



COMSATS University, Islamabad Pakistan

FelineConnect

By

Saifullah CIIT/SP21-BSE-104/ISB

Talha Ziaullah CIIT/SP20-BSE-094/ISB

Supervisor

Mr. Zahid Anwar

Bachelor of Science in Software Engineering (2021-2025)

The candidate confirms that the work submitted is their own and appropriate credit has been given where reference has been made to the work of others.



COMSATS University, Islamabad Pakistan

FelineConnect

**A project presented to
COMSATS University, Islamabad**

**In partial fulfillment
of the requirement for the degree of**

Bachelor of Science in Software Engineering (2021-2025)

By

Saifullah CIIT/SP21-BSE-104/ISB

Talha Ziaullah CIIT/SP20-BSE-094/ISB

DECLARATION

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other, we will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Saifullah

Talha Ziaullah

CERTIFICATE OF APPROVAL

It is to certify that the final year project of BS (SE) FelineConnect was developed by **Saifullah (CIIT/SP21-BSE-104)** and **Talha Ziaullah (CIIT/SP20-BSE-094)** under the supervision of Zahid Anwar and that in his opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Software Engineering.

Supervisor

External Examiner

Head of Department
(Department of Computer Science)

Executive Summary

FelineConnect is a comprehensive web application created to improve the cat ownership experience by utilizing cutting-edge artificial intelligence and robust software design processes. This platform combines advanced capabilities for cat breed identification, e-commerce, user engagement, and administrative management to create a cohesive ecosystem for cat lovers.

FelineConnect is built around an AI-powered cat breed detection feature that analyzes user-uploaded pictures using ResNet-50 deep learning models. This revolutionary feature helps people better understand their pets by giving breed-specific features, behaviors, and health insights.

The application also contains an online store that sells a variety of cat breeds and cat food products. Users may customize their shopping experience by using straightforward search criteria and receiving timely notifications about new products, discounts, or order updates.

FelineConnect employs an AI-powered chatbot that is designed to reply to customer queries and assist with navigation, product selection, and other questions. Furthermore, a secure payment processing system ensures smooth transactions, and the admin interface enables easy control of user accounts, inventory, and feedback.

The development takes an object-oriented approach for scalability and maintainability, and it uses an incremental process model to allow for iterative enhancements based on user feedback. TensorFlow, Stripe API, and Gemini API are among the key technologies used.

FelineConnect combines technology innovation with user-centric design to alter how cat owners interact with their pets and the larger feline community. This idea is a forward-thinking solution that addresses the changing needs of pet owners in the digital age.

Acknowledgement

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Zahid Anwar”. Without their personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are grateful to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us with the values of honesty & hard work.

Saifullah

Talha Ziaullah

Abbreviations

SRS	Software Requirement Specification
AI	Artificial Intelligence
OOP	Object Oriented Programming
SDS	Software Design Document
URL	Uniform Resource Locator

Table of Contents

Contents

1	Introduction	10
1.1	Vision Statement.....	10
1.2	Related System Analysis/Literature Review.....	11
1.3	Project Deliverables	11
1.4	System Limitations/Constraints	11
1.5	Tools and Technologies	12
1.6	Relevance to Course Modules.....	13
2	Problem Definition.....	14
2.1	Problem Statement	14
2.2	Problem Solution.....	14
2.3	Objectives of the Proposed System.....	14
2.4	Scope.....	15
2.5	Modules.....	15
2.5.1	Module 1: User Authentication and Profile Management.....	15
2.5.2	Module 2: Cat Breed Detection.....	16
2.5.3	Module 3: Online Store	16
2.5.4	Module 4: Search Filters and Notifications.....	16
2.5.5	Module 5: Payment Processing	16
2.5.6	Module 6: AI Chatbot	16
2.5.7	Module 7: Admin Panel	17
2.5.8	Module 8: Statistical Analytics	17
3	Requirement Analysis	18
3.1	User classes and characteristics	18
3.2	Requirement Identifying Technique	19
3.2.1	Use Case Diagram.....	19
3.2.2	Detail Use Case:	22
3.3	Functional Requirements	47
3.3.1	Module 1: User Authentication and Profile Management.....	47
3.3.2	Module 2: Cat Breed Detection.....	62
3.3.3	Module 3: Online Store	74
3.3.4	Module 4: Search Filters and Notifications.....	81
3.3.5	Module 5: Payment Processing	82
3.3.6	Module 6: AI Chatbot	91
3.3.7	Module 7: Admin Panel	96
3.3.8	Module 8: Statistical Analytics	100
3.4	Non-Functional Requirements	104
3.4.1	NFR-1: Reliability.....	104
3.4.2	NFR-2: Usability	104
3.4.3	NFR-3: Performance	105
3.4.4	NFR-4: Portability	105
3.5	External Interface Requirements.....	106
3.5.1	User Interfaces Requirements	106
3.5.2	Software interfaces.....	106
3.5.3	Hardware interfaces.....	107
3.5.4	Communications interfaces	107
4	Design and Architecture.....	108

4.1	Architectural Design	108
4.2	Design Models	109
4.2.1	Activity Diagrams	109
4.2.2	Class Diagram	120
4.2.3	Sequence Diagram.....	121
4.2.4	State Transition Diagram	125
4.3	Data Design.....	130
4.3.1	Data Dictionary	130
5	Implementation.....	134
5.1	Algorithm	134
5.2	External APIs/SDKs	150
5.3	User Interface.....	151
5.4	Deployment.....	161
6	Testing and Evaluation.....	162
6.1	Unit Testing.....	162
6.1.1	Module 1: User Authentication and Profile Management.....	162
6.1.2	Module 2: Cat Breed Detection.....	163
6.1.3	Module 3: Online Store	164
6.1.4	Module 4: Search Filters and Notifications.....	165
6.1.5	Module 5: Payment Processing	166
6.1.6	Module 6: AI Chatbot	167
6.1.7	Module 7: Admin Panel	168
6.1.8	Module 8: Statistical Analytics	170
6.2	Functional Testing.....	171
6.2.1	Module 1: User Authentication and Profile Management.....	171
6.2.2	Module 2: Cat Breed Detection.....	172
6.2.3	Module 3: Online Store	172
6.2.4	Module 4: Search Filters and Notifications.....	173
6.2.5	Module 5: Payment Processing	173
6.2.6	Module 6: AI Chatbot	173
6.2.7	Module 7: Admin Panel	174
6.2.8	Module 8: Statistical Analytics	174
6.3	Business Rules Testing	175
6.3.1	Decision Table for Product Purchase	175
6.3.2	Decision Table for Admin Management	175
6.3.3	Test Cases for Product Purchase Decision Table.....	176
6.3.4	Test Cases for Admin Management Decision Table.....	176
6.4	Integration Testing	176
6.4.1	Product Purchase and Online Store	176
6.4.2	Search Filters and Notifications	177
6.4.3	Statistical Analytics and Payment Processing.....	177
6.4.4	User Authentication and AI Chatbot	178
6.4.5	Search Filters and Online Store.....	178
6.4.6	Payment Processing and Notifications	178
6.4.7	Admin Panel and Statistical Analytics	179
7	Conclusion and Future Work:.....	180
8	References.....	181
9	Plagiarism Report.....	181

1 Introduction

FelineConnect is an innovative web application designed to revolutionize the way cat owners interact with their feline companions. It leverages cutting-edge artificial intelligence (AI) technology to provide a comprehensive suite of features centered around cat breed identification, online cat and cat food shopping.

At its core, FelineConnect empowers users to identify the breed of their cats by simply uploading an image. The application's AI-powered breed detection engine utilizes deep learning models, specifically the ResNet-50 architecture, to accurately analyze cat images and provide reliable breed information. This not only satisfies curiosity but also helps owners better understand their pets' traits, behaviors, and potential health concerns.

Beyond breed identification, FelineConnect extends its capabilities to include an online store where users can explore a diverse selection of cat breeds available for purchase, along with a variety of cat food products. This seamlessly integrates the identification aspect with a marketplace, catering to the needs of both prospective cat owners and those seeking to enhance their existing pet's lifestyle.

To foster a sense of community and engagement, FelineConnect incorporates features such as search filters, notifications, and an AI-powered chatbot. Users can easily find specific cat breeds or products, stay updated on the latest offerings, and engage in interactive conversations with the chatbot to gather information and resolve queries.

FelineConnect also caters to the needs of administrators through a dedicated admin panel. This panel empowers administrators to manage user accounts, oversee product inventory, monitor orders, and gain insights into user feedback and model predictions.

1.1 Vision Statement

For cat owners and enthusiasts who seek a convenient and innovative way to identify cat breeds, purchase cat-related products, and enhance their pet ownership experience, The FelineConnect is a comprehensive web-based platform that offers accurate AI-powered cat breed identification, an integrated online store for cats and pet products, personalized notifications, and interactive AI chatbot assistance, all in one seamless application. Unlike traditional pet care systems or isolated e-commerce platforms, our product combines advanced AI, user engagement features, and robust administrative tools to create a unique, user-friendly

experience that enhances pet care, fosters community, and supports informed decision-making for cat owners.

1.2 Related System Analysis/Literature Review

Table 1: Related System Analysis with FelineConnect

Application Name	Weakness	Proposed Project Solution
Cat breed identifier App	Does not provide accurate result as it covers less cat breeds.	FelineConnect utilizes advanced deep learning and computer vision techniques to accurately identify cat breeds.
Cat Breeds App	User will pay for Cat breed Recognition feature.	FelineConnect will be free for each user to identify their cat breed accurately.
Cat Scanner App	Complex user interface	FelineConnect will adopt a minimalist design philosophy, reducing clutter and unnecessary elements to create a clean and uncluttered interface

1.3 Project Deliverables

Comprehensive SRS, SDS document, detailing the project's summary, architecture, design models, and data structures. Testing Reports, including unit, integration, and functional testing results and a web application.

1.4 System Limitations/Constraints

LI-1: The accuracy of breed identification may be affected by the quality and clarity of uploaded cat images, leading to potential misidentifications for low-quality images.

LI-2: The system may not be able to identify all cat breeds, especially rare or mixed breeds.

LI-3: While the system provides comprehensive cat care information and expert advice, it is not a substitute for professional veterinary consultation.

LI-4: The effectiveness of the user feedback loop is dependent on user participation.

LI-5: The system's performance and response time may vary based on user traffic and server load, potentially causing delays or temporary unavailability during peak usage periods.

C-1: The development and continuous improvement of the system requires a large and diverse dataset of cat images.

C-2: The deep learning and computer vision algorithms used for breed identification require significant computational resources

C-3: Technical constraints related to hardware and software compatibility, especially for image processing and deep learning algorithms, may limit the system's scalability and performance on certain devices or platforms.

1.5 Tools and Technologies

Table 2: Tools and Technologies for FelineConnect

Tools And Technologies	Tools	Version	Rationale
	Visual Studio Code	Latest	IDE
	MS Word	365	Documentation
	MS PowerPoint	365	Presentation
	Canva	Latest	Design & mockups
	Project Professional	2019	Design Work
	Git/GitHub	Latest	Version Control
Technology	Technology	Version	Rationale
	Python	Latest	Backend development
	HTML/CSS/JavaScript	HTML5/CSS3/ ES6	Front-end Development
	React JS	Latest	Front-end Development
	MySQL	Latest	DBMS
	Node JS	Latest	Backend Framework
	Express	Latest	Server-Side

	TensorFlow	Latest	Deep Learning Library
	Pytorch	Latest	Deep learning Framework
	Keras	Latest	Neural Network Library

1.6 Relevance to Course Modules

FelineConnect brings together knowledge from numerous courses taken during the Software Engineering curriculum. The cat breed detection tool is based on artificial intelligence concepts, which use machine learning models such as ResNet-50. Object-Oriented Programming ensures modular and scalable code for user administration, shopping, and analytics functions. The responsive user interface and RESTful APIs both make use of web development techniques. Database Systems knowledge aids in the design of secure and efficient data storage for users, products, and orders. Software Design and Architecture directed the system's layered structure, while Software Testing assures reliability. Human-Computer Interaction principles improve user experience, and E-commerce Systems inspired the online store module.

2 Problem Definition

2.1 Problem Statement

Currently, cat owners face challenges in accurately identifying the breed of their domestic cats, relying on manual recognition or external sources. This lack of a streamlined solution often leads to misidentification or uncertainty about the specific breed. Additionally, there's a need for a centralized platform that integrates cat breed recognition with an online store for cat food products. Users lack a comprehensive system where they can seamlessly upload images of their cats, receive accurate breed identification through a deep learning model, and then explore an online store for purchasing cat food items. The absence of such a system results in a fragmented user experience, where cat owners may struggle to find reliable information about their pets' breeds and encounter challenges in sourcing relevant products.

2.2 Problem Solution

The proposed cat breed recognition system with an integrated online store addresses the challenges faced by cat owners in accurately identifying their pets' breeds and sourcing relevant products. Leveraging deep learning models like ResNet-50, the system ensures precise breed identification from uploaded images, eliminating uncertainties. Users can easily upload cat images, receive instant breed identification results, and conveniently purchase cat food items tailored to their pets' needs. The platform's user-friendly interface, coupled with features like user authentication, profile management, and feedback collection, enhances user engagement and satisfaction. By providing a centralized solution, the platform streamlines the user experience, offering convenience and reliability in managing cat-related information and purchases. Continuous updates and improvements further enhance the platform's value, ensuring a comprehensive and effective solution for cat owners.

2.3 Objectives of the Proposed System

BO-1: Provide accurate and instant cat breed identification using deep learning and computer vision techniques.

BO-2: Create a user-friendly interface that allows users to easily upload cat images, receive prompt breed identification, and navigate through the online store.

BO-3: Implement an integrated online store allowing users to purchase cat food products.

BO-4: Establish a feedback loop for users to provide insights into breed recognition accuracy, enabling continuous model improvement.

BO-5: Optimize the online store for efficient transactions, including secure payment processing and order confirmation.

BO-6: Ensure the system is accessible to a diverse user base, considering different devices and user needs.

2.4 Scope

The scope of FelineConnect includes developing an AI-powered, web-based platform suited to the needs of cat owners and aficionados. The system includes a variety of features, including cat breed recognition using deep learning models such as ResNet-50, which allows users to upload photos and receive correct breed information with confidence scores. An online store is included, allowing users to explore, buy, and manage products like cat breeds and food. Users can restrict their searches by breed type, product category, or price range, and they will receive timely notifications about new items, special offers, and order updates.

Smooth transactions are made possible by a secure payment processing module that offers cancellation or modification options for orders. An AI chatbot helps consumers by responding to their questions, providing tailored advice, and increasing interaction. A strong admin panel facilitates effective user account administration, product inventory control, and feedback tracking for administrators. A statistical analytics module also ensures data-driven decision-making by offering insights into customer preferences, sales trends, and inventory levels.

Along with a scalable design that supports future expansion, the project also has an intuitive and responsive user interface for improved usage. Features unrelated to e-commerce or cat care are not included in the scope, guaranteeing a targeted solution for pet owners.

2.5 Modules

2.5.1 Module 1: User Authentication and Profile Management

FE-1: Users can create an account by providing personalized information.

FE-2: Registered user can access the account by providing information such as email and password.

FE-3: Registered user can add their names, profile photo, and bio information.

FE-4: Registered user can permanently delete his account.

FE-5: Registered user can change existing password.

FE-6: Registered user can use forgot password option to recover account access via email.

2.5.2 Module 2: Cat Breed Detection

- FE-1:** Users can upload images of their cats for breed identification.
- FE-2:** Split dataset into training and testing sets for model training.
- FE-3:** Integrate ResNet-50 architecture for accurate cat breed recognition.
- FE-4:** Transfer learning on a pre-trained ResNet-50 model.
- FE-5:** Fine-tune the model on a dataset of domestic cat breeds.
- FE-6:** Display detected cat breed along with confidence score.
- FE-7:** Allow users to provide feedback on the accuracy of breed recognition.

2.5.3 Module 3: Online Store

- FE-1:** Registered users can buy different cat breeds from the store.
- FE-2:** Registered users can buy cat food items from the store.
- FE-3:** Users can add Cat and food items to the shopping cart.
- FE-4:** Users can remove cat and food items from the cart.

2.5.4 Module 4: Search Filters and Notifications

- FE-1:** Implement a search bar for users to find cat breeds within the application.
- FE-2:** Allow users to refine search results using filters like breed type, food category, price range.
- FE-3:** Enable users to sort search results by relevance, price (ascending or descending).
- FE-4:** Allow users to search for specific cat food items.
- FE-5:** Send notifications to users about new products, or order status updates (e.g., order confirmation, delivery).
- FE-6:** Alert users about upcoming maintenance or updates to the app.

2.5.5 Module 5: Payment Processing

- FE-1:** Facilitate cart checkout.
- FE-2:** Integrate a secure payment gateway.
- FE-3:** Confirm orders and provide order details.
- FE-4:** Enable order cancellation or modification within a specific timeframe.
- FE-5:** Provide a review and rating system for purchased food items.

2.5.6 Module 6: AI Chatbot

- FE-1:** Allow users to submit queries or questions.
- FE-2:** Implement a chatbot for quick responses.
- FE-3:** Users can ask questions, provide commands, or engage in conversations with the chatbot.
- FE-4:** The chatbot would be able to generate relevant responses based on the user's input.
- FE-5:** The chatbot would be trained on custom data to provide user with their desired information.

2.5.7 Module 7: Admin Panel

- FE-1:** Allow admin to add, modify, or remove user accounts.
- FE-2:** View and manage cat food inventory in the store.
- FE-3:** Allow the admin to view and manage orders.
- FE-4:** Admin can change delivery status of product to pending or delivered successfully.
- FE-5:** View and manage user feedback on model predictions.

2.5.8 Module 8: Statistical Analytics

- FE-1:** Generate reports on the most popular cat breeds sold in the store.
- FE-2:** Track the performance of individual products in terms of sales, revenue, and customer ratings.
- FE-3:** Implement real-time data dashboards for quick insights into sales performance.
- FE-4:** Monitor inventory levels, track stockouts, and generate alerts for low stock or overstock situations.

3 Requirement Analysis

The functional and non-functional requirements necessary for creating the FelineConnect platform are described in this chapter. By defining the essential features and how they function to satisfy user and business needs, it provides information on the system's scope. We make sure that the software's design satisfies user expectations, technical limitations, and business objectives by conducting a thorough examination. The section examines the system's non-functional needs, which describe quality features like performance, scalability, and security, as well as its functional requirements, which outline what the software must accomplish to achieve its goals. In order to guarantee a thorough grasp of the solution's requirements and deliverables, it also specifies user roles and interactions.

3.1 User classes and characteristics

Table 3: User classes and characteristics

User Class	Description
Admin	Admins in FelineConnect have administrative privileges and responsibilities. They can manage user accounts, including Users. Admins ensure smooth operation of the platform, address user issues, manage payments, and handle feedback from users, maintaining the platform's functionality.
User	Cat owner are central users of FelineConnect, utilizing the platform to identify their cat's breed accurately and purchase relevant cat food products. They can upload images of their cats, receive instant breed identification results, explore the online store, add items to the cart, make secure payments, and provide feedback on their experiences, contributing to continuous improvement.

3.2 Requirement Identifying Technique

For the FelineConnect project, we utilized Use Case Analysis as the primary requirement identification technique. This technique is particularly effective for interactive, end-user-focused applications like FelineConnect, as it emphasizes the interactions between users and the system.

Additionally, **interviews** with potential users and stakeholders were conducted to gather insights into their expectations, challenges, and desired features. This approach ensures that the identified requirements align with real-world needs and address existing gaps.

3.2.1 Use Case Diagram

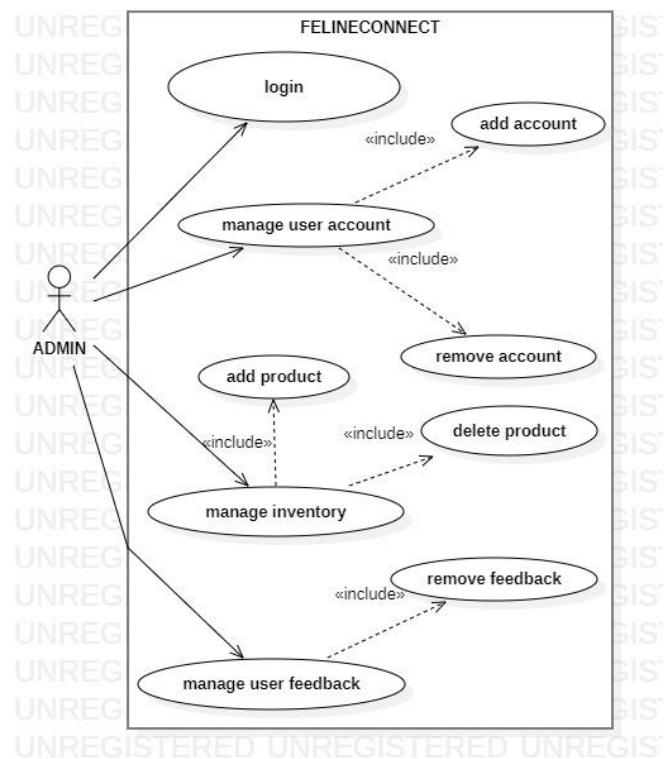


Figure 1: Use Case Diagram for Admin

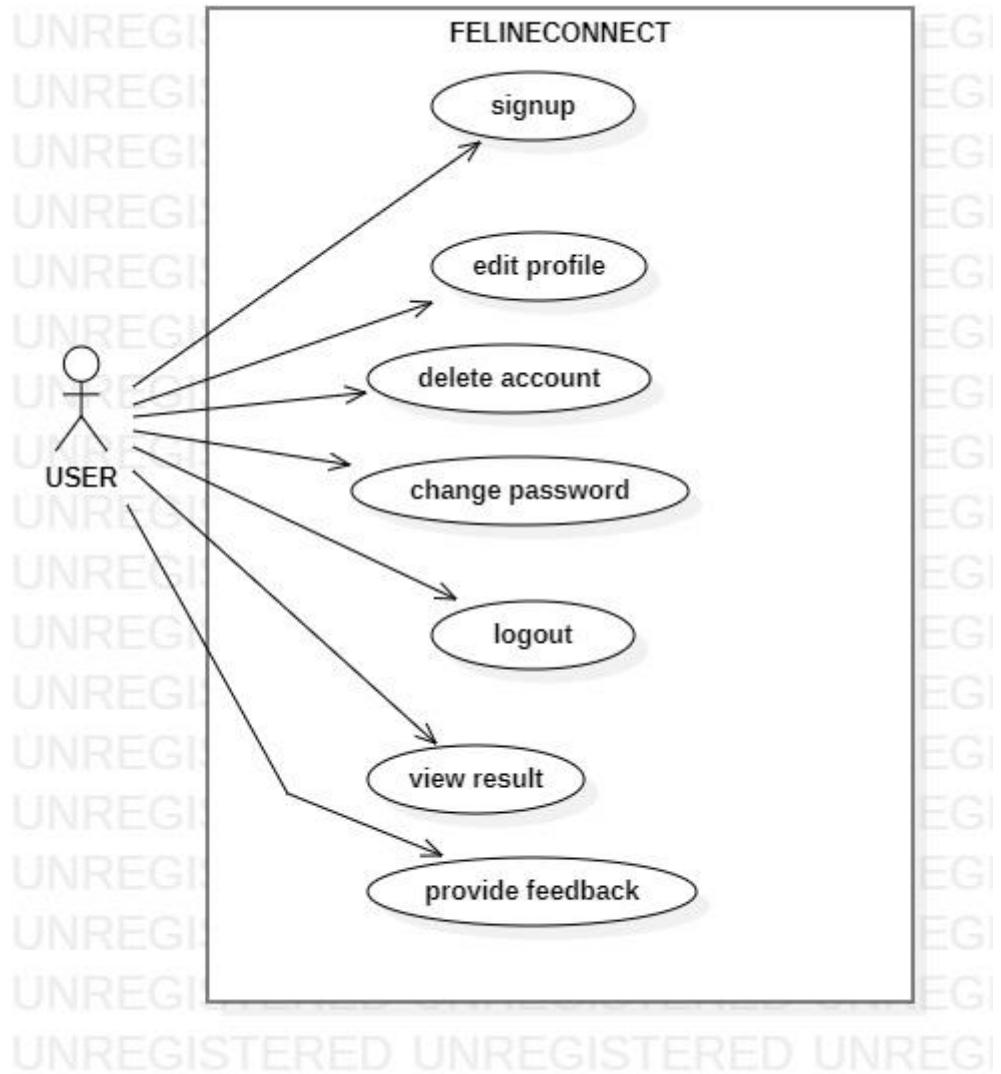


Figure 2: Use Case Diagram for User

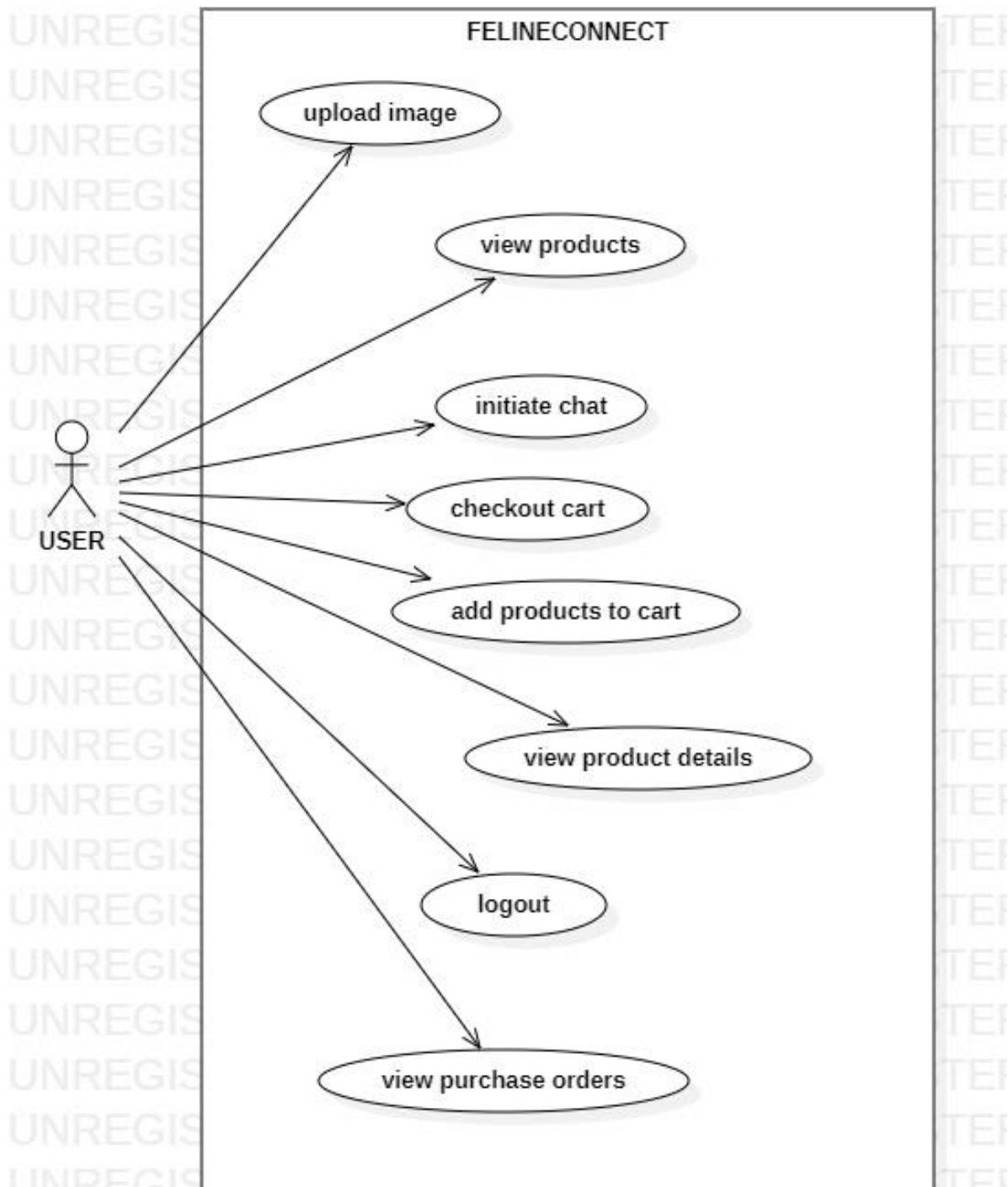


Figure 2.1: Use Case Diagram for User (Continue)

3.2.2 Detail Use Case:

3.2.2.1 Module 1: User Authentication and Profile Management

3.2.2.1.1 UC-1: User Signup

Use Case ID:	UC-1
Use Case Name:	User Signup
Actors:	User
Description:	The user will enter their full name, email, password, city, address and phone number during the signup process.
Trigger:	The User presses the Register Button.
Preconditions:	PRE-1. The user must access the signup page. PRE-2. The user should not be logged in.
Postconditions:	POST-1. A Success Message will be displayed, indicating that the user has successfully signed up. POST-2. The user will be navigated to the homepage.
Normal Flow:	<ol style="list-style-type: none"> 1. System displays the signup form to be filled by the user. 2. Enter the full name. 3. Enter a valid email address. 4. Enter a secure password. 5. Enter City. 6. Enter Address. 7. Enter valid Phone number. 8. Press the Register button. 9. A Success Message will be displayed, and the user is taken to the homepage.
Alternative Flows:	<ol style="list-style-type: none"> 1. The User enters the first name. 2. The User enters the last name. 3. The User enters a valid password. 4. The User presses the register button. 5. An Error Message will be shown, indicating a lack of network connectivity.
Exceptions:	<ol style="list-style-type: none"> 1. The User enters incorrect or incomplete first name, last name, or email. 2. The email provided is already registered. 3. The password is not secure enough.
Business Rules:	N/A
Assumptions:	<ol style="list-style-type: none"> 1. The User should understand basic English language to fill out the signup form. 2. An active internet connection is required to complete the signup process.

3.2.2.1.2 UC-2: Login

Use Case ID:	UC-2
Use Case Name:	Login
Actors:	User
Description:	The user will enter their email, password to login to the system.
Trigger:	The Client presses the Login Button.
Preconditions:	N/A.
Post conditions:	POST-1. User will be logged in to the system POST-2. System redirects the user to homepage
Normal Flow:	<ol style="list-style-type: none"> 1. System displays the login form. 2. Enter the email. 3. Enter the password. 4. Click the log in button. 5. User will be logged in to the system. 6. Success message will be displayed. 7. The user will be redirected to the homepage.
Alternative Flows:	<ol style="list-style-type: none"> 1. Enter the email. 2. Click login button. 3. Error message will be displayed.
Exceptions:	The user enters incorrect email or password.
Business Rules:	BR-1: all the mandatory fields are required to be filled with correct data.
Assumptions:	The user has already signed up

3.2.2.1.3 UC-3: Signup with google

Use Case ID:	UC-3
Use Case Name:	Signup with Google
Actors:	User
Description:	User can login to the system using third party authentication with Google
Trigger:	Users presses the Sign up with Google icon button.
Preconditions:	N/A
Post conditions:	POST-1. Users will be logged in/ registered into the system. POST-2. The user will be redirected to the homepage.
Normal Flow:	<ol style="list-style-type: none"> 1. The system displays the signup form 2. User clicks the Sign up with Google icon button. 3. The system redirects the user to Google's authentication page. 4. User enters their Google credentials. 5. User is redirected back to the system. 6. The system retrieves user information from Google. 7. The user is logged in/registered, and the system may prompt the user to complete their profile

Alternative Flows:	<ol style="list-style-type: none"> 1. User has already signed into Google on browser 2. User will click Sign up with Google icon button on the system. 3. User will select his account 4. User will be logged in/ registered
Exceptions:	The user enters an incorrect email or password.
Business Rules:	N/A
Assumptions:	<ol style="list-style-type: none"> 1. The user has a valid Google account. 2. The user is willing to grant access to their Google account. 3. The user is on the signup page of the system. 4. The user is not already logged into the system.

3.2.2.1.4 UC-4: Update user credentials

Use Case ID:	UC-4
Use Case Name:	Update User Credentials
Actors:	User.
Description:	The user be able to update their profiles.
Trigger:	The user presses the update button in the user profile
Preconditions:	PRE-1. The User has logged in to the system.
Post conditions:	POST-1. The User will have successfully updated their credentials.
Normal Flow:	<ol style="list-style-type: none"> 1. User is logged into the system. 2. User navigates to settings. 3. User selects "Personal Information." 4. User is presented with the option to change their: <ul style="list-style-type: none"> • Name • Address • Phone Number • Password 5. User makes the desired changes. 6. User clicks "Save Changes." 7. The credentials are updated in the system.
Alternative Flows:	<ol style="list-style-type: none"> 1. User is logged into the system. 2. User navigates to settings. 3. User selects "Personal Information." 4. User is presented with the option to change their: <ul style="list-style-type: none"> 5. Name 6. Address 7. Phone Number 8. Password 9. User makes the desired changes. 10. User clicks "Save Changes." 11. Error message is generated
Exceptions:	The user enters any incorrect data.
Business Rules:	N/A
Assumptions:	<ol style="list-style-type: none"> 1. The user has logged in. 2. Internet connect is available

3.2.2.1.5 UC-5: Forget password

Use Case ID:	UC-5
Use Case Name:	Forgot Password
Actors:	User
Description:	If the user forgets their password, then they can reset the password to a new one
Trigger:	The user presses the forgot password button
Preconditions:	PRE-1. User account already exist.
Post conditions:	POST-1. Password of the user account will be successfully reset.
Normal Flow:	<ol style="list-style-type: none"> 1. The user will click Forgot Password button 2. The user will provide the email 3. The user will get a recovery link on the email 4. The user will click on the recovery email 5. The user will provide a new password 6. The user will provide confirmation of new password 7. The user will press Save Changes button.
Alternative Flows:	N/A

3.2.2.1.6 UC-6: Delete user account

Use Case ID:	UC-6
Use Case Name:	Delete User Account
Actors:	User
Description:	User can Delete their account.
Trigger:	The user presses the Delete Account button
Preconditions:	PRE-1. The user has logged in to the account
Post conditions:	POST-1. User account will be successfully deleted POST-2. The user will be redirected to the homepage
Normal Flow:	<ol style="list-style-type: none"> 1. The user will login the system 2. The user will go to Settings 3. The user will select Delete Account Button 4. A box will appear the user will enter the account password. 5. User will click on Delete button. 6. A confirmation Dialogue Box will appear. 7. User will select Confirm button. 8. The user account has been deleted. 9. User will be redirected to the signup page
Alternative Flows :	<ol style="list-style-type: none"> 1. The user will login the system 2. The user will go to their profile 3. The user will select Delete Account Button 4. A box will appear the user will enter incorrect account password. 5. User will click on Delete button. 6. A message box will appear displaying incorrect password. 7. The account will not be deleted
Exceptions:	

Business Rules:	N/A
Assumptions:	The user has logged in to the system

3.2.2.1.7 UC-7: Sign-out

Use Case ID:	UC-7
Use Case Name:	Sign out
Actors:	User
Description:	User can log out of the system
Trigger:	Users presses the Log out button
Preconditions:	PRE-1. User is logged in to the system
Post conditions:	POST-1. Users will be logged out of the system
Normal Flow:	<ol style="list-style-type: none"> 1. User clicks the logout button 2. User is logged out 3. User is redirected to the homepage
Alternative Flows:	N/A
Exceptions:	
Business Rules:	N/A
Assumptions:	<ol style="list-style-type: none"> 1. The user has already logged in 2. The user has working internet connection

3.2.2.2 Module 2: Cat Breed Detection

3.2.2.2.1 UC-8: Upload Cat Images for Breed Recognition

Use Case ID	UC-8
Use Case Name	Upload Cat Images for Breed Recognition
Actors	User
Description	Users can upload images of their cats to FelineConnect for breed recognition.
Trigger	User wants to upload cat images for breed identification.
Preconditions	User is logged into their FelineConnect account.
Postconditions	Cat images are successfully uploaded for breed recognition.

