

Koi Care System At Home

Software Requirement Specification

Project Code: SE1863

Document Code: SE1863 – 7.0

Class Code: SE1876

Group Code: 4

**Binh Duong, 16/9/2024**

TABLE OF CONTENTS

[1 Introduction 5](#_Toc179354700)

[1.1 Purpose 5](#_Toc179354701)

[1.2 Scope 5](#_Toc179354702)

[1.3 Definitions, Acronyms, and Abbreviations 5](#_Toc179354703)

[1.4 Overview 6](#_Toc179354705)

[2 Overall Description 7](#_Toc179354706)

[2.1 Product Perspective 7](#_Toc179354707)

[2.2 User classes and characteristics 7](#_Toc179354708)

[2.3 Design and implementation constraints 7](#_Toc179354709)

[3 FUNCTIONAL Requirements 8](#_Toc179354710)

[3.1 Swimlane Diagrams 8](#_Toc179354711)

[3.2 Use Case Diagrams 8](#_Toc179354712)

[3.3 State Diagrams 9](#_Toc179354715)

[3.4 Logical Data Model 9](#_Toc179354716)

[3.5 Wire Flows 9](#_Toc179354717)

[4 NON-FUNCTIONAL Requirements 10](#_Toc179354718)

[4.1 Usability 10](#_Toc179354719)

[4.2 Reliability 10](#_Toc179354720)

[4.3 Performance 11](#_Toc179354721)

[4.4 Security 11](#_Toc179354722)

# Introduction

## Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the functional and non-functional requirements for the Koi care system at home. This document provides a detailed description of the system's features, behavior, and constraints to ensure that the project meets both business and user needs. It serves as a communication bridge between stakeholders, developers, and designers to maintain clarity and alignment throughout the project lifecycle.

## Scope

The **scope** of the system described in the Software Requirements Specification (SRS) outlines the key functionalities, user roles, and interactions the system will support. The system is designed to serve multiple user types (Guests, Members, Shops, and Admins), each having different levels of access and capabilities. The system will manage koi ponds, facilitate online shopping, and provide administrative oversight for user accounts and orders.

## Definitions, Acronyms, and Abbreviations

* **Account**: A personal user profile containing information like username, password, email, and other relevant details that allow access to the system.
* **CRUD**: An acronym for Create, Read, Update, and Delete. It refers to the basic operations of managing any database record or entity.
* **Guest**: A user who has not yet registered or logged into the system. They have limited access to functionalities such as viewing news, blogs, and creating an account.
* **Registered User**: A user who has successfully created an account and logged into the system. They have access to additional features like managing their personal information and logging out.
* **Member**: A registered user who has additional privileges, such as managing koi, ponds, shopping features (like managing the cart and checkout), and viewing personal orders.
* **Admin**: An administrator with the highest level of access to the system. Admins manage user accounts, blogs, news, product categories, revenue, and other system settings.
* **Blog**: An online entry where users or admins can post articles or updates related to koi, ponds, or other relevant topics. Admins are responsible for confirming and managing blog entries.
* **News**: Updates or announcements available to all users, including guests, regarding koi, ponds, shop products, or general system-related news.
* **Cart**: A virtual basket that allows users to collect products they wish to purchase. Users can add, remove, and adjust product quantities before proceeding to checkout.
* **Wishlist**: A feature that allows users to save products for future consideration without committing to purchase them immediately.
* **Checkout**: The process of confirming the purchase of items in the cart, selecting a payment method, and finalizing the order.
* **Order**: A request made by a member to purchase products. Orders are tracked by status and can be reviewed by both members and admins.
* **Pond**: A user-managed entity in the system representing a physical pond where koi are kept. The system allows users to input pond details and track maintenance requirements.
* **Koi**: Refers to the fish that are being managed by members. Users can add koi information, track their growth, and view related statistics.
* **Payment Methods**: Options for members to pay for their purchases, typically including cash and bank transfer options.
* **Shipping Address**: The destination address provided by the member for the delivery of products purchased from the shop.
* **Revenue**: Refers to the income generated by the shop through various means, including member subscriptions and commission fees.
* **Order Status**: The current state of an order, which could include stages like processing, shipped, delivered, or completed.

