VIETNAM INTERNATIONAL UNIVERSITY - HO CHI MINH CITY

INTERNATIONAL UNIVERSITY

WEB APPLICATION DEVELOPMENT PROJECT

RESTAURANT WEB

By

Đỗ Gia Thụy - Student ID: ITCSIU21237

Lê Huỳnh Minh Triết - Student ID: ITITIU13086

Phan Ngọc Hưng - Student ID: ITITIU13026

Advisor: Dr. Nguyen Van Sinh

A report submitted to the School of Computer Science and Engineering in partial fulfillment of the requirements for the Final Project in Web Application Development course - 2024

Ho Chi Minh city, Vietnam, 2024

Table of Contents

I. INTRODUCTION

This section briefly introduces the background information of our software development team – 3T. More importantly, the basic information about our project is also mentioned in this part. Furthermore, the constraints during our project are included at the end of this section.

1. ABOUT US

Company name: 3T

Team name: HKT Web Application Solution

Business: TTT Restaurant Website

Customer: Haidilao International Holding Ltd.

Contact: 1B Vo Van Ngan, Linh Trung ward, Thu Duc district, HCMC, Vietnam

Email: 3T@hcmiu.com

Phone number: +84 (08) 3823 4567

Figure 1

2. THE PRODUCT'S INFORMATION

In today's digital era, the restaurant industry is increasingly relying on technology to improve customer experience and streamline operations. At 3T Company, we specialize in creating innovative web solutions, and we are proud to introduce our latest offering: a comprehensive website platform designed specifically to manage and optimize restaurant operations.

Our web-based system provides a complete solution for managing all aspects of a restaurant, from order taking and menu management to payment processing and kitchen coordination. The platform is designed to enhance both the customer and staff experience, making everyday operations smoother and more efficient.

3. WORK BREAKDOWN STRUCTURE

The structure of our project can be expressed in the ***Figure 3.1:***

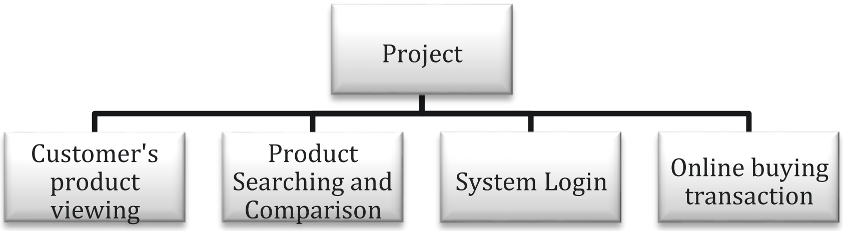

Project
Customer's product viewing
Product Searching and Comparison
System Login
Online buying transaction

Figure 3.1

In each branch of this tree, we also have subtrees which describes the tasks needed to be accomplished of each team member.

The tasks for the Customer's product viewing is described in the ***Figure 3.2***:

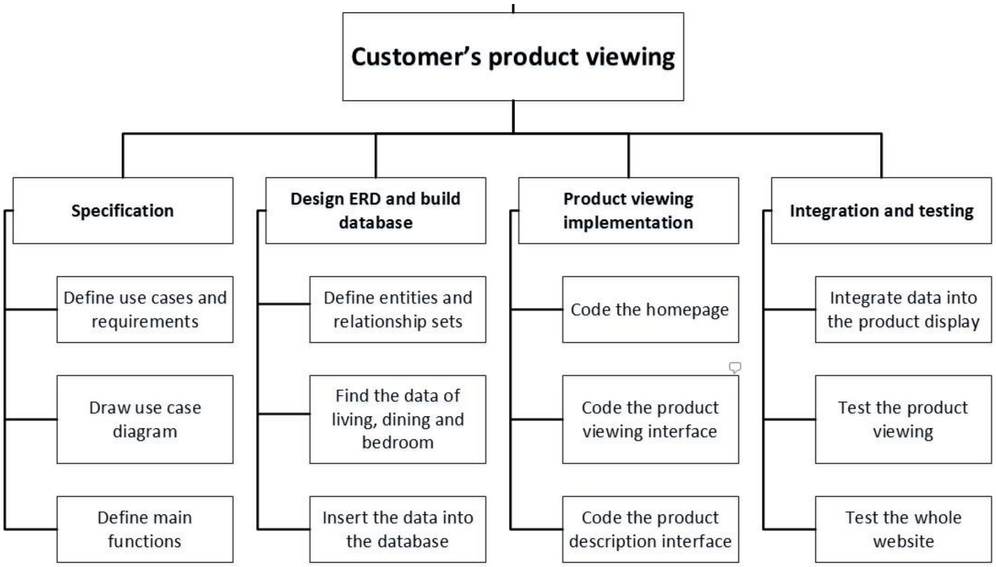

Customer's product viewing
Specification
Define use cases and requirements
Draw use case diagram
Define main functions
Design ERD and build database
Define entities and relationship sets
Find the data of living, dining and bedroom
Insert the data into the database
Product viewing implementation
Code the homepage
Code the product viewing interface
Code the product description interface
Integration and testing
Integrate data into the product display
Test the product viewing
Test the whole website

Figure 3.2

The tasks for the Product Searching and Comparison is described in the ***Figure 3.3***:

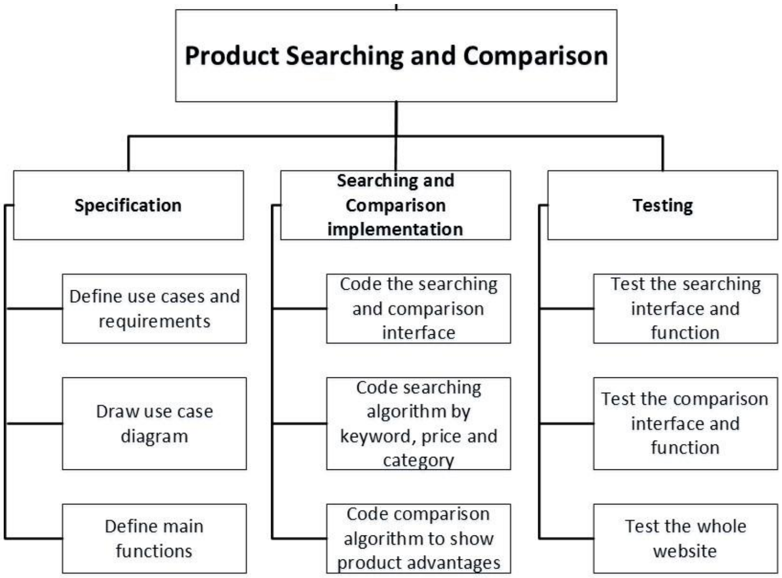

Product Searching and Comparison
Specification
Define use cases and requirements
Draw use case diagram
Define main functions
Searching and Comparison implementation
Code the searching and comparison interface
Code searching algorithm by keyword, price and category
Code comparison algorithm to show product advantages
Testing
Test the searching interface and function
Test the comparison interface and function
Test the whole website

Figure 3.3

The tasks for the System Login is described in the ***Figure 3.4***:

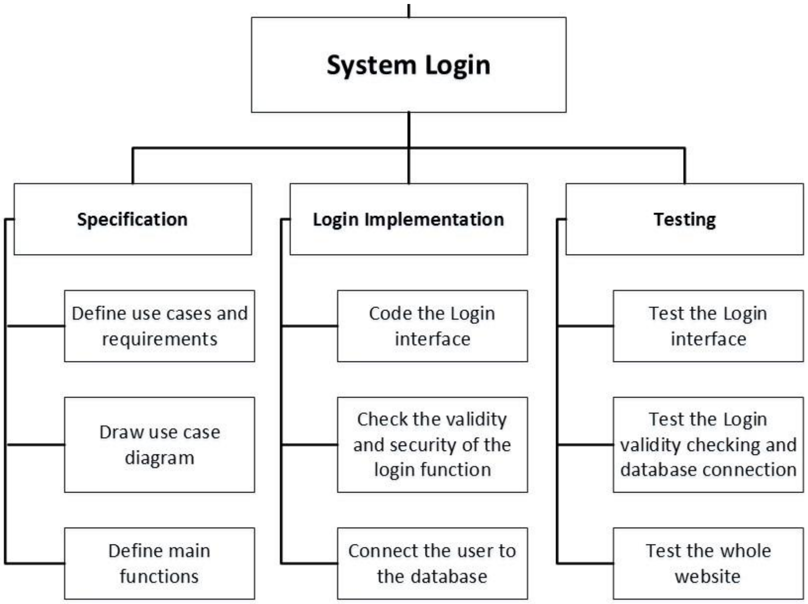

System Login
Specification
Define use cases and requirements
Draw use case diagram
Define main functions
Login Implementation
Code the Login interface
Check the validity and security of the login function
Connect the user to the database
Testing
Test the Login interface
Test the Login validity checking and database connection
Test the whole website

Figure 3.4

The tasks for the Online Buying Transaction is described in the ***Figure 3.5***:

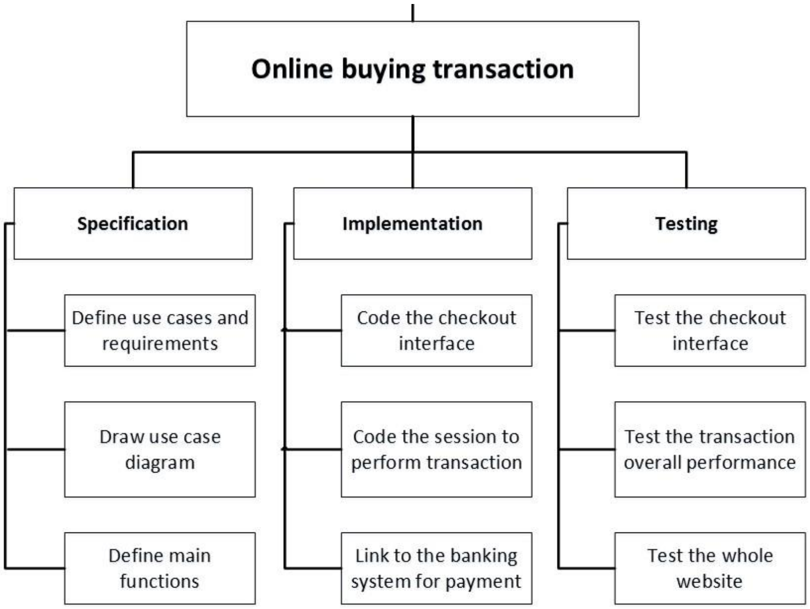

Online buying transaction
Specification
Implementation
Testing
Define use cases and requirements
Draw use case diagram
Define main functions
Code the checkout interface
Code the session to perform transaction
Link to the banking system for payment
Test the checkout interface
Test the transaction overall performance
Test the whole website

Figure 3.5

These tasks are expected to be completed and thoroughly tested after each iteration of the development process since we aim at applying the agile method rather than the traditional one which separates the steps.

4. DEVELOPMENT PROCESS

Agile

Agile is an approach to project management that centers around incremental and iterative steps to completing projects. The incremental parts of a project are carried out in short-term development cycles. The approach prioritizes quick delivery, adapting to change, and collaboration rather than top-down management and following a set plan.

In the Agile process, there is continuous feedback, allowing team members to adjust to challenges as they arise and stakeholders an opportunity to communicate consistently. Though originally created for software development, the Agile approach is now widely used in executing many different types of projects and in running organizations.

SCRUM

Scrum is an agile team collaboration framework commonly used in software development and other industries.

Scrum prescribes for teams to break work into goals to be completed within time-boxed iterations, called *sprints*. Each sprint is no longer than one month and commonly lasts two weeks. The scrum team assesses progress in time-boxed, stand-up meetings of up to 15 minutes, called *daily scrums*. At the end of the sprint, the team holds two further meetings: one sprint review to demonstrate the work for stakeholders and solicit feedback, and one internal sprint retrospective. A person in charge of a scrum team is typically called a scrum master.

Scrum's approach to product development involves bringing decision-making authority to an operational level. Unlike a sequential approach to product development, scrum is an iterative and incremental framework for product development. Scrum allows for continuous feedback and flexibility, requiring teams to self-organize by encouraging physical co-location or close online collaboration, and mandating frequent communication among all team members. The flexible approach of scrum is based in part on the notion of requirement volatility, that stakeholders will change their requirements as the project evolves.

Our method can be described in the ***Figure 4.1***:

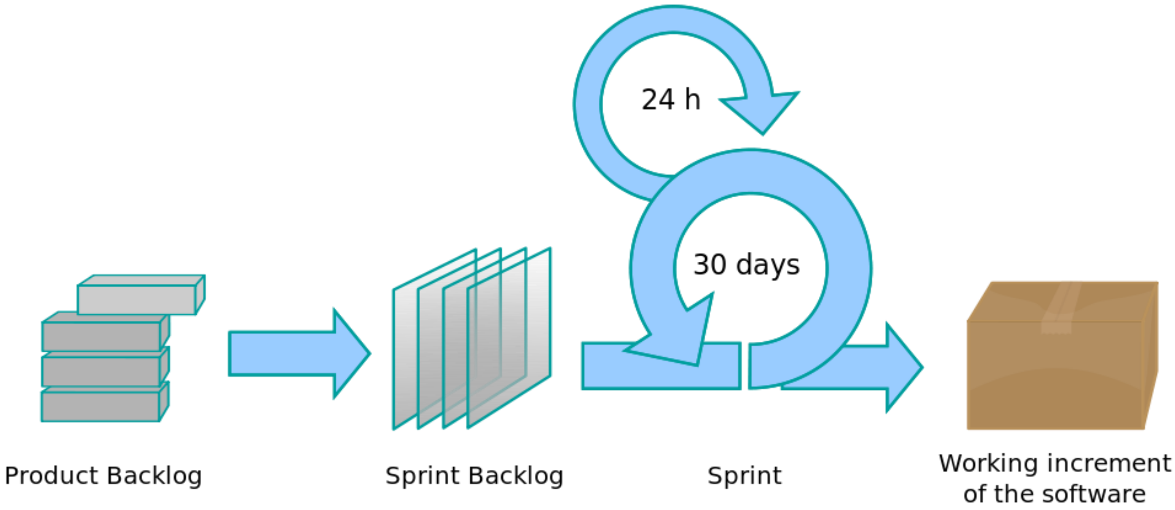


Figure 4.1

5. DEVELOPMENT ENVIRONMENT

Since this is a web-based product, the project is conducted using some Web Design and Programming Language under the model of MVC.

The following Programming Languages are used within our system:

1. **HTML:** to create an outline of the webpages
2. **CSS:** to design the looks of the webpages
3. **React.js**: Handles dynamic content, animations, and real-time updates on the frontend.

4. **Node.js**: Processes customer transactions and handles requests using **Express.js**.
5. **Express.js**: Manages backend logic, processes form data, and serves dynamic content.
6. **MySQL:** to store the database of the system

The code are implemented according to the MVC model in the ***Figure 5.1***:

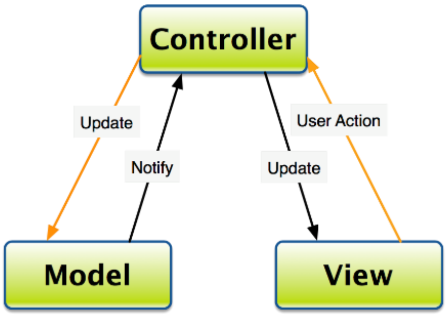


Figure 5.1

1. ***Model:*** the data of the system stored in the database and external system
2. ***View:*** the user interface
3. ***Control:*** the logics and algorithms used to develop a dynamic website with required functionalities

Moreover, our group has also utilized the cloud tools such as Google Docs to make it easier to incorporate the works of all the members in the team.

We also use UML tools to help us draw Entity Relationship Diagram as well Use Case needed in the project such as: Microsoft Visio, Edraw Max.

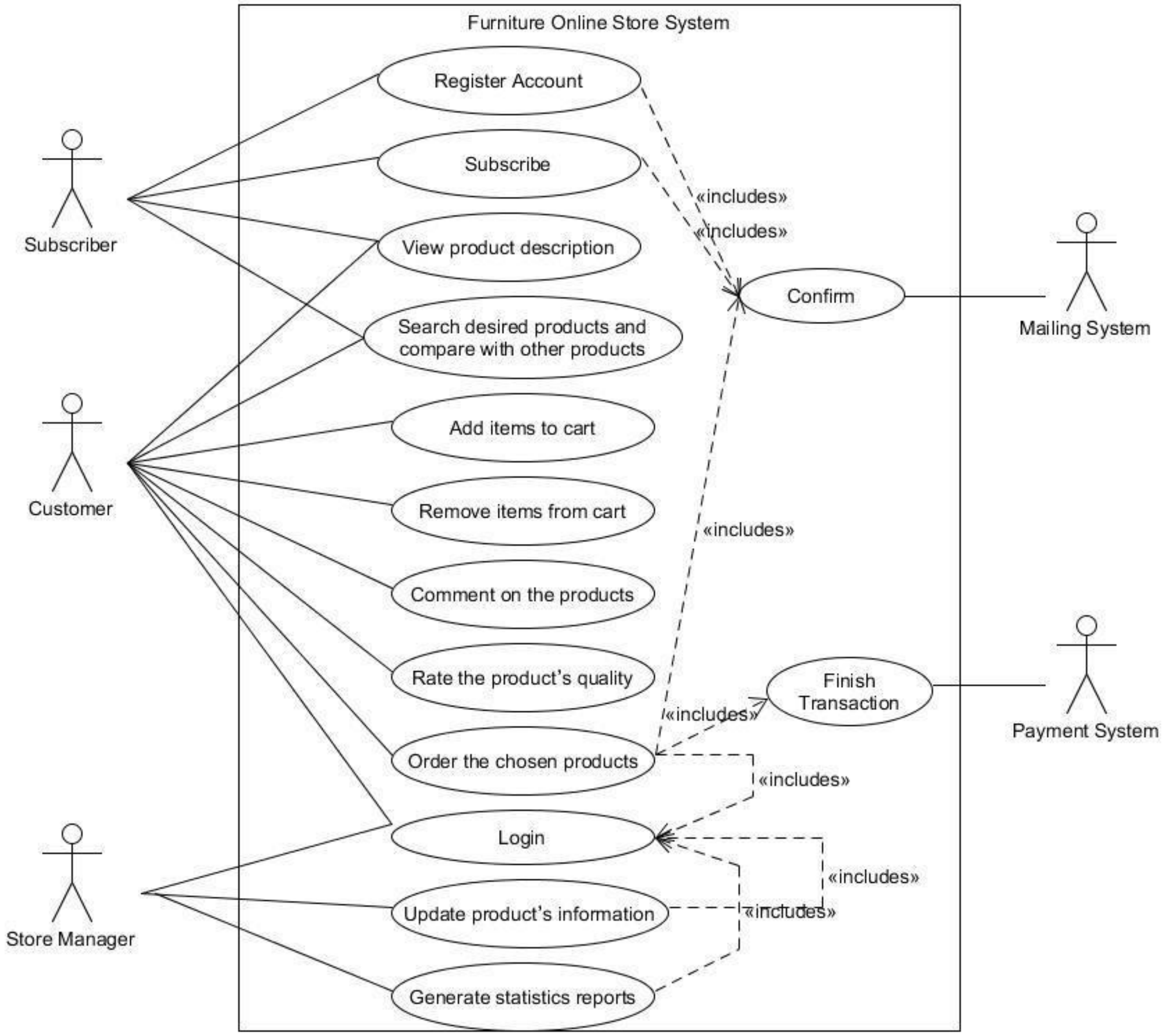
Moreover, we also make use of Microsoft Project - Management Tool - to keep track of the overall progress as well as each team member's work performance.

