

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

Project Overview:

The Salesforce CRM project for WhatNext Vision Motors focuses on enhancing customer experience and streamlining internal operations through automation and smart workflow design. The system simplifies the vehicle ordering process by automatically assigning orders to the nearest dealer, validating stock availability, updating order statuses in real time, and sending scheduled email reminders for test drives. These features address key business needs such as reducing manual work, preventing order errors, improving response time, and ensuring accurate stock management to support a more efficient and customer-centric operation.

Objectives:

The main goal of this CRM project is to enhance the end-to-end customer ordering process. More specifically, it aims to:

- Automatically route customer orders to the most appropriate dealer based on location to accelerate processing.
- Eliminate issues caused by unavailable inventory by applying stock checks through Apex logic and system validations.
- Keep order information consistently up to date using automated batch operations and scheduled Apex routines.
- Equip management with clear, data-driven insights through customized dashboards and reports for smarter decision-making.

By fulfilling these goals, the CRM provides measurable value through quicker order handling, fewer manual mistakes, an improved customer experience, and stronger operational control.

Phase 1: Requirement Analysis & Planning

Understanding Business Requirements

- Manual processes slow down order handling.
- Dealers lack updated stock information.
- Stock checks and order reviews often contain errors.

Defining Project Scope and Objectives

- Automate dealer assignment for each new order.
- Build a centralized system for vehicles, customers, orders, test drives, and dealer records.
- Add validations to block orders for unavailable vehicles.
- Provide real-time dashboards and reports for monitoring.

Design Data Model and Security Model

- **Data Model:** Custom objects include Vehicles, Orders, Customers, Dealers, Test Drives, and Support Requests.
- **Security Model:** Uses roles, profiles, permission sets, and sharing rules for controlled access.

Stakeholders Mapping

- Customers – Place orders and request services.
- Dealers – Manage stock and process orders.
- Sales Team – Handle leads and customer interactions.
- Management – Use dashboards and reports for decisions.

Execution Roadmap

1. Finalize requirements and scope.
2. Develop backend automation.
3. Customize Lightning UX/UI design.
4. Conduct data migration, testing, and security checks.
5. Deploy system, prepare documentation, and maintenance.

Phase 2: Salesforce Development – Backend & Configurations

Setup Environment & DevOps Workflow

- A dedicated Salesforce sandbox was utilized for building features, running tests, and ensuring system stability.
- A deployment process was defined using change sets to move completed components into the production environment.

