

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

## **A Salesforce CRM Project Documentation**

### **Project Overview**

WhatNext Vision Motors is a Salesforce-based dealership management system that helps automobile companies organize their operations. Users can manage vehicle inventory, client information, and vehicle orders from a single, centralized platform. The solution uses Salesforce's automation and CRM features to accelerate order processing, decrease manual effort, and improve customer experience. The goal is to give dealerships with a more efficient, accurate, and user-friendly approach to manage their everyday operations.

### **Abstract**

This project aims to improve the way WhatNext Vision Motors manages clients, car inventories, and orders with Salesforce. The system organizes key data, such as vehicles, dealers, and customer information, into a logical and interconnected framework. Salesforce automation technologies, such as record-triggered flows, batch Apex, and Apex triggers, improve process speed, accuracy, and reliability.

Key features include:

- Preventing customers from ordering vehicles that are out of stock.
- Automatically assigning orders to the nearest dealer based on customer location.
- Updating order statuses to "Pending" or "Confirmed" through scheduled batch processes.
- Sending automated email notifications for test drives, low inventory, and order updates.

The technology improves customer happiness and streamlines dealership operations by decreasing human labor, errors, and response times. Overall, WhatNext Vision Motors benefits from a Salesforce-powered CRM that is current, organized, and efficient.

### **General Objective**

To create a Salesforce-based dealership management system that will allow WhatNext Vision Motors to organize vehicle data, customer information, and vehicle orders more quickly and efficiently.

### **Specific Objectives**

- To create a structured Salesforce database for vehicles, customers, and dealers.
- To automate key dealership tasks such as generating order numbers, checking stock, and updating order status.
- To suggest the nearest dealer to customers based on their location for a faster and more convenient ordering experience.
- To allow customers to order only vehicles that are in stock, ensuring clarity and satisfaction.
- To automatically update order status and notify customers and staff via email, keeping everyone informed.
- To provide an intuitive interface for staff to easily manage records and reduce manual errors.

- To improve overall efficiency and customer service by streamlining workflows and access to information.

## **Methodology / Technique Used**

### **Salesforce**

Salesforce is a cloud-based CRM software that helps businesses manage client information, interactions, and procedures more effectively. This project saves and organizes vehicle information, dealer locations, and customer orders. The system uses automation tools such as Flows, Apex triggers, and batch processes to maintain precise stock tracking and send orders to the nearest dealer. This enables WhatsNext Vision Motors to streamline operations and provide a faster, more convenient experience to customers.

**Custom Data Modelling:** Created custom objects (Vehicles, Customers, Dealers, Orders, etc) to efficiently track orders, inventory, and dealer assignments for WhatsNext Vision Motors.

- Vehicle
- Vehicle Dealer
- Vehicle Customer
- Vehicle Order
- Vehicle Test Drive
- Vehicle Service Request

**Validation Rules:** Implemented in the Salesforce interface to ensure accurate and consistent data entry. These rules prevent incorrect or incomplete information, such as ordering vehicles that are out of stock or entering invalid customer details.

**Flow Builder / Process Automation:** Automated key dealership workflows, including order confirmation, stock availability checks, and dealer assignment based on customer location. These flows reduce manual work and ensure smooth, error-free operations.

**Email Alerts:** Configured to send email notifications to customers one day before their scheduled test drive, ensuring timely reminders. Staff are also alerted about pending orders or low stock as needed. Emails are triggered using record-triggered flows and scheduled flows.

**Scheduled Apex Jobs / Scheduled Flows:** Set up to perform daily automated tasks such as updating bulk order statuses and checking inventory levels.

**Reports & Dashboards:** Visual tools created to monitor orders, vehicle availability, and customer activity. These dashboards help staff track performance, manage inventory, and make informed business decisions quickly.

## **Detailed Execution of Project Plan**

### **1. Developer Account Creation/ Setup**

Salesforce org was created by the link provided,  
<https://developer.salesforce.com/signup>

The account was verified by resetting the password, and our Developer Edition Org was successfully created.

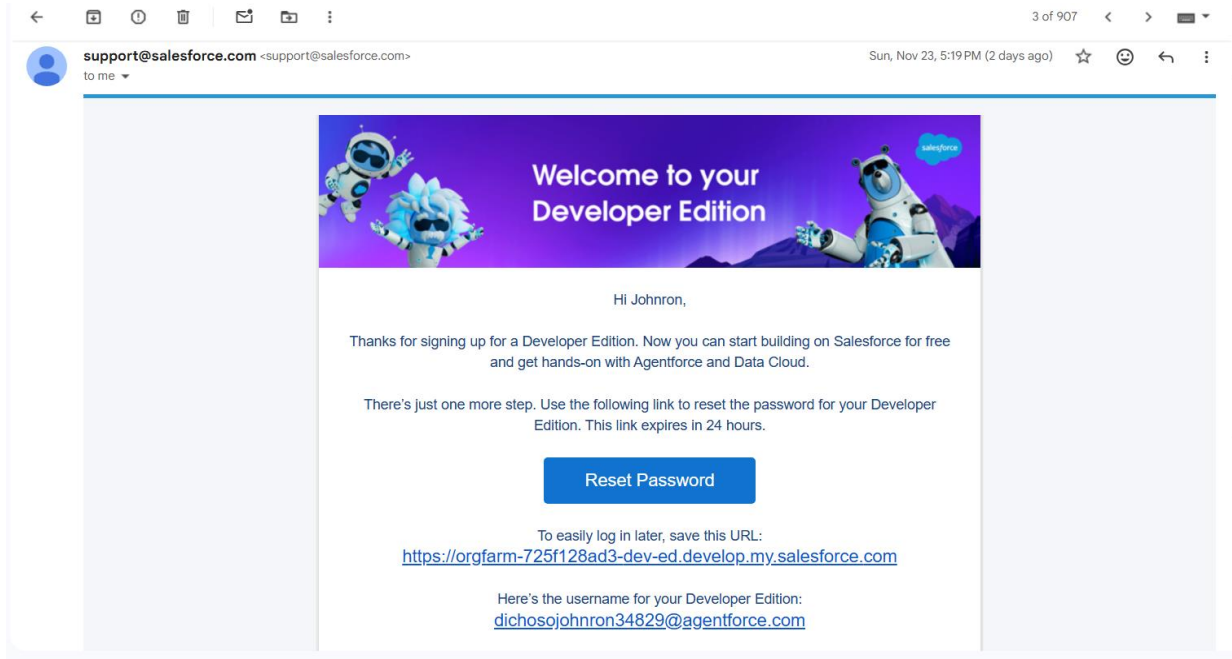


Figure 1:

## 2. Data Management Object

We were required to create some custom objects were:

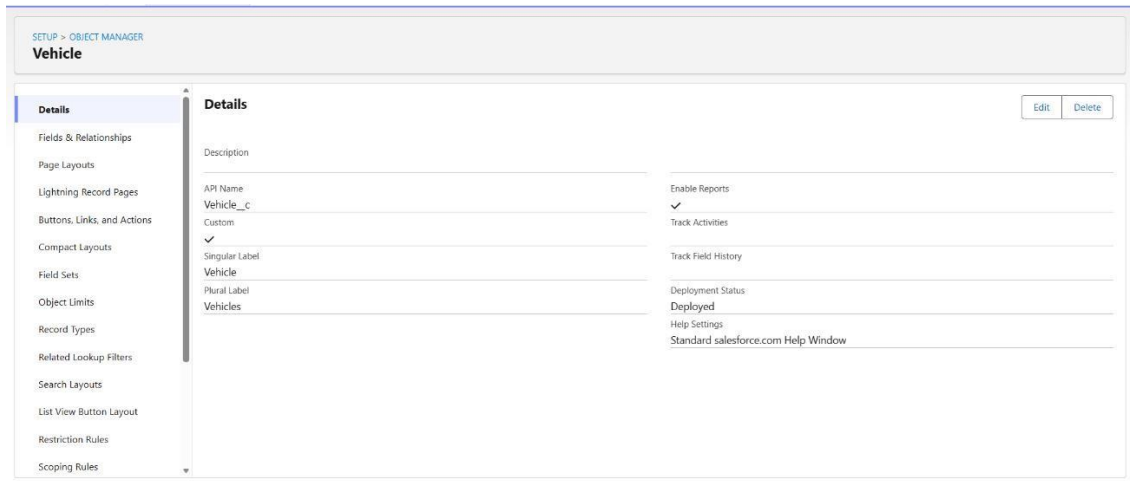
- Vehicle
- Vehicle Dealer
- Vehicle Customer
- Vehicle Order
- Vehicle Test Drive
- Vehicle Service Request

A screenshot of the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. A search bar at the top right contains the text 'vehicle'. Below the navigation bar, the 'Object Manager' section is active, showing a list of 6 items sorted by label. The table has columns for LABEL, API NAME, TYPE, DESCRIPTION, LAST MODIFIED, and DEPLOYED. The objects listed are Vehicle, Vehicle Customer, Vehicle Dealer, Vehicle Order, Vehicle Service Request, and Vehicle Test Drive, all of which are Custom Objects and have been deployed.

