P1A1 S2G1

Authors: Robert Kemp, Willow Sapphire, Tiehang Zheng, Rishi Jeswani, Matthew Nguyen

---

🍴 Food Delivery System – Stakeholder Analysis

---

1. Customers (Primary)

- **Role:** End-users who browse, order, pay, and review.
- **Needs:** Easy ordering, fast delivery, transparent tracking, secure payment.
- **Influence:** High — their adoption and satisfaction drive revenue.
- **Concerns:** Order accuracy, food safety, delivery speed, data privacy.

---

2. Restaurants / Food Providers (Primary)

- **Role:** Prepare and supply food; manage menus and promotions.
- **Needs:** Efficient order notifications, flexible menu tools, sales analytics.
- **Influence:** High — without them, the platform has no supply.
- **Concerns:** Commission rates, fair order distribution, reliability of payments.

---

3. Restaurant Staff (Chefs, Kitchen, Front Desk) (Secondary)

- **Role:** Operational users interacting with order dashboards.
- **Needs:** Clear order details, minimal errors, manageable ticket flow.
- **Influence:** Medium — they affect fulfillment efficiency.
- **Concerns:** Confusing UI, overload during peak hours, errors in customization.

---

4. Delivery Drivers (Couriers) (Primary)

- **Role:** Independent contractors delivering food.
- **Needs:** Fair job assignment, accurate navigation, prompt payments.
- **Influence:** High — they're essential to completing transactions.
- **Concerns:** Earnings, safety, workload balance, traffic conditions.

---

5. Platform Administrators / Owners (Primary)

- **Role:** Manage system, users, disputes, compliance, financials.
- **Needs:** Monitoring tools, analytics dashboards, customer support features.
- **Influence:** High — accountable for entire ecosystem.
- **Concerns:** Scalability, legal compliance, fraud prevention, profitability.

---

6. Customer Support Representatives (Secondary)

- **Role:** Handle customer and driver disputes, refunds, complaints.
- **Needs:** Access to order history, refund tools, escalation workflows.
- **Influence:** Medium — frontline of customer experience.
- **Concerns:** Long resolution times, lack of authority, system bugs.

---

7. Payment Providers (Secondary)

- **Role:** Process payments and refunds securely.
- **Needs:** Secure API integrations, accurate transaction records.
- **Influence:** High — without smooth payments, trust is lost.
- **Concerns:** Fraud prevention, compliance (PCI DSS), chargebacks.

---

8. Marketing & Advertising Partners (Secondary)

- **Role:** Run campaigns, loyalty programs, customer acquisition.
- **Needs:** Access to customer data (within privacy limits), promo tools.
- **Influence:** Medium — drive growth, but not core operations.
- **Concerns:** ROI on campaigns, compliance with data protection laws.

---

9. Technology Vendors / Service Providers (Secondary)

(Cloud hosting, maps, SMS/email providers, etc.)

- **Role:** Provide infrastructure and third-party APIs.
- **Needs:** Stable integration, proper usage monitoring.
- **Influence:** High — system downtime directly affects service.
- **Concerns:** SLAs, overuse of APIs, vendor lock-in.

---

10. Local Governments & Regulators (External)

- **Role:** Enforce laws on food safety, taxes, gig economy regulations.
- **Needs:** Compliance reporting, tax records, health certifications.
- **Influence:** High — can impose restrictions or penalties.
- **Concerns:** Labor classification of drivers, hygiene standards, consumer protection.

---

11. Insurance Providers (External)

- **Role:** Offer coverage for drivers, restaurants, and the platform.
- **Needs:** Risk data, accident records, claims process integration.
- **Influence:** Medium — required for trust and liability coverage.
- **Concerns:** Fraudulent claims, high-risk profiles, uninsured drivers.

---

12. Investors / Business Owners (External)

- **Role:** Fund operations, expect returns.
- **Needs:** Growth metrics, financial reports, market adoption.
- **Influence:** High — funding drives scaling and expansion.
- **Concerns:** Profitability, competition, scalability.

---

13. Competitors (External)

- **Role:** Competing platforms (Uber Eats, DoorDash, Grubhub, etc.).
- **Needs:** N/A (indirect stakeholder).
- **Influence:** Medium — shape market strategy and pricing models.
- **Concerns:** Market share, customer loyalty.

---

14. Community & Local Residents (External)

- **Role:** Indirectly affected by delivery operations (traffic, safety).
- **Needs:** Reduced disruption, community benefits (local restaurant growth).
- **Influence:** Low individually, but collective perception affects reputation.
- **Concerns:** Traffic congestion, noise, gig worker treatment.

