

Assignment 1 - Phase 3: Features, Prioritization + PRD

Step 1: Brainstorm Solutions (Diverge)

P0 Problem:

"Bigbasket users are struggling with real-time item unavailability at the final checkout stage because the app's inventory data does not sync instantly with the local warehouse stock. This leads to last-minute cart abandonment and forcing users to switch to competitors to complete their shopping."

Brainstorming for Bigbasket's Problem

Solution	Short Description	Key Assumption	Category
Inventory Reservation	Just like booking a movie ticket or an airline seat, once a user adds a "Low Stock" item to their cart, keep it reserved for 15-20 minutes	It gives user the certainty that if it's in their cart, it is certainly available to buy for the stipulated time	Medium Confidence
Real-Time Stock Availability Tags	Display stock availability tag "Only 5 left"	This manages expectations early so the user is aware and need to make the purchase fast and is not shocked at checkout if the item goes "out of stock"	High Confidence
One-Tap Substitutes	If an item goes out of stock at the last second, use AI to suggest the best 1-tap substitutes (e.g., a different brand of Atta)	It prevents the user frustration. It keeps the user on the app instead of switching to a competitor.	Low Confidence
Smart Alerts	The app triggers automated push notifications as stock levels drop at 5 units, then again at 3 If a user has an item in their cart and the stock level hits 5 units (and again at 3), The app triggers automated push notifications as stock levels drop at 5 units, then again at 3	This moves the user from "browsing" to "buying" by creating a legitimate, data-driven sense of urgency.	High Confidence

Step 2: Prioritize Smartly (From Possibility to Focus)

Top Ideas to Proceed With: **Real-Time Stock Availability Tags**, **Smart Alerts**

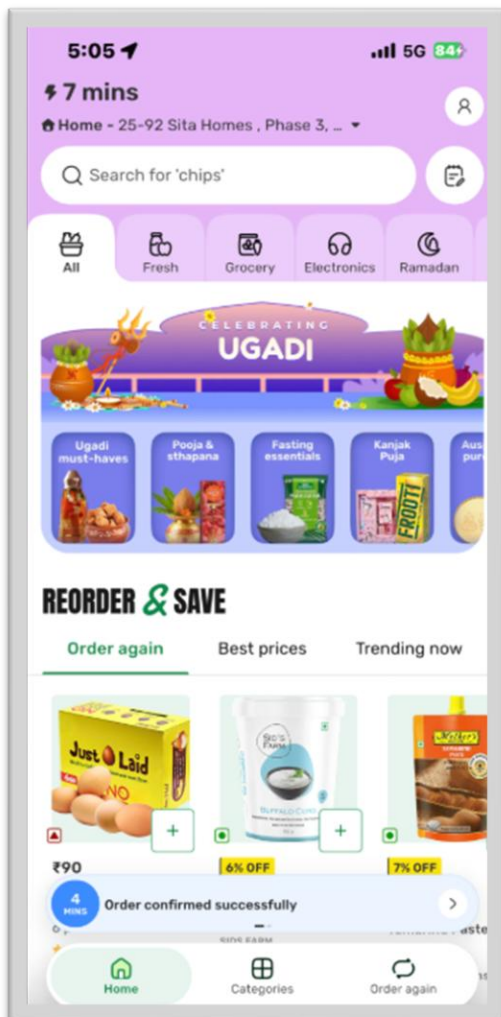
P0 solution:

Implement **Real-Time Stock Availability Tags** for transparency and **Smart Alerts** for proactive urgency to eliminate instances of last-minute stock-outs and reduce cart abandonment

