Project 1a1 (Group 2)

Jawad Saeed (jsaeed@ncsu.edu)
Mason Cormany (mcorman@ncsu.edu)
Himanshu Agarwal (hagarwa4@ncsu.edu)
Omkar Joshi (ojjoshi@ncsu.edu)
Mulya Patel (mspate22@ncsu.edu)

Stakeholders

1. **Admin:** System overseer; manages staff and customer accounts, configures tax rates, and ensures system-wide consistency.

2. **Staff:** Essentially, employees who prepare and fulfill customer orders, and also manage items/recipes and inventory.

3. **Customer:** End user who places orders (items or recipes), applies tips, and picks up fulfilled orders.

4. **Developers:** Build, test, and document the WolfCafe system by extending CoffeeMaker.

5. **Delivery Driver:** Deliver orders to customers and accept tips

6. **University / Campus Management (WolfCafe's setting):** The "owner" of the WolfCafe system, setting policy (e.g., hours, pricing strategy). Interested in compliance, student satisfaction, and financial reporting.

7. **Payment Processor (e.g., Stripe, PayPal, campus card system):** Handles transactions, tips, refunds, and tax compliance. Critical for ensuring secure financial flows.

8. **Health & Safety Inspectors (beyond FDA, local regulators):** Ensure WolfCafe adheres to food handling, hygiene, and NC-specific food service regulations. Their approval is necessary for operation.

9. **Suppliers / Vendors (Food & Packaging):** Provide raw materials (ingredients, bottled drinks, containers). System may generate purchase orders or inventory reports for them.

10. **Customer Support / Help Desk:** Assists customers with order issues (wrong item, refund requests). Ensures smooth resolution of disputes.

11. **Marketing / Promotions Team:** Designs loyalty programs, promotions, or discounts. Interested in usage data and customer trends.

12. **Accessibility Advocates / Compliance Officers:** Ensure the app/website is accessible to people with disabilities. Important for legal compliance and inclusivity.

13. **Auditors / Tax Authorities:** Concerned with the correct application of NC's 2% sales tax. Relevant for financial transparency and avoiding penalties.

14. **Competitors (Indirect Stakeholders):** Other food ordering services (GrubHub, DoorDash). Influence WolfCafe's feature set and user expectations.

Biases

1. Administration vs Customers

- **Bias:** Admins may prioritize efficiency, cost-cutting, or enforcing rules (e.g., minimum orders, restricted hours) that customers dislike.
- **Example:** Admin sets higher prices to increase revenue → customers perceive it as unfair.

2. Administration vs Staff

- **Bias:** Admins may push for stricter policies (speed targets, order quotas) that increase staff workload without considering well-being.
- **Example:** Enforcing a “2-minute order fulfillment” rule stresses staff and reduces quality.

3. Administration vs Developers

- **Bias:** Admins may demand quick feature delivery without considering technical feasibility or long-term maintainability.
- **Example:** Requesting a real-time pickup notification system on a short deadline → developer burnout.

4. Administration vs FDA / Health Inspectors

- **Bias:** Admins may try to cut corners on compliance (e.g., cheaper suppliers, faster food prep) which regulators oppose.
- **Example:** Admin reduces hygiene protocols to save costs → FDA rejects.

5. Customers vs Staff

- **Bias:** Customers want fast, accurate, personalized service; staff want manageable workloads and fair treatment.
- **Example:** Customers expect special modifications not supported by the system → staff frustration

6. Customers vs Developers

- **Bias:** Customers expect seamless, intuitive apps, while developers focus on technical correctness and backend efficiency.
- **Example:** Developer prioritizes system architecture; customers find the UI confusing.

7. Customers vs Payment Processors

- **Bias:** Customers want fee-free, instant payments/refunds; payment processors take transaction fees and may delay refunds.
- **Example:** Customer requests refund → processor takes days → dissatisfaction.

