Project 1a1 – Problem Familiarization

CSC510: Software Engineering – Fall 2025

Zhaonan Meng, Rahmy Salman, Travis Thompson, John Stack

1. Stakeholders

A food delivery system involves some groups relying on the platform for different goals.

· **Customers** – Individuals placing food orders through the app/website.

· **Restaurant Owners/Staff** – Businesses preparing food for delivery.

· **Delivery Drivers** – Independent contractors or employees responsible for delivering food.

· **System Administrators** – Engineers and support staff maintaining the system.

· **Platform Company** – Business entity responsible for strategy, pricing, and compliance.

· **Payment Providers** – Banks and third-party payment processors handling transactions.

· **Regulators** – Government agencies overseeing labor laws, food safety, and consumer protection.

· **Customer Support Agents** – Staff resolving complaints, refunds, or delivery issues.

· **Marketing/Advertisers** – Third parties promoting restaurants and deals on the platform.

· **Data Analysts** – Teams studying user behavior, logistics, and optimization.

2. Stakeholder Biases and Conflicts

When multiple stakeholders interact, priorities may conflict. Below are five examples of clashes:

· **Customer vs. Driver** – Customers want the lowest delivery fees, but drivers want higher pay.

· **Restaurant vs. Platform Company** – Restaurants want minimal commission fees, but the platform may increase fees for revenue.

· **Driver vs. Restaurant** – Drivers prefer quick pickup, but restaurants may delay orders when busy.

· **Customer vs. Restaurant** – Customers may demand customization or rush orders, while restaurants prefer standardized, manageable workflows.

· **Platform vs. Regulators** – Platform seeks flexible labor models (e.g., gig work), while regulators may enforce stricter employment protections.

3. Reflection on Prompt Crafting

We experimented with LLM-based brainstorming to generate stakeholders, conflicts, and use cases. The results showed differences between zero-shot prompting (no context given) and careful prompting (providing structure and examples).

Zero-shot prompting:

• *Strengths:* Fast and surprising variety of ideas.
• *Weaknesses:* Inconsistent structure, missing details, and sometimes irrelevant stakeholders (e.g., 'drone operators' suggested without context).

Careful prompting:

• *Strengths:* More relevant, complete, and aligned with the project requirements.
• *Weaknesses:* Requires more effort up front to design the prompt.

Conclusion: Structured prompting improves quality and alignment with assignment goals. Zero-shot can be useful for creative brainstorming, but careful prompting is better for formal deliverables.

4. Use Cases

Below are 10 use cases for the food delivery system. Each use case includes Preconditions, Main Flow, Subflows, and Alternative Flows.

Use Case 1: Place an Order

Preconditions:

- Customer has an active account and is logged in.
- Valid payment method is stored.
- Restaurants are available in the delivery zone.

Main Flow:

1. Customer opens the app and browses restaurants.
2. System shows menus, prices, and estimated delivery times.
3. Customer selects a restaurant and adds items to the cart.
4. System updates cart with order summary and total cost.
5. Customer reviews cart and confirms delivery address.
6. Customer selects payment method.
7. Customer submits the order.
8. System processes payment and confirms the order with the restaurant.
9. System notifies customer with confirmation and estimated delivery time.

Subflows:

- Customer applies promo code or loyalty points.
- Customer customizes menu items (e.g., toppings, portion size).
- Customer schedules the order for later delivery.

Alternative Flows:

- Payment fails → Customer prompted to retry or change method.
- Item unavailable → System alerts customer and suggests replacements.
- Restaurant closed → System prompts customer to choose another restaurant.

Postconditions:

- Success: Order is confirmed, payment is processed, and restaurant receives details.
- Failure: Order is not placed; customer is notified and can retry.

Use Case 2: Accept an Order (Restaurant)

Preconditions:

- Restaurant account is active and logged into the system.
- Restaurant is marked as "open" and able to take new orders.
- Kitchen has sufficient staff and resources to prepare food.

