

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

ABSTRACT:

This project introduces a Salesforce CRM solution built specifically for WhatNext Vision Motors, an innovative car company focused on changing how people experience mobility and service. The system is designed to make daily operations smoother and improve customer satisfaction by tackling important needs like keeping track of stock in real time, automatically assigning dealers, simplifying order processing, and managing service requests. By using Salesforce's automation features, validation checks, and organized data handling, the CRM helps reduce mistakes and provides clear, actionable insights through easy-to-understand reports and dashboards helping the team make smarter decisions and run the business more effectively.

OBJECTIVES:

The main aim of this CRM project is to modernize and streamline the customer ordering process. Key goals include:

- **Automatically assigning dealers** based on where customers are located to speed up orders.
- **Using validation rules and Apex triggers** to avoid stock errors and ensure accuracy.
- **Automatically updating order statuses** with batch and scheduled Apex classes for up-to-date information.
- **Offering clear, actionable insights** through reports and dashboards to support smarter management decisions.

Phase 1: Requirement Analysis & Planning

Understanding Business Requirements

- Customers face delays in order processing due to manual steps.
- Dealers lack real-time stock visibility, causing missed sales chances.
- Manual errors frequently occur in stock updates and order validation.

Defining Project Scope and Objectives

- Implement data validation to prevent orders for out-of-stock items.
- Develop a centralized system to manage vehicles, orders, test drives, service requests, and dealer details.
- Automate dealer assignment for incoming orders.
- Deliver real-time reports and dashboards to track operational performance.

Design Data Model and Security Model

- **Data Model:** Custom objects cover Vehicles, Dealers, Customers, Orders, Test Drives, and Service Requests.
- **Security Model:** Role-based access is ensured with profiles, roles, permission sets, and sharing rules.

Stakeholders Mapping

- **Customers** - Main users ordering services.
- **Dealers** - Responsible for vehicle inventory and order management.
- **Sales Team** - Track leads, orders, and customer engagement.
- **Management** - Use dashboards and reports for strategic decisions.

Execution Roadmap

1. Analyze requirements and define scope.
2. Develop Salesforce backend and automate processes.
3. Design user interface and Lightning components.
4. Perform data migration, testing, and security checks.
5. Deploy solution, create documentation, and provide maintenance.

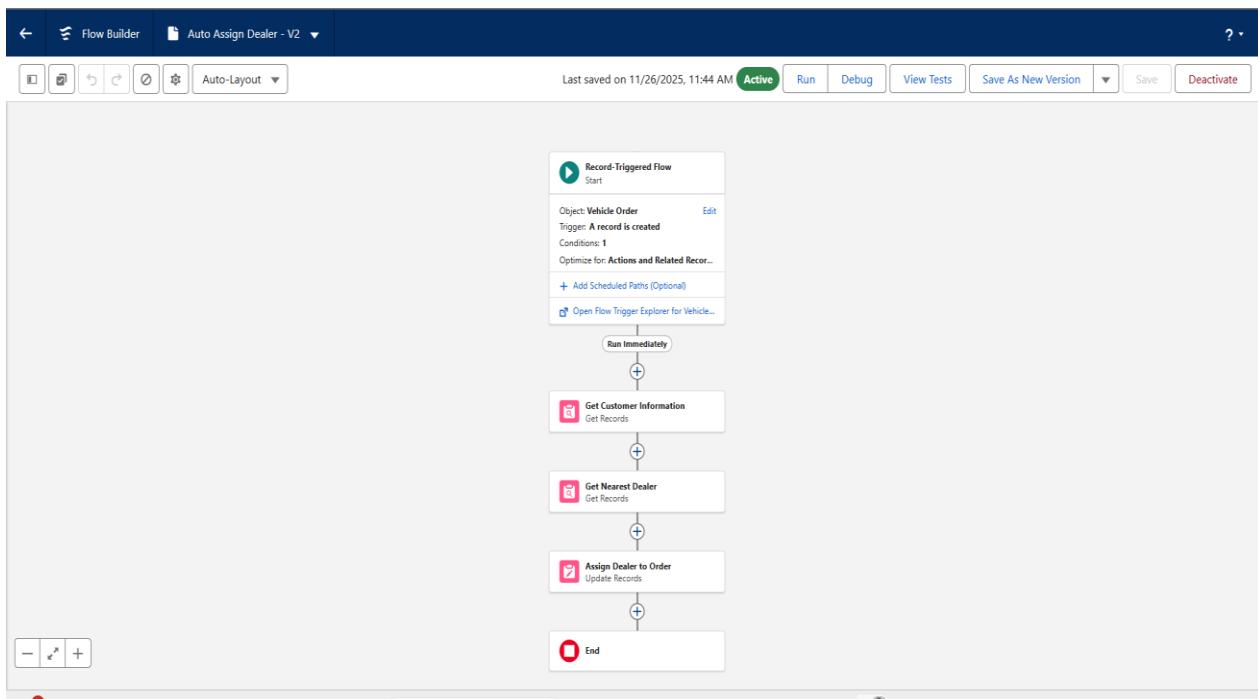
Phase 2: Salesforce Development – Backend & Configuration

