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

## Project Overview

This Salesforce CRM implementation for HandsMen Threads, a dynamic organization in the fashion industry, constitutes a critical digital transformation initiative centered on establishing a **robust, centralized data model** designed to optimize operational efficiency and elevate customer relationship management. The core objective is to ensure unparalleled **data integrity** through direct enforcement at the user interface, thus guaranteeing the accuracy and consistency of business-critical information for reliable decision-making. The system integrates several automated processes intended to strengthen the business workflow, including: automated post-order confirmations to foster immediate customer engagement; a **dynamic loyalty program** that adjusts customer status based on purchase history for personalized retention strategies; **proactive stock alerts** that immediately notify warehouse staff when inventory drops below five units to prevent stockouts; and essential **scheduled bulk order updates** executed nightly via **Asynchronous Apex** to maintain accurate, up-to-date financial and inventory records. Ultimately, this project furnishes HandsMen Threads with the comprehensive, scalable architecture necessary for superior data governance and responsive operational execution, delivered through a phased approach encompassing Architecture & Planning, rigorous Development using **Apex and Flows**, thorough Testing & QA, and controlled Deployment & Training.

## Objectives

This project aims to achieve a strategic digital transformation by implementing a customized Salesforce CRM solution to centralize data, automate critical business processes, and elevate the overall operational and customer experience.

**Main Objective**

To establish a comprehensive, accurate, and automated Salesforce CRM platform for HandsMen Threads, thereby enabling strategic, data-driven decision-making and significantly improving customer relationship management and operational efficiency across the organization.

**Specific Objectives**

- **Data Governance & Quality:** Construct a robust data model and implement front-end validation rules (Validation Rules/Flows) to enforce superior **data integrity and quality**, guaranteeing the accuracy required for informed business analysis and trustworthy reporting.
- **Enhanced Customer Engagement:** Deploy automated post-order confirmations (Flows/Email Templates) to ensure real-time communication, fostering immediate transparency and boosting **customer satisfaction and loyalty**.
- **Revenue Generation & Retention:** Automate the calculation and update of **Dynamic Loyalty Program** statuses based on purchase history, enabling personalized reward strategies that promote **customer retention and drive repeat business**.
- **Inventory Optimization:** Implement **Proactive Stock Alerts** (Flows/Apex) to notify the warehouse team when inventory levels drop below five units, guaranteeing **timely replenishment** and minimizing revenue loss due to stockouts.
- **Financial Accuracy:** Develop and execute **Scheduled Bulk Order Updates** (Asynchronous Apex/Batch Jobs) nightly to process high-volume orders, ensuring **accurate and current financial and inventory records** are available for optimized daily planning and reconciliation.

# Phase 1: Requirement Analysis & Planning

## Understanding Business Requirements: Strategic Gaps and Solution Imperatives

This phase critically assessed the current state of HandsMen Threads' operations to define the required strategic shifts. The organization's reliance on siloed data and manual processes resulted in three core business challenges: operational inefficiency, compromised data integrity, and reactive customer/inventory management.

**Identified Strategic Gaps:**

1. **Data Dispersal and Integrity Risk:** Customer, order, and inventory data were fragmented, preventing a unified view and introducing significant risk of manual data entry errors.
2. **Reactive Inventory Control:** The lack of automated stock monitoring resulted in frequent, avoidable stockouts of high-demand fashion items, leading directly to lost sales and customer frustration.

3. **Inconsistent Customer Journey:** Customer engagement was inconsistent, with no formalized mechanism for immediate order updates or personalized recognition for loyalty, hindering retention efforts.

4. **Scalability Bottleneck:** Manual processing of large, bulk orders proved inefficient and delayed the accurate reconciliation of financial and stock figures.

**Solution Imperatives (Key User Needs):**

- **Single Source of Truth (SSoT):** Develop a centralized data model accessible through a fast, unified interface for all Sales and Operations staff.
- **Proactive Warning Systems:** Implement immediate, automated notifications for the Inventory team when stock levels approach critical lows.
- **Automated Customer Communications:** Ensure customers receive timely, automated confirmation emails upon order creation to enhance trust and brand communication.
- **High-Volume Data Processing:** Utilize asynchronous capabilities to handle high-volume batch updates efficiently outside of business hours.

## Defining Project Scope: Core Deliverables and Exclusions

The project scope is rigorously defined to ensure efficient delivery of the highest-impact features, establishing a strong foundation before pursuing future enhancements.

**Core Deliverables (In Scope):**

- **Data Architecture Foundation:** Full design and development of the **Data Model**, including custom objects (HandsMen_Order__c, Inventory__c, Loyalty_Status__c, HandsMen_Customer__c, HandsMen_Product__c, Marketing_Campaign__c) and field-level definitions.
- **Automation Suite:** Implementation of the four critical business automations using an appropriate mix of declarative (**Record-Triggered Flows, Validation Rules**) and programmatic (**Apex Triggers, Asynchronous Batch Apex**) development.
- **Security and Governance:** Establishment of a robust **Security Model** encompassing OWD, Role Hierarchy, Sharing Rules, Profiles, and Permission Sets to control access and maintain data confidentiality.
- **Operational Interface:** Configuration of the **"HandsMenThreads" Lightning Application** with streamlined navigation, custom List Views, and user-friendly Lightning Page Layouts.

- **Data Quality Assurance:** Implementation of comprehensive **Validation Rules** to ensure mandatory data capture and adherence to business logic at the point of data entry.

**Exclusions and Future Considerations (Out of Scope for current phase):**

- Integration with any external Enterprise Resource Planning (ERP) or third-party accounting systems.
- Custom development using Lightning Web Components (LWC) is excluded, unless specifically required for bonus points or a future phase.
- Development of advanced analytical models (e.g., AI/ML recommendations).
- Mobile application development or complex self-service portal functionality.

## Design Data Model and Security Model

### Data Model Architecture: The Relationship Blueprint

The architectural blueprint leverages Salesforce's standard capabilities while extending them with critical custom objects to manage the unique data relationships of a modern fashion retailer:

- **Customer Registry (HandsMen_Customer__c):** A custom object that serves as the nexus for all customer data, capturing personal identifiers, contact information, and linking directly to their Loyalty Status and Order History.
- **Product Catalog (HandsMen_Product__c):** A custom object used as the definitive product catalog, documenting item specifications, descriptions, pricing structures, and linking to the Inventory control system.
- **Transactional Records (HandsMen_Order__c):** Tracks the detailed history of every purchase, maintaining a strong relationship back to the originating Customer and associated Inventory/Products.
- **Inventory Control (Inventory__c):** A custom entity for precise, real-time tracking of item quantities and safety stock thresholds, serving as the source for stock alert automation.
- **Customer Retention (Loyalty_Status__c):** A dedicated object for managing the customer's dynamic loyalty tier and qualifying metrics, driven by purchase history updates.
- **Marketing Tracker (Marketing_Campaign__c):** A custom object designed to track promotional initiatives and measure marketing effectiveness against customer segmentation and sales data.

### Security Model: Data Governance Framework

A comprehensive security framework ensures that access is provisioned based on the principle of least privilege, adhering to the following governance layers:

- **Organization-Wide Defaults (OWD):** Initial baseline security is set restrictively, typically **Private** for sensitive records like HandsMen_Order__c and Loyalty_Status__c, necessitating explicit sharing mechanisms.
- **Role Hierarchy:** Used for vertical data access, granting managers visibility into their direct reports' data.
- **Profiles and Permission Sets:** Control **CRUD** (Create, Read, Update, Delete) permissions on objects and **FLS** (Field-Level Security) to mask sensitive data fields from unauthorized users.
- **Sharing Rules:** Implemented to open access horizontally where collaboration is required, overriding the restrictive OWD for specific user groups (e.g., Marketing needing Read Access to recent Orders).

**Operational Interface: The "HandsMenThreads" Lightning Application**

The **"HandsMenThreads" Lightning Application** is configured as the streamlined command center for all internal users. This single, dedicated application eliminates navigational friction by consolidating all essential business functions—including Customer Management, Order Processing, Inventory Control, and Reporting—into a unified, intuitive workspace. Key features include a tailored navigation bar to provide rapid access to core custom object tabs, optimized Lightning Page Layouts for improved data entry efficiency, and a dedicated utility bar for quick access to system-wide tools and internal resources. This design ensures user adoption is maximized by providing a focused, high-performance interface aligned with daily operational workflows.

## Stakeholders Mapping

This map identifies the core project participants, defining their roles, primary interests, and their designated data access level within the new CRM environment.

| Stakeholder Role | Primary Interest/Focus | CRM Access Level |
|---|---|---|
| **CEO** | Strategic ROI, Data-Driven Insights | Executive Profile (High-Level Read/Reporting) |
| **Sales Manager** | Team Productivity, Sales Performance Metrics | Sales Manager Profile (High-Level Write/Team Visibility) |
| **Inventory Manager** | Stock Accuracy, Alert Reliability | Inventory Ops Profile (Read/Write on Inventory/Product) |

| | | |
|---|---|---|
| **Marketing Manager** | Customer Segmentation, Loyalty Program Data | Marketing Analyst Profile (Read/Write on Customer/Loyalty) |
| **Sales Representative (Users)** | Order Entry Speed, Customer History Access | Sales Rep Profile (Transactional Write/Standard User) |
| **IT Administrators** | System Health, Deployment Strategy | System Admin (Full Configuration/Maintenance) |
| **End Customers** | Order Communications, Loyalty Benefits | External (No Direct Salesforce Access) |

## Execution Roadmap

This roadmap details the immediate, foundational milestones necessary to transition from architecture to the development phase.

| Milestone | Description | Timeline Focus |
|---|---|---|
| **Environment Readiness** | Finalize sandbox provisioning (Developer/Dev Pro) and establish best practices for source control and DevOps workflow setup. | Week 1 - Foundation |
| **Schema Deployment** | Complete the creation of all Custom Objects, Fields, Relationships, and foundational Data Quality Validation Rules defined in Phase 1. | Week 1 - Data Structures |
| **Security Configuration Baseline** | Implement the base Role Hierarchy and initial Profiles/Permission Sets, including FLS, to prepare for user access testing. | Week 1 - Governance |
| **Automation Design Review** | Final technical review and sign-off on the automation designs (Flow decision logic, Apex trigger handling, Batch Apex scheduling parameters). | Week 1 - Pre-Development |
| **Phase 2 Kickoff** | Formal commencement of back-end coding and declarative build (implementing the core automations). | Start of Week 2 |

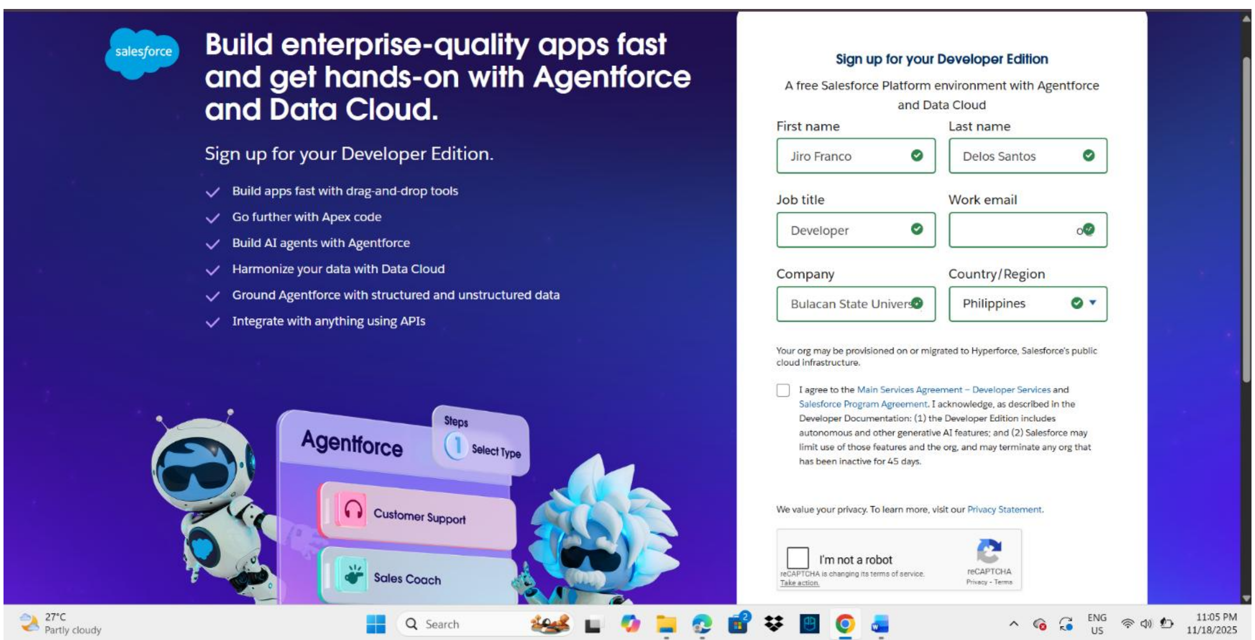
# Phase 2: Salesforce Development – Backend & Configurations

Phase 2 constitutes the critical implementation stage where the architectural blueprints designed in Phase 1 are transformed into a functional Salesforce CRM for HandsMen Threads. This phase involves establishing the necessary **DevOps workflow and environment readiness** to ensure a controlled and high-quality build. The core focus is on backend development, which includes configuring all custom objects and their relationships, implementing comprehensive **Validation Rules** to enforce superior data quality at the point of entry , and developing the four essential business automations. These automations leverage a strategic mix of declarative tools, primarily **Record-Triggered Flows**, for tasks like post-order confirmations and proactive stock alerts, alongside programmatic solutions, specifically **Asynchronous Apex (Batch Jobs)**, to efficiently manage high-volume, scheduled tasks such as nightly bulk order updates. Successful completion of this phase furnishes the organization with a robust, automated operational backbone for superior data governance and responsive execution.

## Setup environment & DevOps workflow

### Developer Org Setup:

1. Registered at [https://developer.salesforce.com/signup](https://developer.salesforce.com/signup)
2. On the sign-up form, I enter the following details:

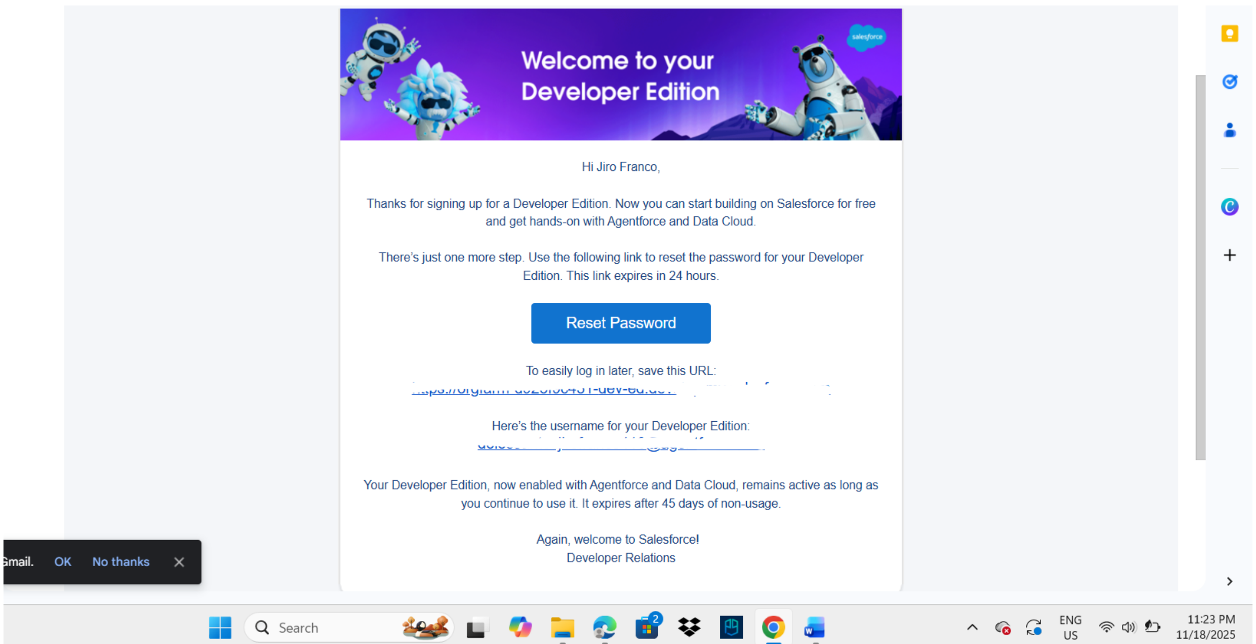



   a. First Name and Last Name
   b. Email (Personal Email)
   c. Job Title (Developer)
   d. Company (Bulacan State University)
   e. Country/Region (Philippines)

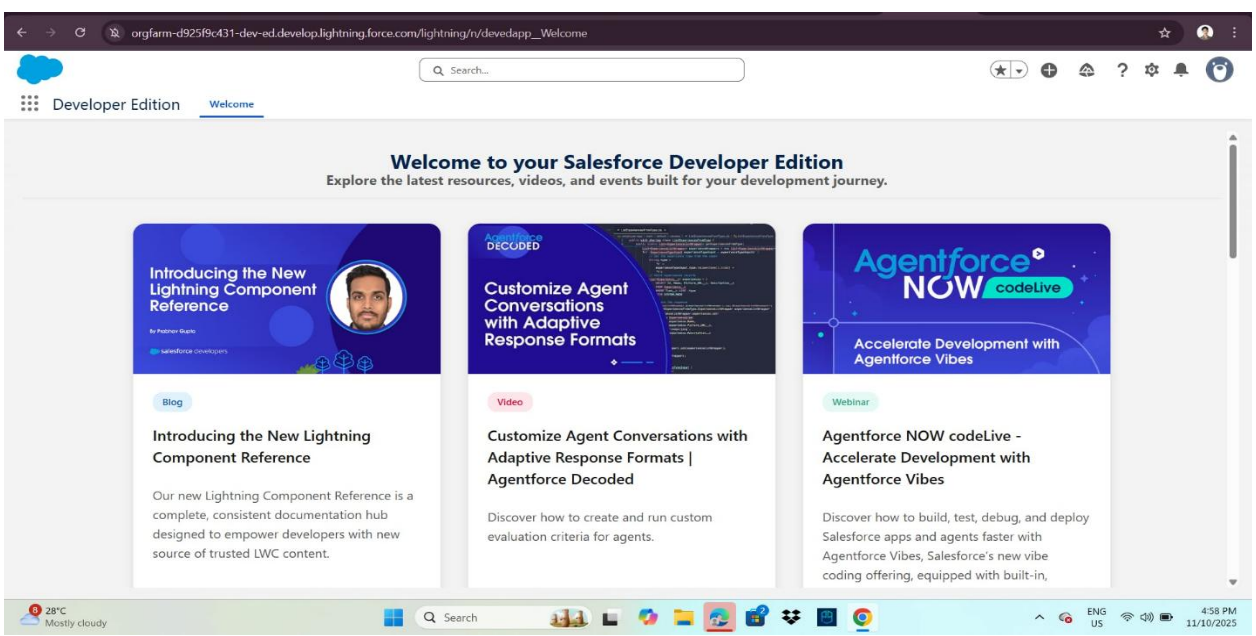
After completing all the information needed in the sign-up, I submit it to activate my developer edition account.

## Account Activation

1. After signing up I receive an email to verify my registered account.




2. To verify my account, I reset my password and answer a security question.
3. After the verification of my account, it redirects me to my salesforce set-up page.



**DevOps Workflow:**

1. **Source Control & Versioning:** A **Git repository** will be utilized as the single source of truth for all programmatic assets (**Apex Classes**) and metadata configurations. This ensures robust version tracking, collaborative development, and the ability to revert changes seamlessly.
2. **Change Migration:** For controlled deployments across the sandboxes and eventually to the Production environment, the native **Change Sets** tool will be employed. This method packages metadata components to facilitate precise, documented migration of configuration and code.
3. **Testing Protocol:** A strict **Testing Strategy** is mandated, requiring a minimum of **75% code coverage** for all Apex Classes and Triggers before deployment is authorized. This rigorous threshold ensures the reliability and stability of the programmatic business logic.
4. **Sandbox Utilization:** The **Developer Sandbox** will serve as the **primary development and unit testing environment** for all new features and bug fixes.
5. **Environment Readiness:** This strategy aligns with the foundational milestone of **Environment Readiness** by finalizing sandbox provisioning and establishing source control best practices early in the project lifecycle.
6. **Data Backup Strategy:** To ensure rapid recovery and compliance, a scheduled **Backup Strategy** will be executed. This involves performing **weekly data exports** of critical transactional and customer data using the native **Data Loader** utility.