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Vehicle	Vehicle_c	Custom Object		11/21/2025	✓
Vehicle Customer	Vehicle_Customer_c	Custom Object		11/22/2025	✓
Vehicle Dealer	Vehicle_Dealer_c	Custom Object		11/21/2025	✓
Vehicle Order	Vehicle_Order_c	Custom Object		11/22/2025	✓
Vehicle Service Request	Vehicle_Service_Request_c	Custom Object		11/21/2025	✓
Vehicle Test Drive	Vehicle_Test_Drive_c	Custom Object		11/21/2025	✓

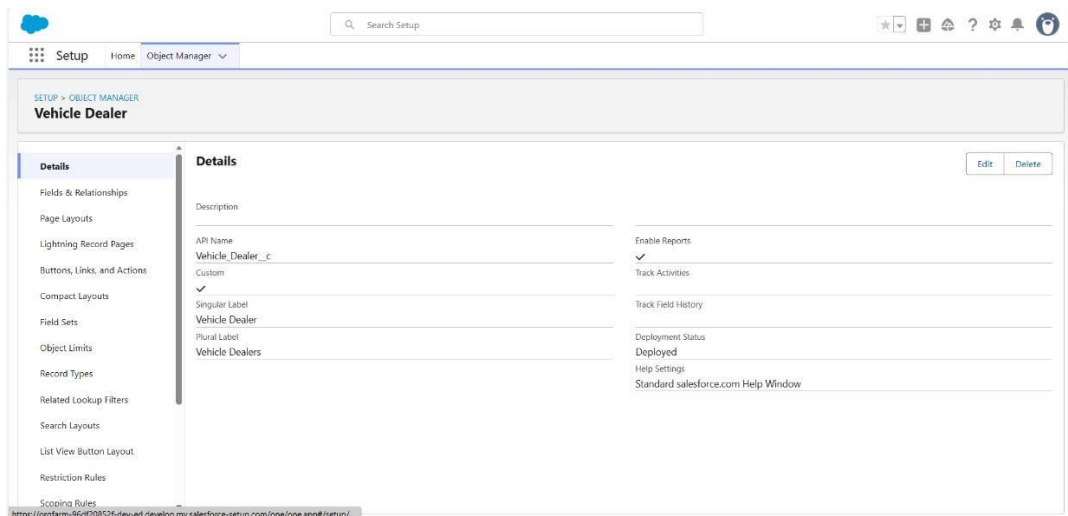
Figure 2: Custom Objects for WhatNext Vision Application

This is created to store all the cars available in the dealership, including the model, price, and stock status.



*Figure 3: Vehicle Custom Object*

This is created to store information about each dealer branch, such as their name and location, so the system can find the nearest dealer for customers.



*Figure 4: Vehicle Dealer Custom Object*

This is created to store customer details, including their name, address, and contact info, which are needed for orders and test drives.

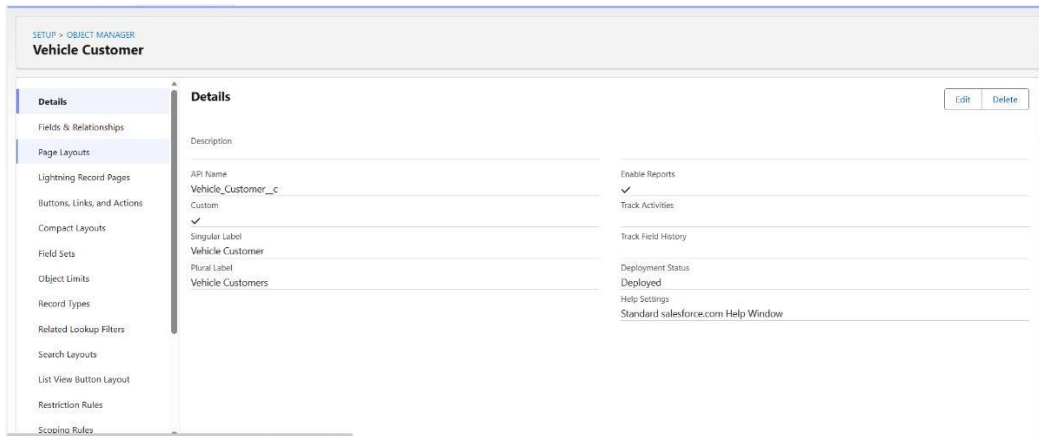


Figure 5: Vehicle Customer Custom Object

This is created to record every vehicle order placed by customers. It keeps track of the car they chose, the order status, and the assigned dealer.

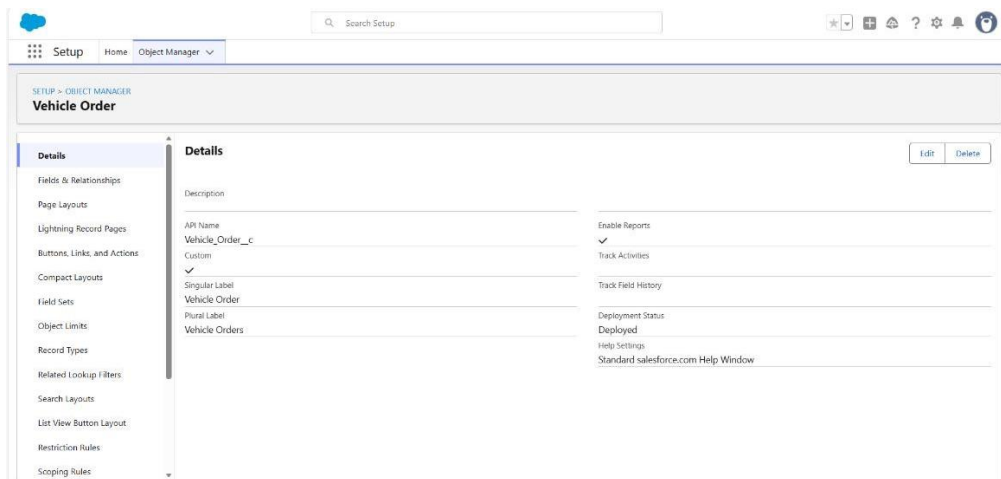
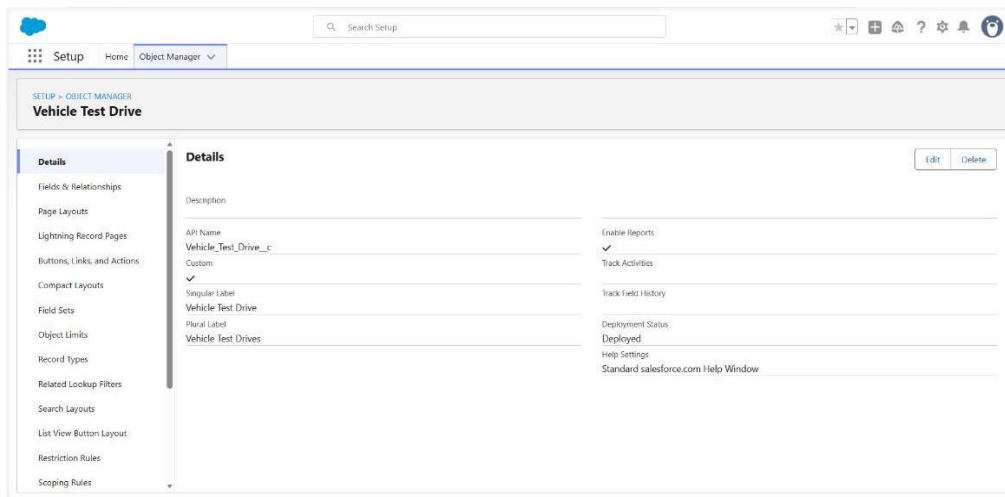


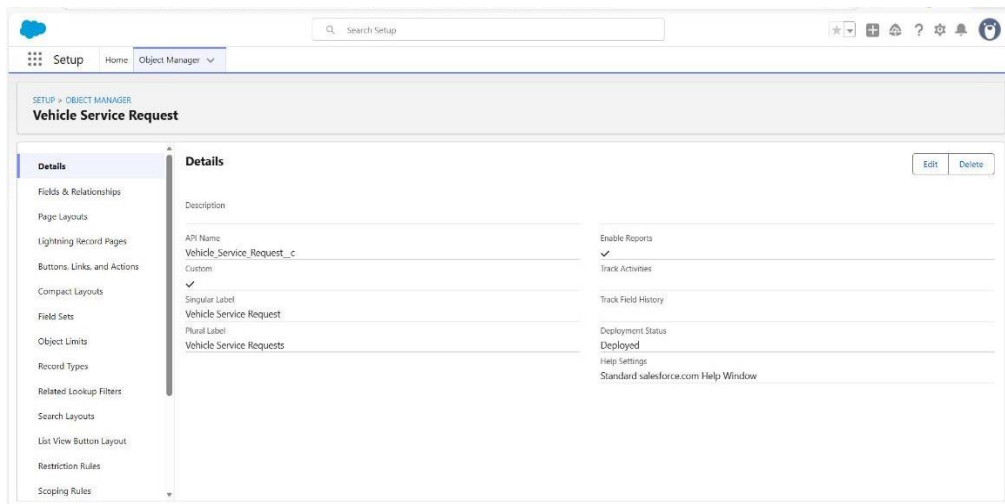
Figure 6: Vehicle Order Custom Object

This is created to manage and schedule customer test drives, including the selected vehicle and the preferred test drive date.