II. REQUIREMENT ANALYSIS AND DESIGN

This section briefly introduces the requirement analysis and design process. This version is created to provide the path for future implementation of the project. Based on this requirements specification, we will implement each function including all of the conditions as well as functional and non-functional requirements supplied by the customers. During our implementation, we will constantly revise and update the newer version so that we can always keep track of the progress of the project.

1. REQUIREMENT ANALYSIS

USE CASE DIAGRAM



Use Case 1:

Name: Log in the store system

Identifier *UC1*

Inputs:

1.User name

2.Password **Outputs:**

1. The home page with user's authorization [If success]
2. The login page [If fail]

Basic Course

Actor: User (Customer/Store Manager)	System
1. Open the login page	1.1. Display the login page
2. Enter user name and password	
3. Submit	3.1. Check the user's info. 3.2. If success, return the home page 3.3. Else return the login page

Precondition

1. User has an registered account of online store that is created earlier (ID and password)

Post condition

1. None

User story: As an online shop visitor, I want to log in the shop system so that I can use functions of the shop like buy products, gain the promotion, discuss about the products …

Use Case 2:

Name: Rate/Vote the quality of products

Identifier *UC2*

:

Input:

1. Vote value from 1 (worst) to 5 (best) stars by clicking the vote section **Outputs:**

1. New voting value of product in database
2. Product page with updated voting value

Basic Course

Actor: User (Customer/Store Manager)	System
1. Open that product detail page	1.1. Display the page with details of product
2. Vote the product by choosing the number of stars from 1 (worst) to 5 (best)	2.1. Display user's selection
3. Submit	3.1. Update the voting value in database. 3.2. Reload the page to show the updated voting value from user

Preconditions

1. User must log in the store system earlier
2. User opens the product page with the details of a certain product

Post condition

1. None

User Story:

As a shop customer, I want to vote or rate the product to recommend other customers to buy that products

Use Case 3

Name: Comment and Discussion about the products

Identifier *UC3*

Input:

1. Comment of user in form of text **Outputs:**

1. Updated comment storage in database
2. Reloaded product page that show the posted comment from user

Basic Course

Actor: User (Customer/Store Manager)	System
1. Open that product detail page	1.1. Display the page with details of product
2. Click on "Reply" button	2.1. Display the text area to get input from user
3. Typing the comment in the text area	
4. Submit	4.1. Insert the comment instance with content input from user into database 4.2. Reload the page to show the posted comment that user has just posted

Precondition

1. User must log in the store system earlier

Post condition

1. None **User**

Story:

As a shop customer, I want to discuss about the products to gain more detailed information from the shop manager as well as other customers

:

Use Case 4

Name: Search the product by name, price, category, functionalities

Identifier *UC4*

Inputs:

1.Searching keywords

2.Category where keyword will be searched **Output:**

1. The result page with products related to the searching keyword

Basic Course

Actor: User (Customer/Store Manager)	System
1. Access a page of online shop	1.1. Display the page
2. Click on the text area of search section	
3. Input the keyword used to search the product	
4. Select the searching section (name/price/category/functionalities)	4.1. Display the selected section
5. Submit	5.1. Sort out the products matching with the search condition from database 5.2. Load a result page and display the search result.

Precondition

1. None

Post condition

1. None

User story:

As a shop visitor, I want to search the product with search engine so that I can find out the appropriate product conveniently

Use Case

Use Case 5

Name: Register a new account of online store

Identifier *UC5*

Input:

1. Personal information from user **Outputs:**

1.	The home page	[If success]
2.	The register page	[If fail]

Basic Course

Actor: User (Customer/Store Manager)	System
1. Access a page of online shop	1.1. Display the page
2. Click on the “register” button	2.1. Display the register page
3. Input the personal information (ID, password, Full name, DOB, email…)	
4. Submit	4.1. Check the validity of account information 4.2. If success, display the page that remind user check the mailbox 4.3. If fail, return back to the register page

Preconditions

1. None

Post condition

1. None

User story:

As a visitor, I want to have a shop account so that I can take advantage of the offered function from the online shop

6:

Name: Add/Remove desired products to/from wish list and shopping cart

:

Identifier *UC6*

Input:

1. None **Output:**

1. Display message that user has successfully added items to shopping cart or wish list **Basic Course**

Actor: member	System
1. Click the “Add to Cart” button	1.1. If logged in, add items to member’s shopping cart 1.2. Else display “Must log in” message
2. Click the “Add to Wish List” button	2.1. If logged in, add items to member’s wish list 2.2. Else display “Must log in” message
3. Click the “My Shopping Cart” image	3.1. Open the “Your Shopping Cart” page 3.2. If logged in, display all the products added 3.3. Else display “Must log in” message
4. Click “Remove from Cart” of item	4.1. Remove item from Shopping Cart
5. Input new quantity for the selected item	5.1. Update the quantity for the selected item in theshopping cart
6. Click the “My Wish List” image	6.1. Open the “Your Wish List” page 6.2. If logged in, display all the products added 6.3. Else display “Must log in” message
7. Click “Remove item” in wish list	7.1. Remove item from Wish List
8. Click the “Add to Cart” button in wish list	8.1. Add items to member’s shopping cart

Preconditions

1.User has logged in

2.User is currently in the product viewing or product description page **Post conditions**

1. Items added to or removed from the shopping cart or wish list
2. Database updated

User story 1:

As a user, I want to add the selected item to cart so that I can check them out later with the quantity I want.

User story 2:

As a user, I want to remove the item from my shopping cart so that I can choose others which suit me better.

User story 3:

As a user, I want to alter the quantity of the item so that I can buy just enough what I need

User story 4:

As a user, I want to add item into my wish list so that I can save them for future references.

User story 5:

As a user, I want to remove some item from my wish list so that I can buy them now or I can add new items which appeals me more.

User story 6:

As a user, I want to add the product into my wish list to my shopping cart so that I can check out the items I need.

7:

Name: Perform buying transaction and checkout

Identifier *UC7*

Input: None
Output: Display message that user has successfully purchased products

Basic Course

Actor: member	System
1. Click Checkout button	1.1. Ask user to log in

	1.2. Navigate to Checkout page 1.3. Create a session 1.4. Display the user’s shopping cart
2. Click Next	2.1. Display the form for user to input their information, type of payment for the current transaction
3. Fill out the form and click Submit	3.1. Display message about successful transaction 3.2. Update the database 3.3. Destroy the current session 3.4. Navigate back to homepage
4. Click Update shopping cart	4.1. Destroy the current session 4.2. Navigate to Shopping cart page

Preconditions

1. Shopping cart is not empty
2. User has register an account

Post conditions

1.Mail confirmation is successfully sent

2.Database updated **User story 1:**

As a user, I want to check out the products in my shopping cart so that I can buy them online and get them shipped to my place.

User story 2:

As a user, I want to update my shopping cart so that I can add more items or remove the items that I find not so much necessary for now.

8:

Name: Get the purchase receipt via email

Identifier *UC8*

Input:

1. User’s email **Output:**

1. Display message that email has been successfully sent

Basic Course

Actor: member	System

Use Case

1. Click Done button	1.1. Take the user's all information and combine in one PDF file 1.2. Attach this file and send an email to the provided email 1.3. Display message that email has been sent

Preconditions

1. The buying transaction must be completed
2. User's email must be provided

Post conditions

1. Email must have all information of the user
2. Ensure the email Sent to the correct destination

User story:

As a user, I want to receive a receipt via email after I have purchased from your store so that I can confirm what I have ordered and be acknowledged when the stuff will be shipped.

9:

Name: View user's profile

Identifier *UC9*

Input:

1. None **Output:**

1. Display the user's profile

Basic Course

Actor: member	System

1. Click My Profile button	1.1. If logged in, navigate to the User's Account page 1.2. Else ask the user to log in 1.3. If logged in, display the user's information as well as previous purchases and discounts

Preconditions

1. The user has registered an account
2. The user has logged in

Post condition

1. Display all information of the user

User story:

As a user, I want to view my profile so that I can change my personal information and view the discount I am offered by the store for being a loyal customer.

10:

Name: Update the product information and clearance sale-off Identifier: UC10 **Inputs**:

1. Product's Updated Information
2. Sale-off event's Description (option) **Output**:

1. Display result on the store's page **Basic Course:**

Actor: Store Manager	System
1. Login as store manager	1.1 System will resolve this step base on UC1
2. Request for store's database	2.1 System will display the store's database to the user on screen
3. Update the product's information	3.1 If nothing wrong occurs, update the information. 3.2 Else, notify the user of the error.
4. Create an sale events by picking out the items and update the new price for those particular items	4.1 If nothing wrong occurs, update the price for the item. 4.2 Else, notify the user of the error.

Use Case

5. Input for sale's event description	5.1 Display form for the user to input description text 5.2 Will show the event on the store's homepage
6. Submit the form	6.1 Update the database with input information

Precondition

1. The user has logged in as store manager

Post condition

2. Display all information of the user

User Stories: Store manager (user) wants to update the quantity of certain items because the store have just received the supplies from the suppliers. Store manager also want to create a sale event on some particular items so that they have to reduce the current price of those items. Store manager also create the description for sale event.

11:

Name: Generate reports of sales after each day/month/quarter/year and customers' interests

Identifier: UC11

Input:

1. Period of time of the sale

Output:

1. Reports of sale containing all products sale information during input period

Basic Course:

Actor: Store Manager	System
1. Login as store manager	1.1 System response similar as UC1

2. Request to generate report of sales for months/quarter/year	2.1 Summarize and produce report 2.2 Display on screen 2.3 Print out report

Pre-condition*:*

1. Valid period of time input

Post-condition*:* None

User Stories: As a store manager, I want to generate the report of sale of the store for months/quarters/year to generalize how well the business is doing as well as other important business factors to determine the store's activities.

12:

Name: View the product's information on the web page

Identifier: UC12

Input:

1. Click on the product after searching

Outputs:

1. Display the more detailed version of the product (description, in stock or out of stock, etc.)
2. Photos of the products taken from different perspectives
3. Rates and comments from different users

Basic course

Actor: Visitor	**System**
1. Click on the searched product	**1.1 Display the product description in more details**
2. View the comments in more details	**2.1 Expand the comments' pages and lists**

Pre-condition*:* Visitors searched the product and got the desire result.

Use Case

Post-condition*:* None

User Stories*:* As a visitor, I want to see the product in more details along with comments and ratings to determine if the product is of high quality, no fake, from prestigious brand or got any errors during the progress of production.

A. FUNCTIONAL REQUIREMENTS

Use Case 1: QR Code Ordering System

1. **The Scope of the Work**
   - This occurs in sprint [X] in the development process.
   - 3 tasks are needed for this function.
   - 15 hours of effort are needed for this function.
2. **The Scope of the Product**:
   This function enables customers to scan a QR code at their table and be redirected to the restaurant's online ordering system.
3. **Functional and Data Requirements**
   a. **Functional Requirements**
   - Shall generate a unique QR code for each table.
   - Shall direct customers to the corresponding table's ordering page upon scanning.
   - Shall allow customers to view the menu and place orders directly from the page.

   b. **Data Requirements**

   - Table-specific data (table ID, order details) will be stored in the database.

Use Case 2: Menu Management

1. **The Scope of the Work**
   - This occurs in sprint [X] in the development process.
   - 4 tasks are needed for this function.
   - 20 hours of effort are needed for this function.
2. **The Scope of the Product**:
   This function allows restaurant admins and staff to manage the menu, including adding, updating, or removing menu items.
3. **Functional and Data Requirements**
   a. **Functional Requirements**
   - Shall allow admins to add, edit, or delete menu items.
   - Shall allow menu items to be categorized (e.g., appetizers, main dishes, drinks).
   - Shall allow updates to item descriptions, prices, and images.
   - Shall display the menu with filtering and searching options for customers.

   b. **Data Requirements**

   - Menu item data (name, description, price, category, image) will be stored in the database.

Use Case 3: Ordering System

1. **The Scope of the Work**

- This occurs in sprint [X] in the development process.
- 5 tasks are needed for this function.
- 30 hours of effort are needed for this function.

2. **The Scope of the Product**:
   This function allows customers to browse the menu, select dishes, enter quantities, view estimated completion times, and add items to their orders.
3. **Functional and Data Requirements**

   a. **Functional Requirements**

   - Shall allow the customer to select dishes from a categorized list.
   - Shall allow the customer to enter the quantity of each selected dish.
   - Shall display the estimated completion time for the selected items.
   - Shall allow customers to add more items to their current order.
   - Shall automatically calculate the total order value as the user adds/removes items.
   - Shall generate an invoice with dish names, prices, and total order value.

   b. **Data Requirements**

   - Order data (dishes, quantities, total cost) will be stored in the database.
   - Customer order data will be linked to the corresponding table number and customer session.

Use Case 4: Kitchen Management

1. **The Scope of the Work**
   - This occurs in sprint [X] in the development process.
   - 4 tasks are needed for this function.
   - 25 hours of effort are needed for this function.
2. **The Scope of the Product**:
   This function allows kitchen staff to receive customer orders, track progress, and manage the cooking schedule.
3. **Functional and Data Requirements**

   a. **Functional Requirements**

   - Shall automatically send customer orders to the kitchen.
   - Shall allow kitchen staff to mark orders as “in progress” or “ready.”
   - Shall allow the chef to prioritize orders based on estimated cooking time.
   - Shall display the list of orders and their current status.

   b. **Data Requirements**

   - Kitchen order data (order details, status) will be stored in the database.

Use Case 5: Food Delivery System

1. **The Scope of the Work**

- This occurs in sprint [X] in the development process.
- 3 tasks are needed for this function.
- 15 hours of effort are needed for this function.

2. **The Scope of the Product**:
   This function displays completed orders, showing the dish names and corresponding table numbers for delivery by staff.
3. **Functional and Data Requirements**
   a. **Functional Requirements**
   - Shall display a list of completed dishes with dish names and table numbers.
   - Shall allow staff to mark dishes as "delivered" once served.
   - Shall update the order status in real-time.

   b. **Data Requirements**

   - Delivery status data (dish name, table number, delivery status) will be stored in the database.

Use Case 6: Payment System

1. **The Scope of the Work**
   - This occurs in sprint [X] in the development process.
   - 6 tasks are needed for this function.
   - 40 hours of effort are needed for this function.
2. **The Scope of the Product**:
   This function allows customers to request payment, choose between cash or online payment options, and receive an electronic invoice.
3. **Functional and Data Requirements**
   a. **Functional Requirements**
   - Shall allow customers to request the bill.
   - Shall allow customers to pay via cash or online (QR code for online payment).
   - Shall generate an electronic invoice with order details.
   - Shall verify online payments before confirming the transaction.

   b. **Data Requirements**

   - Payment data (payment method, amount, invoice details) will be stored in the database.

Use Case 7: Food Status Notifications

1. **The Scope of the Work**
   - This occurs in sprint [X] in the development process.
   - 3 tasks are needed for this function.
   - 20 hours of effort are needed for this function.

2. **The Scope of the Product**:
   This function sends notifications to customers when their food is ready or if the estimated completion time changes.
3. **Functional and Data Requirements**
   a. **Functional Requirements**
   - Shall notify the customer when their food is ready.
   - Shall notify the customer if the estimated completion time is updated.
   - Shall update the notification status in real-time.

   b. **Data Requirements**

   - Notification data (order status, estimated time) will be stored in the database.

Use Case 8: Login/Signup System

1. **The Scope of the Work**
   - This occurs in sprint [X] in the development process.
   - 4 tasks are needed for this function.
   - 15 hours of effort are needed for this function.
2. **The Scope of the Product**:
   This function enables users (customers, staff, admins) to register accounts and log in.
3. **Functional and Data Requirements**
   a. **Functional Requirements**
   - Shall allow users to register an account.
   - Shall allow users to log in with email and password.
   - Shall support password recovery.
   - Shall differentiate between user roles (customer, staff, admin).

   b. **Data Requirements**

   - User data (email, password, role) will be stored in the database.

