



Requirement & Design Specification

Koi Care System At Home

Version: 1.0

– Ho Chi Minh Campus, 2024 –

Record of Changes

Version	Date	A* M, D	In charge	Change Description
V1.0	8/9/2024	A	Nguyen Nhat Truong	Topic Selection and setup basic team
V1.0	8/9/2024	A	Nguyen Song Trong Thang	Wireframe
V1.0	8/9/2024	A	Nguyen Song Trong Thang, Le Viet Quoc Thinh	Git-setup
V1.0	8/9/2024	A	Nguyen Nhat Truong	Main scope description
V1.0	8/9/2024	A	Nguyen Nhat Truong	Requirement document
V1.0	8/9/2024	A	Hoang Quy Duong	Main screen flow
V1.0	8/9/2024	A	Le Viet Quoc Thinh	Database -(Variable Description)
V1.0	8/9/2024	A	Le Viet Quoc Thinh	ERD- Entity relationship diagram
V1.0	8/9/2024	A	Nguyen Song Trong Thang, Hoang Quy Duong, Nguyen Nhat Truong	DRD - Design requirements document +Figma
V1.0	13/9/2024	A	Nguyen Nhat Truong, Hoang Quy Duong, Nguyen Song Trong Thang, Tran Dai Nghia, Le Viet Quoc Thinh	SRS - Software requirement specification
V1.0	13/9/2024	A	Nguyen Nhat Truong	Class Diagram
V1.0	23/9/2024	A	Nguyen Nhat Truong	Business workflow and business rule
V1.0	23/9/2024	A	Hoang Quy Duong	Front-End Package design
V1.0	23/9/2024	A	Le Viet Quoc Thinh	Back-End package design
V1.0	13/9/2024	A	Nguyen Nhat Truong, Hoang Quy Duong, Nguyen Song Trong Thang, Tran Dai Nghia, Le Viet Quoc Thinh	Use case Specification
V1.0	18/9/2024	A	Nguyen Nhat Truong	Use case diagram
V1.0	2/10/2024	A	Nguyen Nhat Truong, Hoang Quy Duong, Nguyen Song Trong Thang, Tran Dai Nghia, Le Viet Quoc Thinh	Sequence diagram
V1.0	27/9/2024	A	Nguyen Nhat Truong, Hoang Quy Duong, Nguyen Song Trong Thang, Tran Dai Nghia, Le Viet Quoc Thinh	Activity Diagram
V1.0	6/10/2024	A	Nguyen Nhat Truong, Hoang Quy Duong, Nguyen Song	Function class Diagram

			Trong Thang, Tran Dai Nghia, Le Viet Quoc Thinh	
V1.0	20/9/2024	A	Tran Dai Nghia	Context Diagram
V1.0	20/9/2024	A	Le Viet Quoc Thinh	Database Design
V1.0	13/9/2024	A	Tran Dai Nghia	Software architecture
V1.0	2/10/2024	A	Nguyen Nhat Truong, Hoang Quy Duong, Nguyen Song Trong Thang, Tran Dai Nghia, Le Viet Quoc Thinh	Implementation

*A - Added M - Modified D - Deleted

Contents

Record of Changes	2
I. Overview	4
1. User Requirements	4
1.1 Actors	4
1.2 Use Cases	4
2. Overall Functionalities	5
2.1 Screens Flow	5
2.2 Screen Descriptions	5
2.3 Screen Authorization	5
2.4 Non-UI Functions	6
3. System High Level Design	6
3.1 Database Design	6
3.2 Code Packages	7
II. Requirement Specifications	8
1. Koi Fish	8
1.1 UC-1 _Add New Koi Fish	
1.2 UC-2 _Update Koi Fish	
1.3 UC-3 _View Koi Fish Record	
1.4 UC-4 _Remove Koi Fish	8
2. Manage Koi pond	11
2.1 UC-5 _Create Koi Pond Information	
2.2 UC-6 _Update Koi Pond Information	
2.3 UC-7 _Remove Koi Pond Information	
2.4 UC-8 _View Koi Pond Information	
2.5 UC-9 _Count Koi Fish in Pond	11
3. Manage Water quality	12
3.1 UC-10 _Add New Water Quality	12
3.2 UC-11 _Update Water Quality	
3.3 UC-12 _View Water Quality	
3.4 UC-13 _Remove Water Quality	
4. Shop Koi Fish Product	
4.1 UC-15 _Add to Cart Koi Fish Product	
4.2 UC-16 _View Product Details	
4.3 UC-17 _Search and View Product	

4.4 UC-18_View Product Reviews and Comments	
5. Manage Cart	
5.1 UC-17_View Cart	
5.2 UC-18_Pay for Cart	
5.3 UC-19_Remove Product from Cart	
5.4 UC-20_Add More Product	13
III. Design Specifications	14
1. <<Feature Name>	14
1.1 <<SubFeature Name>>	14
1.2 System Access	15
IV. Appendix	19
1. Assumptions & Dependencies	19
2. Limitations & Exclusions	19
3. Business Rules	19
4. ..	19

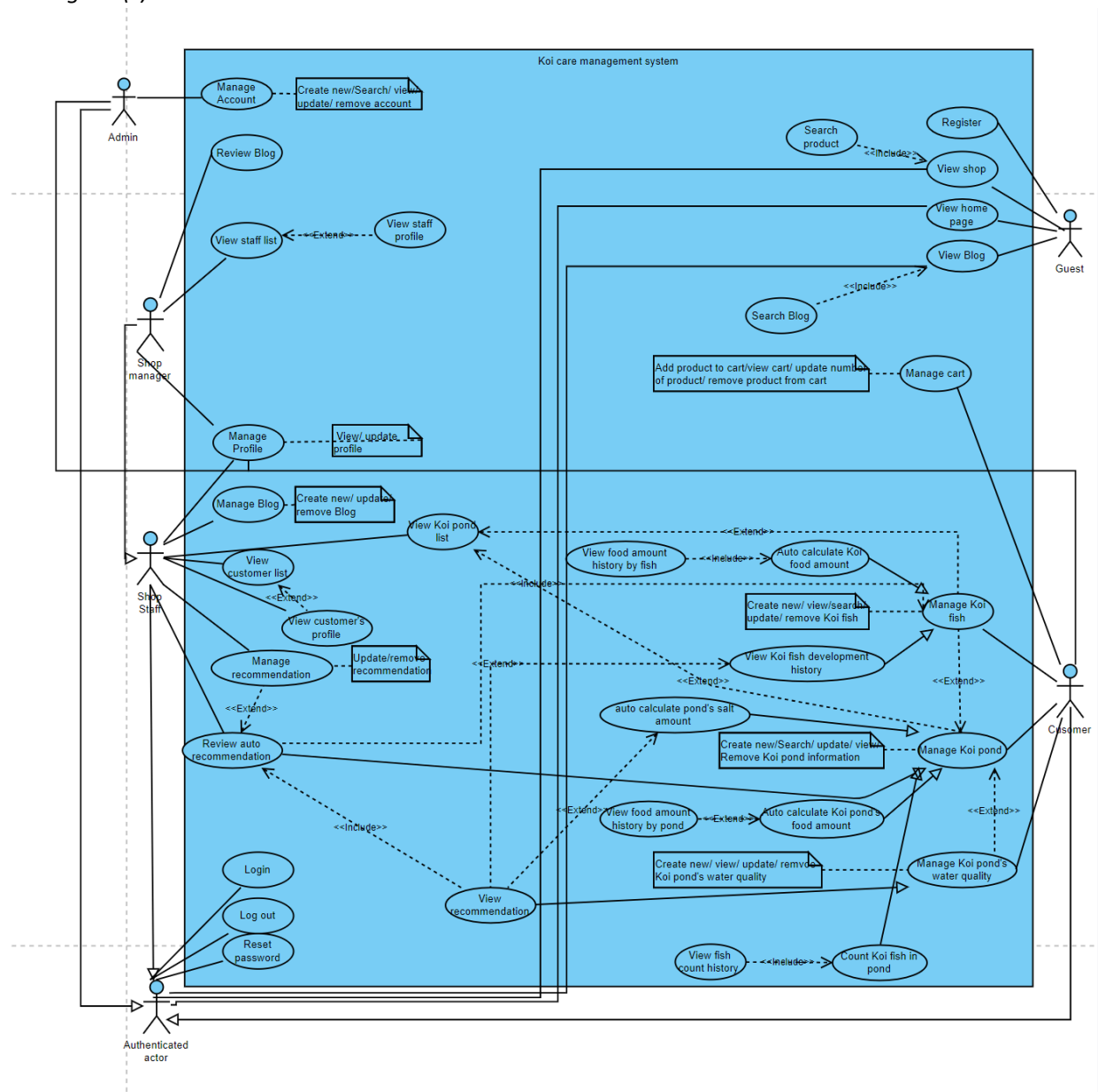
I. Overview

1. User Requirements

1.1 Actors

#	Actor	Description
1	Admin	An account holder has rights to manage other staff accounts and mission to keep the website working correctly.
2	Customer	Accounts that have been registered and authenticated in the system, the direct users of the features on the website
3	Shop manager	System administrator in terms of business, the person directly responsible for all aspects of the website shop.
4	Shop staff	This actor responsible for the main coordination of the website, basic management and use of system tasks and customer management
5	Guest	Individuals who want to use website for getting KOI Pond and KOI fish information but are restricted from using specific features of the system and lack a login account.

a. *Diagram(s)*



**Note*

manage sale promotion-> view
manage koi pond không extend lại
use case cần review lại

b. Descriptions

I D	Feature	Use Case	Use Case Description
--------	---------	----------	----------------------

01	Menu Operations	View home page	This use case allows actor to view main home page screen
02	Authentication	Login	This use case allows actor to log into system for using system's function
03	Authentication	Log-out	This use case allows actor to sign out of system
04	Authentication	Reset password	This use case allows actor to reset forgotten password
05	Manage Koi pond information	Create Koi pond	This use case allows actor create a new Koi pond to manage in system
06	Manage Koi pond information	Update Koi pond information	This use case allows actor to update managed Koi pond information
07	Manage Koi pond information	Remove Koi pond information	This use case allows actor to remove the current Koi pond's information
08	Manage Koi pond information	View Koi pond information	This use case allows actor to view the Koi pond information at a moment
09	Manage Koi pond information	Count Koi fish in pond	This use case allows actor to count the number of Koi fish in a pond
10	Manage Koi pond information	View count-history	This use case allows actor to view the history of Koi fish number in a pond
11	Manage Koi pond's water quality	Add new water quality	This use case allows actor to add new Koi pond's water quality data
12	Manage Koi pond's water quality	Update water quality	This use case allows actor to update information of Koi pond's water quality
13	Manage Koi pond's water quality	View water quality	This use case allows actor to view Koi pond's water quality
14	Manage Koi pond's water quality	Remove water quality	This use case allows actor to remove Koi lake water quality information
15	Manage Koi pond information	Calculate Koi food amount	This use case allows actor to calculate the amount of food for a Koi pond
16	Manage Koi pond information	View Koi food amount history	This use case allows actor to calculate the necessary food amount for Koi pond
17	Manage Koi pond information	Calculate salt amount	This use case allows actor to calculate the necessary salt amount for Koi pond

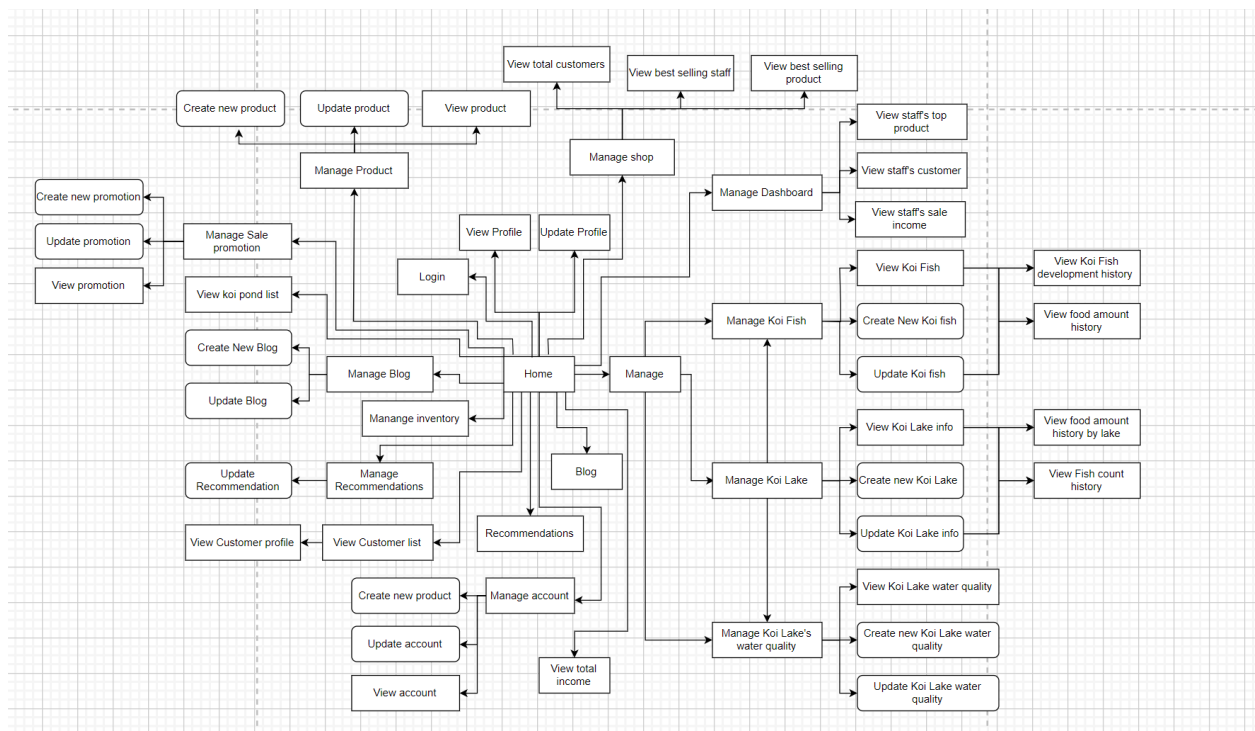
18	Manage Koi pond information	View salt history	This use case allows actor to view the history of Koi pond's salt quality
19	Manage Koi fish information	Create Koi Fish	This use case allows actor to create new Koi fish for storing data in system
20	Manage Koi fish information	Update Koi Fish	This use case allows actor to update new Koi fish information
21	Manage Koi fish information	View Koi fish development history	This use case allows actor to view the historic information of Koi fish
22	Manage Koi fish information	Remove Koi Fish	This use case allows actor to remove the Koi fish information from system
23	Manage Koi fish information	View Koi fish Information	This use case allows actor to view current Koi fish information
24	Manage Koi fish information	Calculate Koi food amount	This use case allows actor to calculate the Koi pond's necessary food amount
25	Manage Koi fish information	View Koi food amount history	This use case allows actor to view the Koi fish food amount history
26	Manage staff's customer information	View staff's customer list	This use case allows actor to view customers of a staff
27	Manage staff's customer information	View customer's Koi pond list	This use case allows actor to view the customer's Koi pond
28	Manage customer's information	View customer's profile	This use case allows actor to view customer profile
29	Manage account	Remove account	This use case allows actor to deactivate an account
30	Manage account	View account	This use case allows actor to view account information
31	Manage account	Update account	This use case allows actor to update an account data
32	Manage account	Create new account	This use case allows actor to create new account
33	Manage blog	Create blog	This use case allows actor to create new blog for website
34	Manage blog	Update blog	This use case allows actor to update blog's information
35	Manage blog	View blog	This use case allows actor to view blog

36	Manage blog	Remove blog	This use case allows actor to hide blog post from website
37	Manage cart	Add product to cart	This use case allows actor to add a product to cart
38	Manage cart	View cart	This use case allows actor to view cart information
39	Manage cart	remove product from cart	This use case allows actor to remove a product from cart
40	Manage cart	Update number of product	This use case allows actor to update number of product directly from cart
41	Manage Koi pond's water quality	View recommendation history	This use case allows actor to view the history of recommendation or suggestion to improve Koi pond's water quality
42	Manage Koi pond's information	Auto recommend for Koi pond	This use case allows actor to receive an auto recommendation for Koi pond base on pond's quality
43	Manage profile	Update profile	This use case allows actor to update actor's profile
44	Manage profile	View profile	This use case allows actor to view actor's profile
45	Manage Koi pond's information	View recommendation	This use case allows actor to view and track recommendation which auto generate from system
46	Manage Koi pond's information	Remove recommendation	This use case allows actor to remove system recommendation
47	Manage Koi pond's information	Update recommendation	This use case allows actor to update recommendation
48	Manage blog	Review Blog	This use case allows actor to evaluate, approve or reject to appear in website
49	Manage staff's information for manager	View staff's customer	This use case allows actor to view a staff's customer
50	Manage account	Search account	This use case allows actor to search an account
51	Menu operations	Search product	This use case allows actor to search a product in website
52	Menu operations	View shop	This use case allows actor to view shop in website for business
53	Manage blog	Search blog	This use case allows actor to search a blog

54	Manage Koi fish information	Search Koi fish	This use case allows actor to search a Koi fish
55	Manage Koi pond's information	Search Koi pond	This use case allows actor to search a Koi pond
56	Manage Koi pond's information	Search Koi pond	This use case allows actor to search a Koi pond

2. Overall Functionalities

2.1 Screens Flow



2.2 Screen Descriptions

#	Feature	Screen	Description
1	Manage Koi fish	View Koi fish	This screen allows customer to view koi fish
2	Manage Koi fish	Create Koi fish	This screen allows customer to create koi fish
3	Manage Koi fish	Update Koi fish	This screen allows customer to update koi fish
4	Manage Koi fish	View Koi fish development history	This screen allows customer to view koi fish development history

5	Manage Koi fish	View food amount history	This screen allows customer to view food amount history
6	Manage Koi lake	View Koi lake info	This screen allows customer to view koi lake info
7	Manage Koi lake	Create Koi lake	This screen allows customer to create koi lake
8	Manage Koi lake	Update Koi lake	This screen allows customer to update koi lake
9	Manage Koi lake	View food amount history by lake	This screen allows customer to view food amount history by lake
10	Manage Koi lake	View fish count history	This screen allows customer to view fish count history
11	Manage Koi lake's water quality	View Koi lake water quality	This screen allows customer to view koi lake water quality
12	Manage Koi lake's water quality	Create Koi lake's water quality	This screen allows customer to create koi lake water quality
13	Manage Koi lake's water quality	Update Koi lake water quality	This screen allows customer to update koi lake water quality
14	Manage Dashboard	View staff's top product	This screen allows shop staff to view staff's top product
15	Manage Dashboard	View staff's customer	This screen allows shop staff to view staff's customer
16	Manage Dashboard	View staff's sale income	This screen allows shop staff to view staff's sale income
17	Manage Shop	View total customers	This screen allows shop manager to view total customers
18	Manage Shop	View best selling staff	This screen allows shop manager to view best selling staff
19	Manage Shop	View best selling product	This screen allows shop manager to view best selling product
20	Manage Product	Create new product	This screen allows shop staff to create new product
21	Manage Product	Update product	This screen allows shop staff to update product
22	Manage Product	View product	This screen allows shop staff to view product
23	Manage sale promotion	Create new promotion	This screen allows shop staff to create new sale promotion

24	Manage sale promotion	Update promotion	This screen allows shop staff to update promotion
25	Manage sale promotion	View promotion	This screen allows shop staff to view promotion
26	Manage blog	Create new blog	This screen allows shop staff to create new blog
27	Manage blog	Update blog	This screen allows shop staff to update blog
28	Manage inventory	Check inventory	This screen allows shop staff to check inventory
29	Manage account	Create new account	This screen allows admin to create account
30	Manage account	Update account	This screen allows admin to update account
31	Manage account	View account	This screen allows admin to view account
32	Recommendations	Recommendations	This screen allows actor to view recommendations
33	View blog	Blog	This screens allows actor to view blog
34	View total income	View total income	This screen allows shop manager to view total income
35	Manage profile	View profile	This screen allows actor to view their profile
36	Manage profile	Update profile	This screen allows actor to update their profile
37	Login	Login	This screen allows unauthenticated actor to login
38	View koi pond list	View koi pond list	This screen allows shop staff to view koi pond list
39	View customer	View customer list	This screen allows shop staff to view customer list
40	View customer	View customer profile	This screen allows shop staff to view customer profile
41	Manage recommendations	Update recommendations	This screen allows shop staff to update recommendations

2.3 Screen Authorization

Screen	Admin	Customer	Shop Manager	Shop Staff	Guest
Manage Koi fish					
View Koi fish		X			
Create new Koi fish		X			
Update Koi fish		X			
Remove Koi fish		X			

View Koi fish development history		X			
View food amount history		X			
Manage Koi lake					
View Koi lake info		X			
Create new Koi lake		X			
Update new Koi lake		X			
Remove Koi lake		X			
View food amount history by lake		X			
View fish count history		X			
Manage Koi lake's water quality					
View Koi lake water quality		X			
Create Koi lake's water quality		X			
Update Koi lake water quality		X			
Remove Koi lake water quality		X			
Manage Dashboard					
View staff's top product				X	
View staff's customer				X	
View staff's sale income				X	
Manage Shop					
View total customers			X		
View best selling staff			X		
View best selling product			X		
Manage Product					
Create new product				X	
Update product				X	
View product				X	
Remove product				X	
Manage sale promotion					
Create new promotion				X	

Update promotion				X	
View promotion				X	
Remove promotion				X	
Manage blog					
Create new blog				X	
Update blog				X	
Remove blog				X	
Manage inventory					
Check inventory				X	
Manage account					
Create new account	X				
Update account	X				
View account	X				
Remove account	X				
Recommendations		X		X	
Manage recommendations					
Update recommendations				X	
Remove recommendations				X	
Blog					
View blog	X	X	X	X	X
Total income					
View total income			X		
Manage profile					
View profile	X	X	X	X	
Update profile	X	X	X	X	
Reset password	X	X	X	X	
Login	X	X	X	X	
View Koi pond list				X	
View customer					
View customer list				X	
View customer profile				X	

