Project 1c1 - Food Delivery System: The Hungry Wolf

New Use cases

Use Case 31: Customer Login & Session Management

Preconditions: Customer has already registered.

Main Flow:

1. Customer selects "Log In."
2. Enter email/phone and password.
3. System validates credentials.
4. The customer is logged in and redirected to the home page.

Subflows:

2a. "Remember Me" option → keeps the customer logged in for the next session.

2b. Option to log in with OTP (instead of password).

Alternative Flows:

3a. Invalid credentials → show "Invalid Email/Password" error.

3b. Forgotten password → redirect to "Reset Password" flow.

Use Case 32: Password Reset / Recovery

Preconditions: The customer/Restaurant has a registered account but has forgotten the password.

Main Flow:

1. The user selects "Forgot Password."
2. System prompts for email/phone.
3. The system sends a password reset link or OTP.
4. The user enters a new password.
5. System updates password and confirms reset.

Subflows:

2a. Option to choose SMS OTP or email link.

Alternative Flows:

3a. Email/phone not found → system shows "Account does not exist."

4a. Weak new password → system asks for a stronger password.

Use Case 33: Search by Cuisine / Category

Preconditions: Customer is logged in and on the Browse screen.

Main Flow:

1. Customers enter cuisine (e.g., "Italian") or category (e.g., "Pizza").
2. The system filters restaurants and items.
3. Results displayed with restaurant name, ratings, and delivery time.

Subflows:

1a. Voice search option.

2a. Suggested keywords displayed.

Alternative Flows:

2b. No results → show "No items found" with related suggestions.

Use Case 34: Favorite Restaurants

Preconditions: Customer is logged in.

Main Flow:

1. The customer views a restaurant page.
2. Selects "Add to Favorites."
3. The system saves the restaurant to their favorites list.
4. Customers can quickly access their favorite restaurants later.

Subflows:

2a. The favorite list can be sorted by cuisine or order frequency.

Alternative Flows:

2b. Restaurant becomes unavailable → system greys out restaurant in favorites.

Use Case 35: Order Cancellation (Customer-Initiated)

Preconditions: Customer has placed an order, and the restaurant has not started cooking yet.

Main Flow:

1. The customer goes to "My Orders."
2. Selects an active order.
3. Click "Cancel Order."
4. System checks status → if allowed, cancels and processes refund.
5. Restaurant and delivery partners are notified.

Subflows:

3a. Cancellation reason required (e.g., "Changed mind," "Ordered by mistake").

Alternative Flows:

4a. Order already prepared → cancellation not allowed, customer prompted to contact support.

4b. Refund fails → mark as pending in admin dashboard.

Use Case 36: Order Cancellation (Restaurant-Initiated)

Preconditions: A customer has placed an order, but the restaurant cannot fulfill it.

Main Flow:

1. The restaurant views incoming orders.
2. Selects "Cancel Order."
3. Provides cancellation reason (e.g., "Out of stock," "Closed early").
4. The customer is notified instantly of the refund initiation.
5. Admin dashboard logs cancellation reason.

Subflows:

3a. Restaurants may suggest alternative items before cancellation.

Alternative Flows:

4a. Customer disputes → system escalates to admin support.

Use Case 37: Basic Push Notifications

Preconditions: Customer has the app installed and notifications enabled.

Main Flow:

1. System sends push notifications for key order updates:
2. Order Accepted
3. Food Picked Up
4. Delivery Arriving Soon
5. Order Delivered
6. Customer taps notification → redirected to order tracking screen.

Subflows:

1a. Notification includes estimated delivery time.

Alternative Flows:

2a. Notifications disabled → updates shown only in-app.

Use Case 38: Delivery Partner Check-in/Check-out

Preconditions: The Delivery partner has an account.

Main Flow:

1. Rider opens a delivery partner app.
2. Selects "Go Online."
3. The system marks riders as available for orders.
4. At the end of the shift, the rider selects "Go Offline."
5. The system removes riders from the pool.