*Figure 7: Vehicle Test Drive Custom Object*

This is created to record customer service or maintenance requests, helping the dealership track what kind of service is needed and when.



*Figure 8: Vehicle Service Request Custom Object*

### 3. Data Management – Tabs

In the WhatNext Vision Motors Salesforce CRM, custom tabs were created to make navigation easier and give users quick access to all important records. These tabs serve as entry points to the custom objects built for the project, such as vehicle details, dealers, customers, orders, test drives, and service requests. By organizing data into easy-to-use tabs, the system allows staff to view, create, and manage dealership records more efficiently, improving both workflow and customer service.

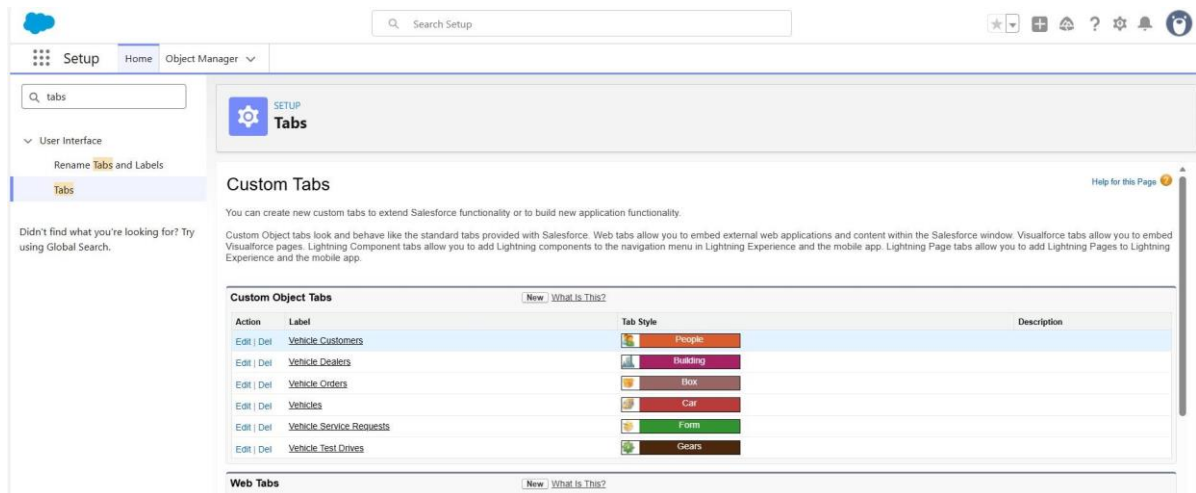


Figure 9: Custom Object Tabs

Tabs Created for these custom objects:

- Vehicle Customers (People)
  - Vehicle Dealers (Building)
  - Vehicle Orders (Box)
  - Vehicles (Car)
  - Vehicle Service Requests (Form)
  - Vehicle Test Drives (Gears)

#### 4. Data Management – App Manager

To keep all system features organized in one place, a custom Lightning App called “WhatNext Vision Motors” was created using Salesforce App Manager. This app serves as the main workspace for users and brings together all important tools needed by the dealership team — such as managing vehicles, checking orders, viewing customer details, and handling test drives or service requests.

By grouping all related tabs and objects into one clean interface, the app makes navigation easier and helps staff work faster and more efficiently.

##### Included Tabs:

The app includes a mix of custom objects built for the project and standard Salesforce objects that support reporting and monitoring.

##### Custom Objects:

- Vehicle
- Vehicle Dealer
- Vehicle Customer
- Vehicle Order
- Vehicle Test Drive
- Vehicle Service Request

##### Standard Objects:

- Reports
- Dashboards

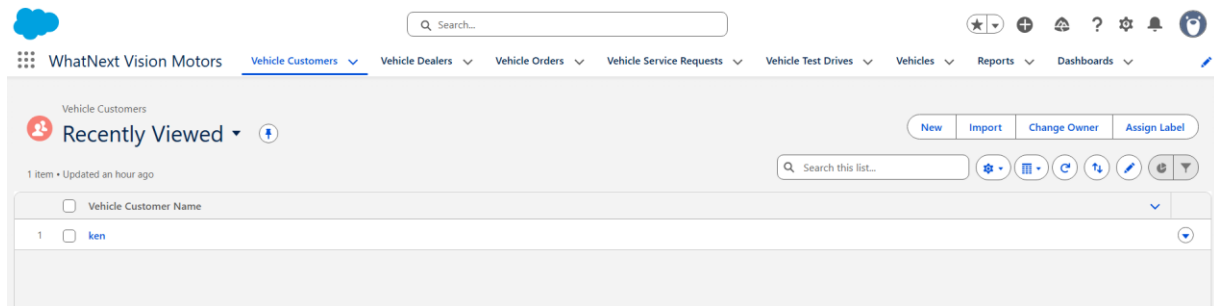


Figure 10: WhatNext Vission Motors Application View

## 5. Data Management – Fields

For the WhatNext Vision Motors CRM, I generated numerous bespoke objects, each of which required its own set of fields. These fields were introduced to allow the system to retain the specific information that the dealership requires, such as automobile data, customer information, dealer locations, order status, test drive schedules, and servicing requests.

By adding the appropriate fields to each object, the system becomes more user-friendly, organized, and capable of supporting all of the project's automations.

### Custom Fields:

Object Name	Key Fields
<b>Vehicle__c</b>	Vehicle Name (Record Name) Vehicle Model (Text) Stock Quantity (Number) Price (Currency) Vehicle Dealer (Look-up) Status (Picklist: Available, Out of Stock)
<b>Vehicle_Dealer__c</b>	Dealer Name (Record Name) Dealer Location (Text) Dealer Code (Text) Phone (Phone) Email (Email)
<b>Vehicle_Customer__c</b>	Customer Name (Record Name) Email (Email) Phone (Phone) Address (Text) Preferred Vehicle Type (Picklist)
<b>Vehicle_Order__c</b>	Order Number (Auto-number) Customer (Lookup) Order Date (Date) Status (Picklist: Pending, Confirmed, Cancelled) Assigned Dealer (Lookup) Delivered,

<b>Vehicle_Test_Drive__c</b>	Test Drive Name (Record Name) Customer (Lookup) Vehicle (Lookup) Test Drive Date (Date) Status (Picklist: Scheduled, Completed, Cancelled)
<b>Vehicle_Service_Request__c</b>	Service Request Name (Record Name) Customer (Lookup) Vehicle (Lookup) Service Date (Date) Issue Description (Text) Status (Picklist: Requested, In Progress, Completed)

Similar to this we have all the fields written inside every custom object as said.

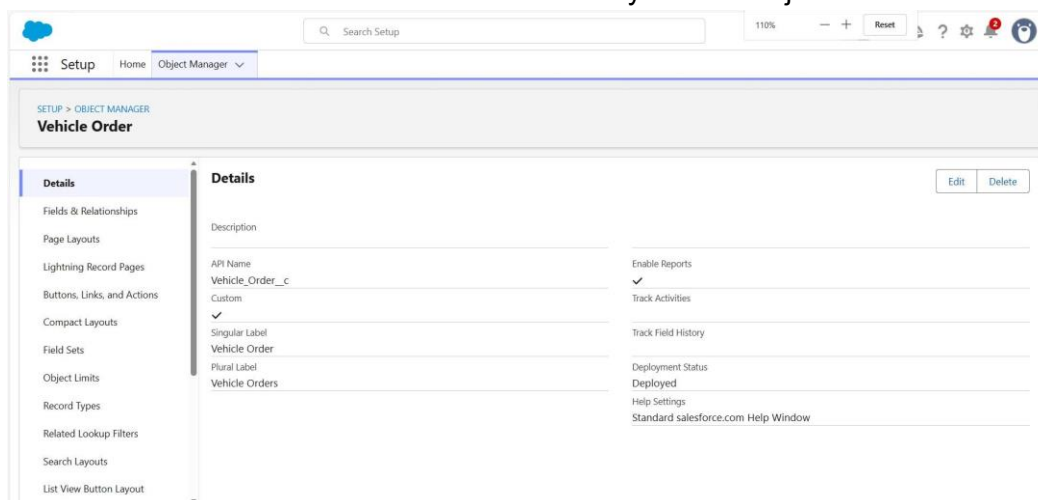


Figure 11: Vehicle Order Custom Object

This diagram shows all the custom objects in the system—Vehicle, Dealer, Customer, Order, Test Drive, and Service Request—and how they are connected. It provides a clear view of the data structure, making it easier to manage orders, stock, and customer interactions efficiently.