## Customization of Objects, Fields, Validation Rules, Automation

1. **Custom Object:** HandsMen_Customer__c

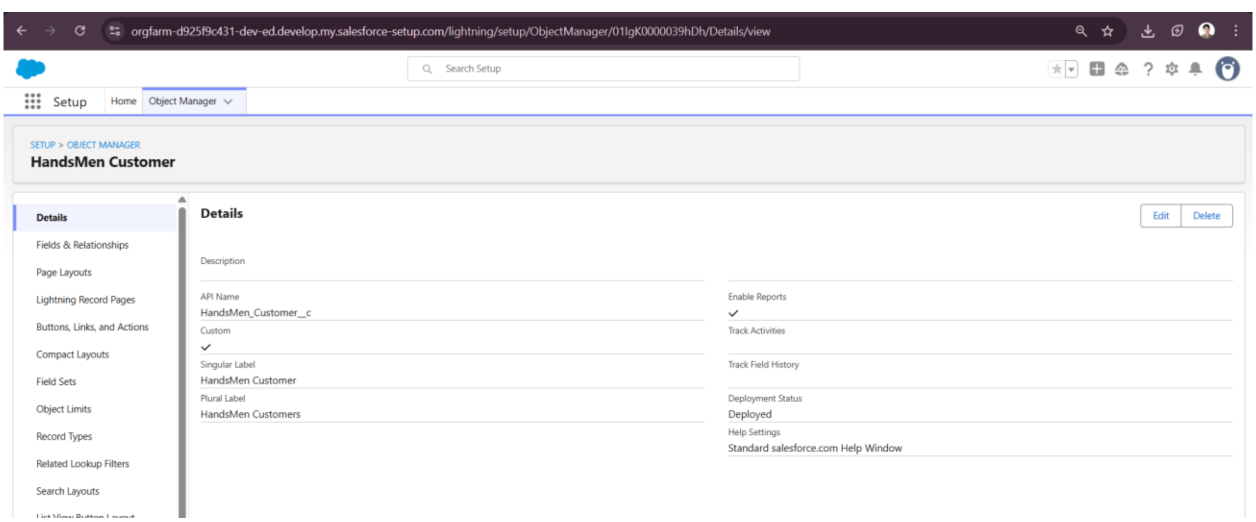


*Figure: HandsMen Customers (Custom Object)*

### 1.1. Fields Created for HandsMen Customers

- HandsMen Customer Name (Name - Text, 80) - Required, Indexed
- Email (Email__c - Email)
- FirstName (FirstName__c - Text, 60)
- LastName (LastName__c - Text, 60)
- FullName (FullName__c - Formula Text)
- Phone (Phone__c - Phone)
- Loyalty Status (Loyalty_Status__c - Picklist: Bronze, Silver, Gold) - Default: Bronze
- Total Purchases (Total_Purchases__c - Number, 18, 0)
- Owner (OwnerId - Lookup User,Group) - Indexed
- Created By (CreatedById - Lookup User)
- Last Modified By (LastModifiedById - Lookup User)

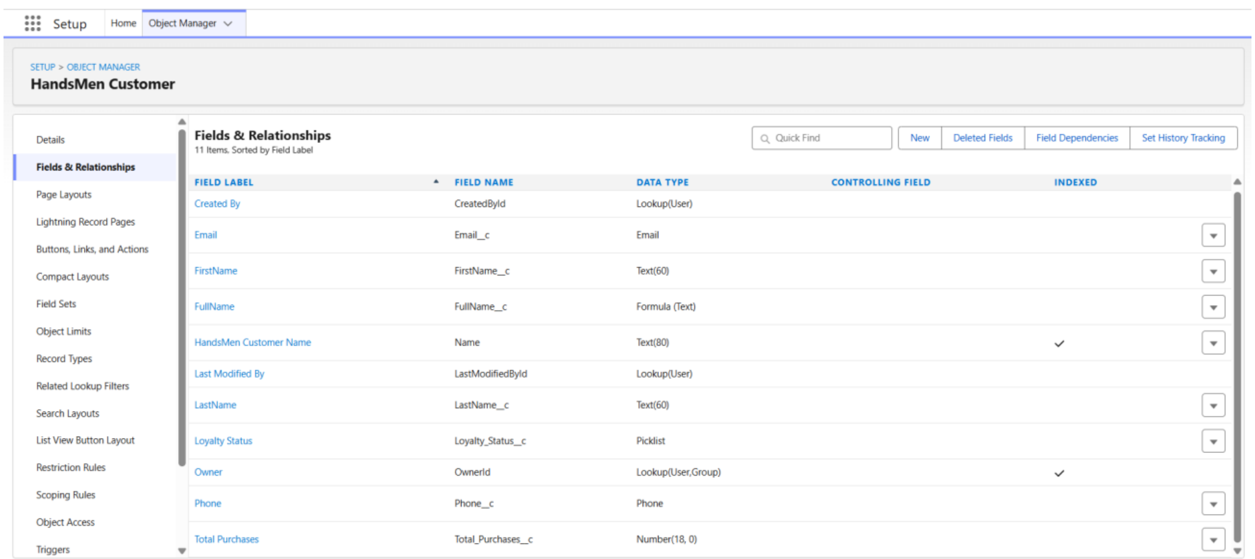



*Figure: HandsMen Customers (Fields)*

### 1.2. HandsMen Customers' Validation Rules

**Error Condition Formula:** NOT CONTAINS( Email__c , "@gmail.com")

The purpose of this validation rule, named **Email**, is to **enforce a specific email domain standard** for all records on the HandsMen_Customer__c custom object. It prevents a user from successfully saving a customer record if the value entered in the **Email** field does not contain the substring "@gmail.com". The **Error Condition Formula** used to check this condition is NOT CONTAINS(Email, "@gmail.com"), which evaluates to True (triggering the error) when the specified domain is missing. If the rule is triggered, the user is presented with the error message **"Please fill Correct Gmail"** at the **Top of the Page**, ensuring data integrity before the record is saved.

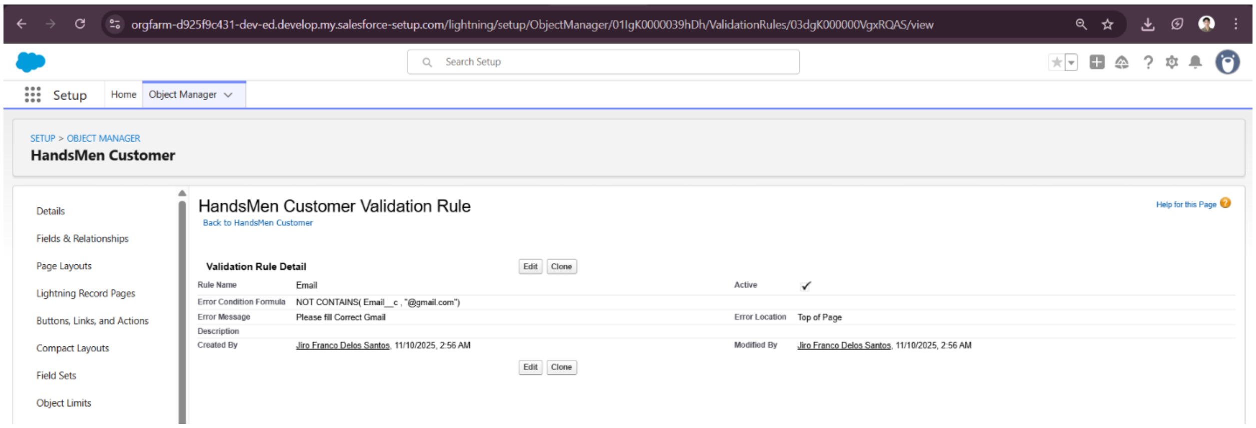



*Figure: HandsMen Customers (Validation Rule)*

## 2. **Custom Object:** HandsMen_Order__c

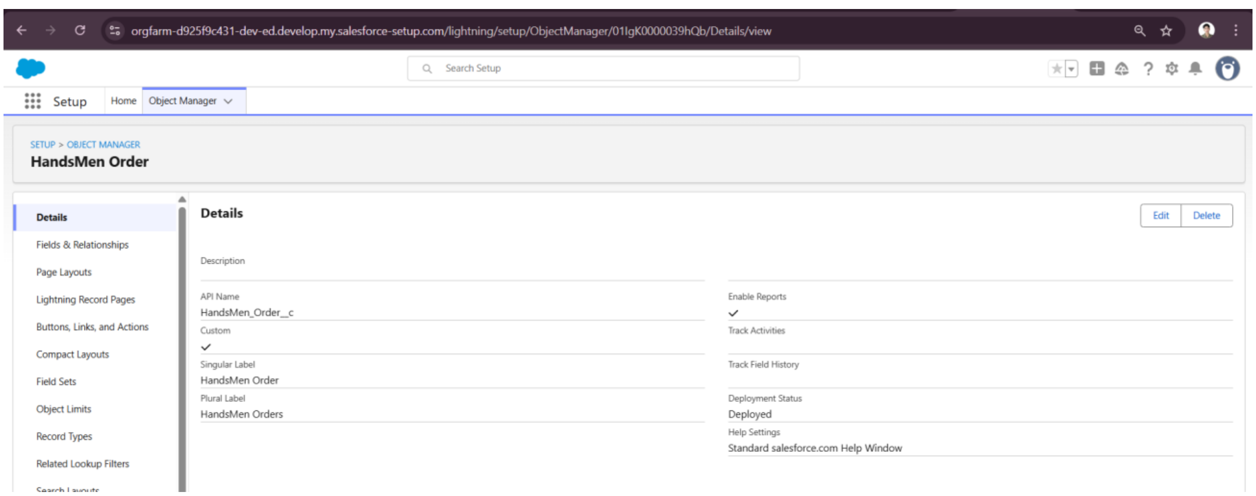



*Figure: HandsMen Orders (Custom Object)*

### 2.1. Fields Created for HandsMen Orders

- HandsMen OrderNumber (Name - Auto Number) - Indexed
- HandsMen Customer (HandsMen_Customer__c - Lookup to HandsMen Customer) - Required, Indexed
- Customer Email (Customer_Email__c - Email)
- HandsMen Product (HandsMen_Product__c - Lookup to HandsMen Product) - Required,

Indexed

• Quantity (Quantity__c - Number, 18, 0)
• Status (Status__c - Picklist: Pending, Confirmed, Shipped, Delivered, Cancelled)
• Total Amount (Total_Amount__c - Number, 18, 0) - Auto-calculated
• Owner (OwnerId - Lookup User,Group) - Indexed
• Created By (CreatedById - Lookup User)
• Last Modified By (LastModifiedById - Lookup User)

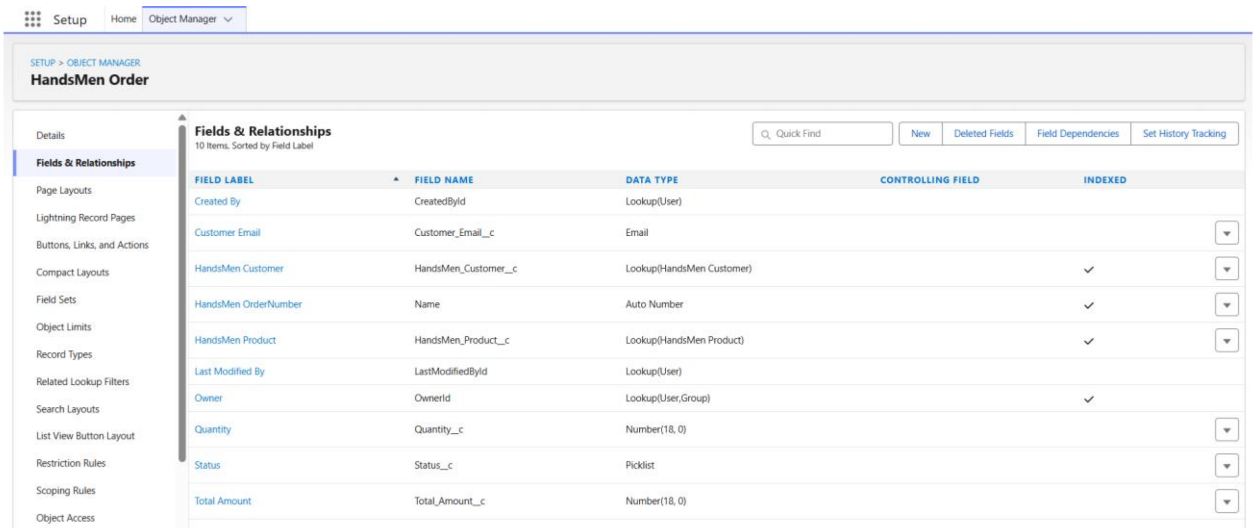



*Figure: HandsMen Orders (Fields)*

### 2.2. HandsMen Orders' Validation Rules

**Error Condition Formula:** Total_Amount__c <= 0

The purpose of this validation rule, named **Total_Amount**, is to ensure that all transactional records on the HandsMen_Order__c custom object represent a positive financial value. The rule is triggered when the **Error Condition Formula** evaluates to True, which occurs if the custom field Total_Amount__c is less than or equal to zero. This constraint prevents the creation or saving of any order record that erroneously lists a zero or negative total, which is critical for maintaining accurate financial and inventory records[1]. If the condition is met, the system displays the error message **"Please Enter Correct Amount"** at the **Top of Page**, blocking the user from saving the invalid order.

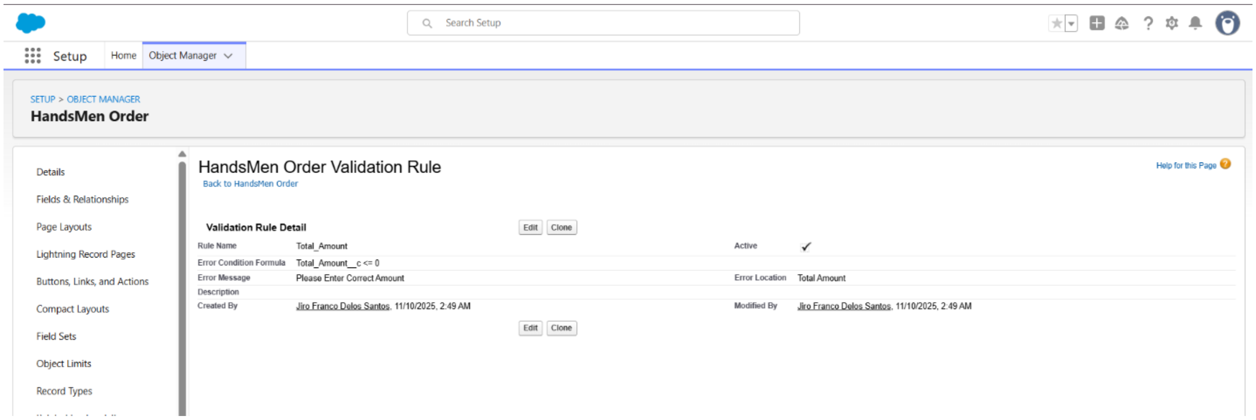



*Figure: HandsMen Orders (Validation Rule)*

**3. Custom Object:** HandsMen_Product__c

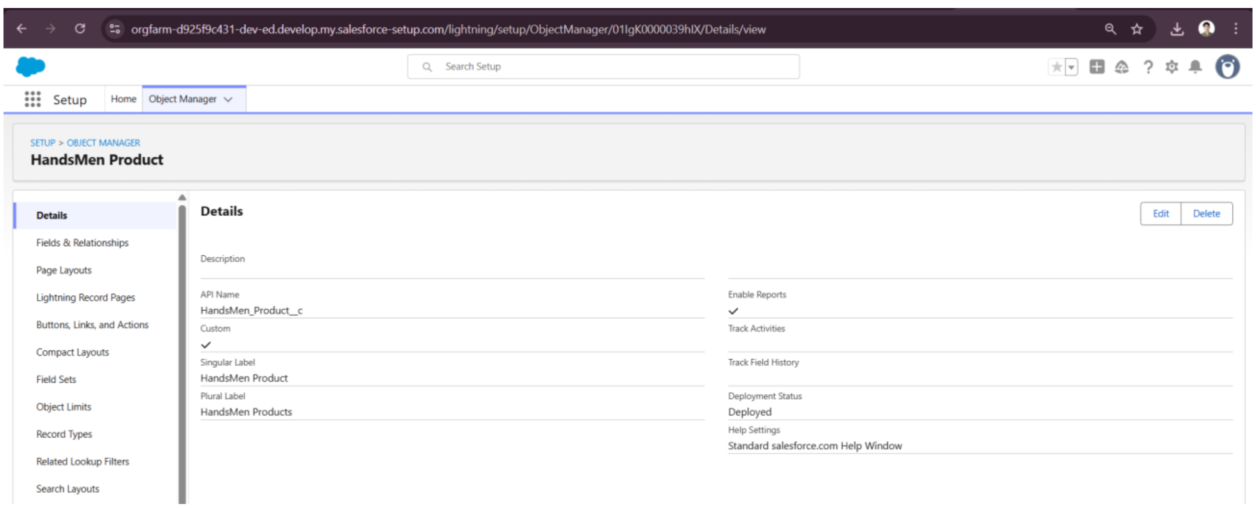



*Figure: HandsMen Products (Custom Object)*

**3.1. Fields Created for HandsMen Products**

- HandsMen Product Name (Name - Text, 80) - Required, Indexed
- Price (Price__c - Currency, 18, 0)
- SKU (SKU__c - Text, 60)
- Stock Quantity (Stock_Quantity__c - Number, 18, 0)
- Order (HandsMen_Order__c - Lookup to HandsMen Order) - Indexed
- Owner (OwnerId - Lookup User,Group) - Indexed
- Created By (CreatedById - Lookup User)
- Last Modified By (LastModifiedById - Lookup User)

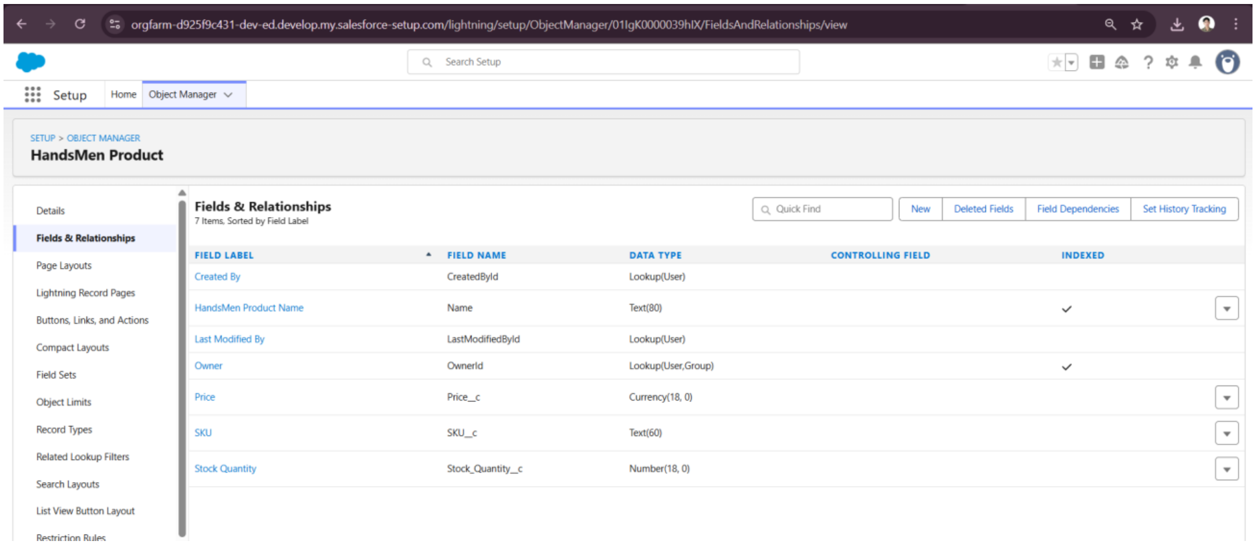


*Figure: HandsMen Products (Fields)*

### 3.2. HandsMen Products' Validation Rules

**Error Condition Formula:** Stock_Quantity__c <= 0

This validation rule, named **Stock_Quantity**, is designed to maintain the integrity of inventory records on the HandsMen_Product__c custom object by ensuring a positive stock count. The rule is activated when the **Error Condition Formula** evaluates to True, which occurs if the custom field Stock_Quantity__c contains a value that is less than or equal to zero. This constraint prevents users from saving product records with an impossible or nonsensical stock level, supporting accurate inventory management. If the condition is met, the user is blocked from saving the record and shown the error message **" the quantity count is never less than zero."** at the **Top of Page**.

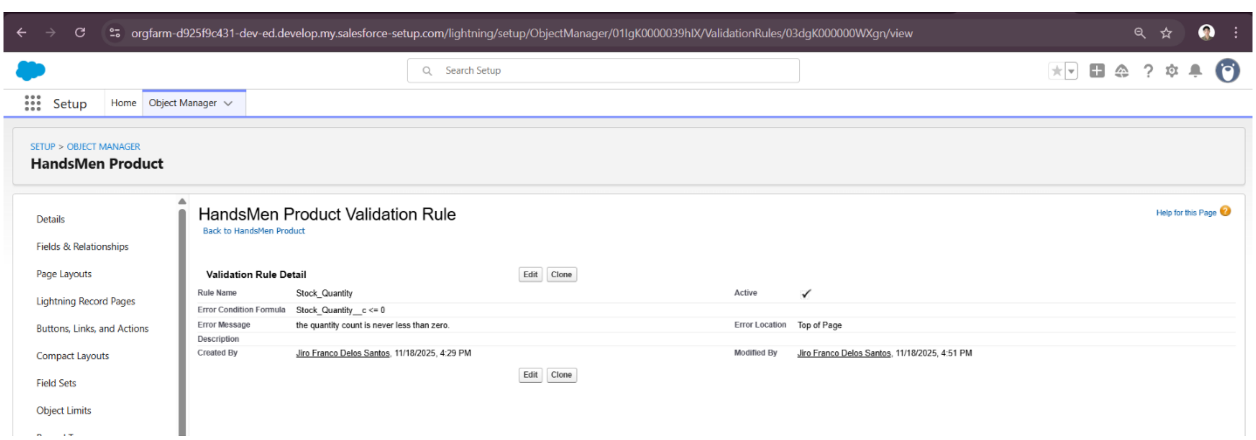


*Figure: HandsMen Products (Validation Rules)*

## 4. **Custom Object:** Inventory__c

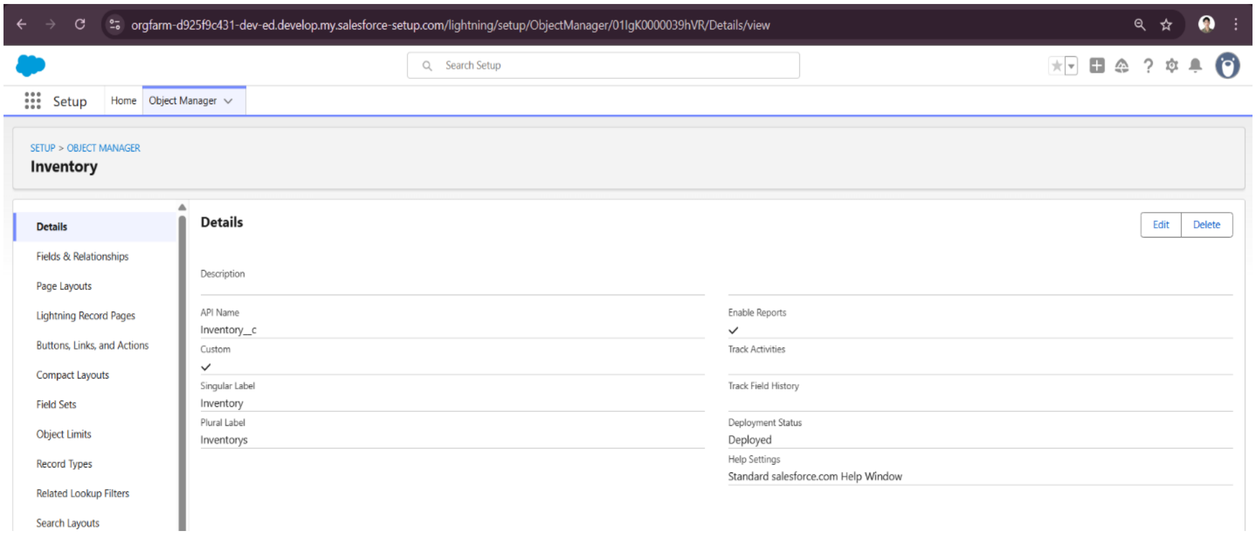



*Figure: Inventory (Custom Object)*

### 4.1. Fields Created for Inventory

- Inventory Number (Name - Auto Number) - Indexed
- Product (HandsMen_Product__c - Master-Detail to HandsMen Product) - Required, Indexed
- Stock Quantity (Stock_Quantity__c - Number, 18, 0)
- Stock Status (Stock_Status__c - Formula Text)
- Warehouse (Warehouse__c - Text, 60)
- Created By (CreatedById - Lookup User)
- Last Modified By (LastModifiedById - Lookup User)

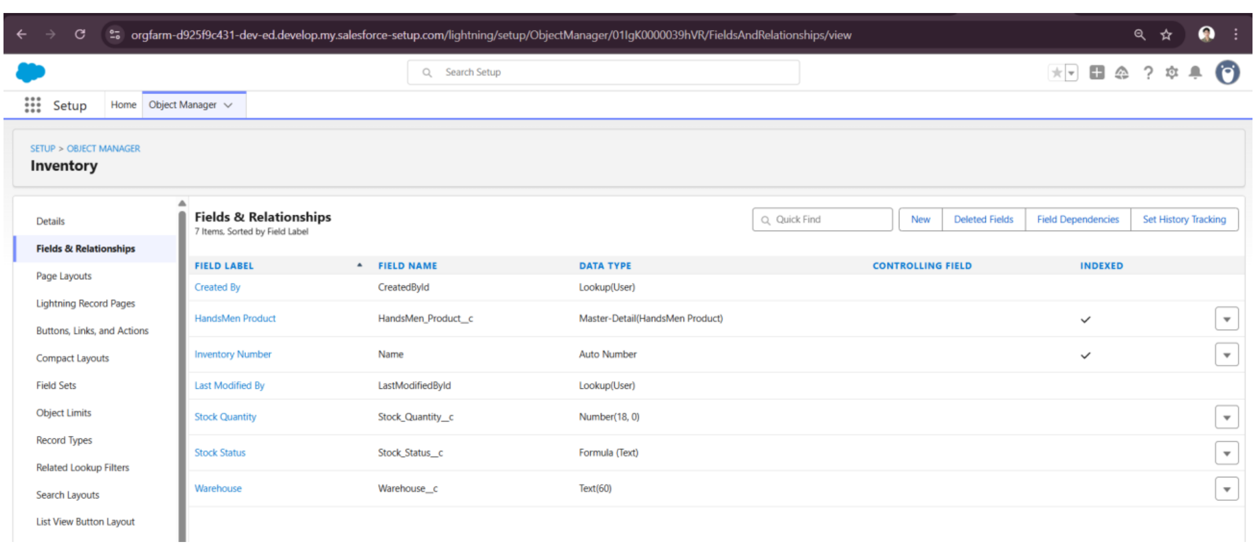


*Figure: Inventory (Fields)*

### 4.2. Inventory's Validation Rules

**Error Condition Formula:** Stock_Quantity__c <= 0

This validation rule, named **Stock_Quantity**, is implemented on the custom **Inventory__c** object to enforce that all physical stock counts are strictly positive. The rule activates when the **Error Condition Formula** evaluates to True, which specifically occurs if the custom field Stock_Quantity__c is zero or a negative number. This constraint is crucial for maintaining real-time tracking of item quantities and safety stock thresholds, which serves as the source for the stock alert automation. If the condition is met, the system prevents the record from being saved and displays the error message **"the inventory count is never less than zero."** at the **Top of Page**.

