

HandsMen Threads: Salesforce CRM Implementation Project Documentation

Project Overview

HandsMen Threads is a premium men's fashion brand aiming to streamline bespoke tailoring operations and enhance customer experience via a robust Salesforce CRM. This project focuses on building a dynamic system for managing customer relationships, automating order management, tracking inventory, and personalizing marketing efforts.

Key Features: - Custom objects for Orders, Inventory, Customers, Products, and Campaigns - Automated order confirmation and loyalty update flows - Real-time stock alerts and batch inventory updates - Streamlined user access with role-based controls

Objectives

The primary goals of this Salesforce CRM implementation include: - Centralized data management across departments - Improved customer engagement through email automation - Real-time stock and order tracking for better operational flow - Personalized loyalty programs based on customer activity - Secure and role-based access for platform users

Phase 1: Requirement Analysis & Planning

Understanding Business Requirements

- Seamless order processing for tailored clothing
- Real-time inventory updates to avoid stockouts
- Loyalty-based marketing campaigns
- Centralized customer information management

Defining Project Scope and Objectives

- Build custom data model
- Automate notifications and data updates
- Enable admin-level security and audit tracking

Design Data Model and Security Model

- 5 Custom Objects: Customer, Order, Product, Inventory, Campaign
 - Lookup and Master-Detail relationships for structured data
 - Role hierarchy: CEO → Sales, Inventory, Marketing
 - Profile: Platform 1 with CRUD access
-

Phase 2: Salesforce Development - Backend & Configurations

Setup Environment & DevOps Workflow

- Developer org created at developer.salesforce.com
- Configured permissions, tabs, and navigation in Lightning App Builder

Customization

- Fields: Loyalty Status (Picklist), Full Name (Formula), Email, Phone
- Validation Rules: Minimum order amount, valid email format, non-negative inventory
- Automations: Record-Triggered and Scheduled Flows

Apex Development

- **OrderTriggerHandler:** Validates order quantity based on status
 - **OrderTrigger:** Invokes handler before insert/update
 - **InventoryBatchJob:** Batch + Schedulable class for daily restocking
-

Phase 3: UI/UX Development & Customization

Lightning App Setup

- App Name: HandsMen Threads
- Added Tabs: Customers, Orders, Products, Inventory, Campaigns, Dashboards

UI Components

- Dynamic Forms on Page Layouts for Customer Object
 - Reports on Order and Inventory metrics
 - Dashboard visualizing stock levels and customer loyalty
 - Profile-specific layout visibility
-

Phase 4: Data Migration, Testing & Security

Data Loading

- Used Data Import Wizard for Customers and Products
- Data Loader for Orders and Inventory in bulk

Security Model

- Profiles: Platform 1 (custom cloned from Standard)
- Permission Sets: Permission_Platform_1 (read/edit/delete access)
- Roles: Sales, Inventory, Marketing
- Sharing Rules: Default private, manual sharing for campaigns

Duplicate & Matching Rules

- Applied email-based duplicate rule on Customers
- Matching rules set for email uniqueness

Test Classes

- Created test class for **OrderTriggerHandler**
- Used System.assert to validate trigger conditions

Testing Cases (Examples)

Feature	Test Case	Expected Result	Actual Result	Screenshot
Order Confirmation Flow	Confirm order → Email sent	Email received		[✓]
Inventory Stock Alert	Stock < 5 → Alert sent	Email sent to warehouse manager		[✓]
Loyalty Flow	Purchases > 1000 → Loyalty = Gold	Gold status updated		[✓]

Phase 5: Deployment, Documentation & Maintenance

Deployment Strategy

- Used Change Sets for metadata migration to Production
- Pre-deployment validations and post-deployment smoke test

Maintenance & Monitoring

- Scheduled job monitoring via Apex Jobs page
- Periodic audits for data integrity and automation logs

Troubleshooting

- Used Debug Logs for trigger/flow failures
 - Standard error messages for validation rule blocks
-

Conclusion

The HandsMen Threads Salesforce CRM implementation offers a complete digital solution for customer and inventory management, marketing, and business automation. The system's scalability and security ensure readiness for future growth while significantly reducing manual workload.

