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

PROJECT OVERVIEW

This paper presents the project entitled HandsMen Threads: Elevating the Art of Sophistication in Men’s Fashion, a dynamic organization in the fashion industry that utilizes state-of-the-art technology and Customer Relationship Management (CRM) through Salesforce, designed to revolutionize the organization’s data management and customer relations. As the company continues to grow, the need to manage customer data, track orders, personalize services, and ensure data integrity, reliability, and security has become increasingly complex. Operational delays and potential loss for business may have occurred if the organization had still heavily relied on manual processes (e.g., manual updates, fragmented tools, and inconsistent data). With Salesforce CRM implementation, business data will be centralized and managed into a unified and scalable platform to provide satisfaction to customers with the use of automation, timely communication, and personalized loyalty rewards.

OBJECTIVES

The primary objective of this Salesforce CRM is to implement and develop a customized and scalable system that supports the business’s operational needs. The aim of this project is to streamline and enhance data management by centralizing customer information, orders, inventory into one reliable and scalable platform. With the implementation of automations such as order confirmations, low stock alerts, loyalty program updates, and scheduled batch processing, the CRM will greatly reduce the manual work, eliminate human errors, and support faster business processes.

These objectives directly contribute to key business value, including:

- Enhanced customer management support with the use of timely and automated communication.
- More efficient operational tasks by minimizing manual processes and speeding up workflows.
- More reliable and accurate inventory control, reducing shortages and delays.

- Faster order processing, resulting in smoother daily operations.
- Stronger decision-making capabilities through access to real-time, consistent data.
- Higher customer satisfaction and retention, supported by personalized loyalty program updates.
- Long-term scalability, enabling the CRM to grow alongside the business.

PHASE 1: REQUIREMENT ANALYSIS AND PLANNING

Understanding Business Requirements

HandsMen Threads requires a CRM solution that addresses both operational inefficiencies and customer engagement challenges. Users currently face difficulties such as manual order confirmations, inconsistent tracking of loyalty rewards, delayed stock alerts, and cumbersome bulk order updates. A key requirement of this project is the maintenance of data integrity directly from the user interface (UI), ensuring that all information is accurate, consistent, and reliable for decision-making. The system will automate critical business processes: customers will receive automated order confirmations, loyalty statuses will update dynamically based on purchase history, proactive stock alerts will notify the warehouse when inventory is low, and scheduled bulk order updates will streamline end-of-day operations. These features collectively enhance operational efficiency, minimize errors, and improve customer satisfaction.

Project Scope and Objectives

The project scope and objectives for HandsMen Threads' Salesforce CRM define the key deliverables and intended outcomes of the implementation. The following are the main aspects of the project scope and objectives, highlighting the processes to be automated, the system features to be implemented, and the business value to be achieved.

Project Scope

- Centralization of customer data, orders, and inventory into a unified Salesforce platform.
- Automation of key processes including order confirmations, loyalty status updates (Bronze, Silver, Gold), low stock alerts, and bulk order processing.
- Customization of Salesforce objects, fields, validation rules, and page layouts to meet business workflows.

- Development of backend logic using Apex, triggers, and batch processes where automation cannot be achieved declaratively.
- Implementation of reporting tools and dashboards for real-time business insights.
- Integration with email notifications to improve customer engagement.

Project Objectives

- Ensure data accuracy and integrity directly from the UI for reliable decision-making.
- Enhance customer satisfaction and engagement through timely automated communications.
- Improve operational efficiency by reducing manual tasks and human errors.
- Enable real-time inventory tracking to prevent stock shortages and optimize warehouse management.
- Support scalability and future growth through a flexible and robust CRM architecture.
- Provide administrative and end-user training, ensuring smooth adoption and long-term sustainability.

Data Model

The figure 1 below illustrates the Entity–Relationship Diagram (ERD) of the HandsMen Threads Salesforce CRM, showing how key business records interact within the system. Each customer can have multiple orders, and every order is linked to specific products. Products are further connected to inventory records to monitor stock levels across various locations. Additionally, customers can be associated with multiple marketing campaigns. This structured model ensures organized, scalable, and efficient data management, enabling accurate reporting, smooth workflow automation, and seamless integration between business processes.

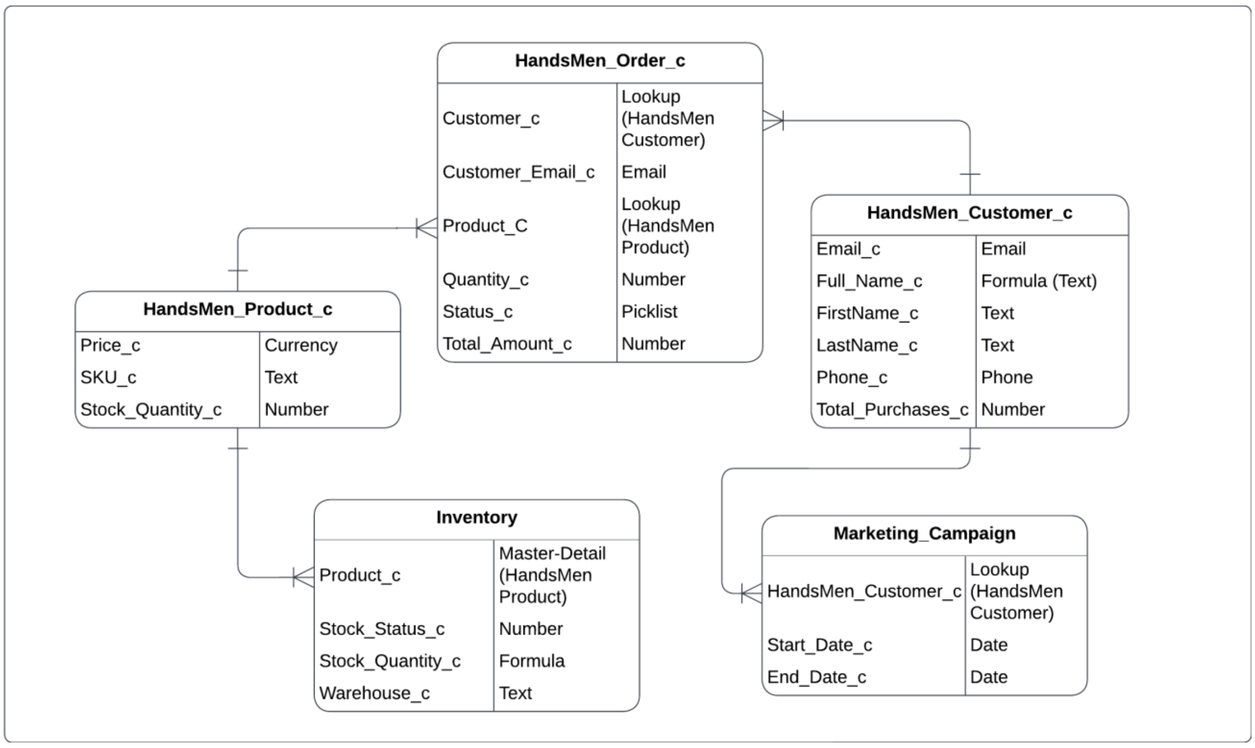


Fig. 1. Entity Relationship Diagram of HandsMen Threads

Security Model

The figure 2 illustrates CRM employs a role- and profile-based security framework to control access to objects and data. For example, Sales Managers have full access to customer and order records, Marketing Team members can read customer information and manage campaigns, and Inventory Managers have read and edit access to product and stock data. The platform also uses standard Salesforce profiles, permission sets, and role hierarchy to ensure that users can only view and modify records relevant to their roles. This security design safeguards sensitive business data while enabling effective collaboration across teams.

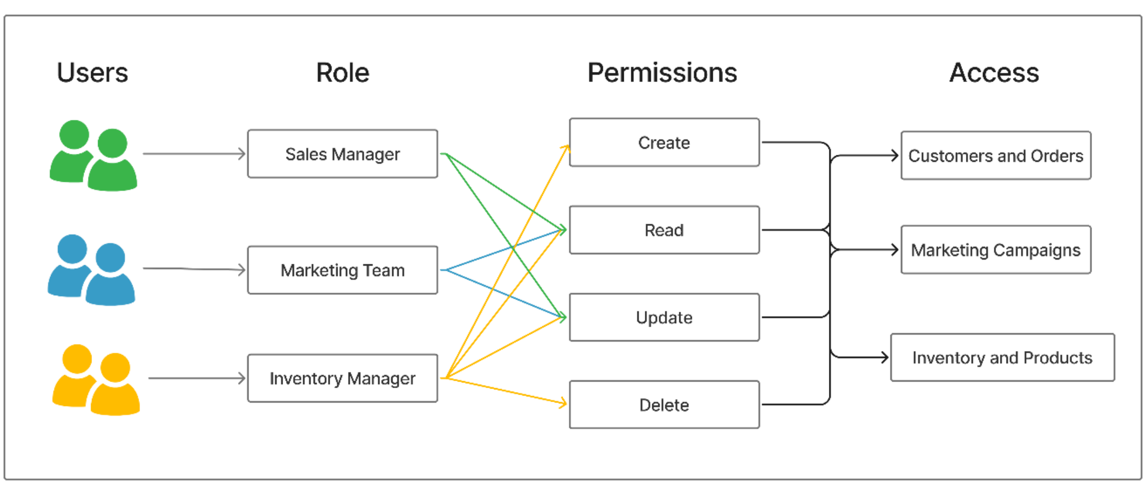


Fig. 2. Role-Based Access Control of HandsMen Threads

Stakeholders Mapping

The stakeholder structure of the HandsMen Threads Salesforce CRM follows a top-down organizational hierarchy, ensuring clear responsibility, accountability, and efficient data flow across business functions. The figure 3 below illustrates how roles are distributed and how each department contributes to the CRM's operations.

Key Stakeholders:

1. **CEO -** Holds the highest authority and has full visibility and control across all CRM data and processes. Oversees strategic decisions and system-wide performance.
2. **Sales Manager -** Reports directly to the CEO and manages customer records, sales transactions, and order-related processes. Ensures data accuracy in customer and order management.
3. **Marketing Team -** Reports to the CEO and is responsible for managing marketing campaigns, tracking customer engagement, and analyzing campaign performance within the CRM.
4. **Inventory Manager -** Also reports to the CEO and ensures proper stock tracking, product updates, and timely restocking using the inventory and product data maintained in the CRM.

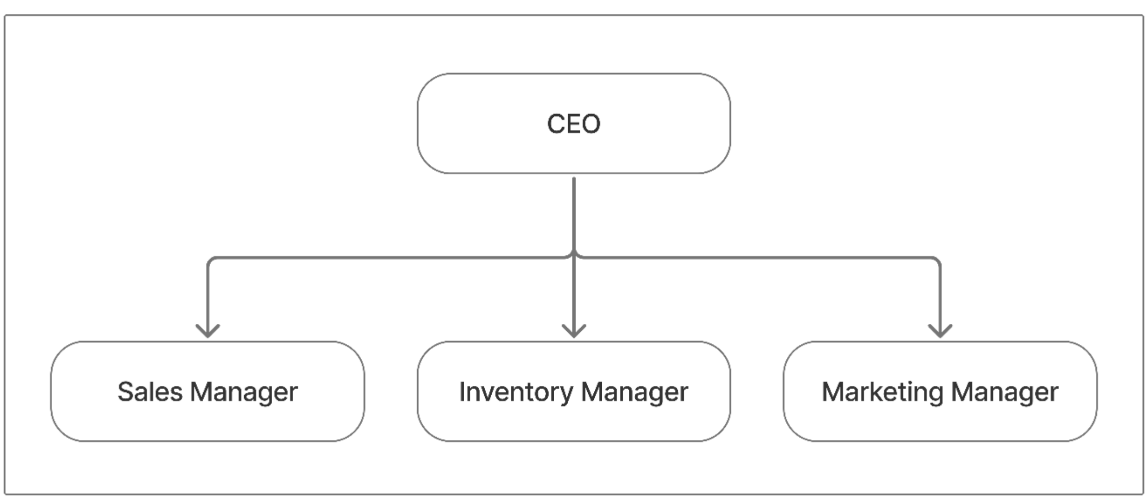


Fig. 3. Role Hierarchy of HandsMen Threads

PHASE 2: SALESFORCE DEVELOPMENT – BACKEND AND CONFIGURATIONS

Environment Setup and DevOps Workflow

To setup the environment, a special Developer Org account was created (See figure 4). The email sent a verification to create and verify the account (See figure 5). In addition, the password has been reset to a new and unique password and granted access to the Salesforce Setup Page.

