

HandsMen Threads – Salesforce CRM Project Documentation

1. Project Overview

HandsMen Threads is a Salesforce CRM solution built for a luxury men's fashion brand. It enhances customer engagement and automates backend processes such as loyalty tracking, order processing, and inventory management.

Built using Lightning components, Flows, Apex Triggers, and Batch Apex, this project ensures seamless business logic execution, enabling real-time data visibility, streamlined workflows, and improved customer retention through loyalty programs.

2. Objectives

- Automate total amount calculation for each order.
- Dynamically assign loyalty tiers (Bronze, Silver, Gold) based on purchase history.
- Trigger automatic updates to customer total purchases.
- Maintain accurate inventory levels by syncing with order data.
- Eliminate manual errors using UI validation and backend automation.

3. Phase 1: Requirement Analysis & Planning

Understanding Business Requirements

- Maintain a product catalog with pricing and stock.
- Enable customers to place orders with automated amount calculation.
- Track customer loyalty and purchase history.
- Send real-time alerts on stock depletion.
- Manage marketing campaigns for promotions.

Defining Project Scope and Objectives

- Build custom objects: Product, Customer, Order, Inventory, Campaign.
- Automate loyalty calculation and stock deduction.
- Integrate workflows, validation, and Apex logic.
- Provide role-specific UI with dashboards and reports.

Data Model and Security Model

- Custom objects with defined relationships.
- Role-based page layouts and tabs.
- Record-level access through roles and sharing rules.

4. Phase 2: Salesforce Development – Backend & Configurations

Custom Objects:

Object Name	Type	Description	Key Fields
<i>HandsMen_Customer__c</i>	Custom Object	Stores customer details	Name (Record Name), Email (Email), Phone (Phone), Loyalty_Status__c (Picklist: Bronze, Gold, Silver), Total_Purchases__c (Number)
<i>HandsMen_Product__c</i>	Custom Object	Stores product catalog	Name (Record Name), SKU (Text), Price (Currency), Stock_Quantity__c (Number)
<i>HandsMen_Order__c</i>	Custom Object	Stores customer orders	Order_Number (Record Name), Status (Picklist: Pending, Confirmed, Rejection), Quantity__c (Number), Total_Amount__c (Number)
<i>Inventory__c</i>	Custom Object	Tracks inventory levels	Auto Number (Record Name), Warehouse (Text), Stock_Quantity__c (Number)
<i>Marketing_Campaign__c</i>	Custom Object	Manages promotions & campaigns	Campaign_Name (Record Name), Start_Date (Date), End_Date (Date)

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main area is titled 'Object Manager' with a sub-header '52+ items. Sorted by Deployed'. The table below lists the custom objects:

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Marketing Campaign	Marketing_Campaign__c	Custom Object		7/26/2025	✓
Inventory	Inventory__c	Custom Object		7/26/2025	✓
HandsMen Order	HandsMen_Order__c	Custom Object		7/26/2025	✓
HandsMen Product	HandsMen_Product__c	Custom Object		7/26/2025	✓
HandsMen Customer	HandsMen_Customer__c	Custom Object		7/26/2025	✓

Validation Rules and Logic:

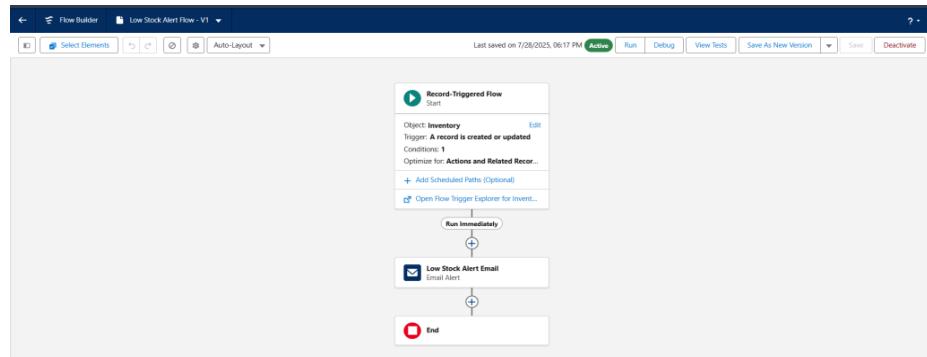
Object	Field	Validation Rule	Description
<i>HandsMen_Order_c</i>	<i>Total_Amount_c</i>	<i>Total_Amount_c <= 0</i>	Prevents saving an order if the total amount is negative.
<i>Inventory_c</i>	<i>Stock_Quantity_c</i>	<i>Stock_Quantity_c <= 0</i>	Ensures inventory stock is never negative
<i>HandsMen_Customer_c</i>	Email	NOT CONTAINS>Email, "@gmail.com")	Restricts email addresses to only Gmail domains.