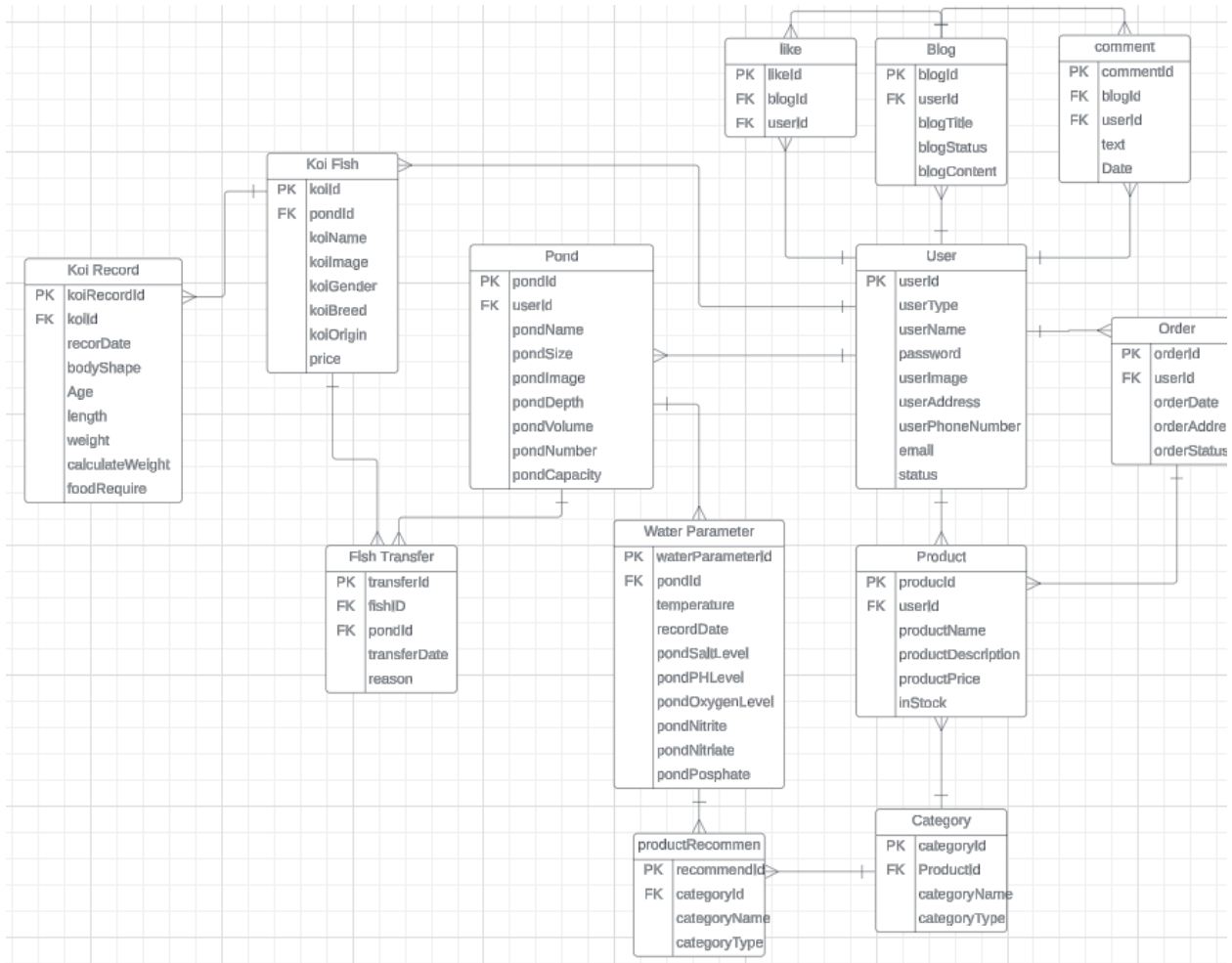
2.4 Non-UI Functions

#	Feature	System Function	Description
1	Reset password	Send email notification	System sends user an email to reset password
2			

3. System High Level Design

3.1 Database Design

a. Database Schema



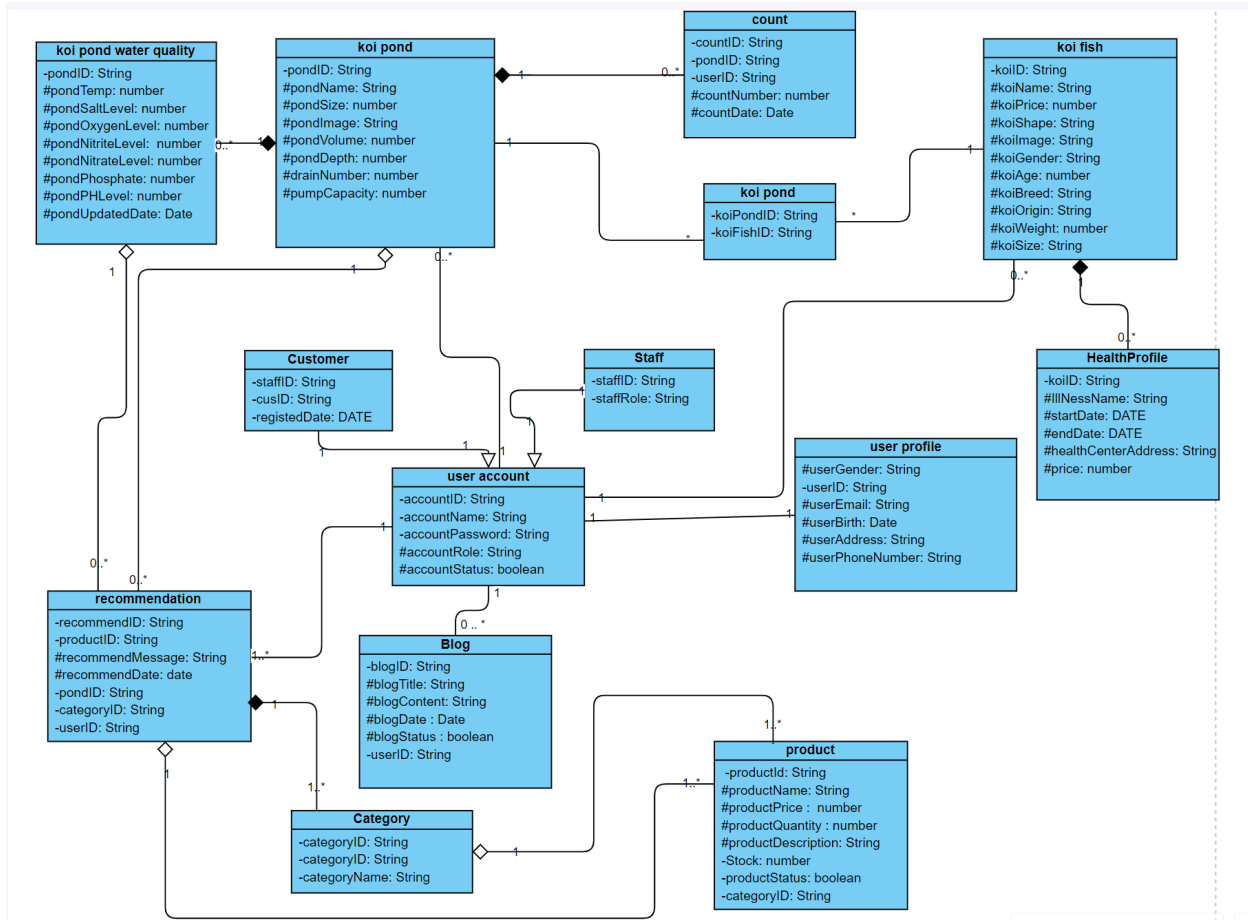
b. Table Descriptions

No	Table	Description
01	<i>user</i>	This table stores information about users of the system, such as admins, shop owners, staff, and customers. Each user has a unique username, email, and password. User types are admin, shop, staff, customer - Primary keys: <i>userId</i>
02	<i>pond</i>	This table stores details about each koi fish, including its name, gender, breed, origin, and price. It represents individual koi fish in the system, which may be transferred between ponds. - Primary keys: <i>pondId</i> - Foreign keys : <i>userId</i>

3	<i>water parameter</i>	<p>This table records the water parameters for each pond on a given date. Parameters include temperature, salt level, pH, oxygen levels, and the concentrations of NO₂, NO₃, and PO₄.</p> <ul style="list-style-type: none"> - Primary keys : waterParameterId - Foreign keys : pondId
4	<i>koi Fish</i>	<p>This table stores details about each koi fish, including its name, gender, breed, origin, and price. It represents individual koi fish in the system, which may be transferred between ponds.</p> <ul style="list-style-type: none"> -Primary keys : fishId -Foreign keys :pondId
5	<i>Koi Record</i>	<p>This table holds records for each koi fish, tracking their growth over time. Records include data like length, weight, body shape, age, calculated weight, and food requirements for the fish at a given date.</p> <ul style="list-style-type: none"> -Primary key : koiRecordId -Foreign key : koiId
6	<i>Fish Transfer</i>	<p>This table store transfer data of a fish when the fish is moved to a different pond with reason and transfer date</p> <ul style="list-style-type: none"> -Primary key : transferId -Foreign keys : koiId , pondId
7	<i>product</i>	<p>This table stores information about products available for sale, such as the name, description, price, category, and stock quantity. Each product is associated with a seller (user).</p> <ul style="list-style-type: none"> -Primary key : productId -Foreign key : userId
8	<i>Order</i>	<p>This table stores information about orders placed by users. It includes the order date, total amount, and status (e.g., pending, shipped). Each order is associated with a user who placed it.</p>

		-Primary key : orderId -Foreign key : userId
9	<i>Category</i>	This table help product have a different category for suggest product for customer - Primary key : categoryId -Foreign key : productId
10	<i>product recommendat ion</i>	This table stores product recommendations, which are triggered by specific conditions (like product popularity or a sale). Each recommendation is associated with a product. -Primary key : recommedId -Foreign key : categoryId
11	<i>Blog</i>	This table stores blog posts created by users. It includes the post's title, content, and publication date. Each post is associated with an author (a user). -Primary key : blogId -Foreign key : userId

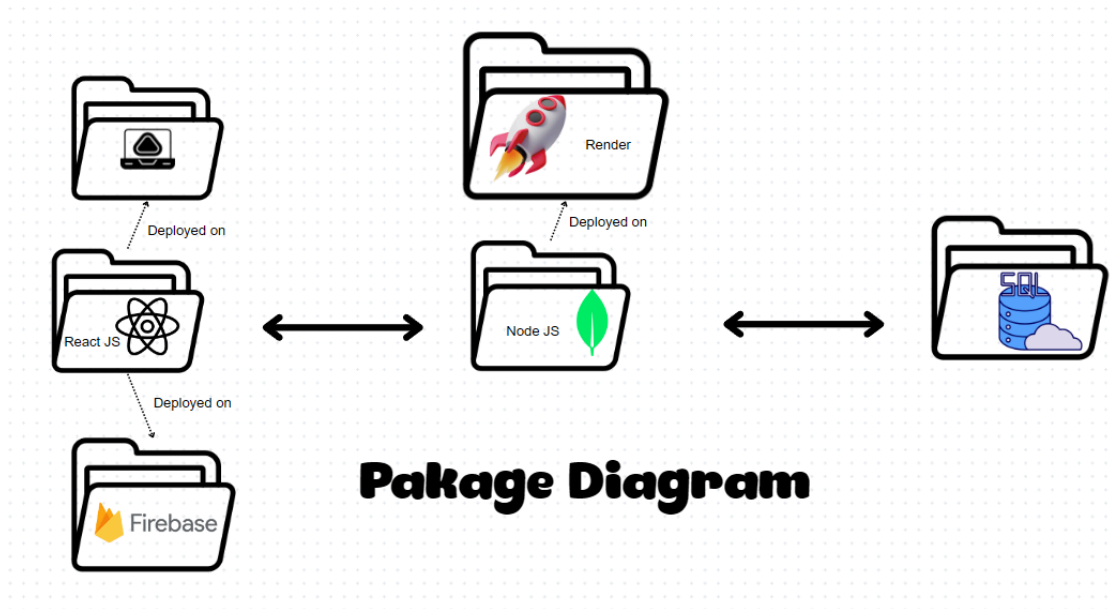
c. Class diagram



3.2 Code Packages

1 .FE:

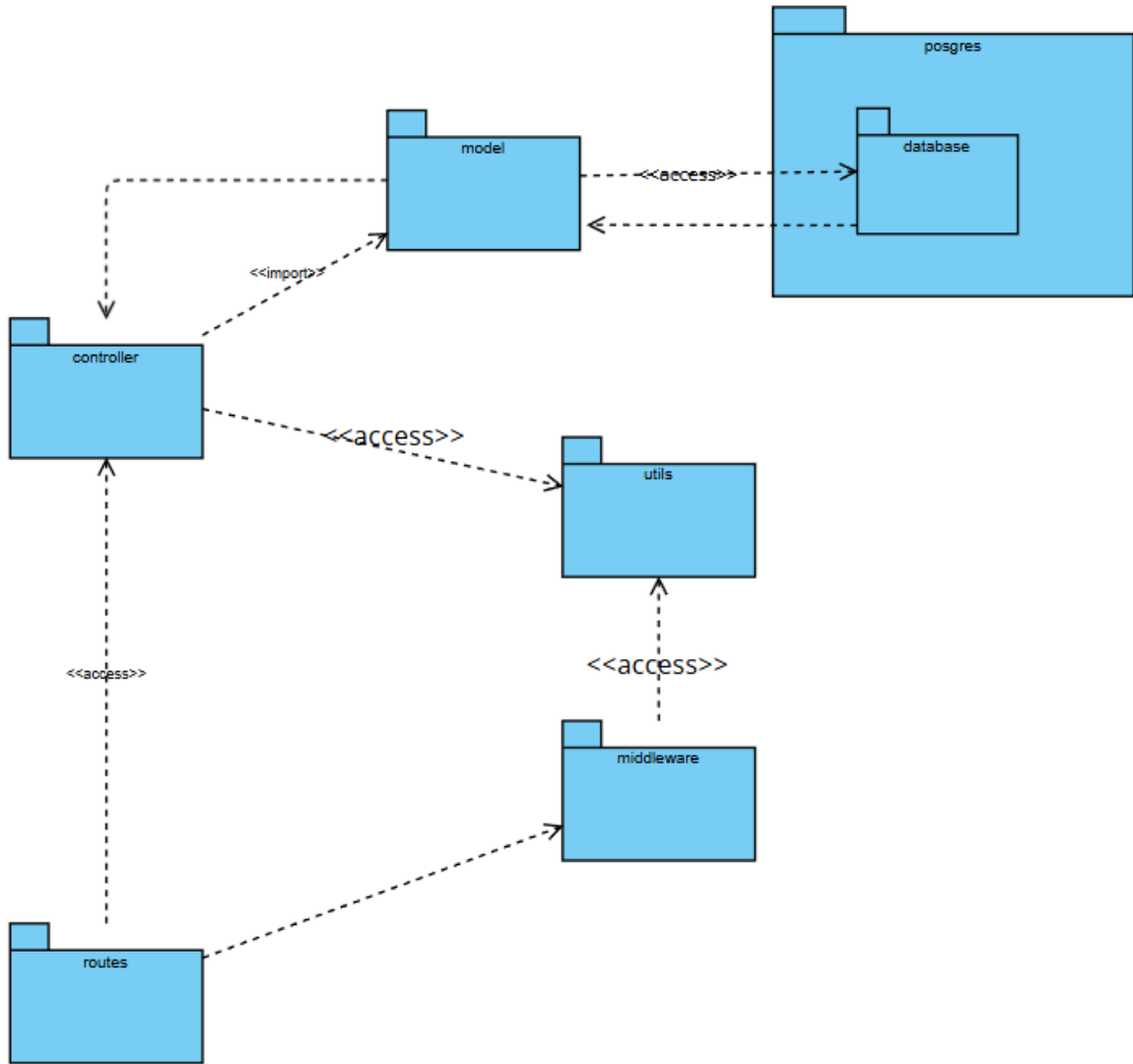
In this section, we will provide the package diagram for each sub-system of the web management system for Koi fish. This includes the overall package diagram, detailed explanations of each package, and the naming conventions used for packages and classes within each package.



Package descriptions

No	Package	Description
01	ReactJS	<i>This package is responsible for the frontend of the Koi Fish management system, handling user interface and interactions. It is deployed on Vercel.</i>
02	NodeJS	<i>This package manages the backend logic, API endpoints, and business logic, interacting with both the frontend and the database. Deployed on Render.</i>
03	PostgreSQL	<i>Acts as the primary database for the system, storing data about Koi Fish, users, and pond information.</i>
04	Firebase	<i>Provides authentication and real-time storage for user profiles and images related to Koi Fish. Integrates with the frontend.</i>
05	Vercel	<i>The deployment platform for the ReactJS package. It ensures the frontend is hosted and available to users.</i>
06	Render	<i>This platform deploys and hosts the NodeJS backend package. It manages server-side operations and interacts with the PostgreSQL database.</i>
07	...	

2 .BE



Package description

No	Package	Description
01	<i>routes</i>	handles all API request and routing to controller
02	<i>middleware</i>	Authentication User and authorization logic
03	<i>utils</i>	Swagger for API testing and handle error message
04	<i>controller</i>	Handle core business logic
05	<i>model</i>	Defines database schemas and relationship
06	<i>database</i>	Manages database connection and config

II. Requirement Specifications

1. Koi Fish

1.1 UC-1 _ Add New Koi Fish

a. Functionalities

Provide the functional description for the use cases using the template/guides below

Functional Description Template

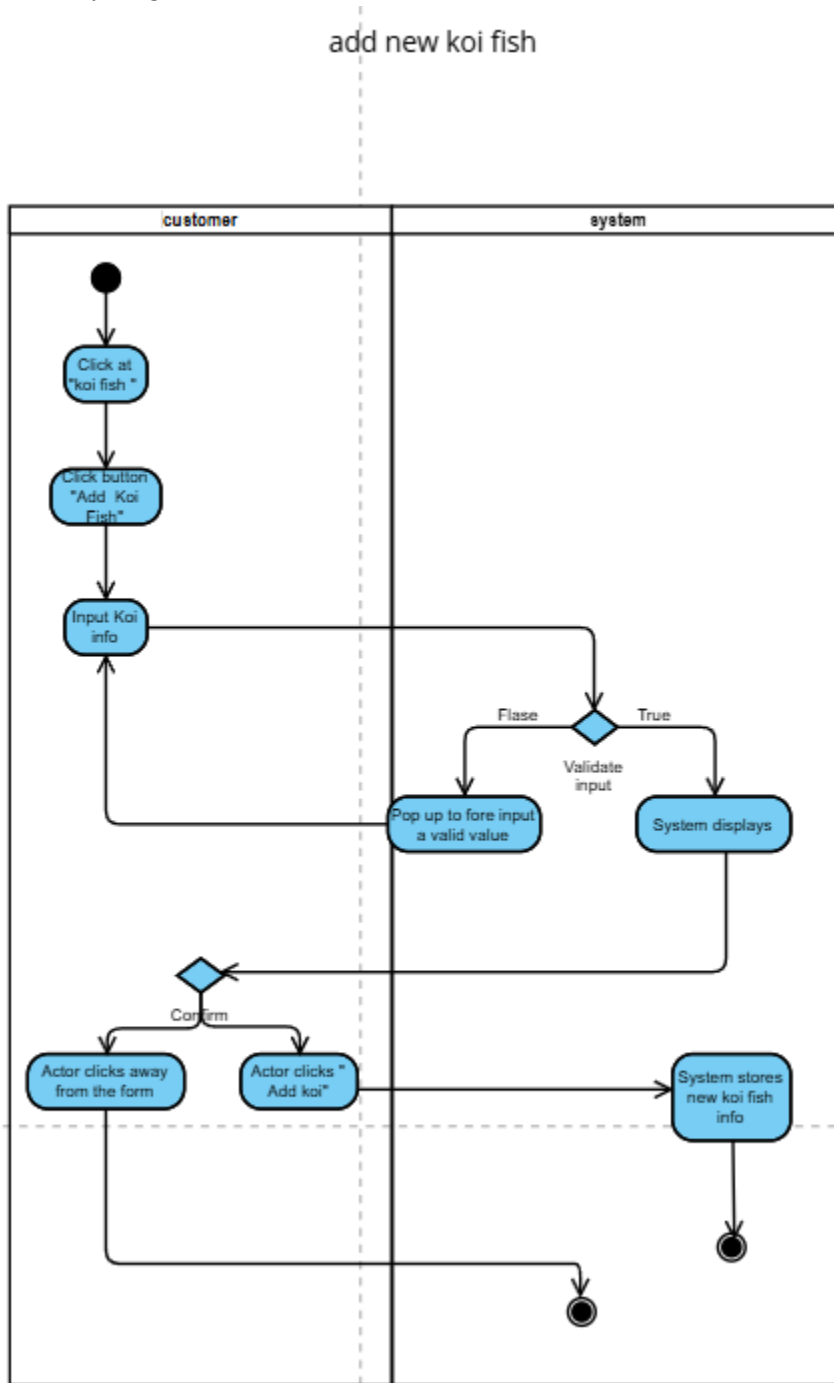
UC ID and Name:	UC-1 Add New Koi Fish		
Created By:	Le Viet Quoc Thinh	Date Created:	14/9
Primary Actor:	Customer	Secondary Actors:	N/A
Trigger:	Customer click add koi button		
Description:	This use case allows actor to add new koi fish to the pond		
Preconditions:	PRE-1. Actor must be authenticated PRE-2. Actor must have an existing pond		
Postconditions:	POST-1.The koi fish is added to the pond.		
Normal Flow:	<ol style="list-style-type: none">1. Actor enter "Manage Koi" Page2. Actor clicks "My Koi"3. Actor click "Add Koi Fish"4. The system show a detail table of fish data (Koi Name, image URL, Gender , Breed , Origin , Current Pond)5. Actor inputs the details.6. The system validates and stores the information.7. A confirmation message is shown.		
Alternative Flows:	Actor has no ponds, the system prompts them to create one before adding a koi fish.		
Exceptions:	The system fails to validate koi data due to missing or incorrect information.		
Priority:	High, Must Have		
Frequency of Use:	frequently, when a customer want to add new fish		
Business Rules:	Only members with an account can add koi fish. A member must have at least one pond to add a koi fish.		
Other Information:	N/A		
Assumptions:	The customer has permissions to manage ponds and koi fish.		

b. Business Rules

Provide the business rules those are applied only to the use case

ID	Business Rule	Business Rule Description

c. Activity diagram



1.2 UC002_Update Koi Fish

a. Functionalities

Provide the functional description for the use cases using the template/guides below

