

**PrestigeKoi Auctions**

**Software Requirement Document**

– HCM, Jan 2024 –

**Record of changeS**

| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
| 26/Oct | A | SonLQT | Tạo SRD Document và thêm mọi thông tin cần thiết |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

**Table of Contents**

[I. Overview 4](#_heading=h.30j0zll)

[1. Introduction 4](#_heading=h.1fob9te)

[2. Existing Systems 4](#_heading=h.3znysh7)

[2.1 System name1 4](#_heading=h.2et92p0)

[2.2 System name2 4](#_heading=h.tyjcwt)

[3. Business Process 5](#_heading=h.3dy6vkm)

[4. System Functions 6](#_heading=h.1t3h5sf)

[5. Entity Relationship Diagram 7](#_heading=h.4d34og8)

[6. State transition diagram 7](#_heading=h.2s8eyo1)

[II. Functional Requirements 9](#_heading=h.17dp8vu)

[1. <<Feature Name 1>> 9](#_heading=h.3rdcrjn)

[a. <<Function Name 1>> 9](#_heading=h.26in1rg)

[b. <<Function Name 2>> 9](#_heading=h.lnxbz9)

[2. <<Feature Name 2>> 9](#_heading=h.35nkun2)

[III. Database Design 10](#_heading=h.44sinio)

[1. Database Schema 10](#_heading=h.2jxsxqh)

[2. Table Detail 10](#_heading=h.3j2qqm3)

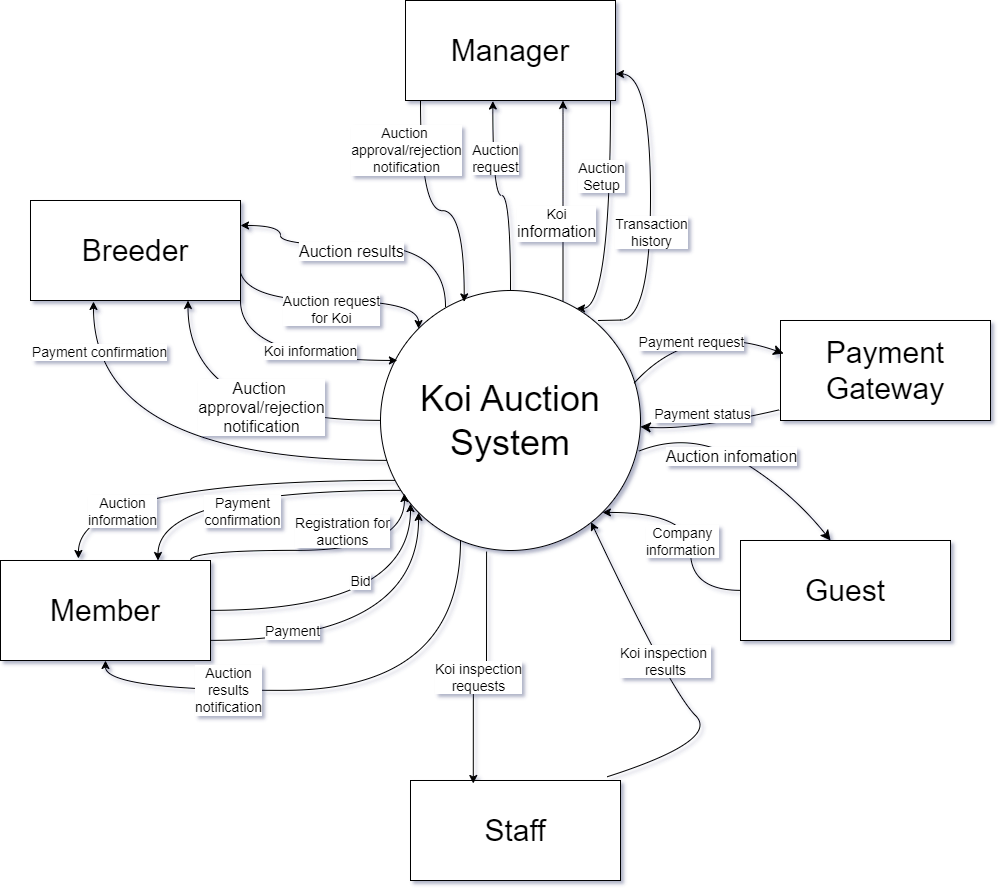
# I. Overview

## 1. Introduction

*[Content part 1: presents a high-level overview of the product and the environment in which it will be used, the users, and known constraints, assumptions, and dependencies]*

*[Content part 2: describes the product's context in the form of a context diagram in which you present the boundary and connections between the system you’re developing and everything else in the universe. This identifies external entities (or terminators – software, hardware, human components, and other systems) outside the system that interface to it in some way, as well as data, control, and material flows between the terminators and the system]*

The **Koi Auction System** is a comprehensive platform designed to facilitate the online auction and transportation of koi fish. This system replaces the manual processes used by breeders and buyers and provides an automated, user-friendly environment for managing auctions, bids, transportation logistics, and financial transactions. The system caters to a variety of users, each with specific roles and responsibilities, ensuring smooth operations from listing koi fish for auction to the final delivery and payment settlement.



Koi Auction System Context Diagram

### 2. Existing Systems

This section outlines existing systems that are relevant to solving the problem by studying their features, actors, pros, and cons. By examining these systems, we can identify valuable components that can be incorporated into the design of a new solution. Two systems that serve as reference points for our design are **Auction Koi** and **eBay**.

### 2.1 Auction Koi

**Website**: aucionkoi.com**System Actors**:

* **Sellers**: Individuals or businesses who list items for auction.
* **Buyers**: Participants who place bids to purchase items.
* **Admin**: System administrators who manage and oversee auctions and resolve disputes.

**Features**:

* **Auction-based Sales**: Auction Koi specializes in online auctions, allowing sellers to list items and buyers to bid within a set timeframe.
* **User Registration**: Both buyers and sellers must register an account to participate in auctions.
* **Real-Time Bidding**: Bidders can participate in real-time, with updates to bid status shown live.
* **Watchlist Feature**: Buyers can add items to their watchlist to monitor auctions closely.
* **Automatic Bidding**: Buyers can set maximum bids, and the system will automatically place bids up to that limit.
* **Messaging**: Sellers and buyers can communicate directly through the platform.

**Pros**:

* **Specialized Niche**: Auction Koi focuses on a specific niche (e.g., koi fish auctions), creating a highly targeted user base.
* **Ease of Use**: Simple and user-friendly interface tailored to quick and easy bidding.
* **Real-time Bidding Alerts**: Bidders receive real-time notifications when they are outbid or when an auction is about to end.

**Cons**:

* **Limited to Auctions**: The platform is auction-specific, meaning there are no options for direct purchases or 'Buy Now' features.
* **Small Market**: Due to its niche, Auction Koi attracts a smaller audience compared to broader platforms like eBay.
* **Limited Categories**: The focus on specific types of items limits the range of products available for sale.

### 2.2 eBay

**Website**: ebay.com**System Actors**:

* **Sellers**: Individuals or businesses who list items for sale or auction.
* **Buyers**: Customers who purchase items or place bids in auctions.
* **Admins**: eBay support and administrators who oversee platform operations and customer service.
* **Third-Party Payment Processors**: Systems like PayPal (historically) or managed payments by eBay for transaction handling.

**Features**:

* **Auction & Direct Sales**: eBay offers both auction-based sales and fixed-price (Buy It Now) options.
* **Wide Product Range**: eBay supports numerous categories, from electronics to collectibles, fashion, and more.
* **Global Reach**: Users can buy and sell items across the world, with eBay facilitating shipping options.
* **Seller Ratings & Reviews**: Buyers can leave feedback and ratings for sellers, contributing to their overall trust score.
* **Advanced Search**: eBay’s search engine allows users to filter results by price, location, auction type, and more.
* **Payment Protection**: eBay offers buyer protection and seller guarantees to prevent fraud and mishandling.
* **Mobile App**: eBay has a fully-featured mobile app for both iOS and Android devices.

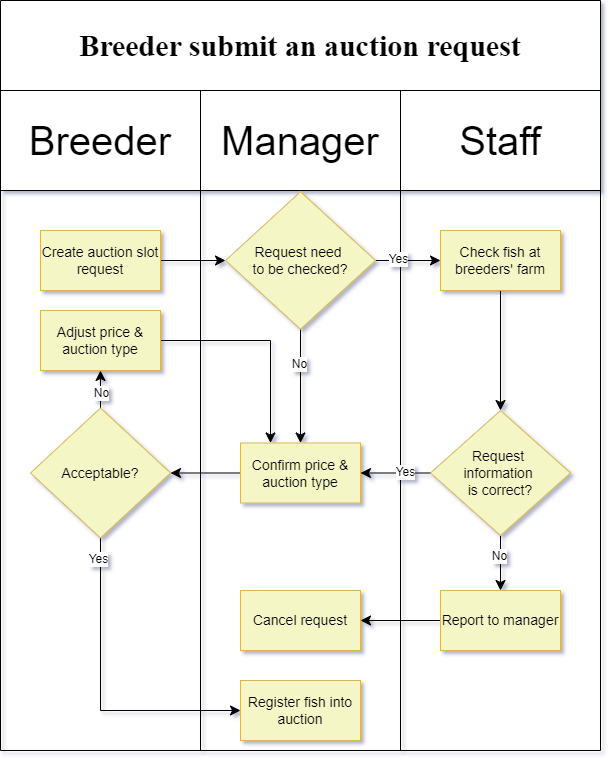
**Pros**:

* **Vast User Base**: eBay is a global marketplace, attracting millions of buyers and sellers worldwide.
* **Multiple Sale Formats**: Users can choose between auctions, fixed-price sales, or a combination of both, providing flexibility for different types of transactions.
* **Buyer/Seller Protections**: eBay's comprehensive system of buyer and seller protections increases trust and reduces fraud.
* **Third-Party Integrations**: eBay integrates with various shipping, payment, and logistics systems, streamlining the buying and selling process.

