

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

Project Overview

This initiative developed a custom Salesforce CRM for HandsMen Threads, a modern men's fashion company, specifically designed to drive operational efficiency and customer loyalty. Key features include a robust data model with custom objects for Customers, Orders, Products, Inventory, and Campaigns, ensuring unified data management. The system automates core business processes (confirmations, loyalty updates, low-stock alerts) using Flows, Email Alerts, and Apex, while enforcing data integrity through strict validation rules and UI validations. Role-based access for Sales, Inventory, and Marketing teams, combined with a scheduled stock monitoring batch, ensures a secure, scalable, and highly optimized platform that significantly enhances customer engagement.

Objectives

This Salesforce CRM implementation for HandsMen Threads creates a unified operational hub, centralizing sales, inventory, marketing, and customer service data. By employing stringent validation rules and role-based access, the platform ensures data integrity and supports informed decision-making. Key processes are automated including order confirmations, loyalty program updates, and proactive stock monitoring to enhance efficiency and agility. The system focuses on driving long-term loyalty through personalized customer communications, ultimately delivering a scalable foundation that optimizes workflows, tightens inventory control, and positions HandsMen Threads for measurable and sustained growth.

Phase 1: Requirement Analysis & Planning

1.1 Understanding Business Requirements

To address fragmented data, manual workflows, and poor cross-team visibility, HandsMen Threads requires a **centralized Salesforce CRM**. This solution is essential for unifying sales, inventory, marketing, and customer service data. The platform's core objective is to **boost customer engagement, streamline operational workflows,**

and provide the scalable foundation necessary for sustained growth in the competitive men's fashion sector.

1.2 Defining Project Scope and Objectives

Scope

- Configuring the Salesforce platform to meet unique business requirements.
- Integrating key data entities, specifically the Customer, Order, Product, Inventory, and Marketing Campaign objects.
- Developing process automation (Flows and Apex) for critical tasks, including order confirmations, loyalty status updates, and inventory notifications.

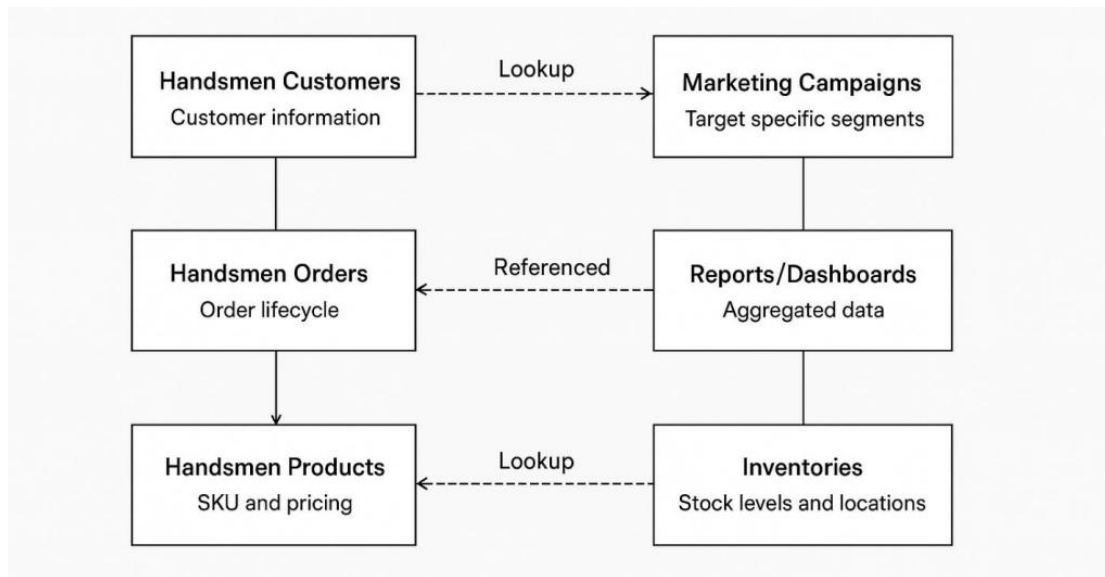
Objectives

- Ensure high data accuracy and consistency through the implementation of rigorous validation rules and UI controls.
- Establish secure, role-based access controls tailored for the Sales, Inventory, and Marketing user teams.
- Enhance the overall customer experience by enabling personalized and timely communications.
- Provide HandsMen Threads with a future-ready, scalable operational platform capable of supporting continuous innovation and growth.

1.3 Design Data Model and Security Model

Data Model

Entities are HandsMen Customer, HandsMen Order, HandsMen Product, Inventory, Marketing Campaign.



Security Model:

- Role-based access controls specifically for Sales, Inventory, and Marketing teams.
- validation rules throughout the system to strictly enforce data accuracy and integrity.
- Scheduled Apex batch job for proactive and automated monitoring of low stock levels.

1.4 Stakeholders Mapping

Internal Stakeholders:

- Sales Team - Focused on customer acquisition and relationship management.
- Inventory Team - Manages stock monitoring and timely replenishment.
- Marketing Team - Executes campaigns and oversees loyalty programs.

External Stakeholders:

- Customers - improved service and personalized engagement.
- Suppliers - Benefit from streamlined inventory coordination.
- Management - Uses reliable data for strategic decision-making.