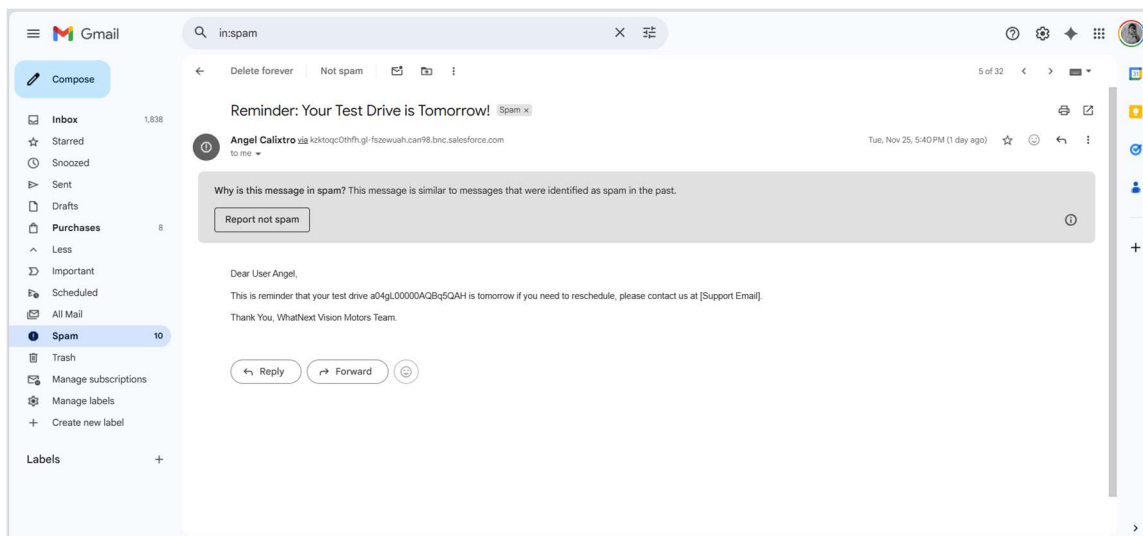
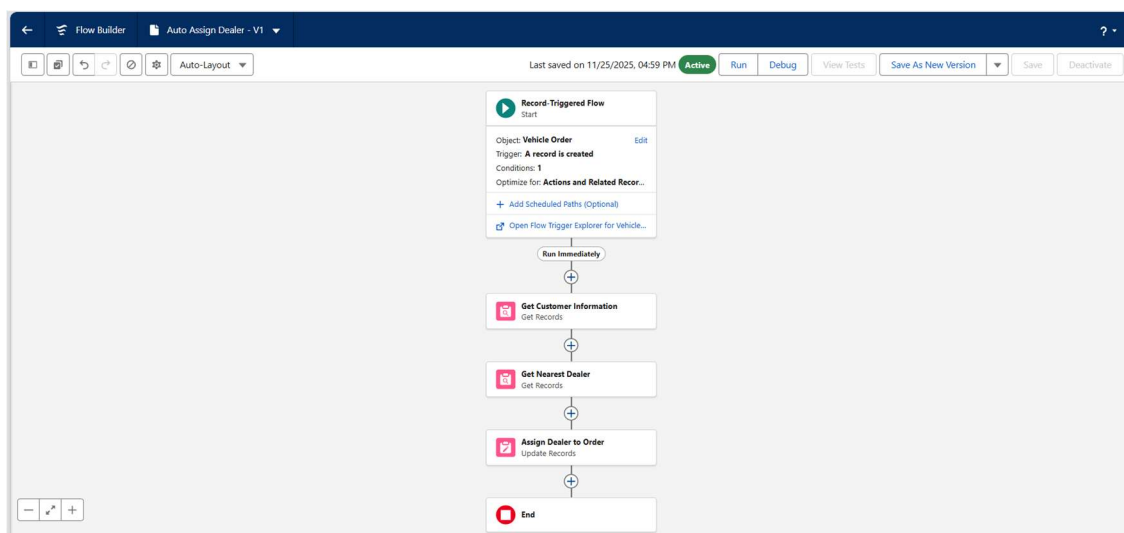
Customizations

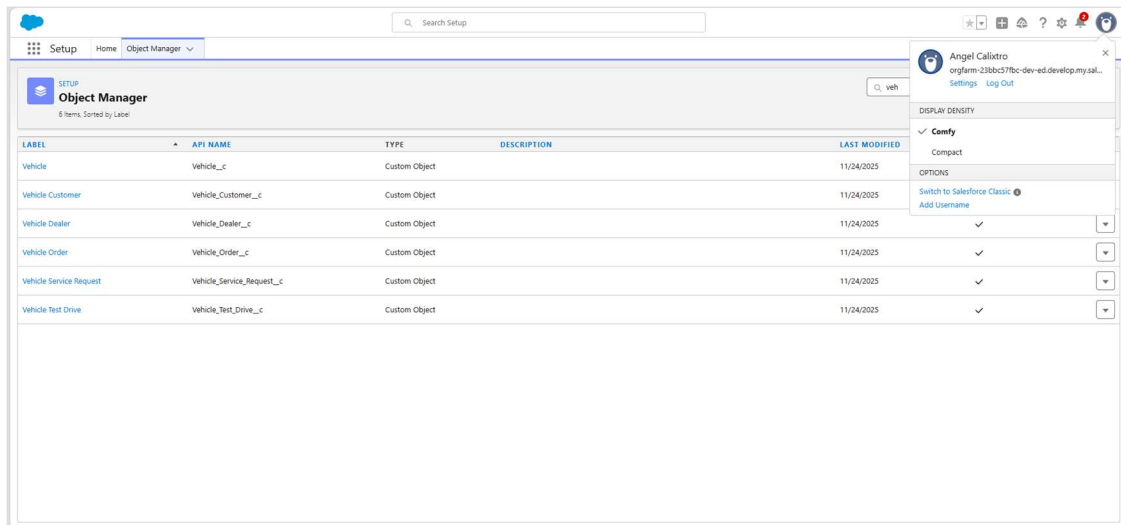
- **Objects & Fields:** Created for Vehicles, Dealers, Customers, Orders, Test Drives, and Support Requests.

