

HandsMen Threads — Project Documentation

Elevating the Art of Sophistication in Men's Fashion

Author: Arvin T. De Vera

Date: November 27, 2025

Environment: Salesforce Developer Org / Lightning Experience

Table of Contents

1. Use Case Scenario
2. Project Overview
3. Custom App & Architecture
4. Data Model and Key Fields
5. Creating Data Records & Validation
6. Business Automations (Flows)
7. Apex Development (Triggers & Batch)
8. Email Templates
9. Conclusion

Use Case

I. Description

Developing a premium fashion platform to streamline men's bespoke tailoring and enhance customer experience through personalized styling and seamless order management.

II. User Scenario

- a. HandsMen Threads, a dynamic organization in the fashion industry, is embarking on a Salesforce project designed to revolutionize their data management and enhance customer relations. The project involves building a robust data model tailored to store all pertinent business data, ensuring a seamless flow of information across the organization.

A key aspect of this project is the maintenance of data integrity directly from the user interface (UI). This feature will safeguard the accuracy and consistency of the data, which is crucial for informed decision-making and reliable business operations. The project will integrate several new processes into the business workflow to improve customer service and operational efficiency:

- i. **Automated Order Confirmations:** Post-order confirmation, customers will receive an email update, fostering engagement and strengthening customer relations.
- ii. **Dynamic Loyalty Program:** Customer loyalty statuses will be updated based on purchase history, enabling personalized rewards and promoting repeat business.
- iii. **Proactive Stock Alerts:** When stock levels drop below five units, automatic emails will notify the warehouse team, ensuring timely restocking and preventing stockouts.
- iv. **Scheduled Bulk Order Updates:** Daily midnight, the system will process bulk orders, updating financial records and adjusting inventory, ensuring accurate stock levels for daily operations.

Project Overview

HandsMen Threads is a **custom Salesforce CRM solution** designed to elevate bespoke men's tailoring operations.

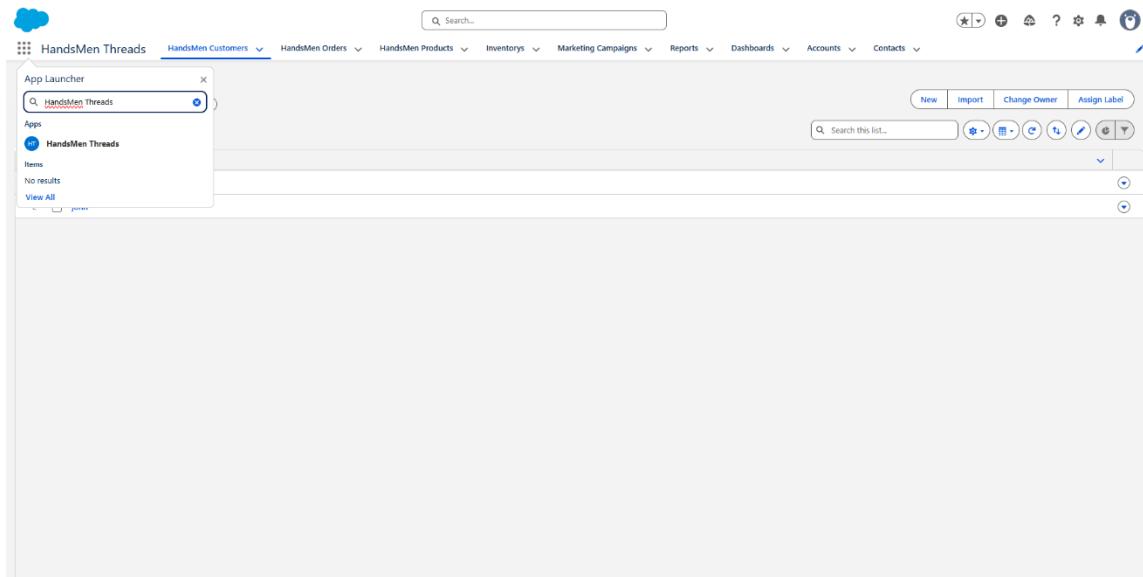
This platform centralizes customer management, loyalty programs, orders, product tracking, inventory deduction, and automated communication workflows.

Key Goals

- Manage customers and track loyalty tiers (Gold, Silver, Bronze).
- Maintain accurate inventory with automated stock deduction.
- Automate communication via Flow email alerts.
- Implement complex business logic using Apex triggers and Batch Apex.
- Enhance overall tailoring workflow from inquiry to delivery.

Tools Used

- Custom Objects & Tabs
- Lightning App
- Validation Rules
- Record-Triggered Flows & Scheduled Flows
- Apex Triggers & Batch Apex
- Reports & Dashboards



Custom App & Architecture

Name: HandsMen Threads Lightning App

Serves as the operational hub for:

- Sales
- Inventory team
- Admins
- Service team

Custom Objects Implemented

Object	Description
HandsMen Customers	Stores customer details, Gmail-validated email, loyalty status & purchase history
HandsMen Orders	Stores orders, quantities, statuses, total calculation
HandsMen Products	Product catalog with Price & SKU
Inventory	Tracks stock quantity per product / warehouse
Marketing Campaigns	Tracks promotional campaigns & engagement

Object Relationships

- Customer → Orders
- Orders → Product
- Product → Inventory
- Orders → Inventory (via Apex logic)

The screenshot shows the HandsMen Threads Lightning App interface. At the top, there's a navigation bar with links for HandsMen Customers, HandsMen Orders, HandsMen Products, Inventory, Marketing Campaigns, Reports, Dashboards, Accounts, and Contacts. Below the navigation is a search bar and a toolbar with various icons. The main area displays a customer record for "John Dela Cruz". The record includes fields for Name, Email (jdelacruz@gmail.com), Phone (09123456789), Loyalty Status (Silver), First Name (John), Last Name (Dela Cruz), Full Name (John Dela Cruz), Total Purchases (700), and Created By (Arvin De Vera). The record was last modified by Arvin De Vera on 11/26/2025, 7:21 PM.

