Introduction to Software Engineering

Software Design

*The student team is required to complete the **Design Document** for the assigned course project, following the attached template.*

Software Engineering Department
Faculty of Information and Technology
University of Science

Table of Contents

1 Member Contribution Assessment..... 2
2 Conceptual Model..... 3
3 Architectural Design..... 4
3.1 Architecture Diagram..... 4
3.1.1 System decomposition tree diagram..... 4
3.1.2 Overall system architecture diagram..... 5
3.1.3 Architectural Patterns & Design Principles..... 5
3.2 Class Diagram..... 7
3.3 Class Specifications..... 8
3.3.1 Class: Account (Abstract)..... 8
3.3.2 Class: Customer..... 9
3.3.3 Class: Admin..... 11
3.3.4 Class: Product..... 12
3.3.5 Class: Order..... 14
3.3.6 Class: OrderItem..... 15
3.3.7 Class: Cart..... 16
3.3.8 Class: Promotion..... 17
3.3.9 Class: Payment (Abstract)..... 18
3.3.10 Class: Banking Payment..... 19
4 Data Design..... 21
4.1 Data Diagram..... 21
4.2 Data Specification..... 22
4.2.1 Collection: accounts..... 22
4.2.2 Collection: products..... 24
4.2.3 Collection: orders..... 25
4.2.4 Collection payments..... 28
4.2.5 Collection: reviews..... 29
5 User Interface and User Experience Design..... 31
5.1 Screen Diagram..... 31
5.2 Screen Specifications..... 33
5.2.1 Screen “Home Page”..... 34
5.2.2 Screen “Product Detail Page”..... 42
5.2.3 Screen “Shopping Cart Page”..... 49
5.2.4 Screen “Checkout Page”..... 54

5.2.5 Screen “Admin Dashboard”.....59
5.2.6 Screen “Order Management (Admin)”.....64
5.2.7 Screen “Product Management (Admin)”.....69

Software Design

Objectives

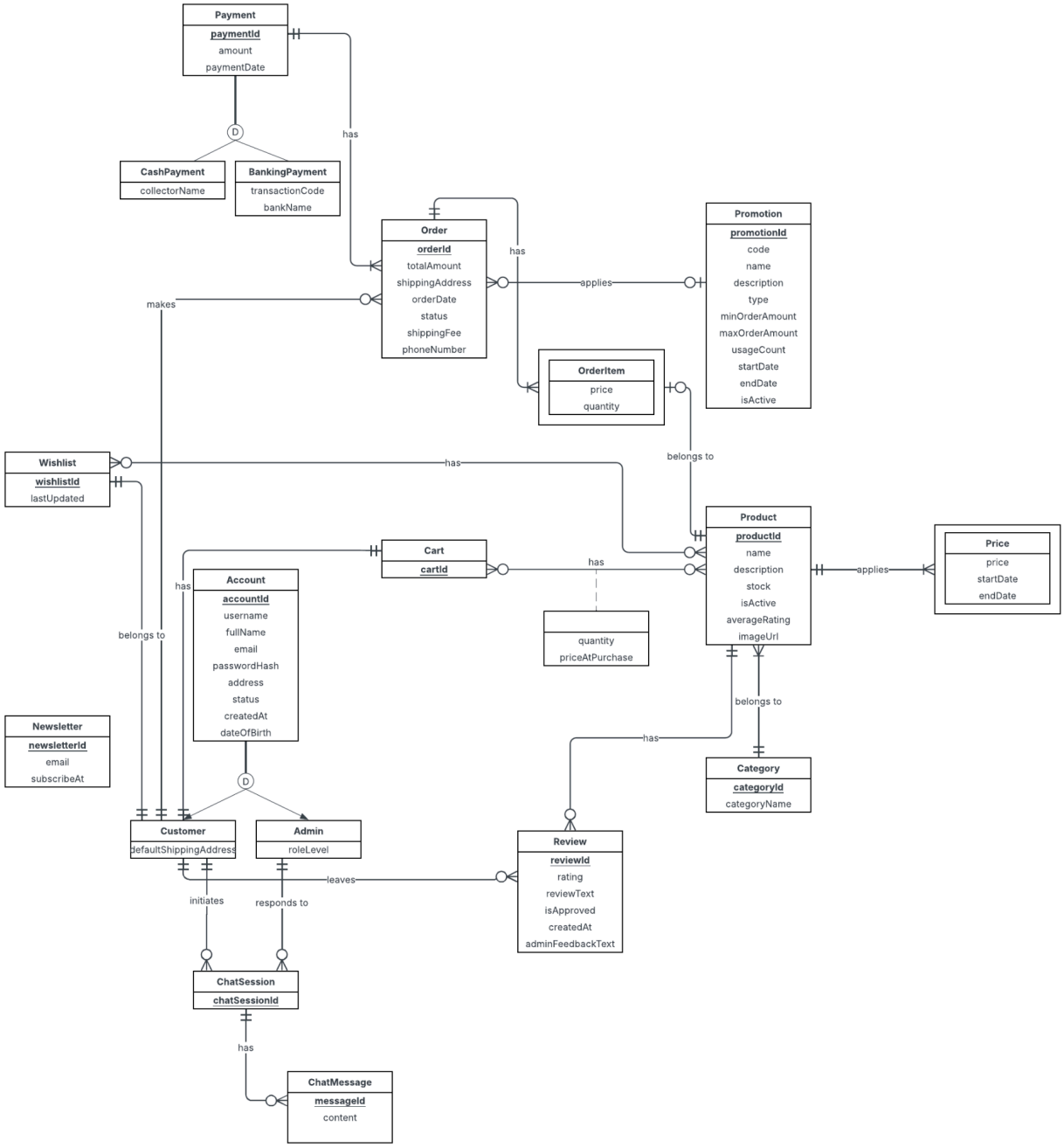
This document focus on the following topics:

- ✔ Complete the Software Design Document with the following contents:
  - Conceptual Model
  - Architectural Design
  - Data Design
  - User Interface Design
- ✔ Understanding the Software Design Document.

1 Member Contribution Assessment

ID	Name	Contribution (%)	Signature
23127128	Nguyễn Thành Tiến	100%	
23127157	Nguyễn Hoàng Gia Bảo	100%	
23127184	Lê Phạm Kiều Duyên	100%	
23127396	Lương Linh Khôi	100%	