Automation:

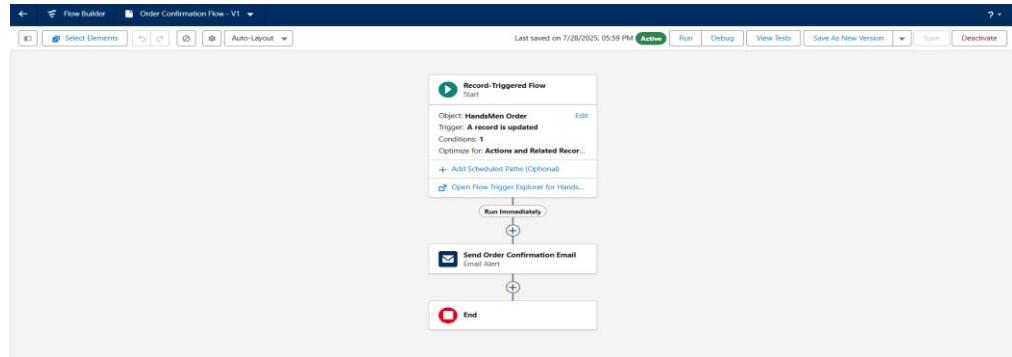
Flows:

- Loyalty Status Flow (updates tier based on total purchase):

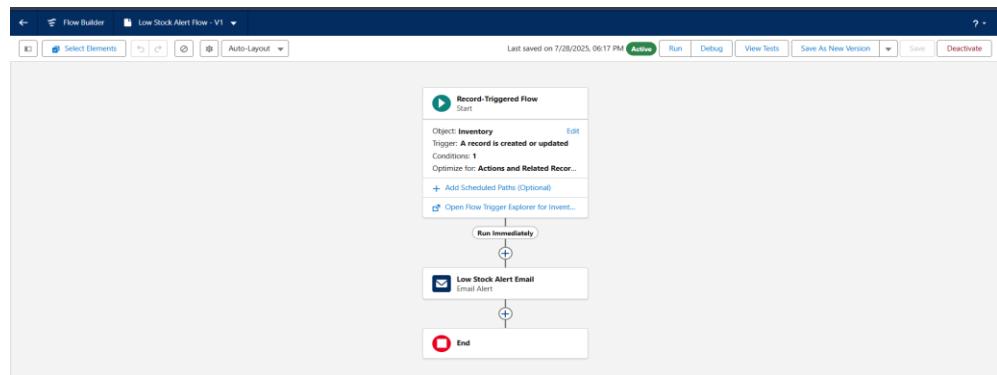
Loyalty Tier	Condition (Total Purchases)
<i>Gold</i>	Greater than \$1000
<i>Silver</i>	Between \$500 and \$1000
<i>Bronze</i>	Less than \$500



- Order Confirmation Flow (sends confirmation email).



- Stock Alert Flow (notifies when stock < 5).



