Zomato’s Dynamic Pricing Transparency

Author: Jateen Patil
Version: 1.0 — February 2026
Platform: iOS & Android (Zomato Consumer App)
Market: India — Tier 1 & Tier 2 cities (Phase 1)
Launch Target: Q3 2026

Table of Contents

1.Executive Summary & Market Opportunity

Food delivery in India is a Rs. 60,000 crore market growing at 18% YoY — and Zomato sits at the centre of it with over 100 million transacting users. But growth at this scale brings a problem that most product teams underestimate: **trust erosion through pricing opacity.**

Surge pricing is economically necessary. It balances supply and demand in real time, ensures delivery partners stay active during difficult conditions, and keeps the platform functional during peak load. The issue is not the pricing model — it is the silence around it. When a user's Rs. 30 delivery fee becomes Rs. 95 on a rainy Tuesday evening, and the app offers no explanation, that user does not conclude there is a logical reason. They conclude they are being taken advantage of.

This document defines the product requirements for the **Dynamic Pricing Transparency Dashboard** — a feature that surfaces real-time, ML-attributed explanations for elevated pricing directly inside the Zomato cart and checkout experience.

The market opportunity is significant:

Signal	Data Point
Zomato MAUs (2025)	~100 million
Orders affected by surge pricing	~22% of all orders
Avg. cart value during surge	Rs. 420
Cart abandonment uplift during surge	29% above baseline
Revenue lost to surge abandonment (est.)	Rs. 180–220 crore/year
Pricing-related CS tickets (monthly)	~4.2 lakh
Cost per CS ticket (fully loaded)	Rs. 85–110

Even a 25% reduction in surge abandonment and a 40% drop in pricing complaints would recover an estimated **Rs. 60–80 crore annually** in lost GMV and support costs — making this one of the highest-ROI trust investments Zomato can make in FY26.

2.Problem Statement

What is happening today

Zomato shows users a fee breakdown at checkout — delivery fee, platform fee, GST — but never explains *why* those fees are elevated on a given occasion. A small banner saying *"High demand in your area"* occasionally appears, but it is vague, unsourced, and does nothing to reduce the frustration of a user staring at a Rs. 95 delivery fee for a 3 km order.

The current experience creates three compounding harms:

1. Immediate abandonment — Users who encounter an unexplained fee spike abandon their carts at a 29% higher rate than during non-surge windows. Many switch to Swiggy, not knowing Swiggy is equally surged at the same moment.

2. Support volume spike — Pricing confusion is the #1 driver of inbound CS contacts during peak windows. Each contact costs Rs. 85–110 to resolve and rarely ends with a satisfied user.

3. Long-term trust damage — First-time surge experiences with no explanation reduce 30-day repeat order probability by 18%. For a platform that spends Rs. 180–250 in CAC per new user, losing them after one confusing pricing moment is an expensive problem.

Why this hasn't been solved

The challenge is not data — Zomato has all the signals needed to explain pricing in real time. The challenge is that no one has built the layer that translates those signals into user-facing language. This PRD defines exactly that layer.

The hypothesis

Users who understand *why* prices are higher are significantly less likely to cancel, less likely to complain, and more likely to return — even if they ultimately choose not to order right now.

3.User Personas

Persona 1 — Riya Sharma, 26 | Software Engineer, Mumbai

> *"I don't mind paying more sometimes. I just want to know why. Is it because of the rain? Tell me that. I can handle that."*

Riya orders dinner through Zomato 4–5 times a week, usually between 8 and 10pm after work. She is a Gold member and considers herself a loyal user. On rainy evenings, she has noticed her delivery fee jumping from Rs. 30 to Rs. 85–95, often with no warning until she reaches checkout. Her first instinct is to close Zomato and check Swiggy — not because Swiggy is cheaper, but because she feels Zomato is being sneaky about it.

She does not file complaints. She just quietly switches. And she tells her friends.

Behavioral pattern: Comparison shops across platforms during surge; app-hops without realising both platforms are surged simultaneously. **What she needs:** A one-line honest reason at the moment the elevated fee appears — before she hits checkout.

