CSC 510 Project 1a1

Tiva Rocco, Rujuta Palimkar, Aayushi Masurekar,Anusha Upadhyay

Group 5

Stakeholders

A. Direct Stakeholders :

1. Customers – orders, pays, provides feedback
2. Admin - helps when new staff joins, keeps separation of concerns
3. Waiters/Cashier - takes orders, serves food, interacts with customers.
4. Owner - oversees operations, tracks sales, manages inventory.
5. Barista - prepares coffee/drinks based on orders.
6. Developers/Engineers - build and maintain the system.
7. Product Owners/Managers - oversees development team, defines requirements, priorities
8. Customers with Special Needs – accessibility requirements
9. Cleaning Staff – indirectly impacted by scheduling and shift management.

B. Indirect Stakeholders :

1. Suppliers – coffee bean vendors, dairy suppliers, sugar suppliers
2. Transport providers - Shipping raw materials
3. Accountant/ Finance team - uses reports for revenue, expenses, and taxes.
4. Maintenance/ support – IT support for technical issues
5. Marketing team – uses data for promotions and loyalty programs.
6. Delivery companies – uber, doordash
7. Investors / Shareholders – interested in financial performance and growth.
8. Regulators / Inspectors – health, safety, and food compliance authorities.
9. Payment Gateway Providers (e.g., Visa, Mastercard, PayPal, UPI, Square) – integrated into transactions.
10. Banking Partners – handle settlements of customer payments and business loans.

Stakeholder Biases

Stakeholders	Need / Goal	Clash / Irrelevance	Example Scenario
Admin vs Customer	Admin enforces tax policies, strict roles.	Customers may see policies as unfair/irrelevant to daily use.	Admin enforces 2% NC sales tax → Customer compares with cheaper cash purchase.

Staff vs Accessibility Users	Staff want simple UIs (fast to train on, easy to click).	Accessible UIs (tooltips, tab order) may add extra design complexity staff don't value.	Staff only use mouse clicks → visually impaired customers need keyboard/tab support.
Customers vs Vendors/Suppliers	Customers want wide menu variety.	Suppliers push certain items based on inventory contracts, not demand.	Customers want bubble tea → Supplier contract only supports coffee products.
Staff vs Admin	Staff want autonomy in handling orders.	Admin imposes rigid permission boundaries.	Staff can't comp/refund an item without Admin approval.
Customers vs Wider Community (Non-users)	Customers benefit from streamlined mobile pickup.	Non-users (walk-ins) may face longer lines as staff prioritize app orders.	Cafeteria customers without WolfCafe wait longer → negative perception.

How to find the clashes?
Method: Using an LLM to Brainstorm Stakeholder Biases & Conflicts

1. Map Core Stakeholders First
Prompt:
"List all direct and indirect stakeholders for a food-ordering app like WolfCafe, including users, maintainers, institutions, and non-users who are affected."
LLM will surface a broad set beyond just Admin/Staff/Customer (e.g., IT, future developers, accessibility users).

2. Elicit Stakeholder Needs
Prompt:
"For each stakeholder, list their primary goals and needs in the system."
Example outputs:
Admin → control, compliance, permissions.
Staff → fast workflows, manageable orders.
Customer → convenience, choice, privacy.
IT → security, uptime, audit logs.

3. Cross-Compare Needs
Prompt:
"Identify how the goals of one stakeholder might clash with or be irrelevant to another's goals."
Example outputs:
Customer's desire for anonymity ↔ IT's demand for authentication.

Instructor's grading rubric ↔ future maintainer's need for readable code.

4. Generate "What If" Scenarios
Prompt:
"Imagine a scenario where staff need faster order fulfillment but customers demand more customization. What's the conflict? Who wins?"
This lets the LLM play out realistic stories of clashes.

5. Rank Conflicts
Prompt:
"Rank these conflicts by short-term grading impact vs long-term system sustainability."Helps distinguish classroom-relevant vs real-world-relevant conflicts.

Zero Prompting Vs Careful Prompting:

Case	Zero-Shot Prompting	Careful Prompting	Key Output Difference
Stakeholder Identification	"Identify stakeholders for the WolfCafe system."	"Generate a list of potential stakeholders for the WolfCafe system. Organize them into categories: primary (direct users), secondary (support roles), and external (indirect influence). Include a short description for each role."	Zero-shot gives a simple, unordered list. Careful prompting produces a structured, categorized list with detailed descriptions for clarity.
Stakeholder Biases	"Identify stakeholder biases."	"Identify at least 5 stakeholder bias scenarios for the WolfCafe system. Format each as: (a) Stakeholder A's priority, (b) Stakeholder B's conflicting or irrelevant perspective, (c) short rationale."	Zero-shot provides a vague list of biases. Careful prompting gives clearly formatted scenarios showing conflicts between stakeholders, with rationale.

