Algaba, Althea L.

WhatNext Vision Motors: Shaping the Future of Mobility with Innovation and Excellence

Project Overview:

WhatsNext Vision Motors, A leading innovator in the automotive industry, the company is committed to improving mobility with new technologies that make driving more comfortable and operations more efficient. To strengthen dealer partnerships, speed up customer orders, and automate important business tasks, the company has begun a full Salesforce CRM rollout. This system is designed to enhance customer experience by assigning dealers more intelligently, validating orders based on available stock, and automating updates for large batches of orders. Using batch processing, Apex automation, and connected workflows, the CRM helps create a faster, more accurate, and customer-focused order process.

Objectives:

The main goal of creating the WhatsNext Vision Motors CRM is to make the entire customer order process smoother and more efficient through automation and smart, data-driven features. The system boosts customer satisfaction by automatically suggesting the nearest dealer based on a customer's location, preventing orders for vehicles that are out of stock, and keeping customers updated with real-time order statuses. These improvements cut down on manual work, reduce mistakes, speed up decision-making, and ensure a seamless experience from the moment a customer inquiries to the moment their vehicle is delivered ultimately adding strong value to the business.

Phase 1: Requirement Analysis & Planning:

In the initial stage of the project, we studied the existing business processes of WhatsNext Vision Motors and turned them into clear functional requirements for Salesforce CRM. The goal was to design a system that could efficiently manage vehicle inventory, customer orders, dealer assignments, and post-sales activities while reducing the amount of manual work required.

Key Business Requirements:

The CRM system should be able to:

- Keep a unified database of all vehicles, dealers, and customers.

- Check vehicle availability before an order is created.
- Assign orders to the closest dealer automatically using the customer's location.
- Log and manage test drive bookings and service requests.
- Automate routine tasks to minimize manual effort and reduce errors.

Project Scope and Objectives

To meet the identified business needs, the CRM system was built with features that support smooth handling of vehicles, orders, dealers, and customer interactions. Its key goals were to automate important processes, keep stock information accurate, and simplify dealer assignments while creating a scalable foundation that can support future growth.

Key Components and Functionalities:

- Custom objects for Vehicles, Orders, Customers, Dealers, Test Drives, and Service Requests to organize all important data.
- Record-triggered Flows that automatically assign the closest dealer and send email alerts for test drives or service-related updates.
- Apex Triggers that check stock availability and maintain accurate inventory records during the order process.
- Batch Apex jobs that manage pending orders and update their statuses efficiently on a scheduled basis.

Data Model

To accurately reflect the business structure of the WhatsNext Vision Motors CRM, six custom objects were developed, each tailored to capture and manage specific data related to vehicles, orders, and customer interactions.

Custom Objects and Their Purpose:

Object Name	Purpose
Vehicle__c	Keeps comprehensive details on vehicles and their stock availability.
Vehicle_Dealer__c	Keeps records of dealer details and their locations.
Vehicle_Customer__c	Keeps records of customer information and contact details.
Vehicle_Order__c	Records all vehicle orders, including their status and fulfillment details.
Vehicle_Test_Drive__c	Logs and manages customer test drive appointments.
Vehicle_Service_Request__c	Keeps track of service history, maintenance requests, and related issues.

This data model captures all essential business entities in Salesforce, allowing for automation, reporting, and seamless integration across sales, service, and inventory management.

Security Model

To provide secure and controlled access to data in the WhatsNext Vision Motors CRM, a mix of standard Salesforce features and custom settings was implemented. The security model was designed to safeguard sensitive information while enabling users to work efficiently.

Key Security Configurations:

- Standard Salesforce profiles were used, with Permission Sets added to provide appropriate access to custom objects.

- **Field-Level Security and the Role Hierarchy** were applied to ensure users can only view or edit records relevant to their responsibilities.

- **Field History Tracking** was enabled for key fields, such as Stock_Quantity__c (Vehicle) and Status__c (Order), to support auditing and monitor changes over time.

This strategy helps maintain data accuracy, enforces security policies, and provides clear visibility for administrators to monitor system activity.

Phase 2: Salesforce Development – Backend & Configuration

Setup Environment & DevOps Workflow

At the beginning of the project, a dedicated Salesforce Developer Org was set up to build, configure, and test all custom features and automation. This environment was established to ensure a smooth development process and a seamless transition to production.

