

HandsMen Threads

Elevating the Art of Sophistication in Men's Fashion
Capstone Project — Salesforce Demo Documentation

Author: Principio, Angelo M.

Date: 11/18/2025

Environment: Salesforce Developer Org / Lightning Experience

Table of Contents

1. Project Overview
2. Custom App & Architecture
3. Data Model and Key Fields
4. Creating Data Records Step by Step
5. Business Automations Flows and Scheduled Processes
6. Apex Triggers and Batch Jobs Summaries
7. Email Templates Placeholders
8. Testing Validation and QA Scenarios
9. Reports and Dashboards Placeholders
10. Deployment Notes and Permissions
11. Future Scope and Recommendations
12. Appendix Sample SOQL and Test Checklist

Project Overview

Project name: HandsMen Threads

Description: A custom Salesforce CRM to manage bespoke men's tailoring: customers, products, inventory, orders, loyalty tiers, and automated communication.

Business goals

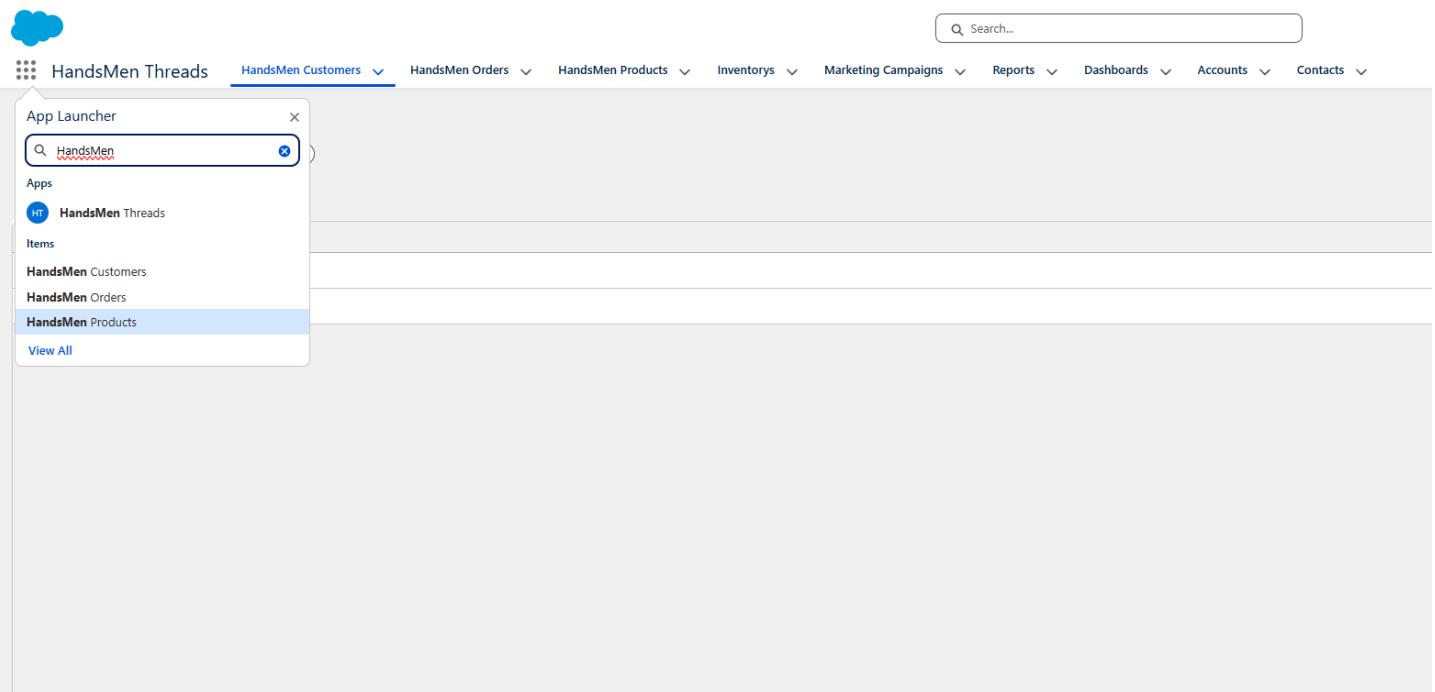
- Streamline tailoring workflows from inquiry to delivery
- Improve customer experience with personalized interactions and loyalty rewards
- Ensure accurate inventory and order management to avoid stock issues
- Automate routine processes to reduce manual work and accelerate fulfillment

Salesforce tools used :

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

Script for Project Overview tab

HandsMen Threads is a custom Salesforce CRM solution designed to elevate men's bespoke fashion management. The goal of this project is to streamline custom tailoring operations, enhance customer experience with personalized interactions, optimize order and inventory management, and automate business processes using Salesforce capabilities. This platform was built using Custom Objects, Flows, Validation Rules, Apex Triggers, Email Alerts, and Reports & Dashboards. This demo walks through the design, configuration, automation, and end-to-end functionality of the CRM.