HandsMen Threads

HandsMen Customers HandsMen Orders HandsMen Products Inventories Marketing Campaigns Reports Dashboards Accounts Contacts

HandsMen Order O-0003

New Contact Edit New Opportunity

Related Details

Owner Avin De Vera

HandsMen OrderNumber O-0003

HandsMen Product Shorts

HandsMen Customer John De Vera

Status Confirmed

Quantity 600

Total Amount 3,000

Customer Email deveras50@gmail.com

Created By Avin De Vera - 11/26/2025, 7:27 PM

Last Modified By Avin De Vera - 11/26/2025, 7:27 PM

HandsMen Threads

HandsMen Customers HandsMen Orders HandsMen Products Inventories Marketing Campaigns Reports Dashboards Accounts Contacts

HandsMen Product Shorts

New Contact Edit New Opportunity

Related Details

Owner Avin De Vera

HandsMen Product Name Shorts

SKU 0002

Price \$5

Stock Quantity 2,000

Created By Avin De Vera - 11/26/2025, 7:23 PM

Last Modified By Avin De Vera - 11/26/2025, 7:23 PM

HandsMen Threads

HandsMen Customers HandsMen Orders HandsMen Products Inventories Marketing Campaigns Reports Dashboards Accounts Contacts

Inventory I -0002

New Contact Edit New Opportunity

Related Details

Inventory Number I -0002

HandsMen Product Shorts

Stock Quantity 1,500

Stock Status Available

Warehouse HandsMen Shorts Warehouse

Created By Avin De Vera - 11/26/2025, 7:24 PM

Last Modified By Avin De Vera - 11/26/2025, 7:24 PM

Data Model and Key Fields

HandsMen Customer

- Full Name
- Email (Validation: must be Gmail)
- Phone
- Loyalty_Status__c (*None, Gold, Silver, Bronze*)
- Total_Purchases__c

HandsMen Product

- SKU__c
- Name
- Price__c
- Stock_Quantity__c

Inventory

- HandsMen_Product__c (*Master-Detail*)
- Stock_Quantity__c
- Stock_Status__c
- Warehouse__c

HandsMen Order

- Name (Order Number)
- HandsMen_Customer__c
- HandsMen_Product__c
- Customer_Email__c
- Quantity__c
- Status__c (*Pending, Confirmed, Rejection*)
- Total_Amount__c (*Apex-calculated*)
- Order_Date__c

Creating Data Records & Validation

I. Customer Creation with Validation

- a. A validation rule ensures only Gmail accounts are used.

i. Scenario:

1. **Entered Email:** johndc@asdf.com → Error: "Please fill Correct Gmail"
2. **Corrected to:** johndc@gmail.com → Saved successfully.

ii. Sample Data:

1. **Customer:** John Dela Cruz
2. **Loyalty Status:** Silver

The screenshot shows the 'Validation Rule Edit' page for the 'HandsMen Customer' object. The 'Rule Name' is set to 'Email'. The 'Error Condition Formula' field contains the formula 'NOT CONTAINS(Email__c, "@gmail.com")'. The 'Error Message' field contains the message 'Please fill Correct Gmail'. The 'Save' button is visible at the bottom right.

The screenshot shows the 'New HandsMen Customer' creation page. The 'HandsMen Customer Name' field is populated with 'John Dela Cruz'. The 'Email' field is populated with 'johndc@asdf.com'. A validation error message box is displayed, stating 'We hit a snag.' with the sub-instruction 'Review the errors on this page.' and a bullet point '• Please fill Correct Gmail'. The 'Save' button is visible at the bottom right.

HandsMen Customer Validation Rule and Sample data to be tested for validation

Steps: App Launcher → HandsMen Threads → New Handsman Customer → fill fields → Save Demo note: Enter invalid email "johndc@xyz.com" → validation error "Please fill correct Gmail" → change to "johndc@gmail.com" → Save success.

HandsMen Threads

HandsMen Customers

HandsMen Orders

HandsMen Products

Inventory

Marketing Campaigns

Reports

Dashboards

Accounts

Contacts

Search...

Owner: Arvin De Vera

HandsMen Customer Name: John Dela Cruz

Email: johndc@gmail.com

Phone: 09123456789

Loyalty Status: Silver

FirstName: John

LastName: Dela Cruz

FullName: John Dela Cruz

Total Purchases: 700

Created By: Arvin De Vera, 11/26/2025, 7:21 PM

Last Modified By: Arvin De Vera, 11/26/2025, 7:21 PM

New Contact | Edit | New Opportunity

II. Product Creation

- a. **HandsMen Product Name:** Shorts
- b. **SKU:** 0002
- c. **Price:** \$5
- d. **Stock Quantity:** 2,000

HandsMen Threads

HandsMen Customers

HandsMen Orders

HandsMen Products

Inventory

Marketing Campaigns

Reports

Dashboards

Accounts

Contacts

Search...