Persona 2 — Arjun Mehta, 34 | Product Manager, Bengaluru

> *"My wife and I ordered for the family on a Saturday night. The cart was Rs. 980. At checkout it was Rs. 1,260. I thought it was a bug. I reported it. Nothing happened. We just stopped using Zomato for a month."*

Arjun is deliberate about spending. He plans family meals on weekends, usually ordering from 2–3 restaurants for a combined Rs. 900–1,200 cart. He is tech-savvy — he actually understands surge pricing conceptually — but he has zero tolerance for surprise fees at the final step of checkout. The Rs. 280 difference felt like deception, not economics.

He filed a complaint and got a generic response. He gave Zomato a 2-star review. He returned a month later only because his kids wanted a specific restaurant that wasn't on Swiggy.

Behavioural pattern: High cart value, weekend orders; price delta feels larger and more personal at his spend level. **What he needs:** Upfront visibility of surge before he builds his cart — not a surprise at checkout.

Persona 3 — Meera Iyer, 54 | Homemaker, Pune

> *"My son set up the app for me during COVID. I like it, but sometimes the price changes and I don't understand why. I just feel like something is wrong, so I cancel."*

Meera orders 2–3 times a month, mostly lunch for herself and her husband. She uses the app in Marathi when possible. She does not understand the difference between delivery fee, platform fee, and surge — it all looks like one confusing number to her. When that number changes unexpectedly, she cancels because she assumes something has gone wrong technically, or worse, that she is being overcharged.

She doesn't complain. She doesn't switch apps. She just calls her son and says she'll cook instead.

Behavioural pattern: Low frequency, high trust requirement; any friction causes permanent drop-off for weeks. **What she needs:** A simple, reassuring explanation in Marathi — one sentence, no jargon — that tells her everything is normal and here's why.

4.User Stories

Consumer Stories

1. As a user, I want to see a clear one-line explanation of why my delivery fee is higher right now — before I reach the final checkout screen — so I can make an informed decision about whether to order now or wait.
2. As a user, I want a visual demand indicator (Low / Medium / High) on the cart screen so I can immediately gauge how busy things are without reading any text.
3. As a user, I want a *"Schedule for Later"* prompt when surge is active, showing me an estimated lower-fee time window, so I have a real alternative to cancelling outright.
4. As a user, I want the pricing explanation in my preferred language — Hindi, Marathi, Kannada, Tamil — so I understand it without needing help.
5. As a user, I want to be able to tap on the surge indicator to learn more, so I can satisfy my curiosity without the explanation being forced on me.
6. As a user, I want a subtle notification when surge pricing ends for my usual order zone, so I can reorder at the normal fee without having to keep checking.

Operations & Analytics Stories

1. As a data analyst, I want event-level A/B test data on cart abandonment for users who see vs. don't see the transparency module, so I can measure the causal impact of the feature precisely.
2. As a product manager, I want a real-time audit dashboard showing ML signal attribution accuracy vs. actual pricing drivers, so I can monitor model health and trigger retraining when needed.
3. As a customer support lead, I want pricing-related ticket volume to be tagged by whether the user saw the transparency module before contacting us, so I can track deflection rate.

5.Feature Design & UI/UX Flows

5.1 Entry Points

The transparency module appears in three places — each non-intrusive and contextually appropriate:

Screen	Placement	Behaviour
Restaurant listing	Colour-coded delivery fee badge on restaurant card (🟢 Normal / 🟡 Moderate / 🔴 High)	Static indicator — no interaction required
Cart screen	Inline surge chip next to the delivery fee line item: *"High demand · Why?"*	Tap expands a 2-line explanation inline
Checkout screen	Expandable accordion below the price summary: *"Why is my delivery fee higher today?"*	Collapsed by default; one tap to expand
Post-cancellation screen	Full contextual card with demand explanation + Schedule for Later CTA	Shown automatically after user cancels during a surge window

5.2 UI States

State 1 — No surge active The module is completely hidden. No badges, no chips, no mentions of pricing. The experience is identical to today.