**Cons**:

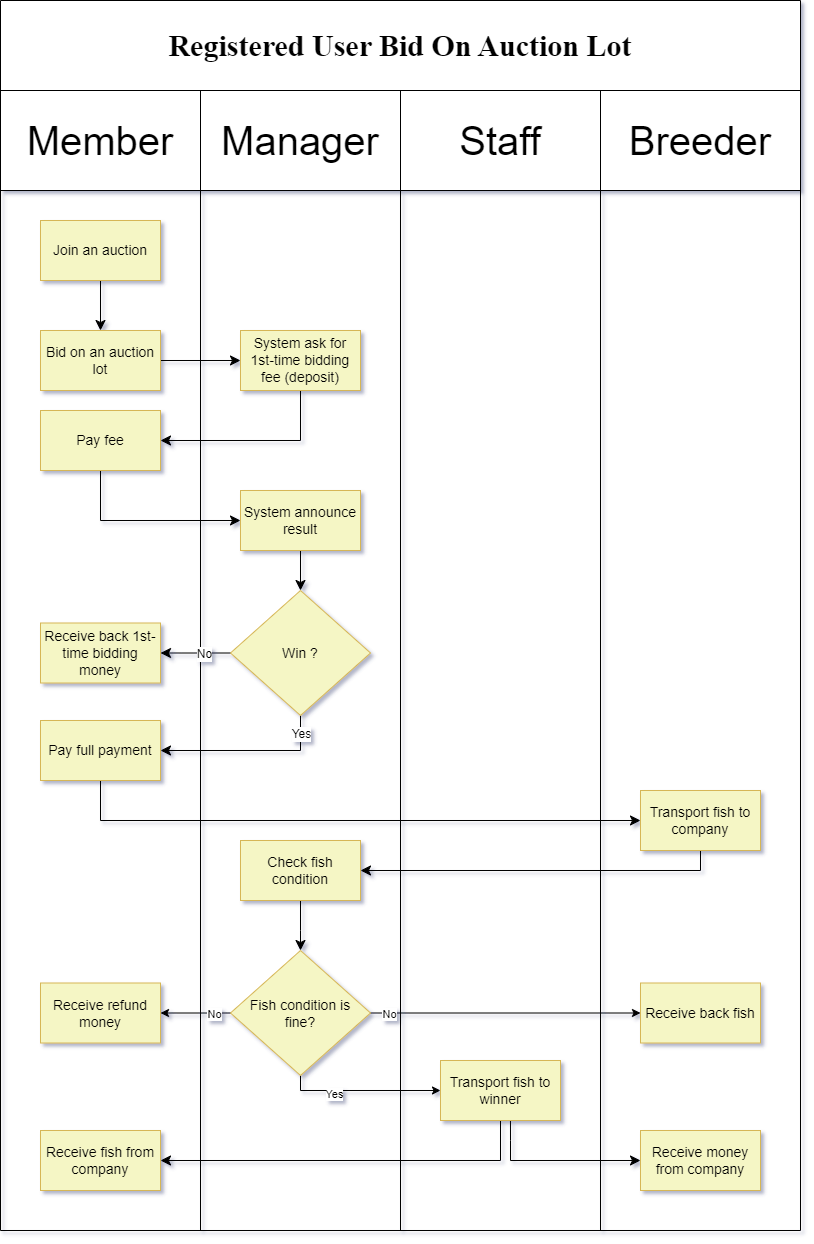
* **Fees**: Sellers are charged multiple fees, including listing fees, final value fees, and potentially PayPal transaction fees (for historical sales), which can reduce profit margins.
* **Competition**: Due to the large number of sellers, it can be difficult to stand out unless you have highly desirable or unique products.
* **Complexity**: The system is large, and navigating its numerous features, settings, and fees can be overwhelming for new users.
* **Fraud Risk**: While eBay offers protections, the large volume of transactions means there are still risks of scams, especially for high-value items.

## 3. Business Process



Picture of “Breeder submit an auction swimlane” diagram. The diagram describes the process of how a request is made and confirmed for the fish to be registered into the database.

| **#** | **Process Step** | **Description** |
| --- | --- | --- |
| 1 | Create auction slot request | Breeder will create an auction request, in which they will input Koi fishes info and offer price along with auction type. |
| 2 | Check fish at breeders' farm | Staff will check fish conditions and info at breeders’ farms to ensure they match request information. |
| 3 | Confirm price & auction type | Manager will confirm price and auction type after the request is checked as pass by staff |
| 4 | Report to manager | Staff will report to the manager about requests if request information is wrong. |
| 5 | Cancel request | Manager will cancel requests made by breeder after confirming the report from staff. |
| 6 | Adjust price & auction type | Breeder will adjust price and auction type if they want to change information. |
| 7 | Register fish into auction | Manager will register fish into the database when breeder agreed with all information. |



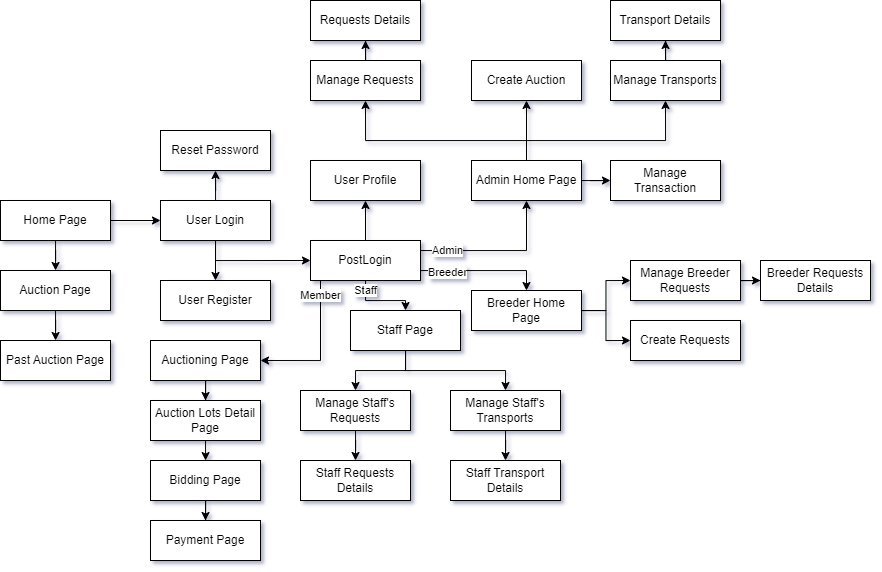
Picture of “Registered User Bid On Auction Lot” diagram. The diagram describes the process of how a user can bid into an auction lot, how the fish can be deliver to winner and how breeder can receive money.

| **#** | **Process Step** | **Description** |
| --- | --- | --- |
| 1 | Join an auction | Users will join an auction that they want. |
| 2 | Bid on an auction lot | User will bid on an auction lot they want |
| 3 | System ask for 1st-time bidding fee (deposit) | System will ask user to pay for deposit at that specific lot |
| 4 | Pay fee | User will pay the deposit fee |
| 5 | System announce result | Users will receive the result of their bid. |
| 6 | Pay full payment | Users will pay the full payment of the bid they have won, including tax, platform fee,... |
| 7 | Transport fish to company | Breeder will transport fish to company |
| 8 | Check fish condition | Staff will check fish conditions so as if it can be transport to users |
| 9 | Receive refund money | Users will receive refund money if fish is not available to be transport to them |
| 10 | Receive back fish | Breeder will receive back fish if fish conditions is not good |
| 11 | Transport fish to winner | Staff will transport fish to winner if fish conditions is good |
| 12 | Receive money from company | Breeder will receive money from the company when transport is completed. |
| 13 | Receive fish from company | Users will receive fish from staff when they transport to them |

## 

## 4. System Functions

#### a. Screen Flow



#### b. Screen Details

| **#** | **Feature** | **Screen** | **Description** |
| --- | --- | --- | --- |
| 1 | View information about web | Home Page | This page shows information about the web, and a guide to use all of the features on the website. |
| 2 | View auction | Auction Page | This page shows information about the auction |
| 3 | View past auction | Past Auction Page | This page shows information about the past auction |
| 4 | Login | User Login | This page shows user login information and allow user to login |
| 5 | Reset Password | Reset Password | This page allows users to reset password |
| 6 | Register | User Register | This page allows users to register an account |
| 7 | View Profile | User Profile | This page allows users to view profile information. |
| 8 | View staff features | Staff Page | This page allows staffs to view staff feature |
| 9 | Manage staff’s request | Manage Staff’s Request | This page shows staff list of request to manage |
| 10 | Manage Staff’s Transport | Manage Staff’s Transport | This page shows staff list of transport to manage |
| 11 | View staff requests details | Staff Requests Details | This page shows staff the request details of a specific request, allowing them to interact with it. |
| 12 | View staff transport details | Staff Transport Details | This page shows staff the transport details of a specific transport, allowing them to interact with it. |
| 13 | View breeder features | Breeder Home Page | This page allows breeders to view breeder feature |
| 14 | Manage Breeder Requests | Manage Breeder Requests | This page shows breeders the request details of a specific request, allowing them to interact with it. |
| 15 | Create Auction Requests | Create Requests | This page allows breeders to create a request to submit to the manager. |
| 16 | View and update breeder request details | Breeder Requests Details | This page shows breeders the request details of a specific request, allowing them to interact with it. |
| 17 | View admin features | Admin Home Page | This page allows admin to view admin feature |
| 18 | Create Auction | Create Auction | This page allows admin to create auction |
| 19 | Manage Transports | Manage Transports | This page allows admin to manage all transports |
| 20 | Manage Requests | Manage Requests | This page allows admin to manage all requests |
| 21 | View requests details | Requests Details | This page shows admin the request details of a specific request, allowing them to interact with it. |
| 22 | View transport details | Transport Details | This page shows admin the transport details of a specific transport, allowing them to interact with it. |
| 23 | View happening auction | Auctioning Page | This page shows information about the auction that is happening |
| 24 | View auction lots detail | Auction Lots Detail Page | This page shows information about the lots of auction that is happening |
| 25 | Bidding | Bidding Page | This page allows registered users to bid on the lots |
| 26 | Payment | Payment Page | This page allow users to pay for deposit and full payment |
| 27 | Manage Transaction | Manage Transaction | This page allow admins to manage and view transactions |

