

# **Salesforce CRM Project Documentation**

## **Project Title:**

**Handsmen Threads: Elevating the Art of Sophistication in Men's Fashion**

**Author:** John Vincent Del Rosario

**Date:** November 27, 2025

## **Project Overview**

**Handsmen Threads is a Salesforce Customer Relationship Management/CRM solution that designed to streamline a customer engagement and sales processes for a premium men's fashion brand. The CRM focuses on a personalized customer management, order tracking and loyalty programs, ensuring that the brand provide a sophistication both in its products and customer experience.**

## **Objectives**

- **Enhancing customer relationship management in terms of tracking preferences and purchased history.**
- **Streamline booking and order processes for bespoke tailoring services.**
- **Provide an actionable insights through dashboards and reports.**
- **Support scalability for future improvements like AI-driven suggestions and chatbot integration.**

## **Phase 1: Requirement Analysis & Planning**

- **Business Requirements** – a customers need a personalized service, efficient order tracking and loyalty rewards.

- **Project Scope** – build a CRM with modules specifically for customer profiles, booking, payments and loyalty programs.
- **Data & Security Model** – custom objects for customers, orders, loyalty points (role-based access for sales staff and admins).
- **Stakeholders** – sales team, tailoring staff, management and IT support.
- **Execution Roadmap** – Requirement gathering, Development, Testing, Deployment lastly is Maintenance.

## **Phase 2: Salesforce Development – Backend & Configurations**

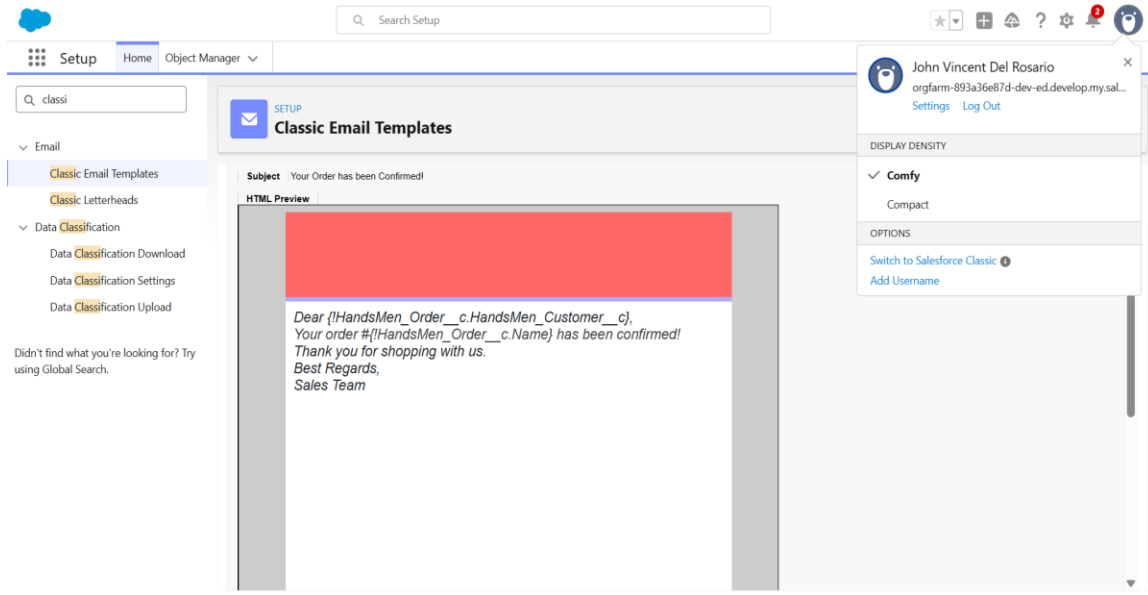
**Environment Setup: Sandbox and DevOps workflow established.**

