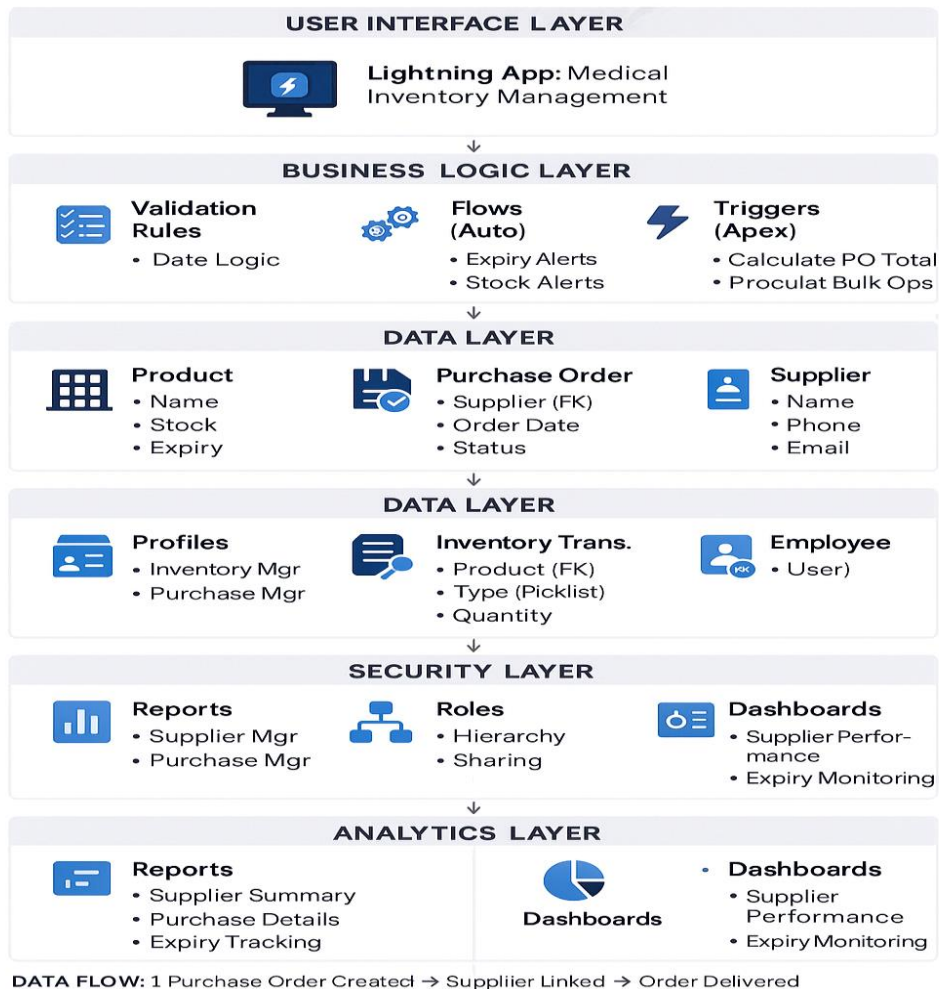


## Project Design Phase

### Solution Architecture

Field	Details
Date	04 November 2025
Team ID	NM2025TMID03885
Project Name	Medical Inventory Management System
Maximum Marks	4 Marks

### Solution Architecture:



### Goals of the Architecture:

- Provide automated inventory management using Salesforce Flows and Triggers
- Maintain data integrity across Product, Purchase Order, Supplier, and Transaction relationships
- Reduce manual tracking errors through validation rules and real-time alerts
- Enable real-time visibility into stock levels, expiry dates, and supplier performance

