

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

Project Overview

WhatNext Vision Motors' Salesforce-based CRM is designed to modernize the way the company handles vehicle orders and customer services. It provides a smoother customer experience by automating essential processes such as dealer recommendations, vehicle availability checks, and order management. The system also supports scheduling test drives and organizing service requests, making interactions faster and more convenient. All information related to customers, dealers, vehicles, orders, test drives, and service requests is stored in one centralized platform. This unified database allows the company to monitor every stage of the sales and service cycle with greater accuracy. Automated dealer assignment ensures customers are connected to the most suitable dealer without manual effort. Real-time stock checking helps prevent delays and ensures that vehicle availability is always up to date. Scheduled notifications keep both customers and staff informed about important updates and next steps. Batch updating of order statuses reduces repetitive tasks and helps maintain consistent and reliable records. Overall, the CRM increases efficiency, strengthens communication, improves customer satisfaction, and supports WhatNext Vision Motors' commitment to delivering innovative and seamless mobility solutions.

Objectives

The primary objective of this project is to design and deploy a customized Salesforce CRM solution for WhatNext Vision Motors: Shaping the Future of Mobility with Innovation and

Excellence to modernize the vehicle ordering workflow, automate critical operational processes, and enhance overall service effectiveness. By creating a centralized platform that manages customers, dealers, vehicles, orders, test drives, and service requests, the project aims to:

- streamline the vehicle ordering workflow through automated dealer assignment, order validation, and real-time status updates;
- improve stock visibility and accuracy to avoid out-of-stock orders and minimize processing delays;
- enhance the customer experience by enabling faster transactions, automated reminders, and clear, consistent communication;
- increase operational efficiency by consolidating essential processes into a single, integrated Salesforce system; and
- strengthen customer satisfaction and loyalty by delivering a smooth, transparent, and dependable service journey.

Phase 1: Requirement Analysis & Planning

- **Understanding Business Requirements:**
 - Capture and manage essential user information, including name, email, phone number, and address.
 - Identify inefficiencies in the existing processes for vehicle ordering, dealer management, and service requests.
 - Ensure accurate recording, organization, and easy accessibility of customer, vehicle, and order data.

- Implement secure, role-based access for sales teams, dealership managers, IT administrators, and other key stakeholders.
- **Defining Project Scope and Objectives:**
 - Scope: Develop a Salesforce-based CRM system for WhatNext Vision Motors
 - Objectives:
 - Streamline the vehicle ordering workflow.
 - Improve stock visibility and data accuracy.
 - Enhance the overall customer experience.
 - Increase operational efficiency through system integration.
 - Strengthen customer satisfaction and long-term loyalty.
- **Designing the Data Model and Security Model:**
 - Establish relationships among key objects such as Vehicle, Vehicle Dealer, Vehicle Customer, Vehicle Order, Vehicle Test Drive, and Vehicle Service Request.
 - Implement security components—including profiles, roles, and permission sets—to ensure proper access control.
- **Stakeholder Mapping:**
 - Primary stakeholders: Sales team, dealership managers, customers, and IT administrators.
 - Secondary stakeholders: Senior management and supporting staff.
- **Execution Roadmap:**
 - Prepare a detailed, phase-by-phase plan that covers analysis, development, testing, and deployment.

- Define clear milestones to track progress throughout the development and implementation stages.

Phase 2: Salesforce Development – Backend & Configurations

- **Setup Environment & DevOps Workflow:**

- Configured a Salesforce sandbox environment to support development activities.
- Set up version control and deployment workflows using tools such as VS Code and Git to ensure smooth, trackable development processes.

- **Customization of Objects, Fields, Validation Rules, and Automation:**

- Created custom objects including Vehicle, Vehicle Dealer, Vehicle Customer, Vehicle Order, Vehicle Test Drive, and Vehicle Service Request.
- Customized fields for each object, such as in the Vehicle object:

Vehicle_Name__c (Text)

Vehicle_Model__c (Picklist: Sedan, SUV, EV, etc.)

Stock_Quantity__c (Number)

Price__c (Currency)

Dealer__c (Lookup to Dealer__c)

Status__c (Picklist: Available, Out of Stock, Discontinued)

- Implemented validation rules to ensure data quality and accuracy.
- Configured automation using workflow rules, Process Builder, and Salesforce Flows to streamline processes and reduce manual work.

- **Apex Classes, Triggers, and Asynchronous Apex:**

- Built Apex classes and triggers to handle and automate backend processes.