State 2 — Moderate surge (1.3x–1.8x) A yellow chip appears on the cart screen: *"Busy right now · Tap to see why"*. Tapping expands a 2-line explanation. The Schedule for Later CTA is shown if demand is projected to normalise within 90 minutes.

State 3 — High surge (>1.8x) An orange/red chip appears more prominently. The accordion on the checkout screen is pre-expanded (not collapsed). Schedule for Later is shown with a specific time and estimated fee.

State 4 — Signal data unavailable The module is hidden entirely. No broken state, no placeholder, no generic "we don't know" messaging. Graceful silent degradation.

State 5 — User taps "Why?" A bottom sheet slides up with:

- The top 2 contributing signals in plain language
- A small icon per signal (rain cloud, clock, location pin etc.)
- An estimated time until pricing normalises (if available)
- The Schedule for Later CTA

5.3 Signal → Copy Mapping

Signal	Trigger Threshold	User-Facing Copy
Heavy rain	Rainfall > 5mm/hr in delivery zone	*It's raining heavily — fewer delivery partners are available right now.*
Peak hour	12–2pm or 7:30–9:30pm window	*Peak meal-time rush — delivery fees are higher for the next ~45 minutes.*
High demand volume	> 2x baseline orders in last 15 min	*A lot of people are ordering right now — demand is unusually high in your area.*
Low delivery supply	Active partners < 40% of zone baseline	*Fewer delivery partners are online in your area at the moment.*
Festival / event	Calendar flag + local order spike > 1.5x	*High demand due to a local event or festival in your area.*
Restaurant overload	Restaurant accept rate < 60%	*Your restaurant is very busy — this may slightly affect delivery time and fee.*

Only the **top 2 signals by contribution weight** are shown to the user at any time. Showing more increases cognitive load without meaningfully improving comprehension.

5.4 Schedule for Later — UX Flow

User reaches checkout during surge
↓
Surge module visible (State 2 or 3)
↓
User taps "Schedule for Rs.[X] less"
↓
Pre-filled scheduling screen opens
→ Nearest 30-min lower-fee slot pre-selected
→ Estimated fee shown as range (e.g. Rs. 28–35)

→ User confirms or picks a different slot
↓
Order scheduled → confirmation screen
→ Surge end notification opt-in offered

6.Technical Requirements

6.1 Architecture Overview

The feature is built on three layers:

- **Signal Aggregation Service** — ingests real-time weather, demand, and supply data per delivery zone
- **Pricing Context ML Model** — ranks contributing signals, generates explanation metadata, estimates normalisation time
- **Frontend Transparency Module** — renders explanation in the appropriate UI state; lazy-loads after primary cart render

All three communicate via an event-driven pipeline. The ML model output is cached at the edge (Redis, 90-second TTL) so the app fetches a pre-computed payload rather than triggering inference on every cart load.

6.2 Signal Aggregation Service

Req ID	Signal Source	Specification
PTD-001	Weather API (IMD / OpenWeather)	Poll every 5 min per delivery zone; ingest rainfall mm/hr, wind speed, visibility index
PTD-002	Order Volume Stream	Real-time order event stream; compute rolling 15-min order count vs. 30-day zone baseline
PTD-003	Delivery Partner Supply	Active partner GPS heartbeat; zone-level online count vs. 30-day baseline per 5-min window
PTD-004	Restaurant Accept Rate	Accept/reject ratio per restaurant; 10-min rolling window; flag if < 60%
PTD-005	Event / Holiday Calendar	Pre-loaded national + regional calendar; cross-referenced with order spike detection logic