Use Case 9: Admin Management

1. **The Scope of the Work**
   - This occurs in sprint [X] in the development process.
   - 3 tasks are needed for this function.
   - 20 hours of effort are needed for this function.
2. **The Scope of the Product**:
   This function allows admins to manage staff roles and access, and view system reports.
3. **Functional and Data Requirements**
   a. **Functional Requirements**
   - Shall allow admins to assign roles (staff, admin) to users.
   - Shall allow admins to view reports on sales and order history.

- Shall allow admins to manage menu items and prices.

b. **Data Requirements**

- Role and access data (user roles, permissions) will be stored in the database.

Use Case 10: Token Authentication System

1. **The Scope of the Work**
   - This occurs in sprint [X] in the development process.
   - 4 tasks are needed for this function.
   - 20 hours of effort are needed for this function.
2. **The Scope of the Product**:
   This function handles secure authentication for users using access tokens and refresh tokens.
3. **Functional and Data Requirements**
   a. **Functional Requirements**
   - Shall use JWT (JSON Web Tokens) for authentication.
   - Shall store access tokens in local storage and refresh tokens in cookies.
   - Shall allow token refresh when the access token expires.
   - Shall securely verify user identity for accessing restricted areas.

   b. **Data Requirements**

   - Token data (access tokens, refresh tokens) will be stored in the system for authentication purposes.

B. NON-FUNCTIONAL REQUIREMENTS

1. Operational requirements:

- Must make sure all of the components of the software operate in good manner
- Must keep the host running licensed so that it can handle all of the requests properly
- Must keep an administrator to frequently update the software as well as checking for the system’s errors

2. Legal requirements:

- Must cite all the components integrated into the system to avoid copyright violation
- Must keep the own-written source code confidential to avoid unauthorized use

3. Usability requirements:

- Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or component
- Well-structured user manuals
- Informative error messages – Error messages must state clear and might include hint to retrieve the solutions.
- Help facilities.
- Well-formed graphical user interfaces – easy to learn and navigate.
- Efficiency of use: goals are easy to accomplish and with few or no user error.

4. Humanity Requirements:

The users of the system typically have little to no background in using computers. Therefore, the development team must create comprehensive documentation that allows users to easily understand the system by simply reading it. The graphical user interfaces (GUIs) should be intuitively designed to ensure ease of learning and usability. Users should be able to quickly adapt to the system without needing extensive training. Additionally, the interface must be visually appealing and responsive to enhance the overall user experience.

5. Performance Requirements:

a. Response Time Requirements (How quickly the system responds to user requests)

I. The login response time for both customers and managers must be quick.
II. The system should update the database efficiently, with minimal delay when modifying multiple database tables simultaneously.
III. The system must process services swiftly with no noticeable delays. It should be capable of handling 10 different transactions at the same time.
IV. The website loading time must be fast, with all items correctly generated. Specifically, the system should load over 50 products per page in the shortest time possible.

b. Throughput Requirements (The volume of tasks the system can handle within a certain time)

I. The system must handle a large number of simultaneous requests from customers without delay. It should support over 1000 visitors browsing the store page at once.
II. The system must quickly process multiple database modifications without performance degradation.

c. Availability Requirements (Ensuring the system is available when users request it)

I. The system should utilize efficient memory management, including garbage collection in the programming language.

Description:

This is an e-commerce platform that may experience thousands of visitors daily, which places significant strain on the server if requests are not properly managed. The shopping cart system must be capable of processing orders simultaneously without delays or overload. The website must efficiently handle various types of requests from numerous customers on different operating systems, ensuring smooth operation without any interruptions.

6. Maintainability Requirements:

The **Online Store system** is designed to require continuous updates and changes. Since the users have limited knowledge about the internal workings of the system, the development team must implement it in a way that allows both users and future maintenance teams to easily manage and modify the system.

- Provide features that enable users to update and manage product and customer information.
- Write clear and well-structured source code, including detailed comments, to ensure better understanding and ease of modification.
- Design a straightforward and reliable database that facilitates easy maintenance and future updates.

7. Support Requirements

The system users, particularly the **Store Manager**, typically have limited technical knowledge regarding system maintenance. Therefore, in addition to developing the system, the development team should ensure that users can receive prompt support whenever needed.

- Provide a **hotline** for the store manager to report issues and receive immediate assistance.
- Offer a **remote maintenance tool** (such as TeamViewer or Remote Desktop) to provide direct support to the store manager.
- Perform regular system checks and maintenance to ensure stability, with updates scheduled monthly or every 2–3 months.

8. Security requirements:

- The web-based system shall ensure that data is safeguarded against unauthorized access.
- The system must maintain its integrity, protecting it from accidental or malicious damage.
- Login attempts shall be restricted: After five failed login attempts, access to the website will be blocked for 24 hours.
- Only the **administrator** shall have permission to make changes to system data.

- All modification events will be logged, with each log entry containing the following details: **date**, **time**, **user**, **action**, **object**, **prior value**, and **new value**.
- All communication between the system's data server and clients must be encrypted using secure protocols such as **SSL** or **SSH** for HTTP.
- A session should be initiated upon customer login, and a timeout should be applied after a period of inactivity.
- Customer information must be encrypted before being stored in the database.
- All system data, including product information and customer data (such as cookies and sessions), must be backed up every 24 hours, with one copy stored securely in a location separate from the primary system.
- A **privacy policy** must be provided to restrict third-party access to customer information.

9. Interface requirements:

- Describe all of the technical requirements that affect interfaces such as protocol management, scheduling, directory services, broadcasts, message types, error and buffer management, security, etc. Assign a unique ID number to each requirement. Some non-functional requirements of interface:

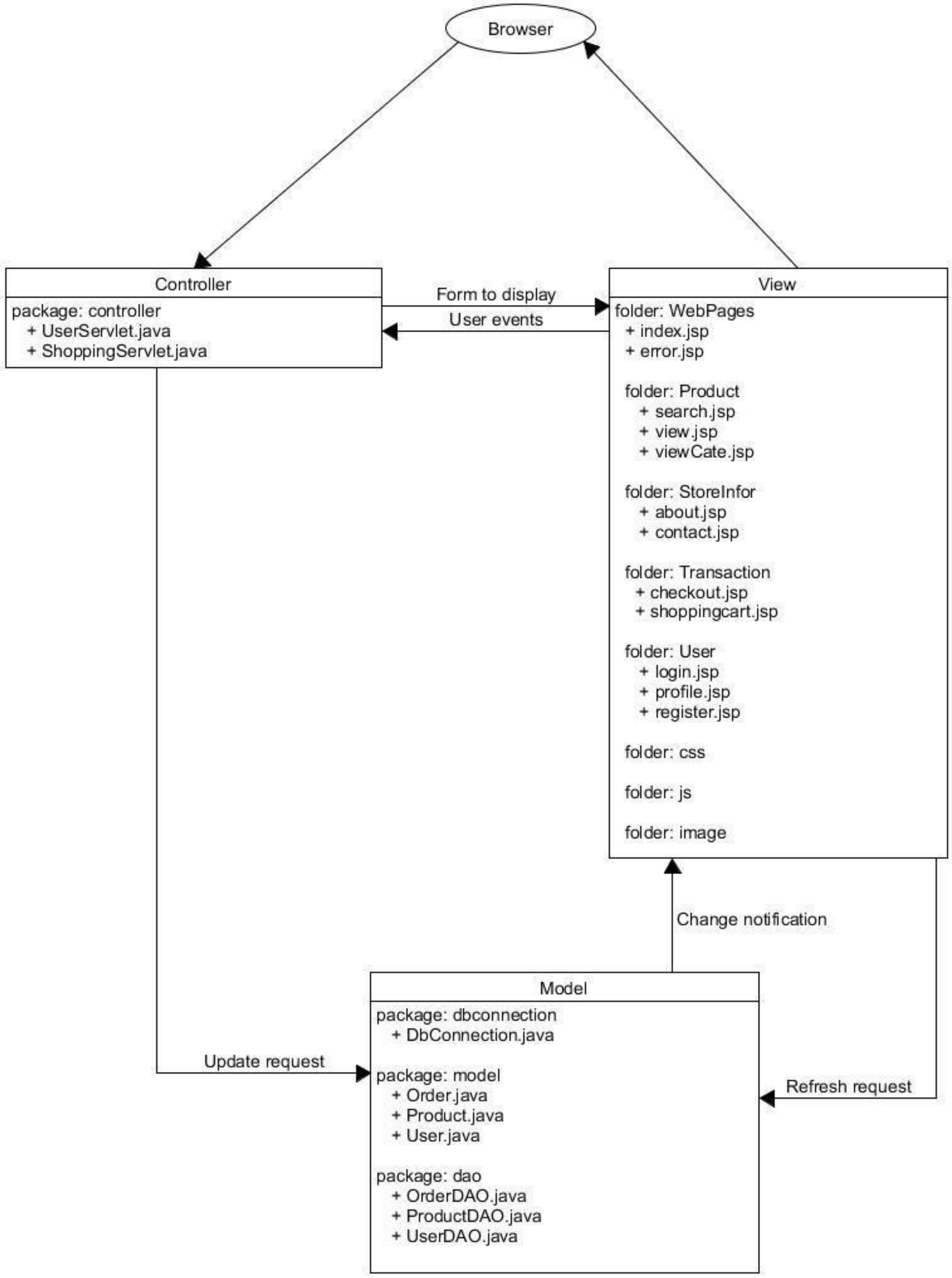
  +Buttons should include feature symbols equivalent to their functionalities .

  + Sales report included in every successful transaction.

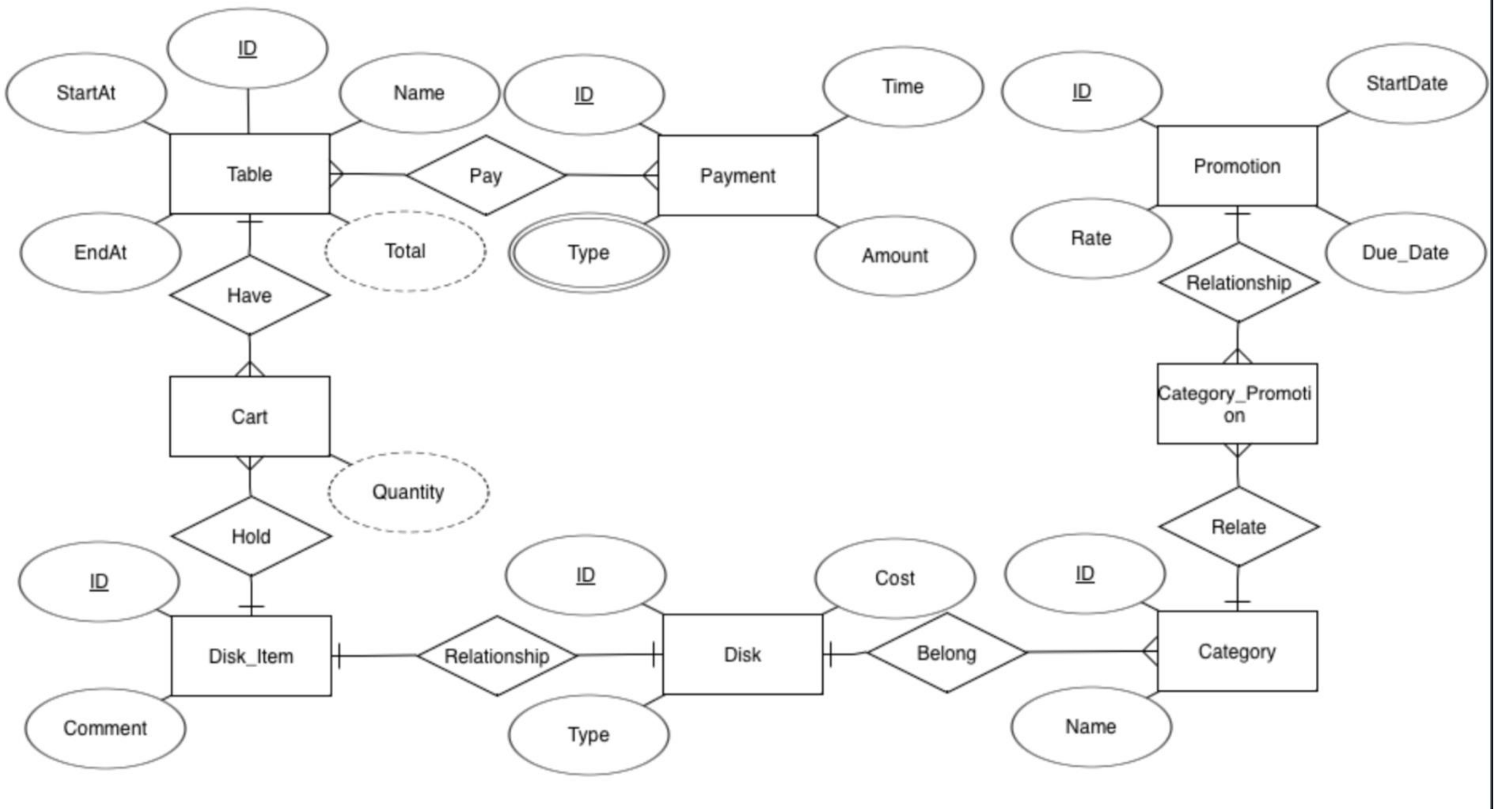
  + Include language options for native and foreign visitors/customers

2. DESIGN

System Architecture Model



Entity-Relationship Diagram (ERD)



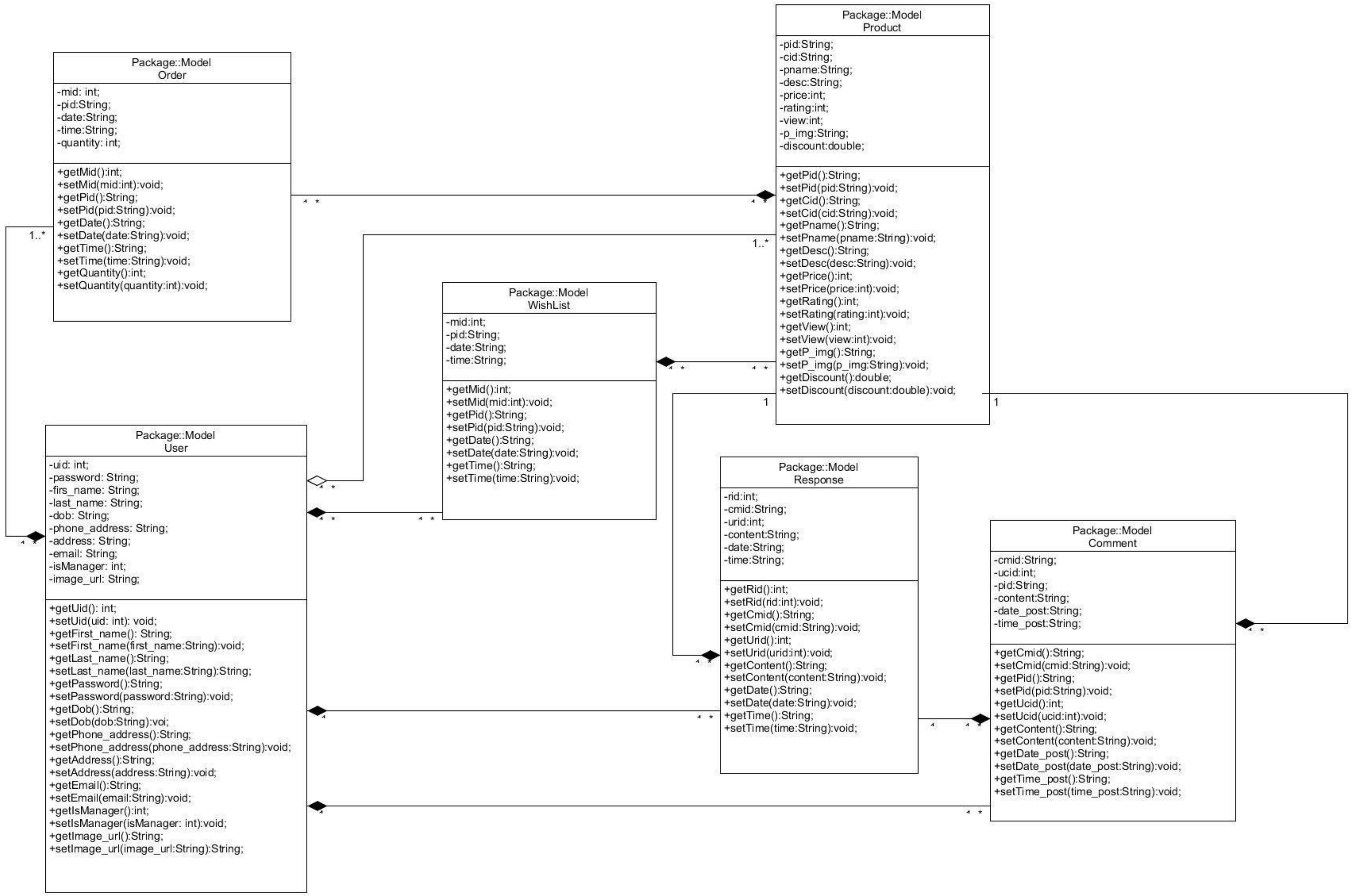
Entities

1. **Table**
   - Attributes: `ID`, `StartAt`, `EndAt`, `Name`, `Pay`
   - Weak Attribute: `Total`
2. **Payment**
   - Attributes: `ID`, `Time`, `Amount`, `Type`
3. **Promotion**
   - Attributes: `ID`, `StartDate`, `Due_Date`, `Rate`
4. **Cart**
   - Attribute: `Quantity`
5. **Disk_Item**
   - Attributes: `ID`, `Comment`, `Relationship`
6. **Disk**
   - Attributes: `ID`, `Cost`, `Type`
7. **Category**
   - Attributes: `ID`, `Name`
8. **Category_Promotion**