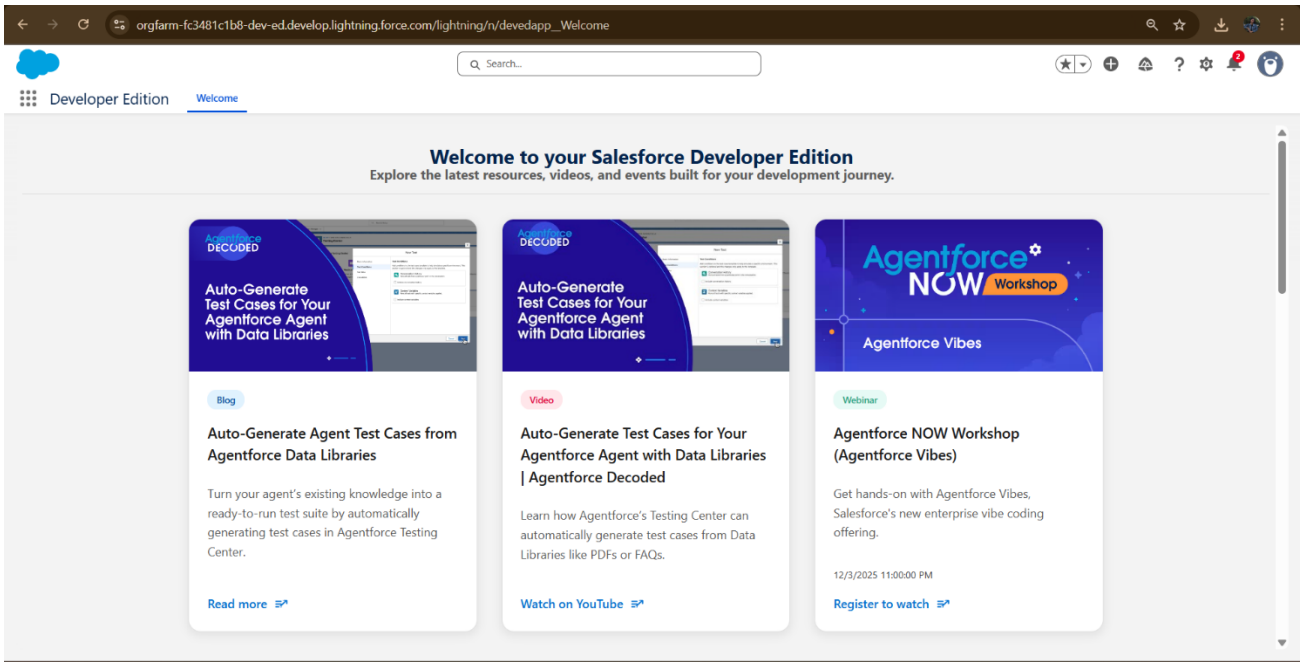
Key Setup Details:

- **Environment:** Development and testing were carried out in Salesforce Lightning Experience (Developer Edition).

- **User Profiles & Roles:** Standard Salesforce profiles were used for testing, with no custom profiles created.

- **Deployment Method:** All metadata and configurations were moved from sandbox to production using **Change Sets**, ensuring a controlled and reliable deployment process.

This configuration created a reliable environment for development, testing, and deployment, while ensuring appropriate user access and preserving system integrity.



Customization of Objects, Fields, Validation Rules, and Automation

To support WhatsNext Vision Motors' business processes, multiple custom objects and fields were created and configured in Salesforce. These objects store all relevant details about vehicles, dealers, customers, and orders, while preserving the relationships needed for automation and reporting.Custom Objects and Fields:

- **Vehicle** – Holds information about vehicles, including name, model, and stock quantity.
- **Dealer** – Records dealer details, such as location and available inventory.
- **Customer** – Stores customer information, including contact details and addresses.
- **Order** – Monitors vehicle orders, tracking their status and fulfillment.

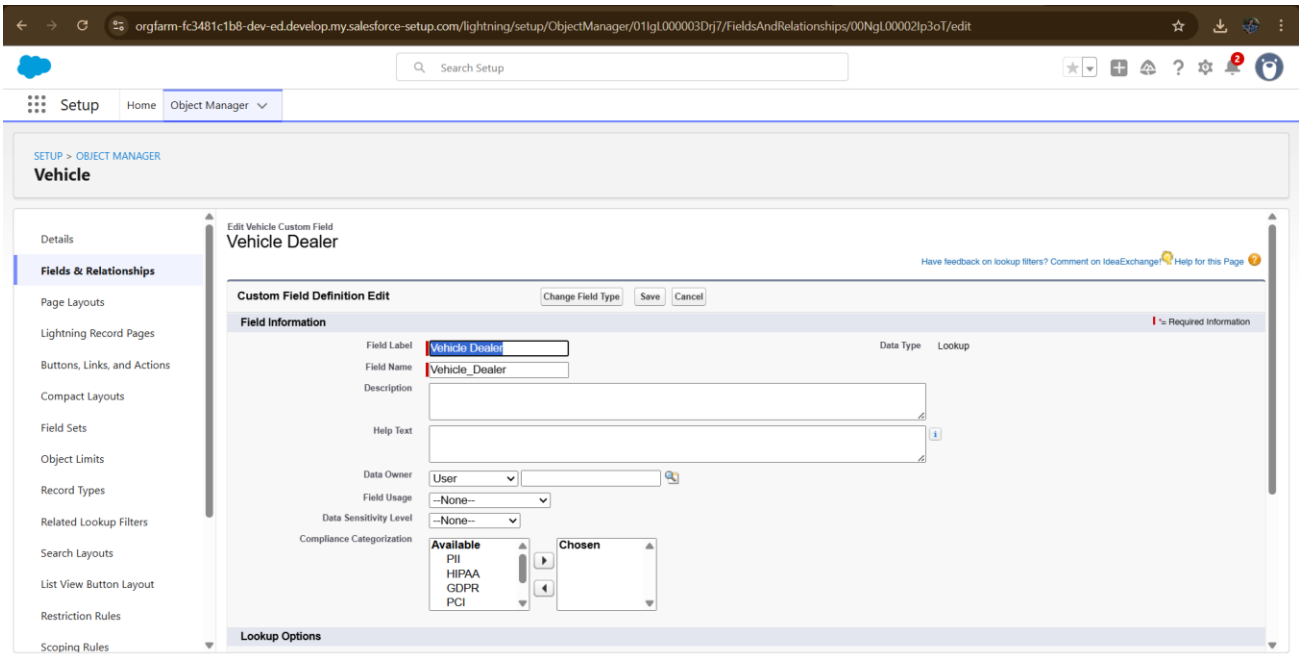
Relationships Between Objects:

- **Order → Vehicle:** A lookup relationship to connect each order with a specific vehicle.

- **Order → Dealer:** A lookup relationship linking orders to their respective dealers.

