

# **MEDICAL INVENTORY MANAGEMENT**

**College name:** Srinivasa Ramanujan Institute of Technology

**TEAM ID :** LTVIP2025TMID30494

## **TEAM MEMBERS**

**1. Team Leader:** Balaji G

**Mail-id:** 224G1A0510@srit.ac.in

**2. Team member:** Y Bhavitha

**Mail-id:** 224G1A0511@srit.ac.in

**3. Team member:** J Sree Abhilash

**Mail-id:** 224G1A05A5@srit.ac.in

**4. Team member:** Juturu Lakshmanji

**Mail-id:** 224G1A0539@srit.ac.in

# FINAL DOCUMENTATION

Date	28 June 2025
Team ID	LTVIP2025TMID30494
Project Name	Medical Inventory Management

## 1. INTRODUCTION

### 1.1 Project Overview

The **Medical Inventory Management** system is a Salesforce-based application designed to simplify inventory operations in healthcare. It streamlines medicine tracking, supplier coordination, and purchase order management with automation features such as Flows, Scheduled Jobs, and Apex triggers.

#### Key Modules:

- Stock Monitoring
- Purchase Order Lifecycle
- Automation
- Supplier Performance Management
- Real-time Reporting & Dashboards

### 1.2 Purpose

The main objective is to eliminate manual inventory tracking errors and enable timely procurement, expiry management, and process transparency—ensuring safe and efficient hospital operations.

## 2. IDEATION PHASE

### 2.1 Problem Statement

Manual inventory tracking leads to expired stock use, supply delays, and limited visibility into stock levels—posing risks to patient safety and operational efficiency.

### 2.2 Empathy Map

- **Think & Feel:** “What if the stock expires before use?”
- **Hear:** “Supplies are delayed again.”
- **See:** Irregular delivery timelines, outdated stock entries
- **Say & Do:** Uses Excel sheets, manually checks inventory

- **Pain:** Unreliable alerts, high error risk
- **Gain:** Automated stock alerts, supplier history tracking

## 2.3 Brainstorming

Ideas included:

- PO auto-calculation using Apex
- Inventory dashboards
- Custom supplier scorecards

## 3. REQUIREMENT ANALYSIS

### 3.1 Customer Journey Map

From stock shortage detection to PO approval and product delivery, Inventory and Purchase Managers interact with dashboards, flows, and data-driven insights throughout the process.

### 3.2 Solution Requirements

- **Salesforce Platform**
  - **Custom Objects:** Product, Supplier, Purchase Order, Order Item, Inventory Transaction
  - **Automation:** Flows, Apex, Validation Rules
  - **Reports:** Supplier Summary, Expiry Forecast, PO History
  - **Access Control:** Profiles, Permission Sets

### 3.3 Data Flow Diagram

- **Inventory Manager:** Updates stock, monitors expiry
- **Purchase Manager:** Generates PO → Flow triggers approval → Order delivered → Inventory updated

### 3.4 Technology Stack

- **Platform:** Salesforce
- **Logic Layer:** Apex, SOQL, Flows
- **Data Layer:** Custom Objects
- **Visualization:** Reports & Dashboards

## **4. PROJECT DESIGN**

### **4.1 Problem–Solution Fit**

Automating the inventory and procurement cycle addresses expiry risks, approval delays, and inconsistent data tracking.

### **4.2 Proposed Solution**

A low-code Salesforce application using Apex, Flows, and dashboards to manage inventory lifecycle, with alerts, approval logic, and real-time tracking.

