

CSC 510 SE Proj1a1
Section 001 - Group 29
Aditya Deshpande, Shreeya Ranwadkar, Yash Dhavale, Ravi Goparaju

TiffinTrail — Smarter routes, Warmer meals.

A) Define Stakeholders:

1. Primary Stakeholders (Direct Users)

- **Customers** - People ordering food.
- **Restaurants** - Entities responsible for providing the food (dine-in, ghost/dark kitchens, cloud brands)
- **Delivery Drivers / Riders** – delivering the food.

2. Internal Stakeholders (Within the App Organization)

- **App Owners / Founders** – decision-makers, product vision.
- **Project Managers** – coordinate tasks, timelines, features.
- **Developers (Frontend, Backend, Mobile, DevOps, QA/Testers)** – build and maintain the system.
- **Design/UI-UX Team** – ensures usability and customer experience.
- **Marketing Team** – customer acquisition, promotions, campaigns.
- **Customer Support / Helpdesk** – handling complaints, queries.
- **Admin Panel Users** – oversee orders, users, disputes, system monitoring.
- **Data Analysts / Data Scientists** – analyze trends, personalization, fraud detection.
- **Security & Compliance Team** – ensure app and payment safety, privacy compliance.
- **Finance/Accounting Team** – manages transactions, commissions, payouts to restaurants/drivers.
- **Operations / Logistics Team** – optimize delivery times, zones, and fleet management.

3. External Stakeholders

- **Investors / Shareholders** – funding and expecting ROI.
- **Payment Gateway Providers (Stripe, PayPal, UPI, etc.)** – handle secure transactions.
- **Third-party API Providers** – maps (Google Maps), push notifications (Firebase), SMS/email providers, cloud hosting, etc.
- **Regulatory Authorities / Government Bodies** – compliance with food safety, labor laws, taxation, and data protection.
- **Restaurant Aggregators / Partners** – suppliers, franchise networks.

4. Indirect / Supporting Stakeholders

- **End Customers' Families** – indirectly impacted by service quality (e.g., family dinner orders).
- **Restaurant Staff (Chefs, Managers, Cashiers)** – who process app orders internally.
- **Local Communities** – traffic, environment, waste management (sustainability concerns).
- **Competitors** – indirectly influence strategy (benchmarking, features, pricing).

B) Potential stakeholder biases:

1. Customers ↔ Delivery Drivers – Speed vs. Fair Compensation

- **Needs:** Customers want fast, cheap delivery; drivers want fair pay and realistic workloads.
- **Clash:** Customer demand for instant or low-cost delivery may pressure drivers, risking low earnings or overwork.

2. Restaurants ↔ Customers ↔ App Owner/Investors – Pricing, Profit Tension

- **Needs:** Restaurants want higher margins and menu control; customers want affordable prices and frequent promotions; app owners/investors want maximum profit and growth.
- **Clash:** Discount demands, high commission fees, or driver pay cuts can boost app profitability but reduce satisfaction and retention for restaurants, drivers, and customers.

3. Restaurants ↔ Platform Admin – Menu Control vs. Data-Driven Decisions

- **Needs:** Restaurants want autonomy over pricing, menus, and promotions; platform admins want to optimize offerings using analytics to maximize engagement and revenue.
- **Clash:** Admin-driven recommendations or algorithmic promotions may override restaurant preferences, potentially forcing menu changes or pricing adjustments that conflict with the restaurant's strategy or brand identity.

4. Development Team ↔ Marketing Team – Stability vs. Feature Push

- **Needs:** Developers want realistic feature requests, system stability, and technical feasibility; marketing wants frequent new features, promotions, and personalized options.
- **Clash:** Marketing may push for flashy features and rapid rollouts, while developers prioritize reliability, security, and maintainable code.

5. Drivers ↔ Sustainability Policies – Speed vs. Eco-Routes

- **Needs:** Drivers want fastest routes to maximize pay; platform promotes low-emission bike/electric routes.
- **Clash:** Eco-friendly routing may increase delivery time and reduce driver earnings.

C) Prompt crafting:

Prompt crafting is essentially the **art of communicating clearly with AI** so that you get the kind of response you want. The way you phrase, structure, and contextualize your prompt has a big effect on the model's output.

D) Compare zero-shot prompting to careful prompting.

Zero-shot prompting is simply asking the model to do something without examples or detailed instructions. Careful prompting is giving clear structure, examples, or constraints to guide the model's response.