Additional Documentation

Features Described:

- **Validation Rules:** Prevent zero-value orders/inventory; enforce Gmail domain
- **Flows:** Order confirmation, loyalty updates, low stock alerts
- **Apex:** Class-based trigger logic, batch processing for inventory

Testing Strategy

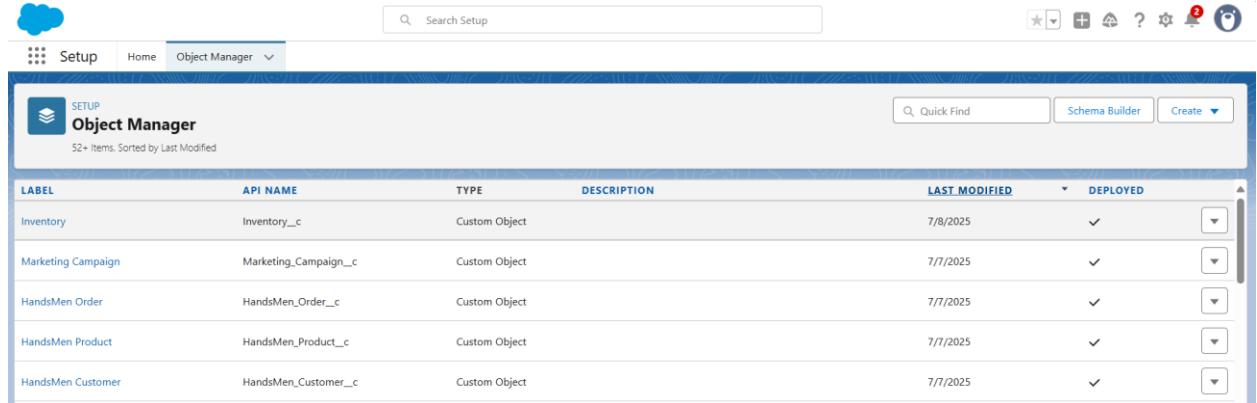
- Manual tests for flows, validation, and batch updates
- Email simulation for customer and internal communications

Future Enhancements

- Integrate AI Chatbot for real-time customer support
 - Predictive analytics for loyalty recommendations
 - WhatsApp/SMS notifications for order confirmation
-

