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

Project Overview:

HandsMen Threads, a dynamic organization in the fashion industry, is embarking on a Salesforce project designed to revolutionize their data management and enhance customer relations. The project involves building a robust data model tailored to store all pertinent business data, ensuring a seamless flow of information across the organization.

Objectives:

A key aspect of this project is the maintenance of data integrity directly from the user interface (UI). This feature will safeguard the accuracy and consistency of the data, which is crucial for informed decision-making and reliable business operations. The project will integrate several new processes into the business workflow to improve customer service and operational efficiency:

1. Automated Order Confirmations: Post-order confirmation, customers will receive an email update, fostering engagement and strengthening customer relations.
2. Dynamic Loyalty Program: Customer loyalty statuses will be updated based on purchase history, enabling personalized rewards and promoting repeat business.
3. Proactive Stock Alerts: When stock levels drop below five units, automatic emails will notify the warehouse team, ensuring timely restocking and preventing stockouts.
4. Scheduled Bulk Order Updates: Daily midnight, the system will process bulk orders, updating financial records and adjusting inventory, ensuring accurate stock levels for daily operations.

Phase -1: Architecture & Planning

Created Objects like HandsMen Customer for the customer info, HandsMen Order for the order info, Inventory to know about the stock of the product, HandsMen Product for the product info and Marketing Campaign for marketing related issues.

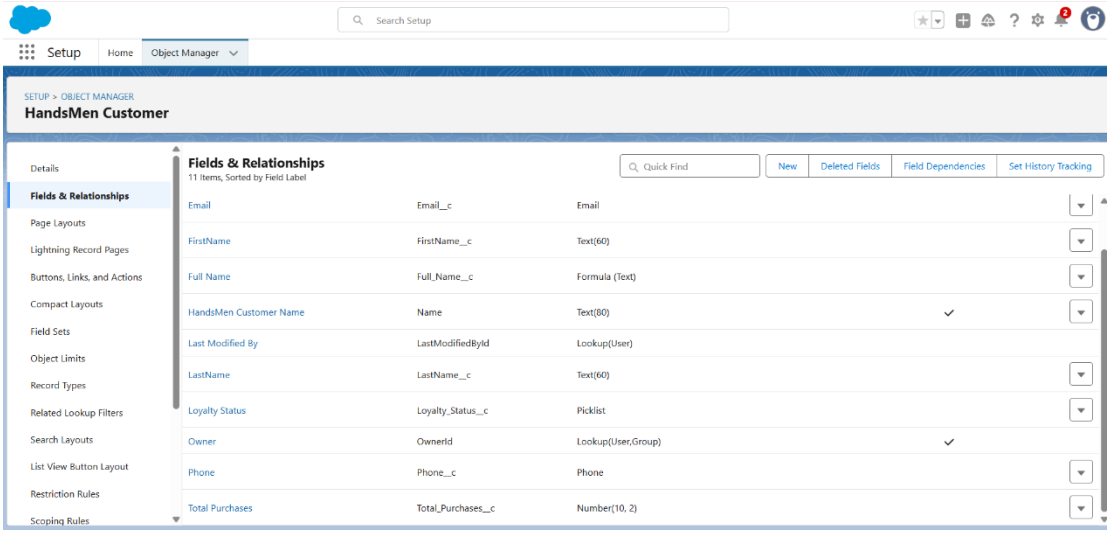
Created different validation rules on the field of objects so that user can able to see which he/she can see their own mistakes which are not satisfying the requirements.

Created flows, Apex triggers, Batch jobs for the purpose of automation and scheduled activities.

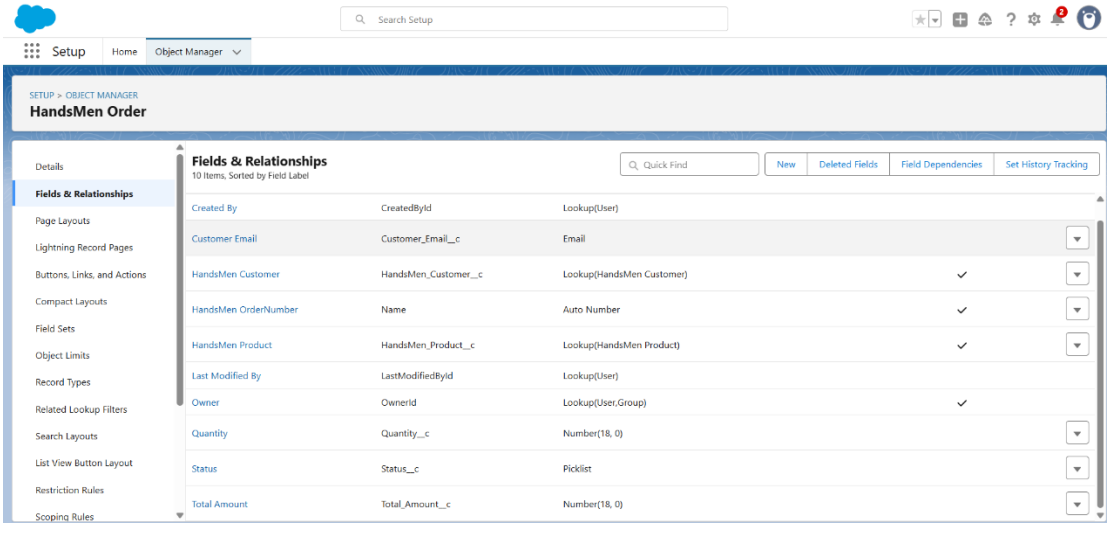
Designed some email alerts for knowing whether the order is confirmed or not for knowing the status of the stock and for loyalty status.

Phase -2:Development

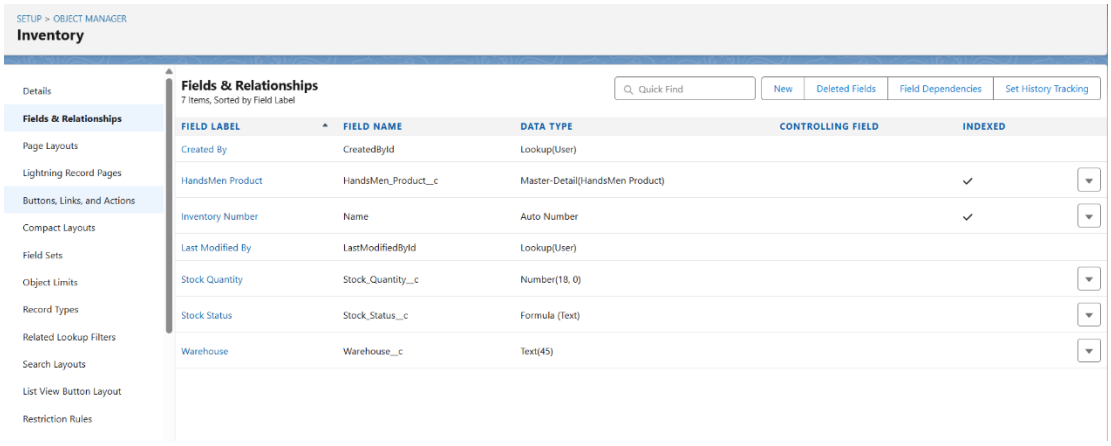
HandsMen Customer Object and it's fields:



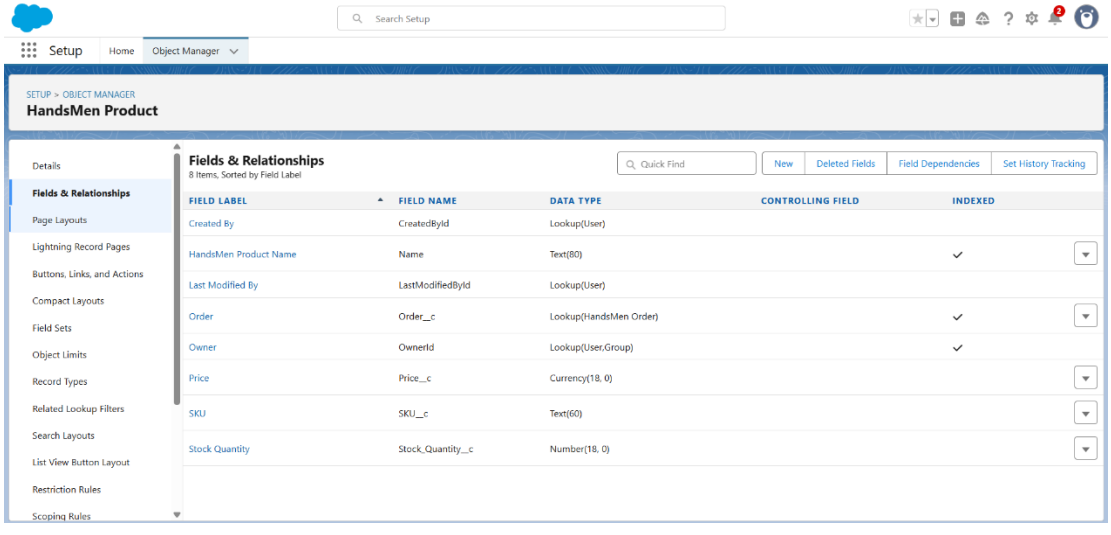
HandsMen Order Object and it's fields:



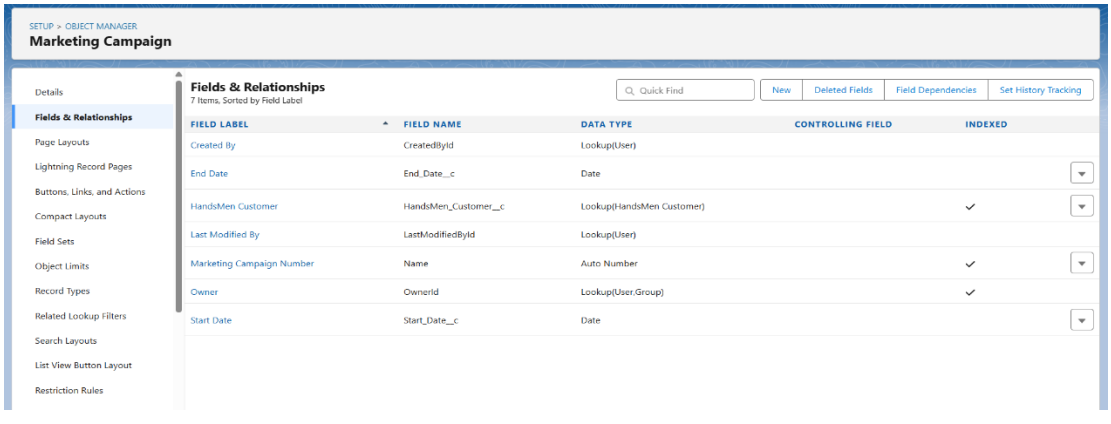
Inventory Object and it’s fields:



HandsMen Product Object and it’s fields:

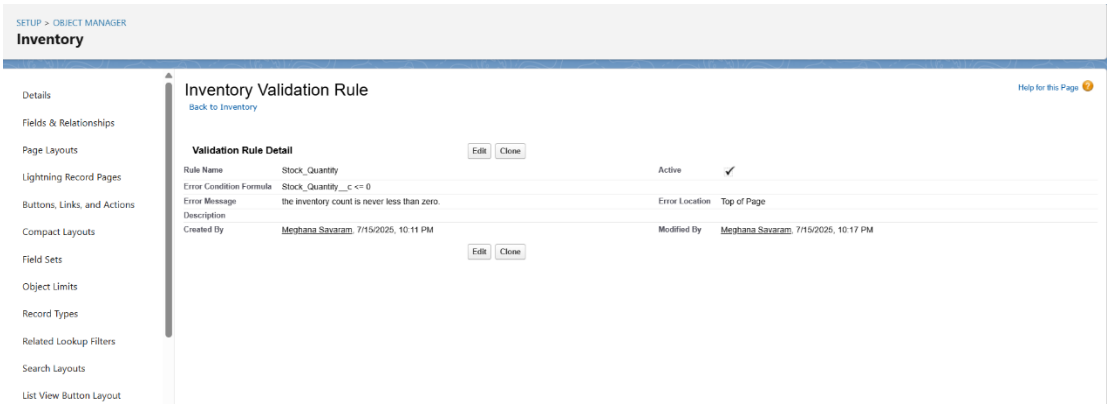


Marketing Campaign Object and it’s fields:

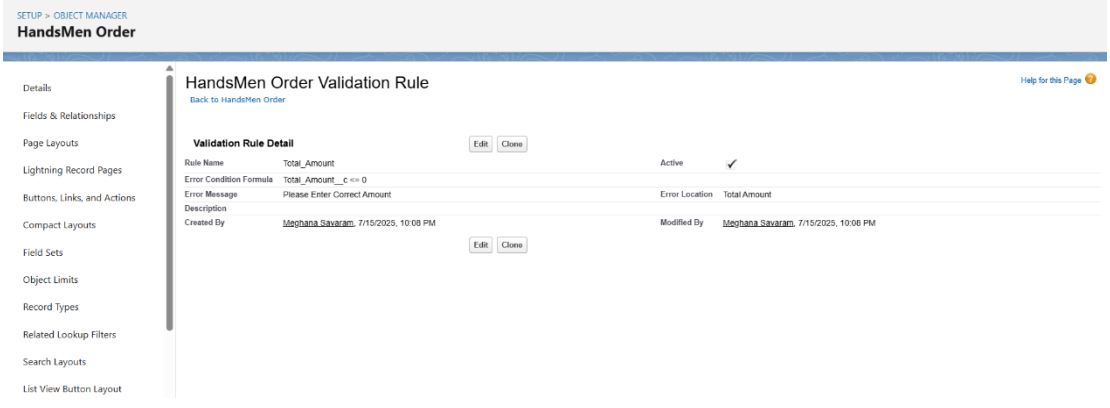


Validation Rules on the Objects:

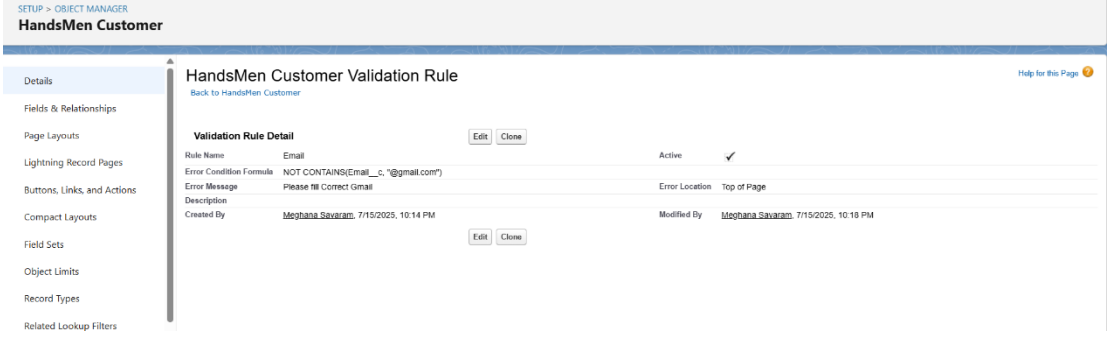
Inventory Validation Rule for the Stock Quantity:



HandsMen Order Validation Rule for Total Amount

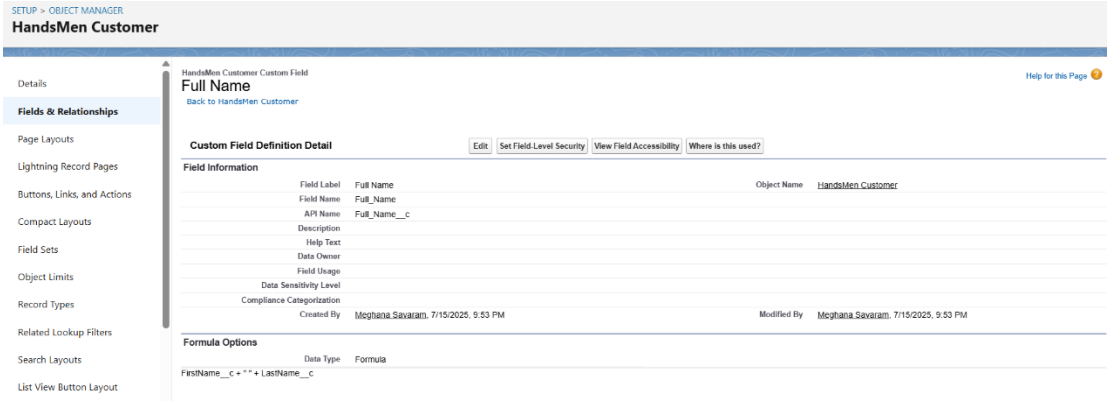


HandsMen Customer Validation Rule for Email

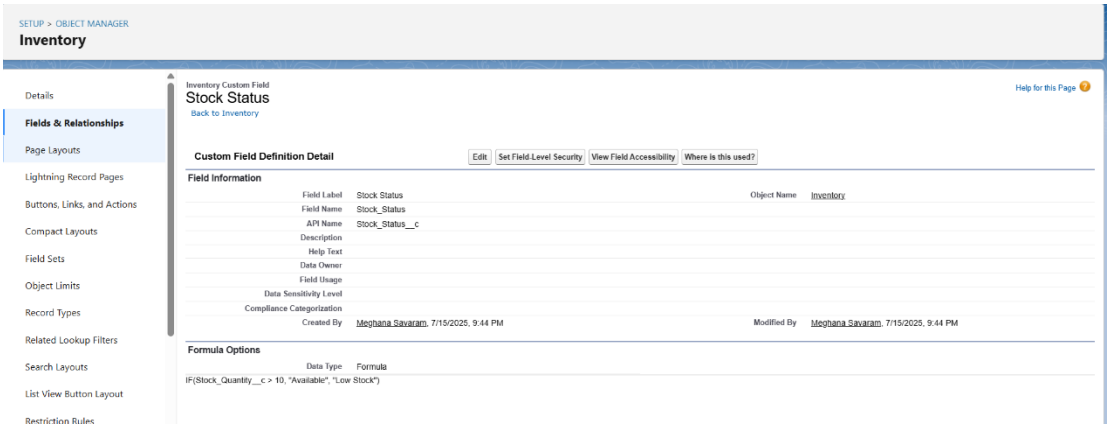


Formula Fields Used for the Objects:

FullName on HandsMen Customer for combining both FirstName and LastName

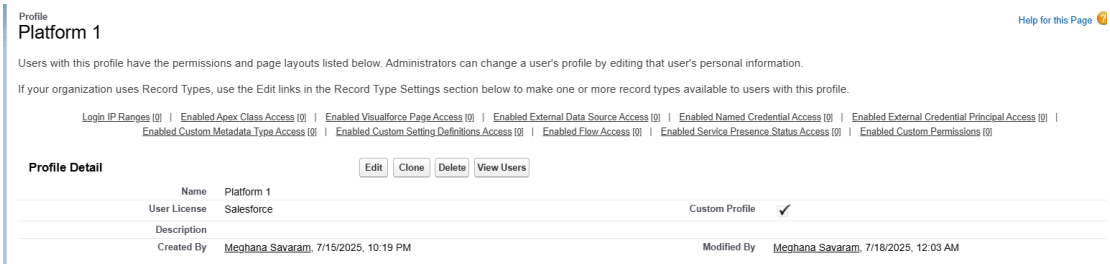


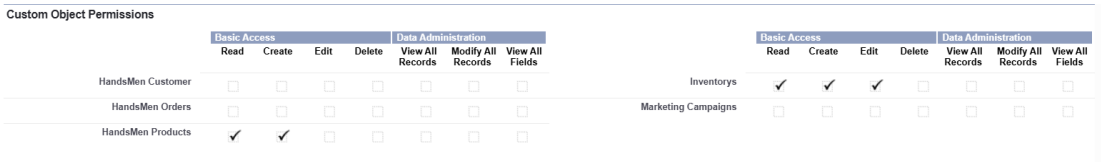
Stock Status on Inventory for knowing low stock and available stock



Profile:

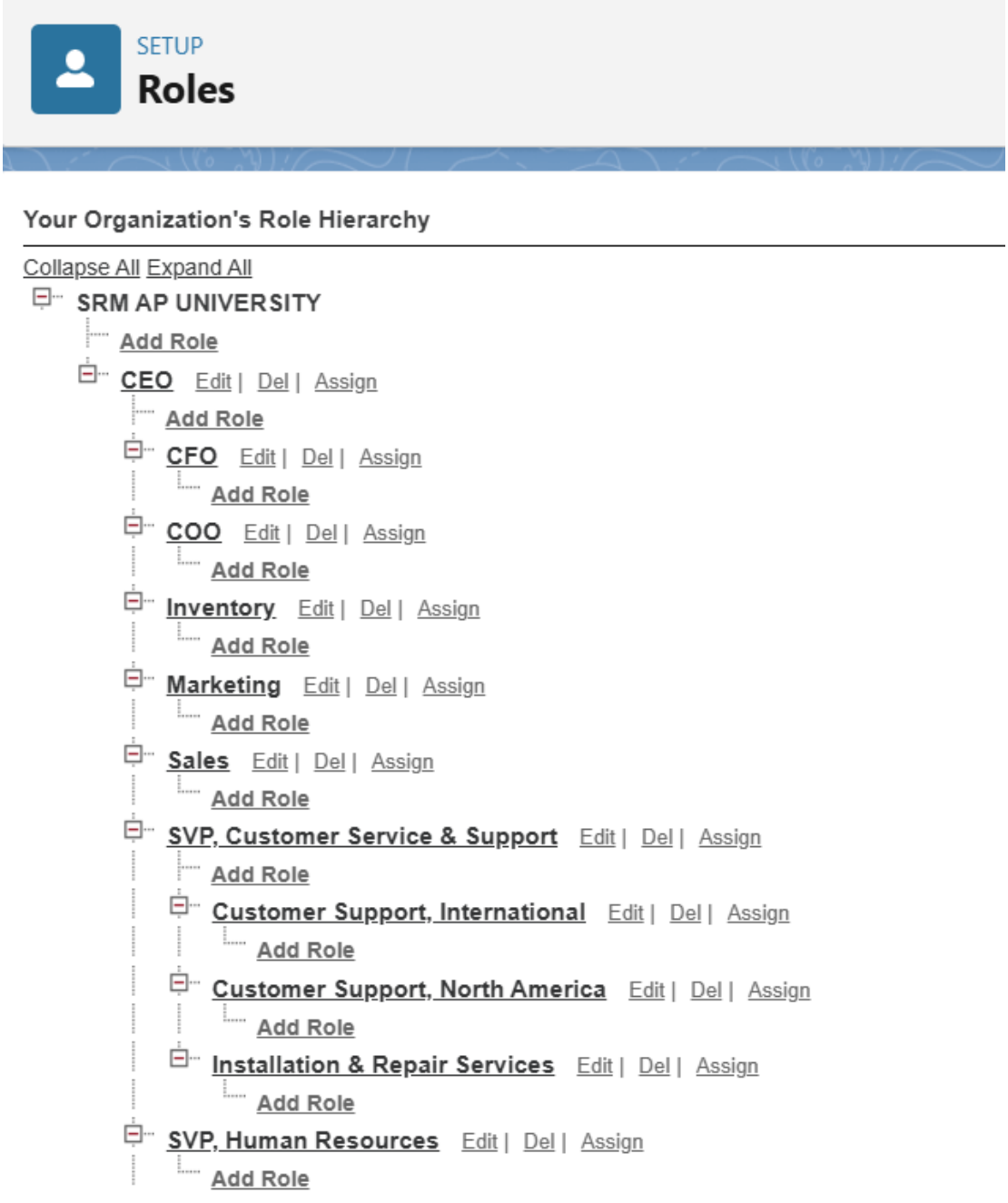
Created one profile for the roles of sales, inventory and marketing for the salesforce platform.