6.3 Pricing Context ML Model

Req ID	Specification
PTD-006	Model ranks contributing signals by weight; returns top 2 with confidence scores and user_copy_key
PTD-007	Model output includes: signal_id, confidence_score, user_copy_key, estimated_normalisation_minutes
PTD-008	Inference latency < 200ms P95
PTD-009	If confidence < 70% on all signals, serve fallback: *"High demand in your area right now"*
PTD-010	Model retrained weekly on signal-vs-actual-multiplier pairs via automated CI/CD pipeline
PTD-011	Retraining triggered automatically if explanation accuracy drops below 88% in weekly audit

6.4 Frontend Module

Req ID	Specification
PTD-012	Module lazy-loads after primary cart content renders — must not block the Order CTA button
PTD-013	Copy strings fetched from localisation service; language set by user's app language preference
PTD-014	Supported languages at launch: English, Hindi, Tamil, Telugu, Kannada, Bengali, Marathi
PTD-015	If signal data unavailable, hide module entirely — no empty states, no error copy
PTD-016	Schedule for Later CTA links directly to pre-filled scheduling screen with slot pre-selected
PTD-017	Bottom sheet must animate in ≤ 250ms; no jank on mid-range Android devices (tested on Redmi Note 12)
PTD-018	WCAG 2.1 AA compliant; full TalkBack (Android) and VoiceOver (iOS) support
PTD-019	Module must not increase checkout screen load time by more than 120ms

6.5 Schedule for Later — Accuracy SLA

- Projected lower-fee time window must be within ± Rs. 15 of actual fee at time of dispatch in > 85% of cases
- Fee shown to user as a range (e.g. *Rs. 28–40*), never a fixed value, to manage expectations
- If demand forecast confidence is low, Schedule for Later CTA is hidden entirely rather than shown with a potentially inaccurate estimate

---

7.Success Metrics & A/B Testing Plan

7.1 Primary KPIs

Metric	Baseline	3-Month Target	6-Month Target	Data Source
Cart abandonment rate (surge periods)	29% above baseline	–15%	–25%	App analytics
Pricing-related CS tickets (monthly)	~4.2 lakh	–25%	–40%	CS platform
30-day repeat order rate (surge-exposed cohort)	Baseline cohort	+8%	+15%	CRM cohort analysis
Schedule for Later conversion rate	N/A	> 12%	> 18%	App analytics
ML explanation accuracy	N/A	> 88%	> 92%	ML audit dashboard
App store rating	4.2	No regression	+0.2	Play Store / App Store

7.2 Guardrail Metrics

These must not regress at any point after launch. If any guardrail is breached, a feature flag rollback is triggered automatically.

- Checkout screen load time increase: **≤ 120ms**
- ML false attribution rate: **≤ 8%** (audited weekly via sampling)
- Schedule for Later fee accuracy SLA: **≥ 85% within ± Rs. 15**
- Battery impact: **≤ 1.5% additional drain per hour** during active app session

7.3 A/B Testing Plan

Experiment 1 — Core transparency module (Primary)

	Control	Treatment
Description	Current experience — no surge explanation	Transparency module visible during surge
Sample size	500,000 users per arm (surge-period eligible)	500,000 users per arm
Duration	4 weeks	4 weeks
Primary metric	Cart abandonment rate during surge	Cart abandonment rate during surge
Secondary metrics	CS ticket rate, 30-day repeat order rate	CS ticket rate, 30-day repeat order rate
MDE	5% relative reduction in abandonment	—

Experiment 2 — Schedule for Later placement

Variant	Description
A	Schedule for Later shown inline on cart screen
B	Schedule for Later shown only in bottom sheet after user taps "Why?"
C	Schedule for Later shown on post-cancellation screen only

Metric: Schedule for Later conversion rate + downstream order completion rate.

Experiment 3 — Copy tone

Variant	Description
A	Factual: *"It's raining heavily — fewer partners available."*
B	Empathetic: *"We know this isn't ideal — here's what's happening right now."*

Metric: Bottom sheet engagement rate + "Order now anyway" vs. Schedule for Later split.

8.Competitive Analysis

No Indian food delivery platform currently offers a data-backed, real-time explanation for surge pricing. Internationally, only Uber Eats has experimented with demand transparency messaging — and only in select US markets, without ML-attributed copy.

