

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

**Salesforce Talent Accelerator Program  
(STAP)**

**Submitted by:**  
Naura Hirzia Nawarika

**Universitas Pendidikan Indonesia**

## A. PROJECT OVERVIEW

To assist with operational efficiency and customer experience, HandsMen Threads built a fully customized Salesforce CRM. Most processes related to customer data, orders, inventory, and marketing campaigns are streamlined through the brand's internal systems.

At the center system's structure is a data model with five main custom objects: Customer, Order, Product, Inventory, and Marketing Campaign. This structure facilitates important workflows and uniformity of data across business activities.

The CRM uses several automation features, such as Record-Triggered Flows, Scheduled Flows, Email Alerts, and Apex processes. These automations enable seamless order handling, timely customer notifications, loyalty program management, and proactive inventory oversight, reducing the need for manual work.

The systems use role-based access for Sales, Inventory, and Marketing teams to secure data and restrict access appropriately. Inventory management is assisted with a Scheduled Apex Batch that updates stock status to automate marginal availability and to keep the uninterrupted status.

By combining automation, structured data management, and secure access controls, the Salesforce CRM implementation strengthens HandsMen Threads' ability to engage customers effectively, optimizes internal operations, and establishes a scalable framework for future growth.

## B. OBJECTIVE

This project is focused on implementing a customized Salesforce CRM for HandsMen Threads, designed to centralize customer, order, product, inventory, and marketing data into a single, cohesive platform. The system seeks to streamline operations, enhance data accuracy, and deliver improved customer engagement.

The specific objectives of the project are to:

- **Enhance customer management** by consolidating profiles, purchase history, and interactions to enable more effective tracking and personalized engagement.

- **Streamline order and inventory processes** through automation, including real time stock monitoring and notifications to minimize manual effort.
- **Increase operational efficiency** by standardizing workflows and reducing repetitive tasks across Sales, Inventory, and Marketing teams.
- **Support secure and collaborative team operations** with role-based access controls and coordinated processes to ensure data integrity and smooth interdepartmental collaboration.

## C. DETAILED EXECUTION OF PROJECT PHASE

### a. Requirement Analysis & Planning

The first step focused on evaluating the challenges the Salesforce CRM can solve. The developer identified essential needs, including accurate tracking of customers, orders, and products, effective inventory management, improved customer communication through timely notifications, a loyalty program, and automation of repetitive tasks to enhance efficiency.

Based on this analysis, the project scope was defined to include customer and order management, inventory tracking, a loyalty program, automated notifications, and reporting dashboards, while excluding physical tailoring work, manual delivery tracking, and unrelated software. The goals of the project include unifying business data on a single system, automating important processes, improving customer communication with reward personalization, and ensuring the company has consistent and reliable data to make better business decisions.

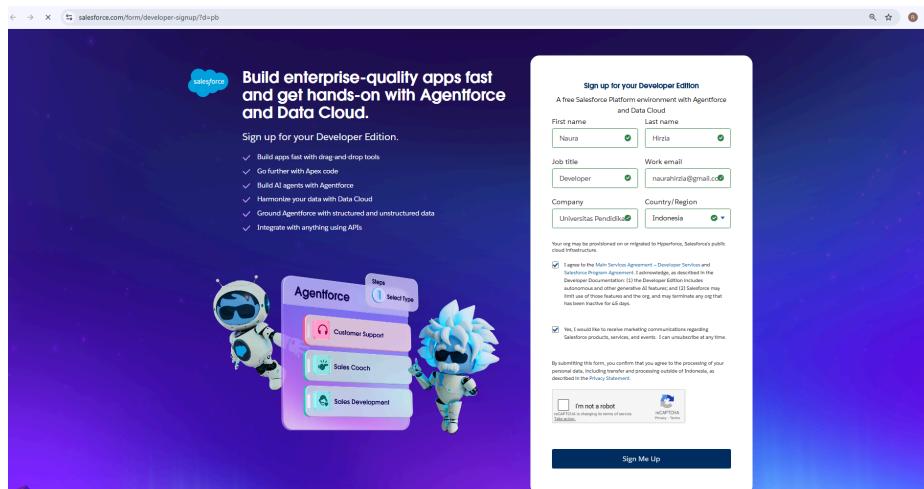
A comprehensive data model was developed, connecting Orders, Products, Customers, and Loyalty Programs, while a security model was implemented with roles, profiles, and field-level access to protect sensitive information and ensure smooth workflows. Primary stakeholders were also selected, which included the Sales Team for operational frontline and Inventory Managers for stock level oversight. This design ensures the CRM enhances operational efficiency, data security, and overall stakeholder satisfaction.

## b. Salesforce Development - Backend & Configuration

### i. Environment Setup & DevOps Workflow

The project was built and tested within a Salesforce Developer Org, a personal workspace that allows safe configuration of objects, fields, automation, and Apex code.

The developer created the org via [Salesforce Developer Signup](#), completed account verification, and gained access to the Setup interface. This environment served as the main platform for developing, testing, and validating all CRM components before deployment to the live system.



### ii. Custom Objects and Fields

- **HandsMen Customer** : Stores customer details: Name, Email, Phone, Loyalty\_Status\_\_c (Bronze, Silver, Gold), Total\_Purchases\_\_c
- **HandsMen Product** : Stores product information: Name, SKU, Price, Stock\_Quantity\_\_c
- **HandsMen Order** : Tracks customer orders: Order\_Number, Status (Pending, Confirmed, Rejected), Quantity\_\_c, Total\_Amount\_\_c
- **Inventory** : Monitors stock levels: Auto Number, Stock\_Quantity\_\_c, Warehouse

- **Marketing Campaign** : Manages promotions: Campaign\_Name, Start\_Date, End\_Date

### iii. Validation Rules

#### 1) Inventory Stock Quantity

Prevents saving any inventory record with a stock quantity of zero or less. If someone tries, Salesforce stops the action and displays the error message at the top of the page, ensuring inventory data remains accurate.

**Rule Name:** Stock\_Quantity

**Object:** Inventory

**Formula:** Stock\_Quantity\_\_c <= 0

**Error Message:** “The inventory count is never less than zero.”

SETUP > OBJECT MANAGER  
Inventory

<a href="#">Details</a> <a href="#">Fields &amp; Relationships</a> <a href="#">Page Layouts</a> <a href="#">Lightning Record Pages</a> <a href="#">Buttons, Links, and Actions</a> <a href="#">Compact Layouts</a> <a href="#">Field Sets</a> <a href="#">Object Limits</a> <a href="#">Record Types</a> <a href="#">Related Lookup Filters</a> <a href="#">Search Layouts</a> <a href="#">List View Button Layout</a> <a href="#">Restriction Rules</a> <a href="#">Scoping Rules</a> <a href="#">Object Access</a> <a href="#">Triggers</a> <a href="#">Flow Triggers</a> <b>Validation Rules</b> <a href="#">Conditional Field Formatting</a>	<p><b>Inventory Validation Rule</b></p> <p>Define a validation rule by specifying an error condition and a corresponding error message. The error condition is written as a Boolean formula</p> <p><b>Validation Rule Edit</b></p> <p>Rule Name: Stock_Quantity Active: <input checked="" type="checkbox"/> Description: <input type="text"/></p> <p><b>Error Condition Formula</b></p> <p>Example: Discount_Percent__c &lt; 0.30 <a href="#">More Examples...</a> Display an error if Discount is more than 30% If this formula expression is true, display the text defined in the Error Message area</p> <p>Insert Field <a href="#">Use Formula Assistant</a> Insert Operator <input type="button" value="▼"/></p> <p><code>Stock_Quantity__c &lt;= 0</code></p> <p><b>Functions</b></p> <p>-- All Function Categories --</p> <ul style="list-style-type: none"> <li>ABS</li> <li>ACOS</li> <li>ADDMONTHS</li> <li>AND</li> <li>ASCII</li> <li>ASIN</li> </ul> <p><a href="#">Insert Selected Function</a> ABS(number) Returns the absolute value of a number, a number without its sign <a href="#">Help on this function</a></p> <p><b>Error Message</b></p> <p>Example: Discount percent cannot exceed 30% This message will appear when Error Condition formula is true Error Message: the inventory count is never less than zero.</p> <p>This error message can either appear at the top of the page or below a specific field on the page Error Location: <input checked="" type="radio"/> Top of Page <input type="radio"/> Field <a href="#">i</a></p> <p><input type="button" value="Save"/> <input type="button" value="Save &amp; New"/> <input type="button" value="Cancel"/></p>
--	--

#### 2) Customer Email

Prevents saving any customer record that does not contain a valid Gmail address. If the entered email lacks "@gmail.com", Salesforce stops the save and displays the error message, ensuring only valid Gmail addresses are recorded.