- **Order → Customer:** A lookup or master-detail relationship (depending on the setup) to associate orders with customers.

These configurations lay the groundwork for additional automation, like validation rules, flows, and Apex triggers helping maintain data integrity and enabling more efficient business processes.



Validation Rules, Automation, and Apex Classes/Triggers

To maintain data accuracy and automate key business processes, the CRM incorporates multiple validation rules, workflows, flows, and Apex components. These features help reduce errors, streamline operations, and keep records accurate.

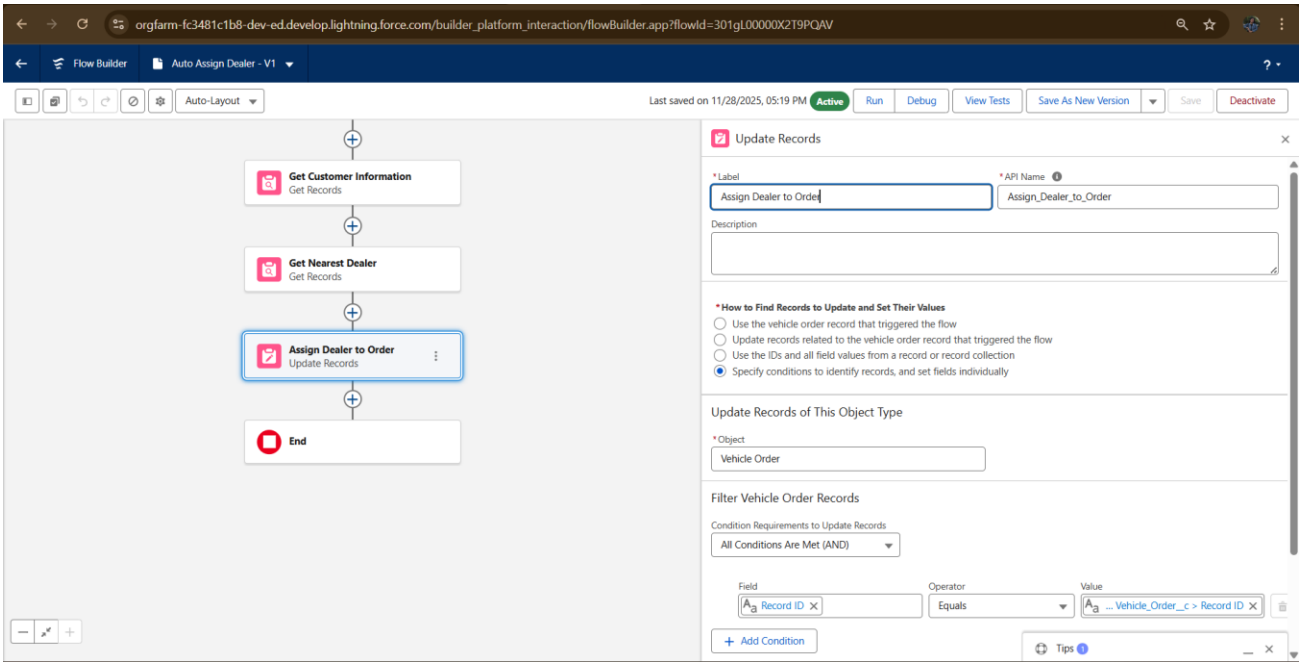
Validation Rules:

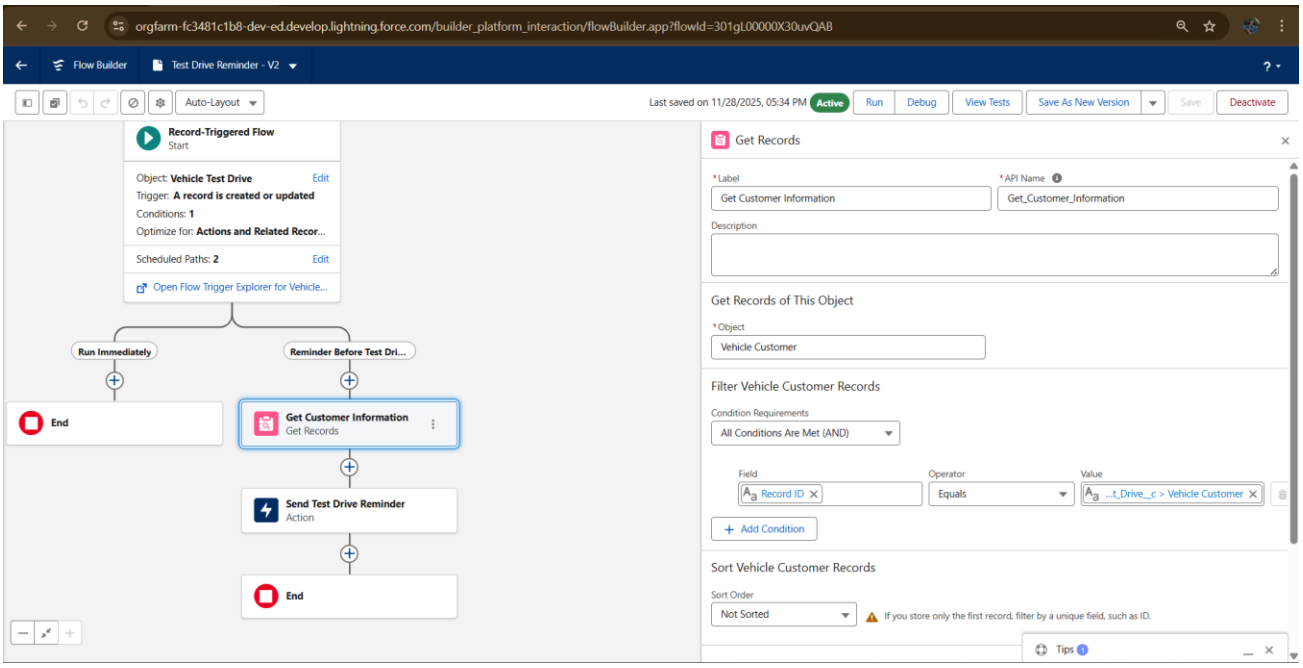
- **Out-of-Stock Order Blocker:** Stops users from placing an order if the chosen vehicle is out of stock, helping maintain accurate inventory records.

