

# HandsMen Threads: Elevating the Art of Sophistication in Men's Fashion

## Salesforce CRM Project Documentation

---

### Project Overview

HandsMen Threads is a dynamic organization in the fashion industry embarking on a Salesforce CRM 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. This premium fashion platform streamlines men's bespoke tailoring and enhances customer experience through personalized styling and seamless order management.

Key Features:

- Customer relationship management for men's fashion retail
  - Order tracking and confirmation automation
  - Dynamic loyalty program management
  - Inventory management with proactive stock alerts
  - Marketing campaign management
- 

### Objectives

The main goals of building this CRM are:

1. Enhanced Data Management - Build a robust data model to store all pertinent business data ensuring seamless information flow
  2. Data Integrity Maintenance - Safeguard accuracy and consistency of data directly from the user interface
  3. Customer Engagement - Automate order confirmations via email to foster customer relations
  4. Loyalty Program Automation - Dynamically update customer loyalty statuses based on purchase history
  5. Inventory Control - Implement proactive stock alerts when inventory drops below threshold levels
  6. Operational Efficiency - Process bulk orders daily, update financial records and adjust inventory automatically
-

**Phase 1: Requirement Analysis & Planning**

**Understanding Business Requirements**

The user needs and problems being solved include:

- Centralized customer data management
- Automated email notifications for order confirmations
- Real-time inventory tracking and alerts
- Customer loyalty tracking and rewards
- Streamlined order processing workflow

**Project Scope**

- Industry: Manufacturing/Fashion
- Project Type: Individual
- Complexity: Medium
- Duration: 30+ hours estimated
- Total Epics: 16
- Total Stories: 35

**Stakeholders Mapping**

- Sales Team (Primary Users)
- Warehouse Team (Inventory Notifications)
- Marketing Team (Campaign Management)
- Customers (Email Recipients)

**Design Data Model**

Custom Objects:

Object Name	API Name	Description
HandsMen Customer	HandsMen_Customer __c	Stores customer details

HandsMen Product	HandsMen_Product__c	Stores product catalog
HandsMen Order	HandsMen_Order__c	Stores customer orders
Inventory	Inventory__c	Tracks inventory levels
Marketing Campaign	Marketing_Campaign__c	Manages promotions & campaigns

### Security Model

- Role Hierarchy Implementation
  - Profile-based access (Sales Profile)
  - Permission Sets for extended access
  - Field-Level Security
- 

## Phase 2: Salesforce Development - Backend Configurations

### Custom Objects Configuration

#### 1. HandsMen Customer Object (HandsMen\_Customer\_\_c)

- Label: HandsMen Customer
- Plural Label: HandsMen Customer
- Record Name: HandsMen Customer Name (Data Type: Text)
- Allow Reports: Enabled
- Allow Search: Enabled

Fields:

Field Label	API Name	Data Type	Description
Name	Name	Text (Record Name)	Customer name
Email	Email__c	Email	Customer email address
Phone	Phone__c	Phone	Customer phone number
Loyalty Status	Loyalty_Status__c	Picklist (Bronze, Gold, Silver)	Customer loyalty tier
Total Purchases	Total_Purchases__c	Number	Total purchase count

## 2. HandsMen Product Object (HandsMen\_Product\_\_c)

- Label: HandsMen Product
- Plural Label: HandsMen Products
- Record Name: HandsMen Product Name (Data Type: Text)

Fields:

Field Label	API Name	Data Type	Description
Name	Name	Text (Record Name)	Product name
SKU	SKU__c	Text	Stock Keeping Unit
Price	Price__c	Currency	Product price
Stock Quantity	Stock_Quantity__c	Number	Available inventory

### 3. HandsMen Order Object (HandsMen\_Order\_\_c)

- Label: HandsMen Order
- Plural Label: HandsMen Orders
- Record Name: Order Number (Data Type: Text)

Fields:

Field Label	API Name	Data Type	Description
-------------	----------	-----------	-------------

Order Number	Name	Text (Record Name)	Unique order identifier
Status	Status__c	Picklist (Pending, Confirmed, Rejection)	Order status
Quantity	Quantity__c	Number	Order quantity
Total Amount	Total_Amount__c	Number	Order total

#### 4. Inventory Object (Inventory\_\_c)

- Record Name: Auto Number

Fields:

Field Label	API Name	Data Type	Description
Auto Number	Name	Auto Number (Record Name)	Auto-generated ID
Warehouse	Warehouse__c	Text	Warehouse location

Stock Quantity	Stock_Quantity__c	Number	Stock level
-------------------	-------------------	--------	-------------

## 5. Marketing Campaign Object (Marketing\_Campaign\_\_c)

Fields:

Field Label	API Name	Data Type	Description
Campaign Name	Name	Text (Record Name)	Campaign identifier
Start Date	Start_Date__c	Date	Campaign start date
End Date	End_Date__c	Date	Campaign end date

## Lookup Relationships

- HandsMen Order → HandsMen Customer (Customer lookup)
- HandsMen Order → HandsMen Product (Product lookup)
- Inventory → HandsMen Product (Product lookup)

## Validation Rules

Validation rules are implemented through the Apex Trigger Handler to validate order quantities based on status:

- Confirmed Status: Quantity must be greater than 500
- Pending Status: Quantity must be greater than 200
- Rejection Status: Quantity must be 0