**Rule Name:** Email

**Object:** HandsMen Customer

**Formula:** NOT CONTAINS( Email\_\_c , "@gmail.com")

**Error Message:** "Please fill Correct Gmail"

SETUP > OBJECT MANAGER  
HandsMen Customer

Validation Rule Edit

Rule Name: Email  
Active:   
Description:

Error Condition Formula

Example: Discount\_Percent\_c>0.30 More Examples...  
Display an error if Discount is more than 30%  
If this formula expression is true, display the text defined in the Error Message area

Insert Field Use Formula Assistant Insert Operator ▾  
NOT CONTAINS( Email\_\_c , "@gmail.com")

Functions

- All Function Categories -- ▾
- ABS
- ACOS
- ADDMONTHS
- AND
- ASCII
- ASIN

Insert Selected Function  
ABS(number)  
Returns the absolute value of a number, a number without its sign  
Help on this function

Error Message

Example: Discount percent cannot exceed 30%  
This message will appear when Error Condition formula is true  
Error Message: Please fill Correct Gmail

This error message can either appear at the top of the page or below a specific field on the page  
Error Location: Top of Page  Field

Save Save & New Cancel

### 3) Total Amount

Prevents saving any order with a total amount of zero or less. If a user tries to save such an order, Salesforce stops the action and displays the error message next to the Total Amount field, ensuring all orders have a valid total.

**Rule Name:** Total\_Amount

**Object:** HandsMen Order

**Formula:** Total\_Amount\_\_c <= 0

**Error Message:** "Please Enter Correct Amount"

The screenshot shows the 'Validation Rule Edit' screen for the 'HandsMen Order' object. The left sidebar lists various configuration options under 'SETUP > OBJECT MANAGER'. The 'Validation Rules' section is selected. The main area displays the validation rule details:

- Rule Name:** Total\_Amount
- Active:** Checked
- Description:** (Empty)
- Error Condition Formula:** Total\_Amount\_\_c <= 0
- Error Message:** Please Enter Correct Amount

A formula builder interface is shown, allowing the user to build the formula using functions like ABS, ACOS, ADDMONTHS, AND, ASCII, ASIN, etc. A tooltip for the ABS function is visible, stating it returns the absolute value of a number.

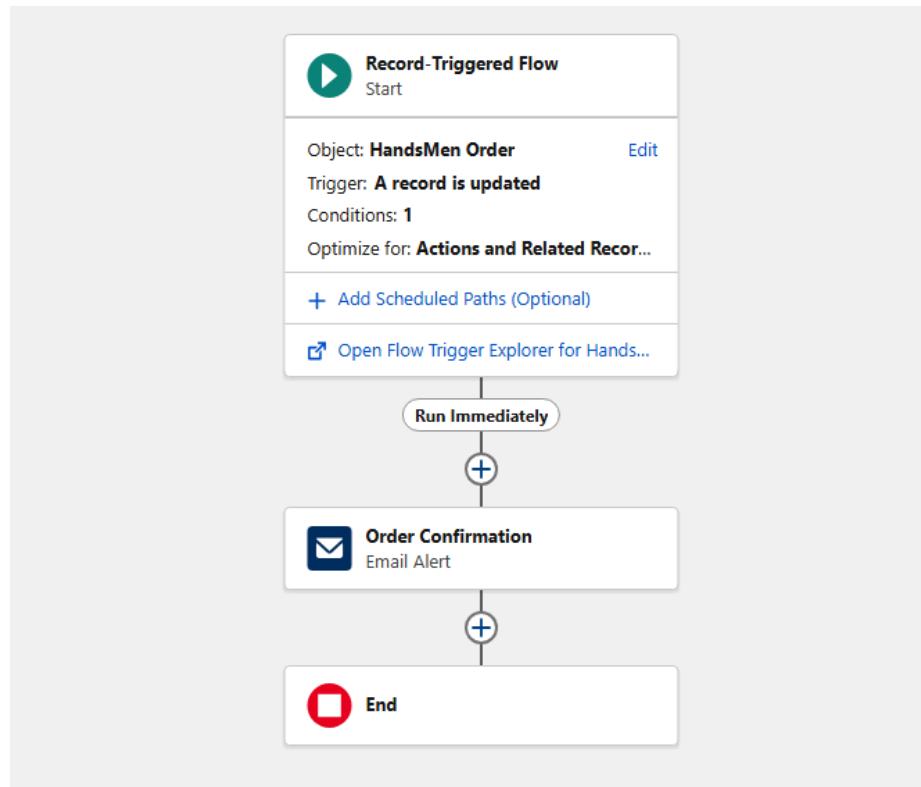
#### iv. Automation (Flows, Workflow Rules, Approval Processes)

##### 1) Order Confirmation Flow

Automatically sends a confirmation email to the customer once their order is confirmed, ensuring timely notification without manual intervention.

**Type:** Record-Triggered Flow

**Trigger:** When an order's status is Confirmed

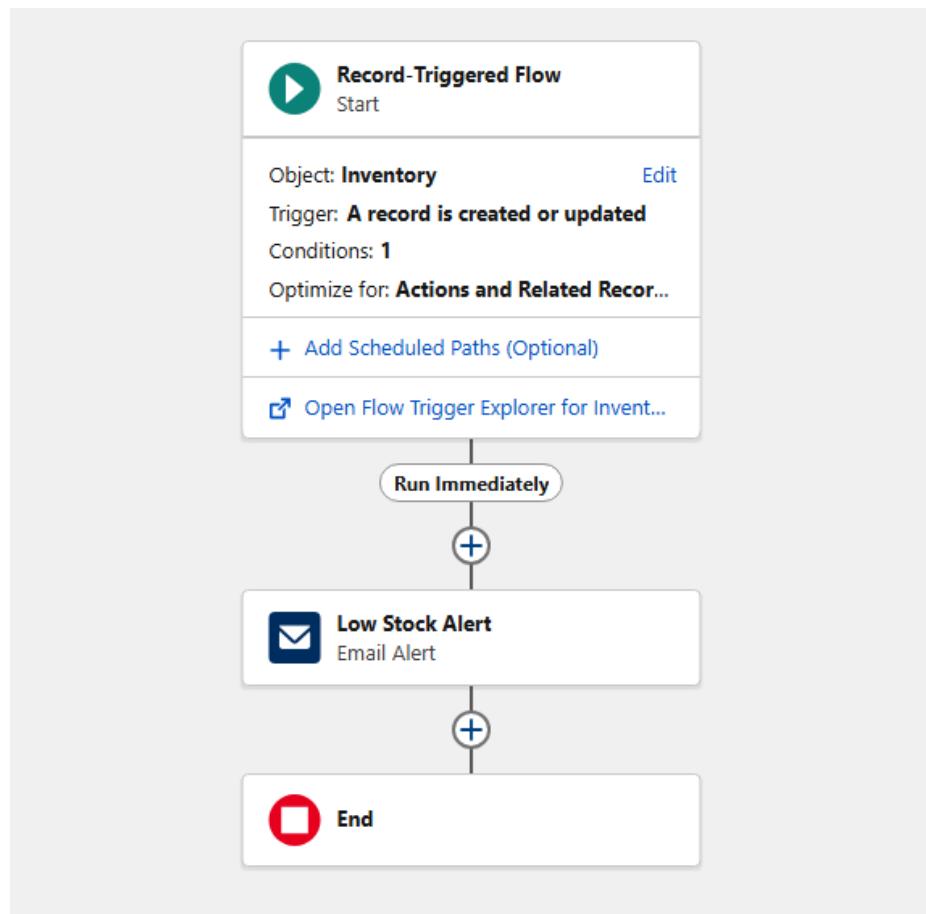


## 2) Stock Alert Flow

Sends an alert to the staff to restock the item before it runs out, preventing stockouts and ensuring smooth operations.