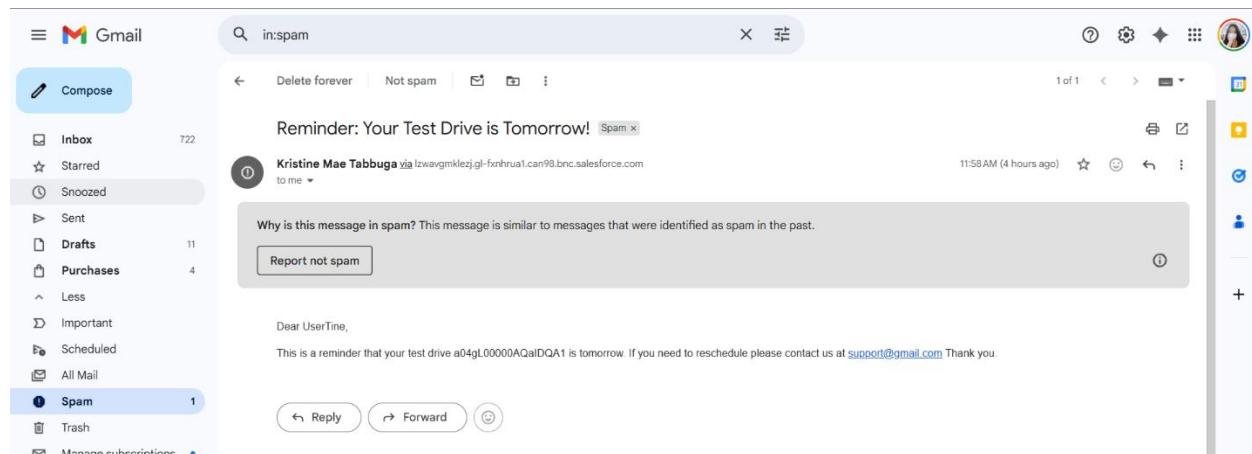
Setup Environment & DevOps Workflow

- A Salesforce sandbox is used for development, testing, and quality checks.
- Deployment to production is managed through change sets.

Customization

- Objects and fields include Vehicles, Dealers, Customers, Orders, Test Drives, and Service Requests.
- Validation rules and Apex triggers ensure customers cannot order vehicles that are out of stock.
- **Automation:**
 - **Flows:** An Auto Assign Dealer Flow automatically assigns the closest dealer to a new order.
 - **Workflow Rules & Process Builder:** Send email alerts and notifications when orders are updated.
 - **Batch and Scheduled Apex:** Automatically update order statuses based on stock changes.





LABEL	API NAME	TYPE	DESCRIPTION	LAS
Vehicle	Vehicle__c	Custom Object		11/2
Vehicle Customer	Vehicle_Customer__c	Custom Object		11/2
Vehicle Dealer	Vehicle_Dealer__c	Custom Object		11/25/2025
Vehicle Order	Vehicle_Order__c	Custom Object		11/25/2025
Vehicle Service Request	Vehicle_Service_Request__c	Custom Object		11/25/2025
Vehicle Test Drive	Vehicle_Test_Drive__c	Custom Object		11/25/2025

Phase 3: UI/UX Development & Customization

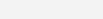
- Lightning App Configuration:** The “WhatNext Vision” application was created and set up using the App Manager.
- Page Layouts & Dynamic Forms:** Customized layouts and dynamic forms were arranged to improve user experience across all objects.
- User Access Management:** Roles, profiles, and permission sets were properly set to control user access.
- Reports & Dashboards:** Personalized dashboards were built to monitor orders, inventory levels, and dealer performance.
- Lightning Components:** Optional Lightning Web Components (LWC) were included to enhance interactivity.

- **Lightning Pages:** Designed and organized to present object information clearly and efficiently.