8. Marketing vs Privacy Advocates / Customers

- **Bias:** Marketing wants to track customer behavior for promotions; customers demand privacy and minimal data collection.
- **Example:** Customers reject targeted ads → marketing sees lost revenue.

9. Competitors vs WolfCafe

- **Bias:** Competitors may undercut pricing, impacting WolfCafe's business sustainability.
- **Example:** Customer loyalty pulled toward GrubHub because of lower delivery fees

10. Delivery Driver vs Food Service Worker

- **Bias:** Drivers want grab-and-go readiness; cafes prioritize correct prep.
- **Example:** Incomplete or late bags cause unpaid driver wait time and friction.

11. Suppliers vs Admin

- **Bias:** Suppliers want bulk/standing orders.
- **Example:** Admin may want dynamic just-in-time ordering based on sales data.

Prompt Crafting

Approaches

- **Multi-shot:** Prompting done using ChatGPT. Step-by-step instructions focused on generation of all possible stakeholders and biases, followed by specialized instructions on generation of use cases with a template and example provided for reference. Chat can be found here.

- **Zero-shot:** Prompting done for generating the stakeholders, biases, and use cases in a single prompt using this chat.

- **Meta Prompting:** After some yielding some positive results of stakeholder and use case generation, questions were asked along the lines of "What can I do in my prompting to get the above results faster, better, and with less queries. The answers given were core rules of things to consider when attempting to optimize prompting. Some examples are using proper verbiage like structured or expanded, and to put the question in the framing of a perspective.

- **Model Priming:** Giving the LLM a bigger context window is the best way to prime it. This was done by giving it examples of what was needed and using many short and concise prompts before asking for big or important tasks.

- **CRC Persona Generation:** Created a simple multiagent discussion model that assigned stakeholder roles to LLM personas. Each persona was modeled by an API call to DeepSeek v3. The discussion output was dumped into a structured text file which was then read by GPT5 to generate a list of biases and a list of formatted use cases. The top ten use cases and biases were then considered to be the output of this strategy.

- **Chain-of-Thought:** With chain of thought prompting, the model produced more structured and step-by-step reasoning compared to direct prompts. By first providing a prompt crafting guideline and then asking questions, the responses became longer, more detailed, and better rounded than those generated through simple direct questioning. This method proved particularly useful when asking for further information about any topic.

Observations

Zero Shot vs Multishot

1. The first visible difference between multishot and zero-shot prompting is visible in the stakeholders and their biases. In multishot prompting, we instructed the LLM to generate all possible stakeholders it can think of, in addition to the ones that we felt are integral to the system. Although there is overlap between the zero-shot and multishot approaches, the stakeholders generated by the latter are much richer in terms of covering different aspects. Similarly, this can be observed in biases where the zero-shot approach generates fewer biases

as compared to the multi-shot, which generates twice the amount of biases while addressing all possible stakeholders.

2. The next major difference we observed was in the generation of use cases. While both prompting strategies resulted in 10 use cases, the ones generated by multi-shot prompting are evidently more detailed and address various possible scenarios as compared to the ones generated by the zero-shot approach. Although the use case template was provided in both prompting strategies, the presence of an example use case in the multi-shot prompting approach led to better generation of the use cases, highlighting the importance of examples in addition to providing templates to the LLM

CRC Persona Generation

1. The Persona-based generation-analysis technique was quite good at generating really granular examples and use cases.

2. It was also able to describe the biases between different stakeholders better than other techniques.

3. Playing around with different models, I was able to get more coherent discussion from reasoning focused models like DeepSeek than Gemini although this may also be due to parameter size differences.

4. GPT 5 was able to analyse the discussion transcript to generate sensible results with minimal prompting.

Use Cases:

UC1: Create/Edit/Delete Staff/Customers (Admin)

1.1 Preconditions

- Admin is logged into the system.
- Staff role functionality exists in the system.
- Customer role functionality exists in the system

1.2 Main Flow

1. Admin selects “Manage Users.”
2. Admin chooses to create, edit, or delete a staff/customer account.
3. System updates user roles accordingly.

