Google Gemini Pro (2.5) Use cases:

**UC1: Customer Places an Order**

- **Preconditions:** Customer is logged in or anonymous checkout is enabled.
- **Main Flow:**
    - Customer selects items to order.
    - System displays order summary with prices and tax.
    - Customer selects tip percentage or enters a custom tip.
    - Customer confirms the order.
    - System generates the order and assigns it to staff.
- **Subflows:**
    - [Select Items] Customer adds multiple items, including multiples of the same item.
    - [Confirm Order] If payment fails, customer is prompted to retry payment.
- **Alternative Flows:**
    - [Payment Failure] System notifies customer of payment failure and allows retry.
    - [Empty Cart] System prevents order confirmation if no items are selected.

**UC2: Staff Fulfills an Order**

- **Preconditions:** Staff member is logged in with fulfillment permissions.
- **Main Flow:**
    - Staff views list of pending orders.
    - Staff selects an order to fulfill.
    - System marks the order as in progress.
    - Staff completes preparation.
    - System marks order as ready for pickup.
- **Subflows:**
    - [Select Order] Staff can filter by order time or priority.
    - [Mark Ready] Staff confirms items are complete before updating status.
- **Alternative Flows:**
    - [Incorrect Item] Staff can edit the order if a mistake occurred.

**UC3: Customer Picks Up Order**

- **Preconditions:** Customer has an order marked ready for pickup.
- **Main Flow:**
    - System notifies customer that the order is ready.
    - Customer arrives at pickup location.
    - Staff confirms order pickup.
    - System marks order as completed.
- **Subflows:**
    - [Notify Customer] Notification via app, email, or SMS.
    - [Pickup Verification] Staff scans order ID or confirms customer identity.

- **Alternative Flows:**
    - [Order Not Ready] System informs customer if the order is not yet ready.

**UC4: Administrator Adds New Staff Member**

- **Preconditions:** Administrator is logged in with user management permissions.
- **Main Flow:**
    - Admin selects “Add Staff.”
    - Admin enters staff details and role.
    - System validates data and creates staff account.
- **Subflows:**
    - [Validate Data] System checks for missing or duplicate info.
    - [Assign Role] Admin selects permissions for staff.
- **Alternative Flows:**
    - [Invalid Input] System rejects incomplete or invalid input and prompts admin to correct.

**UC5: Administrator Edits Customer Profile**

- **Preconditions:** Administrator is logged in.
- **Main Flow:**
    - Admin searches for the customer.
    - Admin selects the customer profile.
    - Admin edits necessary details (e.g., contact info).
    - System saves changes and confirms update.
- **Subflows:**
    - [Search Customer] Admin can filter by name, email, or ID.
- **Alternative Flows:**
    - [Customer Not Found] System displays error and suggests retry.

**UC6: Administrator Sets Sales Tax Rate**

- **Preconditions:** Admin is logged in.
- **Main Flow:**
    - Admin navigates to tax settings.
    - Admin enters new tax rate.
    - System validates rate and updates pricing calculations.
- **Subflows:**
    - [Validate Rate] System ensures rate is within legal limits (e.g., the 2% food tax in Raleigh, NC).
- **Alternative Flows:**
    - [Invalid Rate] System rejects negative or non-numeric input.

**UC7: Customer Requests Anonymous Order**

- **Preconditions:** Anonymous checkout is enabled.

- **Main Flow:**
    - Customer selects items and proceeds to checkout without logging in.
    - System collects minimal necessary info (pickup code, payment).
    - Customer confirms order.
    - System generates order and assigns to staff.
- **Subflows:**
    - [Assign Pickup Code] System generates temporary order identifier.
- **Alternative Flows:**
    - [Payment Failure] System prompts for retry or cancellation.

**UC8: Staff Views Order History**

- **Preconditions:** Staff member is logged in.
- **Main Flow:**
    - Staff navigates to order history.
    - System displays completed and past orders.
    - Staff can filter orders by date, customer, or status.
- **Subflows:**
    - [Filter Orders] Staff can generate reports for inventory or performance.
- **Alternative Flows:**
    - [No Orders Found] System displays message if no orders match filters.

**UC9: Customer Leaves a Tip**

- **Preconditions:** Customer has items in cart and is at checkout.
- **Main Flow:**
    - Customer selects tip amount (15%, 20%, 25%, custom).
    - System calculates total with tip included.
    - Customer confirms payment.
    - System updates order with tip details.
- **Subflows:**
    - [Custom Tip] Customer enters manual tip amount.
- **Alternative Flows:**
    - [Negative Tip] System prevents negative or invalid input.

**UC10: Customer Receives Order Notification**

- **Preconditions:** Customer has an active order.
- **Main Flow:**
    - System tracks order status.
    - When order is ready, system sends notification (app, email, SMS).
    - Customer is informed of pickup location and order ID.
- **Subflows:**
    - [Notification Channels] System chooses available channel(s) for customer.
- **Alternative Flows:**

    - [Notification Failure] System retries sending or displays warning in app.

