AI4STYLE

Use-Case Specification

Version 1.3

Revision History

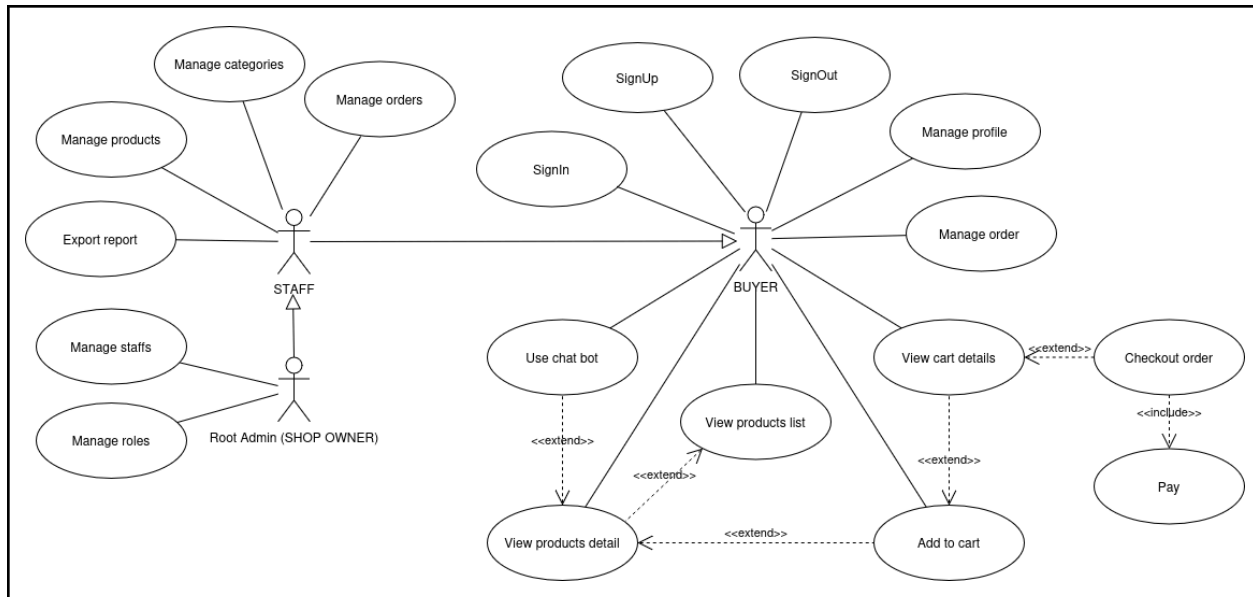| Date	Version	Description	Author
09/11/2025	1.0	First completed version	All team's members
27/11/2025	1.1	Merge use-case Checkout order + Place order → Checkout order	Tạ Xuân Trường
07/12/2025	1.2	Add use-case: role management	Tạ Xuân Trường
22/12/2025	1.3	Update name use-case: Browse product → View product list	Tạ Xuân Trường

AI4STYLE	Version: 1.3
Use-Case Specification	Date: 22/12/2025
UCS-A4S	

Table of Contents

1. Use-case Model

2. Use-case Specifications

2.1 Use-case: Sign-in

Use case Name	Sign in an existing account
Brief description	This use case describes how a User can sign in an account they created.
Actors	Buyer/Staff/Admin (User)
Basic Flow	1. At sign-in page, user enters email and password 2. System verify this credential 3. Direct user to appropriate homepage based on role

Alternative Flows	**Alternative flow 1: User enters the wrong email**
	1. After step #2, system show a error message
	2. User has 2 options:
	   o Go back to #1, enter correct email
	   o Click to sign-up tab for sign up new account
	
	Alternative flow 2: User forgets their password
	1. After step #2, System show a error message
	2. User has 2 options:
	   o Go back to #1, enter correct password
	   o Click to forget-password button, for reset password

AI4STYLE	Version: 1.3
Use-Case Specification	Date: 22/12/2025
UCS-A4S	

Pre-conditions	User clicks on the login button
	- At default, admin/staff need to sign in before going to admin page
Post-conditions	The user is successfully signed in to an account they created and redirected to a corresponding homepage

