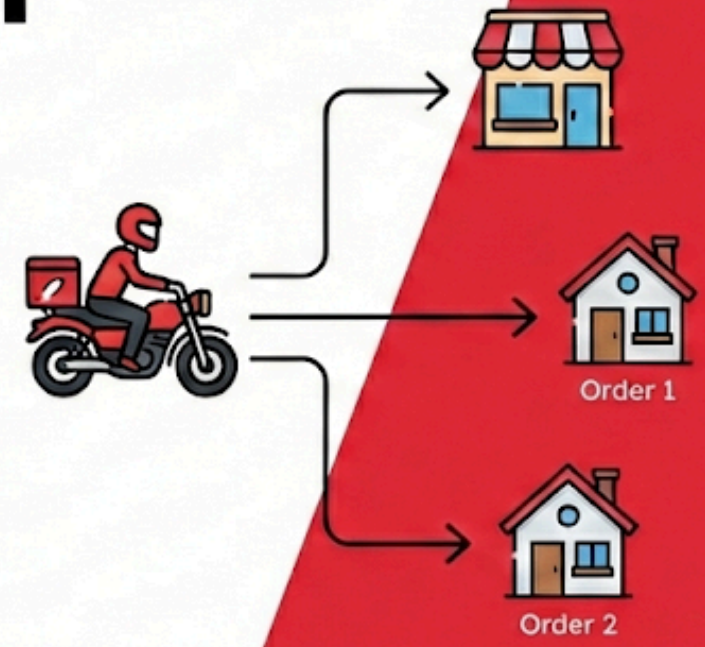
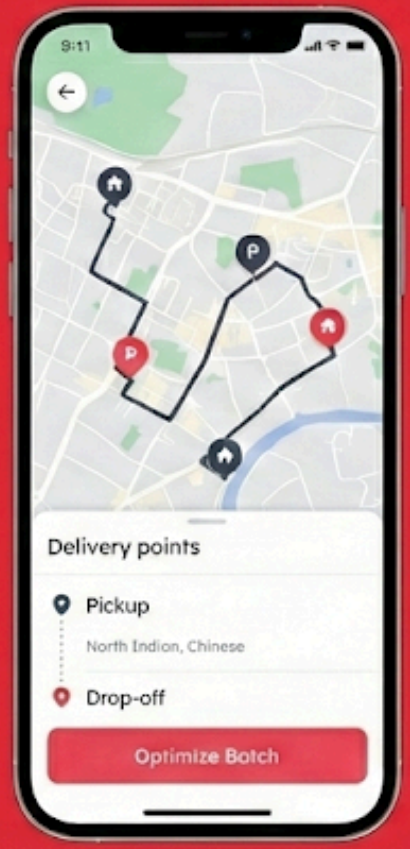
PRD: Intelligent Multi-Order Batching

Optimizing Delivery Efficiency & Costs



zomato



project Name: Intelligent Multi-Order Batching

Version: 1.0

Status: Draft for Review

Author: Aishwarya – MBA Class of 2027

Date: January 01, 2026

---

1. Executive Summary

Context: Currently, Zomato's logistics cost is the primary drag on unit economics. While we have basic batching (1 Restaurant to 2 Customers), we are missing significant efficiencies in "**Multi-Pick, Multi-Drop" (MPMD)** scenarios. Riders often travel empty or with suboptimal capacity while passing nearby restaurants with pending orders.

The Opportunity: By enabling advanced batching algorithms that group orders from different restaurants (R1, R2) to different customers (C1, C2) based on route proximity, we can significantly increase **Rider Utilization** and reduce **Cost Per Delivery (CPD)**.

Objective: To implement a dynamic batching engine that bundles orders O1 and O2 if the incremental distance/time fits within a strictly defined "Freshness Buffer," ensuring operational efficiency without eroding the Net Promoter Score (NPS).

---

2. Problem Statement

- ➢ **Inefficiency:** Riders currently treat most orders as point-to-point (P2P), leading to high "dead miles" (empty runs).
- ➢ **Cost:** The Cost Per Delivery (CPD) remains high because we pay minimum guarantees even during peak demand if routing isn't optimized.
- ➢ **Scalability:** During peak hours (Dinner: 8 PM - 10 PM), order velocity exceeds rider supply. We need to do more with fewer riders.

---

3. Target Audience & Persona

1. **The Cost-Conscious Customer:** Willing to tolerate a minor delay (5-8 mins) if it keeps delivery fees low.
2. **The Delivery Partner (Rider):** Wants to maximize earnings per hour. Batching allows them to earn more per trip (via "Batch Incentives") with less riding distance.
3. **The Restaurant Partner:** Wants food picked up before it gets cold.