Owner: Arvin De Vera

HandsMen Product Name: Shorts

SKU: 0002

Price: \$5

Stock Quantity: 2,000

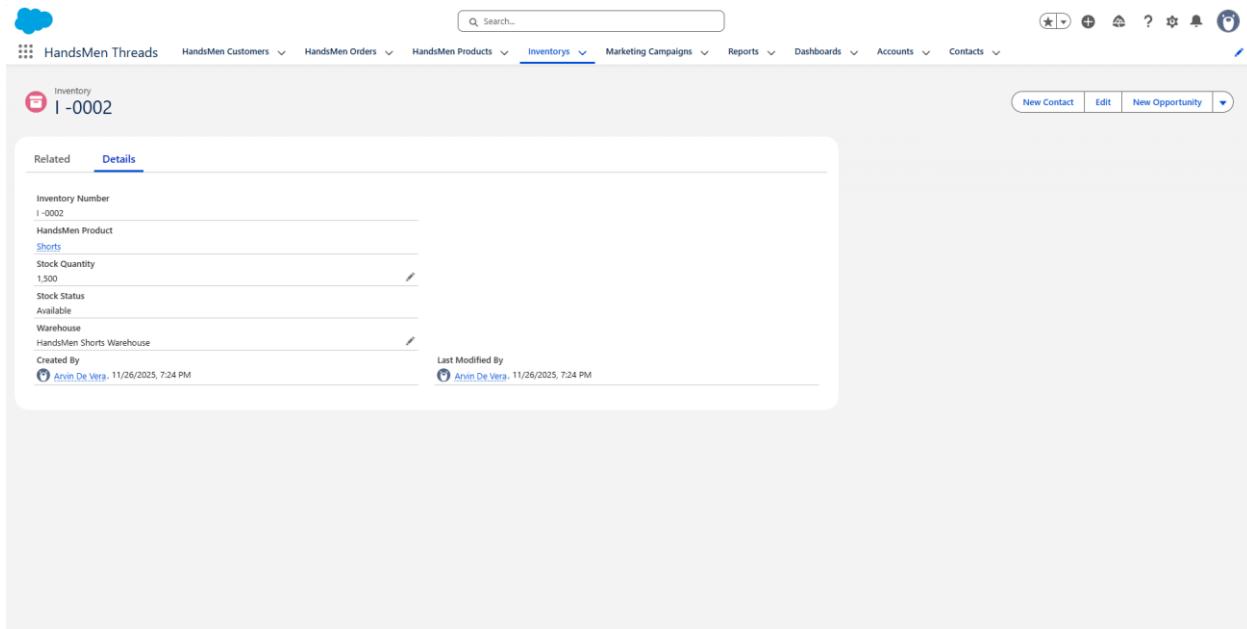
Created By: Arvin De Vera, 11/26/2025, 7:23 PM

Last Modified By: Arvin De Vera, 11/26/2025, 7:23 PM

New Contact | Edit | New Opportunity

III. Inventory Setup

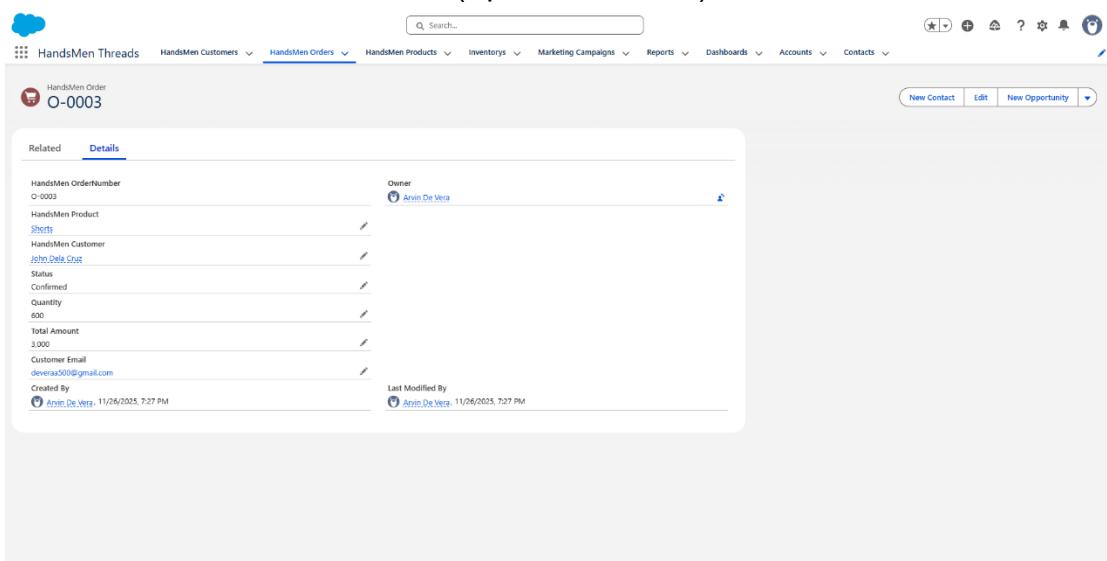
- a. **Inventory Number:** I-0002
- b. **HandsMen Product:** Shorts
- c. **Stock Quantity:** 1,500
- d. **Stock Status:** Available
- e. **Warehouse:** HandsMen Shorts Warehouse



The screenshot shows the HandsMen Threads software interface with the 'Inventory' tab selected. The main area displays a single record for an inventory item with the number I-0002. The details shown are: Inventory Number (I-0002), HandsMen Product (Shorts), Stock Quantity (1,500), Stock Status (Available), and Warehouse (HandsMen Shorts Warehouse). The record was created by Arvin De Vera on 11/26/2025, 7:24 PM. The interface includes a top navigation bar with links like 'HandsMen Threads', 'HandsMen Customers', 'HandsMen Orders', 'HandsMen Products', 'Inventory', 'Marketing Campaigns', 'Reports', 'Dashboards', 'Accounts', and 'Contacts'. A bottom navigation bar offers options for 'New Contact', 'Edit', and 'New Opportunity'.