Functional Description Template

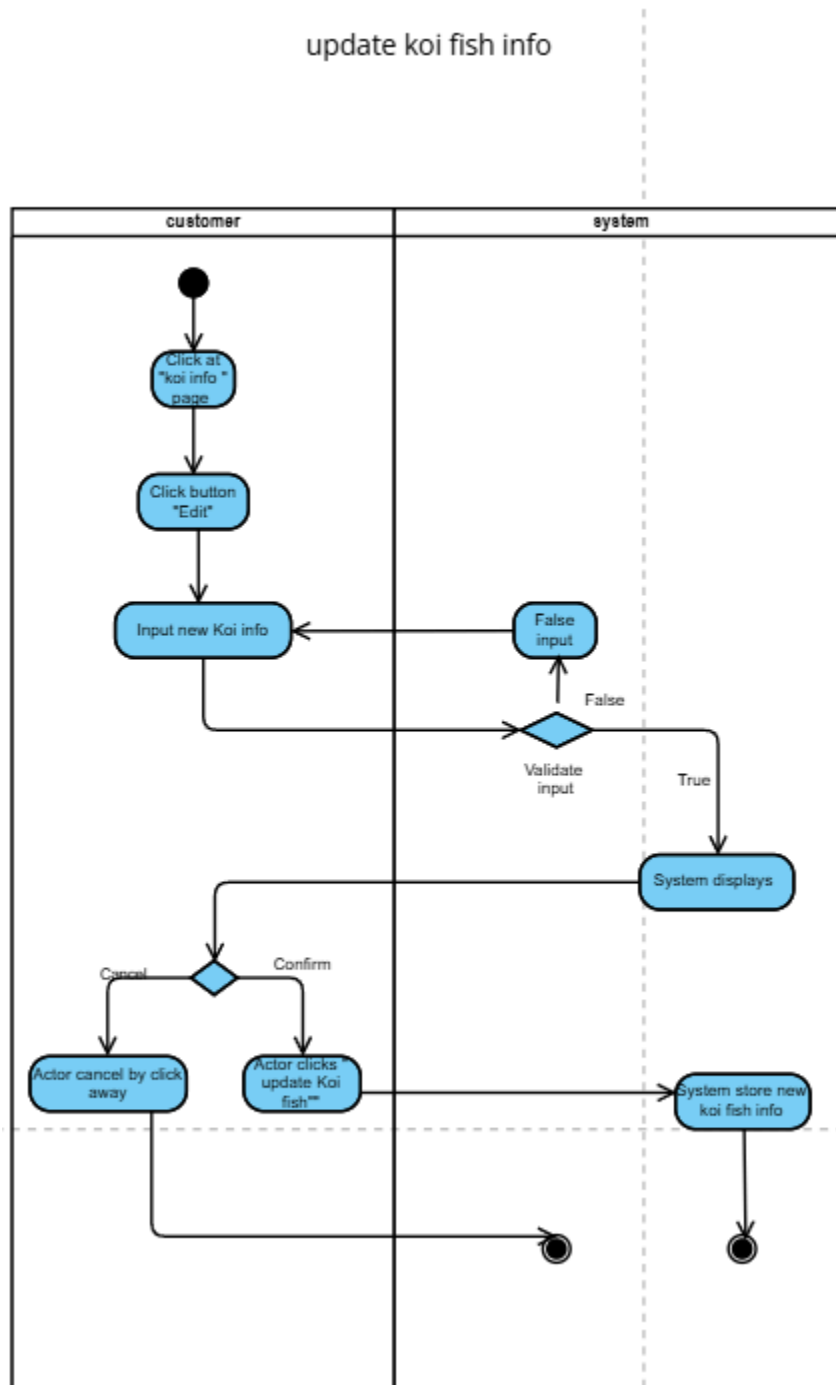
UC ID and Name:	UC-02		
Created By:	Le Viet Quoc Thinh	Date Created:	15/9
Primary Actor:	Customer	Secondary Actors:	N/A
Trigger:	Actor clicks update details of a koi fish		
Description:	This use case allows the actor to update a koi fish in the pond.		
Preconditions:	The koi fish must already exist in the system.		
Postconditions:	The updated details are saved in the system. Koi development trends are updated if applicable.		
Normal Flow:	<ol style="list-style-type: none">1. Actor selects a koi fish from their list in "koi info" page.2. Actor chooses the "Edit" button.3. The system displays the current koi details.4. Actor updates the details (koi Name, image URL, Gender).5. The system validates and saves the updated information.		
Alternative Flows:	N/A		
Exceptions:	The system fails to save updates due to validation errors.		
Priority:	High, Must Have		
Frequency of Use:	Regularly.		
Business Rules:	Only the owner of the pond can edit koi fish details.		
Other Information:	N/A		
Assumptions:	N/A		

b. Business Rules

Provide the business rules those are applied only to the use case

ID	Business Rule	Business Rule Description

c. Activity diagram



1.3 UC03 - View Koi Fish Record

a. Functionalities

Provide the functional description for the use cases using the template/guides below

Functional Description Template

UC ID and Name:	UC03 - View Koi Fish Development Trends
-----------------	---

Created By:	Le Viet Quoc Thinh	Date Created:	14/9
Primary Actor:	Customer	Secondary Actors:	N/A
Trigger:	Actor clicks koi fish's growth trends.		
Description:	This use case allows actor to view koi fish's growth		
Preconditions:	PRE-1.The koi fish must have sufficient recorded updates over time.		
Postconditions:	POST-1.Actor is shown a chart of koi development trends.		
Normal Flow:	<ol style="list-style-type: none"> 1. Actor clicks the "Manage Koi" Page. 2. Actor click "My Koi" 3. Actors navigate to Koi records and view the development of Koi Fish. 4. Actor reviews the chart. 		
Alternative Flows:	N/A		
Exceptions:	The system fails to retrieve data due to a server issue.		
Priority:	Medium		
Frequency of Use:	Occasionally, when members want to track koi record.		
Business Rules:	Data points must be available to generate growth trends. Trends are based on size, weight, and other development metrics.		
Other Information:	N/A		
Assumptions:	N/A		

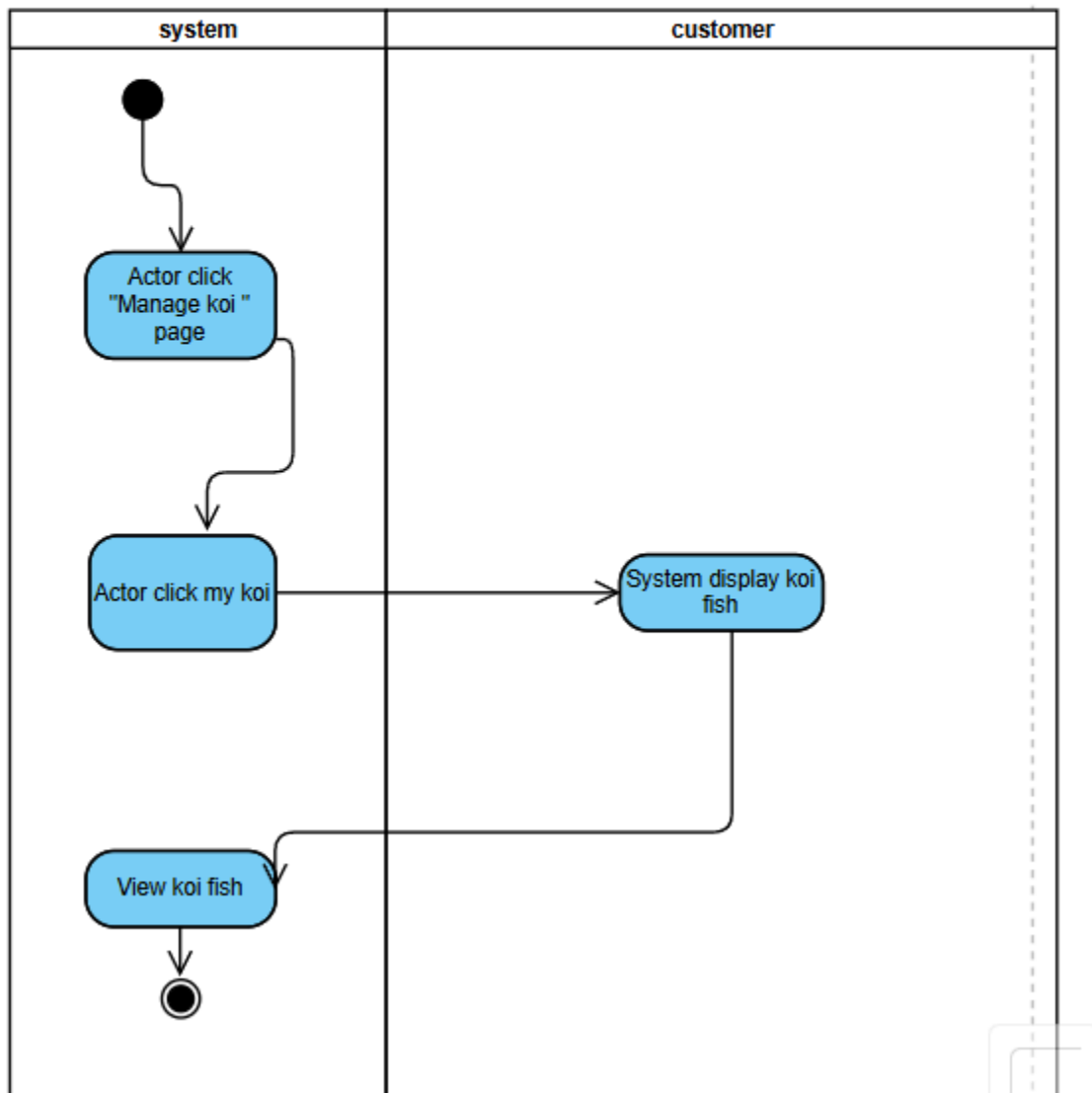
b. Business Rules

Provide the business rules those are applied only to the use case

ID	Business Rule	Business Rule Description
FR1	Password Encodin	

c. Activity diagram

View Koi fish



1.4 UC004_Remove a Koi Fish

a. Functionalities

Provide the functional description for the use cases using the template/guides below

Functional Description Template

UC ID and Name:	UC04 - Remove a Koi Fish		
Created By:	Le Viet Quoc Thinh	Date Created:	15/9
Primary Actor:	Customer	Secondary Actors:	N/A
Trigger:	The actor clicks remove a koi fish button from the system.		
Description:	This use case allows the actor to delete a koi fish in the pond.		
Preconditions:	PRE-1 The koi fish must exist in the system.		
Postconditions:	The koi fish is deleted from the system.		
Normal Flow:	<ol style="list-style-type: none"> 1. Actor click the "Manage Koi" Page. 2. Actor clicks the "My koi" button. 3. Actor click "delete " icon. 4. The system asks for confirmation. 5. Actor confirms deletion. 6. The system deletes the koi fish and associated records. 		
Alternative Flows:	Actor cancels deletion, the koi fish remains in the system.		
Exceptions:	<ol style="list-style-type: none"> 1. The system fails to delete the koi fish due to a server error. 		
Priority:	High, Must Have		
Frequency of Use:	Not Often		
Business Rules:	Deletion of a koi fish is irreversible. Only the owner of the koi fish can delete it.		
Other Information:	N/A		
Assumptions:	N/A		

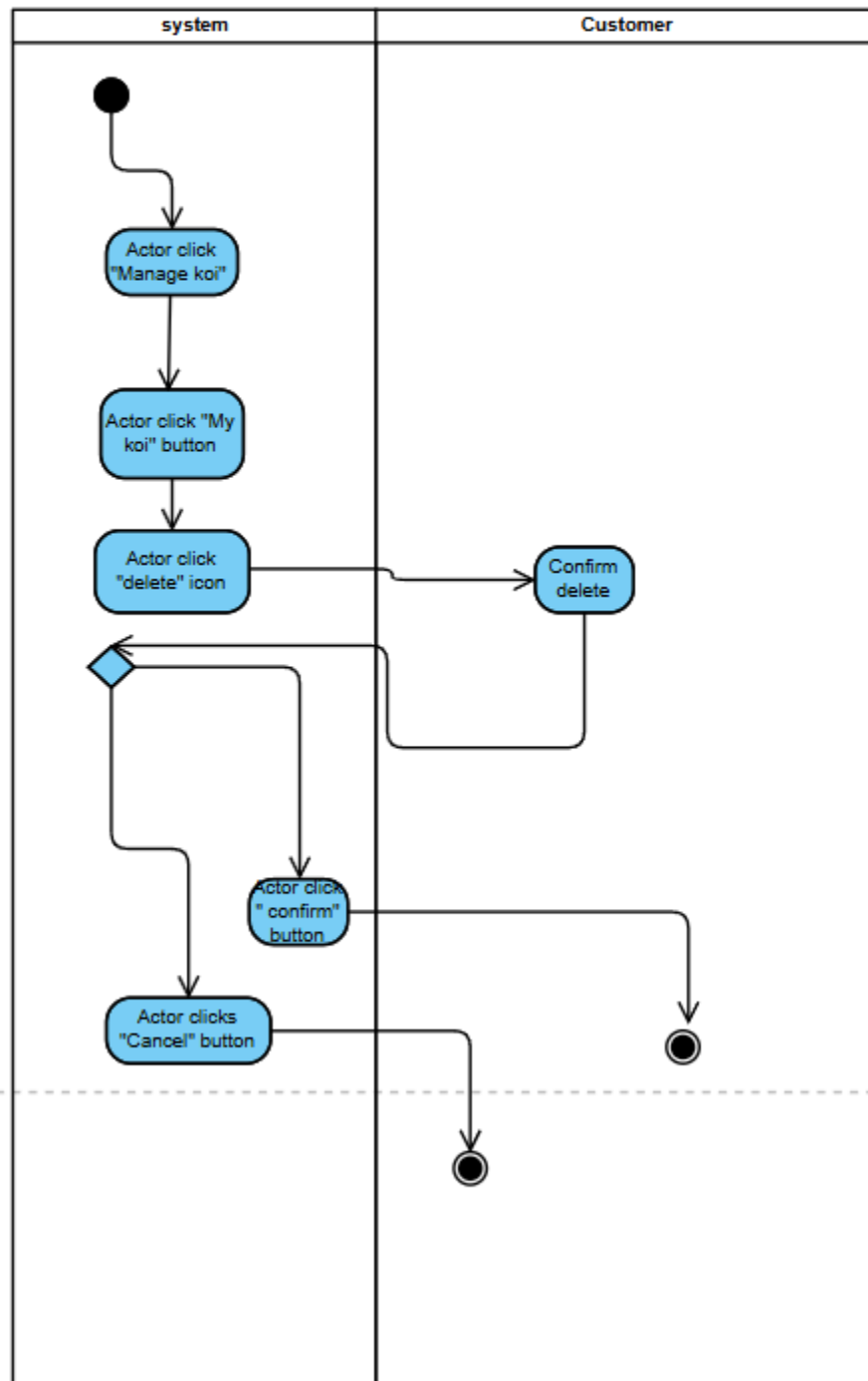
b. Business Rules

Provide the business rules those are applied only to the use case

ID	Business Rule	Business Rule Description
FR1	Password Encoding	User's password must be encoded with MD5 hashing

c. Activity diagram

remove a koi fish



2. Manage Koi pond

2.1 UC-06 Update Koi pond information

a. Functional Description

ID and Name:	UC-06 Update Koi pond information		
Created By:	Nguyen Nhat Truong	Date Created:	16/09/2024
Primary Actor:	Customer	Secondary Actors:	N/A
Description:	This use case allows actor to update the Koi pond information to store in system		
Trigger:	An actor want to update Koi pond information		
Preconditions:	PRE-1. Actor must be authenticated		
Postconditions:	POST-1. Koi pond's information must be updated and stored in database		
Normal Flow:	Update Koi pond information <ol style="list-style-type: none">1. Actor enters the "Manage Koi" page2. Actor clicks the "My Pond" card.3. Actor clicks button "Edit" icon.4. The system displays all fields of Koi pond for actor to update information include:<ul style="list-style-type: none">+ Pond Name: Required, text input 1- 50 characters+ Pond Image URL: Required, text input+ Pond Length: Required, real number input+ Pond Width: Required, real number input+ Pond Size (m2): Required, real number input+ Depth (m): Required, real number input+ Pond Drains: Required, real number input+ Pond Aeration Capacity (m3/hour): Required, real number input+ Pond Capacity of Koi Fish: Required, real number input5. Actor fills these information and clicks button "Save changes" [Alternative 2]6. System validates input information [Alternative 1]		

	7. System notifies message “Pond updated successfully!”[Exception 1], system stores new Koi pond’s information to database.
Alternative Flows:	<ol style="list-style-type: none"> Actor fills invalid information about Koi pond <ul style="list-style-type: none"> 1.1 System informs error message “Invalid input values.” [Normal flow 4] Actor clicks “Revert” button <ul style="list-style-type: none"> 2.1 System does not save any information of Koi pond and reverts to the previous version.
Exceptions:	<p>System break-time</p> <ul style="list-style-type: none"> System does not store any new information and keeps last version of Koi pond information System displays message “Failed to update pond.”
Priority:	High
Frequency of Use:	N/A.
Business Rules:	BR-01, BR-02, BR-03, BR-04
Other Information:	N/A
Assumptions:	N/A

b. Business Rules

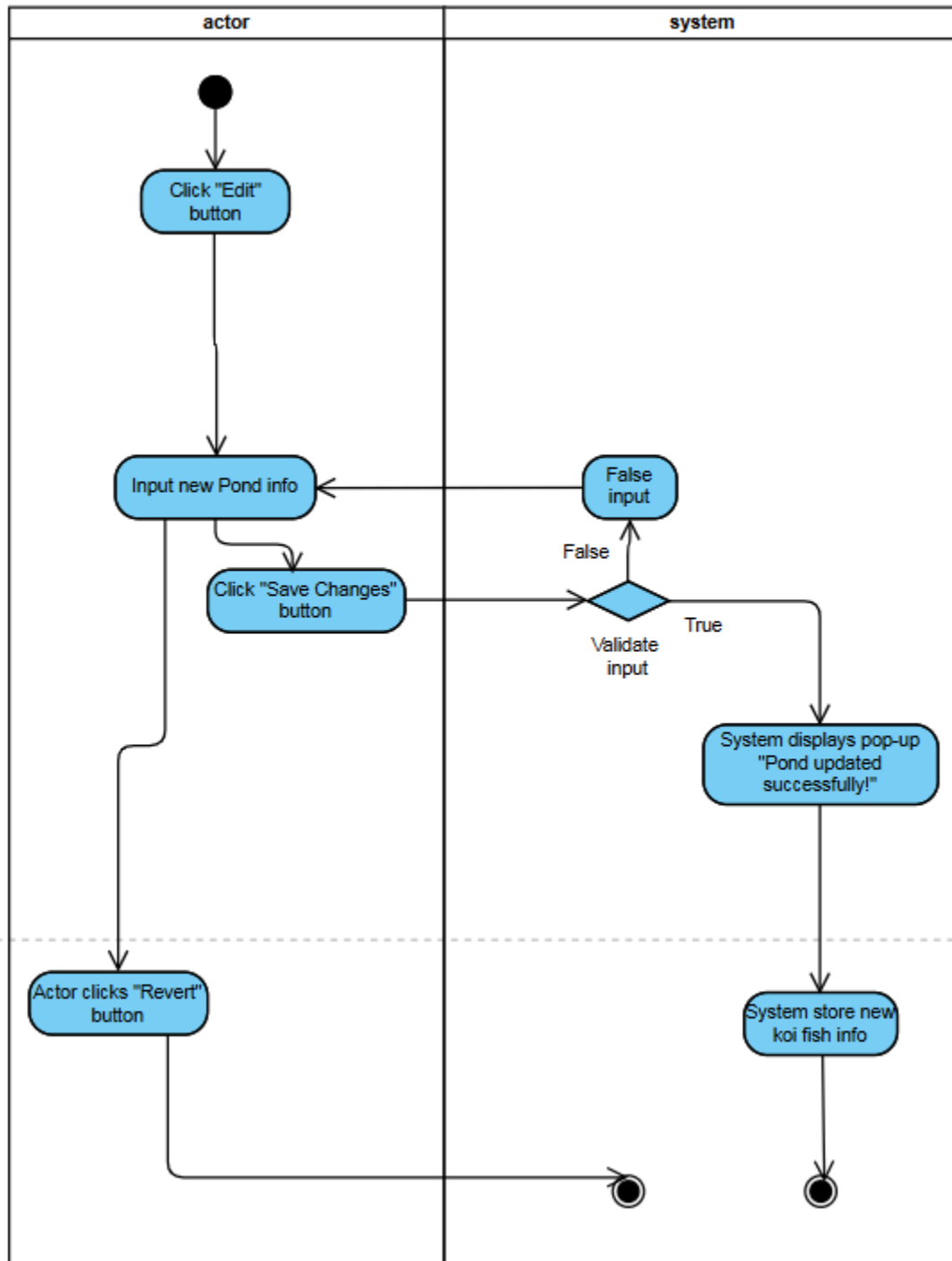
BR-01: When updating, if system is down-time, the system keeps the pre-version of information and does not store any new updated information of Koi pond.

BR-02: System must verify all information again before storing new updated information to database.

