

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

ABSTRACT

This project presents the implementation of a customized Salesforce CRM solution for **WhatNext Vision Motors**, a forward-thinking automotive company dedicated to redefining customer experience and operational excellence. The objective was to automate the vehicle ordering process, enhance service efficiency, and ensure accurate stock and dealer management through intelligent workflows.

The solution involves designing a comprehensive data model featuring six key custom objects: **Vehicle, Dealer, Customer, Order, Test Drive, and Service Request**. Business processes were automated using **Record-Triggered Flows, Scheduled Flows, Email Alerts, Apex Triggers, and Batch Apex** to manage dealer assignment, order validations, stock updates, and customer test drive reminders.

To ensure data consistency and operational transparency, validation mechanisms and automated batch processes were established to handle order fulfillment and inventory updates. The system also features scheduled Apex jobs for periodic order processing, ensuring real-time accuracy and improved customer communication.

This end-to-end Salesforce CRM implementation significantly improves customer engagement through process automation, ensures operational efficiency by reducing manual interventions, and provides a scalable platform for sustainable business growth in the automotive sector.

OBJECTIVE

The main objective of this project is to develop and implement a customized Salesforce CRM solution for WhatNext Vision Motors to streamline core automotive operations, maintain accurate vehicle and order data, and enhance customer satisfaction through automation and data-driven efficiency.

By building a centralized system to manage vehicles, dealers, customers, orders, test drives, and service requests, the project aims to:

- **Automate key processes** such as order assignments, stock validation, and test drive reminders.

- **Ensure accurate and consistent data** through validation rules and real-time stock updates.
- **Enable real-time visibility** of vehicle availability, customer requests, and dealer performance.
- **Improve internal coordination** by automating dealer assignment and reducing manual workload.
- **Deliver enhanced customer experiences** through timely updates, transparent order status, and reliable service communication.

TECHNOLOGY DESCRIPTION

Salesforce

Salesforce is a cloud-based Customer Relationship Management (CRM) platform that helps organizations manage customers, streamline operations, and automate business processes across sales, service, and marketing. It provides both declarative (point-and-click) and programmatic tools (Apex, Flows) to build scalable solutions. In this project, Salesforce serves as the core platform to manage vehicles, customer requests, dealer assignments, and service workflows with high efficiency and accuracy.

Custom Objects

Custom Objects store business-specific data similar to database tables. They were used to represent major entities within WhatNext Vision Motors such as:

- **Vehicle_c** – Stores vehicle details and availability.
- **Vehicle_Dealer_c** – Stores dealer details and location.
- **Vehicle_Customer_c** – Stores customer information.
- **Vehicle_Order_c** – Stores and tracks customer vehicle orders.
- **Vehicle_Test_Drive_c** – Stores test-drive requests.
- **Vehicle_Service_Request_c** – Stores vehicle service requests.

These objects enable structured storage of business data and define relationships between vehicles, customers, dealers, and service requests.

Tabs

Tabs allow users to view and interact with object records through the Salesforce UI. For this project, tabs were created for key custom objects such as **Vehicle, Dealer, Customer, Order, Test Drive, and Service Request**, enabling quick navigation and efficient record management.

Custom App

A Salesforce Custom App organizes related tabs to support a specific purpose. A custom Lightning app titled “WhatNext Vision Motors” was developed. The app centralizes access to custom objects, reports, dashboards, and workflows, allowing users to seamlessly manage customer and vehicle operations.

Profiles

Profiles define user permissions such as access to objects, fields, and record operations. In this project, profiles ensured that administrators could manage records, while other users could only view or update data based on assigned access levels.

Roles

Roles determine record visibility based on the organization's data hierarchy. This allowed structured access to records, ensuring that sensitive dealership and customer records are shared only with the correct stakeholders.

Permission Sets

Permission Sets provide additional access without altering user profiles. They were used to grant extended capabilities to users who needed special permissions for specific tasks.

Validation Rules

Validation Rules ensure data integrity by enforcing business requirements. Examples include:

- Preventing negative stock quantities
- Ensuring proper field formatting (e.g., email format)

These rules reduced data entry errors and maintained consistency across vehicle and order records.

Email Templates

Email Templates are predefined message formats for communication. They were used to send automated email communications such as:

- Test drive reminders
- Order confirmation messages

This standardized customer messaging and reduced manual intervention.

