ĐẠI HỌC FPT

Koi Auction System
Software Requirement Specification

Project Code: KAS001

Document Code: KAS001-SRS-v1.0

Class Code: SE1857

Group Code: 1

Ho Chi Minh City, October 24, 2024

RECORD OF CHANGE

*A - Added M - Modified D - Deleted

Effective Date	Changed Items	A* M, D	Change Description	New Version
	Initial	a	Add project overview	
	Auction Business Rules	m	Modified rules for auction start time and bid process	
	Create Member Account	a	Added functionality to register and create member accounts	
	Create Auction Lots	a	Added functionality to create auction lots in the system	
	Manage Auctions	a	Added functionality for administrators to manage auctions	
	Payment Methods	a	Added MoMo integration for wallet top-up	
	Wallet System	a	Updated wallet balance after transactions	
	Bidding Logic	a	Updated real-time bid validation using SignalR	

SIGNATURE PAGE

ROLE/TITLE	NAME	SIGNATURE	DATE
Project Manager	Trần Nhật Thắng	THANG	20/10/2024
Technical Lead	Võ Gia Linh	GLINH	21/10/2024
Business Analyst	Nguyễn Thị Diệu Vi	VI	21/10/2024
QA Lead	Nguyễn Trình Cát Linh	CLINH	22/10/2024
Development Team Lead	Nguyễn Phi Long	LONG	24/10/2024

TABLE OF CONTENTS

1 Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to outline the requirements for developing the Koi Auction platform. This platform connects Koi fish buyers and sellers, offering a secure, real-time bidding experience, payment processing, and shipping services. The SRS defines key functionalities and operational requirements to ensure a smooth and user-friendly auction process, serving as a reference for all stakeholders to align on the system's goals and development.

1.2 Scope

The Koi Auction platform will enable secure online auctions for Koi fish, connecting buyers and sellers through real-time bidding. It supports multiple auction methods, including fixed-price, single-bid, ascending-bid, and descending-price auctions. The platform handles user registration, auction management, and secure payment processing via an integrated wallet system, along with shipping coordination.

Key features include:

- Real-time bidding with live updates.
- Payment integration (Zalo pay) for wallet top-ups and withdrawals.
- User account and balance management.
- Notifications for auction events.

The platform is designed to be user-friendly, scalable, and accessible via web browsers, ensuring smooth auction experiences for all users.

1.3 Definitions, Acronyms, and Abbreviations

Term/Acronym	Definition
SRS	Software Requirements Specification
ZaloPay	A mobile payment platform used for top-ups and withdrawals in the auction system
Real-time Bidding	A bidding process where bids are updated instantly for all participants
Fixed-price method	Users purchase Koi fish available at a fixed price.
Single-bid method	Users place a secret bid above the minimum price, and at the end of the auction, the highest-bid user wins.
Ascending-bid method	Users place bids higher than the current bid with a minimum step. The countdown resets with each new bid placed, and the highest-bid user wins once no more bids.
Descending-bid method	Users follow Koi fish auctions with prices decreasing at regular intervals. The first user to accept the current price is the winner.
lot	An object that are offered for sale at auction

1.4 References

- *IEEE Recommended Practice for Software Requirements Specifications," in IEEE Std 830-1998, vol., no., pp.1-40, 20 Oct. 1998.*
- *Node.js Documentation*
- *React Documentation*
- *ASP.NET Documentation*
- *Zalo pay API Documentation*
- *JSON Web Token Document*

1.5 Overview

The structure of the document is divided into four main parts:

Part 1: Introduction

Provides an overview of the purpose, scope, and structure of the document. It introduces the target audience and the scope of the system.

Part 2: Overall Description

This section of the Software Requirements Specification (SRS) provides a high-level overview of the product and outlines the general factors influencing its design and requirements.

Part 3: Functional Requirements

Describes in detail the factors, constraints, user characteristics, and operating environments that impact the system. This section focuses on outlining the specific functional requirements, ensuring that the software developers have all the necessary information to implement the required features.

Part 4: Non-Functional Requirements

Describes the non-functional requirements, including performance, security, scalability, and usability of the system, which are crucial to ensuring the overall quality of the software.

2 Overall Description

2.1 Product Perspective

The Koi Auction Platform is the next addition to a growing real-time bidding product line, designed specifically for Koi fish auctions. It provides a tailored online auction environment for Koi fish enthusiasts, offering real-time bidding, secure payments, and shipping services, ensuring a seamless experience for both buyers and sellers.

2.2 User classes and characteristics

1. **Users**:

   Description: Users are users who participate in auctions to purchase Koi fish. They may range from casual hobbyists to serious collectors or breeders.

   Characteristics:

   ○ Primarily interested in viewing, bidding on, and purchasing Koi fish in real-time.
   ○ Require simple, intuitive navigation and secure payment options to ensure a smooth buying experience.
   ○ Require notifications for auction status updates (e.g., bid confirmations, auction results) are essential.

2. **Breeders**:

   Description: Breeders are users who request their Koi fish for bidding on the platform.

   Characteristics:

   ○ Experts in Koi fish breeding, quality, and pricing
   ○ Require functionality for managing listings, setting auction prices, and monitoring bid activity.

- Require secure and reliable ways to manage transactions, including receiving payment for sold fish.
- Require assistance with coordinating shipping and delivery logistics.

3. **Staff**:

   Description: Staffs are platform supporters responsible for managing daily operations and assisting users.

   Characteristics:

   - Oversee auction processes to ensure smooth and efficient operations.
   - Handle user inquiries and provide support for any technical or auction-related issues.
   - Monitor auction activities to identify and address any irregularities or problems during live auctions

4. **Administrators**:

   Description: Administrators are responsible for managing the platform, monitoring auctions, resolving disputes, and handling system maintenance.

   Characteristics:

   - Technically skilled users familiar with system management, user account administration, and troubleshooting.
   - Require tools for monitoring auction activity, managing user accounts, and ensuring that the platform operates smoothly.
   - Require detailed access to transaction histories and bid activities for oversight and fraud prevention.
   - Require a comprehensive dashboard is crucial for administrators to analyze auction data, user activity, and system performance, aiding in decision-making and system improvements.

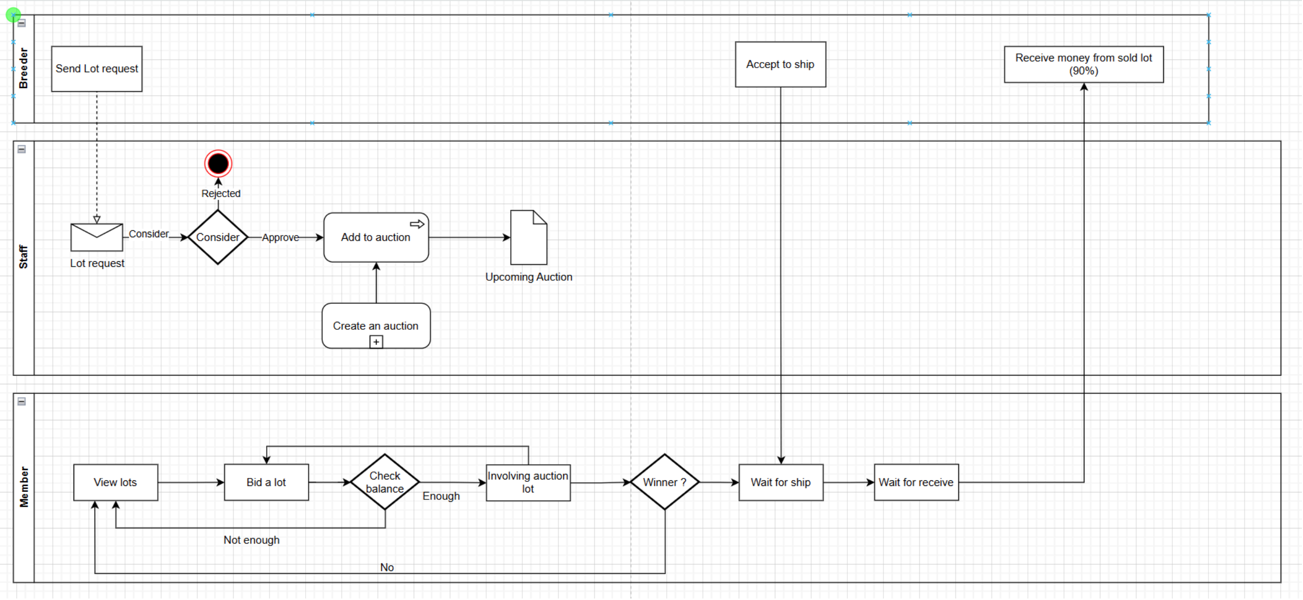
2.3 Design and implementation constraints

Developers should also be careful about the privacy of users. All user data will be kept on database and necessary precautions should be taken to protect user data.

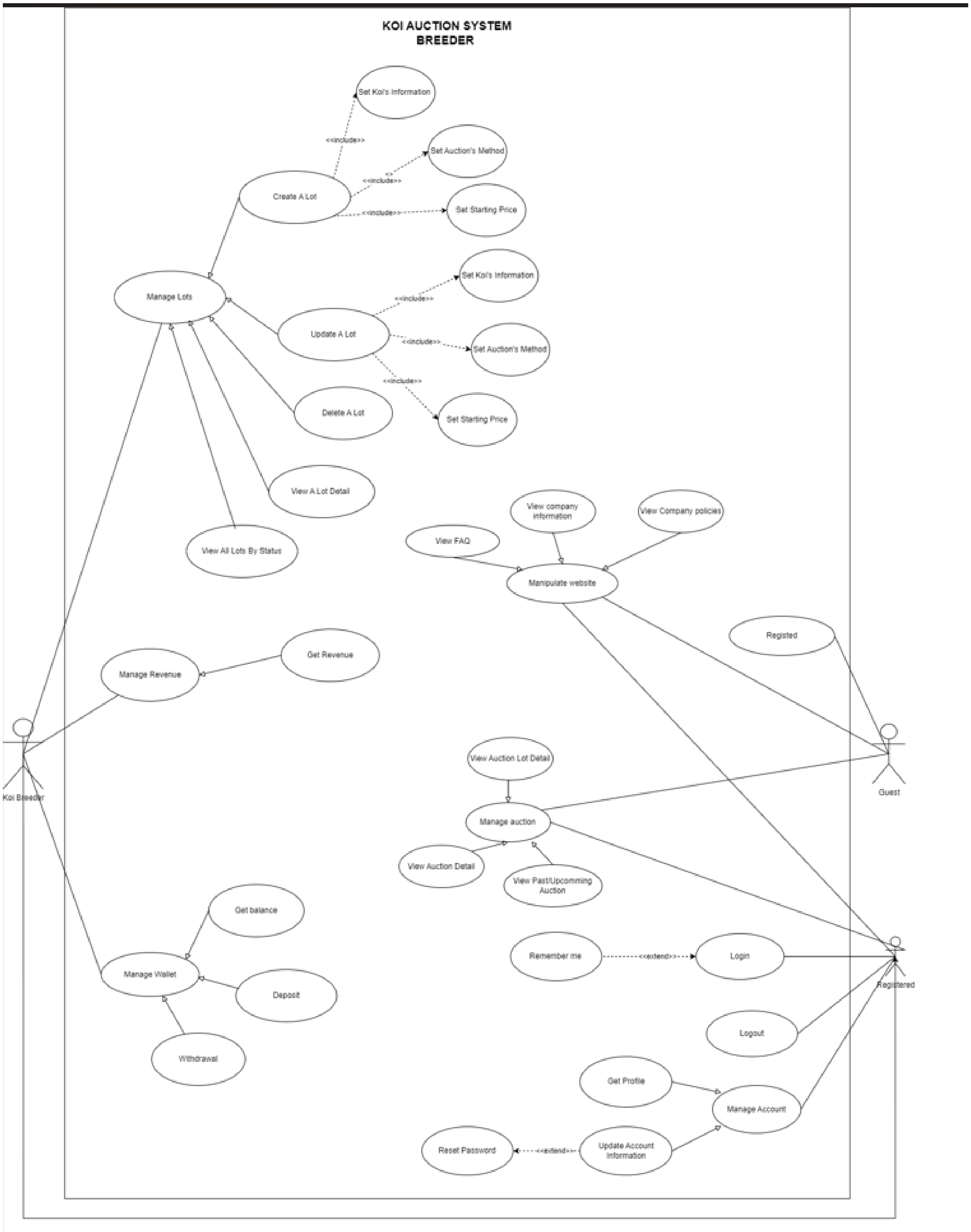
The Internet connection is a constraint for the application. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to function.