- **Documentation:**

- Included screenshots of the Apex classes and triggers.

```

1 // Clase de servicio para manejar los vehículos.
2 // Este servicio se utiliza para crear, actualizar, eliminar y obtener vehículos.
3
4 // Importación de la API de Salesforce
5 import System;
6
7 // Definición de la clase de servicio
8 public class VehicleService {
9
10     // Método para crear un vehículo
11     public static Vehicle createVehicle(Vehicle__c vehicle) {
12         // Creación de un nuevo registro de vehículo
13         return [insert vehicle];
14     }
15
16     // Método para actualizar un vehículo
17     public static Vehicle updateVehicle(Vehicle__c vehicle) {
18         // Actualización de un registro de vehículo
19         return [update vehicle];
20     }
21
22     // Método para eliminar un vehículo
23     public static void deleteVehicle(Vehicle__c vehicle) {
24         // Eliminación de un registro de vehículo
25         delete vehicle;
26     }
27
28     // Método para obtener un vehículo por ID
29     public static Vehicle getVehicleById(Id vehicleId) {
30         // Obtención de un registro de vehículo por ID
31         return [select * from Vehicle__c where Id = :vehicleId];
32     }
33
34 }

```

```

1 // Trigger para manejar los vehículos.
2 // Este trigger se ejecuta antes de insertar o actualizar un vehículo.
3
4 // Importación de la API de Salesforce
5 import System;
6
7 // Definición del trigger
8 trigger VehicleTrigger on Vehicle__c before insert or before update {
9
10     // Lógica del trigger
11     for (Vehicle__c vehicle : Trigger.new) {
12         // Verificación de la condición
13         if (vehicle.Status__c == 'Inactivo') {
14             // Llamada al servicio para actualizar el vehículo
15             VehicleService.updateVehicle(vehicle);
16         }
17     }
18 }

```

```

1 // Clase de controlador para manejar los vehículos.
2 // Este controlador se utiliza para crear, actualizar, eliminar y obtener vehículos.
3
4 // Importación de la API de Salesforce
5 import System;
6
7 // Definición de la clase de controlador
8 public class VehicleController {
9
10     // Método para crear un vehículo
11     public static Vehicle createVehicle(Vehicle__c vehicle) {
12         // Creación de un nuevo registro de vehículo
13         return [insert vehicle];
14     }
15
16     // Método para actualizar un vehículo
17     public static Vehicle updateVehicle(Vehicle__c vehicle) {
18         // Actualización de un registro de vehículo
19         return [update vehicle];
20     }
21
22     // Método para eliminar un vehículo
23     public static void deleteVehicle(Vehicle__c vehicle) {
24         // Eliminación de un registro de vehículo
25         delete vehicle;
26     }
27
28     // Método para obtener un vehículo por ID
29     public static Vehicle getVehicleById(Id vehicleId) {
30         // Obtención de un registro de vehículo por ID
31         return [select * from Vehicle__c where Id = :vehicleId];
32     }
33
34 }

```

```

1 // Trigger para manejar los vehículos.
2 // Este trigger se ejecuta después de insertar o actualizar un vehículo.
3
4 // Importación de la API de Salesforce
5 import System;
6
7 // Definición del trigger
8 trigger VehicleTrigger on Vehicle__c after insert or after update {
9
10     // Lógica del trigger
11     for (Vehicle__c vehicle : Trigger.new) {
12         // Llamada al servicio para actualizar el vehículo
13         VehicleService.updateVehicle(vehicle);
14     }
15 }

```

Phase 3: UI/UX Development & Customization

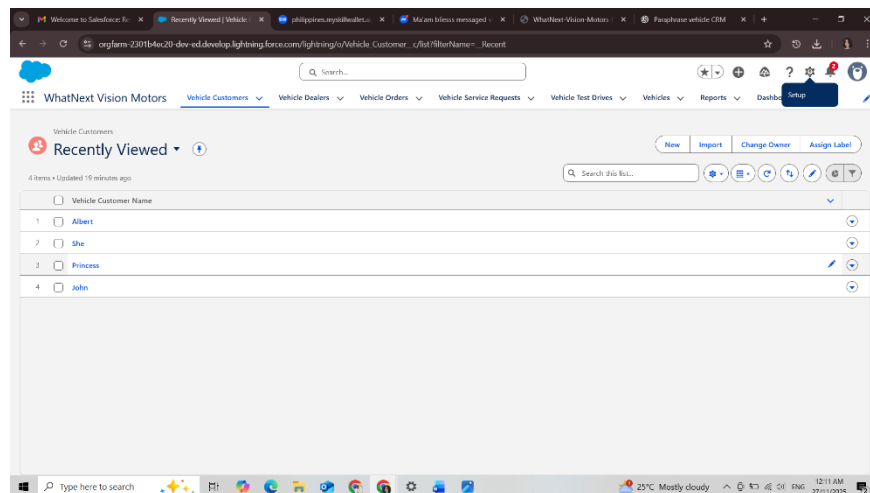
- **Lightning App Setup:**

- Created a personalized Lightning application titled WhatNext Vision Motors.