IV. Order Creation

- a. **Order:** O-0003
- b. **HandsMen Product:** Shorts
- c. **HandsMen Customer:** John Dela Cruz
- d. **Status:** Confirmed
- e. **Quantity:** 600
- f. **Total Amount:** 3,000 (Apex calculation)

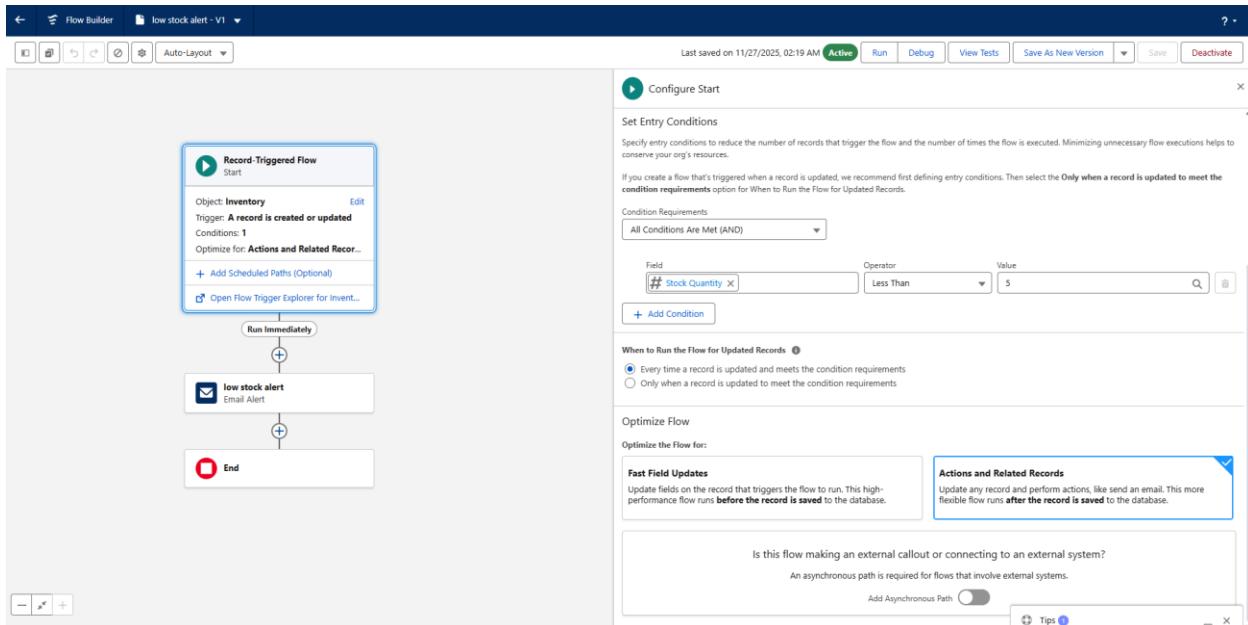


The screenshot shows the HandsMen Threads software interface with the 'HandsMen Orders' tab selected. The main area displays a single record for an order with the number O-0003. The details shown are: HandsMen OrderNumber (O-0003), HandsMen Product (Shorts), HandsMen Customer (John Dela Cruz), Status (Confirmed), Quantity (600), and Total Amount (3,000). The record was created by Arvin De Vera on 11/26/2025, 7:27 PM. The interface includes a top navigation bar with links like 'HandsMen Threads', 'HandsMen Customers', 'HandsMen Orders', 'HandsMen Products', 'Inventory', 'Marketing Campaigns', 'Reports', 'Dashboards', 'Accounts', and 'Contacts'. A bottom navigation bar offers options for 'New Contact', 'Edit', and 'New Opportunity'.

Business Automations (Flows)

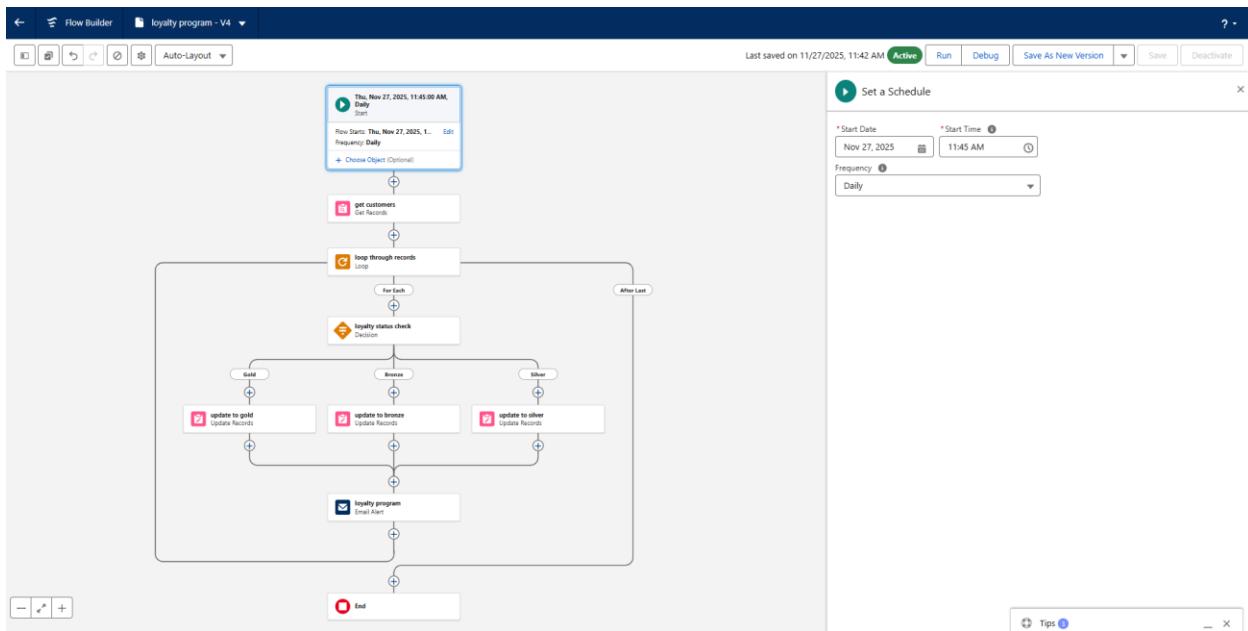
I. Low Stock Alert (Record-Triggered Flow)