Automation: Workflow and Flows

- **Record-Triggered Flow (Order Object):** Automatically assigns the closest dealer based on the customer's address, improving order processing efficiency.

- **Scheduled Flows:** Sends automated reminders for upcoming test drives, boosting customer engagement and service.





Apex Classes and Triggers:

- **VehicleOrderTriggerHandler:** Manages trigger logic for validating stock and updating orders.

- **VehicleOrderBatch:** Regularly reviews pending orders and confirms them when stock is available.

- **VehicleOrderBatchScheduler:** Runs the batch job daily at 12 PM to ensure timely processing of pending orders.

Together, these customizations minimize manual work, enforce business rules, and enable seamless, automated management of orders and dealers within the CRM.

Apex Trigger on Order Object

An Apex Trigger was added to the Order object to enforce critical business rules and automate processes that declarative tools alone couldn't handle. The trigger ensures correct stock management, proper dealer assignment, and accurate order status updates, following Salesforce best practices through a dedicated Trigger Handler.Key Functions of the Trigger:

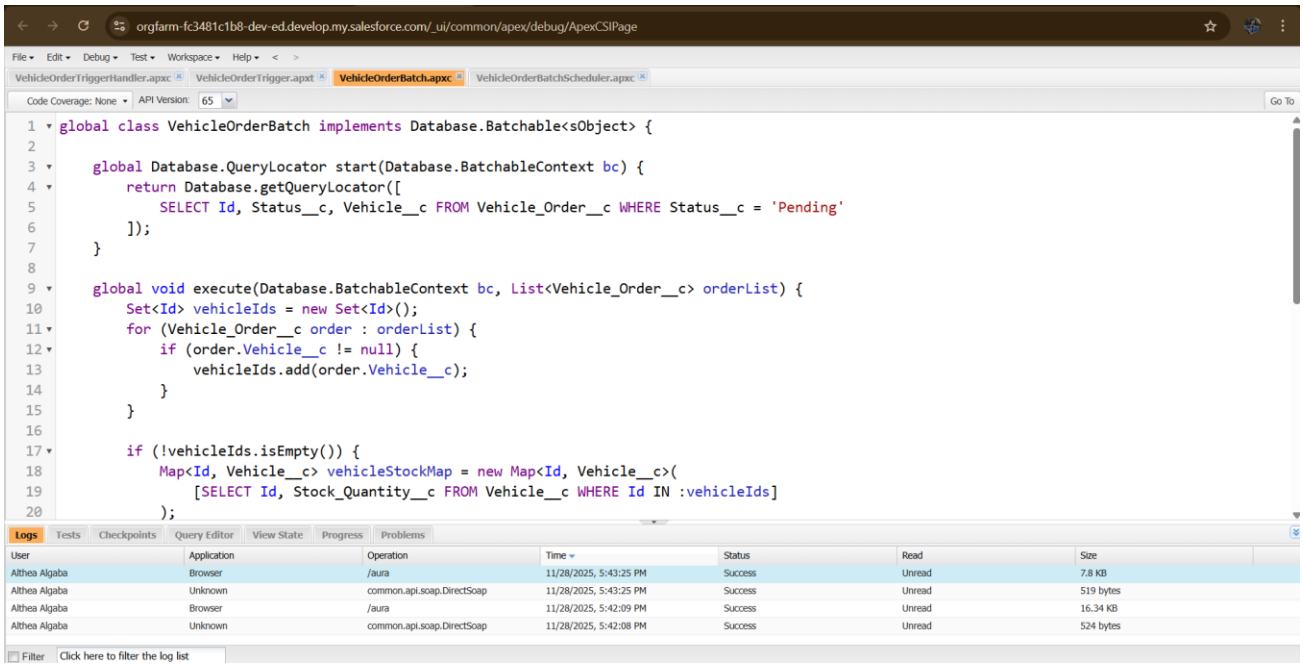
- **Stock Validation:** Verifies vehicle availability before processing an order.