Main Flow:

1. Restaurant receives a new order notification from the system.
2. Staff reviews the order details (items, quantities, special instructions).
3. Restaurant verifies inventory and confirms ability to prepare the order.
4. Restaurant accepts the order in the system.
5. Restaurant provides or confirms an estimated preparation time.
6. System updates the order status and notifies the customer and driver (when assigned).

Subflows:

- Restaurant adjusts the estimated preparation time (e.g., from 15 minutes to 25 minutes).
- Restaurant prints a kitchen ticket or pushes the order to its internal POS system.

Alternative Flows:

- Restaurant rejects the order:
    - System cancels the order, refunds the customer, and suggests nearby alternatives.
- Item out of stock:
    - Restaurant marks an item unavailable; system asks the customer to confirm replacement or remove it from the order.
- Unexpected kitchen delay:
    - Restaurant updates preparation time; system notifies customer and adjusts driver assignment if needed.

Postconditions:

- Success: Order is accepted, confirmed with preparation time, and queued for kitchen staff.
- Failure: Order is rejected or modified, and customer is notified with refund/alternatives.

Use Case 3: Assign Delivery Driver

Preconditions:

- Order has been accepted by the restaurant.
- Delivery drivers are registered in the system and may be available.

Main Flow:

1. System searches for available drivers near the restaurant.
2. System considers factors such as driver proximity, current workload, and performance rating.
3. System sends a delivery request to the selected driver.
4. Driver reviews order details (pickup location, delivery address, estimated earnings).
5. Driver accepts the assignment.
6. System updates the order status and notifies customer and restaurant that a driver is assigned.

Subflows:

- Multiple drivers receive the request; the first to accept is assigned.

- System may prioritize drivers who are already near the restaurant.

Alternative Flows:

- No driver accepts: System retries with a wider search radius or escalates with higher pay incentives.
- Driver declines: System reassigns request to another available driver.
- Driver app offline: System automatically removes driver from candidate list.

Postconditions:

- Success: A driver is assigned and notified of pickup.
- Failure: No driver available; system may cancel order and notify customer.

Use Case 4: Pickup Food

Preconditions:

- Restaurant has accepted and prepared the order.
- Driver has been assigned and accepted the delivery request.

Main Flow:

1. Driver arrives at the restaurant.
2. Driver identifies themselves and provides order ID to staff.
3. Restaurant staff verifies the order details and hands food to the driver.
4. Driver confirms receipt in the app.
5. System updates order status to "Out for Delivery."

Subflows:

- Driver checks packaging for accuracy (items, condiments, utensils).
- Restaurant requires driver to sign or digitally confirm pickup.

Alternative Flows:

- Order not ready: Driver waits, and preparation time is updated.
- Wrong order prepared: Restaurant corrects and re-prepares items.
- Restaurant closed unexpectedly: Driver reports issue, system cancels order, and customer refunded.

Postconditions:

- Success: Driver leaves restaurant with the correct order.
- Failure: Order not collected; system updates customer with delay or cancellation.

Use Case 5: Deliver Food

Preconditions:

- Driver has picked up the correct order.
- Customer's delivery address is valid and within service area.

Main Flow:

1. Driver navigates to the customer's delivery address using integrated GPS.
2. System provides optimized route and traffic updates.
3. Driver arrives at delivery location.
4. Driver hands the order to the customer or leaves it at the specified location.
5. Driver marks the delivery as completed in the app.
6. System updates the status and notifies the customer.

Subflows:

- Contactless delivery: Driver leaves food at the door and takes a photo as proof.
- Secure delivery: Customer provides PIN or code to verify receipt.

Alternative Flows:

- Customer unavailable: Driver calls or messages the customer. If still unreachable, driver follows retry/return protocol.
- Incorrect address: Driver contacts customer or support for clarification.
- Order damaged in transit: Driver reports issue; support arranges refund or replacement.

Postconditions:

- Success: Customer receives order, and transaction is marked complete.
- Failure: Delivery unsuccessful; order returned or refunded.