#### c. User Authorization

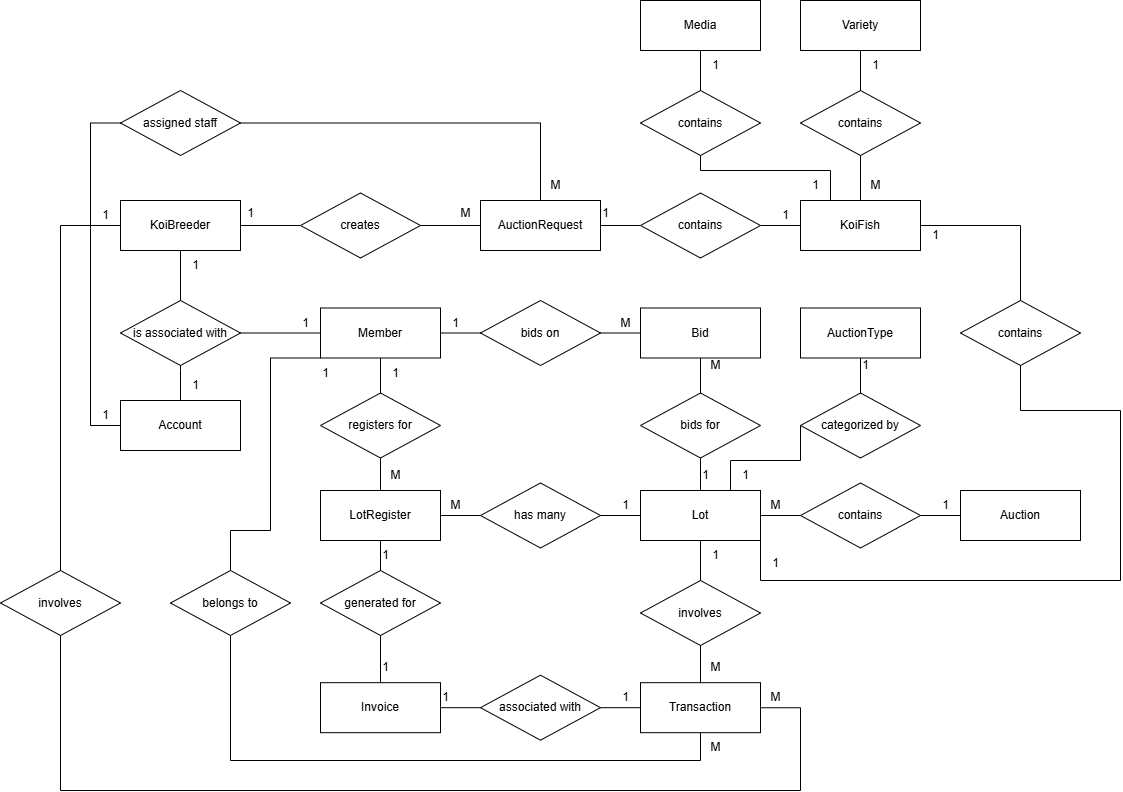
| **Screen** | **Admin** | **Staff** | **Breeder** | **Member** | **Guest** |
| --- | --- | --- | --- | --- | --- |
| Home Page | x | x | x | x | x |
| Auction Page | x | x | x | x | x |
| User Login | x | x | x | x | x |
| Past Auction Page | x | x | x | x | x |
| Reset Password | x | x | x | x | x |
| User Register |  |  |  |  | x |
| User Profile | x | x | x | x | x |
| Edit User Profile | x | x | x | x | x |
| Staff Page |  | x |  |  |  |
| Manage Staff’s Request |  | x |  |  |  |
| View requests list |  | x |  |  |  |
| Update request status |  | x |  |  |  |
| Search request by status, id,.. |  | x |  |  |  |
| View requests details |  |  |  |  |  |
| Manage Staff’s Transport |  | x |  |  |  |
| View transport list |  | x |  |  |  |
| Update transport status |  | x |  |  |  |
| Search request by status, id,.. |  | x |  |  |  |
| View transport details |  |  |  |  |  |
| Breeder Home Page |  |  | x |  |  |
| Manage Breeder Requests |  |  | x |  |  |
| View requests list |  |  | x |  |  |
| Create request details |  |  | x |  |  |
| Search request by status, id,.. |  |  | x |  |  |
| Update request details |  |  | x |  |  |
| Admin Home Page | x |  |  |  |  |
| Create Auction | x |  |  |  |  |
| Manage Requests | x |  |  |  |  |
| View requests list | x |  |  |  |  |
| View request details | x |  |  |  |  |
| Search request by status, id,.. | x |  |  |  |  |
| Assign staff to request | x |  |  |  |  |
| Negotiating with breeder request | x |  |  |  |  |
| Cancel request | x |  |  |  |  |
| Manage Transport | x |  |  |  |  |
| View transport list | x |  |  |  |  |
| View transport details | x |  |  |  |  |
| Search transport by status, id,.. | x |  |  |  |  |
| Assign staff to transport | x |  |  |  |  |
| Cancel transport | x |  |  |  |  |
| Manage Transaction | x |  |  |  |  |
| View transport list | x |  |  |  |  |
| Auctioning Page | x | x | x | x | x |
| Auction Lots Detail Page | x | x | x | x | x |
| Bidding Page |  |  |  | x |  |
| Payment Page |  |  |  | x |  |

In which:

* Guest: user who doesn't have an account.
* Member: user who has a member account.
* Staff: user who help moderate systems under permission of administrator
* Admin: who manages this system.
* Breeder : users who make auction request

## 5. Entity Relationship Diagram

*[Provide the entity relationship diagram and the entity descriptions in the table format as below]*



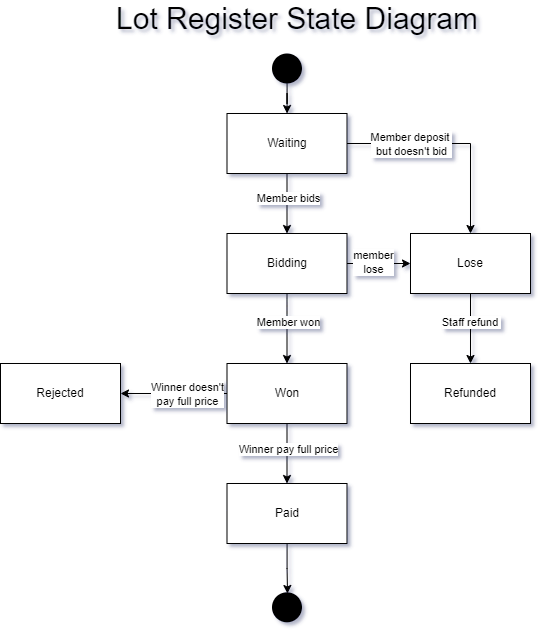
**Entities Description**

| **#** | **Entity** | **Description** |
| --- | --- | --- |
| 1 | Account | Represents the user accounts in the system |
| 2 | KoiBreeder | Represents individuals or entities that breed koi fish. Each KoiBreeder is associated with an Account and can create AuctionRequests for their koi fish |
| 3 | Member | Represents users who participate in the auction process. Members are linked to an Account and can register for lots, place bids, and receive invoices if they win. |
| 4 | AuctionRequest | Represents requests made by KoiBreeders to auction their koi fish. Each request contains details about the fish being auctioned |
| 5 | KoiFish | Represents the individual koi fish that are up for auction. |
| 6 | Variety | Represents the different breeds or types of koi fish. Each Variety can have multiple KoiFish associated with it |
| 7 | Media | Represents the media (such as images or videos) associated with each KoiFish. Each Media entry corresponds to only one KoiFish |
| 8 | LotRegister | Represents the registration of a Member to participate in a specific lot auction. Each LotRegister links a Member to a Lot, indicating their intention to bid. |
| 9 | Bid | Represents the bids placed by Members on specific Lots. Each Bid entry records the amount bid and links to both the Lot and the Member who placed the bid. |
| 10 | Lot | Represents a specific koi fish being auctioned in an auction session. Each Lot is linked to an Auction and has a defined AuctionType. |
| 11 | AuctionType | Represents the type or category of an auction. Each Lot is associated with one AuctionType, providing context for the auction format. |
| 12 | Auction | Represents the overall auction event that includes multiple Lots. Each Auction has a defined duration and can contain various types of Lots for bidding. |
| 13 | Invoice | Represents the invoices generated for Members who win bids in auctions. Each Invoice includes details about the winning Member, the KoiFish sold, and the associated Lot and LotRegister. |
| 14 | Transaction | Represents the financial transactions occurring within the system. Each Transaction records the flow of money, including deposits, refunds, and payments associated with auctions and bids. |

