

CSC 510 Project 1C1 - Fall 2025

Group 28 - Dhananjay Raghu, Ishaan Patel, Shuba Kalyanasundaram

New Use Cases

1. Place Guest Order

Preconditions:

- The café has a menu with items and the customer is on the ordering screen.

Main Flow:

- Customer selects an item from the menu [Select Item].
- System adds the item to the cart [Add Item to Cart].
- Customer proceeds to checkout [Checkout].
- Customer enters their name and phone number [Enter Contact Info].
- Customer confirms the order [Confirm Order].
- System creates the order, calculates the total (8.5% tax).

Subflows:

- [Select Item] Display item name, price, and stock status.
- [Add Item to Cart] Store item ID and quantity.
- [Checkout] Show cart summary: items, subtotal, 8.5% tax, total.
- [Enter Contact Info] Collect name (required) and phone number (optional).
- [Confirm Order] Validate name, create order, show "Order #123 ready in 5 min."
- [Generate Pickup Code] Auto-generate a 4-digit code.

Alternative Flows:

- [Out of Stock] Disable the "Add to Cart" button. Show a "Sold Out" badge.
- [No Name Entered] Show "Name is required for pickup" and block confirmation.

2. View Active Orders

Preconditions: The system is running and there are active orders.

Main Flow:

- Staff logs in to the system [Login].
- System displays a list of active orders [View Active Orders] with columns: Order #, Item, Customer Name, Status.
- Each order row shows the current status (Open or Ready).
- Staff can view details for each order by clicking on the order number.

Subflows:

- [View Active Orders] Display all active orders in a table format, sorted by order date.
- [Login] Prompt staff to enter credentials and authenticate.

Alternative Flows:

- If an item is out of stock, the system prevents checkout until the item is removed.
- If the system cannot record the order due to an error, it prompts the customer to retry.

3. Fulfill Order

Preconditions: The order exists in the system and is not already marked as fulfilled.

Main Flow:

- Staff logs into the fulfillment screen [Log In].
- System displays available orders for pickup [Display Available Orders].
- Staff selects an order to fulfill [Select Order].
- System marks the order as ready for pickup [Mark Ready] with timestamp and staff ID.
- System updates the public pickup board with the fulfilled order information.

Subflows:

- [Log In] Authenticate staff credentials, display available orders.

Alternative Flows:

- [Order Not Found] Display "Order not found" message and allow staff to search or return to main menu.
- [Order Already Fulfilled] Display "Order already marked as ready" message and prevent further fulfillment attempts.

4. Add Inventory

Preconditions: The customer is on the ordering screen and the item exists in the menu.

Main Flow:

- Customer selects an existing item [Select Existing Item].
- System displays current stock quantity for the selected item [Display Stock Quantity].
- Customer enters new quantity to add to the existing stock [Enter New Quantity].
- System updates the inventory with the new quantity and shows a confirmation message [Update Inventory].

Subflows:

- [Select Existing Item] Display item name, price, and current stock status.
- [Display Stock Quantity] Show the current quantity available for the selected item.
- [Enter New Quantity] Allow customer to enter a new quantity to add to the existing stock.
- [Update Inventory] Store the updated inventory quantity and show a confirmation message.

Alternative Flows:

- [Invalid Input] If the entered quantity is invalid (e.g., not a positive integer), show an error message and allow re-entry.

5. Create Menu Item

Preconditions:

- The user is logged in as an admin.

Main Flow:

- Admin enters menu item name [Enter Name].
- Admin sets menu item price [Set Price].
- Admin links menu item to existing inventory item [Link to Inventory].
- System validates input and creates new menu item.
- System displays confirmation message "Menu item created successfully".

Subflows:

- [Enter Name] Validate name length (min 1, max 50 characters).
- [Set Price] Validate price value (positive number).
- [Link to Inventory] Display existing inventory items for selection.

Alternative Flows:

- [Invalid Input] Show error message "Invalid input. Please try again." and reset form.
- [Item Already Exists] Show error message "Menu item already exists. Please choose a different name."

6. Public Pickup Board

Preconditions:

- The customer has placed an order and the staff has marked it as ready for pickup.

Main Flow:

- System displays the public pickup board with the order number and first name.
- Order remains on the board for 10 minutes.
- After 10 minutes, system automatically removes the order from the board.

Subflows:

- [Public Pickup Board] Display order # and first name on screen. No additional information (e.g., no item names or quantities).

Alternative Flows:

- If the board fails to update, staff can verbally call out the order.
- If multiple orders are ready at the same time, each is listed separately with its own identifiers.

7. Checkout

Preconditions: The customer has added at least one item to their cart.

Main Flow:

- System displays the checkout summary screen [Checkout Summary].
- System shows a list of items in the cart, including name, price, and quantity.

- System calculates the subtotal by multiplying the price of each item by its quantity.
- System adds 8.5% tax to the subtotal.
- System displays the total amount due.

Subflows:

- [Checkout Summary] Display the checkout summary screen with a header showing the store name and a footer showing the total amount due.

Alternative Flows:

- [Empty Cart] Show "Your cart is empty" message and block confirmation.

8. Low Stock Alert

Preconditions: The customer has added at least one item to their cart.

Main Flow:

- System displays the checkout summary screen [Checkout Summary].
- System shows a list of items in the cart, including name, price, and quantity.
- System calculates the subtotal by multiplying the price of each item by its quantity.
- System adds 8.5% tax to the subtotal.
- System displays the total amount due.

Subflows:

- [Checkout Summary] Display the checkout summary screen with a header showing the store name and a footer showing the total amount due.

Alternative Flows:

- [Empty Cart] Show "Your cart is empty" message and block confirmation.

9. Order History

Preconditions: The customer has added at least one item to their cart.