**Customizations:**

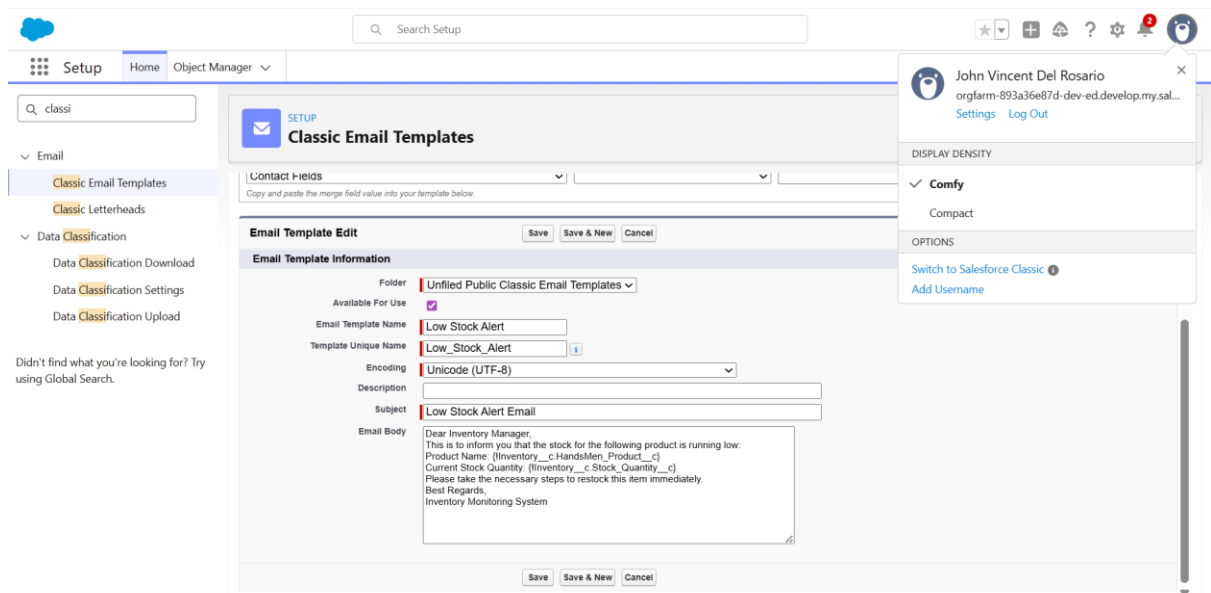
- **Custom objects:**  
*Customer, Booking, Payment, Loyalty Points.*
- **Validation rules:**  
Ensure accurate customer measurements and payment details.
- **Automation:**  
Flows for loyalty point updates, approval processes for bespoke orders.
- **Apex Development:**  
Triggers for automatic loyalty point calculation.
- **Screenshots:**  
(Insert relevant Salesforce setup screenshots).

Here's the Relevant Screenshots:

## 1. Order Confirmation Email



## 2. Low Stock Alert Email



### 3. Loyalty Points Email

The screenshot shows the Salesforce Classic Email Template Editor interface. The left sidebar contains navigation links for Setup, Home, and Object Manager. The main content area is titled 'Classic Email Templates' and shows the 'Loyalty Program Email' template. The template is in 'HTML Email Template Edit' mode. The 'Text-Only Email Content' section is visible, showing the subject 'Loyalty Program Email' and the body text: '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 interface includes a search bar at the top, a user profile dropdown for John Vincent Del Rosario, and a 'Switch to Salesforce Classic' button.

### 4. Order Total Trigger

The screenshot shows the Salesforce Apex Trigger code for the 'OrderTotalTrigger' class. The code is written in Apex and is triggered on the 'HandsMen\_Order\_\_c' object (before insert, before update). The trigger logic is as follows:

```
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 }
```

The code is displayed in the 'OrderTotalTrigger.apex' file in the Salesforce IDE. The interface includes a menu bar (File, Edit, Debug, Test, Workspace, Help) and a toolbar with buttons for Code Coverage, API Version, and Go To. The bottom of the screen shows a 'Logs' tab with a table of log entries.