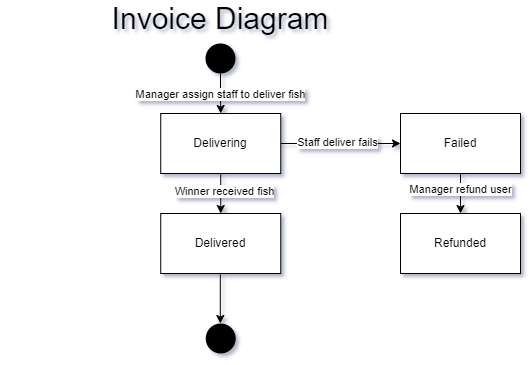
## 

## 6. State transition diagram

Picture of “Auction Request State Diagram”



Picture of “Lot Register State Diagram”



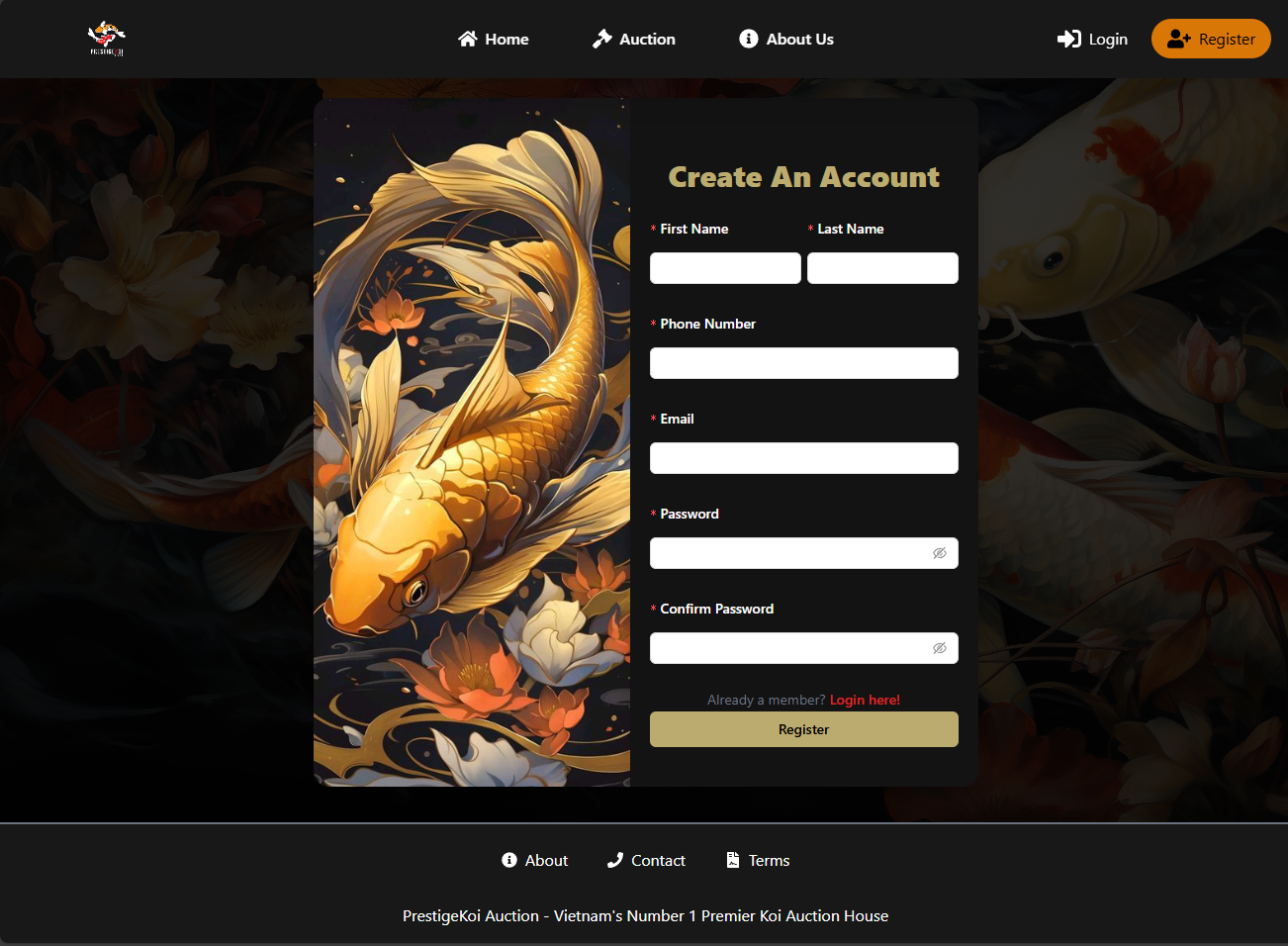
Picture of “Invoice State Diagram”

# II. Functional Requirements

## 1. Login & Sign Up

### a. Register an account

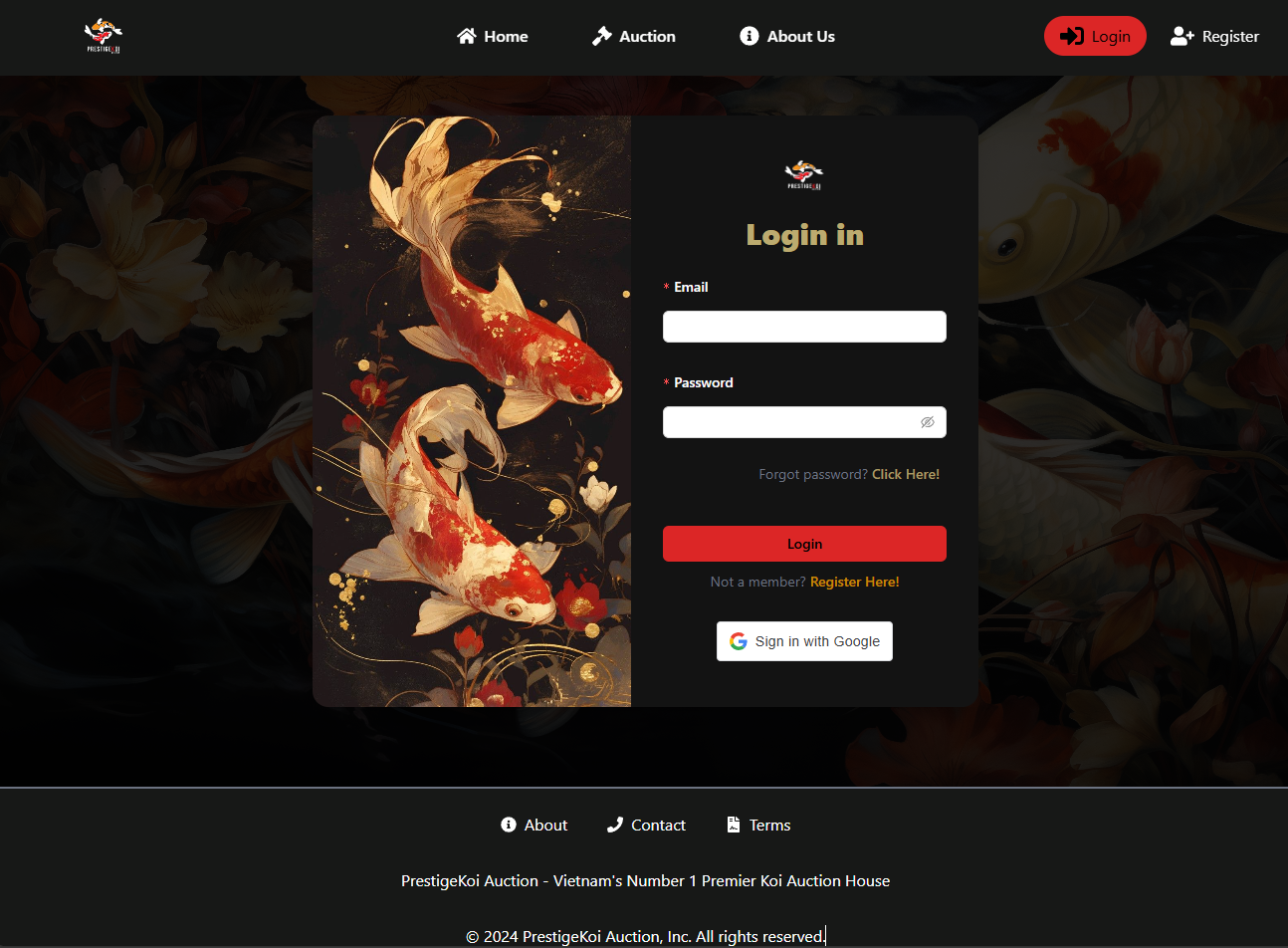
* *Function trigger: When users click on “Register Here!” on Login page*
* *Function description: Allow user to enter the necessary information to create an account*
* *Screen layout*



* *Function Details: This is a screen that allows users to input the following information: First Name, Last Name, Phone Number, Email, Password and Confirm Password for creating a new account to the system.*