### **4.3 Solution Architecture**

- **Frontend:** Lightning Experience UI
- **Backend:** Custom Objects & Triggers
- **Automation:** Scheduled Flows, Approval Processes
- **Security:** Role-based access
- **Reporting:** Dashboards + Custom Report Types

## **5. PROJECT PLANNING & SCHEDULING**

- **Week 1:** Object & Field Setup
- **Week 2:** Flow & Validation Rule Implementation
- **Week 3:** Apex Trigger Creation & Testing
- **Week 4:** Report Building & Dashboard Design
- **Week 5:** Final Testing & Documentation

### **Objects:**

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', 'Object Manager', and a search bar. The main area displays the 'Product' object details. On the left, a sidebar lists various object settings: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, Scoping Rules, Object Access, and Triggers. The main content area shows the 'Details' tab for the Product object. It includes fields for Description, API Name (Product\_\_c), Singular Label (Product), Plural Label (Products), and several checkboxes for Enable Reports, Track Activities, and Track Field History. Deployment status is listed as Deployed. A link to Standard salesforce.com Help Window is also present. At the bottom right are 'Edit' and 'Delete' buttons.

SETUP > OBJECT MANAGER  
**Purchase Order**

**Details**

Fields & Relationships  
Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets  
Object Limits  
Record Types  
Related Lookup Filters  
Search Layouts  
List View Button Layout  
Restriction Rules  
Scoping Rules  
Object Access  
Triggers

**Details**

Description  
API Name: Purchase\_Order\_c  
Custom: ✓  
Singular Label: Purchase Order  
Plural Label: Purchase Orders

Enable Reports: ✓  
Track Activities  
Track Field History  
Deployment Status: Deployed  
Help Settings  
Standard salesforce.com Help Window

Edit Delete

SETUP > OBJECT MANAGER  
**Order Item**

**Details**

Fields & Relationships  
Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets  
Object Limits  
Record Types  
Related Lookup Filters  
Search Layouts  
List View Button Layout  
Restriction Rules  
Scoping Rules  
Object Access  
Triggers

**Details**

Description  
API Name: Order\_Item\_c  
Custom: ✓  
Singular Label: Order Item  
Plural Label: Order Items

Enable Reports: ✓  
Track Activities  
Track Field History  
Deployment Status: Deployed  
Help Settings  
Standard salesforce.com Help Window

Edit Delete

SETUP > OBJECT MANAGER  
**Inventory Transaction**

**Details**

Fields & Relationships  
Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets  
Object Limits  
Record Types  
Related Lookup Filters  
Search Layouts  
List View Button Layout  
Restriction Rules  
Scoping Rules  
Object Access  
Triggers

**Details**

Description  
API Name: Inventory\_Transaction\_c  
Custom: ✓  
Singular Label: Inventory Transaction  
Plural Label: Inventory Transactions

Enable Reports: ✓  
Track Activities  
Track Field History  
Deployment Status: Deployed  
Help Settings  
Standard salesforce.com Help Window

Edit Delete

**SETUP > OBJECT MANAGER**  
**Supplier**

**Details**

Description

API Name  
Supplier\_\_c

Custom  
✓  
Singular Label  
Supplier  
Plural Label  
Suppliers

Enable Reports  
✓  
Track Activities

Track Field History

Deployment Status  
Deployed  
Help Settings  
Standard salesforce.com Help Window

**Fields & Relationships**

- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters
- Search Layouts
- List View Button Layout
- Restriction Rules
- Scoping Rules
- Object Access
- Triggers

## Fields:

**SETUP > OBJECT MANAGER**  
**Product**

**Fields & Relationships**  
10 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Current Stock Level	Current_Stock_Level__c	Number(18, 0)		
Expiry Date	Expiry_Date__c	Date		
Last Modified By	LastModifiedBy	Lookup(User)		
Minimum Stock Level	Minimum_Stock_Level__c	Number(18, 0)		
Owner	OwnerId	Lookup(User,Group)	✓	
Product Description	Product_Description__c	Text Area(255)		
Product Id	Name	Text(80)	✓	
Product Name	Product_Name__c	Text(25)		
Unit Price	Unit_Price__c	Currency(16, 2)		