📸 Screenshots

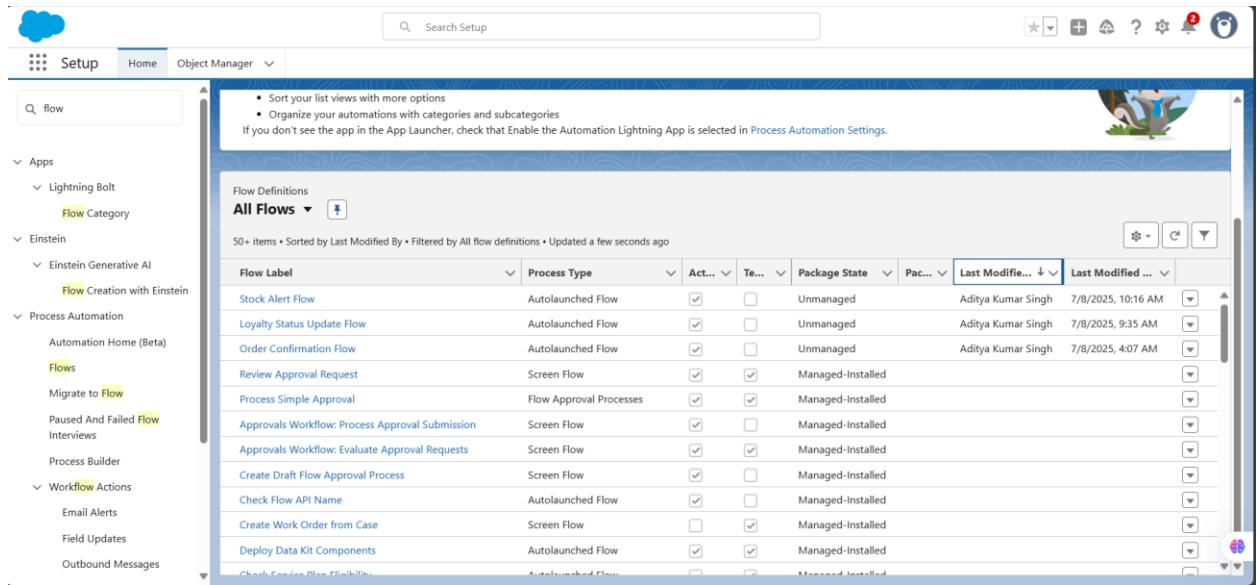
- Object Manager: Custom Objects



The screenshot shows the Salesforce Object Manager interface. At the top, there's a search bar labeled "Search Setup" and a toolbar with various icons. Below the toolbar, the "Object Manager" tab is selected. The main area displays a table of custom objects:

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Inventory	Inventory__c	Custom Object		7/8/2025	✓
Marketing Campaign	Marketing_Campaign__c	Custom Object		7/7/2025	✓
HandsMen Order	HandsMen_Order__c	Custom Object		7/7/2025	✓
HandsMen Product	HandsMen_Product__c	Custom Object		7/7/2025	✓
HandsMen Customer	HandsMen_Customer__c	Custom Object		7/7/2025	✓

- Flow Builder: All flow nodes



The screenshot shows the Salesforce Flow Builder interface. On the left, there's a sidebar with a search bar for "flow" and a tree view of apps and categories. The "Flows" category is expanded, showing sub-categories like "Lightning Bolt", "Einstein Generative AI", "Process Automation", "Workflow Actions", etc. The main area is titled "All Flows" and shows a table of flow definitions:

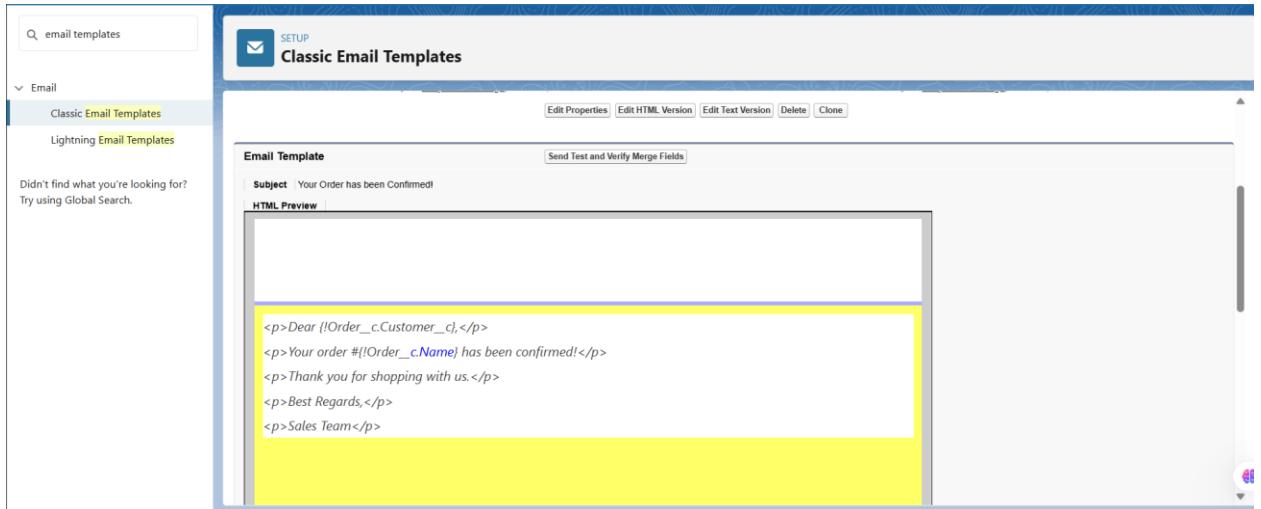
Flow Label	Process Type	Act...	Te...	Package State	Pac...	Last Modified...	Last Modified ...
Stock Alert Flow	Autolaunched Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Unmanaged		Aditya Kumar Singh	7/8/2025, 10:16 AM
Loyalty Status Update Flow	Autolaunched Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Unmanaged		Aditya Kumar Singh	7/8/2025, 9:35 AM
Order Confirmation Flow	Autolaunched Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Unmanaged		Aditya Kumar Singh	7/8/2025, 4:07 AM
Review Approval Request	Screen Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Process Simple Approval	Flow Approval Processes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Approvals Workflow: Process Approval Submission	Screen Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Managed-Installed			
Approvals Workflow: Evaluate Approval Requests	Screen Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Create Draft Flow Approval Process	Screen Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Managed-Installed			
Check Flow API Name	Autolaunched Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Managed-Installed			
Create Work Order from Case	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Deploy Data Kit Components	Autolaunched Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Check Complex Plan Availability	Autolaunched Flow	<input type="checkbox"/>	<input type="checkbox"/>	Managed-Installed			