2.2 Use-case: Sign-up

Use case Name	Sign up
Brief description	This use case describes how a new buyer registers an account by providing required information. The system validates the input, ensures the email is not already registered, creates the account, and automatically logs the buyer in if successful.
Actors	Buyer
Basic Flow	1. Buyer navigates to the Sign-Up page
	2. Buyer enters required information on the 'Full name', 'Email', 'Password' fields
	3. The system performs input validation
	4. Buyer clicks the 'Sign Up' button
	5. System verifies that the email is not already registered
	6. System creates a new buyer account
	7. System logs the buyer in automatically
	8. System redirects the buyer to the homepage
Alternative Flows	**Alternative flow 1: Invalid or incomplete input**
	1. From Step #3 in basic flow, the system detects invalid or missing information ('empty field', 'invalid email format', 'weak password')
	2. System displays appropriate error messages near the fields
	3. The 'Sign Up' button remains disabled or submission is blocked
	4. Buyer corrects the information
	5. Continue step #2 in the basic flow
	Alternative flow 2: Email already registered
	1. From Step #5 in the Basic Flow, the system detects the email already exists
	2. System displays an error message: "Email already registered"
	3. Buyer chooses one of the following options:
	● Enter a different email → Continue step #2 in the basic flow
	● Click 'Sign In' button → System redirects to Login page
	Alternative flow 3: System error
	1. From Step #6 in the Basic Flow, the system encounters an internal error and cannot create an account.
	4. System displays message: "An error occurred. Please try again later."
	5. Buyer may retry or leave the 'Sign Up' page.
	6. Use case ends without account creation.
Pre-conditions	Buyer goes to the 'Sign In' page at *__domain-name/sign-in__*
Post-conditions	The buyer successfully creates a new account using an email.

2.3   Use-case: Sign-out

Use case Name	Sign out
Brief description	This use case describes how an authenticated user logs out of the application. The system terminates the current session, clears authentication data (such as tokens or cookies), and redirects the user to the appropriate public page.
Actors	Buyer/Staff/Admin (User)
Basic Flow	1. User clicks on the avatar button on the Navbar. 2. System displays a dropdown menu. 3. User clicks on the Logout option in the dropdown. 4. System receives the logout request. 5. System invalidates the user's session. 6. System clears any user-specific cached data in the client (if applicable). 7. System redirects the user to the Login page or the public homepage.
Alternative Flows	**Alternative flow 1: Session already expired** 1. Occurs from Step #4 in Basic Flow. 2. System detects the session is already invalid or expired. 3. System clears any remaining local authentication data. 4. System redirects user to the Login page. 5. Use case ends. **Alternative flow 2: System error** 1. Occurs from Step #5 in Basic Flow. 2. System encounters an error while invalidating the session (e.g., server timeout, internal exception). 3. System still attempts to clear local authentication data. 4. System displays error message: "An error occurred while logging out. Please try again." 5. User may retry the logout action or navigate away manually. 6. Use case ends.
Pre-conditions	User is authenticated and currently logged in. Avatar button is visible on the Navbar.
Post-conditions	User's session is invalidated. Authentication data is cleared from the client. User is redirected to homepage.

2.4   Use-case: Manage-profile

Use case Name	Manage-profile
Brief description	This use case describes how an authenticated User (Customer, Staff, or Shop Owner) can view, update their personal information, and review their recent activity (e.g., Order History).
Actors	Buyer / Staff / Shop Owner (User)