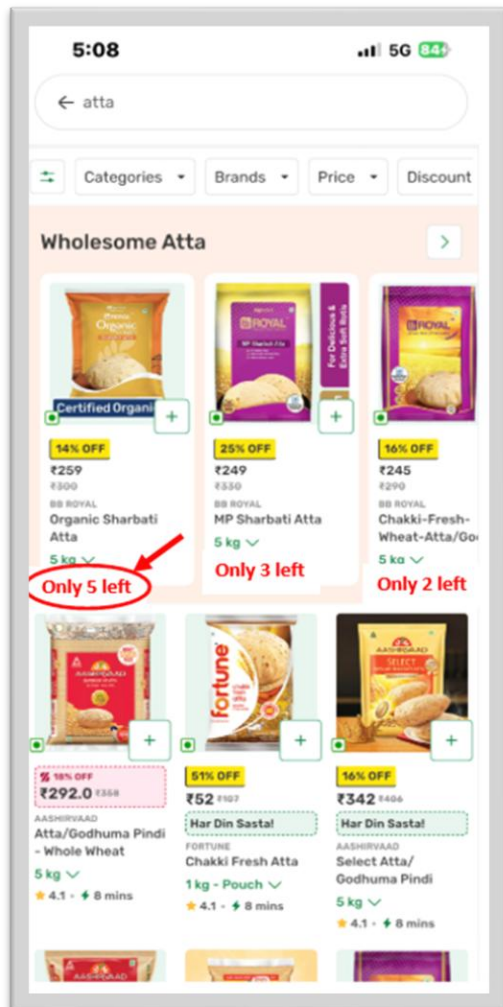
Step 3: Wireframe Your Top Ideas

1) Wireframe Bigbasket Search screen with “Real-Time Stock Availability Tag”

Screen 1: Bigbasket Home Screen



Screen 2: Bigbasket Search screen with “Real-Time Stock Availability Tag”



2) Wireframe for “Smart Alert” push notification

Alert sent to user when stock level hits 5 units

Dear Teja,

Only 5 units left of 'BB Royal Sharbati Atta' in your cart. Check out now to secure it

<https://bigbasket.com/cart/checkout>

Regards,
Team bigbasket

Alert sent to user when stock level hits 3 units

Hurry Teja!

Only 3 units left of 'BB Royal Sharbati Atta' in your cart. Check out now to secure it

<https://bigbasket.com/cart/checkout>

Regards,
Team bigbasket

STEP 4: Define Success Metrics

Success Metrics for “Real-Time Stock Availability Tag” & “Smart Alerts”

Metric Type	Metric Name	Definition	Why it matters
Key Metric	Checkout Completion Rate (Low-Stock Items)	% of users who successfully purchase an item after viewing a "Low Stock" Availability Tag or receiving a Smart Stock Alert	It directly measures if real-time data reduces cart abandonment.
Secondary Metric	Stock Alert Click-Through Rate (CTR)	% of users who click on a "Low Stock" push notification and return to their cart.	Indicates if the alert actually attracts the user’s immediate interest and brings them back into the buying journey before the item sells out.
Guardrail Metric	Notification Opt-out / Uninstall Rate	% of users who disable notifications or uninstall the app within 24 hours of receiving a Smart Stock Alert.	Ensures the scarcity signals don't feel like "spam" or manipulative marketing, which would hurt the long-term user experience and trust.

STEP 5: Identify Pitfalls & Mitigations

Assumption at risk	Failure mode	Impact	Likelihood	Mitigation	Detection Signal	Containment / Rollback
The inventory count displayed on the app/website perfectly matches stock in the warehouse	A user receives a "3 left" alert, pays for it, but the item delivered is damaged or missing	High	High	Program the system to trigger "Out of Stock" when physical units hit 2. Only send alerts when stock is between 5 and 2	Rate of warehouse-initiated cancellations for "Low Stock" flagged orders.	Automatically disable alerts for specific warehouses showing >5% mismatch.
Users will perceive "Only 5 left" as a helpful data point rather than a fake marketing tactic.	If every item shows "Low Stock," users stop trusting the signal, leading to "Alert Fatigue" and platform distrust.	Medium	High	Only trigger the tag if the item has a high velocity (selling fast) and is actually below 10% of the daily average stock.	Decrease in Click-Through Rate (CTR) on tags over time or increase in "Spam" reports.	Cap the number of "Low Stock" tags a single user can see in one session (e.g., max 3 per page).
The server can handle the massive spike in checkout attempts triggered by a simultaneous push	Sending a "3 left" alert to 50 users at once causes a sudden spike that crashes the	Low	Medium	Send notifications in micro-batches (e.g., to the 5 users who have had the item in their cart the longest first).	Spike in Server Errors or Latency on the Checkout page immediately	Immediately disable automated push notifications while maintaining static UI tags.

notification to many users.	checkout API or causes payment timeouts.				following a push alert	
-----------------------------	--	--	--	--	------------------------	--

Product Requirements Document

Title/Initiative	Real-Time Inventory Transparency & Smart Alerts
Date & Version	16/03/26 Version 1
Product - Point of Contact (POC)	Sridevi Nakka
Design POC	Umesh T
Tech POC	Tejasri K
Marketing POC	Subhash K

Why? (Objective)

To eliminate the "Inventory Betrayal" at checkout by syncing real-time warehouse data with the user interface through visual signals and proactive notifications.

For Business:

- **Increase Retention:** Reduce platform churn by 15% by preventing high-intent users from switching to competitors (Blinkit/Zepto) due to unfulfilled carts.
- **Optimize Operations:** Reduce warehouse-initiated cancellations and customer support tickets related to "ghost stock."