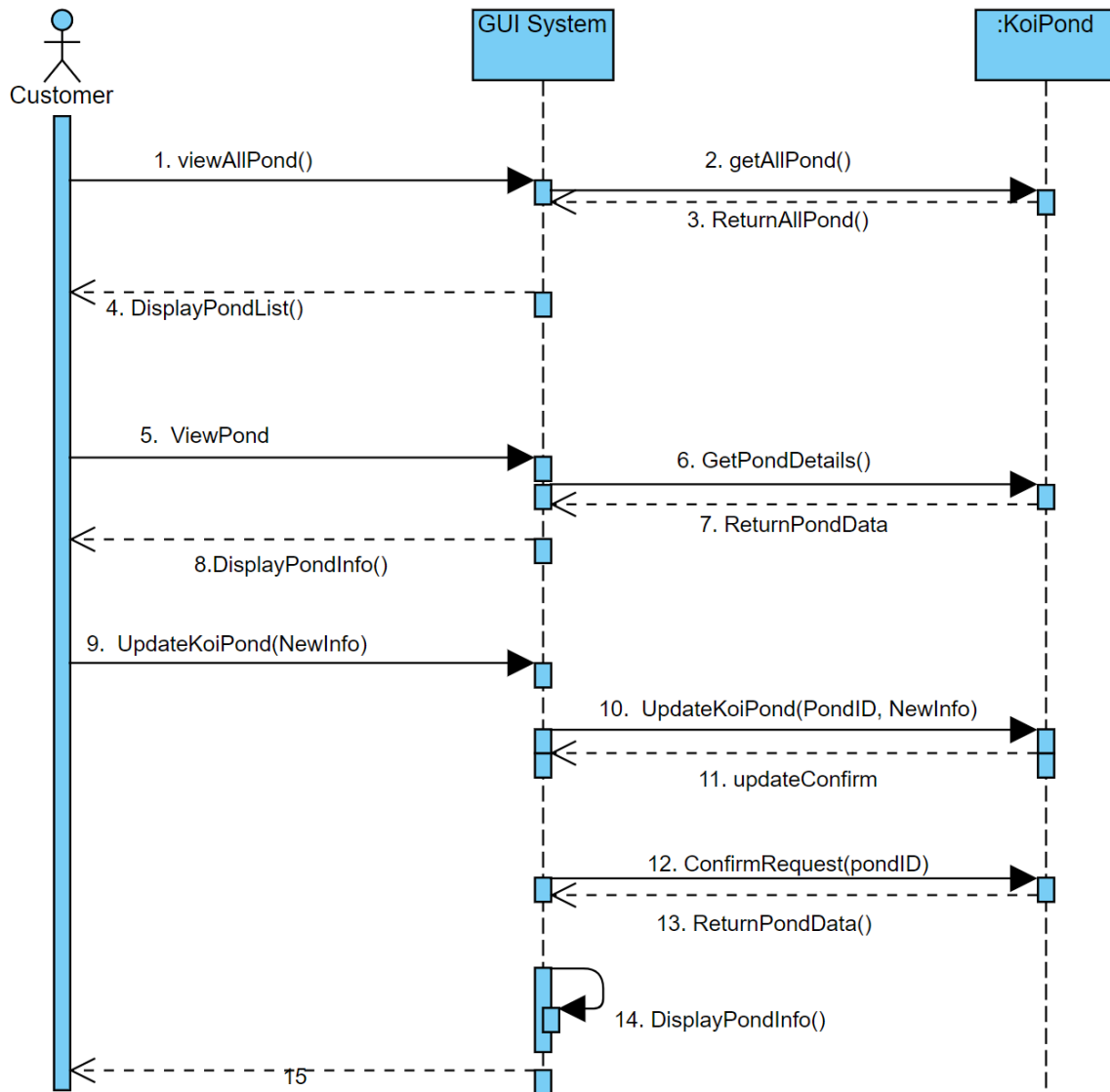
BR-03: Actor must be authenticated

BR-04: Only the owner actor has the right to edit.

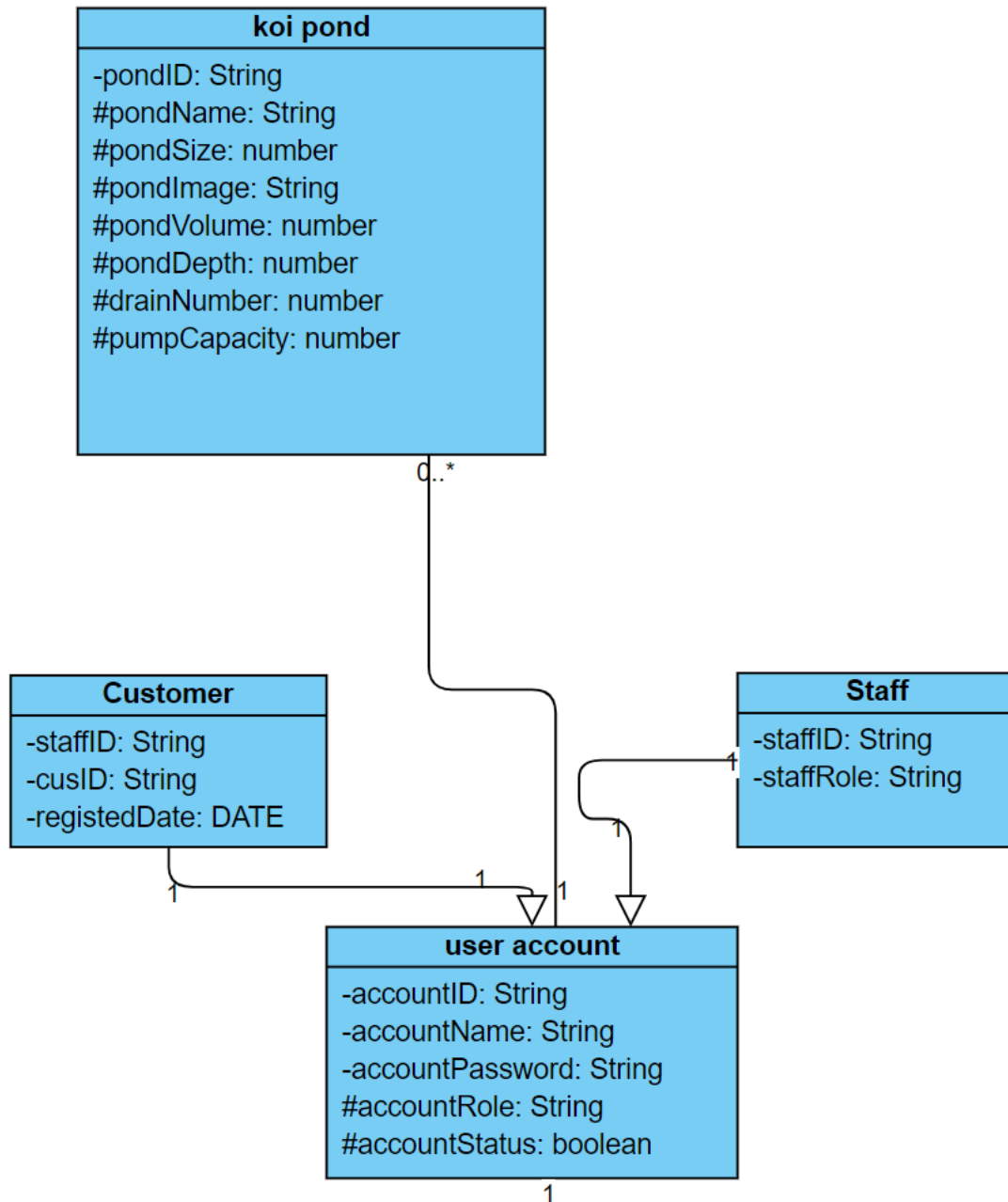
c. Activity diagram



d. Sequence Diagram



e. function class diagram



2.2 UC-05 Create Koi pond information

a. Functional Description

ID and Name:	UC-05 Create Koi pond information		
Created By:	Nguyen Nhat Truong	Date Created:	16/09/2024
Primary Actor:	Customer	Secondary Actors:	N/A

Description:	This use case allows actor to input the Koi pond information for the first time to store in system
Trigger:	An actor wants to create new Koi pond
Preconditions:	PRE-1. Actor must be authenticated
Postconditions:	POST-1. Koi pond's information must be stored as first version in database
Normal Flow:	Create Koi pond information <ol style="list-style-type: none"> Actor navigates to the "Manage Koi" page. Actor clicks "My Ponds" card. Actor clicks the "Create new pond" card. The system displays all fields of Koi pond for actor to fill in information include: <ul style="list-style-type: none"> Pond Name: Required, text input 1- 50 characters Pond Image URL: Required, text input Pond Length: Required, real number input Pond Width: Required, real number input Pond Size (m2): Required, real number input Depth (m): Required, real number input Pond Drains: Required, real number input Pond Aeration Capacity (m3/hour): Required, real number input Pond Capacity of Koi Fish: Required, real number input Actor fills these information and clicks button "Submit" System validates input information [Alternative 1] [Alternative 2] System auto generates new id for Koi pond [Exception 1] System notifies message "Pond Created Successfully!", System stores new Koi pond's information to database.
Alternative Flows:	<ol style="list-style-type: none"> Actor fills invalid information about Koi pond <ul style="list-style-type: none"> 1.1 System informs error message "Invalid input values." [Normal flow 4] Actor clicks "Cancel" button <ul style="list-style-type: none"> 2.1 System does not save any information of Koi pond.
Exceptions:	<ol style="list-style-type: none"> Pond ID auto generate failed , system notifies "Failed to create pond. Please try again." <ul style="list-style-type: none"> System notifies message "Failed to create pond. Please try again." System does not store any new information about Koi pond to the database.
Priority:	High
Frequency of Use:	N/A.
Business Rules:	BR-01, BR-02, BR-03, BR-06, BR-04

b. Business Rules

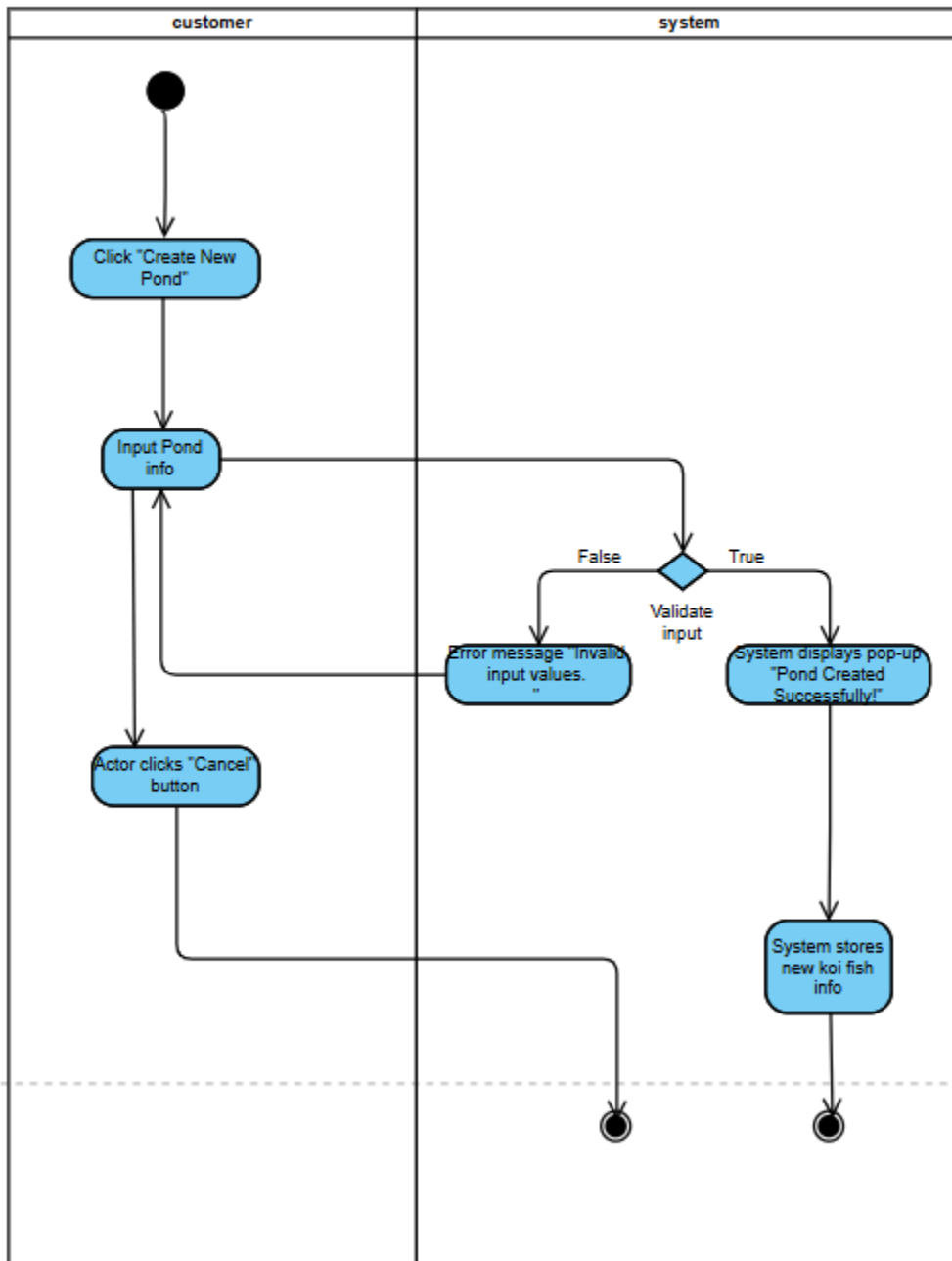
BR-03: Actor must be authenticated

BR-04: Only the owner actor has the right to edit

BR-06: Pond ID is distinguished and can not be duplicated.

c. Activity diagram

Create New Pond



d. Sequence Diagram

2.3 UC-08 View Koi pond information

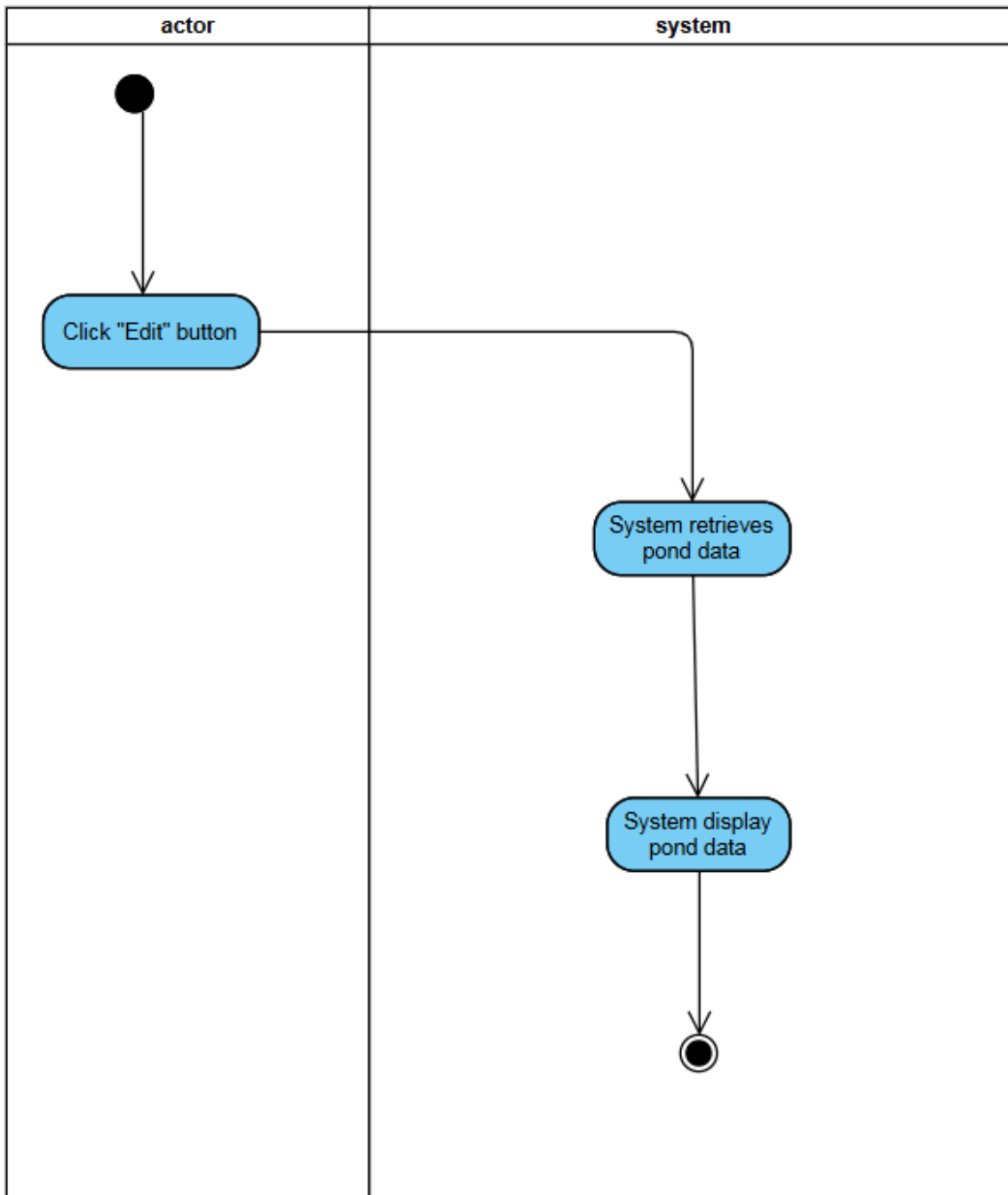
a. Functional Description

ID and Name:	UC-08 View Koi pond information		
Created By:	Nguyen Nhat Truong	Date Created:	16/09/2024
Primary Actor:	Customer	Secondary Actors:	Staff, Manager
Description:	This use case allows actor to view the Koi pond information		
Trigger:	An actor wants to view Koi pond information		
Preconditions:	PRE-1. Actor must be authenticated PRE-2. Koi pond must be created before and be stored in database		
Postconditions:	N/A		
Normal Flow:	View Koi pond information <ol style="list-style-type: none"> Actor navigates to the “Manage Koi” page Actor clicks the “My Pond” card. Actor clicks the “Edit” icon. [Exception 1,2]The system displays all fields of Koi pond for actor to view information include: <ol style="list-style-type: none"> Pond Name Pond Image URL Pond Size (m2) Pond Depth (m) Pond Volume (m3) Pond Drains Pond Aeration Capacity (m3/hour) Pond Capacity of Koi Fish Pond Slots <p>and Water Parameter, Fish List sections below</p>		
Alternative Flows:	N/A		
Exceptions:	<ol style="list-style-type: none"> System break-time <ul style="list-style-type: none"> System displays message “Please try again later!” System does not shows any data about Koi pond to actor The system failed to retrieve data, system displays “Failed to load pond data.”. 		
Priority:	Normal		
Frequency of Use:	N/A.		
Business Rules:	BR-03		

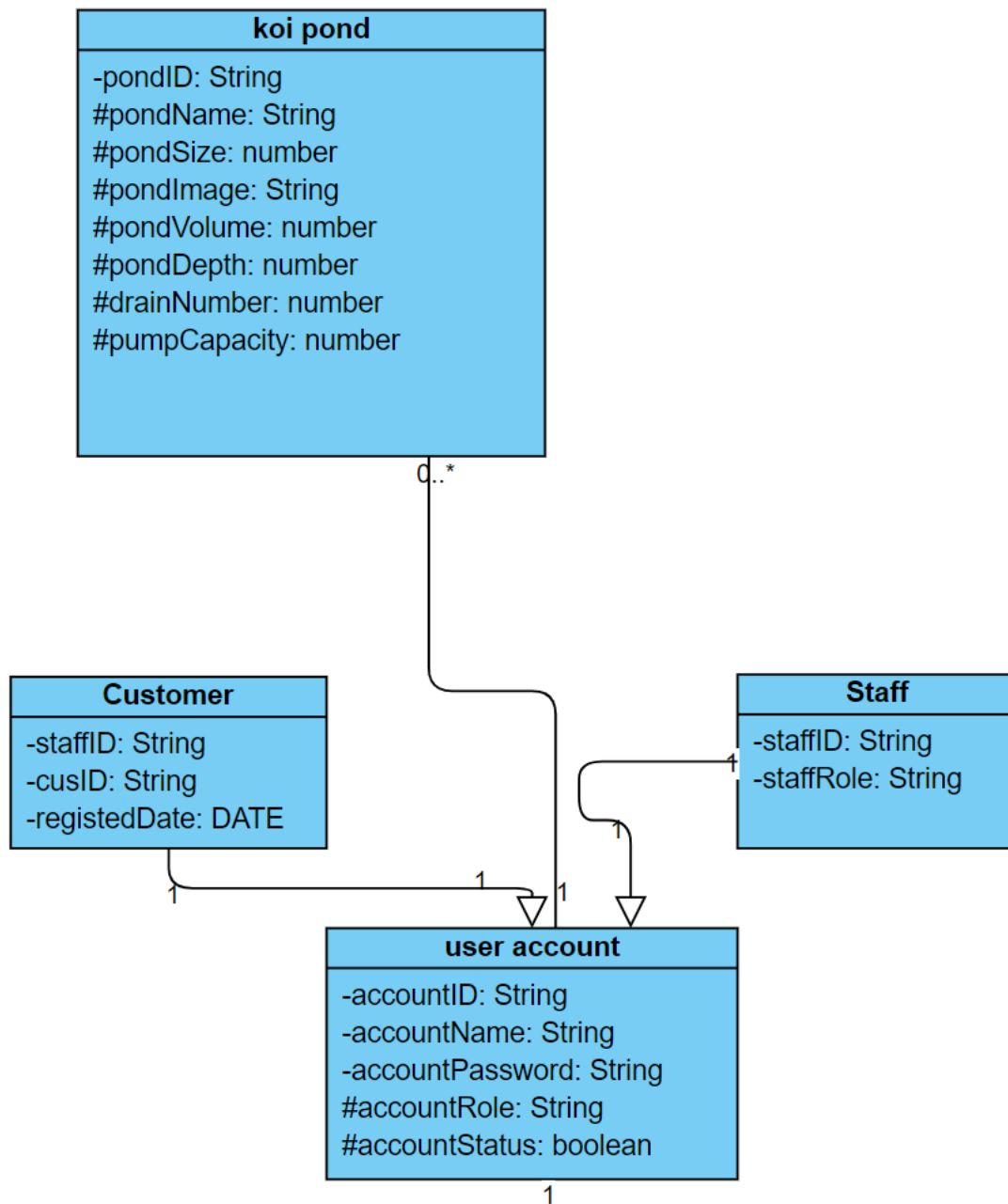
b. Business Rules:

BR-03: Actor must be authenticated

c. Activity diagram



e. Function class diagram



2.4 UC-09_Count Koi fish in pond

a. Functional Description

ID and Name:	UC-09 Count Koi fish in pond		
Created By:	Nguyen Nhat Truong	Date Created:	16/09/2024