Email Alerts

Email Alerts execute notifications based on automation such as Flows. In this project, they were triggered for:

- Vehicle test-drive reminders
- Order updates

These alerts ensured timely communication and improved customer engagement.

Flows

Flows automate business operations without requiring code. Several Record-Triggered and Scheduled Flows were implemented to:

- Assign orders to the nearest dealer

- Send test-drive reminders
- Update order information

These declarative automations streamlined operational activities and reduced manual workload.

Apex

Apex is Salesforce's object-oriented programming language used to automate complex logic. In this project, Apex was used to:

- Validate stock availability during order creation
- Reduce stock count on confirmed orders
- Execute scheduled batch processes

Apex triggers and classes enabled custom logic beyond declarative capabilities.

Apex Triggers

Triggers allow code execution before or after record operations. Project usage:

- Prevent order creation if a vehicle is out of stock (before insert/update)
- Decrease stock quantity when an order is confirmed (after insert/update)
- This ensured accurate inventory tracking in real time.

Batch Apex

Batch Apex enables processing of large datasets asynchronously. Here, it automatically:

- Checked stock for pending orders
- Confirmed orders when stock was replenished
- Updated vehicle inventory

This maintained order accuracy and supported large-scale processing efficiently.

Scheduled Apex

Scheduled Apex runs jobs at specific times. It was used to:

- Automatically schedule batch jobs daily
- Process pending vehicle orders
- Update order statuses based on stock

Scheduling automated backend updates ensured timely order fulfillment without manual input.

DETAILED EXECUTION OF PROJECT PHASES

1. Developer Org Setup

- A Salesforce Developer Org was created using:
 - <https://developer.salesforce.com/signup>
- The account was verified, a password was set, and access was granted

- to Salesforce Setup.
- Initial environment configuration was completed, enabling the creation of custom objects, apps, automation, and Apex components.

2. Custom Object Creation

Six custom objects were created to store business-critical data for WhatNext Vision Motors:

- Vehicle_c** – Stores vehicle details such as model, type, stock availability, and pricing.
- Vehicle_Dealer_c** – Stores dealer information including dealer name, location, and contact details.
- Vehicle_Customer_c** – Captures customer details such as name, contact info, and address.
- Vehicle_Order_c** – Tracks vehicle purchase requests, customer details, and status.
- Vehicle_Test_Drive_c** – Stores test drive booking information.
- Vehicle_Service_Request_c** – Tracks customer service and maintenance requests.

The screenshot shows the Salesforce Object Manager interface. At the top, there are tabs for Setup, Home, and Object Manager. The Object Manager tab is selected. A search bar at the top right contains the text "vehicle". Below the search bar are buttons for Schema Builder and Create. The main area displays a table of custom objects:

Label	API Name	Type	Description	Last Modified	Deployed
Vehicle	Vehicle_c	Custom Object		05/11/2025	✓
Vehicle Customer	Vehicle_Customer_c	Custom Object		05/11/2025	✓
Vehicle Dealer	Vehicle_Dealer_c	Custom Object		05/11/2025	✓
Vehicle Order	Vehicle_Order_c	Custom Object		05/11/2025	✓
Vehicle Service Request	Vehicle_Service_Request_c	Custom Object		05/11/2025	✓
Vehicle Test Drive	Vehicle_Test_Drive_c	Custom Object		05/11/2025	✓

3. Creating the Lightning App

- A custom Lightning App named WhatNext Vision Motors was created.
- Navigation items were added, including:
 - Vehicle, Dealer, Customer, Order, Test Drive, Service Request, Reports, Dashboards.
- The app was assigned to the System Administrator profile.
- This centralized workspace enabled efficient management of operational workflows.

The screenshot shows the 'App Settings' section of the Lightning App Builder. The left sidebar has tabs for 'App Details & Branding', 'App Options', 'Utility Items (Desktop Only)', and 'Navigation Items' (which is selected). The main area is titled 'Navigation Items' and contains a description: 'Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.' Below this is a 'Available Items' list with a search bar and a 'Create' button, containing items like Accounts, Action Hub, Activation Targets, Activations, All Sites, Alternative Payment Methods, Analytics, App Launcher, Appointment Categories, Appointment Invitations, and Approval Requests. To the right is a 'Selected Items' list with icons and names: Vehicle Customers, Vehicle Dealers, Vehicle Orders, Vehicle Service Requests, Vehicle Test Drives, Vehicles, Reports, and Dashboards. Navigation arrows between the lists allow items to be moved.

