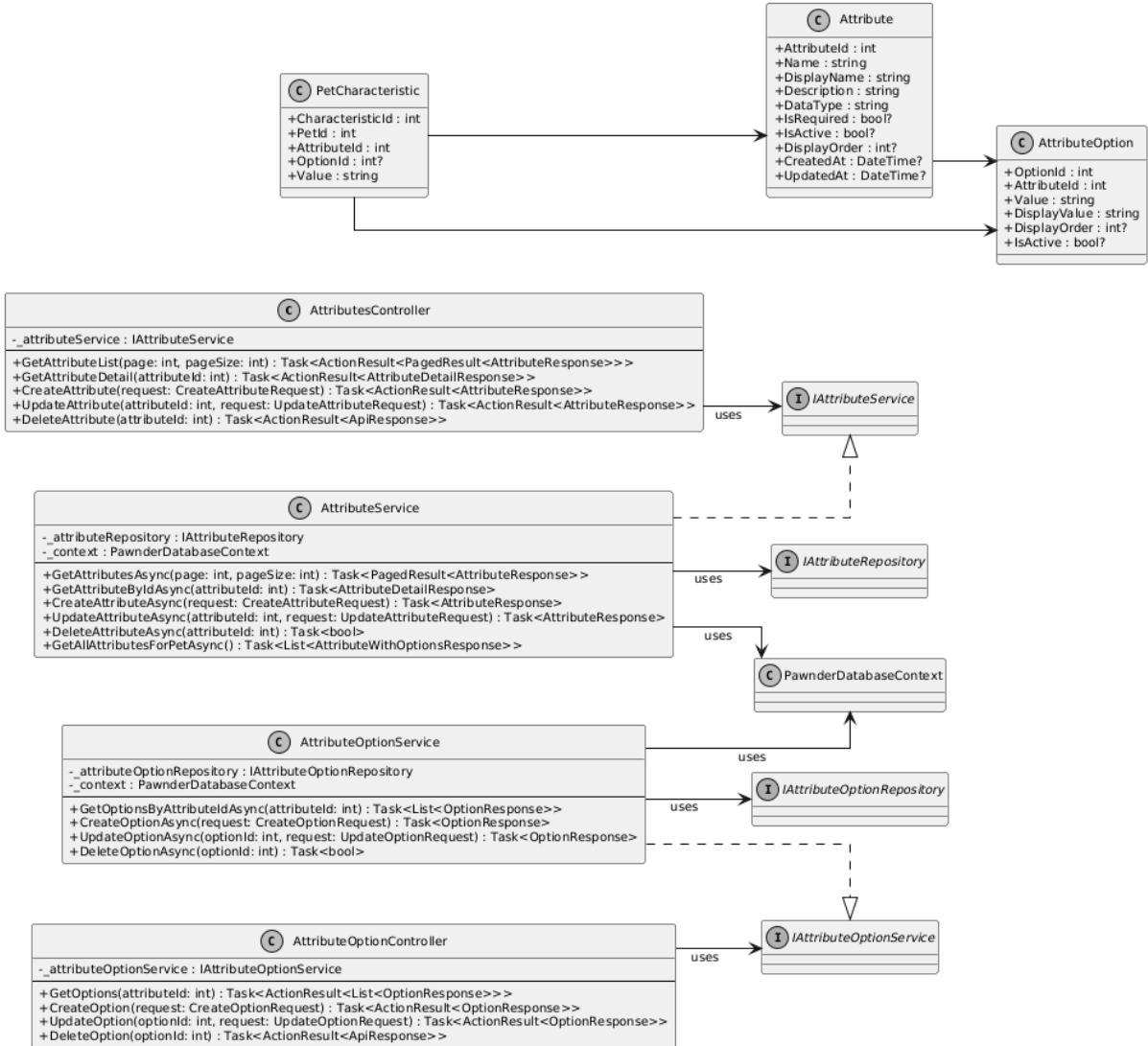


3.15.2.4 Unban User



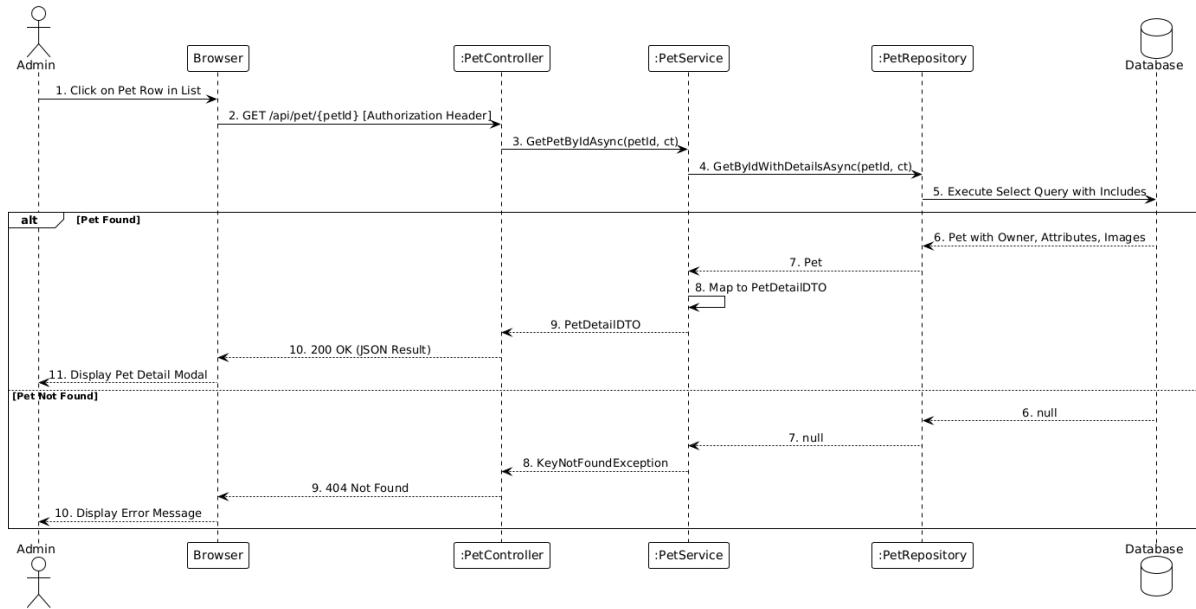
3.16 Attribute Management

3.16.1 Class Diagram

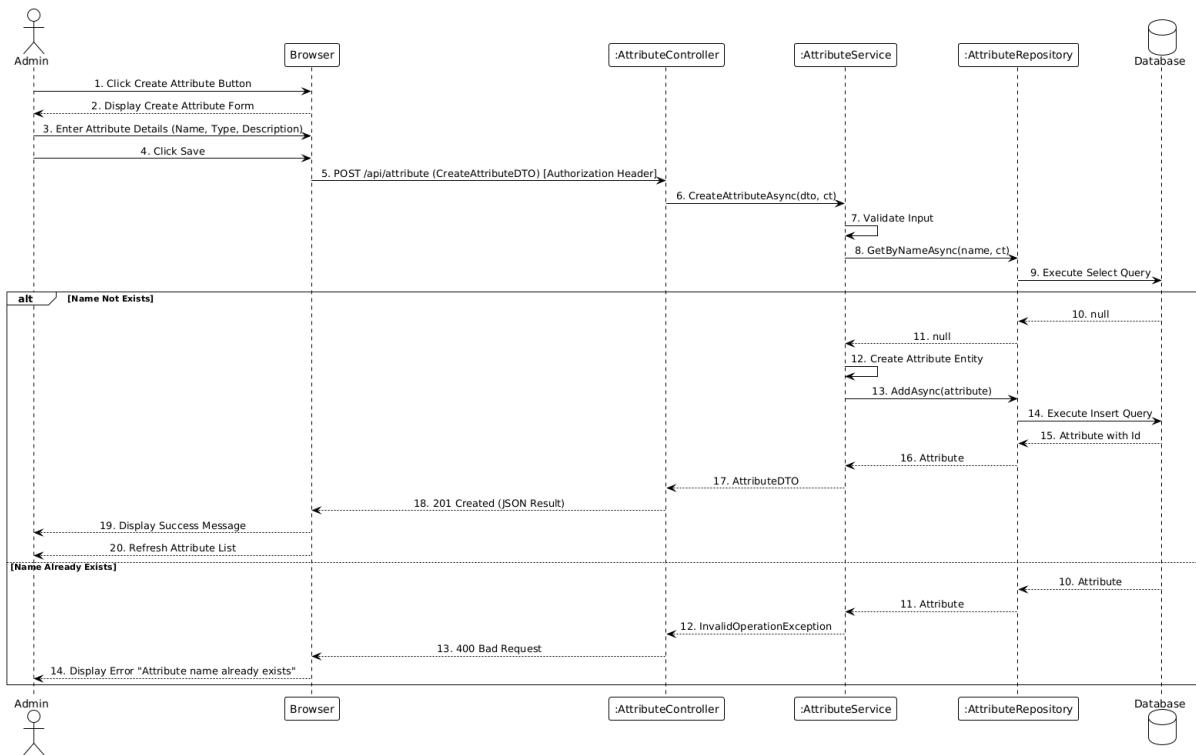


3.16.2 Sequence Diagram

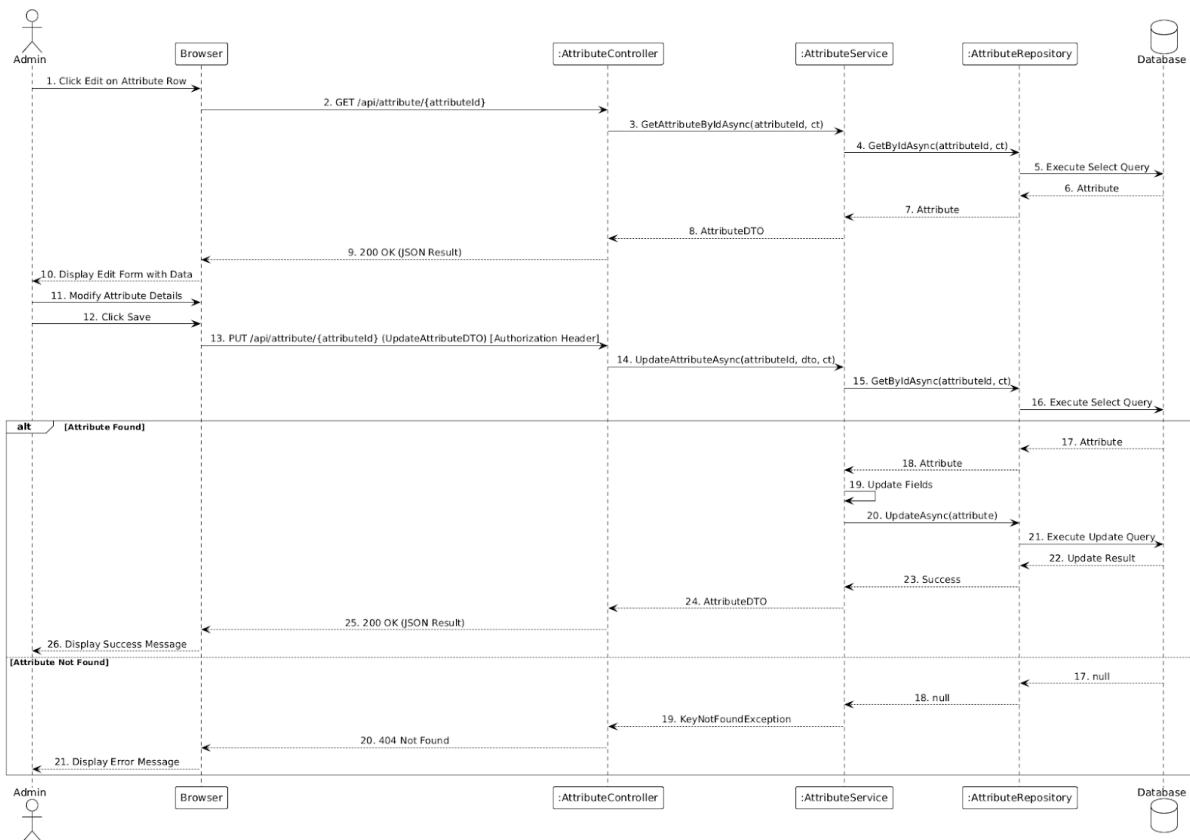
3.16.2.1 View Attribute List



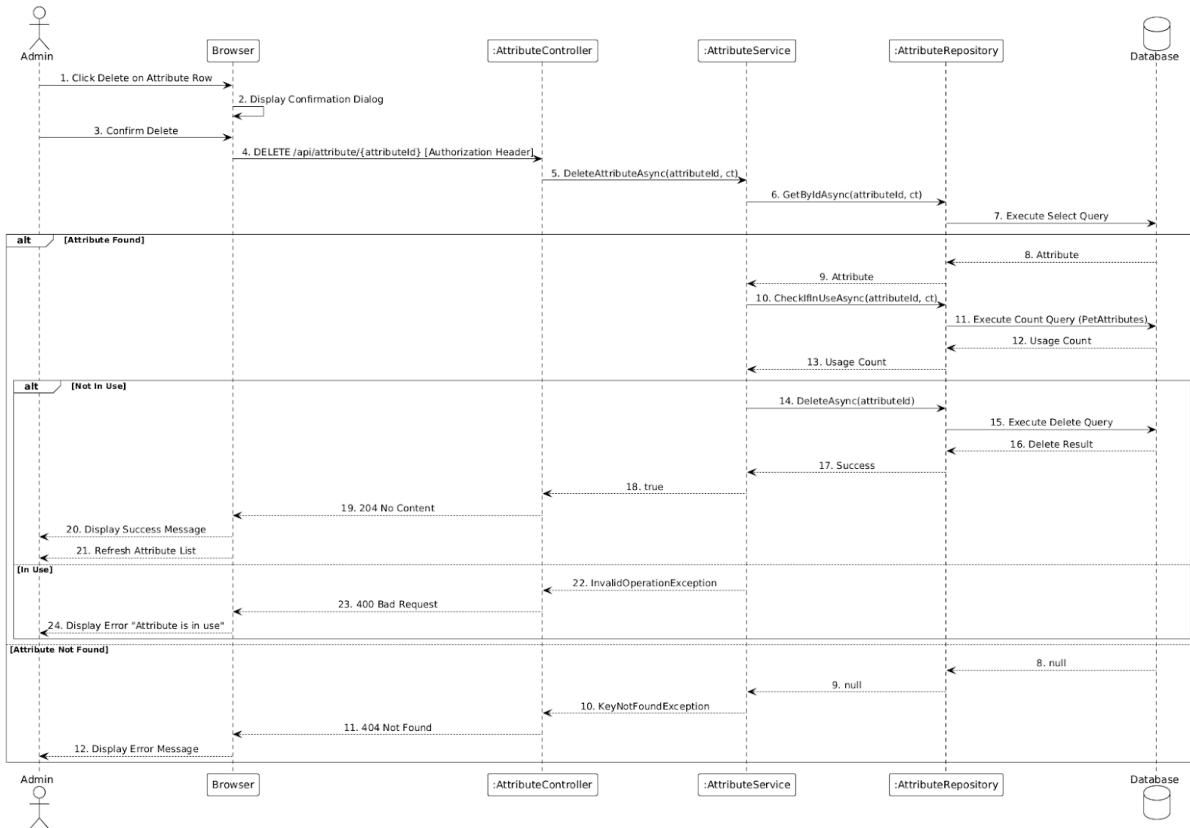
3.16.2.2 Create Attribute



3.16.2.3 Edit Attribute

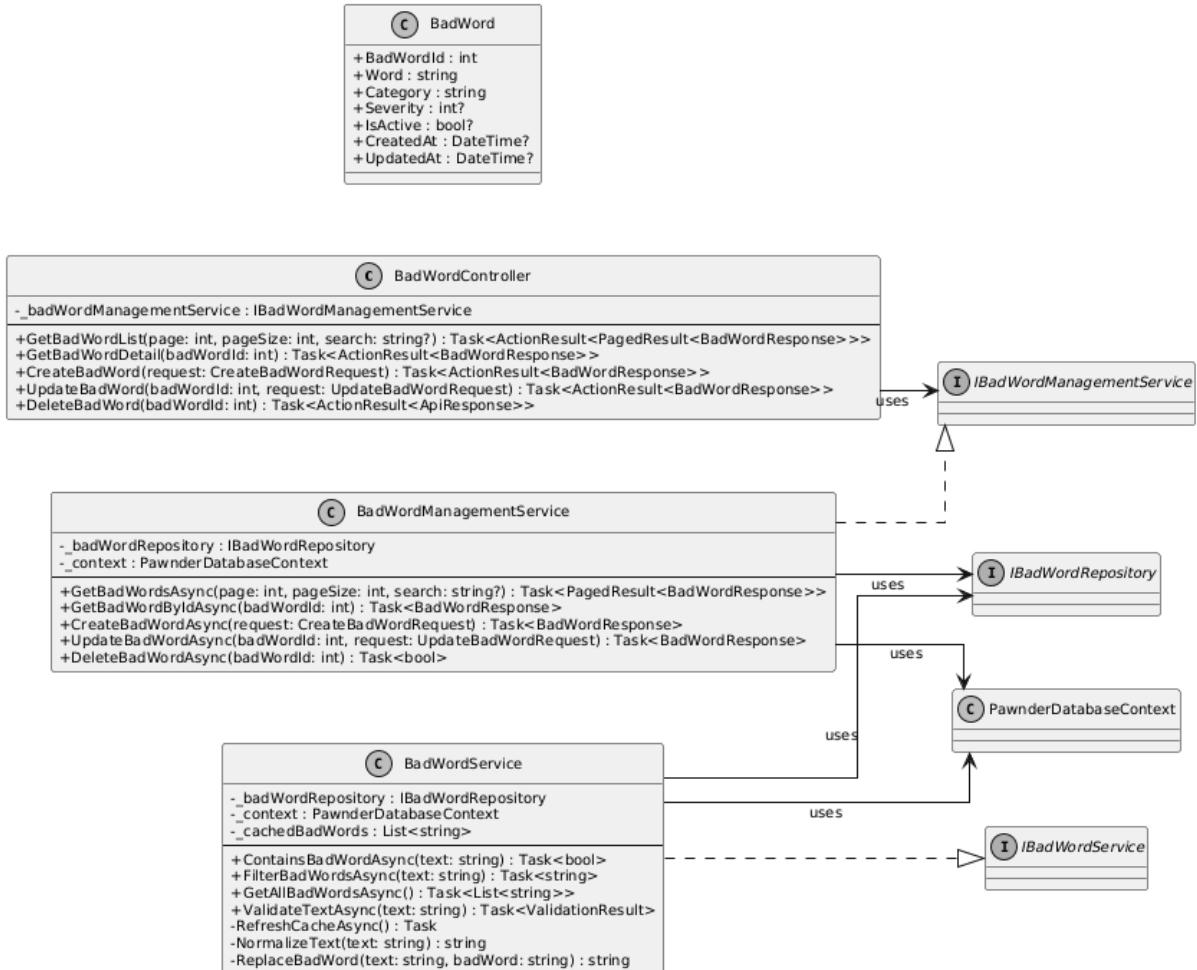


3.16.2.4 Delete Attribute



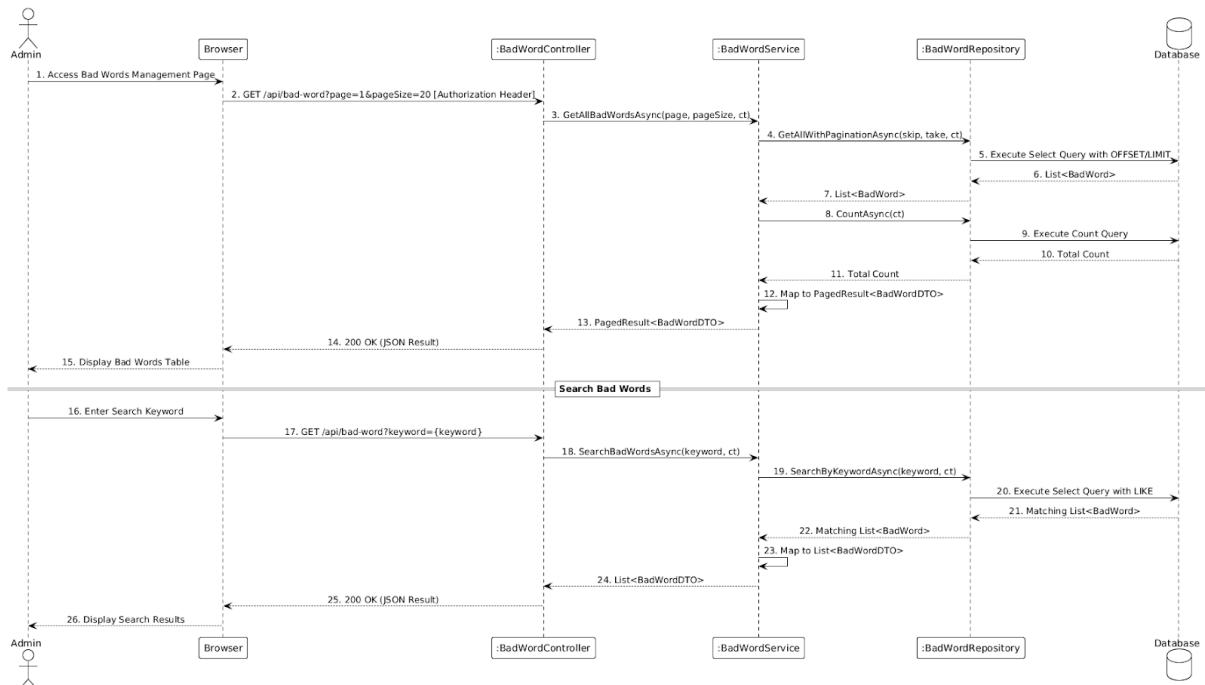
3.17 Bad Word Management

3.17.1 Class Diagram

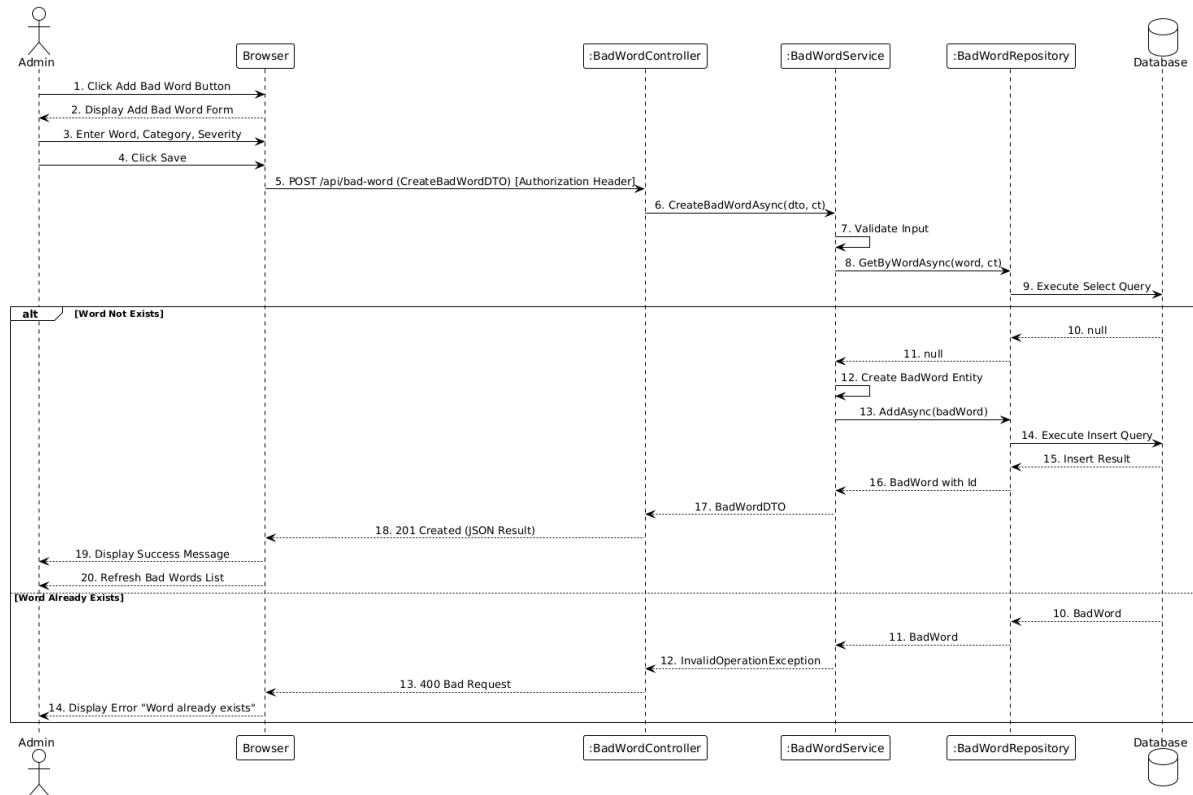


3.17.2 Sequence Diagram

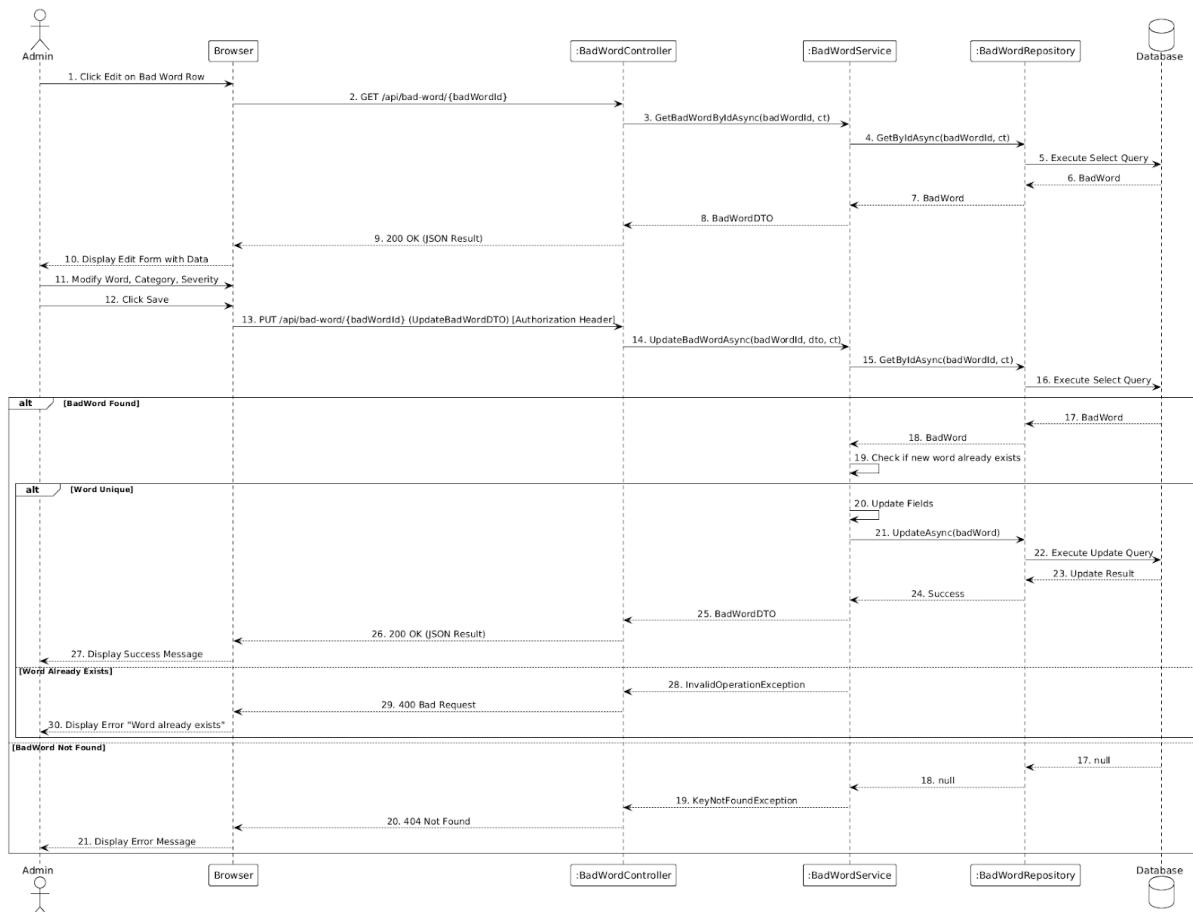
3.17.2.1 View Bad Word List



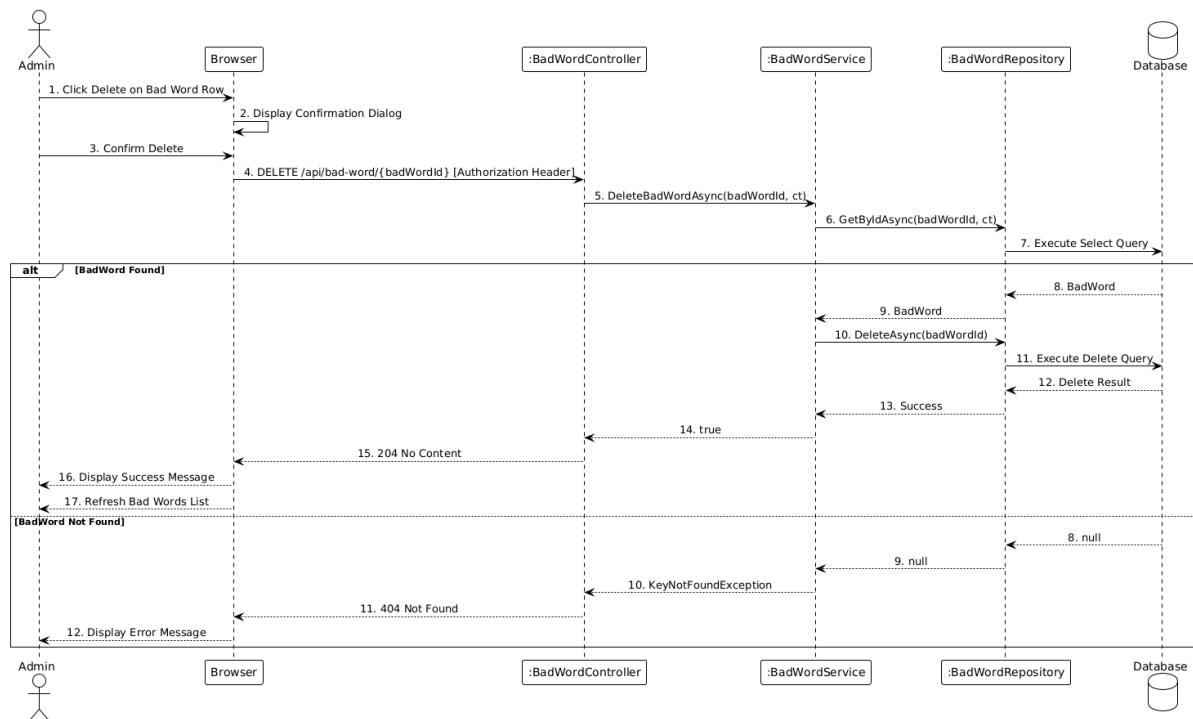
3.17.2.2 Create Bad Word



3.17.2.3 Edit Bad Word



3.17.2.4 Delete Bad Word



V. Software Testing Documentation

1. Scope of Testing

1.1. Testing Targets

1.1.1. Feature, Functional

The test scope of the project includes all features – functions defined in [Report 1_Project Introduction]

1.1.2. Non-Functional

Scope

- Backend (ASP.NET Core) and PostgreSQL.
- Includes APIs, background services, DB interactions, logging, configuration, deployment, and monitoring.

Availability

- Graceful Shutdown: Background services must respect CancellationToken for safe shutdown (the current service uses stoppingToken).
- Rollback: DB changes must have rollback plans and use transactions where appropriate.

Reliability & Consistency

- Atomic Updates: Payment status updates should be atomic; use transactions or compensating logic if multiple tables are modified.

- Retry Policy: DB connection retries are configured (EnableRetryOnFailure); background jobs should implement retry with backoff on transient failures. Current code logs exceptions and waits 5 minutes — consider structured retry/backoff and alerting.
- Acceptance Criteria: On temporary DB outage, job retries at least 3 times with exponential backoff and avoids duplicate updates.

Security

- Auth/Z: JWT authentication is used; ensure secret rotation and secure storage.
- Data in Transit/At Rest: Use TLS for external communications and consider encryption for sensitive fields.
- Acceptance Criteria: No sensitive secrets committed to repo; secret-scan CI passes.

Deployment & Configuration:

- Config-Driven: Important parameters (interval, enable/disable flags, batch size) configurable via appsettings or environment variables.
- Acceptance Criteria: Service can be disabled via configuration without code changes or redeployment.

2. Test Strategy

We determine that Agile testing is the best option for our project's testing procedure, given the specifics of that project. This test methodology offers the ideal stages for our project, from test preparation to test execution to test conclusion.

Agile Testing Process

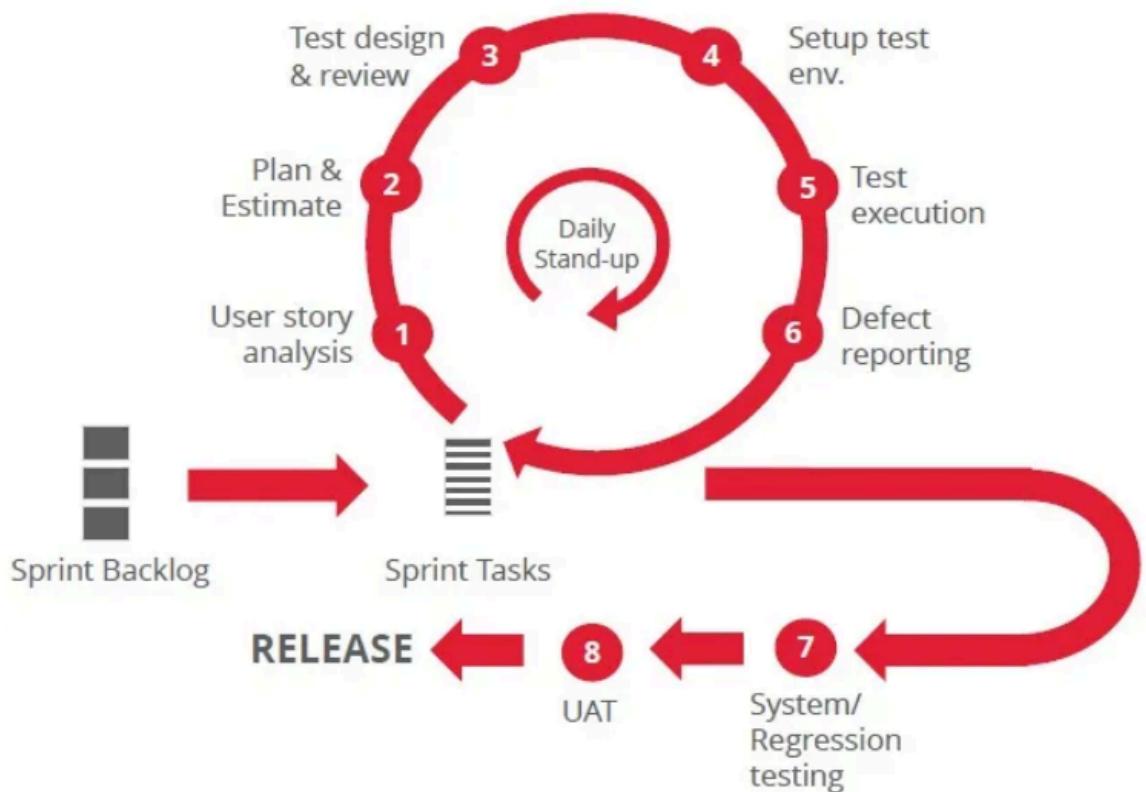


Figure 2.1. Agile Testing Process

No	Objective	Techniques	Completion criteria	Test level	Frequency
1	Verify small units of code (methods, classes) behave correctly in isolation (business rules, utilities)	Mock dependencies (repositories, DB context, external services) with test doubles; assert outputs, exceptions, and state changes.	All unit tests pass in CI; minimum coverage target for core modules.	Unit	Developers

2	Verify interactions between components, especially EF Core and PostgreSQL behavior, DI scopes, repository implementations, and transaction correctness.	Seed data, run code paths. Also test wiring of hosted service registration if needed.	Integration tests pass in CI stage; no unintended side effects on other tables.	Integration	Testers
3	Validate controller endpoints, request/response contracts, validation and auth enforcement.	Superficial startup in-memory server or hitting staging endpoints with test accounts; mock external calls.	All endpoints respond with expected status codes/data for defined inputs; authentication and authorization enforced per policy.	System	Testers
4	Validate full workflows with real integrations	Deploy to staging; use automated scripts to simulate user flows; include real or sandbox credentials for Cloudinary/Sepa y or use service mocks where necessary.	All critical scenarios complete successfully; background job processed expected records; no regression on API beyond threshold.	Acceptance	End-users

2.1 Testing Types

Types	Objective	Technique	Completion Criteria
-------	-----------	-----------	---------------------

Unit Test	Verify the correctness of individual units of code in isolation so logic errors are caught early and cheaply.	Use a unit test framework. Replace external dependencies with mocks or fakes. Write focused tests for normal cases, edge cases, error paths, and parameterized inputs; include boundary and negative tests. Run fast and locally/CI on every commit.	All unit tests pass in CI; agreed coverage threshold for core modules met; new PRs do not introduce failing or flaky unit tests; execution time stays within fast-feedback limits.
GUI Testing	Validate user-facing UI behavior, layout, and usability so the application delivers the intended user experience across supported browsers/devices.	Combination of automated end-to-end UI tests for critical flows and manual exploratory/usability testing for edge-case UX. Use visual-regression tools for layout diffs, cross-browser runs, and test accounts or staging backends. Isolate flaky selectors and use stable locators.	All critical UI flows pass automated checks in CI/staging; visual-regression differences are within acceptable thresholds; cross-browser smoke matrix passes for supported browsers; no open high-severity UI defects.
API Testing	Ensure API endpoints meet contract, authentication, validation, error handling, and data correctness expectations	Automated contract tests Postman covering positive, negative, boundary, and security scenarios. Validate JSON schema, status codes, headers, and auth/permission enforcement. Run tests against ephemeral DB or staging.	All API contract tests in CI pass; authentication/authorization enforced for protected endpoints; backwards-compatibility checks for published contracts; endpoints return expected payloads and status codes for representative scenarios.
Non-functional Testing	Verify the system meets NFRs such as performance, scalability, reliability, availability, security, and maintainability	Run performance/load tests for throughput and latency; stress and soak tests for stability; security scans and secret scanning; reliability tests including chaos/injection if needed.	System survives stress/soak within defined degradation behavior; no unresolved critical/ high security findings; monitoring dashboards and alerts verified;
Regression Testing	Ensure that new code changes do not break existing functionality across the system —	Maintain an automated regression suite that includes a combination of unit, integration, and E2E	Regression suite passes before merging major changes or releasing; any test failures are triaged

	protect previously validated behavior.	tests. Use selective running, and nightly/full-suite execution in CI. Prioritize tests by risk and impact.	and fixed before release; acceptable flakiness rate with flaky tests either fixed or quarantined.
Database Testing	Verify database schema correctness, data integrity, migration safety, query performance, and that DB interactions behave as expected in realistic conditions. Ensure migrations apply cleanly and do not corrupt or lose data.	Use automated integration tests against an isolated PostgreSQL instance seeded with representative test data.	Migrations apply cleanly in staging; integrity checks pass; queries meet performance criteria; backup/restore completes within RTO and RPO targets.
UI/UX Testing	Validate that the product is usable, intuitive, and meets user expectations and accessibility standards.	Usability testing with representative users , heuristic evaluations, accessibility audits, analytics review.	Users complete target tasks with acceptable success/time/error rates; identified usability issues are addressed or logged with priorities; accessibility violations reduced to acceptable levels.