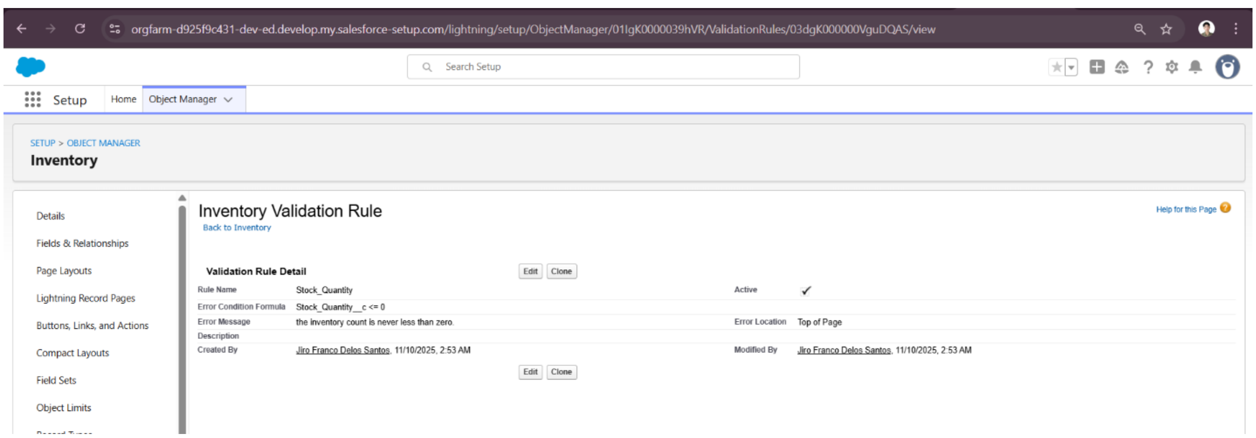
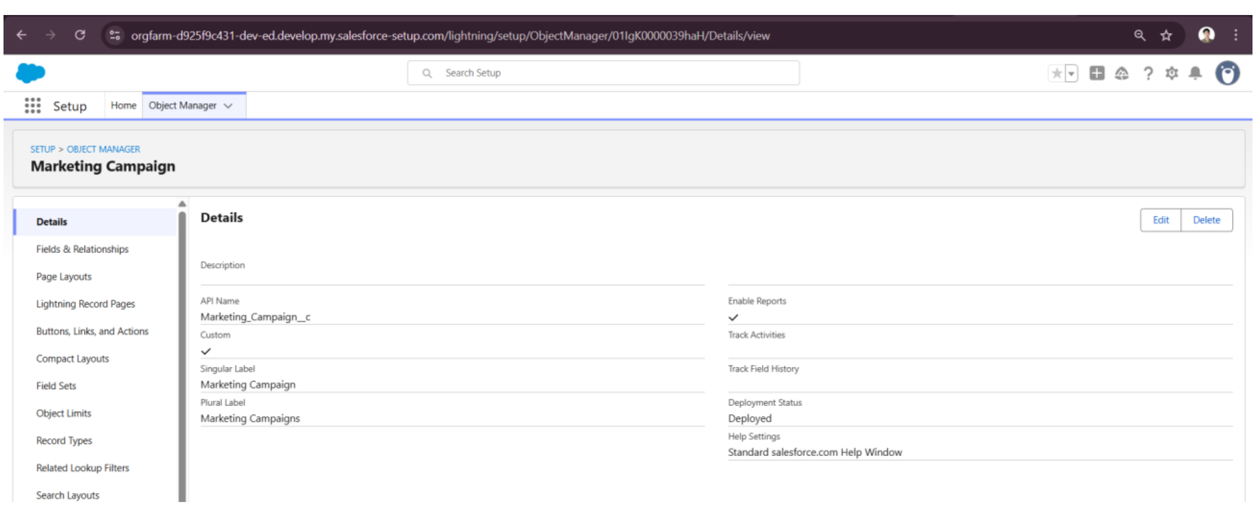



*Figure: Inventory (Validation Rule)*

## 5. Custom Object: Marketing_Campaign__c



### 5.1. Fields Created for Marketing Campaign

- Marketing Campaign Number (Name - Auto Number) - Indexed
- HandsMen Customer (HandsMen_Customer__c - Lookup to HandsMen Customer) - Indexed
- Start Date (Start_Date__c - Date)
- End Date (End_Date__c - Date)
- Owner (OwnerId - Lookup User,Group) - Indexed
- Created By (CreatedById - Lookup User)
- Last Modified By (LastModifiedById - Lookup User)

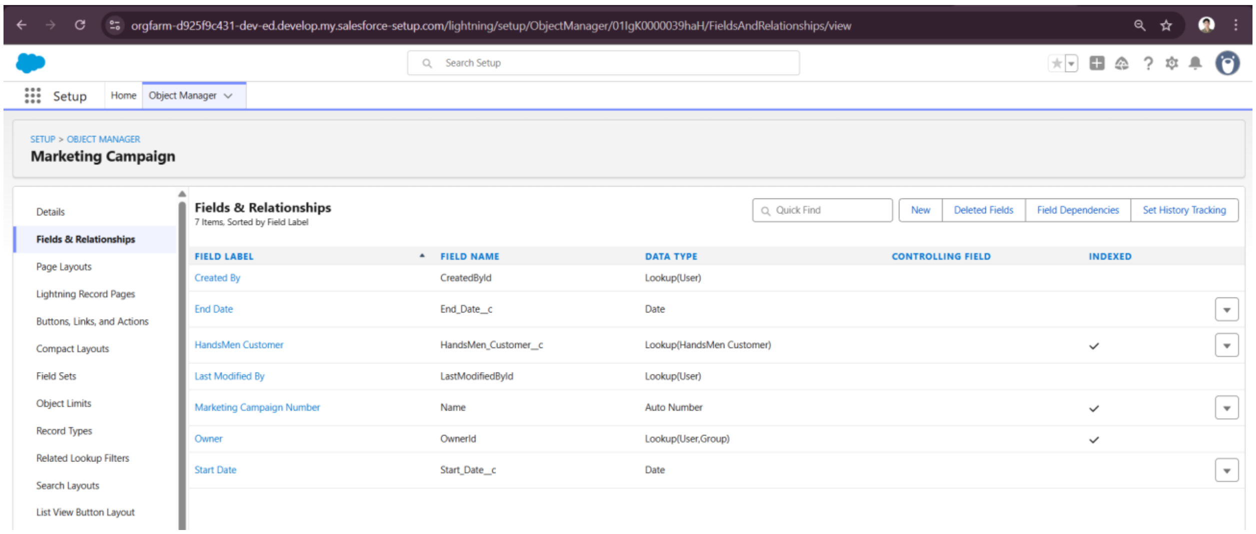



*Figure: Marketing Campaign (Field)*

## Automation: Workflow Rules, Process Builder, Flows, Approval Process

**Flows** are the cornerstone of the declarative automation strategy for the HandsMen Threads project, serving as the modern, powerful tool used to implement intricate business logic without writing Apex code. This advanced functionality allows the system to execute sophisticated automations, replacing older legacy tools like Workflow Rules and Process Builder for enhanced efficiency and future scalability. Specifically, Flows will be deployed to realize two critical project deliverables: the automated post-order confirmations to ensure timely customer engagement and the proactive stock alerts that notify.

5. **Order Confirmation Flow:** Implemented as a Record-Triggered Flow on the HandsMen_Order__c object, designed to execute immediately whenever a new order record is successfully created. The Flow's primary function is to instantly transmit a customized, automated order acknowledgment email to the associated customer,

confirming the transaction details. This critical, real-time communication enhances immediate transparency for the customer and directly contributes to the project's goal of boosting customer satisfaction and brand loyalty.

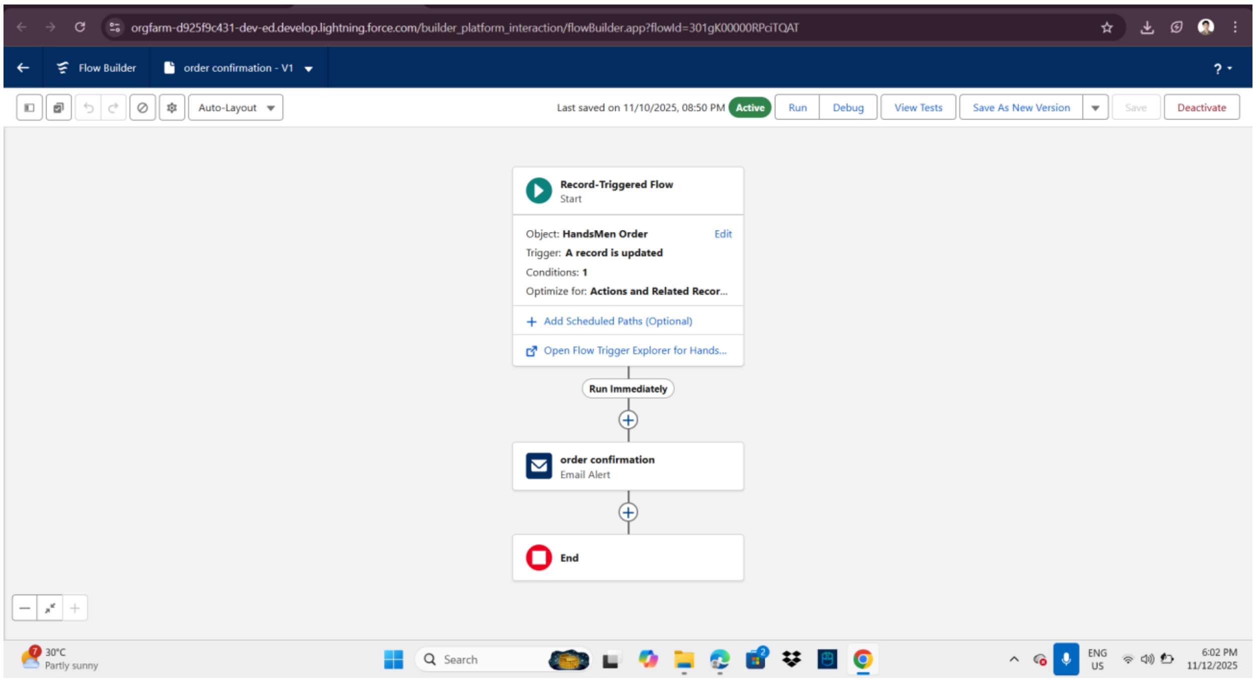



*Figure: Order Confirmation Flow*

6. **Customer Assessment Flow:** Implemented as a Scheduled Flow designed to execute automatically during off-peak hours, specifically targeting midnight processing to ensure minimal impact on the operational environment. Its core responsibility is to systematically assess every HandsMen_Customer__c record by evaluating the cumulative purchase values and history against predefined criteria for the loyalty tiers. By dynamically updating the customer's Loyalty_Status__c, this automation guarantees that the retention program is always current and enables the Marketing team to deploy personalized strategies to drive repeat business.

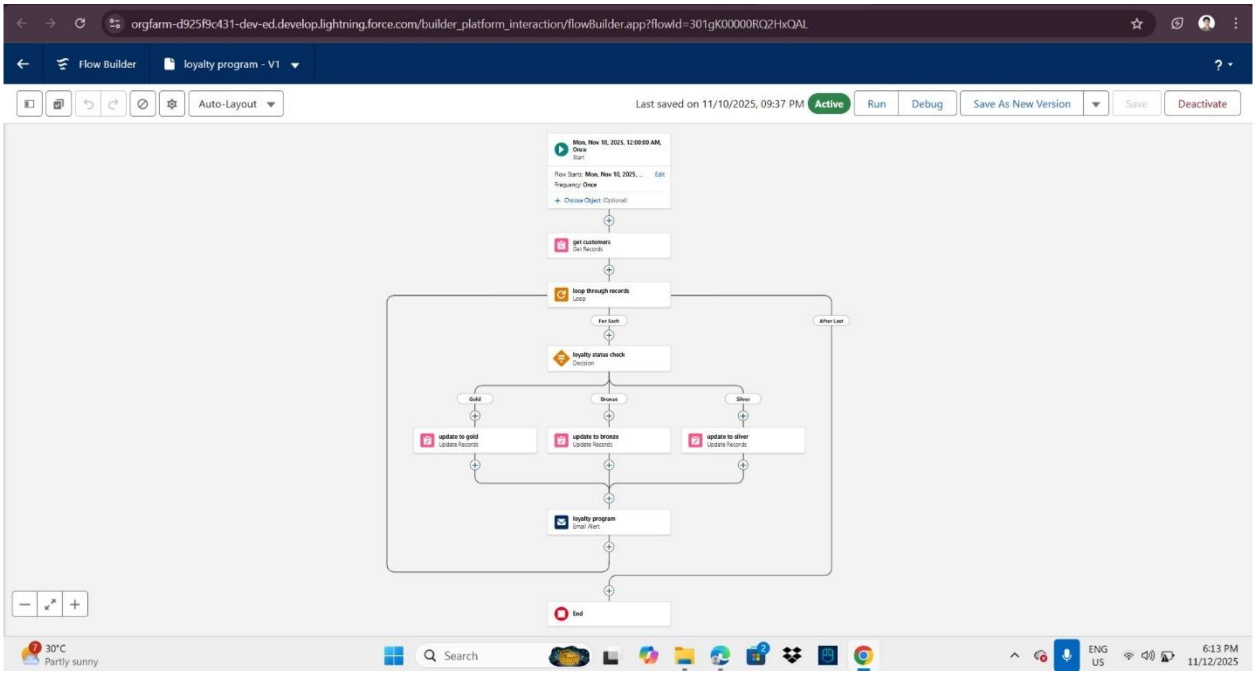



*Figure: Customer Assessment Flow*

7. **Low Stock Alert Flow:** Implemented as a Record-Triggered Flow that continuously monitors the Inventory__c object for any changes to the Stock_Quantity__c field. When the quantity drops below the critical threshold of five units , the Flow immediately executes a Proactive Stock Alert, delivering an essential notification to the Warehouse staff to ensure timely replenishment and prevent detrimental stockouts.

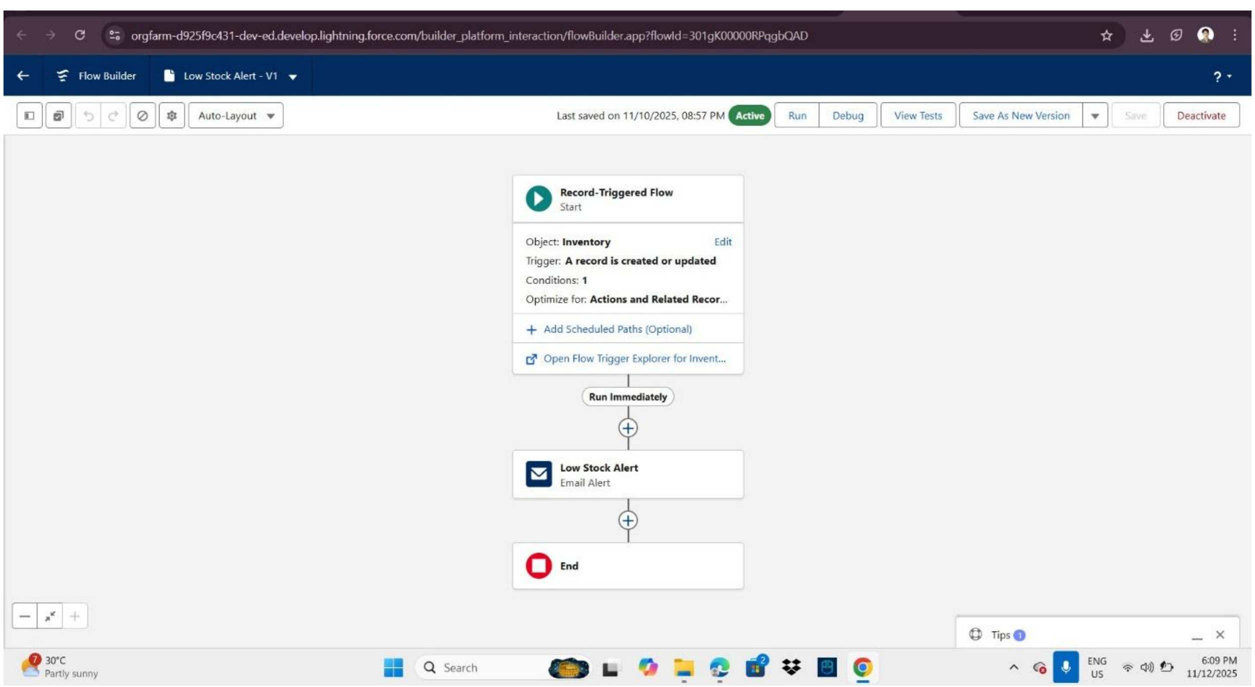


*Figure: Low Stock Alert Flow*

8. **Customer Confirmation Email**: The successful update of the customer's new loyalty status immediately triggers an automated email notification to be sent directly to them**,** ensuring they are informed of their tier change and enabling the Marketing team to deploy timely, personalized retention strategies.

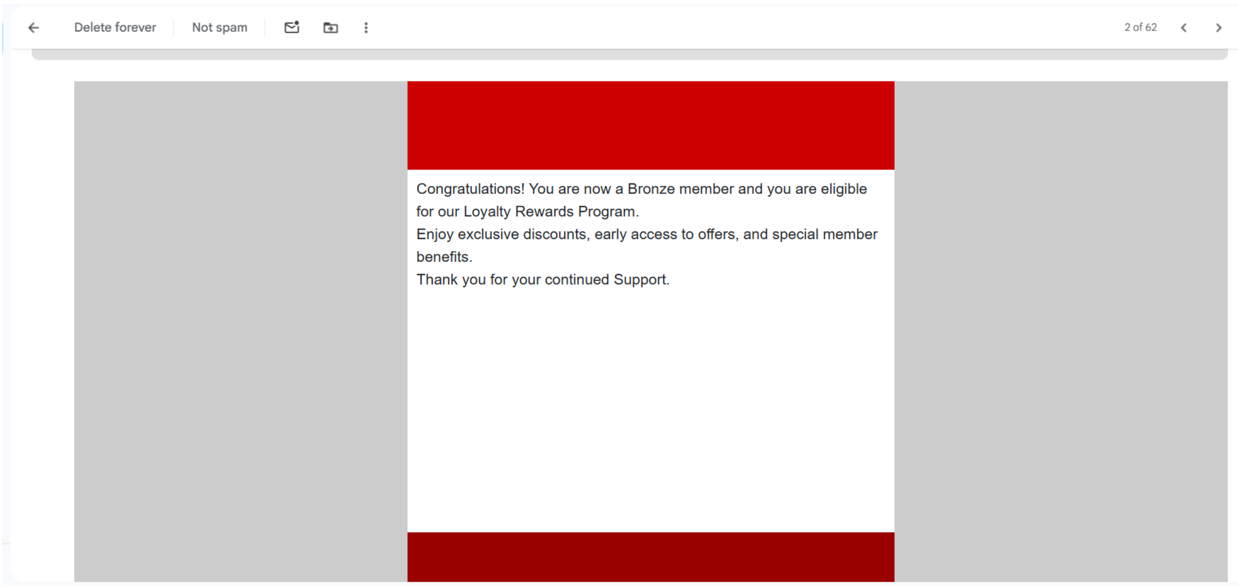



*Figure: Customer Confirmation Email*

9. **Order Confirmation Email:** The customer will instantly receive a customized, automated order acknowledgment email, which confirms the transaction details and serves as a critical, real-time communication that directly enhances customer satisfaction and brand loyalty.