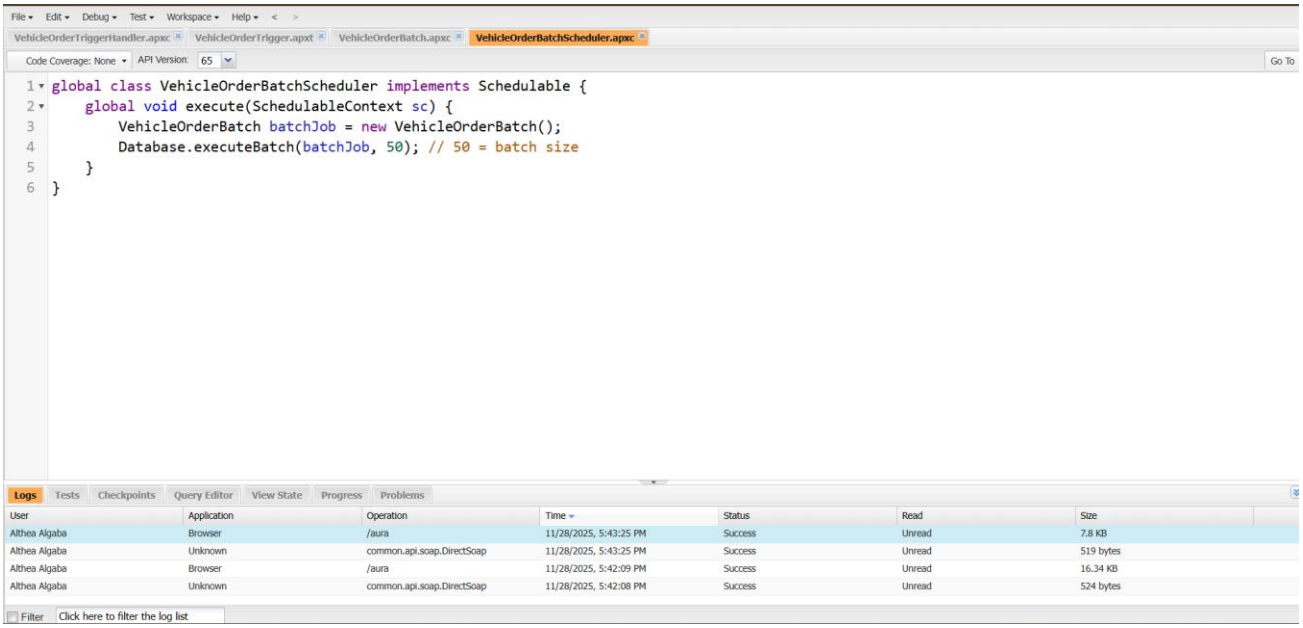
- **Dealer Assignment:** Automatically assigns the nearest dealer when not managed by a Flow.

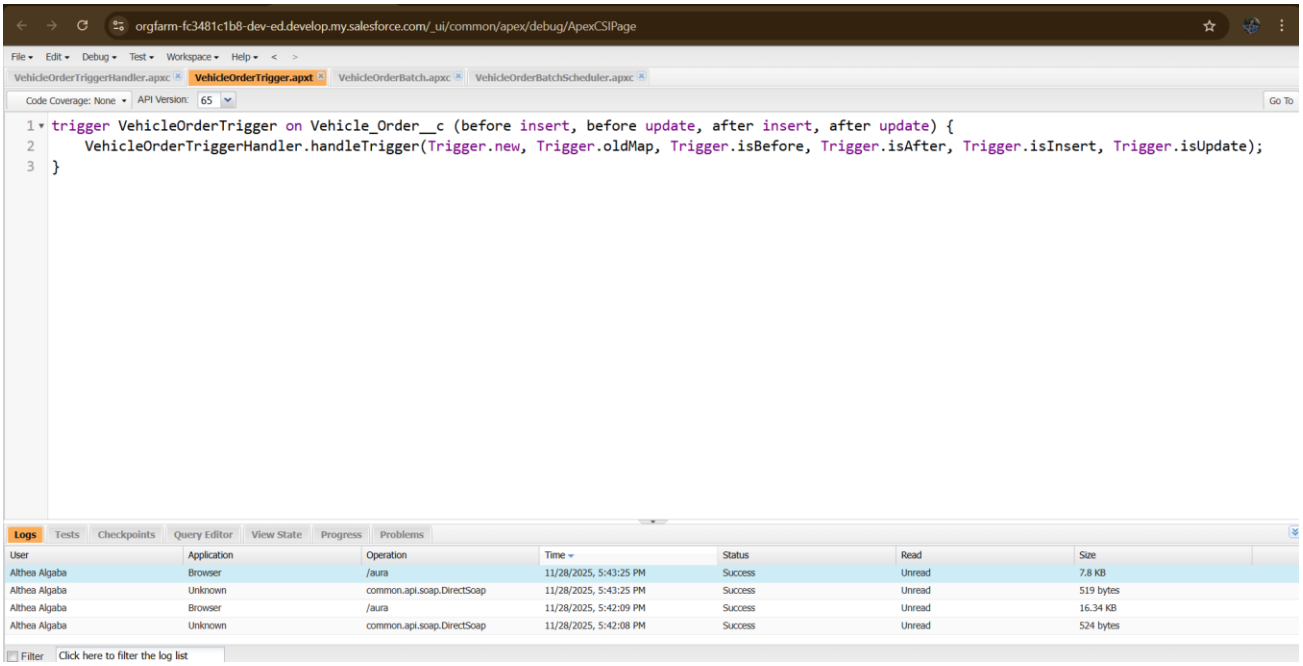
- **Order Status Updates:** Changes the order status to Pending or Confirmed depending on stock availability.

