WolfCafe

CSE 510 Software Engineering : Project 1a1

Contributors : Janam Patel, Namit Patel, Pranshav Patel, Vivek Vanera

Question 1 : Create list of stakeholders (e.g., Admin, Staff, Customer, etc.). Who else ? How to find ? Brainstorm with an LLM.

Response :

Table 1 – List of Stakeholders, provided by different LLMs

Admin	Staff	Customers	Investors
TA (Managers)	Developers	Testers	IT/System Admin
Delivery Partners	Payment Processors	Food Safety Regulators	Ingredient Supplier

Question 2 : Identify stakeholder biases : List 5 ways needs of one stakeholder might clash/be irrelevant to another. How to find ? Brainstorm with an LLM.

Response : When analyzing stakeholder needs, certain priorities may clash or become irrelevant across groups. The following five examples highlight the most significant tensions :

1. **Customers vs. Staff (Customization vs. Efficiency)**
   Customer Need : Flexible order customization (extra toppings, substitutions, preferences).
   Staff Need : A streamlined workflow to fulfill orders quickly, especially during peak hours.
   Clash : Extensive customization slows service, increases error risk, and reduces staff efficiency.
2. **Business Owners vs. Regulators (Profit vs. Compliance)**
   Owner Need : Minimize costs and maximize profits, sometimes by reducing investment in compliance.
   Regulator Need : Strict adherence to tax rules, privacy laws, and data security regulations.
   Clash : Cutting compliance costs may result in legal violations, fines, or reputational damage.
3. **Admins vs. Developers (Control vs. Quality)**
   Admin Need : Rapid deployment of new features (e.g., tax updates, user management) with minimal delays.
   Developer Need : Adequate time for testing, documentation, and code quality to prevent technical debt.
   Clash : Pressure for speed may lead to unstable releases and long-term maintenance issues.
4. **Business Analytics vs. Customer Privacy (Data Collection vs. Trust)**
   Owner/Admin Need : Collect detailed customer data to optimize inventory, marketing, and personalization.
   Customer Need : Privacy and minimal data tracking.
   Clash : Excessive data collection risks customer distrust or avoidance of the system.
5. **Accessibility vs. Aesthetic Design (Inclusion vs. Branding)**
   Accessibility Need : Compliance with WCAG guidelines (high contrast, screen-reader support, keyboard navigation).
   Marketing/Owner Need : Visually appealing, trendy design that reinforces brand identity.
   Clash : Striking visual designs may reduce accessibility, while strict compliance may limit creative freedom.

Question 3 : Comment on prompt crafting : Compare zero-shot prompting to careful prompting.

Response : Below is comparison between Zero-Shot and Few-Shot Prompting (for Stakeholder Clashes)
Zero-Shot Prompting

— Responses were broader and often repetitive across LLMs (e.g., multiple mentions of "Staff vs. Customer : Speed vs. Customization").
— Some clashes were well-defined but lacked nuance (e.g., simple cost vs. compliance framing).
— Tended to default to common and predictable stakeholder conflicts, showing limited variety.

Few-Shot Prompting

— Responses became more context-specific and creative (e.g., "Legal vs. Marketing on privacy," "Investors vs. Payment Processors," "Admin vs. Accessibility Auditors").
— Less redundancy : greater diversity of stakeholders introduced (community managers, auditors, investors, IT staff).
— Improved depth : conflicts included both operational (e.g., staff workload) and strategic dimensions (e.g., investor expectations).

Key Difference : Few-shot prompting grounded the LLMs in richer stakeholder perspectives, reducing overlap and increasing originality, while zero-shot prompting produced safe but repetitive answers.

Differential Analysis of LLMs

1. **Grok-4**
   Strength : Balanced detail and structured explanations with clear "need vs clash" formatting.
   Weakness : Sometimes drifted into irrelevant cases (e.g., Suppliers vs. Admins).
2. **DeepSeek**
   Strength : Concise, high-level contrasts (e.g., Staff vs. Customer, Admin vs. Community).
   Weakness : Some answers lacked elaboration and nuance (stated conflicts but not fully explained).
3. **Gemini**
   Strength : Introduced fresh stakeholders not seen elsewhere (e.g., Quality Assurance Testers, Community Managers).
   Weakness : Denser phrasing ; occasionally complex without grounding in WolfCafe's context.
4. **GPT-5**
   Strength : Strong focus on operational and regulatory aspects (e.g., Admin vs. Accessibility Auditor, Customer vs. Tax Authority).
   Weakness : Repetition of common conflicts (e.g., Staff vs. Customer customization) seen across both prompt types.

Question 4 : Use Cases for WolfCafe System

The following use cases describe the main interactions between customers, staff, and administrators in the WolfCafe system. Each use case is structured with preconditions, main flow, subflows/extensions, and alternative/error flows.

UC1 : User Login (All Roles)

Preconditions

User has a valid account (Customer, Staff, or Admin). System is online.

Main Flow

1. User navigates to login page.
2. User enters username/email and password [Enter Credentials].

3. System verifies credentials [Authenticate].
4. System redirects user to role-specific dashboard [Redirect].

Subflows

[Enter Credentials] → User provides login details. [Authenticate] → System validates password hash. [Redirect] → Customer → ordering page, Staff → order queue, Admin → admin panel.

Alternative Flows

— Invalid Credentials → Error message and retry option.
— Account Locked → Notify user and require admin unlock.
— System Offline → Display maintenance page.