Normal Flow	<ol style="list-style-type: none"> 1. User navigates to the "Image Submission" section. 2. User selects the option to upload cat images. 3. User chooses one or more images from their device. 4. User confirms the image selection and submission. System processes and stores the uploaded images.
Exceptions	Invalid image format, size limit exceeded, or incomplete submission.
Business Rules	N/A
Assumptions	Users have access to their device's storage for image selection.

3.2.2.2.2 Integrate ResNet-50 for Cat Breed Recognition

Use Case ID	UC-9
Use Case Name	Integrate ResNet-50 for Cat Breed Recognition
Actors	System
Description	System integrates ResNet-50 architecture for accurate cat breed recognition.
Trigger	System initiates integration of ResNet-50 for breed recognition.
Preconditions	ResNet-50 architecture is available for integration.
Postconditions	ResNet-50 is successfully integrated for breed recognition.
Normal Flow	<ol style="list-style-type: none"> 1. System prepares for integrating ResNet-50 architecture. 2. System configures the necessary environment for ResNet-50 integration. 3. System downloads the pre-trained ResNet-50 model. 4. System integrates ResNet-50 architecture into the FelineConnect system.
Exceptions	None
Business Rules	ResNet-50 integration enhances accuracy in breed recognition.
Assumptions	Sufficient computational resources are available for model integration and training.

3.2.2.2.3 Transfer Learning with ResNet-50 Model

Use Case ID	UC-10
Use Case Name	Perform Transfer Learning with ResNet-50 Model
Actors	System
Description	System performs transfer learning on a pre-trained ResNet-50 model.
Trigger	Pre-trained ResNet-50 model is available for transfer learning.
Preconditions	Pre-trained ResNet-50 model is downloaded and ready for transfer learning.
Postconditions	Transfer learning is successfully applied to the ResNet-50 model.
Normal Flow	<ol style="list-style-type: none"> 1. System prepares the pre-trained ResNet-50 model for transfer learning. 2. System defines the transfer learning strategy and objectives. 3. System executes transfer learning on the pre-trained model using a dataset of domestic cat breeds.
Exceptions	None
Business Rules	Transfer learning optimizes the ResNet-50 model for cat breed recognition.
Assumptions	Sufficient data and computational resources are available for transfer learning.

3.2.2.2.4 Fine-tune ResNet-50 Model on Cat Breed Dataset

Use Case ID	UC-11
Use Case Name	Perform Fine-tuning on ResNet-50 Model with Cat Breed Dataset
Actors	System
Description	System fine-tunes the ResNet-50 model using a dataset of domestic cat breeds.
Trigger	System initiates fine-tuning of ResNet-50 model on a specific dataset.
Preconditions	ResNet-50 architecture is integrated and ready for fine-tuning.
Postconditions	ResNet-50 model is fine-tuned for improved accuracy in cat breed recognition.
Normal Flow	<ol style="list-style-type: none"> 1. System prepares the dataset of domestic cat breeds for fine-tuning. 2. System defines the fine-tuning parameters and objectives. 3. System executes fine-tuning of the ResNet-50 model using the prepared dataset.
Exceptions	None
Business Rules	Fine-tuning improves the ResNet-50 model's accuracy and performance on cat breed recognition tasks.
Assumptions	Sufficient computational resources and quality dataset are available for fine-tuning.

3.2.2.2.5 Display Detected Cat Breed with Confidence Score

Use Case ID	UC-14
Use Case Name	Display Detected Cat Breed with Confidence Score
Actors	System
Description	System displays the detected cat breed along with the confidence score.
Trigger	Cat breed recognition is successfully completed.
Preconditions	Cat breed recognition algorithm produces results.
Postconditions	Detected cat breed and confidence score are displayed to the user.
Normal Flow	<ol style="list-style-type: none"> 1. System receives the results from the breed recognition algorithm. 2. System formats and presents the detected cat breed and confidence score on the user interface.
Exceptions	None
Business Rules	Displaying confidence scores helps users assess the accuracy of breed recognition.
Assumptions	Breed recognition algorithm provides accurate results with confidence scores.

3.2.2.2.6 Provide Visual Representation of Key Features

Use Case ID	UC-15
Use Case Name	Provide Visual Representation of Key Features
Actors	System
Description	System provides a visual representation of key features contributing to breed identification.
Trigger	User requests detailed information about breed identification.
Preconditions	Detailed feature analysis data is available.
Postconditions	Visual representation of key features is displayed to the user.

Normal Flow	1. User requests detailed information about the detected cat breed. 2. System retrieves and processes key feature data. 3. System generates a visual representation (e.g., heatmap, feature map) of key features contributing to breed identification.
Exceptions	Key feature data not available or cannot be processed.
Business Rules	Visual representations aid users in understanding the basis of breed identification.
Assumptions	Sufficient data and algorithms are available for generating visual representations of key features.

3.2.2.2.7 Allow Users to Provide Feedback on Accuracy

Use Case ID	UC-16
UseCase Name	Allow Users to Provide Feedback on Accuracy
Actors	User
Description	System allows users to provide feedback on the accuracy of breed recognition.
Trigger	User interacts with the breed detection result interface.
Preconditions	Breed detection results are displayed to the user.
Postconditions	User feedback on accuracy is recorded and processed.
Normal Flow	1. User views the detected cat breed and confidence score. 2. System provides an option for the user to provide feedback on the accuracy of the detected breed. 3. User submits feedback (e.g., rating, comment) through the interface. 4. System records and processes the user feedback for analysis.
Exceptions	User does not provide feedback.
Business Rules	User feedback helps improve the accuracy of breed recognition algorithms.
Assumptions	User interface includes an interactive feedback mechanism for accuracy assessment.

3.2.2.3 Module 3: Online Store

3.2.2.3.1 Buy cat food items

Use Case ID:	UC-17
Use Case Name:	Buy Cat Food Items
Actors:	Primary Actor: User Secondary Actor: Online Store System
Description:	Users can purchase cat food items from the online store.
Trigger:	User indicates the desire to buy cat food items.
Preconditions:	PRE-1. User is logged into the online store. PRE-2. User has selected cat food items to purchase.
Postconditions:	POST-1. Purchased cat food items are added to the user's shopping cart. POST-2. User can proceed to checkout or continue shopping.
Normal Flow:	1. User browses cat food items in the online store. 2. User adds selected cat food items to the shopping cart. 3. User reviews the shopping cart and proceeds to checkout. 4. Online store system confirms the purchase and updates the shopping cart.

	5. User completes the payment process.
Alternative Flows:	Ex-1. If the selected cat food items are out of stock, the online store system notifies the user and removes them from the cart. Ex-2. If the payment process fails, the user is prompted to retry or use a different payment method.
Business Rules:	BR-1. Cat food items must be in stock to be added to the shopping cart. BR-2. Payment must be successfully processed before completing the purchase.
Assumptions:	Assume that users have registered accounts and payment methods set up in the online store system.

3.2.2.3.2 Add food items to the shopping cart

Use Case ID:	UC-18
Use Case Name:	Add Items to Shopping Cart
Actors:	Primary Actor: User Secondary Actor: Online Store System
Description:	Users can add cat food items to the shopping cart for purchase.
Trigger:	User selects cat food items and chooses to add them to the cart.
Preconditions:	PRE-1. User is logged into the online store. PRE-2. Cat food items are available for purchase.
Postconditions:	POST-1. Selected cat food items are added to the shopping cart. POST-2. User can proceed to checkout or continue shopping.
Normal Flow:	1. User selects cat food items to add to the shopping cart. 2. Online store system adds selected items to the cart. 3. User reviews the shopping cart.
Alternative Flows:	None
Exceptions:	Ex-1. If the selected cat food items are out of stock, the online store system notifies the user and prevents adding them to the cart.
Business Rules:	BR-1. Cat food items must be in stock to be added to the shopping cart.

3.2.2.3.3 Remove food items from the cart

Use Case ID:	UC-19
Use Case Name:	Remove Items from Shopping Cart
Actors:	Primary Actor: User Secondary Actor: Online Store System
Description:	Users can remove cat food items from the shopping cart before completing the purchase.
Trigger:	User selects cat food items in the cart and chooses to remove them.
Preconditions:	PRE-1. User is logged into the online store. PRE-2. Cat food items are in the shopping cart.
Postconditions:	POST-1. Selected cat food items are removed from the shopping cart. POST-2. User can continue shopping or proceed to checkout.
Normal Flow:	1. User views the shopping cart with selected cat food items. 2. User selects items to remove from the cart. 3. Online store system updates the cart accordingly.
Alternative Flows:	None
Exceptions:	None

Business Rules:	BR-1. User can only remove items that are currently in the shopping cart.
------------------------	---

3.2.2.3.4 Provide a review for purchased food items

Use Case ID:	UC-20
Use Case Name:	Review and Rate Purchased Items
Actors:	Primary Actor: User Secondary Actor: Online Store System
Description:	Users can review and rate cat food items after purchasing them from the online store.
Trigger:	User completes the purchase of cat food items.
Preconditions:	PRE-1. User is logged into the online store. PRE-2. User has purchased cat food items.
Postconditions:	POST-1. User submits a review and rating for the purchased items. POST-2. Online store system records the review and rating for the items.
Normal Flow:	1. User completes the purchase of cat food items. 2. Online store system prompts user to review and rate the purchased items. 3. User provides a review and rating.
Alternative Flows:	None
Exceptions:	None
Business Rules:	BR-1. Users can only review and rate items they have purchased.

3.2.2.4 Module 4: Search Filters and Notifications

3.2.2.4.1 FE-1: Implement a Search Bar for Cat Breeds

Use Case ID	UC-21
Use Case Name	Implement Search Bar for Cat Breeds
Actors	User
Description	Users can use a search bar to find specific cat breeds within the application.
Trigger	User wants to search for a particular cat breed.
Preconditions	User is logged into their FelineConnect account.
Postconditions	Search results for the specified breed are displayed.
Normal Flow	1. User navigates to the search bar section.2. User types the name of the cat breed into the search bar.3. User submits the search query.4. System processes the query and retrieves matching results.5. Search results are displayed to the user.
Exceptions	No results found for the search query.
Business Rules	Search should be responsive and provide relevant results quickly.
Assumptions	The system has a comprehensive database of cat breeds.

3.2.2.4.2 Refine Search Results Using Filters

Use Case ID	UC-22
Use Case Name	Refine Search Results Using Filters
Actors	User
Description	Users can refine their search results using various filters like breed type, food category, and price range.
Trigger	User wants to narrow down search results.
Preconditions	User has initiated a search query.
Postconditions	Filtered search results are displayed.
Normal Flow	1. User views the search results page.2. User selects desired filters (breed type, food category, price range).3. User applies the selected filters.4. System processes the filter criteria.5. Filtered results are displayed to the user.
Exceptions	No results match the applied filters.
Business Rules	Filters should be clear and easy to use.
Assumptions	The system provides a variety of relevant filters for user selection.

3.2.2.4.3 Sort Search Results by Relevance or Price

Use Case ID	UC-23
Use Case Name	Sort Search Results by Relevance or Price
Actors	User
Description	Users can sort search results by relevance or price (ascending or descending).
Trigger	User wants to organize search results for easier viewing.
Preconditions	User has search results displayed.
Postconditions	Sorted search results are displayed.
Normal Flow	1. User views the search results page.2. User selects the sorting criteria (relevance, price ascending, price descending).3. User applies the sorting criteria.4. System processes the sorting request.5. Sorted results are displayed to the user.
Exceptions	None
Business Rules	Sorting options should be clear and intuitive.
Assumptions	The system has capabilities to sort data efficiently.

3.2.2.4.4 Search for Specific Cat Food Items

Use Case ID	UC-24
Use Case Name	Search for Specific Cat Food Items
Actors	User
Description	Users can search for specific cat food items using the search bar.
Trigger	User wants to find a specific cat food item.
Preconditions	User is logged into their FelineConnect account.
Postconditions	Search results for the specified cat food item are displayed.

Normal Flow	1. User navigates to the search bar section. 2. User types the name of the cat food item into the search bar. 3. User submits the search query. 4. System processes the query and retrieves matching results. 5. Search results are displayed to the user.
Exceptions	No results found for the search query.
Business Rules	Search should be responsive and provide relevant results quickly.
Assumptions	The system has a comprehensive database of cat food items.

3.2.2.4.5 Send Notifications About New Products or Order Status

Use Case ID	UC-25
Use Case Name	Send Notifications About New Products or Order Status
Actors	System, User
Description	The system sends notifications to users about new products or updates on their order status.
Trigger	A new product is added or there is an update in order status.
Preconditions	User has enabled notifications.
Postconditions	User receives a notification.
Normal Flow	1. A new product is added to the store or an order status is updated. 2. System generates a notification. 3. System sends the notification to the user's device. 4. User receives and views the notification.
Exceptions	Notification delivery failure.
Business Rules	Notifications should be timely and relevant.
Assumptions	Users have notifications enabled and their devices are connected to the internet.

3.2.2.4.6 Alert Users About Maintenance or App Updates

Use Case ID	UC-26
Use Case Name	Alert Users About Maintenance or App Updates
Actors	System, User
Description	The system alerts users about upcoming maintenance or updates to the app.
Trigger	Scheduled maintenance or new app updates are planned.
Preconditions	User has enabled notifications.
Postconditions	User receives an alert notification.
Normal Flow	1. Maintenance or app update is scheduled. 2. System generates an alert notification. 3. System sends the alert notification to the user's device. 4. User receives and views the alert notification.
Exceptions	Notification delivery failure.
Business Rules	Alerts should be sent with sufficient notice.
Assumptions	Users have notifications enabled and their devices are connected to the internet.

3.2.2.4.7 Save User Preferences for Notifications

Use Case ID	UC-27
-------------	-------

Use Case Name	Save User Preferences for Notifications
Actors	User, System
Description	Users can save their preferences for receiving different types of notifications.
Trigger	User wants to customize their notification settings.
Preconditions	User is logged into their FelineConnect account.
Postconditions	User preferences for notifications are saved.
Normal Flow	1. User navigates to the notification settings section.2. User selects their preferred notification types (e.g., product updates, order status, maintenance alerts).3. User saves the selected preferences.4. System records the preferences in the user's profile.5. System sends notifications based on saved preferences.
Exceptions	Preferences not saved due to system error.
Business Rules	User preferences should be easy to update and save.
Assumptions	The system can handle and store individual user preferences.

3.2.2.4.8 View Notification History

Use Case ID	UC-28
Use Case Name	View Notification History
Actors	User
Description	Users can view their notification history within the application.
Trigger	User wants to review past notifications.
Preconditions	User is logged into their FelineConnect account.
Postconditions	User views a list of past notifications.
Normal Flow	1. User navigates to the notification history section.2. System retrieves past notifications from the database.3. Past notifications are displayed to the user.<4. User reviews the notification history.
Exceptions	No past notifications found.
Business Rules	Notification history should be displayed in a user-friendly manner.
Assumptions	The system stores and maintains a log of all notifications.

3.2.2.4.9 Implement Push Notifications

Use Case ID	UC-29
Use Case Name	Implement Push Notifications
Actors	System, User
Description	The system sends push notifications to users' devices about new updates or alerts.
Trigger	A new update or alert is generated by the system.
Preconditions	User has enabled push notifications.
Postconditions	User receives a push notification.
Normal Flow	1. A new update or alert is generated by the system.2. System creates a push notification. 3. System sends the push notification to the user's device.4. User receives and views the push notification.
Exceptions	Push notification delivery failure.
Business Rules	Push notifications should be immediate and relevant.
Assumptions	Users have push notifications enabled and their devices are connected to the internet.

3.2.2.4.10 Provide In-App Notifications

Use Case ID	UC-30
Use Case Name	Provide In-App Notifications
Actors	System, User
Description	The system provides in-app notifications for users about updates, new products, and alerts.
Trigger	A new update or alert is generated by the system.
Preconditions	User is logged into their FelineConnect account.
Postconditions	User sees an in-app notification.
Normal Flow	1. A new update or alert is generated by the system.2. System creates an in-app notification.3. System displays the in-app notification to the user.4. User sees and interacts with the in-app notification.
Exceptions	None
Business Rules	In-app notifications should be prominent and actionable.
Assumptions	Users regularly interact with the app and can see notifications.

3.2.2.5 Module 5: Payment Processing

3.2.2.5.1 Facilitate cart checkout

Use Case ID:	UC-30
Use Case Name:	Cart Checkout
Actors:	Primary Actor: User Secondary Actor: Payment System
Description:	Users can proceed with the checkout process to complete their purchase.
Trigger:	User indicates the desire to checkout the items in the shopping cart.
Preconditions:	PRE-1. User is logged into the online store. PRE-2. User has items in the shopping cart.
Postconditions:	POST-1. User completes the checkout process and receives confirmation of the order.
Normal Flow:	1. User views the items in the shopping cart and selects checkout. 2. Online store system prompts user for shipping and payment information. 3. User enters shipping details and selects payment method. 4. Online store system processes payment and confirms the order.
Alternative Flows:	None
Exceptions:	Ex-1. If payment authorization fails, user is prompted to retry or use a different payment method.
Business Rules:	BR-1. User must provide valid shipping and payment information to complete checkout.

3.2.2.5.2 Integrate a secure payment gateway

Use Case ID:	UC-31
Use Case Name:	Secure Payment Integration
Actors:	Primary Actor: System Secondary Actor: Payment Gateway

Description:	Online store system integrates with a secure payment gateway for processing payments.
Trigger:	User initiates payment during checkout process.
Preconditions:	PRE-1. User has completed the checkout process.
Postconditions:	POST-1. Payment gateway confirms payment authorization.
Normal Flow:	<ol style="list-style-type: none"> 1. Online store system sends payment request to the payment gateway. 2. Payment gateway processes payment authorization. 3. Payment gateway sends confirmation to online store system.
Alternative Flows:	None
Exceptions:	Ex-1. If payment authorization fails, online store system notifies user and prompts for alternative payment method.
Business Rules:	BR-1. Payment gateway must be integrated securely to ensure transaction security.

3.2.2.5.3 Confirm orders and provide order details

Use Case ID:	UC-32
Use Case Name:	Order Confirmation and Details
Actors:	Primary Actor: System Secondary Actor: User
Description:	Online store system confirms user orders and provides order details.
Trigger:	Payment authorization is successful.
Preconditions:	PRE-1. User has completed the checkout process. PRE-2. Payment authorization is successful.
Postconditions:	POST-1. User receives confirmation of the order and order details.
Normal Flow:	<ol style="list-style-type: none"> 1. Online store system generates order confirmation with order details. 2. User receives order confirmation via email or on the website.
Alternative Flows:	None
Exceptions:	None
Business Rules:	BR-1. Order confirmation must include detailed order information such as items purchased, shipping details, and payment summary.

3.2.2.6 Module 6: AI Chatbot

3.2.2.6.1 Initiate Chat Session

Use Case ID:	UC-33
Use Case Name:	Initiate Chat
Actors:	User
Trigger:	User initiates a conversation with the AI chatbot.
Preconditions:	PRE-1. The user must be logged in to the system
Post conditions:	None

Normal Flow:	1. User initiates a conversation with the AI chatbot 2. User asks questions, provides commands, or engages in conversation. 3. The system processes the users input and generates response
Alternative Flows	N/A
Exceptions:	N/A
Business Rules:	BR-1: The chatbot should provide accurate, up-to-date information from authoritative sources. BR-2: User interactions with the chatbot are logged and may be used for analysis and system improvement
Assumptions:	1. The User must understand basic English language. 2. Internet connection is available.

3.2.2.6.2 Close a chat session

Use Case ID:	UC-34
Use Case Name:	Close a chat session
Actors:	User
Trigger:	User wants to end the chat session with the chatbot
Preconditions:	PRE-1. User is currently engaged in a chat session
Post conditions:	POST-1. Chat session is successfully closed
Normal Flow:	1. User initiates the request to close the chat session by expressing their intent to end the conversation. 2. The chatbot ask for confirmation of the closing of chat session 3. The User confirm to close the chat session 4. The chatbot closes the chat session.
Alternative Flows	
Exceptions:	N/A
Business Rules:	BR-1: The chatbot should provide accurate, up-to-date information from authoritative sources. BR-2: User interactions with the chatbot are logged and may be used for analysis and system improvement
Assumptions:	3. The User must understand basic English language. 4. Internet connection is available.

3.2.2.7 Module 7: Admin Panel

3.2.2.7.1 View Dashboard

Use Case ID:	UC-35
Use Case Name:	View dashboard
Actors:	Admin

Description:	The admin will be able to view dashboard
Trigger:	The user pressed Dashboard option from sidebar.
Preconditions:	N/A
Post conditions:	POST-1. Admin successfully views the dashboard
Normal Flow:	<ol style="list-style-type: none"> 1. Admin will Login. 2. Admin will navigate to dashboard from the sidebar. 3. The system will display the dashboard to the admin.
Alternative Flows	N/A
Exceptions:	N/A
Business Rules:	N/A
Assumptions:	<ol style="list-style-type: none"> 1. The admin should understand basic English language to fill out the signup form. 2. The admin is signed up. 3. The admin should have an active internet connection.

3.2.2.7.2 Add another admin

Use Case ID:	UC-36
Use Case Name:	Add another Admin
Actors:	Admin
Description:	The admin will be able to add another admin
Trigger:	The admin presses the Register Button.
Preconditions:	N/A
Post conditions:	<p>POST-1. A Success Message will be shown which means admin is successfully signed in and they will be navigated to the homepage.</p> <p>POST-2. An Error Message will be shown if there is something wrong in the fields.</p>
Normal Flow:	<ol style="list-style-type: none"> 1. Admin will Login. 2. Admin will navigate to add Another Admin button from the sidebar. 3. The admin will enter the first name 4. The admin will enter the last name 5. The admin will enter the email 6. The admin will enter the password. 7. Press the register button 8. A Success Message will be shown.
Alternative Flows	<ol style="list-style-type: none"> 1. The admin will enter the first name 2. The admin will enter the last name 3. The admin will enter his password. 4. The admin will press the register button 5. An Error Message will be shown.
Exceptions:	The admin enters incorrect first name, last name, email.
Business Rules:	N/A

Assumptions:	<ol style="list-style-type: none"> 1. The admin should understand basic English language to fill out the signup form 2. The admin should have an active internet connection.
---------------------	--

3.2.2.7.3 Admin login

Use Case ID:	UC-37
Use Case Name:	Admin Login
Actors:	Admin
Description:	The admin will be required to enter their email, password to login to the system.
Trigger:	The admin presses the Login Button.
Preconditions:	N/A
Post conditions:	POST-1. Admin will be logged in to the system
Normal Flow:	<ol style="list-style-type: none"> 1. System displays the login form to be filled by the user. 2. Enter the email. 3. Enter the password. 4. Press the register button. 5. A success message is displayed. 6. User will be logged in to the system 7. The admin is redirected to the dashboard
Alternative Flows	N/A
Exceptions:	The admin enters incorrect email or password.
Business Rules:	N/A
Assumptions:	The admin has already signed up

3.2.2.7.4 View specific user information

Use Case ID:	UC-38
Use Case Name:	View Specific User Information
Actors:	Admin
Description:	The admin will be able to view the user information like name, emails, feedbacks etc.
Trigger:	The admin presses the View Details button
Preconditions:	N/A
Post conditions:	POST-1. Admin will be able to view the specific user information
Normal Flow:	<ol style="list-style-type: none"> 1. Admin logs in 2. Admin selects the View Users menu from the navbar. 3. List of all the users will be displayed. 4. The admin will click on a specific user details button. 5. All the users are displayed.

Alternative Flows	N/A
Exceptions:	No user account exists
Business Rules:	N/A
Assumptions:	The admin has already logged in

3.2.2.7.5 View all users list

Use Case ID:	UC-39
Use Case Name:	View All Users List
Actors:	Admin
Description:	The admin will be able to view the list of all users in the system.
Trigger:	The admin presses the View Users button from the side menu bar.
Preconditions:	N/A
Post conditions:	POST-1. Admin will be able to view the list of all users.
Normal Flow:	<ol style="list-style-type: none"> 1. Admin logs in 2. Admin selects the View Users menu from the navbar. 3. List of all the users will be displayed.
Alternative Flows	N/A
Exceptions:	N/A
Business Rules:	N/A
Assumptions:	<ol style="list-style-type: none"> 1. The admin has already logged in. 2. Some users already exist.

3.2.2.7.6 Delete user account

Use Case ID:	UC-40
Use Case Name:	Delete a user account
Actors:	Admin
Description:	The admin will be able to delete a user account.
Trigger:	The admin presses the View Users button from the side menu bar.
Preconditions:	N/A
Post conditions:	POST-1. Admin will be able to delete a specific user account.

Normal Flow:	<ol style="list-style-type: none"> 1. Admin logs in 2. Admin selects the View Users menu from the navbar. 3. List of all the users will be displayed. 4. The admin clicks on view button. 5. The page containing the information of the user will be displayed. 6. The admin clicks on the delete user account. 7. A confirmation Dialogue Box will appear. 8. admin will select Confirm button. 9. A success message will be displayed.
Alternative Flows	N/A
Exceptions:	N/A
Business Rules:	N/A
Assumptions:	<ol style="list-style-type: none"> 1. The admin has already logged in. 2. Some users already exist.

3.2.2.7.7 View all feedbacks list

Use Case ID:	UC-41
Use Case Name:	View All Feedback List
Actors:	Admin
Description:	The admin will be able to view the list of all user feedbacks.
Trigger:	The admin presses the View Feedback button Side menu.
Preconditions:	N/A
Post conditions:	POST-1. Admin will be able to view the list of all users feedbacks.
Normal Flow:	<ol style="list-style-type: none"> 1. Admin logs in 2. Admin selects the View Feedbacks menu from the side menu. 3. List of all the feedbacks will be displayed.
Alternative Flows	N/A
Exceptions:	N/A
Business Rules:	N/A
Assumptions:	<ol style="list-style-type: none"> 1. The admin has already logged in. 2. Some feedbacks already exist.

3.2.2.7.8 View User Feedback description

Use Case ID:	UC-42
Use Case Name:	View User Feedback Description
Actors:	Admin
Description:	The admin will be able to view the user feedback description.
Trigger:	The admin presses the View Details button
Preconditions:	N/A
Post conditions:	POST-1. Admin will be able to view the user feedback description
Normal Flow:	<ol style="list-style-type: none"> 1. Admin logs in 2. Admin selects the View Feedback button from the sidebar. 3. List of all the feedbacks will be displayed. 4. The admin will click on a specific user feedback details button. 5. The details of the user feedback are displayed.
Alternative Flows	N/A
Exceptions:	N/A
Business Rules:	N/A
Assumptions:	The admin has already logged in

3.2.2.7.9 Sign-out

Use Case ID:	UC-43
Use Case Name:	Sign out
Actors:	Admin
Description:	Admin can log out of the system
Trigger:	Admin presses the Log out button
Preconditions:	PRE-1. Admin is logged in to the system
Post conditions:	POST-1. Admin will be logged out of the system
Normal Flow:	<ol style="list-style-type: none"> 1. Admin logs in 2. Admin clicks the logout button 3. Admin is logged out 4. Admin is redirected to the homepage
Alternative Flows	N/A
Exceptions:	
Business Rules:	N/A
Assumptions:	<ol style="list-style-type: none"> 1. The admin has already logged in 2. The admin has working internet connection

3.2.2.7.10 Manage cat food inventory in the store

Use Case ID:	UC-44
Use Case Name:	Manage Cat Food Inventory
Actors:	Primary Actor: Admin Secondary Actor: System
Description:	Admin can view and manage cat food inventory in the store to ensure availability and update stock levels.
Trigger:	Admin accesses the inventory management interface.
Preconditions:	PRE-1. Admin is logged into the system. PRE-2. Cat food inventory data is available.
Postconditions:	POST-1. Admin updates inventory levels, adds new products, or removes outdated items as needed.
Normal Flow:	1. Admin logs into the system and navigates to the inventory management section. 2. Admin views the current inventory status and stock levels. 3. Admin makes changes to inventory (e.g., add products, update quantities).
Alternative Flows:	Ex-1. If an item is out of stock, Admin can place orders with suppliers or update expected arrival dates.
Exceptions:	None
Business Rules:	BR-1. Admin should regularly update inventory data to reflect current stock levels. BR-2. Admin actions should align with store policies and product availability.

3.2.2.8 Module 8: Statistical Analytics

3.2.2.8.1 Generate Reports on Most Popular Cat Breeds Sold

Use Case ID	UC-45
Use Case Name	Generate Reports on Most Popular Cat Breeds Sold
Actors	Admin, System
Description	System generates reports on the most popular cat breeds sold in the store.
Trigger	Admin requests a report on popular cat breeds.
Preconditions	Sales data is available and up-to-date.
Postconditions	Report on popular cat breeds is generated and accessible to the admin.
Normal Flow	1. Admin navigates to the analytics section.2. Admin selects the option to generate a report on popular cat breeds.3. System processes the sales data.4. System generates the report highlighting the most popular cat breeds.5. Admin reviews the report.
Exceptions	Insufficient sales data to generate a report.
Business Rules	Reports should be accurate and reflect the latest sales data.
Assumptions	The system has comprehensive sales data for analysis.

3.2.2.8.2 Track Performance of Individual Products

Use Case ID	UC-46
Use Case Name	Track Performance of Individual Products
Actors	Admin, System

Description	System tracks the performance of individual products in terms of sales, revenue, and customer ratings.
Trigger	Admin wants to analyze the performance of a specific product.
Preconditions	Product performance data is available.
Postconditions	Performance data for individual products is displayed.
Normal Flow	1. Admin navigates to the product performance section.2. Admin selects a product to analyze.3. System retrieves performance data for the selected product.4. System displays data on sales, revenue, and customer ratings.5. Admin reviews the performance metrics.
Exceptions	No performance data available for the selected product.
Business Rules	Performance metrics should be up-to-date and accurate.
Assumptions	The system has historical data for individual products.

3.2.2.8.3 Implement Real-Time Data Dashboards

Use Case ID	UC-47
Use Case Name	Implement Real-Time Data Dashboards
Actors	Admin, System
Description	System provides real-time data dashboards for quick insights into sales performance.
Trigger	Admin wants to view real-time sales data.
Preconditions	Real-time data collection is enabled.
Postconditions	Real-time data dashboards are displayed to the admin.
Normal Flow	1. Admin navigates to the real-time dashboard section.2. System retrieves and displays real-time sales data.3. Admin reviews the real-time data on the dashboard.4. System continuously updates the dashboard with new data.
Exceptions	Real-time data retrieval failure.
Business Rules	Dashboards should be updated continuously and reflect the latest data.
Assumptions	The system can collect and process real-time sales data efficiently.

3.2.2.8.4 Monitor Inventory Levels

Use Case ID	UC-48
Use Case Name	Monitor Inventory Levels
Actors	Admin, System
Description	System monitors inventory levels, tracks stockouts, and generates alerts for low stock or overstock situations.
Trigger	Inventory levels reach critical thresholds.
Preconditions	Inventory tracking is enabled.
Postconditions	Alerts for low stock or overstock situations are generated and sent to the admin.
Normal Flow	1. System continuously monitors inventory levels.2. System identifies low stock or overstock situations.3. System generates alerts for these situations.4. Admin receives and reviews the alerts.5. Admin takes necessary actions to address the inventory issues.
Exceptions	Inventory tracking system failure.
Business Rules	Inventory monitoring should be accurate and proactive.

Assumptions	The system can accurately track and report inventory levels.
-------------	--

3.2.2.8.5 Generate Monthly Sales Reports

Use Case ID	UC-49
Use Case Name	Generate Monthly Sales Reports
Actors	Admin, System
Description	System generates monthly sales reports to provide insights into the store's performance.
Trigger	End of the month.
Preconditions	Sales data for the month is complete.
Postconditions	Monthly sales report is generated and available for review.
Normal Flow	1. System identifies the end of the month.2. System compiles sales data for the month.3. System generates the monthly sales report.4. Admin reviews the sales report.
Exceptions	Incomplete sales data for the month.
Business Rules	Reports should be comprehensive and accurate.
Assumptions	The system can compile and analyze monthly sales data efficiently.

3.2.2.8.6 Provide Yearly Sales Analysis

Use Case ID	UC-50
Use Case Name	Provide Yearly Sales Analysis
Actors	Admin, System
Description	System provides a detailed yearly sales analysis, including trends and growth rates.
Trigger	End of the year.
Preconditions	Sales data for the year is complete.
Postconditions	Yearly sales analysis report is generated and available for review.
Normal Flow	1. System identifies the end of the year.2. System compiles sales data for the year.3. System analyzes trends and growth rates.4. System generates the yearly sales analysis report.5. Admin reviews the yearly sales analysis.
Exceptions	Incomplete sales data for the year.
Business Rules	Analysis should include trends, growth rates, and other relevant metrics.
Assumptions	The system can compile and analyze yearly sales data effectively.

3.2.2.8.7 Generate Custom Sales Reports

Use Case ID	UC-51
Use Case Name	Generate Custom Sales Reports
Actors	Admin, System
Description	Admin can generate custom sales reports based on specific criteria such as date range, product category, and region.
Trigger	Admin requests a custom sales report.
Preconditions	Sales data for the selected criteria is available.
Postconditions	Custom sales report is generated and available for review.

Normal Flow	1. Admin navigates to the custom reports section. 2. Admin specifies criteria for the custom report. 3. System processes the request based on the selected criteria. 4. System generates the custom sales report. 5. Admin reviews the custom sales report.
Exceptions	No data available for the selected criteria.
Business Rules	Custom reports should be flexible and tailored to admin needs.
Assumptions	The system can process and generate reports based on various criteria.

3.2.2.8.8 Analyze Customer Feedback Trends

Use Case ID	UC-52
Use Case Name	Analyze Customer Feedback Trends
Actors	Admin, System
Description	System analyzes customer feedback trends to identify areas for improvement.
Trigger	Admin requests an analysis of customer feedback.
Preconditions	Customer feedback data is available.
Postconditions	Feedback trends report is generated and available for review.
Normal Flow	1. Admin navigates to the feedback analysis section. 2. Admin requests an analysis of customer feedback. 3. System compiles and analyzes feedback data. 4. System generates a report highlighting feedback trends. 5. Admin reviews the feedback trends report.
Exceptions	Insufficient feedback data for meaningful analysis.
Business Rules	Analysis should provide actionable insights for improvement.
Assumptions	The system has comprehensive feedback data for analysis.

