Gemini Pro

Part 1: New Use Cases for the MVP (Minimal Viable Product)

Here are 10 new use cases that define the core functionality of the WolfCafe MVP. They are designed to validate the primary user journey—ordering and fulfillment—with the minimum required features.

---

01. Use Case: Manage Menu Item

- **Actor:** Administrator, Manager
- **Description:** An authorized user creates, edits, or deactivates a menu item, including its price, description, and basic health information like calorie counts and allergen tags. This use case consolidates adding and updating items.
- **Preconditions:**
  - The user is logged in with Administrator or Manager privileges.
- **Main Flow:**
  - The user navigates to the "Menu Management" section.
  - The user selects the option to create a new item or selects an existing item to edit.
  - The system presents a form with fields for the item's name, description, price, calorie count, and a list of allergen tags (e.g., "Contains Nuts," "Gluten-Free").
  - The user enters or modifies the details.
  - The user saves the changes.
  - The system validates that the price and calorie count are non-negative numerical values.
  - The system updates the menu item in the database.
  - The system displays a success message.
- **Subflows:**
  - **S1: Deactivate Item:** At step 2, the user selects an item and chooses to "deactivate." The system makes the item unavailable for ordering but retains its data for historical records.
- **Alternative Flows:**
  - **A1: Invalid Input:** At step 6, if the system detects invalid input (e.g., text in the price field), it displays an error message, highlights the problematic field, and prompts the user to correct it, resuming at step 4.

---

02. Use Case: Manage Inventory

- **Actor:** Staff, Manager

- **Description:** A staff or manager updates the stock level for a menu item to ensure the menu accurately reflects what can be prepared.
- **Preconditions:**
  - The user is logged in with Staff or Manager privileges.
  - The menu item already exists in the system.
- **Main Flow:**
  - The user navigates to the "Inventory Management" section.
  - The system displays a list of menu items and their current stock counts.
  - The user selects the item whose inventory they want to update.
  - The user enters the new total quantity available.
  - The user confirms the update.
  - The system validates the input is a non-negative integer.
  - The system updates the inventory count for the selected item.
  - The system displays a confirmation message with the new stock level.
- **Alternative Flows:**
  - **A1: Invalid Quantity Entered:** At step 6, if the input is not a valid non-negative integer, the system shows an error and prompts for correction, resuming at step 4.

---

03. Use Case: Configure System Tax Rate

- **Actor:** Administrator
- **Description:** An administrator sets a single, system-wide sales tax rate that will be applied to all taxable orders.
- **Preconditions:**
  - The user is logged in as an Administrator.
- **Main Flow:**
  - The Administrator navigates to the "System Settings" panel.
  - The Administrator locates the "Sales Tax Rate" input field.
  - The Administrator enters a numerical value for the tax percentage.
  - The Administrator saves the setting.
  - The system validates the input is a valid, non-negative number.
  - The system updates the global sales tax rate.
  - The system displays a success message.
- **Alternative Flows:**
  - **A1: Invalid Tax Rate Input:** At step 5, if the input is invalid (e.g., contains text or is negative), the system displays an error message and the use case resumes at step 3 for re-entry.

---

04. Use Case: Register Customer Account

- **Actor:** Customer
- **Description:** A new user creates a personal account to save their information and order history for future use.
- **Preconditions:**
  - The user is not logged in.
- **Main Flow:**
  - The user selects the "Register" or "Sign Up" option.
  - The system presents a registration form (e.g., name, email, password).
  - The user fills in the required details and submits the form.
  - The system validates the input (e.g., checks for a valid email format and ensures the username is unique).
  - The system creates the new customer account.
  - The system automatically logs the user in.
  - The system displays a welcome message.
- **Alternative Flows:**
  - **A1: Username Already Exists:** At step 4, if the system detects the email or username is already in use, it displays an error message and prompts the user to choose a different one, resuming at step 3.

---

05. Use Case: Browse Menu and View Item Details

- **Actor:** Customer
- **Description:** A customer browses the menu, views details for specific items including price, calories, and allergen information, and adds items to their cart.
- **Preconditions:**
  - At least one menu item is active and has available inventory.
- **Main Flow:**
  - The customer navigates to the main menu screen.
  - The system displays a list of available menu items.
  - The customer selects an item to view more details.
  - The system displays the item's full description, price, calorie count, and any associated allergen tags.
  - The customer selects an option to "Add to Order."
  - The system adds the item to the customer's cart and updates the subtotal.
- **Alternative Flows:**
  - **A1: Item is Out of Stock:** At step 2, if an item has zero inventory, it is either not displayed or is shown as "Unavailable," and the "Add to Order" option is disabled.