2.2 Test Levels

Type of Tests	Test Level			
	Unit	Integration	System	Acceptance
Unit Test	X			
GUI Test			X	
API Test		X	X	X
Non-functional Test			X	X
Regression Test	X	X	X	X
Database Test		X	X	
UI/UX Test			X	X

2.3 Supporting Tools

Purpose	Tool	Vendor/In-house	Version
Manage API test collections and run automated API tests	Postman	Postman Inc.	Postman latest
Manage test/staging database instances and versioned schema	PostgreSQL	PostgreSQL Global	PostgreSQL 18

3. Test Plan

3.1 Test Environment

Purpose	Tool	Provider
Unit test documents	Excel	Microsoft
Integration test documents	Excel	Microsoft
System test documents	Excel	Microsoft
Run Unit Test	Visual Studio	Microsoft
Run Integration Test	Postman	QA Team
Run System Test	Android Emulator / Real Device	QA Team
Testing tracking	GitHub, Excel	Project Team

3.2 Test Milestones

Milestone Task	Start Date	End Date
Create test plan	06/8/2025	09/8/2025
Create UAT test cases	22/8/2025	24/8/2025
Create ST test cases	25/8/2025	29/8/2025
Create and update UAT test cases	06/9/2025	08/9/2025
Create and update ST test cases	09/9/2025	10/9/2025
Create IT test cases	11/9/2025	12/9/2025
Create UT test cases	12/9/2025	13/9/2025
Create, update, and execute UT test cases	28/9/2025	03/10/2025
Create, update, and execute IT test cases	28/11/2025	03/11/2025
Create, update, and execute ST test cases	01/11/2025	03/11/2025

Create, update, and execute UAT test cases	01/11/2025	03/11/2025
Create, update, and execute UT test cases	14/11/2025	17/11/2025
Create, update, and execute IT test cases	14/11/2025	17/11/2025
Create, update, and execute ST test cases	14/11/2025	17/11/2025
Create, update, and execute UAT test cases	14/11/2025	17/11/2025
Create, update, and execute UT test cases	30/11/2025	06/12/2025
Create, update, and execute IT test cases	30/11/2025	06/12/2025
Create, update, and execute ST test cases	30/11/2025	06/12/2025
Create, update, and execute UAT test cases	30/11/2025	06/12/2025
Full system testing	04/12/2025	06/12/2025
Create Test Report	04/12/2025	06/12/2025

4. Test Cases

- Unit Test: [SEP490_G151_UnitTest](#)
- Integration test: [SEP490_G151_IntegrationTest](#)
- System test: [SEP490_G151_SystemTest](#)

5. Test Reports

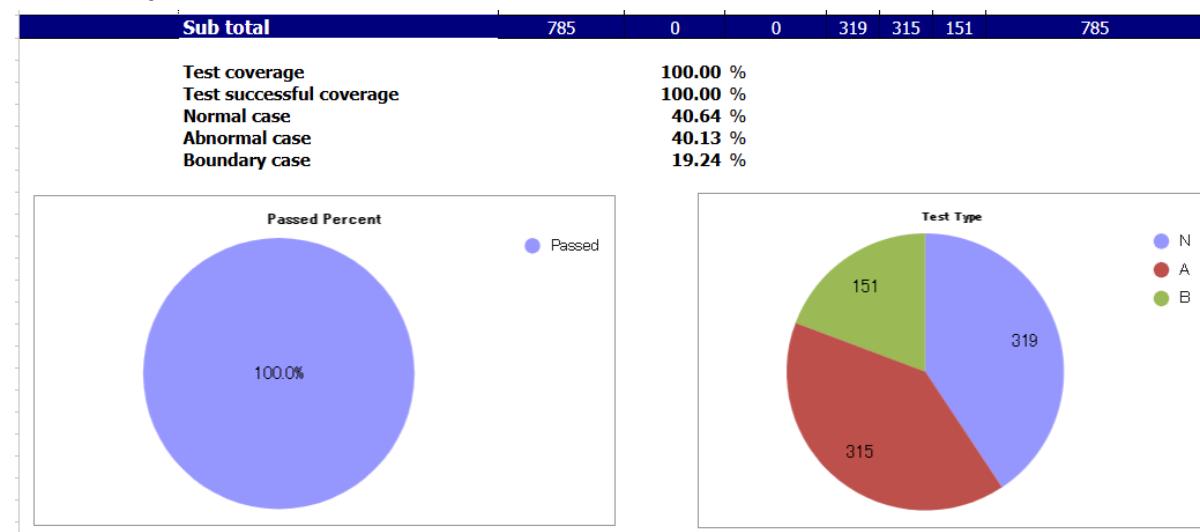


Figure 5.1: Unit Test Statistic

No	Module code	Passed	Failed	Pending	N/A	Number of test cases
1	Address	27	0	0		27
2	Admin	21	0	0		21
3	Attributes	23	0	0		23
4	Attribute Options	17	0	0		17
5	Auth	32	0	0		32
6	User Preference	20	0	0		20
7	Block	11	0	0		11
8	Chat AI	29	0	0		29
9	Chat Expert	21	0	0		21
10	Chat User	16	0	0		16
11	Expert Confirmation	22	0	0		22
12	Match	20	0	0		20
13	Notification	21	0	0		21
14	Payment	19	0	0		19
15	Pet	50	0	0		50
16	Pet Characteristic	34	0	0		34
17	Pet Photo	37	0	0		37
18	Pet Recommendation	25	0	0		25
19	Pet Image Analysis	24	0	0		24
20	Report	28	0	0		28
21	User	51	0	0		51
22	User Preference	20	0	0		20
23	Appointment	43	0	0		43
24	Bad Word	18	0	0		18
25	Event	36	0	0		36
26	Policy	57	0	0		57
		722	0	0	0	722
		Test coverage	100	%		
		Test successful coverage	100	%		

Figure 5.2: Integration Test Statistic

No	Module code	Passed	Failed	Pending	N/A	Number of test cases
1	Address	6	0	0	0	6
2	Admin	13	0	0	0	13
3	Attributes	18	0	0	0	18
4	Attribute Options	17	0	0	0	17
5	Auth	35	0	0	0	35
6	User Preference	14	0	0	0	14
7	Block	15	0	0	0	15
8	Chat AI	27	0	0	0	27
9	Chat Expert	33	0	0	0	33
10	Chat User	31	0	0	0	31
11	Expert Confirmation	23	0	0	0	23
12	Match	36	0	0	0	36
13	Notification	23	0	0	0	23
14	Payment	15	0	0	0	15
15	Pet	26	0	0	0	26
16	Pet Characteristic	12	0	0	0	12
17	Pet Photo	17	0	0	0	17
18	Report	20	0	0	0	20
19	User	17	0	0	0	17
20	Badword	24	0	0	0	24
21	Policy	19	0	0	0	19
22	Event	24	0	0	0	24
23	Appointment	23	0	0	0	23
Sub total		398	0	0	0	398
		Test coverage	100	%		
		Test successful coverage	100	%		

Figure 5.3 System Test Statistic

VI. Release Package & User Guides

1. Deliverable Package

No.	Deliverable Item	Description
Source Code Package		
1	Source code + DB script	Source Code.zip
Documents		
2	Integration Test	Report5_IntegrationTest.xlsx
3	System Test	Report5_SystemTest.xlsx
4	Unit Test	Report5_UnitTest.xlsx
6	Presentation	Presentation Slide.ppt
7	Report 1	Report1_Project Introduction.pdf
8	Report 2	Report2_Project Management Plan.pdf
9	Report 3	Report3_Software Requirement Specification.pdf
10	Report 4	Report4_Software Design Document.pdf
11	Report 5	Report5_Test Documentation.pdf
14	Report 6	Report6_Software User Guides.pdf
15	Report 7	Report7_Final Project Report.pdf

2. Installation Guides

2.1 System Requirements

2.1.1 For Front-end Web Application (Admin/Expert - React)

- Hardware Requirements:
 - CPU with at least 4 cores (Intel i5/AMD Ryzen 5 or higher)
 - RAM with at least 8.00GB (16.00GB recommended)
 - 10GB available disk space
- Operating System:
 - Windows 10/11 64-bit
 - Ubuntu 20.04/22.04 LTS 64-bit
- Software & Tools:
 - Visual Studio Code version ≥ 1.80 or WebStorm version ≥ 2023.1
 - Node.js version ≥ 18.0
 - Npm version ≥ 10.0
 - Git version ≥ 2.43.0 or GitHub Desktop
 - Modern web browser (Chrome ≥ 120, Firefox ≥ 120, Edge ≥ 120)
- Framework & Libraries:
 - ReactJS version ≥ 18.0 (Project uses 19.2.0)
 - React Router DOM version ≥ 6.30

- Axios version \geq 1.13.0
- @microsoft/signalr version \geq 10.0.0
- Recharts version \geq 3.3.0

2.1.2 For Front-end Mobile Application (User App - React Native)

- Hardware Requirements:
 - CPU with at least 8 cores (Intel i7/AMD Ryzen 7 or higher recommended)
 - RAM with at least 16.00GB
 - 50GB available disk space (for Android Studio/Xcode)
- Operating System:
 - For Android Development:
 - Windows 10/11 64-bit
 - Ubuntu 20.04/22.04 LTS 64-bit
- Software & Tools:
 - Visual Studio Code version \geq 1.80 or WebStorm version \geq 2023.1
 - Node.js version \geq 18.0
 - Npm version \geq 10.0
 - Git version \geq 2.43.0 or GitHub Desktop
 - For Android:
 - Android Studio version \geq 2023.1 (Hedgehog)
 - Java JDK 17
 - Android SDK (API Level 33 or higher)
 - Android Emulator or Physical Device (Android 8.0+)
- Framework & Libraries:
 - React Native version 0.74.3
 - TypeScript version \geq 5.0
 - @reduxjs/toolkit version \geq 2.9.0
 - @react-navigation/native version \geq 7.1.18
 - @react-navigation/native-stack version \geq 7.3.27
 - Axios version \geq 1.12.2
 - @microsoft/signalr version \geq 9.0.6
 - React Native Keychain version \geq 10.0.0
 - React Native Vector Icons version \geq 10.3.0

2.1.3 For Back-end Application

- Hardware Requirements:
 - CPU with at least 4 cores (Intel i5/AMD Ryzen 5 or higher)
 - RAM with at least 8.00GB (16.00GB recommended)
 - 20GB available disk space
- Operating System:
 - Windows 10/11 64-bit
 - macOS 11 (Big Sur) or later
 - Ubuntu 20.04/22.04 LTS 64-bit
- Software & Tools:
 - .NET 8.0 SDK or later
 - Visual Studio 2022 version \geq 17.8 or Visual Studio Code with C# extension or JetBrains Rider version \geq 2023.3
 - Git version \geq 2.43.0 or GitHub Desktop

- Postman or Swagger (API testing)
- Framework & Dependencies:
 - ASP.NET Core 8.0
 - Entity Framework Core version ≥ 9.0
 - Npgsql.EntityFrameworkCore.PostgreSQL version ≥ 9.0
 - Microsoft.AspNetCore.SignalR version ≥ 1.2.0
 - Microsoft.AspNetCore.Authentication.JwtBearer version ≥ 8.0.0
 - CloudinaryDotNet version ≥ 1.27.8
 - BCrypt.Net-Next version ≥ 4.0.3
 - System.IdentityModel.Tokens.Jwt version ≥ 8.14.0
 - Swashbuckle.AspNetCore version ≥ 6.6.2

2.2 Installation Instruction

2.2.1 Installation Front-end Web Application (Admin/Expert Web)

- Dev Environment
 - Install Node.js v18.0.0 or higher at <https://nodejs.org/en/download/>
 - Install Visual Studio Code at <https://code.visualstudio.com/download>
 - Install Git at <https://git-scm.com/downloads> OR Install GitHub Desktop at <https://desktop.github.com/>
 - Verify installations:
 - `node --version`
 - `npm --version`
 - `git --version`
- Project Setup
 - Clone the repository:
 - Using Git command line:
 - `git clone`
https://github.com/your-repo/SEP490_G151-Pawnder-Pet-Dating-App.git
 - `cd SEP490_G151-Pawnder-Pet-Dating-App`
 - Using GitHub Desktop:
 - Open GitHub Desktop
 - File → Clone Repository
 - URL tab → Enter repository URL
 - Choose local path → Click "Clone"
 - Navigate to frontend admin directory: `cd FE/fe-admin`
 - Install dependencies: `npm install`
 - Open the project in Visual Studio Code: `code .`
- Configuration
 - Open file src/shared/constants/index.js and update API_BASE_URL:
`"export const API_BASE_URL = 'http://localhost:5297';"`
- Run Application
 - Start development server: `npm start`
 - The application will open automatically at `http://localhost:3000`

2.2.2 Installation front-end

- Dev Environment - Common

- Install Node.js v18.0.0 or higher at <https://nodejs.org/en/download/>
 - Install Visual Studio Code at <https://code.visualstudio.com/download>
 - [Install Git at https://git-scm.com/downloads](https://git-scm.com/downloads) OR Install GitHub Desktop at <https://desktop.github.com/>
- Dev Environment - Android
 - Install Java JDK 17 at <https://www.oracle.com/java/technologies/downloads/>
 - Install Android Studio at <https://developer.android.com/studio>
 - Open Android Studio → Settings → Android SDK
 - Install:
 - Android SDK Platform 33 (Android 13.0)
 - Android SDK Build-Tools
 - Android Emulator
 - Set environment variables:
 - Windows:
 - ANDROID_HOME = C:\Users\YourUsername\AppData\Local\Android\Sdk
 - Add to PATH: %ANDROID_HOME%\platform-tools
 - Mac/Linux:
 - Add to ~/.zshrc or ~/.bash_profile:
 - export ANDROID_HOME=\$HOME/Library/Android/sdk
 - export PATH=\$PATH:\$ANDROID_HOME/platform-tools
 - Verify installation:
 - adb --version
- Dev Environment - iOS (macOS only)
 - Install Xcode from Mac App Store (version 15.0 or higher)
 - Install Xcode Command Line Tools: `xcode-select --install`
 - Install CocoaPods: `sudo gem install cocoapods`
 - Verify installation: `pod --version`
- Project Setup
 - Clone the repository (if not already done):
 - Using Git command line:
 - `git clone https://github.com/your-repo/SEP490_G151-Pawnder-Pet-Dating-App.git`
 - `cd SEP490_G151-Pawnder-Pet-Dating-App`
 - Using GitHub Desktop:
 - Open GitHub Desktop
 - File → Clone Repository
 - Clone the repository to your local machine
 - Navigate to mobile app directory: `cd FE/FE-User`
 - Install dependencies: `npm install`
 - For iOS, install pods:
 - `cd ios`
 - `pod install`
 - `cd ..`
 - Open the project in Visual Studio Code: `code .`
- Configuration
 - Open file `src/config/api.config.ts` and update configuration:
 - For Android Emulator:

- const CURRENT_ENVIRONMENT: Environment = 'android';
 - ANDROID_EMULATOR: 'http://10.0.2.2:5297'
- For iOS Simulator:
 - const CURRENT_ENVIRONMENT: Environment = 'ios';
 - IOS_SIMULATOR: 'http://localhost:5297'
- Run Application - Android
- Start Metro bundler in Terminal 1: *npm start*
- Start Android Emulator via Android Studio or run: *emulator -avd Pixel_5_API_33*
- Run the app in Terminal 2: *npm run android*
- Run Application - iOS (macOS only)
 - Start Metro bundler: *npm start*
 - Run the app in Terminal 2: *npm run ios*

2.2.3 Installation front-end

- Dev Environment
 - Install .NET 8.0 SDK at <https://dotnet.microsoft.com/download/dotnet/8.0>
 - Install Visual Studio 2022 at <https://visualstudio.microsoft.com/downloads/> OR
Install Visual Studio Code at <https://code.visualstudio.com/download> (If using VS Code, install C# extension)
 - Install Git at <https://git-scm.com/downloads> OR Install GitHub Desktop at <https://desktop.github.com/>
 - Verify installation:
 - dotnet --version
 - Git --version
- Project Setup
 - Clone the repository (if not already done):
 - Using Git command line:
 - git clone https://github.com/your-repo/SEP490_G151-Pawnder-Pet-Dating-App.git
 - cd SEP490_G151-Pawnder-Pet-Dating-App
 - Using GitHub Desktop:
 - Open GitHub Desktop
 - File → Clone Repository
 - Clone the repository to your local machine
 - Navigate to backend directory: *cd BackEnd/BE*
 - Restore NuGet packages: *dotnet restore*
 - Open the project:
 - Visual Studio 2022: Open file BE.sln
 - Visual Studio Code: code .
- Configuration
 - Open file appsettings.json and update with your credentials:
 - Database Connection:

```
"ConnectionStrings": {  
    "DbContext": "your-connection-string"  
}
```

JWT Settings:

```
"Jwt": {  
    "Issuer": "your-issuer",  
    "Audience": "your-audience",  
    "Secret": "your-secret-key",  
    "AccessTokenExpirationMinutes": "30",  
    "RefreshTokenExpirationDays": "30"  
}
```

Email Settings:

```
"EmailSettings": {  
    "SmtpServer": "smtp.gmail.com",  
    "SmtpPort": 587,  
    "SenderEmail": "your-email@gmail.com",  
    "SenderPassword": "your-app-password"  
}
```

Cloudinary Settings:

```
"Cloudinary": {  
    "CloudName": "your-cloud-name",  
    "ApiKey": "your-cloudinary-api-key",  
    "ApiSecret": "your-cloudinary-api-secret",  
    "Folder": "pawnder/pets"  
}
```

Gemini AI:

```
"GeminiAI": {  
    "ApiKey": "your-gemini-api-key"  
}
```

Payment Settings:

```
"VietQr": {  
    "ApiKey": "your-vietqr-api-key",  
    "ClientId": "your-client-id",  
    "AccountInfo": {  
        "AccountNo": "your-account-number",  
        "AccountName": "YOUR ACCOUNT NAME",  
        "AcqId": "your-bank-code",  
        "Template": "compact"  
    }  
}
```

Payment Settings:

```
"Sepay": {  
    "ApiKey": "your-seipay-api-key",  
    "AccountNumber": "your-account-number"  
}
```

Email Verification:

```
"KickboxSettings": {  
    "ApiKey": "your-kickbox-api-key"  
}
```

Location Service:

```
"LocationIQ": {  
    "ApiKey": "your-locationiq-api-key"  
}
```

- Run Application

- Using Visual Studio 2022:
 - Press F5 or click "Start Debugging"
 - Application will run on <https://localhost:5297>
- Using Visual Studio Code or Command Line: *dotnet run*
- The API will be available at:
 - <https://localhost:5297>
 - Swagger UI: <https://localhost:5297/swagger>

3. User Manual

3.1 Overview

The "Pawnder - Pet Dating App" is a specialized mobile platform designed for cat owners, providing a comprehensive experience by combining social networking and AI technology. The system enables automatic AI-powered pet profile creation, intelligently suggests breeding matches based on health, characteristics, and location. Additionally, the system integrates an intelligent AI chatbot to help users address issues related to cat care, nutrition, health, and training. The platform also fosters a safe community where users can share experiences, chat, and receive direct consultation from certified experts. Pawnder serves as a smart, reliable, and modern solution that comprehensively addresses the fragmentation and lack of transparency in existing tools for cat owners.

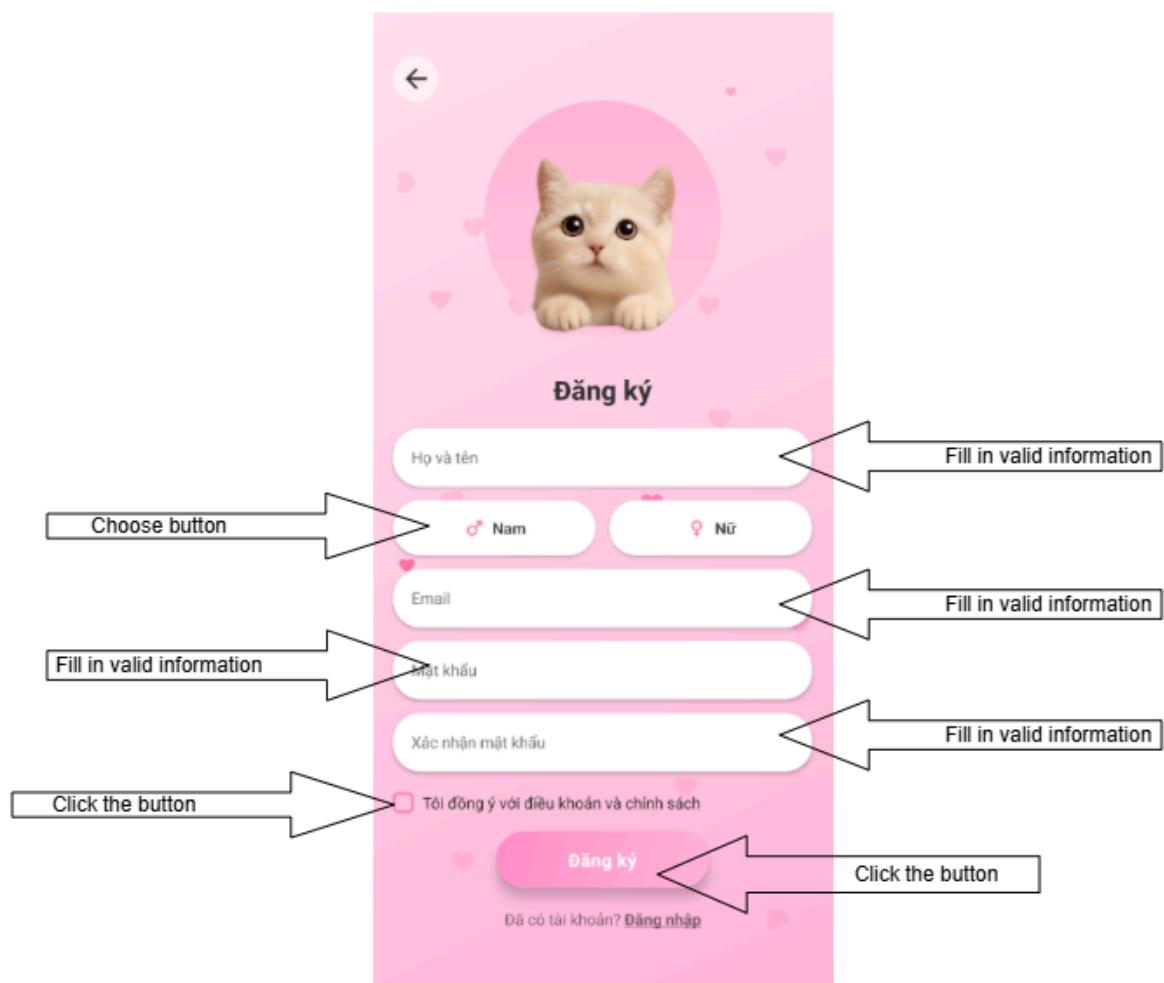
Key Features:

- User Authentication & Profile Management - Secure registration, login, and comprehensive user profile management with multiple pet profiles support.
- Pet Profile Management - Create and manage detailed pet profiles including photos, characteristics, breed information, and personality traits.
- AI-powered Image Analysis - Automatic pet breed recognition and characteristic detection from uploaded photos using AI technology.
- Smart Matching System - AI-powered recommendation algorithm that matches pets based on user preferences, pet characteristics, and geographic location.
- Real-time Communication - Instant messaging between matched users with real-time notifications and chat history.
- AI Pet Advisor - Integrated AI chatbot providing pet care advice, tips, and general consultation.
- Expert Consultation - Direct communication channel with certified pet experts and veterinarians for professional guidance.
- Daily Usage Limits - Fair usage system with daily limits on matches and messages for free users, upgradeable through premium membership.
- Premium Membership - VIP subscription system with enhanced features including unlimited matches, priority support, and advanced visibility.
- Safety & Moderation - Comprehensive reporting system, user blocking capabilities, and administrative content moderation tools.
- Payment Integration - Secure payment processing via QR code (VietQR) for premium subscriptions.
- Administrative Dashboard - Web-based management panel for monitoring users, pets, reports, payments, and platform analytics.
- Notification System - Real-time push notifications for matches, messages, and important updates.

3.2.1. Authentication & Account Management

[**3.2.1.1. User Registration**](#)

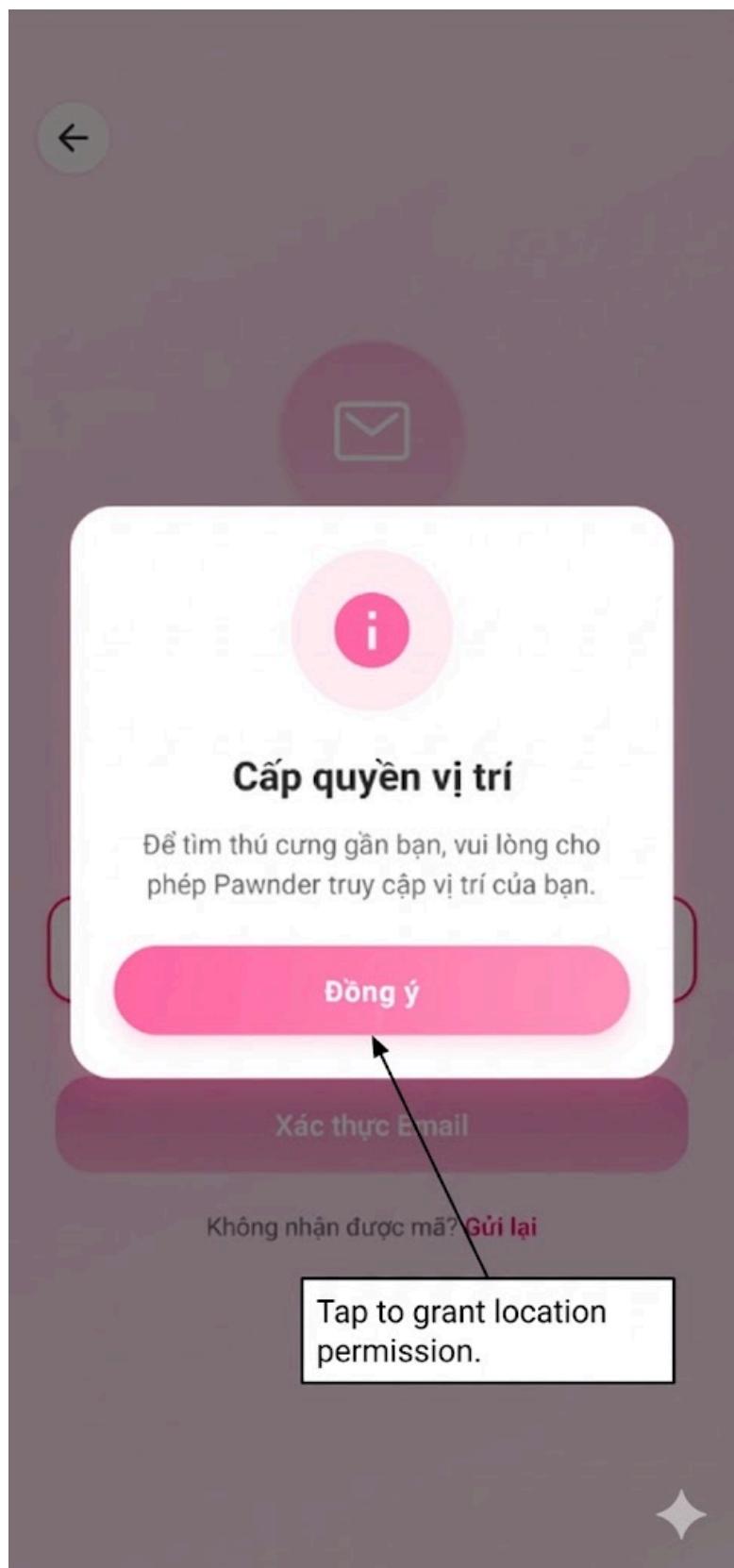
Step 1:



Step 2:



Step 3:



Step 4:



Go back to the previous screen.

BƯỚC 1 / 3

Hồ sơ thú cưng

Hãy cho chúng tôi biết về người bạn lông xù của bạn

Tên

Enter your pet's name.

LyLy



Giống

Enter your pet's breed.

Nga

Tùy chọn

Giới thiệu

Write a short introduction about your pet.

Để thường nghe

Tùy chọn • 21/200

Tap to continue to the next step.

Tiếp tục →

Step 5:



Step 6:

← Go back

BƯỚC 3 / 3

Đặc điểm thú cưng

Giúp mọi người hiểu rõ hơn về thú cưng của bạn

AI analysis result

Phân tích AI hoàn tất!
Đã phát hiện 16 đặc điểm. Hãy xem xét và chỉnh sửa nếu cần.

Hình dạng đầu

- Tròn
- Cân đối
- Dài
- Vuông

Hình dạng mõm

- Ngắn
- Trung bình
- Dài

Màu lông

- Trắng
- Vàng
- Nâu
- Đen
- Xám
- Đỏ
- Bạc
- Xanh

Use AI assistance →

Tap to view AI analysis for this category. →

Trạng thái cơ thể

- Gầy
- Săn chắc
- Cân đối
- Mùm mím
- Béo

Tap to select the pet's body condition. →

Tuổi (năm)

3 → **năm**

Tap to enter the pet's age. →

Giới tính

- Đực
- Cái

Tap to select the pet's gender. →

Chiều cao (cm)

30 → **cm**

Tap to enter the pet's height. →

Thông tin hữu ích

Những thông tin này giúp chúng tôi tìm được bạn đồng hành hoàn hảo cho thú cưng của bạn. Bạn có thể cập nhật bất cứ lúc nào.

Tap to continue to the next step. →

Tiếp tục →

Step 7:



Step 8:



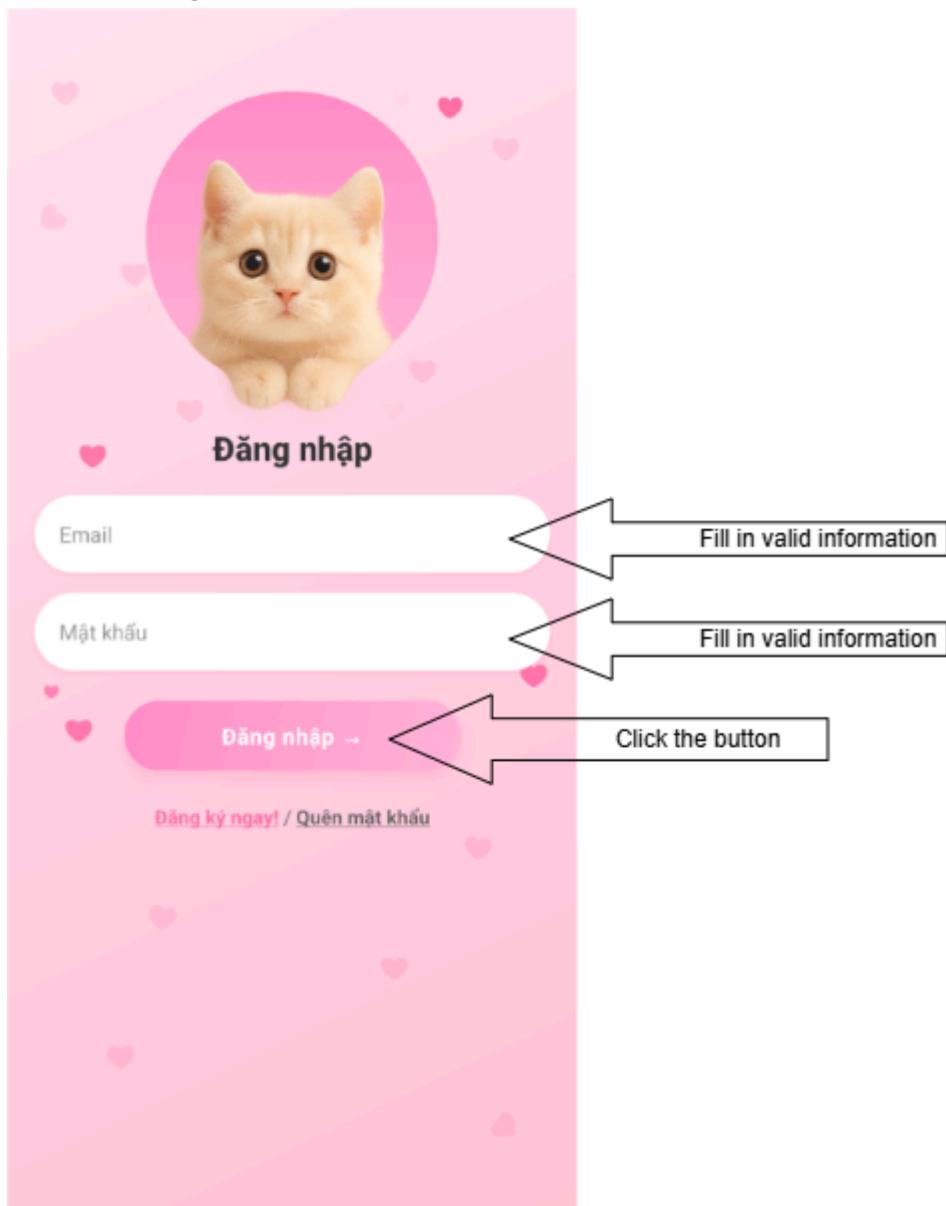
Step 9:



Step 10:

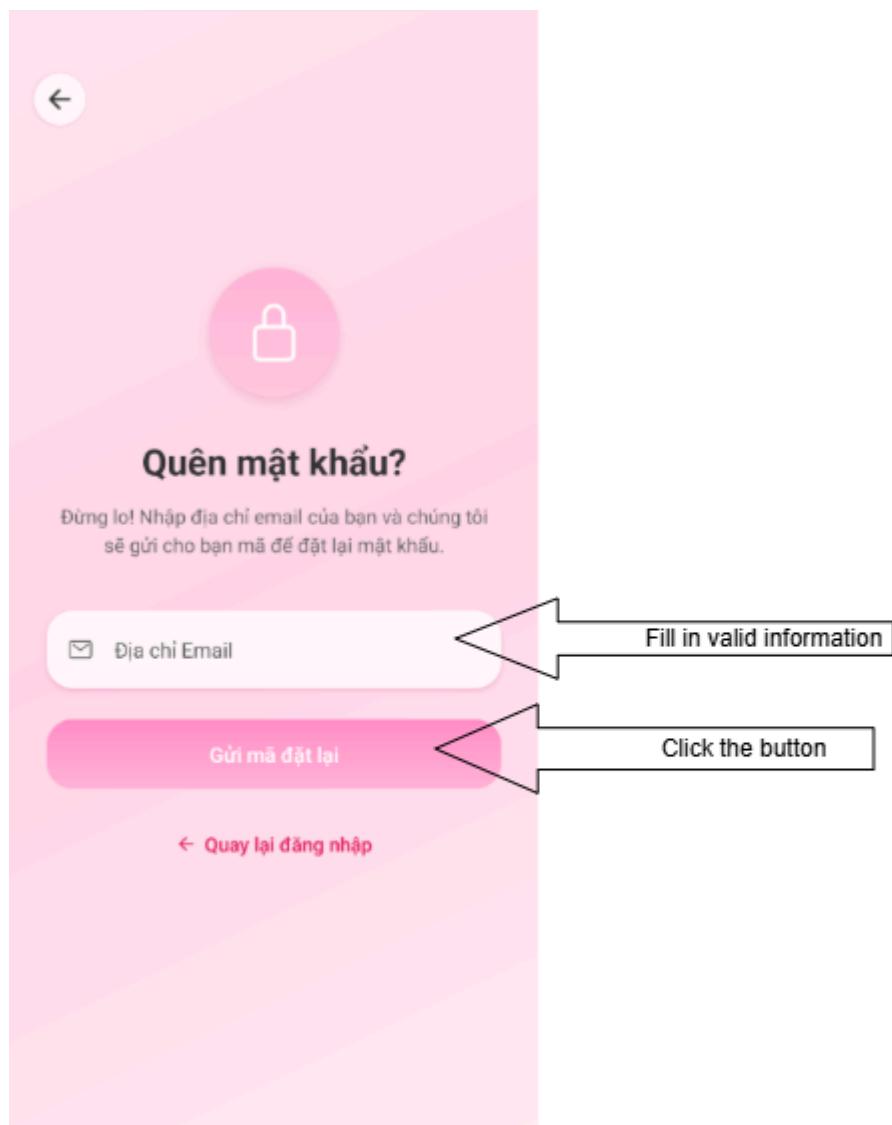


3.2.1.2. User Login



3.2.1.3. Forgot Password

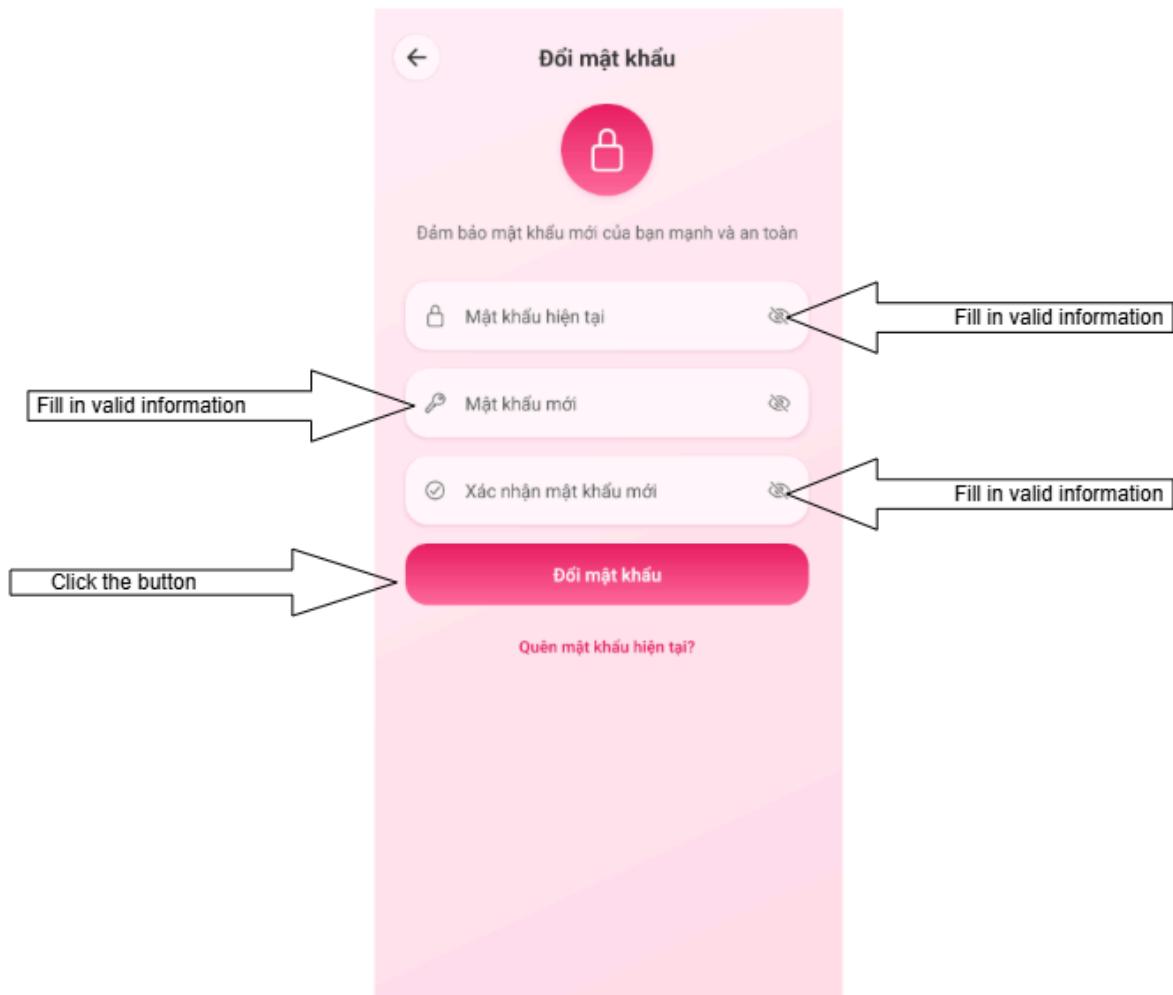
Bước 1:



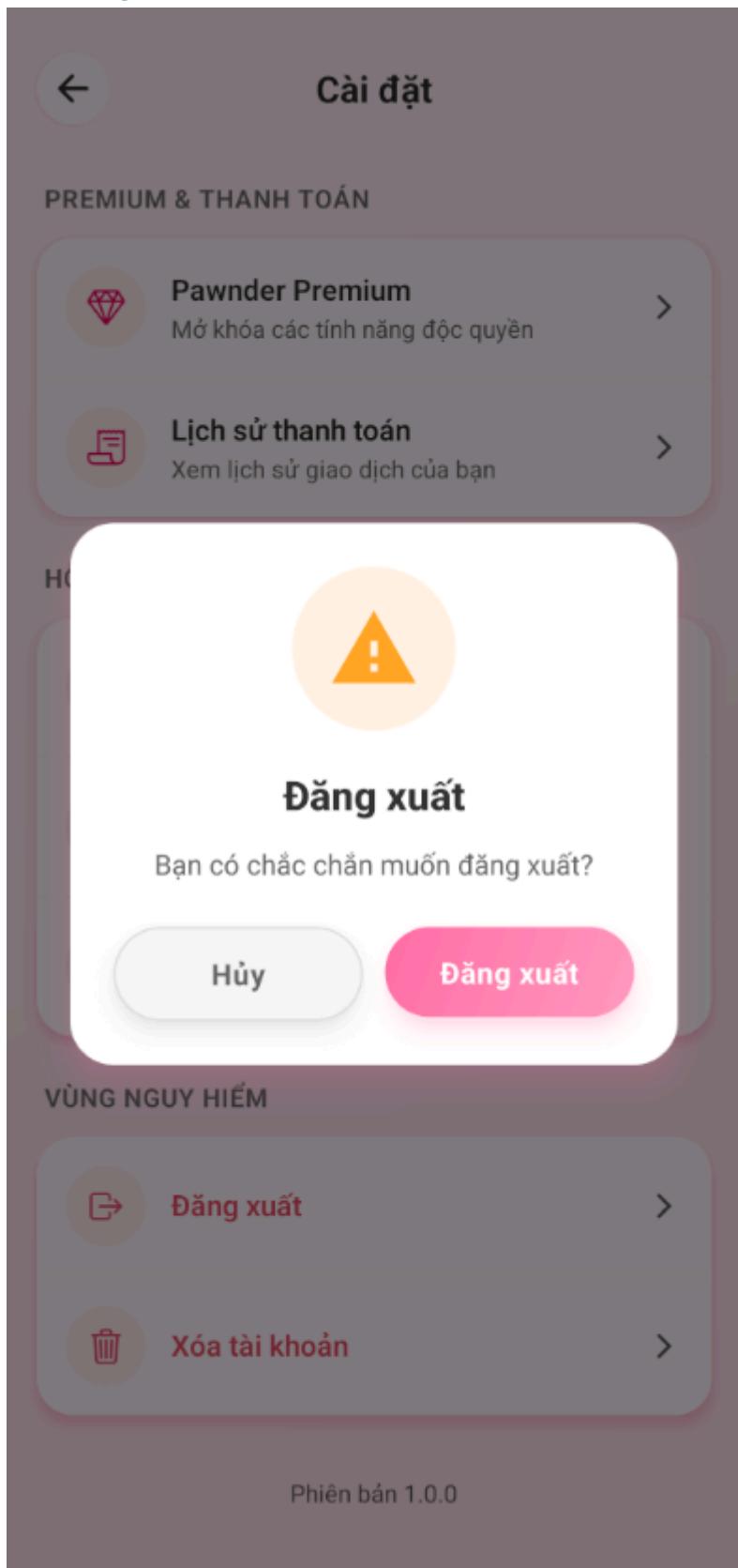
Bước 2:

3.2.1.4. Reset Password

3.2.1.5. Change Password

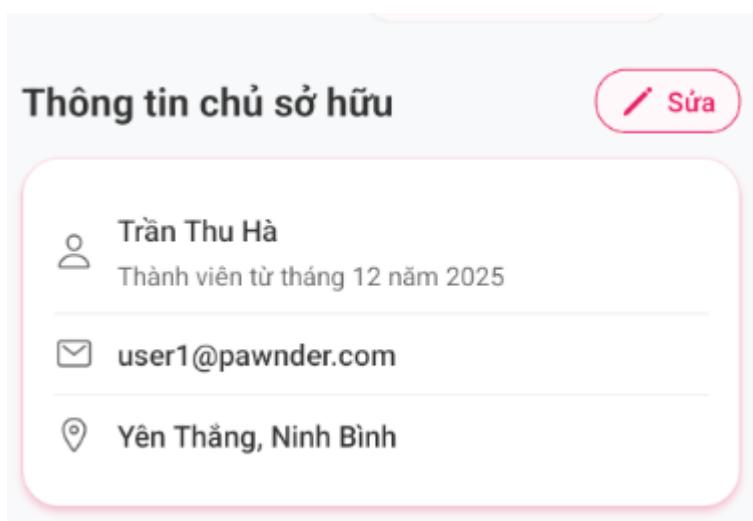


3.2.1.6. Logout



3.2.2. Profile Management

3.2.2.1. View User Profile



3.2.2.2. Edit Personal Information

← Chỉnh sửa hồ sơ

Thông tin cá nhân

Họ và tên

Trần Thu Hà

Email

user1@pawnder.com

Emai không thể thay đổi

Giới tính

Nam

Nữ

Vị trí (GPS)

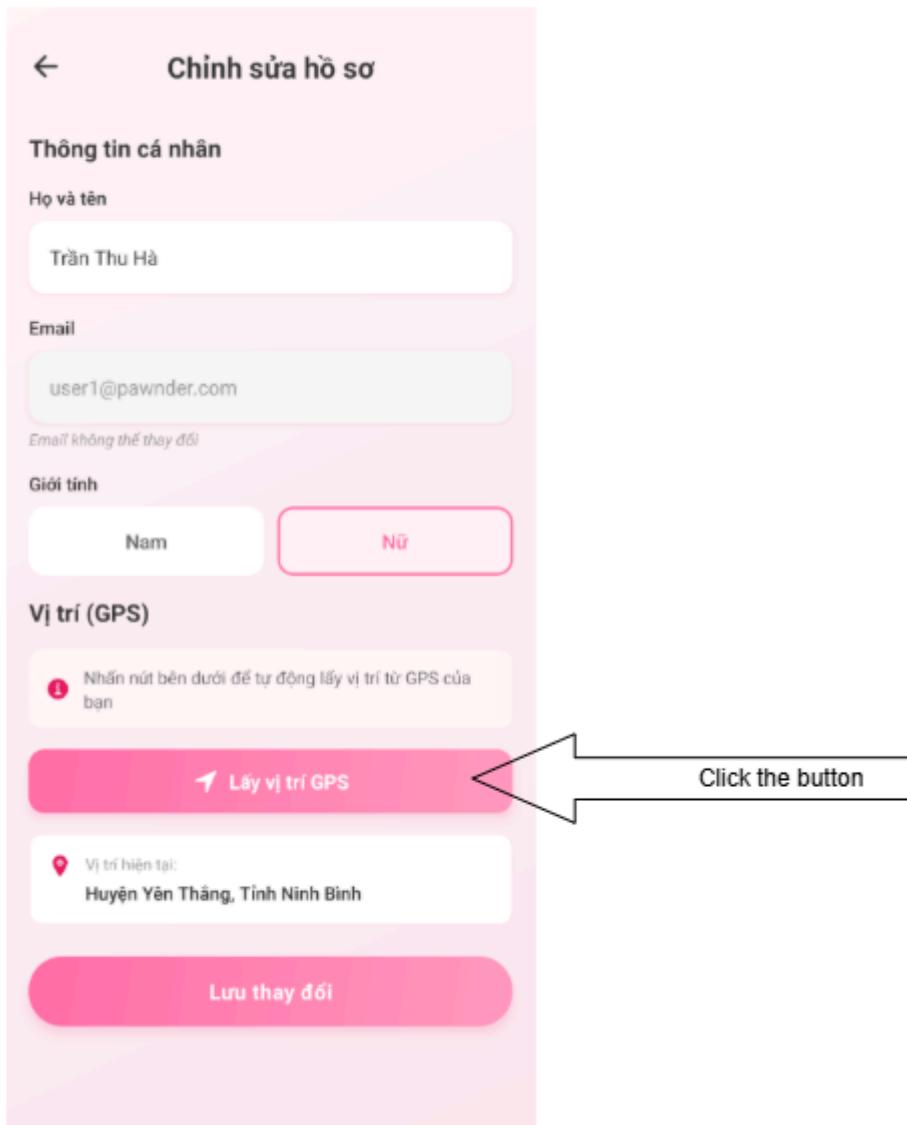
Nhấn nút bên dưới để tự động lấy vị trí từ GPS của bạn

Lấy vị trí GPS

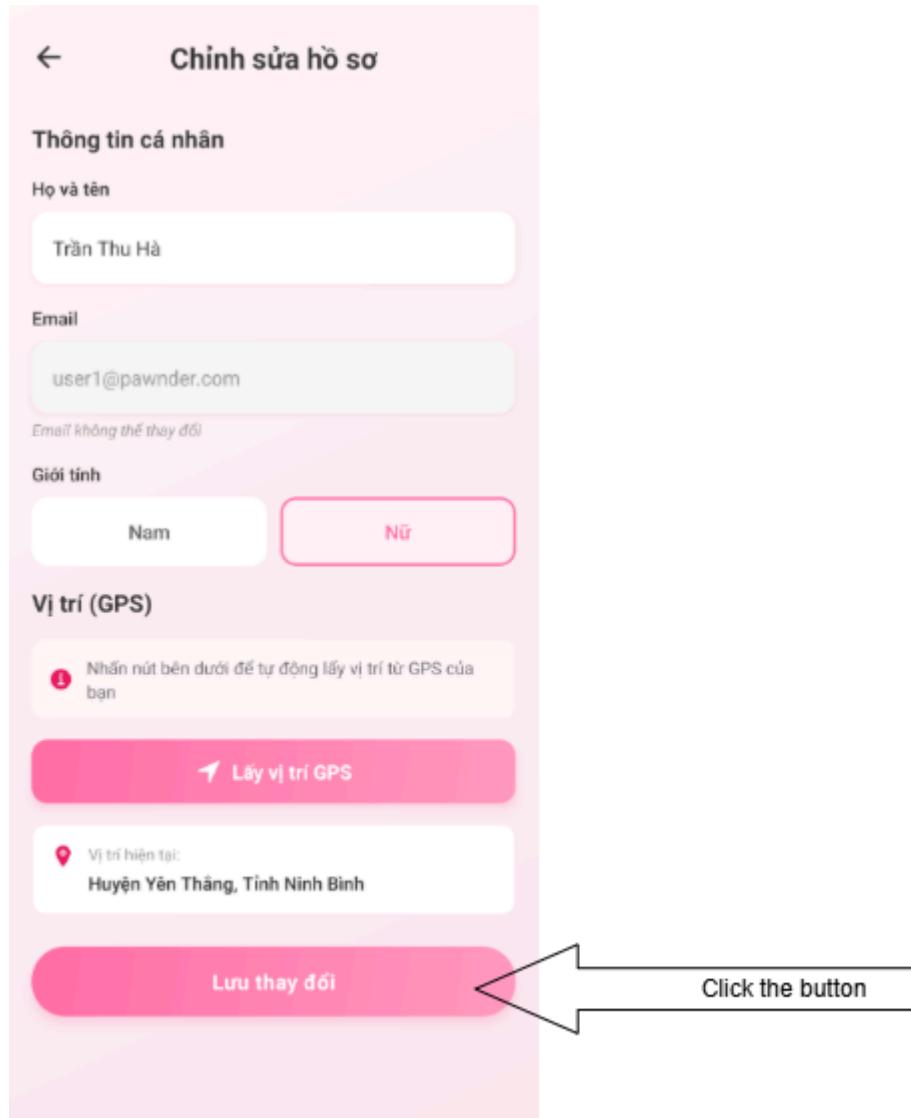
Vị trí hiện tại:
Huyện Yên Thành, Tỉnh Ninh Bình

Lưu thay đổi

3.2.2.3. Update Address/Location



3.2.2.4. Save Personal Information



3.2.3. Pet Management

3.2.3.1. Add New Pet

Bước 1:



Bước 2:



BƯỚC 1 / 3

Hồ sơ thú cưng

Hãy cho chúng tôi biết về người bạn lông xù của bạn

Tên

Tên thú cưng của bạn là gì?

Fill in valid information

Giống

VD: Ba Tư, Anh Lông Ngắn

Fill in valid information

Tùy chọn

Giới thiệu

Mô tả tính cách và đặc điểm của thú cưng...

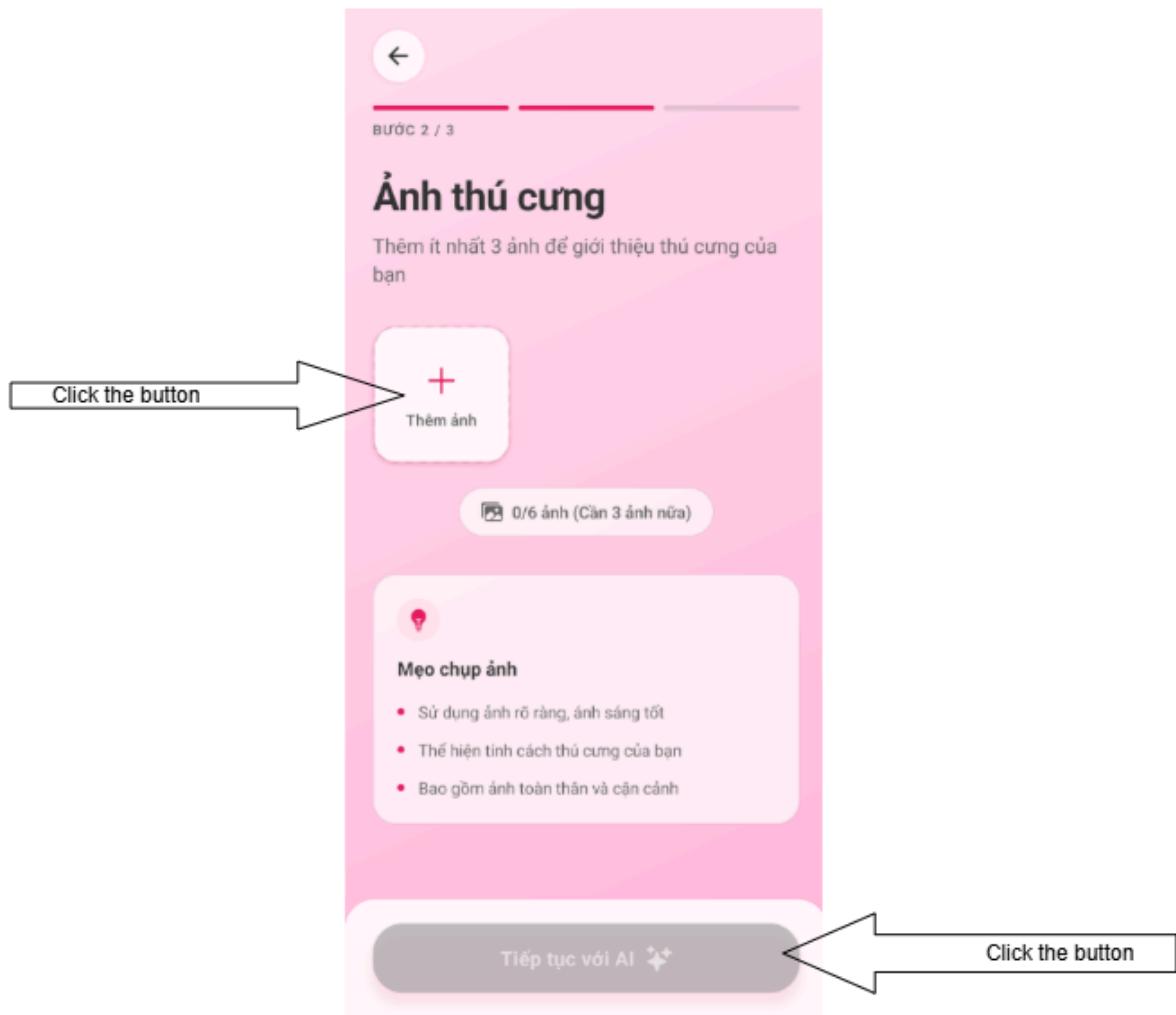
Fill in valid information

Tùy chọn • 0/200

Tiếp tục →

Click the button

3.2.3.2. Upload Pet Photos



3.2.3.4. Edit Pet Basic Information

3.2.3.5. Edit Pet Characteristics

Bước 1:



Bước 2:



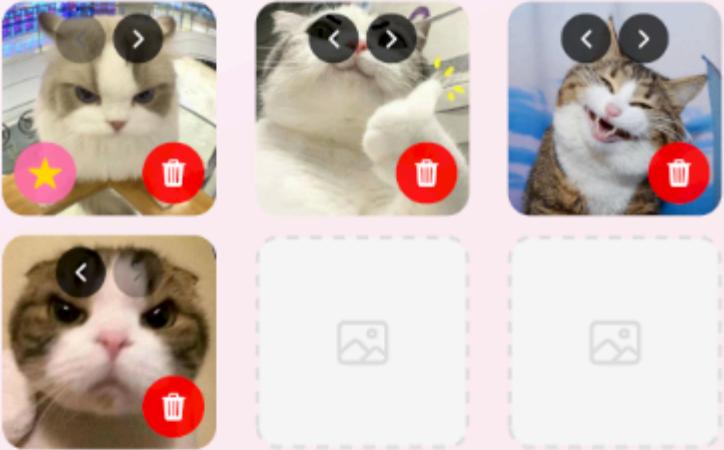
3.2.3.6. View Pet Profile

← **Chỉnh sửa hồ sơ thú cưng**



Đổi ảnh thú cưng

Ảnh thú cưng (4/6) (+)



Thông tin thú cưng

Tên thú cưng

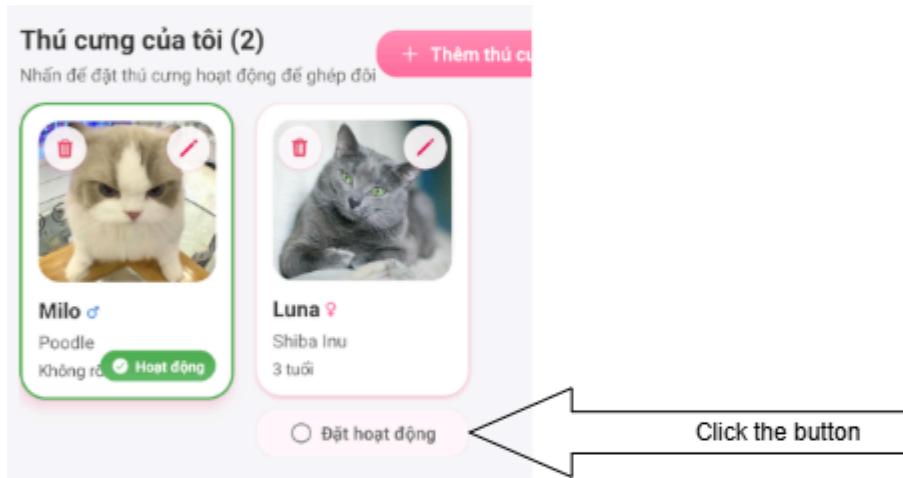
Milo

Giống

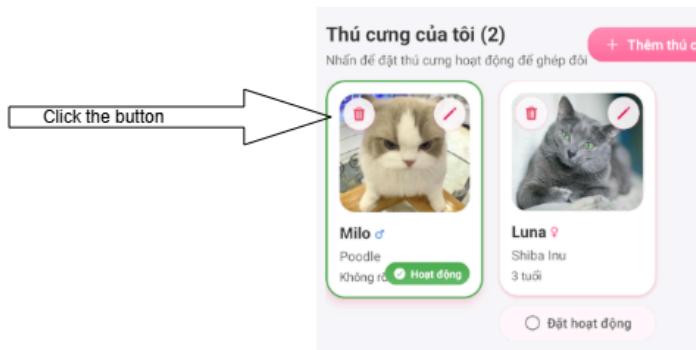
Poodle

Tuổi (năm)

3.2.3.7. Set Active Pet



3.2.3.8. Delete Pet Profile



3.2.4. Matching & Dating

3.2.4.1. Browse Recommended Pets (Home Screen)



3.2.4.2. View Pet Details



Mimi ♀

✿ Chihuahua • 3 tuổi
📍 Tỉnh Hưng Yên

 Chihuahua
Giống

 3 tuổi
Tuổi

 Female
Giới tính

Về Mimi

Tiny dog who loves cuddles.

Đặc điểm 14

Chủ sở hữu

 **Trần Hoàng Phúc**
Member
📍 Tỉnh Hưng Yên

 Gửi yêu cầu ghép đôi

 Chặn

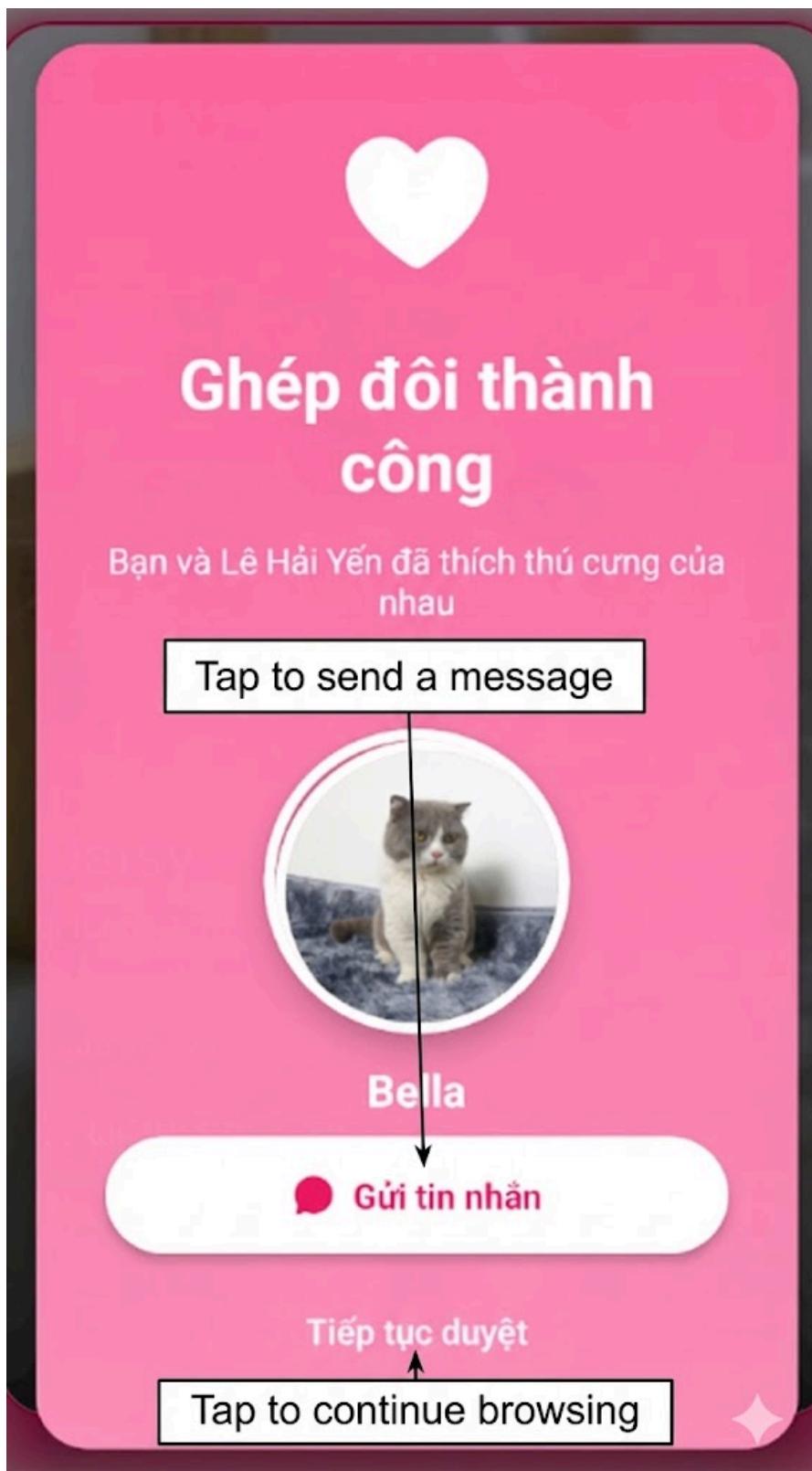
3.2.4.3. Swipe Right (Like)