### b. Login

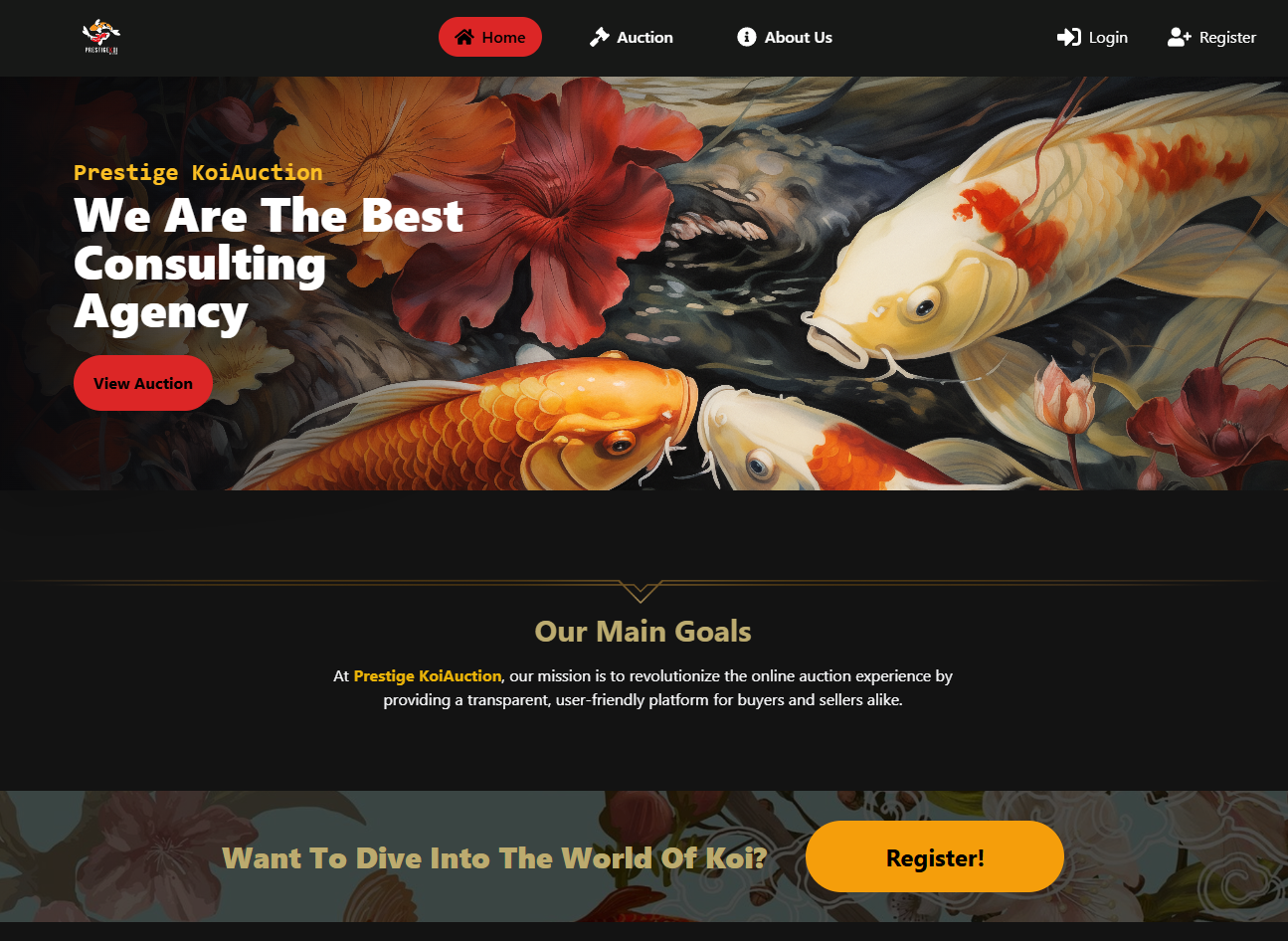
* *Function trigger: When users access a web page, and want to use the website's features.*
* *Function description: Users need to enter Username and Password or use google account to access the system.*
* *Screen layout*



* *Function Details: Users need to enter the correct Email and Password or use google account to access their account.*

## 2. Home page

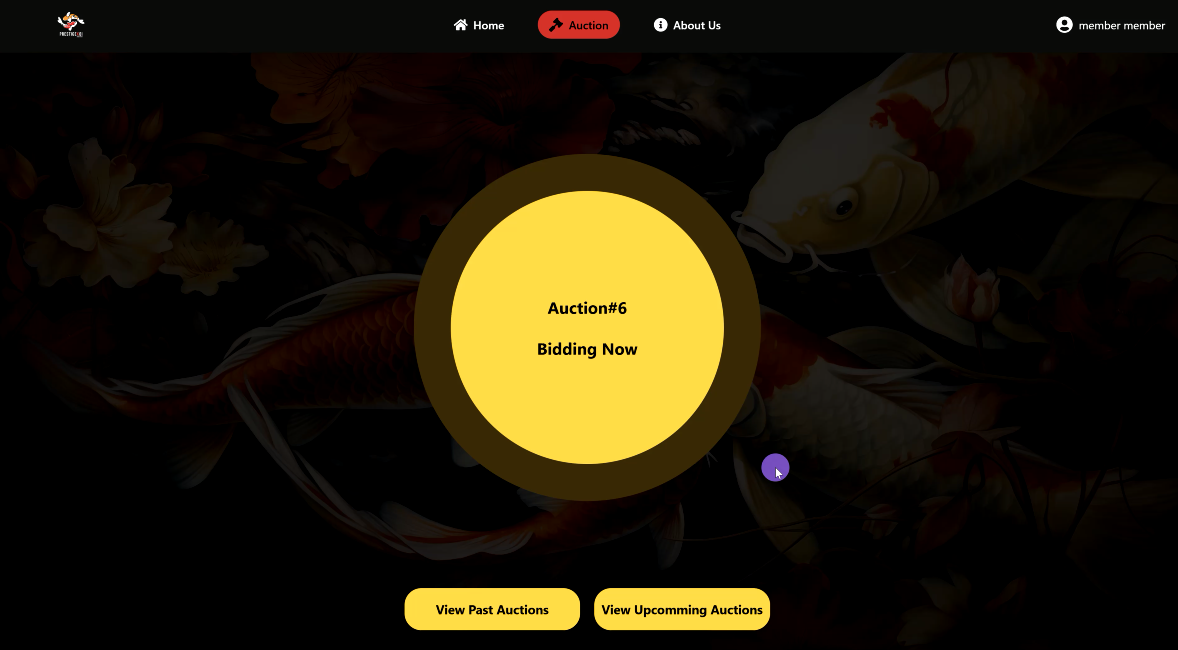
* *Function trigger: This function is triggered when the users visits the website.*
* *Function description: Users will see a brief introduction to the website when they visit the website.*
* *Screen layout:*

**

* *Function Details: This is the screen that is displayed when the user visits the website or is logged in. For users visiting the website, they can see a brief introduction to our website. For users who are logged in, they have access to Auction to participate in auction activities.*

## 3. Auction Overview

* *Function Trigger****:*** *This function is triggered when the user clicks on Auction on the Home page.*
* *Function description: Users will be presented with three options: Auctioning (Bidding now) to view ongoing auctions (if there are no ongoing auctions, users will see a message indicating "No scheduled auctions"), View Past Auction to see completed auctions, and View Upcoming for upcoming auctions.*
* *Screen layout:*

**

* *Function Details: Upon selecting an option, the user can view relevant information about each auction type. For Auctioning, they can see and participate in live auctions. For View Past Auction, they can review completed auctions and their details. For View Upcoming, they can preview information on auctions that will be available soon.*

## 4. Auction Participation (For User)

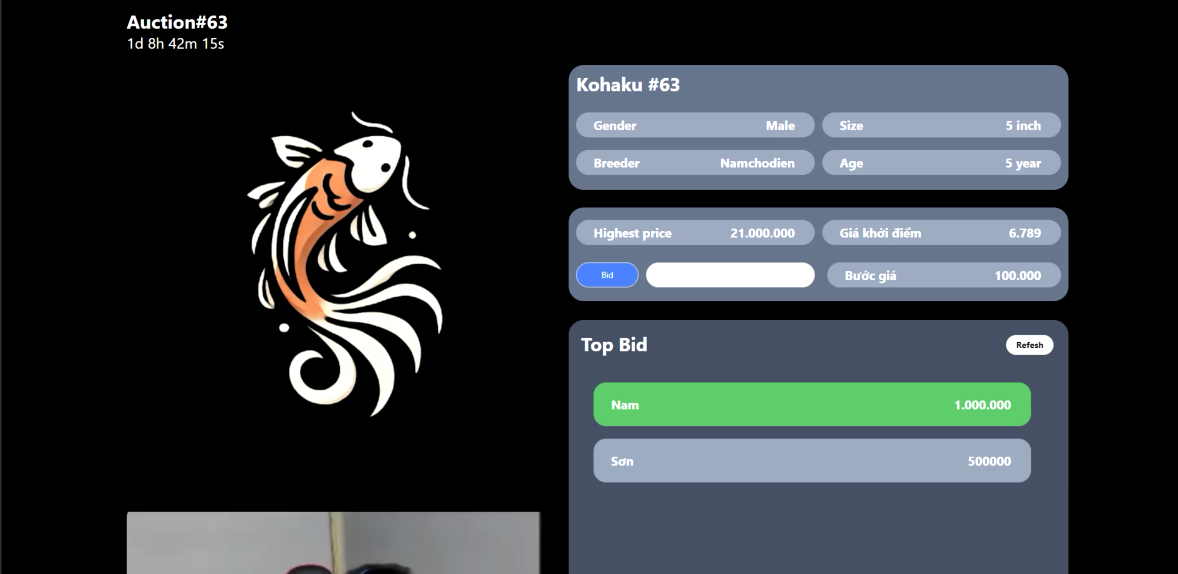
### a. Selecting lot

* *Function Trigger****:*** *This function is triggered when the user clicks on Bidding Now.*
* *Function description: Allows users to view multiple lots and participate in lots.*
* *Screen layout:*