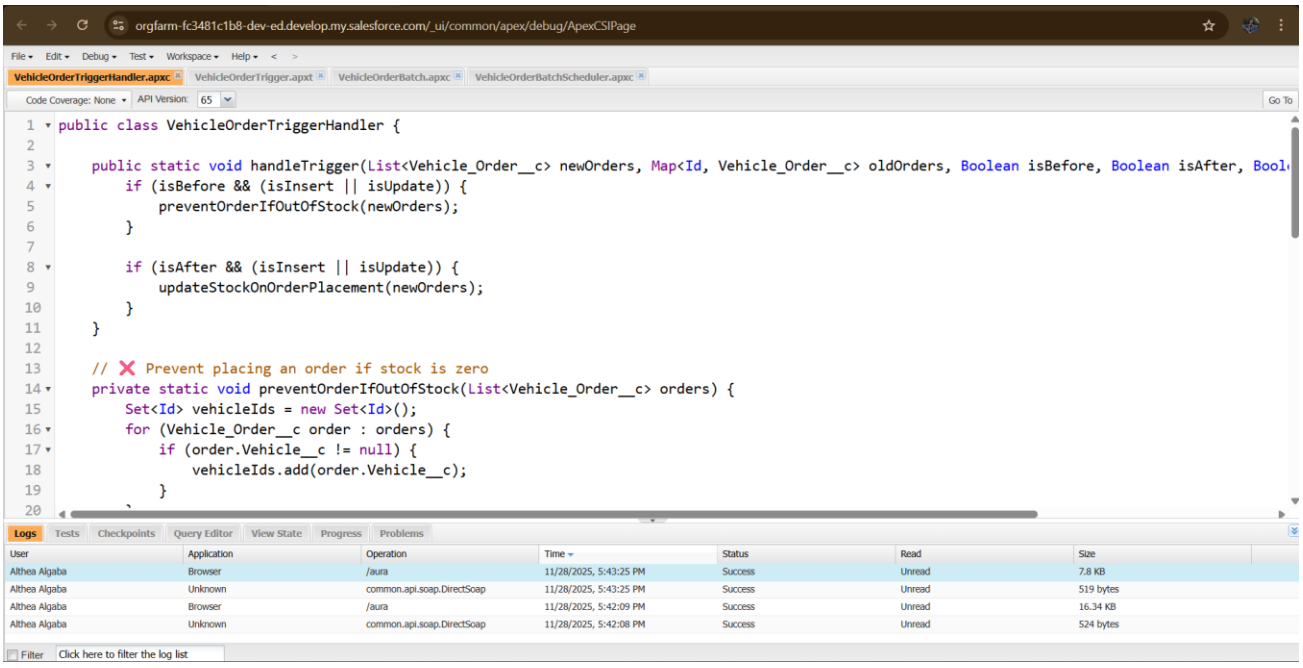
- **Trigger Handler:** Organizes the logic for better modularity, maintainability, and compliance with best practices.

This approach guarantees accurate order processing, consistent stock management, and efficient, reliable automation.


orgfarm-fc3481c1b8-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
VehicleOrderTriggerHandler.apxc
VehicleOrderTrigger.apxt
VehicleOrderBatch.apxc
VehicleOrderBatchScheduler.apxc
Code Coverage: None
API Version: 65
Go To
1 global class VehicleOrderBatch implements Database.Batchable<sObject> {
2
3 global Database.QueryLocator start(Database.BatchableContext bc) {
4 return Database.getQueryLocator([
5 SELECT Id, Status__c, Vehicle__c FROM Vehicle_Order__c WHERE Status__c = 'Pending'
6]);
7 }
8
9 global void execute(Database.BatchableContext bc, List<Vehicle_Order__c> orderList) {
10 Set<Id> vehicleIds = new Set<Id>();
11 for (Vehicle_Order__c order : orderList) {
12 if (order.Vehicle__c != null) {
13 vehicleIds.add(order.Vehicle__c);
14 }
15 }
16
17 if (!vehicleIds.isEmpty()) {
18 Map<Id, Vehicle__c> vehicleStockMap = new Map<Id, Vehicle__c>(
19 [SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds]
20);
Logs Tests Checkpoints Query Editor View State Progress Problems
User Application Operation Time Status Read Size
Althea Algaba Browser /aura 11/28/2025, 5:43:25 PM Success Unread 7.8 KB
Althea Algaba Unknown common.api.soap.DirectSoap 11/28/2025, 5:43:25 PM Success Unread 519 bytes
Althea Algaba Browser /aura 11/28/2025, 5:42:09 PM Success Unread 16.34 KB
Althea Algaba Unknown common.api.soap.DirectSoap 11/28/2025, 5:42:08 PM Success Unread 524 bytes
Filter
Click here to filter the log list


File Edit Debug Test Workspace Help
VehicleOrderTriggerHandler.apxc
VehicleOrderTrigger.apxt
VehicleOrderBatch.apxc
VehicleOrderBatchScheduler.apxc
Code Coverage: None
API Version: 65
Go To
1 global class VehicleOrderBatchScheduler implements Schedulable {
2 global void execute(SchedulableContext sc) {
3 VehicleOrderBatch batchJob = new VehicleOrderBatch();
4 Database.executeBatch(batchJob, 50); // 50 = batch size
5 }
6 }
Logs Tests Checkpoints Query Editor View State Progress Problems
User Application Operation Time Status Read Size
Althea Algaba Browser /aura 11/28/2025, 5:43:25 PM Success Unread 7.8 KB
Althea Algaba Unknown common.api.soap.DirectSoap 11/28/2025, 5:43:25 PM Success Unread 519 bytes
Althea Algaba Browser /aura 11/28/2025, 5:42:09 PM Success Unread 16.34 KB
Althea Algaba Unknown common.api.soap.DirectSoap 11/28/2025, 5:42:08 PM Success Unread 524 bytes
Filter
Click here to filter the log list