- **Validation Rules & Apex Triggers:** Prevent orders when stock is insufficient.

- **Automation:**

- **Flows:** Auto-assigns the nearest dealer for new orders.
- **Workflow Rules & Process Builder:** Sends email notifications and updates order statuses.
- **Batch & Scheduled Apex:** Handles automated updates to stock and orders.





Phase 3: UI/UX Development & Customization

- **Lightning App Setup:** 'WhatNext Vision' app was created and configured using the App Manager.
- **Page Layouts & Dynamic Forms:** Designed for smooth navigation and usability.
- **User Management:** Profiles, roles, and permission sets were established to control access and permissions.
- **Reports & Dashboards:** Custom dashboards and reports were developed to monitor orders, inventory, and dealer activity.
- **Lightning Components:** Lightning Web Components (LWC) were optionally implemented to enhance interactivity.
- **Lightning Pages:** Pages were structured to present object data clearly and efficiently for users.

WhatNext Vision Motors

Vehicles

Vehicle Dealers

Vehicle Customers

Vehicle Orders

Vehicle Service Requests

Vehicle Test Drives

Reports

Dashboards

Search...

Star

Plus

Home

Help

Settings

Notifications

User

Vehicles

Recently Viewed

NewImportChange OwnerAssign Label

1 item • Updated a few seconds ago

Search this list...

Vehicle Name

1Toyota

WhatNext Vision Motors

Vehicles

Vehicle Dealers

Vehicle Customers

Vehicle Orders

Vehicle Service Requests

Vehicle Test Drives

Reports

Dashboards

Search...

Star

Plus

Home

Help

Settings

Notifications

User

Vehicle Dealers

Recently Viewed

NewImportChange OwnerAssign Label

2 items • Updated a few seconds ago

Search this list...

Vehicle Dealer Name

1Liam

2Ian

WhatNext Vision Motors

Vehicles

Vehicle Dealers

Vehicle Customers

Vehicle Orders

Vehicle Service Requests

Vehicle Test Drives

Reports

Dashboards

Search...