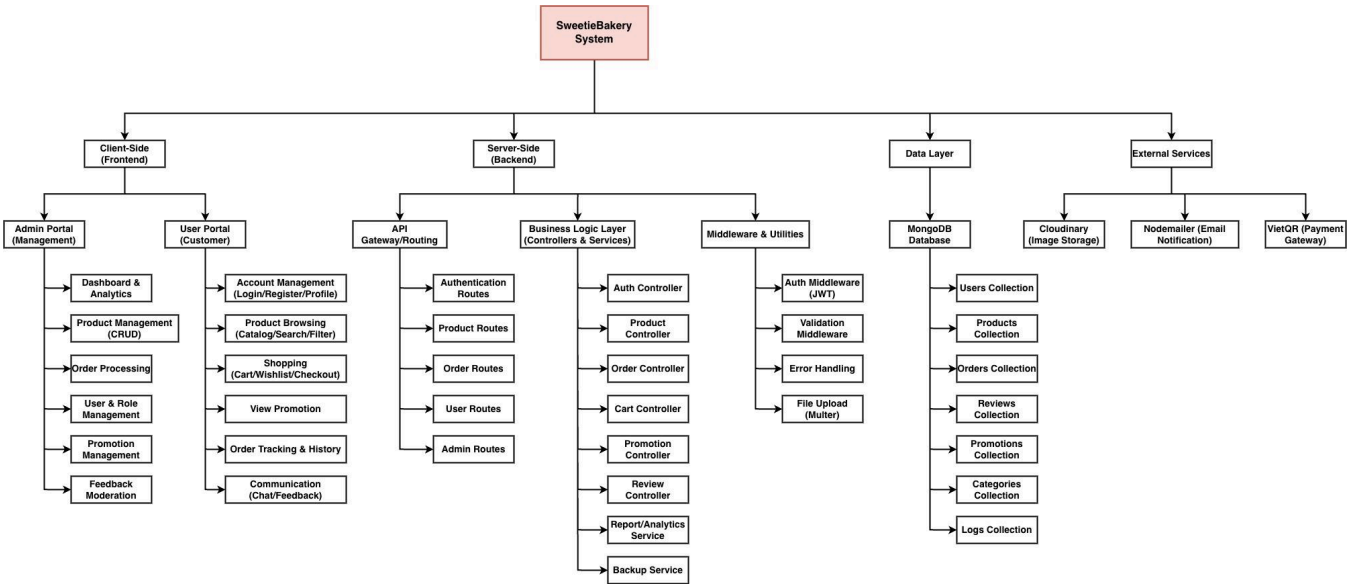
2 Conceptual Model



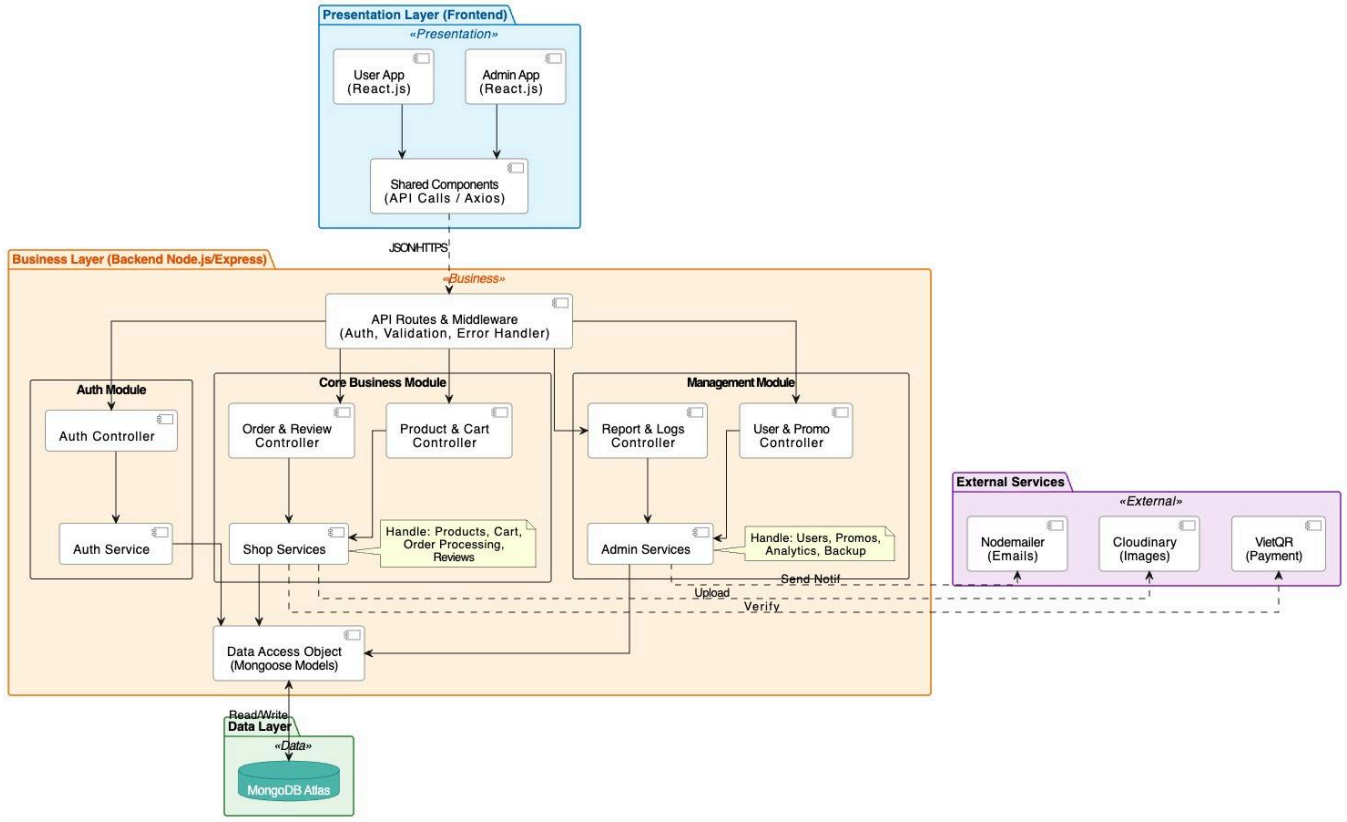
3 Architectural Design

3.1 Architecture Diagram

3.1.1 System decomposition tree diagram



3.1.2 Overall system architecture diagram



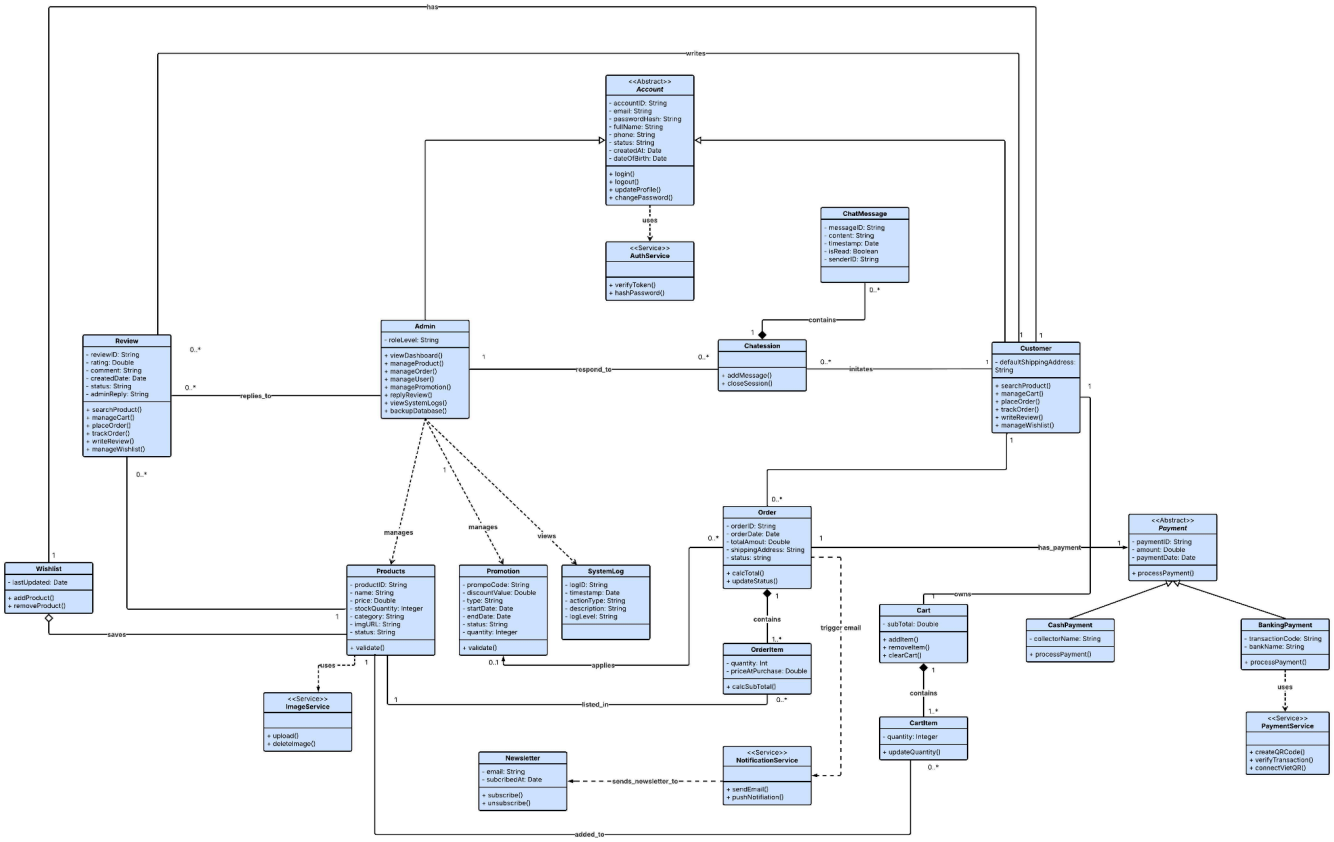
3.1.3 Architectural Patterns & Design Principles

The SweetieBakery system is built upon several key architectural patterns and design principles to ensure scalability, maintainability, and security:

- **Client-Server Architecture:** The system maintains a clear separation of concerns between the **Frontend** (React.js executing in the client browser) and the **Backend** (Node.js/Express running on the server). Communication occurs via **RESTful APIs** using JSON format. This decoupling ensures system flexibility, facilitates scalability, and allows for independent maintenance of the client and server sides.

- **MVC Pattern (Model-View-Controller) on Backend:** The backend logic is structured using the MVC pattern to organize code efficiently:
    - **Model (Mongoose Models):** Defines the data structure (Schema) and handles direct interactions with the MongoDB database.
    - **View (React Components):** Although React operates on the client-side, within the overall system context, it functions as the presentation layer responsible for rendering and displaying data received from the API.
    - **Controller:** Acts as the entry point for requests from Routes, handles coordination logic, and delegates specific tasks to the Service layer.
- **Service-Oriented Logic:** Complex business logic (such as shopping cart calculations, payment processing, and email notifications) is decoupled from the Controller and encapsulated within the **Service layer**. This adheres to the **"Separation of Concerns"** principle, resulting in cleaner code architecture that is significantly easier to test and maintain.
- **Layered Architecture:** The system is structured into three distinct layers: **Presentation Layer**, **Business Layer**, and **Data Layer**. This strict layering strategy enhances system security and ensures centralized, consistent data management throughout the application.

3.2 Class Diagram



3.3 Class Specifications

3.3.1 Class: Account (Abstract)

Inherits from: *None (Base Class)*

Attributes List:

Seq	Property	Modifier	Constraint	Description
1	accountID	Private	PK, Unique	Unique identifier for the account.
2	email	Private	Unique, Not Null, Valid Email	Email address used for login and notifications.
3	passwordHash	Private	Not Null, Min 8 chars	Encrypted password hash for security.
4	fullName	Private	Not Null	Full name of the user.
5	phone	Private	10-11 digits	Contact phone number.
6	isActive	Private	Boolean, Default: True	Operational status of the account (True: Active).

7	createdAt	Private	Date	Timestamp when the account was created.

Methods/Operations List:

Seq	Operation	Modifier	Constraint	Description
1	login()	Public		Authenticates the user credentials.
2	logout()	Public		Terminates the current user session.
3	updateProfile()	Public		Updates personal information (excluding Email/ID).
4	changePassword()	Public	Match old password	Updates the account password.

3.3.2 Class: Customer

Represents a customer who uses the system to browse products and place orders.

Inherits from: Account

Attributes List:

Seq	Property	Modifier	Constraint	Description
1	defaultShippingAddress	Private	String	The default delivery address for the customer.
2	loyaltyPoints	Private	Int, >= 0	Accumulated loyalty points (if applicable).

(Inherits attributes: email, passwordHash, fullName... from Account)

Methods/Operations List:

Seq	Operation	Modifier	Constraint	Description
1	searchProduct()	Public		Searches for products by keywords or category.
2	manageCart()	Public		View, add, or remove items in the shopping cart.
3	placeOrder()	Public	Cart not empty	Initiates the order creation process from the cart.
4	trackOrder()	Public		Retrieves the status of placed orders.

5	manageWishlist()	Public		Manages the list of favorite products.

3.3.3 Class: Admin

Represents a system administrator with privileges to manage business data and operations.

Inherits from: Account

Attributes List:

Seq	Property	Modifier	Constraint	Description
1	roleLevel	Private	String (e.g., "SuperAdmin")	Level of administrative privileges.

(Inherits basic attributes from Account)

Methods/Operations List:

Seq	Operation	Modifier	Constraint	Description
1	viewDashboard()	Public		Visualizes sales statistics and order summaries.
2	manageProduct()	Public		Performs CRUD operations on product data.

3	manageOrder()	Public		Updates the processing status of customer orders.
4	manageUser()	Public		Locks or unlocks user accounts.
5	managePromotion()	Public		Creates and manages promotional campaigns.
6	viewSystemLogs()	Public		Reviews system activity logs.

3.3.4 Class: Product

Represents a bakery item available for sale in the system.

Inherits from: *None*

Attributes List:

Seq	Property	Modifier	Constraint	Description
1	productID	Private	PK, Unique	Unique identifier for the product.
2	name	Private	Not Null	Name of the product.

3	price	Private	Double, > 0	Current selling price.
4	stockQuantity	Private	Int, >= 0	Current quantity in stock.
5	category	Private	Not Null	Product classification (e.g., Cake, Bread).
6	imageURL	Private	Valid URL	Link to the product image (hosted on Cloud).
7	status	Private	Enum (Active, Inactive)	Business status of the product.

Methods/Operations List:

Seq	Operation	Modifier	Constraint	Description
1	updateStock()	Public	Amount != 0	Updates stock quantity (restock or sold).
2	validate()	Public		Validates product data integrity.

3.3.5 *Class:* Order

Represents a purchase order created by a customer.

Inherits from: *None*

Attributes List:

Seq	Property	Modifier	Constraint	Description
1	orderID	Private	PK, Unique	Unique identifier for the order.
2	orderDate	Private	DateTime, Default Now	Timestamp when the order was placed.
3	totalAmount	Private	Double, >= 0	Final total value after discounts.
4	shippingAddress	Private	Not Null	Specific delivery address for this order.
5	status	Private	Enum (New, Processing...)	Current processing status of the order.

Methods/Operations List:

Seq	Operation	Modifier	Constraint	Description
1	calcTotal()	Public		Calculates total amount from

				OrderItems and Promotions.
2	updateStatus()	Public	Valid Transition	Updates the order status (Admin/System).

3.3.6 Class: OrderItem

Represents a specific line item within an order, capturing the price at the moment of purchase.

Inherits from: *None*

Attributes List:

Seq	Property	Modifier	Constraint	Description
1	quantity	Private	Int, > 0	Quantity of the product ordered.
2	priceAtPurchase	Private	Double, > 0	Snapshot of the product price at the time of order.

Methods/Operations List:

Seq	Operation	Modifier	Constraint	Description
1	calcSubTotal()	Public		Calculates line total (quantity * priceAtPurchase).

3.3.7 Class: Cart

Represents the shopping cart for a user's current session.

Inherits from: *None*

Attributes List:

Seq	Property	Modifier	Constraint	Description
1	quantity	Private	Int, > 0	Quantity of the product ordered.
2	priceAtPurchase	Private	Double, > 0	Snapshot of the product price at the time of order.

Methods/Operations List:

Seq	Operation	Modifier	Constraint	Description
1	addItem()	Public	Stock > 0	Adds a product to the cart.

2	removeItem()	Public		Removes a product from the cart.
3	clearCart()	Public		Empties the cart after successful checkout.

3.3.8 Class: Promotion

Manages discount codes and promotional campaigns.

Inherits from: *None*

Attributes List:

Seq	Property	Modifier	Constraint	Description
1	promoCode	Private	PK, Unique	The discount code string (e.g., WELCOME).
2	discountValue	Private	Double, > 0	The value of the discount.
3	type	Private	Enum (Percent, Fixed)	Type of discount (Percentage or Fixed amount).