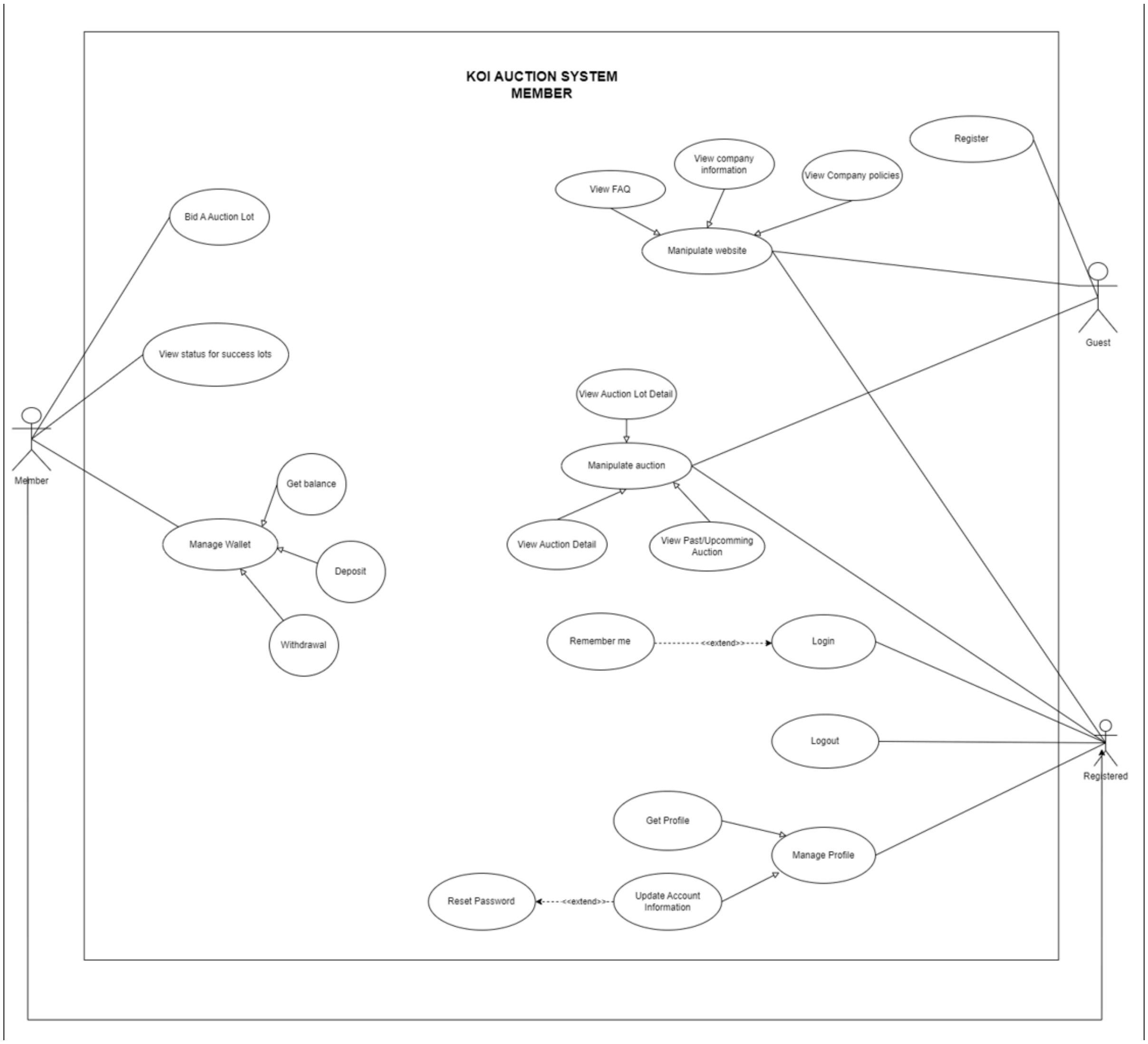
3 FUNCTIONAL REQUIREMENTS

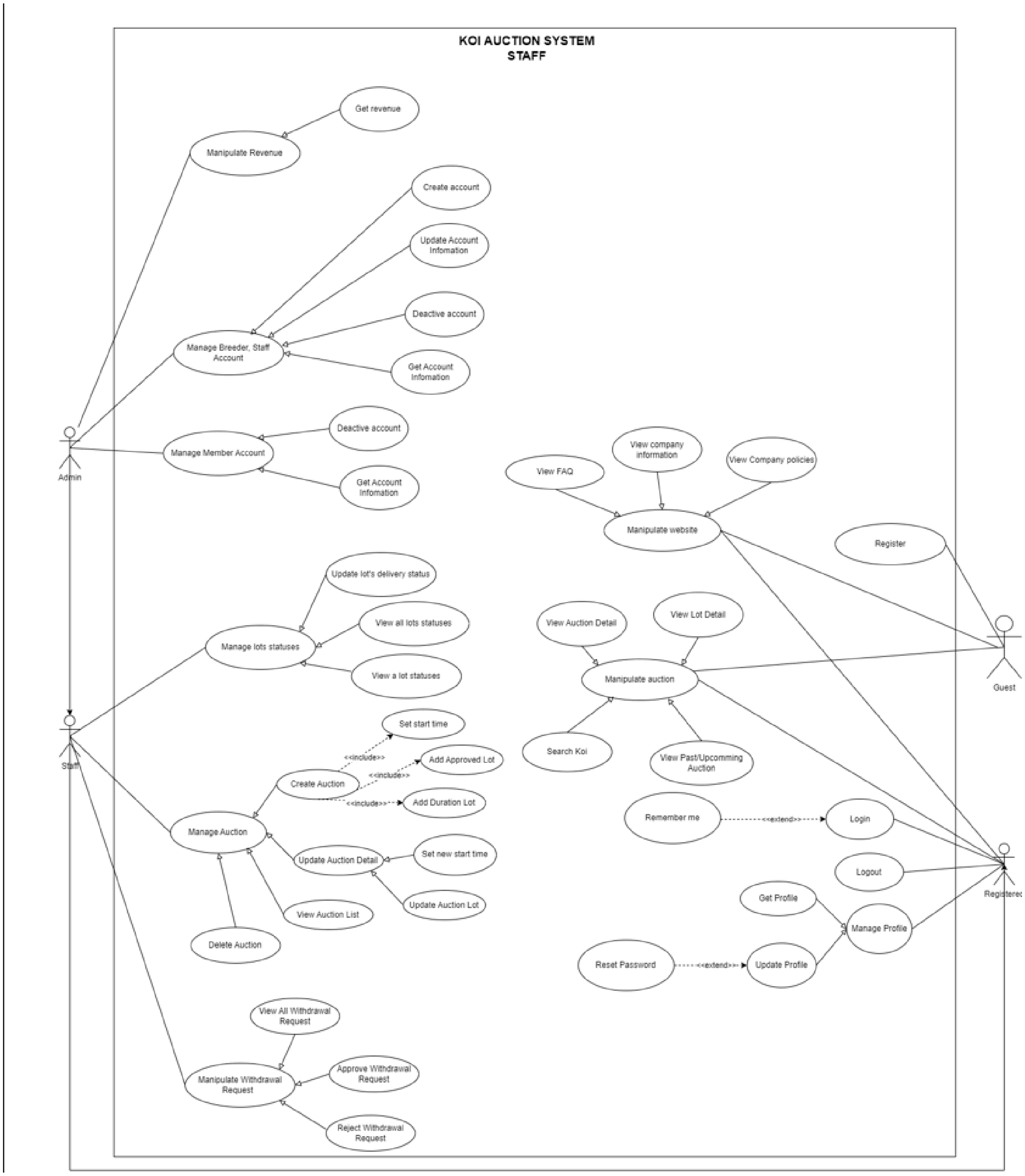
3.1 Swimlane Diagrams



3.2 Use Case Diagrams




KOI AUCTION SYSTEM
MEMBER
Register
View company information
View Company policies
View FAQ
Bid A Auction Lot
Manipulate website
Guest
View status for success lots
View Auction Lot Detail
Member
Manipulate auction
Get balance
Manage Wallet
View Auction Detail
View Past/Upcomming Auction
Deposit
Withdrawal
Remember me
<<extend>>
Login
Logout
Registered
Get Profile
Manage Profile
Reset Password
<<extend>>
Update Account Information


KOI AUCTION SYSTEM
STAFF
Get revenue
Manipulate Revenue
Create account
Update Account Infomation
Deactive account
Manage Breeder, Staff Account
Get Account Infomation
Deactive account
Manage Member Account
Get Account Infomation
Admin
View company information
View FAQ
View Company policies
Manipulate website
Register
Update lot's delivery status
View all lots statuses
Manage lots statuses
View a lot statuses
View Auction Detail
View Lot Detail
Manipulate auction
Guest
Staff
Set start time
<<include>>
Add Approved Lot
<<include>>
Create Auction
<<include>>
Add Duration Lot
Search Koi
View Past/Upcomming Auction
Manage Auction
Update Auction Detail
Set new start time
Remember me
<<extend>>
Login
Logout
Registered
View Auction List
Update Auction Lot
Get Profile
Manage Profile
Delete Auction
Reset Password
<<extend>>
Update Profile
View All Withdrawal Request
Manipulate Withdrawal Request
Approve Withdrawal Request
Reject Withdrawal Request

Use Case 1: Create A Lot

Use-case No.	UC001	Use-case Version	1.0
Use-case Name	Create A Lot		
Author	Trần Nhật Thắng		
Date	24/10/2024	Priority	High

Actor:

Breeder

System

Summary:

The Breeder creates a lot (Koi fish lot) to prepare for the auction. This process includes entering required information about the Koi fish, the auction method, and the starting price.