Basic Flow	1. User is successfully signed in and clicks the Profile icon or navigates to the Profile page. 2. The system displays the Profile Information (Full Name, Email, Phone Number, Address, User ID, Member Status) and other relevant sections (e.g., Order History). 3. User clicks the "Edit" button (located near Profile Information) to modify their details. 4. The system enables input fields for editing. 5. User updates the desired information (e.g., Address, Phone Number). 6. Uer clicks the "Save" or "Update" button. 7. The system validates the input. 8. The system updates the information in the database. 9. The system displays a success message: "Profile updated successfully."
Alternative Flows	**Alternative flow 1: Invalid/Incomplete Input** 1. From Step #7 in Basic Flow, the system detects invalid or missing information (e.g., empty phone number, invalid email format). 2. System displays appropriate error messages near the fields. 3. User corrects the information, continue Step #6. **Alternative flow 2: User Changes Profile Picture** 1. From Step #2, User clicks the camera icon on the profile picture. 2. System prompts User to upload a new image. 3. User selects and uploads an image. 4. System crops/validates the image and updates the profile picture. 5. Alternative flow 3: User Logs Out 6. From Step #2, User clicks the "Logout" button. 7. The system immediately terminates the current session. 8. System redirects the User to the homepage or login page.
Pre-conditions	The User must be successfully signed in to their account.
Post-conditions	The User's personal information is successfully updated in the system, or the User successfully ends their session.

2.5   Use-case: Manage orders (buyer)

Use case Name	Manage orders (buyer)
Brief description	This use case describes how the Buyer can view, track, and manage their placed orders within the system.
Actors	Buyer (User)
Basic Flow	1. The buyer navigates to the "My Orders" page from the main menu or profile section. 2. The system displays a list of all orders placed by the buyer, including order ID, date, total amount, and status (e.g., Pending, Processing, Shipped, Completed). 3. The buyer clicks on a specific order to view its details.

	4. The system displays the detailed order information, including product list, quantities, prices, delivery address, and payment method. 5. If the order is still in a cancellable state (e.g., "Pending" or "Processing"), the buyer clicks on the "Cancel Order" button. 6. The system confirms the cancellation and updates the order status to "Cancelled."
Alternative Flows	**Alternative Flow 1: Buyer views only completed or cancelled orders** 1. From #2 of the Basic Flow, the buyer clicks on the "Filter" or "Sort" option. 2. The buyer selects "Completed Orders" or "Cancelled Orders." 3. The system displays only the orders that match the selected filter criteria. 4. Continue with #3 of the Basic Flow. **Alternative Flow 2: Buyer tracks a shipped order** 1. From #3 of the Basic Flow, if the selected order has a status of "Shipped," the buyer clicks on the "Track Order" button. 2. The system displays the real-time shipment tracking information (e.g., courier, tracking ID, delivery status). 3. Continue with #4 of the Basic Flow. **Alternative Flow 3: Buyer reorders a completed purchase** 1. From #3 of the Basic Flow, if the selected order has a status of "Completed," the buyer clicks on the "Reorder" button. 2. The system adds all items from that order into the buyer's current shopping cart. 3. The system notifies the buyer that the items have been added successfully. 4. The user may then proceed to checkout as in the "Checkout and Payment" use case.
Pre-conditions	● The buyer is logged into their account. ● The buyer has at least one existing order in the system.
Post-conditions	The buyer successfully views, tracks, or manages their orders and the system updates any changes accordingly.

2.6   Use-case: View product list

Use case Name	View products list
Brief description	This use case describes how the User browses available products in the store, including viewing product lists, applying filters, and searching for products. The System displays the product list and allows the User to select a product to view detailed information.
Actors	Buyer/Staff/Admin (User)
Basic Flow	1. User navigates to the product listing page. 2. System displays available products. 3. User scrolls through the product list. 4. User enter keywords in search bar. 5. System displays matching search results. 6. User apply filters (category, size, color, price).

	7. System updates and displays filtered product results.

Alternative Flows	**Alternative flow 1: No products available**
	1. From Step #2 in the Basic Flow, the System detects there are no products available in the database.
	2. System displays message: "No products available."
	3. User chooses one of the following options:
	   ● Leave the browsing page → Use case ends
	   ● Stay on the page and retry (refresh) → Continue at Step #2 in the Basic Flow
	
	Alternative flow 2: No results after filtering/search
	1. From step #5 or #7 in the Basic Flow, the System detects that no products match the search or filter criteria.
	2. System displays message: "No products match your search."
	3. User may clear filters or modify search terms.
	4. Continue at Step #4 or Step #6 in the Basic Flow.
	