3.3 Functional Requirements

3.3.1 Module 1: User Authentication and Profile Management

Table 1: Customer.Signup

Identifier	FR-1
Title	Customer.Signup
Requirement	The system shall allow a customer to sign up.
Source	Team Member 1
Rationale	To allow a customer to make an account and signup for the website.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 2: Customer.Signup.Duplicate

Identifier	FR-2
Title	Customer.Signup.Duplicate
Requirement	The system shall confirm that customer is not already a register user.
Source	Team Member 2
Rationale	To ensure system does not accept signup on email linked to an existing account.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 3: Customer.Signup.Form

Identifier	FR-3
Title	Customer.Signup.Form
Requirement	The system shall require the necessary information for signup process.
Source	Team Member 1
Rationale	This requirement ensures that the system collects all necessary information for customer signup.
Business Rule	N/A

Dependencies	N/A
Priority	High

Table 4: Customer.Signup.Form.Empty

Identifier	FR-4
Title	Customer.Signup.Form.Empty
Requirement	The system shall display an alert if any of the form fields are left empty.
Source	Team Member 1
Rationale	This requirement ensures that customers provide all necessary information for registration.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 5: Customer.Signup.Cancel

Identifier	FR-5
Title	Customer.Signup.Cancel
Requirement	The system shall terminate the signup process if the customer cancels.
Source	Team Member 1
Rationale	This requirement ensures that the customer signup process is stopped if the customer cancels at any point.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 6: Customer.Signup.Form.Email.Standard

Identifier	FR-6
Title	Customer.Signup.Form.Email.Standard
Requirement	The system shall display an alert if the entered email does not adhere to the standard format.
Source	Team Member 1
Rationale	This requirement ensures that customers provide a valid email address for registration.
Business Rule	N/A
Dependencies	N/A

Priority	High
-----------------	------

Table 7: Customer.Signup.Form.Password.Standard

Identifier	FR-7
Title	Customer.Signup.Form.Password.Standard
Requirement	The system shall display an alert if the entered password does not adhere to the standard format.
Source	Team Member 1
Rationale	This requirement ensures that customers provide a strong password for their account.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 8: Customer.Signup.Email.Duplicate.Registered

Identifier	FR-8
Title	Customer.Signup.Email.Duplicate.Registered
Requirement	The system shall display an alert if the entered email is already registered for another account.
Source	Team Member 1
Rationale	Customer should be notified if the email is already a registered account.
Business Rule	N/A
Dependencies	FR-1.2
Priority	High

Table 9: Customer.Signup.Email.Duplicate.Registered.Redirect.Login

Identifier	FR-9
Title	Customer.Signup.Email.Duplicate.Registered.Redirect.Login
Requirement	The system shall redirect the customer to the login page and terminate the use case if the email is already registered.
Source	Team Member 1
Rationale	If the customer email already links to an account, the system shall redirect to login page.
Business Rule	N/A
Dependencies	FR-1.2

Priority	High
-----------------	------

Table 10: Customer.Signup.Create.Account

Identifier	FR-10
Title	Customer.Signup.Create.Account
Requirement	The system shall create a customer account with the provided information.
Source	Team Member 1
Rationale	This requirement ensures that the customer registration process is completed successfully.
Business Rule	The customer shall be at least 18 years old to register.
Dependencies	N/A
Priority	High

Table 11: Customer.Signup.Create.Account.Failed

Identifier	FR-11
Title	Customer.Signup.Create.Account.Failed
Requirement	The system shall notify customer for any errors in signup process regarding customer data.
Source	Team Member 1
Rationale	This requirement allows the customer to correct any errors made in the registration process
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 12: Customer.Signup.Success

Identifier	FR-12
Title	Customer.Signup.Success
Requirement	The system shall direct the customer to the homepage after successful registration.
Source	Team Member 1
Rationale	Once signup is successful the customer account is created, customer is logged in and redirected to homepage
Business Rule	N/A
Dependencies	FR-1.10
Priority	High

Table 13: Signup.Google

Identifier	FR-13
Title	Signup.Google
Requirement	The system shall have an option to sign up with google.
Source	Team Member 1
Rationale	This requirement provides customers with an alternative path for registration using their Google account for a seamless account creation.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 14 Signup.Google.Permission

Identifier	FR-14
Title	Signup.Google.Permission
Requirement	The system shall redirect the customer to Google's permission page after clicking the Signup with Google button.
Source	Team Member 1
Rationale	The system directs the customer to give permission to the system to connect with google account.
Business Rule	N/A
Dependencies	FR-1.13
Priority	High

Table 15: Signup.Google.Permission.No

Identifier	FR-15
Title	Signup.Google.Permission.No
Requirement	The system shall redirect the customer to the manual signup page if they deny access to their Google account.
Source	Team Member 1
Rationale	If the google account access is not provided, then the customer can signup manually
Business Rule	N/A
Dependencies	FR-1.13
Priority	High

Table 16: Signup.Google.Account.Link

Identifier	FR-16
Title	Signup.Google.Account.Link
Requirement	The system shall create a customer account using the Google account information and direct the customer to the homepage.
Source	Team Member 1
Rationale	Once the google account access is provided by customer system will make an account with the google information and allow customer system access.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 17: Customer.Login

Identifier	FR-17
Title	Customer.Login
Requirement	The system shall allow customers to login using their email and password.
Source	Team Member 1
Rationale	Customer with accounts can login using their email and password.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 18: Customer.Account.Active

Identifier	FR-18
Title	Customer.Account.Active
Requirement	The system shall confirm that customer have an active account.
Source	Team Member 2
Rationale	This requirement is necessary to ensure that only customers with active accounts can log in to the system and perform actions that are specific to their account.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 19: Customer.Account.Inactive

Identifier	FR-19
Title	Customer.Account.Inactive
Requirement	The system shall prompt the customer that their account has been deactivated.
Source	Team Member 2
Rationale	If admin bans a customer, then they cannot be logged in to the system
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 20: Customer.Login.Form.Empty

Identifier	FR-20
Title	Customer.Login.Form.Empty
Requirement	The system shall display an alert message when the customer leaves any form field empty on login page.
Source	Team Member 2
Rationale	Email and password are mandatory to proceed with the login process.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 21: Customer.Login.Cancel

Identifier	FR-21
Title	Customer.Login.Cancel
Requirement	The system shall terminate the use case if the customer cancels.
Source	Team Member 2
Rationale	This requirement is necessary to provide customers with the ability to cancel the login process if they change their mind or encounter issues, and to prevent any unintended actions or errors.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 22: Customer.Login.Email.Standard

Identifier	FR-22
Title	Customer.Login.Email.Standard
Requirement	The system shall display an alert message when the customer enters an invalid email format while logging in.
Source	Team Member 2
Rationale	This requirement is necessary to ensure that customers provide valid email addresses when logging in.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 23: Customer.Login.No.Account

Identifier	FR-23
Title	Customer.Login.No.Account
Requirement	The system shall display an alert message when the customer's email does not link to any registered account.
Source	Team Member 2
Rationale	This requirement is necessary to prevent unauthorized access to accounts by ensuring that only email addresses that linked to registered accounts are allowed to log in.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 24: Customer.Login.Success

Identifier	FR-24
Title	Customer.Login.Success
Requirement	The system shall redirect the customer to the homepage upon successful login.
Source	Team Member 2
Rationale	Customers are directed to homepage after successful login process.
Business Rule	N/A
Dependencies	FR-18,FR-22
Priority	High

Table 25: Customer.Login.Google

Identifier	FR-25
Title	Customer.Login.Google
Requirement	The system shall provide a "Login with Google" option for customers to log in using their Google account.
Source	Team Member 2
Rationale	This requirement is necessary to provide customers with an alternative and convenient way to log in to their accounts using their Google account.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 26: Customer.Login.Google.Permission

Identifier	FR-26
Title	Customer.Login.Google.Permission
Requirement	The system shall display a pop-up asking for permission to access the customer's Google account when the customer clicks the "Login with Google" button.
Source	Team Member 2
Rationale	This requirement is necessary to ensure that customers are aware of and consent to the system's access to their Google account information, and to prevent unauthorized access to their account information.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 27: Customer.Login.Google.Permission.Denied

Identifier	FR-27
Title	Customer.Login.Google.Permission.Denied
Requirement	The system shall display an alert message when the customer denies access to their Google account while logging in and direct to homepage.
Source	Team Member 2
Rationale	This requirement is necessary to provide customers with a clear and visible notification if they deny access to their Google account information
Business Rule	N/A
Dependencies	N/A

Priority	High
-----------------	------

Table 28: Customer.Login.Google.Success

Identifier	FR-28
Title	Customer.Login.Google.Success
Requirement	The system shall redirect the customer to the homepage upon successful login using Google.
Source	Team Member 2
Rationale	Customer is logged in and redirected to homepage after successful link to google account.
Business Rule	N/A
Dependencies	FR-24
Priority	High

Table 29: Customer.Authenticate

Identifier	FR-29
Title	Customer.Authenticate
Requirement	The system shall validate the customer's entered credentials
Source	Team Member 1
Rationale	The system verifies that the entered credentials are valid and exist in the system.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 30: Customer.Credentials.Invalid

Identifier	FR-30
Title	Customer.Credentials.Invalid
Requirement	The system shall prompt the customer if the given credentials are wrong.
Source	Team Member 1
Rationale	The system should tell the customer that the given credentials are wrong
Business Rule	N/A
Dependencies	FR-29
Priority	High

Table31: Customer.Ban.Check

Identifier	FR-31
Title	Customer.Ban.Check
Requirement	The system shall check whether the customer is banned before granting access to the system.
Source	Team Member 1
Rationale	The system checks whether the customer is banned or not. Banned Customer should not be allowed to the system.
Business Rule	BR-1: Banned customers are not allowed to use the system.
Dependencies	FR-29
Priority	High

Table 32: Customer.Authenticate.SystemError

Identifier	FR-32
Title	Customer.Authenticate.SystemError
Requirement	The system shall prompt the customer if the system is down.
Source	Team Member 1
Rationale	The system should respond appropriately if the system is down or under maintenance.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 33: Sign.out

Identifier	FR-33
Title	Sign.out
Requirement	The users will be able to sign out of their application on clicking sign out
Source	Team Member 1
Rationale	To allow users to sign out of the application so they cannot access the app anymore
Business Rule (if required)	N/A
Dependencies	N/A
Priority	High

Table 34: Select.Profile

Identifier	FR-34
Title	Select.Profile
Requirement	The users will be able to view their profile and edit their personal information by selecting the personal information button in the settings
Source	Talha Ziaullah
Rationale	To empower users to access and update their personal information as needed
Business Rule (if required)	N/A
Dependencies	N/A
Priority	Medium

Table 35: Select.Personal.Information

Identifier	FR-35
Title	Select.Personal.Information
Requirement	The users will be able to view the personal information by select the personal information button
Source	Team Member 1
Rationale	To view user's personal information so they can update it if they want
Business Rule (if required)	N/A
Dependencies	N/A
Priority	Medium

Table 36: Select.Update.Button

Identifier	FR-36
Title	Select.Update.Button
Requirement	The users will be able to update the personal information by select the update button within their profile settings
Source	Team Member 1
Rationale	To allow the user to update the personal information if they want to
Business Rule (if required)	N/A
Dependencies	N/A
Priority	Medium

Table 37: Change.Username

Identifier	FR-37
Title	Change.Username
Requirement	The users will be able to change their email to a new one
Source	Team Member 1
Rationale	To allow users to update their email address as needed.
Business Rule (if required)	N/A
Dependencies	N/A
Priority	Medium

Table 38: Change.Address

Identifier	FR-38
Title	Change.Address
Requirement	The users will be able to change their address
Source	Team Member 1
Rationale	To change their address to a new one
Business Rule (if required)	N/A
Dependencies	N/A
Priority	Medium

Table 39: Change.Phone.Number

Identifier	FR-39
Title	Change.Phone.Number
Requirement	The users will be able to change their phone number
Source	Team Member 1
Rationale	To change their phone number to a new one
Business Rule (if required)	N/A
Dependencies	N/A
Priority	Medium

Table 40: Save.Changes

Identifier	FR-40
Title	Save.Changes
Requirement	Users shall be able to save their changes by clicking the “Save Changes” button in the profile settings. This action will update their profile information with the newly provided details.
Source	Team Member 1
Rationale	To allow users to persist the changes made to their profile information.
Business Rule (if required)	N/A
Dependencies	N/A
Priority	Medium

Table 41: Forget.Password

Identifier	FR-41
Title	Forget.Password
Requirement	The users will be able to reset the password by clicking on Forgot password button
Source	Team Member 1
Rationale	To reset their password in case they have forgotten one
Business Rule (if required)	N/A
Dependencies	N/A
Priority	Medium

Table 42: Verification.Link

Identifier	FR-42
Title	Verification.Link
Requirement	The users will be able to click the verification link that is sent to the provided email address
Source	Team Member 1
Rationale	To generate a link that will reset the password of that user
Business Rule (if required)	N/A
Dependencies	FR-41
Priority	High

Table 43: Enter.New.Password

Identifier	FR-43
Title	Enter new password
Requirement	The users will be able to enter a new password
Source	Team Member 1
Rationale	To allow authorized access to the application
Business Rule (if required)	N/A
Dependencies	FR-41
Priority	High

Table 44: Enter.New.Confirm.Password

Identifier	FR-44
Title	Enter.New.Confirmation.Password
Requirement	The users will be able to confirm the new password
Source	Team Member 1
Rationale	To ensure no mistyping of password occurs
Business Rule (if required)	N/A
Dependencies	FR-43
Priority	High

Table 45: Delete.Account

Identifier	FR-45
Title	Delete.Account
Requirement	The user shall be able to initiate the process of deleting their account by clicking the “Delete Account” button within their profile settings.
Source	Team Member 1
Rationale	To provide users with the option to permanently remove their account from the application if they no longer wish to use it.
Business Rule (if required)	N/A
Dependencies	N/A
Priority	Low

Table 46: Confirm.Deletion

Identifier	FR-46
Title	Confirm.Deletion
Requirement	User shall be required to confirm the account deletion process by clicking “Yes” on the confirmation dialogue box. The extra step is added to minimize the risk of accidental account deletion.
Source	Team Member 1
Rationale	To safeguard users against accidental account deletion by introducing a confirmation step.
Business Rule (if required)	N/A
Dependencies	FR-45
Priority	Medium

3.3.2 Module 2: Cat Breed Detection

Table 47: Image.Upload.Functionality

Identifier	FR-47
Title	Image.Upload.Functionality
Requirement	Users shall be able to upload images of their cats for breed recognition. The system shall process the uploaded image to identify the breed of the cat.
Source	Team Member 2
Rationale	This requirement enables users to utilize the system for identifying the breed of their cats through image submission, enhancing user experience and engagement.
Business Rule	The uploaded images must be clear and of sufficient quality for accurate breed recognition
Dependencies	N/A
Priority	High

Table 48: Error Handling for Unsupported Image Formats

Identifier	FR-48
Title	Error Handling for Unsupported Image Formats.

Requirement	If a user attempts to upload an image in an unsupported format, the system shall display an error message indicating the acceptable formats (e.g., JPEG, PNG).
Source	Team Member 2
Rationale	This ensures users are informed about the acceptable image formats, reducing frustration and guiding them to provide the correct input.
Business Rule	Users must be notified of the supported image formats to ensure successful upload.
Dependencies	FR-47
Priority	High

Table 49: Resizing and Normalizing Images

Identifier	FR-49
Title	Resizing and Normalizing Images
Requirement	System shall, upon receiving input images, automatically resize them to a standardized dimension to ensure uniformity in image size across the dataset.
Source	Team Member 2
Rationale	To ensure consistent input data for processing and analysis of images.
Business Rule	N/A
Dependencies	FR-47
Priority	Medium

Table 50: Image.Submission.Confirmation

Identifier	FR-50
Title	Image.Submission.Confirmation
Requirement	Upon successful image upload, the system shall display a confirmation message to the user.
Source	Team Member 2
Rationale	Providing confirmation reassures users that their submission was successful, enhancing their confidence in the system.
Business Rule	N/A
Dependencies	FR-47
Priority	Low

Table 51: Camera.Access

Identifier	FR-51
Title	Camera.Access
Requirement	The system must ask for camera access to the customer.
Source	Team Member 2
Rationale	This requirement ensures that the system requests permission from the customer to access their device's camera, which is necessary for accessing of image.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 52: Standardization.Pixel.Values.Sizes

Identifier	FR-52
Title	Standardization.Pixel.Values.Sizes
Requirement	The system shall preprocess images submitted by users to standardize pixel values and sizes. This process involves ensuring that all images have consistent pixel values and sizes, regardless of their original format or resolution.
Source	Team Member 2
Rationale	Standardizing pixel values and sizes ensures that images processed by the system are uniform, facilitating easier analysis, comparison, and manipulation.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 53: Handling.Image.Formats

Identifier	FR-53
Title	Handling.Image.Formats
Requirement	The system shall support various image formats (e.g., JPEG, PNG, GIF) for submission within the Image Submission module.
Source	Team Member 2

Rationale	Supporting multiple image formats enhances user flexibility and convenience, allowing them to submit images in their preferred format.
Business Rule	Users are allowed to submit images in any of the supported formats specified by the system.
Dependencies	N/A
Priority	Medium

Table 54: Image.Quality.Assurance

Identifier	FR-54
Title	Image.Quality.Assurance
Requirement	The system shall perform quality checks on submitted images to ensure they meet predefined quality standards.
Source	Team Member 2
Rationale	Implementing quality assurance measures helps prevent the submission of low-quality or corrupted images, thus maintaining data accuracy and reliability.
Business Rule	Images failing to meet the quality standards will not be accepted for further processing within the system.
Dependencies	FR-52
Priority	Medium

Table 55: Image Validation and Normalization

Identifier	FR-55
Title	Image.Validation.Normalization
Requirement	The Image Submission module shall validate and normalize input images to enhance model accuracy.
Source	Team Member 2
Rationale	Validating and normalizing input images is crucial for ensuring that the model receives consistent and reliable data, which ultimately enhances its accuracy in image recognition tasks.
Business Rule	N/A
Dependencies	N/A
Priority	Medium

Table 56: Format.Compatibility.Check

Identifier	FR-56
Title	Format.Compatibility.Check
Requirement	The Image Submission module shall verify the format compatibility of input images. Supported formats include JPEG, PNG, and GIF.
Source	Team Member 2
Rationale	Ensuring format compatibility helps prevent processing errors and maintains the integrity of the image data throughout the submission process.
Business Rule	N/A
Dependencies	N/A
Priority	Medium

Table 57: Splitting.Dataset.for.Model.Training

Identifier	FR-57
Title	Splitting Dataset for Model Training
Requirement	The system shall split the dataset into training and testing sets for model training. The splitting process should ensure that a certain percentage of data is allocated for training and the remaining for testing.
Source	Team Member 1
Rationale	Splitting the dataset into training and testing sets is crucial for evaluating the performance of machine learning models accurately. It allows the model to learn patterns from the training data and assess its generalization ability on unseen data.
Business Rule	N/A
Dependencies	N/A
Priority	Medium

Table 58: Integration of ResNet-50 Architecture

Identifier	FR-58
Title	Integration of ResNet-50 Architecture
Requirement	The system shall integrate the ResNet-50 architecture for accurate cat breed recognition.
Source	Team Member 1

Rationale	The integration of ResNet-50 architecture is essential to leverage its state-of-the-art deep learning capabilities, ensuring accurate identification of cat breeds within the FelineConnect system.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 59: Cat.Image.Processing

Identifier	FR-59
Title	Cat.Image.Processing
Requirement	The system shall preprocess uploaded cat images before feeding them into the ResNet-50 architecture for breed recognition.
Source	Team Member 1
Rationale	Preprocessing cat images ensures optimal input for the ResNet-50 architecture, enhancing the accuracy and speed of breed recognition.
Business Rule	N/A
Dependencies	FR-58
Priority	High

Table 60: Feedback.Loop.Implementation

Identifier	FR-60
Title	Feedback.Loop.Implementation
Requirement	The system shall implement a feedback loop mechanism for users to provide insights into the accuracy of breed recognition results.
Source	Team Member 1
Rationale	Feedback from users facilitates continuous improvement of the breed recognition model, ensuring enhanced accuracy and user satisfaction over time.
Business Rule	N/A
Dependencies	N/A
Priority	Medium

Table 61: Integration of ResNet-50 Model for Transfer Learning

Identifier	FR-61
Title	Integration of ResNet-50 Model for Transfer Learning

Requirement	The system shall integrate a pre-trained ResNet-50 model for transfer learning purposes.
Source	Team Member 1
Rationale	This requirement is essential to leverage the existing knowledge encoded in the ResNet-50 model for fine-tuning on specific cat breed recognition tasks, which will expedite the development process and improve model performance.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 62: Data Preparation for Transfer Learning

Identifier	FR-62
Title	Data Preparation for Transfer Learning
Requirement	Prior to transfer learning, the system shall preprocess the input cat image data to ensure compatibility with the ResNet-50 model architecture.
Source	Team Member 1
Rationale	Proper data preprocessing is crucial for effective transfer learning, as it ensures that the input data aligns with the requirements of the ResNet-50 model, thereby enhancing the efficiency and accuracy of the training process.
Business Rule	N/A
Dependencies	FR-61
Priority	High

Table 63: Fine-Tuning ResNet-50 Model

Identifier	FR-63
Title	Fine-Tuning ResNet-50 Model
Requirement	The system shall fine-tune the integrated ResNet-50 model using the preprocessed cat image dataset to adapt its weights and parameters for accurate breed recognition.
Source	Team Member 1
Rationale	Fine-tuning the ResNet-50 model on cat image data is necessary to customize its features for the specific task of breed recognition, thereby improving the model's accuracy and performance in identifying different cat breeds.
Business Rule	N/A

Dependencies	FR-61
Priority	High

Table 64: Model Evaluation and Validation

Identifier	FR-64
Title	Model Evaluation and Validation
Requirement	The system shall evaluate and validate the performance of the fine-tuned ResNet-50 model using appropriate metrics
Source	Team Member 2
Rationale	Model evaluation and validation are crucial steps to ensure the effectiveness and reliability of the integrated ResNet-50 model for breed recognition tasks, enabling informed decisions regarding its deployment in the FelineConnect system.
Business Rule	N/A
Dependencies	FR-63
Priority	High

Table 65: Fine-tuning ResNet-50 Model on Domestic Cat Breeds Dataset

Identifier	FR-65
Title	Fine-tuning ResNet-50 Model on Domestic Cat Breeds Dataset
Requirement	The system shall fine-tune the ResNet-50 deep learning model on a dataset comprising various domestic cat breeds.
Source	Team Member 2
Rationale	Fine-tuning the ResNet-50 model specifically on a dataset of domestic cat breeds is essential to ensure accurate breed identification for the target user base of FelineConnect.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 66: Model Validation for Breed Recognition

Identifier	FR-66
-------------------	-------

Title	Model Validation for Breed Recognition
Requirement	The system shall fine-tune the ResNet-50 deep learning model on a dataset comprising various domestic cat breeds.
Source	Team Member 2
Rationale	Validating the model's accuracy with an independent test dataset ensures robustness and reliability in breed recognition, thereby enhancing user trust in the system's capabilities.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 67: User Feedback Integration

Identifier	FR-67
Title	User Feedback Integration
Requirement	The system shall integrate a user feedback mechanism allowing users to provide insights into breed recognition accuracy.
Source	Team Member 2
Rationale	Integrating a user feedback loop facilitates continuous improvement of the model's accuracy, ensuring enhanced performance over time and increased user satisfaction with breed identification results.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 67: Display detected cat breed along with confidence score.

Identifier	FR-67
Title	Display detected cat breed along with confidence score.
Requirement	The system shall display the detected cat breed along with the confidence score after analyzing the uploaded cat image.
Source	Team Member 2
Rationale	Users need to know the detected breed and the confidence level to trust the accuracy of the system's identification.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 68: Provide options for user feedback on breed detection accuracy.

Identifier	FR-68
Title	Provide options for user feedback on breed detection accuracy.
Requirement	The system shall offer users the option to provide feedback on the accuracy of the detected cat breed, allowing them to indicate if the identification was correct or incorrect.
Source	Team Member 2
Rationale	User feedback is essential for continuous improvement of the breed detection model, enhancing its accuracy and reliability over time.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 69: Implement algorithm for determining confidence score.

Identifier	FR-69
Title	Implement algorithm for determining confidence score.
Requirement	The system shall employ an algorithm to calculate the confidence score for each detected cat breed, based on the analysis of features extracted from the uploaded cat image.
Source	Team Member 2
Rationale	The confidence score provides users with insight into the reliability of the detected breed identification, enhancing trust in the system's capabilities.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 70: Allow users to view breed detection history.

Identifier	FR-70
Title	Allow users to view breed detection history.
Requirement	The system shall enable users to access their breed detection history, displaying previously uploaded images along with the corresponding detected breeds and confidence scores.
Source	Team Member 2
Rationale	Providing access to detection history allows users to track past identification results and monitor changes in accuracy over time.

Business Rule	N/A
Dependencies	N/A
Priority	High

Table 71: Visual Representation of Breed Identification Features

Identifier	FR-71
Title	Visual Representation of Breed Identification Features
Requirement	The system shall display a visual representation of key features contributing to the identification of a cat breed based on uploaded images.
Source	Team Member 1
Rationale	Providing users with a visual representation of key features contributing to breed identification enhances user understanding and confidence in the accuracy of the system's results.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 72: Instant Breed Identification Feedback

Identifier	FR-72
Title	Instant Breed Identification Feedback
Requirement	Upon uploading an image, the system shall promptly generate and display the breed identification results, along with the visual representation of key features contributing to the identification.
Source	Team Member 2
Rationale	Instant feedback enhances user experience by providing prompt results, contributing to user satisfaction and engagement with the platform.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 73: Customizable Visual Representation Settings

Identifier	FR-73
Title	Customizable Visual Representation Settings
Requirement	The system shall allow users to customize the visual representation settings for breed identification features. Users shall have the option to adjust the

	transparency, color, and size of highlighted areas or overlays according to their preferences.
Source	Team Member 2
Rationale	Providing customization options enhances user engagement and satisfaction by allowing users to tailor the visual representation to their preferences and needs.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 74: Breed Identification Result Sharing

Identifier	FR-74
Title	Breed Identification Result Sharing
Requirement	The system shall provide functionality for users to share breed identification results via various social media platforms or email.
Source	Team Member 1
Rationale	Enabling result sharing enhances user engagement and promotes platform visibility, allowing users to easily share their cat's breed identification with friends, family, or social media followers.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 75: User Feedback on Breed Recognition Accuracy

Identifier	FR-75
Title	User Feedback on Breed Recognition Accuracy
Requirement	Users shall be able to provide feedback on the accuracy of breed recognition results generated by the system.
Source	Team Member 2
Rationale	Continuous improvement of breed recognition accuracy is crucial for enhancing user satisfaction and system reliability.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 76: Feedback Analysis and Reporting

Identifier	FR-76
Title	Feedback Analysis and Reporting
Requirement	The system shall analyze aggregated user feedback data periodically to identify trends, patterns, and areas for improvement in breed recognition accuracy.
Source	Team Member 2
Rationale	Analyzing user feedback enables systematic evaluation of the system's performance and guides decision-making for future enhancements and optimizations.
Business Rule	N/A
Dependencies	FR-75
Priority	High

3.3.3 Module 3: Online Store

Table 77: Product.Stock

Identifier	FR-77
Title	Product.Stock
Requirement	The system shall check that the product is in stock.
Source	Team Member 2
Rationale	Ensuring product availability to add in cart.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 78: Product.Stock.No

Identifier	FR-78
Title	Product.Stock.No
Requirement	The system shall alter a message that the product is out of stock.
Source	Team Member 2
Rationale	Providing message when product is out of stock.

Business Rule	N/A
Dependencies	FR-77
Priority	High

Table 79: Product.Customer.LoggedIn

Identifier	FR-79
Title	Product.Customer.LoggedIn
Requirement	The system shall check that that the customer is logged in.
Source	Team Member 2
Rationale	Checking customer login status before adding a product to cart
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 80: Product.Customer.LoggedIn.No

Identifier	FR-80
Title	Product.Customer.LoggedIn.No
Requirement	The system shall direct the customer to login page if the customer is not logged in.
Source	Team Member 2
Rationale	Directing customer to login page if not logged in
Business Rule	N/A
Dependencies	FR-79
Priority	High

Table 81: Product.Viewdetail

Identifier	FR-81
Title	Product.Viewdetail
Requirement	The system shall display the product details on the product page
Source	Team Member 2
Rationale	Displaying product details to the customer.
Business Rule	N/A
Dependencies	N/A

Priority	High
-----------------	------

Table 82: Product.Quantity

Identifier	FR-82
Title	Product.Quantity
Requirement	The system shall allow the customer to set the quantity of the product they want to add to the cart.
Source	Team Member 2
Rationale	Allowing customer to set quantity of the product they want to add
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 83: AddToCart.Button

Identifier	FR-83
Title	AddToCart.Button
Requirement	The system shall enable the customer to press the "Add to Cart" button.
Source	Team Member 1
Rationale	Enabling customer to add product to cart
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 84: AddToCart

Identifier	FR-84
Title	AddToCart
Requirement	The system shall add the selected product to the cart.
Source	Team Member 1
Rationale	Adding selected product to the cart
Business Rule	N/A
Dependencies	FR-82
Priority	High

Table 85: AddToCart.Yes

Identifier	FR-85
-------------------	-------

Title	AddToCart.Yes
Requirement	The system shall allow the customer to surf other product pages after adding a product to the cart.
Source	Team Member 1
Rationale	Allowing customer to surf other product pages after adding a product to cart
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 86: AddToCart.Identical

Identifier	FR-86
Title	AddToCart.Identical
Requirement	The system shall allow the customer to add multiple identical products to the cart.
Source	Team Member 2
Rationale	Allowing customer to add multiple identical products to cart
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 87: Product.Quantity.Insufficaint

Identifier	FR-87
Title	Product.Quantity.Insufficaint
Requirement	The system shall generate an alert if there is insufficient inventory to fulfill the required quantity of a product.
Source	Team Member 1
Rationale	Alerting customer if required product quantity is not available
Business Rule	N/A
Dependencies	FR-83
Priority	High

Table 88: AddToCart.Limit

Identifier	FR-88
Title	AddToCart.Limit

Requirement	The system shall limit a single product to be added to the cart with a maximum quantity of 10.
Source	Team Member 1
Rationale	Limiting a single product to be added to cart with a maximum quantity of 10
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 89: Cart.Product

Identifier	FR-89
Title	Cart.Product
Requirement	The system shall verify that user have at least one product in cart.
Source	Team Member 1
Rationale	The system ensures that the user has at least one product in their cart before deleting.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 90: Cart.Product.Empty

Identifier	FR-90
Title	Cart.Product.Empty
Requirement	The system shall direct the user to home page for add to cart.
Source	Team Member 1
Rationale	The system redirects the user to the home page if they try to access the cart without any product added to it
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 91: Product.Cart.Delete

Identifier	FR-91
Title	Product.Cart.Delete

Requirement	The system shall allow the customer to delete a single or more quantity of a product from the cart.
Source	Team Member 1
Rationale	The system allows the customer to delete one or more quantity of a product from the cart
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 92: Product.Cart.Delete.Recalculate

Identifier	FR-92
Title	Product.Cart.Delete.Recalculate
Requirement	The system shall recalculate the cart amount after deleting a product from the cart.
Source	Team Member 1
Rationale	The system recalculates the cart amount after a product is deleted from the cart
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 93: Delete.Single.Quantity

Identifier	FR-93
Title	Delete.Single.Quantity
Requirement	The system shall allow the customer to delete more than one quantity of a single product from the cart.
Source	Team Member 1
Rationale	Customer may want to delete more than one quantity of a single product.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 94: Clear.Empty.Cart

Identifier	FR-94
Title	Clear.Empty.Cart
Requirement	The system shall allow the customer to clear the complete cart.

Source	Team Member 2
Rationale	The system allows the customer to clear the complete cart.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 95: User Feedback on Purchased Cat Food Items

Identifier	FR-95
Title	User Feedback on Purchased Cat Food Items
Requirement	After purchasing cat food items from the online store, users shall be prompted to provide feedback and ratings for the purchased items. This feedback will be collected and displayed on the product pages to assist other users in making informed purchasing decisions.
Source	Team Member 1
Rationale	Incorporating a review and rating system for purchased cat food items enhances user engagement and satisfaction by facilitating peer-to-peer recommendations and fostering a sense of community within the platform.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 96: Display Product Reviews and Ratings

Identifier	FR-96
Title	Display Product Reviews and Ratings
Requirement	The online store interface shall display user-generated reviews and ratings for each cat food item on the respective product pages.
Source	Team Member 1
Rationale	Providing access to product reviews and ratings enhances the transparency and trustworthiness of the online store, empowering users to make informed purchasing decisions based on the experiences of others.
Business Rule	N/A
Dependencies	N/A
Priority	High

3.3.4 Module 4: Search Filters and Notifications

Table 97: Customer.Implement.search.bar

Identifier	FR-97
Title	Customer.Implement.search.bar
Requirement	Implement a search bar for users to find cat breeds within the application.
Source	Team Member 2
Rationale	This FR is required to allow customers to pay for their purchases and is necessary for the business to accept payments.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 98: Refine.Search.Result

Identifier	FR-98
Title	Refine.Search.Result
Requirement	Allow users to refine search results using filters like breed type, food category, price range.
Source	Team Member 1
Rationale	By enabling users to refine search results with filters, we enhance the search experience, allowing users to narrow down their options and find the most relevant information more efficiently.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 99: Search.Specific.Cat

Identifier	FR-99
Title	Search.Specific.Cat
Requirement	Allow users to search for specific cat food items.
Source	Team Member 1
Rationale	Expanding the search functionality to include specific cat food items enhances the utility of the application, catering to the diverse needs of users interested in purchasing cat food products.
Business Rule	N/A
Dependencies	N/A
Priority	High

3.3.5 Module 5: Payment Processing

Table 100: Customer.Payment.Method

Identifier	FR-100
Title	Customer.Payment.Method
Requirement	The system shall allow the customer to add a payment method during checkout.
Source	Team Member 1
Rationale	This FR is required to allow customers to pay for their purchases and is necessary for the business to accept payments.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 101: Customer.Card.Detail

Identifier	FR-101
Title	Customer.Card.Detail
Requirement	The system shall request the customer to provide the Card Number, Expiry Date, and CVC while adding a payment method.
Source	Team Member 2
Rationale	This FR is necessary to ensure that the correct payment details are collected and processed during the checkout process, ensuring accurate and secure payments.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 102: Payment.CashOnDelivery

Identifier	FR-102
Title	Paymnet.CashOnDelivery
Requirement	The system shall allow the customer to select Cash on Delivery as a payment method.
Source	Team Member 1

Rationale	This FR is required to provide customers with a convenient and familiar payment method.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 103: Payment.Method.Future

Identifier	FR-103
Title	Payment.Method.Future
Requirement	The system shall allow the customer to save a payment method for future purchases.
Source	Team Member 1
Rationale	This FR is necessary to save time for the customer during future checkouts and provides a seamless checkout experience.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 104: Payment.Method.Once

Identifier	FR-104
Title	Payment.Method.Once
Requirement	The system shall allow the customer to use a payment method once for the current purchase.
Source	Team Member 1
Rationale	This FR provides flexibility to the customer to choose how they want to pay for their purchase.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 105: CashOnDelivery.Limit

Identifier	FR-105
Title	CashOnDelivery.Limit
Requirement	The system shall not allow Cash on Delivery for orders above Rs. 15,000.
Source	Team Member 2

Rationale	This FR is necessary to mitigate risks associated with high-value purchases and to ensure that payments are received securely.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 106: Customer.Card.Format

Identifier	FR-106
Title	Customer.Card.Format
Requirement	The system shall display an alert if the Card Number entered by the customer is in the wrong format.
Source	Team Member 1
Rationale	This FR is necessary to ensure that customers enter valid payment details, and to avoid errors that could result in failed payments.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 107: Customer.Card.Expire

Identifier	FR-107
Title	Customer.Card.Expire
Requirement	This FR is necessary to ensure that customers enter valid payment details, and to avoid errors that could result in failed payments.
Source	Team Member 1
Rationale	The system allows the customer to clear the complete cart.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 108: Edit.Payment.Method

Identifier	FR-108
Title	Edit.Payment.Method
Requirement	The system shall allow the customer to edit their payment method.
Source	Team Member 2
Rationale	This requirement enables the customer to modify their payment method.

Business Rule	N/A
Dependencies	N/A
Priority	High

Table 109: Payment.Method.Valid

Identifier	FR-109
Title	Payment.Method.Valid
Requirement	The system shall check the customer has a valid payment method linked to the account.
Source	Team Member 1
Rationale	This requirement ensures that the customer has an existing payment method linked to their account before allowing them to edit it.
Business Rule	N/A
Dependencies	FR-105
Priority	High

Table 110: Payment.Method.Valid.No

Identifier	FR-110
Title	Payment.Method.Valid.No
Requirement	The system shall alter a message that user don't have a valid payment method linked.
Source	Team Member 2
Rationale	This requirement provides feedback to the user if they attempt to edit a payment method that is not linked to their account.
Business Rule	N/A
Dependencies	FR-106
Priority	High

Table 111: Payment.Method.Edit.New

Identifier	FR-111
Title	Payment.Method.Edit.New
Requirement	The system shall require the customer to provide a new Card Number, Expiry Date, and CVC for the updated payment method.
Source	Team Member 1
Rationale	This requirement ensures that the customer enters valid and updated payment details.