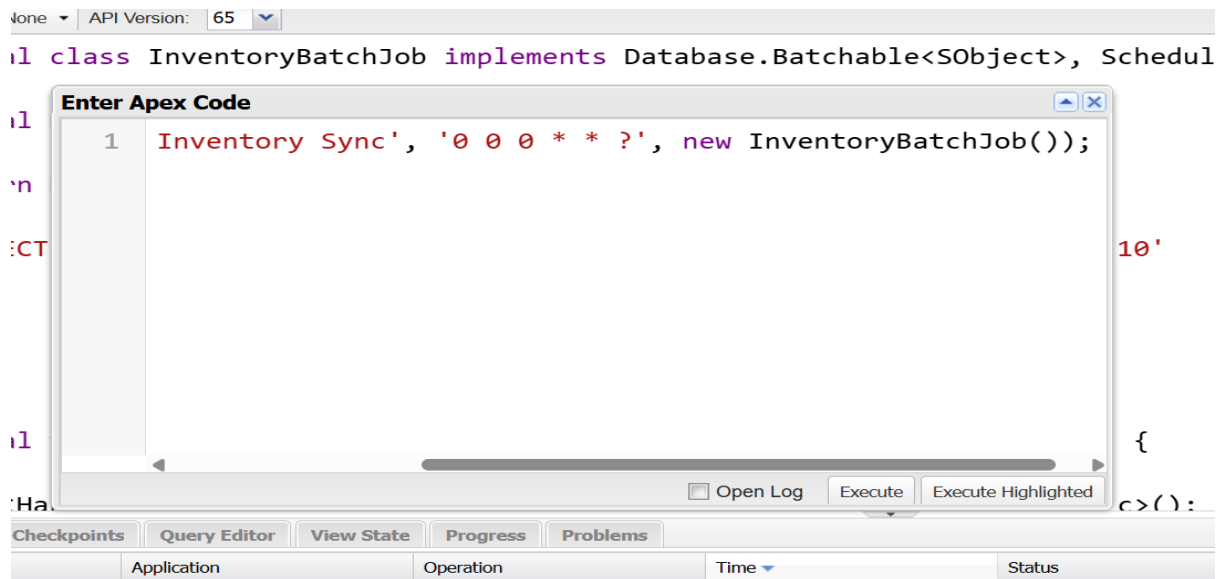
## 5. Stock Deduction Trigger

```
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    List<Inventory__c> inventoriesToIupdate = new List<Inventory__c>();
19
20    for (HandsMen_Order__c order : Trigger.new) {
21        if (order.Status__c == 'Confirmed' && order.HandsMen_Product__c != null) {
22            for (Inventory__c inv : inventoryMap.values()) {
23                if (inv.HandsMen_Product__c == order.HandsMen_Product__c) {
24                    inv.Stock_Quantity__c -= order.Quantity__c;
25                    inventoriesToIupdate.add(inv);
26                    break;
27                }
28            }
29        }
30    }
31
32    if (!inventoriesToIupdate.isEmpty()) {
33        update inventoriesToIupdate;
34    }
35 }
36 }
```