Goal:

The Breeder can successfully create a lot with complete information in preparation for the auction.

Triggers

The Breeder selects the "Create A Lot" function from the system.

Preconditions:

The Breeder is logged into the system.

The Breeder has permission to create a new lot.

Post Conditions:

The lot is successfully created and stored in the system.

Main Success Scenario:

The Breeder selects the "Create A Lot" function.

The system displays fields for required information for the new lot.

The Breeder enters information about the Koi fish, the auction method, and the starting price.

The Breeder confirms to create the lot.

The system saves the new lot and displays a success notification.

Alternative Scenario:

3a. If the Breeder does not provide all required information, the system displays an error message and prompts for re-entry.

Exceptions:

If the system is unavailable, the Breeder will not be able to create the lot, and the system will display an error message.

Relationships:

Include: Set Koi's Information, Set Auction's Method, Set Starting Price

Business Rules:

BR-01: Lot Information Completeness
All mandatory fields (Koi's information, auction method, and starting price) must be completed before the lot can be created. If any field is left blank, the system will prevent the lot from being created and will display an error message.

BR-02: Unique Lot Identifier
Each lot created must have a unique identifier (e.g., Lot ID) assigned by the system to distinguish it from other lots.

BR-03: Starting Price Validation
The starting price for the lot must be a positive number and cannot be lower than the minimum threshold set by the auction system. If the entered starting price is below this threshold, the system will display a warning and prevent the lot from being created.

Use Case 2: Update A Lot

Use-case No.	UC002	Use-case Version	1.0
Use-case Name	Update A Lot		
Author	Trần Nhật Thắng		
Date	24/10/2024	Priority	High

Actor:

Breeder

System

Summary:

The Breeder updates the information for an existing lot, such as modifying the Koi fish details, adjusting the auction method, or changing the starting price if needed.

Goal:

The Breeder can successfully update the lot details to reflect accurate and up-to-date information.

Triggers

The Breeder selects the "Update A Lot" function from the system.

Preconditions:

The Breeder is logged into the system.

The Breeder has permission to update the lot.

The lot exists in the system and is accessible to the Breeder for editing.

Post Conditions:

The lot information is successfully updated and saved in the system.

Main Success Scenario:

The Breeder selects the "Update A Lot" function for the desired lot.

The system displays the current information of the lot.

The Breeder makes the necessary changes, such as editing the Koi fish details, auction method, or starting price.

The Breeder confirms the updates.

The system saves the updated lot information and displays a success notification.

Alternative Scenario:

3a. If the Breeder enters invalid data (e.g., negative starting price), the system displays an error message, and the Breeder is prompted to re-enter valid information.

Exceptions:

If the system is unavailable, the Breeder will not be able to update the lot, and the system will display an error message.

If the lot has already been approved for auction, the Breeder may only update limited fields or request approval for changes.

Relationships:

Include: Set Koi's Information, Set Auction's Method, Set Starting Price

Business Rules:

BR-04: Unique Lot Identifier
Each lot created must have a unique identifier (e.g., Lot ID) assigned by the system to distinguish it from other lots.

BR-05: Starting Price Validation
The starting price for the lot must be a positive number and cannot be lower than the minimum threshold set by the auction system. If the entered starting price is below this threshold, the system will display a warning and prevent the lot from being created.

Use Case 3: Delete A Lot

Use-case No.	UC003	Use-case Version	1.0
Use-case Name	Delete A Lot		
Author	Trần Nhật Thắng		
Date	24/10/2024	Priority	High

Actor:

Breeder

System

Summary:

The Breeder deletes a lot that is no longer needed or if the Koi fish will not be auctioned. This process permanently removes the lot and its associated data from the system, provided the lot has not been scheduled for an upcoming auction.

Goal:

The Breeder can successfully delete a lot that they no longer wish to auction.

Triggers

The Breeder selects the "Delete A Lot" function from the system.

Preconditions:

The Breeder is logged into the system.

The Breeder has permission to delete the lot.

The lot exists in the system and is not scheduled for an auction

Post Conditions:

The lot is permanently removed from the system.

Main Success Scenario:

The Breeder selects the "Delete A Lot" function for the chosen lot.

The system displays a confirmation prompt to verify the deletion request.

The Breeder confirms the deletion.

The system deletes the lot and displays a success notification.

Alternative Scenario:

If the lot is scheduled for an upcoming auction, the system displays a warning message stating that the lot cannot be deleted.

If the Breeder cancels the deletion at the confirmation prompt, the system retains the lot data, and no changes are made.

Exceptions:

If the system is unavailable, the Breeder will not be able to delete the lot, and the system will display an error message.

Relationships:

None

Business Rules:

Use Case 4: Delete A Lot

Use-case No.	UC003	Use-case Version	1.0
Use-case Name	Delete A Lot		
Author	Trần Nhật Thắng		
Date	24/10/2024	Priority	High

Actor:

Breeder

System

Summary:

The Breeder deletes a lot that is no longer needed or if the Koi fish will not be auctioned. This process permanently removes the lot and its associated data from the system, provided the lot has not been scheduled for an upcoming auction.

Goal:

The Breeder can successfully delete a lot that they no longer wish to auction.

Triggers

The Breeder selects the "Delete A Lot" function from the system.

Preconditions:

The Breeder is logged into the system.

The Breeder has permission to delete the lot.

The lot exists in the system and is not scheduled for an auction

Post Conditions:

The lot is permanently removed from the system.

Main Success Scenario:

The Breeder selects the "Delete A Lot" function for the chosen lot.

The system displays a confirmation prompt to verify the deletion request.

The Breeder confirms the deletion.

The system deletes the lot and displays a success notification.

Alternative Scenario:

If the lot is scheduled for an upcoming auction, the system displays a warning message stating that the lot cannot be deleted.

If the Breeder cancels the deletion at the confirmation prompt, the system retains the lot data, and no changes are made.

Exceptions:

If the system is unavailable, the Breeder will not be able to delete the lot, and the system will display an error message.

Relationships:

None

Business Rules:

Use Case 5: View A Lot Detail

Use-case No.	UC005	Use-case Version	1.0
Use-case Name	View A Lot Detail		
Author	Trần Nhật Thắng		
Date	24/10/2024	Priority	High

Actor:

Breeder

System

Summary:

The Breeder views detailed information about a specific lot, including the Koi fish characteristics, auction method, starting price, and current auction status.

Goal:

The Breeder can access and review detailed information about a specific lot.

Triggers

The Breeder selects the "View A Lot Detail" function from the system.

Preconditions:

The Breeder is logged into the system.

The lot exists in the system and is accessible to the Breeder.

Post Conditions:

The detailed information for the selected lot is displayed to the Breeder.

Main Success Scenario:

The Breeder selects the "View A Lot Detail" function.

The system retrieves and displays the full details of the lot.

Alternative Scenario:

1a. If the lot does not exist or has been deleted, the system displays an error message indicating that the lot is not available.

Exceptions:

If the system is unavailable, the Breeder will not be able to view lot details, and the system will display an error message.

Relationships:

None

Use Case 6: Manage Auction			
Use-case No.	UC001	Use-case Version	1.0
Use-case Name	Manage Auction		
Author	Nguyễn Thị Diệu Vi		
Date	24/10/2024	Priority	High

Actor: Staff

Summary:

Staff can create, update, delete, and view auction details. This use case allows staff to manage the overall auction process, including setting start times, adding approved lots, and updating auction information.

Goal:

Enable staff to manage auctions effectively, ensuring each auction is scheduled correctly and contains the necessary details for smooth operation.

Triggers:

Staff initiates the use case to create a new auction, update existing auction information, delete an auction, or view auction details.

Preconditions:

- Staff must be logged in and have the necessary permissions to manage auctions.
- The auction being managed must exist in the system database for update or deletion.

Post Conditions:

- The auction is successfully created, updated, deleted, or retrieved as requested by the staff.
- Any changes to auction details are logged for audit purposes.
- Changes reflect in the frontend interface for users to view updated auction information.

Main Success Scenario:

1. Staff selects the "Manage Auction" option.
2. The system displays available actions: create, update, delete, and view auction details.
3. Staff chooses an action:
    - **Create Auction**: Staff enters details, including auction name, start time, and any approved lots. The system validates and saves the new auction.
    - **Update Auction**: Staff selects an auction, modifies details such as start time or lot duration, and saves changes.
    - **Delete Auction**: Staff selects an auction to delete. The system confirms and removes it from the database.
    - **View Auction Details**: Staff selects an auction to view its details, including start time, lot details, and status.
4. The system confirms the action and displays a success message.

Alternative Scenario:

- **A1**: If the auction does not exist (for update or deletion), the system notifies staff and suggests creating a new auction if required.
- **A2**: If the staff does not have permission, the system displays an authorization error.

Exceptions:

- **E1**: If validation fails during auction creation or update (e.g., missing required fields like start time), the system displays an error message.
- **E2**: If the auction is already in progress, the system restricts deletion and notifies staff.

Relationships:

- Relates to "Manage Lot Statuses" and "View Auction Detail" use cases, as auction management includes setting lot start times and viewing auction statuses.

Business Rules:

- Only authorized staff can manage auctions.
- Auctions can only be deleted if they are not currently active.
- Each auction must have a unique name and start time.

- Every change to an auction must be logged with a timestamp and staff ID for auditing purposes.
- Staff must ensure that lots are approved and available before adding them to an auction.

Use Case 7: Manage Lot Status

Use-case No.	UC001	Use-case Version	1.0
Use-case Name	Manage Lot Status		
Author	Nguyễn Thị Diệu Vi		
Date	24/10/2024	Priority	High

Actor: Staff

Summary:

Staff can update the status of individual lots within an auction. This includes setting start times, updating delivery statuses, and viewing the current status of lots.

Goal:

Allow staff to manage and monitor the status of each lot in an auction, ensuring a smooth transition between different phases of the auction process.

Triggers:

Staff initiates this use case to update or monitor the status of lots in an auction, especially before, during, or after an auction event.

Preconditions:

- Staff must be logged in and have the necessary permissions to manage lot statuses.

- The lots should be associated with an active auction for status management.

Post Conditions:

- The lot status is successfully updated as requested by the staff.
- Any status changes are logged for audit purposes.
- Changes are reflected in the frontend interface for real-time updates to users.

Main Success Scenario:

1. Staff selects the "Manage Lot Status" option.
2. The system displays a list of lots associated with the selected auction.
3. Staff selects a lot and chooses an action:
    - **Set Start Time**: Staff sets the start time for the lot. The system validates the timing in relation to the auction schedule.
    - **Update Delivery Status**: Staff updates the delivery status (e.g., "Pending", "In Transit", "Delivered") after the auction is completed.
    - **View Lot Status**: Staff views the current status and history of actions taken on the lot.
4. The system confirms the action and displays a success message.