Main Flow:

- System displays the checkout summary screen [Checkout Summary].
- System shows a list of items in the cart, including name, price, and quantity.
- System calculates the subtotal by multiplying the price of each item by its quantity.
- System adds 8.5% tax to the subtotal.
- System displays the total amount due.

Subflows:

- [Checkout Summary] Display the checkout summary screen with a header showing the store name and a footer showing the total amount due.

Alternative Flows:

- [Empty Cart] Show "Your cart is empty" message and block confirmation.

10. View Staff

Preconditions: The admin is logged in and has permission to view staff.

Main Flow:

- Admin navigates to the staff view page [Navigate to Staff View].
- System displays a list of staff members with columns for Name, Role, and Last Login [Display Staff List].
- No edit or delete buttons are shown for each staff member [No Edit/Delete].

Subflows:

- [Staff List] Display list of staff members with the specified columns.

Alternative Flows:

- [Empty List] Show message "No staff members found" if no staff members exist in the system.

Reflection

What we cut from 1b1 and why:

1. **User accounts and profiles** were excluded, so the system only supports guest checkout with name and phone number. This decision eliminates the complexity of login flows, password management, and data privacy requirements.
2. **Configurable tax rates** were also cut, instead a fixed 8.5% rate was applied. This simplified both the database design and the administrator interface.
3. **Real-time notifications** were replaced with a public pickup board, avoiding the need for websockets or push infrastructure.
4. **Discounts and promotions** were not included, reducing both implementation complexity and potential confusion at checkout.
5. **Menu item modifiers**, such as custom toppings or extra shots, were dropped. This kept the ordering flow clean and made inventory management straightforward.

These cuts align with the stakeholder bias analysis from 1a1. Customers expressed a preference for faster service over heavy customization. Staff valued reduced workload and fewer exceptions in the ordering process. Developers prioritized building core business requirements over additional features. While administrators and management sacrificed configurability and promotional options, the essential cycle of ordering, fulfillment, and pickup was preserved.

Stakeholder disappointments:

The MVP inevitably creates disappointments for some groups.

1. Customers cannot customize their drinks or save profiles, which limits satisfaction for frequent visitors or those with dietary needs.
2. Staff have fewer dashboard capabilities and cannot filter or prioritize orders beyond the active queue.
3. Administrators lose the ability to manage tax settings, staff roles, or customer accounts, which restricts their control.
4. Managers cannot implement promotions, upselling strategies, or loyalty programs, which reduces opportunities to optimize revenue.

These issues mirror the trade-offs highlighted in 1a1. Customers favor simplicity and speed, but administrators and managers push for profit-driven features. Our MVP deliberately prioritizes simplicity and operational stability over configurability and expansion.

Appeasement changes:

To balance these trade-offs, several lightweight features were added.

1. **Added “Order History by Phone”:** Customers are given a limited form of continuity through order history based on phone number.
2. **Added “Low Stock Alert (red highlight)”:** Staff benefit from low stock alerts, which reduce fulfillment issues and prevent delays.

3. **Added “Read-Only Staff View”:** Administrators gain some visibility through a read-only staff list, even though they cannot directly manage users.
4. **Hardcoded tax = 8.5%:** Finally, the decision to fix the tax at 8.5 percent, while not configurable, matches local rates and provides predictability for both customers and administrators.

These additions help reduce stakeholder frustration without undermining the principle of a minimal viable product.

Prompt History

PROJECT 1C1 — PROMPT HISTORY

Generated on: 2025-09-13 18:01:45

Model: llama3

Method: Pure LLM (RAG disabled for MVP condensation)

Total prompts logged: 10

PROMPT #1

Timestamp: 2025-09-13T18:00:14.803994

Mode: PURE_LLM

PROMPT:

You are a senior product manager for a café ordering system. Your goal is to define a MINIMAL VIABLE PRODUCT (MVP) — the smallest set of features that validates the core loop: Customer places order → Staff fulfills → Customer picks up.

You write use cases in the EXACT format used in Proj1b1 (traffic light style):

Preconditions: [list]

Main Flow

1. [Step 1]
2. [Step 2]
3. [Step 3]

Subflows

- [Module Name] [Description, may include nested modules like [SubModule]]
- [Another Module] ...

Alternative Flows

- [Edge Case Name] [Description]

Example (from Proj1b1, simplified for MVP):

Preconditions: The item exists in the menu and is in stock.

Customer is on the ordering screen.

Main Flow

1. Customer selects item [Select Item].
2. System adds item to cart [Add Item to Cart].
3. Customer proceeds to checkout [Checkout].
4. Customer enters name and phone [Enter Contact Info].
5. Customer confirms order [Confirm Order].
6. System creates order and shows pickup code [Generate Pickup Code].

Subflows

- [Select Item] Display item name, price, and stock status. If out of stock, disable selection [Out of Stock].
- [Add Item to Cart] Store item ID and quantity. No modifiers (e.g., no “extra shot”).
- [Checkout] Show cart summary: items, subtotal, tax (hardcoded 8.5%), total. No tips, no discounts.
- [Enter Contact Info] Collect name (required) and phone (optional). No account creation.
- [Confirm Order] Validate name → create order → show “Order #123 ready in 5 min.”
- [Generate Pickup Code] Auto-generate 4-digit code. Display on screen and print receipt.

Alternative Flows

- [Out of Stock] Disable “Add to Cart” button. Show “Sold Out” badge.
- [No Name Entered] Show “Name is required for pickup” and block confirmation.
- [System Error] Show “Order failed — please try again or contact staff.”

