

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

## **ABSTRACT**

This project presents the implementation of a customized Salesforce CRM solution for **HandsMen Threads**, a premium men's fashion and tailoring brand. The objective was to streamline business operations, enhance customer engagement, and maintain data integrity across departments.

The solution involves designing a robust data model featuring five key custom objects: Customer, Order, Product, Inventory, and Marketing Campaign. Business processes were automated using Record-Triggered Flows, Scheduled Flows, Email Alerts, and Apex to handle order confirmations, loyalty status updates, and proactive stock alerts.

To ensure clean and reliable data, validation rules were established, and a role-based security model was implemented for the Sales, Inventory, and Marketing teams. The solution also includes a scheduled batch job using Apex to update low stock quantities.

This end-to-end CRM implementation improves customer experience through personalized communication, ensures operational efficiency with automation, and lays a scalable foundation for future business growth using the Salesforce Platform.

## **OBJECTIVE**

The main objective of this project is to develop and implement a customized Salesforce CRM solution for HandsMen Threads to streamline core business operations, maintain data integrity, and enhance customer satisfaction.

By building a centralized system to manage customers, orders, products, inventory, and marketing campaigns, the project aims to:

- Automate key processes such as order confirmations, loyalty status updates, and stock alerts.
- Ensure accurate and consistent data entry using validation rules.
- Enable real-time visibility of inventory and customer interactions.
- Improve internal team coordination through role-based access control.
- Deliver personalized customer experiences through targeted communication and loyalty programs.

## **TECHNOLOGY DESCRIPTION**

### **Salesforce:-**

Salesforce is a cloud-based Customer Relationship Management (CRM) platform that helps businesses manage customer data, automate processes, and improve service, marketing, and sales operations. It provides point-and-click tools as well as programmatic capabilities (like Apex and Flows) to build custom business solutions.

### **Custom Objects:-**

Objects in Salesforce are like tables in a database. Custom Objects are created to store specific data.

Example:

- Customer\_\_c – Stores customer info
- Product\_\_c – Stores product details
- Order\_\_c – Stores orders

### **Tabs:-**

Tabs are used to display object data in the Salesforce UI.

Example: A tab for Product\_\_c allows users to easily view and manage products.

### **Custom App:-**

An App in Salesforce is a collection of tabs grouped together for a specific business purpose.

### **Profiles:-**

Profiles define what a user can see, do, and edit in Salesforce. It controls object permissions, field access, and more.

### **Roles:-**

Roles control the data visibility in Salesforce's role hierarchy. It's used for sharing settings and reporting.

### **Permission Sets:-**

Permission Sets grant additional permissions to users without changing their profile.

## **Validation Rules:-**

Validation Rules ensure data entered meets business criteria.

Example:

- Email must contain @gmail.com
- Stock cannot be negative

## **Email Templates:-**

Predefined formats for sending emails to customers or users.

Example:

- "Order Confirmation" template

## **Email Alerts:-**

Email Alerts are actions in Flows or Workflow Rules that send emails using predefined templates.

Example: When a loyalty level changes, an email is sent to the customer.

## **Flows:-**

Flows automate business logic without code. They can create, update, or send notifications.

Example:

- Flow triggers email alerts on new order

## **Apex:-**

Apex is Salesforce's object-oriented programming language. It allows developers to write custom logic.


Example Triggers:

- Update Total\_Amount\_\_c in orders
- Reduce inventory stock

## **DETAILED EXECUTION OF PROJECT PHASES**

### **1. Developer Org Setup**

- A Salesforce Developer Org was created using <https://developer.salesforce.com/signup>.
- The account was verified, password set, and access was granted to the Salesforce Setup page.



## Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud.

Sign up for your Developer Edition.

- ✓ Build apps fast with drag-and-drop tools
- ✓ Go further with Apex code
- ✓ Build AI agents with Agentforce
- ✓ Harmonize your data with Data Cloud
- ✓ Ground Agentforce with structured and unstructured data
- ✓ Integrate with anything using APIs

### Sign up for your Developer Edition

A free Salesforce Platform environment with Agentforce and Data Cloud

First name  Last name

Job title  Work email

Company  Country/Region

Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud infrastructure.

☐ I agree to the [Main Services Agreement – Developer Services](#) and [Salesforce Program Agreement](#). I acknowledge, as described in the [Developer Documentation](#): (1) the Developer Edition includes autonomous and other generative AI features; and (2) Salesforce may limit use of those features and the org, and may terminate any org that has been inactive for 45 days.

We value your privacy. To learn more, visit our [Privacy Statement](#).