**Type:** Record-Triggered Flow

**Trigger:** When an inventory item's stock quantity drops below 5

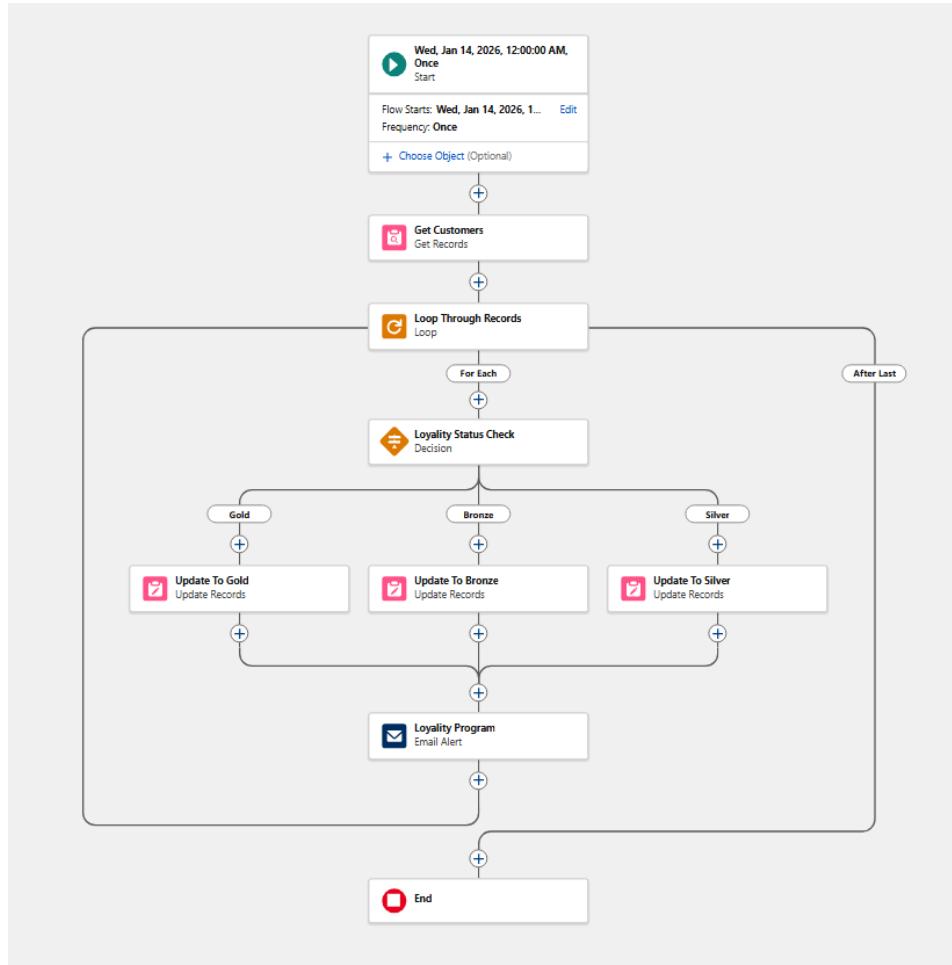


### 3) Loyalty Status Update Flow

Automatically updates the Loyalty\_Status\_\_c field for customers based on their points or activity, keeping the loyalty program current without manual input.

**Type:** Scheduled Flow

**Trigger:** Runs daily

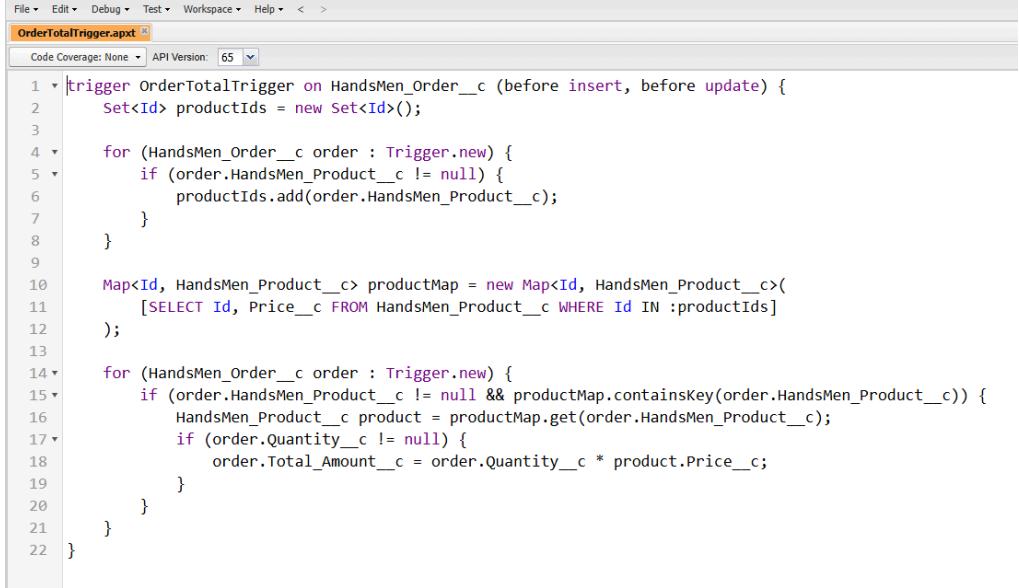


#### iv. Apex Classes & Triggers

##### 1) OrderTotalTrigger

- **Purpose:** Automatically calculates the total amount of each order based on product price and quantity. Ensures the Total\_Amount\_\_c field is accurate before saving, preventing manual errors.
- **How It Works:**
  - 1. Collect Product IDs:** The trigger scans incoming orders during insert or update. Product IDs from orders are added to a set to identify which products need pricing information.