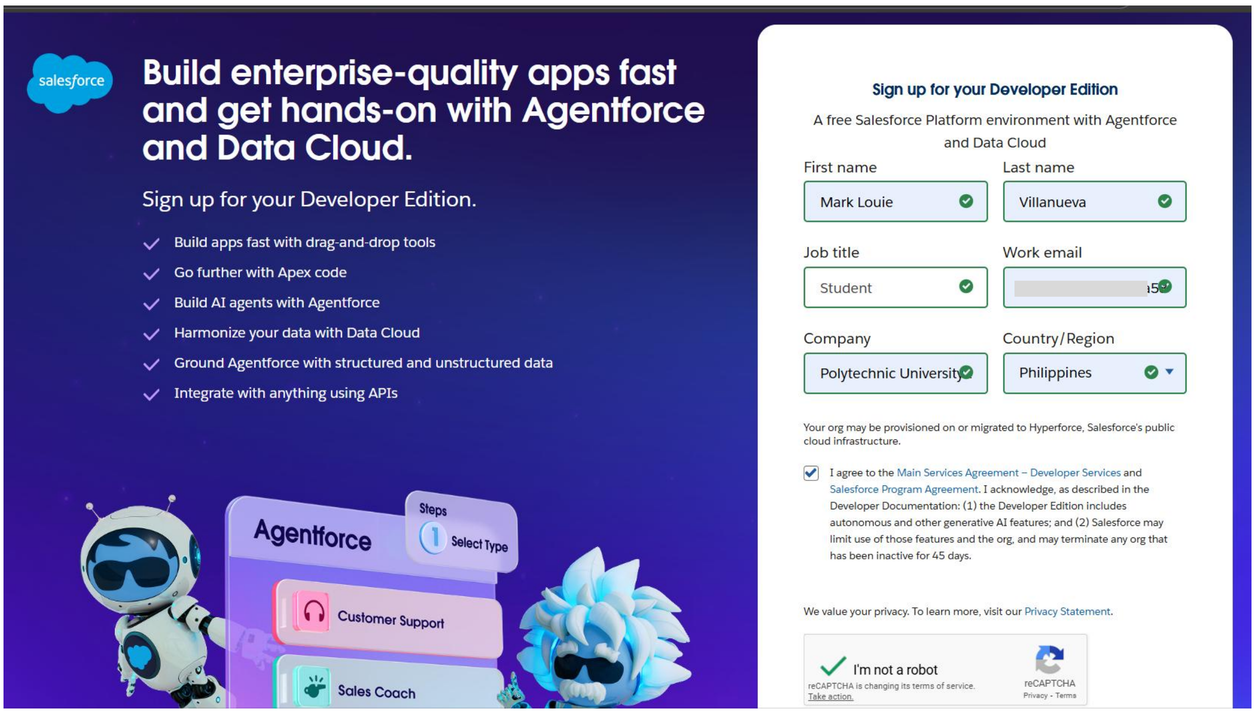


Fig. 4. Developer Edition Sign Up

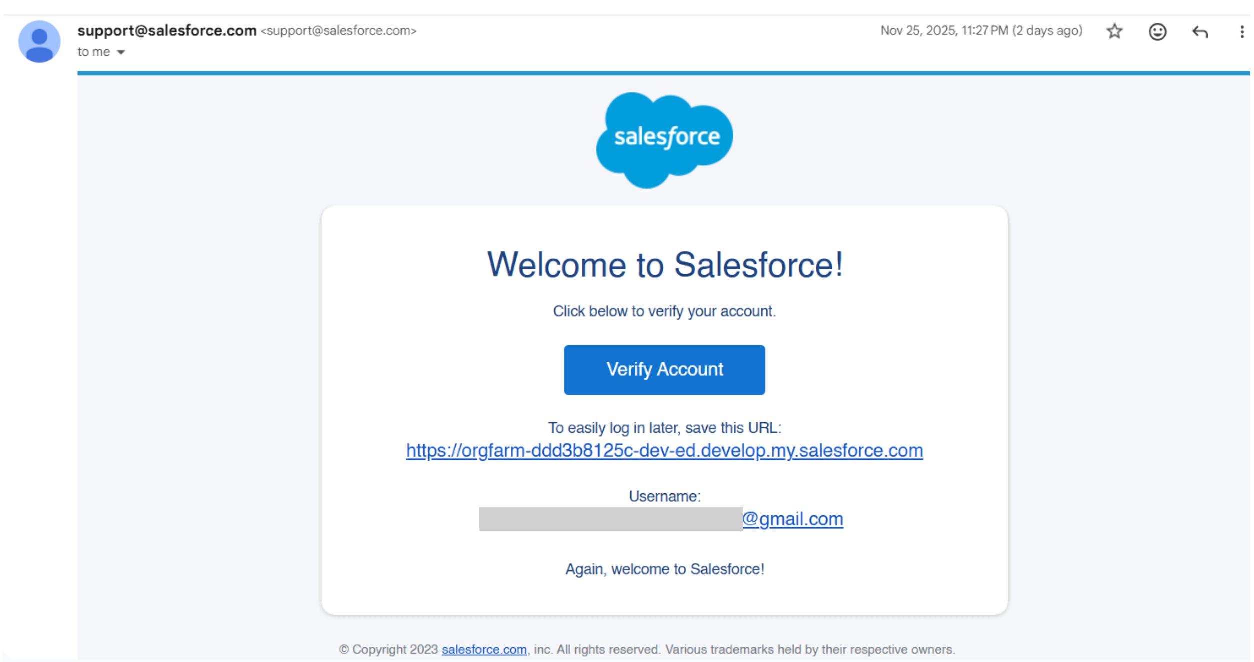


Fig. 5. Verify account sent through email

Customization of Objects, Fields, Validation Rules, Automation

To model business operations the following custom objects and fields were created:

1. **HandsMen Customer –** Customer details such as email, phone, loyalty status, total purchaes, were stored.
2. **HandsMen Order –** Customer orders such as product, order status, and total amount were stored.
3. **HandsMen Product –** Product details such as SKU, price, and quantity were stored.
4. **Inventory –** Stock level was tracked for shortages and tracked warehouse location.
5. **Marketing Campaign –** For managing promotions and campaigns which includes customer name, start and end date.

Table 1 illustrates the object name, API name, and their key fields of each custom object of HandsMen Threads.

OBJECT NAME	API NAME	KEY FIELDS
HANDSMEN CUSTOMER	HandsMen_Customer__c	Email__c, FirstName__c, Full_Name__c, LastName__c, Loyalty_Status__c, Phone__c, Total_Purchases__c
HANDSMEN PRODUCT	HandsMen_Product__c	Customer__c, Customer_Email__c, Product__c, Quantity__c, Status__c, Total_Amount__c
HANDSMEN ORDER	HandsMen_Order__c	Order__c, Price__c. SKU__c, Stock_Quantity__c
INVETORY	Inventory__c	Product__c, Stock_Quantity__c, Stock_Status__c, Warehouse__c
MARKETING CAMPAIGN	Marketing_Campaign__c	Start_Date__c, End_Date__c, HandsMen_Customer__c

Table 1. Custom Objects and their Key Fields

Table 2 presents the validation rules implemented across key objects to ensure consistent and error-free data entry. For HandsMen_Order__c, the rule blocks any order record with a total amount less than or equal to zero, protecting the accuracy of sales and financial data. The Inventory__c object restricts stock values from being zero or negative to maintain proper inventory tracking. Meanwhile, HandsMen_Customer__c includes a validation that requires email addresses to contain a valid domain format (such as "@gmail.com"), ensuring dependable customer communication records.

Object Name	Field	Validation Rule
HandsMen Customer	Email__c	NOT CONTAINS (Email__c, “@gmail.com”)
HandsMen Order	Total_Amount__c	Total_Amount__c ≤ 0
Inventory	Stock_Quantity__c	Stock_Quantity__c ≤ 0

Table 2. Validation rules

Automation Flows

Order Confirmation Flow

Created a flow in flow builder for order confirmation flow using a record-trigger flow that is triggered when the customer’s order is confirmed (See figure 6). An automated email will be sent to the customer informing their order is confirmed.

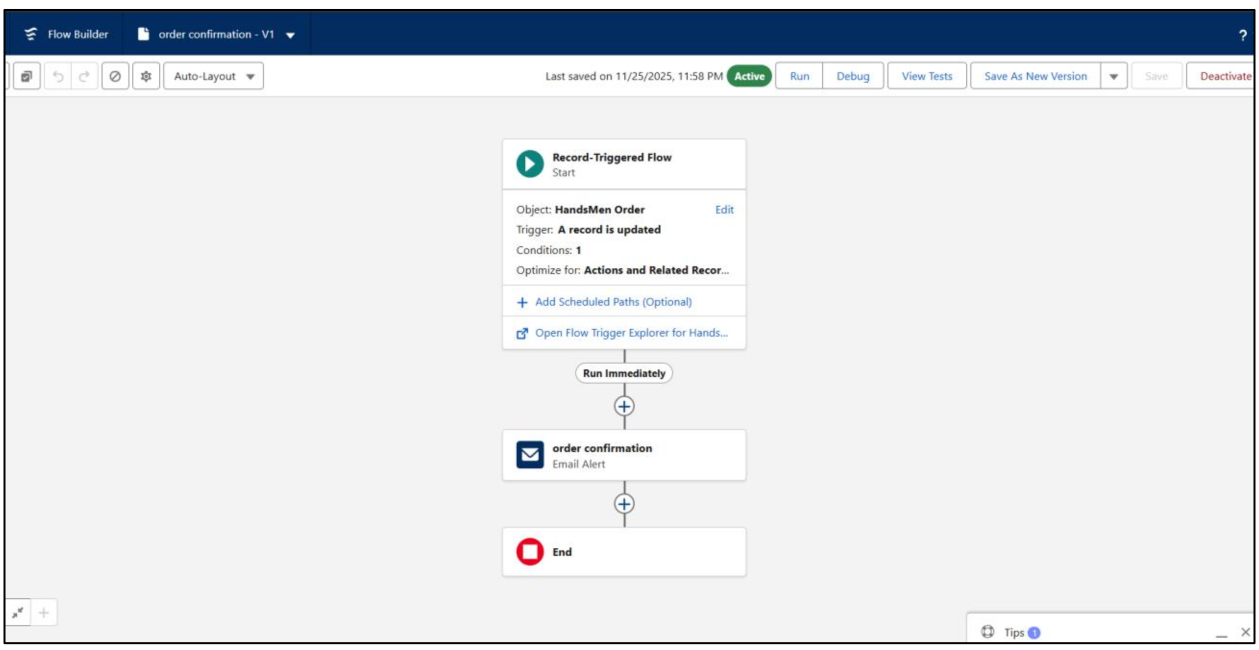


Fig. 6. Order Confirmation Flow in Flow Builder

Low Stock Alert Flow

Created a flow in flow builder for low stock alert flow using a record-trigger flow that is triggered when the inventory stock quantity is less than 5 (See figure 7). An automated email is also sent to the inventory manager to alert them.

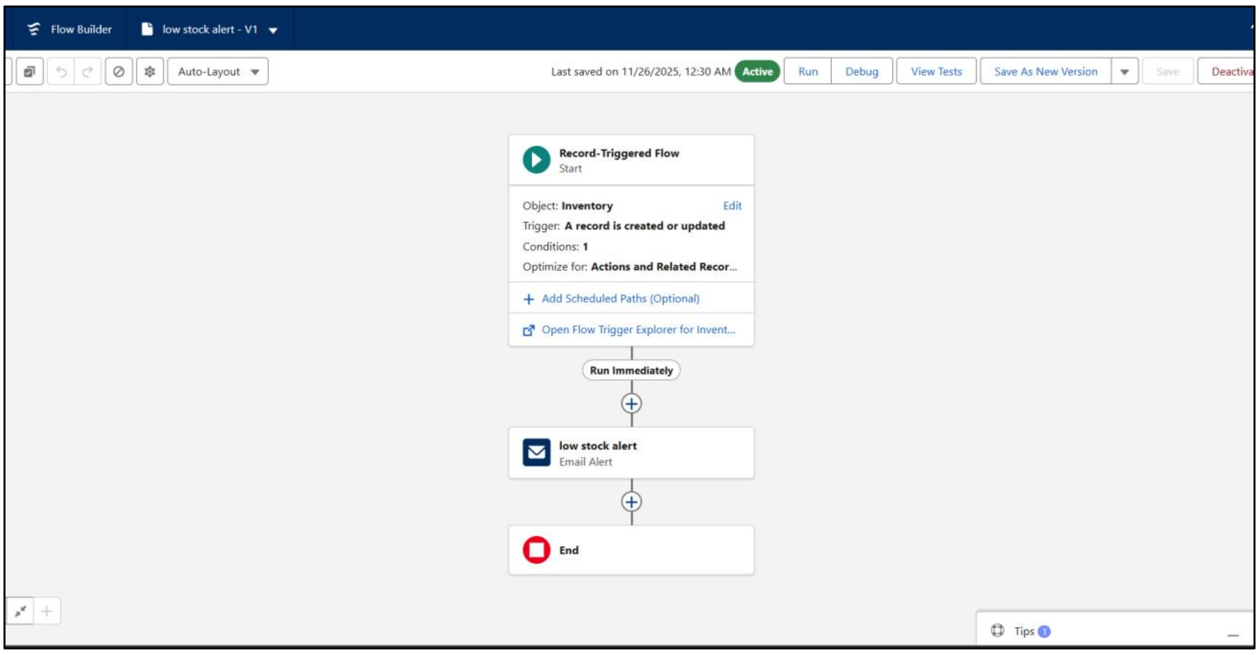


Fig. 7. Low Stock Alert Flow in Flow Builder

Loyalty Status Update

Created a flow in flow builder for loyalty status update using a schedule flow that is triggered when the inventory stock quantity is less than 5 (See figure 8). An automated email is also sent to the customer informing their status update whether they belong to Bronze, Silver, or Gold status based on their total purchases.