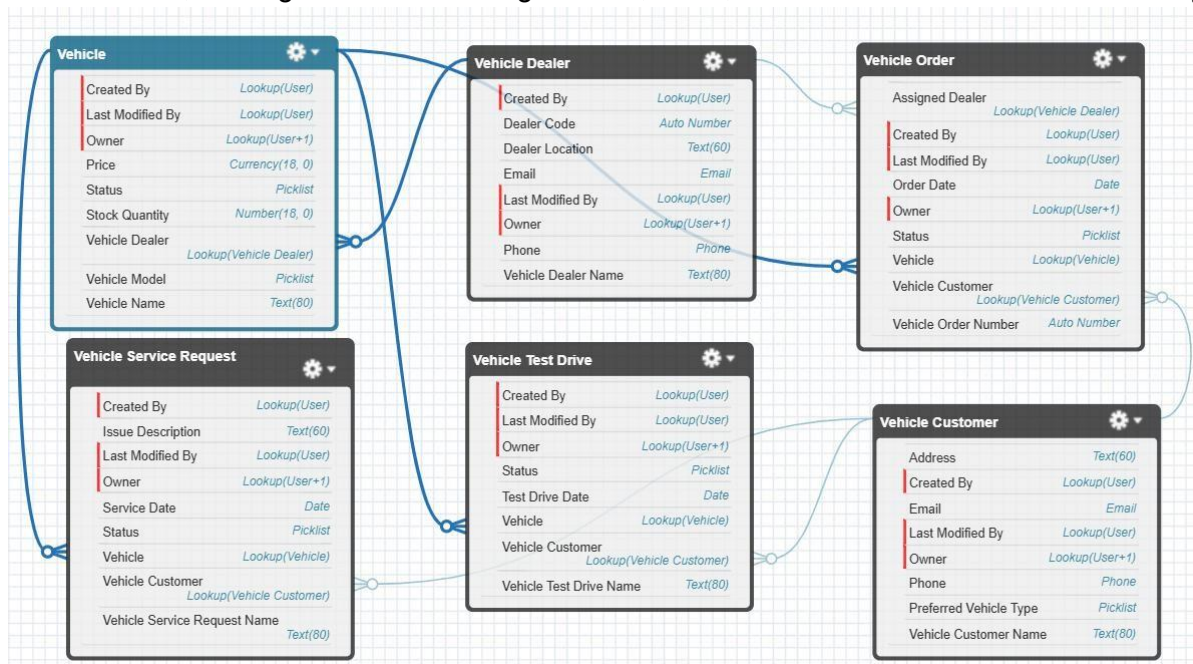


Figure 12: Salesforce Schema Builder for WhatNext Vision Motors

## Creation of Lookup Relationship

Lookup relationships were developed between custom objects to establish links and dependencies, allowing the system to link records and automatically refer to related data.

The project's lookup relationships are listed below:

### 1. Creating a lookup relationship between Vehicle Order and Vehicle Customer

- This connects each order to the customer who placed it, making it easy to track customer orders and history.

### 2. Creating a lookup relationship between Vehicle Order and Vehicle

- This links each order to the specific vehicle being purchased, ensuring that stock and order details are accurately managed.

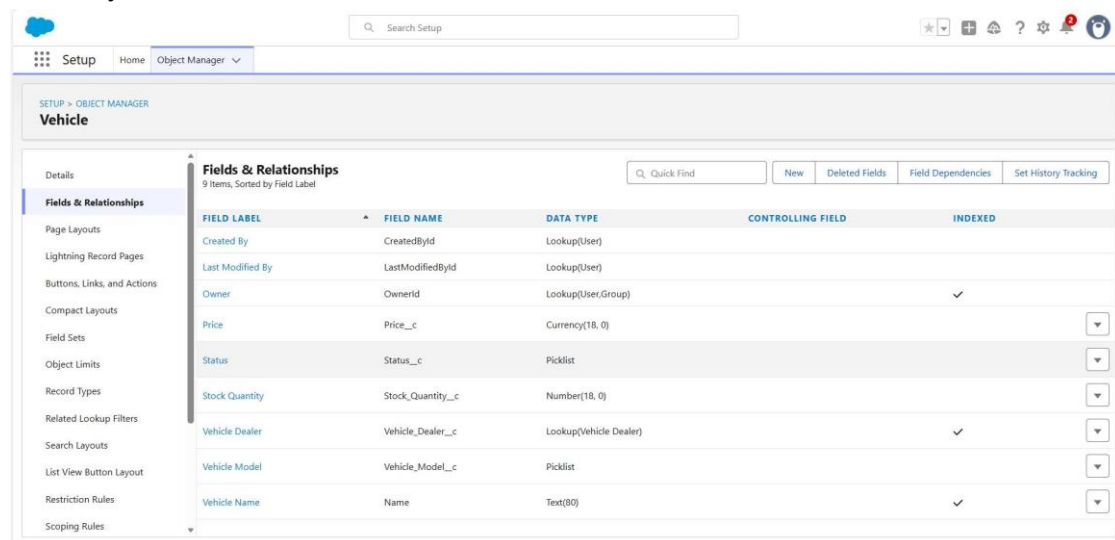
### 3. Creating a lookup relationship between Vehicle and Vehicle Dealer

- This associates each vehicle with a dealer, allowing the system to automatically assign the nearest dealer to a customer's order and track vehicle availability per location.

## 6. Creation of Master – Detail Relationship

A master-detail relationship was developed between custom objects to ensure that changes to one object are automatically reflected in the related object.

In this project, a master-detail relationship was established between Vehicle and Vehicle Order. This enables the system to automatically monitor stock levels: when a customer confirms an order, the vehicle's stock quantity is updated correspondingly. It also guarantees that all vehicle-related data, such as dealer assignment and availability, are correctly linked and maintained.



The screenshot displays the Salesforce Setup interface for the 'Vehicle' custom object. The 'Fields & Relationships' section is active, showing a list of 9 fields. The fields are sorted by Field Label. The table includes columns for Field Label, Field Name, Data Type, Controlling Field, and Indexed status. The fields listed are: Created By, Last Modified By, Owner, Price, Status, Stock Quantity, Vehicle Dealer, Vehicle Model, and Vehicle Name.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User.Group)		✓
Price	Price__c	Currency(18, 0)		
Status	Status__c	Picklist		
Stock Quantity	Stock_Quantity__c	Number(18, 0)		
Vehicle Dealer	Vehicle_Dealer__c	Lookup(Vehicle Dealer)		✓
Vehicle Model	Vehicle_Model__c	Picklist		
Vehicle Name	Name	Text(80)		✓

Figure 13: Vehicle Custom Object Fields and Relationship

## 7. Creation of Formulas Fields

Formula fields were added to the WhatNext Vision Motors Salesforce system to automatically calculate and display values based on other fields. These fields help to reduce manual labor, avoid errors, and provide immediate insights. Some examples of formula fields in the project are:

1. **Order Status Indicator (Vehicle\_Order\_\_c)** – Displays a text message like “Pending” or “Confirmed” based on stock availability or approval.

New Vehicle Order

\* = Required Information

Information

Vehicle Order Number

Owner  
Johnron Dichoso

Vehicle Customer  
ken

Vehicle  
Honda

Order Date  
11/26/2025

Status  
Pending

Assigned Dealer  
Search Vehicle Dealers...

Cancel Save & New Save

Formula: IF(Stock\_Quantity\_\_c > 0, "Confirmed", "Pending")

Figure 14: Vehicle Order Form

2. **Stock Alert (Vehicle\_\_c)** – Shows a warning if the vehicle stock is low.  
Formula: IF(Stock\_Quantity\_\_c <= 5, "Low Stock", "In Stock")

New Vehicle Order

\* = Required Information

Information

Vehicle Order Number

Owner  
Johnron Dichoso

Vehicle Customer  
ken

Vehicle  
Honda

Order Date  
11/26/2025

Status  
Pending

Assigned Dealer  
justin

We hit a snag.  
Review the errors on this page.  
• This vehicle is out of stock. Order cannot be placed.

Cancel Save & New Save

Figure 15: Vehicle Order Form showing warning

- Dealer Assignment Check (Vehicle\_Order\_\_c)** – Confirms if an order has been assigned to a dealer or not.  
Formula: IF(ISBLANK(Assigned\_Dealer\_\_c), "Dealer Not Assigned", "Dealer Assigned")

Vehicle Order  
**O-0002**

Related
Details

Vehicle Order Number	O-0002	Owner	Johnron Dichoso
Vehicle Customer	ken		
Vehicle	Honda		
Order Date	11/24/2025		
Status	Confirmed		SSSSS
Assigned Dealer	jacob		
Created By	Johnron Dichoso, 11/23/2025, 4:58 AM	Last Modified By	Johnron Dichoso, 11/23/2025, 4:58 AM

Vehicle Dealer  
**jacob**

Related
Details

Vehicle Dealer Name	jacob	Owner	Johnron Dichoso
Dealer Location	Hyderabad		
Dealer Code	DC-0001		
Phone	(123) 456-7890		
Email	dichosojohnron34@gmail.com		
Created By	Johnron Dichoso, 11/23/2025, 3:40 AM	Last Modified By	Johnron Dichoso, 11/23/2025, 3:40 AM

Vehicle Customer  
**ken**

Related
Details

Vehicle Customer Name	ken	Owner	Johnron Dichoso
Email	dichosojohnron34@gmail.com		
Phone	09649692648		
Address	Philippines		
Preferred Vehicle Type	Sedan		
Created By	Johnron Dichoso, 11/23/2025, 3:38 AM	Last Modified By	Johnron Dichoso, 11/23/2025, 3:38 AM

Figure 16: Assigned Dealer Based on Customer Location

As shown, the system automatically assigns the order to the vehicle dealer located closest to the customer, ensuring efficient service and faster processing.