4	startDate	Private	Date	Date when the promotion becomes effective.
5	endDate	Private	Date > startDate	Date when the promotion expires.
6	status	Private	Enum (Active, Expired)	Current status of the promotion.

Methods/Operations List:

Seq	Operation	Modifier	Constraint	Description
1	validate()	Public		Checks if the code is valid and active.

3.3.9 Class: Payment (Abstract)

An abstract class handling the payment process structure.

Inherits from: *None (Base Class)*

Attributes List:

Seq	Property	Modifier	Constraint	Description
1	paymentID	Private	PK, Unique	Internal payment transaction identifier.

2	amount	Private	Double, > 0	The amount to be paid.
3	paymentDate	Private	DateTime	Timestamp when the payment was initiated.

Methods/Operations List:

Seq	Operation	Modifier	Constraint	Description
1	processPayment()	Public	Abstract	Abstract method to execute the payment logic.

3.3.10 Class: Banking Payment

Handles payments made via Bank Transfer (e.g., VietQR).

Inherits from: Payment

Attributes List:

Seq	Property	Modifier	Constraint	Description
1	transactionCode	Private	Unique	Reference transaction code from the Bank/Gateway.

2	bankName	Private	String	Name of the beneficiary bank.

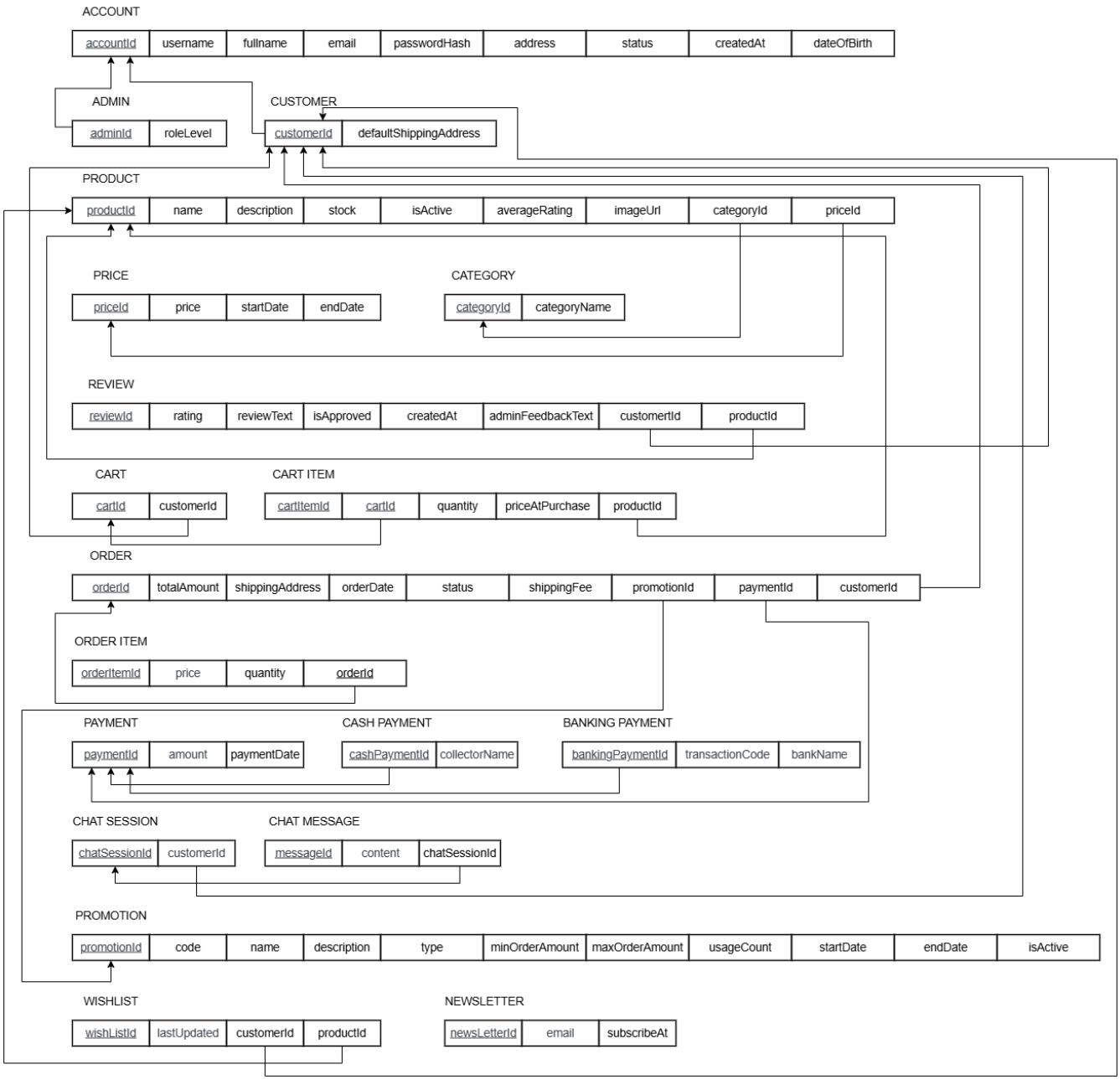
(Inherits amount, paymentID... from Payment)

Methods/Operations List:

Seq	Operation	Modifier	Constraint	Description
1	processPayment()	Public	Abstract	Executes the specific payment logic required for the chosen payment type.

4 Data Design

4.1 Data Diagram



4.2 Data Specification

4.2.1 Collection: accounts

This collection stores all user accounts, embedding related customer-specific data (Newsletter, defaultShippingAddress) and Wishlist references.

Attribute Name	Data Type	Value Constraints	Explanation
_id	ObjectId	Required, Unique	MongoDB's primary identifier
accountId	String	Required, Unique	Business-level account ID
username	String	Required, Unique	User login name
email	String	Required, Unique, Valid Email format	User's contact email
passwordHash	String	Required, Min Length 60 (for strong hash)	Hashed password for security
status	String	Required ('active', 'inactive')	Account status
role	String	Required ('User' or 'Admin')	User role inheritance (discriminator)
fullName	String	Optional	Full name of the user.
dateOfBirth	Date	Optional	User's date of birth
defaultShippingAddress	Object	Optional (Customer-specific)	Embedded object containing street, city...

newsletter	Object	Optional (Customer-specific)	**Embedded** structure: { subscribedAt: Date }
wishlists	Array of Objects	Optional (Customer-specific)	Array of references/ embedded IDs: { wishlistId: String, lastUpdated: Date }

Example Document Structure (JSON):

```
{
    "_id": ObjectId("65c5c9f3e4b0c2a8f9d0c2c3"),
    "accountId": "ACC-0001",
    "username": "user",
    "email": "user@gmail.com",
    "passwordHash": "$2a$10$...",
    "status": "active",
    "role": "User",
    "fullName": "User 1",
    "defaultShippingAddress": {
        "street": "123 A",
        "city": "HCM",
        "zipCode": "999"
    },
    "newsletter": {
        "subscribedAt": ISODate("2025-12-22T10:00:00Z")
    },
    "wishlists": [
        { "wishlistId": "WL-456", "lastUpdated": ISODate("2025-11-23T14:30:00Z") }
    ]
}
```

4.2.2 Collection: products

This collection holds core product information, embedding categories and prices for efficient lookups.

Attribute Name	Data Type	Value Constraints	Explanation
_id	ObjectId	Required, Unique	MongoDB's primary identifier.
productId	String	Required, Unique	Business-level product ID.
name	String	Required	Product name.
description	String	Optional	Detailed product description.
stock	Number	Required, Integer, (<= 0)	Current stock quantity.
isActive	Boolean	Required	Availability status.
availableSizes	Array of String	Optional	List of available sizes/options.
imageUrls	Array of String	Required	URLs for product images.
category	Object	**Embedded** (1:1 / 1:n relationship)	Embedded category info: { categoryId: String, categoryName: String }.
prices	Array of Objects	**Embedded** (1:n relationship)	Embedded price history: Array of { price: Decimal,

			startDate: Date, endDate: Date }.

Example Document Structure (JSON):

```
{
    "_id": ObjectId("65c5c9f3e4b0c2a8f9d0c2d4"),
    "productId": "PROD-A01",
    "name": "Cake",
    "description": "Birthday Cake",
    "stock": 150,
    "isActive": true,
    "imageUrls": ["/ex.png"],
    "category": {
        "categoryId": "CAT-003",
        "categoryName": "Examples"
    },
    "prices": [
        {
        "price": 29.99,
        "startDate": ISODate("2025-01-01T00:00:00Z"),
        "endDate": null
        }
    ]
}
```

4.2.3 Collection: orders

This collection handles all order information, embedding line items and referencing customer, payment, and promotion documents.

Attribute Name	Data Type	Value Constraints	Explanation
_id	ObjectId	Required, Unique	MongoDB's primary identifier.
orderId	String	Required, Unique	Business-level order ID.
customerRef	ObjectId	Reference to accounts, Required	Link to the customer who placed the order.
orderDate	Date	Required	Date the order was placed.
totalAmount	Decimal	Required, (>=0)	Final amount paid (including shipping/fees).
shippingAddress	Object	Required	Embedded address used for shipping (to maintain historical accuracy).
status	String	Required ('pending', 'shipped', 'delivered')	Current order status.
shippingFee	Decimal	Required, (>=0)	Fee charged for shipping.
phoneNumber	String	Required	Customer contact phone number.
promotionRef	ObjectId	Optional (Reference to promotions)	Link to the applied promotion, if any.
paymentRef	ObjectId	Reference to payments, Required	Link to the payment transaction document.

orderItems	Array of Objects	**Embedded** (1:n relationship)	Array of line items: { productId: String, name: String, quantity: Number, priceAtPurchase: Decimal }.

Example Document Structure (JSON):

```
{
    "_id": ObjectId("65c5c9f3e4b0c2a8f9d0c2e5"),
    "orderId": "ORD-20251123-001",
    "customerRef": ObjectId("65c5c9f3e4b0c2a8f9d0c2c3"),
    "orderDate": ISODate("2025-11-23T10:00:00Z"),
    "totalAmount": 155.00,
    "shippingAddress": {
        "street": "14 A B C",
        "city": "HCM"
    },
    "status": "delivered",
    "shippingFee": 5.00,
    "phoneNumber": "0888888888",
    "promotionRef": null,
    "paymentRef": ObjectId("65c5c9f3e4b0c2a8f9d0c2f6"),
    "orderItems": [
        {
        "productId": "PROD-A01",
        "name": "Birthday Cake",
        "quantity": 2,
        "priceAtPurchase": 50.00
        },
        {
        "productId": "PROD-B05",
        "name": "Tiramisu",
        "quantity": 1,
        "priceAtPurchase": 20.00
        }
    ]
}
```

4.2.4 Collection payments

This collection uses a single structure to accommodate both CashPayment and BankingPayment using a discriminator field and embedded details.

Attribute Name	Data Type	Value Constraints	Explanation
_id	ObjectId	Required, Unique	MongoDB's primary identifier.
paymentId	String	Required, Unique	Business-level payment ID.
paymentType	String	Required ('CashPayment' or 'BankingPayment')	Discriminator for payment method.
paymentDate	Date	Required	Date the payment was processed.
amount	Decimal	Required, (>0)	Amount of the transaction.

paymentDetails	Object	Required	**Embedded** details specific to the type. Object of {
			transactionCode: String
			bankName: String
			}

Example Document Structure (JSON - Banking Payment):

4.2.5 Collection: reviews

This collection stores customer reviews for products, referencing both the product and the customer.

Attribute Name	Data Type	Value Constraints	Explanation
_id	ObjectId	Required, Unique	MongoDB's primary identifier.
reviewId	String	Required, Unique	Business-level review ID.
productRef	ObjectId	Reference to products, Required	Link to the product being reviewed.
customerRef	ObjectId	Reference to accounts, Required	Link to the customer who left the review.