Roles:

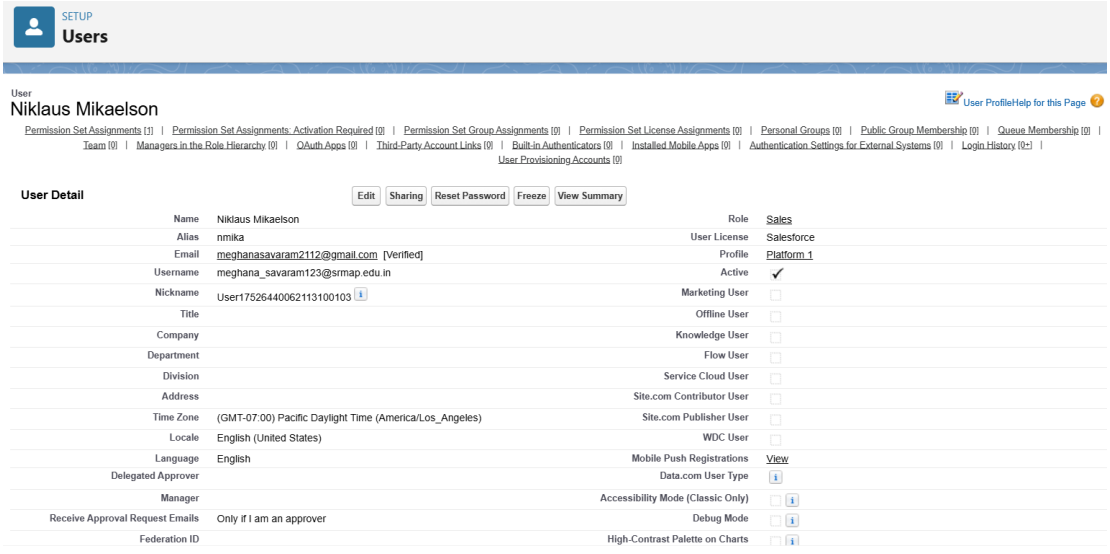
Roles like Inventory, Sales, Marketing are created under the CEO.



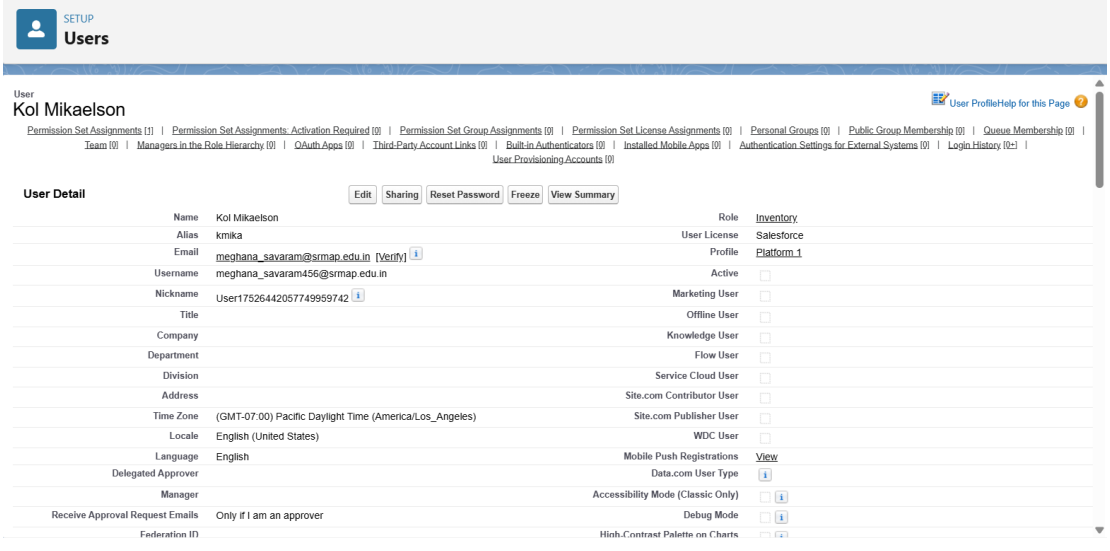
Users:

Three users are created for the roles created in the above phase.

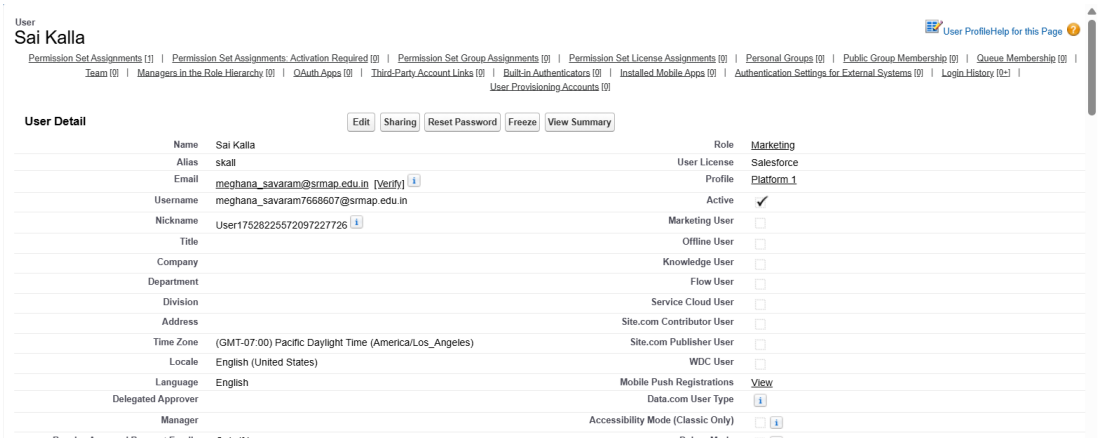
Sales User



Inventory User



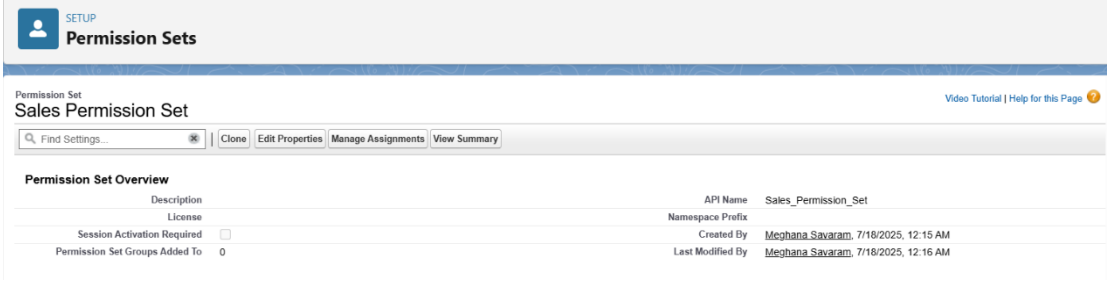
Marketing User:



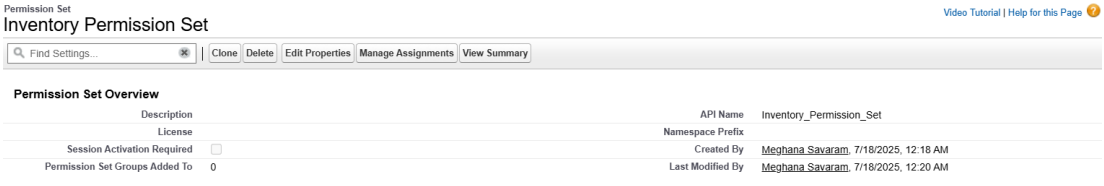
Permission Sets:

Created Pemission Sets for Inventory, Sales, Marketing.