2. **Query Product Prices:** Products in the set are queried for their Price\_\_c values. Prices are stored in a map (productMap) with product IDs as keys for efficient lookup.
3. **Calculate Total Amount:** The trigger loops through orders again, retrieves product prices from the map, and multiplies by the order quantity to compute the total amount.



```

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
OrderTotalTrigger.apxt
Code Coverage: None API Version: 65
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 |

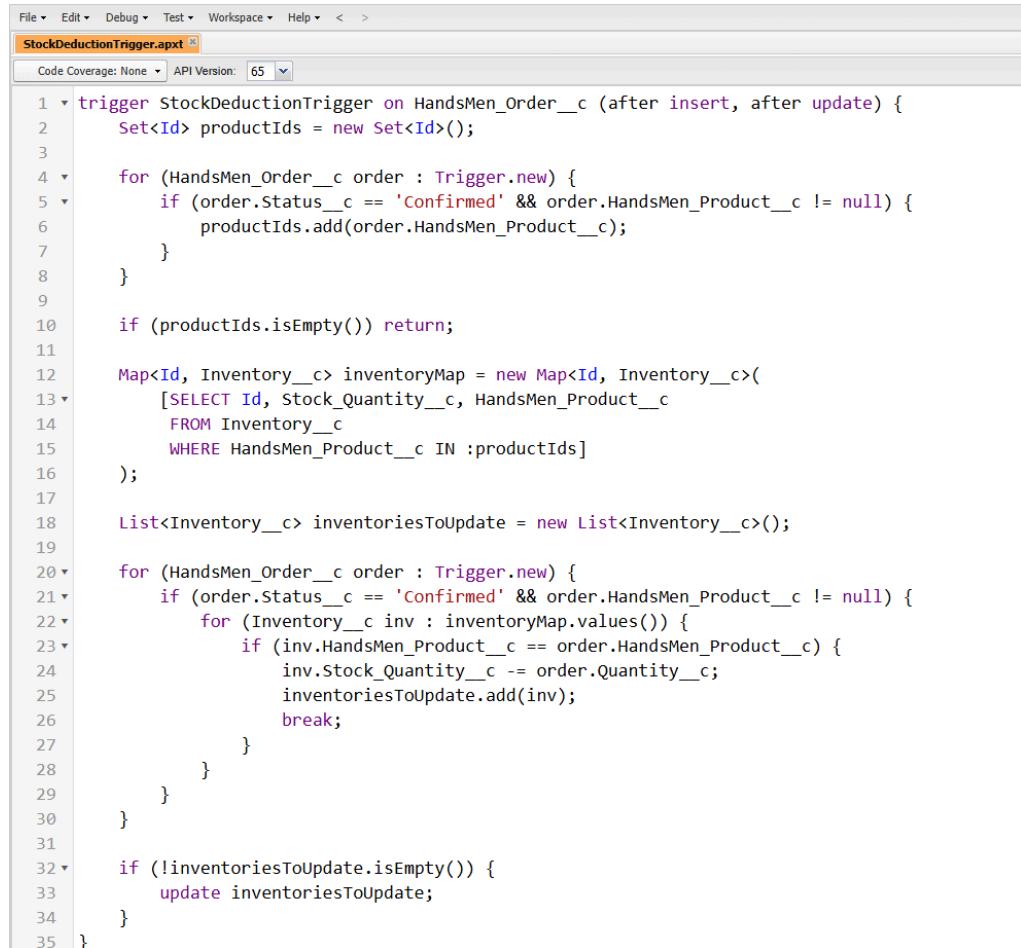
```

## 2) StockDeductionTrigger

- **Purpose:** Automatically reduces inventory stock when an order is confirmed, ensuring real-time stock accuracy.
- **How It Works:**
  1. **Collect Product IDs:** Scan incoming orders during insert or update; confirmed orders with selected products have their product IDs added to a set to identify relevant inventory records.
  2. **Retrieve Inventory Records:** Queries all corresponding Inventory\_\_c records and stores them in a map keyed by product ID for efficient access.
  3. **Deduct Stock Quantities:** Loops through confirmed orders, subtracts the order quantity from the inventory's

`Stock_Quantity__c`, and adds updated records to a list.

4. **Update Inventory:** Performs a single update operation for all modified inventory records to maintain accurate stock levels without manual intervention.



The screenshot shows the Salesforce IDE interface with the file `StockDeductionTrigger.apxt` open. The code is a Apex trigger named `StockDeductionTrigger` that runs on the `HandsMen_Order__c` object. It performs an update operation after insert or update. The trigger first collects product IDs from confirmed orders where the product is not null. Then, it queries the `Inventory__c` object to find matching products. For each found product, it decrements the stock quantity by the order quantity. Finally, it updates the found inventory records.

```
trigger StockDeductionTrigger on HandsMen_Order__c (after insert, after update) {
    Set<Id> productIds = new Set<Id>();
    for (HandsMen_Order__c order : Trigger.new) {
        if (order.Status__c == 'Confirmed' && order.HandsMen_Product__c != null) {
            productIds.add(order.HandsMen_Product__c);
        }
    }
    if (productIds.isEmpty()) return;
    Map<Id, Inventory__c> inventoryMap = new Map<Id, Inventory__c>(
        [SELECT Id, Stock_Quantity__c, HandsMen_Product__c
         FROM Inventory__c
         WHERE HandsMen_Product__c IN :productIds]
    );
    List<Inventory__c> inventoriesToUpdate = new List<Inventory__c>();
    for (HandsMen_Order__c order : Trigger.new) {
        if (order.Status__c == 'Confirmed' && order.HandsMen_Product__c != null) {
            for (Inventory__c inv : inventoryMap.values()) {
                if (inv.HandsMen_Product__c == order.HandsMen_Product__c) {
                    inv.Stock_Quantity__c -= order.Quantity__c;
                    inventoriesToUpdate.add(inv);
                    break;
                }
            }
        }
    }
    if (!inventoriesToUpdate.isEmpty()) {
        update inventoriesToUpdate;
    }
}
```

## v. User Roles & Profiles

A new profile, Platform1, was created by copying the Standard User profile and granted access to all necessary custom objects.

Roles were defined for different departments to control data visibility:

- Sales Manager
- Inventory Manager
- Marketing Team

SETUP		Profiles									
		Custom Object Permissions									
		Basic Actions					Object Permissions				
		Read	Create	Edit	Delete	View All Records	Modify All Records	View All Fields	Read	Create	Edit
HandleDen Customers		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
HandleDen Orders		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
HandleDen Products		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>					
		Inventory					✓	✓	✓	✓	✓
		Marketing Campaigns					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## vi. User Accounts

New users were created in Salesforce and assigned appropriate roles and profiles:

- Niklaus Mikaelson – Sales Manager
- Kol Mikaelson – Inventory Manager
- Lila Mikaelson – Marketing

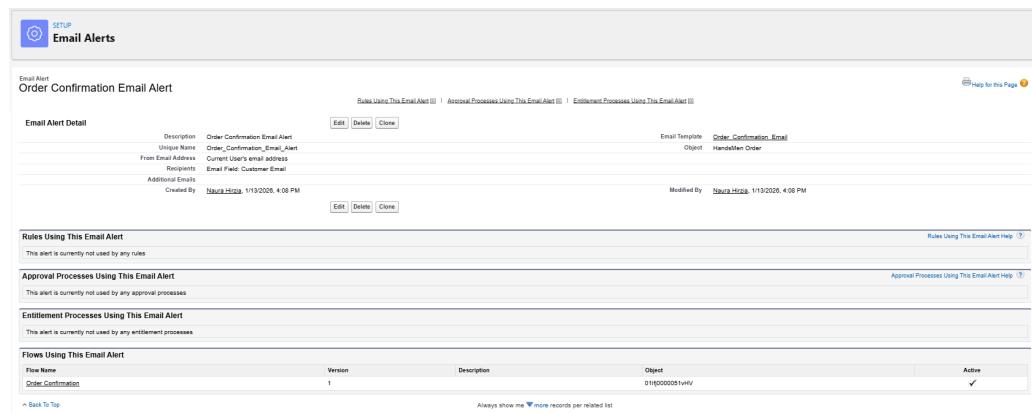
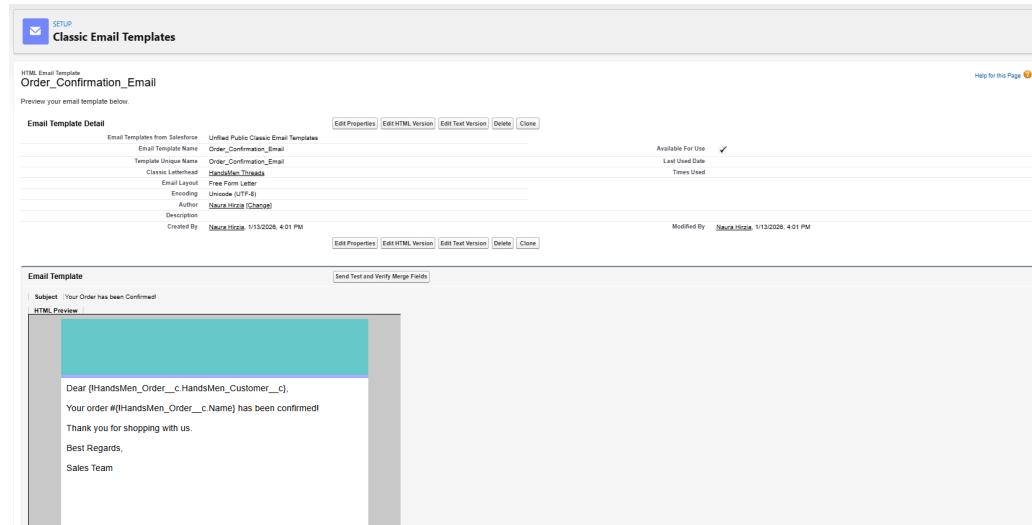
These role and profile configurations ensure that users can access only the data relevant to their responsibilities, maintaining system security and proper workflow.

