# SE Project 1a1

Group 16: Dhruva Kamble, Rishitha Ramesh, Rujuta Budke

## WolfCafe Stakeholders

### Primary Stakeholders

- **Customer** – orders food/drinks, pays (optionally tips), and picks up orders.
- **Staff (Barista/Kitchen)** – prepares orders, marks them ready, updates inventory counts.
- **Store Manager / Admin** – manages menu/recipes, prices, sales tax, discounts, user roles/permissions; oversees refunds/voids and overall inventory.

### Secondary Stakeholders

- **Payment Provider (External System)** – processes customer payments, refunds, and settlements.
- **University IT / Security Team** – manages authentication (SSO), ensures data privacy/security, and reviews for compliance.
- **Finance / Accounting (Dining Services)** – handles reconciliation, payouts, and tax remittance.
- **Regulators / Tax Authority** – enforce sales tax, accessibility, and data privacy compliance.
- **Accessibility Office (University or Legal)** – reviews WCAG/ADA compliance and accommodations.
- **Suppliers / Vendors** *(if in scope)* – provide ingredients, track purchases, and restocking.

### Additional Stakeholders

- **Developers / Maintainers** – implement new features, fix bugs, and maintain the system over time.
- **Auditors / QA Testers** – verify reliability, accessibility, compliance, and correctness.
- **Accessibility Users (Cross-Cutting Segment)** – any customer, staff, or admin requiring screen reader support, keyboard navigation, or high-contrast UI.

# Stakeholder Biases and Their Impact

## Customer

- **Bias**: Wants fast, convenient, and customizable ordering with minimal friction.
- **Impact on Others:**
  - Conflicts with Admin, who prioritizes security and permissions.
  - Increases workload for Staff, who prefer standardized processes.
  - Clashes with Payment Provider, who enforces extra verification.
  - Frustrated by Regulators, whose compliance rules (privacy, taxes) add barriers.

## Staff

- **Bias**: Wants efficient workflows and manageable order complexity.
- **Impact on Others:**
  - Conflicts with Customer customization demands.
  - Restricted by Admin permissions on inventory and recipes.
  - Dependent on Developers to avoid overly complex tools.

## Admin

- **Bias**: Wants full control, compliance, and secure system operation.
- **Impact on Others**:
  - Adds friction for Customers with mandatory logins and rules.
  - Imposes stricter processes on Staff.
  - Aligns with Regulators but may slow down Developers.

## Accessibility Users

- **Bias**: Require accessible UI (screen reader support, keyboard navigation, high contrast).
- **Impact on Others:**
  - Pushes Developers to prioritize accessibility over flashy design.
  - May affect Staff workflows if accessibility accommodations change UI flow.
  - Forces Admin/IT to enforce compliance policies.

## Regulators / Tax Authority

- **Bias**: Enforce compliance with laws (sales tax, privacy, accessibility).
- **Impact on Others:**
  - Adds friction for Customers, who want simple ordering.
  - Increases workload for Admin, who must configure tax rules.
  - Forces Developers to add tax calculations, privacy prompts, and logging.

# Zero Shot VS Careful Prompting

| Aspect | Zero-Shot Prompting | Careful Prompting |
|---|---|---|
| **Stakeholders** | Generic roles: Customers, Staff, Admin/Manager, Delivery Drivers, Developers. | Aligned with WolfCafe roles: Admin, Staff, Customer, plus accessibility users and IT/security. |
| **Biases** | General goals (speed, efficiency, control). | Anchored to WolfCafe tasks (Admins set tax rates, Staff manage inventory, Customers choose tips). |
| **Conflicts** | Broad (speed vs. security, compliance vs. convenience). | WolfCafe-specific (anonymous ordering vs. login, NC 2% tax enforcement, accessibility vs. developer effort). |
| **Use Cases** | Generic: place order, pay, deliver, manage menu. | Project-specific: Customer Orders, Fulfill Orders, Tax Rate Setting, Privacy Policy. |
| **Non - Functional** | General: secure, reliable, fast. | Specific: 70% backend test coverage, accessibility, privacy policy, human flourishing. |

**Reflection:**
Zero-shot prompting gave quick but vague answers that often introduced irrelevant elements (like delivery drivers). Careful prompting, guided by the WolfCafe documents, produced refined outputs directly tied to the project's scope and done criteria. This mirrors requirements engineering: early requirements are broad and ambiguous, but refinement with context yields precise, testable, and actionable results.