Star

Plus

Home

Help

Settings

Notifications

User

Vehicle Customers

Recently Viewed

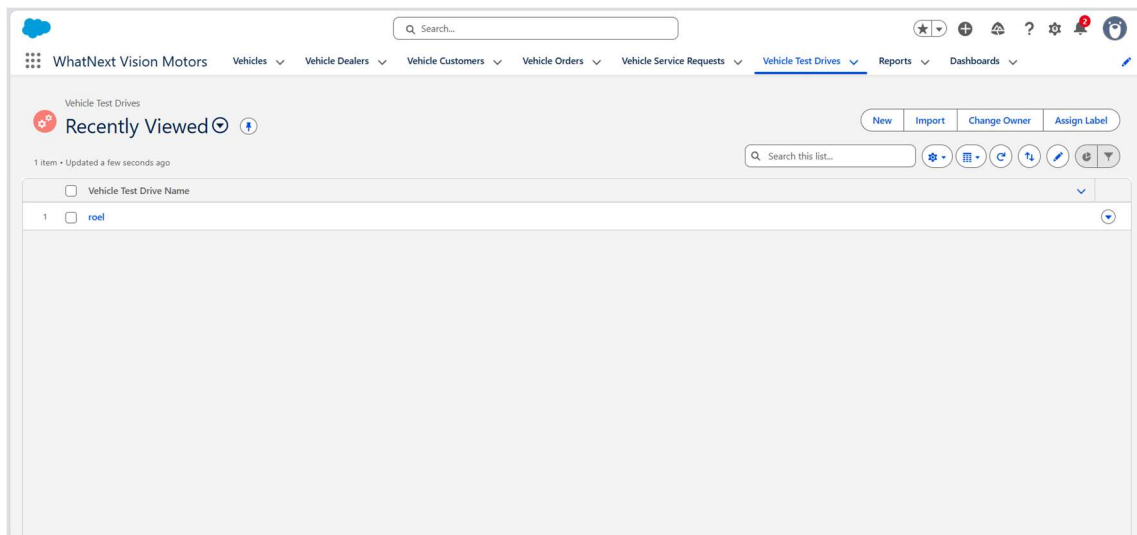
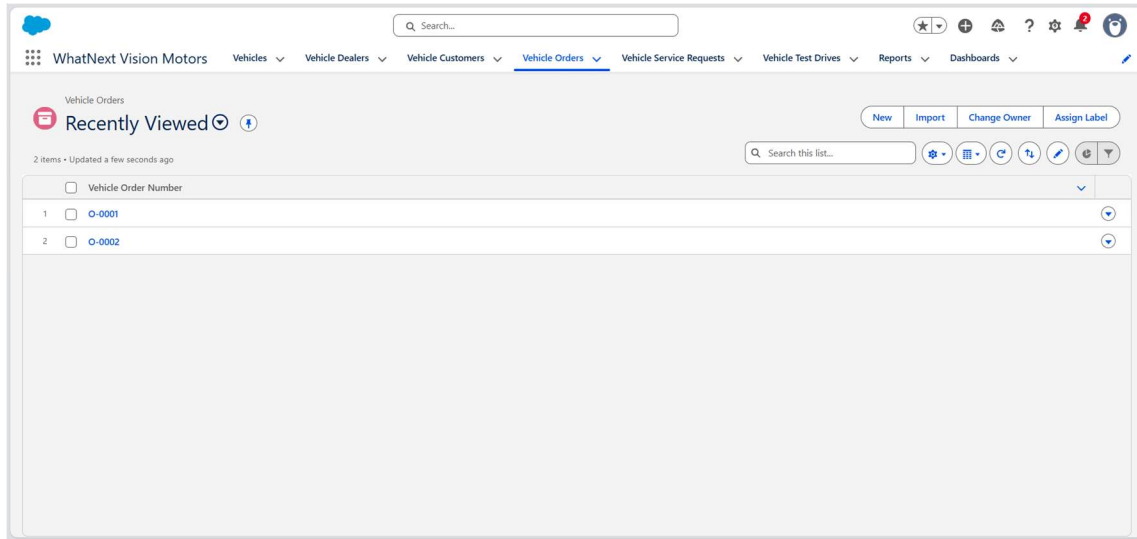
NewImportChange OwnerAssign Label

1 item • Updated a few seconds ago

Search this list...

Vehicle Customer Name

1Angel



Phase 4: Data Migration, Testing & Security

- **Data Loading:** Transferred Vehicles, Dealers, and Customer records into Salesforce using Data Loader and Data Import Wizard.
- **Data Accuracy:** Implemented field history tracking, duplicate rules, and matching rules to maintain clean and reliable data.
- **Security Configuration:** Set up roles, profiles, role hierarchy, and permission sets to control data access.
- **Apex Testing:** Created test classes to validate triggers, batch processes, and automated

workflows.

- **Testing Procedures:**

- Verified the Auto Dealer Assignment flow.
- Tested approval processes, validation rules, and task automation to ensure proper functionality.
-

WhatNext Vision Motors

Vehicle Customers Vehicle Dealers Vehicle Orders Vehicle Service Requests Vehicles More