1.3 Subflows / Extensions

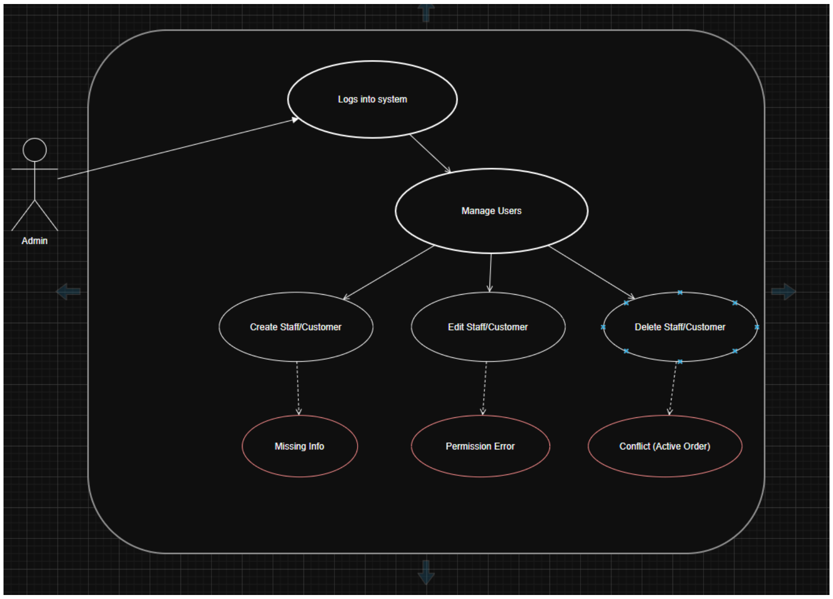
- [Create Staff/Customer] Admin enters staff/customer details → system stores new user.
- [Edit Staff/Customer] Admin updates staff/customer info → system saves changes.
- [Delete Staff/Customer] Admin removes staff/customer account → system deletes user from database.

1.4 Alternative / Error Flows

- [Missing Info] Admin leaves required fields blank → system prompts correction.
- [Permission Error] Non-admin attempts access → system denies.

- [Conflict] Staff/customer account is tied to an active order → deletion blocked until resolved.

1.5 Diagram



UC2: Place Customer Order

2.1 Preconditions

- Customer is logged in.
- Recipes/Items exist in inventory.

2.2 Main Flow

1. Customer browses menu.
2. Customer adds items to cart (quantity specified).
3. System calculates subtotal + tax [Apply Tax].
4. Customer selects tip amount [Add Tip].
5. Customer confirms and places order.