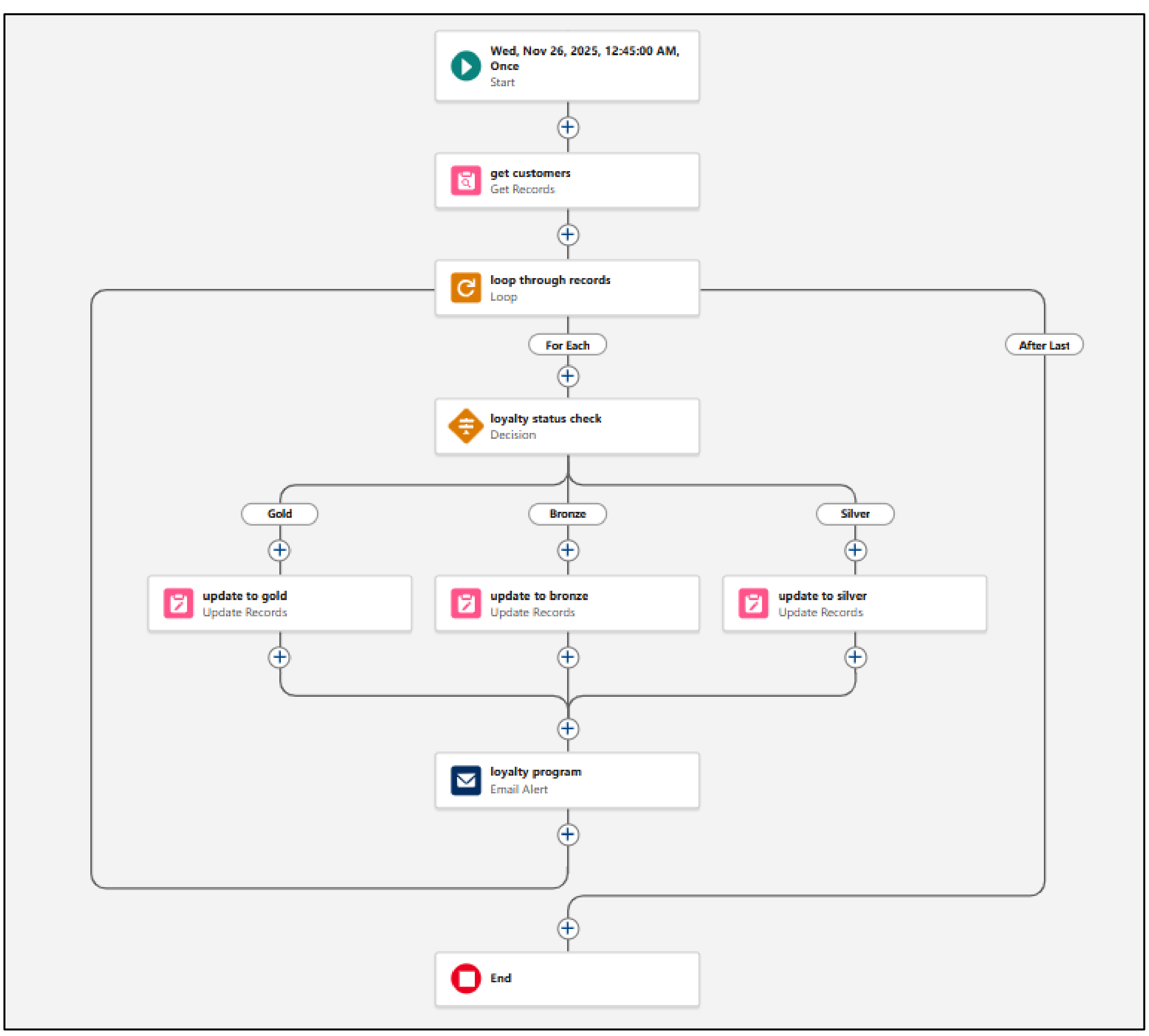


Fig. 8. Loyalty Status Update using Scheduled Flow in Flow Builder

Apex Classes and Triggers

1. **Update Order Total** – automatically trigger the total amount once the order record was saved (See figure 9).

StockDeductionTrigger.apxt | OrderTotalTrigger.apxt

Code Coverage: None | API Version: 65

```
trigger OrderTotalTrigger 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;
            }
        }
    }
}
```

Fig. 9. Order Total Apex Trigger using developer console

2. **Stock Deduction** – When an order is created, it reflects to the stock quantity of the inventory once the order is confirmed (See figure 10).

StockDeductionTrigger.apxt OrderTotalTrigger.apxt

Code Coverage: None API Version: 65

```
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;
    }
}
```

Fig. 10. Stock deduction Apex Trigger using developer console

Batch Jobs Processes

This automated job regularly checks for products with low stock and replenishes their quantity to prevent shortages. It runs on a set schedule, keeping inventory levels accurate and reducing manual workload for the warehouse team (See figure 11).

```
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);}}
```

Fig.11. Inventory Batch Job Process using Apex Class in developer console

PHASE 3: UI/UX DEVELOPMENT AND CUSTOMIZATION

Lightning App Setup

A custom Lightning App is configured to provide a tailored workspace for users. This includes tab arrangement, branding, and navigation setup, ensuring an intuitive user experience (See Figure 12 and 13).

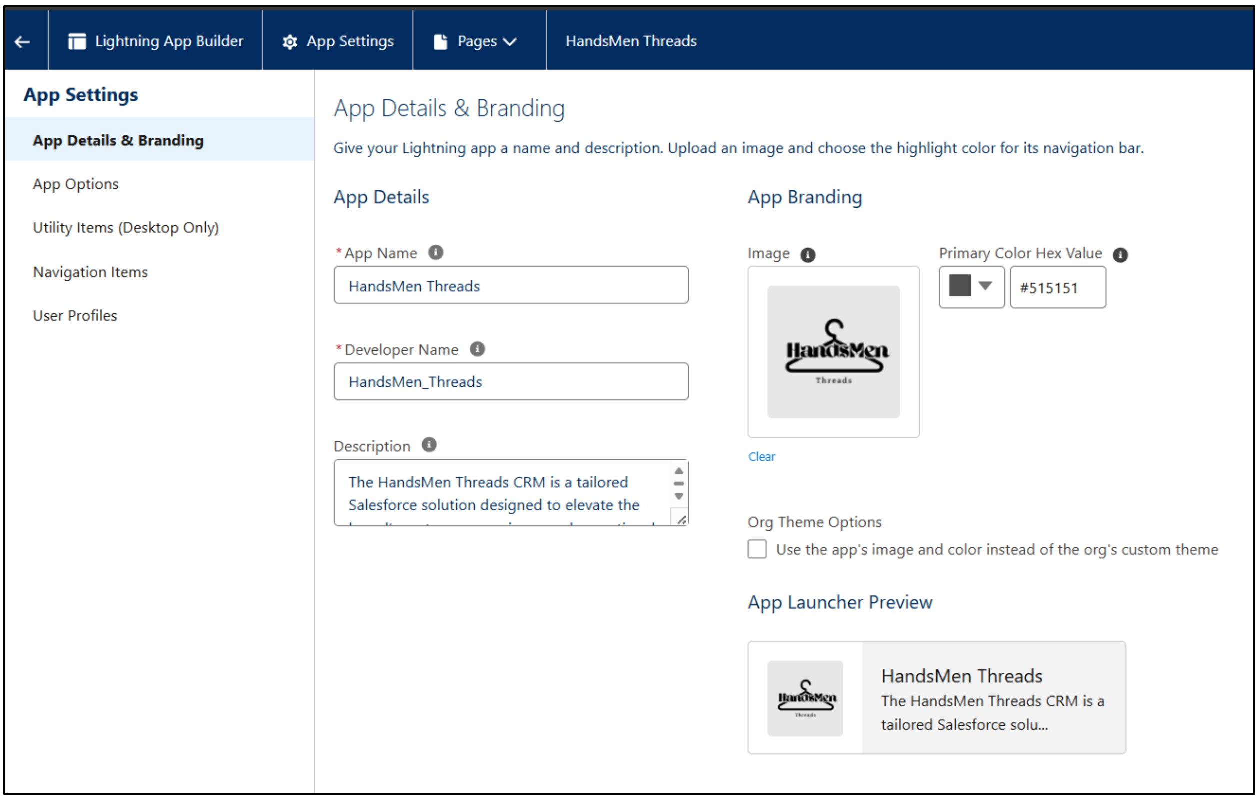


Fig.12. Lightning App Builder through App Manager

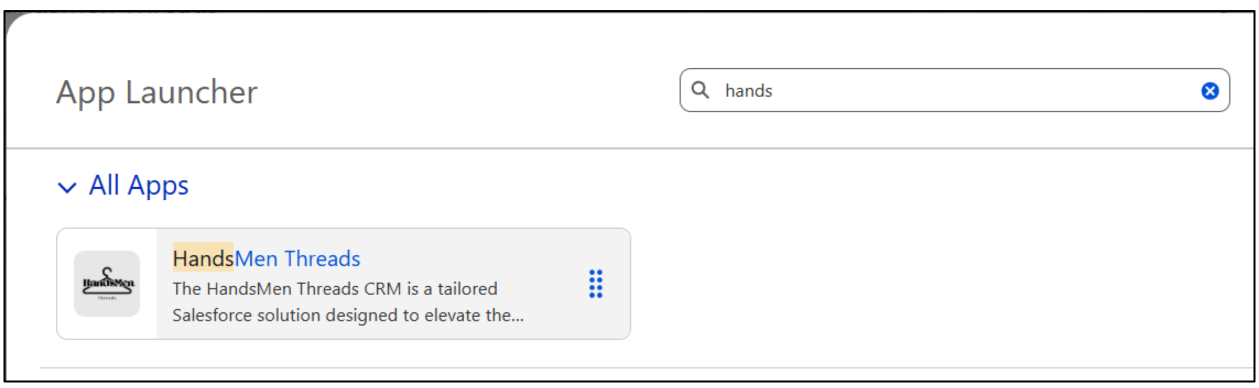


Fig.13. Launching the HandsMen Threads in App Launcher

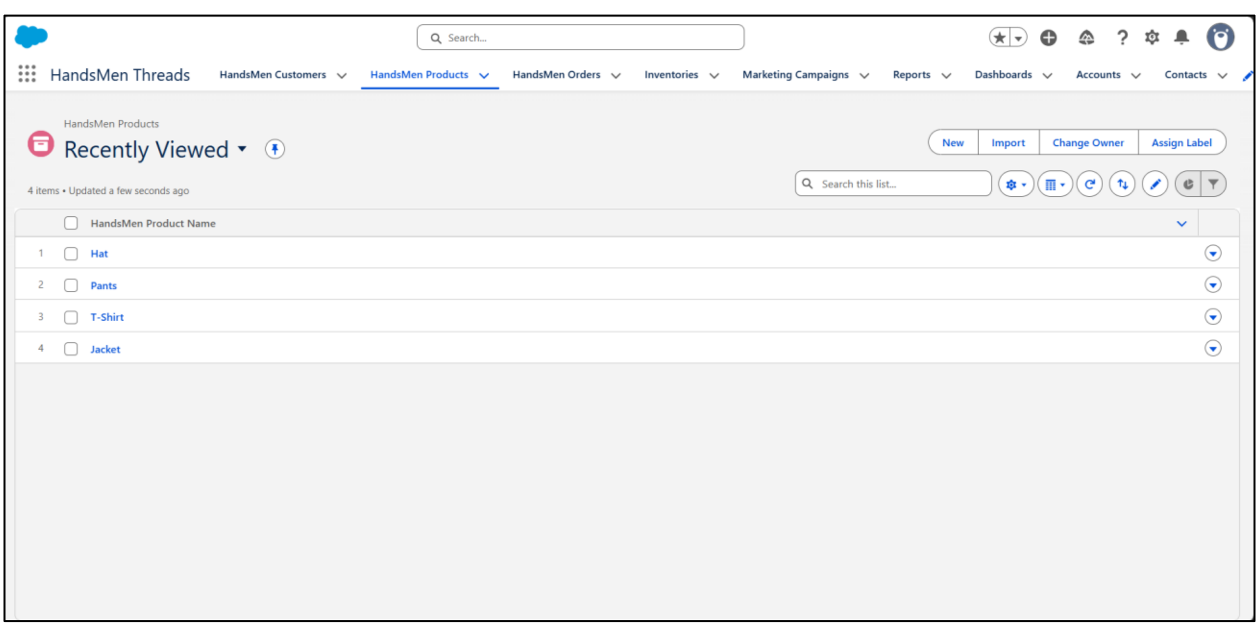


Fig.14 HandsMen Threads Tabs

Page Layouts and Dynamic Forms

Layouts and Dynamic Forms are customized to display relevant fields, sections, and components for different user profiles. This ensures a clean and efficient interface that supports fast data entry.

