

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

## **Project Overview:**

The Handsmen Threads CRM project transforms the traditional, manual operations of our bespoke tailoring business into a centralized, cloud-based system. By leveraging Salesforce, we aim to streamline client measurement tracking, inventory management, and appointment scheduling, ensuring a seamless experience from the initial fitting to the final suit delivery.

## **Objectives:**

1. **Operational Efficiency:** Automate manual booking and order tracking to reduce human error.
2. **Data Integrity:** Securely store customer body measurements and preferences in a single digital location.
3. **Inventory Control:** Real-time tracking of fabric usage to prevent material shortages.

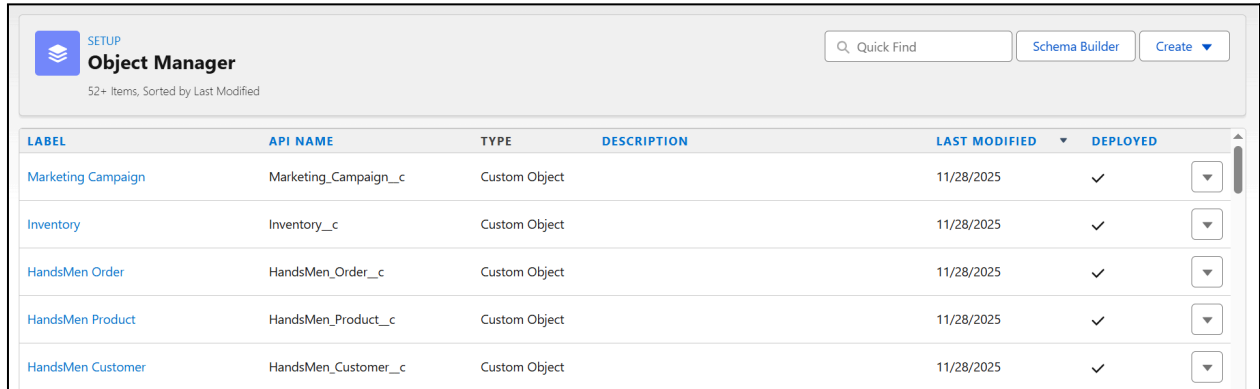
## **Phase 1: Requirement Analysis & Planning**

In this phase, I focused on mapping out the system architecture before implementation. I defined the data structure, business logic, and communication strategies needed to support Handsmen Threads' operations.

### **1. Define Objects, Fields, and Relationships (ERD)**

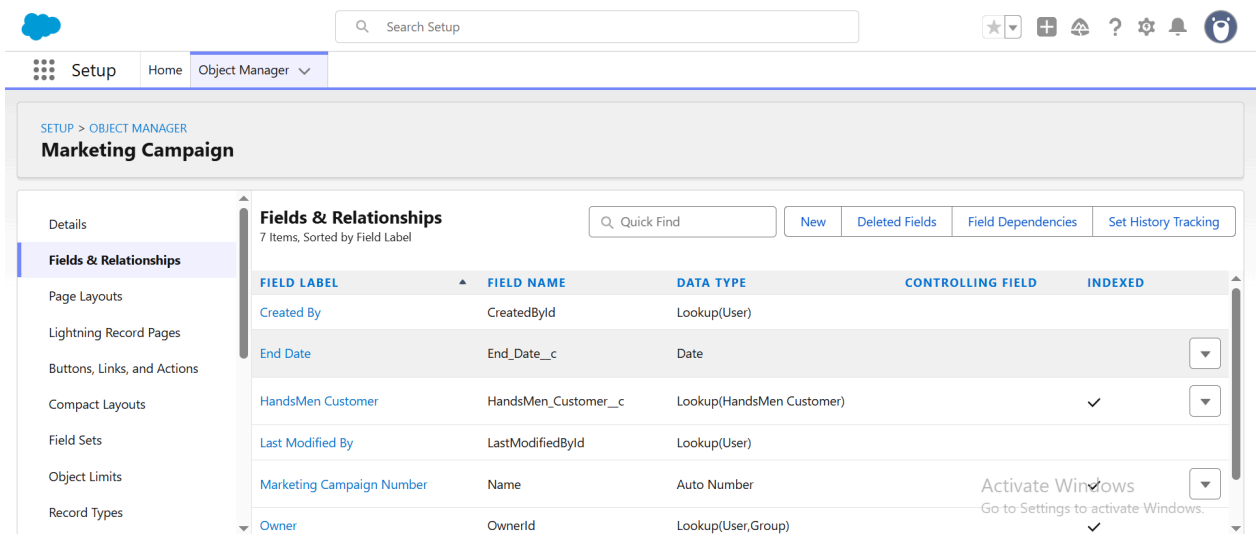
I designed a data model that extends standard Salesforce capabilities. I analyzed the business process to determine necessary Custom Objects.

- **Custom Objects:** HandsMen\_Customer\_\_c (to store customer data), HandsMen\_Order\_\_c (order tracking), HandsMen\_Product\_\_c (product listing), Marketing\_Campaign\_\_c, and Inventory\_\_c.
- **Relationships:** Established Lookup Relationships and Master-Detail relationships between different custom objects to ensure data cascading.