# Use-cases

## UC1 — Customer Places an Order

**Preconditions**

- Customer is authenticated (or guest checkout is enabled, if supported).
- Menu items exist and at least one is in stock.
- Sales tax rate is configured.

**Main Flow**

1. Customer browses the menu.
2. Customer selects one or more items and quantities [Select Item].

3. System shows cart with subtotal.
4. Customer proceeds to checkout.
5. System calculates tax [Apply Tax].
6. Customer chooses a tip (15%, 20%, 25%, or custom) [Add Tip].
7. Customer confirms payment details and submits [Process Payment].
8. System records order and returns order ID & “pending” status.

**Subflows**

- [Select Item] Add, change qty, or remove items.
- [Apply Tax] Apply current configured rate to subtotal.
- [Add Tip] Apply selected percentage or validated custom amount.
- [Process Payment] Send charge to payment provider; on success, mark order “paid”.

**Alternative Flows**

- [Item Out of Stock] System prevents add; suggest alternatives.
- [Payment Failed] Show failure reason; allow retry or cancel.
- [Tax Not Configured] Default to 0% tax (flag for admin) or block checkout (policy decision).

## UC2 — Customer Receives Notification & Picks Up Order

**Preconditions**

- A paid order exists.
- Staff has marked the order “fulfilled/ready”.

**Main Flow**

1. System updates customer-facing display/notifications [Notify Ready].
2. Customer arrives at pickup station.
3. Staff verifies the order ID or customer identity [Verify Pickup].
4. Staff marks order “picked up”; system closes the order.

**Subflows**

- [Notify Ready] In-app banner/queue screen updates; optional email/SMS/push.
- [Verify Pickup] Match order ID, name, or QR; log timestamp.

**Alternative Flows**

- [Wrong Customer] Verification fails; refuse release; leave order “ready”.
- [Stale Pickup] Order not collected within window; mark “expired” and follow policy (discard/refund/hold).

## UC3 — Customer Cancels an Order

**Preconditions**

- Customer has an order in “pending” or “paid but not fulfilled” status.
- Cancellation window/policy applies.

**Main Flow**

1. Customer opens order details.
2. Customer selects “Cancel Order”.
3. System checks status & policy window.
4. If eligible, system cancels order and initiates refund (if paid) [Refund].

**Subflows**

- [Refund] Reverse transaction via payment provider; update order to “canceled”.

**Alternative Flows**

- [Too Late] Order already fulfilled or outside window → show policy; no cancel allowed.
- [Partial Cancel] Some items unavailable; offer partial cancel/refund.

## UC4 — Staff Fulfills an Order

**Preconditions**

- Staff is authenticated with fulfill permissions.
- One or more paid orders exist in “pending”.

**Main Flow**

1. Staff opens “Pending Orders” queue [View Queue].
2. Staff selects an order.
3. Staff prepares items.
4. Staff marks order “fulfilled/ready” [Update Status].
5. System notifies the customer (see UC2 subflow).

**Subflows**

- [View Queue] Filter/sort by time, priority, or item type.
- [Update Status] Transition: pending → in-progress → fulfilled.

**Alternative Flows**

- [Out of Ingredients] Mark item(s) unavailable, notify customer, propose substitution or partial refund.
- [Accidental Fulfillment] Allow revert to “pending” with audit log.

## UC5 — Staff Manages Inventory

**Preconditions**

- Staff is authenticated with inventory permissions.

**Main Flow**

1. Staff opens Inventory.
2. Staff searches or selects an item [Find Item].
3. Staff adjusts quantity or restocks [Adjust Stock].
4. System updates stock levels and logs change.