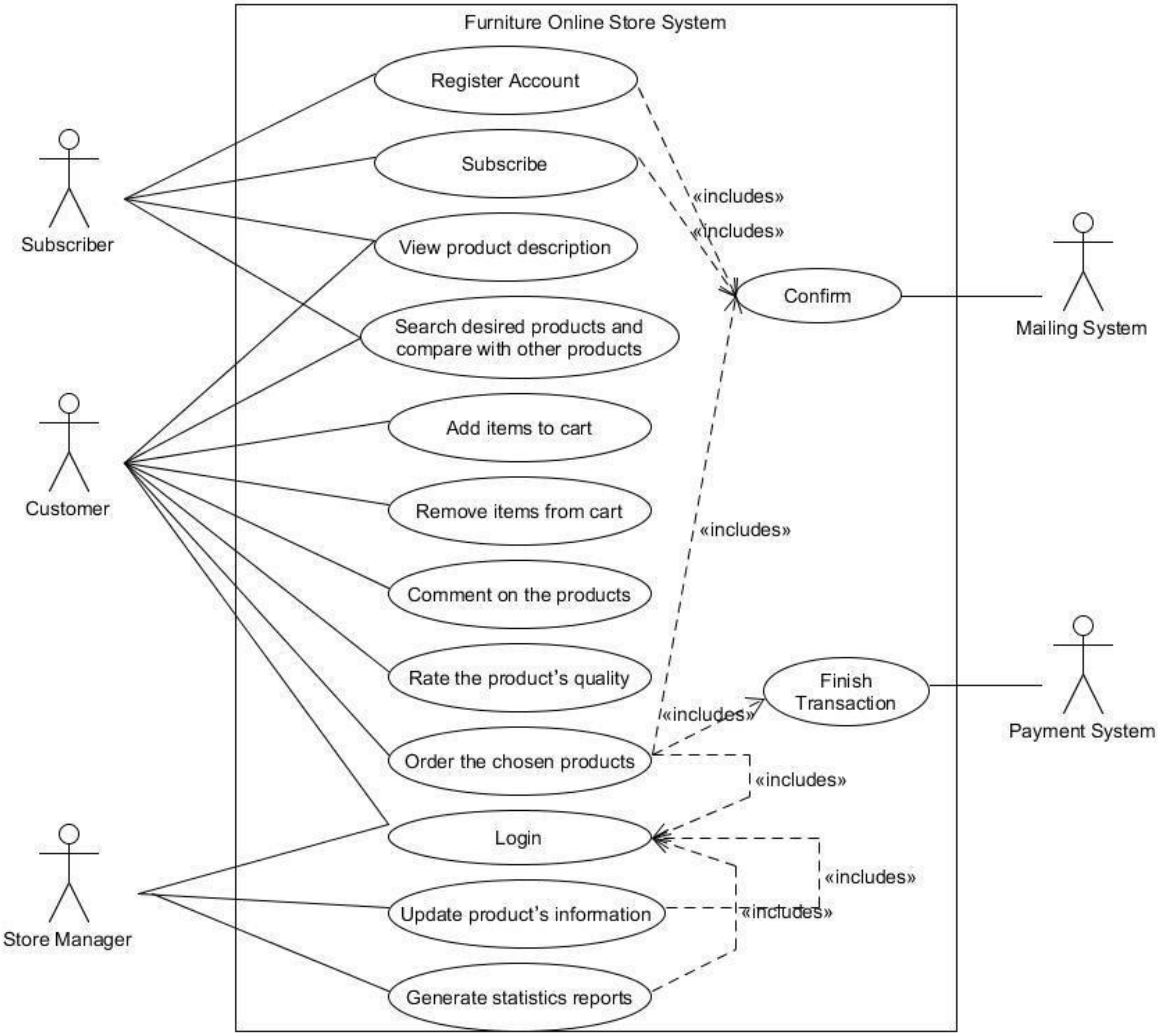
Relationships

1. **Have** (between Table and Cart)
2. **Hold** (between Cart and Disk_Item)
3. **Relationship** (between Disk_Item and Disk)
4. **Belong** (between Disk and Category)
5. **Relate** (between Category_Promotion and Category)
6. **Relationship** (between Promotion and Category_Promotion)
7. **Relationship** (between Promotion and Category_Promotion)