When using a rapid **zero-shot prompt**, the LLM generates answers immediately without examples or detailed guidance. The output is often **generic, high-level, and sometimes repetitive**, covering obvious points but lacking depth, nuance, or creativity. For instance, asking "List stakeholder conflicts in a food delivery system" produced only predictable clashes like "speed vs. cost" or "cheap delivery vs. driver pay," which did not add much new insight.

In contrast, a **longer, guided interaction**—where we specified the format ("Stakeholders – Needs – Clash") and asked for more original perspectives—led to answers that were **sharper, more creative, and structured for direct use**. In our own session, this iterative prompting helped surface subtler tensions, such as eco-friendly route choices for drivers, algorithmic menu influence on restaurants, or late-night delivery vs. labor laws. These were far more useful for analysis compared to the zero-shot output.

Wrap-up: Zero-shot prompting worked well for **quick brainstorming**, but careful prompting clearly produced **higher-quality, professional results**. From our own experience, the best approach was to start with a zero-shot run for breadth, then refine it through guided prompts to achieve depth, consistency, and report-ready clarity.

E) Use Cases for Food Delivery App

1) User Registration

a) Preconditions:

- The user has installed the app.
- The user has internet access.

b) Main Flow:

1. The user opens the app and selects "Sign Up."
2. System prompts for details (name, email, phone, password).
3. The user enters details and submits.
4. System validates data format.
5. The system sends OTP (One-Time Password) for verification.
6. The user enters OTP.
7. The system confirms registration and creates a new account.

c) Subflows:

- If social login (Google/Apple/Facebook) is selected, the system fetches details from the provider and skips manual entry.
- The user adds optional details (name, profile picture).

d) Alternative Flows:

- Invalid OTP → System notifies user, allows retry.
- Email/phone already registered → System prompts to log in instead.
- Network issue → Registration is paused, with retry option.

2) User Login

a) Preconditions:

- Users are already registered.

b) Main Flow:

1. The user opens the app, selects "Login."
2. The user enters email/phone and password.
3. System verifies credentials.
4. If valid, the system grants access to app features.

c) Subflows:

- Biometric login (Face ID / Fingerprint) available if enabled.

d) Alternative Flows:

- Wrong password → System shows error, offers "Forgot Password."
- Account locked after multiple failed attempts → User must verify identity.

3) Browse Restaurants

a) Preconditions:

- User logged in.
- GPS/location enabled OR delivery address saved.

b) Main Flow:

1. The user selects "Browse Restaurants."
2. The system fetches a restaurant list for the location.
3. The user applies filters (cuisine, rating, delivery time, offers).
4. The system displays results with restaurant details and menus.

c) Subflows:

- User taps on a restaurant → Menu is displayed.
- Users add restaurants to favorites for later.

d) Alternative Flows:

- No restaurants available in the area → System shows "Not Available" message.
- API failure → System retries or shows cached results.

4) Add Items to Cart

a) Preconditions:

- User logged in.
- Restaurant selected.

b) Main Flow:

1. Users browse menu items.
2. The user selects an item and chooses quantity/customization
3. The user taps "Add to Cart."
4. System updates cart and displays confirmation.

c) Subflows:

- User edits cart (increase/decrease quantity).
- The user adds special instructions (e.g., "No onions").

d) Alternative Flows:

- Item unavailable → System notifies user and prevents adding.
- Restaurant closed → User cannot add items.

5) Place an Order

a) Preconditions:

- User logged in.
- Items added to cart.

b) Main Flow:

1. The user proceeds to checkout.
2. The system displays delivery address and payment options.
3. The user selects address and payment method.
4. System confirms order summary.
5. The user clicks "Place Order."
6. The system creates order, forwards to the restaurant.
7. Payment processed via gateway.
8. Confirmation sent to the user.

c) Subflows:

- Apply promo code → System validates and adjusts price.
- The user adds a tip for the driver.

d) Alternative Flows:

- Payment fails → System allows retry or choose different method.
- Restaurant rejects order → Refund initiated, user notified.

6) Order Tracking

a) Preconditions:

- Order successfully placed.

b) Main Flow:

1. Customer navigates to "My Orders."
2. The system shows order status (accepted, preparing, out for delivery).
3. When a driver is assigned, the system shows driver details and live location.
4. Notifications sent at each status update.

c) Subflows:

- The user contacts the driver via call/message.
- The user contacts customer support from the tracking screen.

d) Alternative Flows:

- GPS not available → System shows approximate delivery time.
- Driver reassigned due to cancellation → System updates customer.

