

Product Requirements Document (PRD)

Feature: Live Partner Availability at Checkout

Product: **Blitz** (same-day delivery) | Author: **Sidharth Nair** | Status: Draft — for review | Last updated: 12th Aug 2026

Why? (Objective)

1. For business. Checkout today shows nothing about fulfillment feasibility before payment — no signal that the order can actually move fast once placed. Cab-hailing apps solved this exact trust gap by showing "cars nearby" before booking; same-day delivery checkout has no equivalent, despite carrying the same anxiety. Closing this gap should lift checkout completion rate and order volume, with no change to pricing, discounts, or acquisition spend — it's a conversion lever inside the funnel we already own.

2. For users. Customers placing a time-boxed order (e.g., "Today, 6:00–8:00 PM") have no way to judge, before they pay, whether that promise is realistic. That uncertainty peaks at the exact moment they're deciding whether to trust the app with their money — the "Confirm and pay" screen. Showing live partner availability answers "can this actually get to me on time" before checkout, not after the order is already placed and out of their control.

How do we measure success?

1. Associated OKR/Goal : Increase checkout-to-order conversion by 10% this quarter without degrading delivery experience.

2. Success Metrics

- **Primary:** Checkout completion rate — % of sessions that reach "Place order" from "Confirm and pay."
- **Secondary:** Time-to-complete-checkout — does the added signal speed up the decision, or introduce friction/hesitation (e.g., users staring at a low count)?

3. Guardrail / Do-not-disturb metrics

Delivery complaint rate within 24 hours of the promised slot. This is the check against gaming the primary metric — it would be trivial to inflate checkout completion by always showing a healthy partner count (cached, rounded up, or shown regardless of real supply). If the number shown stops reflecting real assignment speed, complaints will rise even as completion looks like a win. The test only ships if completion improves with no negative movement on this guardrail.

Who are the users:

1. Persona Time-pressed urban shoppers ordering small-basket, same-day delivery (e.g., a ₹500–₹1,000 order in HSR Layout, Bengaluru) into a fixed delivery window. Price-insensitive relative to the value of the item, but time- and trust-sensitive about whether the delivery promise will actually hold.

2. Problems we are solving [Needs, Problems, Jobs to be done]

- **Need:** Reassurance that the order will arrive within the promised window, before committing payment.
- **Problem:** Checkout today is a leap of faith — the customer has zero visibility into whether the order can realistically move fast once placed. The slot ("Today, 6:00–8:00 PM") is a promise with no evidence behind it.

- **Job to be done:** “Help me decide, in the next few seconds, whether I can trust this delivery promise enough to pay right now.”

3. How do we know these problems exist (research)

Cab-hailing apps (Uber, Ola) built the “X cars nearby” pattern specifically to solve this same trust gap at the moment of booking — a well-established, user-tested pattern we’re borrowing, not inventing.

Solution:

1. Brief of the solution

Add a live “delivery partners active near you” module to the checkout (“Confirm and pay”) screen, sourced from the same real-time partner-availability data our dispatch system already uses internally. The module is shown only when supply is above a minimum threshold — it never displays a low or discouraging number, it simply doesn’t render.

2. Other alternatives considered, with prioritization

Alternative	Why not chosen (for MVP)
Show an estimated delivery-time countdown instead of partner count	Higher risk — an ETA is a stronger, more specific promise. Breaking it damages trust more directly than an imprecise count would.
Show partner count earlier, on the product/cart page	Weaker link to the actual moment of hesitation (payment commitment); testing it at checkout isolates the effect where it matters most.
Do nothing — rely on existing slot copy alone	This is effectively the “Before” / control variant already being tested against.

Prioritized using **RICE framework**. Checkout placement scored highest given it reuses existing dispatch data (no new back-end build) and sits directly at the drop-off point we’re trying to fix.

Product flow (Details of the feature/product):

1. Customer Journey

Cart (“1 item · ₹649”) → Checkout / “Confirm and pay” (partner-availability module shown here) → Place order → Order-placed / tracking screen (“We will notify you once a partner is assigned”).

2. Wireframes and Flow diagrams See clickable Before/After prototype:

<https://www.howdysid.com/projects/partner-availability-test-BlitzCaseStudy/index.html>

3. User Stories

- As a customer, I want to see how many delivery partners are active near me at checkout, so I can trust my order will move quickly once placed.
- As a customer, when supply is low, I want the app to say nothing rather than show a discouraging number, so I’m not scared off right before paying.
- As the business, I want this tested safely, so a guardrail metric is tracked alongside the primary metric before we roll out to 100%.

4. User acceptance criteria

- Given supply is above the minimum threshold, when the customer reaches "Confirm and pay," then the live partner count renders before the "Place order" button.
- Given supply is below the minimum threshold, when the customer reaches checkout, then the module does not render at all (no zero-state, no low-number state).
- Given the customer is in the test's control (Before) group, when they reach checkout, then no partner-availability module is shown, and the screen is otherwise identical to the treatment group.

5. Edge cases

- Zero or near-zero partners near the delivery address.
- Partner count changes between page load and "Place order" click.
- Live-count API fails to respond (timeout/error) — module should fail silently, not block checkout.
- Customer's delivery address is outside serviceable/mapped supply zones.
- User re-enters checkout after backgrounding the app — is the count refreshed or stale?

6. Event tracking sheet (starter structure)

Event name	Trigger	Key properties	Screen
checkout_viewed	Checkout screen loads	variant (before/after), user_id, cart_value	Checkout
partner_widget_shown	Module renders	partner_count, zone_id	Checkout
partner_widget_hidden_low_supply	Count below threshold	zone_id, actual_count	Checkout
place_order_clicked	"Place order" tapped	variant, partner_count_shown (if any)	Checkout
order_placed	Order successfully created	order_id, variant, slot_promised	Order confirmation
delivery_complaint_filed	Complaint logged within 24h of slot	order_id, complaint_type, variant	Support/CRM

Tentative Timelines:

Task	Date

Dependencies:

1. Open Questions
2. Infrastructure requirements Real-time (or near-real-time) read access to the dispatch system's live partner-availability data, exposed to the checkout client with acceptable latency.
3. Budget approvals None expected for MVP — reuses existing dispatch data; no new data source to license or build.
4. Partner Support (APIs, Partnerships) Dispatch/Ops team API for live partner-count-by-zone.
5. Internal Dependencies (Tech, Design, Other Teams)

- **Data/Analytics:** event tracking implementation, guardrail dashboard.
- **Design:** widget states (shown / hidden), loading and error states.
- **Engineering:** checkout integration, experiment framework for the 50/50 split.

Related documents:

1. Tech planning document — If any
2. Design planning document — If any
3. Go-to-market planning — N/A for this experiment; internal conversion test, no external launch