

# **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.

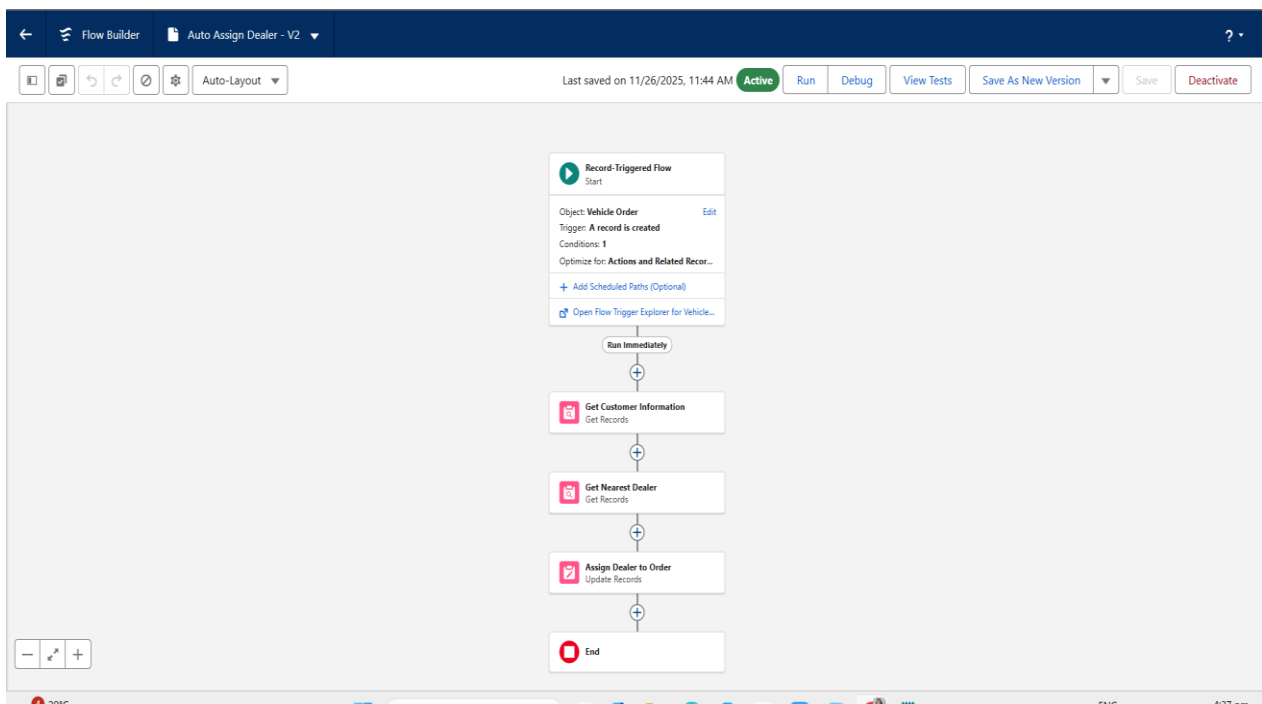
## Phase 2: Salesforce Development – Backend & Configuration