orgfarm-fc3481c1b8-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
VehicleOrderTriggerHandler.apxc
VehicleOrderTrigger.apxt
VehicleOrderBatch.apxc
VehicleOrderBatchScheduler.apxc
Code Coverage: None
API Version: 65
Go To
1 trigger VehicleOrderTrigger on Vehicle_Order__c (before insert, before update, after insert, after update) {
2 VehicleOrderTriggerHandler.handleTrigger(Trigger.new, Trigger.oldMap, Trigger.isBefore, Trigger.isAfter, Trigger.isInsert, Trigger.isUpdate);
3 }
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
User
Application
Operation
Time
Status
Read
Size
Althea Algaba
Browser
/aura
11/28/2025, 5:43:25 PM
Success
Unread
7.8 KB
Althea Algaba
Unknown
common.api.soap.DirectSoap
11/28/2025, 5:43:25 PM
Success
Unread
519 bytes
Althea Algaba
Browser
/aura
11/28/2025, 5:42:09 PM
Success
Unread
16.34 KB
Althea Algaba
Unknown
common.api.soap.DirectSoap
11/28/2025, 5:42:08 PM
Success
Unread
524 bytes
Filter
Click here to filter the log list


orgfarm-fc3481c1b8-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
VehicleOrderTriggerHandler.apxc
VehicleOrderTrigger.apxt
VehicleOrderBatch.apxc
VehicleOrderBatchScheduler.apxc
Code Coverage: None
API Version: 65
Go To
1 public class VehicleOrderTriggerHandler {
2
3 public static void handleTrigger(List<Vehicle_Order__c> newOrders, Map<Id, Vehicle_Order__c> oldOrders, Boolean isBefore, Boolean isAfter, Bool
4 if (isBefore && (isInsert || isUpdate)) {
5 preventOrderIfOutOfStock(newOrders);
6 }
7
8 if (isAfter && (isInsert || isUpdate)) {
9 updateStockOnOrderPlacement(newOrders);
10 }
11 }
12
13 // ❌ Prevent placing an order if stock is zero
14 private static void preventOrderIfOutOfStock(List<Vehicle_Order__c> orders) {
15 Set<Id> vehicleIds = new Set<Id>();
16 for (Vehicle_Order__c order : orders) {
17 if (order.Vehicle__c != null) {
18 vehicleIds.add(order.Vehicle__c);
19 }
20
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
User
Application
Operation
Time
Status
Read
Size
Althea Algaba
Browser
/aura
11/28/2025, 5:43:25 PM
Success
Unread
7.8 KB
Althea Algaba
Unknown
common.api.soap.DirectSoap
11/28/2025, 5:43:25 PM
Success
Unread
519 bytes
Althea Algaba
Browser
/aura
11/28/2025, 5:42:09 PM
Success
Unread
16.34 KB
Althea Algaba
Unknown
common.api.soap.DirectSoap
11/28/2025, 5:42:08 PM
Success
Unread
524 bytes
Filter
Click here to filter the log list

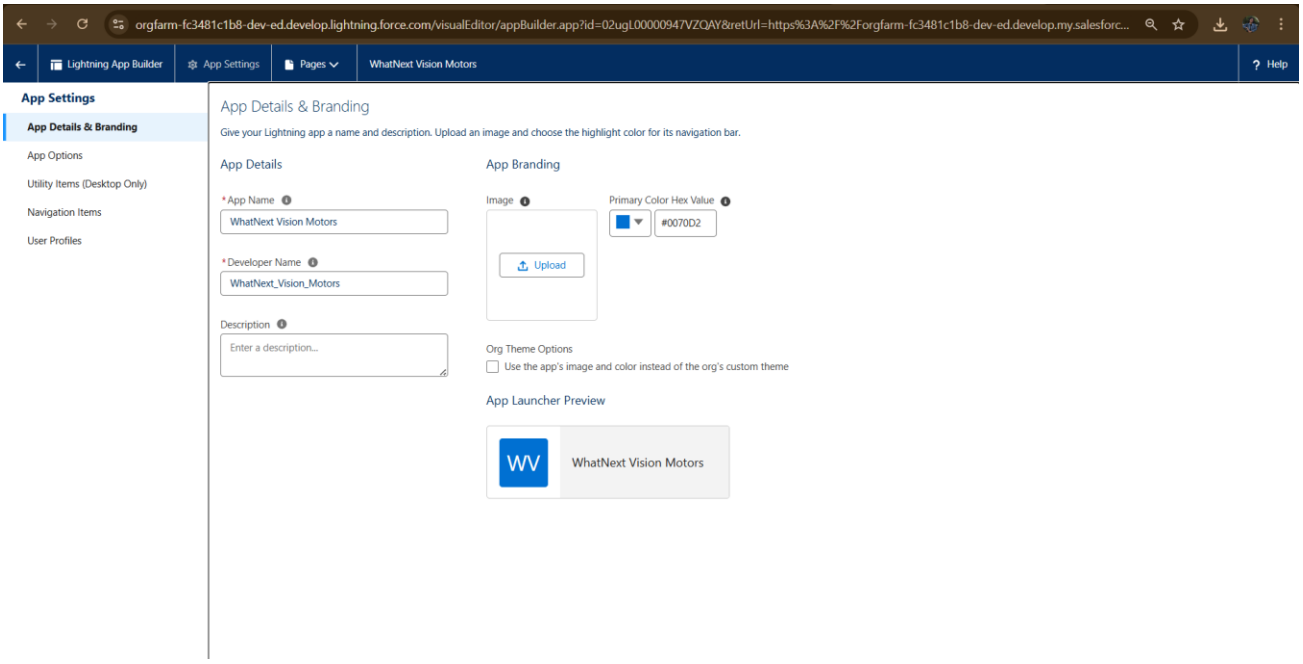
Phase 3: UI/UX Development & Customization