- Email Templates: HTML view

Action	Email Template Name	Template Type	Available For Use	Description	Author	Last Modified Date
Edit Del	Loyalty_Program_Email	HTML	✓		100	7/8/2025
Edit Del	Low_Stock_Alert	HTML	✓		100	7/8/2025
Edit Del	Order_Confirmation_Email	HTML	✓		100	7/8/2025
Edit Del	SUPPORT_Self-Service_New_Comment_Notification_(SAMPLE)	Text	✓	Sample email template that can be sent to your Self-Service customers to notify them a public comment has been added to their case.	QEPIC	6/30/2025
Edit Del	Support_Self-Service_Reset_Password	Text	✓	Notification of new password when self-service password is reset	QEPIC	6/30/2025
Edit Del	Support_Self-Service_New_Login_and_Password	Text	✓	Notification of login and password to new self-service user	QEPIC	6/30/2025
Edit Del	Support_Case_Assignment_Notification	Text	✓	Notification to rep when case is auto-assigned	QEPIC	6/30/2025
Edit Del	Support_Case_Created_(Phone_Inquiries)	Text	✓	Notification to customer about case created through phone call	QEPIC	6/30/2025
Edit Del	Support_Case_Created_(Web_Inquiries)	Text	✓	Notification to customer about case created online	QEPIC	6/30/2025
Edit Del	Appointment_for_Unauthenticated_User_using_Appointment_Types_-For_Amazon_Chime	Custom	✓	Email template for confirmation of an appointment when appointments are scheduled using appointment types with Amazon Chime	sfdcadmin	6/30/2025
Edit Del	Appointment_for_Unauthenticated_User_using_Engagement_Channels-For_Amazon_Chime	Custom	✓	Email template for confirmation of an appointment when appointments are scheduled using engagement channels with Amazon Chime	sfdcadmin	6/30/2025

Subject	HTML Preview
You're Part of Our Loyalty Program!	<p>Dear {!HandsMen_Customer__c.Full_Name__c},</p><p>We're excited to have you in our {!HandsMen_Customer__c.Loyalty_Status__c} Loyalty Program!</p><p>As a valued customer, you'll receive special offers, early access to new products, and exclusive deals.</p><p>Thank you for your continued support.</p><p>Warm regards, HandsMen Sales Team</p>
Low Stock Alert: (Inventory__c.Name) is Running Low	<p>Dear Inventory Manager,</p><p>This is to notify you that the stock level for {!Inventory__c.Name} is currently at {!Inventory__c.Stock_Quantity__c}.</p><p>Please consider restocking this item as soon as possible.</p><p>Regards, Inventory Management System</p>

