# HandsMen Threads:

Elevating the Art of Sophistication in Men's Fashion

A Salesforce Project Documentation Report of a Capstone Project for the Salesforce Developer Virtual Internship Program

**Submitted by:**

Aries John D. Tolentino

Polytechnic University of the Philippines – Santa Maria, Bulacan Campus

*Bachelor of Science in Information Technology 4-1*

**November 2025**

# Project Overview

HandsMen Threads is a growing brand in the fashion industry. They intend to launch a Salesforce project to transform how they manage data and connect with customers. This project focuses on creating a strong, well-structured data model that stores all essential business information and ensures a smooth, consistent flow of data across the company.

The goal of Handsmen Threads CRM project is to provide a functional and well-structured system that facilitates the company's primary business functions, covering marketing, sales, and inventory management Custom objects for customers, orders, products, and stock levels are included, ensuring accurate and consistent storage of all crucial business data. The system's key features include scheduled flows that update customer loyalty based on total purchases, automatic email alerts for order confirmation and low-stock situations, and loyalty tier tracking. To maintain proper data access and improve internal coordination, the project also defines user roles, profiles, and permission sets for the Sales, Marketing, and Inventory teams.

# Objectives

The primary objectives of the Handsmen Threads CRM project are to develop a centralized and reliable system for tracking customers, products, orders, and inventories, as well as to streamline each department's daily workflow. By designing the system, the goal is to:

1. Improve customer management by storing all relevant customer information in one location.

2. Streamline the sales process by organizing orders, pricing, and status updates clearly.

3. Maintain accurate product and inventory records to avoid supply shortages and overstocking.

4. Automate key processes such as order confirmations, low-stock alerts, and loyalty tier updates.

5. Strengthen data security and access control with defined roles, profiles, and permission sets.

6. Enable better reporting and decision-making through structured data and dashboards.

7. Increase overall operational efficiency by reducing manual work and system errors.

# Phase 01: Requirement Analysis & Planning

## A. Understanding Business Requirements

Based on the user scenario, HandsMen Threads required a solution that would:

- Centralize customer data for complete relationship management
- Automate order confirmations to enhance customer experience
- Implement dynamic loyalty programs based on purchase history
- Provide real-time inventory tracking with proactive alerts
- Enable bulk processing for nightly operational updates
- Maintain data integrity directly from the user interface

The scope of this project includes:

- Custom object development (Customers, Products, Orders, Inventory)
- Automated email communication system
- Loyalty program automation
- Real-time inventory management
- Role-based security model

However, it will not include the following**:**

- Third-party ERP integration
- Mobile application development
- AI and predictive analytics
- Supplier portal development

## B. Design Data Model and Security Model

**Data Model:**

- **HandsMen Customers** (enhanced Account/Contact objects)
- **HandsMen Products** with pricing and categorization
- **HandsMen Orders** with status tracking and line items
- **Inventory** with stock levels and threshold management
- **Marketing Campaign** Object for customer engagement

**Security Model:**

- Platform 1 profile as baseline access

- Role hierarchy: Sales, Marketing, Inventory

- Permission sets for departmental access control

- Field-level security for sensitive data

## C. Stakeholders Mapping

- **Sales Team:** Order management and customer relations

- **Inventory Team:** Stock monitoring and replenishment

- **Marketing Team:** Campaign management and loyalty programs

- **Management:** Reporting and business intelligence

## D. Execution RoadMap (as shown in User Story > Project Overview)

*Phase 1: Architecture & Planning*

- Define objects, fields, relationships, formula fields.
- Establish validation rules, flows, Apex triggers, batch jobs.
- Design email templates for notifications and customer communication.

*Phase 2: Development*

- Object and field creation.
- Implement automation (flows, process builders, Apex triggers).
- Set up data security and sharing rules.
- Develop batch jobs for scheduled processing.
- Configure email templates and notifications.

*Phase 3: Testing & QA*

- Unit testing of objects and automation.
- End-to-end testing with sample data.
- Performance testing and security checks.

*Phase 4: Deployment & Training*
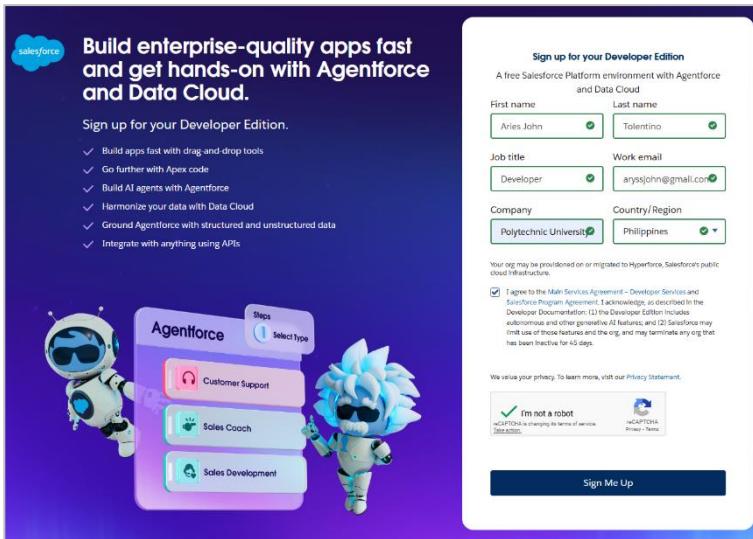
- Deploy to production.
- Train users on new functionality.
- Post-go-live support and monitoring

**Deliverable**
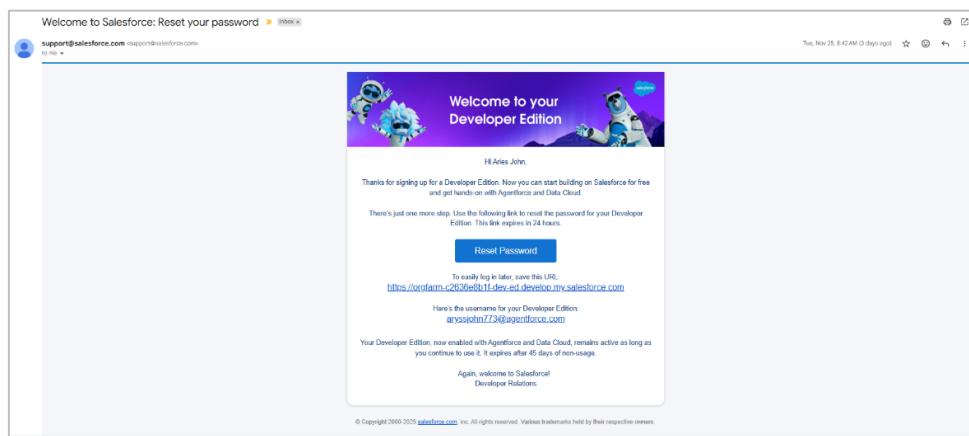
- Solution Design Document including Object Model, ERD, and Automation Strategy