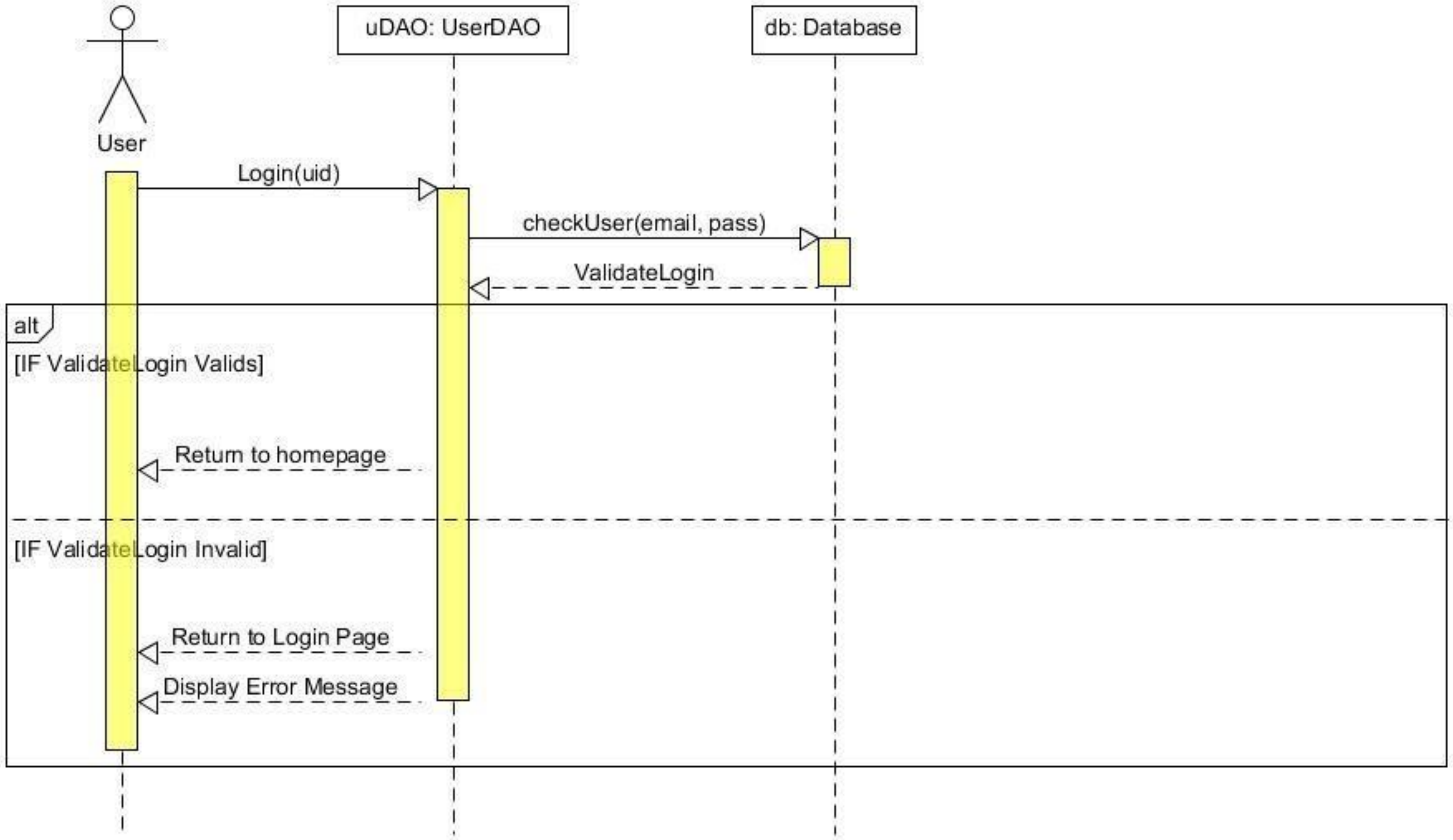
Class Diagram



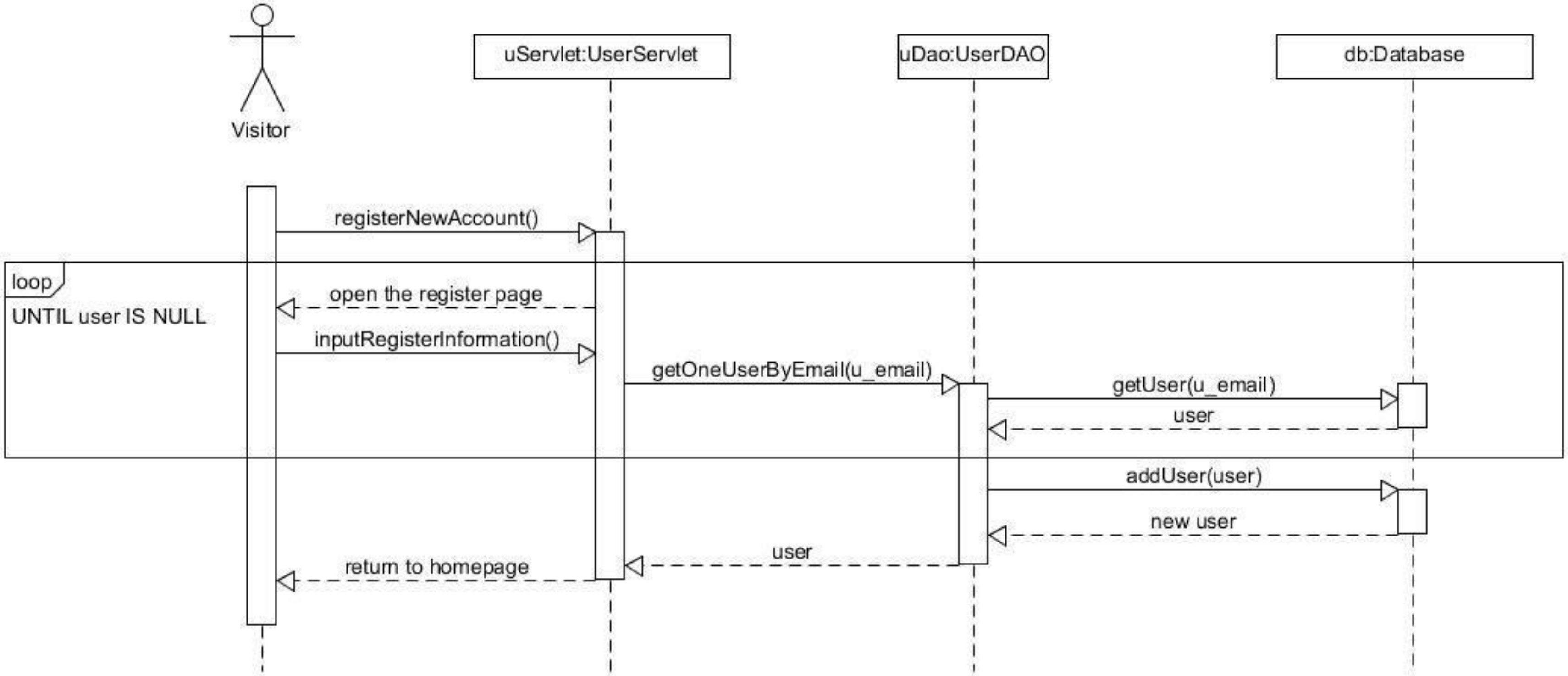
Use Case Diagram



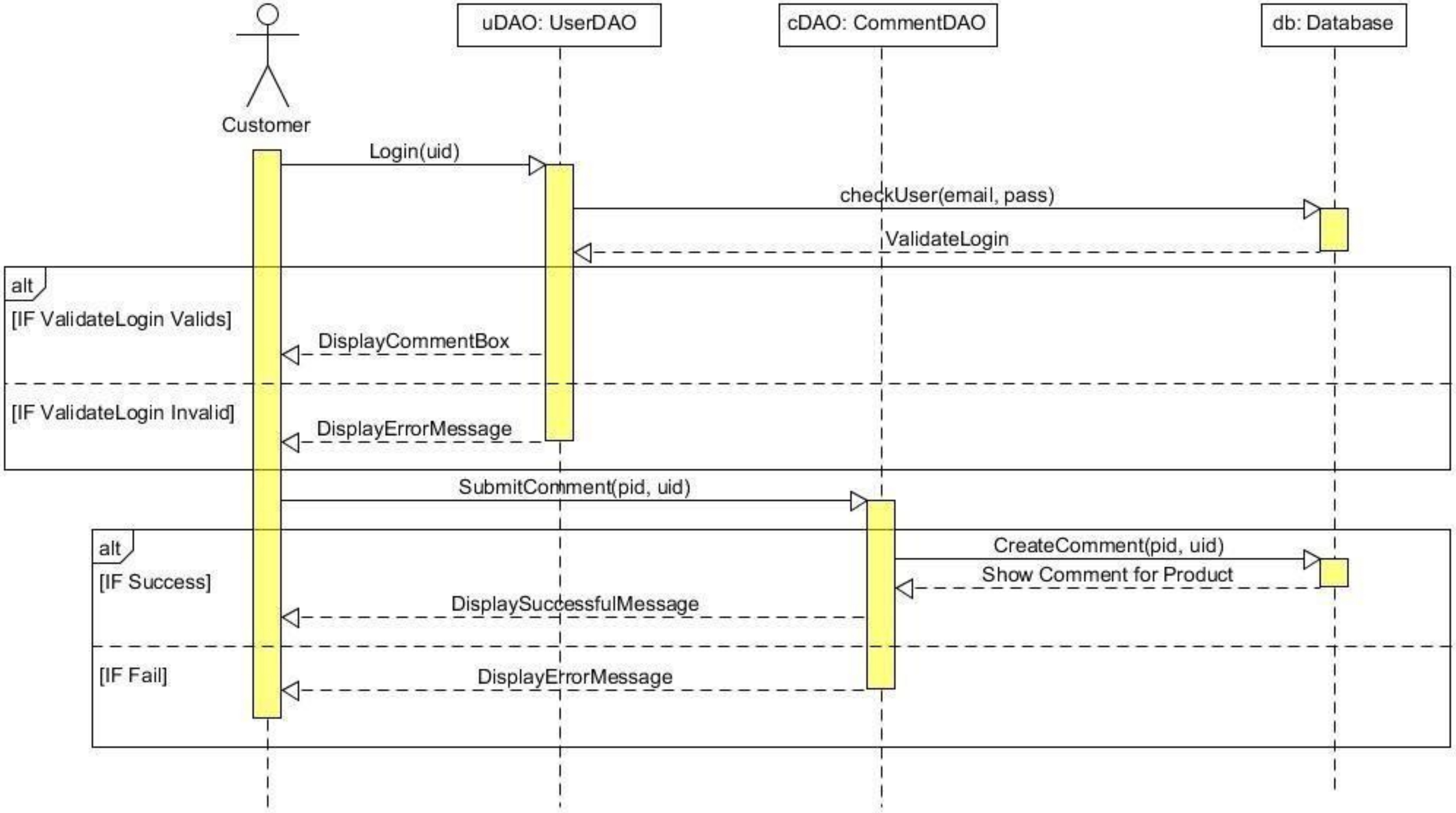
Use Case 1: Log in the store system



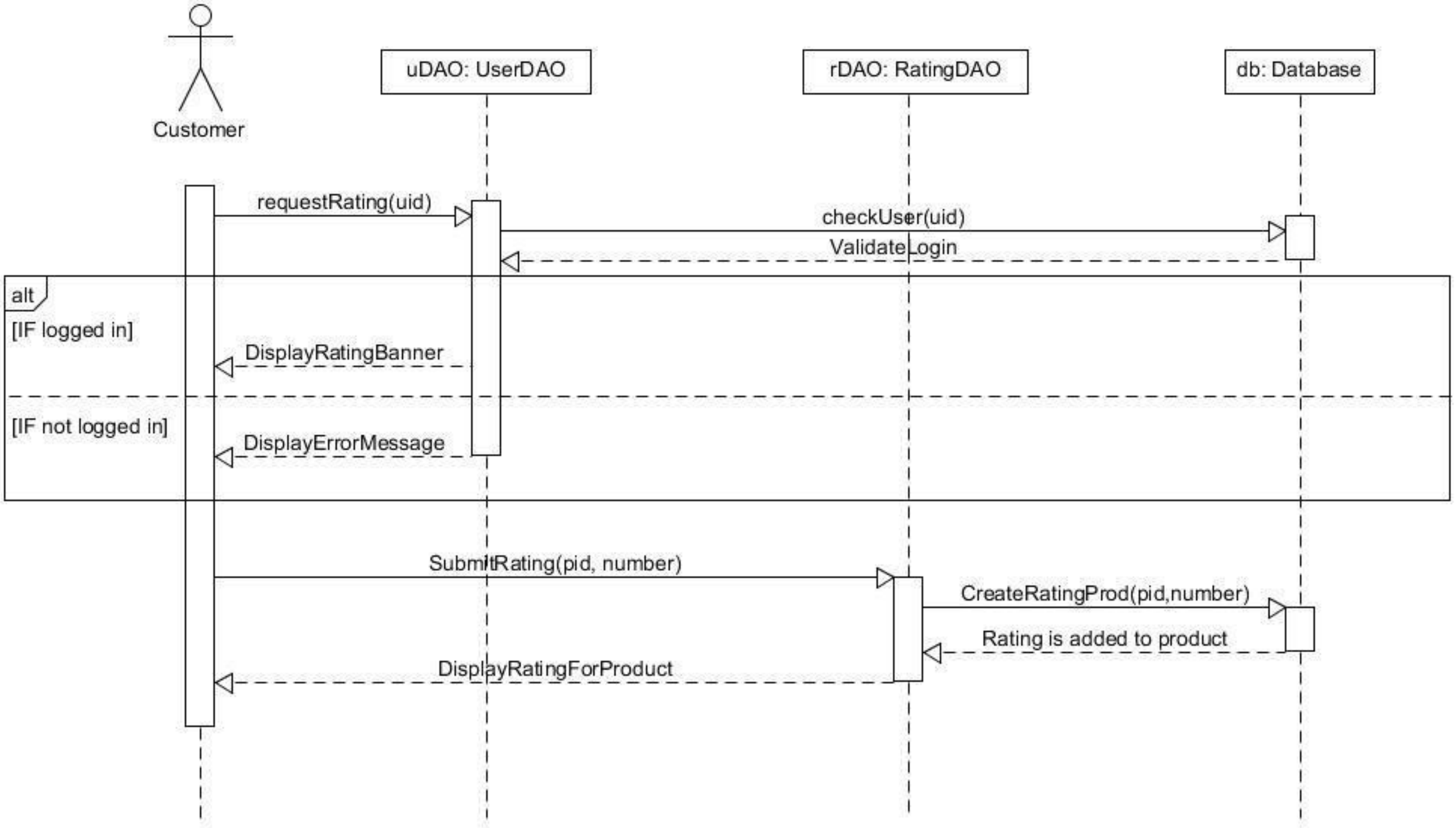
Use Case 2: Create a new store account



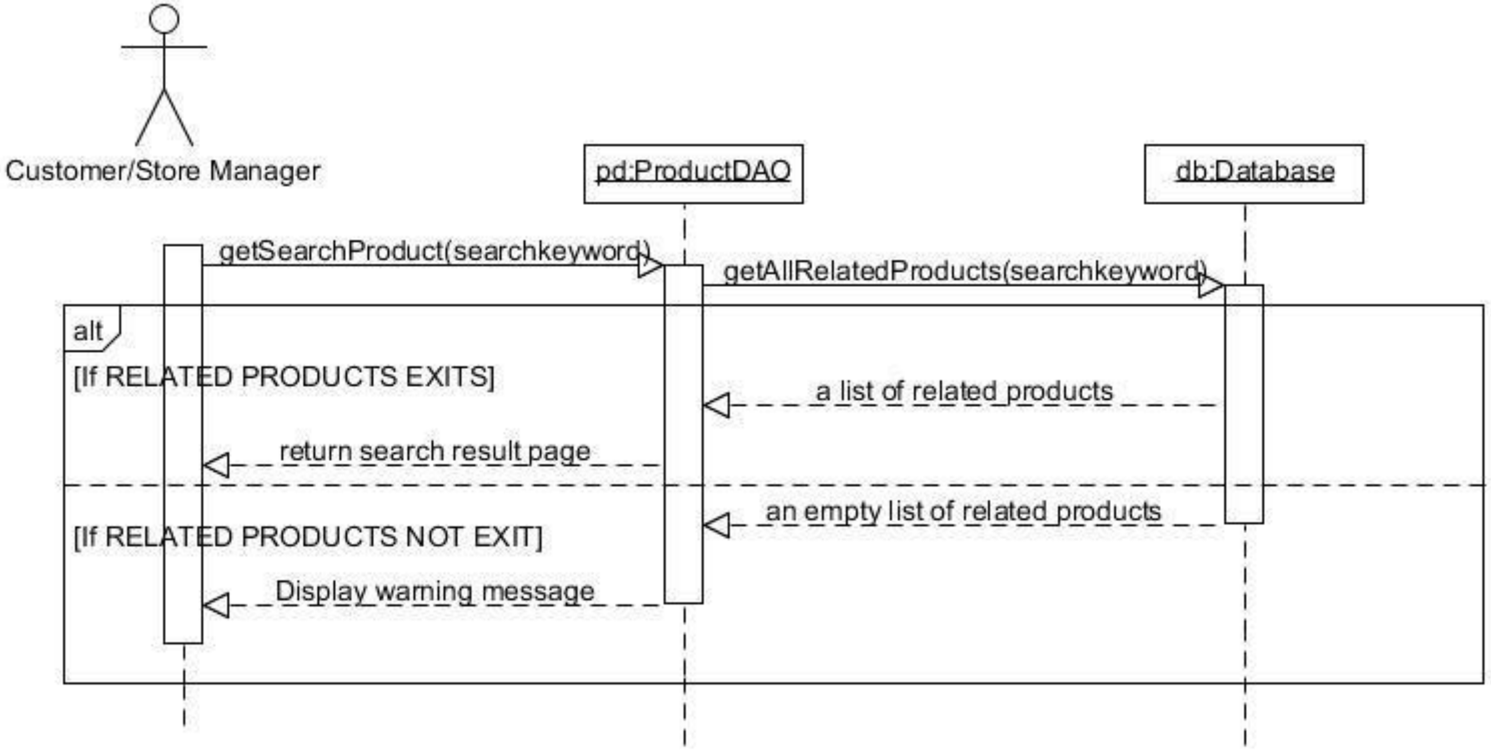
Use Case 3: Comment and Discussion about the products



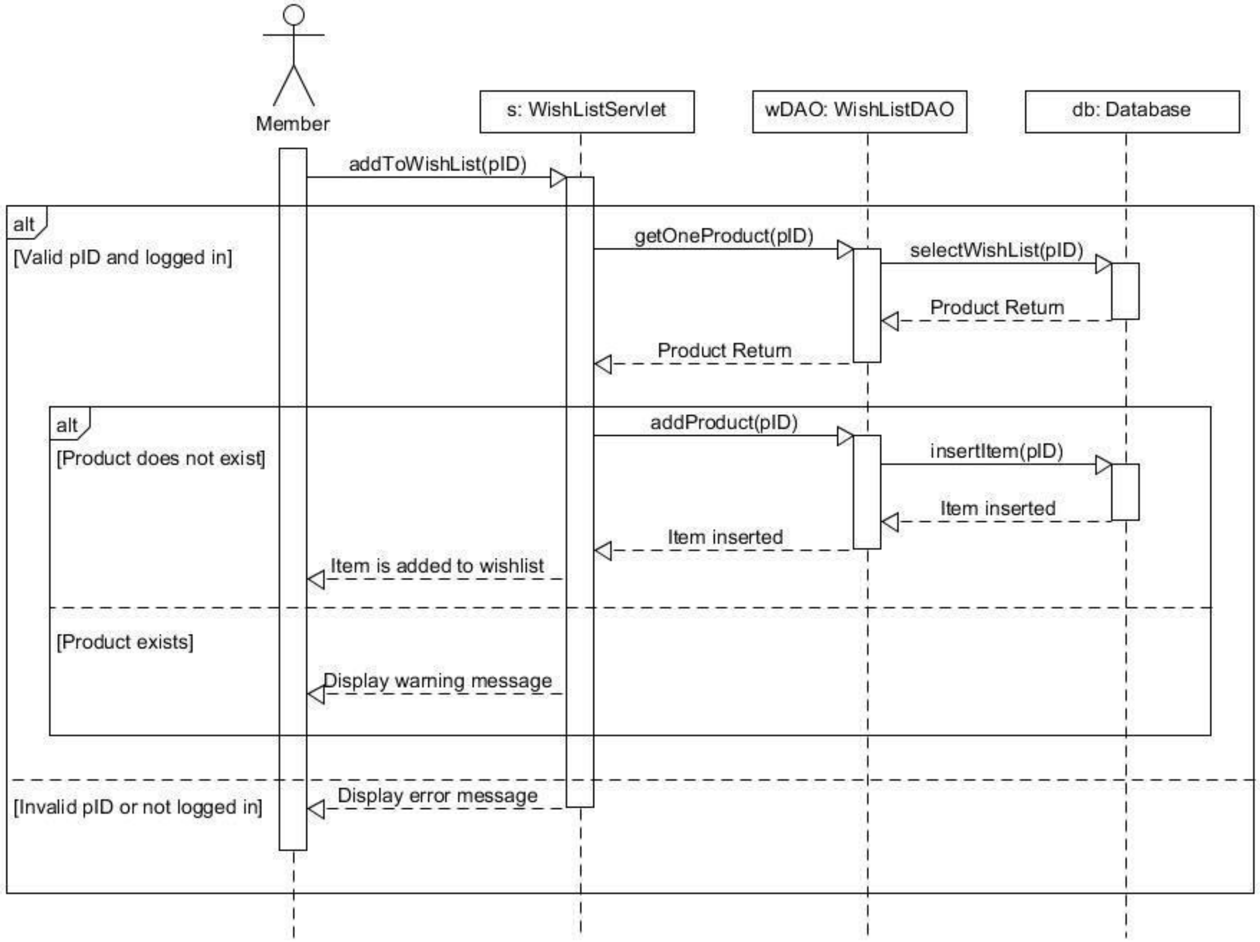
Use Case 4: Rate/Vote the quality of products



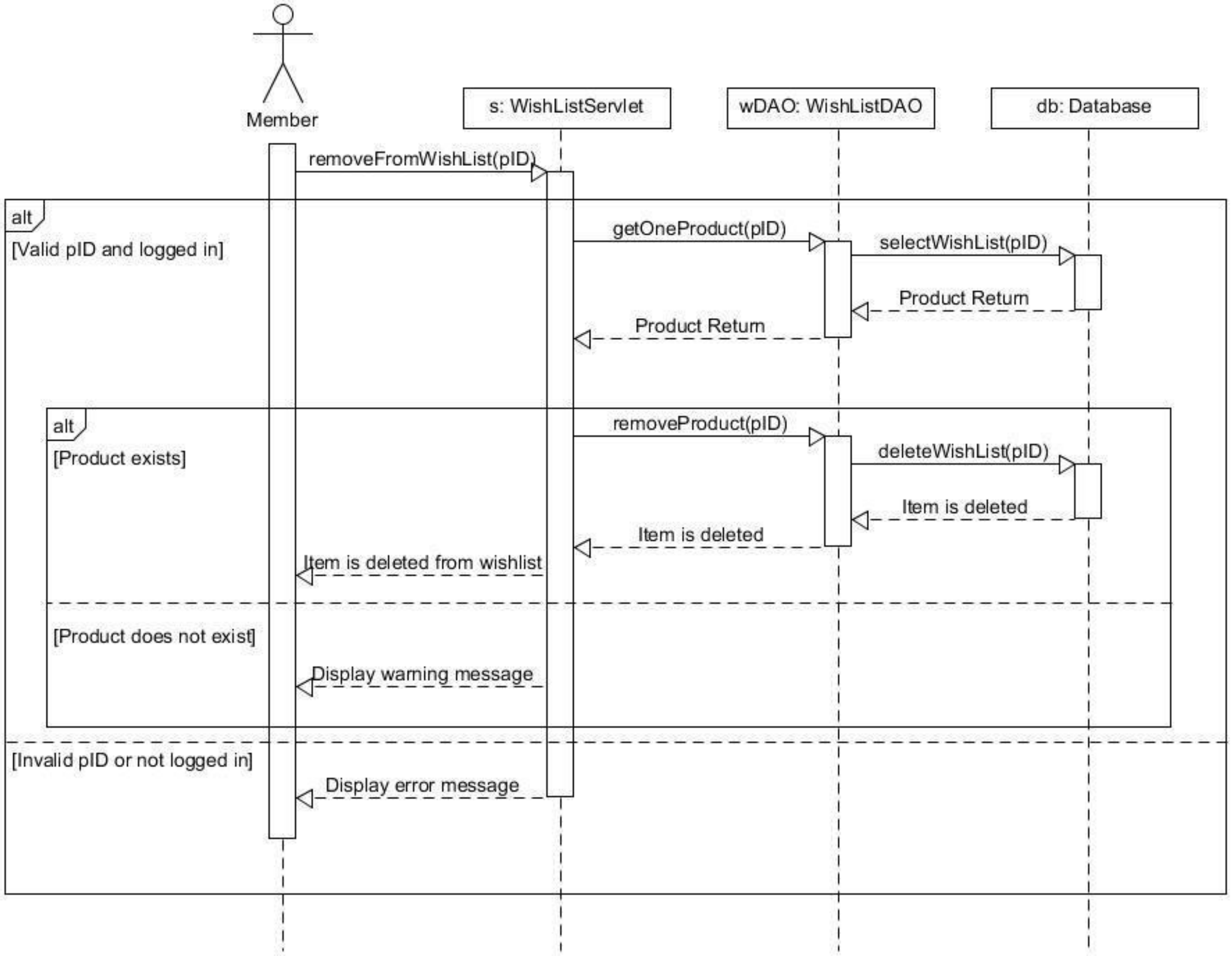
Use Case 5: Search the product by name, price, category, functionalities


Customer/Store Manager
pd:ProductDAO
db:Database
getSearchProduct(searchkeyword)
getAllRelatedProducts(searchkeyword)
alt
[If RELATED PRODUCTS EXITS]
a list of related products
return search result page
an empty list of related products
[If RELATED PRODUCTS NOT EXIT]
Display warning message

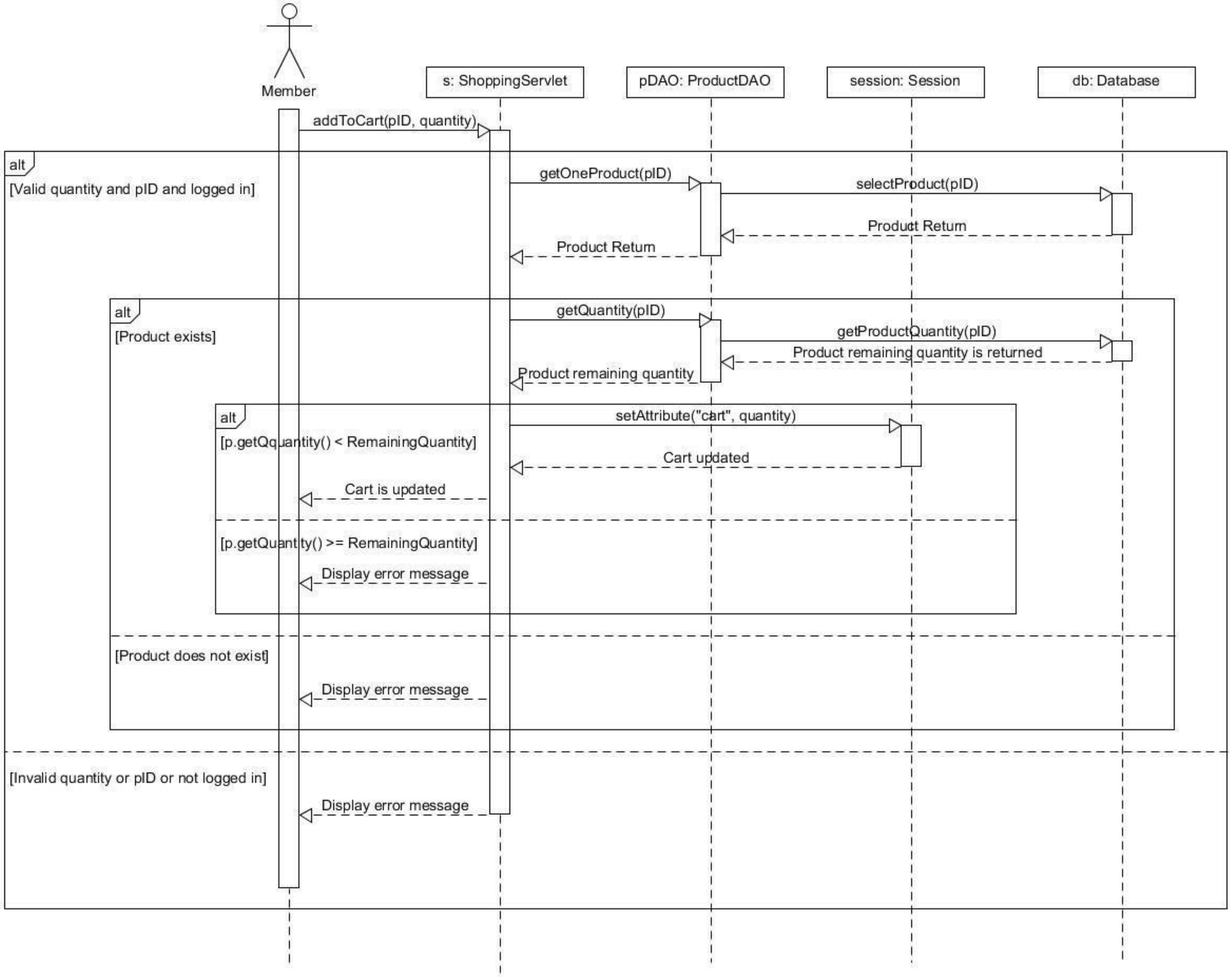
Use Case 6: Add/ Remove desired products to/from wish list and shopping cart *Use Case 6.1: Add to wish list*