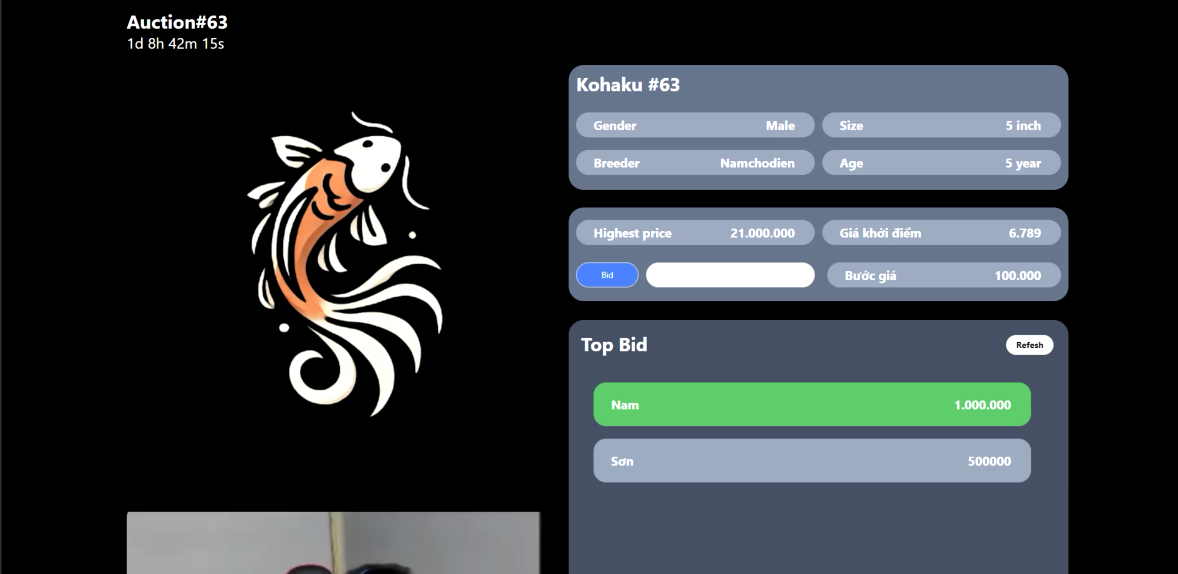
### 

* *Function Details: There are many lots and each lot contains 1 Koi fish for auction. The user can select 1 lot to auction that Koi fish.*

### b. Depositing

* *Function Trigger****:*** *This function is triggered when the user clicks on a lot.*
* *Function description: The user can click on the “Deposit here!” button to make a deposit.*
* *Screen layout: *
* *Function Details: In order for the user to be able to bid, they are required to make a deposit first. To make a deposit, they click on the "Deposit here" button and then proceed to pay.*

### c. Bidding

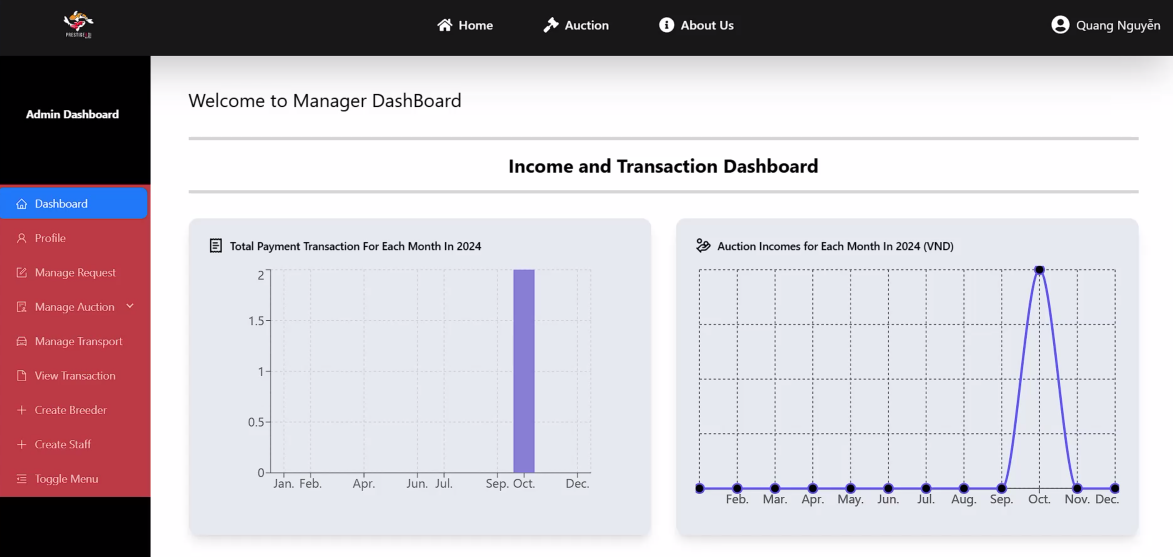
* *Function Trigger****:*** *This function is triggered when the user has successfully deposited.*
* *Function description: Users can enter a price then press the "Bid" button to evaluate..*
* *Screen layout:*
* **
* *Function Details: Once the user has successfully deposited, they can enter a bid amount in the provided input field. After entering the desired bid, the user presses the "Bid" button to submit their offer. The system will then evaluate the bid to ensure it meets the minimum required increment and is higher than the current highest bid. If the bid is valid, it will be registered, and the user will see an updated display showing their bid as the highest. If the bid is invalid, an error message will notify the user of the issue.*

## 5. Payment

* *Function Trigger****:*** *This function is activated when the user has won that lot.*
* *Function description: When the lot ends, the winning user will click on the "Payment here" notification to proceed with the payment.*
* *Screen layout:*
* *Function Details: When the auction ends, the winning user will click on the "Payment here" notification. Next, the user will select the location to receive the fish and click "Calculate" for the system to compute the total amount due (including shipping fees, auction fees, and deposit refunds). Then, a payment link will appear, and the user can click on it to proceed with the payment.*

## 6. Dashboard (For Manager)

* *Function Trigger****:*** *This function is activated after the manager login .*
* *Function description: After logging in, the manager will see charts about requests of breeders and income.*
* *Screen layout:*

**

* *Function Details: After logging in, the manager will see a chart of the number of requests, number of invoices and total income by month.*

## 7. Request Management (For Manager)

* *Function Trigger****:*** *This function is activated after the manager login and chooses “Manage Request”.*
* *Function description: The manager will see requests of breeders.*
* *Screen layout:*
* *Function Details: The manager has the right to view the requests, can assign the staff to check the fish, offer price and auction type for the breeder.*

## 8. Auction Creating (For Manager)

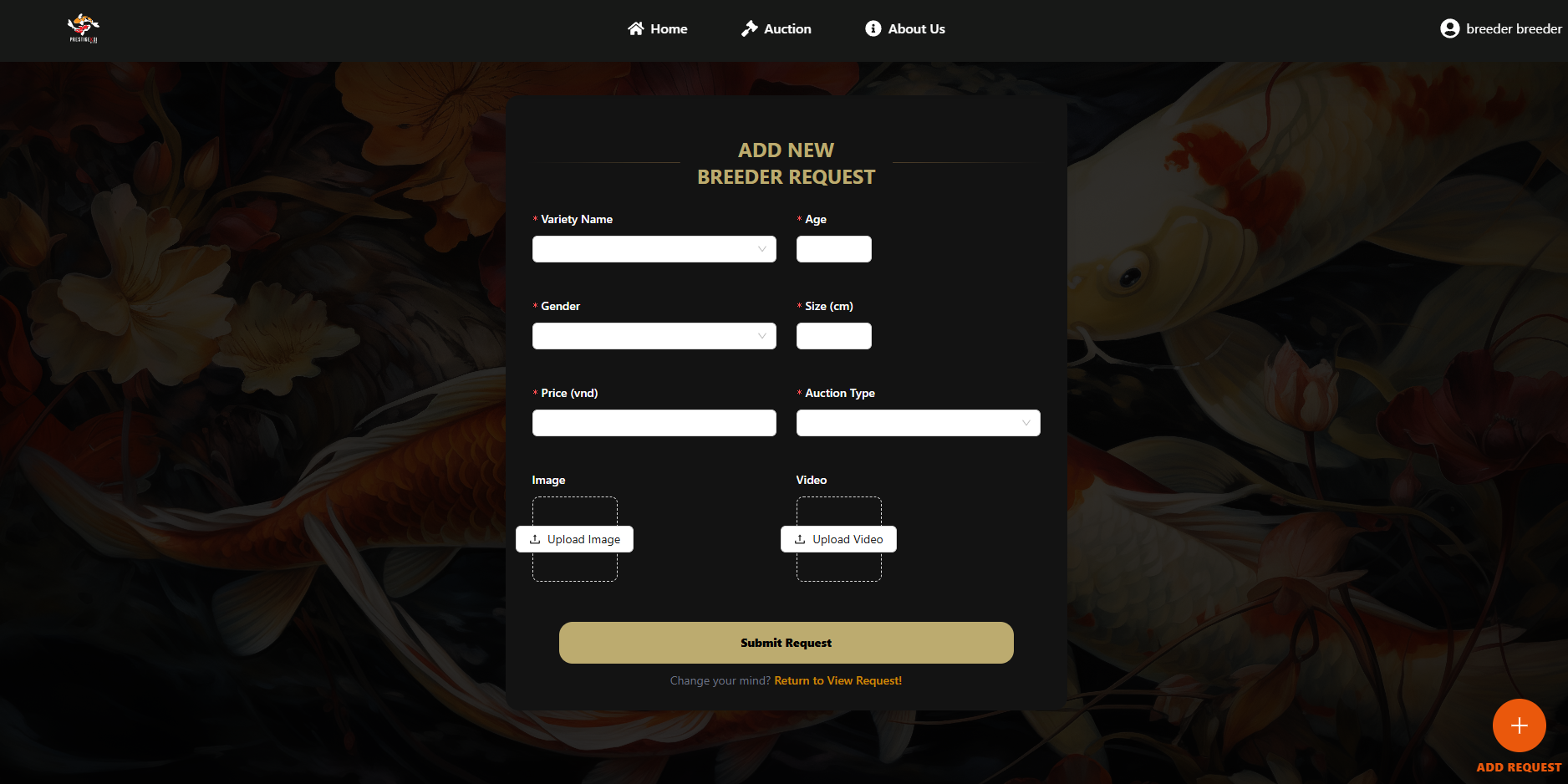
* *Function Trigger****:*** *This function is activated after the manager login and chooses “Create Auction” in “Manage Auction”.*
* *Function description: The manager will select information and Koi fish to create an auction.*
* *Screen layout:*
* *Function Details: The manager can select the start time, end time, and auction type. Then, the manager will choose multiple Koi fish. Finally, the manager will confirm the auction information one last time before creating the auction.*