SETUP Object Manager  
52+ Items, Sorted by Last Modified

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Marketing Campaign	Marketing_Campaign__c	Custom Object		11/28/2025	✓
Inventory	Inventory__c	Custom Object		11/28/2025	✓
HandsMen Order	HandsMen_Order__c	Custom Object		11/28/2025	✓
HandsMen Product	HandsMen_Product__c	Custom Object		11/28/2025	✓
HandsMen Customer	HandsMen_Customer__c	Custom Object		11/28/2025	✓



SETUP > OBJECT MANAGER  
Marketing Campaign

Details  
Fields & Relationships  
Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets  
Object Limits  
Record Types

Fields & Relationships  
7 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
End Date	End_Date__c	Date		
HandsMen Customer	HandsMen_Customer__c	Lookup(HandsMen Customer)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Marketing Campaign Number	Name	Auto Number		
Owner	OwnerId	Lookup(User,Group)		✓

Activate Windows  
Go to Settings to activate Windows.

## 2. Establish Business Logic Strategy

I planned the automation logic required to enforce business rules:

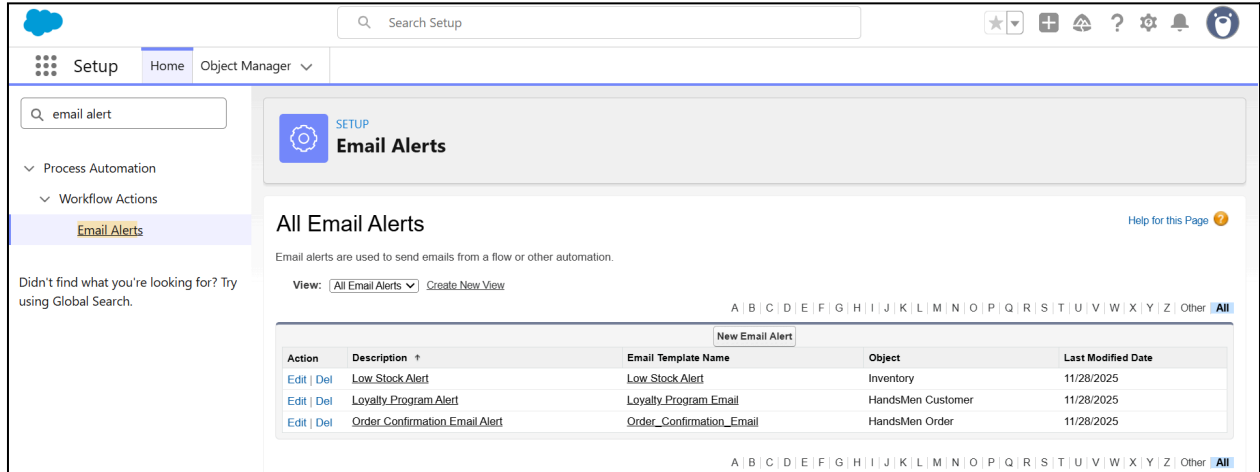
- **Validation Rules:** defined to prevent errors, such as entering negative values for measurements.
- **Flows & Triggers:** Designed workflows to auto-update inventory when an order is placed.
- **Batch Jobs:** Planned a nightly batch job to scan for orders pending delivery for more than 30 days.

## 3. Design Email Templates

I drafted communication templates to keep customers informed.

- *Order Confirmation Email:* Order Confirmation.
- *Low Stock Alert:* “Low Stock” Notification.

- *Loyalty Program Email*: Either Bronze, Silver, or Gold member based on purchases.



The screenshot shows the Salesforce Setup interface. The left sidebar has a search bar with "email alert" and a list of navigation items: Process Automation, Workflow Actions, and Email Alerts (highlighted). The main content area is titled "Email Alerts" and shows a table of existing alerts. The table has columns for Action, Description, Email Template Name, Object, and Last Modified Date. There are three alerts listed: Low Stock Alert, Loyalty Program Alert, and Order Confirmation Email Alert. A "New Email Alert" button is visible above the table.

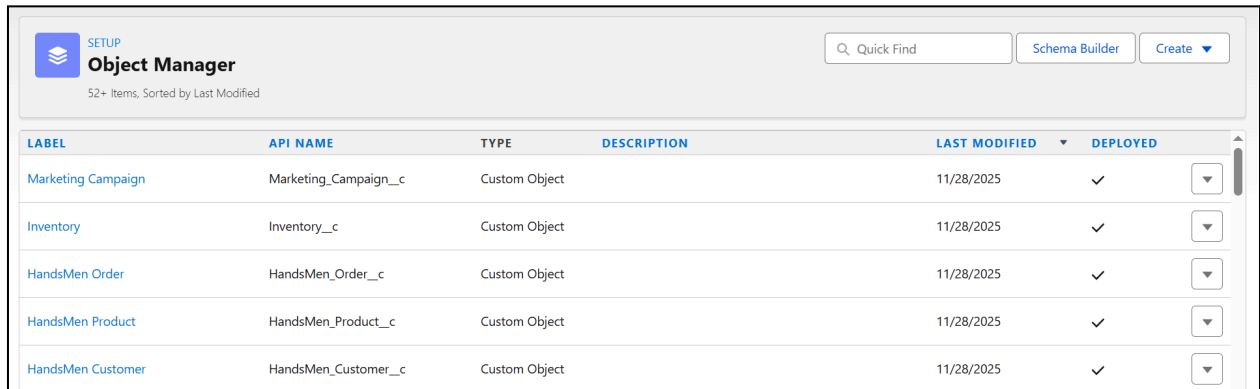
Action	Description	Email Template Name	Object	Last Modified Date
<a href="#">Edit</a>   <a href="#">Del</a>	Low Stock Alert	Low Stock Alert	Inventory	11/28/2025
<a href="#">Edit</a>   <a href="#">Del</a>	Loyalty Program Alert	Loyalty Program Email	HandsMen Customer	11/28/2025
<a href="#">Edit</a>   <a href="#">Del</a>	Order Confirmation Email Alert	Order Confirmation Email	HandsMen Order	11/28/2025

## Phase 2: Salesforce Development - Backend & Configurations

This phase involved the actual construction and configuration of the Salesforce environment based on the architecture defined in Phase 1.