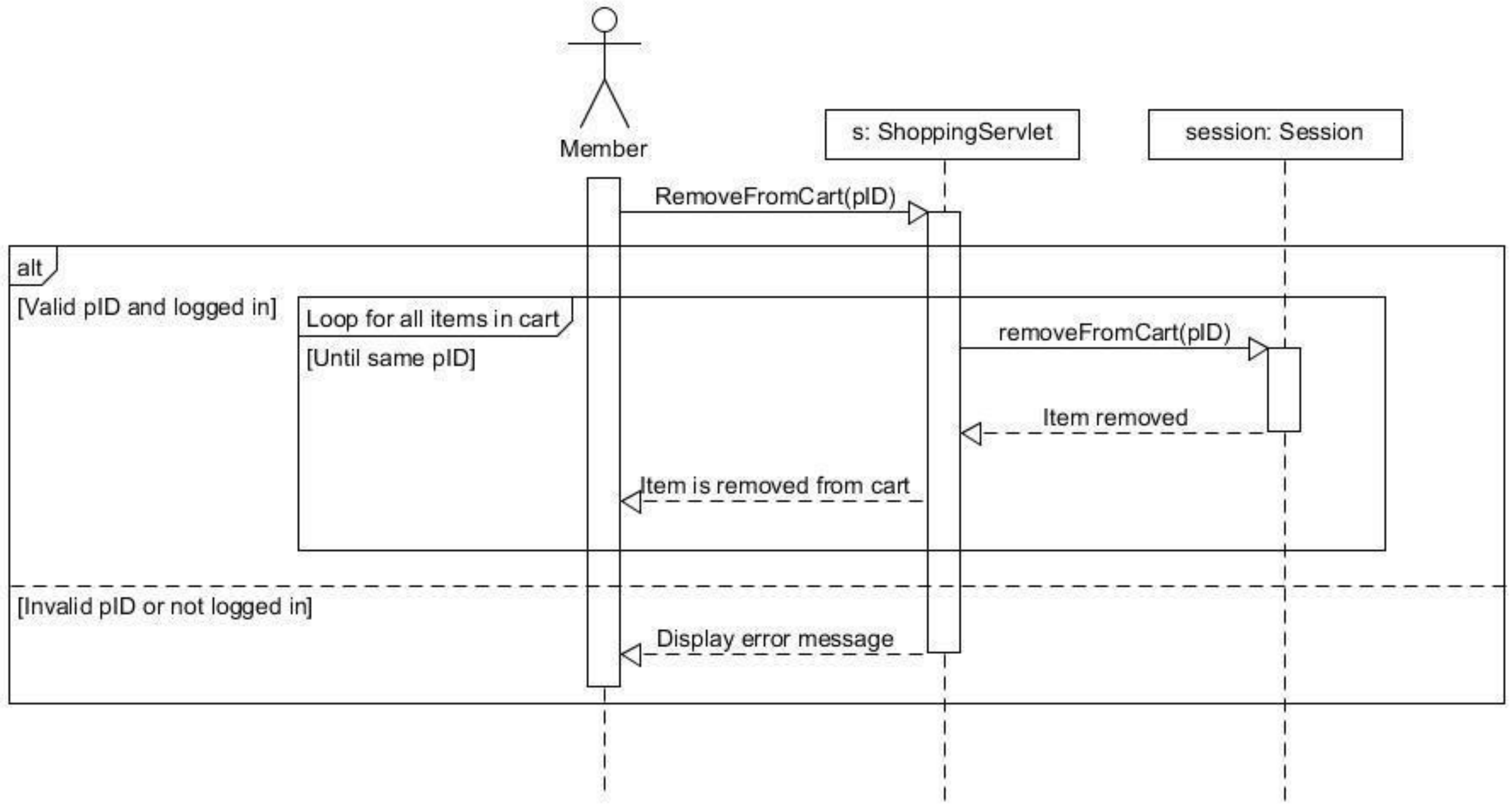
Use Case 6.2: Remove from wish list



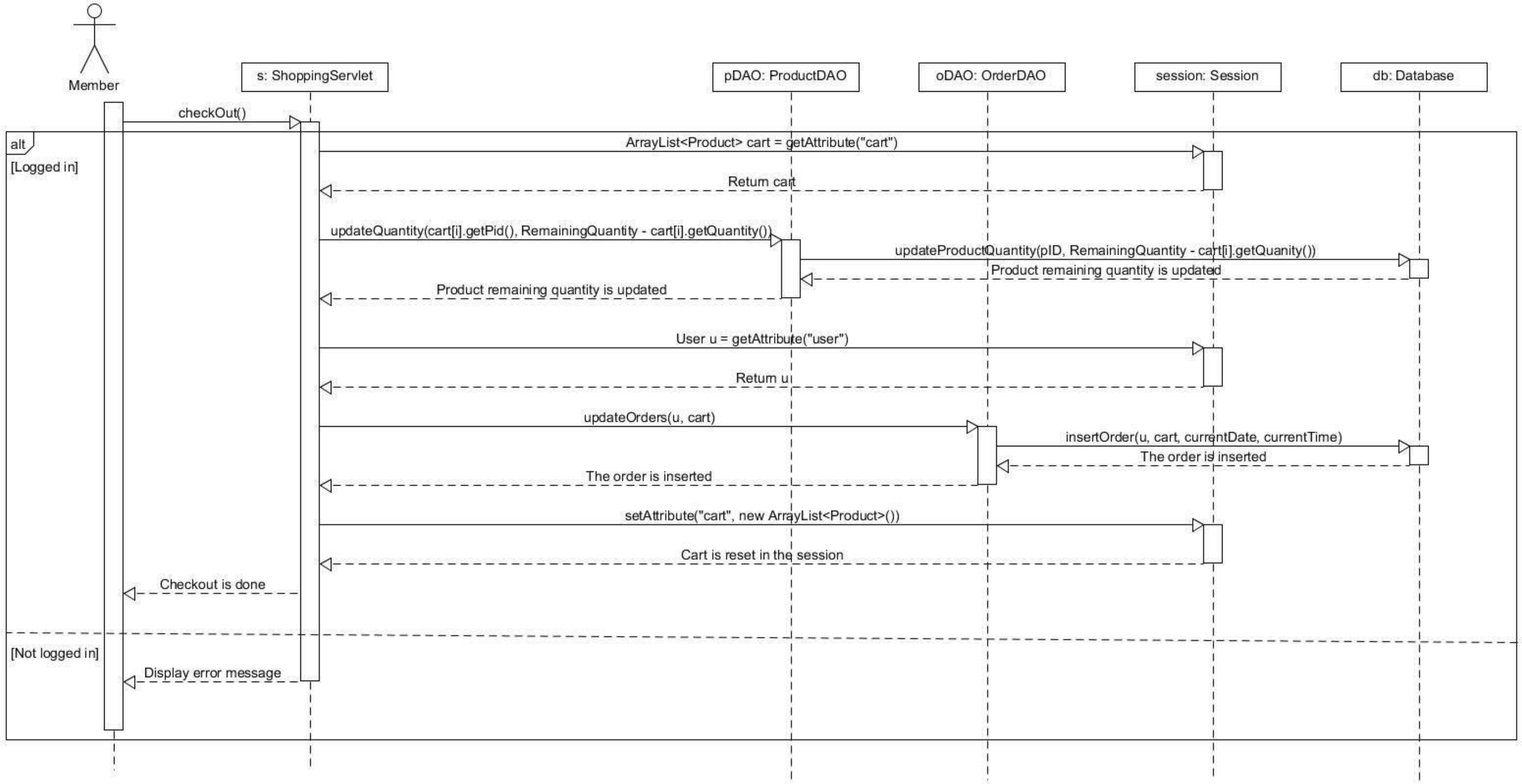
Use Case 6.3: Add to shopping cart



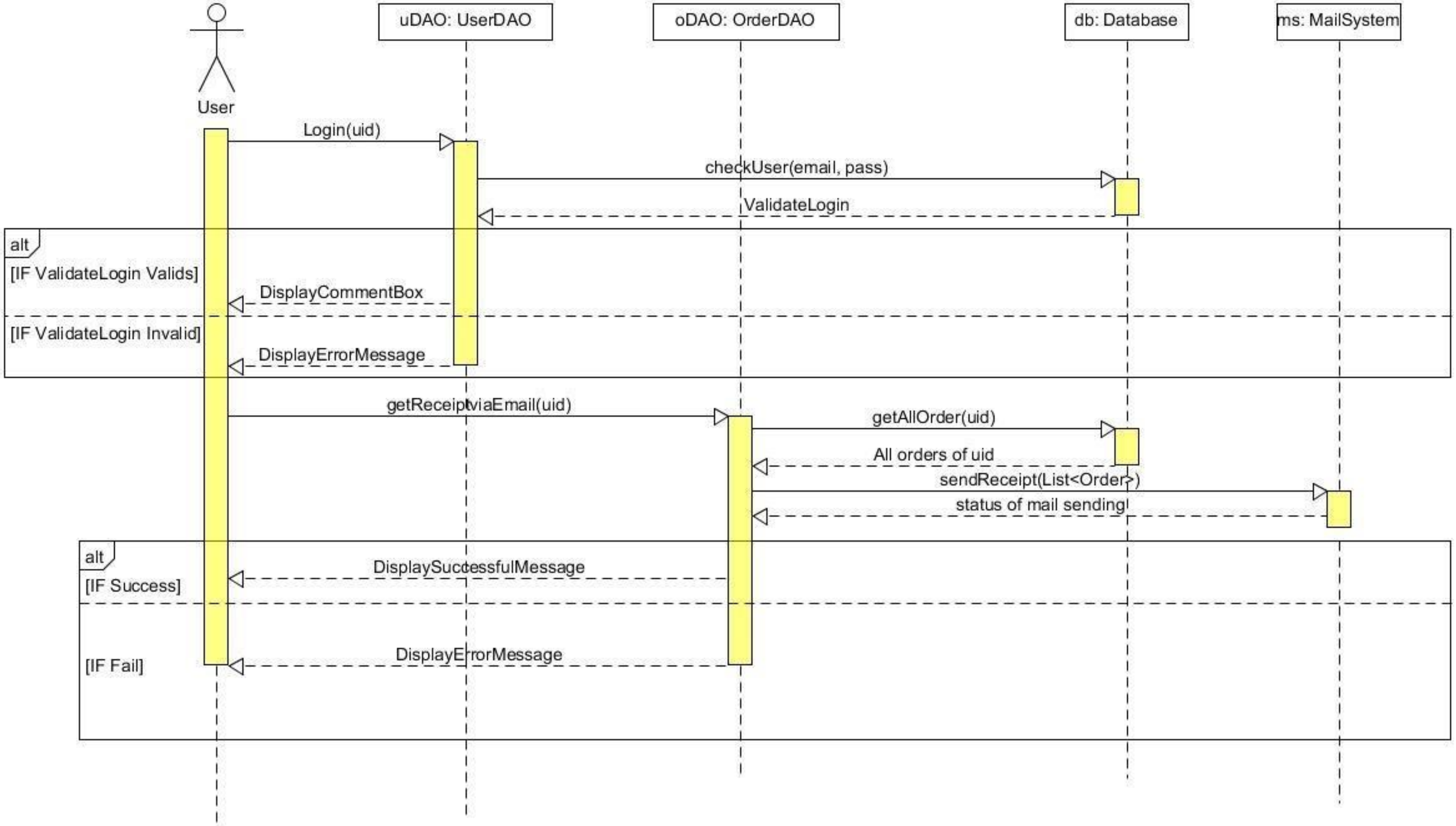
Use Case 6.4: Remove from shopping cart



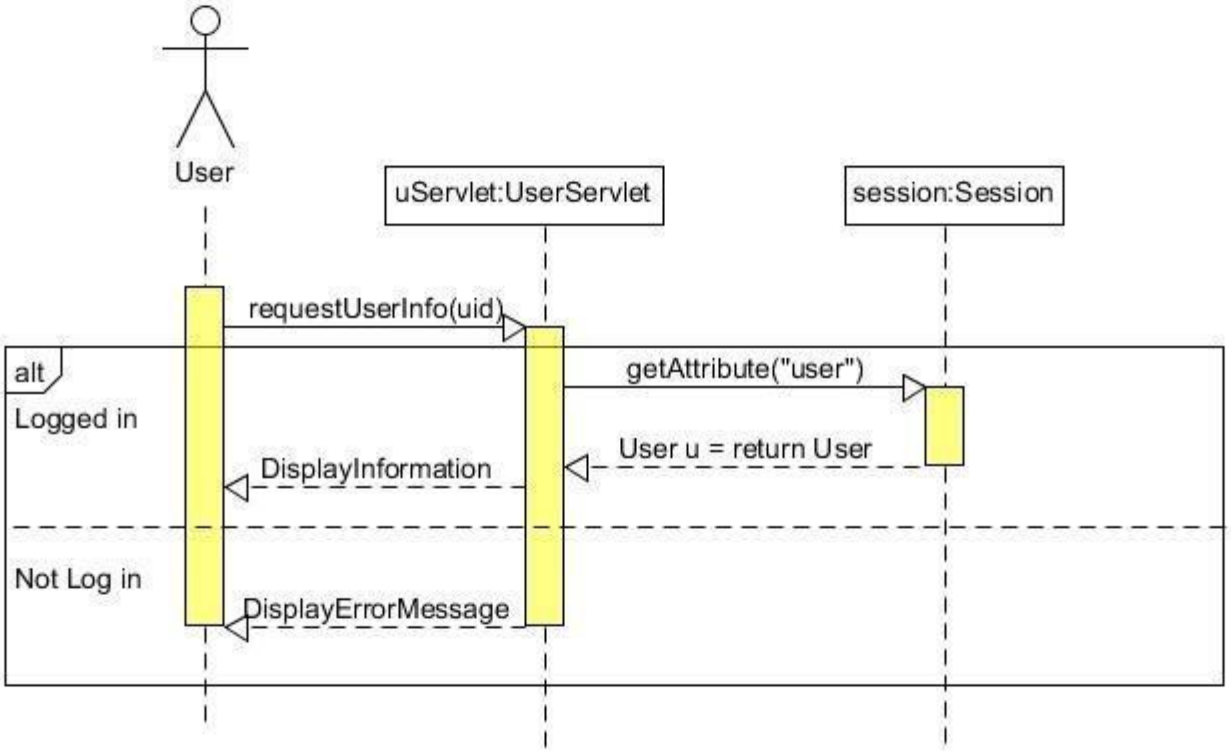
Use Case 7: Perform buying transaction and checkout


Member
s: ShoppingServlet
pDAO: ProductDAO
oDAO: OrderDAO
session: Session
db: Database
checkOut()
alt
[Logged in]
ArrayList<Product> cart = getAttribute("cart")
Return cart
updateQuantity(cart[i].getPid(), RemainingQuantity - cart[i].getQuantity())
updateProductQuantity(pID, RemainingQuantity - cart[i].getQuanity())
Product remaining quantity is updated
Product remaining quantity is updated
User u = getAttribute("user")
Return u
updateOrders(u, cart)
insertOrder(u, cart, currentDate, currentTime)
The order is inserted
The order is inserted
setAttribute("cart", new ArrayList<Product>())
Cart is reset in the session
Checkout is done
[Not logged in]
Display error message

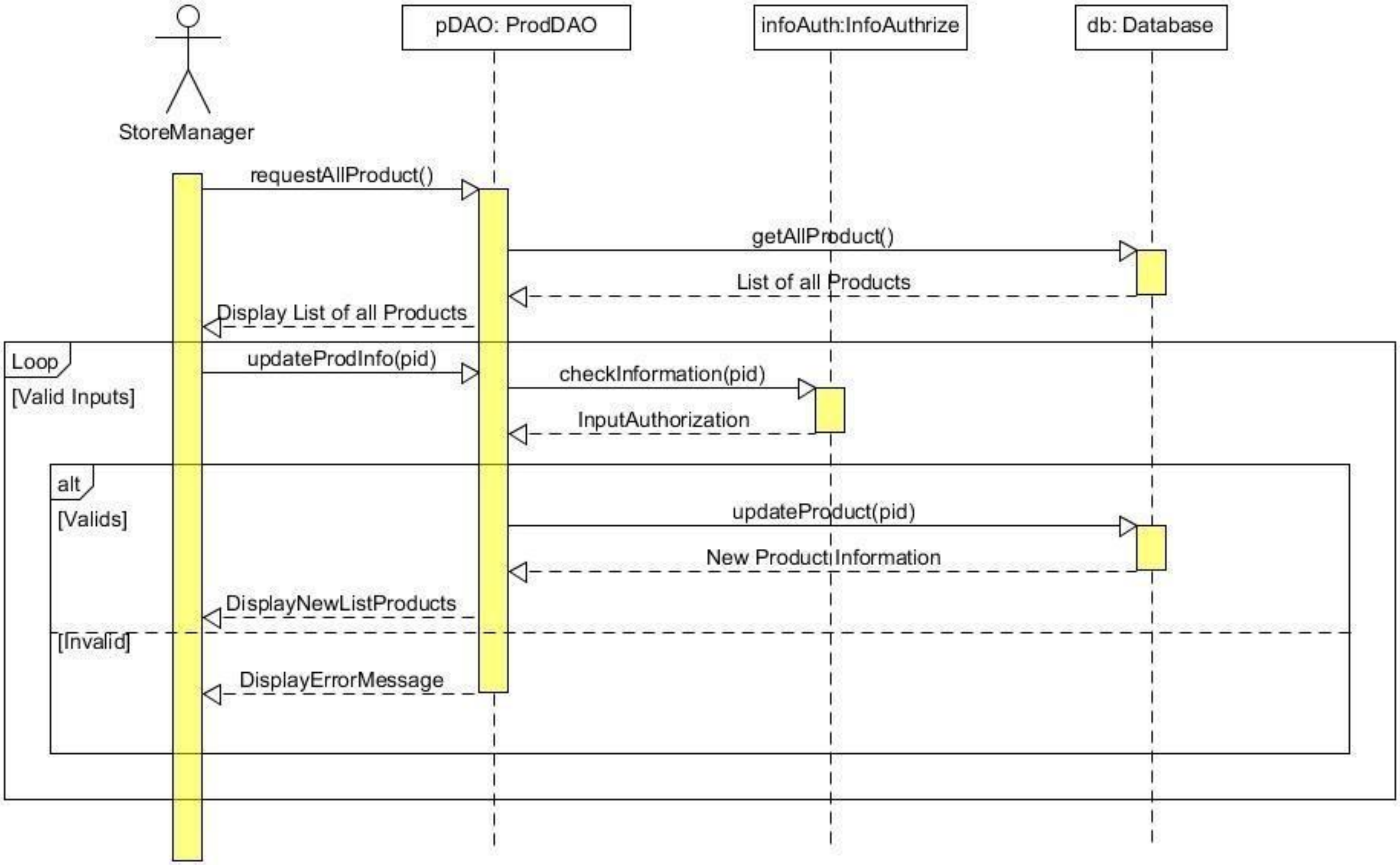
Use case 8: Get the purchase receipt via email