**Subflows**

- [Find Item] Search by name/ID; view current levels and thresholds.
- [Adjust Stock] Increment/decrement; optional note (delivery, shrinkage).

**Alternative Flows**

- [Invalid Quantity] Negative or non-numeric rejected with message.
- [Item Not Found] Create new inventory item (if allowed) or request admin action.

## UC6 — Staff Creates or Edits Recipes/Menu Items

**Preconditions**

- Staff is authenticated with menu-management permissions.

**Main Flow**

1. Staff opens Menu Management.
2. Staff creates a new item or edits an existing one [Edit Form].
3. Staff sets name, price, optional ingredients and availability.
4. System validates and saves changes; menu updates.

**Subflows**

- [Edit Form] Structured fields (name, price, description, category, availability, optional ingredient list).

**Alternative Flows**

- [Duplicate Name] Reject; prompt to choose a unique name.
- [Missing Required Fields] Highlight errors; prevent save.
- [Price Out of Range] Reject with guidance.

## UC7 — Admin Manages User Accounts

**Preconditions**

- Admin is authenticated with user-admin permissions.

**Main Flow**

1. Admin opens User Management.
2. Admin creates, edits, or deletes a user [Modify User].
3. System validates role & fields; saves changes; logs action.

**Subflows**

- [Modify User] For create: set username, role (staff/customer), temp password; for edit: update profile; for delete: confirm and deactivate/delete per policy.

**Alternative Flows**

- [Unauthorized Role Change] Prevent elevating to Admin (unless current user qualifies); log attempt.
- [Invalid Input] Reject malformed emails, weak passwords, or empty fields.
- [Delete Constraints] Prevent deleting the last Admin; require transfer of ownership.

## UC8 — Admin Updates Sales Tax Rate

**Preconditions**

- Admin is authenticated.
- System supports configurable tax.

**Main Flow**

1. Admin opens Settings → Tax.
2. Admin enters a new rate (%) [Validate Rate].
3. System saves new rate and applies to future calculations.

**Subflows**

- [Validate Rate] Accept numeric values within allowed bounds (e.g., 0–20%).

**Alternative Flows**

- [Invalid Entry] Reject non-numeric/negative/out-of-range.
- [Insufficient Permission] Block and log attempt.

## UC9 — System Ensures Accessibility Compliance

**Preconditions**

- Frontend uses accessible components; accessibility checks enabled.

**Main Flow**

1. System exposes semantic labels/roles for interactive elements [SR Labels].
2. System enforces logical tab order and visible focus states [Keyboard Nav].
3. System maintains minimum color contrast for text/icons [Contrast].
4. System surfaces alt text for images and ARIA attributes where appropriate.

**Subflows**

- [SR Labels] aria-label/aria-labelledby associated to controls.
- [Keyboard Nav] All actions reachable by keyboard; no keyboard traps.
- [Contrast] Check against WCAG thresholds; apply tokens/themes.

**Alternative Flows**

- [Missing Labels Detected] Log accessibility violation; show dev warning; create issue.
- [Theme Breaks Contrast] Fall back to safe palette; flag to UI team.

## UC10 — System Maintains Privacy & Compliance

**Preconditions**

- Privacy Policy content exists.
- Logging/auditing configured.

**Main Flow**

1. System displays Privacy Policy link and obtains consent where required [Consent].
2. System stores only necessary personal data with least privilege [Data Min].
3. System logs admin/security-relevant actions for audit [Audit Log].
4. System applies configured tax policy to orders [Tax Apply].

**Subflows**

- [Consent] First-run or pre-checkout notice with link to policy.
- [Data Min] Avoid storing sensitive data unnecessarily; mask tokens.
- [Audit Log] Immutable logs for user/role changes, tax changes, inventory adjustments.
- [Tax Apply] Use the current configured rate in order totals.

**Alternative Flows**

- [Consent Declined] Provide limited experience or block checkout per policy.
- [Logging Failure] Alert ops/admin; queue logs for retry; fail-safe restrictions until resolved.