# Phase 02: Salesforce Development - Backend & Configurations
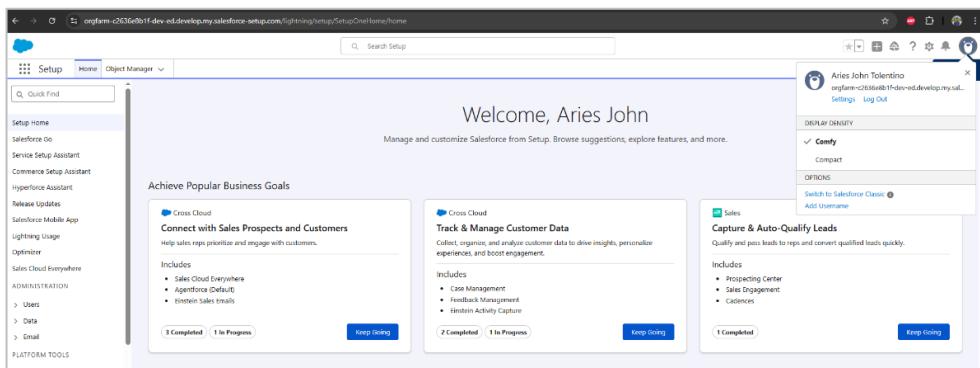
## A. Setup environment & DevOps workflow



- Signing up for a Developer Edition account



- Once the reset password email has been received, I then changed my password.



- I then logged in to my developer edition account.

# B. Customization of Objects & Fields

## HandsMen Customer

**Fields & Relationships**
11 Items, Sorted by Field Label

| FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD | INDEXED |
|---|---|---|---|---|
| Created By | CreatedById | Lookup(User) | | |
| Email | Email__c | Email | | |
| FirstName | FirstName__c | Text(60) | | |
| FullName | FullName__c | Formula (Text) | | |
| HandsMen Customer Name | Name | Text(80) | | ✓ |
| Last Modified By | LastModifiedById | Lookup(User) | | |
| LastName | LastName__c | Text(60) | | |
| Loyalty Status | Loyalty_Status__c | Picklist | | |
| Owner | OwnerId | Lookup(User,Group) | | ✓ |
| Phone | Phone__c | Phone | | |
| Total Purchases | Total_Purchases__c | Number(18, 0) | | |

- This is set to store customer information.

## HandsMen Product

**Fields & Relationships**
7 Items, Sorted by Field Label

| FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD | INDEXED |
|---|---|---|---|---|
| Created By | CreatedById | Lookup(User) | | |
| HandsMen Product Name | Name | Text(80) | | ✓ |
| Last Modified By | LastModifiedById | Lookup(User) | | |
| Owner | OwnerId | Lookup(User,Group) | | ✓ |
| Price | Price__c | Currency(18, 0) | | |
| SKU | SKU__c | Text(60) | | |
| Stock Quantity | Stock_Quantity__c | Number(18, 0) | | |

- Details of items sold by HandsMen Threads

## HandsMen Order

**Fields & Relationships**
10 Items, Sorted by Field Label

| FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD | INDEXED |
|---|---|---|---|---|
| Created By | CreatedById | Lookup(User) | | |
| Customer Email | Customer_Email__c | Email | | |
| HandsMen Customer | HandsMen_Customer__c | Lookup(HandsMen Customer) | | ✓ |
| HandsMen OrderNumber | Name | Auto Number | | ✓ |
| HandsMen Product | HandsMen_Product__c | Lookup(HandsMen Product) | | ✓ |
| Last Modified By | LastModifiedById | Lookup(User) | | |
| Owner | OwnerId | Lookup(User,Group) | | ✓ |
| Quantity | Quantity__c | Number(18, 0) | | |
| Status | Status__c | Picklist | | |
| Total Amount | Total_Amount__c | Number(18, 0) | | |

- This tracks customer purchases and status of orders.

## Inventory



- Monitors product stock levels and their respective availability.

## Marketing Campaign



- Tracks promotional activities and customer engagement

# C. Validation Rules

## HandsMen Customer Validation Rule - Email

HandsMen Customer Validation Rule
Back to HandsMen Customer

**Validation Rule Detail**   Edit   Clone

| | | | |
|---|---|---|---|
| Rule Name | Email | Active | ✓ |
| Error Condition Formula | NOT CONTAINS( Email__c , "@gmail.com") | | |
| Error Message | Please fill Correct Gmail | Error Location | Top of Page |
| Description | | | |
| Created By | Aries John Tolentino, 11/24/2025, 5:20 PM | Modified By | Aries John Tolentino, 11/24/2025, 5:20 PM |

Edit   Clone

- Customer email format validation.

  **Error Message:** "Please fill Correct Gmail"

## Inventory Validation Rule – Stock_Quantity

Inventory Validation Rule
Back to Inventory

**Validation Rule Detail**   Edit   Clone

| | | | |
|---|---|---|---|
| Rule Name | Stock_Quantity | Active | ✓ |
| Error Condition Formula | Stock_Quantity__c <= 0 | | |
| Error Message | the inventory count is never less than zero. | Error Location | Top of Page |
| Description | | | |
| Created By | Aries John Tolentino, 11/24/2025, 5:19 PM | Modified By | Aries John Tolentino, 11/24/2025, 5:19 PM |

Edit   Clone

- The inventory count can never be less than zero.

  **Error Message:** "the inventory count is never less than zero."

## HandsMen Order Validation Rule – Total_Amount

HandsMen Order Validation Rule
Back to HandsMen Order
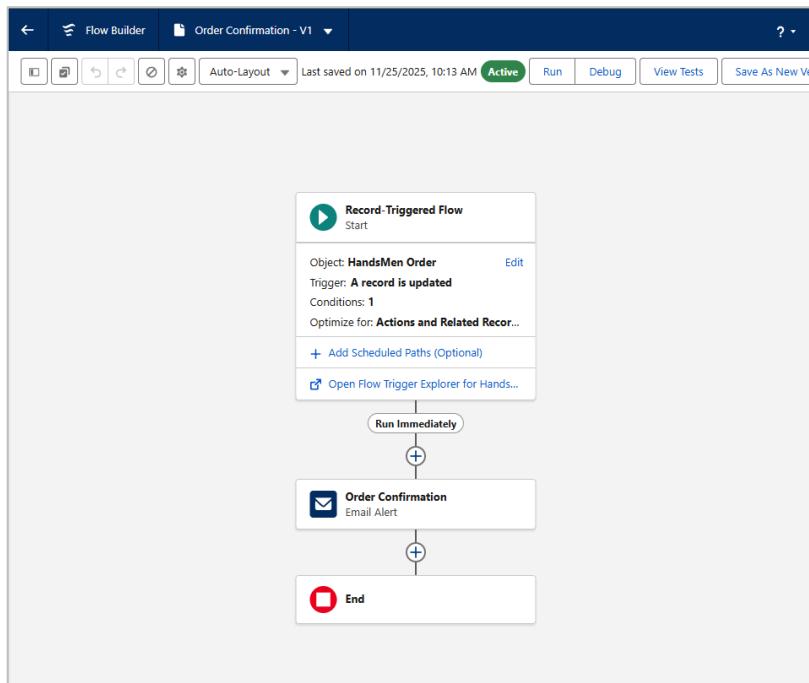
**Validation Rule Detail**   Edit   Clone