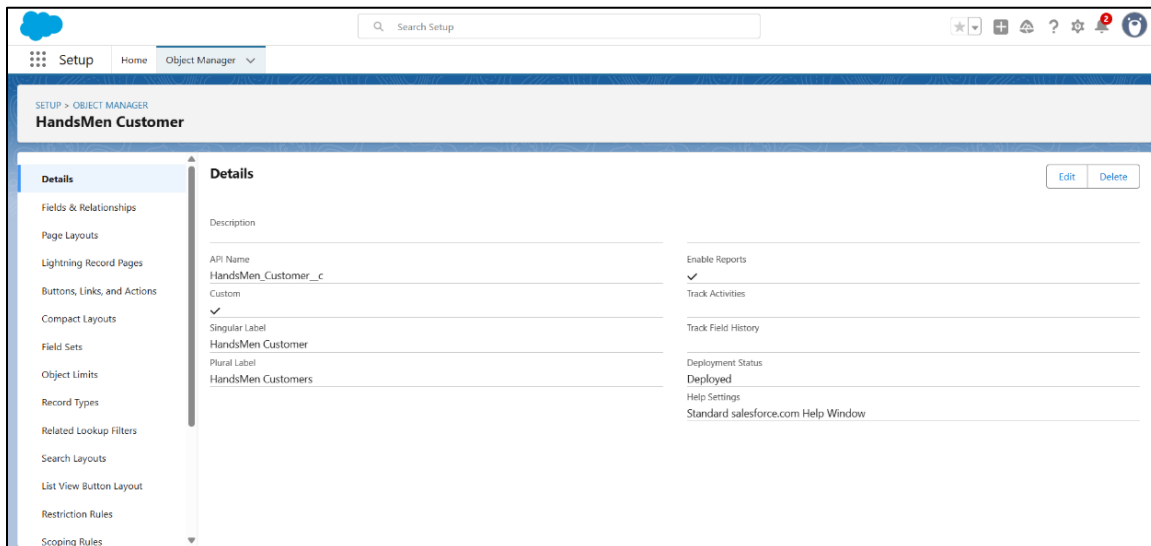
## 2. Custom Object Creation

Five custom objects were created to store business-critical data:

- **HandsMen Customer** – Stores customer info like email, phone, loyalty status.
- **HandsMen Product** – Stores product catalog details like SKU, price, and stock.
- **HandsMen Order** – Stores orders placed by customers, including quantity and status.
- **Inventory** – Tracks stock quantity and warehouse location.
- **Marketing Campaign** – Stores promotional campaigns and scheduling.

Steps followed:

- Navigated to Setup → Object Manager → Create → Custom Object
- Provided label, name, and enabled reports/search
- Saved and created Tabs for each object



The screenshot shows the Salesforce Setup interface for creating a custom object. The breadcrumb trail is: SETUP > OBJECT MANAGER. The object being configured is "HandsMen Customer".

**Details**

Description:

API Name: HandsMen\_Customer\_\_c

Custom: ☒

Singular Label: HandsMen Customer

Plural Label: HandsMen Customers

Enable Reports: ☒

Track Activities: ☒

Track Field History: ☐

Deployment Status: Deployed

Help Settings: Standard salesforce.com Help Window

Buttons: Edit, Delete

### 3. Creating the Lightning App

- A custom Lightning App named HandsMen Threads was created.
- Included tabs: HandsMen Customer, Order, Product, Inventory, Campaign, Reports, etc.
- Assigned to the System Administrator profile.

### 4. Validation Rules

To ensure accurate data entry and enforce business logic, the following validation rules were applied:

- Order Object: Prevents saving if Total\_Amount\_\_c <= 0.  
Error: "Please Enter Correct Amount"
- Customer Object: Validates email contains @gmail.com.  
Error: "Please fill Correct Gmail"

### 5. User Role & Profile Setup

- Cloned the Standard User profile to a new profile named Platform 1 and added access to necessary custom objects.
- Created roles for different departments:
  - Sales Manager, Inventory Manager, Marketing Team

### 6. User Creation

Users were created in Salesforce and assigned appropriate roles and profiles to reflect their responsibilities:

- Niklaus Mikaelson - Assigned the Sales role
- Kol Mikaelson - Assigned the Inventory role
- These role-based assignments help enforce proper data access and process control within the system.