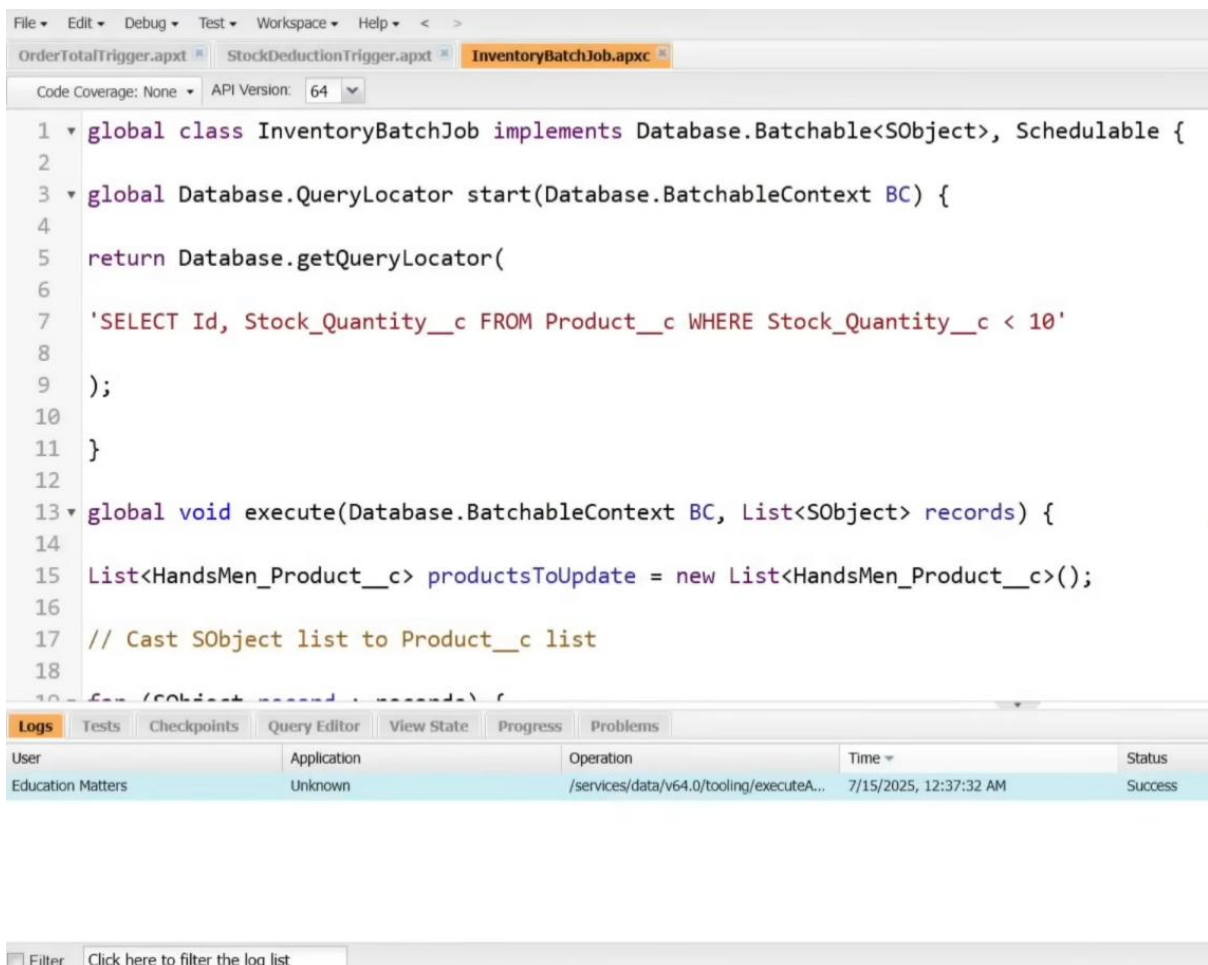
## 6. Inventory Batch Job

```
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> productsToIupdate = new List<HandsMen_Product__c>();
15
16        // Cast SObject list to Product__c list
17        for (SObject record : records) {
18
19            HandsMen_Product__c product = (HandsMen_Product__c) record;
20
21            product.Stock_Quantity__c += 50; // Restock logic
22
23            productsToIupdate.add(product);
24
25        }
26
27        if (!productsToIupdate.isEmpty()) {
28
29            try {
30
31                update productsToIupdate;
32
33            } catch (DmlException e) {
34
35                System.debug('Error updating Inventory: ' + e.getMessage());
36
37            }
38
39        }
40    }
41 }
```

## 7. Inventory Executing



## 8. Inventory Executing Success



## **Phase 3: UI/UX Development & Customization**

- **Lightning App:**  
Handsmen Threads CRM app created via App Manager.
- **Page Layouts & Dynamic Forms:**  
Tailored layouts for customer profiles and bookings.
- **Reports & Dashboards:**  
Sales performance, customer loyalty trends, booking pipeline.
- **LWC Development:**  
Optional components for real-time order status.

## **Phase 4: Data Migration, Testing & Security**

- **Data Migration:**  
Imported customer records via Data Loader.
- **Security:**  
Profiles for Admin, Sales, Tailor; role hierarchy for managers.
- **Testing:**  
Test cases for booking creation, loyalty point updates, approval workflows.

# Date Model and Key Fields

## Handsman Customer

- **FullName**
- **Email**
- **Phone**
- **Loyalty\_Status\_c**
- **Total Purchases\_c**

## Handsman Product

- **SKU\_c**
- **Name**
- **Price\_c**
- **Stock\_Quantity\_c**

## Inventory

- **HandsMen\_Product\_c(Master Detail)**
- **CreatedById**
- **Warehouse**
- **Stock\_Quantity\_c**
- **Stock\_Status\_c**

## Handsman Orders

- **Quantity\_c**
- **Customer\_Email\_c**
- **HandsMen\_Customer\_c (lookup)**
- **Status\_c (Pending Confirmed Cancelled)**
- **HandsMen\_Product\_c (lookup)**
- **Name (Order Number)**