| | | | |
|---|---|---|---|
| Rule Name | Total_Amount | Active | ✓ |
| Error Condition Formula | Total_Amount__c <= 0 | | |
| Error Message | Please Enter Correct Amount | Error Location | Total Amount |
| Description | | | |
| Created By | Aries John Tolentino, 11/24/2025, 6:45 PM | Modified By | Aries John Tolentino, 11/24/2025, 6:45 PM |

Edit   Clone

- Validation for Total Amount to make sure it's correct.

  **Error Message:** "Please Enter Correct Amount"
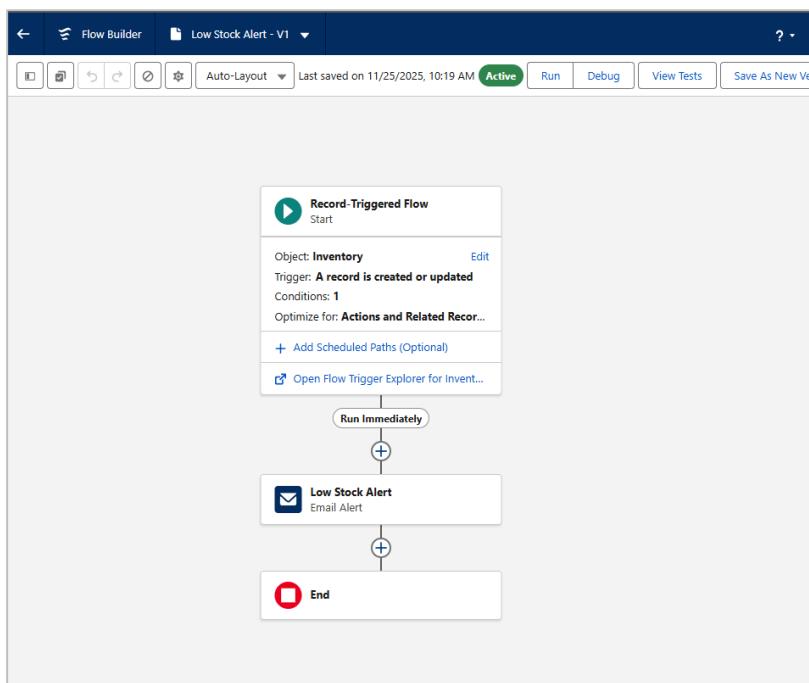
# D. Automation

## Order Confirmation Flow
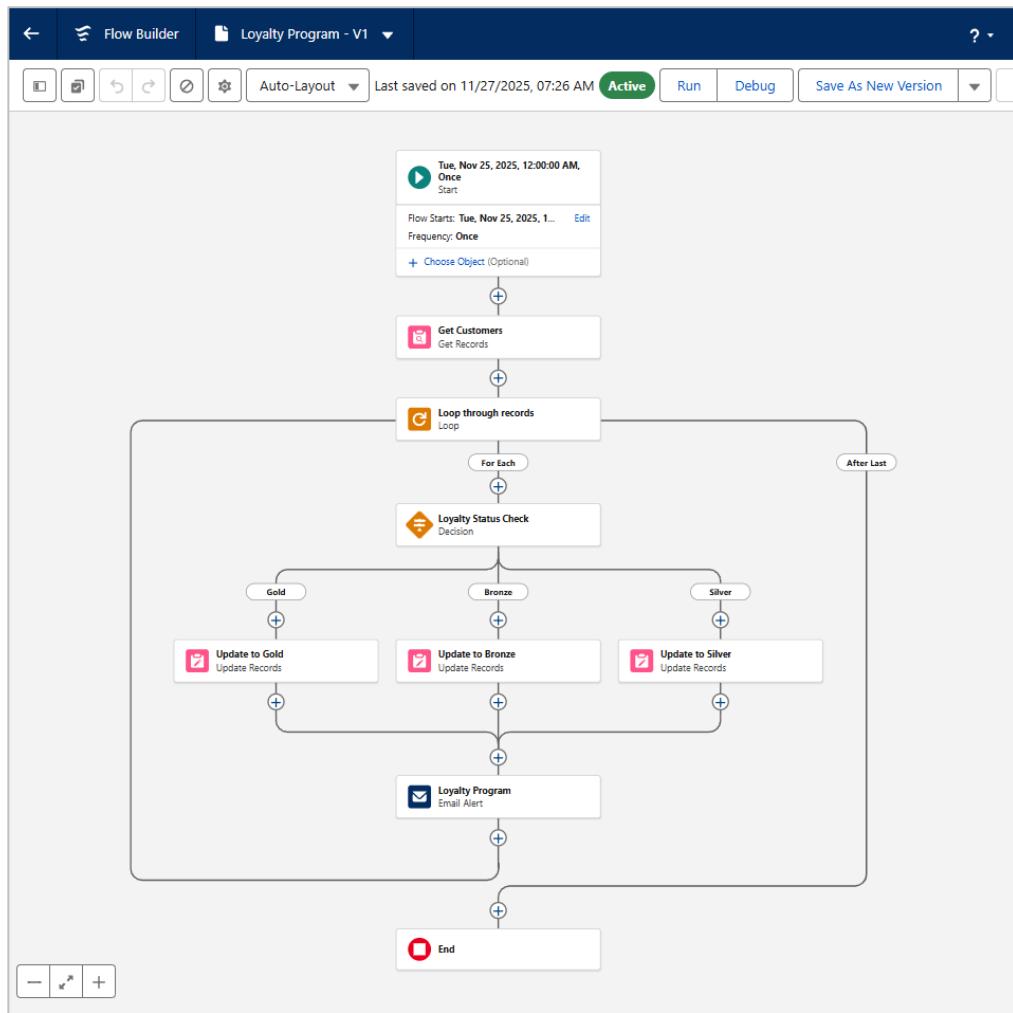


- This is the record-triggered flow for emailing customers that their order is confirmed