## Inspired by Public Health & Nutrition Research

#### UC11: Staff Manages Nutritional Information

- **Preconditions:** Staff member is logged in with permissions to edit menu items.
- **Main Flow:**
    - Staff selects a menu item to edit.
    - Staff navigates to the "Nutritional Information" section.
    - Staff enters or updates data (calories, fat, protein, carbs) and allergen warnings (e.g., "contains nuts, dairy").
    - System validates and saves the information.
- **Subflows:**
    - [Add Allergen] Staff selects from a predefined list of common allergens.
- **Alternative Flows:**
    - [Invalid Data] System rejects non-numeric input for nutritional values.

#### UC12: Customer Views Nutritional and Allergen Information

- **Preconditions:** Customer is browsing the menu.
- **Main Flow:**
    - Customer selects a menu item.
    - System displays the item's details, including a button or link for "Nutrition & Allergens."
    - Customer clicks to view detailed nutritional information and allergen warnings.
- **Alternative Flows:**
    - [Info Unavailable] If no information is entered for an item, the system displays "Nutritional information not available."

## Inspired by Consumer Behavior & Loyalty Research

#### UC13: Customer Views Order History and Reorders

- **Preconditions:** Customer is logged into their account.
- **Main Flow:**
    - Customer navigates to their profile and selects "Order History."
    - System displays a list of their past orders.
    - Customer selects a past order and clicks a "Reorder" button.
    - System adds all items from the selected past order to the current cart for checkout.
- **Alternative Flows:**

    - [Item Unavailable] If an item from the past order is no longer available, the system adds the available items and notifies the customer which items could not be added.

**UC14: Administrator Creates a Promotional Discount Code**

- **Preconditions:** Administrator is logged in with marketing permissions.
- **Main Flow:**
    - Admin navigates to "Promotions" and selects "Create Discount."
    - Admin enters discount details (e.g., code name "WOLF15", percentage/fixed amount, expiration date, usage limits).
    - System validates the data and activates the discount code.
- **Alternative Flows:**
    - [Duplicate Code] System prevents the creation of a discount code that already exists.

**UC15: Customer Applies a Discount Code**

- **Preconditions:** Customer has items in their cart and is at the checkout screen.
- **Main Flow:**
    - Customer enters a valid discount code into the "Promo Code" field.
    - Customer applies the code.
    - System validates the code, applies the discount to the order subtotal, and updates the final price.
- **Alternative Flows:**
    - [Invalid/Expired Code] System displays an error message that the code is invalid or expired, and the total remains unchanged.

**UC16: Customer Leaves a Rating and Review**

- **Preconditions:** Customer has a completed order in their history.
- **Main Flow:**
    - Customer navigates to a completed order in their "Order History."
    - Customer selects a star rating (1-5) and optionally enters a text review.
    - Customer submits the feedback.
    - System saves the rating and review associated with the order.
- **Alternative Flows:**
    - [Review Already Submitted] System prevents the customer from reviewing the same order twice.

## Inspired by Operational Efficiency & Business Management

**UC17: Staff Marks an Item as "Out of Stock"**

- **Preconditions:** Staff member is logged in.

- **Main Flow:**
    - Staff navigates to the "Inventory Management" or "Menu Management" page.
    - Staff selects an item that is currently unavailable.
    - Staff toggles the item's status to "Out of Stock."
    - System immediately removes the item from the customer-facing menu.
- **Subflows:**
    - [Set Re-Stock ETA] Staff can optionally enter an estimated time when the item will be available again.

#### UC18: Manager Views Sales and Feedback Reports

- **Preconditions:** A user with a "Manager" role is logged in.
- **Main Flow:**
    - Manager navigates to the "Dashboard" or "Reports" section.
    - Manager selects a report type (e.g., "Sales Summary," "Customer Feedback").
    - Manager filters the report by a date range.
    - System generates and displays the report with relevant data and visualizations.
- **Alternative Flows:**
    - [No Data] System displays a message indicating no data is available for the selected period.

#### UC19: Customer Customizes a Menu Item

- **Preconditions:** Customer is adding an item to their order.
- **Main Flow:**
    - Customer selects a configurable menu item (e.g., a latte).
    - System displays customization options (e.g., milk type, extra shot, syrup flavor).
    - Customer makes their selections.
    - System updates the item price based on selections and adds the customized item to the cart.
- **Subflows:**
    - [Upcharges] Certain options add an additional cost, which is clearly displayed.

## Inspired by Legal, Tax & Privacy Regulations

#### UC20: Administrator Configures Advanced Tax Rules

- **Preconditions:** Administrator is logged in.
- **Main Flow:**
    - Admin navigates to "Tax Settings."
    - Admin configures different tax rates based on item category (e.g., "Prepared Food," "Packaged Goods," "Hot Beverage"), reflecting complex local tax laws mentioned in the articles.
    - System saves the rules.

    - During checkout, the system applies the correct tax rate to each item in the cart based on its category.
- **Alternative Flows:**
    - [No Category Rule] If an item has no category-specific rule, the system applies the default tax rate.