## Overview

The koi management system is designed to support various user roles, enabling effective management of koi, ponds, and related products. The system provides a comprehensive platform that integrates koi care, e-commerce, and content management, tailored for different types of users: **Guests**, **Registered Users**, **Members**, **Shop Owners**, and **Admins**. Each user role has specific functionalities aimed at enhancing the user experience and ensuring smooth operations for both koi management and shop transactions.

The system’s primary objectives include:

* Facilitating the **management of koi and ponds** for members, including tracking fish growth and monitoring water parameters.
* Offering **e-commerce capabilities**, allowing users to browse products, manage shopping carts, and complete purchases.
* Providing a platform for **content creation** where users can access blogs and news related to koi care, while admins oversee content management.
* Enabling **shop owners** to manage products, categories, and orders efficiently.
* Allowing **admins** to maintain system integrity by managing user accounts, revenue, and overseeing content.

# Overall Description

## Product Perspective

The koi management system is an integrated web-based application that serves multiple user roles such as Guests, Registered Users, Members, Shop Owners, and Admins. The system combines functionality for koi and pond management with an e-commerce platform that enables users to browse, purchase products, and manage orders. Additionally, the system supports content management through blogs and news articles.

## User classes and characteristics

* **Guest**: Unregistered users who can view news, blogs, and register for an account.
* **Member**: Registered users with access to manage koi, ponds, shopping carts, wishlists, and place orders.
* **Shop Owner**: Manages the shop’s products and orders, with CRUD capabilities for products and categories, and can update order statuses.
* **Admin**: Has the highest level of access, managing user accounts, revenue, blogs, news, and the entire system.

## Design and implementation constraints

* **Technology**: Uses ASP.NET Core for the backend and SQL Server for data management.
* **Security**: Adheres to security standards (OAuth, data encryption, secure API communication).
* **Performance**: The system must handle high traffic efficiently.
* **Scalability**: Designed for scalability to support growth in users and data.
* **Compliance**: Ensures compliance with regulations like GDPR.
* **Third-Party Integration**: Integrates with payment.
* **Device Compatibility**: Responsive interface supporting various devices and browsers.

# FUNCTIONAL Requirements

Guest:

 FR-01: Can Register an Account

* Guests can create a new account.

 FR-02: Can Log In/Log Out

* Guests can log in or log out of the system.

 FR-03: View News

* Guests can search and read news.

 FR-04: View Blogs

* Guests can search and read blogs.

Registered User (Member, Shop, Admin):

 FR-05: Must Manage Personal Account

* Users can update their personal information.
* Users can log out of the system.

 FR-06: Can Log In to System

* Registered users can log in to their account.

Member

 FR-07: Must Manage Koi

* Create, read, update, and delete (CRUD) fish information.
* View fish growth chart.

 FR-08: Must Manage Pond

* Create, read, update, and delete (CRUD) pond information.
* Calculate salt and food requirements.
* View water parameter statistics chart.

 FR-09: Can View Shop’s Product List

* Allow users to view product list and can search product detail.

 FR-10: Must Manage Cart

* Add products to the cart.
* View cart contents.
* Update cart (Change products quantity and remove product from cart).

 FR-11: Can Add Products to Wishlist

* Allow users to add products to a wishlist for future reference.

 FR-12: Must Checkout

* Review cart contents before checkout.
* Select payment method (by cash or bank transfer).
* Select shipping address.
* View order summary and create an order upon confirmation.

 FR-13: Must Manage Personal Orders

* View order history.
* Track current order status.

Shop:

 FR-14: Must Manage Product List

* CRUD products
* Search products.

 FR-15: Must Manage Order List

* View order.
* Update Order Status.