Business Rule	N/A
Dependencies	N/A
Priority	High

Table 112: Payment.Method.Edit.New.Save

Identifier	FR-112
Title	Payment.Method.Edit.New.Save
Requirement	The system must save the new payment method and link it to the customer's account.
Source	Team Member 1
Rationale	This requirement stores the updated payment details and links it to the customer's account for future use.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 113: Payment.Method.Edit.Card.Format

Identifier	FR-113
Title	Payment.Method.Edit.Card.Format
Requirement	The system shall verify that the Card Number entered by the customer is in the correct format.
Source	Team Member 2
Rationale	This requirement checks that the card number entered by the customer is in the correct format before saving it.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 114: Payment.Method.Edit.Card.Format.Wrong

Identifier	FR-114
Title	Payment.Method.Edit.Card.Format.Wrong
Requirement	The system shall display an alert message if the Card Number entered by the customer is not in the correct format.
Source	Team Member 1

Rationale	This requirement provides feedback to the user if they enter an incorrect card number format.
Business Rule	N/A
Dependencies	FR-81
Priority	High

Table115: Payment.Method.Edit.Card.ExpiryDate

Identifier	FR-115
Title	Payment.Method.Edit.Card.ExpiryDate
Requirement	The system shall verify that the Card Expiry Date entered by the customer is not in the past.
Source	Team Member 1
Rationale	This requirement checks that the card expiry date entered by the customer is not in the past.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 116: Payment.Method.Edit.Card.ExpiryDate.Passed

Identifier	FR-116
Title	Payment.Method.Edit.Card.ExpiryDate.Passed
Requirement	The system shall display an alert message if the Card Expiry Date entered by the customer is in the past.
Source	Team Member 1
Rationale	This requirement provides feedback to the user if they enter an expired card expiry date.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 117: Checkout.Cart

Identifier	FR-117
Title	Checkout.Cart

Requirement	The system shall not allow the checkout process if the customer cart is empty.
Source	Team Member 1
Rationale	This ensures that orders cannot be placed without any items.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 118: Checkout.Cart.Product.List

Identifier	FR-118
Title	Checkout.Cart.Product.List
Requirement	The system shall display the list of products in the customer's cart for confirmation during checkout
Source	Team Member 2
Rationale	This allows customers to verify their order before making a purchase.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 119: Checkout.Shipping.Option

Identifier	FR-119
Title	Checkout.Shipping.Option
Requirement	The system shall allow the customer to select shipping options during checkout.
Source	Team Member 1
Rationale	This allows customers to choose their preferred delivery method.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 120: Checkout.Confirm.Order

Identifier	FR-120
-------------------	--------

Title	Checkout.Confirm.Order
Requirement	The system shall allow the customer to confirm the order during checkout.
Source	Team Member 1
Rationale	This allows customers to place their order and finalize their purchase.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 121: Checkout.Select.Payment.Method

Identifier	FR-121
Title	Checkout.Select.Payment.Method
Requirement	The system shall allow the customer to select a payment method during checkout
Source	Team Member 2
Rationale	This allows customers to choose their preferred payment method.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 122: Checkout.Add.Payment.Method

Identifier	FR-122
Title	Checkout.Add.Payment.Method
Requirement	The system shall allow the customer to add a payment method if N/A is available.
Source	Team Member 1
Rationale	This allows customers to add or modify their payment information.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 123: Checkout.Process.Payment.Method

Identifier	FR-123
Title	Checkout.Process.Payment.Method

Requirement	The system shall process the payment from the customer's payment method during checkout.
Source	Team Member 1
Rationale	This ensures that payment is collected securely and accurately.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 124: Checkout.Payment.Method.Not.Successful

Identifier	FR-124
Title	Checkout.Payment.Method.Not.Successful
Requirement	The system shall display an alert if payment is not successful
Source	Team Member 2
Rationale	This informs the customer of a payment failure and allows them to take appropriate action.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 125: Checkout.Payment.Method.Retry

Identifier	FR-125
Title	Checkout.Payment.Method.Retry
Requirement	The system shall allow the customer to retry payment if it fails.
Source	Team Member 1
Rationale	This allows customers to try again if there was a payment failure.
Business Rule	N/A
Dependencies	FR-121
Priority	High

Table 126: Checkout.Confirmation.Email

Identifier	FR-126
Title	Checkout.Confirmation.Email
Requirement	The system shall send an email with the order summary to the customer after the checkout process is complete.

Source	Team Member 2
Rationale	This provides customers with a record of their purchase and allows them to track their order
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 127: Checkout.Notify.Admin

Identifier	FR-127
Title	Checkout.Notify.Admin
Requirement	The system shall notify the admin about the new order.
Source	Team Member 1
Rationale	This informs the admin that a new order has been placed and requires attention.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 128: Checkout.Stock

Identifier	FR-128
Title	Checkout.Stock
Requirement	The system shall update the stock after the order is converted from the cart.
Source	Team Member 1
Rationale	This ensures that inventory levels are accurate and prevents overselling.
Business Rule	N/A
Dependencies	N/A
Priority	High

3.3.6 Module 6: AI Chatbot

Table 129: User Query Submission

Identifier	FR-129
Title	User Query Submission

Requirement	The user shall be able to submit queries or questions through the FelineConnect interface.
Source	Team Member 1
Rationale	Users may have specific questions or require assistance regarding cat breeds, products, or platform functionalities.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 130: Query History and Tracking

Identifier	FR-130
Title	Query History and Tracking
Requirement	The system shall maintain a record of user queries and their corresponding responses for future reference and analysis purposes.
Source	Team Member 2
Rationale	Tracking query history enables the platform to identify recurring inquiries, trends, and areas for improvement, ultimately enhancing the quality of user support and system performance.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 131: Chatbot

Identifier	FR-131
Title	Chatbot
Requirement	The system shall provide a chatbot interface for customers to interact with.
Source	Team Member 1
Rationale	Customers can interface with a chatbot.

Business Rule	N/A
Dependencies	N/A
Priority	High

Table 132: Chatbot.Available

Identifier	FR-132
Title	Chatbot.Available
Requirement	The system shall check that the Chatbot is available.
Source	Team Member 2
Rationale	To ensure that the Chatbot is available for use.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 133: Chatbot.Available.No

Identifier	FR-133
Title	Chatbot.Available.No
Requirement	The system shall show an alert message if the Chatbot is not available.
Source	Team Member 2
Rationale	To ensure that customers is informed when the Chatbot is not available.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 134: Chatbot.Search.Field

Identifier	FR-134
Title	Chatbot.Search.Field
Requirement	The system shall display a textbox for customers to enter their queries.
Source	User
Rationale	To enable customers to enter their queries, the system should display a textbox.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 135: Chatbot.Search.Response

Identifier	FR-135
Title	Chatbot.Search.Response
Requirement	The system shall process the customer's query and generate a response.
Source	Team Member 2
Rationale	To provide responses to customer queries, the system should process the query and generate a response.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 136: Chatbot.UnableToUnderstand

Identifier	FR-136
Title	Chatbot.UnableToUnderstand
Requirement	The system shall handle situations where the chatbot is unable to understand the customer's query by showing a message asking the customer to rephrase the question.
Source	Team Member 2
Rationale	To handle situations where the Chatbot is unable to understand a query, the system should ask the customer to rephrase the question.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 137: Chatbot.Cancel

Identifier	FR-137
Title	Chatbot.Cancel
Requirement	The system shall terminate the process if the customer cancels interacting with the chatbot.
Source	Team Member 1
Rationale	To terminate the process when an customer cancels interacting with the Chatbot.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 138: Chatbot.Techical.Issue

Identifier	FR-138
Title	Chatbot.Techical.Issue
Requirement	The system shall show an alert if any technical issues are encountered with chatbot.
Source	Team Member 2
Rationale	To notify the customer about the status of chatbot.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 139: Customer.Write.Message

Identifier	FR-139
Title	Customer.Write.Message
Requirement	The system shall provide a text box for the customer to write a message to the admin.
Source	Team Member 2
Rationale	To allow customers to communicate with the admin and seek product-related queries.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 140: System.Alert.TextLimit

Identifier	FR-140
Title	System.Alert.TextLimit
Requirement	The system shall alert the customer if the message text exceeds 255 characters.
Source	Team Member 2
Rationale	To ensure customers do not exceed the message character limit.
Business Rule	N/A
Dependencies	N/A
Priority	High

3.3.7 Module 7: Admin Panel

Table 141: Admin.Notification

Identifier	FR-141
Title	Admin.Notification
Requirement	The system shall notify the admin of the customer message.
Source	Team Member 2
Rationale	To ensure the admin is aware of customer queries and can respond accordingly.
Business Rule	N/A
Dependencies	FR-136
Priority	High

Table 142: Super.Admin.Login

Identifier	FR-142
Title	Super.Admin.Login
Requirement	The system shall check that super admin is logged in to the system before adding a new admin.
Source	Team Member 2
Rationale	To ensure only authorized users can add admins
Business Rule	None
Dependencies	None
Priority	High

Table 143: Super.Admin.Permission

Identifier	FR-143
Title	Super.Admin.Permission
Requirement	The system shall check that the super admin has the permission to add a new admin.
Source	Team Member 2
Rationale	To maintain proper access control
Business Rule	None
Dependencies	None
Priority	High

Table 144: New.Admin

Identifier	FR-141
Title	New.Admin
Requirement	The system shall add the new Admin to the system when the Super Admin submits the information.
Source	Team Member 1
Rationale	To create the new admin account
Business Rule (if required)	None
Dependencies	None
Priority	High

Table 145: Customer.Authenticate

Identifier	FR-145
Title	Customer.Authenticate
Requirement	The system shall validate the Custmer's entered credentials
Source	Team Member 1
Rationale	The system verifies that the entered credentials are valid and exist in the system.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 146: Admin.Authenticate.SystemError

Identifier	FR-146
Title	Admin.Authenticate.SystemError
Requirement	The system shall prompt the Admin if the system is down.
Source	Team Member 1
Rationale	The system should respond appropriately if the system is down or under maintenance.
Business Rule	N/A
Dependencies	N/A

Priority	High
-----------------	------

Table 147: Admin.Faq.View

Identifier	FR-147
Title	Admin.Faq.View
Requirement	System shall allow the admin to view all faqs.
Source	Team Member 2
Rationale	Faqs are a list of questions and answers that are visible to customers. Admin can view a list of all active faq that are shown on the customer side.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 148: Admin.Faq.View.Empty

Identifier	FR-148
Title	Admin.Faq.View.Empty
Requirement	System shall notify the admin if there are no faqs to show.
Source	Team Member 2
Rationale	Notifying the admin if the list of faqs are empty.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 149: Admin.Faq.Add.Privilege

Identifier	FR-149
Title	Admin.Faq.Add.Privilege
Requirement	System shall verify that the admin has the right to manage FAQs.
Source	Team Member 2
Rationale	To restrict unauthorized access to FAQs.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 150: Admin.Faq.Add.Answer

Identifier	FR-150
Title	Admin.Faq.Add.Answer
Requirement	System shall allow the admin to add an answer.
Source	Team Member 1
Rationale	The answer is the admin response to the frequently asked question available in the FAQ section.
Business Rule	N/A
Dependencies	FR-146
Priority	High

Table 151: Cat Food Inventory Management

Identifier	FR-151
Title	Cat Food Inventory Management
Requirement	The admin shall be able to view and manage the cat food inventory in the online store, including adding new products, updating existing ones, and removing discontinued items.
Source	Team Member 1
Rationale	To ensure that the online store's inventory is accurately maintained, reflecting the availability of cat food products for purchase.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 152: Admin.MonitorSystemHealth

Identifier	FR-152
Title	Admin.MonitorSystemHealth
Requirement	System shall show an overview of the current system health status.
Source	Team Member 1
Rationale	System health status can be viewed by admin to know the current load on system, error rates, response times, and other customer requests to the system.
Business Rule	N/A
Dependencies	N/A
Priority	High

Table 153: Admin.MonitorSystemHealth.Cancel

Identifier	FR-153
Title	Admin.MonitorSystemHealth.Cancel
Requirement	System shall allow admin to return to dashboard.
Source	Team Member 1
Rationale	Once admin has viewed system health they can then return to their dashboard for other activities.
Business Rule	N/A
Dependencies	N/A
Priority	High

3.3.8 Module 8: Statistical Analytics

Table 154: Generating.Reports

Identifier	FR-154
Title	Generating.Reports
Requirement	The system should generate the statistical report based on the user-selected time range, providing relevant data such as charts, graphs, tables, and other relevant information.
Source	Team Member 2
Rationale	To allow the user to generate report in specific format
Business rule	Generate the accurate report
Dependencies	N/A
Priority	High

Table 155: Data.selection

Identifier	FR-155
Title	Data.selection
Requirement	The user should be able to select the type of data to be included in the report.
Source	Team Member 2
Rationale	Help system to show specific data such as for inventory, cattle, milk
Business rule	N/A

Dependencies	N/A
Priority	High

Table 156: Report.customization

Identifier	FR-156
Title	Report.customization
Requirement	The user should be able to customize the report by choosing different colors, fonts, and layout options.
Source	Team Member 2
Rationale	This will help the user to collect data in different options
Business rule	N/A
Dependencies	N/A
Priority	Low

Table 157: User.access.control

Identifier	FR-157
Title	User.access.control
Requirement	The app should have appropriate user access control mechanisms to ensure that only authorized users can generate reports and view sensitive data.
Source	Team Member 2
Rationale	This will help the system to secure sensitive data.
Business rule	N/A
Dependencies	N/A
Priority	Low

Table 158: Report.Sharing

Identifier	FR-158
Title	Report.Sharing
Requirement	The app should allow users to share the reports with others via email, messaging, or social media.
Source	Team Member 2
Rationale	This will help the system to share our report with our team members.
Business rule	Report sharing will be accurate and in best quality.
Dependencies	N/A
Priority	Medium

Table 159: Report.Automation

Identifier	FR-159
Title	Report.Automation
Requirement	The app should allow users to schedule the reports to be generated automatically at specific intervals, such as daily, weekly, or monthly, and send them to specific recipients.
Source	Team Member 2
Rationale	This will help user to generate report automatically.
Business rule	N/A
Dependencies	N/A
Priority	Medium

Table 160: Graph.Types

Identifier	FR-160
Title	Graph.Types
Requirement	The feature should allow users to select from a variety of graph types such as line graph, bar graph, pie chart that are relevant for the type of data being presented.
Source	Team Member 2
Rationale	This will help user to generate report graph according to his circumstances
Business rule	Only select those Graph Types, which are in system
Dependencies	N/A
Priority	Medium

Table 161: Data.Filtering

Identifier	FR-161
Title	Data Filtering

Requirement	The feature should allow users to filter the data used to generate the graph, based on specific criteria such as date ranges or specific values.
Source	Team Member 2
Rationale	This will help the user to generate a graph on specific criteria.
Business rule	Filter the data by given tags.
Dependencies	N/A
Priority	Medium

Table 162: Exporting

Identifier	FR-162
Title	Exporting
Requirement	The feature should allow users to export the graph in a variety of formats, such as PNG, JPEG, or PDF, for sharing or inclusion in other documents.
Source	Team Member 2
Rationale	This will help user to export graph in specific format.
Business rule	Export file only in given format.
Dependencies	FR-99
Priority	Medium

Table 163: Graph.Interaction

Identifier	FR-163
Title	Graph.Interaction

Requirement	Users should be able to interact with the graph by selecting data points, highlighting different sections, and toggling on and off certain variables
Source	Team Member 2
Rationale	This will help user to export graphs on specific circumstances.
Business rule	N/A
Dependencies	N/A
Priority	Medium

3.4 Non-Functional Requirements

3.4.1 NFR-1: Reliability

Identifier	NFR-1
Title	Reliability
Availability	System is online all the time 24/7 except in case of some update or error on server side.
Accuracy	The system will show up to date and accurate information to users. Exact accuracy rate can be given after implementation
Error Handling	If there is error processing the image, you just need to refresh the application and wait for 5-6 seconds, and application will run smooth again.

3.4.2 NFR-2: Usability

Identifier	NFR-2
Title	Usability
Adaptability	The interface will have options that are easy to understand and naturally mapped. It will follow Design standards and conventions of commonly used applications so user can adapt easily

Learnability	A person with basic grasp on language can understand. 97% of users will be able to use efficiently after 10 minutes of initial use with 0.5% error rate.
Design	The styling will be clean, and options will be clearly readable. Font style will be clear
Guidance	A help option will direct user to detailed explanation of each aspect of product and application
Visibility	Notification messages will be made clearly visible to user in a seamless way that does not disturb their other activities.

3.4.3 NFR-3: Performance

Identifier	NFR-3
Title	Performance
Response Time	Our system is real-time, it will have fast response time. It will provide a response in max 2-3 minutes. The speed depends on network connection strength as well.
Throughput	The application can perform navigation with streaming or cleaning. The processes run in parallel.

3.4.4 NFR-4: Portability

Identifier	NFR-4
Title	Portability
Platform independent	Our system is tailored for web-based systems, ensuring compatibility across different platforms and responsive performance.

3.5 External Interface Requirements

3.5.1 User Interfaces Requirements

UI-1: The system's interface shall abide by vanilla CSS and Bootstrap UI design principles.

UI-2: The system shall use JPEG and PNG vector graphics for icons.

UI-3: Every popup, dialog, or page of the system shall let the user easily navigate to previous page, by providing backward navigation buttons.

UI-4: The system shall use a consistent coloring scheme for user interface components throughout the app.

UI-5: The size of UI elements shall be stated in screen size percentage.

UI-6: The system UI layout shall be highly responsive.

3.5.2 Software interfaces

6.2.1	Canva will be used for creating mockups of the application
6.2.2	React.js will be used for the front-end development
6.2.3	Node Js will be used for backend development
6.2.4	MySQL will be used for database
6.2.5	Keras and TensorFlow will be used for model training

3.5.3 Hardware interfaces

HI-1: As the application requires internet to run, all the components related to it can be considered as hardware components of the system such as modem, WAN, Ethernet etc.

3.5.4 Communications interfaces

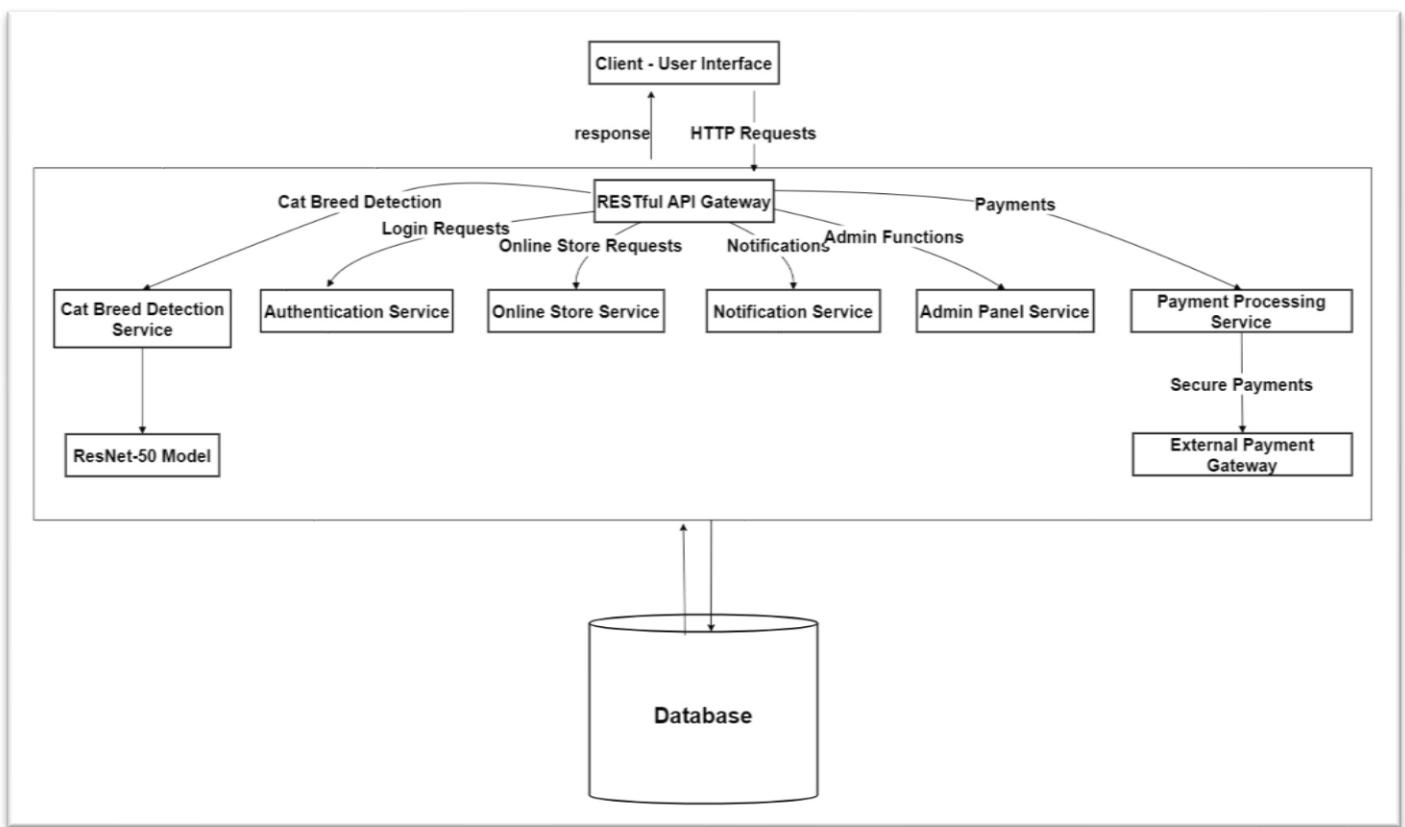
6.4.1 The system will be sending verification links through emails.

6.4.2 Our system will use the HTTP protocol to communicate with node server.

4 Design and Architecture

4.1 Architectural Design

FelineConnect's architecture follows a RESTful design, facilitating seamless communication between its components. The User Interface (UI) interacts with the backend Application Logic through RESTful APIs, enabling actions like user authentication, breed detection, and product browsing. The Application Logic further communicates with the AI Model for breed analysis and the Database for data storage and retrieval. This modular approach promotes flexibility, scalability, and maintainability while ensuring efficient data exchange and processing within the FelineConnect ecosystem.



4.2 Design Models

Following are the design models for FelineConnect application.