RULES FOR MVP (CUT FROM 1B1):

- ~~×~~ NO user accounts (cut “Create Customer User”)
- ~~×~~ NO configurable tax (cut “Set Tax Rate” — hardcode 8.5%)
- ~~×~~ NO real-time notifications (cut “Order Pickup Notification” — use public board)
- ~~×~~ NO discounts or promo codes (cut “Apply Discount”)
- ~~×~~ NO item modifiers (cut “Add Customization” — only base items)
- ~~×~~ NO multi-location or complex inventory (cut “Set Inventory Location” — just quantity)

KEEP:

- Guest checkout (name + phone)
- Staff fulfillment (mark ready)
- Public pickup board (order # + first name)
- Basic inventory (add stock, low stock alert)

- Admin create item (name, price, link to inventory)

OUTPUT ONLY THE USE CASE. START WITH "Preconditions:". NO TITLES. NO DISCLAIMERS.

Now, generate 1 USE CASE for the following MVP requirement:

MVP: Place Guest Order — simplify 'Place Order' from 1b1. Remove user accounts, discounts, tips, modifiers. Just item + name + phone. Tax = 8.5% hardcoded.

RESPONSE:

Preconditions:

- The café has a menu with items.
- The customer is on the ordering screen.

Main Flow

1. Customer selects an item from the menu [Select Item].
2. System adds the item to the cart [Add Item to Cart].
3. Customer proceeds to checkout [Checkout].
4. Customer enters their name and phone number [Enter Contact Info].
5. Customer confirms the order [Confirm Order].
6. System creates the order, calculates the total (8.5% tax), and shows a pickup code [Generate Pickup Code].

Subflows

- [Select Item] Display item name, price, and stock status.
- [Add Item to Cart] Store item ID and quantity.
- [Checkout] Show cart summary: items, subtotal, 8.5% tax, total.
- [Enter Contact Info] Collect name (required) and phone number (optional).
- [Confirm Order] Validate name → create order → show “Order #123 ready in 5 min.”
- [Generate Pickup Code] Auto-generate a 4-digit code. Display on screen and print receipt.

Alternative Flows

- [Out of Stock] Disable the "Add to Cart" button. Show a "Sold Out" badge.
- [No Name Entered] Show "Name is required for pickup" and block confirmation.
- [System Error] Show "Order failed — please try again or contact staff."

PROMPT #2

Timestamp: 2025-09-13T18:00:22.564908

Mode: PURE_LLM

PROMPT:

You are a senior product manager for a café ordering system. Your goal is to define a MINIMAL VIABLE PRODUCT (MVP) — the smallest set of features that validates the core loop: Customer places order → Staff fulfills → Customer picks up.

You write use cases in the EXACT format used in Proj1b1 (traffic light style):

Preconditions: [list]

Main Flow

1. [Step 1]
2. [Step 2]
3. [Step 3]

Subflows

- [Module Name] [Description, may include nested modules like [SubModule]]
- [Another Module] ...

Alternative Flows

- [Edge Case Name] [Description]

Example (from Proj1b1, simplified for MVP):

Preconditions: The item exists in the menu and is in stock.

Customer is on the ordering screen.

Main Flow

1. Customer selects item [Select Item].
2. System adds item to cart [Add Item to Cart].
3. Customer proceeds to checkout [Checkout].
4. Customer enters name and phone [Enter Contact Info].
5. Customer confirms order [Confirm Order].
6. System creates order and shows pickup code [Generate Pickup Code].

Subflows

- [Select Item] Display item name, price, and stock status. If out of stock, disable selection [Out of Stock].

- [Add Item to Cart] Store item ID and quantity. No modifiers (e.g., no “extra shot”).
- [Checkout] Show cart summary: items, subtotal, tax (hardcoded 8.5%), total. No tips, no discounts.
- [Enter Contact Info] Collect name (required) and phone (optional). No account creation.
- [Confirm Order] Validate name → create order → show “Order #123 ready in 5 min.”
- [Generate Pickup Code] Auto-generate 4-digit code. Display on screen and print receipt.

Alternative Flows

- [Out of Stock] Disable “Add to Cart” button. Show “Sold Out” badge.
- [No Name Entered] Show “Name is required for pickup” and block confirmation.
- [System Error] Show “Order failed — please try again or contact staff.”

RULES FOR MVP (CUT FROM 1B1):

- ~~✗~~ NO user accounts (cut “Create Customer User”)
- ~~✗~~ NO configurable tax (cut “Set Tax Rate” — hardcode 8.5%)
- ~~✗~~ NO real-time notifications (cut “Order Pickup Notification” — use public board)
- ~~✗~~ NO discounts or promo codes (cut “Apply Discount”)
- ~~✗~~ NO item modifiers (cut “Add Customization” — only base items)
- ~~✗~~ NO multi-location or complex inventory (cut “Set Inventory Location” — just quantity)

KEEP:

- Guest checkout (name + phone)
- Staff fulfillment (mark ready)
- Public pickup board (order # + first name)
- Basic inventory (add stock, low stock alert)
- Admin create item (name, price, link to inventory)

OUTPUT ONLY THE USE CASE. START WITH "Preconditions:". NO TITLES. NO DISCLAIMERS.

Now, generate 1 USE CASE for the following MVP requirement:

MVP: View Active Orders — simplify 'View Orders' from 1b1. Remove filters, search, status history. Just list: Order #, Item, Customer Name, Status (Open/Ready).

RESPONSE:

Preconditions: The system is running and there are active orders.

Main Flow

1. Staff logs in to the system [Login].
2. System displays a list of active orders [View Active Orders] with columns: Order #, Item, Customer Name, Status.
3. Each order row shows the current status (Open or Ready).
4. Staff can view details for each order by clicking on the order number.

Subflows

- [View Active Orders] Display all active orders in a table format, sorted by order date.
- [Login] Prompt staff to enter credentials and authenticate.

PROMPT #3

Timestamp: 2025-09-13T18:00:32.406188

Mode: PURE_LLM

PROMPT:

You are a senior product manager for a café ordering system. Your goal is to define a MINIMAL VIABLE PRODUCT (MVP) — the smallest set of features that validates the core loop: Customer places order → Staff fulfills → Customer picks up.