Sales Permission Set



Inventory Permission Set

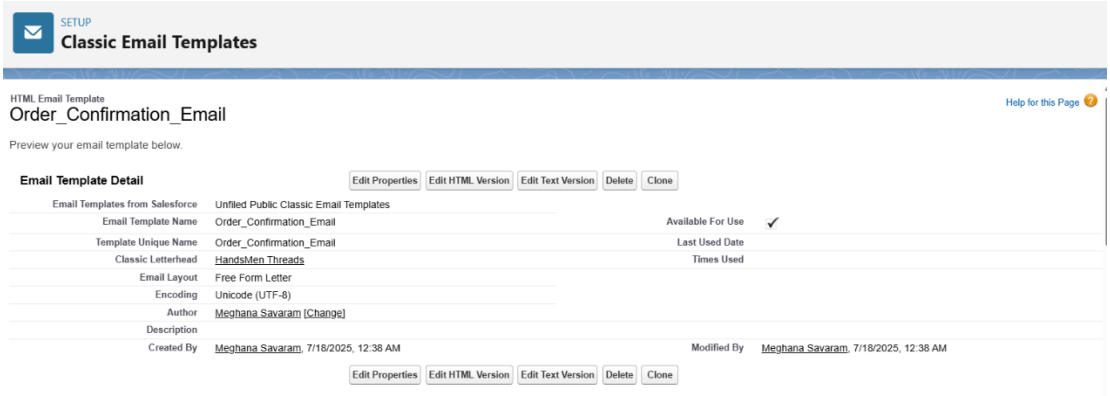


Marketing Permission Set

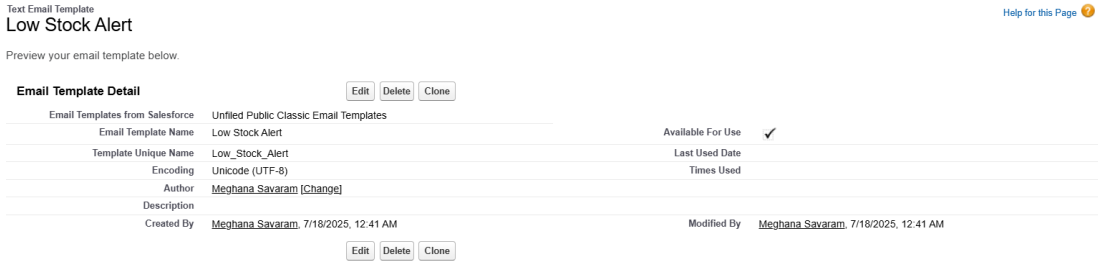
Email Templates:

Created Email Templates for the Email Alerts like what is the email that the particular alert should send when some update happens in the record of the objects.

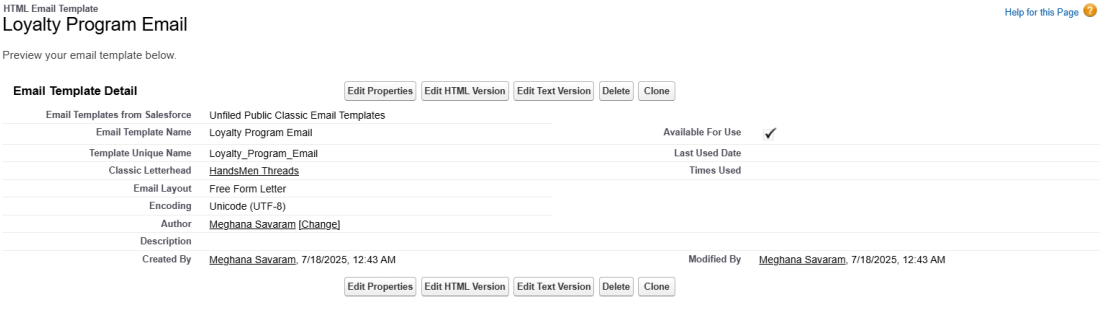
Order Confirmation Email Template



Low Stock Email Template

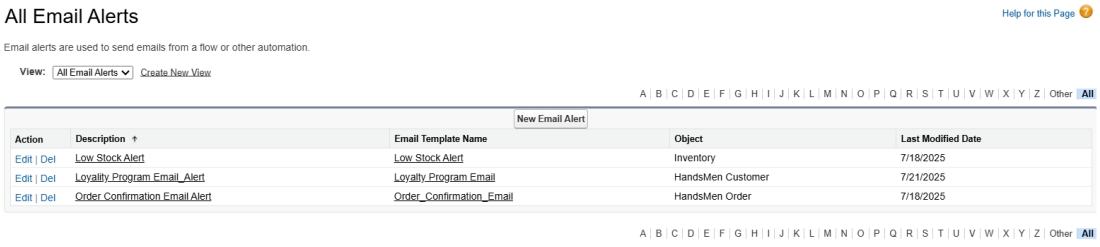


Loyalty Program Email Template



Email Alerts:

Created email alerts by associating email templates with them respectively



All Email Alerts

Help for this Page

Email alerts are used to send emails from a flow or other automation.

View: All Email Alerts Create New View

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All

New Email Alert

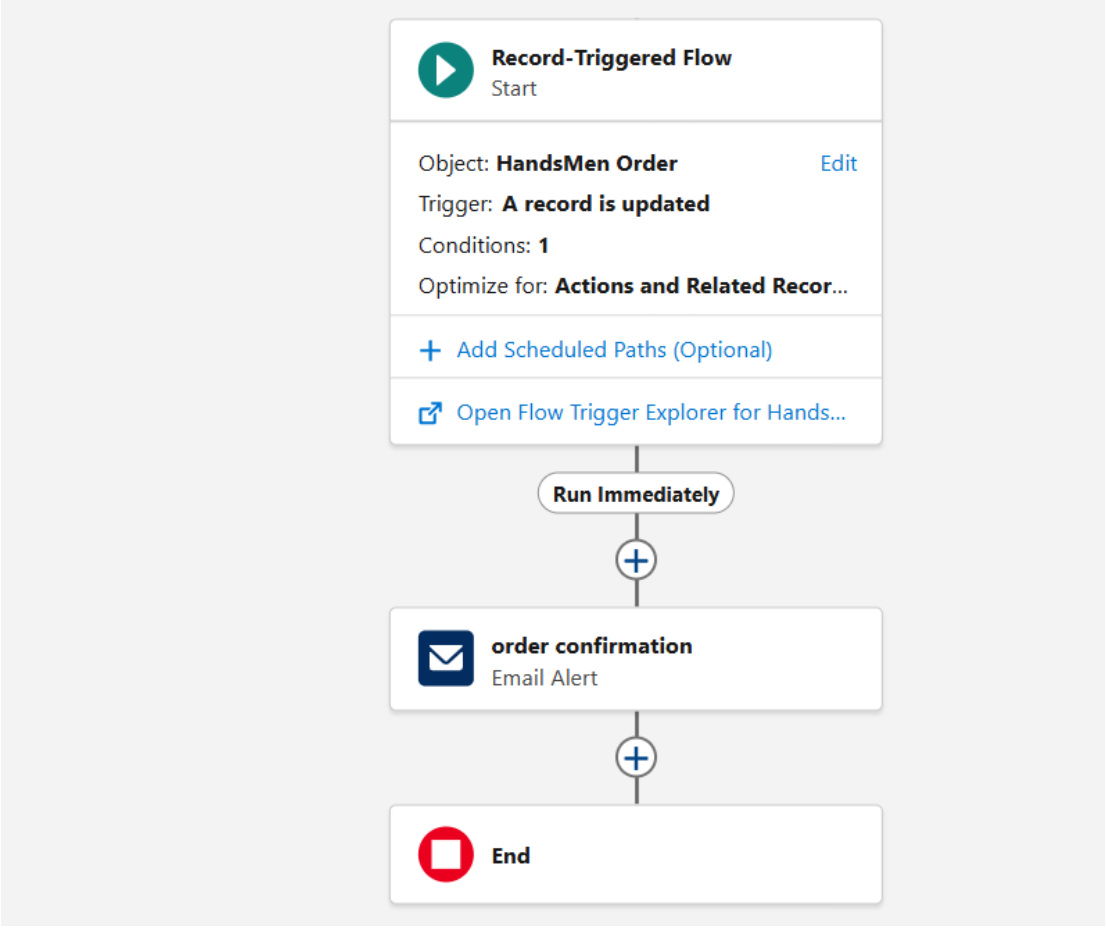
Action	Description	Email Template Name	Object	Last Modified Date
Edit | Del	Low Stock Alert	Low Stock Alert	Inventory	7/18/2025
Edit | Del	Loyality Program Email_Alert	Loyalty Program Email	HandsMen Customer	7/21/2025
Edit | Del	Order Confirmation Email Alert	Order_Confirmation_Email	HandsMen Order	7/18/2025