UC2 : Customer Places Order

Preconditions

Customer is logged in. Items exist in catalog with sufficient inventory.

Main Flow

1. Customer browses menu [Browse Menu].
2. Customer selects items and adds to cart [Add Items].
3. Customer reviews order with subtotal, sales tax, and tip [Review Cart].
4. Customer proceeds to checkout and submits order [Submit Order].
5. System deducts inventory and creates order record.
6. System notifies customer of order ID and estimated pickup.

Subflows

[Browse Menu] → Display items with names, prices, descriptions. [Add Items] → Update cart with quantities. [Review Cart] → Apply tax (e.g., 2.0%) and tip (15%, 20%, 25%, or custom).

Alternative Flows

— Insufficient Inventory → Prevent checkout and suggest alternatives.
— Payment Failure → Cancel order and request new payment method.
— Empty Cart → Disable checkout and show warning.

UC3 : Staff Views Orders

Preconditions

Staff is logged in. At least one customer order exists.

Main Flow

1. Staff opens dashboard [Load Orders].
2. System displays queue of pending/active orders.
3. Staff selects order to view details [View Order Details].

Subflows

[Load Orders] → Retrieve all orders with status (PLACED, IN_PREP, READY). [View Order Details] → Show items, customer notes, time placed.

Alternative Flows

— No Orders → Display "No pending orders."
— DB Error → Retry with error message.

UC4 : Staff Prepares and Fulfills Order

Preconditions

Order exists with status = PLACED.

Main Flow

1. Staff accepts an order [Accept Order].
2. System reserves inventory and updates status to IN_PREP.
3. Staff completes preparation [Complete Prep].
4. Staff marks order fulfilled [Mark Fulfilled].
5. System updates status to READY and notifies customer.

Subflows

[Accept Order] → Lock ingredient quantities. [Complete Prep] → Confirm all items prepared. [Mark Fulfilled] → Update DB and push notification.

Alternative Flows

— Insufficient Inventory → Reject order acceptance.
— Order Cancelled by Customer → Remove from queue.
— Prep Issue → Flag admin for resolution.

UC5 : Customer Tracks Order

Preconditions

Customer has placed an order. Order exists with status = PLACED, IN_PREP, or READY.

Main Flow

1. Customer logs into dashboard [Access Orders].
2. Customer selects order [Select Order].
3. System displays current status [Show Status].

Subflows

[Access Orders] → Display customer's active/past orders. [Show Status] → Poll DB and update timeline.

Alternative Flows

— No Orders → Display “No active orders.”
— System Error → Show last known status with error note.

UC6 : Customer Picks Up Order

Preconditions

Order status = READY. Customer has valid order ID.

Main Flow

1. Customer receives notification [Receive Notification].
2. Customer arrives at pickup counter.
3. Customer presents order ID [Present ID].
4. Staff verifies and hands over order [Handover].
5. System updates status to PICKED_UP.

Subflows

[Receive Notification] → Real-time update via app/email. [Handover] → Staff cross-checks ID with system record.

Alternative Flows

— Delayed Notification → Customer checks manually.
— Wrong/Expired ID → Deny pickup until resolved.
— Pickup Timeout → Order discarded, inventory adjusted.

UC7 : Admin Manages Users

Preconditions

Admin is logged in. System user management enabled.

Main Flow

1. Admin accesses user management panel [Access Panel].
2. Admin creates, edits, or deletes user accounts [Manage User].
3. System validates and updates database.

Subflows

[Manage User] → Enter details (name, role, password). [Validate] → Ensure uniqueness and proper format.

Alternative Flows

— Duplicate Username → Prompt for another.
— Invalid Role → Restrict to staff/customer only.
— Active Orders → Prevent deletion until resolved.

UC8 : Staff Manages Inventory

Preconditions

Staff is logged in. Inventory items exist.

Main Flow

1. Staff accesses inventory management [Access Inventory].
2. Staff selects item and updates stock [Update Stock].
3. System saves update and displays new stock level.

Subflows

[Access Inventory] → Display items with stock counts. [Update Stock] → Add or remove quantities.

Alternative Flows

— Item Not Found → Suggest creating item.
— Invalid Quantity → Reject non-numeric/negative values.
— Overstock Limit → Warn staff and cap entry.

UC9 : Staff Creates New Item/Recipe

Preconditions

Staff is logged in. Item creation enabled.

Main Flow

1. Staff navigates to item creation page [Access Creation].
2. Staff enters item details [Enter Details].
3. Staff adds ingredients if applicable [Add Ingredients].
4. Staff saves item [Save Item].

Subflows

[Enter Details] → Name, price, description. [Add Ingredients] → Link to inventory with quantities.

Alternative Flows

— Duplicate Item → Prompt edit instead.
— Missing Fields → Highlight and request completion.
— Ingredient Shortage → Warn but allow save.

UC10 : Admin Sets Sales Tax Rate

Preconditions

Admin is logged in. System has default tax rate.

Main Flow

1. Admin accesses system settings [Access Settings].
2. Admin locates tax rate option [Find Option].
3. Admin enters and applies new rate [Enter Rate].
4. System updates tax rate for future orders.

Subflows

[Validate Rate] → Check positive numeric input.

Alternative Flows

— Invalid Rate → Reject negative/non-numeric values.
— No Change → Confirm and exit.
— Active Orders → Apply only to new orders.

Thank you.