## 9. Request Management (For Staff)

* *Function Trigger****:*** *This function is activated after the staff login and chooses “Manage Request”.*
* *Function description: The staff will see requests from the manager and respond to the manager.*
* *Screen layout:*
* *Function Details: The staff will see the requests of the managers who have assigned them requests and they can respond to approved or unapproved requests.*

## 10. Create Request (For Breeder)

* *Function Trigger****:*** *This function is activated after the breeder login and chooses “Add Request”.*
* *Function description: The Breeder will fill in the information and create the request.*
* *Screen layout:*

**

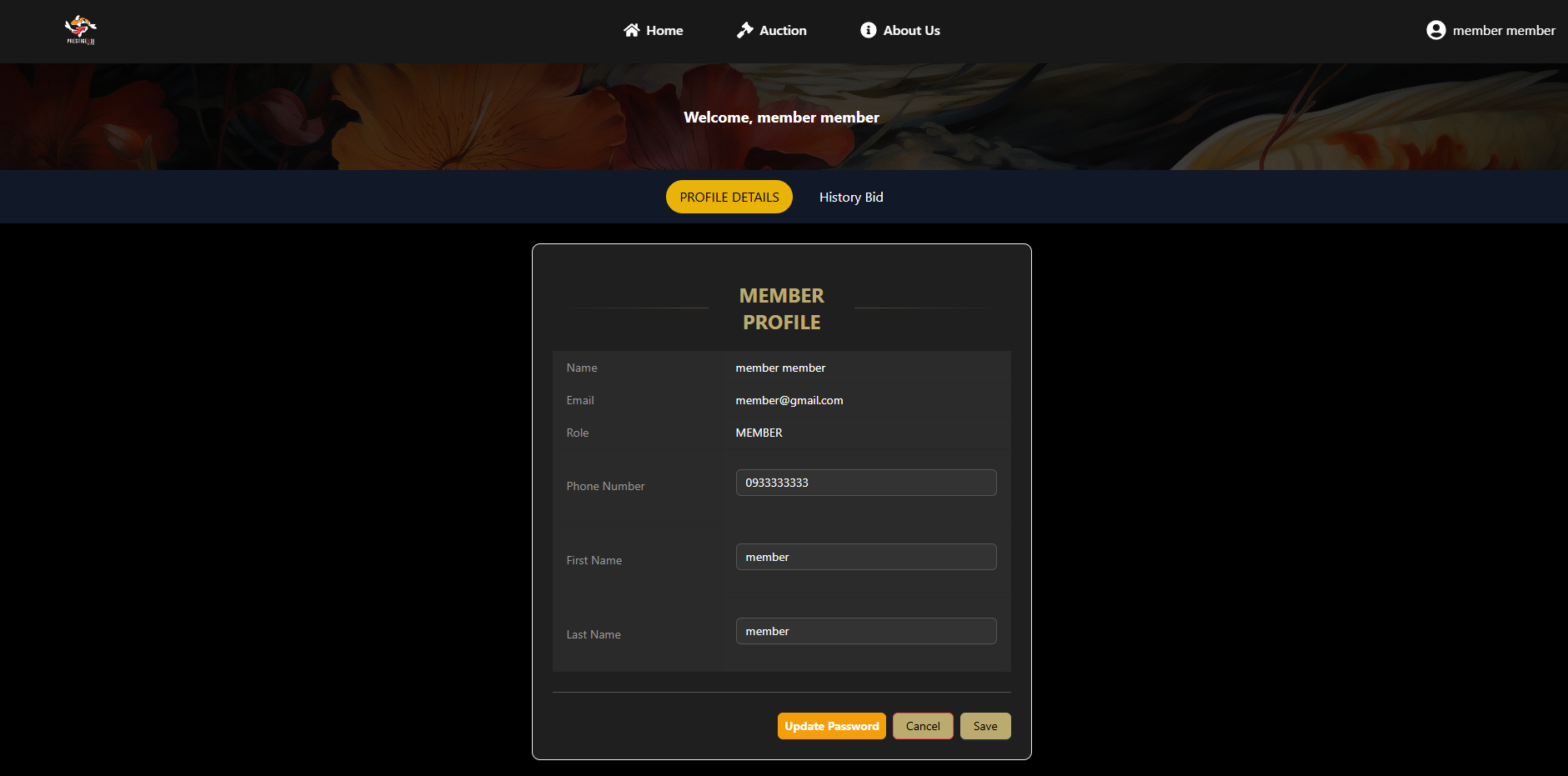
* *Function Details: This is a screen that allows the breeder to input the following information: Variety Name, Age, Gender, Size, Price, Auction Type, Image, Video.*

## 11. Request Management(For Breeder)

* *Function Trigger****:*** *This function is activated after the breeder login, clicks profile and chooses “View Request”.*
* *Function description: The Breeder will see the request they have created.*
* *Screen layout:*
* *Function Details: This is a screen that allows the breeder to view all requests they have created. They also can click view detail to see more information and they can cancel or update requests when status is “Requesting”.*

## 11. Update profile

* *Function Trigger****:*** *This function is activated after the breeder, user, manager, staff click to profile.*
* *Function description: The breeder, user, manager, staff edit personal information.*
* *Screen layout:*

**

* *Function Details: This is a screen that allows the breeder, user, manager, staff edit the following personal information: Phone Number, First Name, Last Name.*

# III. Database Design

## 1. Database Schema

*[Provide the tables relationship like example below – following MySQL database naming convention]*