- Trigger:** When Inventory is updated.
- Logic:** If Stock Quantity < 5 → send email.
- Email:** "Dear Inventory Manager, This is to inform you that the stock for the following product is running low: Product Name: {!Inventory__c.HandsMen_Product__c} Current Stock Quantity: {!Inventory__c.Stock_Quantity__c} Please take the necessary steps to restock this item immediately. Best Regards, Inventory Monitoring System"



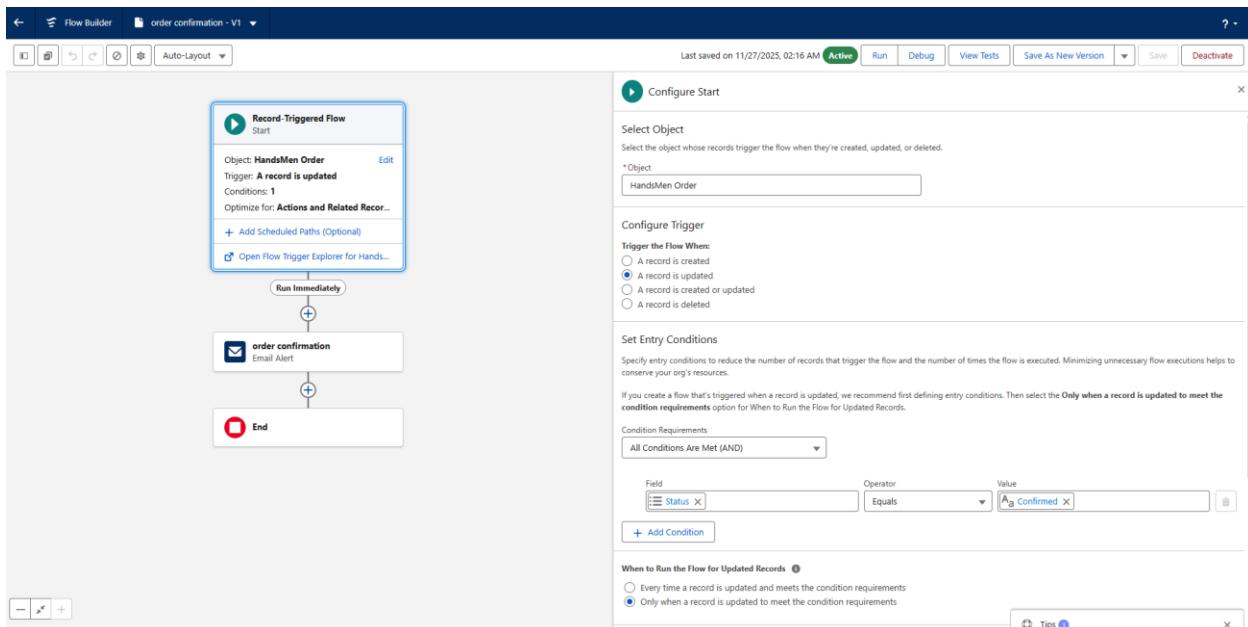
II. Loyalty Program (Scheduled Flow)

- Runs:** Daily at 11:45 AM
- Logic:**
 - >1000 → **Gold**
 - 501–1000 → **Silver**
 - ≤500 → **Bronze**
- Email:** "Congratulations! You are now a Gold member and you are eligible for our Loyalty Rewards Program. Enjoy exclusive discounts, early access to offers, and special member benefits. Thank you for your continued Support."



III. Order Confirmation Flow

- Trigger:** Order Status → Confirmed
- Action:** Send confirmation email.
- Sample Text:** “Dear John, Your order #O-0002 has been confirmed! Thank you for shopping with us. Best Regards, Sales Team”



Apex Development

I. OrderTotalTrigger — Order Amount Calculation

- a. **Type:** before insert, before update on Handsman Orders
- b. **Logic:** $\text{order.Total_Amount_c} = \text{order.Quantity_c} * \text{product.Price_c}$; $\text{Total_Amount_c} = \text{Product.Price_c} \times \text{Quantity_c}$



```
File • Edit • Debug • Test • Workspace • Help • < >
OrderTotalTrigger.apex StockReductionTrigger.apex OrderTrigger.apex InventoryBatchJob.apxc OrderTriggerHandler.apxc
Code Coverage: None • API Version: 65 • Go To
1 * trigger OrderTotalTrigger on HandsMen_Order__c (before insert, before update) {
2     Set<Id> productIds = new Set<Id>();
3
4     for (HandsMen_Order__c order : Trigger.new) {
5         if (order.HandsMen_Product__c != null) {
6             productIds.add(order.HandsMen_Product__c);
7         }
8     }
9
10    Map<Id, HandsMen_Product__c> productMap = new Map<Id, HandsMen_Product__c>(
11        [SELECT Id, Price__c FROM HandsMen_Product__c WHERE Id IN :productIds]
12    );
13
14    for (HandsMen_Order__c order : Trigger.new) {
15        if (order.HandsMen_Product__c != null && productMap.containsKey(order.HandsMen_Product__c)) {
16            HandsMen_Product__c product = productMap.get(order.HandsMen_Product__c);
17            if (order.Quantity__c != null) {
18                order.Total_Amount__c = order.Quantity__c * product.Price__c;
19            }
20        }
21    }
22 }
```

II. OrderTrigger—Quantity Validation Dispatcher

- a. **Type:** before insert, before update
- b. **Logic:**
 - i. **Confirmed** → Quantity > 500
 - ii. **Pending** → Quantity > 200
 - iii. **Rejection** → Quantity = 0