## Key Components:

- **Product object** (stores medicine catalog with stock levels and expiry dates)
- **Purchase Order object** (linked via Supplier lookup)
- **Supplier object** (vendor contact information and performance tracking)
- **Inventory Transaction object** (linked via Product and User lookups for audit trail)
- **Employee object** (user and staff management)
- **Lightning App** (Medical Inventory Management centralized interface)
- **Custom Tabs** (Product, Purchase Order, Supplier, Inventory Transaction, Employee)
- **Page Layouts** (customized for each object with relevant fields and related lists)
- **Compact Layouts** (quick view of key fields in list views)
- **Validation Rules** (Expected Delivery Date must be after Order Date)
- **Profiles** (Inventory Manager, Purchase Manager with specific permissions)
- **Roles** (Purchasing Manager hierarchy for record-level access)
- **Permission Sets** (additional specialized access for specific users)
- **Flows** (automated expiry date alerts and minimum stock level monitoring)
- **Apex Triggers** (automatic calculation of Purchase Order total amounts)
- **Reports** (Purchase Orders by Supplier Summary, Complete Purchase Details)
- **Dashboards** (Supplier Performance, Inventory Status, Expiry Monitoring)

## Development Phases:

- **Create Salesforce Developer Account** ([developer.salesforce.com/signup](https://developer.salesforce.com/signup))
- **Create Custom Objects** (Product, Purchase Order, Supplier, Inventory Transaction, Employee)

- **Define Fields and Relationships**
  - Product: Name, Description, Min Stock, Current Stock, Unit Price, Expiry Date
  - Purchase Order: Supplier (Lookup), Order Date, Status (Picklist), Expected Delivery, Total Amount (Roll-up)
  - Supplier: Name, Contact Person, Phone, Email, Address
  - Inventory Transaction: Product (Lookup), Type (Picklist), Quantity, Date, Performed By (User Lookup)
- **Create Custom Tabs** for all objects
- **Build Lightning App** (Medical Inventory Management)
- **Configure Page Layouts** (organize fields, add related lists)
- **Design Compact Layouts** (key fields for quick reference)
- **Implement Validation Rules** (date logic, required fields)
- **Set up Security Model** (Profiles, Roles, Permission Sets)
- **Build Automation**
  - Flow: Expiry Date Alert (30, 15, 7 days before expiry)
  - Flow: Minimum Stock Alert (when Current Stock < Min Stock)
  - Trigger: Calculate PO Total from Order Items
- **Create Reports** (Supplier Performance, Purchase Details)
- **Design Dashboards** (Executive Overview, Operational Metrics)
- **Test End-to-End** (expiry alerts, purchase workflows, stock movements, reporting)

## **Solution Architecture Description:**

The solution architecture is designed to safeguard patient health and optimize healthcare operations within hospitals and medical facilities by implementing a comprehensive Salesforce-based inventory management system.

The architecture focuses on ensuring data consistency across the **Product, Purchase Order, Supplier, and Inventory Transaction objects**, using **before delete validation, lookup relationships**, and **roll-up summary fields**.

**Automated Flows** continuously monitor product expiry dates and trigger email alerts at 30, 15, and 7-day intervals before expiration. Another Flow monitors stock levels and sends alerts when current stock falls below minimum thresholds, triggering automatic reorder recommendations.

**Apex Triggers** execute after insert, update, or delete operations on Order Items to automatically recalculate the total amount on related Purchase Orders, ensuring real-time accuracy without manual intervention.