---

06. Use Case: Place Order as Registered User

- **Actor:** Customer
- **Description:** A logged-in customer reviews their cart, adds a tip, confirms, and submits their order for fulfillment.
- **Preconditions:**
    - The customer is logged in.
    - The customer has added at least one item to their cart.
    - The system tax rate has been configured.
- **Main Flow:**
    - The customer navigates to the checkout screen.
    - The system displays the order summary, including subtotal, calculated sales tax, and final total.
    - The system prompts for a tip (pre-set percentages or custom).
    - The customer selects a tip amount.
    - The system updates the final total.
    - The customer confirms and submits the order.
    - The system validates that all items still have sufficient inventory.
    - The system creates a permanent order record associated with the customer's account, places it in the fulfillment queue, and decrements inventory.
    - The system displays a confirmation screen with the order number and status.
- **Subflows:**
    - **S1: Add Custom Tip:** At step 4, the customer selects "custom," enters a specific amount, and the system updates the total accordingly.
- **Alternative Flows:**
    - **A1: Item Becomes Unavailable:** At step 7, if an item in the cart has run out of stock since it was added, the system displays an error, removes the item from the cart, and prompts the user to review their updated order, resuming at step 2.

---

07. Use Case: Place Order as Anonymous User

- **Actor:** Customer (not logged in)
- **Description:** A user without an account places an order by providing minimal identifying information.
- **Preconditions:**
    - The user is not logged in.
    - Items are available for sale.
- **Main Flow:**
    - The user adds items to a temporary cart.
    - The user proceeds to checkout.

    - The system displays the order summary with tax.
    - The system prompts for identifying information (e.g., a name for the order).
    - The user provides the required information.
    - The user selects a tip and confirms the order.
    - The system validates inventory, creates the order associated with the identifier, and places it in the fulfillment queue.
    - The system displays a confirmation screen with an order number for tracking.
- **Alternative Flows:**
    - **A1: Required Information Not Provided:** At step 5, if the user does not provide the required identifier, the system displays an error and prompts for the information again.

---

08. Use Case: View and Fulfill Incoming Order

- **Actor:** Staff
- **Description:** A staff member views the queue of pending orders, prepares an order, and marks it as fulfilled and ready for pickup.
- **Preconditions:**
    - The user is logged in with a "Staff" role.
    - At least one order is pending in the fulfillment queue.
- **Main Flow:**
    - The staff member navigates to the order fulfillment screen.
    - The system displays a list of all active orders.
    - The staff member selects an order to prepare.
    - The system displays the order details.
    - After preparing the items, the staff member selects the "Fulfill Order" button.
    - The system updates the order's status to "Fulfilled."
    - The system removes the order from the active queue and triggers a notification to the customer that the order is ready.
- **Alternative Flows:**
    - **A1: Order Canceled:** At step 3, if a customer cancels their order (assuming the feature exists), the order is removed from the queue. The system displays a notification, and the staff member returns to the active order list.

---

09. Use Case: Monitor Order Status

- **Actor:** Customer
- **Description:** A customer checks the real-time status of their placed order to know when it is ready.

- **Preconditions:**
  - The customer has successfully placed an order.
- **Main Flow:**
  - The customer views their active order display screen (e.g., order confirmation page or "My Orders" section if logged in).
  - The system displays the current status (e.g., "In Progress," "Received").
  - When a Staff member fulfills the order, the system automatically updates the status on the customer's display to "Ready for Pickup."
- **Alternative Flows:**
  - There are no significant alternative flows for this passive use case.

---

10. Use Case: View Order History

- **Actor:** Staff, Manager
- **Description:** An authorized user views a log of past completed orders for basic business analysis and record-keeping.
- **Preconditions:**
  - The user is logged in with Staff or Manager privileges.
  - At least one order has been fulfilled.
- **Main Flow:**
  - The user navigates to the "Order History" section.
  - The system displays a chronological list of all previously fulfilled orders.
  - For each order, the system shows key details such as the order ID, timestamp, items purchased, and total cost.
- **Subflows:**
  - **S1: Filter History:** At step 2, the user can filter the list by a date range to narrow down the results.
- **Alternative Flows:**
  - **A1: No Order History:** At step 2, if no orders have been completed yet, the system displays a message indicating "No order history found."

Part 2: Reflection Document

How We Decided What NOT To Do

For Project 1c1, our primary goal was to define a Minimum Viable Product (MVP). The core principle of an MVP is to build only the essential features required to test the product's primary hypothesis and gain validated user feedback with the least amount of development effort. Our hypothesis is: "Customers want a simple digital interface to order from the cafe, and staff can use the same system to fulfill those orders efficiently."