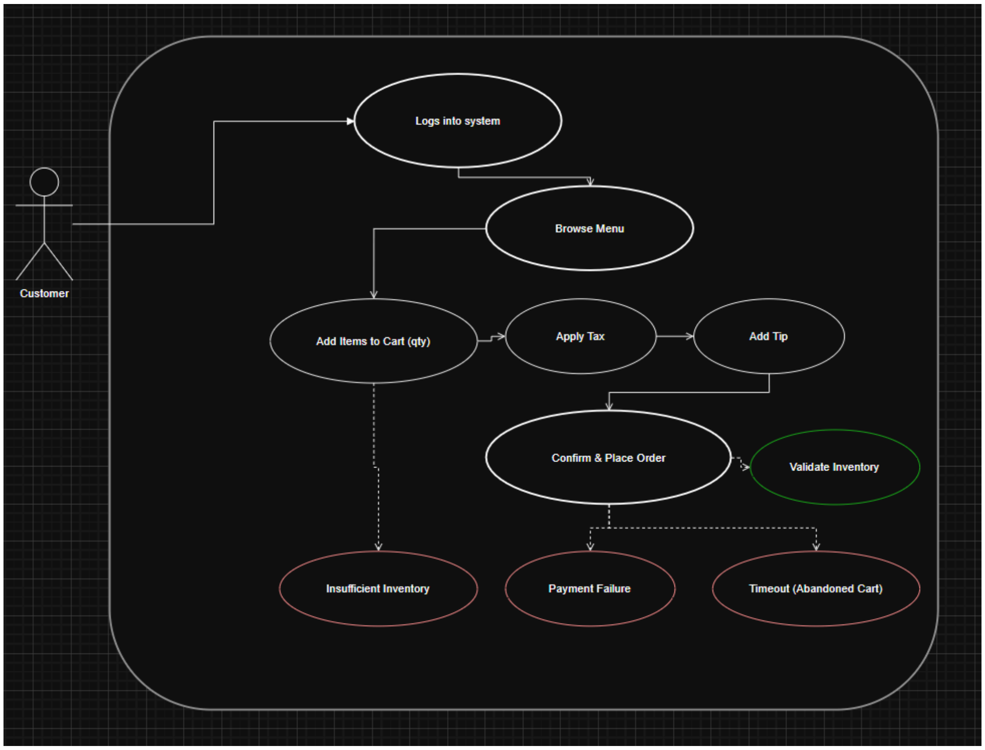
2.3 Subflows / Extensions

- [Apply Tax] System applies NC’s 2% sales tax or Admin-set tax.
- [Add Tip] Customer selects 15%, 20%, 25%, or enters custom tip.
- [Validate Inventory] System checks item stock before confirming.

2.4 Alternative / Error Flows

- [Insufficient Inventory] Item unavailable → system alerts and removes from cart.
- [Payment Failure] Payment processor error → order not submitted.
- [Timeout] Customer abandons cart → order cancelled.

2.5 Diagram



UC3: Staff Views & Fulfills Orders

3.1 Preconditions

- Staff is logged in.
- Orders exist in “New” status.

3.2 Main Flow

1. Staff opens “Order Queue.”
2. Staff selects an order to fulfill.
3. Staff assembles items.
4. Staff marks order as “Fulfilled.”

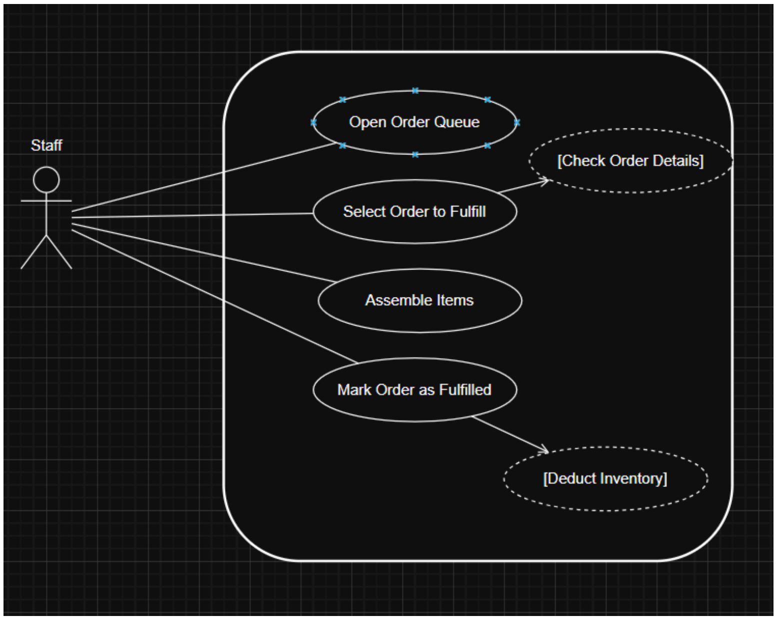
3.3 Subflows / Extensions

- [Check Order Details] Staff views item list & quantities.
- [Deduct Inventory] System updates inventory upon fulfillment.