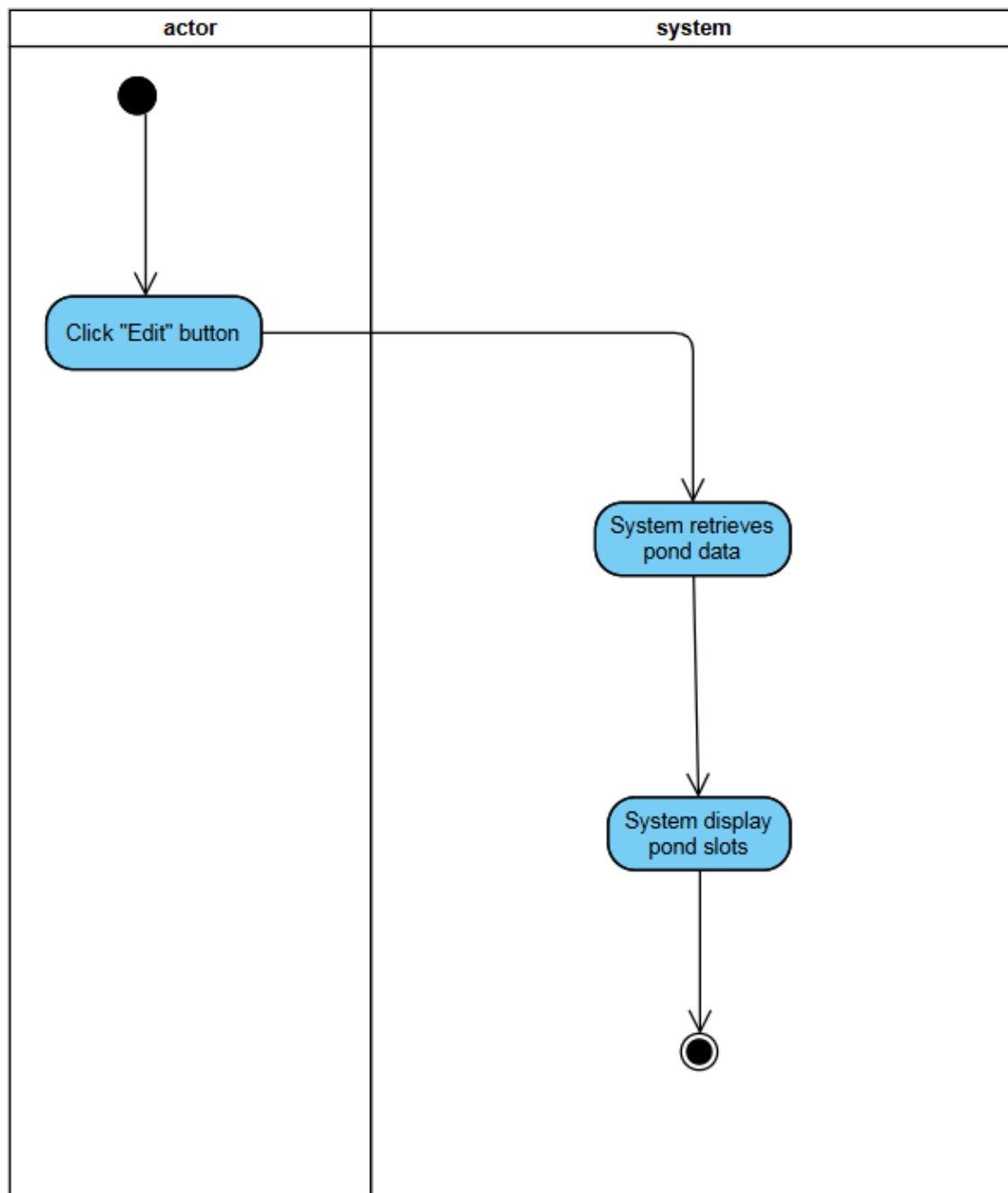
Primary Actor:	Customer	Secondary Actors:	N/A
Description:	This use case allows actor to count the number of Koi fish in a pond		
Trigger:	An actor wants to count number of Koi fish		
Preconditions:	PRE-1. Actor must be authenticated PRE-2. Koi pond must be created before and exists in the database.		
Postconditions:	POST-1. Koi fish count data must be stored in a database. POST-2. Count data must be displayed in Koi pond count history.		
Normal Flow:	<p>Count Koi fish in pond</p> <ol style="list-style-type: none"> 1. Actor navigates to the "Manage Koi" page 2. Actor clicks the "My Pond" card. 3. Actor clicks the "Edit" icon. 4. The system displays all fields of Koi pond and a field "Pond Slots" [Exception 1, Exception 2] 		

Alternative Flows:	N/A
Exceptions:	<ol style="list-style-type: none"> 1. System break-time <ul style="list-style-type: none"> - System displays message "Please try again later!" - System does not shows any data about Koi pond to actor 2. The system failed to retrieve data, system displays "Failed to load pond data.".
Priority:	High
Frequency of Use:	N/A.
Business Rules:	BR-03

b. Business Rules

BR-03: Actor must be authenticated.

c. Activity diagram



III. Design Specifications

3. Manage Water Quality

3.1 UC10_Add new water quality

a. Functionalities

Provide the functional description for the use cases using the template/guides below

Functional Description Template

UC ID and Name:	UC10_Add new water quality		
Created By:	Hoàng Quý Dương	Date Created:	15/9
Primary Actor:	Customer	Secondary Actors:	N/A
Trigger:	The actor clicks on the “+” button.		
Description:	This use case allows the actor to provide the water’s information to the system so it can evaluate the water’s quality.		
Preconditions:	The actor must have an account in the system. The actor must have created at least one pond.		
Postconditions:	The water information was provided successfully. The system saves the water information.		
Normal Flow:	<ol style="list-style-type: none"> Actor navigates to the “Manage Koi” page Actor clicks the “My Pond” card. Actor clicks the “Edit” icon in pond list. The actor scrolls down to “Water Parameter” section. The actor clicks the “+” icon. The system displays all the fields about water data including: <ul style="list-style-type: none"> - Temperature: required, number input - Salt level: required, number input - pH level : required, number input - Pond oxygen level: required, number input - NO2: required, number input - NO3: required, number input - PO4: required, number input The actor types in the details and clicks the “Submit” button. The system validates [Alternative 1][Exception 2] and shows a message “Parameter created successfully!”, [Exception 1] [Exception 3] then stores the information to the database. 		
Alternative Flows:	<ol style="list-style-type: none"> The actor types in invalid information: <ul style="list-style-type: none"> - System displays the correct type of input at the invalid input field. [Normal flow 5] 		
Exceptions:	<ol style="list-style-type: none"> The system fails to validate water data due to missing information and displays an error message “Please input [Field name]!”. The system fails to validate water data due to incorrect information and displays an error message. System break-time 		

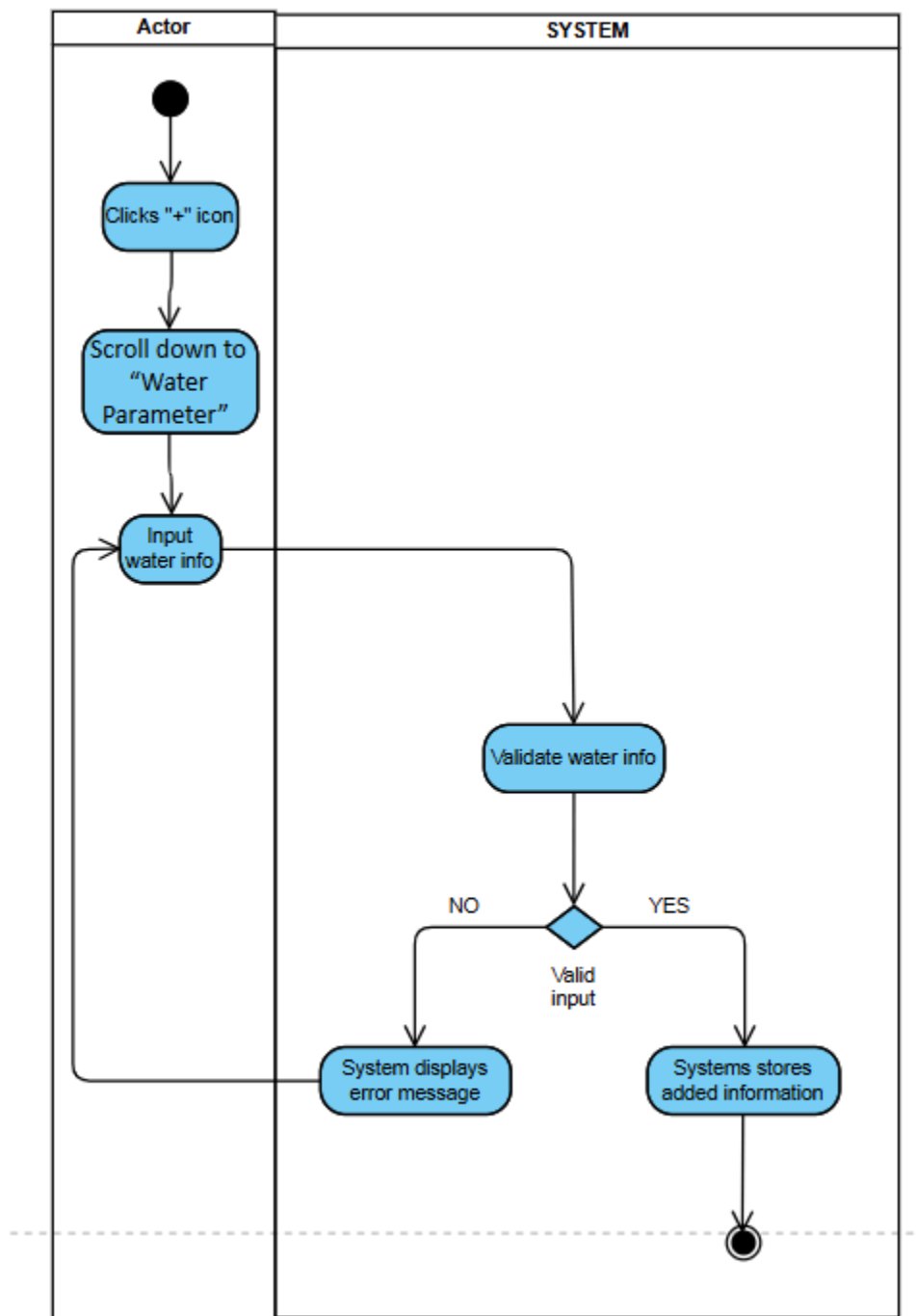
	<ul style="list-style-type: none"> - System does not store any information. - System displays message "Failed to add parameter!"
Priority:	High
Frequency of Use:	N/A
Business Rules:	BR-03, BR-04
Other Information:	N/A
Assumptions:	The actor has permission to manage ponds and koi fish.

b. Business Rules

BR-03: Actor must be authenticated.

BR-04: Only the owner actor has the right to edit

c. Activity diagram



3.2 UC11_Update water quality

a. Functionalities

Provide the functional description for the use cases using the template/guides below

Functional Description Template

UC ID and Name:	UC11_Update water quality
-----------------	---------------------------

Created By:	Hoàng Quý Dương	Date Created:	15/9
Primary Actor:	Customer	Secondary Actors:	N/A
Trigger:	The actor clicks on the “Update water” icon.		
Description:	This use case allows the actor to update the water’s information to the system so it can evaluate the water’s quality.		
Preconditions:	The water quality information has already existed in the system.		
Postconditions:	The water information was updated successfully. The system saves the water information.		
Normal Flow:	<ol style="list-style-type: none"> Actor navigates to the “Manage Koi” page Actor clicks the “My Pond” card. Actor clicks the “Edit” icon in pond list. The actor scrolls down to “Water Parameter” section. The actor clicks the “Edit” icon. The system displays all the fields about water data including: <ul style="list-style-type: none"> Temperature: required, number input Salt level: required, number input pH level : required, number input Pond oxygen level: required, number input NO2: required, number input NO3: required, number input PO4: required, number input The actor types in the details and clicks the “Submit” button. The system validates information [Alternative 1][Exception 1] [Alternative 2] The system shows a message “Updated parameter successfully” [Exception 2] and stores the information to the database. 		
Alternative Flows:	<ol style="list-style-type: none"> The actor types in invalid information: <ul style="list-style-type: none"> System informs error message “Please input [Field name]!”. [Normal flow 4]. Actor clicks “Cancel” button <ul style="list-style-type: none"> System does not save any updated water quality information. 		
Exceptions:	<ol style="list-style-type: none"> The system fails to validate water data due to incorrect information and displays an error message. System break-time <ul style="list-style-type: none"> System does not store any new information and keeps the last version of water quality information. System displays message “Failed to update parameter.” 		
Priority:	High		
Frequency of Use:	N/A		
Business Rules:	BR-02, BR-03, BR-04		
Other Information:	N/A		
Assumptions:	The system has sufficient validation to ensure accurate updates.		

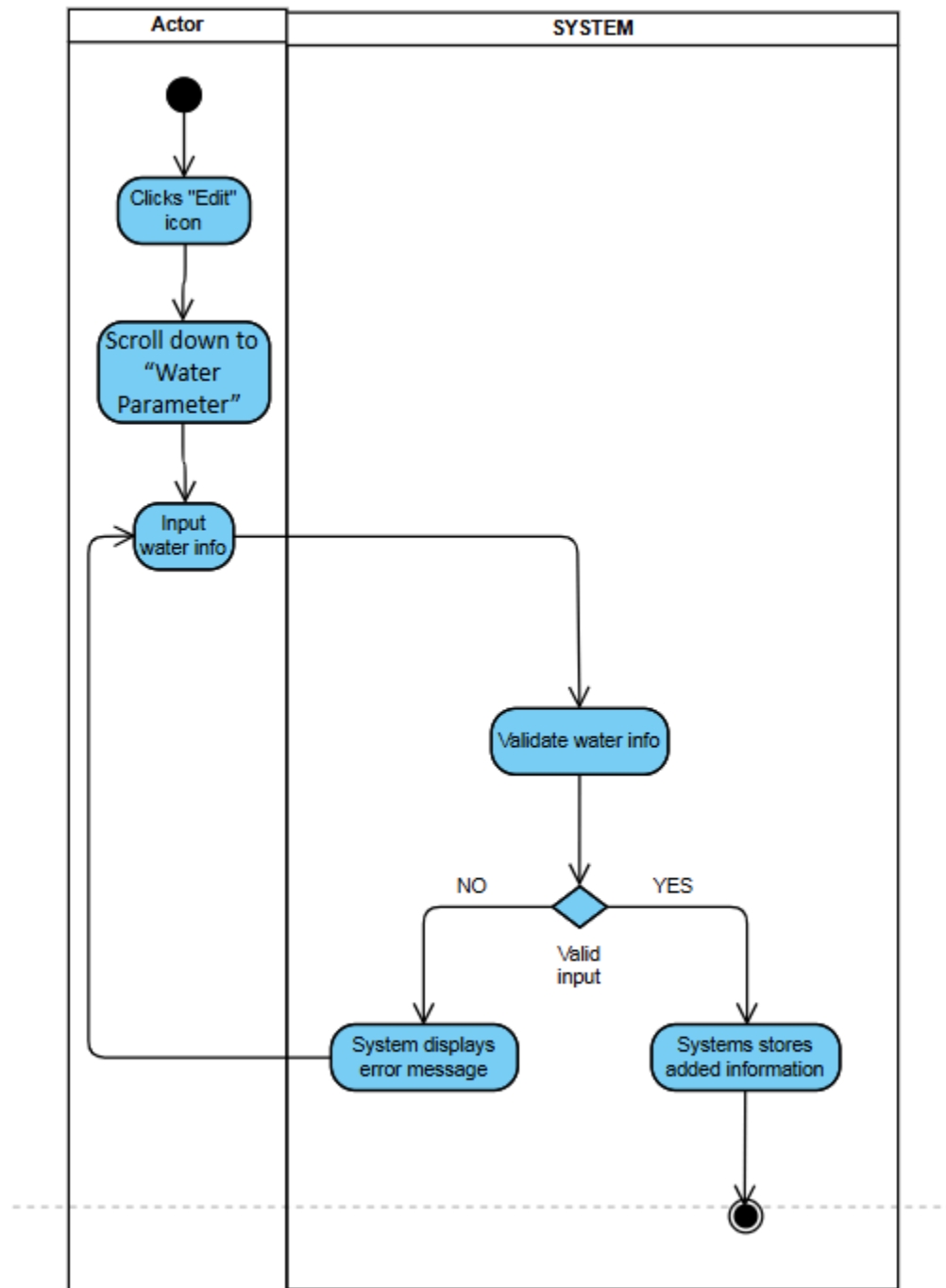
b. Business Rules

BR-02: System must verify all information again before storing new updated information to database.

BR-03: Actor must be authenticated.

BR-04: Only the owner actor has the right to edit

c. Activity diagram



3.3 UC12_View water quality

a. Functionalities

Provide the functional description for the use cases using the template/guides below

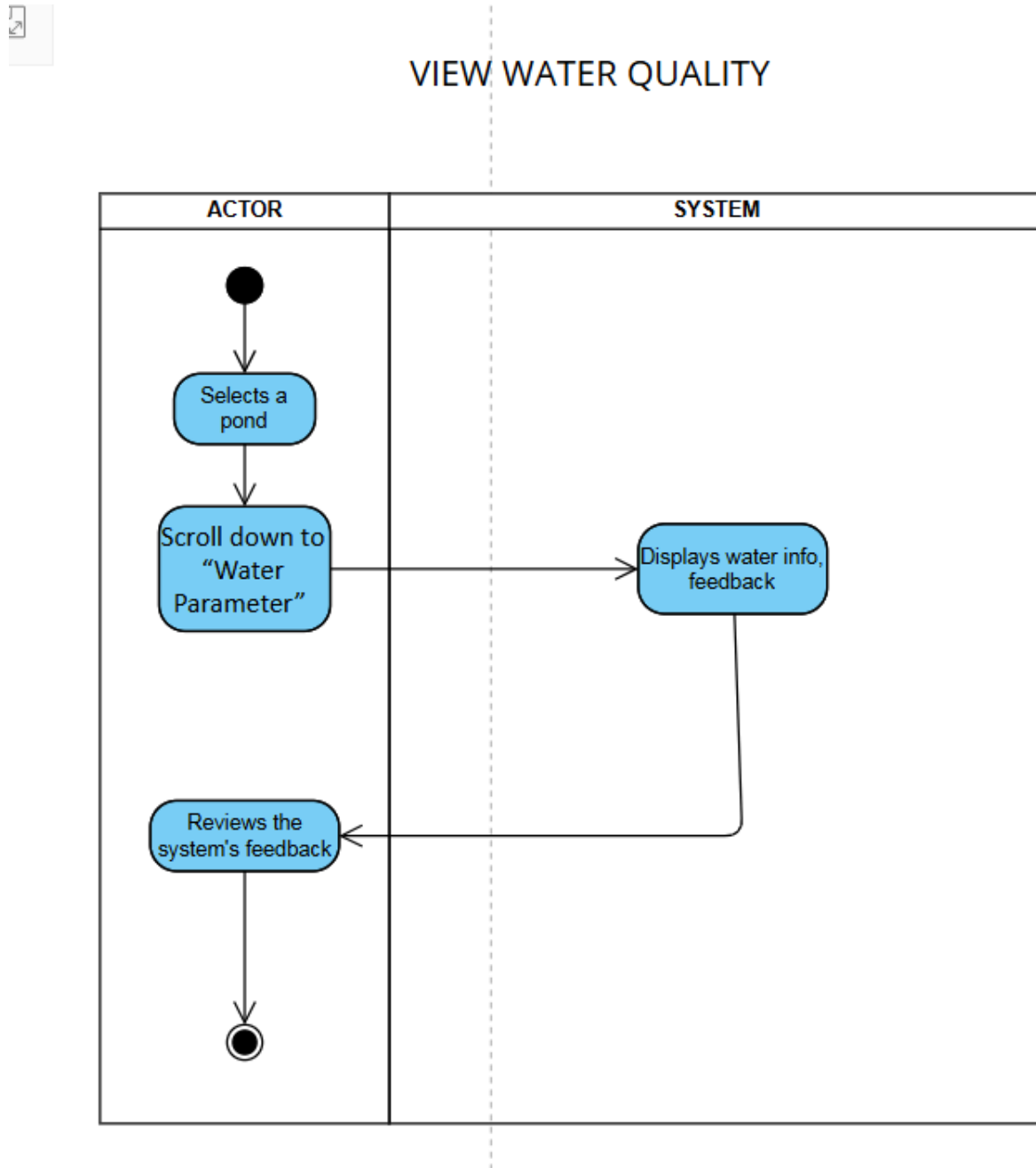
Functional Description Template

UC ID and Name:	UC12_View water quality		
Created By:	Hoàng Quý Dương	Date Created:	15/9
Primary Actor:	Customer	Secondary Actors:	N/A
Trigger:	The actor navigates to the “My Pond” section.		
Description:	This use case allows the actor to view the water’s quality.		
Preconditions:	The water information has already existed in the system.		
Postconditions:	The actor is able to view the water’s quality.		
Normal Flow:	<ol style="list-style-type: none">1. Actor navigates to the “Manage Koi” page2. Actor clicks the “My Pond” card.3. The actor scrolls down to “Water Parameter” section.4. The system displays a feedback of the water’s current state based on the information provided by the actor:<ul style="list-style-type: none">- Temperature- Salt level- pH level- Oxygen level- NO2 concentration- NO3 concentration- PO4 concentration5. The actor reviews the system’s feedback.		
Alternative Flows:	N/A		
Exceptions:	1. The system fails to retrieve data due to a server issue.		
Priority:	High		
Frequency of Use:	N/A		
Business Rules:	BR-03		
Other Information:	N/A		
Assumptions:	The actor regularly checks on the water quality and updates it.		

b. Business Rules

BR-03: Actor must be authenticated.

c. Activity diagram



3.4 UC13_Remove water quality

a. Functionalities

Provide the functional description for the use cases using the template/guides below