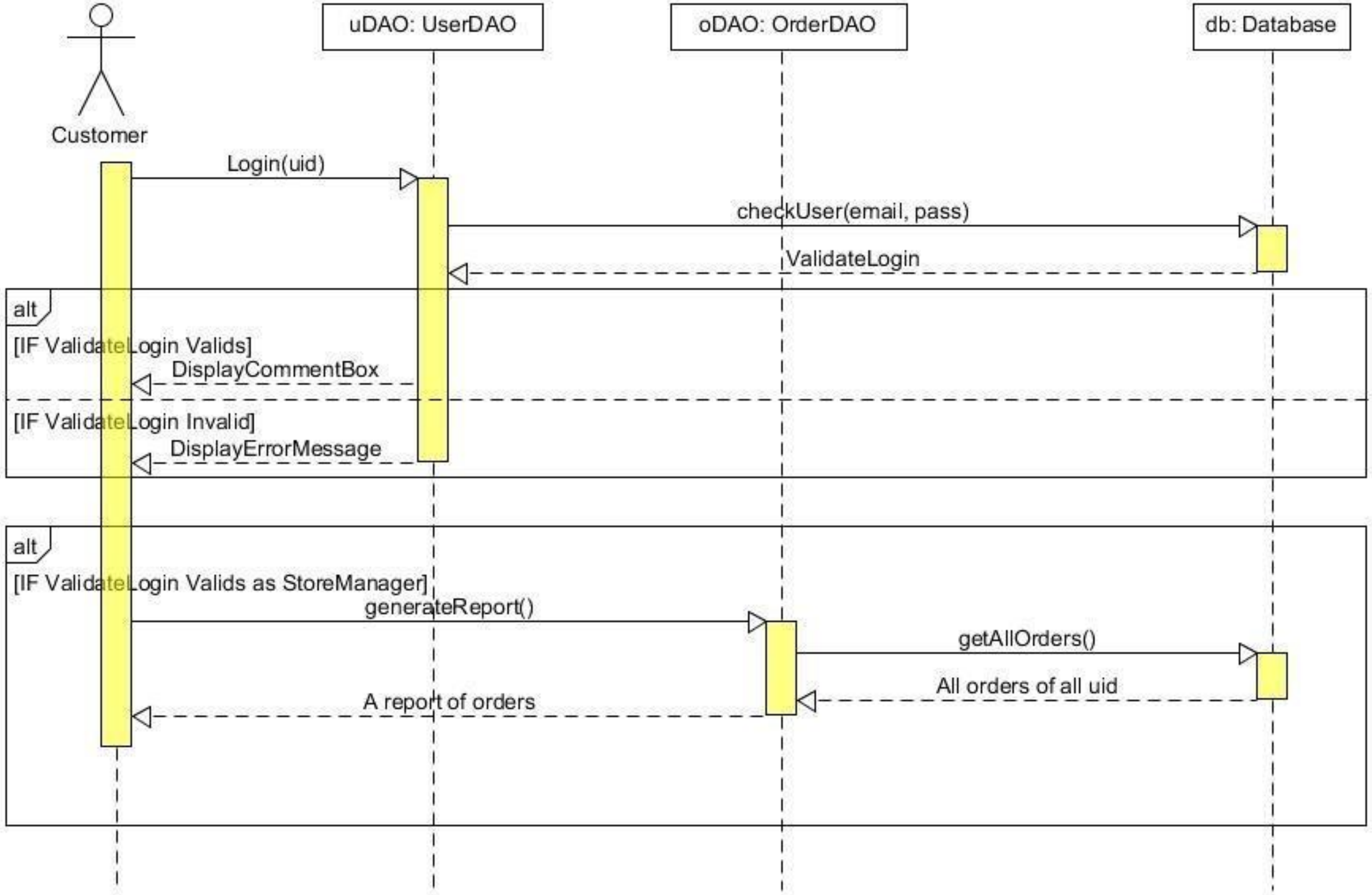
Use Case 9: Display user's profile



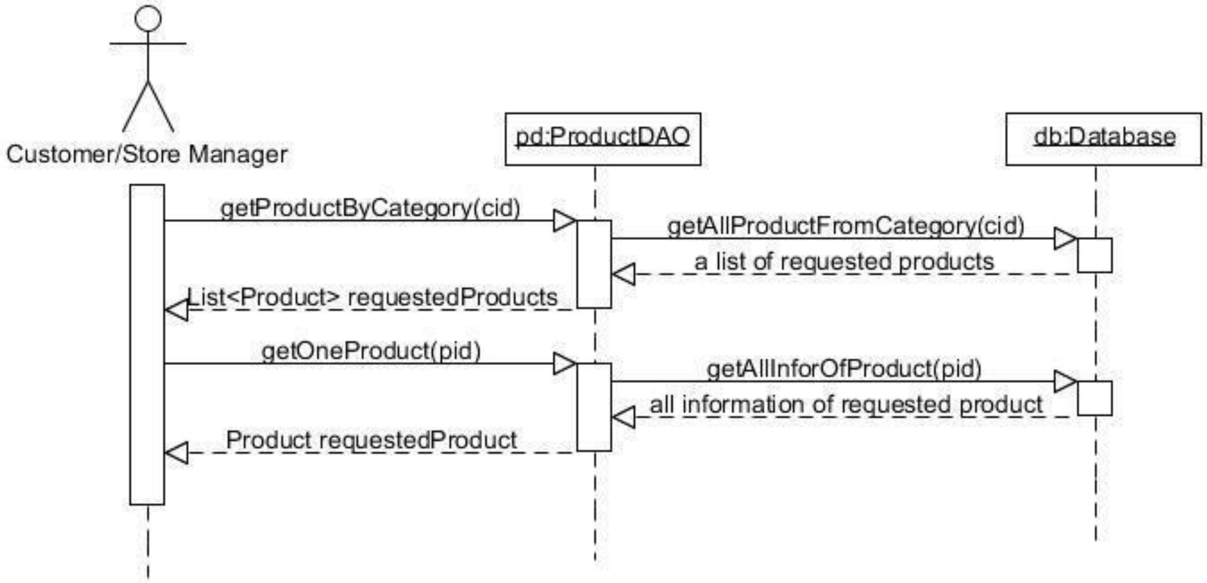
Use Case 10: Check availability of merchandise, Update the product information and clearance sale-off



Use Case 11: Generate reports of sales after each day/month/quarter/year and customers’ interests



Use Case 12: View the product's information on the web page (for visitor)


Customer/Store Manager
pd:ProductDAO
db:Database
getProductByCategory(cid)
getAllProductFromCategory(cid)
a list of requested products
List<Product> requestedProducts
getOneProduct(pid)
getAllInforOfProduct(pid)
all information of requested product
Product requestedProduct

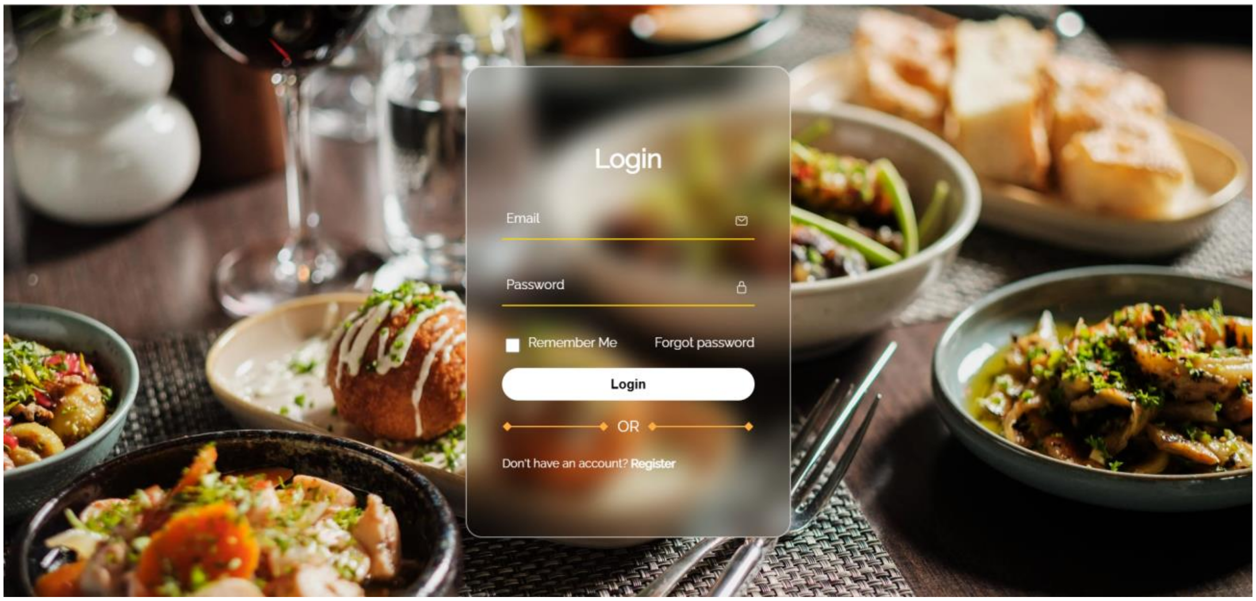
III. IMPLEMENTATION

1. USER’S ACCOUNT MANAGEMENT FUNCTIONS

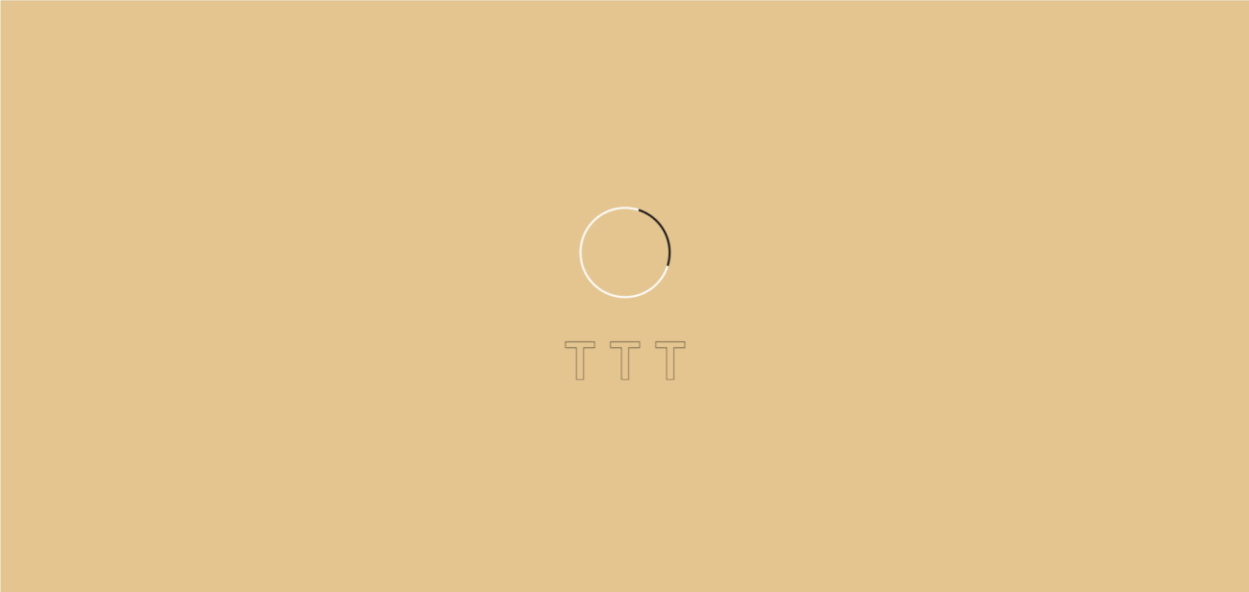
1.1. Login to the store:

Step 1: **Go to the shopping page via the link:**

Click on Login button from the homepage, the login page will appear.



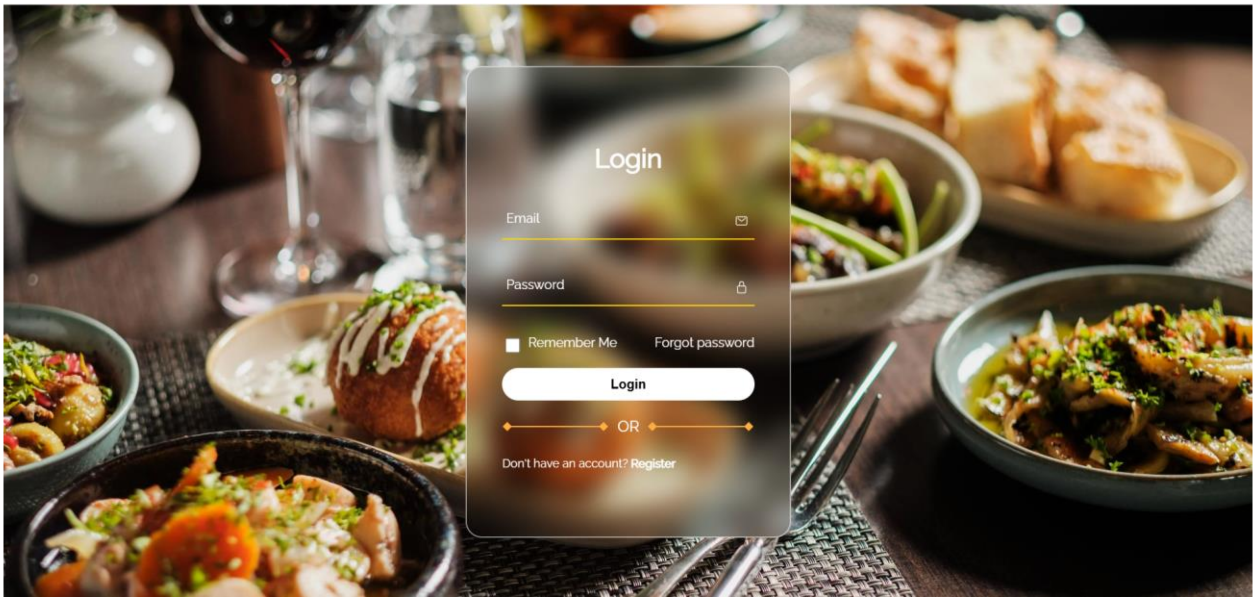
Step 2: Type user account information (email and password) in form and click on Login button.



Loading page

1.2. Register a new account:

<u>Step 1:</u> In the login page, create Register button


Login

Email

Password

□ Remember Me

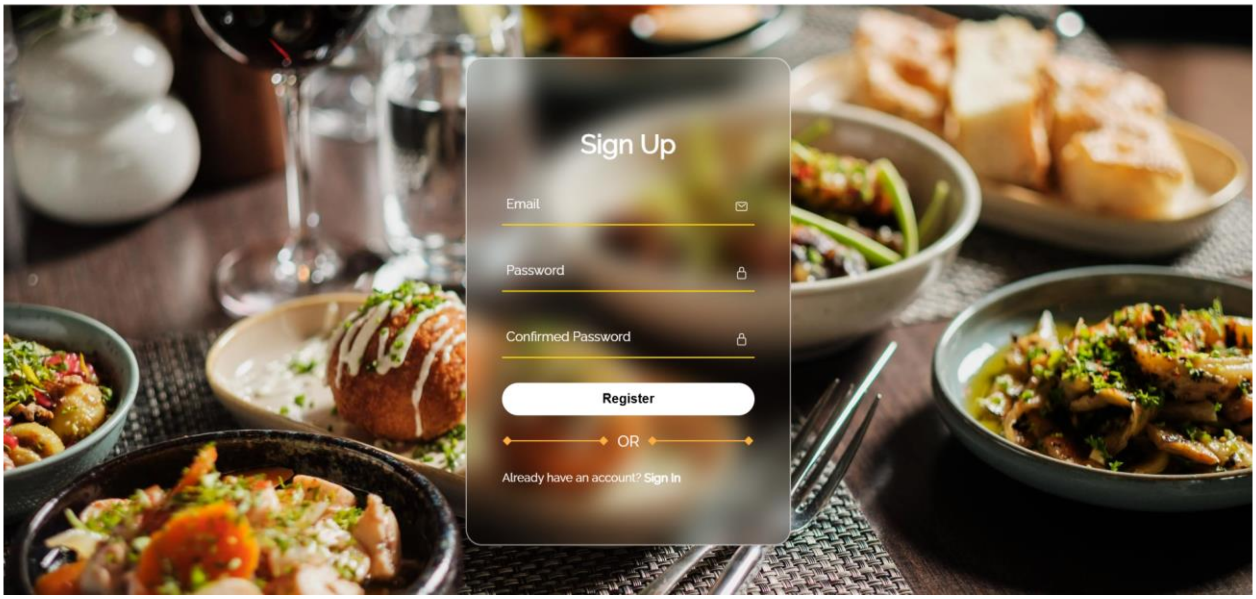
Forgot password

Login

OR

Don't have an account? Register

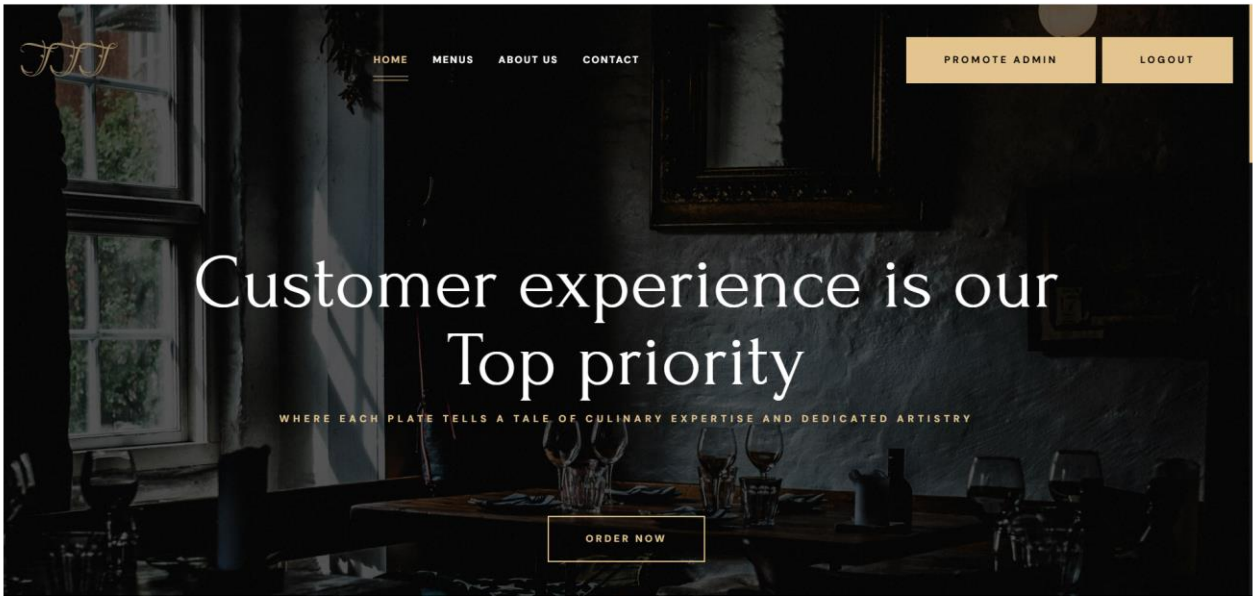
The register page will appear asking user for typing login information.