1.5 Execution Roadmap

Week 1-2: Discovery and Design

Week 3-4: Core Development

Week 5-6: User Interface & QA

Week 7-8: Deployment and Go-Live

Phase 2:Salesforce Development - Backend & Configurations

2.1 Setup environment & DevOps workflow

A **Salesforce Developer Organization** was created and configured through the signup portal, followed by email verification and password establishment, granting successful access 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: Nicole Jheseney ✓ Last name: Marquez ✓

Job title: Developer ✓ Work email: nicolejheseney@gmail.com ✓

Company: Polytechnic University ✓ Country/Region: Philippines ✓

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](#).

☒ I'm not a robot
reCAPTCHA is changing its terms of service. [Take action.](#)

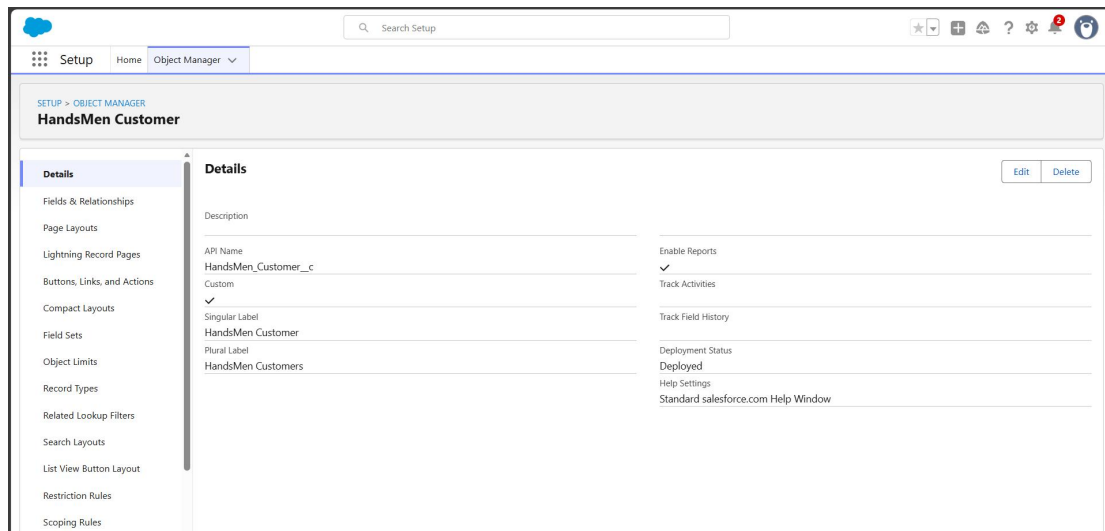
2.2 Customization of Objects, Fields, Validation Rules, Automation

Custom Objects

- HandsMen Customer – Stores customer information
- HandsMen Product – Tracks product attributes
- HandsMen Order – Stores customer orders
- Inventory – Manages stock levels
- Marketing Campaign – Tracks promotions

Next Steps:

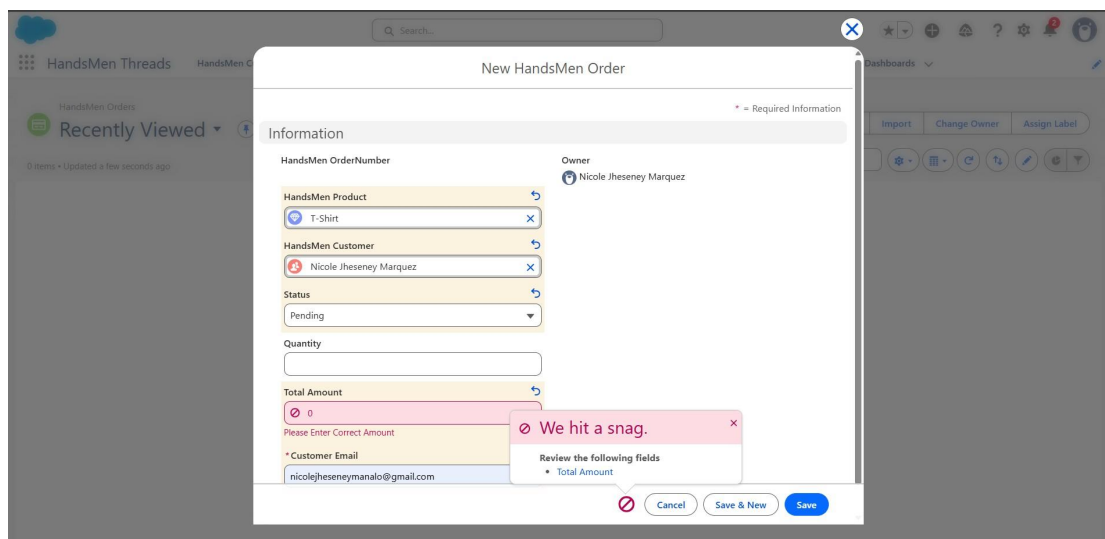
- Created a custom object by going to Setup > Object Manager > Create.
- Defined the object's label and name, then enabled reporting and search features.



- Saved the custom object.

Validation Rules

Order Object: Stops saving the record if `Total_Amount__c <= 0`.