1. **HandsMen Customer**

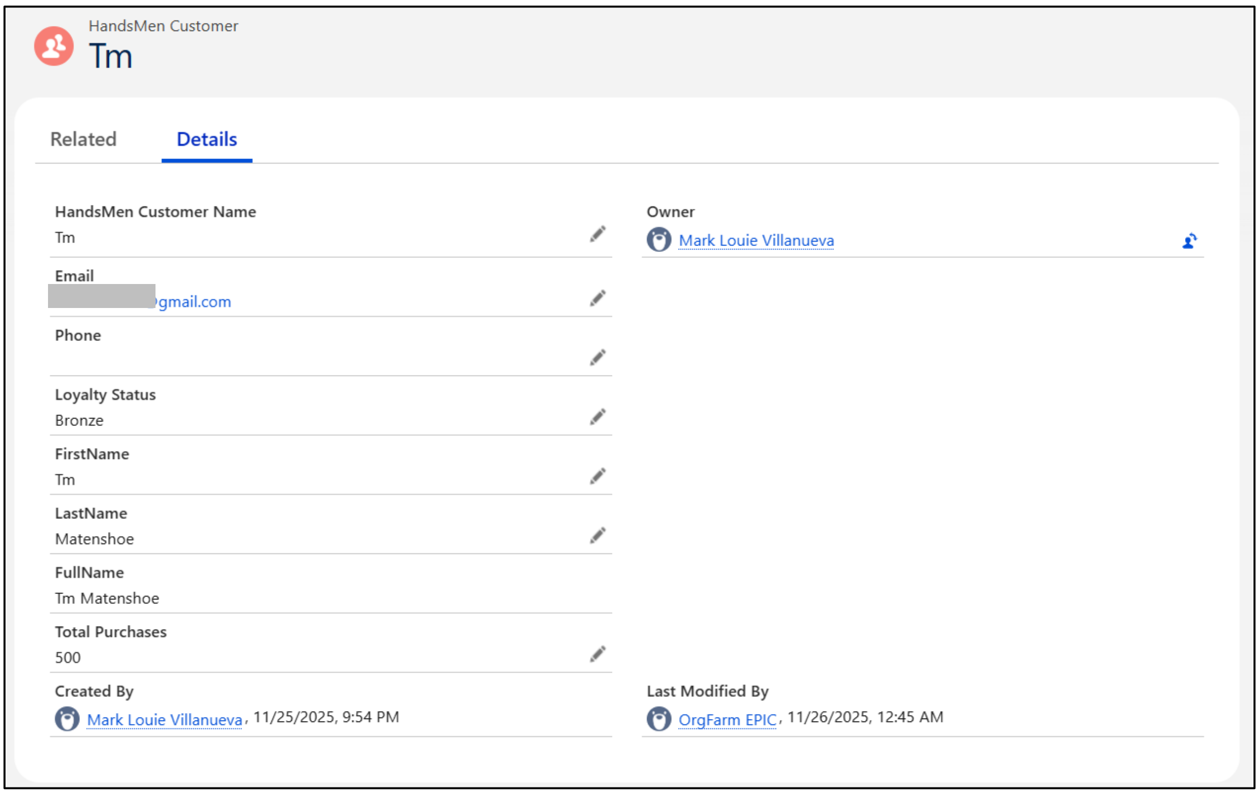


Fig.15. HandsMen Customer Detail Page Layout

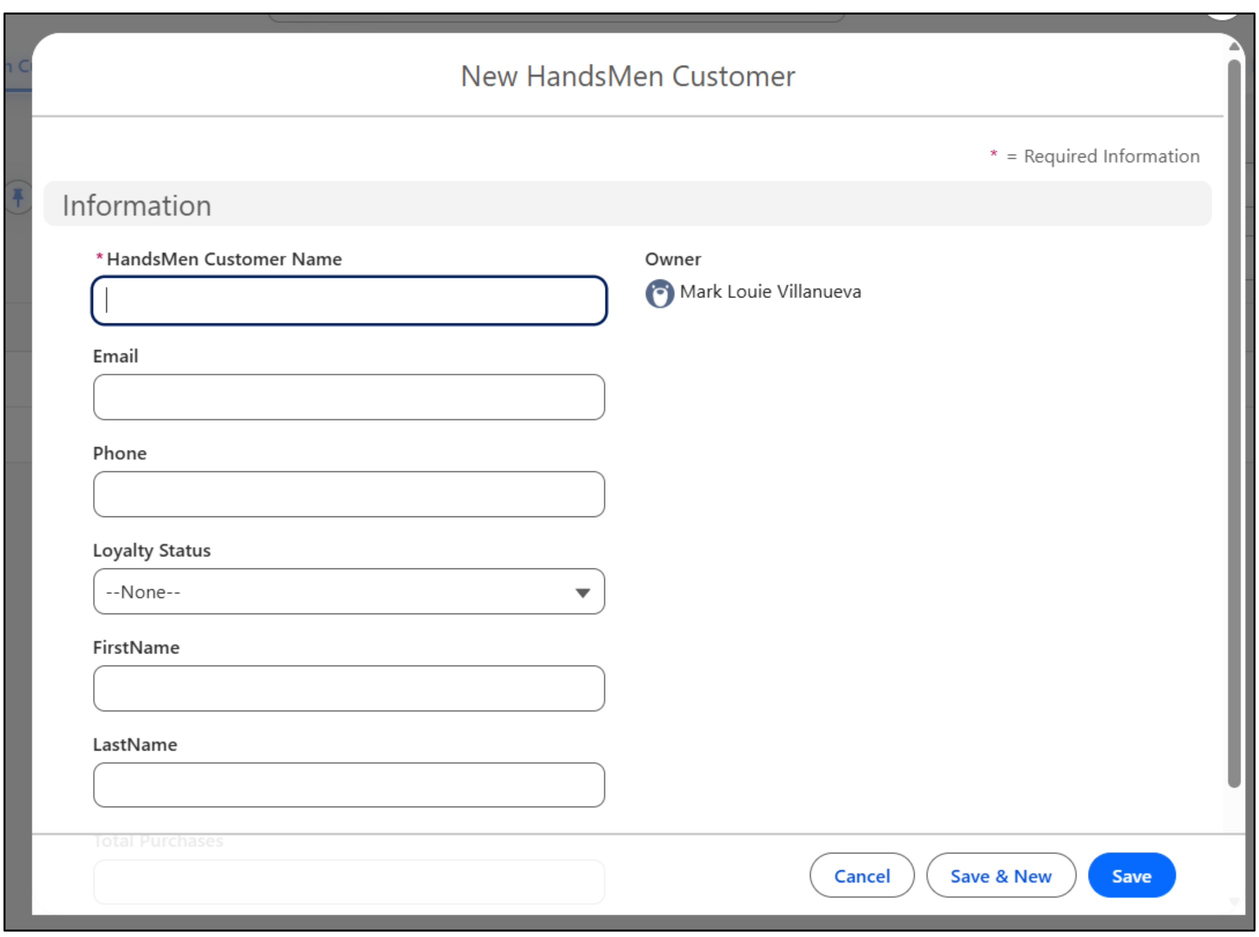


Fig.16. Adding New HandsMen Customer Form

2. HandsMen Order

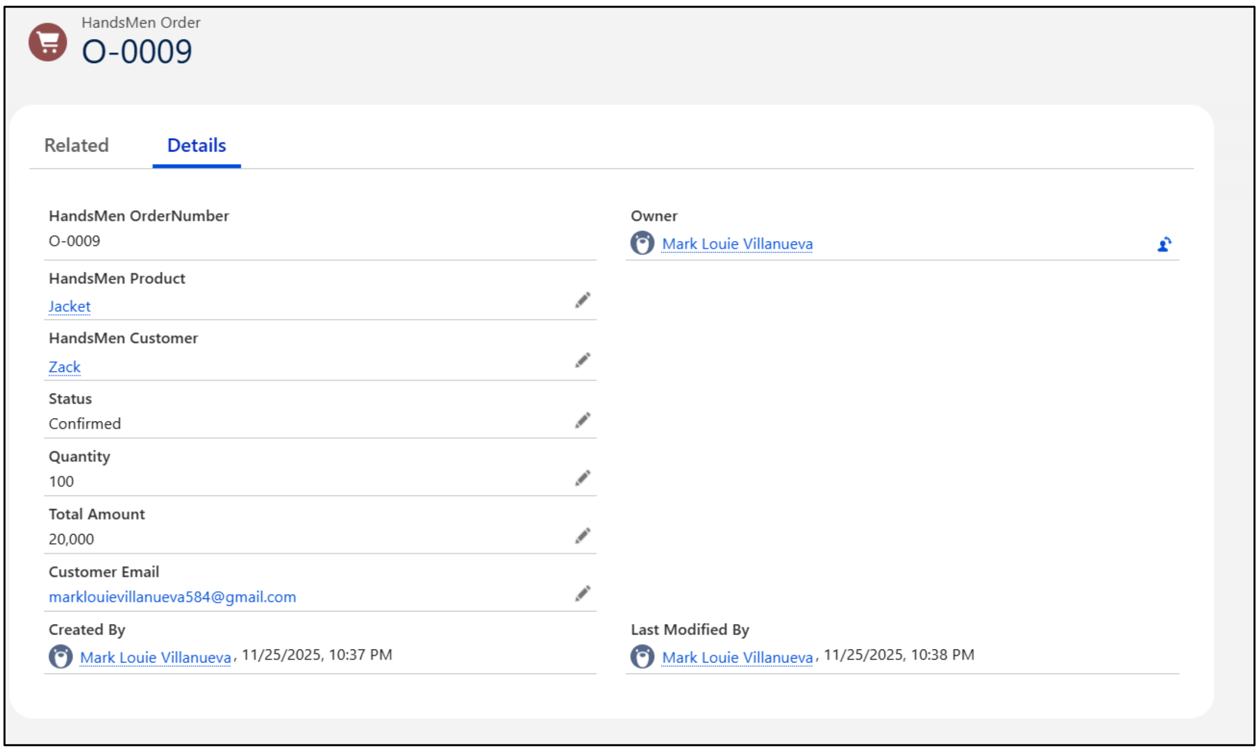


Fig.17. HandsMen Order Detail Page

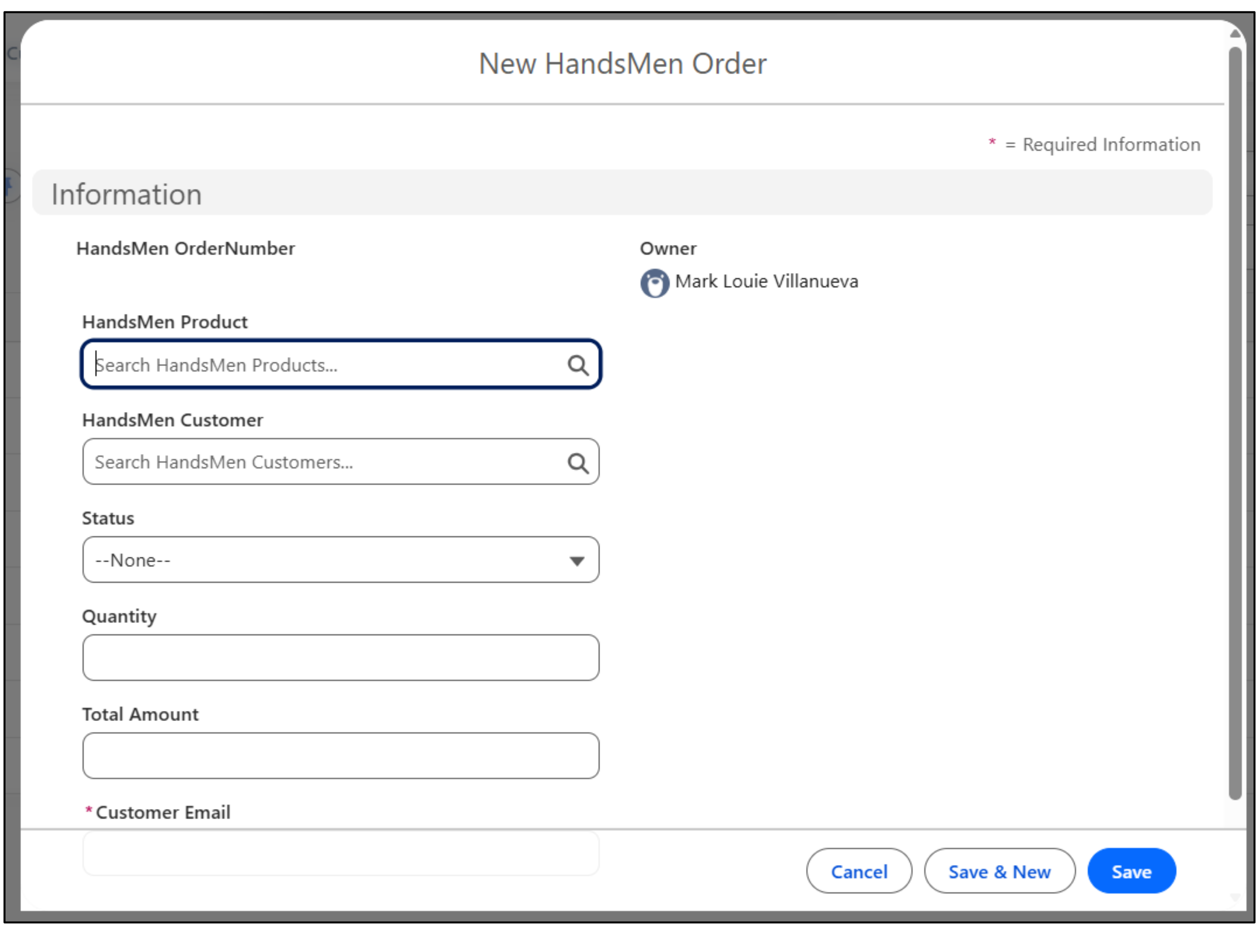


Fig.18. Adding New HandsMen Order Form

3. HandsMen Product

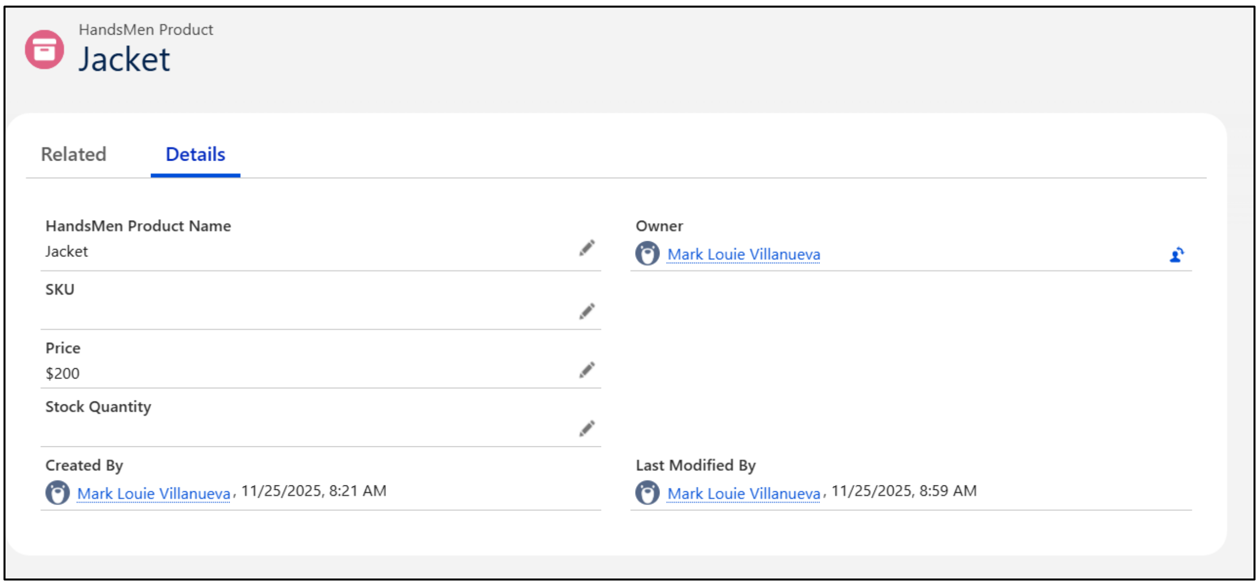


Fig.19. HandsMen Product Detail Page

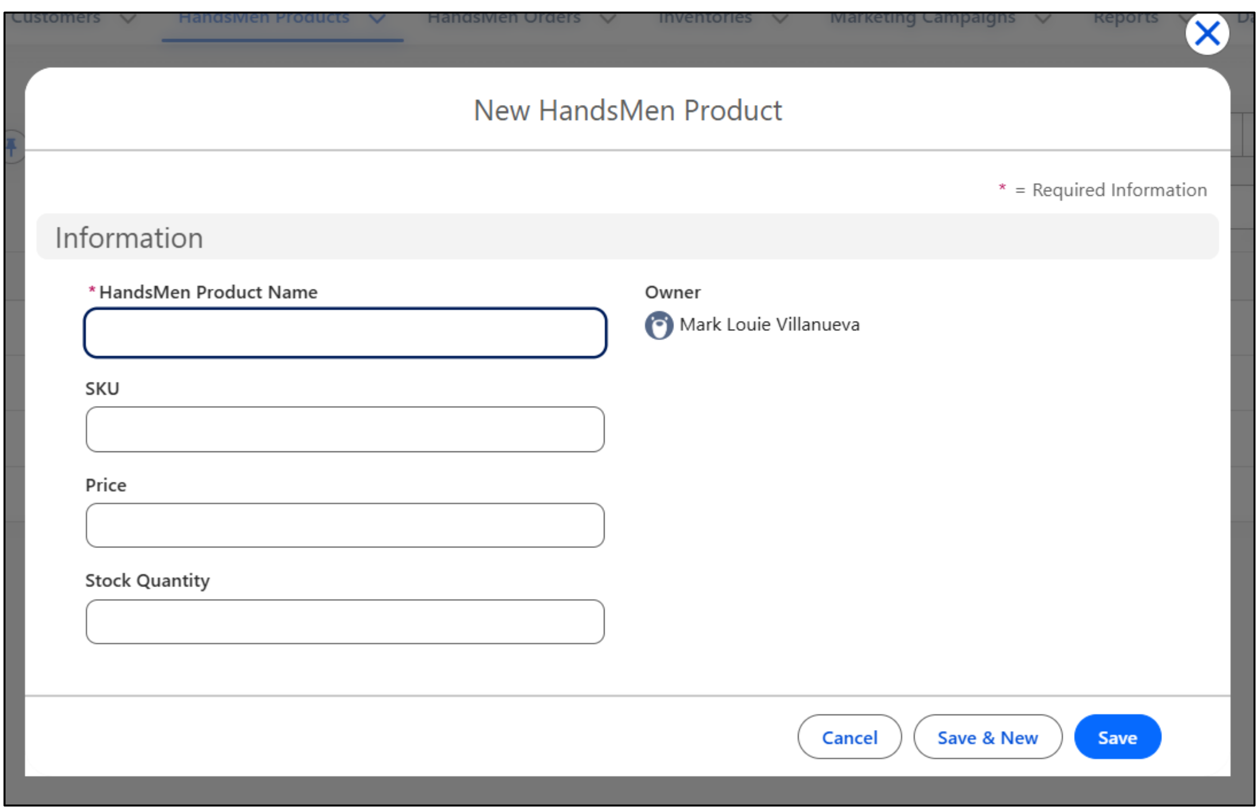


Fig.20. Adding New HandsMen Product Form

