Project 1a1 - Food Delivery System: The Hungry Wolf

List of Stakeholders

Core Stakeholders

- Customers – individuals placing food orders.
- Restaurants/Food Vendors – businesses that provide meals.
- Delivery Partners / Riders / Drivers – people delivering food.
- Platform Admins – manage the system, monitor activity.
- Investors / Shareholders – interested in financial performance.

Extended Stakeholders

- Restaurant Staff – chefs, kitchen workers, and order packers.
- Restaurant Managers / Owners – oversee restaurant performance on the platform.
- Payment Service Providers – banks, wallets, credit card companies, UPI, PayPal.
- Marketing & Sales Teams – design promotions, campaigns, discounts.
- Technology Teams – developers, designers, QA testers, data scientists.
- Customer Support Staff – handle complaints, refunds, and order issues.

Indirect / Overlooked Stakeholders

- Food Packaging Suppliers – provide containers, cutlery, and eco-friendly options.
- Insurance Providers – cover drivers, accidents, and business liability.
- Regulators / Government Agencies – driver safety, food safety, road laws, taxation.

Stakeholder biases

1. Customer vs. Delivery Partner

- Customers need fast delivery at the lowest possible fee.
- Delivery partners need fair compensation, safe working conditions, and manageable delivery times.
- Clash: Customers may expect 30-minute delivery guarantees, while drivers want enough time to deliver safely without penalties.

2. Restaurant vs. Platform Admin

- Restaurant needs: Higher margins, freedom to set prices, and visibility.
- Platform needs: Competitive pricing, commissions on sales, and control over promotions.
- Clash: Restaurants often feel platforms take excessive commissions, while platforms push discounts that reduce restaurant profits.

3. Customer vs. Restaurant Staff

- Customer need: Wide customization options (extra toppings, no onions, special diets).
- Restaurant staff need: Streamlined kitchen operations with minimal complexity.
- Clash: Excessive customization slows down the kitchen, increases errors, and frustrates staff.

4. Investors vs. Environmental Groups

- Investor needs: Rapid growth, market share, cost efficiency.
- Environmental need: Sustainable packaging, reduced carbon footprint.
- Clash: Cheap single-use plastics help cut costs and scale fast, but harm sustainability goals.

5. Regulators vs. Customers

- Regulator needs: Compliance with food safety, labor rights, and taxation laws.
- Customer needs: Convenience, lowest prices, quick delivery.
- Clash: Stricter labor regulations or food standards may increase costs, which ultimately raise customer prices or slow delivery.

Review on Prompt crafting

Aspect	**Zero-Shot Prompting**	**Careful Prompting**
Definition	Asking the model to generate ideas for the food delivery system (features, use cases, stakeholders, etc.) without giving it detailed instructions..	Providing the model with structured instructions, templates, or constraints (e.g., preconditions, flows, stakeholder conflicts) to guide the model’s response.
Example	“List use cases for a food delivery app like The Hungry Wolf.”	“Write a use case for ‘Place Order’ in The Hungry Wolf. Include: Preconditions, Main Flow, Subflows, Alternative Flows. Keep it under 1 page.”

Strengths	- Quick brainstorming of features (order tracking, tipping, loyalty programs). - Helps explore stakeholder conflicts without much setup. - Useful for prototyping ideas like marketing campaigns or eco-friendly packaging.	- Ensures structured outputs that match assignment requirements (e.g., 10 use cases with flows). - Reduces ambiguity in requirements (e.g., defining roles for restaurant vs. rider). - Produces clear, consistent documentation.
Limitations	- May give vague or incomplete use cases (e.g., skipping subflows like payment failure). - Inconsistent format makes it harder to combine into the final report. - Might miss course-specific details (like stakeholder biases).	- Requires more time to design prompts. - Can feel rigid and reduce creative ideas. - Needs iteration when outputs are too mechanical.
Best Used For	- Generating stakeholder lists and conflicts. - Brainstorming potential system features (subscription meals, drone delivery, etc.). - Early exploration of risks and opportunities.	- Writing **final project deliverables**: stakeholder analysis, structured use cases, prompt crafting review. - Technical sections like system design, requirements, or complaint-handling flows.
Overall Approach	**Efficiency and exploration** → Use it to rapidly gather diverse ideas for The Hungry Wolf.	**Precision and control** → Use it to turn brainstorming into detailed use cases and stakeholder analyses.
Complementary Use	Start with zero-shot prompting to brainstorm raw ideas (e.g., "List 20 conflicts between Hungry Wolf stakeholders"). Then switch to careful prompting to rewrite them into structured deliverables (e.g., 5 conflicts explained in paragraph form)..	Works best when paired with zero-shot prompting as the first stage → ensures creativity is captured before being polished into final structured outputs.