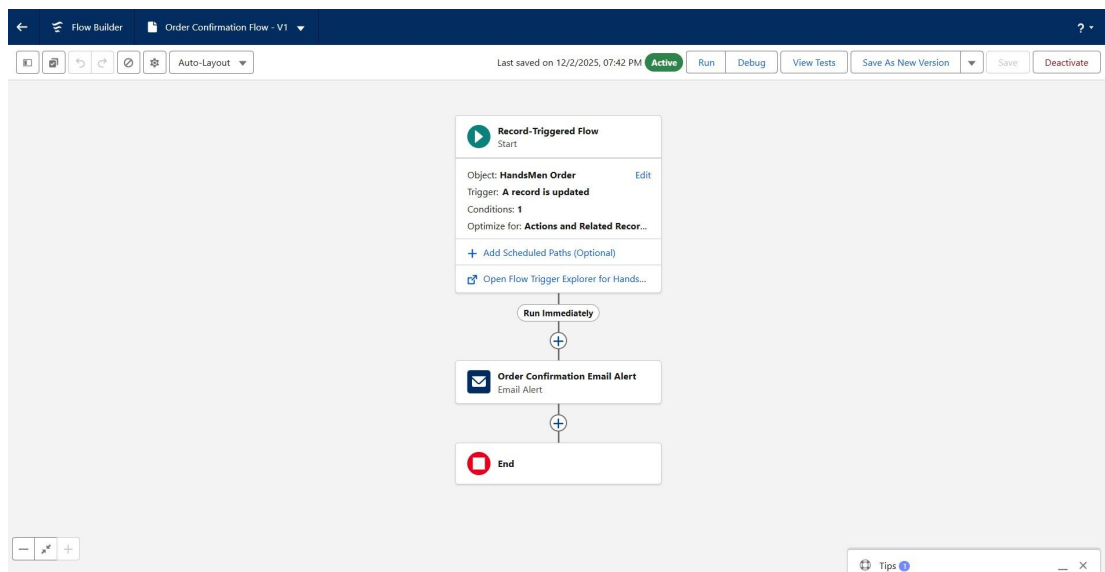
Customer Object: Validates that the email address contains ["@gmail.com"](mailto:nicolejhesneymano@gmail.com)

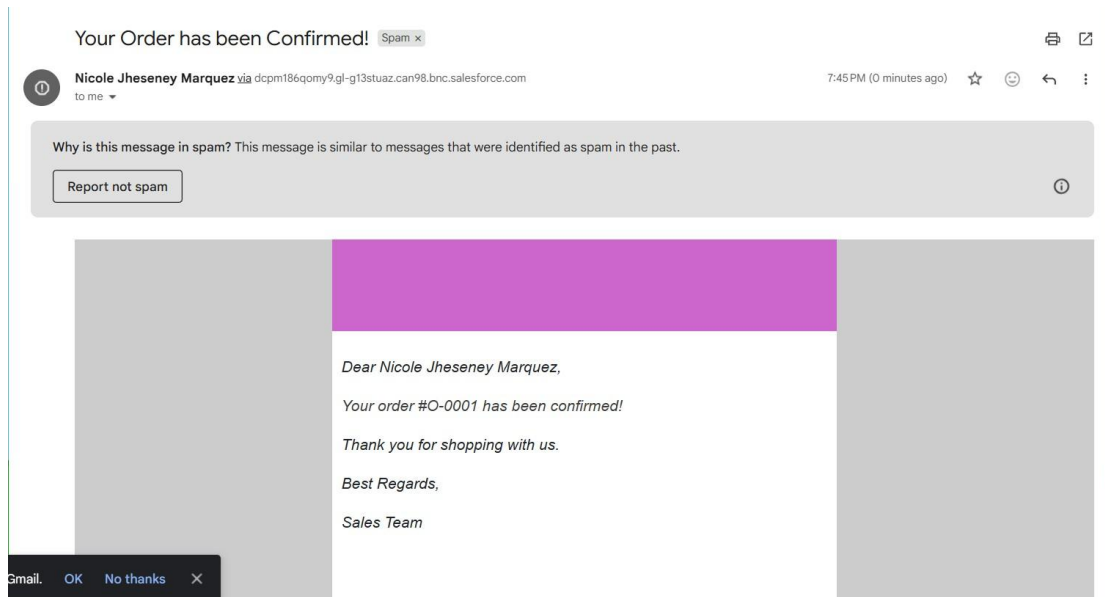
The screenshot shows a web application interface for creating a new customer. A modal window titled "New HandsMen Customer" is open, displaying a form with the following fields: "HandsMen Customer Name" (filled with "Nicole Jheseney Marquez"), "Email" (filled with "nicole@sales.com"), "Phone", "Loyalty Status" (dropdown menu), "FirstName", "LastName", and "Total Purchase". The "Owner" field is set to "Nicole Jheseney Marquez". A red error message box is overlaid on the form, stating "We hit a snag." and "Review the errors on this page." with a sub-message "Please fill Correct Gmail". The form has "Cancel", "Save & New", and "Save" buttons at the bottom.