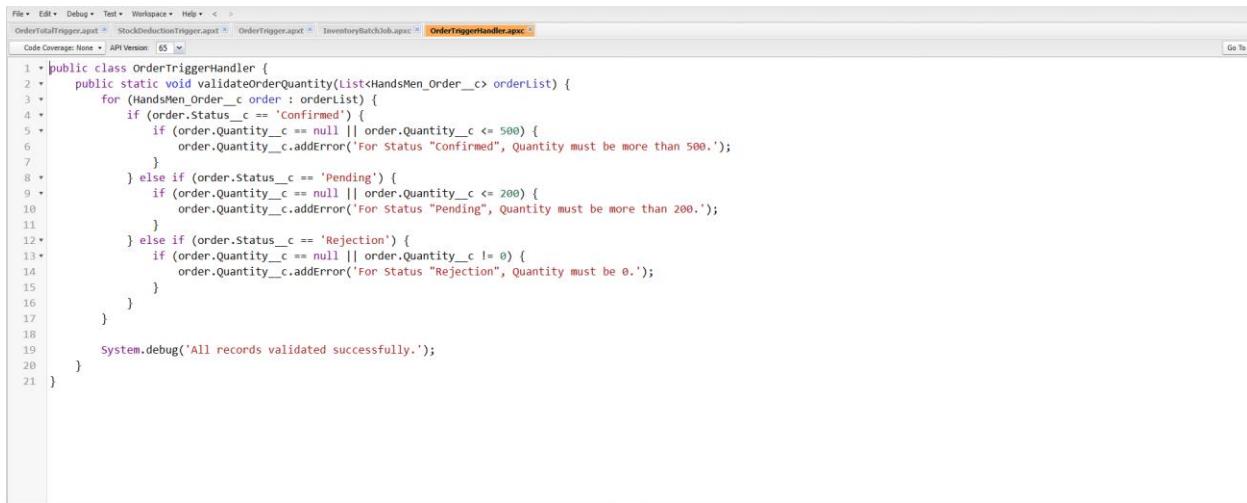


```
File • Edit • Debug • Test • Workspace • Help • < >
OrderTotalTrigger.apex StockReductionTrigger.apex OrderTrigger.apex InventoryBatchJob.apxc OrderTriggerHandler.apxc
Code Coverage: None • API Version: 65 • Go To
1 * trigger OrderTrigger on HandsMen_Order__c (before insert, before update) {
2     if (Trigger.isBefore && (Trigger.isInsert || Trigger.isUpdate)) {
3         OrderTriggerHandler.validateOrderQuantity(Trigger.new);
4     }
5 }
```

III. OrderTriggerHandler — Quantity Validation

a. Business Rules:

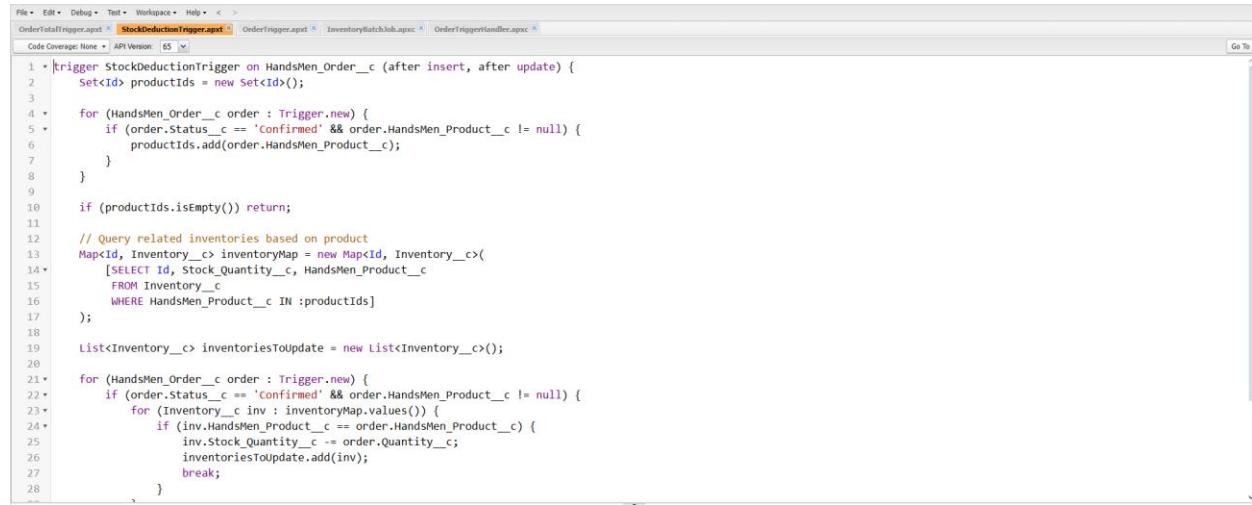
Status	Required Quantity
Confirmed	> 500
Pending	> 200



```
File • Edit • Debug • Test • Workspace • Help • < >
OrderTotalTrigger.apxt StockDeductionTrigger.apxt OrderTrigger.apxt InventoryBatchJob.apxc OrderTriggerHandler.apxc
Code Coverage: None • API Version: 65 Go To
1 public class OrderTriggerHandler {
2     public static void validateOrderQuantity(List<HandsMen_Order__c> orderList) {
3         for (HandsMen_Order__c order : orderList) {
4             if (order.Status__c == 'Confirmed') {
5                 if (order.Quantity__c == null || order.Quantity__c <= 500) {
6                     order.Quantity__c.addError('For status "Confirmed", Quantity must be more than 500.');
7                 }
8             } else if (order.Status__c == 'Pending') {
9                 if (order.Quantity__c == null || order.Quantity__c <= 200) {
10                     order.Quantity__c.addError('For status "Pending", Quantity must be more than 200.');
11                 }
12             } else if (order.Status__c == 'Rejection') {
13                 if (order.Quantity__c == null || order.Quantity__c != 0) {
14                     order.Quantity__c.addError('For status "Rejection", Quantity must be 0.');
15                 }
16             }
17         }
18         System.debug('All records validated successfully.');
19     }
20 }
21 }
```