To achieve this, we systematically analyzed the ~30 use cases proposed in Proj1b1.pdf and categorized them into three groups:

1. **Core Functionality:** Features essential for the primary user journey (ordering and fulfillment).
2. **Optimizations & Enhancements:** Features that improve the user experience or business intelligence but are not critical for the system to function (e.g., personalized recommendations, customer dashboards).
3. **Advanced Compliance & Operations:** Features that address complex regulatory or operational needs that are beyond the scope of an initial launch (e.g., WIC eligibility, IRS fringe benefit reporting, detailed food safety logs).

We decided to **cut all features in the "Optimizations & Enhancements" and "Advanced Compliance & Operations" categories**. This included:

- **Advanced Tax/Health Features:** Applying WIC eligibility rules, generating IRS-compliant reports, and auto-generating nutrition labels from recipes were excluded. These require complex external integrations and logic that add significant development overhead without testing the core ordering hypothesis. A single tax rate and manual entry for nutrition data are sufficient for an MVP.
- **Customer Behavior and Personalization:** Personalized recommendations, healthy choice nudging, and customer behavior tracking were deferred. These are powerful features for a mature product but are classic "Version 2.0" items. The MVP does not need to optimize user choice—it first needs to prove it can *service* a user choice.
- **Complex Operational Tooling:** Digital food safety logs (temperature checks), ingredient traceability and recall systems, and staff training management were removed. While critical for a real-world restaurant, they are backend-heavy systems that do not impact the core customer-staff interaction we aim to validate. For an MVP, these processes can be handled manually outside the system.

This ruthless prioritization allows the team to focus entirely on building a stable, functional ordering and fulfillment loop, which is the heart of the WolfCafe system.

Negative Impacts and Disappointments for Stakeholders

This minimal approach will inevitably lead to disappointment for several stakeholders who have broader expectations:

- **Customers:** Health-conscious customers will be disappointed by the lack of advanced dietary filters, nutrition tracking dashboards, and suggestions for healthier alternatives. Regular customers will miss out on personalized recommendations that could speed up their ordering process. The experience will be functional but not tailored.

- **Administrators & Managers:** Their ability to manage the business will be severely limited. The lack of detailed tax reports, sales analytics, and nutrition-based sales reports means they cannot perform deep business analysis. They also bear the risk of non-compliance with advanced regulations (WIC, food safety) because the system does not support them.
- **Government / Regulators:** The MVP would fail audits related to specific programs like WIC or SNAP. It also lacks the robust digital logging features a health inspector might look for, increasing the compliance burden on the manager.
- **Staff:** While the core fulfillment loop is present, staff may miss quality-of-life features like receiving prominent allergy alerts on order tickets, which must now be handled through manual communication.

Changes Made to Appease Stakeholders

Recognizing the significant drawbacks of a purely transactional MVP, we made one key strategic change to appease the most critical stakeholder needs without succumbing to feature creep.

The Change: We decided to incorporate basic health information management into the MVP. Specifically, we modified Use Case 01: Manage Menu Item to include fields for **manual entry of calorie counts and allergen tags**. Correspondingly, Use Case 05: Browse Menu and View Item Details was designed to display this information to the customer.

The "Why":

1. **Appeases Customers & Regulators:** This simple addition directly addresses the primary concern of health-conscious **Customers** and helps the business meet basic FDA menu labeling regulations, a key concern for **Regulators**. It provides crucial information at the point of decision-making.
2. **Low Development Effort, High Value:** Unlike automated nutrition calculation or complex filtering logic, adding a few text fields to a menu item's data model and displaying them is a low-cost development task. The value it provides to the user experience and regulatory compliance is disproportionately high compared to the effort required.
3. **Mitigates a Key Risk:** It reduces the risk that customers with allergies or dietary needs would immediately reject the platform. This makes the MVP a more viable and safer product to test in a real-world context.

This change represents a calculated trade-off: a minor increase in scope to significantly improve the MVP's viability and address the most pressing concerns of its end-users and administrators.

Part 3: Prompt History Log

Here is a reconstructed log of prompts that could be used to generate the components of this project.

Prompt 1: Brainstorming MVP Scope

1. You are a product manager designing an MVP for a cafe ordering system called "WolfCafe". I have provided two documents: "Project-1a1.pdf" which outlines stakeholders and original use cases, and "Proj1b1.pdf" which contains about 30 expanded, advanced use cases generated by other LLMs.
2. Based on the principle of an MVP (maximum validated learning with minimum effort), analyze all the use cases from "Proj1b1.pdf" and decide which ones should be EXCLUDED from the initial product launch.
3. Provide a list of the excluded features and a brief justification for each, grouping them into logical categories (e.g., "Advanced Compliance," "Personalization," etc.). Your goal is to strip the system down to its absolute core: ordering and fulfillment.