**UC21: Customer Requests Personal Data Download (Privacy)**

- **Preconditions:** Customer is logged into their account.
- **Main Flow:**
    1. Customer navigates to "Privacy Settings" in their profile.
    2. Customer selects "Download My Data."
    3. System verifies the request (e.g., re-enter password).
    4. System compiles the user's personal data (profile info, order history) into a portable format (e.g., JSON, CSV) and sends it to their verified email address.

**UC22: Customer Requests Account Deletion**

- **Preconditions:** Customer is logged in.
- **Main Flow:**
    1. Customer navigates to "Account Settings" and selects "Delete Account."
    2. System displays a confirmation warning about the irreversible nature of the action.
    3. Customer confirms deletion.
    4. System deletes the user's personal identifiable information (PII) and anonymizes their past orders for sales reporting purposes.

## Inspired by App Development & Technology Features

**UC23: User Resets a Forgotten Password**

- **Preconditions:** User is on the login page.
- **Main Flow:**
    - User clicks the "Forgot Password?" link.
    - User enters their account email address.
    - System sends a password reset link to that email.
    - User clicks the link, is taken to a secure page, and enters a new password.
    - System updates the user's credentials.
- **Alternative Flows:**
    - [Email Not Found] System informs the user if the provided email does not match any account.
    - [Link Expired] The reset link is valid for a limited time; if expired, the user is prompted to start over.

**UC24: Customer Saves a Payment Method**

- **Preconditions:** Customer is logged in and at the payment stage of checkout.
- **Main Flow:**
    - Customer enters their credit card information.
    - Customer checks a box labeled "Save this card for future orders."
    - System securely tokenizes and saves the payment information to the customer's profile via a payment gateway.
- **Subflows:**
    - [Use Saved Card] On subsequent orders, the customer can select their saved card instead of re-entering details.

**UC25: Customer Sets a Favorite Order**

- **Preconditions:** Customer is logged in and has at least one past order.
- **Main Flow:**
    1. Customer views their order history.
    2. Customer marks an order as a "Favorite."
    3. On the home screen or in their profile, a "My Favorite" section appears.
    4. Customer can click a single button to add their favorite order directly to the cart.

**UC26: Staff Receives a New Order Notification**

- **Preconditions:** A staff member is logged into the order fulfillment view.
- **Main Flow:**
    1. A customer successfully places a new order.
    2. The staff-facing system plays an audible alert and/or displays a prominent visual notification.
    3. The list of pending orders automatically refreshes to show the new order at the top.

**UC27: Administrator Sets Cafe Operating Hours**

- **Preconditions:** Administrator is logged in.
- **Main Flow:**
    - Admin navigates to "Store Settings."
    - Admin sets the opening and closing times for each day of the week.
    - System saves the hours. The customer-facing application now prevents orders from being placed outside of these hours.
- **Alternative Flows:**
    - [Holiday Closure] Admin can set specific dates where the cafe is closed, disabling all orders for that day.

**UC28: Customer Schedules an Order for a Future Time**

- **Preconditions:** Customer is at checkout.
- **Main Flow:**
    1. System displays an option to "Pick up ASAP" or "Schedule for later."

2. Customer selects "Schedule for later."
3. Customer chooses a future date and time slot from a list of available times.
4. Customer completes the order.
5. System holds the order and releases it to the staff fulfillment queue at the appropriate time before the scheduled pickup.

**UC29: Staff Cancels an Order**

- **Preconditions:** An order is in the "Pending" or "In Progress" state.
- **Main Flow:**
    - Staff selects an active order that needs to be canceled (e.g., due to an ingredient shortage).
    - Staff clicks "Cancel Order" and provides a reason.
    - System triggers a refund to the customer's payment method.
    - System sends a notification to the customer explaining the cancellation.
- **Alternative Flows:**
    - [Manager Approval] Cancellation of paid orders might require manager-level permission.

**UC30: System Places an Order into a Pickup Queue**

- **Preconditions:** Multiple orders are marked "Ready for Pickup."
- **Main Flow:**
    1. Multiple orders are completed by staff.
    2. System displays the orders on a customer-facing screen (e.g., a TV monitor in the cafe).
    3. The screen shows the Order ID or a pickup code for each completed order.
    4. When an order is picked up (UC3), the system removes its ID from the queue screen.

ChatGPT-5 Thinking Use Cases:

**UC1: Customer Places an Order**
**Preconditions:** Customer is logged in or anonymous checkout is enabled; menu and prices are up to date.
**Main Flow:**

1. Customer browses the menu and selects items (with options/modifiers).

2. System builds a dynamic order summary (items, modifiers, quantity, subtotals, taxes/fees).

3. Customer chooses a tip (preset or custom) and reviews final total.

4. Customer selects payment method and confirms the order.

5. System authorizes payment, creates the order record, assigns a unique order ID, and routes to staff with a “Pending” status.
   **Subflows:**
   • [Select Items] Quantity changes update availability and totals in real time.
   • [Confirm Order] Payment UI supports saved cards and one-time methods; system shows a clear breakdown of charges.
   **Alternative Flows:**
   • [Payment Failure] System explains the error and offers retry/change-method; idempotency prevents double charges.
   • [Empty Cart] Checkout button disabled until an item is added; system shows a prompt to continue shopping.