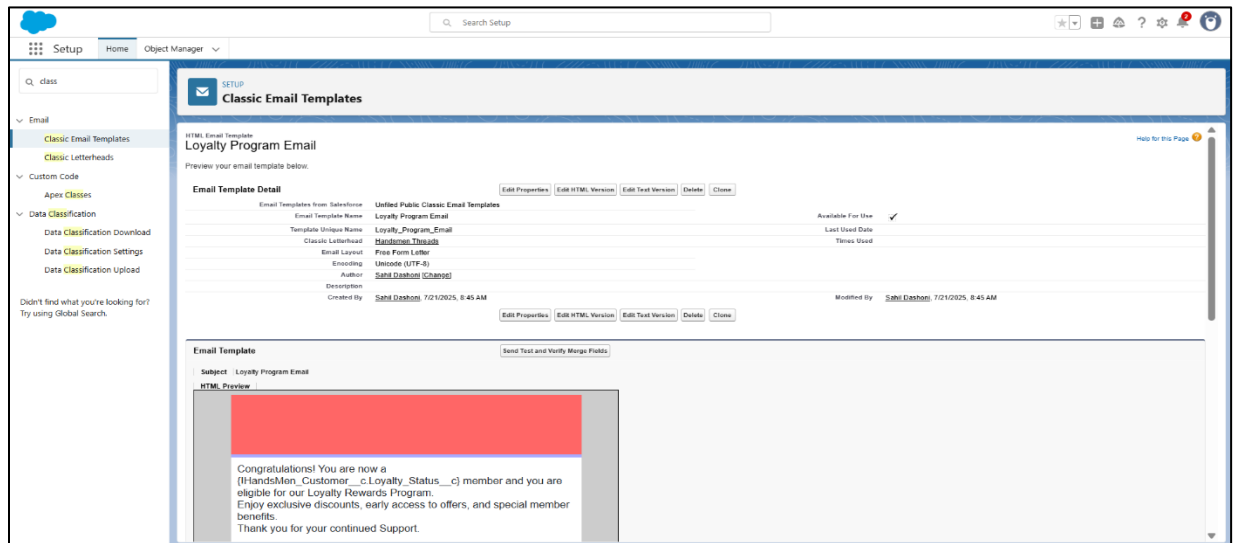
The screenshot displays the Salesforce Setup interface for editing a user. The left sidebar shows the navigation menu with 'Users' selected. The main content area is titled 'User Edit: Niklaus Mikaelson' and includes a 'User Edit' section with 'Save', 'Save & New', and 'Cancel' buttons. Below this is the 'General Information' section, which is divided into two columns. The left column contains fields for First Name (Niklaus), Last Name (Mikaelson), Alias (nmika), Email (sahildashon453@gmail.com), Username (sahildashon2107@gmail.co), Nickname (User175310908894021374), Title, Company, Department, and Division. The right column contains fields for Role (Sales), User License (Salesforce), Profile (Platform 1), Active (checked), Marketing User, Offline User, Knowledge User, Flow User, Service Cloud User, Site.com Contributor User, Site.com Publisher User, WDC User, Data.com User Type (None), Data.com Monthly Addition Limit (300), and Accessibility Mode (Classic Only). A red asterisk indicates required information.

## 7. Email Template & Alerts

Created three email templates:

- Order Confirmation – Sent on order status = Confirmed
- Low Stock Alert – Sent when Inventory < 5 units
- Loyalty Program Email – Sent when loyalty status changes

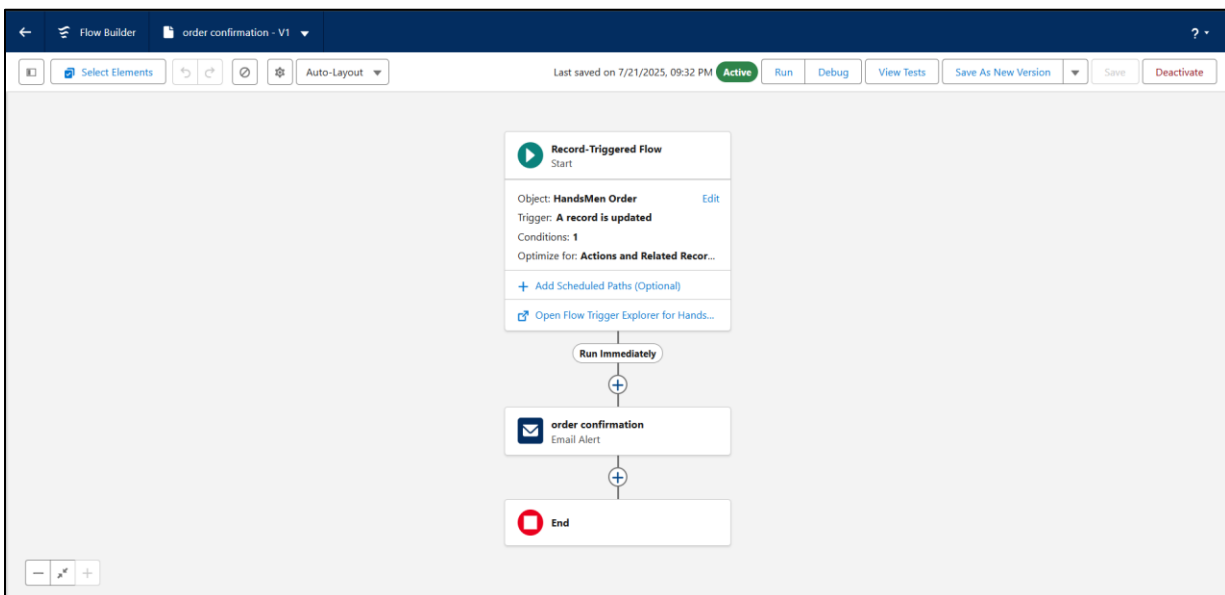
Corresponding Email Alerts were created using these templates and linked to automation flows.