Apex Triggers:

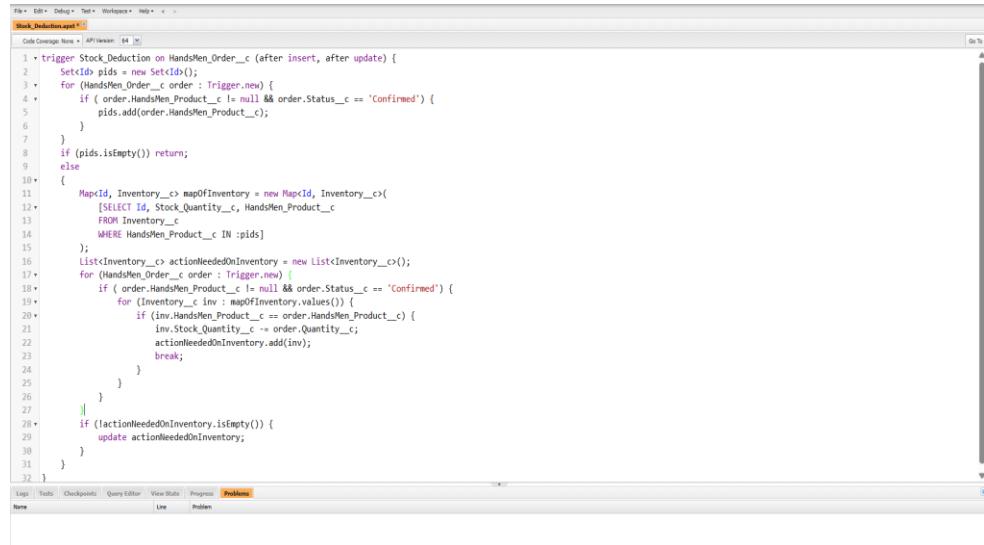
- Update_Order_Total: Calculates Total_Amount__c.

```

trigger Update_Order_Total on HandsMen_Order__c (before insert, before update) {
    Set<Id> pids = new Set<Id>();
    for (HandsMen_Order__c order : Trigger.new) {
        if (order.HandsMen_Product__c != null) {
            pids.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 :pids]
    );
    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: Updates inventory after order.

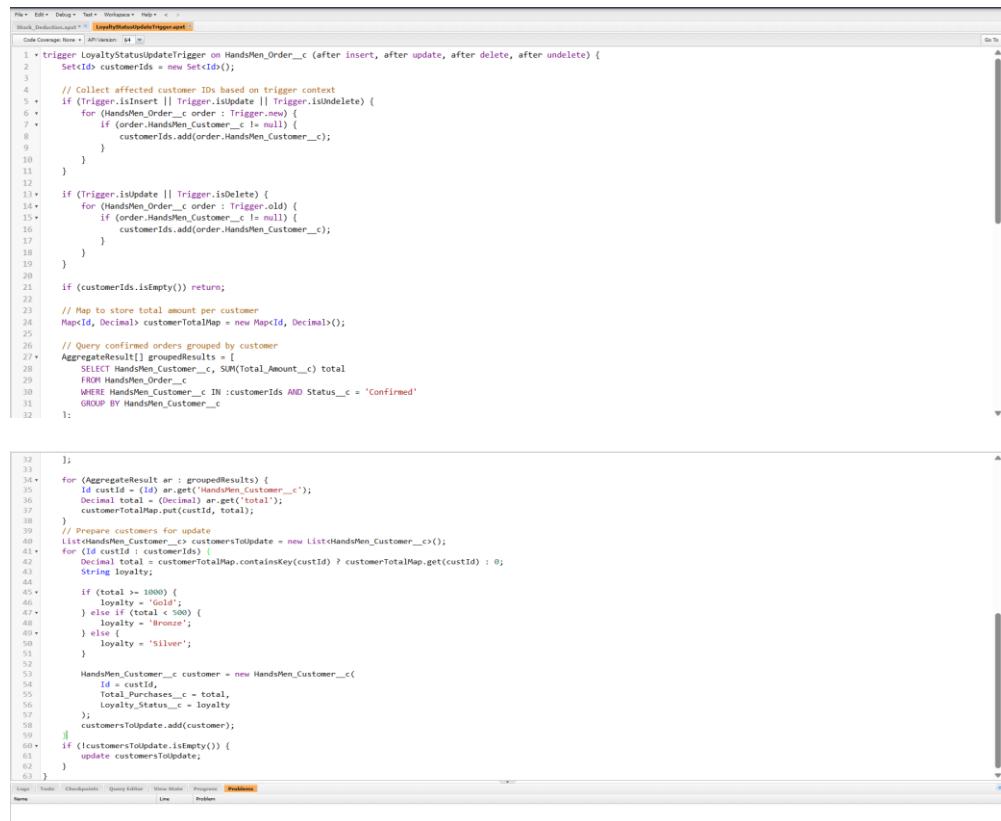