Custom App and Architecture

Custom App: HandsMen Threads Lightning App — central hub for operations and admin

tasks **Assigned profiles:** System Administrator, Sales, Service, Inventory Manager

Object architecture

Custom Objects

- Handsman Customer — customer name, email, phone, loyalty status, total purchases
- Handsman Orders — order number, product, customer, status, quantity, total amount, customer email
- Handsman Products — SKU, price, name, stock quantity
- Inventory — product, stock quantity, warehouse location
- Marketing Campaigns — campaign number, customer, start date, end date

Standard Objects used

- Accounts and Contacts
- Reports and Dashboards

Data flow

Customer → Orders → Product → Inventory. Orders update Inventory when confirmed; scheduled processes reconcile pending orders with stock.

This screenshot shows the 'Details' tab of a HandsMen Customer record for 'John Dela Cruz'. The record includes fields for Name, Email, Phone, Loyalty Status, and various purchase details. It also shows the creation and modification history.

Field	Value
HandsMen Customer Name	John Dela Cruz
Email	john@gmail.com
Phone	09123456789
Loyalty Status	Silver
FirstName	John
LastName	Dela Cruz
Total Purchases	100
Created By	Angelo Principio, 11/17/2025, 9:05 AM
Last Modified By	Angelo Principio, 11/17/2025, 9:20 AM

This screenshot shows the 'Details' tab of a HandsMen Order record for 'O-0007'. The record includes fields for OrderNumber, Product, Customer, Status, Quantity, Total Amount, and Customer Email. It also shows the creation and modification history.

Field	Value
HandsMen OrderNumber	O-0007
HandsMen Product	Shorts
HandsMen Customer	John Dela Cruz
Status	Confirmed
Quantity	600
Total Amount	1,800
Customer Email	principioangelo24@gmail.com
Created By	Angelo Principio, 11/17/2025, 9:07 AM
Last Modified By	Angelo Principio, 11/17/2025, 9:08 AM

This screenshot shows the 'Details' tab of a HandsMen Product record for 'Shorts'. The record includes fields for Order Name, SKU, Price, Stock Quantity, and Creation history.

Field	Value
HandsMen Order Name	Shorts
SKU	0002
Price	\$3
Stock Quantity	1,000
Created By	Angelo Principio, 11/17/2025, 9:06 AM
Last Modified By	Angelo Principio, 11/17/2025, 9:06 AM

The screenshot shows a Salesforce interface for managing inventory. At the top, there's a navigation bar with links for HandsMen Threads, Customers, Orders, Products, Inventory (which is currently selected), Marketing Campaigns, Reports, Dashboards, Accounts, and Contacts. Below the navigation is a header bar with the title "Inventory" and the item number "I -0006". The main content area is titled "Details" and contains the following fields:

- Inventory Number: I -0006
- HandsMen Product: Shorts
- Stock Quantity: 1,400
- Stock Status: Available
- Warehouse: Warehouse Shorts
- Created By: Angelo Principio, 11/17/2025, 9:06 AM
- Last Modified By: Angelo Principio, 11/17/2025, 9:08 AM

Data Model and Key Fields

Handsman Customer

- FullName
- Email
- Phone
- Loyalty_Status__c
- Total_Purchases__c

Handsman Products

- SKU__c
- Name
- Price__c
- Stock_Quantity__c

Inventory

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

Handsman Orders

- Name (Order Number)
- HandsMen_Customer__c (lookup)
- Customer_Email__c
- HandsMen_Product__c (lookup)
- Quantity__c
- Status__c (Pending Confirmed Cancelled)

- Total_Amount__c
- Order_Date__c

Validation rules examples

Email must contain "gmail.com" → Error text: Please fill correct Gmail

The screenshot shows the 'Validation Rule Edit' screen for the 'HandsMen Customer' object. The 'Rule Name' is 'Email'. The 'Active' checkbox is checked. The 'Error Condition Formula' is set to 'NOT CONTAINS(Email__c , "@gmail.com")'. The 'Error Message' is 'Please fill correct Gmail'. The 'Save' button is visible at the bottom.