## 8. Flow Implementations

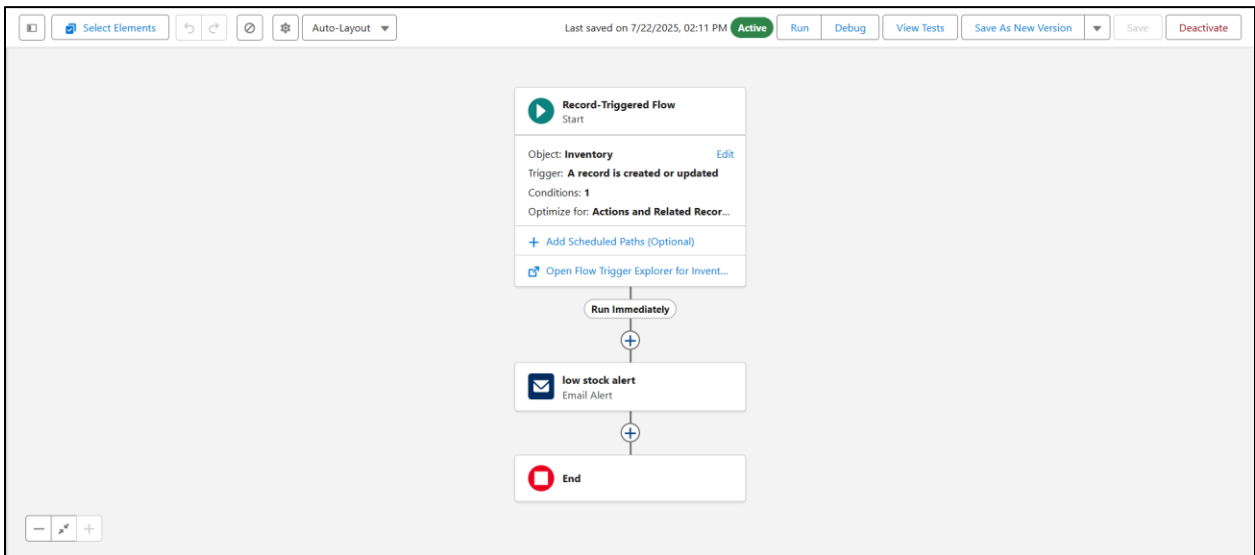
### a. Order Confirmation Flow

- Triggered when an order is updated to Confirmed.
- Sends an Order Confirmation email to the related customer.