Prompt 2: Generating the MVP Use Cases

1. Based on our MVP scope defined in the previous turn, now write 10 detailed use cases that describe the functionality of the WolfCafe MVP.
2. Each use case must be written in the following strict format, referencing the style and detail level found in "Project-1a1.pdf" (pages 4-8):
    a. - Actor: [Role]
    b. - Description: [A one-sentence summary]
    c. - Preconditions: [Bulleted list of what must be true to start]
    d. - Main Flow: [Numbered list of steps for the success path]
    e. - Subflows: (Optional) [Detailed steps for a minor, contained workflow]
    f. - Alternative Flows: [Bulleted list of exceptions or error paths, like "A1: [Condition]: [Outcome]"]
3. The 10 use cases should cover the complete, end-to-end core loop: menu/inventory management, user registration, browsing, placing an order (both logged-in and anonymous), order fulfillment, and viewing order history.

Prompt 3: Drafting the Reflection Document

1. Now, write a reflection document (maximum 2 pages) for this project. The document needs to cover three specific points:
    a. How did you decide what NOT to do? Explain the methodology for cutting features from the "Proj1b1.pdf" list, connecting it directly to the MVP philosophy.
    b. What negative impacts or disappointments could this MVP have for your stakeholders? Use the stakeholder list from "Project-1a1.pdf" (Customers,

Administrators, Staff, Regulators) and describe the specific functionality each one would miss in this minimal version.

c. What changes you made (and why) to the MVP to appease at least some of the stakeholders? Describe a specific, minimal feature that was added back into the MVP as a compromise. Justify why this particular feature provides high value for low effort and which stakeholders it appeases.

None

```
■ Now, write a reflection document (maximum 2 pages) for this project. The document needs to cover three specific points:
■
■ 1.  **How did you decide what NOT to do?** Explain the methodology for cutting features from the "Proj1b1.pdf" list, connecting it directly to the MVP philosophy.
■ 2.  **What negative impacts or disappointments could this MVP have for your stakeholders?** Use the stakeholder list from "Project-1a1.pdf" (Customers, Administrators, Staff, Regulators) and describe the specific functionality each one would miss in this minimal version.
■ 3.  **What changes you made (and why) to the MVP to appease at least some of the stakeholders?** Describe a specific, minimal feature that was added back into the MVP as a compromise. Justify why this particular feature provides high value for low effort and which stakeholders it appeases.
```

Perplexity Pro

Project 1c1 - MVP Problem Condensation

Overview

We have successfully completed Project 1c1 by transforming the 30 use cases from Project 1b1 into a streamlined Minimum Viable Product (MVP) with 10 essential use cases. This represents a 67% scope reduction while maintaining the core functionality needed to validate the food delivery business model.

Key Deliverables

1. New MVP Use Cases (≥10 use cases, 5+ pages)

Project 1c1 - MVP Use Cases

WolfCafe Food Delivery System - Minimal Viable Product

Executive Summary

This document contains 10 streamlined use cases for the WolfCafe MVP, reduced from the original 30 use cases in Project 1b1. The MVP focuses on core food delivery functionality to validate the business concept with minimal development effort.

MVP Use Cases

01. Use Case: Create Staff User

- Actor: Administrator
- Description: An Administrator creates a new user account with staff privileges to manage orders and menu items.
- Preconditions:
  - The user is logged in as an Administrator.
- Main Flow:
  - Administrator navigates to user management section.
  - Administrator selects "Create New User" option.
  - System presents basic user creation form (username, password, role).
  - Administrator enters details and selects "Staff" role.

    - Administrator submits form.
    - System validates required fields and username uniqueness.
    - System creates staff user account.
    - System displays success confirmation.
- Alternative Flows:
    - A1: Duplicate Username: At step 6, if username exists, system shows error and returns to step 3.

02. Use Case: Create Menu Item

- Actor: Staff
- Description: A staff member adds a new food item to the menu with basic details needed for ordering.
- Preconditions:
    - The user is logged in with Staff privileges.
- Main Flow:
    - Staff navigates to menu management section.
    - Staff selects "Add New Item" option.
    - System displays item creation form.
    - Staff enters item name, price, and basic description.
    - Staff submits form.
    - System validates price is positive number.
    - System creates menu item with zero initial inventory.
    - System displays success message.
- Alternative Flows:
    - A1: Invalid Price: At step 6, if price is not a positive number, system shows error and returns to step 4.