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All

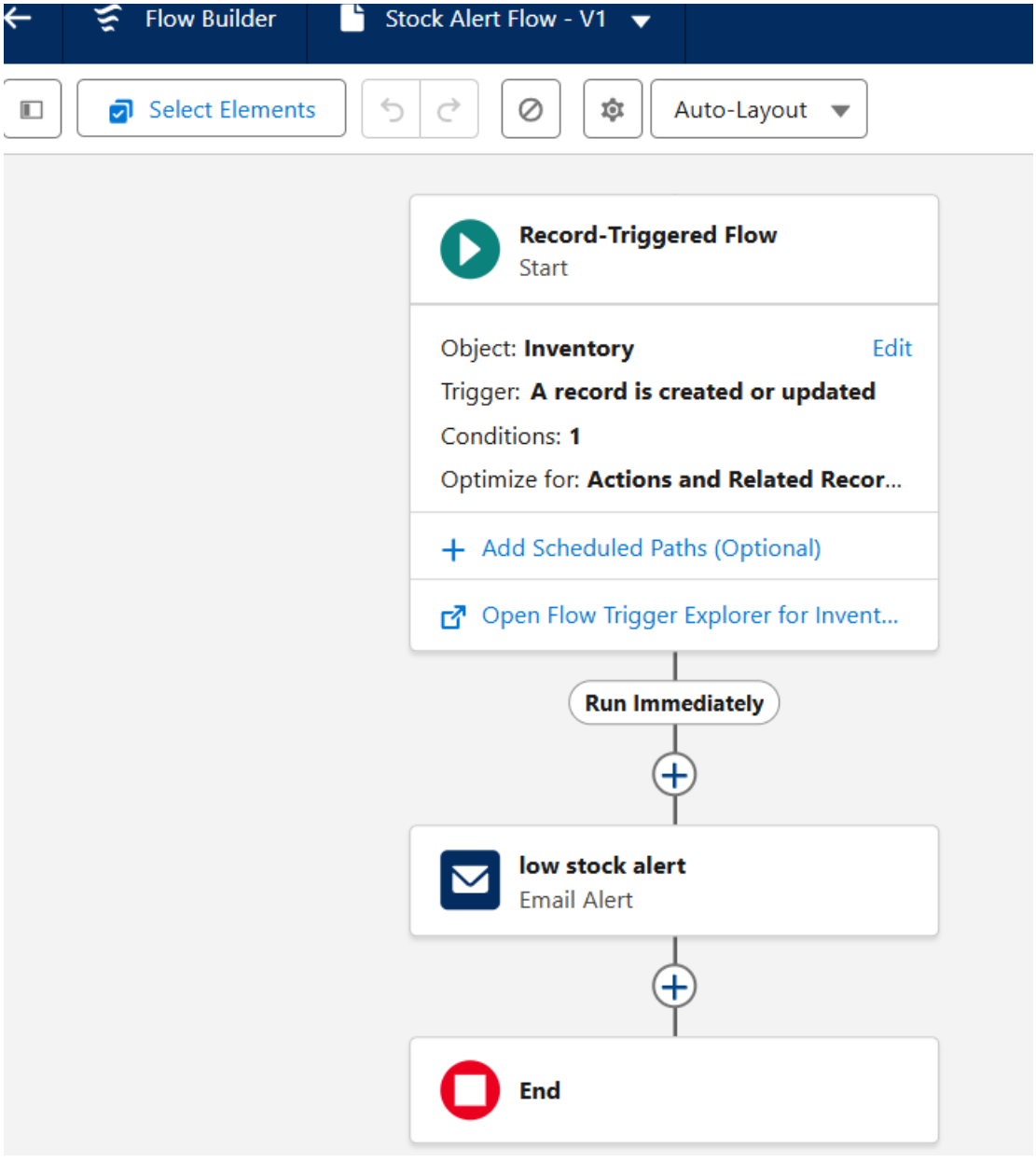
Flows, Apex triggers and Batch Jobs:

Designed for Automation Purpose and for the email alerts and also where there is an update in the records too.

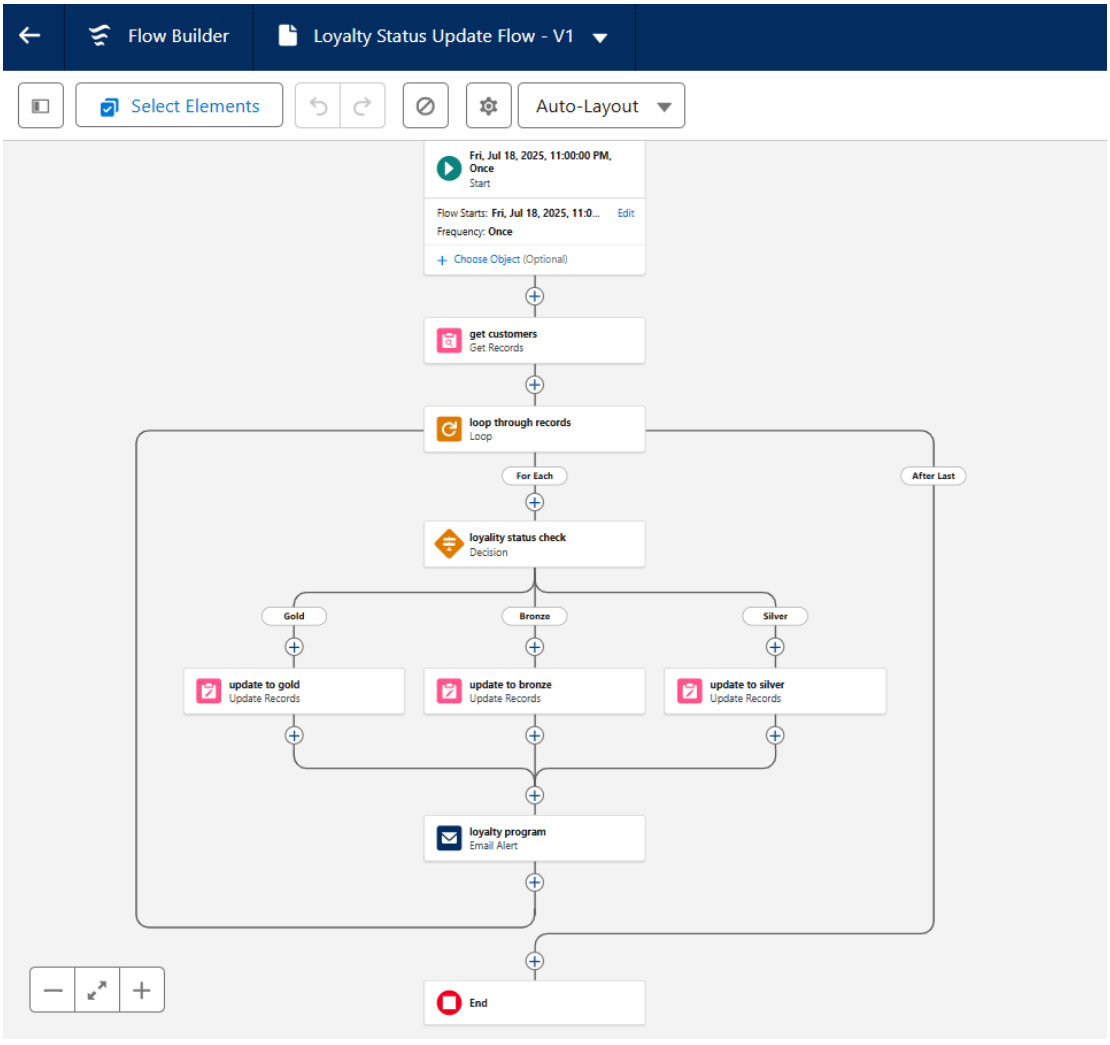
Order Confirmation Flow



Stock Flow Alert



Loyalty Update Flow



Apex Triggers

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

UpdateOrderTotal.apxt | Stock_Deduction.apxt

Code Coverage: None ▾ API Version: 64

```
trigger Stock_Deduction 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);
```