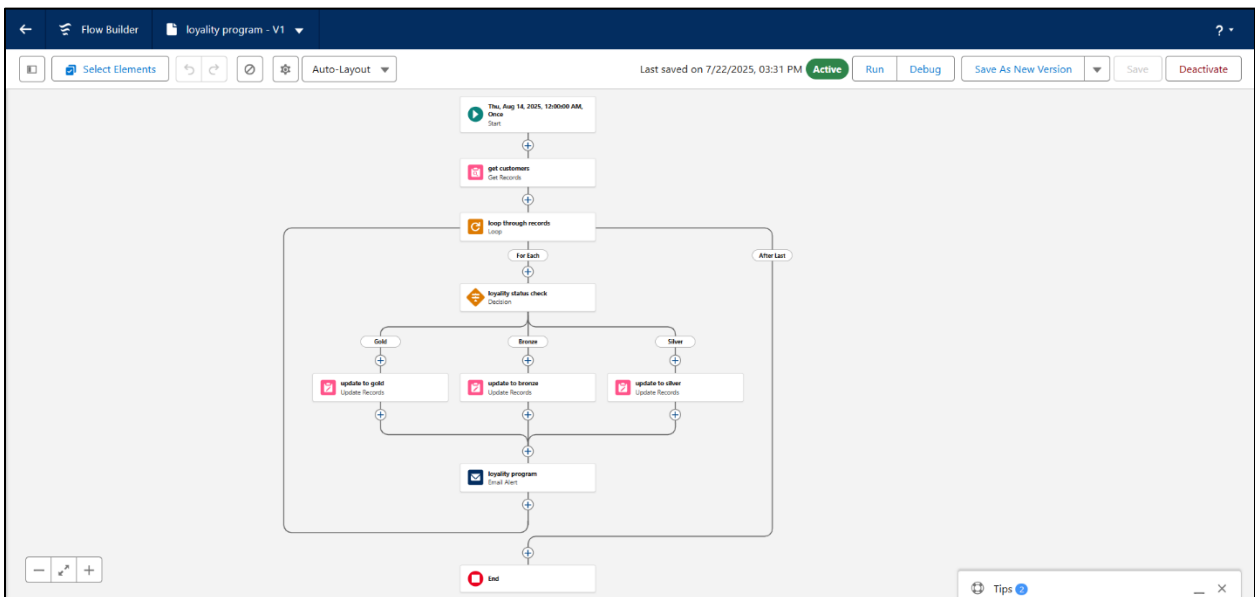
## b. Stock Alert Flow

- Triggered when Inventory stock drops below 5.
- Sends Low Stock email to Inventory Manager.



## c. Scheduled Flow: Loyalty Update

- Runs daily at midnight.
- Loops through customers and updates their Loyalty Status based on total purchases.



## 7. Apex Triggers

- **Order Total Trigger:** Auto-calculates Total Amount based on quantity and unit price.
- **Stock Deduction Trigger:** Reduces stock when an order is placed.
- **Loyalty Status Trigger:** Updates Loyalty Status based on total purchases.

# **PROJECT EXPLANATION WITH REAL-WORLD EXAMPLE**

Let's walk through it like a real customer interaction.

## **1. Customer Registration**

- A customer, Elijah Mikaelson, visits the store or website.
- In Salesforce: A record is created in the Customer object with his name, phone, email, etc.
- Validation Rule: Ensures the email is valid (e.g., must contain @gmail.com).

## **2. Product Setup**

- The admin adds products like Shirts, Jeans, etc., into the Product\_\_c object.
- Each product has a price and other details.
- Inventory is also created to manage stock for these products.

## **3. Order Placement**

- Elijah decides to buy 2 shirts (each ₹500). An order is placed.
- In Salesforce: A new Order record is created.
- Apex Trigger: Automatically calculates  $\text{Total\_Amount\_c} = 2 * 500 = ₹1000$ .

## **4. Inventory Update**

As soon as the order is placed:

- Apex Trigger on Inventory: Reduces shirt stock by 2.
- Validation Rule: Ensures stock never goes below 0.

## **5. Loyalty Program**

- Elijah now has a total purchase of ₹1000.
- A trigger on Customer checks his total purchases.

Based on the value:

< ₹500 → Bronze

₹500–₹1000 → Silver

₹1000 → Gold

- So, Elijah becomes a Silver member.

## **6. Email Notifications**

- When a new order is placed or loyalty status is updated:
- Flow + Email Alert is triggered.
- Elijah gets an email:  
“Thanks for your purchase! Your loyalty status is now Silver.”

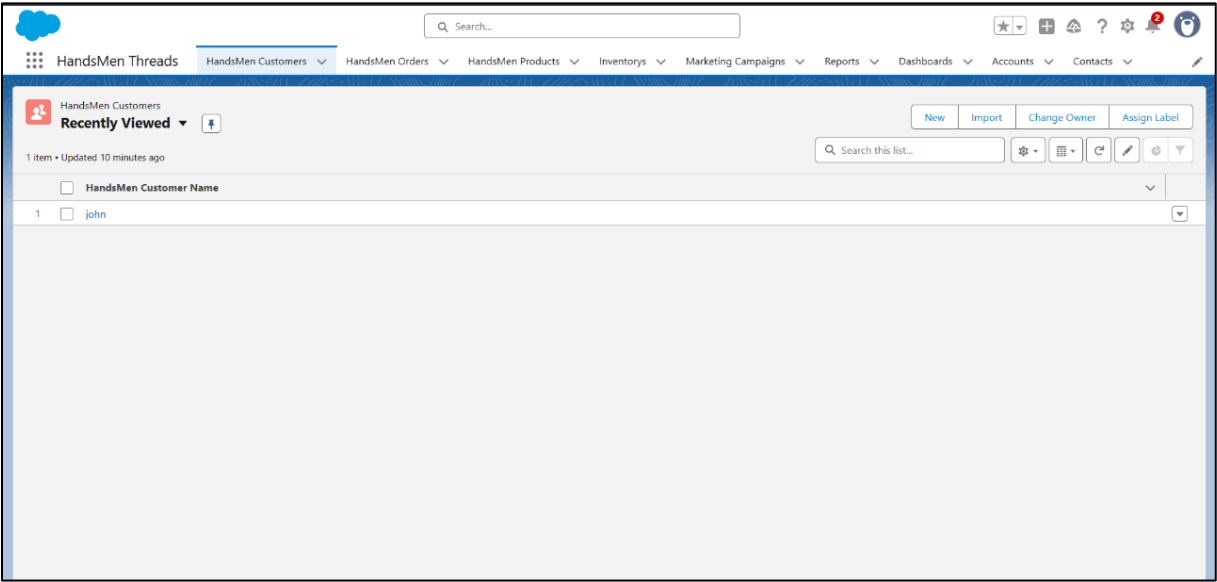
## **7. Users and Roles**

Salesforce users like store staff are created:

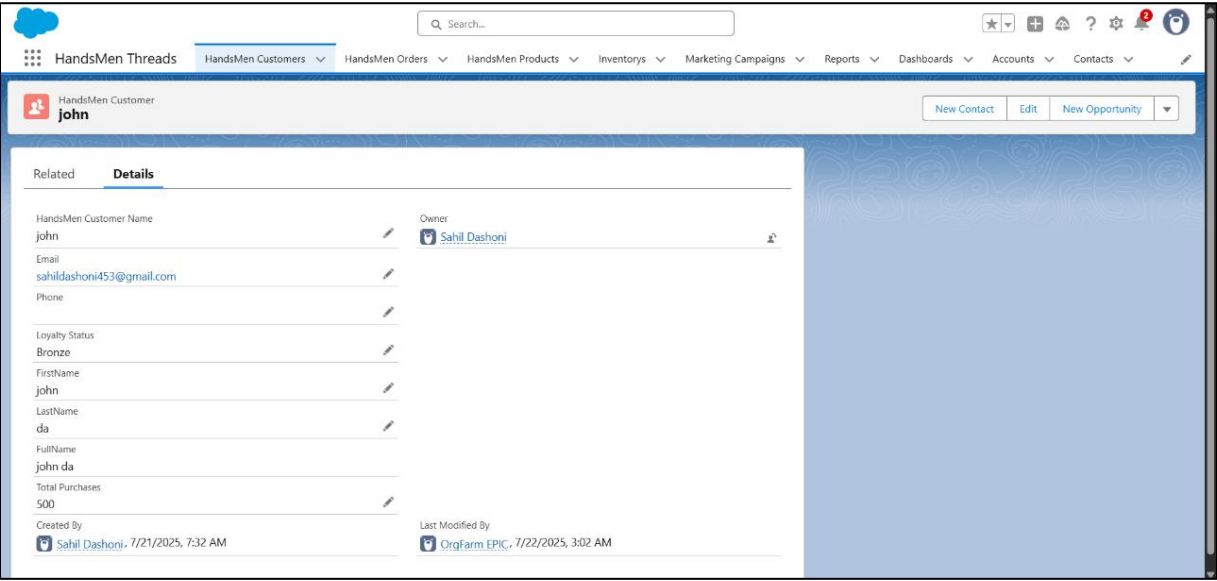
- Niklaus Mikaelson – Sales Role (Platform 1 Profile)
- Kol Mikaelson – Inventory Role (Platform 1 Profile)