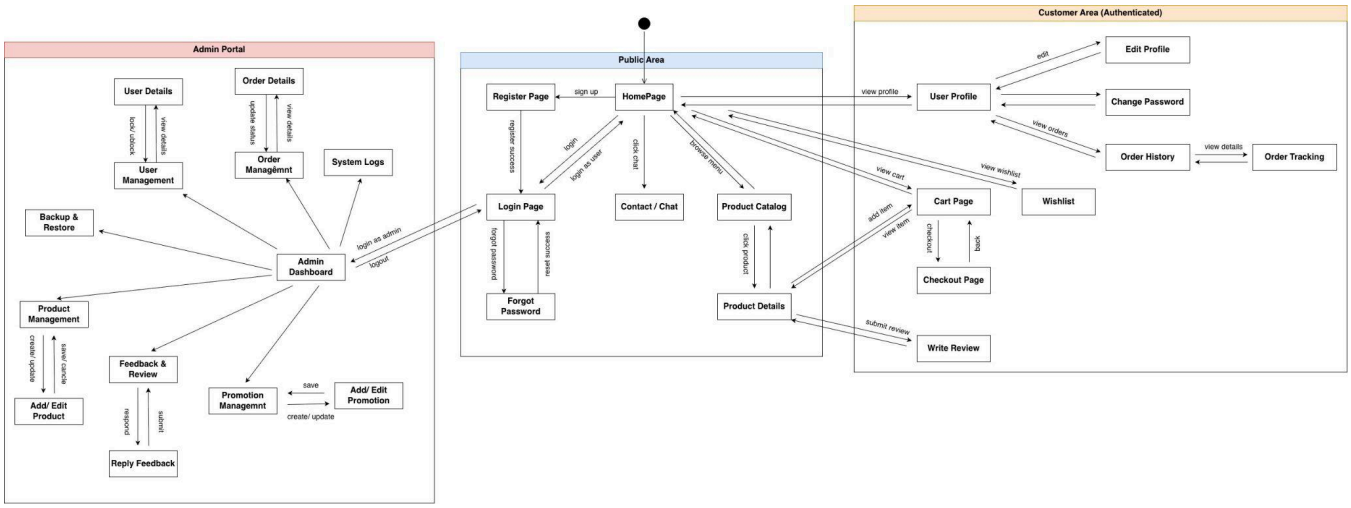
rating	Number	Required, Integer, Range 1 to 5	Star rating.
reviewText	String	Optional, Max Length 1000	Detailed text of the review.
isApproved	Boolean	Required, Default false	Status indicating if the review is public.
createdAt	Date	Required	Date the review was created.
adminFeedbackText	String	Optional	Internal feedback or response from admin.

Example Document Structure (JSON):

```
{
    "_id": ObjectId("65c5c9f3e4b0c2a8f9d0c3a7"),
    "reviewId": "REV-001",
    "productRef": ObjectId("65c5c9f3e4b0c2a8f9d0c2d4"),
    "customerRef": ObjectId("65c5c9f3e4b0c2a8f9d0c2c3"),
    "rating": 5,
    "reviewText": "I love this shop!",
    "isApproved": true,
    "createdAt": ISODate("2025-11-24T09:00:00Z"),
    "adminFeedbackText": null
}
```

5 User Interface and User Experience Design

5.1 Screen Diagram

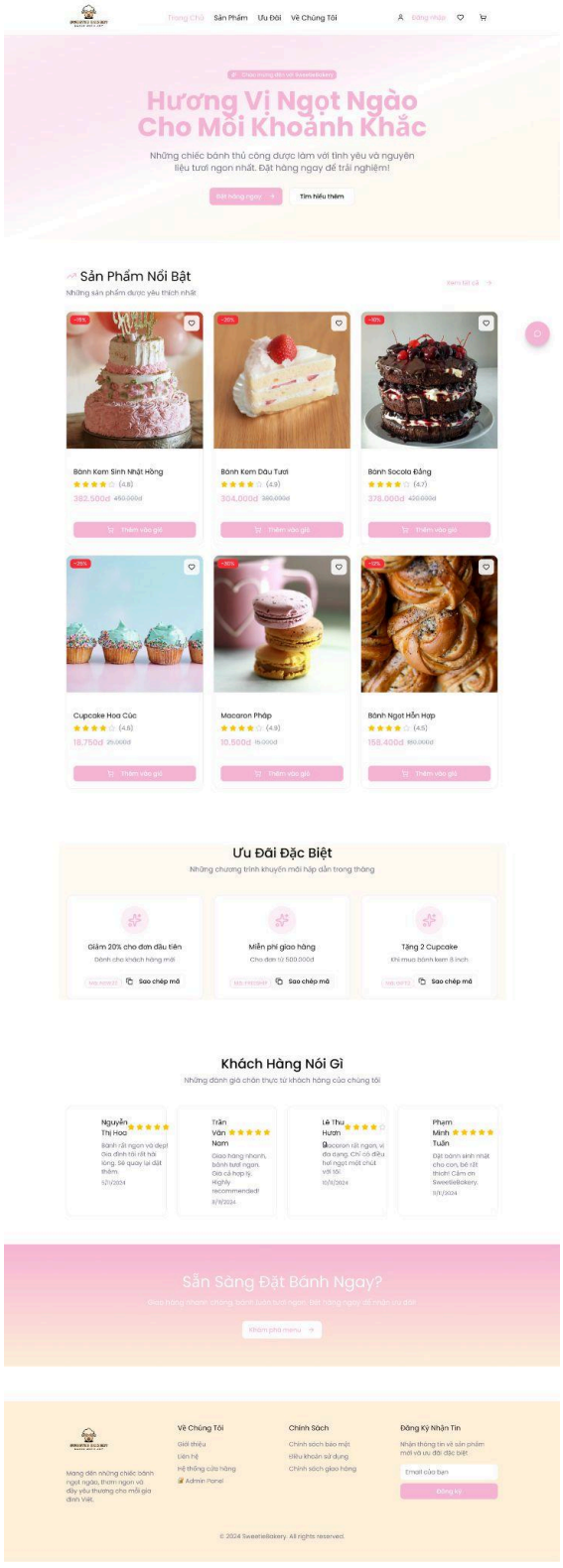


Seq	Area	Screen
1	**Public Area**	1. Home Page 2. Register Page 3. Login Page 4. Forgot Password 5. Contact / Chat 6. Product Catalog 7. Product Details
2	**Customer Area (Authenticated)**	8. User Profile 9. Edit Profile

		10. Change Password 11. Order History 12. Order Tracking 13. Wishlist 14. Cart Page 15. Checkout Page 16. Write Review
3	**Admin Portal**	17. Admin Dashboard 18. User Management 19. User Details 20. Order Management 21. Order Details 22. System Logs 23. Backup & Restore 24. Product Management 25. Add/ Edit Product 26. Feedback & Review 27. Reply Feedback 28. Promotion Management 29. Add/ Edit Promotion

5.2 Screen Specifications

5.2.1 Screen "Home Page"



Seq	Screen	Description
1	**Home Page**	**Overview:** The Home Page serves as the landing area for users, showcasing the brand, featured products, special offers, customer testimonials, and navigation options. **UI Components & Events handling:** **1. Global Navigation (Header)** **1.1. Logo** ● **Presentation Format:** The brand logo "Sweet Bakery" is positioned on the top-left. ● **Event Handling:** ○ **Click:** Redirects the user to the **Home Page** (refreshes the page if already on Home). **1.2. Navigation Menu** ● **Presentation Format:** Horizontal text links in the center of the header. ● **Event Handling:** ○ **Hover:** The text color changes (e.g., to pink) or an underline appears to indicate interactivity. ○ **Click:** Navigates the user to the respective page (e.g., Products Page, About Us Page). 1.3. User Actions (Account, Wishlist, Cart)

		• **Presentation Format:** Icons located on the top-right (User icon, Heart icon, Cart icon).
		• **Event Handling:**
		  ○ **Login/Account (User Icon):**
		  ■ **Click:** If not logged in, redirects to the **Login/Register** page. If logged in, opens a dropdown menu for Profile/Logout.
		  ○ **Wishlist (Heart Icon):**
		  ■ **Click:** Redirects to the **Wishlist** page.
		  ○ **Cart (Cart Icon):**
		  ■ **Click:** Redirects to the **Cart** page or opens a mini-cart side drawer.
		  ■ **Badge:** If items exist in the cart, a small number badge appears over the icon indicating quantity.
		
		2. Hero Section (Banner)
		
		2.1. Primary Call-to-Action (CTA) - Order Now
		
		• **Presentation Format:** A solid pink button with white text, located centrally under the main headline.
		• **Event Handling:**
		  ○ **Hover:** The button background color darkens slightly or scales up (1.05x).

		○ **Click:** Redirects the user to the full **Product Listing/Shop** page. **2.2. Secondary CTA - Learn More** ● **Presentation Format:** A white button with a border and dark text. ● **Event Handling:** ○ **Click:** Scrolls the page down to the "Featured Products" or "About Us" section. **3. Featured Products Section** **3.1. See All Link** ● **Presentation Format:** Small text link with an arrow icon on the right side of the section title. ● **Event Handling:** ○ **Click:** Redirects to the **Product Listing** page filtered by "Best Sellers" or "Featured." **3.2. Product Card Interaction** ● **Presentation Format:** A card containing the product image, discount badge (top-left), wish list icon (top-right), name, star rating, price, and an "Add to Cart" button. ● **Event Handling:** ○ **Click (on Image or Name):** Redirects to the specific **Product Detail** page.