15. Loyalty & Rewards Partners (Future)

- **Role:** Credit card companies, point-based programs.
- **Needs:** Data sharing agreements, smooth redemption process.
- **Influence:** Medium — drive customer retention.
- **Concerns:** Fraud, reward abuse, low adoption.

---

16. Logistics / Fleet Management Companies (Future)

- **Role:** Provide centralized delivery fleets if scaling beyond gig drivers.
- **Needs:** API integrations, delivery tracking, bulk assignment tools.
- **Influence:** High if contracted → could replace independent drivers.
- **Concerns:** Cost-effectiveness, maintenance, coordination.

---

17. Healthcare Authorities (Future / Conditional)

- **Role:** Regulate food safety in crises (pandemics, contamination).
- **Needs:** Real-time reporting of hygiene and food handling.
- **Influence:** High during emergencies.
- **Concerns:** Public health risks, contactless delivery protocols.

⚖️ Stakeholder Biases and Collisions

In a food delivery ecosystem, different stakeholders have **conflicting interests** that may create friction. Below are key tensions to address during design and governance.

---

1. Restaurants/Businesses vs. Drivers

- **Bias/Collision:**
  - Drivers may get frustrated if delivery distances are too long or inefficiently assigned.
  - Larger restaurants may receive more visibility, while smaller businesses risk being overlooked.
- **Implications:**
  - Need for **fair job assignment algorithms** (consider distance, driver availability, restaurant priority).
  - Ensure **equitable exposure** for small and local restaurants on the platform.

---

2. Drivers vs. Customers

- **Bias/Collision:**
  - Customers may file complaints about drivers (late delivery, behavior, etc.).
  - Drivers need a way to **defend against false allegations**.
- **Implications:**
  - Establish **transparent dispute resolution** processes with evidence (e.g., GPS logs, delivery photos).
  - Maintain **balanced rating systems** to protect both drivers and customers from bias.

---

3. Admin vs. Customers/Drivers/Restaurants

- **Bias/Collision:**
  - Risk of admins or internal employees **accessing customer data unnecessarily**.
  - Admins may make changes or decisions **without permission or oversight**.
- **Implications:**
  - Enforce **role-based access control** (RBAC) to restrict sensitive data access.
  - Implement **audit trails and approval workflows** to prevent misuse of authority.

---

4. Admin vs. Employees (Internal Staff)

- **Bias/Collision:**
  - Employees may feel unable to perform tasks if the admin is too restrictive.
  - Overbearing admin policies could stifle productivity.
- **Implications:**
  - Balance **security vs. operational flexibility**.
  - Define **clear responsibilities and escalation paths** for admins and employees.

---

5. Customers vs. Restaurants

- **Bias/Collision:**
  - Customers may complain about food quality, service, or incorrect orders.
  - Restaurants need a fair way to **respond to complaints and protect their reputation**.
- **Implications:**
  - Provide **two-way review systems** (customers rate restaurants, restaurants can respond).
  - Allow **mediated dispute resolution** to avoid bias toward one side.

🍴 Food Delivery System – Formal Use Cases

UC-01: Browse and Search Restaurants

- **Actors:** Customer
- **Description:** Customer searches and browses available restaurants and menus.
- **Preconditions:** Customer is logged into the platform; internet connection available.
- **Main Flow:**
  - Customer opens the app/web platform.
  - The system displays a list of nearby restaurants.
  - Customer applies filters (e.g., cuisine, rating, delivery time).
  - The system shows an updated list with menus.
- **Subflows:**
  - Apply Filters: Customer chooses cuisine, price range, delivery time → system updates list.
  - Sort Results: Customer sorts by “highest rated,” “closest,” or “fastest delivery.”
  - View Restaurant Profile: Customer clicks a restaurant → system displays menu, ratings, operating hours.
  - Check Availability: If a restaurant is closed, system shows “Closed” with next opening time.
- **Alternate Flows:**

    - A1: No restaurants available in the area → system shows “No results found.”
    - A2: API call to external services (maps, location) fails → system shows error message.
- **Postconditions:** Customer views restaurant options and menus.

UC-02: Place a Food Order

- **Actors:** Customer, Restaurant, Payment Provider
- **Description:** Customer selects items, customizes them, and checks out.
- **Preconditions:** Customer has an active account and valid payment method.
- **Main Flow:**
    - Customer adds items to the cart.
    - Customer reviews the cart and customizations.
    - Customer selects delivery address and payment method.
    - Customer confirms and pays for the order.
    - System processes payment through provider.
    - Order request is sent to the restaurant.
- **Subflows:**
    - Customize Order: Customer adds notes (e.g., “extra cheese”) → system validates customization.
    - Apply Promo Code: Customer enters discount code → system verifies and recalculates price.
    - Confirm Address: Customer selects saved address or enters new one → system validates against delivery zone.
    - Payment Verification: Payment provider performs fraud/security checks → confirms success.
- **Alternate Flows:**
    - A1: Payment fails → system prompts for another payment method.
    - A2: Restaurant rejects order (item out of stock) → system notifies customer and refunds.
- **Postconditions:** Order and payment recorded; restaurant notified.