Alternative Scenario:

- **A1**: If the selected lot is not associated with an active auction, the system notifies staff and restricts updates until the lot is scheduled in an active auction.

Exceptions:

- **E1**: If validation fails for the start time (e.g., conflicts with another lot's timing), the system displays an error message.
- **E2**: If the delivery status is updated for an undelivered lot that has not been won, the system notifies the staff to verify status.

Relationships:

- Related to "Manage Auction" as it involves handling lots within an auction.
- Relates to "View Lot Detail" since managing status often involves viewing detailed information about the lot.

Business Rules:

- Only authorized staff can update lot statuses.
- Start times for lots must be sequential and cannot overlap with the active time of another lot in the same auction.
- Delivery status updates are only allowed once an auction has ended and the lot has been won.
- All changes to lot statuses must be logged with timestamps and staff IDs for auditing purposes.

Use Case 8: Create A Auction			
Use-case No.	UC001	Use-case Version	1.0
Use-case Name	Create Auction		
Author	Nguyễn Thị Diệu Vi		
Date	24/10/2024	Priority	High

Actor: Staff

Summary:

Staff can create a new auction by setting up the auction details, such as auction name, start time, and adding approved lots. This use case allows staff to prepare auctions for upcoming events, ensuring all necessary information is configured correctly.

Goal:

Enable staff to create a new auction with all relevant details, allowing it to be scheduled and visible for users interested in participating.

Triggers:

Staff initiates this use case to set up a new auction event.

Preconditions:

- Staff must be logged in and have the necessary permissions to create auctions.
- Approved lots must be available in the system to add to the auction.

Post Conditions:

- The auction is successfully created with all required details.
- The auction is marked as "upcoming" and scheduled to be visible in the user interface.

Main Success Scenario:

1. Staff selects the "Create Auction" option.
2. The system prompts the staff to enter auction details, including:
    - **Auction Name**: Unique identifier for the auction.
    - **Start Time**: Scheduled time for the auction to begin.
3. Staff reviews and confirms the auction details.
4. The system prompts the staff to add approved lots to the auction. Staff selects available lots and sets:
    - **Order of Lots**: Sequence in which lots will be auctioned.
    - **Duration for Each Lot**: Time allocated for bidding on each lot.
5. Staff reviews and confirms the lot details.
6. The system validates the entered information and saves the new auction as "upcoming."
7. The system confirms the auction creation and displays a success message.

Alternative Scenario:

- **A1**: If there are no approved lots available, the system displays a message indicating that lots need to be approved before creating the auction.
- **A2**: If the entered auction name is not unique, the system prompts the staff to enter a different name.

Exceptions:

- **E1**: If validation fails for the start time (e.g., overlaps with another scheduled auction), the system displays an error message.
- **E2**: If the required fields (e.g., auction name, start time) are missing, the system prompts the staff to complete all fields.

Relationships:

- Related to "Manage Auction" as it involves setting up auctions that will later be managed.
- Relates to "Manage Lot Status" since creating an auction includes adding lots that will need status management.

Business Rules:

- Only authorized staff can create auctions.
- Each auction must have a unique name and start time.
- The auction cannot start unless there is at least one approved lot assigned to it.
- Lot durations must align with the auction start time to avoid conflicts.
- All actions related to auction creation must be logged with a timestamp and staff ID for auditing purposes.

Use Case 9: Update Auction Lot Detail

Use-case No.	UC001	Use-case Version	1.0
Use-case Name	Update Auction Lot Detail		
Author	Nguyễn Thị Diệu Vi		
Date	24/10/2024	Priority	High

Actor: Staff

Summary:

Staff can update details of an auction lot, including information such as the starting price, duration, step percentage, and start time. This use case enables staff to make necessary adjustments to individual auction lots before or during an auction session.

Goal:

Allow staff to update the details of an auction lot to ensure accurate and relevant information is available for users participating in the auction.

Triggers:

Staff initiates this use case when there is a need to modify details of an auction lot, either to correct information or to adjust lot parameters for the auction.

Preconditions:

- Staff must be logged in and have the necessary permissions to update auction lots.
- The auction lot must exist in the system and be part of an active or upcoming auction.

Post Conditions:

- The auction lot details are updated successfully.
- Changes are saved and reflected in the frontend interface for real-time updates to users.
- Changes are logged for audit purposes.

Main Success Scenario:

1. Staff selects the "Update Auction Lot Detail" option.
2. The system displays a list of auction lots associated with the selected auction.
3. Staff selects a specific auction lot to update.
4. The system displays the current details of the lot, including:
    - **Starting Price**: The initial bidding price for the lot.
    - **Duration**: The time allocated for bidding on this lot.
    - **Step Percentage**: The bid increment percentage.
    - **Start Time**: Scheduled start time for this lot.
5. Staff updates one or more details as needed.
6. The system validates the changes (e.g., ensuring start times don’t overlap).
7. Staff confirms the changes.
8. The system saves the updated details and displays a success message.

Alternative Scenario:

- **A1**: If the auction lot is already closed, the system restricts updates and notifies the staff that closed lots cannot be modified.
- **A2**: If the start time conflicts with another lot’s timing in the same auction, the system prompts the staff to select a different time.

Exceptions:

- **E1**: If required fields (e.g., starting price or duration) are missing, the system prompts the staff to complete all mandatory fields.
- **E2**: If validation fails for any detail (e.g., incorrect duration format), the system displays an error message.

Relationships:

- Related to "Manage Auction" as updating auction lot details is part of managing the auction process.
- Relates to "Manage Lot Status" since updating details can affect the lot’s scheduling and timing within the auction.

Business Rules:

- Only authorized staff can update auction lot details.
- Changes to the lot’s start time must align with the auction schedule to prevent overlap with other lots.
- The starting price, duration, and step percentage must adhere to predefined system limits.
- All updates to auction lot details must be logged with a timestamp and staff ID for auditing purposes.

Use Case 10: Manipulate Withdrawal Request

Use-case No.	UC001	Use-case Version	1.0
Use-case Name	Manipulate Withdrawal Request		
Author	Nguyễn Thị Diệu Vi		
Date	24/10/2024	Priority	High

Actor: Staff

Summary:

Staff can review and approve or reject withdrawal requests submitted by users. This use case ensures that withdrawal requests are processed accurately, securely, and promptly, maintaining user trust and system integrity.

Goal:

Enable staff to manage withdrawal requests by either approving or rejecting them based on the user’s account balance and other predefined criteria.

Triggers:

A user submits a withdrawal request, prompting staff to review and process the request.

Preconditions:

- Staff must be logged in and have the necessary permissions to handle withdrawal requests.
- The withdrawal request must exist in the system and be in a "Pending" status.

Post Conditions:

- The withdrawal request is either approved or rejected, with the user notified of the decision.
- The user’s account balance is updated accordingly if the request is approved.
- The action is logged for auditing purposes.

Main Success Scenario:

1. Staff selects the "Manipulate Withdrawal Request" option.
2. The system displays a list of pending withdrawal requests.
3. Staff selects a specific withdrawal request to review.
4. The system displays the request details, including:
   - **Request Amount**: The amount the user wants to withdraw.
   - **User Account Balance**: Current balance of the user’s account.
   - **Request Date**: Date and time of the withdrawal request.
5. Staff reviews the request and either:
   - **Approve the Request**: Staff confirms the approval, and the system deducts the request amount from the user’s balance and updates the request status to "Approved."
   - **Reject the Request**: Staff provides a reason for rejection, and the system updates the request status to "Rejected."
6. The system notifies the user of the decision and displays a success message to the staff.

Alternative Scenario:

- **A1**: If the user’s balance is insufficient to cover the withdrawal request, the system restricts approval and notifies staff to reject the request.

Exceptions:

- **E1**: If the withdrawal request is no longer pending (e.g., already approved or rejected), the system notifies staff and restricts further actions.
- **E2**: If there is a system error when updating the user’s balance, the system displays an error message and logs the issue for resolution.

Relationships:

- Related to "Manage User Account" as it involves handling user account balances.
- Relates to "View Withdrawal Request" since staff might need to view request details before taking action.

Business Rules:

- Only authorized staff can approve or reject withdrawal requests.
- Staff must verify that the user’s account has sufficient funds to cover the requested withdrawal amount.
- Each action (approval or rejection) must be logged with the timestamp, staff ID, and any comments provided for auditing purposes.
- Rejected requests require a reason to be provided, which is communicated to the user.

User use case:

Use Case 11: Register			
Use-case No.	UC002	Use-case Version	1.0
Use-case Name	Register		
Author	Võ Gia Linh		
Date	24/10/2024	Priority	High

Actor: Member

Summary:

The user creates a new account in the system.

Goal:

Allow new users to register and create an account.

Triggers:

The user initiates registration by selecting the register option.

Preconditions:

The user must not already have an account.

Post Conditions:

The user account is created and stored in the system.

Main Success Scenario:

1. User selects the register option.
2. The system prompts for personal details.
3. User submits details.
4. The system validates the information.
5. The system creates the account and confirms registration.

Alternative Scenario

:

If the user already has an account, the system notifies the user and aborts the registration process.

Exceptions:

1. If the information is invalid, the system prompts the user to correct the details.
2. Relationships:
3. Related to user management and onboarding processes.
4. Business Rules:
5. Only new users can register for an account.

Relationships:

Related to user management and onboarding processes.

Business Rules:

Only new users can register for an account.

Use Case 12: Login			
Use-case No.	UC002	Use-case Version	1.0
Use-case Name	Login		
Author	Võ Gia Linh		
Date	24/10/2024	Priority	High

Actor: Member

Summary:

Allows a registered user to access the system.

Goal:

Allow users to login into their accounts.

Triggers:

The user selects the login option.

Preconditions:

The user must have a registered account.

Post Conditions:

The user is logged into the system.

Main Success Scenario:

1. User selects the login option.
2. The system prompts for credentials.
3. User submits credentials.
4. The system validates the credentials.
5. The user is granted access.

Alternative Scenario:

If the credentials are invalid, the system notifies the user and prompts for re-entry.

Exceptions:

If the system is down, the user is notified of the issue.

Relationships:

Related to session management and user authentication.

Business Rules:

Only registered users can log in.

Use Case 13: View Auction Catalog			
Use-case No.	UC003	Use-case Version	1.0
Use-case Name	View auction catalog		
Author	Võ Gia Linh		
Date	24/10/2024	Priority	High

Actor: Member

Summary:

Allows the user to browse the list of available auctions.

Goal:

Provide users access to the auction catalog.

Triggers:

The user selects the view auction catalog option.

Preconditions:

The user must be logged in.

Post Conditions:

The user views the auction catalog.

Main Success Scenario:

1. User selects view auction catalog.
2. The system displays the list of auctions.
3. The user browses the catalog.

Alternative Scenario:

If there are no auctions available, the system notifies the user.