SETUP		Users																																																																								
		All Users																																																																								
		Help for this Page ⓘ																																																																								
On this page you can create, view, and manage users. To get more licenses, use the Your Account app. <a href="#">Let's Go</a>																																																																										
View: <a href="#">All Users</a> <a href="#">Edit</a>   <a href="#">Create New User</a>																																																																										
<table border="1"> <thead> <tr> <th>Action</th> <th>Full Name</th> <th>Alias</th> <th>Username</th> <th>Role</th> <th>Active</th> <th>Profile</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>Charter_Erceld</td> <td>Charter</td> <td>charter.00100000000000000@chatter.salesforce.com</td> <td>System Administrator</td> <td><input checked="" type="checkbox"/></td> <td>Charter_Fire_User</td> </tr> <tr> <td><input type="checkbox"/></td> <td>ERIC_Grafham</td> <td>ERIC</td> <td>eric.140154efef000000@chatter.salesforce.com</td> <td>System Administrator</td> <td><input checked="" type="checkbox"/></td> <td>System Administrator</td> </tr> <tr> <td><input type="checkbox"/></td> <td>Hirka_Nura</td> <td>Nira</td> <td>hirka.nura.00113200004@chatter.salesforce.com</td> <td>System Administrator</td> <td><input checked="" type="checkbox"/></td> <td>System Administrator</td> </tr> <tr> <td><input type="checkbox"/></td> <td>Mikaelson_Kol</td> <td>kolka</td> <td>mikaelson.kolka.000@chatter.salesforce.com</td> <td>Inventory</td> <td><input checked="" type="checkbox"/></td> <td>Platform_1</td> </tr> <tr> <td><input type="checkbox"/></td> <td>Mikaelson_Lila</td> <td>lila</td> <td>mikaelson.lila.000@chatter.salesforce.com</td> <td>Sales</td> <td><input checked="" type="checkbox"/></td> <td>Platform_1</td> </tr> <tr> <td><input type="checkbox"/></td> <td>Mikaelson_Niklaus</td> <td>nikka</td> <td>mikaelson.niklaus.000@chatter.salesforce.com</td> <td>Sales</td> <td><input checked="" type="checkbox"/></td> <td>Platform_1</td> </tr> <tr> <td><input type="checkbox"/></td> <td>User_Innovation</td> <td>innov</td> <td>user.innovation.000@chatter.salesforce.com</td> <td>Marketing</td> <td><input checked="" type="checkbox"/></td> <td>Analytics_Cloud_Interaction_User</td> </tr> <tr> <td><input type="checkbox"/></td> <td>User_Search</td> <td>sear</td> <td>user.search.000@chatter.salesforce.com</td> <td>Marketing</td> <td><input checked="" type="checkbox"/></td> <td>Analytics_Cloud_Deputy_User</td> </tr> </tbody> </table>												Action	Full Name	Alias	Username	Role	Active	Profile	<input type="checkbox"/>	Charter_Erceld	Charter	charter.00100000000000000@chatter.salesforce.com	System Administrator	<input checked="" type="checkbox"/>	Charter_Fire_User	<input type="checkbox"/>	ERIC_Grafham	ERIC	eric.140154efef000000@chatter.salesforce.com	System Administrator	<input checked="" type="checkbox"/>	System Administrator	<input type="checkbox"/>	Hirka_Nura	Nira	hirka.nura.00113200004@chatter.salesforce.com	System Administrator	<input checked="" type="checkbox"/>	System Administrator	<input type="checkbox"/>	Mikaelson_Kol	kolka	mikaelson.kolka.000@chatter.salesforce.com	Inventory	<input checked="" type="checkbox"/>	Platform_1	<input type="checkbox"/>	Mikaelson_Lila	lila	mikaelson.lila.000@chatter.salesforce.com	Sales	<input checked="" type="checkbox"/>	Platform_1	<input type="checkbox"/>	Mikaelson_Niklaus	nikka	mikaelson.niklaus.000@chatter.salesforce.com	Sales	<input checked="" type="checkbox"/>	Platform_1	<input type="checkbox"/>	User_Innovation	innov	user.innovation.000@chatter.salesforce.com	Marketing	<input checked="" type="checkbox"/>	Analytics_Cloud_Interaction_User	<input type="checkbox"/>	User_Search	sear	user.search.000@chatter.salesforce.com	Marketing	<input checked="" type="checkbox"/>	Analytics_Cloud_Deputy_User
Action	Full Name	Alias	Username	Role	Active	Profile																																																																				
<input type="checkbox"/>	Charter_Erceld	Charter	charter.00100000000000000@chatter.salesforce.com	System Administrator	<input checked="" type="checkbox"/>	Charter_Fire_User																																																																				
<input type="checkbox"/>	ERIC_Grafham	ERIC	eric.140154efef000000@chatter.salesforce.com	System Administrator	<input checked="" type="checkbox"/>	System Administrator																																																																				
<input type="checkbox"/>	Hirka_Nura	Nira	hirka.nura.00113200004@chatter.salesforce.com	System Administrator	<input checked="" type="checkbox"/>	System Administrator																																																																				
<input type="checkbox"/>	Mikaelson_Kol	kolka	mikaelson.kolka.000@chatter.salesforce.com	Inventory	<input checked="" type="checkbox"/>	Platform_1																																																																				
<input type="checkbox"/>	Mikaelson_Lila	lila	mikaelson.lila.000@chatter.salesforce.com	Sales	<input checked="" type="checkbox"/>	Platform_1																																																																				
<input type="checkbox"/>	Mikaelson_Niklaus	nikka	mikaelson.niklaus.000@chatter.salesforce.com	Sales	<input checked="" type="checkbox"/>	Platform_1																																																																				
<input type="checkbox"/>	User_Innovation	innov	user.innovation.000@chatter.salesforce.com	Marketing	<input checked="" type="checkbox"/>	Analytics_Cloud_Interaction_User																																																																				
<input type="checkbox"/>	User_Search	sear	user.search.000@chatter.salesforce.com	Marketing	<input checked="" type="checkbox"/>	Analytics_Cloud_Deputy_User																																																																				

## vii. Email Templates & Alerts

- **Order Confirmation Email:** Sent automatically when an order status changes to Confirmed.
- **Low Stock Alert:** Sent when an inventory item's stock drops below 5 units.
- **Loyalty Program Update:** Sent when a customer's loyalty status is updated.

**Automation:** Email alerts were linked to the corresponding flows to ensure notifications are sent accurately and on time.