---

**UC2: Staff Fulfills an Order**
**Preconditions:** Staff is logged in with fulfillment permissions; there are orders in “Pending.”
**Main Flow:**

1. Staff views the queue, sorted by promise time and priority flags.

2. Staff claims an order; status changes to “In Progress” with timestamp.

3. System displays item list, modifiers, and any allergen or special instructions.

4. Staff prepares items and marks each component complete.

5. Staff marks the order “Ready for Pickup,” triggering customer notification.
   **Subflows:**
   • [Select Order] Filters by status, wait time, or station (drinks/food).
   • [Mark Ready] System validates all items are completed; prompts for bag seal code or sticker ID if required.
   **Alternative Flows:**
   • [Incorrect Item] Staff edits/replaces an item (with audit note); system updates totals/stock and, if needed, seeks customer approval.

---

**UC3: Customer Picks Up Order**
**Preconditions:** Customer has an order marked “Ready”; pickup location is open.
**Main Flow:**

1. System sends a “Ready” notification with pickup instructions and order ID/QR.

2. Customer arrives and presents the pickup code/QR.

3. Staff verifies match (name/code/last four digits of phone).

4. System marks order “Completed,” recording the pickup time.
   **Subflows:**
   • [Notify Customer] App push, SMS, and/or email based on user preferences.
   • [Pickup Verification] Optional last-name or partial phone verification for anonymous checkout.
   **Alternative Flows:**
   • [Order Not Ready] App shows live status; staff provides ETA and offers to wait or cancel if delayed beyond threshold.

---

**UC4: Administrator Adds New Staff Member**
**Preconditions:** Admin is logged in with user-management permissions.
**Main Flow:**

1. Admin opens “Staff Management” and selects “Add Staff.”

2. Admin enters name, contact, role(s), and location(s).

3. System validates uniqueness and required fields; creates the account.

4. Invitation email/SMS is sent with first-login instructions.
   **Subflows:**
   • [Validate Data] Enforce strong temporary password or SSO binding.
   • [Assign Role] Role templates (Barista, Cashier, Manager) set default permissions.
   **Alternative Flows:**
   • [Invalid Input] System highlights fields and explains fixes; prevents creation until resolved.

---

**UC5: Administrator Edits Customer Profile**
**Preconditions:** Admin is logged in and has customer-data access.
**Main Flow:**

1. Admin searches for a customer by name, email, phone, or ID.

2. Admin opens profile and edits contact data or preferences (e.g., language, notifications).

3. System saves changes, records an audit entry, and confirms success.
   **Subflows:**
   • [Search Customer] Advanced filters (date created, last order, loyalty tier).
   **Alternative Flows:**
   • [Customer Not Found] System suggests refined filters or creating a support ticket for assistance.

---

**UC6: Administrator Sets Sales Tax Rate**
**Preconditions:** Admin is logged in; environment allows manual rate (e.g., pilot/campus).
**Main Flow:**

1. Admin navigates to “Tax Settings.”

2. Admin inputs new rate(s) or schedule(s) and effective dates.

3. System validates format and updates calculation rules for new orders.
   **Subflows:**
   • [Validate Rate] Previews example orders showing old vs new totals before saving.
   **Alternative Flows:**
   • [Invalid Rate] System blocks save and displays corrective guidance.

---

**UC7: Customer Requests Anonymous Order**
**Preconditions:** Anonymous checkout is enabled; payment and pickup code workflows are active.
**Main Flow:**

1. Customer adds items and proceeds to checkout without logging in.

2. System collects minimal info (e.g., initials and phone or pickup code only).

3. Customer pays and confirms.

4. System creates order, generates pickup code, and assigns to staff.
   **Subflows:**
   • [Assign Pickup Code] Short code/QR generated and stored with the order.
   **Alternative Flows:**

• [Payment Failure] Customer can retry with a different method or cancel.

---

**UC8: Staff Views Order History**
**Preconditions:** Staff is logged in; order history feature is enabled for role.
**Main Flow:**

1. Staff opens “Order History.”

2. System displays past orders with filters (date range, status, customer).

3. Staff exports or prints a subset for shift reports.
   **Subflows:**
   • [Filter Orders] Quick filters for “Late,” “Refunded,” or “Allergen orders.”
   **Alternative Flows:**
   • [No Orders Found] UI suggests removing filters or expanding date range.

---

**UC9: Customer Leaves a Tip**
**Preconditions:** Items exist in cart; payment method selected.
**Main Flow:**

1. Customer chooses a tip (preset or custom).

2. System updates the order total and shows the new breakdown.

3. Customer confirms payment.

4. System stores tip allocation for payouts.
   **Subflows:**
   • [Custom Tip] Input field supports currency format and rounding.
   **Alternative Flows:**
   • [Invalid Amount] System rejects negative or malformed values and prompts correction.