Exceptions:

If the catalog fails to load, the user is notified.

Relationships:

Related to auction listing and browsing features.

Business Rules:

Only logged-in users can view the catalog.

Use Case 14: View Bidding History			
Use-case No.	UC004	Use-case Version	1.0
Use-case Name	View bidding history		
Author	Võ Gia Linh		
Date	24/10/2024	Priority	High

Actor: Member

Summary:

Allows the user to view their past bids.

Goal:

Provide users access to their bidding history.

Triggers:

The user selects the view bidding history option.

Preconditions:

The user must be logged in.

Post Conditions:

The user views their bidding history.

Main Success Scenario:

User selects view bidding history.

The system retrieves and displays past bids.

Alternative Scenario:

If there are no past bids, the system notifies the user.

Exceptions:

If the history fails to load, the user is notified.

Relationships:

Related to user activity tracking and history logs.

Business Rules:

Only logged-in users can view bidding history.

Use Case 15: Log out			
Use-case No.	UC001	Use-case Version	1.0
Use-case Name	Create A Lot		
Author	Nguyễn Phi Long		
Date	24/10/2024	Priority	High

Actor:

Member

System

Allows the user to log out of the system.

Goal:

The Breeder can successfully create a lot with complete information in preparation for the auction.

Triggers

User selects the logout option.

Preconditions:

The user must be logged in.

Post Conditions:

The user is logged out of the system.

Main Success Scenario:

User selects log out.

System prompts confirmation (optional).

User confirms.

System invalidates session and clears data.

System clears cookies or tokens.

UI updates to logged-out state.

User is redirected to login or initial screen.

Alternative Scenario:

Session Timeout: System detects inactivity, warns the user, and auto-logs out if no response.

Log Out on Multiple Devices: User selects log out from all devices; system invalidates sessions on all devices.

Log Out without Confirmation: User selects log out, and the system immediately logs them out without confirmation.

Exceptions

Network Failure: System retries log out; if unsuccessful, user session remains active.

Session Already Expired: System notifies user and redirects to login.

Failed Session Invalidation: System retries; user advised to close app if unsuccessful.

Relationships

Includes: Session Timeout.

Extends: Log Out on Multiple Devices.

Business Rules

Securely clear session data on log out.

Invalidate session tokens on server.

Redirect user to login screen, requiring re-authentication.

2.3.1

Actor:

- ***Staff***

Summary:

Staff can create, update, deactivate, and retrieve account information for members (including breeders and other staff).

Goal:

Enable staff to efficiently manage user accounts to maintain the integrity and functionality of the auction platform.

Triggers:

Staff initiates the use case to manage member accounts for administrative purposes, such as updating account information, deactivating accounts, or retrieving account details.

Preconditions:

- *Staff must be logged in and have the necessary permissions to manage accounts.*
- *The account being managed should exist in the system database.*

Post Conditions:

- *The member account is successfully created, updated, deactivated, or retrieved as requested by the staff.*
- *Changes are logged for auditing purposes.*

Main Success Scenario:

1. *Staff selects the "Manage Member Account" option.*
2. *The system displays the account management options: create, update, deactivate, and retrieve account information.*
3. *Staff chooses an option:*
    - ***Create Account**: Staff enters details for a new account, and the system validates and saves the new account.*
    - ***Update Account**: Staff selects an existing account, modifies details, and saves changes.*
    - ***Deactivate Account**: Staff selects an account to deactivate, and the system marks it as inactive.*
    - ***Get Account Information**: Staff selects an account to view its information.*
4. *The system confirms the action and displays a success message.*

Alternative Scenario:

- ***A1**: If the account does not exist for retrieval or update, the system notifies the staff and suggests creating a new account if needed.*

Exceptions:

- ***E1: Required fields are missing.***
- ***E2: Account creation fails due to server error.***.

Relationships:

- *Relates to "Manage Breeder/Staff Account" as a sub-function.*

Business Rules:

- *Only staff members with the appropriate permissions can create, update, deactivate, or view accounts.*
- *Each account change must be logged with a timestamp and staff ID for auditing purposes.*
- *Deactivated accounts should not be allowed to participate in auctions or access system functionalities.*

Use Case 16: Create An Account

Use-case No.	UC001	Use-case Version	1.0
Use-case Name	Create An Account		
Author	Nguyễn Trình Cát Linh		
Date	24/10/2024	Priority	High

Actor:

Admin

System

Summary:

The Admin creates a new account for breeders, staff, or other members by providing required details such as name, email, and role, isActive….

Goal:

The Admin can successfully create a new user account with all necessary details, allowing the user access based on their role..

Triggers

Admin decides to add a new user account.

Preconditions:

PRE-1: Admin has appropriate permissions to create accounts.

PRE-2: Required fields for account creation are available.

Post Conditions:

POST-1: New account is successfully created and stored in the system.

POST-2: The account status is set to active.

Main Success Scenario:

The Admin selects the "Create Account" function.

The system displays fields for required account information.

The Admin enters information, including name, email, and role.

The Admin confirms the creation of the account.

The system saves the new account and displays a success notification.

Alternative Scenario:

If the Admin does not provide all required information, the system displays an error message and prompts for re-entry

Exceptions:

If the system is unavailable, the Admin will not be able to create an account, and an error message is displayed.

Relationships:

Include: Set User Information, Set Role

Business Rules:

BR-01: Required Information Completeness – All mandatory fields must be completed before the account can be created.

BR-02: Unique Email Validation – The email for each account must be unique within the system.

Use Case 17: Update Account Information

Use-case No.	UC001	Use-case Version	1.0
Use-case Name	Update Account Information		
Author	Nguyễn Trình Cát Linh		
Date	24/10/2024	Priority	Medium

Actor:

Admin

System

Summary:

The Admin updates information for an existing account, such as changing the email, role, or status….

Goal:

The Admin can update user account information, ensuring it remains accurate and current

Triggers

The Admin selects an account to update.

Preconditions:

The Admin is logged into the system.

The account exists in the system.

Post Conditions:

The updated account information is saved in the system

Main Success Scenario:

The Admin selects an account to update.

The system displays current account details.

The Admin updates the information.

The Admin confirms the update.

The system saves the updated information and displays a success notification.

Alternative Scenario:

If required information is missing or invalid, the system displays an error message.

Exceptions:

If the system is unavailable, the update cannot proceed.

Relationships:

Include: Edit Account Information

Business Rules:

BR-03: Role Modification Authorization – Only authorized Admins can modify the roles of other users.

Use Case 18: Deactivate Account

Use-case No.	UC001	Use-case Version	1.0
Use-case Name	Deactivate Account		
Author	Nguyễn Trình Cát Linh		
Date	24/10/2024	Priority	Medium

Actor:

Admin

System

Summary:

The Admin deactivates an account, restricting the user's access to the system

Goal:

The Admin can deactivate user accounts as needed to revoke system access.

Triggers

The Admin initiates deactivation for an account.

Preconditions:

The Admin is logged into the system.

The account exists in the system.

Post Conditions:

Account information is displayed to the Admin.

Main Success Scenario:

The Admin selects "Get Account Information."

The system displays the account details.

Alternative Scenario:

Exceptions:

If the account does not exist, the system displays an error message.

Relationships:

Include: Confirm Deactivation

Business Rules:

BR-04: Deactivation Authorization – Only Admins with deactivation privileges can perform this action.

Use Case 19: Get Account Information			
Use-case No.	UC001	Use-case Version	1.0
Use-case Name	Get Account Information		
Author	Nguyễn Trình Cát Linh		
Date	24/10/2024	Priority	Low

Actor:

Admin

System

Summary:

The Admin views details of a specific user account

Goal:

The Admin can retrieve and view account details for reference or management purposes.

Triggers

The Admin requests account details

Preconditions:

The Admin is logged into the system.

The account exists in the system.

Post Conditions:

The account status is updated to "deactivated" in the system.

Main Success Scenario:

The Admin selects the "Deactivate Account" function.

The system prompts for confirmation.

The Admin confirms the deactivation.

The system deactivates the account and displays a success message.

Alternative Scenario:

Exceptions:

If the system is unavailable, the deactivation cannot proceed.

Relationships:

Include: Display Account Information

Business Rules:

BR-05: Account Retrieval Permission – Only authorized Admins can view sensitive account information.

Use Case 20: Get Revenue			
Use-case No.	UC001	Use-case Version	1.0
Use-case Name	Get Revenue		
Author	Nguyễn Trình Cát Linh		
Date	24/10/2024	Priority	High

Actor:

Admin

System

Summary:

The Admin retrieves and views the revenue data of the auction system, including total sales, transaction fees, and any additional charges or costs. This information is crucial for monitoring the financial performance of the auction system and for making business decisions.

Goal:

To allow the Admin to access accurate and up-to-date revenue data for financial assessment and reporting purposes.

Triggers

The Admin initiates the "Get Revenue" function in the system.

Preconditions:

The Admin is logged into the system.

The Admin has financial data access permissions.

Post Conditions:

Revenue data, including relevant metrics like total sales, auction fees, and transaction costs, is successfully displayed to the Admin.

The system logs this access event for auditing purposes.

Main Success Scenario:

The Admin navigates to the financial or revenue section of the system.

The Admin selects the "Get Revenue" option to request revenue information.

The system validates the Admin’s permissions to access financial data.

The system retrieves the latest revenue data, calculating totals if necessary.

The system displays a detailed revenue report to the Admin, including:

- Total revenue generated from auctions
- Transaction fees and processing charges
- Breakdown of revenue per auction, if applicable
- Any other relevant financial metrics (e.g., refunds, deductions).

The Admin reviews the displayed revenue report for analysis.

Alternative Scenario:

Insufficient Permissions: If the Admin does not have sufficient permissions:

- The system displays an "Access Denied" message.
- The Admin is unable to proceed further without appropriate permissions.

Exceptions:

E1. Data Unavailability: If the system is unable to retrieve revenue data due to network or database issues:

- The system displays an error message, informing the Admin that revenue data is temporarily unavailable.
- The Admin is advised to try again later.

E2. System Error: If there is a processing error during data retrieval:

- The system logs the error.
- An error notification is displayed to the Admin.

Relationships:

Include: Display Revenue Data, Access Control Check

Extend: Generate Custom Revenue Report (Admin may generate reports based on specific time frames, auction types, etc.)

Business Rules:

BR-06: **Financial Data Access** – Only Admins with explicit permissions can access revenue information to maintain data privacy.

BR-07: **Revenue Accuracy** – All revenue figures displayed must reflect the latest transaction records to ensure accuracy.

BR-08: **Audit Logging** – Every access to revenue data is logged in the system for security and auditing purposes

Use Case 21: Manipulate Revenue			
Use-case No.	UC001	Use-case Version	1.0
Use-case Name	Manipulate Revenue		
Author	Nguyễn Trình Cát Linh		
Date	24/10/2024	Priority	Low

Actor:

Admin

System

Summary:

The Admin modifies specific revenue records, such as adjusting transaction amounts or correcting financial data. This feature is necessary for cases where adjustments are needed to maintain accurate and up-to-date financial reporting within the auction management system.