3.2.4.4. Swipe Left (Pass)

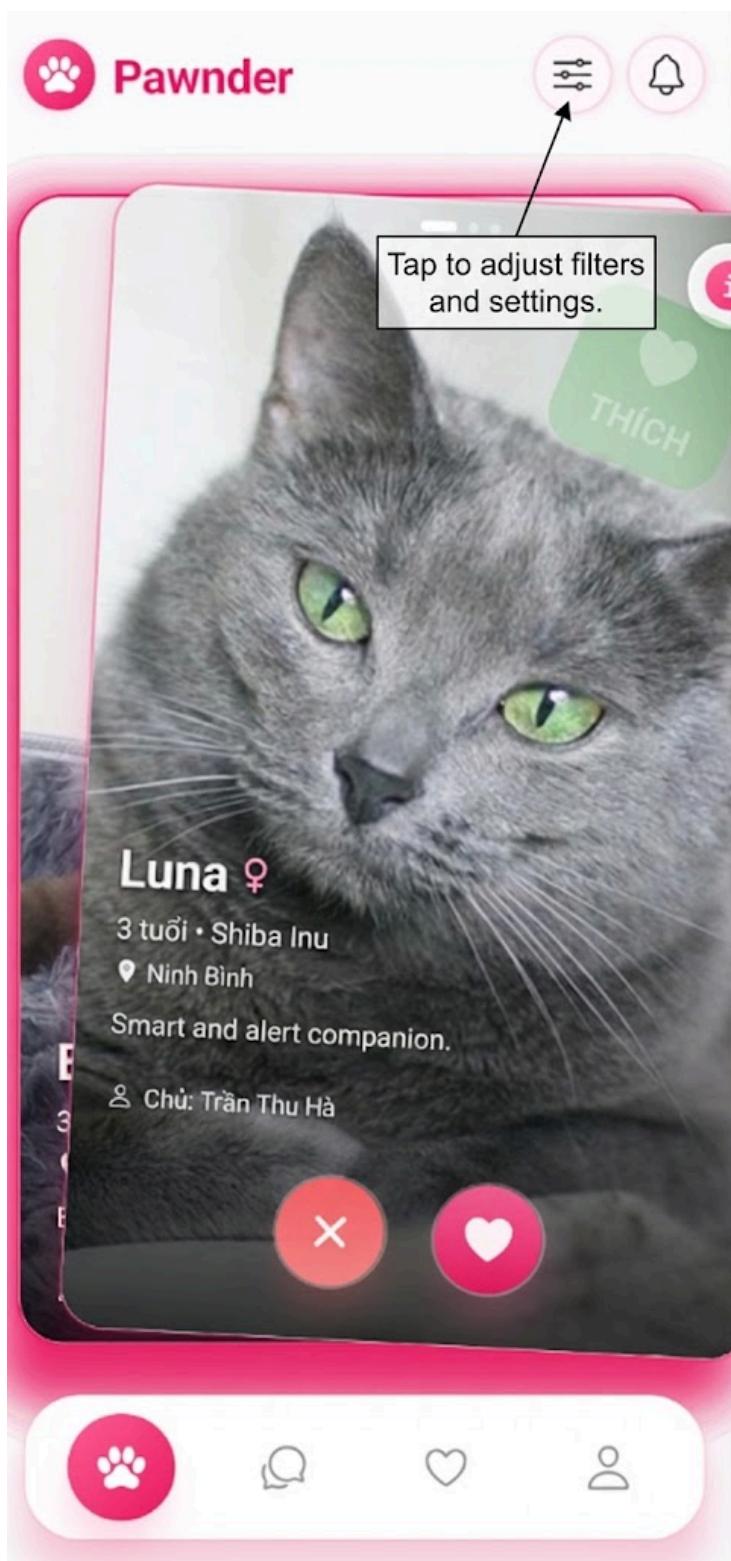


3.2.4.5. View Match Success Popup

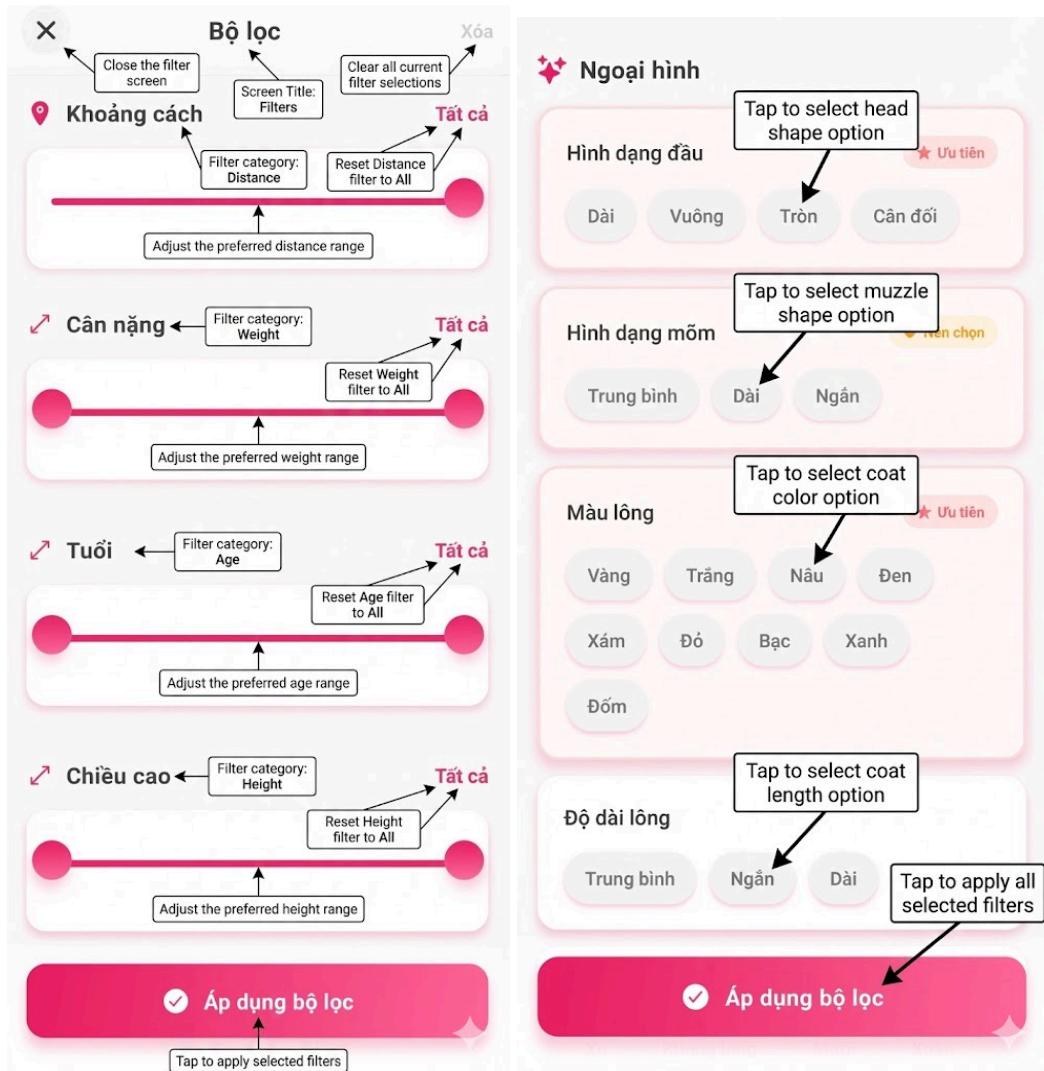


3.2.5. Filter & Preferences

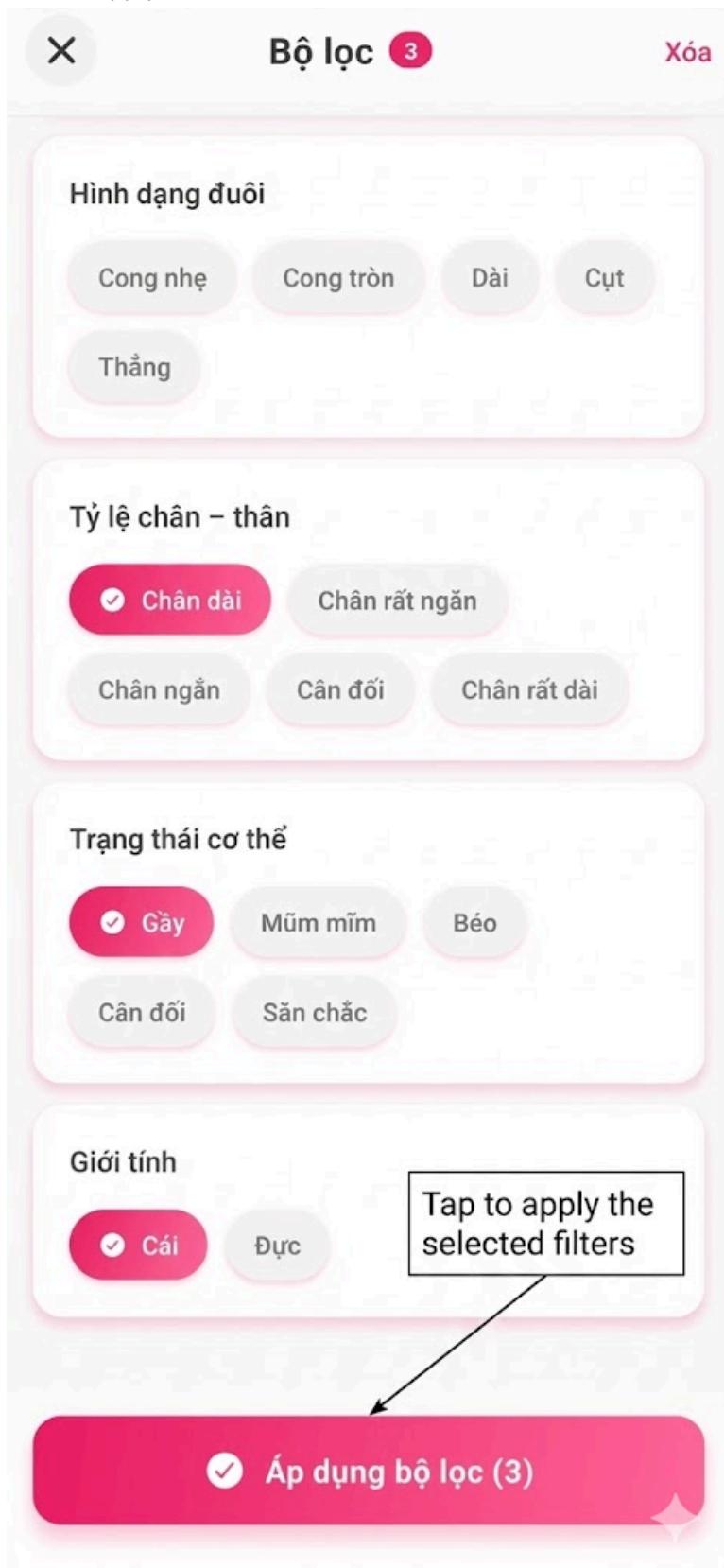
3.2.5.1. Access Filter Screen



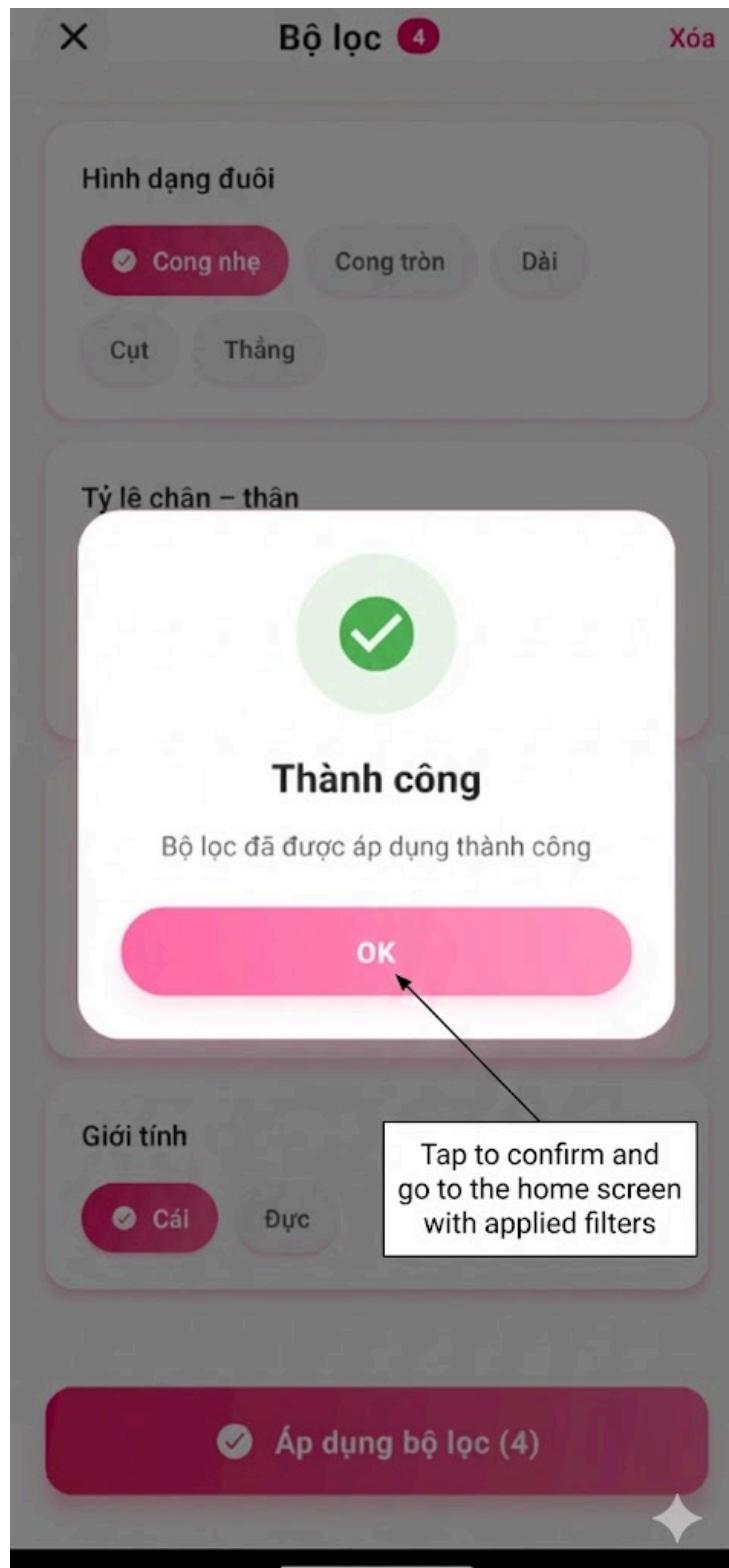
3.2.5.2. Set Matching Preferences



3.2.5.3. Apply Filters

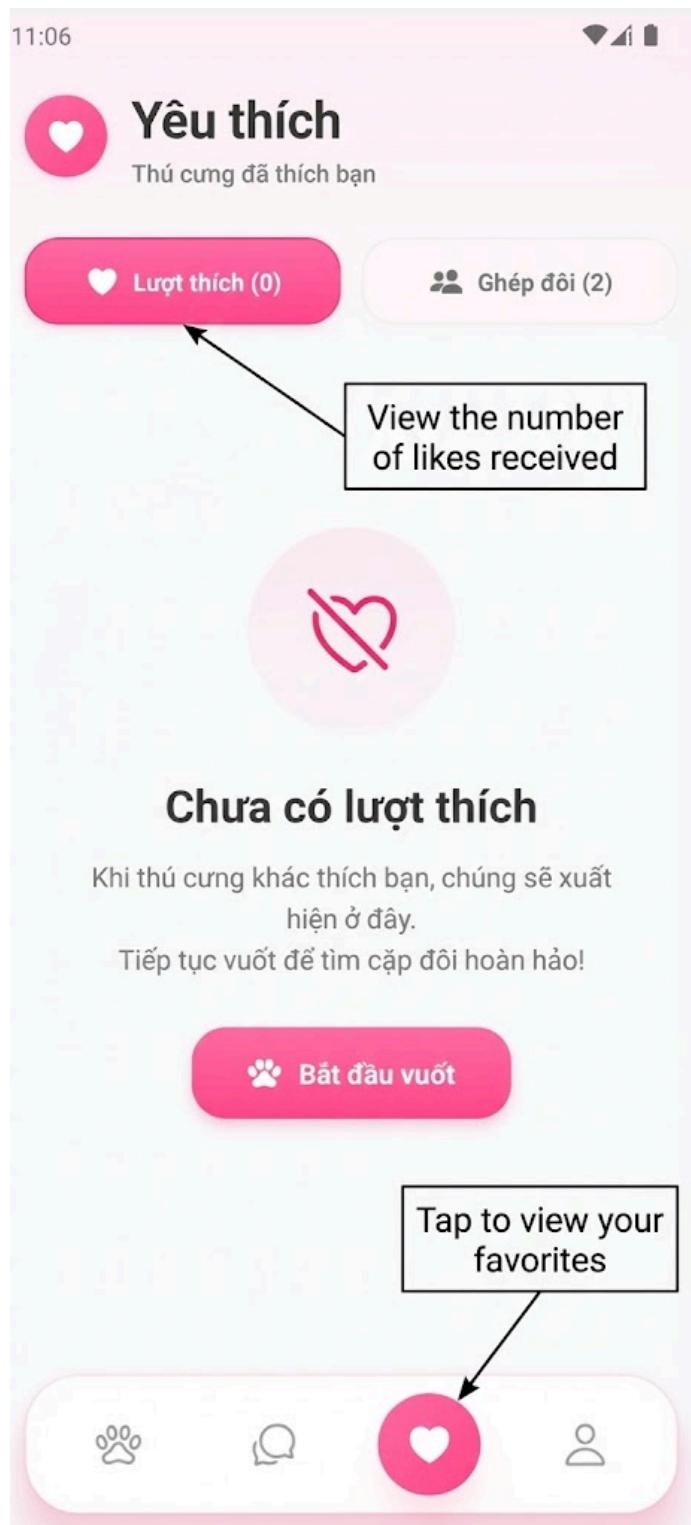


3.2.5.4. Save Preferences

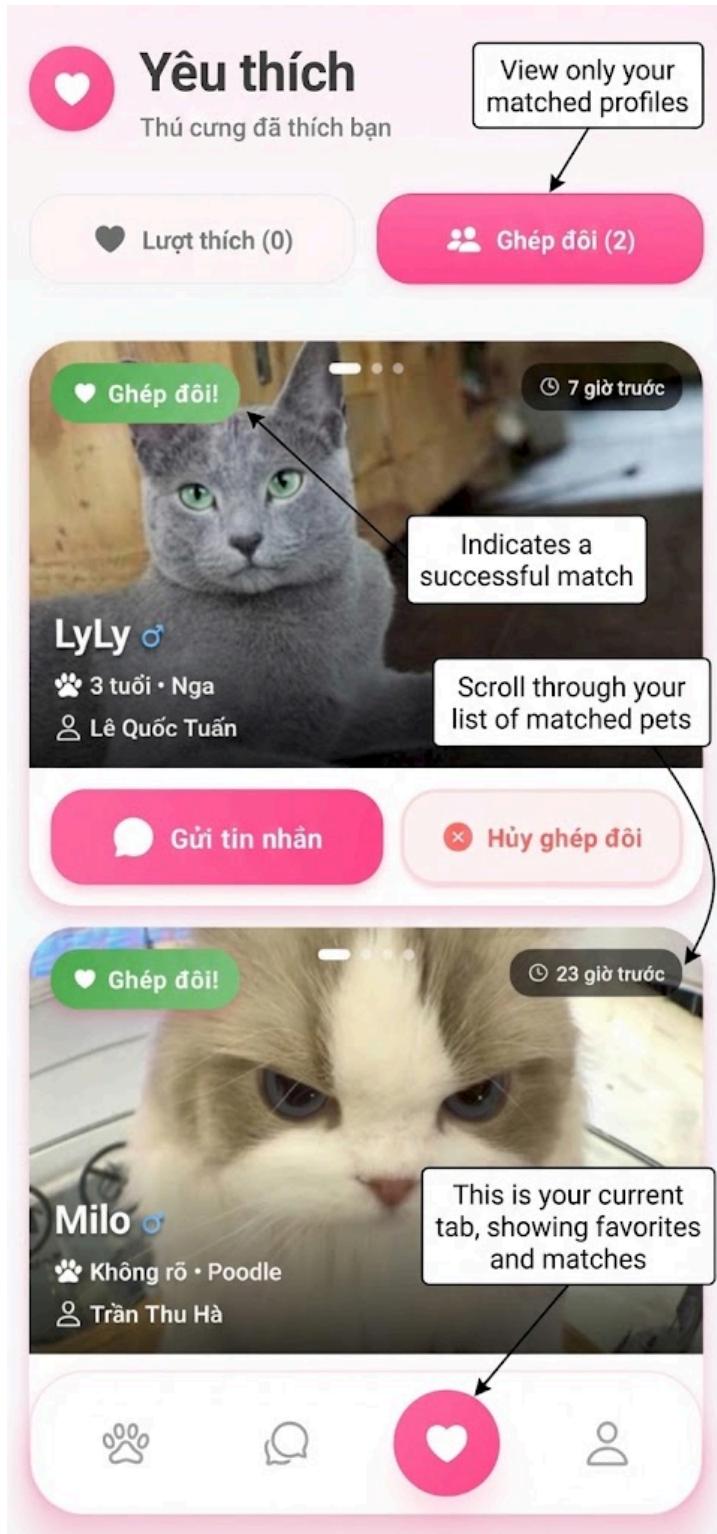


3.2.6. Favorites & Matches

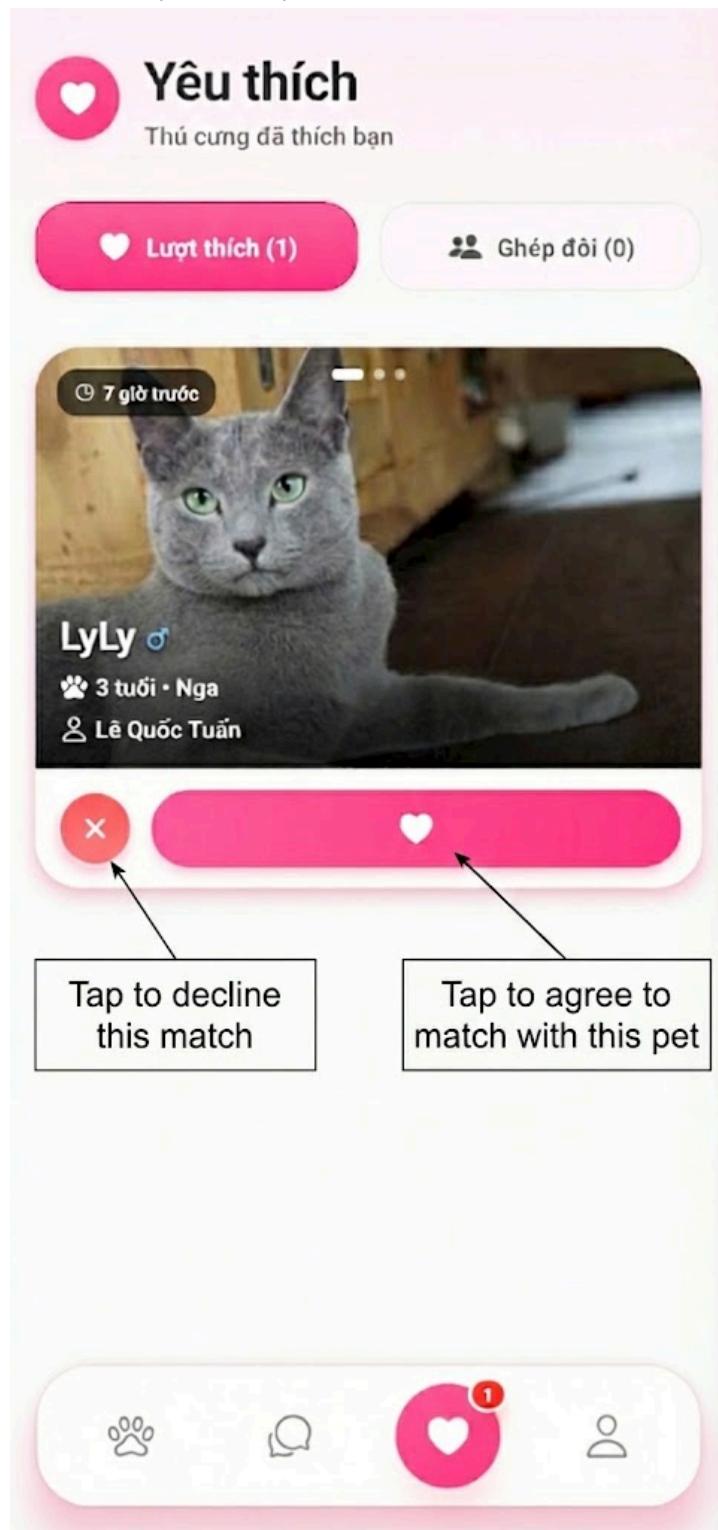
3.2.6.1. View Received Likes (Likes Tab)



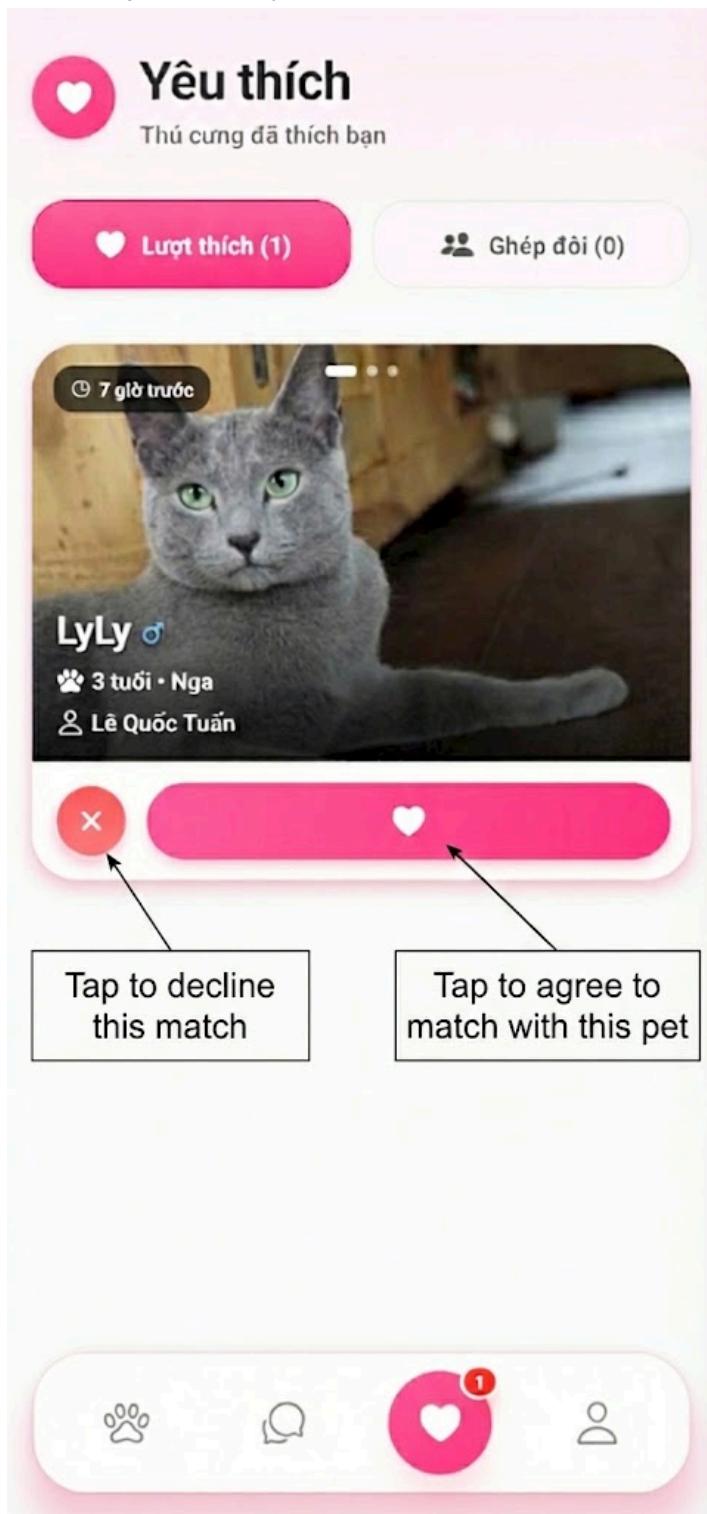
3.2.6.2. View Mutual Matches (Matches Tab)



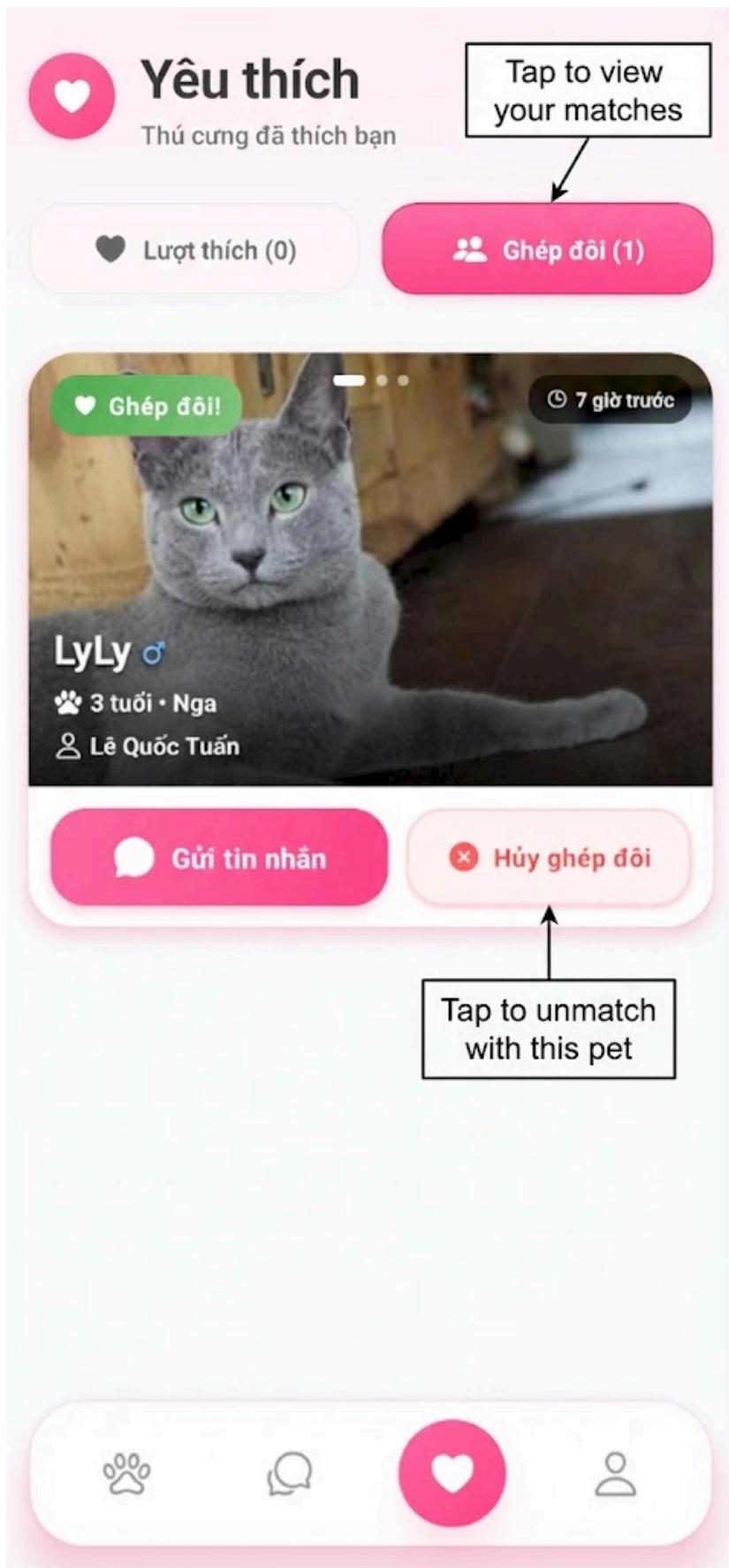
3.2.6.3. Accept Like Request



3.2.6.4. Reject Like Request (Pass)

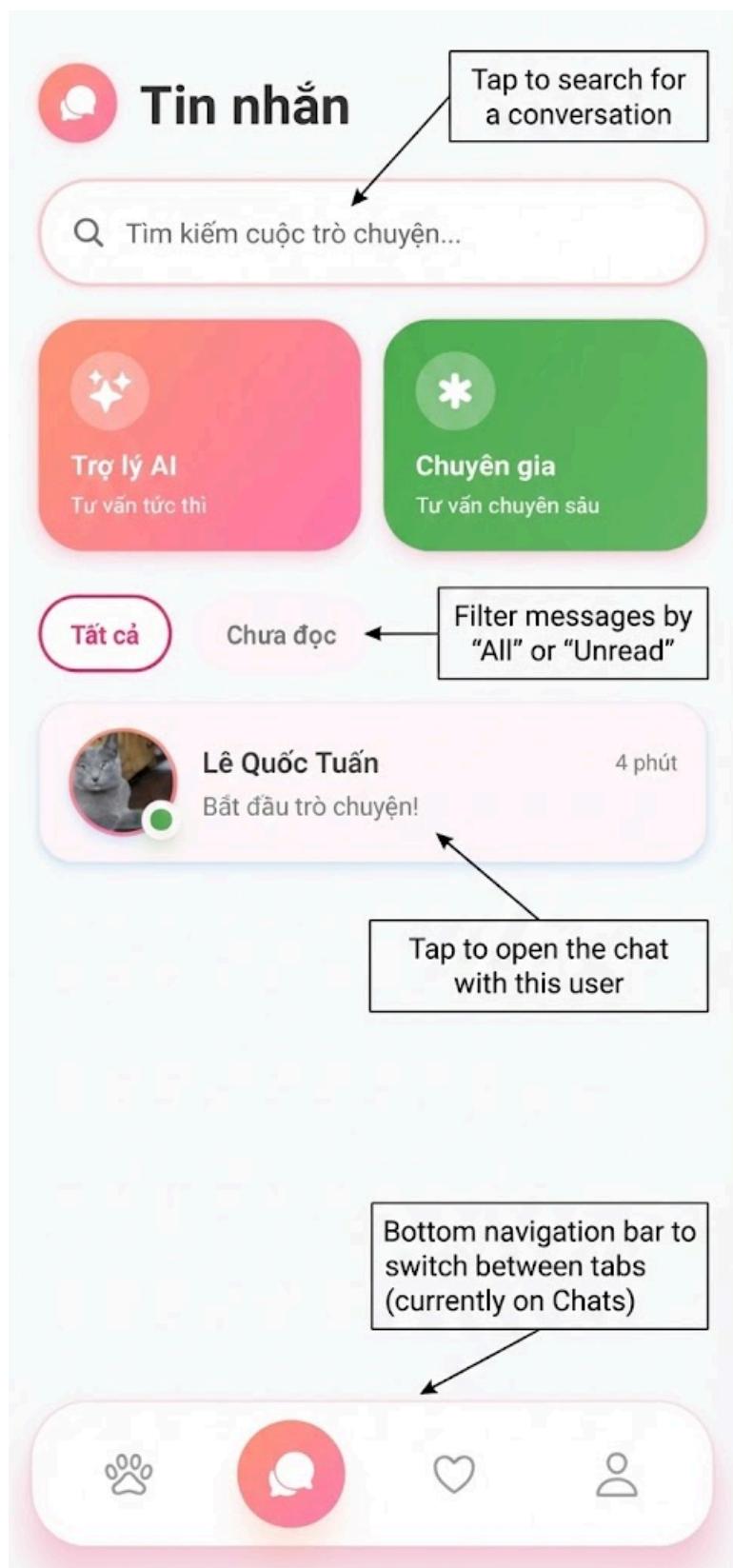


3.2.6.5. Unmatch with User

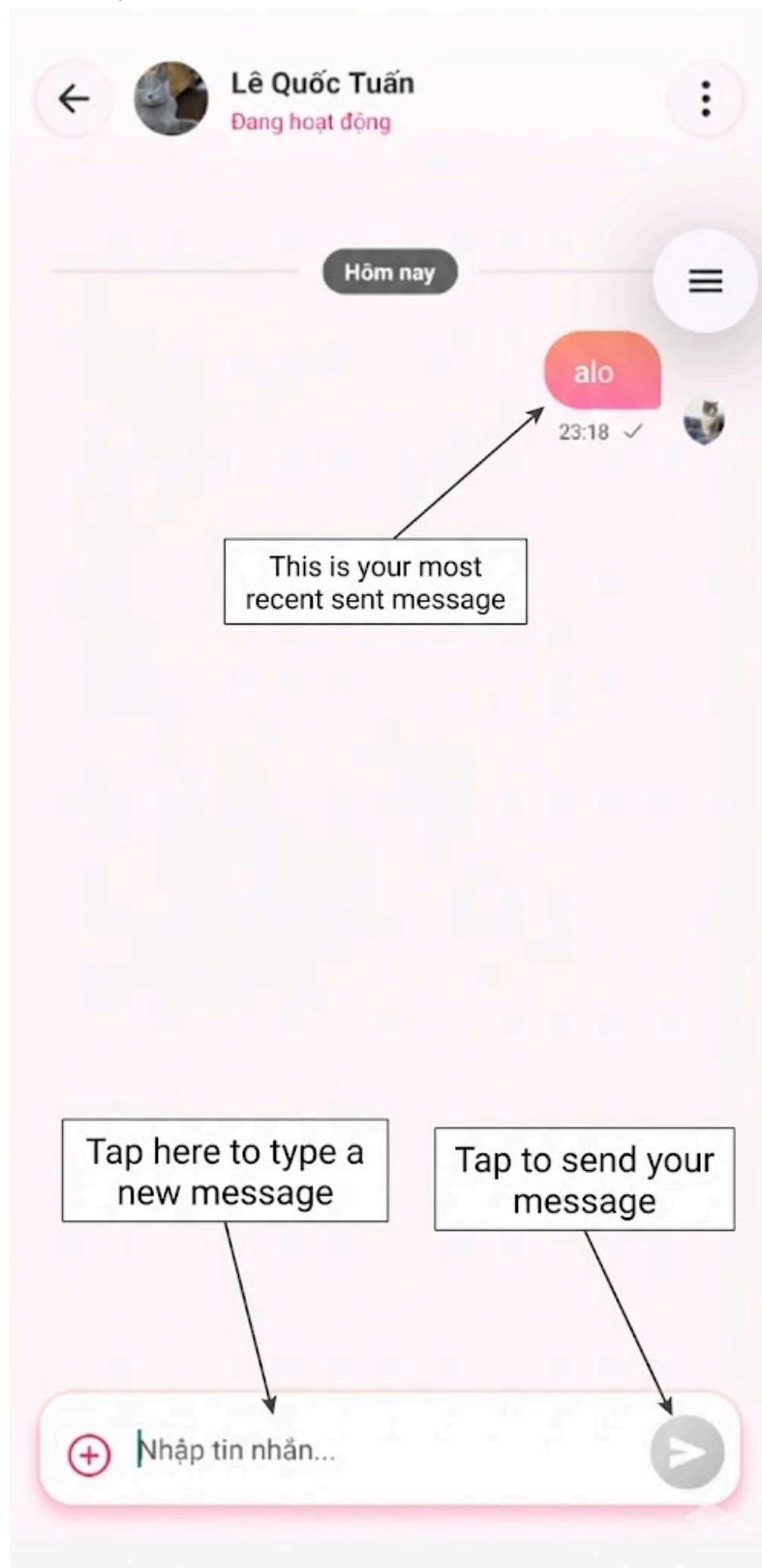


3.2.7. Communication (User Chat)

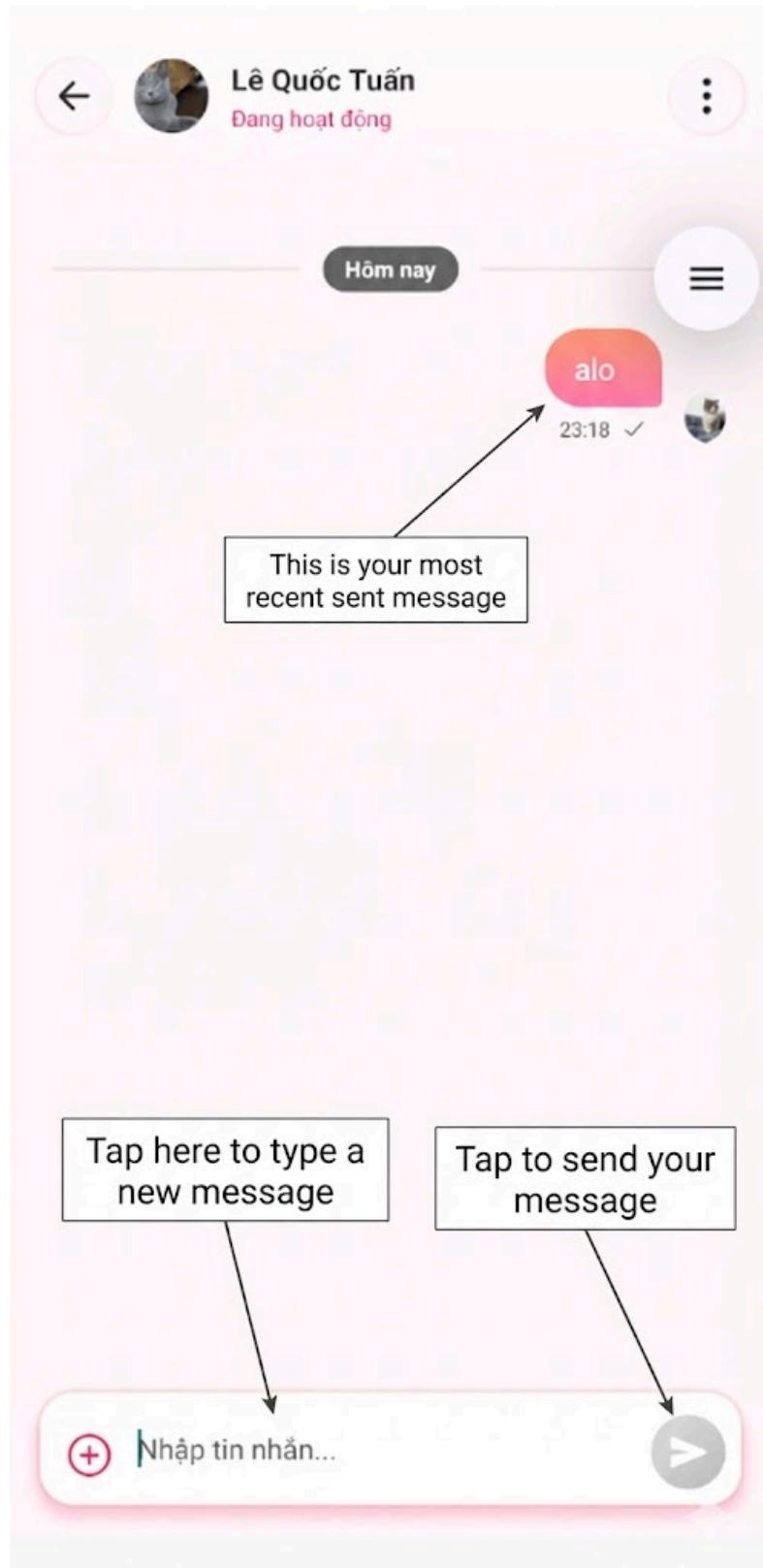
3.2.7.1. View Chat List



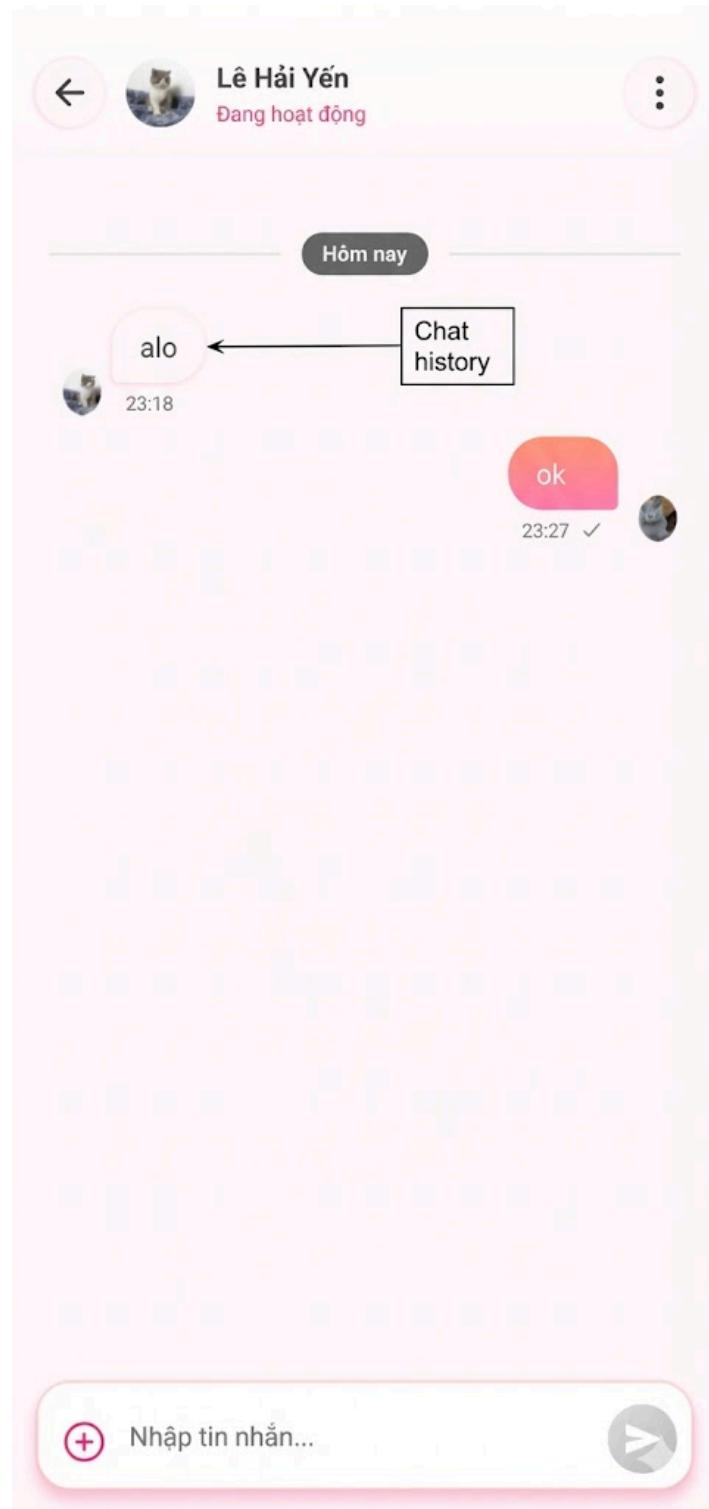
3.2.7.2. Open Chat Conversation



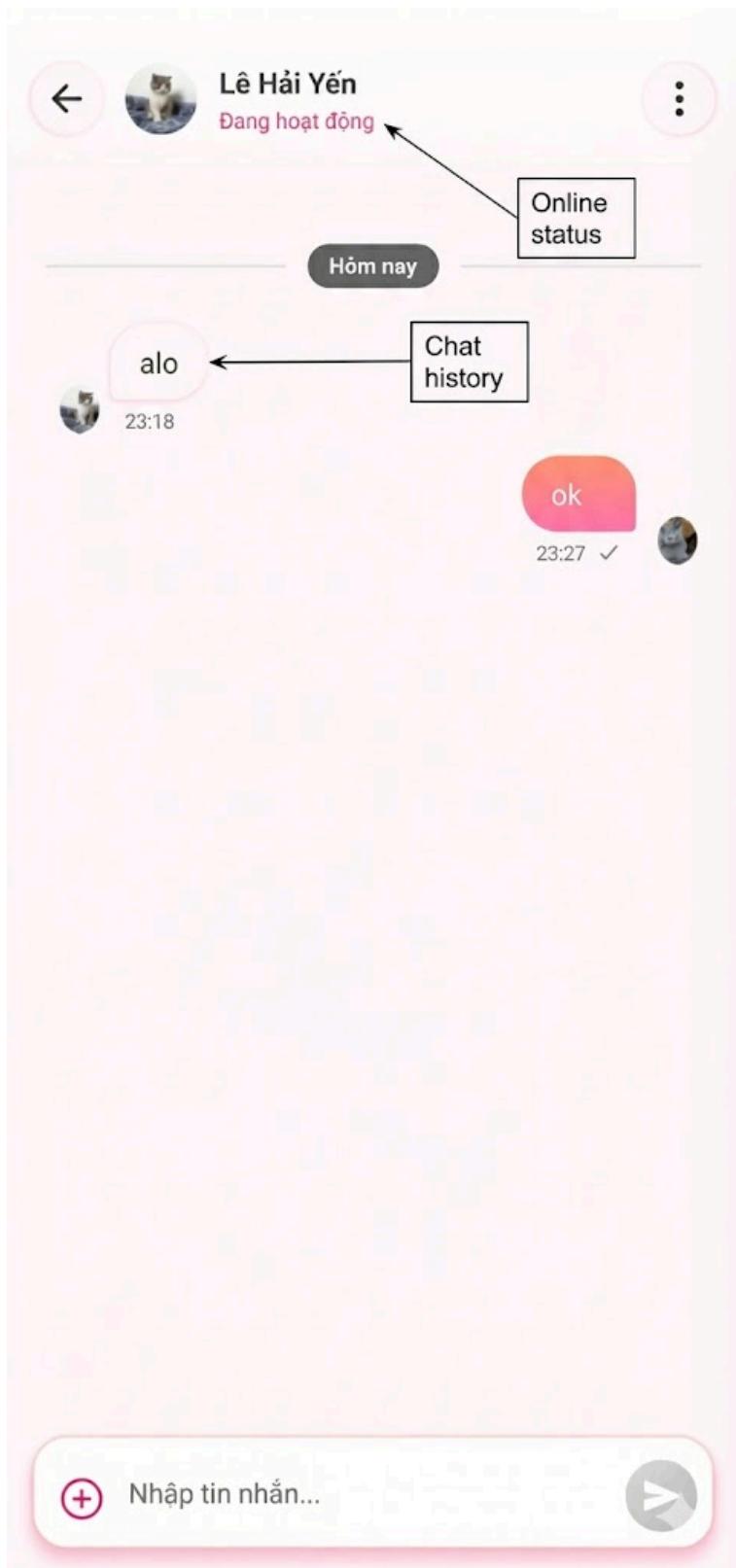
3.2.7.3. Send Text Message



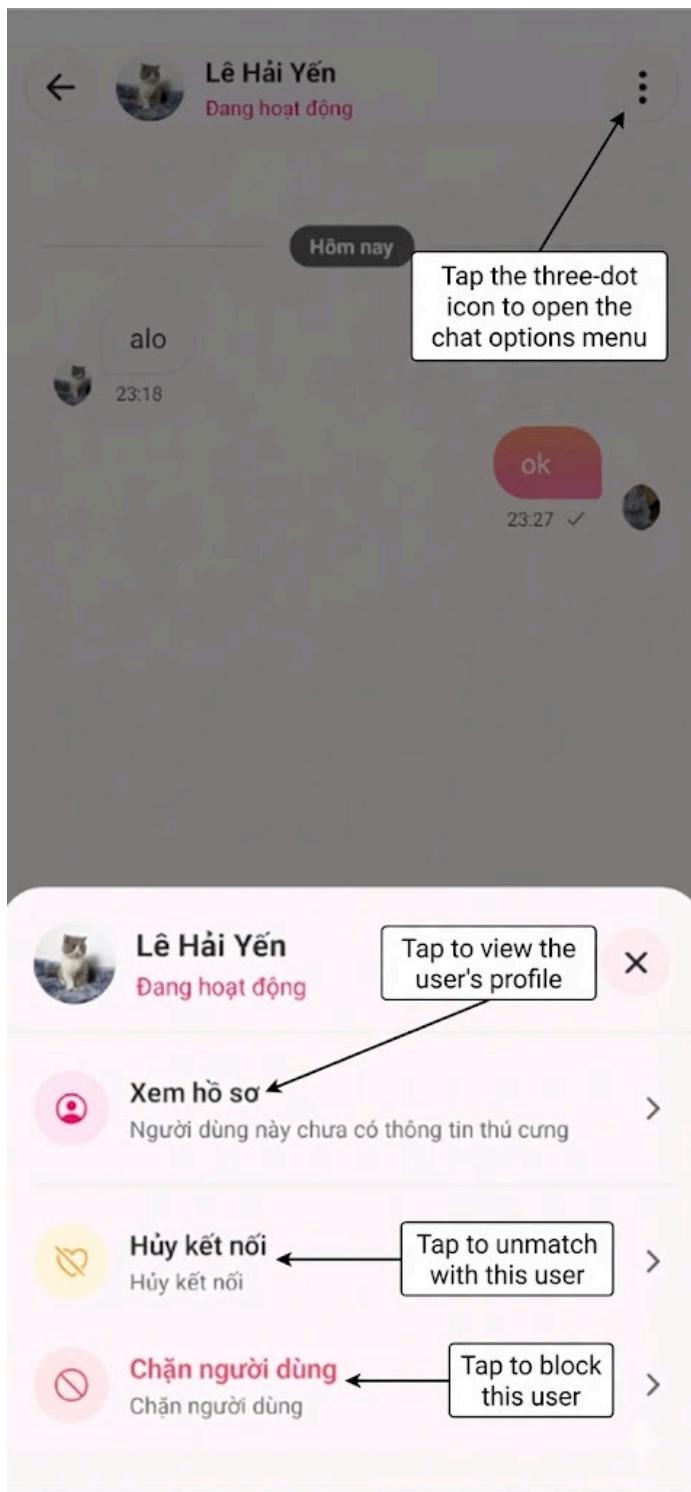
3.2.7.4. View Message History



3.2.7.5. View Online Status



3.2.7.6 View Chat Menu



3.2.7.7. Delete Conversation (Unmatch)



3.2.7.8 View Profie User Chat

The image consists of two side-by-side screenshots from a mobile application.