4. Inventory

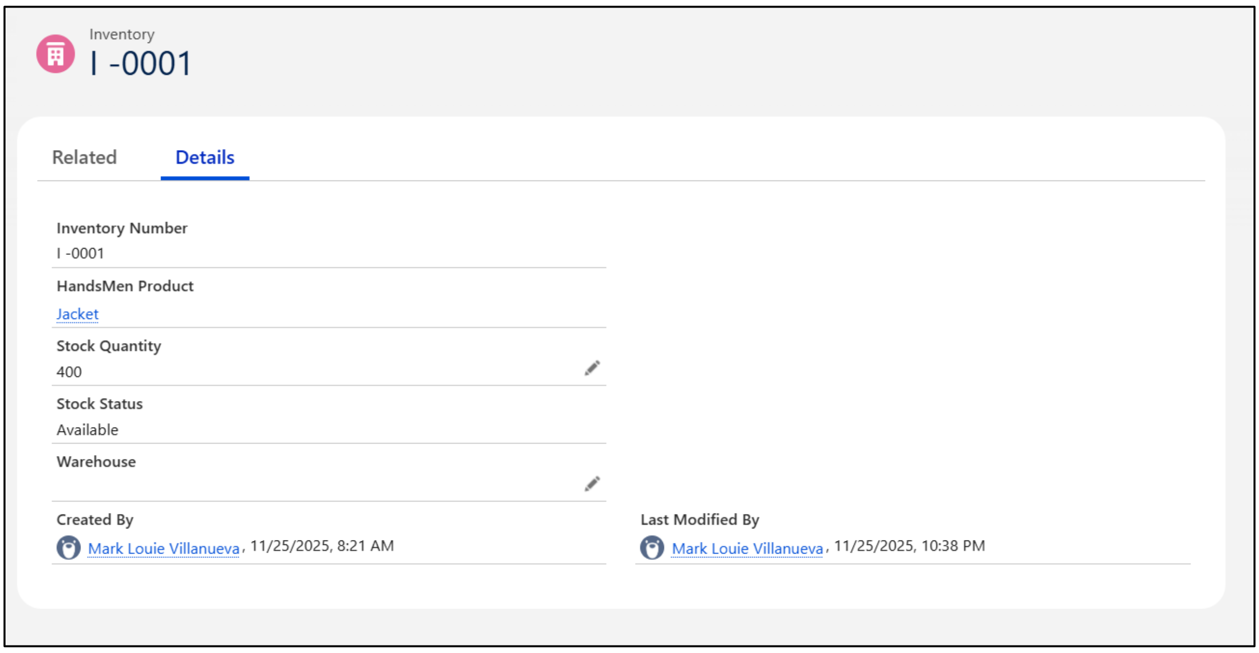


Fig.2. Inventory Detail Page

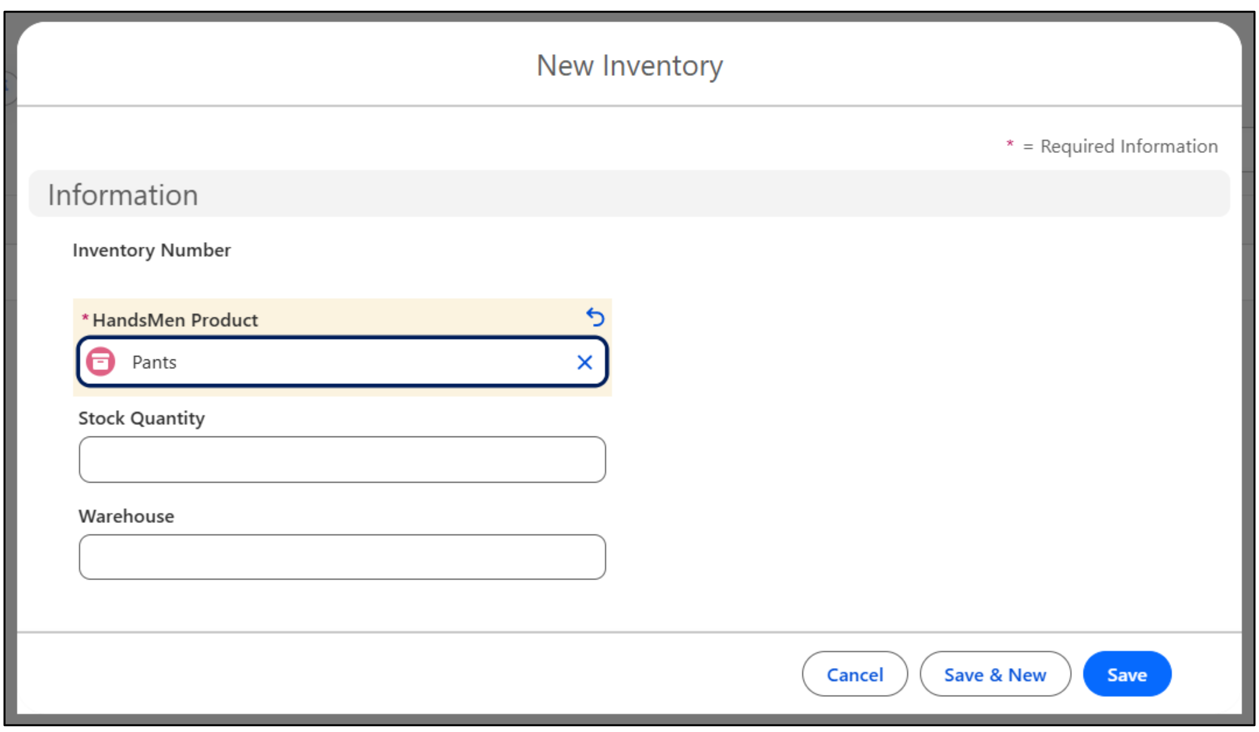


Fig.22. Adding New Inventory Record Form

5. **Marketing Campaign**

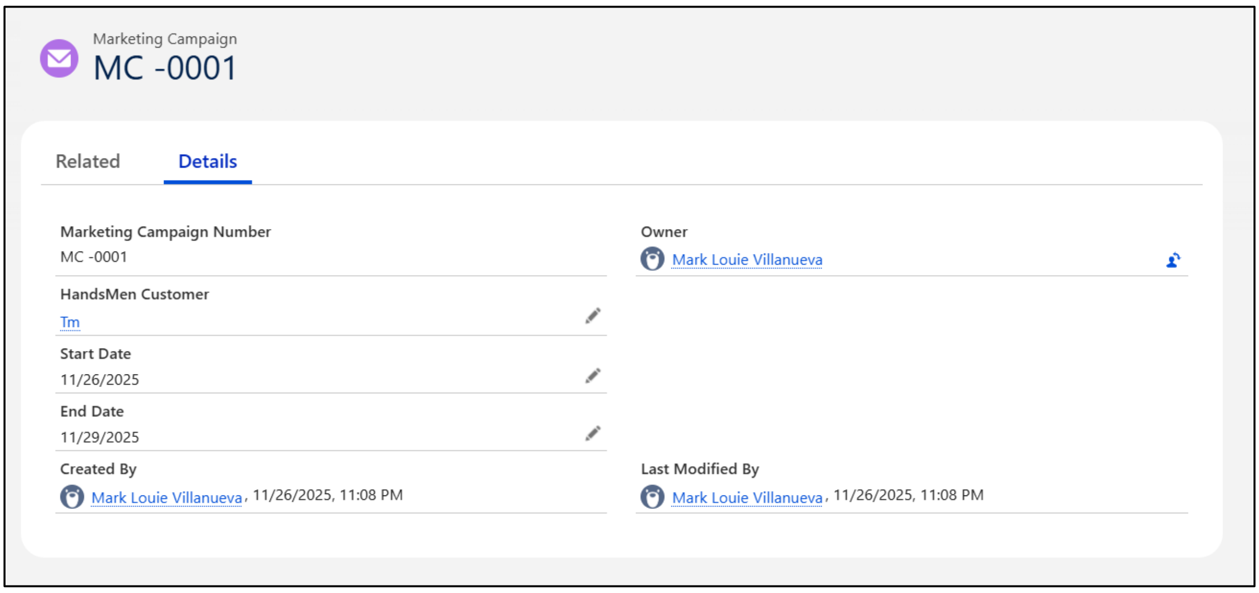


Fig.23. Marketing Campaign Detail Page

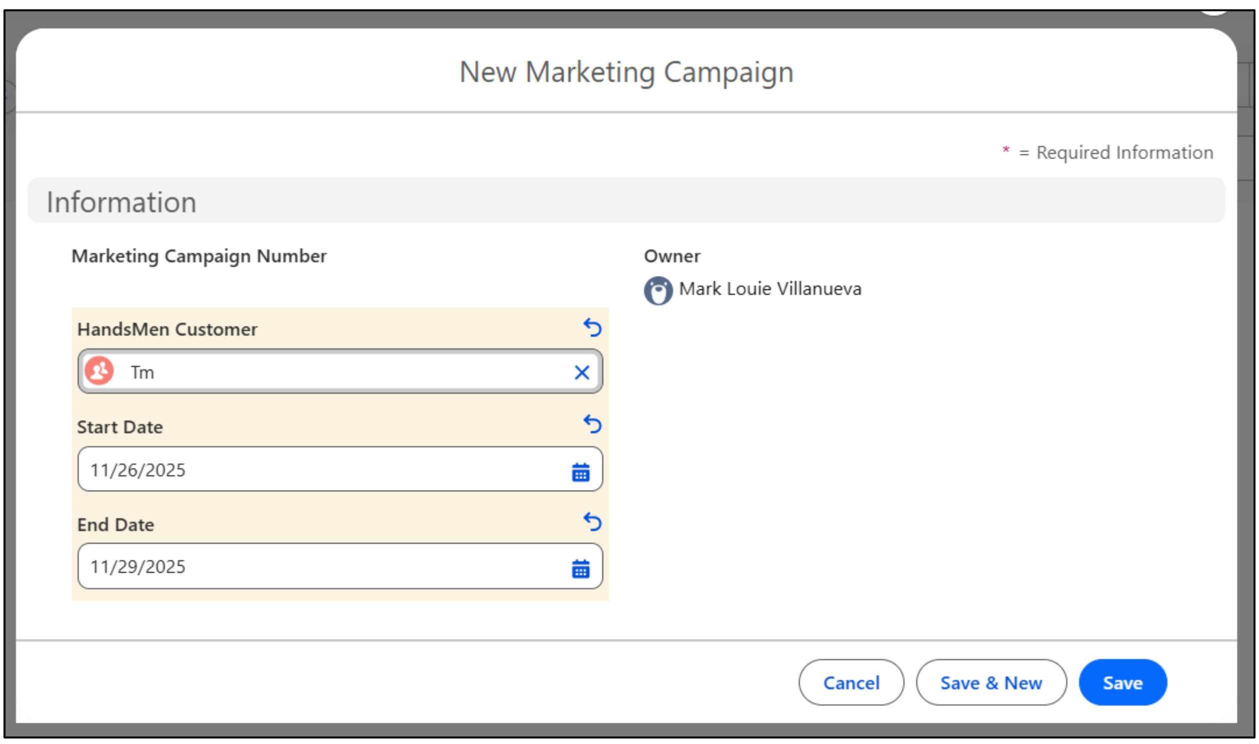


Fig.24. Adding New Marketing Campaign Form

User Management

Users are created and assigned appropriate roles and permissions. This ensures that each user has the correct access levels aligned with their job function.