```

1 trigger Stock_Deduction on HandsMen_Order__c (after insert, after update) {
2     Set<Id> pids = new Set<Id>();
3     For (HandsMen_Order__c order : Trigger.new) {
4         If (order.HandsMen_Product__c != null && order.Status__c == 'Confirmed') {
5             pids.add(order.HandsMen_Product__c);
6         }
7     }
8     If (pids.isEmpty()) return;
9     Else {
10        Map<Id, Inventory__c> mapOfInventory = new Map<Id, Inventory__c>{
11            [SELECT Id, Stock.Quantity__c, HandsMen_Product__c
12             FROM Inventory__c
13             WHERE HandsMen_Product__c IN :pids]
14        };
15        List<Inventory__c> actionNeededOnInventory = new List<Inventory__c>();
16        For (HandsMen_Order__c order : Trigger.new) {
17            If (order.HandsMen_Product__c != null && order.Status__c == 'Confirmed') {
18                For (Inventory__c inv : mapOfInventory.values()) {
19                    If (inv.HandsMen_Product__c == order.HandsMen_Product__c) {
20                        If (inv.Stock.Quantity__c <= order.Quantity__c) {
21                            inv.Stock.Quantity__c -= order.Quantity__c;
22                            actionNeededOnInventory.add(inv);
23                        }
24                    }
25                }
26            }
27        }
28        If (!actionNeededOnInventory.isEmpty()) {
29            Update actionNeededOnInventory;
30        }
31    }
32 }

```

- LoyaltyStatusUpdateTrigger: Aggregates order totals and updates Loyalty_Status__c.



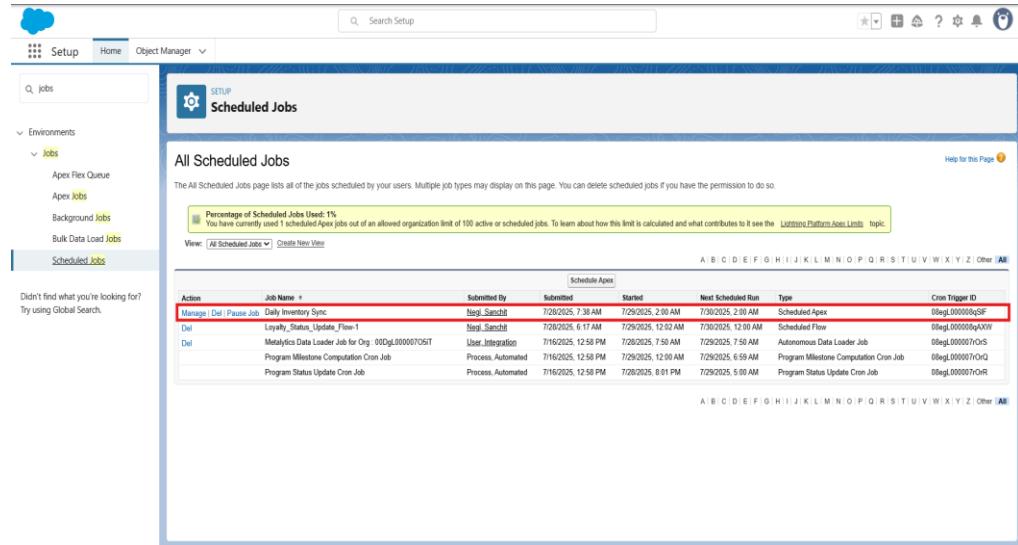
```