3.4 Alternative / Error Flows

- [Inventory Shortage] Staff discovers missing ingredient → order flagged as delayed.
- [Wrong Items Prepared] Staff updates fulfillment → order corrected.

3.5 Diagram



UC4: Customer Pickup Notification

4.1 Preconditions

- Customer has placed an order.
- Staff has marked order as fulfilled.

4.2 Main Flow

1. System updates customer portal to show “Order Ready.”
2. Customer arrives at WolfCafe counter.
3. Customer picks up order.

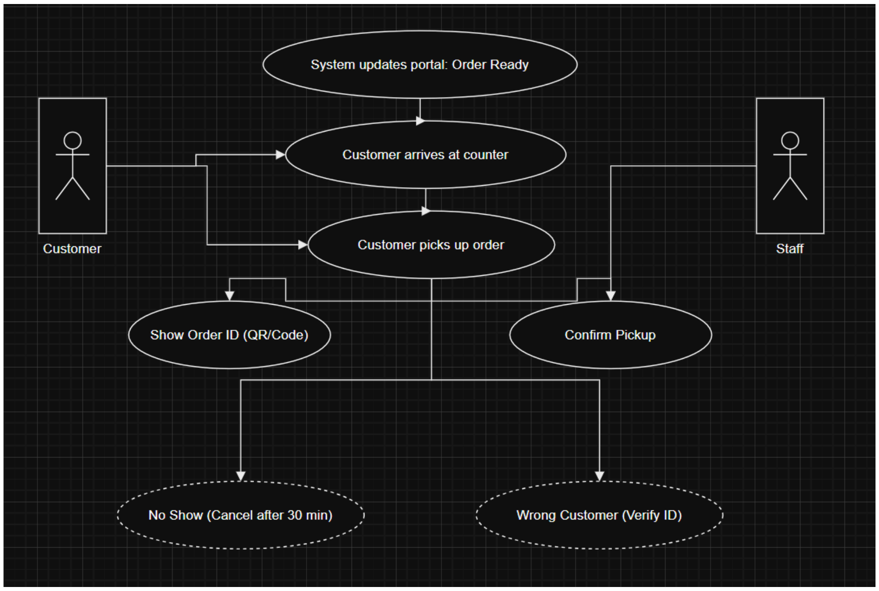
4.3 Subflows / Extensions

- [Show Order ID] Customer shows confirmation code/QR.
- [Confirm Pickup] Staff updates order to “Picked Up.”

4.4 Alternative / Error Flows

- [No Show] Customer doesn’t arrive → order held for 30 min, then cancelled.
- [Wrong Customer] Another person attempts pickup → staff verifies identity.

4.5 Diagram


System updates portal: Order Ready
Customer arrives at counter
Customer picks up order
Customer
Staff
Show Order ID (QR/Code)
Confirm Pickup
No Show (Cancel after 30 min)
Wrong Customer (Verify ID)

UC5: Set Tax Rate (Admin)

5.1 Preconditions

- Admin is logged in.
- System tax rate setting is accessible.