Creating Data Records Step by Step

1 Create Handsman Customer

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.

The screenshot shows the 'New HandsMen Customer' form. The 'Email' field contains 'johndc@xyz.com'. A validation error message box is displayed at the bottom right, stating 'We hit a snag.' and 'Review the errors on this page.' with the sub-item '• Please fill correct Gmail'. The 'Save' button is visible at the bottom right of the form.

2 Create Handsman Product

Fields: Order Name: Shorts; SKU 0002; Price \$3; Quantity 1000;

The screenshot shows a software interface for creating a new product. The top navigation bar includes links for 'HandsMen Threads', 'HandsMen Customers', 'HandsMen Orders', 'HandsMen Products' (which is underlined in blue, indicating it's the active tab), and 'Inventorys'. Below the navigation is a header section with a pink icon of a clipboard and the text 'HandsMen Product' followed by the product name 'Shorts'. The main content area is titled 'Details' and contains the following fields:

HandsMen Order Name	Shorts	Owner
SKU	0002	
Price	\$3	
Stock Quantity	1,000	
Created By	Angelo Principio, 11/17/2025, 9:06 AM	Last Mo

3 Create Inventory record

New Product → HandsMen Product Shorts → Stock Quantity 1400 → Warehouse Shorts.

The screenshot shows a software interface for creating a new inventory record. The top navigation bar includes links for 'HandsMen Threads', 'HandsMen Customers', 'HandsMen Orders', 'HandsMen Products', and 'Inventorys' (which is underlined in blue). Below the navigation is a header section with a pink icon of a clipboard and the text 'Inventory' followed by the inventory number 'I -0006'. The main content area is titled 'Details' and contains the following fields:

Inventory Number	I -0006
HandsMen Product	Shorts
Stock Quantity	1,400
Stock Status	Available
Warehouse	Warehouse Shorts
Created By	Angelo Principio, 11/17/2025, 9:06 AM

4 Create an Order

Handsman Orders → New → select Customer Product Quantity → Save → Total Amount auto-calculates via Apex.

The screenshot shows the HandsMen Orders detail page for Order O-0007. The page has a header with navigation links: HandsMen Threads, HandsMen Customers, HandsMen Orders (selected), HandsMen Products, Inventory, Marketing Campaigns, Reports, Dashboards, Accounts, and Contacts. Below the header, the order details are listed:

Field	Value	Action
HandsMen OrderNumber	O-0007	
Owner	Angelo Principio	
HandsMen Product	Shorts	
HandsMen Customer	John Dela Cruz	
Status	Confirmed	
Quantity	600	
Total Amount	1,800	
Customer Email	principioangelo24@gmail.com	
Created By	Angelo Principio, 11/17/2025, 9:07 AM	
Last Modified By	Angelo Principio, 11/17/2025, 9:08 AM	