**SETUP > OBJECT MANAGER**  
**Purchase Order**

**Details**

Description

API Name  
Purchase\_Order\_\_c

Custom  
✓  
Singular Label  
Purchase Order  
Plural Label  
Purchase Orders

Enable Reports  
✓  
Track Activities

Track Field History

Deployment Status  
Deployed  
Help Settings  
Standard salesforce.com Help Window

**Fields & Relationships**

- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters
- Search Layouts
- List View Button Layout
- Restriction Rules
- Scoping Rules
- Object Access
- Triggers

**Purchase Order**

Fields & Relationships				
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Actual Delivery Date	Actual_Delivery_Date__c	Date		
Created By	CreatedById	Lookup(User)		
Expected Delivery Date	Expected_Delivery_Date__c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
Order Count	Order_Count__c	Roll-Up Summary (COUNT Order Item)		
Order Date	Order_Date__c	Date		
Owner	OwnerId	Lookup(User/Group)	✓	
Purchase Order Id	Name	Text(80)	✓	
Supplier ID	Supplier_ID__c	Lookup(Supplier)	✓	
Total Order Cost	Total_Order_Cost__c	Currency(16, 2)		

**Order Item**

Fields & Relationships				
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount__c	Formula (Currency)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Order Item Id	Name	Text(80)	✓	
Product ID	Product_ID__c	Lookup(Product)	✓	
Purchase Order ID	Purchase_Order_ID__c	Master-Detail(Purchase Order)	✓	
Quantity Ordered	Quantity_Ordered__c	Number(18, 0)		
Quantity Received	Quantity_Received__c	Number(18, 0)		
Unit Price	Unit_Price__c	Formula (Currency)		

**Inventory Transaction**

Fields & Relationships				
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User/Group)	✓	
Purchase Order ID	Purchase_Order_ID__c	Lookup(Purchase Order)	✓	
Total Order Cost	Total_Order_Cost__c	Formula (Currency)		
Transaction Date	Transaction_Date__c	Date		
Transaction Id	Name	Text(80)	✓	
Transaction Type	Transaction_Type__c	Picklist		

## Validation and Flows :

**Purchase Order Validation Rule**

**Validation Rule Detail**

- Rule Name: Expected\_Delivery\_Date\_Validation
- Error Condition Formula: (Expected\_Delivery\_Date\_\_c - Order\_Date\_\_c) > 7
- Error Message: The Expected Delivery Date should not exceed 7 days
- Description: Expected Delivery Date Validation
- Created By: Balaji Gudur (6/23/2025, 3:29 AM)
- Modified By: Balaji Gudur (6/23/2025, 3:29 AM)

**Actual Delivery Date Updating**

**Flow Detail**

Flow Label: Actual Delivery Date Updating	Description: Actual Delivery Date Updating	Flow API Name: Actual_Delivery_Date_Updating
Environment: Default	Namespace Prefix:	Type: Auto-launched Flow
Active Version: 1	Activated/Deactivated by: Balaji Gudur (6/23/2025, 4:00 AM)	URL: /flow/Actual_Delivery_Date Updating
Trigger: Record--Run Before Save	Created By: Balaji Gudur (6/23/2025, 4:00 AM)	
Modified By: Balaji Gudur (6/23/2025, 4:00 AM)		

**Flow Versions**

Action	Flow Label	Version	Description	Built with	Created Date	Type	Status	Progress Status	Run in Mode	API Version for Running the Flow	Log Metrics to Data Cloud
Open   Run   Deactivate	Actual Delivery Date Updating	1	Actual Delivery Date Updating	Flow Builder	6/23/2025, 5:09 AM	Auto-launched Flow	Active	Activated	Default Mode	64.0	

## Apex Code :

```
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
CalculateTotalAmountTrigger.aprx CalculateTotalAmountHandler.apxc
Code Coverage: None ▾ API Version: 64 ▾ Go To
trigger CalculateTotalAmountTrigger on Order_Item__c (after insert, after update, after delete, after undelete) {
    CalculateTotalAmountHandler.calculateTotal(Trigger.new, Trigger.old, Trigger.isInsert, Trigger.isUpdate, Trigger.isDelete, Trigger.isUndelete);
}
```