### 1. Object and Field Creation

I built the custom objects and fields in the Object Manager. I configured page layouts to ensure the interface is user-friendly for the tailors and store managers.



The screenshot shows the Salesforce Object Manager interface. The top bar includes a search bar with "Quick Find", a "Schema Builder" button, and a "Create" button. Below the header, there is a table listing custom objects. The table has columns for LABEL, API NAME, TYPE, DESCRIPTION, LAST MODIFIED, and DEPLOYED. Five objects are listed: Marketing Campaign, Inventory, HandsMen Order, HandsMen Product, and HandsMen Customer.

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
<a href="#">Marketing Campaign</a>	Marketing_Campaign__c	Custom Object		11/28/2025	✓
<a href="#">Inventory</a>	Inventory__c	Custom Object		11/28/2025	✓
<a href="#">HandsMen Order</a>	HandsMen_Order__c	Custom Object		11/28/2025	✓
<a href="#">HandsMen Product</a>	HandsMen_Product__c	Custom Object		11/28/2025	✓
<a href="#">HandsMen Customer</a>	HandsMen_Customer__c	Custom Object		11/28/2025	✓

SETUP > OBJECT MANAGER

**HandsMen Customer**

Details

**Fields & Relationships**

11 Items, Sorted by Field Label

Q, Quick Find

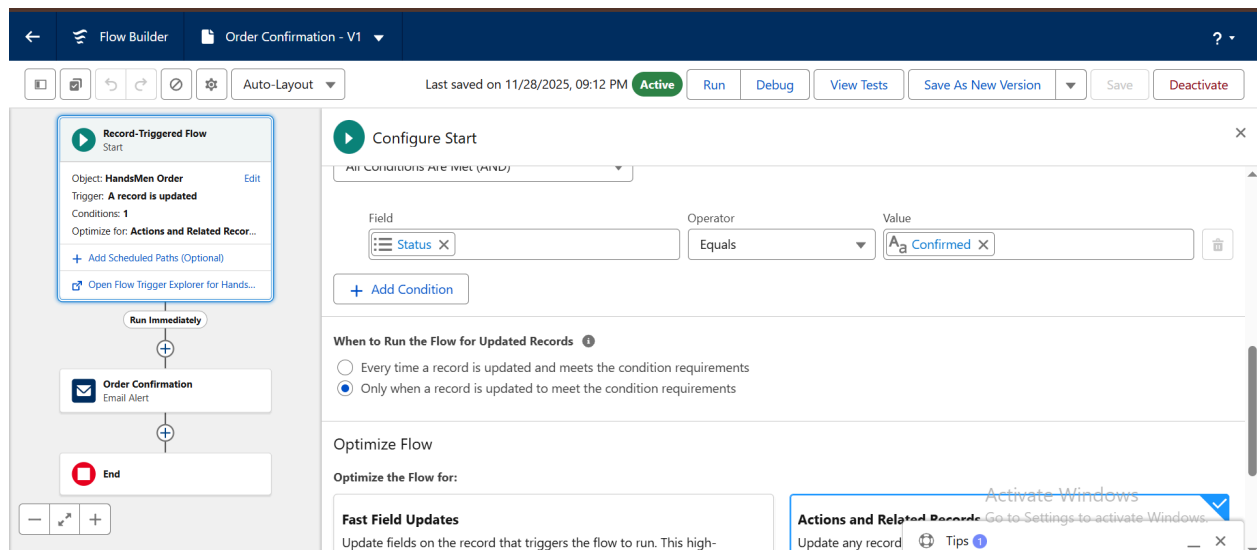
New Deleted Fields Field Dependencies Set History Tracking

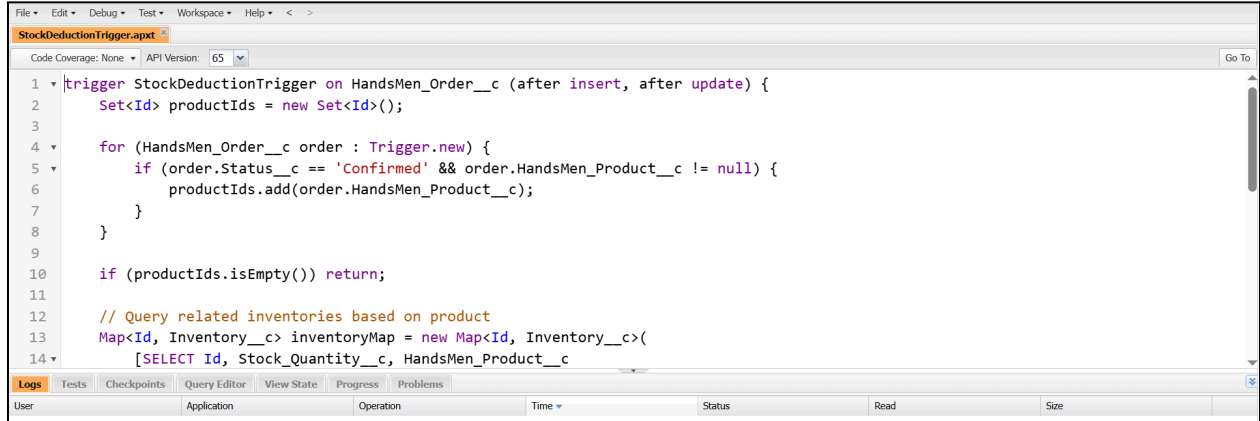
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Email	Email__c	Email		
FirstName	FirstName__c	Text(60)		
FullName	FullName__c	Formula (Text)		
HandsMen Customer Name	Name	Text(80)		✓
Last Modified By	LastModifiedById	Lookup(User)		
LastName	LastName__c	Text(60)		
Loyalty Status	Loyalty_Status__c	Picklist		
Owner	OwnerId	Lookup(User/Group)		✓
Phone	Phone__c	Phone		
Total Purchases	Total_Purchases__c	Number(18, 0)		