Admin:

 FR-16: Must Manage Categories List

* CRUD Categories.
* Search Categories.

 FR-17: Must Manage Account List

* CRUD Account
* Search Account.

 FR-18: Must Manage Revenue.

* View Revenue by Member Subscription Chart.
* View Revenue by Commission fees Chart.

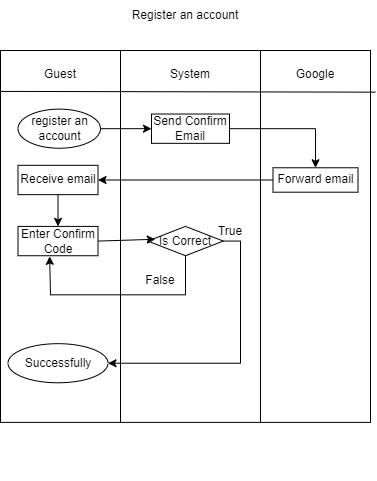
 FR-19: Must Manage Blogs

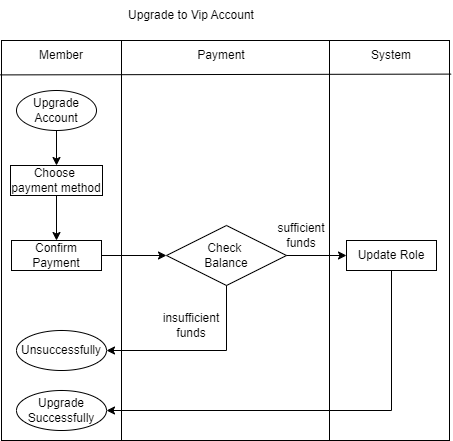
* Confirm the request to create blogs.
* Update Blogs Status (Locked/ UnLocked).
* Delete Blogs.

 FR-20: Must Manage News

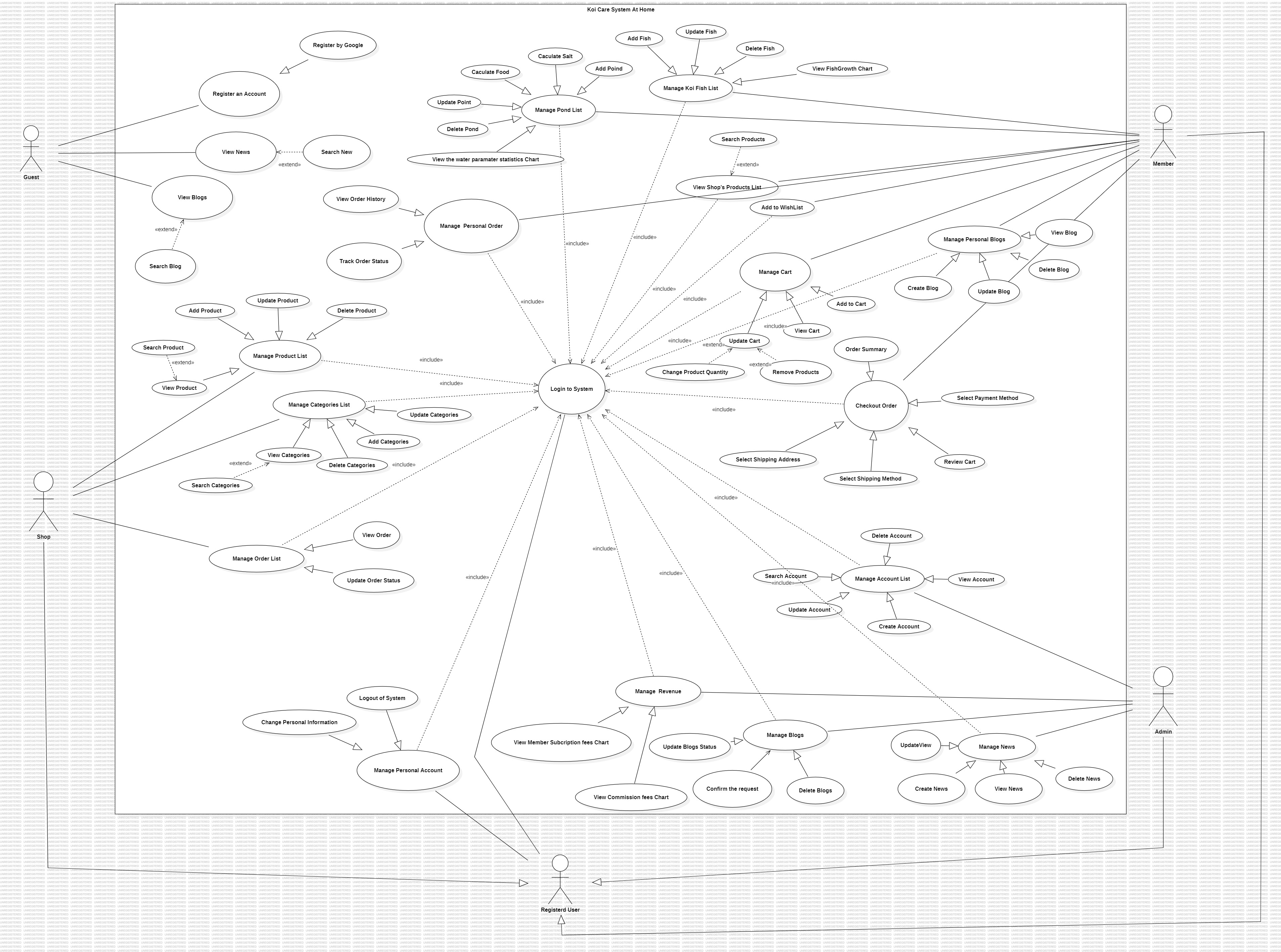
CRUD News.