---

**UC10: Customer Receives Order Notification**
**Preconditions:** Customer has an active order; notifications are configured.
**Main Flow:**

1. System watches status changes (In Progress → Ready → Completed).

2. When “Ready,” notification is sent with pickup steps.

3. Customer can view live map/status (optional).
   **Subflows:**
   • [Notification Channels] Fallback logic if push fails (send SMS/email).
   **Alternative Flows:**
   • [Notification Failure] App shows banner; user can pull-to-refresh status.

---

**UC11: Restaurant & Item Discovery (Search/Filters)**
**Preconditions:** Catalog is indexed with cuisine, tags, distance/ETA, and availability.
**Main Flow:**

1. Customer searches or browses featured categories.

2. Customer applies filters (diet, allergen, price, distance).

3. System sorts results by relevance or user preference and shows delivery/pickup times.
   **Subflows:**
   • [Personalized Re-Rank] Prioritize frequent cuisines or favorites.
   • [Promo Flags] Clearly label discounts, BOGO, and free-delivery thresholds.
   **Alternative Flows:**
   • [Sparse Market] Expand radius, suggest preorder or nearby pickup alternatives.

---

**UC12: Nutrition Display & Facts**
**Preconditions:** Nutrition data is captured for items (mandatory or voluntary).
**Main Flow:**

1. Customer opens item details.

2. System displays calories per serving and highlights key nutrients.

3. Customer expands “Full Nutrition” for extended facts and disclaimers.

4. Customer adds item to cart, preserving visibility of nutrition info.
   **Subflows:**
   • [Reference Intakes] Short context (e.g., “% of daily value”).
   • [Allergen Icons] Prominent placement near “Add to Cart.”

**Alternative Flows:**
• [No Data] Show a consistent message and prompt to contact the restaurant.

---

**UC13: Allergen & Dietary Safety Filters**
**Preconditions:** Items are tagged with allergens and dietary attributes.
**Main Flow:**

1. Customer toggles specific allergens/diets to avoid.

2. System hides unsafe items and marks “may contain” with explanations.

3. Customer selects safe items; system preserves filters across sessions.
   **Subflows:**
   • [Persistent Preferences] Store in profile; apply on future visits.
   • [Cross-Contact Advisory] Remind customers about shared preparation areas.
   **Alternative Flows:**
   • [Unknown Tags] Soft-block or de-emphasize items with missing data.

---

**UC14: Healthier Defaults & Nudges**
**Preconditions:** Nutrient scoring and swap suggestions are configured.
**Main Flow:**

1. List view highlights healthier options first (badge, short note).

2. When customer adds an item, system suggests a healthier swap.

3. Customer accepts/declines; cart updates accordingly.
   **Subflows:**
   • [Kids Defaults] Water/fruit auto-selected for kids’ meals (editable).
   • [Price Delta] Show price difference and nutrition trade-offs clearly.
   **Alternative Flows:**
   • [Opt-Out] Customer disables nudges in settings; preference persists.

---

**UC15: Menu-Labeling Compliance (Covered Establishments)**
**Preconditions:** Establishment is flagged as covered; menu versioning is enabled.
**Main Flow:**

1. Admin sets “covered” flag with effective date.

2. System requires calorie display on menus and item details.

3. System retains a change log for audits and updates.
   **Subflows:**
   • [Combos/Ranges] Handle variable options and display calorie ranges.
   • [Change Log] Track who changed what and when.
   **Alternative Flows:**
   • [Not Covered] Optional fields remain available without enforcement.

---

**UC16: WIC — Eligible Item Filtering**
**Preconditions:** Customer selects WIC; state-approved UPC/size list is current.
**Main Flow:**

1. Customer toggles “Pay with WIC.”

2. System restricts the catalog to eligible items and package sizes.

3. Customer adds eligible items to a dedicated WIC cart.
   **Subflows:**
   • [Benefit Balance Check] Show remaining benefits and alerts on shortfalls.
   • [State Substitutions] Offer permitted brand or size substitutions.
   **Alternative Flows:**
   • [Mixed Cart] System auto-splits WIC vs non-WIC items into separate tenders.

---

**UC17: WIC — Split Tender for Fees**
**Preconditions:** WIC cart exists; order includes delivery/service/bag fees.
**Main Flow:**

1. System totals WIC-eligible items.

2. Fees and ineligible items are routed to a non-WIC payment.

3. Customer authorizes WIC payment (e.g., PIN), then pays remaining balance with card.
   **Subflows:**
   • [Receipt Partition] Distinct lines for each tender on the receipt.
   • [Refund Rules] Enforce program-specific constraints on returns.
   **Alternative Flows:**

• [Fee Waiver] If fees are waived, a single WIC tender completes the order.

---

**UC18: WIC — Remote Issuance & Wallet**
**Preconditions:** State supports remote issuance; identity verification succeeds.
**Main Flow:**

1. Customer requests a benefit load in the app.

2. Wallet shows updated benefits with start/end dates.

3. System refreshes balances after each WIC redemption.
   **Subflows:**
   • [Device Binding] Pair wallet to device/account for security.
   • [Card Replacement] Handle lost/reissued cards seamlessly.
   **Alternative Flows:**
   • [Not Supported] App provides instructions for in-person issuance.