Salesforce Chatter Home Chatter People Groups Files Vehicles **Vehicle Customers** Vehicle Dealers Vehicle Orders

Vehicle Customers

Recently Viewed  

1 item • Updated a few seconds ago

<input type="checkbox"/> Vehicle Customer Name
<input type="checkbox"/> Tine

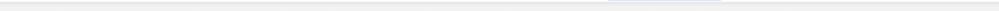
 Kristine Mae Tabbuga
orgfarm-6a9b75c4da-dev-ed.develop.my.salesforce.com
[Settings](#) [Log Out](#)

 [jayjay1@email.com](#)
dd100000f12duas-dev-ed.develop.my.salesforce.com

 [kris@email.com](#)
polytechnicuniversity-of-7eb-dev-ed.develop.my.salesforce.co

 [bebe@email.com](#)
polytechnicuniversity-of-7d3-dev-ed.develop.my.salesforce.co

[14 More Usernames](#)



Salesforce Chatter

Vehicle Dealers

Recently Viewed ▾ 

2 items • Updated a few seconds ago

	Vehicle Dealer Name	
1	<input type="checkbox"/> Mae	
2	<input type="checkbox"/> Ben	

Search this list...      

Salesforce Chatter

Vehicle Orders

Recently Viewed

2 items • Updated a few seconds ago

	Vehicle Order Number
1	O-0002
2	O-0001

New Import Change Owner Assign Label

Search this list...

Salesforce Chatter

Vehicle Test Drives

Recently Viewed

1 item • Updated a few seconds ago

	Vehicle Test Drive Name
1	Len

New Import Change Owner Assign Label

Search this list...

Salesforce Chatter

Vehicles

Recently Viewed

1 item • Updated a few seconds ago

	Vehicle Name
1	Honda

New Import Change Owner Assign Label

Search this list...

Phase 4: Data Migration, Testing & Security

- **Data Loading:** Used the Data Import Wizard and Data Loader to upload and transfer Vehicle, Dealer, and Customer records.
- **Field History Tracking, Duplicate Rules & Matching Rules:** Set up to ensure clean, accurate, and reliable data.
- **Profiles, Roles & Role Hierarchy:** Configured to provide proper access control and data visibility.
- **Test Classes:** Developed Apex test classes to validate triggers and batch processes.
- **Testing Approach:**
 - Reviewed and validated the flow for automatically assigning dealers.
 - Checked approval workflows, automated tasks, and validation rule functionality.

The screenshot shows a Salesforce Lightning interface for creating a new vehicle order. The 'Information' section includes fields for Vehicle Order Number (set to 'O-0002'), Vehicle Customer (set to 'Tine'), Vehicle (set to 'Honda'), Order Date ('11/26/2025'), Status ('Pending'), and Assigned Dealer (a dropdown menu). A red error message box appears, stating 'We hit a snag.' with the sub-instruction 'Review the errors on this page.' and a note: 'This vehicle is out of stock. Order cannot be placed.' Below the message are 'Cancel', 'Save & New', and 'Save' buttons.

The screenshot shows the Flow Builder interface for a 'Record-Triggered Flow'. The flow starts with a 'Record Triggered Flow Start' step (Object: Vehicle Order, Trigger: A record is created). It then branches into three parallel steps: 'Get Customer Information' (Get Records), 'Get Nearest Dealer' (Get Records), and 'Assign Dealer to Order' (Update Records). Finally, it concludes with an 'End' step. The flow is currently active and was last saved on 11/26/2025 at 11:44 AM. Buttons for Run, Debug, View Tests, Save As New Version, Save, and Deactivate are visible at the top.

Developer Console - Google Chrome
orgfarm-6a9b75c4da-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     public static void handleTrigger(List<Vehicle_Order__c> newOrders, Map<Id, Vehicle_Order__c> oldOrders, Boolean isBefore, Boolean isAfter, Boolean isInsert, Boolean isUpdate) {  
3         if (isBefore && (isInsert || isUpdate)) {  
4             preventOrderIfOutOfStock(newOrders);  
5         }  
6         if (isAfter && (isInsert || isUpdate)) {  
7             updateStockOnOrderPlacement(newOrders);  
8         }  
9     }  
10    // ✘ Prevent placing an order if stock is zero  
11    private static void preventOrderIfOutOfStock(List<Vehicle_Order__c> orders) {  
12        Set<Id> vehicleIds = new Set<Id>();  
13        for (Vehicle_Order__c order : orders) {  
14            if (order.Vehicle__c != null) {  
15                vehicleIds.add(order.Vehicle__c);  
16            }  
17        }  
18        if (!vehicleIds.isEmpty()) {  
19            Map<Id, Vehicle__c> vehicleStockMap = new Map<Id, Vehicle__c>(  
20                [SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds]  
21            );  
22            for (Vehicle_Order__c order : orders) {  
23                if (order.Vehicle__c != null && vehicleStockMap.get(order.Vehicle__c).Stock_Quantity__c <= 0) {  
24                    preventOrderIfOutOfStock(order);  
25                }  
26            }  
27        }  
28    }  
29}
```

Flood warning In effect 204 pm 26/11/2025

Developer Console - Google Chrome
orgfarm-6a9b75c4da-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,  
2     VehicleOrderTriggerHandler.handleTrigger(Trigger.new, Trigger.oldMap, Trigger.isBefore,  
3 );  
4 }
```