4.2.1 Activity Diagrams

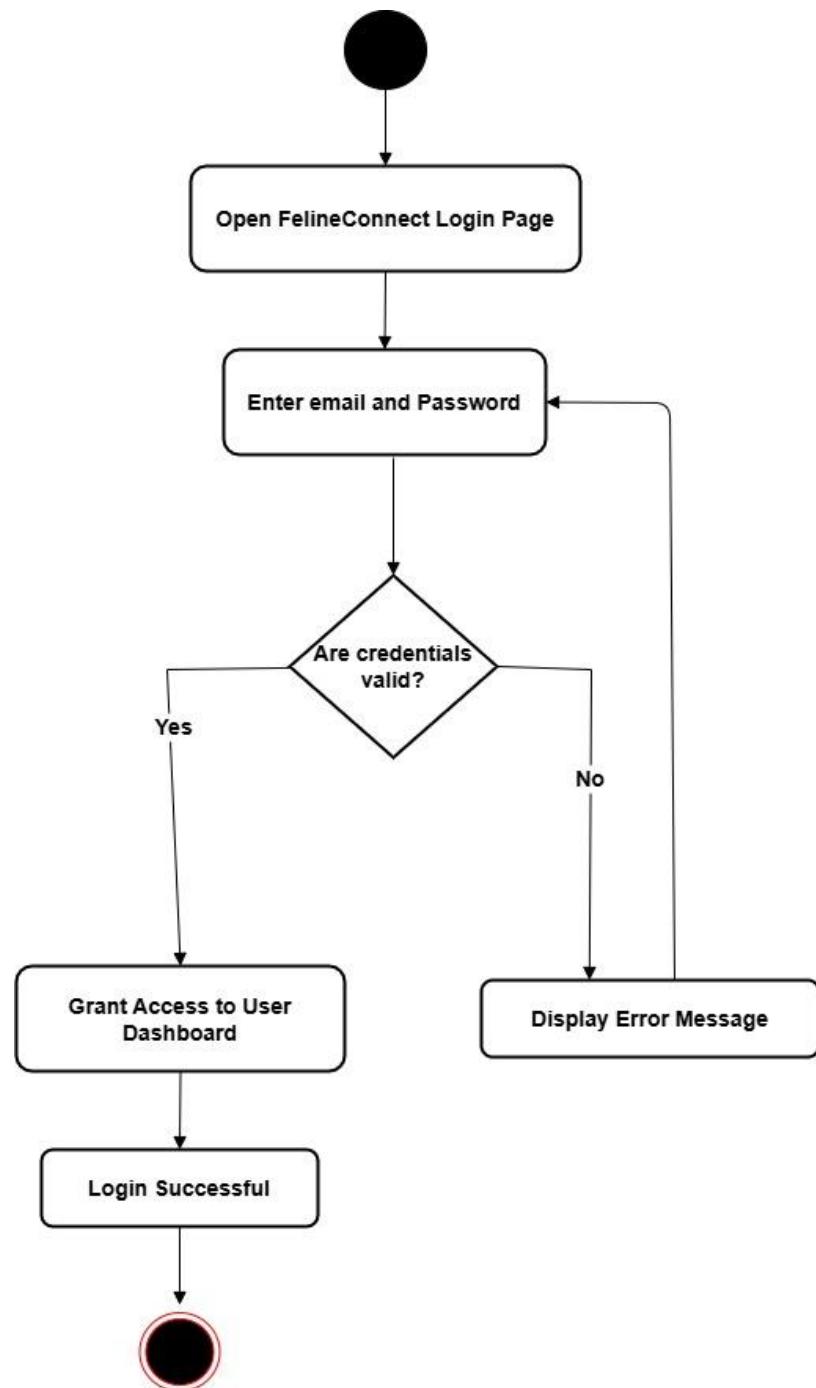


Figure 2: Activity diagram for login process

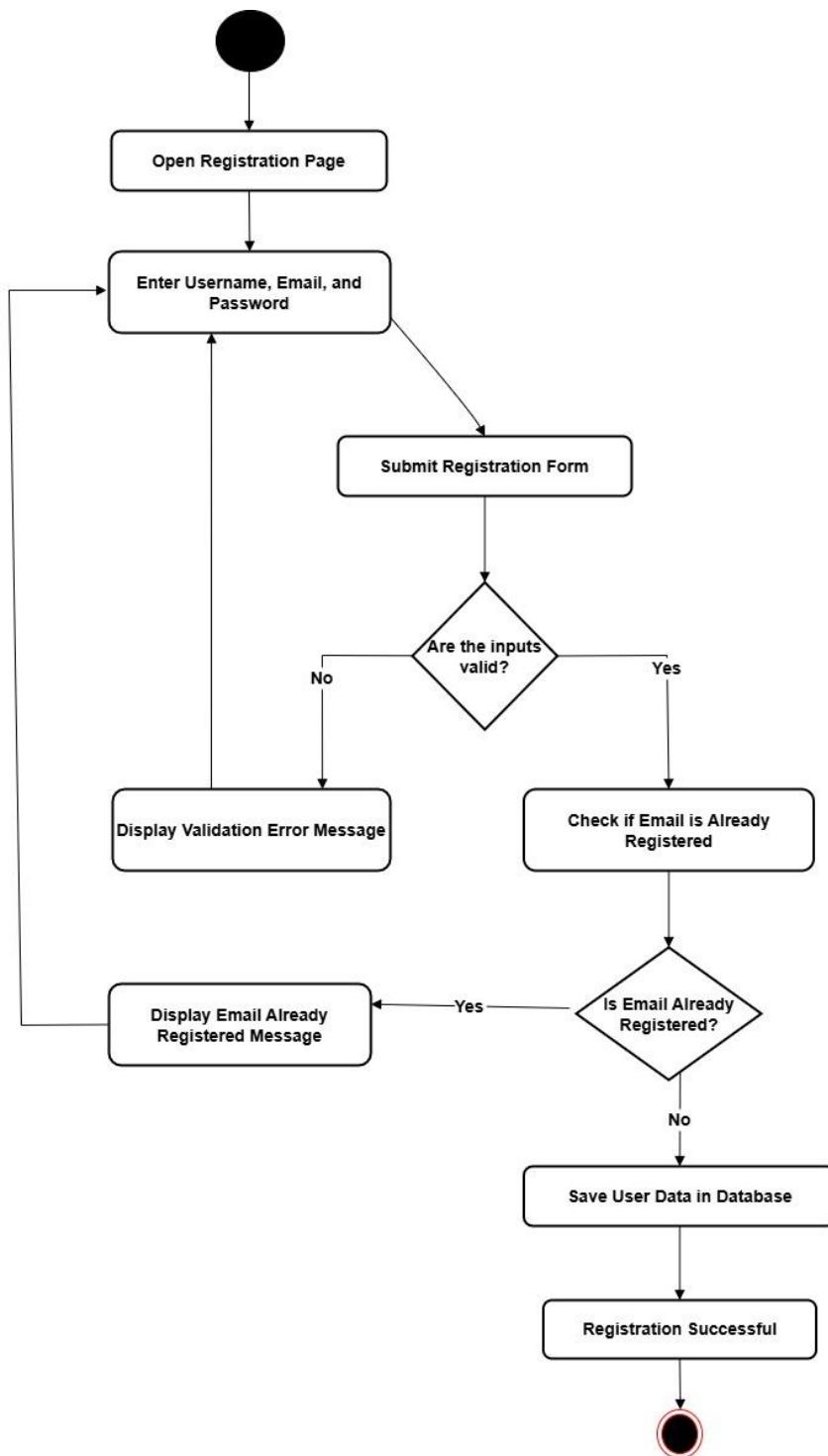


Figure 3: Activity diagram for Registration process

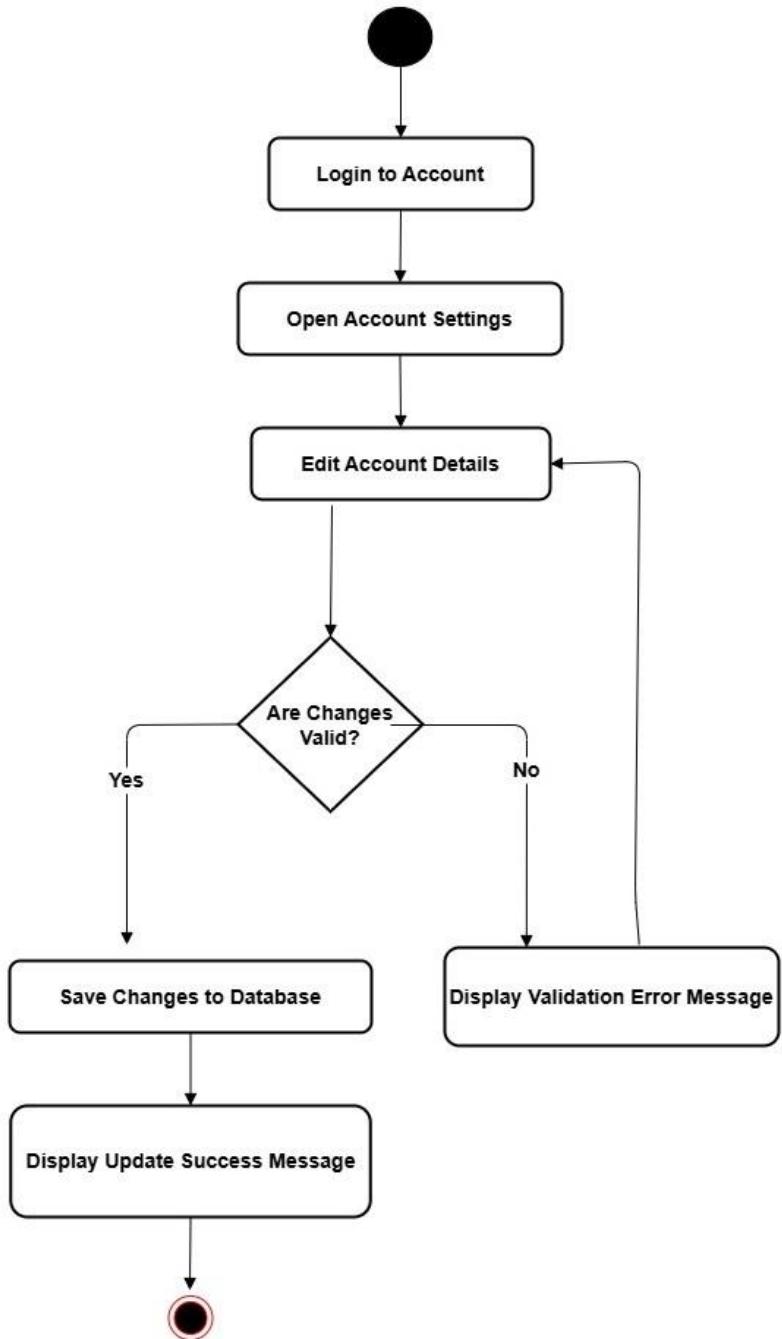


Figure 4: Activity diagram for edit profile

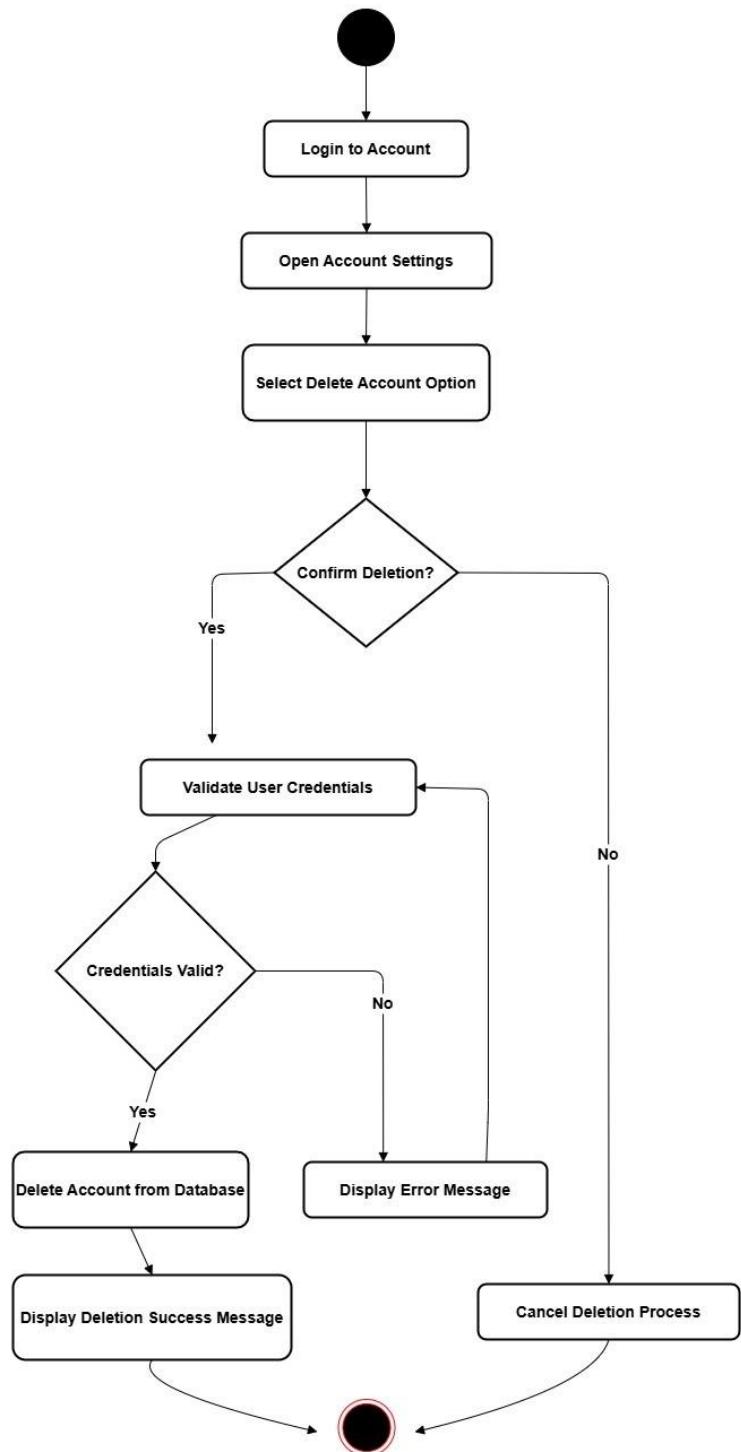


Figure 5: Activity diagram for delete account process

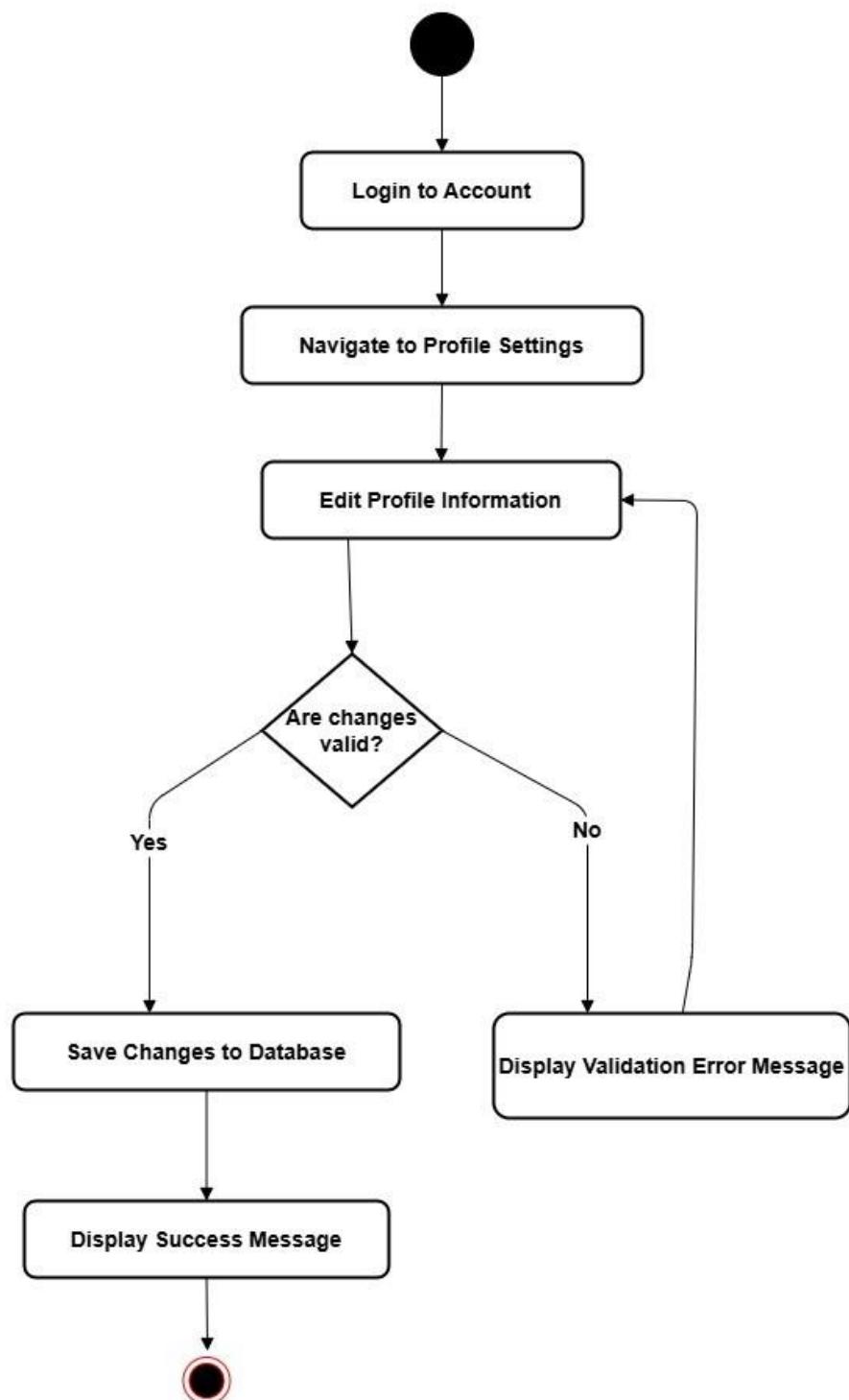


Figure 6: Activity Diagram for Edit Profile

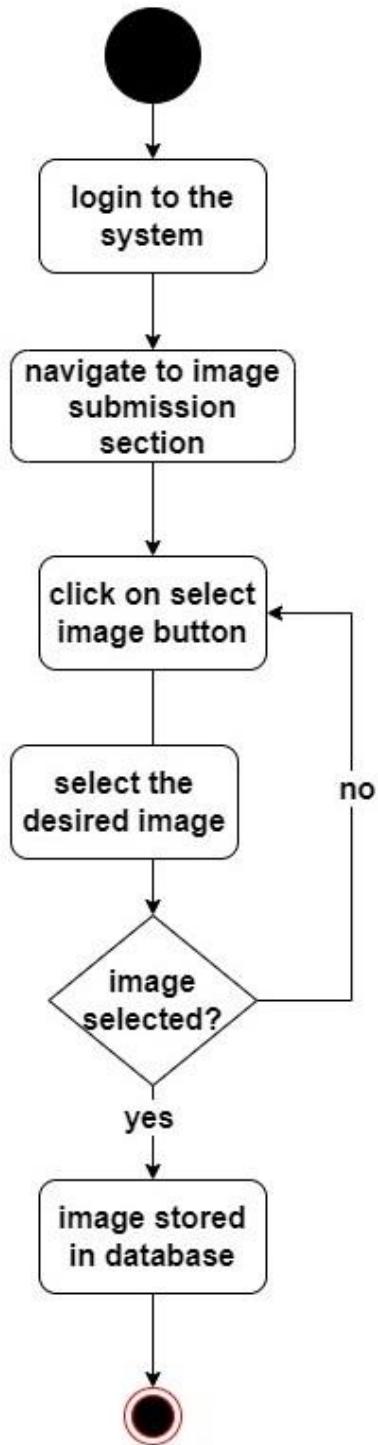


Figure 7: Activity Diagram for Upload image process

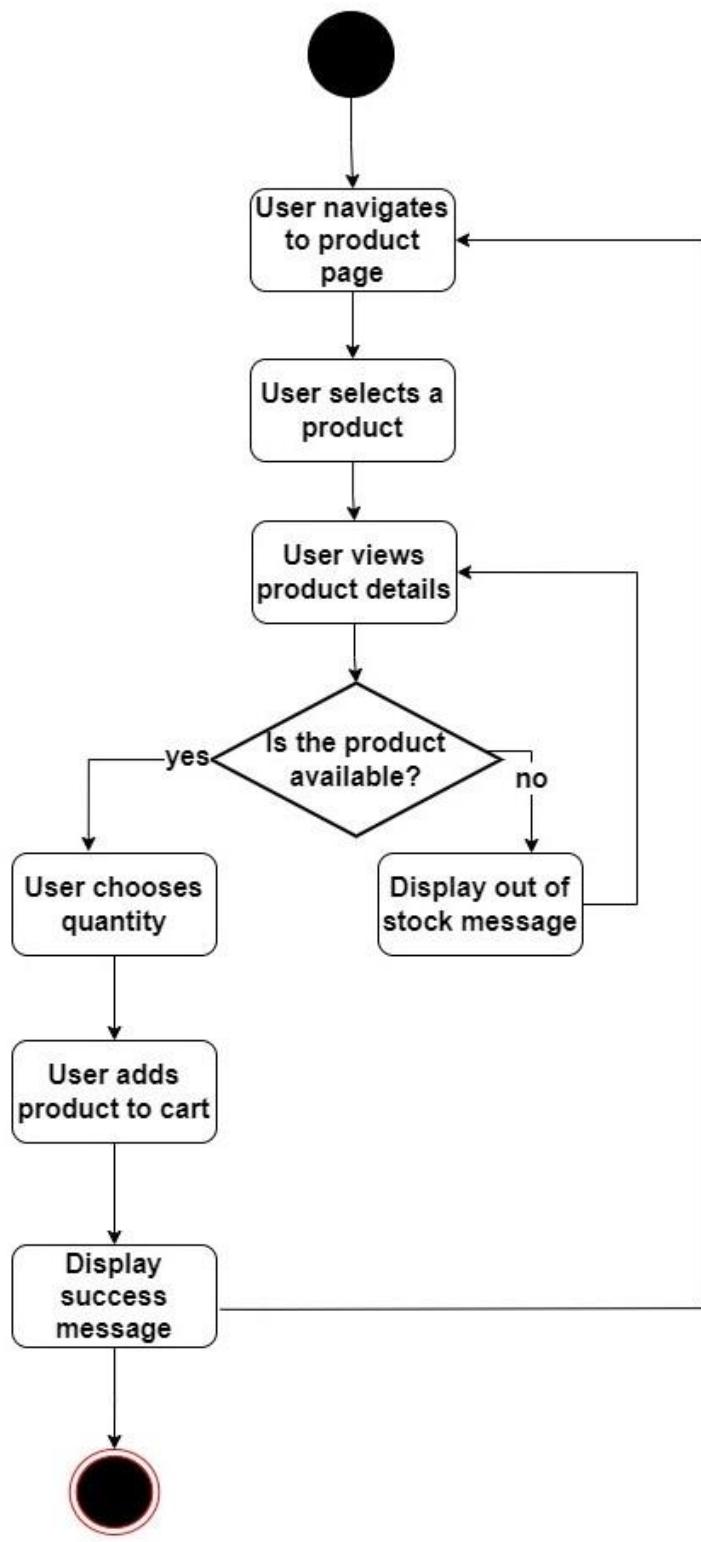


Figure 8: Activity Diagram for Add Product to Cart Process

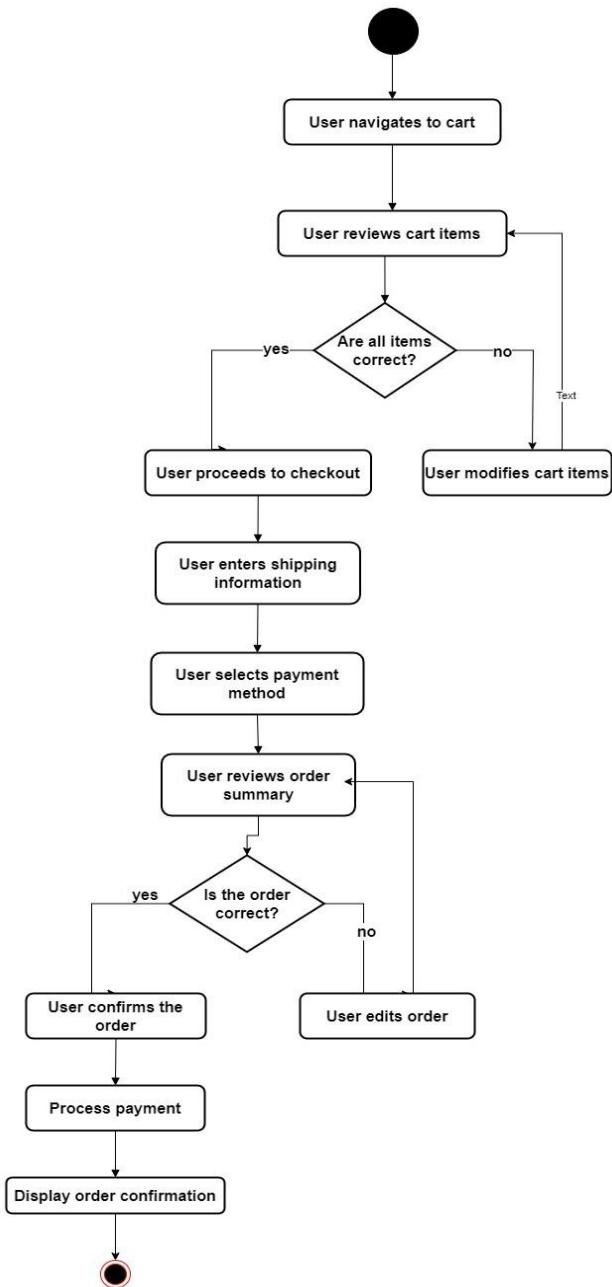


Figure 9: Activity Diagram for Checkout Process

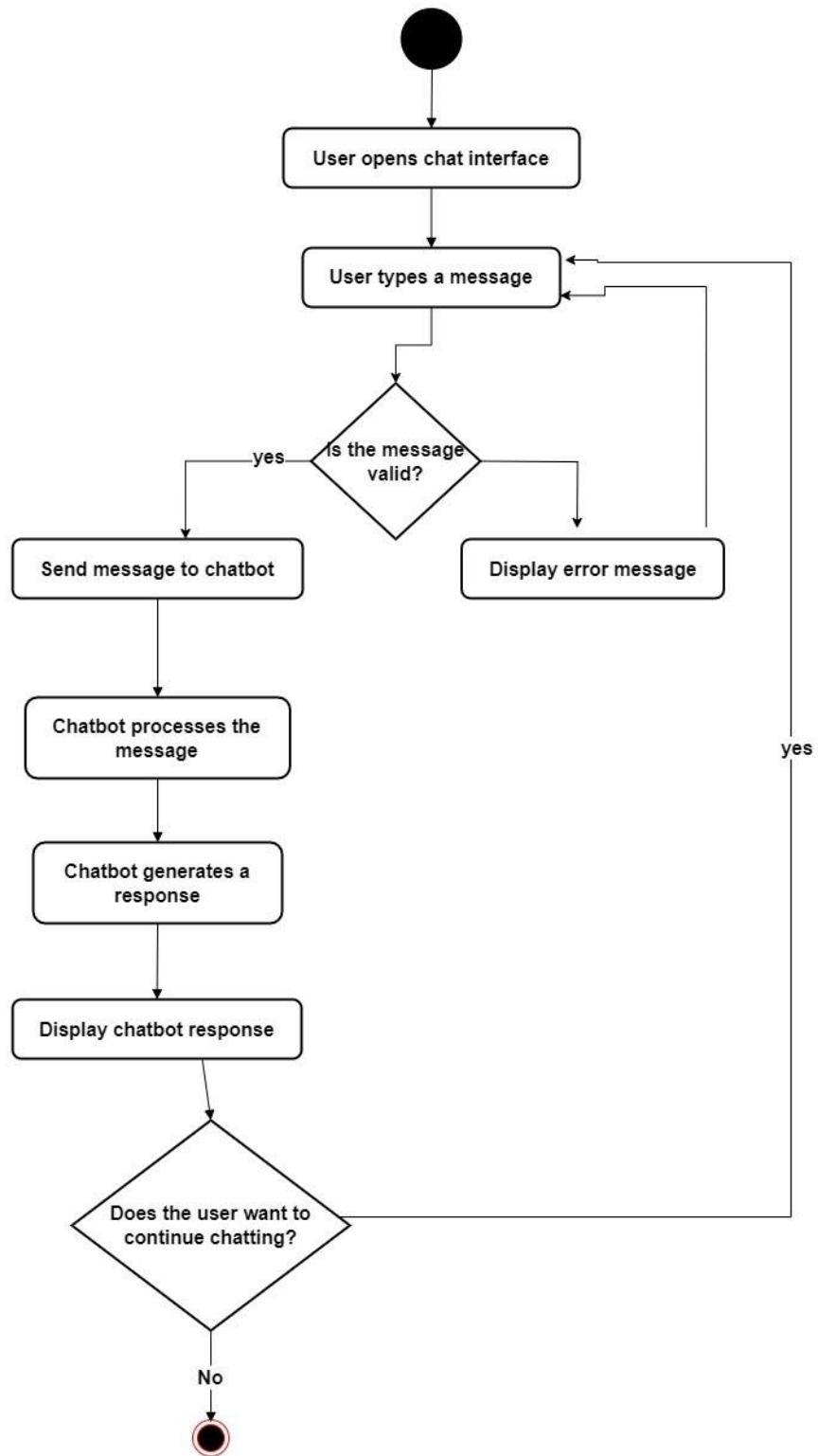


Figure 10: Activity Diagram for Send Message to Chatbot

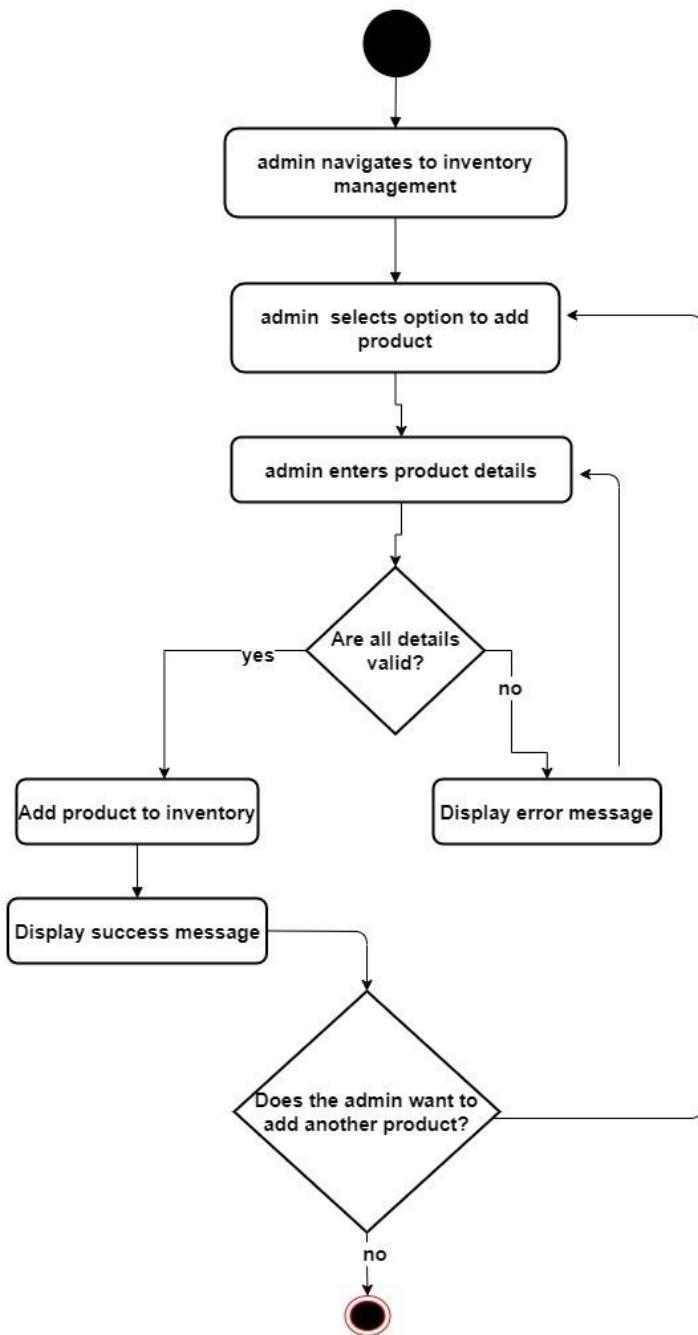


Figure 11: Activity Diagram for Add a Product to inventory process

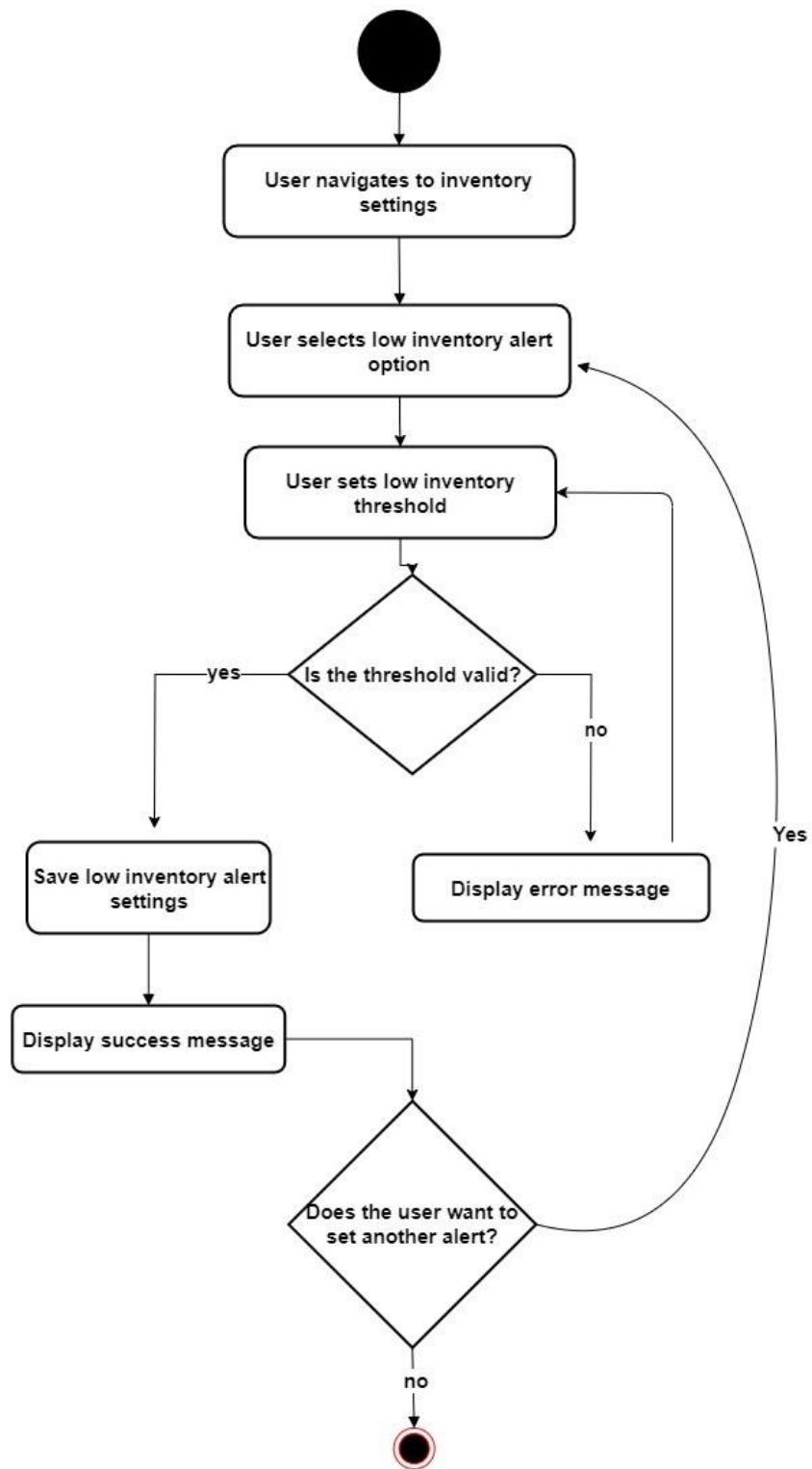


Figure 12: Activity Diagram for set alert for low inventory process

4.2.2 Class Diagram

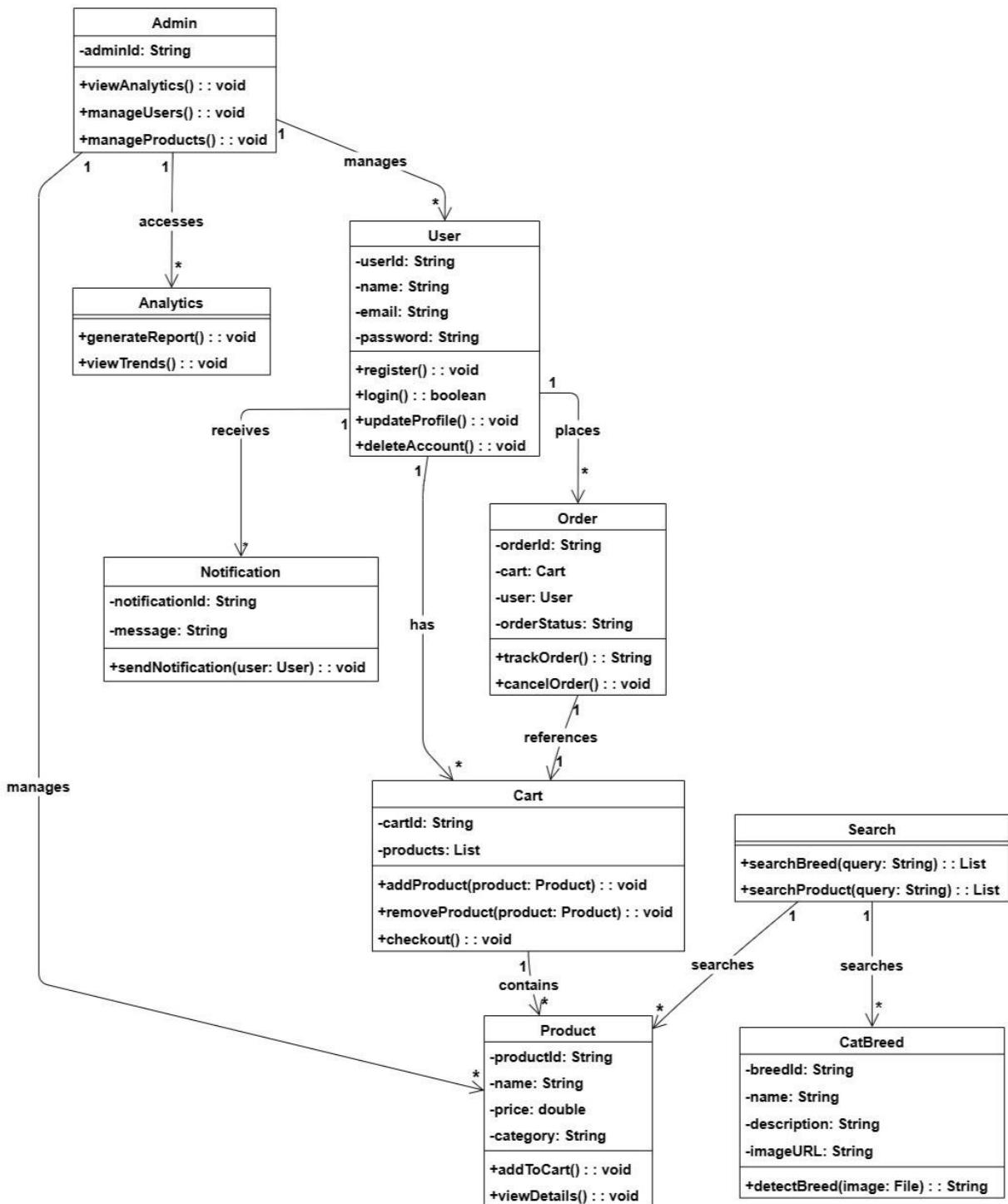
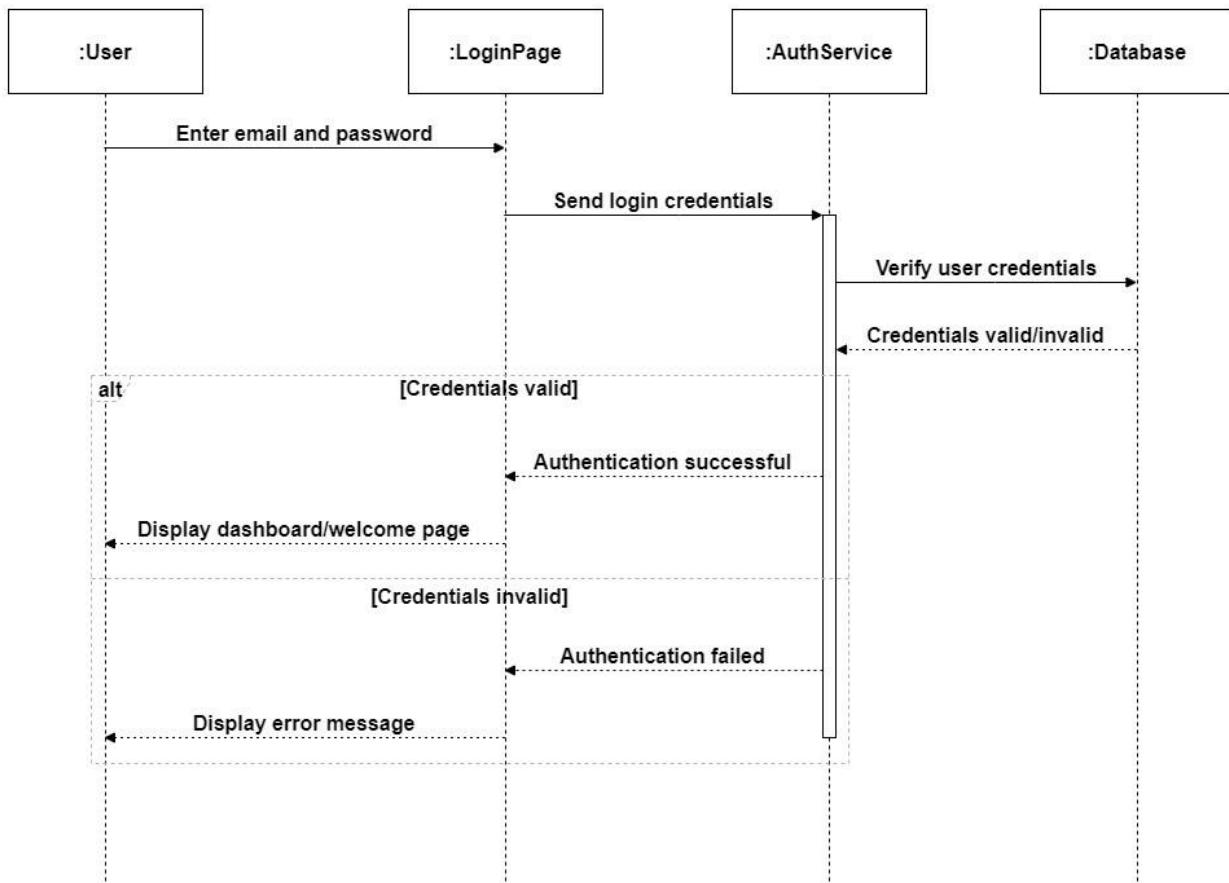


Figure 13: Class diagram for FelineConnect project



4.2.3 Sequence Diagram

Figure 14: Sequence Diagram for Login

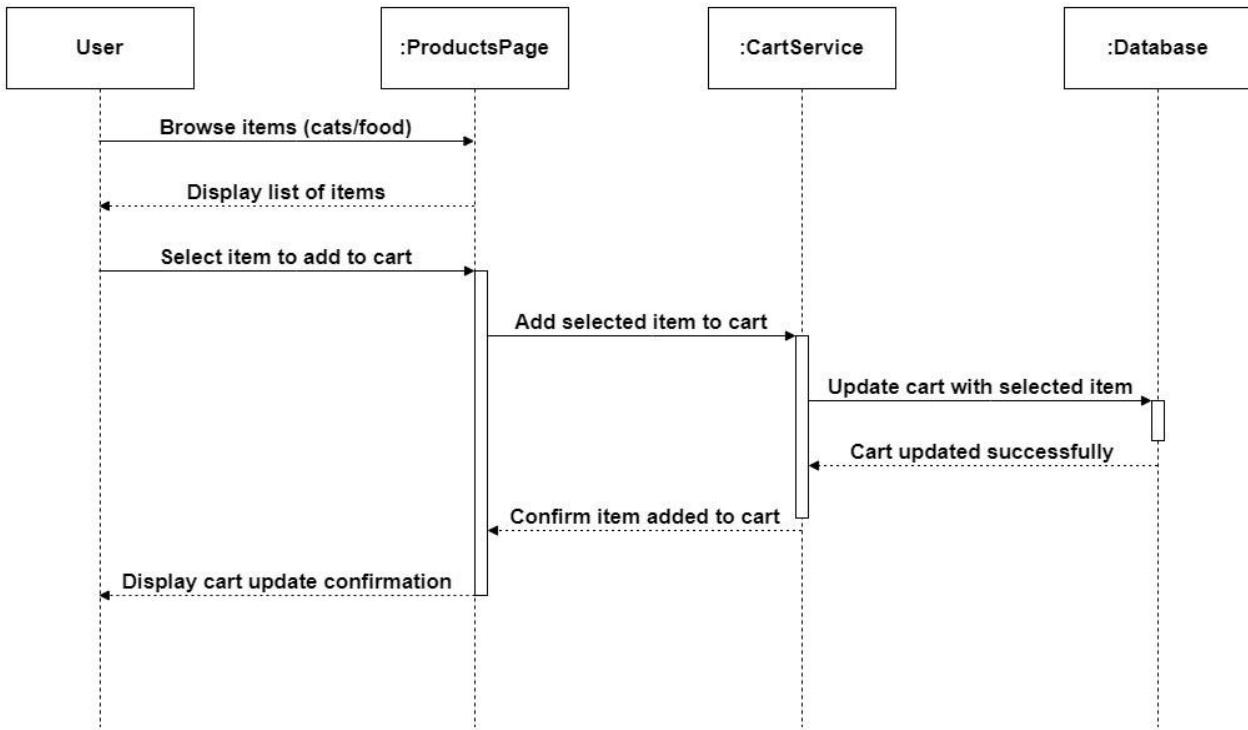


Figure 15: Sequence Diagram for adding items (cats or food) to the cart

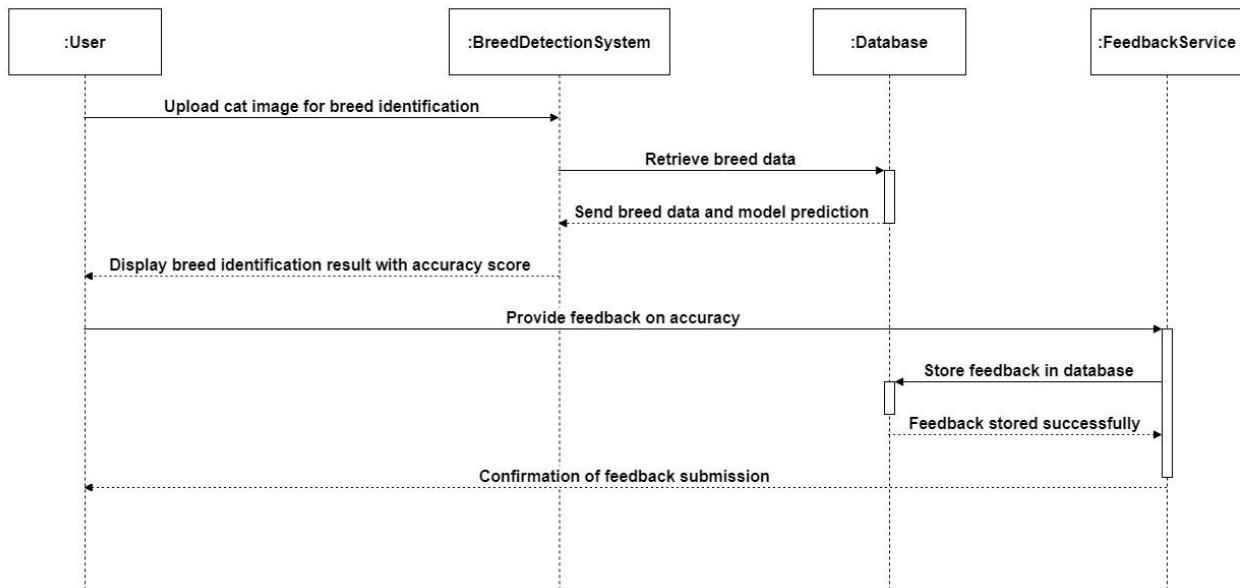


Figure 16: Sequence Diagram for breed detection result and feedback

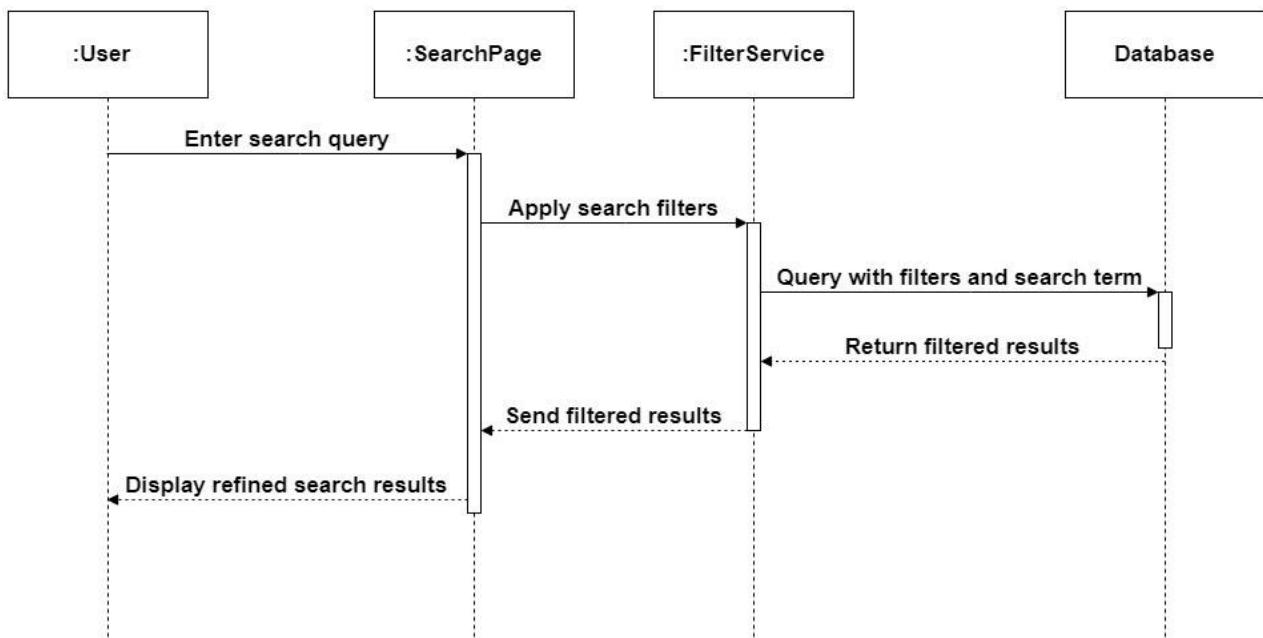


Figure 17: Sequence Diagram for how users interact with search filters to refine results