1. **Niklaus Mikaelson - Sales Management**

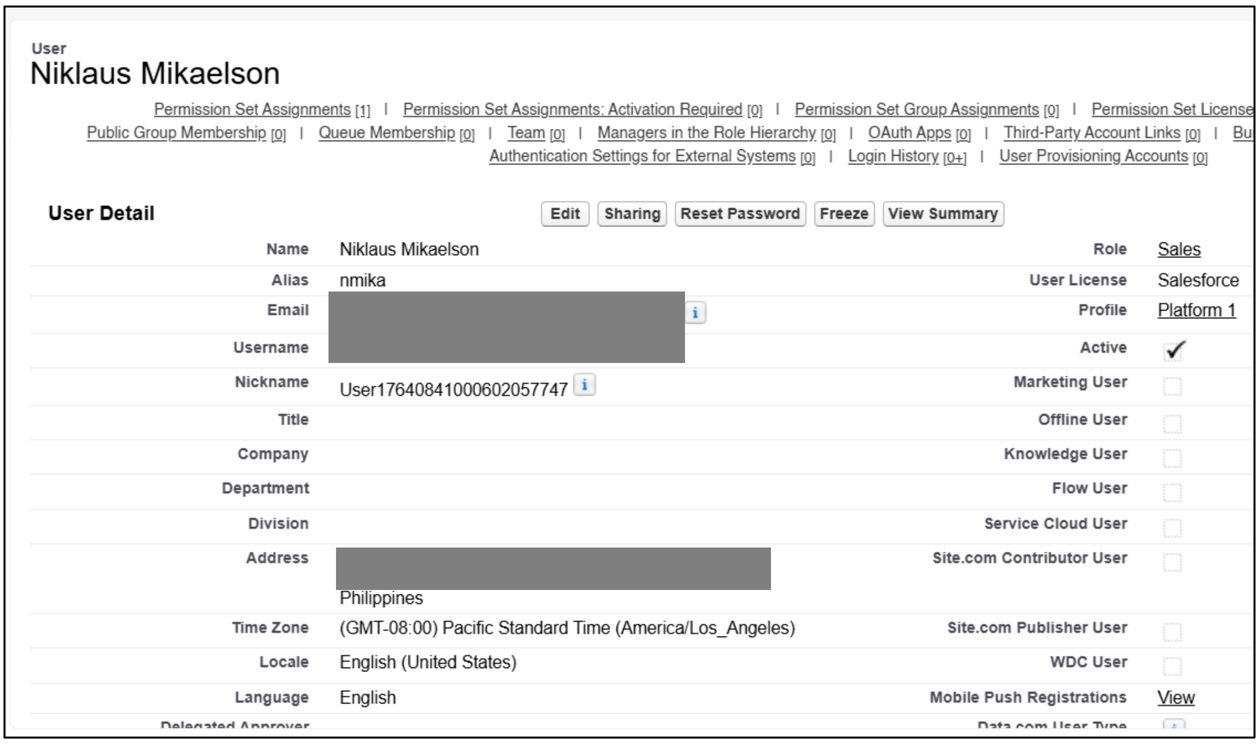


Fig.25. User Management 1

2. **Kol Mikaelson – Inventory Management**

Fig.26. User Management 2

3. **Jake Mikaelson – Marketing Manager**

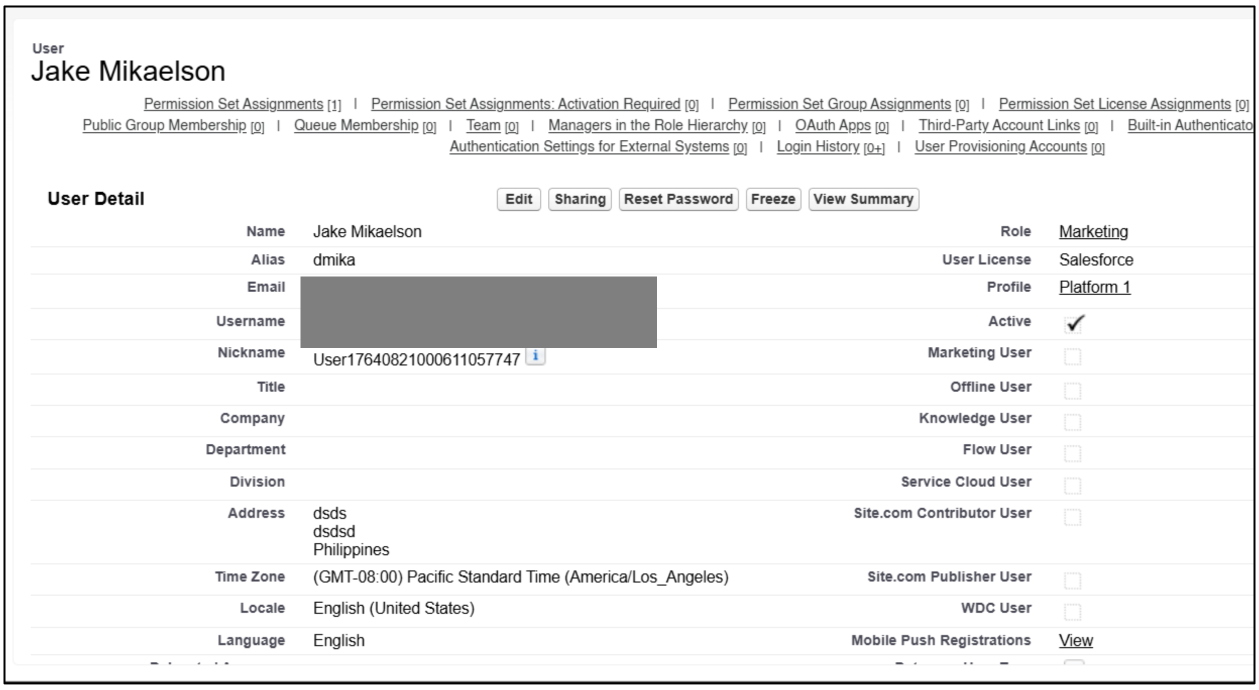


Fig.27. User Management 3

Reports and Dashboard

Reports and dashboards are built to provide real-time insights into sales performance, inventory status, and customer activity. These visuals help stakeholders track metrics and make informed decisions (See Figure 28 & 29) .