III. StockDeductionTrigger — Inventory Deduction

- a. Type: after insert, after update on Inventory
- b. Logic: inv.Stock_Quantity__c -= order.Quantity__c;



```
File • Edit • Debug • Test • Workspace • Help • < >
OrderTotalTrigger.apxt StockDeductionTrigger.apxt OrderTrigger.apxt InventoryBatchJob.apxc OrderTriggerHandler.apxc
Code Coverage: None • API Version: 65 Go To
1 trigger StockDeductionTrigger on HandsMen_Order__c (after insert, after update) {
2     Set<Id> productids = new Set<Id>();
3
4     for (HandsMen_Order__c order : Trigger.new) {
5         if (order.Status__c == 'Confirmed' && order.HandsMen_Product__c != null) {
6             productids.add(order.HandsMen_Product__c);
7         }
8     }
9
10    if (productids.isEmpty()) return;
11
12    // Query related inventories based on product
13    Map<Id, Inventory__c> inventoryMap = new Map<Id, Inventory__c>(
14        [SELECT Id, Stock_Quantity__c, HandsMen_Product__c
15         FROM Inventory__c
16         WHERE HandsMen_Product__c IN :productids]
17    );
18
19    List<Inventory__c> inventoriesToUpdate = new List<Inventory__c>();
20
21    for (HandsMen_Order__c order : Trigger.new) {
22        if (order.Status__c == 'Confirmed' && order.HandsMen_Product__c != null) {
23            for (Inventory__c inv : inventoryMap.values()) {
24                if (inv.HandsMen_Product__c == order.HandsMen_Product__c) {
25                    inv.Stock_Quantity__c -= order.Quantity__c;
26                    inventoriesToUpdate.add(inv);
27                    break;
28                }
29            }
30        }
31    }
32 }
```

IV. InventoryBatchJob — Batch Apex

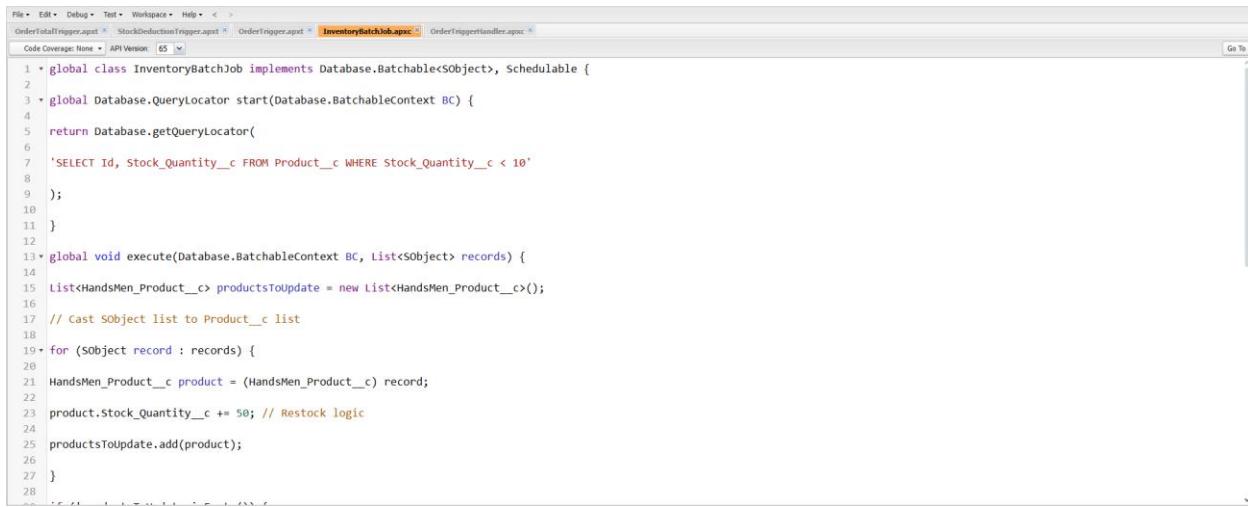
a. Implements:

- i. Database.Batchable
- ii. Schedulable

b. **Query:** Select products where stock < 18.

c. Purpose:

- i. Bulk restocking
- ii. Reporting



The screenshot shows a Salesforce code editor window with the following details:

- File Menu:** File, Edit, Debug, Test, Workspace, Help.
- Toolbars:** Standard toolbar.
- Code Coverage:** None.
- API Version:** 65.
- Open Files:** OrderTotalTrigger.apc, StockDeletionTrigger.apc, OrderTrigger.apc, **InventoryBatchJob.apc** (highlighted in orange), and OrderTriggerHandler.apc.
- Code Content:**

```
1 global class InventoryBatchJob implements Database.Batchable<SObject>, Schedulable {
2
3     global Database.QueryLocator start(Database.BatchableContext BC) {
4
5         return Database.getQueryLocator(
6
7             'SELECT Id, Stock_Quantity__c FROM Product__c WHERE Stock_Quantity__c < 10'
8
9         );
10    }
11
12    global void execute(Database.BatchableContext BC, List<SObject> records) {
13
14        List<HandsMen_Product__c> productsToUpdate = new List<HandsMen_Product__c>();
15
16        // Cast SObject list to Product__c list
17
18        for (SObject record : records) {
19            HandsMen_Product__c product = (HandsMen_Product__c) record;
20
21            product.Stock_Quantity__c += 50; // Restock logic
22
23            productsToUpdate.add(product);
24
25        }
26    }
27}
```