5.2 Main Flow

1. Admin selects “System Settings.”
2. Admin updates tax rate.
3. System saves new rate.

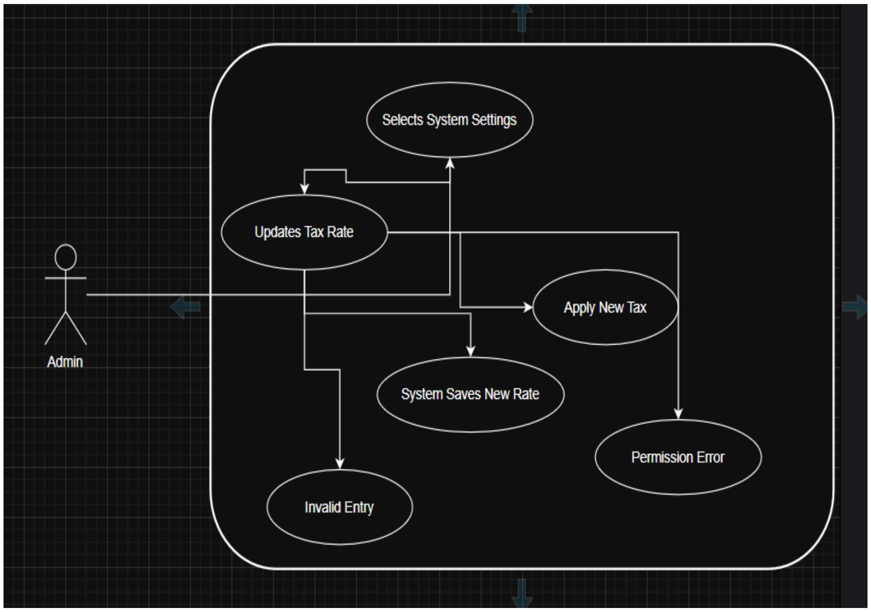
5.3 Subflows / Extensions

- [Apply New Tax] Future orders calculate with updated rate.

5.4 Alternative / Error Flows

- [Invalid Entry] Admin enters negative or >100% → system rejects.
- [Permission Error] Non-admin attempts change → denied.

5.5 Diagram



UC6: Manage Inventory (Supply Coordinator)

6.1 Preconditions

- Supply Coordinator is logged in.
- Supplier contracts exist.

6.2 Main Flow

1. Coordinator reviews current stock levels.
2. Coordinator places restock order with supplier.
3. Inventory updates upon delivery.

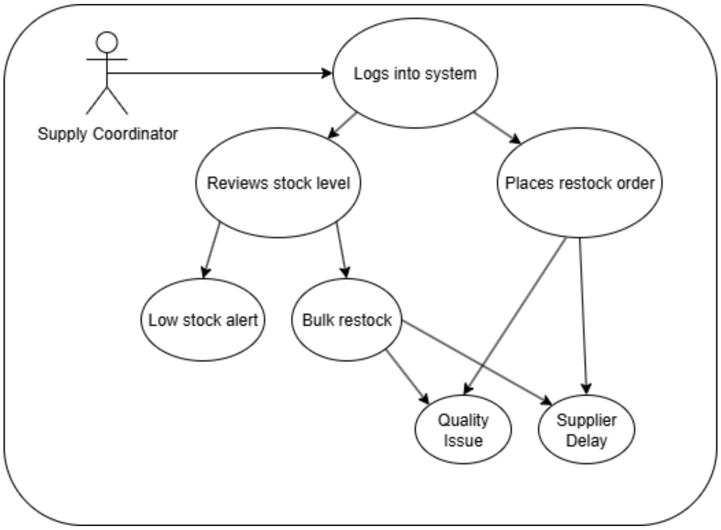
6.3 Subflows / Extensions

- [Low Stock Alert] System notifies when item below threshold.
- [Bulk Restock] Coordinator orders multiple items at once.

6.4 Alternative / Error Flows

- [Supplier Delay] Supplier shipment late → stockout occurs.
- [Quality Issue] Received goods defective → rejected.

6.5 Diagram



UC7: Ensure Regulatory Compliance (FDA / Health Inspectors)

7.1 Preconditions

- Orders and ingredients exist.
- Regulatory audits scheduled.