Goal:

To enable the Admin to make necessary corrections to revenue data for accurate financial record-keeping and reporting.

Triggers

The Admin initiates the "Manipulate Revenue" function by selecting a revenue record that requires adjustment..

Preconditions:

The Admin is logged into the system.

The Admin has the appropriate permissions to modify revenue data.

The revenue data exists and is available for modification

Post Conditions:

The selected revenue record is successfully modified and saved in the system.

The system logs this action for audit purposes.

Main Success Scenario:

The Admin navigates to the revenue management section and selects a revenue record to manipulate.

The system retrieves the current data for the selected revenue record and displays it.

The Admin edits the required fields (e.g., transaction amount, tax adjustments, corrections).

The Admin confirms the changes.

The system validates the input data to ensure consistency and accuracy.

Upon successful validation, the system saves the modified revenue record.

The system displays a confirmation message to the Admin, indicating that the revenue record has been updated.

The system logs the update event, recording details for future auditing

Alternative Scenario:

Invalid Data Entry: If the Admin enters incorrect or invalid data (e.g., negative transaction amount):

- The system displays an error message specifying the validation issue.
- The Admin is prompted to correct the data before proceeding.

Exceptions:

E1. Insufficient Permissions: If the Admin lacks permission to modify revenue data:

- The system displays an "Access Denied" message.
- The Admin cannot proceed with the manipulation.

E2. System Unavailability: If the system or database is down:

- The system displays an error message indicating that the revenue data is currently unavailable.
- The Admin is advised to try again later.

Relationships:

Include: Validate Revenue Data, Log Revenue Manipulation Action

Extend: Generate Revenue Adjustment Report (for tracking changes made to revenue data)

Business Rules:

BR-07: **Revenue Adjustment Authorization** – Only Admins with specific permissions can access and modify revenue data to ensure security and data integrity.

BR-08: **Data Validation** – Any changes to revenue entries must pass validation checks, ensuring no negative or logically incorrect values are saved.

BR-09: **Audit Logging** – All revenue adjustments are logged with details such as Admin ID, timestamp, and specific changes made, allowing for comprehensive audit trails.

Use Case 22: Place Bid

Use-case No.	UC001	Use-case Version	1.0
Use-case Name	Create A Lot		
Author	Nguyễn Phi Long		
Date	24/10/2024	Priority	High

Actor:

Member

System

Allows the user to place a bid on an auction item.

Goal:

User to place a bid on an auction item.

Triggers

User selects an auction item.

Preconditions:

The user must be logged in and have sufficient funds.

Post Conditions:

The bid is placed and recorded in the system.

Main Success Scenario:

1. User selects an auction item.
2. System displays item details.
3. User enters bid amount.
4. System validates bid. 5. System records the bid.

Alternative Scenario:

Insufficient Balance: System rejects bid, prompting user to top up or place a lower bid.

Simultaneous Higher Bid: System rejects lower bid and notifies user of higher bid.

Auction Lot Expired: System informs user that bidding has closed.

Bid in Last Seconds: System extends countdown, notifying users of additional time.

Exceptions

Network Failure: User is notified and can retry.

Processing Delay: System queues bid, informs user of delay.

Invalid Bid Format: System prompts for bid correction.

Unauthorized User: System denies access and notifies user.

Relationships

Includes: Balance Check.

Extends: Bid Extension for last-second bids.

Business Rules

Bids must meet minimum increment.

Users need sufficient wallet balance.

Only authorized users can bid.

Last-second bids reset countdown.

Use Case 23: Manage Account

Use-case No.	UC001	Use-case Version	1.0
Use-case Name	Create A Lot		
Author	Nguyễn Phi Long		
Date	24/10/2024	Priority	High

Use Case 24: Register

Use-case No.	UC001	Use-case Version	1.0
Use-case Name	Register		
Author	Nguyễn Trình Cát Linh		
Date	24/10/2024	Priority	High

Actor:

Guest

System

Summary:

The Guest registers for an account on the auction platform to gain access to additional features such as bidding and viewing detailed auction information.

Goal:

To allow the Guest to create an account, enabling them to participate in auctions and access member-specific features.

Triggers

The Guest selects the "Register" option on the website.

Preconditions:

The Guest has access to the registration form

Post Conditions:

A new user account is created and saved in the system.

Main Success Scenario:

The Guest selects the "Register" option.

The system displays the registration form with required fields.

The Guest fills in the necessary details (e.g., username, password, email).

The Guest submits the form.

The system validates the information, creates the account, and confirms the registration to the Guest.

Alternative Scenario:

3a. If required fields are incomplete or invalid, the system prompts the Guest to correct the information.

Exceptions:

If the system is down, registration cannot proceed, and an error message is displayed

Relationships:

Business Rules:

BR-01: Unique Email – Each registered account must have a unique email address

Use Case 25: Search Koi			
Use-case No.	UC001	Use-case Version	1.0
Use-case Name	Search Koi		
Author	Nguyễn Trình Cát Linh		
Date	24/10/2024	Priority	Low

Actor:

Guest

System

Summary:

The Guest searches for available Koi fish using filters such as size, color, breed, and price range to view listings that match their preferences.

Goal:

To enable the Guest to find specific Koi fish that they might be interested in for bidding or purchase..

Triggers

The Guest initiates a search query using the search bar or filter options.

Preconditions:

Koi listings are available in the system.

Post Conditions:

The system displays a list of Koi fish that match the search criteria

Main Success Scenario:

The Guest enters search criteria (e.g., size, color, breed, price range) in the search interface.

The system retrieves Koi listings matching the search criteria.

The system displays matching results to the Guest.

Alternative Scenario:

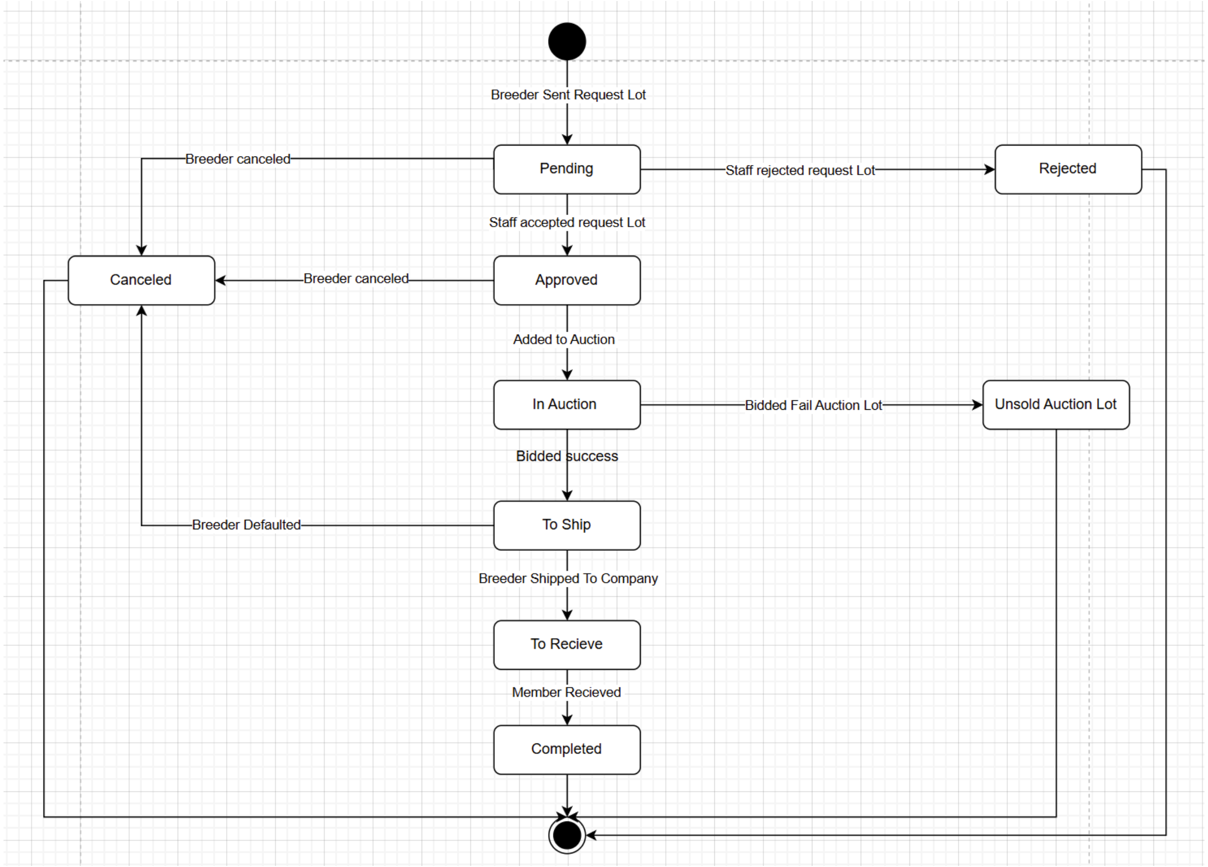
No Results Found: If no Koi match the criteria, the system displays a "No results found" message.

Exceptions: Search System Error: If the search operation fails due to technical issues, the system displays an error message.

Business Rules:

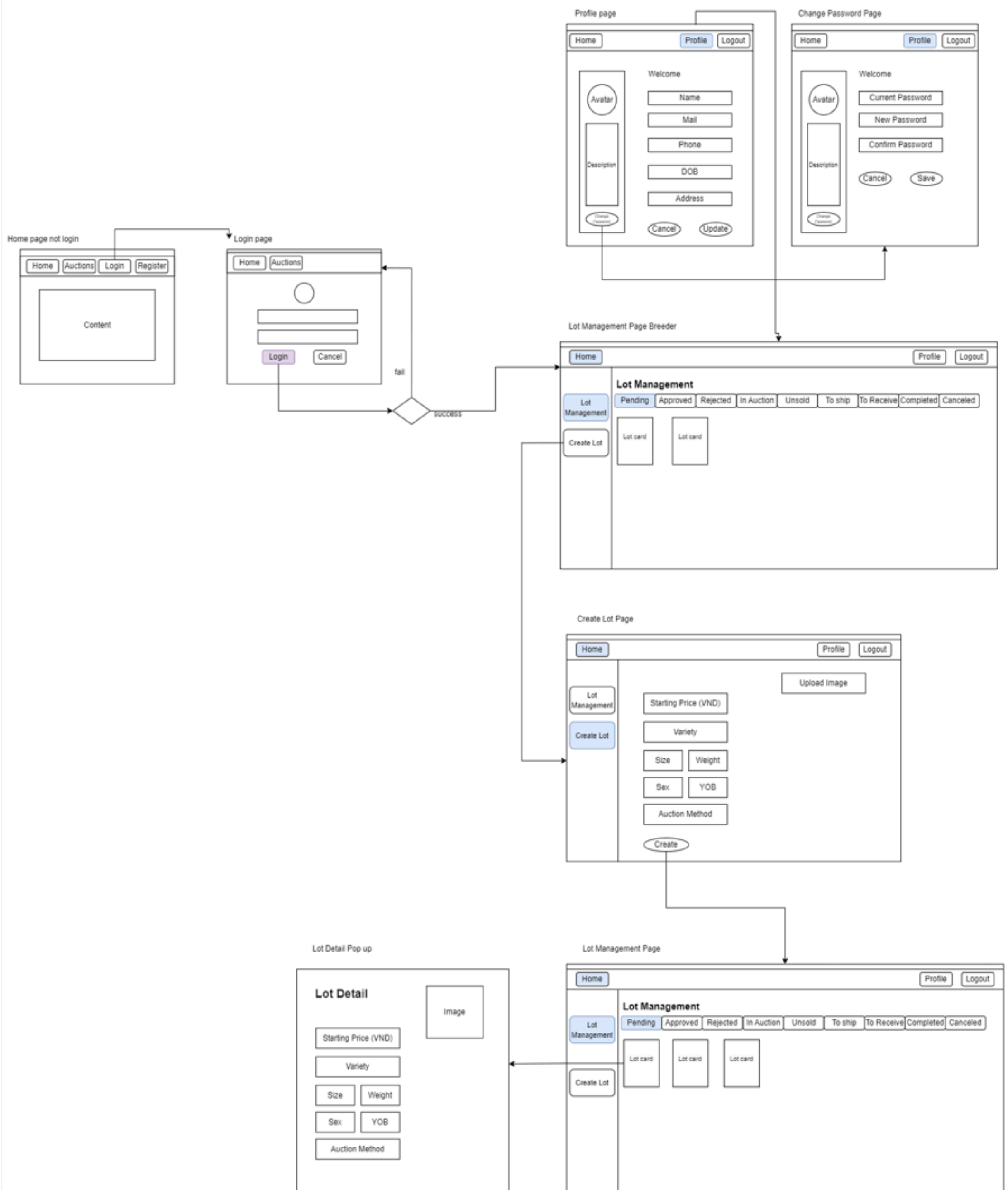
BR-06: Search Accuracy – Search results should accurately reflect the filters and criteria selected by the Guest.

3.3 State Diagrams

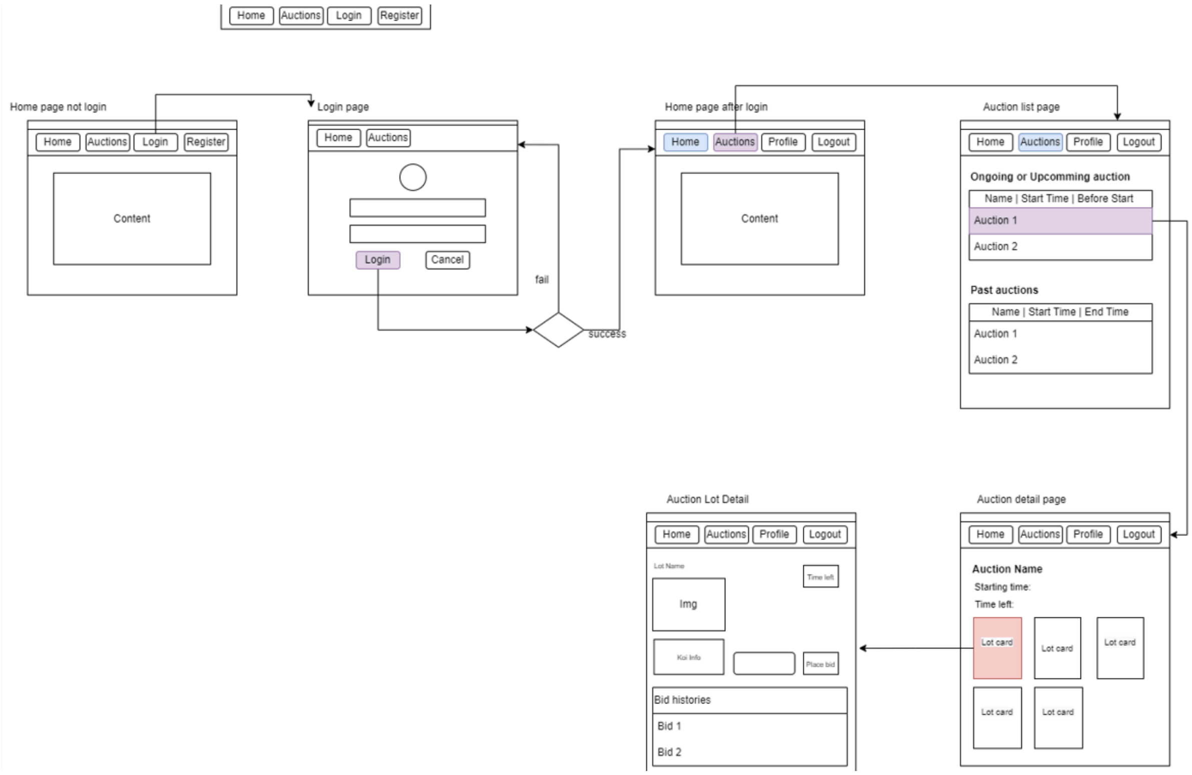


3.4 Wire Flows

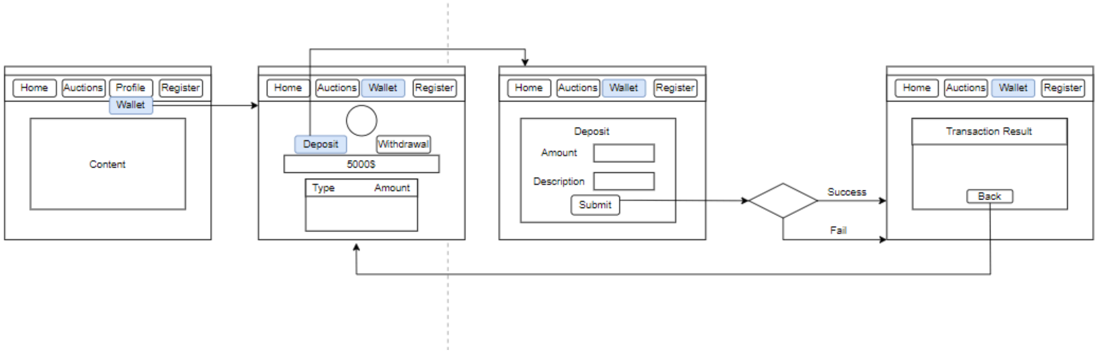
Breeder


Profile page
Home
Profile
Logout
Avatar
Description
Change Password
Welcome
Name
Mail
Phone
DOB
Address
Cancel
Update
Change Password Page
Home
Profile
Logout
Avatar
Description
Change Password
Welcome
Current Password
New Password
Confirm Password
Cancel
Save
Home page not login
Home
Auctions
Login
Register
Content
Login page
Home
Auctions
Login
Cancel
fail
success
Lot Management Page Breeder
Home
Profile
Logout
Lot Management
Create Lot
Lot Management
Pending
Approved
Rejected
In Auction
Unsold
To ship
To Receive
Completed
Canceled
Lot card
Lot card
Create Lot Page
Home
Profile
Logout
Lot Management
Create Lot
Upload Image
Starting Price (VND)
Variety
Size
Weight
Sex
YOB
Auction Method
Create
Lot Detail Pop up
Lot Detail
Image
Starting Price (VND)
Variety
Size
Weight
Sex
YOB
Auction Method
Lot Management Page
Home
Profile
Logout
Lot Management
Create Lot
Lot Management
Pending
Approved
Rejected
In Auction
Unsold
To ship
To Receive
Completed
Canceled
Lot card
Lot card
Lot card