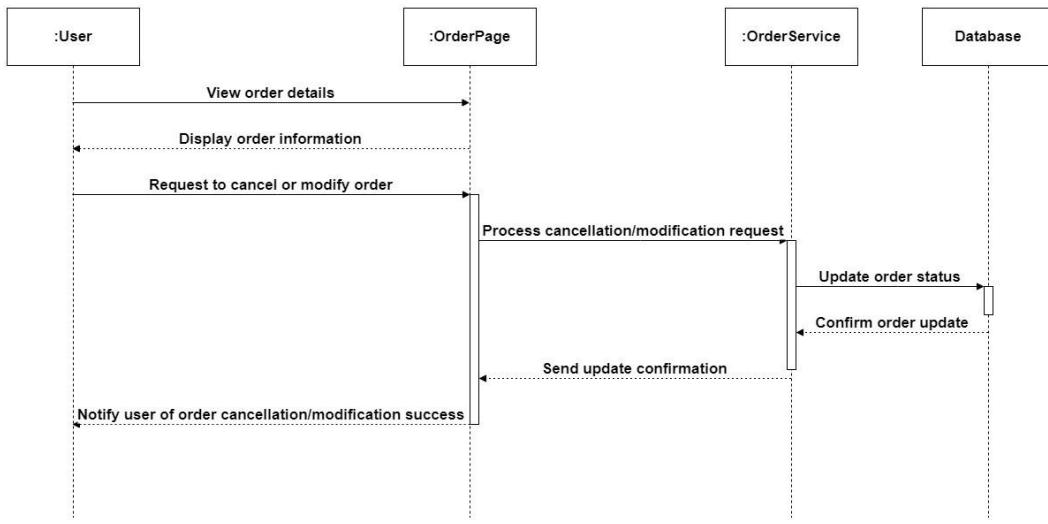


Figure 18: Sequence Diagram for handling order cancellations or modifications

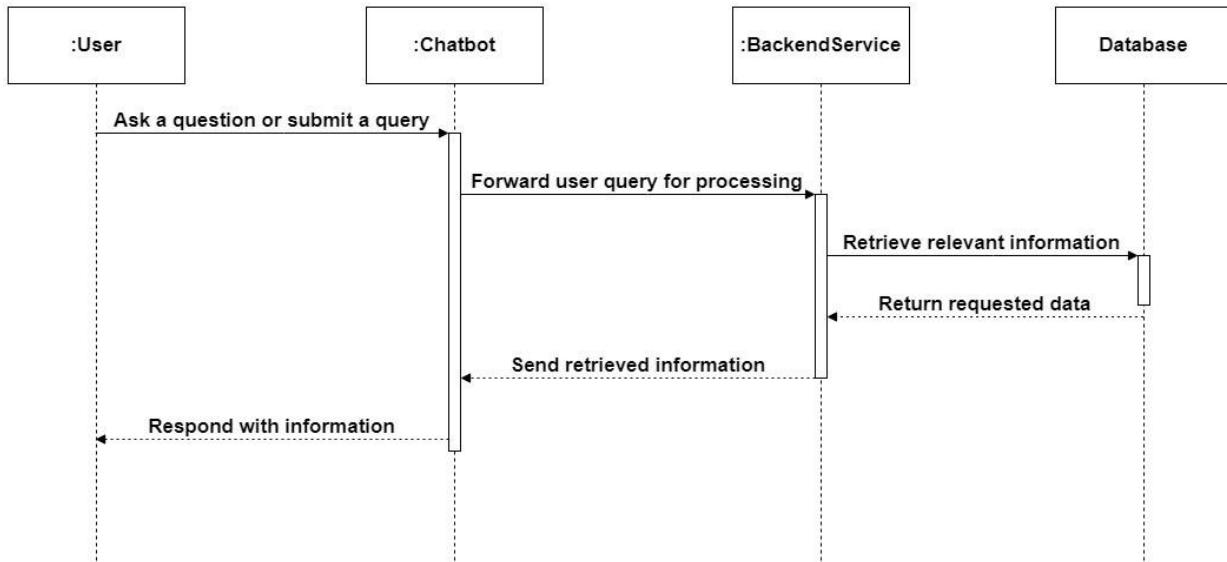


Figure 19: Sequence Diagram for communication between the chatbot and backend system

4.2.4 State Transition Diagram

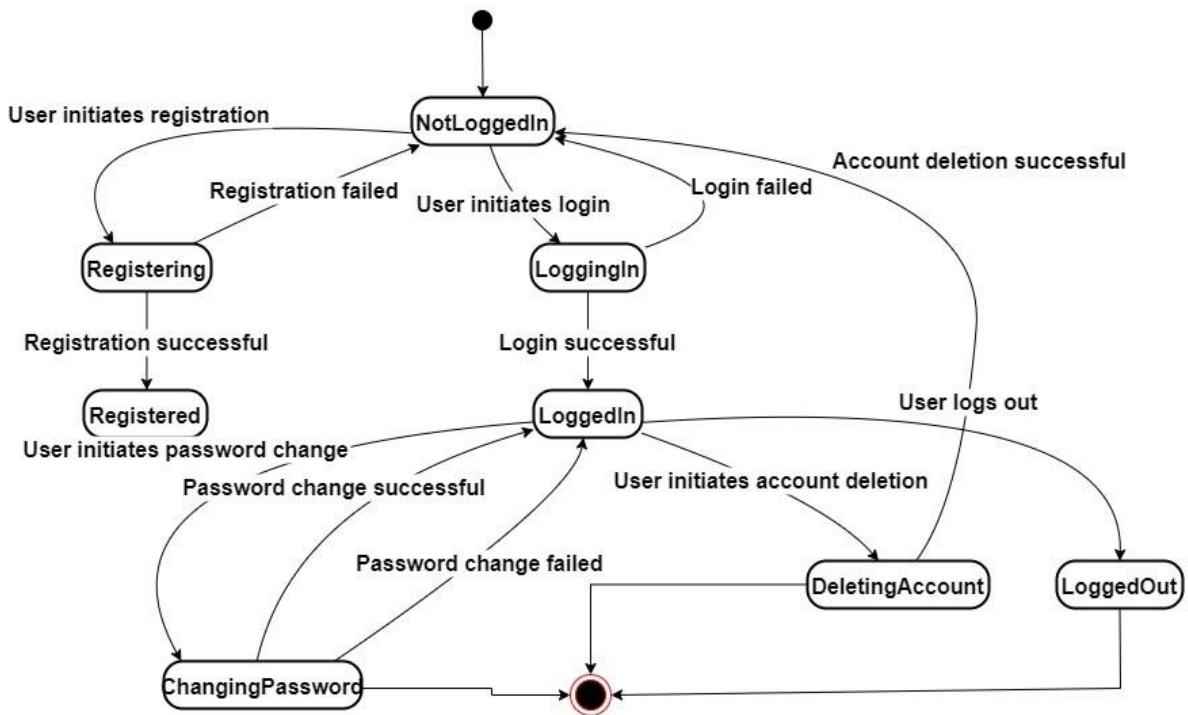


Figure 20: State transition diagram for user authentication

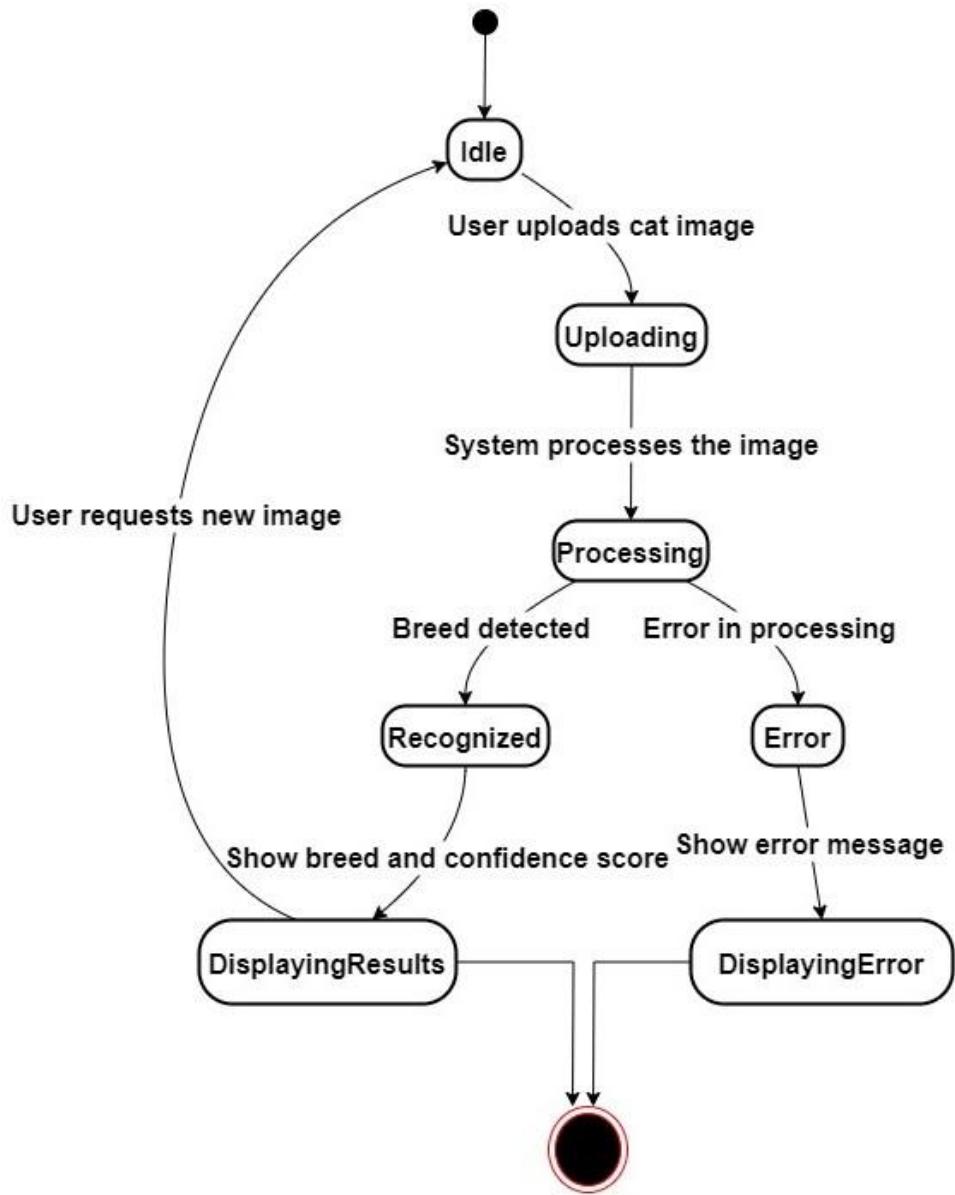


Figure 21: State transition diagram for breed detection

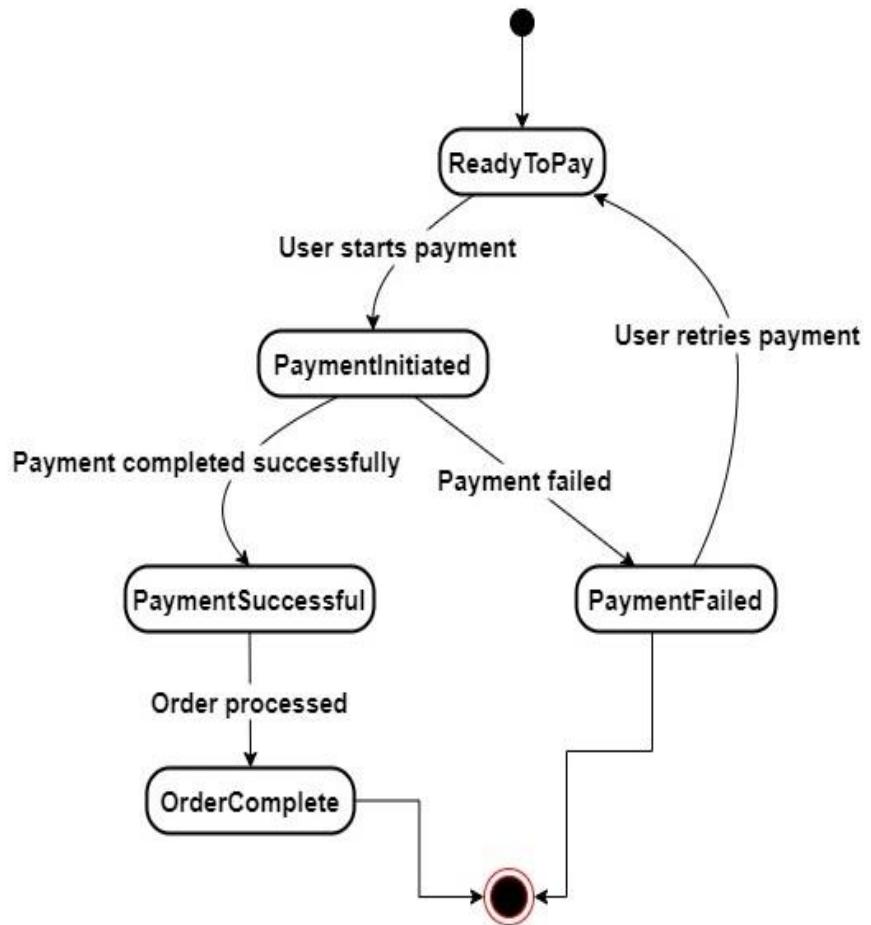


Figure 22: State transition diagram for payment

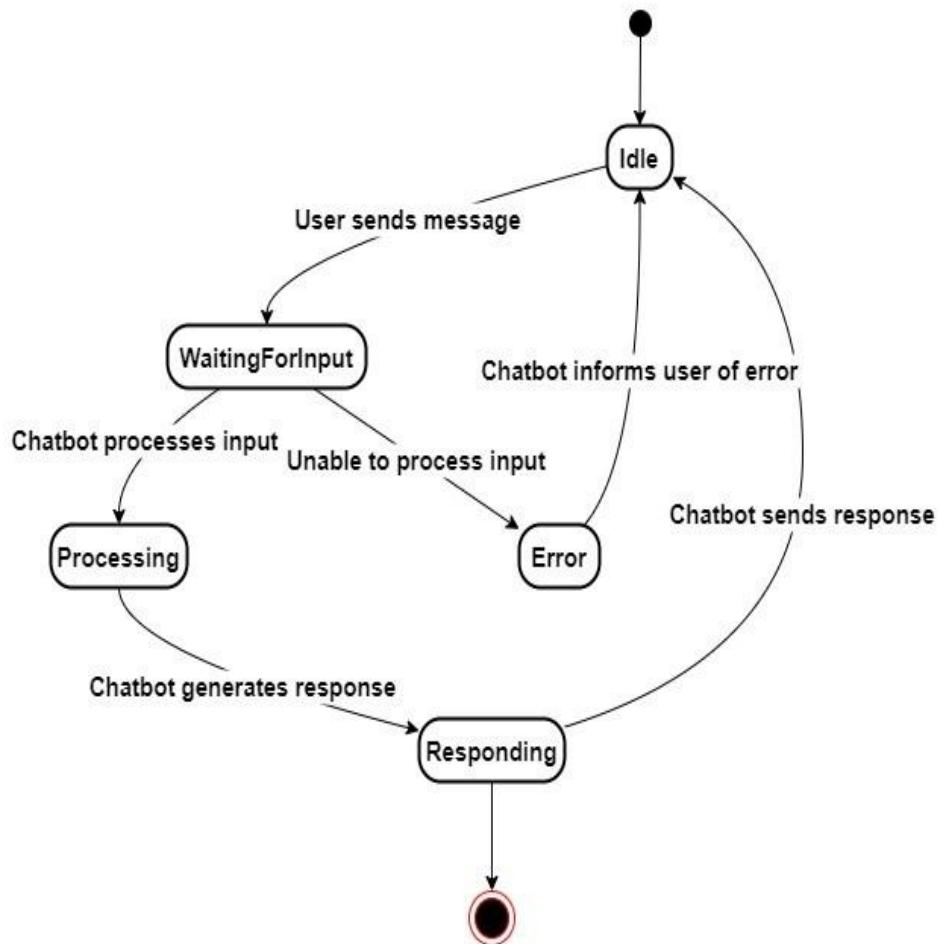


Figure 23: State transition diagram for AI chatbot

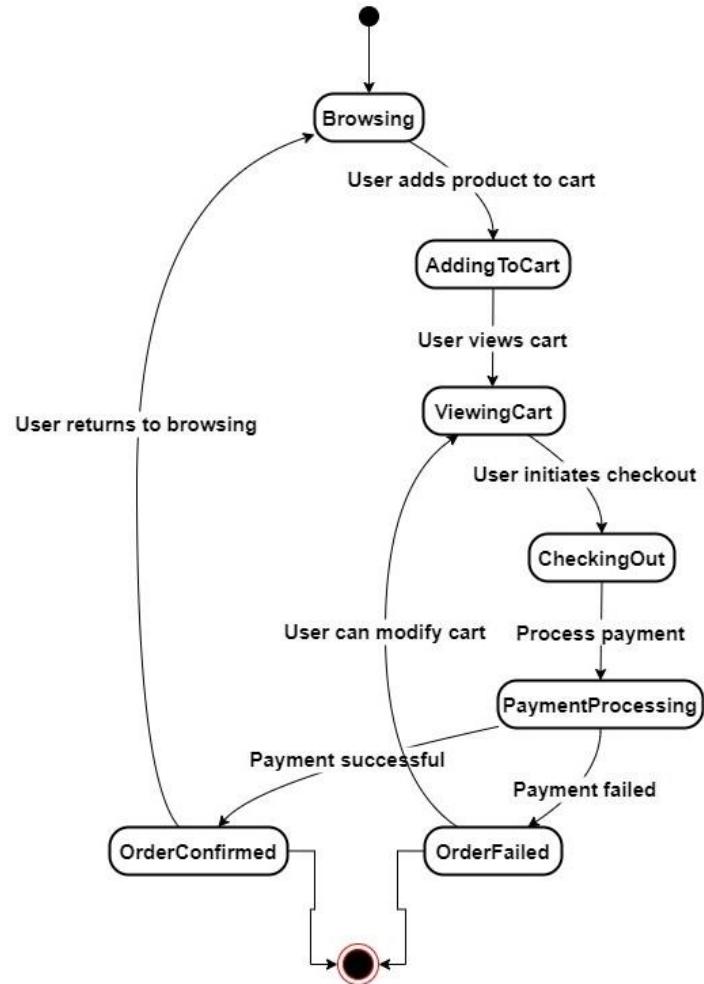


Figure 24: State transition diagram for Online Store

4.3 Data Design

4.3.1 Data Dictionary

4.3.1.1 Table 1: User

Entity	Attributes	Data Type	Description
User	UserID	String	Unique identifier for each user
	Name	String	User's full name
	Email	String	User's email address
	Password	String	User's login password
	ProfilePhoto	String	URL of the user's profile photo
	Bio	String	Brief biography of the user
	Phone	Int	User's mobile number

4.3.1.2 Table 2: Cat Breed Detection

Entity	Attributes	Data Type	Description
CatBreedDetection	DetectionID	String	Unique identifier for the detection
	UserID	String	ID of the user uploading the image
	ImageURL	String	URL of the uploaded image
	DetectedBreed	String	Identified breed of the cat
	ConfidenceScore	Double	Confidence score for breed recognition
	Feedback	String	User feedback on breed detection accuracy

4.3.1.3 Table 3: Online Store

Entity	Attributes	Data Type	Description
Product	ProductID	String	Unique identifier for each product
	Name	String	Name of the product
	Description	String	Description of the product
	Price	Double	Price of the product
	Quantity	Int	Available quantity of the product
	Category	String	Category of the product
	Ingredients	List	List of ingredients for the product

4.3.1.4 Table 4: Shopping Cart

Entity	Attributes	Data Type	Description
Cart	CartID	String	Unique identifier for each shopping cart
	UserID	String	ID of the user owning the cart
	Products	List	List of product IDs in the cart
	TotalPrice	Double	Total price of items in the cart

4.3.1.5 Table 5: Order

Entity	Attributes	Data Type	Description
Order	OrderID	String	Unique identifier for each order
	UserID	String	ID of the user placing the order
	Products	List	List of product IDs included in the order
	OrderDate	DateTime	Date and time when the order was placed
	OrderStatus	String	Status of the order (e.g., pending, delivered)
	TotalAmount	Double	Total amount for the order

4.3.1.6 Table 6: Payment

Entity	Attributes	Data Type	Description
Payment	PaymentID	String	Unique identifier for each payment
	OrderID	String	ID of the associated order
	UserID	String	ID of the user making the payment
	Amount	Double	Amount paid
	PaymentDate	DateTime	Date and time when the payment was made
	PaymentMethod	String	Method used for payment (e.g., credit card)

4.3.1.7 Table 7: Admin

Entity	Attributes	Data Type	Description
Admin	AdminID	String	Unique identifier for each admin
	Name	String	Admin's full name
	Email	String	Admin's email address
	Password	String	Admin's login password
	Phone	Int	Admin's mobile number

4.3.1.8 Table 8: Feedback

Entity	Attributes	Data Type	Description
Feedback	FeedbackID	String	Unique identifier for each feedback
	UserID	String	ID of the user giving feedback
	Content	String	Actual text content of the feedback
	Attachments	List	List of images sent as attachments with feedback

4.3.1.9 Table 9: Notifications

Entity	Attributes	Data Type	Description
Notification	NotificationID	String	Unique identifier for each notification
	UserID	String	ID of the user receiving the notification
	Message	String	Content of the notification
	DateTime	DateTime	Date and time when the notification was sent

4.3.1.10 Table 10: AI Chatbot

Entity	Attributes	Data Type	Description
Chatbot	ChatbotID	String	Unique identifier for each chatbot interaction
	UserID	String	ID of the user interacting with the chatbot
	UserQuery	String	Query or command input by the user
	BotResponse	String	Response generated by the chatbot
	DateTime	DateTime	Date and time of the interaction

5 Implementation

5.1 Algorithm

Algorithm 1: Login for Users
Input: Enter Email, Password
Output: Redirection to Dashboard after successful login

begin

valid_input_email:= false

valid_input_password := false

logged_in := false

function(string email, string password)

while (logged_in = false) do

begin

if username is Users:

 valid_input_email := true

if password is Equals User.Password then

 valid_input_password := true logged_in

 := true

else

 valid_input_password := false

 logged_in := false

end if

else

 valid_input_email:= false

 logged_in := false

end if

while

if logged_in = true then

 return (success)

else

 return(failure)

else if

end

Algorithm 2 Registration for Users

Input: Enter Email, Password, Contact No, Gender, Location

Output: Redirection to Login after successful Registration

begin

valid_input_email:= false

valid_input_password:= false

valid_contact_no:= false

logged_in:= false

function(string email, string password, string contact_no)

while (logged_in = false) do

begin

if(email is null or password is null or contact_no is null)

if validRegex.matches(email):

 valid_input_email:= true

if validRegex.matches(password)

 valid_input_password:= true

if validRegex.matches(contact_no)

 valid_input_contact_no:= true

if(valid_input_email is true and valid_input_password is true and valid_contact_no is true)

 logged_in:= true

end if

end while

if logged_in = true then

 return (success)

else

 return(failure)

end if

end

Algorithm 3: RESNET Transfer Learning for Image Classification

Input: image where it must be in jpg, png or jpeg

Output: predicted category of image

Step 1 Split Dataset R into non-test set Z, and test set Z;

Step 2 **for** S=[CLFS-I, CLFS-I, CLFS-III, CLFS-IV]

 CLFS = S;

for L=[121, 169, 201]

 M=ResNet-(L);

for r= 1:10

 Split Dataset non-test set Z into training set Xr, validation set Yr;

Model(M, S, r) = TrainNetwork[M, S, Xp Yr];

Result(M, S, r) = Predict{Model(M, S,r), Yr};

Pval (M, S,7) =Compare(Result(M, S,r),£));

 end

Pval (M, S) = mean,[Pyqi(M,5,r)],

end

end

Step 3 [M*, S*] = argmax[?,i(M,5)],

Step 4 **for** r=1:10

 Split dataset non-test set Z into a training set Xr, validation set Yr;

Model(r) = TrainNetwork[M*, S*, X, Yr,];

Result(r) = Predict[Model(r), Z];

Ptest (r) = Compare[Result(r), £];

end

Step 5 **Output Prog** = mean,[Prese(T)]-

Algorithm 4: Chat Bot**Input:** Customer Query**Output:** Chatbot Response**begin**

valid_input_query := false

generated_response := ""

function(string user_query)

while (valid_input_query = false) **do** **begin** **if** user_query is not empty **then**

valid_input_query := true

// Process the query to generate a response

generated_response := generate_response(user_query)

// Check if a response was generated

if generated_response is not empty **then**

return (generated_response)

else

return ("Sorry, I couldn't understand your query. Can you please rephrase?")

end if **else**

return ("Please enter a query.")

end if **end while****end**

Algorithm 5: Order Payment**Input:** Payment Method, Order Price**Output:** order confirmation**begin**

```
    valid_order := false  
    payment_successful := false  
    order_status := "Pending"
```

```
    function(string user_id, string order_id, payment_details)
```

```
        // Step 1: Validate Order
```

```
        valid_order := validate_order(user_id, order_id)
```

```
        if valid_order = true then
```

```
            // Step 2: Process Payment
```

```
            payment_successful := process_payment(payment_details)
```

```
            if payment_successful = true then
```

```
                // Step 3: Update Order Status
```

```
                order_status := "Confirmed"
```

```
                update_order_status(order_id, order_status)
```

```
                return ("Payment successful. Order is confirmed.")
```

```
            else
```

```
                return ("Payment failed. Please check your payment details and try again.")
```

```
            end if
```

```
        else
```

```
            return ("Invalid order. Please check the order ID and try again.")
```

```
        end if
```

```
end
```

Algorithm 6: Sending notifications to users**Input:** user_id, product_id, order_id**Output:** success or failure message**begin**

notification_sent := false

notification_type := ""

function(string user_id, string notification_type, string product_id, string order_id)

// Step 1: Identify User Preferences

user_preferences := get_user_preferences(user_id)

if user_preferences[notification_type] = true **then**

// Step 2: Gather Notification Data

if notification_type = "New Product" **then**

product_details := get_product_details(product_id)

message := "New product alert! Check out " + product_details.name + " now available!"

else if notification_type = "Order Status Update" **then**

order_status := get_order_status(order_id)

message := "Your order #" + order_id + " status has been updated to " + order_status + "."

end if

// Step 3: Send Notification

notification_sent := send_notification(user_id, message)

if notification_sent = true **then**

// Step 4: Log Notification

log_notification(user_id, notification_type, message)

return ("Notification sent successfully.")

else

return ("Failed to send notification. Please try again later.")

```

end if
else
    return ("User has opted out of " + notification_type + " notifications.")
end if
end

```

Algorithm 7: Admin User Management

Input: admin_id, operation_type, user_id, user_details

Output: success or failure message

```

begin
    operation_successful := false
    operation_type := ""

    function(string admin_id, string operation_type, string user_id, string user_details)

        // Step 1: Validate Admin Credentials
        if validate_admin(admin_id) = true then

            if operation_type = "Add User" then
                // Step 2: Add User
                operation_successful := add_user(user_details)

                if operation_successful = true then
                    return ("User added successfully.")
                else
                    return ("Failed to add user. Please check the details and try again.")
                end if

            else if operation_type = "Modify User" then
                // Step 3: Validate User ID
                if validate_user(user_id) = true then
                    operation_successful := modify_user(user_id, user_details)
                end if
            end if
        end if
    end

```

```

if operation_successful = true then
    return ("User modified successfully.")

else
    return ("Failed to modify user. Please check the details and try again.")

end if

else
    return ("Invalid user ID. Please check and try again.")

end if

else if operation_type = "Remove User" then
    // Step 4: Validate User ID
    if validate_user(user_id) = true then
        operation_successful := remove_user(user_id)

        if operation_successful = true then
            return ("User removed successfully.")

        else
            return ("Failed to remove user. Please try again.")

        end if

    else
        return ("Invalid user ID. Please check and try again.")

    end if

else
    return ("Invalid operation type. Please specify 'Add User', 'Modify User', or 'Remove User'.")
end if

else
    return ("Admin validation failed. Access denied.")
end if

end

```

Algorithm 8: Admin Order Management**Input:** admin_id, operation_type, order_id, updated_order_details**Output:** success or failure message**begin**

operation_successful := false

operation_type := ""

function(string admin_id, string operation_type, string order_id, string updated_order_details)

// Step 1: Validate Admin Credentials

if validate_admin(admin_id) = true **then** **if** operation_type = "View Orders" **then**

// Step 2: Retrieve Orders

orders := get_all_orders()

return (orders) // Return the list of orders **else if** operation_type = "Update Order" **then**

// Step 3: Validate Order ID

if validate_order(order_id) = true **then**

operation_successful := update_order(order_id, updated_order_details)

if operation_successful = true **then** **return** ("Order updated successfully.") **else** **return** ("Failed to update order. Please check the details and try again.") **end if** **else** **return** ("Invalid order ID. Please check and try again.") **end if**

```
else if operation_type = "Remove Order" then
    // Step 4: Validate Order ID
    if validate_order(order_id) = true then
        operation_successful := remove_order(order_id)

        if operation_successful = true then
            return ("Order removed successfully.")
        else
            return ("Failed to remove order. Please try again.")
        end if
    else
        return ("Invalid order ID. Please check and try again.")
    end if

    else
        return ("Invalid operation type. Please specify 'View Orders', 'Update Order', or 'Remove Order'.")
    end if
else
    return ("Admin validation failed. Access denied.")
end if
end
```

Algorithm 9: Product Performance Tracking**Input:** product_id**Output:** Performance Report, Error Message**begin**

function(string product_id)

// Step 1: Validate Product ID

if validate_product(product_id) = true **then**

// Step 2: Retrieve Product Data

sales_data := get_sales_data(product_id) // Get

total sales for the product

revenue_data := get_revenue_data(product_id) //

Get total revenue generated by the product

ratings_data := get_ratings_data(product_id) // Get

customer ratings for the product

// Step 3: Calculate Average Rating

average_rating :=

calculate_average_rating(ratings_data)

// Step 4: Prepare Performance Report

performance_report := {

"product_id": product_id,

"total_sales": sales_data.total_sales,

"total_revenue": revenue_data.total_revenue,

"average_rating": average_rating

}

```

    return (performance_report)

else
    return ("Invalid product ID. Please check and try
again.")

end if

end

```

Algorithm 10: Inventory Monitoring and Alert System

Input: product_id

Output: Alert Messages

```

begin
    function(string product_id)

        // Step 1: Validate Product ID
        if validate_product(product_id) = true then

            // Step 2: Retrieve Current Inventory Level
            current_inventory := get_current_inventory(product_id) // Get current stock level for the
product

            // Step 3: Define Thresholds
            low_stock_threshold := get_low_stock_threshold(product_id) // Minimum stock level before
alert
            overstock_threshold := get_overstock_threshold(product_id) // Maximum stock level before
alert

            // Step 4: Check Inventory Levels
            if current_inventory < low_stock_threshold then
                return ("Alert: Low stock for product " + product_id + ". Current inventory: " +
current_inventory)

```

```

else if current_inventory > overstock_threshold then
    return ("Alert: Overstock for product " + product_id + ". Current inventory: " +
current_inventory)

else
    return ("Inventory level for product " + product_id + " is healthy. Current inventory: " +
current_inventory)
end if

else
    return ("Invalid product ID. Please check and try again.")
end if
end

```

Algorithm 11: Cat Food Item Search

Input: search_query

Output: Matching Items, error message

begin

function(string search_query)

// Step 1: Validate Search Query

if search_query is empty **then**

return ("Please enter a search term.")

// Step 2: Retrieve Cat Food Items

all_cat_food_items := get_all_cat_food_items() // Get the list of all available cat food items

// Step 3: Filter Cat Food Items

matching_items := filter_cat_food_items(all_cat_food_items, search_query)

// Step 4: Check for Matches

if matching_items is not empty **then**

return (matching_items) // Return the list of matching cat food items

else

```

    return ("No cat food items found matching your search.")

end if

end

```

Algorithm 12: Remove Items from Cart

Input: user_id, item_id

Output: Message

```

begin

    function(string user_id, string item_id)

        // Step 1: Validate User ID and Item ID
        if validate_user(user_id) = false then
            return ("Invalid user ID. Please check and try again.")

        if validate_item(item_id) = false then
            return ("Invalid item ID. Please check and try again.")

        // Step 2: Retrieve User's Cart
        user_cart := get_user_cart(user_id) // Get the user's current cart

        // Step 3: Check if Item Exists in Cart
        if item_id in user_cart then

            // Step 4: Remove Item from Cart
            remove_item(user_cart, item_id)
            update_cart(user_id, user_cart) // Update the cart in the database

            return ("Item " + item_id + " has been removed from your cart.")