Feature	Zomato (Today)	Swiggy (Today)	Uber Eats (US)	This PRD
Fee breakdown shown	✅ Yes	❌ No	✅ Yes	✅ Yes
Surge notification	🟡 Vague banner	❌ No	✅ Yes	✅ Yes
Real-time signal explanation	❌ No	❌ No	❌ No	✅ Yes
ML-attributed copy	❌ No	❌ No	❌ No	✅ Yes
Schedule for lower price	🟡 Pre-order only	❌ No	❌ No	✅ Yes
Multilingual support	❌ No	❌ No	❌ No	✅ Yes (7 languages)
Post-cancellation recovery	❌ No	❌ No	❌ No	✅ Yes

Strategic implication: Shipping this feature makes Zomato the most transparent food delivery platform in the world on pricing — a genuinely defensible trust differentiator that Swiggy cannot match without significant infrastructure investment. It should be announced publicly and highlighted in the app store listing.

9.Stakeholder Map & RACI

Stakeholder	Role	R	A	C	I
Jateen Patil (Author)	Product Manager — Consumer Experience	✅	✅		
Engineering Lead — Consumer App	Mobile & backend implementation	✅			
ML Platform Team	Signal model, inference pipeline, retraining	✅		✅	

Design Lead	UI states, copy, bottom sheet, iconography	✅		✅	
Data Analytics	A/B test design, KPI tracking, ML audit			✅	✅
Customer Support Lead	CS ticket tracking, deflection measurement			✅	✅
Legal & Compliance	PDPA review, pricing disclosure obligations			✅	
Localisation Team	Translation of all 7 copy strings × 7 languages	✅			
Marketing	Public launch comms, app store listing update				✅
Finance	Revenue impact modelling, GMV recovery tracking			✅	✅

R = Responsible · **A** = Accountable · **C** = Consulted · **I** = Informed

---

10.Risks & Mitigations

Risk	Likelihood	Impact	Mitigation
ML model attributes wrong primary signal	Medium	High — user trusts incorrect explanation; backlash if exposed	Weekly retraining; confidence threshold fallback; audit sampling with manual review
Schedule for Later projected fee is wrong	Medium	High — user feels deceived when actual fee differs	Show fee as range not fixed; enforce ± Rs.15 SLA guardrail; hide CTA if forecast confidence low
Feature increases checkout cognitive load	Low	Medium — slower checkout, lower conversion	Lazy load; collapsed by default; A/B test placement before full rollout
Users demand price cuts, not just explanation	High	Medium — PR risk if feature is seen as deflection	Pair launch with a visible surge cap commitment; frame as Step 1 of a broader pricing transparency journey