Automation(Flows)

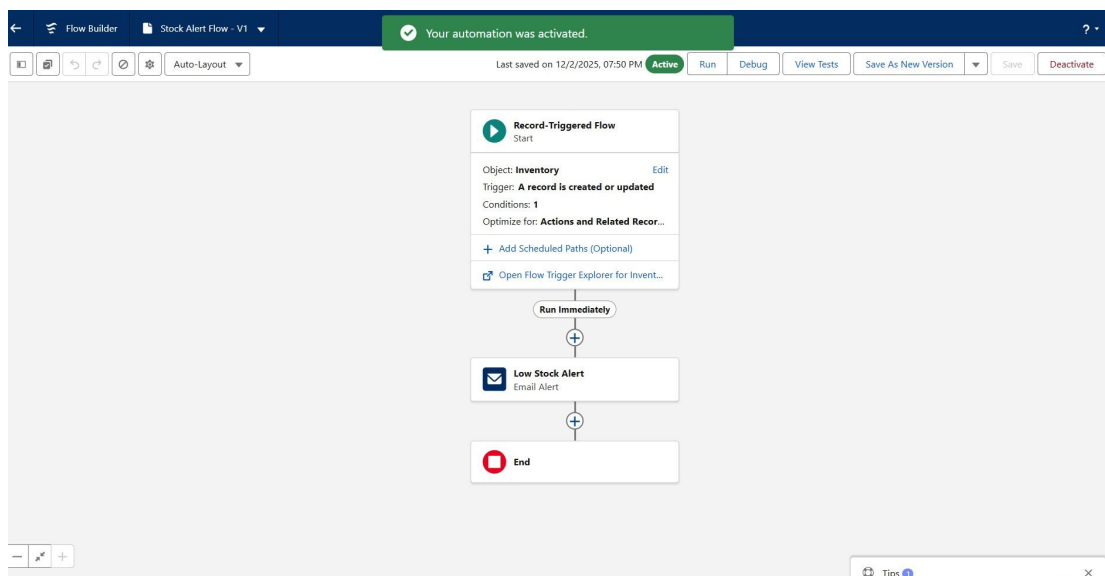
Record Triggered Flows

■ Order Confirmation Flow





Low Stock Alert Flow



←Delete foreverNot spam📧📎⋮

1 of 8<>

Stocks are getting low! Spam x

🕒

Nicole Jheseney Marquez via ln3btelqc99j-gl-g13stuaz.can98.bnc.salesforce.com to me

7:51 PM (0 minutes ago) ☆ 😊 ↶ ⋮

Why is this message in spam? This message is similar to messages that were identified as spam in the past.

Report not spam ⓘ

Dear Inventory Manager,

This is to inform you that the stock for the following product is running low.

Product Name: T-Shirt

Current Stock Quantity: 4

Please take the necessary steps to restock this item immediately.

