List of Stakeholders

Primary Stakeholders:

- Admin → manages users (staff/customers) and tax rates.
- Staff → fulfills orders, manages inventory, and creates items/recipes.
- Customer → places orders, makes payments, and picks up the item/recipe.

Secondary Stakeholders:

- Developers → build and maintain WolfCafe.
- Testers/QA Engineers → ensure system quality.
- Teaching Staff → act as project managers/product owners.
- University → providing the project context and is responsible for security/privacy compliance.
- Accessibility Users → rely on accessible frontend design.
- Payment Providers (ex, credit card system, digital wallet integration) → process payments and tips.

External Stakeholders:

- Vendors/Suppliers → provide ingredients or items/recipes to the cafe.
- System Administrators → maintain server, security, and backups.
- Regulatory/Compliance Bodies (ex, state tax authority) → enforce tax collection, privacy laws, accessibility, and payment security standards.

How to find stakeholders?

- Start with system roles (Admin, Staff, Customer).
- Expand to anyone who interacts indirectly (developers, testers, managers).
- Consider external systems or organizations (payment providers, regulators).
- Think of future impact (maintainers, auditors, accessibility users).

Stakeholder Biases

1. Admin vs. Customer

- Admins may enforce stricter security/privacy requirements (ex, complex logins, mandatory accounts).
- Customers may want quick, anonymous, frictionless ordering.

2. Staff vs. Customer

- Staff may prefer batching orders for efficiency.
- Customers want their individual orders fulfilled quickly, even if it's inefficient.

3. Developer vs. Admin/Staff/Customer

- Developers may want a clean, maintainable system with slower rollouts.
- Admins and customers may push for quick new features or bug fixes, even if quality suffers.

4. Accessibility Users vs. Developers

- Accessibility users require careful UI design (ex, ARIA labels, tab order, screen reader support).
- Developers may want to build a "flashy" interface quickly, neglecting accessibility.

5. Product Owners (University) vs. Customers

- Product owners (University) may emphasize stricter goals (ex, automated testing, strict processes).
- Customers care only about usability and convenience, not whether developers followed best practices.

<u>Prompt Crafting: Zero-Shot vs Careful Prompting</u>

Zero-Shot Prompting:

- Definition: Asking an LLM to perform a task without providing examples, detailed structure, or constraints.
- Pros: Fast, simple, requires little effort. Useful for brainstorming or when flexibility is desired.
- Cons: Results can be vague, inconsistent, or off-target since the model has to guess the intended style or depth.
- Ex: User asks the LLM something like "List WolfCafe stakeholders."

Careful Prompting:

- Definition: Giving the LLM clear instructions, examples, formatting requirements, and constraints.
- Pros: Produces more accurate, relevant, and structured outputs. Reduces ambiguity and aligns the response with user needs.
- Cons: Takes more effort from the user to design and refine prompts. May reduce creativity if overly constrained.
- Ex: User asks the LLM something like "List WolfCafe stakeholders in the format: Role → one-line description of responsibility (ex, Customer → places orders). Include both primary and secondary stakeholders."

<u>Use Cases for WolfCafe</u>

Use Case U1: Manage Loyalty Points

Preconditions:

- User is logged in as Customer.
- User has at least one prior completed order.

Main Flow:

1. Customer navigates to “Loyalty Points” section.
2. System displays current points balance [No Points Available] [View Redemption History].
3. Customer chooses a discount based on the amount of points they have.
4. Customer selects “Redeem Points” [Invalid Redemption].
5. System applies the discount to customer’s next order and confirms success.
6. Customer returns to main menu.

Subflows:

- [View Redemption History] Customer may view previous redemptions.

Alternative Flows:

- [No Points Available] If the customer has no loyalty points, the system shows “0 points” and disables redemption.
- [Invalid Redemption] If a discount requires more points than available, an error is shown, and the user is sent back to edit the form.

Use Case U2: Request Refund

Preconditions:

- User is logged in as Customer.
- Customer has a fulfilled order in history.

Main Flow:

1. Customer selects an order from “Order History”.
2. System displays order details.
3. Customer selects “Request Refund” [Cannot Refund].
4. Customer provides refund reason and submits [Invalid Reason].
5. System logs request and notifies staff for approval.
6. System confirms refund request submitted.

Alternative Flows:

- [Cannot Refund] If the order is older than the refund window (e.g., 7 days), the system shows an error.
- [Invalid Reason] If the reason field is empty, an error is shown, and the user is sent back to edit the form.

Use Case U3: Customize Item/Recipe

Preconditions:

- User is logged in as Customer.
- Customer is placing an order.