UC-03: Track Order in Real Time

- **Actors:** Customer, Driver, System
- **Description:** Customer monitors order status (preparing, en route, delivered).
- **Preconditions:** Customer has placed an order; driver assigned.
- **Main Flow:**
    - Customer opens order tracking page.
    - The system displays live updates from the restaurant (prep) and driver (location).
    - Customer views estimated delivery time.
- **Subflows:**
    - Preparation Updates: Restaurant updates status → system pushes notification to customer.
    - Driver Assigned: System notifies customer of driver’s name, photo, and vehicle details.
    - GPS Tracking: Driver location continuously updated → customer views ETA.
    - Delay Alerts: If driver is delayed (traffic, accident), system recalculates ETA and informs customer.
- **Alternate Flows:**
    - A1: GPS data unavailable → system shows last known location.
    - A2: Driver cancels mid-delivery → system reassigns another driver.

- **Postconditions:** Customer receives continuous updates until delivery is complete.

UC-04: Accept and Fulfill Order

- **Actors:** Restaurant Staff (kitchen/front desk)
- **Description:** Restaurant receives and prepares the order.
- **Preconditions:** Customer has placed an order; restaurant is open.
- **Main Flow:**
  - Restaurant receives new order notification.
  - Staff reviews and accepts order.
  - Kitchen prepares the order.
  - Staff marks order as “Ready for pickup.”
- **Subflows:**
  - Kitchen Queue Management: Restaurant places order into prep queue → staff sees ticket.
  - Substitute Items: If ingredient is out of stock, restaurant proposes substitute → customer accepts or declines.
  - Ready Notification: Once food is prepared, staff marks order “Ready for pickup” → system alerts driver.
- **Alternate Flows:**
  - A1: Restaurant rejects order → customer notified and refunded.
  - A2: Ingredient unavailable → restaurant suggests replacement; customer approves/denies.
- **Postconditions:** Order is prepared and marked ready.

UC-05: Assign and Dispatch Driver

- **Actors:** Driver, System, Restaurant
- **Description:** System assigns available driver to order for pickup.
- **Preconditions:** Order is marked ready; drivers are available nearby.
- **Main Flow:**
  - System identifies available drivers.
  - System sends job request to nearest driver.
  - Driver accepts job.
  - System notifies restaurant and customer.
- **Subflows:**
  - Driver Matching: System selects nearest available driver → pings them with job details.
  - Driver Acceptance: Driver has X seconds to accept/reject → system reassigns if rejected.
  - Batch Delivery: If multiple orders are nearby, system assigns multiple pickups to one driver.
  - Driver Unavailable: If no drivers accept → system alerts customer and offers order cancellation or delay.
- **Alternate Flows:**
  - A1: No drivers available → customer notified of delay/cancellation.
  - A2: Driver rejects job → system assigns another driver.
- **Postconditions:** Driver assigned; order pickup in progress.

UC-06: Pickup and Deliver Food

- **Actors:** Driver, Restaurant Staff, Customer
- **Description:** Driver picks up food and delivers it to the customer.

- **Preconditions:** Driver has accepted delivery assignment.
- **Main Flow:**
  - Driver arrives at restaurant.
  - Restaurant hands order to driver.
  - Driver updates status: “Order picked up.”
  - Driver navigates to customer address.
  - Driver delivers food and marks order as “Delivered.”
- **Subflows:**
  - Restaurant Handoff: Driver verifies order ID before collecting food.
  - Customer Contact: Driver can message/call customer for delivery instructions.
  - Proof of Delivery: Driver takes photo or requests digital signature.
  - Failed Delivery Attempt: If customer not available → driver retries or returns food to restaurant.
- **Alternate Flows:**
  - A1: Driver can’t find customer address → system provides support/call option.
  - A2: Customer unavailable → driver retries or returns food.
- **Postconditions:** Order delivered; system records completion.