## Low Stock Alert Flow



- This record-triggered flow alerts the team when there is a product low in stock (less than 5).
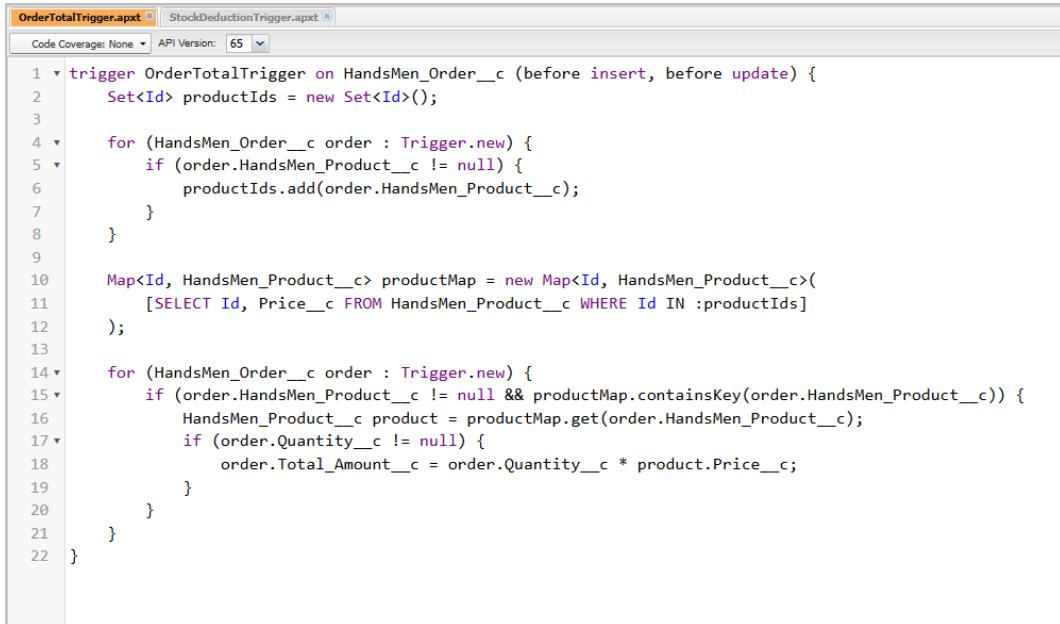
**Loyalty Program Alert Flow**



- This will trigger the 'Loyalty Program' Email Alert. Which sends a personalized congratulations email using the template I showed to your earlier.

# E. Apex Triggers

## Order Total Trigger

```
OrderTotalTrigger.apxt    StockDeductionTrigger.apxt
Code Coverage: None ▾    API Version: 65 ▾

1   ▾ trigger OrderTotalTrigger on HandsMen_Order__c (before insert, before update) {
2         Set<Id> productIds = new Set<Id>();
3
4   ▾     for (HandsMen_Order__c order : Trigger.new) {
5   ▾         if (order.HandsMen_Product__c != null) {
6                 productIds.add(order.HandsMen_Product__c);
7             }
8         }
9
10        Map<Id, HandsMen_Product__c> productMap = new Map<Id, HandsMen_Product__c>(
11            [SELECT Id, Price__c FROM HandsMen_Product__c WHERE Id IN :productIds]
12        );
13
14  ▾     for (HandsMen_Order__c order : Trigger.new) {
15  ▾         if (order.HandsMen_Product__c != null && productMap.containsKey(order.HandsMen_Product__c)) {
16                HandsMen_Product__c product = productMap.get(order.HandsMen_Product__c);
17  ▾             if (order.Quantity__c != null) {
18                    order.Total_Amount__c = order.Quantity__c * product.Price__c;
19                }
20            }
21        }
22  }
```

- Calculates the total before the order is saved

## Stock Deduction Trigger

```
OrderTotalTrigger.apxt    StockDeductionTrigger.apxt
Code Coverage: None ▾    API Version: 65 ▾

1   ▾ trigger StockDeductionTrigger on HandsMen_Order__c (after insert, after update) {
2         Set<Id> productIds = new Set<Id>();
3
4   ▾     for (HandsMen_Order__c order : Trigger.new) {
5   ▾         if (order.Status__c == 'Confirmed' && order.HandsMen_Product__c != null) {
6                 productIds.add(order.HandsMen_Product__c);
7             }
8         }
9
10        if (productIds.isEmpty()) return;
11
12        // Query related inventories based on product
13        Map<Id, Inventory__c> inventoryMap = new Map<Id, Inventory__c>(
14  ▾         [SELECT Id, Stock_Quantity__c, HandsMen_Product__c
15             FROM Inventory__c
16             WHERE HandsMen_Product__c IN :productIds]
17        );
18
19        List<Inventory__c> inventoriesToUpdate = new List<Inventory__c>();
20
21  ▾     for (HandsMen_Order__c order : Trigger.new) {
22  ▾         if (order.Status__c == 'Confirmed' && order.HandsMen_Product__c != null) {
23  ▾             for (Inventory__c inv : inventoryMap.values()) {
24  ▾                 if (inv.HandsMen_Product__c == order.HandsMen_Product__c) {
25                        inv.Stock_Quantity__c -= order.Quantity__c;
26                        inventoriesToUpdate.add(inv);
27                        break;
28                    }
29                }
30            }
31        }
32
33  ▾     if (!inventoriesToUpdate.isEmpty()) {
34            update inventoriesToUpdate;
35        }
36  }
```

- Updates the inventory after the order is confirmed.

# Phase 03: UI/UX Development & Customization

## A. Lightning App Setup





- This is the Lightning App for HandsMen Threads. It includes navigation tabs for the custom objects and other objects.
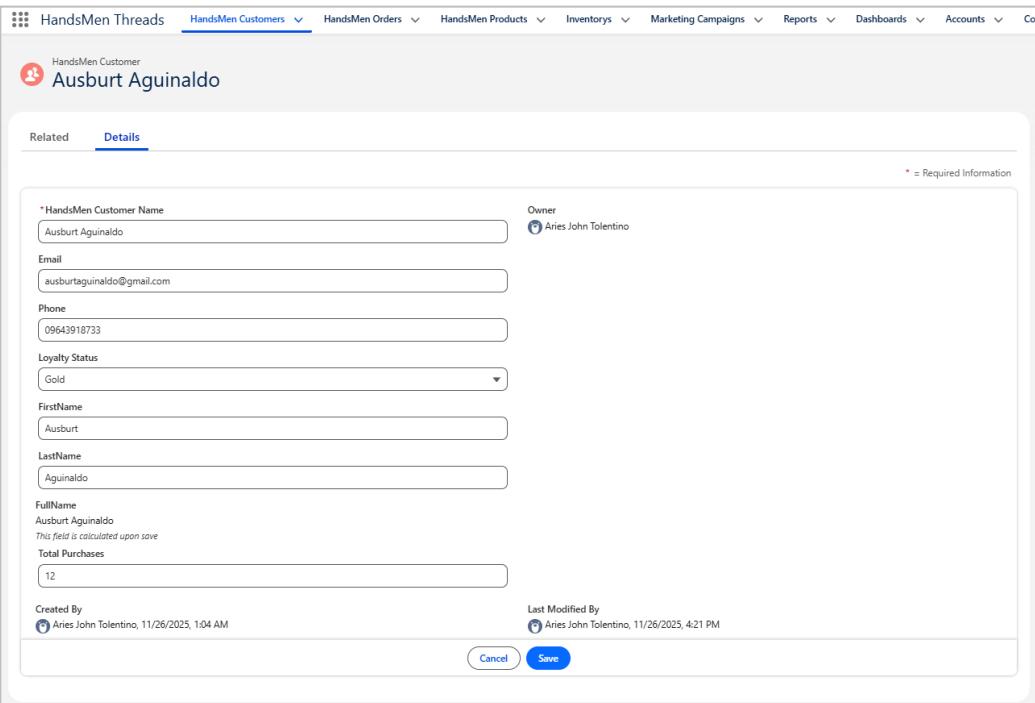
## B. Page Layouts, Dynamic Forms, User Management, Reports, and Dashboards