```

```

else
    return ("Item " + item_id + " is not in your cart.")
end if
end

```

Algorithm 13: Add Items to Cart

Input: user_id, item_id, quantity

Output: Message

```

begin
    function(string user_id, string item_id, integer quantity)

        // Step 1: Validate User ID and Item ID
        if validate_user(user_id) = false then
            return ("Invalid user ID. Please check and try again.")
        if validate_item(item_id) = false then
            return ("Invalid item ID. Please check and try again.")

        // Step 2: Validate Quantity
        if quantity <= 0 then
            return ("Invalid quantity. Please enter a quantity greater than zero.")

        // Step 3: Retrieve User's Cart
        user_cart := get_user_cart(user_id) // Get the user's current cart

        // Step 4: Check if Item Already Exists in Cart
        if item_id in user_cart then
            // Update quantity if item already exists
            user_cart[item_id].quantity := user_cart[item_id].quantity + quantity
        else
            // Add new item to cart
            user_cart[item_id] := {
                "quantity": quantity,
                "product_name": get_product_name(item_id), // Get product details
            }

```

```

    "price": get_product_price(item_id) // Get product price
}

// Step 5: Update Cart
update_cart(user_id, user_cart) // Update the cart in the database

return (quantity + " of item " + item_id + " has been added to your cart.")

end

```

Algorithm 14: Refine Search Results

Input: search_query, breed_type, food_category,

Output: Matching Items, error message

```

begin

    function(string search_query, string breed_type, string food_category, float min_price, float
max_price)

        // Step 1: Validate Search Query
        if search_query is empty then
            return ("Please enter a search term.")

        // Step 2: Retrieve All Cat Food Items
        all_cat_food_items := get_all_cat_food_items() // Get the list of all available cat food items

        // Step 3: Filter Cat Food Items by Search Query
        matching_items := filter_cat_food_items(all_cat_food_items, search_query)

        // Step 4: Apply Additional Filters
        if breed_type is not empty then
            matching_items := filter_by_breed(matching_items, breed_type)

        if food_category is not empty then
            matching_items := filter_by_food_category(matching_items, food_category)

        if min_price >= 0 or max_price >= 0 then

```

```

matching_items := filter_by_price_range(matching_items, min_price, max_price)

// Step 5: Check for Matches

if matching_items is not empty then

    return (matching_items) // Return the filtered list of matching items

else

    return ("No items found matching your criteria.")

end if

end

```

5.2 External APIs/SDKs

API used in FelineConnect are as follows:

Table 1: Details of APIs used in the FelineConnect

Name of API and version	Description of API	Purpose of usage	List down the API endpoint/function/class in which it is used
TensorFlow/PyTorch	Libraries for building and training machine learning models, especially CNNs for image-related tasks.	Training and implementing convolutional neural networks for image processing.	tensorflow.keras.models tensorflow.keras.utils tensorflow.nn
Stripe API	Comprehensive API for online payment processing for internet businesses.	Handling payments, transactions for store.	https://api.stripe.com
Gemini API	Chatbot	To integrate an automated chatbot to help users with latest alerts and information	https://ai.google.dev/gemini-api/docs/api-key

5.3 User Interface

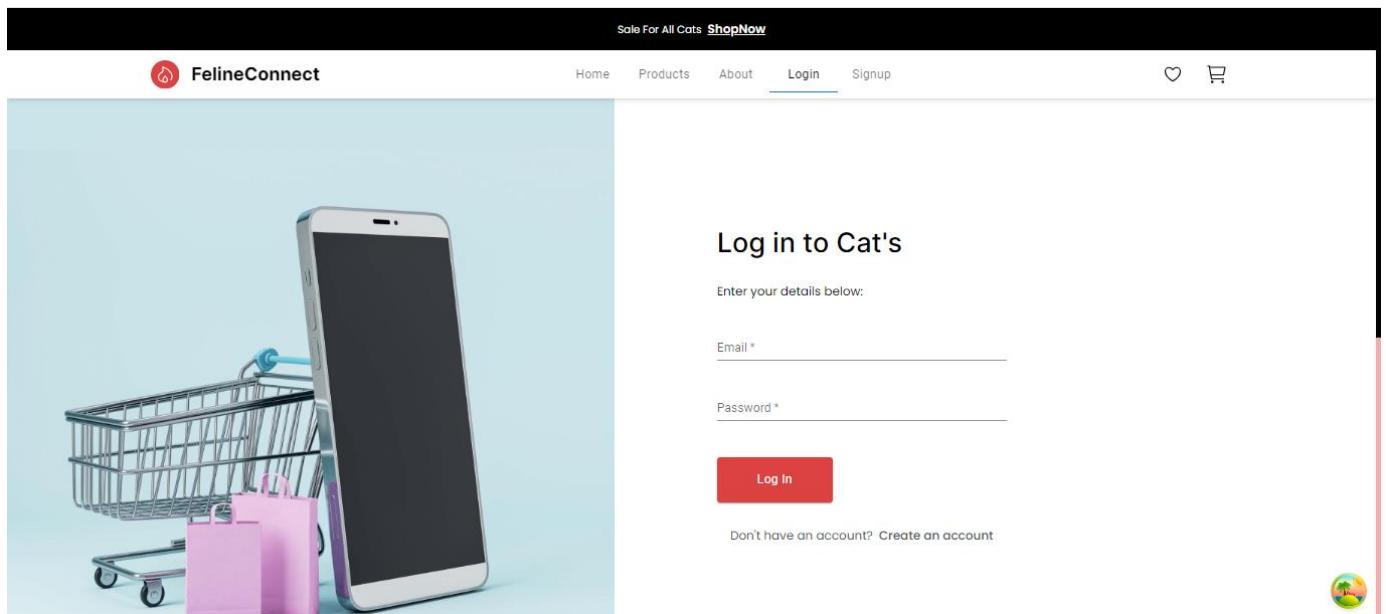


Figure 25: Login page of Feline Connect

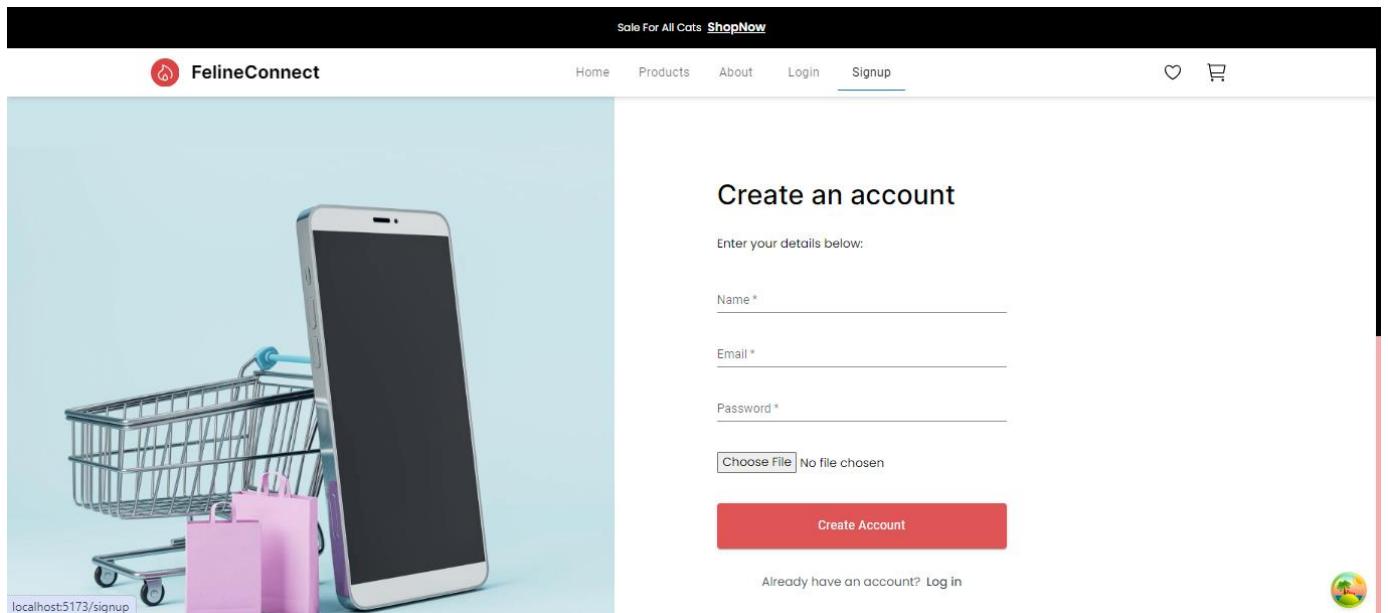


Figure 26: Sign up page of FelineConnect

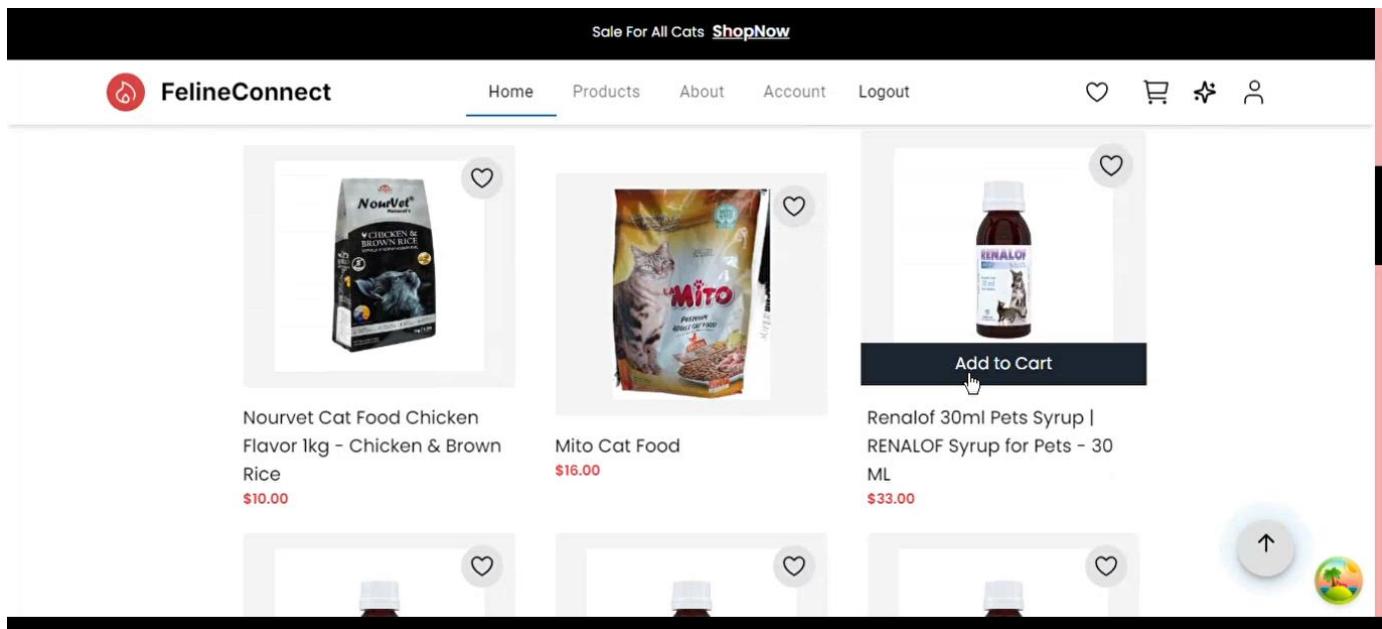


Figure 27: Dashboard Page of FelineConnect

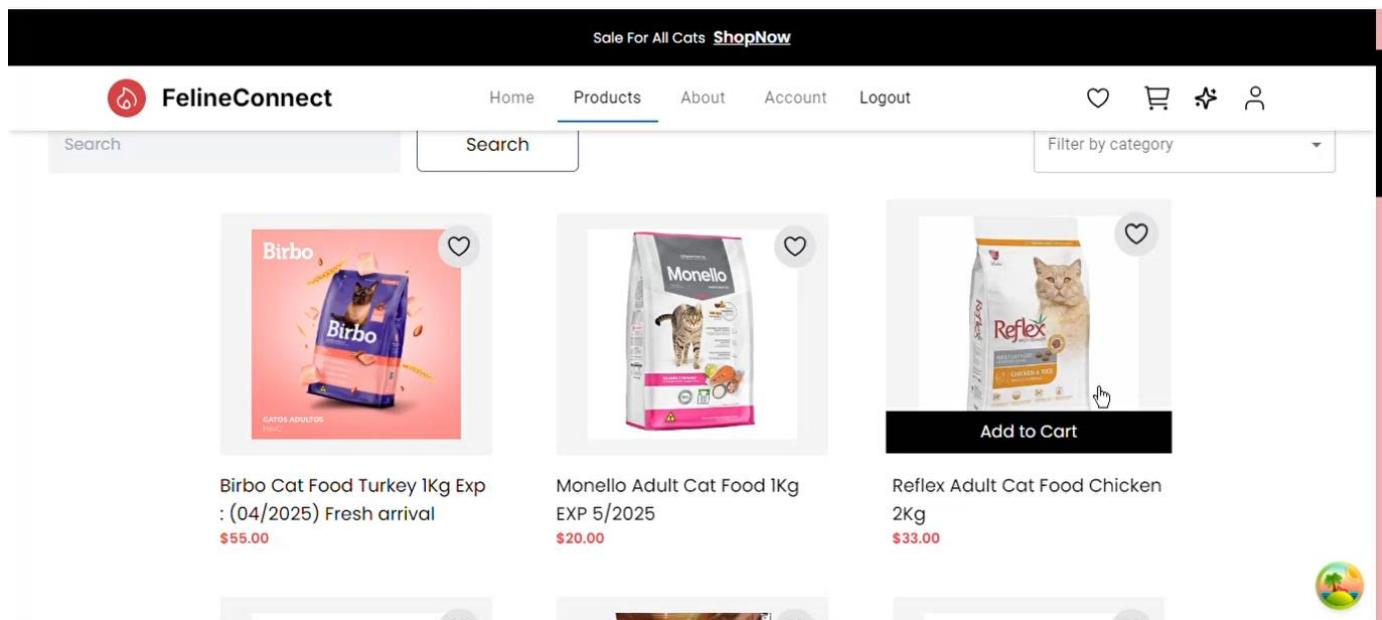


Figure 28: Products page

Sale For All Cats [ShopNow](#)

 FelineConnect

Home Products About Account Logout

Heart Cart Star User

Home / Cart

Product	Price	Quantity	Sub total
 Reflex Adult Cat Food Chicken 2Kg	\$33.00	<input type="button" value="5"/> + -	\$165

[Return to Shop](#)



Figure 29: Cart page

Sale For All Cats [ShopNow](#)

 FelineConnect

Home Products About Account Logout

Heart Cart Star User

Wishlist (1)

[Move All to Bag](#)



[Remove from Cart](#)

Reflex Adult Cat Food
Chicken 2Kg



Figure 30: Wishlist page

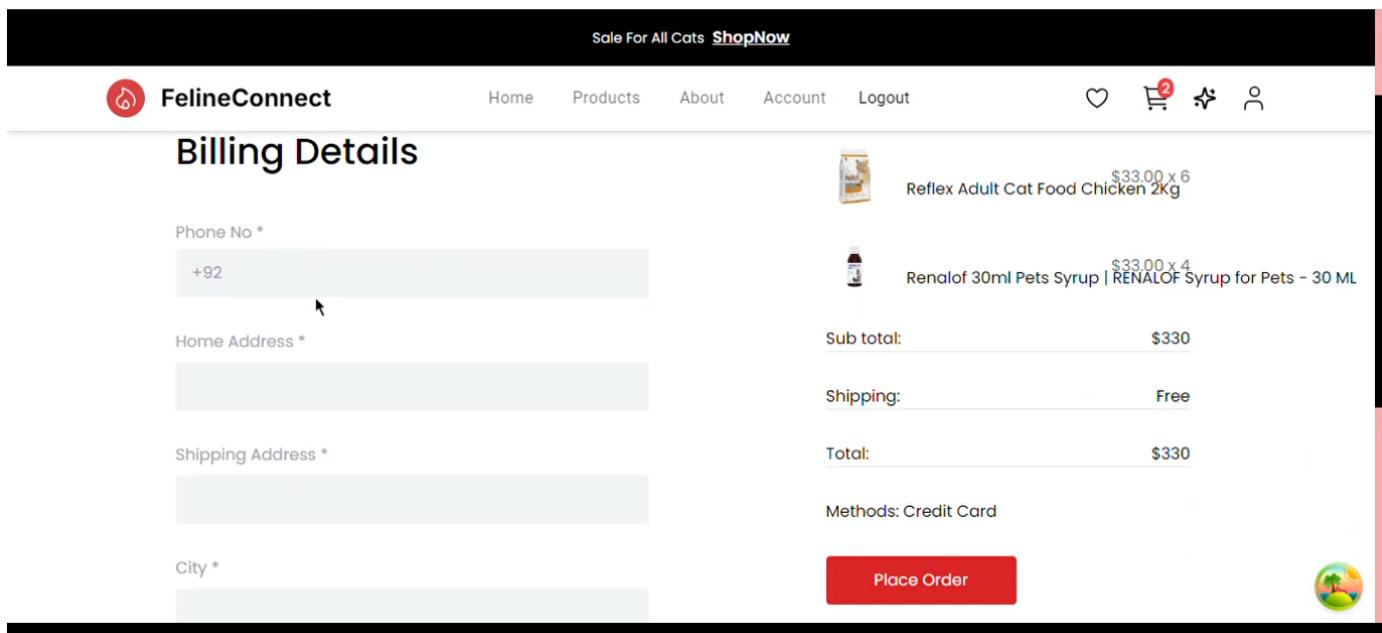


Figure 31: Checkout page

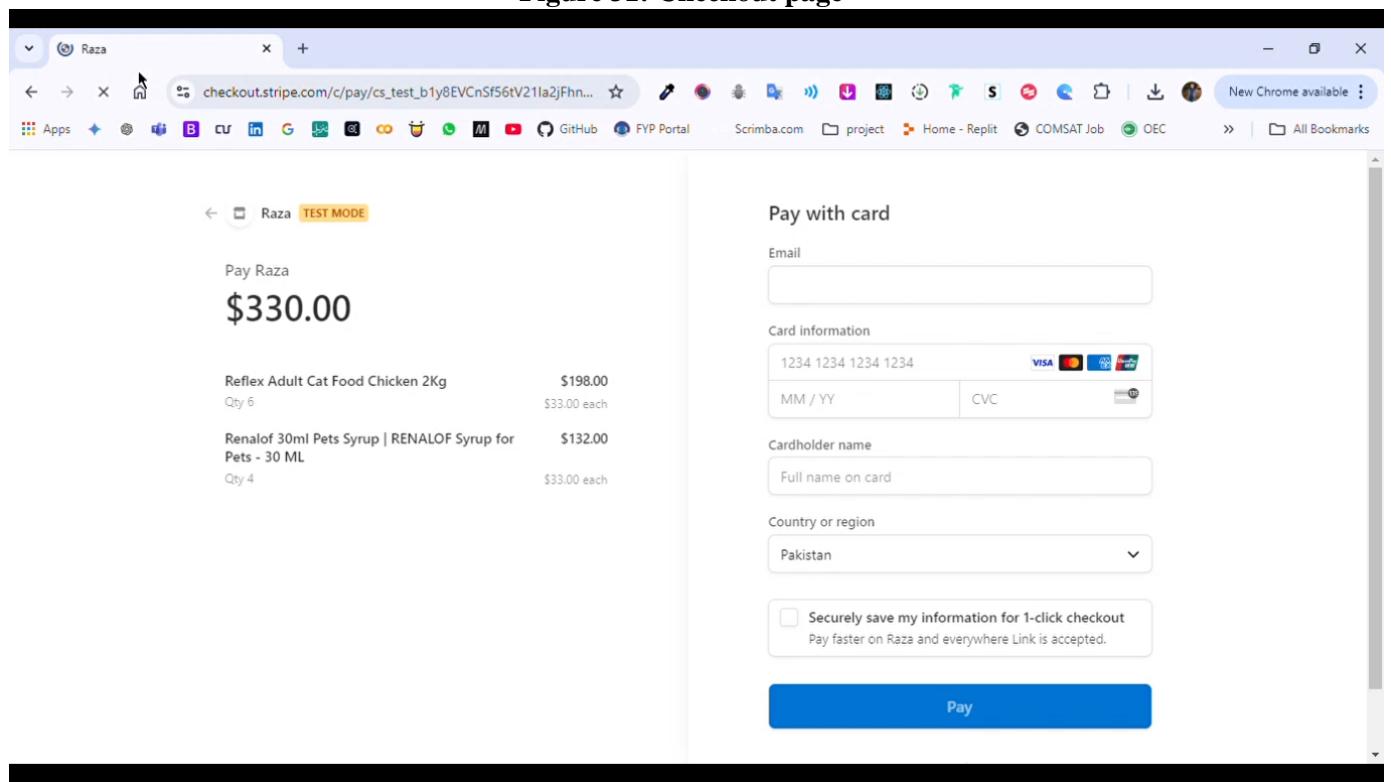


Figure 32: Payment page

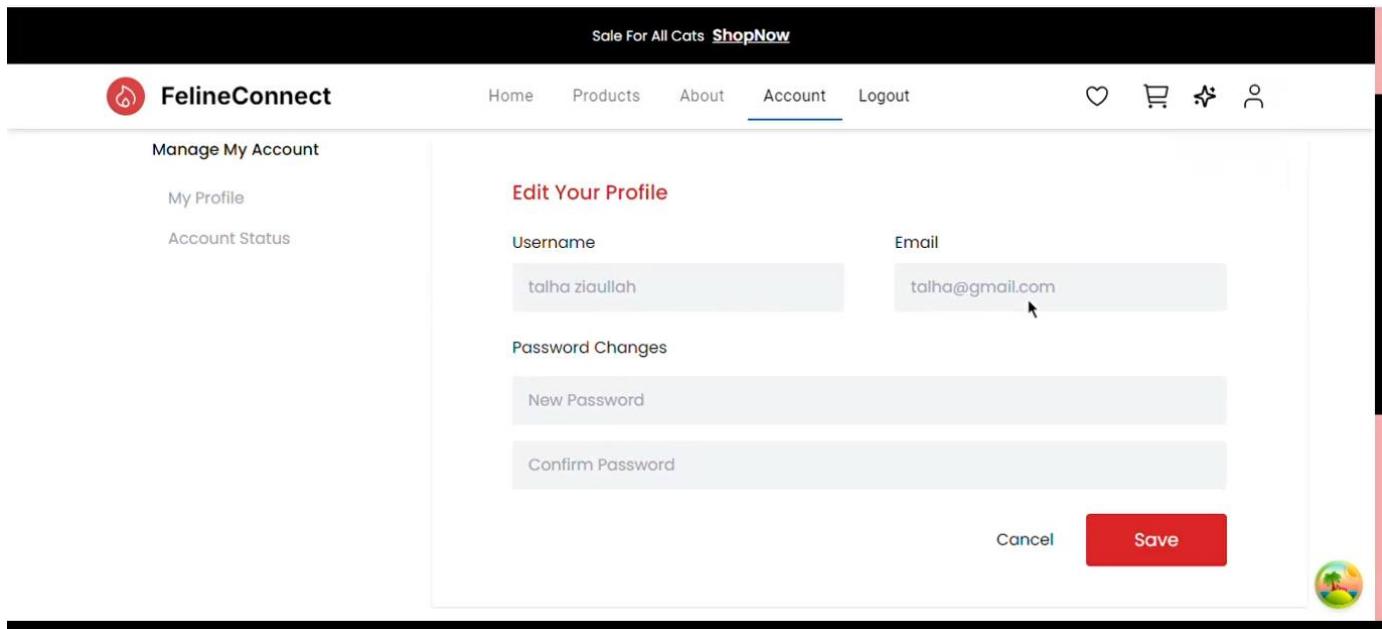


Figure 33: Edit profile page

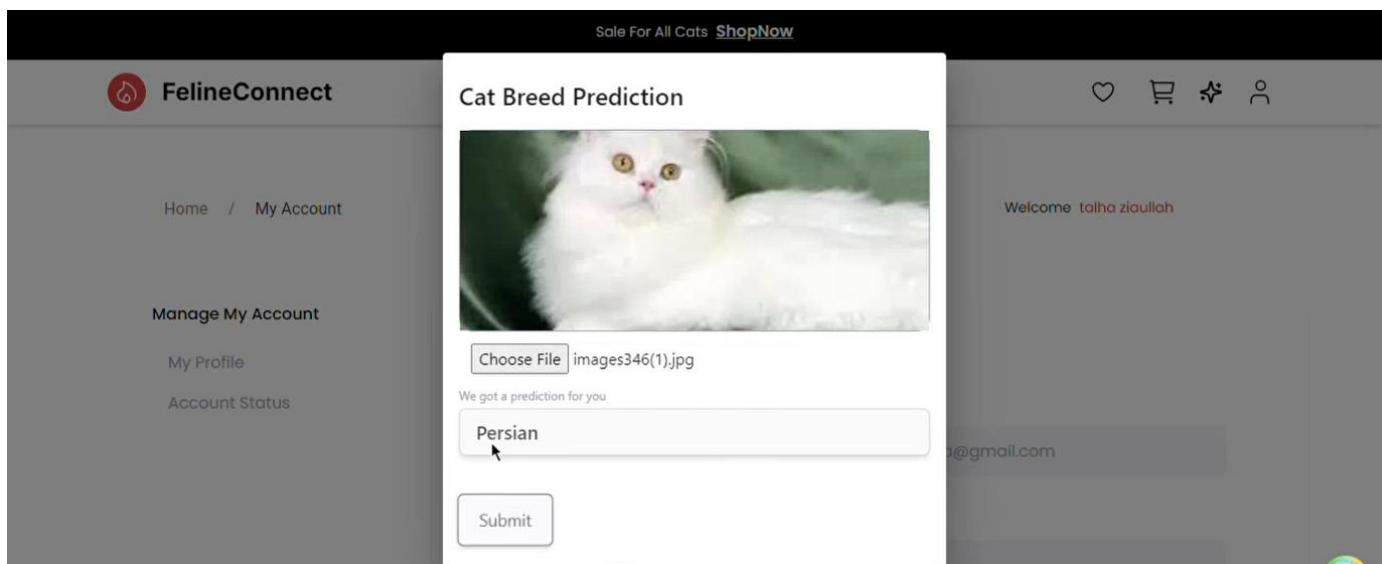


Figure 34: Breed detection page

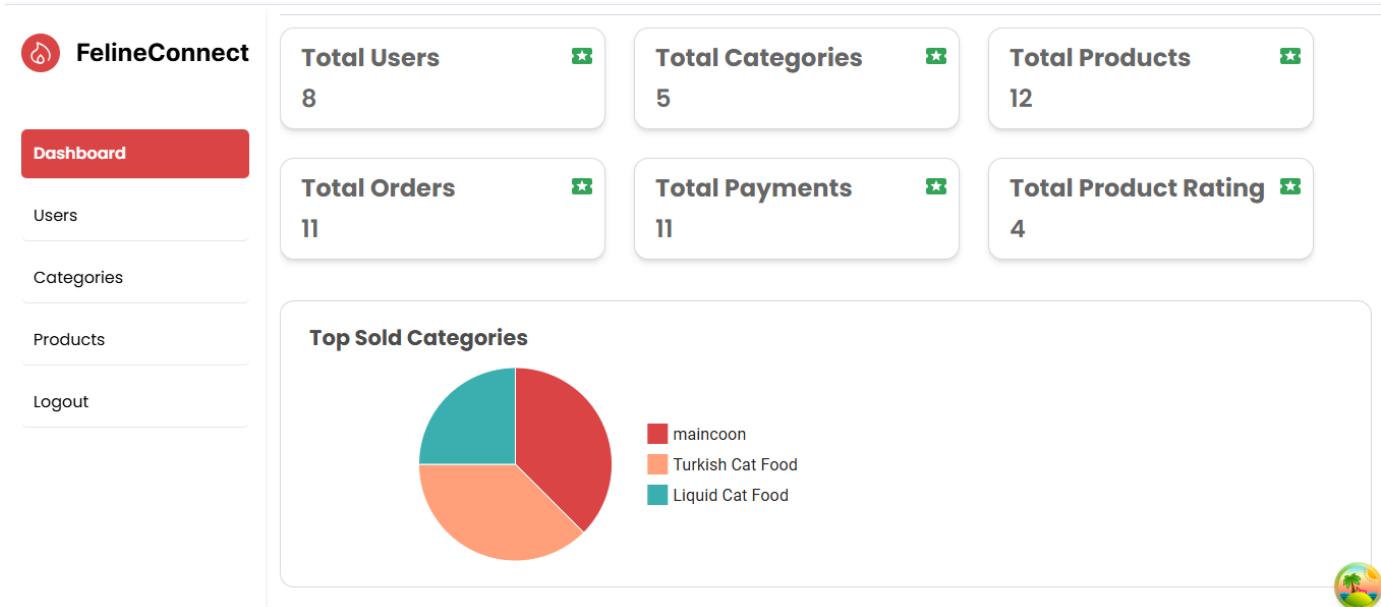


Figure 35: Admin Dashboard

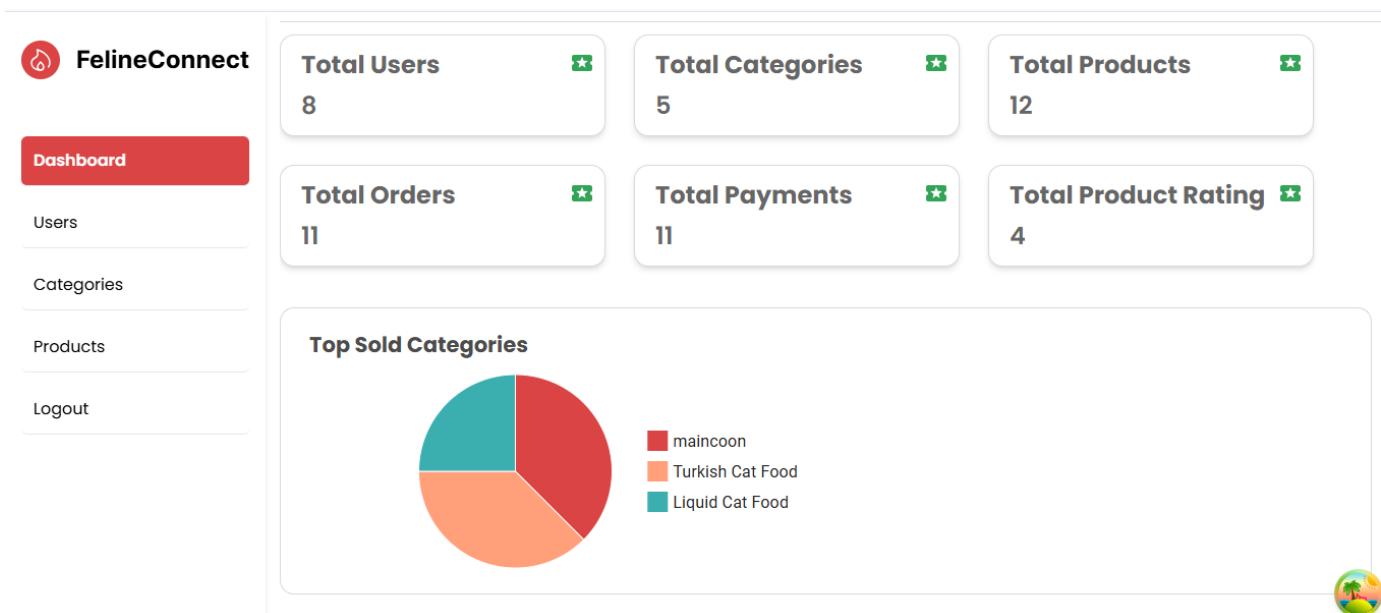


Figure 36: Admin Dashboard (continue)

FelineConnect

Dashboard

Users

Categories

Products

Logout

User Management

Create User

ID	Username	Email	Total Orders	Total Reviews	Edit	Delete
22	mohsin	mohsin@gmail.com	1	2	Edit	Delete
21	123hassan	hassan@gmail.com	0	0	Edit	Delete
20	talha123	talha@gmail.com	1	1	Edit	Delete
19	Saifullah	khanben123@gmail.com	5	1	Edit	Delete
18	Raza123	test@gmail.com	0	0	Edit	Delete
17	test123	admin@gmail.com	0	0	Edit	Delete
12	Raza123	raza@gmail.com	4	0	Edit	Delete

Figure 37: User Management Admin side

FelineConnect

Dashboard

Users

Categories

Products

Logout

Category Management

Create Category

ID	Name	Edit	Delete
1	Turkish Cat Food	Edit	Delete
2	Dry Cat Food	Edit	Delete
3	Liquid Cat Food	Edit	Delete
4	Persian Cats	Edit	Delete
9	maincoon	Edit	Delete

Figure 38: Category Management Admin side



FelineConnect

[Dashboard](#)

[Users](#)

[Categories](#)

Products

[Logout](#)

Product Management

[Create Product](#)

ID	Name	Description	Price	Quantity	Rating	Edit	Delete
1	Birbo Cat Food Turkey 1Kg Exp : (04/2025)	This is one of the most popular item	90.00	99	3.5	Edit	Delete
2	Monello Adult Cat Food 1Kg EXP 5/2025	This is one of the most popular item	20.00	120	5	Edit	Delete
3	Reflex Adult Cat Food Chicken 2Kg	This is one of the most popular item	33.00	120	3	Edit	Delete
4	Nourvet Cat Food Chicken Flavor 1kg - Chicken & Brown Rice	This is one of the most popular item	10.00	110	0	Edit	Delete
5	Mito Cat Food	This is one of the most popular item	16.00	70	0	Edit	Delete

Figure 39: Product Management Admin side



FelineConnect

[Dashboard](#)

[Users](#)

[Categories](#)

Products

[Logout](#)

Add Product

[Create Product](#)

ID	Name	Price	Quantity	Rating	Edit	Delete
1	Birbo Cat Food Turkey 1Kg Exp : (04/2025)	90.00	99	3.5	Edit	Delete
2	Monello Adult Cat Food 1Kg EXP 5/2025	20.00	120	5	Edit	Delete
3	Reflex Adult Cat Food Chicken 2Kg	33.00	120	3	Edit	Delete
4	Nourvet Cat Food Chicken Flavor 1kg - Chicken & Brown Rice	10.00	110	0	Edit	Delete
5	Mito Cat Food	16.00	70	0	Edit	Delete

Add Product

Title *

Details *

Price *

Quantity *

Product Image*

Choose File No file chosen

Turkish Cat Food

Figure 40: Add product by Admin

FelineConnect

Dashboard

Users

Categories

Products

Logout

User Management

ID	Username	Email	Total Reviews	Edit	Delete
22	mohit	mohit@gmail.com	2	Edit	Delete
21	123haha	123haha@gmail.com	0	Edit	Delete
20	talha	talha@gmail.com	0	Edit	Delete
19	Saifullah	saifullah@gmail.com	0	Edit	Delete
18	Raza	raza@gmail.com	4	Edit	Delete
17	test123	admin@gmail.com	0	Edit	Delete
12	Razal23	raza@gmail.com	0	Edit	Delete

Create User

Add User

username *

Email *

Password

Submit

Figure 41: Add user by Admin

Sale For All Cats [ShopNow](#)

FelineConnect

Home Products About Account Logout

Cat Foods

Flash Sales

Next →

ChatBot

Hello there! I'm Chatbot, a language model trained to assist you with various topics and questions. What can I help you with today? Feel free to type your query or topic of interest, and I'll do my best to provide you with helpful and informative responses.

Send a chat

Figure 42: Gemini Chatbot

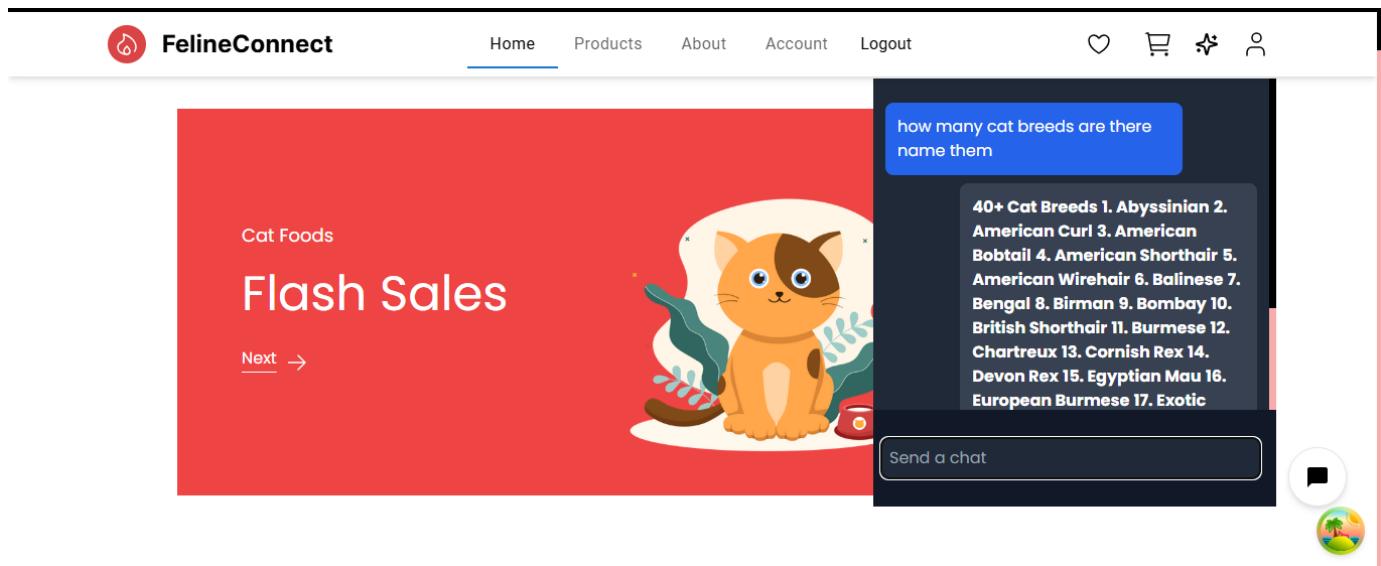


Figure 43: Gemini Chatbot

5.4 Deployment

The project's deployment involves a comprehensive approach encompassing multiple modules for seamless functionality. For user authentication and profile management, we utilize a **Node.js server with Express.js** running locally, paired with **MySQL** for database management. This setup ensures robust account creation, access, and management functionalities.