UpdateOrderTotal Apex Trigger

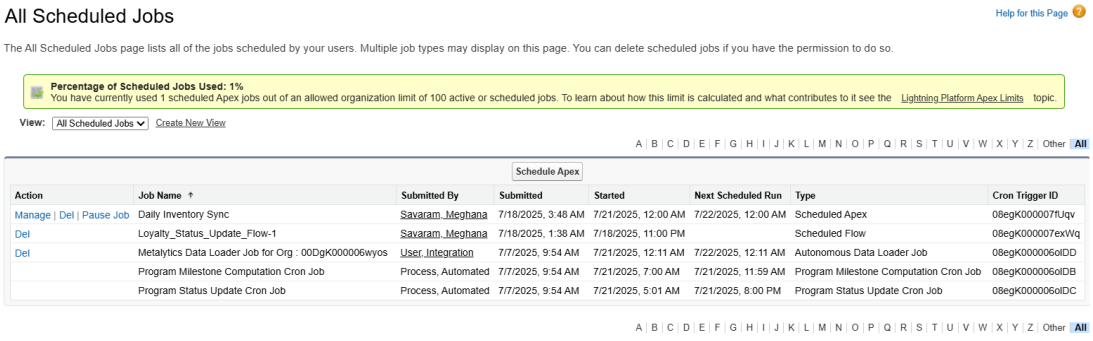
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

UpdateOrderTotal.apxt | Stock_Deduction.apxt

Code Coverage: None ▾ API Version: 64

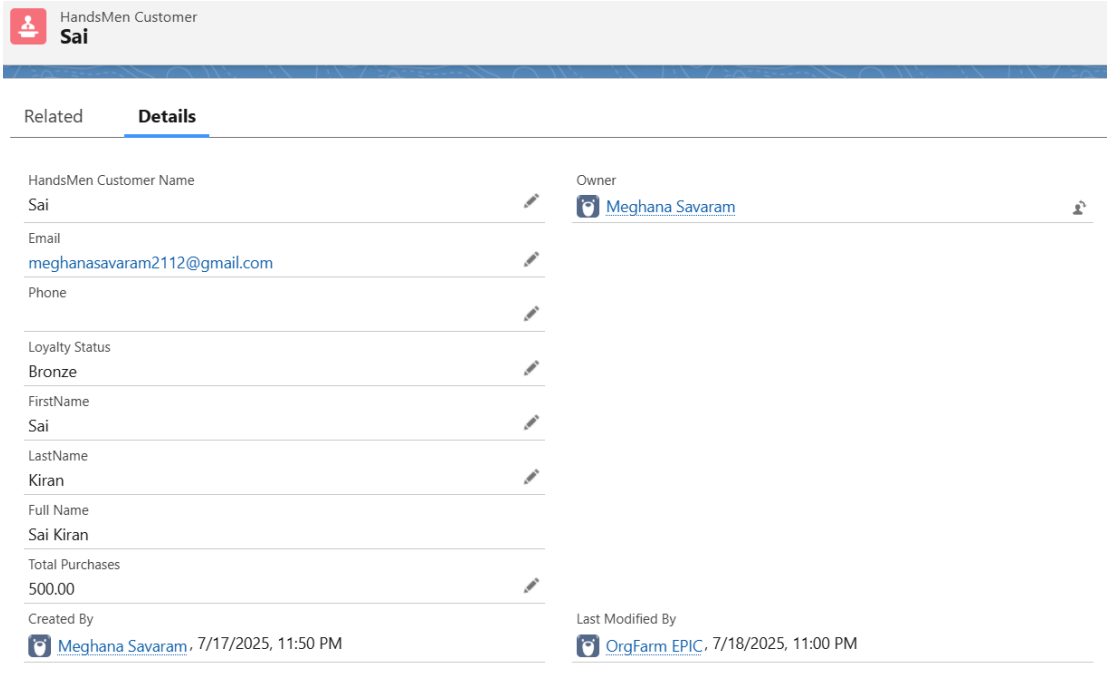
```
trigger UpdateOrderTotal on HandsMen_Order__c (before insert, before update) {
 Set<Id> productIds = new Set<Id>();

    for (HandsMen_Order__c order : Trigger.new) {
        if (order.HandsMen_Product__c != null) {
            productIds.add(order.HandsMen_Product__c);
        }
    }

    Map<Id, HandsMen_Product__c> productMap = new Map<Id, HandsMen_Product__c>(
        [SELECT Id, Price__c FROM HandsMen_Product__c WHERE Id IN :productIds]
    );

    for (HandsMen_Order__c order : Trigger.new) {
        if (order.HandsMen_Product__c != null && productMap.containsKey(order.HandsMen_Product__c)) {
            HandsMen_Product__c product = productMap.get(order.HandsMen_Product__c);
            if (order.Quantity__c != null) {
                order.Total_Amount__c = order.Quantity__c * product.Price__c;
            }
        }
    }
}
```

Batch Jobs

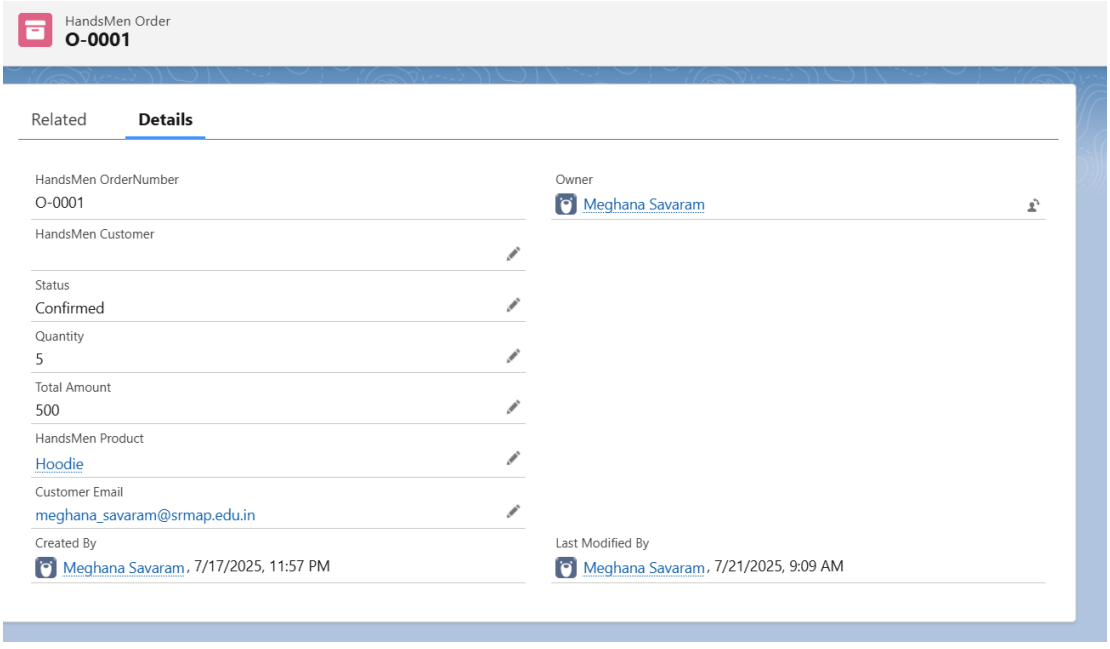


Phase -3: Testing

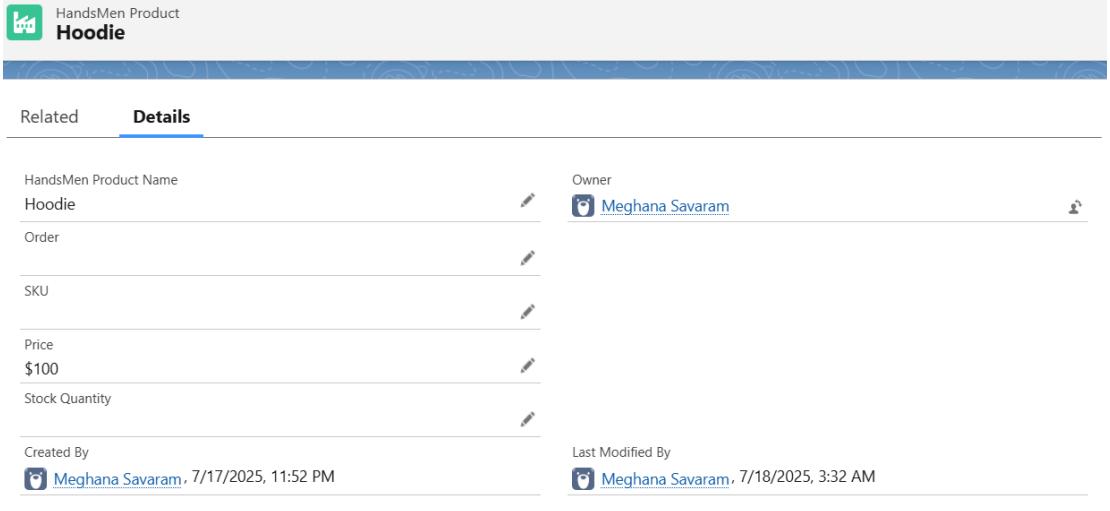
As the total purchases is equal to 500 this comes into the category of the Bronze and FullName also reflected due to the formula field.