You write use cases in the EXACT format used in Proj1b1 (traffic light style):

Preconditions: [list]

Main Flow

1. [Step 1]
2. [Step 2]
3. [Step 3]

Subflows

- [Module Name] [Description, may include nested modules like [SubModule]]
- [Another Module] ...

Alternative Flows

- [Edge Case Name] [Description]

Example (from Proj1b1, simplified for MVP):

Preconditions: The item exists in the menu and is in stock.
Customer is on the ordering screen.

Main Flow

1. Customer selects item [Select Item].
2. System adds item to cart [Add Item to Cart].
3. Customer proceeds to checkout [Checkout].
4. Customer enters name and phone [Enter Contact Info].
5. Customer confirms order [Confirm Order].
6. System creates order and shows pickup code [Generate Pickup Code].

Subflows

- [Select Item] Display item name, price, and stock status. If out of stock, disable selection [Out of Stock].
- [Add Item to Cart] Store item ID and quantity. No modifiers (e.g., no “extra shot”).
- [Checkout] Show cart summary: items, subtotal, tax (hardcoded 8.5%), total. No tips, no discounts.
- [Enter Contact Info] Collect name (required) and phone (optional). No account creation.
- [Confirm Order] Validate name → create order → show “Order #123 ready in 5 min.”
- [Generate Pickup Code] Auto-generate 4-digit code. Display on screen and print receipt.

Alternative Flows

- [Out of Stock] Disable “Add to Cart” button. Show “Sold Out” badge.
- [No Name Entered] Show “Name is required for pickup” and block confirmation.
- [System Error] Show “Order failed — please try again or contact staff.”

RULES FOR MVP (CUT FROM 1B1):

- ~~×~~ NO user accounts (cut “Create Customer User”)
- ~~×~~ NO configurable tax (cut “Set Tax Rate” — hardcode 8.5%)
- ~~×~~ NO real-time notifications (cut “Order Pickup Notification” — use public board)
- ~~×~~ NO discounts or promo codes (cut “Apply Discount”)
- ~~×~~ NO item modifiers (cut “Add Customization” — only base items)
- ~~×~~ NO multi-location or complex inventory (cut “Set Inventory Location” — just quantity)

KEEP:

- Guest checkout (name + phone)
- Staff fulfillment (mark ready)
- Public pickup board (order # + first name)

- Basic inventory (add stock, low stock alert)
- Admin create item (name, price, link to inventory)

OUTPUT ONLY THE USE CASE. START WITH "Preconditions:". NO TITLES. NO DISCLAIMERS.

Now, generate 1 USE CASE for the following MVP requirement:

MVP: Fulfill Order — simplify 'Fulfill Order' from 1b1. One click to mark ready. Log fulfilledBy, fulfilledAt. No undo, no versioning.

RESPONSE:

Preconditions: The order exists in the system and is not already marked as fulfilled.

Main Flow

1. Staff logs into the fulfillment screen [Log In].
2. System displays available orders for pickup [Display Available Orders].
3. Staff selects an order to fulfill [Select Order].
4. System marks the order as ready for pickup [Mark Ready] with timestamp and staff ID.
5. System updates the public pickup board with the fulfilled order information.

Subflows

- [Log In] Authenticate staff credentials, display available orders.

Alternative Flows

- [Order Not Found] Display "Order not found" message and allow staff to search or return to main menu.
- [Order Already Fulfilled] Display "Order already marked as ready" message and prevent further fulfillment attempts.

PROMPT #4

Timestamp: 2025-09-13T18:00:44.967742

Mode: PURE_LLM

PROMPT:

You are a senior product manager for a café ordering system. Your goal is to define a MINIMAL VIABLE PRODUCT (MVP) — the smallest set of features that validates the core loop: Customer places order → Staff fulfills → Customer picks up.

You write use cases in the EXACT format used in Proj1b1 (traffic light style):

Preconditions: [list]

Main Flow

1. [Step 1]
2. [Step 2]
3. [Step 3]

Subflows

- [Module Name] [Description, may include nested modules like [SubModule]]
- [Another Module] ...

Alternative Flows

- [Edge Case Name] [Description]

Example (from Proj1b1, simplified for MVP):

Preconditions: The item exists in the menu and is in stock.

Customer is on the ordering screen.

Main Flow

1. Customer selects item [Select Item].
2. System adds item to cart [Add Item to Cart].
3. Customer proceeds to checkout [Checkout].
4. Customer enters name and phone [Enter Contact Info].
5. Customer confirms order [Confirm Order].
6. System creates order and shows pickup code [Generate Pickup Code].

Subflows

- [Select Item] Display item name, price, and stock status. If out of stock, disable selection [Out of Stock].
- [Add Item to Cart] Store item ID and quantity. No modifiers (e.g., no “extra shot”).
- [Checkout] Show cart summary: items, subtotal, tax (hardcoded 8.5%), total. No tips, no discounts.
- [Enter Contact Info] Collect name (required) and phone (optional). No account creation.
- [Confirm Order] Validate name → create order → show “Order #123 ready in 5 min.”

- [Generate Pickup Code] Auto-generate 4-digit code. Display on screen and print receipt.

Alternative Flows

- [Out of Stock] Disable “Add to Cart” button. Show “Sold Out” badge.
- [No Name Entered] Show “Name is required for pickup” and block confirmation.
- [System Error] Show “Order failed — please try again or contact staff.”