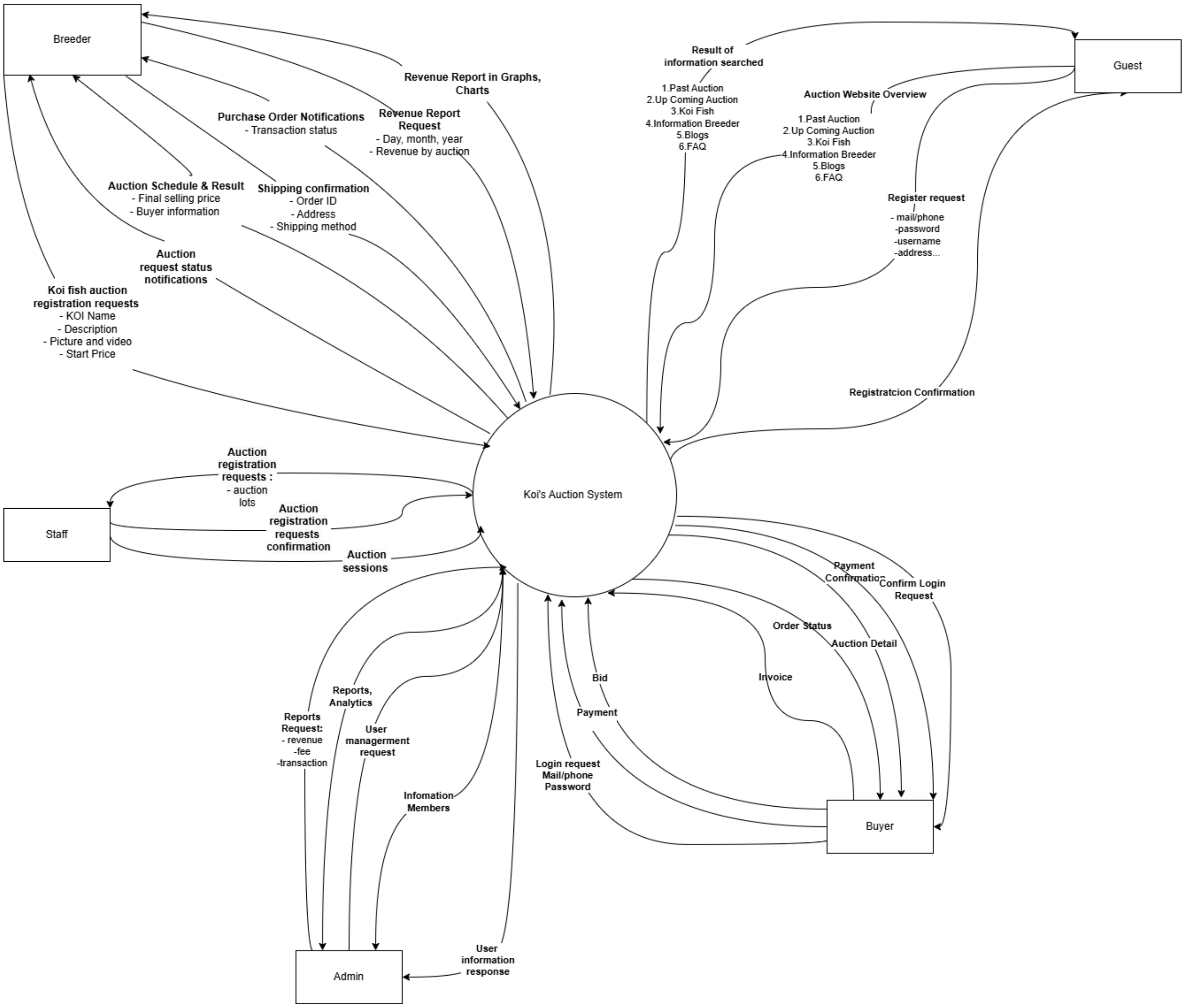
Member



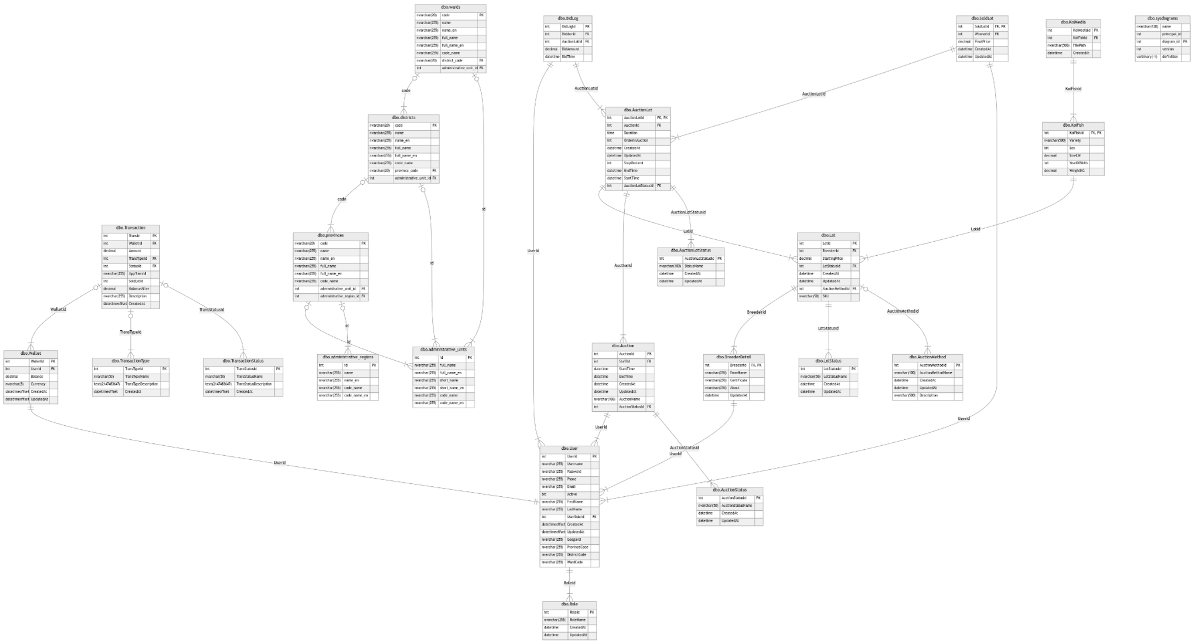
4.3.3 Payment (member)



3.5 Data context



4 Entity Relationship diagram



dbo.wards

nvarchar(20)	code	PK
nvarchar(255)	name	
nvarchar(255)	name_en	
nvarchar(255)	full_name	
nvarchar(255)	full_name_en	
nvarchar(255)	code_name	
nvarchar(20)	district_code	FK
int	administrative_unit_id	FK

dbo.BidLog

int	BidLogId	PK
int	BidderId	FK
int	AuctionLotId	FK
decimal	BidAmount	
datetime	BidTime	

dbo.SoldLot

int	SoldLotId	FK, PK
int	WinnerId	FK
decimal	FinalPrice	
datetime	CreatedAt	
datetime	UpdatedAt	

dbo.KoiMedia

int	KoiMediaId	PK
int	KoiFishId	FK
nvarchar(500)	FilePath	
datetime	CreatedAt	

dbo.sysdiagrams

nvarchar(128)	name	
int	principal_id	
int	diagram_id	PK
int	version	
varbinary(-1)	definition	

dbo.districts

nvarchar(20)	code	PK
nvarchar(255)	name	
nvarchar(255)	name_en	
nvarchar(255)	full_name	
nvarchar(255)	full_name_en	
nvarchar(255)	code_name	
nvarchar(20)	province_code	FK
int	administrative_unit_id	FK

dbo.AuctionLot

int	AuctionLotId	FK, PK
int	AuctionId	FK
time	Duration	
int	OrderInAuction	
datetime	CreatedAt	
datetime	UpdatedAt	
int	StepPercent	
datetime	EndTime	
datetime	StartTime	
int	AuctionLotStatusId	FK

dbo.KoiFish

int	KoiFishId	FK, PK
nvarchar(500)	Variety	
bit	Sex	
decimal	SizeCM	
int	YearOfBirth	
decimal	WeightKG	

dbo.Transaction

int	TransId	PK
int	WalletId	FK
decimal	Amount	
int	TransTypeId	FK
int	StatusId	FK
nvarchar(255)	AppTransId	
int	SoldLotId	
decimal	BalanceAfter	
nvarchar(255)	Description	
datetimeoffset	CreatedAt	

dbo.provinces

nvarchar(20)	code	PK
nvarchar(255)	name	
nvarchar(255)	name_en	
nvarchar(255)	full_name	
nvarchar(255)	full_name_en	
nvarchar(255)	code_name	
int	administrative_unit_id	FK
int	administrative_region_id	FK

dbo.AuctionLotStatus

int	AuctionLotStatusId	PK
nvarchar(100)	StatusName	
datetime	CreatedAt	
datetime	UpdatedAt	

dbo.Lot

int	LotId	PK
int	BreederId	FK
decimal	StartingPrice	
int	LotStatusId	FK
datetime	CreatedAt	
datetime	UpdatedAt	
int	AuctionMethodId	FK
nvarchar(50)	SKU	

dbo.Wallet

int	WalletId	PK
int	UserId	FK
decimal	Balance	
nvarchar(5)	Currency	
datetimeoffset	CreatedAt	
datetimeoffset	UpdatedAt	

dbo.TransactionType

int	TransTypeId	PK
nvarchar(50)	TransTypeName	
text(2147483647)	TransTypeDescription	
datetimeoffset	CreatedAt	

dbo.TransactionStatus

int	TransStatusId	PK
nvarchar(50)	TransStatusName	
text(2147483647)	TransStatusDescription	
datetimeoffset	CreatedAt	

dbo.administrative_regions

int	id	PK
nvarchar(255)	name	
nvarchar(255)	name_en	
nvarchar(255)	code_name	
nvarchar(255)	code_name_en	

dbo.administrative_units

int	id	PK
nvarchar(255)	full_name	
nvarchar(255)	full_name_en	
nvarchar(255)	short_name	
nvarchar(255)	short_name_en	
nvarchar(255)	code_name	
nvarchar(255)	code_name_en	

dbo.Auction

int	AuctionId	PK
int	StaffId	FK
datetime	StartTime	
datetime	EndTime	
datetime	CreatedAt	
datetime	UpdatedAt	
nvarchar(100)	AuctionName	
int	AuctionStatusId	FK

dbo.BreederDetail

int	BreederId	FK, PK
nvarchar(255)	FarmName	
nvarchar(255)	Certificate	
nvarchar(255)	About	
datetime	UpdatedAt	

dbo.LotStatus

int	LotStatusId	PK
nvarchar(50)	LotStatusName	
datetime	CreatedAt	
datetime	UpdatedAt	

dbo.AuctionMethod

int	AuctionMethodId	PK
nvarchar(100)	AuctionMethodName	
datetime	CreatedAt	
datetime	UpdatedAt	
nvarchar(500)	Description	

dbo.User

int	UserId	PK
nvarchar(255)	Username	
nvarchar(255)	Password	
nvarchar(255)	Phone	
nvarchar(255)	Email	
bit	Active	
nvarchar(255)	FirstName	
nvarchar(255)	LastName	
int	UserRoleId	FK
datetimeoffset	CreatedAt	
datetimeoffset	UpdatedAt	
nvarchar(255)	GoogleId	
nvarchar(255)	ProvinceCode	
nvarchar(255)	DistrictCode	
nvarchar(255)	WardCode	

dbo.AuctionStatus

int	AuctionStatusId	PK
nvarchar(50)	AuctionStatusName	
datetime	CreatedAt	
datetime	UpdatedAt	

dbo.Role

int	RoleId	PK
nvarchar(255)	RoleName	
datetime	CreatedAt	
datetime	UpdatedAt	

4.1 NON-FUNCTIONAL REQUIREMENTS

4.2 Usability

User

A new user should be able to navigate the platform, browse auctions, place bids, and complete transactions within **15 minutes** of first use, without requiring additional training.

Breeder

Breeders should be able to manage their lot list, request new lots, and monitor lot status within **30 minutes** of training or initial use.

Staff

Staff members responsible for platform operations and user assistance should be able to perform their tasks, including monitoring auctions, assisting users, and resolving issues, after **30 minutes** of hands-on training.

Administrator

Administrators, responsible for overseeing platform operations, user management, and system maintenance, should be able to manage all administrative features and troubleshoot system issues within **60 minutes** of training or initial use, including tasks such as transaction monitoring and auction oversight.

4.3 Reliability

Availability

The platform must be operational **24/7,** with scheduled maintenance during off-peak hours. In the event of degradation, basic auction functionality should remain available while advanced features are temporarily disabled.

Mean Time Between Failures (MTBF)

The system should achieve a minimum of **1,000 hours** of continuous operation without failure.

Mean Time to Repair (MTTR)

In the event of failure, the system must be restored within **1 hour.**

Accuracy

The platform must ensure **99.99%** accuracy for auction results, bids, and payments, with real-time updates and countdown timers refreshing with an accuracy deviation of no more than **100 milliseconds.**

4.4 Performance

Any response must be within 2 seconds or less

After press any button, there're always respond within 1 seconds (loading process, save success, etc)

4.5 Supportability

Naming Conventions:

- Consistent **naming conventions** must be used across the stack:
  - **CamelCase** for JavaScript/React variables and function names.
  - **PascalCase** for React components and class names.
  - **PascalCase** for **C# class** and **method names** in the ASP.NET backend.
- Descriptive and meaningful names should be used to ensure that code is easy to understand and maintain.

4.6 Licensing Requirements

Third-Party Software Licensing

Any **third-party libraries**, frameworks, or tools integrated into the platform must be used under valid and appropriate licenses (e.g., **MIT**, **GPL**, **Apache 2.0**). The development team must ensure compliance with the terms of these licenses.

All third-party components must be properly documented, including their licensing terms and any obligations for attribution, modification, or distribution.

Commercial Licensing

If any commercial software or tools are used in the development or operation of the platform (e.g., cloud services, payment gateways), the necessary **commercial licenses** must be obtained and maintained. This includes ensuring all subscription or usage-based services are in good standing.

User Licensing Enforcement:

The platform may include **user licensing enforcement** mechanisms to control access to certain features based on the user’s role or subscription level (e.g., breeders).