	Alternative flow 3: System error
	1. From any step in the Basic Flow, the System encounters an internal error and cannot load product data.
	2. System displays message: "An error occurred. Please try again later."
	3. User may refresh the page or return later.
	4. User chooses one of the following options:
	   ● Leave the browsing page → Use case ends
	   ● Stay on the page and retry (refresh) → Continue at Step #2 in the Basic Flow

Pre-conditions	User has accessed the website or product browsing section.
Post-conditions	The user successfully browses through available products by scrolling.

2.7 Use-case: View product detail

Use case Name	View a product
Brief description	This use case describes how the user can view a product's information.
Actors	Buyer/Staff/Admin (User)
Basic Flow	1. At the homepage, the user sees multiple lists of items such as Best Sellers or Recently Added. 2. The user clicks on any item image or name. 3. The system redirects the user to the individual product information page. 4. The system displays the item's information, including the name, images, description, pricing and the options to Try-on or Add to cart.

Alternative Flows	**Alternative flow 1: User is at the Shopping cart page**
	   1. The user clicks on any item image or name in the cart.
	   2. Continue with #3 of the Basic flow.
	
	Alternative flow 2: User is at the Product page
	   1. The user clicks on any item image or name in the Product page.
	   2. Continue with #3 of the Basic flow.

Pre-conditions	User goes to the homepage at **'domain-name/home'.**
Post-conditions	The user successfully views a specific product's information.

2.8 Use-case: Add product to cart

Use case Name	Add a product to cart.
Brief description	This use case describes how the Customer can add a product to the shopping cart.
Actors	Buyer

Basic Flow	1. At the homepage, the buyer sees multiple lists of items such as Best Sellers or Recently Added.
	2. The buyer clicks on the "Add to cart" button on their preferred product.
	3. The system displays the product added, including its image and quantity:
	   o If the shopping cart hasn't had the product, its quantity will be 1.
	   o If the shopping cart has already had the product, its quantity will be increased.

Alternative Flows	**Alternative flow 1: User searches or filters out for a specific item** 1. From #1 of the basic flow, the user clicks on the "Product" button to go to the Product page. 2. The user clicks on the "Search" field to start searching or "Filter" button to start filtering items by categories. 3. The system displays the products found. If none is found, the user returns to #2 of this flow to enter different keywords or categories. 4. Continue with #2 of the Basic flow. **Alternative flow 2: User views a product's information** 1. From #1 of the basic flow, the user clicks on an item image or name. 2. The system redirects the user to the individual product information page. 3. The system displays the item's information. 4. The user clicks on the "Add to cart" button. 5. Continue with #3 of the Basic flow.
Pre-conditions	User goes to the homepage at **'domain-name/home'.**
Post-conditions	The user successfully adds a new item to the cart or changes the quantity of the existing item in the cart.

2.9 Use-case: View cart details

Use case Name	View cart details

Brief description	This use case describes how the Customer can view their shopping cart.
Actors	Customer
Basic Flow	1. At the homepage, the user clicks on the "Cart" button. 2. The system displays the Shopping cart page, with information about the current products added, their quantity and prices.
Alternative Flows	**Alternative flow 1: User needs to change a product's quantity** 1. From #2 of the Basic flow, the user clicks on the "+" or "-" icon to respectively increase or decrease the product's quantity. Alternatively, the user clicks on the quantity field (number) to manually enter the quantity number.
Pre-conditions	User goes to the homepage at **'domain-name/home'.**
Post-conditions	The user successfully adds a new item to the cart or changes the quantity of the existing item in the cart.

2.10   Use-case: Checkout order

Use case Name	Place order
Brief description	This use case describes how the Customer can place an order with the current items in the Shopping cart.
Actors	Customer
Basic Flow	1. The system displays the current items in the Shopping cart, including their pricing, quantities and the total of the order. 2. The user clicks on and starts filling out the order information fields, including the delivery address, name and phone number. 3. The user clicks on the "Order" button to complete the ordering process. 4. The system displays a pop-up, notifying about the order's success. 5. The system redirects the user to the payment gateway corresponding to the payment method that they chose. *# For Momo, the user is prompted to scan a QR code to finish the checkout.* 6. Afterwards, the system redirects the user back to the Home page, with a pop-up, notifying the user of the order's success.