**Customers List** – An overview of all HandsMen Customers.



**HandsMen Customer Details** – This shows customer information.

**Orders List –** An overview of HandsMen Orders



**Order Details –** Information of each order is shown here



**Product List -** An overview of all HandsMen Thread's Products

**Product Details –** View of more details of each product



**Inventory List –** An overview of HandsMen Thread's Inventory



**Inventory Details –** Details for each inventory item to manage them.

## Marketing Campaigns



## Reports



## Dashboards

## Accounts



## Contacts



## User Management

# Phase 04: Data Migration, Testing & Security

## A. Profiles





## B. Roles and Role Hierarchy



### Inventory Role – Kol Mikaelson

**Sales Role – Niklaus Mikaelson**

Role
**Sales**

Help for this Page

Below is the list of users assigned to this role. Click Edit to modify the role name. Click Assign Users to Role to assign existing users to this role. Click New User to create a user for this role.

Hierarchy: Polytechnic University of the Philippines Santa Maria Bulacan Campus » CEO » Sales
Siblings: SVP Sales & Marketing, SVP Customer Service & Support, CFO, SVP Human Resources, COO, Inventory, Marketing

Users in Sales Role [1]

**Role Detail**                               Edit   Delete

| | | | |
|---|---|---|---|
| Label | Sales | Role Name | Sales |
| This role reports to | CEO | Role Name as displayed on reports | |
| Modified By | Aries John Tolentino, 11/24/2025, 5:27 PM | Sharing Groups | Role, Role and Internal Subordinates |
| Opportunity Access | Users in this role can edit all opportunities associated with accounts that they own, regardless of who owns the opportunities | | |
| Case Access | Users in this role can edit all cases associated with accounts that they own, regardless of who owns the cases | | |

👤 **Users in Sales Role**          Assign Users to Role    New User          Users in Sales Role Help

| Action | Full Name | Alias | Username | Active |
|---|---|---|---|---|
| Edit | Niklaus Mikaelson | nmika | aryssjohn123@gmail.com | ✓ |

**Marketing Role – Ian Mikaelson**

Role
**Marketing**

Help for this Page

Below is the list of users assigned to this role. Click Edit to modify the role name. Click Assign Users to Role to assign existing users to this role. Click New User to create a user for this role.

Hierarchy: Polytechnic University of the Philippines Santa Maria Bulacan Campus » CEO » Marketing
Siblings: SVP Sales & Marketing, SVP Customer Service & Support, CFO, SVP Human Resources, COO, Sales, Inventory

Users in Marketing Role [1]

**Role Detail**                               Edit   Delete

| | | | |
|---|---|---|---|
| Label | Marketing | Role Name | Marketing |
| This role reports to | CEO | Role Name as displayed on reports | |
| Modified By | Aries John Tolentino, 11/24/2025, 5:27 PM | Sharing Groups | Role, Role and Internal Subordinates |
| Opportunity Access | Users in this role can edit all opportunities associated with accounts that they own, regardless of who owns the opportunities | | |
| Case Access | Users in this role can edit all cases associated with accounts that they own, regardless of who owns the cases | | |

👤 **Users in Marketing Role**          Assign Users to Role    New User          Users in Marketing Role Help

| Action | Full Name | Alias | Username | Active |
|---|---|---|---|---|
| Edit | Ian Mikaelson | imika | aryssjohn098@gmail.com | ✓ |

## C. Permission sets

### Sales Permission Set - HandsMen Customers

Permission Set
**Sales Permission Set**

🔍 Find Settings...   ✖  |  Clone   Edit Properties   Manage Assignments   View Summary

Permission Set Overview  >  Object Settings  ▼  HandsMen Customers  ▼

**HandsMen Customers**                                    Edit

**Tab Settings**

| Available | Visible |
|---|---|
| ☐ | ☐ ⓘ |

**Object Permissions**

| Permission Name | Enabled |
|---|---|
| Read | ☑ |
| Create | ☑ |
| Edit | ☑ |
| Delete | ☑ |
| View All Records | ☐ |
| Modify All Records | ☐ |
| View All Fields | ☐ |

## Sales Permission Set - HandsMen Orders

**Permission Set**
## Sales Permission Set

Q Find Settings...  ⊗  |  Clone   Edit Properties   Manage Assignments   View Summary

Permission Set Overview  >  Object Settings  ▼  **HandsMen Orders** ▼

### HandsMen Orders                                                    Edit

**Tab Settings**

| Available | Visible |
|-----------|---------|
| ☐ | ☐ i |

**Object Permissions**

| Permission Name | Enabled |
|-----------------|---------|
| Read | ☑ |
| Create | ☑ |
| Edit | ☑ |
| Delete | ☑ |
| View All Records | ☐ |
| Modify All Records | ☐ |
| View All Fields | ☐ |

## Inventory Permission Set – HandsMen Products

**Permission Set**
## Inventory Permission Set

Q Find Settings...  ⊗  |  Clone   Edit Properties   Manage Assignments   View Summary

Permission Set Overview  >  Object Settings  ▼  **HandsMen Products** ▼

### HandsMen Products                                                    Edit

**Tab Settings**

| Available | Visible |
|-----------|---------|
| ☐ | ☐ i |

**Object Permissions**

| Permission Name | Enabled |
|-----------------|---------|
| Read | ☑ |
| Create | ☐ |
| Edit | ☑ |
| Delete | ☐ |
| View All Records | ☐ |
| Modify All Records | ☐ |
| View All Fields | ☐ |

## Inventory Permission Set – Inventorys

**Permission Set**
### Inventory Permission Set

| 🔍 Find Settings... ✖ | Clone | Edit Properties | Manage Assignments | View Summary |