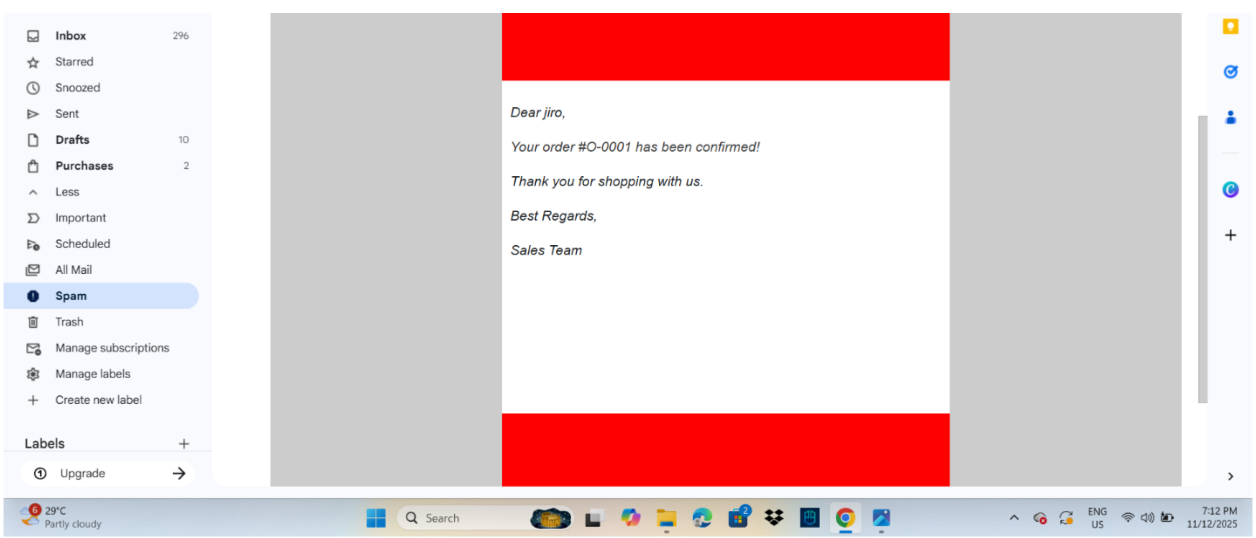


*Figure: Order Confirmation Email*

10. **Low Stock Alert Email:** When the quantity drops below the critical threshold of five units, the Flow immediately executes a Proactive Stock Alert, triggering an essential email notification to the Inventory Manager and warehouse staff to ensure timely replenishment and prevent detrimental stockouts.

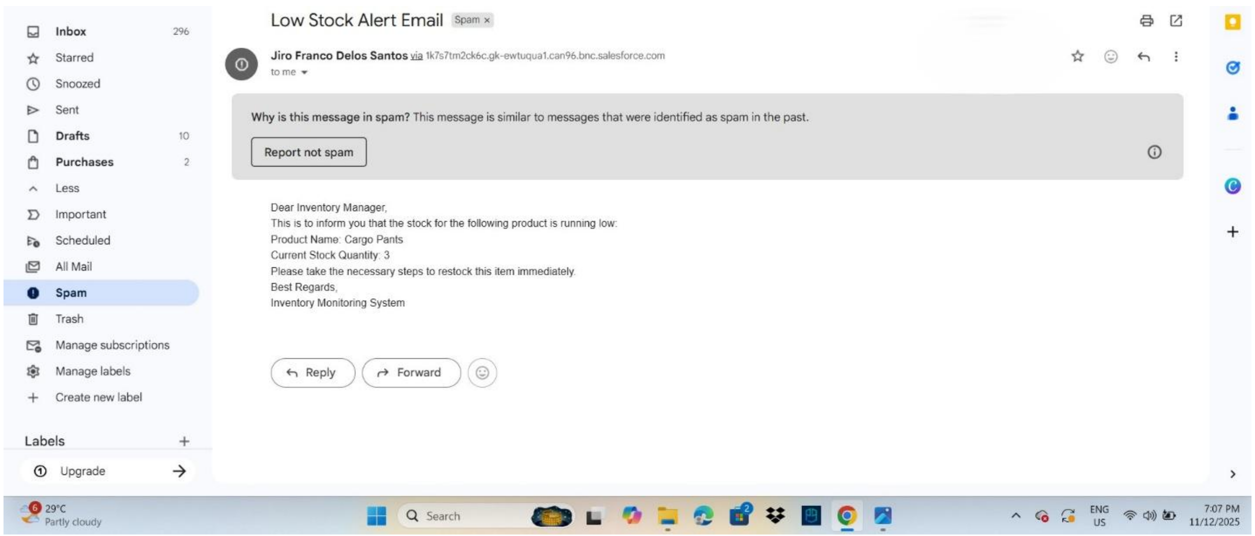



*Figure: Low Stock Alert Email*

The **Approval Process** is a powerful automation tool in Salesforce designed to enforce an internal, multi-step sequence for reviewing and authorizing critical records. Its primary purpose in the HandsMen Threads project is to ensure governance and oversight over high-risk or high-value actions, such as managerial sign-off required before finalizing an order with a large discount. By automating the routing of records to specific approvers and executing automated actions upon approval or rejection, the process guarantees that all crucial business policies are formally enforced before a record is committed.

1. **HandsMen Customer: Customer Status Approval**

| Detail | Purpose |
|---|---|
| **Goal** | Require **manual approval** before a new customer's Loyalty_Status__c is set to the highest tier ("Gold") when they have |

| Detail | Purpose |
|---|---|
| | not yet met the automated purchase criteria (e.g., for executive overrides). |
| **Entry Criteria** | Loyalty_Status__c = 'Gold' AND Total_Purchases__c >= 5000 (Triggers when status is manually set high before earned). |
| **Approver** | Marketing Manager Profile |
| **Actions** | **On Submission:** Set Loyalty_Status__c to "Pending Gold Review." **On Approval:** Set Loyalty_Status__c to "Gold." **On Rejection:** Set Loyalty_Status__c back to "Bronze" |

2. **HandsMen Order: High-Value Order Review**

| Component | Setting/Value | Rationale |
|---|---|---|
| **Target Object** | HandsMen_Order__c | The process must run on the transactional record. |
| **Approver** | Sales Manager (Based on Role Hierarchy) | The appropriate level of management for financial oversight. |
| **Error Location** | Top of Page | Provides immediate feedback to the user upon submission. |

3.

| Detail | Purpose |
|---|---|
| **Goal** | **Control critical changes to the safety stock level** (Stock_Quantity__c), as this directly impacts the **Proactive Stock Alert** Flow, requiring expert validation before the new threshold is active. |
| **Entry Criteria** | Stock_Quantity__c field is changed AND Stock_Quantity__c <= 0 (Any change that modifies the critical alert logic). |
| **Approver** | Inventory Manager Profile (Read/Write on Inventory/Product) |

**Apex Classes and Triggers** represent the programming language, deployed when standard configuration tools and declarative automations cannot efficiently accommodate specific complex or high-volume business logic requirements. This programmatic layer is essential for processes demanding extensive control, such as preventing governor limits during bulk record manipulation or integrating intricate business calculations that Flows cannot handle. Our custom triggers, essential for real-time transactional integrity, include the **Customer Loyalty Update Trigger** (to instantly adjust loyalty status upon order completion), the **Stock Deduction Trigger** (to accurately reduce inventory levels post-order), and the **Order Total Trigger** (to ensure precise calculation of the final order amount). Furthermore, Apex will be utilized to develop the **Asynchronous Batch Job** needed to execute the nightly, high-volume **Scheduled Bulk Order Updates**, ensuring accurate financial and inventory reconciliation outside of peak business hours.

1. **Apex Trigger:** CustomerLoyaltyUpdateTrigger
   **Purpose:** The Customer Loyalty Update Trigger dynamically updates a customer's loyalty tier based on their cumulative confirmed order totals in real-time. This trigger executes immediately following any new or updated order to aggregate the total purchase value tied to that customer. By analyzing this running total against predefined thresholds, the system ensures the customer's status is always accurately reflected, which is vital for enabling timely, personalized marketing, and retention strategies.

```
trigger CustomerLoyaltyUpdateTrigger on HandsMen_Order__c (after insert, after update) {

    Set<Id> customerIds = new Set<Id>();

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

    List<HandsMen_Customer__c> customersToUpdate = new List<HandsMen_Customer__c>();

    for (Id customerId : customerIds) {

        AggregateResult[] results = [
            SELECT SUM(Total_Amount__c) totalSum
            FROM HandsMen_Order__c
            WHERE HandsMen_Customer__c = :customerId
        ];

        Decimal totalPurchases = 0;
        if (results[0].get('totalSum') != null) {
            totalPurchases = (Decimal)results[0].get('totalSum');
        }

        String newLoyaltyStatus;
        if (totalPurchases >= 1000) {
            newLoyaltyStatus = 'Gold';
        } else if (totalPurchases >= 500) {
            newLoyaltyStatus = 'Silver';
        } else {
            newLoyaltyStatus = 'Bronze';
        }

        HandsMen_Customer__c customer = [
            SELECT Id, Loyalty_Status__c
            FROM HandsMen_Customer__c
            WHERE Id = :customerId
            LIMIT 1
        ];

        if (customer.Loyalty_Status__c != newLoyaltyStatus) {
            customer.Loyalty_Status__c = newLoyaltyStatus;
            customersToUpdate.add(customer);
        }
    }

    if (!customersToUpdate.isEmpty()) {
        update customersToUpdate;
    }
}
```

*Figure: CustomerLoyaltyUpdateTrigger (Loyalty Status Automation)*

2. **Apex Trigger:** OrderTotalTrigger
   **Purpose:** It executes before an HandsMen_Order__c record is inserted or updated, guaranteeing the accurate calculation of the order's final cost. Its primary purpose is to dynamically calculate and set the Total_Amount__c field by querying the corresponding HandsMen_Product__c record to retrieve the unit Price__c. By multiplying the order's Quantity__c by the product's Price__c within the trigger

execution context, it enforces strict transactional integrity and supports the project's requirement for accurate and current financial records.

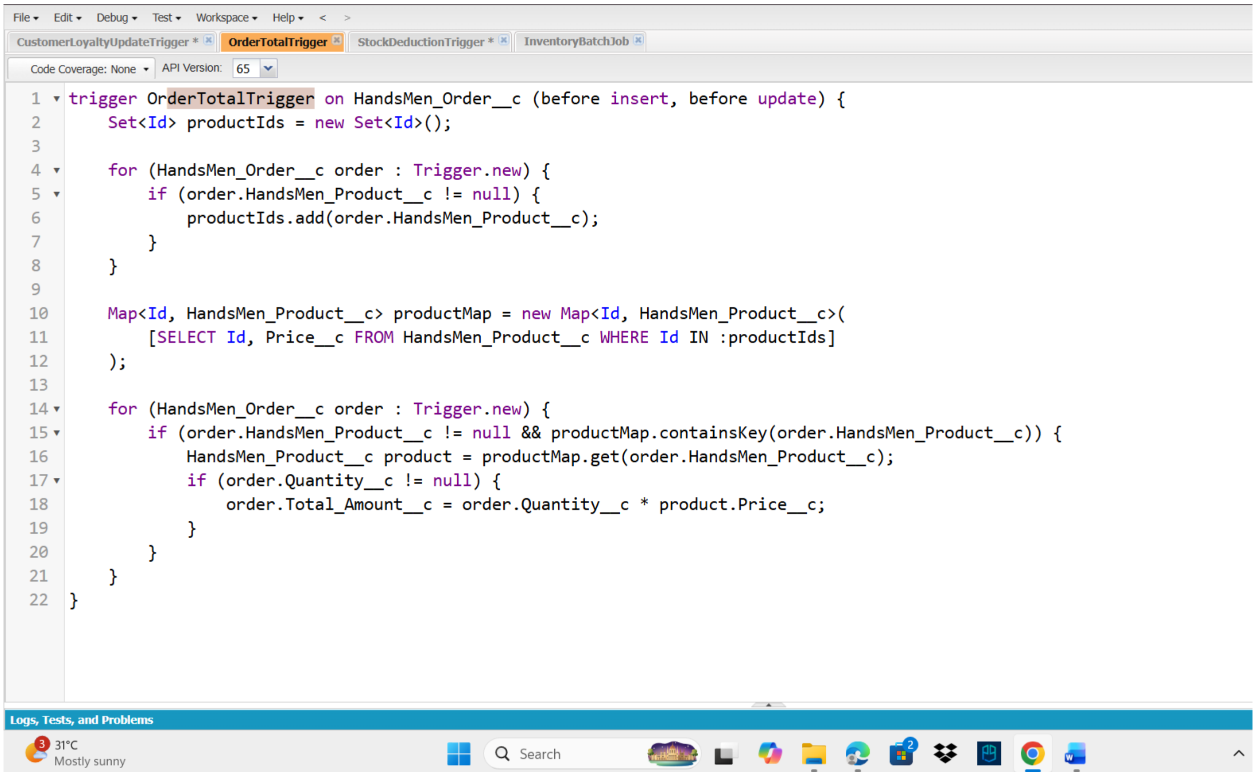



*Figure: OrderTotalTrigger (Order Total Calculation)*

3. **Apex Trigger:** StockDeductionTrigger
   **Purpose:** It executes after the successful creation of a HandsMen_Order__c record, guaranteeing the accuracy of inventory levels. Its purpose is to efficiently deduct the purchased quantity from the corresponding product's available stock on the Inventory__c object. By ensuring that inventory is immediately reconciled upon order confirmation, this trigger maintains real-time stock availability, which is essential for accurate fulfillment and the **Proactive Stock Alert** automation.

```
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;

    // Query related inventories based on product
    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;
    }
}
```

*Figure: StockDeductionTrigger*

4. **Apex Trigger:** InventoryBatchJob
   **Purpose:** Implemented as an **Asynchronous Apex** class dedicated to executing high-volume, complex data operations outside of standard business hours. Its purpose is to perform essential data maintenance, such as the comprehensive recalculation of customer loyalty points or the alignment of internal stock levels with external warehouse systems. This scheduled processing (e.g., at 2 AM) safely bypasses Salesforce's governor limits, ensuring data integrity and providing up-to-date, accurate records for operational decision-making at the start of the business day.

```
global class InventoryBatchJob implements Database.Batchable<SObject>, Schedulable {

global Database.QueryLocator start(Database.BatchableContext BC) {

return Database.getQueryLocator(

'SELECT Id, Stock_Quantity__c FROM Product__c WHERE Stock_Quantity__c < 10'

);

}

global void execute(Database.BatchableContext BC, List<SObject> records) {

List<HandsMen_Product__c> productsToUpdate = new List<HandsMen_Product__c>();

// Cast SObject list to Product__c list

for (SObject record : records) {

HandsMen_Product__c product = (HandsMen_Product__c) record;

product.Stock_Quantity__c += 50; // Restock logic

productsToUpdate.add(product);

}

if (!productsToUpdate.isEmpty()) {

try {

update productsToUpdate;

} catch (DmlException e) {

System.debug('Error updating inventory: ' + e.getMessage());

}

}

}

global void finish(Database.BatchableContext BC) {

System.debug('Inventory Sync Completed');

}

// Scheduler Method

global void execute(SchedulableContext SC) {

InventoryBatchJob batchJob = new InventoryBatchJob();

Database.executeBatch(batchJob, 200);

}

}
```

*Figure: InventoryBatchJob*

## Phase 3: UI/UX Development & Customization

This phase is dedicated to translating the foundational data model and business logic into a cohesive and intuitive user experience (UI/UX) within the Salesforce Lightning environment. This phase begins with the essential setup of the Lightning Application via the App Manager, providing the specific container and branding for the developed solution. Following the application setup, attention shifts to core interface customization, which includes refining the user interaction through tailored Page Layouts and the strategic implementation of Dynamic Forms to surface only necessary information contextually. Essential to any successful enterprise application is secure and controlled access; therefore, comprehensive User Management will be established to define roles and permissions. Finally, the phase concludes by empowering end-users and stakeholders with actionable intelligence through the configuration of Reports and Dashboards. Furthermore, the commitment to best-in-class user experience is demonstrated through the optional development of Lightning Web Components (LWC), integrating custom, modern interactivity where standard platform features require extension. As proof of concept and execution, comprehensive screenshots for each Salesforce concept developed are mandatory.

## Lightning App Setup through App Manager

### Custom Lightning App: "HandsMen Threads"

Configuration Details:

- App Type: Lightning App
- App Logo: Custom HandsMen Threads logo uploaded

Tabs Included:

- HandsMen Customers
- HandsMen Orders
- HandsMen Products
- Inventorys
- Marketing Campaigns
- Reports
- Dashboards
- Accounts
- Contacts