Activate Windows  
Go to Settings to activate Windows.

## 2. Implement Automation

- **Flows:** I utilized Salesforce Flow to handle email scheduling. When a customer confirms an order, the flow automatically sends an email confirmation and updates the customer record.
- **Apex Triggers:** Developed an Apex Trigger on the Handsmen\_Order\_\_c object. When a new order is placed, the trigger checks the available quantity in the Handsmen\_Product\_\_c inventory. It automatically deducts the ordered quantity from the inventory levels to keep stock accurate, and blocks the transaction if the stock is insufficient.

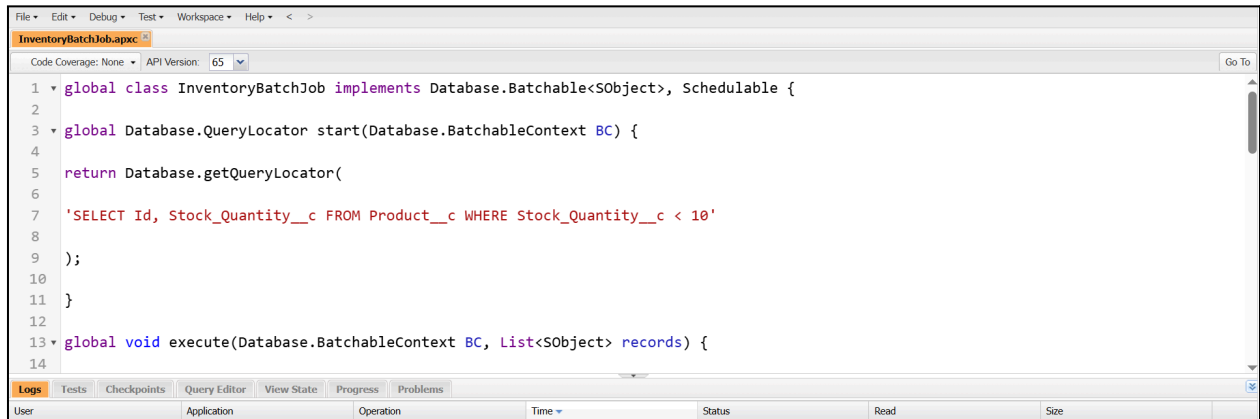




```
1 trigger StockDeductionTrigger on HandsMen_Order__c (after insert, after update) {
2     Set<Id> productIds = new Set<Id>();
3
4     for (HandsMen_Order__c order : Trigger.new) {
5         if (order.Status__c == 'Confirmed' && order.HandsMen_Product__c != null) {
6             productIds.add(order.HandsMen_Product__c);
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>([SELECT Id, Stock_Quantity__c, HandsMen_Product__c
14
```

### 3. Develop Batch Jobs

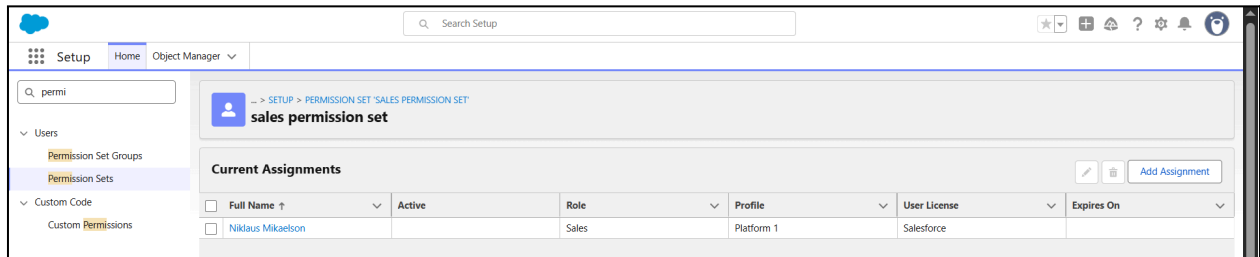
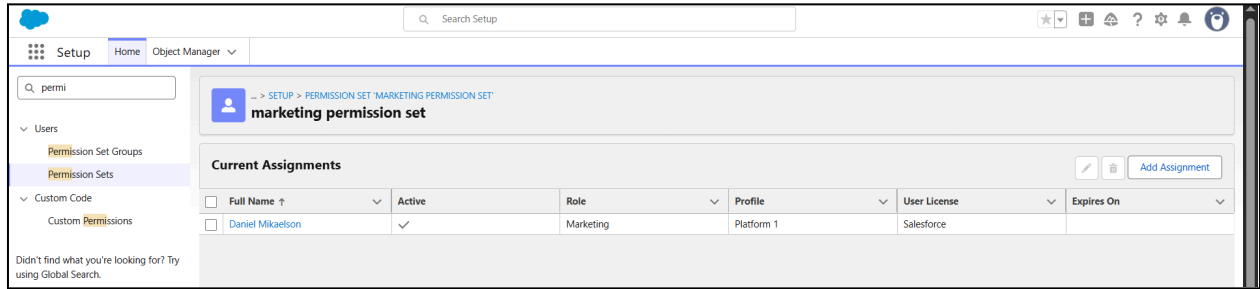
I developed a Batch Apex class named InventoryBatchJob that implements the Schedulable interface to ensure inventory stability. This job is scheduled to run periodically to scan for products with low stock levels (less than 10 units). When identified, the system automatically simulates a restocking process by increasing the Stock\_Quantity\_\_c by 50 units, ensuring that popular fabrics are always available for new orders.



```
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
```