Best Regards,

Inventory Monitoring System

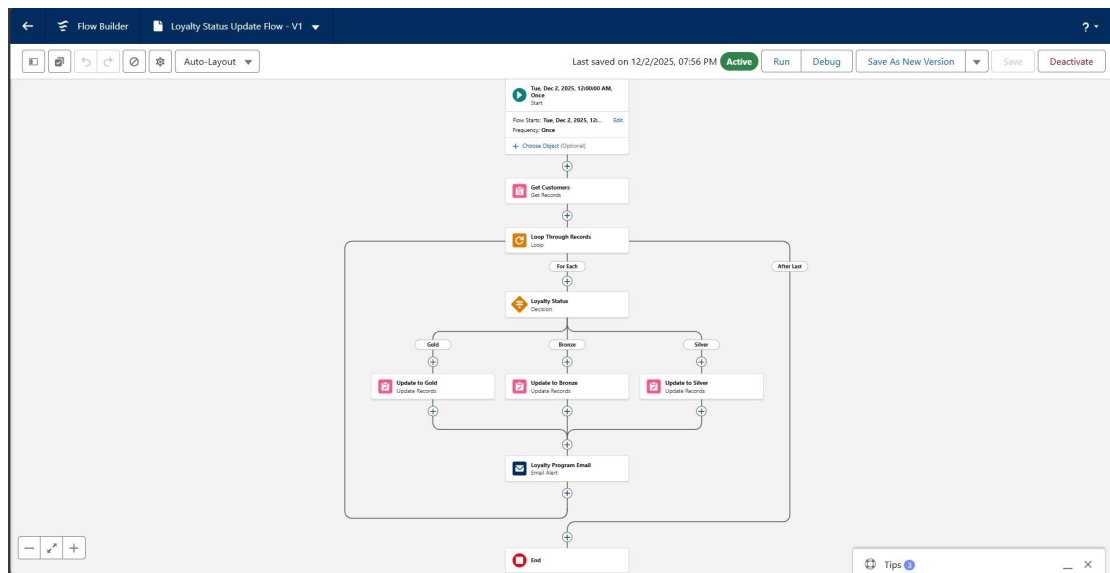
↶ Reply

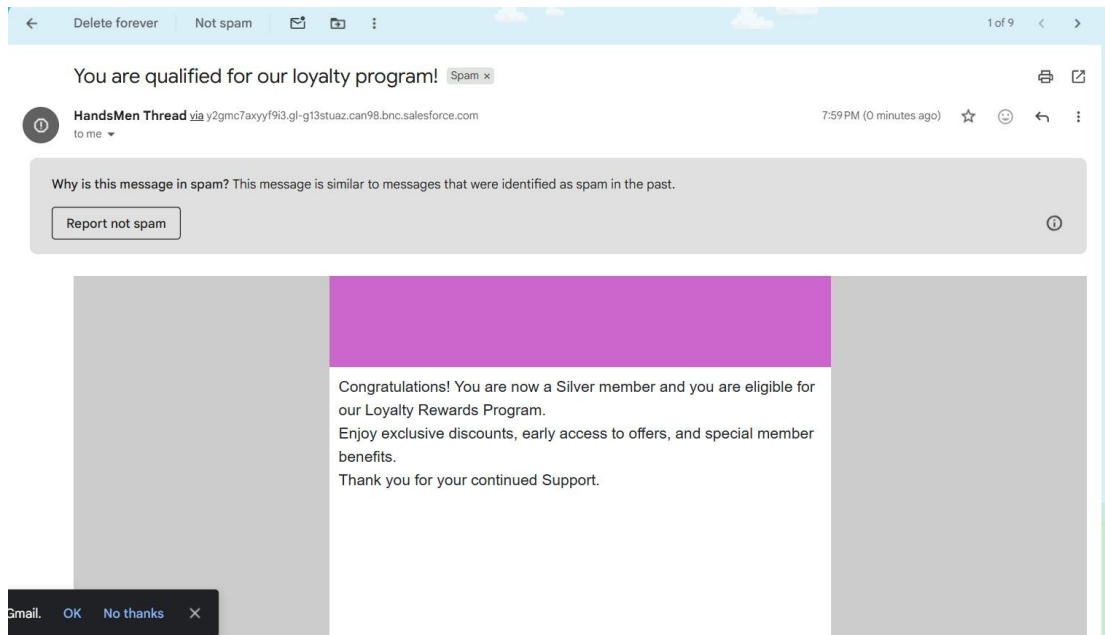
↷ Forward

😊

Scheduled-Triggered Flows

■ Loyalty Program Flow





2.3 Apex Classes, Triggers, Asynchronous Apex Classes

Apex Classes

- Configured a scheduled batch job to automatically monitor and update inventory levels, ensuring both data accuracy and timely replenishment.

```

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
13 • global void execute(Database.BatchableContext BC, List<SObject> records) {
14
15 List<HandsMen_Product__c> productsToUpdate = new List<HandsMen_Product__c>();
16
17 // Cast SObject list to Product__c list
18
19 • for (SObject record : records) {
20
21 HandsMen_Product__c product = (HandsMen_Product__c) record;
22
23 product.Stock_Quantity__c += 5; // Restock logic
24
25 productsToUpdate.add(product);
26
27 }
28
29 • if (!productsToUpdate.isEmpty()) {
30
31 • try {
32
33 update productsToUpdate;
34
35 • } catch (DmlException e) {
36
37 System.debug('Error updating inventory: ' + e.getMessage());
38
39 }
40
41 }
42
43 }
44
45 • global void finish(Database.BatchableContext BC) {
46
47 System.debug('Inventory Sync Completed');
48
49 }
50
51 // Scheduler Method
52
53 • global void execute(SchedulableContext SC) {
54
55 InventoryBatchJob batchJob = new InventoryBatchJob();
56
57 Database.executeBatch(batchJob, 200);
58
59 }
60
61 }

```

Apex Triggers

- Order Total Trigger - Features automatic calculation of the Total Amount from order quantity and unit price.

```

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 }

```

- Stock Deduction Trigger - Features automatic inventory deduction upon order confirmation.

```

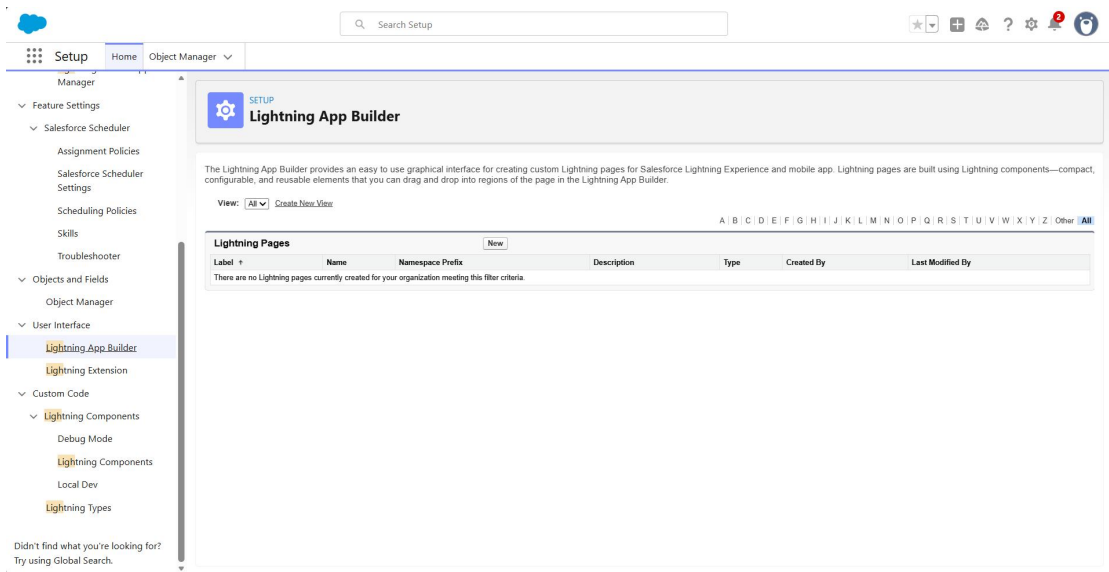
1 trigger StockDeductionTrigger on HandsMen_Order__c (after insert, after update) {
2     Set<Id> productIds = new Set<Id>();
3
4     for (HandsMen_Order__c order : Trigger.new) {
5         if (order.Status__c == 'Confirmed' && order.HandsMen_Product__c != null) {
6             productIds.add(order.HandsMen_Product__c);
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>{
14        [SELECT Id, Stock_Quantity__c, HandsMen_Product__c
15         FROM Inventory__c
16         WHERE HandsMen_Product__c IN :productIds]
17    };
18
19    List<Inventory__c> inventoriesToUpdate = new List<Inventory__c>();
20
21    for (HandsMen_Order__c order : Trigger.new) {
22        if (order.Status__c == 'Confirmed' && order.HandsMen_Product__c != null) {
23            for (Inventory__c inv : inventoryMap.values()) {
24                if (inv.HandsMen_Product__c == order.HandsMen_Product__c) {
25                    inv.Stock_Quantity__c -= order.Quantity__c;
26                    inventoriesToUpdate.add(inv);
27                    break;
28                }
29            }
30        }
31    }
32
33    if (!inventoriesToUpdate.isEmpty()) {
34        update inventoriesToUpdate;
35    }
36 }