For User:

- **Certainty & Trust:** Provide shoppers with the confidence that their "Master Order" will be fulfilled exactly as planned.
- **Efficiency:** Save time for the user by managing stock expectations early in the journey, avoiding last-minute disappointments.

How do we measure success?

1. Associated OKR/Goal:

The primary objective is to increase **User Retention** by ensuring a reliable "One-and-Done" shopping experience.

2. Success Metrics & Guardrail Metrics for “Real-Time Stock Availability Tag” & “Smart Alerts”

Metric Type	Metric Name	Definition	Why it matters
Key Metric	Checkout Completion Rate (Low-Stock Items)	% of users who successfully purchase an item after viewing a "Low Stock" Availability Tag or receiving a Smart Stock Alert	It directly measures if real-time data reduces cart abandonment.
Secondary Metric	Stock Alert Click-Through Rate (CTR)	% of users who click on a "Low Stock" push notification and return to their cart.	Indicates if the alert actually attracts the user's immediate interest and brings them back into the buying journey before the item sells out.
Guardrail Metric	Notification Opt-out / Uninstall Rate	% of users who disable notifications or uninstall the app within 24 hours of receiving a Smart Stock Alert.	Ensures the scarcity signals don't feel like "spam" or manipulative marketing, which would hurt the long-term user experience and trust.

Who are the users:

1. User Persona:



Name	Stockpiler Teja
Tagline	"The Super Saver Family Manager"
Age/Role	38, Senior Accountant & Father of two
Location	Hyderabad

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

- **The Need:** Reliability and transparency in the grocery fulfilment process.
- **The Problem:** Users spend 15–20 minutes building a cart only to be told at the final payment screen that essential items are unavailable.
- **Jobs to be Done (JTBD):**

"When I am planning my weekly household restock, I want to know exactly what is available in real-time, so I don't waste time on a cart that cannot be fulfilled."

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

- **User Interviews:** 2/3 power shoppers identified "checkout stock-outs" as their primary frustration.
- **Survey Evidence:** Users explicitly marked "Item Unavailability" as the top reason for abandoning Bigbasket for competitors.
- **Competitor Benchmarking:** Rivals like Zepto and Blinkit have higher real-time sync accuracy, raising the market standard for user expectations

Solution

Brief of the Solution

We are implementing a dual-layer communication system:

1. **Transparency:** Real-time **Availability Tags** (e.g., "Only 5 left") on search and product pages to manage expectations during browsing.
2. **Urgency: Smart Alerts** that trigger push notifications when an item already in a user's cart hits critical low-stock thresholds (5 and 3 units), prompting them to complete the purchase immediately.

Product flow (Details of the feature/product):

Customer Journey

To show the impact of the P0 solution, we need to map the **Customer Journey**. This journey highlights the "Before" (The Pain) and the "After" (The Solution) for the primary persona, **Stockpiler Teja**.

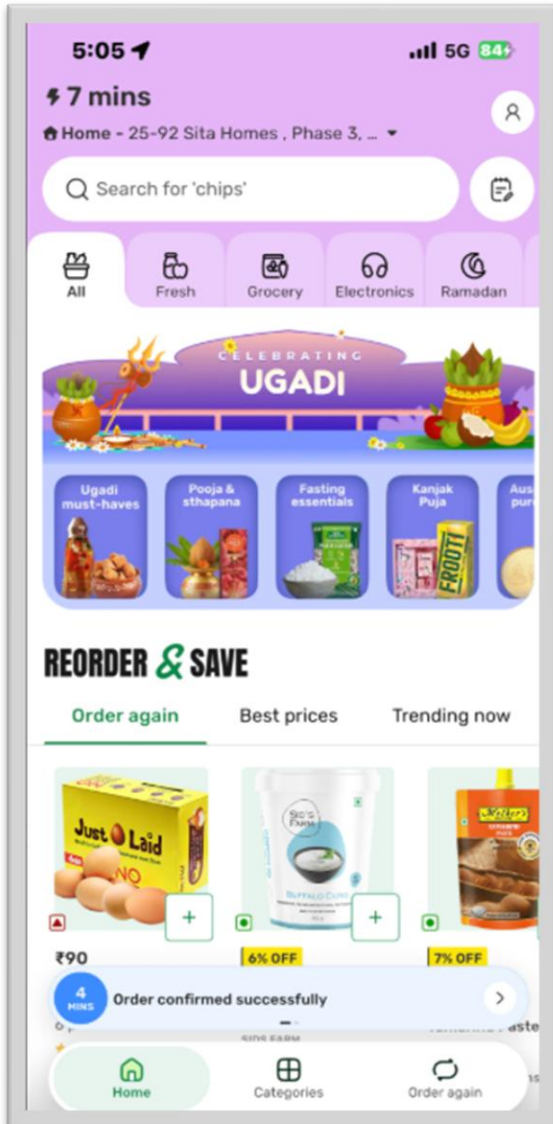
This journey focuses on the "Inventory Betrayal" at the moment of checkout and how real-time tags and alerts fix it.