Alternative Flows	**Alternative flow 1: User enters invalid information.**
	1. From #3 of the Basic flow, the system displays an "error" message, reminding the user to edit the necessary fields.
	2. The user clicks on and starts filling out the invalid fields.
	3. Continue with #3 of the Basic flow.
	Alternative flow 2: Shopping cart has no products
	1. From #3 of the Basic flow, the system displays an "error" message about the empty Shopping cart.
	2. Afterwards, the user has 2 options to add products to cart:
	o Goes back to the Home page.
	o Go to the Product page.
	Alternative flow 1: User doesn't finish the checkout
	1. From #2, the system redirects the user back to the Shopping cart page, with all the information stored.
	2. The system displays a pop-up, notifying the user of the payment failure.
	3. Afterwards, the user clicks on the "Order" button to repeat the payment process (Basic flow).
Pre-conditions	User goes to the Shopping cart page at **'domain-name/cart'.**
Post-conditions	The user successfully completes the checkout for their current order.

2.11   Use-case: Use chatbot

Use case Name	Use chatbot
Brief description	This use case describes how the Customer can use the chatbot feature.
Actors	User
Basic Flow	1. User opens the Chatbot side by clicking on the 'Chat' button
	2. System displays the chatbot interface
	3. User enters a question or request into the chatbot
	4. System processes the user's message and determines intent
	5. System retrieves relevant data or generates a response
	6. System displays an answer, suggestion, product results, or image output
Alternative Flows	**Alternative flow 1: Product Information or Pricing Request**
	1. From Step #2 in the Basic Flow, the System detects the user is requesting product information or pricing
	2. System retrieves relevant product details and pricing data
	3. System displays the response to the user
	4. Continue at Step #6 in the Basic Flow
	Alternative flow 2: Outfit Suggestion Request
	1. From Step #2 in the Basic Flow, the System detects the user is requesting outfit suggestions.
	2. System generates suitable outfit recommendations.
	3. System displays suggested items and images (if applicable).
	4. Continue at Step #6 in the Basic Flow.

	Alternative flow 3: Virtual Try-On Request
	1. From Step #2 in the Basic Flow, the System detects the user is requesting a virtual try-on.
	2. System prompts the user to upload an image or select an existing one.
	3. System processes the virtual try-on request.
	4. System displays the generated try-on result image(s).
	5. Continue at Step #6 in the Basic Flow.
	
	Alternative flow 4: Unrecognized Query
	1. From Step #2 in the Basic Flow, the System detects the user is requesting a virtual try-on.
	2. System prompts the user to upload an image or select an existing one.
	3. System processes the virtual try-on request.
	4. System displays the generated try-on result image(s).
	5. Continue at Step #6 in the Basic Flow.
	
	Alternative flow 5: User Clicks on Product Detail
	1. From Step #6 in the Basic Flow, the User clicks on a product detail displayed in the chatbot response.
	2. System opens the corresponding product detail page in a new view or redirects the user to that page.
	3. Alternative flow ends here.
	4. The main use case (Use chatbot) continues as normal.
	
	Alternative flow 4: System Error / Processing Failure
	1. From any step in the Basic Flow, the System encounters an internal error and cannot process the request.
	2. System displays message: "An error occurred. Please try again later."
	3. User chooses one of the following options:
	    ● Exit the chatbot → Use case ends
	    ● Retry → Continue at Step #2 in the Basic Flow

Pre-conditions	User has opened the chatbot interface from the website.
Post-conditions	The chatbot provides responses based on user inputs, including product information, answers, suggestions, or try-on assistance. The user may view returned images or product items where applicable.