---

## 8. Data Configuration – Validation Rules

To ensure data accuracy, enforce business logic, and prevent errors, several Validation Rules were implemented across the custom objects in the WhatNext Vision Motors CRM system. These rules help maintain clean, reliable data and prevent incorrect or illogical records from being saved.

### 1. Vehicle\_Order\_\_c → Status

- **Validation Rule Name:** “Valid Order Status”
- **Rule:** ISBLANK(Status\_\_c)
- **Purpose:** Ensures that every order has a status assigned before it can be saved

### 2. Vehicle\_\_c → Stock\_Quantity\_\_c □ **Validation Rule Name:** “Stock Quantity”

- **Rule:** Stock\_Quantity\_\_c < 0
- **Purpose:** Prevents vehicles from having negative stock values.

### 3. Vehicle\_Customer\_\_c → Email □ **Validation Rule Name:** “Valid Email”

- **Rule:** NOT(CONTAINS(Email, "@"))
- **Purpose:** Ensures that every customer record has a valid email address format.

### 4. Vehicle\_Order\_\_c → Vehicle □ **Validation Rule Name:** “Vehicle Required”

- **Rule:** ISBLANK(Vehicle\_\_c)
  - **Purpose:** Prevents saving an order if no vehicle is selected.
- 

## 9. Automated Email Reminders for Test Drives

To improve customer service and ensure smooth scheduling, the system sends automated email reminders to customers one day before their scheduled test drive.

- **Object Involved:** Vehicle\_Test\_Drive\_\_c
- **Automation:** Record-Triggered Flow □ **Action Name:** Send Test Drive Reminder □  
**Purpose:**
  1. Reminds customers of their upcoming test drive
  2. Reduces missed appointments and improves dealership efficiency.
  3. Provides a professional, customer-friendly experience.

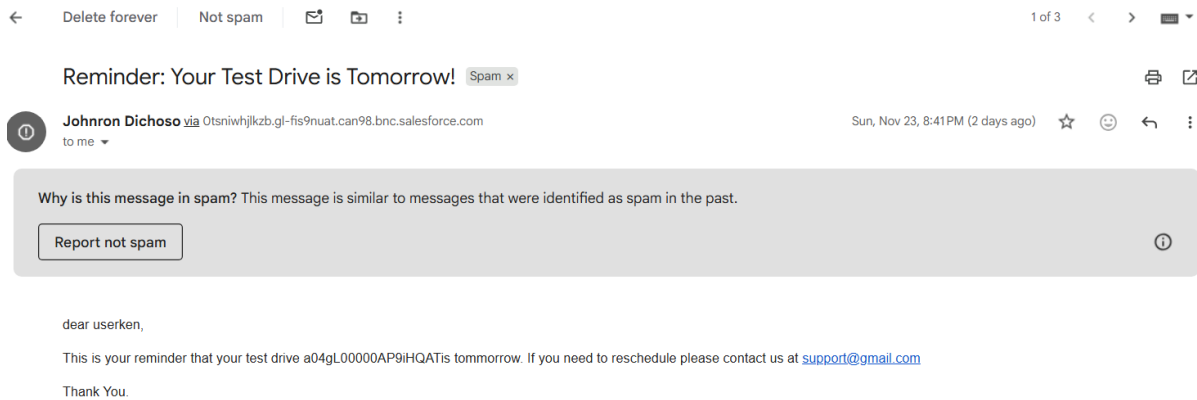


Figure 17: Email Reminder for Test Drive

### How it works:

- The Record-Triggered Flow initiates when a test drive record is created or changed.
- The Send Test Drive Reminder action automatically sends an email to the customer's email address stored in Vehicle\_Customer\_\_c.
- The email includes information about the test drive date and code.

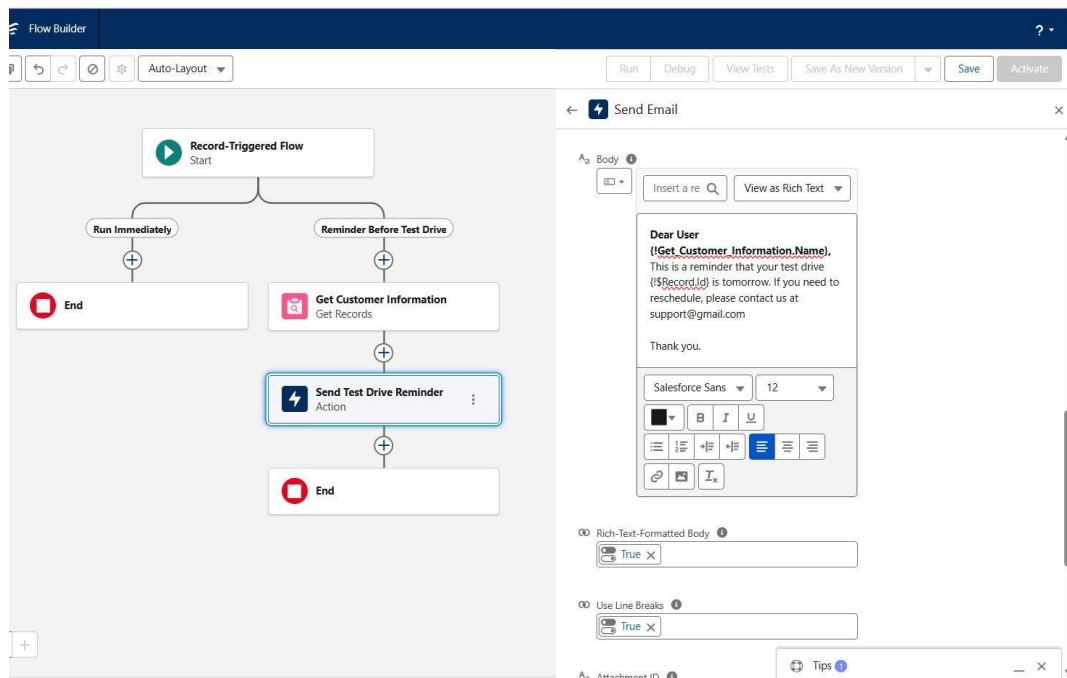


Figure 18: Record-Triggered Flow Displaying the Send Test Drive Reminder Email Action

## 10. Flows

WhatNext Vision Motors utilized Record-Triggered Flows in Salesforce to automate important dealership procedures and reduce human work. These flows conduct operations automatically depending on predefined parameters, ensuring data accuracy and timely notifications to both employees and customers.

For this project, two Record-Triggered Flows were created:

## 1. Auto Assigned Dealer Flow

- **Type:** Record-Triggered Flow
- **Purpose:** Automatically assigns a customer's vehicle order to the nearest dealer based on the customer's location.
- **Business Impact:** Improves operational efficiency and ensures faster order processing by connecting customers to the most convenient dealer.

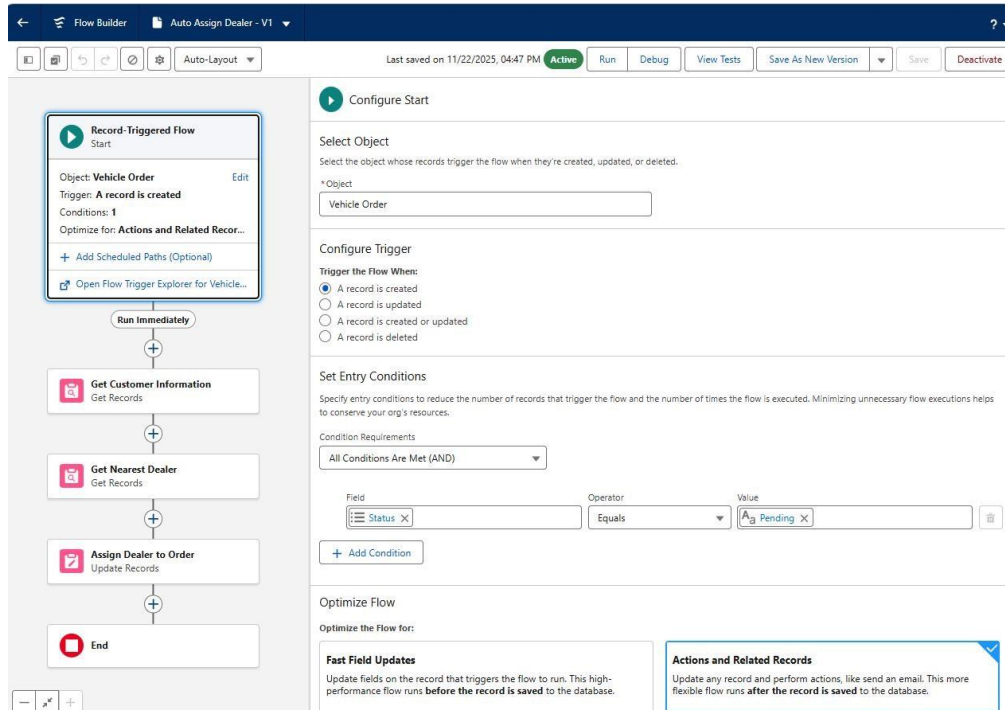


Figure 19: Auto Assign Dealers Record-Triggered Flow

## 2. Send Test Drive Reminder

- **Type:** Record-Triggered Flow
- **Purpose:** Sends an automated email reminder to customers one day before their scheduled test drive.
- **Business Impact:** Reduces missed appointments and enhances the customer experience by keeping them informed in advance.