03. Use Case: Set Basic Tax Rate

- Actor: Administrator
- Description: Administrator sets a single system-wide tax rate for all orders (simplified from complex tax rules).
- Preconditions:
    - The user is logged in as Administrator.
- Main Flow:
    - Administrator navigates to system settings.
    - Administrator selects "Tax Configuration" option.
    - System displays current tax rate setting.
    - Administrator enters new tax rate percentage.
    - Administrator saves changes.
    - System validates input is between 0-50%.

  - System updates tax rate globally.
  - System displays confirmation message.
- Alternative Flows:
  - A1: Invalid Rate: At step 6, if rate is outside valid range, system shows error and returns to step 4.

04. Use Case: Add Inventory

- Actor: Staff
- Description: Staff increases available quantity for a menu item to enable customer orders.
- Preconditions:
  - User is logged in as Staff.
  - At least one menu item exists.
- Main Flow:
  - Staff navigates to inventory management.
  - Staff selects menu item from list.
  - System displays current inventory count.
  - Staff enters quantity to add.
  - Staff confirms addition.
  - System validates quantity is positive integer.
  - System updates inventory count.
  - System displays new total inventory.
- Subflows:
  - S1: Quick Add: At step 4, staff can use preset quantity buttons (5, 10, 25) for faster entry.
- Alternative Flows:
  - A1: Invalid Quantity: At step 6, if quantity is not positive integer, system shows error and returns to step 4.

05. Use Case: Place Order

- Actor: Customer (Logged In)
- Description: Customer selects items, reviews total with tax, and submits order for fulfillment.
- Preconditions:
  - Customer is logged into system.
  - Menu items available with inventory > 0.
  - Tax rate is configured.
- Main Flow:
  - Customer browses available menu items.
  - Customer selects item and adds to cart.
  - System updates cart with item and running total.

    - Customer repeats steps 2-3 for additional items.
    - Customer proceeds to checkout.
    - System calculates subtotal, tax, and final total.
    - System displays order summary.
    - Customer confirms order.
    - System validates inventory availability.
    - System creates order and adds to fulfillment queue.
    - System displays order confirmation with order number.
- Subflows:
    - S1: Modify Quantity: At step 3, customer can adjust item quantity in cart before checkout.
- Alternative Flows:
    - A1: Insufficient Inventory: At step 9, if any item is out of stock, system shows error and removes item from cart, returns to step 1.

06. Use Case: Place Simple Anonymous Order

- Actor: Anonymous Customer
- Description: User places order without creating account, providing only basic contact info for pickup.
- Preconditions:
    - User is not logged in.
    - Menu items available.
- Main Flow:
    - Anonymous user browses menu items.
    - User adds items to temporary cart.
    - System updates running total with tax.
    - User proceeds to checkout.
    - System prompts for pickup name and phone number.
    - User enters contact information.
    - User confirms order.
    - System validates contact fields are filled.
    - System creates order linked to contact info.
    - System displays order confirmation and pickup number.
- Alternative Flows:
    - A1: Missing Contact Info: At step 8, if required fields empty, system shows error and returns to step 6.

07. Use Case: Monitor Order Status

- Actor: Customer

- Description: Customer checks current status of their order to know when it's ready for pickup.
- Preconditions:
  - Customer has placed an order.
  - Order is not yet completed.
- Main Flow:
  - Customer navigates to order status page.
  - System displays current order with status indicator.
  - System shows estimated completion time.
  - Customer views real-time status updates.
- Subflows:
  - S1: Status Updates: System automatically refreshes status when staff updates order (preparing → ready).
- Alternative Flows:
  - A1: No Active Orders: At step 2, if no pending orders exist, system shows message "No active orders found."

08. Use Case: Fulfill Order

- Actor: Staff
- Description: Staff views pending orders, prepares food, and marks orders complete when ready for customer pickup.
- Preconditions:
  - User logged in as Staff.
  - At least one customer order pending.
- Main Flow:
  - Staff navigates to order fulfillment queue.
  - System displays list of pending orders sorted by time.
  - Staff selects order to prepare.
  - System displays order details and items.
  - Staff prepares order items.
  - Staff marks order as "Ready for Pickup".
  - System updates order status.
  - System removes order from pending queue.
  - System sends notification to customer display.
- Alternative Flows:
  - A1: Order Cancelled: At step 3, if order was cancelled by customer, system shows message and removes from queue.