2.12   Use-case: Manage categories

Use case Name	Category Management
Brief description	This use case describes how the Administrator can manage product categories, including adding, updating, and deleting them in the system.
Actors	Staff

Basic Flow	1. Admin navigates to the "Category Management" page from the admin dashboard. 2. The system displays a list of all existing categories with options to add, edit, or delete. 3. Admin selects one of the following actions: **Add a category** 3.1 Admin clicks the "Add Category" button. 3.2 The system displays a form to enter category name, description, and optional parent category. 3.3 Admin fills the form and clicks "Save". 3.4 The system validates input data and adds the new category to the database. 3.5 System refreshes the list and displays the newly added category. **Update a category** 3.1 Admin clicks "Edit" on an existing category. 3.2 The system displays the category's details for modification. 3.3 Admin updates the information and clicks "Save changes". 3.4 System validates and updates the category record. 3.5 System displays the updated category list. **Delete a category** 3.1 Admin clicks "Delete" on a category. 3.2 System asks for confirmation. 3.3 Admin confirms deletion. 3.4 System deletes the category and removes it from the list
Alternative Flows	**Alternative flow 1: Duplicate category name** 1. From step 3.3 of "Add a category", the system detects that a category with the same name already exists. 2. The system displays an error message "Category name already exists." 3. Admin corrects the name and retries adding. **Alternative flow 2: Circular dependency in category hierarchy** 1. From step 3.3 of "Update a category", the system detects that the selected parent category would create a circular relationship. 2. The system displays an error message "Invalid parent category (circular dependency detected)." 3. Admin selects a valid parent category and retries. **Alternative flow 3: Delete category with subcategories or items** 1. From step 3.3 of "Delete a category", the system detects that the category has existing products or subcategories. 2. The system displays a warning "Cannot delete a category that contains items or child categories." 3. Admin either cancels the deletion or removes dependent data before retrying.

	Alternative flow 4: Missing required fields 1. From step 3.3 of "Add" or "Update", the system detects empty required fields (e.g., name). 2. The system displays a validation message prompting the admin to fill in all required fields.
Pre-conditions	Admin must be logged into the system and have access rights to the Category Management page.
Post-conditions	The category data is successfully created, updated, or deleted in the system. The system displays the latest category list reflecting all changes.

2.13 Use-case: Manage products

Use case Name	Manage products
Brief description	This use case describes how a Staff member or Administrator can manage the full lifecycle of products in the system — including creating, viewing, editing, deleting, and managing related categories.
Actors	**Primary**: Staff (with product management permissions) **Secondary**: Admin, System
Basic Flow	1. Staff navigates to the Product Management page. 2. The system displays the list of products (with search, filter, and pagination features). 3. Staff selects one of the available actions: 4. Create a new product → The system displays an empty form. 5. View product details → The system displays read-only details. 6. Edit a product → The system displays the form with current data. 7. Delete a product → The system prompts for confirmation. 8. Manage categories → The system allows category CRUD actions. 9. Staff fills in or modifies product information: name, description, SKU, price, stock quantity, category, status (active/inactive), image, and other attributes. 10. The system performs input validation (required fields, non-negative price, integer stock quantity, valid image format, etc.). 11. If all fields are valid, the staff clicks Save / Update / Create. 12. The system updates the database, logs the change, and shows a success message: 13. "Product saved/updated successfully." 14. The system returns to the product list page or stays on the detail page (based on UI design).
Alternative Flows	**Alternative flow 1: Invalid or incomplete input (from Step 5)** 1. The system detects invalid or missing information. 2. The system displays inline or summarized error messages near the invalid fields. 3. Staff corrects the input and continues from Step 5 of the Basic Flow. **Alternative flow 2: Deleting a product (with confirmation)** 1. When Staff clicks Delete, the system asks for confirmation (and optionally a reason).

	2. Upon confirmation, the system either performs a soft delete or permanent delete, depending on policy. 3. The system displays a confirmation message. **Alternative flow 3: System error during update (from Step 7)** 1. The system encounters a technical issue (e.g., DB failure, timeout). 2. The system displays: "An error occurred. Please try again later." 3. Staff can retry or cancel the operation. **Alternative flow 4: Product linked to existing orders (business constraint)** 1. If the product has an existing order history, the system prevents deletion. 2. The system allows only disabling the product (mark as inactive). 3. A message explains why deletion is not possible.
Pre-conditions	The user is logged in and has permission to manage products. The user is on the **Product Management** page/module.
Post-conditions	The product data (create/edit/delete) is successfully updated in the database, or an appropriate error message is displayed if the operation fails.