Screenshot 1 (Left): Chat Screen

- Top bar: "Lê Hải Yến" (User Name), "Dang hoạt động" (Active).
- Message list:
 - "alo" (23:18)
 - "ok" (23:27 ✓)
- Bottom bar: "+" (New message), "Chat hình hiện" (Current chat), and an upward arrow icon.
- Bottom navigation:
 - Xem hồ sơ (View profile) - Tap to view the detailed pet profile of the user you are chatting with.
 - Hủy kết nối (Cancel connection)
 - Chặn người dùng (Block user)

Screenshot 2 (Right): Pet Profile Screen

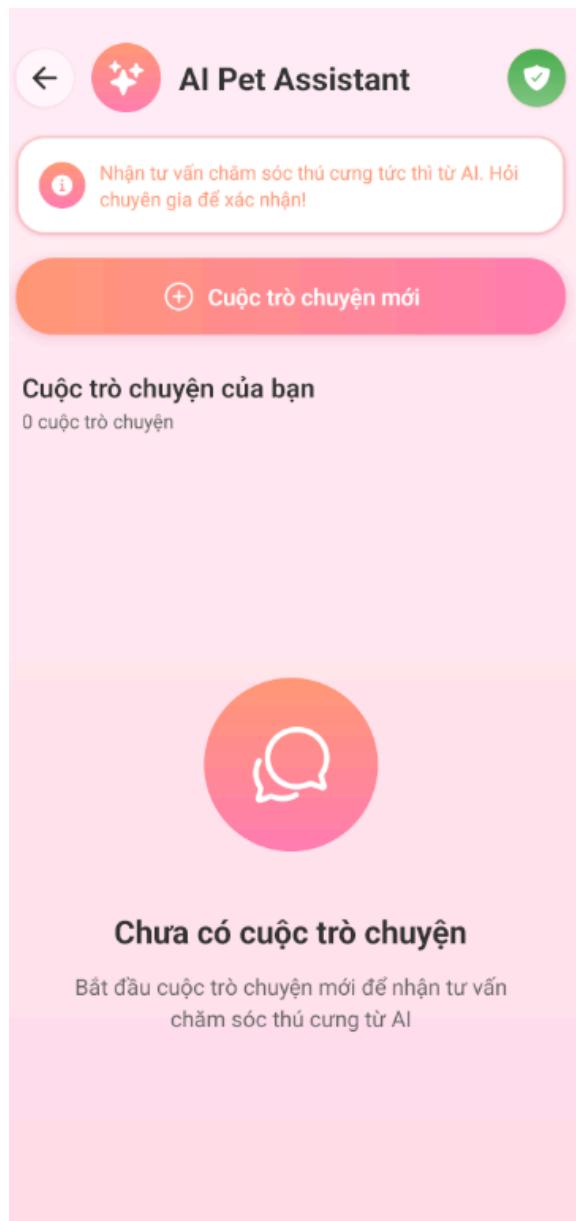
- Top bar: "11:38" and a back arrow icon.
- Image: A fluffy, grey and white cat sitting on a dark blue textured surface.
- Profile details:
 - Bella ♀**
 - Corgi** • 3 tuổi
 - Tỉnh Ninh Bình
- Bottom navigation:
 - Corgi (Giống)
 - 3 tuổi (Tuổi)
 - Female (Giới tính)
- Description: "Về Bella" (About Bella): Energetic and playful.
- Summary: "Đặc điểm" (Features): 15

3.2.7.9 Block User Chat

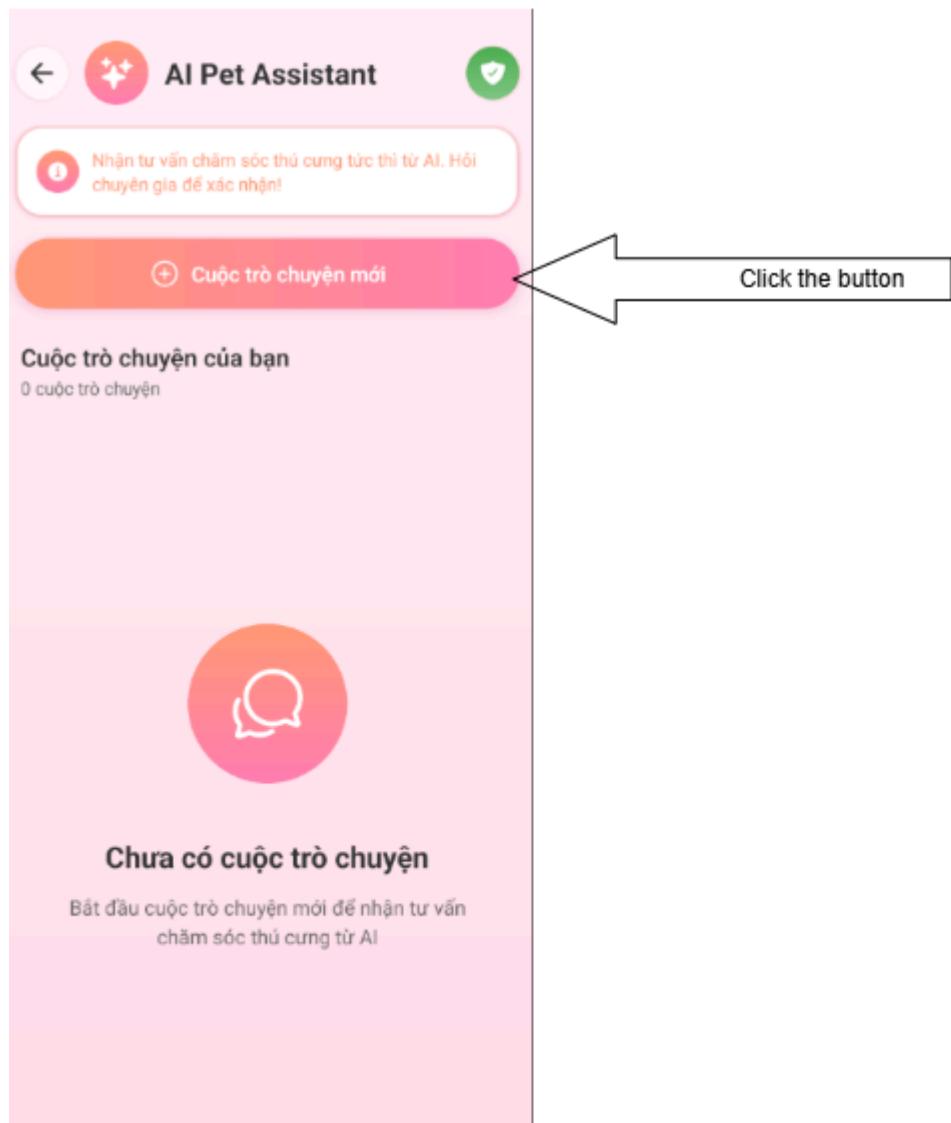


3.2.8. AI Pet Advisor

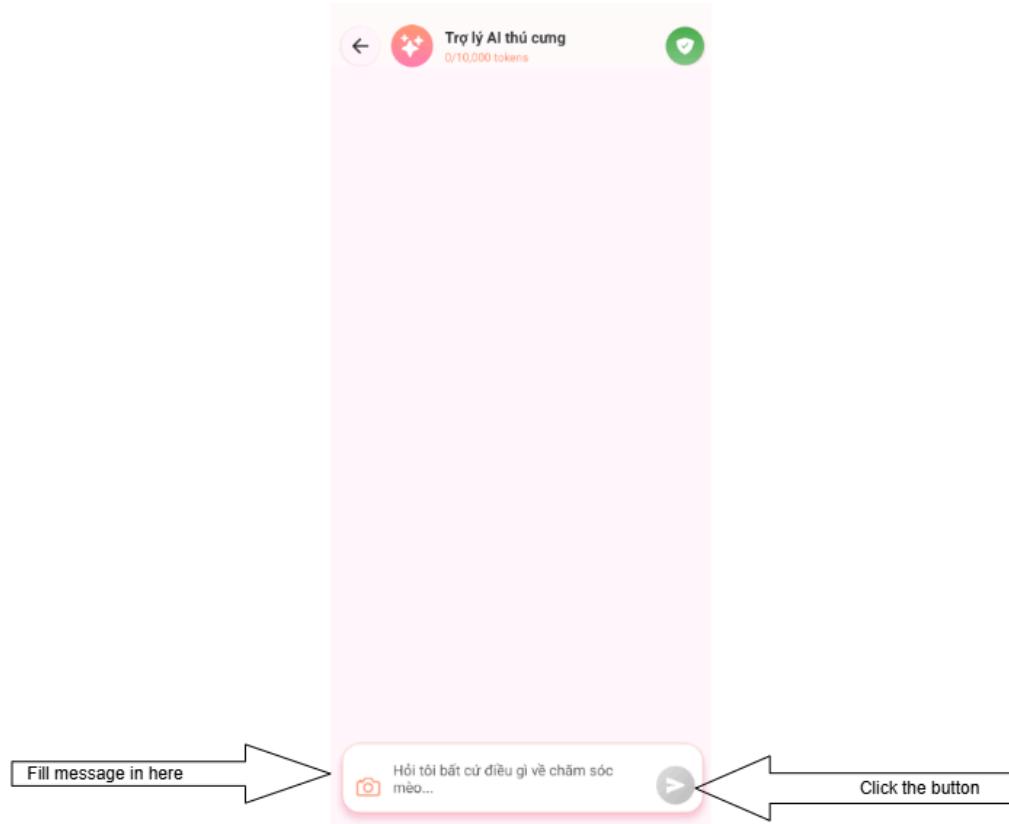
3.2.8.1. View AI Chat List



3.2.8.2. Create New AI Conversation



3.2.8.3. Ask Pet Care Questions

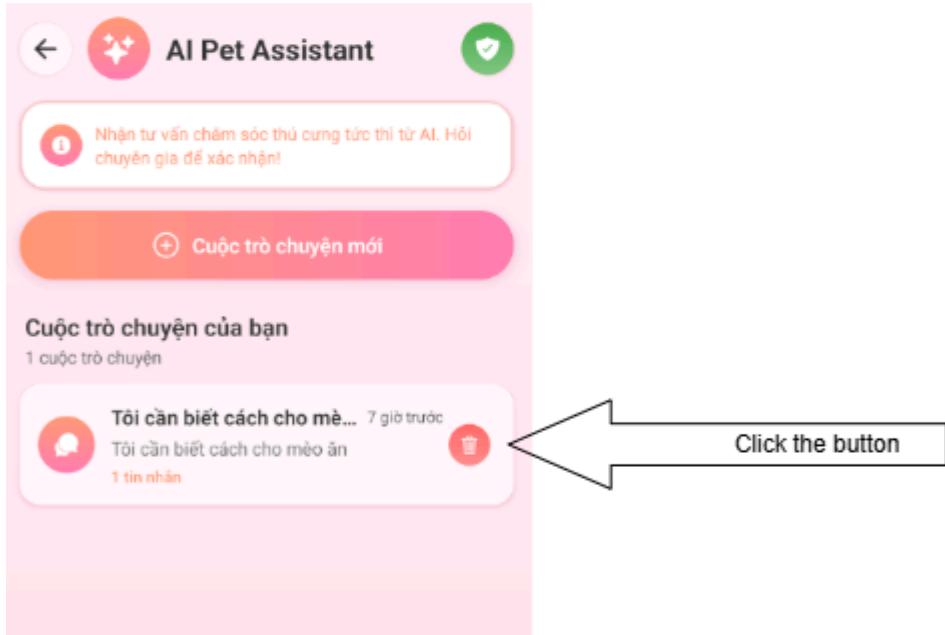


3.2.8.4. View AI Conversation History

The screenshot shows a mobile application interface for an AI pet helper. At the top, there's a back arrow, a profile icon with a starburst, the text "Trợ lý AI thú cưng", and a green circular icon with a checkmark. Below this is a pink header bar with the text "Tôi cần biết cách cho mèo ăn". The main conversation area has a light blue background with rounded corners. On the left, the AI's name "Trợ lý AI" is followed by the timestamp "11:17 PM". The AI's message is: "Chào bạn, mình là Pawnder AI, rất vui được giúp bạn về vấn đề cho mèo ăn! Việc cho mèo ăn đúng cách rất quan trọng để chúng khỏe mạnh và hạnh phúc. Dưới đây là một vài gợi ý hữu ích: * **Chọn loại thức ăn phù hợp:** Có rất nhiều loại thức ăn cho mèo trên thị trường (khô, ướt, thức ăn tươi). Hãy chọn loại phù hợp với độ tuổi, tình trạng sức khỏe và sở thích của mèo. Đọc kỹ thành phần để đảm bảo thức ăn giàu protein và ít carb bạn nhé! * **Định lượng thức ăn:** Đọc hướng dẫn trên bao bì thức ăn và điều chỉnh theo cân nặng, mức độ hoạt động của mèo. Tránh cho mèo ăn quá nhiều gây béo phì. * **Cho ăn đúng giờ:** Cho mèo ăn vào một giờ cố định mỗi ngày sẽ giúp chúng hình thành thói quen tốt và tiêu hóa tốt hơn. * **Nước sạch luôn có sẵn:** Mèo cần nước sạch để giữ cơ thể đủ nước. Hãy đảm bảo mèo luôn có nước tươi trong bát.

At the bottom, there's a pink input field with a camera icon and the placeholder text "Hỏi tôi bất cứ điều gì về chăm sóc mèo...". To its right is a grey circular button with a white right-pointing arrow.

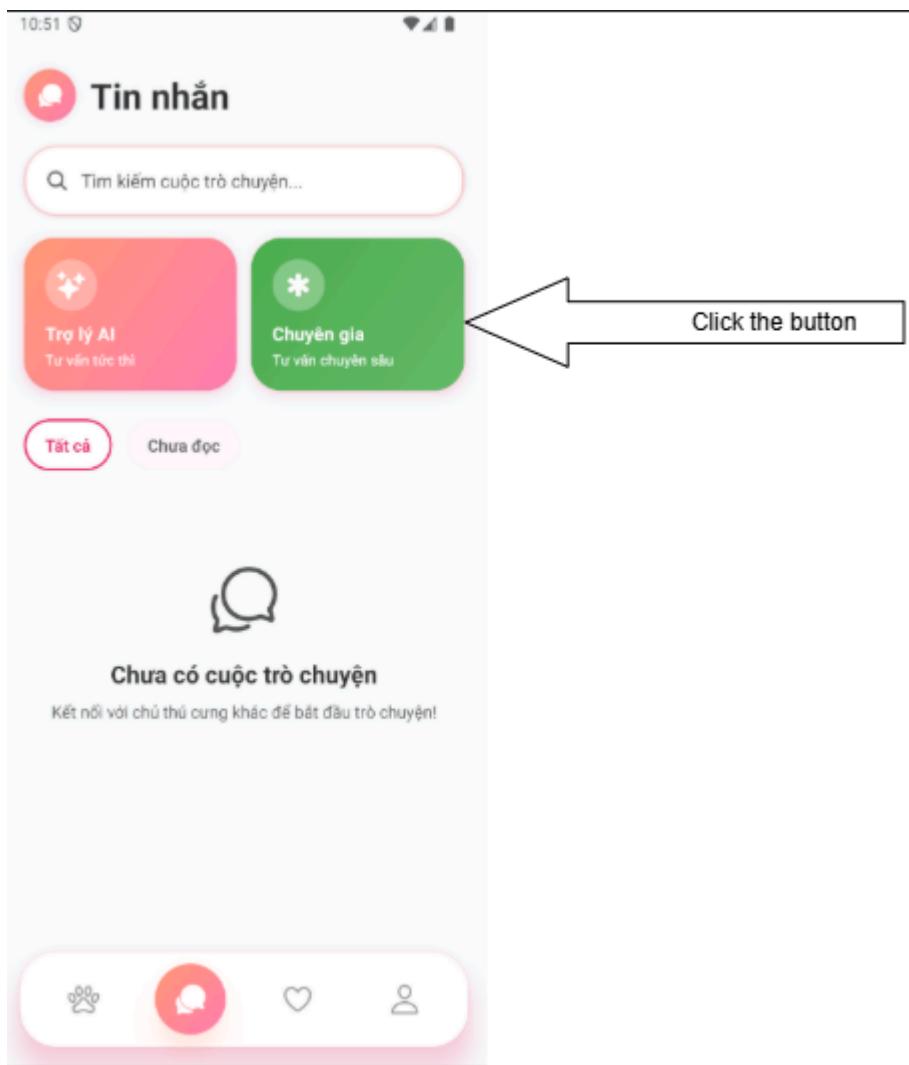
3.2.8.5. Delete AI Conversation



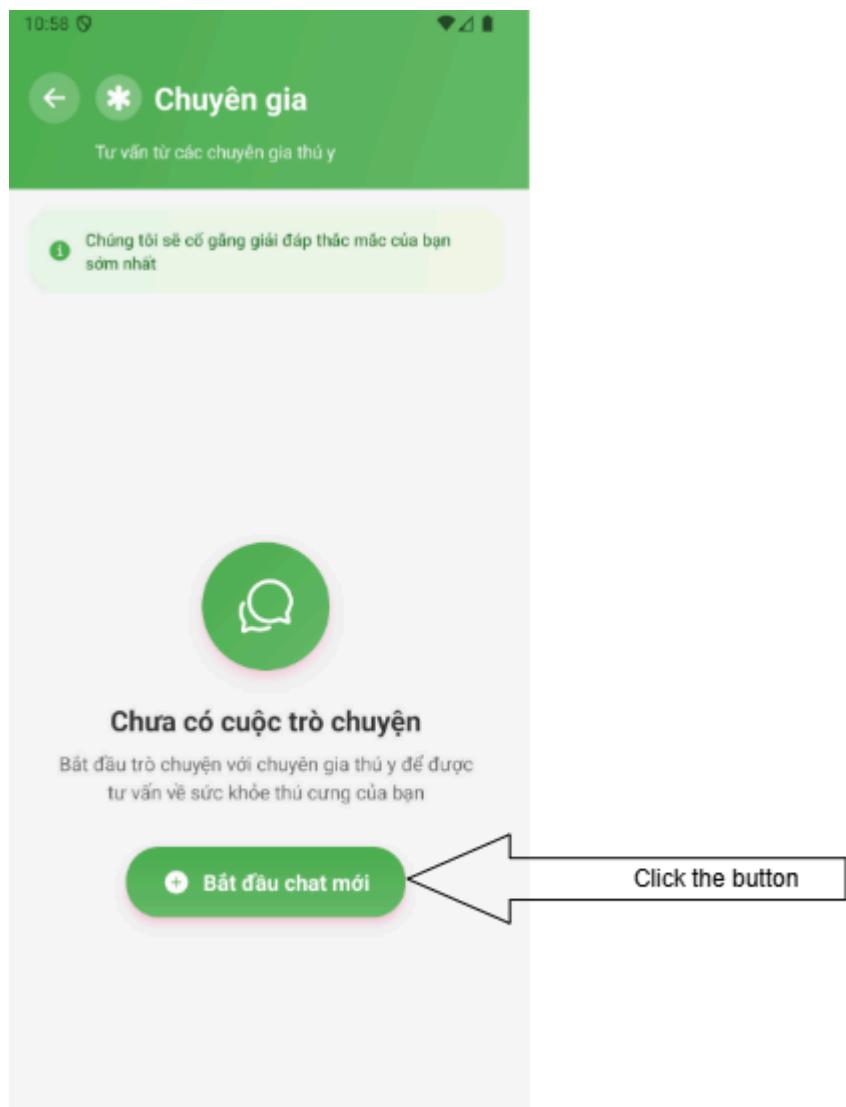
3.2.9. Expert Consultation

3.2.9.1. View Expert Chat List

Step 1:



Step 2:



Step 3:

10:59 ⓘ

← * Chuyên gia

Tư vấn từ các chuyên gia thú y

i Chúng tôi sẽ cố gắng giải đáp thắc mắc của bạn sớm nhất

Chọn chuyên gia ×

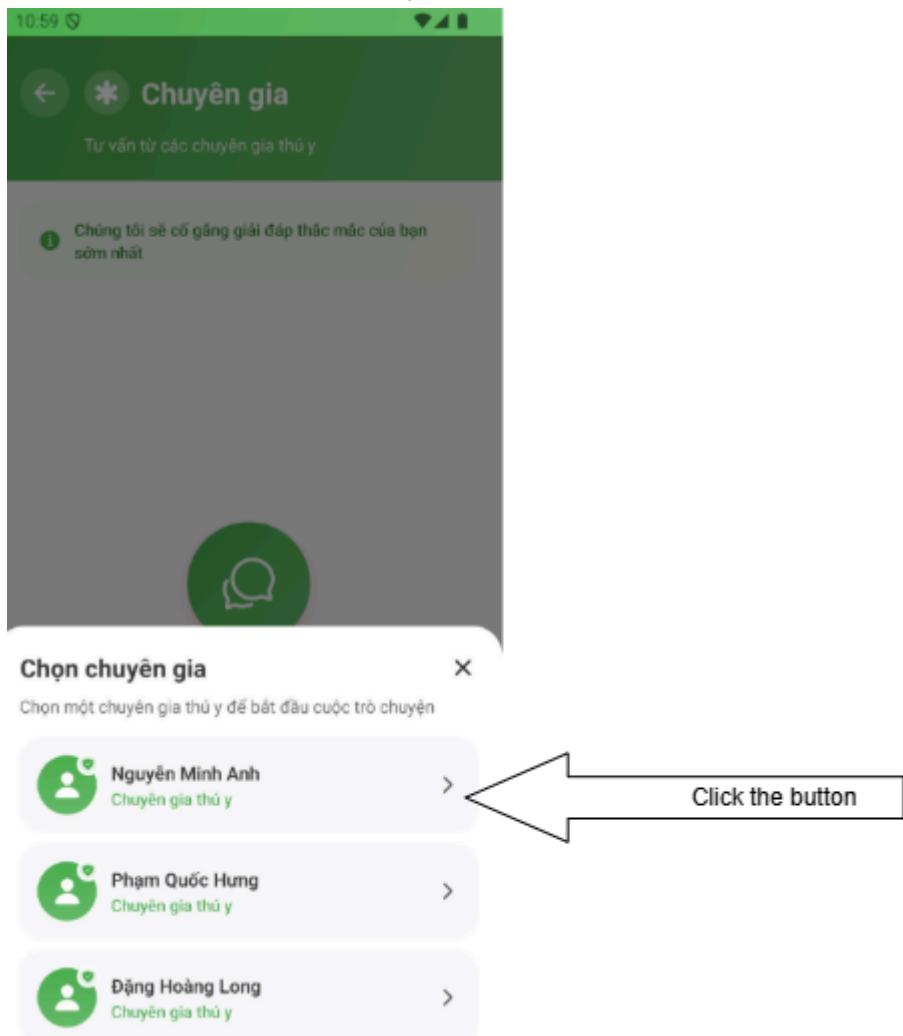
Chọn một chuyên gia thú y để bắt đầu cuộc trò chuyện