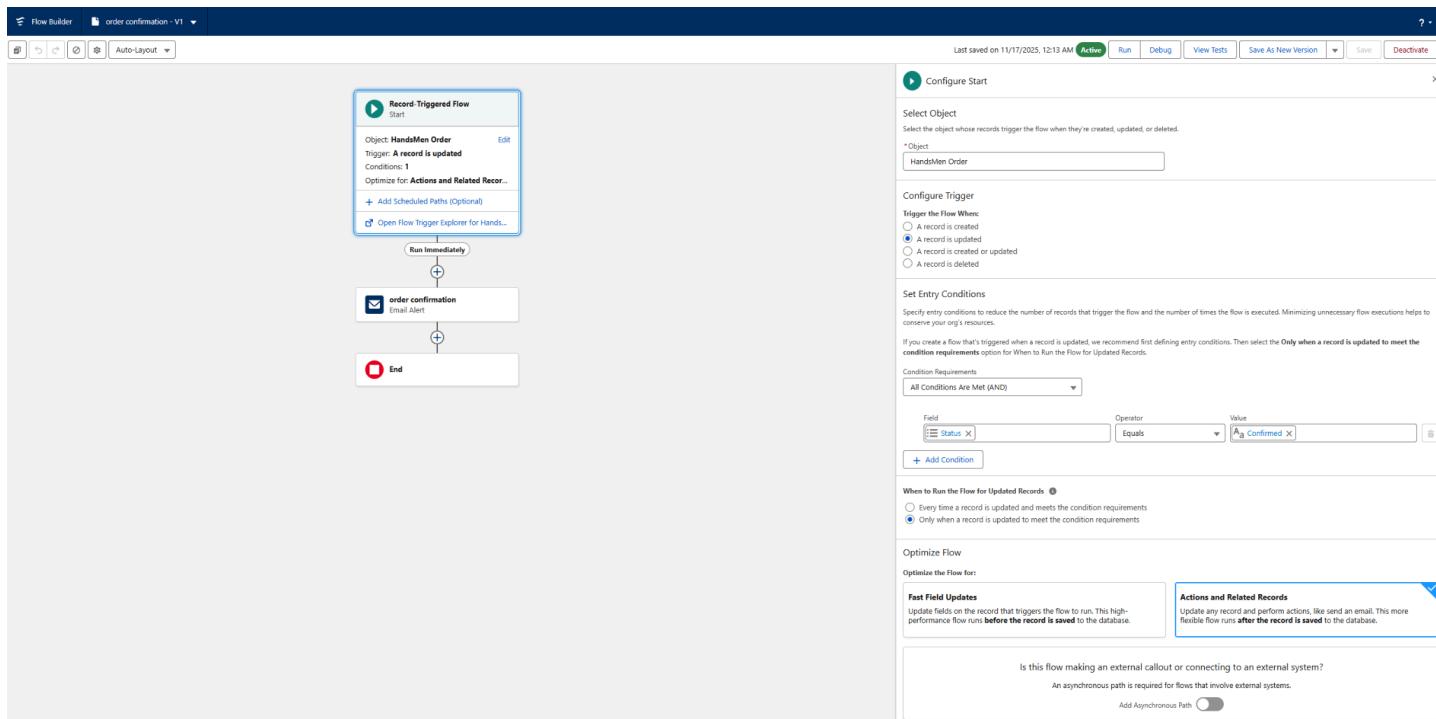
Business Automations Flows and Scheduled Processes

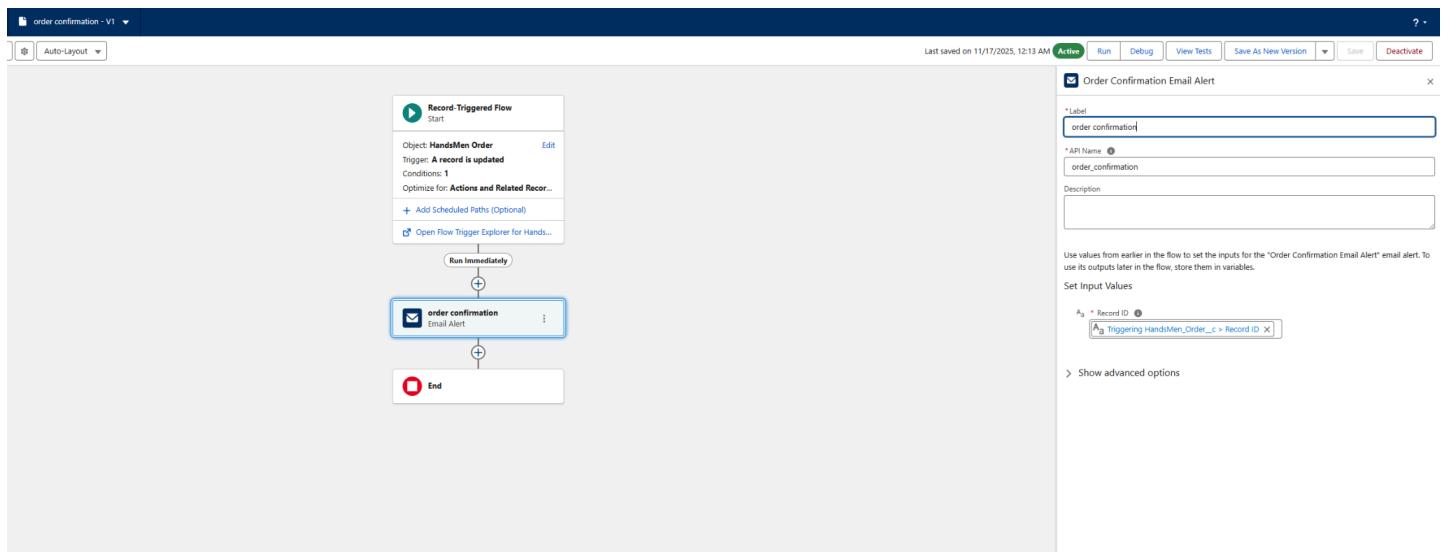
This section details automations to demonstrate system intelligence.