## Automation - Apex Classes

### OrderTriggerHandler.cls

text

```
public class OrderTriggerHandler {
    public static void validateOrderQuantity(List<HandsMen_Order__c>
orderList) {
        for (HandsMen_Order__c order : orderList) {
            if (order.Status__c == 'Confirmed') {
                if (order.Quantity__c == null || order.Quantity__c <= 500) {
                    order.Quantity__c.addError('For Status "Confirmed", Quantity must
be more than 500.');
```

## Asynchronous Apex - Batch Jobs

### InventoryBatchJob.cls



text

global class InventoryBatchJob implements Database.Batchable<SObject>, Schedulable {

```
    global Database.QueryLocator start(Database.BatchableContext BC) {
        return Database.getQueryLocator(
            'SELECT Id, Stock_Quantity__c FROM Product__c WHERE
Stock_Quantity__c < 10'
        );
    }
```

```
    global void execute(Database.BatchableContext BC, List<SObject> records)
    {
        List<HandsMen_Product__c> productsToUpdate = new
List<HandsMen_Product__c>();

        for (SObject record : records) {
            HandsMen_Product__c product = (HandsMen_Product__c) record;
            product.Stock_Quantity__c += 50; // Restock logic
            productsToUpdate.add(product);
        }

        if (!productsToUpdate.isEmpty()) {
            update productsToUpdate;
        }
    }
```

```
    global void finish(Database.BatchableContext BC) {
        // Post-processing logic
    }
```

```
    global void execute(SchedulableContext SC) {
        Database.executeBatch(new InventoryBatchJob());
    }
}
```

---

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

### **Lightning App Setup**

- App Name: HandsMen Threads
- Navigation Style: Standard Navigation
- Included Objects: HandsMen Customer, HandsMen Product, HandsMen Order, Inventory, Marketing Campaign

### **Custom Tabs Created**

1. HandsMen Customer Tab
2. HandsMen Product Tab
3. HandsMen Order Tab
4. Inventory Tab
5. Marketing Campaign Tab

### **Page Layouts**

Custom page layouts configured for each object with relevant fields organized in logical sections.

### **Email Templates**

- Order Confirmation Email Template - Sent when order status is updated to "Confirmed"
- Stock Alert Email Template - Sent to warehouse team when stock drops below 5 units

---

## **Phase 4: Automation - Flows**

### **Flow 1: Order Confirmation Email Flow**

- Type: Record-Triggered Flow
- Object: HandsMen\_Order\_\_c
- Trigger: When a record is updated
- Condition: Status\_\_c = "Confirmed"

- Action: Send Email Alert to customer using Order Confirmation Email Template

### **Flow 2: Stock Alert Email Flow**

- Type: Record-Triggered Flow
- Object: Inventory\_\_c
- Trigger: When a record is updated
- Condition: Stock\_Quantity\_\_c < 5
- Action: Send Email Alert to warehouse team

### **Flow 3: Scheduled Flow for Bulk Order Updates**

- Type: Scheduled Flow
  - Schedule: Daily at midnight
  - Action: Process bulk orders, update financial records, adjust inventory levels
- 

## **Phase 5: Data Security Configuration**

### **Profiles**

- Sales Profile - Full access to HandsMen Customer, HandsMen Order, HandsMen Product objects
- Custom profile configurations for appropriate object and field-level permissions

### **Roles & Role Hierarchy**

- CEO
  - Sales Manager
    - Sales Representative
  - Warehouse Manager
    - Warehouse Staff

### **Users**

- Users created and assigned to appropriate profiles and roles based on job function

## Permission Sets

- Additional permissions granted through permission sets for specific functionality access beyond base profile
- 

## Phase 6: Testing

### Test Cases

Test Case ID	Feature	Test Description	Expected Result	Actual Result
TC001	Customer Creation	Create new HandsMen Customer record	Record created successfully with all required fields	Pass
TC002	Order Validation - Confirmed	Create order with Status=Confirmed and Quantity=300	Validation error displayed	Pass
TC003	Order Validation - Confirmed	Create order with Status=Confirmed and Quantity=600	Order saved successfully	Pass

TC00 4	Order Validation - Pending	Create order with Status=Pending and Quantity=100	Validation error displayed	Pass
TC00 5	Order Confirmation Flow	Update order status to Confirmed	Email sent to customer	Pass
TC00 6	Stock Alert Flow	Update inventory below 5 units	Email sent to warehouse team	Pass
TC00 7	Batch Job Execution	Execute InventoryBatchJob	Products with stock <10 restocked by 50	Pass

### Test Classes

Test classes created with minimum 75% code coverage for:

- OrderTriggerHandler test class
  - InventoryBatchJob test class
- 

## Phase 7: Deployment & Maintenance

### Deployment Strategy

- Use Change Sets for deployment from sandbox to production

- Include all custom objects, fields, flows, Apex classes, profiles, and permission sets

### **Maintenance Plan**

- Regular monitoring of batch job execution
- Review of email delivery logs
- Periodic review of user access and permissions
- Regular backup of data

### **Troubleshooting Approach**

- Use Debug Logs to trace issues
  - Monitor Flow Interviews for automation failures
  - Check Apex Jobs for batch processing status
- 

### **Conclusion**

The HandsMen Threads Salesforce CRM implementation successfully delivers a comprehensive solution for managing customer relationships, orders, products, and inventory in the men's fashion industry. The system automates key business processes including order confirmations, loyalty status updates, and stock level monitoring, resulting in improved customer engagement and operational efficiency.

Key Achievements:

- Robust data model with 5 custom objects and appropriate relationships
- Automated email notifications for customer engagement
- Proactive inventory management through batch processing
- Secure access control through profiles, roles, and permission sets
- Validated data integrity through Apex triggers