```

Phase 3: UI/UX Development & Customization

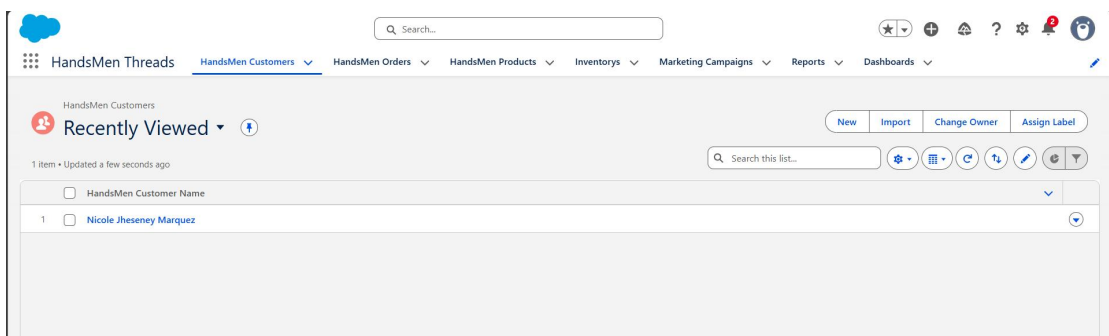
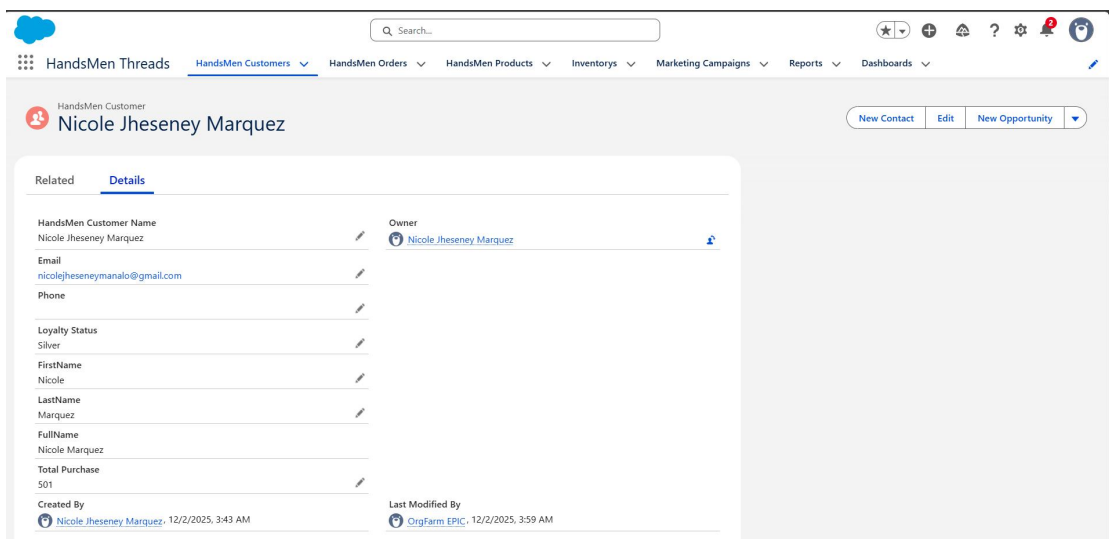
3.1 Lightning App Setup

Configured a custom Lightning App utilizing the App Manager tool.



3.2 Page Layouts and Dynamic Forms

Refined page layouts for the core objects (Customer, Order, Product, Inventory) to ensure a streamlined data entry experience. Deployed dynamic forms to contextually control field and section visibility, thereby ensuring appropriate data visibility and improving user efficiency.