Automated Order Confirmation Record Triggered Flow

Trigger: Order.Status changes Pending → Confirmed

Actions: Send confirmation email to customer; update Inventory.Stock_Available minus Quantity Demo example: John DC orders 600 shorts; stock 2000 → new stock 1400; confirmation email sent.





Low Stock Alert Record Triggered Flow

Trigger: Inventory update

Condition: Stock_Quantity < 5

Actions: Send low-stock email to inventory manager

Sample email:

Dear Inventory Manager,

This is to inform you that the stock for the following product is running low:

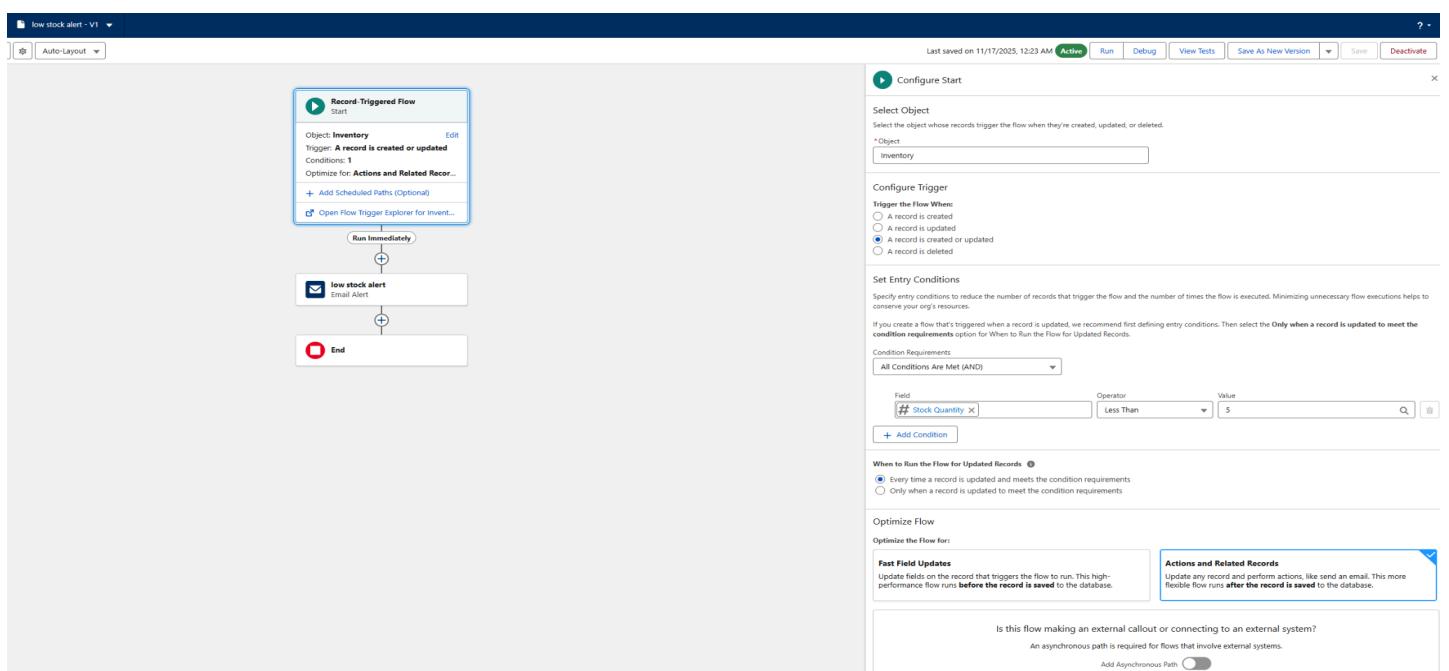
Product Name: {!Inventory__c.HandsMen_Product__c}

Current Stock Quantity: {!Inventory__c.Stock_Quantity__c}

Please take the necessary steps to restock this item immediately.

Best Regards,

Inventory Monitoring System





SETUP

Classic Email Templates

[Edit Text Email Template](#)

Low Stock Alert

Use merge fields to personalize your email content. You can add substitute text to any merge field. Substitute text displays only if the merge record does not contain data for that field. Enter substitute text after a comma in the merge field value: {!NullValue(Contact.FirstName,"Sir or Madam")}. Click on the link below to see a sample email template.

[View Sample Template](#)

Note that the Description field is for internal use only. It will be listed as the title of any email activities you log when sending mass email.

Available Merge Fields

Select Field Type	Select Field	Copy Merge Field Value
Contact Fields		

Copy and paste the merge field value into your template below.