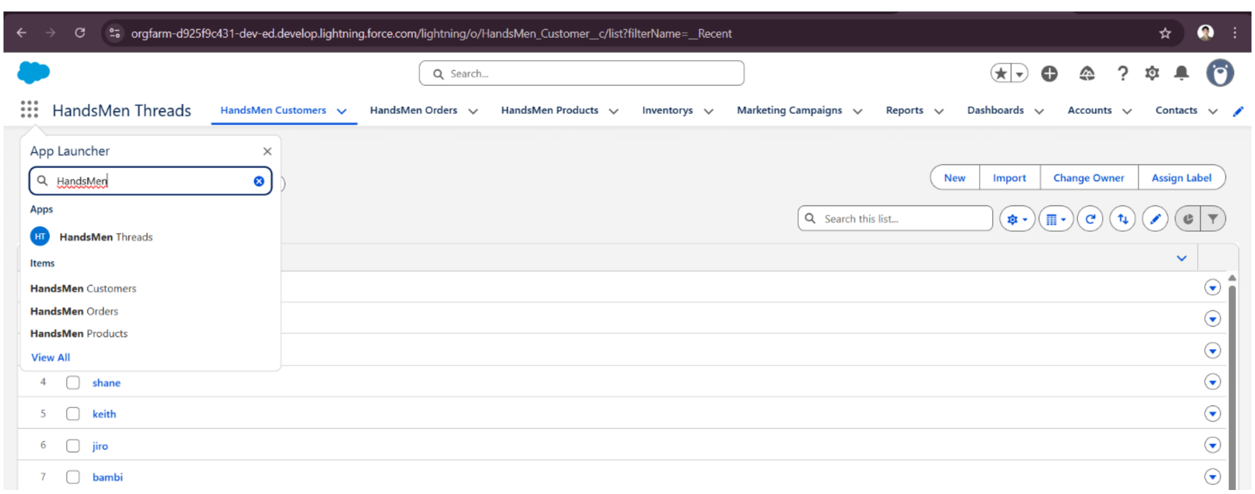



*Figure: Lightning App Setup through App Manager*

# Page Layouts and Dynamic Forms

## **Page Layout:** HandsMen Customer

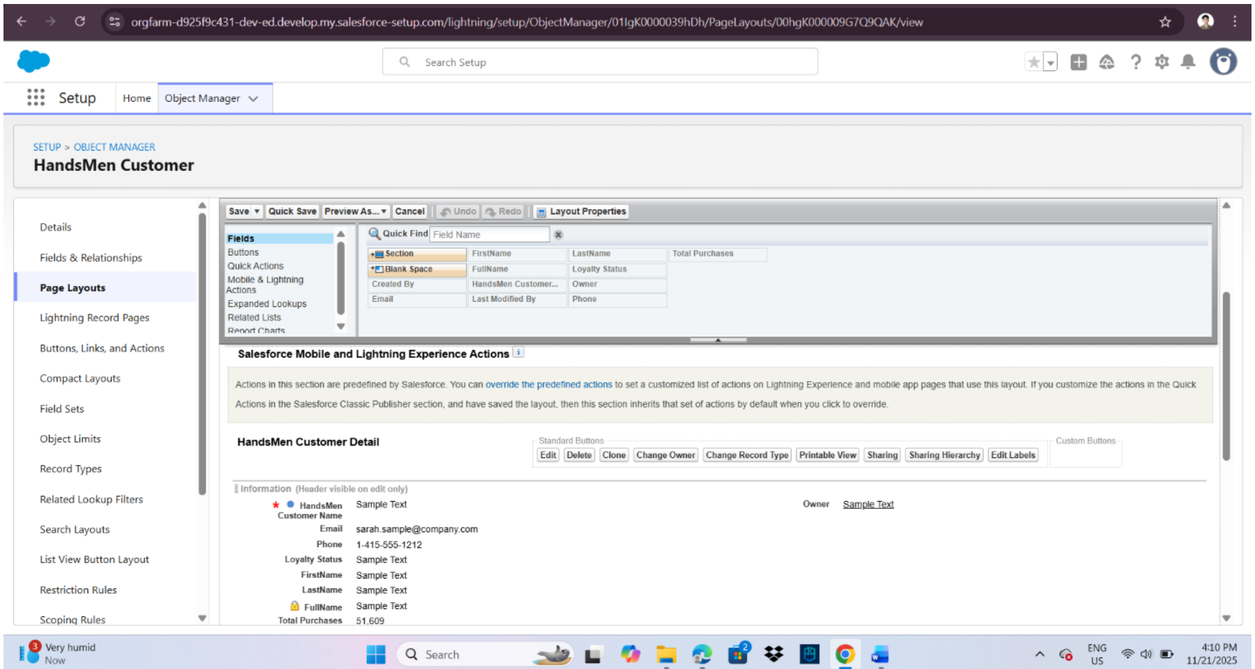


*Figure: HandsMen Customer Page Layout*

## **Dynamic Forms:** HandsMen Customer

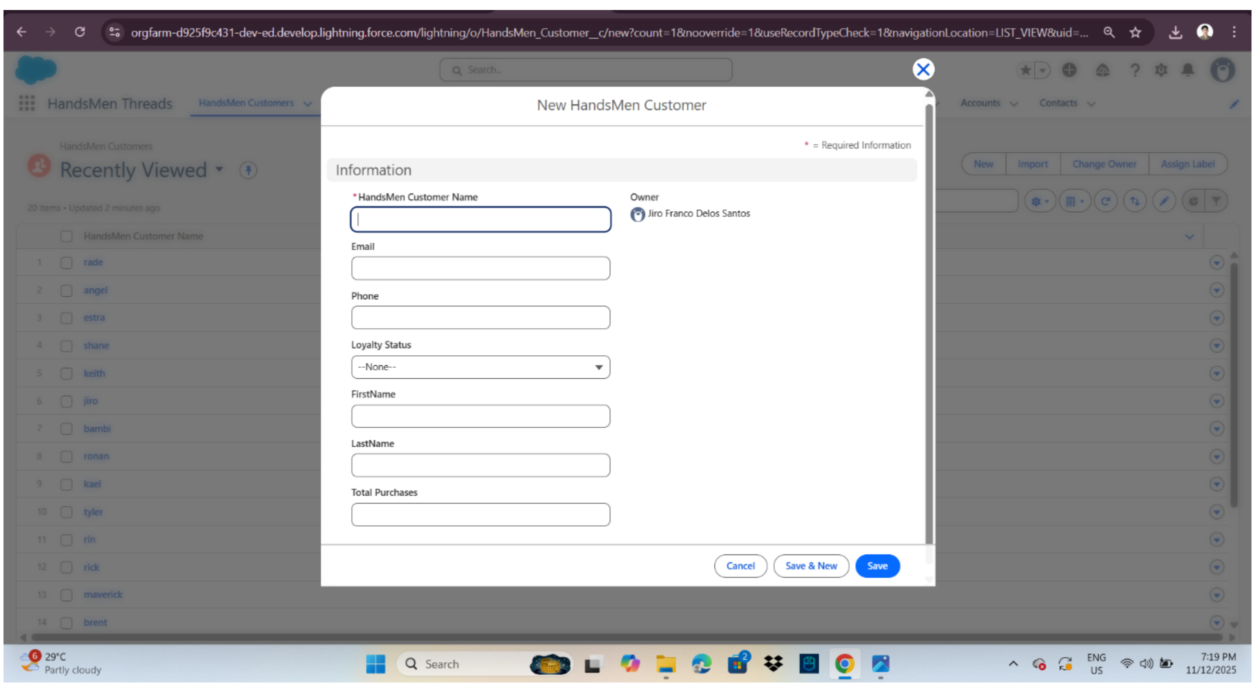


*Figure: HandsMen Customer Forms*

**Page Layout:** HandsMen Order

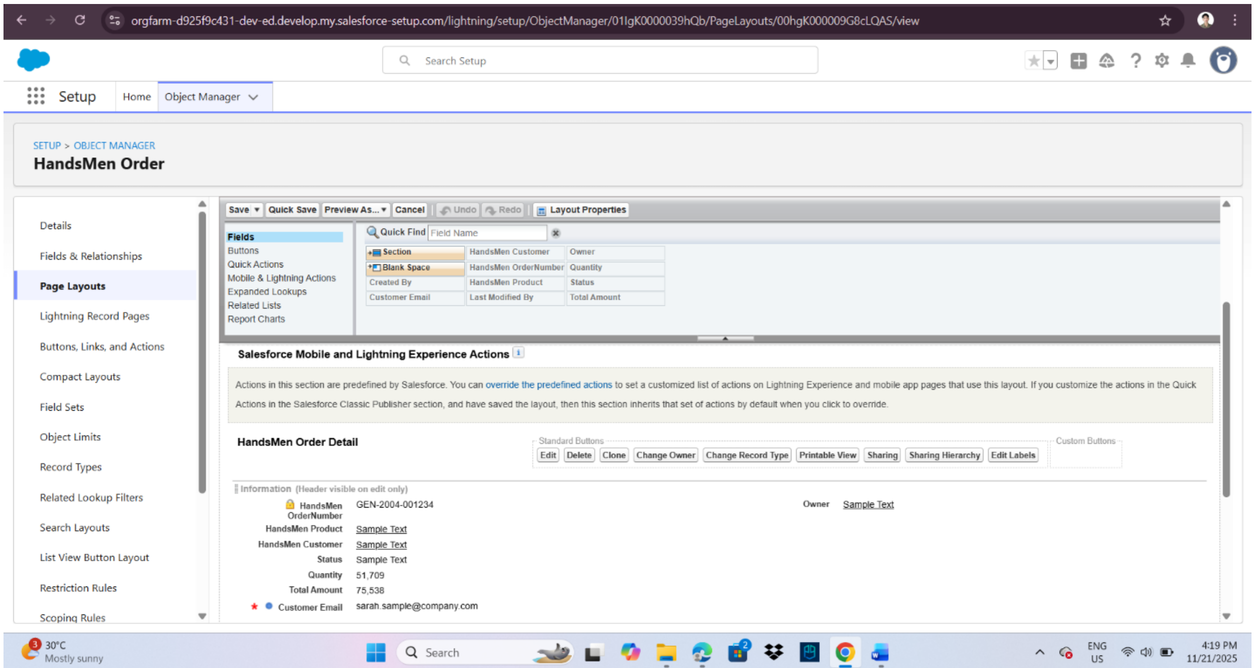



*Figure: HandsMen Order Page Layout*

**Dynamic Forms:** HandsMen Order

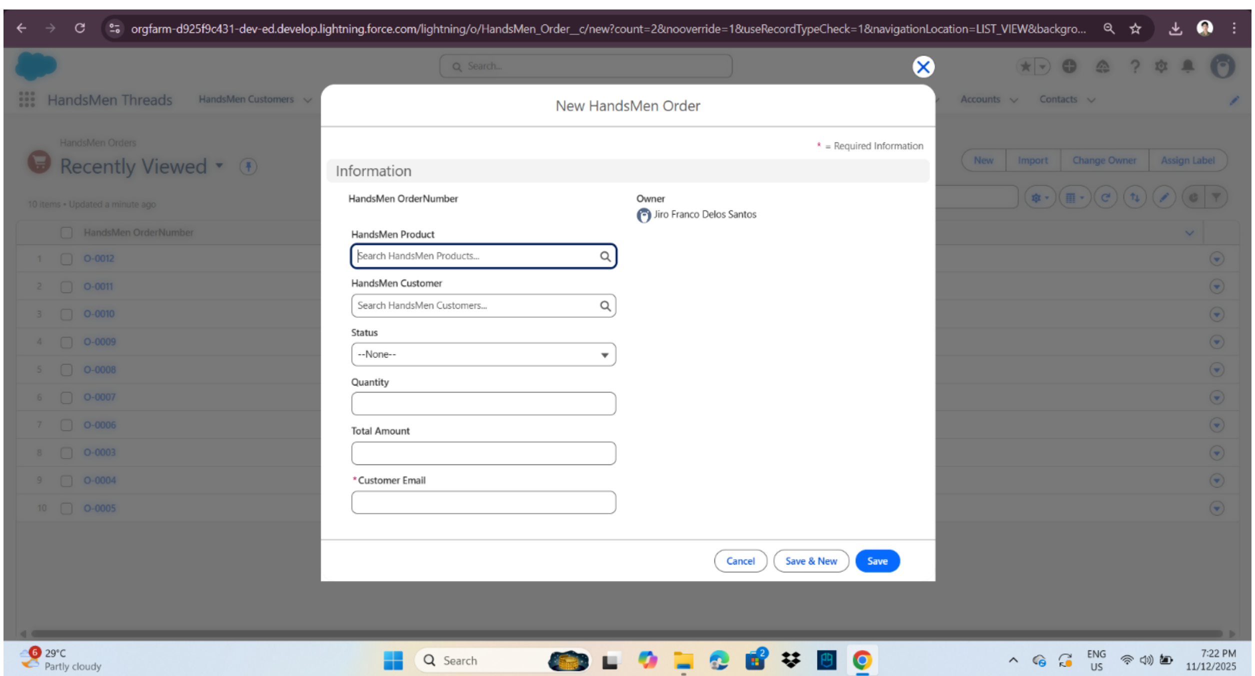



*Figure: HandsMen Order Forms*

**Page Layout:** HandsMen Product

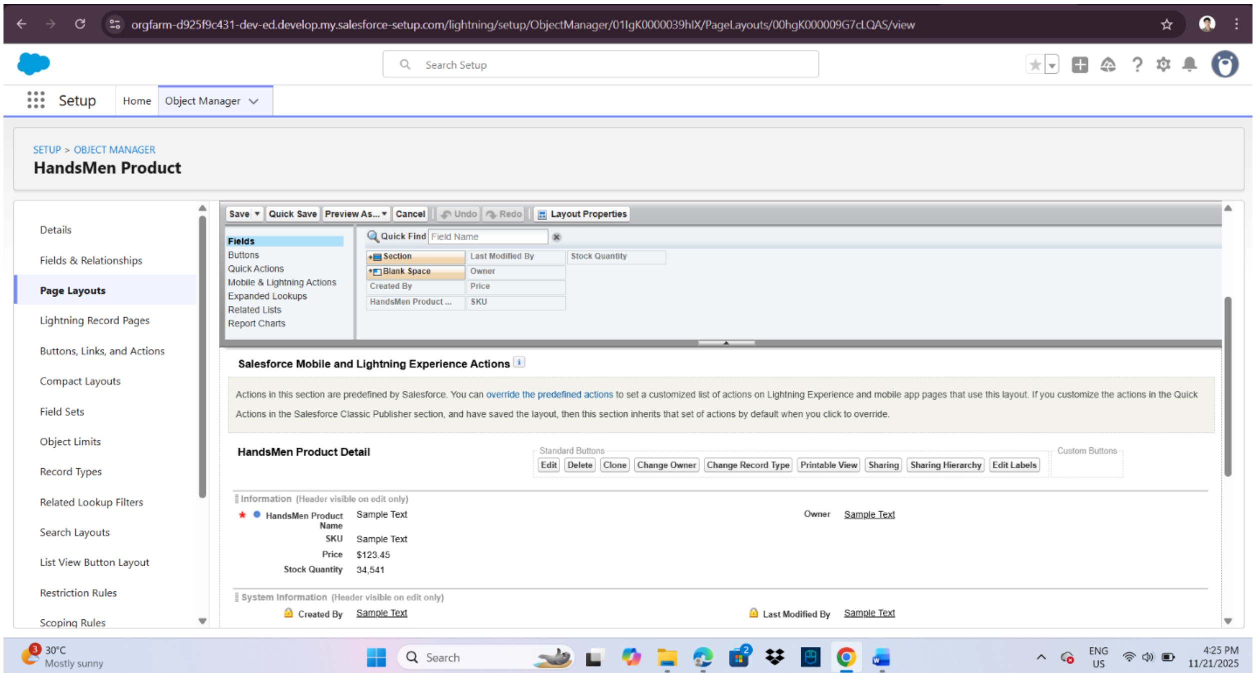



*Figure: HandsMen Product Page Layout*

**Dynamic Forms:** HandsMen Product

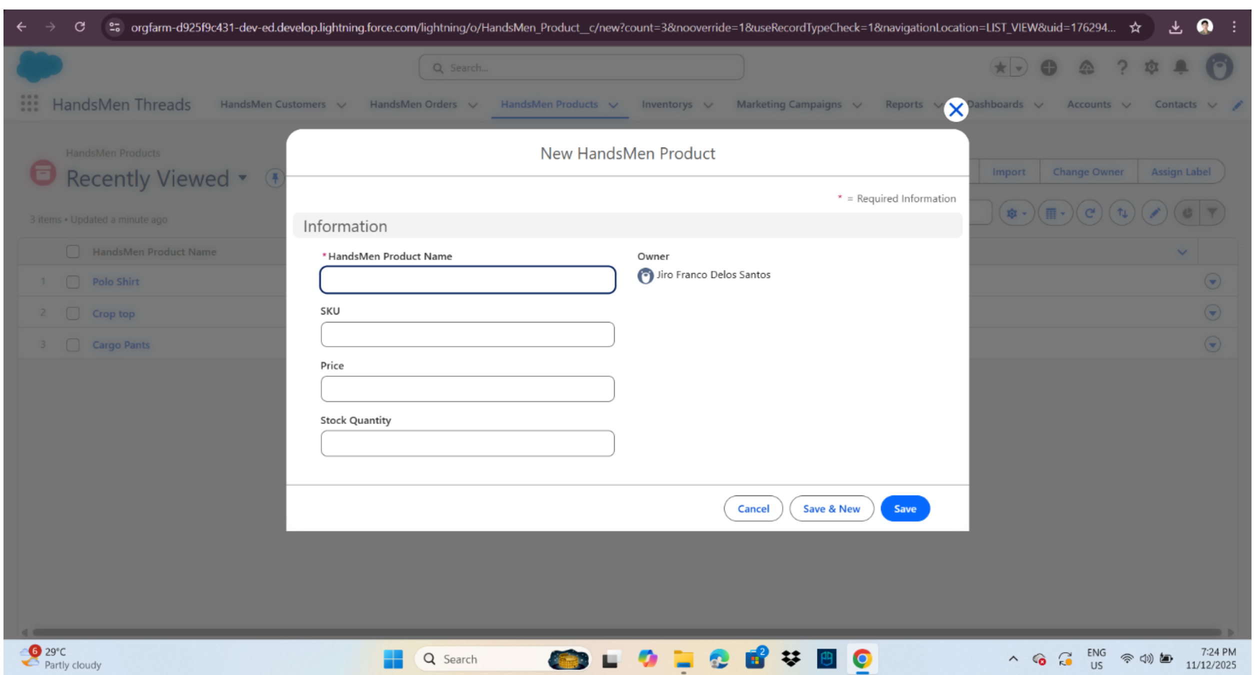



*Figure: HandsMen Product Forms*

## **Page Layout:** Inventory

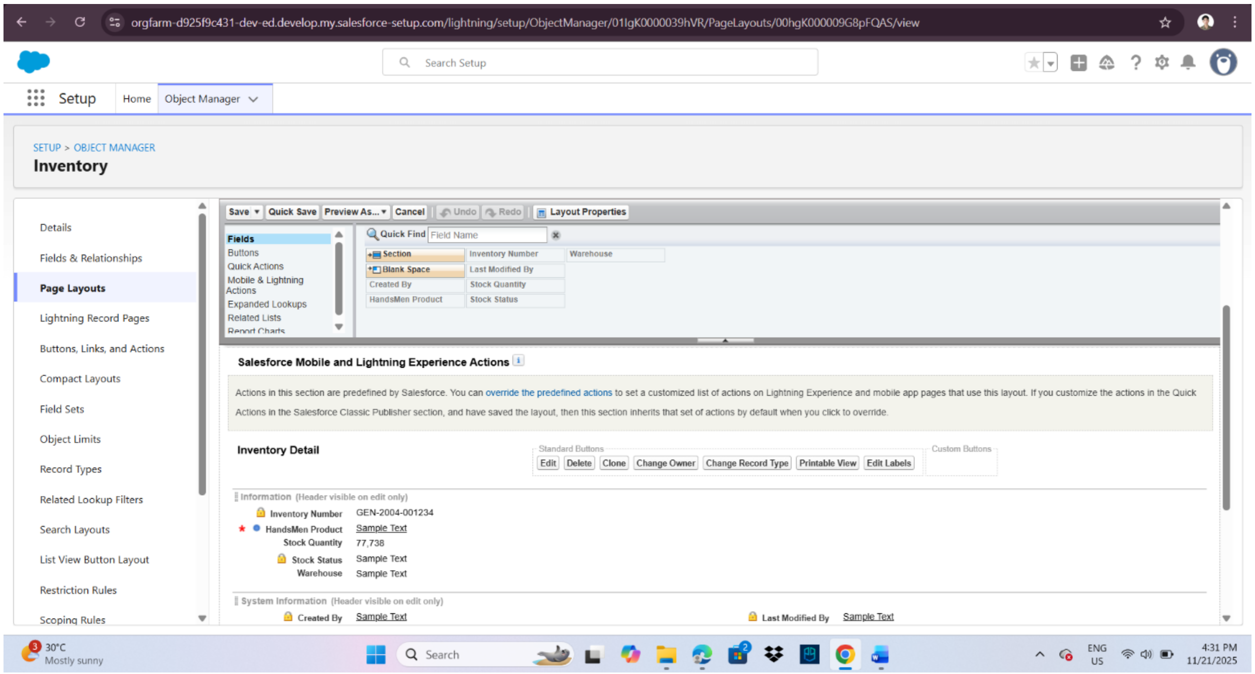


*Figure: Inventory Page Layout*

## **Dynamic Forms:** Inventory

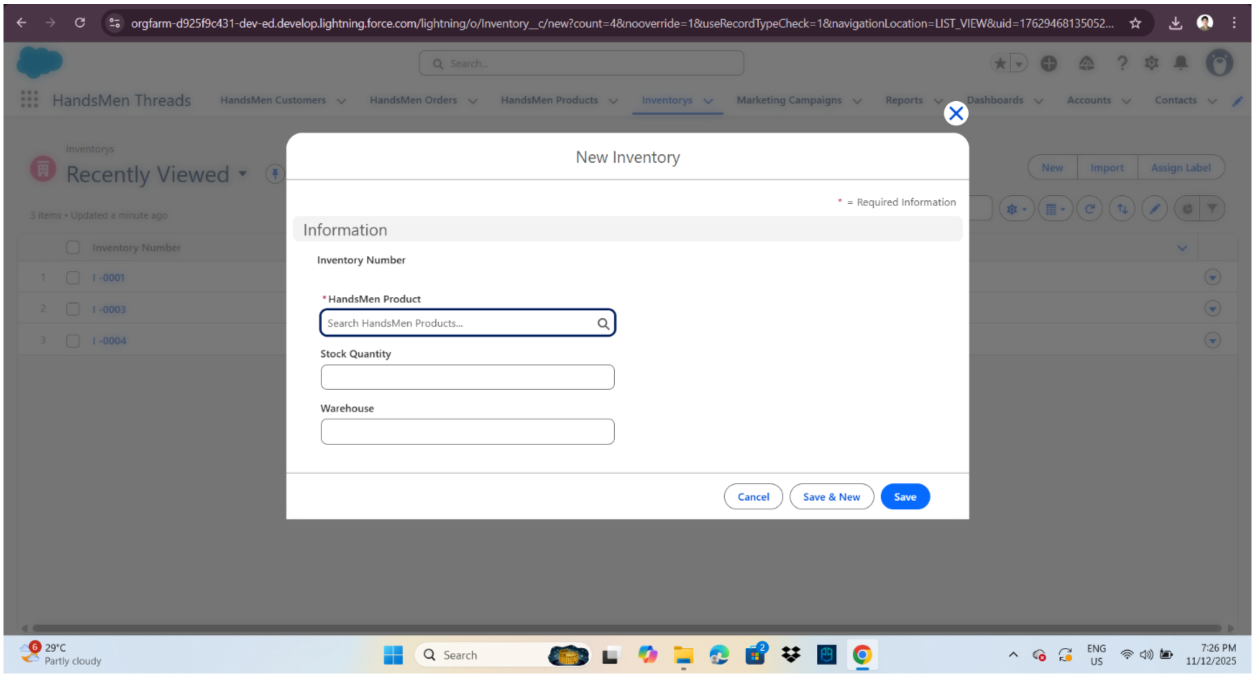


*Figure: Inventory Forms*

**Page Layout:** Marketing Campaign

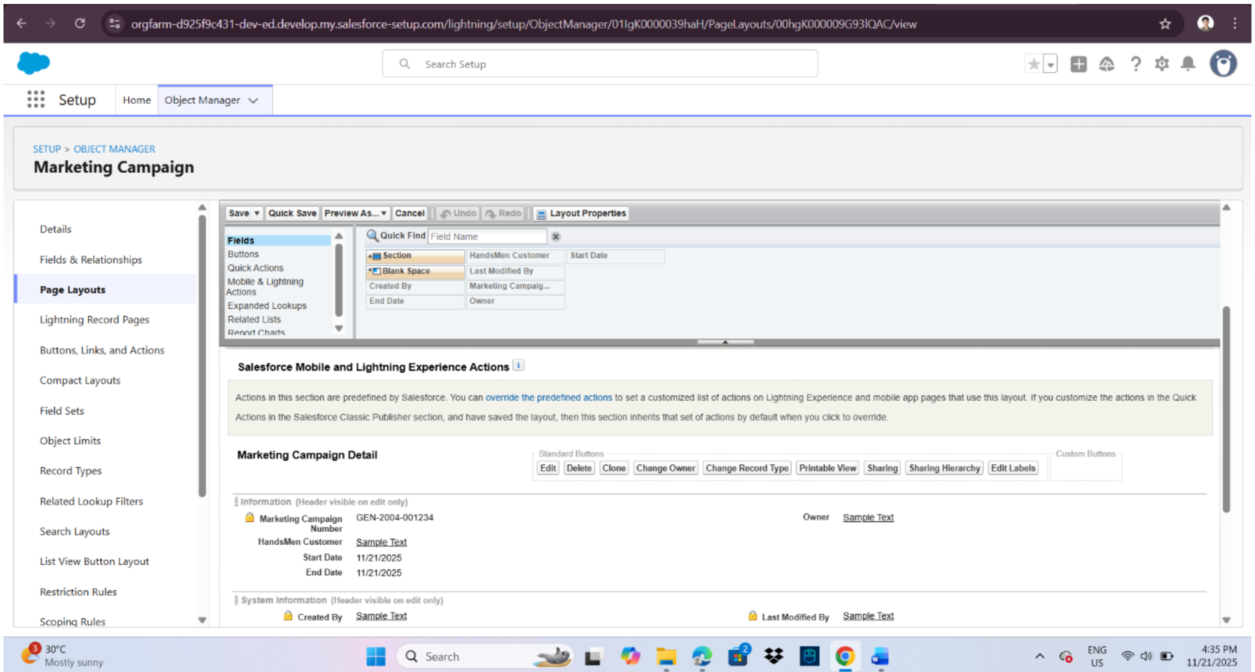



*Figure: Marketing Campaign Page Layout*

**Dynamic Forms:** Marketing Campaign

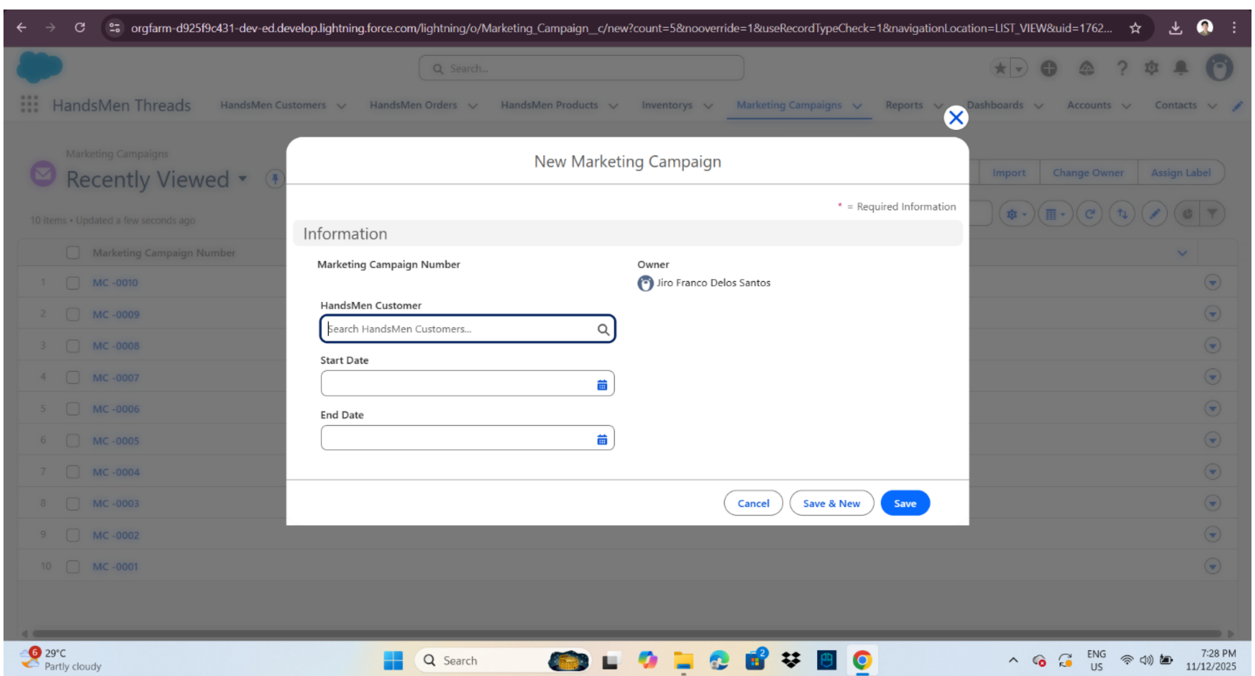



*Figure: Marketing Campaign Forms*

# User Management

## Users Created:

| **Stakeholder** | **Role** | **Responsibilities** | **CRM Access Level/Profile** |
|---|---|---|---|
| **Delos Santos, Jiro Franco** | System Administrator | Manages user permissions, maintains data integrity, and system performance. | Platform 1 |
| **Mikaelson, Jiro** | Marketing | Creates and manages marketing campaigns and promotions for HandsMen Threads. | Platform 1 |
| **Mikaelson, Kol** | Inventory | Oversees product inventory, restocks, and item availability. | Platform 1 |
| **Mikaelson, Niklaus** | Sales | Manages sales operations, team targets, and customer transactions. | Platform 1 |

## User Setup Details:

- **Role Hierarchy:** Assign all users to the correct **Roles**.
- **User Credentials:** Ensure each user has a **unique email and username**.
- **License:** Set **License Type: Salesforce Platform**.
- **Time:** Set **Time Zone: Pacific Standard Time**.
- **Language:** Set **Locale: English (United States)**.