Stage	1. Discovery & Browsing	2. Cart Building	3. Critical Trigger	4. Checkout & Payment	5. Post-Purchase
User Action	Searches for weekly staples (Rice, Atta, Oil).	Adds items to cart; looks for "Value Packs."	Continues shopping for secondary items.	Proceeds to final payment screen.	Receives order confirmation.
The "Before" (The Pain)	No indication of stock levels.	Adds 10kg Atta assuming it's available.	The Blind Spot: Item sells out in the warehouse while he is still browsing.	The Betrayal: Red error message: "Item Out of Stock." He has to delete and find a new brand.	Frustration: Abandons cart; opens Blinkit.
The "After" (The Solution)	Real-Time Tags: Sees "Only 5 left" on his favorite Atta brand.	Prioritizes the "Low Stock" items immediately.	Smart Alert: Receives push: "Only 3 left of Atta in your cart! Checkout now."	Certainty: Hits 'Pay' with confidence. No last-minute errors.	Success: Trust in Bigbasket restored.
Emotional State	😊 Neutral / Planning	😊 Productive	👁️ Alerted / Focused	😊 Relieved / Satisfied	👍 Loyal / Delighted

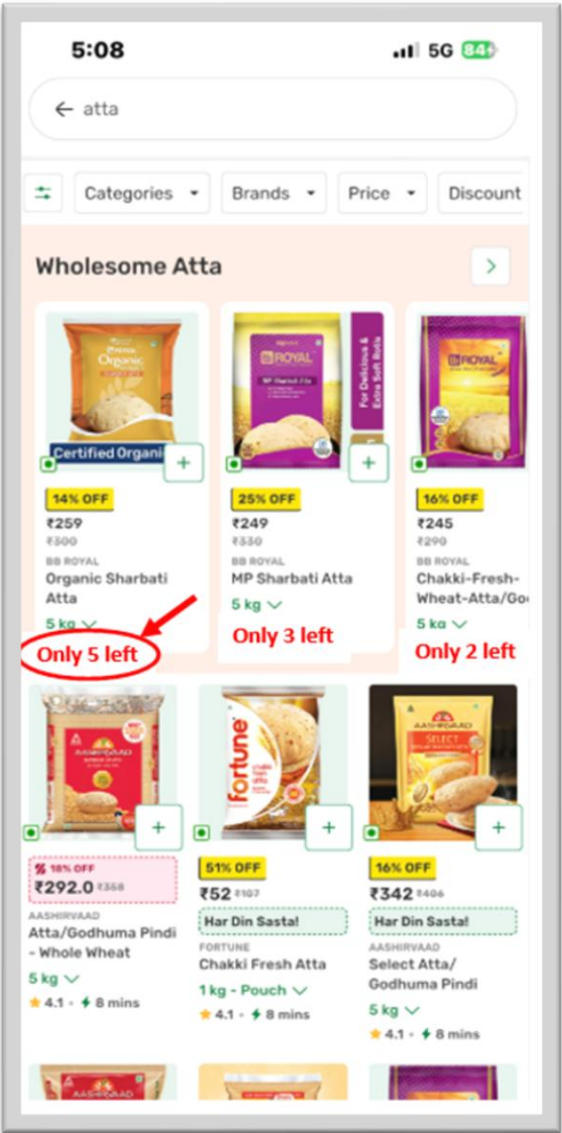
2. Wireframes

1) Wireframe Bigbasket Search screen with “Real-Time Stock Availability Tag”

Screen 1: Bigbasket Home Screen



Screen 2: Bigbasket Search screen with “Real-Time Stock Availability Tag”



2) Wireframe for “Smart Alert” push notification

Alert sent to user when stock level hits 5 units

Dear Teja,
Only 5 units left of ‘BB Royal Sharbati Atta’ in your cart. Check out now to secure it
<https://bigbasket.com/cart/checkout>

Regards,
Team bigbasket

Alert sent to user when stock level hits 3 units

Hurry Teja!
Only 3 units left of ‘BB Royal Sharbati Atta’ in your cart. Check out now to secure it
<https://bigbasket.com/cart/checkout>