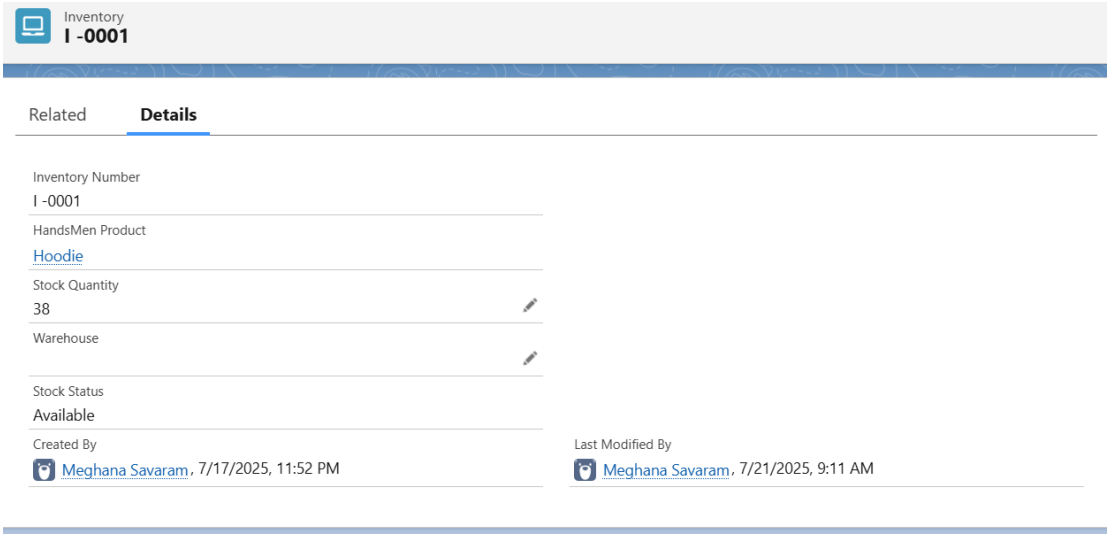
Here the total amount is reflected as the result of the quantity given to it because 100 is the price of 1 unit.



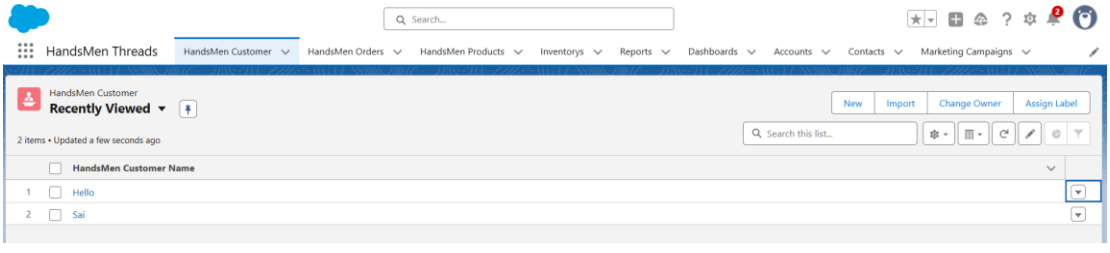
HandsMen Product Details:



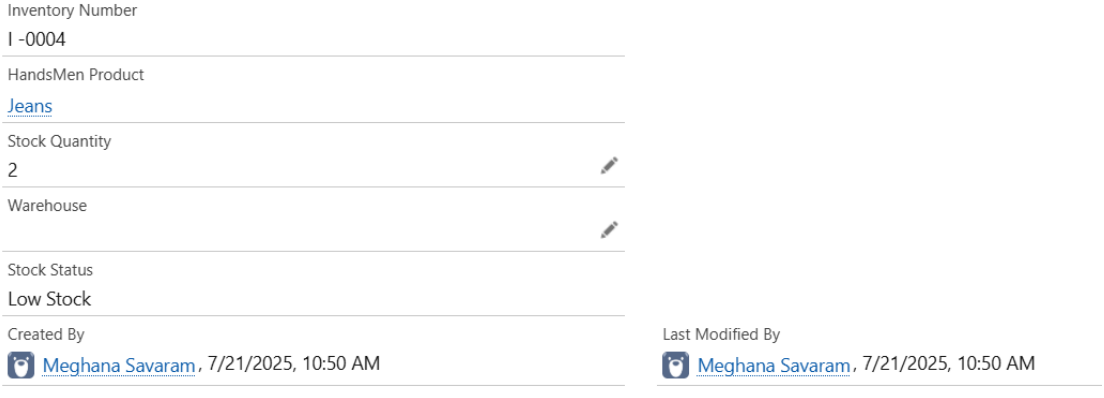
Whenever the customer buys the stock the stock quantity will be updated in the Inventory Object.



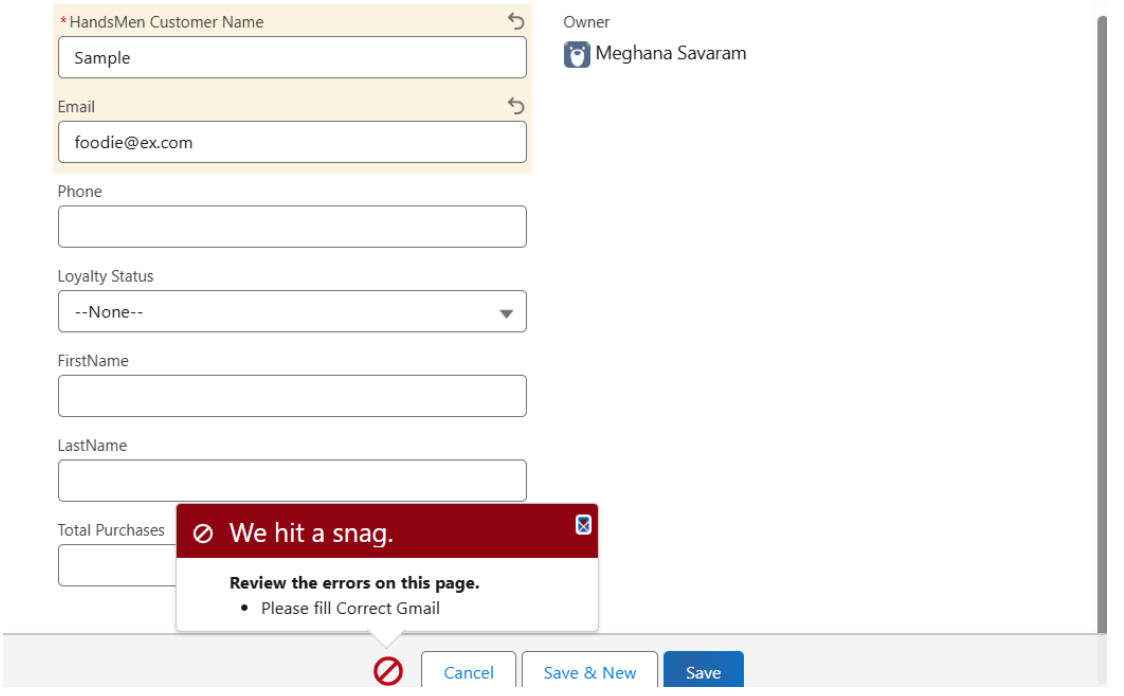
Total Application of the Project:



Low Stock for the Inventory:



Validation Rule for the Email:



Phase -4: Deployment:

After we made and tested the project, the app is fully ready. To share or move it, we can first push the org to VS Code, then send it to GitHub.

Conclusion:

The HandsMen Threads project used Salesforce to make business work better. It helped with things like handling customers, showing products, taking orders, checking stock, and running loyalty offers. Using tools like Custom Objects, Flows, Rules, Email Alerts, and Apex, the system keeps data right and updates fast. With less manual work and clear user roles, it makes jobs quicker and gives a better view of sales and stock.