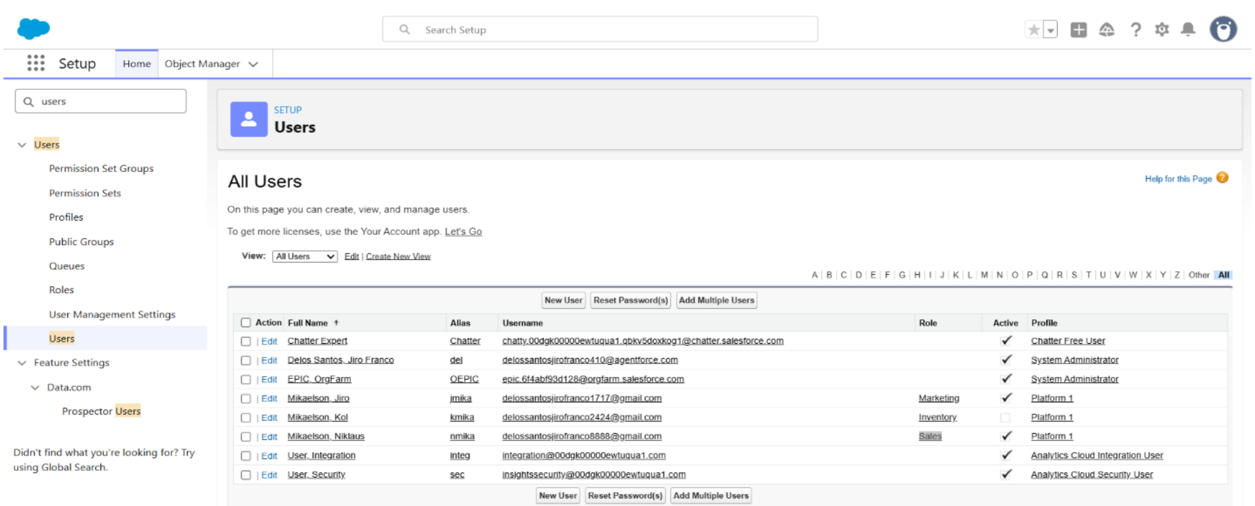


*Figure: Created Users*

## Reports and Dashboards

### Reports:

- **HandsMen Customer Reports**

  This report identifies your most valuable customers based on their **Total Purchases** (Lifetime Value). You can segment your base by **Loyalty Value** to tailor retention and marketing efforts. The **Last Purchased Date** helps flag customers who are becoming inactive and may require re-engagement campaigns.

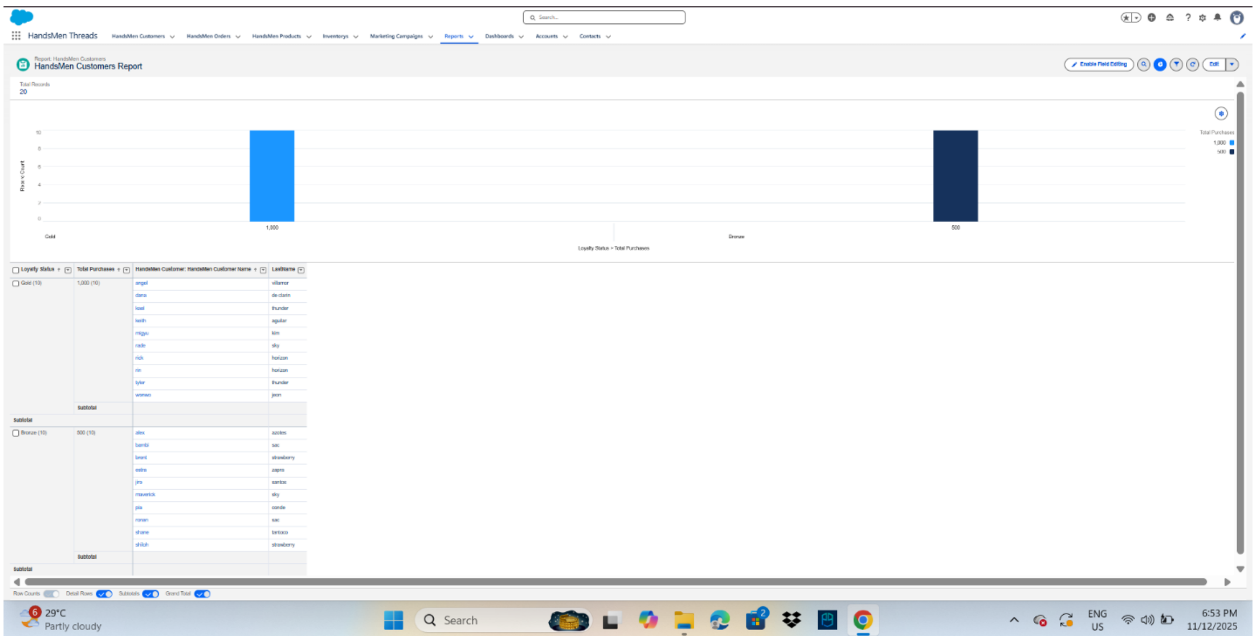



*Figure: HandsMen Customer Reports*

- **HandsMen Order Reports**

  This report tracks the entire sales and fulfillment pipeline, showing the **Status** of all active and completed orders. It provides crucial data on the **Total Amount** and **Total Quantity** of products ordered, informing financial and inventory planning. Use the **Order Date** to analyze order volume trends and measure the efficiency of your order processing time.

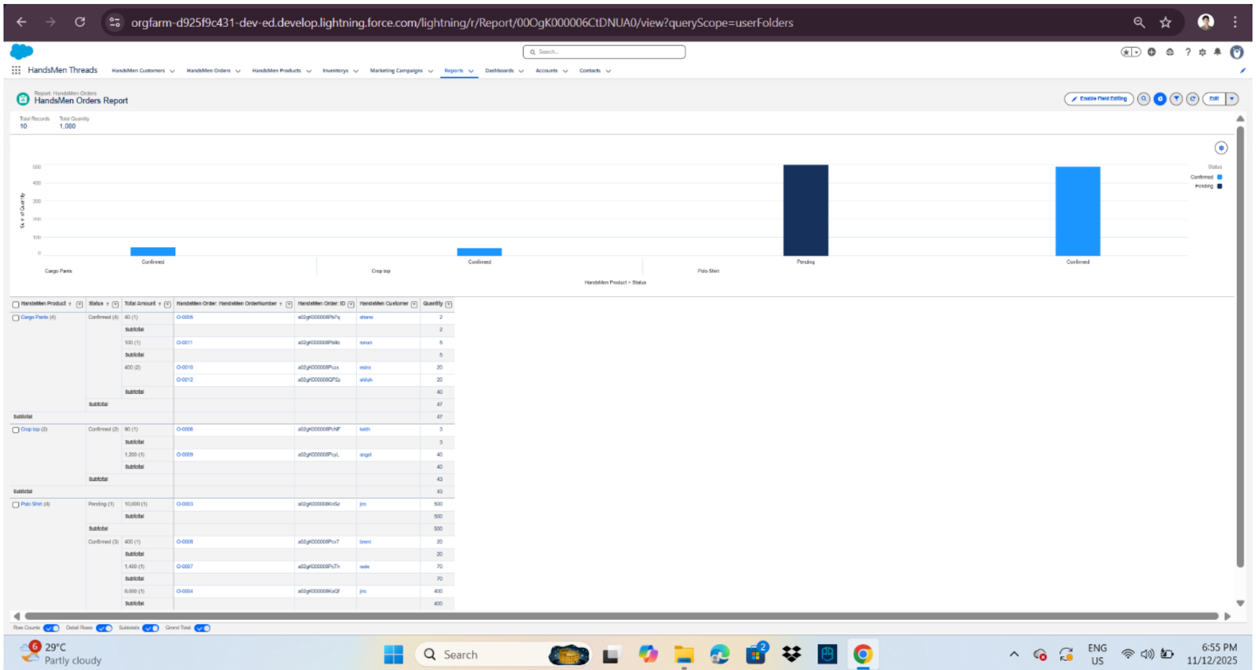


*Figure: HandsMen Order Reports*

- **HandsMen Products**

  This report is essential for product margin analysis by displaying both the Product Price and the underlying Product Cost. It provides a clear view of your entire product catalog, referenced by Product Code and Product Name. Regularly review this report to maintain accurate pricing and maximize your gross profit margins.

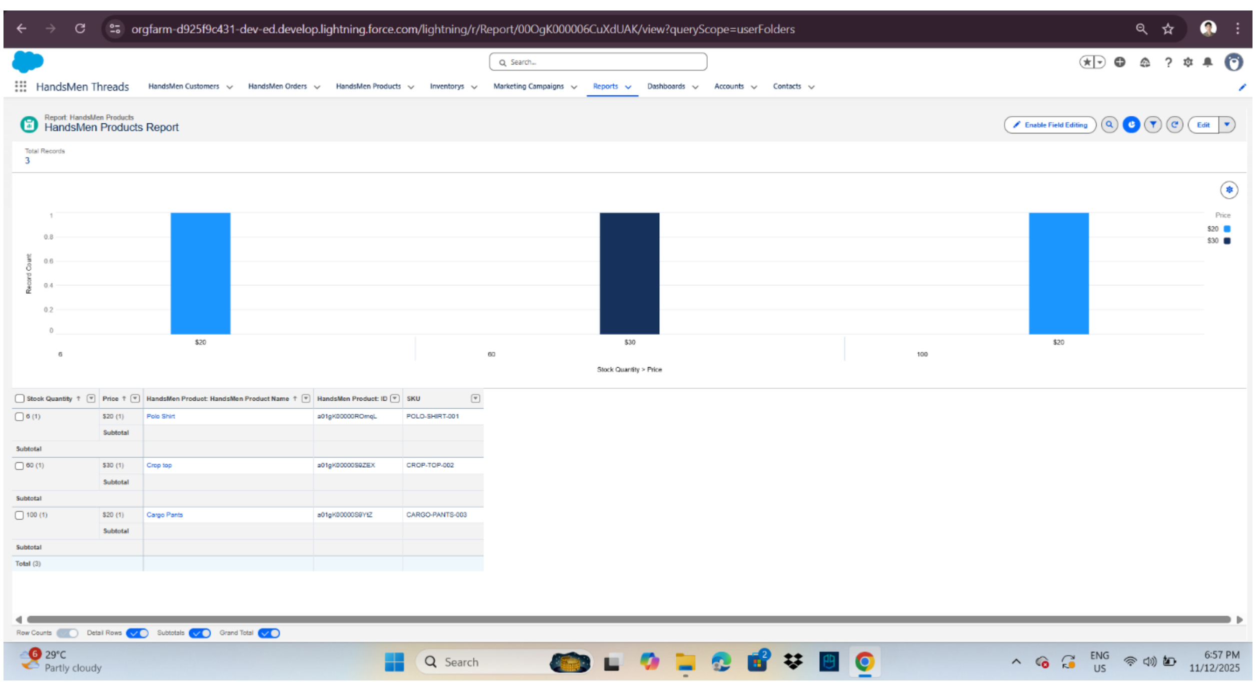


*Figure: HandsMen Product Reports*

- **Inventory Reports**

  The primary function of this report is to flag products that are at risk of a Stockout due to low Available Stock. It enables management to calculate the total current financial value of all goods in stock using the Product Cost. Use this report to trigger timely purchase orders to meet anticipated customer demand.

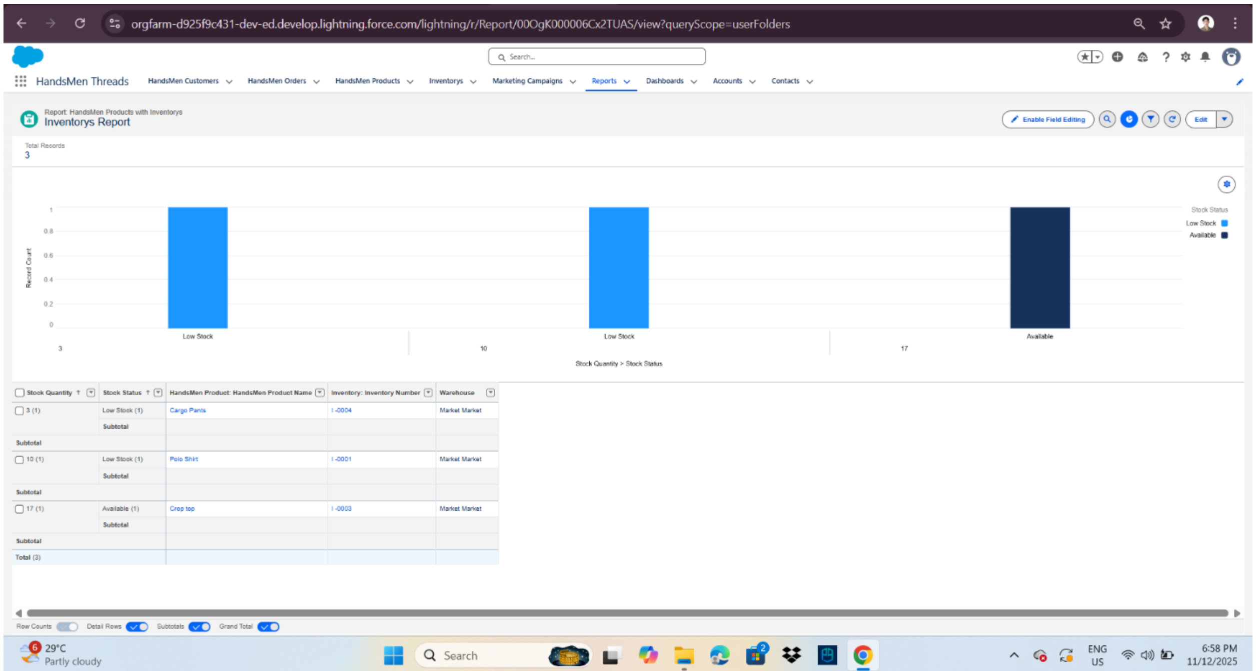



*Figure: Inventory Reports*

- **Marketing Campaign Reports**

  This report provides a scheduled overview of all current and upcoming marketing activities using the **Start Date** and **End Date**. It is crucial for understanding who is accountable for results via the **Campaign Owner**. Tracking the campaign **Status** ensures all activities are running as planned and budgets are being managed effectively.

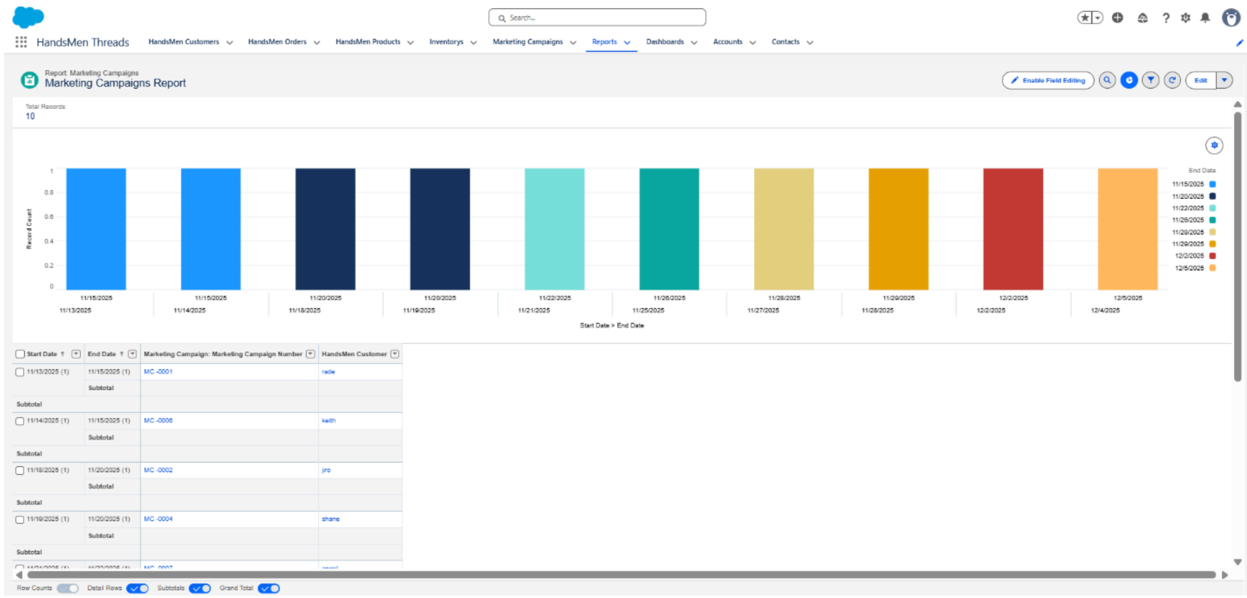


*Figure: Marketing Campaign Reports*

**Dashboard:**

The dashboard clearly shows that customer base is heavily skewed towards Gold-tier loyalty customers over Bronze, indicating an opportunity to increase engagement or rewards to elevate more customers to the highest tier. Fulfillment operations are weighted with a large volume of Pending orders, which is significantly higher than the confirmed or canceled counts, suggesting a current bottleneck or delay in order processing. From a product and inventory perspective, Low Stock is a critical issue as it accounts for the majority of the inventory status, specifically impacting the ability to fulfill sales.

*Figure: Dashboard*

## Lightning Pages

A **Lightning Page** is a custom user interface screen in Salesforce that you build using the Lightning App Builder. It acts as a container to organize standard components, reports, and dashboards into a coherent layout. These pages are assigned dynamically to users based on their profile, creating a tailored, modern experience for viewing records, the home screen, or dedicated app tabs.

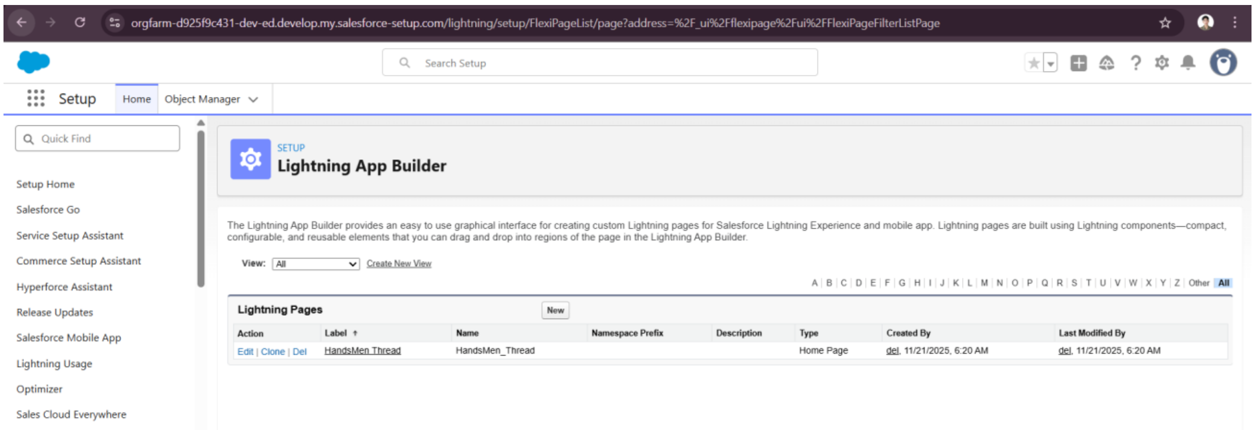



*Figure: Lightning Page*

# Phase 4: Data Migration, Testing & Security

This Phase is critical step before deployment, focusing on the secure population and rigorous validation of the Salesforce environment. This phase begins by executing the data migration plan, employing tools like the **Data Loader** to accurately move all necessary records into the system. Next, we lock down data integrity by documenting and implementing governance features such as **Field History Tracking** and **Duplicate Rules**, alongside finalizing the security model with defined **Profiles, Permission Sets, and Sharing Rules**. The core of this phase is comprehensive quality assurance, which requires both the creation of **Test Classes** for all custom code and the development of detailed **Test Cases** for every feature, including Flows and Triggers. Finally, all feature testing must be substantiated with mandatory **input and output screenshots**, ensuring verifiable proof that every custom process functions perfectly before go-live.

## Data Migration Process

### Customer Data Import:

The batch log confirms the **successful import of 50 customer records** into Salesforce with zero errors. The entire process was extremely fast, utilizing an efficient bulk API process. This validates that the foundational customer data set is now successfully populated and ready for immediate use across your organization.

Reload

**Batches**

| View Request | View Result | Batch ID | Start Time | End Time | Total Processing Time (ms) | API Active Processing Time (ms) | Apex Processing Time (ms) | Records Processed | Records Failed | Retry Count | State Message | Status |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| View Request | View Result | 751g5000000Ny8L | 11/10/2025, 2:29 AM | 11/10/2025, 2:29 AM | 159 | 92 | 4 | 50 | 0 | 0 | | Completed |

*Figure: Customer Data Import*

**Product Data Import:**

This batch log shows the **successful creation of 25 product records**, achieving a perfect 100% success rate. The rapid total processing time confirms the efficiency of this data load. This verifies the completion of the product catalog setup, allowing your inventory and ordering processes to fully commence.