## Swimlane Diagrams





## Use Case Diagrams



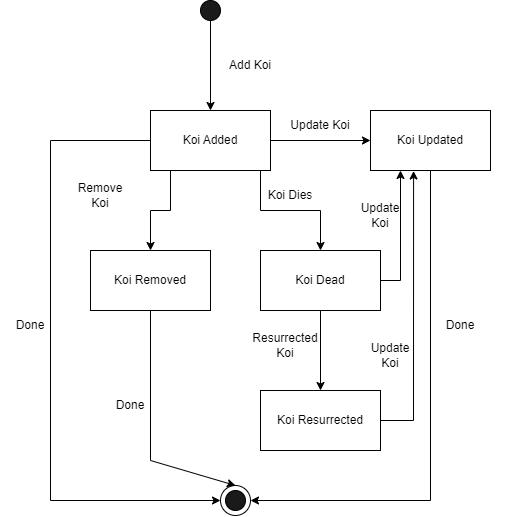
## Manage Personal Account

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE-1 SPECIFICATION | | | | |
| Use-case No. | <UC001> | Use-case Version | | <1.0> |
| Use-case Name | Manage Personal Account | | | |
| Author | Le Vinh Phat | | | |
| Date | 20/10/2024 | Priority | High | |
| Actor:  Registered User (Shop,Member,Admin)  Summary:  The user can update personal information such as name, email, password,…  Goal:  The goal of this use case is to ensure users can modify and update their account details to keep the system up to date and accurate  Triggers  User selects the "Manage Personal Account" option after logging in.  Preconditions:  The user must be logged into the system.  Post Conditions:   * The personal account information is updated in the system. * The changes are reflected in the user's profile and applied immediately.   Main Success Scenario:   1. User logs into the system. 2. User navigates to the "Manage Personal Account" section. 3. User selects the desired action (e.g., update personal details). 4. User inputs the necessary changes, such as updating name, email, or password. 5. System validates the input, ensuring the data meets required formats (e.g., valid email format, password strength). 6. If validation is successful, the system updates the personal account information in the database. 7. System confirms the successful update by displaying a confirmation message. 8. The updated information is reflected immediately in the user's profile.   Alternative Scenario:   * If the user enters invalid or incomplete data, the system shows an error message indicating what needs to be corrected (e.g., incorrect email format, weak password). The user is prompted to correct the information before the system saves the changes. * If the user cancels the action, no changes are made, and the system returns to the previous screen without updating the account details.   Exceptions:   * The system is down or unable to process requests: The user cannot access the "Manage Personal Account" feature, and the system displays an appropriate error message or notification. * User does not have permission to update certain fields: Some fields may be locked or restricted from user modification (e.g., user roles, account creation date), and an error message will be shown when attempting to update those fields. * The account to be managed does not exist: If the user’s account is somehow missing from the system, the user will not be able to proceed, and the system will show an error message.   Relationships:   * Depends on the "Login to System" use case: The user must log in to access their personal account management options. * Relates to "Change Personal Information": Users can update personal details, which is a part of the larger "Manage Personal Account" process.   Business Rules:   * Only authenticated users can manage their own personal accounts. * Personal account data must be validated to ensure compliance with system policies (e.g., valid email format, password strength). * The system should ensure that users can only manage fields that they are authorized to update. | | | | |