Use Cases	“Write 10 use cases for WolfCafe.”	“Generate 10 detailed use cases for the WolfCafe system. For each, include preconditions, main flow, sub-flow, and alternative flows. Clearly label each section and Use Case title.”	Zero-shot produces a flat list of use cases. Careful prompting provides fully structured use cases with all labeled sections, ready for documentation.
Order Fulfillment Flow	“Describe how orders are handled in WolfCafe.”	“Explain the full order fulfillment process in WolfCafe. Include steps for customer placing order, staff preparing it, notifications, and order completion. Highlight exceptions such as out-of-stock items or failed payment.”	Zero-shot gives a general overview. Careful prompting produces a detailed, step-by-step flow with exception handling, suitable for system design reference.
Item Management Features	“List item management features in WolfCafe.”	“Generate a detailed list of item management features in WolfCafe. For each feature, describe its purpose, the actors involved, and how it interacts with inventory, recipes, and orders.”	Zero-shot produces a simple feature list. Careful prompting provides detailed, contextual descriptions showing relationships between items, inventory, and orders.

Use Cases

UC1 Create new Item/Recipe

1.1 Preconditions

The WolfCafe user is authenticated as Staff or Admin

1.2 Main Flow

1. Staff selects “Create New Item.”
2. Staff enters item or recipe name, description, and base price [S1]
3. Staff submits the form
4. The system saves the new item to catalog
5. Item is now visible to customers

1.3 Subflows

- [S1] Staff may add optional image of product
- [S2] Staff links item to existing ingredients

1.4 Alternative Flows

- [E1] An error message “Invalid price” is displayed if the user enters a price that is negative or blank
- [E2] A prompt is displayed to confirm or cancel the task if the item name is shared with another existing item

UC2 Add Inventory

2.1 Preconditions

The WolfCafe user is authenticated as Staff and the item the inventory applies to exists in the catalog

2.2 Main Flow

1. Staff selects “Manage Inventory”
2. Staff chooses an item
3. Staff enters amount to add
4. System updates the inventory

2.3 Subflows

- [S1] Staff scans barcode to identify them

2.4 Alternative Flows

- [E1] System rejects the input and an error message is displayed if a negative quantity is entered
- [E2] An error message is displayed if the item does not exist

UC3 Manage Staff Users

3.1 Preconditions

The WolfCafe user is authenticated as Administrator

3.2 Main Flow

1. Admin opens “User Management”
2. Admin selects “Add Staff” [S1]
3. Admin enters staff information (i.e. username, role, id)
4. System creates Staff account

3.3 Subflows

- [S1] Admin edits existing staff profile
- [S2] Admin deletes or deactivates staff account

3.4 Alternative Flows

- [E1] System rejects the input and an error message is displayed if the staff username already exists
- [E2] An error is displayed if there are missing required fields

UC4 Manage Customers

4.1 Preconditions

The WolfCafe user is authenticated as an Administrator

4.2 Main Flow

1. Admin opens “User Management”
2. Admin selects customer account
3. Admin edits details or deletes customer account [S1]
4. System saves changes

4.3 Subflows

- [S1] Admin resets customer password

4.4 Alternative Flows

- [E1] An error is displayed if the customer is not found
- [E2] Admin cancels operation.

UC5 Place Customer Order

5.1 Preconditions

The WolfCafe user is authenticated as a Customer [S1] and the inventory has available items

5.2 Main Flow

1. Customer browses products
2. Customer adds items to cart
3. Customer selects “Checkout” [S1]
4. Customer choose tip (0%, 15%, 20%, 25%, or custom)
5. System calculates and applies subtotal, tax, and total
6. Payment is rendered and the order is stored

5.3 Subflows

- [S1] Customer updates cart quantities (add or remove) after checkout

5.4 Alternative Flows

- [E1] An error is displayed if the item is not in stock
- [E2] The order is not stored and and error is displayed if the payment fails

UC6 Fulfill Customer Order

6.1 Preconditions

The WolfCafe user is authenticated as Staff and there are stored, unfulfilled orders

6.2 Main Flow

1. Staff views pending order list
2. Staff selects an order
3. Staff prepares order
4. Staff marks order as fulfilled
5. Staff notifies Customer
6. System updates the order status