1 trigger LoyaltyStatusUpdateTrigger on HandsMen_Order__c (after insert, after update, after delete, after undelete) {
2     Set<Id> customerIds = new Set<Id>();
3
4     // Collect affected customer IDs based on trigger context
5     If (Trigger.isInsert || Trigger.isUpdate || Trigger.isDelete) {
6         For (HandsMen_Order__c order : Trigger.old) {
7             If (order.HandsMen_Customer__c != null) {
8                 customerIds.add(order.HandsMen_Customer__c);
9             }
10        }
11    }
12
13    If (Trigger.isUpdate || Trigger.isDelete) {
14        For (HandsMen_Order__c order : Trigger.old) {
15            If (order.HandsMen_Customer__c != null) {
16                customerIds.add(order.HandsMen_Customer__c);
17            }
18        }
19    }
20
21    If (customerIds.isEmpty()) return;
22
23    // Map to store total amount per customer
24    Map<Id, Decimal> customerTotalMap = new Map<Id, Decimal>();
25
26    // Query confirmed orders grouped by customer
27    AggregateResult[] groupedResults = [
28        SELECT HandsMen_Customer__c, SUM(Total_Amount__c) total
29        FROM HandsMen_Order__c
30        WHERE HandsMen_Customer__c IN :customerIds AND Status__c = 'Confirmed'
31        GROUP BY HandsMen_Customer__c
32    ];
33
34    For (AggregateResult ar : groupedResults) {
35        Id custId = (Id) ar.get('HandsMen_Customer__c');
36        Decimal total = (Decimal) ar.get('total');
37        customerTotalMap.put(custId, total);
38    }
39
40    // Process customer for update
41    List<HandsMen_Customer__c> customersToUpdate = new List<HandsMen_Customer__c>();
42    For (Id custId : customerIds) {
43        Decimal total = customerTotalMap.containsKey(custId) ? customerTotalMap.get(custId) : 0;
44        String loyalty;
45
46        If (total > 1000) {
47            loyalty = 'Gold';
48        } Else If (total < 500) {
49            loyalty = 'Bronze';
50        } Else {
51            loyalty = 'Silver';
52        }
53        HandsMen_Customer__c customer = new HandsMen_Customer__c(
54            Id = custId,
55            Total_Purchases__c = total,
56            Loyalty_Status__c = loyalty
57        );
58        customersToUpdate.add(customer);
59    }
60    If (!customersToUpdate.isEmpty()) {
61        Update customersToUpdate;
62    }
63 }

```

Batch Apex:

- **InventorySyncBatch:** Nightly sync with external inventory systems.



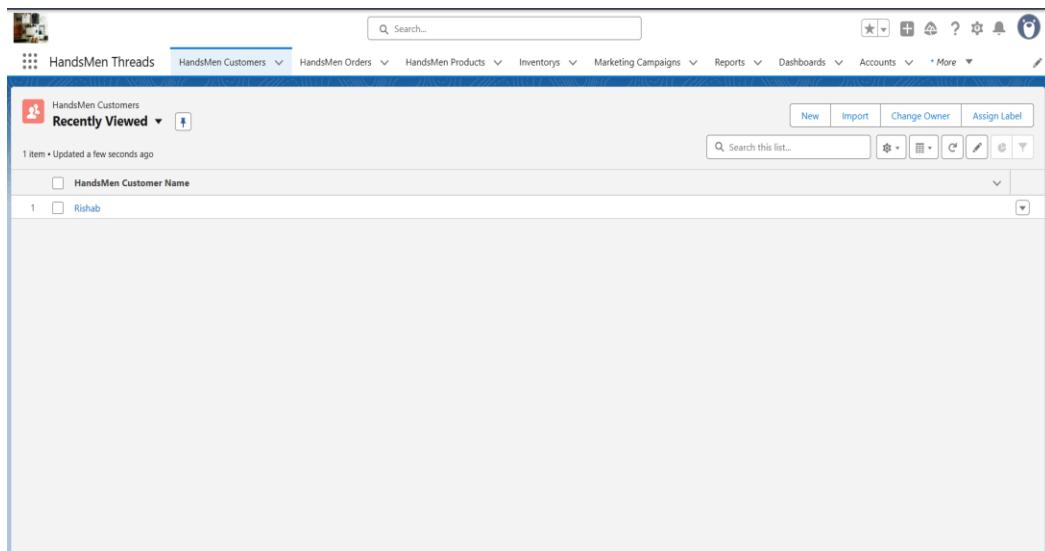