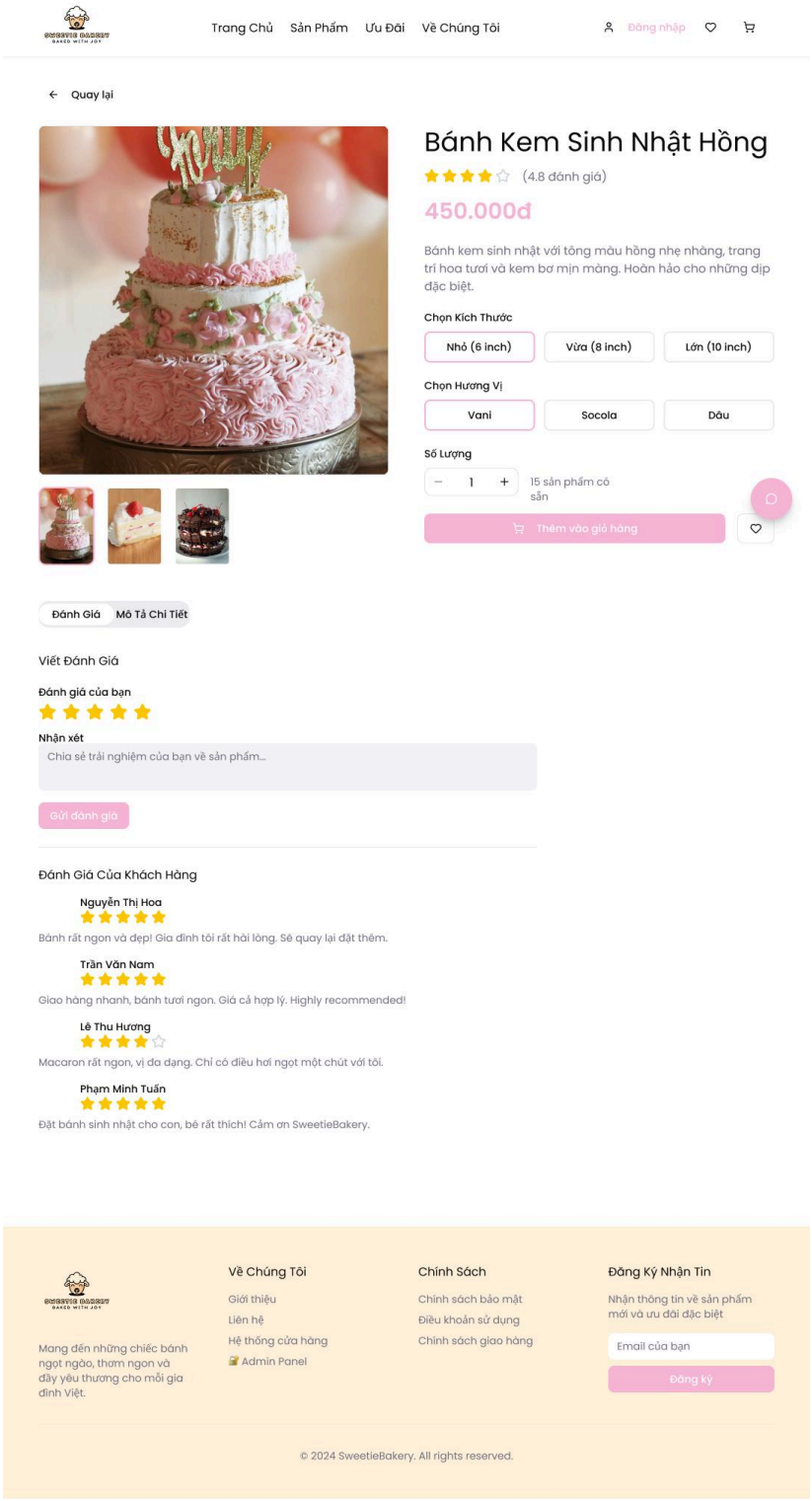
		○ **Hover:** The card may elevate (box-shadow increases) to indicate selection. **3.3. Wishlist Toggle (Heart Icon on Card)** ● **Presentation Format:** A small heart icon in the top-right corner of the product image. ● **Event Handling:** ○ **Click:** Toggles the state. ■ *Active:* Icon fills with red color; product is added to Wishlist database. ■ *Inactive:* Icon reverts to outline; product is removed from Wishlist. **3.4. Button (Add to Cart)** ● **Presentation Format:** A full-width pink button at the bottom of each product card. ● **Event Handling:** ○ **Click:** ■ Adds 1 unit of the product to the Shopping Cart. ■ Displays a "Success" toast notification (e.g., "Item added to cart"). ■ Updates the Cart Icon counter in the Header. **4. Floating Action Button (Right Side)** **4.1. Support/Messenger Button**

		• **Presentation Format:** A circular floating button (pink with headset icon) fixed to the middle-right of the viewport.
		• **Event Handling:**
		  ○ **Click:** Opens a chat widget (e.g., Facebook Messenger or Live Chat) to allow the user to talk to customer support.
		
		5. Special Offers Section
		
		5.1. Coupon Copy Action
		
		• **Presentation Format:** A card displaying a discount offer (e.g., "Giảm 20%") with a button labeled "Sao chép mã".
		• **Event Handling:**
		  ○ **Click:**
		  ■ Copies the specific coupon code (e.g., "SALE20") to the user's clipboard.
		  ■ Changes button text temporarily to "Đã sao chép" (Copied) to confirm the action.
		
		6. Testimonials Section
		
		(Note: Based on the static image, these look like display cards, but usually include navigation controls)
		
		6.1. Review Cards

		• **Presentation Format:** Cards displaying user avatar, name, star rating, date, and text review.
		• **Event Handling:**
		  ○ *Assumption (if carousel):* Users can swipe left/right on mobile or click hidden arrows to view more reviews.
		
		7. Bottom CTA Section
		
		7.1. Button (Explore Menu)
		
		• **Presentation Format:** A white button with an arrow icon on a pink gradient background.
		• **Event Handling:**
		  ○ **Click:** Redirects the user to the **Menu/Shop** page.
		
		8. Footer
		
		8.1. Quick Links
		
		• **Presentation Format:** Lists of text links organized by category.
		• **Event Handling:**
		  ○ **Click:** Redirects to the corresponding informational pages (Privacy Policy, Shipping Policy, etc.).
		
		8.2. Newsletter Subscription Form
		
		• **Presentation Format:** An input field labeled "Your email" and a button (Subscribe).

		• **Event Handling:**
		  ○ **Input Focus:** The user can type their email address.
		  ○ **Click "Subscribe":**
		  ■ *Validation:* Checks if the email format is valid. If invalid, shows an error message (red text).
		  ■ *Success:* If valid, sends the email to the backend API and displays a success message ("Thank you for subscribing!").

5.2.2 Screen "Product Detail Page"



Seq	Screen	Description
2	**Product Detail Page**	**Overview:** This page displays detailed information about a specific product (e.g., "Bánh Kem Sinh Nhật Hồng"), allows users to customize options (size, flavor, quantity), read reviews, and add the item to their cart. **UI Components & Events handling:** **1. Navigation & Header** **1.1. Back Button** ● **Presentation Format:** A text link with a left arrow icon positioned at the top-left of the content area, just below the main header. ● **Event Handling:** ○ **Click:** Navigates the user back to the previous screen (usually the **Product Listing** or **Home Page**). *(Note: The Global Header events—Logo, Menu, User Icons—remain the same as defined in the Home Page specification).* **2. Product Image Gallery (Left Column)** **2.1. Thumbnail Selection** ● **Presentation Format:** A row of small images (thumbnails) located beneath the large main product image. ● **Event Handling:** ○ **Click:**

		■ Updates the **Main Image** container to display the high-resolution version of the selected thumbnail. ■ Adds a visual border or opacity change to the selected thumbnail to indicate it is active. **2.2. Main Image Zoom (Optional)** ● **Presentation Format:** The large image displaying the currently selected view. ● **Event Handling:** ○ **Hover:** Activates a magnifier lens or zooms in on the specific area of the image for a detailed view. **3. Product Configuration (Right Column)** **3.1. Size Selection** ● **Presentation Format:** A group of buttons labeled "Nhỏ (6 inch)", "Vừa (8 inch)", and "Lớn (10 inch)". The selected option has a pink border/background. ● **Event Handling:** ○ **Click:** ■ Updates the visual state to highlight the selected button. ■ **Dynamic Update:** Updates the **Product Price** displayed at the top (e.g., 10 inch might cost more than 6 inch).

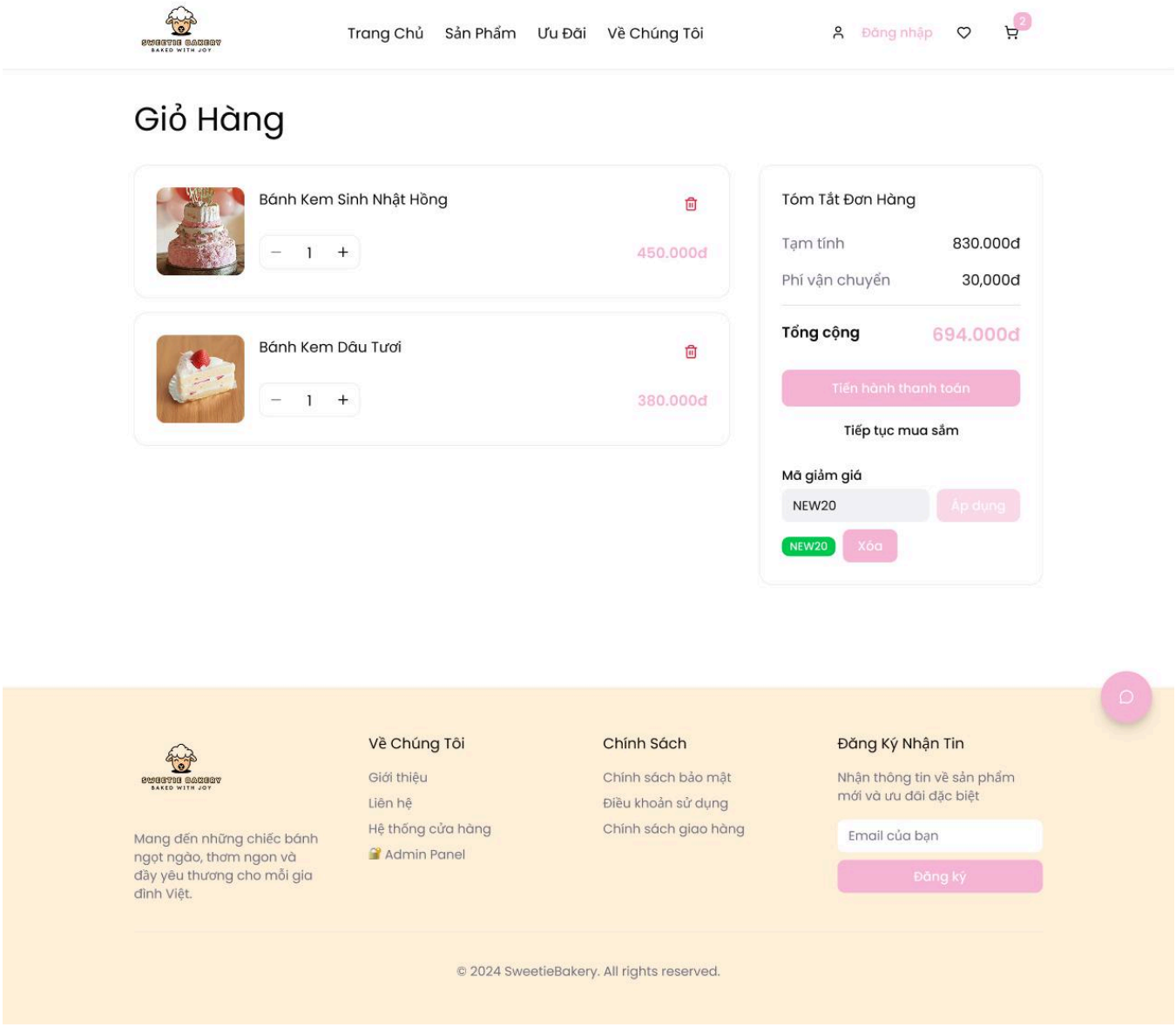
		■ Stores the selected size value in the temporary state. **3.2. Flavor Selection** ● **Presentation Format:** A group of buttons labeled "Vani", "Socola", "Dâu". The selected option has a pink border. ● **Event Handling:** ○ **Click:** ■ Highlights the clicked button. ■ Stores the selected flavor in the temporary state. **3.3. Quantity Selector** ● **Presentation Format:** A control with a Minus button (-), an Input field (displaying number), and a Plus button (+). Text next to it indicates stock availability (e.g., "15 sản phẩm có sẵn"). ● **Event Handling:** ○ **Click Minus (-):** Decreases quantity by 1. Disabled if quantity is 1. ○ **Click Plus (+):** Increases quantity by 1. Disabled if quantity equals available stock (15). ○ **Input Change:** Allows manual typing. Validates that input is numeric and within stock limits on blur. **4. Primary Actions**

		4.1. "Add to Cart" Button ● **Presentation Format:** A large, full-width pink button with a cart icon and text. ● **Event Handling:** ○ **Click:** ■ **Validation:** Checks if required options (Size, Flavor) are selected. ■ **Action:** Adds the specific configuration (Product + Size + Flavor + Qty) to the Cart. ■ **Feedback:** Displays a success notification (Toast) and updates the Cart Icon badge in the Header. 4.2. Wishlist Toggle (Heart Button) ● **Presentation Format:** A square button with a heart icon located next to the "Add to Cart" button. ● **Event Handling:** ○ **Click:** Toggles the product's presence in the user's Wishlist. Changes icon style (filled vs. outline) to reflect state. **5. Information Tabs ("Đánh Giá" / "Mô Tả Chi Tiết")** **5.1. Tab Switching** ● **Presentation Format:** Two toggle buttons/tabs. "Đánh Giá" is currently

		active (white background, bold text), while "Mô Tả Chi Tiết" is inactive (gray background).
		• **Event Handling:**
		  ○ **Click:**
		  ■ Changes the active tab style.
		  ■ Switches the content section below to show either the **Review List/Form** or the **Text Description**.
		