New Vehicle Order

* = Required Information

Information

Vehicle Order Number

Owner Angel Calixtro

Vehicle Customer Angel

Vehicle Toyota

Order Date 11/27/2025

Status Pending

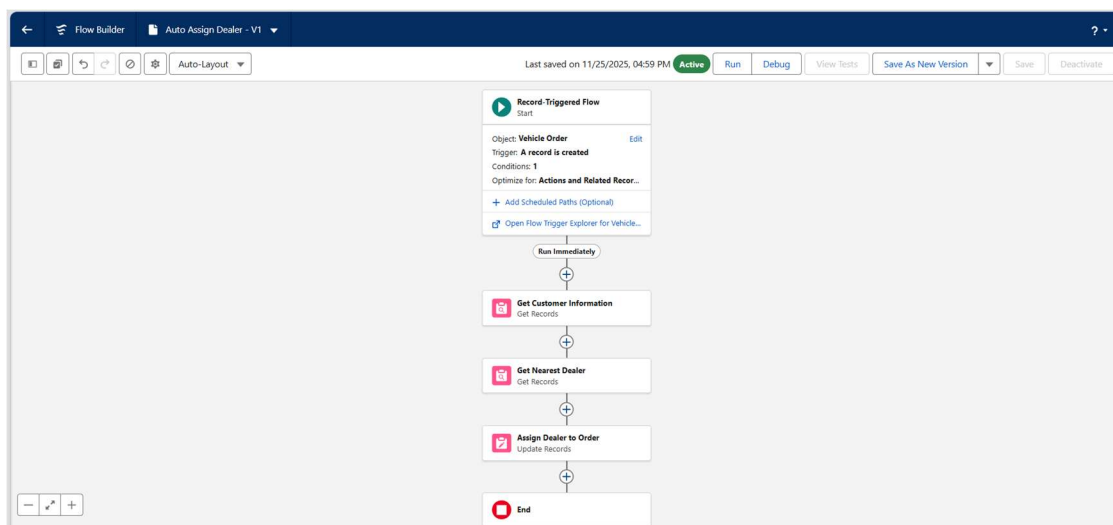
Assigned Dealer Search Vehicle Dealers...

We hit a snag.

Review the errors on this page.

- This vehicle is out of stock. Order cannot be placed.

Cancel Save & New Save



Developer Console - Personal - Microsoft Edge

https://orgfarm-23bbc57fbc-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)
4
5     {
6         if (isBefore) {
7             if (isInsert || isUpdate) {
8                 preventOrderIfOutOfStock(newOrders);
9             }
10        }
11    }
12
13 }
14
15
16
```

Logs Tests Checkpoints Query Editor View State Progress Problems

User	Application	Operation	Time	Status	Read	Size
------	-------------	-----------	------	--------	------	------

Filter Click here to filter the log list

Developer Console - Personal - Microsoft Edge

https://orgfarm-23bbc57fbc-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);
3 }

```

Logs Tests Checkpoints Query Editor View State Progress Problems

User	Application	Operation	Time	Status	Read	Size
------	-------------	-----------	------	--------	------	------

Filter Click here to filter the log list

The screenshot shows the Salesforce Developer Console with the `VehicleOrderBatch.apex` file open. The code defines a global class `VehicleOrderBatch` that implements `Database.Batchable<Object>`. It includes a `start` method to get a query locator for pending orders and an `execute` method to process a list of orders by adding their IDs to a set.

```
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 = '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 }
```

The interface at the bottom shows tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Logs tab is active, displaying a table with columns: User, Application, Operation, Time, Status, Read, and Size.

The screenshot shows the Salesforce Developer Console with the `VehicleOrderBatchScheduler.apex` file open. The code defines a global class `VehicleOrderBatchScheduler` that implements `Schedulable`. It includes an `execute` method that creates a new `VehicleOrderBatch` instance and executes it as a batch job with a size of 50.

```
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 }
```

The interface at the bottom shows tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Logs tab is active, displaying a table with columns: User, Application, Operation, Time, Status, Read, and Size.

Phase 5: Deployment, Documentation & Maintenance

- **Deployment:** Configurations were transferred from sandbox to production using change sets.
- **Maintenance:** System performance, scheduled automation, and logs are regularly monitored to ensure smooth operation.
- **Troubleshooting:** Common issues with triggers, flows, and batch processes were recorded along with their solutions.

Conclusion

The Salesforce CRM for WhatNext Vision Motors effectively improved the customer ordering process, inventory control, and dealer assignment workflows. Automation with Apex triggers, Flows, and Batch Apex minimized errors, enhanced data accuracy, and sped up operations. The project provided practical experience in developing a scalable solution that boosts efficiency and customer satisfaction.

Future Enhancements

- Incorporate AI-driven recommendations to optimize orders.
 - Implement a chatbot for automatic customer support.
- Improve mobile accessibility for sales teams and dealer operations in the field.