# Inventory Maintainer SAAS – Phase 1

We’re building a SaaS application to help omni-channel retailers manage their inventory efficiently across a network of stores and warehouses.

**1. Control Tower (Landing Page & Dashboard)**

**Goal:** Provide a high-level snapshot of inventory health and make the Control Tower a gateway to deeper functionalities.

**Key Elements to Explore and Prototype:**

* **Network View**  
  Visual layout showing all nodes (warehouses, DCs, stores), with key metrics or alerts overlaid.
* **KPI Cards**  
  Display summarized, real-time indicators:
  + Current Demand
  + Inventory Position
  + Weeks of Supply
* **Alerts System**  
  Trigger and display issues like stockouts, overstock, or demand-supply mismatches.
  + When there are multiple alerts can we group/tag them based on similarity or urgency?
* **Quick Access to Key Functions**  
  Links or modules that lead into deeper capabilities:
  + Rebalancer
  + Forecast
  + Settings

**Think about:** UI/UX, responsive layout, how to modularize the dashboard for scale and future integrations.

### ****2. Data Ingestion Module****

**Goal:** Design the system to ingest, validate, and integrate various data feeds from customers.

**Key Topics to Explore:**

* **Types of Incoming Data**
  + Master
    - Catalogue
    - Location
  + Snapshots
    - Inventory
  + Temporal
    - Sales
    - Transfers
    - Inbound
  + Computed
    - Forecast
    - Assortment
* **Frequency and Triggers**
  + What should be ingested daily, weekly, or monthly?
  + What happens if a file is missed?
* **Customer Data Mapping**
  + Design mapping logic between customer-provided fields (e.g., SKU\_ID, Store\_Code) and system internal IDs.
  + Should customers provide config upfront or will system infer it?
* **Ingestion Methods**
  + *File Upload (CSV/Excel) – Focus on this for now*
  + *API integrations – Focus on this for now*
  + GCP/AWS buckets
* What checks has to be done on the data?
  + How to detect missing data/data type errors upfront?
  + This should pop us as alerts in the control tower page.

**Think about:** How to make the ingestion process fault-tolerant, track errors, and enable easy onboarding of new customers.

### ****3. Systems Configuration****

**Goal:** Define configurable parameters that impact how the system behaves and makes decisions.

**Key Settings to Design and Document:**

* Network Nodes.
  + Warehouses and Stores
* Products
  + Product Hierarchies.
* **Inventory Planning Parameters**
  + Safety Stock (by SKU/store or globally defined)
  + Reorder Level
  + Lead Time assumptions
* **Supply Chain Paths**
  + Transfer paths (e.g., Warehouse A → Store B)
  + Prioritization logic (e.g., based on capacity, proximity, etc.)
* **Operational Constraints**
  + Node/route capacities (daily limits)
  + Minimum Order Quantities
* **User & Role Management**
  + Who can view/edit what?
  + Roles for planners, regional managers, admins
* **Strategy Modes**
  + Push vs Pull inventory logic
  + Static vs dynamic safety stock calculation

**Think about:** Creating intuitive UI forms or APIs for users to manage these settings, and how default settings might apply in absence of customer inputs.