Email Template Edit

Email Template Information

Folder	Unfiled Public Classic Email Templates
Available For Use	<input checked="" type="checkbox"/>
Email Template Name	Low Stock Alert
Template Unique Name	Low_Stock_Alert
Encoding	Unicode (UTF-8)
Description	
Subject	Low Stock Alert Email
Email Body	<p>Dear Inventory Manager,</p> <p>This is to inform you that the stock for the following product is running low:</p> <p>Product Name: {!Inventory__c.HandsMen_Product__c}</p> <p>Current Stock Quantity: {!Inventory__c.Stock_Quantity__c}</p> <p>Please take the necessary steps to restock this item immediately.</p> <p>Best Regards, Inventory Monitoring System</p>

Actions

Save | Save & New | Cancel

Loyalty Program Scheduled Flow

Schedule: Daily

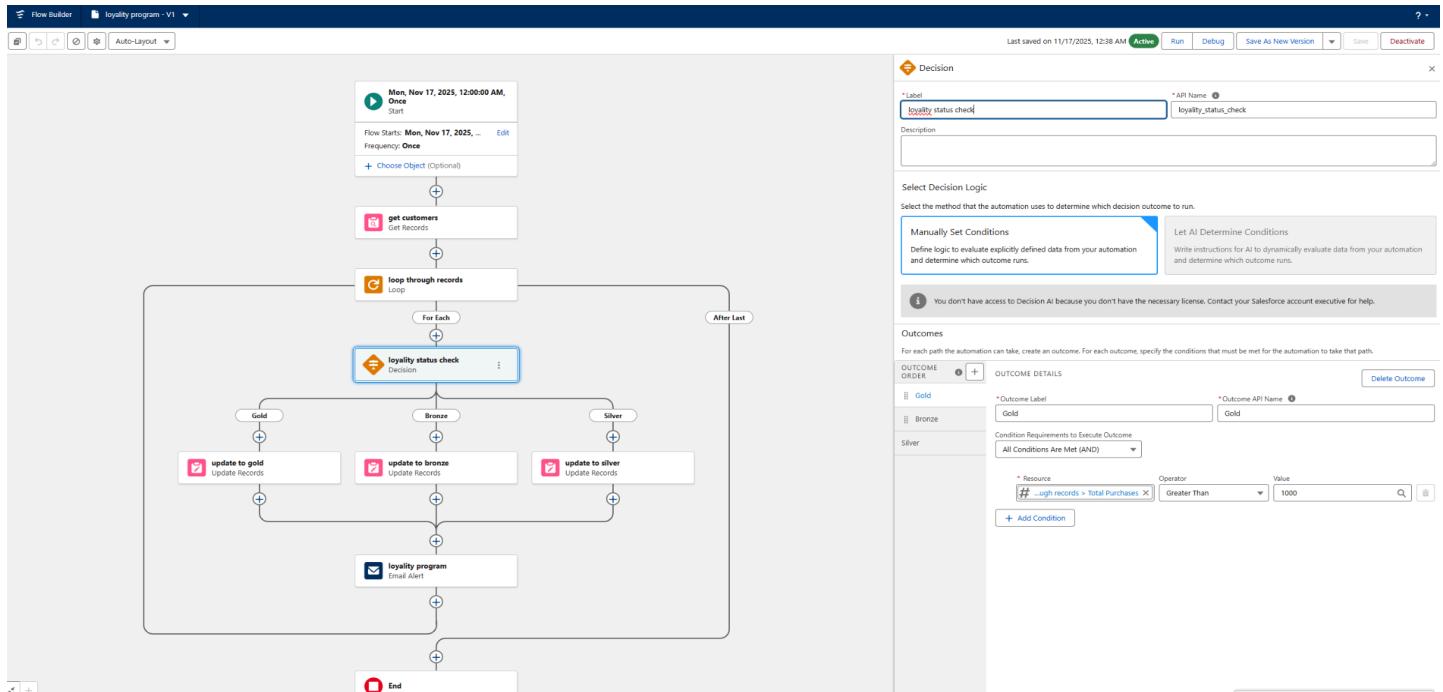
Logic: Total_Purchases thresholds set Loyalty_Status