Use Cases

Use Case 1: Customer Registration

Preconditions: Customer has internet access and opens the Hungry Wolf app/website.

Main Flow:

1. Customer selects "Sign Up."
2. Enter name, email, phone, and password.
3. System validates input.
4. An account is created, and confirmation is sent.

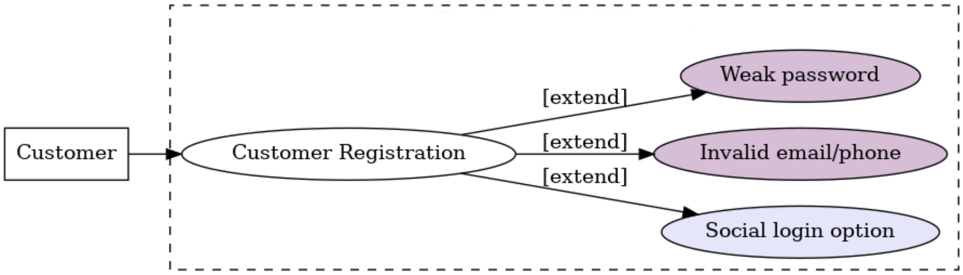
Subflows:

1a: Social login option (Google, NCSU email).

Alternative Flows:

3a: Invalid email/phone → system shows error.

3b: Password too weak → system prompts for a stronger password.



Use Case 2: Restaurant Registration

Preconditions: The Restaurant owner wants to join the platform.

Main Flow:

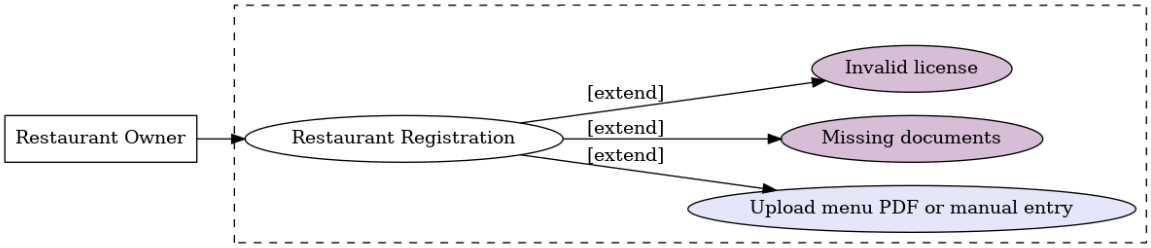
1. The restaurant selects "Register Restaurant."
2. Fills business details, menu, tax ID, and license info.
3. The system verifies documents.
4. Admin approves restaurant accounts.

Subflows:

2a: Upload menu as PDF or manual entry.

Alternative Flows:

3a: Missing documents → registration kept pending.



3b: Invalid license → rejection notice sent.

Use Case 3: Browse Menu

Preconditions: Customer logged in.

Main Flow:

1. The customer searches for a restaurant or food item.
2. The menu displayed items, prices, and offers.
3. The customer selects the desired items.

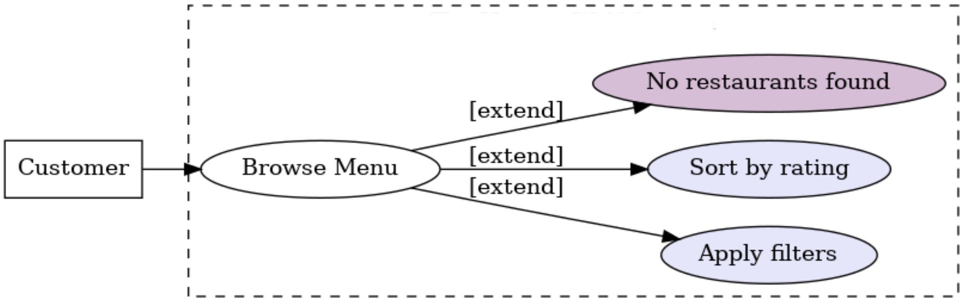
Subflows:

1a: Apply filters (veg/non-veg, cuisine, price range).

1b: Sort by popularity or rating.

Alternative Flows:

1a: No results → show "No restaurants found" message.



Use Case 4: Place Order

Preconditions: The customer has items in their cart.

Main Flow:

1. Customer reviews cart.
2. Selects the delivery address and time.
3. Choose a payment method.
4. Confirms order.
5. The system sends orders to restaurants.