09. Use Case: View Recent Orders

- Actor: Staff

- Description: Staff reviews recently completed orders for basic sales tracking and inventory planning.
- Preconditions:
    - User logged in as Staff.
    - At least one order has been completed.
- Main Flow:
    - Staff navigates to "Recent Orders" section.
    - System displays list of completed orders from last 24 hours.
    - System shows order totals and popular items.
    - Staff reviews information for planning purposes.
- Subflows:
    - S1: Order Details: At step 3, staff can click individual orders to see full details.
- Alternative Flows:
    - A1: No Recent Orders: At step 2, if no orders in timeframe, system displays "No orders in last 24 hours."

10. Use Case: Basic User Login

- Actor: Staff, Administrator, Customer
- Description: Users authenticate to access role-appropriate system features.
- Preconditions:
    - User account exists in system.
- Main Flow:
    - User navigates to login page.
    - User enters username and password.
    - User clicks "Login" button.
    - System validates credentials.
    - System determines user role (Customer/Staff/Admin).
    - System redirects to appropriate dashboard.
    - System displays welcome message with user role.
- Subflows:
    - S1: Remember Login: At step 3, user can check "Remember Me" for extended session.
- Alternative Flows:
    - A1: Invalid Credentials: At step 4, if credentials invalid, system shows error and returns to step 2.
    - A2: Account Locked: At step 4, if multiple failed attempts, system temporarily locks account and displays lockout message.

Summary

Total Use Cases: 10 (reduced from 30 in Project 1b1)
Reduction: 67% of original scope removed
Focus: Core ordering workflow, basic management, essential user access

This MVP validates the fundamental food delivery business model:

1. Menu Management - Staff can add items and inventory
2. Order Processing - Customers can place and track orders
3. Fulfillment - Staff can prepare and complete orders
4. Basic Administration - User management and tax configuration

Key Simplifications:

- Single tax rate instead of complex tax rules
- Basic inventory management without advanced tracking
- Simplified user roles (Customer/Staff/Admin only)
- Essential reporting instead of comprehensive analytics
- Core ordering without personalization or recommendations

Project 1c1 - Reflection Document

MVP Problem Condensation Analysis

How We Decided What NOT to Do

Our MVP decision-making process followed established minimum viable product principles, focusing on validating the core business hypothesis with minimal development effort. We applied a systematic filtering approach:

Decision Framework

1. Core Business Validation Test: Does this feature directly support the fundamental value proposition of food delivery?
2. User Journey Criticality: Is this feature essential for completing the primary user flows (order → prepare → fulfill)?
3. Development Complexity vs. Value: Does the implementation complexity justify the immediate business learning?
4. Stakeholder Impact Priority: Which stakeholders are most critical for initial business validation?

Exclusion Categories

Advanced Compliance Features (7 use cases removed - 23% of total)

- WIC eligibility rules, IRS reporting, tax exemptions, regulatory audits
- *Rationale*: While important for full deployment, these specialized compliance features don't validate the core food delivery business model and require significant regulatory expertise

Nutrition & Health Analytics (8 use cases removed - 27% of total)

- Nutrition labeling, health warnings, dietary preferences, nutrition dashboards
- *Rationale*: These are valuable differentiators but not essential for proving customers will order food online. Can be added post-validation

Advanced Personalization (5 use cases removed - 17% of total)

- AI recommendations, behavior tracking, customer analytics, health data exports
- *Rationale*: Complex machine learning features require substantial data and development time. MVP should focus on basic ordering functionality first

Complex Administrative Features (1 use case removed - 3% of total)

- User deletion, advanced reporting
- *Rationale*: Basic user creation is sufficient for MVP; complex admin features can wait until user base grows

Total Reduction: 21/30 use cases (70% scope reduction)

Negative Impacts and Stakeholder Disappointments

Administrators

Disappointments:

- Limited user management capabilities (no user deletion)
- No advanced reporting or compliance features
- Simplified tax system (single rate only vs. complex tax rules)

Impact: Administrators must handle edge cases manually, reducing operational efficiency and requiring more hands-on oversight.

Staff

Disappointments:

- Basic inventory management without advanced tracking
- No nutrition labeling or health compliance features

- Limited analytics (only recent orders vs. comprehensive reporting)

Impact: Staff cannot leverage data-driven insights for menu optimization or health-conscious positioning, potentially missing market opportunities.

Customers

Disappointments:

- No personalized recommendations or dietary filtering
- No nutrition information or health warnings
- Limited order customization options

Impact: Customers get a basic ordering experience without modern conveniences they may expect from competitors like DoorDash or Uber Eats.

Developers

Disappointments:

- Reduced scope may not showcase full technical capabilities
- Limited opportunity to implement advanced features like AI/ML
- Less comprehensive system architecture

Impact: MVP may appear too simplistic and not demonstrate the team's technical range for evaluation purposes.