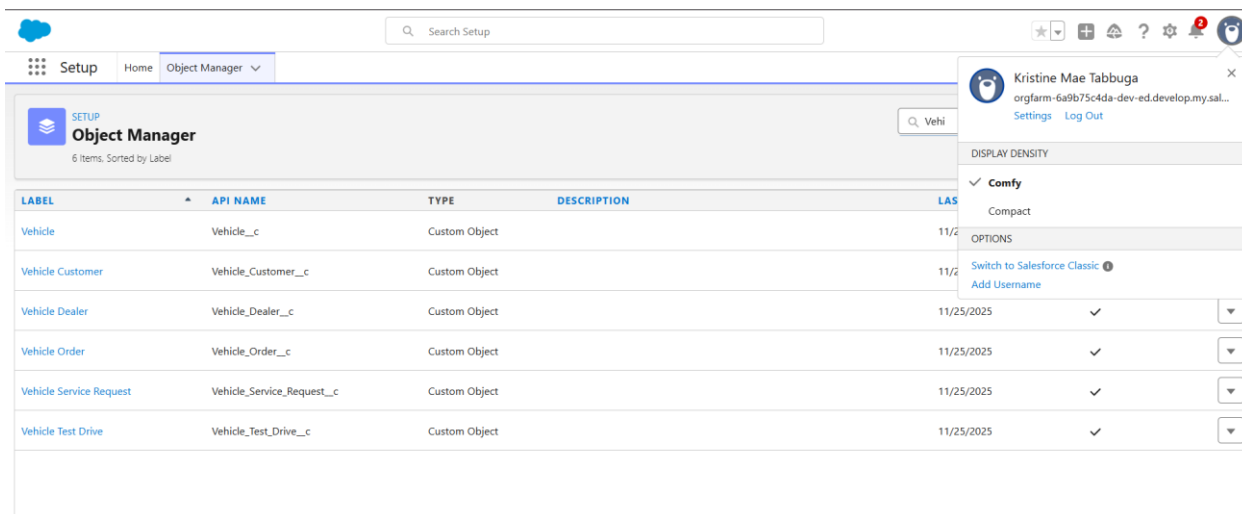
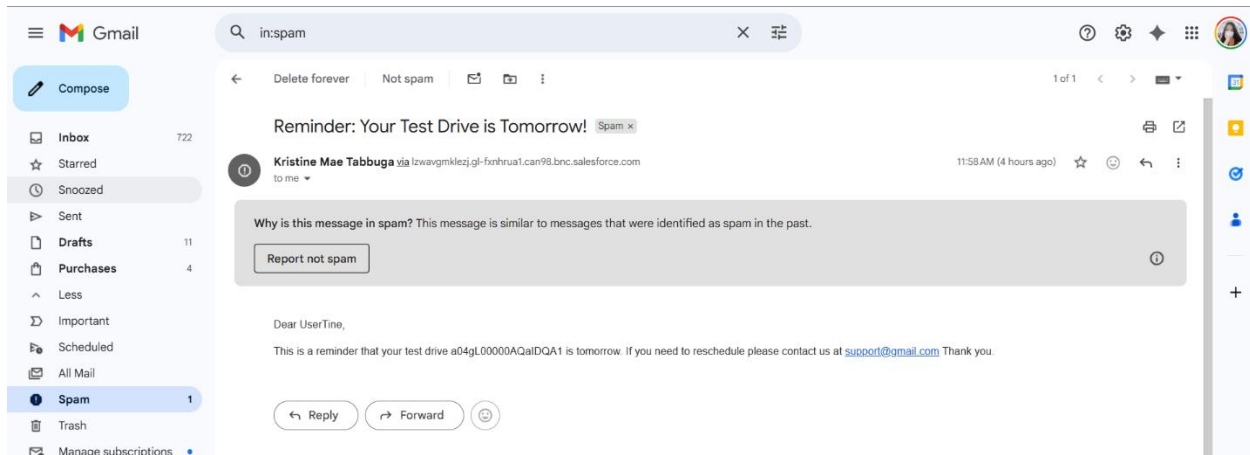
### Setup Environment & DevOps Workflow

- 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.





### 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.

The screenshot shows the Salesforce Chatter interface for the 'Vehicle Customers' object. The top navigation bar includes the Salesforce logo, a search bar, and various utility icons. The main header shows 'Vehicle Customers' and a 'Recently Viewed' dropdown menu. Below this, a table lists recently viewed items, with one item visible: 'Vehicle Customer Name' with a 'Time' column. On the right, a user profile sidebar for Kristine Mae Tabbuga is open, displaying her name, email, and a 'Quick Find' search bar. Below the search bar is a list of 'USERS' with their names and email addresses.

Vehicle Customer Name	Time
1	Time


USERS

- jayjay1@email.com
- kris@email.com
- bebe@email.com








The screenshot shows the Salesforce Chatter interface for the 'Vehicle Dealers' object. The top navigation bar is similar to the previous one. The main header shows 'Vehicle Dealers' and a 'Recently Viewed' dropdown menu. Below this, a table lists recently viewed items, with two items visible: 'Mae' and 'Ben'. On the right, a sidebar contains action buttons: 'New', 'Import', 'Change Owner', and 'Assign Label'. Below these buttons is a search bar and a set of icons for various actions.

Vehicle Dealer Name	
1	Mae
2	Ben

New Import Change Owner Assign Label



Q Search...



Salesforce ChatterHomeChatterPeopleGroupsFilesVehiclesVehicle CustomersVehicle DealersVehicle OrdersVehicle Service RequestsVehicle Test Drives







Vehicle Orders

Recently Viewed


NewImportChange OwnerAssign Label

2 items • Updated a few seconds ago








Q Search this list...



	Vehicle Order Number	
1	<input type="checkbox"/> O-0002	
2	<input type="checkbox"/> O-0001	



Q Search...



Salesforce ChatterHomeChatterPeopleGroupsFilesVehiclesVehicle CustomersVehicle DealersVehicle OrdersVehicle Service RequestsVehicle Test Drives







Vehicle Test Drives

Recently Viewed


NewImportChange OwnerAssign Label

1 item • Updated a few seconds ago








Q Search this list...



	Vehicle Test Drive Name	
1	<input type="checkbox"/> Len	



Q Search...



Salesforce ChatterHomeChatterPeopleGroupsFilesVehiclesVehicle CustomersVehicle DealersVehicle OrdersVehicle Service RequestsVehicle Test Drives







Vehicles

Recently Viewed

NewImportChange OwnerAssign Label

1 item • Updated a few seconds ago

Q Search this list...

