

PRD: Project Infinity

Predictive Inventory &
Auto-Replenishment

blinkit



Project Name: Project "**Infinity**" Predictive Inventory & Auto-Replenishment

Feature Name: The Infinite Fridge

Version: 1.0

Status: Draft for Review

Author: Aishwarya– MBA Class of 2027

Date: January 15, 2026

1. Executive Summary & Strategic Alignment

Context: Quick Commerce (Q-Commerce) currently operates on a purely "Reactive" model—we wait for the user to realize they need milk, then we rush to deliver it in 10 minutes. This creates high demand volatility and forces us to maintain expensive "Safety Stock" in Dark Stores to prevent stockouts.

The Opportunity: By shifting from **Reactive** to **Predictive**, we can anticipate demand before it happens. "The Infinite Fridge" uses historical consumption data to predict when a user is running low on staples (Milk, Bread, Eggs) and prompts a "Just-in-Time" restock.

Objective: To launch a predictive cart engine that converts high-frequency staples into "Auto-Suggested Orders," thereby increasing **Customer Lifetime Value (LTV)** and smoothing out demand spikes for better logistics planning.

2. Problem Statement

- **The User Friction:** "I forgot to order milk last night, and now I have no coffee." This "mental load" causes users to switch to competitors or local Kirana stores.
 - **The Operational Inefficiency:** Ad-hoc orders for single items (e.g., just one packet of milk) have terrible Unit Economics. The delivery cost often exceeds the margin on the item.
 - **Inventory Waste:** Without demand visibility, Dark Stores overstock perishables, leading to high "Shrinkage" (wastage).
-

3. Target Audience & Persona

1. **The "Busy Parent" (Primary):** Orders milk/bread every 2 days. Values reliability over speed. Hates running out of breakfast items.
 2. **The "Bachelor" (Secondary):** Erratic ordering patterns. Needs help "adulting" (reminders to buy basics).
 3. **The Dark Store Manager (Internal):** Needs better demand forecasting to optimize shelf space and rider allocation.
-

4. User Scenarios (The "Prediction Loops")

Scenario A: The "Nudge" (Active Confirmation)

- **Data:** User buys 500g Bread every 4 days. It has been 4 days since the last order.
- **Trigger:** Push notification at 6:00 PM: *"Looks like you're out of bread. Added a fresh loaf to your cart. Order now for tomorrow morning?"*
- **Action:** User clicks "Yes." Order is placed.

Scenario B: The "Smart Slot" (Operational Optimization)

- **Logic:** Since this isn't an "emergency" (predicted in advance), the system defaults this order to a **"Green Slot"** (delivery within a 60-minute window, not 10 minutes).
- **Benefit:** This allows the Logistics Engine to **batch** this order with other nearby urgency orders, reducing Cost Per Delivery (CPD).

Scenario C: The "Subscription-Lite"

- **Logic:** For highly predictable items (e.g., 1L Toned Milk Daily), the user enables "Auto-Checkout." The app automatically orders it daily at 7 AM unless paused.

5. Functional Requirements

5.1. The Prediction Engine (Backend - Python/ML)

The core asset is the **Consumption Velocity Algorithm**.

- **Input:** Order History, SKU Shelf Life (e.g., Milk = 2 days, Atta = 30 days), Family Size (inferred from avg. basket size).
- **Logic:** Calculate **Days_To_Zero** for every SKU.
 - Days to zero = Quantity Purchased / Avg Daily Consumption Rate
- **Trigger:** When **Days_To_Zero < 1**, trigger the "Restock Nudge."

5.2. User Experience (App Frontend)

- **"The Infinite Fridge" Dashboard:** A new section in the app showing "Pantry Levels" (Visual UI showing a milk bottle at 10% capacity).
- **One-Tap Reorder:** Notifications must allow checkout without opening the full app.
- **Snooze:** "I still have some left." (Feeds back into the algorithm to correct the consumption rate).

5.3. Logistics Integration (Rider App)

- **De-prioritization Tag:** Orders generated via "Infinite Fridge" are tagged `PRIORITY_LOW`.
 - **Batching Logic:** These orders sit in the "Dispatch Queue" for up to 20 minutes, waiting to be batched with a high-priority "10-minute delivery" going to the same sector.
-

6. Non-Functional Requirements

- **Accuracy:** False Positives (predicting empty when user has stock) must be $< 10\%$ to avoid annoying users.
 - **Data Privacy:** Explicit consent required to analyze consumption patterns.
 - **Inventory Sync:** Dark Store inventory must be reserved *before* the notification is sent to ensure the item doesn't go Out-of-Stock (OOS) by the time the user clicks.
-

7. Metrics & Success Indicators (KPIs)

To measure the success of Project Infinity, we track:

Metric Category	Metric Name	Target
North Star	Stockout Prevention %	Reduce user "zero-stock" days by 40%
Financial	Average Order Value (AOV)	Increase by 15% (upsell with staples)
Operational	Cost Per Delivery (CPD)	Reduce by 8% (via "Green Slot" batching)
Retention	D30 Retention	Increase from 22% to 28%
Tech	Prediction Accuracy	> 85% acceptance rate of suggestions

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

Phase 1: The "Breakfast Club" (Beta)

- **SKUs:** Only high-velocity breakfast items (Milk, Bread, Eggs, Butter).
- **Users:** Power users (>10 orders/month) in high-density zones (e.g., Indiranagar, Bangalore).

Phase 2: "Pantry Loader"

- **SKUs:** Expand to "medium velocity" items (Atta, Rice, Detergent).
- **Incentive:** "Use Smart-Cart and get ₹10 off (Green Discount)."

Phase 3: Automated Replenishment

- Full rollout with "Auto-Debit" wallet integration for seamless daily milk delivery.

9. Risks & Mitigation

Risk	Impact	Mitigation Strategy
The "Vacation" Problem	High	User goes on holiday, milk keeps arriving/piling up. Solution: "Vacation Mode" toggle + Geolocation check (if user is 500km away, pause auto-orders).
Prediction Failure	Medium	Algorithm suggests buying Atta when user just bought it from a store offline. Solution: "I bought this elsewhere" button to reset the counter.
Dark Store Overload	Low	Too many "Green Slot" orders clogging the dispatch area. Solution: Cap the number of predictive orders per hour per store.

Conclusion

Project Infinity transforms Blinkit from a **Utility (Delivery App)** into a **Service (Home Manager)**. By shifting demand from "Stochastic" (Random) to "Deterministic" (Predicted), we unlock massive efficiencies in the supply chain while solving a genuine human problem—forgetfulness.

Immediate Next Step: Initiate A/B test on "Push Notification Copy" to determine the optimal time of day for "Milk Reminders" (Morning vs. Evening).