4. Validation Rules

Validation Rules were implemented to ensure accurate and clean data entry:

- Vehicle_Order__c
 - Prevent order creation when vehicle stock is unavailable.
 - Example error:
 - ◆ "This vehicle is out of stock. Order cannot be placed."
 - Vehicle__c
 - Prevent negative stock values.
 - Example error:
 - ◆ "Stock value must be zero or greater."
 - Vehicle_Customer__c
 - Ensure correct email format.
 - Example error:
 - ◆ "Please enter a valid email address."

These rules enforced business logic, ensured accurate stock data, and maintained customer information quality.

File Edit Debug Test Workspace Help < >

VehicleOrderTriggerHandler.apxc VehicleOrderTrigger.apxt VehicleOrderBatch.apxc VehicleOrderBatchScheduler.apxc

Code Coverage: None API Version: 65 Go To

```
31     }
32 }
33 }
34 }
35 }
36 // Decrease stock when an order is confirmed
37 private static void updateStockOnOrderPlacement(List<Vehicle_Order__c> orders) {
38     Set<Id> vehicleIds = new Set<Id>();
39     for (Vehicle_Order__c order : orders) {
40         if (order.Vehicle__c != null && order.Status__c == 'Confirmed') {
41             vehicleIds.add(order.Vehicle__c);
42         }
43     }
44 }
45 
46 if (!vehicleIds.isEmpty()) {
47     Map<Id, Vehicle__c> vehicleStockMap = new Map<Id, Vehicle__c>(
48         [SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds]
49     );

```

Logs Tests Checkpoints Query Editor View State Progress Problems

User	Application	Operation	Time	Status	Read	Size
France Shelby Valencia	Browser	/aura	11/7/2025, 10:44:10 PM	Success	Unread	309 bytes
France Shelby Valencia	Browser	/aura	11/7/2025, 8:27:07 PM	Success	Unread	7.8 kB
France Shelby Valencia	Unknown	common.api.soap.DirectSoap	11/7/2025, 8:27:07 PM	Success	Unread	521 bytes
France Shelby Valencia	Browser	/aura	11/7/2025, 8:25:57 PM	Success	Unread	16.35 kB
France Shelby Valencia	Unknown	common.api.soap.DirectSoap	11/7/2025, 8:25:57 PM	Success	Unread	521 bytes

Filter Click here to filter the log list

5. User Role & Profile Setup

- Standard User profile was cloned to create a new custom profile with

- access to custom objects.
- Roles were created to support Salesforce data visibility hierarchy:
 - Sales Manager
 - Inventory Manager
 - Service Coordinator
 - Marketing Team
- Profiles controlled functional access; roles controlled record visibility for reporting and sharing.

6. Record-Triggered Flow Development

Two primary Flows were implemented:

A. Dealer Auto-Assignment

- Object: Vehicle_Order__c
- Trigger: On record creation when order status = “Pending”
- Logic:
 - Retrieve customer record.
 - Identify nearest dealer based on customer address.
 - Update order with assigned dealer.
- Result: Faster and more accurate order routing.

B. Test Drive Reminder

- Object: Vehicle_Test_Drive__c
- Trigger: On record creation/update where status = “Scheduled”
- Scheduled Path: 1 day before Test_Drive_Date__c
- Logic:
 - Retrieve customer email
 - Send reminder email
- Result: Improved customer experience and higher attendance rate.

7. Apex Trigger & Handler Implementation

A central trigger handler was developed following best practices.

Key business logic executed:

- Prevent vehicle orders when stock is unavailable.
- Auto-deduct vehicle stock when order is confirmed.

Triggers executed:

- before insert / before update
 - Validated stock to prevent invalid order placement
- after insert / after update
 - Updated stock count appropriately

This ensured real-time stock synchronization and reduced ordering errors.

8. Batch Apex Job

A Batch Apex process was created to manage large-volume order updates.

Batch Job Capabilities:

- Scans all pending orders

- Verifies stock availability
- Automatically updates:
 - Order status → Confirmed
 - Vehicle stock → Deducted

Result: Accurate fulfillment processing and minimal manual intervention.