Regards,
Team bigbasket

3. User Stories

User Stories: Real-Time Inventory & Smart Alerts

User Role	User Story	Value / "So That"	Priority
Stockpiler Teja	As a monthly shopper, I want to see a "Low Stock" tag on my staples while browsing.	I can prioritize adding them to my cart before they sell out, ensuring my "Master Order" is complete.	P0
Quick-Fix Kavya	As a recipe-based shopper, I want to receive a push notification if an item in my cart is about to sell out.	I can checkout immediately and not lose the critical ingredients needed for my evening meal.	P0

Business Analyst	As a system manager, I want to set a 2-unit safety buffer for physical stock.	I don't accidentally promise a "last unit" that might be damaged or misplaced in the warehouse.	P1
All Users	As a shopper, I want to be prevented from adding more units than are physically available.	I don't face a quantity-mismatch error at the final payment step.	P1

4. User Acceptance Criteria:

Feature 1: Real-Time Stock Availability Tags (FR-01)

- The "Low Stock" tag must trigger automatically when a specific SKU's warehouse stock drops to **10 units or fewer**.
- The UI must display the exact count remaining (e.g., "Only 4 left!") in a high-contrast color (Amber for 6-10 units, Red for <5 units).
- The tag must be visible on the **Search Results Page (SRP)**, **Category Pages**, and the **Product Description Page (PDP)**.
- If the stock reaches the "Safety Buffer" (2 units), the tag must change to "Out of Stock" and the "Add" button must be disabled.

Feature 2: Smart Stock Alerts / Push Notifications (FR-02)

- A push notification must be sent only to users who have the "Low Stock" SKU in their **active cart**.
- The first alert must trigger at **5 units** and a final "Urgency" alert at **3 units**.
- The notification must include the specific product name and be personalized (e.g., "*Hurry Teja! Only 3 left...*").
- **Deep-linking:** Tapping the notification must bypass the homepage and land the user directly on the **Checkout/Payment** screen.

Feature 3: Dynamic Quantity Capping (FR-03)

- If a user attempts to use the "+" button to add more units than are available in the local warehouse, the button must become inactive.
- A "Toast" message must appear stating: "*Maximum available quantity reached for this item.*"
- If stock drops *while* the user is on the cart page, the quantity must auto-adjust to the maximum available, with a clear UI notification to the user.

5. Edge Cases

- **The Simultaneous Checkout:** If the last item is bought by another user while a customer is on the payment page, the app must provide an immediate "Item just sold out" popup with a 1-tap alternative suggestion.
- **Stale Data:** If a user opens the app after a long period of inactivity, the cart must perform a "Silent Refresh" to update stock tags before the user hits checkout.
- **Notification Overload:** Users will not receive more than 2 Smart Alerts within a 2-hour window to prevent "Alert Fatigue."

6. Event tracking sheet

Event Name	Trigger (When is it fired?)	Properties / Metadata to Capture	Goal / Purpose
scarcity_tag_viewed	When a "Low Stock" tag (e.g., Only 5 left) enters the user's viewport on the Search or Product page.	product_id, category, current_stock_count, user_segment	Measures Visibility: Are users actually seeing the transparency signals?
scarcity_tag_clicked	When a user clicks on a product card that is displaying a "Low Stock" tag.	product_id, stock_at_click, position_in_list	Measures Intent: Does the tag increase the click-through rate to the product page?
low_stock_add_to_cart	When a user adds an item to the cart that is currently flagged with a scarcity tag.	product_id, quantity, stock_level	Measures Urgency: Does the tag drive immediate "Add to Cart" behavior?
smart_alert_sent	When the system triggers a push notification (at 5 or 3 units).	product_id, threshold_level (5 or 3), notification_id	Baseline: Tracks how many nudges are being sent per day.

smart_alert_opened	When a user taps the push notification.	product_id, time_to_open, deep_link_success (True/False)	Measures Effectiveness: Is the notification compelling and the deep-link working?
checkout_started_from_alert	When a user lands on the Checkout page specifically via the smart alert deep-link.	product_id, cart_value, is_power_shopper	Measures Conversion Path: Does the alert successfully bypass friction?
inventory_mismatch_error	When a user hits 'Pay' but the item is actually OOS (post-safety buffer failure).	product_id, warehouse_id, error_type	Failure Signal: Measures the "False Hope" rate and system accuracy.
notification_disabled	When a user goes to settings and opts out of "Inventory Alerts."	user_id, last_alert_sent_timestamp	Guardrail: Measures if our alerts are causing negative user sentiment (Spam).

Tentative Timelines

Task	Date
Leadership approval	28/03/2026
Design ready	10/04/2026
Prototype testing	24/04/2026
Development starts	01/05/2026
Beta Launch	07/06/2026