Use Case 6: Process Payment

Preconditions:

- Customer has a valid payment method (credit card, debit card, digital wallet, or gift card).
- Customer has confirmed order checkout.

Main Flow:

1. Customer submits the order at checkout.
2. System sends a payment authorization request to the payment provider.
3. Payment provider validates credentials and checks for sufficient balance.
4. Upon approval, payment is authorized and the order is confirmed.
5. After successful delivery, funds are distributed:
    - Restaurant receives portion for food.
    - Driver receives delivery fee and tip.
    - Platform retains service fee.

Subflows:

- Customer uses promotional credit or coupon, reducing payment amount.
- Split payment between multiple sources (e.g., gift card + credit card).

Alternative Flows:

- Payment declined: System notifies customer and prompts another method.
- Network error during transaction: System retries authorization or cancels order.

Postconditions:

- Success: Payment recorded, and order proceeds.
- Failure: No funds captured; order not confirmed.

Use Case 7: Handle Customer Complaint

Preconditions:

- A completed or active order exists.
- Customer has reported an issue (e.g., missing item, late delivery, incorrect food).

Main Flow:

1. Customer submits a complaint through the app or customer service.
2. System logs complaint and assigns it to a support agent.
3. Support agent reviews order details and validates the complaint.

4. Support agent decides resolution path (refund, credit, re-delivery).
5. *Customer receives notification of resolution.*

Subflows:

- Automated system checks common issues (e.g., order never delivered → instant refund).
- Customer uploads photos as evidence (e.g., damaged food).

Alternative Flows:

- Complaint dismissed: If evidence is insufficient or claim invalid, case is closed.
- Escalation: If unresolved, complaint is escalated to supervisor.

Postconditions:

- Success: Customer complaint resolved and logged.
- Failure: Customer dissatisfaction persists; negative impact on ratings.

Use Case 8: Rate Order Experience

Preconditions:

- Order was delivered and marked as complete.
- Customer has access to rating interface.

Main Flow:

1. Customer is prompted to rate food quality and delivery service.
2. Customer selects rating (e.g., 1–5 stars) and may add written feedback.
3. System stores feedback in the database.
4. Average ratings update restaurant and driver profiles.

Subflows:

- Customer uploads photos to support review.
- Customer leaves only a written review without rating.

Alternative Flows:

- Customer skips rating: Order is archived without feedback.

Postconditions:

- Success: Ratings stored and used for analytics.
- Failure: No rating submitted; order closes without feedback.

Use Case 9: Restaurant Updates Menu

Preconditions:

- Restaurant account is active and logged in.
- Staff has menu editing permissions.

Main Flow:

1. Restaurant logs into the platform.
2. Staff navigates to menu management section.
3. Restaurant adds, updates, or removes menu items.
4. System validates changes (e.g., price format, availability).
5. Approved updates appear in the customer-facing app.

Subflows:

- Restaurant adds limited-time promotional items.
- Restaurant updates preparation times for seasonal dishes.

Alternative Flows:

- Menu update rejected: If incomplete, violates policy, or contains errors, system notifies restaurant to revise.

Postconditions:

- Success: Menu changes published to app.
- Failure: Menu remains unchanged until corrected.

Use Case 10: Driver Withdraws Earnings

Preconditions:

- Driver has completed deliveries and accumulated earnings.
- Driver's bank account is linked to their profile.

Main Flow:

1. Driver logs into the app and navigates to "Earnings."
2. Driver requests payout of available balance.
3. System processes payout request.
4. Bank transfer initiated to driver's account.
5. System updates driver's account balance to reflect withdrawal.

Subflows:

- Driver requests instant payout (with additional fee).
- Driver schedules automatic weekly payout.

Alternative Flows:

- Bank rejects transfer: Driver notified to update account info.
- System error: Payout request delayed and queued for retry.

Postconditions:

- Success: Driver's bank account credited, and earnings balance updated.
- Failure: Driver notified of failure; payout not completed.