9. Scheduled Apex

Scheduled Apex runs the batch job periodically.

- Scheduling:
 - Frequency: Daily
 - Purpose:
 - ◆ Ensure timely order processing
 - ◆ Keep inventory accurate

This automated recurring process ensured operational consistency.

10. Reporting & Visibility

- Basic reports and dashboards were configured to track:
 - Vehicle availability
 - Dealer performance
 - Order fulfillment rate
 - Test drive statistics
- These visual assets supported decision-making and operational transparency.

PROJECT EXPLANATION WITH REAL-WORLD EXAMPLE

Let's walk through the Salesforce solution as if a real customer is interacting with WhatNext Vision Motors.

1. Customer Registration

- A new customer, John Reyes, visits WhatNext Vision Motors' website to explore available vehicles and service offerings.
- In Salesforce:
 - A record is created in Vehicle_Customer__c capturing:
 - Name
 - Email
 - Phone
 - Address
 - Preferred Vehicle Type
- A **Validation Rule** ensures the customer's email is formatted correctly (e.g., must be a valid email format).
- This customer record is now available for test drive booking, vehicle ordering, and service requests.

2. Vehicle Setup

- The admin configures vehicle data in Vehicle_c, including details such as:
 - Vehicle Name
 - Model Type (Sedan, SUV, EV, etc.)
 - Stock Quantity
 - Price
 - Status
- This setup enables sales representatives to view real-time availability and recommend suitable models to customers.

3. Dealer Assignment

- Multiple dealership locations are registered in Vehicle_Dealer_c, with details such as:
 - Dealer Name
 - Location
 - Contact Information
- When a customer places an order, a Record-Triggered Flow automatically identifies the nearest dealer based on the customer's address and assigns the order to that dealer.
- This ensures faster customer service and improved coordination between customer and dealer.

4. Vehicle Order Placement

- After reviewing available models, John places an order for an EV model.
- In Salesforce:
 - A new record is created in Vehicle_Order_c.
 - The system immediately checks if stock is available.
 - ◆ If in stock → Status = Pending
 - ◆ If not in stock → System prevents order creation with a validation message.
- Apex Trigger validates stock and ensures orders are only created for available models.
- This prevents false demand and ensures stock accuracy.

5. Order Confirmation & Stock Deduction

- Once the order reaches the “Confirmed” status:
 - Apex logic automatically reduces the Vehicle stock quantity by 1.
- If stock becomes zero, vehicle status is automatically updated to Out of Stock to prevent further orders.

6. Test Drive Scheduling

- John wants to test the vehicle before finalizing delivery.
- A **Vehicle_Test_Drive__c** record is created with:
 - Customer
 - Vehicle
 - Preferred Test Drive Date
- The status is set to Scheduled.
- A **Scheduled Flow** triggers:
 - Sends an automated email reminder 1 day before the test drive.
- This improves customer experience and reduces missed appointments.

7. Scheduled Batch: Pending Order Processing

- Some orders remain as “Pending” due to insufficient stock.
- When stock is replenished:
 - The Batch Apex job periodically checks pending orders and available stock.
 - If stock becomes available:
 - ◆ Order Status → Updated to Confirmed
 - ◆ Stock → Deducted accordingly
- The Scheduled Apex runs this batch every day to ensure automated fulfillment without manual intervention.

8. Service Request Management

- After several months, John submits a request for vehicle servicing.
- A record is created in **Vehicle_Service_Request__c** capturing:
 - Vehicle
 - Service Date
 - Issue Description
 - Status
- Service team processes the request and updates status to:
 - In Progress → Completed
- This creates full life-cycle tracking for every customer.

9. Monitoring & Reporting

- Administrators can view:
 - Vehicle stock status
 - Dealer performance
 - Order fulfillment lifecycle
 - No. of confirmed vs pending orders
 - Test drive summary
 - Service requests per month
- Dashboards and reports help the team with performance insights and improved decision-making.