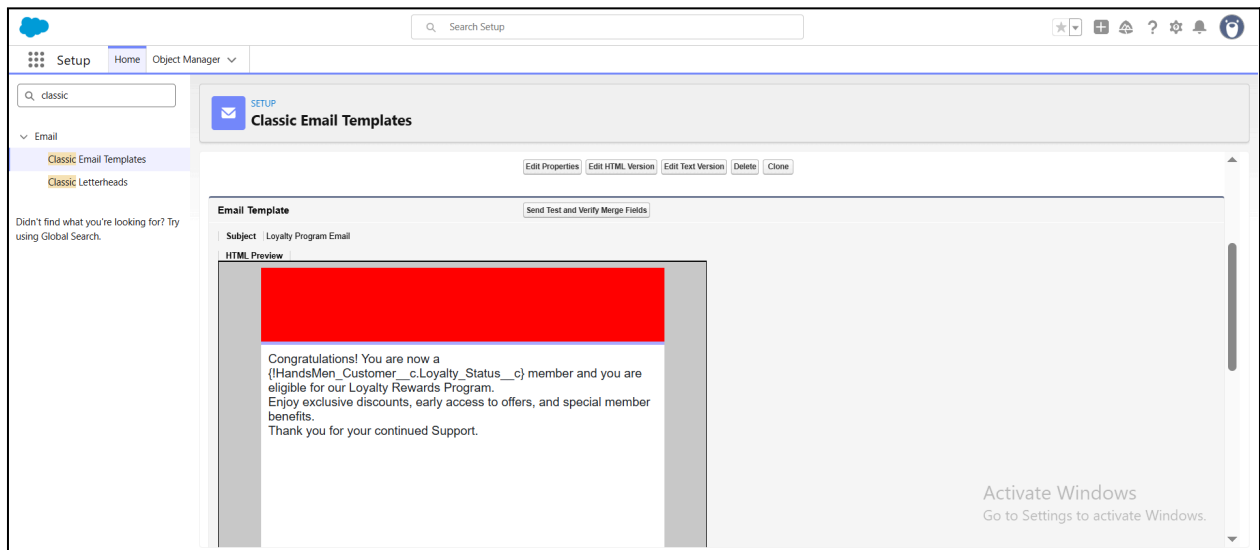
### 4. Data Security and Sharing Rules

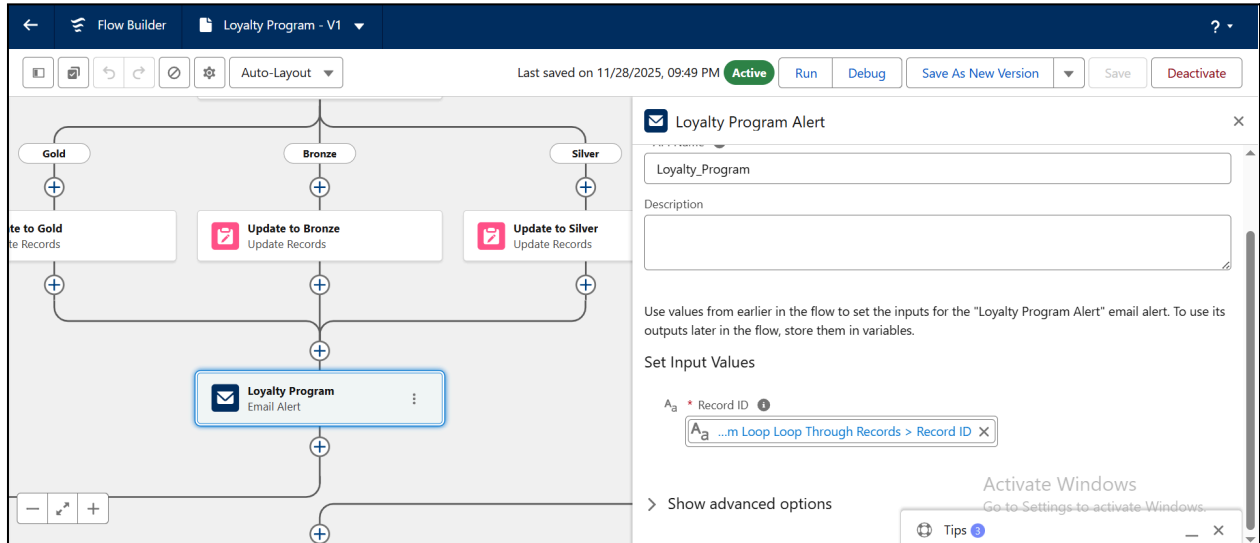
- Sharing Rules: Created rules to assign “Sales Permission Set” with the “Sales” role, while “Marketing Permission Set” are only assigned with the “Marketing” role.



