Assignment-1 Phase-3

Phase 3: Features, Prioritization + PRD

Step 1: Brainstorm Solutions (Diverge)

P0 Problem : Urban working professionals placing high-intent, urgent orders struggle with late discovery of out-of-stock items because real-time inventory visibility is inaccurate, leading to order cancellations, loss of trust, and reduced repeat purchases.

Solutions:

1. Real-time out of stock Notification and Auto Cart Sync

- **Description:** Even if a user adds items to cart and returns later (e.g., after 1 hour), the system continuously validates cart items in the background; if any item goes out of stock in that duration, the user is notified instantly (or on app open) and the item is auto-removed or marked unavailable before checkout.
- **Assumption:** Users prefer being proactively informed about stock changes during their cart lifecycle rather than discovering it at order or delivery time.
- **Category:** High Confidence

2. Checkout-time Inventory Lock

- **Description:** Lock inventory (selected items)for 2–3 minutes once user enters checkout
- **Assumption:** Temporary reservation reduces failed orders until complete buyout.
- **Category:** High Confidence

3. Smart Substitution Engine

- **Description:** Suggest closest alternatives (brand/price/quantity match) when OOS
- **Assumption:** Users accept substitutes in urgent scenarios
- **Category:** High Confidence

4. **Dynamic ETA + Availability Tradeoff Toggle**

- **Description: “**Get in 10 min (limited items)” vs “Get in 20 min (full availability)”
- **Assumption:** Users are willing to trade speed for certainty
- **Category:** High Confidence

5. Post-order OOS Recovery Flow

- **Description:** If item goes OOS after order then instant options like:
  - Replace
  - Remove
  - Cancel item with refund
- **Assumption:** Control reduces frustration
- **Category:** High Confidence

6. Cross-store Fulfillment

- **Description:** Fetch item from nearby store if unavailable locally

- **Assumption:** Logistics complexity won't heavily impact delivery time though minor impact could be there.
- **Category:** Low Confidence

7. Predictive Inventory (AI-based)
   - **Description:** Predict fast-moving SKUs and pre-stock nearby stores
   - **Assumption:** Demand patterns are predictable enough.
   - **Category:** Low Confidence

8. Restock ETA Display
   - **Description:** we can show like "Back in 15 mins / 1 hour" for OOS items
   - **Assumption:** Users are willing to wait if informed
   - **Category:** Low Confidence

9. Cart Expiry Timer for High-demand Items
   - **Description:** Show timer: "Item reserved for 2 mins"
   - **Assumption:** Urgency reduces cart abandonment + stock conflicts
   - **Category:** Low Confidence

10. Cross Inventory pooling Network
    - **Description:** All nearby dark stores act as one unified inventory system
    - **Assumption:** Real-time syncing across stores is feasible at scale
    - **Category:** Moonshot
11. AI "OOS Prevention Engine"
    - **Description:** System predicts stockouts + auto-triggers:
      1. Rebalancing inventory
      2. Dynamic pricing (reduce demand)
    - **Assumption:** AI can influence both supply and demand in real-time
    - **Category:** Moonshot
12. "Never OOS" Membership (Premium Guarantee)
    - **Description:** Paid users get priority inventory allocation
    - **Assumption:** Users will pay for reliability in urgent use cases
    - **Category:** Moonshot

Step 2: Prioritize Smartly (From Possibility to Focus)

Reach = No of users facing issue(rough)/ No of total users (rough)

Let's assume total users= 100000 for (solution -1)

No of users add to cart = 50000

out of 50k assume 10% face OOS issues

50k of 10%= 5000

here Reach = 5000 users/day

total users = 100000

% Reach = 5000/100000= 5% (very few users but impactful)