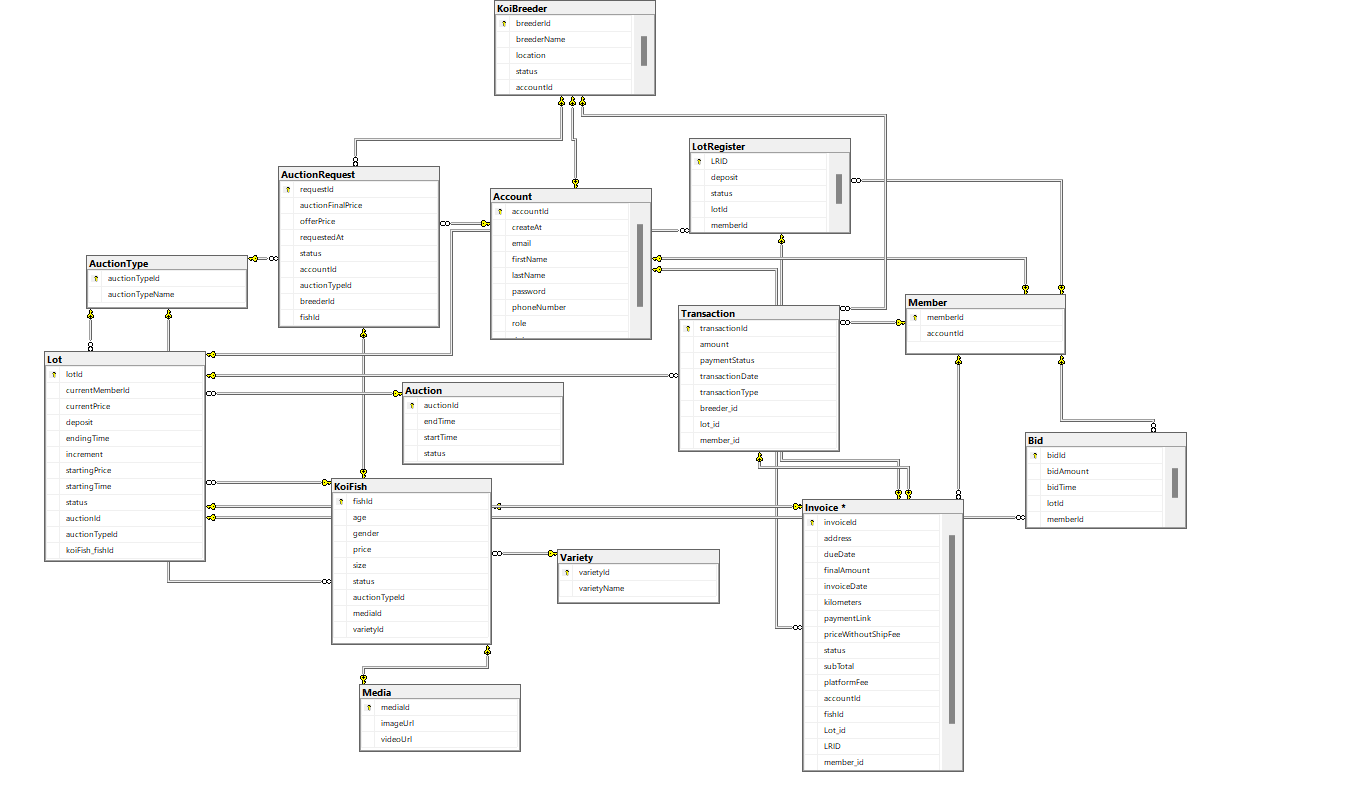
**Table Description**

| **No** | **Table** | **Description** |
| --- | --- | --- |
| 1 | Account | Information about accounts in the system, including admin, member, manager |
| 2 | Member | Contains details of all accounts in the system with the role of "MEMBER." |
| 3 | KoiBreeder | Stores details of all accounts with the role of "BREEDER." |
| 4 | AuctionRequest | Records requests from breeders to auction their Koi fish. |
| 5 | KoiFish | Contains information on all Koi fish submitted by breeders for auction. |
| 6 | Media | Holds image and video data related to Koi fish. |
| 7 | Variety | Lists available Koi fish varieties for breeders to select when creating auction requests. |
| 8 | AuctionType | Contains data about different auction types available in the system. |
| 9 | Auction | Stores information about auctions hosted in the system. |
| 10 | Lot | Holds information about individual lots within each auction. |
| 11 | LotRegister | Records details of members who registered for specific lots within an auction. |
| 12 | Bid | Contains the history of all bids placed on each lot. |
| 13 | Invoice | Stores invoice data generated for lot winners. |
| 14 | Transaction | Records all transactions related to invoices and payments within the system. |

## 2. Table Detail

*[Give some lines about the table here>>*

*[Table fields, in the form of table format as below]*

******

**Table Account**

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | accountId | int |  | X | X | X | ID of account |
| 2 | createAt | datetime2 | 6 |  |  |  | Date and time the account was created |
| 3 | email | varchar | 255 | X | X |  | Email of account |
| 4 | firstName | nvarchar | 100 |  |  |  | First name of account |
| 5 | lastName | nvarchar | 100 |  |  |  | Last name of account |
| 6 | password | varchar | 255 |  | X |  | Password for account authentication |
| 7 | phoneNumber | varchar | 255 |  |  |  | Contact number of the account holder |
| 8 | role | varchar | 255 |  |  |  | Role of the account (e.g., Admin, Member, Breeder, Manager) |
| 9 | status | bit |  |  | X |  | Account status (1 for active, 0 for inactive) |

**Table Member**

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | memberId | int |  | X | X | X | Unique ID for each member |
| 2 | accountId | int |  | X | X | X | Foreign key linking to the Account table |

**Table Breeder**

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | breederId | int |  | X | X | X | Unique ID for each breeder |
| 2 | breederName | varchar | 255 |  | X |  | Name of the breeder |
| 3 | location | varchar | 255 |  | X |  | Location of the breeder |
| 4 | status | bit |  |  | X |  | Status (1 for active, 0 for inactive) |
| 5 | accountId | int |  | X | X | X | Foreign key linking to the Account table |

**Table AuctionRequest**

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | requestId | int |  | X | X | X | Unique ID for each auction request |
| 2 | auctionFinalPrice | float |  |  |  |  | Price to pay for breeder after user paid and received fish |
| 3 | offerPrice | float |  |  | X |  | Price offered by the breeder |
| 4 | requestedAt | datetime |  |  |  |  | Date the auction request was made |
| 5 | status | varchar | 255 |  | X |  | Status of the auction request |
| 6 | accountId | int |  |  |  | X | Foreign key linking to the Account table |
| 7 | auctionTypeId | int |  |  |  | X | Foreign key linking to the AuctionType table |
| 8 | breederId | int |  |  |  | X | Foreign key linking to the Breeder table |
| 9 | fishId | int |  |  |  | X | Foreign key linking to the KoiFish table |

**Table KoiFish**

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | fishId | int |  | X | X | X | Unique ID for each Koi fish |
| 2 | age | int |  |  | X |  | Age of the Koi fish |
| 3 | gender | varchar |  |  | X |  | Gender of the Koi fish |
| 4 | price | float |  |  | X |  | Price of the Koi fish |
| 5 | size | float |  |  | X |  | Size of the Koi fish in cm |
| 6 | status | varchar | 255 |  | X |  | Status of the Koi fish |
| 7 | auctionTypeId | int |  |  |  | X | Foreign key linking to the AuctionType table |
| 8 | mediaId | int |  |  |  | X | Foreign key linking to the Media table |
| 9 | varietyId | int |  |  | X | X | Foreign key linking to the Variety table |

**Table Variety**

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | varietyId | int |  | X | X | X | Unique ID for each Koi variety |
| 2 | varietyName | varchar | 255 | X | X |  | Name of the Koi variety |

**Table Media**

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | mediaId | int |  | X | X | X | Unique ID for each media record |
| 2 | imageUrl | varchar | 255 |  | X |  | URL of the image |
| 3 | videoUrl | varchar | 255 |  | X |  | URL of the video |

**Table AuctionType**

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | auctionTypeId | int |  | X | X | X | Unique ID for each auction type |
| 2 | auctionTypeName | varchar | 255 |  | X |  | Name of the auction type |

**Table Auction**

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | mediaId | int |  | X | X | X | Foreign key linking to the Media table |
| 2 | endTime | datetime2 |  |  | X |  | End time of the auction |
| 3 | startTime | datetime2 |  |  | X |  | Start time of the auction |
| 4 | status | varchar | 255 |  | X |  | Status of the auction |

**Table Lot**

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | lotId | int |  | X | X | X | Unique ID for each lot |
| 2 | currentMemberId | int |  |  |  |  | Current highest bidder member ID |
| 3 | currentPrice | float |  |  | X |  | Current highest bid price |
| 4 | deposit | float |  |  | X |  | Required deposit for the lot |
| 5 | endingTime | datetime |  |  | X |  | Ending time of the lot |
| 6 | increment | float |  |  | X |  | Bid increment value |
| 7 | startingPrice | float |  |  | X |  | Starting price of the lot |
| 8 | startingTime | datetime |  |  | X |  | Start time of the lot |
| 9 | status | varchar |  |  | X |  | Status of the lot |
| 10 | auctionId | int |  |  | X | X | Foreign key linking to the Auction table |
| 11 | auctionTypeId | int |  |  | X | X | Foreign key linking to the AuctionType table |
| 12 | koiFish\_fishId | int |  |  | X | X | Foreign key linking to the KoiFish table |

**Table LotRegister**

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | LRID | int |  | X | X | X | Unique ID for each lot registration |
| 2 | deposit | float |  |  | X |  | Deposit amount for the lot registration |
| 3 | status | varchar |  |  | X |  | Status of the registration |
| 4 | lotId | int | 255 |  | X |  | Foreign key linking to the Lot table |
| 5 | memberId | int |  |  | X |  | Foreign key linking to the Member table |

**Table Bid**

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | bidId | int |  | X | X | X | Unique ID for each bid |
| 2 | bidAmount | float |  |  | X |  | Amount of the bid |
| 3 | bidTime | datetime |  |  | X |  | Time the bid was placed |
| 4 | lotId | int |  |  | X | X | Foreign key linking to the Lot table |
| 5 | memberId | int |  |  | X | X | Foreign key linking to the Member table |

**Table Invoice**

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | invoiceId | int |  | X | X | X | Unique ID for each invoice |
| 2 | address | nvarchar | MAX |  |  |  | Billing or shipping address |
| 3 | dueDate | datetime |  |  | X |  | Due date for payment |
| 4 | finalAmount | float |  |  | X |  | Total amount including shipping fees and platform fees |
| 5 | invoiceDate | datetime |  |  | X |  | Date of invoice issuance |
| 6 | kilometers | float |  |  |  |  | Distance in kilometers for delivery |
| 7 | paymentLink | varchar | 2048 |  |  |  | Link for online payment |
| 8 | priceWithoutShipFee | float |  |  |  |  | Price excluding shipping fees |
| 9 | status | varchar | 255 |  |  |  | Status of the invoice (e.g., paid, pending) |
| 10 | subTotal | float |  |  | X |  | Raw bid price used to calculate finalAmount |
| 11 | platformFee | float |  |  |  |  | Platform or service fee |
| 12 | accountId | int |  |  | X | X | Foreign key linking to the Account table |
| 13 | fishId | int |  |  | X | X | Foreign key linking to the KoiFish table |
| 14 | Lot\_id | int |  |  | X | X | Foreign key linking to the Lot table |
| 15 | member\_id | int |  |  | X | X | Foreign key linking to the Member table |
| 16 | LRID | int |  |  | X | X | Foreign key linking to the LotRegister table |
| 17 | transaction\_id | int |  |  | X | X | Foreign key linking to the Transaction table |

**Table Transaction**

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | transactionId | int |  | X | X | X | Unique ID for each transaction |
| 2 | amount | float |  |  | X |  | Transaction amount |
| 3 | paymentStatus | varchar |  |  | X |  | Status of payment (e.g., completed, pending) |
| 4 | transactionDate | datetime |  |  | X |  | Date of the transaction |
| 5 | transactionType | varchar |  |  | X |  | Type of transaction (e.g., payment, refund) |
| 6 | breeder\_id | int |  |  |  | X | Foreign key linking to the Breeder table |
| 7 | lot\_id | int |  |  | X | X | Foreign key linking to the Lot table |
| 8 | member\_id | int |  |  |  | X | Foreign key linking to the Member table |