Subject	HTML Preview
You're Part of Our Loyalty Program!	<p>Dear {!HandsMen_Customer__c.Full_Name__c},</p><p>We're excited to have you in our {!HandsMen_Customer__c.Loyalty_Status__c} Loyalty Program!</p><p>As a valued customer, you'll receive special offers, early access to new products, and exclusive deals.</p><p>Thank you for your continued support.</p><p>Warm regards, HandsMen Sales Team</p>
Low Stock Alert: (Inventory__c.Name) is Running Low	<p>Dear Inventory Manager,</p><p>This is to notify you that the stock level for {!Inventory__c.Name} is currently at {!Inventory__c.Stock_Quantity__c}.</p><p>Please consider restocking this item as soon as possible.</p><p>Regards, Inventory Management System</p>



- Apex Classes & Triggers in Developer Console

```

1 *global class InventoryBatchJob implements Database.Batchable<SObject>, Schedulable {
2
3     global Database.QueryLocator start(Database.BatchableContext BC) {
4
5         return Database.getQueryLocator(
6             'SELECT Id, Stock_Quantity__c FROM Product__c WHERE Stock_Quantity__c < 10'
7         );
8     }
9
10    global void execute(Database.BatchableContext BC, List<SObject> records) {
11        List<HandsMen_Product__c> productsToUpdate = new List<HandsMen_Product__c>();
12
13        // Cast SObject list to Product__c list
14        for (SObject record : records) {
15            HandsMen_Product__c product = (HandsMen_Product__c) record;
16
17            product.Stock_Quantity__c += 50; // Restock logic
18
19            productsToUpdate.add(product);
20        }
21
22    }
23
24    global void finish(Database.BatchableContext BC) {
25        if (!productsToUpdate.isEmpty()) {
26            try {
27                Database.update(productsToUpdate);
28            } catch (Exception e) {
29                System.debug('Error updating products: ' + e.getMessage());
30            }
31        }
32    }
}

```

File • Edit • Debug • Test • Workspace • Help • < >

InventoryBatch.apex • OrderTriggerHandler.apex • OrderTrigger.apex

Code Coverage: None • API Version: 64

```

1 • public class OrderTriggerHandler {
2
3 •     public static void validateOrderQuantity(List<HandsMen_Order__c> orderList) {
4
5 •         for (HandsMen_Order__c order : orderList) {
6
7 •             if (order.Status__c == 'Confirmed') {
8
9 •                 if (order.Quantity__c == null || order.Quantity__c <= 500) {
10
11                     order.Quantity__c.addError('For Status "Confirmed", Quantity must be more than 500.');
12
13                 }
14
15             } else if (order.Status__c == 'Pending') {
16
17                 if (order.Quantity__c == null || order.Quantity__c <= 200) {
18
19                     order.Quantity__c.addError('For Status "Pending", Quantity must be more than 200.');
20
21                 }
22
23             } else if (order.Status__c == 'Rejection') {
24
25                 if (order.Quantity__c == null || order.Quantity__c != 0) {
26
27                     order.Quantity__c.addError('For Status "Rejection", Quantity must be 0.');
28
29                 }
30
31             }
32
}

```

Logs Tasks Checkpoints Query Editor View Status Progress Problems

User	Application	Operation	Time	Status	Read	Size

Filter Click here to filter the log list

File • Edit • Debug • Test • Workspace • Help • < >

InventoryBatch.apex • OrderTriggerHandler.apex • OrderTrigger.apex

Code Coverage: None • API Version: 64

```

1 • trigger OrderTrigger on HandsMen_Order__c (before insert, before update) {
2
3 •     if (Trigger.isBefore && (Trigger.isInsert || Trigger.isUpdate)) {
4
5         OrderTriggerHandler.validateOrderQuantity(Trigger.new);
6
7     }
8
9 }

```

Logs Tasks Checkpoints Query Editor View Status Progress Problems

User	Application	Operation	Time	Status	Read	Size

Filter Click here to filter the log list