Email Templates

I. Low Stock Alert

- a. **Subject:** Current Stock Quantity: {!Inventory__c.Stock_Quantity__c}
- b. **Body sample:** Dear Inventory Manager, This is to inform you that the stock for the following product is running low: Product Name: {!Inventory__c.HandsMen_Product__c} Current Stock Quantity: {!Inventory__c.Stock_Quantity__c} Please take the necessary steps to restock this item immediately. Best Regards, Inventory Monitoring System

The screenshot shows the 'Email Template Detail' page for a template named 'Low Stock Alert'. The 'Email Template Detail' section includes fields for 'Email Templates from Salesforce' (Unified Public Classic Email Templates), 'Email Template Name' (Low Stock Alert), 'Template Unique Name' (Low_Stock_Alert), 'Encoding' (Unicode (UTF-8)), 'Author' (Avin De Vera [Change]), 'Description' (None), and 'Created By' (Avin De Vera, 11/28/2025, 7:54 AM). The 'Available For Use' checkbox is checked. The 'Email Template' section contains the email body with a subject line 'Low Stock Alert Email' and a preview message: 'Dear Inventory Manager, This is to inform you that the stock for the following product is running low: Product Name: {!Inventory__c.HandsMen_Product__c} Current Stock Quantity: {!Inventory__c.Stock_Quantity__c}. Please take the necessary steps to restock this item immediately. Best Regards, Inventory Monitoring System'. The 'Attachments' section shows 'No records to display'.

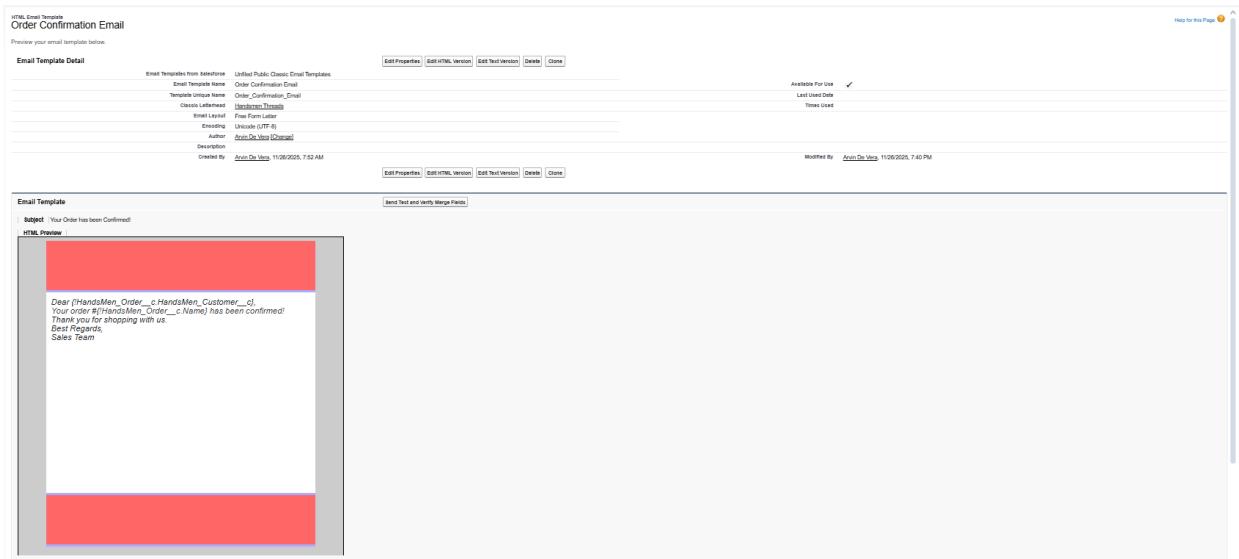
II. Loyalty Status Update

- a. **Subject:** Congratulations! You are now a {!HandsMen_Customer__c.Loyalty_Status__c} member
- b. **Body sample:** Congratulations! You are now a {!HandsMen_Customer__c.Loyalty_Status__c} member and you are eligible for our Loyalty Rewards Program. Enjoy exclusive discounts, early access to offers, and special member benefits. Thank you for your continued support

The screenshot shows the 'Email Template Detail' page for a template named 'Loyalty Program Email'. The 'Email Template Detail' section includes fields for 'Email Templates from Salesforce' (Unified Public Classic Email Templates), 'Email Template Name' (Loyalty_Program_Email), 'Template Unique Name' (Loyalty_Program_Email), 'Description' (None), and 'Created By' (Avin De Vera, 11/28/2025, 7:54 AM). The 'Available For Use' checkbox is checked. The 'Email Template' section contains the email body with a subject line 'Loyalty Program Email' and a preview message: 'Congratulations! You are now a {!HandsMen_Customer__c.Loyalty_Status__c} member and you are eligible for our Loyalty Rewards Program. Enjoy exclusive discounts, early access to offers, and special member benefits. Thank you for your continued support.' The 'Attachments' section shows 'No records to display'.

III. Order Confirmation

- a. **Subject:** Your order # {!HandsMen_Order__c.Name} has been confirmed!
- b. **Body sample:** Dear {!HandsMen_Order__c.HandsMen_Customer__c},
Your order # {!HandsMen_Order__c.Name} has been confirmed! Thank you for shopping with us. Best Regards, Sales Team



Conclusion

The HandsMen Threads CRM successfully streamlines bespoke tailoring operations by integrating customer management, order processing, inventory tracking, loyalty automation, and Apex-driven logic into one cohesive platform. Through the combined use of Flows, Validation Rules, Email Alerts, and Apex Triggers, the system delivers accurate data handling, reduced manual work, and enhanced customer engagement.

Overall, the solution demonstrates how Salesforce can effectively support and optimize business processes for a modern, custom-tailoring business.