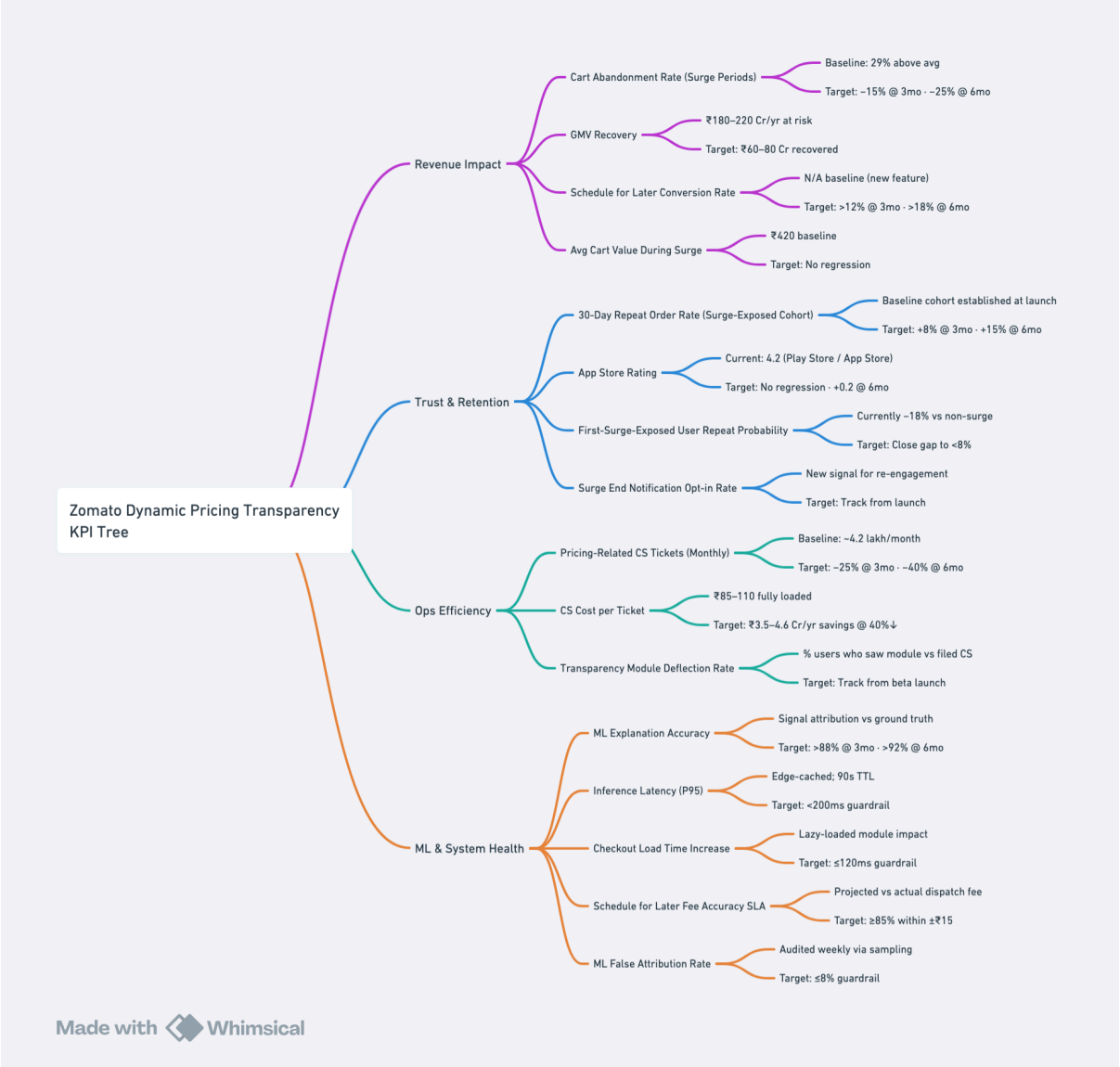
Localisation quality is poor	Medium	Medium — Meera-type users (Persona 3) disengage	All translations reviewed by native speakers, not just machine-translated; regional user testing in beta
Data privacy — zone-level location correlation	Low	High — PDPA compliance risk	Use zone-level aggregates only, never individual GPS; legal review mandatory before launch

---

11.Launch Plan

Phase	Timeline	Scope
Alpha	Q2 2026 — 6 weeks	500 internal Zomato employees in Mumbai & Bengaluru. Validate ML signal accuracy against ground truth. No external users.
Closed Beta	Q2 2026 — 8 weeks	50,000 users across Mumbai, Delhi, Bengaluru, Hyderabad. Run Experiment 1 (core module A/B). Tune copy thresholds. Localisation live for Hindi + English only.
Soft Launch	Q3 2026 — 4 weeks	Top 10 Indian cities. 20% traffic ramp. Monitor all KPIs and CS ticket volume in real time. All 7 languages live. Incident response team on standby.
Full India Rollout	Q3 2026	100% traffic across all cities. Schedule for Later fully enabled. Public announcement + app store listing update highlighting transparency commitment.
Phase 2 — Blinkit	Q1 2027	Extend feature to Blinkit quick-commerce. Adapt signal model for dark store supply dynamics — different from restaurant + delivery partner model.

KPI Tree:



PRD authored by Jateen Patil — Zomato