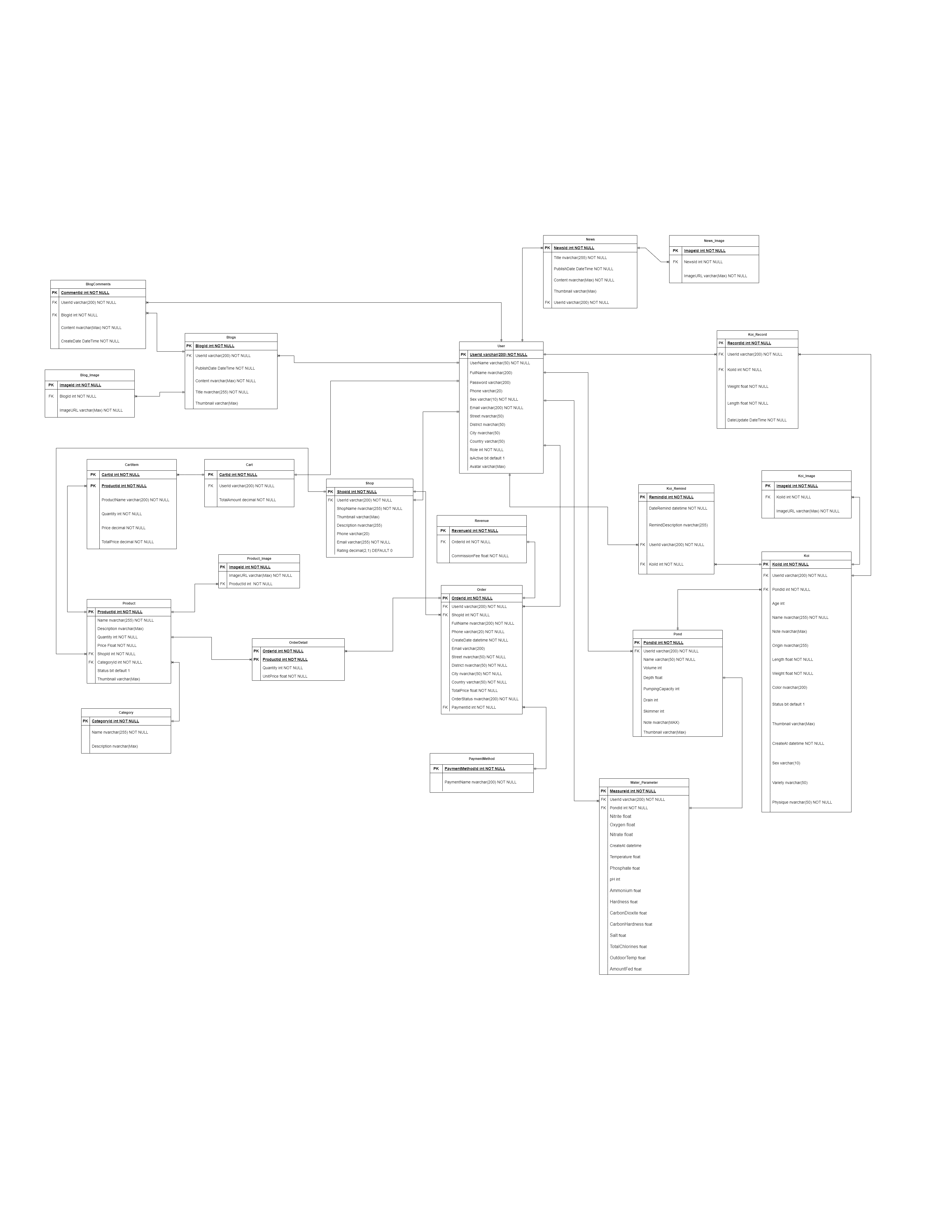
## Order Managent

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE-2 SPECIFICATION | | | | |
| Use-case No. | Use-case 2 | Use-case No. | | Use-case No. |
| Use-case Name | Order Management | | | |
| Author | Nguyen Duc Thang | | | |
| Date | Date | Date | Date | |
| Actor:  Shop  Summary:  A use case that allows shop users to view and manage the list of orders in the system, including viewing order details and updating order statuses.  Goal:  To enable efficient management of orders through a centralized interface where shop users can view orders and update their statuses.  Triggers   * Shop user needs to view orders * Shop user needs to update order status * Periodic order review requirement   Preconditions:   * User must be logged into the system * User must have shop role permissions * System must contain existing orders   Post Conditions:   * Order list is successfully displayed * Order status changes are saved in the system * System logs are updated with any changes made   Main Success Scenario:   1. Shop user logs into the system 2. System displays order management interface 3. User views list of orders 4. User can select specific orders 5. User can update order status as needed 6. System saves and confirms changes   Alternative Scenario:   * User can search/filter orders * User can sort orders by different criteria * User can export order list * User can generate order reports   Exceptions:   * System cannot retrieve order list * Database connection error * Invalid order status update * Unauthorized access attempt   Relationships:   * Includes "View Order" * Includes "Update Order Status" * Related to "Login to System"   Business Rules:   * Only authorized shop users can access order management * All order status changes must be logged * Completed orders cannot be modified * Order history must be maintained * Status updates must follow predefined workflow | | | | |