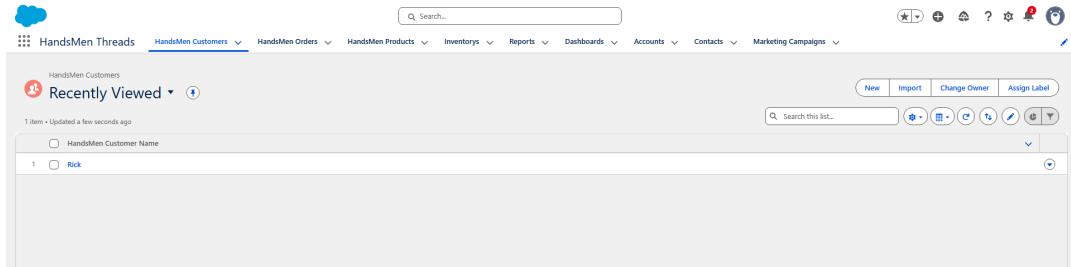


## c. UI/UX Development & Customization

A custom Lightning App named HandsMen Threads was created using Salesforce App Manager to centralize access to all CRM objects and features. The app was configured with a meaningful name and icon, while default colors and utility items were maintained.

Navigation items, including HandsMen Customer, HandsMen Order, Inventory, HandsMen Product, Reports, Dashboards, Account, Contact, and Marketing Campaign, were added to provide users with quick access to relevant data.

The System Administrator profile was granted access to the app, ensuring that authorized users could efficiently view and use its functionality, resulting in organized navigation and streamlined CRM operations.



## D. PROJECT EXPLANATION WITH REAL-WORLD EXAMPLE

### a. Customer Registration

A customer, Rika Sancheez visits the store

**Customer Name :** Rika Sancheez

**Email :** [rika332@delta.com](mailto:rika332@delta.com)

**Other Details :** Add phone number

**Validation Rule :** “Please fill Correct Gmail” (e.g., must contain [@gmail.com](mailto:@gmail.com))

Beth entered [naurahirzia@gmail.com](mailto:naurahirzia@gmail.com), which meets the Gmail validation requirement

### b. Product Setup

The admin adds product like Leather Jacket, Windbreaker, etc., into the Product\_\_c object.

Each product has a price and other details.

Inventory is also created to manage stock for the products.

### c. Order Placement

Rika decides to **buy 50 Wool Jacket (\$42 per piece)**

In Salesforce, a new order record is created

**Apex Trigger:** Automatically calculates Total\_Amount\_\_c = 50 x 42 = \$2100

#### d. Inventory Update

**Apex Trigger:** reduced Wool Jacket stock by 50 pieces

**Validation Rule:** Ensure stock never goes below 0

#### e. Loyalty Program

Rika now has a **total purchase of \$2100**

A trigger on Customer checks her total purchases

Based on the value:

<\$500 Bronze

\$500-\$1000 Silver

\$1000 Gold

**Apex Trigger:** Evaluates cumulative purchases and updates Loyalty\_Status\_\_c field

So, Beth becomes a **Gold member**

#### f. Email Notification

When a new order is placed or royalty status is updated:

**Flow + Email Alerts is triggered**

## E. SCREENSHOTS

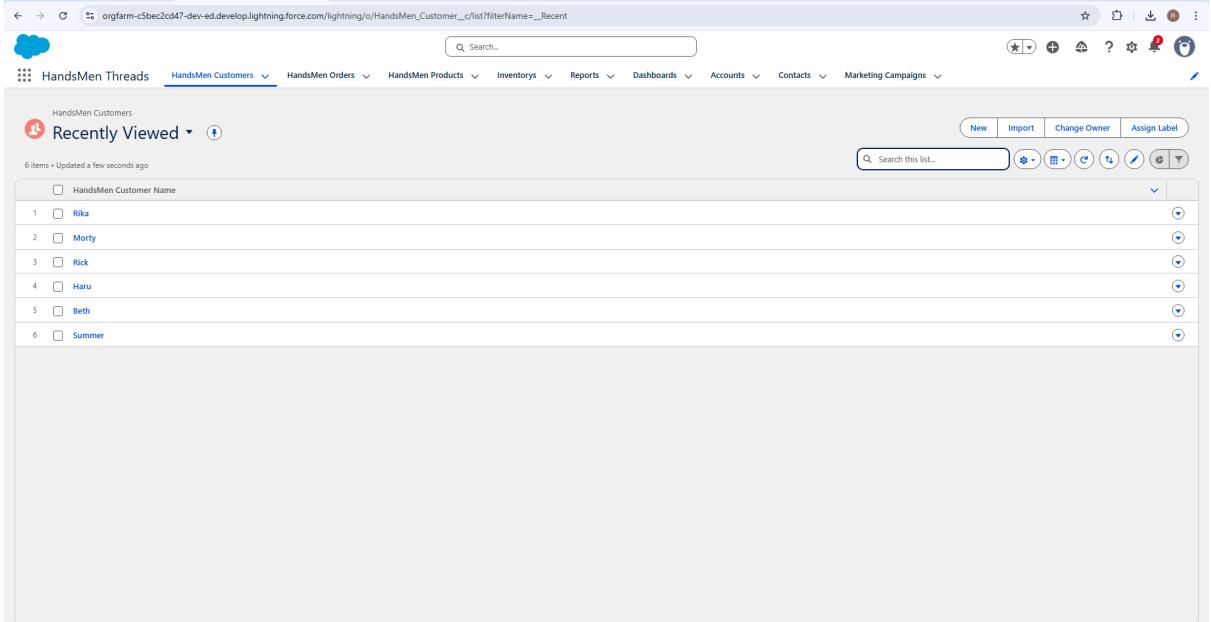


Fig: Custom App for HandsMen Threads

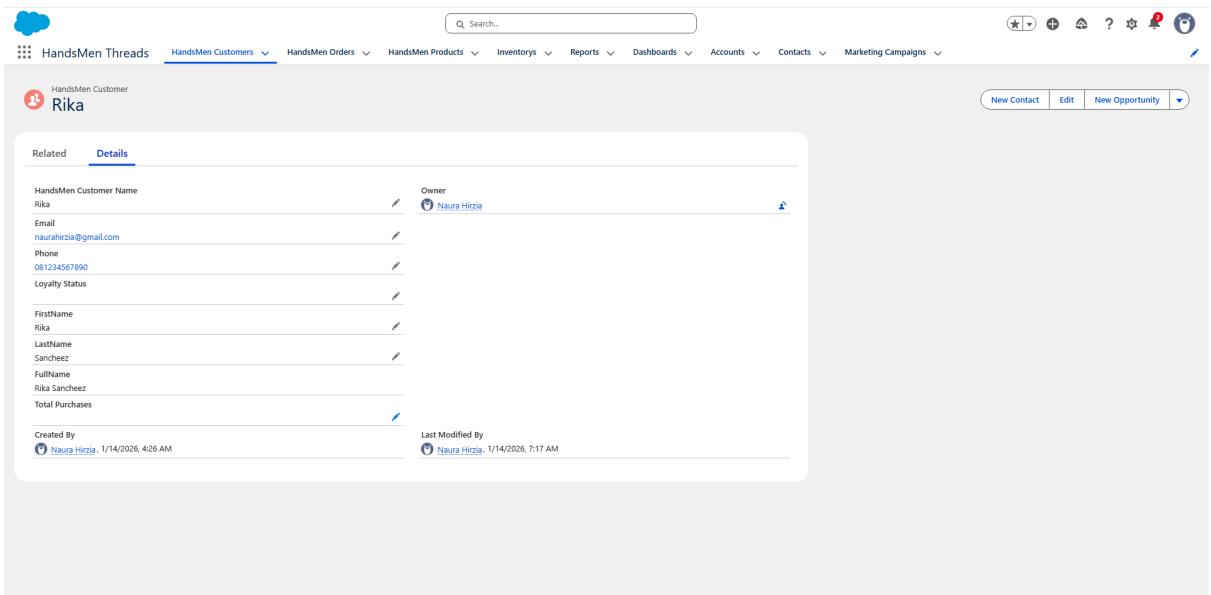
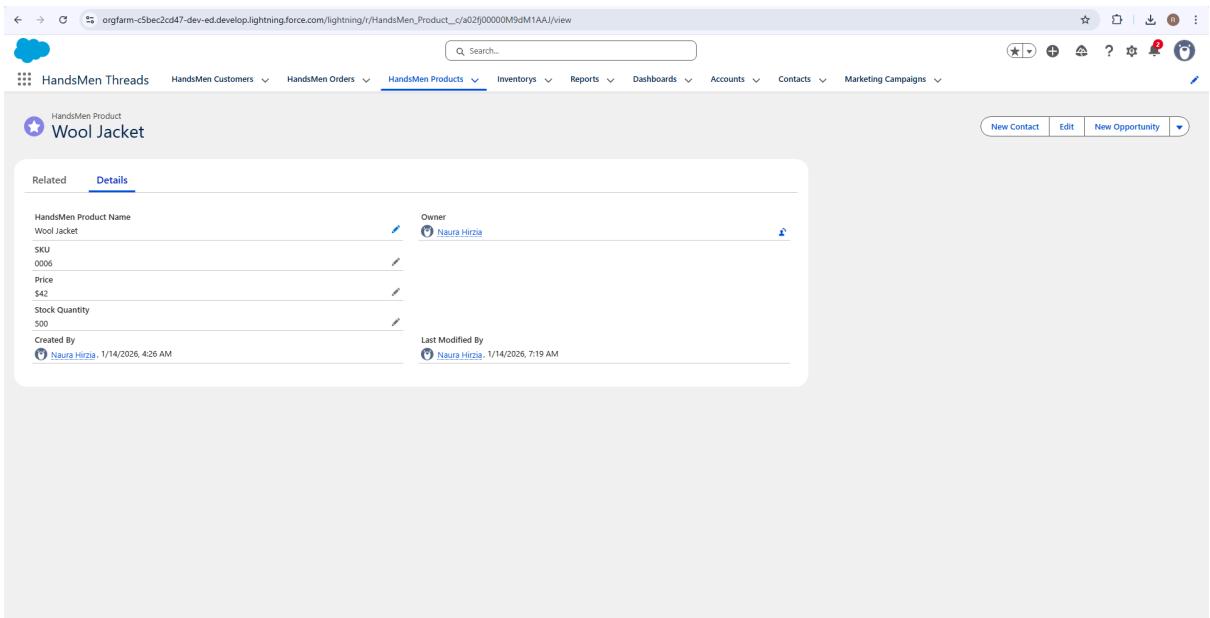
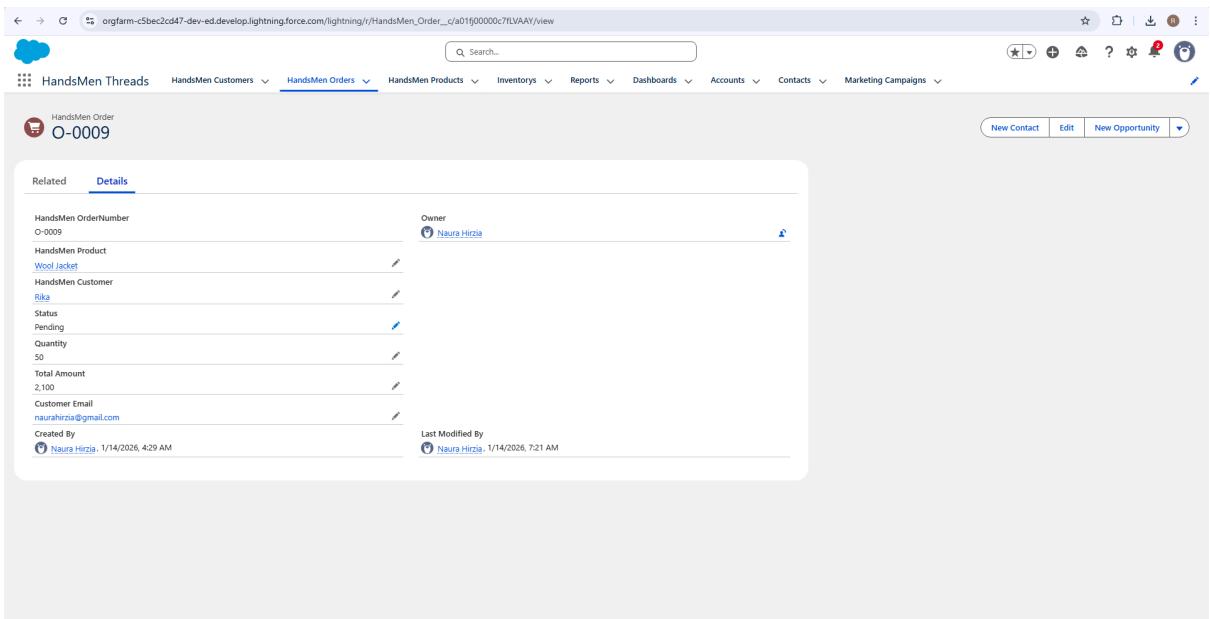


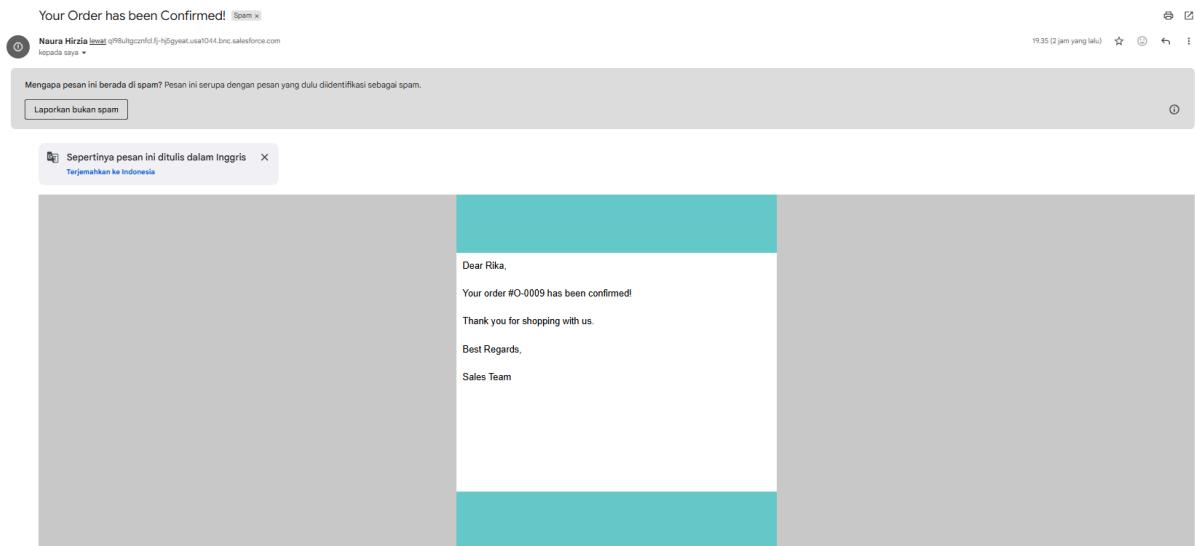
Fig: Customer Creation in HandsMen Threads