7.2 Main Flow

1. Inspector logs into compliance dashboard.
2. Inspector reviews safety reports & supplier certifications.

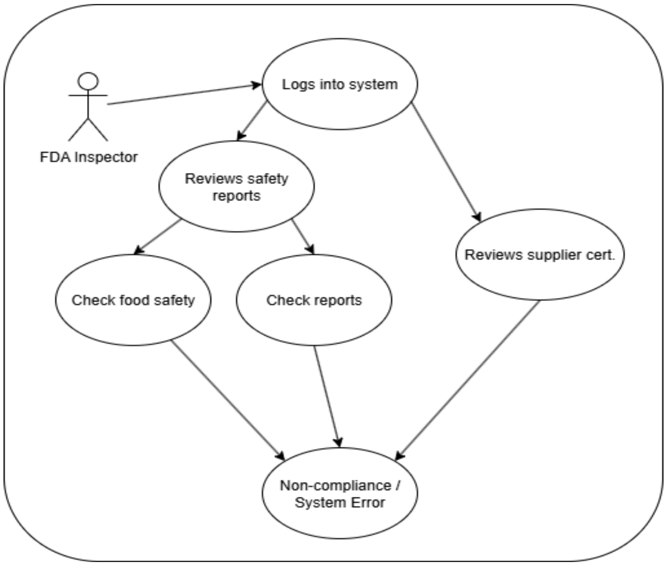
7.3 Subflows / Extensions

- [Check Food Safety] Verify expiration dates & storage conditions.
- [Check Reports] Review temperature logs, hygiene checks.

7.4 Alternative / Error Flows

- [Non-Compliance] Report filed against WolfCafe → penalties applied.
- [System Error] Logs missing or corrupted → investigation triggered.

4.5 Diagram



UC8: Marketing Campaign Setup (Marketing Team)

8.1 Preconditions

- Marketing account exists.
- Campaign system enabled.

8.2 Main Flow

1. Marketing logs into portal.
2. Creates discount or loyalty campaign.
3. System applies promotion to customer accounts.

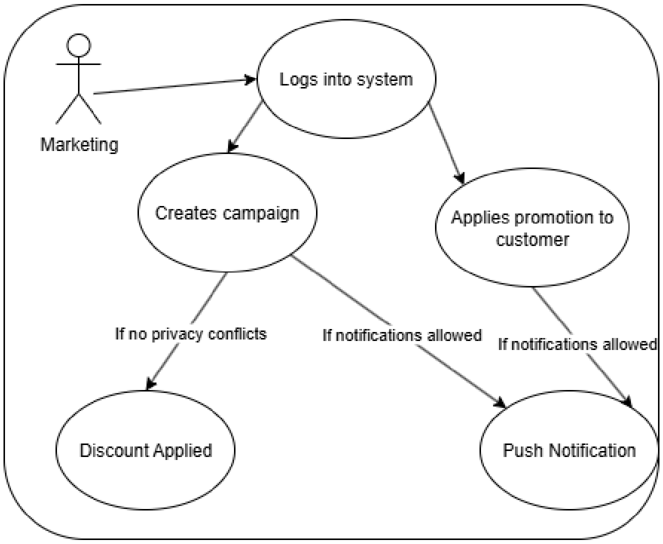
8.3 Subflows / Extensions

- [Push Notification] Promo sent to app users.
- [Discount Applied] System updates checkout totals.

8.4 Alternative / Error Flows

- [Privacy Conflict] Promotion requires excessive data collection → blocked by privacy officer.
- [Customer Opt-Out] User disables marketing notifications.

8.5 Diagram



UC9: Accessibility Review (Accessibility Advocates)

9.1 Preconditions

- System UI deployed.
- Accessibility audit planned.

9.2 Main Flow

1. Advocate tests WolfCafe interface with assistive tools.
2. Reports issues (e.g., missing alt text, low contrast).
3. Developers update design.