---

**UC19: WIC — Integrity & Audit Reporting**
**Preconditions:** WIC transactions, approvals, and exceptions are recorded.
**Main Flow:**

1. Admin opens the WIC dashboard.

2. System compiles transactions, eligibility checks, overrides, and declines.

3. Admin exports reports in the required file format for agency review.
   **Subflows:**
   • [Exception Log] Highlight suspected violations or disallowed redemptions.
   • [Data Quality] Tools to fix missing UPC or size metadata.
   **Alternative Flows:**
   • [Data Gaps] Block further WIC checkout until critical gaps are corrected.

---

**UC20: Real-Time Availability & Smart Substitutions**
**Preconditions:** Inventory sync is active; substitution rules are defined.
**Main Flow:**

1. When an item becomes out-of-stock, the system hides or flags it.

2. System proposes substitutes that match size/price/allergen constraints.

3. Customer accepts or rejects; totals and prep time update.
   **Subflows:**
   • [Allergen-Safe Sub] Only show substitutes that respect active allergen filters.
   • [WIC-Eligible Sub] Suggest permitted alternatives for WIC carts.
   **Alternative Flows:**
   • [No Substitute] Offer waitlist or remove item with a one-tap undo.

---

**UC21: Food Safety Compliance Workflow**
**Preconditions:** Food safety program is configured for the location.
**Main Flow:**

1. Pre-shift staff completes illness/hygiene attestation.

2. System schedules hot/cold-hold temperature checks with reminders.

3. System prompts cleaning/sanitization tasks throughout the day.

4. Manager reviews logs, signs off, and addresses exceptions.
   **Subflows:**
   • [Allergen Cross-Contact] Tickets show required prep steps; staff confirms completion.
   • [Corrective Actions] Capture what was done, by whom, and when.
   **Alternative Flows:**
   • [Device Offline] Manual entries allowed with photo proof; sync later.

---

**UC22: Recall & Advisory Handling**
**Preconditions:** Recall/advisory feed is enabled; lots are linked to items.
**Main Flow:**

1. System ingests a recall/advisory for specific lots or SKUs.

2. Affected items are automatically blocked from sale.

3. System alerts admins, staff, and customers with recent purchases.
   **Subflows:**
   • [Lot Substitution] Swap to safe inventory where possible.

• [Customer Remediation] Standardized refund/replace flows with timestamps.
**Alternative Flows:**
• [Unknown Lot] Temporarily block the entire item class until resolved.

---

**UC23: Contactless Pickup/Delivery**
**Preconditions:** Customer selects contactless; address or pickup locker is valid.
**Main Flow:**

1. App provides exact placement instructions and a one-time code if applicable.

2. Driver completes contactless drop and captures photo proof.

3. System confirms delivery with time and location metadata.
   **Subflows:**
   • [Secure Entry] Provide temporary access/locker codes; auto-expire after use.
   • [Temp-Sensitive] Emphasize placement guidance for hot/cold items.
   **Alternative Flows:**
   • [Verification Required] In-person check if policy or tender (e.g., specific programs) requires it.

---

**UC24: Driver Hazard & Safety Reporting**
**Preconditions:** Driver is online; device has GPS and data connection (or offline queue).
**Main Flow:**

1. Driver taps “Report Hazard” during or after a route.

2. Driver categorizes the hazard and adds optional notes/photo.

3. System updates a hazard map, adjusts routing, and notifies operations for hotspots.
   **Subflows:**
   • [Escalation] Critical events generate incident tickets for follow-up.
   • [Offline Queue] Reports are stored and synced when online.
   **Alternative Flows:**
   • [False Alarm] Operations closes with a reason to improve model accuracy.

---

**UC25: Cold-Chain Priority Handling**
**Preconditions:** Order contains perishable or temperature-controlled items.
**Main Flow:**

1. System flags the order and prioritizes dispatch.

2. App shows an insulated-bag checklist and time-to-customer target.

3. Delivery photo and timestamp are captured as proof.
   **Subflows:**
   • [Temp Tracker] Optional sensor handoff and readback flow.
   • [Locker Drop] Use temperature-appropriate lockers where available.
   **Alternative Flows:**
   • [Exceeded Window] System proactively offers refund or replacement.

---

**UC26: Jurisdiction-Aware Sales Tax & Fee Taxability**
**Preconditions:** Merchant and customer locations are known; rules are configured.
**Main Flow:**

1. System determines sourcing (origin vs destination) and applicable jurisdictions.

2. System applies item and fee taxability rules (delivery/service/bag).

3. System calculates taxes, stores the breakdown and an audit trace.

4. Calculations are displayed transparently in the checkout summary.
   **Subflows:**
   • [Local Meals/City Taxes] Add local overlays as required.
   • [Economic Nexus] Monitor thresholds and adjust collection logic.
   **Alternative Flows:**
   • [Manual Override] Allowed with reason, approver, and audit note.