---

4. User Scenarios (The "3 Dimensions of Batching")

We are solving for the **Vehicle Routing Problem (VRP)** across three specific complexities:

Scenario A (Current capability):

- **Logic:** Same Restaurant -> Different Customers (C1, C2).
- **Constraint:** C1 and C2 must be within a 30 degrees cone of the rider's direction.

Scenario B (New Feature):

- **Logic:** Different Restaurants (R1, R2) -> Same Destination (C1).

- **Use Case:** A user orders a main course from *Restaurant A* and dessert from *Restaurant B* (nearby).
- **Execution:** Rider picks up R1 -> picks up R2 -> delivers to C1.

Scenario C (New Feature):

- **Logic:** Different Restaurants (R1, R2) ->Different Customers (C1, C2).
- **Execution:** Rider route: R1 -> R2 -> C1 -> C2 (or R1 -> C1 -> R2 -> C2 depending on vector).
- **Constraint:** The ***Detour Index*** must be <**X%**

---

5. Functional Requirements

5.1. The Batching Algorithm (Backend)

The system must calculate a **"Feasibility Score"** for every potential batch based on:

1. **Prep Time Sync:** PrepTime(R1) - PrepTime(R2) <= 7 mins .
    - *Rationale:* Rider shouldn't wait at R2 while food from R1 gets cold.
2. **Detour Logic:**
    - Total Distance of batch < (distance_order1 + distance_order2) *
    - *Goal:* Ensure the batch saves at least 25% travel distance compared to two separate riders.
3. **SLA Guardrails:** The estimated arrival time (ETA) for the *first* order must not increase by more than 8 minutes due to the batch.

5.2. Rider App Experience

- **Offer Screen:** Rider sees "Batched Order" tag with total earnings (e.g., "₹80 for 2 orders").

- **Navigation:** "Smart Route" implementation. The app strictly enforces the sequence. The rider cannot jump to Deliver C1 before Picking R2 if the algorithm dictates otherwise.
- **Exceptions:** "Unbatch Button" – If R2 is delayed by >10 mins, riders can request to un-batch and deliver O1 immediately.

5.3. Customer App Experience

Gamification: Offer "Green Delivery" option. "Select 'No Rush' to save ₹10 and allow us to batch your order to save fuel."

---

6. Non-Functional Requirements

- **Latency:** The batching decision must happen in **200ms** post-order placement.
- **Scalability:** The engine must handle 5000 orders/minute during peak times (IPL finals or New Year's Eve).
- **Fail-safe:** If the algorithm fails or hangs, default to Single Order assignment immediately.

---

7. Metrics & Success Indicators (KPIs)

To measure the success of Project Synapse, we will track:

Metric Category	Metric Name	Target
North Star	**Batch Rate%** (**Shows how many % of orders we are able to batch*)	Increase from 12% to **25%**
Financial	**Cost Per Delivery (CPD)**	Reduce by **12%**
Operational	**Rider Utilization**	Increase active time/hour by **15%**
Guardrail	**DOT (Delivery On Time)**	Maintain > **92%**
Customer	**CSAT / NPS** *(*customer satisfaction score*)	No drop > 0.5 points

8. Go-to-Market Strategy (Rollout Plan)

Phase 1: Pilot (Alpha)

- **Location:** Gurgaon (Sec 29 & DLF Cyber Hub). High density of restaurants and corporate orders.
- **Users:** Top 5% rated riders only (to minimize human error).

Phase 2: Data Calibration

- Analyze "Food Freshness" complaints. Adjust the *Prep Time Sync* window if necessary.

Phase 3: Pan-India Rollout

- Enable specifically during Peak Hours (Lunch: 1-3 PM, Dinner: 8-10 PM).

---

9. Risks & Mitigation

Risk	Impact	Mitigation Strategy
Rider Confusion	High	Riders might mix up orders (give O1 to C2). **Solution:** Color-coded order IDs on the app (e.g., "Deliver RED Order").
The "Cold Pizza" Effect	High	Customer 2 gets cold food because Rider waited at Restaurant 2. **Solution:** Use thermal tags in the backend; only batch "High Heat Retention" items (e.g., Biryani) initially, avoid Ice Creams.
Algorithmic Bias	Medium	Outskirt locations might never get batched, increasing wait times for riders in those zones. **Solution:** Surge pay for unbatched zones.

Next Steps

This feature moves Zomato from a **Service-First** model to an **Efficiency-First** model without compromising the core value proposition. By optimizing the "**Traveling Salesman Problem**" in real-time, we unlock margin that can be reinvested into customer acquisition.

Immediate Next Step: Approval required for engineering bandwidth to build the *Prep-Time Prediction Model* v2.0, which is the backbone of this PRD.