File • Edit • Debug • Test • Workspace • Help • < >

CalculateTotalAmountTrigger.apxc CalculateTotalAmountHandler.apxc

Code Coverage: None • API Version: 64 Go To

```

1 public class CalculateTotalAmountHandler {
2     public static void calculateTotal(List<Order_Item__c> newItems, List<Order_Item__c> oldItems, Boolean isInsert, Boolean isUpdate, Boolean isDelete, Boolean isUndelete) {
3         Set<Id> parentIds = new Set<Id>();
4         if (isInsert || isUpdate || isUndelete) {
5             for (Order_Item__c ordItem : newItems) {
6                 parentIds.add(ordItem.Purchase_Order__Id__c);
7             }
8         }
9         if (isUpdate || isDelete) {
10            for (Order_Item__c ordItem : oldItems) {
11                parentIds.add(ordItem.Purchase_Order__Id__c);
12            }
13        }
14     Map<Id, Decimal> purchaseToUpdateMap = new Map<Id, Decimal>();
15     if (!parentIds.isEmpty()) {
16         List<AggregateResult> aggrList = [
17             SELECT Purchase_Order__Id__c, SUM(Amount__c) totalAmount
18             FROM Order_Item__c
19             WHERE Purchase_Order__Id__c IN :parentIds
20             GROUP BY Purchase_Order__Id__c
21         ];
22         for (AggregateResult aggr : aggrList) {
23             Id purchaseOrderId = (Id)aggr.get('Purchase_Order__Id__c');
24             Decimal totalAmount = (Decimal)aggr.get('totalAmount');
25             purchaseToUpdateMap.put(purchaseOrderId, totalAmount);
26         }
27     List<Purchase_Order__c> purchaseToUpdate = new List<Purchase_Order__c>();
28     for (Id purchaseOrderId : purchaseToUpdateMap.keySet()) {
29         Purchase_Order__c purchaseOrder = new Purchase_Order__c(Id = purchaseOrderId, Total_Order_Cost__c = purchaseToUpdateMap.get(purchaseOrderId));
30         purchaseToUpdate.add(purchaseOrder);
31     }
32     if (!purchaseToUpdate.isEmpty()) {
33         update purchaseToUpdate;
34     }
35   }
36 }
37 }
```

## Dashboards

Medical Inventory ... Products Purchase Orders Order Items Inventory Transactions Suppliers Reports Dashboards

Search... ★ + ? ⚡ 🔍

**Medical Inventory Dashboard**  
Medical Inventory DashBoard  
As of Jun 28, 2025, 10:41 PM Viewing as Balaji Gudur

Refresh Edit Subscribe

**Purchase Orders based on Suppliers**

Supplier ID	Sum of Total Order Cost
SUP001	\$168.8
SUP002	\$135
SUP003	\$100
SUP004	\$90
SUP005	\$40
SUP006	\$300

[View Report \(Purchase Orders based on Suppliers\)](#) As of Jun 28, 2025, 10:41 PM

**Complete Purchase Details Report**

Purchase Order	Sum of Amount
PO011	\$168.8
PO012	\$135
PO013	\$100
PO014	\$90
PO015	\$40
PO016	\$300

[View Report \(Complete Purchase Details Report\)](#) As of Jun 28, 2025, 10:41 PM

## Functional Testing

New Purchase Order

\* = Required Information

**Information**

* Purchase Order Id PO021	Actual Delivery Date 7/10/2025	
* Supplier ID SUP002	Total Order Cost \$100.00	
* Order Date 6/28/2025	Owner Balaji Gudur	
Expected Delivery Date 7/12/2025	<span style="background-color: red; color: white; padding: 5px;">∅ We hit a snag.</span> <b>Review the errors on this page.</b> <ul style="list-style-type: none"> <li>The Expected Delivery Date should not exceed 7 days</li> </ul>	