---

**UC27: Refunds, Chargebacks & Adjustments with Tax/Tip Recalculation**
**Preconditions:** Order is completed; an adjustment or dispute exists.
**Main Flow:**

1. Staff selects item(s)/fees to adjust or refund (full or partial).

2. System recalculates taxes, fees, and tip allocations accordingly.

3. System issues refund, updates ledgers, and labels the event for exports.
   **Subflows:**
   • [Promo/Gift Card Scope] Determine taxability after discounts.
   • [Marketplace Fees] Adjust platform vs merchant shares per policy.
   **Alternative Flows:**
   • [Program Restrictions] Enforce tender-specific limits before issuing refunds.

---

**UC28: Tax Filing & Merchant Reconciliation**
**Preconditions:** Accounting period is closed; data is locked for reporting.
**Main Flow:**

1. System aggregates tax by jurisdiction and tender type.

2. System generates returns and payment files for the period.

3. System reconciles marketplace-collected vs merchant-collected sales to prevent double reporting.
   **Subflows:**
   • [Exemption Certificates] Store, validate, and apply to orders.
   • [Amended Returns] Create revisions with full audit trails.
   **Alternative Flows:**
   • [Mismatch Detected] Exception report prompts data cleanup before filing.

---

**UC29: Privacy & Data Rights Center**
**Preconditions:** User is authenticated or can complete identity verification.
**Main Flow:**

1. User reviews current consents (analytics, marketing, personalization).

2. User opts in/out and saves preferences; system records timestamp and scope.

3. User requests data access or deletion; system fulfills or denies with rationale.
   **Subflows:**
   • [Retention Policy] Apply retention schedules and archival rules.
   • [Breach Notices] User can set preferred contact channel.
   **Alternative Flows:**

• [Guest User] Preferences stored locally until an account is created.

---

**UC30: Inclusive Access (Accessibility & Language/Literacy Support)**
**Preconditions:** Accessibility or locale preferences are detected or chosen.
**Main Flow:**

1. System enables high-contrast themes, keyboard navigation, proper focus order, and screen-reader labels.

2. System presents localized UI text with clear icons/pictograms for key actions.

3. Checkout provides spoken/visual confirmation of totals, tips, and pickup instructions.
   **Subflows:**
   • [Reduced Motion] Minimize animations; respect OS settings.
   • [RTL & Glossaries] Support right-to-left languages and simple allergen glossaries.
   **Alternative Flows:**
   • [Unavailable Language] Default to English with concise tooltips and iconography.

What Gemini thinks we are missing from part 1a1:

## # Introduction and Problem Overview

Your current introduction likely frames the project as a technical task of building an ordering system. You can elevate this by incorporating the broader context from the articles.

- **What to Change:** Add language that frames WolfCafe not just as a tool, but as a modern, responsible digital platform.
- **Why:** The articles on public health, consumer behavior, and privacy show that successful platforms are built on trust and customer-centric values.
- **Example Wording to Add:**
  "The goal of the WolfCafe project is to develop a seamless online ordering system that not only provides operational efficiency but also fosters customer loyalty through an accessible and transparent experience. The system will be designed to empower customers with clear nutritional information, protect their privacy, and provide a reliable, user-friendly interface."

---

## ## Scope

The scope section defines the project's boundaries. Explicitly adding items mentioned in the articles makes your project's goals clearer.

- **What to Change:** Add specific items to the "In Scope" list that directly reflect the new, article-inspired requirements.
- **Why:** This formally includes the new feature domains within the project's boundaries, ensuring they are not overlooked during design and development.
- **Example "In Scope" Items to Add:**
    - Management and display of menu item **nutritional information and allergen warnings**.
    - Functionality for **promotional codes and customer feedback**.
    - Implementation of user controls for **data privacy**, including data access and account deletion.

---

## ## Non-Functional Requirements (NFRs)

This is the section where the articles can have the biggest impact. Your current NFRs are likely generic. Let's make them specific, professional, and directly linked to your research.

**1. Add a "Privacy & Data Governance" Section 🔐**

The requirement for a privacy policy and the articles on data handling (like HIPAA) imply this is critical. A dedicated NFR section is the best practice.

- **What to Change:** Create a new top-level NFR for privacy.
- **Why:** This moves beyond basic security and addresses how user data is ethically and legally handled, a major theme in the regulatory documents.
- **Example Wording:**
    - **Data Minimization:** The system shall only collect personal data that is essential for order processing, account management, and improving the user experience.
    - **User Control:** The system must provide users with the ability to access and delete their personal data, in compliance with the WolfCafe Privacy Policy.
    - **Data Anonymization:** For reporting and analytics, personally identifiable information (PII) must be anonymized to protect user privacy.

**2. Enhance the "Usability" Section to "Accessibility & Usability" ♿**

The project description and app development guides emphasize making technology accessible to everyone, including users of programs like WIC.

- **What to Change:** Make the usability requirement specific and add a focus on accessibility.
- **Why:** A generic "easy to use" statement is not a testable requirement. Referencing established standards makes it concrete.