		6. Review Section ("Viết Đánh Giá")
		
		6.1. Star Rating Input ("Đánh giá của bạn")
		
		• **Presentation Format:** A row of 5 empty star icons.
		• **Event Handling:**
		  ○ **Hover:** Fills stars up to the cursor position (e.g., hovering over 4th star fills 4 stars).
		  ○ **Click:** Sets the rating value (e.g., 4/5). Stars remain filled to indicate selection.
		
		6.2. Review Text Input ("Nhận xét")
		
		• **Presentation Format:** A textarea with placeholder text "Chia sẻ trải nghiệm của bạn về sản phẩm...".
		• **Event Handling:**
		  ○ **Focus:** Allows user to type text.
		
		6.3. "Gửi đánh giá" (Submit Review) Button

		• **Presentation Format:** A small pink button.
		• **Event Handling:**
		  ○ **Click:**
		  ■ Validates that a star rating and text are provided.
		  ■ Submits data to backend.
		  ■ Displays a "Thank you" message and appends the new review to the list below (pending moderation if applicable).
		
		7. Footer & Floating Actions
		
		(Note: Footer events are identical to the Home Page).
		
		7.1. Chat Support
		
		• **Presentation Format:** A pink circular floating button with a chat icon on the right edge.
		• **Event Handling:**
		  ○ **Click:** Opens the live chat widget.

5.2.3 Screen "Shopping Cart Page"



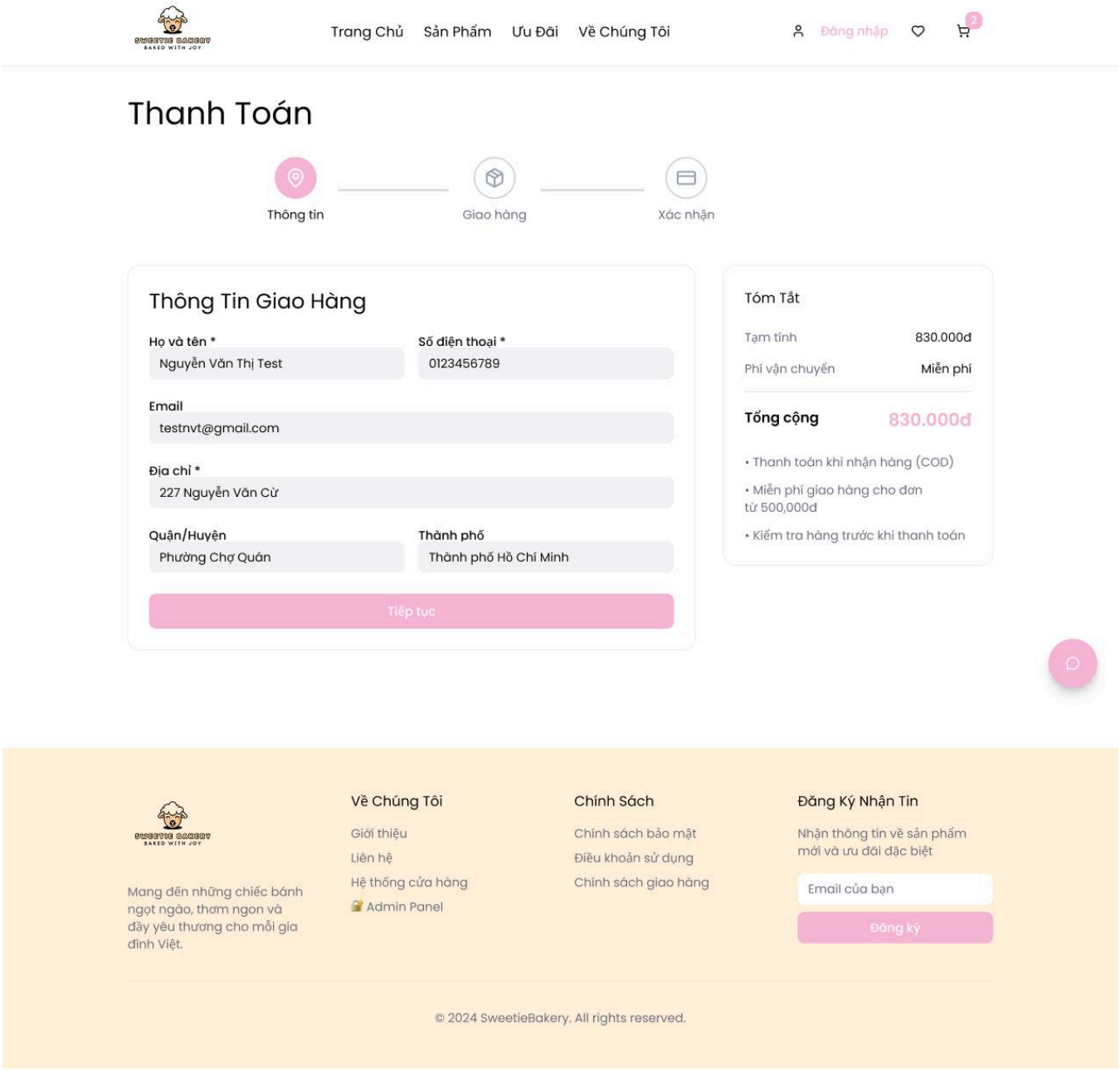
Seq	Screen	Description
3	**Shopping Cart Page**	Overview: This screen allows users to review items they intend to purchase, adjust quantities, remove items, apply discount codes, and proceed to checkout. It has two distinct states: empty and populated. **UI Components & Events handling:** **1. Cart Items List (Left Column – Populated State)** **1.1. Item Quantity Adjustment** ● **Presentation Format:** Each cart item card has a quantity control: a Minus button (-), a Number display, and a Plus button (+). ● **Event Handling:** ○ **Click Minus (-):** ■ **Logic:** Decreases the quantity of the specific item by 1. ■ **Constraint:** If quantity is 1, the button may be disabled or trigger a removal confirmation (depending on business logic). ■ **Update:** Recalculates the **Item Subtotal** and the **Order Summary** (Total). ○ **Click Plus (+):** ■ **Logic:** Increases the quantity by 1. ■ **Constraint:** Checks stock availability. If max stock is reached, disables button or

		shows a "Max stock reached" toast. ■ **Update:** Recalculates totals. **1.2. Remove Item (Trash Icon)** ● **Presentation Format:** A red trash bin icon located at the top-right of each item card. ● **Event Handling:** ○ **Click:** ■ **Action:** Removes the specific item card from the list. ■ **Update:** Recalculates the **Order Summary**. If the last item is removed, the screen transitions to the **Empty Cart State**. ■ **Feedback:** Optionally shows a "Undo" toast or a confirmation dialog before deletion. **2. Order Summary Panel (Right Column - Populated State)** **2.1. Proceed to Checkout ("Tiến hành thanh toán")** ● **Presentation Format:** A prominent pink button labeled "Tiến hành thanh toán". ● **Event Handling:** ○ **Click:** ■ **Validation:** Checks if the cart has items and if user is logged in (optional flow). ■ **Action:** Redirects the user to the **Checkout /**

		Shipping Information page. 2.2. Continue Shopping ("Tiếp tục mua sắm") ● **Presentation Format:** A text link or secondary button style below the main checkout button. ● **Event Handling:** ○ **Click:** Redirects the user back to the **Product Listing** or **Home Page**. 2.3. Apply Voucher Code ● **Presentation Format:** An input field labeled "Mã giảm giá" and an "Áp dụng" (Apply) button. ● **Event Handling:** ○ **Input:** User types a code (e.g., "NEW20"). ○ **Click "Áp dụng":** ■ **Validation:** Checks if the code is valid, active, and applicable to current items. ■ **Success:** Displays a success message/tag (e.g., green "NEW20" tag), deducts the discount amount from the "Tổng cộng" (Total), and disables the input/button to prevent double entry. ■ **Failure:** Displays an error message (e.g., "Invalid code"). 2.4. Remove Applied Voucher ● **Presentation Format:** A small "X" or "Xóa" button next to an applied voucher tag (e.g., the green "NEW20" tag).

		• **Event Handling:**
		  ○ **Click:**
		  ■ **Action:** Removes the discount from the calculation.
		  ■ **Update:** Reverts the "Tổng cộng" price to the original amount and re-enables the voucher input field.
		
		3. Empty Cart State
		
		3.1. "Khám phá sản phẩm" (Discover Products) Button
		
		• **Presentation Format:** A centered pink button displayed when the cart has zero items.
		• **Event Handling:**
		  ○ **Click:** Redirects the user to the **Product Listing / Shop** page to start adding items.
		
		4. Footer & Floating Actions
		
		(Note: Standard global elements)
		
		4.1. Chat Support
		
		• **Presentation Format:** A pink circular floating button with a chat icon.
		• **Event Handling:**
		  ○ **Click:** Opens the live chat widget.

5.2.4 Screen "Checkout Page"



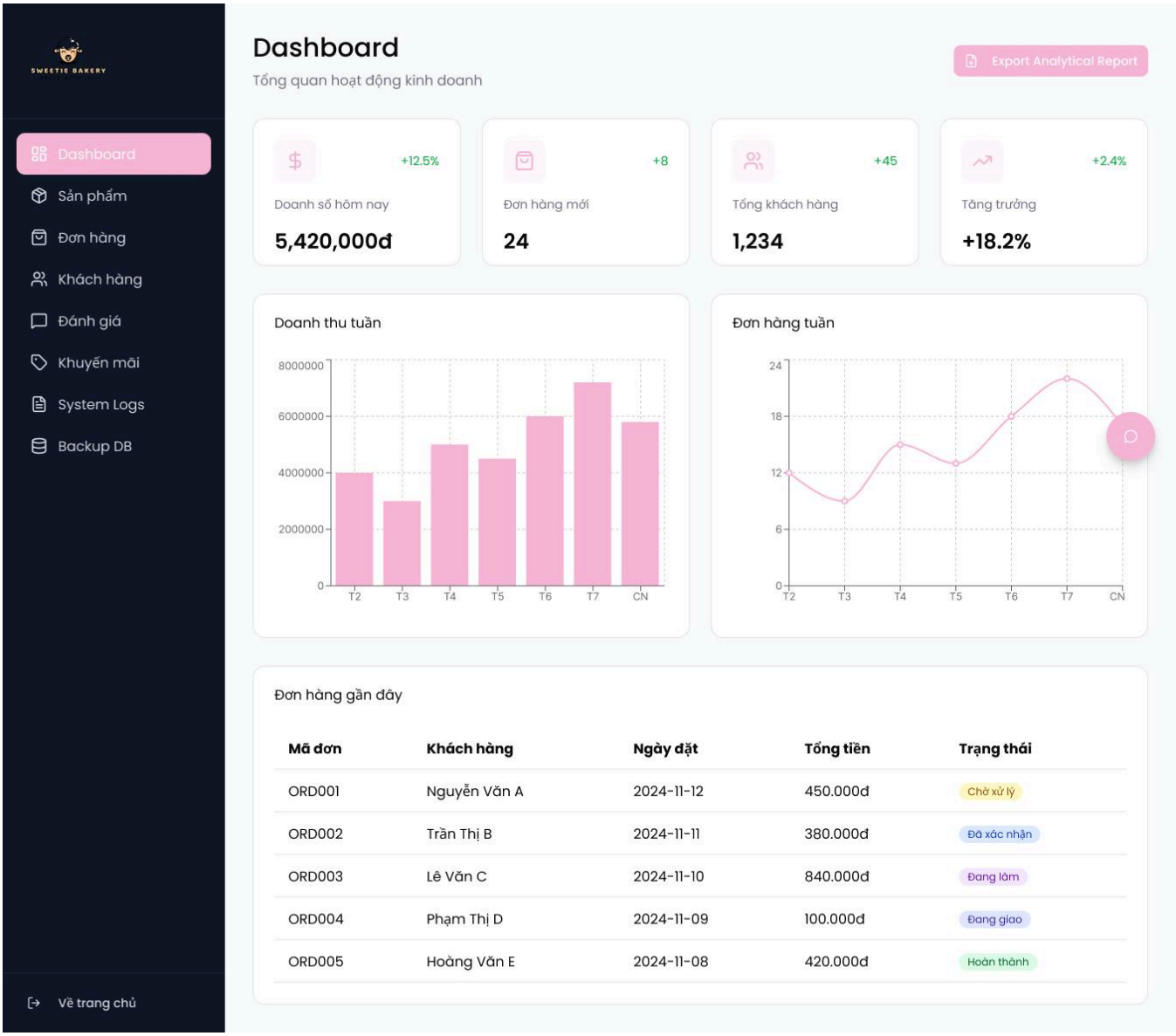
Seq	Screen	Description
4	**Checkout Page**	Overview: This is the first step of the checkout process. The user provides their shipping details to calculate final shipping costs (if applicable) and prepare the order for delivery. The screen displays a progress bar, a data entry form, and a summary of the costs. **UI Components & Events handling:** **1. Checkout Progress Indicator ("Thanh Toán")** **1.1. Progress Steps (Thông tin - Giao hàng - Xác nhận)** ● **Presentation Format:** Three circular icons connected by a line. ○ **Step 1 (Location Icon - "Thông tin"):** Pink background (Active state). ○ **Step 2 (Box Icon - "Giao hàng"):** Gray/White background (Inactive/Pending state). ○ **Step 3 (Card Icon - "Xác nhận"):** Gray/White background (Inactive/Pending state). ● **Event Handling:** ○ **Click (Step 1):** No action (current step). ○ **Click (Step 2 & 3):** Disabled. Users cannot jump ahead without completing the current form. **2. Delivery Information Form ("Thông Tin Giao Hàng") - Left Panel**