3.3 User Management

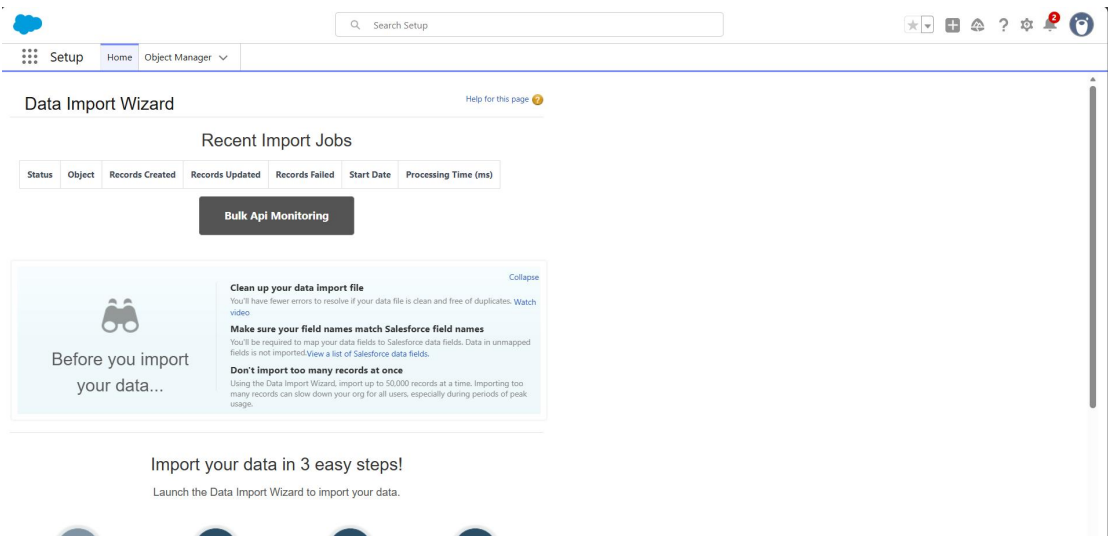
Configured user roles, profiles, and permission sets to govern access to the Lightning App's data and features, thereby ensuring a secure and customized experience for every user.

The screenshot shows the Salesforce Setup interface for User Management. The left sidebar contains a navigation menu with options like Setup, Home, Object Manager, and Users. The main content area is titled 'All Users' and provides a list of users with columns for Action, Full Name, Alias, Username, Role, Active, and Profile. Below the list, there are sections for user details, including 'Permission Set Assignments', 'Permission Set Group Assignments', and 'Permission Set License Assignments'. The interface is clean and professional, with a clear layout for managing users.

Phase 4:Data Migration, Testing & Security

4.1 Data Loading Process

Migrated existing customer, order, and inventory data using the Data Import Wizard and Data Loader. Ensured data quality by completing cleansing and mapping operations prior to the migration.



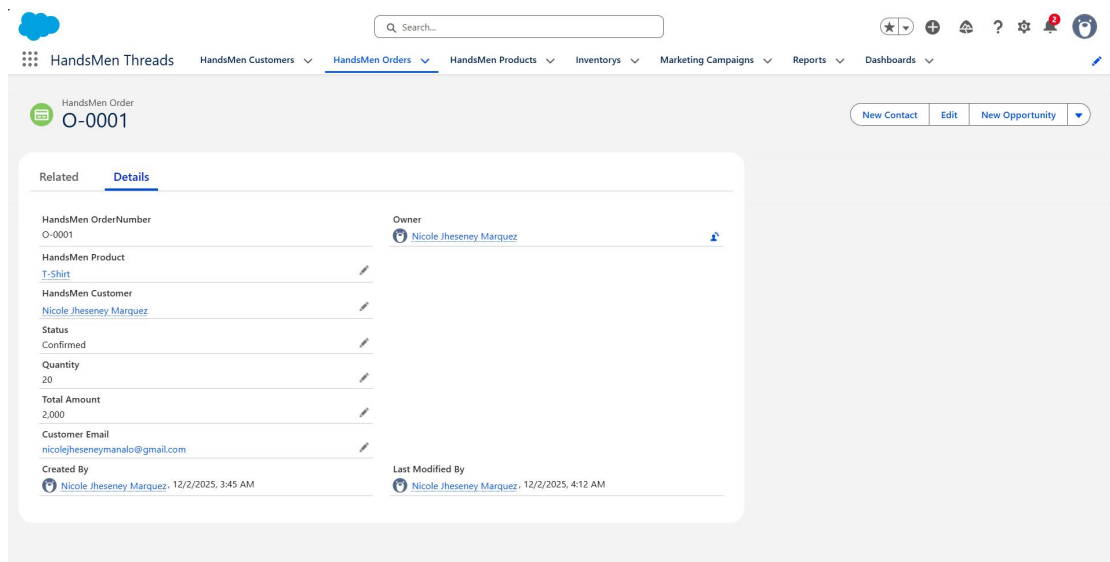
4.2 Field History Tracking, Duplicate Rules, Matching Rules

Tracked Objects & Fields:

HandsMen Orders – Fields tracked: Status and Total Amount

- Monitors the order lifecycle and financial value.
- Example: Nicole decides to purchase 50 t-shirts, each priced at \$2.
- A new Order record is created in Salesforce that captures all details of her purchase.

An Apex Trigger automatically calculates the Total Amount.



HandsMen Products – Fields Tracked: Price and Quantity

- The admin adds products into the Product__c object.
- Each product record includes key details like price, SKU, and other attributes.
- Monitors pricing updates and stock changes to maintain accuracy.
- Inventory object is automatically updated to reflect stock level.