Permission Set Overview > Object Settings ▼ **Inventorys** ▼

**Inventorys**    Edit

**Tab Settings**

| Available | Visible |
|-----------|---------|
| ☐ | ☐ ⓘ |

**Object Permissions**

| Permission Name | Enabled |
|-----------------|---------|
| Read | ☑ |
| Create | ☐ |
| Edit | ☑ |
| Delete | ☐ |
| View All Records | ☐ |
| Modify All Records | ☐ |
| View All Fields | ☐ |

## Marketing Permission Set - HandsMen Customers

**Permission Set**
### Marketing Permission Set

| 🔍 Find Settings... ✖ | Clone | Edit Properties | Manage Assignments | View Summary |

Permission Set Overview > Object Settings ▼ **HandsMen Customers** ▼

**HandsMen Customers**    Edit

**Tab Settings**

| Available | Visible |
|-----------|---------|
| ☐ | ☐ ⓘ |

**Object Permissions**

| Permission Name | Enabled |
|-----------------|---------|
| Read | ☑ |
| Create | ☐ |
| Edit | ☐ |
| Delete | ☐ |
| View All Records | ☐ |
| Modify All Records | ☐ |
| View All Fields | ☐ |

**Marketing Permission Set – Marketing Campaigns**



## D. Creation of Apex Triggers and Classes

**Order Total Trigger**

```
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;
            }
        }
    }
}
```

## Stock Deduction Trigger

```
OrderTotalTrigger.apxt ×   StockDeductionTrigger.apxt ×
Code Coverage: None ▾   API Version:  65 ▾

1 ▾ trigger StockDeductionTrigger on HandsMen_Order__c (after insert, after update) {
2        Set<Id> productIds = new Set<Id>();
3
4 ▾     for (HandsMen_Order__c order : Trigger.new) {
5 ▾         if (order.Status__c == 'Confirmed' && order.HandsMen_Product__c != null) {
6                productIds.add(order.HandsMen_Product__c);
7            }
8        }
9
10       if (productIds.isEmpty()) return;
11
12       // Query related inventories based on product
13       Map<Id, Inventory__c> inventoryMap = new Map<Id, Inventory__c>(
14 ▾        [SELECT Id, Stock_Quantity__c, HandsMen_Product__c
15           FROM Inventory__c
16           WHERE HandsMen_Product__c IN :productIds]
17       );
18
19       List<Inventory__c> inventoriesToUpdate = new List<Inventory__c>();
20
21 ▾     for (HandsMen_Order__c order : Trigger.new) {
22 ▾         if (order.Status__c == 'Confirmed' && order.HandsMen_Product__c != null) {
23 ▾             for (Inventory__c inv : inventoryMap.values()) {
24 ▾                 if (inv.HandsMen_Product__c == order.HandsMen_Product__c) {
25                        inv.Stock_Quantity__c -= order.Quantity__c;
26                        inventoriesToUpdate.add(inv);
27                        break;
28                    }
29                }
30            }
31        }
32
33 ▾     if (!inventoriesToUpdate.isEmpty()) {
34           update inventoriesToUpdate;
35        }
36 }
```

## Order Trigger Handler

```
OrderTotalTrigger.apxt ×   StockDeductionTrigger.apxt ×   OrderTriggerHandler.apxc ×
Code Coverage: None ▾   API Version:  65 ▾

1 ▾ public class OrderTriggerHandler {
2 ▾     public static void validateOrderQuantity(List<HandsMen_Order__c> orderList) {
3 ▾         for (HandsMen_Order__c order : orderList) {
4 ▾             if (order.Status__c == 'Confirmed') {
5 ▾                 if (order.Quantity__c == null || order.Quantity__c <= 500) {
6                        order.Quantity__c.addError('For Status "Confirmed", Quantity must be more than 500.');
7                    }
8 ▾             } else if (order.Status__c == 'Pending') {
9 ▾                 if (order.Quantity__c == null || order.Quantity__c <= 200) {
10                       order.Quantity__c.addError('For Status "Pending", Quantity must be more than 200.');
11                   }
12 ▾             } else if (order.Status__c == 'Rejection') {
13 ▾                 if (order.Quantity__c == null || order.Quantity__c != 0) {
14                       order.Quantity__c.addError('For Status "Rejection", Quantity must be 0.');
15                   }
16                }
17            }
18
19           System.debug('All records validated successfully.');
20        }
21 }
```

## Order Trigger

```
OrderTotalTrigger.apxt ×   StockDeductionTrigger.apxt ×   OrderTriggerHandler.apxc ×   OrderTrigger.apxt ×
Code Coverage: None ▾   API Version:  65 ▾

1 ▾ trigger OrderTrigger on HandsMen_Order__c (before insert, before update) {
2
3 ▾     if (Trigger.isBefore && (Trigger.isInsert || Trigger.isUpdate)) {
4
5            OrderTriggerHandler.validateOrderQuantity(Trigger.new);
6
7        }
8
9 }
```