Government/Regulators

Disappointments:

- No compliance reporting or audit trails
- Missing nutrition labeling required by some jurisdictions
- No accessibility features for disability compliance

Impact: System cannot be deployed in regulated environments without additional development.

Changes Made to Appease Stakeholders

While maintaining MVP focus, we made several adjustments to address stakeholder concerns:

For Administrators

- Retained Core User Management: Kept "Create Staff User" to ensure basic system administration
- Included Basic Tax Configuration: Provided single tax rate setting to handle legal requirements

- Added Recent Orders View: Gives basic operational oversight capability

For Staff

- Maintained Essential Workflow: Preserved complete order fulfillment process
- Kept Basic Analytics: "View Recent Orders" provides minimal business intelligence
- Included Inventory Management: Core inventory tracking for operational needs

For Customers

- Dual Order Methods: Kept both logged-in and anonymous ordering for flexibility
- Real-time Status: Maintained order tracking for modern user experience expectations
- Clean User Experience: Focused on streamlined, intuitive ordering process

For Developers

- Technical Variety: MVP still includes different user roles, database operations, and business logic
- Scalable Architecture: Design allows for easy addition of excluded features post-MVP
- Complete End-to-End System: Demonstrates full-stack development capabilities

Stakeholder Communication Strategy

We addressed stakeholder disappointments by:

1. Setting Clear Expectations: Emphasized MVP as *Phase 1* with explicit roadmap for excluded features
2. Demonstrating Learning Focus: Positioned excluded features as "validated before development" rather than "cut permanently"
3. Highlighting Speed-to-Market: Emphasized competitive advantage of faster launch over comprehensive features
4. Providing Future Vision: Created clear post-MVP roadmap showing how excluded features would be prioritized based on user feedback

Validation Strategy

Our MVP focuses on validating three core business hypotheses:

1. Demand Validation: Will customers use an online food ordering system?
2. Operational Validation: Can staff efficiently manage digital orders and inventory?
3. Business Model Validation: Does the order → payment → fulfillment cycle work profitably?

By excluding non-essential features, we can test these fundamental assumptions quickly and pivot based on real user behavior rather than assumptions. The 67% scope reduction allows us to deliver a working system in significantly less time while still providing genuine value to early adopters.

Success Metrics for MVP:

- Order completion rate > 80%
- Average order fulfillment time < 15 minutes
- Customer return rate > 25% within 30 days
- Staff adoption rate > 90% within first week

These metrics will guide decisions about which excluded features to implement in subsequent iterations based on actual user needs rather than hypothetical requirements.

Project 1c1 - Executive Summary

WolfCafe Food Delivery System - MVP Problem Condensation

Project Overview

Objective: Transform the 30 use cases from Project 1b1 into a streamlined MVP (Minimum Viable Product) with 10 essential use cases that validate the core food delivery business model.

Methodology: Applied established MVP principles and food delivery industry best practices to systematically reduce scope while maintaining core functionality.

Key Result: 67% scope reduction (from 30 to 10 use cases) focusing on essential business validation.

Scope Reduction Analysis

Use Cases EXCLUDED from MVP (21 total - 70% reduction):

Advanced Tax & Compliance (7 cases - 23%)

- WIC eligibility rules, IRS reporting, tax exemptions, regulatory compliance audits
- *Decision*: Specialized compliance features don't validate core business hypothesis

Nutrition & Health Features (8 cases - 27%)

- Nutrition labeling, health warnings, dietary filtering, nutrition dashboards
- *Decision*: Valuable differentiators but not essential for proving ordering demand

Advanced Personalization (5 cases - 17%)

- AI recommendations, behavior analytics, customer tracking, health data exports
- *Decision*: Complex ML features require substantial development time vs. learning value

Complex Administration (1 case - 3%)

- Advanced user deletion and reporting capabilities
- *Decision*: Basic admin functions sufficient for MVP testing

Use Cases RETAINED for MVP (10 total):

1. Create Staff User - Essential system administration
2. Create Menu Item - Core content management
3. Set Basic Tax Rate - Legal/pricing requirements
4. Add Inventory - Operational necessity
5. Place Order (Logged In) - Primary customer value
6. Place Simple Anonymous Order - Alternative customer path
7. Monitor Order Status - Core user experience
8. Fulfill Order - Essential staff workflow
9. View Recent Orders - Basic business intelligence
10. Basic User Login - System access control

MVP Validation Strategy

Core Business Hypotheses to Test:

1. Demand Validation: Will customers adopt online food ordering?
2. Operational Validation: Can staff efficiently manage digital orders?
3. Business Model Validation: Does order → payment → fulfillment cycle work profitably?