Functional Description Template

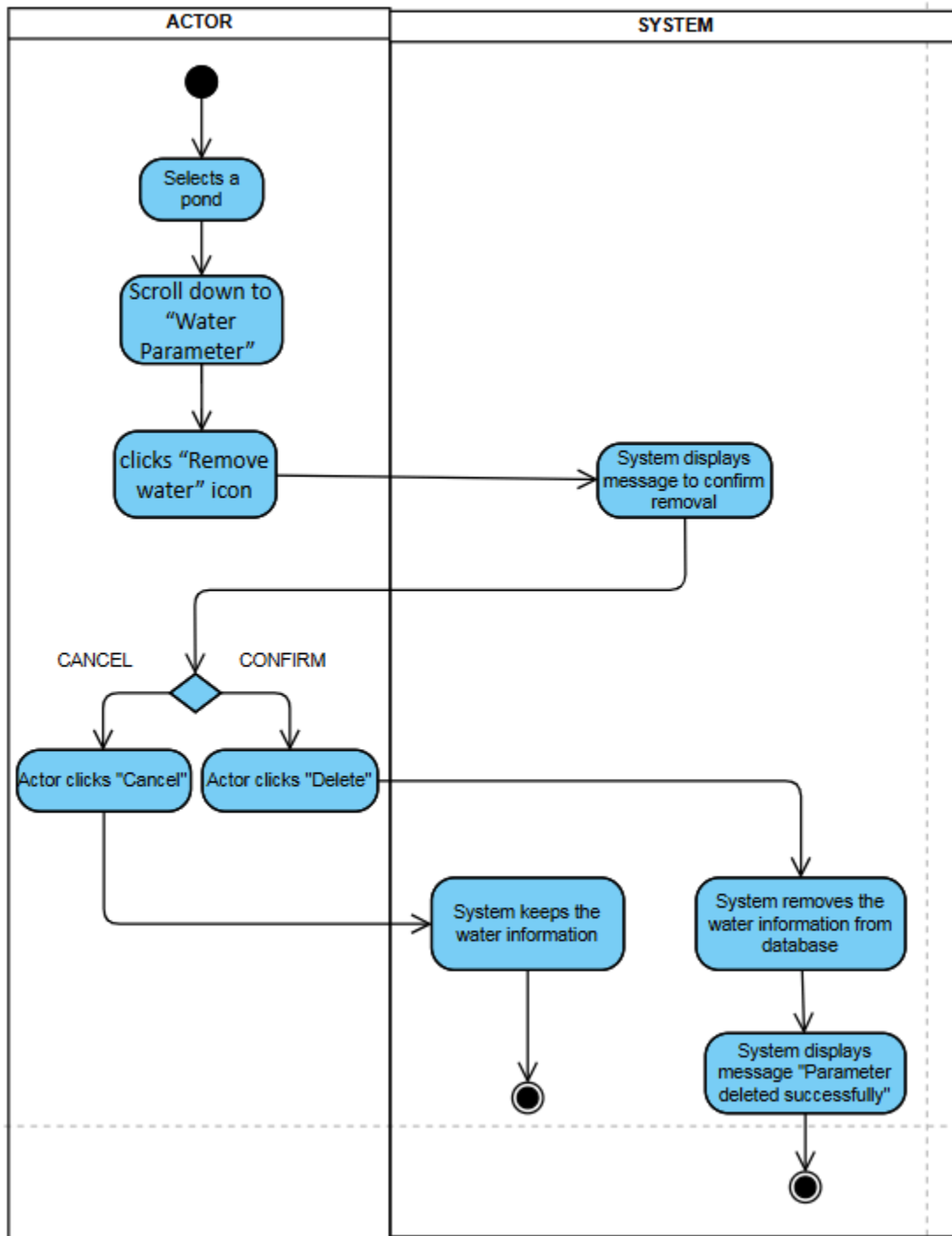
UC ID and Name:	UC13_Remove water quality		
Created By:	Hoàng Quý Dương	Date Created:	15/9

Primary Actor:	Customer	Secondary Actors:	N/A
Trigger:	The actor clicks the “Remove water” icon.		
Description:	This use case allows the actor to remove water from the pond.		
Preconditions:	The water information has already existed in the system.		
Postconditions:	The water information was removed successfully.		
Normal Flow:	<ol style="list-style-type: none"> 1. Actor navigates to the “Manage Koi” page 2. Actor clicks the “My Pond” card. 3. The actor scrolls down to “Water Parameter” section. 4. The actor clicks the “Remove water” icon. 5. The system shows a popup asking “Are you sure you want to delete this parameter?” with buttons “Delete” and “Cancel” [Alternative 1]. 6. The actor clicks “Delete”. 7. The system displays the message “Parameter deleted successfully!” [Exception 1] and disables the data to hide from the UI. 		
Alternative Flows:	<ol style="list-style-type: none"> 1. The actor clicks “Cancel”, the system will keep the water quality information. 		
Exceptions:	<ol style="list-style-type: none"> 1. The system fails to delete data due to a server issue and will display a message “Failed to delete parameter!”. 		
Priority:	High		
Frequency of Use:	N/A		
Business Rules:	BR-03		
Other Information:	N/A		
Assumptions:	The system performs a final confirmation before deletion to avoid accidental removal.		

b. Business Rules

BR-03: Actor must be authenticated.

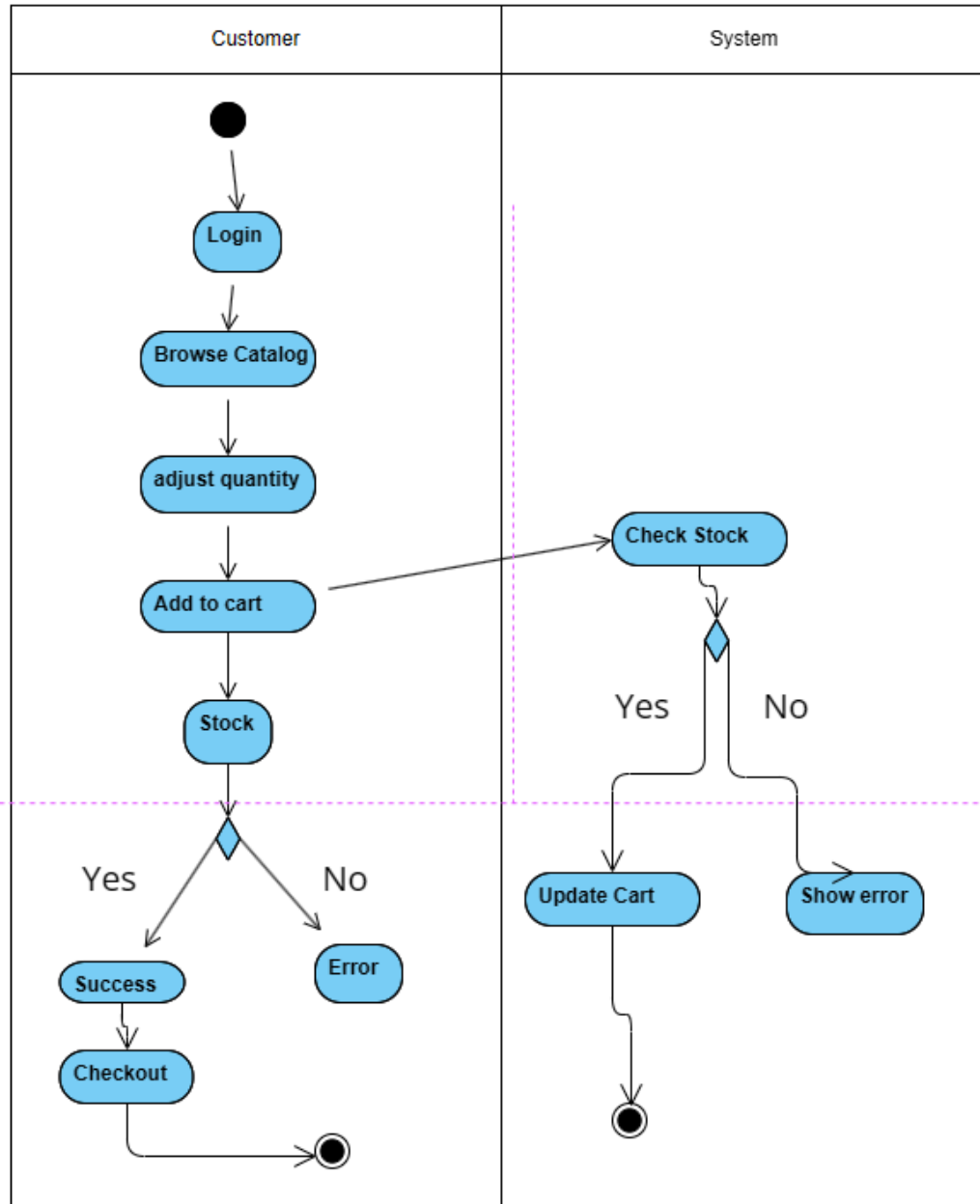
c. Activity diagram



4. Shop Koi Fish Products

4.1 UC15_Add to Cart Koi Fish Product

activity diagram

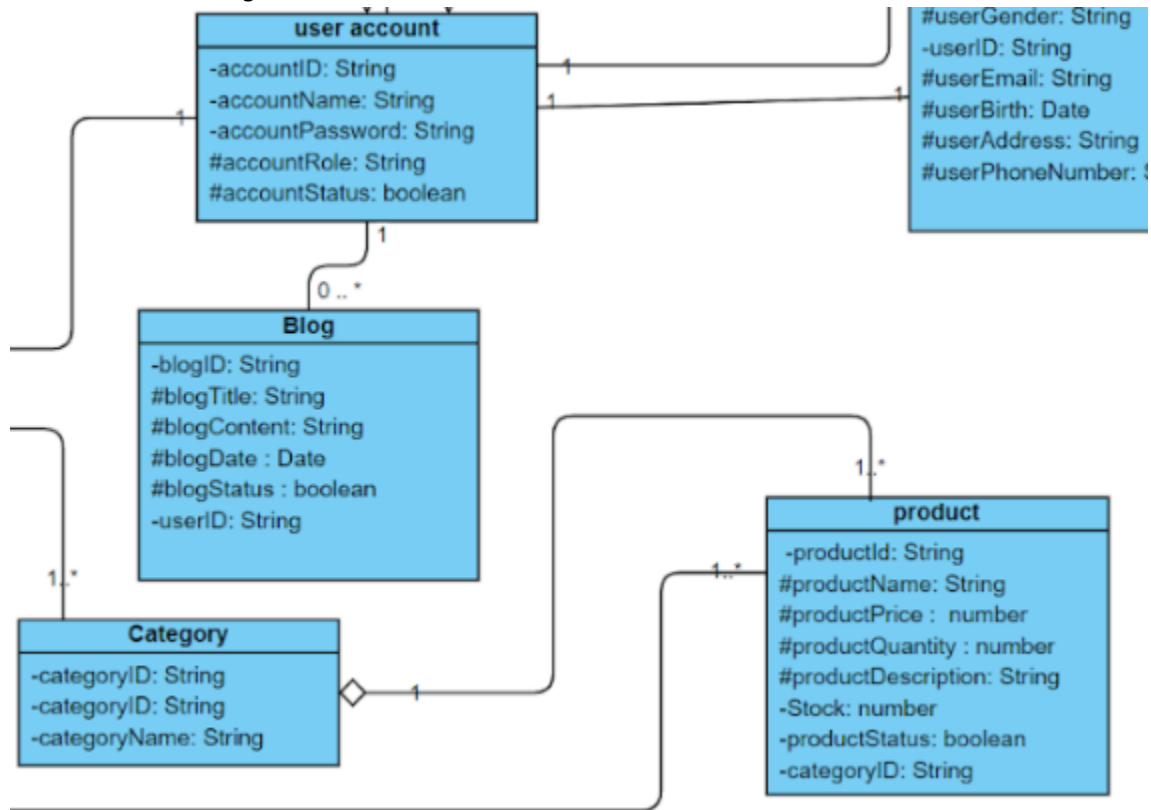


a. Functional Description

ID and Name:	UC15 - Add to Cart Koi Fish Product		
Created By:	Trong Thang	Date Created:	15/09/2024
Primary Actor:	Customer	Secondary Actors:	Staff, System

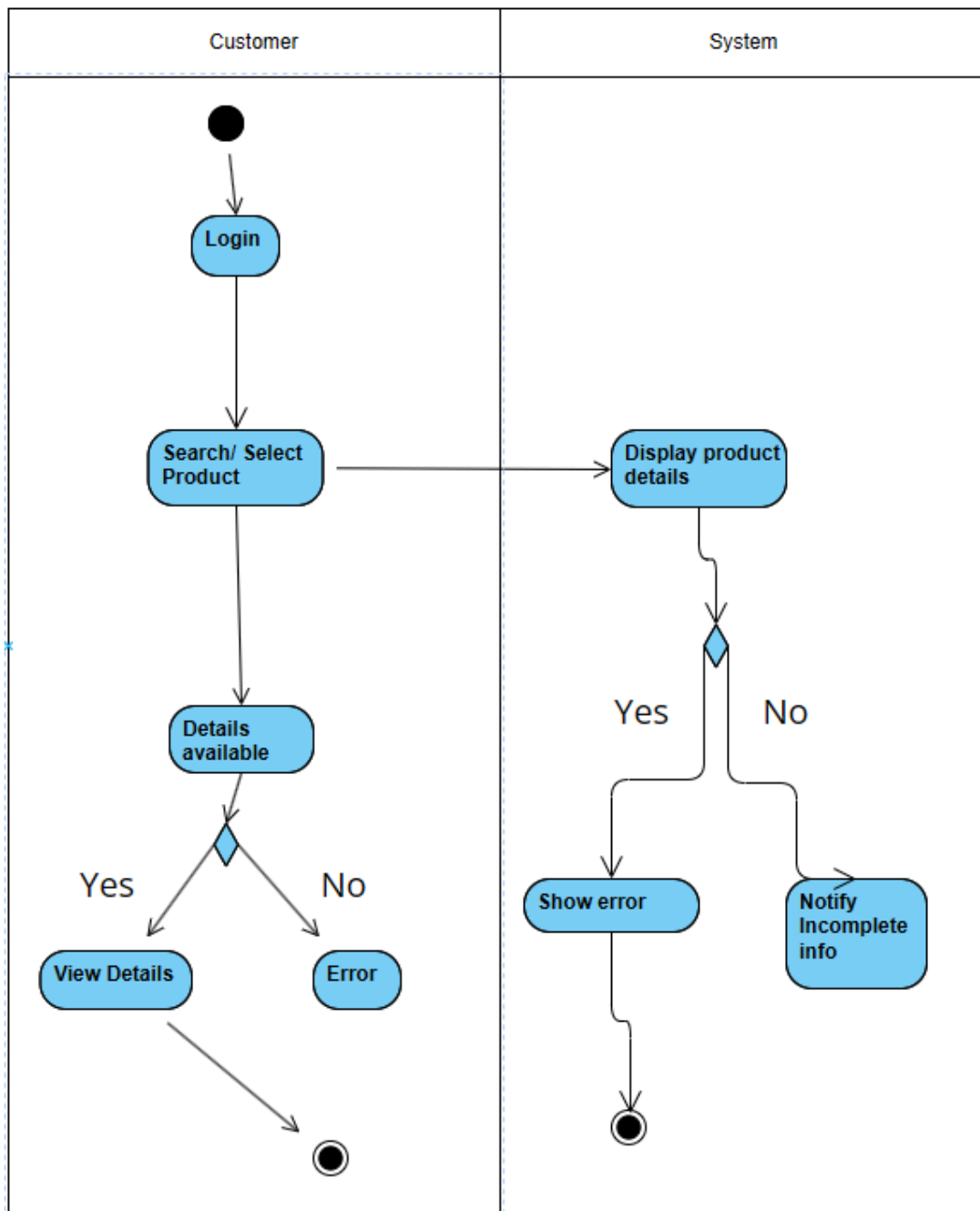
Description:	This use case allows actor to browse and add various Koi fish product to their cart from the shop's catalog.
Trigger:	Actor clicks the "Add to Cart" button for Koi fish product from the catalog.
Preconditions:	1. Actor must be logged into the system. 2. The product must be in stock.
Postconditions:	1. The item is successfully added to the cart. 2. Actor can proceed to checkout or continue shopping.
Normal Flow:	1. Actor logs into the system. 2. Actor browses the catalog. 3. Actor click button "-" or "+" to adjust the number of items 4. Actor click button "Add to Cart" button 5. The system updates the cart and stock accordingly.
Alternative Flows:	Show message "Stock issues", the system alerts the customer with an error message when adding to the cart.
Exceptions:	The system fails to process the order due to out-of-stock items.
Priority:	High, Must Have
Frequency of Use:	N/A
Business Rules:	BR9, BR10
Other Information:	N/A
Assumptions:	The system has accurate stock management

e. Function class diagram



4.2 UC16_View Product Details

activity diagram

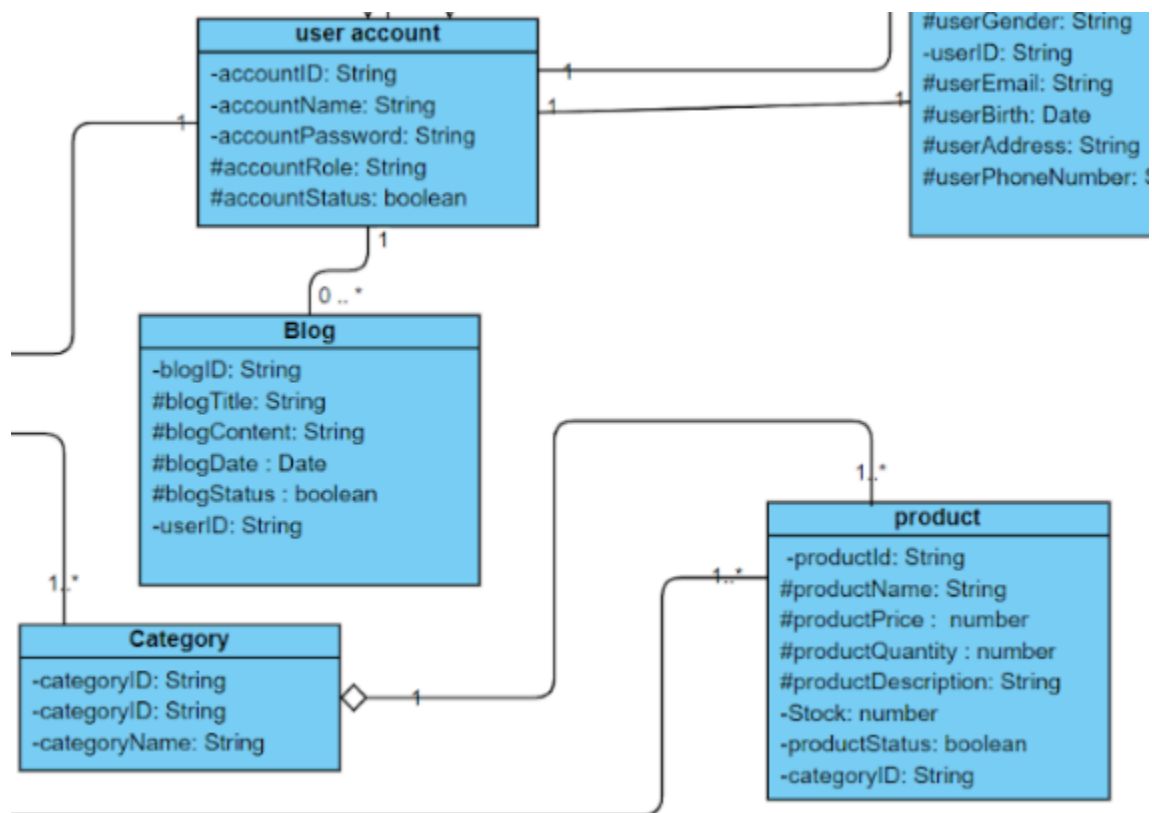
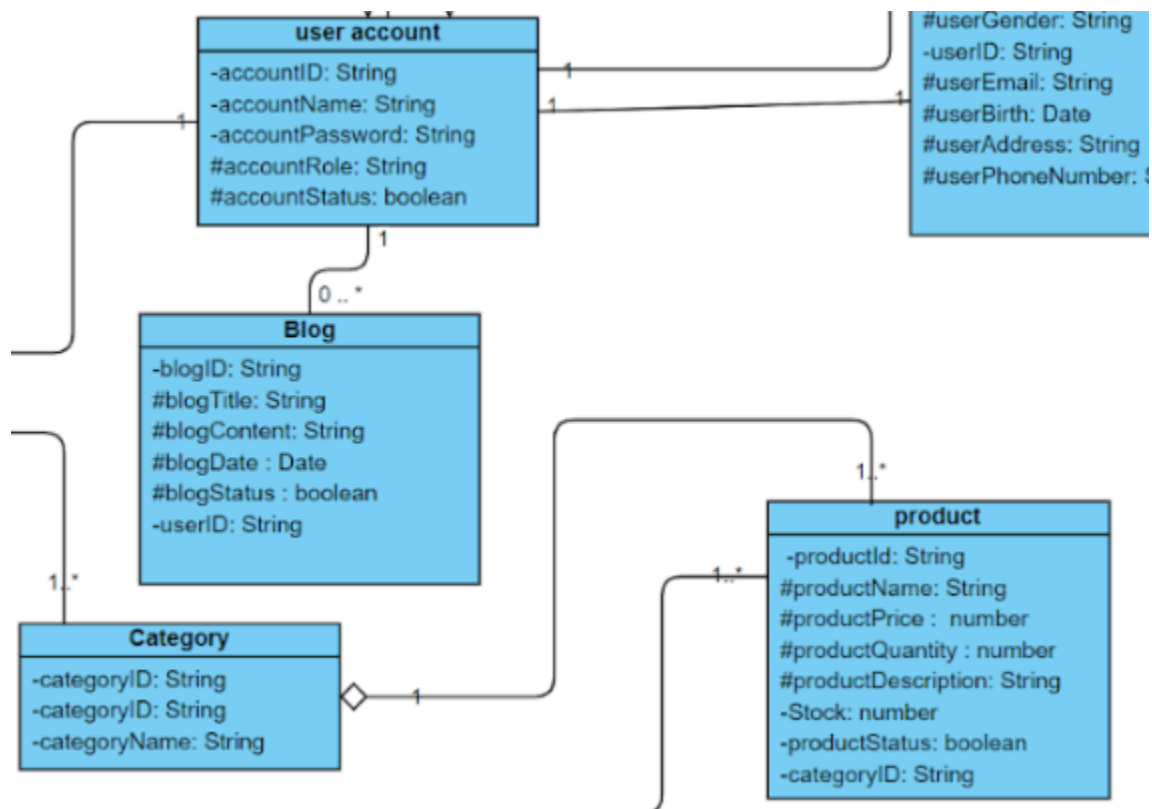


a. Functional Description

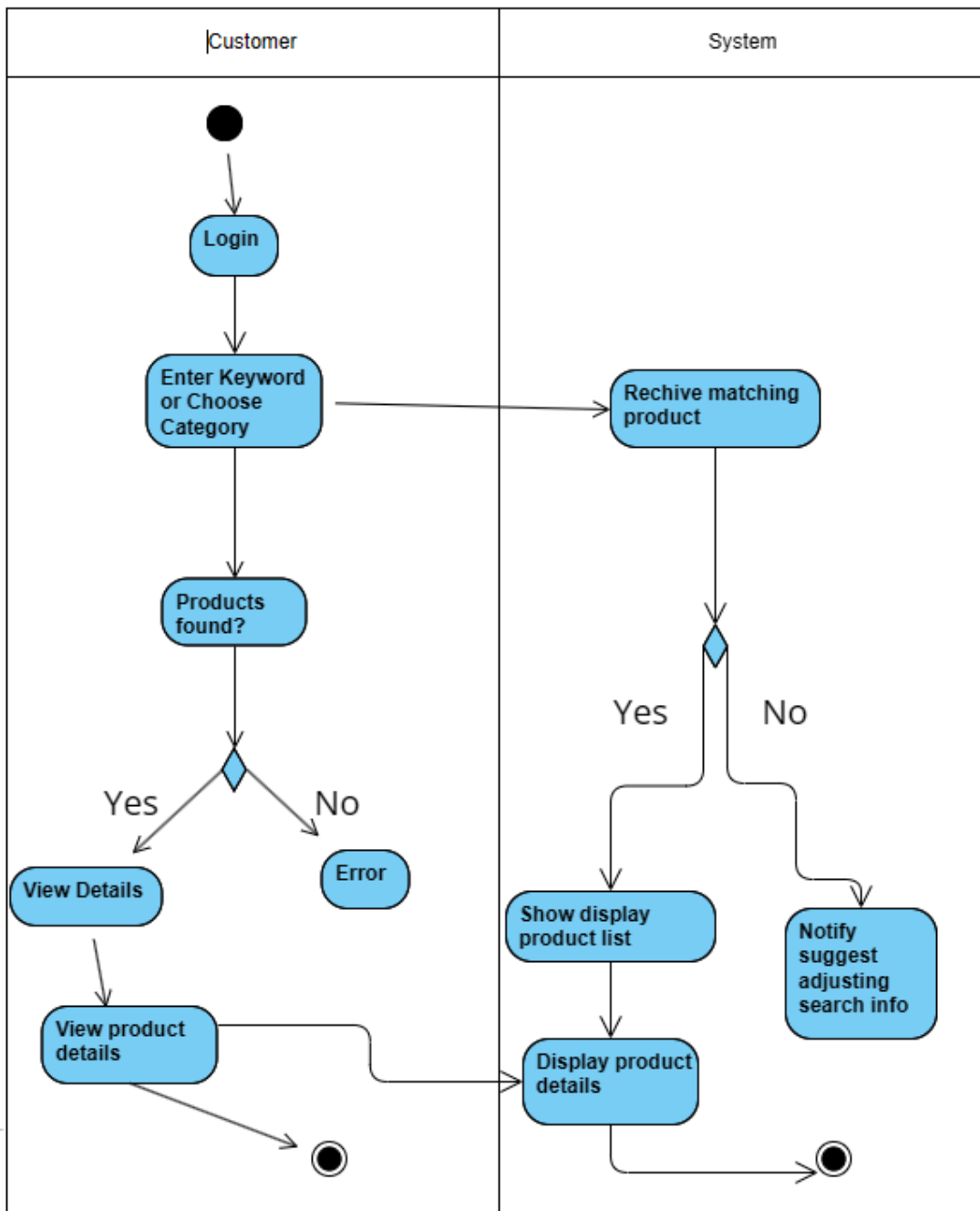
ID and Name:	UC16 - View Product Details		
Created By:	Trong Thang	Date Created:	15/09/2024
Primary Actor:	Customer	Secondary Actors:	Staff, System