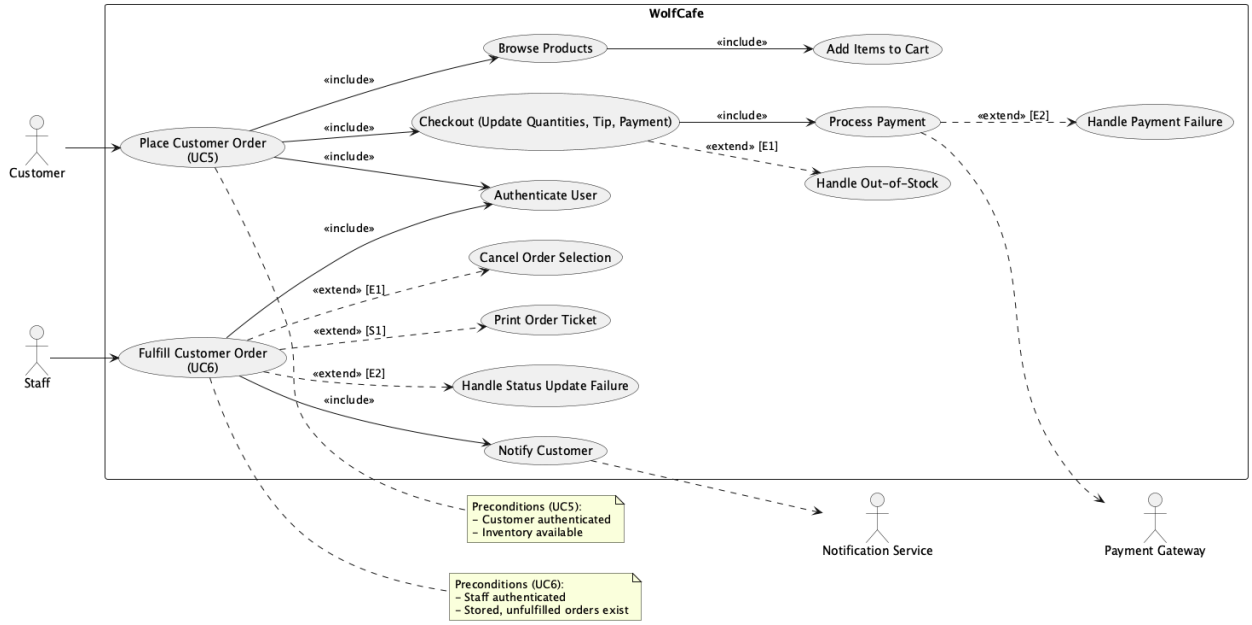
6.3 Subflows

- [S1] Staff prints out order ticket before preparing

6.4 Alternative Flows

- [E1] Staff cancels order selection if wrong order is selected
- [E2] The status will not be updated if there is a system error

UML example for Use Case 5 and Use Case 6:



UC7 Set Sales Tax Rate

7.1 Preconditions

The WolfCafe user is authenticated as an Admin

7.2 Main Flow

1. Admin navigates to “Tax Settings”
2. Admin enters new tax rate [S1]
3. System validates input
4. System saves new tax rate
5. All future orders apply updated rate

7.3 Subflows

- [S1] Admin sets default NC tax rate (2.0%)

7.4 Alternative Flows

- [E1] The system rejects the input tax rate if it is invalid (>100% or negative)
- [E2] The system rejects the tax rate if it doesn’t comply with the NC sales tax laws

UC8 Anonymous Order (optional requirement)

8.1 Preconditions

The WolfCafe user is unauthenticated and considered a guest user

8.2 Main Flow

1. Guest browses menu
2. Guest adds items to cart [S1]

3. Guest proceeds to checkout and enters payment information [S2]
4. Order is stored as anonymous

8.3 Subflows

- [S1] Guest abandons cart and does not place an order
- [S2] Guest provides email to receive receipt of purchase

8.4 Alternative Flows

- [E1] Payment fails; no order created

UC9 Add Discount / Offer / Combo (optional requirement)

9.1 Preconditions

- The user is authenticated as a staff member with sufficient permissions (e.g., Manager or Admin)
- The user is on the Promotions/Offers Management screen in the app.

9.2 Main Flow

1. Staff selects "Add New Promotion" from the app.
2. Staff chooses the type of promotion
    a. Discount (percentage or fixed amount)
    b. Offer (e.g., "Buy 1 Get 1")
    c. Combo (e.g., coffee + pastry at a set price)
3. Staff enters promotion details
4. The system validates input and stores the promotion
5. Promotion becomes active and visible in the ordering interface

9.3 Subflows

- [S1] Save as Draft
- [S2] Schedule Activation

9.4 Alternative Flows

- [E1] Validation Error
- [E2] Unauthorized Role

UC10 View Order History (optional requirement)

10.1 Preconditions

The WolfCafe user is authenticated as a Customer and is logged in

10.2 Main Flow

1. Customer selects "Order History"
2. System displays past orders
3. Customer may review order details [S1]

10.3 Subflows

- [S1] Customer selects an order to repeat/repurchase

10.4 Alternative Flows

- [E1] There are no orders to review; system notifies customer