RULES FOR MVP (CUT FROM 1B1):

- ~~✗~~ NO user accounts (cut “Create Customer User”)
- ~~✗~~ NO configurable tax (cut “Set Tax Rate” — hardcode 8.5%)
- ~~✗~~ NO real-time notifications (cut “Order Pickup Notification” — use public board)
- ~~✗~~ NO discounts or promo codes (cut “Apply Discount”)
- ~~✗~~ NO item modifiers (cut “Add Customization” — only base items)
- ~~✗~~ NO multi-location or complex inventory (cut “Set Inventory Location” — just quantity)

KEEP:

- Guest checkout (name + phone)
- Staff fulfillment (mark ready)
- Public pickup board (order # + first name)
- Basic inventory (add stock, low stock alert)
- Admin create item (name, price, link to inventory)

OUTPUT ONLY THE USE CASE. START WITH "Preconditions:". NO TITLES. NO DISCLAIMERS.

Now, generate 1 USE CASE for the following MVP requirement:

MVP: Add Inventory — simplify 'Add Inventory Item' from 1b1. Just add quantity to existing item. No locations, no expiration, no batch IDs.

RESPONSE:

Preconditions: The customer is on the ordering screen and the item exists in the menu.

Main Flow

1. Customer selects an existing item [Select Existing Item].
2. System displays current stock quantity for the selected item [Display Stock Quantity].
3. Customer enters new quantity to add to the existing stock [Enter New Quantity].

4. System updates the inventory with the new quantity and shows a confirmation message [Update Inventory].

Subflows

- [Select Existing Item] Display item name, price, and current stock status.
- [Display Stock Quantity] Show the current quantity available for the selected item.
- [Enter New Quantity] Allow customer to enter a new quantity to add to the existing stock.
- [Update Inventory] Store the updated inventory quantity and show a confirmation message.

Alternative Flows

- [Invalid Input] If the entered quantity is invalid (e.g., not a positive integer), show an error message and allow re-entry.

PROMPT #5

Timestamp: 2025-09-13T18:00:56.354971

Mode: PURE_LLM

PROMPT:

You are a senior product manager for a café ordering system. Your goal is to define a MINIMAL VIABLE PRODUCT (MVP) — the smallest set of features that validates the core loop: Customer places order → Staff fulfills → Customer picks up.

You write use cases in the EXACT format used in Proj1b1 (traffic light style):

Preconditions: [list]

Main Flow

1. [Step 1]
2. [Step 2]
3. [Step 3]

Subflows

- [Module Name] [Description, may include nested modules like [SubModule]]
- [Another Module] ...

Alternative Flows

- [Edge Case Name] [Description]

Example (from Proj1b1, simplified for MVP):

Preconditions: The item exists in the menu and is in stock.

Customer is on the ordering screen.

Main Flow

1. Customer selects item [Select Item].
2. System adds item to cart [Add Item to Cart].
3. Customer proceeds to checkout [Checkout].
4. Customer enters name and phone [Enter Contact Info].
5. Customer confirms order [Confirm Order].
6. System creates order and shows pickup code [Generate Pickup Code].

Subflows

- [Select Item] Display item name, price, and stock status. If out of stock, disable selection [Out of Stock].
- [Add Item to Cart] Store item ID and quantity. No modifiers (e.g., no “extra shot”).
- [Checkout] Show cart summary: items, subtotal, tax (hardcoded 8.5%), total. No tips, no discounts.
- [Enter Contact Info] Collect name (required) and phone (optional). No account creation.
- [Confirm Order] Validate name → create order → show “Order #123 ready in 5 min.”
- [Generate Pickup Code] Auto-generate 4-digit code. Display on screen and print receipt.

Alternative Flows

- [Out of Stock] Disable “Add to Cart” button. Show “Sold Out” badge.
- [No Name Entered] Show “Name is required for pickup” and block confirmation.
- [System Error] Show “Order failed — please try again or contact staff.”

RULES FOR MVP (CUT FROM 1B1):

- ~~✗~~ NO user accounts (cut “Create Customer User”)
- ~~✗~~ NO configurable tax (cut “Set Tax Rate” — hardcode 8.5%)
- ~~✗~~ NO real-time notifications (cut “Order Pickup Notification” — use public board)
- ~~✗~~ NO discounts or promo codes (cut “Apply Discount”)
- ~~✗~~ NO item modifiers (cut “Add Customization” — only base items)
- ~~✗~~ NO multi-location or complex inventory (cut “Set Inventory Location” — just quantity)

KEEP:

- Guest checkout (name + phone)
- Staff fulfillment (mark ready)

- Public pickup board (order # + first name)
- Basic inventory (add stock, low stock alert)
- Admin create item (name, price, link to inventory)

OUTPUT ONLY THE USE CASE. START WITH "Preconditions:". NO TITLES. NO DISCLAIMERS.

Now, generate 1 USE CASE for the following MVP requirement:

MVP: Create Menu Item — simplify 'Create Menu Item' from 1b1. Just name, price, link to inventory item. No images, categories, descriptions, or modifiers.

RESPONSE:

Preconditions:

The user is logged in as an admin.

Main Flow

1. Admin enters menu item name [Enter Name].
2. Admin sets menu item price [Set Price].
3. Admin links menu item to existing inventory item [Link to Inventory].
4. System validates input and creates new menu item.
5. System displays confirmation message "Menu item created successfully".

Subflows:

- [Enter Name] Validate name length (min 1, max 50 characters).
- [Set Price] Validate price value (positive number).
- [Link to Inventory] Display existing inventory items for selection.

Alternative Flows:

- [Invalid Input] Show error message "Invalid input. Please try again." and reset form.
- [Item Already Exists] Show error message "Menu item already exists. Please choose a different name."

PROMPT #6

Timestamp: 2025-09-13T18:01:03.561209

Mode: PURE_LLM

PROMPT:

You are a senior product manager for a café ordering system. Your goal is to define a MINIMAL VIABLE PRODUCT (MVP) — the smallest set of features that validates the core loop: Customer places order → Staff fulfills → Customer picks up.