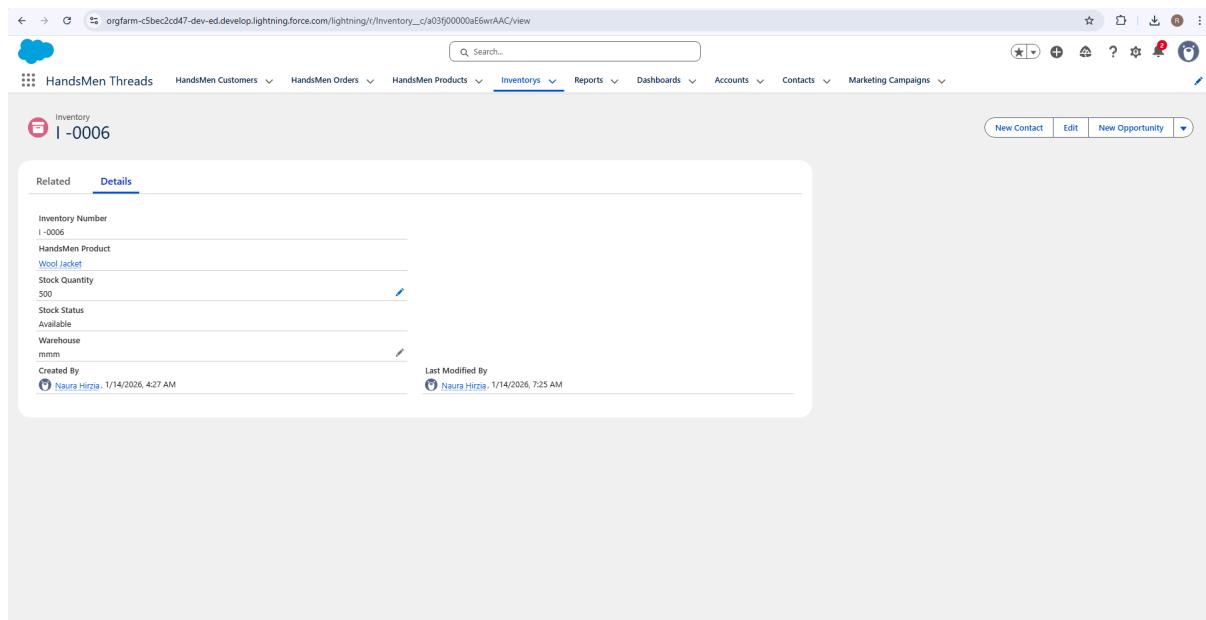
**Fig: Products in HandsMen Threads**



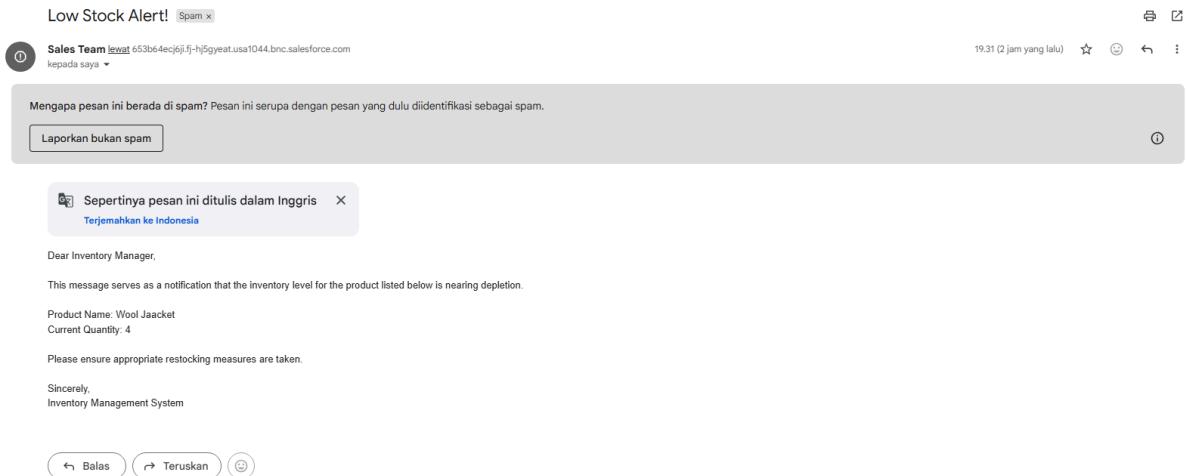
**Fig: Order Confirmation**



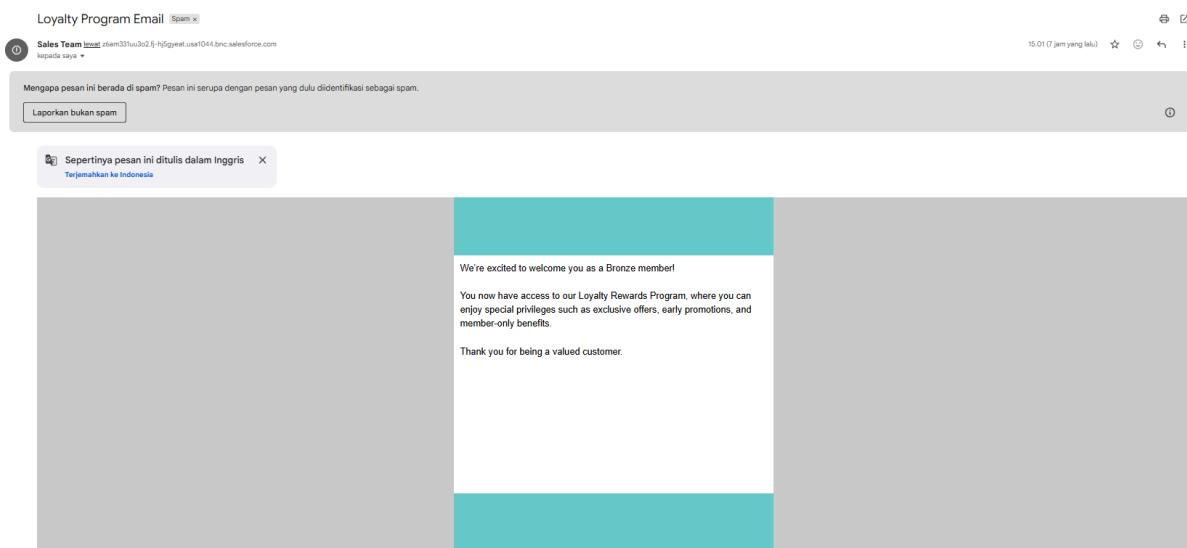
**Fig: Order Confirmation Email**



**Fig: Inventory Creation**



**Fig: Low Stock Alert Email**



**Fig: Loyalty Program Email**

## F. CONCLUSION

The HandsMen Threads Salesforce CRM project delivered a customized system that centralizes customer, product, order, inventory, and marketing data while automating key business processes. Using structured data models, validation rules, and automations such as Flows, Apex Triggers, and Email

Alerts, the system ensures accurate order processing, timely customer communication, and consistent inventory and loyalty updates.

The CRM also strengthens operational efficiency through role-based access controls, secure data management, and centralized reporting. Overall, the solution provides a scalable and reliable platform that enhances decision-making, improves customer engagement, and supports the long-term growth of HandsMen Threads.

## **G. FUTURE SCOPE**

### **a. AI Chatbots and Customer Self-Service**

Deploy intelligent chatbots and self-service portals to assist customers with inquiries, order tracking, and product recommendations, enhancing customer convenience and support efficiency.

### **b. Supplier and Procurement Integration**

Connect the CRM with supplier systems to automate restocking alerts, purchase orders, and supply chain visibility.

### **c. E-commerce Platform Integration**

Integrate the CRM with online sales platforms to enable real-time order synchronization, automated inventory updates, and seamless customer purchase tracking.

### **d. Mobile CRM Enablement**

Introduce mobile-optimized features that allow sales and inventory teams to manage orders, track stock levels, and access customer information in real time from any location.

### **e. Advanced Marketing Automation**

Strengthen marketing capabilities through automated email campaigns, social media integration, customer segmentation, and performance analytics to improve campaign effectiveness.

**f. Customer Chat and Messaging Integration**

Integrate chat-based communication channels such as Salesforce Messaging, live chat, or third-party platforms (e.g., WhatsApp) to enable real-time customer support, order updates, and direct engagement within the CRM system.