10. Result

Through automated processes, the system:

- Reduces dealer assignment time
- Improves customer interaction
- Ensures accurate stock & order management
- Eliminates invalid orders
- Provides timely alerts
- Enhances customer satisfaction

SCREENSHOTS

The screenshot shows the 'Vehicle Customers' list page. At the top, there's a navigation bar with links for Vehicle Customers, Vehicle Dealers, Vehicle Orders, Vehicle Service Requests, Vehicle Test Drives, Vehicles, Reports, and Dashboards. Below the navigation is a search bar labeled 'Search...' and a toolbar with icons for New, Import, Change Owner, and Assign Label. A 'Recently Viewed' section shows a single item: 'John'. There's also a 'Search this list...' field and a set of filter and sort icons.

Fig: Vehicle Customers in WhatNext Vision Motors

The screenshot shows the 'Vehicle Dealers' list page. The layout is identical to the Vehicle Customers page, with a navigation bar, search bar, and toolbar at the top. The 'Recently Viewed' section shows two items: 'TM' and 'Em'. The main list area is currently empty.

Fig: Vehicle Dealers in WhatNext Vision Motors

The screenshot shows the 'Vehicle Orders' section of the software. At the top, there's a navigation bar with links for 'Vehicle Customers', 'Vehicle Dealers', 'Vehicle Orders', 'Vehicle Service Requests', 'Vehicle Test Drives', 'Vehicles', 'Reports', and 'Dashboards'. Below the navigation is a search bar and a toolbar with buttons for 'New', 'Import', 'Change Owner', and 'Assign Label'. A 'Recently Viewed' dropdown shows 'O-0002' and 'O-0001'. The main area displays a table with columns for 'Vehicle Order Number' and 'Status'. Two items are listed: 'O-0002' and 'O-0001'. Each item has a checkbox and a blue circular icon.

Fig: Vehicle Orders in WhatNext Vision Motors

The screenshot shows the 'Vehicle Test Drives' section. The navigation bar and search bar are identical to the previous screen. A 'Recently Viewed' dropdown shows 'xyz'. The main area displays a table with a single row for 'xyz', which has a checkbox and a blue circular icon.

Fig: Vehicle Test Drives in WhatNext Vision Motors

The screenshot shows the 'Vehicles' section. The navigation bar and search bar are identical. A 'Recently Viewed' dropdown shows 'Honda'. The main area displays a table with a single row for 'Honda', which has a checkbox and a blue circular icon.

Fig: Vehicles in WhatNext Vision Motors

CONCLUSION

The WhatNext Vision Motors solution built on Salesforce successfully streamlines core automotive operations such as customer management, dealer assignment, vehicle ordering, inventory tracking, test-drive appointments, and service request management. By leveraging Salesforce tools including Custom Objects, Record-Triggered Flows, Email Alerts, Validation Rules, Apex Triggers, Batch Apex, and Scheduled Apex, the system ensures accurate data entry, real-time stock updates, and enhanced communication across the customer lifecycle.

Through efficient automation and structured user access, the platform significantly reduces manual intervention, minimizes operational errors, and accelerates business processes. Intelligent workflows—such as automatic dealer assignment, preventive order checks for out-of-stock vehicles, and automated email reminders for test drives—improve customer satisfaction, while batch and scheduled processing ensure timely order fulfillment.

Overall, this Salesforce-based solution provides WhatNext Vision Motors with a scalable, data-driven platform that enhances customer experience, strengthens operational performance, and offers valuable insights into inventory, orders, and service requests. It supports future innovation while laying a strong technical foundation for long-term business growth.

1. Customer Portal Integration

- a) Implement an Experience Cloud portal where customers can log in, view orders, book test drives, and submit service requests.

2. Mobile App using Salesforce Mobile SDK

- a) Build a mobile application for staff to manage inventory, orders, and customer interactions, and for customers to explore vehicles and receive updates.

3. Advanced Reports & Dashboards

- a) Develop detailed dashboards to monitor vehicle inventory, order trends, dealer performance, and service insights in real time.

4. AI-Powered Recommendations

- a) Use Salesforce Einstein to suggest vehicles based on customer preference, predict service needs, and automate request handling.

5. ERP & Inventory System Integration

- a) Integrate with ERP to sync stock, streamline logistics, and automate spare-parts and supplier communication.

6. Telematics / IoT Integration

- a) Connect vehicle sensors to enable predictive maintenance, automatic service requests, and real-time vehicle monitoring.

7. Multi-Location Dealer Support

- a) Extend system capabilities to manage multiple dealerships, share orders, and provide unified test-drive scheduling.