You write use cases in the EXACT format used in Proj1b1 (traffic light style):

Preconditions: [list]

Main Flow

1. [Step 1]
2. [Step 2]
3. [Step 3]

Subflows

- [Module Name] [Description, may include nested modules like [SubModule]]
- [Another Module] ...

Alternative Flows

- [Edge Case Name] [Description]

Example (from Proj1b1, simplified for MVP):

Preconditions: The item exists in the menu and is in stock.

Customer is on the ordering screen.

Main Flow

1. Customer selects item [Select Item].
2. System adds item to cart [Add Item to Cart].
3. Customer proceeds to checkout [Checkout].
4. Customer enters name and phone [Enter Contact Info].
5. Customer confirms order [Confirm Order].
6. System creates order and shows pickup code [Generate Pickup Code].

Subflows

- [Select Item] Display item name, price, and stock status. If out of stock, disable selection [Out of Stock].
- [Add Item to Cart] Store item ID and quantity. No modifiers (e.g., no “extra shot”).
- [Checkout] Show cart summary: items, subtotal, tax (hardcoded 8.5%), total. No tips, no discounts.
- [Enter Contact Info] Collect name (required) and phone (optional). No account creation.

- [Confirm Order] Validate name → create order → show “Order #123 ready in 5 min.”
- [Generate Pickup Code] Auto-generate 4-digit code. Display on screen and print receipt.

Alternative Flows

- [Out of Stock] Disable “Add to Cart” button. Show “Sold Out” badge.
- [No Name Entered] Show “Name is required for pickup” and block confirmation.
- [System Error] Show “Order failed — please try again or contact staff.”

RULES FOR MVP (CUT FROM 1B1):

- ~~✗~~ NO user accounts (cut “Create Customer User”)
- ~~✗~~ NO configurable tax (cut “Set Tax Rate” — hardcode 8.5%)
- ~~✗~~ NO real-time notifications (cut “Order Pickup Notification” — use public board)
- ~~✗~~ NO discounts or promo codes (cut “Apply Discount”)
- ~~✗~~ NO item modifiers (cut “Add Customization” — only base items)
- ~~✗~~ NO multi-location or complex inventory (cut “Set Inventory Location” — just quantity)

KEEP:

- Guest checkout (name + phone)
- Staff fulfillment (mark ready)
- Public pickup board (order # + first name)
- Basic inventory (add stock, low stock alert)
- Admin create item (name, price, link to inventory)

OUTPUT ONLY THE USE CASE. START WITH "Preconditions:". NO TITLES. NO DISCLAIMERS.

Now, generate 1 USE CASE for the following MVP requirement:

MVP: Public Pickup Board — replace 'Order Pickup Notification' from 1b1. Show only order # + first name on public screen. Auto-remove after 10 min. No websockets, no accessibility live regions.

RESPONSE:

Preconditions: The customer has placed an order and the staff has marked it as ready for pickup.

Main Flow

1. System displays the public pickup board with the order number and first name.

2. Order remains on the board for 10 minutes.
3. After 10 minutes, system automatically removes the order from the board.

Subflows

- [Public Pickup Board] Display order # and first name on screen. No additional information (e.g., no item names or quantities).

PROMPT #7

Timestamp: 2025-09-13T18:01:14.215282

Mode: PURE_LLM

PROMPT:

You are a senior product manager for a café ordering system. Your goal is to define a MINIMAL VIABLE PRODUCT (MVP) — the smallest set of features that validates the core loop: Customer places order → Staff fulfills → Customer picks up.

You write use cases in the EXACT format used in Proj1b1 (traffic light style):

Preconditions: [list]

Main Flow

1. [Step 1]
2. [Step 2]
3. [Step 3]

Subflows

- [Module Name] [Description, may include nested modules like [SubModule]]
- [Another Module] ...

Alternative Flows

- [Edge Case Name] [Description]

Example (from Proj1b1, simplified for MVP):

Preconditions: The item exists in the menu and is in stock.

Customer is on the ordering screen.

Main Flow

1. Customer selects item [Select Item].

2. System adds item to cart [Add Item to Cart].
3. Customer proceeds to checkout [Checkout].
4. Customer enters name and phone [Enter Contact Info].
5. Customer confirms order [Confirm Order].
6. System creates order and shows pickup code [Generate Pickup Code].

Subflows

- [Select Item] Display item name, price, and stock status. If out of stock, disable selection [Out of Stock].
- [Add Item to Cart] Store item ID and quantity. No modifiers (e.g., no “extra shot”).
- [Checkout] Show cart summary: items, subtotal, tax (hardcoded 8.5%), total. No tips, no discounts.
- [Enter Contact Info] Collect name (required) and phone (optional). No account creation.
- [Confirm Order] Validate name → create order → show “Order #123 ready in 5 min.”
- [Generate Pickup Code] Auto-generate 4-digit code. Display on screen and print receipt.

Alternative Flows

- [Out of Stock] Disable “Add to Cart” button. Show “Sold Out” badge.
- [No Name Entered] Show “Name is required for pickup” and block confirmation.
- [System Error] Show “Order failed — please try again or contact staff.”

RULES FOR MVP (CUT FROM 1B1):

- ~~✗~~ NO user accounts (cut “Create Customer User”)
- ~~✗~~ NO configurable tax (cut “Set Tax Rate” — hardcode 8.5%)
- ~~✗~~ NO real-time notifications (cut “Order Pickup Notification” — use public board)
- ~~✗~~ NO discounts or promo codes (cut “Apply Discount”)
- ~~✗~~ NO item modifiers (cut “Add Customization” — only base items)
- ~~✗~~ NO multi-location or complex inventory (cut “Set Inventory Location” — just quantity)

KEEP:

- Guest checkout (name + phone)
- Staff fulfillment (mark ready)
- Public pickup board (order # + first name)
- Basic inventory (add stock, low stock alert)
- Admin create item (name, price, link to inventory)

OUTPUT ONLY THE USE CASE. START WITH "Preconditions:". NO TITLES. NO DISCLAIMERS.