RICE MODEL

Solution	Reach	Impact	Confidence	Effort	RICE Score	Notes
Real-time out of stock Notification and Auto Cart Sync	1	3	3	2	4.5	Strong UX fix; eliminates surprise, low effort, high trust gain.
Checkout-time Inventory Lock	1	2	3	2	3	Reduces last-minute failures needs careful timeout and fairness handling.
Dynamic ETA + Availability Tradeoff Toggle	2	2	2	2	4	Gives user control, improves transparency but adds decision complexity.
Post-order OOS Recovery Flow	1	3	2	2	3	Saves bad experience post-failure, reactive, not preventive.
Cross-store Fulfillment	1	3	1	3	1	Improves availability via logistics, risk of delivery delay and cost.
Predictive Inventory (AI-based)	2	3	1	3	2	Reduces OOS frequency, depends heavily on data quality.
Restock ETA Display	2	2	1	1	4	Sets expectations, accuracy challenge can break trust.
Cross Inventory pooling Network	3	3	1	3	3	Maximizes availability across stores, high system complexity.

AI “OOS Prevention Engine”	3	3	1	3	3	End-to-end optimization (supply and demand), ambitious, long-term bet.
Never OOS” Membership (Premium Guarantee)	2	3	1	2	3	Monetizes reliability, risk of unfair allocation perception.

highest rice score we are getting for the first entry in the model that is 4.5

Step 3: Wireframe Your Top Ideas

Screen-1

https://www.figma.com/design/xC4pAeRaGt2sJb6RkLwn4F/Untitled?node-id=1-3&p=f&t=exP1K8Vsx6G2jnT6-0

Screen-2

https://www.figma.com/design/xC4pAeRaGt2sJb6RkLwn4F/Untitled?node-id=5-47&p=f&t=exP1K8Vsx6G2jnT6-0

Screen-3

https://www.figma.com/design/xC4pAeRaGt2sJb6RkLwn4F/Untitled?node-id=8-2&p=f&t=exP1K8Vsx6G2jnT6-0

Screen-4

https://www.figma.com/design/xC4pAeRaGt2sJb6RkLwn4F/Untitled?node-id=8-21&p=f&t=w0mkGXkD4EW8pgQA-0

STEP 4: Define Success Metrics

Metric Type	Metric Name	Definition	Why it matters
Key Metric	Order Completion Rate (without OOS issues)	10% of users who successfully place and receive orders without	Directly measures if late OOS discovery is reduced and user

		any item going OOS post-checkout	experience is improved
Secondary Metric	Cart Abandonment Rate (due to OOS)	10% of users who drop off after items go out of stock in cart	Shows whether real-time updates are reducing frustration or causing drop-offs
Secondary Metric	OOS Detection Rate (Pre-checkout)	10% of OOS events identified before checkout vs after order	Measures effectiveness of real-time inventory sync
Secondary Metric	Alternative Item Conversion Rate	10% of users who select suggested alternatives after an item goes OOS	Indicates recovery success and retained conversions
Guardrail Metric	Cart Abandonment Rate (Overall)	Total 20% of users abandoning cart	Ensures solution doesn't increase overall drop-offs
Guardrail Metric	Average Order Value (AOV)	Average cart value per order	Ensures revenue is not negatively impacted

STEP 5: Identify Pitfalls & Mitigations

Assumption at risk	Failure mode	Impact	**Likelihood**	**Mitigation**	**Detection Signal**	**Containment / Rollback**
Users will trust real-time inventory updates	Users see incorrect OOS updates will lose trust and stop relying on app	High	Medium	Add validation layer (double-check before marking OOS); show "high demand" instead of hard removal in uncertain cases	OOS Detection Accuracy	Disable auto-removal, switch to "mark unavailable" instead
Users will continue checkout instead of abandoning cart	Users get frustrated due to frequent OOS updates results in abandon entire cart	High	High	send Batch notifications, limit frequency, show alternatives instantly	Cart Abandonment Rate (post-OOS)	Reduce notification frequency,temporarily remove auto-removal feature
System can accurately detect OOS in real-time	Delay or mismatch in inventory sync → users still face OOS at delivery	High	Medium	Add sync checkpoints (cart view, checkout) fallback to final validation before order placement.	% OOS detected post-checkout	-

Step 6: PRD Creation

here is the link to my personalized PRD

https://www.notion.so/Product-Requirements-Document-PRD-Template-fd6e62b86671837eb36581d5bf6910f9?source=copy_link