Subflows:

2a. Riders can set a temporary "Busy" mode during a shift.

Alternative Flows:

3a. Rider tries to go online without required documents (e.g., ID verification pending) → system blocks access.

Use Case 39: Customer Receives Invoice / Receipt

Preconditions: Order is successfully placed and paid.

Main Flow:

1. The system generates a digital invoice after payment.
2. The invoice includes order details, taxes, discounts, and delivery charges.
3. Customers can download receipts (PDF) or view them in-app.

Subflows:

2a. Invoice automatically emailed to customer.

Alternative Flows:

3a. Email delivery fails → system retries or prompts customers to update email.

Use Case 40: Customer Quick Reorder

Preconditions: Customer has an order history.

Main Flow:

1. Customer navigates to "Order History."
2. Selects a past order.
3. Click "Reorder."
4. The system adds the same items to the cart.
5. The customer reviews the cart and places an order again.

Subflows:

3a. Option to modify quantities or remove items before checkout.

Alternative Flows:

4a. Item unavailable → system suggests alternatives.

4b. Restaurant closed → system suggests similar open restaurants.

Use Case 41: Customer Multiple Payment Methods

Preconditions: Customer is logged in and at checkout.

Main Flow:

1. Customer opens the "Payment Options" section.
2. Adds a new card, UPI, or wallet.
3. Selects the preferred payment method for the order.
4. System securely saves encrypted payment details for future use.

Subflows:

3a. Option to set a “Default Payment Method.”

Alternative Flows:

2a. Invalid card/UPI → system shows error.

3b. Payment provider unavailable → system prompts customer to try another method.

Use Case 42: Restaurant Sets Delivery Radius

Preconditions: Restaurant is registered and logged in.

Main Flow:

1. Restaurant navigates to delivery settings.
2. Enters maximum delivery distance (e.g., 5 km).
3. The system restricts visibility of the restaurant to customers within that range.

Subflows:

2a. Different radii can be set for peak vs. off-peak hours.

Alternative Flows:

3a. Restaurant enters illogical values (e.g., 0 km or 100 km) → system shows validation error.

Use Case 43: Rider Earnings Summary

Preconditions: Rider has completed at least one order.

Main Flow:

1. Rider navigates to “Earnings.”

2. The system shows completed deliveries, payouts, and tips.
3. The earnings summary is displayed daily/weekly.

Subflows:

2a. Rider can export earnings summary as PDF.

Alternative Flows:

3a. No completed deliveries → system shows “No earnings yet.”

Use Case 44: Order Status Notifications for Restaurants

Preconditions: Restaurant has active orders.

Main Flow:

1. System notifies restaurant for each key order stage:
2. New Order Received
3. Order Accepted
4. Food Picked Up
5. Delivered
6. Restaurant dashboard updates automatically.

Subflows:

2a. Restaurants can enable/disable SMS/email alerts.

Alternative Flows:

1a. Notification delivery fails → order still appears in the dashboard.

Use Case 45: Guest Checkout with Mobile Number

Preconditions: Customer does not want to create a full account.

Main Flow:

1. The customer selects "Continue as Guest."
2. Provides mobile number for verification (OTP).
3. Places order as usual (browse → cart → checkout).
4. System links order to guest profile.

Subflows:

2a. Guests may later upgrade to a full account, keeping order history.

Alternative Flows:

2b. OTP not received → retry option provided.

3a. Guest tries to apply for a promotion → system shows “Login Required.”

Reflection Document

1. Output given by ChatGPT

How we decided what NOT to do

We asked the LLM these 3 questions:

- Does it directly validate the food delivery loop (customer orders → restaurant prepares → delivery partner delivers)?
- Is it complex but not essential (e.g., subscriptions, fraud detection), cut it.
- Can humans/admins handle it at a low scale (e.g., customer support, account deletion)?