Now, generate 1 USE CASE for the following MVP requirement:

MVP: Checkout Summary — simplify 'Checkout' from 1b1. Show items, subtotal, tax (8.5%), total. No payment methods — assume paid in store. No tips, no discounts.

RESPONSE:

Preconditions: The customer has added at least one item to their cart.

Main Flow

1. System displays the checkout summary screen [Checkout Summary].
2. System shows a list of items in the cart, including name, price, and quantity.
3. System calculates the subtotal by multiplying the price of each item by its quantity.
4. System adds 8.5% tax to the subtotal.
5. System displays the total amount due.

Subflows

- [Checkout Summary] Display the checkout summary screen with a header showing the store name and a footer showing the total amount due.

Alternative Flows

- [Empty Cart] Show "Your cart is empty" message and block confirmation.

PROMPT #8

Timestamp: 2025-09-13T18:01:23.121664

Mode: PURE_LLM

PROMPT:

You are a senior product manager for a café ordering system. Your goal is to define a MINIMAL VIABLE PRODUCT (MVP) — the smallest set of features that validates the core loop: Customer places order → Staff fulfills → Customer picks up.

You write use cases in the EXACT format used in Proj1b1 (traffic light style):

Preconditions: [list]

Main Flow

1. [Step 1]
2. [Step 2]
3. [Step 3]

Subflows

- [Module Name] [Description, may include nested modules like [SubModule]]
- [Another Module] ...

Alternative Flows

- [Edge Case Name] [Description]

Example (from Proj1b1, simplified for MVP):

Preconditions: The item exists in the menu and is in stock.

Customer is on the ordering screen.

Main Flow

1. Customer selects item [Select Item].
2. System adds item to cart [Add Item to Cart].
3. Customer proceeds to checkout [Checkout].
4. Customer enters name and phone [Enter Contact Info].
5. Customer confirms order [Confirm Order].
6. System creates order and shows pickup code [Generate Pickup Code].

Subflows

- [Select Item] Display item name, price, and stock status. If out of stock, disable selection [Out of Stock].
- [Add Item to Cart] Store item ID and quantity. No modifiers (e.g., no “extra shot”).
- [Checkout] Show cart summary: items, subtotal, tax (hardcoded 8.5%), total. No tips, no discounts.
- [Enter Contact Info] Collect name (required) and phone (optional). No account creation.
- [Confirm Order] Validate name → create order → show “Order #123 ready in 5 min.”
- [Generate Pickup Code] Auto-generate 4-digit code. Display on screen and print receipt.

Alternative Flows

- [Out of Stock] Disable “Add to Cart” button. Show “Sold Out” badge.
- [No Name Entered] Show “Name is required for pickup” and block confirmation.
- [System Error] Show “Order failed — please try again or contact staff.”

RULES FOR MVP (CUT FROM 1B1):

- ~~✗~~ NO user accounts (cut “Create Customer User”)
- ~~✗~~ NO configurable tax (cut “Set Tax Rate” — hardcode 8.5%)
- ~~✗~~ NO real-time notifications (cut “Order Pickup Notification” — use public board)
- ~~✗~~ NO discounts or promo codes (cut “Apply Discount”)
- ~~✗~~ NO item modifiers (cut “Add Customization” — only base items)
- ~~✗~~ NO multi-location or complex inventory (cut “Set Inventory Location” — just quantity)

KEEP:

- Guest checkout (name + phone)
- Staff fulfillment (mark ready)
- Public pickup board (order # + first name)
- Basic inventory (add stock, low stock alert)
- Admin create item (name, price, link to inventory)

OUTPUT ONLY THE USE CASE. START WITH "Preconditions:". NO TITLES. NO DISCLAIMERS.

Now, generate 1 USE CASE for the following MVP requirement:

MVP: Low Stock Alert — simplify 'Check Inventory' from 1b1. Highlight items with ≤ 5 units in red. No emails, no popups, no auto-ordering.

RESPONSE:

Preconditions: The item exists in the menu and has a quantity.

Main Flow

1. Customer views menu [View Menu].
2. System displays item name, price, and stock status.
3. If stock quantity is ≤ 5 units, system highlights item in red [Low Stock Alert].

Subflows

- [Check Inventory] Display all items with inventory, including low stock items (≤ 5 units) highlighted in red.

Alternative Flows

- [No Low Stock] System displays normal menu without highlighting any items.

PROMPT #9

Timestamp: 2025-09-13T18:01:33.595007

Mode: PURE_LLM

PROMPT:

You are a senior product manager for a café ordering system. Your goal is to define a MINIMAL VIABLE PRODUCT (MVP) — the smallest set of features that validates the core loop: Customer places order → Staff fulfills → Customer picks up.

You write use cases in the EXACT format used in Proj1b1 (traffic light style):

Preconditions: [list]

Main Flow

1. [Step 1]
2. [Step 2]
3. [Step 3]

Subflows

- [Module Name] [Description, may include nested modules like [SubModule]]
- [Another Module] ...

Alternative Flows

- [Edge Case Name] [Description]

Example (from Proj1b1, simplified for MVP):

Preconditions: The item exists in the menu and is in stock.

Customer is on the ordering screen.

Main Flow

1. Customer selects item [Select Item].
2. System adds item to cart [Add Item to Cart].
3. Customer proceeds to checkout [Checkout].
4. Customer enters name and phone [Enter Contact Info].
5. Customer confirms order [Confirm Order].
6. System creates order and shows pickup code [Generate Pickup Code].