 Nguyễn Minh Anh
Chuyên gia thú y >

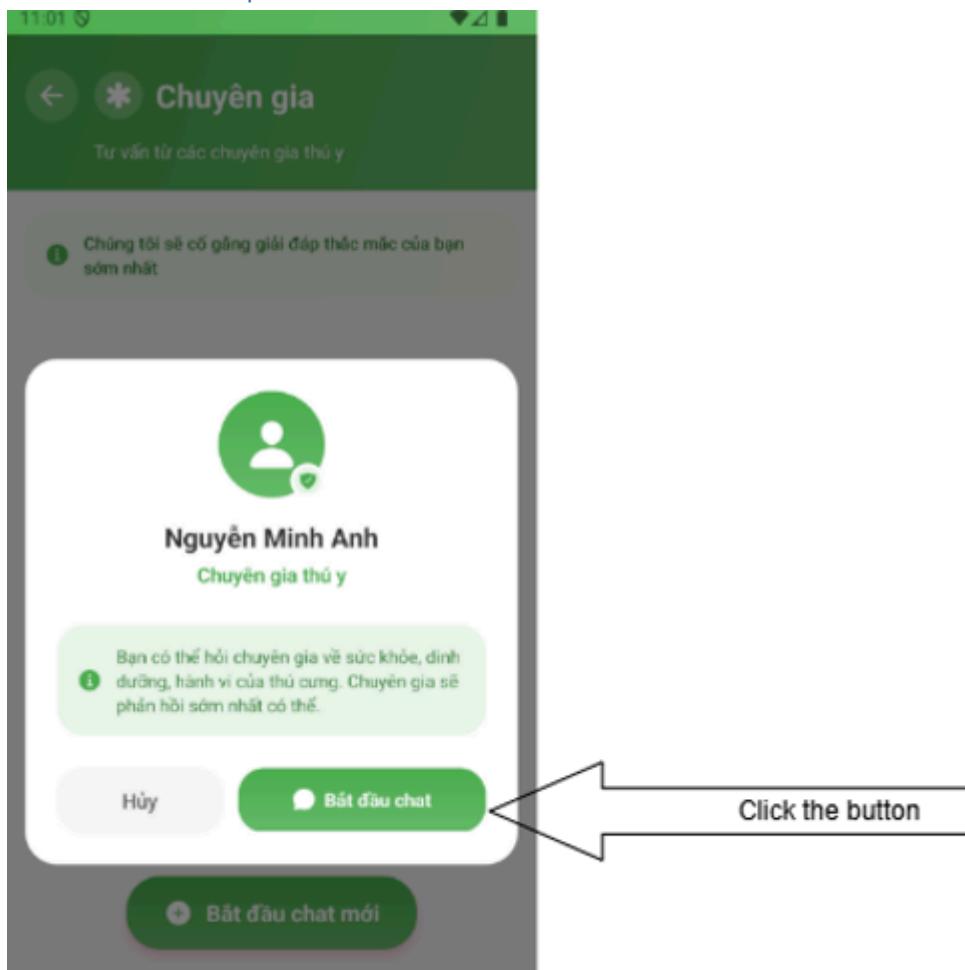
 Phạm Quốc Hưng
Chuyên gia thú y >

 Đặng Hoàng Long
Chuyên gia thú y >

3.2.9.2. Start Consultation with Expert

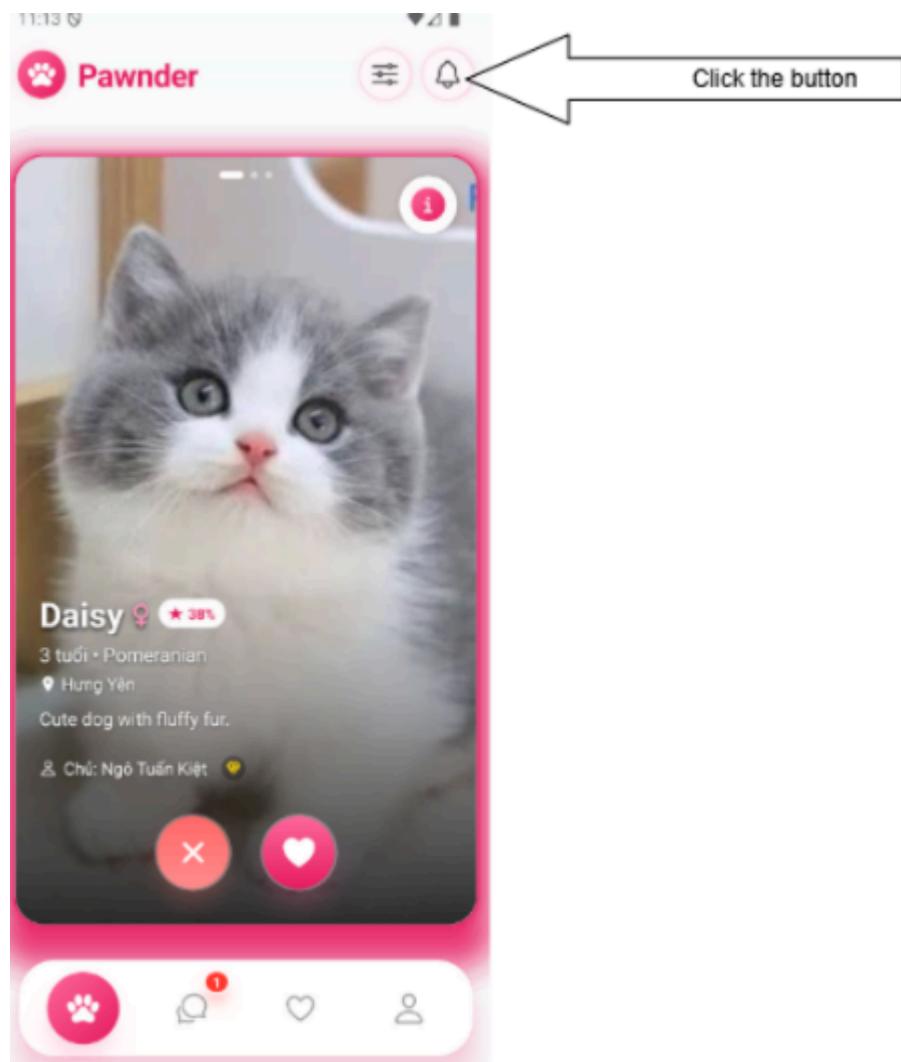


3.2.9.3. Chat with Expert

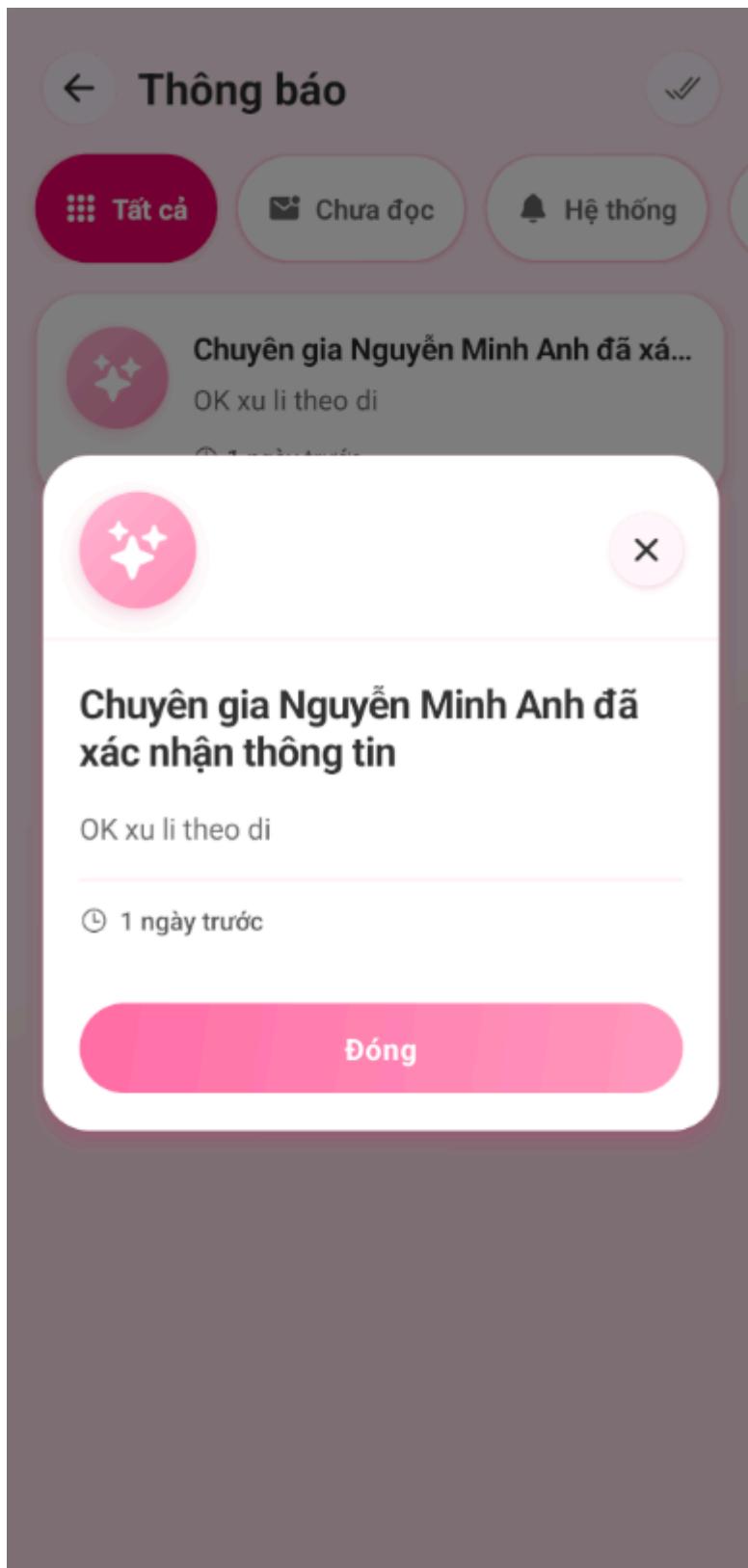


3.2.10. Notifications

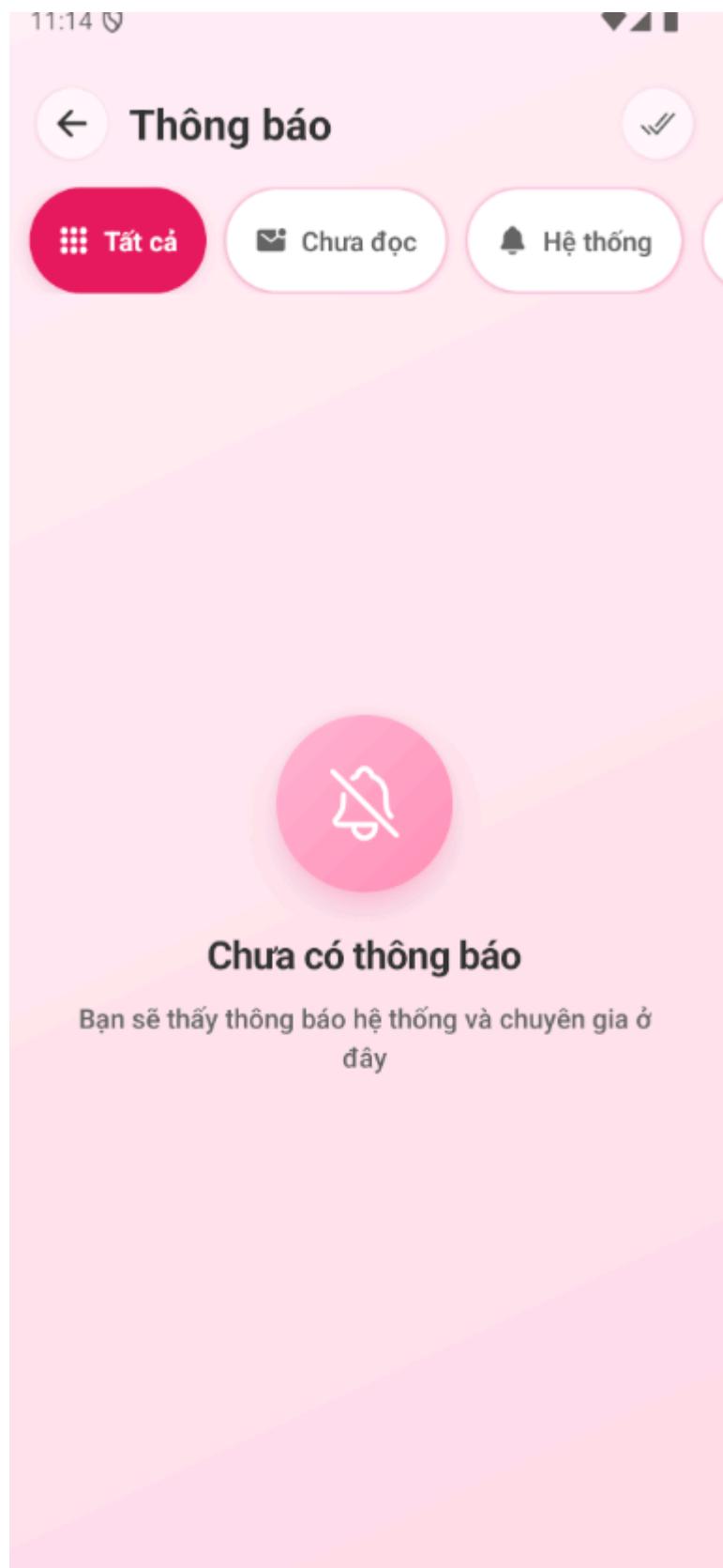
3.2.10.1. View Notification List



3.2.10.2. Read Notification Details



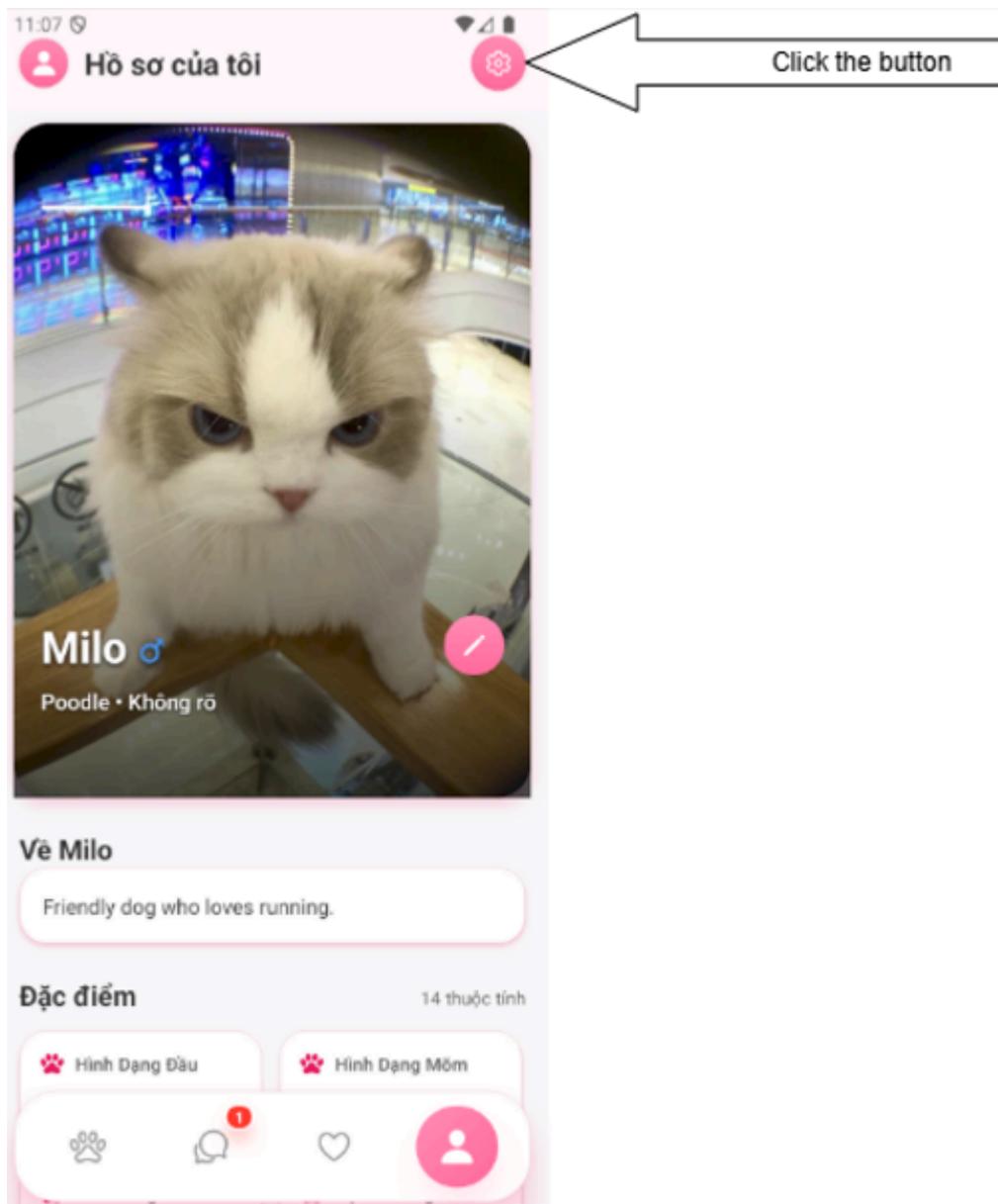
3.2.10.3. Mark as Read



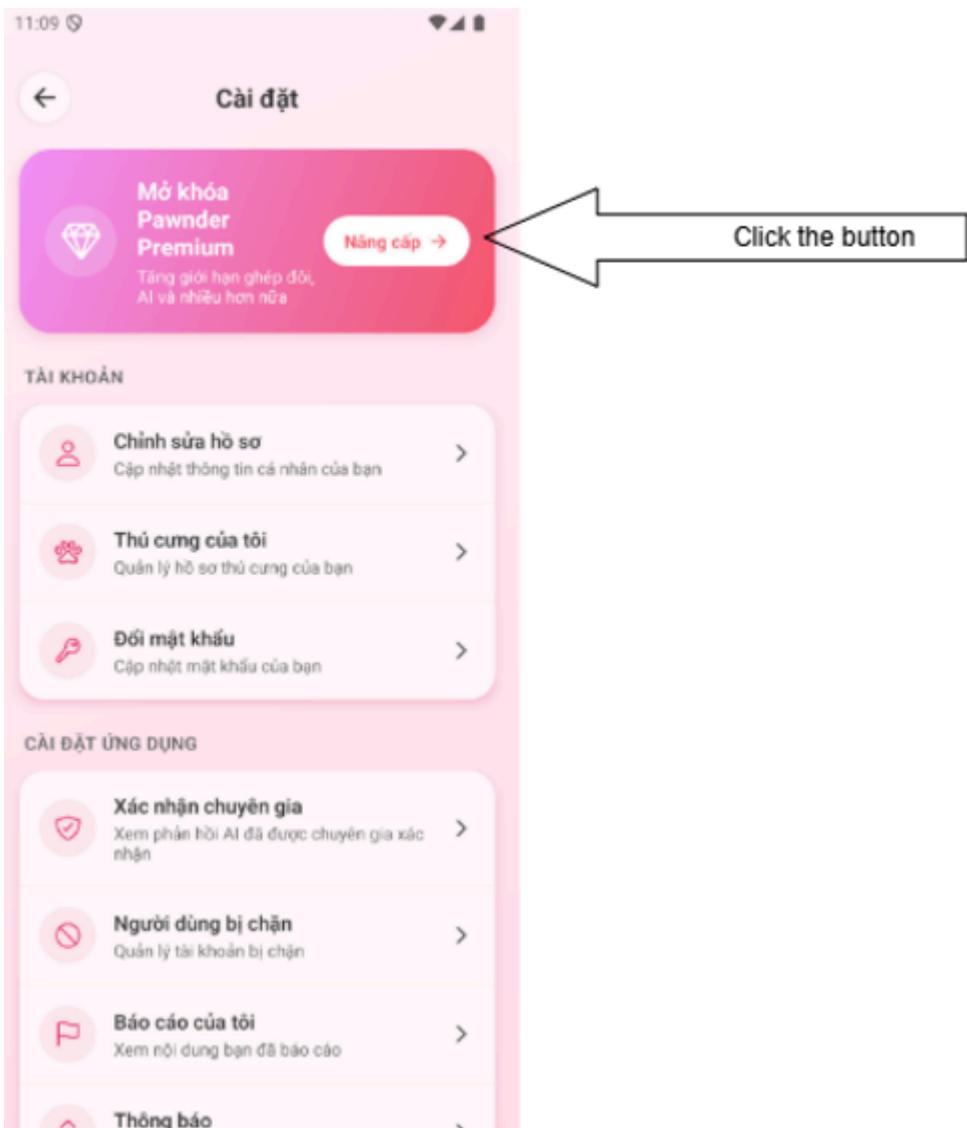
3.2.11. Premium Membership

3.2.11.1. View Premium Plans

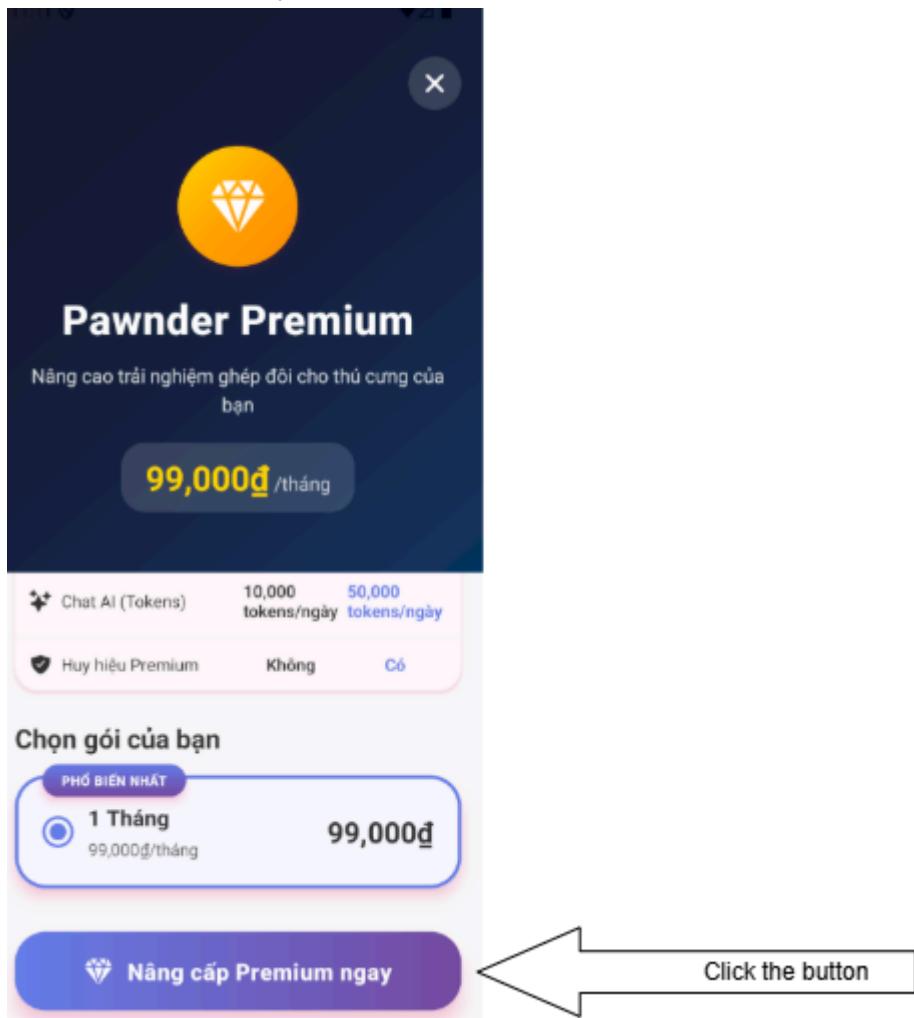
Step 1:



Step 2:



3.2.11.2. Select Subscription Plan



3.2.11.3. Complete Payment via QR Code

11:15 9

Thanh toán

Gói dịch vụ Pawnder Premium - 1 Tháng

Thời hạn 1 Tháng

Số tiền **99.000đ**

Quét mã QR để thanh toán

Mở ứng dụng ngân hàng và quét mã QR bên dưới

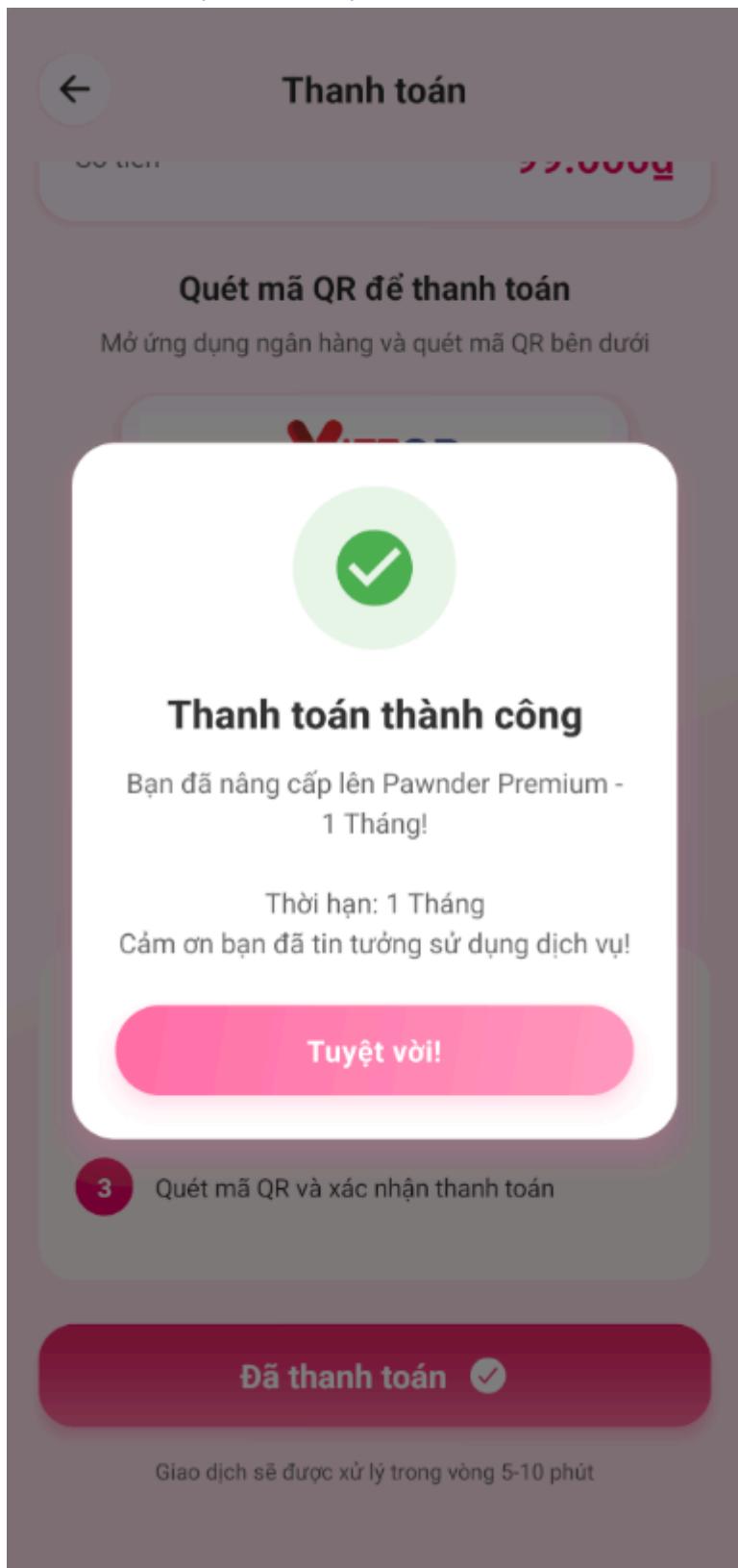
VIETQR



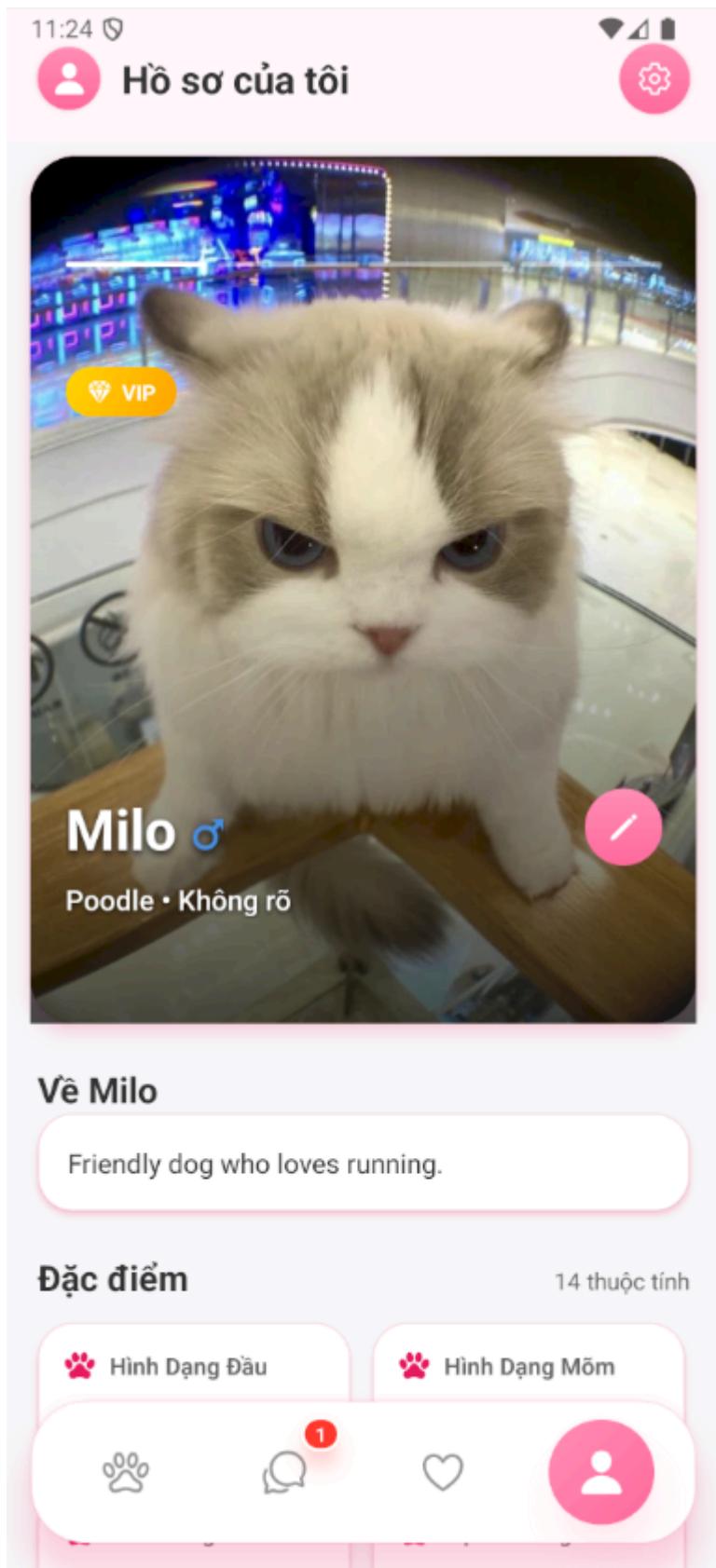
napas 247 | MB

- 1 Mở ứng dụng ngân hàng của bạn
- 2 Chọn chức năng quét mã QR
- 3 Quét mã QR và xác nhận thanh toán

3.2.11.4. View Payment History

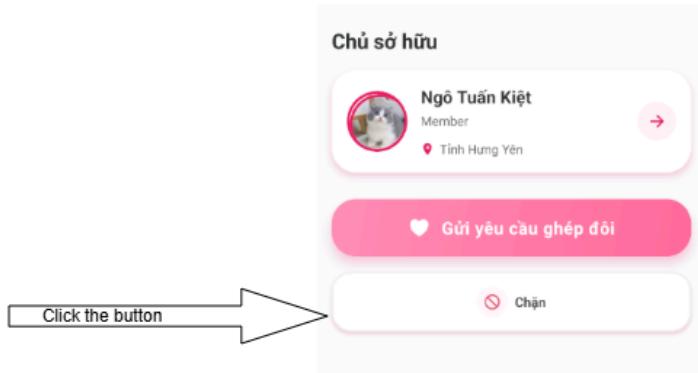


3.2.11.5. Check VIP Status



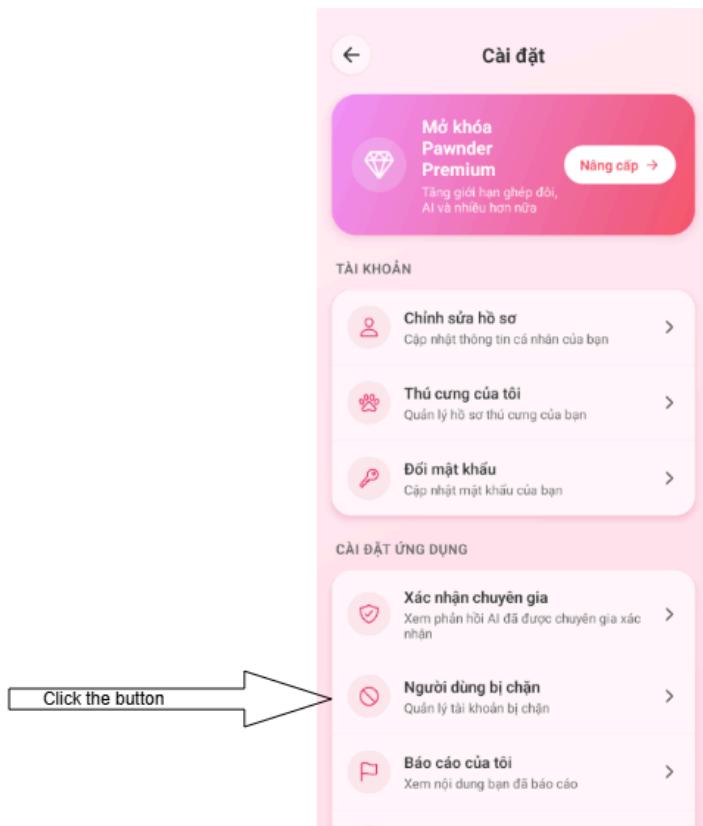
3.2.12. Safety & Privacy

3.2.12.1. Block User

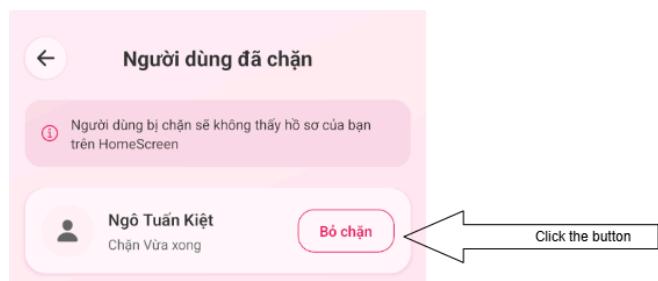


3.2.12.2. Unblock User

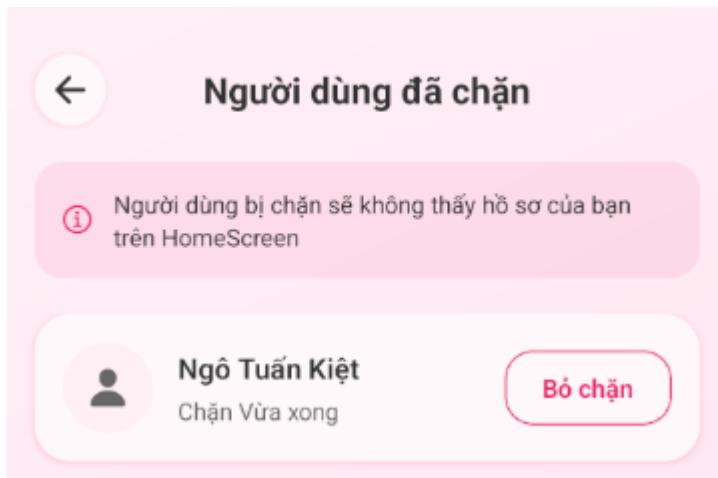
Step 1 :



Step 2 :

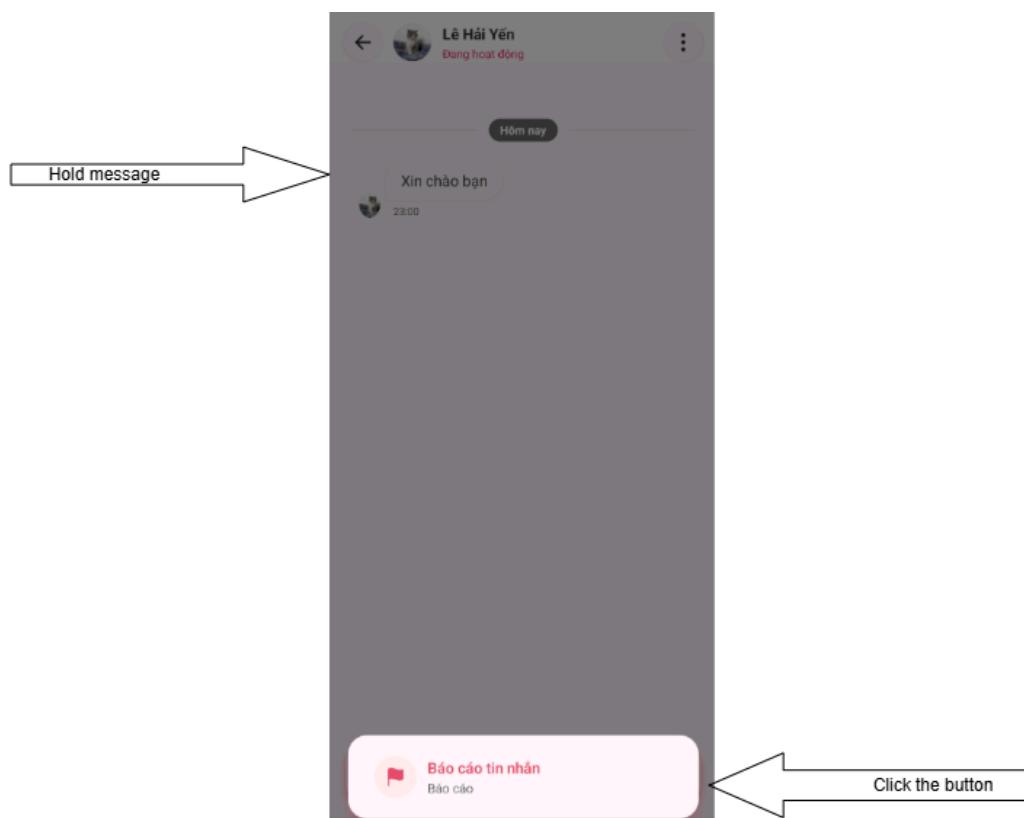


3.2.12.3. View Blocked Users List

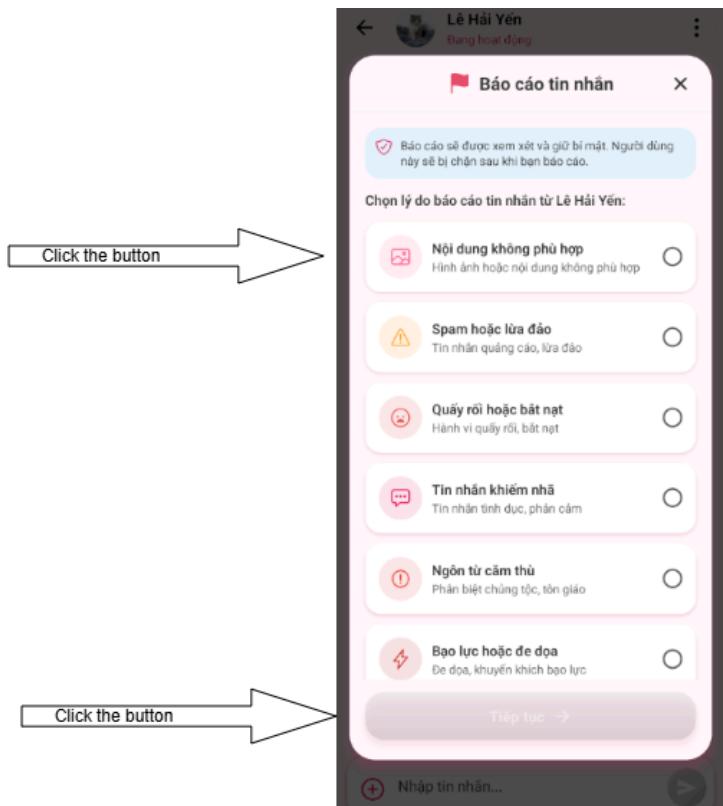


3.2.12.4. Report User/Content

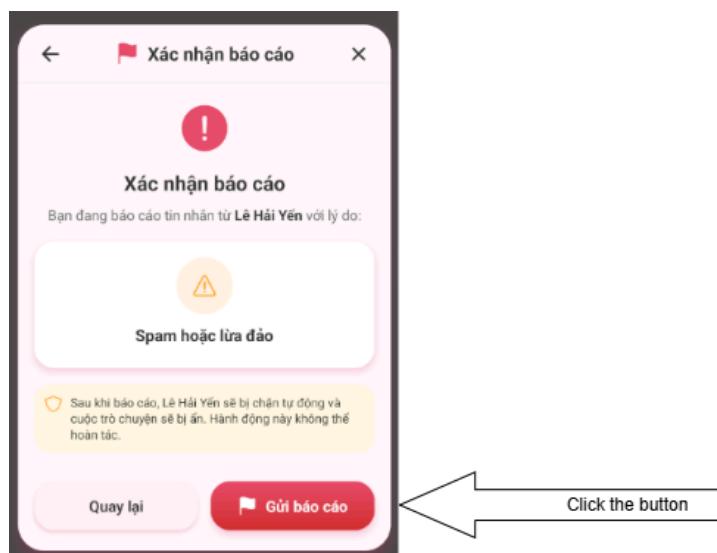
Step 1:



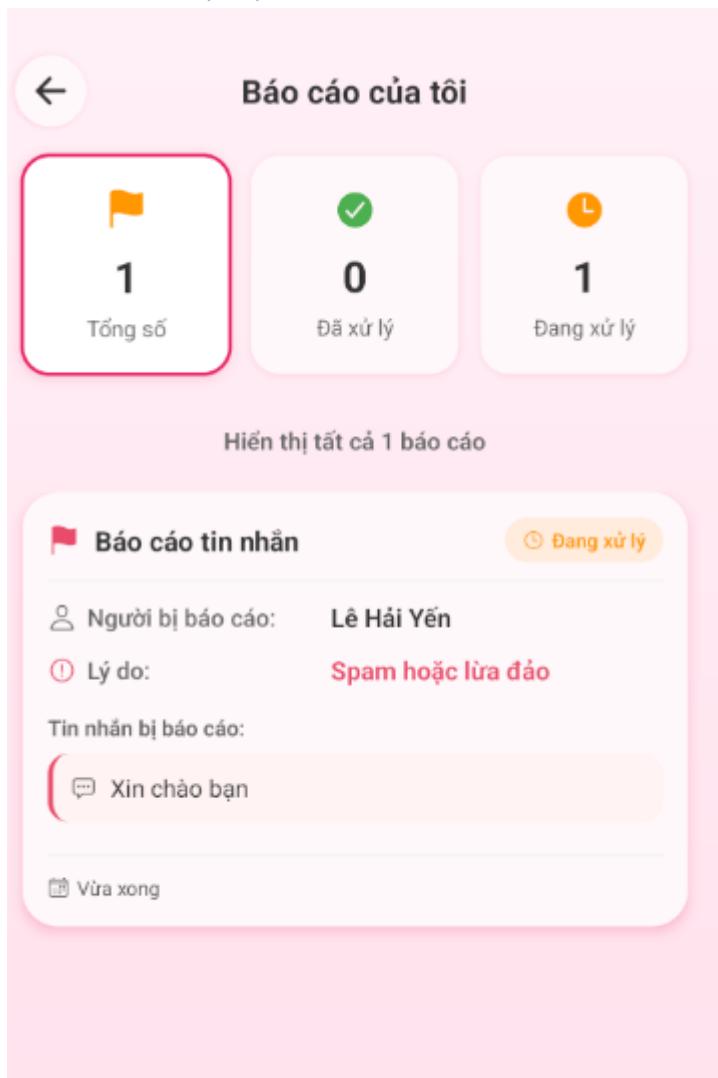
Step 2:



Step 3 :



3.2.12.5. View My Reports



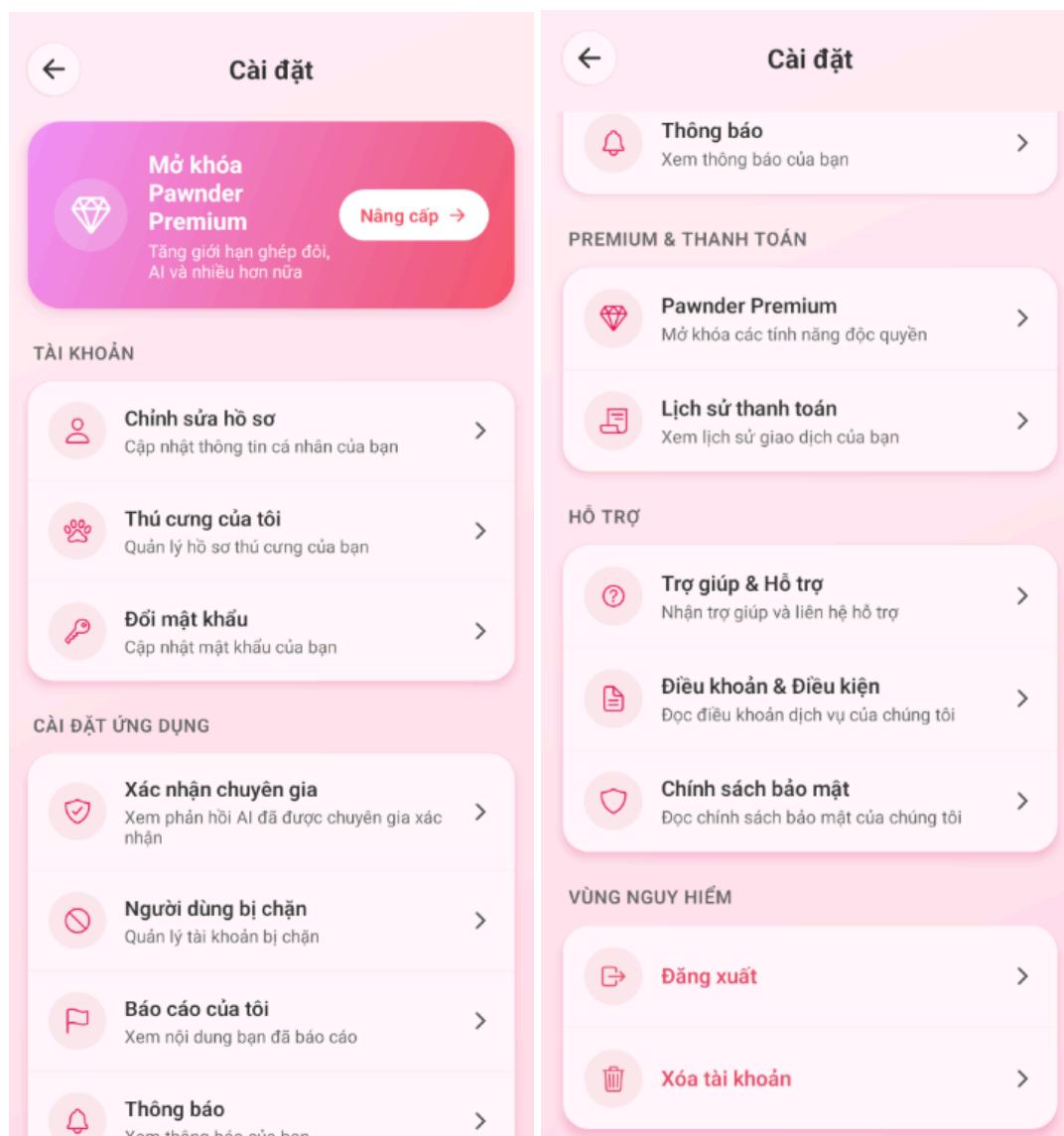
3.2.13. Settings & Support

3.2.13.1. View Settings Menu

Step 1:

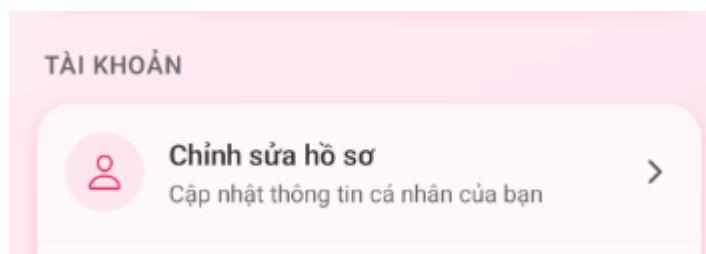


Step 2 :

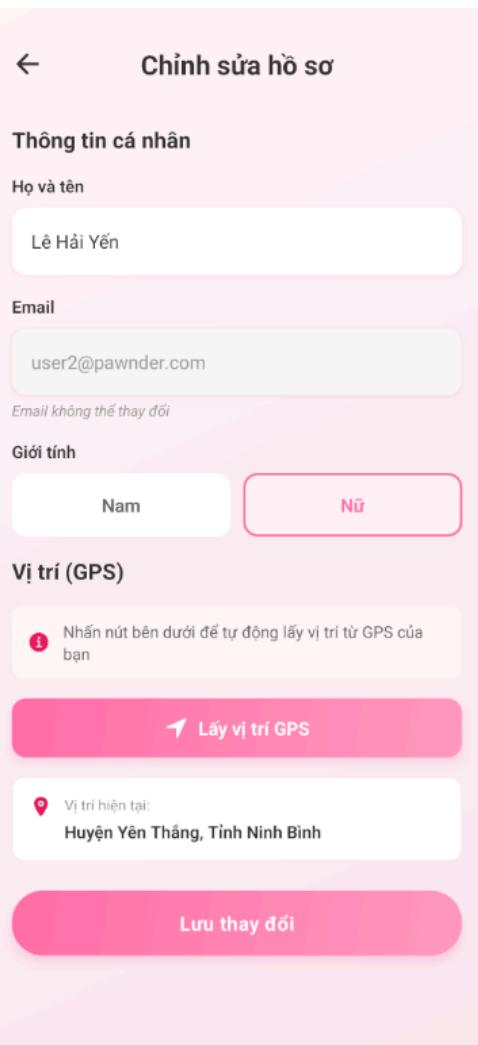


3.2.13.2. Edit Profile (from Settings)

Step 1 :



Step 2 :

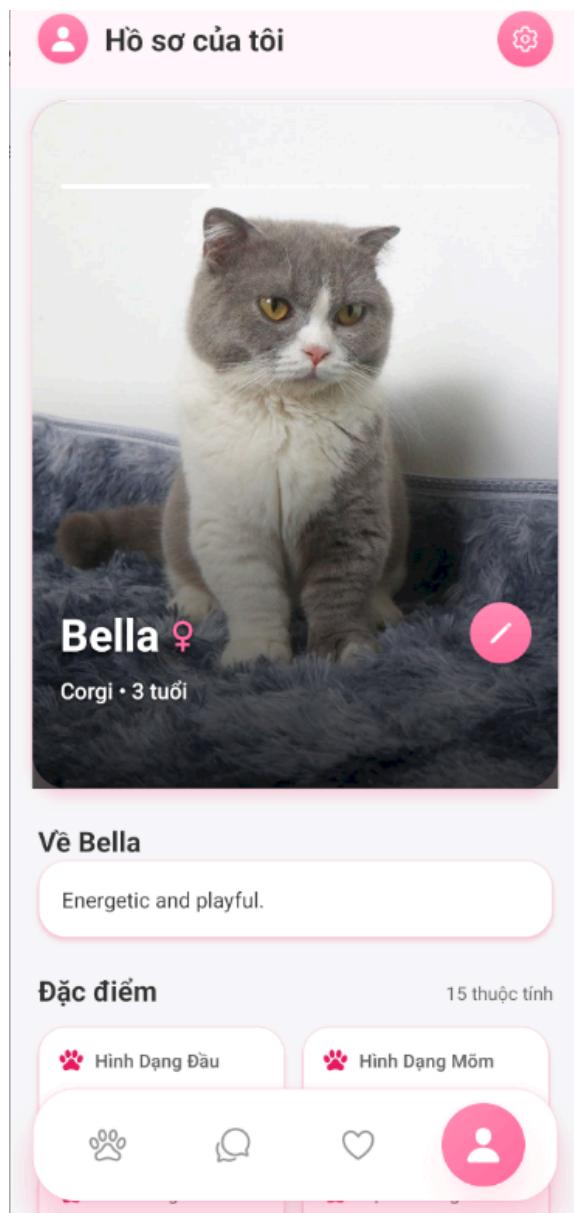


3.2.13.3. View My Pets (from Settings)

Step 1 :



Step 2 :

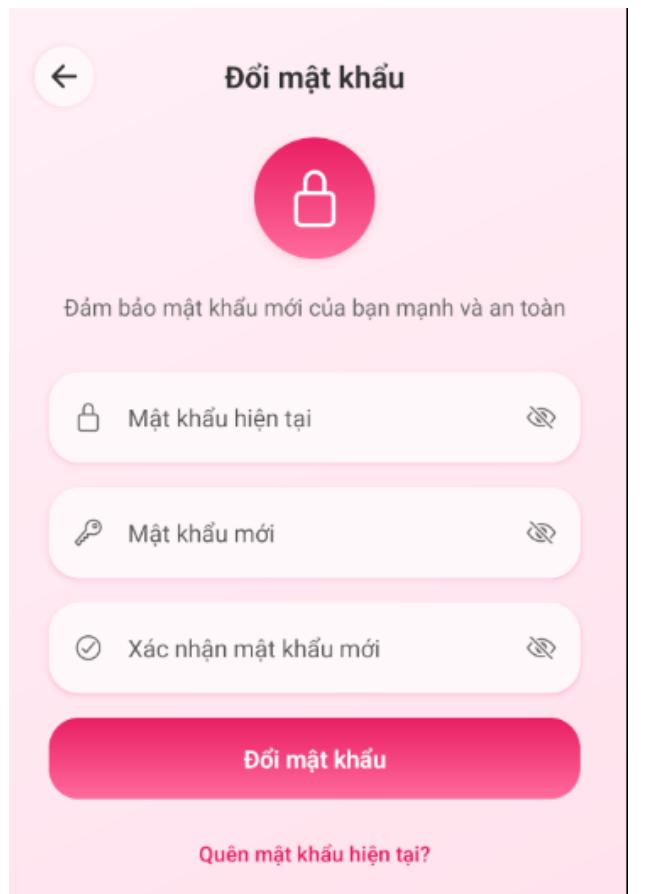


3.2.13.4. Change Password (from Settings)

Step 1 :



Step 2 :

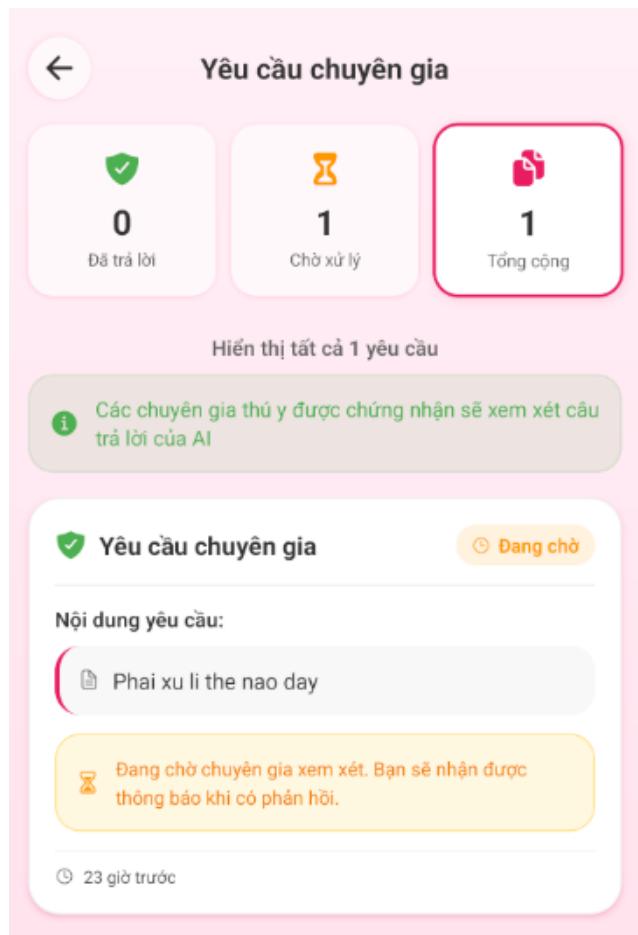


3.2.13.5. Expert Confirmation Application

Step 1 :



Step 2 :



3.2.13.6. View Blocked Users (from Settings)

Step 1 :



Step 2 :



Người dùng đã chặn

Người dùng bị chặn sẽ không thấy hồ sơ của bạn trên HomeScreen



Không có người dùng bị chặn

Bạn chưa chặn ai. Danh sách người dùng bị chặn sẽ hiển thị ở đây.

3.2.13.7. View My Reports (from Settings)

Step 1 :



Báo cáo của tôi

Xem nội dung bạn đã báo cáo



Step 2 :

Báo cáo của tôi

1	0	1
Tổng số	Đã xử lý	Đang xử lý

Hiển thị tất cả 1 báo cáo

Báo cáo tin nhắn ⌚ Đang xử lý

👤 Người bị báo cáo: **Trần Thu Hà**

⌚ Lý do: **Nội dung không phù hợp**

Tin nhắn bị báo cáo:

(💬 Alo

⌚ 22 giờ trước

3.2.13.8. Help & Support

Step 1 :

Trợ giúp & Hỗ trợ >

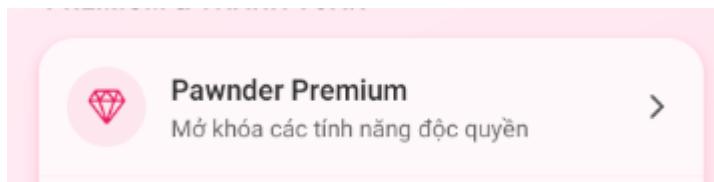
Nhận trợ giúp và liên hệ hỗ trợ

Step 2 :

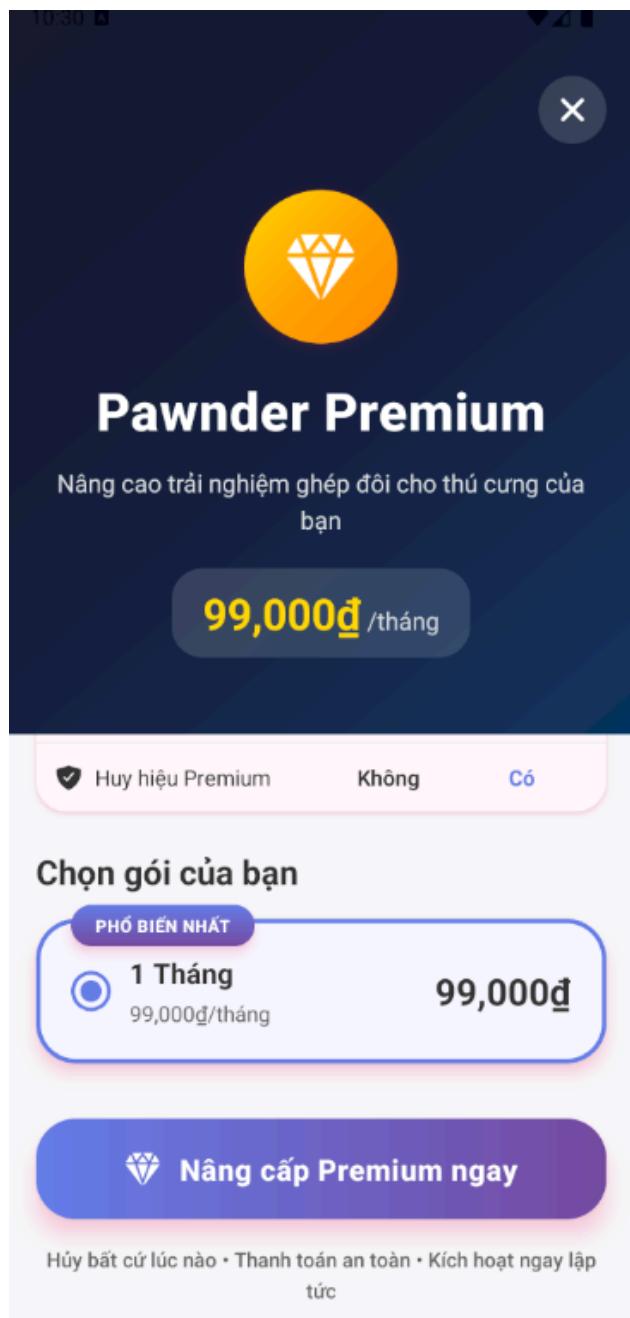
The screenshot shows the 'Trợ giúp & Hỗ trợ' (Help & Support) page. At the top, there is a back arrow icon and the title 'Trợ giúp & Hỗ trợ'. Below the title, there is a section titled 'Contact Us' with three buttons: 'Email' (pink circle with envelope icon), 'Call' (purple circle with phone icon), and 'Live Chat' (orange circle with speaker icon). Underneath this, there is a section titled 'Frequently Asked Questions' containing five collapsed FAQ items, each with a question mark icon and a dropdown arrow: 'How do I create a cat profile?', 'What does 'Match' mean?', 'How do I report inappropriate behavior?', 'Can I change my cat's information?', and 'What is Premium membership?'. At the bottom, there is a section titled 'Send us a message' with a text input field placeholder 'Describe your issue or question...' and a pink button labeled 'Send Message ➤'.

3.2.13.10. Premium Membership (from Settings)

Step 1 :

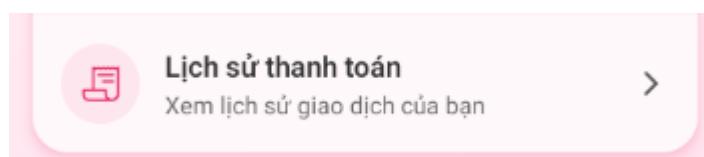


Step 2 :



3.2.13.11. Payment History (from Settings)

Step 1 :



Step 2 :

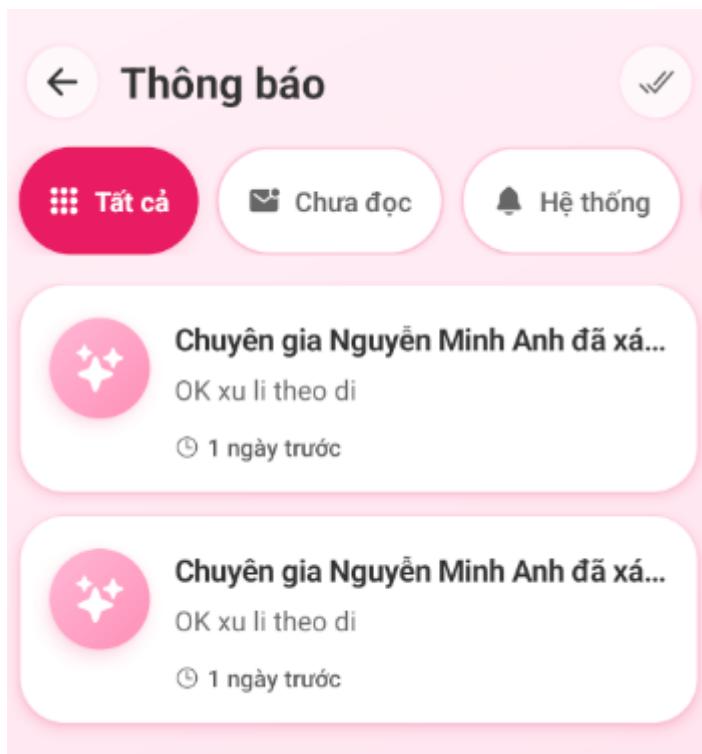


3.2.13.13. Notification Settings

Step 1 :



Step 2 :



3.2.13.14. Privacy Settings

Step 1 :



Step 2 :

Chính sách bảo mật

1. Information We Collect

We collect information you provide directly to us, including your name, email address, phone number, and your cat's information (name, breed, age, photos, personality traits).

Điều khoản dịch vụ

1. Acceptance of Terms

By accessing and using Pawnder, you accept and agree to be bound by the terms and provision of this agreement. If you do not agree to abide by the above, please do not use this service.

3.2.13.15. About Application

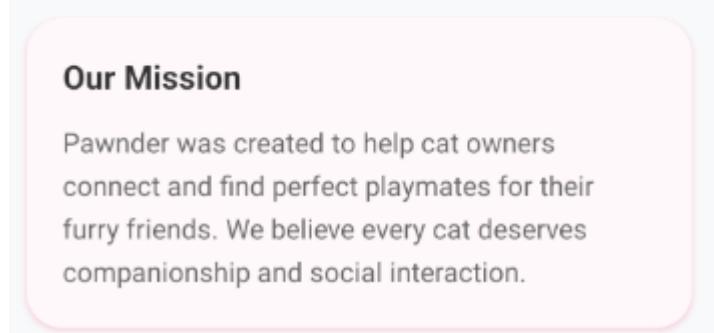
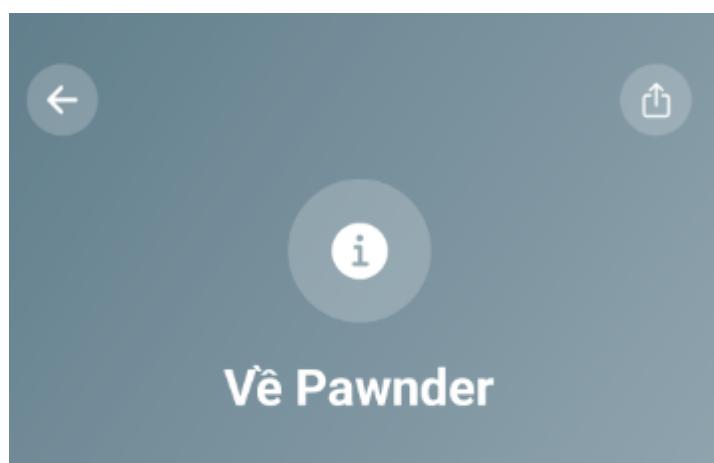
Step 1 :



Step 2 :



Step 3:



3.3. Admin (Web)

3.3.1. Authentication

3.3.1.1. Admin Login

The image shows a login form with a pink header featuring a heart icon. The main text says "Chào mừng trở lại" (Welcome back) and "Đăng nhập vào không gian làm việc của bạn" (Log in to your workspace). The form includes fields for "Địa chỉ email" (Email address) and "Mật khẩu" (Password), both with eye icon placeholders. There is a checkbox for "Ghi nhớ đăng nhập" (Remember login). A large pink-to-teal gradient button at the bottom right contains the text "Đăng nhập" (Log in).

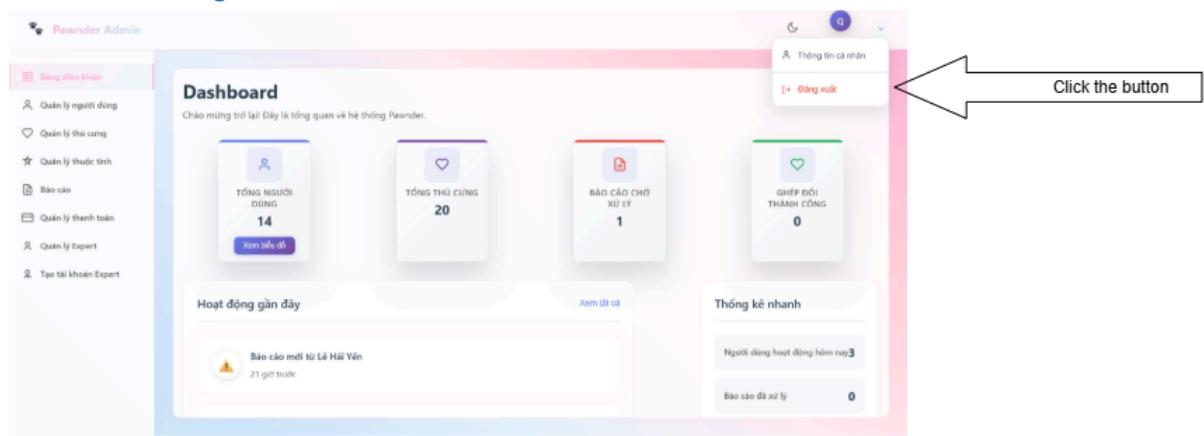
Fill in valid information →

Fill in valid information →

Click the button →

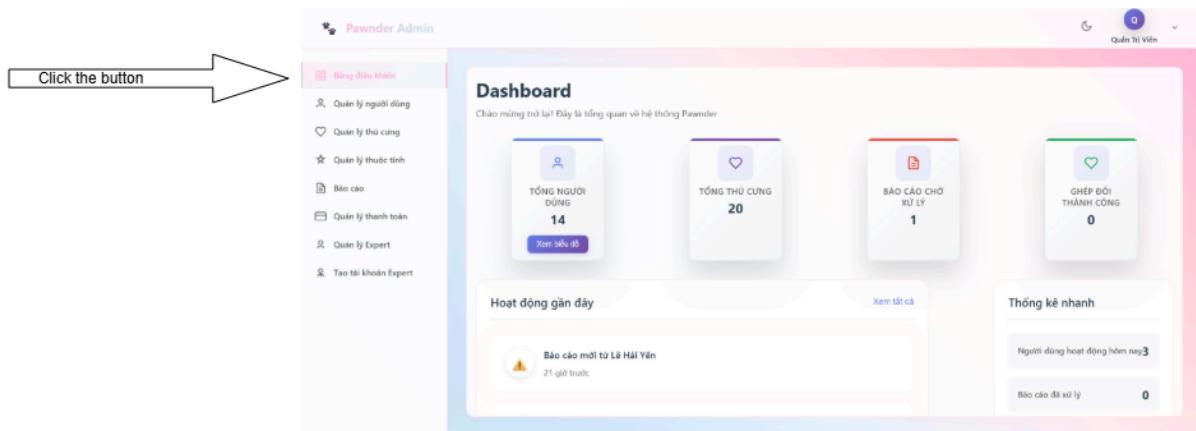
Click the button →

3.3.1.2. Admin Logout

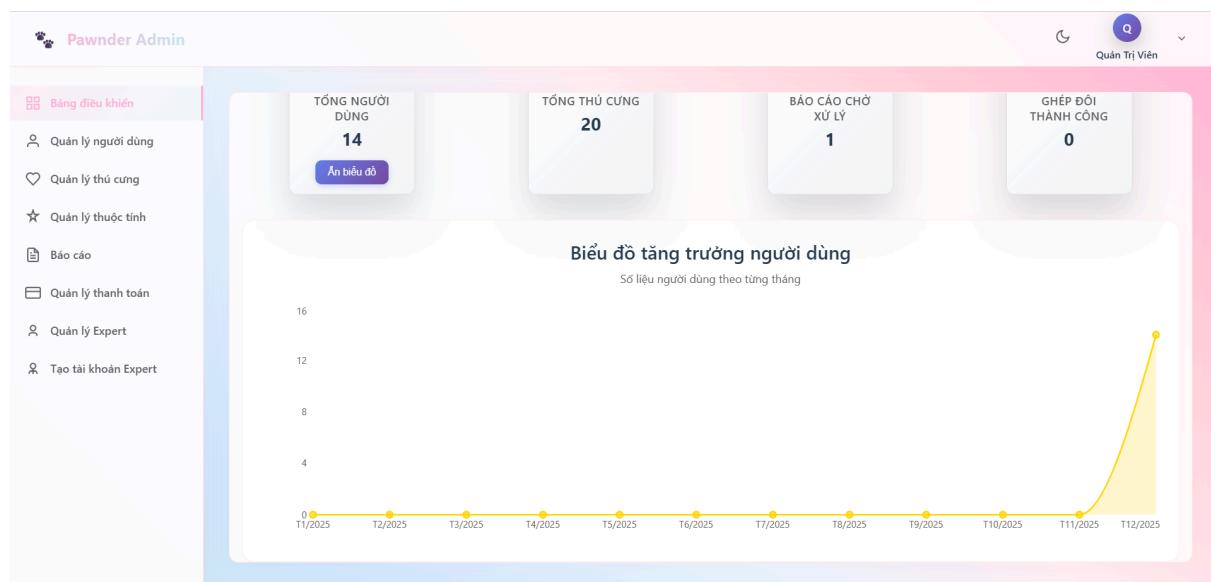
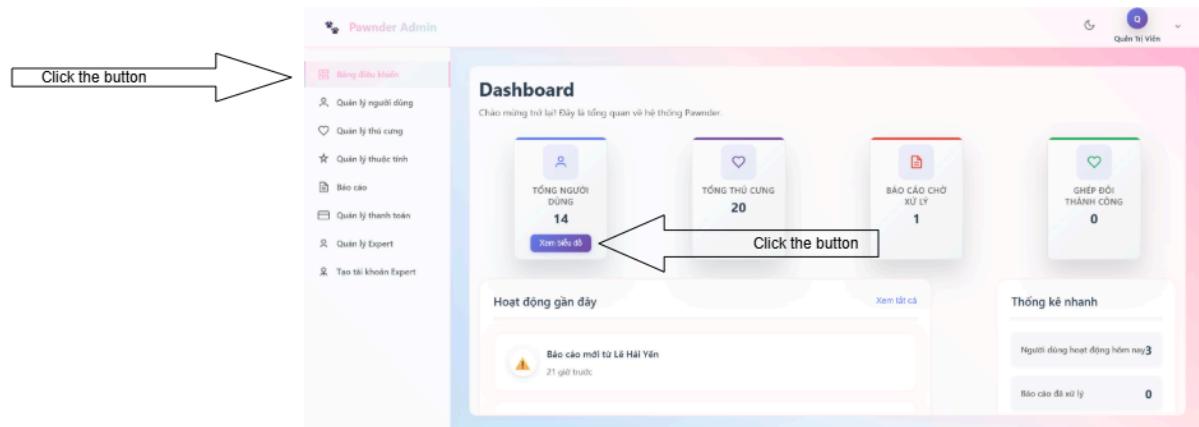


3.3.2. Dashboard

3.3.2.1. View Dashboard Overview



3.3.2.2. View Statistics Charts



3.3.2.3. View Recent Activities

The screenshot displays two views of the Pawnder Admin dashboard under the 'Quản Trị Viên' role.

Top View: Shows a summary timeline from T1/2025 to T12/2025 with a total count of 4 activities. A large arrow points to a button labeled 'Click the button' which links to the detailed activity list. To the right, there's a 'Thống kê nhanh' (Quick Statistics) section showing 3 new user reports and 0 resolved reports.

Bottom View: A detailed view of the 'Hoạt động gần đây' (Recent Activity) section. It shows 14 total activities, 14 new users, and 0 new reports. Three specific activity items are listed:

- Báo cáo mới từ Lê Hải Yến** (Report from Le Hai Yen) - 21 giờ trước (21 hours ago)
- Người dùng mới đăng ký: Ngô Tuấn Kiệt** (New user registered: Ngan Tuann Kiet) - 2 ngày trước (2 days ago)
- Người dùng mới đăng ký: Đặng Hoàng Long** (New user registered: Dang Hoang Long) - 2 ngày trước (2 days ago)

3.3.3. User Management

3.3.3.1. View Users List

The screenshot shows the Pawnder Admin dashboard with the 'Quản lý người dùng' (User Management) section selected. On the left sidebar, there are several other menu items: 'Bảng điều khiển', 'Quản lý người dùng' (highlighted in pink), 'Quản lý thú cưng', 'Quản lý thuộc tính', 'Báo cáo', 'Quản lý thanh toán', 'Quản lý Expert', and 'Tạo tài khoản Expert'. The main content area is titled 'Quản lý người dùng' and displays a summary of user statistics: 11 TOTAL USERS, 10 NORMAL users, 1 PREMIUM user, and 10 VERIFIED users. Below this, a table lists individual users with columns for Avatar, Thông tin cá nhân (Name, Email, Gender, Status), Liên hệ (Phone), Trạng thái (Status), Thông kê (Statistics), Ngày tạo (Created Date), and Hành động (Actions). One row is highlighted for 'Le Tuan'.

3.3.3.2. View User Details

Step 1:

This screenshot shows the 'User Details' page within the Pawnder Admin interface. The left sidebar remains the same as the previous screenshot. The main content area displays a table of users with the following columns: Avatar, Thông tin cá nhân (Name, Email, Gender, Status), Liên hệ (Phone), Trạng thái (Status), Thông kê (Statistics), Ngày tạo (Created Date), and Hành động (Actions). An arrow points to the 'Trạng thái' column of the first user, 'Le Tuan', which is currently set to 'NORMAL'. The user details for 'Le Tuan' are shown in a modal-like overlay.

Step 2:

Pawnder Admin

Quản Trị Viên

Bảng điều khiển

Quản lý người dùng

Quản lý thú cưng

Quản lý thuộc tính

Báo cáo

Quản lý thanh toán

Quản lý Expert

Tạo tài khoản Expert

Chi tiết người dùng

← Quay lại danh sách

Le Tuan

N/A • Male

X Chưa xác thực

0 THÚ CUNG

0 GHÉP ĐÔI

6/12/2025 THAM GIA

This screenshot shows the Pawnder Admin interface for managing users. On the left sidebar, there are several menu items including 'Bảng điều khiển', 'Quản lý người dùng' (selected), 'Quản lý thú cưng', 'Quản lý thuộc tính', 'Báo cáo', 'Quản lý thanh toán', 'Quản lý Expert', and 'Tạo tài khoản Expert'. The main content area is titled 'Chi tiết người dùng' (User Detail) with a back button '← Quay lại danh sách' (Back to list). It displays the user's name 'Le Tuan', gender 'Male', and status 'N/A'. A note 'X Chưa xác thực' (Not verified) is present. To the right, there are three boxes showing '0 THÚ CUNG' (0 Pets), '0 GHÉP ĐÔI' (0 Spay/Neuter), and a date '6/12/2025' with the word 'THAM GIA' (Participated). The overall theme is pink and light blue.

Pawnder Admin

Quản Trị Viên

Bảng điều khiển

Quản lý người dùng

Quản lý thú cưng

Quản lý thuộc tính

Báo cáo

Quản lý thanh toán

Quản lý Expert

Tạo tài khoản Expert

Thông tin cá nhân

Thú cưng

Ghép đôi

Hoạt động

Thông tin liên hệ

Email: lqt1@satato.com.vn

Số điện thoại: N/A

Thông tin cá nhân

Giới tính: Male

Vai trò: User

Trạng thái tài khoản

Trạng thái: NORMAL

Xác thực: X Chưa xác thực

Thời gian

Ngày tạo: 01:12:56 6/12/2025

Cập nhật cuối: 01:13:27 6/12/2025

This screenshot shows a more detailed view of the Pawnder Admin user profile for Le Tuan. It includes tabs for 'Thông tin cá nhân' (Personal Information), 'Thú cưng' (Pet), 'Ghép đôi' (Spay/Neuter), and 'Hoạt động' (Activities). Under 'Thông tin cá nhân', it shows contact details like email and phone number, and personal details like gender and role. The 'Trạng thái tài khoản' (Account Status) section shows 'NORMAL' status and 'X Chưa xác thực' (Not verified). The 'Thời gian' (Time) section provides creation and update dates. The overall layout is clean with a pink and white color scheme.

Pawnder Admin

Quản Trị Viên

Bảng điều khiển

Quản lý người dùng

Quản lý thú cưng

Quản lý thuộc tính

Báo cáo

Quản lý thanh toán

Quản lý Expert

Tạo tài khoản Expert

Thông tin liên hệ

Email: lqt1@satato.com.vn

Số điện thoại: N/A

Thông tin cá nhân

Giới tính: Male

Vai trò: User

Trạng thái tài khoản

Trạng thái: NORMAL

Xác thực: X Chưa xác thực

Thời gian

Ngày tạo: 01:12:56 6/12/2025

Cập nhật cuối: 01:13:27 6/12/2025

This screenshot is identical to the one above, showing the Pawnder Admin user profile for Le Tuan. It displays the same sections: 'Thông tin cá nhân', 'Trạng thái tài khoản', 'Thời gian', and 'Cập nhật cuối'. The data values remain the same as in the previous screenshots. The layout is consistent with the first one, featuring a pink and white color palette.

3.3.3.3. Edit User Information

The screenshot shows a user management interface for 'Pawnder Admin'. On the left, there's a sidebar with navigation links: 'Bảng điều khiển', 'Quản lý người dùng' (highlighted in pink), 'Quản lý thú cưng', 'Quản lý thuộc tính', 'Báo cáo', 'Quản lý thanh toán', 'Quản lý Expert', and 'Tạo tài khoản Expert'. The main area displays a list of users:

Tên	Giới tính	Email	Mức độ	Thứ cung	Ngày tạo	Thời gian cuối
Le Tuan	Male	lqt1@sata.com.vn	NORMAL	0	6/12/2025 01:12:50	
Ngô Tuấn Kiệt	Male	user3@pawnder.com	PREMIUM	2	4/12/2025 00:38:51	
Trần Thu Hà	Female	user1@pawnder.com	NORMAL	3	2/12/2025 16:38:54	
Lê Hải Yến	Female	user2@pawnder.com	NORMAL	1	2/12/2025 16:38:54	
Nguyễn Minh An	Male	user4@pawnder.com	NORMAL	2	2/12/2025 16:38:54	

Each user row has two small circular icons with arrows at the end of the list.

3.3.3.4. Ban User (Temporary)

The dialog box is titled 'Xử lý sai phạm - Le Tuan'. It contains the following fields:

Ban người dùng

Thời gian ban:

1 ngày

Lý do ban: *

Nhập lý do ban người dùng...

Thời gian ban sẽ hết hạn vào:
13:47:01 7/12/2025

Hủy Xác nhận ban

Annotations:

- A large arrow points to the '1 ngày' dropdown with the text 'Click the button'.
- A large arrow points to the 'Nhập lý do ban người dùng...' text area with the text 'Fill in valid information'.
- A large arrow points to the 'Xác nhận ban' button with the text 'Click the button'.

3.3.3.5. Ban User (Permanent)

Xử lý sai phạm - Le Tuan

Ban người dùng

Thời gian ban:

Vĩnh viễn

Lý do ban: *

Nhập lý do ban người dùng...

Hủy Xác nhận ban

Click the button

Fill in valid information

Click the button

3.3.3.6. Unban User

Step 1:

Avatar	Thông tin cá nhân	Tên tài khoản	Trạng thái	Thống kê	Ngày tạo	Hành động
LT	Le Tuan Mote	ltt1@pawnder.com	BANHỘ	THU ĐƯỢC: 0	6/12/2025 01/12/2025	
HT	Ngô Tấn Kiệt Mote	htt2@pawnder.com	Premium	THU ĐƯỢC: 2	4/12/2025 02/12/2025	
TT	Bùi Thủ Hà Femal	ttt3@pawnder.com	NORMAL	THU ĐƯỢC: 3	2/12/2025 10/12/2025	
LH	Lê Hải Yến Female	lhy4@pawnder.com	NORMAL	THU ĐƯỢC: 1	2/12/2025 10/12/2025	
NM	Nguyễn Minh An Male	nma5@pawnder.com	NORMAL	THU ĐƯỢC: 0	2/12/2025 10/12/2025	

Step 2:



3.3.3.7. View User Activity History

Avatar	Thông tin cá nhân	Liên hệ	Trạng thái	Thông kê	Ngày tạo	Hành động
	Le Tuan Male	lt1@pawnder.com	BANNED	THỦ ĐUNG: 8	6/12/2025 01:12:06	
	Ngõ Tuấn Kiệt Male	tuan2@pawnder.com	PREMIUM	THỦ ĐUNG: 2	4/12/2025 00:09:51	

3.3.4. Pet Management

3.3.4.1. View Pets List

Ảnh	Thông tin thú cưng	Chủ sở hữu	Hành động
	Poodle Male - 0 tuổi Kindly dog who loves running.	Triệu Thủ Hà user1@pawnder.com 00.1	
	Shih Tzu Female - 3 tuổi Smart and alert companion.	Triệu Thủ Hà user1@pawnder.com 00.2	
	Arih Male - 0 tuổi Very Cute	Triệu Thủ Hà user1@pawnder.com 00.3	
	Corgi Female - 0 tuổi Energetic and playful.	Le Hải Yến user2@pawnder.com 00.4	

3.3.4.2. View Pet Details

Step 1:

Pawnder Admin

Danh sách tất cả thú cưng trong hệ thống

Tìm kiếm theo tên, giống, chủ sở hữu...