The screenshot shows the 'HandsMen Threads' CRM interface. The top navigation bar includes 'HandsMen Threads', 'HandsMen Customers', 'HandsMen Orders', 'HandsMen Products', 'Inventories', 'Marketing Campaigns', 'Reports', and 'Dashboards'. The 'HandsMen Products' section is active, displaying 'T-Shirt' details. The 'Details' tab is selected, showing fields for 'HandsMen Product Name' (T-Shirt), 'SKU' (I-0002), 'Price' (\$100), 'Stock Quantity' (50), 'Created By' (Nicole Jheseney Marquez), and 'Last Modified By' (Nicole Jheseney Marquez). The 'Owner' field is also visible, showing 'Nicole Jheseney Marquez'.

The screenshot shows the 'HandsMen Threads' CRM interface. The top navigation bar includes 'HandsMen Threads', 'HandsMen Customers', 'HandsMen Orders', 'HandsMen Products', 'Inventories', 'Marketing Campaigns', 'Reports', and 'Dashboards'. The 'Inventories' section is active, displaying 'I -0001' details. The 'Details' tab is selected, showing fields for 'Inventory Number' (I -0001), 'HandsMen Product' (T-Shirt), 'Stock Quantity' (80), 'Stock Status' (Available), 'Warehouse' (Nics), 'Created By' (Nicole Jheseney Marquez), and 'Last Modified By' (Nicole Jheseney Marquez).

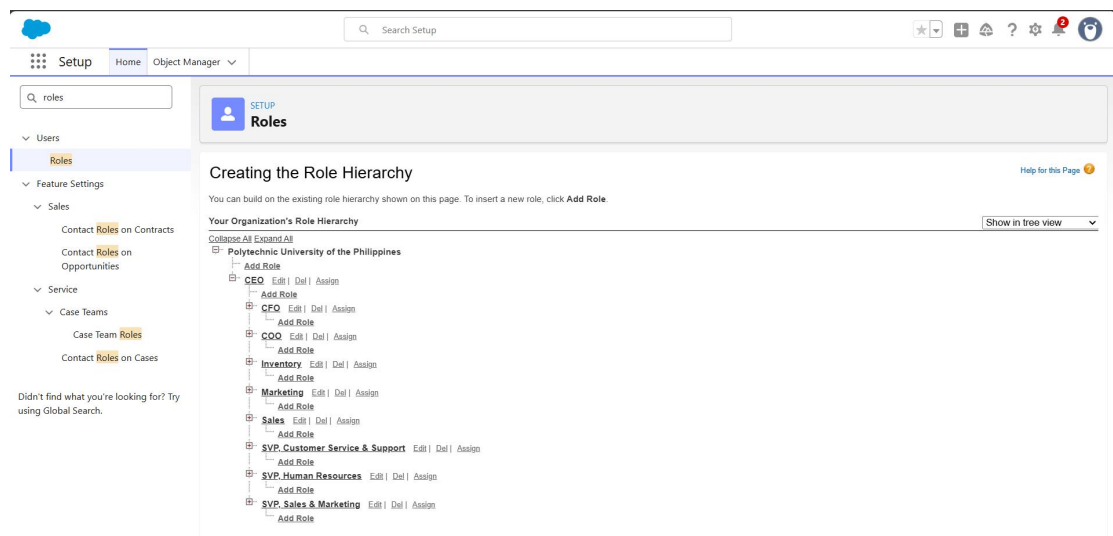
Duplicate and Matching Rules are implemented to prevent duplicate entries.

Standard Customer Matching Rule:

- Compares records based on First Name, Last Name, and Email.
- Prevents the creation of new customer records if it already exists.

4.3 Profiles, roles, role hierarchy, permission sets, and sharing rules

Used a combination of profiles, roles, permission sets, and sharing rules to implement a security model that mirrors the organizational structure. This ensures targeted and secure access, visibility, and editing rights for the Sales, Inventory, and Marketing teams.



Phase 5: Deployment, Documentation & Maintenance

5.1 Deployment Strategy

- Used Salesforce Change Sets to manage the deployment and migration of configurations and metadata between environments.
- Complex updates were managed using the Salesforce CLI and DevOps methodologies, ensuring a controlled and streamlined release process.
- Deployment adhered to a clear staging process: Development, Testing, and Production.

5.2 System Maintenance & Monitoring

- Ensured system health by establishing regular monitoring of performance, data integrity, and automation workflows.
- Configured Scheduled Apex for automated inventory tracking and proactive replenishment management.
- Maintained a commitment to improvement via user feedback mechanisms and periodic system audits.
- Maintained compliance and data security via scheduled security updates and role-based access reviews.

5.3 Troubleshooting Approach

- Implemented centralized logging where issues are categorized by severity.
- Predefined guides are used to address common problems (e.g., validation and automation errors).
- Critical issues follow an escalation procedure for prompt resolution, supported by available rollback plans.
- All fixes and resolutions are documented to support knowledge sharing and future reference.

Conclusion

The Salesforce CRM implementation for HandsMen Threads successfully delivered a scalable, secure, and automated platform that unifies all core data (customer, order, product, inventory, marketing). This transformation enhanced customer engagement, streamlined workflows, and significantly improved decision-making. By integrating automation and role-based security with proactive monitoring, the project positions HandsMen Threads for sustained growth and competitive advantage in the men's fashion industry.