Reload

**Batches**

| View Request | View Result | Batch ID | Start Time | End Time | Total Processing Time (ms) | API Active Processing Time (ms) | Apex Processing Time (ms) | Records Processed | Records Failed | Retry Count | State Message | Status |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| View Request | View Result | 751g5000000Npmg | 11/10/2025, 2:32 AM | 11/10/2025, 2:32 AM | 96 | 50 | 0 | 25 | 0 | 0 | | Completed |

*Figure: Product Data Import*

**Data Import Wizard Usage:**

It shows the **Recently Viewed list page for my HandsMen Customer object**, confirming that the customer records were successfully imported. This list view provides a quick visual validation of the names. that were created or updated during the migration process. This visual evidence verifies the successful use of a data loading tool to populate the custom **HandsMen Customer** object.

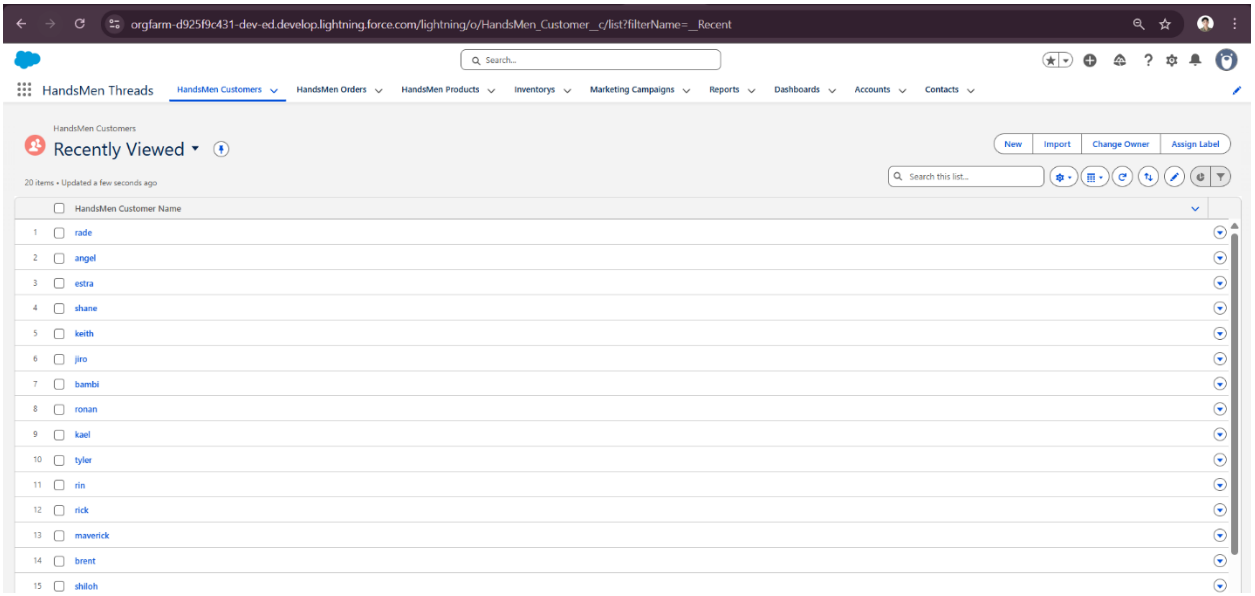



*Figure: Data Import Wizard Usage*

# Profiles

Profiles define a user's **maximum permissions and object access** across the entire organization, controlling which tabs, apps, and general functions they can see. Every user must have exactly one Profile assigned, making it the foundational layer of access control.

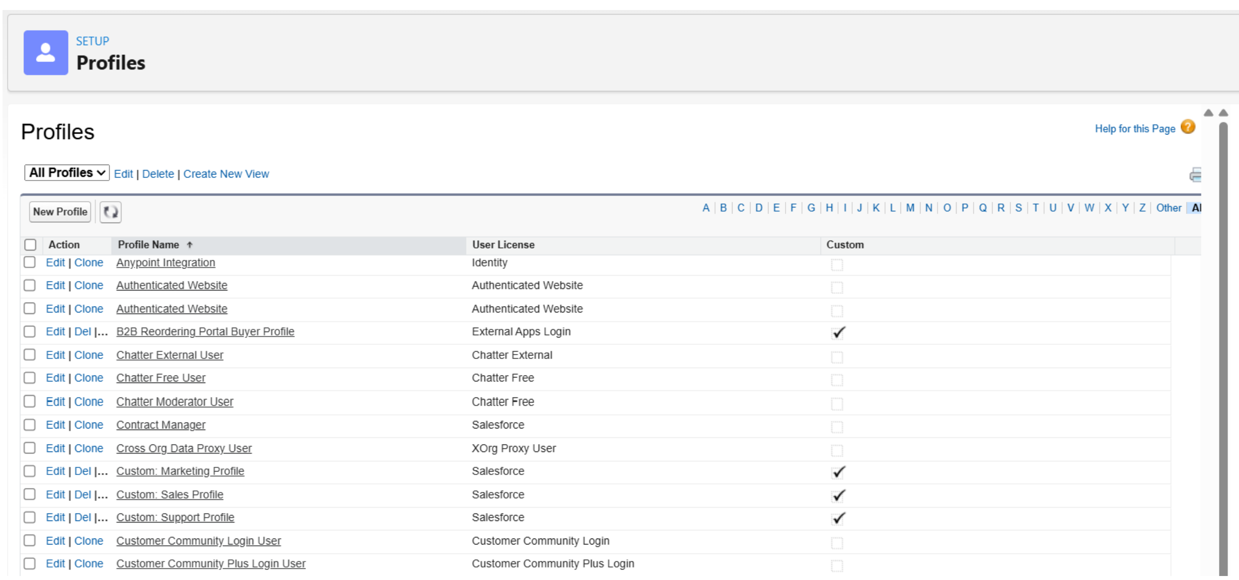



*Figure: Profiles*

## Role Hierarchy

The Role Hierarchy primarily dictates **data visibility** by opening up record access based on a user's position in the organizational chart. Users higher in the hierarchy can typically view records owned by users below them, unless overridden by sharing settings.

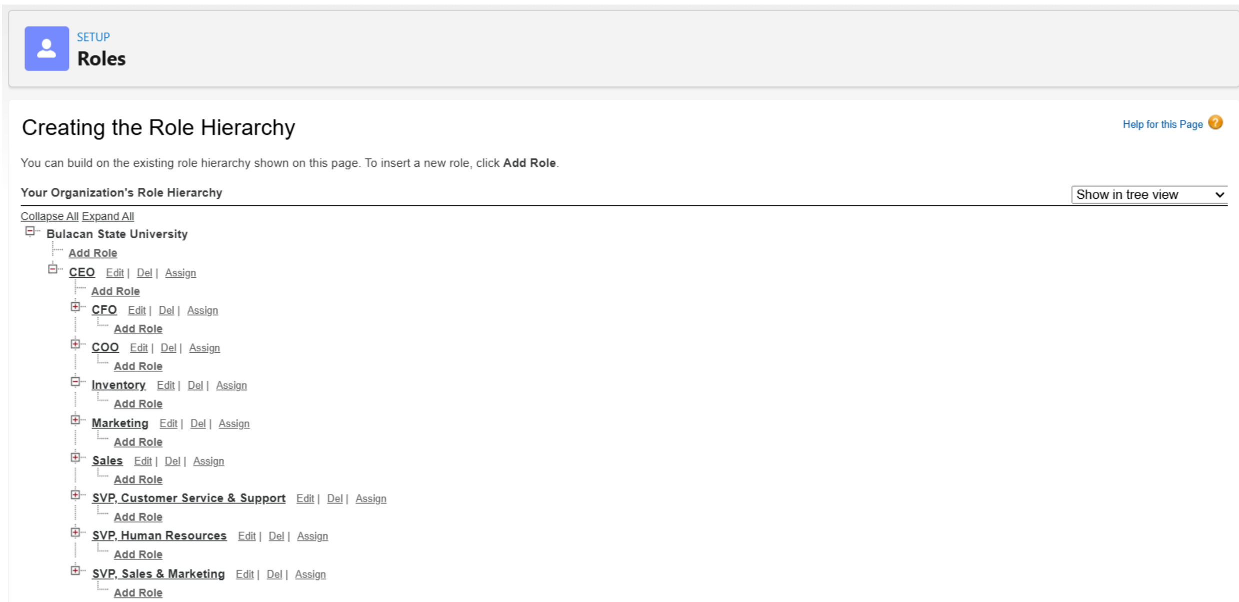



*Figure: Role Hierarchy*

## Permission Sets Created

Permission Sets are additive tools used to grant **extra permissions and settings** without changing a user's base Profile, making them ideal for managing exceptions or temporary access. A user can be assigned multiple Permission Sets, allowing for flexible and granular access control.

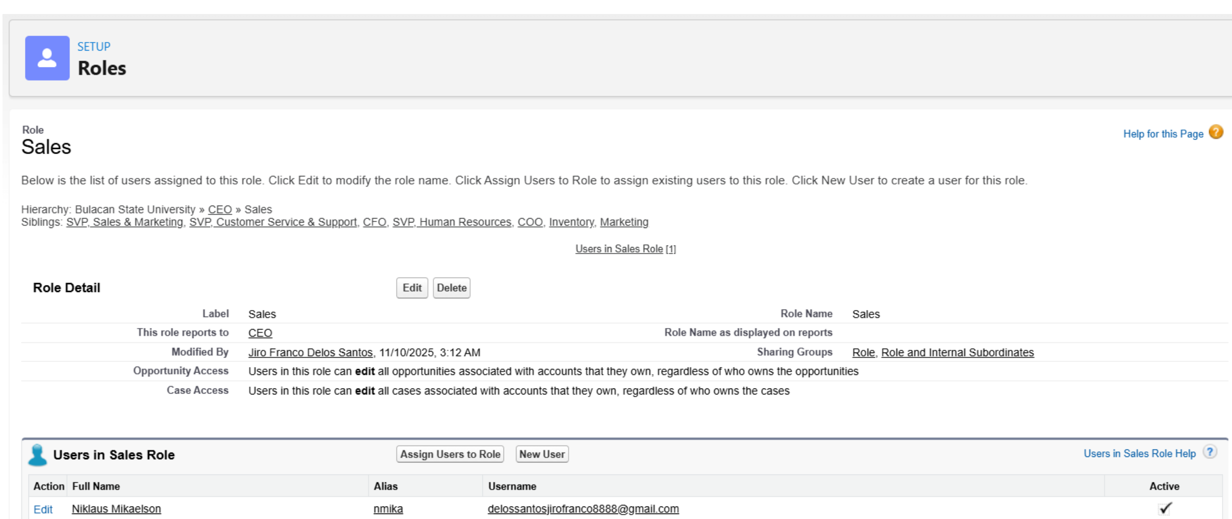


*Figure: Sales Permission Sets*

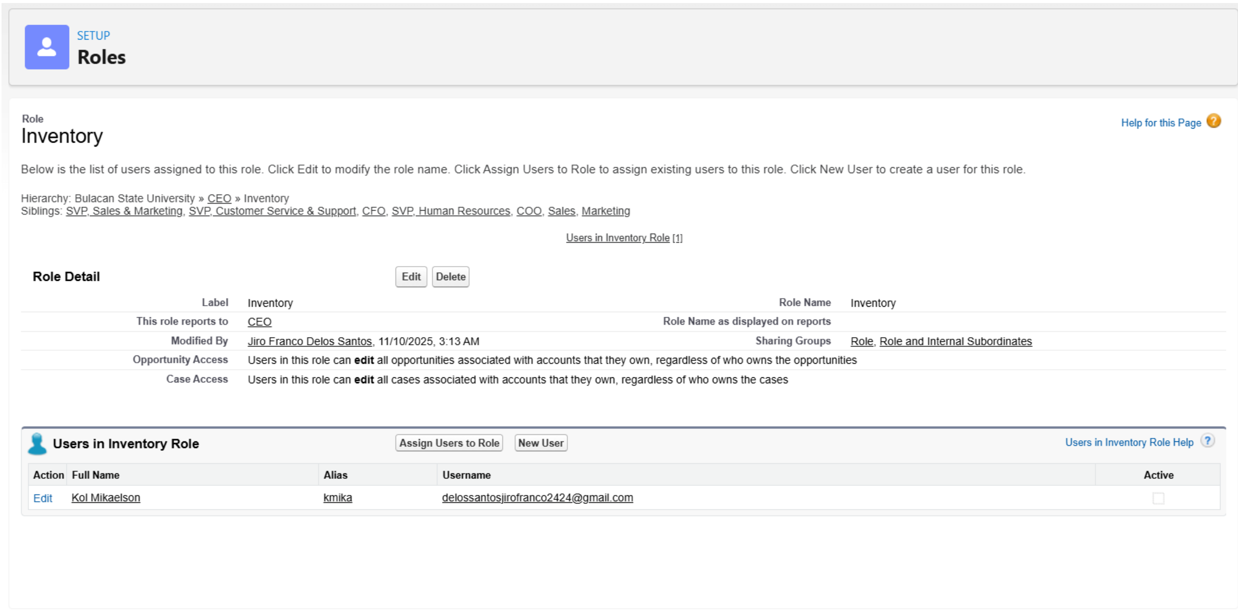



*Figure: Inventory Permission Set*

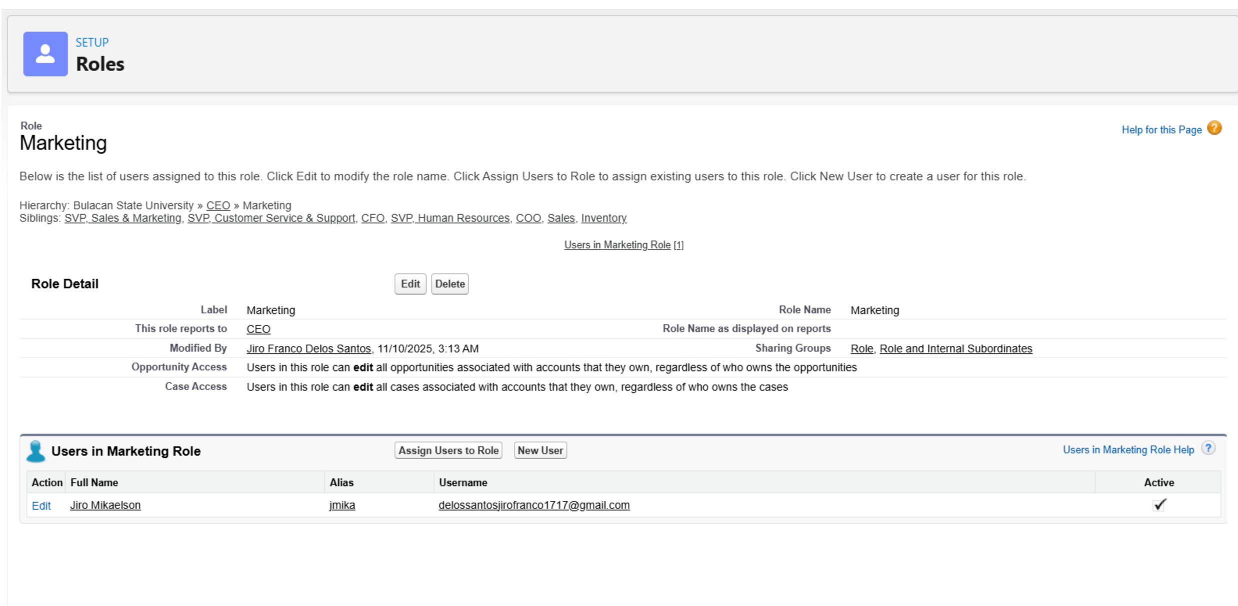



*Figure: Marketing Permission Sets*

## Sharing Rules

Sharing Rules grant exceptions to your organization's default sharing settings, automatically providing **read or read/write access** to groups of users based on record ownership or specific criteria.

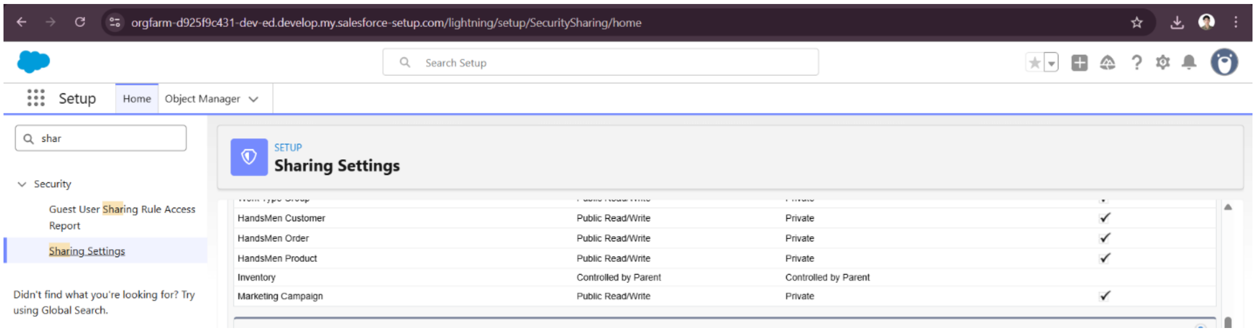



*Figure: Sharing Settings*

## Test Cases for each Salesforce Features

**Test Case1:** Customer creation and email validation

This test case **passed** because the system successfully validated the email format, initially rejecting the Yahoo address and accepting the Gmail address, before ultimately creating the new customer record for Rade Sky. It confirms that both the basic customer creation functionality and the specific email validation rule are working as expected.

| **Test Step** | **Input** | **Expected Output** | **Actual Output** | **Status** |
|---|---|---|---|---|
| Navigate to Customers tab | Click **"New"** button | New customer form opens | Form opened successfully | Pass |
| Enter customer name | **"Rade Sky"** | Name field populated | Name accepted | Pass |
| Enter invalid email | **"radesky@yahoo.com"** | Validation error displayed | Error: **"Please fill correct gmail"** | Pass |

| **Test Step** | **Input** | **Expected Output** | **Actual Output** | **Status** |
|---|---|---|---|---|
| Correct email | **"radesky@gmail.com"** | Email accepted | Email saved successfully | Pass |
| Enter phone number | **"888-7777"** | Phone saved | Phone number recorded | Pass |
| Save record | Click **"Save"** | Customer record created | **Rade Sky Was Created** | Pass |