## 5. Configure Email Templates & Notifications

I implemented the HTML email templates and set up Email Alerts within our Flows to send them automatically upon specific triggers.





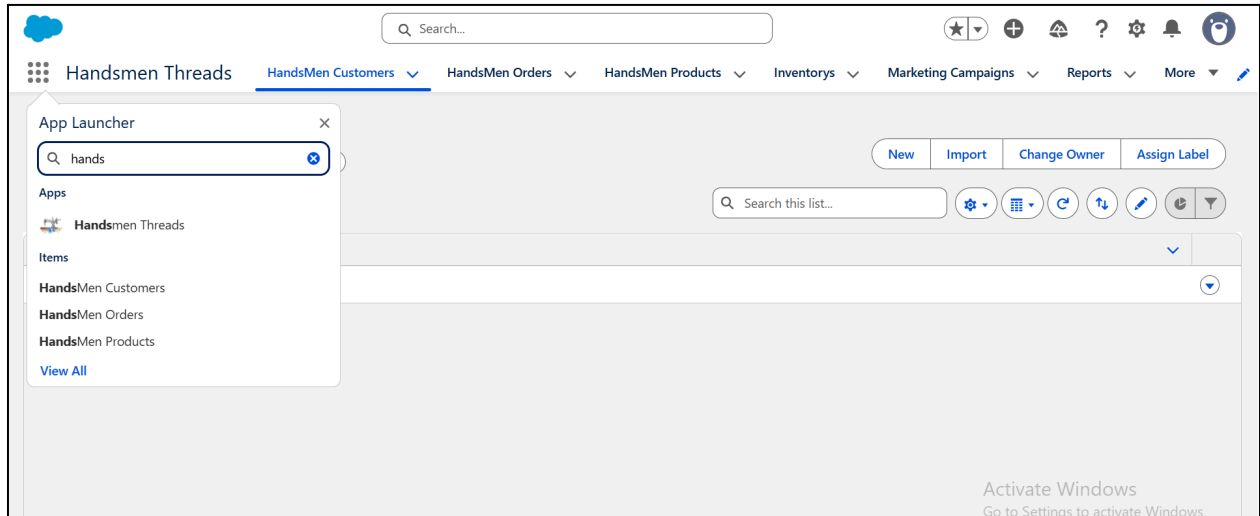
## Phase 3: UI/UX Development & Customization

The screenshot displays the Salesforce Lightning Experience App Manager interface. The left sidebar shows the navigation menu with options like Setup, Home, and Object Manager. The main area shows a list of installed apps, including Data Loader Bulk, Data Loader Partner, Digital Experiences, Force.com IDE, Handsmen Threads, Lightning Usage A..., Marketing CRM CL..., My Service Journey, Platform, Playground Starter, Sales, LightningSales, Sales Cloud Mobile, Sales Console, and Salesforce Chatter.

Lightning Experience App Manager

32 items • Sorted by App Name • Filtered by All appnumentems - TabSet Type, App Type

App Name	Developer Name	Description	Last Modified	App Type	Visibility
11 Dataloader Bulk	Dataloader_Bulk	The Data Loader is an easy to use graphical tool that helps you to get your data into Salesforce objects.	11/6/2025, 3:55 PM	Connected (Managed)	
12 Dataloader Partner	Dataloader_Partner	The Data Loader is an easy to use graphical tool that helps you to get your data into Salesforce objects.	11/6/2025, 3:55 PM	Connected (Managed)	
13 Digital Experiences	SalesforceCMS	Manage content and media for all of your sites.	11/6/2025, 3:55 PM	Lightning	✓
14 Force.com IDE	Forcecom_IDE	The Force.com IDE is a powerful client application for creating, modifying, testing and deploying Force.com ap...	11/6/2025, 3:55 PM	Connected (Managed)	
15 Handsmen Threads	Handsmen_Threads		11/28/2025, 7:49 PM	Lightning	✓
16 Lightning Usage A...	LightningInstrumentation	View Adoption and Usage Metrics for Lightning Experience	11/6/2025, 3:55 PM	Lightning	✓
17 Marketing CRM CL...	Marketing	Track sales and marketing efforts with CRM objects.	11/6/2025, 3:55 PM	Classic	✓
18 My Service Journey	MSIApp	Discover new customer service capabilities.	11/6/2025, 4:03 PM	Lightning	✓
19 Platform	Platform	The fundamental Lightning Platform	11/6/2025, 3:55 PM	Classic	✓
20 Playground Starter	Playground_Starter	Get started with your Trailhead Playground.	11/6/2025, 3:55 PM	Lightning (Managed)	✓
21 Sales	Sales	The world's most popular sales force automation (SFA) solution	11/6/2025, 3:55 PM	Classic	✓
22 Sales	LightningSales	Manage your sales process with accounts, leads, opportunities, and more	11/6/2025, 3:55 PM	Lightning	✓
23 Sales Cloud Mobile	SalesCloudMobile	New seller focused mobile first experience	11/6/2025, 3:55 PM	Lightning	✓
24 Sales Console	LightningSalesConsole	(Lightning Experience) Lets sales reps work with multiple records on one screen	11/6/2025, 3:55 PM	Lightning	✓
25 Salesforce Chatter	Chatter	The Salesforce Chatter social network, including profiles and feeds	11/6/2025, 3:55 PM	Classic	✓



## Phase 4: Data Migration, Testing & Security

In this phase, I validated the system architecture to ensure that the custom objects (HandsMen Customer\_\_c, HandsMen Product\_\_c, etc.) function correctly and that data integrity is maintained through formulas and automation.