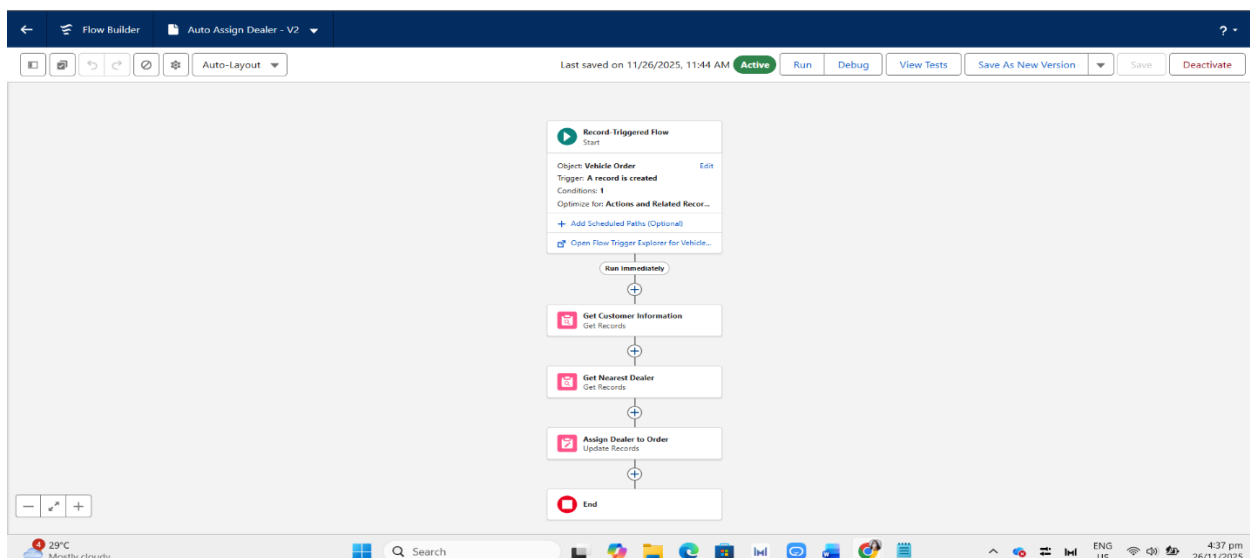


	Vehicle Name	
1	<input type="checkbox"/> Honda	

## 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 the 'New Vehicle Order' form in Salesforce. The form is titled 'New Vehicle Order' and has a sub-header 'Information'. It contains several fields: 'Vehicle Order Number', 'Vehicle Customer' (with a dropdown menu showing 'Tine'), 'Vehicle' (with a dropdown menu showing 'Honda'), 'Order Date' (with a date picker showing '11/26/2025'), 'Status' (with a dropdown menu showing 'Pending'), and 'Assigned Dealer' (with a search bar). A validation error message is displayed in a red box: 'We hit a snag. Review the errors on this page. This vehicle is out of stock. Order cannot be placed.' The error message is located near the 'Vehicle' field. The form also has buttons for 'Cancel', 'Save & New', and 'Save'.



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
3     public static void handleTrigger(List<Vehicle_Order__c> newOrders, Map<Id, Vehicle_Order__c> oldOrders, Boolean isBefore, Boolean isAfter, Boolean isOnUpdate) {
4         if (isBefore && (isInsert || isUpdate)) {
5             preventOrderIfOutOfStock(newOrders);
6         }
7
8         if (isAfter && (isInsert || isUpdate)) {
9             updateStockOnOrderPlacement(newOrders);
10        }
11    }
12
13    // X 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        }
21
22        if (!vehicleIds.isEmpty()) {
23            Map<Id, Vehicle__c> vehicleStockMap = new Map<Id, Vehicle__c>{
24                [SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds]
25            };
26
27            for (Vehicle_Order__c order : orders) {
28                if (order.Vehicle__c != null && vehicleStockMap.containsKey(order.Vehicle__c) && vehicleStockMap.get(order.Vehicle__c).Stock_Quantity__c == 0) {
29                    return;
30                }
31            }
32        }
33    }
34
35    private static void updateStockOnOrderPlacement(List<Vehicle_Order__c> newOrders) {
36        for (Vehicle_Order__c order : newOrders) {
37            if (order.Vehicle__c != null) {
38                Vehicle__c vehicle = [SELECT * FROM Vehicle__c WHERE Id = :order.Vehicle__c];
39                if (vehicle.Stock_Quantity__c > 0) {
40                    vehicle.Stock_Quantity__c -= order.Stock_Quantity__c;
41                    update vehicle;
42                }
43            }
44        }
45    }
46}
```

Logs, Tests, and Problems

Flood warning In effect

Search

ENG US 2:04 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(Triiger.new, Triiger.oldMap, Triiger.isBefore,
3 )
4 }
```

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



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<Object> {
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 = 'Penc
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     }
23 }
```

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 Click here to filter the log list

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
8
```

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.