Subflows

- [Select Item] Display item name, price, and stock status. If out of stock, disable selection [Out of Stock].
- [Add Item to Cart] Store item ID and quantity. No modifiers (e.g., no “extra shot”).
- [Checkout] Show cart summary: items, subtotal, tax (hardcoded 8.5%), total. No tips, no discounts.
- [Enter Contact Info] Collect name (required) and phone (optional). No account creation.
- [Confirm Order] Validate name → create order → show “Order #123 ready in 5 min.”
- [Generate Pickup Code] Auto-generate 4-digit code. Display on screen and print receipt.

Alternative Flows

- [Out of Stock] Disable “Add to Cart” button. Show “Sold Out” badge.
- [No Name Entered] Show “Name is required for pickup” and block confirmation.
- [System Error] Show “Order failed — please try again or contact staff.”

RULES FOR MVP (CUT FROM 1B1):

- ~~×~~ NO user accounts (cut “Create Customer User”)
- ~~×~~ NO configurable tax (cut “Set Tax Rate” — hardcode 8.5%)
- ~~×~~ NO real-time notifications (cut “Order Pickup Notification” — use public board)
- ~~×~~ NO discounts or promo codes (cut “Apply Discount”)
- ~~×~~ NO item modifiers (cut “Add Customization” — only base items)
- ~~×~~ NO multi-location or complex inventory (cut “Set Inventory Location” — just quantity)

KEEP:

- Guest checkout (name + phone)
- Staff fulfillment (mark ready)
- Public pickup board (order # + first name)
- Basic inventory (add stock, low stock alert)
- Admin create item (name, price, link to inventory)

OUTPUT ONLY THE USE CASE. START WITH "Preconditions:". NO TITLES. NO DISCLAIMERS.

Now, generate 1 USE CASE for the following MVP requirement:

MVP: Order History — simplify 'View Order History' from 1b1. Customer enters phone → see last 3 orders. No login, no password, no export.

RESPONSE:

Preconditions: The customer has placed at least three orders in the system.

Main Flow

1. Customer enters their phone number [Enter Phone].
2. System retrieves and displays the last three orders [Retrieve Orders] with order #, date, and item(s) ordered.
3. Customer views order history [View Order History].

Subflows

- [Retrieve Orders] Fetch orders from database using phone number as identifier.

Alternative Flows

- [No Previous Orders] Display "You don't have any previous orders" message if customer has less than three orders.

PROMPT #10

Timestamp: 2025-09-13T18:01:45.331876

Mode: PURE_LLM

PROMPT:

You are a senior product manager for a café ordering system. Your goal is to define a MINIMAL VIABLE PRODUCT (MVP) — the smallest set of features that validates the core loop: Customer places order → Staff fulfills → Customer picks up.

You write use cases in the EXACT format used in Proj1b1 (traffic light style):

Preconditions: [list]

Main Flow

1. [Step 1]
2. [Step 2]
3. [Step 3]

Subflows

- [Module Name] [Description, may include nested modules like [SubModule]]
- [Another Module] ...

Alternative Flows

- [Edge Case Name] [Description]

Example (from Proj1b1, simplified for MVP):

Preconditions: The item exists in the menu and is in stock.

Customer is on the ordering screen.

Main Flow

1. Customer selects item [Select Item].
2. System adds item to cart [Add Item to Cart].
3. Customer proceeds to checkout [Checkout].
4. Customer enters name and phone [Enter Contact Info].
5. Customer confirms order [Confirm Order].
6. System creates order and shows pickup code [Generate Pickup Code].

Subflows

- [Select Item] Display item name, price, and stock status. If out of stock, disable selection [Out of Stock].
- [Add Item to Cart] Store item ID and quantity. No modifiers (e.g., no “extra shot”).
- [Checkout] Show cart summary: items, subtotal, tax (hardcoded 8.5%), total. No tips, no discounts.
- [Enter Contact Info] Collect name (required) and phone (optional). No account creation.
- [Confirm Order] Validate name → create order → show “Order #123 ready in 5 min.”
- [Generate Pickup Code] Auto-generate 4-digit code. Display on screen and print receipt.

Alternative Flows

- [Out of Stock] Disable “Add to Cart” button. Show “Sold Out” badge.
- [No Name Entered] Show “Name is required for pickup” and block confirmation.
- [System Error] Show “Order failed — please try again or contact staff.”

RULES FOR MVP (CUT FROM 1B1):

- ~~✗~~ NO user accounts (cut “Create Customer User”)
- ~~✗~~ NO configurable tax (cut “Set Tax Rate” — hardcode 8.5%)
- ~~✗~~ NO real-time notifications (cut “Order Pickup Notification” — use public board)
- ~~✗~~ NO discounts or promo codes (cut “Apply Discount”)
- ~~✗~~ NO item modifiers (cut “Add Customization” — only base items)
- ~~✗~~ NO multi-location or complex inventory (cut “Set Inventory Location” — just quantity)

KEEP:

- Guest checkout (name + phone)
- Staff fulfillment (mark ready)
- Public pickup board (order # + first name)
- Basic inventory (add stock, low stock alert)
- Admin create item (name, price, link to inventory)

OUTPUT ONLY THE USE CASE. START WITH "Preconditions:". NO TITLES. NO DISCLAIMERS.

Now, generate 1 USE CASE for the following MVP requirement:

MVP: View Staff — simplify 'Create/Edit Staff User' from 1b1. Admin sees list: Name, Role, Last Login. No edit, no delete, no permissions UI.

RESPONSE:

Preconditions: The admin is logged in and has permission to view staff.

Main Flow

1. Admin navigates to the staff view page [Navigate to Staff View].
2. System displays a list of staff members with columns for Name, Role, and Last Login [Display Staff List].
3. No edit or delete buttons are shown for each staff member [No Edit/Delete].

Subflows

- [Staff List] Display list of staff members with the specified columns.

Alternative Flows

- [Empty List] Show message "No staff members found" if no staff members exist in the system.