Image submission leverages **local storage** for storing files and **Python with OpenCV** for image preprocessing, ensuring standardized input for accurate breed recognition.

ResNet-50 integration runs locally using **Python and TensorFlow** for machine learning tasks, facilitating precise cat breed identification through transfer learning.

The breed detection result is processed and displayed using a **Node.js backend** and a **React.js frontend**, both running locally. The system presents users with detected breed information and visual insights alongside a feedback mechanism for accuracy enhancement.

The online store for cat food operates using a **Node.js backend** with **MySQL** as the database, providing a seamless shopping experience and a review system.

Payment processing integrates a secure payment gateway via **Node.js and Express.js**, running locally, ensuring secure transactions and order confirmation.

User interaction and queries are managed using a **Node.js backend with Express.js** and **Gemini for chatbot implementation**, enabling quick responses to user inquiries.

Lastly, admin management employs a **Node.js backend** and a **React.js frontend**, both hosted locally, to enable efficient monitoring and management of user feedback, inventory, accounts, and system performance, ensuring the overall security and effectiveness of the system.

6 Testing and Evaluation

6.1 Unit Testing

6.1.1 Module 1: User Authentication and Profile Management

No.	Test Case / Test Script	Attribute and Value	Expected Result	Result
1	Check user registration with valid information	Email: khanben123@gmail.com Password: saifullah Name: Saifullah	Account is created, and a confirmation message is displayed	Pass
2	Check user registration with missing email	Email: "" Password: saifullah Name: Saifullah	Registration fails, error message "Email is required" is displayed	Pass
3	Check user registration with missing password	Email: khanben123@gmail.com Password: "" Name: Saifullah	Registration fails, error message "Password is required" is displayed	Pass
4	Check user login with valid credentials	Email: khanben123@gmail.com Password: saifullah	User successfully logs in and is redirected to the user profile page	Pass
5	Check user login with invalid credentials	Email: khanben123@gmail.com Password: wrongpassword	Login fails, error message "Invalid credentials" is displayed	Pass
6	Check user login with empty email field	Email: "" Password: saifullah	Login fails, error message "Email is required" is displayed	Pass
7	Check user login with empty password field	Email: khanben123@gmail.com Password: ""	Login fails, error message "Password is required" is displayed	Pass
8	Check user profile update with valid information	Name: Saifullah Profile Photo: <path-to-photo> Bio: "Loves cats"	Profile is updated, and confirmation message "Profile updated successfully" is displayed	Pass
9	Check user profile update with missing name	Name: "" Profile Photo: <path-to-photo> Bio: "Loves cats"	Profile update fails, error message "Name is required" is displayed	Pass
10	Check user profile update with missing photo	Name: Saifullah Profile Photo: "" Bio: "Loves cats"	Profile update succeeds without photo, and confirmation message "Profile updated successfully" is displayed	Pass

11	Check user deleting account	Action: Click "Delete Account"	Account is permanently deleted, and the confirmation message "Account deleted successfully" is displayed	Pass
12	Check user changing existing password	Old Password: saifullah New Password: newpassword123	Password is changed, and the confirmation message "Password changed successfully" is displayed	Pass
13	Check user changing password with incorrect old password	Old Password: wrongpassword New Password: newpassword123	Password change fails, error message "Incorrect old password" is displayed	Pass
14	Check user using the forgot password option	Email: khanben123@gmail.com	Password reset email is sent, and confirmation message "Password reset link sent" is displayed	Pass
15	Check forgot password with unregistered email	Email: unregistered@example.com	Forgot password fails, error message "Email not found" is displayed	Pass
16	Check user login after changing the password	Email: khanben123@gmail.com Password: newpassword123	User logs in successfully with new password and is redirected to the user profile page	Pass
17	Check user account creation with duplicate email	Email: khanben123@gmail.com Password: newpassword123 Name: John	Registration fails, error message "Email already exists" is displayed	Pass
18	Check user viewing profile information	Action: Click "View Profile"	User is shown their profile with correct name, photo, and bio	Pass
19	Check user logout functionality	Action: Click "Logout"	User is logged out and redirected to the login page	Pass

6.1.2 Module 2: Cat Breed Detection

No.	Test Case / Test Script	Attribute and Value	Expected Result	Result
1	Check uploading an image of a cat for breed identification	Image: <path-to-cat-image>	Image is successfully uploaded, and breed prediction options are displayed	Pass
2	Check uploading an invalid image (not a cat) for breed identification	Image: <path-to-invalid-image>	Error message "Invalid image. Please upload a valid cat image" is displayed	Pass
3	Check splitting the dataset into training and testing sets	Dataset: <path-to-dataset>	Dataset is split successfully into training and testing sets with 80-20 split ratio	Pass
4	Check integrating ResNet-50 architecture for breed recognition	Action: Use ResNet-50 architecture	Model is initialized with ResNet-50 architecture, and setup for breed recognition is complete	Pass

5	Check training ResNet-50 with a dataset of cat images	Dataset: <path-to-cat-image-dataset>	Model training is completed successfully with an accuracy report	Pass
6	Check transfer learning on a pre-trained ResNet-50 model	Action: Use pre-trained ResNet-50 model	Pre-trained model is successfully loaded and adapted for the cat breed classification task	Pass
7	Check fine-tuning ResNet-50 on a specific dataset of domestic cat breeds	Dataset: <path-to-domestic-cat-breeds-dataset>	Model is fine-tuned, and training progress is displayed	Pass
8	Check model accuracy after fine-tuning	Action: Evaluate fine-tuned model	Accuracy improves after fine-tuning compared to the pre-trained model	Pass
9	Check breed prediction time for uploaded image	Image: <path-to-uploaded-image>	Breed prediction result is provided within 2 seconds	Pass

6.1.3 Module 3: Online Store

No.	Test Case / Test Script	Attribute and Value	Expected Result	Result
1	Check buying a cat breed from the store	Item: Persian Cat	Item is successfully added to the cart and can be purchased	Pass
2	Check buying a food item from the store	Item: Premium Cat Food	Item is successfully added to the cart and can be purchased	Pass
3	Check adding a cat breed to the shopping cart	Item: Bengal Cat	Bengal Cat is added to the shopping cart	Pass
4	Check adding a food item to the shopping cart	Item: Salmon Cat Food	Salmon Cat Food is added to the shopping cart	Pass
5	Check removing a cat breed from the shopping cart	Item: Bengal Cat	Bengal Cat is removed from the shopping cart	Pass
6	Check removing a food item from the shopping cart	Item: Salmon Cat Food	Salmon Cat Food is removed from the shopping cart	Pass
7	Check checking out the cart with multiple items	Cart: Bengal Cat, Salmon Cat Food	Checkout process proceeds successfully with both items in the cart	Pass
8	Check adding multiple different cat breeds to the cart	Items: Bengal Cat, Persian Cat, Maine Coon	All cat breeds are added to the cart successfully	Pass
9	Check adding multiple food items to the cart	Items: Salmon Cat Food, Chicken Cat Food	Both food items are added to the cart successfully	Pass
10	Check removing one item while multiple items are in the cart	Cart: Bengal Cat, Salmon Cat Food, Persian Cat	Removing Bengal Cat from the cart should leave Salmon Cat Food and Persian Cat	Pass
11	Check checking out an empty cart	Cart: Empty	Error message "Your cart is empty" is displayed	Pass

12	Check updating quantity of an item in the cart	Item: Bengal Cat, Quantity: 2	Cart updates quantity of Bengal Cat to 2	Pass
13	Check displaying total price after adding items to the cart	Items: Persian Cat, Salmon Cat Food	Total price is correctly calculated and displayed	Pass
14	Check successful payment with valid payment method	Payment Method: Credit Card	Payment is processed successfully, and a confirmation message is shown	Pass
15	Check adding items to the cart after checkout is complete	Cart: Persian Cat, Payment Method: Credit Card	Error message "You cannot add items after checkout" is displayed	Pass
16	Check viewing cart before proceeding to checkout	Cart: Persian Cat, Salmon Cat Food	Cart is displayed with item details (name, price, quantity) before checkout	Pass
17	Check removing an item while checkout process is ongoing	Cart: Persian Cat, Salmon Cat Food, Checkout: Ongoing	Error message "Cannot remove item while checkout is in progress" is displayed	Pass

6.1.4 Module 4: Search Filters and Notifications

No.	Test Case / Test Script	Attribute and Value	Expected Result	Result
1	Check searching for cat breeds using the search bar	Search Query: Persian Cat	Search results display the relevant cat breed (Persian Cat)	Pass
2	Check refining search results by breed type	Filter: Persian, Bengal, Maine Coon	Results display only selected breeds (Persian, Bengal, Maine Coon)	Pass
3	Check refining search results by food category	Filter: Wet Food	Results display only food items categorized as Wet Food	Pass
4	Check searching for a specific cat food item	Search Query: Salmon Cat Food	Search results display the Salmon Cat Food item	Pass
5	Check removing a filter from the search results	Filter: Price Range: \$10 - \$50, Remove Price Filter	Results display all items without price filter applied	Pass
6	Check clearing all filters and search terms	Clear Filters and Search Query	Search results display all available cat breeds and food items	Pass
7	Check sending a notification for order confirmation	Notification: Order Confirmation	User receives a notification confirming their order	Pass
8	Check sending a notification for order delivery status	Notification: Order Shipped	User receives a notification that their order has been shipped	Pass
9	Check sending a notification for out-of-stock item	Notification: Item Out of Stock	User receives a notification that a previously selected item is out of stock	Pass

10	Check updating the notification for order delivery	Notification: Order Delivery Status: Out for Delivery	User receives an updated notification about their order delivery status	Pass
11	Check re-enabling notifications from the user settings	Setting: Enable Notifications	Notifications are enabled, and the user starts receiving them again	Pass
12	Check search bar functionality when no results match the search query	Search Query: "Unicorn Cat"	"No results found" message is displayed	Pass

6.1.5 Module 5: Payment Processing

No.	Test Case / Test Script	Attribute and Value	Expected Result	Result
1	Check cart checkout process	Cart: 2 Cat Items, 1 Food Item	User is redirected to the payment gateway with the correct total amount	Pass
2	Check integration with a secure payment gateway	Payment Gateway: PayPal	Payment gateway securely processes the payment and redirects the user to the confirmation page	Pass
3	Check confirming orders after successful payment	Order Confirmation: Order #12345	Order details are displayed, including product name, quantity, price, and total amount	Pass
4	Check submitting a review without a rating	Review: "Tasty food, my cat loved it!"	Error message is displayed: "Please provide a rating"	Pass
5	Check successful payment using a credit card	Payment Method: Credit Card, Amount: \$35.00	Payment is processed successfully, and user is redirected to the order confirmation page	Pass
6	Check payment failure due to insufficient funds	Payment Method: Credit Card, Amount: \$500.00	Payment is declined due to insufficient funds, and an error message is displayed	Pass
7	Check successful payment using digital wallet (e.g., Google Pay)	Payment Method: Google Pay, Amount: \$25.00	Payment is processed successfully, and user is redirected to the order confirmation page	Pass
8	Check payment cancellation by the user	Payment Method: PayPal, Amount: \$50.00	User cancels the payment before confirming, and the order is not processed	Pass
9	Check handling of invalid payment credentials	Payment Method: PayPal, Invalid Credentials	Error message is displayed: "Invalid payment credentials"	Pass
10	Check payment confirmation after successful transaction	Payment Confirmation: Order #12345, Payment Status: Success	Payment status is updated to "Paid," and the user is	Pass

			shown the payment confirmation details	
11	Check sending an order confirmation email after successful payment	Email: khanben123@gmail.com, Order: #12345	Order confirmation email is sent with payment and order details	Pass
12	Check displaying the payment summary before finalizing the checkout	Cart Summary: 3 Cat Items, 2 Food Items, Total: \$75.00	Payment summary is displayed correctly before user finalizes the checkout	Pass

6.1.6 Module 6: AI Chatbot

No.	Test Case / Test Script	Attribute and Value	Expected Result	Result
1	Check submitting a simple query	Query: "What is the best cat breed?"	The chatbot should generate a relevant response, e.g., "Persian, Maine Coon, and Siamese are popular cat breeds."	Pass
2	Check submitting a complex query	Query: "Can you recommend me a good cat food for my kitten?"	The chatbot should respond with a recommendation based on the user's input, e.g., "I recommend you try the 'Royal Canin Kitten Food.'"	Pass
3	Check chatbot response speed	Query: "Tell me about Maine Coon."	The chatbot should respond within 2 seconds with relevant information about the Maine Coon breed.	Pass
4	Check chatbot handling of a question about a breed not in database	Query: "Tell me about Abyssinian cat breed."	Chatbot should either respond with relevant information or say "Sorry, I don't have information about that breed."	Pass
5	Check chatbot handling of unrecognized input	Query: "asdffg dfgf?"	Chatbot should provide a response like "I'm sorry, I didn't understand that. Can you please rephrase?"	Pass
6	Check chatbot generating responses based on user input	Query: "What do I feed my cat?"	Chatbot should generate a response like "You can feed your cat a balanced diet including wet and dry food."	Pass
7	Check chatbot handling of a follow-up query	Query 1: "Tell me about Siamese cats." Query 2: "What about their behavior?"	The chatbot should respond appropriately, continuing the conversation with information about Siamese cat behavior.	Pass
8	Check chatbot response to greetings	Query: "Hi, how are you?"	Chatbot should respond with a friendly greeting, e.g., "Hello! I'm here to assist you. How can I help?"	Pass
9	Check chatbot generating responses for	Command: "Show me food options for kittens."	The chatbot should respond with a list of kitten food options.	Pass

	specific cat-related commands			
10	Check chatbot's knowledge about cat food brands	Query: "What is the best food brand for adult cats?"	Chatbot should respond with relevant recommendations, e.g., "Royal Canin and Hill's Science Diet are great for adult cats."	Pass
11	Check chatbot's response when asked for product recommendations	Query: "Can you recommend me a good cat litter?"	Chatbot should provide recommendations, e.g., "I recommend Tidy Cats or Fresh Step for clumping cat litter."	Pass
12	Check chatbot handling of invalid commands	Command: "Delete cat food history."	Chatbot should provide an error message like "Sorry, I can't process that command."	Pass
13	Check chatbot providing information on delivery times	Query: "When will my order arrive?"	Chatbot should provide relevant information such as "Your order will arrive within 3-5 business days."	Pass
14	Check chatbot providing information on payment status	Query: "Was my payment successful?"	Chatbot should respond with relevant information, e.g., "Yes, your payment was successful."	Pass
15	Check chatbot answering FAQs about account	Query: "How do I change my password?"	Chatbot should guide the user with instructions on how to change their password, e.g., "To change your password, go to Settings > Account > Change Password."	Pass
16	Check chatbot handling of incorrect data input	Query: "I want to buy cat food now!"	Chatbot should respond with "Please specify the type of cat food you'd like to buy."	Pass
17	Check chatbot handling of general cat care questions	Query: "How do I groom my cat?"	Chatbot should respond with grooming tips, e.g., "Brush your cat's fur regularly and trim their nails."	Pass
18	Check chatbot providing health-related information	Query: "What are common cat diseases?"	Chatbot should list common diseases, e.g., "Some common cat diseases are feline leukemia, feline immunodeficiency virus, and upper respiratory infections."	Pass

6.1.7 Module 7: Admin Panel

No.	Test Case / Test Script	Attribute and Value	Expected Result	Result
1	Check adding a new user account	Email: "newuser@gmail.com", Password: "newpassword123", Name: "New User"	Admin should be able to successfully add the new user to the system. The user should appear in the user list.	Pass
2	Check modifying an existing user account	Email: "khanben123@gmail.com", New Name: "Saifullah Khan"	Admin should be able to successfully update the user's name. The updated	Pass

			name should be reflected in the user list.	
3	Check removing a user account	Email: "olduser@gmail.com"	Admin should be able to delete the user from the system. The user should no longer appear in the user list.	Pass
4	Check viewing cat food inventory	Action: View Inventory	Admin should be able to view a list of all cat food items, including product details (name, category, price, stock).	Pass
5	Check modifying cat food inventory	Product: "Royal Canin Adult", New Price: "\$25", New Stock: "50"	Admin should be able to modify the price and stock quantity of the product. The updated details should be reflected in the inventory list.	Pass
6	Check removing a cat food item from the inventory	Product: "Whiskas Kitten Food"	Admin should be able to remove the product from the inventory. The item should no longer appear in the inventory list.	Pass
7	Check viewing order list	Action: View Orders	Admin should be able to see a list of all orders with details like order ID, customer, product, status, etc.	Pass
8	Check managing an order status	Order ID: "12345", Status: "Pending" to "Delivered"	Admin should be able to change the order status. The updated status should be reflected in the order list.	Pass
9	Check managing delivery status of an order	Order ID: "67890", Status: "Delivered"	Admin should be able to change the delivery status to "Delivered" successfully. The status change should be visible in the order details.	Pass
10	Check changing delivery status of a product	Product: "Maine Coon Kitten Food", Status: "Pending" to "Delivered"	Admin should be able to change the delivery status of the product. The status should be updated to "Delivered" in the product list.	Pass
11	Check viewing user feedback on model predictions	Action: View Feedback	Admin should be able to view the feedback submitted by users regarding the accuracy of breed prediction or other models.	Pass
12	Check deleting user feedback	Feedback ID: "Feedback003"	Admin should be able to delete user feedback. The	Pass

			feedback should no longer appear in the feedback list.	
13	Check adding new product to inventory	Product: "Hill's Science Diet Adult", Price: "\$30", Stock: "100"	Admin should be able to add a new product to the inventory. The new product should appear in the inventory list with the correct details.	Pass
14	Check modifying product details in inventory	Product: "Blue Buffalo Indoor", New Price: "\$40", New Description: "Premium Indoor Cat Food"	Admin should be able to successfully modify the details of a product. The updated product information should be displayed.	Pass
15	Check filtering orders by status	Filter: "Pending"	Admin should be able to filter and view only the orders with a "Pending" status. The filtered results should match the selected criteria.	Pass
16	Check checking out a user account for issues	User: "user1@gmail.com"	Admin should be able to identify and resolve issues with user accounts such as login problems, password reset, etc.	Pass
17	Check viewing and managing returns or cancellations	Order ID: "98765", Action: "Cancel"	Admin should be able to view the order details and process the cancellation if applicable. The order status should be updated to "Cancelled."	Pass
18	Check viewing and managing product restocking	Product: "Whiskas Adult", Stock: "Out of Stock"	Admin should be able to add more stock for the product. The stock status should be updated accordingly.	Pass
19	Check viewing sales statistics for inventory management	Action: View Sales Stats	Admin should be able to view statistics like best-selling products, sales trends, etc., to manage inventory effectively.	Pass

6.1.8 Module 8: Statistical Analytics

No.	Test Case / Test Script	Attribute and Value	Expected Result	Result
1	Generate report for most popular cat breeds	Action: Generate Report	The report should list the most popular cat breeds sold, including quantities sold and revenue generated.	Pass

2	Track sales performance of a specific product	Product: "Maine Coon", Sales Data	The sales report should show the number of units sold, total revenue, and average customer rating for "Maine Coon."	Pass
3	Track revenue of a specific product	Product: "Persian Cat", Time Range: Last Month	The revenue report for "Persian Cat" should accurately reflect sales within the last month, showing total sales amount.	Pass
4	Track customer ratings for a specific product	Product: "Siamese Cat", Date Range: Last 30 Days	The ratings report should display average customer ratings and number of reviews for "Siamese Cat" in the last 30 days.	Pass
5	View real-time sales dashboard	Action: View Dashboard	The dashboard should display real-time data with charts, graphs, and KPIs showing current sales performance, including total sales, most sold products, and revenue.	Pass
6	View real-time revenue dashboard	Action: View Revenue Dashboard	The dashboard should display live revenue data, showing total sales revenue for the day, week, and month.	Pass
7	Monitor inventory levels for a specific product	Product: "Sphynx Cat Food", Stock Level: "50"	The inventory level for "Sphynx Cat Food" should be displayed as 50, with an option to update the stock.	Pass
8	Track stockouts for a product	Product: "Bengal Cat Toy", Stock Level: "0"	The system should highlight "Bengal Cat Toy" as out of stock and send a notification alert for restocking.	Pass
9	Track overstock for a product	Product: "Catnip Spray", Stock Level: "500"	The system should display an alert that "Catnip Spray" is overstocked, providing options to adjust inventory levels.	Pass
10	Track best-selling products	Time Range: "Last 30 Days"	The report should display a list of best-selling products within the last 30 days, sorted by revenue or units sold.	Pass

6.2 Functional Testing

6.2.1 Module 1: User Authentication and Profile Management

No.	Test case / Test script	Attribute and value	Expected result	Actual result	Result
1	Register a new user account	Username: khanben123@gmail.com, Email: khanben123@gmail.com@example.com, Password: saifullah	User account is created and confirmation message is displayed.	User account created successfully.	Pass
2	Login with valid credentials	Username: khanben123@gmail.com, Password: saifullah	User is redirected to their dashboard.	Login successful, redirected to dashboard.	Pass
3	Login with incorrect password	Username: khanben123@gmail.com, Password: wrongpass	Login fails, error message is displayed.	Login failed, incorrect password.	Pass

4	Update profile information	Name: John Doe, Profile photo: Upload photo, Bio: "Lover of cats"	Profile is updated successfully, with updated details displayed.	Profile updated, new details visible.	Pass
5	Delete user account	Username: khanben123@gmail.com, Password: saifullah	Account is deleted, and user is logged out.	Account deleted, logged out.	Pass
6	Change password	Old Password: saifullah, New password: 5678	Password is changed successfully.	Password updated successfully.	Pass
7	Forgot password	Email: khanben123@gmail.com@example.com	Password reset link is sent to the email.	Reset link sent to email.	Pass

6.2.2 Module 2: Cat Breed Detection

No.	Test case / Test script	Attribute and value	Expected result	Actual result	Result
1	Upload image for breed detection	Image: Upload a cat photo	Image is successfully uploaded and breed is detected.	Breed detected successfully.	Pass
2	Train model with new dataset	Dataset: Cat breeds dataset	Model is trained with new dataset and updated.	Model trained successfully.	Pass
3	Detect breed using ResNet-50	Image: Cat photo	Breed is detected with a confidence score.	Breed detected with confidence.	Pass
4	Feedback on breed recognition	Image: Cat photo, Feedback: "Accurate"	Feedback is submitted and stored.	Feedback submitted successfully.	Pass

6.2.3 Module 3: Online Store

No.	Test case / Test script	Attribute and value	Expected result	Actual result	Result
1	Buy a cat breed	Product: Maine Coon, User: Registered	Cat breed is added to cart and checkout page is shown.	Cat breed added to cart.	Pass
2	Buy cat food item	Product: Salmon Cat Food, User: Registered	Cat food item is added to cart and checkout page is shown.	Cat food added to cart.	Pass
3	Add items to shopping cart	Items: Maine Coon, Salmon Cat Food	Items are added to cart successfully.	Items added to cart.	Pass
4	Remove items from shopping cart	Items: Maine Coon	Item is removed from cart successfully.	Item removed from cart.	Pass

6.2.4 Module 4: Search Filters and Notifications

No.	Test case / Test script	Attribute and value	Expected result	Actual result	Result
1	Search for a cat breed	Search term: "Persian"	Results showing Persian cats are displayed.	Persian cat breed displayed.	Pass
2	Apply breed filter	Filter: "Maine Coon"	Only Maine Coon breed is shown in results.	Results filtered to Maine Coon.	Pass
3	Sort search results by price	Sort by: Price (ascending)	Products are sorted by price in ascending order.	Products sorted by price.	Pass
4	Search for cat food items	Search term: "Salmon food"	Results showing Salmon-based cat food are displayed.	Salmon food displayed.	Pass
5	Receive notification for order status	Event: Order shipped	Notification is displayed about order shipment.	Order shipped notification received.	Pass
6	Receive maintenance alert	Event: Upcoming app maintenance	Alert about app maintenance is displayed.	Maintenance alert received.	Pass

6.2.5 Module 5: Payment Processing

No.	Test case / Test script	Attribute and value	Expected result	Actual result	Result
1	Checkout cart	Cart: Maine Coon, Salmon Cat Food	Cart is ready for checkout, and user is redirected to payment page.	Checkout successful, redirected to payment page.	Pass
2	Process payment	Payment method: Credit Card	Payment is processed and order is confirmed.	Payment successful, order confirmed.	Pass
3	Cancel order	Order ID: 12345	Order is canceled successfully and status is updated.	Order canceled, status updated.	Pass
4	Modify order	Order ID: 12345, New item: Persian Cat Food	Order is updated with new items.	Order updated successfully.	Pass
5	Review purchased items	Item: Salmon Cat Food, Rating: 5 stars	Review is submitted successfully.	Review submitted.	Pass

6.2.6 Module 6: AI Chatbot

No.	Test case / Test script	Attribute and value	Expected result	Actual result	Result

1	Submit a query	Query: "What is the best food for cats?"	Chatbot responds with a relevant answer.	Chatbot response received.	Pass
2	Engage in conversation	Message: "Tell me about Maine Coon"	Chatbot provides a detailed answer about Maine Coon breed.	Chatbot response received.	Pass
3	Submit command	Command: "Order Maine Coon food"	Chatbot confirms order for Maine Coon food.	Order confirmed.	Pass
4	Provide feedback on chatbot response	Feedback: "Helpful"	Feedback is recorded and stored.	Feedback stored successfully.	Pass

6.2.7 Module 7: Admin Panel

No.	Test case / Test script	Attribute and value	Expected result	Actual result	Result
1	Add new user account	Username: admin01, Email: admin01@example.com	User account is added successfully.	User account added.	Pass
2	Modify user account	Username: admin01, New email: newadmin@example.com	User account is updated successfully.	User account updated.	Pass
3	Manage cat food inventory	Product: "Tuna Cat Food", Stock: 100	Inventory is updated successfully.	Inventory updated.	Pass
4	View and manage orders	Order ID: 12345, Status: Shipped	Order status is updated to "Shipped".	Order status updated.	Pass
5	View user feedback on model predictions	Feedback: "Accurate"	User feedback is displayed and stored.	Feedback displayed.	Pass

6.2.8 Module 8: Statistical Analytics

No.	Test case / Test script	Attribute and value	Expected result	Actual result	Result
1	Generate popular breed report	Breed: Maine Coon	Report showing sales and popularity of Maine Coon.	Report generated.	Pass
2	Track product performance	Product: "Persian Cat "	Performance data for the product is shown (sales, revenue).	Performance data displayed.	Pass
3	View real-time sales dashboard	Action: View dashboard	Sales performance dashboard is displayed with real-time data.	Dashboard displayed.	Pass

4	Monitor inventory levels	Product: "Siamese Cat Food", Stock: 50	Inventory level for "Siamese Cat Food" is displayed as 50.	Inventory displayed.	Pass
---	--------------------------	--	--	----------------------	------

6.3 Business Rules Testing

6.3.1 Decision Table for Product Purchase

Conditions	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8
Is Customer Logged In	T	T	T	T	T	F	T	T
Cat Food in Stock	T	T	T	T	T	T	F	T
Valid Payment Method	T	T	T	T	F	T	-	T
Enough Money Available	T	T	T	F	F	T	-	F
Coupon Applied	T	F	F	F	F	-	-	-
Sale Activated	F	F	F	T	F	-	-	-
Is Banned	F	F	T	F	F	F	F	-
Actions								
Product Purchased	T	T	F	F	F	F	F	F
Product Discounted	T	F	F	T	F	-	-	-
Customer Authorized	T	T	F	T	T	T	T	T

6.3.2 Decision Table for Admin Management

6.3.2.1 Table: Admin Conditions Decision Table

Conditions	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8
Category Available	T	F	T	T	F	F	F	T
Product Required Fields	T	T	F	T	F	F	T	F
Is Admin Applicable	T	T	T	F	F	T	F	F
Rules								
Product Added	T	F	F	F	F	F	F	F
Admin Authorized	T	T	T	F	F	T	F	F

6.3.3 Test Cases for Product Purchase Decision Table

Test Case ID	Conditions	Expected Result	Actual Result	Status
TC-1	Logged in, in stock, valid payment, enough money, coupon applied, no sale, not banned	Product purchased with discount.	As expected	Pass
TC-2	Logged in, in stock, valid payment, enough money, no coupon, no sale, not banned	Product purchased without discount.	As expected	Pass
TC-3	Logged in, in stock, valid payment, enough money, coupon applied, no sale, banned	Purchase not authorized.	As expected	Pass
TC-4	Logged in, in stock, valid payment, insufficient money, no coupon, sale active, not banned	Purchase not completed.	As expected	Pass
TC-5	Logged in, in stock, invalid payment, insufficient money, no coupon, no sale, not banned	Purchase not completed.	As expected	Pass

6.3.4 Test Cases for Admin Management Decision Table

Test Case ID	Conditions	Expected Result	Actual Result	Status
TC-1	Category available, valid fields, admin applicable	Product added successfully, admin authorized.	As expected	Pass
TC-2	Category unavailable, valid fields, admin applicable	Product not added, admin authorized.	As expected	Pass
TC-3	Category available, invalid fields, admin applicable	Product not added, admin authorized.	As expected	Pass
TC-4	Category available, valid fields, admin not applicable	Product not added, admin not authorized.	As expected	Pass
TC-5	Category unavailable, invalid fields, admin not applicable	Product not added, admin not authorized.	As expected	Pass

6.4 Integration Testing

6.4.1 Product Purchase and Online Store

Testing Objective: Ensure that purchasing a product updates the store inventory and the interface between modules *Product Purchase* and *Online Store* functions properly.

No.	Test Case/Test Script	Attribute and Value	Expected Result	Actual Result	Result
1	Purchase Product	Product: Cat Food, Stock: 10	Reduce inventory by 1 and mark the purchase as completed.	Inventory updated, purchase successful.	Pass
2	Out of Stock Product Purchase	Product: Cat Toy, Stock: 0	Display "Out of Stock" error message and do not proceed with the purchase.	Error message displayed successfully.	Pass
3	Cancel Purchase	Product: Cat Litter, Stock: 5, Canceled Order	Inventory remains unchanged, purchase not completed.	Inventory unchanged, purchase canceled.	Pass

6.4.2 Search Filters and Notifications

Testing Objective: Ensure that the search filters trigger appropriate notifications and the interface between modules *Search Filters* and *Notifications* is functional.

No.	Test Case/Test Script	Attribute and Value	Expected Result	Actual Result	Result
1	Search Filter Matched Product	Filter: Cat Food, Brand: Whiskas	Notification for matching product availability is triggered.	Notification triggered successfully.	Pass
2	Search Filter Unmatched Product	Filter: Cat Food, Brand: Non-Existent Brand	No notification is triggered, search results show no products.	No notification triggered, results empty.	Pass
3	New Stock Notification	Filter: Cat Food, Notify on new stock: Yes	Notification is sent when a new stock of the selected product is available.	Notification sent successfully.	Pass

6.4.3 Statistical Analytics and Payment Processing

Testing Objective: Verify that sales data is accurately updated in analytics reports after successful payment and the interface between modules *Payment Processing* and *Statistical Analytics* works correctly.

No.	Test Case/Test Script	Attribute and Value	Expected Result	Actual Result	Result
1	Successful Purchase Update Analytics	Product: Cat Collar, Price: \$20	Update analytics with the sale, including product name, price, and date of purchase.	Analytics updated successfully.	Pass
2	Failed Payment Does Not Update	Product: Cat Food, Payment: Declined	No update in analytics, sale not recorded.	No update in analytics.	Pass
3	Bulk Purchase Update	Multiple products purchased: Total \$100	Analytics updated with aggregated purchase data.	Analytics updated successfully.	Pass

6.4.4 User Authentication and AI Chatbot

Testing Objective: Ensure authenticated users can interact with the AI chatbot for relevant queries, and the chatbot retrieves data correctly.

No.	Test Case/Test Script	Attribute and Value	Expected Result	Actual Result	Result
1	Authenticated User Chatbot Interaction	Logged in: Yes, Query: "Best cat breed?"	Chatbot retrieves and displays information on popular cat breeds.	Information displayed successfully.	Pass
2	Unauthenticated User Chatbot Interaction	Logged in: No, Query: "Order status?"	Chatbot prompts the user to log in before providing order details.	Prompt displayed successfully.	Pass
3	Query for Out of Scope Information	Logged in: Yes, Query: "Weather forecast?"	Chatbot responds with a message indicating the query is outside its scope.	Message displayed successfully.	Pass

6.4.5 Search Filters and Online Store

Testing Objective: Verify that applying search filters displays accurate product results from the store module.

No.	Test Case/Test Script	Attribute and Value	Expected Result	Actual Result	Result
1	Filter Search by Price Range	Filter: Cat Food, Price Range: \$10–\$50	Display products within the price range.	Products displayed successfully.	Pass
2	Filter Search by Category	Filter: Category: Cat	Display only cat products.	Cat displayed successfully.	Pass
3	Combined Filters	Filter: Category: Cat Food, Price: \$20–\$40	Display products matching both category and price criteria.	Relevant products displayed.	Pass

6.4.6 Payment Processing and Notifications

Testing Objective: Ensure successful payment triggers notifications, and canceled payments do not.

No.	Test Case/Test Script	Attribute and Value	Expected Result	Actual Result	Result
1	Payment Success Notification	Payment: Successful	Notification is sent confirming the order and payment.	Notification sent successfully.	Pass
2	Payment Failure Notification	Payment: Failed	Notification is sent indicating payment failure.	Notification sent successfully.	Pass
3	Order Cancellation Notification	Payment: Successful, Order: Canceled	Notification is sent confirming the order has been canceled.	Notification sent successfully.	Pass

6.4.7 Admin Panel and Statistical Analytics

Testing Objective: Verify that actions taken by the admin update analytics appropriately.

No.	Test Case/Test Script	Attribute and Value	Expected Result	Actual Result	Result
1	Add Product Updates Analytics	Product: New Cat	Analytics reflect the addition of the new product.	Analytics updated successfully.	Pass
2	Remove Product Updates Analytics	Product: Outdated Cat Collar	Analytics reflect the removal of the product.	Analytics updated successfully.	Pass
3	Modify Product Updates Analytics	Product: Cat Food, New Price: \$25	Analytics reflect the updated price of the product.	Analytics updated successfully.	Pass

7 Conclusion and Future Work:

The FelineConnect project successfully tackles the issues that cat owners encounter by offering an AI-powered, user-centric platform for cat breed identification, online buying, and tailored support. The platform's revolutionary features, such as deep learning models, a seamless user interface, and powerful backend support, provide a comprehensive solution that improves pet care, streamlines e-commerce, and promotes community involvement.

The project met its primary objectives, which included the creation of modules for breed detection, e-commerce, payment processing, and analytics. It also has a scalable design and an easy-to-use user interface, laying a solid platform for future improvements.

Future development on FelineConnect may include:

Mobile Application Development: Extending platform accessibility to mobile devices to increase user engagement.

Advanced Analytics: Using predictive algorithms to forecast sales patterns and recommend inventory modifications.

Global Expansion: Adding language support to reach a larger, worldwide user base.

User Engagement Features: Introducing a social network where people may share their advice, stories, and experiences.

These enhancements will guarantee FelineConnect stays current, competitive, and impactful, fulfilling the changing needs of its consumers.

8 References

- <https://ieeexplore.ieee.org/document/8672684>
- <https://universe.roboflow.com/diffran-nur-cahyo-egmuz/cat-breed-classification>
- https://www.irjmets.com/uploadedfiles/paper/issue_5_may_2023/38683/final/fin_irjmets1683962083.pdf
- <https://core.ac.uk/download/pdf/552632009.pdf>
- <https://www.geeksforgeeks.org/cat-dog-classification-using-convolutional-neural-network-in-python/>
- <https://www.kaggle.com/datasets/timost1234/cat-individuals>
- <https://www.kaggle.com/datasets/yapwh1208/cats-breed-dataset>
- <https://www.kaggle.com/datasets/imsparsh/animal-breed-cats-and-dogs>

9 Plagiarism Report