# E. Testing

## Creating a customer

### Input



### Output

## Creating an order

### Input

New HandsMen Order

| | |
|---|---|
| * = Required Information | |

Information

HandsMen OrderNumber

Owner
Aries John Tolentino

HandsMen Product
Blue Jeans ✕

HandsMen Customer
Aries John Tolentino ✕

* Customer Email
aryssjohn@gmail.com

Status
Pending ▼

Quantity
2

Total Amount
120

Cancel   Save & New   **Save**

### Output

HandsMen Order
O-0006

✓ HandsMen Order "O-0006" was created.   ✕

Related   **Details**

HandsMen OrderNumber
O-0006

Owner
Aries John Tolentino

HandsMen Product
Blue Jeans ✎

HandsMen Customer
Aries John Tolentino ✎

Customer Email
aryssjohn@gmail.com ✎

Status
Pending ✎

Quantity
2 ✎

Total Amount
160 ✎

Created By
Aries John Tolentino , 11/27/2025, 5:37 PM

Last Modified By
Aries John Tolentino , 11/27/2025, 5:37 PM

## Creating a new product

**Input**



**Output**

## Creating an inventory item

**Input**

| New Inventory |
| --- |

* = Required Information

### Information
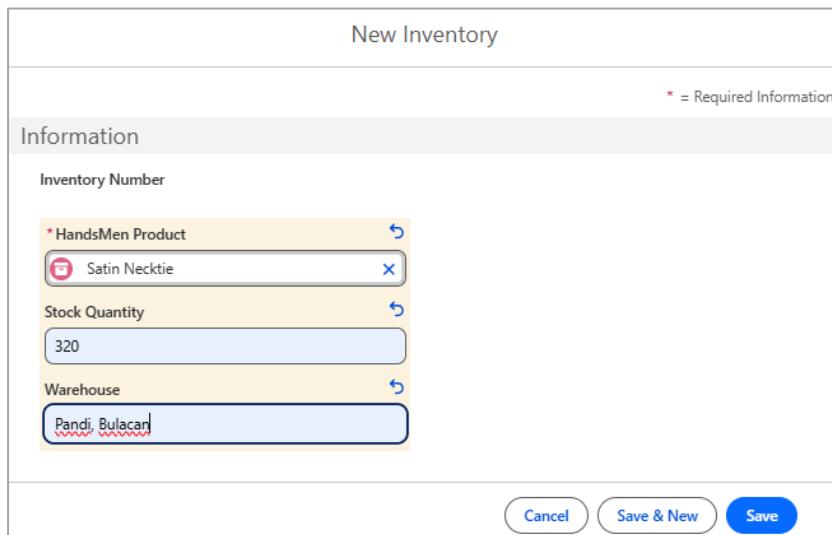
Inventory Number

* HandsMen Product ↺
Satin Necktie ✕

Stock Quantity ↺
320

Warehouse ↺
Pandi, Bulacan

Cancel   Save & New   **Save**

**Output**

Inventory
**I -0009**

✓ Inventory "I -0009" was created. ✕

Related   **Details**

Inventory Number
I -0009

HandsMen Product
Satin Necktie

Stock Quantity
320                                                     ✎
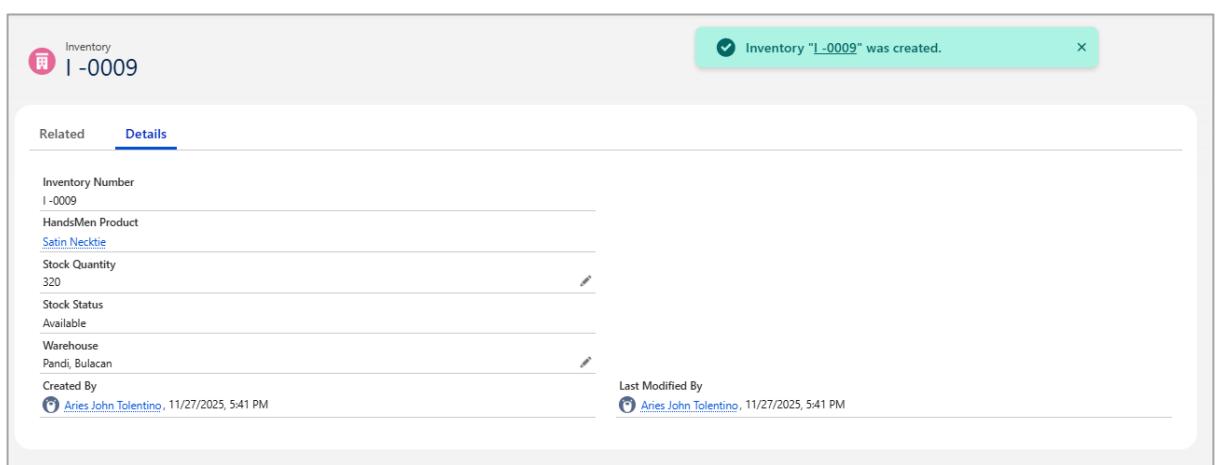
Stock Status
Available

Warehouse
Pandi, Bulacan                                          ✎

Created By                                              Last Modified By
Aries John Tolentino , 11/27/2025, 5:41 PM            Aries John Tolentino , 11/27/2025, 5:41 PM
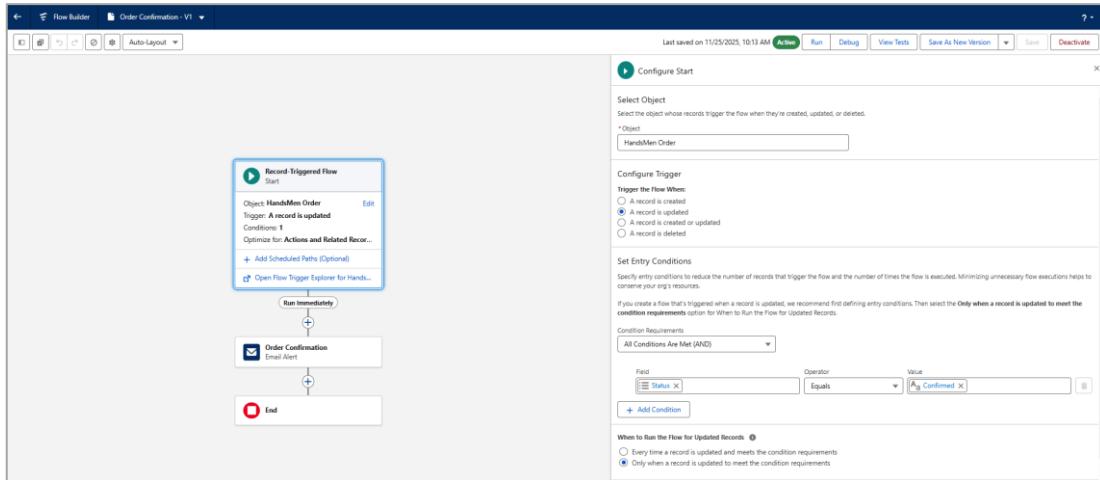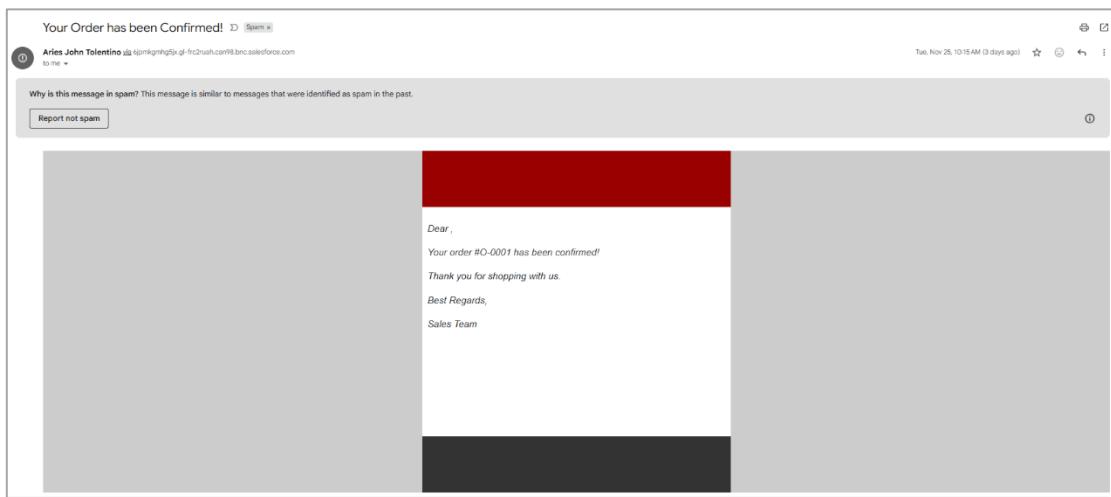
# F. Testing Record Triggered Flow (Email Alerts)

