Salesforce Project Documentation

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

Project Overview

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

- Automated Order Confirmations

- Dynamic Loyalty Program

- Proactive Stock Alerts

- Scheduled Bulk Order Updates

This project includes use of salesforce functionalities such as:

- Data Modelling

- Data Quality

- Lightning App Builder

- Record Triggered Flows

- Apex and Apex Triggers

Business Impact

- **Customer Retention:** Expected increase by 25% through personalized shopping experiences.
- **Sales Growth:** Anticipated 30% growth in revenue within the first-year post-deployment.
- **Operational Efficiency:** Reduction in order processing time by 40%.

Objectives

- **Enhance Customer Engagement:** Deliver an AI-powered personalized journey.
- **Implement a Scalable CRM:** Configure Salesforce Sales and Service Clouds.
- **Streamline Operations:** Automate sales, inventory, and customer support processes.
- **Provide Real-time Insights:** Enable management to make data-backed decisions.
- **Ensure Robust Security:** Deploy multi-layered security measures.

Phase 1: Requirement Analysis and Planning

Activities Conducted:

- **Stakeholder Mapping:** Identified executives, sales teams, and IT admins as key stakeholders.
- **Workshops & Interviews:** Gathered insights on challenges and desired outcomes.
- **Risk Analysis:** Identified potential risks such as data migration issues and scoped mitigation strategies.
- **Feasibility Study:** Evaluated technical and financial feasibility.

Phase 2: Salesforce Development (Backend & Configurations)

Configurations Implemented:

- **Custom Objects:** Created for products, orders, customers, and feedback.
- **Automation:** Designed approval processes, validation rules, and scheduled workflows.
- **Apex Classes:** Developed for complex pricing logic and product recommendations.
- **Integrations:** Connected payment gateways, email tools, and inventory systems.

Flowchart:

Requirements → Object Modeling → Automation & Apex → Integration → Testing → Deployment

Phase 3: UI/UX Development and Customization\

- **Wireframes & Mockups:** Designed intuitive layouts for web and mobile interfaces.
- **Customer Portal:** Built with Experience Cloud for personalized access.
- **Dashboards:** Created real-time dashboards for sales performance and inventory tracking.
- **Accessibility:** Ensured compliance with WCAG standards.

Phase 4: Data Migration, Testing, and Security

- **Data Migration:** Migrated customer, order, and product data using ETL processes.
- **Testing:** Conducted unit, integration, regression, and UAT testing.
- **Security:** Implemented two-factor authentication, role-based access, and GDPR compliance.

Testing Overview:

Type	Purpose	Tools Used
Unit Testing	Validate components	Apex Test Classes
Integration	End-to-end flow validation	Sandbox
UAT	Feedback from stakeholders	Staging Environment

Phase 5: Deployment, Documentation, and Maintenance

- **Deployment:** Deployed solution using change sets and a CI/CD pipeline.
- **Documentation:** Created detailed admin/user manuals and training modules.
- **Maintenance:** Scheduled regular audits, performance monitoring, and support plans.

Additional Highlights

- **AI-Powered Personalization:** Enhanced recommendations with Salesforce Einstein.
- **Omnichannel Experience:** Unified engagement across digital and in-store touchpoints.
- **Scalability:** Future-proofed to handle new product lines and international markets.

Conclusion

HandsMen Threads successfully integrates technology and fashion, offering a customer-centric, efficient, and scalable platform. This Salesforce-powered ecosystem ensures improved sales

performance, stronger customer relationships, and a competitive edge in the men's fashion industry.

Summary

HandsMen Threads is a transformative project designed to elevate the men's fashion retail experience through Salesforce technology. By leveraging the capabilities of Salesforce Sales Cloud, Service Cloud, and Experience Cloud, this project delivers a personalized, intelligent, and seamless customer journey. It improves operational efficiency, drives data-driven decision-making, and enables future scalability. The project integrates AI-powered recommendations, secure data handling, and automated workflows to strengthen HandsMen Thread's market presence.