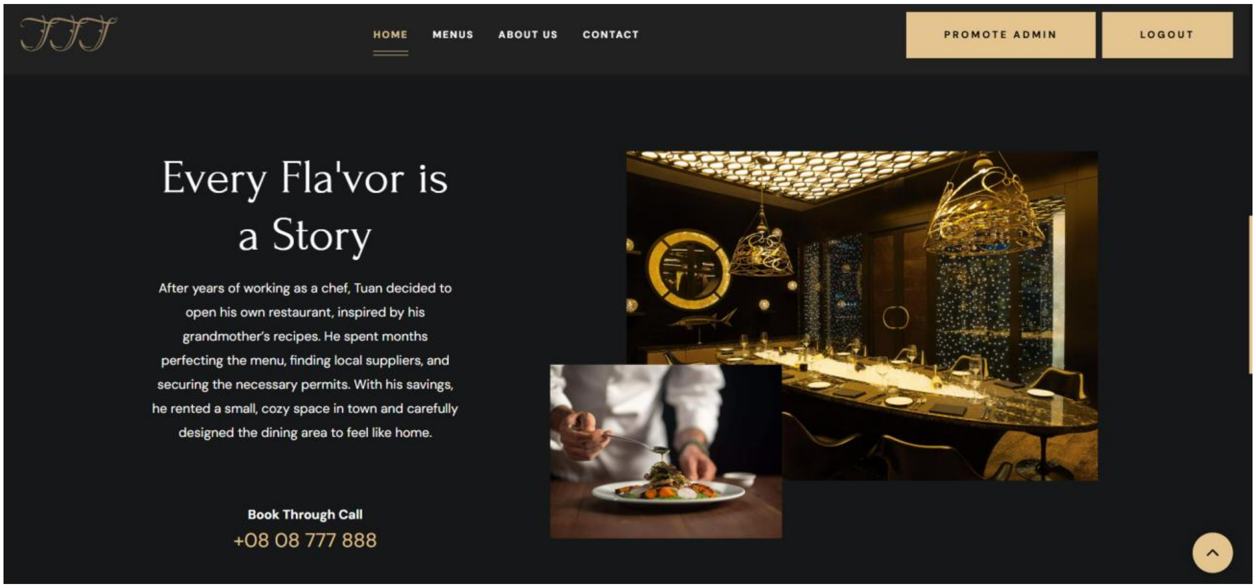
Step 2: Fill all the fields required and click Register to finish the registration

1.3. Order process

After successfully register, customer will need to log in agai). Then they will be redirected to the homepage.


HOME
MENUS
ABOUT US
CONTACT
PROMOTE ADMIN
LOGOUT
Customer experience is our Top priority
WHERE EACH PLATE TELLS A TALE OF CULINARY EXPERTISE AND DEDICATED ARTISTRY
ORDER NOW

Step 1: Click on the About Us to know more about us

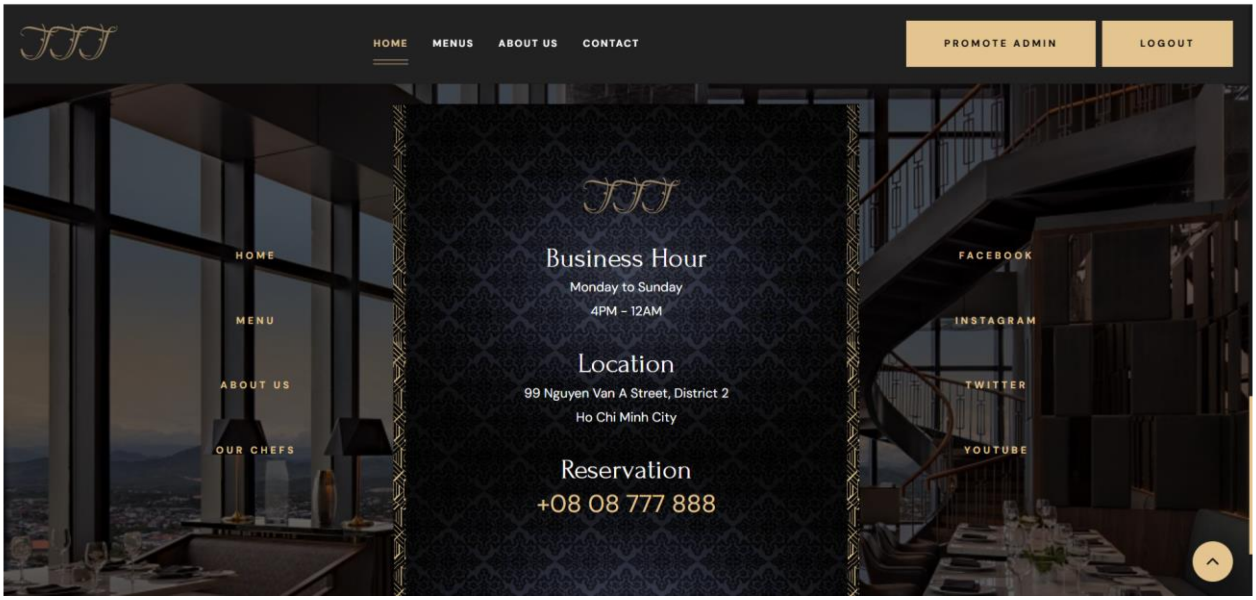


Every Fla'vor is a Story

After years of working as a chef, Tuan decided to open his own restaurant, inspired by his grandmother's recipes. He spent months perfecting the menu, finding local suppliers, and securing the necessary permits. With his savings, he rented a small, cozy space in town and carefully designed the dining area to feel like home.

Book Through Call
+08 08 777 888

If you want to contact, clicking on Contact

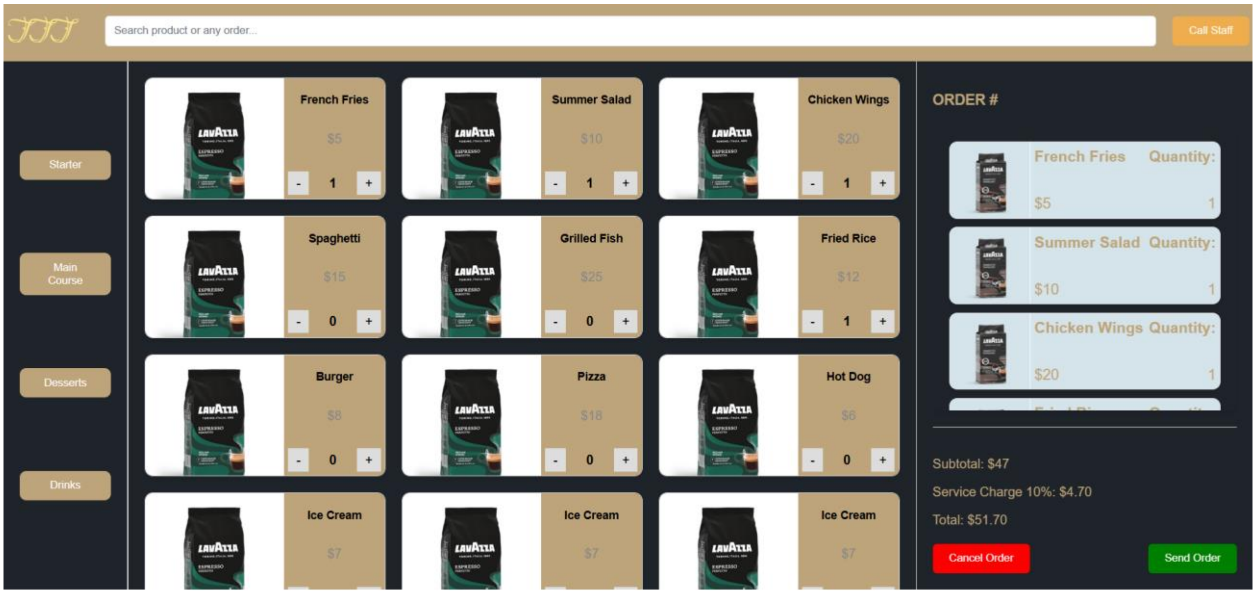


Business Hour
Monday to Sunday
4PM - 12AM

Location
99 Nguyen Van A Street, District 2
Ho Chi Minh City

Reservation
+08 08 777 888

Step 2: Click on the Cart page to start order



ORDER #

French Fries	Quantity:
$5	1
Summer Salad	Quantity:
$10	1
Chicken Wings	Quantity:
$20	1

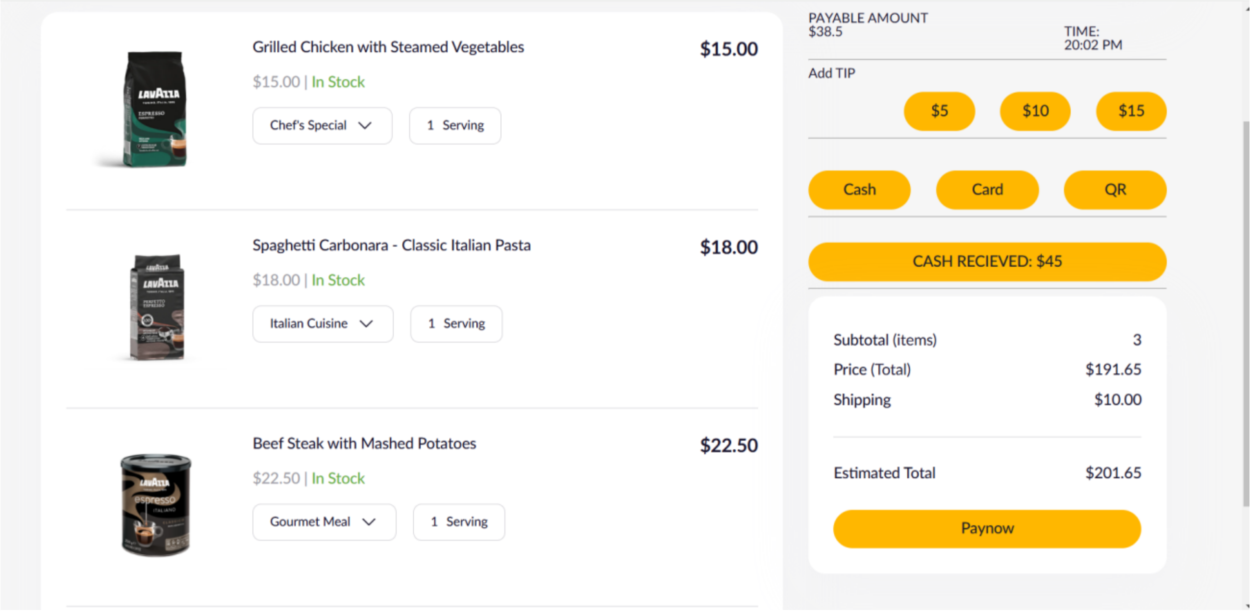
Subtotal: $47

Service Charge 10%: $4.70

Total: $51.70

Step 3: Click on button to choose, it will appear on your right

Step 4: Click on Send Order to go to check out and pay



Grilled Chicken with Steamed Vegetables $15.00 | In Stock	$15.00
Spaghetti Carbonara - Classic Italian Pasta $18.00 | In Stock	$18.00
Beef Steak with Mashed Potatoes $22.50 | In Stock	$22.50

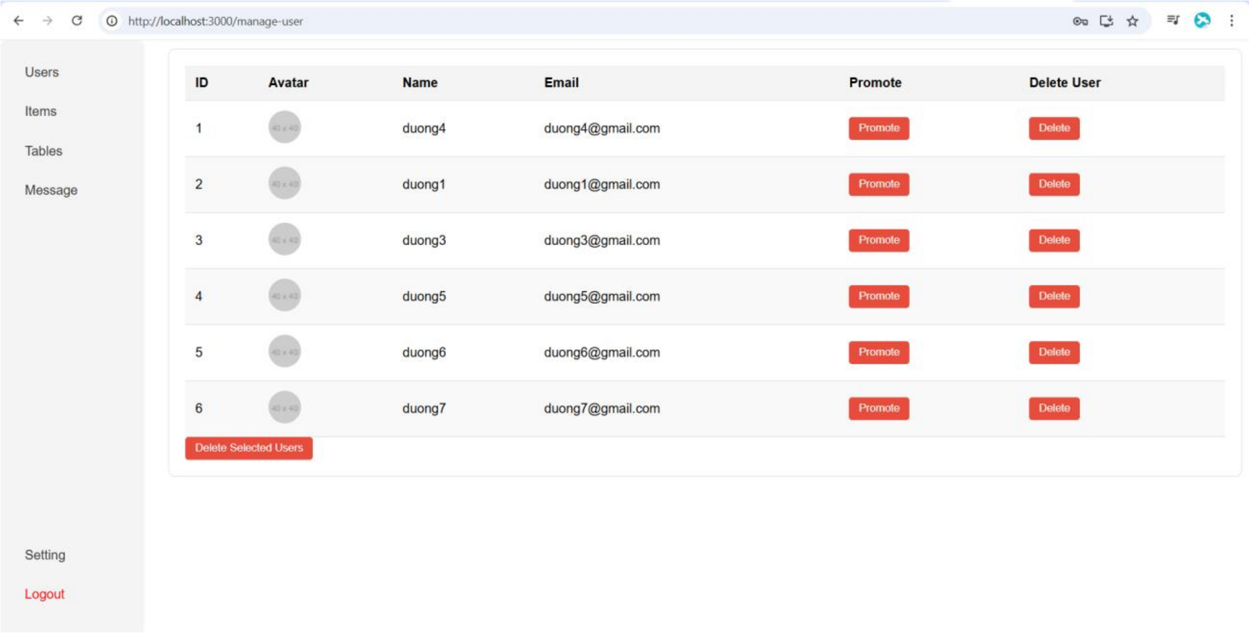
PAYABLE AMOUNT
$38.5

TIME:
20:02 PM

CASH RECIEVED: $45

Subtotal (items)	3
Price (Total)	$191.65
Shipping	$10.00
Estimated Total	$201.65

1.4 Manage User



ID	Avatar	Name	Email	Promote	Delete User
1	40 x 40	duong4	duong4@gmail.com	Promote	Delete
2	40 x 40	duong1	duong1@gmail.com	Promote	Delete
3	40 x 40	duong3	duong3@gmail.com	Promote	Delete
4	40 x 40	duong5	duong5@gmail.com	Promote	Delete
5	40 x 40	duong6	duong6@gmail.com	Promote	Delete
6	40 x 40	duong7	duong7@gmail.com	Promote	Delete

IV. DISCUSSION AND CONCLUSION

The **Order Menu** project has provided us with a deeper understanding of the **Web Application Development process**. Now that the project is complete, there are several improvements and extensions that could be made in the future. Firstly, introducing a **product recommendation feature** would enhance the system's effectiveness and user engagement. This function would suggest related products to customers based on their previous orders, encouraging additional purchases and boosting overall sales for the store. Secondly, as the current system runs as a demo on a single machine’s local host, it can only accommodate one user session at a time. To make the system scalable and deploy it in a real-world setting, the task of supporting **multiple user sessions** must be addressed. Lastly, for a fully integrated e-commerce experience, the system should incorporate **payment gateway systems**, allowing both customers and store managers to complete transactions smoothly within the platform.

This Web Application Development project has also provided valuable learning experiences. Our **coding skills** and ability to organize and structure code within a project have improved significantly. This will be crucial for maintaining and expanding the system in the future. The project also helped us strengthen our ability to **analyze requirements** and translate them into practical designs before beginning development. Furthermore, working on a project of this scale has enhanced our **teamwork** and **communication skills**, making it easier to collaborate and share ideas effectively.

In summary, the Order Menu project has not only allowed us to apply the knowledge gained in the course but also provided us with **hands-on experience** and **valuable skills** that will be beneficial for future projects.

V. REFERENCES

- **SCRUM**
- https://www.scrum.org/
- **Agile**
- http://agilemethodology.org/
- **Web-page template**
- http://themeforest.net/
- **Database Management System (DBMS)**
- http://www.oracle.com/
- **HTML, CSS, techniques**
- www.w3schools.com