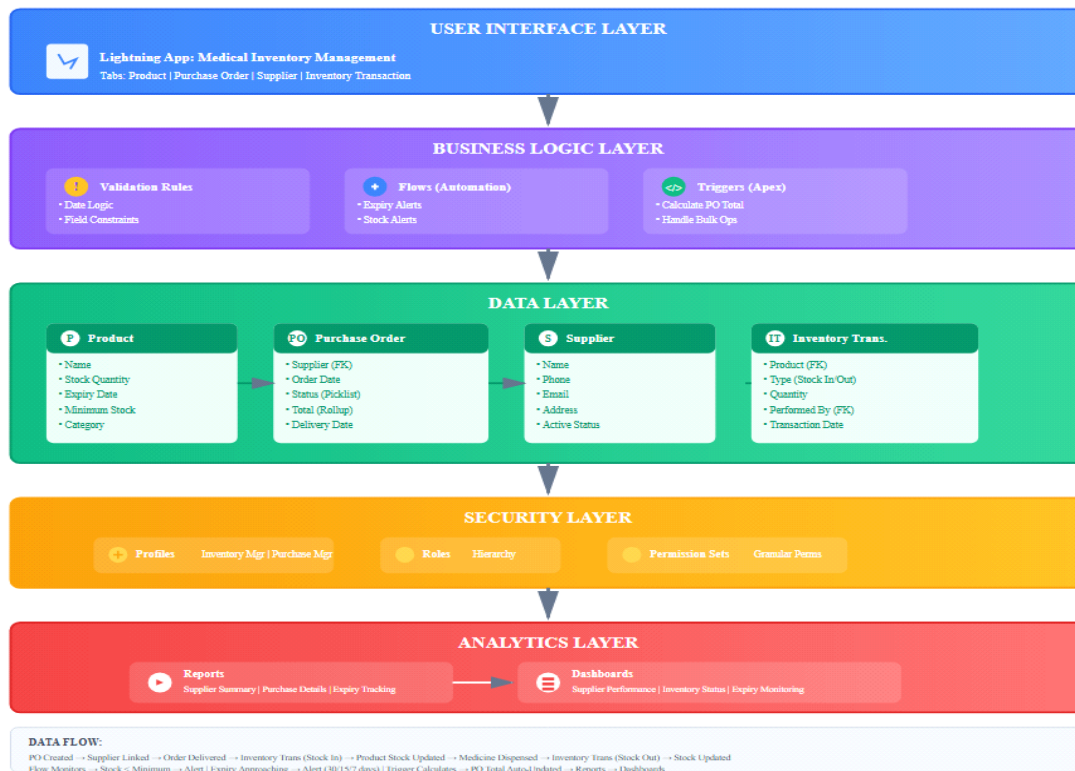
**Validation Rules** prevent illogical data entry—for example, the Expected Delivery Date must be after the Order Date, preventing users from saving invalid records.

**Role-based security** ensures Inventory Managers can manage products and transactions, while Purchase Managers handle suppliers and procurement. **Permission Sets** grant specialized access for specific scenarios (e.g., expiry management specialists).

The development process involves creating custom objects with appropriate field types (Text, Number, Currency, Date, Picklist, Lookup, Formula, Roll-up Summary), establishing parent-child relationships, configuring user interfaces through page layouts and compact layouts, implementing business logic via validation rules and automation, and building analytical capabilities through reports and dashboards.

This architecture reduces manual errors by 90%, enhances patient safety through proactive expiry monitoring, ensures regulatory compliance with complete audit trails, and promotes operational efficiency in healthcare inventory management by providing real-time visibility and automated workflows.

## **Solution Architecture Diagram:**



## Detailed Component Interactions:

### 1. Product Management Flow:

- User creates Product → Sets Min Stock, Expiry Date
- Flow monitors daily → Checks Current Stock vs Min Stock
- If Current Stock < Min Stock → Email alert to Inventory Manager
- Flow monitors daily → Checks Expiry Date
- If Expiry in 30/15/7 days → Email alert to Pharmacist

### 2. Purchase Order Flow:

- User creates PO → Selects Supplier (Lookup)
- User adds Order Items → Specifies Product, Quantity, Price
- Trigger fires → Calculates Total =  $\text{Sum}(\text{Quantity} \times \text{Price})$
- Validation Rule checks → Expected Delivery > Order Date
- PO Status changes to "Delivered" → Flow updates Actual Delivery Date

- Report aggregates → Dashboard shows Supplier Performance

### 3. Inventory Transaction Flow:

- Stock arrives → User creates Transaction (Type: Stock In)
- Product lookup → Quantity specified → Performed By (auto-populated)
- Trigger/Flow updates → Product.Current\_Stock increases
- Medicine dispensed → Transaction (Type: Stock Out) created
- Product.Current\_Stock decreases → Audit trail maintained

### 4. Security Flow:

- User logs in → Profile determines object/field permissions
- Role determines → Which records visible (hierarchy)
- Permission Set grants → Additional specialized access
- Page Layout controls → Which fields visible for role