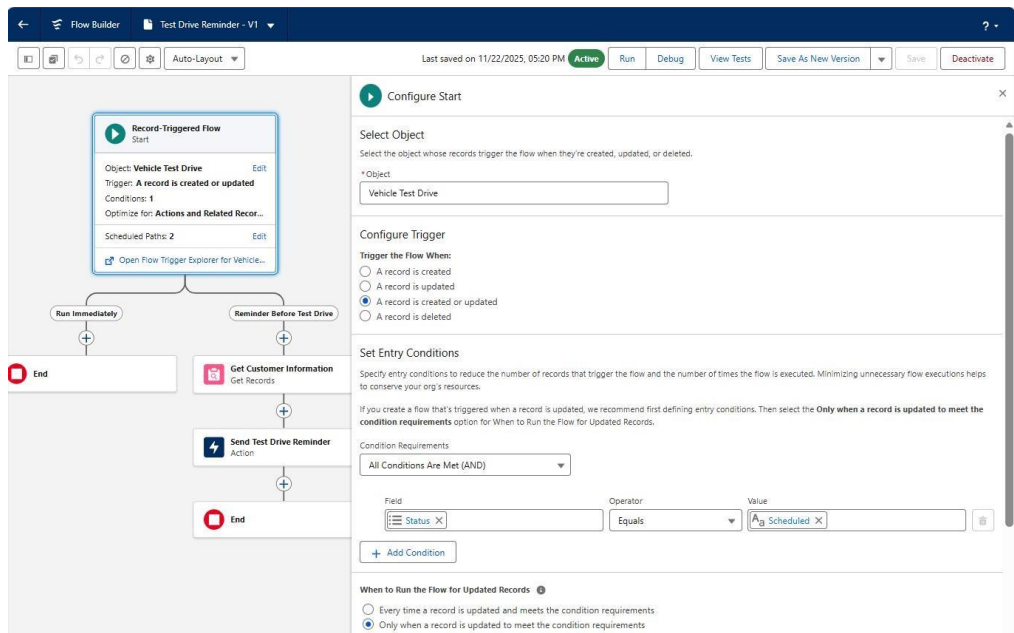


Figure 20: Test Drive Reminder Record-Triggered Flow

These Flows streamline dealership processes, keep data accurate, and provide timely notifications to both employees and customers.

## 11. Automation using Apex

Apex Triggers and Batch Apex were integrated into the WhatNext Vision Motors system to provide smooth backend processing and data consistency. These automations enforce crucial business standards, carry out key operations, and manage repetitive chores without the need for manual involvement, freeing up workers to focus on other important tasks.

### Apex Triggers

Two Apex components were created to handle order automation:

#### 1. VehicleOrderTriggerHandler

- **Purpose:** Contains the logic for validating stock and updating inventory when vehicle orders are placed or updated.
- **How it works:**
- Checks if vehicles are in stock before allowing an order to be placed.
- Automatically decreases the stock quantity of vehicles when an order is confirmed.

#### 2. VehicleOrderTrigger

- **Purpose:** Calls the VehicleOrderTriggerHandler to execute the trigger logic.
- **How it works:**
- Fires before and after an order is inserted or updated to ensure stock validation and automatic updates are applied.

### Batch Apex

Two Batch Apex classes were implemented for bulk processing:

#### 1. VehicleOrderBatch

- **Purpose:** Automatically processes pending vehicle orders in bulk.
- **How it works:** Checks all pending orders, confirms those with available stock, and updates stock quantities accordingly.

## 2. VehicleOrderBatchScheduler

- **Purpose:** Schedules the batch job to run automatically at set intervals.
- **How it works:** Executes the VehicleOrderBatch in batches of 50 records to efficiently manage multiple orders.

These Apex automations ensure accurate order processing, consistent stock management, and streamlined dealership operations, improving both efficiency and customer satisfaction.

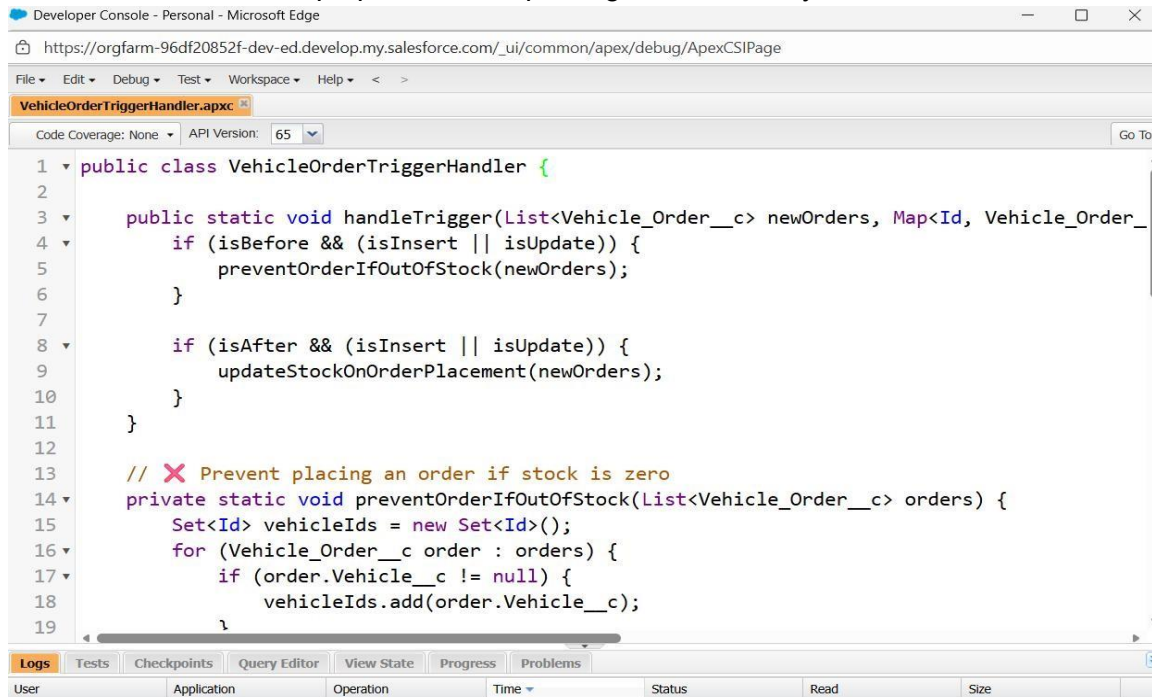
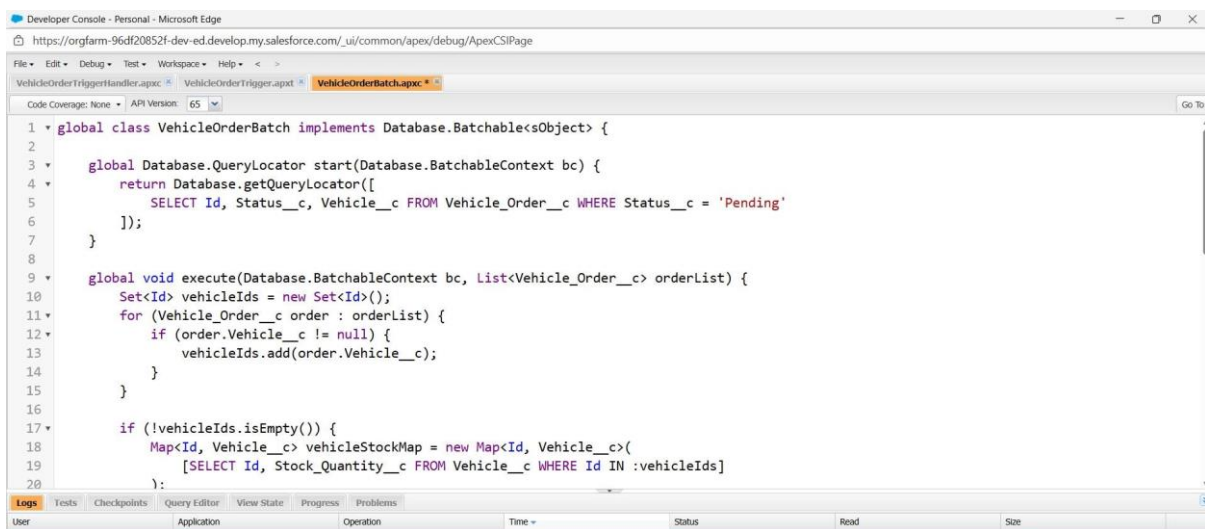