### 1. Data Quality & Integrity Implementation

To automate data formatting and provide instant visual feedback on inventory levels, we implemented the following formula fields:

- **Inventory Status Indicator:** I created a formula field Stock\_Status\_\_c on the Inventory\_\_c object. This text formula automatically flags items as “Low Stock” if the quantity drops to 10 or below, allowing the warehouse manager to spot shortages immediately without running a report.
  - *Formula Logic:* IF(Stock\_Quantity\_\_c > 10, “Available”, “Low Stock”)
- **Customer Name Concatenation:** On the HandsMen Customer\_\_c object, I created the Full\_Name\_\_c formula to combine the First and Last names into a single readable field for easier search and reporting.
  - *Formula Logic:* FirstName & " " & LastName

### 3. Functional Test Cases & Results

I executed functional tests to verify the formulas and object behaviors defined in the schema.

#### Test Case 1: Inventory Status Formula Logic



- **Objective:** Verify that the Stock\_Status\_\_c field on the **Inventory\_\_c** object correctly updates based on the Stock\_Quantity\_\_c.
- **Step 1 (Input):** User creates a new Inventory record and enters 8 in the Stock\_Quantity\_\_c field.

The screenshot shows a 'Create New Inventory' modal form in Salesforce. The form is titled 'Information' and includes a legend indicating that fields with an asterisk (\*) are required. The fields are: 'Inventory Number' (with a value of 'I-0001'), 'HandsMen Product' (a dropdown menu showing 'T-shirt cloth'), 'Stock Quantity' (a text input field with the value '8'), and 'Warehouse' (an empty text input field). At the bottom of the modal, there are three buttons: 'Cancel', 'Save & New', and 'Save'. The background shows the 'Inventories' list page with a search bar and various navigation icons.

- **Step 2 (Output):** Upon saving, the Stock\_Status\_\_c field automatically displays “Low Stock”.

The screenshot shows the 'Inventory Details' page in Salesforce. The page title is 'Inventory' and the sub-title is 'I-0002'. The page includes a 'Details' tab and a 'Related' section. The 'Details' section displays the following information: 'Inventory Number' (I-0002), 'HandsMen Product' (T-shirt cloth), 'Stock Quantity' (8), 'Stock Status' (Low Stock), and 'Warehouse' (empty). The 'Related' section shows 'Created By' (Josephine Claire Nogot, 11/29/2025, 12:07 AM) and 'Last Modified By' (Josephine Claire Nogot, 11/29/2025, 12:07 AM). The page also features a 'New Contact' button and a 'New Opportunity' button. The background shows the 'Inventories' list page with a search bar and various navigation icons.

## Test Case 2: Customer Name Concatenation

- **Objective:** Verify that the Full\_Name\_\_c formula on **HandsMen Customer\_\_c** works as expected.
- **Step 1 (Input):** User creates a new customer record with First Name: “Daniel” and Last

Name: **“Padilla”**.

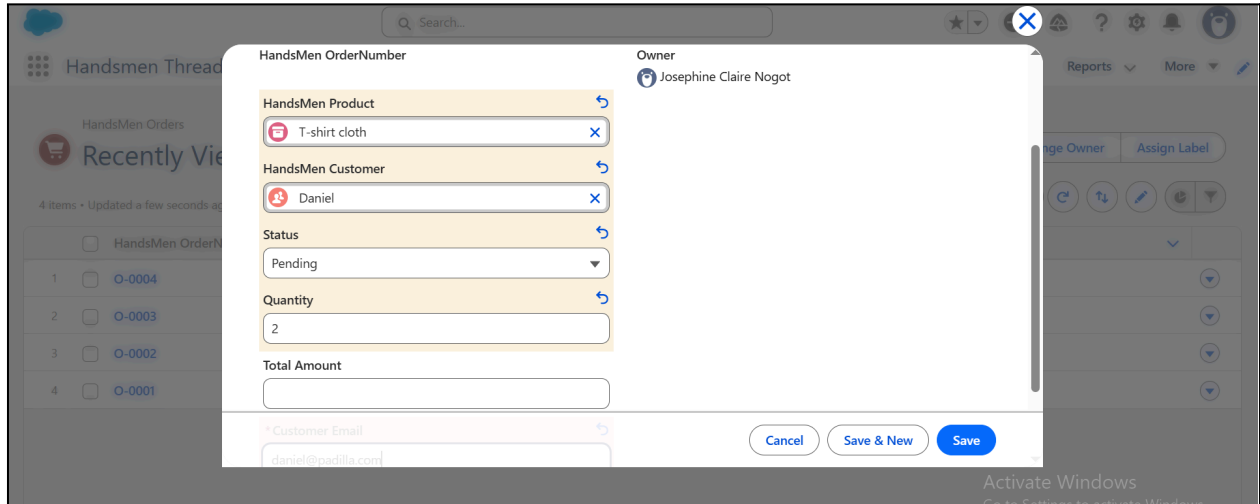
The screenshot shows a mobile application interface with a modal form for editing a customer profile. The form fields are: Phone (empty), Loyalty Status (dropdown menu showing "--None--"), FirstName (text input with "Daniel"), LastName (text input with "Padilla"), and Total Purchases (empty). At the bottom of the modal are three buttons: "Cancel", "Save & New", and "Save". The background shows a list of customers with "Daniel" selected.