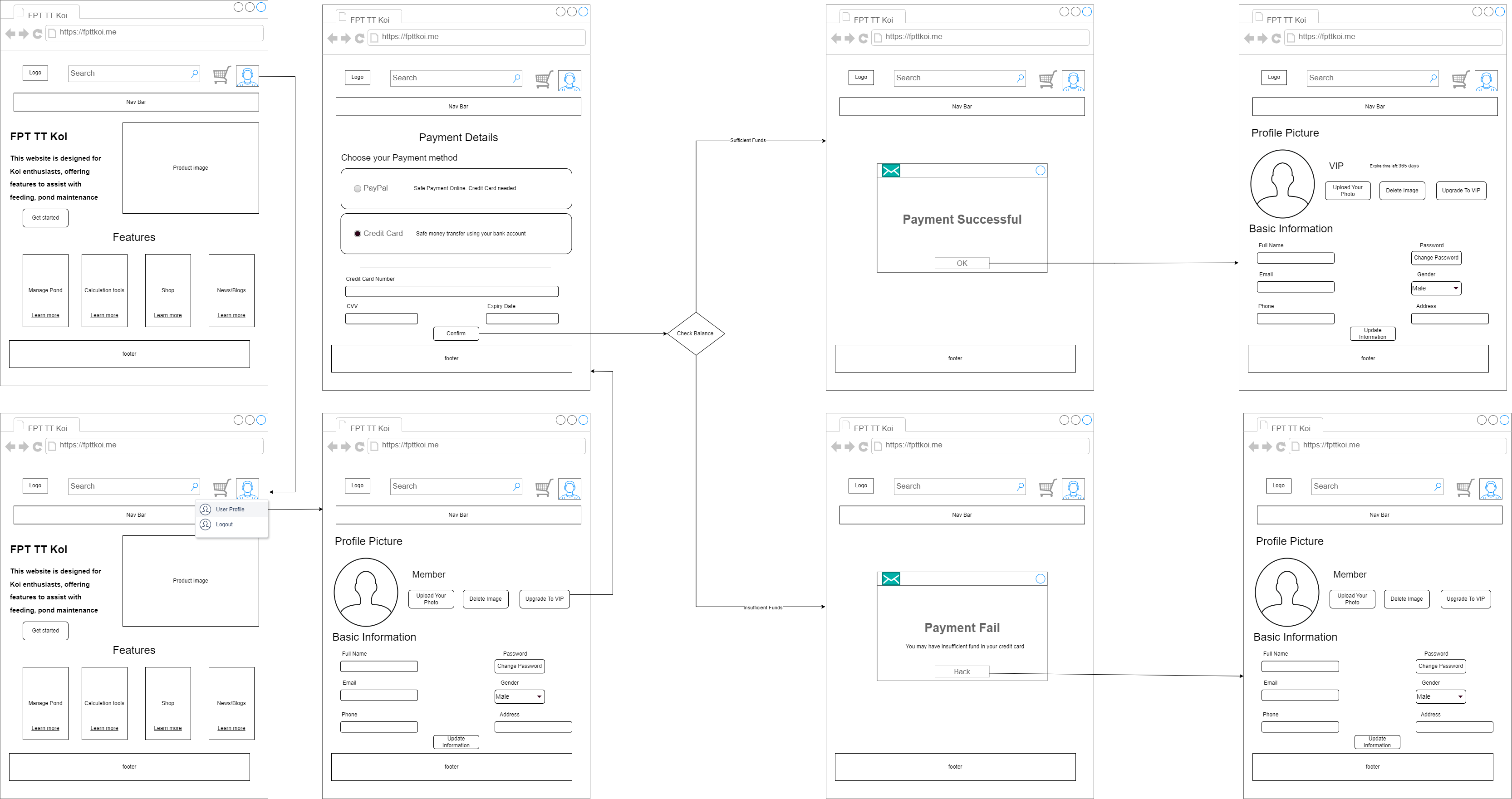
## State Diagrams



## Logical Data Model



## Wire Flows



# NON-FUNCTIONAL Requirements

## Usability

Ease of Use:

* The system must have an intuitive and user-friendly interface that is easy for both technical and non-technical users to navigate.
* All critical user actions (e.g., product checkout, order management) should require no more than 3 clicks.

## Reliability

System Availability:

* The system must be available 99.9% of the time, except for scheduled maintenance periods.

Fault Tolerance:

* The system must be able to recover from any server or network failure without data loss.
* Regular backups of user and product data must occur every 24 hours.

## Performance

System Performance:

* The system must handle a minimum of 100 concurrent users without performance degradation.
* The system response time for any CRUD operation (Create, Read, Update, Delete) must not exceed 2 seconds.

Scalability:

* The system should be able to scale both horizontally and vertically to support the growing number of users and products.

Database Performance:

* All database queries related to product and koi list searches must return results within 1 second.
* The system should optimize caching to minimize database hits for frequently accessed data (e.g., product lists, pond information).

## Security Requirements:

**Data Privacy**:

* Personal data (User Name, Password, Email, Phone, Address) must be encrypted using industry-standard encryption methods (e.g., AES-256).
* Passwords must be stored using salted hashing algorithms.

**Authentication and Authorization**:

* Only authenticated users (Members, Shops, Admins) must have access to the system.
* Role-based access control (RBAC) must be implemented to restrict access to sensitive features (e.g., Account Management, Revenue Management).