- **Example Wording:**
    - **Compliance:** The user interface must comply with **Web Content Accessibility Guidelines (WCAG) 2.1 Level AA** standards.
    - **Navigability:** All primary functions and workflows must be fully navigable using only a keyboard.
    - **Screen Reader Support:** All informational images, icons, and form fields must include appropriate labels and alternative text for screen reader compatibility.

**3. Add a "Data Accuracy" Requirement ✅**

This is crucial, especially with the introduction of health-related information.

- **What to Change:** Add a new NFR ensuring the integrity of sensitive data.
- **Why:** The articles on food safety and nutrition make it clear that providing incorrect allergen or nutritional information can have serious health consequences.
- **Example Wording:**
    - **Integrity:** Nutritional and allergen data displayed to the customer must have 100% integrity with the data entered by authorized staff. The system must prevent data corruption from input to display.

**4. Add an "Extensibility" Requirement 🔧**

The articles on complex tax laws and evolving business needs suggest a system built for today will be obsolete tomorrow. A good design plans for the future.

- **What to Change:** Add an NFR that focuses on the system's ability to adapt.
- **Why:** This shows foresight. It ensures that the system can be updated to handle new business rules (like complex taxes) or features without a complete rewrite.
- **Example Wording:**
    - **Financial Modularity:** The system’s financial modules, particularly for taxation and promotions, must be designed to allow for the addition of new rules without requiring significant architectural refactoring.
    - **Auditability:** The system must generate detailed, immutable logs for all financial transactions and order status changes to support auditing and troubleshooting.

What ChatGPT-5 Thinking states is missing from 1a1:

# What’s missing in your 1a1 (evidence-based)

- **Nutrition transparency & “healthier defaults” on OFD** (calorie display, healthier sorting/defaults, warning icons, kid-directed marketing controls) are absent. These are widely recommended for online delivery platforms and are increasingly reflected in policy

discussions.

- **WIC online ordering specifics** (eligible-item filtering, split/two-tender handling of delivery fees, eWIC remote issuance/verification, integrity reporting) aren't modeled. The USDA proposed rule explicitly enables online WIC, remote issuance, and separate tender for delivery/service fees while stressing program integrity and vendor oversight.

- **Food safety & retail code operations** (employee illness attestation, handwashing, allergen controls, temperature logs, cleaning events, mobile food unit SOPs) are not in flows for staff/restaurants. State adoptions of the FDA Food Code emphasize these controls.

- **Menu labeling support** for covered establishments (calorie counts and nutrient statements in digital menus) isn't in scope; FDA maintains menu-labeling guidance that covered chains must follow (your platform should support compliance).

- **Sales-tax reality**: You hard-code "NC's 2%" in stakeholders; in practice, taxability varies by **state, locality, and fee type**, and marketplace facilitator laws can shift collection/remittance to the platform. You need dynamic, jurisdiction-aware tax logic (items vs prepared food, delivery vs service fees, local meals taxes, marketplace facilitator status), audit trails, and monthly returns.

- **Privacy & data governance**: You list "Privacy & Security Office," but there are no flows for data minimization, consent/opt-out, data retention, breach response, or DSRs (access/delete). Also clarify that HIPAA's Privacy Rule applies to covered entities/business associates handling PHI—most OFD platforms won't be covered *un*l*ess* they handle PHI via a covered entity partnership—so general consumer privacy/security still apply. Add explicit flows.

- **Driver safety & operations** (hazard reporting, safe-delivery options, contactless protocols) and **driver/admin tax documentation** (tip allocation, payout summaries) are missing. Safety is a recognized issue in the literature; tax summaries reduce friction for workers.

**Cost: $20/month**
**$20/month for GPT-5 Plus subscription**
**$20/month for Gemini Pro subscription**

## Zero-Shot vs Careful Prompting:

The zero-shot prompting approach, where the LLM was simply asked to generate more use cases with minimal context and structure, yielded largely unsatisfactory results. The generated use cases were often generic and suffered from significant formatting issues, such as missing required flows like subflows or alternative flows. More critically, because the model lacked the necessary context from the supporting research articles, its output failed to incorporate the specific, evidence-based features that would truly enhance the product's value.

In direct contrast, the careful prompting strategy proved far more effective. By providing the LLM with proper context, a clear structure, and the relevant articles, it was able to generate superior use cases. This method resulted in outputs with significantly improved formatting, greater detail, and a stronger overall structure. Most importantly, the LLM could directly reference and integrate concepts from the provided research, grounding the use cases in real-world requirements and adding meaningful depth

## Reflection:

While the outputs from both LLMs shared a similar high-level structure, they differed profoundly in their strategic focus and level of detail. Gemini Pro produced a well-organized and logical set of use cases that expertly define the core operational workflows of the ordering system. This output provides exceptional clarity and establishes a foundation of testable, functional requirements, effectively creating the essential blueprint for what needs to be built immediately.

In contrast, ChatGPT's use cases were strategically focused on integrating advanced considerations from public health, regulatory compliance, and emerging technological trends. Its output serves as a forward-looking roadmap, detailing the "why" behind specific features and the "what else" we must consider to ensure the product is competitive, compliant, and resilient in a complex market.