Possible negative impacts of MVP

- Customers may be disappointed that there are no loyalty rewards, subscriptions, eco-friendly delivery options, or saved items.
- Restaurants may miss detailed analytics, flexible timing controls, or review management tools.
- Delivery Partners may miss tipping options.
- Admins may struggle without automated fraud detection, promotions, or chatbot support.

Changes made to appease stakeholders

- Added basic feedback/rating (Use Cases 9 & 28) so customers and restaurants feel heard.
- Included promotions for customers (Use Case 24: Apply Promotion), but simplified (only manual, fixed coupon). This gives marketing flexibility without needing a full promo dashboard.
- Kept Menu Management (Use Case 14) so restaurants feel in control and not stuck with static menus.
- Keep Order History (Use Case 16) because customers will expect at least a record of what they ordered.
- Kept Refund/Complaint Handling (Use Case 10) at a minimal manual level, so early adopters don't feel cheated if something goes wrong.

2. Output given by Claude

How we decided what NOT to do

- **We applied three key criteria:** Essential for Core Transaction Flow (does it enable basic order-to-delivery?), Technical Complexity vs. Value (maximum learning with minimum effort), and Market Validation Priority (what must we prove first?).
- **Decision Process:** Keep features enabling money exchange and basic operations (restaurant onboarding, menu management). Exclude experience enhancements, manually manageable admin features, and complex developments without immediate revenue impact.
- **Key Exclusions:** Customer ratings/loyalty programs, order history, restaurant analytics dashboards, promotional tools, delivery partner tipping/ratings, and automated fraud detection.

Negative Impacts on Stakeholders

- **Customers**: Limited engagement without ratings/loyalty programs, a basic experience lacking personalization and order history, and no promotional benefits or convenience features, such as scheduling/group orders.
- **Restaurants:** No performance analytics or marketing tools, missing customer feedback loop, basic menu management without promotional capabilities.
- **Delivery Partners:** No incentive systems (tipping/ratings) or performance tracking, reducing motivation and optimization opportunities.

MVP modifications to address stakeholders' concerns

- **Customer Appeasement:** Added simplified profile management for address saving, basic order history, and a manual complaint handling process.
- **Restaurant Appeasement:** Enhanced menu management with item images and categorization, simple performance metrics (order count/revenue), and direct customer communication via order notes.
- **Business Appeasement:** Basic admin panel for restaurant approval and issue resolution, essential analytics tracking (order volume, popular restaurants, delivery times), and manual discount application for special cases.
- **Technical Focus:** Prioritized robust core features, mobile-first design, and fast loading for optimal user experience.

Prompt History:

- ChatGPT
- ClaudeAI

Use cases we will NOT build in MVP (with justification)

- Loyalty Points & Rewards → Not needed for proving food ordering works.
- Scheduled Order Placement → Adds complexity (future queueing, payment pre-authorization).
- Restaurant Analytics Dashboard → Only useful at scale; restaurants just need to accept orders now.
- Account Deactivation → Low early demand; manual deletion can be handled by support.
- Restaurant Timings → For MVP, assume restaurants are always open/add timings manually.
- Eco-Friendly Delivery Options → Not essential for proving demand.
- Subscription Meal Plans → A future add-on; complex scheduling + recurring payments.
- Customer Service Chatbot → Too advanced; manual support works for MVP.
- Manage Restaurant Ratings & Reviews → Low early volume; MVP can just collect ratings.
- Admin Creates Promotions → Manual discounts can be applied at the MVP stage.
- Wishlist / Save for Later → Nice-to-have, not essential.
- Gift Orders → Edge case; keep for later.
- Tipping Delivery Partner → Good for engagement, but can wait.
- Fraud Detection & Prevention → Heavy infra + AI needed; not MVP.
- Group Order Coordination → Complex coordination; out of scope for MVP.