		2.1. Text Input Fields (Name, Phone, Email, Address) ● **Presentation Format:** Standard text input fields with gray backgrounds. Mandatory fields are marked with an asterisk (*). ○ **"Họ và tên " (Full Name):* Pre-filled if user is logged in (e.g., "Nguyễn Văn Thị Test"). ○ **"Số điện thoại " (Phone):* Numeric input. ○ **"Email":** Standard text input. ○ **"Địa chỉ " (Street Address):* Standard text input. ○ **"Quận/Huyện" & "Thành phố" (District/City):** Standard text inputs (or potentially dropdowns depending on implementation complexity). ● **Event Handling:** ○ **Focus:** The input field border highlights (e.g., turns pink or darker gray) to indicate activity. ○ **Blur (Lose Focus):** Triggers inline validation. ■ *If Mandatory & Empty:* Displays error message "This field is required" in red below the field. ■ *Phone Validation:* Checks if the input contains 10-11 digits. If invalid, shows "Invalid phone number". ■ *Email Validation:* Checks regex format. If invalid, shows "Invalid email format". **2.2. "Tiếp tục" (Continue) Button**

		• **Presentation Format:** A full-width, large pink button located at the bottom of the form.
		• **Event Handling:**
		  ○ **Click:**
		  ■ **Validation:** Checks all mandatory fields (*).
		  ■ **Failure:** If validation fails, scrolls to the first error field and displays error messages.
		  ■ **Success:** Saves the delivery address to the session/state and navigates the user to **Step 2: Shipping/Payment ("Giao hàng")**.
		
		3. Order Summary ("Tóm Tắt") - Right Panel
		
		3.1. Summary Details (Static Display)
		
		• **Presentation Format:** A card displaying "Tạm tính" (Subtotal), "Phí vận chuyển" (Shipping Fee), and "Tổng cộng" (Total).
		• **Event Handling:**
		  ○ **Page Load:**
		  ■ **Subtotal:** Fetched from the previous Cart state (e.g., 830.000đ).
		  ■ **Shipping Fee:** Defaults to a value or "Miễn phí" (Free) based on business logic (See note in image: "Miễn phí giao hàng cho đơn từ 500,000đ"). Since the subtotal is > 500k, this displays "Miễn phí".
		  ■ **Total:** Calculated as Subtotal + Shipping.

		4. Footer & Floating Actions *(Note: Standard global elements)* **4.1. Chat Support** ● **Presentation Format:** A pink circular floating button with a chat icon. ● **Event Handling:** ○ **Click:** Opens the live chat widget for support. **4.2. Newsletter Subscription (Footer)** ● **Presentation Format:** Input field "Email của bạn" and "Đăng ký" button. ● **Event Handling:** ○ **Click:** Validates email and subscribes user to the newsletter.

5.2.5 *Screen "Admin Dashboard"*



Seq	Screen	Description
5	**Admin Dashboard**	Overview: This is the central hub for the administrator. It provides a high-level overview of business performance, including revenue, order statistics, customer growth, and visual charts. It also provides quick access to recent orders and system navigation. **UI Components & Events handling:** **1. Sidebar Navigation (Left Panel)** **1.1. Navigation Links (Sản phẩm, Đơn hàng, Khách hàng, etc.)** ● **Presentation Format:** A vertical list of menu items with icons and text on a dark blue background. ○ **Active State:** The current page ("Dashboard") is highlighted with a pink background and white text. ○ **Inactive State:** Other items (e.g., "Sản phẩm", "Đơn hàng") have a transparent background with gray/white text. ● **Event Handling:** ○ **Hover:** The background color of the list item lightens slightly to indicate interactivity. ○ **Click:** ■ Updates the **Main Content Area** on the right to display the corresponding module (e.g., clicking "Đơn hàng" loads the Order Management Table).

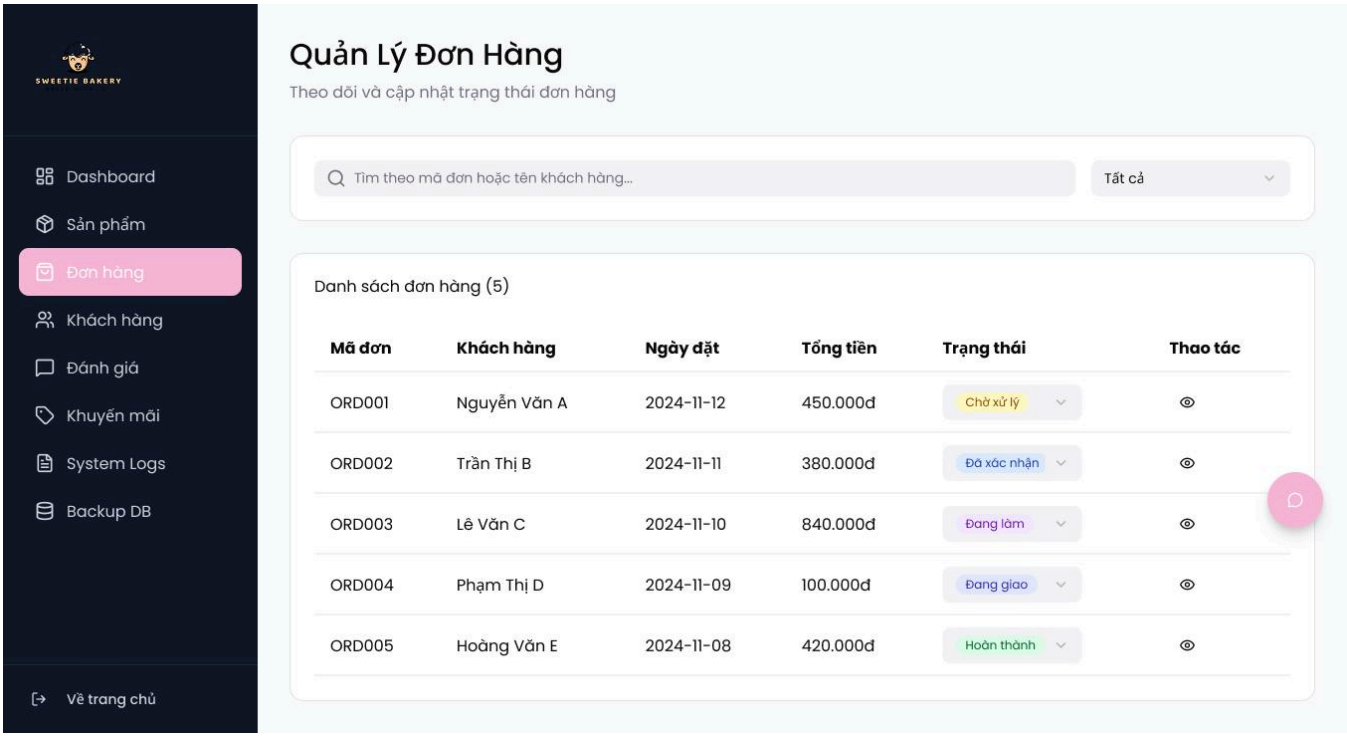
		■ Changes the active state styling to the clicked item. **1.2. "Về trang chủ" (Back to Home) Link** ● **Presentation Format:** A text link with an exit icon located at the very bottom of the sidebar. ● **Event Handling:** ○ **Click:** Logs the admin out (optional) or simply redirects the browser to the user-facing **Home Page** of the bakery website. **2. Dashboard Header** **2.1. "Export Analytical Report" Button** ● **Presentation Format:** A solid pink button with a document icon located at the top-right of the content area. ● **Event Handling:** ○ **Hover:** Button background darkens. ○ **Click:** ■ Triggers a backend process to generate a summary report (PDF or Excel format) containing the current dashboard data. ■ Initiates a file download for the user. **3. Key Metrics Cards (Top Row)** **3.1. Statistic Cards (Revenue, Orders, Customers, Growth)**

		• **Presentation Format:** Four rectangular white cards displaying an icon, a label (e.g., "Doanh số hôm nay"), a main data value (e.g., "5,420,000đ"), and a percentage indicator (e.g., "+12.5%" in green).
		• **Event Handling:**
		  ○ **Hover:** The card elevates (shadow effect) to indicate it is clickable.
		  ○ **Click:** Redirects the admin to the detailed report page for that specific metric (e.g., Clicking the "Revenue" card goes to the detailed Financial Report page).
		
		4. Data Visualization (Middle Row)
		
		4.1. Weekly Revenue Chart ("Doanh thu tuần")
		
		• **Presentation Format:** A Bar Chart displaying revenue figures for days T2 through CN (Monday to Sunday).
		• **Event Handling:**
		  ○ **Hover (over a bar):** A tooltip appears displaying the exact revenue amount for that specific day (e.g., "T6: 6,000,000đ").
		