1000 Gold

500–1000 Silver

≤ 500 Bronze

Actions: Update Loyalty_Status__c and send status update email

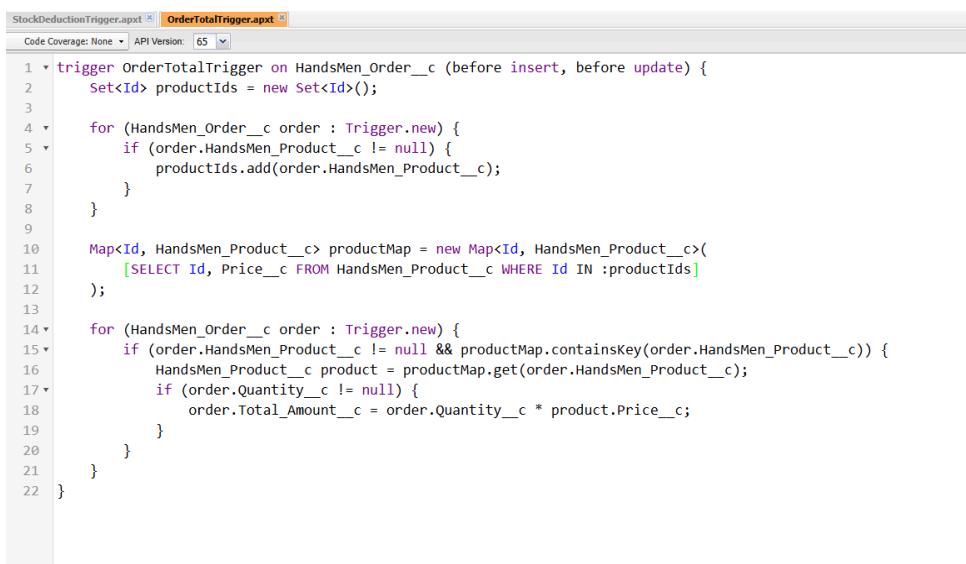


Apex Triggers and Batch Jobs Summaries

Order Total Calculation Trigger

When before insert before update on Handsman Orders

Function: $\text{order.Total_Amount_c} = \text{order.Quantity_c} * \text{product.Price_c}$; $\text{Total_Amount_c} = \text{Product.Price_c} * \text{Quantity_c}$



The screenshot shows the Salesforce Apex code editor with the tab 'OrderTotalTrigger.apxt' selected. The code is a trigger named 'OrderTotalTrigger' on the 'HandsMen_Order__c' object. It performs two main operations: 1) It gathers all product IDs from the new orders and stores them in a set. 2) It performs a query to get the price for each product ID and stores it in a map. Finally, it iterates through the new orders again and calculates the total amount by multiplying quantity and price.

```
trigger OrderTotalTrigger on HandsMen_Order__c (before insert, before update) {
    Set<Id> productIds = new Set<Id>();
    for (HandsMen_Order__c order : Trigger.new) {
        if (order.HandsMen_Product__c != null) {
            productIds.add(order.HandsMen_Product__c);
        }
    }
    Map<Id, HandsMen_Product__c> productMap = new Map<Id, HandsMen_Product__c>(
        [SELECT Id, Price__c FROM HandsMen_Product__c WHERE Id IN :productIds]
    );
    for (HandsMen_Order__c order : Trigger.new) {
        if (order.HandsMen_Product__c != null && productMap.containsKey(order.HandsMen_Product__c)) {
            HandsMen_Product__c product = productMap.get(order.HandsMen_Product__c);
            if (order.Quantity__c != null) {
                order.Total_Amount__c = order.Quantity__c * product.Price__c;
            }
        }
    }
}
```

Stock Deduction Trigger

Use-case: Automatic deduction of stocks

Function: inv.Stock_Quantity__c -= order.Quantity__c;

```
StockDeductionTrigger.apxt OrderTotalTrigger.apxt
Code Coverage: None API Version: 65
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
33    if (!inventoriesToUpdate.isEmpty()) {
34        update inventoriesToUpdate;
35    }
36 }
```

Email Templates Placeholders

Order Confirmation Email

Subject: Your order # {!HandsMen_Order__c.Name} has been confirmed!

Body sample:

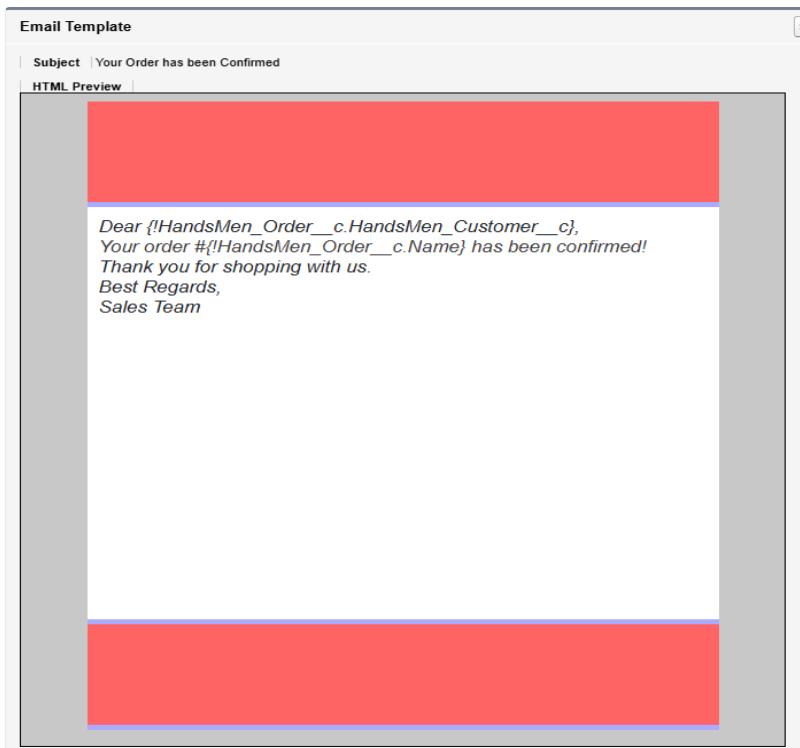
Dear {!HandsMen_Order__c.HandsMen_Customer__c},

Your order # {!HandsMen_Order__c.Name} has been confirmed!

Thank you for shopping with us.

Best Regards,

Sales Team



Low Stock Alert Email

Subject: Current Stock Quantity: {!Inventory__c.Stock_Quantity__c}

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

Email Template

Subject | Low Stock Alert Email

Plain Text Preview

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

Loyalty Status Update Email

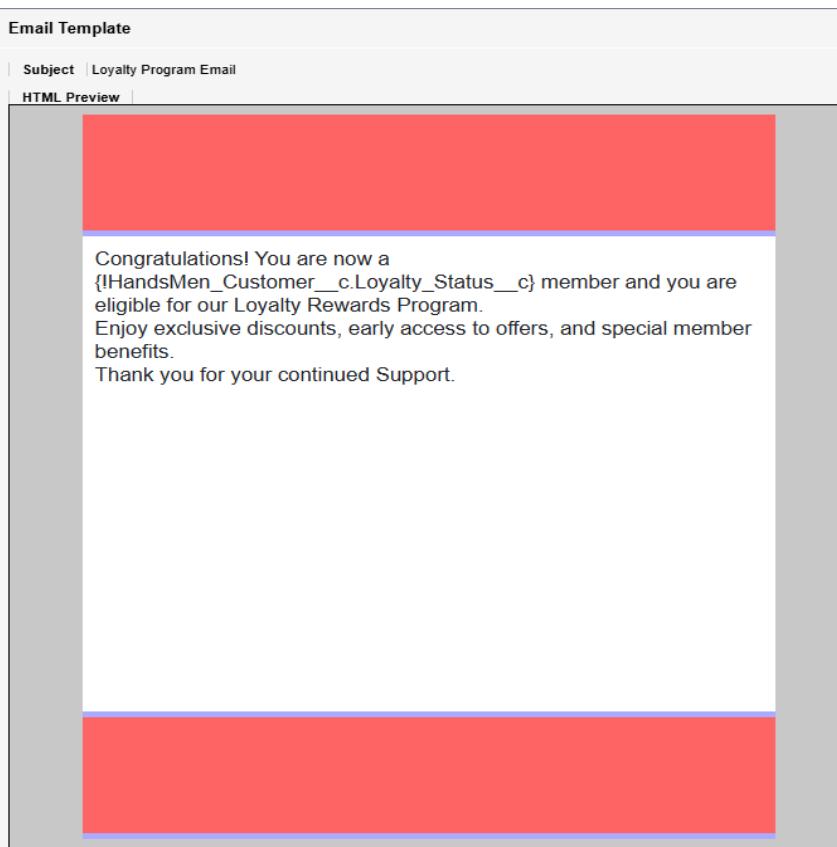
Subject: Congratulations! You are now a {!HandsMen_Customer__c.Loyalty_Status__c} member

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.



Conclusion

The HandsMen Threads CRM successfully integrates customer management, order lifecycle, inventory tracking, loyalty programs, smart automation, email communication, and Apex logic for advanced processing.

This project showcases how Salesforce CRM can streamline bespoke tailoring operations, improve customer experience, and provide real-time data insights for the men's fashion industry.