The screenshot shows the Salesforce Setup interface with the 'Scheduled Jobs' page selected. The left sidebar shows 'Jobs' and 'Scheduled Jobs' under 'Apex'. The main content area displays a table of scheduled jobs. The first job in the list, 'Manage | Del | Pause Job: Daily Inventory Sync', is highlighted with a red border. The table includes columns for Action, Job Name, Submitted By, Submitted, Started, Next Scheduled Run, Type, and Cron Trigger ID. The 'Type' column shows 'Scheduled Apex' for the highlighted job and 'Scheduled Flow' for the others.

Action	Job Name	Submitted By	Submitted	Started	Next Scheduled Run	Type	Cron Trigger ID
Manage Del Pause Job	Daily Inventory Sync	bsg_Sanchit	7/28/2025, 7:38 AM	7/29/2025, 2:00 AM	7/30/2025, 2:00 AM	Scheduled Apex	00eg.00000000SF
Del	Loyalty_Status_Update_Flow-1	bsg_Sanchit	7/28/2025, 6:17 AM	7/29/2025, 12:02 AM	7/30/2025, 12:01 AM	Scheduled Flow	00eg.00000000gXXW
Del	Metalytics Data Loader Job for Org 00DgL00000705iT	User_Integration	7/16/2025, 12:58 PM	7/28/2025, 7:58 AM	7/29/2025, 7:59 AM	Autonomous Data Loader Job	00eg.000007YoHS
	Program Milestone Computation Cron Job	Process_Automated	7/16/2025, 12:58 PM	7/29/2025, 12:00 AM	7/29/2025, 6:59 AM	Program Milestone Computation Cron Job	00eg.000007YoRQ
	Program Status Update Cron Job	Process_Automated	7/16/2025, 12:58 PM	7/28/2025, 8:01 PM	7/29/2025, 5:00 AM	Program Status Update Cron Job	00eg.000007YoR

5. Phase 3: UI/UX Development & Customization

Lightning App Configuration:

- **App Name:** HandsMen Threads
- Includes tabs: Product, Order, Customer



The screenshot shows the HandsMen Threads Lightning App interface. The top navigation bar includes tabs for 'HandsMen Customers', 'HandsMen Orders', 'HandsMen Products', 'Inventories', 'Marketing Campaigns', 'Reports', 'Dashboards', 'Accounts', and 'More'. The 'HandsMen Customers' tab is active. The main content area shows a 'Recently Viewed' section with a single item: 'Rishab'. A search bar at the top right is set to 'Search this list...'. The bottom right corner of the screen shows the standard Salesforce navigation icons.

User Management:

- Custom profiles and role hierarchy.

The screenshots illustrate the creation and assignment of three custom roles in the Salesforce Setup interface:

- Sales Role:**