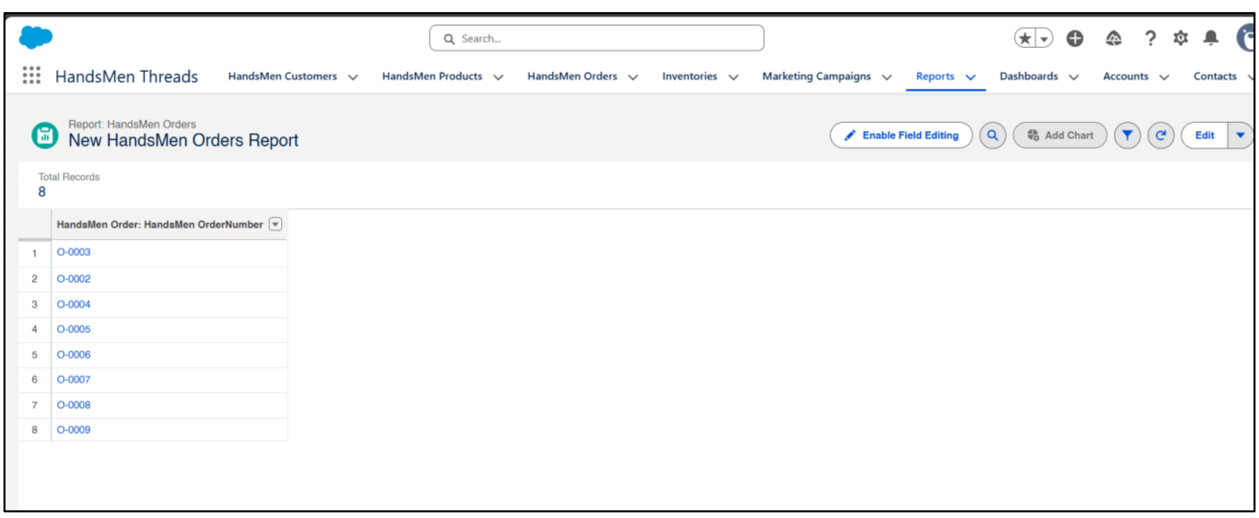


Fig.28. HandsMen Order Report

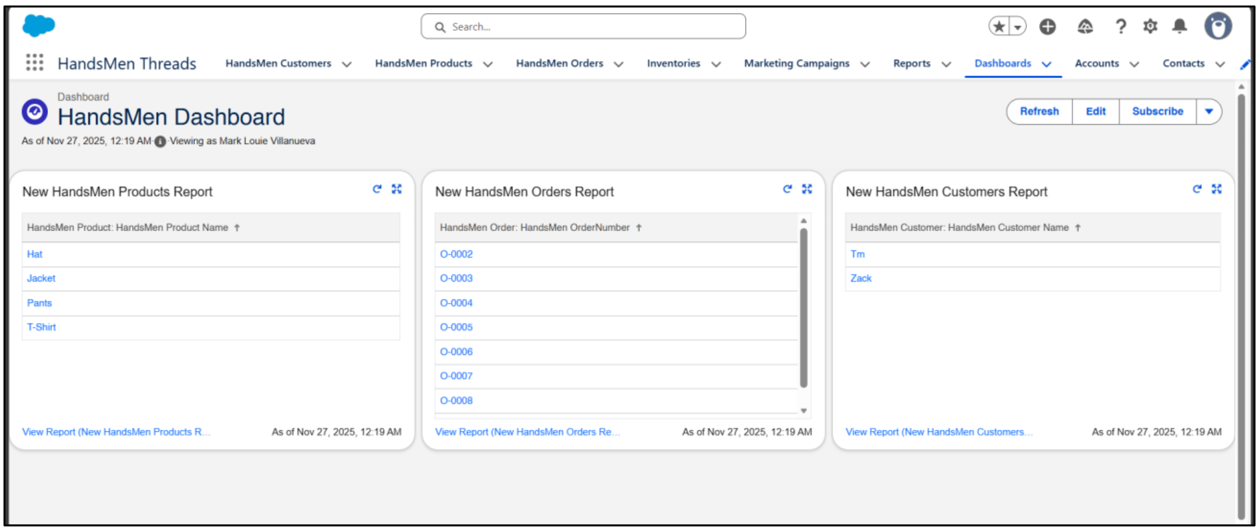
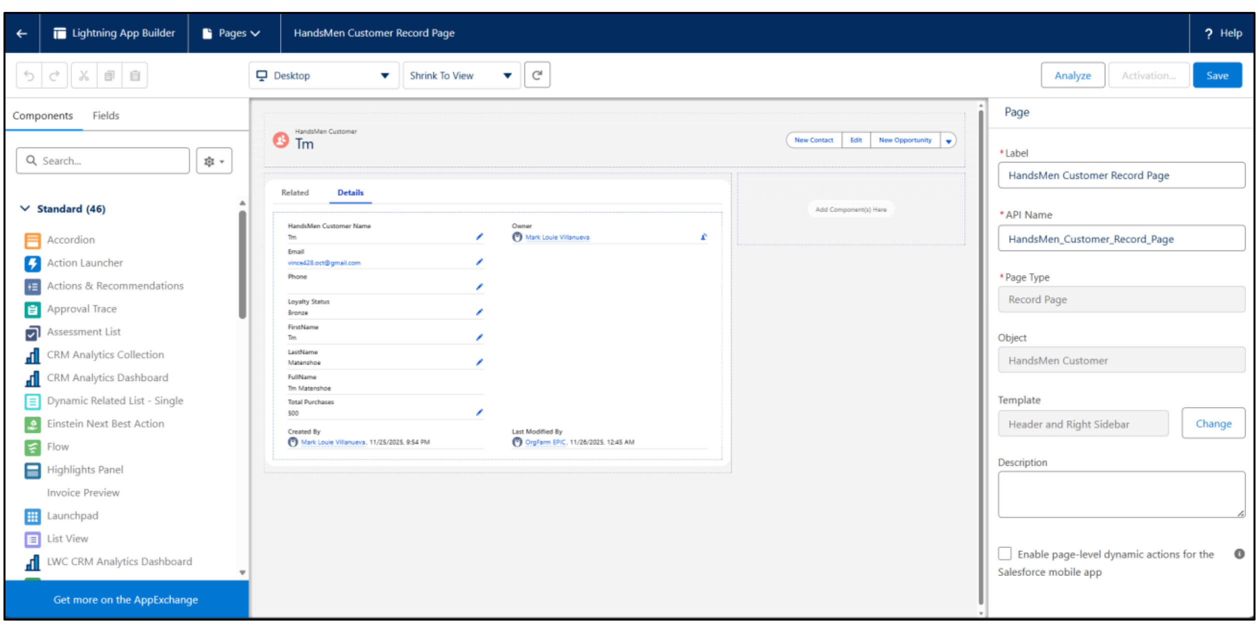


Fig.25. HandsMen Dashboard

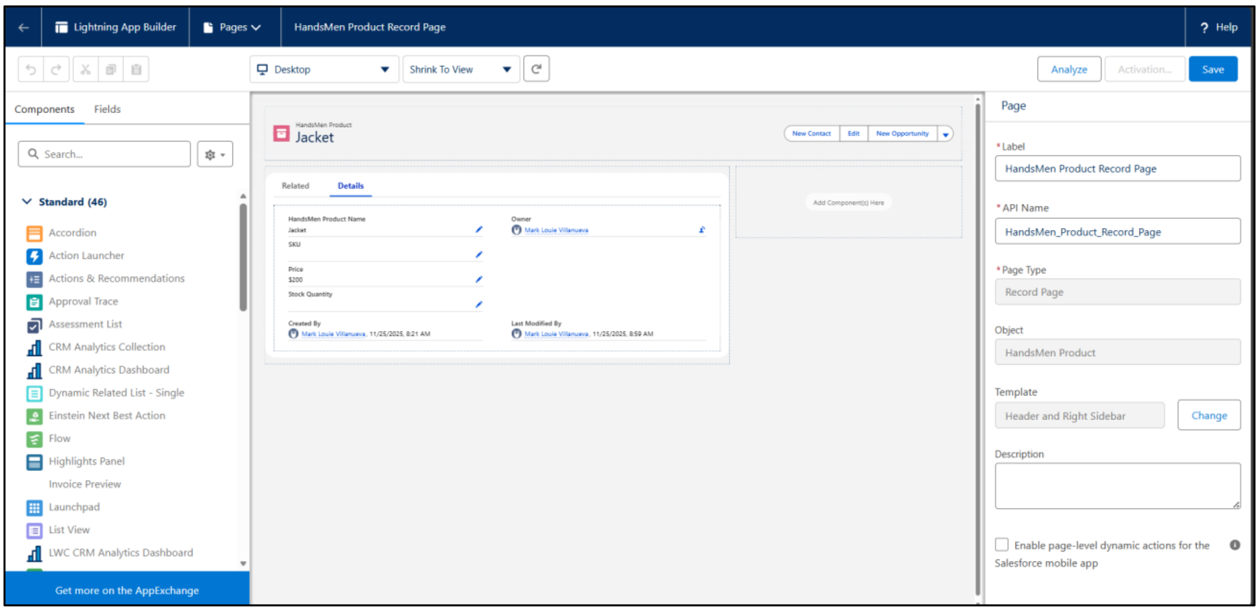
Lightning Pages

Lightning Pages are created using the Lightning App Builder's drag-and-drop tools, allowing a clean and customizable interface design that fits the needs of each user.