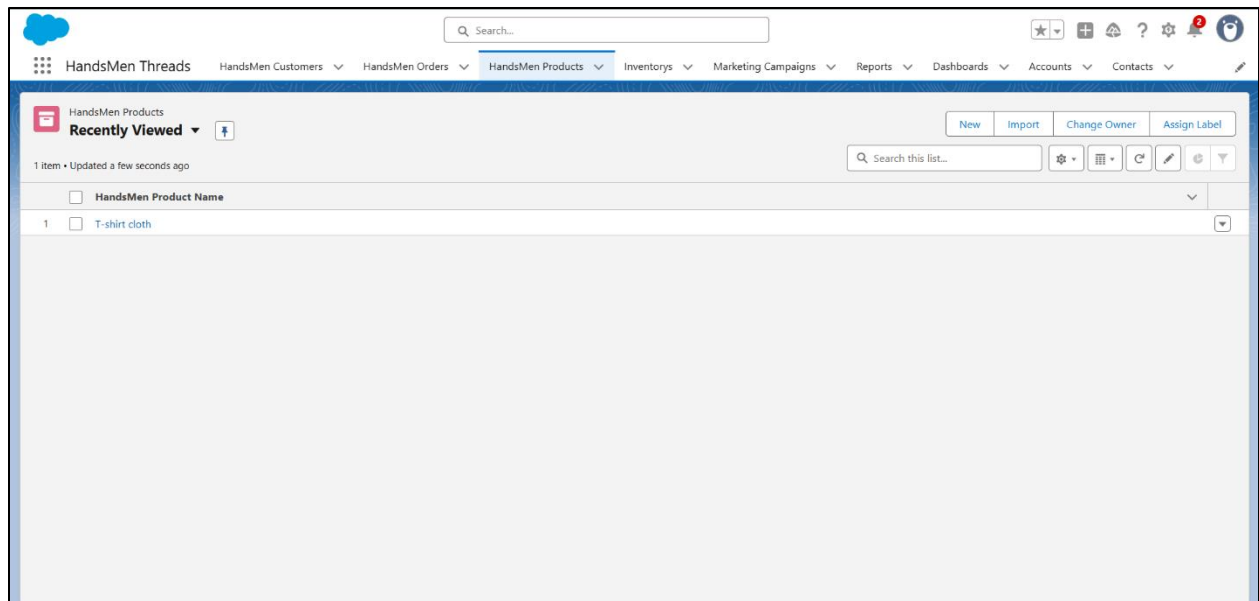
**SCREENSHOTS**



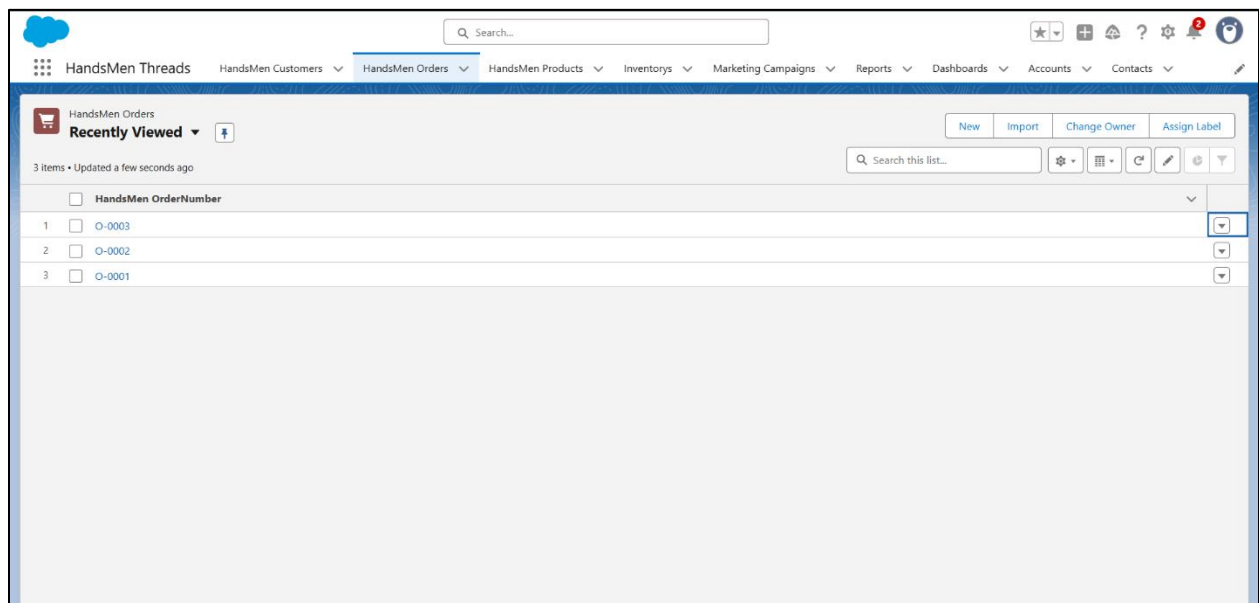
**Fig: Custom App for HandsMen Threads**



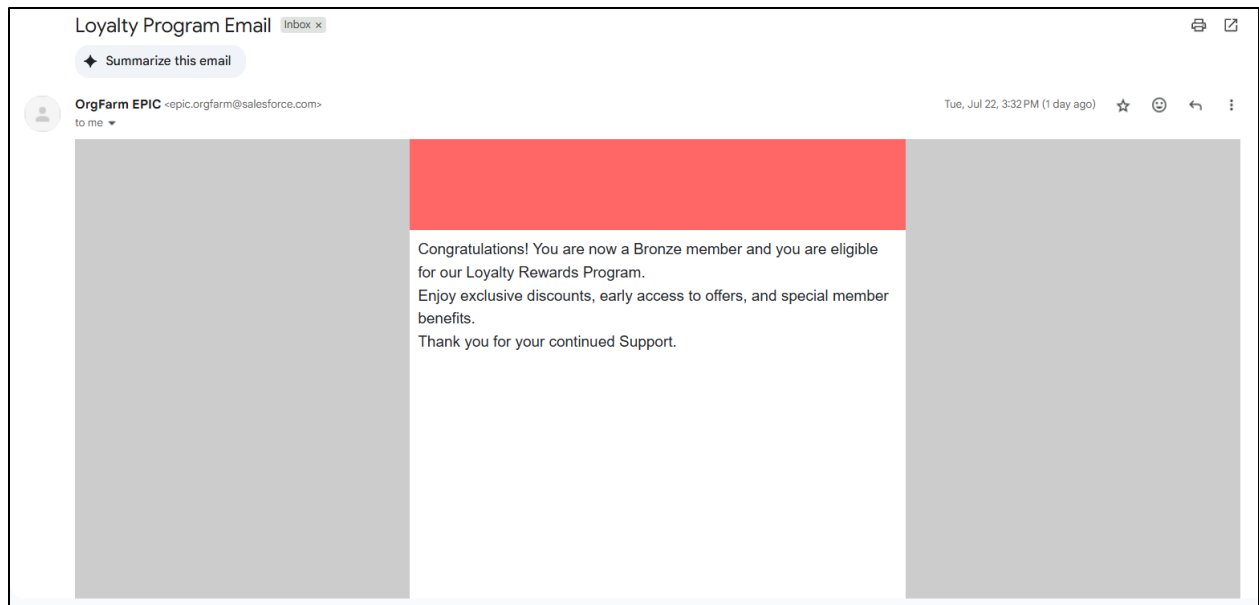
**Fig: Customer Creation in HandsMen Threads**