		4.2. Weekly Orders Chart ("Đơn hàng tuần")
		
		• **Presentation Format:** A Line/Curved Chart displaying the number of orders per day.
		• **Event Handling:**

		○ **Hover (over a data point):** A tooltip appears displaying the exact number of orders for that specific day. **5. Recent Orders Table ("Đơn hàng gần đây")** **5.1. Order Row Interaction** ● **Presentation Format:** A data table with columns: Order ID (Mã đơn), Customer Name, Date, Total, and Status. ● **Event Handling:** ○ **Hover:** The specific row background turns light gray. ○ **Click:** Redirects to the **Order Detail View** for that specific Order ID (e.g., clicking row "ORD001" opens the details for Order #001). **5.2. Status Badges** ● **Presentation Format:** Color-coded pills indicating the state of the order. ○ *Yellow:* "Chờ xử lý" (Pending) ○ *Blue:* "Đã xác nhận" (Confirmed) ○ *Purple:* "Đang làm" (Processing/Baking) ○ *Indigo:* "Đang giao" (Shipping) ○ *Green:* "Hoàn thành" (Completed) ● **Event Handling:** ○ **Click:** Opens a quick-action dropdown to update the status (e.g., changing "Pending" to

		"Confirmed") without leaving the dashboard. **6. Floating Actions** **6.1. System Chat/Support** ● **Presentation Format:** A pink circular floating button with a chat icon fixed to the middle-right edge of the screen. ● **Event Handling:** ○ **Click:** Opens an internal chat widget (for admin-to-staff communication) or a support ticket system.

5.2.6 Screen "Order Management (Admin)"



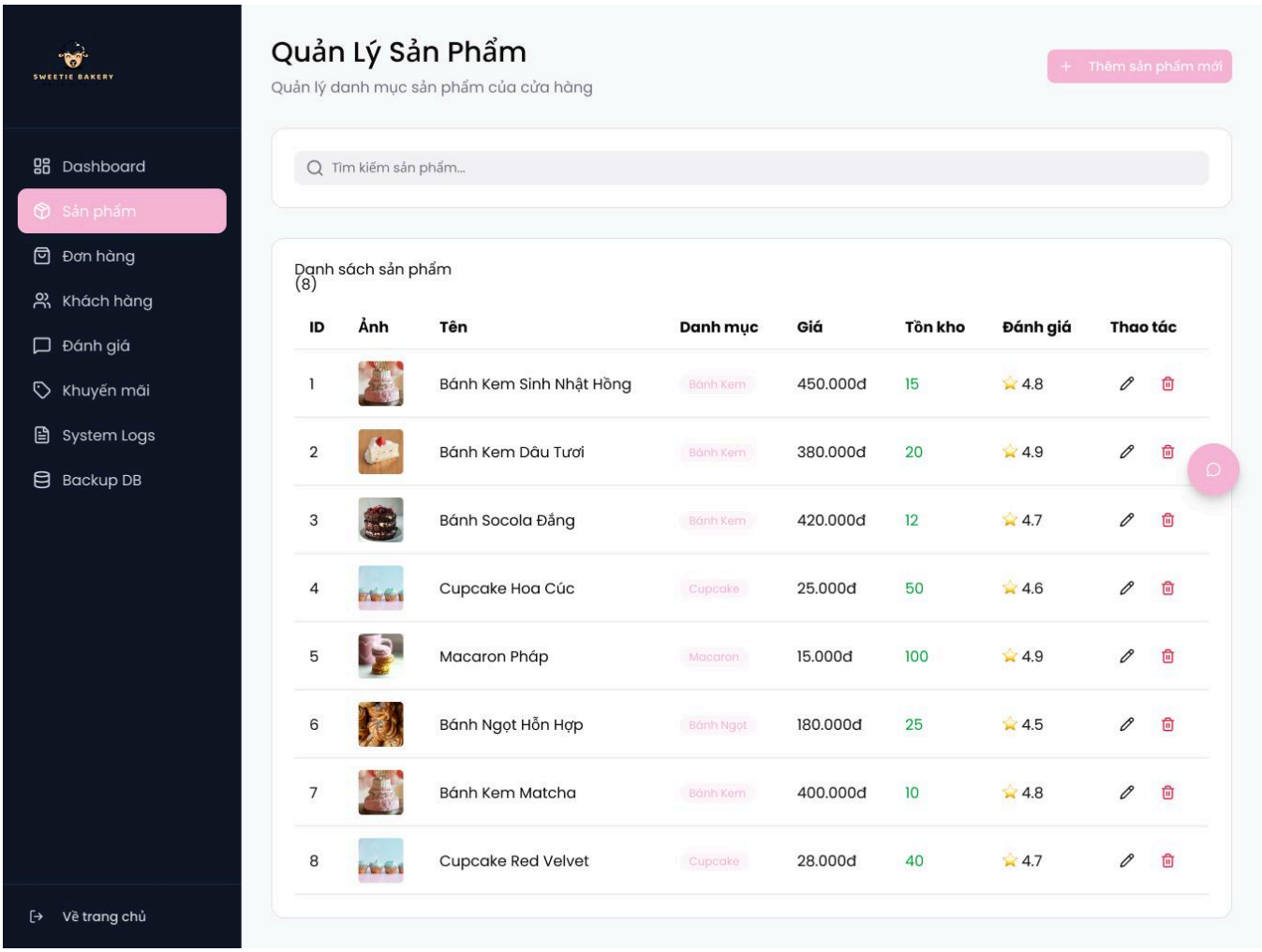
Seq	Screen	Description
6	**Order Management (Admin)**	Overview: This screen allows the administrator to view a list of all customer orders, search for specific transactions, filter by status, and quickly update the status of orders without entering the details page. **UI Components & Events handling:** **1. Sidebar Navigation (Left Panel)** **1.1. Navigation Links** ● **Presentation Format:** A vertical menu. The **"Đơn hàng"** (Orders) item is currently **Active** (Pink background, white text). Other items ("Dashboard", "Sản phẩm", etc.) are inactive (Transparent background, white/gray text). ● **Event Handling:** ○ **Click:** Navigates to the respective module (e.g., clicking "Khách hàng" loads the Customer Management page). **1.2. "Về trang chủ" (Back to Home)** ● **Presentation Format:** Link with an exit icon at the bottom of the sidebar. ● **Event Handling:** ○ **Click:** Redirects the admin to the user-facing Home Page. **2. Search & Filter Section (Top Card)**

		2.1. Search Input ("Tìm theo mã đơn...")
		
		• **Presentation Format:** A large text input field with a magnifying glass icon on the left. Placeholder text: "Tìm theo mã đơn hoặc tên khách hàng..." (Search by order ID or customer name).
		• **Event Handling:**
		  ○ **Input/Typing:** Accepts alphanumeric characters.
		  ○ **Enter Key / Stop Typing:** Triggers a search query to the backend. The **Order List Table** below refreshes to show only rows matching the Order ID or Customer Name.
		
		2.2. Status Filter Dropdown ("Tất cả")
		
		• **Presentation Format:** A dropdown menu located to the right of the search bar. Default value is "Tất cả" (All).
		• **Event Handling:**
		  ○ **Click:** Expands a list of statuses (Pending, Confirmed, Processing, Shipping, Completed).
		  ○ **Select Option:**
		  ■ Updates the dropdown label to the selected status.
		  ■ Filters the **Order List Table** to show only orders with that specific status.
		
		3. Order List Table ("Danh sách đơn hàng")
		
		3.1. Table Structure

		• **Presentation Format:** A table with 6 columns: "Mã đơn" (Order ID), "Khách hàng" (Customer), "Ngày đặt" (Date), "Tổng tiền" (Total), "Trạng thái" (Status), and "Thao tác" (Action).
		• **Event Handling:**
		  ○ **Hover Row:** The background of the row turns light gray to indicate focus.
		
		3.2. Status Update Dropdown (In-Row)
		
		• **Presentation Format:** Inside the "Trạng thái" column, the status is displayed as a color-coded badge inside a clickable dropdown container (indicated by the down arrow v).
		  ○ *ORD001:* Yellow "Chờ xử lý" (Pending)
		  ○ *ORD002:* Blue "Đã xác nhận" (Confirmed)
		  ○ *ORD003:* Purple "Đang làm" (Processing)
		  ○ *ORD004:* Indigo "Đang giao" (Shipping)
		  ○ *ORD005:* Green "Hoàn thành" (Completed)
		• **Event Handling:**
		  ○ **Click:** Opens a dropdown menu showing available next statuses.
		  ○ **Select Status:**
		  ■ Triggers an API call to update the status of that specific order ID.
		  ■ **Feedback:** Displays a "Success" toast notification

		(e.g., "Order status updated"). ■ **Visual Update:** The badge color and text change immediately to reflect the new status. **3.3. "View Details" Action (Eye Icon)** ● **Presentation Format:** An eye icon located in the "Thao tác" column. ● **Event Handling:** ○ **Click:** Redirects the admin to the **Order Details Page** for that specific transaction to view items, shipping address, and specific logs. **4. Floating Actions** **4.1. System Chat** ● **Presentation Format:** A pink circular floating button with a chat icon on the right edge. ● **Event Handling:** ○ **Click:** Opens the internal support chat or system notification widget.

5.2.7 Screen "Product Management (Admin)"



Seq	Screen	Description
7	**Product Management (Admin)**	Overview: This screen allows the administrator to view the full inventory of products, search for specific items, create new products, edit existing details, and delete items from the catalog. **UI Components & Events handling:** **1. Sidebar Navigation (Left Panel)**

		1.1. Navigation Links
		
		• **Presentation Format:** A vertical menu. The **"Sản phẩm"** (Products) item is currently **Active** (Pink background, white text). Other items are inactive (Transparent background, white/gray text).
		• **Event Handling:**
		  ○ **Click:** Navigates to the respective module (e.g., clicking "Dashboard" returns to the main overview).
		
		1.2. "Về trang chủ" (Back to Home)
		
		• **Presentation Format:** Link with an exit icon at the bottom of the sidebar.
		• **Event Handling:**
		  ○ **Click:** Redirects the admin to the user-facing Home Page.
		
		2. Header & Primary Actions
		
		2.1. "+ Thêm sản phẩm mới" (Add New Product) Button
		
		• **Presentation Format:** A solid pink button with a "plus" (+) icon and white text located at the top-right.
		• **Event Handling:**
		  ○ **Hover:** Background color darkens slightly.
		  ○ **Click:** Redirects the admin to the **Create Product Page** (a form to input name, price, image, etc.).

		3. Search Section **3.1. Search Input ("Tìm kiếm sản phẩm...")** ● **Presentation Format:** A wide text input field with a gray background and a search (magnifying glass) icon on the left. ● **Event Handling:** ○ **Input/Typing:** Accepts alphanumeric characters. ○ **Keyup (Debounce):** As the user types, the **Product List Table** automatically filters rows to show only products matching the Name or ID entered. **4. Product List Table ("Danh sách sản phẩm")** **4.1. Row Interaction** ● **Presentation Format:** A data table displaying: ID, Image (Avatar), Name, Category (Badge), Price, Stock (Green text), Rating (Star icon), and Actions. ● **Event Handling:** ○ **Hover:** The row background turns light gray to indicate focus. **4.2. Category Badge** ● **Presentation Format:** A pill-shaped badge with a light pink background (e.g., "Bánh Kem", "Cupcake"). ● **Event Handling:**

		○ **Click:** (Optional) Filters the table to show only products belonging to that specific category. **4.3. "Edit" Action (Pencil Icon)** ● **Presentation Format:** A dark gray pencil icon located in the "Thao tác" (Actions) column. ● **Event Handling:** ○ **Click:** Redirects the user to the **Edit Product Page** for that specific Item ID, pre-filling the form with existing data. **4.4. "Delete" Action (Trash Icon)** ● **Presentation Format:** A red trash can icon located in the "Thao tác" (Actions) column. ● **Event Handling:** ○ **Click:** ■ **Action:** Triggers a **Confirmation Modal** (e.g., "Are you sure you want to delete 'Bánh Kem Sinh Nhật Hồng'?"). ■ **Confirm:** Calls the API to soft-delete the item, removes the row from the table, and displays a "Success" toast. ■ **Cancel:** Closes the modal without changing data. **5. Floating Actions**

		5.1. System Chat ● **Presentation Format:** A pink circular floating button with a chat icon on the right edge. ● **Event Handling:** ○ **Click:** Opens the internal support chat or system notification widget.