 - Role Detail:** Label: Sales, This role reports to: CEO, Modified By: Sanjith Negi, 7/26/2025, 6:16 AM. Opportunity Access: Users in this role can edit all opportunities associated with accounts that they own, regardless of who owns the opportunities. Case Access: Users in this role can edit all cases associated with accounts that they own, regardless of who owns the cases.
 - Users in Sales Role:** Action: Full Name, Alias: nmka, Username: negsanjith001@gmail.com, Active: checked.
- Inventory Role:**
 - Role Detail:** Label: Inventory, This role reports to: CEO, Modified By: Sanjith Negi, 7/26/2025, 6:17 AM. Opportunity Access: Users in this role can edit all opportunities associated with accounts that they own, regardless of who owns the opportunities. Case Access: Users in this role can edit all cases associated with accounts that they own, regardless of who owns the cases.
 - Users in Inventory Role:** Action: Full Name, Alias: smika, Username: negsanjith002@gmail.com, Active: unchecked.
- Marketing Role:**
 - Role Detail:** Label: Marketing, This role reports to: CEO, Modified By: Sanjith Negi, 7/26/2025, 6:17 AM. Opportunity Access: Users in this role can edit all opportunities associated with accounts that they own, regardless of who owns the opportunities. Case Access: Users in this role can edit all cases associated with accounts that they own, regardless of who owns the cases.
 - Users in Marketing Role:** Action: Full Name, Alias: selex, Username: negsanjith003@gmail.com, Active: checked.

6. Phase 4: Data Migration, Testing & Security

❖ Data Loading:

- Sample records created for Customer , Orders, Products and Inventories.

HandsMen Threads

HandsMen Customers

Recently Viewed

5 items • Updated a few seconds ago

HandsMen Customer Name

- 1 Suresh
- 2 Dinesh
- 3 Rishab
- 4 Rahul
- 5 Arman

New Import Change Owner Assign Label

Search this list...

HandsMen Threads

HandsMen Customers

HandsMen Orders

Recently Viewed

3 items • Updated a few seconds ago

HandsMen OrderNumber

- 1 O-0008
- 2 O-0007
- 3 O-0006

New Import Change Owner Assign Label

Search this list...

HandsMen Threads

HandsMen Customers

HandsMen Orders

HandsMen Products

Recently Viewed

3 items • Updated a few seconds ago

HandsMen Product Name

- 1 T-shirt
- 2 Jeans
- 3 Shirts

New Import Change Owner Assign Label

Search this list...

HandsMen Threads

HandsMen Customers

HandsMen Orders

HandsMen Products

Inventory

Marketing Campaigns

Reports

Dashboards

Accounts

Contacts

Recently Viewed

3 items • Updated a few seconds ago

Inventory Number

- 1 I-0004
- 2 I-0003
- 3 I-0002