Figure 21: Vehicle Order Trigger Handler Code



Figure 22: Vehicle Order Trigger Code

The screenshot shows the Salesforce Developer Console with the 'VehicleOrderBatch.apxc' file open. The code defines a global class 'VehicleOrderBatch' that implements 'Database.Batchable<Object>'. It includes a 'start' method that returns a 'Database.QueryLocator' with a SOQL query for 'Vehicle\_Order\_\_c' where 'Status\_\_c' is 'Pending'. It also includes an 'execute' method that takes a 'List<Vehicle\_Order\_\_c>' and processes it by creating a set of vehicle IDs and a stock map, then executing a batch query for 'Vehicle\_Order\_\_c' based on these IDs.

```
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         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 }
```

Figure 23: Vehicle Order Batch Code

The screenshot shows the Salesforce Developer Console with the 'VehicleOrderBatchScheduler.apxc' file open. The code defines a global class 'VehicleOrderBatchScheduler' that implements 'Schedulable'. It includes an 'execute' method that creates a 'VehicleOrderBatch' object and calls 'Database.executeBatch' with a batch 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 }
```

Figure 24: Vehicle Order Batch Scheduler Code

## Project Explanation with real world Example

Let's walk thorough it like a real-world customer interaction.

### 1. Customer Registration

- A customer, Mark Lim, visits the dealership or website.
- In Salesforce: A record is created in the Vehicle\_Customer\_\_c object with his name, phone, email, and address.
- A Validation Rule ensures the email format is correct.

### 2. Vehicle Setup

- The admin adds vehicles into the Vehicle\_\_c object.
- Each vehicle includes details such as model, price, stock quantity, and assigned dealer.

### 3. Order Placement

- Mark places an order for a vehicle.
- In Salesforce, a new record is created in the Vehicle\_Order\_\_c object.
- The VehicleOrderTriggerHandler validates stock availability before confirming the order.

#### 4. Inventory Update

- As soon as the order is confirmed:
- **Apex Trigger** automatically decreases the stock of the selected vehicle. □ **Validation Rule** ensures stock never drops below zero.

#### 5. Dealer Assignment

- The Auto Assigned Dealer Flow automatically assigns Mark's order to the nearest dealer based on his location.
- This ensures faster processing and convenient customer service.

#### 6. Test Drive Reminder

- Mark schedules a test drive.
- One day before the test drive, the Send Test Drive Reminder Flow automatically sends him an email with the vehicle and dealer details.

#### 7. Order Status Updates

- Pending orders are periodically processed by the VehicleOrderBatch.
- Orders with available stock are automatically confirmed, and inventory is updated in bulk.
- **VehicleOrderBatchScheduler** ensures this process runs regularly without manual intervention.

#### 8. Customer Notifications

- Mark receives email notifications when his order is confirmed or test drive reminders are sent.
- This ensures clear communication and enhances his overall experience with the dealership.

---

## Screenshots

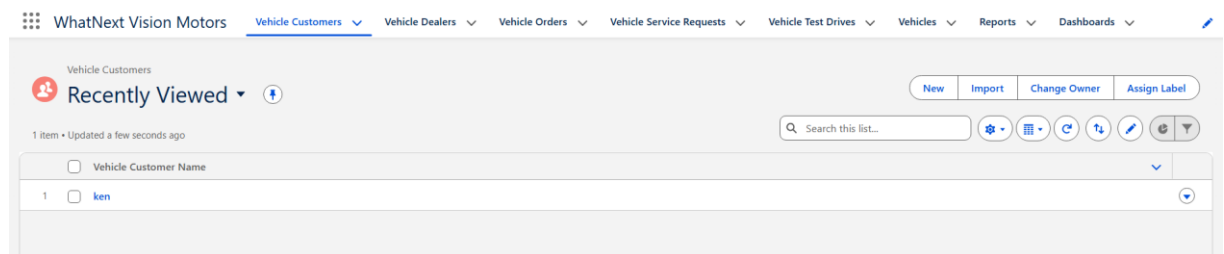



Figure 25: Custom App for WhatNext Vision Motors.

 Vehicle Customer

ken

Related

Details




<div>Vehicle Customer Name</div> <div>ken</div> <div></div>	<div>Owner</div> <div> <a href="#">Johnron Dichoso</a></div> <div></div>
<div>Email</div> <div><a href="mailto:dichosojohnron34@gmail.com">dichosojohnron34@gmail.com</a></div> <div></div>	
<div>Phone</div> <div><a href="tel:09649692648">09649692648</a></div> <div></div>	
<div>Address</div> <div>Philippines</div> <div></div>	
<div>Preferred Vehicle Type</div> <div>Sedan</div> <div></div>	