Description:	This use case allows actor views detailed information about Koi fish product, including descriptions, specifications, and reviews.
Trigger:	The actor selects a product to view more information about it.
Preconditions:	<ol style="list-style-type: none"> 1. Actor must be logged into the system. 2. The product must be listed in the catalog.
Postconditions:	The actor views detailed information (name, price, description, specs) about the selected product.
Normal Flow:	<ol style="list-style-type: none"> 1. Actor logs into the system. 2. Actor searches for or selects a product from the catalog. 3. The system displays detailed information (name, price, description, specs) about the selected product.
Alternative Flows:	If the product information is incomplete, the system notifies the customer.
Exceptions:	Product information is not available due to a system error.
Priority:	High, Must Have
Frequency of Use:	N/A
Business Rules:	BR11, BR12
Other Information:	N/A
Assumptions:	The product catalog is maintained with accurate and up-to-date information.

e. Function class diagram



4.3 UC17_Search and View Product activity diagram

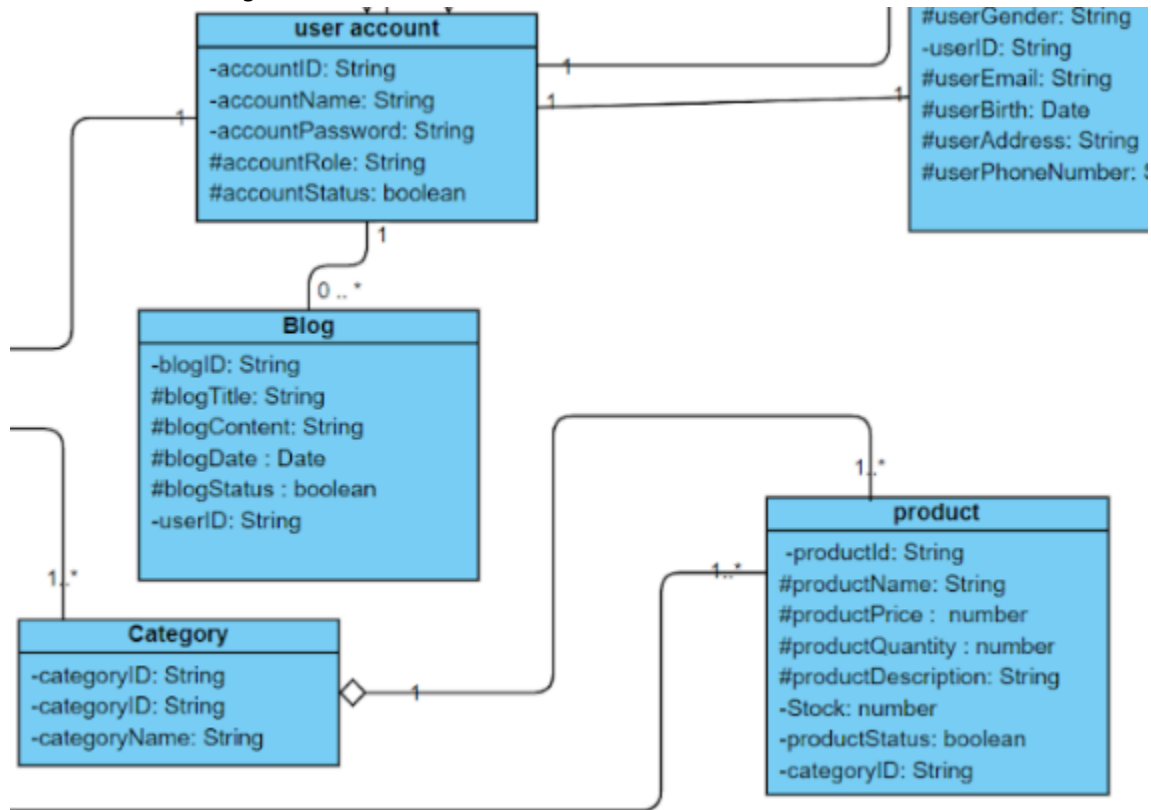


a. Functional Description

ID and Name:	UC17 -Search and View Product		
Created By:	Trong Thang	Date Created:	15/09/2024
Primary Actor:	Customer	Secondary Actors:	N/A

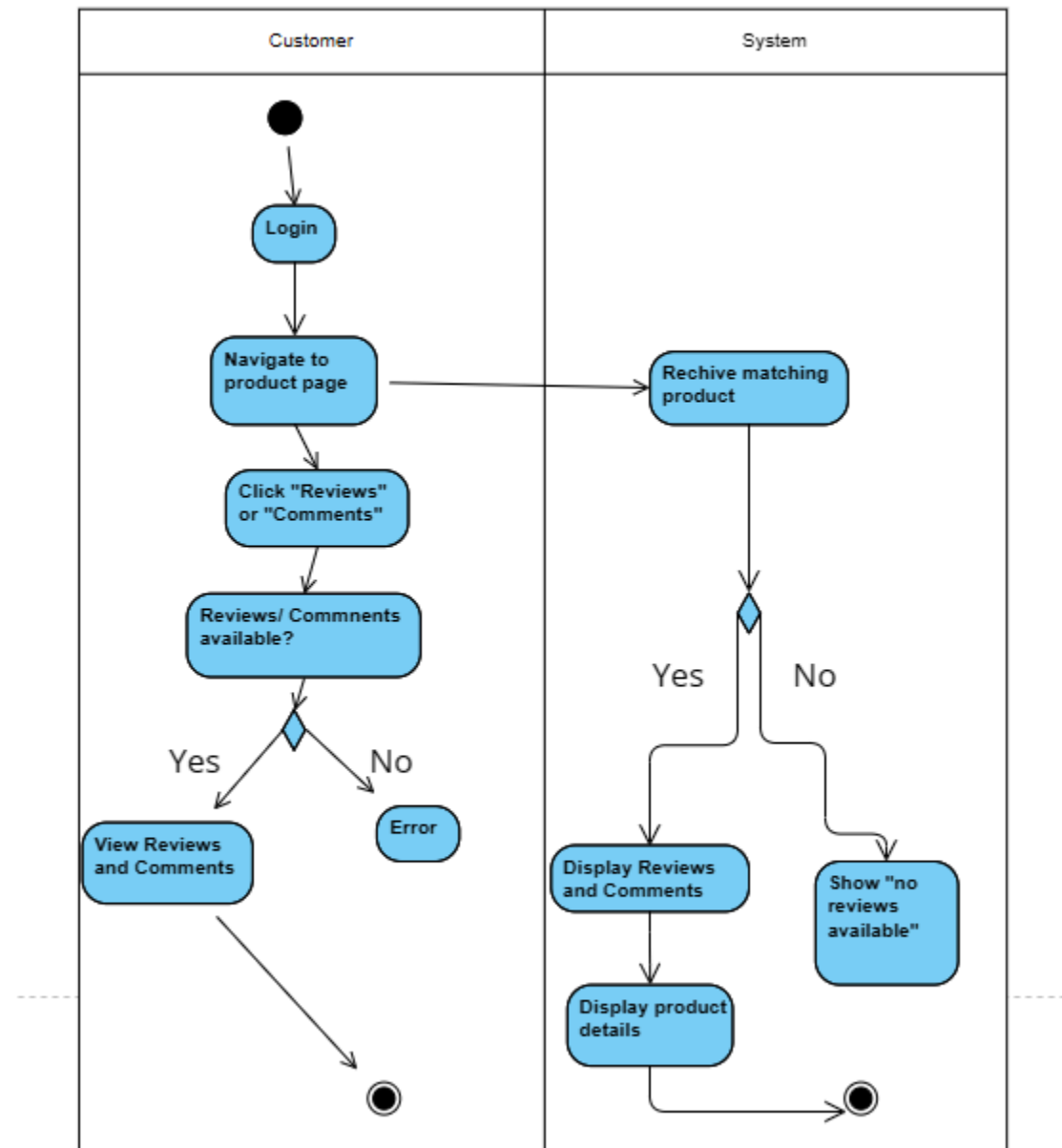
Description:	This use case allows actor to search for koi equipment by keyword or category and view detailed information about the selected product.
Trigger:	The actor enters a keyword or selects a category to search for a product.
Preconditions:	The system must have a database of products available and up to date.
Postconditions:	The actor views a list of products that match their search criteria and can access detailed product information.
Normal Flow:	<ol style="list-style-type: none"> 1. Actor logs into the system. 2. Actor enters a keyword or chooses a category in the search bar. 3. The system retrieves and displays a list of products matching the criteria. 4. Actor selects a product to view. 5. The system displays the product's details (name, price, description, specs).
Alternative Flows:	If no products match the search criteria, the system informs the customer and suggests adjusting the search parameters.
Exceptions:	System fails to load or display search results due to system error.
Priority:	High, Must Have
Frequency of Use:	N/A
Business Rules:	BR13, BR14
Other Information:	N/A
Assumptions:	The product database is well-maintained and searchable.

e. Function class diagram



4.4 UC18_View Product Reviews and Comments

activity diagram

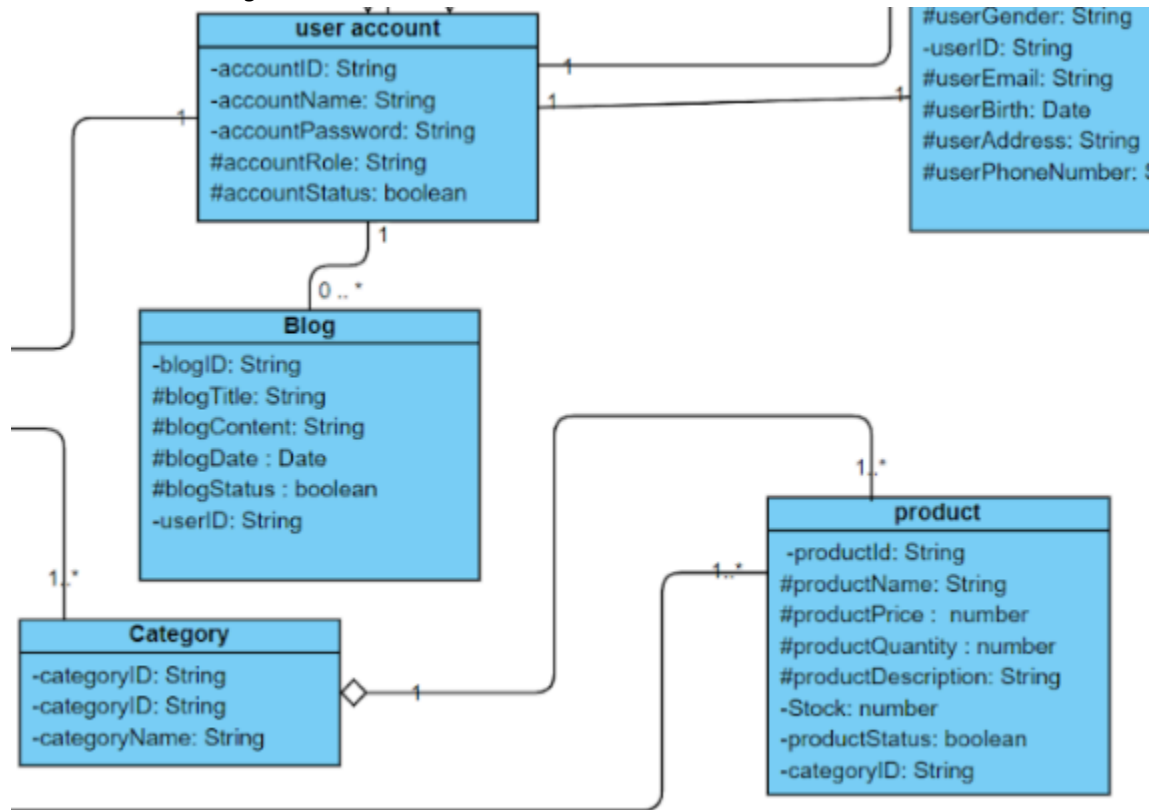


a. Functional Description

ID and Name:	UC18 - View Product Reviews and Comments		
Created By:	Trong Thang	Date Created:	15/09/2024
Primary Actor:	Customer	Secondary Actors:	N/A
Description:	This use case allows actor to view reviews and comments left by other customers for a particular product.		

Trigger:	The actor clicks on the "Reviews" or "Comments" tab on the product page.
Preconditions:	The product must have reviews and comments available in the system.
Postconditions:	Actor views all reviews and comments for the selected product.
Normal Flow:	<ol style="list-style-type: none"> 1. Actor logs into the system. 2. Actor navigates to a product's detail page. 3. Actor clicks on the "Reviews" or "Comments" tab. 4. The system retrieves and displays all reviews and comments for the product.
Alternative Flows:	If no reviews or comments are available, the system informs the customer.
Exceptions:	System fails to load reviews or comments due to system error.
Priority:	Medium, Should Have
Frequency of Use:	N/A
Business Rules:	BR15, BR16
Other Information:	N/A
Assumptions:	The system maintains a clean and organized database of product reviews and comments.

e. Function class diagram



5. Manage Cart

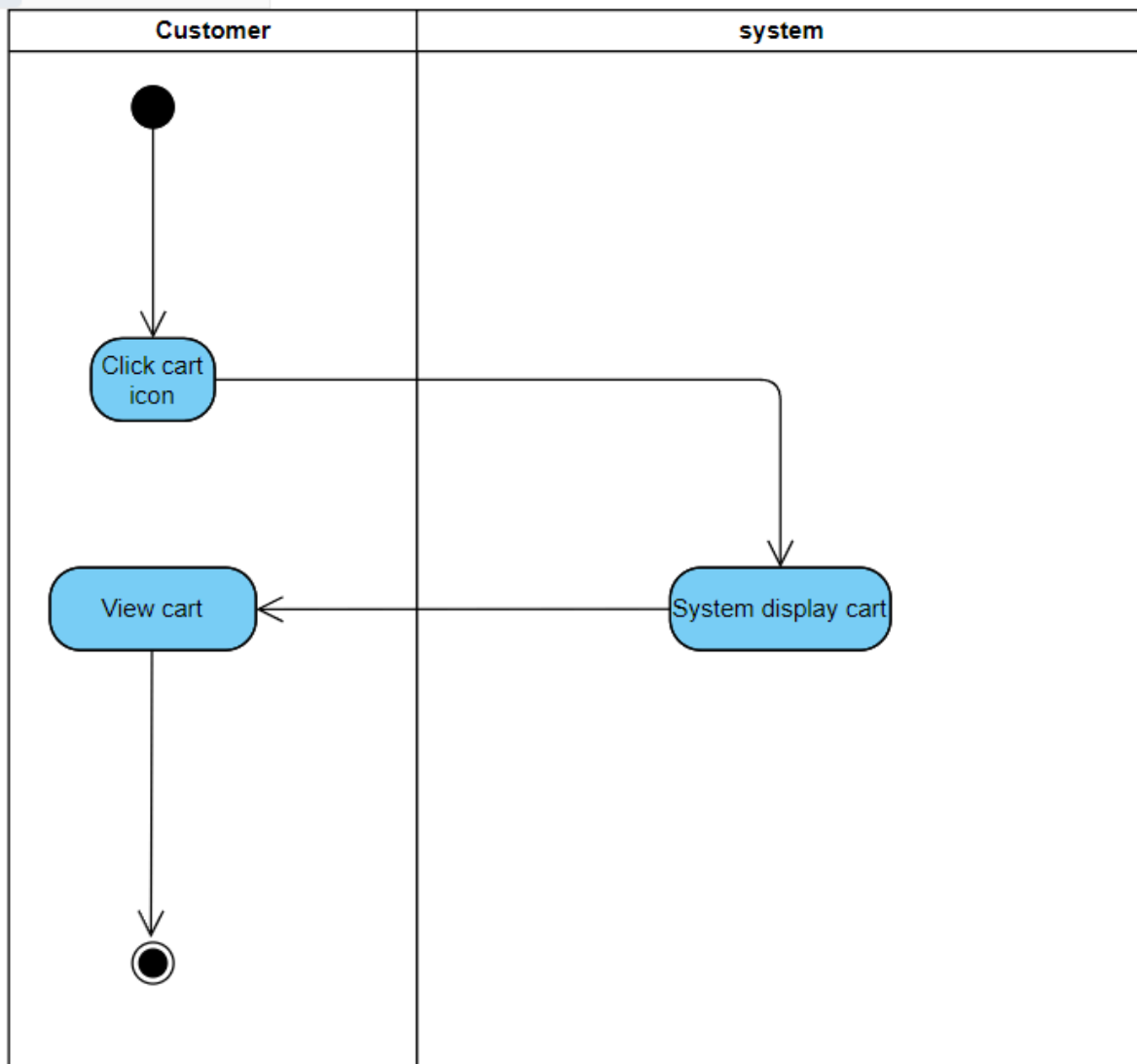
5.1 UC-17 View cart

a. Functional Description

ID and Name:	UC-17 View cart		
Created By:	Trần Đại Nghĩa	Date Created:	16/09/2024
Primary Actor:	Customer	Secondary Actor:	N/A
Description:	This use case allows actor to view their cart		
Trigger:	An actor want to view cart information		
Preconditions:	PRE-1. Actor must be authenticated		
Postconditions:	POST-1. Customer clicks on a product to view more details from the cart		
Normal Flow:	View cart information <ol style="list-style-type: none"> Actor enters the “cart management” page The system displays all product of current cart include: <ul style="list-style-type: none"> Name of product Price of product Amount of product 		
Alternative Flows:	N/A		
Exceptions:	1. System break-time		