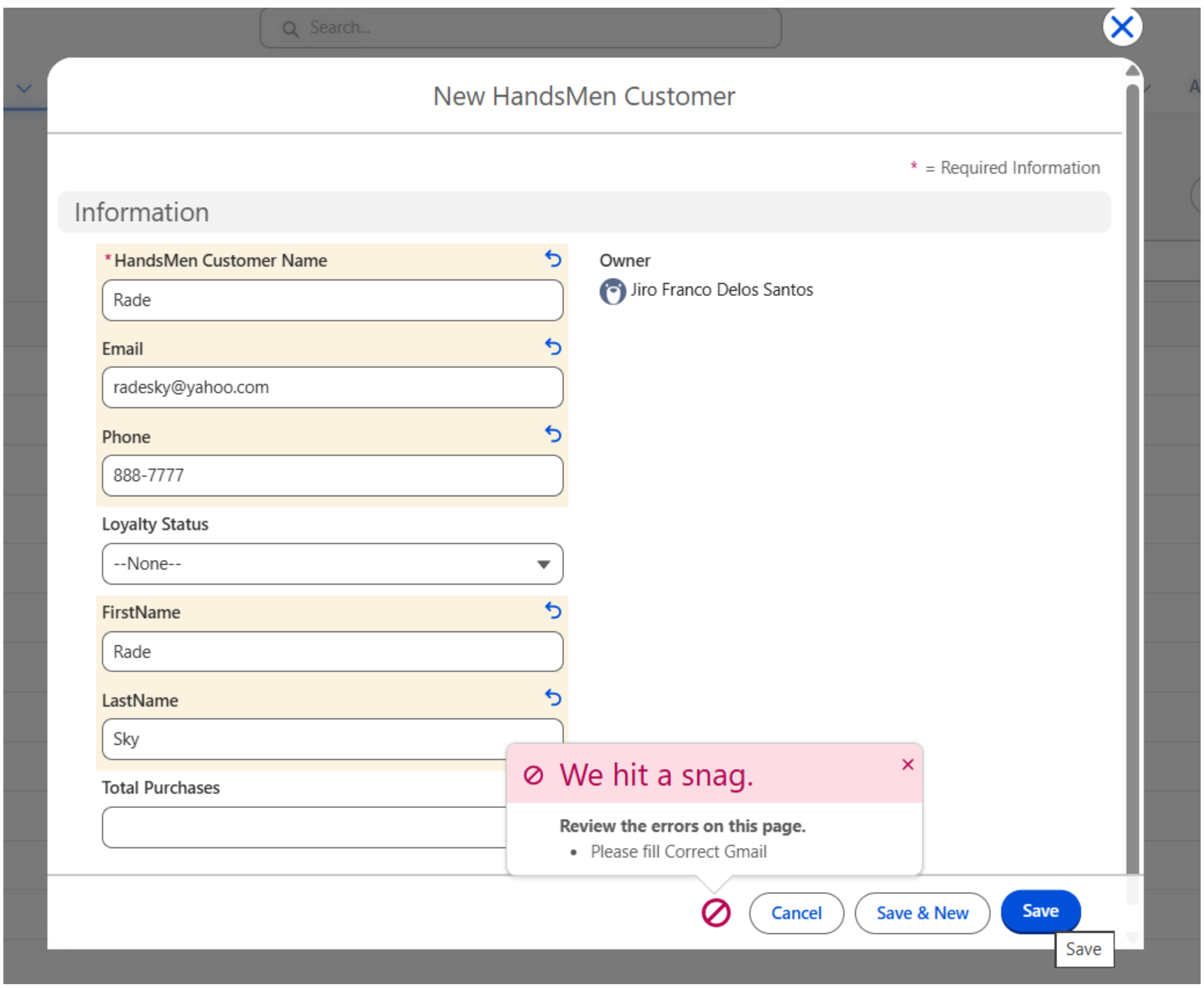


*Figure: Test Case 1—Email Validation*

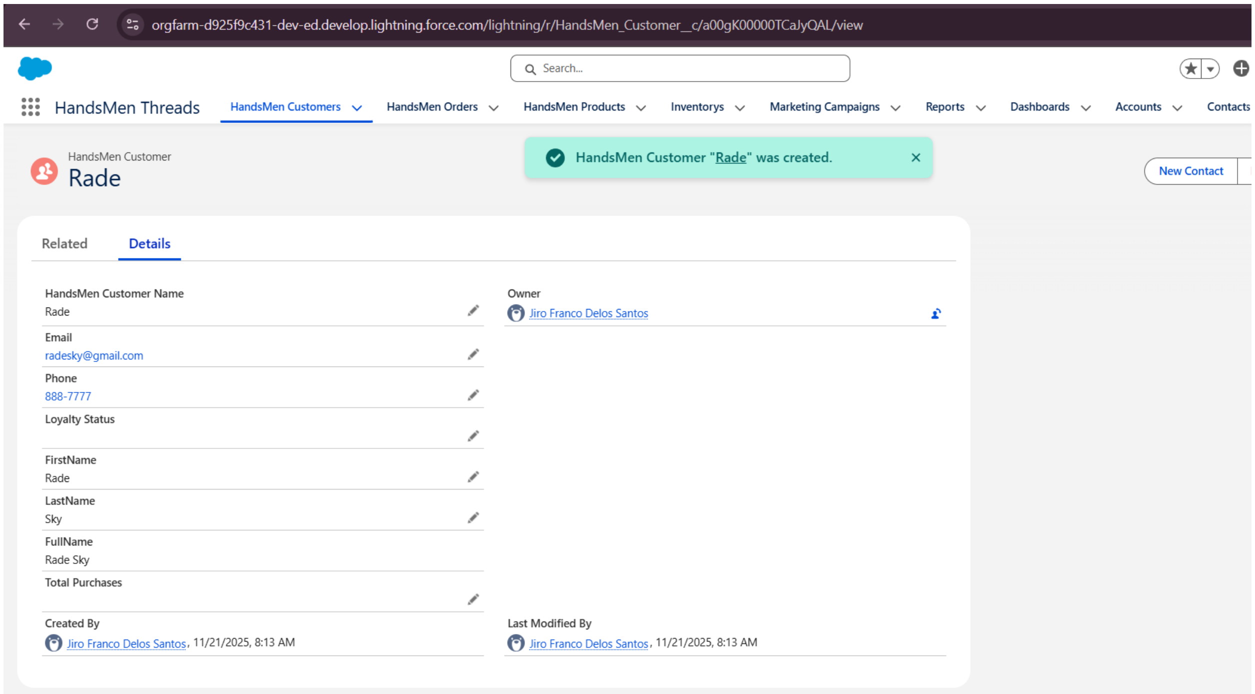
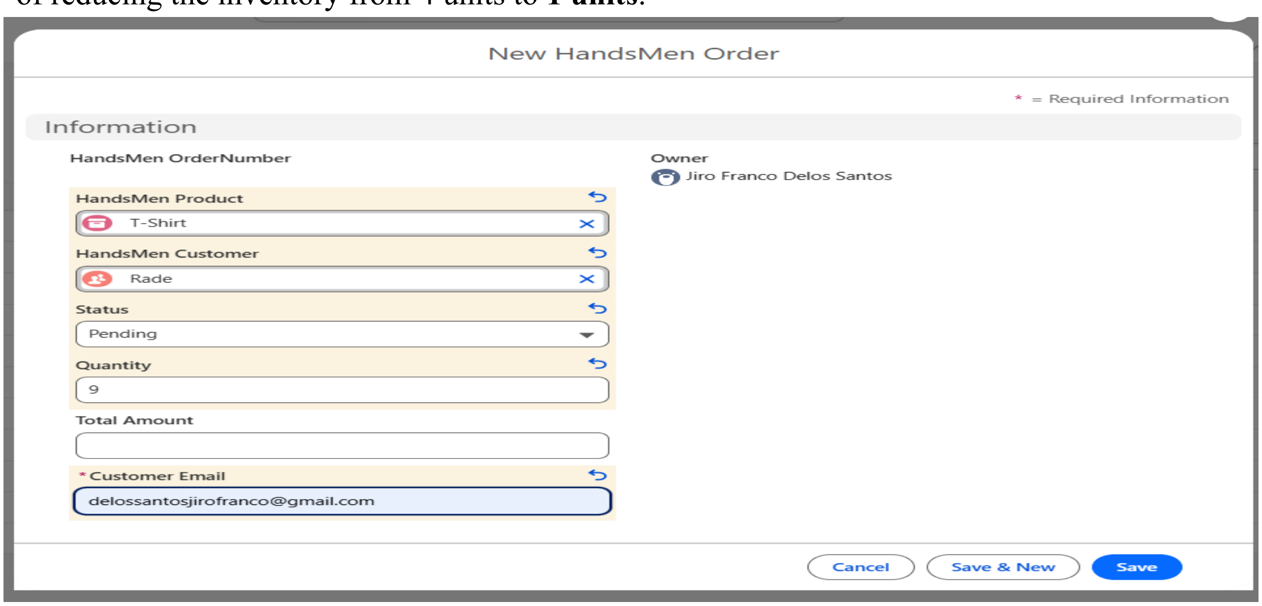



*Figure: Test Case 1—Customer Creation*

**Test Case 2:** Order Creation with Stock Validation

This test case confirms the system's ability to successfully validate and process an order based on current inventory levels. The system correctly triggered an error when the user attempted to order 9 units against the initial stock of 5, preventing an over-sale. Once the quantity was corrected to 4, the order was created successfully, and the system accurately executed the final step of reducing the inventory from 4 units to **1 units**.



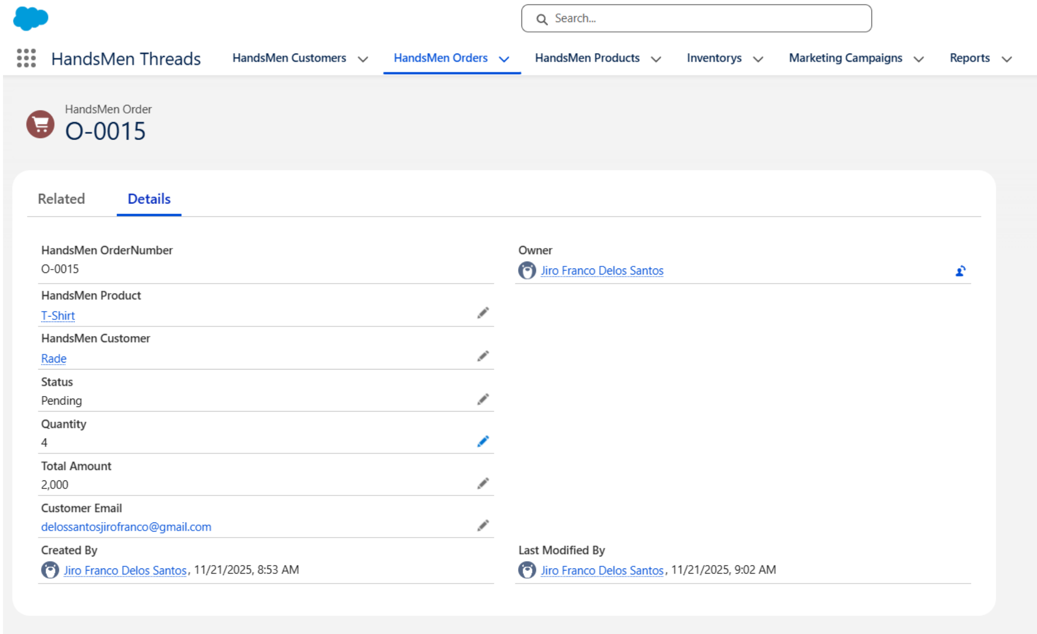
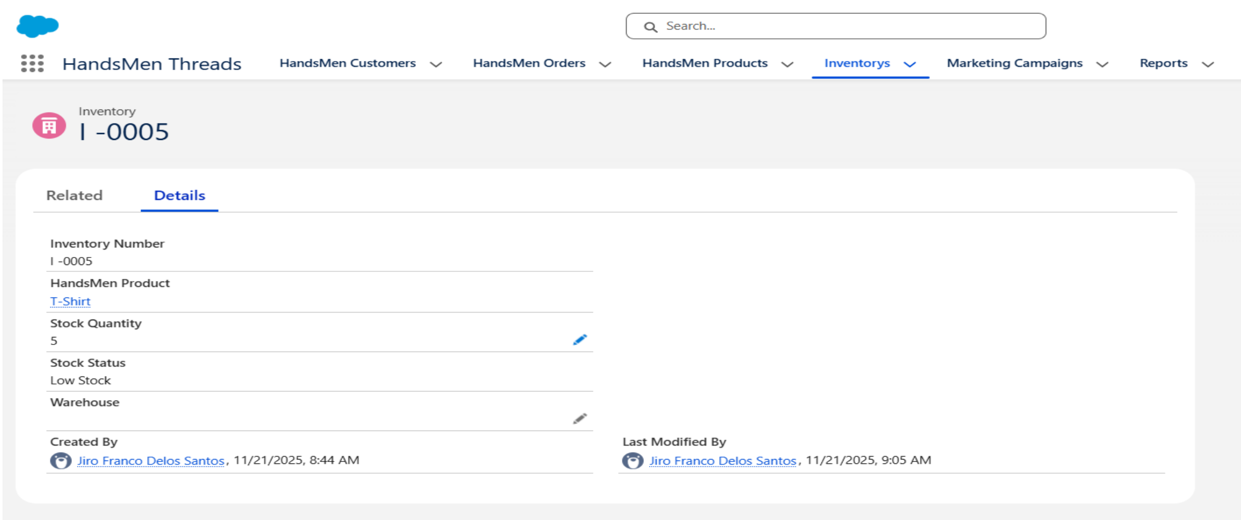
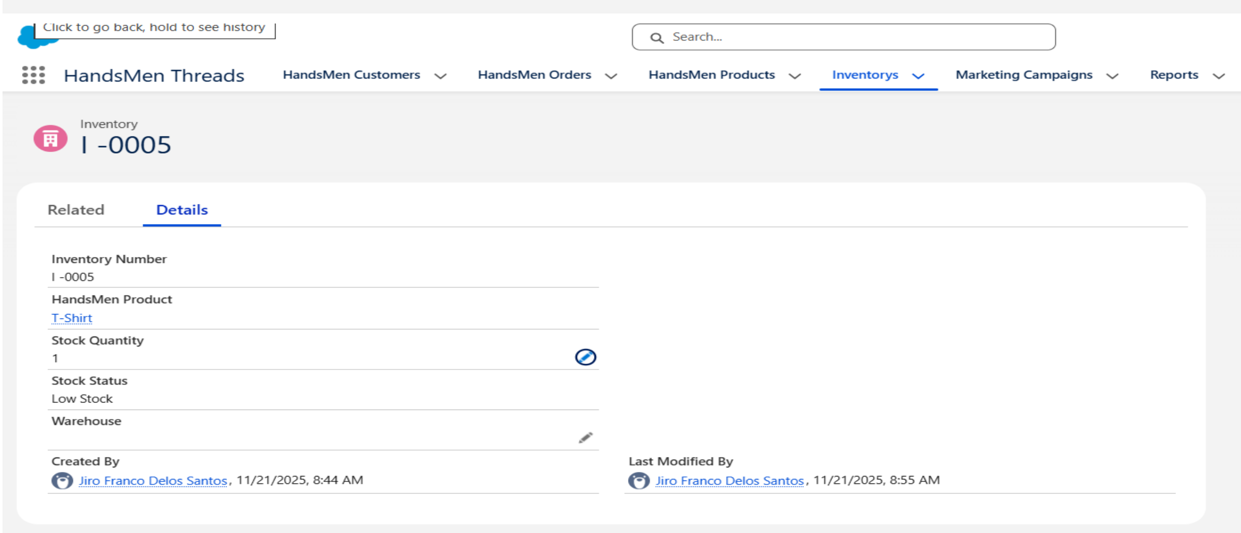
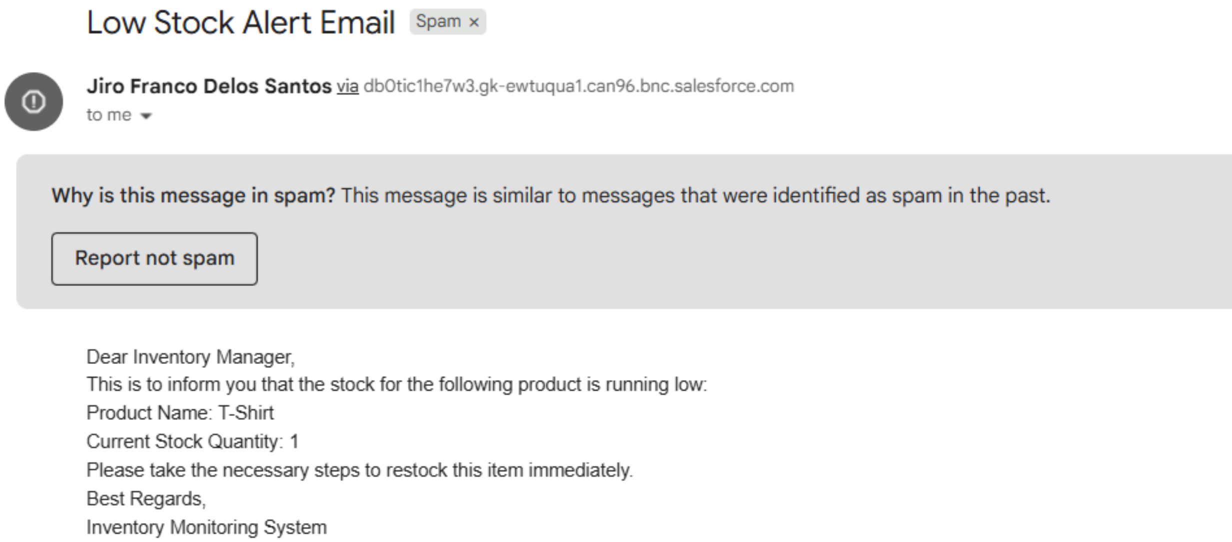



*Figure: Order Creation*





Low Stock Alert Email Spam ×

**Jiro Franco Delos Santos** via db0tic1he7w3.gk-ewtuqua1.can96.bnc.salesforce.com
to me

**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: 1
Please take the necessary steps to restock this item immediately.
Best Regards,
Inventory Monitoring System

*Figure: Stock Validation*

# Phase 5: Deployment, Documentation & Maintenance

The final stage that focuses on delivering the solution and ensuring its long-term viability. The outline highlights the necessity of detailing the **Deployment strategy**, such as using change sets, to ensure a smooth and controlled release of the finished product into the production environment. It requires a **basic description** of the procedures for system maintenance and continuous monitoring to quickly identify and address any operational issues. Comprehensive **documentation of the troubleshooting approach** must be created, providing clear guidelines for resolving common problems. Ultimately, this phase is crucial for the successful handover, operational support, and ongoing stability of the newly deployed system.

## Deployment Strategy

The chosen deployment method is **Change Sets**, which allows for a declarative, point-and-click movement of metadata from the development sandbox to the production organization.

- **Outbound Change Set Name: HandsMenThreads**

## Pre-Deployment Checklist

Successful deployment relies on the completion of the following critical steps to ensure system stability and data integrity:

- **Testing & Activation:**

    - All **Apex test classes** passing.
    - **Validation rules** tested thoroughly in the sandbox environment.
    - **Flows** activated and tested for expected business logic execution.
    - **User Acceptance Testing (UAT)** completed and signed off by key stakeholders.
- **Safety & Contingency:**
    - **Backup of production data** completed before deployment.
    - **Rollback plan** documented and verified in case of critical deployment failure.

**Components Included in Change Set**

The **HandsMenThreads** Change Set contains all the necessary metadata components for the application's functionality:

| Component Type | Quantity | Key Items |
| --- | --- | --- |
| **Custom Objects** | 5 | HandsMen_Customer__c, HandsMen_Product__c, HandsMen_Order__c, Marketing_Campaign__c, Inventory__c |
| **Custom Fields** | 20+ | Fields across all custom objects (e.g., phone, price, loyalty status). |
| **Page Layouts** | 5 | Customer Sales Layout, Product Management Layout, Order Processing Layout, Marketing Campaign Layout, Inventory Campaign Layout |
| **Validation Rules** | 5 | Rules for data quality assurance (e.g., email format, required fields). |
| **Flows** | 3 | Email Notification Flow, Scheduled Loyalty Update Flow, Low Stock Alert Flow |
| **Apex Classes (Triggers)** | 3 | OrderTotalTrigger, StockDeductionTrigger, CustomerLoyaltyUpdateTrigger |

| Component Type | Quantity | Key Items |
|---|---|---|
| **Lightning App** | 1 | HandsMen Threads App (The main application interface). |
| **Profiles** | 3 | HandsMen Sales Profile, HandsMen Inventory Profile, HandsMen Marketing Profile |
| **Permission Sets** | 3 | Used for granular access control. |
| **Email Templates** | 3 | Standardized templates for automated communications. |

## Data Security Model

Security architecture utilizes **Profiles** and **Permission Sets** to enforce a **least-privilege model**, ensuring users only have access necessary for their organizational functions. This model dictates data visibility and manipulation rights across key objects.

### Access Permissions by Role

The following table outlines the specific permissions granted to each functional role within the system, ensuring data integrity and compliance.

| Role | Customer Object | Order Object | Inventory Object | Product Object | Marketing Campaign Object |
|---|---|---|---|---|---|
| **Sales Manager** | **Complete Access** (Read, Create, | **Complete Access** (Read, Create, | Read Only | Read Only | Read Only |

| Role | Customer Object | Order Object | Inventory Object | Product Object | Marketing Campaign Object |
|---|---|---|---|---|---|
| | Edit, Delete) | Edit, Delete) | | | |
| **Inventory Manager** | Read Only | Read Only | **Read and Modify** (Read, Create, Edit) | **Read and Modify** (Read, Create, Edit) | Read Only |
| **Marketing Team** | Read Only | No Access | No Access | No Access | **Modify Permissions** (Read, Create, Edit) |

**Implementation Notes**

- **Profiles & Permission Sets:** Access will be implemented primarily through **Profiles** (for base access) and **Permission Sets** (for additional or specific access) to manage permissions efficiently.
- **Default Access:** The system's **Organization-Wide Defaults (OWD)** will be set to **Private** for all sensitive objects (Customer, Order, Inventory) to restrict record access to owners and users higher in the role hierarchy, requiring explicit sharing (via Sharing Rules or Role Hierarchy) where needed.

## Deliverables

These are the tangible and final outputs achieved upon the completion of the project build, confirming all planned configuration and documentation has been delivered.

**I. Documentation & Design Artifacts**

- **Comprehensive Solution Design Document:** A complete blueprint of the system, including the business requirements, architecture decisions, and implementation details.
  - **Object Model:** Defines all custom and standard objects used, their relationships, and purpose.
  - **Entity Relationship Diagram (ERD):** A visual representation of how all the key custom objects (Customer, Order, Product, Inventory, Marketing Campaign) are connected via lookups and master-detail relationships.
  - **Automation Strategy:** Detailed documentation of the logic and criteria for all implemented Flows and Apex Triggers.

## II. Core System Configuration

- **Fully Configured Custom Objects:** The five key objects (HandsMen_Customer__c, HandsMen_Product__c, etc.) with all **25+ field definitions** (data types, help text, required settings) finalized.
- **Implemented Validation Rules and Data Quality Controls:** The five defined rules (e.g., email format check, stock validation) active and tested across relevant objects to ensure data integrity.
- **Email Templates:** The three finalized templates for customer (e.g., Order Confirmation) and internal (e.g., Low Stock Alert) communications, ready for use by Flows.

## III. Automated Business Logic

- **Record-Triggered and Scheduled Flow Automations:** The three key Flows (Email Notification Flow, Scheduled Loyalty Update Flow, Low Stock Alert Flow) are activated and operational.
- **Custom Apex Triggers for Complex Business Logic:** The three triggers (OrderTotalTrigger, StockDeductionTrigger, CustomerLoyaltyUpdateTrigger) are deployed and confirmed to execute the necessary calculations and updates in real-time.
- **Batch Job Implementations for Scheduled Processing:** The Scheduled Loyalty Update batch job is configured to run at a defined interval (e.g., daily) to maintain data accuracy.

## IV. Security Implementation

- **Security Model Implementation:** The complete access structure is built and deployed, including the three **Profiles**, established **Roles**, and necessary **Permission Sets** to enforce the role-based access defined in the Data Security Model.

# Conclusion

The **HandsMen Threads Salesforce CRM implementation** represents a transformative achievement in operational excellence, successfully migrating core business processes onto a unified, cloud-based platform. This comprehensive solution seamlessly orchestrates vital functions, including **sales pipeline management**, **real-time inventory tracking**, and **targeted marketing campaigns**. By consolidating these areas, the platform eliminates data silos, strengthens customer relationships through a single source of truth, and maintains exceptional data accuracy across all domains.

Harnessing Salesforce's extensive capabilities—specifically the declarative automation (Flows) and programmatic enhancements (Apex Triggers)—we've engineered an integrated platform that moves beyond mere record-keeping. It delivers **real-time insights** into customer purchasing behavior and product demand, empowers **informed decision-making** (e.g., proactive stock alerts and accurate order totaling), and significantly reduces manual intervention. This implementation establishes a **scalable, secure, and resilient foundation** for sustained growth, ensuring HandsMen Threads is equipped to deliver exceptional, personalized customer experiences well into the future while maintaining strict data governance via the implemented security model.

## Future Enhancements

My vision for the HandsMen Threads platform extends beyond current capabilities with several strategic enhancement opportunities designed to deepen customer engagement and expand operational efficiency:

**I. Customer Experience & Accessibility**

- **Customer Self-Service Portal:** A dedicated web-based portal enabling customers to independently access order history, monitor loyalty points, and update profile information, thereby reducing the workload on the sales team and enhancing customer independence.

- **Mobile Application Development:** Leveraging the Salesforce Mobile SDK to develop native mobile applications for field teams, enabling real-time **inventory management** and **order processing** directly from mobile devices while on the go, improving agility.
- **Omnichannel Messaging Integration:** Connecting with communication platforms like **WhatsApp and SMS** to deliver real-time order confirmations, shipping notifications, and personalized loyalty program updates directly to the customer's preferred channel.

**II. Intelligence & Performance**

- **Advanced Analytics & Visualization:** Creating sophisticated **sales and inventory dashboards** within Salesforce for deeper performance insights, KPI monitoring, and trend identification across product lines and customer segments using tools like Einstein Analytics (Tableau CRM).
- **AI-Powered Personalization (Salesforce Einstein):** Integrating **Salesforce Einstein** to deliver intelligent product recommendations based on individual purchase patterns and history, significantly enhancing cross-selling opportunities and marketing relevance.

## Key Learning Outcomes

The project provided extensive hands-on experience and proficiency in modern Salesforce development and administration techniques:

- **Data Modeling best practices** (Custom Objects and Relationships)
- **Data Quality management techniques** (Validation Rules and Error Handling)
- **Lightning App Builder proficiency** for creating intuitive user interfaces.
- **Record-Triggered Flow development** for declarative business process automation.
- **Apex programming and Trigger implementation** for complex, custom business logic.
- **Asynchronous Apex and Batch processing** (Scheduled Loyalty Update) for handling large data volumes efficiently.