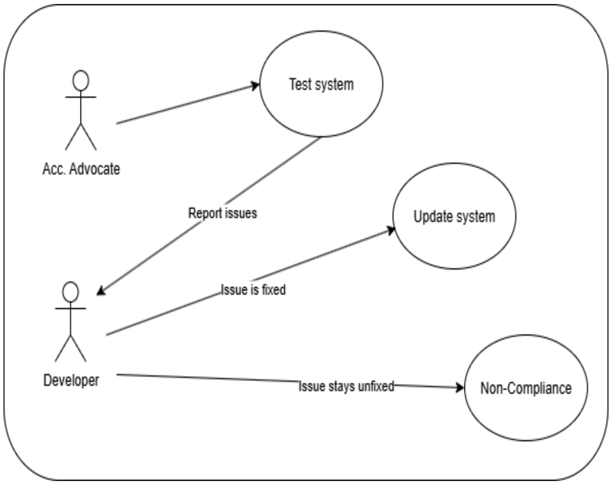
9.3 Subflows / Extensions

- [Screen Reader Test] Verify order placement with text-to-speech.
- [Keyboard Navigation] Ensure usable without mouse.

9.4 Alternative / Error Flows

- [Non-Compliance] Issues unresolved → WolfCafe risks accessibility lawsuit.

4.5 Diagram



UC10: Delivery Flow (Delivery Driver)

10.1 Preconditions

- UC3 marks order Ready; driver supply available.

10.2 Main Flow

- Dispatch offers order with full pay breakdown (base, tip estimate, boosts) and distance/time.
- Driver accepts; in-app navigation starts.
- Driver checks in at the store; staff confirms order readiness.
- Driver verifies sealed bag, labels, and drink carriers; captures pickup confirmation.
- The driver departs for delivery.

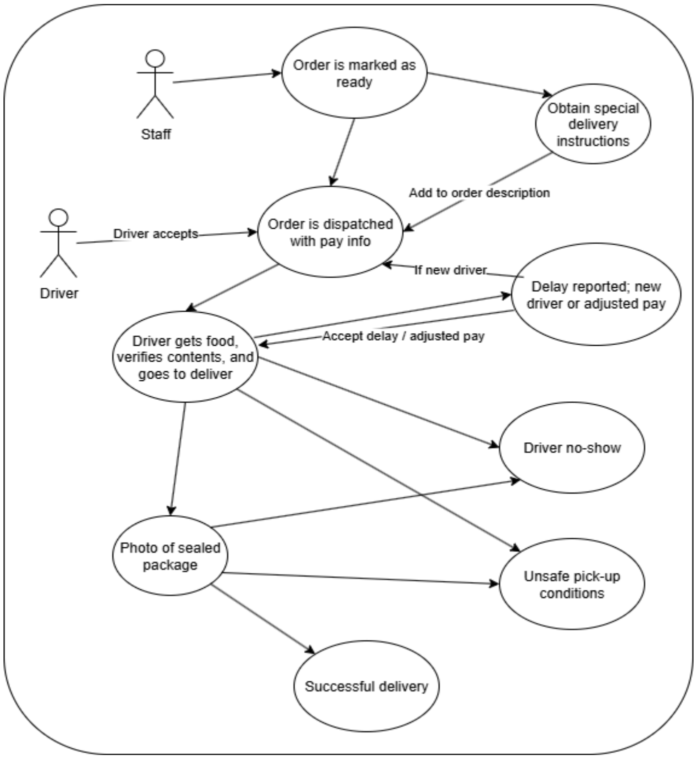
10.3 Subflows / Extensions

- Parking & Access Instructions: Store provides curbside/door details.
- Photo of Sealed Package: For chain-of-custody and dispute resolution.
- ID Check Flow: For age-restricted items if applicable.

10.4 Alternative / Error Flows

- Excessive Wait at Pickup: Driver reports delay; dispatch may reassign; pay adjusted.
- Driver No-Show: System re-dispatches to next driver; customer notified (UC13).
- Unsafe Pickup Conditions: Driver cancels with reason; order rerouted.

10.5 Diagram


Staff
Order is marked as ready
Obtain special delivery instructions
Add to order description
Driver
Driver accepts
Order is dispatched with pay info
If new driver
Delay reported; new driver or adjusted pay
Accept delay / adjusted pay
Driver gets food, verifies contents, and goes to deliver
Driver no-show
Photo of sealed package
Unsafe pick-up conditions
Successful delivery