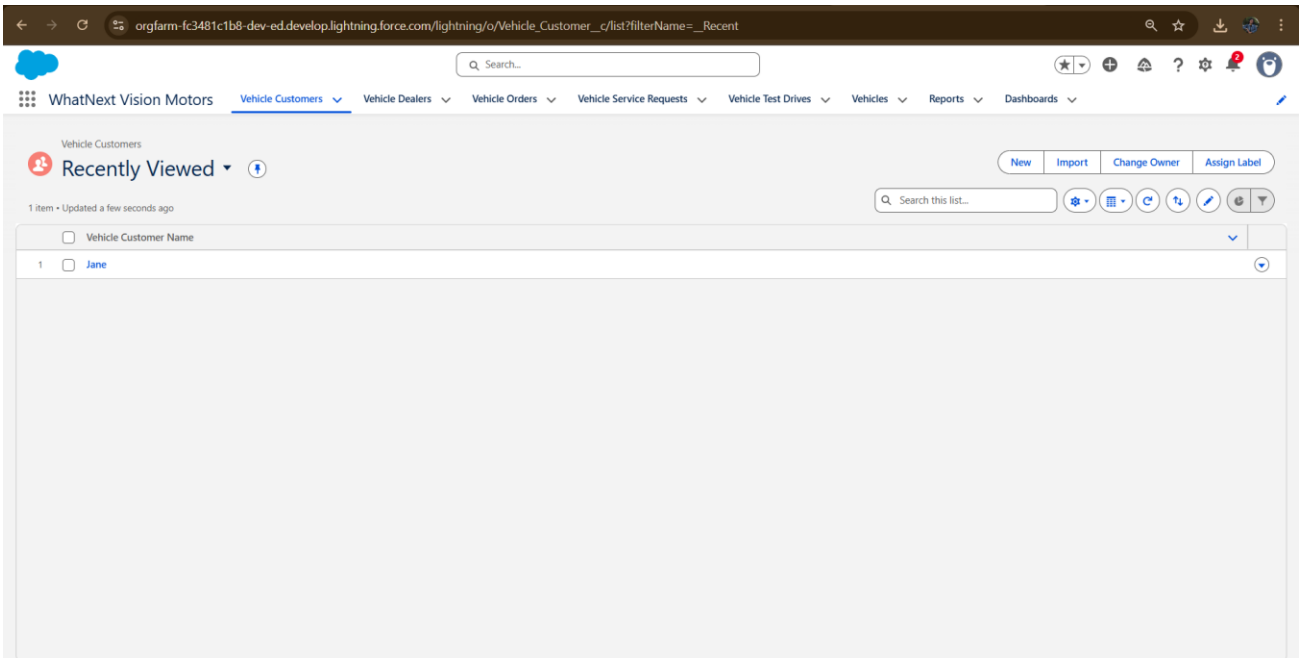
The UI/UX phase centered on designing a user-friendly interface for the WhatsNext Vision Motors CRM using Salesforce Lightning. The aim was to ensure intuitive navigation, clear display of relevant information, and an overall improved user experience.

Lightning App Setup:

- **App Name**: WhatsNext Vision Motors (set up through App Manager)
- **Tabs Included**: Vehicles, Vehicle Dealers, Vehicle Customers, Vehicle Orders, Vehicle Test Drives, and Services
- **Dynamic Forms**: Fields are shown conditionally based on record status and availability, enhancing clarity and usability
- **Lightning Pages Enhancements**: Highlight panels and related lists were added to give quick access to important information and streamline workflows

This configuration allows users to navigate the CRM efficiently, access the information they need quickly, and complete tasks with ease.


orgfarm-fc3481c1b8-dev-ed.develop.lightning.force.com/visualEditor/appBuilder.app?id=02ugL00000947VZQAY&retUrl=https%3A%2F%2Forgfarm-fc3481c1b8-dev-ed.develop.my.salesforc...
Lightning App Builder
App Settings
Pages
WhatNext Vision Motors
Help
App Settings
App Details & Branding
App Options
Utility Items (Desktop Only)
Navigation Items
User Profiles
App Details & Branding
Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.
App Details
App Branding
* App Name
WhatNext Vision Motors
* Developer Name
WhatNext_Vision_Motors
Description
Enter a description...
Image
Upload
Primary Color Hex Value
#0070D2
Org Theme Options
Use the app's image and color instead of the org's custom theme
App Launcher Preview
WV
WhatNext Vision Motors


orgfarm-fc3481c1b8-dev-ed.develop.lightning.force.com/lightning/o/Vehicle_Customer__c/list?filterName=__Recent
Search...
WhatNext Vision Motors
Vehicle Customers
Vehicle Dealers
Vehicle Orders
Vehicle Service Requests
Vehicle Test Drives
Vehicles
Reports
Dashboards
Vehicle Customers
Recently Viewed
1 item • Updated a few seconds ago
New
Import
Change Owner
Assign Label
Search this list...
Vehicle Customer Name
1
Jane

CONCLUSION:

The Salesforce CRM solution effectively addresses WhatsNext Vision Motors' core business needs, including vehicle management, dealer assignments, customer tracking, and post-sales support. By leveraging custom objects, automation flows, validation rules, Apex triggers, batch processes, and Lightning App customizations, the system provides a reliable, accurate, and user-friendly platform for managing orders and inventory.
The initiative has produced a number of significant advantages:

Improved Customer Experience: Automated dealer assignments, stock validation, and test-drive reminders ensure smooth and transparent interactions for customers.

Enhanced Operational Efficiency: Automation reduces manual work, allowing employees to focus on higher-value tasks.

Data Accuracy and Security: Audit tracking, role-based access, and field-level security protect sensitive information while maintaining data integrity.

Scalability: The system's design supports future growth and adapts to evolving business processes.