- **Step 2 (Output):** The system saves the record, and the Full\_Name\_\_c field displays **“Daniel Padilla”**.

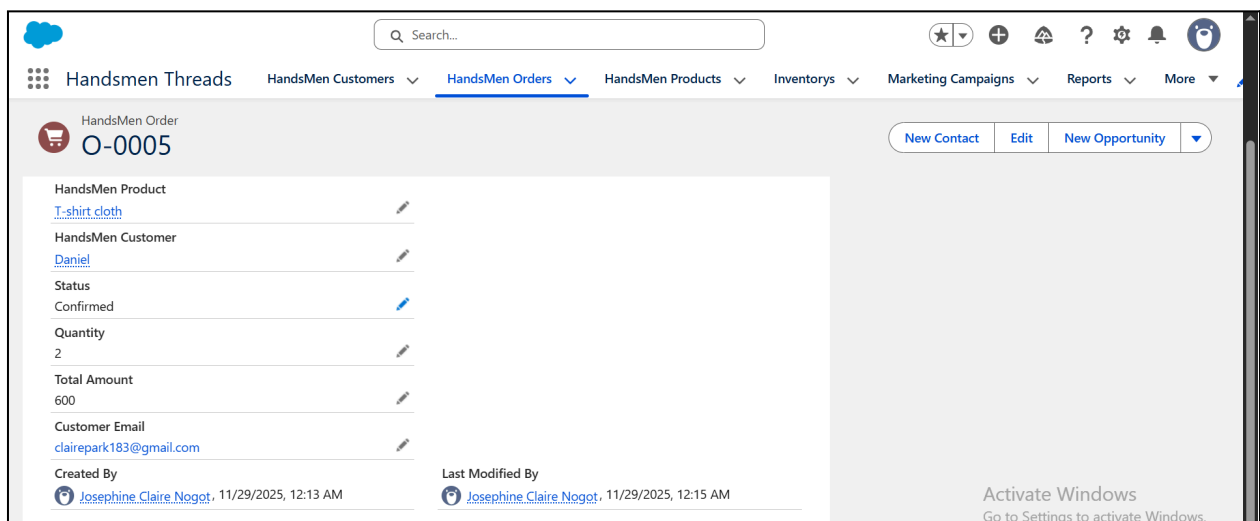
The screenshot shows a desktop web application interface for a customer profile. The profile is for "Daniel Padilla". The fields shown are: Phone, Loyalty Status, FirstName (Daniel), LastName (Padilla), FullName (Daniel Padilla), and Total Purchases. Each field has an edit icon. At the bottom, it shows "Created By" and "Last Modified By" as "Josephine Claire Nogot" on "11/29/2025, 12:09 AM". The top navigation bar includes "HandsMen Threads", "HandsMen Customers", "HandsMen Orders", "HandsMen Products", "Inventories", "Marketing Campaigns", "Reports", and "More".

### Test Case 3: Order Creation & Status Flow

- **Objective:** Verify that a **HandsMen Order\_\_c** can be created and tracked via the Status picklist.
- **Step 1 (Input):** User creates a new order, linking it to a customer, setting Quantity\_\_c to 2, and Status to **“Pending”**.



- **Step 2 (Output):** The record is saved successfully. Later, the user edits the Status to “Confirmed”.



## Phase 5:Deployment, Documentation & Maintenance

When a user reports an error (e.g., “I can't save this Order”), I follow this standardized troubleshooting workflow:

1. **Reproduce the Issue:** Log in as the user (using “Login As”) to see if the error is replicable.
2. **Check Validation Rules:** 90% of save errors are due to Validation Rules. We check if the data input violates rules like the “Low Stock” trigger.
3. **Review Debug Logs:** If the error is technical (like an Apex Trigger failure), we turn on Debug Logs for that user to trace the specific line of code causing the exception.
4. **Flow Fault Emails:** We configured the system to send an email to the Admin whenever a

Flow fails, providing immediate context on what went wrong.

## **Conclusion**

The Handsmen Threads Capstone Project successfully achieved its primary objective: transforming a traditional, manual bespoke tailoring operation into a modernized, digital-first business.

By implementing Salesforce, we addressed the critical pain points of data redundancy and inventory mismanagement. The custom architecture, specifically the HandsMen Customer\_\_c and HandsMen Order\_\_c objects, now provides a “Single Source of Truth” for client measurements and transaction history. The automation we built does more than just save time; the Stock Validation Trigger actively prevents revenue loss by stopping orders for out-of-stock fabrics, while the InventoryBatchJob ensures the warehouse is proactive rather than reactive.

This project demonstrates that Salesforce is not just for large enterprises but can be effectively tailored to small businesses. The system is scalable, secure, and ready to support Handsmen Threads as they expand their customer base, providing a solid foundation for future enhancements like customer self-service portals and automated marketing journeys.