Subflows:

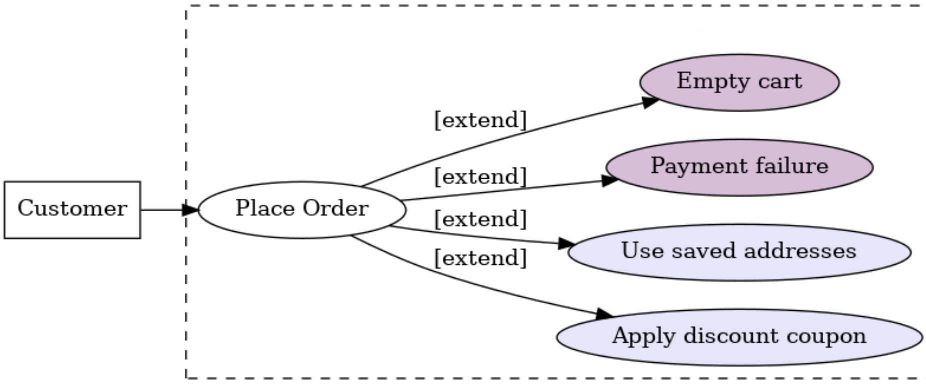
3a: Apply discount coupon.

3b: Use saved addresses.

Alternative Flows:

3a: Payment failure → retry option

4a: Cart empty → system prevents checkout.



Use Case 5: Restaurant Accepts/Rejects Order

Preconditions: Order placed by customer.

Main Flow:

1. The restaurant receives an order notification.
2. Accepts the order.
3. Starts food preparation.

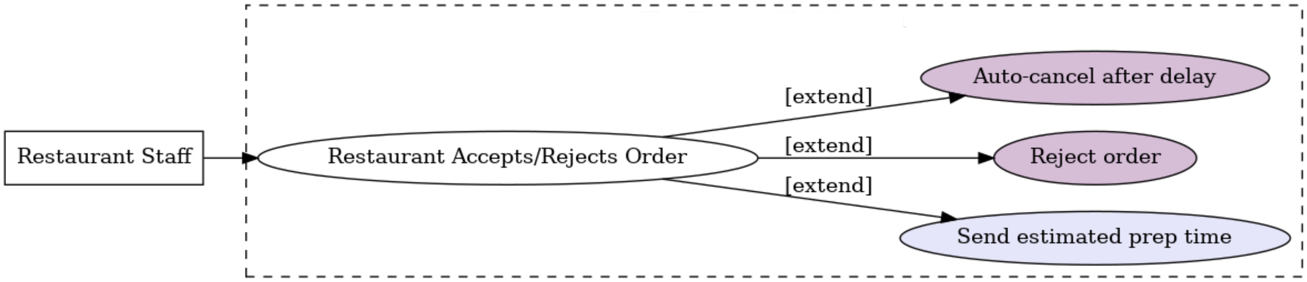
Subflows:

2a: Estimated preparation time sent to the customer.

Alternative Flows:

2b: Rejects order (due to unavailability).

2c: Restaurant delays confirmation → system auto-cancels after set time.



Use Case 6: Assign Delivery Partner

Preconditions: The Restaurant accepts orders.

Main Flow:

1. The system finds nearby delivery partners.
2. Sends requests to the closest rider.
3. Rider accepts.

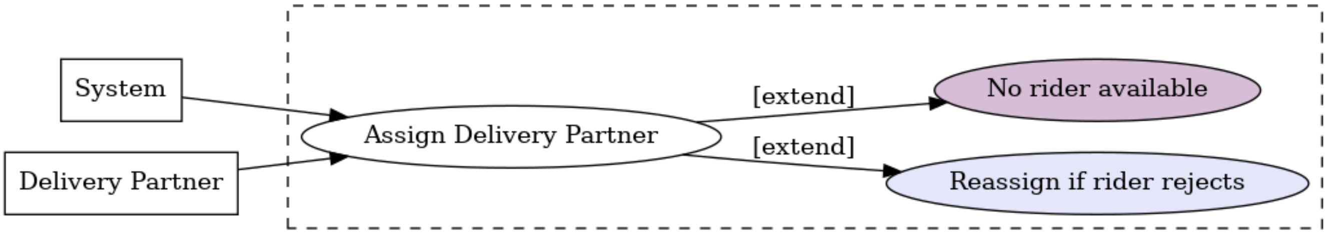
4. Customer notified with ETA and tracking link.

Subflows:

2a: Rider rejects → system reassigns to next rider.

Alternative Flows:

2b: No rider available → system cancels order and refunds.



Use Case 7: Live Order Tracking

Preconditions: The Delivery partner accepted the assignment.