New Import Assign Label

Search this list...

❖ Security Controls:

- Profiles for Sales, Marketing, Admin roles.
- Role Hierarchy ensuring proper data visibility.

❖ Testing:

Testing Evidence:

- Customer Registration (auto-update loyalty on order)

- Order Placement (total amount calculation)

HandsMen Order O-0001

Related Details

HandsMen OrderNumber: O-0001

Owner: Sanchit Negi

HandsMen Product: T-Shirt

HandsMen Customer: Rishab

Status: Confirmed

Quantity: 550

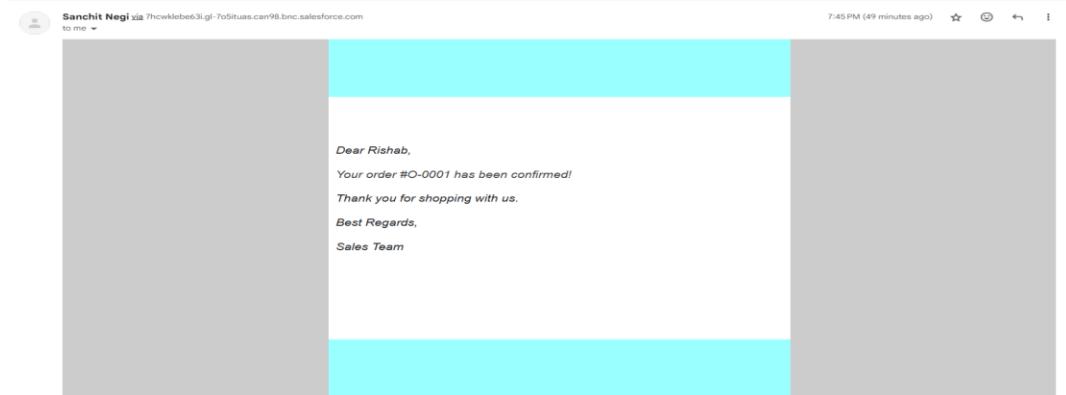
Total Amount: 137,500

Customer Email: sanchitnegi177@gmail.com

Created By: Sanchit Negi, 7/28/2025, 5:33 AM

Last Modified By: Sanchit Negi, 7/28/2025, 7:10 AM

- Order Confirmation (email trigger)



- Inventory Update

Inventory I-0004

Related Details

Inventory Number: I-0004

Stock Status: Available

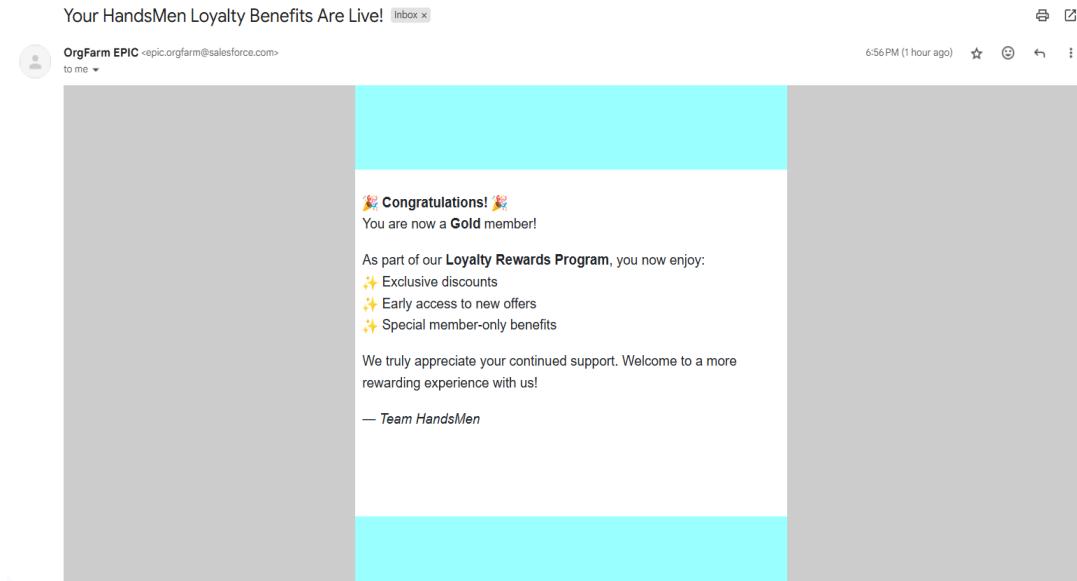
Warehouse: Kotdwara zone 1

Stock Quantity: 100

Created By: Sanchit Negi, 7/30/2025, 9:19 AM

Last Modified By: Sanchit Negi, 7/30/2025, 9:06 AM

- Loyalty Tier Re-evaluation



◆ **Field History Tracking:**

- Enabled on Stock_Quantity__c, Loyalty_Status__c.

7. Phase 5: Deployment, Documentation & Maintenance

Deployment Strategy:

- Change Sets used for migrating components to production.

Maintenance:

- Scheduled batch job monitoring.
- Exception handling in Apex triggers.
- Ongoing campaign creation and product updates via UI.

Troubleshooting:

- Apex debug logs enabled.
- Error emails for flow failures and low stock notifications.

8. Conclusion

The **HandsMen Threads CRM** provides a robust and scalable Salesforce solution for managing a high-end men's fashion business. The system ensures end-to-end automation for order processing, loyalty tracking, and stock management — driving productivity and customer satisfaction.

It empowers the brand to make data-driven decisions, retain loyal customers through tiered programs, and eliminate operational silos. With this Salesforce implementation, HandsMen Threads is well-equipped to scale and deliver a premium customer experience in the fashion retail industry.

9. Future Enhancements

- **AI-powered product recommendation engine** to suggest personalized outfits based on user preferences and purchase history.
- **Integration with WhatsApp** for sending real-time order confirmations and updates to customers.
- **Loyalty redemption point tracking** to allow customers to redeem accumulated points for future purchases.