7) Delivery Driver Accepts Order

a) Preconditions:

- Driver logged into driver app.

b) Main Flow:

1. The system pushes available delivery requests to drivers nearby.
2. The driver receives an order request with details (pickup, drop-off, earnings).
3. The driver accepts the order.
4. System confirms assignment.

c) Subflows:

- Driver rejects order → System offers it to another driver.

d) Alternative Flows:

- No drivers available → System notifies restaurant and customer.
- Driver accepts but cancels later → Reassignment triggered.

8) Payment Processing

a) Preconditions:

- User at checkout.

b) Main Flow:

1. The user selects the payment method (credit card, wallet, UPI).
2. System redirects to secure payment gateway.
3. The user authenticates payment.
4. Gateway confirms success.
5. The system generates payment receipts and sends them to finance.

c) Subflows:

- Wallet payment → Balance deducted directly.
- Cash on delivery → System marks as pending until driver collects.

d) Alternative Flows:

- Payment gateway timeout → Retry option.
- Insufficient funds → System prompts users to choose another method.

9) User Re-orders from a Past Order

a) Preconditions:

- The user has a completed order in their history.
- The items from the past order are still available on the restaurant's menu.

b) Main Flow:

1. The user navigates to their "Order History" or "Past Orders" section.
2. The user selects a past order they wish to re-order.
3. The user taps a "Re-order" or "Order Again" button.
4. The app adds all the items from that past order to the user's current cart.
5. The user is taken to the checkout screen to review and confirm the order.

c) Subflows:

- Editing the Re-order: Before checking out, the user can remove or add items to the cart that was populated from the re-order.

d) Alternative Flows:

- Item Unavailable: An item from the past order is no longer available on the menu. The app notifies the user that the item is out of stock and removes it from the cart. The user can then proceed with the rest of the order or cancel.

10) User Writes a Restaurant Review and Rating

a) Preconditions:

- The user has completed and received an order from a restaurant.
- The review period for the order is still open (e.g., within 7 days of delivery).

b) Main Flow:

1. The app sends a push notification prompting the user to review their recent order.
2. The user navigates to their past orders.
3. The user taps on the "Rate & Review" button for a specific order.
4. The user rates the restaurant on a scale (e.g., 1-5 stars).
5. The user writes a text review in a provided field.
6. The user submits the review.
7. The app confirms the submission.

c) Subflows:

- **Rating Food and Delivery Separately:** The app allows the user to rate the quality of the food and the delivery service separately.

d) Alternative Flows:

- **Review is Flagged:** The review contains inappropriate language. The app's moderation system flags the review and prevents it from being published. The user receives a message that the review violates policy.

11) User Requests Customer Support

a) Preconditions:

- The user has a past or current order.
- The user has encountered an issue (e.g., missing item, late delivery, cold food).

b) Main Flow:

1. The user navigates to the "Help" or "Support" section of the app.
2. The user selects the order they have an issue with.
3. The user chooses a reason for the issue from a list (e.g., "Missing item," "Incorrect order," "Late delivery").
4. The user writes a detailed description of the problem.
5. The user submits the support ticket.
6. The app confirms the ticket has been submitted and provides a reference number.

c) Subflows:

- The user can initiate a live chat with a support agent.
- The user can attach a photo of the incorrect or damaged order.

d) Alternative Flows:

- For common issues like a missing drink, the app's system may automatically process a partial refund without a human agent. The user is notified of the refund confirmation.

12) User Searches for a Specific Restaurant or Cuisine

a) Preconditions:

- The app is open.
- The user has set a valid delivery address.

b) Main Flow:

1. The user taps on the search bar.
2. The user types in the name of a restaurant (e.g., "Taco Bell") or a type of cuisine (e.g., "Pizza").
3. The app displays search results that match the query, filtering by restaurants that deliver to the user's address.
4. The user taps on a search result to view the restaurant's page.

c) Subflows:

- **Filtering Results:** The user can filter the search results by factors like price, delivery time, or ratings.

d) Alternative Flows:

- **No Search Results:** The search query does not match any restaurants that deliver to the user's location. The app displays a "No Results Found" message and suggests a broader search or different cuisines.