Main Flow:

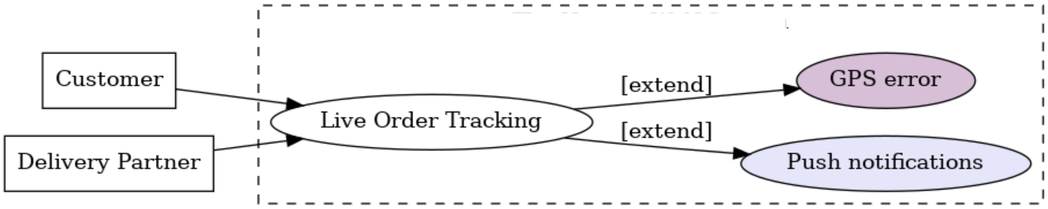
1. Customer opens app → sees real-time map.
2. System updates rider location.
3. Status changes (Picked Up → On the Way → Delivered).

Subflows:

2a: Push notifications at each stage.

Alternative Flows:

2b: GPS error → show "Unable to fetch location."



Use Case 8: Order Delivery & Confirmation

Preconditions: Rider arrives at the customer's address.

Main Flow:

1. Rider hands food to customers.
2. Customer confirms receipt (OTP/signature).

3. System marks the order as delivered.

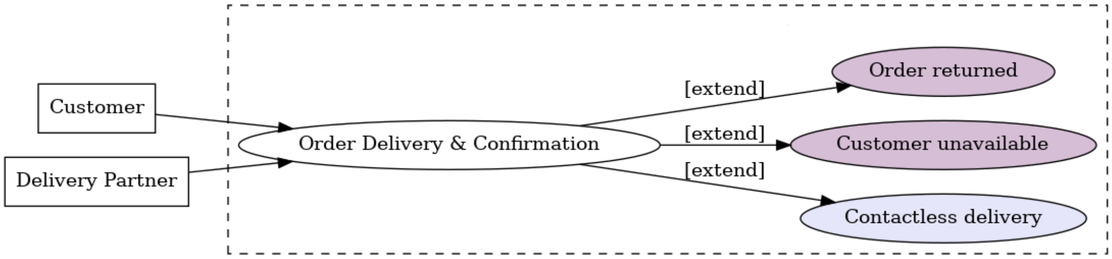
Subflows:

1a: Contactless delivery option → rider drops at doorstep and sends photo.

Alternative Flows:

2a: Customer unavailable → rider calls.

2b: No response → order returned to restaurant.



Use Case 9: Customer Feedback & Rating

Preconditions: Order delivered.

Main Flow:

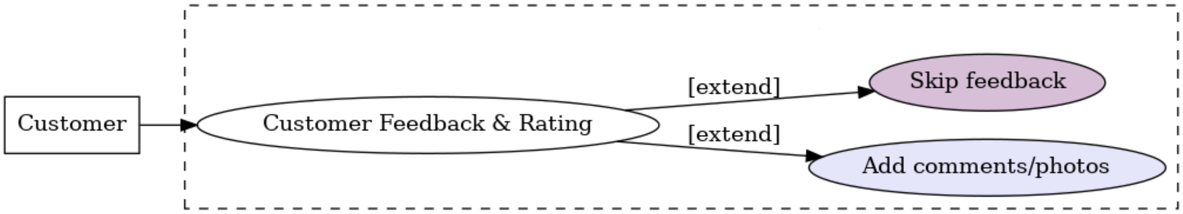
1. The system prompts customers for feedback.
2. Customer rates restaurant and delivery partner.
3. Feedback is stored for analytics.

Subflows:

2a: Option to leave comments/photos.

Alternative Flows:

1a: Customer skips feedback.



Use Case 10: Refund / Complaint Handling

Preconditions: Customer reports an issue.

Main Flow:

1. Customer selects "Help/Support Choose issue type (wrong item, late delivery, etc.).
2. The system checks order details.
3. Admin/Support review the news complaint.

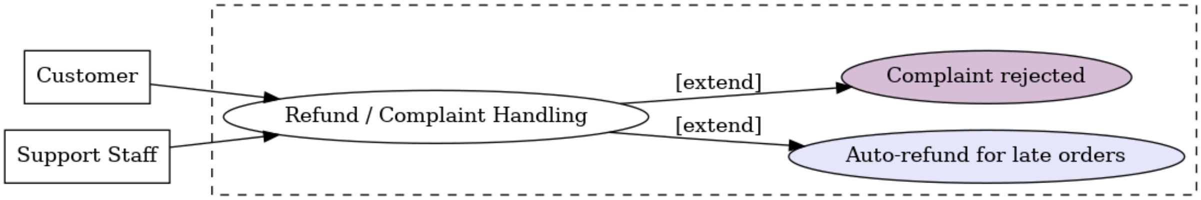
4. Refund or compensation issued.

Subflows:

2a: Automated refund for common issues (late by >30 mins).

Alternative Flows:

4a: Complaint rejected → system explains reason.


Customer
Support Staff
Refund / Complaint Handling
[extend]
Complaint rejected
[extend]
Auto-refund for late orders