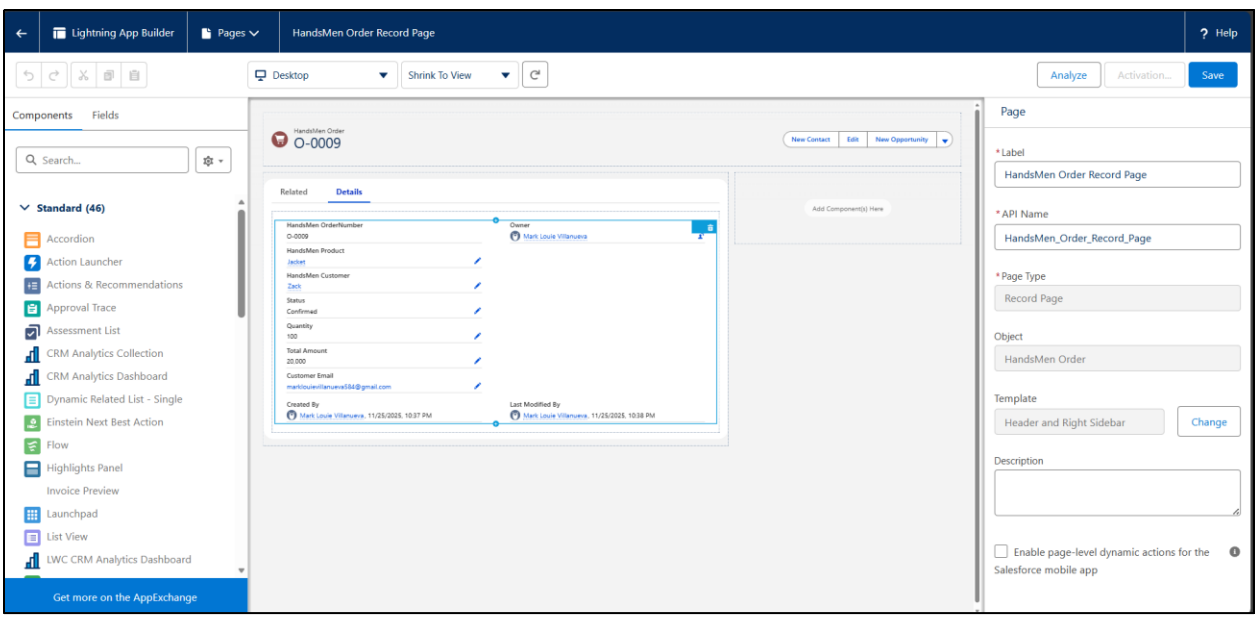
1. **HandsMen Customer Record Page**



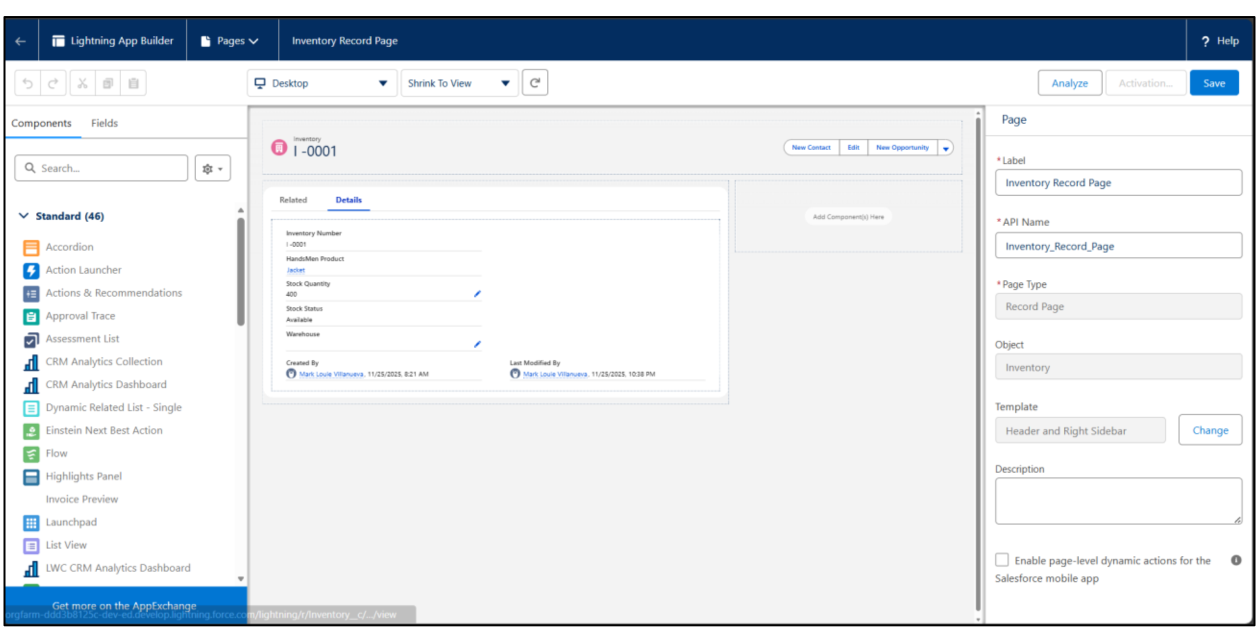
2. **HandsMen Product Record Page**



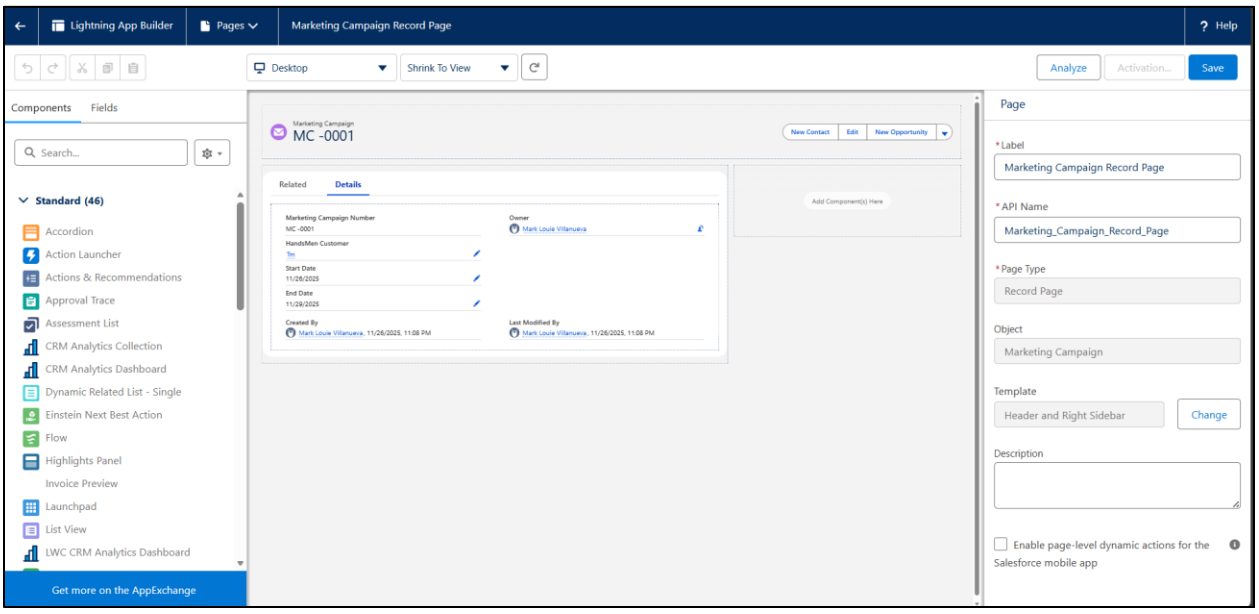
3. **HandsMen Order Record Page**



4. **Inventory Record Page**



5. **Marketing Campaign Record Page**



PHASE 4: DATA MIGRATION, TESTING AND SECURITY

Field History Tracking, Duplicate Rules, Matching Rules

Field History Tracking was enabled for key objects such as Customers, Orders, Products, and Inventory to record changes in critical fields, including total purchases, order amounts, product prices, and stock levels. This ensured transparency, accountability, and an accurate audit trail for monitoring business-critical data. Matching and duplicate rules were also configured, using unique identifiers like customer email, to prevent duplicate records and maintain data integrity across the system (See figure 26 & 27).

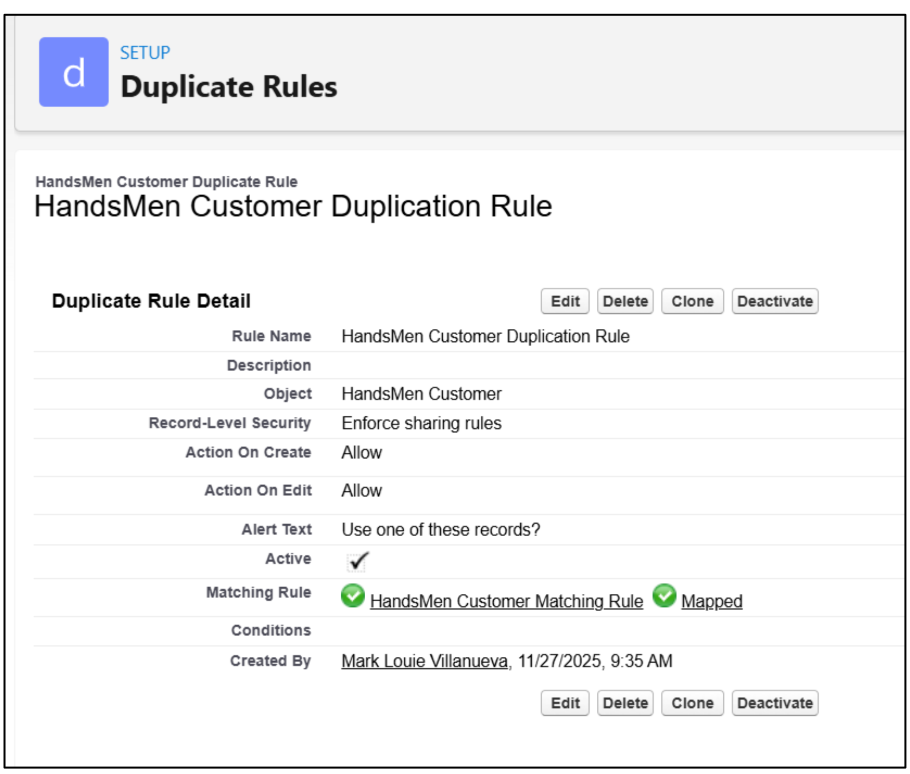


Fig.26. Duplicate Rule for Customer

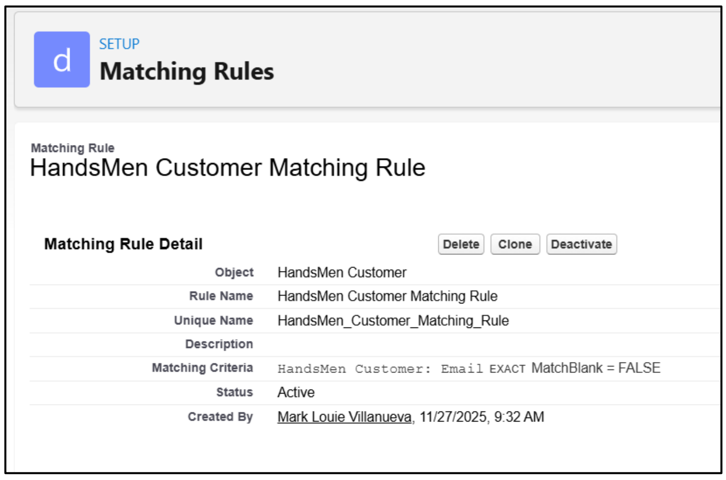


Fig.27. Matching Rule for Customer

Profile and Roles

Standard profiles such as **Standard User** and **System Administrator** were utilized to manage user access. The System Administrator profile provided full access across the Salesforce org for configuration and administrative tasks, while the Standard User profile allowed regular users to perform day-to-day operations within their permitted objects and records. This setup ensured proper access control and maintained system security while supporting operational needs.

Role Hierarchy

Figure 28 illustrates the organizational role hierarchy which CEO held full access to all system data. The Sales team reported to the CEO and focused on managing customer orders, while the Marketing team, also under the CEO, oversaw campaigns and customer information. The Inventory team reported to the CEO and was responsible for maintaining product and stock records. This hierarchical structure provided clear access boundaries, ensuring each team could perform its duties efficiently while safeguarding sensitive business data.

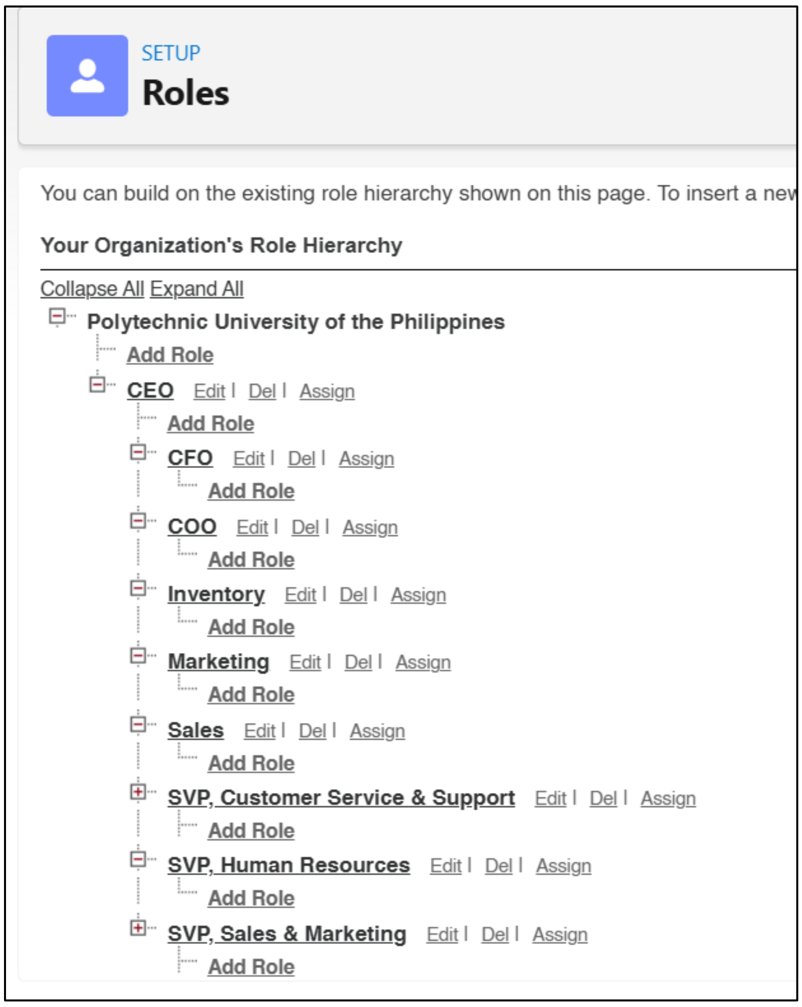


Fig.28. Organizational Hierarchy

Permission Sets

The figure 29 illustrates that the CRM employs a role- and profile-based security framework to control access to objects and data. For example, Sales Managers have full access to customer and order records, Marketing Team members can read customer information and manage campaigns, and Inventory Managers have read and edit access to product and stock data. In addition to standard Salesforce profiles and role hierarchy, Permission Sets are used to grant additional or specialized access, ensuring that users can view and modify records specifically relevant to their responsibilities without changing their base profile permissions.

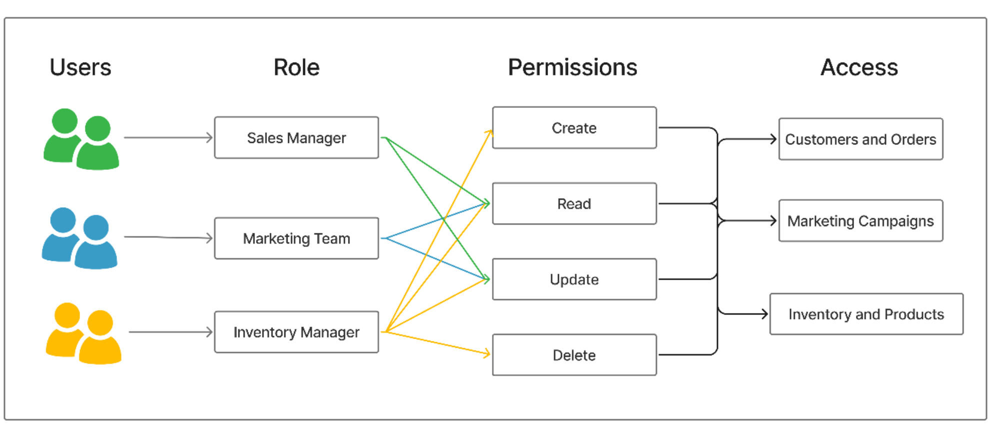


Fig.29. Permission Sets Model

PHASE 5: DEPLOYMENT, DOCUMENTATION AND MAINTENANCE

1. Development Strategy

All customizations, including objects, fields, automation, triggers, flows, and batch jobs, were implemented directly in the main Salesforce org. Features were tested individually to ensure accuracy and system reliability before being finalized for use. This approach minimized deployment complexity and ensured a stable, ready-to-use environment.

2. Maintenance and Monitoring

The system will be maintained through regular reviews of automation rules, batch jobs, validation rules, and user permissions. Logs from Apex Jobs, Debug Logs, and Flow Errors will be monitored to detect and resolve issues promptly. Updates to page layouts, roles, and permission sets will be made as business needs change, keeping the system aligned with organizational requirements.

3. Troubleshooting Approach

Issues are addressed using a structured process: identify errors via UI or logs, determine the root cause (permissions, missing fields, or automation issues), modify affected components, test changes with Apex and manual checks, and document resolutions for reference.

CONCLUSION

The Salesforce CRM for HandsMen Threads successfully streamlined business operations, improved data accuracy, and enhanced overall customer engagement. Automation through Flows, Apex triggers, and scheduled jobs reduced manual tasks and minimized errors in order processing, inventory management, and loyalty tracking. The system's role-based access, profiles, and permission sets ensured secure and appropriate data visibility, maintaining operational integrity across departments. Custom reports, dashboards, and real-time alerts empowered managers to monitor sales, inventory, and marketing performance efficiently. Additionally, the CRM's scalability allows the organization to accommodate growth, integrate future automation, and adopt predictive analytics, supporting informed business decisions and long-term strategic planning. Overall, the system not only improved daily operational efficiency but also strengthened customer satisfaction and organizational decision-making.

RECOMMENDATIONS

- Integrate Agentforce as an intelligent CRM assistant to support users with real-time guidance, automated responses, and streamlined task execution.
- Enhance system security and monitoring by enabling tools such as Salesforce Shield or Event Monitoring to protect sensitive business and customer information as the CRM scales.
- Leverage Einstein AI capabilities, such as prediction models for sales forecasting, customer segmentation, and stock demand estimation, to support strategic planning and operational efficiency.
- Implement data backup strategy like investing in a third-party backup and recovery solution (via the AppExchange) to protect against accidental data deletion or corruption (human error).