Logs	Tests	Checkpoints	Query Editor	View State	Progress	Problems
User Kristine Mae Tabbuga	Application Browser	Operation /aura	Time 11/26/2025, 12:12:51 ...	Status Success	Read Unread	Size 7.8 KB
Kristine Mae Tabbuga	Unknown	common.api.soap.Dire...	11/26/2025, 12:12:50 ...	Success	Unread	526 bytes
Kristine Mae Tabbuga	Browser	/aura	11/26/2025, 12:10:31 ...	Success	Unread	16.33 KB
Kristine Mae Tabbuga	Unknown	common.api.soap.Dire...	11/26/2025, 12:10:31 ...	Success	Unread	526 bytes

Developer Console - Google Chrome
 orgfarm-6a9b75c4da-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 = 'Pend...
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      );
21
22      for (Id id : vehicleIds) {
23        Vehicle__c vehicle = vehicleStockMap.get(id);
24        if (vehicle != null) {
25          vehicleStockMap.put(id, vehicle);
26        }
27      }
28
29      for (Vehicle_Order__c order : orderList) {
30        if (order.Vehicle__c != null) {
31          Vehicle__c vehicle = vehicleStockMap.get(order.Vehicle__c);
32          if (vehicle != null) {
33            order.Vehicle__c = vehicle;
34          }
35        }
36      }
37
38      update orderList;
39    }
40  }
41
42  ▼ global void finish(Database.BatchableContext bc) {
43  }
44}
  
```

Logs Tests Checkpoints Query Editor View State Progress Problems

User	Application	Operation	Time	Status	Read	Size
Kristine Mae Tabbuga	Browser	/aura	11/26/2025, 12:12:51 ...	Success	Unread	7.8 KB
Kristine Mae Tabbuga	Unknown	common.api.soap.Dire...	11/26/2025, 12:12:50 ...	Success	Unread	526 bytes
Kristine Mae Tabbuga	Browser	/aura	11/26/2025, 12:10:31 ...	Success	Unread	16.33 KB
Kristine Mae Tabbuga	Unknown	common.api.soap.Dire...	11/26/2025, 12:10:31 ...	Success	Unread	526 bytes

Filter

Developer Console - Google Chrome
 orgfarm-6a9b75c4da-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 VehicleOrderBatchScheduler implements Schedulable {
2   ▼ global void execute(SchedulableContext sc) {
3     VehicleOrderBatch batchJob = new VehicleOrderBatch();
4     Database.executeBatch(batchJob, 50); // 50 = batch size
5   }
6
7 }
  
```

Logs Tests Checkpoints Query Editor View State Progress Problems

User	Application	Operation	Time	Status	Read	Size
Kristine Mae Tabbuga	Browser	/aura	11/26/2025, 12:12:51 ...	Success	Unread	7.8 KB
Kristine Mae Tabbuga	Unknown	common.api.soap.Dire...	11/26/2025, 12:12:50 ...	Success	Unread	526 bytes
Kristine Mae Tabbuga	Browser	/aura	11/26/2025, 12:10:31 ...	Success	Unread	16.33 KB
Kristine Mae Tabbuga	Unknown	common.api.soap.Dire...	11/26/2025, 12:10:31 ...	Success	Unread	526 bytes

Phase 5: Deployment, Documentation & Maintenance

- **Deployment Strategy:** Used change sets to move configurations from the sandbox environment to the production org.
- **System Maintenance:** Regularly checked scheduled automation jobs and reviewed system logs to keep everything running smoothly.
- **Troubleshooting:** Kept a record of common problems related to triggers, flows, and batch jobs, along with the steps taken to fix them.

Conclusion

The **WhatNext Vision Motors Salesforce CRM** project effectively improved customer ordering, inventory tracking, and dealer assignment processes. By using Apex triggers, Flows, and Batch Apex, the system minimized manual mistakes, enhanced data accuracy, and made operations faster. This project also offered valuable hands-on experience in developing a scalable Salesforce solution that boosts both business efficiency and customer satisfaction.

Future Enhancements

- Add AI-driven recommendations to optimize customer orders.
- Implement a chatbot to handle customer inquiries automatically.
- Improve mobile access for field sales teams and dealer operations.