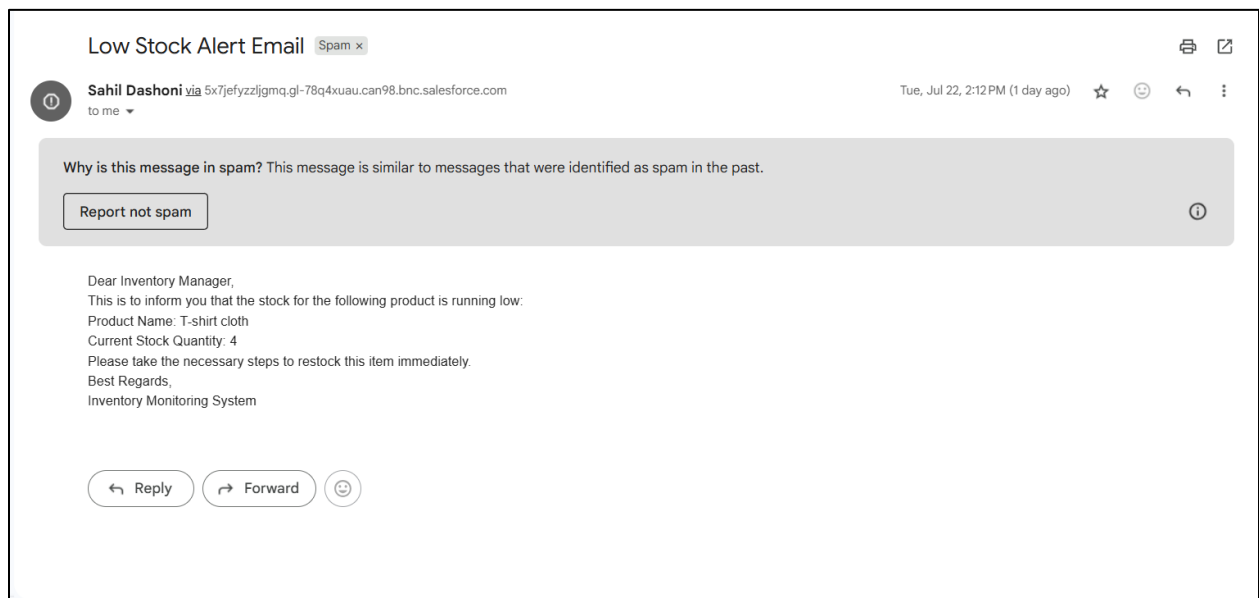
**Fig: Products in HandsMen Threads**



**Fig: Order Confirmation**



**Fig: Loyalty Program Email**



**Fig: Low Stock Alert Email**

## **CONCLUSION**

The HandsMen Threads CRM system built on Salesforce successfully streamlines key business processes like customer management, product cataloging, order processing, inventory tracking, and loyalty program automation. By leveraging Salesforce tools like Custom Objects, Flows, Validation Rules, Email Alerts, and Apex, the system ensures accurate data entry, real-time updates, and enhanced customer experience. Through automation and well-structured user roles, the platform minimizes manual errors, speeds up operations, and provides better insights into sales and stock.

### **Future Scope:-**

#### **1. Customer Portal Integration**

- Build a Customer Community Portal where customers can log in, view orders, and track loyalty status.

#### **2. Mobile App using Salesforce Mobile SDK**

- Enable store staff to manage inventory and orders on the go using a mobile interface.

#### **3. Reports & Dashboards**

- Create detailed sales and inventory dashboards for management to monitor trends and performance in real-time.

#### **4. AI-Powered Recommendations (Einstein)**

- Use Salesforce Einstein to provide personalized product suggestions based on past purchases.

#### **5. WhatsApp/SMS Integration**

- Notify customers via WhatsApp or SMS about order confirmations and loyalty updates.