SREENSHOTS:

Handsmen Customer

Handsmen Order

HandsMen Order
O-0001

New Contact | Edit | New Opportunity

Related | **Details**

HandsMen OrderNumber
O-0001

Owner
Kaumudi Seri

HandsMen Product
Jacket

HandsMen Customer
John

Status
Confirmed

Quantity
10

Total Amount
40

Customer Email
john123@gmail.com

Created By
Kaumudi Seri, 7/31/2025, 6:35 AM

Last Modified By
Kaumudi Seri, 7/31/2025, 10:50 AM

Handsmen Product

HandsMen Product
Jacket

New Contact | Edit | New Opportunity

Related | **Details**

HandsMen Product Name
Jacket

Owner
Kaumudi Seri

SKU
0001

Price
$4

Stock Quantity
300

Created By
Kaumudi Seri, 7/31/2025, 6:31 AM

Last Modified By
Kaumudi Seri, 7/31/2025, 6:31 AM

Inventory

Order Confirmation Flow
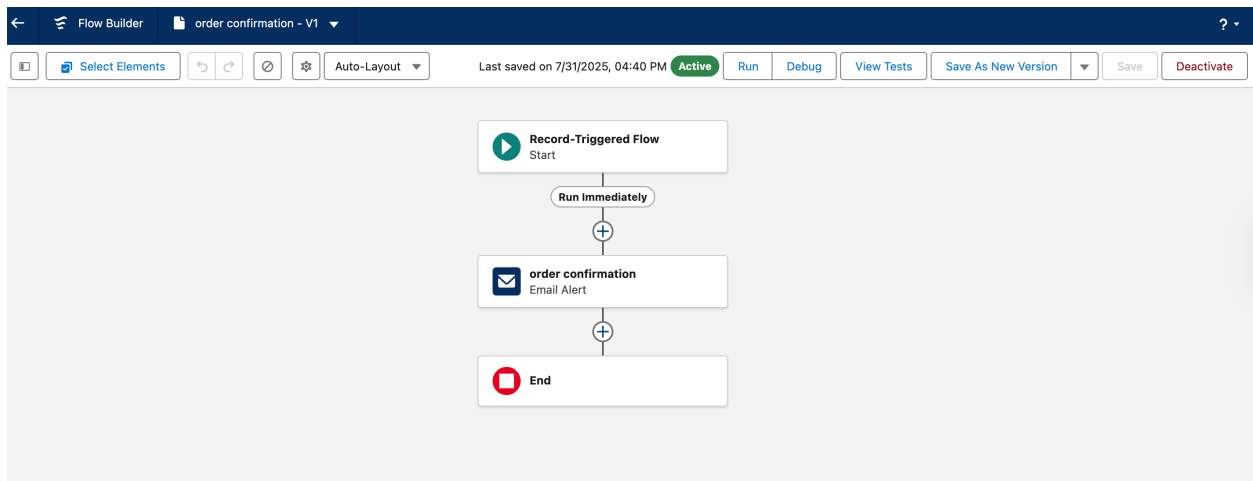
Low Stock Alert Flow

Loyalty Program Flow

APEX TRIGGER CODES

Order Total Trigger:

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;
    }
}
```
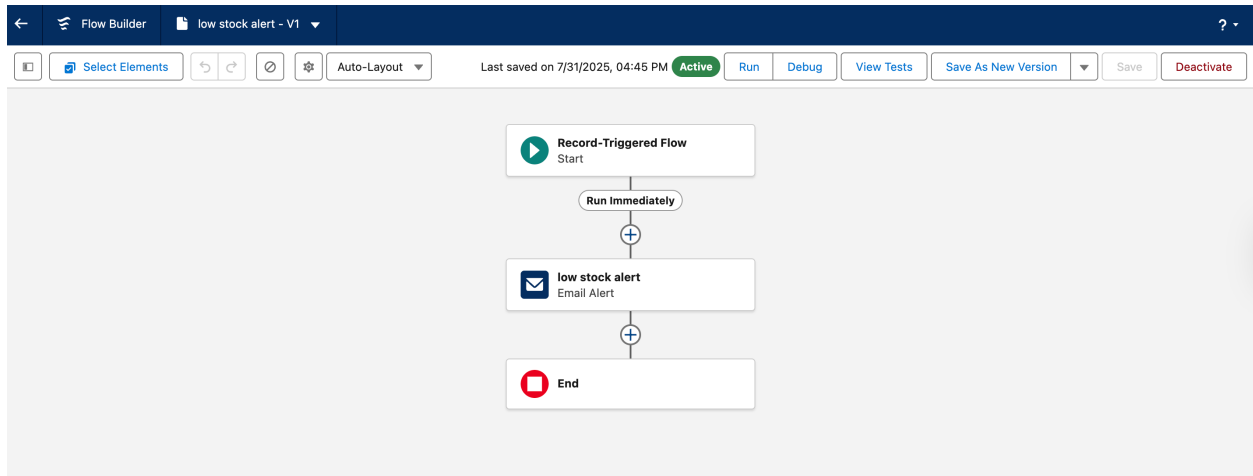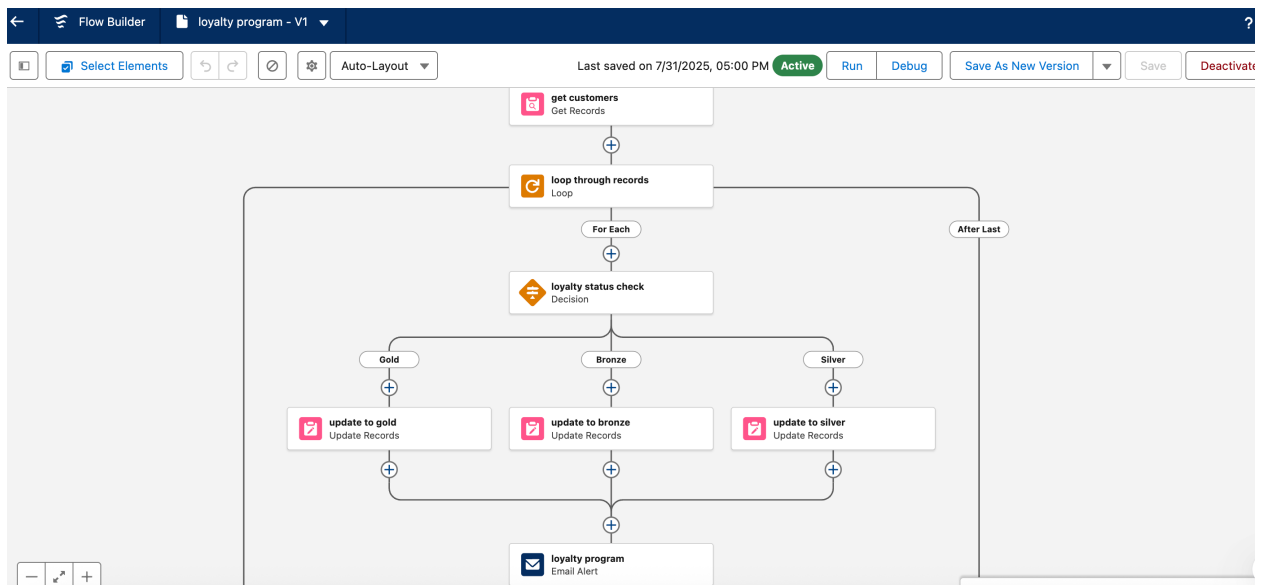
Stock Deduction Trigger:

```
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;
    }
}
```

Inventory Batch Job:

```
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);

}

}

DEBUG RESULT:

Logs	Tests	Checkpoints	Query Editor	View State	Progress	Problems			
User		Application	Operation		Time ▼	Status		Read	Size
Kaumudi Seri		Browser	/aura		31/07/2025, 17:40:48	Success		Unread	307 bytes
Kaumudi Seri		Browser	/aura		31/07/2025, 17:40:46	Success		Unread	307 bytes
Kaumudi Seri		Browser	/aura		31/07/2025, 17:40:42	Success		Unread	306 bytes
Kaumudi Seri		Browser	/aura		31/07/2025, 17:40:22	Success		Unread	307 bytes
Kaumudi Seri		Unknown	/services/data/v64.0/tooling/execut...		31/07/2025, 17:13:16	Success		Unread	2.91 KB

DONE BY:
SERI KAUMUDI