### Order Confirmation

#### Input
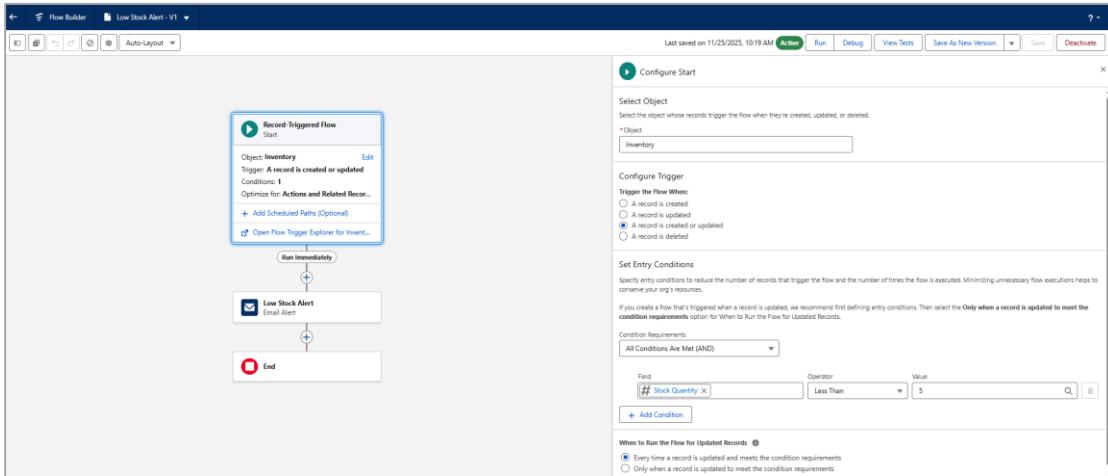


#### Output

## Low Stock Alert Email

### Input
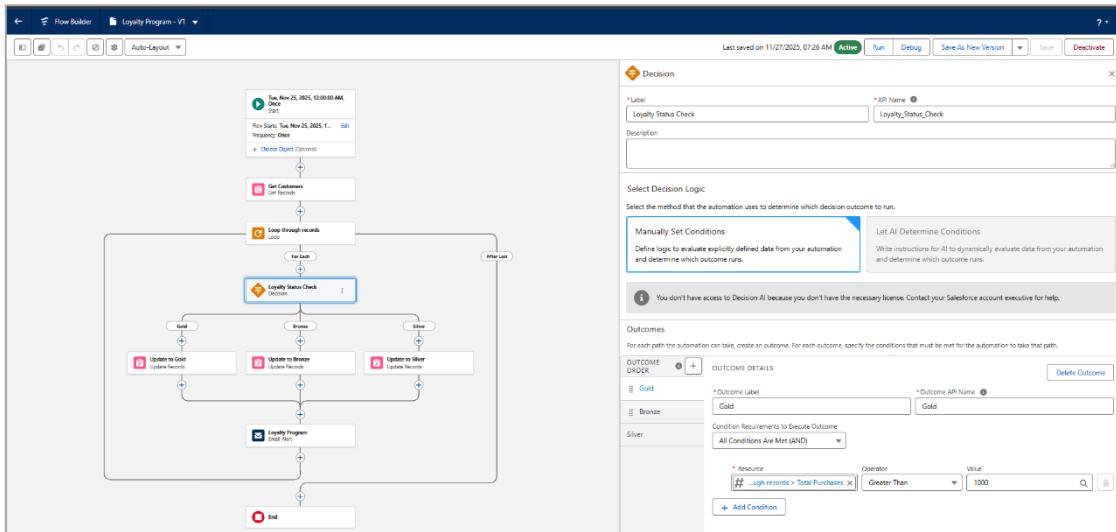


### Output

# Loyalty Program Email

## Input



## Output

**Phase 05: Deployment, Documentation & Maintenance**

**A. Deployment Strategy**

The deployment of the Handsmen Threads CRM solution follows a structured and controlled approach to ensure that all changes move safely from the development environment into production. The primary deployment method used is Salesforce Change Sets, which allow the transfer of custom objects, fields, flows, permission sets, email templates, and Apex triggers with clear traceability. This strategy ensures that every component can be validated, tested, and reviewed before being activated in the live environment.

**B. System Maintenance & Monitoring**

Once deployed, the system will be maintained through regular monitoring of key automation flows, validation rules, user permissions, and Apex trigger performance. Administrators will routinely review system logs, error emails, and queue backlogs to ensure that processes are running as expected. Maintenance also includes updating workflow logic when business requirements change, monitoring inventory thresholds, ensuring email alerts remain relevant, and reviewing data quality across all objects.

**C. Documentation**

Documentation includes only essential details needed to understand and maintain the system—not step-by-step instructions.

**It covers:**

- Brief descriptions of custom objects, fields, and their business purpose

- Overviews of validation rules, flows of key processes, and automation flows

- Explanations of Apex classes and triggers

- Screen captures of Salesforce setup pages:

  a) Object Manager configurations

  b) Permission sets and profiles

  c) Role hierarchy

  d) Flow diagrams

  e) Email template configurations

  f) Reports & dashboards settings

**Testing Approach**

- Functional testing for flows by creating sample records and checking correct updates and email alerts.

- Scenario-based testing for orders, inventory deductions, and loyalty updates.

- Access testing using different profiles and permission sets to verify correct visibility and edit rights.
- Report and dashboard testing to confirm data accuracy.

**Future Enhancements**

- AI-powered product recommendations or customer insights
- Chatbot integration for customer support
- Predictive inventory management
- Automated marketing using Salesforce Marketing Cloud or Einstein GPT
- Mobile app integration for faster on-the-go updates

**Suggestions for Future Improvements**

- AI-powered product recommendations or customer insights
- Chatbot integration for customer support
- Predictive inventory management
- Automated marketing journeys using Salesforce Marketing Cloud or Einstein
- Mobile app integration for faster on-the-go updates

## Conclusion

In summary, the Handsmen Threads CRM is a comprehensive, well-organized solution that improves all aspects of the company, including internal operations, product management, and customer engagement. This system addresses the current needs of Handsmen Threads, the CRM is built with scalability in mind, ensuring that the system can grow alongside the business as demand, products, and processes expand.

By centralizing data and simplifying workflows, the business benefits from:

- More accurate customer information and loyalty tracking
- Streamlined sales and order processing
- Better visibility into product availability and stock levels
- Reduced manual effort through automated flows and Apex triggers
- Stronger data security through defined roles, profiles, and permission sets
- Consistent communication through automated email templates
- Improved reporting for decision-making and performance monitoring

The project demonstrates how structured planning, organized data modeling, proper documentation, and systematic testing can lead to a reliable CRM solution that supports both current needs and future growth.