2.14 Use-case: Manage orders

Use case Name	Manage orders
Brief description	This use case describes how Staff or Admin can manage customer orders across their lifecycle: view, update status, fulfill, cancel, refund, and handle exceptions (payment failures, backorders, returns).
Actors	**Primary:** Staff (Order Manager) **Secondary:** Admin, Customer, System, Warehouse/Logistics, Payment Gateway
Basic Flow	1. Staff navigates to the **Order Management** page. 2. The system displays the order list with search, filters (status, date range, customer, payment method), and pagination. 3. Staff selects an order to view full details (items, quantities, prices, shipping address, payment status, order history/timeline). 4. Staff performs one of the allowed actions depending on permissions and order state: 　● **Confirm / Process Order** (reserve inventory, mark as processing). 　● **Pack / Mark Ready for Shipment** (generate picklist / packing slip). 　● **Ship Order** (enter carrier + tracking number; change status to Shipped). 　● **Cancel Order** (if allowed) and initiate refund if payment is captured. 　● **Issue Refund** or **Partial Refund**. 　● **Update Shipping Details** or **Change Delivery Method**. 　● **Create Return / RMA** and process returned items. 　● **Add an internal note** or **contact the customer**. 5. The system validates action (e.g., check inventory, verify payment status, check cancel window).

	6. If validation passes, the system updates order status, updates inventory if needed, records the change in audit logs, and triggers notifications (email/SMS) to customer and downstream systems (warehouse, shipping provider). 7. The system displays a confirmation to the staff: e.g., "Order status updated to Shipped. Tracking XYZ123." 8. The system returns to the order details or list depending on UI flow.
Alternative Flows	**Alternative flow 1: Payment not captured / payment failed** 1. If payment is pending or failed when trying to process the order: 2. The system blocks certain actions (e.g., shipment) until payment is captured. 3. The system displays payment status and recommended next steps (retry capture, contact customer). 4. Staff may choose to cancel the order or contact the customer to update payment. **Alternative flow 2: Insufficient inventory / backorder** 1. If stock is insufficient for one or more items: 2. The system shows affected items and allows partial fulfillment (ship available items) or place items on backorder. 3. The system reserves inventory where possible and notifies the customer about expected delays. **Alternative flow 3: Cancellation before fulfillment** 1. If staff/customer requests cancellation before shipping and policy allows: 2. The system cancels the order, releases reserved inventory, triggers refund (if paid), and notifies the customer. 3. If cancellation is not allowed (e.g., already shipped), the system informs staff and suggests a return process. **Alternative flow 4: Partial shipment** 1. Staff can ship part of the order: 2. System updates order line statuses, generates separate shipments and tracking numbers, and notifies the customer with partial-shipment details. **Alternative flow 5: Return / Refund after delivery** 1. Customer initiates a return or RMA: 2. Staff records the RMA, provides return instructions, and upon receiving items, inspects them and issues refund or replacement per policy. 3. The system updates inventory and financial records accordingly. **Alternative flow 6: System error (e.g., DB or external API failure)** 1. System displays error message: "An error occurred. Please try again later." 2. System logs the error for investigation; staff may retry the operation or perform fallback actions.
Pre-conditions	The user (Staff or Admin) is authenticated and authorized to manage orders.

	The user is on the **Order Management** page/module.
Post-conditions	Order state is updated (e.g., processed, shipped, cancelled, refunded) and corresponding systems (inventory, billing, shipping) are notified, or an error message is displayed and logged.