- **Page Layouts and Dynamic Forms:**

- Arrange fields, buttons, and related lists systematically.
- Customized layouts based on user roles for the Vehicle, Dealer, Customer, Order, Test Drive, and Service Request records.
- Applied dynamic forms for a more intuitive and flexible user interface.

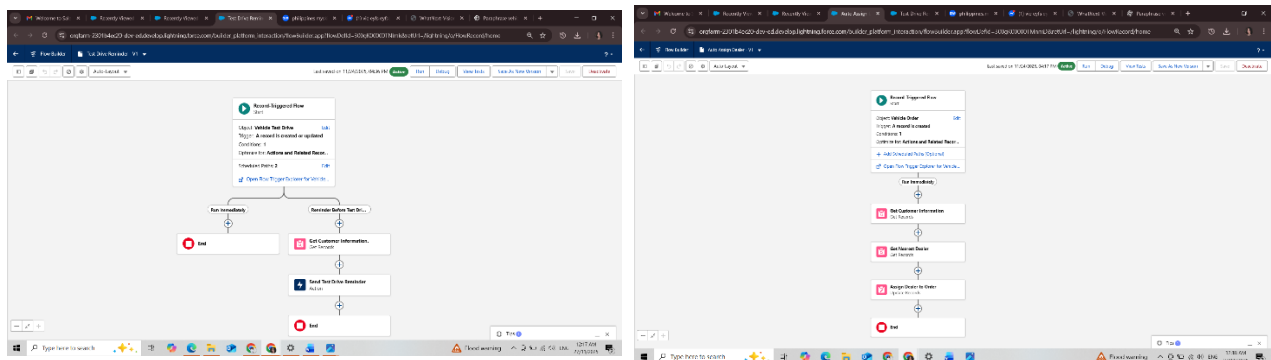
- **User Management:**
 - Established profiles, roles, and permission sets to define user permissions and data access.
- **Reports and Dashboards:**
 - Developed real-time dashboards to monitor sales and track performance.
- **LWC Development (if applicable):**
 - Built Lightning Web Components to support advanced UI features.
- **Lightning Pages:**
 - Customized Lightning Pages to simplify navigation and improve user experience.
- **Documentation:**



Phase 4: Data Migration, Testing & Security

- **Data Loading Process:**
 - Manual record entry demonstrated in the video.

- **Field History, Duplicate and Matching Rules:**
 - Enabled tracking for key fields to monitor important changes.
 - Set up a rule to notify users when a new Lead matches an existing Contact by email or phone.
- **Profiles, Roles, Permission Sets, and Sharing Rules:**
 - Maintained secure access and proper data visibility.
- **Creation of Test Classes:**
 - Developed Apex test classes to ensure code runs correctly.
- **Documentation:**
 - Screenshots of the flows named Auto Assign Dealer and Test Drive Reminder.



Phase 5: Deployment, Documentation & Maintenance

- **Deployment Strategy:**
 - Applied changes via Salesforce change sets alongside metadata deployment tools.
- **System Maintenance & Monitoring:**
 - Conducted regular updates, backups, and system performance monitoring

- Provided continuous user support and issue resolution.

- **Troubleshooting Documentation:**

- Offered a step-by-step guide to identify and resolve common issues, including reviewing flows, automation, and analyzing Apex debug logs.

Conclusion:

The Salesforce CRM solution for WhatNext Vision Motors effectively modernized the vehicle ordering workflow, automated critical processes, and enhanced overall operational efficiency. By centralizing information for customers, dealers, and vehicles, the system improved decision-making, minimized manual errors, and established a scalable foundation for future expansion. Looking ahead, potential enhancements may include integrating advanced marketing automation, incorporating AI-driven analytics for more accurate sales forecasting, and developing a mobile-friendly interface for convenient on-the-go access. Additionally, expanding reporting and dashboard capabilities would offer deeper insights into customer behavior and dealer performance. These improvements will ensure that the CRM remains adaptable, forward-thinking, and aligned with the evolving needs of the company.