<div>Created By</div> <div> <a href="#">Johnron Dichoso</a>, 11/23/2025, 3:38 AM</div>	<div>Last Modified By</div> <div> <a href="#">Johnron Dichoso</a>, 11/23/2025, 3:38 AM</div>

Figure 26: Customer Creation

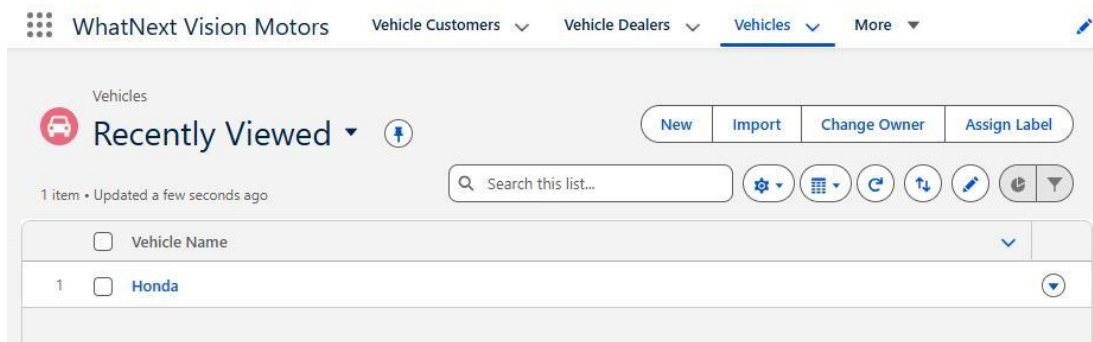


Figure 27: Available Vehicle

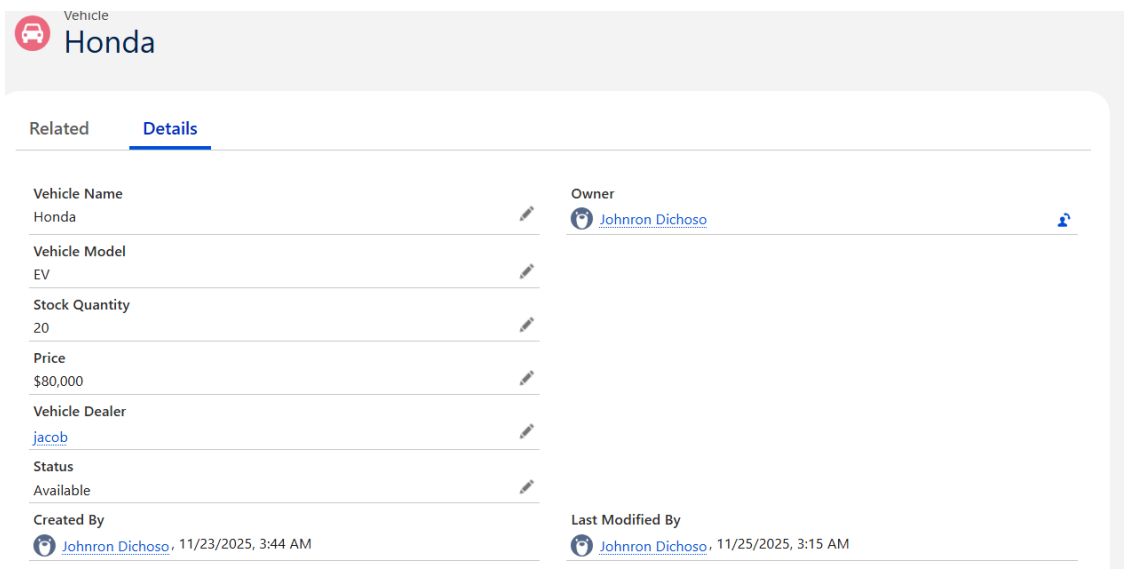



Figure 28: Vehicle Creation



Figure 29: List of Vehicle Dealers


**Vehicle Dealer**  
**jacob**

Related
Details






<b>Vehicle Dealer Name</b> jacob	<b>Owner</b>  <a href="#">Johnron Dichoso</a>
<b>Dealer Location</b> Hyderabad	
<b>Dealer Code</b> DC-0001	
<b>Phone</b> (123) 456-7890	
<b>Email</b> dichosojohnron34@gmail.com	
<b>Created By</b>  <a href="#">Johnron Dichoso</a> , 11/23/2025, 3:40 AM	<b>Last Modified By</b>  <a href="#">Johnron Dichoso</a> , 11/23/2025, 3:40 AM

Figure 30: Vehicle Dealers Creation


**WhatNext Vision Motors**

Vehicle Customers
Vehicle Dealers
Vehicle Orders
More


**Vehicle Orders**

Recently Viewed


New Import Change Owner Assign Label

2 items • Updated a few seconds ago

Search this list...

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

Figure 32: Vehicle Orders Overview


**Vehicle Order**  
**O-0002**

Related
Details




<b>Vehicle Order Number</b> O-0002	<b>Owner</b>  <a href="#">Johnron Dichoso</a>
<b>Vehicle Customer</b> <a href="#">ken</a>	
<b>Vehicle</b> <a href="#">Honda</a>	
<b>Order Date</b> 11/24/2025	
<b>Status</b> Confirmed	
<b>Assigned Dealer</b> <a href="#">jacob</a>	
<b>Created By</b>  <a href="#">Johnron Dichoso</a> , 11/23/2025, 4:58 AM	<b>Last Modified By</b>  <a href="#">Johnron Dichoso</a> , 11/25/2025, 2:36 AM

Figure 32: Vehicle Order Creation

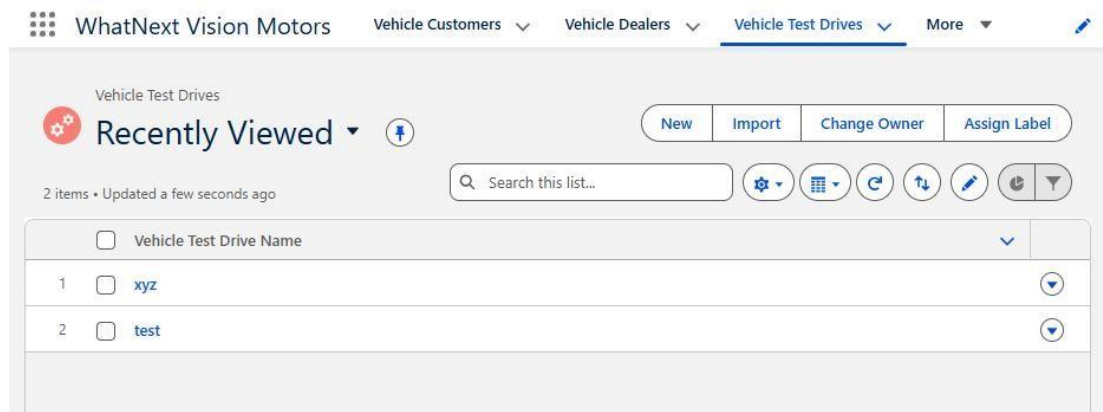


Figure 33: Vehicle Test Drives Overview

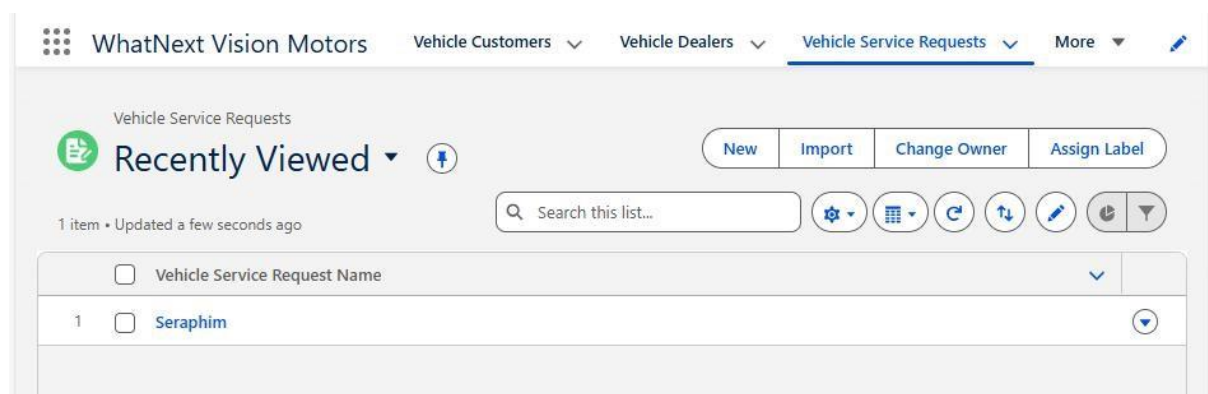
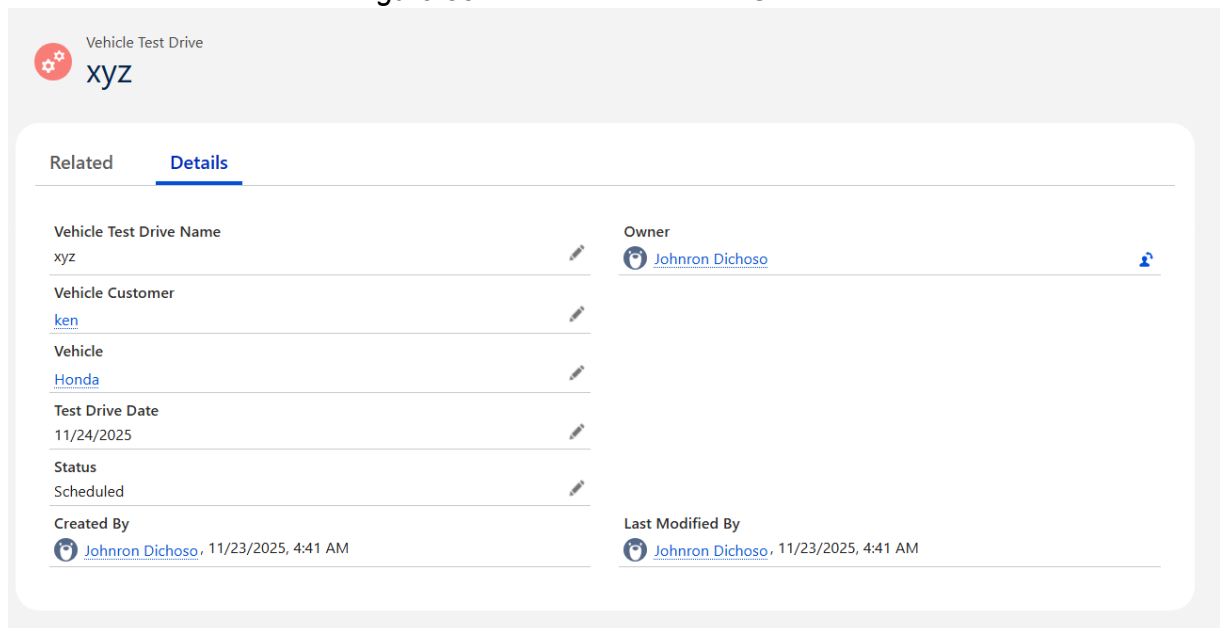


Figure 34: Vehicle Service Request Overview

## **Conclusion**

The WhatNext Vision Motors Salesforce project effectively showcases how a dealership may use technology to streamline operations, increase customer service, and decrease manual labor. By organising car, dealer, customer, and order data into structured objects, the system provides accurate record-keeping and efficient workflow.

Stock validation, dealer assignment, order confirmation, and test drive reminders are all automated using Apex Triggers, Batch Apex, and Flows, which reduces human error and improves operational efficiency. Customers receive timely notifications, which improves the entire experience and reliability.

Overall, the project illustrates Salesforce's ability to transform dealership operations. It not only improves productivity and accuracy for staff but also provides a more easy, transparent, and gratifying experience for customers, positioning WhatNext Vision Motors as a modern, customer-focused automotive company.

### **Future Scope and Enhancement**

The WhatNext Vision Motors system provides a solid foundation for dealership administration, with room for additional improvement. Future upgrades could include:

- Mobile integration enables clients to place orders, schedule test drives, and track orders using a mobile app.
- Utilizing analytics to track sales trends, car popularity, and customer behavior for better decision-making.
- Integrated secure payment channels for faster and more convenient online purchases.
- Improved notification options, including SMS and push notifications for order changes, test drives, and promotions.
- AI recommends autos based on consumer preferences and purchasing history.
- Service Tracking: Documenting vehicle upkeep and history for a better client experience.

Implementing these improvements will increase WhatNext Vision Motors' efficiency, customer friendliness, and competitiveness in the automobile business.