Ảnh	Thông tin thú cưng	Chủ sở hữu	Hành động
	Milo Poodle Male + 0 tuổi Kindly dog who loves running.	Trần Thu Hà user1@pawnder.com ID: 1	
	Luna Shiba Inu Female - 3 tuổi Smart and alert companion.	Trần Thu Hà user1@pawnder.com ID: 2	
	Sofia Shiba Inu Male + 0 tuổi Very Cute	Trần Thu Hà user1@pawnder.com ID: 3	
	Bella Corgi Female - 0 tuổi Energetic and playful.	Le Hải Yến user1@pawnder.com ID: 4	

Step 2:

Chi tiết thú cưng

Ảnh của Milo

Milo

Thông tin cơ bản

Giống:	Poodle
Tuổi:	0 tuổi
Giới tính:	Male

Chủ sở hữu

Tên: Trần Thu Hà

3.3.4.3. View Pet Photos

Ảnh của Milo

Milo

Thông tin cơ bản

Giống:	Poodle
Tuổi:	0 tuổi
Giới tính:	Male

Chủ sở hữu

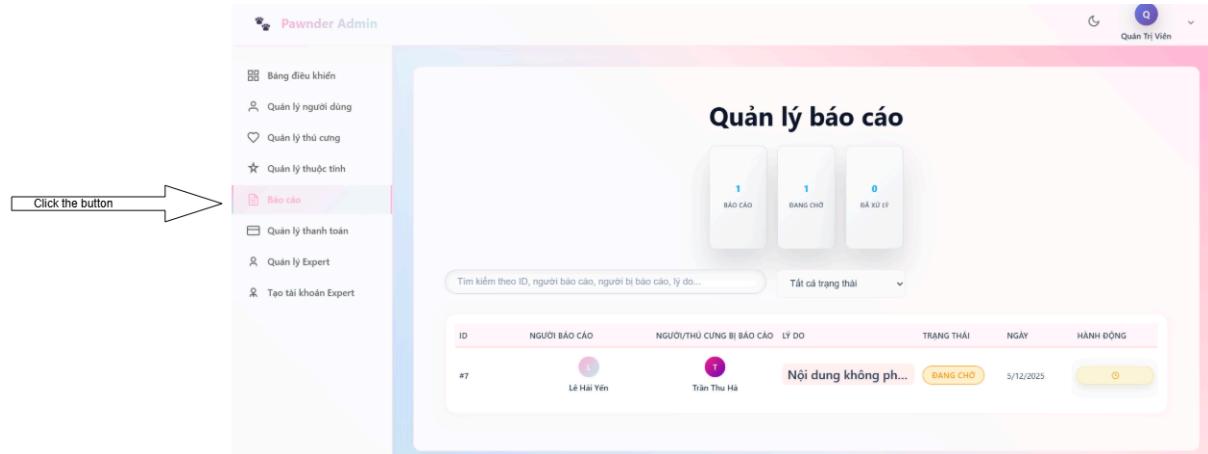
Tên: Trần Thu Hà

Email: user1@pawnder.com

ID: 3

3.3.5. Report Management

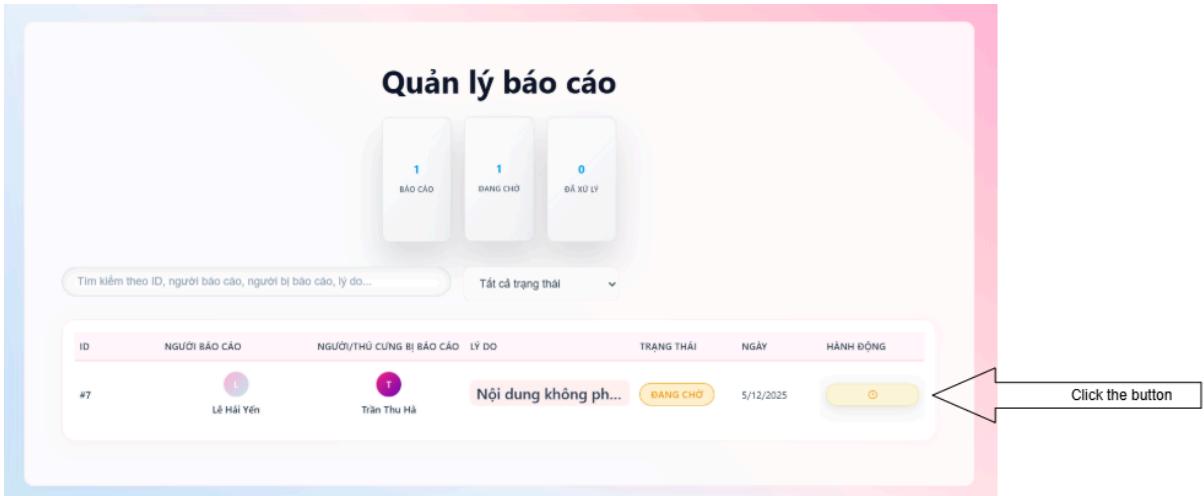
3.3.5.1. View Reports List



The screenshot shows the Pawnder Admin interface with a sidebar on the left containing various management links. A large white arrow points to the 'Báo cáo' (Report) link in the sidebar. The main content area is titled 'Quản lý báo cáo' and displays summary statistics: 1 Báo cáo, 1 Đang chờ, and 0 Đã xử lý. Below this is a search bar and a table listing report details. The table has columns for ID, Người báo cáo, Người/Thú cưng bị báo cáo, Lý do, Trạng thái, Ngày, and Hành động. One row is shown for report #7, which was filed by Lê Hải Yến and is currently in the 'Nội dung không phù hợp' (Content does not fit) status.

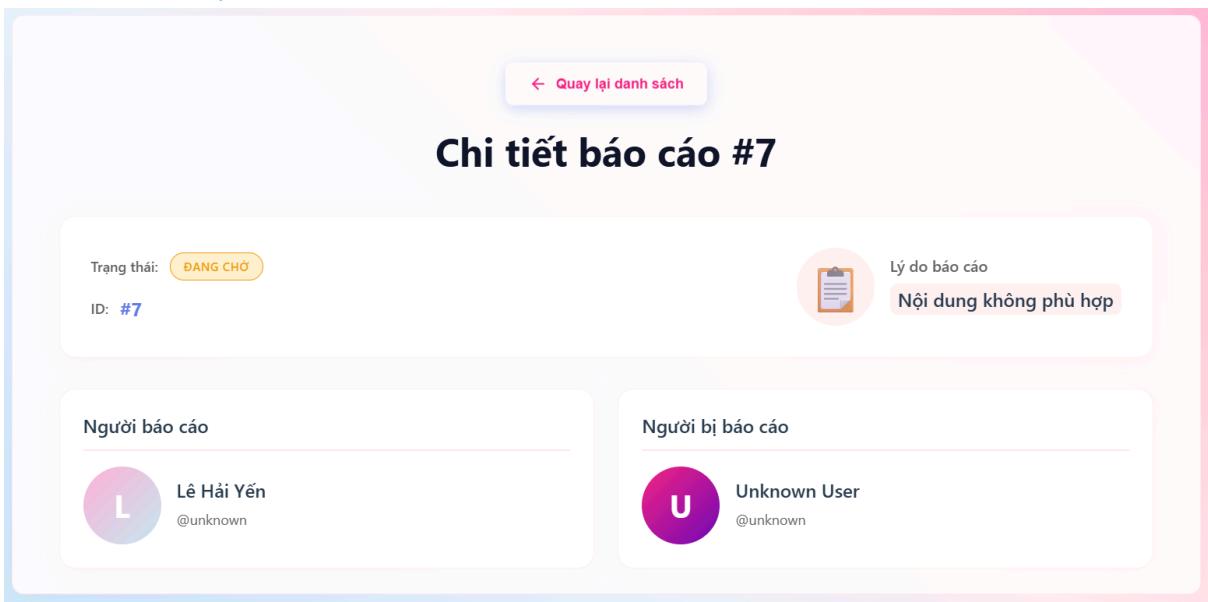
ID	NGƯỜI BÁO CÁO	NGƯỜI/THÚ CUNG BỊ BÁO CÁO	LÝ DO	TRẠNG THÁI	NGÀY	HÀNH ĐỘNG
#7	Lê Hải Yến	Trần Thu Hà	Nội dung không phù hợp...	DANG CHOI	5/12/2025	

3.3.5.2. View Report Details



This screenshot shows the same 'Quản lý báo cáo' page as above, but it is zoomed in on report #7. A large white arrow points to the 'DANG CHOI' (Pending) status button in the table row. The rest of the interface is identical to the previous screenshot.

3.3.5.3. Review Reported Content



The screenshot shows the detailed view for report #7. At the top right is a back button labeled '← Quay lại danh sách'. The title 'Chi tiết báo cáo #7' is centered. On the left, under 'Người báo cáo', is information for user Lê Hải Yến (@unknown). On the right, under 'Người bị báo cáo', is information for 'Unknown User' (@unknown). Between them is a circular icon with a clipboard symbol and the text 'Lý do báo cáo' (Reason for report) next to a box labeled 'Nội dung không phù hợp' (Content does not fit).

Mô tả chi tiết

Nội dung không phù hợp

Nội dung bị báo cáo

MESSAGE

N/A

Thời gian: 16:10:45 5/12/2025

Timeline

Báo cáo được tạo
16:10:45 5/12/2025

✓ Xử lý báo cáo
✗ Từ chối báo cáo

3.3.5.6. Ban Reported User

Mô tả chi tiết

Nội dung không phù hợp

Nội dung bị báo cáo

MESSAGE

N/A

Thời gian: 16:10:45 5/12/2025

Timeline

Báo cáo được tạo
16:10:45 5/12/2025

✓ Xử lý báo cáo
✗ Từ chối báo cáo

Click the button →

3.3.5.7. Dismiss Report

Mô tả chi tiết

Nội dung không phù hợp

Nội dung bị báo cáo

MESSAGE

N/A

Thời gian: 16:10:45 5/12/2025

Timeline

Báo cáo được tạo
16:10:45 5/12/2025

✓ Xử lý báo cáo
✗ Từ chối báo cáo

← Click the button

3.3.6. Payment Management

3.3.6.1. View Payment Transactions

The screenshot shows the Pawnder Admin dashboard. On the left sidebar, there are several menu items: Bảng điều khiển, Quản lý người dùng, Quản lý thủ cung, Quản lý thuộc tính, Báo cáo, Quản lý thanh toán (which is highlighted in pink), Quản lý Expert, and Tạo tài khoản Expert. The main content area is titled "Quản lý thanh toán" and displays a summary: Tổng doanh thu 119.000 ₫ and Tổng giao dịch 3. Below this is a search bar with placeholder text "Tìm kiếm theo email, tên, mã giao dịch...". A table lists three users with their transaction details:

Người dùng	Số tiền	Gói Premium	Trạng thái	Ngày bắt đầu	Ngày kết thúc
Lê Hải Văn	99.000 ₫	Gói VIP	HẾT HẠN	01/12/2025	03/12/2025
Ngo Xuân Kiệt	10.000 ₫	Gói VIP	ACTIVE	04/12/2025	04/01/2026
Trần Thủ Hà	10.000 ₫	Gói VIP	HẾT HẠN	01/11/2025	03/12/2025

3.3.6.6 .Revenue chart

Step 1:

The screenshot shows the Pawnder Admin dashboard. On the left sidebar, there are several menu items: Bảng điều khiển, Quản lý người dùng, Quản lý thủ cung, Quản lý thuộc tính, Báo cáo, Quản lý thanh toán (which is highlighted in pink), Quản lý Expert, and Tạo tài khoản Expert. The main content area is titled "Quản lý thanh toán" and displays a summary: Tổng doanh thu 119.000 ₫ and Tổng giao dịch 3. Below this is a search bar with placeholder text "Tìm kiếm theo email, tên, mã giao dịch...". A button labeled "Click the button" points to the "Quản lý thanh toán" section. A large arrow points from the button to the section. A table lists three users with their transaction details:

Người dùng	Số tiền	Gói Premium	Trạng thái	Ngày bắt đầu	Ngày kết thúc
Lê Hải Văn	99.000 ₫	Gói VIP	HẾT HẠN	01/12/2025	03/12/2025
Ngo Xuân Kiệt	10.000 ₫	Gói VIP	ACTIVE	04/12/2025	04/01/2026
Trần Thủ Hà	10.000 ₫	Gói VIP	HẾT HẠN	01/11/2025	03/12/2025

Step 2:



3.3.7. Expert Management

3.3.7.1. View Expert Applications

The screenshot shows the Pawnder Admin interface. On the left sidebar, there is a navigation menu with the following items:

- Bảng điều khiển
- Quản lý người dùng
- Quản lý thứ cung
- Quản lý thuộc tính
- Báo cáo
- Quản lý thanh toán
- Quản lý Expert** (highlighted with a red box and an arrow pointing to it from the left)
- Tạo tài khoản Expert

The main content area is titled "Quản lý Expert" and contains the sub-instruction: "Xem danh sách, chi tiết và cập nhật thông tin tài khoản chuyên gia." Below this, there is a table listing three experts:

ID	HỌ TÊN	EMAIL	NGÀY TẠO	Chi tiết
2	Nguyễn Minh Anh	expert@pawnder.com	2/12/2025	Chi tiết
5	Phạm Quốc Hưng	expert1@pawnder.com	3/12/2025	Chi tiết
6	Đặng Hoàng Long	expert2@pawnder.com	3/12/2025	Chi tiết

At the bottom, there is a footer with the text "Tổng: 3 expert" and navigation links: Trước, Trang 1/1, Sau.

3.3.7.2. Create New Expert

Pawnder Admin

Thông tin chuyên gia

Họ và tên *
VD: Nguyễn Văn A Fill in valid information

Giới tính
Khác / Không biết
Click the button

EMAIL TỰ ĐỘNG
expert3@pawnder.com
Hệ thống cấp email đang expert + số thứ tự nhỏ nhất chưa dùng.

MẬT KHẨU ĐỂ XUẤT
Cute243 Đổi mật khẩu
Mật khẩu ngắn nhất, dễ nhớ cho chuyên gia.

Tạo tài khoản Click the button

Danh sách 3 tài khoản Expert đang được quản lý nhanh.

- Nguyễn Minh Anh
expert@pawnder.com
- Phạm Quốc Hưng
expert1@pawnder.com
- Đặng Hoàng Long
expert2@pawnder.com

3.3.7.3. View Expert Details

Step 1:

Quản lý Expert

Xem danh sách, chi tiết và cập nhật thông tin tài khoản chuyên gia.

ID	HỌ TÊN	EMAIL	NGÀY TẠO	Chi tiết
2	Nguyễn Minh Anh	expert@pawnder.com	2/12/2025	Chi tiết
5	Phạm Quốc Hưng	expert1@pawnder.com	3/12/2025	Chi tiết
6	Đặng Hoàng Long	expert2@pawnder.com	3/12/2025	Chi tiết

Tổng: 3 expert

Trước Trang 1/1 Sau

Chi tiết Expert

Quay lại danh sách Expert

Thông tin cơ bản Cập nhật họ tên, giới tính, Email và vai trò được cố định.

Họ và tên
Nguyễn Minh Anh

Email
expert@pawnder.com

Giới tính
Nữ

Đặt lại mật khẩu Tạo mật khẩu mới khi có người mới tiếp quản tài khoản Expert. Mật khẩu cũ không được hiển thị lại.

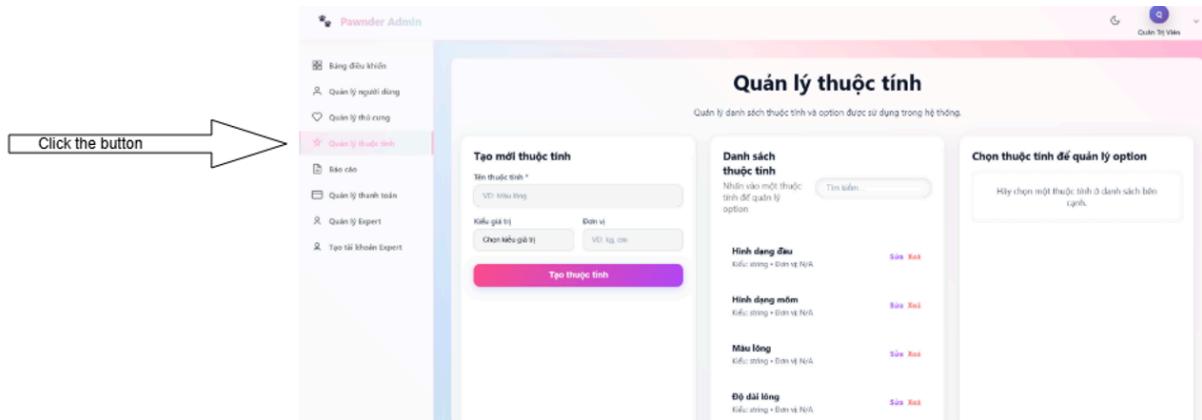
Sau khi đặt lại, bạn cần gửi mật khẩu mới cho Expert hoặc cập nhật qua kênh nội bộ.

Đặt lại mật khẩu

Lưu thông tin

3.3.8. Attribute Management

3.3.8.1. View Attributes List



3.3.8.2. Create New Attribute

This is a detailed view of the 'Tạo mới thuộc tính' (Create New Attribute) form. It has fields for 'Tên thuộc tính*' (e.g., VD: Màu lông), 'Kiểu giá trị' (Chọn kiểu giá trị), 'Đơn vị' (e.g., VD: kg, cm), and a large purple 'Tạo thuộc tính' button. Arrows and text boxes guide the user through the process: 'Fill in valid information' points to the attribute name field; 'Click the button' points to the 'Chọn kiểu giá trị' button; 'Fill in valid information' points to the unit field; and another 'Click the button' points to the main 'Tạo thuộc tính' button.

3.3.8.3. Edit Attribute

Step 1:

Danh sách thuộc tính

Nhấn vào một thuộc
tính để quản lý
option

Tìm kiếm...

Hình dạng đầu

Sửa Xoá

Kiểu: string • Đơn vị: N/A

Hình dạng mõm

Click the button

Sửa Xoá

Kiểu: string • Đơn vị: N/A

Màu lông

Sửa Xoá

Kiểu: string • Đơn vị: N/A

Độ dài lông

Sửa Xoá

Kiểu: string • Đơn vị: N/A

Kiểu lông

Sửa Xoá

Kiểu: string • Đơn vị: N/A

Step 2:

Option của "Hình dạng mõm"

Thêm option mới

Nhập tên option

Thêm

Ngắn

Click the button

Sửa Xoá

Trung bình

Sửa Xoá

Dài

Sửa Xoá

3.3.8.4. Add Attribute Options

Cập nhật thuộc tính

+ Thuộc tính mới

Tên thuộc tính *

Fill in valid information

Hình dạng đầu

Kiểu giá trị

Click the button

Chuỗi (string)

Đơn vị

Vd: kg, cm

Fill in valid information

Lưu thay đổi

Click the button

3.3.8.6. Delete Attribute

Danh sách thuộc tính

Nhấn vào một thuộc tính để quản lý option

Tim kiếm...

Hình dạng đầu
Kiểu: string • Đơn vị: N/A

Hình dạng mõm
Kiểu: string • Đơn vị: N/A

Màu lông
Kiểu: string • Đơn vị: N/A

Độ dài lông
Kiểu: string • Đơn vị: N/A

Kiểu lông
Kiểu: string • Đơn vị: N/A

Sửa Xoá

Click the button

3.4. Expert (Web)

3.4.1. Authentication

3.4.1.1. Expert Login

The image shows a login form titled "Chào mừng trở lại" (Welcome back). It includes fields for "Địa chỉ email" (Email address) and "Mật khẩu" (Password), both with placeholder text. A checkbox for "Ghi nhớ đăng nhập" (Remember login) is present. A large pink button at the bottom is labeled "Đăng nhập" (Login). Annotations with arrows point to each field and the button, indicating the steps: "Fill: Địa chỉ email" points to the email input, "Fill: Mật khẩu" points to the password input, and "CLICK The button: Đăng nhập" points to the login button.

Chào mừng trở lại

Đăng nhập vào không gian làm việc của bạn

Địa chỉ email

Mật khẩu

Ghi nhớ đăng nhập

Đăng nhập

← Fill: Địa chỉ email

← Fill: Mật khẩu

← CLICK The button: Đăng nhập

3.4.1.2. Expert Logout



3.4.2. Dashboard

3.4.2.1. View Expert Dashboard



3.4.2.2. View Expert Notifications

The screenshot shows a dashboard titled "Quản lý thông báo" (Manage Notifications). At the top, there are three summary boxes: "TỔNG SỐ THÔNG BÁO" (Total Notifications) with value 1, "CHỜ XỬ LÝ" (Pending) with value 0, and "ĐÃ XÁC NHẬN" (Confirmed) with value 1. Below these are search and filter buttons: "Tim kiếm theo tên người dùng...", "Xem tất cả đã xử lý" (View all processed), and "Làm mới" (Refresh). A table below lists notifications with columns: #, Người dùng (User), Câu hỏi người dùng (User Question), Nội dung (Content), Ngày tạo (Created Date), Trạng thái (Status), and Thao tác (Action). A message at the bottom states: "Không có thông báo nào chờ xử lý" (No notifications pending).

3.4.2.3. View Pending Consultations

The screenshot shows a large blue button labeled "Xem chờ xử lý" (View pending). Below it is a table with columns: #, Người dùng (User), Câu hỏi người dùng (User Question), Nội dung (Content), Ngày tạo (Created Date), Trạng thái (Status), and Thao tác (Action). A message at the bottom states: "Không có thông báo nào chờ xử lý" (No notifications pending).

3.4.2.4. View Success Consultations

The screenshot shows a large blue button labeled "Xem tất cả đã xử lý" (View all processed). Below it is a table with columns: #, Người dùng (User), Câu hỏi người dùng (User Question), Nội dung (Content), Ngày tạo (Created Date), Trạng thái (Status), and Thao tác (Action). A row is shown with details: User Lê Hải Yến, Question Phai xu li the nao day, Content OK xu li theo di, Created Date 15:03 05/12/2025, Status ĐÃ XÁC NHẬN (Confirmed), and Action button. A message at the bottom states: "Không có thông báo nào chờ xử lý" (No notifications pending).

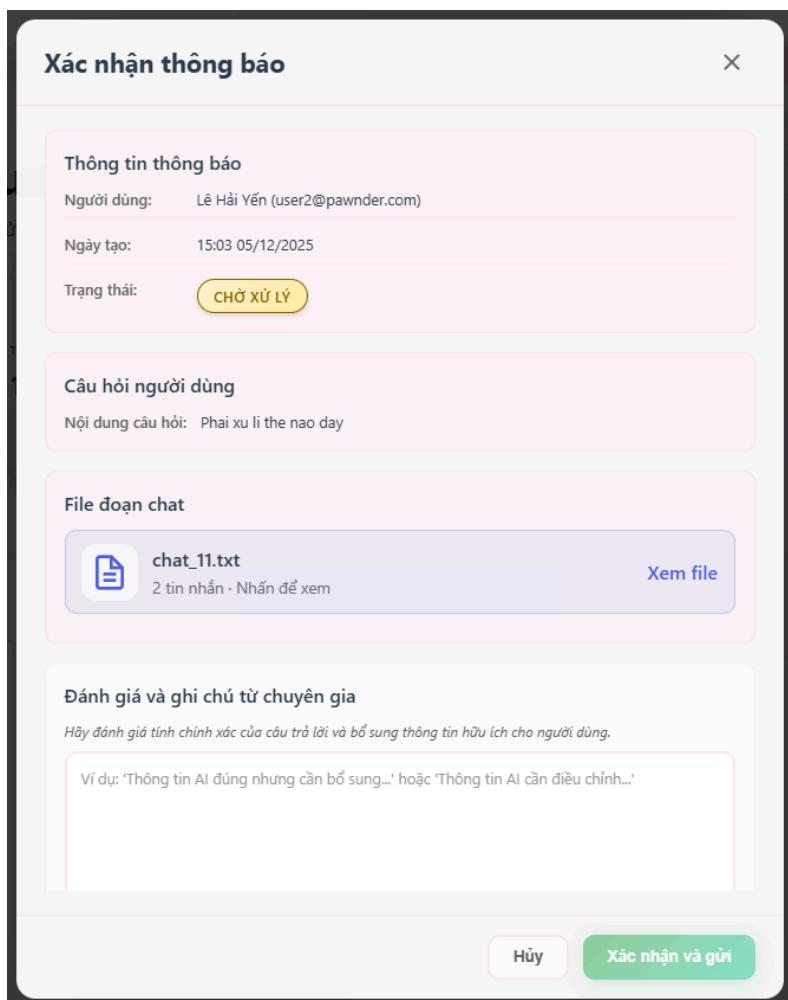
3.4.3. Consultation Management

3.4.3.1. View Consultation Requests

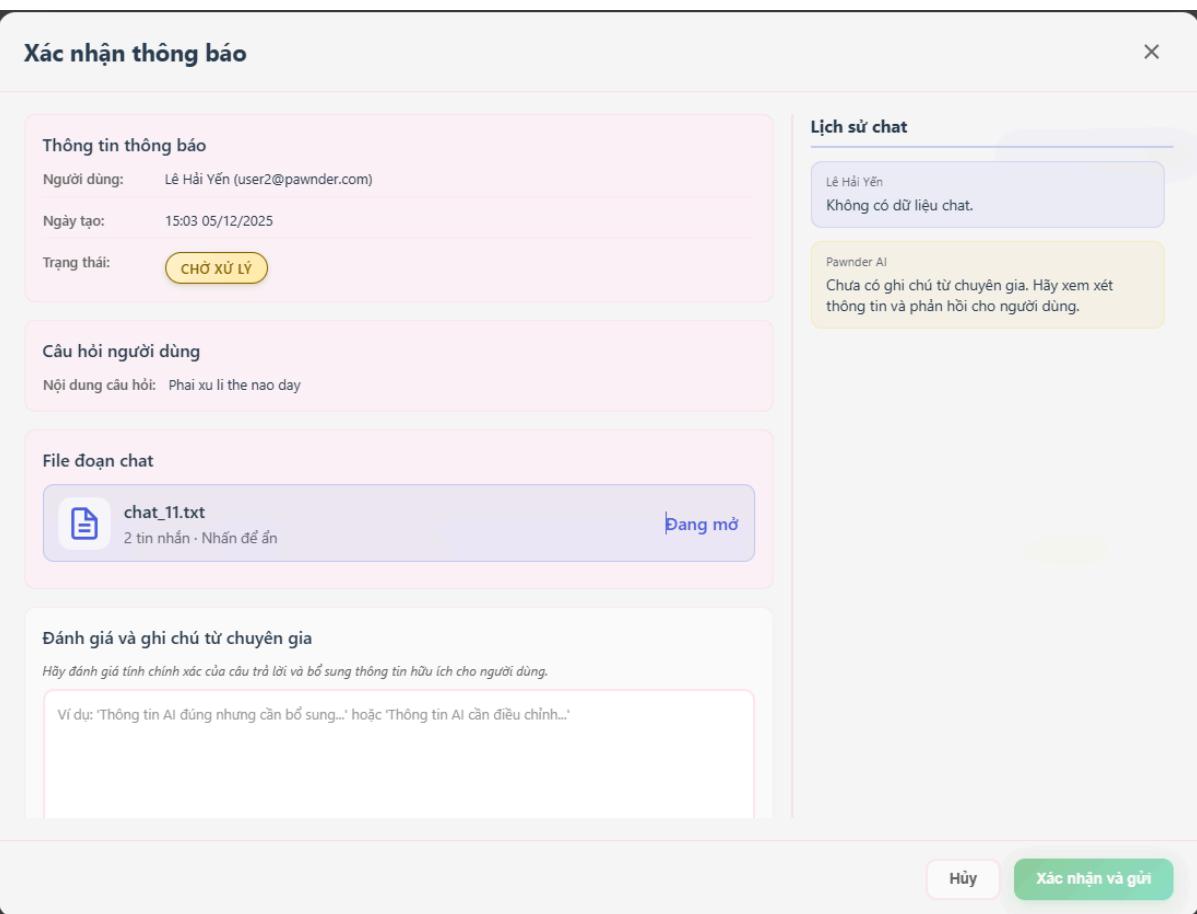
Step 1: click button “Xác nhận”

#	Người dùng	Câu hỏi người dùng	Nội dung	Ngày tạo	Trạng thái	Thao tác
1	Lê Hải Yến	Phai xu li the nao day	-	15:03 05/12/2025	<button>CHỜ XỬ LÝ</button>	<button>Xác nhận</button>

Step 2: View all info notification, user question, file chat, input text by expert

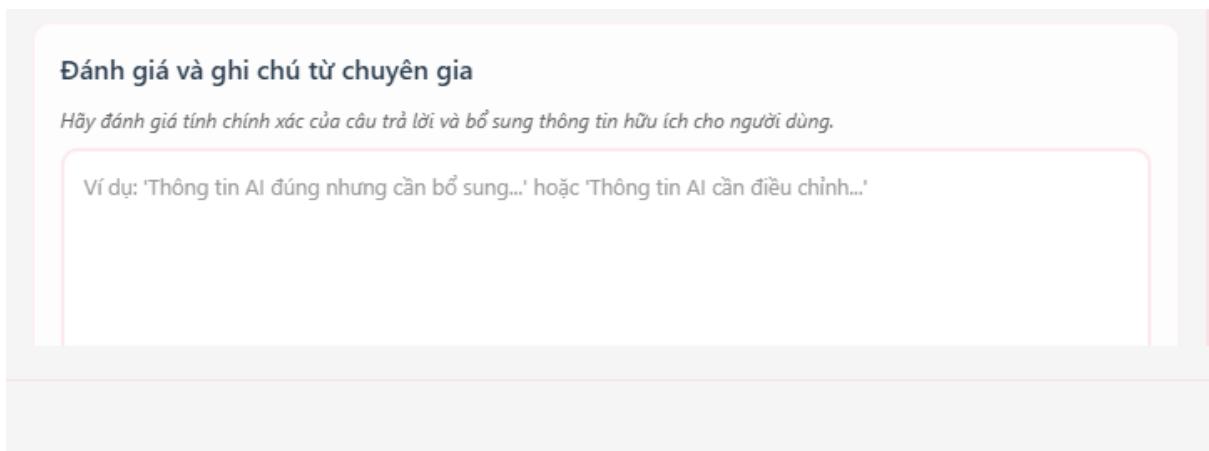


Step 3: click “Xem file” to read history chat User x AI



3.4.3.2. Accept Consultation

Step 1: input text



Step 2: Click button "Xác nhận và gửi" -> Accept and send to notification user

Đánh giá và ghi chú từ chuyên gia

Hãy đánh giá tính chính xác của câu trả lời và bổ sung thông tin hữu ích cho người dùng.

thông tin từ AI rất chính xác.

Hủy **Xác nhận và gửi**

3.4.3.3. View Consultation Successful

Step 1: click button “eye” in column

#	Người dùng	Câu hỏi người dùng	Nội dung	Ngày tạo	Trạng thái	Thao tác
1	Lê Hải Yến	Phai xu li the nao day	OK xu li theo di	15:03 05/12/2025	ĐÃ XÁC NHẬN	

Step 2: view all info and read chat: user x AI

Xác nhận thông báo

Thông tin thông báo

Người dùng: Lê Hải Yến (user2@pawnder.com)

Ngày tạo: 15:03 05/12/2025

Trạng thái: **ĐÃ XÁC NHẬN**

Câu hỏi người dùng

Nội dung câu hỏi: Phai xu li the nao day

File đoạn chat

chat_11.txt 2 tin nhắn · Nhấn để ẩn **Đang mở**

Ghi chú đã gửi cho người dùng

OK xu li theo di

Lịch sử chat

Lê Hải Yến
Không có dữ liệu chat.

Pawnder AI
Chưa có ghi chú từ chuyên gia. Hãy xem xét thông tin và phản hồi cho người dùng.

Đóng

3.4.3.4. Open Chat with User

Step 1:



Step 2: choose user to chat

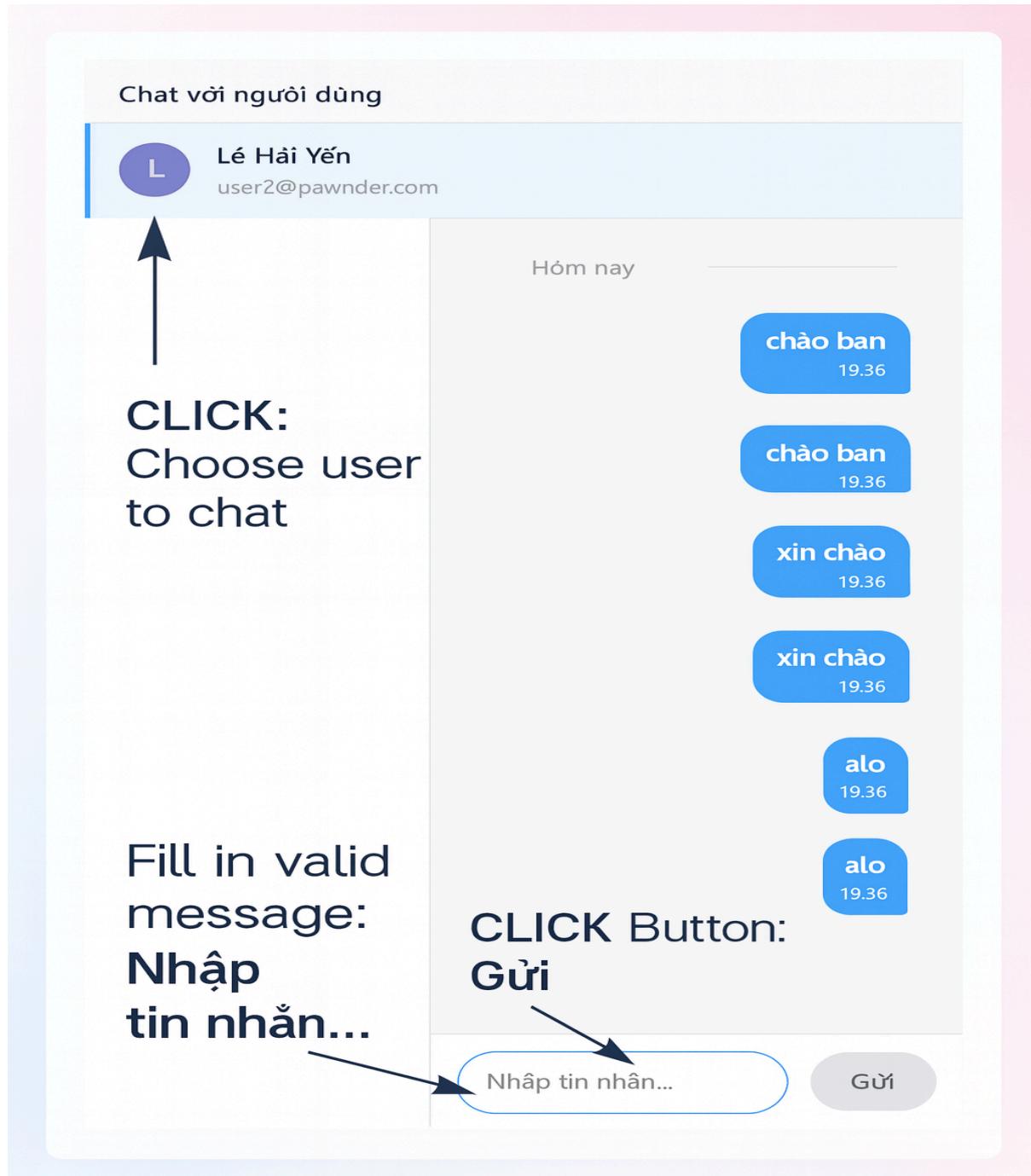
Chat với người dùng



Lê Hải Yến

user2@pawnder.com

3.4.3.5. Send Messages



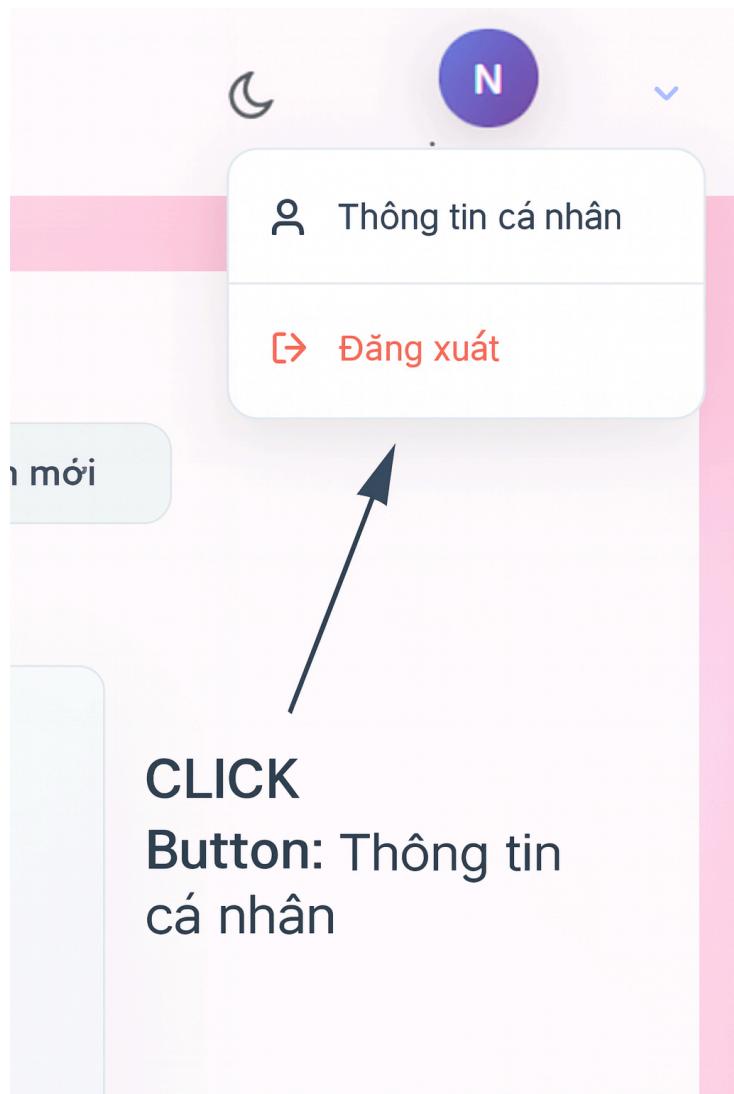
3.4.3.6. View Consultation History

The screenshot shows a digital communication interface for a consultation. At the top left is a purple circular profile icon with a white letter 'L'. To its right, the name 'Lê Hải Yên' is displayed in black text, followed by the email 'user2@pawnder.com' in smaller black text. In the center, the word 'Hôm nay' (Today) is written in black. On the right side, there is a vertical column of four blue rounded rectangular message bubbles. Each bubble contains a white lowercase letter and a timestamp. The first bubble says 'chào bạn' at '19:36'. The second says 'xin chào' at '19:36'. The third says 'alo' at '19:36'. The fourth says 'd' at '20:02'.

Message	Time
chào bạn	19:36
xin chào	19:36
alo	19:36
d	20:02

3.4.4. Profile Management

3.4.4.1. View Expert Profile



Thông tin Expert

X



Nguyễn Minh Anh

Expert

Thông tin cơ bản

EMAIL:

expert@pawnder.com

HỌ VÀ TÊN:

Nguyễn Minh Anh

GIỚI TÍNH:

Không tiết lộ

VAI TRÒ:

Expert

Đóng