	- System do not stores any new information and keeps last version of Cart information
Priority:	medium
Frequency of Use:	N/A.
Business Rules:	BR-01, BR-02, BR-03
Other Information:	N/A
Assumptions:	N/A

activity diagram



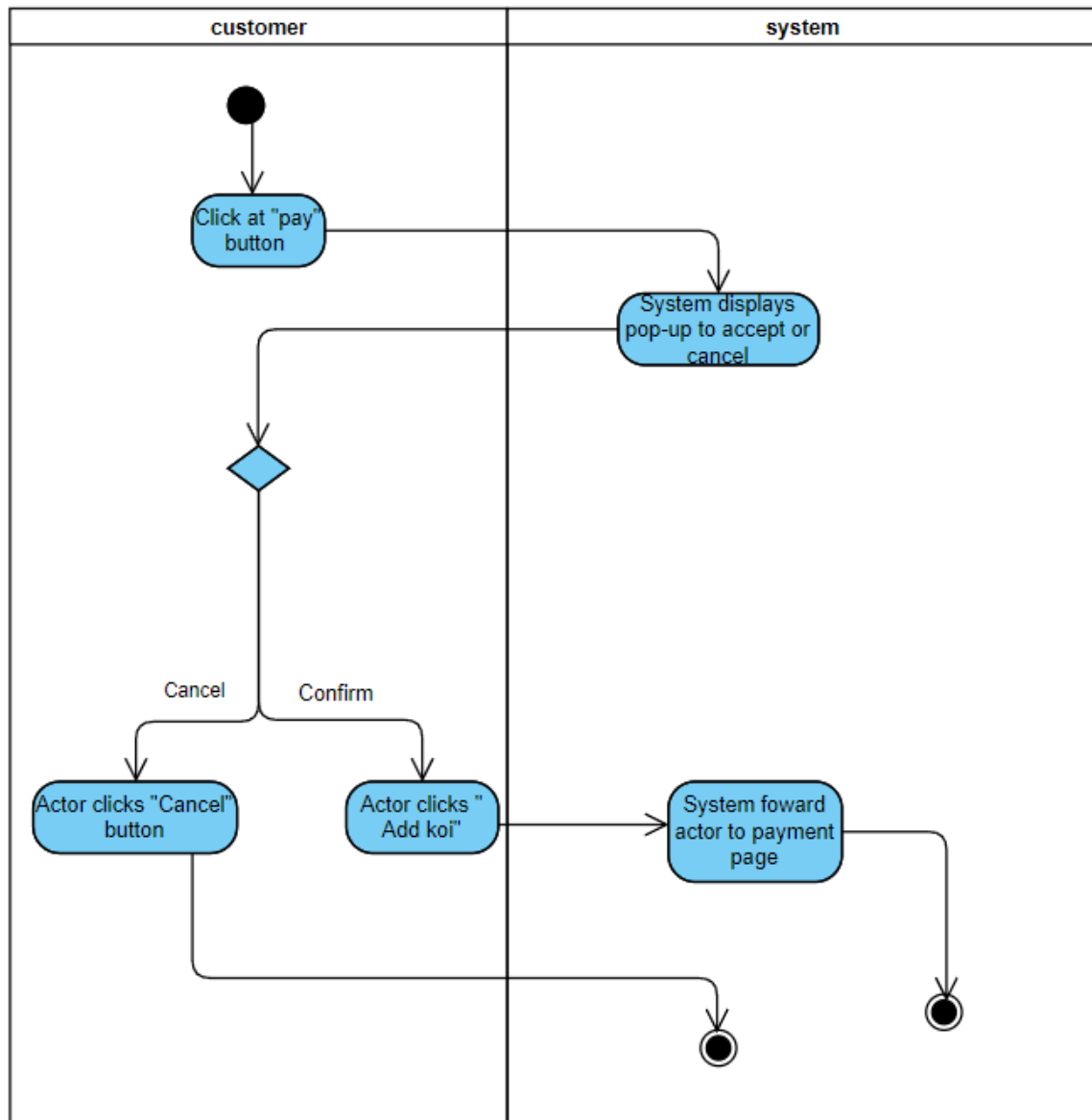
5.2 UC-18_ Pay for cart

a. Functional Description

ID and Name:	UC-18 Pay for cart		
Created By:	Trần Đại Nghĩa	Date Created:	16/09/2024
Primary Actor:	Customer	Secondary Actor:	N/A
Description:	This use case allows actor browse and purchase all product in cart		
Trigger:	Actor clicks the "Pay" button in cart		

Preconditions:	PRE-1. Customer must be logged into the system. PRE-2. The product must be in stock.
Postconditions:	POST-1. The purchase is successfully processed and order details are saved. POST-2. Customer receives confirmation of purchase.
Normal Flow:	Purchase from cart <ol style="list-style-type: none"> 1. Actor enters the “cart management” page 2. Actor click “Pay” button 3. System display a pop up “Do you want to Pay” and require customer accept or cancel 4. Actor click Accept button, system forward to payment page
Alternative Flows:	<ol style="list-style-type: none"> 1. Stock issues or payment problems <ul style="list-style-type: none"> - 1.1 the system alerts the customer with an error message. 2. Actor clicks “Cancel” button <ul style="list-style-type: none"> - 2.1 System displays pop-up message “Cancel successfully”
Exceptions:	<ol style="list-style-type: none"> 1. System break-time <ul style="list-style-type: none"> - System do not stores any new information and keeps last version of cart information 2. The system fails to process the order due to payment gateway issues or out-of-stock items.
Priority:	medium
Frequency of Use:	N/A.
Business Rules:	BR-01, BR-02, BR-03
Other Information:	N/A
Assumptions:	N/A

activity diagram



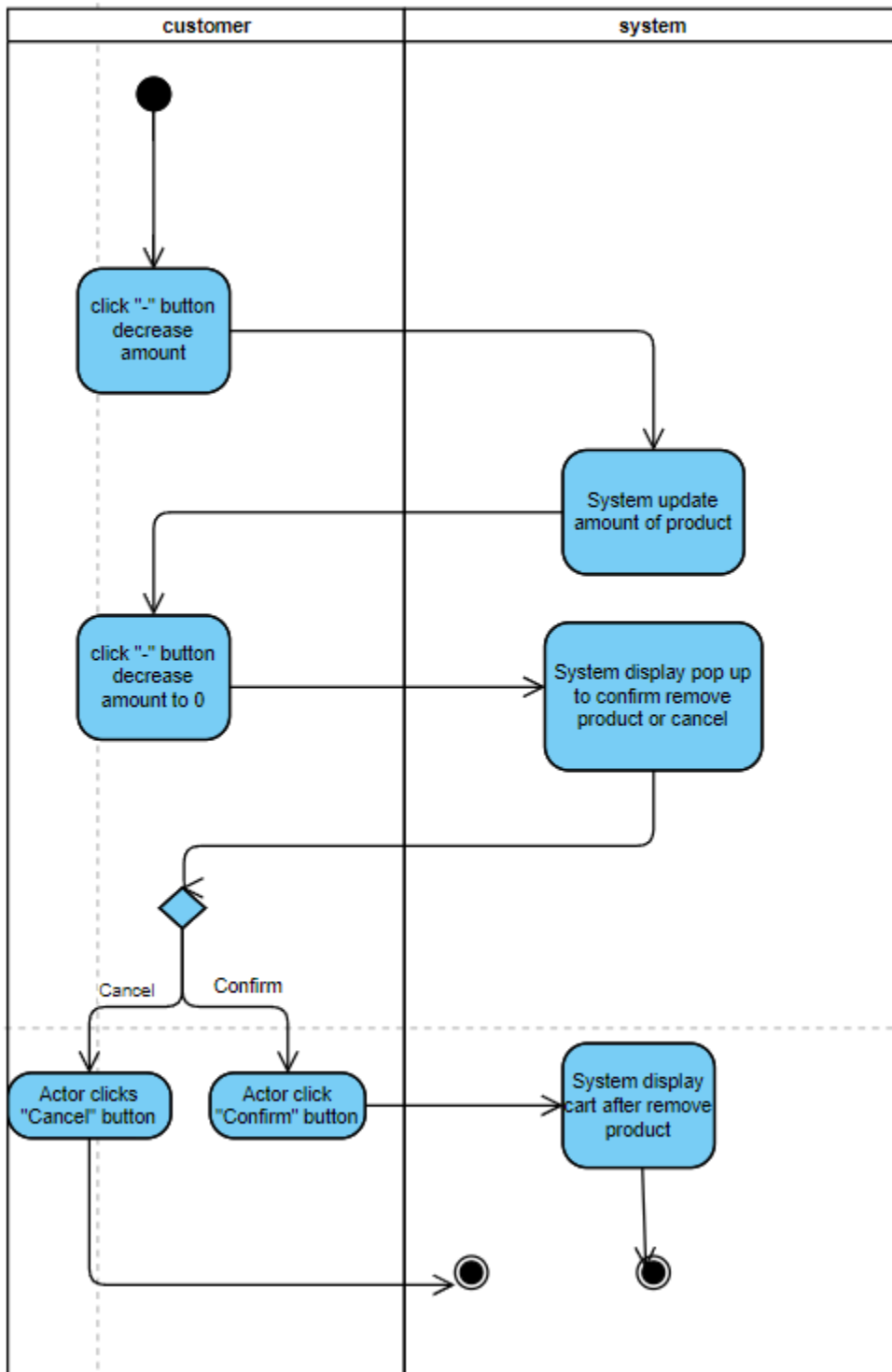
5.3 UC-19_ Remove product from cart

a. Functional Description

ID and Name:	UC - 19 Remove product from cart
--------------	---

Created By:	Trần Đại Nghĩa	Date Created:	16/09/2024
Primary Actor:	Customer	Secondary Actor:	Staff, System
Description:	This use case allows actor remove product from cart		
Trigger:	Actor want to remove a product from cart		
Preconditions:	PRE-1. Customer must be logged into the system. PRE-2. The product must be in the cart		
Postconditions:	POST-1. The Product is successfully remove from cart POST-2. Customer receives successful message		
Normal Flow:	Remove from cart <ol style="list-style-type: none"> 1. Actor enters the “cart management” page 2. Actor click button “-” of amount 3. If amount is 0 the system display a message “Do you want to remove this product” to ask customer if they want to remove that product from cart 4. Customer click Accept button, that product is successfully remove from cart 		
Alternative Flows:	<ol style="list-style-type: none"> 1. Actor clicks “Cancel” button <ul style="list-style-type: none"> - 1.1 System displays pop-up message “Cancel successfully” 		
Exceptions:	The system fails to process the order due to payment gateway issues or out-of-stock items.		
Priority:	medium		
Frequency of Use:	N/A.		
Business Rules:	BR-01, BR-02, BR-03, BR-04		
Other Information:	N/A		
Assumptions:	N/A		

activity diagram

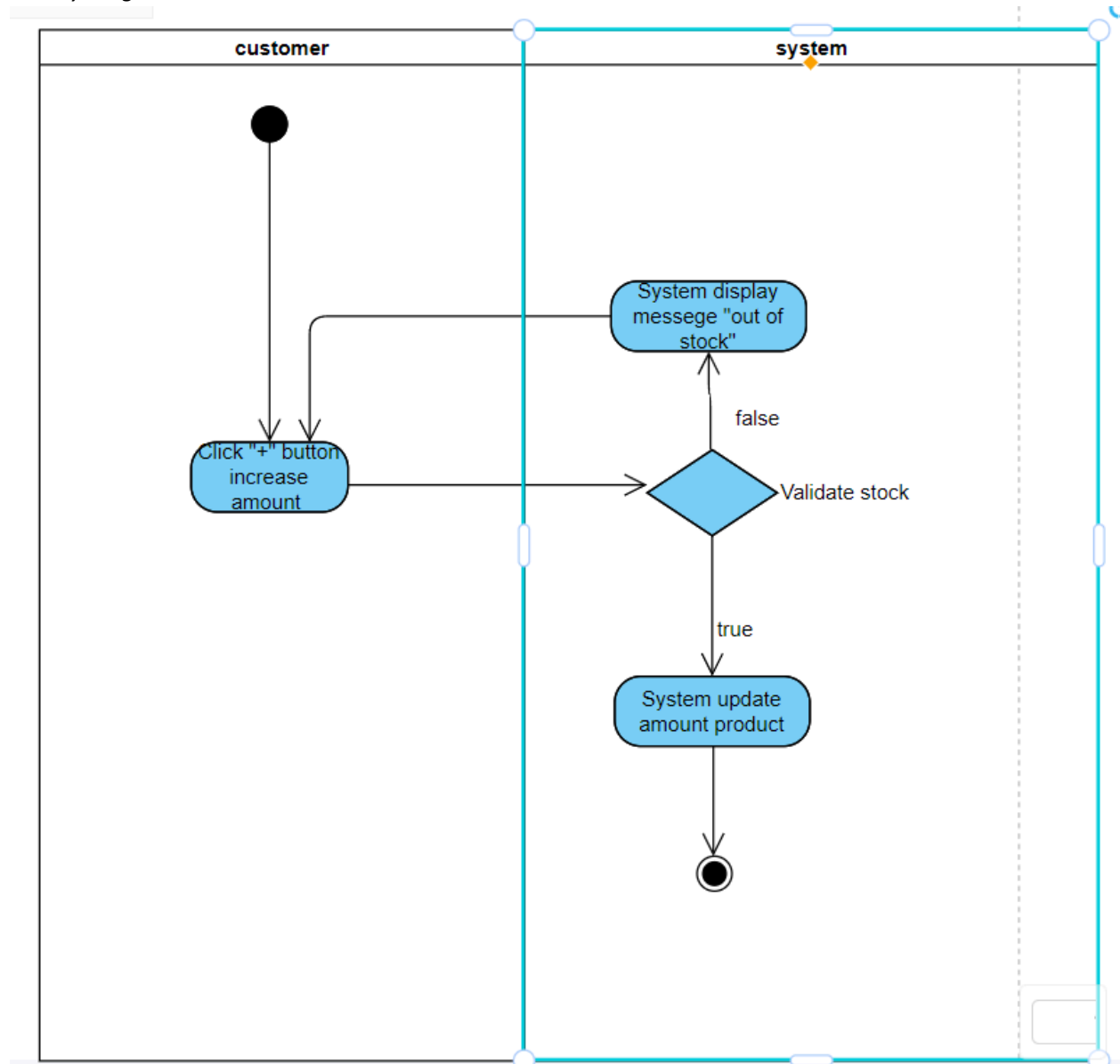


5.4 UC-20_ Add more product

a. Functional Description

ID and Name:	UC - 20 Add more product		
Created By:	Trần Đại Nghĩa	Date Created:	16/09/2024
Primary Actor:	Customer	Secondary Actor:	Staff, System
Description:	This use case allows actor add more a product that already exist in cart		
Trigger:	Actor want to add more a product that already exist in cart		
Preconditions:	PRE-1. Customer must be logged into the system. PRE-2. The product must be in the cart		
Postconditions:	POST-1. The Product is successfully add more amount to cart POST-2. Customer receives successful message		
Normal Flow:	Add more Product to Cart <ol style="list-style-type: none"> 1. Actor enters the “cart management” page 2. Actor click button “+” of amount 3. System display amount of product after add 		
Alternative Flows:	N/A		
Exceptions:	The system fails to process the order due to payment gateway issues or out-of-stock items.		
Priority:	medium		
Frequency of Use:	N/A.		
Business Rules:	BR-01, BR-03, BR-18, BR-19		
Other Information:	N/A		
Assumptions:	N/A		

activity diagram



b. Business Rules

BR-01: When updating, if system is down-time, the system keeps the pre-version of information and does not store any new updated information of cart.

BR-02: Actor must be authenticated

BR-03: Only the owner actor has the right to view their cart

BR-04: The system must update amount after each amount change

IV. Appendix

1. Assumptions & Dependencies

[Record any assumptions that were made when conceiving the project and writing this vision and scope document. Note any major dependencies the project must rely upon for success, such as specific technologies, third-party vendors, development partners, or other business relationships.]

<<Sample:

- AS-1: Systems with appropriate user interfaces will be available for cafeteria employees to process the expected volume of meals ordered.
- AS-2: Cafeteria staff and vehicles will be available to deliver all meals for specified delivery time slots within 15 minutes of the requested delivery time.
- DE-1: If a restaurant has its own on-line ordering system, the Cafeteria Ordering System must be able to communicate with it bi-directionally.

>>

2. Limitations & Exclusions

[Identify any product features or characteristics that a stakeholder might anticipate, but which are not planned to be included in the new product]

3. Business Rules

ID	Category	Rule Definition
BR-01	System update rule	When updating, if system is down-time, the system keeps the pre-version of information and does not store any new updated information of Koi pond.
BR-02	System update rule	System must verify all information again before storing new updated information to database.
BR-03	Authentication rule	Actor must be authenticated
BR-04	Authentication rule	Only the owner actor has the right to edit profile
BR-05	Authentication rule	Only admin of system has right to manage account
BR-06	Information Assurance rule	Pond ID is distinguished and can not be duplicated.
BR-07	Authentication rule	Staff can only view customer information and can not edit
BR-08	System update rule	During the updating time, if the system is down-time, the oldest version of count data would be kept and displayed in count history.
BR-09	System update rule	The system must update amount after each amount change

BR-10	Authentication Requirement	Only authenticated users can add items to the cart.
BR-11	Accurate Product Details	Product details must be accurate and up-to-date.
BR-12	Detailed Information	Detailed information should include specifications, images, and customer reviews.
BR-13	Searchable Products	Products must be searchable by name, category, or keywords.
BR-14	Accurate Product Information	Information displayed must be accurate and updated regularly.
BR-15	Registered Users Only	Only registered customers can leave reviews or comments.
BR-16	Content Moderation	Reviews and comments must be moderated to avoid inappropriate content.

WATER PARAMETER

Standard	Range	Researcher
pH	7 – 8.5	ngहीं
Oxygen	7 - 15 mg/l	ngहीं
Salt	0,1% – 0,3%	ngहीं
temperature	15 - 25 *C	ngहीं
NO ₂ ⁻	0 - 0.2 mg/l	ngहीं
NO ₃ ⁻	0 - 40 mg/l	ngहीं
PO ₄ ⁻	0 - 0.25	ngहीं

POND

Standard	Range	Researcher
Pond Size	3 - 33 m ²	Thịnh
Pond Depth	0 - 3 m	Thịnh
Pond Volume	> 1 m ³	Thịnh

Pond Drains	1 - 2	Thịnh
Pond Aeration Capacity	Volume * 1.5 or Volume * 2	Thịnh
Pond Capacity	PC = 1 m3 * number of fish	Thịnh

Koi Fish

Old Fish	35 -126cm (recorded)	THỊNH
Old Age (Weight)	1kg - 15kg+	THỊNH
Old Age (Age)	more than 25 months / 0.85	THỊNH
Middle Age (Length)	25-35cm	THỊNH
Middle Age (Weight)	200-500 grams	THỊNH
Middle Age	13 months - 24 months/ 0.9	THỊNH
Young Age (Length)	0-20cm	THỊNH
Young Age (Weight)	80-150 grams	THỊNH
Young Age (Age	under 12 month / 1.	
Calculate Weight	Weight = length*length*length * bodyShape Factor * Age factor * 0.0001	THỊNH
Weight by owner input	Input weight	THỊNH
Food Amount	Young Age :2 - 3% body weight Middle Age : 1.5 - 2% Old Age : 1 - 1.5%	THỊNH

Species	Gosanke : +Kohaku (white koi with red pattern) +Taisho Sanke: (white koi with red and black) +Showa Sanshoku:: (black koi with red and white patterns) Utsuri: +Hi Utsuri (black and red) +Shiro Utsuri (white) +ki Utsuri (yellow) + Utsurimono (white and black dot) Tancho : 1 red dot on head (ez aim) - rare Asagi: muted blue-grey and bright red markings Butterfly Koi : symmetric both side like a butterfly Ogon : 1 solid colour (Platinum Ogon are solid, shiny silver-white in color, while yamabuki Ogon exist in warmer shades like gold, yellow, or bronze)	THỊNH
Body shape factor	Input slim/ normal / heavy (1.5 , 1.7 , 2.0)	THỊNH
Color	input	THỊNH
Gender	Input male/ female	THỊNH

S

Bussines rule :

The weight of koi fish depends on length, body shape and age coefficient. Other information such as color, gender, and species will not affect the weight of the fish much, so we can ignore it..

Weight = length*length*length * bodyShape Factor * Age factor * 0.0001

Length : input
 bodyShapeFactor : slim(1,5) normal (1.7) heavy (2)
 Age Factor : young(1) adult(0.9) old(0.85)

The amount of koi food depends mainly on the body quality and age coefficient.
 Formula

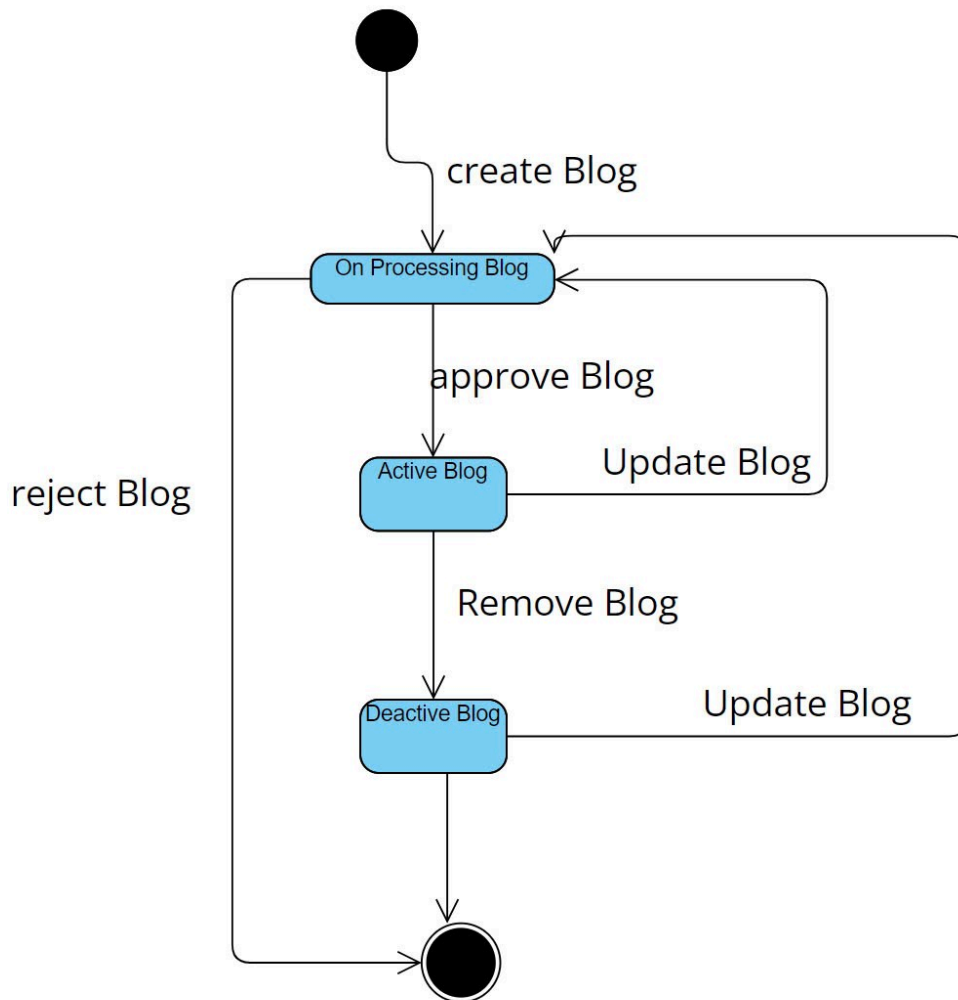
Food Amount = Age Factor * Weight * Food percentage
 Food percentage Age: young(2%-3%) adult(1.5%-2%) old(1%-1.5%)

For a **30 cm** long female koi fish, with a **Body Shape Factor of 2.0**, aged **6 months**:

- **Weight** = $30^3 \times 2.0 \times 1.0 \times 0.0001 = 540 \text{ grams}$
 $30^3 \times 2.0 \times 1.0 \times 0.0001 = 540 \text{ grams}$
- **Daily Food** = $540 \times 0.02 = 10.8 \text{ grams}$
 $540 \times 0.02 = 10.8 \text{ grams}$

4. State Machine Diagram

4.1 Blog Status



4.2 User Status