UC-07: Handle Payments and Refunds

- **Actors:** Customer, Payment Provider, Admin
- **Description:** Process payments securely and handle refunds.
- **Preconditions:** Customer has a valid payment method.
- **Main Flow:**
  - Customer confirms payment.
  - System sends transaction request to payment provider.
  - Provider confirms transaction success.
  - System issues receipt.
- **Subflows:**
  - Transaction Authorization: Payment provider checks card validity and funds.
  - Split Payments: If multiple payment methods (e.g., promo + card) → system allocates correctly.
  - Refund Request: Customer requests refund → admin reviews request and issues refund.
  - Partial Refunds: If part of the order is missing, admin issues partial credit/refund.
- **Alternate Flows:**
  - A1: Payment fails → order not processed; customer prompted to retry.
  - A2: Refund requested → admin reviews and triggers refund.
- **Postconditions:** Transaction completed or refunded; records updated.

UC-08: Rate and Review Experience

- **Actors:** Customer, Driver, Restaurant
- **Description:** Customer reviews restaurant and delivery experience.
- **Preconditions:** Order is completed.
- **Main Flow:**
  - Customer prompted to leave rating and review.
  - Customer submits feedback (e.g., 1–5 stars, comments).
  - System records feedback for analytics.
- **Subflows:**
  - Driver Rating: Customer gives 1–5 stars → driver sees average rating updated.

    - Restaurant Rating: Customer leaves feedback on food quality, packaging, timeliness.
    - Reply to Review: Restaurant/driver can respond to reviews to defend reputation.
    - Review Moderation: System automatically flags inappropriate content for admin review.
- **Alternate Flows:**
    - A1: Customer skips review → no feedback recorded.
    - A2: Inappropriate content detected → system flags for moderation.
- **Postconditions:** Ratings stored and used for future recommendations.

UC-09: Manage Menus and Promotions

- **Actors:** Restaurant Owner/Manager
- **Description:** Restaurants update menus and promotional offers.
- **Preconditions:** Restaurant has a registered account.
- **Main Flow:**
    - Manager logs into dashboard.
    - Manager updates menu items and prices.
    - Manager sets promotions or discounts.
    - System updates menu in real time.
- **Subflows:**
    - Add/Edit Menu Items: Restaurant adds item with name, price, description, and photo.
    - Set Availability: Restaurant marks items as “available,” “out of stock,” or “time-limited.”
    - Promotion Creation: Restaurant sets discount or special offer → system validates compliance.
    - Approval Process: For certain promotions, admin may need to review before publishing.
- **Alternate Flows:**
    - A1: Invalid input (e.g., missing price) → system prompts correction.
    - A2: Promotion violates platform rules → admin approval required.
- **Postconditions:** Updated menus and promotions available to customers.

UC-10: Administer and Monitor Platform

- **Actors:** Platform Administrator, Regulators
- **Description:** Admin oversees platform operations and compliance.
- **Preconditions:** Admin has valid access rights.
- **Main Flow:**
    - Admin logs into platform dashboard.
    - Admin monitors orders, disputes, and performance.
    - Admin reviews flagged issues (fraud, inappropriate content).
    - Admin generates compliance reports for regulators.
- **Subflows:**
    - User Management: Admin adds, edits, or suspends accounts.
    - Fraud Detection: System flags suspicious activities → admin investigates.
    - Dispute Resolution: Admin mediates between customer and restaurant/driver, may issue refunds.
    - Compliance Reporting: Admin generates reports for regulators (taxes, food safety, data privacy).

    - Audit Logging: Every admin action (data access, account changes) is recorded for accountability.
- **Alternate Flows:**
    - A1: Regulator requests audit → admin exports required data.
    - A2: System failure → admin initiates recovery process.
- **Postconditions:** Platform continues to operate smoothly; reports stored.

Prompt Crafting Comments

Zero-Shot Prompting Example:

Prompt: "Tell me about stakeholders for a food delivery app."

LLM Output Likelihood: A generic, unstructured list (e.g., "customers, restaurants, drivers, admin"). It might miss key stakeholders like regulators or payment processors. It provides a good starting point but lacks depth and organization.

Careful Prompting Example:

Prompt: "Act as an expert business analyst. Generate a structured list of stakeholders for a new food delivery platform. Please categorize them into 'Primary', 'Secondary', and 'External' stakeholders. For each stakeholder, provide a one-sentence description of their role. Use a bulleted list format."

LLM Output Likelihood: A detailed, well-organized, and categorized list (much like the one provided in Section 1 above). The output is immediately more useful for formal documentation because it specifies the role, context, and structure required.

Comparison Comment:
Zero-shot prompting is fast and useful for initial, broad brainstorming. However, it relies on the LLM's internal biases and often produces generic results. Careful prompting, which includes role-playing ("Act as an expert..."), context ("for a new food delivery platform"), specific instructions ("categorize," "one-sentence description"), and formatting requests ("bulleted list"), yields significantly more detailed, relevant, and professionally structured outputs.

This reduces the need for extensive editing and ensures the result aligns closely with the project's specific requirements.