Success Metrics:

- Order completion rate > 80%
- Fulfillment time < 15 minutes
- Customer return rate > 25% in 30 days
- Staff adoption > 90% in first week

Stakeholder Impact Assessment

Primary Stakeholders (MVP Focus)

- Customers: Get core ordering and tracking functionality
- Staff: Retain essential order management and fulfillment tools
- Administrators: Basic system configuration and user management

Secondary Stakeholders (Post-MVP)

- Developers: Streamlined scope enables faster development and testing
- Regulators: Compliance features deferred to post-validation phase

---

Strategic Benefits

Speed to Market: 67% scope reduction accelerates development timeline significantly

Risk Mitigation: Validate core assumptions before investing in complex features

Resource Efficiency: Focus development effort on highest-impact functionality

Learning Maximization: Clear success metrics enable data-driven decisions for feature prioritization

Stakeholder Alignment: Core functionality satisfies primary user needs while managing expectations for advanced features

---

Implementation Recommendations

Phase 1 (MVP): Deploy 10 core use cases with success metric tracking

Phase 2 (Post-Validation): Add features based on user feedback priority:

- First: Basic nutrition information (customer-requested)
- Second: Advanced admin features (operational efficiency)
- Third: Personalization (customer retention)
- Fourth: Compliance features (regulatory requirements)

Technical Architecture: Design with modularity to enable rapid feature addition post-MVP

User Feedback Loop: Implement analytics and feedback collection in MVP to guide Phase 2 priorities

---

Document Index

1. [Project-1c1-MVP-Use-Cases.md] - Complete 10 MVP use cases with detailed specifications
2. [Project-1c1-Reflection.md] - Detailed analysis of exclusion decisions and stakeholder impacts
3. [Project-1c1-Prompt-History.md] - Documentation of LLM interaction methodology

4. [Project-1c1-Executive-Summary.md] - This overview document

Total Deliverable: 4 comprehensive documents covering MVP strategy, implementation, and methodology as required by Project 1c1 specifications.

Decision Rationale

What We Excluded (21 use cases - 70% reduction):

Advanced Tax & Compliance (7 cases)

- WIC eligibility, IRS reporting, tax exemptions, regulatory audits
- *Reason*: Specialized compliance features don't validate core business hypothesis

Nutrition & Health Features (8 cases)

- Nutrition labeling, health warnings, dietary preferences, dashboards
- *Reason*: Valuable differentiators but not essential for proving ordering demand

Advanced Personalization (5 cases)

- AI recommendations, behavior tracking, customer analytics
- *Reason*: Complex ML features require substantial development vs. learning value

Complex Administration (1 case)

- Advanced user management and reporting
- *Reason*: Basic admin functions sufficient for MVP testing

What We Retained (10 use cases):

The MVP focuses on the essential end-to-end user journey:

1. System Setup: User creation, menu management, tax configuration
2. Customer Experience: Order placement (logged-in and anonymous), order tracking
3. Staff Operations: Order fulfillment, inventory management, basic reporting
4. Core Infrastructure: User authentication and access control

Stakeholder Impact Management

While significantly reducing scope, we made strategic adjustments to address key stakeholder concerns:

- Administrators: Retained essential user management and system configuration
- Staff: Maintained complete order workflow and basic analytics
- Customers: Preserved core ordering experience with both account-based and anonymous options
- Developers: Ensured technical variety and scalable architecture for future expansion

Validation Strategy

The MVP tests three core business hypotheses:

1. Demand Validation: Will customers use online food ordering?
2. Operational Validation: Can staff efficiently manage digital orders?
3. Business Model Validation: Does the order→payment→fulfillment cycle work profitably?

Success Metrics:

- Order completion rate > 80%
- Average fulfillment time < 15 minutes
- Customer return rate > 25% within 30 days
- Staff adoption rate > 90% within first week

This systematic approach to MVP development follows industry best practices while ensuring we can validate the fundamental business concept with minimal development effort and maximum learning potential.

LLM Prompts

https://g.co/gemini/share/d000ddbb4859
https://g.co/gemini/share/b5c7ba5c5c63
https://g.co/gemini/share/1cd1ae730159
https://g.co/gemini/share/2441ea7b35e4
https://g.co/gemini/share/d24434342fc7
https://g.co/gemini/share/e4a40cce0077
https://g.co/gemini/share/6aee7ccc3805
https://g.co/gemini/share/b7ca359f37a8
https://www.perplexity.ai/search/project-1c1-due-monday-sept-15-YQcZO8EvTL2fP0pw2QuOfA#0