2.15 Use-case: Export report

Use case Name	Export Report
Brief description	This use case describes how the Administrator can view the business performance dashboard and optionally export a detailed report containing additional analytical information such as revenue, top-selling products, total orders, and product returns.
Actors	Administrator (Staff/Shop owner)
Basic Flow	1. Admin logs into the system and navigates to the "Dashboard" page. 2. The system displays analytical charts and key metrics (e.g., total revenue, top products, number of orders, and returns). 3. Admin reviews the information on the dashboard. 4. Admin clicks on the "Export Report" button if a detailed report is required. 5. The system asks for export parameters (e.g., date range, report type, or format such as PDF/Excel). 6. Admin selects desired options and confirms export. 7. The system gathers detailed data from the database and generates the report. 8. The system provides a downloadable report file to the admin.
Alternative Flows	**Alternative flow 1: Admin only views dashboard** 1. Admin performs steps #1–#3 of the basic flow. 2. Admin chooses not to export the report. 3. The system remains on the dashboard displaying visual analytics. **Alternative flow 2: Invalid or missing export parameters** 1. From step #6 of the basic flow, the admin leaves required fields (e.g., date range) empty. 2. The system displays a validation message "Please select all required parameters." 3. Admin goes back to step #5 to correct the input and retry.
Pre-conditions	Admin must be logged into the system and have permission to access the dashboard and report export feature.
Post-conditions	The admin successfully views the dashboard and, if requested, obtains a generated report file containing detailed analytical information.

2.16 Use-case: Manage staffs

Use case Name	Manage staff
Brief description	This use case describes how the Root Admin (shop owner) can manage staff accounts and roles, including creating custom roles with permissions, adding new

	staff accounts, assigning roles, and deleting staff accounts when necessary.
Actors	Root Admin (Shop Owner)
Basic Flow	1. Root Admin logs into the system and navigates to the "Staff Management" page. 2. The system displays a list of existing staff members 3. Root Admin selects one of the following actions: **Create a new staff account** 3.1 Admin clicks "Add Staff". 3.2 The system displays a form to enter staff information (email, password, name) and choose either a default or custom role. 3.3 Admin submits the form. 3.4 The system validates the information and creates the staff account. 3.5 System updates the staff list and displays the new account. **Delete a staff account** 3.1 Admin clicks "Delete" on a specific staff member. 3.2 System asks for confirmation. 3.3 Admin confirms deletion. 3.4 The system removes the staff account and updates the list.
Alternative Flows	**Alternative flow 1: Missing required information** 1. From step 3.3 of "Add staff", required fields are missing. 2. The system displays a validation message "Please fill all required fields." 3. Admin corrects input and retries. **Alternative flow 2: Attempt to delete self or last admin** 1. From step 3.3 of "Delete staff account", the system detects the admin is attempting to delete their own account or the only remaining admin account. 2. The system displays "Cannot delete this account." 3. Admin cancels or chooses another account.
Pre-conditions	Root Admin must be logged into the system with a root admin account.
Post-conditions	The system reflects all updates to staff accounts. New accounts are added successfully, or deleted accounts are removed from the system.

2.17   Use-case: Manage roles

Use case Name	Manage roles
Brief description	This use case describes how the Root Admin (shop owner) can manage roles, including creating custom roles
Actors	Root Admin (Shop Owner)
Basic Flow	1. Root Admin logs into the system and navigates to the "Role management" page. 2. The system displays a list of existing roles with their permissions. 3. Admin clicks "Create Role". **Create a new role** 3.1 Admin clicks "Add role".

	3.2 The system displays a form to enter the role name and select permissions (e.g., manage products, manage orders, view reports, etc.). 3.3 Admin submits the form. 3.4 The system validates the information and creates the new role. 3.5 System updates the role list and displays the new role. **Delete a role** 3.1 Admin clicks "Delete" on a specific role. 3.2 System asks for confirmation. 3.3 Admin confirms deletion. 3.4 The system removes the role and updates the list. **Update a role** 3.1 Admin clicks "Edit" on a specific role. 3.2 The system displays a form to change the role name and select permissions 3.3 Admin click "Save". 3.4 The system validates and saves the role.
Alternative Flows	**Alternative flow 1: Duplicate role name** 1. From step 3.3 of "Create a custom role" or "Update a role", the system detects if a role name already exists. 2. The system displays a message "Role name already exists." 3. Admin changes the role name and retries. **Alternative flow 2: Role linked to active staff** 1. From step 3.3 of "Delete a role", the system detects that a role is assigned to existing staff. 2. System displays "Cannot delete a role assigned to active staff."
Pre-conditions	Root Admin must be logged into the system with a root admin account.
Post-conditions	The system reflects all updates to and roles. New roles are added successfully, or deleted roles are removed from the system. All permission settings are updated accordingly.