∅ Cancel Save & New Save

Flow Builder    Actual Delivery Date Updating - V2    Debug Run: Actual Delivery Date Updating 6/29/2025, 10:20 AM    Completed

Edit Flow Convert to Test Debug Again Save As New Version Save Activate

Record-Triggered Flow Start

```

graph TD
    Start((Record-Triggered Flow Start)) --> GetPurchaseRecord[Get Purchase Record]
    GetPurchaseRecord --> Assignment[Assignment]
    Assignment --> UpdateOrder[Updating Purchasing Order]
    UpdateOrder --> End((End))
  
```

Debug Details

Expand All Basic Debug Log

Search this list...

Get Records: GET PURCHASE RECORD

We couldn't retrieve any records.

Assignment: Assignment

2 variables were updated.

Update Records: Updating Purchasing Order

One or more Purchase\_Order\_c records are ready to be updated.

Rollback

How the Interview Finished

The flow interview ran for 1.07 seconds and finished on June 29, 2025 at 10:20 AM.

## **6. DEVELOPMENT HIGHLIGHTS**

- Built objects with fields: Product (Expiry, Quantity), PO (Expected Delivery, Total Amount), Supplier
- Triggers for PO total calculation
- Validation Rules for delivery constraints
- Record Pages and Tabs for each object
- Dashboards for Stock Overview and Supplier Analysis
- Deployed Permission Sets for role separation

## **7. TESTING**

- Functional testing for record saves, and trigger outputs
- Validation rule triggers (e.g., invalid dates)
- Dashboard visuals checked against data changes
- Flow testing for correct alerting and field auto-updates

## **8. RESULTS**

- Real-time visibility into stock health
- Automated expiry and delivery tracking
- Improved procurement planning and supplier accountability
- Reduced manual errors and data duplication

## **9. ADVANTAGES & DISADVANTAGES**

### **Advantages:**

- Real-time expiry and inventory alerts
- Automated PO lifecycle
- Enhanced staff coordination
- Scalable to multi-unit hospitals

### **Disadvantages:**

- Requires internet access
- Learning curve for non-technical users
- Setup time if expanded to include integrations

## **10. CONCLUSION**

The Medical Inventory Management app improves patient safety, operational reliability, and procurement accuracy by combining intuitive Salesforce design with healthcare-specific workflows.

## **11. FUTURE SCOPE**

The Medical Inventory Management system is built for scalability and forward compatibility. Future enhancements may include:

**1. Integration with Barcode Scanners**

Allow faster stock updates and receipt confirmations using barcode-enabled devices or mobile apps.

**2. AI-Driven Stock Forecasting**

Use predictive analytics to forecast medicine usage trends and automate reordering before shortages occur.

**3. Mobile App Access**

Provide doctors and staff with quick access to inventory data from mobile devices on hospital floors or remote sites.

**4. Multi-Branch Inventory Sync**

Support centralized or distributed tracking across multiple hospital departments, buildings, or locations.

**5. IoT Integration for Storage Monitoring**

Integrate with temperature and humidity sensors to track conditions for sensitive medicine stock.

**6. Digital Signature Workflow**

Approve Purchase Orders and contracts using digital signing tools (e.g. Adobe Sign, DocuSign).

**7. Integration with Government Procurement Portals**

Link your PO process with e-marketplaces for seamless vendor approvals and policy compliance.

**8. Advanced Role-Based Analytics**

Provide role-specific dashboards (e.g. pharmacist vs procurement head) to personalize insights and simplify actions.

### **Demo Video link :**

<https://drive.google.com/file/d/1wl9Q7XZ6qg6vgHKphBI-30DLMTqUTfAh/view?usp=sharing>

### **GITHUB REPO LINK :**

[https://github.com/balaji-510/Medical\\_Inventory\\_Management](https://github.com/balaji-510/Medical_Inventory_Management)