Main Flow:

1. Customer selects an item/recipe to order.
2. System displays customization options (size, milk type, sugar level, toppings).
3. Customer enters customization options amount [Invalid Amount].
4. Customer confirms item/recipe customization [Invalid Customization] [Save Customization].
5. System adds customized item/recipe to cart and updates total.
6. Customer returns to the ordering menu.

Subflows:

- [Save Customization] Customer may save item customization as a preset for future use.

Alternative Flows:

- [Invalid Customization] If the selected option is unavailable (ex, almond milk is out of stock), an error is shown, and the user is sent back to edit the form.
- [Invalid Amount] If input is not a positive integer, the system shows an error, and the user is sent back to edit the form.

Use Case U4: Daily Sales Report

Preconditions:

- User is logged in as Admin.
- At least one order has been fulfilled that day.

Main Flow:

1. Admin selects “Reports” from the menu.
2. System displays available report types.
3. Admin selects “Daily Sales Report” [No Sales].
4. System generates and displays a report with orders, revenue, tips, and taxes collected [Filter Report].
5. Admin returns to the main menu.

Subflows:

- [Filter Report] Admin may filter by time period, staff member, or payment method.

Alternative Flows:

- [No Sales] If no orders are fulfilled that day, the system shows “No data available.”

Use Case U5: Low Inventory Alerts

Preconditions:

- User is logged in as Staff or Admin.
- At least one inventory item/recipe is below threshold.

Main Flow:

1. System checks inventory levels periodically.
2. When the ingredients drop below the threshold, an alert is triggered.
3. Staff/Admin receives notification on the dashboard.
4. Staff/Admin dismisses or acknowledges alert [Dismiss Alert].

Alternative Flows:

- [Dismiss Alert] If the Staff/Admin dismisses the alert, the system removes it from the dashboard but logs an acknowledgment.

Use Case U6: Schedule Order Pickup

Preconditions:

- User is logged in as Customer.
- Customer has items/recipes in cart.

Main Flow:

1. Customer selects “Schedule Pickup” option.
2. System displays available time slots.
3. Customer chooses a pickup time [Invalid Time].
4. System reserves a slot and confirms scheduling.
5. Customer completes payment.
6. System associates the order with the scheduled pickup.

Alternative Flows:

- [Invalid Time] If the chosen slot is unavailable or past, the system shows an error, and the user is sent back to edit the form.

Use Case U7: Staff Shift Management

Preconditions:

- User is logged in as Admin.
- Staff accounts exist in the system.

Main Flow:

1. Admin selects “Manage Shifts”.
2. System displays staff lists and current shifts.
3. Admin assigns or edits staff shifts [Invalid Assignment].
4. System updates schedule and notifies staff.
5. Confirmation message displayed.

Alternative Flows:

- [Invalid Assignment] If an overlapping or invalid shift is assigned, an error is displayed, and the admin is sent back to edit the form.

Use Case U8: Favorite Orders

Preconditions:

- User is logged in as Customer.
- Customer has placed at least one order.

Main Flow:

1. Customer selects “Favorite Orders”.
2. System displays saved favorite orders [No Favorites].
3. Customer selects a favorite order to reorder [Item/Recipe Unavailable].
4. System adds items/recipes from favorites to the cart.
5. Customer proceeds with checkout.

Alternative Flows:

- [No Favorites] If the customer has not saved favorites, the system shows an empty state.
- [Item/Recipe Unavailable] If an item/recipe in favorites is unavailable, the system suggests alternatives.

Use Case U9: Feedback Submission

Preconditions:

- User is logged in as Customer.
- Customer has at least one fulfilled order.

Main Flow:

1. Customer navigates to the “Feedback” section.
2. System displays a form with rating and comment fields.
3. Customer enters rating/comments [Anonymous Feedback].
4. Customer submits feedback [Invalid Feedback].
5. System saves feedback and notifies staff.
6. Confirmation displayed.

Subflows:

- [Anonymous Feedback] Customer may choose to submit feedback anonymously.

Alternative Flows:

- [Invalid Feedback] If required fields are empty, an error is displayed, and the user is sent back to edit the form.

Use Case U10: Staff Performance Report

Preconditions:

- User is logged in as Admin.
- At least one member of staff has fulfilled orders.

Main Flow:

1. Admin selects “Reports”.
2. System displays available reports.
3. Admin selects “Staff Performance Report” [No Data].
4. System generates and displays a report with fulfilled orders, fulfillment times, and ratings.
5. Admin returns to the main menu.

Alternative Flows:

- [No Data] If no staff data exists, the system shows “No performance data available.”